VirtEngine-Waldur Documentation

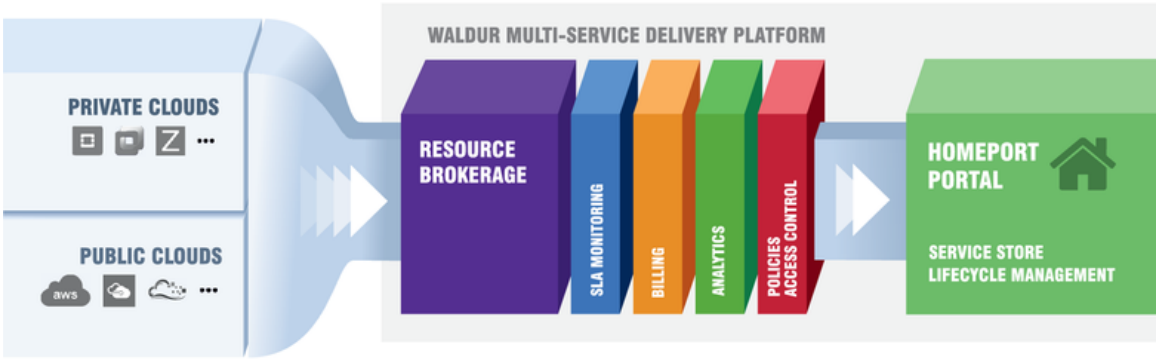
Welcome to VirtEngine-Waldur documentation!

VirtEngine-Waldur is a platform for managing hybrid cloud resources. It is used to control both internal enterprise IT resources and for selling cloud to the public. VirtEngine-Waldur is composed of the following main components:

- VirtEngine-Waldur MasterMind - broker and orchestrator of cloud services. Responsible for technical service delivery and connected matte Exposes REST API for management
- VirtEngine-Waldur HomePort - web-based self-service portal. Talks REST to MasterMind.

VirtEngine-Waldur is open-source, extendable and comes with a professional support provided by DET-IO & OpenNode

> Please note that wiki documentation setup is in progress. We are adding / moving / integrating contents as we go. If there's a particular topic you feel should be covered, please drop a line to hello@virtengine.com.



Recently Updated

Changelog
about 7 hours ago • updated by Ilja Livenson • view change

Cookie policy
Aug 07, 2018 • created by Ilja Livenson

JIRA Service Desk configuration
Aug 05, 2018 • updated by Ilja Livenson • view change

Pre-3.0.0
Aug 05, 2018 • updated by Ilja Livenson • view change

Deployment
Jul 30, 2018 • updated by Ilja Livenson • view change

Setting up system notifications
Jul 25, 2018 • updated by Maxim Prokopenko • view change

Screenshot2.png
Jul 25, 2018 • attached by Maxim Prokopenko

Screenshot.png
Jul 23, 2018 • attached by Maxim Prokopenko

Notification about owners profile changes
Jul 13, 2018 • updated by Ilja Livenson • view change

MasterMind configuration
Jun 21, 2018 • updated by Victor Mireyev • view change

Invoice configuration
Jun 21, 2018 • updated by Victor Mireyev • view change

Support configuration
Jun 21, 2018 • updated by Victor Mireyev • view change

HomePort configuration
Jun 11, 2018 • updated by Ilja Livenson • view change

Request-based offerings
Jun 07, 2018 • updated by Victor Mireyev • view change

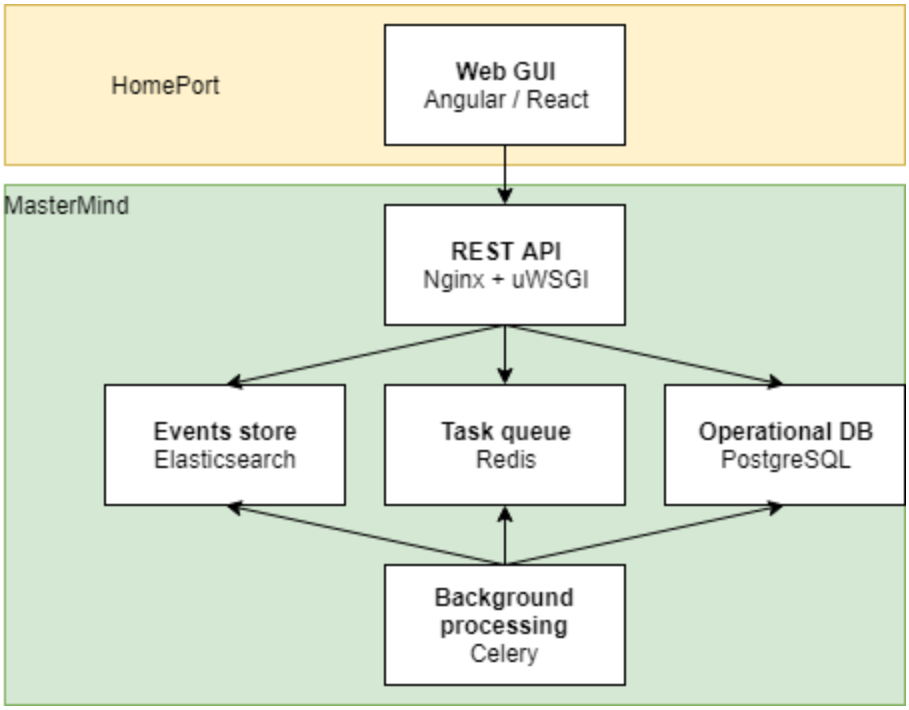
Hardware requirements
May 25, 2018 • updated by Ilja Livenson • view change

Administration Guide

Architecture

Modules diagram

Component diagram



Deployment

Hardware requirements

Please refer to Hardware requirements. In particular, all-in-one setup: Hardware requirements#All-in-one

OS requirements

VirtEngine-Waldur is packaged for the RHEL7-compliant operating systems. All of components are distributed as RPM through YUM repositories.

OS is expected to have SELinux set into non-enforcing mode for MasterMind.

> **RHEL Flavor**
> Please note that instructions below are for CentOS7! Other flavors of RHEL, including RHEL7, might have differences in installation process.

VirtEngine-Waldur MasterMind installation

MasterMind is an assembly of components delivering a broker functionality. Historically the base components were called NodeConductor, name migration is done, however some leftovers might exist. Below is a description of the basic installation process on a vanilla CentOS 7.

Set up database backend

Install PostgreSQL server (version 9.5 or later is required):

```
# CentOS
yum install
https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/p
gdg-centos96-9.6-3.noarch.rpm
# RHEL
yum install
https://download.postgresql.org/pub/repos/yum/9.6/redhat/rhel-7-x86_64/p
gdg-redhat96-9.6-3.noarch.rpm

# All
yum install postgresql96-server
/usr/pgsql-9.6/bin/postgresql96-setup initdb
```

Start PostgreSQL server and enable it to run on startup:

```
systemctl start postgresql-9.6
systemctl enable postgresql-9.6
```

Create database for VirtEngine-Waldur MasterMind:

```
su - postgres -c "/usr/pgsql-9.6/bin/createdb -EUTF8 waldur"
su - postgres -c "/usr/pgsql-9.6/bin/createuser waldur"
```

Set up queue and cache backend

Install Redis server:

```
yum install epel-release
yum install redis
```

Start Redis server and enable it to run on startup:

```
systemctl start redis
systemctl enable redis
```

Set up events backend

Install Elasticsearch and Logstash:

```
yum install https://opennodecloud.com/centos/7/elastic-release.rpm
yum install elasticsearch java-1.8.0-openjdk-headless logstash
```

Configure Logstash:

File: /etc/logstash/conf.d/input.conf

```
input {
  tcp {
    codec => json
    port => 5959
    type => "waldur-event"
  }
}
```

File: /etc/logstash/conf.d/filter.conf

```
filter {
  if [type] == "waldur-event" {
    json {
      source => "message"
    }

    mutate {
      remove_field => ["class", "file", "logger_name", "method", "path",
"priority", "thread"]
    }

    grok {
      match => {
        "host" => "%{IPORHOST:host}:%{POSINT}"
      }
      overwrite => ["host"]
    }
  }
}
```

File: /etc/logstash/conf.d/output.conf

```
output {
  elasticsearch { }
}
```

Start Elasticsearch (version 5.4 is required) and Logstash (version 5.4 is required) and enable them to run on startup:

```
systemctl start elasticsearch
systemctl enable elasticsearch

systemctl start logstash
systemctl enable logstash
```

Set up VirtEngine-Waldur MasterMind

Install VirtEngine-Waldur MasterMind:

```
# CentOS
yum install centos-release-openstack-pike epel-release
# RedHat
yum install
https://repos.fedorapeople.org/repos/openstack/openstack-pike/rdo-releas
e-pike-1.noarch.rpm
yum install
https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
subscription-manager repos --enable=rhel-7-server-optional-rpms
--enable=rhel-7-server-extras-rpms --enable=rhel-7-server-rh-common-rpms

# All
yum install http://opennodecloud.com/centos/7/waldur-release.rpm
yum install waldur-mastermind
```

Initialize VirtEngine-Waldur database:

```
su - waldur -c "waldur migrate --noinput"
```

Start VirtEngine-Waldur MasterMind services and enable them to run on startup:

```
systemctl start waldur-celery
systemctl enable waldur-celery

systemctl start waldur-celerybeat
systemctl enable waldur-celerybeat

systemctl start waldur-uwsgi
systemctl enable waldur-uwsgi
```

VirtEngine-Waldur MasterMind configuration
Default configuration file location for waldur-celery, waldur-celerybeat and waldur-uwsgi processes in /etc/waldur/. If you wish to customize base location of the configuration files, you can pass WALDUR_BASE_CONFIG_DIR=/path/to/new/config/base variable to the correponding processes.

Set up VirtEngine-Waldur HomePort

Hostname
The instructions below assume that you will be accessing VirtEngine-Waldur HomePort from 'http://localhost/'. If you are using a different ir hostname, please update it in:

- /etc/nginx/conf.d/waldur.conf
- /etc/waldur-homeport/config.json (set "apiEndpoint" to a corresponding value).

Install VirtEngine-Waldur HomePort:

```
yum install http://opennodecloud.com/centos/7/waldur-release.rpm
yum install waldur-homeport
```

Set up webserver:

```
yum install epel-release
yum install nginx
```

Configure Nginx:

File: /etc/nginx/conf.d/waldur.conf

```
server {
    # CHANGEME set the name or IP of a server from where VirtEngine-Waldur
is accessed.
    server_name localhost;

    # VirtEngine-Waldur API
    location / {
```

```
        add_header 'Access-Control-Allow-Credentials' 'true' always;
        add_header 'Access-Control-Allow-Headers' 'Accept,
Accept-Encoding, Authorization, Content-Type, Origin, User-Agent,
X-CSRFToken, X-Requested-With' always;
        add_header 'Access-Control-Allow-Methods' 'DELETE, GET, OPTIONS,
PATCH, POST, PUT' always;
        add_header 'Access-Control-Allow-Origin' "$http_origin" always;
        add_header 'Access-Control-Expose-Headers' 'Link,
X-Result-Count' always;

        include uwsgi_params;
        uwsgi_param Host $host;
        uwsgi_param X-Forwarded-For $proxy_add_x_forwarded_for;
        uwsgi_param X-Forwarded-Proto $http_x_forwarded_proto;
        uwsgi_param X-Real-IP $remote_addr;

        uwsgi_pass unix:/run/waldur/uwsgi/uwsgi.sock;
    }

    # VirtEngine-Waldur HomePort
    location /homeport {
        alias /usr/share/waldur-homeport;
        index index.html;
    }

    location /homeport/scripts/configs/config.json {
        alias /etc/waldur-homeport/config.json;
    }

    location /homeport/login-logo.png {
        # Make sure to set 'loginLogo' value to 'login-logo.png' in
        # /etc/waldur-homeport/config.json -- check VirtEngine-Waldur Home
config
        # docs for details.
        alias /etc/waldur-homeport/login-logo.png;
    }

    location /homeport/sidebar-logo.png {
        # Make sure to set 'sidebarLogo' value to 'sidebar-logo.png' in
        # /etc/waldur-homeport/config.json -- check VirtEngine-Waldur Home
config
        # docs for details.
```

```
        alias /etc/waldur-homeport/sidebar-logo.png;
    }
}
```

Start Nginx and enable it to run on startup:

```
systemctl start nginx
systemctl enable nginx
```

Build VirtEngine-Waldur VM

Install Packer: https://www.packer.io/intro/getting-started/install.html

Clone waldur-packer repository:

```
git clone https://github.com/virtengine/waldur-packer.git
cd waldur-packer
```

Build VirtEngine-Waldur VM:

```
packer build waldur.json
```

Built VM image can be found in `output-qemu/waldur.qcow2`.

Introducing LXD on Ubuntu 16.04 LTS to build VirtEngine-Waldur VM

If LXD is not already initialized on the Ubuntu 16.04 LTS host

```
sudo apt-get update
sudo apt-get install lxd zfsutils-linux
sudo lxd init
Name of the storage backend to use (dir or zfs): zfs
Create a new ZFS pool (yes/no)? yes
Name of the new ZFS pool: lxdpool
Would you like to use an existing block device (yes/no)? no
Size in GB of the new loop device (1GB minimum): 100
Would you like LXD to be available over the network (yes/no)?
no
Do you want to configure the LXD bridge (yes/no)? yes
Warning: Stopping lxd.service, but it can still be activated
by: lxd.socket
LXD has been successfully configured.
```

Preparing container to build VirtEngine-Waldur VM

```
lxc launch ubuntu:16.04/amd64 builddemowaldurubuntu1604lxc
lxc config device add builddemowaldurubuntu1604lxc kvm unix-char
path=/dev/kvm
lxc exec builddemowaldurubuntu1604lxc -- bash
```

Start VirtEngine-Waldur VM

Get VirtEngine-Waldur VM

Option 1: Download VM image

URL: http://opennodecloud.com/waldur-vms/waldur.qcow2

Option 2: Build with Packer

See Build VirtEngine-Waldur VM for details.

Start new VM

VirtualBox

Install QEMU: https://www.qemu.org/download/

Convert QCOW2 image to VDI format that VirtualBox can read:

```
qemu-img convert -f qcow2 -O vdi waldur.qcow2 waldur.vdi
```

Install VirtualBox: https://www.virtualbox.org/

Start VirtualBox

> *New*
>
> *Name and operating system*
>
> *Name: (set VM name)*
>
> *Type: Linux*
>
> *Version: Red Hat (64 bit)*
>
> *Memory size*
>
> *2048 MB*

> *Hard disk*
>
> *[X] Use an existing virtual hard disk drive*
>
> *Choose a virtual hard disk file... > (path to waldur.vdi file)*

Select and start created VM.

Virtual Machine Manager (virt-manager)

Install Virtual Machine Manager: https://virt-manager.org/

Start Virtual Machine Manager

> *New VM*
>
> *Step 1 of 4*
>
> *Choose how you would like to install the operating system: [X] Import existing disk image*
>
> *Step 2 of 4*
>
> *Provide the existing storage path: <path-to-waldur.qcow2>*
>
> *OS type: Linux*
>
> *Version: CentOS 7.0*
>
> *Step 3 of 4*
>
> *Memory (RAM): 2048 MiB*
>
> *CPUs: 2*
>
> *Step 4 of 4*
>
> *Name: VirtEngine-Waldur*

QEMU

Install QEMU: https://www.qemu.org/download/

Start VirtEngine-Waldur VM:

```
qemu-system-x86_64 -m 4096 waldur.qcow2
```

If you are using KVM hypervisor you may want to add `-machine accel=kvm` parameter.

You may also want to add `-redir tcp:8080::80` paramater to access Wladur services from the same host.

Once the VM is started, update `apiEndpoint` value in VirtEngine-Waldur HomePort configuration:

File: /etc/waldur-homeport/config.json

```
{                                          |
  ...                                      | Set this to
  "apiEndpoint": "http://localhost/"  <--|  - "http://locaohost:8080/"
if you mapped VM ports to localhost
  ...                                      |  - "http://<vm-ip-address>/"
if you are using bridged networking
}                                          |
```

Once the VM is started you can access VirtEngine-Waldur services on http://localhost:8080 (or http://<vm-ip-address> if you are using bridged net

Pre-created VirtEngine-Waldur data

System user:

- Login: `root`, password: `password`

VirtEngine-Waldur users:

- Login: `admin`, password: `admin`
- Login: `a.owner`, password: `Password1`
- Login: `a1.admin`, password: `Password1`
- Login: `a1.manager`, password: `Password1`
- Login: `a2.admin`, password: `Password1`
- Login: `a2.manager`, password: `Password1`
- Login: `b.owner`, password: `Password1`
- Login: `b1.admin`, password: `Password1`
- Login: `b1.manager`, password: `Password1`
- Login: `b2.admin`, password: `Password1`
- Login: `b2.manager`, password: `Password1`
- Login: `support`, password: `Password1`

VirtEngine-Waldur organizations:

- Name: `Organization A`, owner: `a.owner`
- Name: `Organization B`, owner: `b.owner`

VirtEngine-Waldur projects:

- Name: `Project A1`, customer: `Organization A`, admin: `a1.admin`, manager: `a1.manager`
- Name: `Project A2`, customer: `Organization A`, admin: `a2.admin`, manager: `a2.manager`
- Name: `Project B1`, customer: `Organization B`, admin: `b1.admin`, manager: `b1.manager`
- Name: `Project B2`, customer: `Organization B`, admin: `b2.admin`, manager: `b2.manager`

https://github.com/virtengine/ve-waldur-packer/blob/master/init-data.sh

Upgrade

Release upgrade

VirtEngine-Waldur MasterMind

This instructions demonstrate VirtEngine-Waldur MasterMind upgrade using Nginx HTTP server. Similar steps can be performed for other HTTP s
such as Apache.

Before upgrading

It is a good practice to perform configuration test for Nginx server before upgrading:

```
nginx -t
```

Current VirtEngine-Waldur MasterMind packages' versions can be saved in case rollback would be needed after the upgrade.

Following command can be used to list installed VirtEngine-Waldur MasterMind packages:

```
yum list installed 'waldur*'
```

Upgrade

Stop running services

```
systemctl status nginx && systemctl stop nginx
systemctl status waldur-celery && systemctl stop waldur-celery
systemctl status waldur-celerybeat && systemctl stop waldur-celerybeat
systemctl status waldur-uwsgi && systemctl stop waldur-uwsgi
```

Dump PostgreSQL database data

```
sql_dump_file="/var/lib/waldur/dbdump-$(date '+%Y%m%dT%H%M%S').sql.gz"
su - waldur -c '/usr/pgsql-9.6/bin/pg_dump waldur' | gzip -9 >
"$sql_dump_file"
chmod 600 "$sql_dump_file"
```

This command will dump data from `waldur` database and create a `/var/lib/waldur/dbdump-<datetime>.sql.gz` file. Afterwards the file can be used to restore the database state.

Install latest VirtEngine-Waldur MasterMind packages

```
yum clean all
yum -y update waldur-mastermind
```

Update database schemas

```
su - waldur -c "waldur migrate --noinput"
```

Start VirtEngine-Waldur MasterMind services

```
systemctl start waldur-celery
systemctl start waldur-celerybeat
systemctl start waldur-uwsgi
nginx -t && systemctl start nginx
```

HomePort

Install latest VirtEngine-Waldur HomePort package

```
yum clean all
yum -y update waldur-homeport
```

If interface changes are not visible, try to remove cache for the page in browser (for Chrome users hit Ctrl + Shift + R on the VirtEngine-Waldur Ho page).

Smoke tests

Backend system test for VirtEngine-Waldur MasterMind

[localhost ~]$

```
su - waldur -c "waldur status"
```

Background processing processes

Check that the following processes are running

- waldur-celery
- waldur-celerybeat

Example commands:

[localhost ~]$

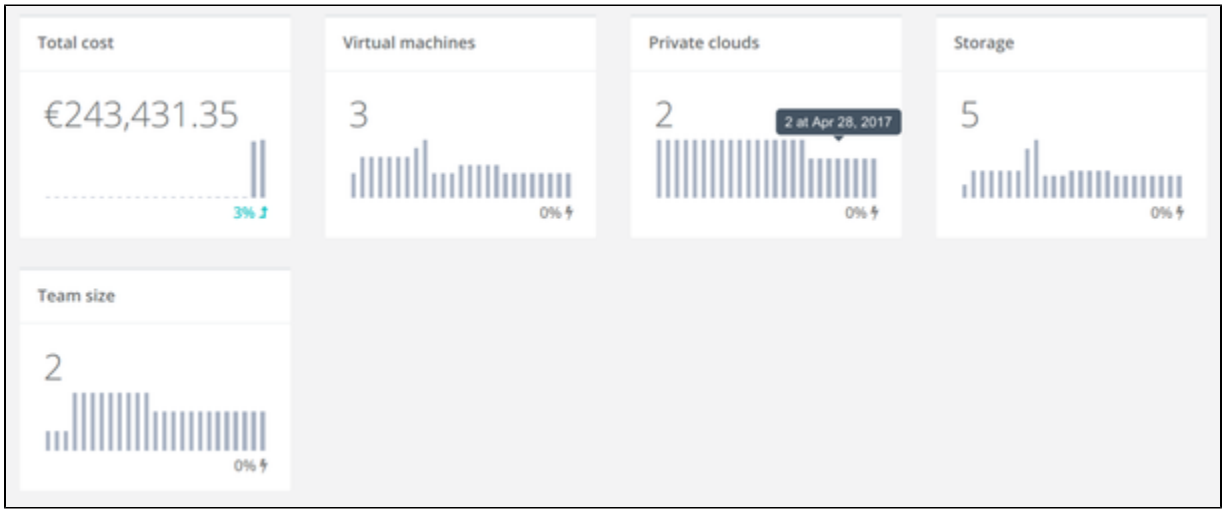
```
# Check if celery worker process is running
systemctl status waldur-celery > /dev/null && echo OK || echo Failed

# Check if celerybeat process is running
systemctl status waldur-celerybeat > /dev/null && echo OK || echo Failed

# Check HTTP response code
test "$(curl -o/dev/null -s -w'%{http_code}\n' -XPOST
'http://localhost/api-auth/password/')" = "400" && echo OK || echo
Failed
```

Monitoring

VirtEngine-Waldur allows you to monitor resource usage via HomePort and Grafana dashboard.

Monitoring in HomePort

Resource usage data is rendered via charts in organization and project dashboards.



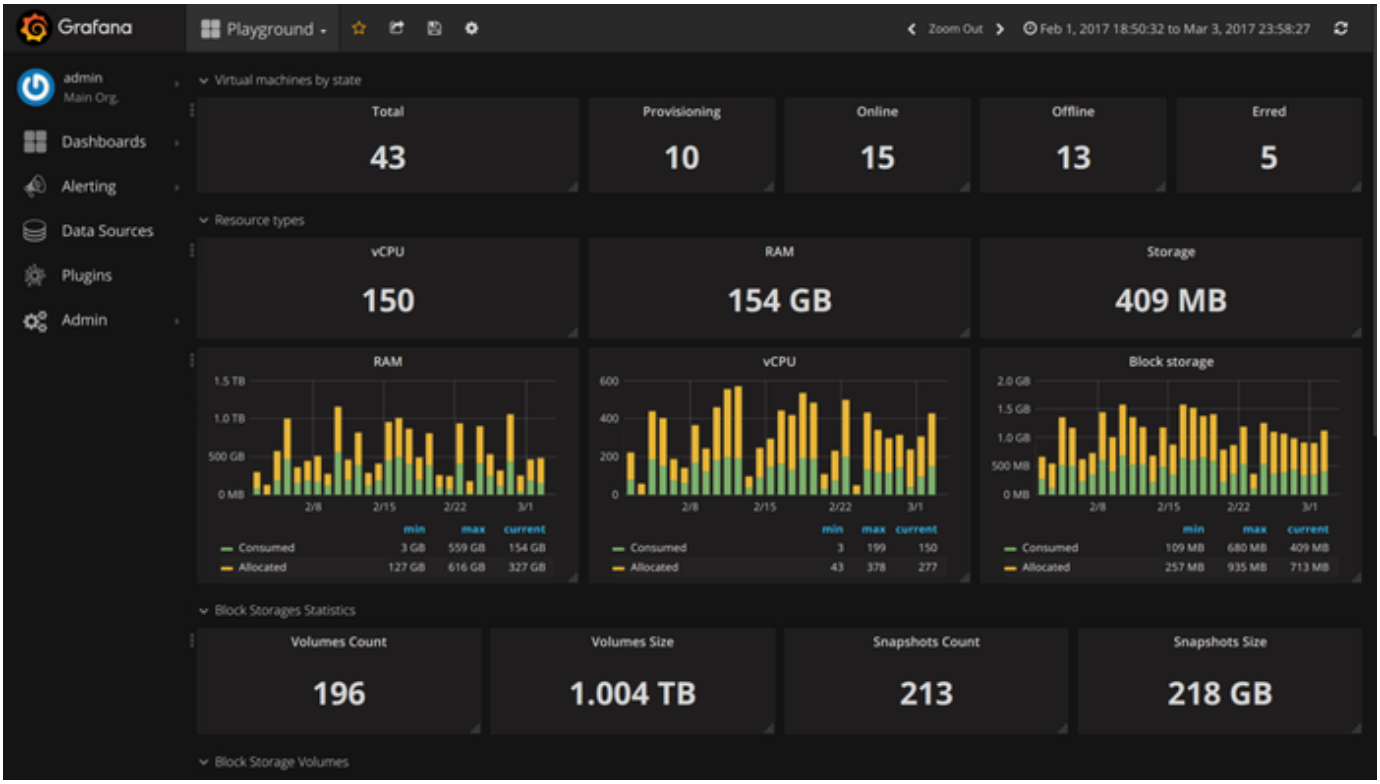
The following charts are available:

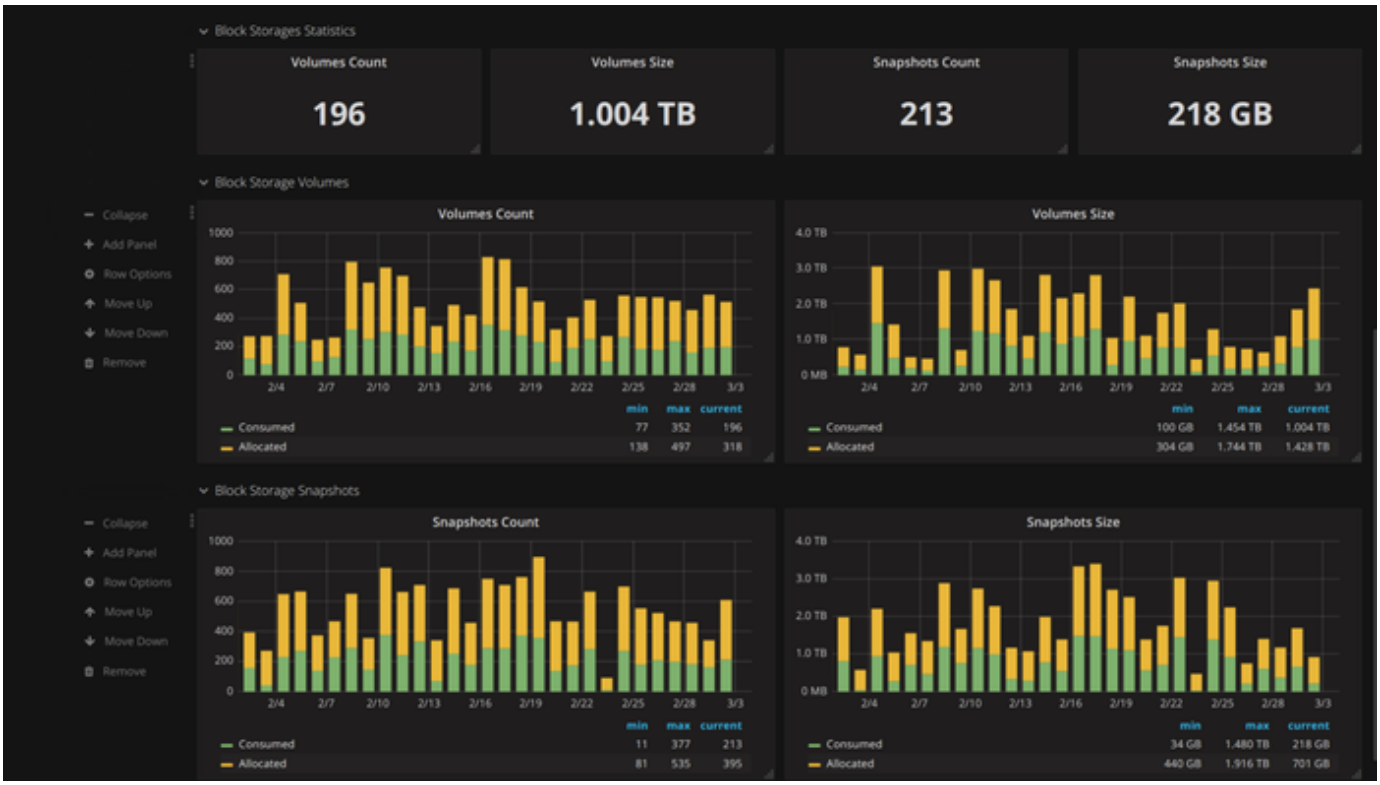
Title	Description
Virtual machines	Total number of virtual machines in the current project or organization.
Private clouds	Total number of virtual private clouds in the current project or organization.

Storage	Total number of block storage devices, such as OpenStack volumes and snapshots, in the current project or organization.
Team size	Total number of users in the current organization.
Total cost	Predicted cost for all resources in the current organization.

Monitoring via Grafana dashboards

Aggregated resources usage across all organizations is available via Grafana dashboards.





The following charts are available:

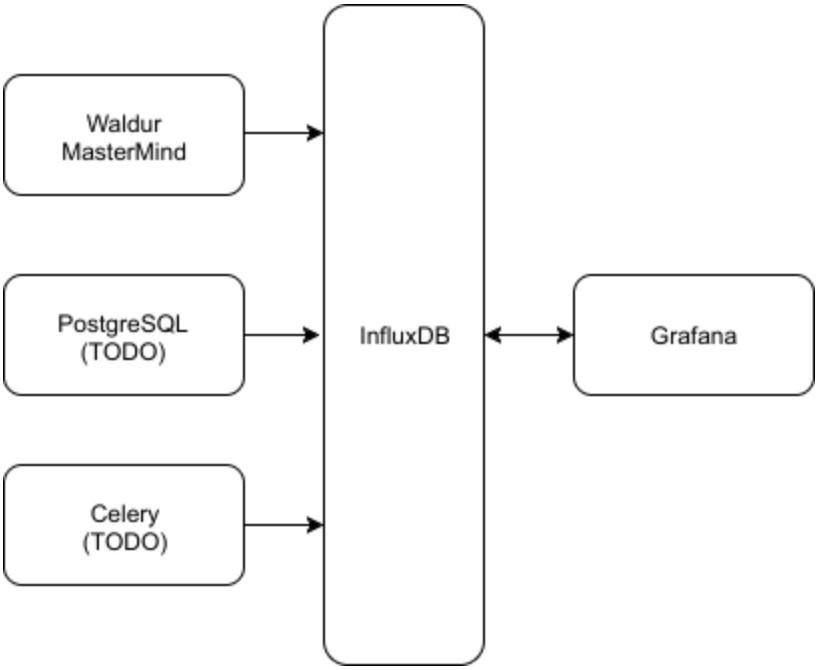
Name	Description
Virtual machines by state	Total number of OpenStack instances by state, provisioning, online, offline, erred.
Resource types	Total number of virtual CPU, RAM size and storage size.
RAM	Consumed and allocated RAM size.
vCPU	Consumed and allocated vCPU count.
Block storage	Consumed and allocated block storage size for OpenStack volumes and snapshots.
Volumes count	Consumed and allocated OpenStack volumes count.
Snapshots count	Consumed and allocated OpenStack snapshots count.
Volumes size	Consumed and allocated OpenStack volumes size.
Snapshots size	Consumed and allocated OpenStack snapshots size.

Note that historical charts for consumed and allocated computing resources are rendered per day for the past month.

Setup instructions

Under the hood, VirtEngine-Waldur periodically collects and pushes actual measurements to InfluxDB via Celery task.

Grafana executes queries to the InfluxDB database and renders charts.



Prerequisites

Setup Grafana and InfluxDB. Setup instructions are outside the scope of this guide and are widely available on the Internet.

Configure VirtEngine-Waldur analytics plugin

First, you should specify InfluxDB credentials in the **/etc/waldur/override.conf.py** file.

VirtEngine-Waldur analytics configuration

```
WALDUR_ANALYTICS = {
    'ENABLED': True,
    'INFLUXDB': {
        'host': 'localhost',
        'port': 8086,
        'username': 'USERNAME',
        'password': 'PASSWORD',
        'database': 'DATABASE',
        'ssl': False,
        'verify_ssl': False,
    }
}
```

Create Grafana datasource

Then, you should specify the same InfluxDB credentials in the Grafana datasource: http://docs.grafana.org/features/datasources/influxdb/

Import VirtEngine-Waldur dashboard to Grafana

Finally, you should upload VirtEngine-Waldur dashboard from JSON file to Grafana: http://docs.grafana.org/reference/export_import/

OPS-257 - Getting issue details... STATUS

Dashboard JSON file can be found in https://github.com/virtengine/waldur-grafana-dashboard

When Grafana asks you about datasource, please use datasource created in the previous step.

Hardware requirements

- VirtEngine-Waldur MasterMind
  - Celery workers
  - Elasticsearch
  - PostgreSQL
  - Redis
  - uWSGI
  - All-in-one
- VirtEngine-Waldur HomePort

VirtEngine-Waldur MasterMind

Celery workers

By default 3 Celery workers and one Celery beat process are started.

Minimum requirements: 1 CPU, 512 MB RAM

Recommended setup: 2 CPU or more, 1 GB RAM or more

> More memory should be added if more Celery worker processes are running on the same host (512 MB for each 4 Celery workers).

Elasticsearch

Minimum requirements: 1 CPU, 1 GB RAM

Recommended setup: 2 CPU or more, 2 GB RAM or more

PostgreSQL

Minimum requirements: 1 CPU, 512 MB RAM

Recommended setup: 2 CPU or more, 1 GB RAM or more

Redis

Minimum requirements: 1 CPU, 512 MB RAM

Recommended setup: 2 CPU or more, 1 GB RAM or more

uWSGI

By default 4 uWSGI processes are started.

Minimum requirements: 1 CPU, 512 MB RAM for every node

Recommended setup: 2 CPU or more, 1 GB RAM or more

All-in-one

It is possible to run Celery, Elasticsearch, PostgreSQL, Redis and uWSGI services on one machine. This setup is not recommended for production environments but can be useful for demonstration and evaluation purposes.

Recommended setup: 4 CPU, 10 GB RAM

VirtEngine-Waldur HomePort

Minimum web server capable to serve static files is enough. HomePort should work just fine with Nginx on 1 CPU and 512 MB RAM.

MasterMind configuration

Introduction

VirtEngine-Waldur is a Django based application, so configuration is done by modifying **settings.py**.

> **Warning**
> If you need to specify Unicode characters in invoice configuration, you should ensure that Python configuration file contains encoding declaration at the top of the file:
>
> ```
> # -*- coding: utf-8 -*-
> from __future__ import unicode_literals
> ```

If you want to configure options related to Django, such as tune caches, database connection, configure custom logging, etc, please refer to Django documentation.

Configuration for VirtEngine-Waldur Core is namespaced inside a single Django setting, named **WALDUR_CORE.**

Usually each plugin's settings are namespaced, so please refer to plugin documentation for details.

Basic configuration might look like this:

```
WALDUR_CORE = {
    'CLOSED_ALERTS_LIFETIME': timedelta(weeks=1),
    'ELASTICSEARCH': {
        'username': 'username',
        'password': 'password',
        'host': 'example.com',
        'port': '9999',
        'protocol': 'https',
    },
   'ENABLE_GEOIP': True,
   'EXTENSIONS_AUTOREGISTER': True,
   'GOOGLE_API': {
        'Android': {
            'server_key': 'AIzaSyA2_7UaVIxXfKeFvxTjQNZbrzkXG9OTCkg',
        },
        'iOS': {
             'server_key': 'AIzaSyA34zlG_y5uHOe2FmcJKwfk2vG-3RW05vk',
        }
    },
    'SHOW_ALL_USERS': False,
    'SUSPEND_UNPAID_CUSTOMERS': False,
    'OWNER_CAN_MANAGE_CUSTOMER': False,
    'CREATE_DEFAULT_PROJECT_ON_ORGANIZATION_CREATION': False,
    'TOKEN_KEY': 'x-auth-token',
    'TOKEN_LIFETIME': timedelta(hours=1),
    'SITE_NAME': 'VirtEngine-Waldur MasterMind',
    'COUNTRIES': ['EE', 'LV', 'LT'],
 'NOTIFICATIONS_PROFILE_CHANGES': {
  'ENABLED': True,
  'FIELDS': ('email', 'phone_number', 'job_title')
 },
}
```

VirtEngine-Waldur will send notifications from email address specified in **DEFAULT_FROM_EMAIL** variable. for example, DEFAULT_FROM_EMA
'noreply@example.com'

VirtEngine-Waldur Core settings

- **BACKEND_FIELDS_EDITABLE** allows to control /admin writable fields. If this flag is disabled it is impossible to edit any field that corresponds to backend value via /admin. Such restriction allows to save information from corruption. Flag is enabled by default.
- **CLOSED_ALERTS_LIFETIME** defines duration of closed alerts storage (timedelta value, for example timedelta(hours=1)). Expired closed alerts will be removed during the cleanup.
- **COUNTRIES** list is used in organization creation dialog in order to limit country choices to predefined set.
- **CREATE_DEFAULT_PROJECT_ON_ORGANIZATION_CREATION** enables generation of the first project on organization creation.
- **ELASTICSEARCH** - dictionary of Elasticsearch parameters.
    - **host** - Elasticsearch host (string).
    - **port -** Elasticsearch port (integer).
    - **protocol -** Elasticsearch server access protocol (string).
    - **username** - username for accessing Elasticsearch server (string).
    - **password -** password for accessing Elasticsearch server (string).
    - **verify_certs** - enables verification of Elasticsearch server TLS certificates (boolean).
    - **ca_certs** - path to the TLS certificate bundle (string).

- **ENABLE_GEOIP** indicates whether geolocation is enabled (boolean).
- **EXTENSIONS_AUTOREGISTER** defines whether extensions should be automatically registered (boolean).
- **GOOGLE_API** is settings dictionary for Google Cloud Messaging.
  - **Android** specifies settings for Android devices.
    - **server_key** is Google Cloud messaging server key.
  - **IOS** specifies settings for IOS devices.
    - **server_key** is Google Cloud messaging server key.
  - **NOTIFICATION_TITLE** is string to be displayed in the notification pop-up title.
- **INVITATION_LIFETIME** defines for how long invitation remains valid. Default value - 1 week.
- **LOGIN_COMPLETED_URL** - user will be redirected to that URL after successful login. The URL must include `{token}` , which will be replaced with user token, e.g., http://example.com/#/login_completed/{token}/{method}/
- **LOGIN_FAILED_URL** - user will be redirected to that URL after unsuccessful login, e.g., http://example.com/#/login_failed/
- **LOGOUT_COMPLETED_URL** - user will be redirected to that URL after successful logout, e.g., http://example.com/#/logout_completed/
- **LOGOUT_FAILED_URL** - user will be redirected to that URL after unsuccessful logout, e.g., http://example.com/#/logout_failed/
- **NATIVE_NAME_ENABLED** allows to render native name field in customer and user forms.
- **NOTIFICATIONS_PROFILE_CHANGES** allows enabling notifications about profile changes of organization owners and allows selecting, which fields of profile should be tracked.
- **OWNER_CAN_MANAGE_CUSTOMER** enables organization owners to create an organization (boolean).
- **OWNERS_CAN_MANAGE_OWNERS** enables organization owners to manage other organization owners, enabled by default.
- **ONLY_STAFF_MANAGES_SERVICES** allows to restrict provider management only to staff users, disabled by default.
  Please also note, that if this flag is enabled, OpenStack package provision & modification would be available for staff users only.
  It means that organization owners wouldn't be able to manage virtual private clouds.
  Also it is assumed that corresponding option is specified in VirtEngine-Waldur HomePort as well.
  It means that if ONLY_STAFF_MANAGES_SERVICES flag is enable in VirtEngine-Waldur MasterMind, onlyStaffManagesServices should enabled in VirtEngine-Waldur HomePort too.
- **SELLER_COUNTRY_CODE** specifies seller legal or effective country of registration or residence as an ISO 3166-1 alpha-2 country code. It is used for computing VAT charge rate.
- **SHOW_ALL_USERS** indicates whether user can see all other users in `api/users/` endpoint (boolean).
- **SITE_NAME** is used in email notifications in order to refer to the current deployment in user-friendly way.
- **SUSPEND_UNPAID_CUSTOMERS** - if it is set to True, then only customers with positive balance will be able to modify entities such as services and resources (boolean).
- **TOKEN_KEY** is header for token authentication. For example, 'x-auth-token'.
- **TOKEN_LIFETIME** defines for how long user token should remain valid if there was no action from user (timedelta value, for example timedelta(hours=1)).
- **ENABLE_ACCOUNTING_START_DATE** allows to enable accounting only for organizations using value of accounting_start_date field. Disabled by default.
- **COMPANY_TYPES** specifies list of available company types, it is used in organization creation dialog. By default Ministry, Private company, Public company, Government owned company are available.
- **INITIAL_CUSTOMER_AGREEMENT_NUMBER** allows to tweak initial value of agreement number (4000 by default). It is assumed that organization owner should accept terms of services when organization is registered via VirtEngine-Waldur HomePort.

Admin dashboard configuration

An admin dashboard supports custom links on Quick access panel. For instance, a panel below was configured with one additional link to https://virtengine.com:

Configuration of custom links is stored under **FLUENT_DASHBOARD_QUICK_ACCESS_LINKS** settings key and for current example has following structure

```
FLUENT_DASHBOARD_QUICK_ACCESS_LINKS = [
  {
   'title': '[Custom] VirtEngine-Waldur - Cloud Service',
virtengine.com
   'external': True, # adds an icon specifying that this link is
external,
   'description': 'Open-source Cloud Brokerage Platform',
   'attrs': {'target': '_blank'} # add an attribute to generated anchor
element which will open link in a new tab.
  },
]
```

Here is a short description of link parameters:

Name	Type	Required	Description
title	string	Yes	A title of the generated link
url	URL	Yes	A URL of the link
external	boolean	No	Specifies whether additional icon indicating an external URL has to be added
description	string	No	Tool tip on the link
attrs	dict	No	A dictionary of anchor attributes to be added to generated element.

It is also possible to omit optional fields and add links by specifying only a title and a URL to the generated link.

```
FLUENT_DASHBOARD_QUICK_ACCESS_LINKS = [
virtengine.com
  ['Find us on GitHub', 'https://github.com/virtengine/waldur-core'],
]
```

Custom templates configuration

To overwrite default templates you should put custom templates in **/etc/waldur/templates** directory.

For example, in order to customize invitation messages you should create the following files:

- /etc/waldur/templates/users/invitation_message.html
- /etc/waldur/templates/users/invitation_message.txt
- /etc/waldur/templates/users/invitation_subject.txt

Note that default invitation templates are located in **/usr/lib/python2.7/site-packages/waldur_core/users/templates/users/** directory.

Use local time zone

Set TIME_ZONE setting in **/etc/waldur/override.conf.py** to use local time zone. By default it is set to UTC. See the list of time zones for possible options.

AuthOpenID plugin configuration

Moved to identity providers.

AuthSocial plugin configuration

Move to identity providers.

AWS plugin configuration

AWS plugin does not support any custom configuration.

Azure plugin configuration

Azure plugin does not require any specific configuration when Azure itself does. The plugin depends on apache-libcloud azure module which provides access only to classic Azure services.

Every cloud service and a storage account to be created has to be marked as 'classic' therefore every created virtual machine is going to be 'classic' as well.

Azure configuration

When subscription is configured please make sure that **classic** cloud services and **classic** storage accounts are configured for usage as well. There is a naming convention between the service and the account, both names must match. Cloud service name is filled on provider creation phase and can be changed per service project link.

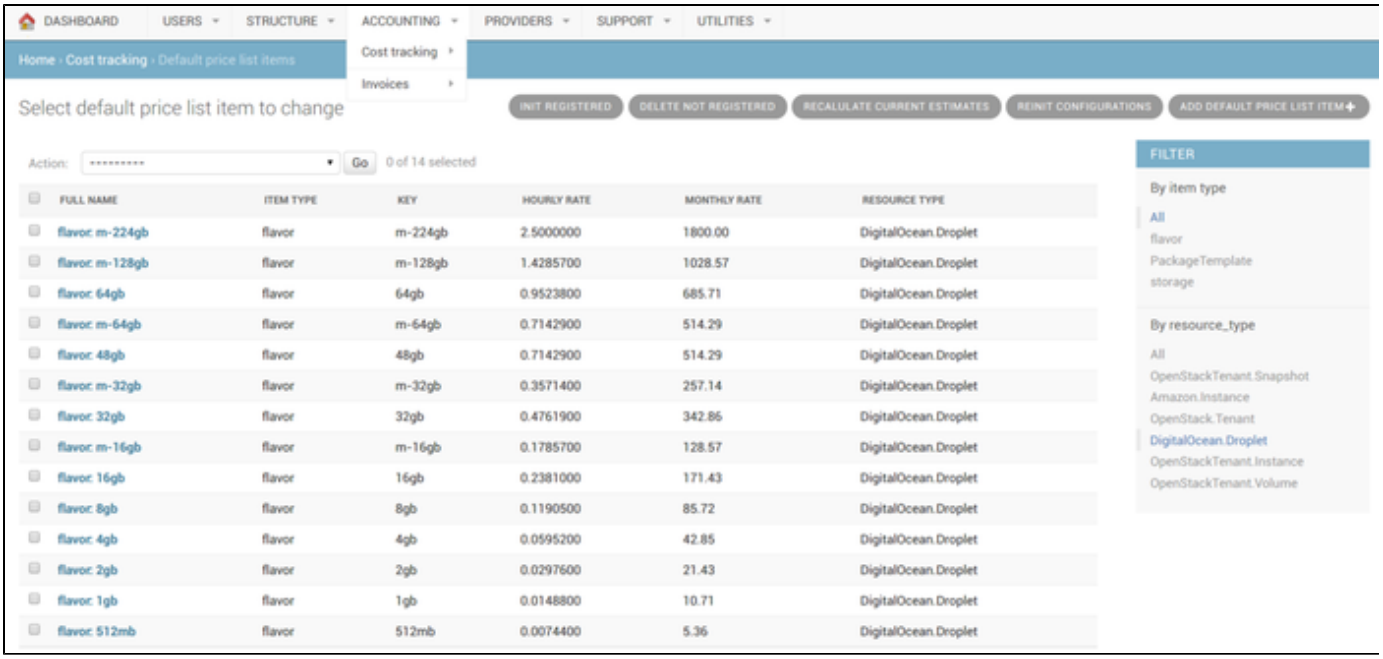
Cost tracking configuration

Overview

Cost tracking helps to predict resources consumption and total price for any project, resource or service.
Resource price calculation is based on price for each consumable item (Price list item) and resource consumption strategy that is predefined in code.

Configuration

To configure prices and recalculate price estimates, please navigate from menu bar to **Accounting Cost tracking Default price list items**.



On the first start

Use command **Init registered** to populate default price list items.

On consumables update

Service can change it consumables or resource consumption strategy. For example: new size was introduced in DigitalOcean or AWS.

In this case staff should execute commands **Delete not registered** and **Init registered** to make sure that default price list items are up-to-date. If resource consumption strategy was changed too - command ***Reinit configuration*** should be called.

To update price estimates

To avoid calculation overhead total estimate for each resource for current month is recalculated every hour. Command **Recalculate current estimates** could be used to recalculate estimates right now.

To restore or update default prices

To restore overridden default price or to update prices for particular service:

- Delete all default price list items that should be updated
- Call command **Init registered**

DigitalOcean plugin configuration

DigitalOcean plugin does not support any custom configuration.

FreeIPA plugin configuration

Update WALDUR_FREEIPA dictionary to define FreeIPA plugin configuration. Available options:

Option name	Type	Description	Default value
HOSTNAME	String	The hostname of FreeIPA server to connect to.	ipa.example.com
USERNAME	String	Name of the FreeIPA user with admin privileges.	admin
PASSWORD	String	Password of the FreeIPA user with admin privileges.	secret
VERIFY_SSL	Boolean	Enable SSL certificates verification for HTTPS requests.	True
USERNAME_PREFIX	String	Each username managed by FreeIPA plugin is prefixed.Prefix should not be empty. Please note that maximum username length is 32 characters so prefix should not be long.	waldur_
GROUPNAME_PREFIX	String	Each group name managed by FreeIPA plugin is prefixed. Prefix should not be empty. Group name is autogenerated, it consists of prefix and group UUID.	waldur_

BLACKLISTED_USERNAMES	List of strings	List of usernames not allowed for usage.	['root']

Invoice configuration

WALDUR_INVOICES

Option name	Type	Description	Default value
ISSUER_DETAILS	Dictionary	Issuer details for invoices	See below
PAYMENT_INTERVAL	Integer	How many days are given to pay for created invoice	30
INVOICE_REPORTING	Dictionary	Specifies options for exporting invoices as CSV attached to email	See below

WALDUR_INVOICES.ISSUER_DETAILS

Option name	Type	Description	Default value
company	String	Legal name of the company which issues invoices	DET-IO & OpenNode
address	String	Address of the company	Lille 4-205
country	String	Country name where company is registered	Estonia
country_name	String	Two-character IS0-3166-1 country code.	EE
email	String	Contact email	info@opennodecloud.com
postal	String	Postal code	80041
bank	String	Name of the bank where company has bank account	Estonian Bank
account	String	Bank account number	123456789
vat_code	String	Value added tax identification number	EE123456789
phone	Dictionary	Company's contact phone	{ 'country_code': '372', 'national_number': '5555555'}

WALDUR_INVOICES.INVOICE_REPORTING

Option name	Type	Description
ENABLE	Boolean	By default invoice reporting is disabled. Set this option to True in order to enable this feature.
EMAIL	String	Default "From" email.
CSV_PARAMS	Dictionary	Allows to customize CSV serialization. See also: Dialects and Formatting Parameters
SERIALIZER_EXTRA_KWARGS	Dictionary	Allows to customize OpenStackItemReportSerializer fields. Consult with source code for more details.
USE_SAF	Boolean	Export invoice report to SAF format.
SAF_PARAMS	Dictionary	Extra params for SAF exporter.

Default value:

```
'ENABLE': False,
'EMAIL': 'accounting@waldur.example.com',
'CSV_PARAMS': {
    'delimiter': str(';'),
},
'SERIALIZER_EXTRA_KWARGS': {
    'start': {
        'format': '%d.%m.%Y',
    },
    'end': {
        'format': '%d.%m.%Y',
    }
},
'USE_SAF': False,
'SAF_PARAMS': {
    'RMAKSULIPP': '20%',
    'ARTPROJEKT': 'PROJEKT',
}
```

JIRA Service Desk configuration

Overview

VirtEngine-Waldur's integration with JIRA Service Desk on the high level comes down to the following steps:

- Install or purchase on-demand JIRA Service Desk;
- Create a Service Desk project for the VirtEngine-Waldur-generated tickets;
- Create a dedicated user for VirtEngine-Waldur;
- Configure issue types to be used by VirtEngine-Waldur;
- Configure custom fields for the issues;
- Check if SLA field is displayed;
- Create a user to be used by VirtEngine-Waldur in order to communicate with Jira;
- Update VirtEngine-Waldur configuration with JIRA Service Desk project and user details;
- Setup JIRA Webhook to report updates from JIRA Service Desk to VirtEngine-Waldur;

JIRA Service Desk Configuration

Below is detailed information on how to configure a project in JIRA Service Desk.

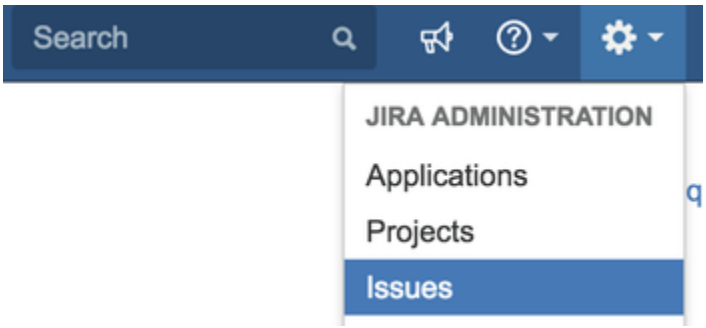
Project configuration

A **WALDUR** project has to be created in JIRA Service Desk as shown below.

	Project	Key	Project Type
	Waldur	WAL	Service Desk

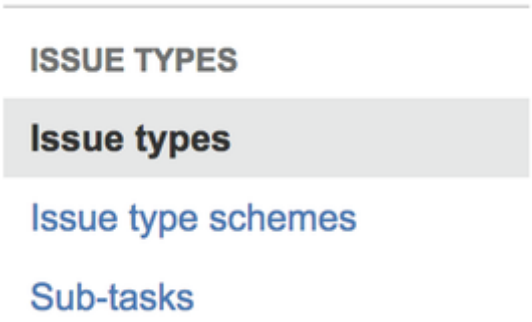
Issue fields configuration

Go to a cogwheel in the right upper corner and pick I**ssues** item in it.

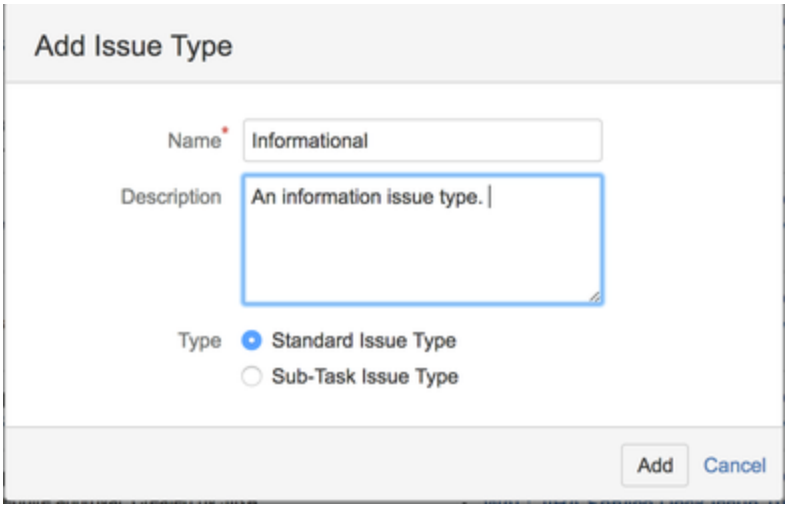


You will be presented with Issues configuration panes.

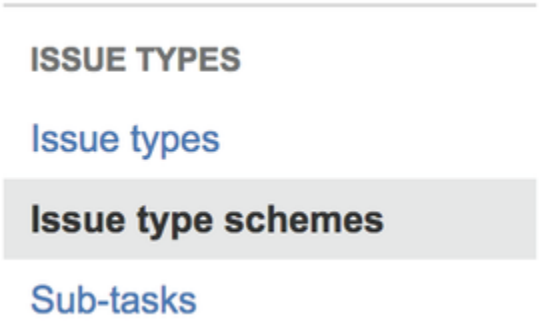
Let's configure Issue types first. Go to Issue types item under **Issue Types** section



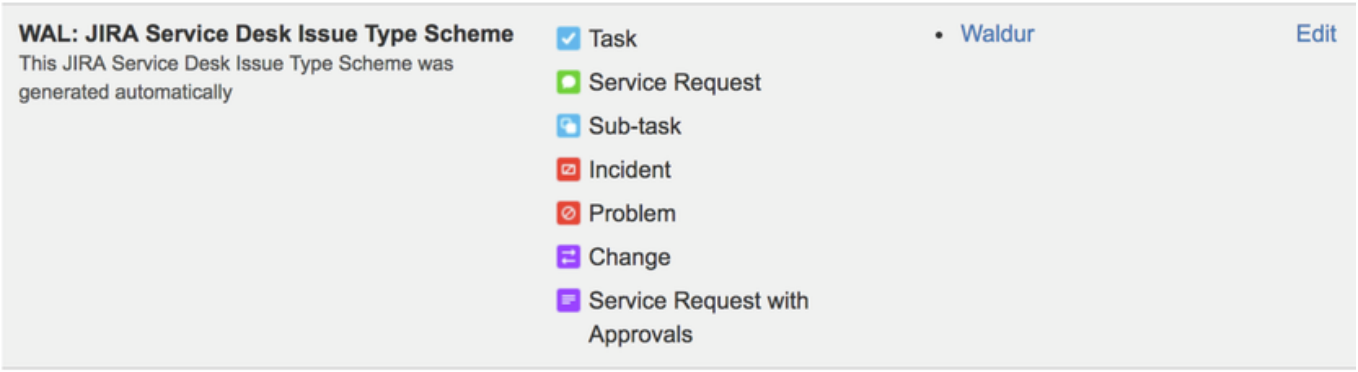
Make sure that all issue types are listed. If more types are required - add them by clicking 'Add issue type' button.



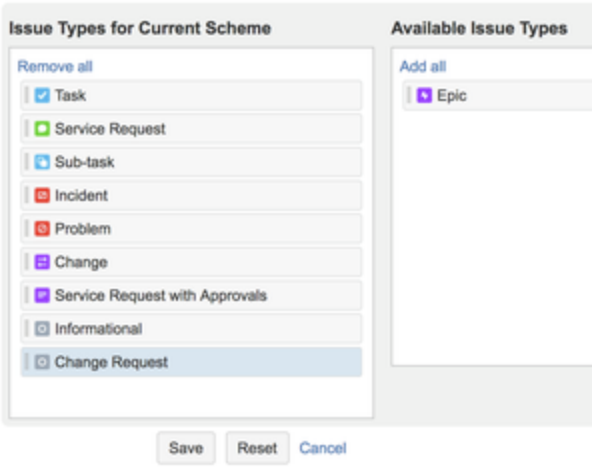
After all issue types have been added you have to configure a scheme for them. To do that open **Issue type schemes** under the same menu.



Find your project issue type scheme and add newly configured issue types to it by clicking '**Edit**'



Make sure that project scheme is updated after moving all issues to the scheme.

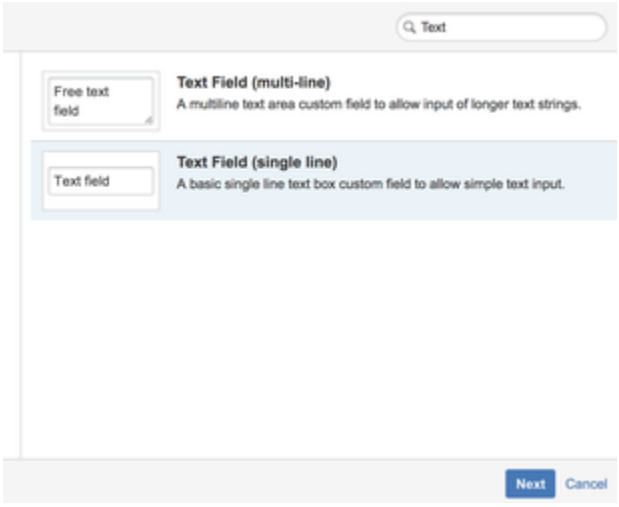


If you need custom fields it is possible to configure them by clicking on 'Custom Fields' section under **Fields** section on the left.

Make sure that 'impact_field', 'reporter_field' and 'sla_field' exist, otherwise create them. These 3 fields are required by WALDUR configuration.

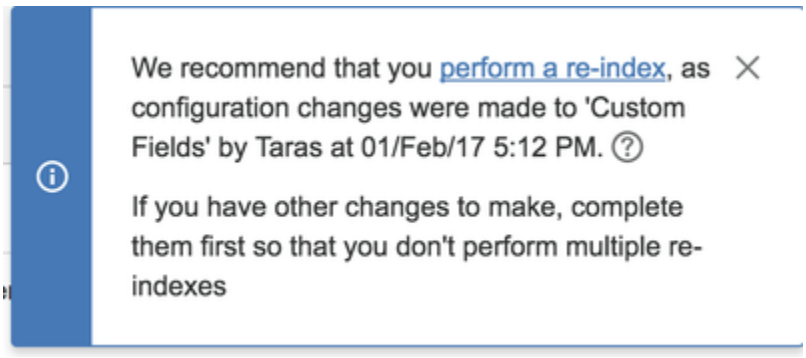
For instance, here is how 'Original Reporter' field creation looks like.

Click on 'Add custom field'. A popup with a list of field types will be presented. You can use search to filter them. Pick 'Text field' and fill it in.

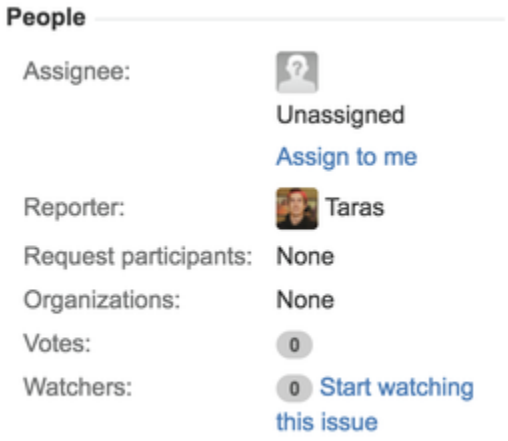


On the last step you will be asked to chose views where current field has to be displayed. Please make sure that issue field is displayed at least in your project.

Sometimes JIRA asks to run re-indexing operation when new issue field is added. It is strongly recommended to run it after all custom fields have been added.



**Note: Please note that configuration contains a 'caller_field' that does not have to be configured as in case of JIRA Service Desk as it is going to be used in order to assign 'caller' to 'Request participants'. Same configuration must not be used with other types of JIRA Services as it may lead to unpredicted behavior.*



Check if SLA field is displayed

Before proceeding please make sure that 'Time to first response' field is displayed on a screen of you project. For instance:

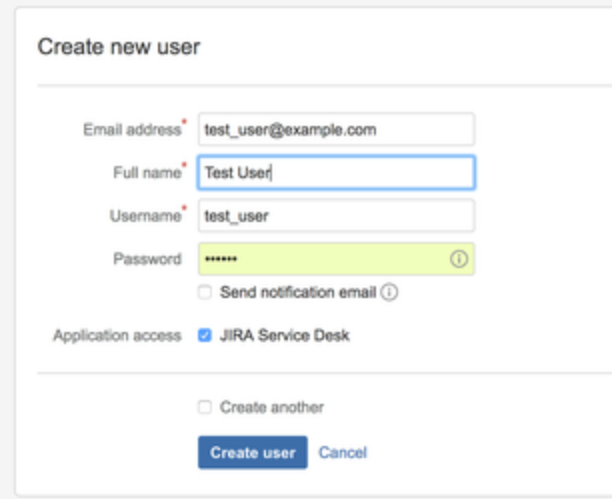
Time to approve normal change LOCKED This custom field was created by JIRA Service Desk.		Screens
Time to close after resolution LOCKED This custom field was created by JIRA Service Desk.		Screens
Time to first response LOCKED This custom field was created by JIRA Service Desk.	• Default Screen • WAL: JIRA Service Desk Screen • WAL: JIRA Service Desk: Change Create Issue Screen • WAL: JIRA Service Desk: Change View/Edit Screen • WAL: JIRA Service Desk: Incident Create Issue Screen • WAL: JIRA Service Desk: Incident View/Edit Screen • WAL: JIRA Service Desk: Problem Create Issue Screen • WAL: JIRA Service Desk: Problem View/Edit Screen • WAL: JIRA Service Desk: Request	Screens

VirtEngine-Waldur user creation

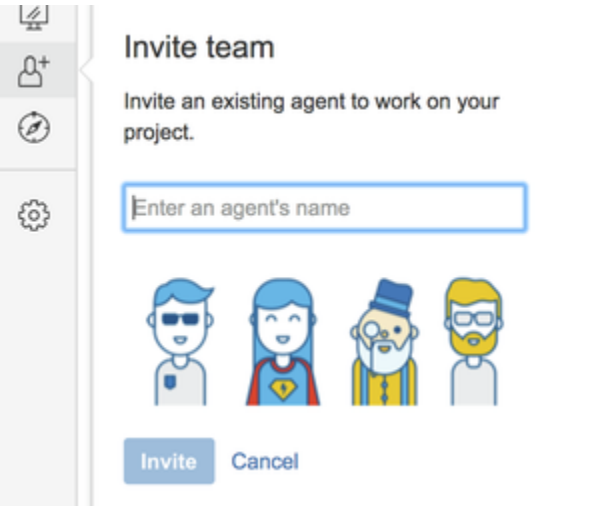
In order to communicate with JIRA a special user has to be created for VirtEngine-Waldur. Please follow next steps to create a user.

Click on a cogwheel and select User Management.

Click on 'Create User' and fill in user form.



After user is created please add it as an agent to the project by opening a project and clicking 'Invite team'



WALDUR configuration

Below is a part of configuration related to JIRA:

WALDUR_SUPPORT configuration

```
WALDUR_SUPPORT = {
 'ACTIVE_BACKEND':
'waldur_mastemind.support.backend.atlassian:ServiceDeskBackend',
 'CREDENTIALS': {
     'server': 'https://waldur.example.com/',
     'username': 'USERNAME',
     'password': 'PASSWORD',
     'verify_ssl': True,
 },
 'PROJECT': {
     'key': 'WAL',
 },
 'ISSUE': {
     'types': ['Informational', 'Service Request', 'Change Request',
'Incident'],
     'impact_field': 'Impact',
      'reporter_field': 'Original Reporter',
      'caller_field': 'Request participants',
     'sla_field': 'Time to first response',
  'summary': '{{issue.summary}}',
  'description': (
      'Description: {{issue.description}}\n'
      '{% if issue.project %}'
      'Project Name: {{issue.project.name}}\n'
      '{% endif %}'
      '{% if issue.customer %}'
      'Organization Name: {{issue.customer.name}}\n'
      '{% endif %}'
      '{% if issue.resource %}'
      'Service Type:
{{issue.resource.service_project_link.service.type}}\n'
      'Affected Resource Name: {{issue.resource}}\n'
      '{% endif %}'
 },
...
}
```

Sections description

ACTIVE_BACKEND

ACTIVE_BACKEND configures which backend waldur has to use in order to communicate with Jira. At the moment it supports two backends and two JIRA services accordingly: JiraBackend and ServiceDeskBackend. In order to configure communication with JIRA Service Desk - **ServiceDeskBackend** must be used.

CREDENTIALS

CREDENTIALS section stores information about a server and credentials to be used by **WALDUR** to communicate with a service desk. For

instance, in current configuration we assume that configured server name 'http://example.com/' is an instance of JIRA Service Desk and it is accessible with username '**USERNAME**' and password '**PASSWORD**' without need to verify ssl.

PROJECT

At the moment it is possible to integrate with only one project which has to be configured under **PROJECT** section. Usually project key has only 3 letters so if the project we want to work with is named 'WALDUR' in JIRA, most likely its key is going to be '**WAL**' which is shown in the configuration above.

ISSUE

ISSUE section defines JIRA issue configuration to be used for integration with internal VirtEngine-Waldur support issue. For instance, using curren configuration an issue type field in JIRA can be populated with 1 of next 4 values '**Information**', '**Service Request**', '**Chage Request**' and '**Incide nt**'. An impact, a reporter and a caller fields must be configured in JIRA Service Desk as custom fields. An 'sla_field' is a name of a field to get VirtEngine-Waldur support issue SLA and usually is called '**Time to first response**'

**Note: Issue fields are found by a field name, not its key.*

Issue template fields

Summary and **Description** fields in the **ISSUE** configuration are considered to be template fields. A template field is rendered by a Django template engine with an issue instance as a parameter.

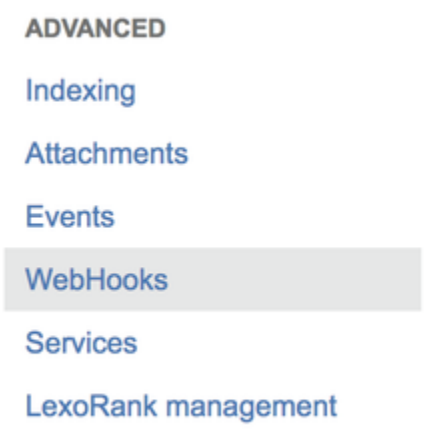
It is also possible to skip field rendering by adding 'if' conditions as in usual Django template.

Updating VirtEngine-Waldur configuration

- List all configured issue types under 'Issuetypes' section. In this example we have only 4 options.
- Update issue next fields 'reporter_field', 'sla_field', 'impact_field' by putting their names ***not keys*** under 'Issue' section
- Put a project **key** under 'Issuekey' section.
- Update **CREDENTIALS** section by filling VirtEngine-Waldur user credentials created in JIRA configuration section

JIRA Webhook configuration

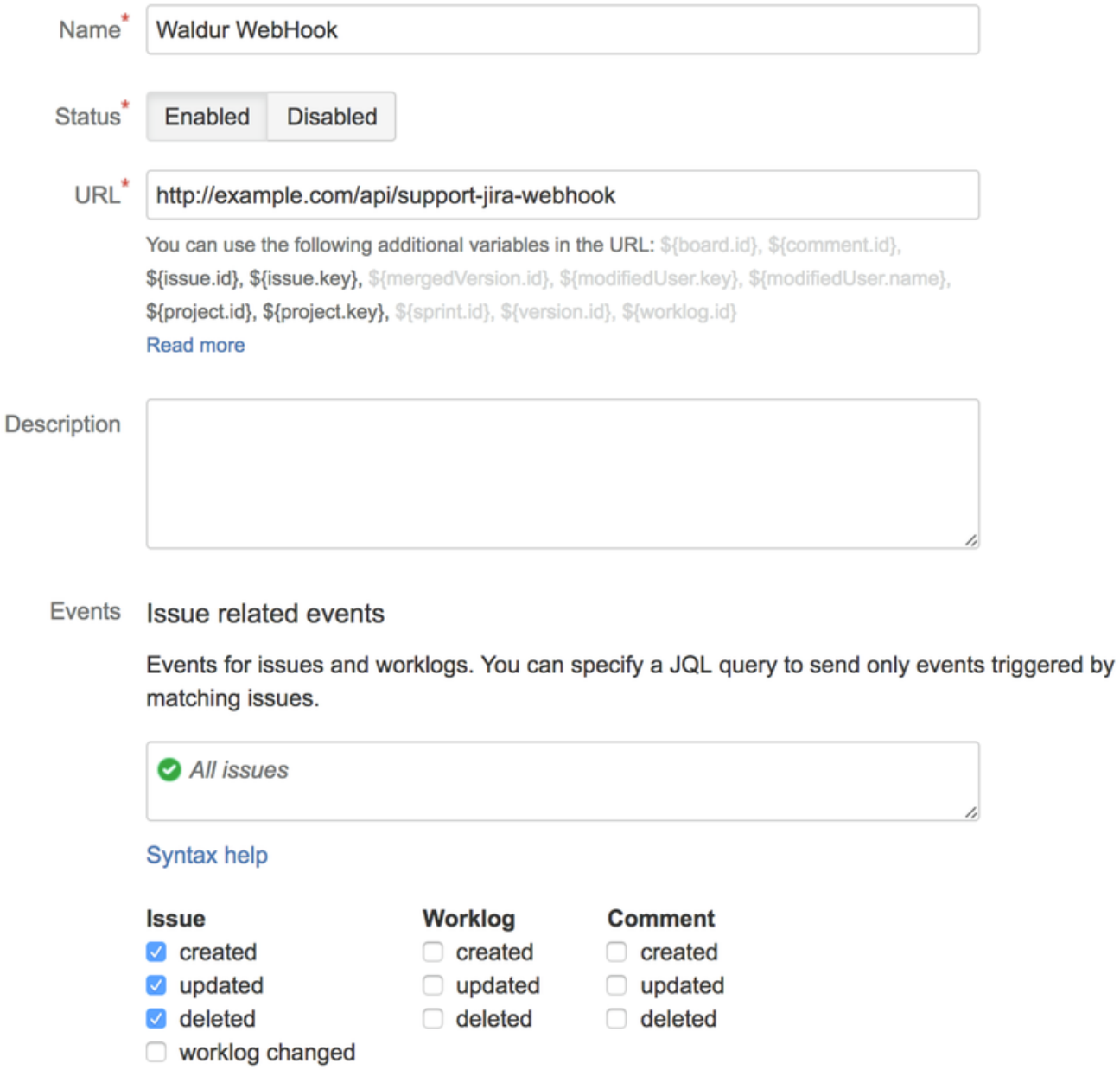
Go to 'System' item in cogwheel on the right upper corner and select WebHooks under Advanced section.



Click on 'Create a WebHook' and add next information:

- Web Hook name
- Enable hook
- VirtEngine-Waldur webhook listening point which ends with 'api/support-jira-webhook/' so the full URL will be: 'http://example.com/api/supp bhook/'
- Add description if required
- Select next Issue related events: created, updated, deleted

Issue comments will be synced automatically, no need to select them. Here is an example:



Email notifications

Email notifications are going to be sent automatically to the issue caller when:

- An issue is updated;
- A new comment is added;

An email includes an information about an event that triggered the notification and a link to the updated issue.

Disable email notifications

Email notifications are enabled by default and can be disabled by setting **SUPPRESS_NOTIFICATION_EMAILS** setting to **True**.

Issue templates

It is possible to provide a list of templates for each of the issue types. Templates will be shown to the end-user as a select box, which pre-fills information in the ticket. Templates can optionally include attachments, which the user would be able to download and connect with the issue. Templates can be managed from MasterMind admin interface: **/admin/support/template/.**

Disable TSL verification (not recommended)

It is possible to disable SSL verification for JIRA by setting **WALDUR_SUPPORT['CREDENTIALS']['verify_ssl']** key to **False**. Please use it at your own risk.

Notification about owners profile changes

VirtEngine-Waldur supports creating more fine-grained notification about owners profile changes that could be sent to a webhook or a concrete em operator of VirtEngine-Waldur). Please note that the notification will be connected to a specific user in VirtEngine-Waldur.

Step 1. Enable notifications and set users fields for change monitoring.

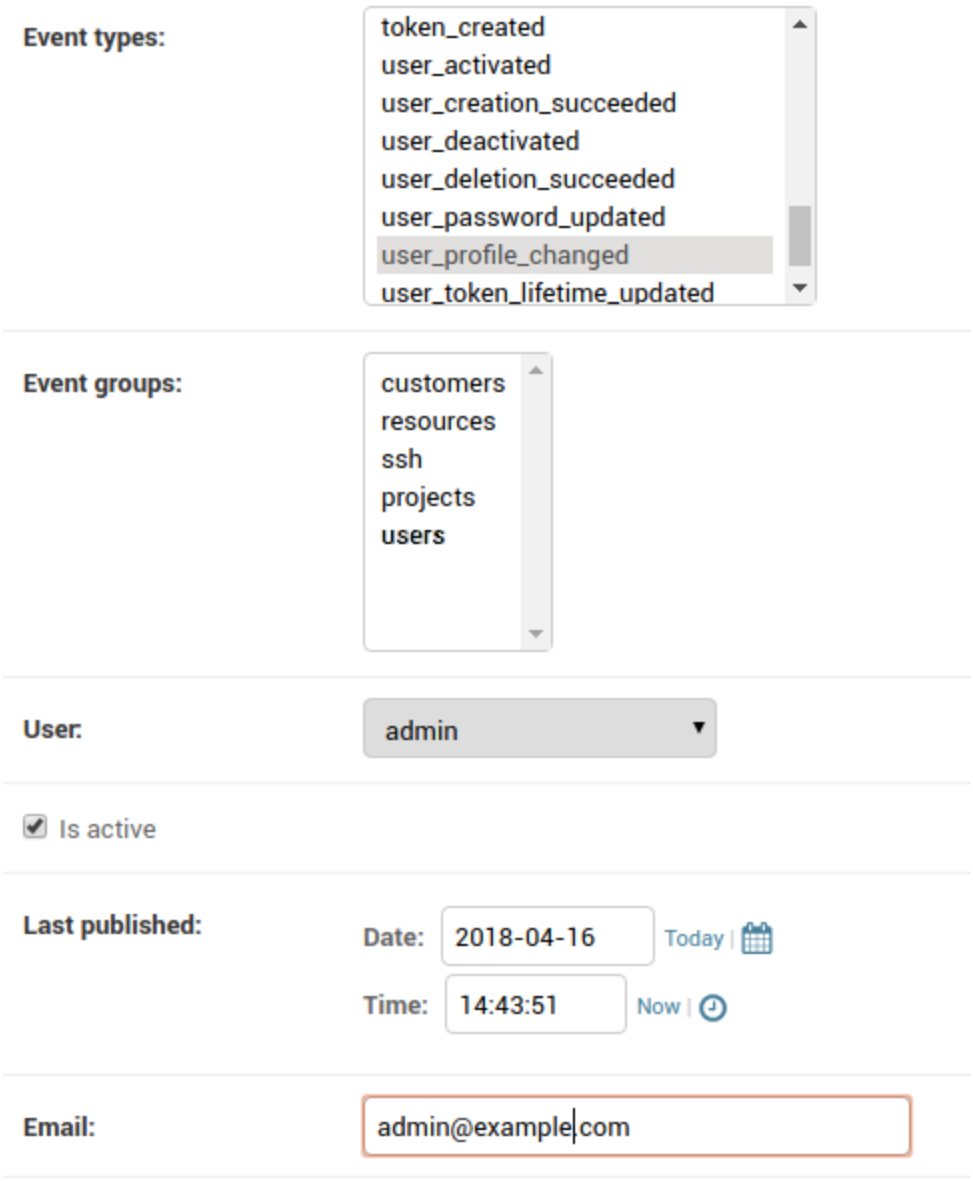
Set NOTIFICATIONS_PROFILE_CHANGES['ENABLED'] as True and set NOTIFICATIONS_PROFILE_CHANGES['FIELDS'] as a list of monitored fields in a settings file.

```
'NOTIFICATIONS_PROFILE_CHANGES': {'ENABLED': True, 'FIELDS': ('email',
'phone_number', 'job_title')}
```

Step 2. Create hook in MasterMind admin.

Login to MasterMind's administration interface and navigate to https://mastermind.example.com/logging/emailhook/add/

Create hook and select 'Event types' as user_profile_changed.



Notification processing settings

There are three types of notifications in VirtEngine-Waldur: webhooks, email hooks and push notifications.

Webhooks

If notification is configured as webhook, VirtEngine-Waldur sends HTTP request at specified endpoint when events specified in notification are trig

VERIFY_WEBHOOK_REQUESTS configuration option allows to send verified request on webhook processing, it is used by default.

Email hooks

If notification is configured as email hook, email with event details is sent to the specified address.

In order to customize email notification message you should put custom template in **/etc/waldur/templates/logging/email.html**

Note that default email notification template is located at **/usr/lib/python2.7/site-packages/waldur_core/logging/templates/logging/email.html**

Push notification

If notification is configured as push notification, it is processed via Google Cloud Messaging API.

So you setup mobile application with GCM and specify server_key in MasterMind **GOOGLE_API** configuration option. For example:

Google Cloud Messaging Configuration Object

```
{
  "NOTIFICATION_TITLE": "VirtEngine-Waldur notification",
  "Android": {
    "server_key": "AIzaSyA2_7UaVIxXfKeFvxTjQNZbrzkXG9OTCkg",
  },
  "iOS": {
    "server_key": "AIzaSyA34zlG_y5uHOe2FmcJKwfk2vG-3RW05vk",
  }
}
```

OpenStack plugin configuration

- OpenStack application configuration
- OpenStackTenant application configuration

OpenStack plugin is divided into 2 applications:

- OpenStack application - allows to operate with tenants, their networks and security groups.
- OpenStackTenant application - allows to operate with resources that are related to a particular tenant - instances, volumes.

OpenStack application configuration

Update **WALDUR_OPENSTACK** dictionary to define OpenStack application configurations.

Available options:

- **DEFAULT_SECURITY_GROUPS** - list of security groups that are added to tenant on creation. Default value:

  Default security groups

- **SUBNET** - restrictions for subnet selection range in form of a dictionary. Supported keys:
  - **CIDR_REGEX** - regular expression for subnet cidr.
  - **CIDR_REGEX_EXPLANATION** - user-friendly regex explanation.
  - **ALLOCATION_POOL_START**,**ALLOCATION_POOL_END** - restrictions for allocation pool range.

  Default value:

SUBNET

```
'SUBNET': {
                # Allow cidr: 192.168.[1-255].0/24
                'CIDR_REGEX':
r'192\.168\.(?:25[0-4]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]?)\.0/24',
                'CIDR_REGEX_EXPLANATION': 'Value should be
192.168.[1-254].0/24',
                'ALLOCATION_POOL_START':
'{first_octet}.{second_octet}.{third_octet}.10',
                'ALLOCATION_POOL_END':
'{first_octet}.{second_octet}.{third_octet}.200',
            },
```

- **DEFAULT_BLACKLISTED_USERNAMES** - list of usernames that cannot be used as tenant user username. Default value - *['admin', 'service']*.
- **MANAGER_CAN_MANAGE_TENANTS** - defines manager permissions. If this flag is true - manager can execute actions that will change cost of the project: delete tenants, change their configuration. False by default.
- **TENANT_CREDENTIALS_VISIBLE** - **True** if tenant user_username, user_password and access_url should be returned by tenants API and it is possible to set user_username on tenant creation. If **False** tenant credentials are going to be autogenerated and cannot be set by a user.

> Please note that there is a similar configuration in HomePort.
>
> You can check OpenStackTenant credentials management for more information.

OpenStackTenant application configuration

Update **WALDUR_OPENSTACK_TENANT** dictionary to define OpenStackTenant application configurations.

Available options:

- **MAX_CONCURRENT_PROVISION** - defines how many resources can be provisioned by one OpenStack simultaneously. Default value:

MAX_CONCURRENT_PROVISION

```
'MAX_CONCURRENT_PROVISION': {
    'OpenStackTenant.Instance': 4,
    'OpenStackTenant.Volume': 4,
    'OpenStackTenant.Snapshot': 4,
},
```

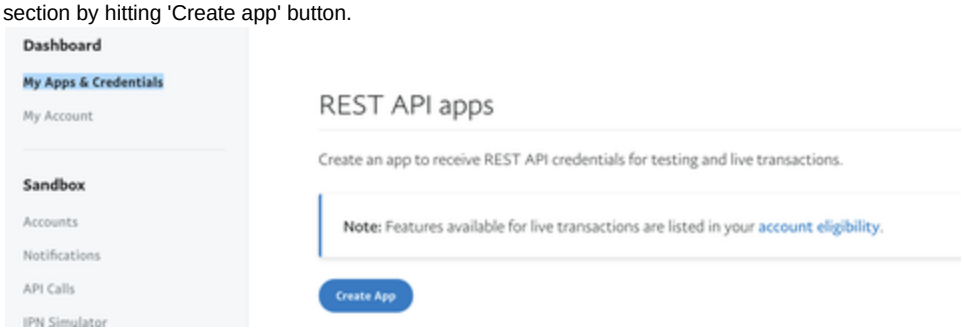
PayPal plugin configuration

Intro

PayPal plugin allows emitting invoices and accepting payments for VirtEngine-Waldur-managed services using PayPal. To enable support, configu needed both on PayPal and VirtEngine-Waldur side. Below we go over the steps required.

How to register a PayPal client

To register a PayPal application login to https://developer.paypal.com, go to My Apps & Credentials and create an app under REST API apps section by hitting 'Create app' button.

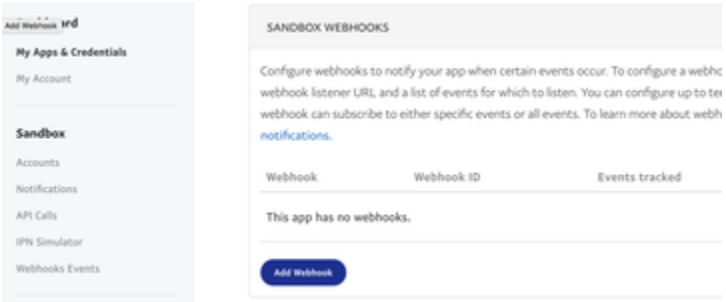


After an application is register click on it and copy credentials from API CREDENTIALS section. Please note a Sandbox/Live switch on top which means sandbox and live clients have different credentials.

How to configure a webhook

PayPal supports webhooks for different type of events. See full list here. At the moment PayPal plugin listens only to invoice related hooks.

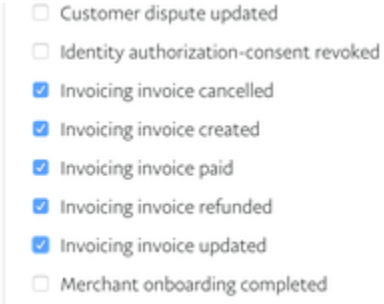
To add a webhook go to the registered app and hit 'Add Webhook' button under Webhooks section. Please notice a mode in which you configured a webhook, it is important to configure a webhook according to the mode used in BACKEND configuration.



Change your server name and put a webhook URL - https://**waldur.example.com**/api/paypal-invoices-webhook/

Please notice that a webhook URL must be in HTTPS.

Current endpoint listens only to invoice related events, so you can select only invoice related events like shown below.



If any non-related event is selected it is going to be ignored by endpoint. It is easier to track webhook events in PayPal if you keep them to minimum.

WALDUR_PAYPAL configuration

Update WALDUR_PAYPAL dictionary to configure PayPal plugin.

Key	Type	Description	Default value
ENABLED	Boolean	Enables/disables plugin	False
BACKEND	Dictionary	PayPal client configuration	See BACKEND configuration section
STALE_PAYMENTS_LIFETIME	Timedelta	Amount of time to keep payments	1 week

BACKEND configuration

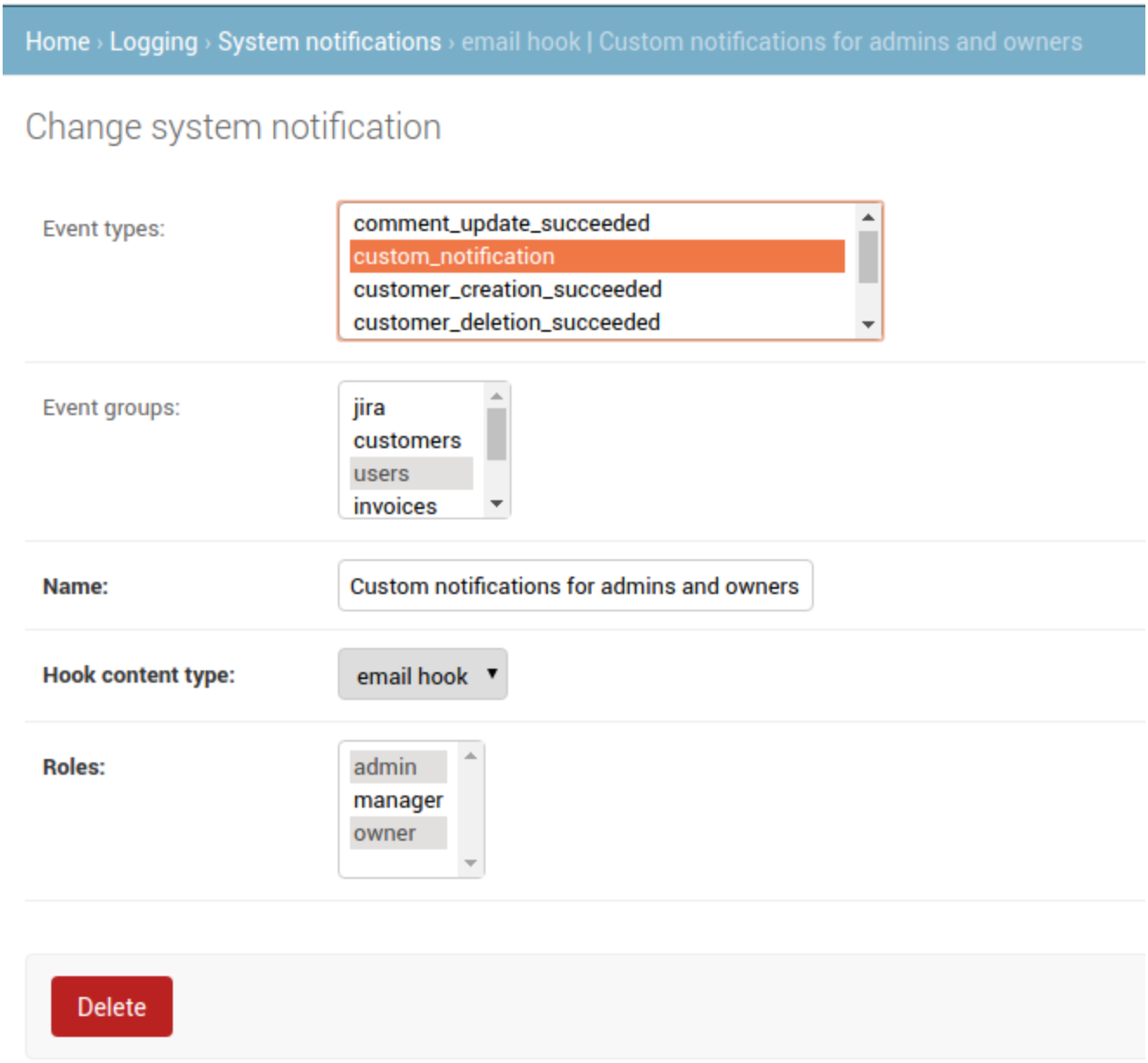
Key	Type	Description	Default value	Supported values
mode	String	Defines server which plugin has to use.	sandbox	live, sandbox
client_id	String	PayPal application client id		
client_secret	String	PayPal application client secret		
currency_name	String	Currency to work with	USD	The three-letter ISO-4217 currency code

Setting up system notifications

DRAFT

System notifications are the one that should be sent to e-mails of connected users without the need for users to subscribe to them.

To configure system notifications, you must specify which events types should be considered that and user roles in a project or a company. To create a system notification please go to MasterMind admin interface and navigate to **/admin/logging/systemnotification/**. Create a new notification with desired events types or groups, user roles and set '**Hook content type**' to '**email hook**'.



In the example above, if an event with type 'custom_notification' is created in the system, users will receive a notification about it, regardless of whether they are subscribed to notifications of this type or not. The user will only receive a notification if he has read access for that object and his role in the project is 'admin' or 'owner'.

SLURM plugin configuration

Configure SLURM cluster

SLURM accounting plugin assumes that at least one cluster is configured. For example:

```
sacctmgr add cluster linux
```

Enforce SLURM accounting limits

In order to enforce limits set on associations and QoS, please modify slurm.conf:

```
AccountingStorageEnforce=limits
```

Please note, that when *AccountingStorageEnforce* is changed, a restart of the slurmctld daemon is required (not just a scontrol reconfig):

```
systemctl restart slurmctld
```

Enable SLURM Multi Priority plugin

In order to enable ordering for the queue of jobs waiting to be scheduled, please modify slurm.conf:

```
PriorityType=priority/multifactor
```

When slurm.conf is changed, you should reload configuration:

```
scontrol reconfig
```

Configure VirtEngine-Waldur SLURM plugin

Update WALDUR_SLURM dictionary to define SLURM plugin configuration. Available options:

Option name	Type	Description	Default value
CUSTOMER_PREFIX	String	Prefix for SLURM account name corresponding to VirtEngine-Waldur organization.	waldur_customer_
PROJECT_PREFIX	String	Prefix for SLURM account name corresponding to VirtEngine-Waldur project.	waldur_project_
ALLOCATION_PREFIX	String	Prefix for SLURM account name corresponding to VirtEngine-Waldur allocation.	waldur_allocation_
PRIVATE_KEY_PATH	String	Path to private key file used as SSH identity file for SLURM master.	/etc/waldur/id_rsa

References

- SLURM Accounting and Resource Limits
- SLURM Resource Limits
- SLURM Multifactor Priority Plugin

Support configuration

Support plugin should be configured simultaneously with support backend.
Documentation for JIRA Service Desk configuration: JIRA Service Desk configuration

Update WALDUR_SUPPORT dictionary to configure backend independent options:

- **ENABLED** is True by default. It if is set to False, all endpoints of VirtEngine-Waldur Support extension return HTTP error with status code
- **OFFERINGS** and **DEFAULT_OFFERING_ISSUE_TYPE** - please check ServiceStore configuration#OFFERINGconfigurationstructure

HomePort configuration

- General configuration options
- Google Analytics
- White-labeling
    - Custom page title

Configuration options are stored in JSON files. After application assets have been loaded, but before application is bootstrapped, configuration files are loaded and their options are merged together with default settings. Custom configuration settings are defined in file:

```
/etc/waldur-homeport/config.json
```

General configuration options

- **apiEndpoint** should contain VirtEngine-Waldur MasterMind REST API server address.
- **enableExperimental** is a boolean value, false by default. Set it to true to enable experimental features.
- **SENTRY_DSN** should contain valid Data Source Name for connecting to Sentry for issue reporting (optional).

Google Analytics

To enable Google Analytics, set its code in the configuration:

- **GoogleAnalyticsID** - Google Analytics ID, used for tracking, if specified, disabled by default.

White-labeling

Custom page title

- **shortPageTitle** is used as prefix for page title.
- **modePageTitle** is used as default page title if it's not specified explicitly.

Custom logo

You may define two logos: colorful and monotone, they are used in different places.

- **loginLogo** should contain path to image; it is rendered on login / signup page, if specified, otherwise shortPageTitle is used (sample dimensions: 300x81px).
- **sidebarLogo** should contain path to image; it is rendered in sidebar header, if specified, otherwise shortPageTitle is used (sample dimensions: 175x48px).

poweredByLogo should contain path to image; it is rendered at the login page, if specified

Custom favicon

Default favicon is located in file **/usr/share/waldur-homeport/images/favicon.ico**

In order to customize it, you should configure alias in NGINX. For example:

```
location /images/favicon.ico {
  alias /var/www/waldur/favicon.ico;
}
```

Custom manifest

Manifest file is used to install web application to the home screen of a device. The default manifest file is located in **/usr/share/waldur-homeport/manifest.json**.

You can read more about manifest.json file here. In order to customize it, you should configure alias in NGINX. For example:

```
location /manifest.json {
  alias /var/www/waldur/manifest.json;
}
```

There are three logos in example manifest file. In order to specify custom logos, you should configure alias in NGINX. For example:

```
location /images/waldur/logo-96x96.png {
  alias /var/www/waldur/logo-96x96.png;
}
location /images/waldur/logo-144x144.png {
  alias /var/www/waldur/logo-144x144.png;
}
location /images/waldur/logo-192x192.png {
  alias /var/www/waldur/logo-192x192.png;
}
```

Custom terms of service

Default placeholder for terms of service is located in file **/usr/share/waldur-homeport/views/tos/index.html**

In order to customize it, you should configure alias in NGINX. For example:

```
location /views/tos/index.html {
  alias /var/www/waldur/tos.html;
}
```

Custom privacy statement

Default placeholder for privacy statement is located in file **/usr/share/waldur-homeport/views/policy/privacy.html**

In order to customize it, you should configure alias in NGINX. For example:

```
location /views/policy/privacy.html {
  alias /var/www/waldur/privacy.html;
}
```

Custom primary colors

VirtEngine-Waldur HomePort design is based on Inspinia theme, which in turn is based on Bootstrap. Primary colors of theme are specified in this

https://github.com/virtengine/waldur-homeport/blob/develop/assets/sass/inspinia/_variables.scss

In order to consistently modify primary colors, you should fetch source code of VirtEngine-Waldur HomePort and build RPM using RPM spec file:

https://github.com/virtengine/waldur-homeport/blob/develop/packaging/waldur-homeport.spec

Additional settings

Option name	Type	Description	Default value
disabledFeatures	List of strings	List of features disabled in current deployment by administrator. For example, in order to conceal support workspace and request-based items it should contain "offering" and "support". See also list of available features below.	Empty list
disabledServices	List of strings	List of disabled services, for example: ['Amazon', 'Azure']	Empty list
accountingMode	String	Set it to "billing" in order to display additional billing details for invoices.	accounting
invitationsEnabled	Boolean	Set it to false in order to disable invitations.	true
allowSignupWithoutInvitation	Boolean	Set it to false in order to make invitation token required for registration.	true
defaultLanguage	String	English by default, you may switch it to 'et' for Estonian.	'en'

languageChoices	List of strings	List of languages to be available on HomePort.
		Format:
		▪ code
		▪ label
			<pre>[
			  {
			  "code": "en",
			  "label": "English"
			  },
			  {
			  "code": "et",
			  "label": "Estonian"
			  }
			]</pre>

supportEmail	Email	Rendered at the footer, if defined.	N/A
supportPhone	String	Rendered at the footer, if defined.	N/A
concealChangeRequest	Boolean	If set to true, "Change request" is concealed from a selection of issue types for non-staff/non-support users	false
currency	String	Override default currency symbol	€
docsLink	URL	Rendered at header, if defined. It should contain valid URL.	N/A
ownerCanManageCustomer	Boolean	Enables owner to remove organization from Manage pane.	false
OWNERS_CAN_MANAGE_OWNERS	Boolean	Enables organization owners to manage other organization owners	true

MANAGER_CAN_MANAGE_TENANTS	Boolean	Enables project manager to provision virtual private cloud.	false
onlyStaffManagesServices	Boolean	Allows to restrict provider management only to staff user. Please note, that if this flag is enabled, OpenStack package provision & modification would be available for staff users only. Also it is assumed that corresponding option is specified in VirtEngine-Waldur MasterMind as well.	false
tenantCredentialsVisible	Boolean	Enables to specify credentials (username and password) in VPC provision form.	true
organizationSubnetsVisible	Boolean	Enables to render subnets from where connection to self-service for organization is allowed.	false
FREEIPA_USERNAME_PREFIX	String	Username prefix for FreeIPA plugin. Note that the value of this option should be the same as for backend.	waldur_
userMandatoryFields	List of strings	List of user profile attributes that would be required for filling in HomePort. Note that backend will not be affected. If a mandatory field is missing in profile, a profile edit view will be forced upon user on any HomePort logged in action. Possible values are: • description • email • full_name • job_title • organization • phone_number	`[ 'full_name', 'email' ]`
userRegistrationHiddenFields	List of strings	List of user profile attributes that would be concealed on registration form in HomePort. Possible values are: • job_title • registration_method • phone_number	`[ 'job_title', 'registration_method', 'phone_number', ]`

Features

In order to conceal optional features, please specify them in **disabledFeatures** setting. Here's a list of available choices:

- alert - manage header alert display
- analytics - enable resource usage and cost analysis
- analytics.cost - enable cost analysis
- analytics.resources - enable resource usage analysis
- ansible - allow to provision applications via Ansible jobs
- appstore - allow to provision resources via Service store
- billing - enable billing (aka accounting)
- compare - allow to compare flavors in Service store
- cost-planning - enable cost planning feature
- dashboard.total_cost - render organization total cost as dashboard chart
- eventlog - enable event logs for users, organizations, projects and resources
- experts - expert providers (aka talent providers) feature
- freeipa - linking of VirtEngine-Waldur account with FreeIPA profile
- help - render help information, for example for providers creation
- import - allow to import existing resources to VirtEngine-Waldur
- intro - step-by-step guide and feature introduction
- mainSearch - full-text-search capability in top-level navigation bar

- monitoring - enable resource monitoring
- notifications - allow user to manage hooks delivered via email, HTTP or Google Cloud Messaging
- offering - enable request-based items feature
- paypal - enable PayPal feature, enabled by default
- premiumSupport - enable premium support service
- private_clouds - enable virtual private cloud management
- projects - render projects link in organization workspace sidebar
- projectCostDetails - allow to manage project cost policy
- providers - allow to manage service providers
- slurm - allow to manage batch processing quotas
- storage - enable virtual block devices and snapshots management
- support - enable support workspace
- support.activity_stream - show activity stream in support workspace
- support.regular_issue_filters - enable issue filters
- team - enable organization team management feature
- user.competence - display "Competence" field in user details.
- user.preferred_language - display "Preferred language" field in user details.
- vms - enable virtual machines management

Language configuration

Default languages are Estonian and English. To add more languages update a **languageChoices** setting.

For instance, lets extend default configuration with Latvian and Lithuanian languages:

```
[
  {
    "code": "en",
    "label": "English"
  },
  {
    "code": "et",
    "label": "Estonian"
  },
  {
    "code": "lv",
    "label": "Latvian"
  },
  {
    "code": "lt",
    "label": "Lithuanian"
  }
]
```

8 languages are supported at the moment. They are: ar, az, en, et, fi, lv, lt, and ru. You can follow ISO 639-1 to match a code with a language.

OpenStack Tenant credentials management

VirtEngine-Waldur HomePort allows to hide OpenStack Tenant credentials in VPC details view. If credentials are hidden the only way to get acces is through **Request direct access** action. Request direct access allows to create a support request for tenant credentials, if support feature is enabled.

Setting **tenantCredentialsVisible** configuration option to **false** will make tenant credentials hidden and **request direct access** action visible.

Please notice that there is a matching configuration setting in VirtEngine-Waldur MasterMind OpenStack plugin which prevents Tenant cre
fields from being exposed over REST API. The settings should be kept in sync for consistency.

See OpenStack plugin configuration for more information.

Identity providers

Overview

There are several authentication methods available:

- **LOCAL_SIGNIN** - allows to sign in using username and password.
- **LOCAL_SIGNUP** - allows to sign up using username and password.
- **SOCIAL_SIGNUP** - allows to sign in and sign up using either Google, Facebook or SmartID.ee OAuth. It is implemented using Satellizer plugin. You can regulate which services should be enabled by specifying client IDs (see below). Note that SmartID.ee is different from SK SmartID.
- **ESTONIAN_ID** - allows to sign in and sign up using either Estonian ID card or Mobile ID. It is implemented using VirtEngine-Waldur Open
- **SAML2** - allows to sign in and sign up using SAML2 identity provider, for example TAAT.
- **VALIMO** - allows to sign in using Mobile ID identity provider.

In order to enable or disable authentication methods, you should ensure that WALDUR_CORE['AUTHENTICATION_METHODS'] configuration variable contains list of enabled authentication methods. By default only 'LOCAL_SIGNIN' is enabled, all other methods are disabled.

Social authentication plugin configuration

To configure Facebook, Google or SmartID login, you should ensure that WALDUR_AUTH_SOCIAL dictionary contains valid configuration.

- **GOOGLE_SECRET** - Google application secret key. Required.
- **GOOGLE_CLIENT_ID** - ID of Google application used for OAuth authentication. If you don't specify it, Google sign up will be disabled.
- **FACEBOOK_SECRET** - Facebook application secret key. Required.
- **FACEBOOK_CLIENT_ID** - ID of Facebook application used for OAuth authentication. If you don't specify it, Facebook sign up will be disabled.
- **SMARTIDEE_SECRET** - SmartID.ee secret key. Required.
- **SMARTIDEE_CLIENT_ID** - ID of SmartID.EE application used for OAuth authentication. If you don't specify it, SmartID.ee sign up will be disabled.
- **USER_ACTIVATION_URL_TEMPLATE** - template for frontend activation url. Will be used in email message. Should contain placeholders for `{user_uuid}` and `{token}`. Example: http://example.com/#/activate/{user_uuid}/{token}/

Setting up of application on Facebook side: https://developers.facebook.com/docs/facebook-login

Setting up of Google profile for VirtEngine-Waldur deployment: https://developers.google.com/identity/protocols/OAuth2 .

Setting up of application on Smart ID side: https://smartid.ee/how/

Estonian ID login

To configure Estonian ID login please configure configuration of waldur-auth-openid plugin and ensure that WALDUR_AUTH_OPENID dictionary contains these options.

- **NAME** - on user registration following name will be used for user's registration_method field. Default value - openid.
- **LOGOUT_URL** - for security reasons OpenID logout URL should be specified so that when user logs out from VirtEngine-Waldur, he is lo
from OpenID account too.

TaaT login

MasterMind configuration: TaaT authentication via SAML2

LDAP login

MasterMind configuration: LDAP authentication using FreeIPA identity management

Valimo login

VirtEngine-Waldur Atuh Valimo plugin enables mobile phone authentication using Valimo PKI.

It is assumed that the following settings are specified in the WALDUR_AUTH_VALIMO dictionary:

Name	Description	Example
URL	Valimo URL.	https://example.com
AP_ID	Application provider ID.	https://example/MobilePKI_AP
AP_PWD	Application provider password.	example_password
DNSName	Name of VSS server, blank by default.	example
SignatureProfile	Signature profile to be used in the transaction processing.	https://example.com/MobilePKI
verify_ssl	Enable SSL certificates validation, disabled by default.	True or False
cert_path	Path to file with SSL certificate.	/home/user/path/to/cert/example.cer
key_path	Path to file with SSL key.	/home/user/path/to/key/example.key
message_prefix	prefix of message that will be send to user.	Login code:

LDAP authentication using FreeIPA identity management

VirtEngine-Waldur allows you to authenticate using LDAP server.

Prerequisites

- It is assumed that LDAP server is managed by FreeIPA. Although LDAP authentication would work without FreeIPA, you may need to use custom configuration for VirtEngine-Waldur MasterMind in this case.
- Please ensure that VirtEngine-Waldur API server has access to the LDAP server. By default LDAP server listens on TCP and UDP port 38 port 636 for LDAPS (LDAP over SSL, see below).
  If this port is filtered out by firewall, you wouldn't be able to authenticate via LDAP.
- You should know LDAP server URI, for example, ldap://ipa.demo1.freeipa.org
- You should know username and password of LDAP admin user. For example, currently FreeIPA demo server uses username=admin and password=Secret123.

Installation instructions

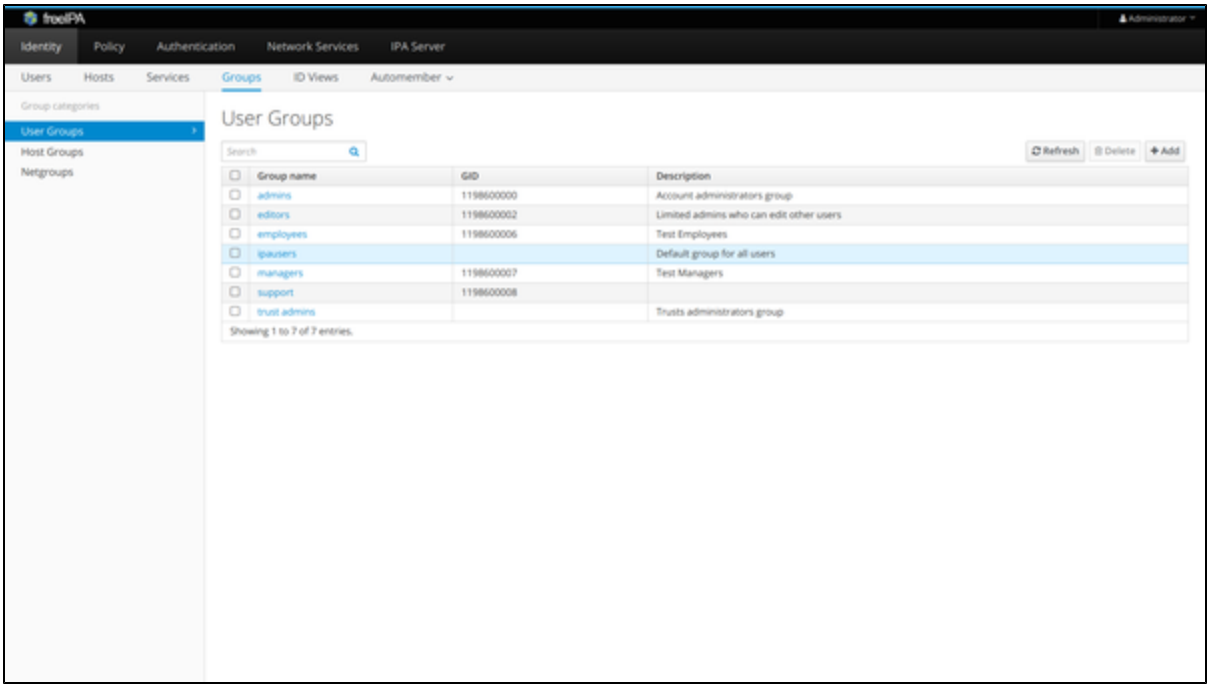
1. Install **django-auth-ldap** from VirtEngine-Waldur RPM repository. This package would also install **python-ldap** package.
2. Add following configuration to /etc/waldur/override.conf.py

LDAP configuration

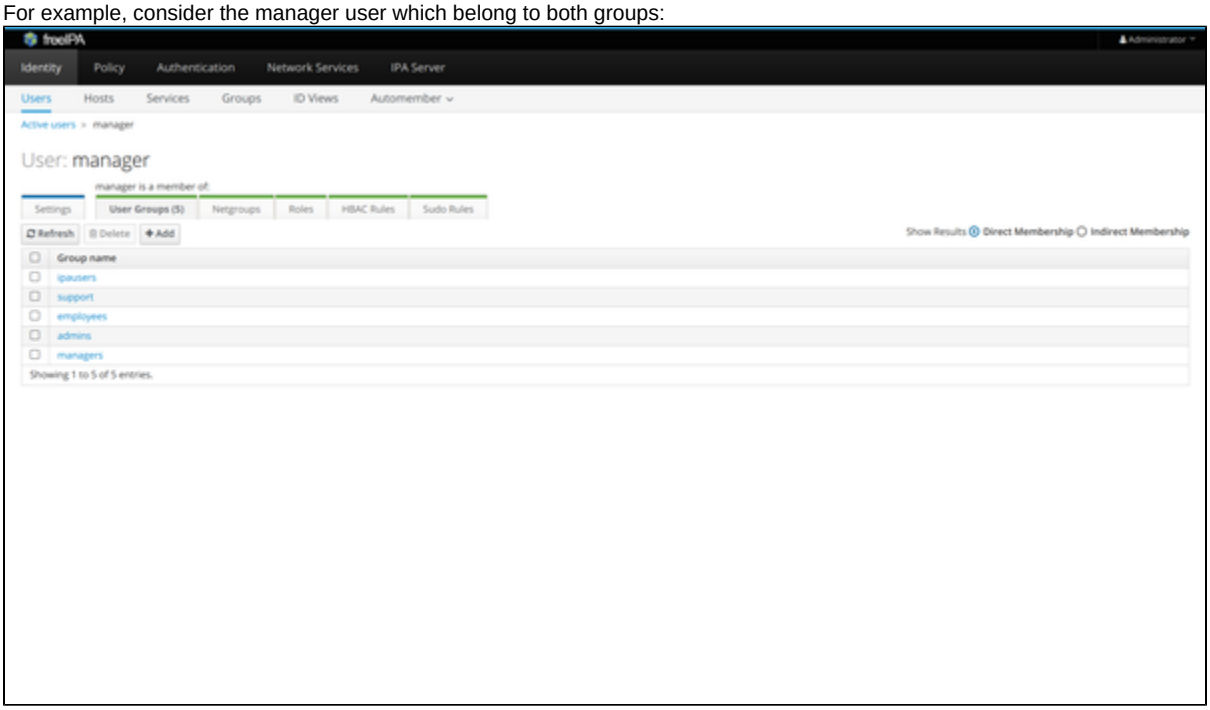
```
import ldap
from django_auth_ldap.config import LDAPSearch, GroupOfNamesType

# LDAP authentication.
# See also:
https://django-auth-ldap.readthedocs.io/en/latest/authentication.ht
ml
AUTHENTICATION_BACKENDS += (
    'django_auth_ldap.backend.LDAPBackend',
)

AUTH_LDAP_SERVER_URI = 'ldap://ipa.demo1.freeipa.org'

# Following variables are not used by django-auth-ldap,
# they are used as templates for other variables
AUTH_LDAP_BASE = 'cn=accounts,dc=demo1,dc=freeipa,dc=org'
AUTH_LDAP_USER_BASE = 'cn=users,' + AUTH_LDAP_BASE

# Format authenticating user's distinguished name using template
AUTH_LDAP_USER_DN_TEMPLATE = 'uid=%(user)s,' + AUTH_LDAP_USER_BASE

# Credentials for admin user
AUTH_LDAP_BIND_DN = 'uid=admin,' + AUTH_LDAP_USER_BASE
AUTH_LDAP_BIND_PASSWORD = 'Secret123'

# Populate the Django user from the LDAP directory.
AUTH_LDAP_USER_ATTR_MAP = {
    'full_name': 'displayName',
    'email': 'mail'
}

# Set up the basic group parameters.
AUTH_LDAP_GROUP_BASE = "cn=groups," + AUTH_LDAP_BASE
AUTH_LDAP_GROUP_FILTER = "(objectClass=groupOfNames)"
AUTH_LDAP_GROUP_SEARCH = LDAPSearch(AUTH_LDAP_GROUP_BASE,
    ldap.SCOPE_SUBTREE, AUTH_LDAP_GROUP_FILTER)
AUTH_LDAP_GROUP_TYPE = GroupOfNamesType(name_attr="cn")

AUTH_LDAP_USER_FLAGS_BY_GROUP = {
    'is_staff': 'cn=admins,' + AUTH_LDAP_GROUP_BASE,
    'is_support': 'cn=support,' + AUTH_LDAP_GROUP_BASE,
}
```

Please replace default values for LDAP URI and credentials with your own values.
444pxDon't forget to reload waldur-uwsgi after configuration is updated in order to apply it.

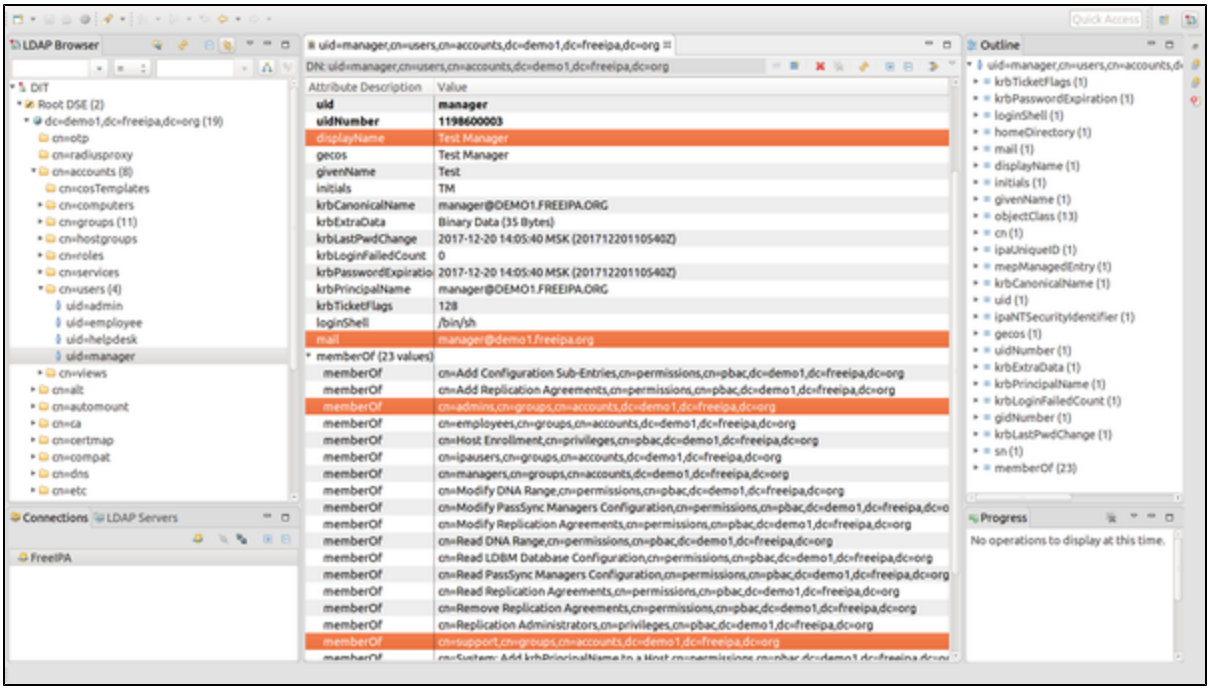
3. Ensure that **admins** and **support** groups exist in LDAP server. You may do it using FreeIPA admin UI.



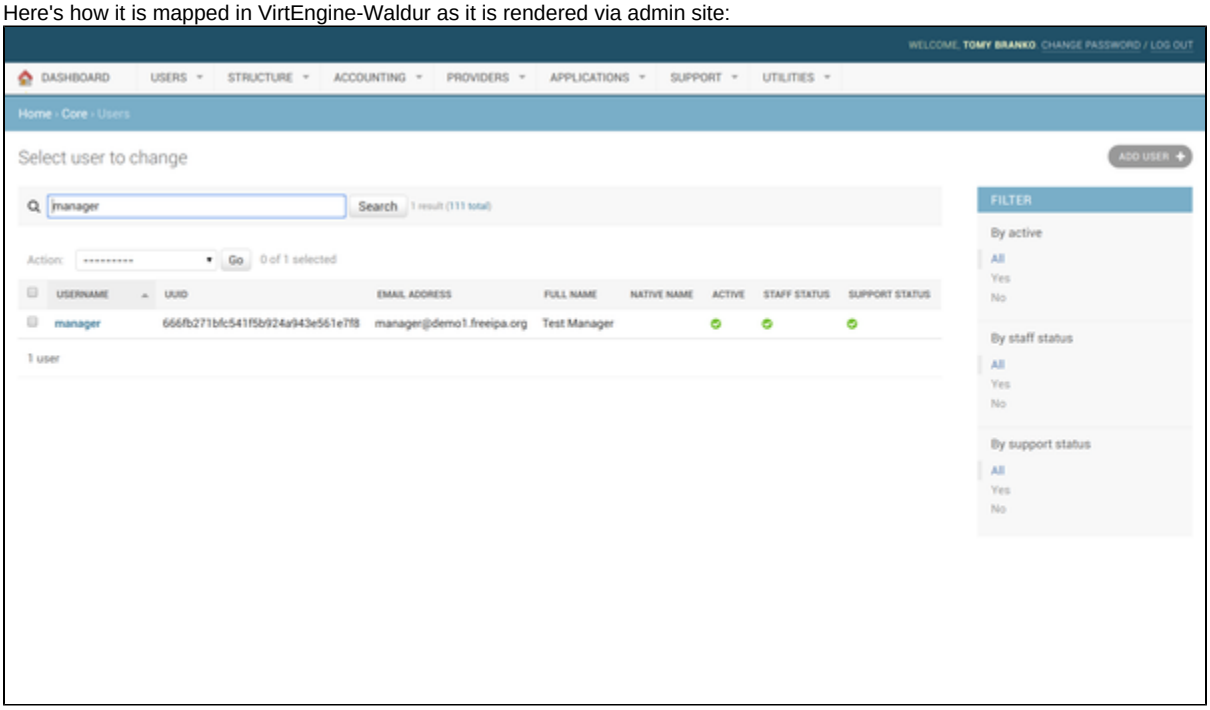
4. If user is assigned to admins group in LDAP, he becomes staff in VirtEngine-Waldur.
   If user is assigned to support group in LDAP, he becomes support user in VirtEngine-Waldur.
   For example, consider the manager user which belong to both groups:



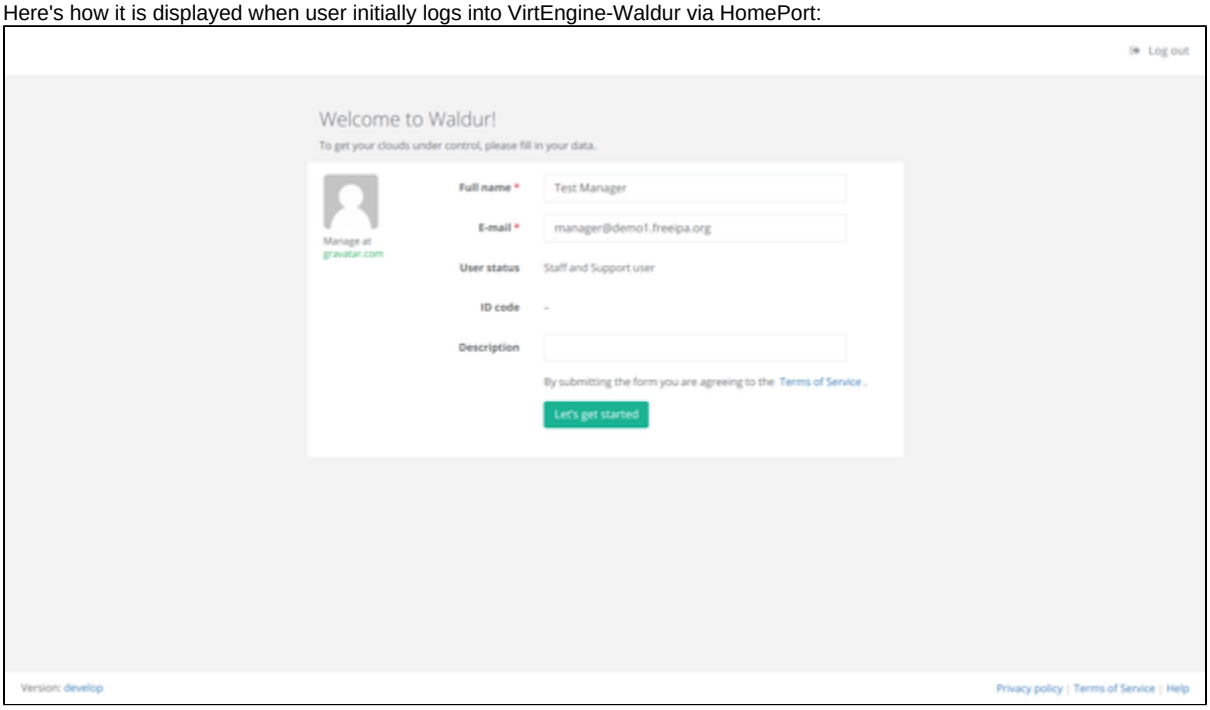
5. displayName attribute in LDAP is mapped to full_name attribute in VirtEngine-Waldur.
   mail field in LDAP is mapped to email attribute in VirtEngine-Waldur.
   Consider for example, the following user attributes:



Here's how it is mapped in VirtEngine-Waldur as it is rendered via admin site:



Here's how it is displayed when user initially logs into VirtEngine-Waldur via HomePort:



TaaT authentication via SAML2

TAAT stands for Estonian Academic Authentication and Authorization Infrastructure (Eesti haridus- ja teadusasutusevahelise autentimise ja autoriseerimise taristu). It enables electronic identities (user accounts) issued by education or research institutions to be used to access several web based services. For more information, please see taat.edu.ee

It is possible to configure VirtEngine-Waldur to use TaaT as an identity provider. TaaT uses SAML2 authentication and authorization standard to communicate with service providers such as VirtEngine-Waldur. There is a VirtEngine-Waldur plugin developed for using SAML2, which is called w
.

Configuration

Retrieve TaaT metadata:

1. You can retrieve TaaT metadata by saving its certificate to `/etc/waldur/saml2` and specifying metadata URL in VirtEngine-Waldur cor
   The certificate is used to retrieve the metadata securely. TaaT certificates can be downloaded from here: http://taat.edu.ee/main/dokume
   ndid/sertifikaadid/.
   Metadata URL for test hub is https://reos.taat.edu.ee/saml2/idp/metadata.php and for production hub is https://sarvik.taat.edu.ee/saml2/id
   p/metadata.php.
   Note, the certificate must correspond to the hub you want connect to.
2. You can save the metadata file to the `/etc/waldur/saml2` and specify its path in VirtEngine-Waldur configuration. It is suggested to us
   approach as
   it does not require to redownload the metadata file every time it changes.

Start by replacing *saml2.conf.py.example* with *saml2.conf.py* `in /etc/waldur. Then update parameters in` **`WALDUR_AUTH_SAML2`** var
iable in *`saml2.conf.py`*:

- **name** - used for assigning the registration method to the user. `saml2` is used by default.
- **ALLOW_TO_SELECT_IDENTITY_PROVIDER** - enabled by default, otherwise identity provider URL and label should be specified.
- **ENABLE_SINGLE_LOGOUT** - allows to enable SAML2 single logout, disabled by default.
- **IDENTITY_PROVIDER_URL** - URL of SAML2 metadata endpoint for selected identity provider.
- **IDENTITY_PROVIDER_LABEL** - name of identity provider rendered in login button.
- **logout_requests_signed** - indicates if the entity will sign the logout requests, enabled by default.
- **authn_requests_signed** - indicates if the authentication requests sent should be signed by default, enabled by default.
- **signature_algorithm** - identifies the Signature algorithm URL according to the XML Signature specification, SHA1 is used by default.
- **digest_algorithm** - identifies the Message Digest algorithm URL according to the XML Signature specification, SHA1 is used by default.
- **nameid_format** - identifies NameID format to use. None means default, empty string ("") disables addition of entity.
- **xmlsec_binary** - full path to the xmlsec1 binary file, e.g. `/usr/bin/xmlsec1`
- **base_url** - specify VirtEngine-Waldur MasterMind base URL, e.g. `https://waldur.example.com`. It is required for assertion consume
  logout services and entity ID.
- **attribute_map_dir** - path to the directory with attribute mapping. If you want to use a default one, specify `/etc/waldur/attribute-m
  aps/`.

- **debug** - set to True to output debugging information.
- **idp_metadata_local** - paths to the metadata files, which were saved locally, e.g. `['/etc/waldur/saml2/sarvik_metadata.xml']`
- **idp_metadata_remote** - dictonaries of urls and certificates to retrieve the metadata files, e.g.

```
[
    {
        # URL to the metadata XML file
        'url': 'https://reos.taat.edu.ee/saml2/idp/metadata.php',
        # path to PEM certificate required to retrieve the remote
metadata XML file
        'cert': '/etc/waldur/saml2/reos.crt',
    },
]
```

- **log_file** - path to the log file. If you want to disable logging of the SAML2 related stuff, leave it empty.
- **log_level** - severity of the logging.
- **cert_file** - path to the certificate. If you want to use the default one (generated when package is installed), specify `/etc/waldur/sp.crt`.
- **key_file** - path to the key file. If you want to use the default one (generated when package is installed), specify `/etc/waldur/sp.key`.
- **required_attributes** - SAML attributes that are required to identify the user, e.g. `['cn', 'mail', 'schacPersonalUniqueID'].`
- **optional_attributes** - SAML attributes that may be useful to have but not required, e.g. `['schacHomeOrganization', 'preferredLanguage']`.
- **saml_attribute_mapping** - mapping between SAML attributes and User fields, e.g.

```
{
    'schacPersonalUniqueID': ['username', 'civil_number'],
    'cn': ['full_name'],
    'mail': ['email'],
    'preferredLanguage': ['preferred_language'],
    'schacHomeOrganization': ['organization'],
}
```

- **organization** - details of the organization, which is responsible for service. You can specify information in multiple languages, e.g.

```
{
    'name': [('DET-IO & OpenNode OU', 'et'), ('DET-IO & OpenNode LLC', 'en
    'display_name': [('DET-IO & OpenNode', 'et'), ('DET-IO & OpenNode', 'e
    'url': [('https://opennodecloud.com/', 'et'),
('https://opennodecloud.com/', 'en')],
}
```

Finally, restart VirtEngine-Waldur MasterMind.

Janus

https://taeva.taat.edu.ee/module.php/janus/index.php

1. Create a new connection:

- New connection ID must be equal to the `base_url` in `saml.conf.py` + `/apu-auth/saml2/metadata/`
- Choose `SAML 2.0 SP` for connection type.
- Click `Create` button

2. In connection tab select or create ARP. Fields that ARP include must be in the `saml_attribute_mapping`.
3. Navigate to the `Import metadata` tab and paste same URL as in the first step. Click on the `Get metadata`.
4. Navigate to the `Validate` tab and check whether all the tests pass. You can fix metadata in `Metadata` tab.

Logging

WAL-723 - Add description of MasterMind logging
TO DO

Multi-tenant deployments

There are two options for multi-tenant deployment of VirtEngine-Waldur. Depending on scenario, business model and security requirements, one c might be preferred.

Single-instance multitenancy

Using in-built Organization structure units.

Pros:

- Easier to setup and manage.
- Small resource consumption.
- Easier statistics aggregation.

Cons:

- Security isolation is limited to application level.
- Whitelabeling is limited, hard to achieve custom domain name / logo.

Multi-instance multi-tenancy

Separate deployments of VirtEngine-Waldur.

Pros:

- Higher security isolation.
- Possibility for full white-labelling

Cons:

- Requires more resources.

Organization management

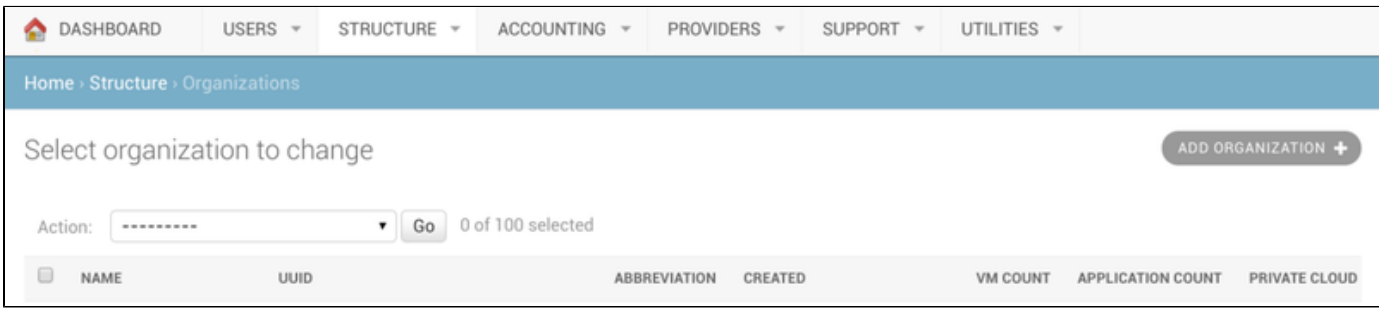
Introduction

There are two modes of operation in VirtEngine-Waldur MasterMind with regard to creation and management of Organizations. The first approach for managed environments where new organizations are created manually or via API by staff personnel. Alternative is to allow self-registration of Organizations and use Plans to limit capabilities of the Organizations. A more controlled approach is described below.

Organization on-boarding

To create a new Organization, you can either use API with staff account, or go to VirtEngine-Waldur MasterMind admin interface, select **Structure**

Organizations and click on **Add organizations**.



Once the creation form is filled, you have a new Organization!

Organization off-boarding

To remove an organization, two scenarios are supported.

First scenario – removal via API. Validates that Organization being removed does not have any existing Resources, otherwise raises a validation error.

Second scenario – removal via Admin interface. Does cascade deletion of the connected entities but does not schedule resource deletion (deletion happens from DB only).

Configuration options

There are several configuration options which allow to enable or disable particular actions depending on user role.

OWNERS_CAN_MANAGE_OWNERS enables organization owners to grant, revoke and update permission to owned organization; create, resend and cancel invitations to owned organization. This option is enabled by default.

WALDUR_OPENSTACK['MANAGER_CAN_MANAGE_TENANTS'] enables project manager to provision virtual private cloud, disabled by default. Staff and owner always have access to package. Project manager has access only if this option is enabled.

System Provider Management

System providers are used for offering services to all organizations in VirtEngine-Waldur deployment. Typically they are used for bulk sales (e.g. V Private Clouds, MS Exchange domains) or for chunking enterprise agreements (AWS, Azure).

System providers can be optionally limited to certain projects by Staff personnel, through "Service-project-link" management.

Shared OpenStack Provider Management

Requirements for OpenStack

In order to integrate an OpenStack-based cloud as a shared provider, the following data is required:

- URL of keystone's public endpoint (v3).
- Access to public interfaces of Keystone, Nova, Cinder, Neutron and Glance should be granted to VirtEngine-Waldur MasterMind server.
- Admin credentials (username/password), can be a domain administrator (in this case also domain name).
- External network UUID - the network will be by default connected to all created tenants.
- Pricing - package configurations. Each package is defined by max core number, RAM and disk.

Optionally, the following is suggested:

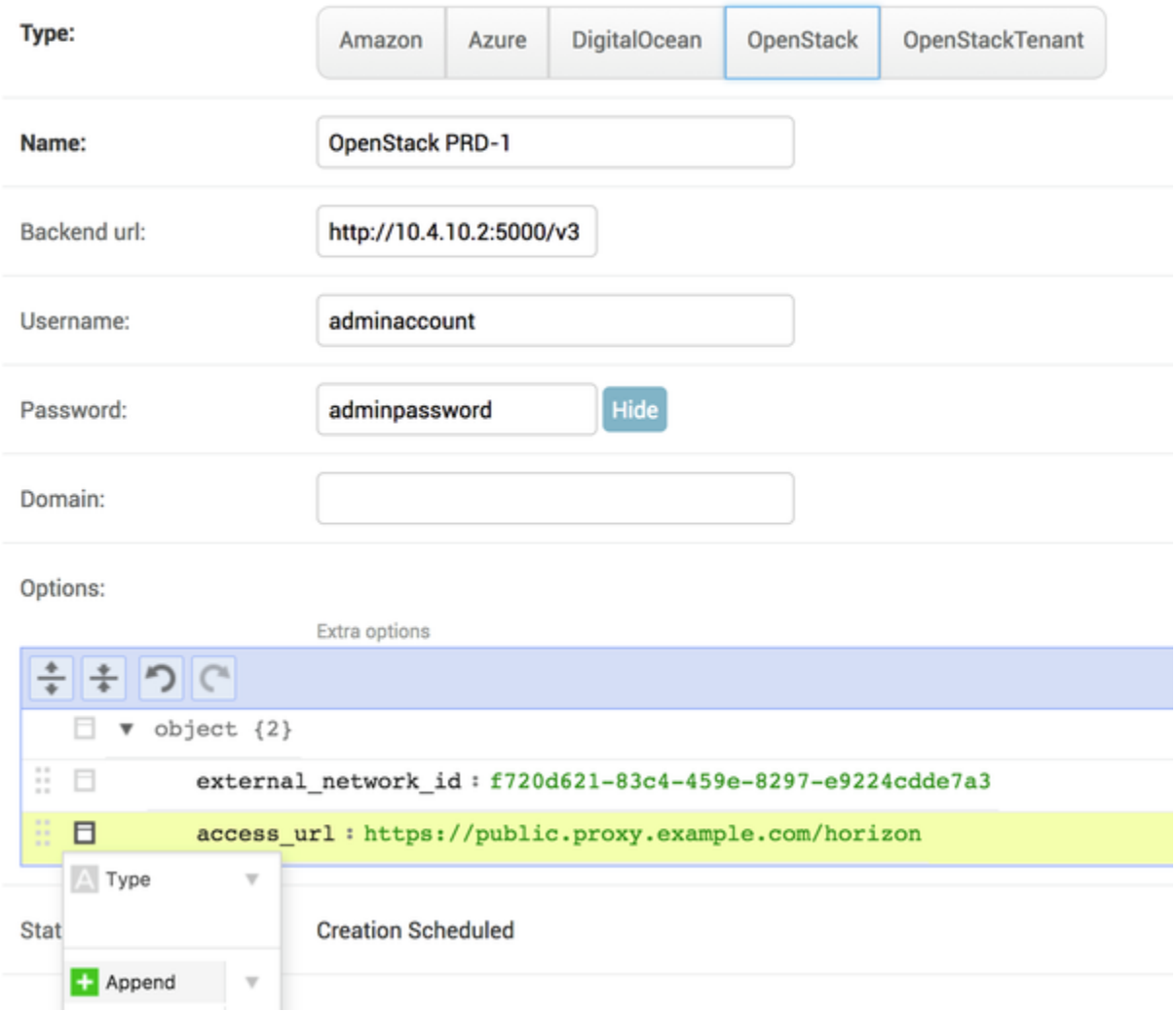
- Homepage of a service provider
- List of certifications

- Geolocation of data center for the map.

Setting up Shared Provider

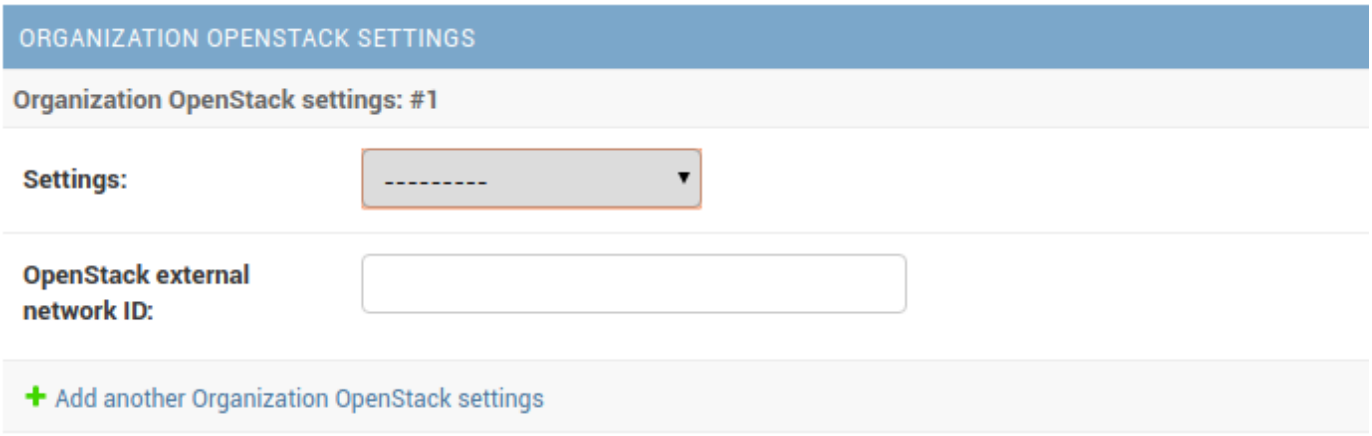
To setup a shared OpenStack as a VPC provider, please go to VirtEngine-Waldur MasterMind Admin interface with a staff account. Go to Structure provider settings and click on adding a new one.

1. Select Type "OpenStack" (NB! Not OpenStackTenant – this one is used for VM provisioning and shouldn't be used as shared provider in production environments).
2. Name a provider – how it will be visible in API and HomePort's Service Store and in Provider list.
3. Configure access parameters: URL, username and password.
4. Define specific customisation options. To add an option select append on item block under the object tree. Most typical are:
    - **external_network_id** – external network to connect to when creating a VPC from this provider.
    - **access_url** - a URL to access OpenStack Horizon dashboard from a public network. Typically a reverse proxy URL in production deployments.
    - **flavor_exclude_regex** - flavors matching this regex expression will not be pulled from the backend.
    - **dns_nameservers** - default value for new subnets DNS name servers. Should be defined as list.
    - **create_ha_routers** - create highly available Neutron routers.
5. Click Save. The provider will be attempted to be configured / synchronised. Should any errors happen, it will be shown in the error_message field of service settings.



Support for Organization specific OpenStack networks

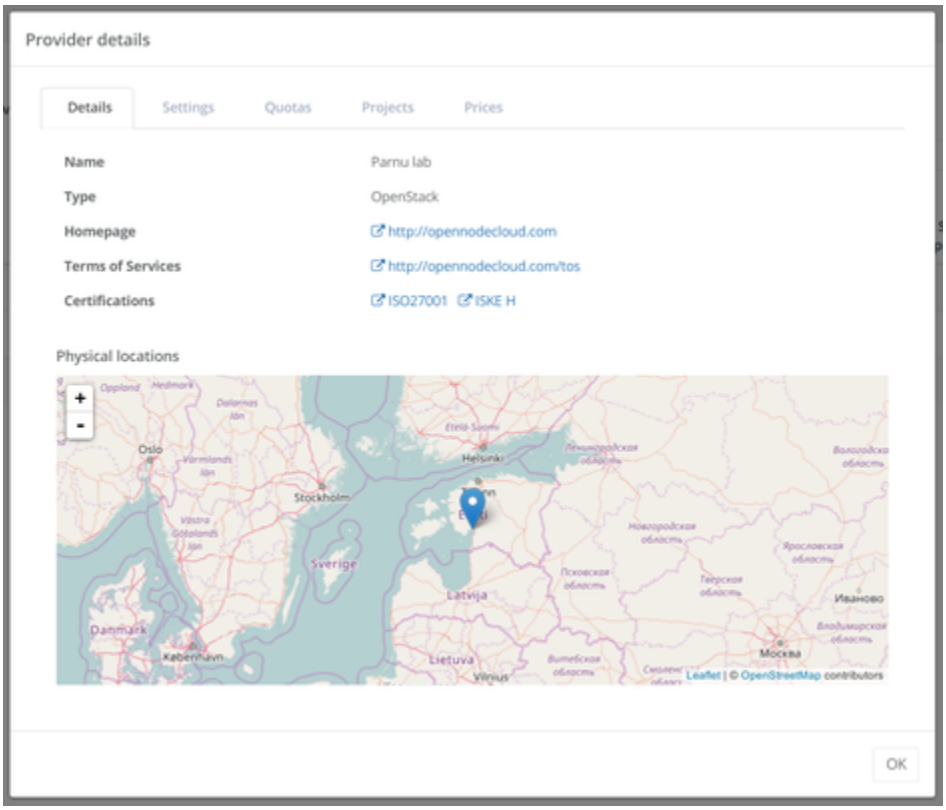
You can provide specific external network for all OpenStack Tenants created by Organiztion by providing external network UUIDs in Organization configuration in VirtEngine-Waldur Mastermind admin portal.



Specify providers physical locations

Physical location may be specified in "Geolocations" field. It is used for rendering map in HomePort.
For example:

```
[
  {
    "latitude": 58.3789835,
    "longitude": 24.4579569
  }
]
```



VPC package template management

Virtual Private Cloud package defines size of a "sandbox" delivered to the user after the purchase. For OpenStack, this corresponds to a Tenant with certan quotas.

VPC package template creation

To create a VPC package template open VirtEngine-Waldur MasterMind Admin as a staff user. From the top menu go to Providers VPC Packages package templates and click 'Add VPC Package Template' button in the top right corner. A VPC package template creation form will be shown. Please complete following steps to finish template creation:

- Specify template name;
- Select size category - used for grouping in UI.
- Add description (optional);
- Add an icon URL (optional);
- Select service settings corresponding to OpenStack for which the packages are defined (a template can be created only with shared service settings);
- Fill package components amount and price for 30 days.

Do not worry if price for 1 day is not shown when you add price for 30 days, it will be updated after template is saved. It is also possible to recalculate price by saving a template.

> Please keep in mind that if `amount` of any resource is set to 0 - a price for the component will be 0 as well. It also means it is possible to create a free of charge template by leaving either an 'amount' or a 'price for 30 days' parameter empty. A price for the template is shown on VPC template list in the 'Price' column.
>
> Price cannot be changed after any package has been created from package template

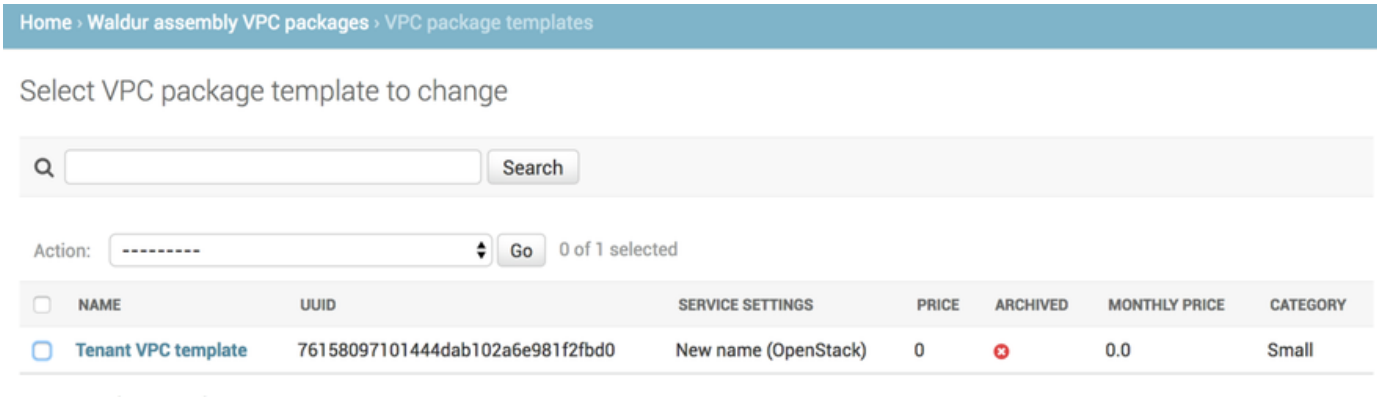
> Please do not set 'Archived' flag. If 'Archived' flag is set it will not be possible to create a tenant using current template.

Click 'Save' button or 'Save and continue editing' if you want to double check the template after creation.

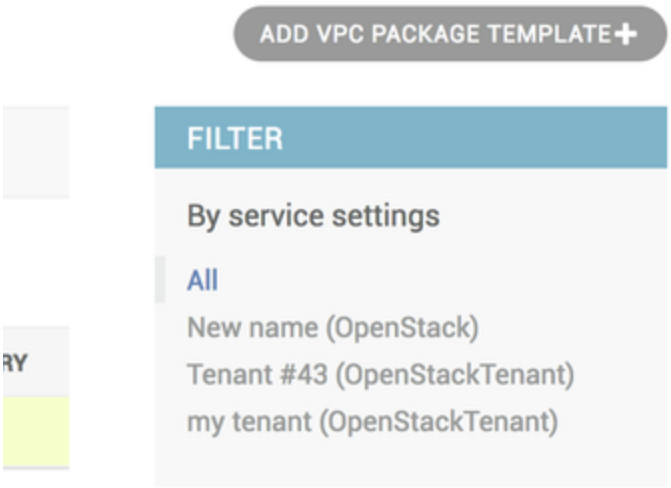
VPC package template list

To list all available package template login to VirtEngine-Waldur MasterMind Admin as a staff user and go to Providers VPC PackagesVPC packa templates from the menu panel.

A list of package template will be shown to you.

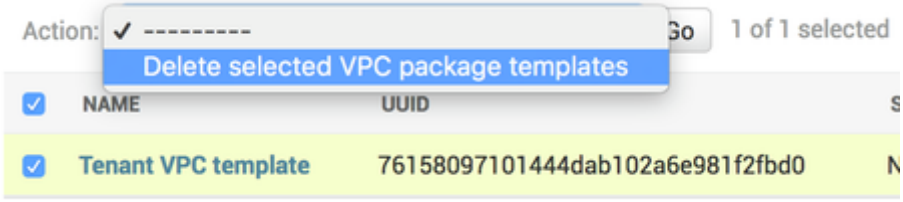


It is possible to filter templates by its name in the Search pane or by a service settings in the filter form.



VPC package template deletion

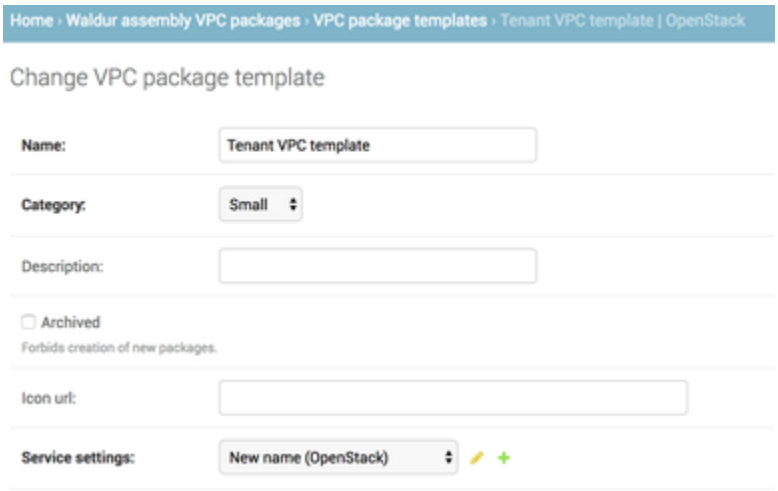
Open a package template list view. Find a template you want to delete and select a template by clicking a checkbox in the template row. If a checkbox in the header is checked all templates are going to be selected. You also can select multiple templates. After desired templates are selected go to action drop down menu, select 'Delete selected VPC package templates and click a 'Go' button.



> Please note that it is impossible to delete a package template if there is at least one package linked to it. In this case a package template should be archived, which will prevent creation of new linked packages.

VPC package template update

To update a template go to a VPC package templates list view and click on a template you want to edit. A template editing form with pre-filled fields will be presented. Change required fields and hit 'Save' button.



> Please keep in mind, for templates that have connected OpenStack packages it is not possible to:
>
> - Update the price;

- Update fields that have been used to create an OpenStack package, which are 'Name', 'Category' and 'Service settings'.

VPC package template archiving

It is possible to forbid any package creation from a template if it is not desired. In order to do that, the template has to be marked as 'Archived'. To mark a template as 'Archived' go to template editing form and check 'Archived' flag.

VPC permissions

By default only staff or organization owner are able to execute actions that have influence on VPC price (delete, change package or create new one). It is possible to allow manager to execute this actions too by updating following variable in VirtEngine-Waldur settings:

```
WALDUR_OPENSTACK['MANAGER_CAN_MANAGE_TENANTS'] = True
```

Shared SLURM

Requirements for SLURM

In order to integrate an SLURM-based batch processing farm as a shared provider, the following is required:

- SSH host address to a node, from where slurm commands could be executed.
- Username that has Slurm operator role (https://slurm.schedmd.com/user_permissions.html). Operator is needed as VirtEngine-Waldur dyn creates accounts based on user's choice of FreeIPA account.
- VirtEngine-Waldur public key must be added as authorized_key for the operator's username.
- Slurm pricing information needs to be provided. Currently supported are price per: cpu/h, gpu/h, ram/h.
- Slurm login node must be configured to authenticate users coming from FreeIPA connected to VirtEngine-Waldur.

For setup of SLURM provider, please see SLURM plugin configuration#ConfigureVirtEngine-WaldurSLURMplugin .

ServiceStore configuration

VirtEngine-Waldur allows you to provision complex infrastructure using following methods:

1. Ansible-based applications
2. Expert request configuration
3. Request-based offerings

Ansible-based applications

- Introduction
- System design
- Quick start guide
- HomePort configuration
- Ansible plugin configuration
- Ansible playbook configuration
- Example Ansible playbook

Prerequisites: it is assumed that reader already has working experience with Ansible. Otherwise, please consult official Ansible guide.

EXPERIMENTAL

Introduction

Consider the following use case. User needs to deploy application which consists of multiple interconnected resources.
For example, let's say he wants to deploy a multi-tiered web application, like an e-commerce, with backend server on first VM and fronted app on second VM.

Currently in order to accomplish such a task user shall:

1. issue request to provision first VM, wait for it to complete, gather some info, such as IP address and store it;
2. execute some script on first VM, for example, via Ansible;
3. issue request to provision second VM, wait for it to complete;
4. execute some script on second VM, passing information gathered in previous steps.

Instead of performing all these steps manually, you may use Ansible plugin for VirtEngine-Waldur. It allows you to provision complex applications u
Ansible playbook.

System design

This feature is implemented as set of plugins:

1. VirtEngine-Waldur Ansible plugin is installed along with VirtEngine-Waldur MasterMind. It allows to upload and execute Ansible playbooks
   Ansible VirtEngine-Waldur module.
   Similarly to Ansible Tower, it's designed to be the hub for all of your automation tasks.
2. Ansible VirtEngine-Waldur module contains collection of Ansible modules to allow provisioning and management of infrastructure under Vi
   through Ansible playbooks.
   Similarly to Ansible OpenStack modules, it could be used as part of your playbook or via Ansible CLI.
3. Ansible module for HomePort enables you to use user-friendly user interface instead of CLI.

Quick start guide

In order to use Ansible-based applications, you should perform following steps:

1. Install ansible, waldur-ansible and ansible-waldur-module packages from RPM.
2. Generate SSH key-pair used by VirtEngine-Waldur for resource provisioning using ssh-keygen utility.
3. Upload private part of SSH key to /etc/waldur/id_rsa
4. Upload public part of SSH key to VirtEngine-Waldur MasterMind database either via admin site or REST API.
5. Specify API_URL, PRIVATE_KEY_PATH, PUBLIC_KEY_UUID in WALDUR_ANSIBLE.
6. Ensure that there's actually SSH public key in VirtEngine-Waldur database with specified identifier PUBLIC_KEY_UUID.
7. Ensure that this SSH public key has is_shared=True. Otherwise you wouldn't be able to provision any virtual machine.
8. Compose Ansible playbook and archive it to ZIP file.
9. Upload ZIP archive to VirtEngine-Waldur via admin site.
10. Check that everything works by provisioning application via HomePort.

VirtEngine-Waldur Ansible plugin is available in VirtEngine-Waldur RPM repository. In order to install it you should issue following command:

```
yum install waldur-ansible
```

Ansible VirtEngine-Waldur module is also available in VirtEngine-Waldur RPM repository:

```
yum install ansible-waldur-module
```

Ansible module for HomePort is distributed as part of HomePort RPM package and doesn't need to be installed separately.

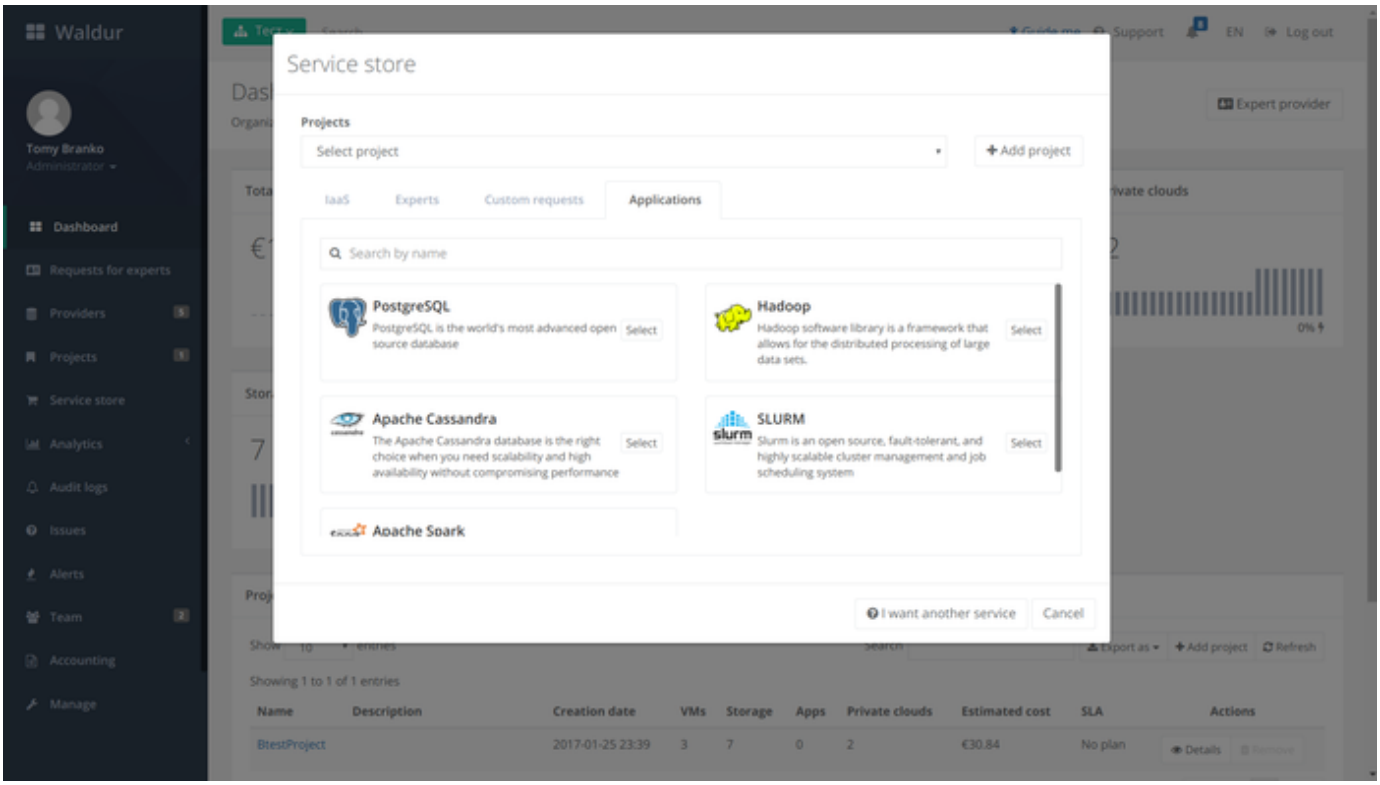
Please note, that currently Ansible VirtEngine-Waldur module supports OpenStack Tenant provider only.

NB! Due to Ansible license restrictions VirtEngine-Waldur is not providing Ansible itself, please install it on MasterMind and Celery nodes as a sepa
component, for example, using this guide: https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-ansible-on-centos-7

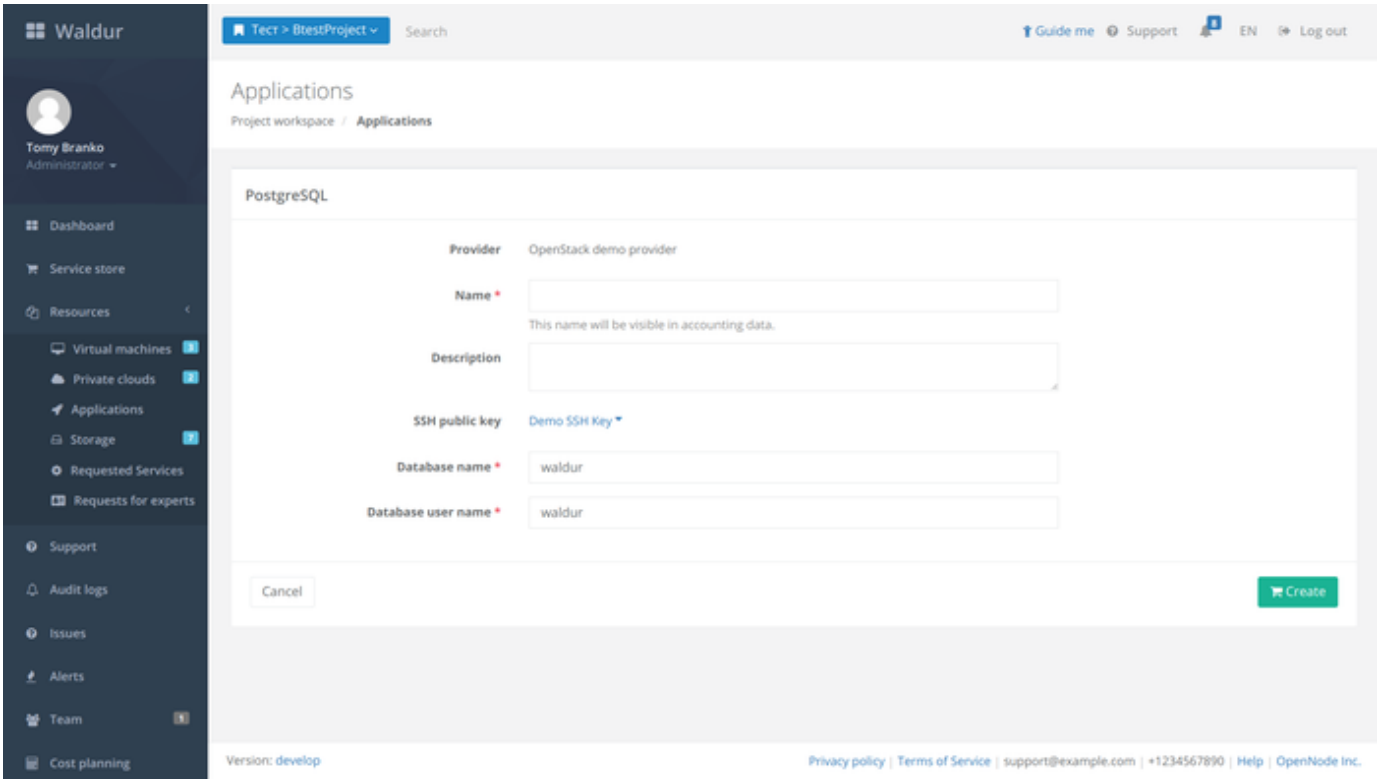
HomePort configuration

In order to enable Ansible applications feature in HomePort, please ensure that **ansible** feature is enabled. See also documentation for feature toggling.

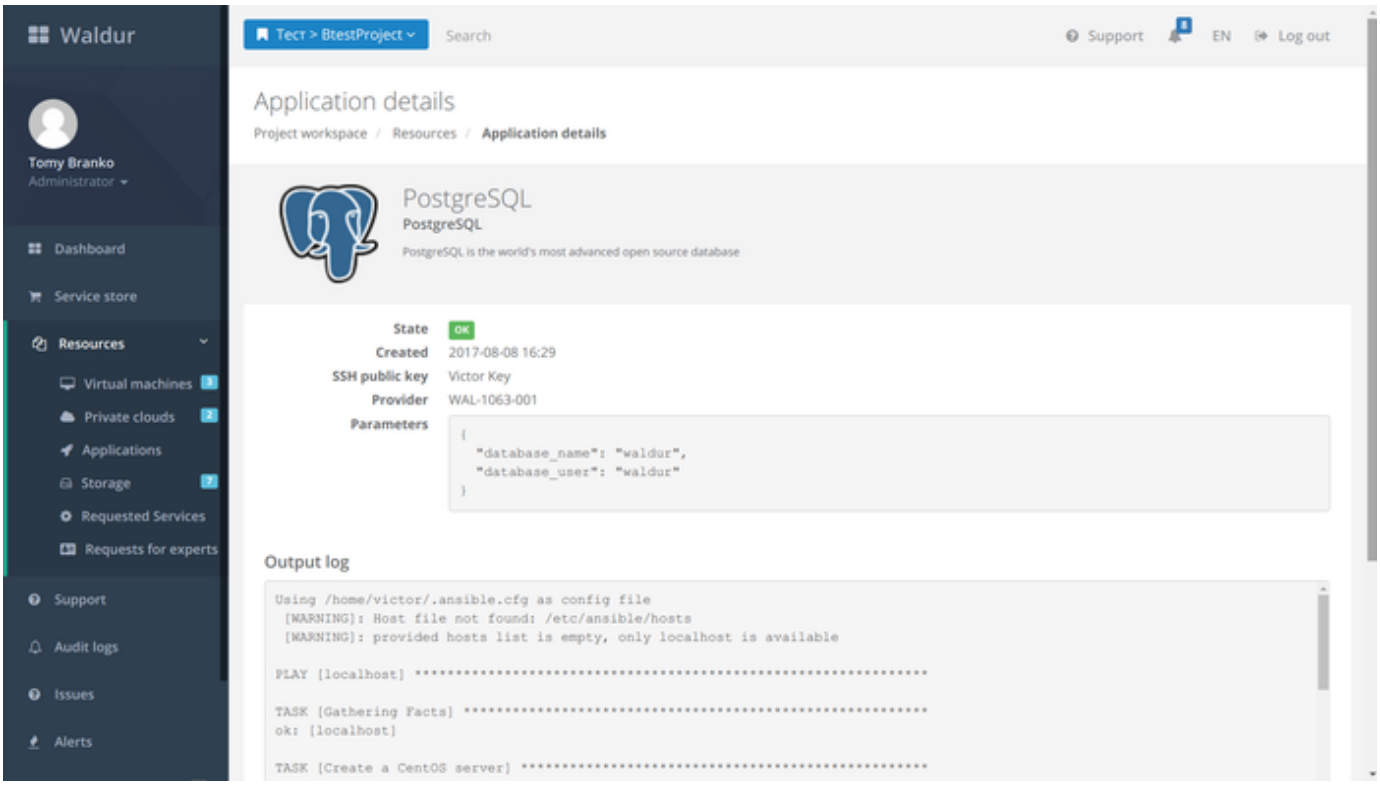
In HomePort, when user clicks on **Service store** link in sidebar menu and selects **Applications** category, all playbooks are rendered:



Then, when user selects application and provider, application provision form is rendered:



Finally, when user clicks on **Create** button, Ansible playbook is executed. Its execution log is rendered below:



Ansible plugin configuration

In order to configure VirtEngine-Waldur Ansible plugin you should update WALDUR_ANSIBLE dictionary. It contains following options:

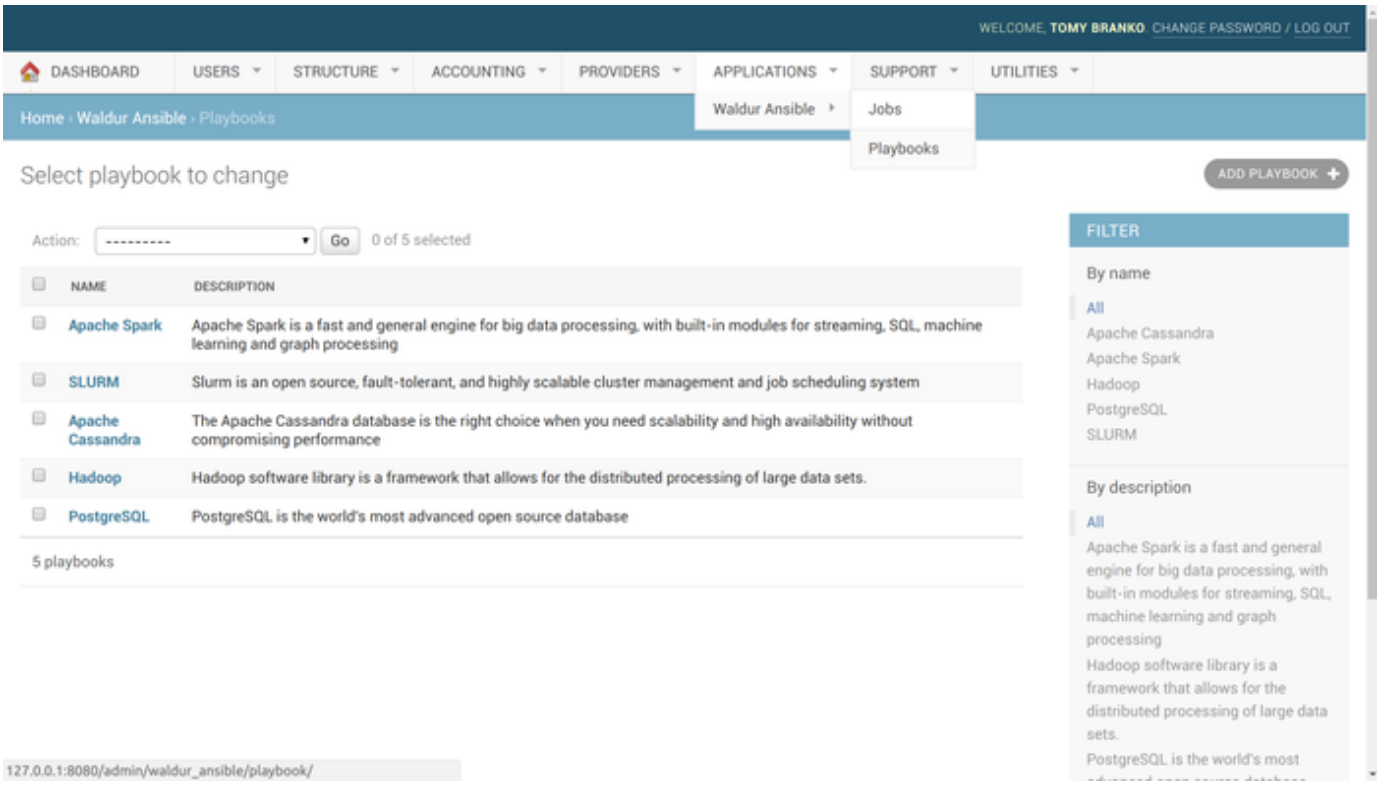
Name	Type	Default value	Description
PLAYBOOKS_DIR_NAME	String	ansible_playbooks	Relative directory name where uploaded playbooks are stored. Note that absolute path looks like /var/lib/waldur/media/ansible_playbooks/
PLAYBOOK_EXECUTION_COMMAND	String	ansible-playbook	Command name used to execute Ansible playbook. Note that it may be tweaked if you have specific version of Ansible installed in virtual Python environment, for example.
PLAYBOOK_ARGUMENTS	Array of strings	['--verbose']	Default arguments passed to Ansible playbook. Note that by default verbose mode is enabled. You may want to reduce verbosity or pass extra flags instead.
ANSIBLE_LIBRARY	String	/usr/share/ansible-waldur/	Absolute directory name for Ansible VirtEngine-Waldur module l Note that it should match location used by ansible-waldur-module RPM package.
PLAYBOOK_ICON_SIZE	Tuple of integers	(64, 64)	Ansible playbook image thumbnail size. It is rendered in HomePort as part of Service store category selector.
API_URL	URL	http://localhost/api/	VirtEngine-Waldur MasterMind REST API root URL used by An VirtEngine-Waldur module. It may be tweaked depending on int network configuration.
PRIVATE_KEY_PATH	String	/etc/waldur/id_rsa	Absolute path to private SSH key used by Ansible VirtEngine-W module for initial virtual machine configuration. Please ensure that all nodes with Celery workers use the same shared private key for Ansible VirtEngine-Waldur module.
PUBLIC_KEY_UUID	String	Not defined	UUID of public key uploaded to VirtEngine-Waldur MasterMind. public key and private key should belong to the same pair.

Please note that you should specify values for API_URL, PRIVATE_KEY_PATH, PUBLIC_KEY_UUID options. It is safe enough to leave other options with default values.

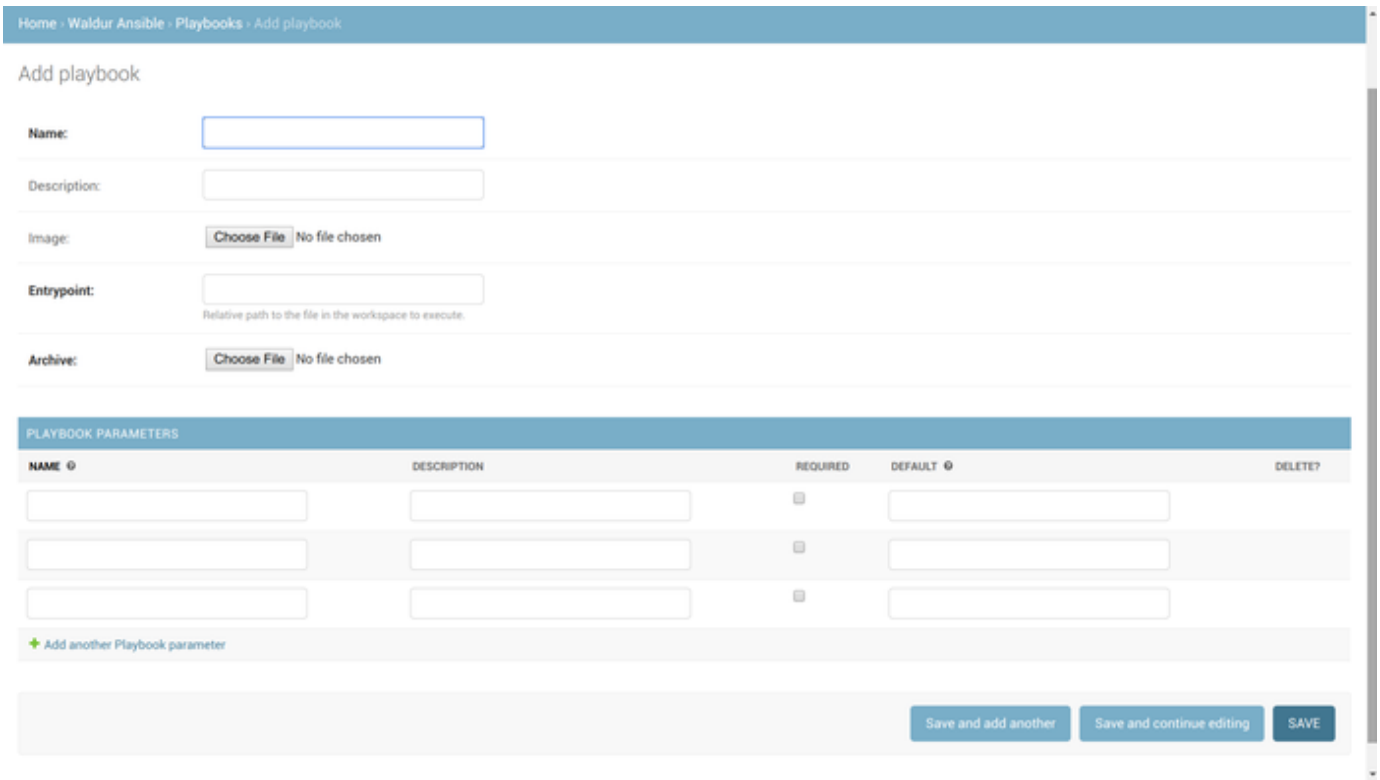
Ansible playbook configuration

In order to upload Ansible playbook to VirtEngine-Waldur MasterMind, in top-level dropdown menu you should go to **Applications VirtEngine-Wa oks**.

Then you should click on **Add playbook** button.



Then you should specify name, description, image, playbook, its entry point, and playbook arguments.



Please note that Ansible playbook should be uploaded as ZIP archive which contains YAML file.

There's only one requirement for archive - playbook file name in archive should match with entrypoint field.

When Ansible playbook is executed It receives following parameters:

Name	Description
api_url	Corresponds to WALDUR_ANSIBLE['API_URL'] configuration option.
access_token	Access token for the user which has created application. Note that it is automatically refreshed when playbook is executed.
project_uuid	Project UUID taken from service project link parameter required by REST API.
provider_uuid	OpenStack Tenant provider UUID taken from service project link parameter required by REST API.
private_key_path	Corresponds to WALDUR_ANSIBLE['PRIVATE_KEY_PATH'] configuration option.
public_key_uuid	Corresponds to WALDUR_ANSIBLE['PUBLIC_KEY_UUID'] configuration option.
user_key_uuid	UUID of SSH key specified by user in application provision form.
subnet_uuid	UUID of OpenStack subnet. Please note that by default first subnet is taken. Also VirtEngine-Waldur MasterMind server validates that selected OpenStack Tenant provider has at least one subnet.

All other extra parameters specified by staff via admin site are passed to Ansible playbook as well.

Example Ansible playbook

Here's an example Ansible playbook which allows you to provision PostgreSQL database server:

```
- hosts: localhost
  connection: local
  tasks:
    - name: Create a CentOS server
      register: database_server
      waldur_os_add_instance:
        api_url: "{{ api_url }}"
        access_token: "{{ access_token }}"
        ssh_key: "{{ public_key_uuid }}"
        project: "{{ project_uuid }}"
        provider: "{{ provider_uuid }}"
        security_groups: ['default', 'ping', 'ssh']
        name: VirtEngine-Waldur
        image: CentOS
        flavor: m1.little
        subnet: "{{ subnet_uuid }}"
        floating_ip: auto
        system_volume_size: 30
        data_volume_size: 1
        wait: true

    - name: Get facts about the server (including its public v4 IP
address)
      set_fact: public_v4="{{ database_server.instance.external_ips.0
}}"

    - name: Add the server to our Ansible inventory
      add_host:
        hostname: "{{ public_v4 }}"
```

```
        groups: sql
        ansible_ssh_user: centos
        ansible_ssh_private_key_file: "{{ private_key_path }}"

    - name: Wait for SSH ping
      wait_for:
        host: "{{ public_v4 }}"
        port: 22
        delay: 10

    - name: Get key data
      uri:
        url: "{{ api_url }}keys/{{ user_key_uuid }}/"
        headers:
          Authorization: "Token {{ access_token }}"
      register: key_response

    - name: Get public key
      set_fact: public_key="{{ key_response.json.public_key }}"


- hosts: sql
  become: True

  tasks:

    - name: Update authorized keys
      authorized_key:
        user: centos
        state: present
        key: "{{ hostvars.localhost.public_key }}"

    - name: Enable PostgreSQL 9.5 repository
      yum:
name=https://download.postgresql.org/pub/repos/yum/9.5/redhat/rhel-7-x86
_64/pgdg-centos95-9.5-2.noarch.rpm state=present

    - name: Install PostgreSQL server
      yum: name={{ item }}
      with_items:
        - postgresql95-server
        - python-psycopg2

    - name: Set up PostgreSQL server
      command: /usr/pgsql-9.5/bin/postgresql95-setup initdb
        creates=/var/lib/pgsql/9.5/data/pg_hba.conf

    - name: Enable and start PostgreSQL server
      service: name=postgresql-9.5 state=started enabled=yes
```

```
    - name: Create databases
      postgresql_db: name={{ database_name }} encoding=utf8
      become: yes
      become_user: postgres

    - name: Create database users
      postgresql_user: name={{ database_user }}
```

```
        become: yes
        become_user: postgres
```

Expert request configuration

Expert request is a custom offering which is designed to create a request for IT services from other organizations.

Name	Description
REQUEST_PROJECT_LINK_TEMPLATE	URL template for expert request details in the project workspace. It is used in the email message. Default value is https://www.example.com/#/projects/{project_uuid}/experts/{request_uuid}/
REQUEST_CUSTOMER_LINK_TEMPLATE	URL template for expert request details in the organization workspace. It is used in the email message. Default value is https://www.example.com/#/organizations/{customer_uuid}/experts/{request_uuid}/
CURRENCY_NAME	Currency name is used to render planned budget in the email message. Default value is EUR
SITE_NAME	Site name is used in the email message. Default value is Example VirtEngine-Waldur Site

Contract configuration

Expert offerings configuration

Expert requests items are configured under **CONTACT offerings** configuration section using the same rules offerings configured, more information on offering configuration can be found in the Request-based offerings section.

Contract form configuration

To extend expert request with more contract details a 'CONTRACT' section has to be configured in the WALDUR_EXPERT settings section. See an example below.

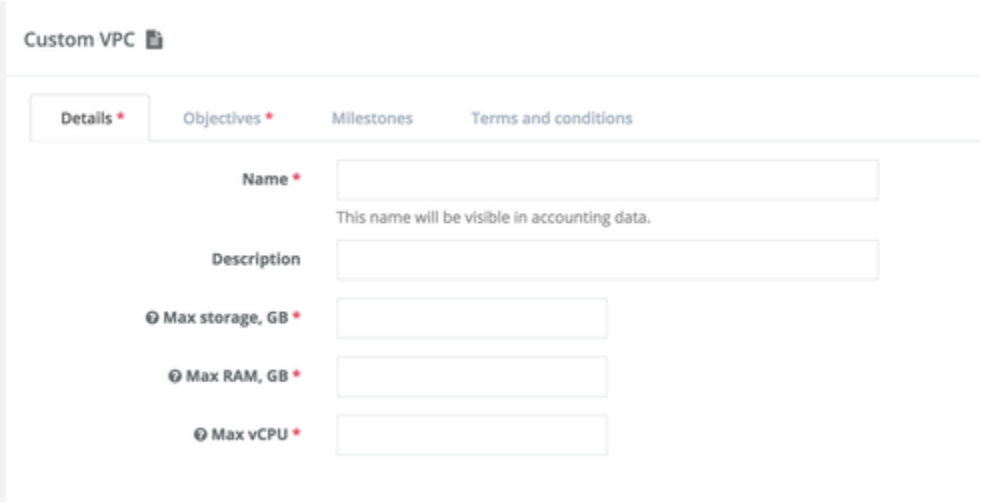
```
'CONTRACT': {
    'offerings': {  # a section can be empty, but it must be present
under CONTRACT config.
        'expert_request_offering_type': {
            'label': 'Website landing page',
            'category': 'Experts',
            'description': 'Descriptive text explaining in 1-2 sentences
what service is about.',
            'recurring_billing': False, # False if expert request
billing is project based, True if monthly occurring.
            'summary': 'A <s>plain</s> text or <b>HTML</b> content to be
rendered under the title in the request-based item form creation.',
            'article_code': '',  # article code is used for encoding
product category in accounting software
            'product_code': '',  # technical code used by accounting
software
            'icon': 'fa-lock',   # awesome icon to use
```

```
            'image': 'https://icons-storage-example.com',  # a full url
of the image to be used instead of the icon.
            'order': ['field_name2', 'field_name1'],
            'options': {
                'field_name1': {
                    'type': 'integer',
                    'label': 'Number of legacy applications',
                    'default_value': 1,
                    'required': True,  # if field must be provided by a
user.
                 },
                 'field_name2': {
                    'type': 'string',
                    'label': 'Estimated deadline',
                    'help_text': 'Free form description when the project
needs to be done'
                 },
            },
            'options_overrides': {
                'objectives': {
                    'default_value': 'The primary goal of the site is to
attract visitors to the site and to get '
                                     'service sales and bookings from
them.'
                },
                'price': {
                    'default_value': 999,
                },
                'milestones': {
                    'default_value': '<ul>'
                                     '<li>Content strategy</li>'
                                     '<li>Design</li>'
                                     '<li>Technical implementation</li>'
                                     '<li>Content in English</li>'
                                     '</ul>',
                },
                'out_of_scope': {
                    'default_value': '<ul>'
                                     '<li>Email newsletter
template</li>'
                                     '<li>Newsletter integration</li>'
                                     '<li>SEO</li>'
                                     '</ul>',
                },
            },
        },
    },
    'order': ['objectives', 'milestones', 'terms-and-conditions'],  #
order of contract tabs
    'options': {
```

```
        'objectives': {
            'order': ['objectives', 'price'],  # fields in the tab
            'label': 'Objectives',  # label of the tab
            'options': {
                'objectives': {
                    'type': 'string',
                    'label': 'Objectives',
                    'required': True,
                    'default_value': 'This is an objective',  # default
value of the field.
                },
                'price': {
                    'type': 'integer',
                    'label': 'Planned budget',
                }
            }
        },
        'milestones': {
            'order': ['milestones'],
            'label': 'Milestones',
            'options': {
                'milestones': {
                    'type': 'html_text',  # html_text stands for rich
html editor which stores html as a plain text.
                    'label': 'Milestones',
                    'help_text': 'Defines project milestones',
                }
            }
        },
        'terms-and-conditions': {
            'order': ['contract_methodology', 'out_of_scope',
'common_tos'],
            'label': 'Terms and conditions',
            'options': {
                'contract_methodology': {
                    'type': 'string',
                    'label': 'Contract methodology',
                },
                'out_of_scope': {
                    'type': 'string',
                    'label': 'Out of scope',
                },
                'common_tos': {
                    'type': 'string',
                    'label': 'Common Terms of Services',
                }
            }
```

```
            },
        }
    }
```

An example of a contract form configuration:



Billing type configuration

Expert requests are automatically included into the monthly invoice and have 2 billing options: monthly recurring payments and project based payment.

In order to specify billing type option set 'recurring_billing' option to **True** in the offering configuration section.

Custom templates configuration

To overwrite default templates you should put custom templates in **/etc/waldur/templates** directory.

In order to customize email message for new expert contract you should create the following files:

- /etc/waldur/templates/experts/contract_subject.txt - template for the email subject
- /etc/waldur/templates/experts/contract_message.txt - template for email body as text
- /etc/waldur/templates/experts/contract_message.html - template for email body as HTML

In order to customize email message for new expert request you should create the following files:

- /etc/waldur/templates/experts/new_request_subject.txt - template for the email subject
- /etc/waldur/templates/experts/new_request_message.txt - template for email body as text
- /etc/waldur/templates/experts/new_request_message.html - template for email body as HTML

In order to customize look & feel of the HTML-formatted message you should create the following file:

- /etc/waldur/templates/experts/style.css

Note that default templates are located in **/usr/lib/python2.7/site-packages/waldur_mastermind/experts/templates/experts/** directory.

Note that all these files are optional. Therefore, you don't have to copy template for the email message if you only need to customize it subject.

Each of these templates receives access to the context data:

- request is expert request object, it has name, state, type, milestones, objectives, etc.
- customer_name refers to the name of the organization, which made this request
- project_name refers to the name of the project, which made this request
- currency_name is the copy of WALDUR_EXPERTS['CURRENCY_NAME'] setting
- site_name is the copy of WALDUR_EXPERTS['SITE_NAME']

Request-based offerings

Introduction

VirtEngine-Waldur supports configuration of offerings, which lead to creation of a new service request in Helpdesk for manual processing. Typically needed for setting up and exposing over API offerings that cannot be automated or require offline negotiations.

Request-based offerings are disabled by default. In order to enable it, you should modify configuration variable as in the following example:

```
WALDUR_SUPPORT['ENABLED'] = True
```

Configuration of offerings is done in VirtEngine-Waldur MasterMind configuration file, under **OFFERINGS** section of **WALDUR_SUPPORT** configu

High-level idea

Provisioning an offering of the request-based type, will lead to creation of offering request model and a ticket in the Helpdesk backend with a summary: **'Request for '{Offering_label}'**. 'Offering_label' is going to be taken from configuration of the offering, in particular from 'offering_typelabel' section. If label is not present a name of the section will be used.

OFFERINGS configuration structure

As an example, let's configure an offering for Digital Transformation of existing applications into cloud-native ones. A block under **'OFFERINGS'** s ection would need to be added to the **WALDUR_SUPPORT** section:

```
WALDUR_SUPPORT = {
...
'DEFAULT_OFFERING_TYPE': 'Service Request',
'OFFERINGS': {
 'offering_type': {
     'label': 'Digital transformation',
        'prefill_name': True,  # If True, name would be pre-filled with
label in request creation form
        'description': 'Descriptive text explaining in 1-2 sentences
what service is about',
        'category': 'Security',
   'price': 1024, # price per month in EUR
        'summary': 'A <s>plain</s> text or <b>HTML</b> content to be
rendered under the title in the request-based item form creation.',
  'article_code': '',  # article code is used for encoding product
category in accounting software
        'product_code': '',  # technical code used by accounting
software
        'icon': 'fa-lock',   # awesome icon to use
        'image': 'https://icons-storage-example.com',  # a full url of
the image to be used instead of the icon.
     'order': ['field_name2', 'field_name1'],
     'options': {
          'field_name1': {
             'type': 'integer',
             'label': 'Number of legacy applications',
             'default': 10,
               'min': 1,
               'max': 20,
               'required': True,  # if field must be provided by a
user.
           },
           'field_name2': {
              'type': 'string',
              'label': 'Estimated deadline'
              'help_text': 'Free form description when the project needs
to be done'
           },
      }
 },
...
}
```

Please note that article code and product code are optional fields. Maximum length of each field is 30 characters. If these fields are defined, they are copied from configuration to newly created offering.

Each field configured in 'options' must be present in the 'order' set otherwise an appropriate error will be raised.

There are 4 pre-defined fields: project, name, description and type. Specifying any of them in configuration may lead to unexpected behavior.

- Project field has to be referenced in a request as a URL link.
- Name allows to redefine label of the offering to be displayed in Service Store.
- Description will be added to the end of the issue description if provided.
- Type which defines an offering type and a configuration section to read from.

Setup of customer configuration options

Type

- integer - number, additional settings include
  - min - minimal accepted value
  - max - maximal accepted value
- string - a string with limitation of 255 symbols.
- text - a string with no length limitations. Considered to be default one and can be omitted.
- html_text - a rich html editor which saves html in a plain text.
- money - an integer which displays default currency and must be greater than or equals to 0. Perfectly fits for price fields.
- boolean
- select_openstack_tenant - allows to select one of the OpenStack.Tenants created in the same project. Passes back name and UUID.
- select_string - allows to select one of pre-defined values. See also example configuration:

Example usage of select string field

Default

Default value to use if field has not been provided in request.

Label

Issue description. Considering configuration below a description of the issue will have the next format:

```
'Name': 'IT Transformation'
'Number of legacy applications' : '10'
'Estimated deadline': 'Deadline in May'
'Description': 'I would like to transform my applications into more
cloud-enabled'
```

Please notice that 'Description' field is always placed on the bottom of the issue description.

Description

Extended description of the service offering displayed in Service Store selector.

Category

A category name of the service. Used for grouping of services in Service Store.

Help Text

A description of a field that will be displayed in the form near a corresponding field.

Example of creating a new request-based offering

For an offering configured as described above, the following API request would be accepted:

Sending POST request to the URL: https://waldur.example.com/api/offering/offering-name-used-in-URL/ with a body payload of:

```
{
'field_name1': '10',
'field_name2': 'Deadline in May',
'project':
'https://waldur.example.com/api/projects/00cf40c2-9332-42c4-bad3-4fd965f
f21dc',
'name': 'IT Transformation',
'description': 'I would like to transform my applications into more
cloud-enabled.',
'type': 'offering_type',
}
```

Issue type configuration

It is possible to configure a type of the issue by editing 'DEFAULT_OFFERING_TYPE' configuration. For instance, default type is configured to 'Service Request' as shown below:

```
WALDUR_SUPPORT = {
...
'DEFAULT_OFFERING_TYPE': 'Service Request',
'OFFERINGS': { ... },
...
}
```

Default offering type is used as an issue type in a configured Helpdesk tool.

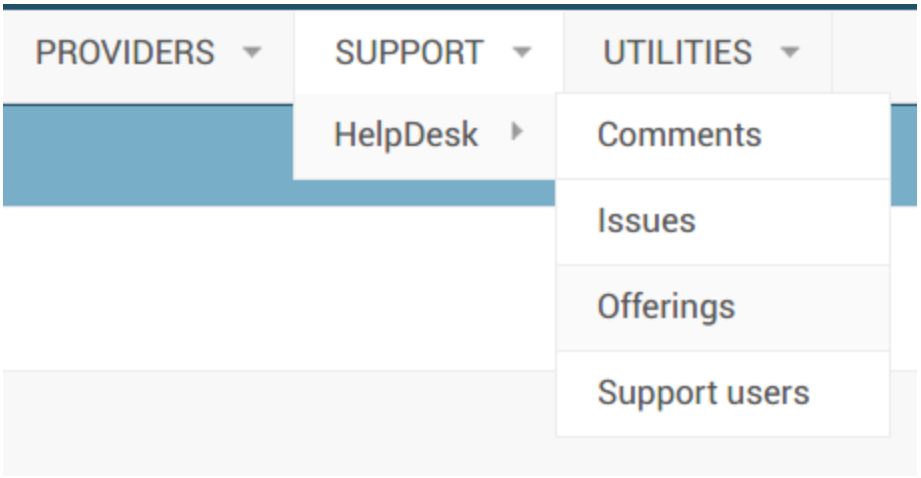
Accounting

A total price for a particular offering is calculated following unit pricing model (see below) and applicatble to the time when offering was in OK status, i.e. available for the end-user for consumption.

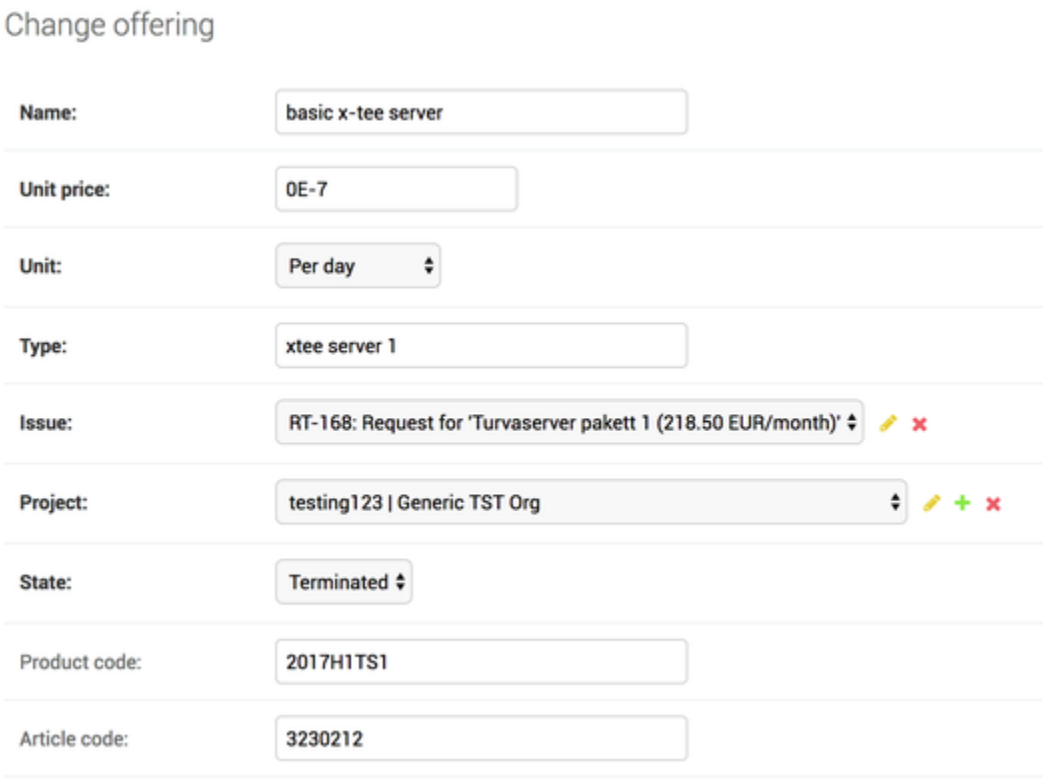
Lifecycle of a typical offering lifecycle is as follows:

- Offering types are configured by operator of VirtEngine-Waldur MasterMind.
- User selects a service offering from a Service Store and submits a request. Request is created with a status of "Requested".
- Operations fulfill the request.
- A staff user defines pricing model used, e.g. "price per day", "per half month", "per month".
- A staff user defines price of the offering and updates it in the administrative panel of MasterMind and sets offering state to OK.
- Accounting starts from the moment of setting the state to OK.
- Accounting ends with a staff user setting offering state to Terminated.

Created offerings can be browsed in **Support HelpDesk Offerings** menu.



Price can be updated by clicking on required offering and setting price in the form that shows up.



Cookie policy

VirtEngine-Waldur can save data about the user in a browser. The following are cases when data is stored in HomePort:

1. Saving of the latest view point of the user.
2. Saving redirect information, i.e. where to forward from a certain state.
3. Saving which authentication method was used by user for logging in.
4. Saving user session token.
5. Saving user language preference.
6. To see latest state about all cases in the code where saving of data is done, please visit: https://github.com/virtengine/waldur-homeport/search?q=localstorage&unscoped_q=localstorage

For MasterMind cases are:

1. Saving authentication token (cookie) when a user logs into /admin management interface.

User Guide

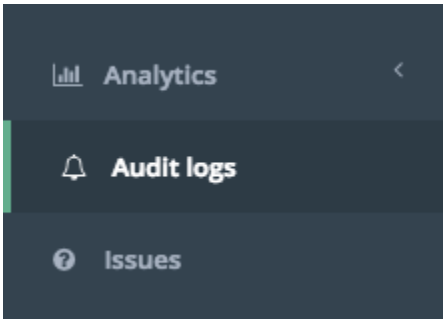
Audit Log

Introduction

Audit logs describe events occurred in organizations, projects and resources. They allow users to track actions performed on any of these entities. For example, project managers can track what users were assigned a role in their project or check what resources were recently created. Audit logs are categorised into three main groups: organization audit logs, project audit logs and resource audit logs. These groups are described in details below.

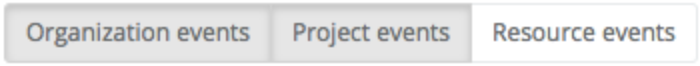
Organization audit logs

Organization audit logs can be accessed by its owners, and they provide information about events occurred within the organization. The organization owners can see any events occurred within the organization, including audit logs for projects and resources. For example, organization owner can see what projects were created in the organization. To access organization audit logs, user must have owner role in the organization, select organization workspace (see HomePort Workspaces) and click on `Audit logs` button in the navigation panel:



The resulting page allows to perform following actions:

- Including/excluding groups of events in the list by clicking on these buttons:

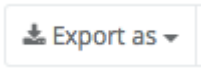
- Regulating number of log entries per page:

- Search for a specific log entry:

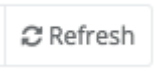
- Export displayed list of log entries in specific format:

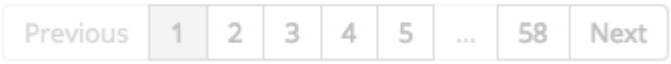
- Check what kind of events system currently supports:

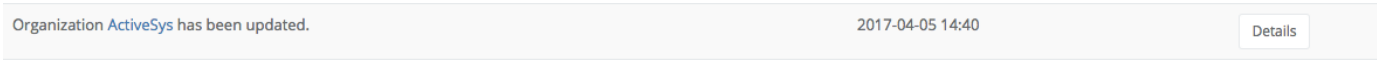
- Refresh the list of log entries:

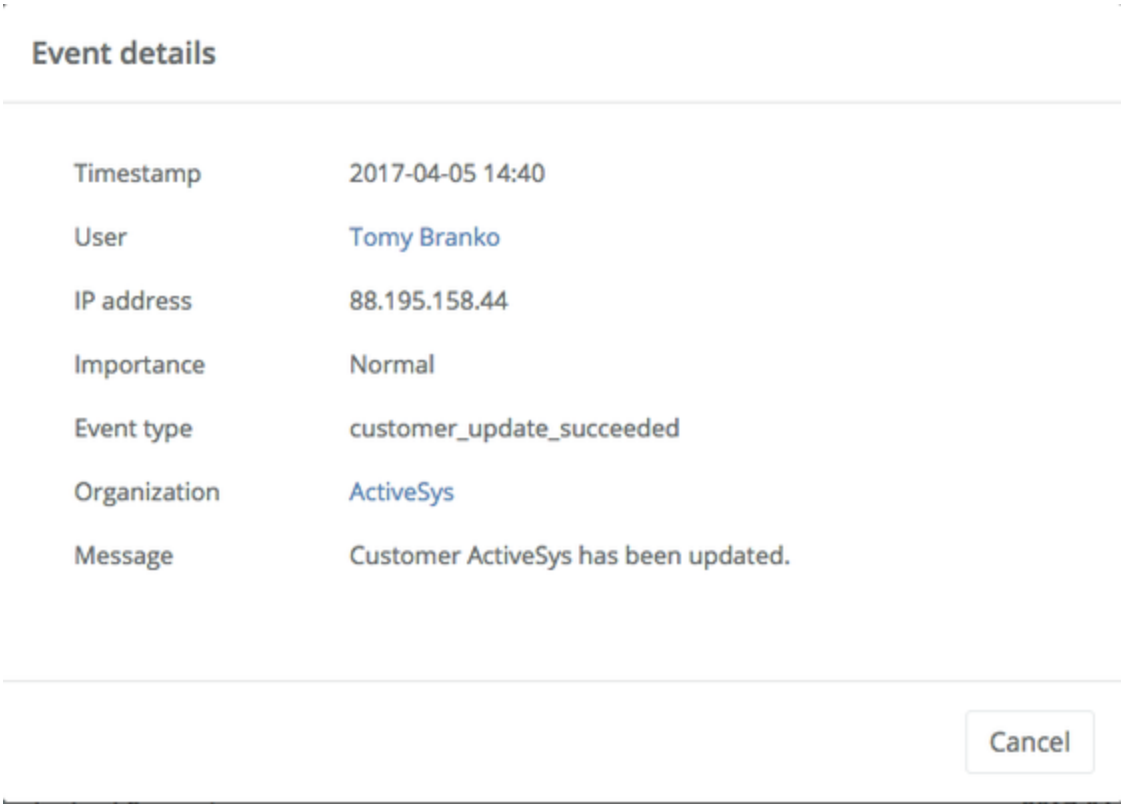
- Navigate through pages:

Each log entry includes message, timestamp when the log entry was created and `Details` button:



By clicking on the `Details` button user can get more information about the log entry:



The log entry details usually contain involved project, organization, resources, the user who triggered the action and other information.

Project audit logs

Project audit logs can be accessed by its managers, administrators and support. They provide information about events occurred within projects. For example, users can check what roles to what users were assigned in the project. The entries occurred in the list are filtered according to the user role in a project. In addition, the user can include/exclude resource event logs in the list. To access project audit logs, user must have role in the project, select project workspace (see HomePort Workspaces) and click on `Audit logs` button in the navigation panel similarly to what was described in "Organization audit logs" section.

Resource audit logs

Resource audit logs can be accessed by any user that has access to the project. They provide information about resource events. For example, users can check what operations were performed on the resource. To access resource audit logs following steps should be performed:

1. Select project workspace (see HomePort Workspaces)
2. Go to `Resources` <Resource type> (e.g. `Virtual machines`) in navigation panel
3. Select resource from the list by clicking on its name
4. Select `Audit log` tab:

Azure provider management

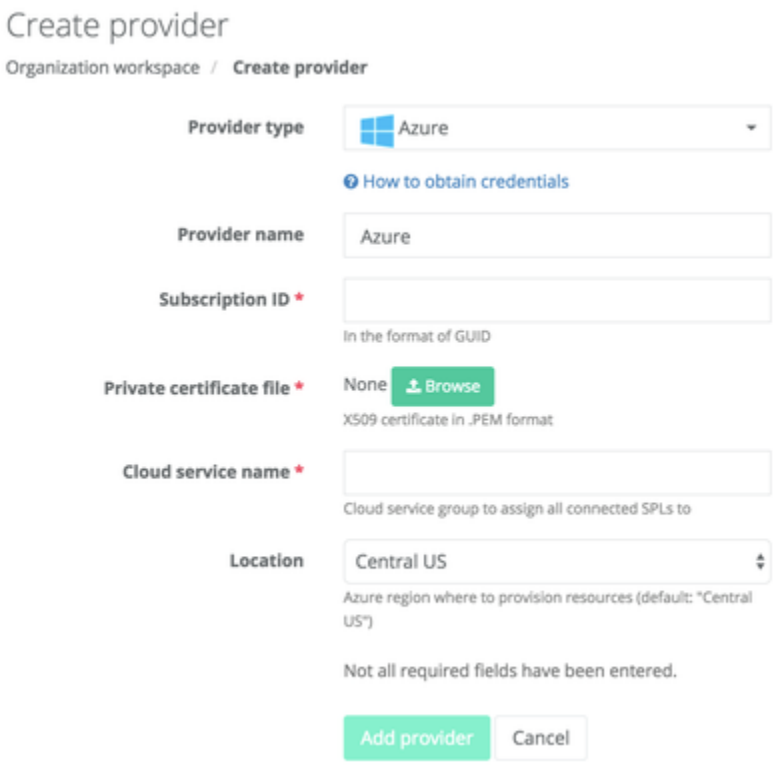
Introduction

An azure provider represents a cloud service and uses a storage account with the same name under the hood. Usage of multiple storage accounts in one cloud service is not supported at the moment. Resources that have to be managed in different accounts have to be created under different cloud services and represented as new Azure providers in VirtEngine-Waldur.

Resource management is focused on Azure Virtual Machine management.

On Azure provider management user is asked to provide (see picture below):

- Azure subscription ID
- A private X.509 certificate in PEM format
- Cloud service name to be used for resource management



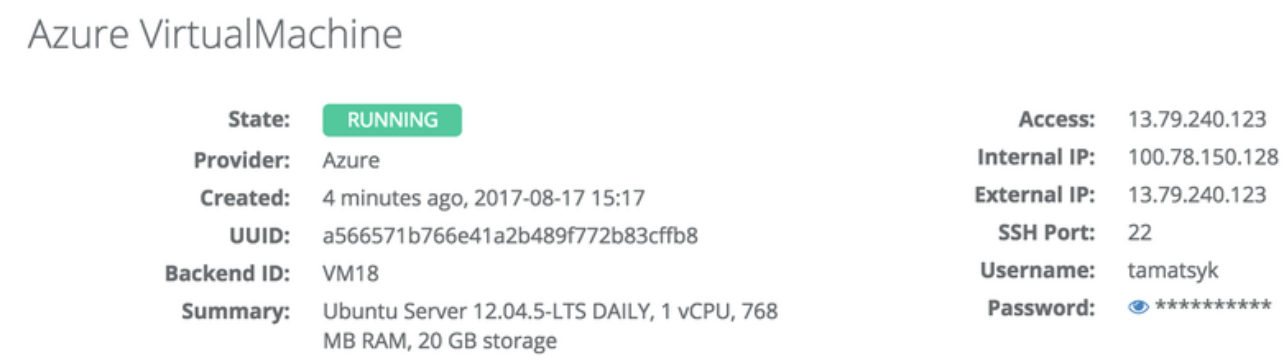
* For more info on how to configure a certificate file please follow "How to obtain credentials" link under provider type selectbox on provider creation page.

RDP access

You can access Windows servers which provider RDP access by downloading a configuration file from virtual machine details view.

SSH access

SSH port is displayed on Virtual Machine details if access is provided by SSH. You can use username and password provided on Virtual Machine creation phase.

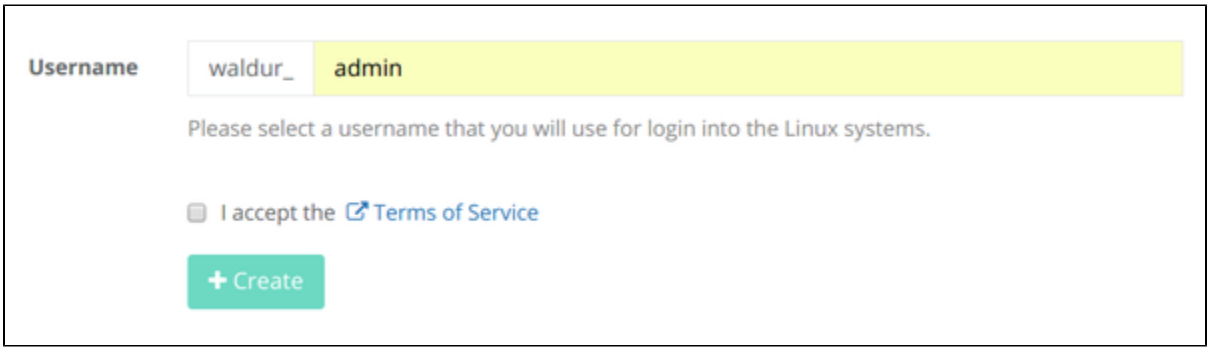


FreeIPA

VirtEngine-Waldur HomePort allows you to manage FreeIPA account for your user and specify quota for organization and project.

Create FreeIPA account

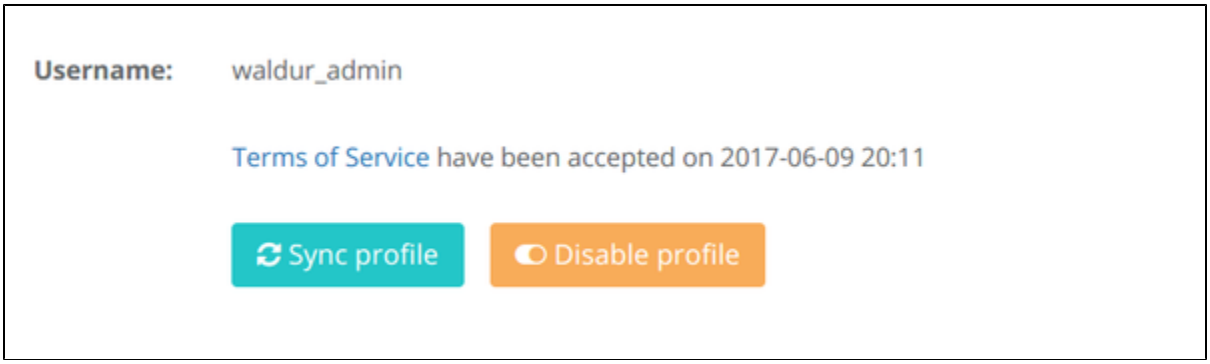
1. In order to create FreeIPA account for your user please switch to user workspace.
2. Then click on **FreeIPA account** link in sidebar menu.
3. Then enter username for your new account. By default your username is used as username for FreeIPA account.
   Note, however, that there are several restrictions for username imposed by the Linux operating system:
   a. Username can contain Latin letters, numbers, dashes, underscores and periods only.
   b. Username contains mandatory prefix.
   c. Maximum username length is 32 characters.
4. Then read and accept Terms of service and click **Create** button.



Manage FreeIPA account

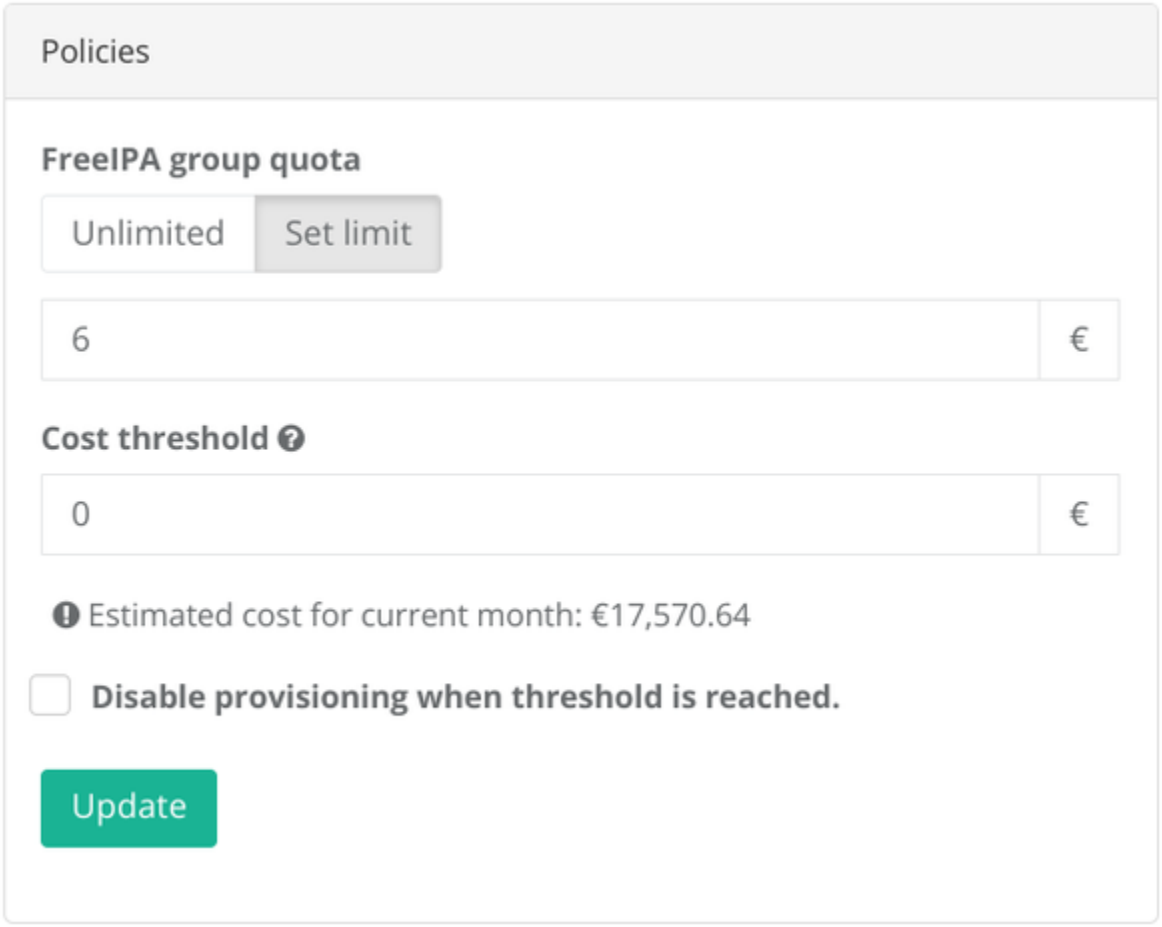
After you have created FreeIPA account you may see your username, date when Terms of service have been accepted.

Also you may enable or disable account, or schedule SSH keys synchronization.



Set FreeIPA batch processing quota for organization and project

1. In order to create FreeIPA account for organization please switch to organization workspace.
2. Then click on **Manage** link in sidebar menu.
3. By default organization doesn't have FreeIPA group quota limit (it quota limit is -1).
4. Then click on **Set limit** button and enter quota value.
5. Then you should click on **Update** button.



Note
FreeIPA group is created automatically by backend processing task whenever organization or project is created, updated or removed.

FreeIPA group user and group membership records are propagated by backend processing task.

Changes for quota value are not applied immediately on backend. In fact, it takes some time, usually 10-15 minutes, in order to propagate quota value.

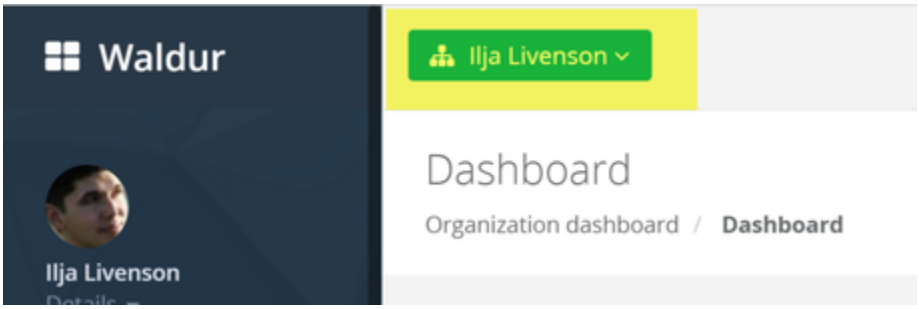
HomePort Workspaces

Introduction

VirtEngine-Waldur HomePort is a self-service aimed to satisfy needs of users in different roles. Roles are connected to a particular structural unit tl a context. For example, Organisation X owner or Project Y system administrator. There are several workspaces available in HomePort described below.

Organization workspace

Organization workspace allows to manage projects, subscriptions to resource providers and teams. It is also intended to provide summary, accounting and auditing information regarding organization, projects and providers. To be able to access Organization workspace, you need to have a role of staff or organization owner. Access is done via a selector in the top section of the user interface.



Project workspace

Project workspace provide tools and information required for day-to-day work and oversight over the managed IT infrastructure. Access is done through the selector on top.

User workspace

User workspace is a personal account management space. Allows to configure notifications, SSH keys, update personal data. Access is done by clicking on user avatar and selecting one of the menus.

Support workspace

Support workspace is dedicated to management of interactions with helpdesk. Support workspace can be enabled by clicking on the "Support" menu in the header row of the HomePort user interface.

OpenStack Instance management

Introduction

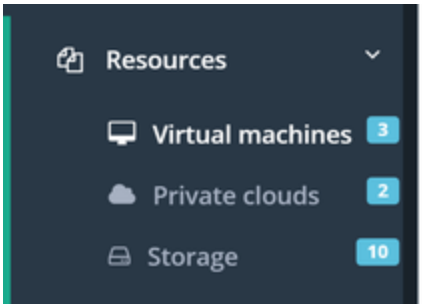
OpenStack Instance is one of the main resources that OpenStack Tenant provider can produce. It corresponds to a classical VM creation in the environment defined by OpenStack Tenant. VirtEngine-Waldur defines a certain model of VM instance creation forcing best practices for instance deployment and management.

Instance creation

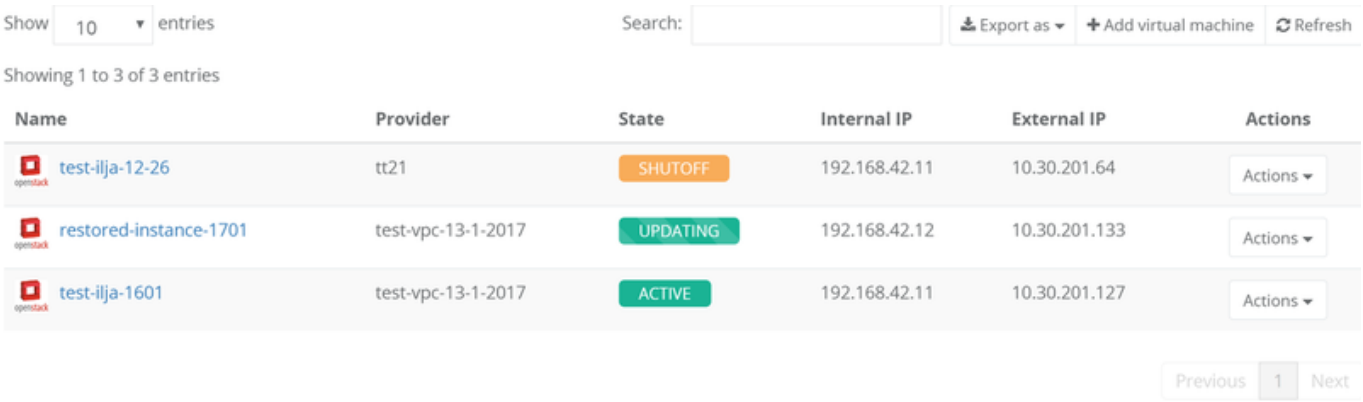
Instance can be created via a Service Store by selecting a "IaaS > Virtual machines" category and picking one of the OpenStack providers. A form will be presented (an example is shown below) that needs to be filled to create an instance. Once the form is submitted, flow for VM creation will start. Please note, that VirtEngine-Waldur is enforcing on creating both System (immutable post-creation) and data volumes. This allows to hav persistent VMs and to perform volume-based backups / DR replication without additional means.

Instance modification

To see details and modify a VM, please select it from the list of VMs, which is accessible from the Project workspace menu:



From the list of VMs you can either perform quick actions using the "Actions" menu, or go into details of a VM and have a more fine-grained set of actions / VM data.

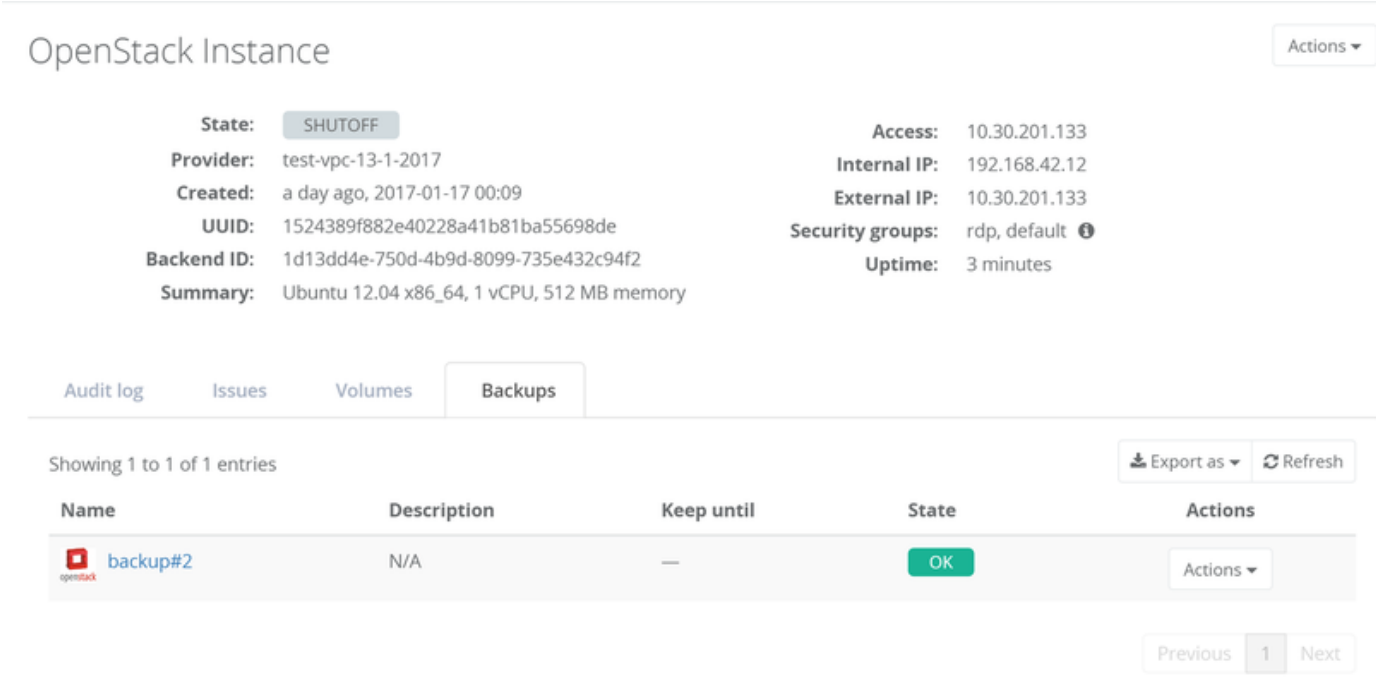


The list of supported actions is dynamic, but the main actions include:

- lifecycle operations (stop/start/restart/deletion)
- Flavor change / resize
- Setting / updating security groups
- Setting / updating floating IPs

Backup management

VirtEngine-Waldur provides a custom way for performing quick backups of volume-based VMs by using snapshotting strategy. To create a backup "Actions" menu of an Instance. You can see existing backups in the backup tab. To restore a backup (creates a new VM), select it from the backup's action menu.



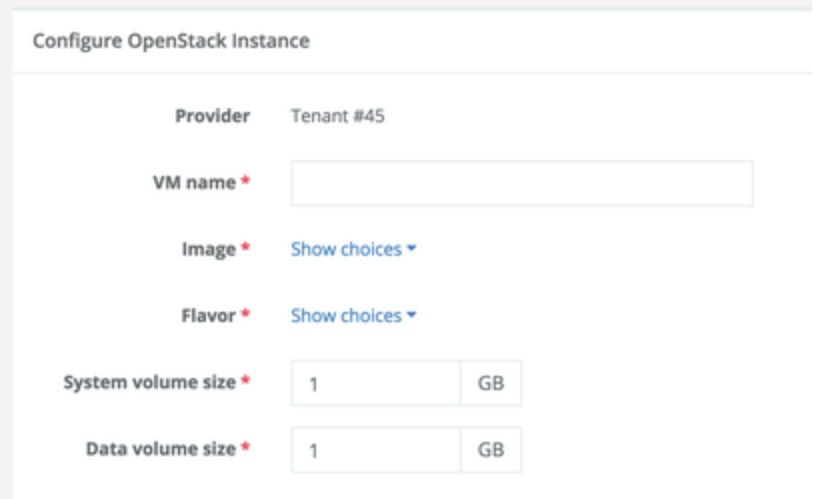
OpenStack Storage Management

Introduction

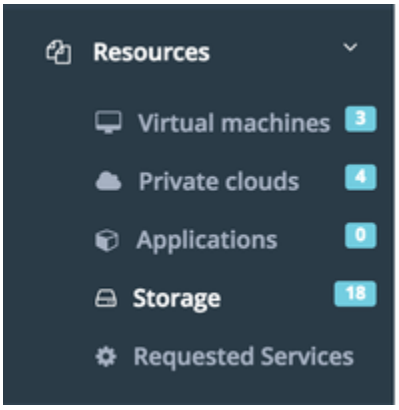
OpenStack Storage is a resource that is automatically produced by managing OpenStack Instances. VirtEngine-Waldur represents OpenStack res Volumes and Snapshots, both of which can be considered as resources stored on a physical device.

Volume creation

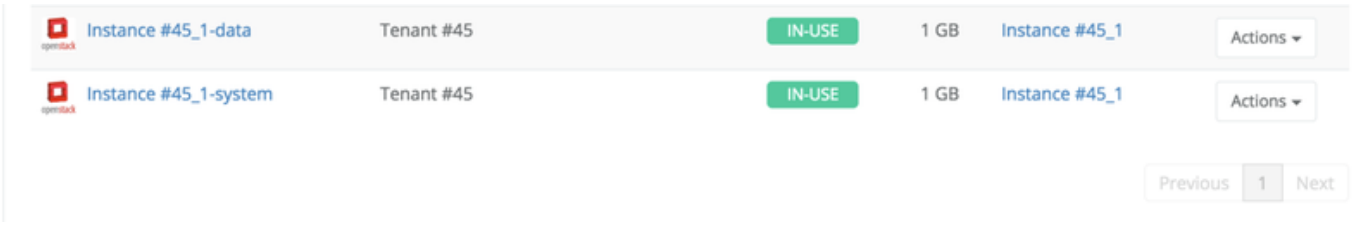
Volume is automatically created by provisioning an OpenStack Instance. VirtEngine-Waldur produces two volumes on instance creation, one for sy another one for data. You can configure a size of each volume at virtual machine creation step by settings **System volume** and **Data volume** siz es. Minimum system volume size depends on the image of desired operation system to be installed.



After virtual machine is provisioned you will be able to find created volumes under **Storage** item menu.



On the list of volumes you can see an instance volumes are attached to, go to volume details by clicking on volume's name or perform quick actions on it.

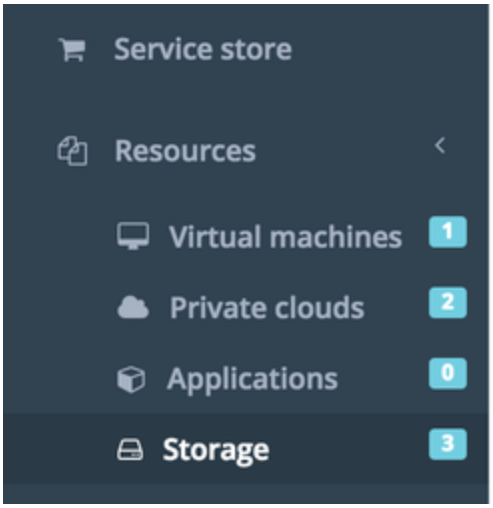


The full list of actions includes:

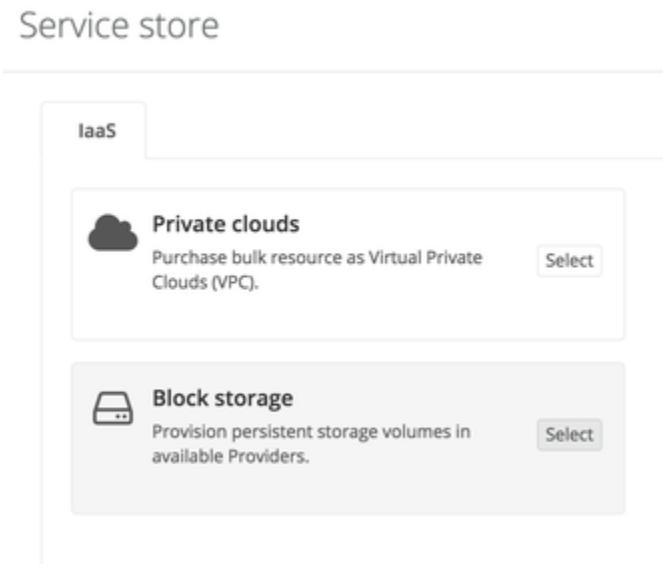
- Attach/Detach volume to/from instance;
- Update name and description;
- Extend volume size;
- Destroy volume;
- TBA

Create a volume

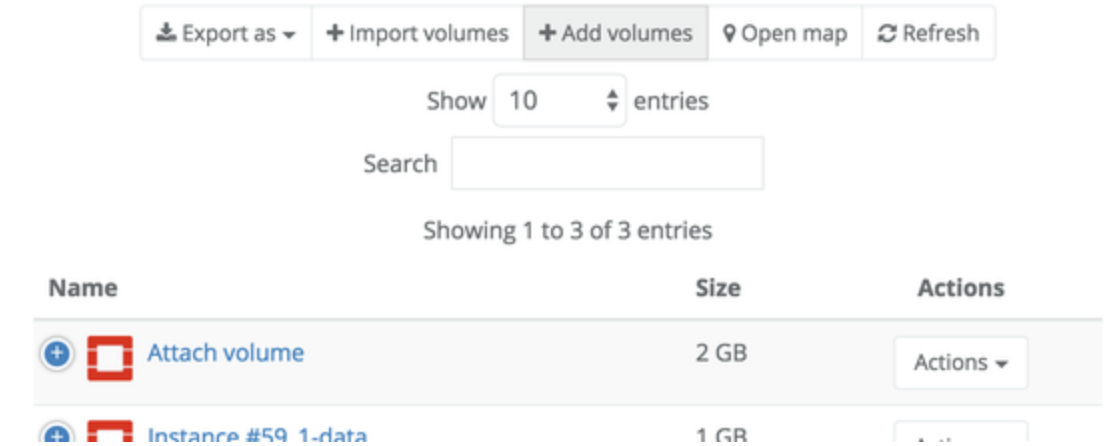
There are 2 ways to order a volume - from a **Service store** or from a **Storage** item menu under **Resource** group.



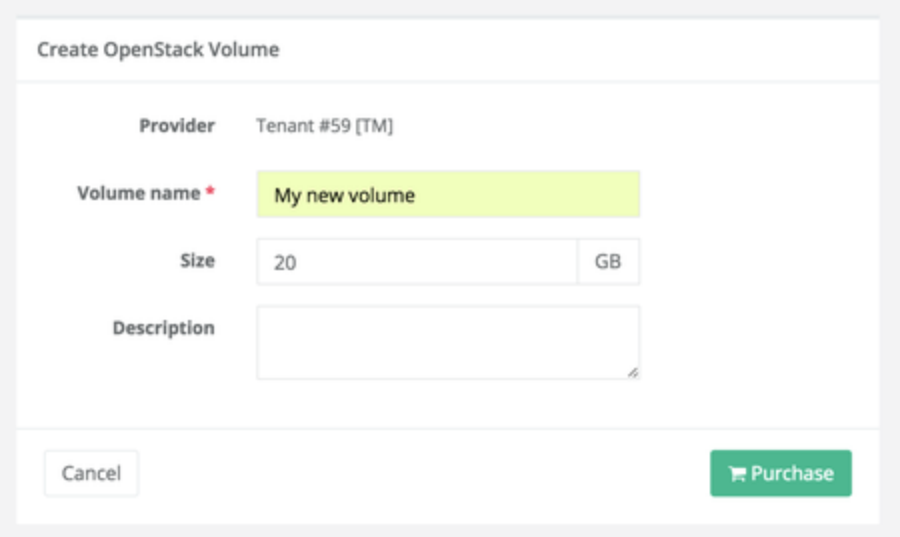
To add a volume for a **Service Store** select a **Block Storage** item.



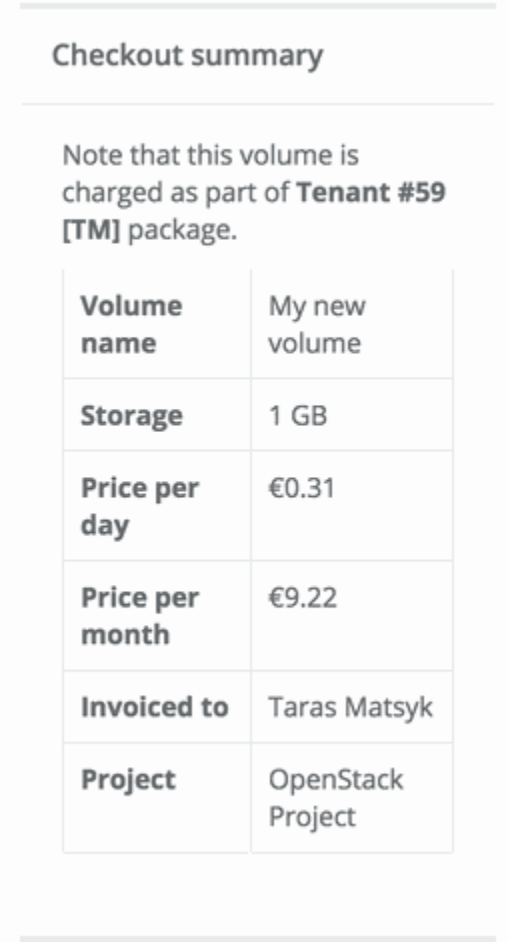
To order a volume from a **Storage** menu by click on the **Add Volumes** button.



Select a volume provider and a volume creation form will be opened. Fill all required fields to order a volume.



When you finished filling the appeared form please check a summary of your order, it displays options of the volume you selected and price of the item. In order to purchase the volume an overall OpenStack Tenant storage should not exceeds its quota.

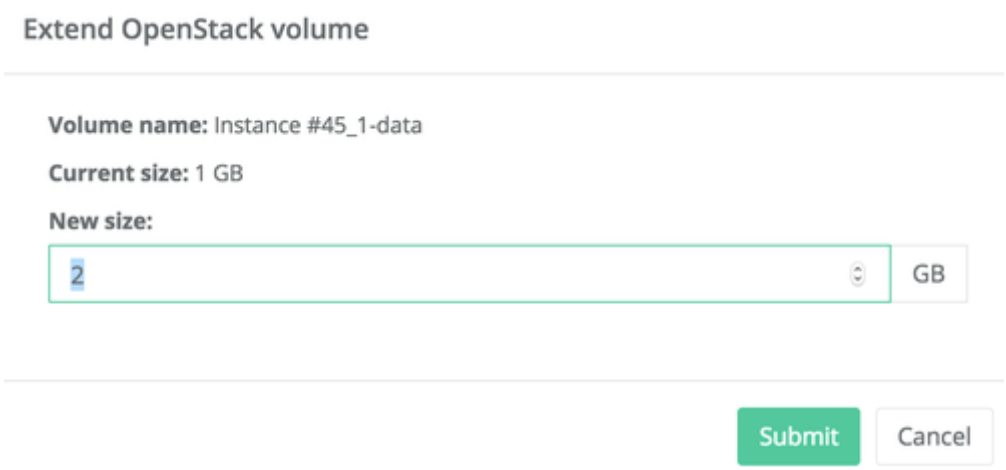


Volume modification

VirtEngine-Waldur simplifies volume management by providing an easy way to extend, attach, detach or destroy managed volumes.

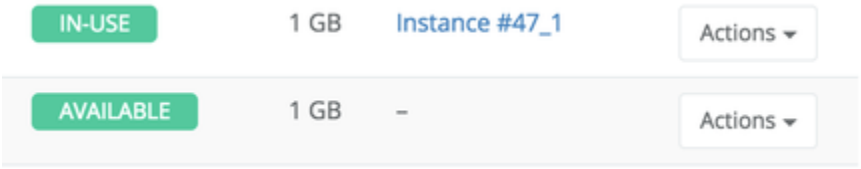
Extend volume

To extend a volume please detach it from the current instance, select a **Detach** action under **Actions** list and put new volume size in **GB** into appeared form.



Detach volume

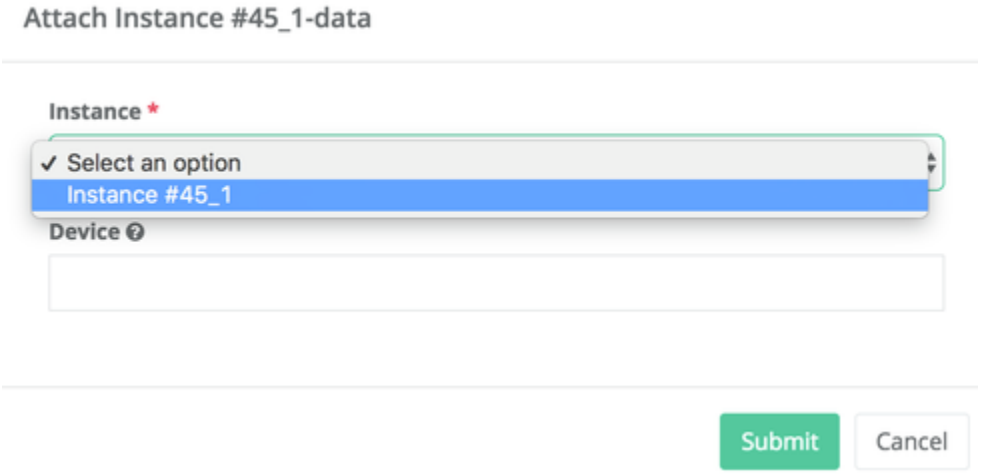
To detach volume select click on **Detach** item under **Action** menu. After volume is detached it will not longer be associated with an instance as shown on the picture below.



> Please keep in mind that it is not possible to detach system volume.

Attach volume

To attach a volume find a volume that is not attached yet and has **Available** state. Select **Attach** action and select available instance to attach the volume to it.



Snapshot creation

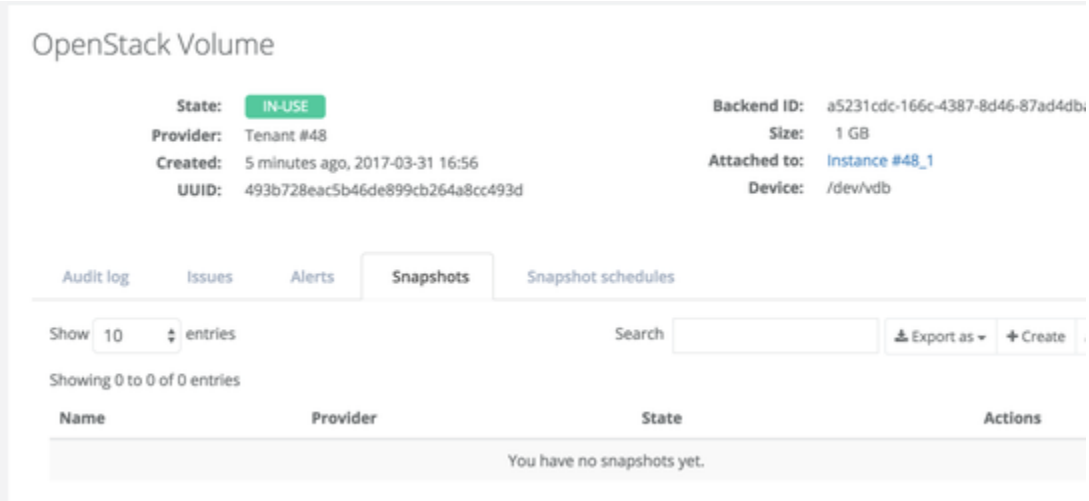
To create a snapshot select a volume to create a snapshot from and open volume details.

There 2 ways to create a snapshot:

- Create a snapshot immediately;
- Create a snapshot on a schedule;

Immediate snapshot creation

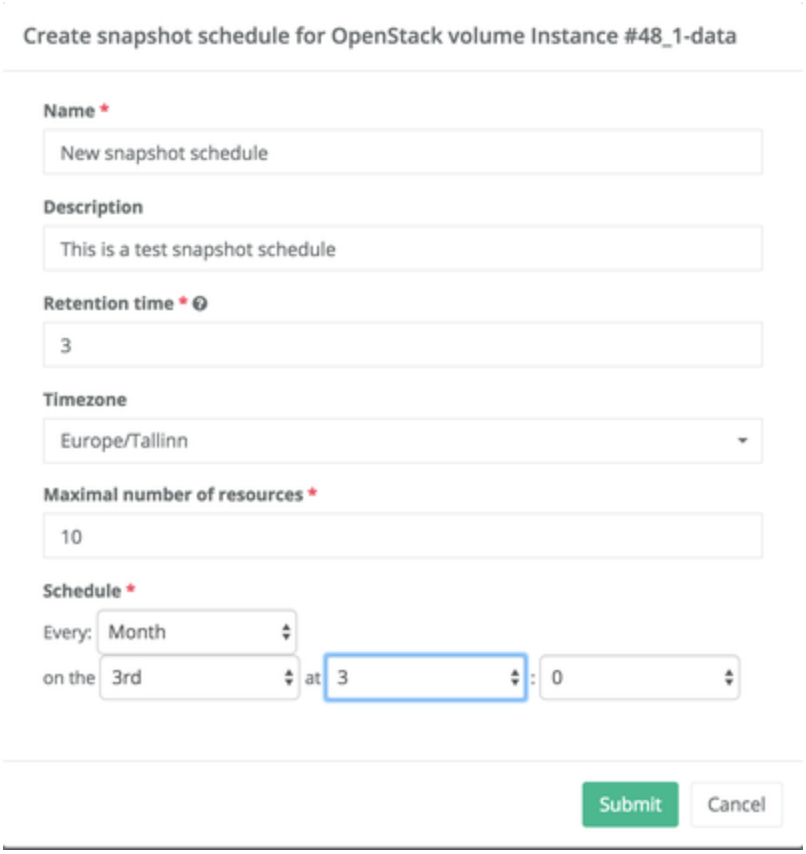
To trigger snapshot creation immediately click on **Create** button under **Snapshots** tab. Click submit button after snapshot name and description are filled.



There 3 main actions that can be performed on a snapshot: deletion, editing and volume restoration. Each of actions are listed under **Actions** dro pdown next to the snapshot.

Snapshot schedule creation

Snapshot creation can also be delayed or created on a regular basis by using a snapshot schedule feature. To schedule a snapshot go to a **Snap shot Schedules** tab and click on **Create** button. Do not worry, it will not trigger a snapshot creation process as each snapshot schedule has to be activated from **Actions** menu on a schedule.



Form parameters are described in the table below.

Parameters Name	Description
Name	Name of the snapshot
Description	Description of the snapshot
Retention time	Time to keep the snapshot in days. Set 0 to keep forever.
Timezone	A snapshot timezone
Maximum number of resources	Maximum number of snapshots to be created, for instance, 10 means, only 10 snapshots are going to be taken.
Schedule	Time to trigger a snapshot creation task

OpenStack Tenant management

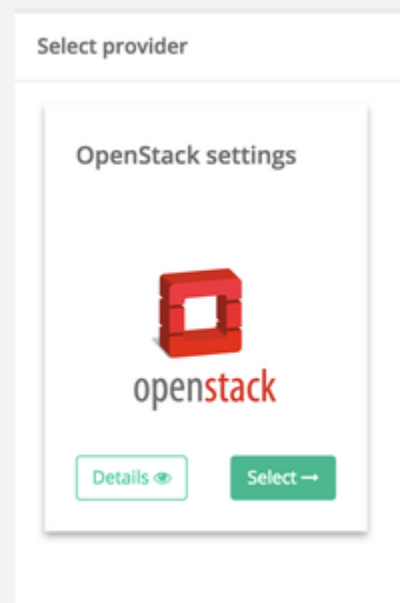
Introduction

OpenStack Tenant is a main resource provided by OpenStack. Tenant is a set of resources like virtual machine, network, storages etc. VirtEngine-represents OpenStack Tenant as a private cloud.

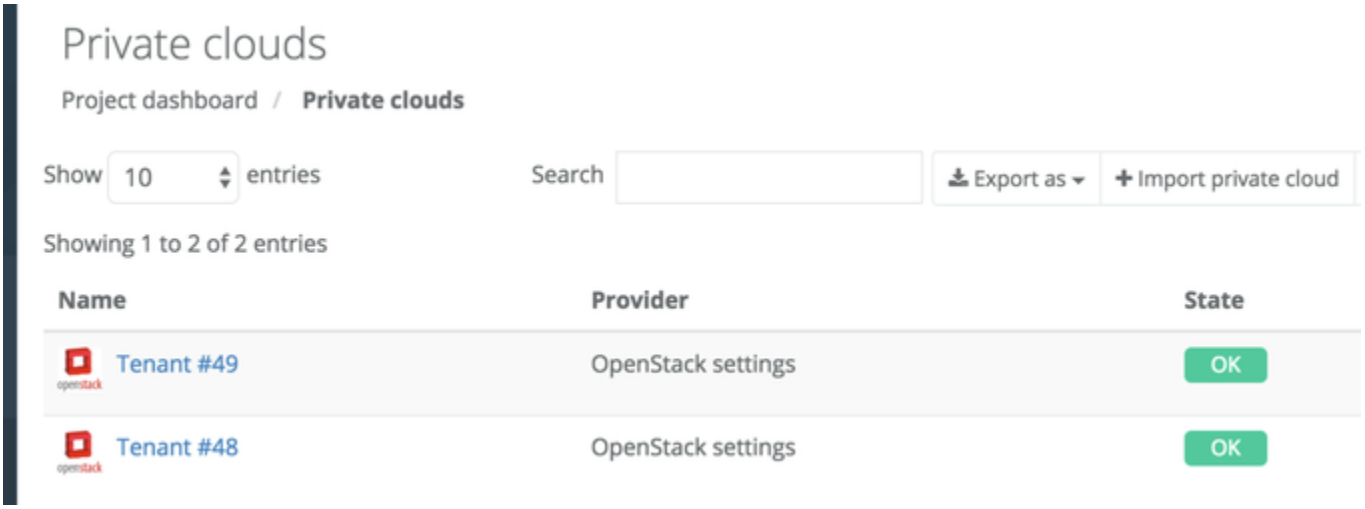
> Please note, that depending on configuration, virtual private cloud management could be unavailable.

Private cloud creation

All private clouds can be found under **Resources** menu item. To create a cloud click on **Add private cloud** button and select a private cloud provider, an OpenStack settings in our case.

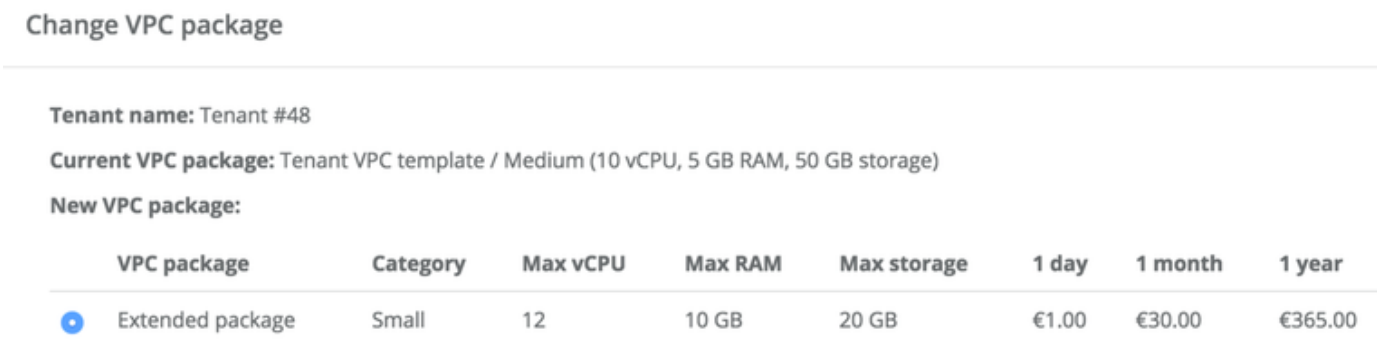


Click **Select** and fill in a private cloud form. Please make sure that you selected a right VPC package, which shows quotas available for a new resource and its price. Hit **Purchase** and wait for cloud creation. After tenant is created you can find in a list of private clouds under **Resources** m enu.



Private cloud modifications

It is possible to change quotas of tenant resources by changing a VPC package. To change the package select **Change VPC package** on the **Ac tions** dropdown. You will be presented with a list of available VPC packages. Select the one you need and click **Submit** button.

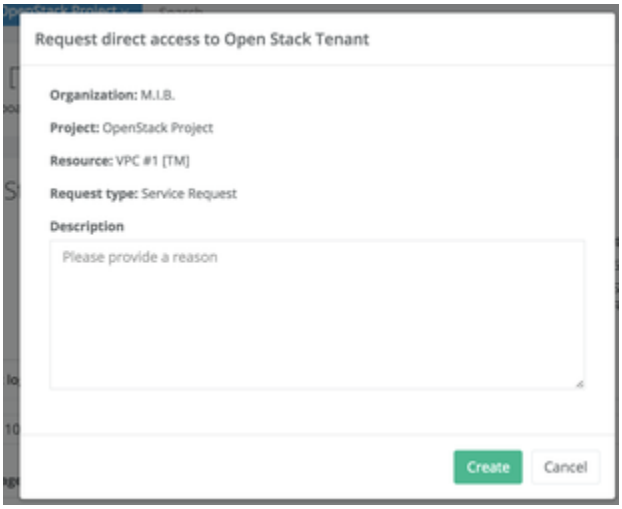


Private cloud deletion

To delete a private cloud simply select and confirm a **Destroy** action on the **Actions** dropdown near the cloud you want to be deleted.

Request direct access

To request tenant credentials select a **Request direct access** action on the **Actions** dropdown and fill in all required fields to create a support ticket with the request.



> Request direct access is only available if tenant credentials are not available due to system configuration.
>
> If request direct access is available and support functionality is not, the only way to get tenant credentials is to send an email with a tenant identification and a reason to support.

Organization management in HomePort

Introduction

In VirtEngine-Waldur organization is a main structural unit. In order to switch to organization workspace please use workspace selector.

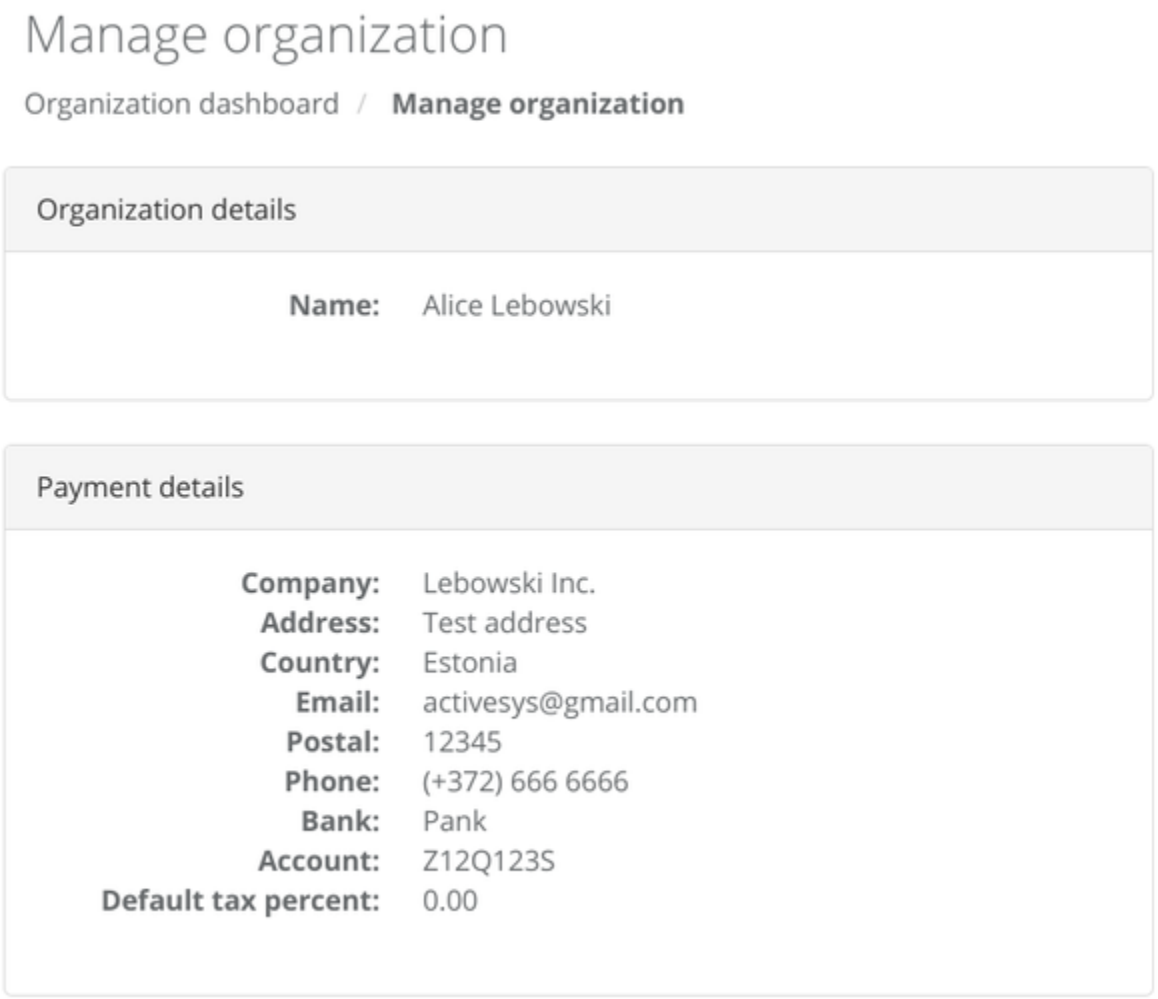
In organization workspace you can manage projects, providers, users, analyze resource usage.

Permissions

Role	Permissions
Global support	Read-only access to any organization.
Organization support	Read-only access to managed organization.
Organization owner	Read-write access to owner organization.
Staff	Read-write access to any organization.

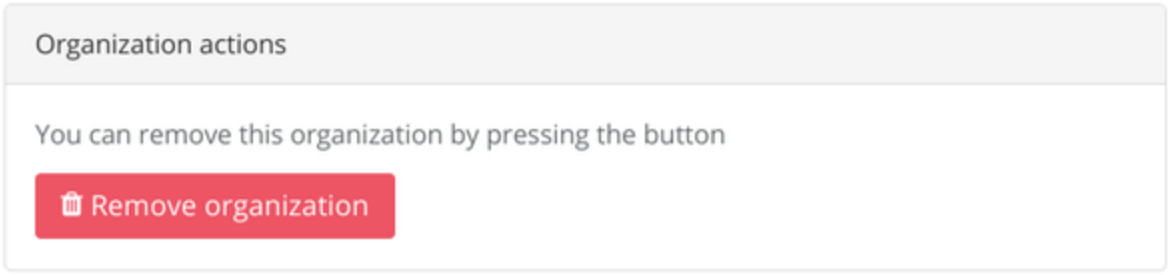
Details

In order to manage organization itself, you should click on **Manage** link in a sidebar of a organization workspace.



Note that payment details may not be present and currently only staff can modify them via admin site.

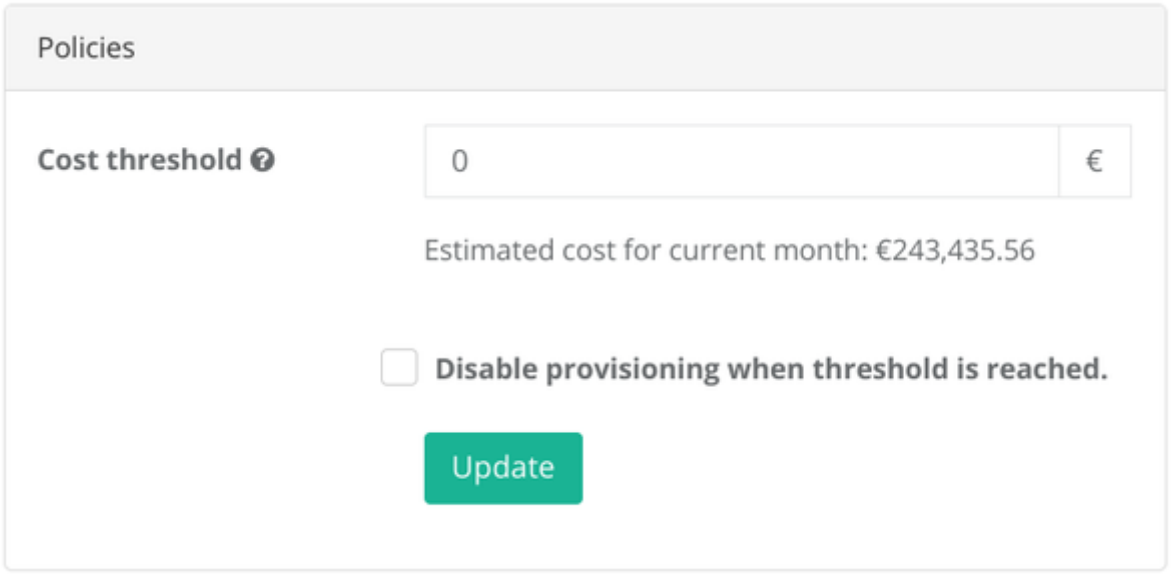
Remove organization



Note that depending on deployment settings, user permissions, this feature may be disabled.

Note that If organization contains projects, they should be removed first.

Manage organization policies



In policies section you may get price estimation for the current month, set cost threshold, and disable provisioning when threshold is reached.

Profile Management

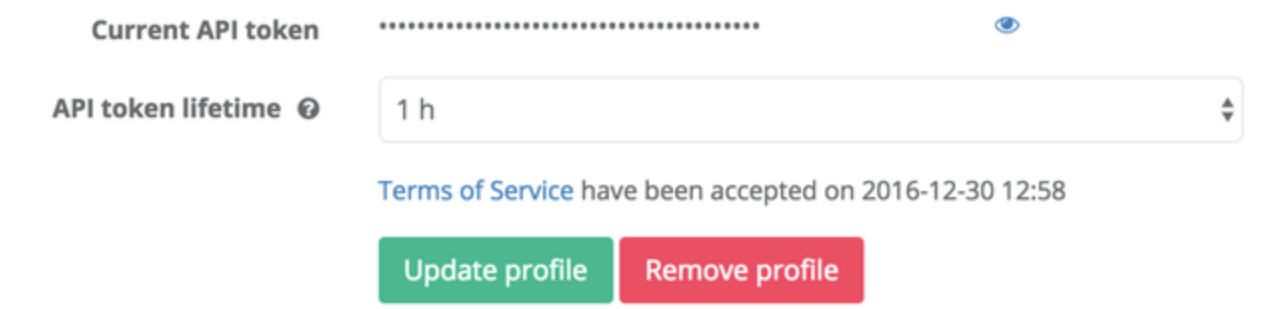
Introduction

VirtEngine-Waldur provides a rich variety of profile management options like event notifications, logs audit, profile and API access management.

Manage user profile

To edit an information about current user or manage its API credentials enter **Manage** section.

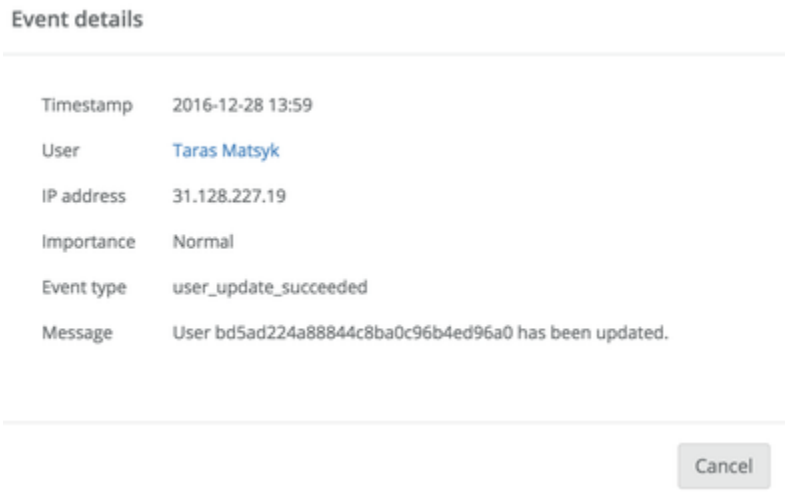
If you use an API to communicate with VirtEngine-Waldur here you can get an access token and set its lifetime.



Audit logs

Audit logs is a place where user can browse, search through logs, export them to various of formats and get an explanation of logs type.

Click **Details** to see an event information.



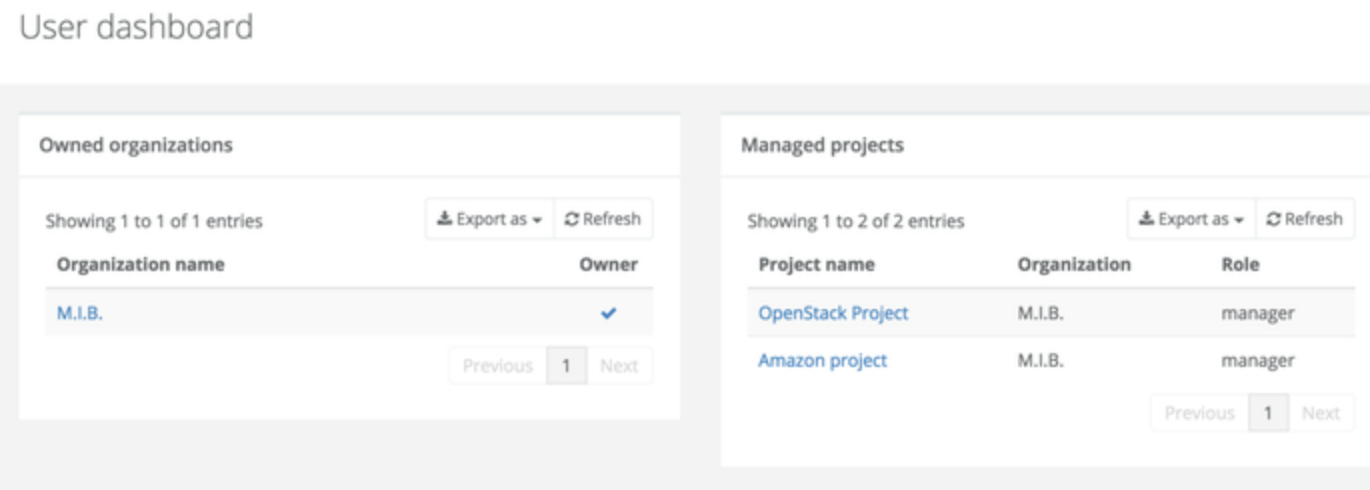
SSH Keys

A menu to manage your SSH keys.

To add a new key click on the **Add SSH key** button and specify an SSH name and a public key.

Dashboard

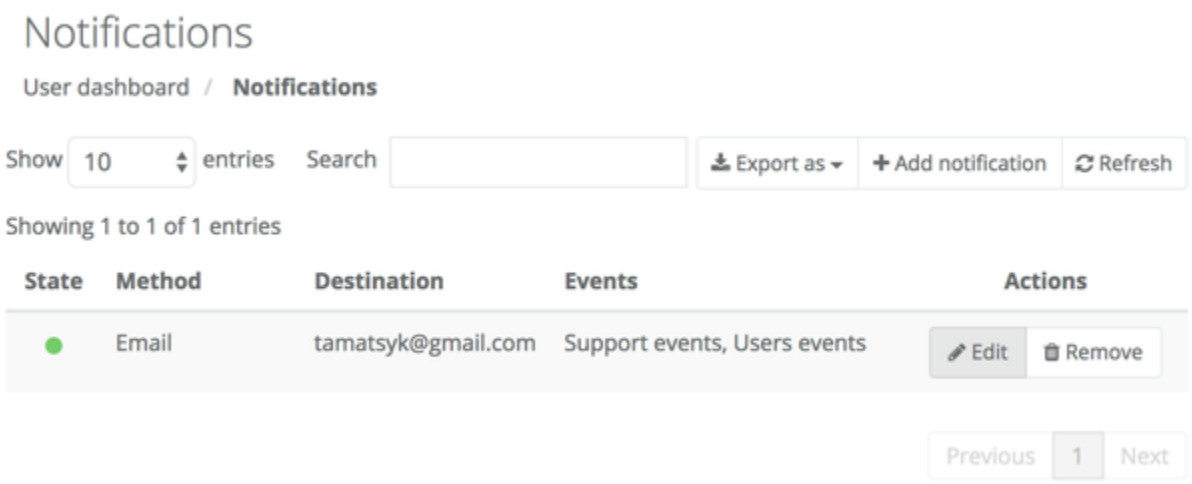
The most used panel on a user management menu. Aggregates organizations, projects managed by user and the most recent logs preview. Each widget supports export to CVS, Excel, PDF and allows to copy or print data from it.



Please note, that this Dashboard is different one from the Dashboard user sees when he selects an organization and a project.

Notifications

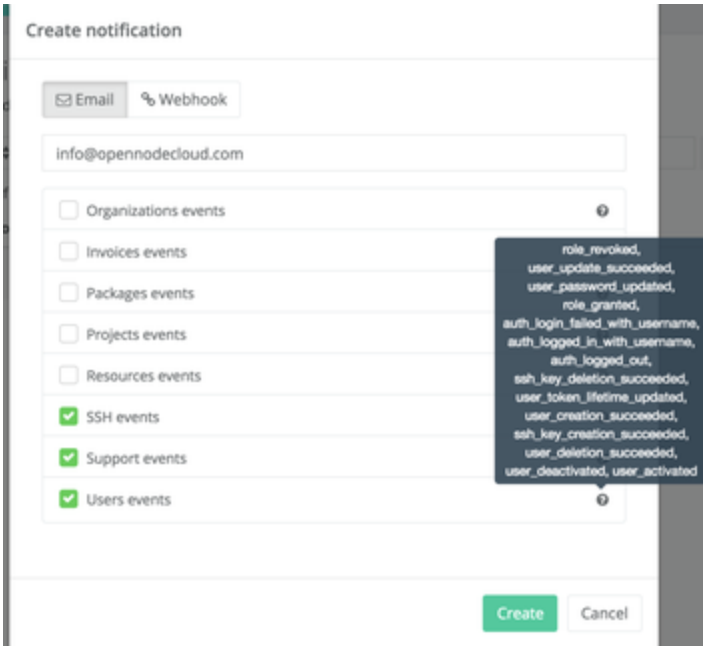
Notification menu is intended for listing of existing notifications or creation of new one. Here a notification can be created, modified or removed. Export of configured notifications is available as usual.



Click **Edit** to update a notification or **Remove** to delete a notification.

Create notification

To create a new notification click on **Add notification** button and select a type of notification. Select an email tab if you are willing to be notified through an email or you can also configure an API endpoint to be listening to VirtEngine-Waldur notifications. Each notification has a question mar the right side, drag your mouse over it to see a list of all available events under selected category. Click **Create** to submit new notification.



Project Management

In VirtEngine-Waldur project is a main structural unit. Each organization consists of projects and each resource belongs to project.

Project permissions

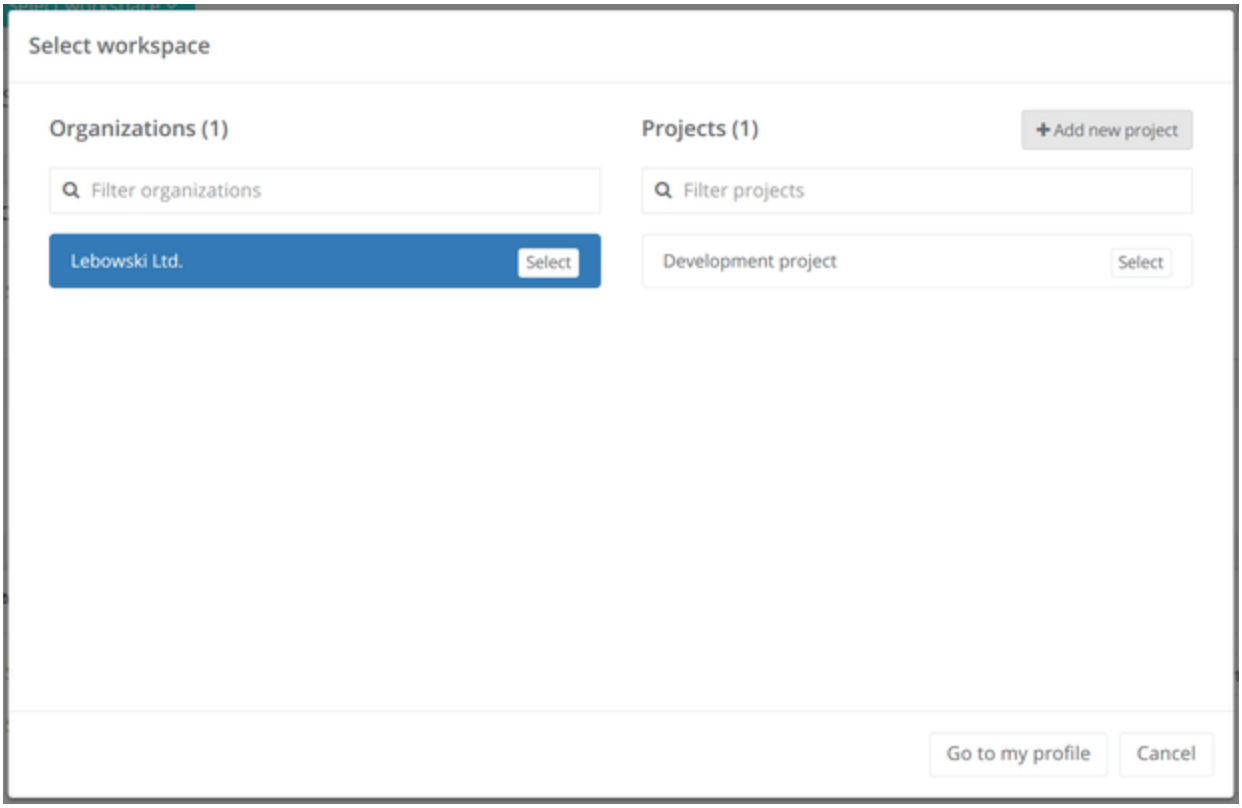
Note that available permissions depend on user role within organization and project.

Role	Permissions
Staff	Can manage (create, update, remove) projects in any organization.
Organization owner	Can manage (create, update, remove) projects only in owned organization.
Organization support	Can view projects only in supported organization.
Project manager	Can view only managed projects.
System administrator	Can view only administrated projects.

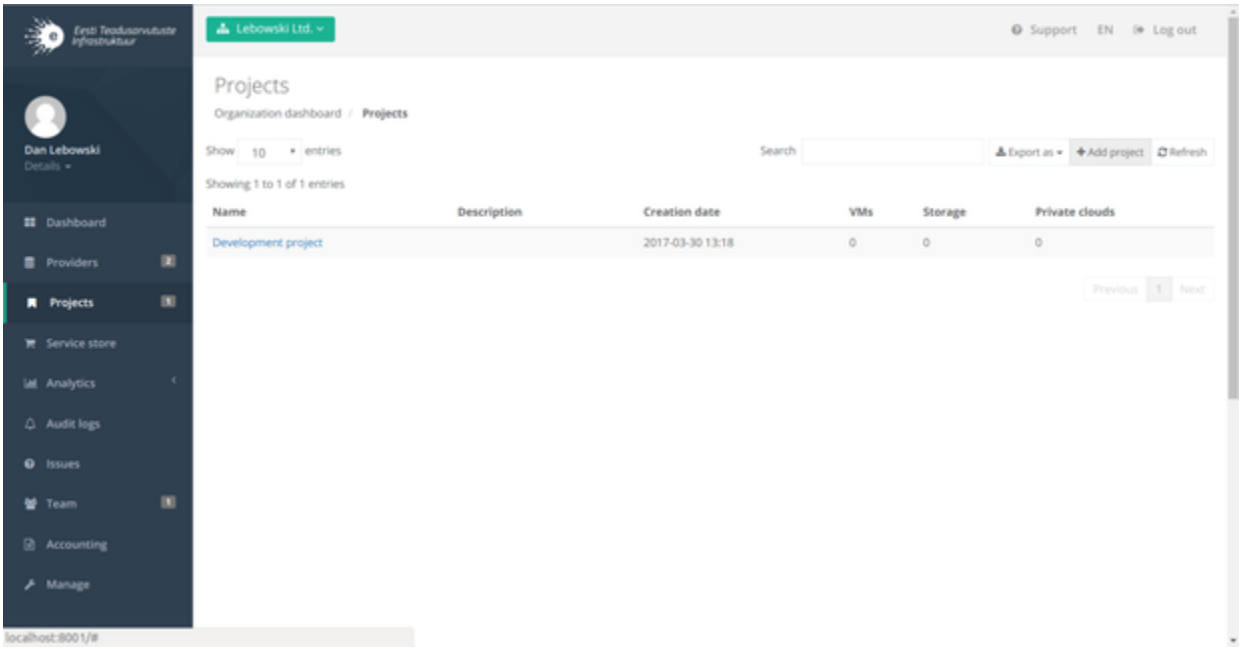
In order to manage project permissions you should click on "**Team**" link in a sidebar of a project workspace.

View projects

1. In order to view available organization and projects you should click "**Select workspace**" button.
2. If there are a lot of organizations or projects available, you can filter them by name.
   Note that project list is filtered by currently selected organization.
3. In order to switch to the project you should click "**Select**" button.

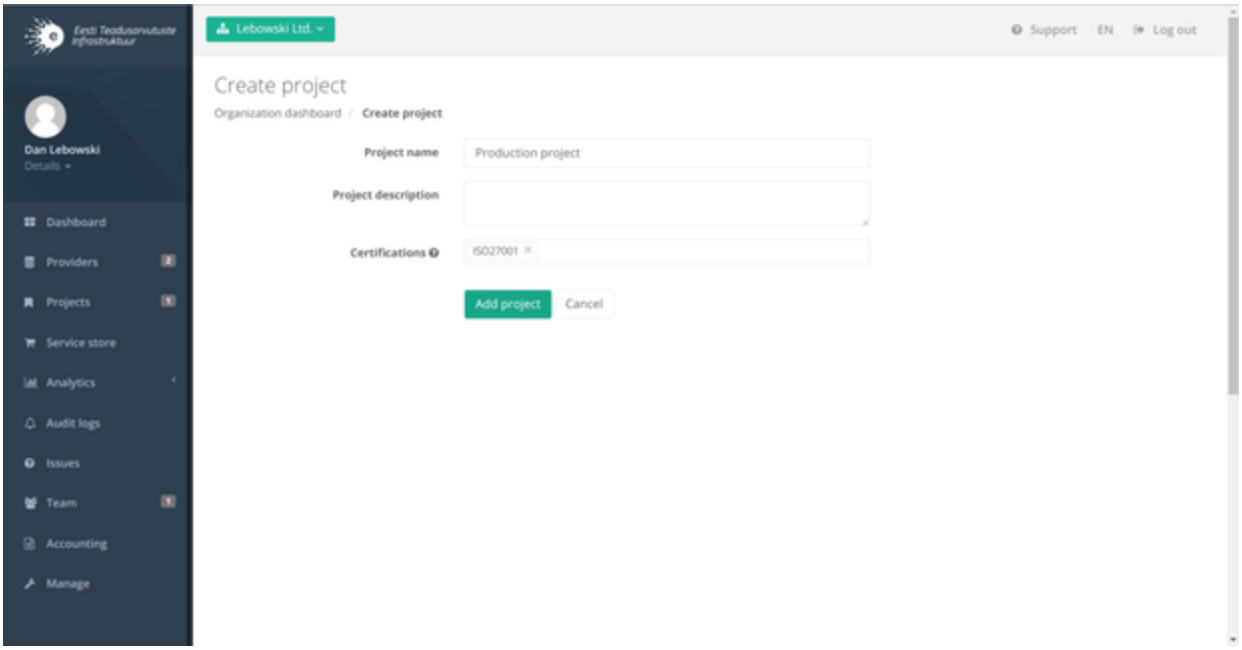


Staff, organization owner, organization support can also list projects in organization workspace.



Add new project

In order to create a new project you should click on "**Add project**" button either in "**Select workspace**" dialog or in projects list in the organization workspace.



Note

- "Add project" button is concealed for organization support, project manager and system administrator.
- "Add project" button may be disabled if project quota usage has reached its limit. In this case a tooltip is shown.
- Project name is mandatory, description and certifications are optional.
- After project is created you're redirected to a project list page.

Manage project

To manage a project you should click on **Manage** button in a projects list item in organization workspace.

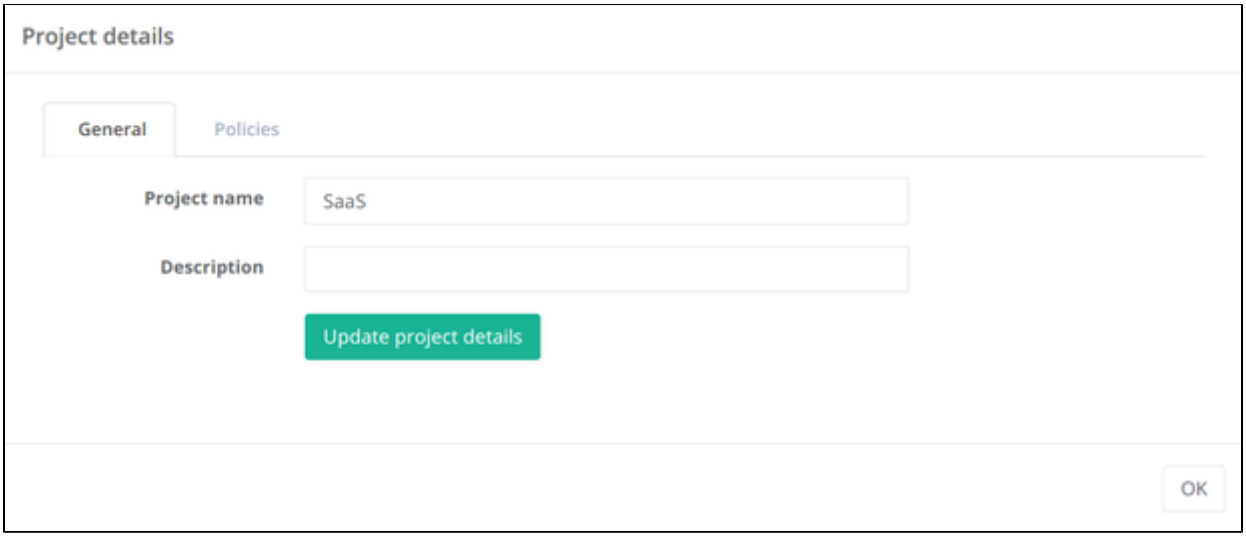
Project details dialog contains two tabs: **General** and **Policies.**

Note

- Staff and organization owner are allowed to change project name, description, project policies and remove a project completely.
- Support user is allowed only to view project details without modifying it.
- Only empty project can be deleted. Otherwise, if project has any resources, they should be deleted first.

Project details

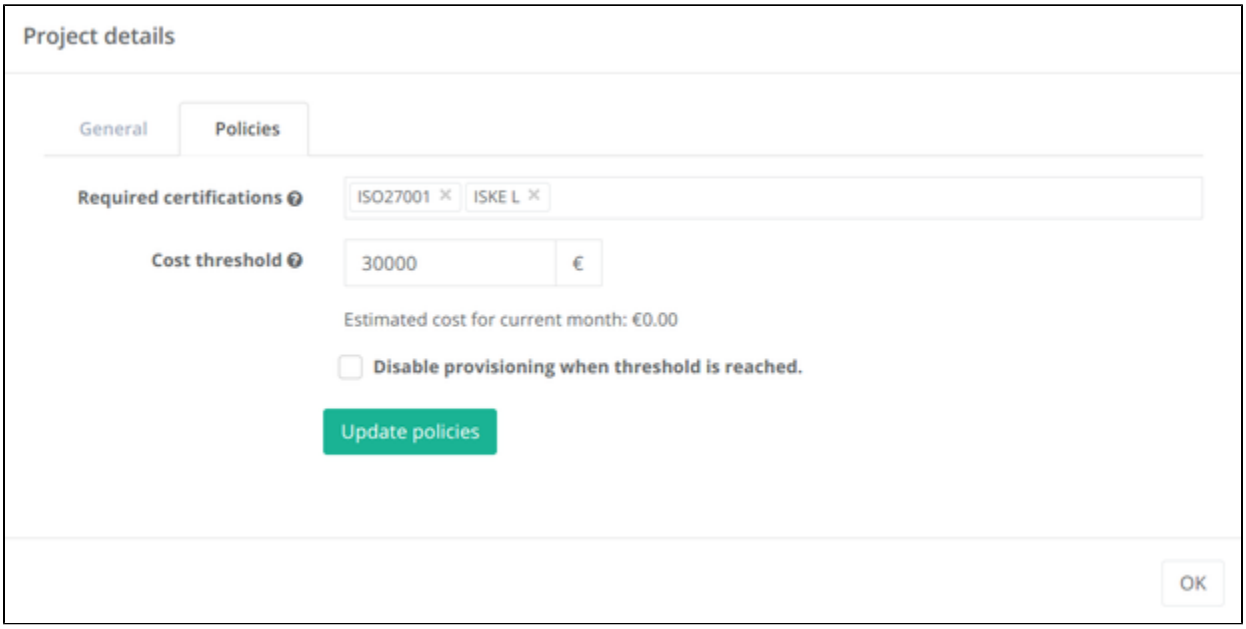
General tab allows to edit project name and description.



Project policies

There are several project policies available:

- **Required certifications** - allows to specify provider certification required by this project. For example, ISO27001.
  Note that available certifications should be specified beforehand by staff via admin site.
- **Cost threshold** - allows to specify upper bound of monthly resource cost provisioned from shared providers.
  Here you may find also estimated cost for current month.

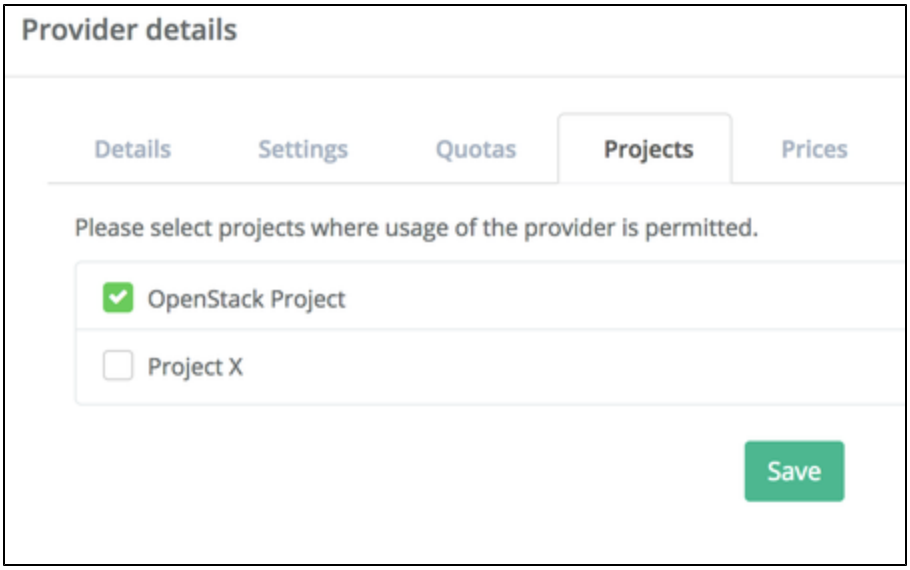


In order to update policies please click on **Update policies** button.

Associate projects with provider

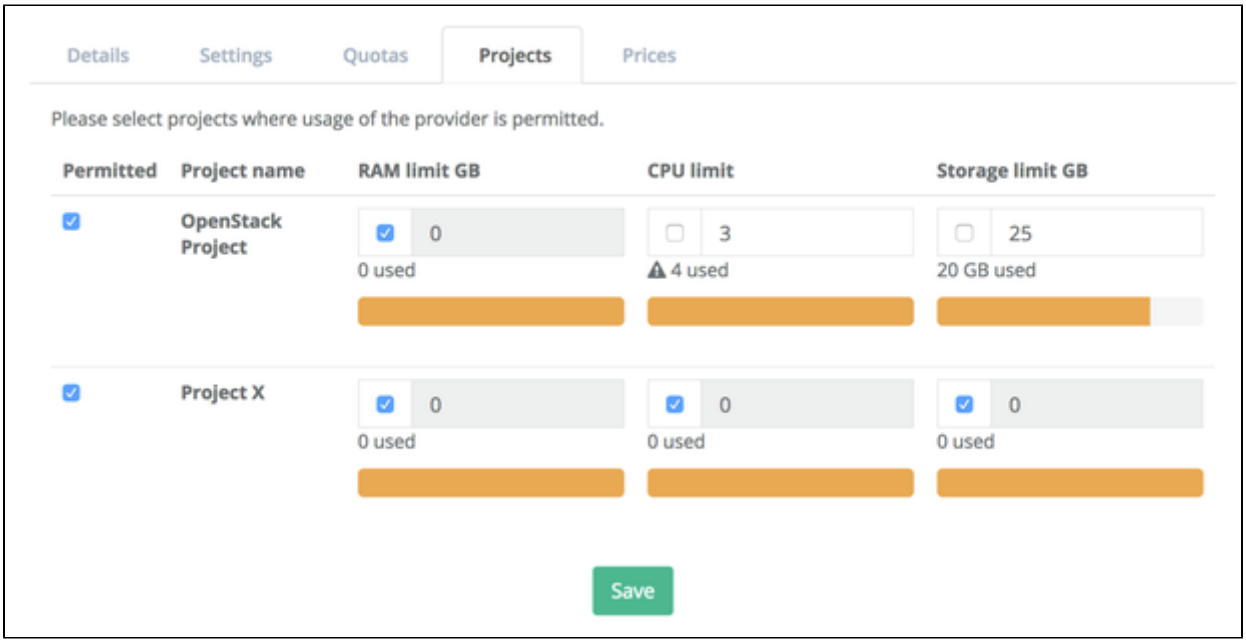
To associate a project with a provider, please follow these steps:

1. Switch to providers tab by clicking on **Providers** item in sidebar menu.
2. Select provider and click on **Details** button, wait for modal dialog to appear.
3. In modal dialog switch to **Projects** tab.
4. Select projects allowed to be used in this provider and click **Save**.

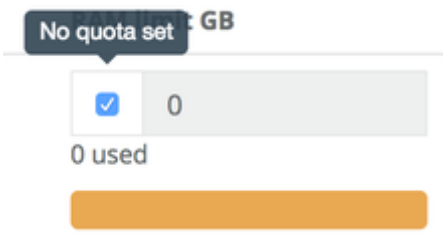


Associate projects with a cloud based provider

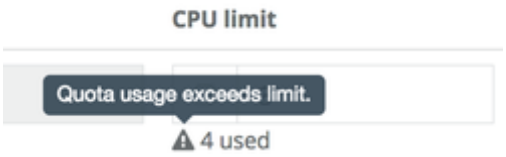
For cloud based providers like DigitalOcean, Amazon AWS or OpenStack Tenant a provider project association tab allows to set quotas per provider project association (link).



It is not required to set a quota limit for all resources, hit a checkbox near quota to set/unset quota limit.



If new quota exceeds usage a notification icon is going to appear.



Provider management

Introduction

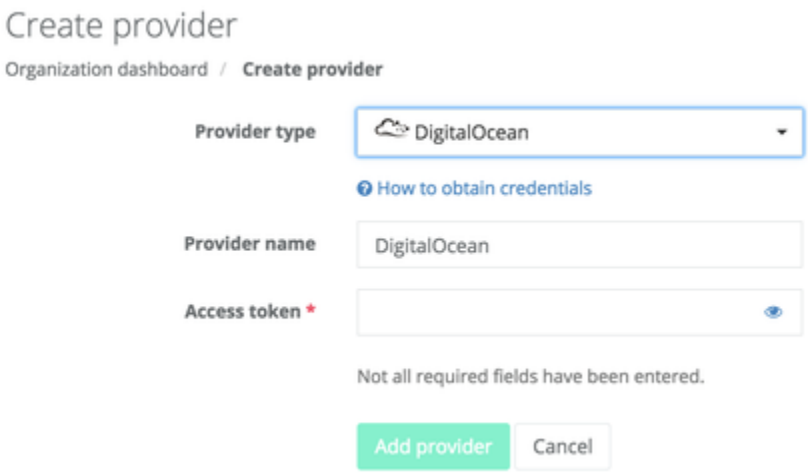
VirtEngine-Waldur provides an ability to combine different resource providers like OpenStack, Amazon, DigitalOcean etc. Each of this services is r as a provider and can be found under **Providers** menu item.

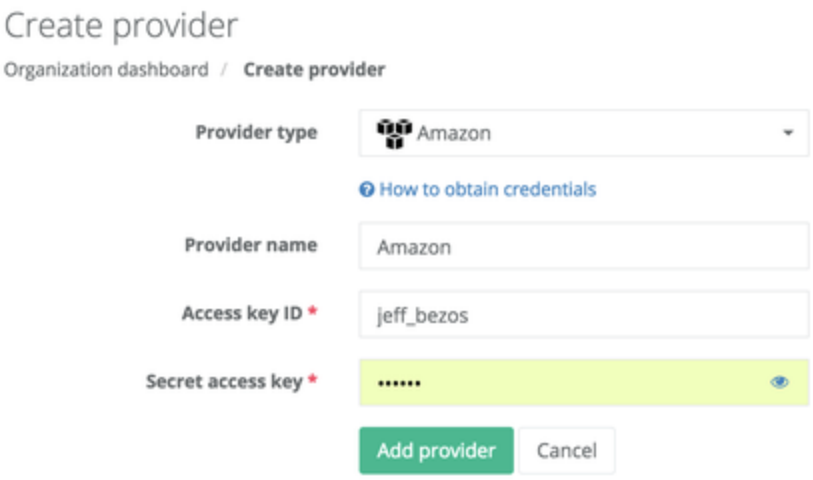
Automatically created providers

Except services as providers every main resource provided by a service is considered to be a provider itself, for instance an OpenStack Tenant is a provider because it provides private resources to configured group of users and, as a resource, OpenStack Tenant is created from the OpenStack provider, a concrete OpenStack system instance. All such providers are created and registered automatically.

Provider creation

To create a provider go to **Providers** menu and click **Add provider** button. Depending on a type of provider you will have to fill different set of configuration data. For instance, DigitalOcean provider requires only an **Access Token** when an OpenStack provider needs much more information to provide its services. Please find a few examples below.



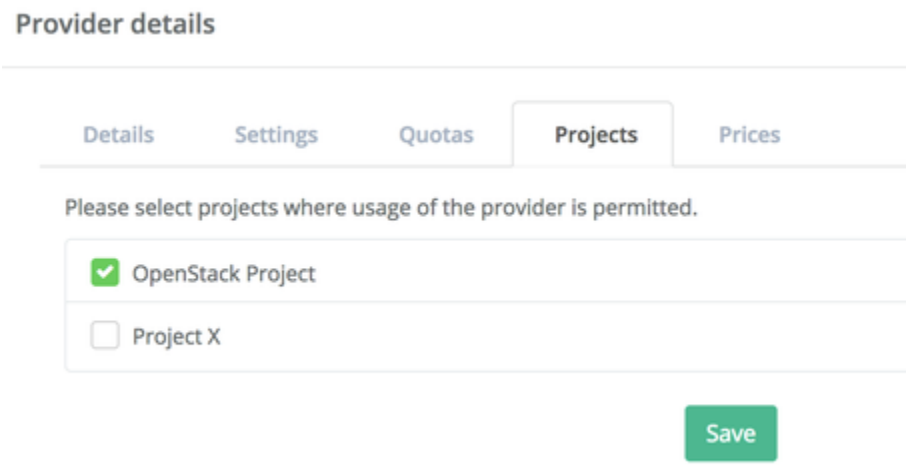


Please note, that depending on configuration, provider creation may be available to staff users only.

Provider modifications

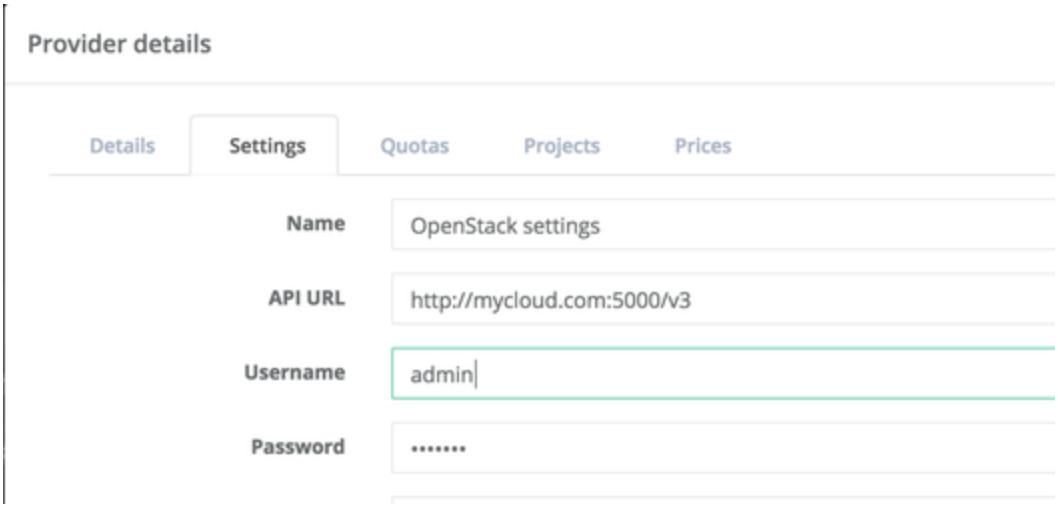
Associate with projects

To change a list of associated projects click **Details** on the provider, go to **Projects** tab, select all projects you want to associated with current provider and click **Save**.



Update settings

To update provider settings go to provider **Details**, select **Settings** tab and update required fields.



Please note, that depending on configuration, provider settings may be concealed.

Provider deletion

To delete a provider simply select **Remove** button.

Please note that provided configured by an administrator and marked as shared cannot be removed.

Please note, that depending on configuration, provider removal could be available to staff users only.

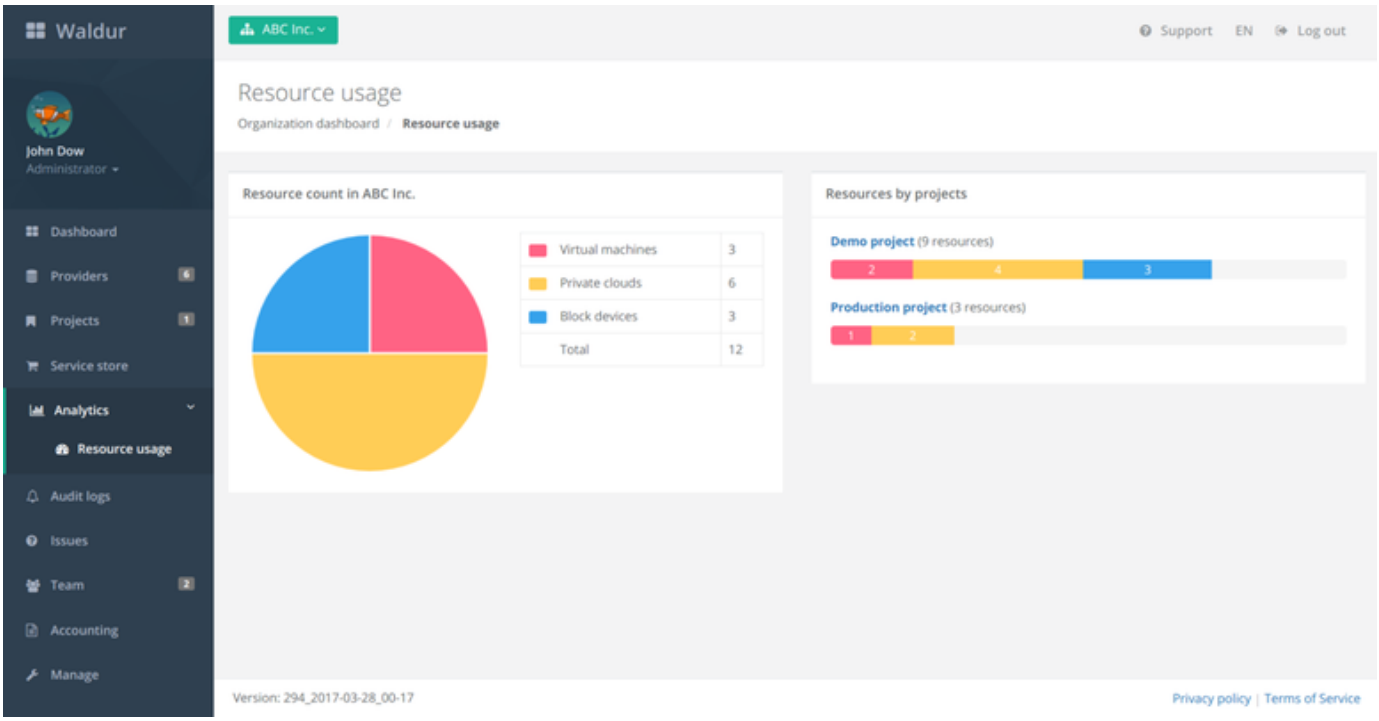
Reporting

Resource usage report

Resource usage report allows you to get an overall overview of the provisioned resources split by project and resource type (virtual machines, private clouds, block devices).

In order to navigate to "Resource usage report" page you should follow these steps:

1. (Optional) Switch to the organization workspace using workspace selector at the top navigation bar.
2. In sidebar menu click on "Analytics" and then "Resource usage".



Note

- Users with roles other than organization owners, global support and staff are not allowed to view this page.
- If there are no resources corresponding to some resource type or project it is not displayed in usage report.

- Quotas and historical queries
- Support for aggregated metrics on organsiation and project levels

Request-based items

Overview

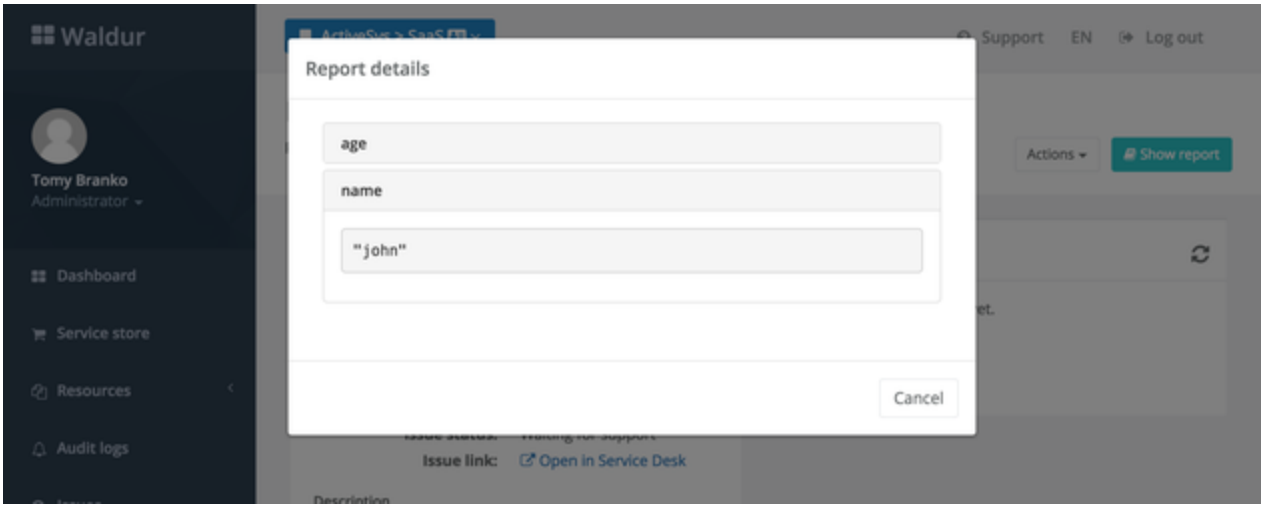
Request-based items are service offerings that are delivered through an offline process. They allow to quickly expose service offerings without performing in-depth API integration.

Requested resources can be seen under "Resources > Requests" menu in Project workspace.

Report

An operator might provide additional details for the requested service, for example connection details or other metadata. Details will be available for users that have access to the project.

If available, it is possible to see it from HomePort by clicking on the "Show report" button in the right part of the screen.



Updating report

Updating the report requires staff role. To update a report of a requested service:

1. Get its UUID (e.g. from HomePort's URL).
2. Execute a PUT call to the resource URL providing report content as a structured JSON dictionary. The top-level key values will be rendered as titles of the blocks and values will be displayed in pre-formatted fashion in the block.

```
$ http -v PUT
https://waldur.example.com/api/support-offerings/73db2edf0278471599
8cb477c74bc898/ Authorization:"Token
6f3ab8c2ab91210faea96b23f0585a765e5f42fd" name=same_name
report:='{"name": "john", "age": 10 }'
```

Roles

Introduction

There are two main structural units in VirtEngine-Waldur: organizations and projects.
Usually, each organization has multiple projects, and each project owns resources.

Users are connected to the organizations and projects through roles. Each user may have several roles.
Role-Based Access Control enables granular permission management to resources for specific organizations and projects.

Role	Permissions
Staff	Can manage anything.

Global support	Has read-only access to all organizations, projects and resources.
Organization owner	Can manage users, projects and resources only in owned organization.
Organization support	Has read-only access to all users, projects and resources in a supported organization.
Project manager	Can manage his project.
System administrator	Can manage only resources in his project.

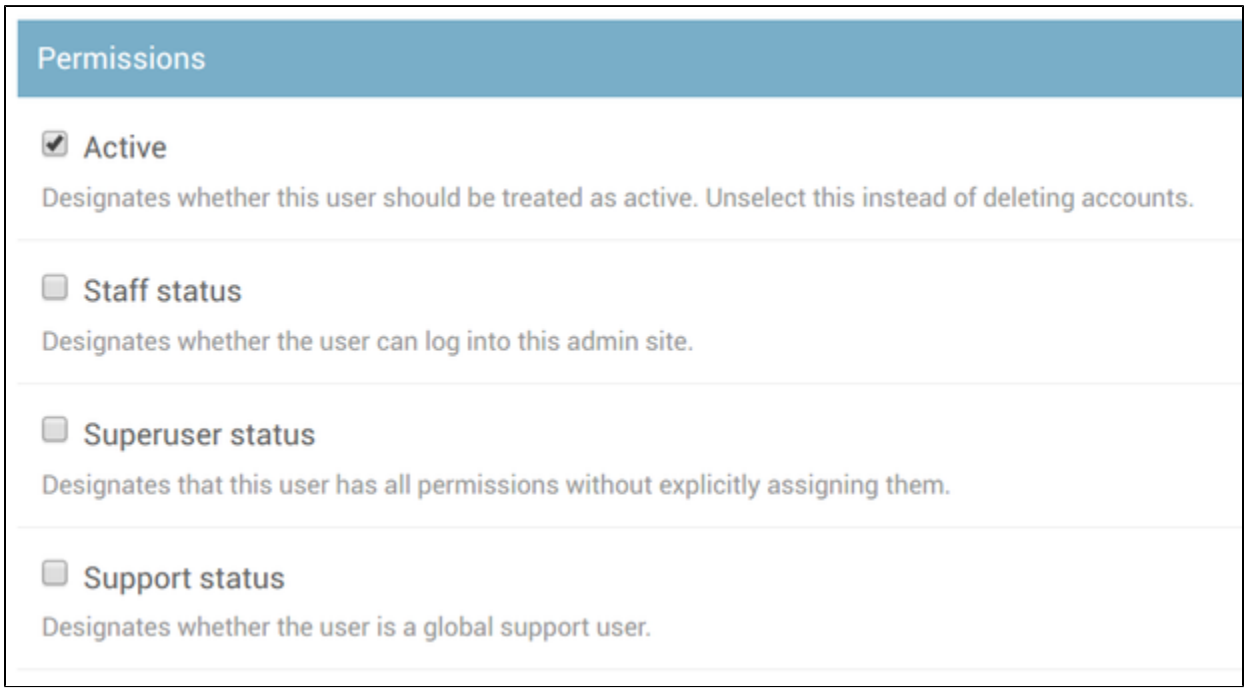
Note that there are several configuration options which allow to enable or disable particular actions depending on user role.

Permissions management via admin site

Global roles management

In order to manage global user roles, you should log into admin site, and click on menu **Structure Users**.

Then in **Permissions** section you may grant user staff or global support role.

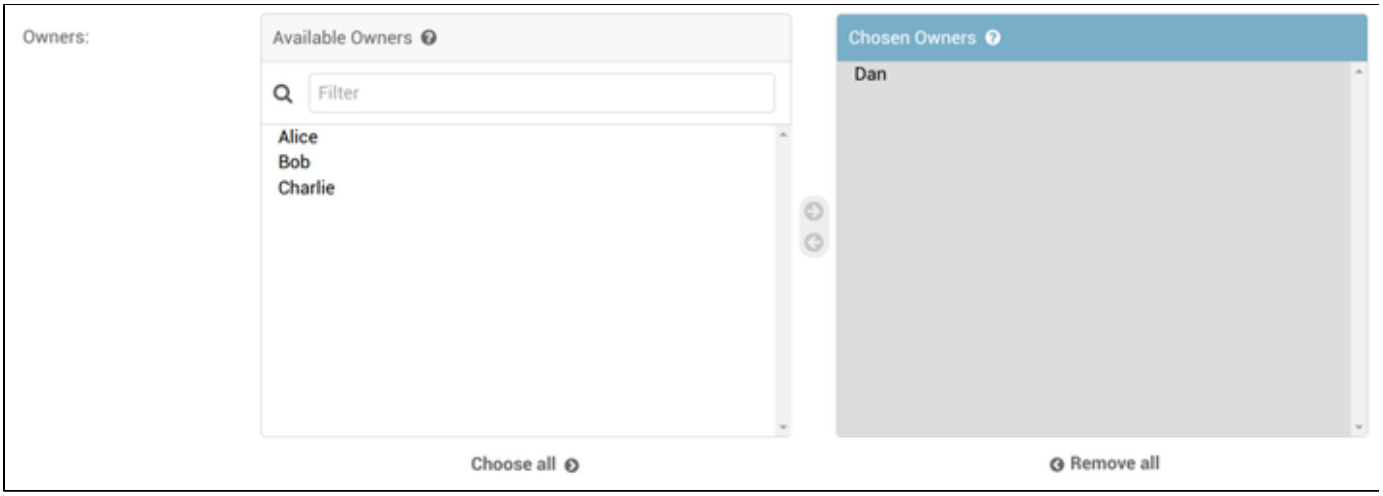


Note that global roles management is available only to staff user and only via admin site.

Organization permissions management

To manage organization permissions as a staff user please follow these instructions:

1. Click on menu **Structure Organizations** and select organization.
2. Grant or revoke permissions by moving users between **Available Owners** and **Chosen Owners** lists in **Owners** section.
3. Click on **Save** button to apply changes.



Project permissions management

To manage project roles please follow these instructions:

1. Click on menu **Structure Projects** and select project.
2. Grant or revoke permissions by moving users in **Admins**, **Managers** and **Support users** sections.
3. Click on **Save** button to apply changes.

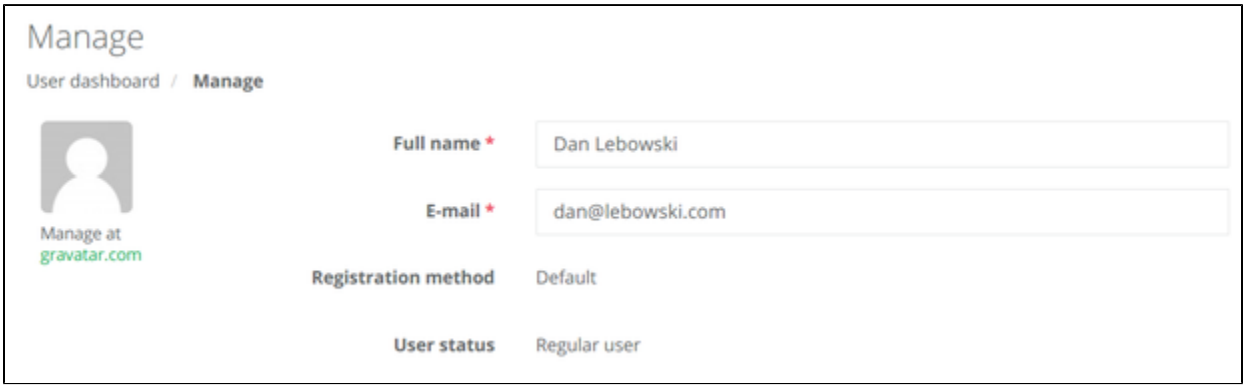
Note that currently expiration time and permissions log feature is not available via admin site. These features are available in VirtEngine-Waldur H only.

Permissions management via VirtEngine-Waldur HomePort

View current user role

The current user role is displayed in **User management** page in VirtEngine-Waldur HomePort.

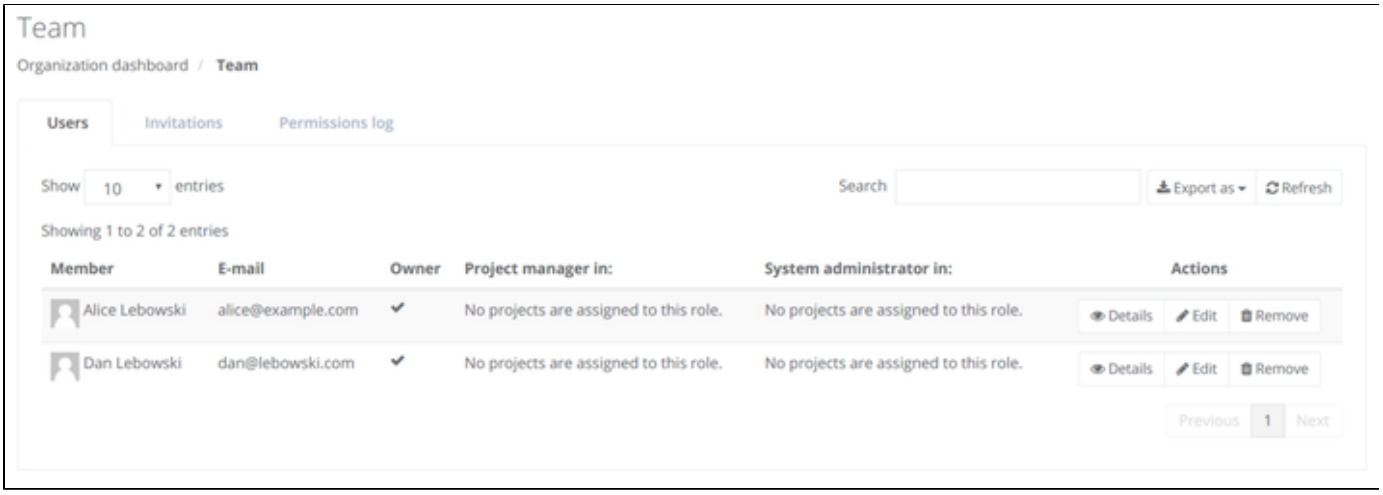
In order to view it, please switch to user workspace and then click on **Manage** link on the sidebar.



List organization users

To manage organization permissions please follow these instructions:

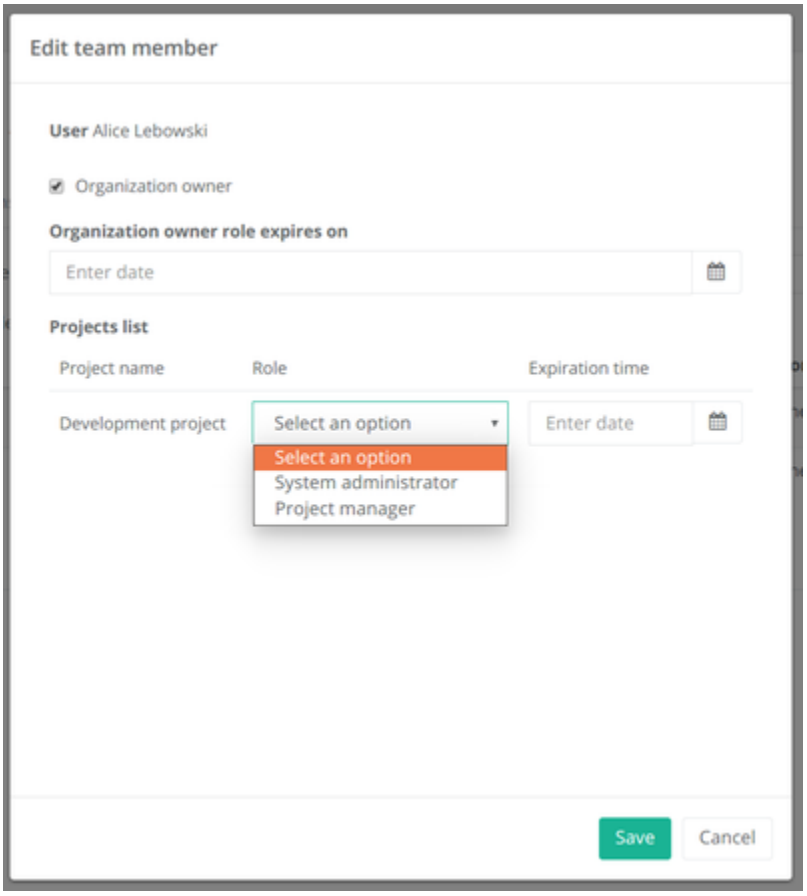
1. Log into VirtEngine-Waldur HomePort as an organization owner or staff.
2. Select an organization, and switch to workspace by clicking on **Select workspace** button.
3. Click on **Team** link in the sidebar menu.



As you can see **Team management** page consists of three tabs: Users, Invitations and Permission log.

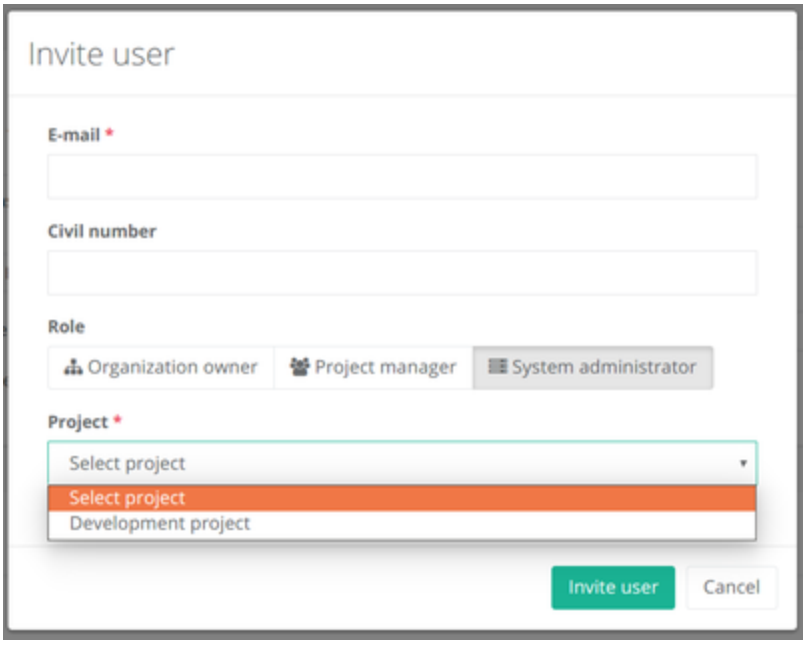
Edit organization roles

Users tab lists all users which have access to the organization, including organization owners, project managers, and system administrators. It also allows to view user details, grant or revoke organization owner role to existing users, and manage user roles for projects within the organization.



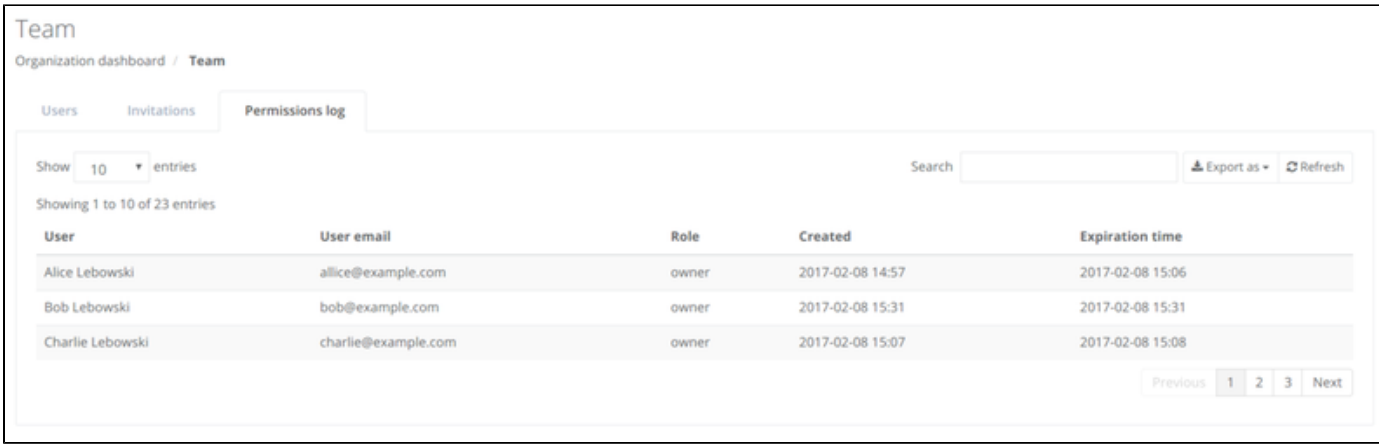
Invite new users

Invitations tab lists all invitations, and allows to create a new invitation, cancel or resend pending invitation.
Note that you can't directly assign organization roles in VirtEngine-Waldur HomePort. Instead, you should create an invitation first.
Only when the invitation is accepted by the user, the role is actually granted.



List revoked permissions

Note that each permission may have an expiration time. When permission is revoked, it's moved to log so that you may identify who had access to resources at the particular point of time.
Permission log tab lists all permissions, which has been previously revoked.

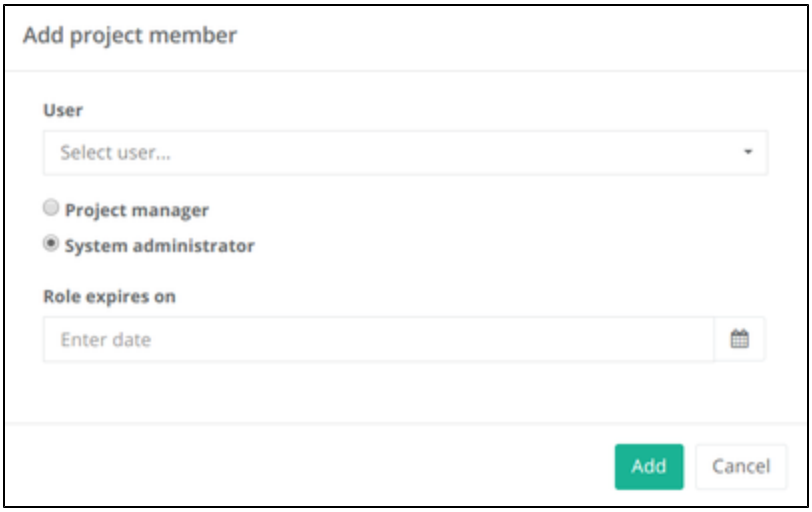


Manage project permissions

To manage organization permissions please follow these instructions:

1. Log into VirtEngine-Waldur HomePort as an organization owner, project manager or staff.
2. Switch to the needed project by clicking on **Select workspace** button.
3. Click on **Team** link in the sidebar menu.

As you can see **Team management** page in project workspace consists of two tabs: **Users** and **Permissions log**.
In Users tab all project members are listed. Its content is similar to the Users tab in the organization workspace. It allows you to grant or revoke permissions.
Note that you may grant project permission to the user which already has some role in the project's organization.



Team Management

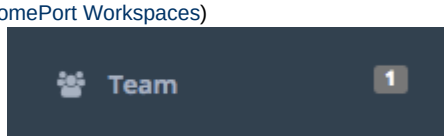
Introduction

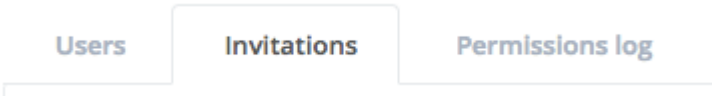
VirtEngine-Waldur has a flexible team management system, which allows to easily assign and revoke roles in organization and projects. It uses inv
adding users to the organizations and supports temporary roles in organization and projects.

Invitations

Invitations are used to add a new user to the team.

Navigate to invitations section

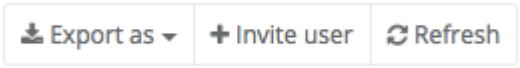
1. Select organization workspace (see HomePort Workspaces)
2. Choose `Team` from the navigation bar: 

3. Select `Invitations` tab: 

Create invitation

Invitation allows new user to join the organization team.

1. Navigate to the invitations section (see above)
2. Click on the `Invite user` button in the top right corner:



3. Fill in the invitation form:
    - `email address` - invitation will be send to this email.
    - `civil number` - only user who has this civil number will be able to accept the invitation. This field is optional. If not defined, a link contained in the invitation e-mail will be enough to get the permission.
    - `role` - specify what role should be assigned to the user in this organization (see Roles). For system administrator or project manager roles you must specify project as well.
      The role expiration time is not set by default. Note, that you can change the role of the user or its expiration time after in `Users` section of the `Team`.
4. Click on the `Invite user` button.

Accept invitation

User should follow the link in invitation email, upon login the invitation will be automatically accepted. Note, that invitation will not be accepted if user has an invalid civil number (see invitation creation above). In addition, invitation will not be accepted if it has already expired or has been cancelled.

Resend invitation

In case when the user has not received the invitation email, it can be resent by clicking on the `Resend` button.

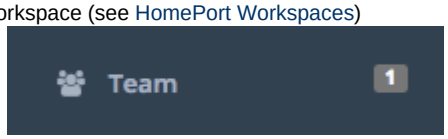
Cancel invitation

The invitation can be cancelled by clicking on the `Cancel` button. If the invitation is canceled, user will not be able to join the organization by following the link in the invitation email.

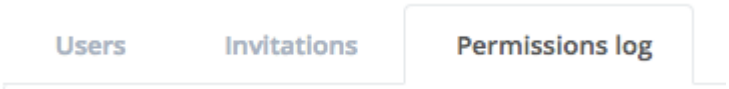
Permissions log

Every time a user role is assigned or revoked the event log is generated. The history of role changes can be reviewed in `Permissions log:`

1. Select either organization or project workspace (see HomePort Workspaces)
2. Choose `Team` from the navigation bar:



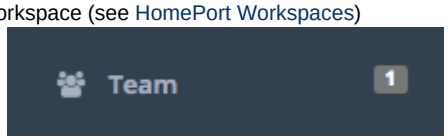
3. Select `Permissions log` tab:



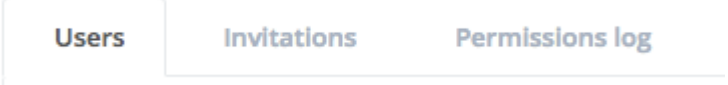
View team member details

Team member details can be view by navigating to the `Users` section:

1. Select either organization or project workspace (see HomePort Workspaces)
2. Choose `Team` from the navigation bar:



3. Select `Users` tab:



4. Click on the `Details` button for the user in the list.

Project team management

Used for manipulating team members in project by managers.

Navigate to the project team management

1. Select project workspace (see HomePort Workspaces)
2. Choose `Team` from the navigation bar.
3. Select `Users` tab.

Assign team member a role in a project

1. Click on the `Add member` button.
2. Fill in the form
    - `user` - the user role who will receive the role. Only users with role in organization can be selected.
    - `role` - System Administrator or Project Manager (see Roles).
    - `Role expires on` - after specified date user role in project will be revoked.
3. Click on `Add` button.

Edit team member role in a project

To change team member role in the project click on the `Edit` button.

Remove team member from a project

To remove team member from the project click on the `Remove` button.

Organization team management

Used for manipulating team members by organization owners. User can assign roles in organization and projects.

Navigate to the organization team management

1. Select organization workspace (see HomePort Workspaces)
2. Choose `Team` from the navigation bar.
3. Select `Users` tab.

Edit team member role in an organization

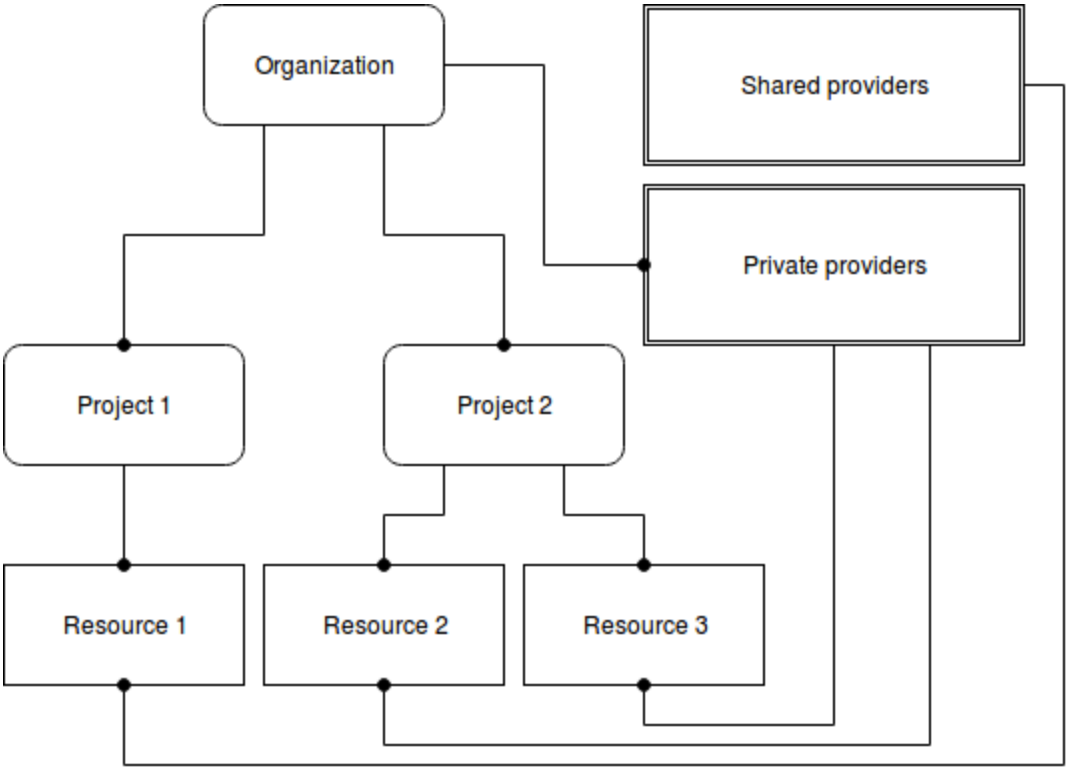
User can assign and revoke organization owner role as well as roles in projects.

Remove team member from an organization

To remove team member from the organization click on the `Remove` button. The user will lose all his roles in organization, and will be removed from the team.

VirtEngine-Waldur Organization Structure

VirtEngine-Waldur defines a structural organization for its elements. Each organization may contain several projects and use different providers for creation. Each resource should belong to some project.



Organization - a standalone entity that represents a company or a department.

Project - an entity within an organization that aggregates and isolates teams and resources.

Provider - an entity that represents account in an external service provider.

- **Private providers** - providers that are available and manageable within a specific organization.
- **Shared providers** - global providers that are available for all organizations.

Resource - an entity within a project and a provider. Represents cloud resource. Examples: instance in AWS, droplet in DigitalOcean.

API

Introduction

VirtEngine-Waldur MasterMind is exposing a REST API for all relevant actions.

API can be learnt from:

- https://mastermind.example.com/docs – a Swagger-based API documentation
- https://mastermind.example.com/api - in development mode (set DEBUG to true in settings), it will show a browsable API interface for easy experimentation.
- API Examples
- Examining requests that VirtEngine-Waldur HomePort is doing against the MasterMind API. Use any modern browser's developer tools fo

Examples

Introduction

VirtEngine-Waldur MasterMind exposes REST API for all of its operations. Below are examples of typical operations performed against APIs. To ru
examples, we are using a HTTPie https://httpie.org/.

Almost all of the operations with API require an authentication token. Below we list two methods on how to get it:

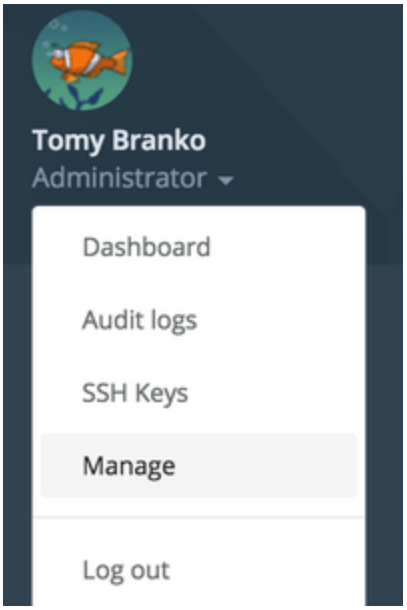
Authentication with username/password

If your account is allowed to use username/password and the method is enabled (e.g. in dev environment), you can get a new token by submitting a username/password as JSON to a specific endpoint.

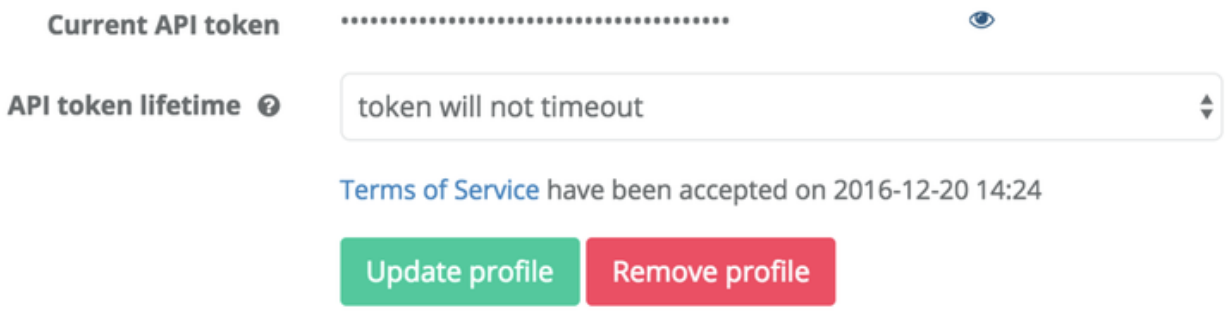
```
$ http -v POST https://waldur.example.com/api-auth/password/
username=user password=password

POST /api-auth/password/ HTTP/1.1
Accept: application/json
Content-Length: 40
Content-Type: application/json
Host: waldur.example.com
User-Agent: HTTPie/0.9.4

{
    "password": "user",
    "username": "password"
}

HTTP/1.1 200 OK
Allow: POST, OPTIONS
Connection: keep-alive
Content-Type: application/json
Vary: Accept, Cookie

{
    "token": "65b4c4f5e25f0cadb3e11c181be4ffa3881741f8"
}
```

Authentication Token management

The easiest way to get your token is via VirtEngine-Waldur HomePort. Go to user workspace by selecting 'Manage' in the user drop-down.



Scroll down to the Current API token field and click on the 'eye' icon to display the token.



Creation of an OpenStack Tenant

To create an OpenStack Tenant send a POST request to **/api/openstack-packages/**.

This will trigger a provisioning operation for OpenStack Tenant and return tenant package details as response.

Name	Type	Required	Description
name	string	True	Tenant name
template	url	True	An OpenStack package template to create a tenant from
service_project_link	url	True	A service project link to associate tenant with. The link is a url to any service settings with 'Open Stack' type.
subnet_cidr	string	False	Defines internal subnet that will be auto-created within a new OpenStack Tenant

Template and service project link restrictions

Please note that OpenStack Provider templates must be configured by an administrator before tenant creation. More information can be found in Shared OpenStack Provider Management#VPCpackagetemplatemanagement

A template must be linked to the OpenStack Tenant provider settings.

```
https://waldur.example.com/api/package-templates/?service_settings_
uuid=dcae83636530464d8fa50c83c8334a6c
```

A service project link must be link to the OpenStack Tenant provider.

```
https://waldur.example.com/api/openstack-service-project-link/?serv
ice_uuid=341f793509e34bec806037a1fd749338
```

Tenant package details contains attributes associated with a standard tenant representation.

Name	Type	Description
name	string	Tenant name
description	string	Tenant description
service_settings	url	Tenant settings
template	url	A template that was used to create the tenant
tenant	url	A link to the tenant
url	url	A package that was used to create the tenant
uuid	uuid	A package identifier

An example of request

Tenant creation

```
http -v POST https://waldur.example.com/api/openstack-packages/ \
  name="tenant name" \
  description="tenant description" \

service_project_link="https://waldur.example.com/api/openstack-service-p
roject-link/1/" \
  availability_zone="us-east-1a" \
  user_username="tenant user name" \
  subnet_cidr="192.168.42.0/24" \

template="https://waldur.example.com/api/package-templates/7615809710144
4dab102a6e981f2fbd0/" \
  Authorization:"token a2eef51ca3c290bdd0b5d9e760b767e9f5d71f87"
POST /api/openstack-packages/ HTTP/1.1
Accept: application/json, */*
Accept-Encoding: gzip, deflate
Authorization: token a2eef51ca3c290bdd0b5d9e760b767e9f5d71f87
Connection: keep-alive
```

```
Content-Length: 346
Content-Type: application/json
Host: waldur.example.com
User-Agent: HTTPie/0.9.8

{
    "availability_zone": "us-east-1a",
    "description": "tenant description",
    "name": "tenant name",
    "service_project_link":
"https://waldur.example.com/api/openstack-service-project-link/1/",
    "subnet_cidr": "192.168.42.0/24",
    "template":
"https://waldur.example.com/api/package-templates/76158097101444dab102a6
e981f2fbd0/",
    "user_username": "tenant user name"
}

HTTP/1.0 201 Created
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Date: Tue, 14 Mar 2017 14:08:46 GMT
Server: WSGIServer/0.1 Python/2.7.13
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN

{
    "description": "tenant description",
    "name": "tenant name",
    "service_settings":
"https://waldur.example.com/api/service-settings/aea99a41affd459f8327fdf
1fb557795/",
    "template":
"https://waldur.example.com/api/package-templates/76158097101444dab102a6
e981f2fbd0/",
    "tenant":
"https://waldur.example.com/api/openstack-tenants/a7e76afdd9b04673b51c46
80658f62fa/",
    "url":
"https://waldur.example.com/api/openstack-packages/6378d0cbea91405e82afd
```

```
3e62ae70420/",
    "uuid": "6378d0cbea91405e82afd3e62ae70420"
}
```

Creation of an OpenStack Instance

To create an OpenStack Instance send a POST request to **/api/openstacktenant-instances/**

Name	Type	Required	Description	Requirements
name	string	True	Instance name.	Max length is 150 symbols.
service_project_link	url	True	A service project link to associate instance with.	
user_data	string	False	Additional data to add to instance on provisioning.	Data has to be provided in YAML format.
flavor	url	True	A predefined set of instance components parameters.	Flavor ram and disk must satisfy minimal image system requirements.
image	url	True	An OS image to apply on instance.	Must belong to the same tenant settings as flavor.
system_volume_size	integer	True	An instance system volume size in Mb.	Volume size has to be greater than minimal image volu me size.
data_volume_size	integer	False	An instance data volume size in Mb.	Minimum value is 1024.
internal_ips_set	list	True	A list of OpenStack Tenant subnets to connect the instance to.	

Image and flavor restrictions

An instance image and flavor must be linked to the OpenStack Tenant provider settings.

There are two ways to filter images and flavors for provider settings.

First - using settings UUID:

```
https://waldur.example.com/api/openstacktenant-flavors/?settings_uu
id=dcae83636530464d8fa50c83c8334a6c
```

Second - using a full settings URL:

```
https://waldur.example.com/api/openstacktenant-images/?settings=htt
ps://waldur.example.com/api/service-settings/dcae83636530464d8fa50c
83c8334a6c/
```

If an image or flavor belongs to a different settings - an instance creation is going to be rejected with the message "Flavor and image must belong to the same service settings as service project link."

Network restrictions

A subnet must belong to the **internal** OpenStack Tenant network which can be found by tenant settings.

```
https://waldur.example.com/api/openstacktenant-networks/?name=&sett
ings_uuid=dcae83636530464d8fa50c83c8334a6c&is_external=False
```

The response will be a JSON representation of the instance.

An example of request

```
http -v POST https://waldur.example.com/api/openstacktenant-instances/ \
  name="my instance name" \

service_project_link="https://waldur.example.com/api/openstacktenant-ser
vice-project-link/9/" \
  user_data="version: \"1.0\"" \

flavor="https://waldur.example.com/api/openstacktenant-flavors/e4c1f185f
1964e3984063581dff728e8/" \

image="https://waldur.example.com/api/openstacktenant-images/3619fa9b5a2
d4cba98ccbc3ccbe044f9/" \
  system_volume_size:=1024 \
  data_volume_size:=1024 \

internal_ips_set:='[{"subnet":"https://waldur.example.com/api/openstackt
enant-subnets/24fd6192a55b4f5e813b33cd15bfbd16/"}]' \
  Authorization:"token a2eef51ca3c290bdd0b5d9e760b767e9f5d71f87"
POST /api/openstacktenant-instances/ HTTP/1.1
Accept: application/json, */*
Accept-Encoding: gzip, deflate
Authorization: token a2eef51ca3c290bdd0b5d9e760b767e9f5d71f87
Connection: keep-alive
Content-Length: 400
Content-Type: application/json
Host: waldur.example.com
User-Agent: HTTPie/0.9.8

{
    "data_volume_size": 1024,
    "flavor":
"https://waldur.example.com/api/openstacktenant-flavors/e4c1f185f1964e39
84063581dff728e8/",
    "image":
"https://waldur.example.com/api/openstacktenant-images/3619fa9b5a2d4cba9
8ccbc3ccbe044f9/",
    "name": "my instance name",
    "service_project_link":
```

```
"https://waldur.example.com/api/openstacktenant-service-project-link/9/"
,
    "system_volume_size": 1024,
    "user_data": "version: \"1.0\""
}

HTTP/1.0 201 Created
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Date: Wed, 15 Mar 2017 11:10:10 GMT
Location:
https://waldur.example.com/api/openstacktenant-instances/65cf08f7af884d4
5ac9479338d3a44ae/
Server: WSGIServer/0.1 Python/2.7.13
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN

{
    "access_url": null,
    "action": "",
    "action_details": {},
    "backend_id": "",
    "cores": 1,
    "created": "2017-03-15T11:10:10.206420Z",
    "customer":
"https://waldur.example.com/api/customers/53dee275177d4b878489eaae4c0690
22/",
    "customer_abbreviation": "TM abbr",
    "customer_name": "Taras Matsyk",
    "customer_native_name": "",
    "description": "",
    "disk": 2048,
    "error_message": "",
    "external_ips": [],
    "flavor_disk": 8192,
    "flavor_name": "m1.tiny",
    "floating_ips": [],
    "image_name": "TestVM",
    "internal_ips": [
        null
    ],
    "internal_ips_set": [
        {
            "ip4_address": null,
            "mac_address": "",
            "subnet":
"https://waldur.example.com/api/openstacktenant-subnets/24fd6192a55b4f5e
813b33cd15bfbd16/",
            "subnet_cidr": "192.168.42.0/24",
            "subnet_description": "",
```

```
            "subnet_name": "tenant-1-sub-net",
            "subnet_uuid": "24fd6192a55b4f5e813b33cd15bfbd16"
        }
    ],
    "key_fingerprint": "",
    "key_name": "",
    "latitude": null,
    "longitude": null,
    "min_disk": 0,
    "min_ram": 64,
    "monitoring_items": null,
    "name": "my instance name",
    "project":
"https://waldur.example.com/api/projects/77c23dbcb94a4ad5bba54d82c17fa86
2/",
    "project_name": "OpenStackProject",
    "project_uuid": "77c23dbcb94a4ad5bba54d82c17fa862",
    "ram": 512,
    "resource_type": "OpenStackTenant.Instance",
    "runtime_state": "",
    "security_groups": [],
    "service":
"https://waldur.example.com/api/openstacktenant/b2045458714e4ba19da129ee
c780ec73/",
    "service_name": "Tenant #43",
    "service_project_link":
"https://waldur.example.com/api/openstacktenant-service-project-link/9/"
,
    "service_settings":
"https://waldur.example.com/api/service-settings/dcae83636530464d8fa50c8
3c8334a6c/",
    "service_settings_error_message": "",
    "service_settings_state": "OK",
    "service_settings_uuid": "dcae83636530464d8fa50c83c8334a6c",
    "service_uuid": "b2045458714e4ba19da129eec780ec73",
    "sla": null,
    "start_time": null,
    "state": "Creation Scheduled",
    "tags": [],
    "url":
"https://waldur.example.com/api/openstacktenant-instances/65cf08f7af884d
45ac9479338d3a44ae/",
    "user_data": "name: user name",
    "uuid": "65cf08f7af884d45ac9479338d3a44ae",
    "volumes": [
        {
            "bootable": true,
            "name": "my instance name-system",
            "resource_type": "OpenStackTenant.Volume",
            "size": 1024,
```

```
            "state": "Creation Scheduled",
            "url":
"https://waldur.example.com/api/openstacktenant-volumes/14d01a2e368646ce
a6df66027a5d0493/",
            "uuid": "14d01a2e368646cea6df66027a5d0493"
        },
        {
            "bootable": false,
            "name": "my instance name-data",
            "resource_type": "OpenStackTenant.Volume",
            "size": 1024,
            "state": "Creation Scheduled",
            "url":
"https://waldur.example.com/api/openstacktenant-volumes/9f0fbb448b8a4dbe
a857b19a2d79f71b/",
            "uuid": "9f0fbb448b8a4dbea857b19a2d79f71b"
```

```
        }
    ]
}
```

Deletion of an OpenStack Instance

An example of request

```
http -v POST
https://waldur.example.com/api/openstacktenant-instances/65cf08f7af884d4
5ac9479338d3a44ae/stop/ \
  Authorization:"token a2eef51ca3c290bdd0b5d9e760b767e9f5d71f87"
POST
/api/openstacktenant-instances/65cf08f7af884d45ac9479338d3a44ae/stop/
HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Authorization: token a2eef51ca3c290bdd0b5d9e760b767e9f5d71f87
Connection: keep-alive
Content-Length: 0
Host: waldur.example.com
User-Agent: HTTPie/0.9.4

HTTP/1.1 202 Accepted
Allow: POST, OPTIONS
Connection: keep-alive
Content-Language: en
Content-Length: 31
Content-Type: application/json
Date: Fri, 25 Aug 2017 14:38:28 GMT
Server: nginx/1.10.2
Vary: Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN

{
    "status": "stop was scheduled"
}

http -v DELETE
https://waldur.example.com/api/openstacktenant-instances/65cf08f7af884d4
5ac9479338d3a44ae/ \
  Authorization:"token a2eef51ca3c290bdd0b5d9e760b767e9f5d71f87"
DELETE /api/openstacktenant-instances/65cf08f7af884d45ac9479338d3a44ae/
HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Authorization: token a2eef51ca3c290bdd0b5d9e760b767e9f5d71f87
Connection: keep-alive
```

```
Content-Length: 0
Host: waldur.example.com
User-Agent: HTTPie/0.9.4

HTTP/1.1 202 Accepted
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS
Connection: keep-alive
Content-Language: en
Content-Length: 34
Content-Type: application/json
Date: Fri, 25 Aug 2017 14:40:48 GMT
Server: nginx/1.10.2
Vary: Accept-Language, Cookie
X-Frame-Options: SAMEORIGIN
```

```
{
    "status": "destroy was scheduled"
}
```

List of resources

Projects list

To get a list of projects send a GET request to **/api/projects/** and add a customer parameters as a filter. Use customer uuid to filter projects by customer.

```
http -v GET
https://waldur.example.com/api/projects/?customer=53dee275177d4b878489ea
ae4c069022 Authorization:"token
a2eef51ca3c290bdd0b5d9e760b767e9f5d71f87"
GET /api/projects/?customer=53dee275177d4b878489eaae4c069022 HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Authorization: token a2eef51ca3c290bdd0b5d9e760b767e9f5d71f87
Connection: keep-alive
Host: waldur.example.com
User-Agent: HTTPie/0.9.8



HTTP/1.0 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Date: Wed, 15 Mar 2017 11:49:58 GMT
Link:
<https://waldur.example.com/api/projects/?customer=53dee275177d4b878489e
aae4c069022>; rel="first",
<https://waldur.example.com/api/projects/?customer=53dee275177d4b878489e
aae4c069022>; rel="last"
Server: WSGIServer/0.1 Python/2.7.13
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
X-Result-Count: 1

[
    {
        "created": "2016-12-30T10:30:53.601417Z",
        "customer":
"https://waldur.example.com/api/customers/53dee275177d4b878489eaae4c0690
22/",
        "customer_abbreviation": "Abbrevi8",
        "customer_name": "Full Name",
        "customer_native_name": "",
```

```
        "customer_uuid": "53dee275177d4b878489eaae4c069022",
        "description": "",
        "name": "OpenStackProject",
        "price_estimate": {
            "limit": -1.0,
            "threshold": 0.0,
            "total": 0.0
        },
        "quotas": [
            {
                "limit": -1.0,
                "name": "nc_private_cloud_count",
                "url":
"https://waldur.example.com/api/quotas/06e99406578c428880f91e832d2e25da/
",
                "usage": 2.0,
                "uuid": "06e99406578c428880f91e832d2e25da"
            },
        ],
        "services": [
            {
                "name": "New name",
                "service_project_link_url":
"https://waldur.example.com/api/openstack-service-project-link/1/",
                "settings":
"https://waldur.example.com/api/service-settings/000f6765bab946a9b9a24e7
36014fa7c/",
                "settings_uuid": "000f6765bab946a9b9a24e736014fa7c",
                "shared": true,
                "state": "OK",
                "type": "OpenStack",
                "url":
"https://waldur.example.com/api/openstack/341f793509e34bec806037a1fd7493
38/",
                "uuid": "341f793509e34bec806037a1fd749338"
            }
        ],
        "url":
"https://waldur.example.com/api/projects/77c23dbcb94a4ad5bba54d82c17fa86
2/",
```

```
        "uuid": "77c23dbcb94a4ad5bba54d82c17fa862"
    }
]
```

Instances list

To get a list of instances send a GET request to **/api/openstacktenant-instances/** and filter them by required parameters, for instance by a project or a customer.

```
http -v GET
https://waldur.example.com/api/openstacktenant-instances/?customer=53dee
275177d4b878489eaae4c069022 Authorization:"token
a2eef51ca3c290bdd0b5d9e760b767e9f5d71f87"
GET
/api/openstacktenant-instances/?customer=53dee275177d4b878489eaae4c06902
2 HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Authorization: token a2eef51ca3c290bdd0b5d9e760b767e9f5d71f87
Connection: keep-alive
Host: waldur.example.com
User-Agent: HTTPie/0.9.8



HTTP/1.0 200 OK
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Date: Wed, 15 Mar 2017 11:57:53 GMT
Link:
<https://waldur.example.com/api/openstacktenant-instances/?customer=53de
e275177d4b878489eaae4c069022>; rel="first",
<https://waldur.example.com/api/openstacktenant-instances/?customer=53de
e275177d4b878489eaae4c069022>; rel="last"
Server: WSGIServer/0.1 Python/2.7.13
Vary: Accept, Cookie
X-Frame-Options: SAMEORIGIN
X-Result-Count: 1

[
    {
        "access_url": [
            "192.168.42.11"
        ],
        "action": "",
        "action_details": {},
        "backend_id": "adb3be5f-02d9-4133-9d5c-17847a1e8152",
```

```
        "cores": 1,
        "created": "2017-03-14T08:05:24.389919Z",
        "customer":
"https://waldur.example.com/api/customers/53dee275177d4b878489eaae4c0690
22/",
        "customer_abbreviation": "Abbrevi8",
        "customer_name": "Full Name",
        "customer_native_name": "",
        "description": "",
        "disk": 2048,
        "error_message": "",
        "external_ips": [],
        "flavor_disk": 8192,
        "flavor_name": "m1.tiny",
        "floating_ips": [],
        "image_name": "TestVM",
        "internal_ips": [
            "192.168.42.11"
        ],
        "internal_ips_set": [
        ],
        "key_fingerprint": "",
        "key_name": "",
        "latitude": 59.4339,
        "longitude": 24.7281,
        "min_disk": 0,
        "min_ram": 64,
        "monitoring_items": null,
        "name": "New instance #43",
        "project":
"https://waldur.example.com/api/projects/77c23dbcb94a4ad5bba54d82c17fa86
2/",
        "project_name": "OpenStackProject",
        "project_uuid": "77c23dbcb94a4ad5bba54d82c17fa862",
        "ram": 512,
        "resource_type": "OpenStackTenant.Instance",
        "runtime_state": "SHUTOFF",
        "security_groups": [],
        "service":
"https://waldur.example.com/api/openstacktenant/b2045458714e4ba19da129ee
c780ec73/",
        "service_name": "Tenant #43",
        "service_project_link":
"https://waldur.example.com/api/openstacktenant-service-project-link/9/"
,
        "service_settings":
"https://waldur.example.com/api/service-settings/dcae83636530464d8fa50c8
3c8334a6c/",
        "service_settings_error_message": "",
        "service_settings_state": "OK",
```

```
        "service_settings_uuid": "dcae83636530464d8fa50c83c8334a6c",
        "service_uuid": "b2045458714e4ba19da129eec780ec73",
        "sla": null,
        "start_time": "2017-03-15T11:00:49.265594Z",
        "state": "OK",
        "tags": [],
        "url":
"https://waldur.example.com/api/openstacktenant-instances/4b5744e2297c47
34842a00c84a33c822/",
        "user_data": "",
        "uuid": "4b5744e2297c4734842a00c84a33c822",
        "volumes": [
            {
                "bootable": true,
                "name": "New instance #43-system",
                "resource_type": "OpenStackTenant.Volume",
                "size": 1024,
                "state": "OK",
                "url":
"https://waldur.example.com/api/openstacktenant-volumes/b5e84a7e0fbf49dd
b32a6d0c2625b0e7/",
                "uuid": "b5e84a7e0fbf49ddb32a6d0c2625b0e7"
            },
            {
                "bootable": false,
                "name": "New instance #43-data",
                "resource_type": "OpenStackTenant.Volume",
                "size": 1024,
                "state": "OK",
                "url":
"https://waldur.example.com/api/openstacktenant-volumes/98c350c6be1b47d7
b3e66fc26119aa68/",
                "uuid": "98c350c6be1b47d7b3e66fc26119aa68"
            }
```

```
            ]
        }
    ]
```

Changelog

Next release

Upgrade guide

- Standard upgrade

Changes

- Nothing yet

Release 3.0.6 - 12.08.2018

Upgrade guide

- Standard upgrade

Changes

- **MasterMind**: Fix conflicting migrations.

Release 3.0.5 - 10.08.2018

Upgrade guide

- Standard upgrade

Changes

- **MasterMind:** Extend/marketplace-service-providers/ with additional fields.
- **OpenStack plugin**: Update backup state in atomic transaction when expired backup is deleted.
- **OpenStack plugin**: Execute deletion of resources created by schedule if limit is reached.

Release 3.0.4 - 9.08.2018

Upgrade guide

- Standard upgrade

Changes

- **OpenStack plugin**: Fix timezone support for OpenStack backup schedules.
- Devops: reduce default MTU for docker-compose to handle deployments in OpenStack (VXLAN) VMs.

Release 3.0.3 - 8.08.2018

Upgrade guide

- Standard upgrade

Changes

- **OpenStack plugin**: add support for OpenStack VIPs in Floating IP network.

Release 3.0.2 - 7.08.2018

Upgrade guide

- Standard upgrade

Changes

- **OpenStack plugin**: add support for OpenStack VIPs in Internal network.

Release 3.0.1 - 5.08.2018

Upgrade guide

- Standard upgrade

Changes

- **HomePort**: Financial export now exports total cost instead of current cost for each customer.
- **HomePort**: Fixed issue with exporting to Excel format.
- **HomePort**: More flexible parsing of sections for Oracle IaaS snapshots.
- **HomePort**: Fix too many lookups when creating resources via HomePort.
- **MasterMind**: Fix display of OfferingItem in Invoices. Use offering name instead of project name.
- **OpenStack plugin**: Extend /admin/openstack_tenant/instance with search and filtering.

Release 3.0.0 - 1.08.2018

Upgrade guide

- Standard upgrade

Changes

- **Mastermind**: Mark JIRA user as disabled instead of deleting him to replicate upstream behaviour.
- **Mastermind**: Protect issue from deletion when support user is deleted.
- **Mastermind**: In SAML2 authentication plugin bump cryptography version so that it supports sign methods to RSA keys.
- **Mastermind**: Expose affected instance type and ID in error message task callback.
- **Mastermind**: Allow to provision OpenStack instance with different subnets within the same network.

Please see Pre-3.0.0 for older releases.

Pre-3.0.0

Release 2.9.9 - 30.07.2018

- **Upgrade guide**
    - Standard upgrade

Changes

- **Mastermind**: Prohibit deletion of Issues if there are connected offerings.
- **Mastermind**: Fix hook processing task to avoid exception raising.
- **Mastermind**: Order marketplace categories by name when displaying in a landing page.
- **Mastermind**: Bump pysaml to latest release to fix security issues.
- **Mastermind**: Enforce unique address for floating IP in OpenStack plugin.

Release 2.9.8 - 25.07.2018

Upgrade guide

New quota for number of order items per marketplace offering has been introduced. Please init it by executing following command line utility: **waldur recalculatequotas**

Changes

- **Ansible module**: Don't wait for floating IP if it is not specified after instance is created.
- **Ansible module**: Avoid API token leakage to output log.
- **Ansible module**: Add documentation for running playbook using virtual Python environment.
- **Ansible module**: Allow to update security groups of the OpenStack instance.
- **Ansible module**: Filter resource by name exactly.
- **Ansible module**: Allow to update security group rules.
- **Ansible module**: Add wheel metadata.
- **Ansible module**: Run unit tests for VirtEngine-Waldur Ansible module using Python 2 & Python 3 via tox.
- **Core**: Fix offering report update action.
- **Core**: Skip CSRF verification for browsable REST API views in debug mode.
- **Core**: Allow to filter project by exact name. Extract name filterset.
- **Core**: Replace custom JSON field with built-in DRF field.
- **Core**: Use immutable default value for JSON field.
- **Core**: Order service settings by primary key to avoid duplicates.
- **Core**: Add caveat about standalone and editable package installation.
- **Core**: Fix wheel deployment configuration.
- **Core**: Fix filtering SSH public key by user name in admin form.
- **Core**: Fix cost tracking test for PostgreSQL: use non-strict decimal comparison.
- **HomePort**: Implement marketplace orders list.
- **HomePort**: Allow to select value from list in offering creation form.
- **HomePort**: Connect frontend with backend for order creation.
- **HomePort**: Allow to upload offering logo.
- **HomePort**: Allow attaching files when creating a new issue.
- **HomePort**: Implement mockup for marketplace offering price form.
- **HomePort**: Fix SSH public key events formatter.
- **HomePort**: Extend support issues with templates.
- **HomePort**: Format units in resource usage.
- **HomePort**: Implement support for SAML2 HTTP POST binding.
- **HomePort**: Fix customer creation dialog closing action.
- **HomePort**: Connect frontend with backend for service offering registration.
- **HomePort**: Implement vendor offerings list.
- **HomePort**: In analytics render VPC quotas instead of project quotas.
- **HomePort**: Convert GBs to TBs in dashboard gauges.
- **HomePort**: Extend organization registration popup in HomePort with roles.
- **HomePort**: Restore removed service for issue comments.
- **HomePort**: Fetch marketplace offerings from backend in marketplace landing page.
- **HomePort**: Cover financial overview component with integration tests using Cypress framework.
- **HomePort**: VirtEngine-Waldur quote notification link should route to a proposal directly.

- **HomePort**: Conceal terms of service consent if ToS is not provided.
- **HomePort**: Users table export should contain "organization" and "organization owner" list.
- **HomePort**: Update Azerbaijani locale.
- **Mastermind**: all backend plugins have been merged into a waldur-mastermind repository.
- **MasterMind**: Implement critical notifications.
- **MasterMind**: Add obsoletes section in RPM spec to uninstall plugin packages.
- **MasterMind**: Introduce attribute option model in marketplace application.
- **MasterMind**: Expose order item count in offering serializer.
- **MasterMind**: Implement bulk create for order items.
- **MasterMind**: Skip invoice item processing if it does not have project.
- **MasterMind**: Make issue type consistent with frontend.
- **MasterMind**: Implement issue template model.
- **MasterMind**: Allow to specify customer instead of provider when offering is created.
- **MasterMind**: Allow to filter marketplace offering by organization.
- **MasterMind**: Inject is_service_provider field to the customer serializer.
- **MasterMind**: Improve order item validation: ensure that offerig is active.
- **MasterMind**: Replace custom JSON field with built-in DRF field.
- **MasterMind**: Implement backend support for shopping order.
- **MasterMind**: Expose sections and attributes in category serializer.
- **MasterMind**: Expose thumbnail for marketplace offering. Implement validation for rating field.
- **MasterMind**: Enable coverage reporting.
- **MasterMind**: Implement test runner using docker compose.
- **MasterMind**: Implement l10n for marketplace.
- **MasterMind**: Add filtering marketplace offerings by attributes.
- **SAML2 plugin**: Implement support for HTTP POST binding.

Release 2.9.7 - 25.06.2018 (bugfix release)

Upgrade guide

- Standard upgrade

Changes

- **Mastermind**: Fix database migration for invoices application.

Release 2.9.6 - 25.06.2018

Upgrade guide

- ENABLE_ACCOUNTING_START_DATE configuration parameter has been moved from WALDUR_INVOICES to WALDUR_CORE. Please update Ansible templates for configuration management appropriately.

Changes

- **Ansible module**: Implement Ansible module to attach/detach volumes from OpenStack VM's via VirtEngine-Waldur REST API.
- **Ansible module**: Add support for deletion of VMs via waldur Ansible module.
- **Ansible module**: Implement Ansible module for OpenStack snapshot operations via VirtEngine-Waldur REST API.
- **Ansible module**: Implement Ansible module for OpenStack volume operations via VirtEngine-Waldur REST API.
- **Ansible module**: It should be possible to mark that floating IP and volumes should not be removed.
- **AWS plugin**: Make service project link optional for resource provisioning.
- **Azure plugin**: Make service project link optional for instance provisioning.
- **Core**: Fix background service synchronization.
- **Core**: Move AccountingStartDateFilter from invoices to structure.
- **Core**: Fix DRF API rendering: make request context optional.
- **Core**: In resource provision API allow to specify service settings and project instead of SPL.
- **Core**: Migrate customer view set to ordering filter backend to make it extendable.
- **Core**: Add ordering by columns in support / financial overview.
- **Core**: Write developer documentation for implementing serializers.
- **DigitalOcean** plugin: Make service project link optional for droplet provisioning.
- **HomePort**: Render loading placeholder for initial screen.
- **HomePort**: Add Azerbaijani locale.
- **HomePort**: Add ordering by columns in support / financial overview.
- **HomePort**: Render error message in resource details if runtime state is ERRED.
- **HomePort**: Allow to order customer by estimated cost.

- **HomePort**: Implement AwesomeCheckbox component.
- **HomePort**: Add developer docs for resource action configuration in HomePort.
- **HomePort**: Hide actions button in user events list rendered inside of modal dialog.
- **JIRA plugin**: Make service project link optional for project provisioning.
- **MasterMind**: Move AccountingStartDateFilter from invoices to structure.
- **MasterMind**: Allow to register service offering for service provider.
- **OpenStack plugin**: Make service project link optional for resource provisioning.
- **Rijkscloud plugin**: Make service project link optional for instance and volume provisioning.
- **SLURM plugin**: Make service project link optional for allocation provisioning.
- **Zabbix plugin**: Make service project link optional for IT service and host provisioning.

Release 2.9.5 - 8.06.2018

Upgrade guide

- Standard upgrade

Changes

- **FreeIPA plugin**: Add GECOS synchronization. Add email and phone number in GECOS param.
- **FreeIPA plugin**: Set First name and last name when mapping to FreeIPA.
- **HomePort**: Implement integration tests for workspace selector and user workspace.
- **HomePort**: Fix table export to CSV, XLS, PDF.
- **HomePort**: Fix table export for financial overview component.
- **HomePort**: Make periodic REST API scheduler more robust.
- **HomePort**: Implement support for ordering result in React table component.
- **HomePort**: Render Oracle report as separate tab in resources sidebar.
- **HomePort**: Extract Oracle snapshots as a separate tab under request details.
- **HomePort**: Fix offering deletion: cleanup service cache after instance is deleted.
- **MasterMind**: Allow to filter request based offering by multiple types simultaneously.
- **MasterMind**: Remove terminated offerings if expiration date has already passed.
- **MasterMind**: Allow staff to delete request based offering. Implement state validation for offering deletion.
- **MasterMind**: Cover VirtEngine-Waldur support plugin with backend unit tests to prevent regressions.
- **MasterMind**: Allow to register customer as a service provider.
- **Openstack plugin**: Fix leftover floating IP, subnet and security groups after deletion of resources.

Release 2.9.4 - 1.06.2018

Upgrade guide

- Standard upgrade

Changes

- **Core**: Fix event context processor.
- **Core**: Pass user which made invitation to the role signal handlers.
- **Core**: Validate that user which has granted/revoked is being passed to log emitter.
- **Core**: Extend customer and project event loggers: allow to override current user.
- **Core**: Emit error when user tries to assign multiple roles to the same user in the same project or organization via admin site.
- **Core**: Expose WALDUR_CORE.ONLY_STAFF_MANAGES_SERVICES configuration via public endpoint.
- **HomePort**: Implement marketplace product details page.
- **HomePort**: Implementation marketplace products list.
- **HomePort**: Fix OpenStack security group rule editor. Ensure that to_port is greater than or equal to from_port.
- **HomePort**: Implement flavor and image popularity table.
- **HomePort**: Fix financial overview column visibility toggle.
- **HomePort**: Fix expert requests list rendering.
- **HomePort**: Fix service store provision dialog: skip extra queries for choices loader.
- **HomePort**: In support workspace move user events log from experimental to stable.
- **HomePort**: Fix comments tab in expert request details view.
- **HomePort**: Migrate event list to ReactJS.
- **HomePort**: Allow to skip connection to external network in VPC provision dialog.
- **HomePort**: Implement asynchronous loading of the flowmap layer for leaflet.
- **HomePort**: Remove leaflet, esri-leaflet and countries.geo.js from the application bundle.
- **JIRA plugin**: Implement async project import.
- **MasterMind**: Fix issue creation typo, add unit test to avoid regression.

- **OpenStack plugin**: Fix security group rule validation.
- **OpenStack plugin**: Make check for existing external network optional for staff-limited tenant creation.
- **OpenStack plugin**: Fix cmd2 requirement: temporarily stick to Python 2.
- **OpenStack plugin**: Updating OpenStack tenant name should lead to update of a provider name.
- **PayPal plugin**: Raise error when user tries to create empty PayPal invoice.
- **Rijkscloud plugin**: Fix network synchronization: gateway_ip may be list or string.
- **Rijkscloud plugin**: Move float field to interfaces field in order to comply with new Swagger schema for Rijkscloud API.
- **Rijkscloud plugin**: Validate floating IP when instance is created.
- **Rijkscloud plugin**: Optimistically mark internal and floating IP as not available.
- **Rijkscloud plugin**: Pull internal and floating IPs after instance is deleted. Otherwise internal and floating IPs remain not available.
- **Rijkscloud plugin**: Order flavors by cores and RAM in ascending order.

Release 2.9.3 - 22.05.2018

Upgrade guide

- Standard upgrade

Changes

- **Core**: Fix error in class based task used as link_error.
- **HomePort**: In resource import dialog fix selection handler: skip duplicate entries.
- **HomePort**: In resource import invalidate cache for importable resources service when import action is executed.
- **HomePort**: In OpenStack Tenant summary skip package rendering if template is not provided.
- **HomePort**: Do not allow to select floating IPs for new VMs that are already linked with internal IPs.
- **HomePort**: Linkify URL even if JIRA markup is not used explicitly.
- **HomePort**: In JIRA issues table fix filter formatting.
- **HomePort**: In marketplace module implement shopping cart components.
- **HomePort**: Skip calling onBlur so that table isn't refreshed when user clicks on details button.
- **HomePort**: Improve layout of resource summary button.
- **HomePort**: Fix financial overview column visibility toggle.
- **HomePort**: Expose organization field in Support/Users table.
- **HomePort**: Allow to resend expired invitations.
- **HomePort**: Implement mockup for a flavor and image popularity feature.
- **JIRA plugin**: Skip authentication error parsing if header is not available.
- **MasterMind**: Ensure that JIRA user is created so that it can be specified in Request participants field when issue is created.
- **OpenStack plugin**: Implement flavor and image popularity endpoints.
- **OpenStack plugin**: Enforce (ID, SPL) uniqueness constraints on tenant, instance, volume and snapshots.
- **OpenStack plugin**: Ensure that duplicate resources are not created during resource import process.
- **OpenStack plugin**: Pull internal IPs even if they are not connected to any instance yet.
- **OpenStack plugin**: Allow nullable instance field for internal IP model.
- **PayPal plugin:** Skip invoice creation if total cost is zero.
- **Zabbix plugin**: Extend Zabbix trigger_status to support pagination.

Release 2.9.2 - 14.05.2018

Upgrade guide

- Standard upgrade

Changes

- **Core**: Added event about owners profile changes that includes diff of changes. Can be used for operator notifications.
- **Core**: Update external network of a provider if Tenant details have been updated.
- **Core**: Extend user views with role details and filters.
- **Core**: Drop organization and organization_approved from user change form.
- **Core & all backend**: Migrate to Celery 4.0
- **Jira plugin**: Add support for importing projects.
- **HomePort**: Move detection of actions for OpenStack Instances to frontend for speeding up UI.
- **HomePort**: Migrate user AuditLog to ReactJS
- **HomePort**: Extend Resource usage with VM usage across projects.
- **HomePort**: Fix "action already in progress" exception in submit button directive.
- **HomePort**: Marketplace initial mocks in experimental mode.
- **HomePort**: Fix column visibility in financial overview tables.
- **HomePort**: Migrate event & alert formatter services to ReactJS.

- **HomePort**: Add SAML2 authentication, offerings and OpenStack package event types.
- **HomePort**: Drop alert feature.
- **HomePort**: Enable user management from support workspace for staff.
- **HomePort**: Remove size column in VM import view.
- **HomePort**: Hide details button in VM import view.
- **HomePort**: Move detection of actions for OpenStack Tenants to frontend for speeding up UI.
- **HomePort**: Mocks for map-based information display.
- **HomePort**: Fix display of user registration method in user details.
- **HomePort**: Display summary of request-based item in its details view.
- **HomePort**: Add VM and volume import, VM-based management for Dutch government cloud.
- **HomePort**: Add role filter to user view.
- **HomePort**: Extend user list with new fields, filters and fix list export functionality.
- **OpenStack plugin**: Internal IP synchronisation improvements: fixed race condition, exposed in /admin, better error and warning messages.
- **SAML2 plugin**: Fix SAML2 logout if session is already expired.
- **Zabbix plugin**: Expose X-Result-Count for trigger_status view.
- **Zabbix plugin**: Extend Zabbix trigger_status to support pagination.

Release 2.9.1 - 9.04.2018

Upgrade guide

- **WALDUR_ANSIBLE** configuration parameters were renamed to **WALDUR_PLAYBOOK_JOBS**.

  All configuration parameters from **WALDUR_ANSIBLE** (now **WALDUR_PLAYBOOK_JOBS**) except for PLAYBOOKS_DIR_NAME and PLAYBOOK_ICON_SIZE were moved to **WALDUR_ANSIBLE_COMMON**.

  New configuration parameters:

WALDUR_PYTHON_MANAGEMENT	
SYNC_PIP_PACKAGES_TASK_ENABLED	Flag to enable execution of a batch task which indexes PyPi repository and p libraries in the DB
SYNC_PIP_PACKAGES_BATCH_SIZE	Configures how many libraries may batch task insert in one INSERT query.
PYTHON_MANAGEMENT_PLAYBOOKS_DIRECTORY	Directory where Python management playbooks are located. By default, formed using WALDUR_ANSIBLE_COMMON.ANSIBLE_LIBRARY/waldur-apps/python_m
WALDUR_JUPYTER_HUB_MANAGEMENT	
JUPYTER_MANAGEMENT_PLAYBOOKS_DIRECTORY	Directory where JupyterHub management playbooks are located. By default, formed using WALDUR_ANSIBLE_COMMON.ANSIBLE_LIBRARY/waldur-apps/jupy

Changes

- **Core**: Ensure that quota usage is not negative value.
- **Core**: Fix volume and snapshot total size quotas.
- **Core**: Drop support for Python 2.4.
- **Core**: Make username and password fields optional instead of mandatory.
- **Core**: Fix Python 3 issues for VirtEngine-Waldur Core.
- **Core**: Squash database migrations in VirtEngine-Waldur Core.
- **Core**: Eliminate duplicates in get_quota_ancestors method.
- **Core**: Add time zone format in email notifications from VirtEngine-Waldur.
- **Core**: SHOW_ALL_USERS default is False,
- **Core**: Refactor filter_queryset_for_user: reduce cyclomatic complexity, drop unused options.
- **HomePort**: Add ASCII name validator to Storage and Snapshot names.
- **HomePort**: Enforce ASCII name when resource is renamed.
- **HomePort**: Implement SSH public key selector field.
- **HomePort**: Toogle visibility of price estimation columns dynamically.
- **HomePort**: Implement quota filters for batch processing, VPC and OpenStack resource usage.
- **HomePort**: Use abbreviation instead of name in selector for mobile view if abbreviation provided.
- **HomePort**: Add treemap under support workspace for displaying VM resource usage.
- **HomePort**: Allow to customize header in resource details page.
- **HomePort**: Render job submitting instruction for MOAB.

- **MasterMind**: Do not add Request-based items to invoices that are not in OK state. Remove invoice items connected to such request-based items.
- **MasterMind**: Include pending expert request in customer workspace counter.
- **MasterMind**: Send notification for expert if added new comment.
- **MasterMind**: Update price estimate when invoice is created.
- **MasterMind**: Validate offering report as list of sections.
- **MasterMind**: Allow to filter offering by issue key.
- **MasterMind**: Refactor mastermind/support.
- **MasterMind**: Use unicode strings for price estimate event message.
- **OpenStack plugin**: Remove unused serializers.
- **OpenStack plugin**: Fix recalculatequotas management command.
- **OpenStack plugin**: Implement aggegate usage quotas for VPC.
- **OpenStack plugin**: Introduce project and customer quotas for OpenStack CPU, RAM and HDD usage.
- **SAML2 plugin**: Remove CSRF validation from login endpoints.

Release 2.9.0 - 24.03.2018

Upgrade guide

- Change WALDUR_ANSIBLE to WALDUR_PLAYBOOK_JOBS in override.conf.py.
- Override WALDUR_AUTH_VALIMO.MOBILE_PREFIX in override.conf.py with values for phone prefix of the country of deployment.

Changes

- **AWS plugin**: Move PollRuntimeStateTask and PollBackendCheckTask from plugins to core.
- **Azure plugin**: Move PollRuntimeStateTask and PollBackendCheckTask from plugins to core.
- **Core**: Validate SSH public key using Python cryptography library.
- **Core**: Move JIRA and Zabbix models under Application category.
- **Core**: Move PollRuntimeStateTask and PollBackendCheckTask from plugins to core.
- **Core:** Allow to filter Resources by backend id (?backend_id=...).
- **Core:** Expose provider UUID in /admin.
- **Core**: Enable Defender for password lockout on invalid attempts.
- **HomePort**: Drop deprecated resources from the category list.
- **HomePort:** Enable "Assign package to OpenStack Tenant" action for staff role to link imported Tenant with a Package.
- **HomePort:** Port user-edit component from AngularJS to React.
- **HomePort:** Expose project type in project details.
- **HomePort:** Add default phone prefix for Valimo logins.
- **HomePort:** Expose volume and snapshot sizes in Project and Organization dashboards.
- **HomePort:** Added refresh buttons to issues and comments to allow refreshing without page reload.
- **HomePort:** Ported issue comments to React.
- **HomePort:** Changed "Request name" to "Title" for support tickets to match with backend name.
- **HomePort**: Limit OpenStack and VM names to ASCII + first letter pattern.
- **HomePort**: Fix "Release floating IP" action to actually do as user requests.
- **HomePort**: Implement button and dialog for price estimate details in billing report.
- **HomePort**: Implement button and dialog for request-based offering report.
- **MasterMind**: Fix OpenStack tenant delete action in admin site.
- **MasterMind**: Add waldur-jira dependency. VirtEngine-Waldur-Jira is a plugin aimed at managing remote Jiras.
- **MasterMind**: Implement support for report field in request-based service.
- **SLURM:** Introduce basic support for MOAB-based clusters.
- **OpenStack plugin**: Move PollRuntimeStateTask and PollBackendCheckTask from plugins to core.
- **OpenStack plugin**: Expose SSH public key details for debug if import to OpenStack fails.

Release 2.8.9 - 26.02.2018

Upgrade guide

- Previously all authentication methods have been enabled by default. Currently only local sign in is enabled by default. All other authentication methods, including SAML2, OpenID, Facebook and Google OAuth2, Valimo, are disabled by default. In order to enable them back, you should tweak WALDUR_CORE['AUTHENTICATION_METHODS'] configuration variable and ensure that it contains the following strings: 'SAML2', 'ESTONIAN_ID', 'SOCIAL_SIGNUP', 'LOCAL_SIGNUP'.
- Previously configuration for authentication methods have been stored spread between MasterMind and HomePort. Currently all configuration for authentication methods have been moved from HomePort to MasterMind. Therefore. you should update your Ansible scripts to match new configuration schema.

Old HomePort configuration	New MasterMind configuration
authenticationMethods	WALDUR_CORE.AUTHENTICATION_METHODS
facebookClientId	WALDUR_AUTH_SOCIAL.FACEBOOK_CLIENT_ID
googleClientId	WALDUR_AUTH_SOCIAL.GOOGLE_CLIENT_ID
smartIdClientId	WALDUR_AUTH_SOCIAL.SMARTIDEE_CLIENT_ID
VALIMO_LABEL	WALDUR_AUTH_VALIMO.LABEL
SAML2_IDENTITY_PROVIDER	WALDUR_AUTH_SAML2.IDENTITY_PROVIDER_URL
SAML2_LABEL	WALDUR_AUTH_SAML2.IDENTITY_PROVIDER_LABEL
allowToSelectSAML2Provider	WALDUR_AUTH_SAML2.ALLOW_TO_SELECT_IDENTITY_PROVIDER
estoniaIdLogoutUrl	WALDUR_AUTH_OPENID.LOGOUT_URL

Changes

- **Core:** Implement view decorator to validate enabled authentication method.
- **Core:** Make resources endpoint filtering consistent with Resource model. Exclude subresources from resources endpoint.
- **Core**: Allow to hide native name in customer and user admin form.
- **Core:** Add LooseMultipleChoiceFilter for filtering by multiple values based on backend data.
- **HomePort**: Fix formatCrontab filter.
- **HomePort**: Use multiplication factor for invoice item by default.
- **HomePort**: Conditionally render native name for user and organization in create and detail views.
- **HomePort**: Move authentication methods configuration from HomePort to MasterMind.
- **Jira plugin**: Support new webhook API. Expose Time to resolution SLA field for Jira issues.
- **Jira plugin:** Support of filtering by issue statuses using LooseMultipleChoiceFilter.
- **MasterMind**: Expose mutliplication factor for invoice item.
- **OpenID plugin**: Allow to disable OpenID authentication.
- **OpenStack plugin:** Exclude floating IPs without internal IP for fetching instance.
- **SAML2 plugin**: Allow to disable SAML2 authentication.
- **Valimo plugin**: Allow to disable Valimo authentication.
- **Zabbix plugin**: Allow to specify exact list of priorities for Zabbix trigger filtering.
- **Zabbix plugin**: Implement support for HEAD request to count number of triggers matching filter.

Release 2.8.8 - 17.02.2018

Upgrade guide

- Callback URL templates have been moved from SAML2 and OpenID plugins to Core. Please update your configuration accordingly.

Changes

- **Core**: Add support for using password-protected Redis servers.
- **Core**: Limit uploaded certificate files for ServiceSettings to 'pem'-files only.
- **HomePort**: Error message is not visible in create bid dialog.
- **HomePort**: Implement users list view in support workspace.
- **HomePort**: Support optional Single-Logout for SAML authentications.
- **HomePort**: Implement form field for OpenStack tenant selection.
- **HomePort**: Prevent redirect to profile page on logout and failed login.
- **HomePort**: Fix display of escaped symbols in request-based item detail view.
- **HomePort**: For request-based items, add support for validating min and max for integer fields.
- **HomePort**: Implement HTML rendering of strings in tooltips in expert request customer list.
- **HomePort**: In financial overview make it clear when "accounting is running / not running" filters have not yet been applied.
- **HomePort**: Migrate user details component from AngularJS to ReactJS.
- **HomePort**: Add "agree with terms of services" for request-based-items.
- **HomePort**: Render empty placeholder if access_url contains only invalid items.
- **Jira plugin**: Add support for filtering of issues by multiple statuses and priorities.
- **OpenID plugin**: Move callback URL templates to Core.
- **OpenStack plugin**: Make tenant_name optional field because default value is already available.
- **OpenStack plugin:** Fix issue when an instance was stealing existing instance's IP on creation of floating IP update.
- **SAML2 plugin**: Support optional Single-Logout for SAML authentications.

- **Zabbix plugin**: Implement Zabbix triggers filtering by host, changed date and priority.

Release 2.8.7 - 11.02.2018

Upgrade guide

- Standard procedure

Changes

- **Core**: start day of accounting should not be read-only for new organization.
- **Core**: in service settings admin form add validation of required extra options.
- **Core**: introduce SITE_NAME configuration parameter.
- **HomePort**: implement modal dialog for user details in support workspace.
- **HomePort**: in create proposal dialog disable 'Create' button if there are no experts in the team yet.
- **HomePort**: render current flavor in OpenStack instance flavor change form.
- **HomePort**: exclude current flavor from list of available flavors so that valid data is submitted.
- **HomePort**: implement support for select string field.
- **HomePort**: move objectives and planned budget fields to details tab.
- **HomePort**: remove support for description field in expert request create and details components.
- **HomePort**: fix Slurm detail display regression
- **Jira plugin**: expose additional Jira fields for filtration.
- **MasterMind**: fix expert provider serialization.
- **MasterMind**: remove description field from expert request serializer.
- **MasterMind**: make backend_id for Jira attachment model unique and non-blank.
- **MasterMind**: use JIRAError except instead of Python Requests exception so that error is logged correctly.
- **MasterMind**: move site name configuration to VirtEngine-Waldur Core.
- **MasterMind**: fix issue with attachment synchronization caused by race conditions.
- **MasterMind**: fix comment prefix when VirtEngine-Waldur user is not associated with Jira author yet.
- **OpenStack plugin**: revert downgrading of admin role to member for shared privileged user.
- **SAML2 plugin**: allow overriding NameID format.
- **SAML2 plugin**: allow to specify signature and digest algorithm.

Release 2.8.6 - 4.02.2018

Upgrade guide

- Standard procedure

Changes

- **Core:** fix duplicate logs from waldur_core applications.
- **Core:** implement filtering for customer and project users
- **Core**: introduce country_name field. Make country field optional.
- **Core**: expose extra fields for payment details in the customer serializer.
- **Core**: expose company types enumeration as public config for organization creation wizard.
- **Core**: validate mandatory fields during service settings creation.
- **Core**: implement support for invitation management in admin site.
- **HomePort**: migrate expert request summary to ReactJS.
- **HomePort**: render links to organization and project from expert request details page.
- **HomePort**: make account activation link in email message clickable.
- **HomePort**: expose provider name in SLURM allocation list.
- **HomePort**: render links to organization and project from expert request details page.
- **HomePort**: ensure that recurring_billing in expert configuration is respected.
- **HomePort**: skip offering rendering in service store selector if feature is disabled.
- **HomePort**: implement frontend support for issue attachments.
- **HomePort**: add new fields to organization creation dialog.
- **HomePort**: migrate customer details components from AngularJS to ReactJS.
- **HomePort**: improve error message for Valimo authentication.
- **HomePort**: disable invalid templates in change VPC package dialog instead of hiding them.
- **HomePort**: render current flavor name in virtual machine summary component.
- **HomePort**: implement users list view in support workspace.
- **HomePort**: implement frontend support for Zabbix host creation and deletion.
- **MasterMind**: switch from EPEL provided package for pdfkit to our own.
- **MasterMind**: fix JIRA attachment download. Use authenticated session instead of anonymous request.

- **MasterMind**: render Zabbix host for OpenStack virtual machine.
- **MasterMind**: render value for current cost even if another period is required.
- **MasterMind**: make customer details serialization format consistent across all plugins.
- **MasterMind**: use customer data instead of payment details for details serialization.
- **MasterMind**: implement filter and serializer for Zabbix and OpenStack integration.
- **MasterMind**: in VPC package template, it shouldn't be possible to set quota amounts to 0 (vcpu, ram, storage). Also set default value of those fields to 1.
- **MasterMind**: implement attachment synchronization for JIRA.
- **MasterMind**: merge organization details and payment details.
- **MasterMind**: skip notification if issue just has been created on backend.
- **OpenStack plugin**: add support for Organization-specific shared OpenStack networks.
- **OpenStack plugin**: add an opt-in cleanup of floating IP when removing instance.
- **OpenStack plugin**: fix default price list item synchronization for OpenStack flavor.
- **OpenStack plugin**: convert query set for internal and external IPs to plain list.
- **OpenStack plugin:** allow to use floating IP with both ACTIVE and DOWN status.
- **OpenStack plugin**: check whether floating IP is already assigned to instance when VM is created or updated.
- **PayPal plugin**: preserve usage days instead of converting them to hours.
- **SAML2 plugin**: remove URL format validation for selecting SAML IdP.
- **SAML2 plugin**: fallback to entity_id when detecting name of the SAML IdP.
- **Zabbix plugin**: add priority field to Zabbix trigger.
- **Zabbix plugin**: fix network interface IP address serialization.
- **Zabbix plugin**: use JSON editor for network interface parameters in admin site.
- **Zabbix plugin**: fix Zabbix host pull in backend layer.

Release 2.8.5 - 13.01.2018

Upgrade guide

- New quota for number of experts requests per organization has been introduced. Please init it by executing following command line utility: **waldur recalculatequotas**
- New quota for number of expert requests per organization has been introduced. Please init it by executing following command line utility: **waldur update_expert_quota**

Changes

- **Core:** implement transaction support for executor mixin.
- **Core**: expose public settings via /api/configuration/ endpoint.
- **Core**: implement project type model, serializer, filter, viewset and admin.
- **HomePort**: implement Jira issue create dialog.
- **HomePort**: allow to specify parent task for the subtask.
- **HomePort**: fix rendering of long label in sidebar.
- **HomePort**: allow to specify resource for issue.
- **HomePort**: compute and display customer's current cost in financial overview.
- **HomePort**: allow to specify priority for issue.
- **HomePort**: allow to hide sidebar menu items depending on project type.
- **HomePort**: migrate summary components to ReactJS
- **HomePort**: allow to specify project type in provision form.
- **HomePort**: migrate event details component from AngularJS to ReactJS.
- **HomePort**: implement details view for JIRA issue.
- **HomePort**: migrate JIRA issues list to ReactJS.
- **HomePort:** allow to select project in workspace selector after clicking on "Request an expert" button.
- **HomePort:** fix 404 error processing in expert request details view.
- **HomePort:** prevent duplicate table actions by introducing unique action key.
- **HomePort:** rename "Request bids" of experts to "Proposals".
- **HomePort:** migrate create & update provider components from AngularJS to ReactJS.
- **HomePort:** conceal civil code field in initial view if it is not provided.
- **HomePort:** in financial overview render experts quota usage for organization.
- render number of request-based items per project and organization.**HomePort:**
- invalidate security groups cache when new security group is created.**HomePort:**
- set default name for expert request and offering from type label.**HomePort:**
- **HomePort**: download expert contract PDF file via XHR and save it via FileSaver.
- **HomePort**: migrate expert contract details to ReactJS.
- **HomePort:** migrate create & update project components from AngularJS to ReactJS.
- **Jira plugin**: drop support for the impact field.
- **Jira plugin**: allow to specify parent task for the subtask.
- **Jira plugin**: implement transaction support for the CRUD operations.
- **Jira plugin**: synchronize priorities with backend; allow to specify priority via URL.
- **Jira plugin**: in JIRA issue serializer add fields for consistency with resource serializer.

- **Jira plugin**: make create project API more robust by converting email to username for assignee field.
- **MasterMind**: add project and organization columns to the list of packages.
- **MasterMind**: filter in admin should only list shared settings.
- **MasterMind**: introduce quota for number of active expert request per organization.
- **MasterMind**: introduce quotas for number of request-based items per project and organization.
- **MasterMind**: add generation of contract upon agreement acceptance by parties.
- **OpenStack plugin**: expose public settings via /api/configuration/ endpoint.
- **OpenStack plugin**: migrate Glance to v2 of API.

Release 2.8.4 - 22.12.2017

Upgrade guide

- New quota for number of Jira projects per VirtEngine-Waldur project has been introduced. Please init it by executing following command li
  waldur recalculatequotas
- New quota for number of expert requests per project has been introduced. Please init it by executing following command line utility:
  waldur update_expert_quota
- INVOICES setting has been renamed to WALDUR_INVOICES. Please find and fix all occurrences.

Changes

- **Ansible VirtEngine-Waldur module:** allow to specify flavor_min_cpu and flavor_min_ram instead of flavor name or UUID.
- **Core:** move update_pulled_fields function from OpenStack plugin to VirtEngine-Waldur Core.
- **Core:** move widget for rendering storage unit from VirtEngine-Waldur MasterMind to VirtEngine-Waldur Core.
- **HomePort:** allow to register JIRA as provider.
- **HomePort**: introduce onlyStaffManagesServices setting.
- **HomePort**: disable OpenStack tenant modification if only staff manages services.
- **HomePort**: render total customer cost in financial overview.
- **HomePort**: temporarily disable offline-plugin for service workers.
- **HomePort**: describe approach to creating plugins.
- **HomePort**: update momentjs locale globally when language is switched.
- **HomePort**: render error notification if resource action API request has failed.
- **HomePort**: migrate dashboard feed component from AngularJS to ReactJS.
- **HomePort**: allow to list JIRA projects as resources.
- **HomePort**: implement service desk project provision form.
- **HomePort**: setup and configure TSLint.
- **HomePort**: render expert counter in the projects list of the organization workspace.
- **HomePort**: render expert counter in the dashboard charts of the project workspace.
- **HomePort**: render list of JIRA issues as a tab in JIRA project details.
- **HomePort**: fix flavor rendering in cost planning dialog.
- **HomePort**: improve header layout.
- **Jira plugin**: introduce quota for number of JIRA projects managed under VirtEngine-Waldur project.
- **Jira plugin**: allow to filter JIRA project by VirtEngine-Waldur project UUID.
- **Jira plugin**: allow to specify issue scope.
- **Jira plugin**: synchronize JIRA project types on backend, allow to specify JIRA project type in provision form.
- **Jira plugin**: skip resource content type and object ID validation if it is not specified.
- **Jira plugin**: allow to filter JIRA issues by JIRA project UUID or URL.
- **Jira plugin**: pull issue types when JIRA project is created, allow to specify type in the issue provision form.
- **OpenStack plugin:** disable OpenStack tenant modification if only staff manages services.
- **OpenStack plugin:** move update_pulled_fields function from OpenStack plugin to VirtEngine-Waldur Core.
- **OpenStack plugin:** Bump OpenStack client libraries to Pike release.
- **MasterMind**: add support for staff only VPC management.
- **MasterMind**: implement endpoint to calculate total cost of customers matching filters.
- **MasterMind**: allow to create and delete attachments for issues.
- **MasterMind**: implement quota for number of expert requests per project.
- **MasterMind**: implement management command to recalculate experts quota usage for each project.
- **MasterMind**: rename INVOICES settings to WALDUR_INVOICES for consistency.
- **MasterMind**: add dependency for django-auth-ldap.
- **Zabbix plugin:** migrate Zabbix plugin to new architecture.

Release 2.8.3 - 4.12.2017

Upgrade guide

- In order to update SAML provider details cached in DB you execute following management command:
  waldur sync_saml2_providers

This command is executed by celerybeat task every hour anyway, but it's better to call it explicitly after new release is installed from the RPM repository.
You may verify that this command has been executed successfully by examining SAML2 identity providers in Django admin or by logging in via SAML2 using HomePort.

- New quotas for total size of volumes and snapshots per project and customer have been introduced. Please init them by executing following command line utility:
  waldur recalculatequotas
- Please find and replace all usages of **nodeconductor** command line utility with **waldur.**

Changes

- **AWS plugin**: rename nodeconductor_aws to waldur_aws.
- **Azure plugin**: rename nodeconductor_azure to waldur_azure
- **Core**: add volume and snapshot size counter quota fields.
- **Core**: add support for staff-only VPC management.
- **Core**: fix messages formatting with Unicode chars.
- **Core**: rename NodeConductor to VirtEngine-Waldur in developer docs.
- **Core**: Python package for nodeconductor has been renamed to waldur-core.
- **Core**: nodeconductor command line utility has been removed, please use waldur CLI instead.
- **Core**: support for /etc/nodeconductor/nodeconductor_plus.py configuration file has been removed, please use /etc/waldur/override.conf.py instead.
- **Core**: support for /etc/nodeconductor/nodeconductor_saml2.py configuration file has been removed, please use /etc/waldur/saml2/saml2.conf.py instead.
- **Core**: support for NODECONDUCTOR configuration setting has been removed, please use WALDUR_CORE instead.
- **Cost planning plugin**: Python package for nodeconductor_cost_planning has been renamed waldur_cost_planning.
- **DigitalOcean plugin**: Python package for nodeconductor_digitalocean has been renamed waldur_digitalocean.
- **HomePort**: apply image disk size as a minimal value to the system volume size.
- **MasterMind:** add support for ONLY_STAFF_MANAGES_SERVICES setting.
- **MasterMind:** Python package for nodeconductor-saml2 has been renamed to waldur-auth-saml2.
- **OpenStack plugin**: allow to provision OpenStack virtual machine even if image min size is over the flavor disk size.
- **SAML2 plugin**: Python package for nodeconductor-saml2 has been renamed to waldur-auth-saml2.
- **SAML2 plugin**: rename NODECONDUCTOR_SAML2 to WALDUR_AUTH_SAML2.

Release 2.8.2 - 27.11.2017

Upgrade guide

- Standard upgrade procedure.

Changes

- **Auth OpenID plugin:** rename nodeconductor_auth_openid to waldur_auth_openid.
- **Auth OpenID plugin:** rename NODECONDUCTOR_AUTH_OPENID to WALDUR_AUTH_OPENID.
- **Auth OpenID plugin:** rename nodeconductor_auth_openid.auth.NodeConductorOpenIDBackend to waldur_auth_openid.auth.VirtEngine-WaldurOpenIDBackend.
- **Auth Social plugin:** rename nodeconductor_auth_social to waldur_auth_social.
- **Auth Social plugin:** rename NODECONDUCTOR_AUTH_SOCIAL to WALDUR_AUTH_SOCIAL.
- **Core:** implement strict validation for SSH key: it should be single line.
- **Core:** allow to provision virtual machine using shared SSH public key.
- **HomePort:** remove premium support feature as it has been obsoleted by offerings and experts.
- **OpenStack plugin:** fix recalculatequotas command for OpenStack SPL quotas.
- **JIRA plugin**: rename nodeconductor-jira to waldur-jira.
- **JIRA plugin**: update to Django 1.11 and latest VirtEngine-Waldur Core API.
- **JIRA plugin**: squash database migrations.
- **JIRA plugin**: rename JIRA_PRIORITY_MAPPING setting to WALDUR_JIRA['PRIORITY_MAPPING'].
- **JIRA plugin**: rename JIRA_COMMENT_TEMPLATE setting to WALDUR_JIRA['COMMENT_TEMPLATE'].
- **MasterMind**: rename nodeconductor_auth_openid to waldur_auth_openid.
- **MasterMind**: rename nodeconductor_auth_social to waldur_auth_social.

Release 2.8.1 - 21.11.2017

Upgrade guide

- The new version of HomePort removes the "static" prefix from the served static files. If a deployment has an override of a static settings (e.g. favicon, PWA manifest icons), overrides should be adjusted accordingly.

- Please replace value of WALDUR_SUPPORT['ACTIVE_BACKEND'] setting from 'nodeconductor_assembly_waldur.support.backend.atlassian:ServiceDeskBackend' to 'waldur_mastermind.support.backend.atlassian:ServiceDeskBackend' if you're using ServiceDesk and 'nodeconductor_assembly_waldur.support.backend.atlassian:JiraBackend' to 'waldur_mastermind.support.backend.atlassian:JiraBackend' if you're using JIRA.
- Please update configuration setting variables from NODECONDUCTOR_OPENSTACK to WALDUR_OPENSTACK and from NODECONDUCTOR_OPENSTACK_TENANT to WALDUR_OPENSTACK_TENANT

Changes

- **Ansible plugin:** rename nodeconductor-openstack to waldur-openstack.
- **Core:** allow to attach external filter to customers endpoint.
- **Core:** support passing configuration base folder as a parameter in case of containerized deployments.
- **Core:** allow to tweak delta calculation for CounterQuotaField.
- **Core:** allow to decrease quota usage even if result exceeds limit.
- **Cost planning plugin**: resolve circular dependency between cost planning and assembly.
- **Cost planning plugin:** rename nodeconductor-openstack to waldur-openstack.
- **HomePort**: eliminate duplicate vendor code via Webpack DLL.
- **HomePort**: strip static prefix for favicon filename.
- **HomePort**: implement frontend support for Mobile ID authentication via Valimo.
- **HomePort**: improve error processing and layout in user details component.
- **HomePort**: add message to index.html when JavaScript is disabled.
- **HomePort**: refresh project cache after update, add project name validation, prevent project name duplication
- **HomePort:** implement basic formatting of JIRA markup in issue comment.
- **HomePort:** add a filter to financial overview to conceal organization whose accounting is not running.
- **HomePort:** allow to see financial overview and resource list by staff and support users if service desk integration is basic.
- **HomePort:** allow to filter billing price estimate by year and month.
- **HomePort:** make application name optional parameter for price estimation in application provision form.
- **HomePort:** render current customer billing price estimate when PayPal mode is active.
- **MasterMind**: add waldur-auth-valimo dependency.
- **MasterMind**: fix HTML templates for experts and notifications.
- **MasterMind**: strip newlines from email subject for invoice notification.
- **MasterMind**: implement filter for customer by accounting_start_date.
- **MasterMind**: fix SLURM usage reporting on a deployment without SLURM services.
- **MasterMind**: fix SAF accounting report (add DOKNR field, filter invoices by the previous month, display price without taxes).
- **MasterMind**: resolve circular dependency between cost planning and assembly.
- **MasterMind**: rename nodeconductor-assembly-waldur to waldur-mastermind.
- **MasterMind**: attach SAF accounting report to the email as a text file.
- **MasterMind:** rename nodeconductor-openstack to waldur-openstack.
- **MasterMind:** replace internal SSH key with user's SSH key in Ansible estimator.
- **MasterMind:** add ability to select a month of the financial overview.
- **OpenStack plugin:** implement custom quota counters for OpenStack tenant SPL.
- **OpenStack plugin:** synchronize price list item for flavor when it is created.
- **OpenStack plugin:** rename nodeconductor-openstack to waldur-openstack.

Release 2.8.0 - 8.11.2017

Upgrade guide

- Standard upgrade procedure.

Changes

- **Core:** emit event when customer or project permission expiration time has been updated.
- **Core:** allow customer owner to edit project permissions for himself.
- **HomePort**: render events in permission log.
- **HomePort**: as a customer owner I should be able to give myself project level roles.
- **HomePort**: in development mode allow to recompile stylesheet without restarting Webpack.
- **HomePort**: improve quota bar chart for exceeded limits.
- **MasterMind:** SAF export email should put all organizations into one email.
- **MasterMind:** add custom admin action in invoices app to send SAF export email.
- **MasterMind:** in SAF export email skip empty invoices and customers without payment details.
- **MasterMind:** update quotas for related service settings when tenant's package has been changed.
- **MasterMind:** schedule Celery task to send email notification only after database transaction has been completed.
- **SLURM plugin**: schedule Celery task only after database transaction has been completed.
- **Social Auth plugin**: fix Facebook OAuth client.

Release 2.7.9 - 3.11.2017

Upgrade guide

- Standard upgrade procedure.

Changes

- **Ansible plugin:** move Ansible application estimator to assembly.
- **Ansible plugin:** delete related OpenStack instances when Ansible job is deleted.
- **Ansible module**: wait until provisioned OpenStack instance has got at least one external IP.
- **AWS plugin**: add support for recursive deletion of projects with all resources by staff.
- **Azure plugin**: add support for recursive deletion of projects with all resources by staff.
- **Core**: add support for recursive deletion of projects with all resources by staff.
- **Core**: allow to install latest release of python2-urllib3 from OpenStack Pike.
- **Core**: skip tags field in action because it is needed only for resource creation.
- **Core**: fix email hook processor when it is used multiple times subsequently.
- **Core**: use application verbose name in list of installed apps in admin site.
- **DigitalOcean plugin**: add support for recursive deletion of projects with all resources by staff.
- **FreeIPA plugin**: fix SSH public key update when keys are not sorted or not specified.
- **FreeIPA plugin**: expose version information for the VirtEngine-Waldur FreeIPA application.
- **HomePort:** implement Ansible application price estimation and explanation.
- **HomePort**: allow to delete Ansible job.
- **HomePort**: cleanup floating IPs cache after OpenStack instance has been created or deleted.
- **HomePort**: implement support for gettext strings extraction from JSX.
- **HomePort**: drop Grunt.
- **HomePort**: display register link on login view.
- **HomePort**: update minimal Node.js dependency requirement from 4 to 6.
- **HomePort**: improve tooltip placement for expert provider indicator in the workspace selector toggle.
- **HomePort**: add an indicator that bid has been submitted for expert request.
- **HomePort**: disable sorting expert requests by objectives field.
- **HomePort**: fix footer of the login screen on mobile.
- **HomePort**: enable proper minification of JavaScript code.
- **HomePort**: add currency sign to expert contract preview form.
- **HomePort**: comments under request offer should indicate role of a person.
- **HomePort**: fix code style for legacy SCSS code. Use 2 spaces for indent instead of 4.
- **HomePort:** implement client-side validation of customer delete action.
- **HomePort:** add missing Angular injection annotation for arrow function.
- **HomePort:** cleanup cache of expert list after "complete/cancel" is clicked in expert request details.
- **HomePort:** do not use redirect to 404 if a page has not been found.
- **HomePort:** fix header layout on 768px width screen.
- **HomePort:** fix code style. Use the type-safe equality operators.
- **MasterMind:** render extra parameters of expert request using JSON editor.
- **MasterMind:** convert expert request description field from varchar to text field.
- **MasterMind**: move Ansible application estimator to assembly.
- **MasterMind:** implement project cache for expert request and expert contract.
- **MasterMind:** don't delete expert request and expert contract when project is deleted.
- **MasterMind:** send out email notifications to matched expert providers about new requests.
- **MasterMind:** render user roles for expert request.
- **MasterMind:** expose author UUID in issue comment serializer.
- **MasterMind:** disable deletion of customer by non-staff if invoices are present.
- **MasterMind:** apply Inspinia them to email message template for expert contract.
- **MasterMind:** skip notifications about comments added to an issue by caller himself.
- **MasterMind:** add notifications about new bids for customer owners.
- **OpenStack plugin**: add support for recursive deletion of projects with all resources by staff.
- **OpenStack plugin**: schedule floating IP synchronization for shared tenant when instance is created or deleted.
- **OpenStack plugin**: expose version information for the VirtEngine-Waldur OpenStack application.
- **SLURM plugin**: add support for recursive deletion of projects with all resources by staff.
- **SLURM plugin**: expose version information for the VirtEngine-Waldur SLURM application.
- **SLURM plugin**: migrate SLURM usage report client from sreport to sacct.

Release 2.7.8 - 17.10.2017

Upgrade guide

- Standard upgrade procedure.

Changes

- **Ansible plugin:** enable waiting for OpenStack resource creation by default.
- **Core:** apply strict filtering for resource**.**
- **HomePort:** render single bid for expert request details in organization workspace.
- **HomePort:** allow to create and delete expert bid on expert request details page.
- **HomePort:** implement "Request custom flavor" action for OpenStack Tenant.
- **HomePort:** implement generic events filtering for all supported invoice items.
- **HomePort:** apply consistent code style for HTML code.
- **HomePort:** implement data processing for SLURM charts. Render allocation usage & cost per user charts.
- **HomePort:** set form dirty when form control is added or removed so that submit form button is enabled.
- **HomePort:** copy initial form value to editor form so that original form is preserved.
- **MasterMind:** implement expert bid deletion for pending expert request.
- **MasterMind:** log offering creation and deletion.
- **MasterMind:** render generic fields scope_type and scope_uuid for all supported invoice items.
- **OpenStack plugin**: allow to pull private flavors and images.
- **OpenStack plugin**: filter stale flavors and images by settings.
- **SLURM plugin**: fetch and store allocation usage by user for SLURM cluster.
- **SLURM plugin**: allow to pull SLURM allocation usage via pull service settings mechanism.

Release 2.7.7 - 10.10.2017

Upgrade guide

- Standard upgrade procedure.

Changes

- **Ansible plugin:** implement support for estimating Ansible-based apps resource requirements.
- **Core:** skip disabled actions in metadata rendering.
- **HomePort:** implement price estimation calculation and rendering for Ansible-based apps.
- **HomePort:** fix provider state display for pagination.
- **HomePort:** show VPC message for VMs and storages only.
- **HomePort:** prevent update of the default security group and rename of a regular security group to the default.
- **HomePort**: fix error with workspace selector in some browsers.
- **MasterMind:** create internal comment if it is not public.
- **OpenStack plugin:** prevent update and deletion of the default security group.
- **VirtEngine-Waldur module for Ansible**: implement support of check mode for the Ansible provisioned VirtEngine-Waldur resources

Release 2.7.6 - 4.10.2017

Release 2.7.5 was skipped, only HomePort was tagged, but did not contain the important bugfix.

Upgrade guide

- Standard upgrade procedure.

Changes

- **MasterMind Core:** return AsyncResult from background task so that celerybeat does not die**.**
- **HomePort:** allow to load user in popover again if last API request failed.
- **HomePort:** implement global filtered list of resources in support workspace.
- **HomePort:** enable SLURM allocation synchronization action.
- **HomePort:** implement global filtered list of resouces in support workspace.
- **MasterMind:** allow user to get comment if he is the author of the comment.
- **OpenStack plugin:** skip network synchronization if its tenant is not available.
- **PayPal plugin**: PayPal celery beat shouldn't be running jobs if PayPal is not configured.
- **SLURM plugin**: fix SLURM allocation creation.
- **SLURM plugin**: allow to specify default user account for each SLURM service setting.
- **SLURM plugin**: implement action to pull single SLURM allocation.
- **SLURM plugin**: delete all users from account only if they exist.

Release 2.7.4 - 30.09.2017

Upgrade guide

- Standard upgrade procedure.

Changes

- **HomePort:** add user filters to expert requests list.
- **HomePort:** preselect project if it is the only one.
- **MasterMind:** allow to filter expert bids by a customer.
- **MasterMind core:** cleanup CSS/HTML of API landing page.
- **OpenStack plugin**: force usage of public endpoints for all OpenStack service. Drops requirement to have access to admin endpoints. Raises minimal required Keystone API version to v3.
- **SLURM plugin**: fix SLURM allocation usage parsing.
- **SLURM plugin**: allow to specify default account when association for user and allocation is created.

Release 2.7.3 - 28.09.2017

Upgrade guide

- Standard upgrade procedure.

Changes

- **HomePort**: fix slimscroll directive on mobile.
- **HomePort**: fix sidebar item duplication.
- **HomePort**: reorder columns in project list using weights.
- **OpenStack plugin**: allow to create highly available Neutron routers.
- **MasterMind**: fix issue with displaying invoices for items with deleted scope
- **SLURM plugin**: fix deletion of allocations and synchronisation of user accounts.

Release 2.7.2 - 27.09.2017

Upgrade guide

- Standard upgrade procedure.

Changes

- **Grafana dashboard:** expose SLURM usage data in Grafana dashboard.
- **HomePort**: add project counter to organization dashboard.
- **HomePort**: translate resource actions.
- **HomePort**: render events for SLURM allocation operations in invoice.
- **HomePort**: drop policy configuration from organization registration wizard.
- **HomePort**: display all errors on customer creation wizard.
- **HomePort**: do not display provider settings for tenants if direct access is not enabled.
- **HomePort**: display quantity for all invoice items in the billing view.
- **HomePort**: fix expert request submission issue.
- **HomePort**: fix expert request state display.
- **HomePort**: integrate summernote as rich text editor.
- **HomePort**: do not submit service store form if it has not been changed.
- **HomePort**: put comments and request bids under expert request tabs.
- **HomePort**: display complete bid info as modal dialog.
- **HomePort**: add ability to complain on expert request item.
- **HomePort**: display more fields errors on service store form.
- **HomePort**: add billing type to request-based item as icon.
- **HomePort**: rename Sign in to Register on registration page.
- **HomePort**: allow to conceal phone, organization and registration method fields on registration form.
- **HomePort**: display expert icon on workspace selector.
- **HomePort**: render pie charts for SLURM allocation quota usage.
- **HomePort**: render stacked bar charts for SLURM allocation usage per user.

- **HomePort**: add "bootable" column to connected volume list in OpenStack instance tab.
- **HomePort**: attach AngularJS scope when table row detail are shown.
- **HomePort**: remove a tenant provider from available choices if tenant has been deleted.
- **HomePort**: render issue type description in issue type selector component.
- **HomePort**: render error message when issue creation request has failed.
- **HomePort**: add shortcuts to the project dashboard to create a VM or an expert request if supported.
- **HomePort**: add shortcuts to the service store and provider configuration if no providers have been configured.
- **HomePort**: render prices for SLURM provider in modal dialog.
- **HomePort**: alow to return from support workspace back to organization, project or user.
- **MasterMind**: only staff is allowed to modify price estimate for customer.
- **MasterMind**: customer owner is allowed to modify price estimates for project only.
- **MasterMind**: render scope resource type and uuid for generic invoice item.
- **MasterMind**: add quantity to generic invoice item to allow billing per single item.
- **MasterMind**: extend expert request with additional contract fields.
- **MasterMind**: register expert request as an invoice item.
- **MasterMind**: forbid multiple bids to the same request from the same customer.
- **MasterMind**: make usage of required attribute explicit for offering configuration.
- **MasterMind**: add support of html rich editor results.
- **MasterMind**: mark string without length limitation as a default type for a configured offering field.
- **MasterMind**: implement SAF format exporter for accounting data.
- **MasterMind core**: allow to configure available country choices in settings.
- **MasterMind core**: allow to toggle first project generation on organization creation in settings.
- **MasterMind core**: make PasswordWidget support fields with different name than password.
- **MasterMind core**: allow to filter events by resource type and uuid.
- **MasterMind core**: allow customer owner to edit quotas for owned project.
- **MasterMind core**: move limit and threshold enforcement model and logic from cost_tracking to billing.
- **OpenStack plugin**: fix security group service property population.
- **OpenStack plugin**: allow to toggle tenant user password which is not hidden by default.
- **OpenStack plugin**: render read-only JSON fields for volume, instance and backup.
- **OpenStack plugin**: don't change subnet's CIDR when name is updated.
- **PayPal plugin**: add support of generic invoice item.
- **SAML2 plugin**: update dependencies to upstream releases.
- **SLURM plugin**: fix permission validation for SLURM allocation provisioning.
- **SLURM plugin**: allow to specify SLURM gateway as service settings option**.**
- **SLURM plugin**: render allocation count in customer and project admin list view.

Release 2.7.1 (HomePort-only) - 19.09.2017

Upgrade guide

- Standard upgrade procedure.

Changes

- **HomePort**: allow to specify and modify CPU, GPU and RAM limits for SLURM allocation.
- **HomePort**: render CPU, GPU and RAM limit and usage in SLURM allocations list and detail view.
- **HomePort**: render FreeIPA login and SLURM gateway in SLURM allocation details view.
- **HomePort**: fix authentication flow.

Release 2.7.0 - 16.09.2017

Upgrade guide

- New model for project & customer cost limits has been introduced, initialize it with:

```
sudo -u waldur "waldur" rebuild_billing
```

- New SLURM quotas were introduced for projects and customers, initialize them with:

```
sudo -u waldur "waldur" recalculatequotas
```

Changes

- **Ansible plugin**: pass job ID as an extra argument to Ansible playbook.
- **Ansible VirtEngine-Waldur module**: allow to specify tags on OpenStack instance and security group provision.
- **Azure plugin**: update prices of Azure instances.
- **Cost planning plugin**: implement cost optimization for Azure.
- **DigitalOcean plugin**: add base resource filter for Droplet.
- **Homeport**: allow to specify row action per feature. Fix issue with missing features service in HomePort 2.8.
- **Homeport**: resolve flavor for Azure optimizers.
- **Homeport**: fix component units display for Azure optimizers.
- **Homeport**: fix memory unit display in total resource consumptions estimate.
- **Homeport**: implement snapshot import.
- **Homeport**: improve issue management events display.
- **Homeport**: disable submit button on action dialog if form data has not been changed. Affects most of row actions.
- **Homeport**: mark import as a stable feature.
- **Homeport**: make cancel button in Service Store detail view redirect to project dashboard.
- **Homeport**: fix expert services duplication.
- **Homeport**: render offering price and summary of request-based items if provided in configuration.
- **Homeport**: implement support for batch processing.
- **Homeport**: implement PayPal invoices support.
- **Homeport**: populate image property on request based items allowing to set custom image icon.
- **Homeport**: allow to define summary in HTML for request-based items.
- **Homeport**: update frontend support for project and customer cost limits and thresholds.
- **Homeport**: refresh site header component internal cache when state is changed.
- **Homeport**: do not display token related section on initial login.
- **Homeport**: improve tooltip for total cost gauge.
- **Homeport**: render generic invoice items with OpenStack items and offering items.
- **Homeport**: render SLURM CPU, GPU and RAM usage as dashboard gauges.
- **Homeport**: add counters for number of allocations in project list.
- **MasterMind core:** save password for newly created staff user.
- **MasterMind core:** wrap resource list view in transaction so that named server cursor in PostgreSQL are not lost.
- **MasterMind core:** allow to specify tags for resource on provisioning and modification.
- **MasterMind core:** allow to pass positional arguments in generic queryset manager for scope filtering.
- **MasterMind**: emit an event for issue creation after backend ID becomes known so that issue key is present in a log context.
- **MasterMind**: populate issue summary and description from ISSUE settings for request based items.
- **MasterMind**: use invoice data for project & customer price estimation.
- **MasterMind**: allow to specify tags on resource provision and modification.
- **MasterMind**: add SLURM, PayPal and Ansible plugins as dependencies.
- **MasterMind**: implement billing for SLURM, add admin for SLURM pricing package.
- **MasterMind:** fix name property serialization for generic invoice item.
- **OpenStack plugin**: handle user not found exception on tenant import.
- **OpenStack plugin**: fix service properties population.
- **OpenStack plugin**: fix OpenStackTenant network, subnetwork and security groups import issues.
- **OpenStack plugin**: do not delete volume when image is deleted.
- **OpenStack plugin**: populate image name for volume when volume is created for instance.
- **OpenStack plugin**: fix Nova client initialization.
- **OpenStack plugin**: upgrade OpenStack dependencies to latest stable release (Pike).
- **PayPal plugin**: update plugin structure to make it a valid plugin.
- **PayPal plugin**: introduce invoices management.
- **SLURM plugin**: implement SLURM accounting synchronization.
- **SLURM plugin**: allow to activate privilege escalation via sudo.
- **SLURM plugin**: allow to skip strict SSH host key verification.
- **SLURM plugin**: implement user role synchronization.
- **SLURM plugin**: allow to set resource allocation limits for CPU, GPU and RAM.
- **SLURM plugin**: add field tracker for SLURM allocation so that usage changes are detected.
- **SLURM plugin**: aggregate CPU, GPU and RAM allocation quotas usage by project and customer.
- **SLURM plugin**: add quota field for number of SLURM allocations in project.
- **SLURM plugin**: expose aggregated data for SLURM allocation usage in Grafana.

Release 2.6.9 - 28.08.2017 (HomePort only)

Upgrade guide

- Standard upgrade

Changes

- Critical bugfix with component initialization order that caused errors when loading HomePort when installed from RPMs.

Release 2.6.8 - 26.08.2017

Upgrade guide

- New block device quotas were introduced for projects, initialize them with:

```
sudo -u waldur "waldur" recalculatequotas
```

Changes

- **Ansible module:** improve resource fetching by UUID.
- **Ansible plugin:** allow to specify service project link and SSH key for Ansible job.
- **Ansible plugin:** add applications counter for project workspace.
- **Azure plugin**: update plugin to new architecture.
- **Azure plugin**: expose initial user name and password to Virtual Machine endpoint.
- **Azure plugin**: extend plugin with endpoints and external/internal IPs to display information required for remote resource management.
- **Azure plugin:** populate SPLs with settings cloud service name on creation if present.
- **Azure plugin**: mark cloud service name as required field.
- **Azure plugin**: generate translation placeholders for Estonian, English, Latvian, Lithuanian, Russian and Ukrainian.
- **HomePort**: introduce Azure support.
- **HomePort**: display credentials and connection info on Virtual Machine details view.
- **HomePort**: clear cache for Ansible jobs list when new job is provisioned.
- **HomePort**: implement periodic pulling for Ansible job details view.
- **HomePort**: skip intro for adding new organization if user is not allowed to create one.
- **HomePort**: implement lazy loading for resource analytics module.
- **HomePort**: add counters for applications, experts and requests.
- **HomePort**: implement OpenStack volume import.
- **HomePort**: show total cost in organization workspace chart without tax.
- **HomePort**: make state icons for Ansible applications consistent with design used for resource states.
- **HomePort**: allow global support role to see detail action button in organization team view.
- **HomePort**: add block volume and snapshot counters for project dashboard.
- **HomePort**: order charts by title in project and organization workspace dashboard.
- **HomePort:** migrate charts in project and organization workspace dashboard to React/Redux.
- **HomePort:** conceal "Preferred language" and "Competence" with a feature toggle.
- **HomePort:** conceal extra columns for non-staff/non-support user in issues list view.
- **HomePort**: add organization name and agreement number to accounting record detail view.
- **HomePort**: implement OpenStack instance import.
- **HomePort**: implement OpenStack tenant import.
- **HomePort**: fix category items duplication bug.
- **HomePort**: render list of related resources in Ansible job details view.
- **MasterMind**: allow to specify list of fields for invoice via REST API.
- **MasterMind**: move tenant service settings creation logic to OpenStack plugin.
- **MasterMind**: add package assign action for imported tenants.
- **MasterMind core**: add offerings and experts counters for project workspace.
- **MasterMind core**: skip sub-resources in provider resource counter, so that only primary resources are counted.
- **MasterMind core:** include tests in package distribution.
- **MasterMind core:** add ability to extend project and customer counters.
- **MasterMind core:** add start of accounting date as a column in organization list view in admin site.
- **MasterMind core:** add volume and snapshot model. Add volume and snapshot count quota for project.
- **OpenStack plugin**: synchronize error message for OpenStack instance if it is defined on backend.
- **OpenStack plugin**: include tests in package distribution.
- **OpenStack plugin**: implement volume import.
- **OpenStack plugin**: implement instance import.
- **OpenStack plugin**: implement tenant import.

- **OpenStack plugin**: implement snapshot import.
- **OpenStack plugin**: remove bootable flag for restored volumes.

Release 2.6.7 - 06.08.2017

Upgrade guide

- Standard upgrade

Changes

- **Ansible plugin**: disable SSH host key verification.
- **Ansible plugin**: package Ansible VirtEngine-Waldur module as part of RPM.
- **Ansible plugin**: filter OpenStack resources and properties by name exactly.
- **Ansible plugin**: fix filtering OpenStack properties by UUID.
- **Ansible plugin**: schedule Ansible job to execute when it's created.
- **Ansible plugin**: allow to specify Ansible library path.
- **Ansible plugin:** implement module to get details of existing virtual machine instance.
- **HomePort**: decrease team member create/edit view.
- **HomePort**: migrate sparkline component from AngularJS to ReactJS.
- **HomePort**: restructure billing details page.
- **HomePort**: make output log scrollable for Ansible job.
- **HomePort**: use cost without VAT for a total unit value.
- **HomePort**: put 'VAT is not included' message near Total to make it more explicit.
- **HomePort:** allow to specify image file as icon in Service store category selector.
- **HomePort:** add scroll and search bar in Service store category selector.
- **MasterMind:** fix unicode encoding for CSV report.
- **MasterMind:** include request based items in CSV report.
- **MasterMind:** allow to upload image file as an icon for Ansible job. Resize and rename icon on upload.
- **OpenStack plugin**: fix floating IP synchronization.

Release 2.6.6 - 01.08.2017

Upgrade guide

- Standard upgrade

Changes

- **HomePort**: allow creation of expert request bids by expert managers.
- **HomePort**: remove unlink action for providers created from shared settings.
- **HomePort**: remove created by info from succeeded events, scheduling events display user info.
- **HomePort**: allow customer owner to accept a request bid.
- **HomePort**: allow customer owner to cancel expert request.
- **HomePort**: in expert request details view allow to list and add comments.
- **HomePort**: optimize header on mobile screen.
- **HomePort**: migrate to Webpack 3.
- **HomePort**: update Inspinia theme to the latest version (from 2.6.0 to 2.7.1).
- **HomePort**: implement lazy loading for datatables plugin.
- **HomePort**: drop Bower dependency.
- **HomePort**: implement lazy loading for leaflet component.
- **HomePort**: add applications category to the service store selector.
- **HomePort**: fix backup recovery form errors preventing instance from restore.
- **HomePort**: render expert provider indicator in workspace selector, user dashboard and organization dashboard.
- **HomePort**: drop organization icon on smaller resolution screens in create wizard.
- **HomePort**: add expert manager introduction using intro.js.
- **HomePort**: expose accounting start date in organization details.
- **HomePort**: in Appstore render application provision form.
- **HomePort**: implement Ansible job details view.
- **MasterMind**: mark OpenStack package events as debug only.
- **MasterMind**: copy article code and product code from offering configuration if it is defined.
- **MasterMind:** enable ordering expert requests by name, state, customer and project fields.
- **MasterMind:** ensure that expert team for bid has at least one member.
- **MasterMind core**: exclude debug events from queryset.
- **MasterMind core**: dumpusers command now works with unicode names too.

- **MasterMind core**: add a password widget to allow user to toggle a password visibility on the admin page.
- **MasterMind core**: allow to filter resources, services and properties by name exactly.
- **OpenStack plugin**: add units to tenant ram, vcpu and storage quotas update event logs.
- **OpenStack plugin**: prevent KeyError in deletion task if price estimate scope is not defined.
- **OpenStack plugin**: add a tenant floating IPs update message text to the event log when update is triggered with the same values.

Release 2.6.5 - 17.07.2017

Upgrade guide

- Standard upgrade

Changes

- **HomePort**: fix broken invoice list display.
- **MasterMind**: implement backend support for terminating expert contract.
- **MasterMind**: implement bimonthly billing model.
- **OpenStack plugin**: make source_snapshot and image fields for volume model optional in admin.

Release 2.6.4 - 14.07.2017

Upgrade guide

- Standard upgrade

Changes

- **FreeIPA:** disable by default Celery beat tasks for FreeIPA group synchronization.
- **HomePort**: implement frontend support for creating expert request.
- **HomePort**: allow to filter organization by abbreviation in the workspace selector.
- **HomePort**: render placeholder in the workspace selector if there are no organizations matching the filter.
- **HomePort**: fix resource pulling if resource is updating.
- **MasterMind core:** fix RPM package installation script to skip creation of "waldur" user if it exists already.
- **MasterMind core:** in admin site fix shared service settings creation.
- **MasterMind core:** in admin site fix display of organization field when service type is selected.
- **OpenStack plugin**: set resource state to OK if update is successful.

Release 2.6.3 - 12.07.2017

Upgrade guide

- Standard upgrade

Changes

- **MasterMind core:** changed VirtEngine-Waldur CLI utility name from "nodeconductor" to "waldur".
- **MasterMind core:** add user full name and username to the event logs about SSH keys.
- **MasterMind core**: merge service settings options on update through API. Fixes an issue which lead to data loss on updating provider settings.
- **MasterMind core and plugins**: generate Russian, Latvian, Ukrainian and Lithuanian locales placeholders.
- **MasterMind core**: disable by default non-staff user to create organization.
- **MasterMind:** implement backend support for creating expert request and bid.
- **OpenStack plugin**: rename tenant service settings if tenant is renamed.
- **OpenStack plugin**: make action_details field readonly on admin panel.
- **HomePort**: add a tooltip on billing item start time explaining timezone used.
- **HomePort:** add button to the service store for creating new project.
- **HomePort**: remove pop-ups on every server error.
- **HomePort**: convert organization creation dialog to multi-step wizard.
- **HomePort**: allow to indicate existing organization as expert provider.
- **HomePort**: allow to create expert provider during organization creation.
- **HomePort**: make app installable on mobile device without the need for an app store.

Release 2.6.1 - 4.07.2017

Upgrade guide

Old name	New name	Description
nodeconductor	waldur-core	RPM package for VirtEngine-Waldur core application
nodeconductor-assembly-waldur	waldur-mastermind	RPM package for VirtEngine-Waldur MasterMind plugins
nodeconductor-auth-openid	waldur-auth-openid	RPM package for VirtEngine-Waldur OpenID authentication pl
nodeconductor-auth-social	waldur-auth-social	RPM package for VirtEngine-Waldur social authuntication plug
nodeconductor-saml2	waldur-auth-saml2	RPM package for VirtEngine-Waldur SAML2 authentication plu
nodeconductor-aws	waldur-aws	RPM package for VirtEngine-Waldur AWS EC2 plugin
nodeconductor-digitalocean	waldur-digitalocean	RPM package for VirtEngine-Waldur DigitalOcean plugin
nodeconductor-openstack	waldur-openstack	RPM package for VirtEngine-Waldur OpenStack plugin
nodeconductor-cost-planning	waldur-cost-planning	RPM package for VirtEngine-Waldur cost planning plugin
nodeconductor-assembly-waldur-check	waldur-mastermind-check	Self status check utility
nodeconductor-celery	waldur-celery	VirtEngine-Waldur Celery process name
nodeconductor-celerybeat	waldur-celerybeat	VirtEngine-Waldur Celery beat process name
nodeconductor-uwsgi	waldur-uwsgi	VirtEngine-Waldur uWSGI process name
/etc/nodeconductor	/etc/waldur	VirtEngine-Waldur configuration directory
/etc/nodeconductor/settings.ini	/etc/waldur/core.ini	VirtEngine-Waldur core configuration file
/etc/nodeconductor/uwsgi.ini	/etc/waldur/uwsgi.ini	VirtEngine-Waldur uWSGI configuration file
/etc/nodeconductor/nodeconductor_plus.py	/etc/waldur/override.conf.py	Another VirtEngine-Waldur configuration file (used to override and – temporarily – configure some plugins)
/etc/nodeconductor/nodeconductor_saml2.py	/etc/waldur/saml2.conf.py	VirtEngine-Waldur SAML2 plugin configuration file
/etc/nodeconductor/saml2	/etc/waldur/saml2	VirtEngine-Waldur SAML2 plugin additional configuration direc
/etc/nodeconductor/saml2/attribute-maps	/etc/waldur/saml2/attribute-maps	VirtEngine-Waldur SAML2 plugin attribute mapping directory
/usr/share/nodeconductor	/usr/share/waldur	VirtEngine-Waldur core data directory
/var/lib/nodeconductor	/var/lib/waldur	VirtEngine-Waldur working directory
/usr/share/nodeconductor/static	/usr/share/waldur/static	VirtEngine-Waldur Django static root
/var/lib/nodeconductor/media	/var/lib/waldur/media	VirtEngine-Waldur Django media root
/var/log/nodeconductor	/var/log/waldur	VirtEngine-Waldur log directory
/var/log/nodeconductor/nodeconductor.log	/var/log/waldur/core.log	VirtEngine-Waldur core main log file
/var/log/nodeconductor/events.log	/var/log/waldur/events.log	VirtEngine-Waldur events log file
/etc/logrotate.d/nodeconductor	/etc/logrotate.d/waldur	Logrotate configuration for VirtEngine-Waldur

- New packages would replace old package during update via package manager.
- Default database has been renamed from nodeconductor to waldur – make sure to update `core.ini` if you are using defaults!
- Default database user has been renamed from nodeconductor to waldur – make sure to update core.ini if you are using defaults!

Migrate database

```
# Execute as postgres user
psql -c 'ALTER DATABASE nodeconductor RENAME TO waldur;'
/usr/pgsql-9.6/bin/createuser waldur
psql waldur -c 'REASSIGN OWNED BY nodeconductor TO waldur;'
```

Update NodeConductor to VirtEngine-Waldur MasterMind

```
#
# Stop NodeConductor services
# Host: waldur-api
#
systemctl stop nodeconductor-uwsgi
systemctl stop nodeconductor-celerybeat
systemctl stop nodeconductor-celery

#
# Back up NodeConductor database
# Host: waldur-db
#
su - postgres -c 'pg_dump nodeconductor' | gzip -9 >
nodeconductor-$(date +'%Y%m%dT%H%M%S').sql.gz

#
# Install VirtEngine-Waldur MasterMind
# Host: waldur-api
#
yum clean all
yum install waldur-mastermind

# Update /etc/waldur/core.ini
# Known settings that need to be fixed:
#  - postgresql.host: make sure this is enabled, i. e. TCP connections
are used, *not* socket connections!
#  - postgresql.database|user|password: defaults are changed from
'nodeconductor' to 'waldur' -- update as needed
#
# See output of this command for details:
#
#  diff -u /etc/nodeconductor/settings.ini.rpmsave /etc/waldur/core.ini

su - waldur -c "nodeconductor migrate --noinput"

#
# Start VirtEngine-Waldur services
# Host: waldur-api
#
systemctl start waldur-celery
systemctl start waldur-celerybeat
systemctl start waldur-uwsgi
```

Changes

- **HomePort:** render list of events for invoice in modal dialog.
- **HomePort**: implement frontend support for registering organization as an expert provider.
- **HomePort:** add cookies consent overlay.
- **HomePort:** display user status only to staff and global support users in user details.
- **HomePort:** add back button to the privacy policy, ToS, and redirect user to the previous page on logo click.
- **MasterMind Core:** fix UUID validation in GenericRelatedField.
- **MasterMind Core:** fix dumpusers management command encoding error.
- **MasterMind:** fix an error in /admin caused by trying to render missing organization payment details.
- **MasterMind**: save package details for OpenStack invoice item when VPC package is deleted.
- **MasterMind**: implement backend support for registering organization as an expert provider
- **OpenStack plugin**: eliminate security groups duplication on tenant pull operation.
- **Social authentication plugin:** handle 401 responses from Google authentication provider

Release 2.6.0 - 23.06.2017

Upgrade guide

New FreeIPA quotas were introduced for projects and organizations, initialize them with:

```
sudo -u nodeconductor "nodeconductor" recalculatequotas
```

Synchronize identity providers metadata with:

```
sudo -u nodeconductor "nodeconductor" sync_saml2_providers
```

Changes

- **Ansible plugin:** implement playbooks upload and their execution using jobs.
- **FreeIPA plugin**: implement backend support for FreeIPA batch processing.
- **FreeIPA plugin**: improve frontend-side validation for FreeIPA profile creation form.
- **FreeIPA plugin**: add logging for FreeIPA profile events.
- **FreeIPA plugin**: enable filtering FreeIPA profile by user UUID.
- **HomePort**: improve user facing message when no VM providers exist.
- **HomePort**: improve provider selector in Service store on smaller resolution screen.
- **HomePort**: migrate build system from Grunt to Webpack.
- **HomePort**: preselect an SSH key if user has only one configured.
- **HomePort**: cover OpenStack Instance provisioning with unit tests.
- **HomePort**: add tooltip for prices that do not include VAT or are estimated.
- **HomePort**: group resources by project in accounting / billing view.
- **HomePort**: display resource type in the resource event list view.
- **HomePort**: filter issues by current user, organization and project.
- **HomePort**: switch to Yarn for dependency management.
- **HomePort**: improve feedback for mandatory fields on user signup.
- **MasterMind:** add a check of admin site endpoints availability with unit tests.
- **MasterMind:** improve project updated event message.
- **MasterMind:** enable parallel test execution.
- **MasterMind:** send accounting data as CSV via email for automated processing.
- **OpenStack plugin**: fix race conditions for service property synchronization.
- **SAML2 plugin:** use keypair for assertion encryption.
- **SAML2 plugin:** fix authentication event notification.
- **SAML2 plugin:** fix memory usage via database cache for identity providers metadata.
- **SAML2 plugin**: add parameters required by eduGAIN specs.
- **SAML2 plugin**: update registration method for user logged in via SAML2.
- **SAML2 plugin**: extend metadata with attributes required by CoC.

Release 2.5.5 - 9.06.2017

Upgrade guide

- Nothing yet

Changes

- **Ansible VirtEngine-Waldur module:** OpenStack instance floating IP assignment module is implemented.
- **FreeIPA plugin:** skip SSH key synchronization if keys are the same.
- **HomePort:** conceal username and password fields in OpenStack tenant provision form if tenantCredentialsVisible is false**.**
- **HomePort:** revert security group rendering in VM create to a textinput.
- **HomePort:** extend organization details view.
- **HomePort:** allow user to report errors in organization details by submitting JIRA issue.
- **HomePort:** preserve sorting order on resources list auto-refresh.
- **HomePort:** implement support for FreeIPA accounts.
- **HomePort:** fix tab links in breadcrumbs.
- **HomePort:** fix duplicating query parameters leading to 502 error.
- **HomePort:** add eduGAIN support for authentication.
- **HomePort:** add link to version number, which opens changelog in a new tab.
- **MasterMind:** quantize daily price for OpenStack package to 2 places after decimal point.
- **MasterMind:** fix user update via admin site when civil number is not specified.
- **MasterMind:** introduce phone number, email address and agreement number fields for the customer model.
- **MasterMind:** allow staff to specify list of authorized subnets for organization via admin site.
- **MasterMind:** implement validator for a comma separated list of IPv4 or IPv6 CIDR addresses.
- **MasterMind:** add billing codes for the service store items: package templates, support offerings and corresponding invoice items.
- **MasterMind**: display user roles in organizations and projects in user profiles in admin interface.
- **OpenID plugin**: fix Django 1.11 compatibility.
- **OpenStack plugin:** fix a pull floating IPs operation causing OpenStackTenant FloatingIP to be duplicated twice. Reproduces on floating IP assignment as well.
- **OpenStack plugin:** pull tenant quotas when tenant is created.
- **OpenStack plugin:** add event log details when floating IP is attached to or detached from OpenStack instance.
- **OpenStack plugin:** fix cleanup of maximal backup count within backup schedules.
- **Social auth plugin**: add authentication events logging.

Release 2.5.4 - 2.06.2017

Upgrade guide

- Django has been upgraded from 1.9 to 1.11.
- ElasticSearch has been upgraded from 1.4 to 5.4. Data migration is needed as well.
- Django model utils has been upgraded from python2-django-model-utils-2.5.2 to python-django-model-utils-3.0.0
  In order to upgrade please remove old version first, because otherwise it leads to incompatibility conflict.

Changes

- **Analytics plugin:** disable plugin in settings by default.
- **Ansible VirtEngine-Waldur module**: OpenStack instance provisioning module is implemented.
- **Ansible VirtEngine-Waldur module:** OpenStack security group provisioning module is implemented.
- **AWS plugin**: mark i18n strings.
- **DigitalOcean plugin**: mark i18n strings.
- **DigitalOcean plugin**: synchronize size name on droplet import, skip invalid droplets in cost tracking.
- **FreeIPA plugin**: backend support for user creation, SSH key synchronization, enable/disable operations is implemented.
- **HomePort**: disable autocomplete in provider create form.
- **HomePort**: implement resources sorting.
- **HomePort**: update package limits in summary when VPC package has been changed.
- **HomePort**: skip unlimited service project link in checkout summary.
- **HomePort**: allow to set cost limit on organization level for both new and existing organizations.
- **HomePort**: allow user to request OpenStack tenant credentials via support.
- **HomePort**: improve tooltip rendering in resource actions dropdown menu.
- **HomePort**: add issue link to the event log details.
- **HomePort**: fix footer overflow.
- **HomePort**: migrate most of Bower and Grunt scripts to Webpack.
- **HomePort**: add verification of emails from IdP and invitations.
- **HomePort**: display an invitation check dialog for invited users.
- **MasterMind**: add validation of the invitation email. Grant role only if user and invitation emails match.
- **MasterMind**: add limit validation if organization cost limit is set.
- **MasterMind**: allow to add usage to unlimited quota.
- **MasterMind**: allow overriding sorting fields for resource filters.
- **MasterMind**: fix field validation for provider provision.

- **MasterMind**: fix quotas creation on provider settings creation via admin site.
- **MasterMind**: synchronize price list item with package template automatically.
- **MasterMind**: fix tests execution on PostgreSQL.
- **MasterMind**: add name, abbreviation and UUID search in organization list in admin interface.
- **MasterMind**: fix filtering OpenStack package by name.
- **MasterMind**: implement filtering OpenStack package by template URL and UUID.
- **MasterMind**: fix system notification, alert and webhook views on admin site.
- **OpenStack plugin:** don't enforce default limits for service project link.
- **OpenStack plugin:** reset invalid default quotas for service project link.
- **OpenStack plugin:** fix tenant quotas usage synchronization.
- **OpenStack plugin:** pull security groups, floating IPs and quotas on tenant pull operation.
- **OpenStack plugin:** delete tenant volumes ignoring their current state.
- **OpenStack plugin**: allow to hide OpenStack tenant credentials and use auto-generated login and password instead.
- **Social authentication plugin**: save civil ID and ignore email coming from smartid.ee
- **Support plugin:** disable plugin in settings by default.

Release 2.5.2 - 6.05.2017

Documentation

Upgrade guide

- Price estimation strategy have been updated, refresh them with:

```
sudo -u nodeconductor "nodeconductor" rebuildpriceestimates
```

- New OpenStack storage quotas were introduced, initialise them with:

```
sudo -u nodeconductor "nodeconductor" recalculatequotas
```

Changes

- **HomePort:** added support for personal notifications (emails and webhooks).
- **HomePort**: enable SAML2 authentication.
- **HomePort**: reduce number of requests to counters endpoint.
- **HomePort**: add links to owned organizations and managed projects on user dashboard.
- **HomePort**: disable "remove button" in project list if removal is not possible.
- **HomePort**: enable/disable "add project" button in project list if project quota has changed.
- **HomePort**: reduce provider selector size in Service Store.
- **HomePort**: enable project cost limit for project.
- **HomePort**: render estimated cost of project in projects list.
- **HomePort**: add project details dialog to project dashboard.
- **HomePort**: make data volume an optional parameter for instance creation.
- **HomePort**: support ticked description has a better format now.
- **HomePort**: fix missing sizes of a DO droplet in a select widget on droplet creation.
- **HomePort**: preselect UTC timezone on backup and snapshot schedule creation.
- **HomePort**: populate retention time and timezone in the backup edit pop up.
- **HomePort**: preselect the `default` security group and make it non-removable in the OpenStack Instance creation pop up.
- **HomePort**: preselect the first subnet if it is the only choice in the OpenStack Instance creation pop up.
- **HomePort**: extend floating IP list with a link to related instance
- **HomePort**: disable tenant password autofill in the tenant creation window
- **MasterMind**: fix issue with "broken" images in admin interface (incorrect mime-type of SVG images).
- **MasterMind**: added BackendURLField and validator supporting HTTP(S) and LDAP(S) schemas.
- **MasterMind**: implement Celery task for pushing statistical data to InfluxDB.
- **MasterMind:** implement Grafana dashboard for VirtEngine-Waldur managed resources and cost tracking across all organizations.
- **MasterMind**: raise ElasticSearchClientError if elastic search configuration is missing or empty.
- **OpenStack plugin**: allow backup and restoration of instances with 1 or more volumes.
- **OpenStack plugin**: make data_volume_size an optional parameter for instance creation.
- **OpenStack plugin**: prevent creation of duplicate floating IP on resource synchronization.

- **OpenStack plugin**: display a warning in VPC package template management form if components prices cannot be changed due to linked VPCs.
- **OpenStack plugin**: introduce quotas for volume and snapshot size.
- **OpenStack plugin**: sync quotas usage when OpenStack Tenant is pulled.
- **OpenStack plugin**: update next trigger time of backup and snapshot scheduling, so only one backup is created between schedule checkpoints.
- **OpenStack plugin**: extend openstack floating ip resource with information about linked instance
- **Support plugin**: disable background processing if basic support mode is enabled.
- **Support plugin**: update setup command to handle Elasticsearch check failures nicely.
- **SAML2 plugin**: skip CSRF check on initial login request.

Release 2.5.0 - 23.04.2017

Documentation

Upgrade guide

- New quotas were introduced, initialise them with:

```
sudo -u nodeconductor "nodeconductor" recalculatequotas
```

Changes

- **HomePort**: project management dialog has been moved to organization workspace, to project list.
- **HomePort**: concealChangeRequest configuration option has been added.
- **HomePort**: allow to manage quotas for provider project links.
- **HomePort**: remove image selector from volume creation form.
- **HomePort**: change creation form title from 'Configure' to 'Create'.
- **MasterMind Core**: ensure that quota usage does not exceed new quota limit when quota is updated.
- **OpenID plugin**: generated usernames have been converted to UUID to avoid confusion with mapping from unicode names.
- **Support plugin**: Disable issue notifications if only assignee changed.
- **OpenStack plugin**: allow attaching a volume to an instance in active state.
- **SAML2 plugin**: implement support for TAAT.
- **Social plugin:** enable authentication for smartid.ee

Previous releases

Please see https://github.com/virtengine/ve-core-v2/blob/develop/CHANGELOG.rst or commit messages.

Project governance

Overview

Below we describe roles and high level process of VirtEngine-Waldur project governance. A person can have multiple roles, however some roles c assigned only to a one person at a time for avoiding split brain problems.

Roles

- **Project manager** - defines requirements for a particular VirtEngine-Waldur-based project. Multiple people can have this role.
- **Product owner** - analysis requirements from project managers and fits into a product roadmap. Maintains technical feature list, maintains changelog. Single person at a time can have this role.
- **Architect** - oversees VirtEngine-Waldur platform development, including backend, frontend, external tools. Has a final word on the way hc components are integrated. Single person at a time can have this role.
- **Component owner** - responsible for the architecture and implementation of a particular component (e.g. VirtEngine-Waldur HomePort, Vi MasterMind). Single person at a time can have this role.
- **Release manager** - packages components, composes release documentation (changelog, notes, upgrade instructions). Multiple people can have this role.
- **Developer** - responsible for fulfilling tasks created. Multiple people can have this role.
- **External contributor** - provides extensions to VirtEngine-Waldur ecosystem (tools, links, documentation, etc). Multiple people can have tl

Work flow

1. **Project manager** communicates with customers, defines requirements for **product owner**.
2. **Product owner** analyses requirements from all projects and creates epics for technical features implementation. Epics are assigned to **ar chitect**.
3. **Architect** defines general development plan for features from product owner, adds description to the epic and creates separate tasks or stories for **component owners**.
4. **Component owner** provides estimates for task from architect, splits them if necessary and implements them with **developers.**
5. **Architect** resolves **Epics, Product owner** validates, updates Changelog**.** Once all Epics are resolved, **Product owner** initiates version release**.**
6. **Release manager** follows release checklist for a new VirtEngine-Waldur version.

Contributing

Source code

VirtEngine-Waldur source code is mirrored to GitHub: https://github.com/virtengine & https://github.com/opennode . Code is split into core, plugins & UI.

Contributing to VirtEngine-Waldur

VirtEngine-Waldur is an open project welcoming contributions. To assure the smooth acceptance of contributions please review the contribution gu below.

Pull requests

We welcome contributions (patches, new features, translations, etc) through pull requests. Please make sure that the following is addressed when making a pull request:

- a request has a clear scope - new feature, bugfix, refactoring. Granular pull requests make integration much faster;

- the code is following pep8/js code conventions;

License

By contributing your code, you agree to license your contribution under the MIT license.

Support

If you want to get guaranteed support for your VirtEngine-Waldur deployment, please contact info@opennodecloud.
For more information check: Waldur.com & VirtEngine.com.

For community support, please leave a comment under the corresponding wiki page.