

# Cloud Computing

SS17 | Assignment 2



## **Group 2**

Yan Sun

Yuchun Chen

Seema Narasimha Swamy

Julie Allard

Navya Basavaraju

## **Task 1: Cloud Benchmark**

- Detailed description of your cloud benchmarking methodology, including any scripts or other code

**Abstract** This test plan aims to provide benchmarking results for different scenarios covering the openstack cloud API operations.

### **Test Plan**

---

This test plan describes test cases that can cover some of the basic cloud operations e.g. Booting VMs and Listing VMs.

## **1. Test Environment**

### **1.1. Preparation**

---

This test plan is performed against existing OpenStack cloud by executing via Rally from very beginning including deployment of the OpenStack cloud.

### **1.2. Environment description**

The environment description includes hardware specification of servers, network parameters, operation system and OpenStack deployment characteristics.

### **Hardware**

---

This section contains list of all types of hardware nodes.

Parameter	Value	Comments
Size		290.3nMB
Container Format		BARE
Min Disk		3GB
Disk Format		QCOW2
Min RAM		512 MB

## Network

---

This section contains list of interfaces and network parameters. For complicated cases this section may include topology diagram and switch parameters.

Parameter	Value	Comments
Network Name	cc17-net	
MTU	1450	
ID	9b345a37-c64d-46b0-9286-d3a8c453e412	
Network ID	64a7d16e-91b9-4ab4-a1fb-8a48368e1fdb	
Subnet pool	None	
IP version	IPv4	
CIDR	10.12.1.0/24	
IP allocation pool	Start 10.12.1.2 - End 10.12.1.254	
Gateway IP	10.12.1.1	
DHCP Enable	Yes	
Additional routes	None	
DNS name server	None	

---

## Software

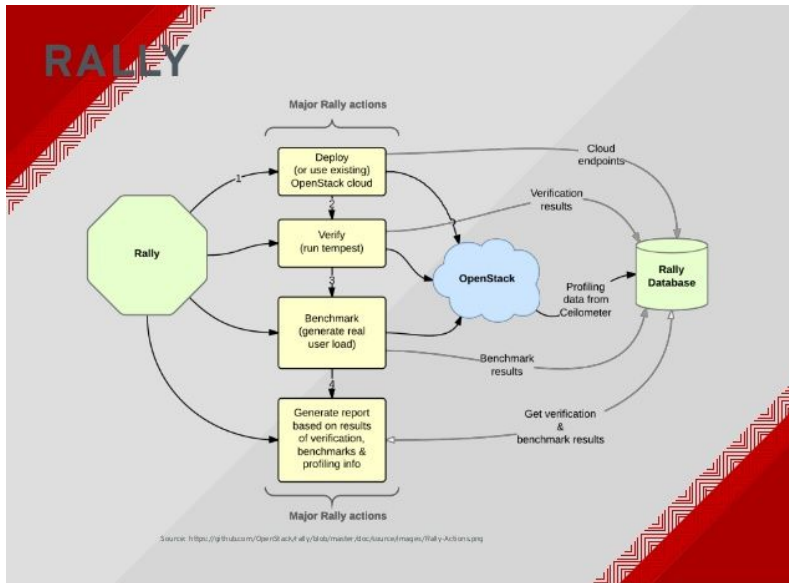
This section describes installed software.

Parameter	Value	Comments
OS	ubuntu-16.04	

## 2. Test tool

---

**Rally** is a benchmarking tool that was designed specifically for OpenStack API testing. To make this possible, **Rally** automates and unifies multi-node OpenStack deployment, cloud verification, benchmarking & profiling. This is a simple way to check cloud workability and performance of control plane operations running on it.



### 3. Test Case 1: Creation and Deletion of VMs

#### 3.1. Description

The most user-facing control plane operation is new virtual machine creation. We used NovaServers.boot\_and\_delete\_server to evaluate this feature. The boot-and-delete.json is the script used. We measured the performance 3 times in a day(Morning, Afternoon, Night).

#### 3.2. Parameters

Name	Description
IMAGE	Image from which boot server
FLAVOR	Flavor type from which boot server
CONCURRENCY	Amount of parallel executors
ITERATIONS	Total amount of iterations processed by all executors
NETWORK	Start CIDR
NIC	The network ID to which servers need to be connected to

#### 3.3. List of performance metrics

Priority	Value	Measurement Units	Description
1		sec	Response time

### 3.4. Measuring performance values

1. Create server with FLAVOR flavor from IMAGE image through Nova API
2. Delete server through Nova API.

This 2 steps executed in 10 cycles. One cycle of this 3 steps is called as iteration. ITERATIONS is a total amount of iterations which was processed by executors.

Using the arguments flavor as “cloud computing”, image as "ubuntu-16.04"

“ and specifying network and nics, we managed to create a network and delete it. We executed the script 10 times to get better accuracy.

## 4. Benchmark Script

```
{
```

```
"NovaServers.boot_and_delete_server": [
```

```
{
```

```
  "args": {
```

```
    "flavor": {
```

```
      "name": "Cloud Computing"
```

```
    },
```

```
    "image": {
```

```
      "name": "ubuntu-16.04"
```

```
    },
```

```
    "network": {
```

```
      "start_cidr": "10.12.1.0/24"
```

```
    },
```

```
    "nics":[{"net-id":"64a7d16e-91b9-4ab4-a1fb-8a48368e1fdb"}],
```

```
    "force_delete": false
```

```
  },
```

```
  "runner": {
```

```
"type": "constant",
"times": 10,
"concurrency": 1
},
"context": {
}
]
}
```

## 5. Test Case 2: Listing Servers

### 5.1. Description

To check whether OpenStack is going to work well on, say, a large number of server installation under high load, we used NovaServers.list\_servers to list the servers. We collected results from different times a day(Morning, Afternoon, Night).

### 5.2. Parameters

Name	Description
IMAGE	Image to upload to glance
CONCURRENCY	Amount of parallel executors
ITERATIONS	Total amount of iterations processed by all executors
NETWORK	Start CIDR
NIC	The network ID to which servers need to be connected to

### 5.3. List of performance metrics

Priority	Value	Measurement Units	Description
1		sec	Response time

### 5.4. Measuring performance values

1. Create server with FLAVOR flavor from IMAGE image through Nova API
2. List all the servers created in step 1

### 3. Delete the servers through Nova API.

This 3 steps executed in 10 cycles. One cycle of this 3 steps is called as iteration. ITERATIONS is a total amount of iterations which was processed by executors.

Using the arguments flavor as "cloud computing", image as "ubuntu-16.04"

and specifying network and nics, we managed to create a network and delete it. We executed the script 10 times to get better accuracy.

This will list the servers in the openstack and the test case is executed multiple times to avoid measurement errors

## 6. Benchmark Script

---

```
{
  "NovaServers.list_servers": [
    {
      "runner": {
        "type": "constant",
        "concurrency": 1,
        "times": 10
      },
      "args": {
        "detailed": true,
      },
    },
    {
      "context": {
        "servers": {
          "servers_per_tenant": 2,
          "flavor": {
            "name": "Cloud Computing"
          },
          "image": {
            "name": "ubuntu-16.04"
          },
        },
      },
    },
  ],
}
```

```

    "nics": [
        "64a7d16e-91b9-4ab4-a1fb-8a48368e1fdb"
    ]
}
}

```

## Benchmark Execution

Benchmarking results of the six different combinations of scenarios and time slots, including plots and interpretation of the results

### • Activate Rally

**/Users/yuchunchen/rally/bin/activate**

Information about your Rally installation:

- \* Method: **virtualenv**
- \* Virtual Environment at: **/Users/yuchunchen/rally**
- \* Database at: **/Users/yuchunchen/rally/database**
- \* Configuration file at: **/Users/yuchunchen/rally/etc/rally**
- \* Samples at: **/Users/yuchunchen/rally/samples**

### • Loading the cc17-group02 project context to the terminal

(rally)~\$ **source cc17-group02-openrc.sh**

Please enter your OpenStack Password:

### • Registering an OpenStack deployment in Rally

### • Register a deployment with the deployment createcommand

(rally)~\$ **rally deployment create --fromenv --name=existing**

Error:~ Deployment name 'existing' already registered.

(rally)~\$ **rally deployment create --fromenv --name=deployment**

2017-06-27 12:51:06.714 11962 INFO rally.deployment.engines.existing [-] Save deployment 'deployment' (uuid=60b29d45-b65c-4554-8458-94f12c309c72) with 'openstack' platform.

uuid	created_at	name	status	active
60b29d45-b65c-4554-8458-94f12c309c72	2017-06-27T10:51:06	deployment	deploy->finished	

Using deployment: 60b29d45-b65c-4554-8458-94f12c309c72  
~/.rally/openrc was updated

### • List all deployments

(rally)~\$ **rally deployment list**

uuid	created_at	name	status	active
------	------------	------	--------	--------



```
| 5c605c05-f936-4c06-b768-f424caab5238 | 2017-06-27T10:48:13 | existing | deploy->finished
|
| 60b29d45-b65c-4554-8458-94f12c309c72 | 2017-06-27T10:51:06 | deployment |
deploy->finished | * |
```

```
+-----+-----+-----+-----+-----+
```

- To customize and manage flavors

```
(rally)~$ openstack flavor list
```

```
+-----+-----+-----+-----+-----+-----+-----+
| ID | Name | RAM | Disk | Ephemeral | VCPUs | Is Public |
+-----+-----+-----+-----+-----+-----+-----+
| 610f44b0-d25a-44bc-a6b1-8b22e68675e5 | Cloud Computing | 512 | 10 | 0 | 1 | False
|
```

```
+-----+-----+-----+-----+-----+-----+-----+
```

- Put the information about your cloud credentials into a JSON configuration file (It is called existing.json)

```
(rally)~$ cd Desktop/
```

```
(rally)~$ rally deployment create --file=existing.json --name=existing
```

```
2017-06-27 12:54:26.856 11993 INFO rally.deployment.engines.existing [-] Save deployment
'existing' (uuid=7c5d3601-c4f1-4934-afd9-821623914fab) with 'openstack' platform.
```

```
+-----+-----+-----+-----+-----+-----+-----+
```

```
| uuid | created_at | name | status | active |
+-----+-----+-----+-----+-----+-----+
| 7c5d3601-c4f1-4934-afd9-821623914fab | 2017-06-27T10:54:26 | existing | deploy->finished |
|
```

```
+-----+-----+-----+-----+-----+-----+-----+
```

```
Using deployment: 7c5d3601-c4f1-4934-afd9-821623914fab
```

```
~/.rally/openrc was updated
```

HINTS:

\* To use standard OpenStack clients, set up your env by running:

```
source ~/.rally/openrc
```

OpenStack clients are now configured, e.g run:

```
openstack image list
```

```
(rally)~$ rally deployment list
```

```
+-----+-----+-----+-----+-----+-----+-----+
```

```
| uuid | created_at | name | status | active |
+-----+-----+-----+-----+-----+-----+
| 7c5d3601-c4f1-4934-afd9-821623914fab | 2017-06-27T10:54:26 | existing | deploy->finished |
* |
```

```
+-----+-----+-----+-----+-----+-----+-----+
```

- Enables to verify that our current deployment is healthy and ready to be benchmarked.

```
(rally)~$ rally deployment check
```

```
-----
```

```
Platform openstack:
```

```
-----
```

Available services:

Service	Service Type	Status
__unknown__	volumev2	Available
ceilometer	metering	Available
cinder	volume	Available
cloud	cloudformation	Available
glance	image	Available
heat	orchestration	Available
keystone	identity	Available
neutron	network	Available
nova	compute	Available
swift	object-store	Available

- To run the benchmark scenarios : to benchmark the performance of a sequence of only two simple operations-it first boots a server (with customizable parameters) and then deletes it.

(rally)~\$ rally task start boot-and-delete.json

Preparing input task

Task is:

```
{
  "NovaServers.boot_and_delete_server": [
    {
      "args": {
        "flavor": {
          "name": "Cloud Computing"
        },
        "image": {
          "name": "ubuntu-16.04"
        },
        "network": {
          "start_cidr": "10.12.1.0/24"
        },
        "nics":[{"net-id":"64a7d16e-91b9-4ab4-a1fb-8a48368e1fdb"}],
        "force_delete": false
      },
      "runner": {
        "type": "constant",
        "times": 10,
        "concurrency": 1
      },
      "context": {
      }
    }
  ]
}
```

}

Task syntax is correct :)  
Running Rally version 0.9.1~dev379

Task 680d3c94-14e6-41df-82cc-5615275934f0: started

Benchmarking... This can take a while...

### Morning readings:

Task 680d3c94-14e6-41df-82cc-5615275934f0 has 0 error(s)

Response Times (sec)								
Action	Min (sec)	Median (sec)	90%ile (sec)	95%ile (sec)	Max (sec)	Avg (sec)	Success	Count
nova.boot_server	12.636	15.589	17.807	18.092	18.377	15.656	100.0%	10
nova.delete_server	2.717	3.082	4.965	5.206	5.447	3.569	100.0%	10
total	14.453	17.892	20.006	21.054	22.101	18.225	100.0%	10

Load duration: 191.286812  
Full duration: 195.673357

### Afternoon readings:

Task d11c7eb6-6466-4ec9-b383-478a1ee3d2f4 has 0 error(s)

Response Times (sec)								
Action	Min (sec)	Median (sec)	90%ile (sec)	95%ile (sec)	Max (sec)	Avg (sec)	Success	Count
nova.list_servers	0.935	0.97	1.249	1.251	1.253	1.026	100.0%	10
total	0.935	0.97	1.249	1.251	1.253	1.026	100.0%	10

Load duration: 10.298575  
Full duration: 37.680844

### Night readings:

Task fada9e96-3aaa-4de3-8325-807d3068c49e has 0 error(s)

Response Times (sec)								

Action	Min (sec)	Median (sec)	90%ile (sec)	95%ile (sec)	Max (sec)	Avg (sec)	Success	Count
nova.list_servers	0.727	0.767	0.83	0.831	0.831	0.774	100.0%	10
total	0.727	0.767	0.83	0.831	0.831	0.774	100.0%	10

Load duration: 7.77622

Full duration: 35.385257

• To run the benchmark scenarios :Listing all the servers on the sourced openstack project.  
(rally)~\$ rally task start nova\_list\_servers.json

Preparing input task

Task is:

```
{
  "NovaServers.list_servers": [
    {
      "runner": {
        "type": "constant",
        "concurrency": 1,
        "times": 10
      },
      "args": {
        "detailed": true,
      },
    },
    "context": {
      "servers": {
        "servers_per_tenant": 2,
        "flavor": {
          "name": "Cloud Computing"
        },
        "image": {
          "name": "ubuntu-16.04"
        },
        "nics": [
          "64a7d16e-91b9-4ab4-a1fb-8a48368e1fdb"
        ]
      }
    }
  ]
}
```

Task syntax is correct :)  
Running Rally version 0.9.1~dev379

Task d078bbdd-13f4-4bde-8a58-007f01ebdc64: started

Benchmarking... This can take a while...

### Morning readings:

Task 1ac9292b-8d27-48ff-a397-516194e15e9e has 0 error(s)

Response Times (sec)								
Action	Min (sec)	Median (sec)	90%ile (sec)	95%ile (sec)	Max (sec)	Avg (sec)	Success	Count
nova.list_servers	0.922	1.146	1.401	1.433	1.465	1.176	100.0%	10
total	0.922	1.146	1.401	1.433	1.465	1.176	100.0%	10

Load duration: 11.790946

Full duration: 43.834879

### Afternoon readings :

Task d078bbdd-13f4-4bde-8a58-007f01ebdc64 has 0 error(s)

Response Times (sec)								
Action	Min (sec)	Median (sec)	90%ile (sec)	95%ile (sec)	Max (sec)	Avg (sec)	Success	Count
nova.boot_server	14.516	14.771	14.978	15.172	15.367	14.794	100.0%	10
nova.delete_server	2.819	2.925	3.614	4.366	5.118	3.197	100.0%	10
total	16.346	16.806	17.361	18.147	18.934	16.992	100.0%	10

Load duration: 178.950542

Full duration: 183.231138

### Night readings:

Task f883997d-08f5-4881-bc76-a465bb2db008 has 0 error(s)

Response Times (sec)								
Action	Min (sec)	Median (sec)	90%ile (sec)	95%ile (sec)	Max (sec)	Avg (sec)	Success	Count
nova.boot_server	13.959	15.09	17.229	17.497	17.764	15.33	100.0%	10
nova.delete_server	2.821	5.045	5.463	5.61	5.757	4.47	100.0%	10
total	15.78	19.105	20.015	21.268	22.522	18.8	100.0%	10

Load duration: 197.036513

Full duration: 201.218683

**Note:**For the Plots , the benchmark result report as html files are uploaded along with the solution files

[BootServerMorningRunReport.html](#),[BootServerAfternoonRunReport.html](#),[BootServerNightRunReport.html](#)

[ListServersMorningRunReport.html](#), [ListServersMorningRunReport.html](#),  
[ListServersMorningRunReport](#)

### Interpretation of the Results:

With the list servers, the execution of the test was successful and about how the values have varied - Load duration: 10.298575. Full duration: 37.680844 in different times of day it is the best at night and we can guess that load on server was less in the night

## **Task 2: Introduction Heat**

### • List flavors of sourced openstack.

\$ openstack flavor list

ID	Name	RAM	Disk	Ephemeral	VCPUs	Is Public
604de11c-32	Cloud	512	10	0	1	False
22-4902-852	Computing					
3-11cc61b5b						
485						

\$ openstack zone list

openstack: 'zone list' is not an openstack command. See 'openstack --help'.

Did you mean one of these?

token issue

token revoke

### • List the availability zone list.

~\$ openstack availability zone list

Zone Name	Zone Status
Default	available
Cloud Computing 2017	available
nova	available
nova	available
nova	available

- The command is to create a stack, or template, from an example template file.

```
~$ openstack stack create --template server.yml --parameter
"name=CCG2_First_Stack;key_pair=cloudgroup02;flavor=604de11c-3222-4902-8523-11cc
61b5b485;image=11f6b8aa-31df-4b66-8b42-5ee9760c47ba;network=64a7d16e-91b9-4ab4-
a1fb-8a48368e1fdb;zone=Cloud Computing
2017;security_groups=a87e65e1-f959-4255-9b40-cafe1d9db47c" CCG2_First_Stack
```

Field	Value
id	8e83ff4d-ac43-4eda-880c-c0285a119f00
stack_name	CCG2_First_Stack
description	One VM instance
creation_time	2017-06-22T12:41:40
updated_time	None
stack_status	CREATE_IN_PROGRESS
stack_status_reason	

- The command is to see which stacks are visible to the current user.

```
wlan-141-23-129-66:~$ openstack stack list
```

ID	Stack Name	Stack Status	Creation Time	Updated Time
8e83ff4d-ac43-4eda-880c-c0285a119f00	CCG2_First_St	CREATE_COMPLE	2017-06-22T12	None

- List all project instances with which a floating IP address could be associated.

```
wlan-141-23-129-66:~$ openstack server list
```

ID	Name	Status	Networks	Image Name
e04034d7-4336-4	CCG2_First_Stac	ACTIVE	cc17-net=10.12.1	ubuntu-16.04
3e3-9424-39f315	k		.110	
592461				

- To list all floating IP addresses that are allocated to our project.

wlan-141-23-129-66:~ \$ openstack floating ip list

ID	Floating IP Address	Fixed IP Address	Port	Floating Network	Project
4f7b88c1	10.200.2.199	None	None	25fe17bd-c5a2-44	4894c882bfcc4
-47fe-4f47-				c6-af49-fe93aad0	558ab515cdc55
8523-a24e6d			66b7	b32446	
6415b4					

- Associate an IP address with an instance in the project.

~\$ openstack server add floating ip CCG2\_First\_Stack 10.200.2.199

- Try to login to the server via SSH and floating point IP.

~\$ ssh -i cloudgroup2.key ubuntu@10.200.2.199

The authenticity of host '10.200.2.199 (10.200.2.199)' can't be established.

ECDSA key fingerprint is SHA256:SWIYznx7vS7hEqWLk3IE7rrluEY5xHM8epZ8mXYMMZg.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '10.200.2.199' (ECDSA) to the list of known hosts.

Saving password to keychain failed

Identity added: cloudgroup2.key ((null))

Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-22-generic x86\_64)

\* Documentation: <https://help.ubuntu.com/>

Get cloud support with Ubuntu Advantage Cloud Guest:

<http://www.ubuntu.com/business/services/cloud>

0 packages can be updated.

0 updates are security updates.

The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/\*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".

See "man sudo\_root" for details.

- It shows that it login to the server.

ubuntu@ccg2-first-stack:~\$



```
yuchunchen — ubuntu@ccg2-first-stack: ~ — ssh -i cloudgroup2.key ubuntu...
Identity added: cloudgroup2.key ((null))
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-22-generic x86_64)

* Documentation:  https://help.ubuntu.com/

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ccg2-first-stack:~$
```

• Carry out suspend stack functions on behalf of your users.

~\$ openstack stack suspend CCG2\_First\_Stack

ID	Stack Name	Stack Status	Creation Time	Updated Time
8e83ff4d-ac43-4eda-880c-c0285a119f00	CCG2_First_St	SUSPEND_IN_PR	2017-06-22T12:41:40	None

wlan-141-23-129-66:~\$ openstack stack show CCG2\_First\_Stack

Field	Value
id	8e83ff4d-ac43-4eda-880c-c0285a119f00
stack_name	CCG2_First_Stack
description	One VM instance
creation_time	2017-06-22T12:41:40
updated_time	None
stack_status	RESUME_COMPLETE
stack_status_reason	Stack RESUME completed successfully
parameters	OS::project_id: 4894c882bfcc4558ab515cdc55b32446
	OS::stack_id: 8e83ff4d-ac43-4eda-880c-c0285a119f00

```

| OS::stack_name: CCG2_First_Stack |
| flavor: 604de11c-3222-4902-8523-11cc61b5b485 |
| image: 11f6b8aa-31df-4b66-8b42-5ee9760c47ba |
| key_pair: cloudgroup02 |
| name: CCG2_First_Stack |
| network: 64a7d16e-91b9-4ab4-a1fb-8a48368e1fdb |
| security_groups: |
| '[u"a87e65e1-f959-4255-9b40-cafe1d9db47c"]' |
| zone: Cloud Computing 2017 |
|
|
| outputs | - description: Virtual IP address of instance |
| | output_key: ip |
| | output_value: 10.12.1.110 |
| | - description: Port ID of instance |
| | output_key: port |
| | output_value: b14ff905-a281-41da-8e18-d1628df94928 |
| |
| links | - href: http://cloud.cit.tu-berlin.de:8004/v1/4894c8 |
| | 82bfcc4558ab515cdc55b32446/stacks/CCG2_First_Stack |
| | /8e83ff4d-ac43-4eda-880c-c0285a119f00 |
| | rel: self |
|
|
| disable_rollback | True |
| parent | None |
| tags | None |
| stack_user_project_id | 192b7bf309984da28923324661121759 |
| capabilities | [] |
| notification_topics | [] |
| timeout_mins | None |
| stack_owner | None |
+-----+-----+

```

- Carry out delete stack functions on behalf of your users.

**\$ openstack stack delete CCG2\_First\_Stack**

Are you sure you want to delete this stack(s) [y/N]? y

wlan-141-23-129-66:~\$ openstack stack show CCG2\_First\_Stack

Stack not found: CCG2\_First\_Stack

- List instances, check status of instance.

**~\$openstack server list**

~\$ ...

- The contents of your server-landscape.yml file:

(server-landscape.yml file is uploaded along with the solution pdf)

*Note: The values for parameters have been defaulted for simplicity sake but it could be also made dynamic by removing defaulted values and specifying as parameters during the stack create operation. We would be taking care of it if required for upcoming Assignment 3*

heat\_template\_version: 2015-10-15

description: Three VM instances

parameters:

name:

type: string

label: Name of the frontend VM

default: Frontend Server

key\_pair:

type: string

label: Key Pair

default: cloudgroup02

constraints:

- custom\_constraint: nova.keypair

flavor:

type: string

label: Flavor

default: 610f44b0-d25a-44bc-a6b1-8b22e68675e5

constraints:

- custom\_constraint: nova.flavor

image:

type: string

label: Image Name

default: 11f6b8aa-31df-4b66-8b42-5ee9760c47ba

constraints:

- custom\_constraint: glance.image

zone:

type: string

label: Availability Zone

default: Cloud Computing 2017

public\_net:

type: string

description: ID or name of public network for which floating IP addresses will be allocated

default: tu-internal

private\_net\_name:

type: string  
description: Name of private network to be created  
default: cc17G2-privnet

private\_subnet\_name:  
type: string  
description: Name of private sub network to be created  
default: cc17G2-subnet

private\_net\_cidr:  
type: string  
description: Private network address (CIDR notation)  
default: 10.2.1.0/24

private\_net\_gateway:  
type: string  
description: Private network gateway address  
default: 10.2.1.1

resources:

# This port is a separate resource used to assign the security groups  
# to the VM. Can also be used to attach a OS::Neutron::FloatingIP to the VM.

private\_net:  
type: OS::Neutron::Net  
properties:  
name: { get\_param: private\_net\_name }

private\_subnet:  
type: OS::Neutron::Subnet  
properties:  
name: { get\_param: private\_subnet\_name }  
network\_id: { get\_resource: private\_net }  
cidr: { get\_param: private\_net\_cidr }  
gateway\_ip: { get\_param: private\_net\_gateway }

router:  
type: OS::Neutron::Router  
properties:  
#name: { get\_param: router\_name }  
external\_gateway\_info:  
network: { get\_param: public\_net }

```
router_interface:
  type: OS::Neutron::RouterInterface
  properties:
    router: { get_resource: router }
    subnet: { get_resource: private_subnet }
```

```
custom_security_group:
  type: OS::Neutron::SecurityGroup
  properties:
    name: cc17g2_security_group
    rules:
      - protocol: tcp
        port_range_min: 80
        port_range_max: 80
      - protocol: tcp
        port_range_min: 22
        port_range_max: 22
```

```
backend_instances:
  type: OS::Heat::ResourceGroup
  properties:
    count: 2
    resource_def:
      type: server.yaml
      properties:
        network: { get_attr: [private_subnet, network_id] }
        security_groups: [{ get_resource: custom_security_group }]
        name: backend_server_%index%
        key_pair: { get_param: key_pair }
        image: { get_param: image }
        flavor: { get_param: flavor }
        zone: { get_param: zone }
```

#

```
frontend_instance:
  type: server.yaml
  properties:
    network: { get_attr: [private_subnet, network_id] }
    security_groups: [{ get_resource: custom_security_group }]
    name: { get_param: name }
    key_pair: { get_param: key_pair }
```

```

image: { get_param: image }
flavor: { get_param: flavor }
zone: { get_param: zone }

```

```

fp_ip:
  type: OS::Neutron::FloatingIP
  properties:
    floating_network: tu-internal

```

```

fp_association:
  type: OS::Neutron::FloatingIPAssociation
  properties:
#    floatingip_id: 4f7b88c1-47fe-4f47-8523-a24e6d6415b4
    floatingip_id: { get_resource: fp_ip }
    port_id: { get_attr: [frontend_instance,port]}

```

outputs:

```

floating_ip:
  description: Floating point IP address of instance
  value: { get_attr: [ fp_ip, floating_ip_address] }

```

### **Task3: Advanced Heat Templates**

- The command is to create a stack, or template, from an [server-landscape.yml](#).

```
wlan-141-23-169-80:~ navsie$ openstack stack create --template server-landscape.yml
CCG2_Network_Instances_Stack
```

```

+-----+-----+
| Field          | Value                                     |
+-----+-----+
| id             | 05d03a98-65a9-40d2-86ba-b8088ef169da |
| stack_name     | CCG2_Network_Instances_Stack         |
| description    | Three VM instances                   |
| creation_time  | 2017-07-02T17:17:54                 |
| updated_time   | None                                 |
| stack_status   | CREATE_IN_PROGRESS                   |
| stack_status_reason |                                         |
+-----+-----+

```

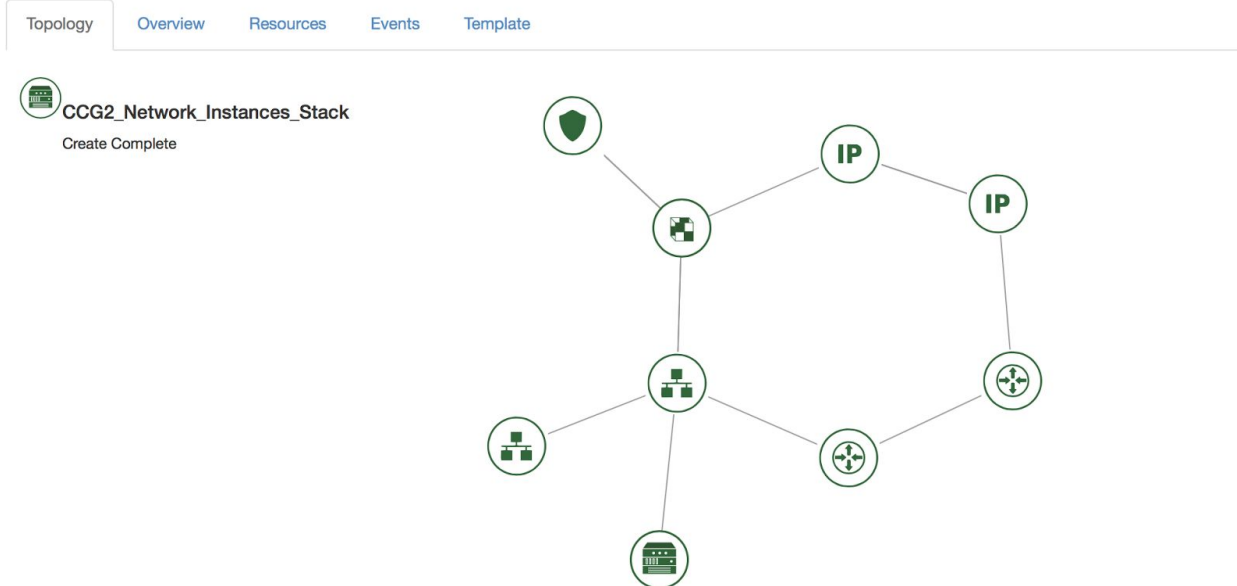
- Show an appropriate quantity of data by default.

```
wlan-141-23-169-80:~ navsie$ openstack stack output show CCG2_Network_Instances_Stack
Either <OUTPUT NAME> or --all must be specified.
```

```
wlan-141-23-169-80:~ navsie$ openstack stack output show CCG2_Network_Instances_Stack
floating_ip -f value
Floating point IP address of instance
floating_ip
10.200.1.121
```

```
wlan-141-23-169-80:~ navsie$ openstack stack create --template server-landscape.yml CCG2_Network_Instances_Stack
+-----+
| Field | Value |
+-----+
| id | 05d03a98-65a9-40d2-86ba-b8088ef169da |
| stack_name | CCG2_Network_Instances_Stack |
| description | Three VM instances |
| creation_time | 2017-07-02T17:17:54 |
| updated_time | None |
| stack_status | CREATE_IN_PROGRESS |
| stack_status_reason | |
+-----+
wlan-141-23-169-80:~ navsie$ openstack stack output show CCG2_Network_Instances_Stack
Either <OUTPUT NAME> or --all must be specified.
wlan-141-23-169-80:~ navsie$ openstack stack output show CCG2_Network_Instances_Stack floating_ip -f value
Floating point IP address of instance
floating_ip
10.200.1.121
```

## Stacks / CCG2\_Network\_Instances\_Stack



### • List instances, check status of instance.

~ \$ openstack server list

```
wlan-141-23-169-80:~ navsie$ openstack server list
+-----+-----+-----+-----+-----+
| ID | Name | Status | Networks | Image Name |
+-----+-----+-----+-----+-----+
| fb5d3b92-f4db-481f-aeca-22966d2d4119 | backend_server_0 | ACTIVE | cc17G2-privnet=10.2.1.5 | ubuntu-16.04 |
| b6f8d9dd-883a-4dbf-af7b-f0d9e9d8edea | backend_server_1 | ACTIVE | cc17G2-privnet=10.2.1.4 | ubuntu-16.04 |
| 8348e79d-754b-4c20-9c9a-4d674ecd61c5 | Frontend Server | ACTIVE | cc17G2-privnet=10.2.1.3, 10.200.1.121 | ubuntu-16.04 |
+-----+-----+-----+-----+-----+
wlan-141-23-169-80:~ navsie$
```

### • Try to login to the server via SSH and floating point IP.



```
~$ ssh -i cloudgroup2.key ubuntu@10.200.1.121
```

```
wlan-141-23-169-80:~ navsie$ ssh -i cloudgroup2.key ubuntu@10.200.1.121
The authenticity of host '10.200.1.121 (10.200.1.121)' can't be established.
ECDSA key fingerprint is SHA256:D78WXVsxBid/3fogkgJHKthrp88wI10jOG/0T1V6j0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.200.1.121' (ECDSA) to the list of known hosts.
[Enter passphrase for key 'cloudgroup2.key':
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-22-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Get cloud support with Ubuntu Advantage Cloud Guest:
  http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@frontend-server:~$
```

- **Testing internet connectivity in the frontend server.**

```
ubuntu@frontend-server:~$ git clone https://github.com/julieallard/TUBCloudComputingGroup2.git
Cloning into 'TUBCloudComputingGroup2'...
Username for 'https://github.com': navyatub
Password for 'https://navyatub@github.com':
remote: Counting objects: 89, done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 89 (delta 0), reused 4 (delta 0), pack-reused 84
Unpacking objects: 100% (89/89), done.
Checking connectivity... done.
ubuntu@frontend-server:~$ ls
TUBCloudComputingGroup2
ubuntu@frontend-server:~/TUBCloudComputingGroup2$ cp cloudgroup2.key cloudgroup2.key.pub /home/ubuntu
ubuntu@frontend-server:~/TUBCloudComputingGroup2$ cd ..
ubuntu@frontend-server:~$ ls
cloudgroup2.key cloudgroup2.key.pub TUBCloudComputingGroup2
ubuntu@frontend-server:~$ chmod 400 cloudgroup2.key
ubuntu@frontend-server:~$ rm cloudgroup2.key.pub
ubuntu@frontend-server:~$ ls
cloudgroup2.key TUBCloudComputingGroup2
```

- **It shows that it login to the frontend server.**

```
ubuntu@frontend-server:~$ ssh -i cloudgroup2.key ubuntu@10.2.1.5
```



```

ubuntu@frontend-server:~$ ssh -i cloudgroup2.key ubuntu@10.2.1.5
The authenticity of host '10.2.1.5 (10.2.1.5)' can't be established.
ECDSA key fingerprint is SHA256:1K0owAYQkjb13WGezBDHcFzhtwe0Wev+QutNXpgp6CA.
[Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.2.1.5' (ECDSA) to the list of known hosts.
[Enter passphrase for key 'cloudgroup2.key':
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-22-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Get cloud support with Ubuntu Advantage Cloud Guest:
  http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@backend-server-0:~$ █

```

- Testing the internet connectivity in the first backend server

```
ubuntu@backend-server-0:~$ git clone
```

```
https://github.com/julieallard/TUBCloudComputingGroup2.git
```

```

ubuntu@backend-server-0:~$ git clone https://github.com/julieallard/TUBCloudComputingGroup2.git
Cloning into 'TUBCloudComputingGroup2'...
[Username for 'https://github.com': navyatub
[Password for 'https://navyatub@github.com':
remote: Counting objects: 89, done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 89 (delta 0), reused 4 (delta 0), pack-reused 84
Unpacking objects: 100% (89/89), done.
Checking connectivity... done.
ubuntu@backend-server-0:~$ ls
TUBCloudComputingGroup2
ubuntu@backend-server-0:~$

```

```
TUBCloudComputingGroup2
[ubuntu@backend-server-0:~$ exit
logout
Connection to 10.2.1.5 closed.
ubuntu@frontend-server:~$
```

- Login to the backend server from frontend server using ssh

```
ubuntu@frontend-server:~$ ssh -i cloudgroup2.key ubuntu@10.2.1.4
```

```
[ubuntu@frontend-server:~$ ssh -i cloudgroup2.key ubuntu@10.2.1.4
The authenticity of host '10.2.1.4 (10.2.1.4)' can't be established.
ECDSA key fingerprint is SHA256:SwFpgPZXLM+iPifea1fZoyAdlYtKzGIgfU59YEcbTUU.
[Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.2.1.4' (ECDSA) to the list of known hosts.
[Enter passphrase for key 'cloudgroup2.key':
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-22-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

    Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@backend-server-1:~$
```

- Testing the internet connectivity in the second backend server

```
ubuntu@backend-server-1:~$ git clone
```

```
https://github.com/julieallard/TUBCloudComputingGroup2.git
```

```
ubuntu@backend-server-1:~$ git clone https://github.com/julieallard/TUBCloudComputingGroup2.git
Cloning into 'TUBCloudComputingGroup2'...
[Username for 'https://github.com': navyatub
>Password for 'https://navyatub@github.com':
remote: Counting objects: 89, done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 89 (delta 0), reused 4 (delta 0), pack-reused 84
Unpacking objects: 100% (89/89), done.
Checking connectivity... done.
ubuntu@backend-server-1:~$ ls
TUBCloudComputingGroup2
ubuntu@backend-server-1:~$
```

```
TUBCloudComputingGroup2
[ubuntu@backend-server-1:~$ exit
logout
Connection to 10.2.1.4 closed.
ubuntu@frontend-server:~$ █
```