



# **OST-104 - Openstack private cloud workshop**

*Activity guide*

*Release L rev149*

**Component Soft Ltd.**

November 26, 2016

**SAMPLE**

The contents of this course and all its modules and related materials, including handouts to audience members, are copyright © 2016 Component Soft Ltd.

No part of this publication may be stored in a retrieval system, transmitted or reproduced in any way, including, but not limited to, photocopy, photograph, magnetic, electronic or other record, without the prior written permission of Component Soft Ltd.

This curriculum contains proprietary information which is for the exclusive use of customers of Component Soft Ltd., and is not to be shared with personnel other than those in attendance at this course.

This instructional program, including all material provided herein, is supplied without any guarantees from Component Soft Ltd. Component Soft Ltd. assumes no liability for damages or legal action arising from the use or misuse of contents or details contained herein.

Photocopying any part of this manual without prior written consent of Component Soft Ltd. is a violation of law. This manual should not appear to be a photocopy. If you believe that Component Soft Ltd. training materials are being photocopied without permission, please write an email to [info@componentsoft.eu](mailto:info@componentsoft.eu).

Component Soft Ltd. accepts no liability for any claims, demands, losses, damages, costs or expenses suffered or incurred howsoever arising from or in connection with the use of this courseware. All trademarks are the property of their respective owners.

---

# Contents

---

<b>Preface</b>	<b>1</b>
About command line outputs . . . . .	1
About the lab environment . . . . .	3
<b>Lab 1: OpenStack overview</b>	<b>5</b>
Task 1: Health check . . . . .	5
Task 2: Test instance creation . . . . .	11
Task 3: Cleanup . . . . .	14
<b>Lab 2: Controller node operations</b>	<b>15</b>
Task 1: Keystone . . . . .	15
Task 2: Glance . . . . .	17
Task 3: Cinder . . . . .	18
Task 4: Cleanup . . . . .	21
<b>Lab 3: Compute</b>	<b>23</b>
Task 1: Setup . . . . .	23
Task 2: VM placement . . . . .	23
Task 3: Instance post configuration . . . . .	25
Task 4: Image customization . . . . .	27
Task 5: Migration (optional) . . . . .	27
Task 6: Cleanup . . . . .	28
<b>Lab 4: Using Neutron</b>	<b>29</b>
Task 1: Neutron CLI . . . . .	29
Task 2: Working with security groups . . . . .	33
Task 3: Debugging data flows . . . . .	33
Task 4: Create Neutron load balancer (optional) . . . . .	36
Task 5: Cleanup . . . . .	38
<b>Lab 5: Ceilometer</b>	<b>39</b>
Task 1: Ceilometer health check . . . . .	39
Task 2: Start instance . . . . .	39
Task 3: Check VM with ceilometer . . . . .	39

Task 4: Alarm setup . . . . .	40
Task 5: Cleanup . . . . .	40
<b>Lab 6: Heat</b>	<b>43</b>
Task 1: Basic static template . . . . .	43
Task 2: Template using input parameters and environment . . . . .	44
Task 3: Nested templates . . . . .	45
Task 4: Template with auto-scaling . . . . .	47
Task 5: Cleanup . . . . .	49
<b>Lab 7: Swift</b>	<b>51</b>
Task 1: Start storage nodes . . . . .	51
Task 2: Upload files . . . . .	52
Task 3: Swift backend for glance . . . . .	52
<b>Solutions</b>	<b>55</b>
Solutions 2: Controller node operations . . . . .	55
Solutions 3: Compute node operations . . . . .	77
Solutions 4: Openstack networking . . . . .	93
Solutions 5: Ceilometer . . . . .	121
Solutions 6: Heat . . . . .	127
Solutions 7: Swift . . . . .	150
<b>Appendix</b>	<b>159</b>
Solutions 8: Installation . . . . .	159

## About command line outputs

### Note:

Here we describe the formatting rules of this book. Take the following commands as examples only. These commands not necessary execute correctly or produce the same output for your actual setup.

### Example 1

```
root@controller1 (admin) $> nova list --minimal
+-----+-----+
| ID                | Name          |
+-----+-----+
| 58012abf-1b73-40ec-989c-96f4592cd277 | test_1       |
+-----+-----+
```

In this case

- The command runs on the controller node, as user root.
- The keystone credentials for tenant admin are loaded into the shell environment.

In the above case it is required to load the admin credentials in order to make the certain command to work properly. In case of the admin user (and tenant), it can be done by sourcing the file `/root/keystonerc_admin`

```
root@controller1 $> source /root/keystonerc_admin
root@controller1 (admin) $> env |grep OS_
OS_REGION_NAME=RegionOne
OS_PASSWORD=makeitso
OS_AUTH_URL=http://10.10.10.51:5000/v2.0/
OS_USERNAME=admin
OS_TENANT_NAME=admin
```

Commands not related to OpenStack (like `ls`) does not take care of the OpenStack credentials at all, so for those, it is irrelevant whether you can `OS_` environment variables or not.

## Example 2

```
root@controller1 (admin) $> nova interface-list vm1
```

```
// Port ID | Net ID >>
//-----|----->>
// 7912e922-e871-4e7a-a943-34017ee29160 | 41f61be7-40b9-4a67-aeca-feae3a5986ac>>
// 93925c7f-0112-493c-8a39-261449128f5f | e3ee2d62-e216-4e76-b438-733e706f1500>>

IP addresses //
-----//
10.40.40.100 //
10.30.30.102 //
```

In this one, the output is too long, so the first and last columns are cut off ( // symbols) and the third column is presented separately (>> symbol)

## Example 3

```
root@controller1 (admin)$> nova host-describe compute2.openstack.local
```

HOST	PROJECT	cpu	memory_mb	disk
compute2.openstack.local	(total)	8	15948	4
compute2.openstack.local	(used_now)	4	2560	4
compute2.openstack.local	(used_max)	4	2048	4
compute2.openstack.local	<b>0cb8d/--/c274e67</b>	4	2048	4

In this case the second column PROJECT is truncated to 20 characters, so UUID 0cb8d6ab778546bbadc69488dc274e67 is shortened to 0cb8d/--/c274e67.

## About the lab environment

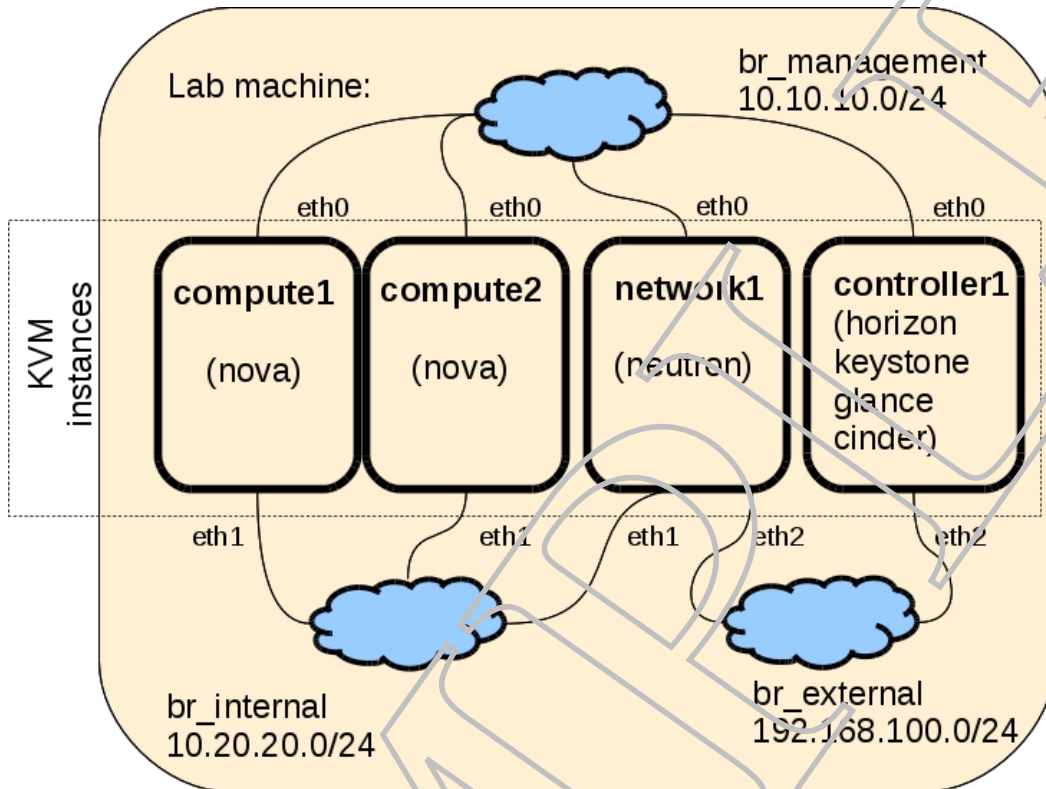


Figure 1: The lab setup

Each student desktop is running a complete Openstack environment. The host machine runs KVM instances for the controller, network and two hypervisor nodes `compute1`, `compute2`. Node `compute1` has 2 GB, while `compute2` has 4GB memory, so at least initially `compute2` is chosen to start a new VM instance.

The lab environment uses 3 networks

- `br_management` (10.10.10.0/24)

This network is used to log into the nodes. Internal signaling and REST API calls also happen on that network.

- `br_external` (192.168.100.0/24)

This is the *external* network.

- Floating IPs are allocated from this IP range
- The API external URLs are using the *external* IP of the controller
- The Horizon dashboard can also be accessed via the *external* IP of the controller

- br\_internal (10.20.20.0/24)

This network is used for communication between the network and the compute nodes.

The IP addresses used are summarized in the following table.

Node	br_management	br_internal	br_external
lab machine	10.10.10.1	10.20.20.1	192.168.100.1
controller1	10.10.10.51	—	192.168.100.51
network1	10.10.10.52	10.20.20.52	192.168.100.52
compute1	10.10.10.53	10.20.20.53	—
compute2	10.10.10.54	10.20.20.54	—
storage1	10.10.10.55	—	—
storage2	10.10.10.56	—	—
storage3	10.10.10.57	—	—

**Note:**

The storage nodes (like storage1) are used later in this course , and not started by default.

## Credentials

An SSH grant is configured between the lab\_machine and the VM instances, so you can ssh in via root without a password. We have the following user/password combinations

User	Password
root of all OpenStack nodes	makeitso
All Keystone users (like nova,cinder, etc.)	makeitso
All MySQL users (like nova,cinder, etc.)	makeitso

## Labfiles

Files referred by this activity guide are placed into the directory /labfiles on lab machine, and shared via NFS to all nodes.

If you use a remote lab, please refer the [Wiki page](#) for further information.



---

# Lab 1: OpenStack overview

---

## Task 1: Health check

Perform basic health check on your local OpenStack installation.

- On your **lab host**, check whether your nodes are running

```
root@lab_machine $> /labfiles/os_nodes list
Id      Name                State
-----
1       controller1         running
2       network1            running
3       compute1            running
4       compute2            running
```

- Check whether VMs have the same system date and time ( 1-2 seconds of difference is OK).

```

root@lab_machine $> for i in controller1 network1 compute1 compute2
> do
> echo -en "$i :"; ssh $i 'date'
> done
controller1 :Mon Jun 16 15:52:49 CEST 2014
network1 :Mon Jun 16 15:52:49 CEST 2014
compute1 :Mon Jun 16 15:52:49 CEST 2014
compute2 :Mon Jun 16 15:52:49 CEST 2014

```

**Note:**

The time values presented by the previous command should not differ more than a couple of seconds. If you see a larger difference, please synchronize manually.

```

root@lab_machine $> for i in network1 compute1 compute2 controller1
> do ssh $i 'systemctl restart ntpd' ;done

root@lab_machine $> for i in network1 compute1 compute2 controller1;
> do ssh $i 'ntpdc <<peers' ;done

      remote          local      st poll reach  delay  offset  disp
=====
=host-mgt.openst 10.10.10.52      4 64 1 0.00006 0.000002 2.81735
      remote          local      st poll reach  delay  offset  disp
=====
=host-mgt.openst 10.10.10.53      4 64 1 0.00006 0.000043 2.81735
      remote          local      st poll reach  delay  offset  disp
=====
=host-mgt.openst 10.10.10.54      4 64 1 0.00006 0.000022 2.81735
      remote          local      st poll reach  delay  offset  disp
=====
=host-mgt.openst 10.10.10.51      4 64 1 0.00015 0.000014 2.81735
..

```

- Check the status of the openstack related services on all nodes

```

root@lab_machine $> for i in controller1 network1 compute{1,2}; \
> do echo "###$i###";ssh $i 'openstack-service status';done
###controller1###
MainPID=967 Id=neutron-server.service ActiveState=active
MainPID=999 Id=openstack-aodh-api.service ActiveState=active
MainPID=988 Id=openstack-aodh-evaluator.service ActiveState=active
MainPID=985 Id=openstack-aodh-notifier.service ActiveState=active
MainPID=979 Id=openstack-ceilometer-api.service ActiveState=active
MainPID=951 Id=openstack-ceilometer-central.service ActiveState=active
MainPID=998 Id=openstack-ceilometer-collector.service ActiveState=active
MainPID=972 Id=openstack-ceilometer-notification.service ActiveState=active
MainPID=994 Id=openstack-cinder-api.service ActiveState=active
MainPID=993 Id=openstack-cinder-scheduler.service ActiveState=active
MainPID=992 Id=openstack-cinder-volume.service ActiveState=active
MainPID=991 Id=openstack-glance-api.service ActiveState=active
MainPID=975 Id=openstack-glance-registry.service ActiveState=active
MainPID=1005 Id=openstack-heat-api-cfn.service ActiveState=active
MainPID=995 Id=openstack-heat-api-cloudwatch.service ActiveState=active
MainPID=981 Id=openstack-heat-api.service ActiveState=active

```

```

MainPID=2540 Id=openstack-heat-engine.service ActiveState=active
MainPID=989 Id=openstack-nova-api.service ActiveState=active
MainPID=978 Id=openstack-nova-cert.service ActiveState=active
MainPID=1009 Id=openstack-nova-conductor.service ActiveState=active
MainPID=947 Id=openstack-nova-consoleauth.service ActiveState=active
MainPID=1003 Id=openstack-nova-novncproxy.service ActiveState=active
MainPID=976 Id=openstack-nova-scheduler.service ActiveState=active
###network1###
MainPID=1463 Id=neutron-dhcp-agent.service ActiveState=active
MainPID=1198 Id=neutron-lbaas-agent.service ActiveState=active
MainPID=1190 Id=neutron-metadata-agent.service ActiveState=active
MainPID=1193 Id=neutron-metering-agent.service ActiveState=active
MainPID=1464 Id=neutron-openvswitch-agent.service ActiveState=active
MainPID=1196 Id=neutron-vpn-agent.service ActiveState=active
###compute1###
MainPID=1340 Id=neutron-openvswitch-agent.service ActiveState=active
MainPID=1021 Id=openstack-ceilometer-compute.service ActiveState=active
MainPID=1341 Id=openstack-nova-compute.service ActiveState=active
###compute2###
MainPID=1416 Id=neutron-openvswitch-agent.service ActiveState=active
MainPID=1092 Id=openstack-ceilometer-compute.service ActiveState=active
MainPID=1415 Id=openstack-nova-compute.service ActiveState=active

```

**Note:**

If you have a “dead” service, please use `systemctl restart <service_name>` or `openstack-service restart` to restart it. For example, if you see that `nova-conductor` has failure, then

```
root@controller1 $> systemctl restart openstack-nova-conductor
```

**Note:**

Keystone has two options:

- Run as a standalone service, so run its own Python based web server to serve REST API on port 5000 and 35357.
- Run as WSGI app behind the Apache webserver. This solution scales better in case of a lot of API calls. Your lab environment uses the second option, so service `keystone` is disabled, by default.

- Get the version of OpenStack

```
root@controller1 $> nova-manage version 2>/dev/null
12.0.1-1.e17
```

**Note:**

The 12 stands for the 12th OpenStack release, which is **Liberty**.

- Verify that Keystone works fine

```

root@controller1 $> ss -antlp |grep -E '(5000|35357)'
LISTEN      0      128      :::5000      :::*
users: (("httpd",pid=2577,fd=6), ("httpd",pid=2576,fd=6), ("httpd",pid=2575,fd=6),
 ("httpd",pid=2574,fd=6), ("httpd",pid=2573,fd=6), ("httpd",pid=2572,fd=6),
 ("httpd",pid=2571,fd=6), ("httpd",pid=2570,fd=6), ("httpd",pid=2563,fd=6))
LISTEN      0      128      :::35357     :::*
users: (("httpd",pid=2577,fd=4), ("httpd",pid=2576,fd=4), ("httpd",pid=2575,fd=4),
 ("httpd",pid=2574,fd=4), ("httpd",pid=2573,fd=4), ("httpd",pid=2572,fd=4),
 ("httpd",pid=2571,fd=4), ("httpd",pid=2570,fd=4), ("httpd",pid=2563,fd=4))

root@controller1 $> curl --silent localhost:35357|python -mjson.tool|grep href
"href": "http://localhost:35357/v3/",
"href": "http://localhost:35357/v2.0/",
"href": "http://docs.openstack.org/",

```

```

root@controller1 (admin) $> grep admin_token /etc/keystone/keystone.conf
#admin_token = ADMIN
admin_token = 16f9c0d43f5b49e4a2774da32c876e91.

root@controller1 (admin) $> openstack --os-token=16f9c0d43f5b49e4a2774da32c876e91\
> --os-url=http://localhost:35357/v2.0 service list
//-----+-----+
// Name      | Type      |
//-----+-----+
// cinderv2   | volumev2  |
// nova       | compute   |
// heat-cfn   | cloudformation |
// ceilometer | metering  |
// heat       | orchestration |
// cinder     | volume    |
// glance     | image     |
// nova_ec2   | ec2       |
// keystone   | identity  |
// novav3     | computev3 |
// swift      | object-store |
// neutron    | network   |
// aodh       | alarming  |
//-----+-----+

```

- Load environment variables for the admin project (tenant)

```

root@controller1 $> cat /root/keystonerc_admin
export OS_USERNAME=admin
export OS_TENANT_NAME=admin
export OS_PASSWORD=narsitso
export OS_AUTH_URL=http://10.10.10.51:5000/v2.0/
export PS1='[\u@\h \W(admin)]\$ '
root@controller1 $> . /root/keystonerc_admin
root@controller1 (admin) $>

```

- List Keystone registered services

```
root@controller1 (admin) $> openstack service list
```

ID	Name	Type
02a90fae0e3840b8b3c67603616ad094	nova	compute
253371d7e5b649fe93513121a5f8df77	aodh	alarming
3619bd365bf54927b6405cc5422463c8	ceilometer	metering
398521d3d9ba4c3eaeb54fc3fe5918c6	neutron	network
6bcc619b359545b19d800e87d872399a	swift	object-store
6e02b5fac6534d9e8f5c41ea8379cf23	cinderv2	volumev2
9213050323a14a5d9ca205e1cad92979	glance	image
96a6c875500846408e18fd14e18d2ddd	cinder	volume
96d88a34294d4c6b93b3d9f7a24804ed	novav3	computev3
982277b5642443468f3b5e0ac7d4132c	nova_ec2	ec2
a2d6abf4fa454cf9809a31cd30d5b84b	heat-cfn	cloudformation
ef8598d6621647d9b92c806a9d76b8be	heat	orchestration
fb9a331944724ec1895b57c95a65a033	keystone	identity

**Note:**

The ID field in the above output may be different for each and every OpenStack installation.

- Get service locations (API endpoint URLs)

```
root@controller1 (admin)$> openstack catalog list | grep -E 'Service:|publicURL'
```

```
| | | publicURL: http://10.10.10.51:8774/v2/eb77b585a9184304a0248537431c723f |
| | | publicURL: http://10.10.10.51:9696 |
| | | publicURL: http://10.10.10.51:8776/v2/eb77b585a9184304a0248537431c723f |
| | | publicURL: http://10.10.10.51:8774/v3 |
| | | publicURL: http://10.10.10.51:9292 |
| | | publicURL: http://10.10.10.51:8042 |
| | | publicURL: http://10.10.10.51:8777 |
| | | publicURL: http://10.10.10.51:8000/v1 |
| | | publicURL: http://10.10.10.51:8776/v1/eb77b585a9184304a0248537431c723f |
| | | publicURL: http://10.10.10.51:8773/services/Cloud |
| | | publicURL: http://10.10.10.51:8004/v1/eb77b585a9184304a0248537431c723f |
| | | publicURL: http://10.10.10.51:8080/v1/AUTH_eb77b585a9184304a0248537431c723f |
| | | publicURL: http://10.10.10.51:5000/v2.0 |
```

**Note:**

Please note, that all the OpenStack related API processes run on the controller node (10.10.10.51)

**Note:**

You can get this info in Horizon, using url <http://10.10.10.51//dashboard/admin/info/>, tab 'Services'

**Note:**

Some OpenStack commands like `nova-manage` or `cinder-manage` do not use any REST API interface but do direct AMQP and DB operations. It implies that those

- has to run on the `controller` node only.
- do not need Keystone credentials, so do not care about the credentials (`OS_` environment variables) loaded.

- List nova related services

```
root@controller1 (admin) $> nova service-list
//-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
// Binary      | Host                               | Zone   | Status  | State  |
//-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
// nova-consoleauth | controller1.openstack.local | internal | enabled | up     |
// nova-scheduler  | controller1.openstack.local | internal | enabled | up     |
// nova-conductor   | controller1.openstack.local | internal | enabled | up     |
// nova-cert        | controller1.openstack.local | internal | enabled | up     |
// nova-compute     | compute1.openstack.local    | nova    | enabled | up     |
// nova-compute     | compute2.openstack.local    | nova    | enabled | up     |
//-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

**Note:**

The above output shows **2 compute nodes**: `compute1` and `compute2`.

**Note:**

If any of these services is reported as down, try to restart. So in case

```
//nova-consoleauth | controller1.openstack.local | internal | enabled | down
```

, perform a restart via

```
root@controller1 (admin) $> systemctl restart openstack-nova-consoleauth
```

**Note:**

You can get this info in Horizon, using url <http://10.10.10.51//dashboard/admin/info/>, tab 'Compute services'

- Check Cinder

```
root@controller1 (admin) $> cinder service-list
//-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
// Binary      | Host                               | Zone   | Status  | State  |
//-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
// cinder-backup | controller1.openstack.local | nova    | enabled | up     |
// cinder-scheduler | controller1.openstack.local | nova    | enabled | up     |
// cinder-volume  | controller1.openstack.local@lvm | nova    | enabled | up     |
//-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

- Check Glance

```
root@controller1 $> openstack-service status |grep glance
openstack-glance-api (pid 818) is active
openstack-glance-registry (pid 817) is active

root@controller1 $> netstat -alntp |grep ':9[12]'
```

tcp	0	0	0.0.0.0:9191	0.0.0.0:*	LISTEN	817/python2
tcp	0	0	0.0.0.0:9292	0.0.0.0:*	LISTEN	818/python2

- Check Neutron

```
root@controller1 (admin)$> neutron agent-list
//-----+-----+-----+-----//
// agent_type          | host                               | alive //
//-----+-----+-----+-----//
// L3 agent            | network1.openstack.local          | :-)| //
// Open vSwitch agent  | network1.openstack.local          | :-)| //
// Metadata agent      | network1.openstack.local          | :-)| //
// Metering agent      | network1.openstack.local          | :-)| //
// Open vSwitch agent  | compute1.openstack.local          | :-)| //
// Loadbalancer agent  | network1.openstack.local          | :-)| //
// Open vSwitch agent  | compute2.openstack.local          | :-)| //
// DHCP agent          | network1.openstack.local          | :-)| //
//-----+-----+-----+-----//
```

#### Note:

This check for neutron can fail, as follows

```
root@controller1 (admin) $> neutron agent-list
Authentication required
```

The root of this authentication (keystone related) problem might be the difference in system datetime on machines.

```
root@controller1 (admin) $> date
Fri Jun 13 22:24:25 CEST 2014
```

```
root@network1 $> date
Mon Jun 16 15:48:05 CEST 2014
```

#### Note:

You can get this info in Horizon, using url <http://10.10.10.51//dashboard/admin/info/>, tab 'Network agents'

## Task 2: Test instance creation

In this task we run command that configures and starts a VM instance in our Openstack environment.



- Use Heat template /labfiles/lab1/test\_instances.yaml to start a test VM.

```
root@controller1 (admin) $> heat stack-create -f /labfiles/lab1/test_instances.yaml lab1
```

id	stack_name	stack_status
0f931871-e7a5-4675-9a38-90bdfd176bbd	lab1	CREATE_IN_PROGRESS

- Check status of the stack.

```
root@controller1 (admin) $> heat stack-list
```

id	stack_name	stack_status
0f931871-e7a5-4675-9a38-90bdfd176bbd	lab1	CREATE_IN_PROGRESS

```
root@controller1 (admin) $> heat stack-list
```

id	stack_name	stack_status
0f931871-e7a5-4675-9a38-90bdfd176bbd	lab1	CREATE_COMPLETE

```
root@controller1 (admin) $> heat resource-list lab1
```

resource_name	resource_type	resource_status
CirrOS	OS::Glance::Image	CREATE_COMPLETE
guest-key	OS::Nova::KeyPair	CREATE_COMPLETE
net_ext	OS::Neutron::Net	CREATE_COMPLETE
net_int	OS::Neutron::Net	CREATE_COMPLETE
subnet_ext	OS::Neutron::Subnet	CREATE_COMPLETE
subnet_int	OS::Neutron::Subnet	CREATE_COMPLETE
int_grel_router	OS::Neutron::Router	CREATE_COMPLETE
port1	OS::Neutron::Port	CREATE_COMPLETE
int_router_interface	OS::Neutron::RouterInterface	CREATE_COMPLETE
test_1	OS::Nova::Server	CREATE_COMPLETE
fl_ip1	OS::Neutron::FloatingIP	CREATE_COMPLETE

**Tip:** You can review also the instance creation process in the Horizon dashboard at [http://10.10.10.51/dashboard/project/network\\_topology/](http://10.10.10.51/dashboard/project/network_topology/)

- Check the instance data for the associated floating IP



```

root@controller1 (admin) $> nova list
+-----+-----+-----+-----+-----+-----+
| ID          | Name    | Status | Task State | Power State |      |
+-----+-----+-----+-----+-----+-----+
| 694c8b79-9868-4c97-918a-0810d45b660a | test_1 | ACTIVE | -          | Running    |      |
+-----+-----+-----+-----+-----+-----+
| Networks   |         |         |           |            |      |
+-----+-----+-----+-----+-----+
| int_gre1=10.30.30.100, 192.168.100.101 |         |         |           |            |      |
+-----+-----+-----+-----+-----+

```

- SSH in with the ssh key /labfiles/id\_rsa with user cirros.

```

root@controller1 (admin) $> ssh -i /labfiles/id_rsa cirros@192.168.100.101
$ uname -a
Linux test-1 3.2.0-60-virtual #91-Ubuntu SMP Wed Feb 19 04:13:28 UTC 2014
x86_64 GNU/Linux

$ ip a s
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether fa:16:3e:f9:33:66 brd ff:ff:ff:ff:ff:ff
    inet 10.30.30.100/24 brd 10.30.30.255 scope global eth0
    inet6 fe80::f816:3eff:fef9:3366/64 scope link
        valid_lft forever preferred_lft forever

```

#### Note:

If ssh fails, you can troubleshoot the problem via VNC, using the noVnc console (<http://10.10.10.51/dashboard/project/instances/>, "Console" in the roll-down menu of the instance)

You can also retrieve the console URL by command `nova console url show`.

```

root@controller1 (admin) $> openstack console url show --novnc <vm UUID>
+-----+-----+
| Value |
+-----+-----+
| novnc |
| http://10.10.10.51:6080/vnc_auto.ncml?token=7a71fa2e-124b-49c2-a388-fdcf63b66809 |
+-----+-----+

```

#### For remote lab,

- Open a SSH session for the lab\_machine, which will create the necessary tunnels (check the [Wiki page](#) for SSH forwarded ports)
- Change 10.10.10.51:6080 to 127.0.0.1:36080 in the VNC session URL.
- In a browser, open the VNC session.

## Task 3: Cleanup

- Clean up the created instance with command `heat stack-delete lab1`

```
root@controller1 (admin) $> heat stack-delete lab1
root@controller1 (admin) $> heat stack-list
+-----+-----+-----+-----+
| id | stack_name | stack_status | creation_time |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

- Check if cleanup is complete

```
root@controller1 (admin) $> nova list
+-----+-----+-----+-----+-----+-----+
| ID | Name | Status | Task State | Power State | Networks |
+-----+-----+-----+-----+-----+-----+
root@controller1 ~(admin) $> neutron net-list
root@controller1 ~(admin) $> neutron router-list
root@controller1 ~(admin) $> glance image-list
+-----+-----+-----+-----+-----+-----+
| ID | Name | Disk Format | Container Format | Size | Status |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
```