Manila plugin for Fuel User Guide

Release 1.0-1.0.0-1

Mirantis Inc.

October 20, 2016

CHAPTER

ONE

OVERVIEW

Overview of the Manila plugin for Fuel

The purpose of this document is to describe how to install, configure and use the Manila plugin 1.0.0 for Fuel 9.1

The Manila is the OpenStack project that provides "File Sharing as a Service". Main goal of the project is providing coordinated access to shared or distributed file systems to OpenStack Compute instances. But as a many other OpenStack services it can be used independently according to modular design established by OpenStack.

The Manila based on that principes:

- Component based architecture: Quickly add new behaviors
- Highly available: Scale to very serious workloads
- Fault-Tolerant: Isolated processes avoid cascading failures
- Recoverable: Failures should be easy to diagnose, debug, and rectify
- Open Standards: Be a reference implementation for a community-driven api
- API Compatibility: Manila strives to provide API-compatible with popular systems like Amazon EC2

This plugin brings features of Manila into Mirantis OpenStack.

Software prerequisites

To use the Manila plugin for Fuel 9.1, verify that your environment meets the following prerequisites:

Prerequisites	Version/Comment
Fuel	9.1
manila-service-image	last
NetApp® ONTAP®	8 or later

The manila-service image is the service image for generic driver. It should be build from https://github.com/openstack/manila-image-elements.

Limitations

The Manila plugin for Fuel 9.1 has some known issues/limitations of usage:

- Manila CLI response with warnings if specific configuration https://bugs.launchpad.net/fuel-plugin-manila/+bug/1633018
- Manila services uses publicURL instead of internalURL https://bugs.launchpad.net/fuel-plugin-manila/+bug/1633456
- Manila share created from snapshot has error status (Generic) https://bugs.launchpad.net/fuel-plugin-manila/+bug/1634818

Licenses

Component	License
Manila plugin for Fuel	Apache 2.0

CHAPTER

TWO

INSTALL AND CONFIGURE MANILA PLUGIN FOR FUEL

Install Manila plugin for Fuel

Before you install Manila plugin for Fuel 9.1, verify that your environment meets the requirements described in *Software prerequisites*. You must have the Fuel Master node installed and configured before you can install the plugin. This plugin is hotpluggable, so you can install the Manila plugin for Fuel after you deploy an OpenStack environment.

To install Manila plugin for Fuel:

1. Download Manila plugin for Fuel from the Fuel Plugins Catalog.
2. Copy the plugin `.rpm` package to the Fuel Master node:

   Example:

   ```
   # scp fuel-plugin-manila-1.0-1.0.0-1.noarch.rpm root@fuel-master:/tmp
   ```

3. Copy the manila-service-vm iso to the Fuel Master node:

   Example:

   ```
   # scp manila-service-image.qcow2 root@fuel-master:/tmp
   ```

4. Log into Fuel Master node CLI as root.
5. Set path to manila service image into environment variable MANILA_IMAGE

   Example:

   ```
   # export MANILA_IMAGE=/tmp/manila-service-image.qcow2
   ```

6. Install the plugin by typing:

   ```
   # fuel plugins --install fuel-plugin-manila-1.0-1.0.0-1.noarch.rpm
   ```

7. Verify that the plugin is installed correctly:

   ```
   # fuel plugins
   id | name               | version | package_version
   ---|------------     ---|---------|----------------
   1  | fuel-plugin-manila | 1.0.0   | 4.0.0
   ```

Uninstall Manila plugin for Fuel

To uninstall Manila plugin for fuel, follow the steps below:

1. Log in to the Fuel Master node CLI
2. Delete all environments in which Manila plugin for Fuel is enabled:

   Example:

   ```
   # fuel --env <ENV_ID> env delete
   ```

3. Uninstall the plugin:

   ```
   # fuel --plugins --remove fuel-plugin-manila==1.0.0
   ```

4. Verify wheter the Manila plugin for Fuel was uninstalled successfully:

   ```
   # fuel plugins
   ```

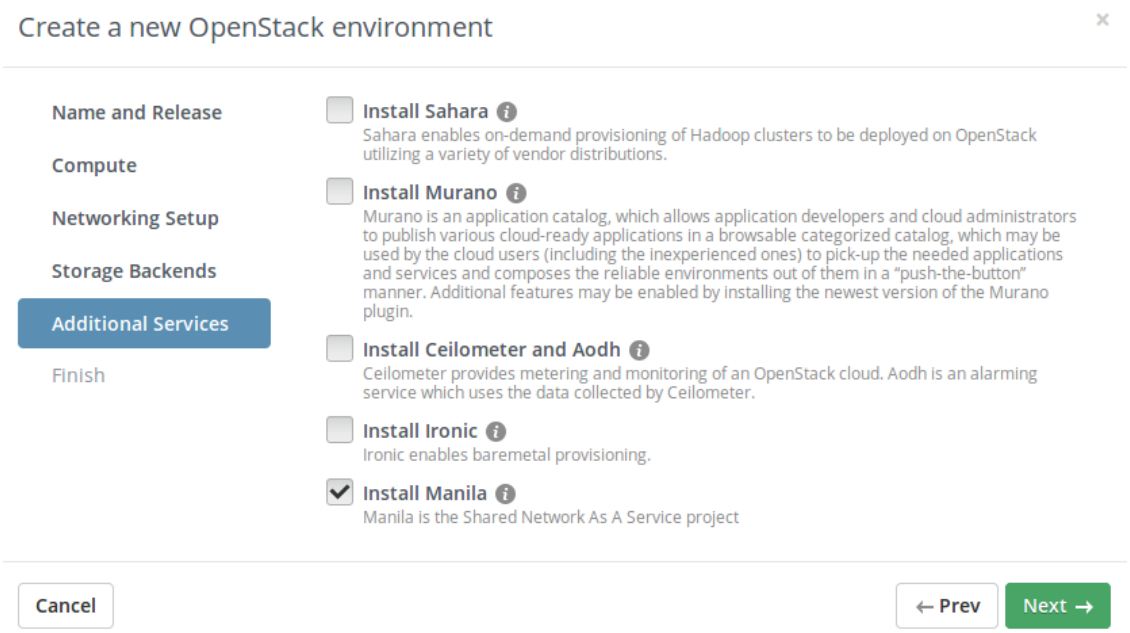
Proceed to *Configure Manila plugin for Fuel*.

Configure Manila plugin for Fuel

Configuring and deploying an environment with Manila plugin for Fuel involves creating an environment in Fuel and modifying the environment settings.

To configure OpenStack environment with Manila plugin:

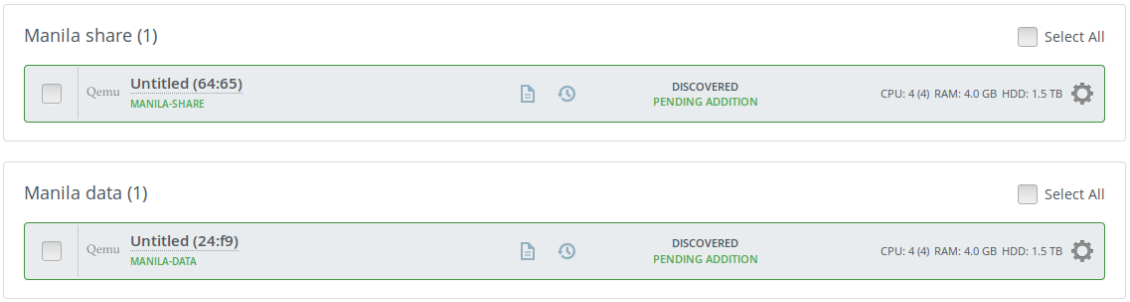
1. Create an OpenStack environment as described in the Fuel User Guide:
2. In the *Additional services* menu, select *Install Manila*:



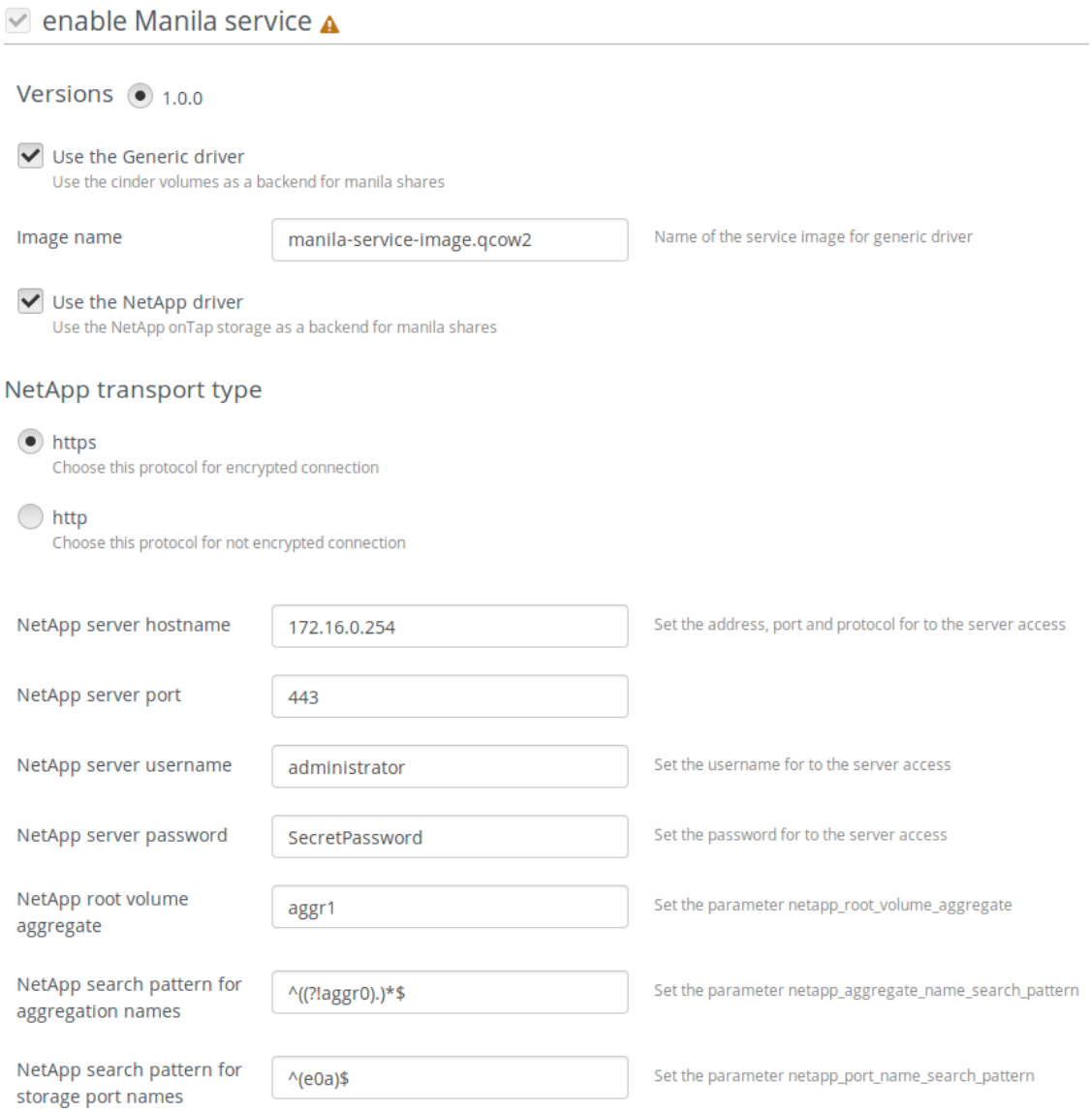
3. Follow next steps of the Create a new OpenStack environment

instruction.

1. In the *Nodes* tab of the Fuel web UI add at least one node with roles manila-share and manila-data:



2. In the *Settings* tab, click *OpenStack Serivces*:
   (a) Check that *Enable Manila service* is enabled.
   (b) Set the choosend backend for Manila.
      i. For generic driver specify the *Image name* exactly same as you set on the plugin installation stage.
      ii. For NetApp driver specify hostname, credential and parameters related to your environment.



CHAPTER

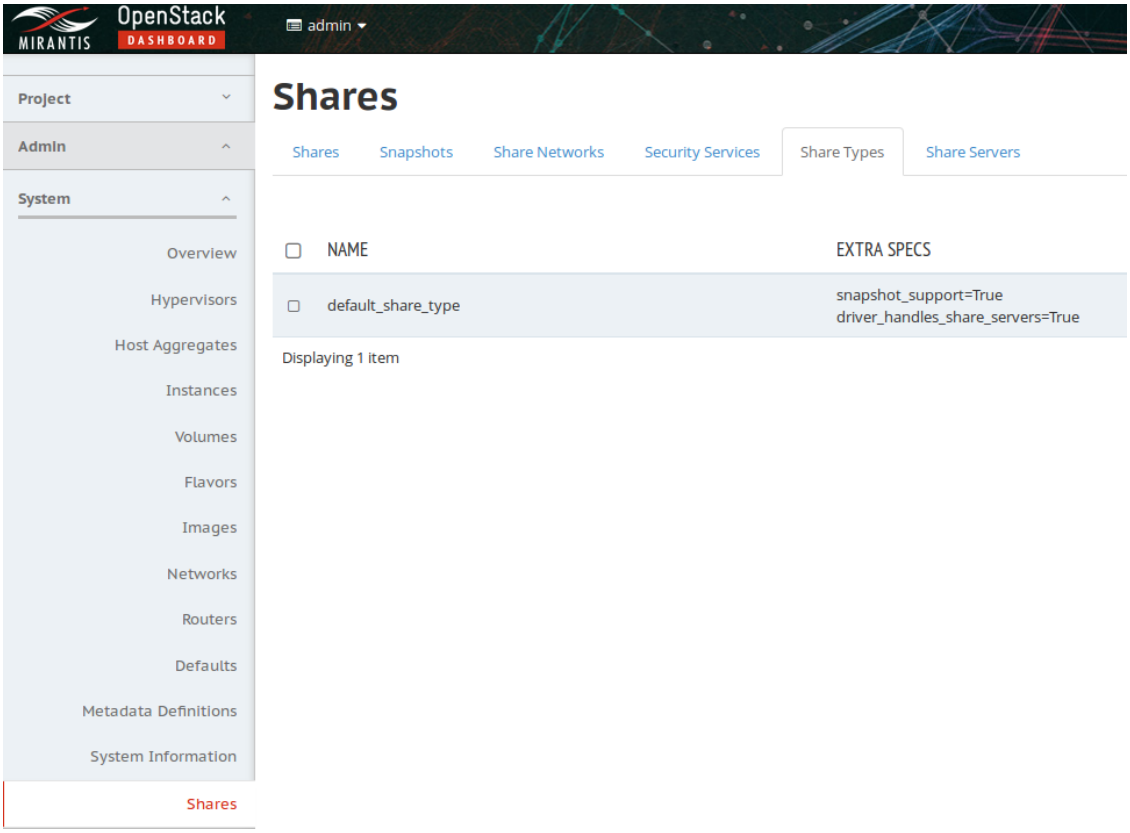
THREE

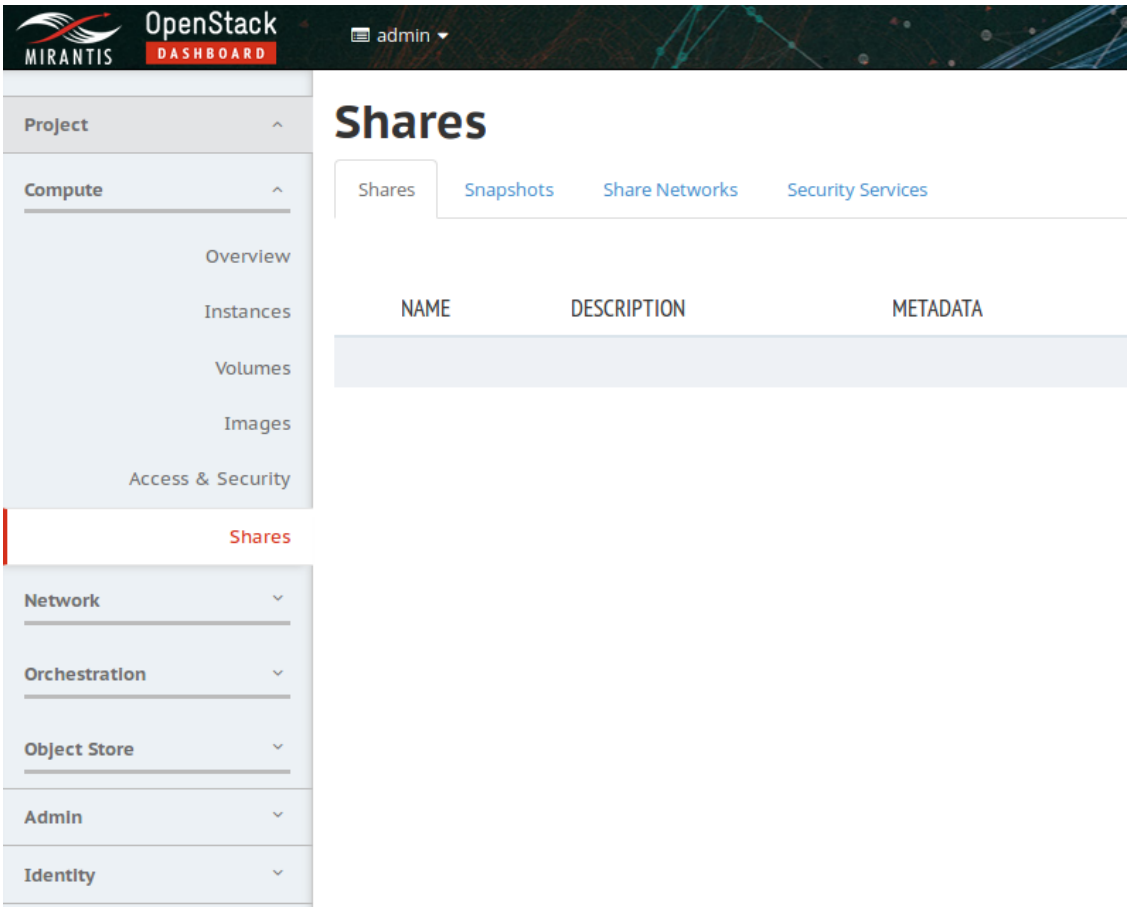
USE MANILA PLUGIN FOR FUEL

Using File Share as a Service possibility with Manila plugin for Fuel

Once you deploy an OpenStack environment with Manila plugin for Fuel, you can start using it both from CLI and Horizon. The topic of CLI usage is too big to put it into this User Guide and it well described here.

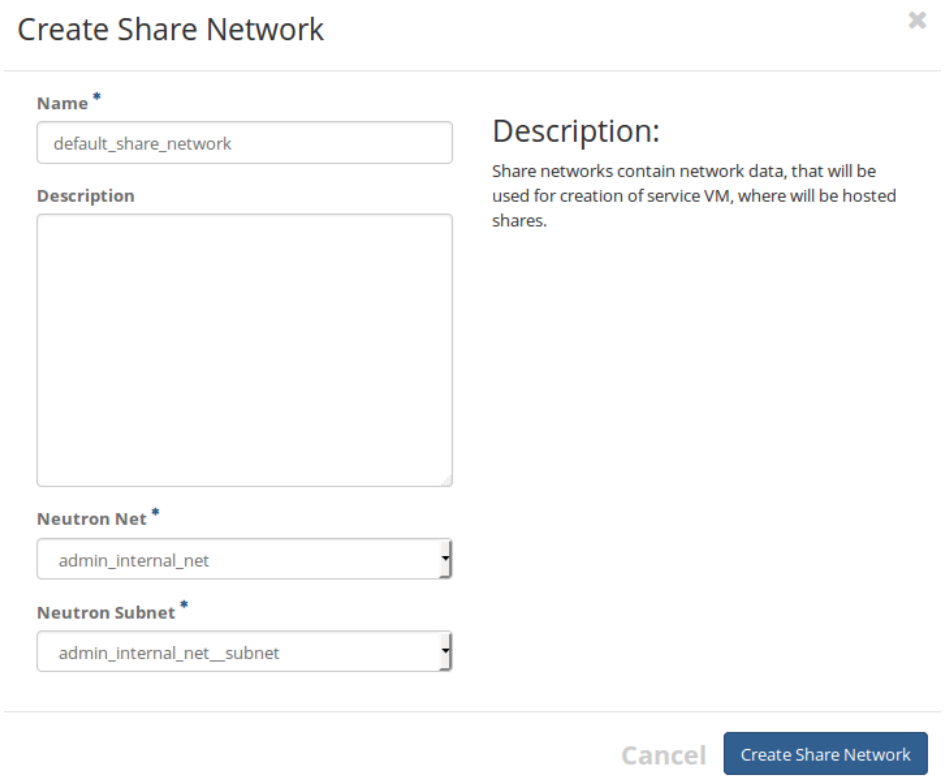
The Horizon usage is very obvious. You can notice that two new tabs appears on the Admin and Project/Compute sections:



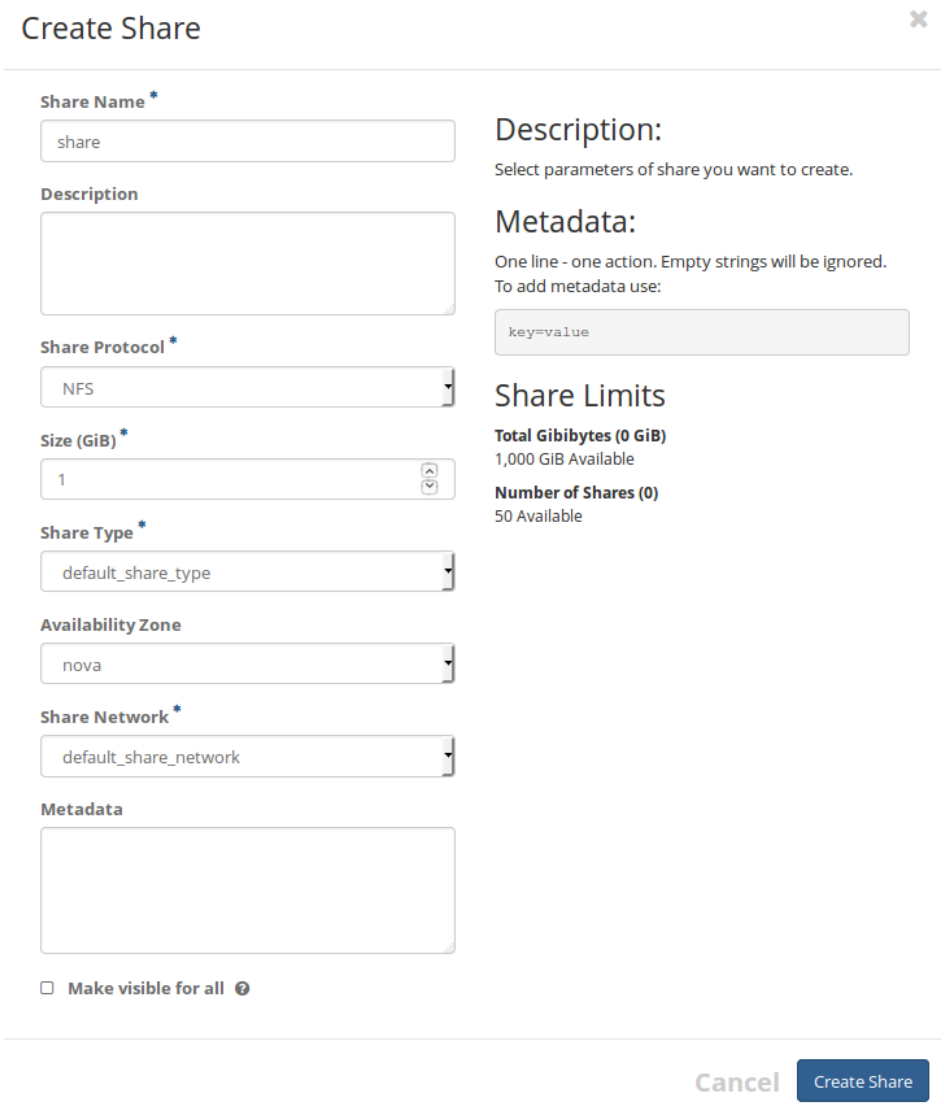

MIRANTIS
OpenStack
DASHBOARD
admin
Project
Compute
Overview
Instances
Volumes
Images
Access & Security
Shares
Network
Orchestration
Object Store
Admin
Identity
Shares
Shares
Snapshots
Share Networks
Security Services
NAME
DESCRIPTION
METADATA

Let's go over the basic usage of Manila:

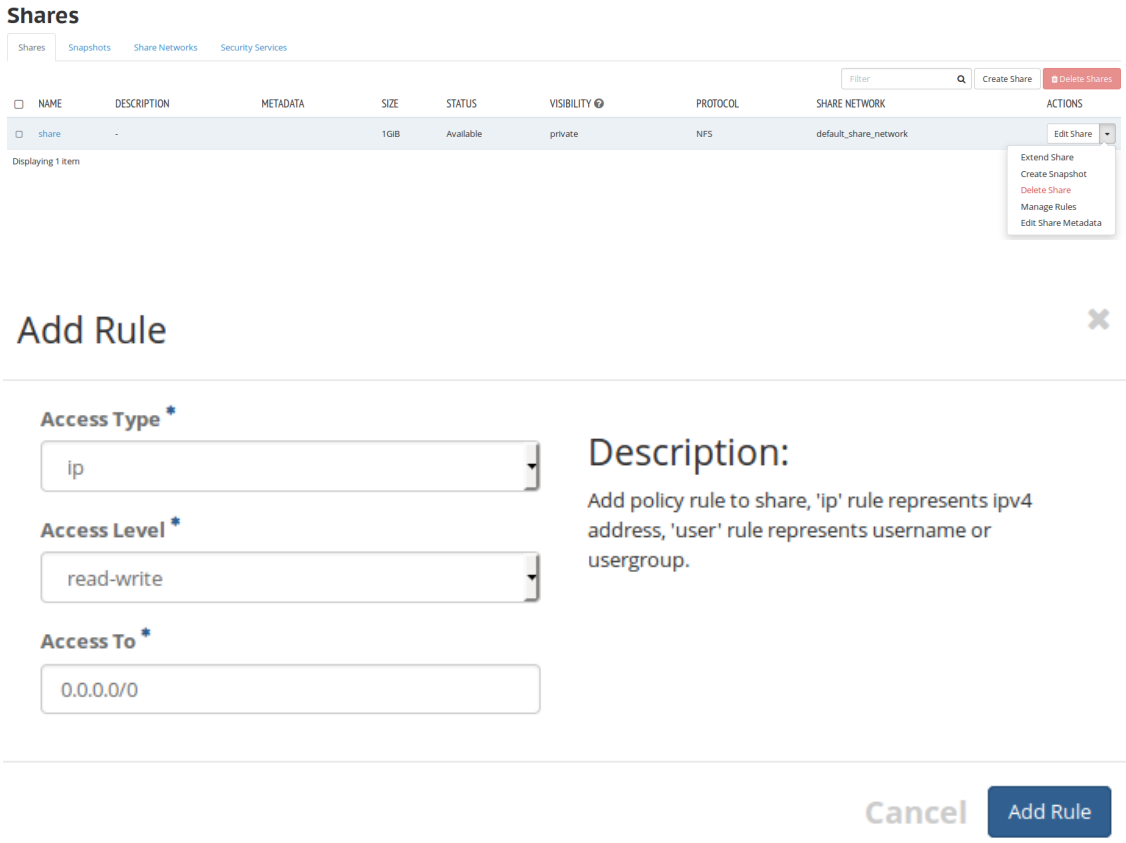
1. Firstly we have to create a share network for access to new share:



2. After that we can create new share:



3. When the share becomes available it should be configured for future usage. At least necessary to add new rule in order to allow mounting new share.



After that new share could be consumed in your environment.

Troubleshooting

This section contains a guidance on how to ensure that the Manila plugin is up and running on your deployed environment.

To find logs

The Manila places its log by convinient path. On controller:

- `/var/log/manila/manila-api.log`
- `/var/log/manila/manila-scheduler.log`

On a manila-share node:

- `/var/log/manila/manila-share.log`

On a manilsa-data node

- `/var/log/manila/manila-data.log`

To verify Manila configuragion files

Check that `/etc/manila` directory contains following files:

- `-rw-r--r-- 1 manila manila 1.8K Oct 19 02:35 api-paste.ini`
- `-rw-r--r-- 1 manila manila 1.3K Oct 19 02:35 logging_sample.conf`
- `-rw-r--r-- 1 root root 2.6K Oct 19 03:44 manila.conf`
- `-rw-r--r-- 1 manila manila 5.2K Oct 19 02:35 policy.json`
- `-rw-r--r-- 1 root root 989 Oct 19 02:35 rootwrap.conf`
- `drwxr-xr-x 2 manila manila 4.0K Oct 19 02:35 rootwrap.d`

To verify Manila services

Check output of the commands on any controller node:

```
# . /root/openrc
# manila service-list
```

All services should be in the *up* stage.

In case of using self signed certificates

Use the `--insecure` option for all console commands. For example:

```
# manila --insecure list
# manila --insecure type-create some_share_type True
```

Links

For more information about the software discussed in this document, see the following links:

- The Manila wiki page.
- The Manila plugin for Fuel repository.

- The LaunchPad project URL used to report bugs found in the plugin.