

Deploying OpenStack on CentOS for Personalized Cloud Environment Using the KVM Hypervisor

¹Rajat Sharma, ²Mayur Deshmukh, ³Swapnil Pote

^{1,2}Acropolis Technical Campus, Rajiv Gandhi Proudhyogiki Vishwavidyalaya, Indore, India

³CTO, BigData Infomatics, India

Email: rajat223098@gmail.com, mayurdeshmukh4@gmail.com

Abstract

Cloud computing has proven to be a successful Computing Technology as it is adopted by leading organizations worldwide. Considering the applications of various public Clouds, such as Amazon EC2, private and hybrid Cloud deployments this technology is evolving like no other. These type of computing environment is rather based on a closed source architecture. Therefore the availability of free open source Cloud platforms is essential for further development of private and hybrid Cloud computing environments. OpenStack is a free and open-source software platform for cloud computing, mostly deployed as infrastructure-as-a-service (IaaS) originally released by Rackspace and NASA, where it provides virtual servers and other resources to customers. OpenStack eliminates the lack of a comprehensive Cloud platform with a fast pace of development for open source community and it is supported by an active community of people and large organizations. Since OpenStack is designed to operate in a large-scale working environment. In this work, we go through and discuss the steps required to deploy OpenStack in a small-scale environment or rather utilizing functionalities of OpenStack for creating a personalized cloud architecture for own use. Every step discussed in this paper is implemented to utilize OpenStack's open source architecture for building a personal cloud application to provide computing resources to its users through a web-based dashboard.

Keywords- Cloud computing, OpenStack, KVM, Linux Operating System, CentOS

INTRODUCTION

The Cloud computing technology uses virtualization to deliver computing resources to users on-demand basis. It enables users to flexibly adjust their resource consumption according to their workload. These advantages of the Cloud computing technologies allow one to avoid high investments in a computing infrastructure, thus reducing the time and resources which can further used in other innovation. Cloud computing resources are delivered to users through three major service models:

- Infrastructure as a service (IaaS) is a cloud computing service in which users can access to computing resources such as storage, networking,

computing remotely. Organizations use their own platforms architecture and applications to provide these services.

- Platform as a service (PaaS) is a cloud computing service that provides users with a environment in which they can manage, develop and execute applications. In addition to computing and other computing resources, users are able to use a set of prebuilt tools to develop and customize their own applications.

Examples of PaaS are Google App Engine and Microsoft Azure

- Software as a service (SaaS) is a cloud computing service that provides users a

cloud-based software. Users cannot install applications on their local devices. Rather, the applications are installed on a remote cloud network accessed through the web or an API. Examples of SaaS are applications from the Amazon Web Services .

In this study, we focus on IaaS.

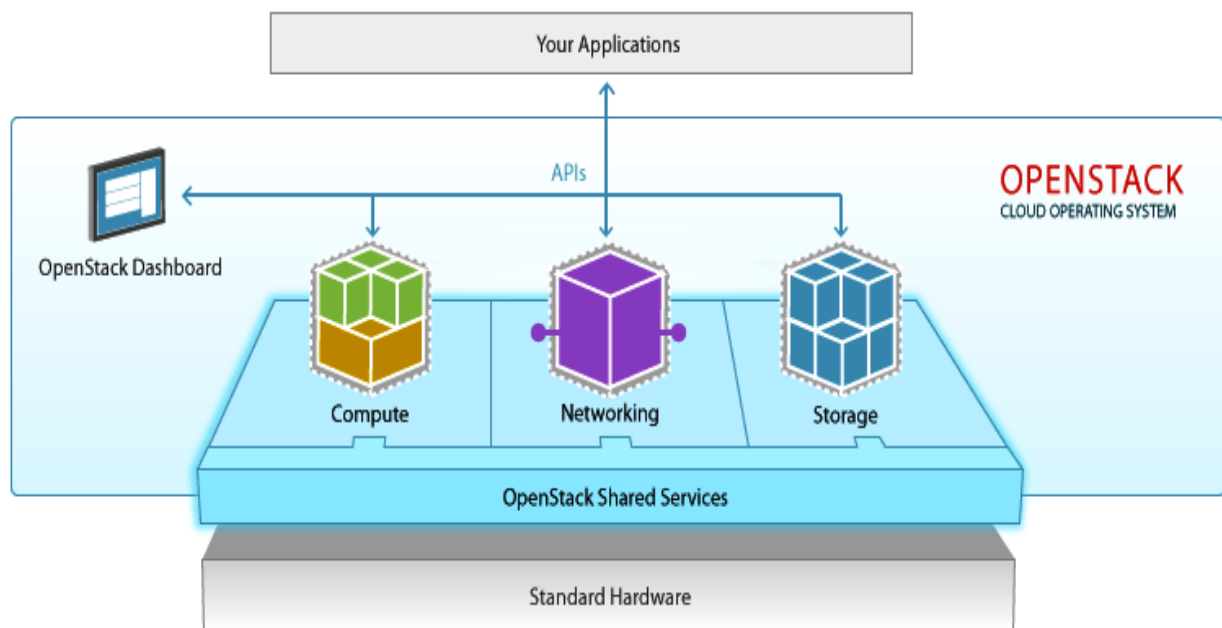
Cloud architecture is further broken into three categories.

- Public Cloud: computing resources are provided publicly over the Internet .
- Private Cloud: the Cloud infrastructure is a closed environment owned by an organization, and operated internally.
- Hybrid Cloud: resources are provided by a mixture of a private and public cloud services.
- In summary, this paper discusses the installation process of the following software:

- CentOS 7: It is a free Linux Operating System derived from the Red Hat Enterprise Linux (RHEL) distribution.
- KVM: It is a hypervisor providing full virtualization for Linux hardware The project
- assisted virtualization support of the Intel VT and AMD-V chipsets.
- OpenStack :OpenStack is a cloud operating system that controls resources such as compute, storage, and networking.

Overview of the OpenStack Platform

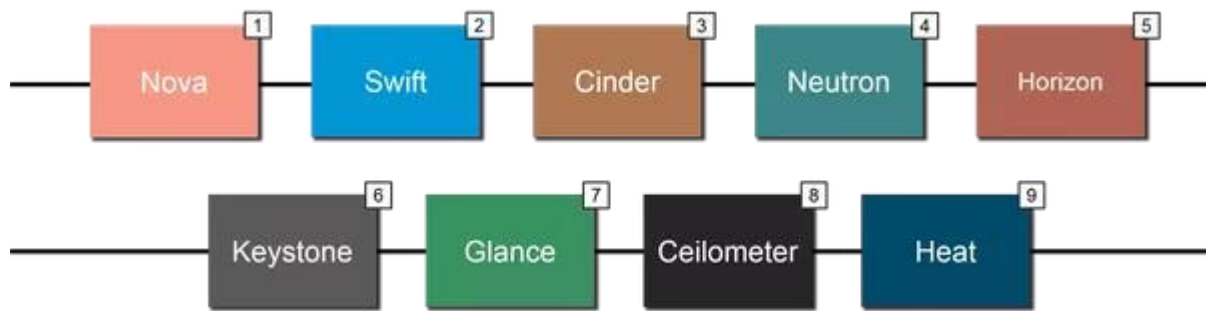
OpenStack is a cloud operating environment that controls resources such as compute, storage, and networking. These components are managed with a dashboard which gives administrators complete control while the end users can utilize resources through a web based interface.



Components of OpenStack

Openstack is a collection of different individual parts, as it is open source cloud computing software. Any one can modify its components to optimize their openstack based cloud infrastructure and to meet their own needs. However, the

openstack community has collaboratively identified some key components that are part of the core of Openstack which are distributed as components of any openstack system and officially maintained by the Openstack community.



a) Nova

Nova is the primary computing engine behind OpenStack. This component allows the end user to create and manage large number of virtual servers using the machine images. It is the brain of the Cloud. OpenStack computes provisions and manages large networks of virtual machines.

b) Swift

Swift is a storage system for objects and files. It stores and retrieves data objects through a web based APIs. Therefore, users don't have to worry about the capacity on a single system behind the software. Because of this data replication and scale out architecture this component makes sure that data is backed up periodically in case of failure of a machine or network service.

c) Neutron

A networking component of Openstack which ensure that each of the components of OpenStack can communicate with one another quickly and efficiently. It is scalable and API-driven system for managing networks which enhance the networking capability for OpenStack.

d) Horizon

This component provides a web-based portal or an interface to interact with all the underlying OpenStack services. It is the only graphical interface to OpenStack. It is a Dashboard which helps system administrators take a look at what is going on in the cloud, and to manage it as needed.

e) Cinder

A block storage service which provides block storage to running cloud instances. The efficient architecture of Cinder makes the task of creating and managing block storage devices easy.

f) Keystone

This component provides a central directory of all users in all OpenStack services where all of the users are tracked against all of the services provided by the cloud which they have permission to use. It is used for authentication and authorization service for all OpenStack services.

g) Ceilometer

This component monitors the usage of all the services of each user and performs the allocating task accordingly. It keeps a verified count of each user's system usage and the components of cloud environment.

h) Glance

This component stores and retrieves the virtual machine disk image. Glance allows these images to be used as Operating System templates when deploying new virtual machine instances.

i) Orchestration Heat

An orchestration component of OpenStack, which manages the infrastructure required for a cloud service to run. This component manages various cloud applications through an OpenStack-native API. It allows users to store the requirements of a various application in a single file that can be referred later what resources are necessary for a particular application.

j) Trove

A Database Service for OpenStack. It allows users to effectively use the features of a database without handling complex administrative tasks such as deployment, configuration, backups and restores in a database.

KVM

KVM stands for Kernel-based Virtual Machine. It is a full virtualization solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V). Using KVM, one can run multiple virtual machines running unmodified Linux or Windows images. Each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, etc. KVM is open source software. The kernel component of KVM is included in mainline Linux, it supports all major operating systems including Linux and Windows. It is developed by Red Hat Corporation to provide a virtualization solution and services on the Linux operating system platform.

KVM Hypervisor

KVM hypervisor is a program in which multiple operating systems can be operated independently while sharing a single hardware host. In KVM, the Linux kernel efficiently manage and improve the performance in virtual environments. The hypervisor creates Virtual Machine environments and coordinates calls for processor, memory, hard disk, network, and other resources through the host OS.

Step by step Openstack Deployment Hardware Setup -

Compute nodes are responsible for deploying virtual machine instances once they have been initiated. These compute nodes will be the hardware infrastructure of openstack. Therefore they should always have enough RAM and disk space available to handle the workload.

Processor

64 bit Multicore processor

Infrastructure hosts should have multicore processors for best performance. Some services, such as MySQL, benefit from additional CPU cores and other technologies, such as Hyper Threading.

Memory

A minimum of 8 GB of RAM is recommended.

Add additional RAM to this requirement based on the amount of memory that you intend to make available to virtual machine instances.

Disk Space

A minimum of 50 GB of available disk space is recommended.

Add additional disk space to this requirement based on the amount of space that you intend to make available to virtual machine instances.

1 TB of disk space is recommended for a realistic environment capable of hosting multiple instances of varying sizes.

Network Interface Cards

2 x 1 Gbps Network Interface Cards.

Hardware used in this Project

Processor: Intel Core i7 (6th Gen) 6700HQ / 2.6 GHz

Ram : 16 GB DDR4 SDRAM 2133 MHz

HDD: 1 TB

Installation Procedure**CentOS**

Step 1: Download The ISO Image

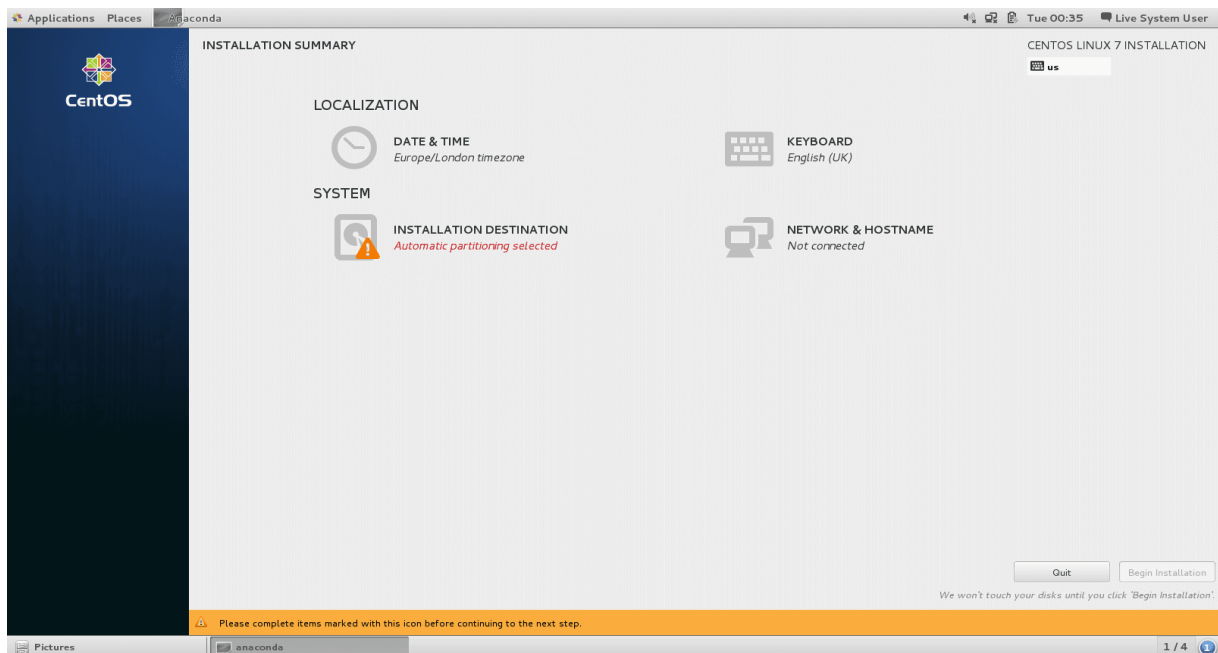
To get a copy of CentOS 7 download from its source mirror.

Step 2: Make A bootable Drive after you have downloaded the ISO image.

Step 3: Begin Installation

Step 4: Change The Installation Destination

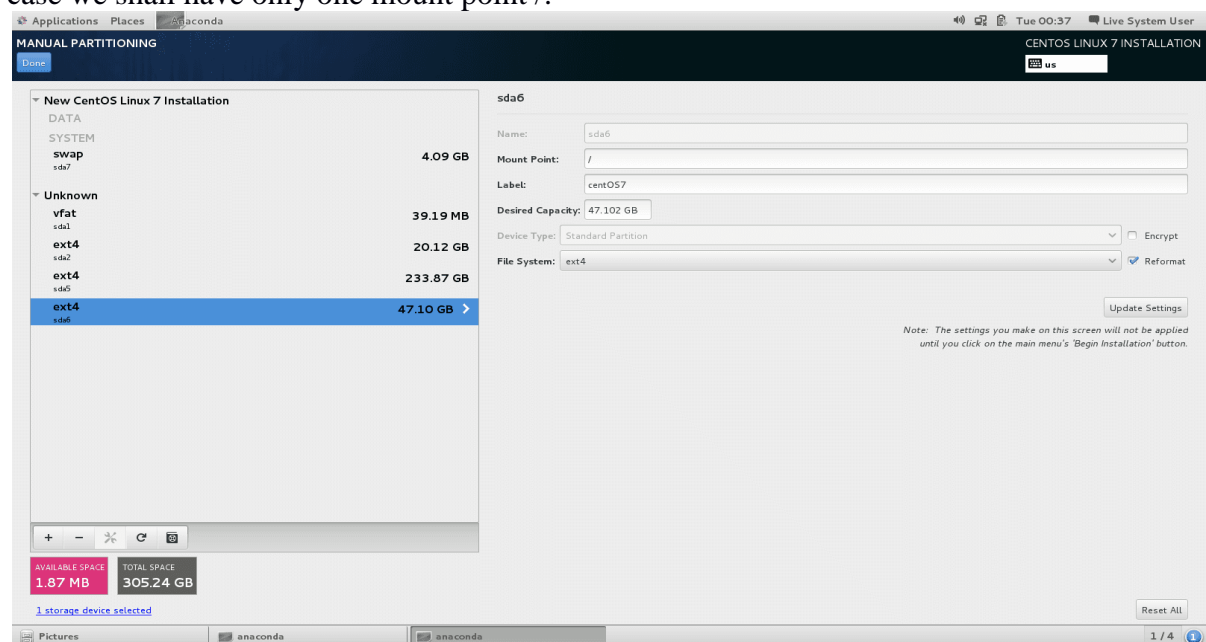
By default the Anaconda installer will choose automatic partitioning for your hard disk.



Step 5: Select The Partitioning Scheme

Step 6: Create A Mount Point

The next step is to create a mount point where the root partition will be installed. For this case we shall have only one mount point /.



Step 7: Begin Installation

Step 8: Set Up Root Password

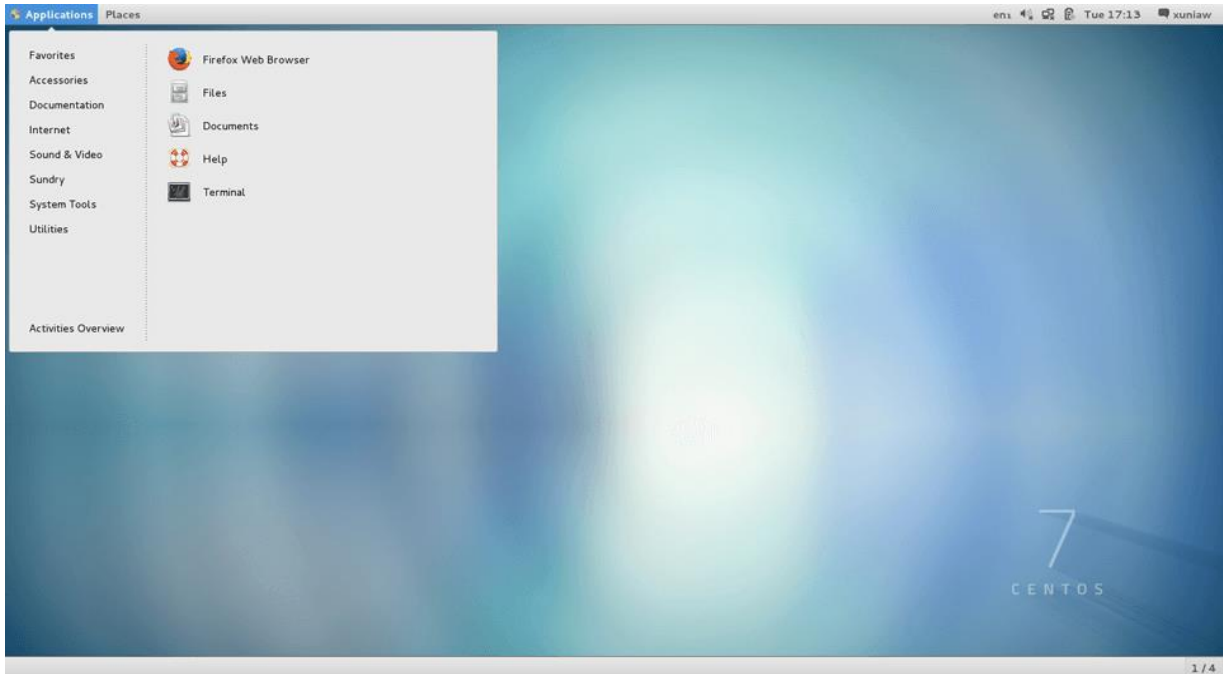
Step 9: Create a User Account

Enter the correct details and if this is the administrator account.

Step 10: Complete Installation

The installer will complete installing the software and the bootloader.

Logout from the Live system and login to your new installed operating system.
Shown below is the environment of CentOS



CentOS 7 has support for the **xfs file system** which is suitable especially in a distributed type of environment. XFS is known for its ability to handle parallel I/O compared to ext4.

3.2.2 Openstack

Install and Configure OpenStack Ocata

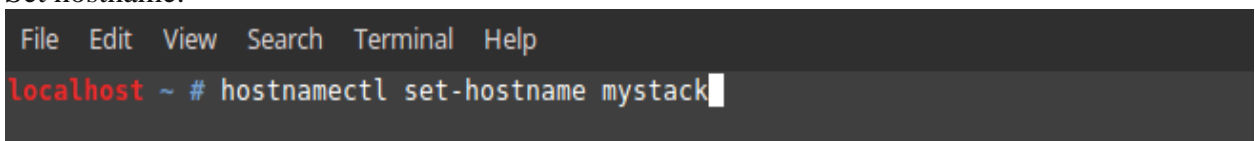
We will focus on installing and configuring OpenStack Ocata using RDO and the packstack installer. RDO is a community platform around Red Hat's OpenStack Platform. It allows you to test the latest OpenStack capabilities on a stable platform such as Red Hat Enterprise Linux (RHEL) or CentOS.

Install RHEL or CentOS 7.3 or 7.4.

Ensure name resolution is working.



Set hostname.



- Disable firewalld since this is for a lab environment.
- # systemctl disable firewalld
- # systemctl stop firewalld
- Disable NetworkManager, it is still not recommended for Liberty (at least RDO).

```
# systemctl stop NetworkManager
# systemctl disable NetworkManager
Install yum-utils and update the system.
# yum install -y yum-utils
# yum update -y
```

- Reboot.
 - Install packstack packages.
- ```
yum install -y openstack-packstack
```
- INSTALL USING ANSWERS FILE**
- Create packstack answers file for customizing the installer.

```
packstack --gen-answer-file /root/answers.txt
```

Update the packstack answers file and enable other OpenStack services.

```
The amount of service workers/threads to use for each service.
Useful to tweak when you have memory constraints. Defaults to the
amount of cores on the system.
CONFIG_SERVICE_WORKERS=%{:::processorcount}

Specify 'y' to install MariaDB. ['y', 'n']
CONFIG_MARIADB_INSTALL=y

Specify 'y' to install OpenStack Image Service (glance). ['y', 'n']
CONFIG_GLANCE_INSTALL=y

Specify 'y' to install OpenStack Block Storage (cinder). ['y', 'n']
CONFIG_CINDER_INSTALL=n

Specify 'y' to install OpenStack Shared File System (manila). ['y',
'n']
CONFIG_MANILA_INSTALL=n

Specify 'y' to install OpenStack Compute (nova). ['y', 'n']
CONFIG_NOVA_INSTALL=y

Specify 'y' to install OpenStack Networking (neutron) ['y']
CONFIG_NEUTRON_INSTALL=y

Specify 'y' to install OpenStack Dashboard (horizon). ['y', 'n']
CONFIG_HORIZON_INSTALL=y

Specify 'y' to install OpenStack Object Storage (swift). ['y', 'n']
CONFIG_SWIFT_INSTALL=n

Specify 'y' to install OpenStack Metering (ceilometer). Note this
will also automatically install gnocchi service and configures it as
the metrics backend. ['y', 'n']
CONFIG_CEILOMETER_INSTALL=n

Specify 'y' to install OpenStack Telemetry Alarming (Aodh). Note
Aodh requires Ceilometer to be installed as well. ['y', 'n']
CONFIG_AODH_INSTALL=n

Specify 'y' to install OpenStack Events Service (panko). ['y', 'n']
CONFIG_PANKO_INSTALL=n

Specify 'y' to install OpenStack Data Processing (sahara). In case
of sahara installation packstack also installs heat. ['y', 'n']
```

```
Specify 'y' to install OpenStack Data Processing (sahara). In case
of sahara installation packstack also installs heat. ['y', 'n']
CONFIG_SAHARA_INSTALL=n

Specify 'y' to install OpenStack Orchestration (heat). ['y', 'n']
CONFIG_HEAT_INSTALL=n

Specify 'y' to install OpenStack Container Infrastructure
Management Service (magnum). ['y', 'n']
CONFIG_MAGNUM_INSTALL=n

Specify 'y' to install OpenStack Database (trove) ['y', 'n']
CONFIG_TROVE_INSTALL=n

Specify 'y' to install OpenStack Bare Metal Provisioning (ironic).
['y', 'n']
CONFIG_IRONIC_INSTALL=n

Specify 'y' to install the OpenStack Client packages (command-line
tools). An admin "rc" file will also be installed. ['y', 'n']
CONFIG_CLIENT_INSTALL=y

Comma-separated list of NTP servers. Leave plain if Packstack
should not install ntpd on instances.
CONFIG_NTP_SERVERS=

Comma-separated list of servers to be excluded from the
installation. This is helpful if you are running Packstack a second
time with the same answer file and do not want Packstack to
overwrite these server's configurations. Leave empty if you do not
need to exclude any servers.
EXCLUDE_SERVERS=

Specify 'y' if you want to run OpenStack services in debug mode;
otherwise, specify 'n'. ['y', 'n']
CONFIG_DEBUG_MODE=n

Server on which to install OpenStack services specific to the
controller role (for example, API servers or dashboard).
CONFIG_CONTROLLER_HOST=192.168.3.224
```

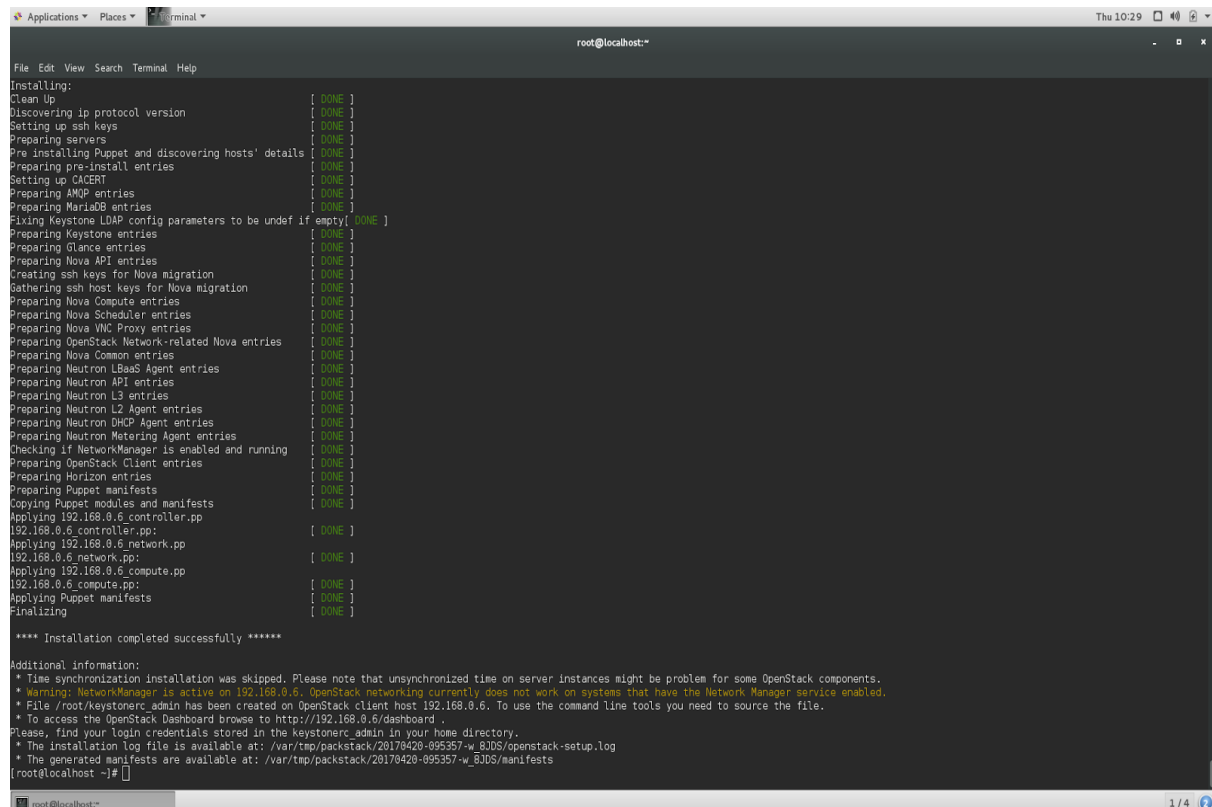
## Configuration Files

We customized answer file as follows -

|                             |                      |
|-----------------------------|----------------------|
| CONFIG_MARIADB_INSTALL=y    | [Database]           |
| CONFIG_GLANCE_INSTALL=y     | [Image Service]      |
| CONFIG_CINDER_INSTALL=n     | [Block Storage]      |
| CONFIG_MANILA_INSTALL=n     | [Shared File System] |
| CONFIG_NOVA_INSTALL=y       | [Compute Engine]     |
| CONFIG_NEUTRON_INSTALL=y    | [Network]            |
| CONFIG_HORIZON_INSTALL=y    | [Dashboard]          |
| CONFIG_SWIFT_INSTALL=n      | [Object Storage]     |
| CONFIG_CEILOMETER_INSTALL=n | [Telemetry]          |
| CONFIG_AODH_INSTALL=n       | [Alarming]           |
| CONFIG_GNOCCHI_INSTALL=n    | [Resource Indexing]  |

Only MariaDB(Database),GLANCE(Image Service), NOVA(Compute),NEUTRON(Network) and HORIZON(Dashboard) are installed because these are the fundamental components to use OpenStack without these you cannot deploy OpenStack. There is no need to install any other components since we are deploying openstack in a small-size or personal cloud architecture.

- Install OpenStack Liberty using packstack.  
# packstack --answer-file /root/answers.txt
- Source the keystone admin profile.  
# . /root/keystonerc\_admin



```

root@localhost:~# packstack --answer-file /root/answers.txt
Installing:
Clean Up [DONE]
Discovering ip protocol version [DONE]
Setting up ssh keys [DONE]
Preparing servers [DONE]
Pre installing Puppet and discovering hosts' details [DONE]
Preparing pre-install entries [DONE]
Setting up CACERT [DONE]
Preparing AMQP entries [DONE]
Preparing MariaDB entries [DONE]
Fixing Keystone LDAP config parameters to be undef if empty [DONE]
Preparing Keystone entries [DONE]
Preparing Glance entries [DONE]
Preparing Nova API entries [DONE]
Creating ssh keys for Nova migration [DONE]
Gathering ssh host keys for Nova migration [DONE]
Preparing Nova Compute entries [DONE]
Preparing Nova Scheduler entries [DONE]
Preparing Nova VNC Proxy entries [DONE]
Preparing OpenStack Network-related Nova entries [DONE]
Preparing Nova Common entries [DONE]
Preparing Neutron LBaaS Agent entries [DONE]
Preparing Neutron API entries [DONE]
Preparing Neutron L3 entries [DONE]
Preparing Neutron L2 Agent entries [DONE]
Preparing Neutron DHCP Agent entries [DONE]
Preparing Neutron Metering Agent entries [DONE]
Checking if NetworkManager is enabled and running [DONE]
Preparing OpenStack Client entries [DONE]
Preparing Horizon entries [DONE]
Preparing Puppet manifests [DONE]
Copying Puppet modules and manifests [DONE]
Applying 192.168.0.6.controller.pp [DONE]
Applying 192.168.0.6.controller.pp [DONE]
Applying 192.168.0.6.network.pp [DONE]
Applying 192.168.0.6.network.pp [DONE]
Applying 192.168.0.6.compute.pp [DONE]
Applying 192.168.0.6.compute.pp [DONE]
Applying Puppet manifests [DONE]
Finalizing [DONE]

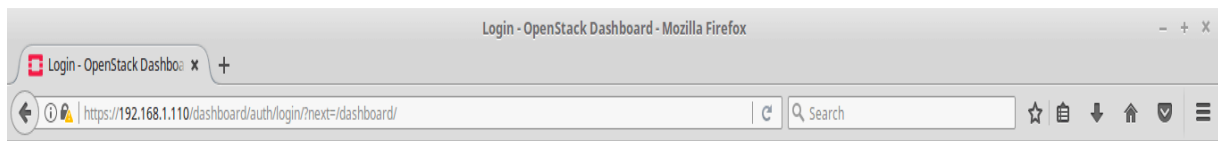
**** Installation completed successfully ****

Additional information:
* Time synchronization installation was skipped. Please note that unsynchronized time on server instances might be problem for some OpenStack components.
* Warning: NetworkManager is active on 192.168.0.6. OpenStack networking currently does not work on systems that have the Network Manager service enabled.
* File /root/keystonerc_admin has been created on OpenStack client host 192.168.0.6. To use the command line tools you need to source the file.
* To access the OpenStack Dashboard browse to http://192.168.0.6/dashboard.
Please, find your login credentials stored in the keystonerc_admin in your home directory.
* The installation log file is available at: /var/tmp/packstack/20170420-095357-w_8JDS/openstack-setup.log
* The generated manifests are available at: /var/tmp/packstack/20170420-095357-w_8JDS/manifests
[root@localhost ~]#

```

The installation of openstack on your centos is now completed and when you open your dashboard console it will look like this





## CUSTOMIZING YOUR OPENSTACK WEB APPLICATION(DASHBOARD)

We have installed the OpenStack dashboard through the openstack-dashboard package. we can customize the dashboard with our own colors, logo, and site title updating a CSS file.

1. Create a graphical logo with a transparent background. The text TGen Cloud in this example is rendered through .png files of multiple sizes created with a graphics program. Use a 200×27 for the logged-in banner graphic, and 365×50 for the login screen graphic.
2. Set the HTML title, which appears at the top of the browser window, by adding the following line to

/etc/openstack-dashboard/local\_settings:  
SITE\_BRANDING = "Example, Inc. Cloud"

3. Upload your new graphic files to the following location:  
/usr/share/openstack-dashboard/openstack\_dashboard/static/dashboard/img/
4. Create a CSS style sheet in the following directory:  
/usr/share/openstack-dashboard/openstack\_dashboard/static/dashboard/css/
5. Change the colors and image file names, but directory paths should be the same. The following file shows you how to customize your CSS file:

```

/*
 * New theme colors for dashboard that override the defaults:
 * dark blue: #355796 / rgb(53, 87, 150)
 * light blue: #BAD3E1 / rgb(186, 211, 225)
 *
 * By Preston Lee <plee@tgen.org>
 */
h1.brand {
background: #355796 repeat-x top left;
border-bottom: 2px solid #BAD3E1;
}
h1.brand a {
background: url(../img/my_cloud_logo_small.png) top left no-repeat;
}
#splash .login {
background: #355796 url(../img/my_cloud_logo_medium.png) no-repeat center
35px;
}
#splash .login .modal-header {
border-top: 1px solid #BAD3E1;
}
.btn-primary {
background-image: none !important;
background-color: #355796 !important;
border: none !important;
box-shadow: none;
}
.btn-primary:hover,
.btn-primary:active {
border: none;
box-shadow: none;
background-color: #BAD3E1 !important;
text-decoration: none;
}

```

6. Open the following HTML template in an editor: /usr/share/openstack-dashboard/openstack\_dashboard/templates/\_stylesheets.html
7. Add a line to include your custom.css file

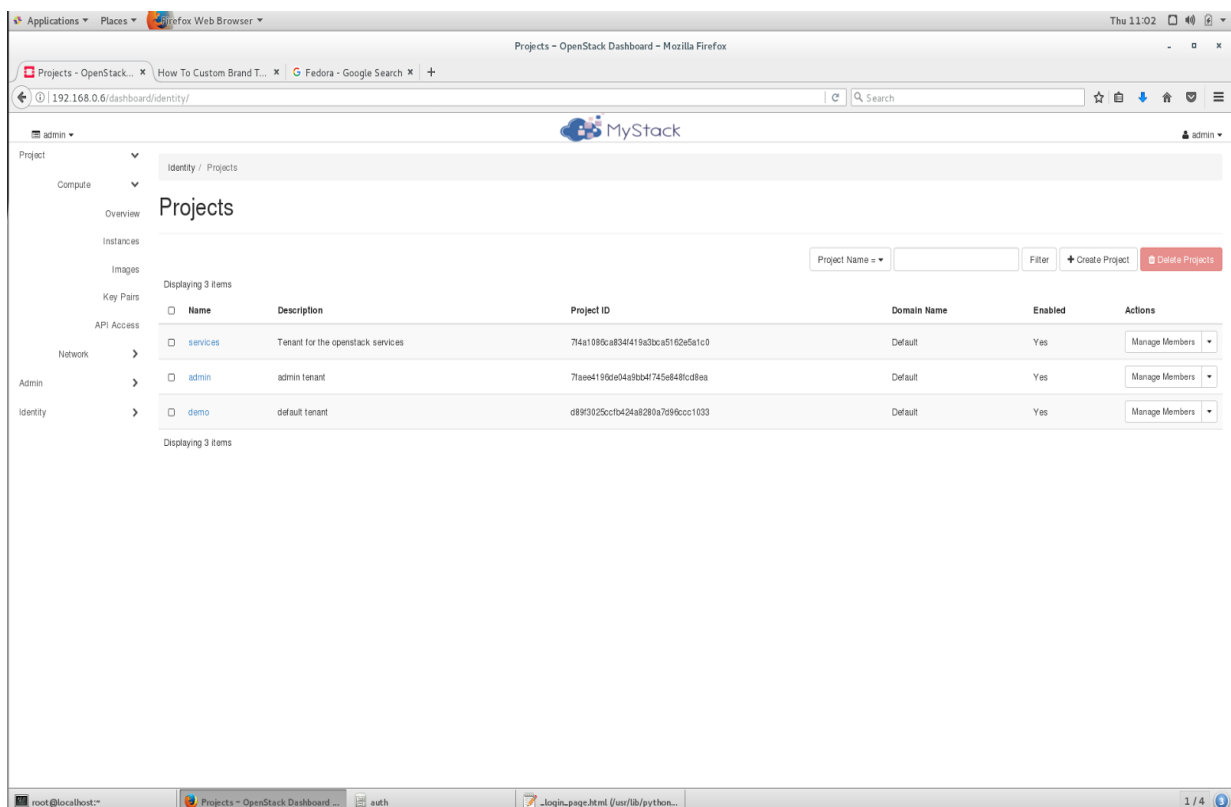
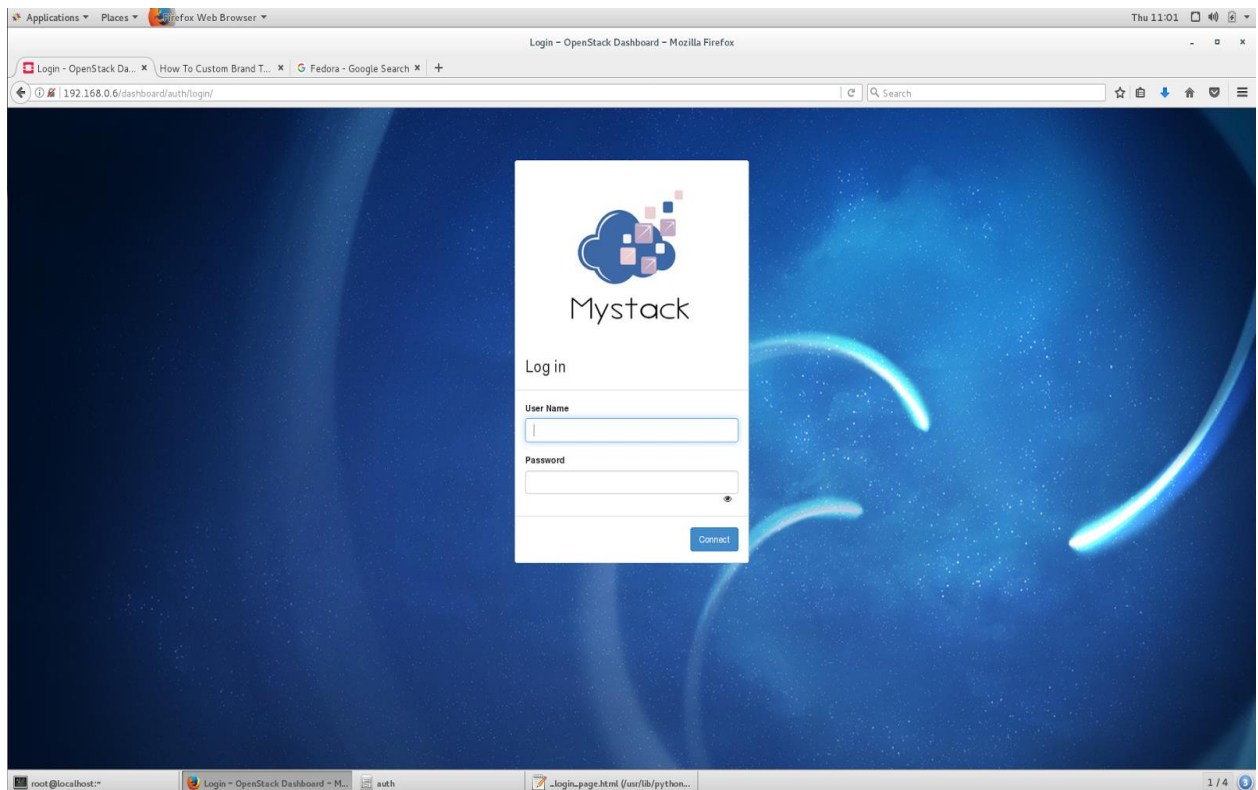
```

...
<link href='{ STATIC_URL }bootstrap/css/bootstrap.min.css' media='screen'
rel='stylesheet' />
<link href='{ STATIC_URL }dashboard/css/{% choose_css %}' media='screen'
rel='stylesheet' />
<link href='{ STATIC_URL }dashboard/css/custom.css' media='screen'
rel='stylesheet' />
...

```

8. Restart httpd

9. Reload the dashboard in your browser to view your changes.  
The dashboard of openstack can be customized like this



## ADVANTAGES AND DISADVANTAGES OF USING OPENSTACK

### Advantages

- Open Source - OpenStack is a open source technology therefore it is very cost effective and easy to deploy in comparison to its other counterparts.
- Accelerated Innovation - Openstack's robust capabilities offer developers and engineers rapid and better resource availability. Openstack provides on demand hardware and software resource on the go, therefore, reducing development period.
- Scalability - OpenStack provide full virtual machine support in its system. So the system admin does not rely on any other third-party services to expand his services. Any vendor can create servers(virtual machines) according to their requirement with customizable resources remotely.

### Disadvantages

- Complex to operate - Despite all its features OpenStack administration is not easy it requires talented and

skilled experts to utilize its full features according to requirement.

- Implementation takes longer time - If anyone wants to use OpenStack up to its full potential. First, they will require skilled personnel, second to deploy openstack you have to setup and configure it from ground up i.e from installing a supported operating system. Both of these steps are very time consuming.

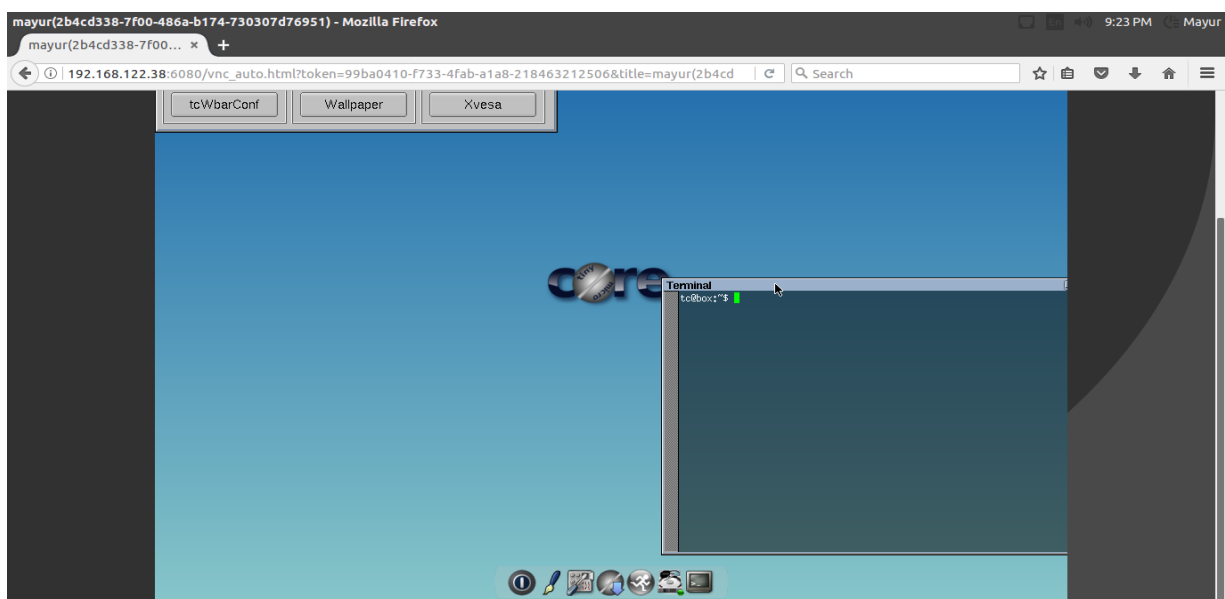
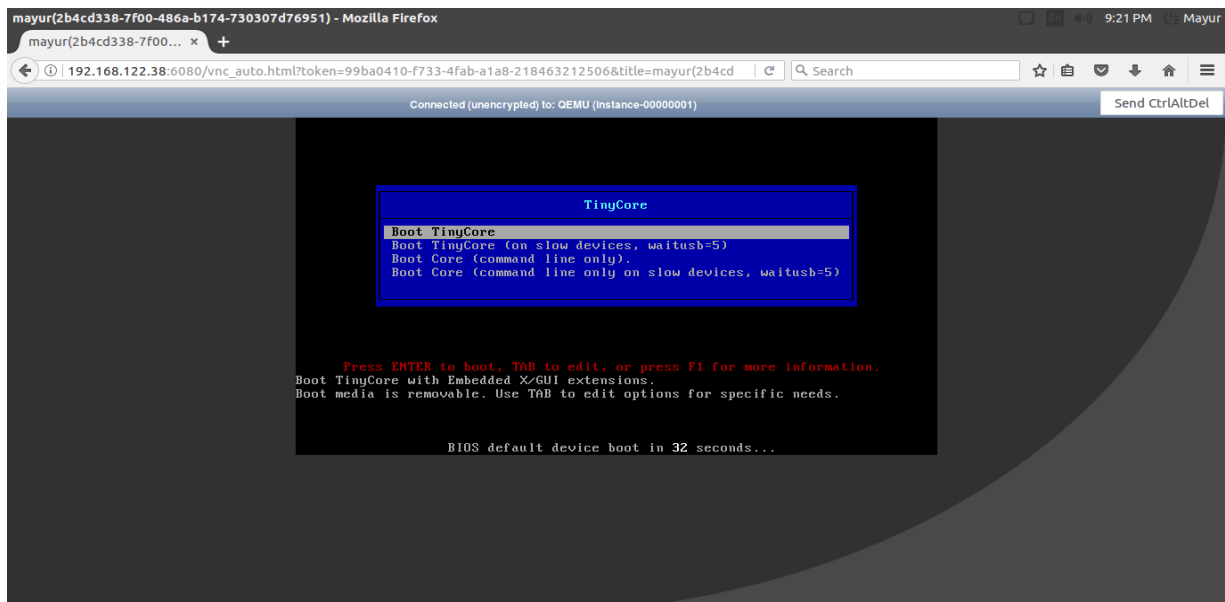
### FUTURE SCOPE

Over the years, OpenStack is able to attract a huge amount of audience because of its unique features and cost effectiveness but relatively very low level of acceptance. Using this technology for personal use can be complex and time-consuming because it is very difficult to deploy and experts suggest that it is still missing pieces. However, the evolution of open source technology is not that rapid because of lack of support. For instance, Mac os and windows are adopted by a huge number of users despite that they offer a closed source environment compared to Linux which not readily

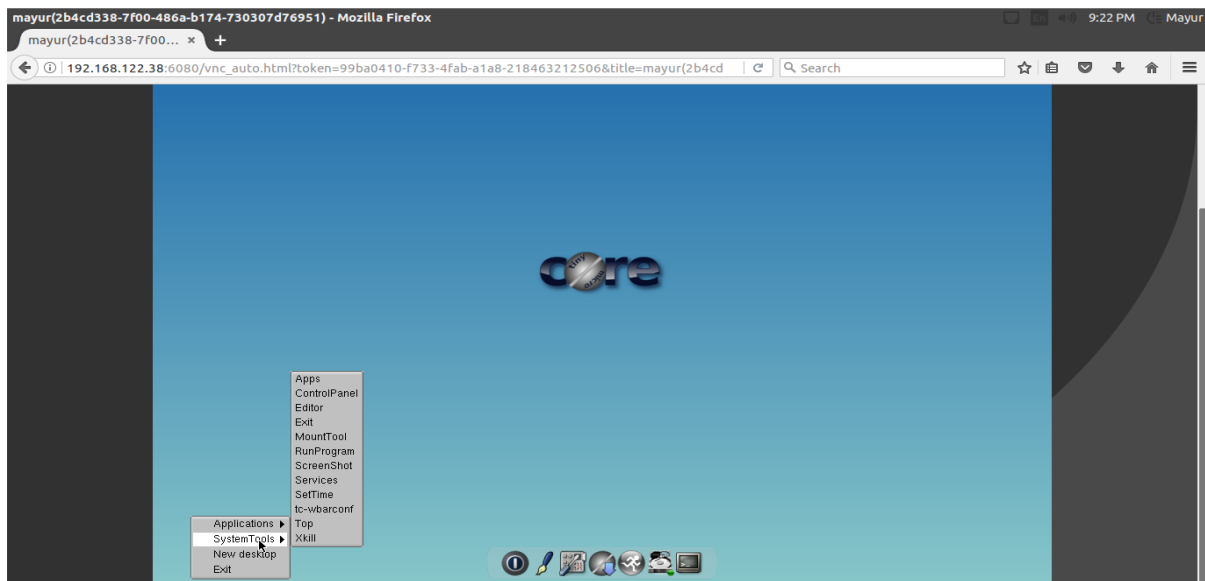
accepted by that amount of users. Mac os and windows provide a reliable user experience with a strong support which Linux distribution does not. But in past years number of Linux user has proliferated because of increased support by its community and robustness. Same can be expected for OpenStack, at this time people are migrating towards big service providers like AWS, but the unique features of OpenStack will definitely outshine within its competitors in the near future.

## RESULTS

OpenStack is fully functional and all the core components are working efficiently. Now we can create virtual machines of desired resources which are available on our server. Then these virtual machines can be accessed from a web-based dashboard from anywhere within the network. Shown below is a virtual machine instance running TinyCore Linux distribution and is fully operational through a web browser.







## CONCLUSION

We have gone through and discuss all the steps required to deploy openstack on a small to medium size cloud environment. We have started from bare hardware to a fully operating OpenStack infrastructure. We have included all the steps starting from installing CentOS on a machine followed by KVM hypervisor and continued through setting up configuration files, network gateway and selectively installing only the required services of OpenStack, so our machine can effectively operate on resource sharing technology within the parameters. Further, we have customized dashboard structure of the OpenStack web application and included all the required steps for the customization.

## REFERENCES

1. Cloud Architecture:  
<https://www.ibm.com/cloud/learn/iaas-paas-saas>
2. Openstack:  
<https://www.hivelocity.net/kb/what-is-openstack/>
- KVM:  
<https://www.fastwebhost.in/blog/what-is-kvm-virtualization/>
3. <https://linuxide.com/how-tos/centos-7-step-by-step-screenshots/>
4. [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux\\_Open\\_Stack\\_Platform/](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_Open_Stack_Platform/)