ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 4, Issue 11, November 2017

Pooling of Computing Resources in Private Cloud Deployment

[1] Adil Husain, [2] M Hamza Zaki, [3] Saiful Islam
[1][2] PG Student, [3] Associate Professor
[1][2][3] Department of Computer Engineering, Zakir Husain College of Engineering and Technology, A.M.U, Aligarh

***Abstract* - Resource pooling in cloud is considered as a shared pool of computing resources such as Cores, RAM, Storage, Disk etc. It helps in the resource allocation among users, thus increasing the utilization rate and saves cost. There are many large shared cloud resource pools such as Amazon EC2 cloud, Google Cloud etc which monitor the usage of computing resources and generates the billing to users based on the usage of resources. These are commercial public cloud and resource pooling implementation is done using cost-optimization approach. In this paper, we deployed a resource pool of opensource private cloud openstack using MAAS (Metal as a Service) and Juju. There are many opensource cloud platforms such as Openstack, Eucalyptus, Opennebula etc for deployment of private cloud. For production use, a openstack tool, MAAS (Metal as a Service) with Juju manages the computing resources more flexibly. First, the MAAS ubuntu server cluster is deployed on a private network in which machine Cores, RAM, Storage, Disk, etc are added dynamically using pre-defined powertype. Furthermore, Juju environment is configured and Juju Bootstrapping is done for assignment of openstack services. The pooling configuration is done step-by-step and a resource pool of six machines is successfully deployed, each machine lists the number of Cores, RAM, Storage, Disk, etc. As a result, the openstack services are assigned to deployed machines, thus allocation of resources would be easily monitored and controlled.**

Keywords — Juju; MAAS; Openstack; Resource Pooling

I. INTRODUCTION

In recent trends and development, Cloud Computing enables rapid provisioning of resources based on user demands. Basically, cloud utilizes the unused computing resources as it improves the utilization rate and decrease the energy consumption of data center[1]. These computing resources are managed by third party owner i.e cloud owner. The computing resources in cloud are abstracted in a way that no one knows where it resides. User rent the cloud and specify their hardware requirements in order to access the resource. The resources can be scaled up and down by suggesting suitable hardware requirements. Once the resources are adjusted, physical hardware utilization is improved and server wastage is minimized. There is ongoing developments of how cloud is used by Amazon EC2, Google in monitoring the usage of computing resources. They provide various services such as data storage, authentication and generates billing on the basis of usage. Due to rapid use of internet and thus increasing user demands; the cost of storage, computing power has increased. So, the use of resource pool improves the utilization rate and thus resources can be allocated among users more flexibly and efficiently. In Resource pooling, the computing resources such as Cores, RAM, Storage, Disk, etc are pooled up together in a cluster and dynamically allocated to users based on their needs. Assignment of distinct IP in the same subnet is required such that there is no IP conflict in pooling of resources. These resources are passed on to the network and enlisted in the cluster. The resource pooling mechanism of giant cloud such as Amazon, Google is provided with a cost-effective optimization approach such that it minimizes cost and improves utilization rate. It also maintains a local cache that keeps track of unused computing resources[2]. They have own resource allocation policies and tariff charges.

There are many opensource private clouds like Openstack , Eucalyptus, Open-Nebula, Cloudstack etc. For opensource cloud, resource scheduling using resource pooling in openstack is done earlier. It makes use of filtering and weighing in an offline manner thus reducing the time and pools are updated on an event basis[3]. Since the complexity of private cloud deployment is high, these opensource clouds require some tools that helps in creation of resource pool as a key step in deployment. Openstack with MAAS, Openstack with Anvil, Openstack with Devstack, Openstack with packstack, Openstack with Mirantis Fuel are some of the generic openstack tools. All openstack tools are opensource and supports multinode except Devstack and Anvil.[4].

In this paper, we choose openstack with MAAS and Juju since it is stable in deployment and supports multi-node and can be employed for production use. The supported operating system is Ubuntu. It can grow up easily by adding more computing resources. The contribution of this paper includes how the resources are pooled up in a MAAS server cluster using openstack tool MAAS and Juju and how these resources are assigned to different machines in order to have specific openstack services. These services are deployed manually by developers. These openstack service provides a feature in cloud such that each openstack service needs computing resources from the MAAS ubuntu server cluster. Authentication , Metering, Billing etc are some the features which can provision the users when they use these resources.
The rest of the paper is organized as follows- Section 2 gives a background of resource pooling. Section 3 gives an overview of MAAS (Metal as a Service), Juju and openstack services. Section 4 provides a pooling configuration and methodology of MAAS ubuntu-server cluster where we showcase the pooling of computing resources and assignment of openstack services to machines from MAAS ubuntu-server. Section 5 discusses the results. Finally, paper is concluded in Section 6.

II. RESOURCE POOLING

The concept of resource pooling in cloud computing is taken from the re-evolution of internet in a way to spread load across various paths in a way to increase the utilization rate of servers. Resource pooling is nothing but a collection of computing resources such as number of cores, RAM, disk, storage treated like a single and shared resource pool which is robust against failure and achieve flexibility at optimal cost[5]. Besides resource pooling, there is also a model named as anti-resource pooling(resource fragmentation) which focuses on enhanced delivery of services at high cost. But resource pooling focuses on unused delivery of services at low cost[6]. These are some criteria's whether to use resource pooling or not. Another criteria of resource pooling is division of resources based on user preferences. This can be treated in monitoring the usage of computing resources and charges on pay-per-use. In wireless technologies, resource pooling is useful for wireless devices where radio channels are pooled up and provide a more robust channel without conflict of other channels[5]. In cloud, resource pooling serve the multi-tenant customers each demanding a substantial resource allocation using a multi-tenant model. This model is helpful for SAAS(Software as a Service) application where it runs in a centralized environment and provide shared resources at operational cost. The advantage of multi-tenancy is that resources are shared among multi-tenants by dividing hardware, software resources and thus it reduces the operational power cost[7]. In private cloud deployment, it is created over the private network where computing resources are transferred to a pool. This is done by adding a range of IP address in the interface. When machine boots up, it automatically recognizes the IP address of machines and aggregate the resources in a pool. MAC address is listed with a unique machine name alongside computing resources. Initially, the machines for which the resources are to be pooled up are recognized and after that the machines are commissioned and deployed to obtain a pool of resources. In parallel, as many machines can transfer the resources to create a resource pool.

III. OPENSTACK TOOLS AND SERVICES

A. MAAS(Metal as a service)

MAAS is an openstack tool that helps in the management of physical infrastructure with same ease and flexibility as virtual machines in cloud[9]. MAAS provides a customized web and command line interface in which physical machines can be added, commissioned, deployed and managed[12]. Any machine can be power off and power on depending on the need[8]. The role of MAAS comes when new machine boots up, it supplies all the information required and provides ubuntu image to install. Once the machines are added, power on the machine, update the boot image and power off the machine whenever changes are required.
The key components of MAAS software are[9]-

- Region Controller
- Cluster Controller
- Nodes

For small setups , Region controller and Cluster controller are installed on the same server and for large setups, it is recommended to have multiple cluster controller to manage different nodes as shown in Figure 1[12].

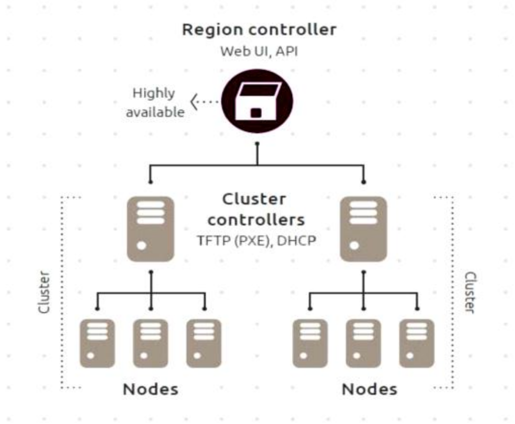


Figure 1 : MAAS Architecture [12]

B. JUJU

Juju allow the users to manage configuration, deploy the machines and maintain resources with ease and provision cloud services in a rapid scale. Juju extends its services to work on public clouds, private clouds and physical servers also[4]. Bootstrapping Juju machine is a key step used to deploy the services and thus take ownership of the machine. It power up the machine so that services are easily configured and deployed. For every OpenStack service components, a yaml configuration file is required to communicate with the cloud providers using MAAS key. The advantage of using MAAS with Juju is ease of deployment. It also adds a benefit of deploying, redeploying, deleting or termination of machine at any time .Automatic detection of the machines in a MAAS cluster is sometimes a big overhead of developers from where to start and managed them manually and thus it becomes annoying due to extensive machine events [4].

C. Openstack Services

When the resources are pooled up in MAAS ubuntu server-cluster, in order to partion the resources or to launch VM, different openstack services are deployed on different machines commissioned in the cluster pool. These services provide specific features to use the application service in cloud more efficiently. Once there is a need of more services, a feature is added in the application. The advantage of openstack services are that it reduces the complexity of managing the hardware requirements whenever you need it.

Openstack have different type of services[11] and their functionality are defined as follows-

- ***Keystone*** – Openstack Identity service (Keystone) provides Authorization and Authentication for users and also manage service catalogs
- ***Glance***- Openstack Image Storage Service (Glance) stores and manages virtual machine images in different format
- ***Neutron*** – Openstack Network Service (Neutron) enables network connectivity to interface devices. It enable users to create and attach interface to networks
- ***Cinder*** – Openstack Block Storage Service (Cinder) provides block storage to guest virtual machines for expanded storage and better performance.
- ***Swift***- Openstack Object Storage Service (Swift) provides a cost-effective, scalable storage platform used for backup.
- ***Ceilometer*** – Openstack Metering/Monitoring Service (Ceilometer) is used for billing, benchmarking and for statistics gathering.

IV. POOLING CONFIGURATION AND METHODOLOGY

The pooling configuration is carried out on a machine having MAAS cluster/region controller as shown in section B. Other machines are - 1 machine for Juju Bootstrap and 4 machines for openstack services. This can be done in step-by-step manner which includes preparing machines to create a resource pool, a key step in private cloud deployment. The machines are Intel-core i3-2130 cpu:3.40 GHZ consisting of 4GB RAM, 500GB Hard-disk, etc. Each machine supporting openstack service assignment must be in private network as MAAS supports DHCP and DNS management. Few IP address should be reserved as MAAS requires floating IP for

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, November 2017

allocation [13]. The workflow of configuration method is shown below in Figure 2

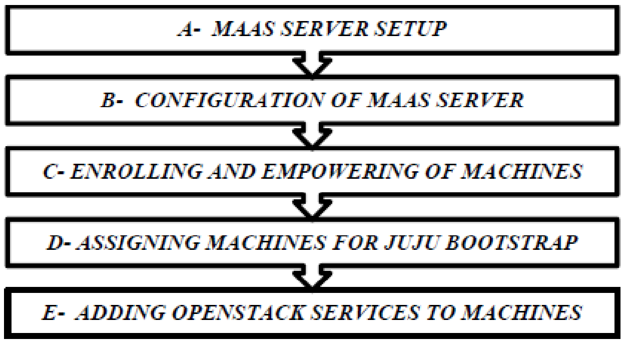

A- MAAS SERVER SETUP
B- CONFIGURATION OF MAAS SERVER
C- ENROLLING AND EMPOWERING OF MACHINES
D- ASSIGNING MACHINES FOR JUJU BOOTSTRAP
E- ADDING OPENSTACK SERVICES TO MACHINES

Figure 2 : Step-by-step workflow of Resource pool Setup

A. MAAS SERVER SETUP

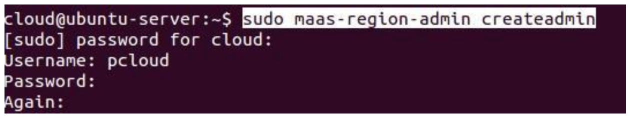
First, we create an environment for MAAS setup by ensuring that machines are in private network. Install ubuntu server 14.04 LTS on the machine for MAAS[10]. Then, ppa repositories are added. Finally MAAS is installed on ubuntu server desktop. Once the MAAS server is deployed, it can be accessed using configured IP address. It is a customized interface which requires username and password to login it.

Install ubuntu server 14.04 LTS on the machine for MAAS

$sudo add-apt-repository ppa:juju/stable
$sudoadd-apt-repositoryppa:maas-maintainers/stable
$sudoadd-apt-repositoryppa:cloud-installer/stable
$sudo apt-get update
$sudo apt-get install maas

Install desktop on ubuntu server : $sudo apt-get install ubuntu-desktop , $sudo taskel

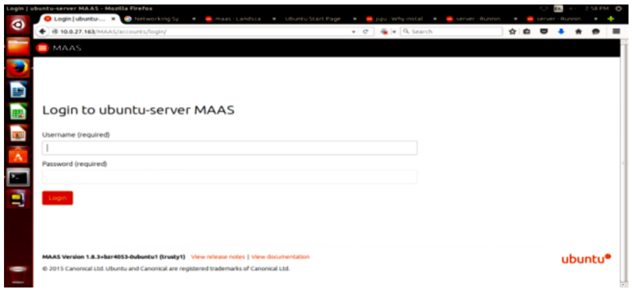
Create username and password to access MAAS UI: $sudo maas-region-admin createadmin as shown in Figure 3



```
cloud@ubuntu-server:~$ sudo maas-region-admin createadmin
[sudo] password for cloud:
Username: pcloud
Password:
Again:
```

Figure 3 :MAAS Server Username and Password

We can login to MAAS server using username and password as shown in Figure 4



Login | ubuntu-server MAAS - Mozilla Firefox

10.0.27.163/MAAS/accounts/login/

MAAS

Login to ubuntu-server MAAS

Username (required)

Password (required)

Login

MAAS Version 1.8.3+bzr4053-0ubuntu1 (trusty1) View release notes | View documentation

© 2015 Canonical Ltd. Ubuntu and Canonical are registered trademarks of Canonical Ltd.

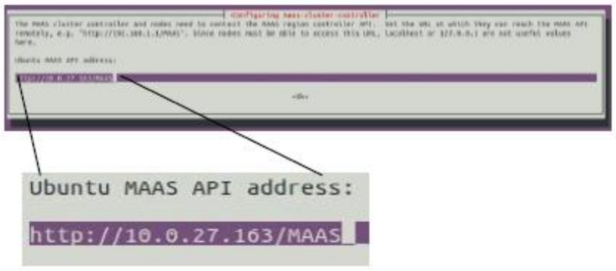
Figure 4 : MAAS Server login

B. CONFIGURATION OF MAAS SERVER

After login to MAAS server, import boot images for Ubuntu 14.04 LTS. Initially, a cluster controller has boot images but a region controller doesnot. So machines will not be provisioned until the boot images are imported into the region controller. Install cluster controller for MAAS as each cluster needs a controller node:

$sudo apt-get install maas-cluster-controller
$sudo apt-get install maas-dhcp maas-dns
$sudo apt-get install maas-dhcp maas-dns

Once the cluster software is installed, it is recommended to run : $sudo dpkg-reconfigure maas-cluster- controller as shown in Figure 5.



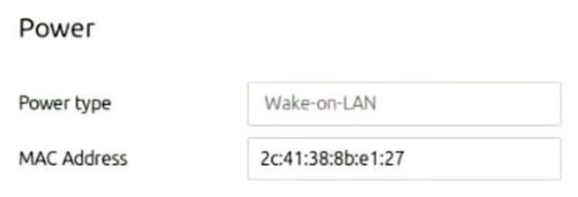
```
Ubuntu MAAS API address:

http://10.0.27.163/MAAS
```

Figure 5 : MAAS controller IP

This makes sure that the cluster controller agent is pointed at the correct address for the MAAS master controller. After that, MAAS automatically recognizes the network interface on the cluster controller. Then, we edit cluster interface and select DHCP and DNS. Lastly, IP is configured by providing subnet mask, broadcast IP, router IP, dynamic range of IP such that IP is assigned dynamically to machines during PXE boot.

C. ENROLLING AND EMPOWERING OF MACHINES

When boot from PXE image, MAAS server stores the information of machines including MAC address, architecture (e.g amd64/generic) and other details. As a result, machines are enrolled in a cluster but does not come in ready state. The machines can come in ready state only after powering on the machines. The machines are powered on by filling power type (e.g wake on LAN) in MAAS as shown in Figure 6 and once machines are completely powered, the machines come in ready state. It can be powered off also when not machines are not in use. In parallel, the machine events such as machine change status from new to ready, powering on/off machines are continuously saved in the cluster to keep track of the machine status.



Power

Power type	Wake-on-LAN
MAC Address	2c:41:38:8b:e1:27

Figure 6 : Power-type for empowering machines

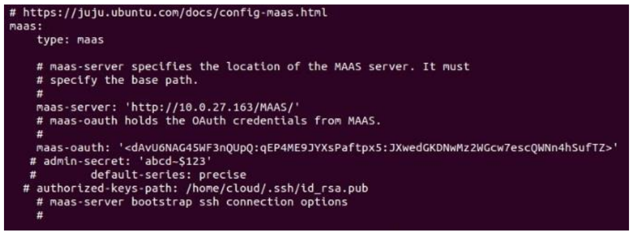
D. ASSIGNING MACHINES FOR JUJU BOOTSTRAP

We power on the machine on which juju bootstrap will take place. This will change the machine state from ready->allocated->deploying->deployed and once it is deployed, it becomes the owner of admin. Then, Juju bootstrapping is done to deploy some openstack services to the machine and the same deployed machines should not be used for assigning any other openstack services. Before continuing bootstrap, atleast one machine is listed in the MAAS cluster. Finally, Juju environment is configured for assignment and deployment of openstack services.

Juju environment is configured by adding MAAS server IP address and MAAS API key as shown in Figure 7.

From the MAAS server account tab, we generate the MAAS API key for each juju environment

sudo nano ~/.juju/environments.yaml



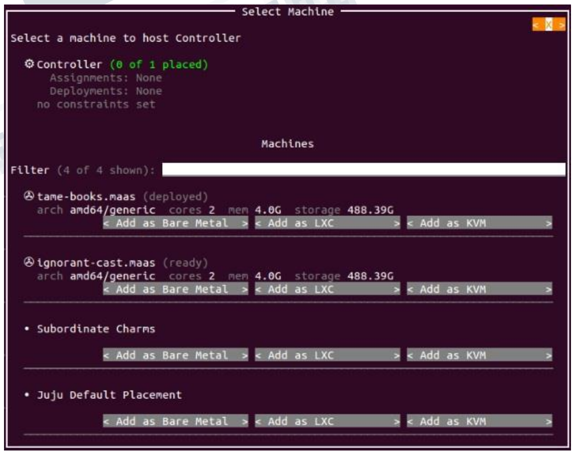
```
# https://juju.ubuntu.com/docs/config-maas.html
maas:
    type: maas

    # maas-server specifies the location of the MAAS server. It must
    # specify the base path.
    #
    maas-server: 'http://10.0.27.163/MAAS/'
    # maas-oauth holds the OAuth credentials from MAAS.
    #
    maas-oauth: '<dAvU6NAG45WF3nQUpQ:qEP4ME9JYXsPaftpx5:JXwedGKDNwMz2WGcw7escQWNn4hSufTZ>'
   # admin-secret: 'abcd~$123'
   #        default-series: precise
  # authorized-keys-path: /home/cloud/.ssh/id_rsa.pub
    # maas-server bootstrap ssh connection options
    #
```

Figure 7 : .yaml configuration of MAAS sever

E. ADDING OPENSTACK SERVICES TO MACHINES

In order to distribute load of services, openstack uses different machines to deploy some complex services like compute, controller, keystone, neutron, ceilometers, dashboard, glance etc to provide a feature to the cloud.

$sudo openstack-status displays the screen in which openstack services will be added to machines. Select machines and assign services based on : Add as Bare Metal , Add as LXC , Add as KVM as shown in Figure 8. Required services should be assigned while additional services is optional. The conflict arises when any constraint of service is assigned to two different machines.

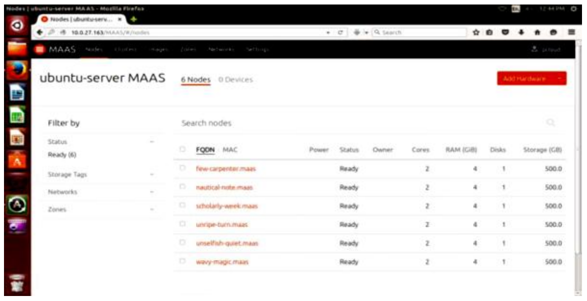


```
                         Select Machine
Select a machine to host Controller

  ⚙Controller (0 of 1 placed)
     Assignments: None
     Deployments: None
    no constraints set


                            Machines
Filter (4 of 4 shown):

  ⊛tame-books.maas (deployed)
    arch amd64/generic  cores 2  mem 4.0G  storage 488.39G
             < Add as Bare Metal > < Add as LXC          > < Add as KVM          >

  ⊛ignorant-cast.maas (ready)
    arch amd64/generic  cores 2  mem 4.0G  storage 488.39G
             < Add as Bare Metal > < Add as LXC          > < Add as KVM          >

  • Subordinate Charms

             < Add as Bare Metal > < Add as LXC          > < Add as KVM          >

  • Juju Default Placement

             < Add as Bare Metal > < Add as LXC          > < Add as KVM          >
```

Figure 8 : Machine selection for openstack services

V. RESULTS AND DISCUSSION

In this section, we showcase the resource pool which lists number of cores, RAM, storage, disk and the assignment of openstack services to machines is done on enrolled and empowered machines in the cluster. Initially, all the machines are in new state when boot from PXE and when select the machine in empowering mode, they all come in commissioning and then ready state listing each machine Cores, Disk, RAM, Storage etc.



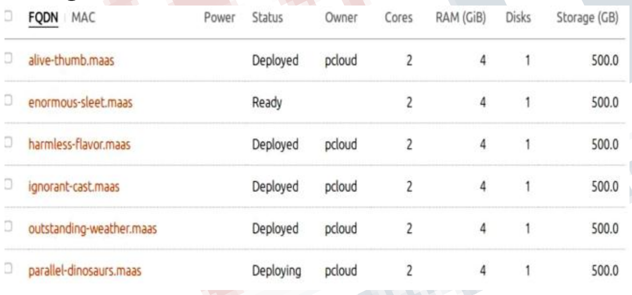
ubuntu-server MAAS 6 Nodes 0 Devices

Filter by
Status
Ready (6)
Storage Tags
Networks
Zones

FQDN MAC	Power	Status	Owner	Cores	RAM (GiB)	Disks	Storage (GB)
few-carpenter.maas		Ready		2	4	1	500.0
nautical-note.maas		Ready		2	4	1	500.0
scholarly-week.maas		Ready		2	4	1	500.0
unripe-turn.maas		Ready		2	4	1	500.0
unselfish-quiet.maas		Ready		2	4	1	500.0
wavy-magic.maas		Ready		2	4	1	500.0

Figure 9 : Resource Pool in MAAS ubuntu-server

This creates a resource pool of 6 machines as shown in Figure 9 which are used for assigning openstack services. The resource pool is scalable and any machine can be added or terminated when required. For termination, first the machines are powered off such that machines come in new state and then it can be terminated and deleted from the MAAS cluster. This resource pool can also handle concurrent machine boot-up as the machine detection is dependent on the corresponding IP Address of the machines that are already listed in the MAAS Network Configuration Tab



FQDN MAC	Power	Status	Owner	Cores	RAM (GiB)	Disks	Storage (GB)
alive-thumb.maas		Deployed	pcloud	2	4	1	500.0
enormous-sleet.maas		Ready		2	4	1	500.0
harmless-flavor.maas		Deployed	pcloud	2	4	1	500.0
ignorant-cast.maas		Deployed	pcloud	2	4	1	500.0
outstanding-weather.maas		Deployed	pcloud	2	4	1	500.0
parallel-dinosaurs.maas		Deploying	pcloud	2	4	1	500.0

Figure 10 : Resource Pool of Deploying/ Deployed machines using Juju Bootstrapping

If there is no IP address of machine, machine cannot be Boot-up for enrolling in the cluster. From Figure 9, the ready state corresponds to the empowered state without ownership. The significance of owner is that machines are assigned openstack services and then deployment is done on the basis of empowering machine. This can be done by Juju Bootstrapping in which the configuration is done such that the machine gets connected to the authorized MAAS server.

Also, the yaml configuration specify the different connections such as Amazon EC2, Azure etc that can also be configured based on machine requirements. When juju environment is configured and juju bootstrapping is done, machines come in deployed state as shown in Figure 10. Once all the machines come in deployed state, we can check the openstack status using sudo openstack-status It shows that openstack services are added first by choosing machines either manually or machines are autoplaced that are enrolled in resource pool and further these machines can be deployed for assigning the service to the machine.

Table 1 : Openstack service assignment to machines

Sno	Openstack Service	Machine Name
1	Compute	Ignorant-cast.maas
2	Glance	Outstanding-weather. maas
3	Keystone, Openstack Dashboard, MYSQL	Harmless-flavor.maas
4	Controller	Alive-thumbs.maas

From Table 1, openstack services like compute, controller, glance, keystone, etc are successfully assigned to the enrolled and empowered machines in the cluster as shown in Figure 9.

Openstack service assignment should be done based on hypervisors such as LXC, Bare-Metal, KVM etc. The number of machines configured for service assignment are not limited such that machines of higher RAM, Disk, Storage, etc can also have one or more openstack services deployed on it. This shows that cluster controller and region controller can also be in two different machines.

VI. CONCLUSION AND FUTURE WORK

In this paper, we discussed how to deploy a resource pool using MAAS and Juju. The deployed resource pool focuses on empowering the machines using PXE boot and enabling wake-on-lan power-type. It is found that computing resources of six machines such as cores, RAM, disk, storage

are successfully pooled up in the MAAS server cluster. Furthermore, we discussed how to deploy the machines using juju bootstrapping so that machines becomes the owner of cloud.

Finally, we discussed how to configure juju environment for assignment of openstack services among enrolled machines. These services are either added by developers or machines are autoplaced randomly. These services are

successfully added and assigned to machines but deployment of service is not done yet due to unavailability of large cluster.
In future, we aim to extend our work in deploying the openstack services for enrolled machines in MAAS server. The challenge in deploying openstack service is due to juju bootstrapping which requires a cluster of as many machines in which cluster controller and region controller are on different machines.The load of openstack components must be distributed to the machines in cluster so as to be more manageable. It can be deployed for large production use only by scaling up the machines that can manage load on server for assignment of complex services.

VII. ACKNOWLEDGEMENT

We would like to thank Dr. Parvez Mahmood Khan, Director, Computer Center, Aligarh Muslim University for his precious suggestions and valuable support to keep up with current trends of cloud in this work.

VIII. REFERENCES

[1] Shuai Zhang, Shufen Zhang, Xuebin Chen, Xiuzhen Huo,"Cloud Computing Research and Development Trend", ICFN 2010 Second International Conference on Future Networks, Sanya, Hainan, China, pp. 93-97, January 2010

[2] Rajkumar Buyya, James Broberg, Cloud Computing: Principles and paradigms, 1st Edition, John Wiley & Sons Inc., Hobokan, New Jersey, February 2011

[3] Shalmali S.Sahasrabudhe, Shilpa S. Sonawani , "Resource scheduling using Resource pool in openstack", International Journal of Computer Engineering and Technology, Vol. 5, Issue 6, pp. 118-123, June 2014

[4] Anshu Awasthi, Ravi Gupta, "Comparison of Openstack Installers", IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 9, pp. 744-748, September 2015

[5] Damon Wischik, Mark Handley, Marcelo Bagnulo Braun, "The Resource Pooling Principle", ACM SIGCOMM Computer Communication Review, Vol. 38, Issue 5, pp. 47-52, October 2008

[6] Junaid Qadir, Arjuna Sathiaseelan, Liang Wang, Jon Crowcraft, "Resource Pooling for Wireless Networks: Solutions for the Developing World", ACM SIGCOMM Computer Communication Review, Vol. 46, Issue 4, pp. 30-35, October 2016

[7] M Saraswathi, T Bhuvaneswari, "Multi-tenancy in cloud software as a service application", International Journal of Advanced Research and Computer Engineering, Vol. 3, Issue 11, pp. 159-162, November 2013

[8]MAASwebpage-1https://www.ubuntu.com/cloud/maas (Retrieved on 15 May 2017)

[9]MAASwebpage2:https://www.ubuntu.com/download/cloud/install-openstack-with-autopilot (Retrieved on 1 May 2017)

[10] Cloud Infrastructure using Ubuntu 14.04 LTS MAAS Webpage https://williamlsd.wordpress.com/2014/04/26/installing-local-cloud-infrastructure-using-ubuntu-14-04-lts-maas/(Retrieved on 20 May 2017)

[11] An Introduction to openstack project – devops and cloudcomputing:http://blog.flux7.com/blogs/openstack/openstacktutorialspart1openstack introduction (Retrieved on 18 May)

[12] MAAS Top 10 questions Webpge https://insights.ubuntu.com/wp-content/uploads/226b/Top-10-Questions-about-MAAS.pdf (Retrieved on 20 April 2017)

[13] MAAS network layouts https://insights.ubuntu.com /2015/04/10/maas -network-layouts-for-the-landscape-autopilot (Retrieved on 2 May 2017)