Installation Runbook for

vIMS (Project Clearwater)

Application Type	*Virtualized Network Function*
Application Version	V1.0
Mirantis OpenStack Version	7.0
OpenStack Version	Kilo

Content

Document History

Version	Revision Date	Description
1.0	02-02-2016	Initial Version

1 Introduction

This document is to serve as a detailed Deployment Guide for a virtualized IMS NFV application based on open source Project Clearwater using Cloudify cloud management platform on Mirantis OpenStack. This document describes the reference architecture, installation steps for certified MOS+Clearwater vIMS automated deployment, monitoring and testing procedures.

1.1 Target Audience

This document is intended for the OpenStack administrators who are looking for deploying and testing VNF application such as distributed vIMS based on Clearwater solution with Mirantis based OpenStack cloud platform.

2 Application overview

2.1 Project Clearwater

Clearwater is IMS in the Cloud. Clearwater follows IMS architectural principles and supports all of the key standardized interfaces expected of an IMS core network. But unlike traditional implementations of IMS, Clearwater was designed from the ground up for the Cloud. To learn more about Project Clearwater, please see http://www.projectclearwater.org/

2.1 MOS Configuration

- Compute type: KVM
- Networking: Neutron with VLAN segmentation
- All OpenStack settings in the Settings tab of the Fuel Web UI at their defaults.

2.3 Nodes

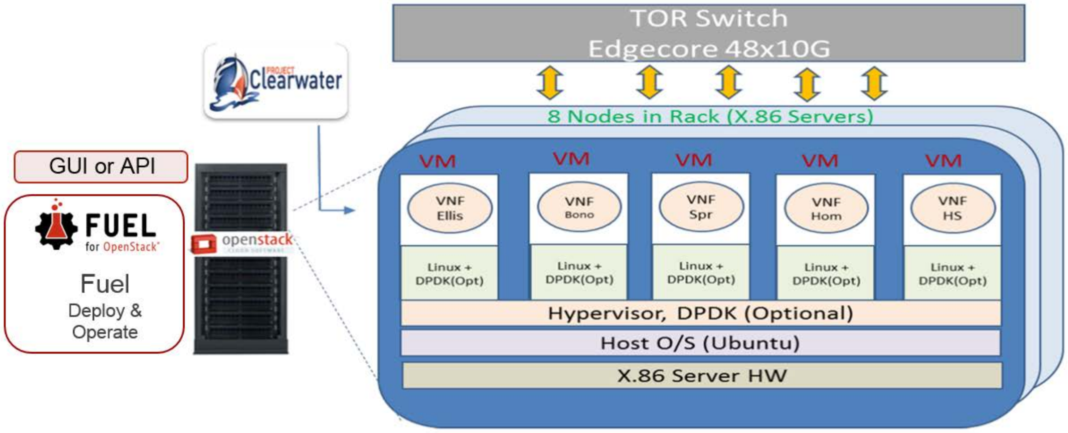
- Fuel Master VM: 8 vcpu, 8192MB RAM, 2 NICs
- 3 Controllers: Ciii S5140 1U Server, 2x CPU Intel E5-2630v3 8c 2.5GHz, 8x Micron 32GB DDR4 RAM, 1x 480GB Intel SSD S3500
- 4 Computes: Ciii S5140 1U Server, 2x CPU Intel E5-2630v3 8c 2.5GHz, 8x Micron 32GB DDR4 RAM, 1x 480GB Intel SSD S3500
- 1 Developer Node: Ciii S5140 1U Server, 2x CPU Intel E5-2630v3 8c 2.5GHz, 8x Micron 32GB DDR4 RAM, 1x 480GB Intel SSD S3500

2.4 Networking topology

- Each host has 1x1Gb ports, used for:
  a. Admin(PXE)
- Each host has 2x10Gb ports, used for:
  b. Storage, Management and Public networks

c. Private network

3 Joint Reference Architecture



Mirantis OpenStack + Clearwater vIMS at CloudLabs on Ciii “Victoria” 1U servers

3.1 Bono

The Bono nodes form a horizontally scalable SIP edge proxy. It provides both a SIP IMS compliant interface as well as a WebRTC interface to clients.

It also comes with a load balancer to handle a lot of requests from various clients. A client maintains connection to the Bono node until it completes its registration. In case the connection drops or fails for some other reason it can move to another Bono node.

Bono is not a compulsory component, Clearwater can also be deployed with a third-party P-CSCF

3.2 Sprout

The Sprout nodes act as a horizontally scalable, combined SIP registrar and authoritative routing proxy, and handle client authentication and the ISC interface to application servers.

The Sprout cluster includes a redundant memcached cluster storing client registration data. SIP transactions are load balanced across the Sprout cluster, so there is no longlived association between a client and a particular Sprout node.
Sprout uses REST API provided by Homestead and Homer to retrieve HSS configuration such as authentication data/user profiles, and other service settings. It communicates with Bono, Application servers using SIP.

Comparing with IMS architecture, Sprout performs most of the roles of both I-CSCF and S-CSCF.

3.3 Homestead

Homestead provides a HTTP REST API to Sprout for retrieving authentication credentials and user profile information.

The Homestead nodes run as a cluster using Cassandra as the store for mastered/mirrored data. In the IMS architecture, the HSS mirror function is considered to be part of the I-CSCF and S-CSCF components, so in Clearwater I-CSCF and S-CSCF function is implemented with a combination of Sprout and Homestead clusters

3.4 Homer

Homer is a standard XML Document Management Server used to store Multimedia Telephony service settings documents for each user of the system. The Homer nodes run as a cluster using Cassandra as the data store for mastered/mirrored data.

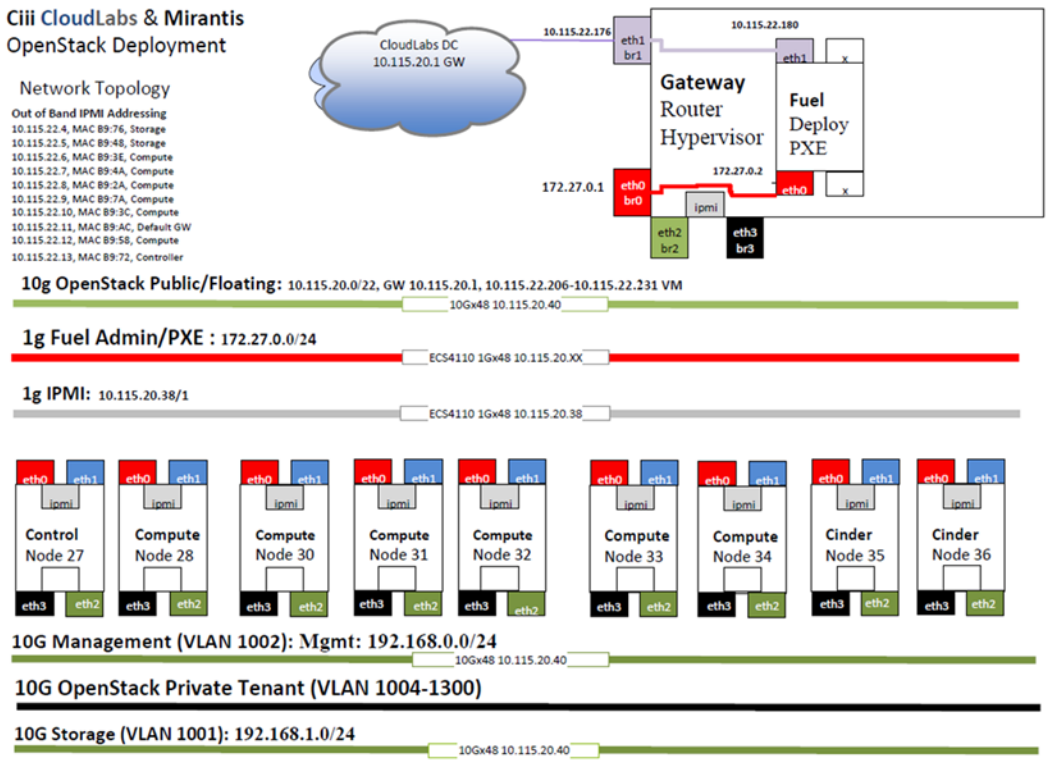
3.5 Ellis

Ellis is a sample provisioning portal providing self sign-up, password management, line management and control of MMTEL service settings. It is not intended to be a part of production Clearwater deployments (it is not easy to horizontally scale because of the MySQL underpinnings for one thing) but to make the system easy to use out of the box.

3.6 Ralf

Ralf provides an HTTP API that both Bono and Sprout can use to report billable events that should be passed to the CDF (Charging Data Function) over the Rf billing interface. Ralf uses a cluster of Memcached instances and a cluster of Chronos instances to store and manage session state, allowing it to confirm to the Rf protocol.

4 Physical & Logical Network Topology


Ciii CloudLabs & Mirantis
OpenStack Deployment
Network Topology
Out of Band IPMI Addressing
10.115.22.4, MAC B9:76, Storage
10.115.22.5, MAC B9:48, Storage
10.115.22.6, MAC B9:3E, Compute
10.115.22.7, MAC B9:4A, Compute
10.115.22.8, MAC B9:2A, Compute
10.115.22.9, MAC B9:7A, Compute
10.115.22.10, MAC B9:3C, Compute
10.115.22.11, MAC B9:AC, Default GW
10.115.22.12, MAC B9:58, Compute
10.115.22.13, MAC B9:72, Controller
CloudLabs DC
10.115.20.1 GW
10.115.22.176
eth1 br1
10.115.22.180
eth1
x
Gateway
Router
Hypervisor
Fuel
Deploy
PXE
172.27.0.2
172.27.0.1
eth0 br0
eth0
x
ipmi
eth2 br2
eth3 br3
10g OpenStack Public/Floating: 10.115.20.0/22, GW 10.115.20.1, 10.115.22.206-10.115.22.231 VM
10Gx48 10.115.20.40
1g Fuel Admin/PXE : 172.27.0.0/24
ECS4110 1Gx48 10.115.20.XX
1g IPMI: 10.115.20.38/1
ECS4110 1Gx48 10.115.20.38
eth0
eth1
ipmi
eth3
eth2
Control Node 27
Compute Node 28
Compute Node 30
Compute Node 31
Compute Node 32
Compute Node 33
Compute Node 34
Cinder Node 35
Cinder Node 36
10G Management (VLAN 1002): Mgmt: 192.168.0.0/24
10Gx48 10.115.20.40
10G OpenStack Private Tenant (VLAN 1004-1300)
10G Storage (VLAN 1001): 192.168.1.0/24
10Gx48 10.115.20.40

NFV Cluster at CloudLabs – Network Topology

5 Installation & Configuration

5.1 Environment preparation

Please refer Mirantis Fuel 7.0 Documentation for environment setup:
https://docs.mirantis.com/openstack/fuel/fuel-7.0/user-guide.html#boot-the-fuel-master-node

MOS 7.0 ISO is available at https://www.mirantis.com/products/mirantis-openstack-software/

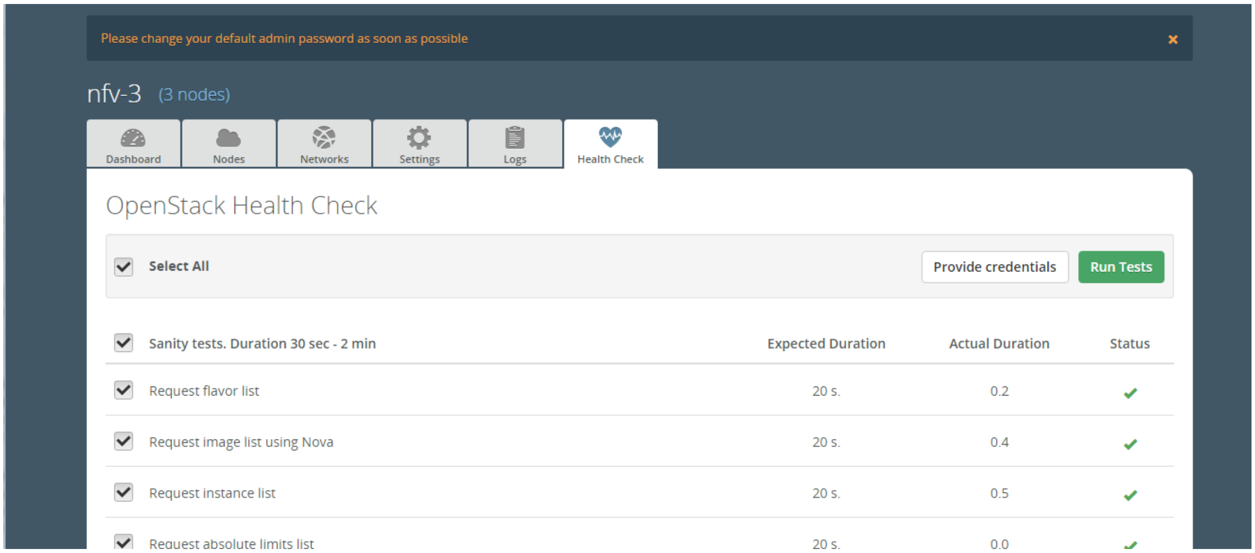
5.2 MOS installation

Please refer Mirantis Fuel 7.0 Documentation for MOS installation:
https://docs.mirantis.com/openstack/fuel/fuel-7.0/#guides

5.2.1 Health Check Results

Please refer to Mirantis Fuel 7.0 Documentation in health check section:
https://docs.mirantis.com/openstack/fuel/fuel-7.0/user-guide.html#post-deployment-check

The deployment should pass for basic sanity test, functional test and HA test if controller are configured in HA mode.



Please change your default admin password as soon as possible ×

nfv-3 (3 nodes)

OpenStack Health Check

☑ Select All — Provide credentials — Run Tests

☑ Sanity tests. Duration 30 sec - 2 min	Expected Duration	Actual Duration	Status
☑ Request flavor list	20 s.	0.2	✓
☑ Request image list using Nova	20 s.	0.4	✓
☑ Request instance list	20 s.	0.5	✓
☑ Request absolute limits list	20 s.	0.0	✓

5.3 Project Clearwater vIMS installation steps

These steps will guide you for installation of Cloudify Cloud Manager Platform (CMP) for Clearwater vIMS deployment on Mirantis OpenStack

A. Cloudify Cloud Manager Server Setup

Cloudify CLI can be used to create cloud manager server on Mirantis Openstack environment.

1. On developer server (marked as bare metal OS while Fuel 7.0 deployment) install openstack clients

```
# apt-get install keystone python-openstackclient apache2 libapache2-mod-wsgi memcached python-memcache
# apt-get install glance python-glanceclient
# apt-get install nova-api nova-cert nova-conductor nova-consoleauth nova-novncproxy nova-scheduler python-novaclient
# apt-get install neutron-server neutron-plugin-ml2 python-neutronclient
```

2. Install python packages

```
# sudo apt-get update
# sudo apt-get install git python-pip python-dev python-virtualenv –y
```

3. Create virtual environment:

```
# virtualenv cloudify
# source cloudify/bin/activate
# cd cloudify
```

4. Install cloudify CLI version 3.2 with the PIP command :

```
# pip install cloudify==3.3
```

5. Test if the command cfy exists

```
# cfy
```

This result appears on console,

```
usage: cfy [-h] [--version]
        {status,blueprints,bootstrap,teardown,workflows,recover,node-
instances,snapshots,deployments,init,agents,dev,use,plugins,nodes,executions,local,events,ssh}
        ...
cfy: error: too few arguments
```

B. Deploy Cloudify Manager Server:

2. Log into the host where you installed the **Cloudify CLI** and enter in the virtual environment with source command.

```
# source cloudify/bin/activate
```

3. Prepare your directory

```
# mkdir -p cloudify-manager
# cd cloudify-manager
```

4. Download manager blueprint version 3.3

```
# git clone -b 3.3-build https://github.com/cloudify-cosmo/cloudify-manager-blueprints.git
```

5. Prepare deployment on OpenStack platform

```
# cfy init
# cd cloudify-manager-blueprints/
```

6. Install required packages for deployment

```
# cfy local create-requirements -o requirements.txt -p openstack-manager-blueprint.yaml
# sudo pip install -r requirements.txt
```

7. The configuration for the cloudify manager deployment is contained in a YAML file. You will need to edit the parameter according to your Mirantis Openstack environment.

```
# vi openstack-manager-blueprint-inputs.yaml
```

```
keystone_username: 'admin'
keystone_password: 'admin'
keystone_tenant_name: 'admin'
keystone_url: 'http://10.115.22.239:5000/v2.0/'
region: 'RegionOne'
ssh_key_filename: ~/.ssh/cloudify-manager-kp.pem
agent_private_key_path: ~/.ssh/cloudify-agent-kp.pem
manager_public_key_name: 'cloudify-manager-kp'
agent_public_key_name: 'cloudify-agent-kp'
image_id: '17d039a7-1fb9-40ac-a0f3-1a90811f80de'
flavor_id: '3'
external_network_name: 'net04_ext'
ssh_user: 'centos’
agents_user: 'centos'
```

8. Launch the deployment of cloudify manager server

```
 # cfy bootstrap --install-plugins -p openstack-manager-blueprint.yaml -i openstack-manager-blueprint-
inputs.yaml
```

9. Check the proper functioning of the server

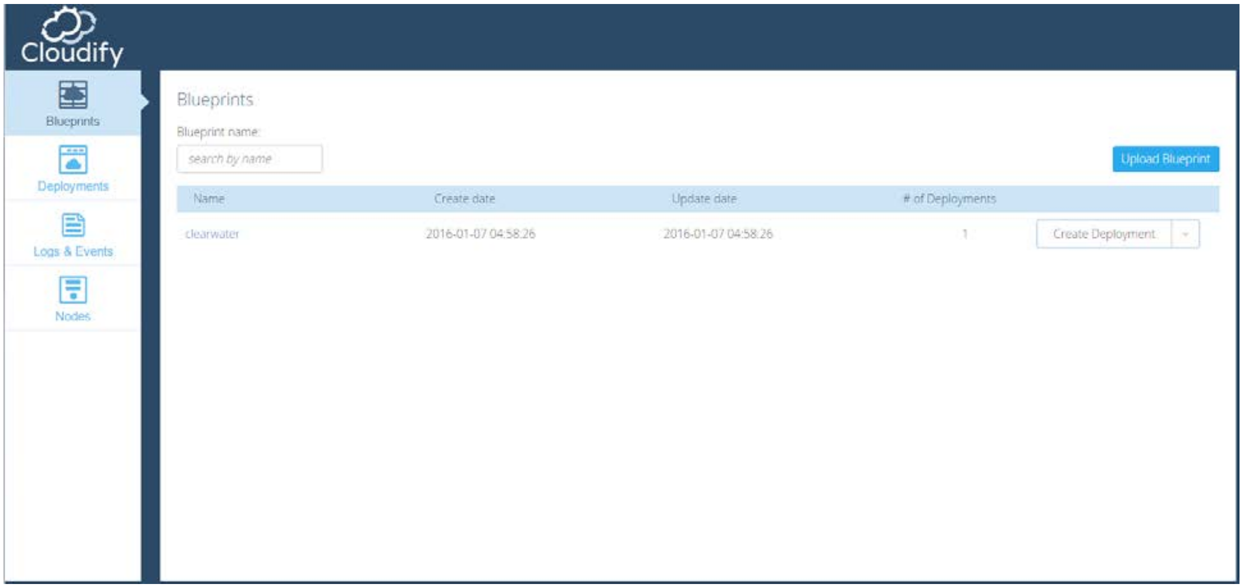
```
# cfy status
```

10. If this result appears on console, your cloudify manager is installed and operating

```
Getting management services status... [ip=10.115.22.240]

Services:
+--------------------------------+---------+
|            service             |  status |
+--------------------------------+---------+
| InfluxDB                       | running |
| Celery Management              | running |
| Logstash                       | running |
| RabbitMQ                       | running |
| AMQP InfluxDB                  | running |
| Manager Rest-Service           | running |
| Cloudify UI                    | running |
| Webserver                      | running |
| Riemann                        | running |
| Elasticsearch                  | running |
+--------------------------------+---------+
```

11. Access the Cloud-manager GUI using the Associate IP address of the cloud-manager VM on Mirantis Openstack



Blueprints

Blueprint name:

Name	Create date	Update date	# of Deployments
clearwater	2016-01-07 04:58:26	2016-01-07 04:58:26	1

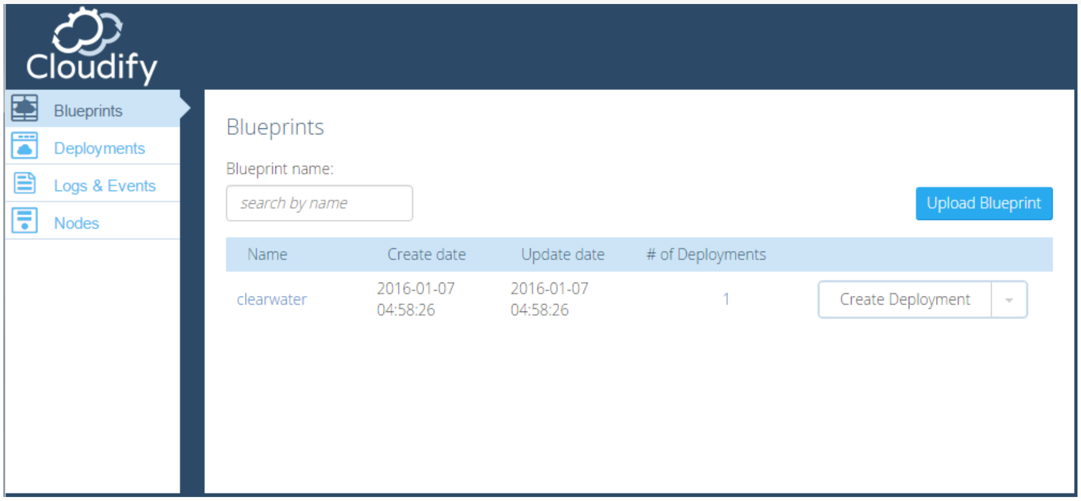
C. Clearwater vIMS deployment on Mirantis OpenStack

1. Download blueprint using git

```
# cd ~/cloudify/cloudify-manager/
# mkdir blueprints
# cd blueprints
# git clone -b stable https://github.com/Orange-OpenSource/opnfv-cloudify-clearwater.git
```

2. Upload blueprint on the cloudify orchestrator :

```
# cd opnfv-cloudify-clearwater
# cfy blueprints upload -b clearwater -p openstack-blueprint.yaml
```



Blueprints

Blueprint name:

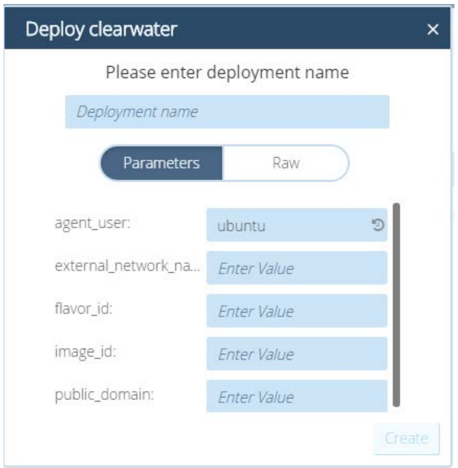
Name	Create date	Update date	# of Deployments
clearwater	2016-01-07 04:58:26	2016-01-07 04:58:26	1

3. To create and launch the deployment you will have to pass the parameters of your openstack environment

```
# cp inputs/openstack.yaml.template inputs/inputs.yaml
# vi inputs/inputs.yaml
```

```
image_id: '1cbecabc-d4a9-41d5-8421-bcfe9324ddea'     # OS image ID (Ubuntu 14.04)
flavor_id: '2'                          # Flavor ID (~ 2 Go RAM)
external_network_name: 'net04_ext'            # external network on Openstack
agent_user: 'ubuntu'                    # By default is ubuntu for ubuntu image
public_domain: 'clearwater.pub'               # SIP domain name
```

You can also pass the parameters from the cloud-manager GUI—Create Deployment using Blueprints



Deploy clearwater

Please enter deployment name

Deployment name

agent_user: ubuntu

external_network_na... Enter Value

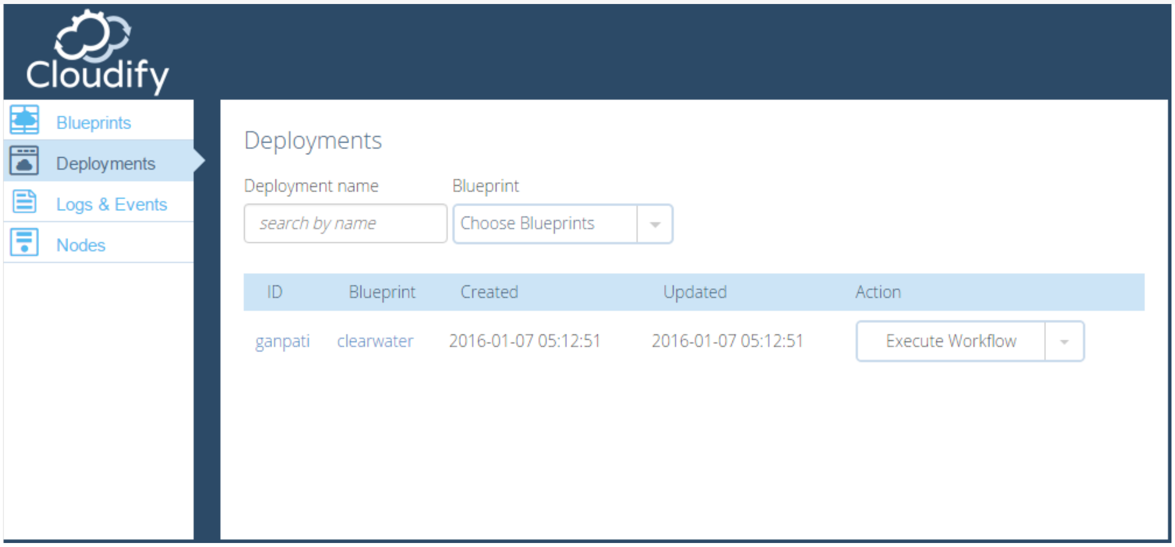
flavor_id: Enter Value

image_id: Enter Value

public_domain: Enter Value

4. Once the input file is completed, you must create the deployment on orchestrator

```
# cfy deployments create -b clearwater -d clearwater-test --inputs inputs/inputs.yaml
```



Deployments

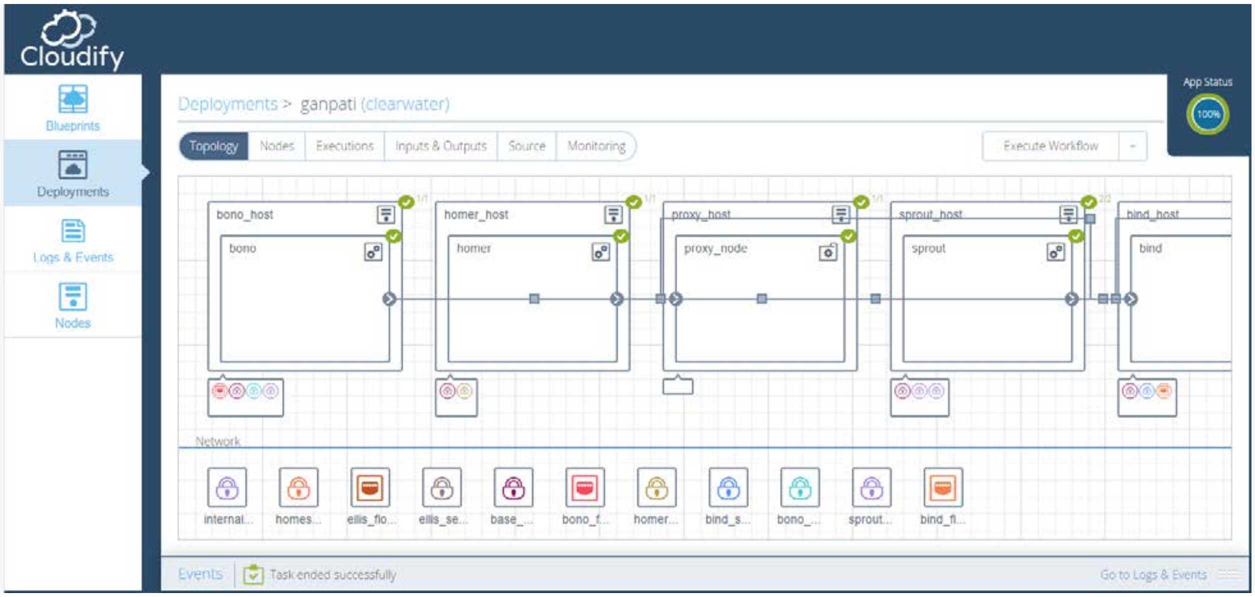
Deployment name: search by name

Blueprint: Choose Blueprints

ID	Blueprint	Created	Updated	Action
ganpati	clearwater	2016-01-07 05:12:51	2016-01-07 05:12:51	Execute Workflow

5. Launch clearwater deployment

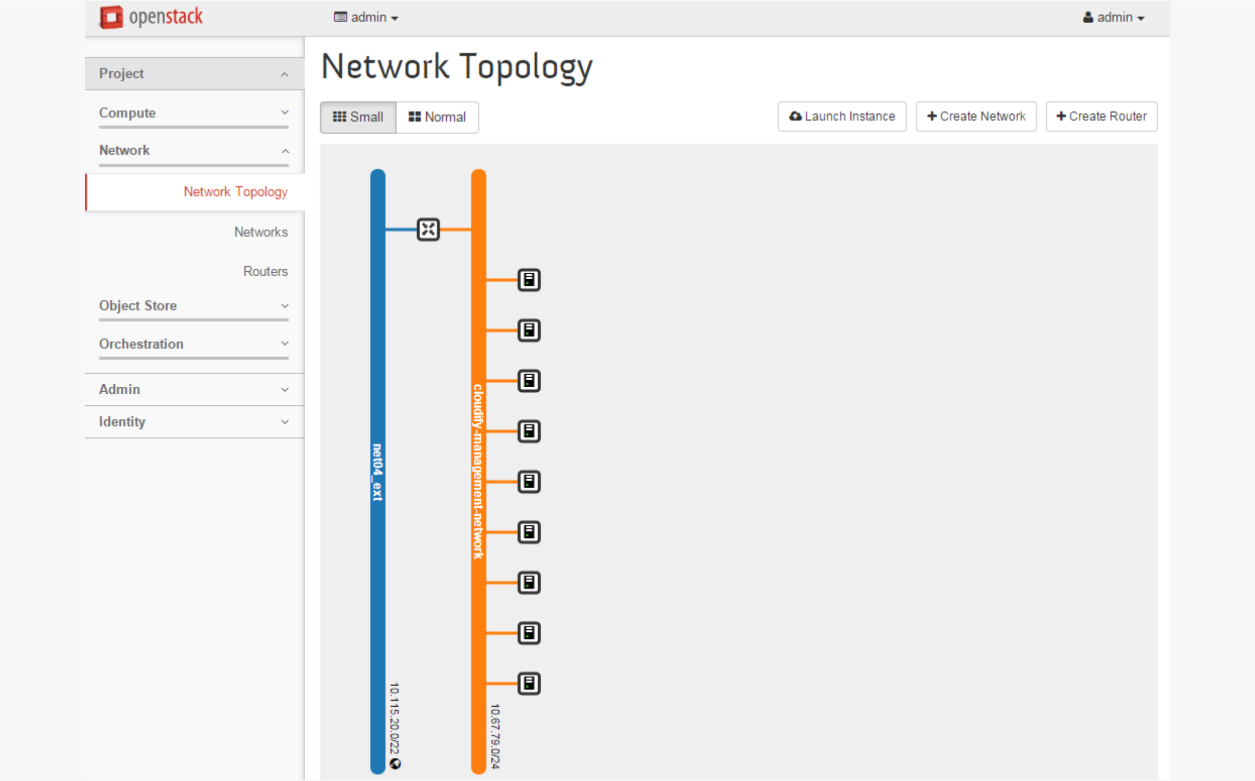
```
# cfy executions start -w install -d clearwater-test
```

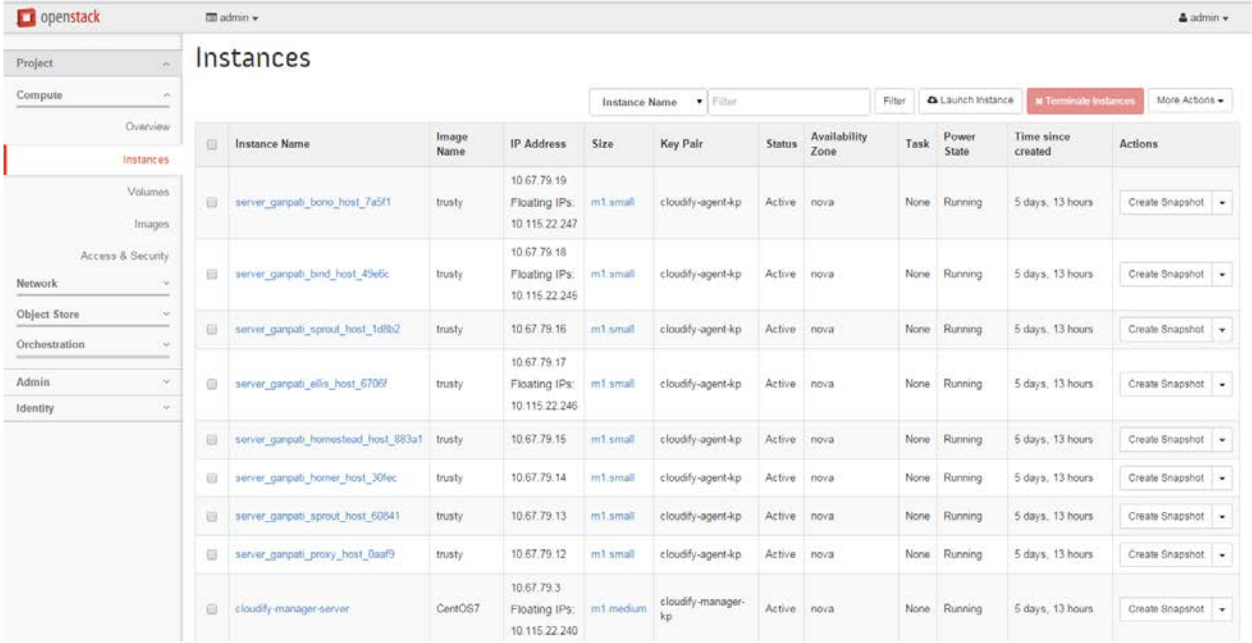


Note:

During the deployment, please confirm if the 'default security' of the openstack is added on VM instances. Pass all UDP, TCP and ICMP traffic during Clearwater installation. The default security group can be deleted from all instances once the application is installed successfully.

Following topology and VMs are created on Mirantis OpenStack after the deployment is successful.





Instances

□	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
□	server_ganpati_bono_host_7a5f1	trusty	10.67.79.19 Floating IPs: 10.115.22.247	m1.small	cloudify-agent-kp	Active	nova	None	Running	5 days, 13 hours	Create Snapshot
□	server_ganpati_bind_host_49e6c	trusty	10.67.79.18 Floating IPs: 10.115.22.245	m1.small	cloudify-agent-kp	Active	nova	None	Running	5 days, 13 hours	Create Snapshot
□	server_ganpati_sprout_host_1d8b2	trusty	10.67.79.16	m1.small	cloudify-agent-kp	Active	nova	None	Running	5 days, 13 hours	Create Snapshot
□	server_ganpati_ellis_host_6706f	trusty	10.67.79.17 Floating IPs: 10.115.22.246	m1.small	cloudify-agent-kp	Active	nova	None	Running	5 days, 13 hours	Create Snapshot
□	server_ganpati_homestead_host_883a1	trusty	10.67.79.15	m1.small	cloudify-agent-kp	Active	nova	None	Running	5 days, 13 hours	Create Snapshot
□	server_ganpati_homer_host_30fec	trusty	10.67.79.14	m1.small	cloudify-agent-kp	Active	nova	None	Running	5 days, 13 hours	Create Snapshot
□	server_ganpati_sprout_host_60841	trusty	10.67.79.13	m1.small	cloudify-agent-kp	Active	nova	None	Running	5 days, 13 hours	Create Snapshot
□	server_ganpati_proxy_host_0aaf9	trusty	10.67.79.12	m1.small	cloudify-agent-kp	Active	nova	None	Running	5 days, 13 hours	Create Snapshot
□	cloudify-manager-server	CentOS7	10.67.79.3 Floating IPs: 10.115.22.240	m1.medium	cloudify-manager-kp	Active	nova	None	Running	5 days, 13 hours	Create Snapshot

5.4 Limitations

5.4.1 Limitations on Built-in Calling Features

Clearwater's built-in TAS is an extension of the transaction-stateful proxy function in Sprout. The functionality of this TAS is therefore limited to calling features that can be implemented by simple proxy-based manipulation of SIP messages. This includes the following kinds of features:

- Call Forwarding in all its varieties
- Call Barring in all its varieties
- Caller ID presentation and restriction services

Calling features that involve manipulation of call legs, for example Call Transfer or Three-Way Calling, typically make use of the SIP REFER request which will require implementing Telephony Application Servers in the form of Back-to-Back User Agents (B2BUA). This is not currently supported in demo deployment.

5.4.2 Session Border Control

Bono implements the IMS Proxy-CSCF function. Session Border Controllers (SBCs) typically also implement the P-CSCF function. This does not mean that Bono is an SBC.

SBCs perform a wide range of functions above and beyond the P-CSCF function, including protecting the IMS network from various kinds of malicious attack, manipulating the contents of SIP messages for reasons of interoperability and relaying media in support of NAT traversal. Bono does not perform any of these functions.

5.5 Testing

5.5.1 Test cases

A. Clearwater Live Tests

The following live tests can be run over a clearwater deployment to confirm that the high level function is working for the vIMS.

1. On the developer node install Ruby version 1.9.3 and its dependencies.

```
# sudo apt-get install build-essential git --yes
# curl -L https://get.rvm.io | bash -s stable
# source ~/.rvm/scripts/rvm
# rvm autolibs enable
# rvm install 1.9.3
# rvm use 1.9.3
```

2. Run the following to download and install the Clearwater test suite

```
# git clone -b stable --recursive git@github.com:Metaswitch/clearwater-live-test.git
# cd clearwater-live-test
# bundle install
```

5.5.2 Test Results

1. Running the tests

```
# rake test[<domain>] SIGNUP_CODE=<code> PROXY=<Bono domain> ELLIS=<Ellis domain>
```

where domain: clearwater.pub, code: secret, PROXY=10.115.22.247, ELLIS: 10.115.22.246

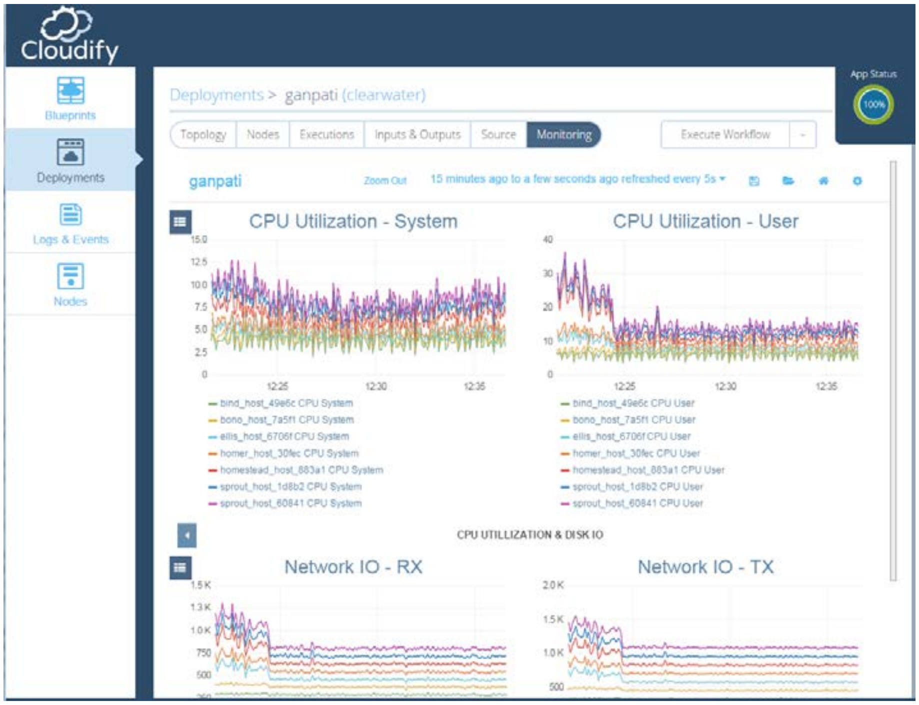
```
rake test[clearwater.pub] SIGNUP_CODE=secret PROXY=10.115.22.247 ELLIS=10.115.22.246
Basic Call - Mainline (TCP) - (6505550065, 6505550741) Passed
Basic Call - Mainline (UDP) - (6505550676, 6505550562) Passed
Basic Call - Tel URIs (TCP) - (6505550120, 6505550024) Passed
Basic Call - Tel URIs (UDP) - (6505550089, 6505550923) Passed
Basic Call - Unknown number (TCP) - (6505550956, 6505550362) Passed
Basic Call - Unknown number (UDP) - (6505550565, 6505550345) Passed
Basic Call - Rejected by remote endpoint (TCP) - (6505550112, 6505550710) Passed
Basic Call - Rejected by remote endpoint (UDP) - (6505550441, 6505550908) Passed
Basic Call - Messages - Pager model (TCP) - (6505550579, 6505550454) Passed
Basic Call - Messages - Pager model (UDP) - (6505550026, 6505550971) Passed
Basic Call - Pracks (TCP) - (6505550460, 6505550922) Passed
Basic Call - Pracks (UDP) - (6505550713, 6505550477) Passed
Basic Registration (TCP) - (6505550972) Passed
Basic Registration (UDP) - (6505550992) Passed
Multiple Identities (TCP) - (6505550767, 6505550346) Passed
Multiple Identities (UDP) - (6505550394, 6505550484) Passed
Call Barring - Outbound Rejection (TCP) - (6505550282, 6505550369) Passed
Call Barring - Outbound Rejection (UDP) - (6505550578, 6505550743) Passed
Call Diversion - Not registered (TCP) - (6505550134, 6505550559, 6505550576) Passed
Call Diversion - Not registered (UDP) - (6505550549, 6505550471, 6505550809) Passed
Call Diversion - Not reachable (not registered) (TCP) - (6505550186, 6505550690, 6505550605) Passed
Call Diversion - Not reachable (not registered) (UDP) - (6505550359, 6505550834, 6505550390) Passed
Call Diversion - Not reachable (408) (TCP) - (6505550224, 6505550985, 6505550452) Passed
Call Diversion - Not reachable (408) (UDP) - (6505550031, 6505550522, 6505550775) Passed
Call Diversion - Not reachable (503) (TCP) - (6505550964, 6505550993, 6505550395) Passed
Call Diversion - Not reachable (503) (UDP) - (6505550469, 6505550968, 6505550844) Passed
Call Diversion - Not reachable (500) (TCP) - (6505550218, 6505550164, 6505550384) Passed
Call Diversion - Not reachable (500) (UDP) - (6505550400, 6505550088, 6505550693) Passed
Call Diversion - Busy (TCP) - (6505550265, 6505550650, 6505550957) Passed
Call Diversion - Busy (UDP) - (6505550146, 6505550700, 6505550935) Passed
Call Diversion - Unconditional (TCP) - (6505550819, 6505550344, 6505550951) Passed
Call Diversion - Unconditional (UDP) - (6505550345, 6505550985, 6505550521) Passed
Call Diversion - No answer (TCP) - (6505550160, 6505550243, 6505550262) Passed
Call Diversion - No answer (UDP) - (6505550801, 6505550022, 6505550071) Passed
Call Diversion - Bad target URI (TCP) - (6505550928, 6505550509) Passed
Call Diversion - Bad target URI (UDP) - (6505550127, 6505550474) Passed
Call Diversion - Audio-only call (TCP) - (6505550163, 6505550147, 6505550706, 6505550717) Passed
Call Diversion - Audio-only call (UDP) - (6505550805, 6505550179, 6505550592, 6505550605) Passed
Call Diversion - Audio-video call (TCP) - (6505550716, 6505550199, 6505550073, 6505550789) Passed
Call Diversion - Audio-video call (UDP) - (6505550345, 6505550551, 6505550741, 6505550692) Passed
Call Waiting - Accepted (TCP) - (6505550672, 6505550020, 6505550780) Passed
Call Waiting - Accepted (UDP) - (6505550160, 6505550556, 6505550223) Failed

Call Waiting - Cancelled (TCP) - (6505550600, 6505550423, 6505550518) Passed
Call Waiting - Cancelled (UDP) - (6505550111, 6505550274, 6505550340) Passed
CANCEL - Mainline (TCP) - (6505550333, 6505550527) Passed
CANCEL - Mainline (UDP) - (6505550120, 6505550756) Passed
Filtering - Accept-Contact (TCP) - (6505550483, 6505550179) Passed
Filtering - Accept-Contact (UDP) - (6505550829, 6505550559) Passed
Filtering - Accept-Contact no match (TCP) - (6505550535, 6505550760) Passed
Filtering - Accept-Contact no match (UDP) - (6505550565, 6505550692) Passed
Filtering - Accept-Contact negated match (TCP) - (6505550027, 6505550721) Passed
Filtering - Accept-Contact negated match (UDP) - (6505550462, 6505550226) Passed
Filtering - RFC3841 example (TCP) - (6505550939, 6505550907) Passed
Filtering - RFC3841 example (UDP) - (6505550126, 6505550409) Passed
Filtering - Reject-Contact no match (TCP) - (6505550042, 6505550218) Passed
Filtering - Reject-Contact no match (UDP) - (6505550888, 6505550981) Passed
Filtering - Reject-Contact match (TCP) - (6505550247, 6505550605) Passed
Filtering - Reject-Contact match (UDP) - (6505550806, 6505550672) Passed
Gemini - INVITE - VoIP device answers (TCP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - VoIP device answers (UDP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - Mobile device answers (TCP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - Mobile device answers (UDP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - VoIP device rejects (TCP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - VoIP device rejects (UDP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - Mobile device rejects (TCP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - Mobile device rejects (UDP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - Mobile device rejects with a 480 (TCP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - Mobile device rejects with a 480 (UDP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - Both reject, choose mobile response (TCP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - Both reject, choose mobile response (UDP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - Both reject, choose VoIP response (TCP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - Both reject, choose VoIP response (UDP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - Successful call with GR (TCP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - Successful call with GR (UDP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - Failed call with GR (TCP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - Failed call with GR (UDP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - Successful call with Accept-Contact (TCP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
```

```
Gemini - INVITE - Successful call with Accept-Contact (UDP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - Failed call with Accept-Contact (TCP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - INVITE - Failed call with Accept-Contact (UDP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - SUBSCRIBE - Mobile Notifies (TCP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - SUBSCRIBE - Mobile Notifies (UDP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - SUBSCRIBE - Joint 408 (TCP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
Gemini - SUBSCRIBE - Joint 408 (UDP) - Skipped (No gemini hostname provided)
  - Call with GEMINI=<hostname>
GRUU - REGISTER - two bindings with and without GRUU (TCP) - (6505550286) Passed
GRUU - REGISTER - two bindings with and without GRUU (UDP) - (6505550878) Passed
GRUU - REGISTER - binding suggested GRUU (TCP) - (6505550838) Passed
GRUU - REGISTER - binding suggested GRUU (UDP) - (6505550231) Passed
GRUU - REGISTER - instance ID requires escaping (TCP) - (6505550091) Passed
GRUU - REGISTER - instance ID requires escaping (UDP) - (6505550203) Passed
GRUU - REGISTER - three bindings with GRUUs (TCP) - (6505550599) Passed
GRUU - REGISTER - three bindings with GRUUs (UDP) - (6505550308) Passed
GRUU - Call - first endpoint GRUU as target (TCP) - (6505550838, 6505550377) Passed
GRUU - Call - first endpoint GRUU as target (UDP) - (6505550772, 6505550410) Passed
GRUU - Call - second endpoint GRUU as target (TCP) - (6505550349, 6505550700) Passed
GRUU - Call - second endpoint GRUU as target (UDP) - (6505550372, 6505550707) Passed
GRUU - Call - only GRUU as target (TCP) - (6505550430, 6505550004) Passed
GRUU - Call - only GRUU as target (UDP) - (6505550554, 6505550561) Passed
GRUU - Call - AoR as target (TCP) - (6505550109, 6505550827) Passed
GRUU - Call - AoR as target (UDP) - (6505550715, 6505550247) Passed
GRUU - Call - unknown GRUU as target (TCP) - (6505550255, 6505550531) Passed
GRUU - Call - unknown GRUU as target (UDP) - (6505550255, 6505550125) Passed
GRUU - Call - unknown GRUU as target - no GRUUs assigned (TCP) - (6505550100, 6505550515) Passed
GRUU - Call - unknown GRUU as target - no GRUUs assigned (UDP) - (6505550440, 6505550394) Passed
GRUU - Call - Reject-Contact interop (TCP) - (6505550063, 6505550397) Passed
GRUU - Call - Reject-Contact interop (UDP) - (6505550538, 6505550748) Passed
GRUU - Call - Accept-Contact interop (TCP) - (6505550384, 6505550451) Passed
GRUU - Call - Accept-Contact interop (UDP) - (6505550035, 6505550345) Passed
GRUU - Call - AoR with other param as target (TCP) - (6505550291, 6505550788) Passed
GRUU - Call - AoR with other param as target (UDP) - (6505550729, 6505550683) Passed
GRUU - Call - GRUU with other param as target (TCP) - (6505550208, 6505550754) Passed
GRUU - Call - GRUU with other param as target (UDP) - (6505550664, 6505550016) Passed
ISC Interface - Terminating (TCP) - Skipped (No hostname given)
  - Call with HOSTNAME=<publicly accessible hostname/IP of this machine>
ISC Interface - Terminating (UDP) - Skipped (No hostname given)
  - Call with HOSTNAME=<publicly accessible hostname/IP of this machine>
ISC Interface - Terminating (UDP AS) (TCP) - Skipped (No hostname given)
  - Call with HOSTNAME=<publicly accessible hostname/IP of this machine>
ISC Interface - Terminating (UDP AS) (UDP) - Skipped (Test is not valid for UDP)
ISC Interface - Terminating Failed (TCP) - Skipped (No hostname given)
  - Call with HOSTNAME=<publicly accessible hostname/IP of this machine>
ISC Interface - Terminating Failed (UDP) - Skipped (No hostname given)
  - Call with HOSTNAME=<publicly accessible hostname/IP of this machine>
ISC Interface - Third-party Registration (TCP) - Skipped (No hostname given)
  - Call with HOSTNAME=<publicly accessible hostname/IP of this machine>
ISC Interface - Third-party Registration (UDP) - Skipped (No hostname given)
  - Call with HOSTNAME=<publicly accessible hostname/IP of this machine>
ISC Interface - Third-party Registration - implicit registration (TCP) - Skipped (No hostname given)
  - Call with HOSTNAME=<publicly accessible hostname/IP of this machine>
ISC Interface - Third-party Registration - implicit registration (UDP) - Skipped (No hostname given)
  - Call with HOSTNAME=<publicly accessible hostname/IP of this machine>
```

```
ISC Interface - Redirect (TCP) - Skipped (No hostname given)
  - Call with HOSTNAME=<publicly accessible hostname/IP of this machine>
ISC Interface - Redirect (UDP) - Skipped (No hostname given)
  - Call with HOSTNAME=<publicly accessible hostname/IP of this machine>
ISC Interface - B2BUA (TCP) - Skipped (No hostname given)
  - Call with HOSTNAME=<publicly accessible hostname/IP of this machine>
ISC Interface - B2BUA (UDP) - Skipped (No hostname given)
  - Call with HOSTNAME=<publicly accessible hostname/IP of this machine>
Live Call - Dial out to a real number (TCP) - Skipped (No live number given)
  - Call with LIVENUMBER=<number>
Live Call - Dial out to a real number (UDP) - Skipped (No live number given)
  - Call with LIVENUMBER=<number>
Memento - Barred Call (UDP) - Skipped (No memento hostnames provided)
  - Call with MEMENTO_SIP=<SIP hostname> and MEMENTO_HTTP=<HTTP hostname>
Memento - Busy Call Forwarding (TCP) - Skipped (No memento hostnames provided)
  - Call with MEMENTO_SIP=<SIP hostname> and MEMENTO_HTTP=<HTTP hostname>
Memento - Busy Call Forwarding (UDP) - Skipped (No memento hostnames provided)
  - Call with MEMENTO_SIP=<SIP hostname> and MEMENTO_HTTP=<HTTP hostname>
Off-net calls - tel: URI (TCP) - Skipped (No off-net number given)
  - Call with OFF_NET_TEL=<a number set up in ENUM/BGCF to route to port 5072 on this machine>
Off-net calls - tel: URI (UDP) - Skipped (No off-net number given)
  - Call with OFF_NET_TEL=<a number set up in ENUM/BGCF to route to port 5072 on this machine>
Off-net calls - sip: URI (TCP) - Skipped (No off-net number given)
  - Call with OFF_NET_TEL=<a number set up in ENUM/BGCF to route to port 5072 on this machine>
Off-net calls - sip: URI (UDP) - Skipped (No off-net number given)
  - Call with OFF_NET_TEL=<a number set up in ENUM/BGCF to route to port 5072 on this machine>
SUBSCRIBE - reg-event (TCP) - (6505550959) Passed
SUBSCRIBE - reg-event (UDP) - (6505550694) Passed
SUBSCRIBE - reg-event with a GRUU (TCP) - (6505550461) Passed
SUBSCRIBE - reg-event with a GRUU (UDP) - (6505550594) Passed
Deleting leaked number: sip:6505550160@clearwater.pub
Deleting leaked number: sip:6505550193@clearwater.pub
Deleting leaked number: sip:6505550223@clearwater.pub
Deleting leaked number: sip:6505550390@clearwater.pub
Deleting leaked number: sip:6505550418@clearwater.pub
Deleting leaked number: sip:6505550556@clearwater.pub
Deleting leaked number: sip:6505550594@clearwater.pub
```

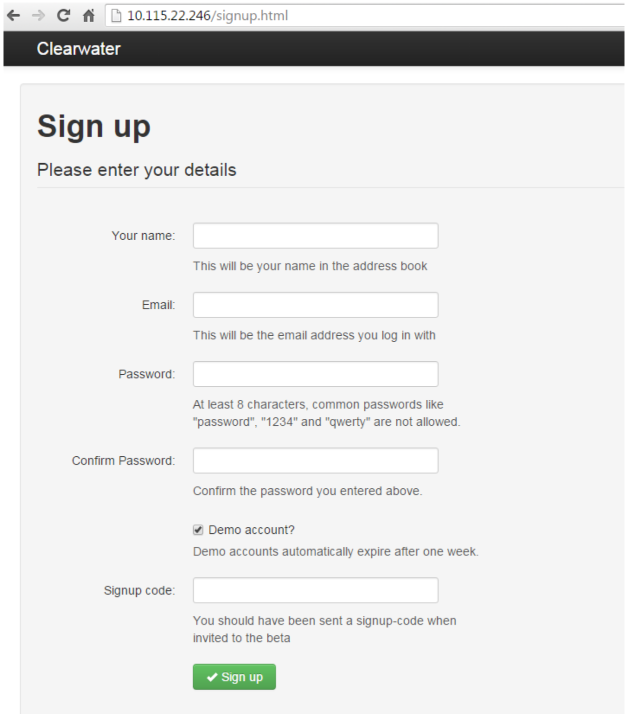
Cloudify Real-time Monitoring for CPU and network performance of the VMs on OpenStack.


Cloudify
Blueprints
Deployments
Logs & Events
Nodes
Deployments > ganpati (clearwater)
App Status
100%
Topology
Nodes
Executions
Inputs & Outputs
Source
Monitoring
Execute Workflow
ganpati
Zoom Out
15 minutes ago to a few seconds ago refreshed every 5s
CPU Utilization - System
15.0
12.5
10.0
7.5
5.0
2.5
0
12:25
12:30
12:35
bind_host_49e6c CPU System
bono_host_7a5f1 CPU System
ellis_host_6706f CPU System
homer_host_30fec CPU System
homestead_host_883a1 CPU System
sprout_host_1d8b2 CPU System
sprout_host_60841 CPU System
CPU Utilization - User
40
30
20
10
0
12:25
12:30
12:35
bind_host_49e6c CPU User
bono_host_7a5f1 CPU User
ellis_host_6706f CPU User
homer_host_30fec CPU User
homestead_host_883a1 CPU User
sprout_host_1d8b2 CPU User
sprout_host_60841 CPU User
CPU UTILLIZATION & DISK IO
Network IO - RX
1.5 K
1.3 K
1.0 K
750
500
Network IO - TX
2.0 K
1.5 K
1.0 K
500

2. Clearwater live voice and video call

These instructions will take you through the process of making a call on a Clearwater deployment.

1. Download and install two SIP clients (X-Lite http://www.counterpath.com/x-lite/ or Bria 4 http://www.counterpath.com/bria/)
2. Signup to the Ellis URL using associate IP address of the ellis node with signup code as 'secret' and your information



10.115.22.246/signup.html

Clearwater

Sign up

Please enter your details

Your name:

This will be your name in the address book

Email:

This will be the email address you log in with

Password:

At least 8 characters, common passwords like "password", "1234" and "qwerty" are not allowed.

Confirm Password:

Confirm the password you entered above.

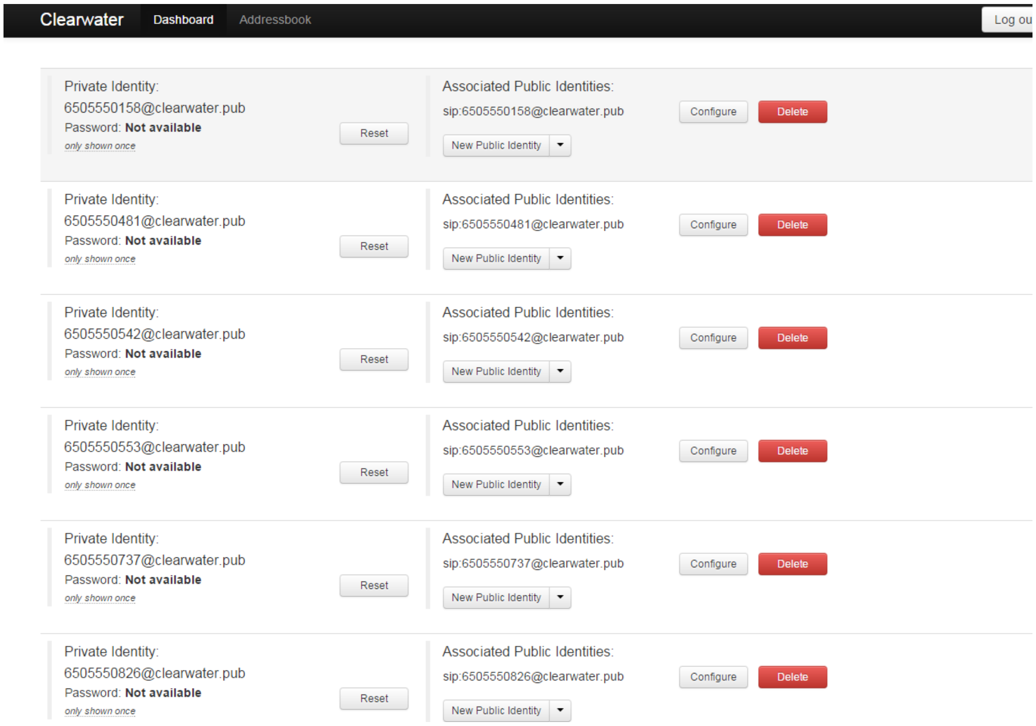
☑ Demo account?

Demo accounts automatically expire after one week.

Signup code:

You should have been sent a signup-code when invited to the beta

3. Generate few SIP numbers for registering SIP clients to the Bono node. When the number is created save the password as it will be used during SIP registration on Clearwater



Private Identity:
6505550158@clearwater.pub
Password: **Not available**
only shown once

Associated Public Identities:
sip:6505550158@clearwater.pub

Private Identity:
6505550481@clearwater.pub
Password: **Not available**
only shown once

Associated Public Identities:
sip:6505550481@clearwater.pub

Private Identity:
6505550542@clearwater.pub
Password: **Not available**
only shown once

Associated Public Identities:
sip:6505550542@clearwater.pub

Private Identity:
6505550553@clearwater.pub
Password: **Not available**
only shown once

Associated Public Identities:
sip:6505550553@clearwater.pub

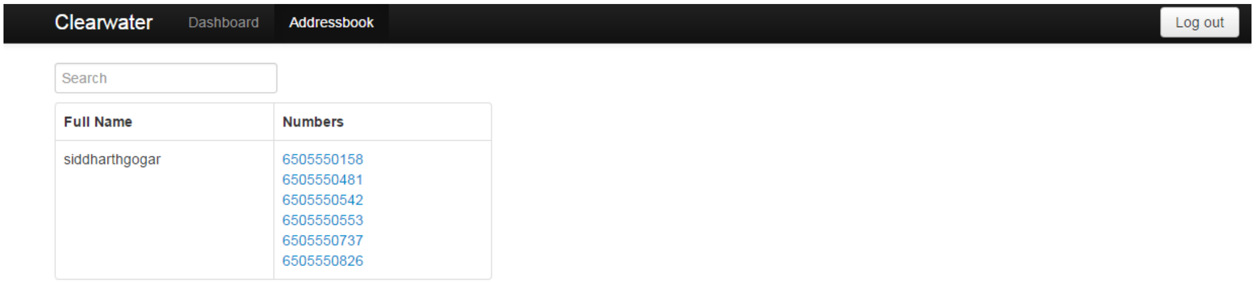
Private Identity:
6505550737@clearwater.pub
Password: **Not available**
only shown once

Associated Public Identities:
sip:6505550737@clearwater.pub

Private Identity:
6505550826@clearwater.pub
Password: **Not available**
only shown once

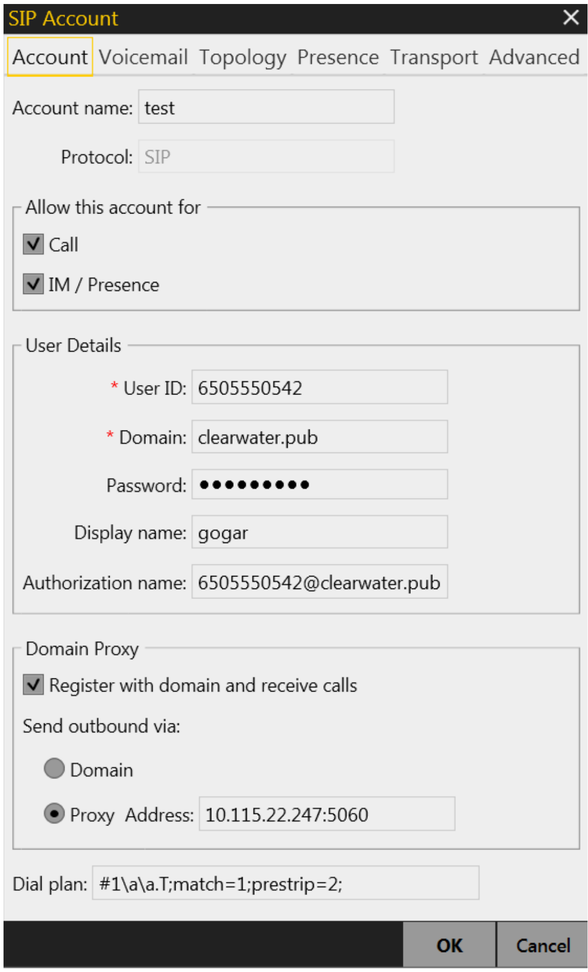
Associated Public Identities:
sip:6505550826@clearwater.pub

4. Confirm that numbers are available in the Ellis Addressbook



Full Name	Numbers
siddharthgogar	6505550158 6505550481 6505550542 6505550553 6505550737 6505550826

5. Register the numbers using SIP clients X-Lite and Bria-4. Use the numbers created in step 3 on ellis for parameters. The proxy address is public IP address of Bono node with TCP port 5060



Account name: test

Protocol: SIP

Allow this account for

☑ Call

☑ IM / Presence

User Details

* User ID: 6505550542

* Domain: clearwater.pub

Password: ●●●●●●●●●

Display name: gogar

Authorization name: 6505550542@clearwater.pub

Domain Proxy

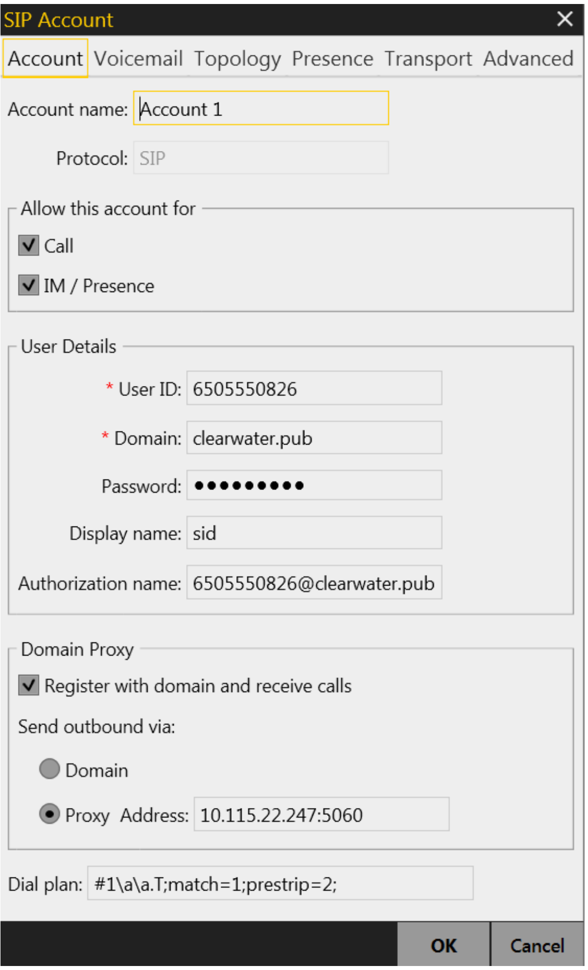
☑ Register with domain and receive calls

Send outbound via:

○ Domain

● Proxy Address: 10.115.22.247:5060

Dial plan: #1\a\a.T;match=1;prestrip=2;



Account name: Account 1

Protocol: SIP

Allow this account for

☑ Call

☑ IM / Presence

User Details

* User ID: 6505550826

* Domain: clearwater.pub

Password: ●●●●●●●●●

Display name: sid

Authorization name: 6505550826@clearwater.pub

Domain Proxy

☑ Register with domain and receive calls

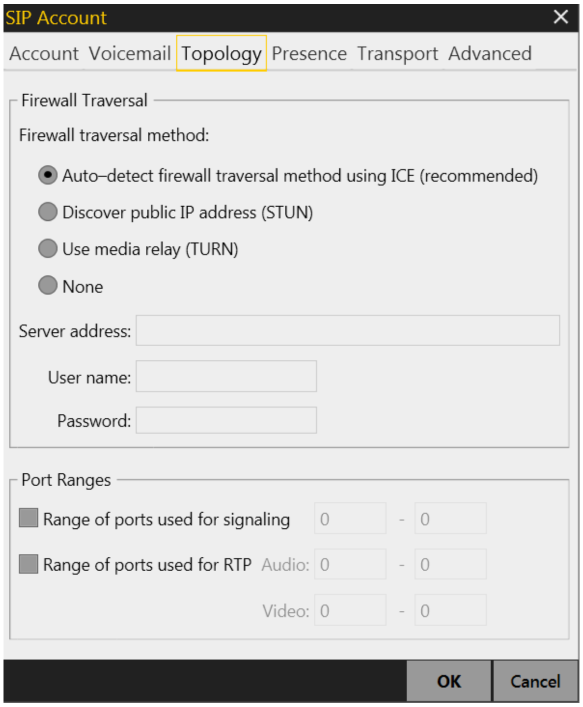
Send outbound via:

○ Domain

● Proxy Address: 10.115.22.247:5060

Dial plan: #1\a\a.T;match=1;prestrip=2;

6. Go under settings tab on the SIP clients and select *'Auto-detect firewall traversal method using ICE'*



SIP Account

Account Voicemail Topology Presence Transport Advanced

Firewall Traversal

Firewall traversal method:

● Auto–detect firewall traversal method using ICE (recommended)

○ Discover public IP address (STUN)

○ Use media relay (TURN)

○ None

Server address:

User name:

Password:

Port Ranges

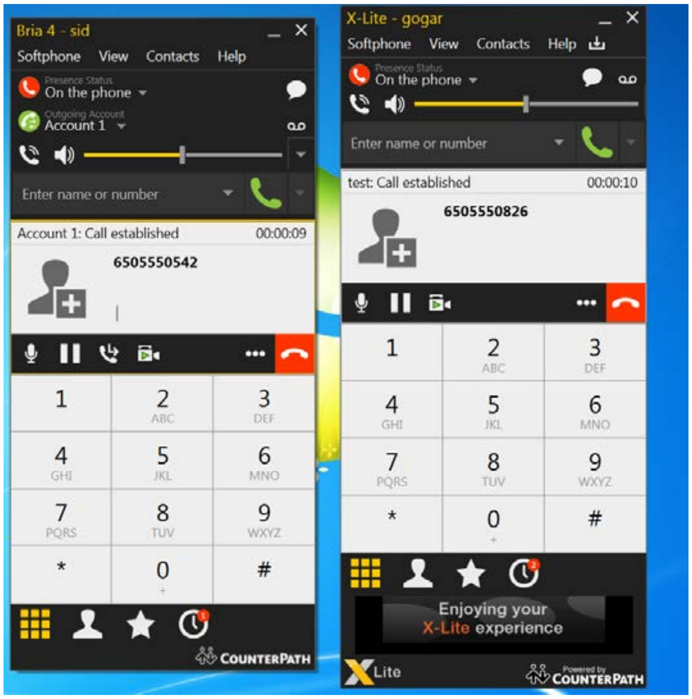
□ Range of ports used for signaling 0 - 0

□ Range of ports used for RTP Audio: 0 - 0

Video: 0 - 0

OK Cancel

7. Make a voice or video call using the two SIP clients



Bria 4 - sid

Account 1: Call established 00:00:09

6505550542

X-Lite - gogar

test: Call established 00:00:10

6505550826