

Murano User Guide

Murano User Guide

v0.4

Table of Contents

1. How can I use Murano Service?	1
Document change history	1
2. Images	2
3. Environments	4
Creating environment	4
Creating service prototype	5
Active Directory	7
Internet Information Service	9
Internet Information Web Farm Service	10
ASP.NET Service	11
ASP.NET Farm Service	12
MS SQL Service	13
MS SQL Server Failover Cluster	14
Linux Apache Service	17
Linux Telnet Service	18
Deploying environment	18
Working with deployed environment	20
Redeploying Murano Environment	22
Deleting	22
Browsing Murano Deployment	23
Renaming Murano Environment	25
Troubleshooting	26
4. Metadata Repository	28
Compose Service	29
Manage Files	31
Manage Service	33

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) [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!



The screenshot displays the OpenStack Murano Dashboard. The sidebar on the left contains the OpenStack logo and navigation tabs for Project, Admin, and Murano. Under the Murano tab, there are sections for Deployment (Environments) and Manage (Images, Service Definitions). The main panel is titled 'Marked Images' and shows a table with the following data:

Image	Type	Title	Actions
cirros-x86_64-murano	cirros.demo	Demo	Delete Metadata
ws-2012-std	windows.2012	Windows Server 2012	Delete Metadata

At the top right of the 'Marked Images' section, there are buttons for '+ Mark Image' and 'Delete Metadata'. The status 'Logged in as: admin' and links for 'Settings', 'Help', and 'Sign Out' are visible in the top right corner of the dashboard.

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:

Mark Image ×

Image *

cirros-x86_64-murano

▼

Title *

Type *

Windows Server 2012

▼

Windows Server 2012

Murano Demo

Generic Linux

Description:

Mark image with specific Murano metadata. Valid image metadata will be added to the selected image so image can be picked during service creation.

Image: Select one of the images registered in the Glance

Title: Image description which will help you to recognize the image during service creation

Type: Pick image type from types supported by Murano

Cancel

Mark

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: <ul style="list-style-type: none">• 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:



The screenshot shows the OpenStack Horizon dashboard. On the left is a sidebar with the OpenStack logo, a 'DASHBOARD' link, and tabs for 'Project', 'Admin', and 'Murano'. The 'Murano' tab is active. Below the tabs are sections for 'CURRENT PROJECT admin', 'Deployment', 'Environments', 'Manage', 'Images', and 'Service Definitions'. The main content area is titled 'Environments' and shows a table with columns 'Name', 'Status', and 'Actions'. The table is empty, displaying 'No items to display.' and 'Displaying 0 items'. A '+ Create Environment' button is located in the top right corner of the table area. The top right of the dashboard shows 'Logged in as: admin' and links for 'Settings', 'Help', and 'Sign Out'.

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.

The screenshot shows the OpenStack Environments dashboard. On the left is a sidebar with the OpenStack logo, a 'DASHBOARD' label, and navigation tabs for 'Project', 'Admin', and 'Murano'. Below these are sections for 'CURRENT PROJECT admin', 'Deployment' (with an 'Environments' link), and 'Manage' (with links for 'Images' and 'Service Definitions'). The main content area is titled 'Environments' and shows a table with columns 'Name', 'Status', and 'Actions'. The table is empty, displaying 'No items to display.' and 'Displaying 0 items'. A '+ Create Environment' button is in the top right corner. The top right of the page indicates the user is logged in as 'admin' with links for 'Settings', 'Help', and 'Sign Out'.

Just created environment has status *Ready to configure*.

This screenshot shows the OpenStack Environments dashboard after creating a new environment. A green success message at the top right states 'Success: Created environment "Demo".' The table now contains one entry: 'Demo' with a status of 'Ready to configure'. The 'Actions' column for this entry includes 'Services' and 'More' buttons. The sidebar and top navigation remain the same as in the previous screenshot.

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:

Create Service
✕

Service Type

ASP.NET Application
ASP.NET Application
ASP.NET Application Web Farm
Active Directory
Apache Service
Demo Service
Internet Information Services
Internet Information Services Web Farm
Linux Telnet Service
MS SQL Server
MS SQL Server Cluster

Description:

The ASP.NET Application Service installs custom application onto one IIS Web Server

If you want your services to work with AD you should create The Active Directory Service first

Next

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.

Create Service ×

Instance flavor

m1.medium ▾

Instance image

Windows Server 2012 Standard ▾

Availability zone

nova ▾

Internet Information Services Service

Instance Configuration: Specify some instance parameters on which service would be created.

Instance flavor: Select registered in Openstack flavor. Consider that service performance depends on this parameter.

Instance image: Select valid image for a service. Image should already be prepared and registered in glance.

Availability zone: Select availability zone where service would be installed.

Back

Create

Active Directory

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

Create Service

✕

Domain Name

Active Directory Service

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. 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.

Passwords: 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.

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.

Instance Count

Account Name

Administrator password

Confirm password

Recovery password

Confirm password

Hostname template

Back
Next

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

Create Service ×

Service Name

Administrator password

Confirm password

Domain

Hostname template

Internet Information Services Service

Standalone IIS Server

Service Name: Enter a desired name for a service. Just A-Z, a-z, 0-9, dash and underline are allowed.

Passwords: 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.

Domain: Service can be joined to the Active Directory domain. If you want to create an AD domain create the AD Service first.

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

Internet Information Web Farm Service

Create Service



Service Name

Administrator password

Confirm password

Domain

Instance Count

Load Balancer port

Hostname template

Internet Information Services Web Farm Service

A load-balanced array of IIS servers

Service Name: Enter a desired name for a service. Just A-Z, a-z, 0-9, dash and underline are allowed.

Passwords: 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.

Domain: Service can be joined to the Active Directory domain. If you want to create an AD domain create the AD Service first.

Instance Count: Several instances with IIS Service can be created at one time.

Load Balancer port: Specify port number where Load Balancer will be running

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](#)

ASP.NET Service

Create Service



Service Name

Administrator password

Confirm password

Domain

Not in domain



Git repository

Hostname template

Optional

ASP.NET Application Service

ASP.NET application will be installed onto one IISWeb Server

Service Name: Enter a desired name for a service. Just A-Z, a-z, 0-9, dash and underline are allowed.

Passwords: 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.

Domain: Service can be joined to the Active Directory domain. If you want to create an AD domain create the AD Service first.

Git repository: URL of a git repository with the application you want to deploy.

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

ASP.NET Farm Service

Create Service ×

Service Name <input type="text"/>	ASP.NET Application Web Farm Service The ASP.NET application will be installed on a number of IIS Web Servers, and load balancing will be configured. Service Name: Enter a desired name for a service. Just A-Z, a-z, 0-9, dash and underline are allowed. Passwords: 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. Domain: Service can be joined to the Active Directory domain. If you want to create an AD domain create the AD Service first. Git repository: URL of a git repository with the application you want to deploy. Instance Count: Several instances with ASP.NET application can be created at one time. Load Balancer port: Specify port number where Load Balancer will be running 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.
Administrator password <input type="password"/>	
Confirm password <input type="password"/>	
Domain <input type="text" value="Not in domain"/> ▼	
Git repository <input type="text"/>	
Instance Count <input type="text" value="2"/>	
Load Balancer port <input type="text" value="80"/>	
Hostname template <input type="text" value="Optional"/>	

Back Next

MS SQL Service

Create Service ×

Service Name

Administrator password

Confirm password

Domain

Not in domain ▼

☒ Mixed-mode Authentication

SA Password

Confirm password

Hostname template

Optional

MS SQL Server Service

MS SQL Server

Service Name: Enter a desired name for a service. Just A-Z, a-z, 0-9, dash and underline are allowed.

Passwords: 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.

Domain: Service can be joined to the Active Directory domain. If you want to create an AD domain create the AD Service 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.

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

MS SQL Server Failover Cluster

Create Service

Service Name

Administrator password

Confirm password

☐ Active Directory is configured by the System Administrator

Domain

demo.com

☒ Mixed-mode Authentication

SA Password

Confirm password

MS SQL Server Cluster Service

MS SQL Failover Cluster

Service Name: Enter a desired name for a service. Just A-Z, a-z, 0-9, dash and underline are allowed.

Passwords: 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.

Domain: Service can be joined to the Active Directory domain. If you want to create an AD domain create the AD Service 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.

Back

Next

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 is configured by the System Administrator**

Active Directory User

Active Directory Password

Confirm password

Domain

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

Hostname template

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

Create Service

Nodes

Node	Sync	Primary
node1	<input checked="" type="checkbox"/>	<input checked="" type="radio"/>
node2	<input checked="" type="checkbox"/>	<input type="radio"/>

Add node
Remove node

Database list

MS SQL Server Cluster Service

Nodes: Configure cluster instances. Cluster node quantity can be set with 'Add' and 'Remove' buttons. Configure Sync mode by enabling corresponding checkbox. All other nodes will be in Async mode. Just 2 nodes are allowed to be Sync. Also one Master node need to be selected. SQL Failover cluster has limit of 5 instances.

Database list: Specify names for new databases which will be created as part of service installation. Here should come comma-separated list of database names, where each name has the following syntax: first symbol should be latin letter or underscore; subsequent symbols can be latin letter, numeric, underscore, at sign, number sign or dollar sign.

Back
Next

Linux Apache Service

Create Service

Service Name

☐ PHP module

Instance Count

1

Key Pair

murano-lb-key
▼

Hostname

Optional

Apache Service Service

Service Name: To identify your service in logs please specify a service name

PHP module: Add php support into Apache web server.

Instance Count: Several instances with Apache web Service can be created at one time.

Key Pair: The Key Pair for VMs with this service

Hostname: For your convenience instance hostname can be specified. Enter a name or leave blank for random name generation.

Back
Next

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

Create Service

Service Name

Key Pair

murano-lb-key

Hostname

Linux Telnet Service

Telnet service that can be installed at linux

Service Name: Enter a desired name for a service. Just A-Z, a-z, 0-9, dash and underline are allowed.

Key Pair: The Key Pair for VMs with this service

Hostname: For your convenience instance hostname can be specified. Enter a name or leave blank for random name generation.

Back

Next

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.

openstack
DASHBOARD

Project

CURRENT PROJECT
Anastasia

Manage Compute

- Overview
- Instances
- Volumes
- Images & Snapshots
- Access & Security

Other

- Environments

Environment: Demo

environments > environment Demo

Logged in as: anastasia [Settings](#) [Help](#) [Sign Out](#)

Services + Create Service Delete Services Deploy This Environment

<input type="checkbox"/>	Name	Type	Status	Last operation	Time updated	Actions
<input type="checkbox"/>	demo.com	Active Directory	Configuring	Service draft created	2013-09-05 01:01:30	Delete Service
<input type="checkbox"/>	ISDemo	Internet Information Services Web Farm	Configuring	Service draft created	2013-09-05 01:02:19	Delete Service

Displaying 2 items

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.

Environments

Environment: Demo
environments > environment Demo

Success: Deploy started

Services

Name	Type	Status	Last operation	Time updated	Actions
demo.com	Active Directory	Deploy in progress	Service draft created	2013-09-05 01:01:30	
IISdemo	Internet Information Services Web Farm	Deploy in progress	Service draft created	2013-09-05 01:02:19	

Displaying 2 items

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.

Environment: Demo
environments > environment Demo

Create Service Delete Services Deploy This Environment

Services

Name	Type	Status	Last operation	Time updated	Actions
demo.com	Active Directory	Ready	Primary Domain Controller created	2013-09-05 21:44:01	Delete Service
IISdemo	Internet Information Services Web Farm	Ready	Unit rnhfthi0713 (IISdemo_instance_0) has joined domain demo.com	2013-09-05 21:47:24	Delete Service

Displaying 2 items

And now you can see installation progress.

Service Detail: demo.com
environments > environment Demo > service demo.com

Service Logs

- 2013-09-05 21:25:41 - Creating instance ad (demo.com_instance_0)
- 2013-09-05 21:28:16 - Instance ad (demo.com_instance_0) created
- 2013-09-05 21:28:19 - Creating Primary Domain Controller on unit ad (demo.com_instance_0)
- 2013-09-05 21:44:01 - Primary Domain Controller created

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.



The screenshot shows the OpenStack dashboard with the 'Environments' section active. The left sidebar contains the OpenStack logo and navigation links for 'Project' (Anastasia) and 'Admin'. The main content area displays a table of environments. The 'Demo' environment is highlighted in yellow and has a status of 'Deploy in progress'. The 'Virtual DC' environment has a status of 'Ready to configure'. Both environments have 'Services' and 'More' buttons in the 'Actions' column. A '+ Create Environment' button is located at the top right of the table.

<input type="checkbox"/>	Name	Status	Actions
<input checked="" type="checkbox"/>	Demo	Deploy in progress	Services More ▾
<input type="checkbox"/>	Virtual DC	Ready to configure	Services More ▾

Displaying 2 items

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



The screenshot shows the OpenStack dashboard with the 'Environments' section active. The left sidebar is the same as the previous screenshot. The main content area displays a table of environments. The 'Demo' environment is now in a 'Ready' status. The 'Virtual DC' environment is no longer visible. The 'Services' and 'More' buttons are still present in the 'Actions' column. A '+ Create Environment' button is located at the top right of the table.

<input type="checkbox"/>	Name	Status	Actions
<input type="checkbox"/>	Demo	Ready	Services More ▾

Displaying 1 item

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 want to know about.

The screenshot shows the OpenStack dashboard interface. On the left is a sidebar with the OpenStack logo and navigation links for Project, Admin, and Manage Compute. The main content area is titled 'Service Detail: IISdemo' and shows various details about the service, including its name, ID, type, status, domain, and load balancer URI. It also lists two service units with their respective hostnames and service instance names.

Service Detail: IISdemo
environments > environment Demo > service IISdemo

Service Info

Name
IISdemo

ID
83e46d472d3d49199763bec4b4effe44

Type
Internet Information Services Web Farm

Status
Ready

Domain
demo.com

Load Balancer URI
80

Service unit1

Hostname
mhjfh8h0713

Service instance name
[ef489cd02b66d4abd8b251fedcd350f.mhjfh8h0713](#)

Service unit2

Hostname
llchhi8h0774

Service instance name
[ef489cd02b66d4abd8b251fedcd350f.llchhi8h0774](#)

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.

Service unit1

Hostname
IIS1

Service instance name
[e98a933da40ee4dc096b7f7d7be4fa3de.IIS1](#)

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:

Environments

Environments

Logged in as: anastasia [Settings](#) [Help](#) [Sign Out](#)

Environments [+ Create Environment](#)

Name	Status	Actions
Demo	Ready	Services More Deploy Environment Edit Environment Delete Environment Show Deployments

Displaying 1 item

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

Deployment logs

environments > Demo deployments

Logged in as: anastasia [Settings](#) [Help](#) [Sign Out](#)

Deployments

Time Started	Time Finished	Status	Actions
2013-09-06 21:29:35	2013-09-06 21:45:09	Successful	Show Details
2013-09-06 20:23:52	2013-09-06 20:31:14	Successful	Show Details

Displaying 2 items

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:

Deployment information

environments > Demo deployments > deployment at 2013-09-06 21:29:35

Logged in as: anastasia [Settings](#) [Help](#) [Sign Out](#)

Configuration Logs

Deployed Services

Name	Type
IISdemo	Internet Information Services Web Farm
demo.com	Active Directory

Displaying 2 items

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

The screenshot shows the OpenStack dashboard with the 'Deployment information' page selected. The page title is 'Deployment information' and the breadcrumb is 'environments > Demo deployments > deployment at 2013-09-06 21:29:35'. The left sidebar shows the 'Project' menu with 'Anastasia' as the current project. The 'Manage Compute' section is expanded, showing 'Overview', 'Instances', 'Volumes', 'Images & Snapshots', 'Access & Security', and 'Other'. The 'Other' section is further expanded, showing 'Environments'. The main content area has tabs for 'Configuration' and 'Logs'. The 'Logs' tab is active, displaying a list of deployment events:

```
2013-09-06 21:29:35 - Deployment scheduled
2013-09-06 21:29:39 - Creating instance IIS1 (IISdemo_instance_0)
2013-09-06 21:29:39 - Creating instance IIS2 (IISdemo_instance_1)
2013-09-06 21:29:39 - Creating instance ad (demo.com_instance_0)
2013-09-06 21:29:54 - Instance IIS1 (IISdemo_instance_0) created
2013-09-06 21:29:55 - Instance IIS2 (IISdemo_instance_1) created
2013-09-06 21:29:55 - Instance ad (demo.com_instance_0) created
2013-09-06 21:29:56 - Creating Primary Domain Controller on unit ad (demo.com_instance_0)
2013-09-06 21:45:03 - Primary Domain Controller created
2013-09-06 21:45:09 - Deployment finished
```

Renaming Murano Environment

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

The screenshot shows the 'Environments' index page. The table lists one environment named 'Demo' with a status of 'Ready'. The 'Actions' column for the 'Demo' environment has a 'More' dropdown menu. The 'More' menu is open, showing options: 'Deploy Environment', 'Edit Environment' (highlighted), 'Delete Environment', and 'Show Deployments'. The left sidebar is the same as in the previous screenshot.

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

The screenshot shows the 'Edit Environment' dialog box. The dialog has a title bar 'Edit Environment' and a close button 'X'. It contains a tab 'Environment Info' and a text input field for 'Name' with the value 'Demo'. To the right of the input field is the text 'From here you can edit the environment details.' At the bottom right of the dialog are 'Cancel' and 'Save' buttons. The background shows the 'Environments' index page with the 'Demo' environment listed.

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 its 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:

Service Definitions

Logged in as: admin Settings Help Sign Out

Service Definitions

+ Compose Service + Upload Service Toggle Active Delete Services Manage Files

Service Name	Active	Valid	Author	Actions
Demo Service	True	True	Mirantis Inc.	Modify Service More ▾
Internet Information Services	True	True	Mirantis Inc.	Manage Service Download Service Toggle Active Delete Service
Active Directory	True	True		
Internet Information Services Web Farm	True	True	Mirantis Inc.	Modify Service More ▾
Linux Apache	True	True	Mirantis Inc.	Modify Service More ▾
ASP.NET Application	True	True	Mirantis Inc.	Modify Service More ▾
MS SQL Server	True	True	Mirantis Inc.	Modify Service More ▾
Linux Telnet	True	True	Mirantis Inc.	Modify Service More ▾
ASP.NET Application Web Farm	True	True	Mirantis Inc.	Modify Service More ▾
MS SQL Server Cluster	True	True	Mirantis Inc.	Modify Service More ▾

Displaying 10 items

172.18.18.90:8000/murano/service_catalog/manage_service/demoService

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:

Compose Service ✕

Manifest * UI Files * Workflows * Heat Templates * Agent Templates * Scripts *

Service Name

Fully Qualified Service Name

Version

1

☒ **Active**

Description

Additional information here...

Service Name: is a human-readable service name.

Fully Qualified Service Name: is an internal service name. It should be unique and can contain only alphanumeric symbols and dots.

Version: is a service version.

Active: whether this service should be visible for users or not.

Description: add text information about service. It will appear in service creation form in the Environments panel.

Cancel Submit

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.

Compose Service ×

Manifest *

UI Files *

Workflows *

Heat Templates *

Agent Templates *

Scripts *

Selected Files *

File Name	Path	Selected
MsSqlClusterServer.yaml		<input type="radio"/>
AspNetAppFarm.yaml		<input type="radio"/>
LinuxApache.yaml		<input type="radio"/>
WebServerFarm.yaml		<input type="radio"/>
Demo.yaml		<input type="radio"/>
WebServer.yaml		<input type="radio"/>
ActiveDirectory.yaml		<input type="radio"/>
MsSqlServer.yaml		<input type="radio"/>
AspNetApp.yaml		<input type="radio"/>
LinuxTelnet.yaml		<input type="radio"/>
Displaying 10 items		

Cancel

Submit

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.

Compose Service ×

Manifest * UI Files * **Workflows *** Heat Templates * Agent Templates * Scripts *

Selected Files *

File Name	Path	Selected
WebApps.xml		<input type="checkbox"/>
AD.xml		<input type="checkbox"/>
LinuxTelnet.xml		<input type="checkbox"/>
MsSqlCluster.xml		<input type="checkbox"/>
Apache.xml		<input type="checkbox"/>
Demo.xml		<input type="checkbox"/>
MsSqlServer.xml		<input type="checkbox"/>
Common.xml		<input type="checkbox"/>
NyCommon.xml		<input type="checkbox"/>
Networking.xml		<input type="checkbox"/>
Displaying 10 items		

Cancel Submit

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.

Murano Repository Files

Logged in as: admin Settings Help Sign Out

Murano Repository Files

UI Files (10) XML Workflows (10) Heat Templates (24) Agent Templates (18) Scripts (25) + Upload File Delete Files

File Name	Nested Path	Using	Actions
MsSqlClusterServer.yaml		False	Download File More
AspNetAppFarm.yaml		False	Download File More
LinuxApache.yaml		False	Download File More
WebServerFarm.yaml		False	Download File More
Demo.yaml		False	Download File More
WebServer.yaml		False	Download File More
ActiveDirectory.yaml		False	Download File More
MsSqlServer.yaml		False	Download File More
AspNetApp.yaml		False	Download File More
LinuxTelnet.yaml		False	Download File More

Displaying 10 items

The following actions are available:

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

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

Upload File To Metadata Repository

Murano Repository File *

Choose File no file selected

Description:

Choose file to upload

Select one of supported metadata type

File data type *

Heat template

Cancel Upload

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.



[Project](#)
[Admin](#)
[Murano](#)

CURRENT PROJECT

admin

Deployment

Environments

Manage

Images

Service Definitions

Manage Service: Demo Service

[Logged in as: admin](#)
[Settings](#)
[Help](#)
[Sign Out](#)

Service Details

Name

Demo Service

ID

demoService

Version

0.1

UI Description

 Demo Service shows how Murano is working.

Author

Mirantis Inc.

Service Version

Active

True

UI Files

[+ UI Files](#)
[Delete Files](#)

File Name	Nested Path	Actions
Demo.yaml		Download File More ^

Displaying 1 item

Heat Templates

[+ Heat Templates](#)
[Delete Files](#)

File Name	Nested Path	Actions
RouterInterface.template		Download File More ^
LinuxSecurity.template		Download File More ^
Network.template		Download File More ^
InstancePortVSubnet.template		Download File More ^
Subnet.template		Download File More ^
Param.template		Download File More ^
InstancePort.template		Download File More ^
DefaultSecurity.template		Download File More ^
NNSecurity.template		Download File More ^
Demo.template		Download File More ^
DemoSecurity.template		Download File More ^

Displaying 11 items

Agent Templates

[+ Agent Templates](#)

File Name	Nested Path	Actions
No items to display.		

Displaying 0 items

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.