Murano User Guide

Murano User Guide

v0.4

Table of Contents

Chapter 1. How can I use Murano Service?

Murano is intended to get opportunity for non-experienced users to deploy different kinds of applications extremely simple. To use Murano you should be familiar with Openstack. Murano Dashboard is just a plugin to Openstack dashboard - Horizon. Please visit horizon user guide first to see how dashboard is organized and how to login into it. This document describes Murano Dashboard usage in detail.

Document change history

The following table describes the most recent changes:

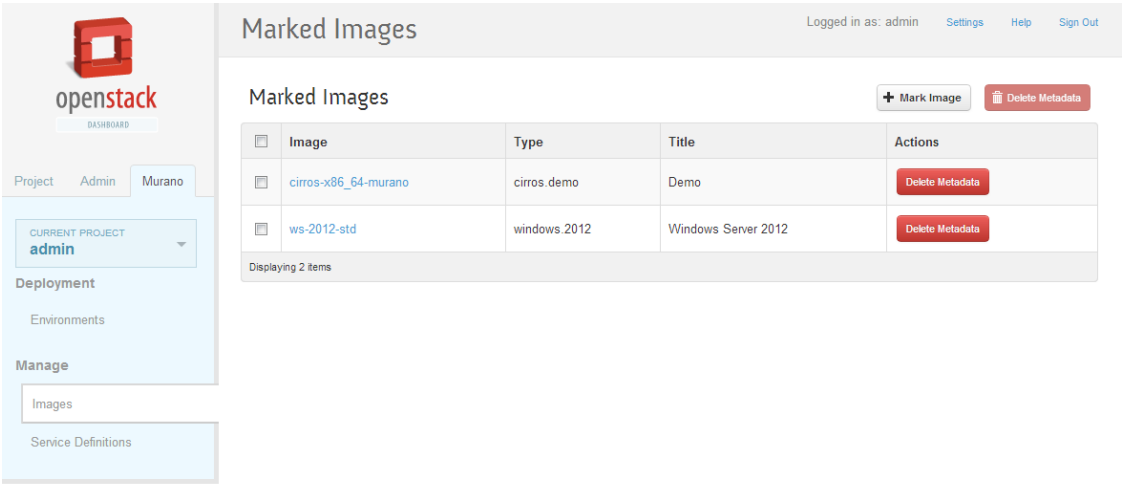
Revision Date	Summary of Changes
September. 4, 2013	• Update for Release-0.2
December. 9, 2013	• Update for Release-0.4

Chapter 2. Images

Murano uses preconfigured images for spawning instances. Image suitable for Murano should have Murano Agent installed at least. See here [http://murano-docs.github.io/latest/administrators-guide/content/ch03.html] how to build Murano images and upload it to Glance. Since Murano supports different operating systems it's not possible to use one image to build every single application. To classify images Metadata property of Glance image is used. Murano Dashboard allows to add that property from a separate view. Navigate to the Murano tab and go to `Images` panel.

Warning

To operate with Glance Images you need to have admin rights!



You are browsing Glance images with correctly set "murano_image_info" property - image metadata. From this view you are able to:

- Delete an existing image metadata with `"Delete Metadata"` button (only image's property get deleted, the image itself stays in Glance)
- Add new murano metadata to an image that is already registered in Glance by clicking on `"Mark Image"` button

Take a look at the "Mark Image" form:

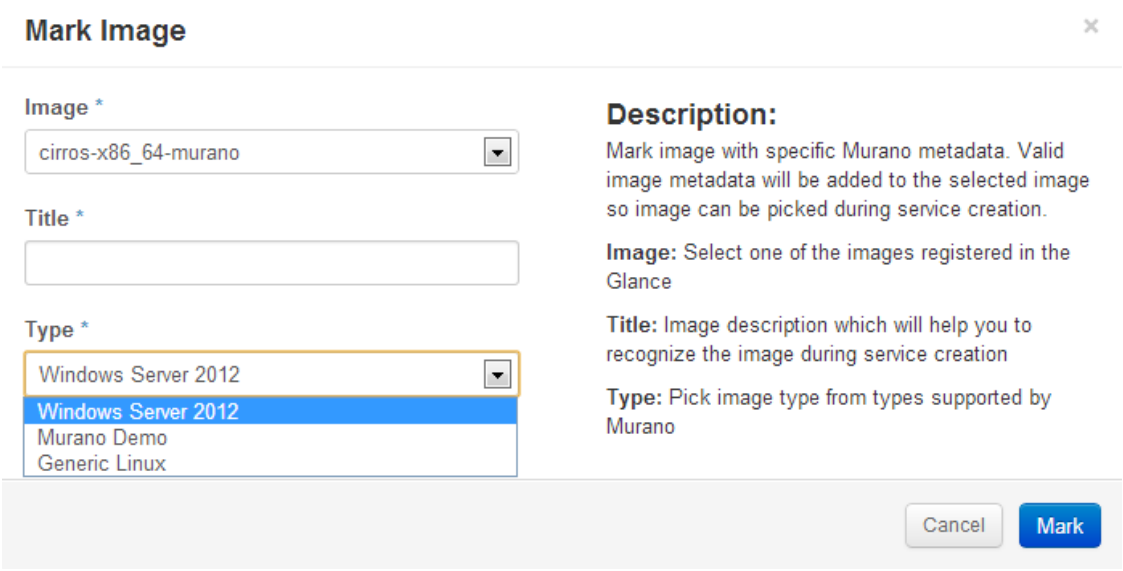


Image	From all images registered in Glance pick one that was created with Murano Image Builder.
Title	Set name for a selected image to recognize it easier during service creation.
Type	For now there are three supported types: • Windows Server 2012 - used to build Windows-based applications • Murano Demo - light-weight Linux Cirros image with Murano Demo Agent • Generic Linux - used to build Linux-based applications

Chapter 3. Environments

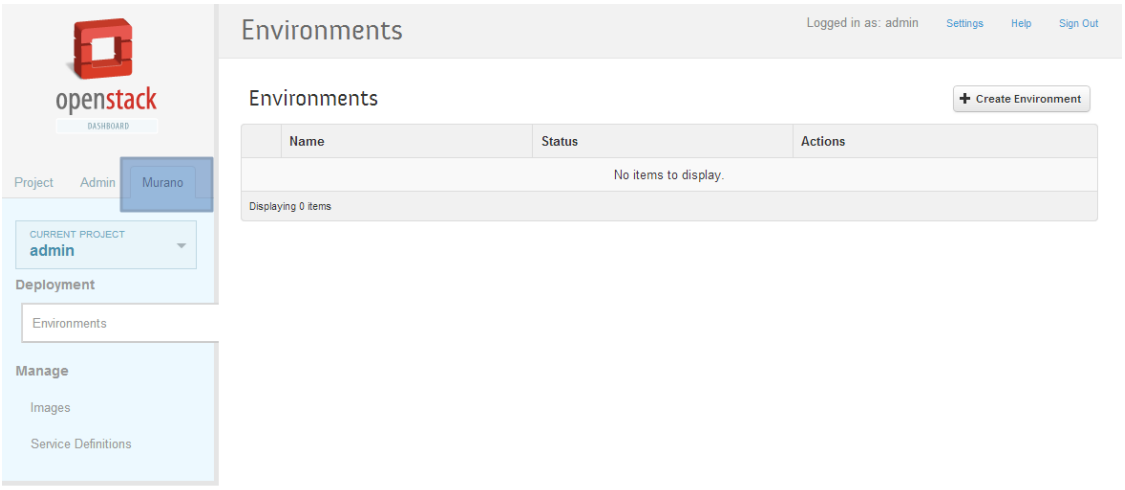
Murano Environment - a virtual unit to store different services (applications). They can be connected with each other or be individual. The following actions with environment are available:

- Create;
- Delete;
- Deploy.

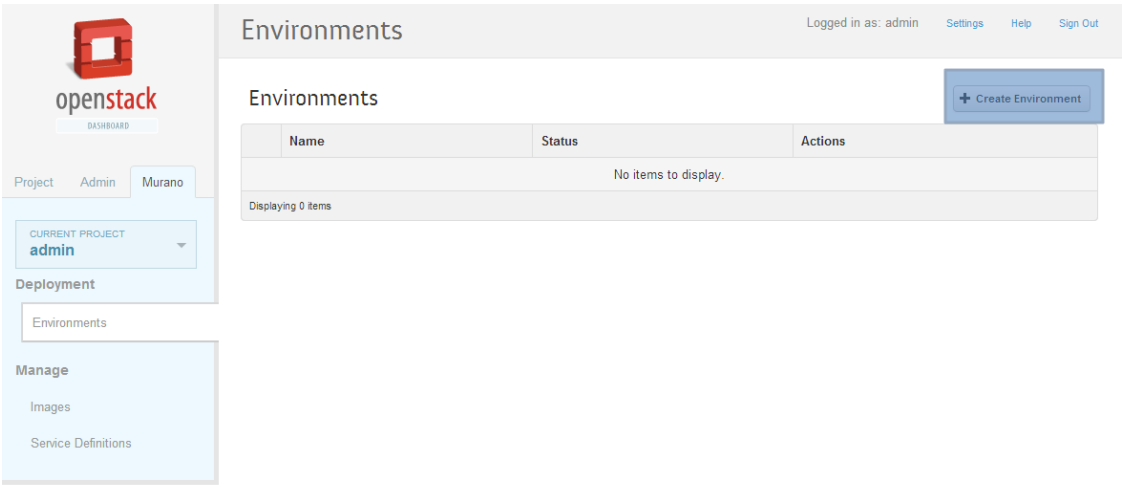
More information about actions with environment are described in this chapter.

Creating environment

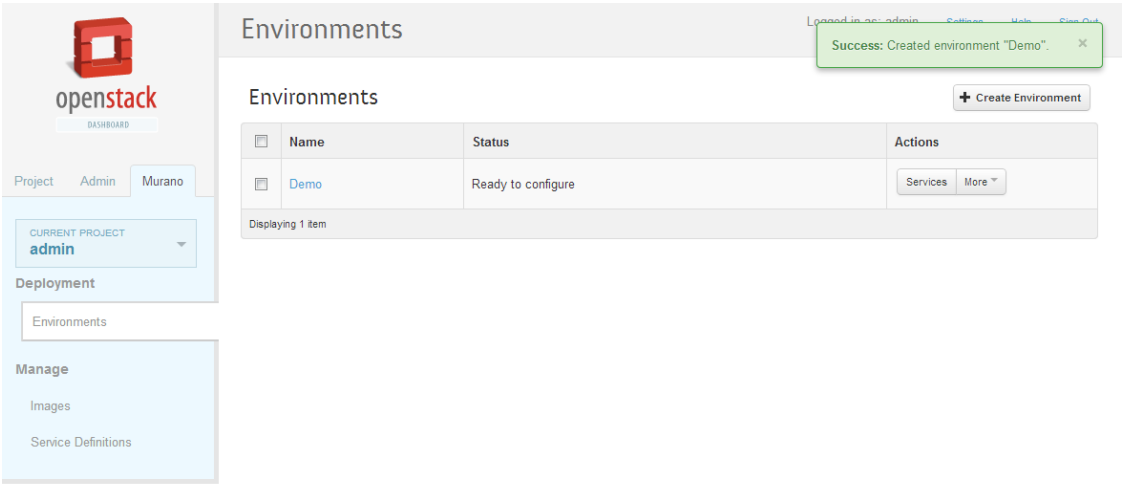
Once you installed all Murano components and logged-in to horizon dashboard successfully you will see Murano tab:



First thing you need to do is to create an environment. To do this, navigate to the "Environments" page and click the "Create Environment". After setting name to your virtual environment it will be created.

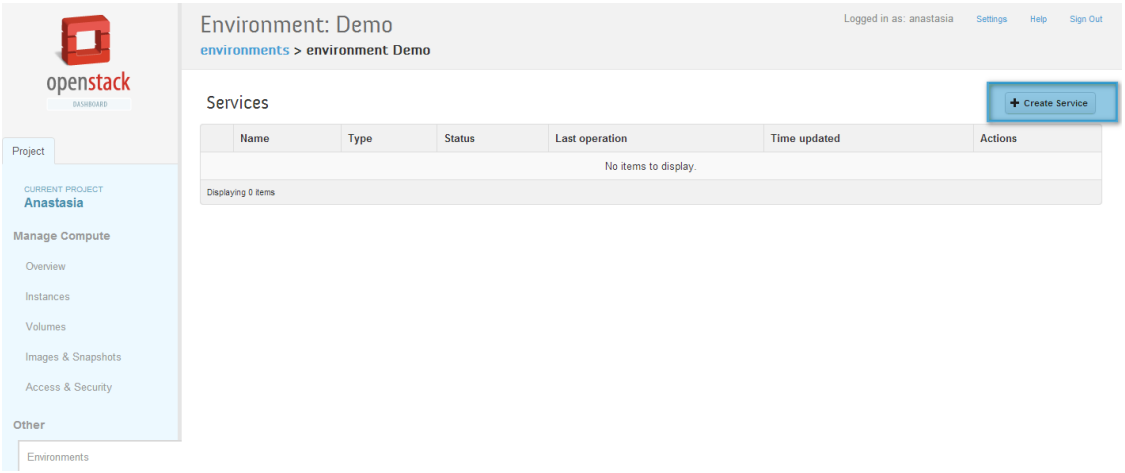


Just created environment has status *Ready to configure*.

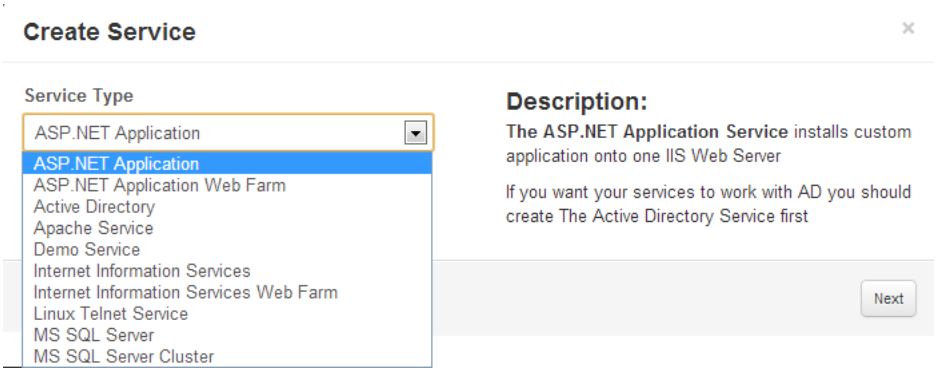


Creating service prototype

All services should be created in the context of Environment. After Environment is created, you need to create service prototypes and then deploy the Environment. When deploy process is done instances with your services will be spawned in Openstack. To create a service prototype navigate to page with the list of environment services by clicking on the environment name (or on the "Services" button) and then, click the "Create Service" button.



You have an opportunity to create one of the following services:

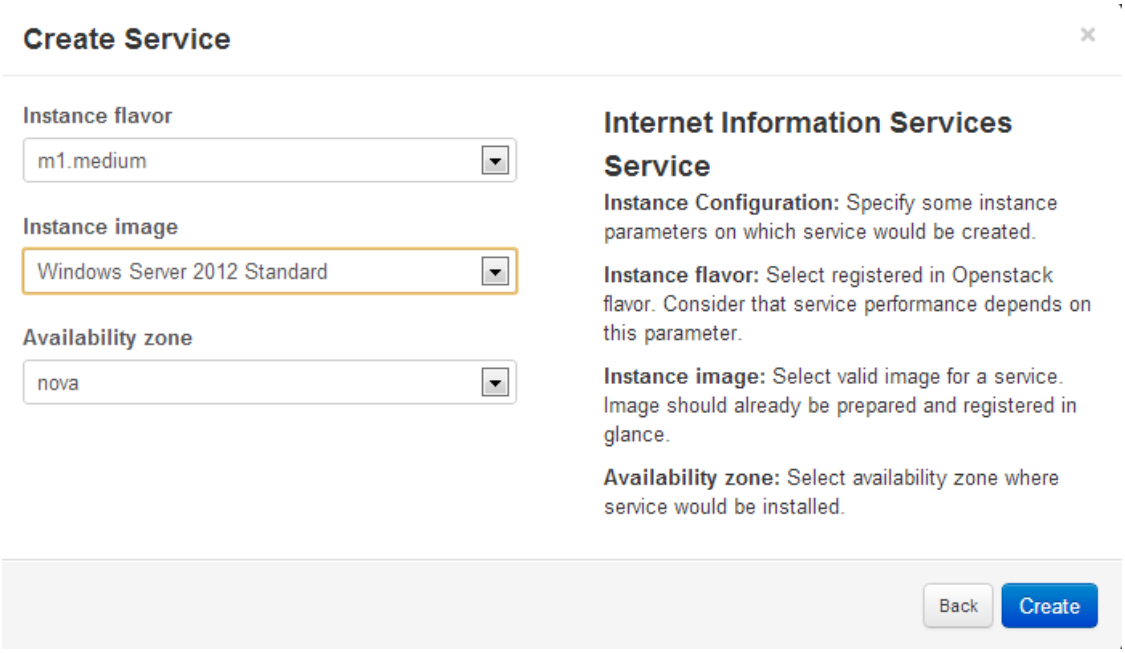


Once you choose service that you want to create click "Next" and fill the form. Forms for each service are specific. To see more information about filling the form for a specific service follow one of the link below:

- **ASP.NET Service:** is a server-side Web application framework designed for Web development to produce dynamic Web pages. Service is able to install custom application onto one IIS Web Server. Murano installs all needed components and makes proper configuration.
- **ASP.NET Farm Service:** ASP.NET Farm Service installs a custom application on a load-balanced array of IIS servers.
- **Active Directory:** Active Directory is a directory service implemented by Microsoft for Windows domain networks. In one installation, in addition to primary Domain Controller, you can add optional count of secondary Domain Controllers. Any other services you are intending to create can be joined to that domain.
- **Apache Service:** Apache HTTP Server is a web server application notable for playing a key role in the initial growth of the World Wide Web. This service can install Apache with PHP module or without it.
- **Demo Service:** Installs test service that demonstrates how to Murano interconnects with Murano Demo Agent. For this service light-weight Linux Cirros image can be used.
- **Internet Information Service:** IIS is a web server and a set of feature extension modules.
- **Internet Information Web Farm Service:** Murano installs the Web Farm Framework on the controller server, configures the primary server and prepares the secondary servers. In addition load balancer is installed to monitor service statuses.

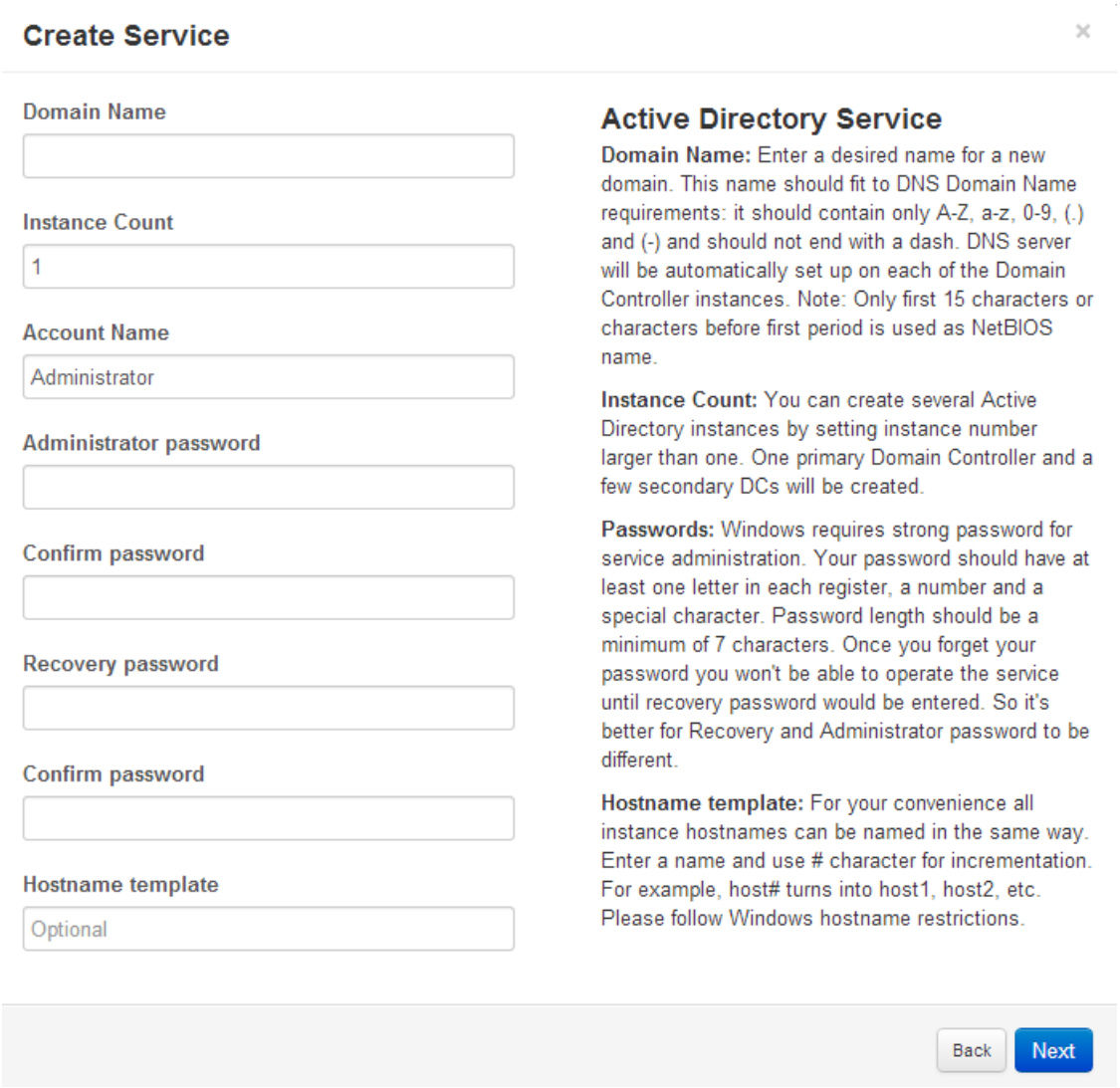
- **Linux Telnet:** Telnet - is a network protocol used to provide a bidirectional interactive text-oriented communication facility using a virtual terminal connection.
- **MS SQL Service:** Microsoft SQL Service is a relational database management system.
- **MS SQL Server Failover Cluster:** Murano installs all needed components and configures your SQL Server Cluster the way you want.

On the last step of creating service prototype you have opportunity to set the hardware flavor of the instance which will be created - and the image with the operating system, which will be installed on the instance. Also you may select availability zone, if there are more then one in your environment.



Active Directory

After you picked the Active Directory service in service creation wizard you'll see the following form:

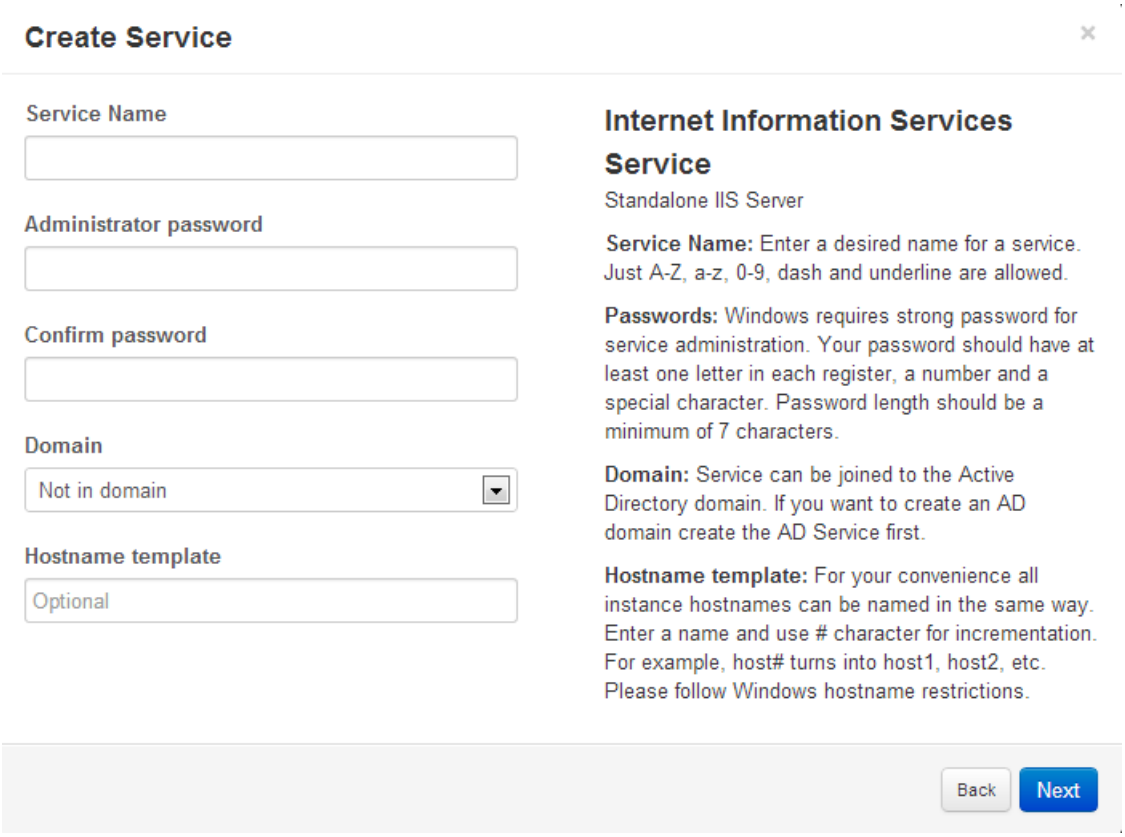


Enter or select values for this fields:

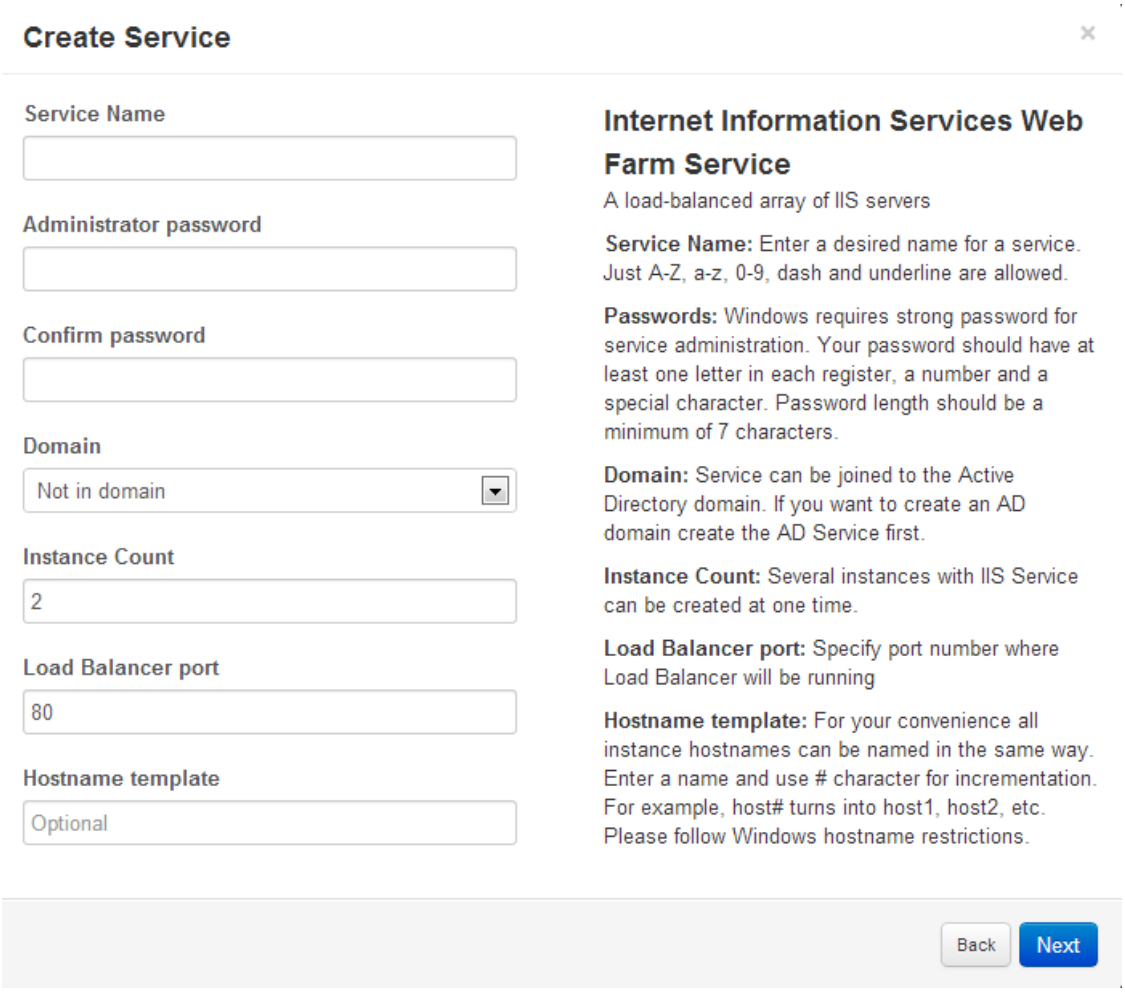
Domain name	Enter a desired name for a new domain. This name should fit to DNS Domain Name requirements: it should contain only A-Z, a-z, 0-9, (.) and (-) and should not end with a dash. DNS server will be automatically set up on each of the Domain Controller instances. Period characters are allowed only when they are used to delimit the components of domain style names. Single-level domain is not appropriate. Note: Only first 15 characters or characters before first period is used as NetBIOS name.
Instance Count	You can create several Active Directory instances by setting instance number larger than one. One primary Domain Controller and a few secondary DCs will be created.
Account Name	You account will have Active Directory administrator rights. So initial value is "Administrator" but you can change it to any name you like.
Administrator password	Windows requires strong password for service administration. Your password should have at least one letter in each register, a number and a special character. Password length should be a minimum of 7 characters.

	Once you forget your password you won't be able to operate the service until recovery password would be entered. So it's better for Recovery and Administrator password to be different.
Confirm password	Password confirmation is required.
Recovery password	Restrictions are the same as for the Administrator password. Please provide password that is different from Administrator. It's not required for form validation though.
Confirm password	Password confirmation is required.

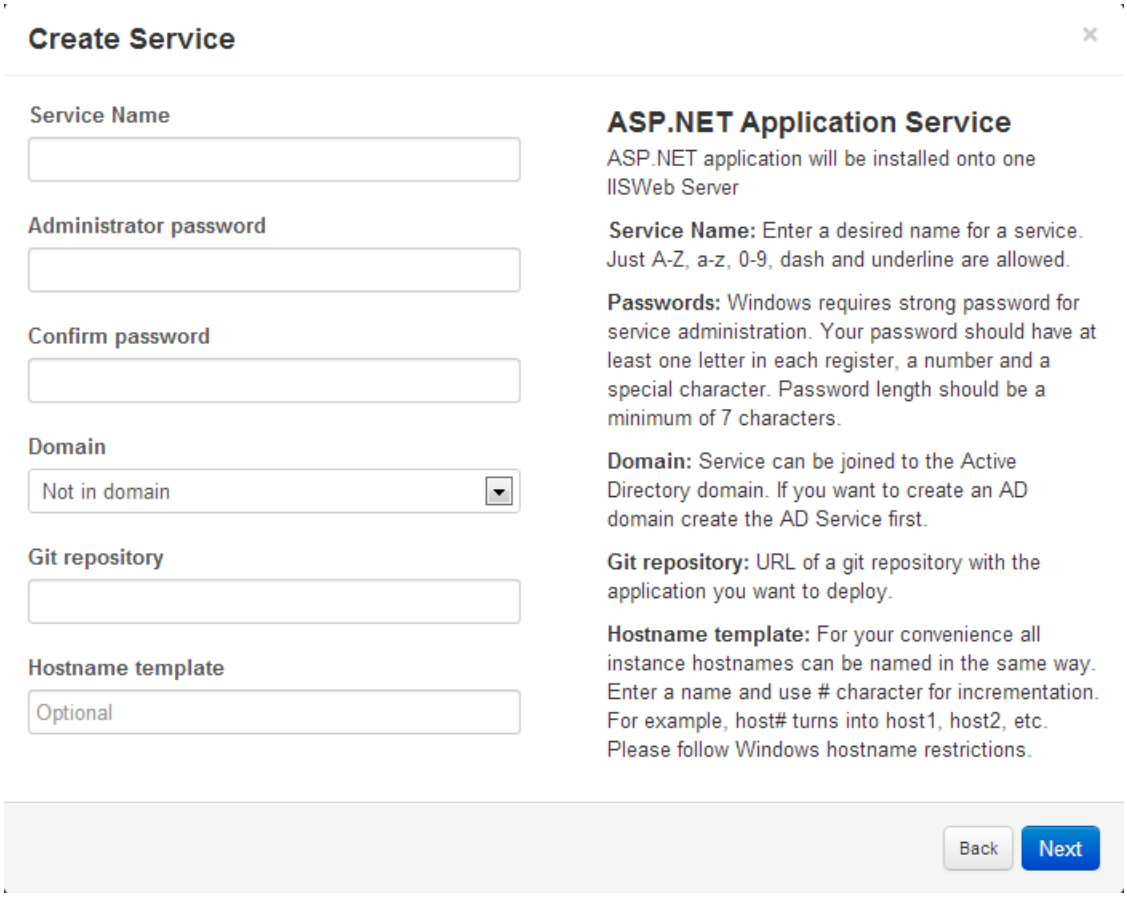
Internet Information Service



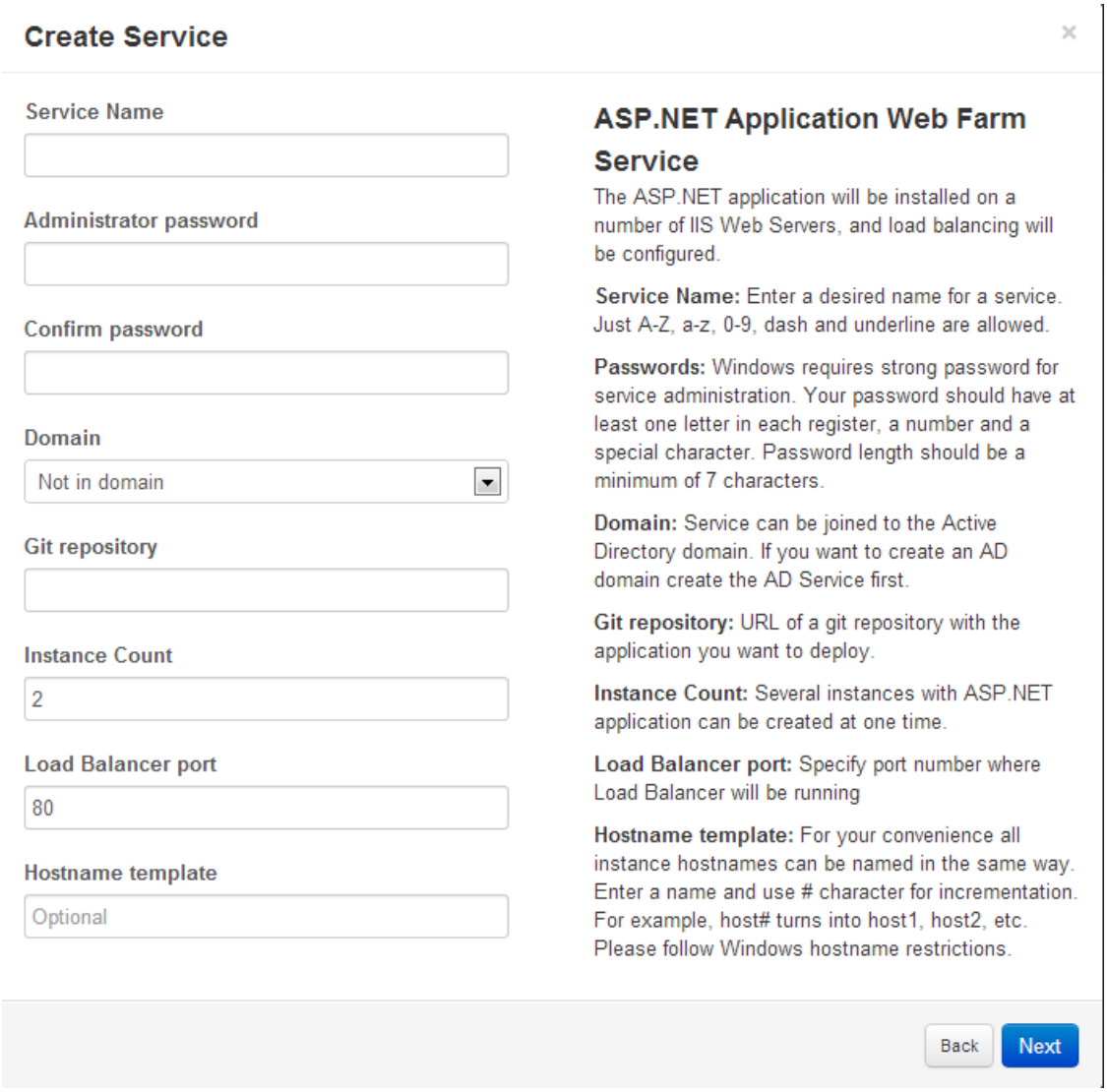
Internet Information Web Farm Service



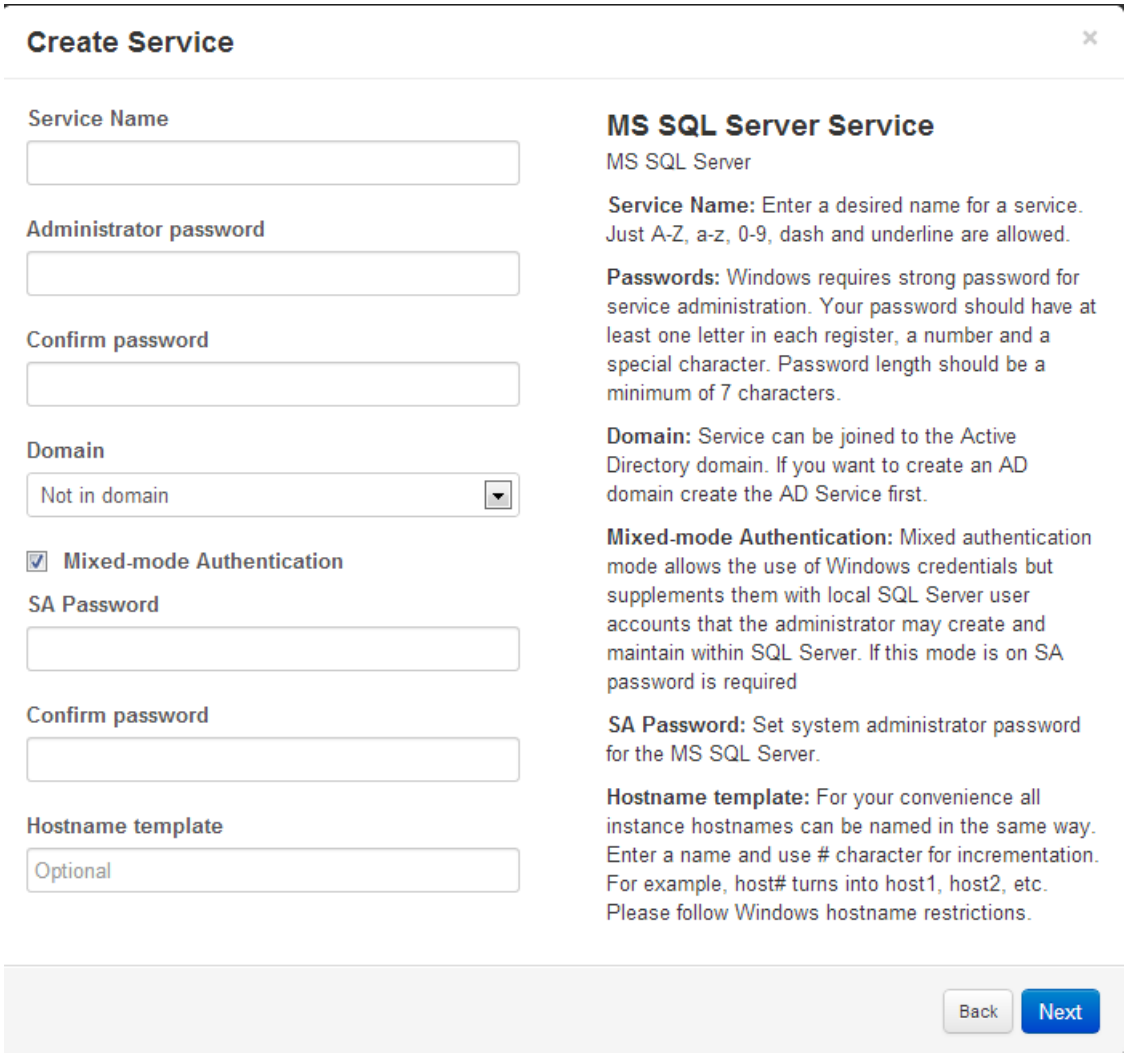
ASP.NET Service



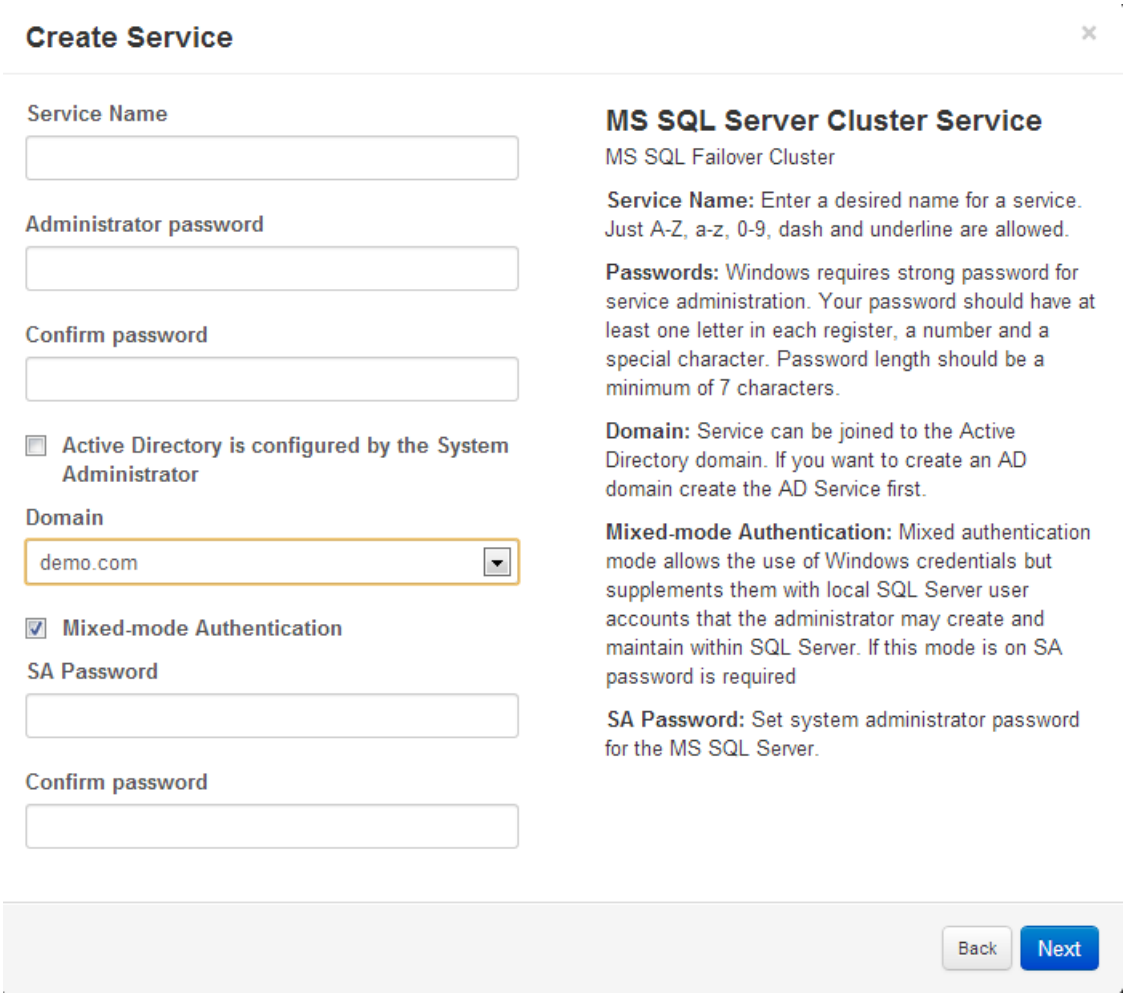
ASP.NET Farm Service



MS SQL Service



MS SQL Server Failover Cluster

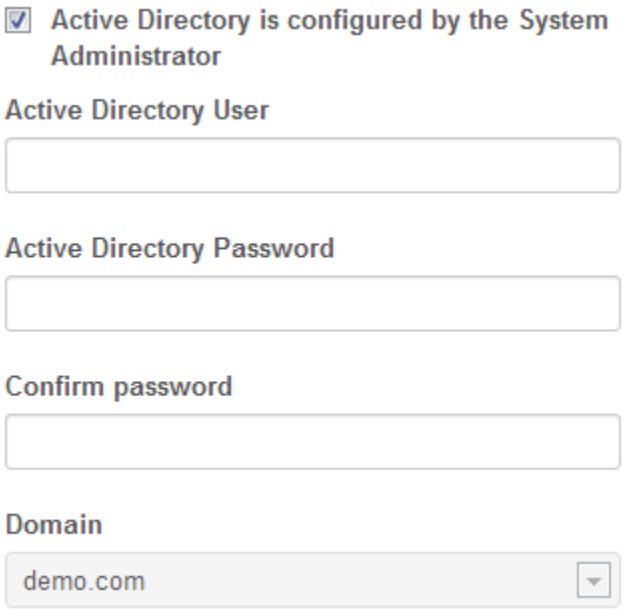


Enter or select values for the following fields:

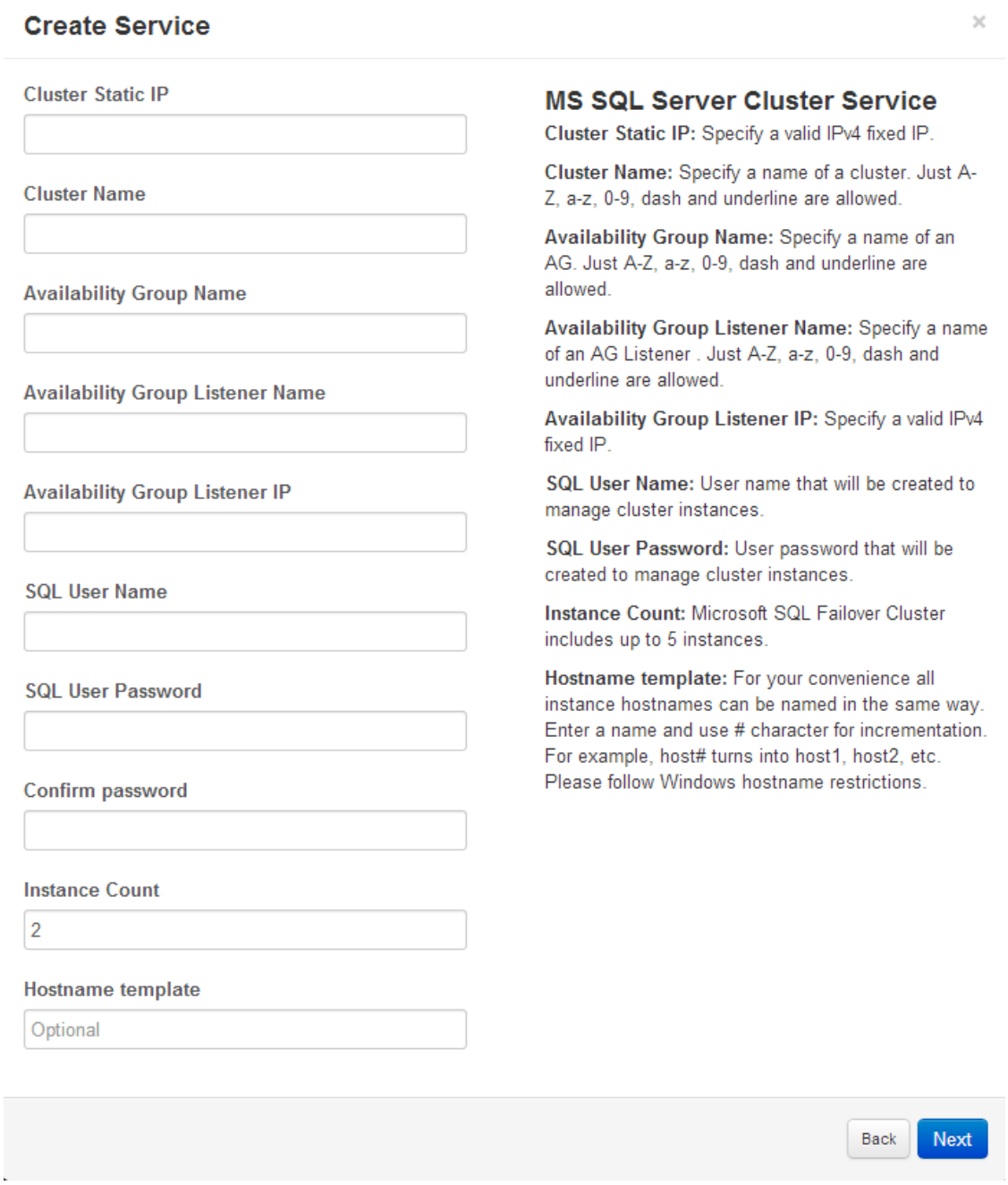
Service Name	Enter a desired name for a service. Just A-Z, a-z, 0-9, dash and underline are allowed.
Administrator password	Windows requires strong password for service administration. Your password should have at least one letter in each register, a number and a special character. Password length should be a minimum of 7 characters.
Confirm password	Password confirmation is required.
Active Directory is configured by the System Administrator	Enable this option only if you have properly configured rules that will include service to the domain, that already exists in you environment. Once you set this option to true, additional fields will appear. (See information below)
Domain	Service should be joined to the Active Directory domain. Please, create Active Directory Service prototype first.
Mixed-mode Authentication	Mixed authentication mode allows the use of Windows credentials but supplements them with local SQL Server user accounts that the administrator may create and maintain within SQL Server. If this mode is on SA password is required

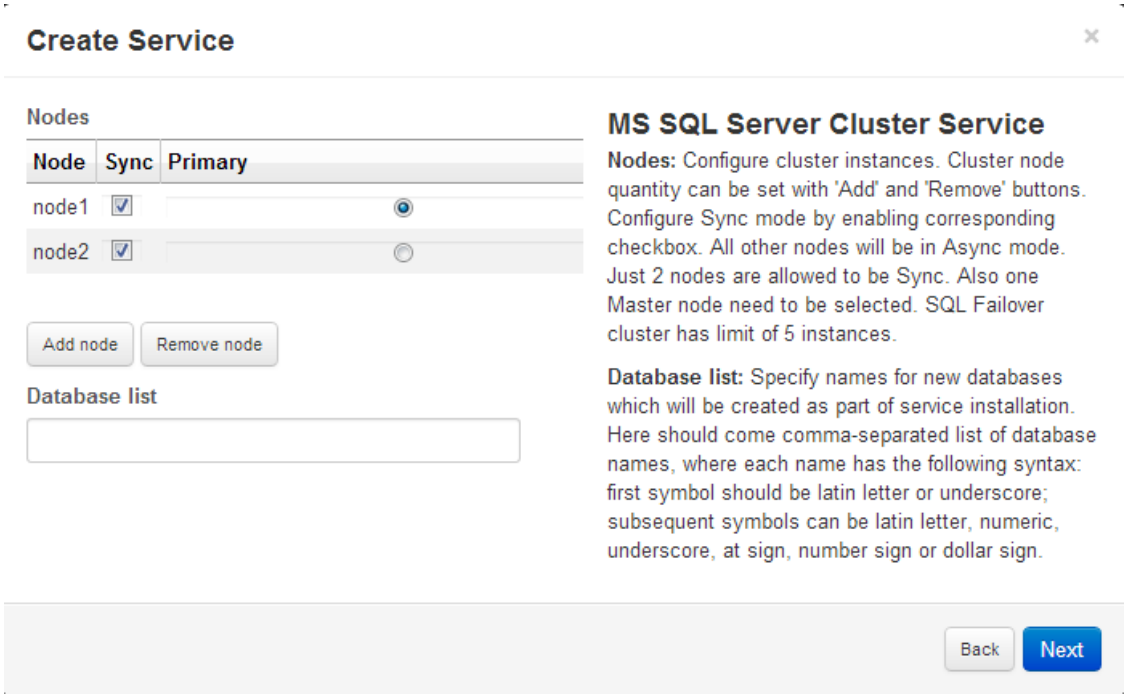
SA Password	Set system administrator password for the MS SQL Server. Password requirements are the same as previous.

In case pre-configured AD is enabled following fields will appeared:

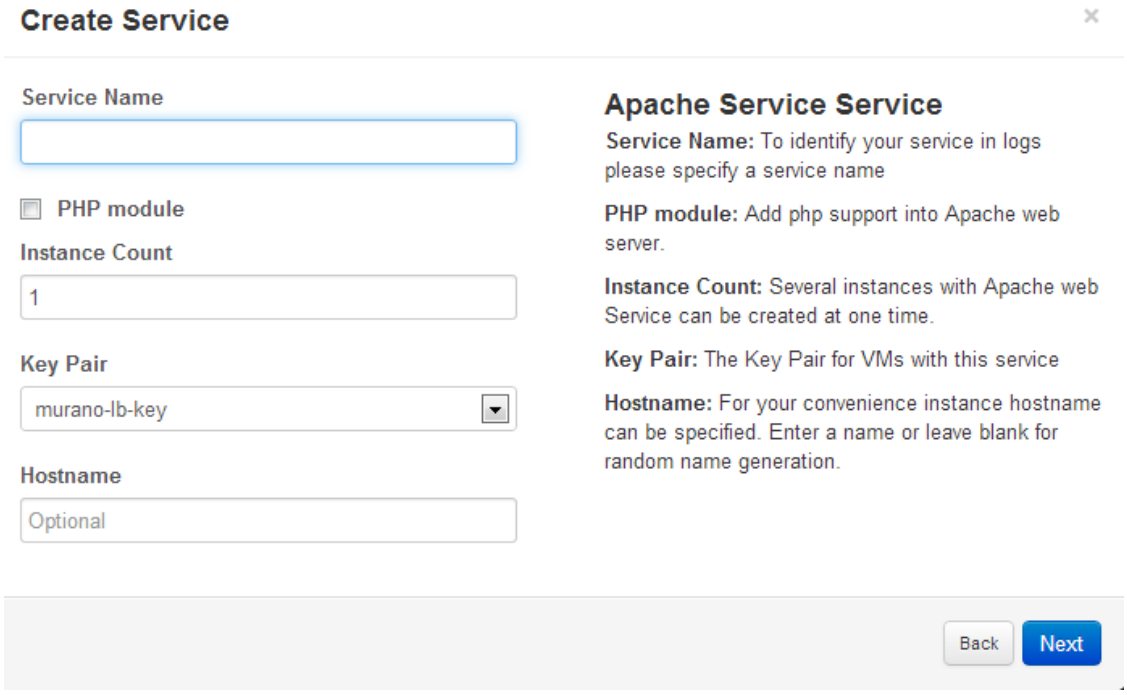


Active Directory User Active Directory Password	Specify administrator user credentials to the existent AD domain (to which service service will be join according to your system automation setup)


Create Service
Cluster Static IP
Cluster Name
Availability Group Name
Availability Group Listener Name
Availability Group Listener IP
SQL User Name
SQL User Password
Confirm password
Instance Count
2
Hostname template
Optional
MS SQL Server Cluster Service
Cluster Static IP: Specify a valid IPv4 fixed IP.
Cluster Name: Specify a name of a cluster. Just A-Z, a-z, 0-9, dash and underline are allowed.
Availability Group Name: Specify a name of an AG. Just A-Z, a-z, 0-9, dash and underline are allowed.
Availability Group Listener Name: Specify a name of an AG Listener . Just A-Z, a-z, 0-9, dash and underline are allowed.
Availability Group Listener IP: Specify a valid IPv4 fixed IP.
SQL User Name: User name that will be created to manage cluster instances.
SQL User Password: User password that will be created to manage cluster instances.
Instance Count: Microsoft SQL Failover Cluster includes up to 5 instances.
Hostname template: For your convenience all instance hostnames can be named in the same way. Enter a name and use # character for incrementation. For example, host# turns into host1, host2, etc. Please follow Windows hostname restrictions.
Back
Next



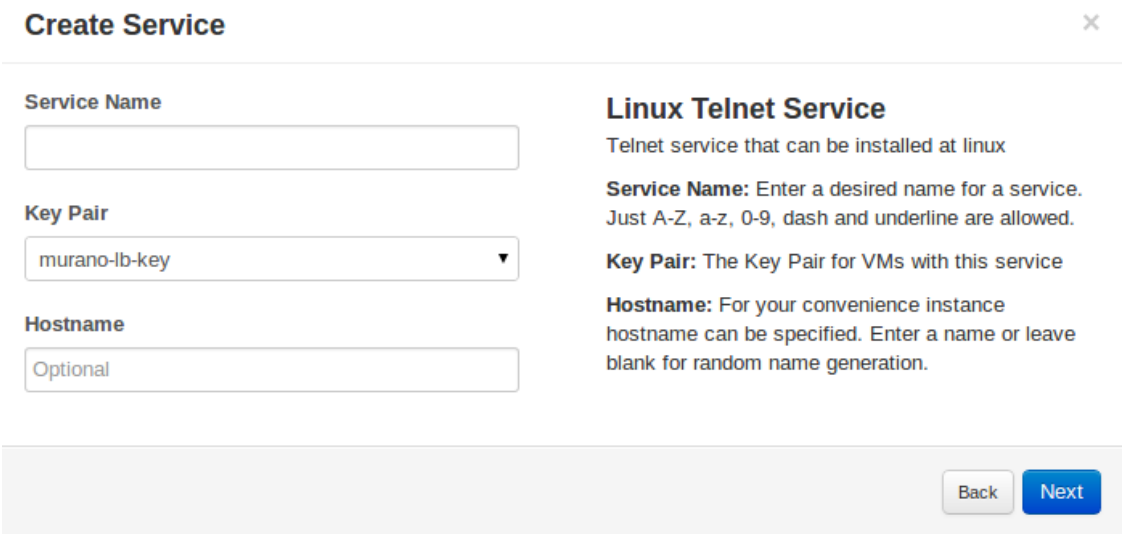
Linux Apache Service



Service Name	Enter a desired name for a service. Just A-Z, a-z, 0-9, dash and underline are allowed.
PHP module	PHP module can be installed with Apache. Enable or disable this options via checkbox.
Key Pair	Select SSH Key Pair from the existing to control access to your instance.

Hostname	Machine hostname. If blank - random name will be used.

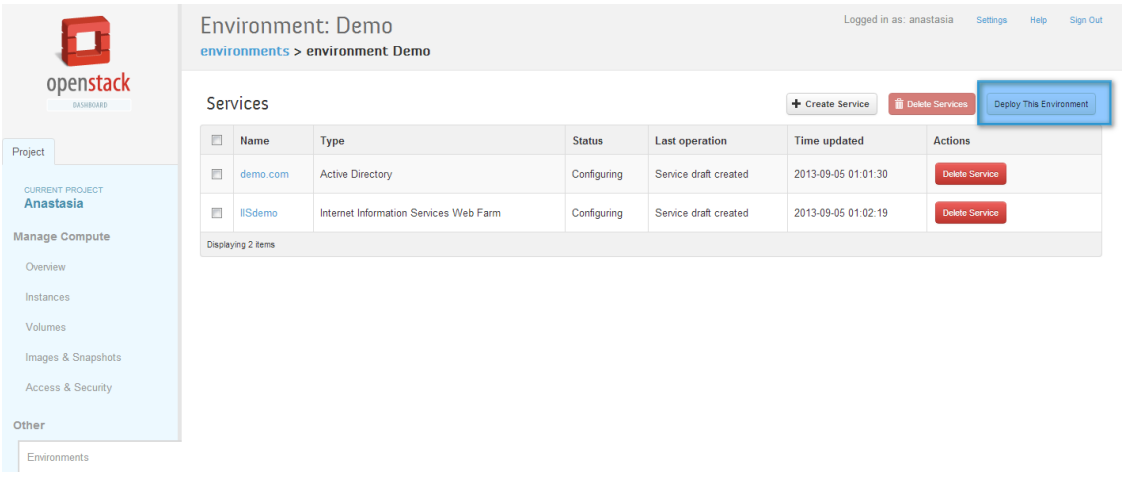
Linux Telnet Service



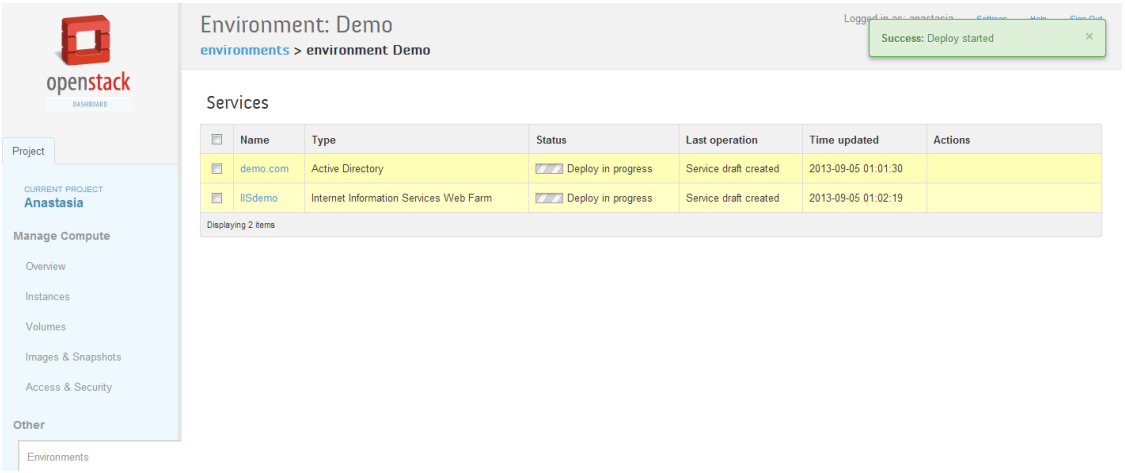
Service Name	Enter a desired name for a service. Just A-Z, a-z, 0-9, dash and underline are allowed.
Key Pair	Select SSH Key Pair from the existing.
Hostname	Machine hostname. If blank - random name will be used.

Deploying environment

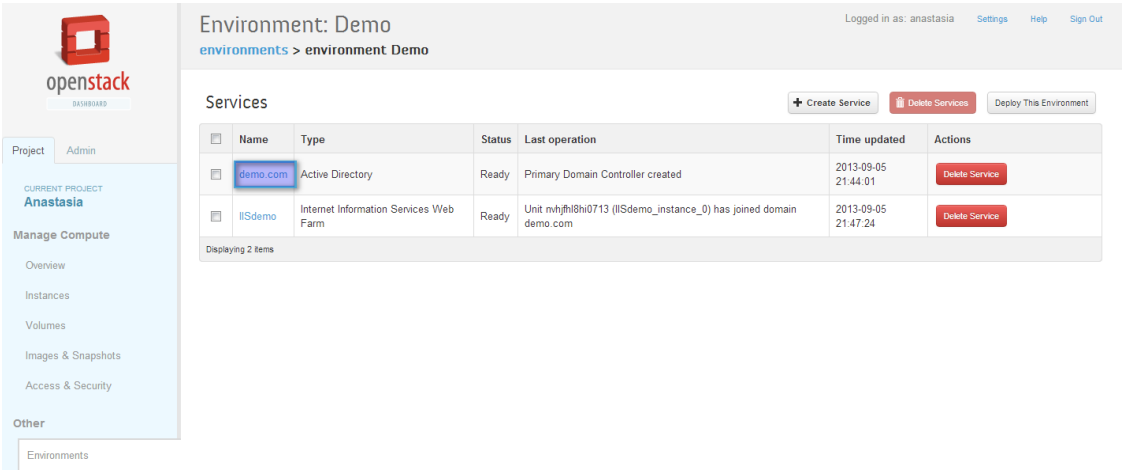
Once all services are prepared you can send environment to deploy. Just press the "Deploy Environment" button.



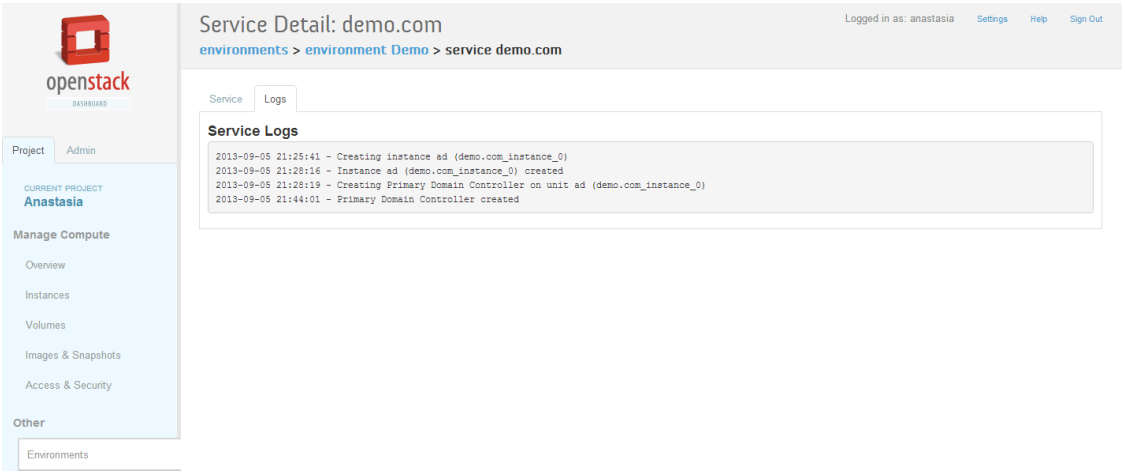
And you'll see a message about successful start of deploying your services in Openstack. Since now all you have to do is just wait for a little bit while Murano installing and configuring your services.



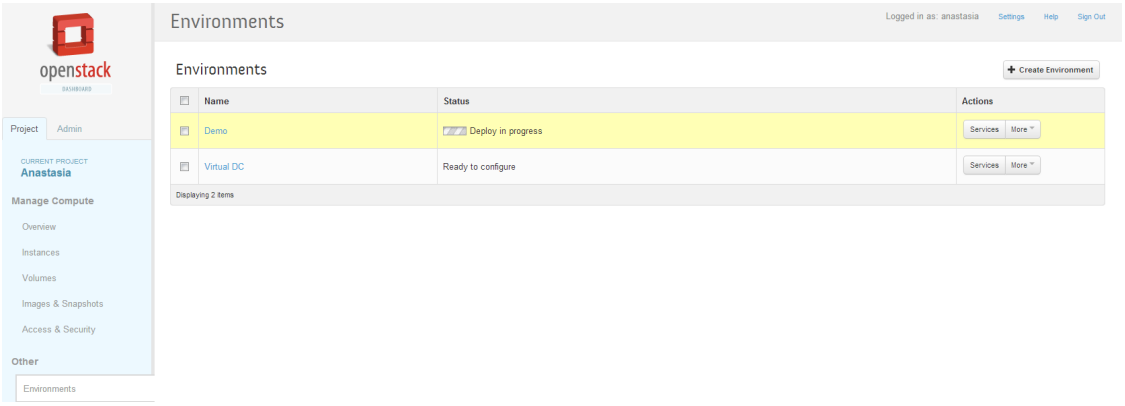
You can monitor deploying process. Just go to the Log tab on service detailed page, where you can get by clicking on the service name.



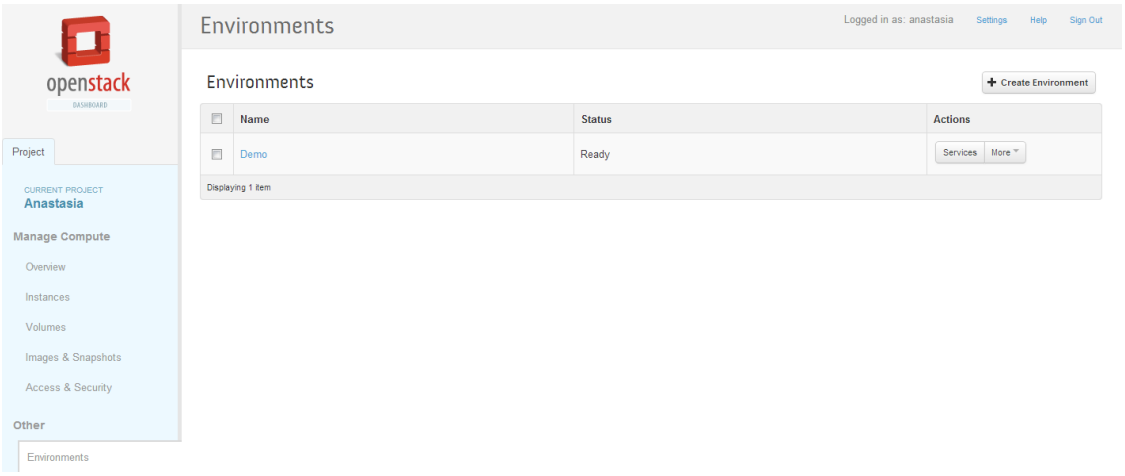
And now you can see installation progress.



As long as installation and configuration are in progress, environment is in *Deploy in progress* state. Depending on how many services you are deploying or how many nodes in your cluster, process of spawning instances, installation and post installation settings takes from 10 minutes up to one hour.

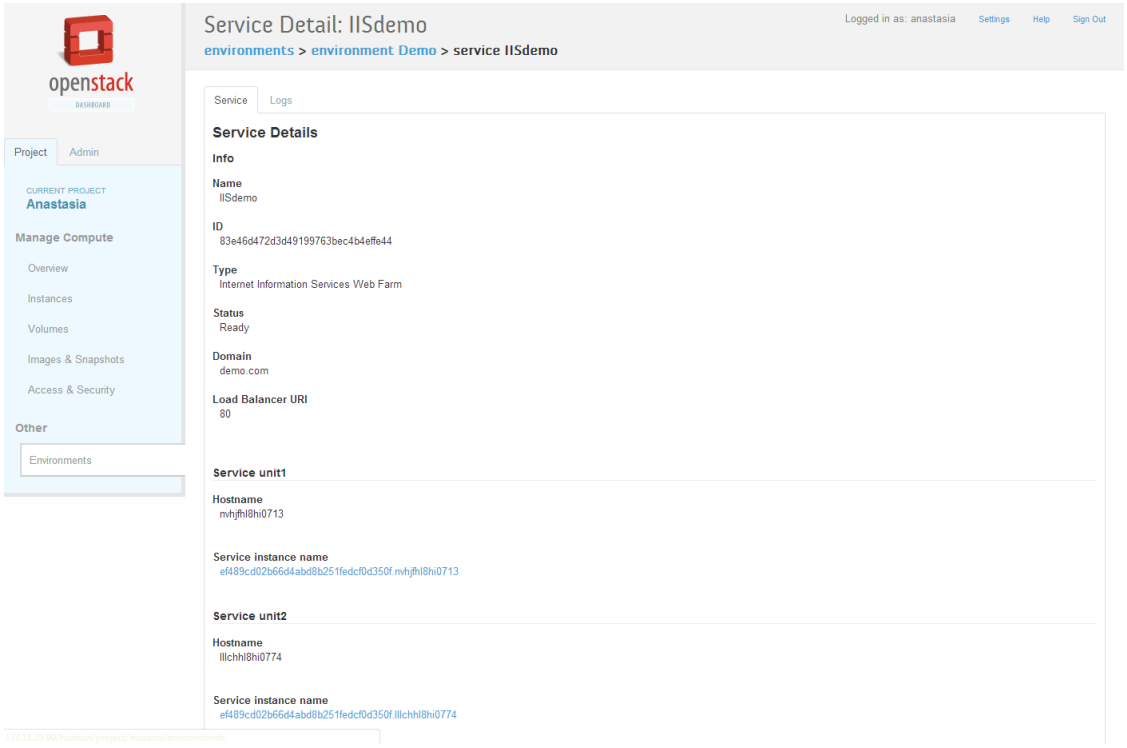


If installation process finished without any errors, environment changes its status to *Ready:*

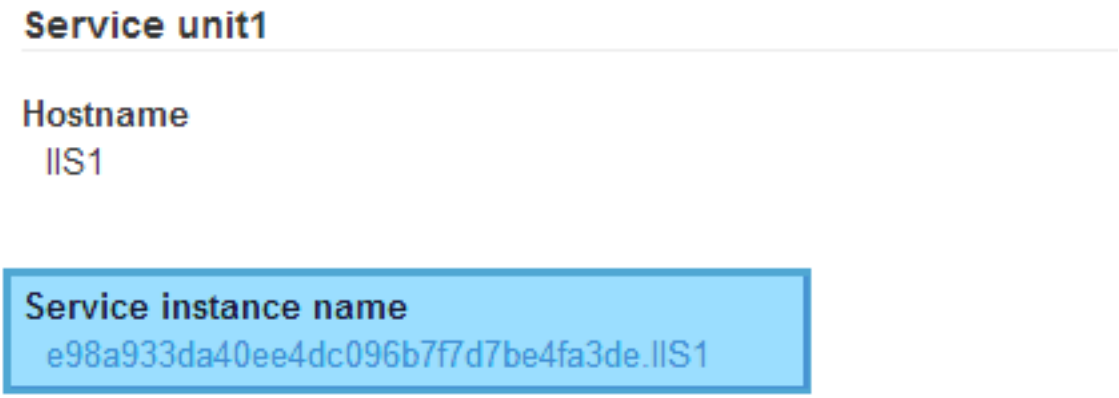


Working with deployed environment

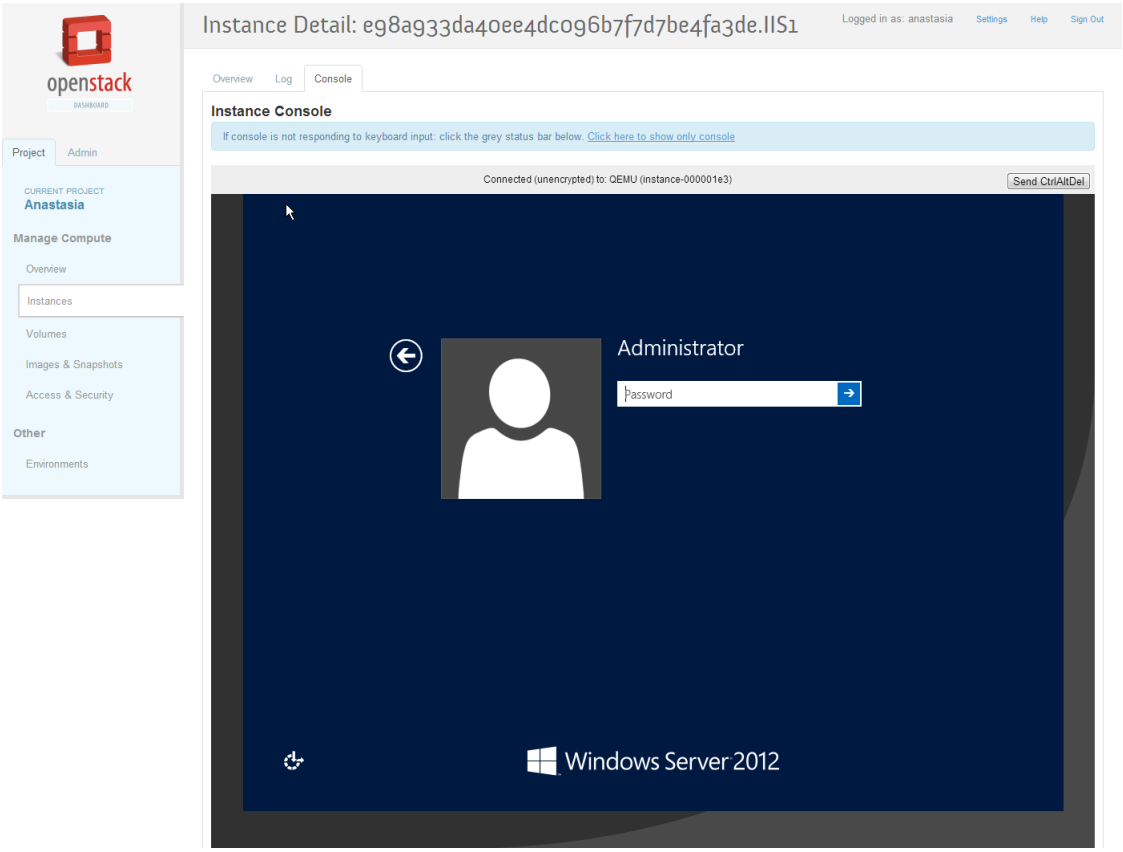
Congratulations! After some time waiting you are able to operate with the services. To get information about installed services, navigate to service detailed page. To do that click on the environment name and then on the name of the service you what to know about.



Now you are seeing general information about the service in terms of Murano Environment. To get information about the instance in Openstack terms follow the link on service instance name.



You can login to the virtual machine directly from the horizon (if your Openstack installation allows you) or by RDP protocol.



There are more things you can do with Murano Environment:

- Add new services and deploy it again;
- Delete outdated and unnecessary environments or services;
- Browse deployment history and service installation logs;
- Rename your environment.

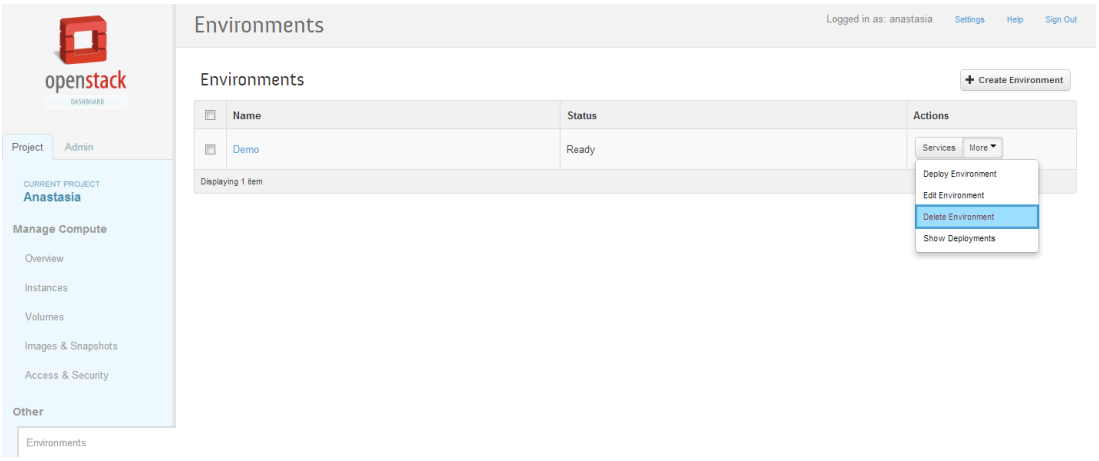
Redeploying Murano Environment

Murano gives an opportunity to supplement already deployed environment. Thus if you already deployed the Active Directory service and want to add any other services just create desired service prototype and click the "Deploy This Environment" button. During service prototype creation you can join this service to the existent Active Directory domain.

Deleting

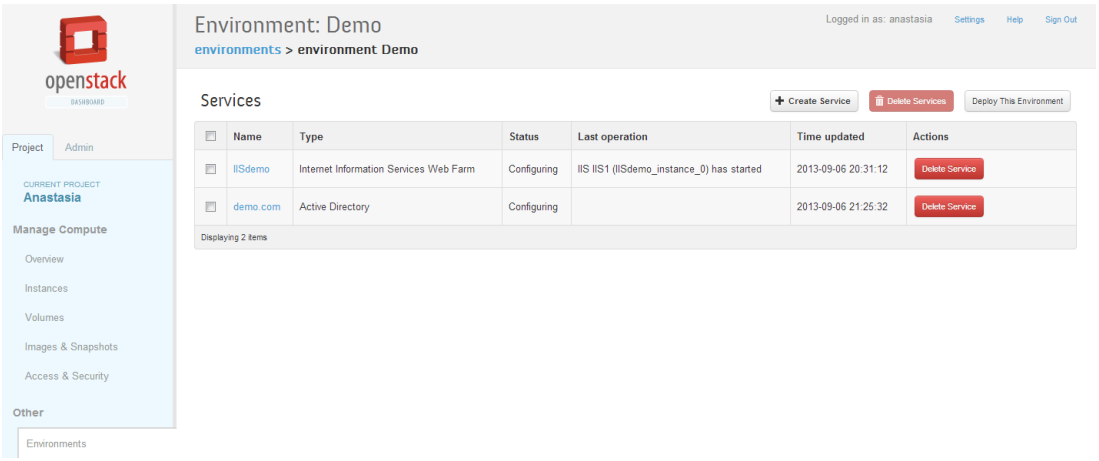
Services as well as environments can be easily deleted.

- To delete an environment go to the environment index page and click "More" -> "Delete Environment" in Actions column of ready to delete environment.



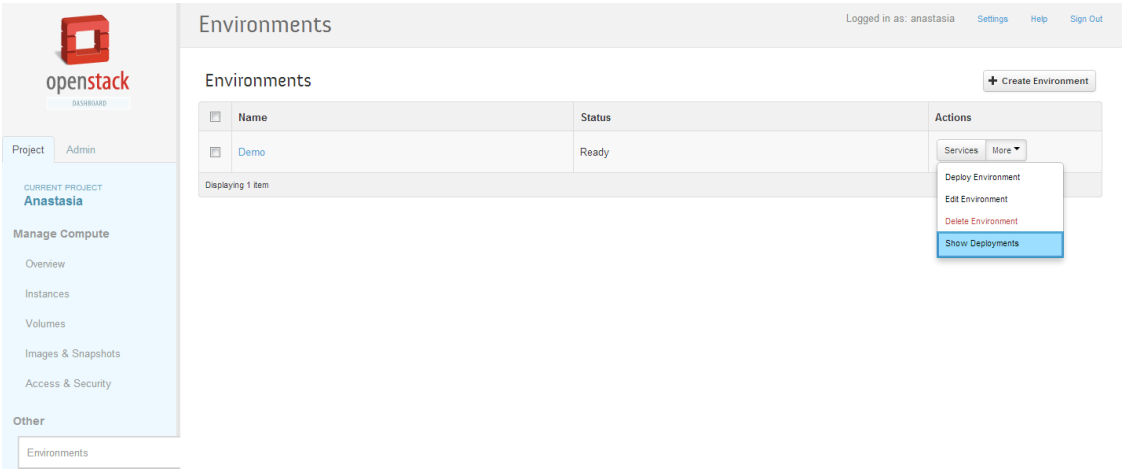
Environment deletion means to kill all services with instances on which they are installed. Instances will be scheduled to delete right after you choose the "Delete Environment" action.

- To delete a service go service list page and click the "Delete Service" button in Actions column. **Note:** If you are deleting service that was already deployed you'll need to *Deploy* the environment again by pressing corresponding button. In case you want to delete service prototype - it has *"Service draft created"* in the *Last operation* column (see the screenshot below) - changes applies right away.

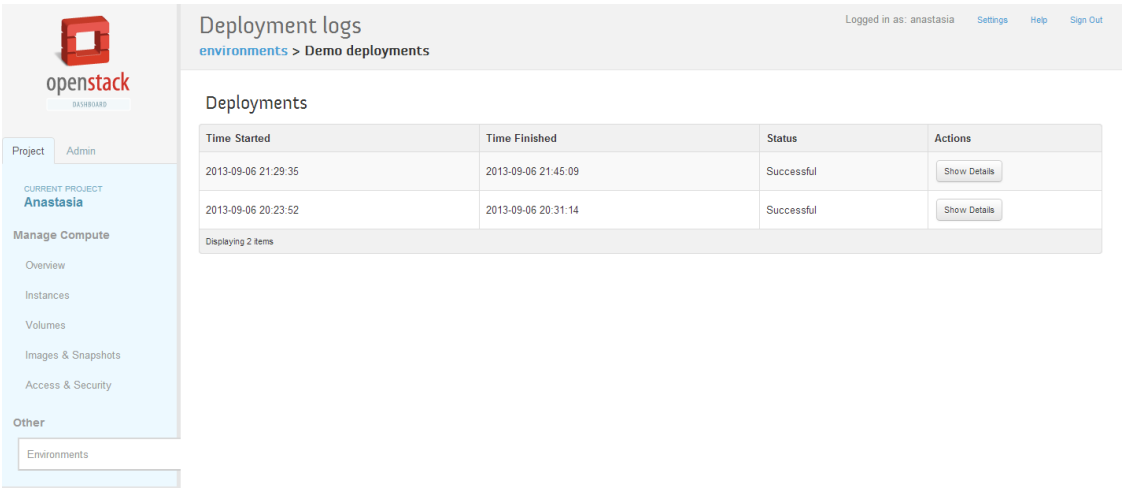


Browsing Murano Deployment

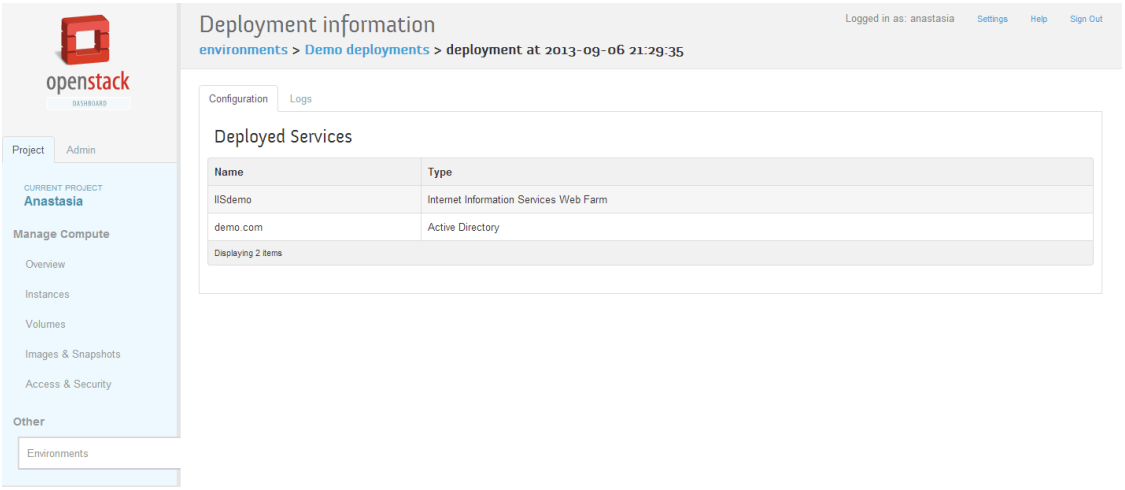
Since Murano Environment can be deployed many times you may want to see the history of its deployments. To do that click the "More-> Show deployments" button on environments index page:



From this page it's easy to see how many times and when Murano Environment was deployed:



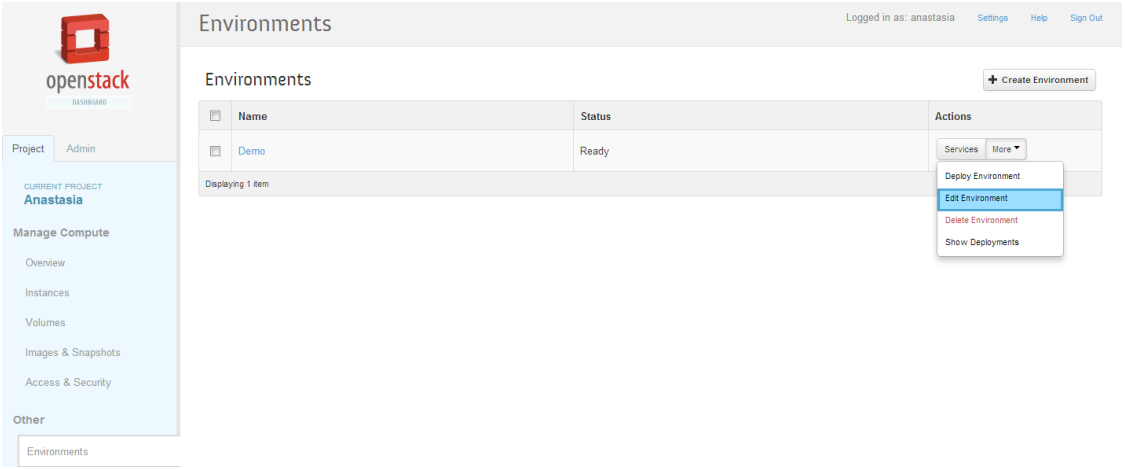
For each deployment you can get a detailed information by clicking the "Show Details" button. You always can go back to any level using navigation string at the page header. From here you can observe what services were installed during deployment:



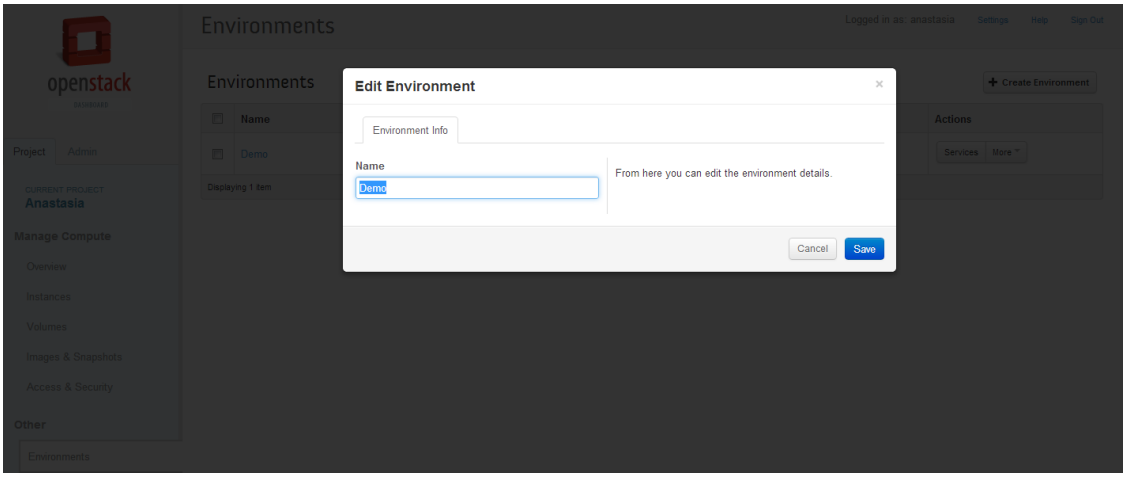
Also deployment logs are available at the "Logs" tab:

Renaming Murano Environment

It's possible to change a name of your environment: just click the "More-> Edit Environment" button on environment index page:



Environment name is not involved in service creation process so you can use spaces and any other characters you want.



Troubleshooting

How to debug OpenStack Heat?

If you can execute Heat command via console interface - all good. It is the most simple way to check Heat state on the node - just execute CLI command 'heat list'. See more information about Heat in openstack wiki page [https://wiki.openstack.org/wiki/Heat/TroubleShooting]

If 'heat list' returns 503 error

It means that OpenStack Heat configuration files contain incorrect credentials. Need to set 'user' = 'heat' and change passwords 'verybadpass' in all configuration files from directory /etc/heat/

If 'heat list' hangs up

Sometimes you can see that 'heat list' hangs up. The root of this problem - connection to the rabbitMQ.

How I can connect to LoadBalancer instance in Server Farms?

First of all you should have KeyPair file 'murano-lb-key'. You can create this file using commands

```
nova keypair-add murano-lb-key > murano-lb-key.priv
chmod 600 murano-lb-key.priv
```

And after that server farms need to be created with this KeyPair. The second step is to 'how to connect to VM with LoadBalancer':

```
ssh -i murano-lb-key.priv root@10.0.0.3
```

Murano dashboard can not connect to Murano API. How I can fix it?

This problem has two ways to fix: Add string

```
MURANO_API_URL='http://localhost:8082'
```

to the /etc/openstack-dashboard/local_settings (or /etc/openstack-dashboard/local_settings.py - it depends on OpenStack configuration) and after that web server restart is needed. Add keystone endpoints for Murano API

```
keystone service-create --name muranoapi --type murano --description "Murano-Api S
keystone endpoint-create --region RegionOne --service-id
--publicurl http://localhost:8082 --internalurl http://localhost:8082 --adminurl h
```

Murano API Service does not work on CentOS 6.x. WebUI can not connect to this service. How to fix this?

The problem in pip lib routes. Need to upgrade this lib and restart Murano API:

```
python-pip install routes --upgrade
initctl stop murano-api
initctl start murano-api
```

Error 'Unexpected state' during the deployment of Web Farms. What the problem?

Sometimes we can see in deployments logs:

```
2013-08-06 09:10:07 - Unable to deploy instance ipkrmhk0vzq4b6 (asp-farm_instance_
2013-08-06 09:10:07 - Unable to create a Server Farm load balancer on unit ipkrmhk
```

The root of this problem is incorrect configuration - Heat can not create Load Balancer instance. Please, remember that you should have admin access for the project in OpenStack to deploy LoadBalancer and also, you should have KeyPair with default name 'murano-lb-key'.

Error in Murano API logs 'No module named helpers.token_sanitizer'

This pip version problem. Need to install pip 1.4 and after that reinstall murano-client, murano-common and murano-api.

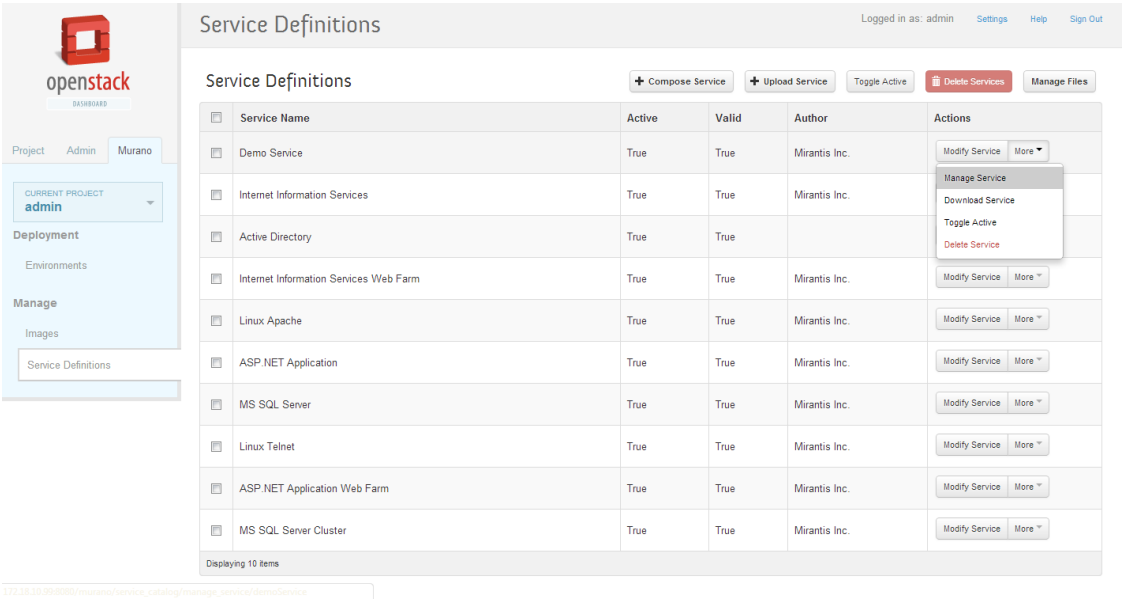
Chapter 4. Metadata Repository

This chapter describes how to operate with Murano Metadata Repository via UI.

Metadata Repository is a REST API server stores all data necessary for a service deployment. Information about service and it's dependencies is called *service definition* or *manifest*. Murano Conductor and Dashboard will load these service definitions and service or application will be available for creation. Besides manifests Murano Repository stores this types of data:

- *ui* - UI definitions for Murano dashboard
- *workflows* - set of rules for Murano Conductor
- *heat* - templates for spawning instance with Heat
- *agent* - Murano Agent templates
- *scripts* - Murano Execution Plans

In "Murano" tab go to the `"Service Definitions"` panel:



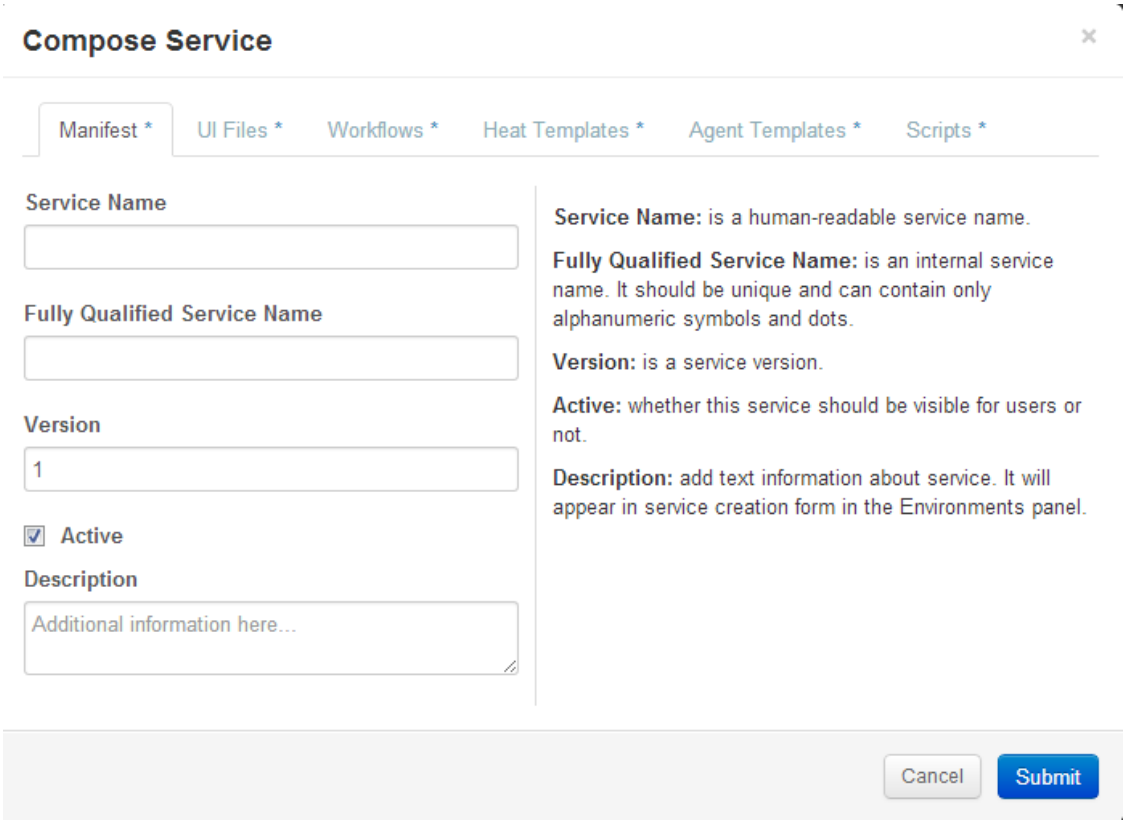
From this view you are able to perform the following actions:

- `Compose Service` provides a form for creating new service manifest. After this operation is complete new service will appear in Service Definitions table and you will be allowed to download this service definition, modify or delete it.
- `Upload Service` provides a form for uploading tar.gz archive with already composed service manifest and other dependency files.
- `Manage Files` button will redirect you to a view where you can manipulate with all files stored in Metadata Repository: download, delete or upload new ones.
- `Toggle Active` button will make service active or inactive depending on the current state.

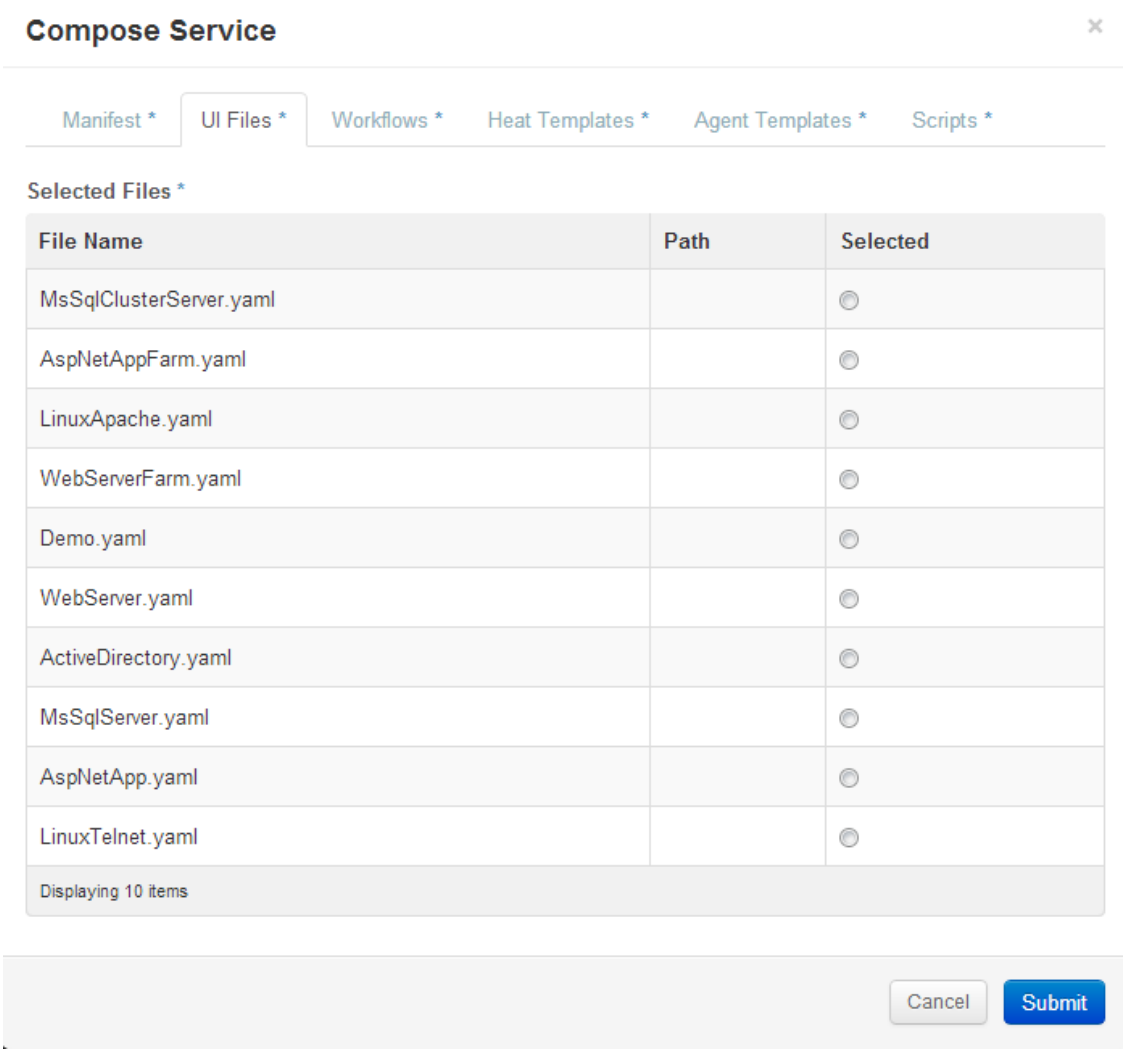
- `Modify Service` gives you opportunity to modify attributes of service definition.
- `Manage Service` redirects you to a view with information about selected service and service files for all data types that Murano Repository supports. From here you can upload file and add it to service dependency in one click.
- `Download Service` saves archive with all files used by specified service.
- `Delete Service` removes service manifest and all dependencies that are not used by other services from Murano Repository server.

Compose Service

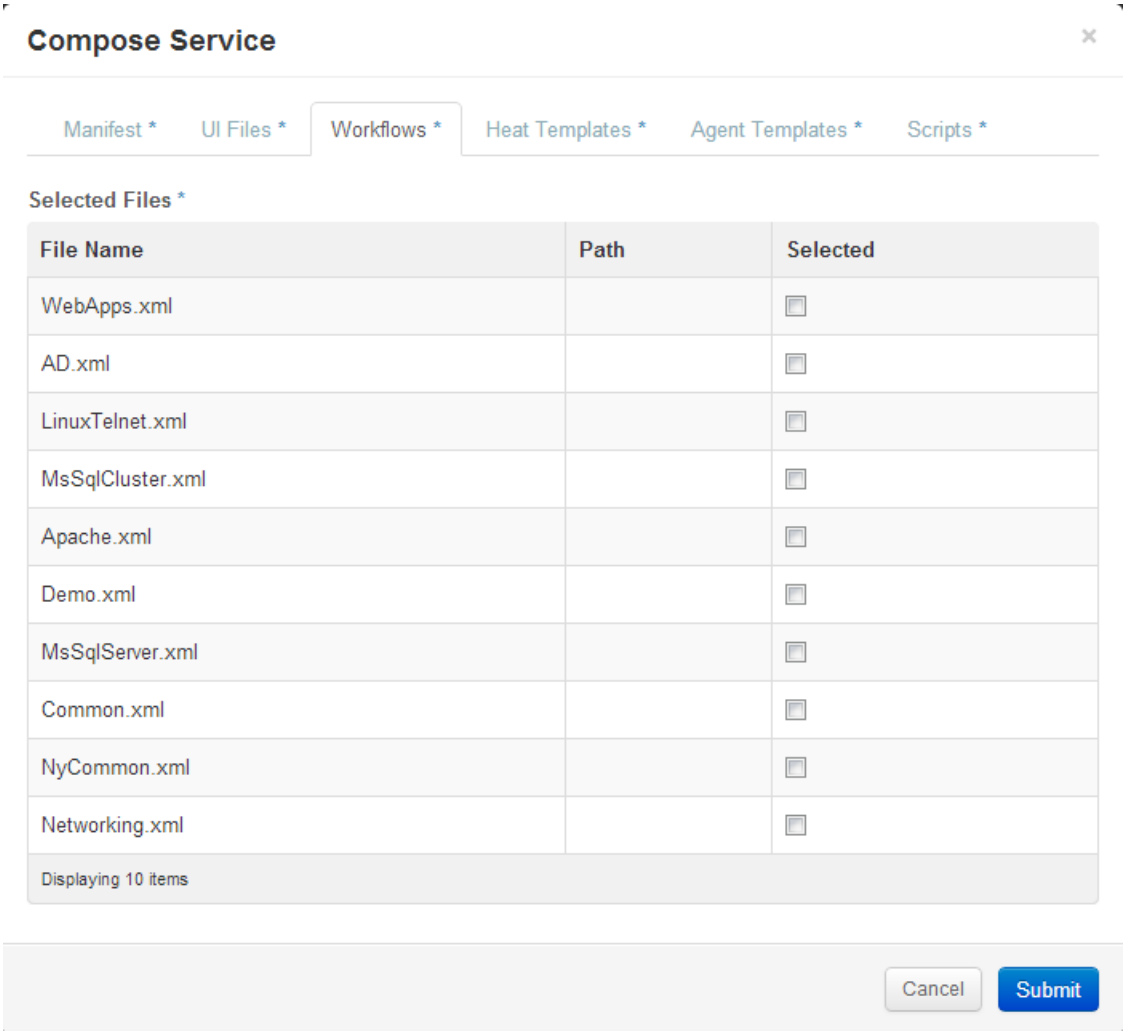
Let's review service creation in details. After clicking on "Compose Service" button from "Service Definitions" view you'll see the following form:



It consists of several tabs. On the first tab you need to fill up general information about service. On the next tab select one of the UI definitions you want to use for this service. You can upload new definition in the *Manage Files* view.

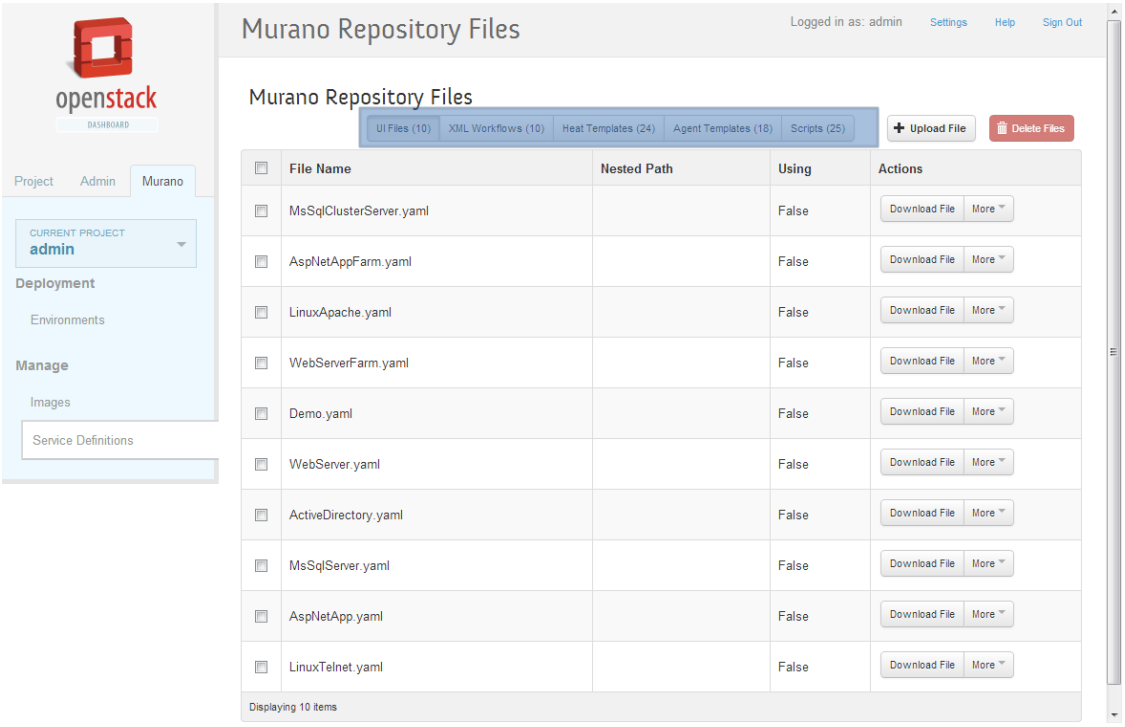


All other tabs correspond to data types. Navigate to desired tab and select files for service dependency. The only mandatory requirement for a new service is a UI definition file - if a service doesn't reference any, it won't be shown in the `"Create Service"` form. All other tabs refer to a files required for a service's deployment phase.



Manage Files

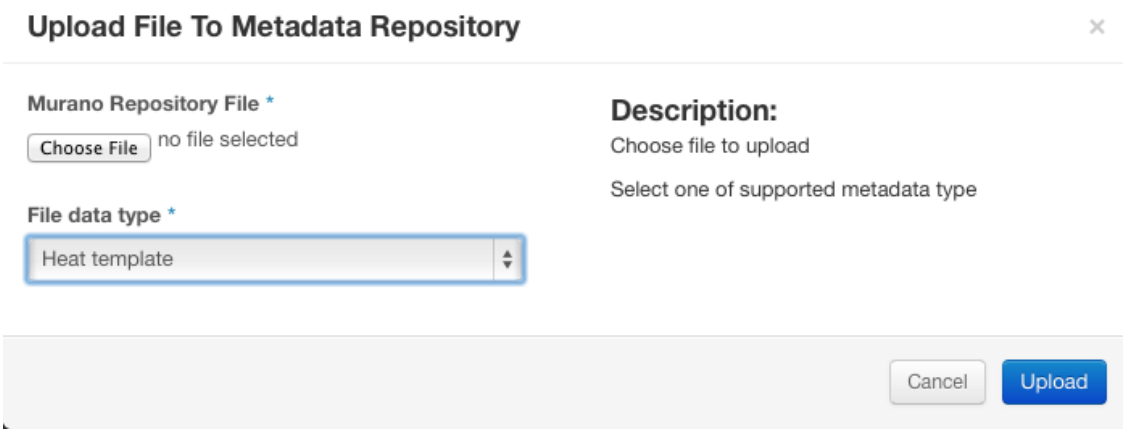
Murano Repository table shows all files stored in Metadata Repository. Files are divided into categories by data types. Button name of each category also contains the number of files in that category. Click on the type name to browse all files of this type.



The following actions are available:

- Download file,
- Delete file,
- Upload file.

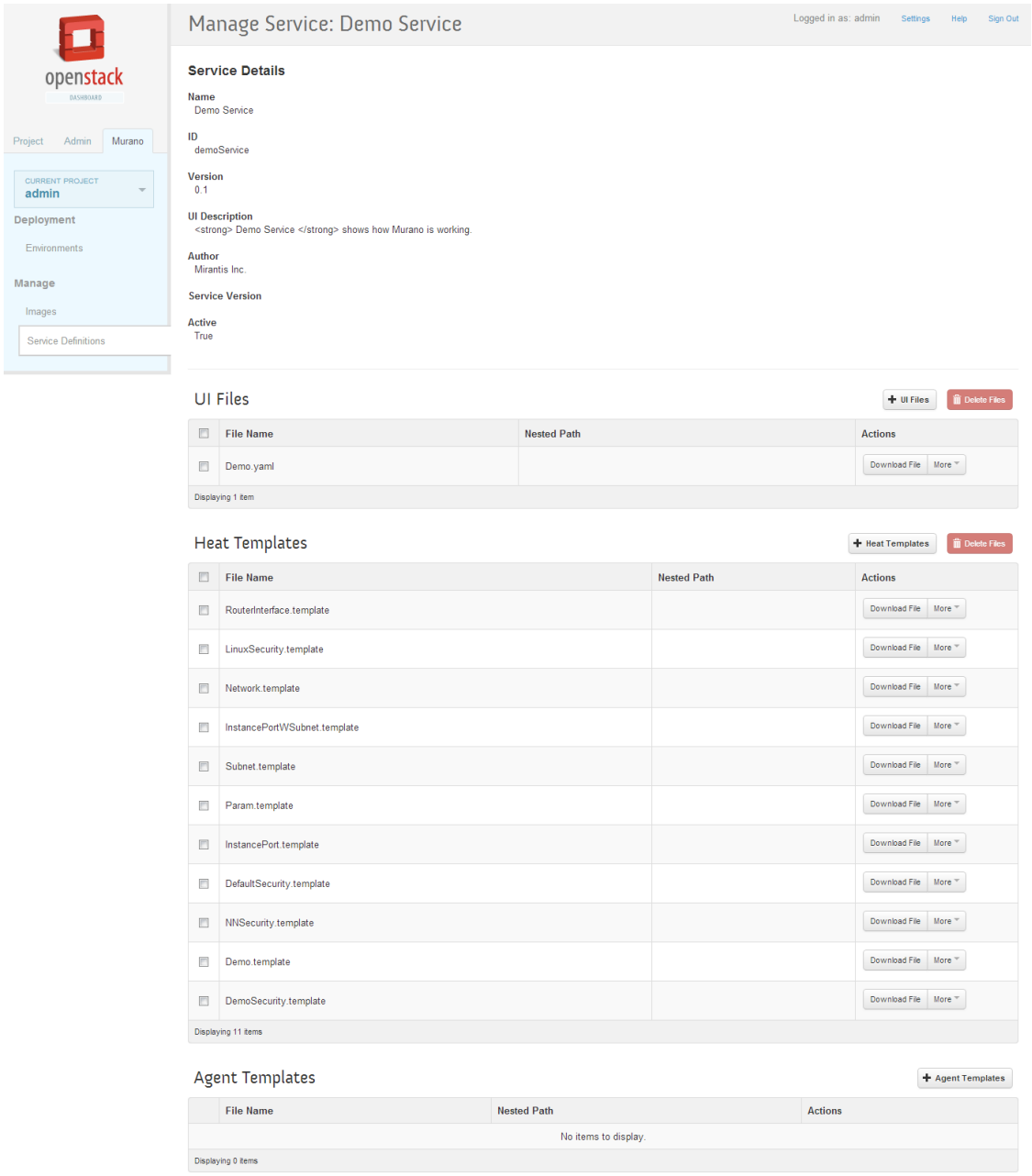
To upload file to repository you need to fill up this form:



Murano Repository File	Select file for uploading. It should be not bigger than 5Mb.
File Type	Select one of supported type. It will be loaded directly to a directory that stores corresponding file types.

Manage Service

Manage Service view displays general information about service and list of file dependency grouped by a separate tables.



In the following view these actions are available:

- View service detail information.
- Download File.
- Delete file.
- Upload file of the exact type by pressing the <File type>" button in the appropriate table.