Installation Runbook for Appcito ADS

Application Version	Release 1.11
MOS Version	Mirantis
OpenStack Version	Kilo
Application Type	*Appcito Application Delivery System(ADS)*

Content

Document History
1 Introduction
    1.1 Target Audience
2 Application overview
3 Joint reference architecture
4 Physical & Logical Network topology
5 Installation & Configuration
    5.1 Overview of MOS installation steps
    5.2 MOS installation in details
        5.2.1 Creation of OpenStack environment
        5.2.2 MOS Deployment
        5.2.3 Health Check Results

    5.3 Appcito CAFÉ installation steps
        5.3.1 Workstation Installation
        5.3.2 Appcito CSP Installation
        5.3.2 Appcito CSC Installation

    5.4 Testing
        5.4.1 Target use case(s)
        5.4.2 Test cases
        5.4.2 Test Tools
        5.4.3 Test Results

Document History

Version	Revision Date	Description
0.1	**13-10-2015**	**Appcito Initial Draft**
1.0	**04-11-2015**	**Modified with more description**
1.1	**16-12-2015**	**Modified with Mirantis Kilo Deployment**
1.5	**06-01-2016**	**Modified based on Mirantis feedback**
1.6	**21-01-2016**	**Expanded the section 5.3.4, Step-02**

1 Introduction

This document is to serve as a detailed Deployment Guide for Appcito Application Delivery System (ADS). Appcito offers ADS with Cloud Services Controller (CSC) for Management & Control Plane and Cloud Services Proxy (CSP) for Data Plane, for Application L4-L7 proxy, Security, Acceleration & Analytics. This document describes the reference architecture, installation steps for certified MOS + Appcito ADS and testing procedures.

1.1 Target Audience

This is meant for the organizational members i.e. Devops, System/Network Administrator etc, who is responsible for running the Application on Mirantis Openstack and looking for services like Application Load Balancing & Proxy functions, Application security and Application Analytics. Appcito ADS transforms the process of deploying and operating cloud applications, making it easy to deliver superior performance, security and agility.

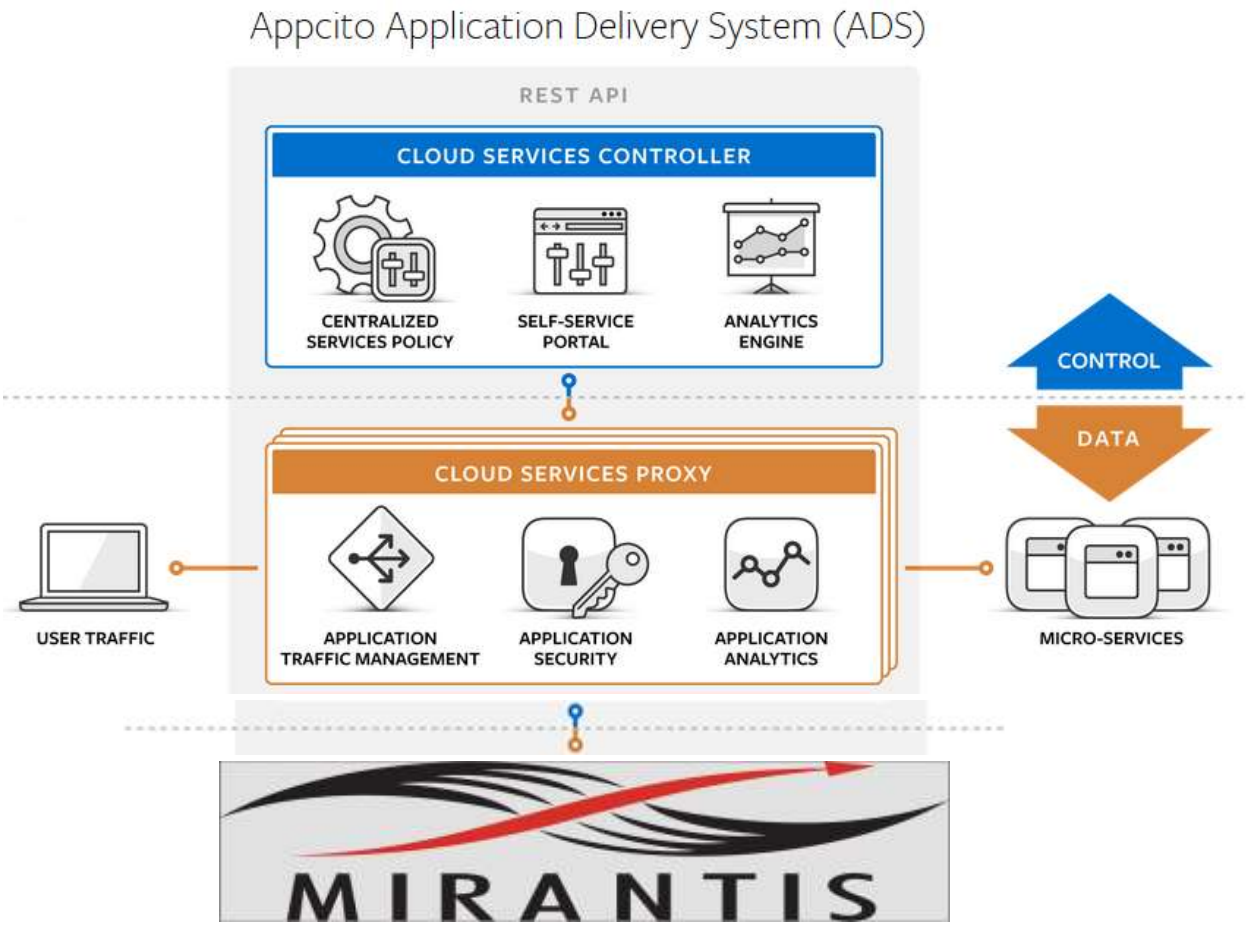
2 Application overview

For organizations embracing cloud computing, Appcito ADS transforms the process of deploying and operating cloud applications, making it easy to deliver superior performance, security and agility.

Appcito ADS is a cloud-native service that makes applications secure, performance and available, without any changes to application source code.
Appcito ADS employs a software-defined, web-scale application delivery architecture and runs comprehensive application delivery including Load Balancing, Application Security and Application analytics

Appcito ADS has two core architectural components. Appcito Cloud Services Proxies (CSP) operate in data plane and are responsible for enforcing application delivery policies. Residing in control plane, Appcito Cloud Services Controller (CSC) is a hyper-scale centralized application controllers responsible for management and control of application delivery policies, and processing of application analytics. Together CSPs and CSC work in concert with your application

3 Joint Reference Architecture



Appcito's Application Delivery System (ADS) brings together application load balancing, traffic engineering, application security and analytics capabilities in an elegantly integrated packaged format, where they can all be consumed together, or as add-on components. This provides for a highly flexible offering that is easy to try, deploy and scale in an extremely agile manner

Cloud Services Controller (CSC)

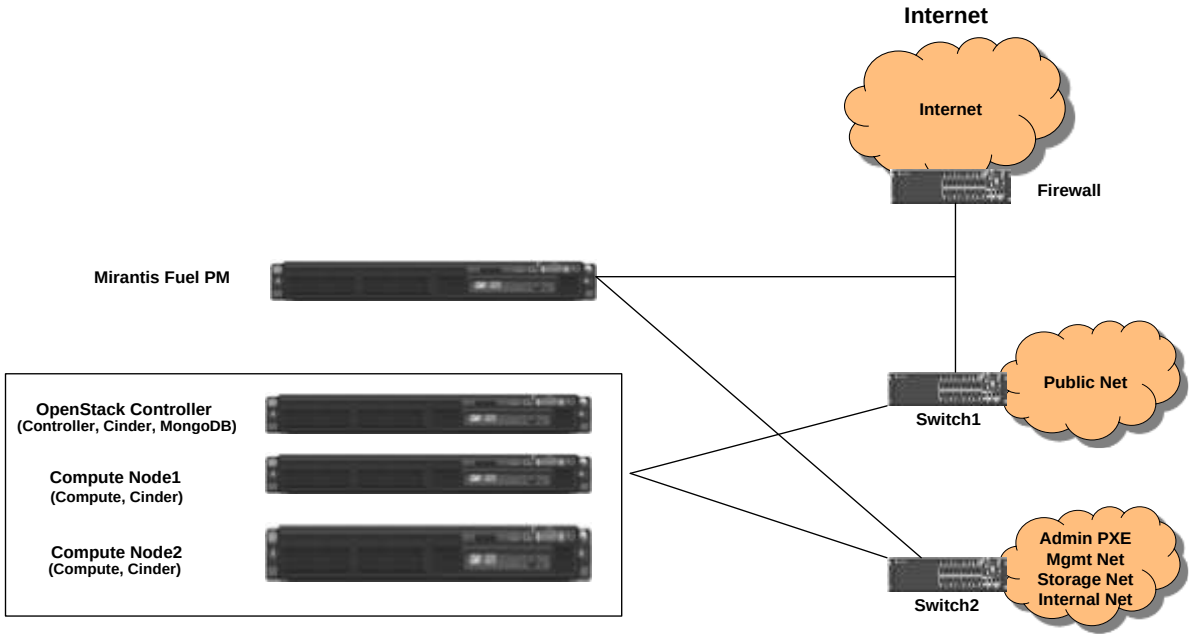
Provides centralized management, control and a big data Analytics for any application across any cloud with an easy way to configure, manage and monitor applications.

Cloud Services Proxy (CSP)

A lightweight, full proxy that front-ends cloud application and executes policies. It can reside in geographically-dispersed public clouds such as AWS, Azure or OpenStack-based private clouds. It is quick to deploy, easy to configure and requires no changes to existing application stack or micro-services components.

4 Physical & Logical Network Topology

5 Installation & Configuration



In this setup, the floating IPs are private IP address range 172.16.14..x and in order for launched instances to access the system, these floating IPs need to be mapped to a public IP address in the Firewall. We have indicated PFSense as the firewall for this function.

5.1 Overview of MOS installation steps

High-level description of MOS installation steps -:

- Total nodes quantity that should be used :
    - 1 - the Fuel Master node,
    - 1 - MOS Controller node,
    - 2 - MOS compute nodes
- Download Mirantis v7.0 ISO from Mirantis website.
- Follow the installation instructions as specified in Mirantis OpenStack User Guide.
- Create a tenant
- Allocate resources to the tenant

5.2 MOS installation in details

5.2.1 Creation of OpenStack environment

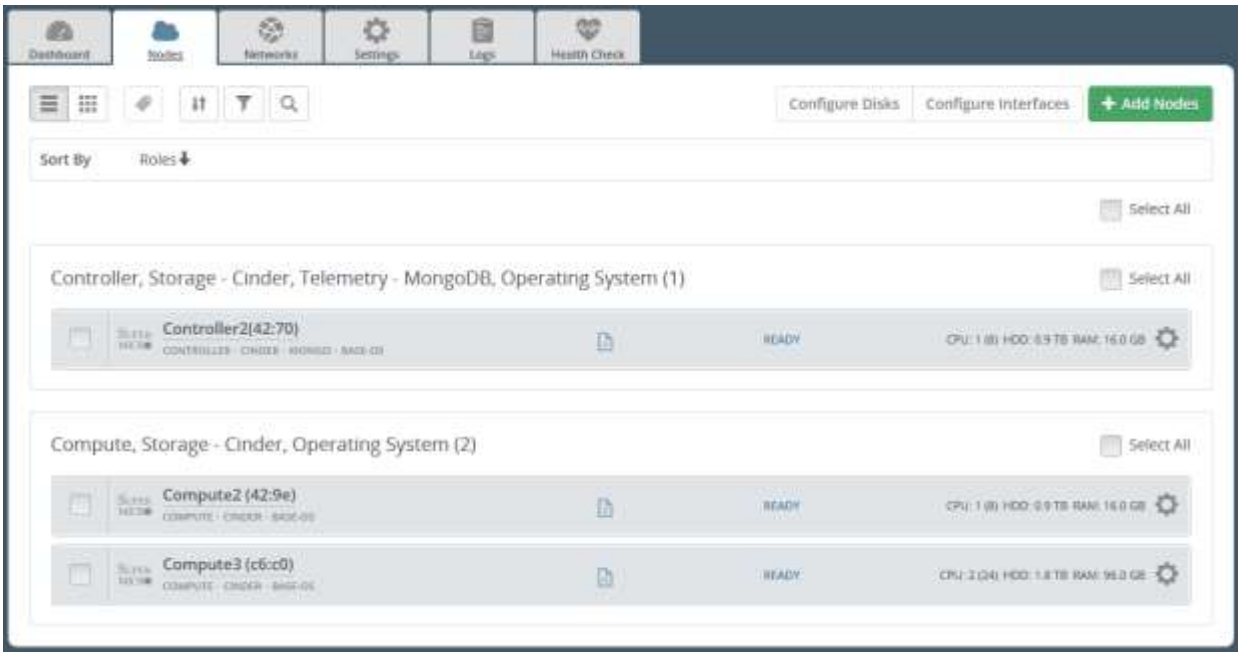
We performed the default environment creation, as per Mirantis OpenStack documentation (Create a new OpenStack environment section).

Additional services:

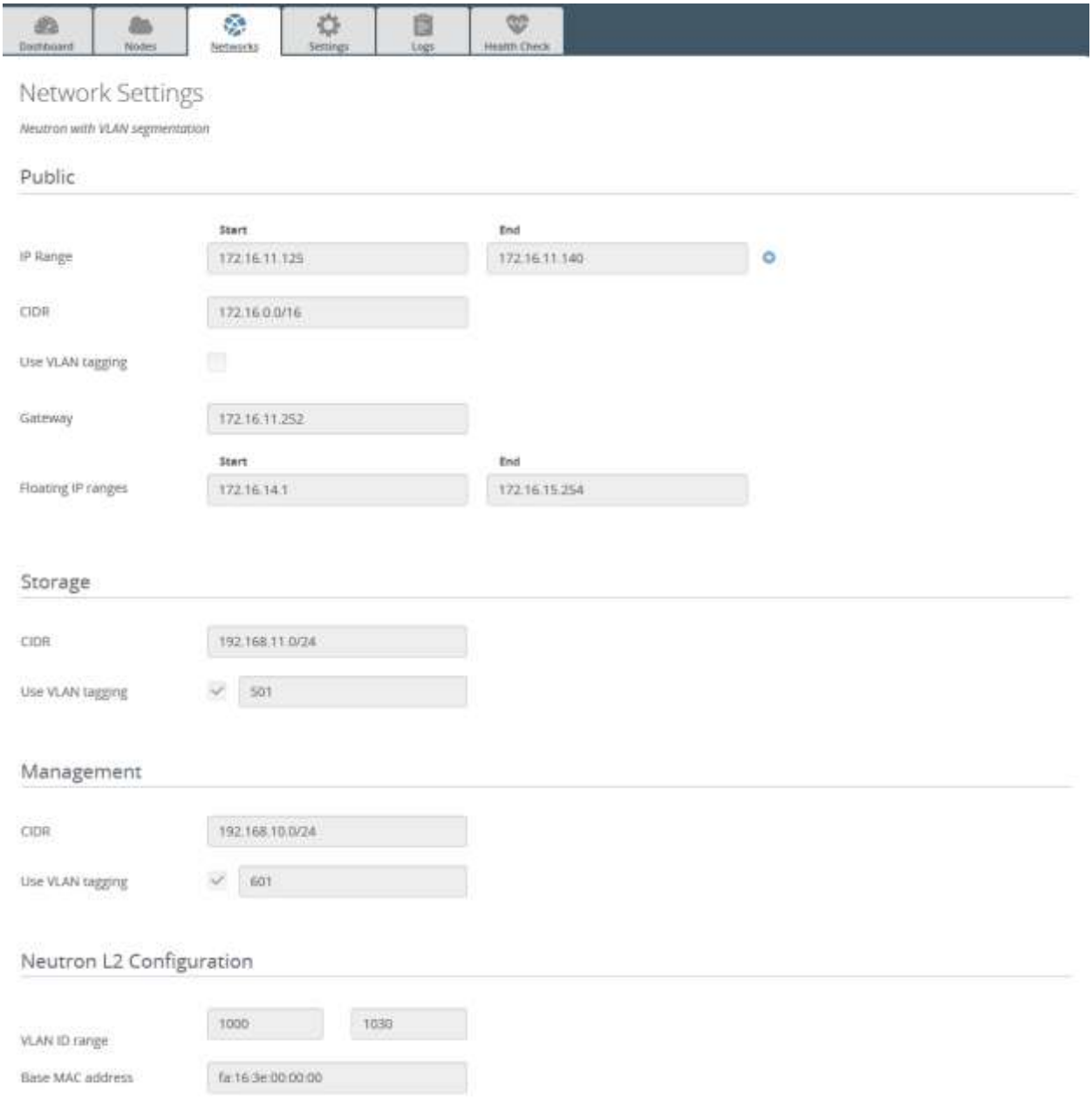
Ceilometer

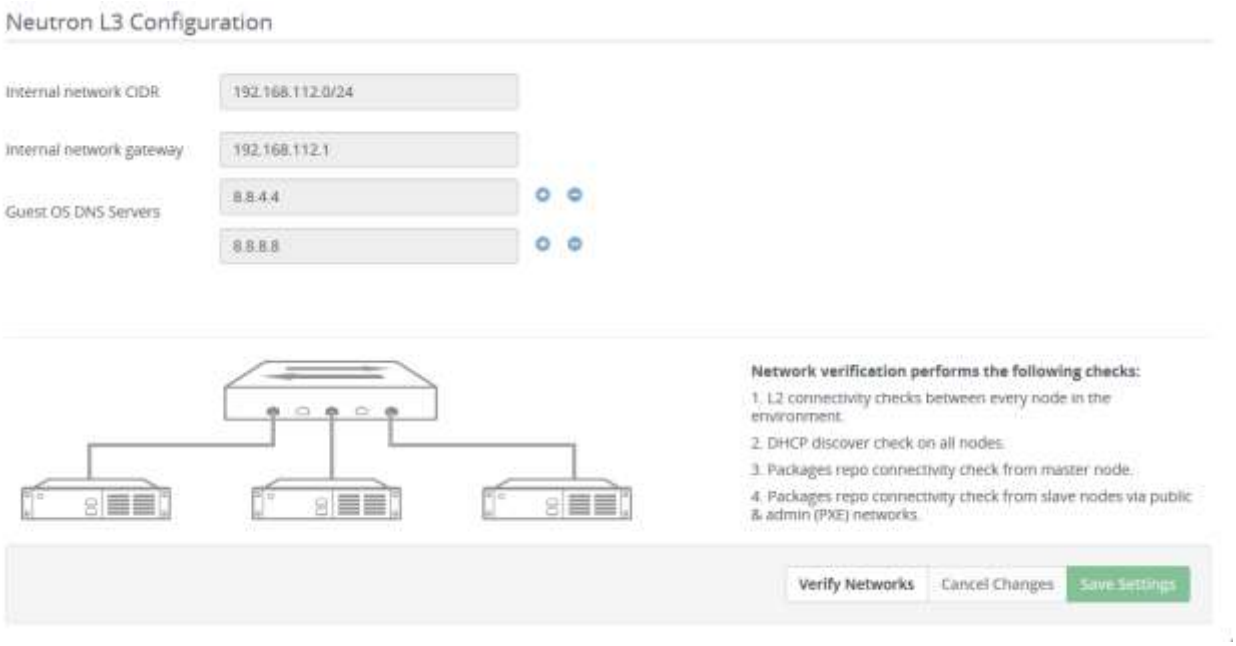
5.2.2 MOS Deployment

Roles and hardware configuration for the Nodes:



Networking and interface configurations:





Specify minimum combination of Mirantis OpenStack deployment options.

OS	Mode	HV	Network	Storage
			VLAN	Cinder
Ubuntu	Non HA	KVM	yes	yes

5.2.3 Health Check Results

OpenStack Health Check

Select All | Provide credentials | Run Tests

Sanity tests. Duration 30 sec - 2 min	Expected Duration	Actual Duration	Status
Ceilometer test to list meters, alarms and resources	180 s.	2.2	✔
Request flavor list	20 s.	0.2	✔
Request image list using Nova	20 s.	0.2	✔
Request instance list	20 s.	0.3	✔
Request instance list	20 s.	0.3	✔
Request absolute limits list	20 s.	0.1	✔
Request snapshot list	20 s.	0.3	✔
Request volume list	20 s.	0.1	✔
Request image list using Glance v1	10 s.	0.0	✔
Request image list using Glance v2	10 s.	0.0	✔
Request stack list	20 s.	0.0	✔
Request active services list	20 s.	0.2	✔

Functional tests. Duration 3 min - 14 min	Expected Duration	Actual Duration	Status
Create instance flavor	30 s.	0.3	✔
Check create, update and delete image actions using Glance v1	130 s.	2.7	✔
Check create, update and delete image actions using Glance v2	70 s.	2.8	✔
Create volume and boot instance from it	350 s.	37.6	✔
Create volume and attach it to instance	350 s.	57.5	✔
Check network connectivity from instance via floating IP	300 s.	30.2	✔
Create keypair	25 s.	0.7	✔
Create security group	25 s.	0.5	✔
Check network parameters	50 s.	0.2	✔
Launch instance	200 s.	22.3	✔

Launch instance with file injection	200 s.	56.4	✔
Launch instance, create snapshot, launch instance from snapshot	300 s.	48.3	✔
Create user and authenticate with it to Horizon	80 s.	0.6	✔

Platform services functional tests. Duration 3 min - 60 min	Expected Duration	Actual Duration	Status
Ceilometer test to check alarm state and get Nova metrics	60 s.	100.6	✔
Ceilometer test to check notifications from Glance	5 s.	2.5	✔
Ceilometer test to check notifications from Keystone	5 s.	2.6	✔
Ceilometer test to check notifications from Neutron	40 s.	3.0	✔
Ceilometer test to check notifications from Cinder	10 s.	3.6	✔
Ceilometer test to create, check and list samples	5 s.	13.6	✔
Ceilometer test to create, update, check and delete alarm	120 s.	103.1	✔
Typical stack actions: create, delete, show details, etc.	560 s.	36.8	✔
Advanced stack actions: suspend, resume and check	660 s.	66.5	✔
Check stack autoscaling	2200 s.	245.4	✔
Check stack rollback	310 s.	14.3	✔
Update stack actions: inplace, replace and update whole template	950 s.	80.0	✔

Cloud validation tests. Duration 30 sec - 2 min	Expected Duration	Actual Duration	Status
Check disk space outage on controller and compute nodes	20 s.	0.9	✔
Check log rotation configuration on all nodes	20 s.	0.9	✔

5.3 Appcito ADS installation steps

5.3.1 Setting up Pre-requisites

Login to Tenant Portal @ http://172.16.11.126/horizon/auth/login/?next=/horizon/
Create the pre-requisites for the tenant through Horizon dashboard:

Step 01

Access & Security - > Security Group - >Edit default security group to add ingress rule for All

TCP/UDP/ICMP

Step 02

Access & Secuirty - > Key Pairs - > Add the ssh key pair to be used

Step 03

Access & Security - > Floating IP - > Add a minimum of 15 floating IPs

Step 04

Admin - > System - > Flavors - > Create appcito.small with 1 VCPU, 2 GB RAM 20GB Disk and mark it public.

5.3.2 Setting up Workstation

Create a Workstation node on default Cent OS 6.5 which will act as the installation node for the application inside the tenant.

Step 01

Login to Tenant Portal @ http://172.16.11.126/horizon/auth/login/?next=/horizon/

Step 02

On Horizon dashboard

Go to Create An Image
Give the Parameters as Necessary
Give Image Source as Image Location as - http://cloud.centos.org/centos/6.5/images/CentOS-6-x86_64-GenericCloud-20140929_01.qcow2 and Download

Step 03

Create a new instance using the image downloaded in the above step

Go to Launch Instance
Pass the Parameters as Needed
Pass "Boot from Image" to Instance Boot Source
Select the Image Name of the previous step
Finally Click Launch

Once the System is Launched, assign a Floating IP and map it to a Public IP address in the firewall for access from outside.

Step 04

Configure workstation:

Setup the root user password

sudo su -

su root

Set Password for root

Modify the password policy in ssh configuration

vi /etc/ssh/sshd_config

Change PasswordAuthentication no to PasswordAuthentication yes

Make PermitRootLogin as yes

Restart SSH

Login to the System we have launched
Run the curl command as root to get the chef to local machine and install
curl -sSL https://get.rvm.io | bash
(logout and login)
source /usr/local/rvm/scripts/rvm
rvm uninstall 1.8.7
rvm install 2.0.0
rvm use 2.0.0 --default
rvm rubygems current
gem install chef --version 11.8.2
gem install knife openstack
yum install git

Step 05

Verify that you are able to login using the ssh key as root user from outside.

5.3.3 Appcito CSP Image Installation

Step 01

Login to Tenant Portal @ http://172.16.11.126/horizon/auth/login/?next=/horizon/
Compute - > images - > Create Image

Give Name, Description, Select Image location option and Image location with the URL mentioned below, Format as QCOW2, Minimum disk as 40GB, Minimum RAM as 2GB and select

public checkbox and Create Image

URL for Centos 7.0 QCOW2 image
http://cloud.centos.org/centos/7/images/CentOS-7-x86_64-GenericCloud.qcow2

Step 02

Launch the Image created in the Step 01 to prepare for installing Appcito CSP

Compute -> Images -> Launch

Assign a Floating IP to this Newly Launched Instance

Instance -> Associate Floating IP

Login to the Newly Launched Instances and we need to download the Respective CSP Release from S3 and Run install_pep_release.sh

ssh -i ~/<ssh_key> appcito-user@<new_floating_ip>
sudo su –
cd /
mkdir pep_scripts
cd /pep_scripts/
wget <pep tgz s3 url>
Eg: wget https://s3-us-west-2.amazonaws.com/appcito-pep-releases-qa/GA-2014-1.0/pep-Production_rel_1_11.tgz

Get the 'install_pep_release.sh' from internal repository and run the script
./install_pep_release.sh -a pep-Production_rel_1_11.tgz -c openstack -n no

Step 03

Once the above step is complete, Logout and shutdown the machine

Step 04

Create a Snapshot out of this machine and this would be the ready to use release specific CSP image.

Instances - > Create Snapshot

5.3.4 Appcito CSC Deployment

Below would be the steps to be followed to have have Appcito Cloud Services Controller (CSC) deployed in Mirantis OpenStack Deployment

Step 01 – Login to WorkStation

ssh -i <ssh_key> centos@<workstation_ip addess>

sudo su -

Step 02 - Git Clone the Appcito Deployment Repo at your own directory. This is an internal Git repo for the configuration management for deployment. We import this to the current working directory to proceed with the deployment steps

Step 03 – Make Changes as per the following guidelines in the configurations

cd cloudzelera.devops/chef-repo/scripts

There are 3 files which requires modification with the openstack tenant details;

vi openstack_configuration.properties

openstack_tenant_name=<tenant> - this is the tenant name
openstack_user_name=<username> - this is the username
openstack_password=<password> - this is the password
datacenter_name=MyOwn - Leave this without change
account=stage - the AWS account where the source build is made
region=regionOne - Leave this without change
environment=release_1_10 - Enter the environment name for the Pod
plan=dev - Leave this without change
user_data=./set_hostname.sh - Leave this without change
identity_file=infra-setup-or.pem - Leave this without change

vi regionOne.properties

network_id=128619cd-f631-42c4-8bea-80337defa79e - Network ID from the Openstack tenant
instance_size=appcito.small - Leave this without change.
image_id=8a499bce-1573-44e2-940e-38742c251d0d - CSC Image ID from Openstack tenant
ssh_user=root - Leave this without change

ssh_password=appcito - Leave this without change
identity_file=infra-setup-or.pem - Leave this without change

This script determines the components those are selected to be launched. When running multiple times, please ensure the entries are corrected properly. 1 refers the selection and 0 will be for not selecting the component.

vi common_openstack.sh

```
edge_count=${edge_count:-1}
barista_count=${barista_count:-1}
config_datastore_count=${config_datastore_count:-1}
registry_server_count=${registry_server_count:-1}
message_bus_count=${message_bus_count:-1}
ui_count=${ui_count:-1}
metrics_datastore_count=${metrics_datastore_count:-1}
metrics_collector_count=${metrics_collector_count:-1}
rtp_coordinator_count=${rtp_coordinator_count:-1}
rtp_manager_count=${rtp_manager_count:-1}
rtp_worker_count=${rtp_worker_count:-1}
ui_domain=os11cafe.appcito.com - UI domain name for configurations
edge_domain=os11api.appcito.com - API domain name for configurations
```

Step 04 – Run the Launch Script on the background and redirect the logs to a file. This will start the launch process process of 11 Appcito CSC components and configurations.

```
./launch_cluster_openstack.sh > /tmp/<logname> &
```

Step 05 – Ensure the previous steps are completed without errors, before coming to this. The previous steps will ensure all the 11 Appcito CSC components are launched and bootstrapped for chef. Login to each of the 11 Appcito CSC components and run chef-client. This can be done by logging into each newly launched CSC servers and run chef-client

```
ssh root@<floating_ip>
chef-client
```

Step 06 – For the instances launched in the Openstack with 10.10.10.x floating IPs, in order to access from outside, a public IP should be mapped to the corresponding floating IP address on the firewall. UI and Edge server requires mapping, and if there is a need to access any other servers from outside for viewing the administration page, those can be optionally mapped.

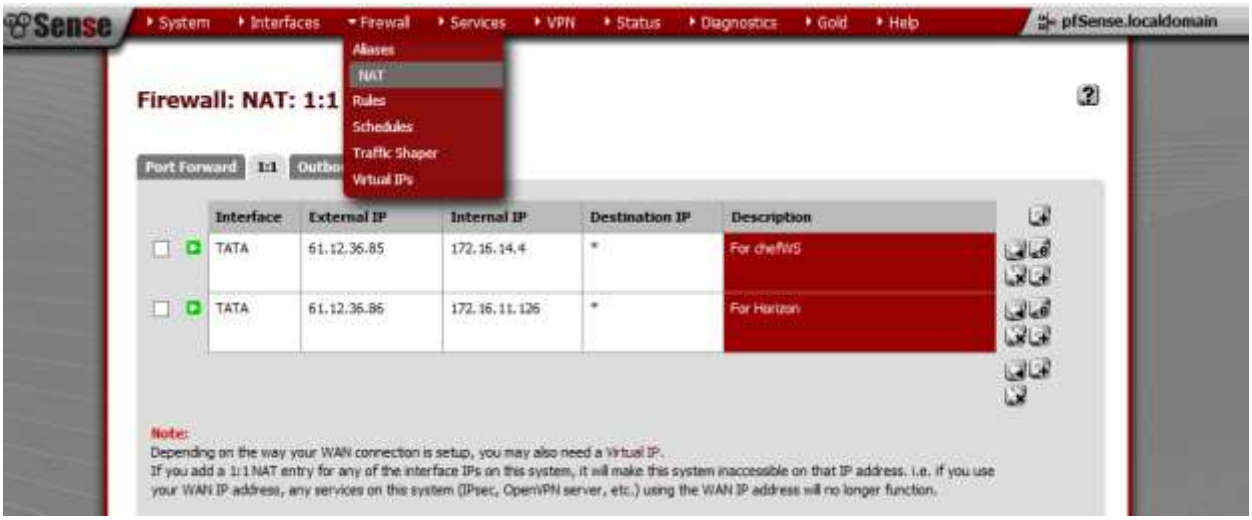
Login into Firewall admin console and have these Floating IP's mapped to a Public IP

Firewall URL: http://<firewall_ip_address>

Firewall – NAT – 1:1

Please check the available Public IP address for use. Ensure there are no repetitions and once added, press the Apply button appearing in the screen.

And example screen is given below:



Step 07 – Verify the installation

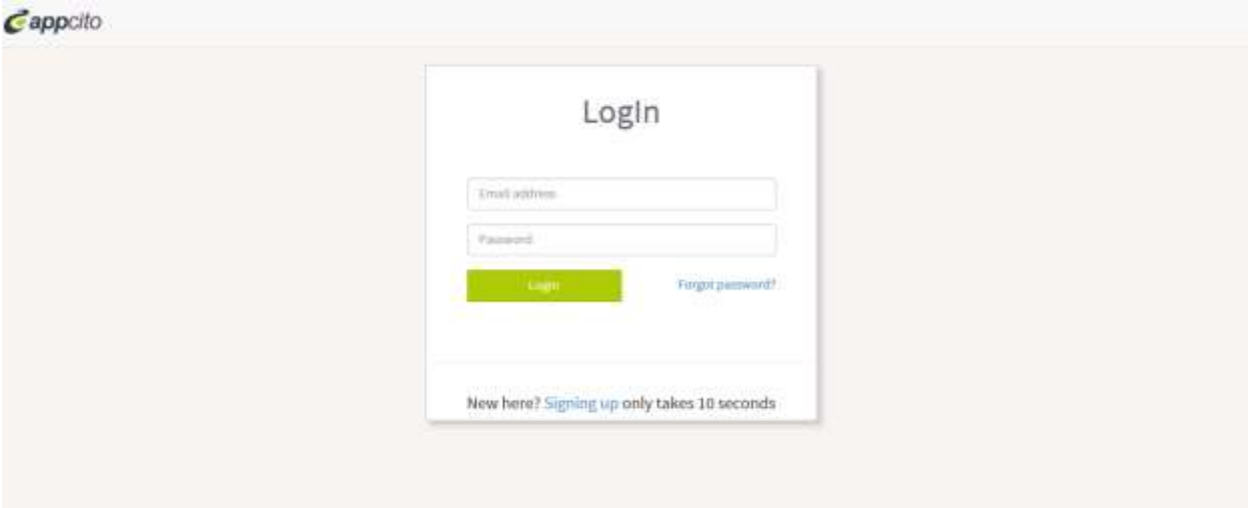
You would see the 11 Appcito Cloud Services Controller (CSC) instances launched in the Horizon UI, with the instance name with

1. UI
2. Edge
3. Barista
4. Message Bus
5. Config Data Store
6. Registry Server
7. Metrics Collector
8. RTP Co-ordinator
9. RTP Worker

10. RTP Manager
11. Metrics Data Store

Make the changes in the DNS map the UI/API Public IP address to a domain name

Step 08 – Verify that a login portal page appears when accessing the UI domain on the browser.



With this, CSC system installation is complete.

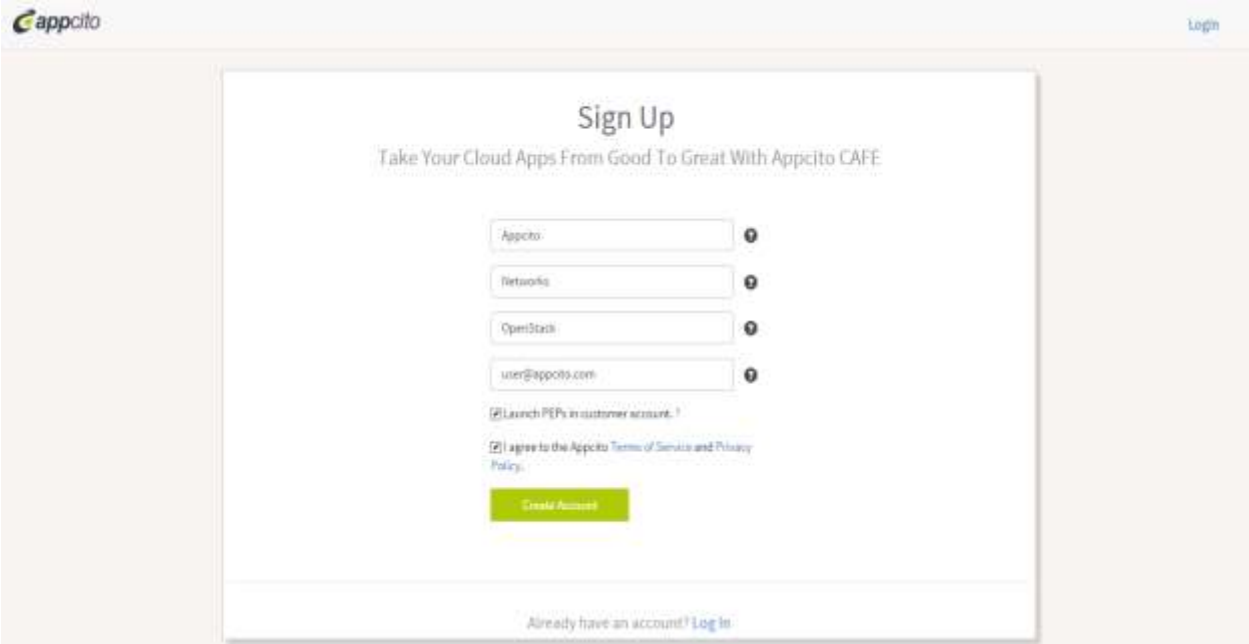
5.4 Testing

5.4.1 Target use case(s)

1. Creating an account on Appcito ADS
2. Add and application definition for a new account
3. Caffeinate the application in the MOS to send traffic
4. Observe metrics as well as traffic flow through Appcito CSP
5. Delete the application

5.4.2 Test cases

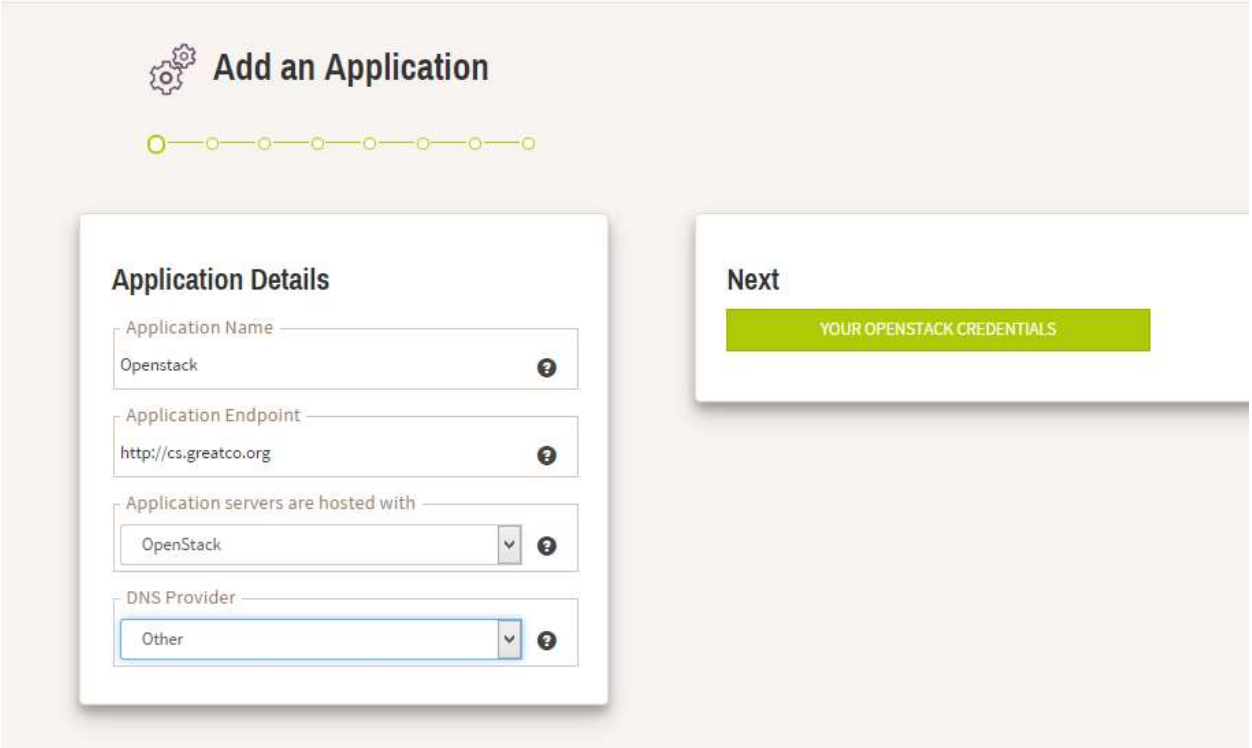
1. Sign up account in Appcito ADS – https://<UI_DOMAIN/#/signup



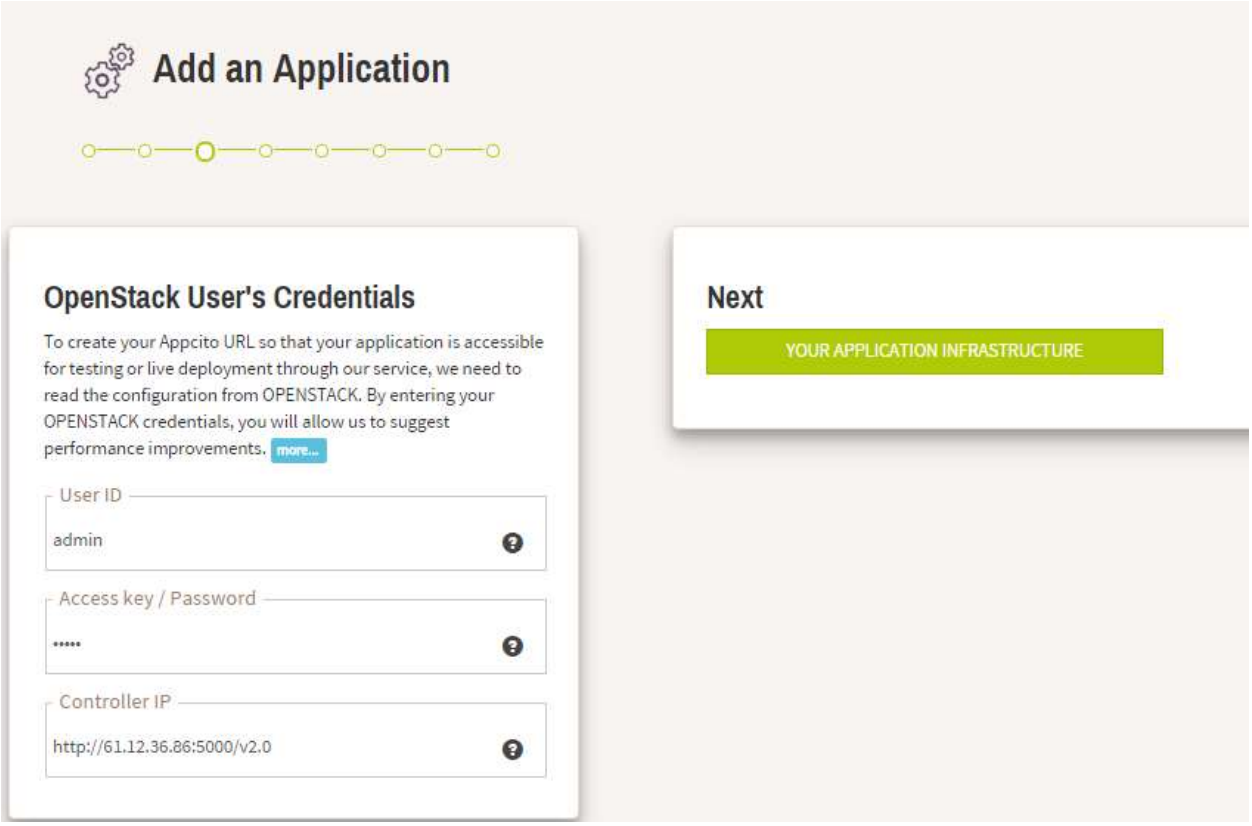
An e-mail will be sent along with the activation URL. Please activate the account and set the password.

2. Adding Application Details:

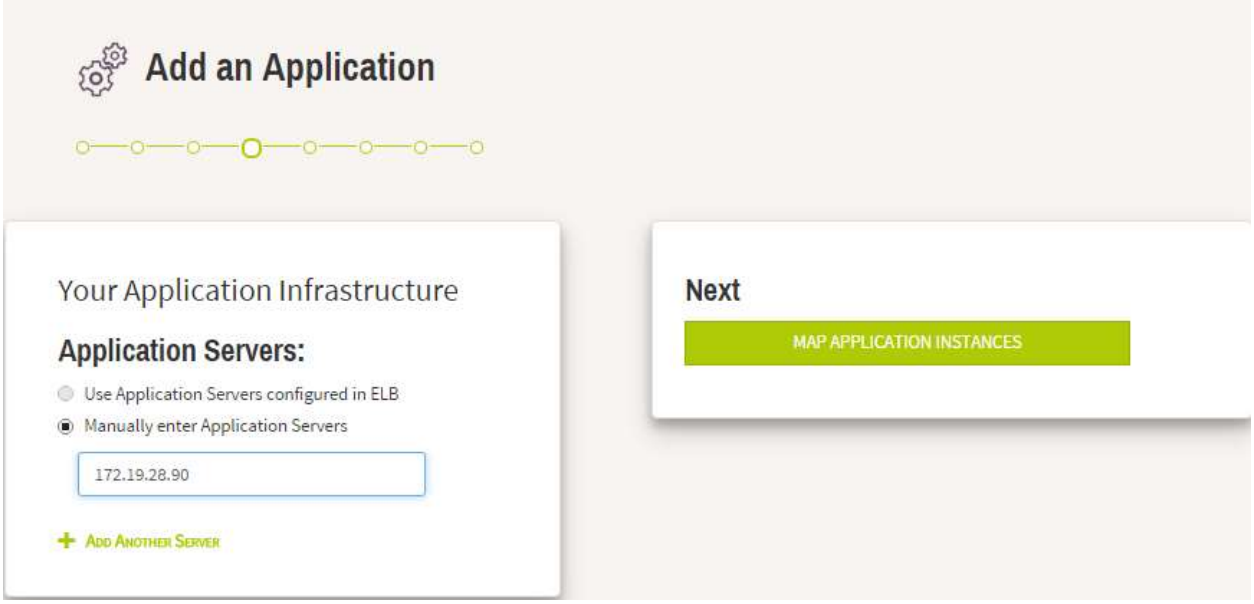
Login to the account created and follow the steps:



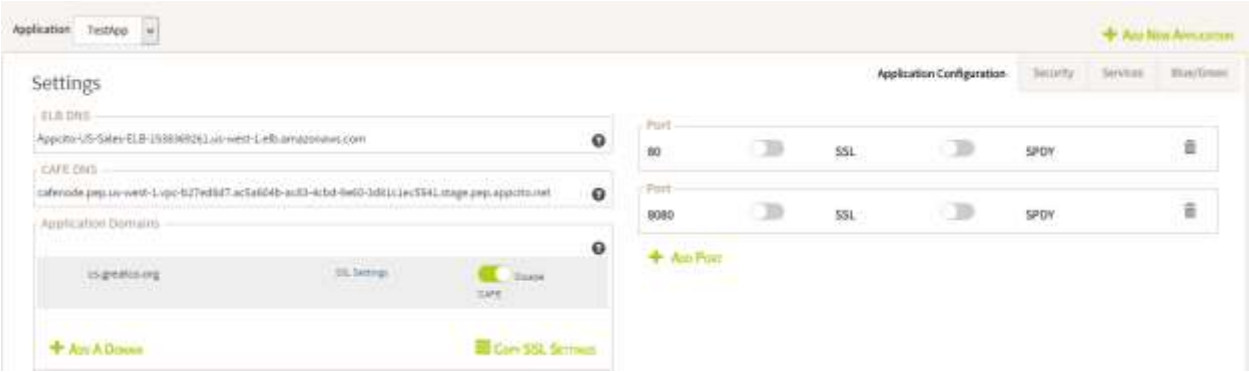
3. Adding Openstack User credentials



4. Adding Application Server details



5. Caffeination successful



Upon successful caffeination, modify the DNS for the application server to point to the CAFÉ DNS in the above screen with CNAME entry. By this step, the traffic will flow through Appcito CSP.

6. Metrics

5.4.2 Test Tools

1. Application server with tomcat available in MOS listening on port 80
2. Laptop as a client to send traffic to the Application server

5.4.3 Test Results

1. User creation should be successful and should be able to login using newly created username and password.
2. After adding the application and Caffeinating the Appcito CSP should launch in the MOS
3. When traffic is sent to the Application domain, the Metrics will get populated in the Dashboard.