AIM : Case Study on open source cloud solutions.

THEORY :

1. List Open source cloud solutions.

1. Openstack
2. nextcloud
3. Cloud Foundary
4. WSO2 stratos
5. Cloudify
6. OpenShift
7. Tsuru
8. Stackato
9. Alibaba

2. Prepare detailed study on anyone

OPENSTACK

a. Software and Hardware requirements

Software Requirements

Supported operating systems:

- Ubuntu Server 14.04 LTS
- RHEL/CentOS
- Debian

System packages for Ubuntu:

- gcc
- python-pip
- python-dev
- libxml2-dev
- libxslt-dev
- libffi-dev
- libpq-dev
- python-openssl
- mysql-client

System packages for CentOS:

- gcc
- python-pip
- python-devel
- libxml2-devel
- libxslt-devel
- libffi-devel
- postgresql-devel
- pyOpenSSL
- mysql

Hardware Requirements

Criteria	**Minimal**	**Recommended**
CPU	4 core @ 2.4 GHz	24 core @ 2.67 GHz
RAM	8 GB	24 GB or more
HDD	2 x 500 GB (7200 rpm)	4 x 500 GB (7200 rpm)
RAID	Software RAID-1 (use mdadm as it improves the read performance almost twice)	Hardware RAID-10

Other possible storage configurations:

- 1x SSD 500+ GB
- 1x HDD (7200 rpm) 500+ GB and 1x SSD 250+ GB (install the system onto the HDD and mount the SSD drive to the directory where the virtual machines images are stored)
- 1x HDD (15000 rpm) 500+ GB

b. How to install and Configure

Install the packages:

 # apt-get install openstack-dashboard

Edit **the /etc/openstack-dashboard/local_settings.py** file and complete the following actions:

- Configure the dashboard to use OpenStack services on the controller node:

OPENSTACK_HOST = "controller"

- Allow all hosts to access the dashboard:

  ALLOWED_HOSTS = ['*',]

- Configure the memcached session storage service:

  SESSION_ENGINE = 'django.contrib.sessions.backends.cache'

  CACHES = {

  'default': {

  'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',

  'LOCATION': 'controller:11211',

  }

  }

  Note : Comment out any other session storage configuration.

- Enable the Identity API version 3:

  OPENSTACK_KEYSTONE_URL = "http://%s:5000/v3" % OPENSTACK_HOST

- Enable support for domains:

  OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True

- Configure API versions:

  OPENSTACK_API_VERSIONS = {

  "identity": 3,

  "image": 2,

  "volume": 2,

  }

- Configure default as the default domain for users that you create via the dashboard:

  OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "default"

- Configure user as the default role for users that you create via the dashboard:

  OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"

- If you chose networking option 1, disable support for layer-3 networking services:

  OPENSTACK_NEUTRON_NETWORK = {

  ...

  'enable_router': False,

  'enable_quotas': False,

  'enable_distributed_router': False,

  'enable_ha_router': False,

  'enable_lb': False,

  'enable_firewall': False,

  'enable_vpn': False,

  'enable_fip_topology_check': False,

  }

- Optionally, configure the time zone:

  TIME_ZONE = "TIME_ZONE"

  Replace TIME_ZONE with an appropriate time zone identifier.

Finalize installation

- Reload the web server configuration:

  # service apache2 reload

c. How to deploy an application

To deploy an application we need to import application packages into murano and then add applications to an environment and deploy it.

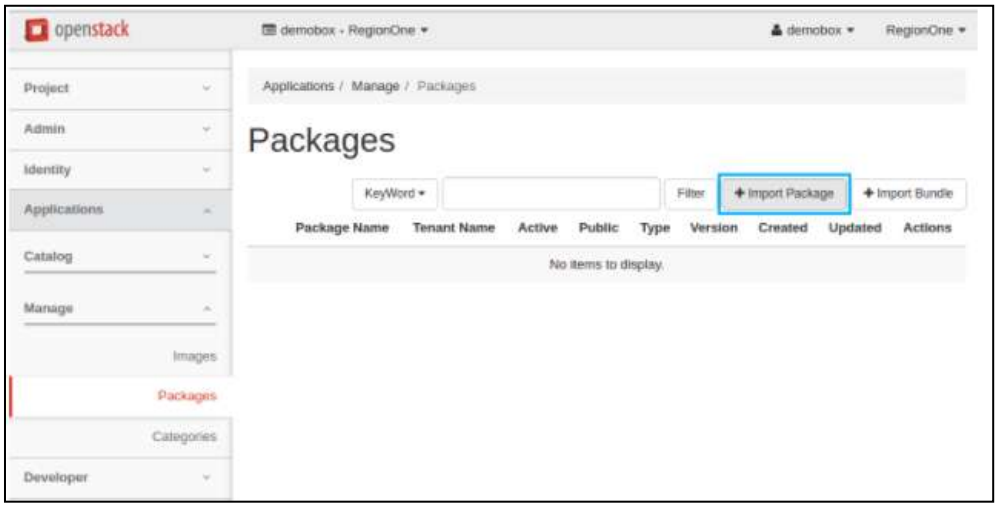
Import an application package

There are several ways of importing an application package into murano:

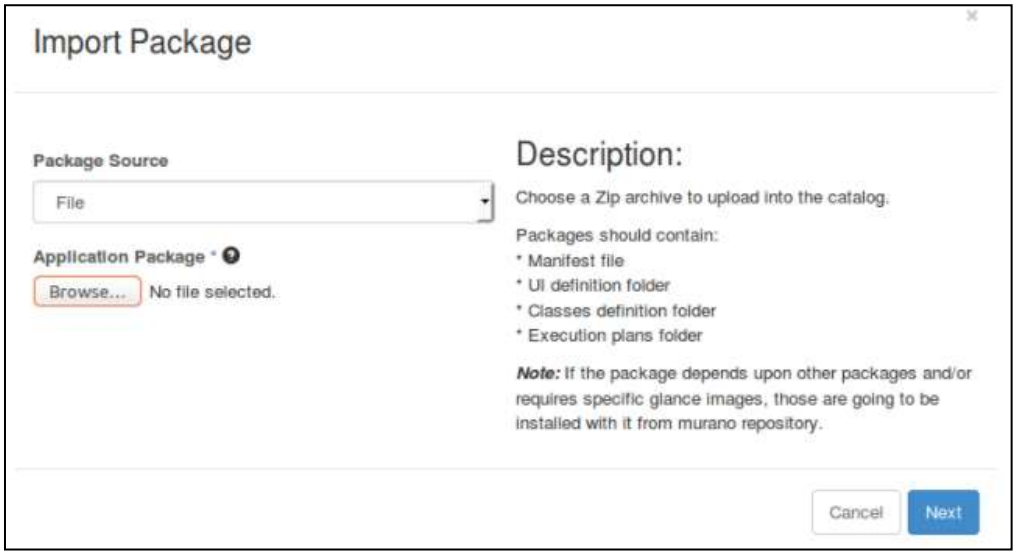
- from a zip file
- from murano applications repository
- from bundles of applications

1. From a zip file

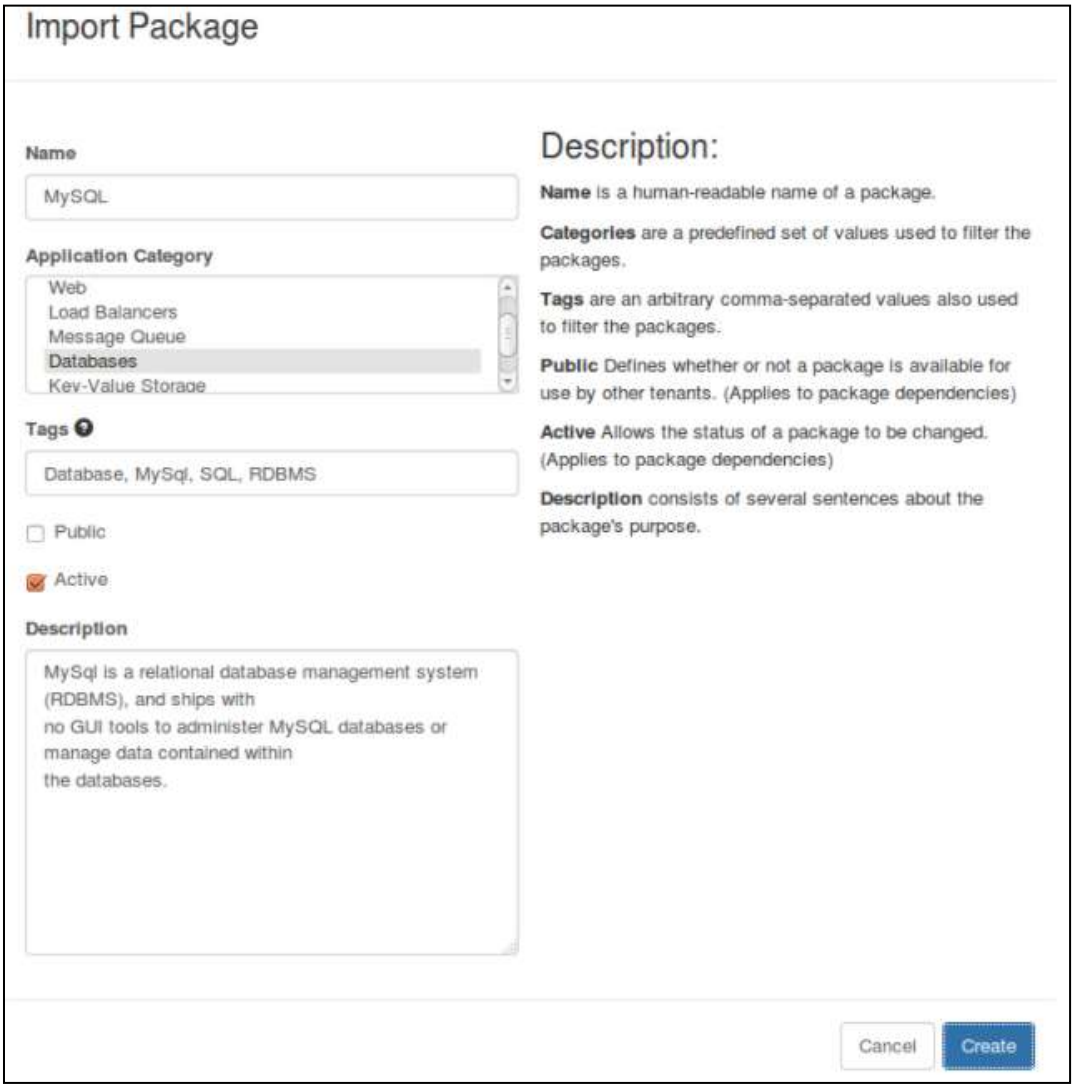
In OpenStack dashboard, navigate to Applications > Manage > Packages.Click the Import Package button on the top right of the page.



From the Package source drop-down list choose File, then click Browse to select a .zip file you want to import, and then click Next.



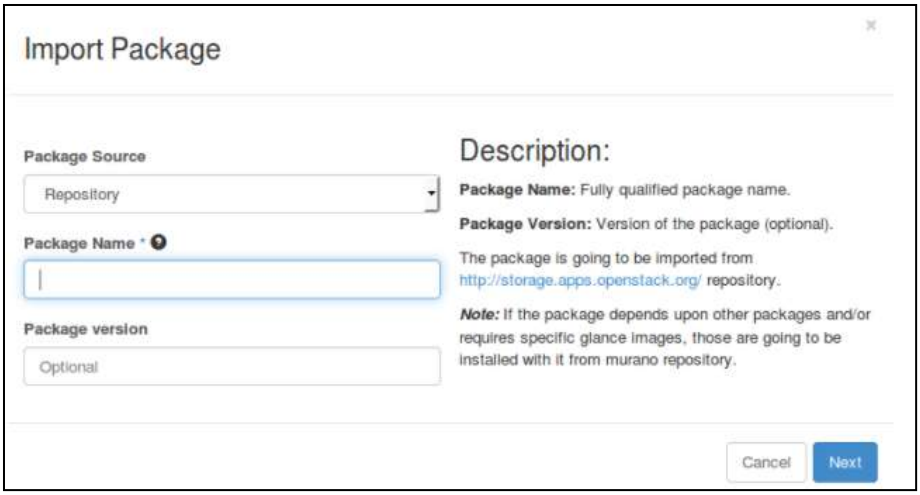
At this step, the package is already uploaded. Choose a category from the Application Category menu. You can select multiple categories while holding down the Ctrl key. If necessary, verify and update the information about the package, then click the Create button.



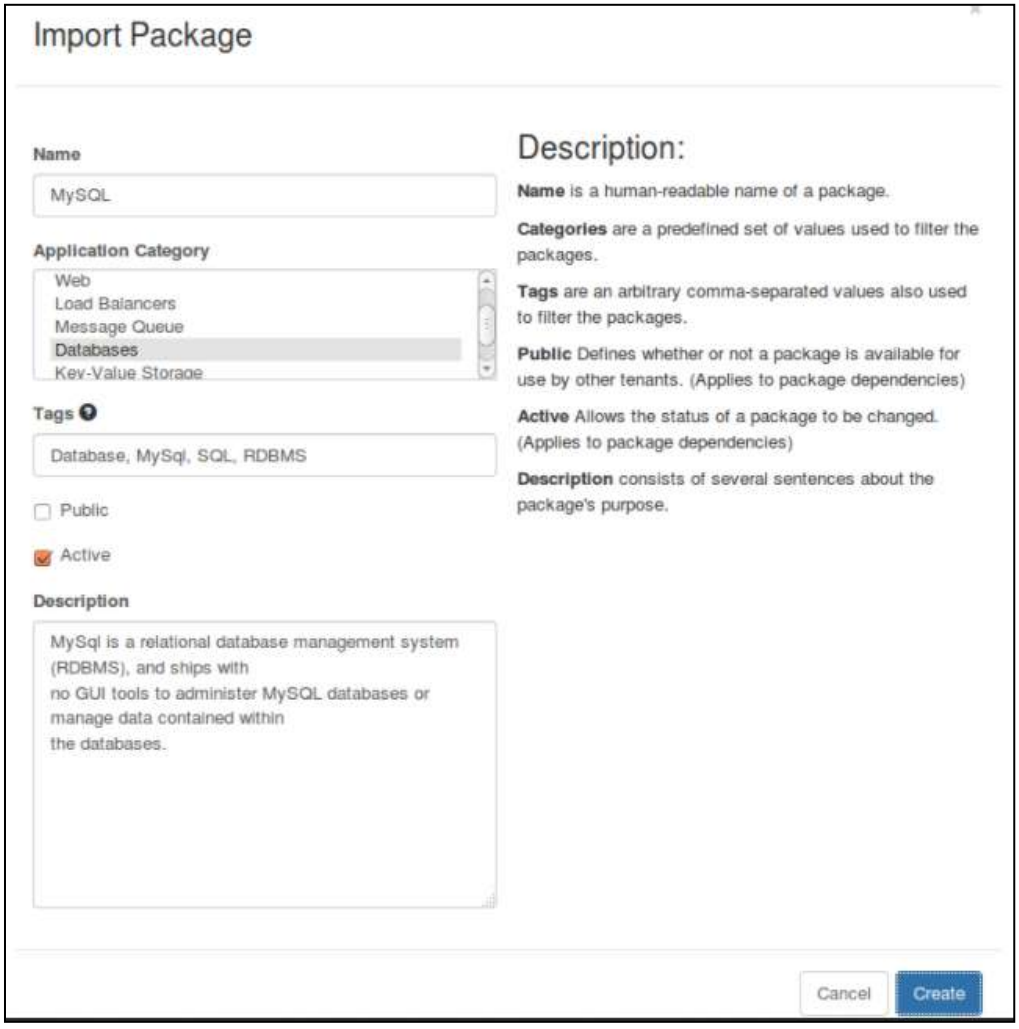
2. From a repository

In OpenStack dashboard, navigate to Applications > Manage > Packages.Click the Import Package button on the top right of the page.

From the Package source drop-down list, choose Repository, enter the package name, and then click Next. Note that you may also specify the version of the package.

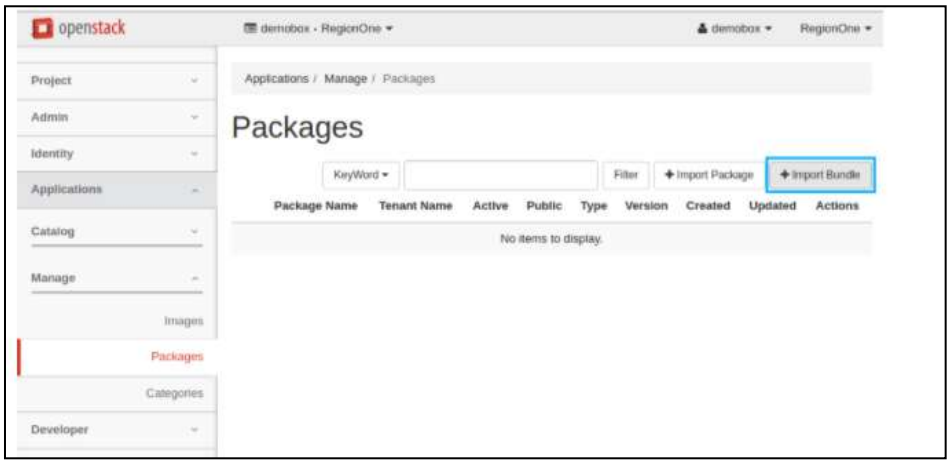


At this step, the package is already uploaded. Choose a category from the Application Category menu. You can select multiple categories while holding down the Ctrl key. If necessary, verify and update the information about the package, then click the Create button.

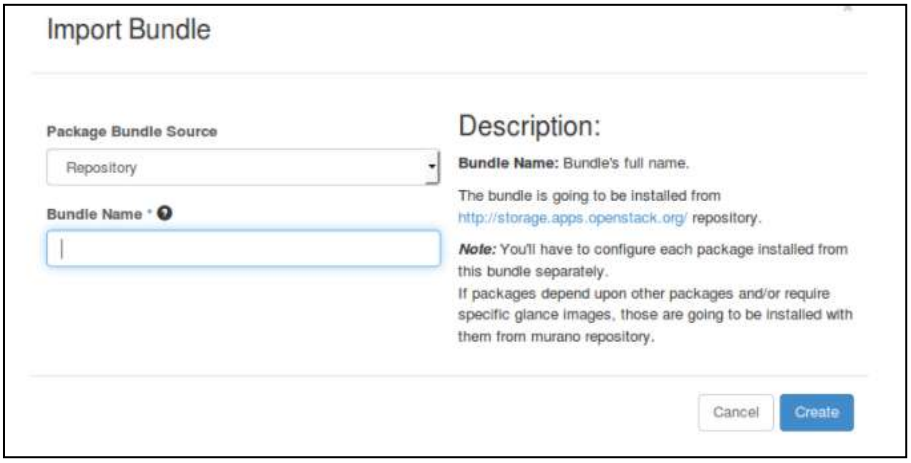


3. From a bundle of applications

In OpenStack dashboard, navigate to Applications > Manage > Packages. Click the Import Bundle button on the top right of the page.



from the Package Bundle Source drop-down list, choose Repository, enter the bundle name, and then click Create.

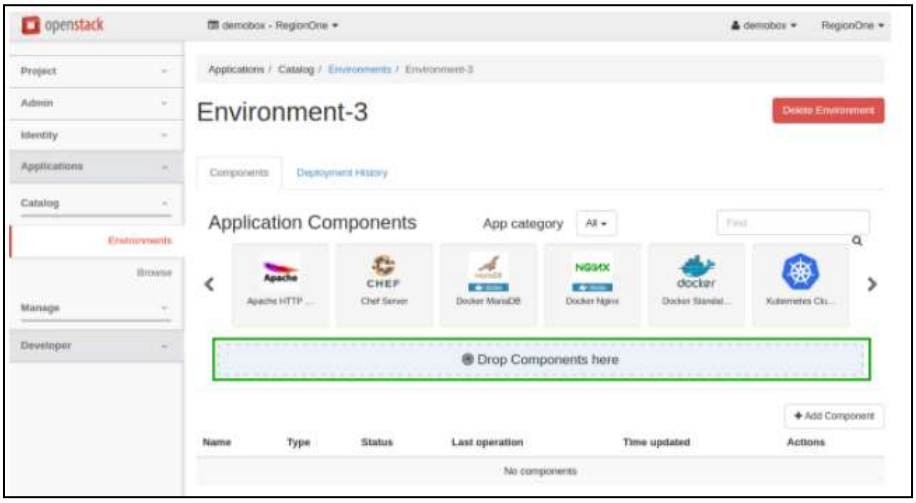


Add an application to environment

After uploading an application, the second step is to add it to an environment. You can do this:

- from environment details page
- from applications catalog page

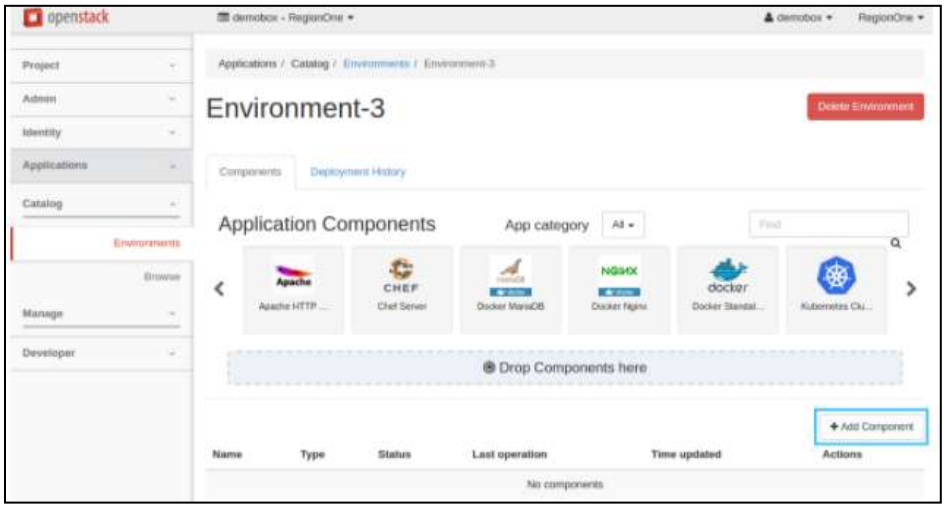
1. From environment details page
In OpenStack dashboard, navigate to Applications > Catalog > Environments.
Find the environment you want to manage and click Manage Components, or simply click on the environment's name.
Procced with the Drop Components here field or the Add Component button.
Use of Drop Components here field
On the Environment Components page, drag and drop a desired application into the Drop Components here field under the Application Components section.



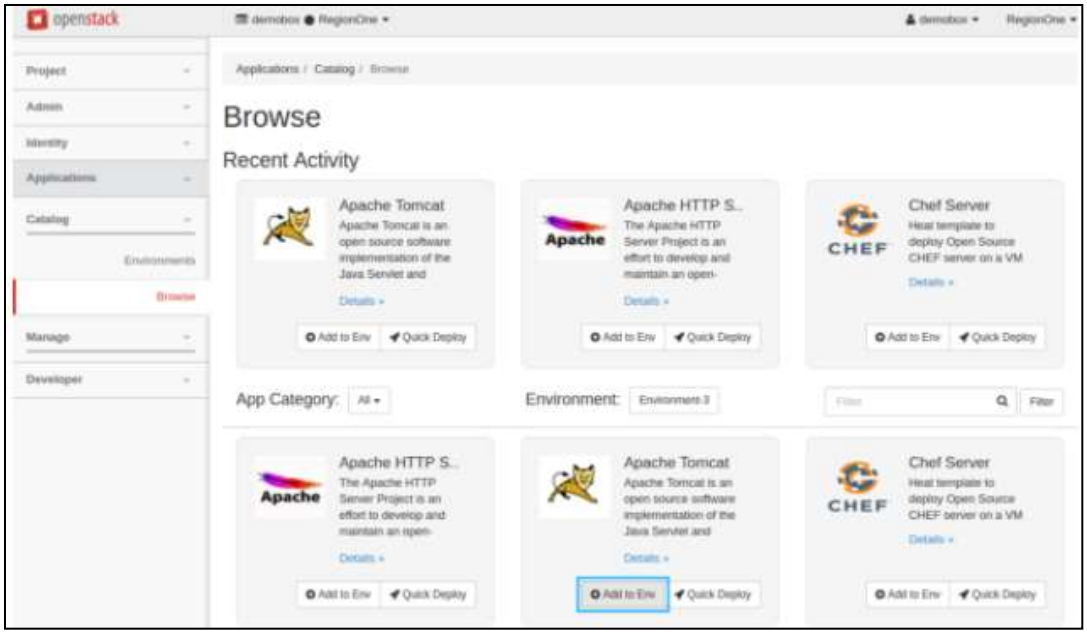
Configure the application. Note that the settings may vary from app to app and are predefined by the application author. When done, click Next, then click Create. Now the application appears in the Component List section on the Environment Components page.

Use of Add Component button

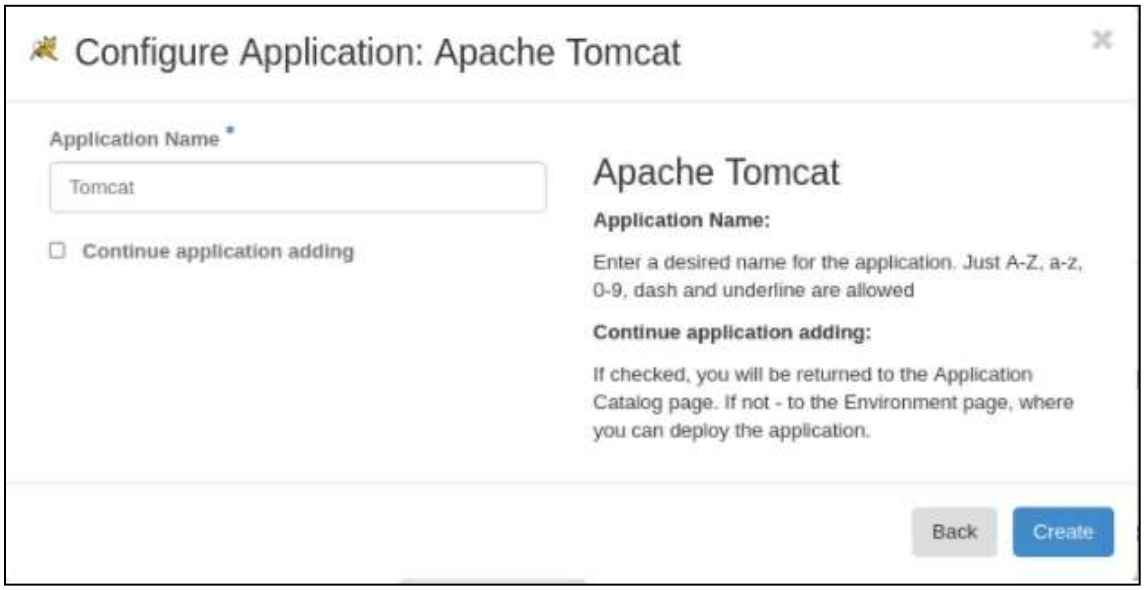
On the Environment Components page, click Add Component.



Find the application you want to add and click Add to Env.



Configure the application and click Next. Note that the settings may vary from app to app and are predefined by the application author.To add more applications, check Continue application adding, then click Create and repeat the steps above. Otherwise, just click Create.



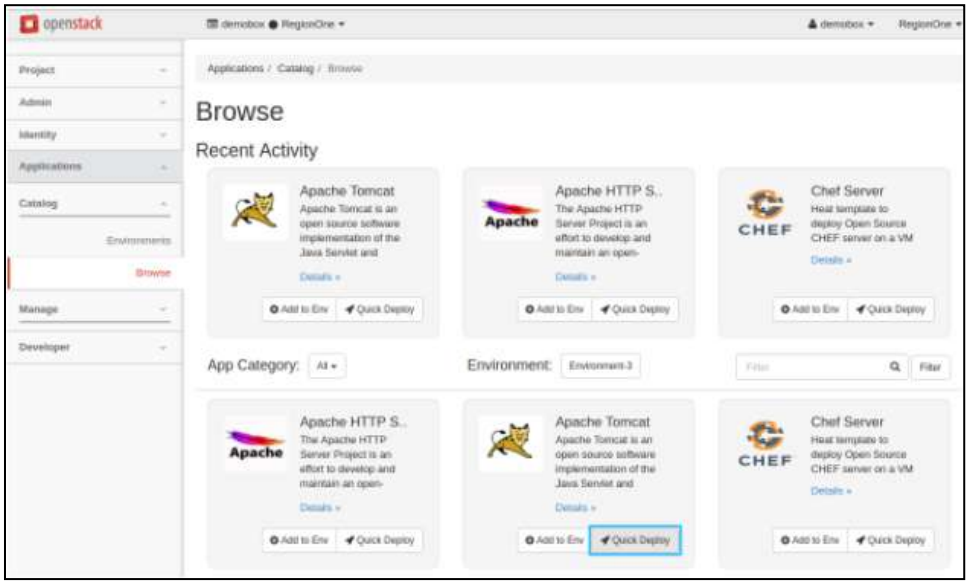
Now the application appears in the Component List section on the Environment Components page.

From applications catalog page

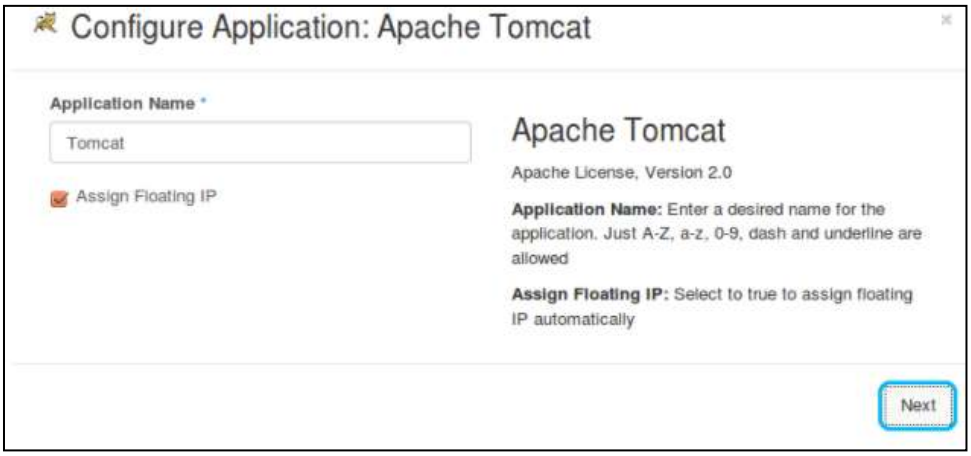
In OpenStack dashboard, navigate to Applications > Catalog > Browse. On the Applications catalog page, use one of the following methods:

Quick deploy. Automatically creates an environment, adds the selected application, and redirects you to the page with the environment components. Add to Env. Adds an application to an already existing environment.

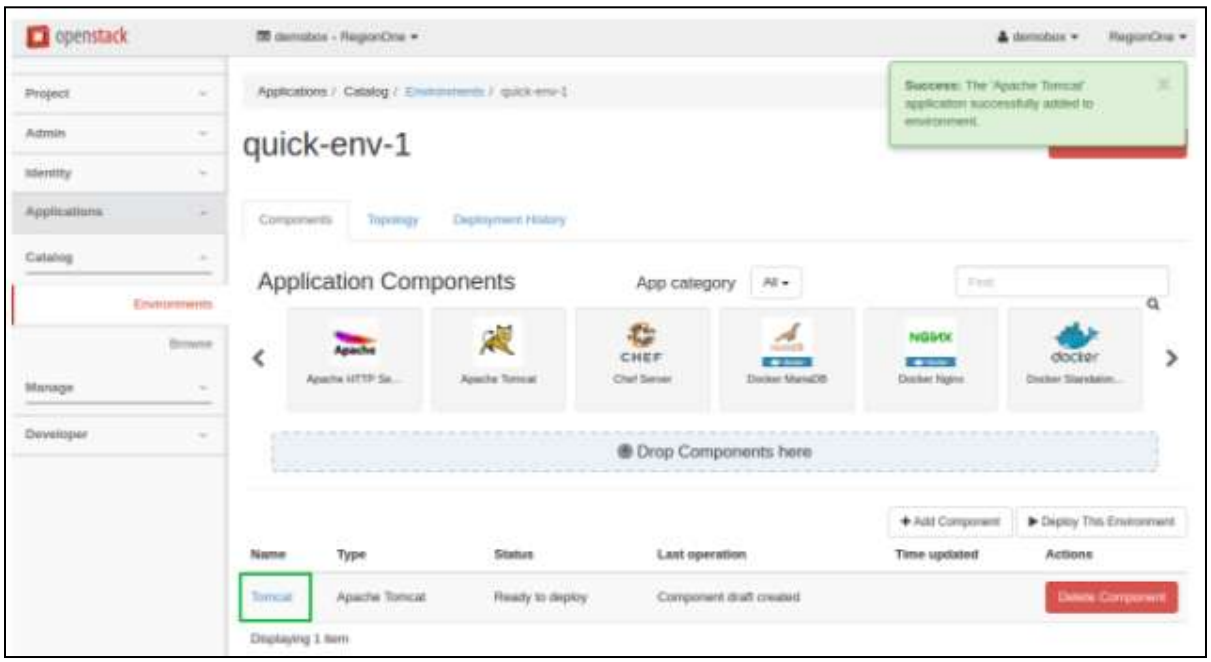
Quick Deploy button : Find the application you want to add and click Quick Deploy. Let's add Apache Tomcat, for example.



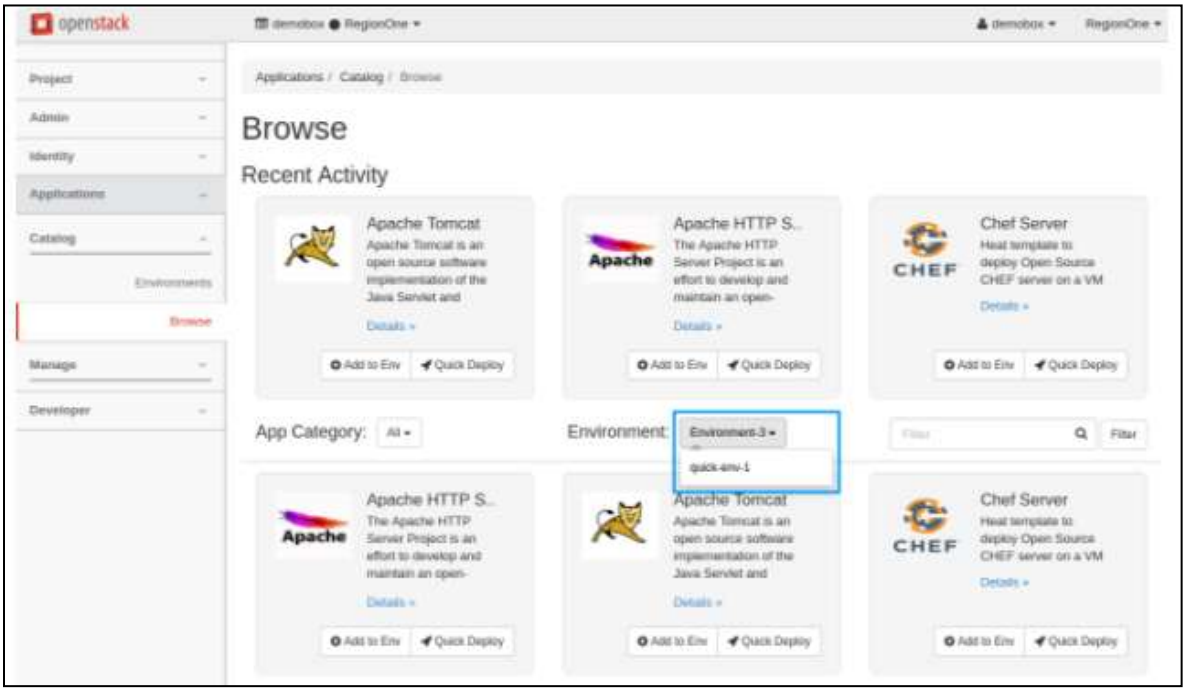
Configure the application. Note that the settings may vary from app to app and are predefined by the application author. When done, click Next, then click Create. In the example below we assign a floating IP address.



Now the Apache Tomcat application is successfully added to an automatically created quick-env-1 environment.

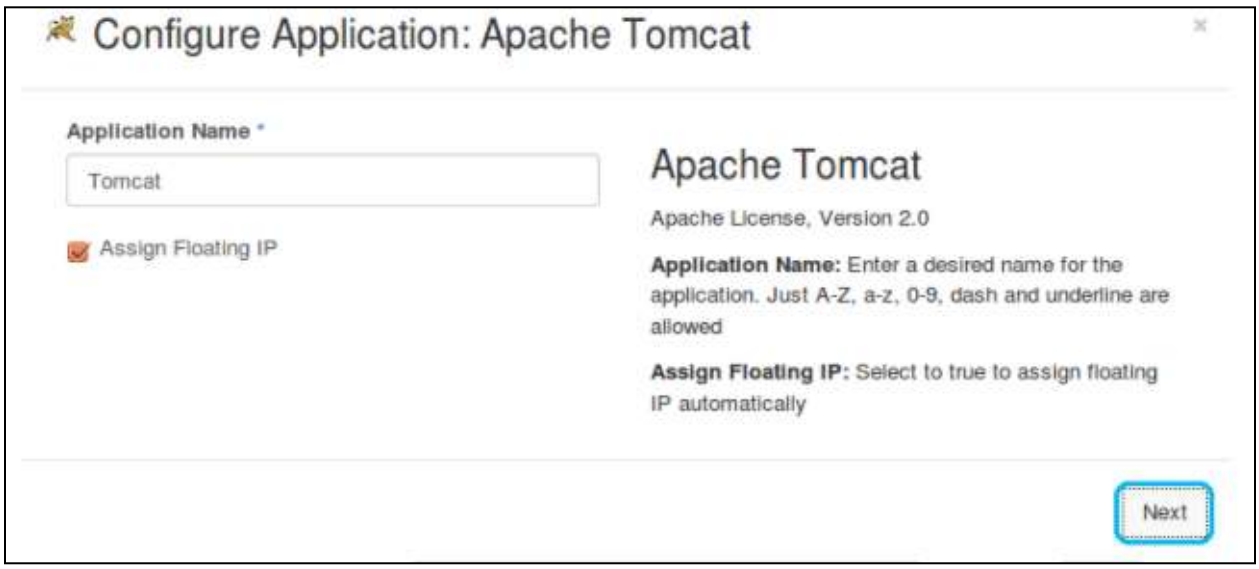


Add to Env button : From the Environment drop-down list, select the required environment.



Find the application you want to add and click Add to Env. Let's add Apache Tomcat, for example. Configure the application and click Next. Note that the settings may vary from app to

app and are predefined by the application author. In the example below we assign a floating IP address.



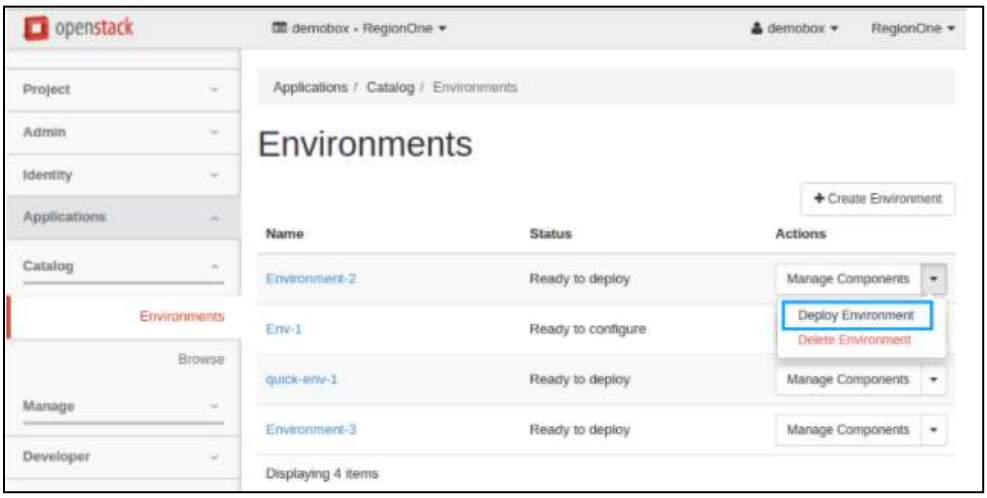
To add more applications, check Add more applications to the environment, then click Create and repeat the steps above. Otherwise, just click Create.

Deploy an environment

Make sure to add necessary applications to your environment, then deploy it following one of the options below:

- Deploy an environment from the Environments page

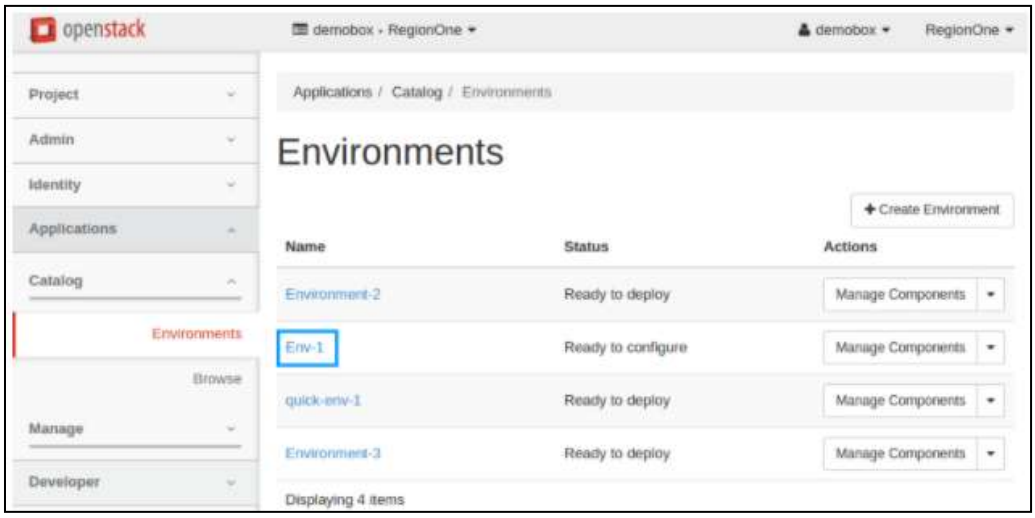
In OpenStack dashboard, navigate to Applications > Catalog > Environments. Select Deploy Environment from the Actions drop-down list next to the environment you want to deploy.



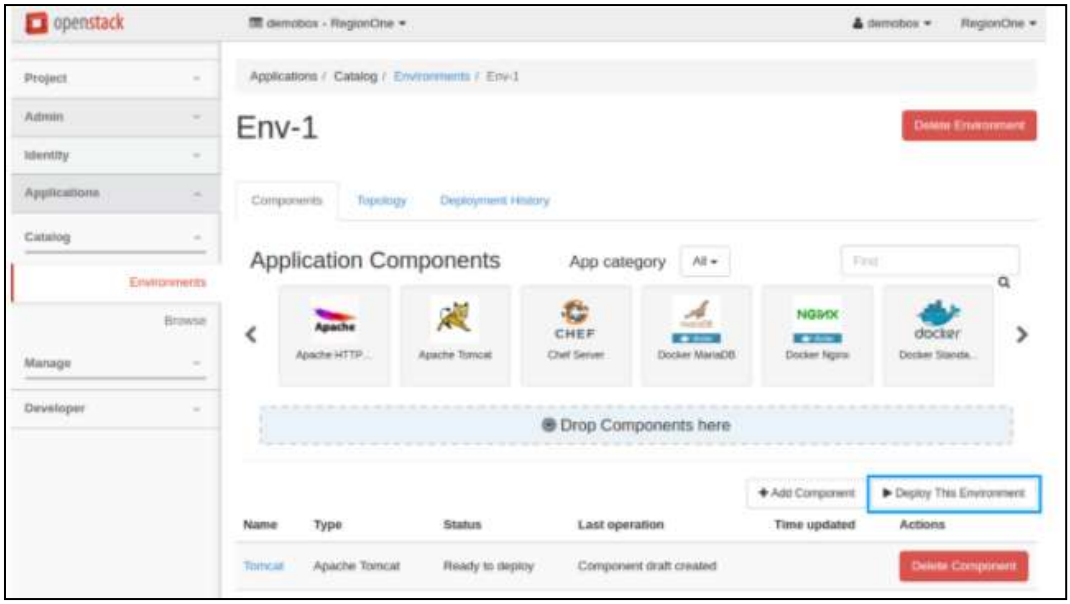
It may take some time for the environment to deploy. Wait for the status to change from Deploying to Ready. You cannot add applications to your environment during deployment.

- Deploy an environment from the Environment Components page

In OpenStack dashboard, navigate to Applications > Catalog > Environments. Click the name of the environment you want to deploy.



On the Environment Components page, click Deploy This Environment to start the deployment.



It may take some time for the environment to deploy. You cannot add applications to your environment during deployment. Wait for the status to change from Deploying to Ready. You can check the status either on the Environments page or on the Environment Components page.

CONCLUSION :

- From this experiment, we learnt about open source cloud solutions and we prepared a case study on OpenStack which is one of the available open source cloud solutions in the market.
- Some advantages of using open-source cloud solutions can be it is cost efficient, since it is open source, the community will be large enough to improve and develop this project.
- Main advantage will be that, we own our data, any place on the cloud is owned by the user.
- Some limitations of using open-source cloud solutions can be technical knowledge is important to change the code of the tool, support teams will be unofficial and will rely on the user base who develop it.
- Finally, in the case study of Openstack which was prepared we learned how to configure and install openstack and how to deploy application using Openstack.

REFERENCES :

1. https://geekflare.com/opensource-cloud-platforms/

2. https://www.openstack.org/

3. https://docs.openstack.org/murano/pike/admin/deploy_murano/prerequisites.html

4. https://docs.openstack.org/mitaka/install-guide-ubuntu/horizon-install.html

5. https://docs.openstack.org/murano/pike/user/userguide/manage_applications.html

6. https://solutionsreview.com/cloud-platforms/the-pros-and-cons-of-open-source-cloud-computing/

7. https://www.rutter-net.com/blog/the-benefits-and-challenges-of-using-openstack-in-your-business