Distributed Cloud Storage for Health Monitoring Data

K.Snehal Reddy
Kothapalli Sandeep
Amshumaan Varma
Robin Babu

Project Overview

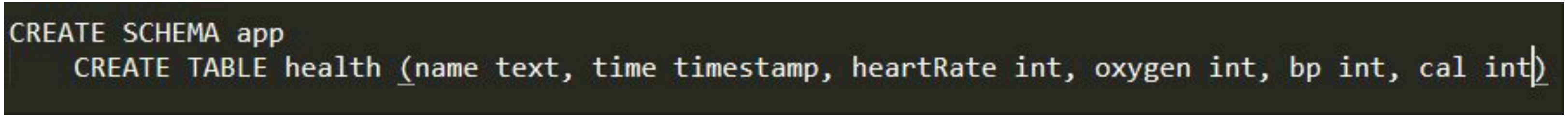
- We aim to build software to aid in health monitoring using real time biometrics of users from health sensors.

Hardware

- Virtual Systems with
  - Ubuntu 20.04
  - 8 Gb RAM
  - 50 Gb Memory
- Installed openstack on all the 4 VMs.

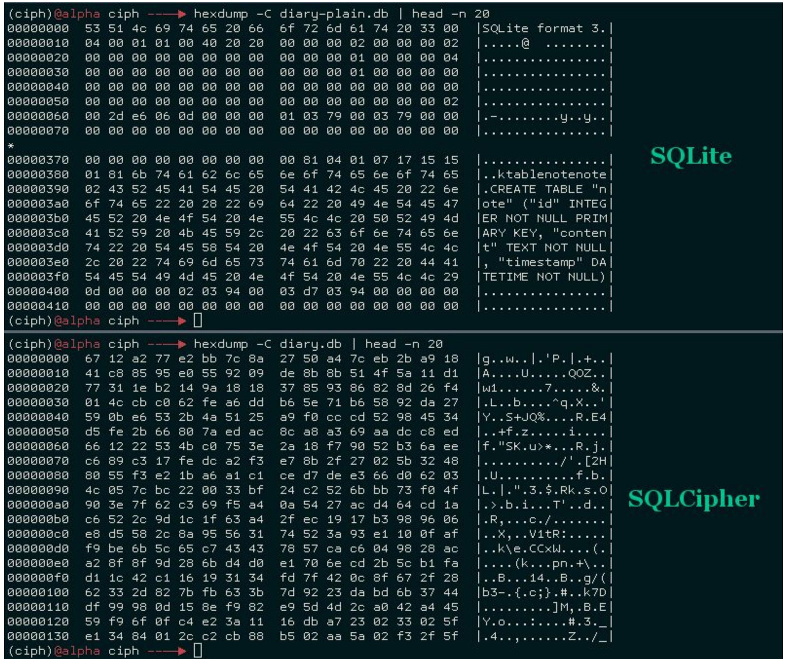
Data specifications

- The data that we plan to collect, store and analyse are as follows -
    - Heart rate
    - Blood oxygen levels
    - Blood pressure
    - Calories burnt
- All these are indexed wrt. a time stamp.
- So the schema is



```
CREATE SCHEMA app
    CREATE TABLE health (name text, time timestamp, heartRate int, oxygen int, bp int, cal int)
```

Encryption using SQLCipher



```
(ciph)@alpha ciph ---> hexdump -C diary-plain.db | head -n 20
00000000  53 51 4c 69 74 65 20 66  6f 72 6d 61 74 20 33 00  |SQLite format 3.|
00000010  04 00 01 01 00 40 20 20  00 00 00 02 00 00 00 02  |.....@  ........|
00000020  00 00 00 00 00 00 00 00  00 00 00 01 00 00 00 04  |................|
00000030  00 00 00 00 00 00 00 00  00 00 00 01 00 00 00 00  |................|
00000040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 02  |................|
00000060  00 2d e6 06 0d 00 00 00  01 03 79 00 03 79 00 00  |.-........y..y..|
00000070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000370  00 00 00 00 00 00 00 00  00 81 04 01 07 17 15 15  |................|
00000380  01 81 6b 74 61 62 6c 65  6e 6f 74 65 6e 6f 74 65  |..ktablenotenote|
00000390  02 43 52 45 41 54 45 20  54 41 42 4c 45 20 22 6e  |.CREATE TABLE "n|
000003a0  6f 74 65 22 20 28 22 69  64 22 20 49 4e 54 45 47  |ote" ("id" INTEG|
000003b0  45 52 20 4e 4f 54 20 4e  55 4c 4c 20 50 52 49 4d  |ER NOT NULL PRIM|
000003c0  41 52 59 20 4b 45 59 2c  20 22 63 6f 6e 74 65 6e  |ARY KEY, "conten|
000003d0  74 22 20 54 45 58 54 20  4e 4f 54 20 4e 55 4c 4c  |t" TEXT NOT NULL|
000003e0  2c 20 22 74 69 6d 65 73  74 61 6d 70 22 20 44 41  |, "timestamp" DA|
000003f0  54 45 54 49 4d 45 20 4e  4f 54 20 4e 55 4c 4c 29  |TETIME NOT NULL)|
00000400  0d 00 00 00 02 03 94 00  03 d7 03 94 00 00 00 00  |................|
00000410  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
(ciph)@alpha ciph ---> 
```

```
(ciph)@alpha ciph ---> hexdump -C diary.db | head -n 20
00000000  67 12 a2 77 e2 bb 7c 8a  27 50 a4 7c eb 2b a9 18  |g..w..|.'P.|.+..|
00000010  41 c8 85 95 e0 55 92 09  de 8b 8b 51 4f 5a 11 d1  |A....U.....QOZ..|
00000020  77 31 1e b2 14 9a 18 18  37 85 93 86 82 8d 26 f4  |w1......7.....&.|
00000030  01 4c cb c0 62 fe a6 dd  b6 5e 71 b6 58 92 da 27  |.L..b....^q.X..'|
00000040  59 0b e6 53 2b 4a 51 25  a9 f0 cc cd 52 98 45 34  |Y..S+JQ%....R.E4|
00000050  d5 fe 2b 66 80 7a ed ac  8c a8 a3 69 aa dc c8 ed  |..+f.z.....i....|
00000060  66 12 22 53 4b c0 75 3e  2a 18 f7 90 52 b3 6a ee  |f."SK.u>*...R.j.|
00000070  c6 89 c3 17 fe dc a2 f3  e7 8b 2f 27 02 5b 32 48  |........../'.[2H|
00000080  80 55 f3 e2 1b a6 a1 c1  ce d7 de e3 66 d0 62 03  |.U..........f.b.|
00000090  4c 05 7c bc 22 00 33 bf  24 c2 52 6b bb 73 f0 4f  |L.|.".3.$.Rk.s.O|
000000a0  90 3e 7f 62 c3 69 f5 a4  0a 54 27 ac d4 64 cd 1a  |.>.b.i...T'..d..|
000000b0  c6 52 2c 9d 1c 1f 63 a4  2f ec 19 17 b3 98 96 06  |.R,...c./.......|
000000c0  e8 d5 58 2c 8a 95 56 31  74 52 3a 93 e1 10 0f af  |..X,..V1tR:.....|
000000d0  f9 be 6b 5c 65 c7 43 43  78 57 ca c6 04 98 28 ac  |..k\e.CCxW....(.|
000000e0  a2 8f 8f 9d 28 6b d4 d0  e1 70 6e cd 2b 5c b1 fa  |....(k...pn.+\..|
000000f0  d1 1c 42 c1 16 19 31 34  fd 7f 42 0c 8f 67 2f 28  |..B...14..B..g/(|
00000100  62 33 2d 82 7b fb 63 3b  7d 92 23 da bd 6b 37 44  |b3-.{.c;}.#..k7D|
00000110  df 99 98 0d 15 8e f9 82  e9 5d 4d 2c a0 42 a4 45  |.........]M,.B.E|
00000120  59 f9 6f 0f c4 e2 3a 11  16 db a7 23 02 33 02 5f  |Y.o...:....#.3._|
00000130  e1 34 84 01 2c c2 cb 88  b5 02 aa 5a 02 f3 2f 5f  |.4..,......Z../_|
(ciph)@alpha ciph ---> 
```

Front end and web server

- Necessary frontend for web pages like, login, signup was written using HTML5 and Bootstrap v5.0
- Currently working on designing the web pages for user dashboard and user health data interface. [completed]
- The user data from login, signup and healthcare data from user dashboard is accessible at the backend. Relevant tables, schema and database models were created using SQLite through SQLAlchemy and Python Flask. [completed]

UI / Frontend

- Functionalities
    - User enters their data they intend to monitor. E.g. Blood Pressure, SPO2, beats-per-minute.
    - Frontend processes it and sends in appropriate format to the backend.
    - User can also see all the previous data they have entered. This will be fetched from the database/object storage implemented using openstack.

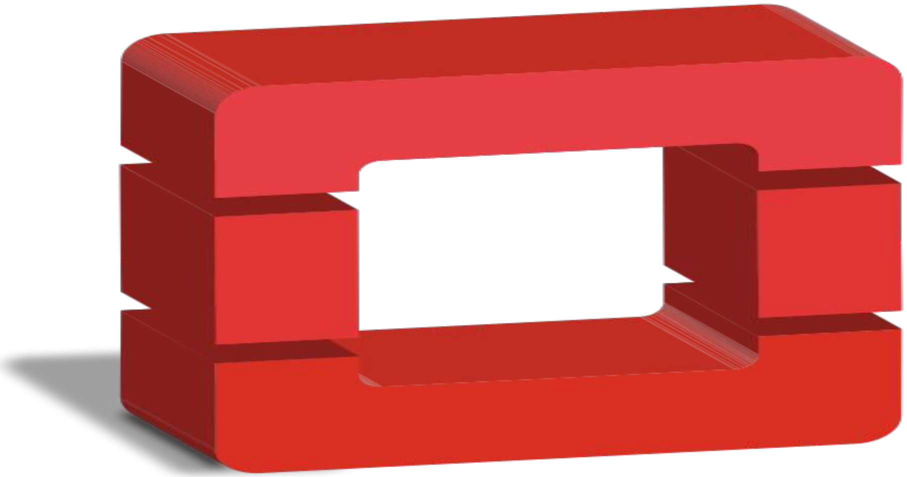
.

Data Processing

- As explained earlier through our front end interface the user can send realtime data of the user’s Heart rate, Oxygen Saturation levels (spO2) and Blood Pressure, Calorie burn rate.
- The real time data will be sent to our OpenStack cloud Virtual Machine. We then compute some results and insights regarding the person’s health data.
- The user can run it in two modes Excercise mode or Sleep Mode.

Data Processing Insights

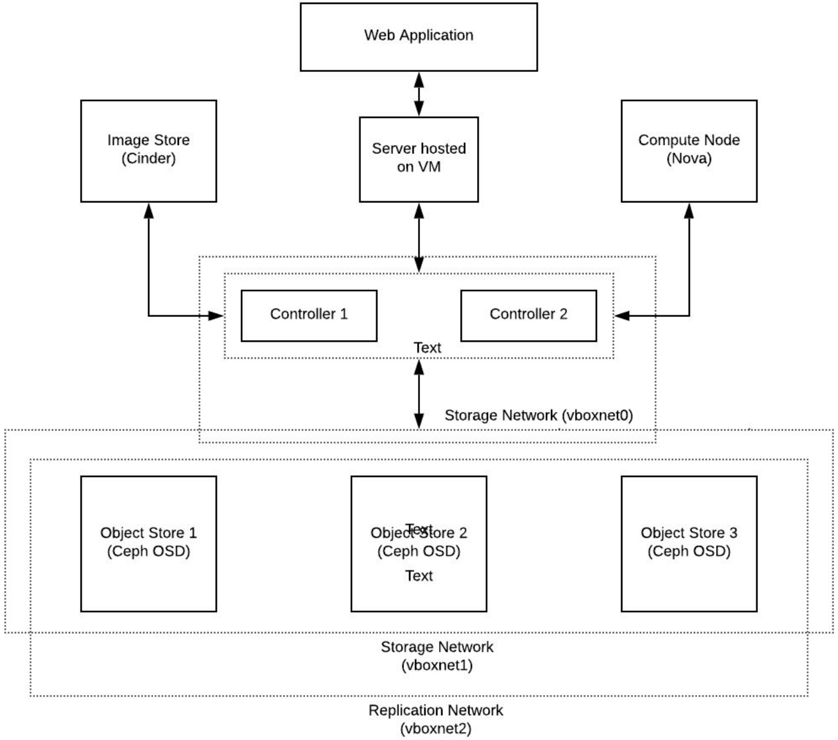
- We will present the minimum, average and maximum data values of the person’s heart rate, Oxygen saturation levels and blood pressure. This will let the user understand if his vitals are in healthy limits or not.
- The exercise mode possesses additional information regarding calorie consumption.
- The sleep mode contains additional information regarding the variance of heart beat rate which is used to understand the sleep conditions of the person.



openstack®

CLOUD SOFTWARE

Architecture Implemented


Web Application
Image Store (Cinder)
Server hosted on VM
Compute Node (Nova)
Controller 1
Controller 2
Text
Storage Network (vboxnet0)
Object Store 1 (Ceph OSD)
Object Store 2 (Ceph OSD)
Text
Text
Object Store 3 (Ceph OSD)
Storage Network (vboxnet1)
Replication Network (vboxnet2)

Openstack Implementation

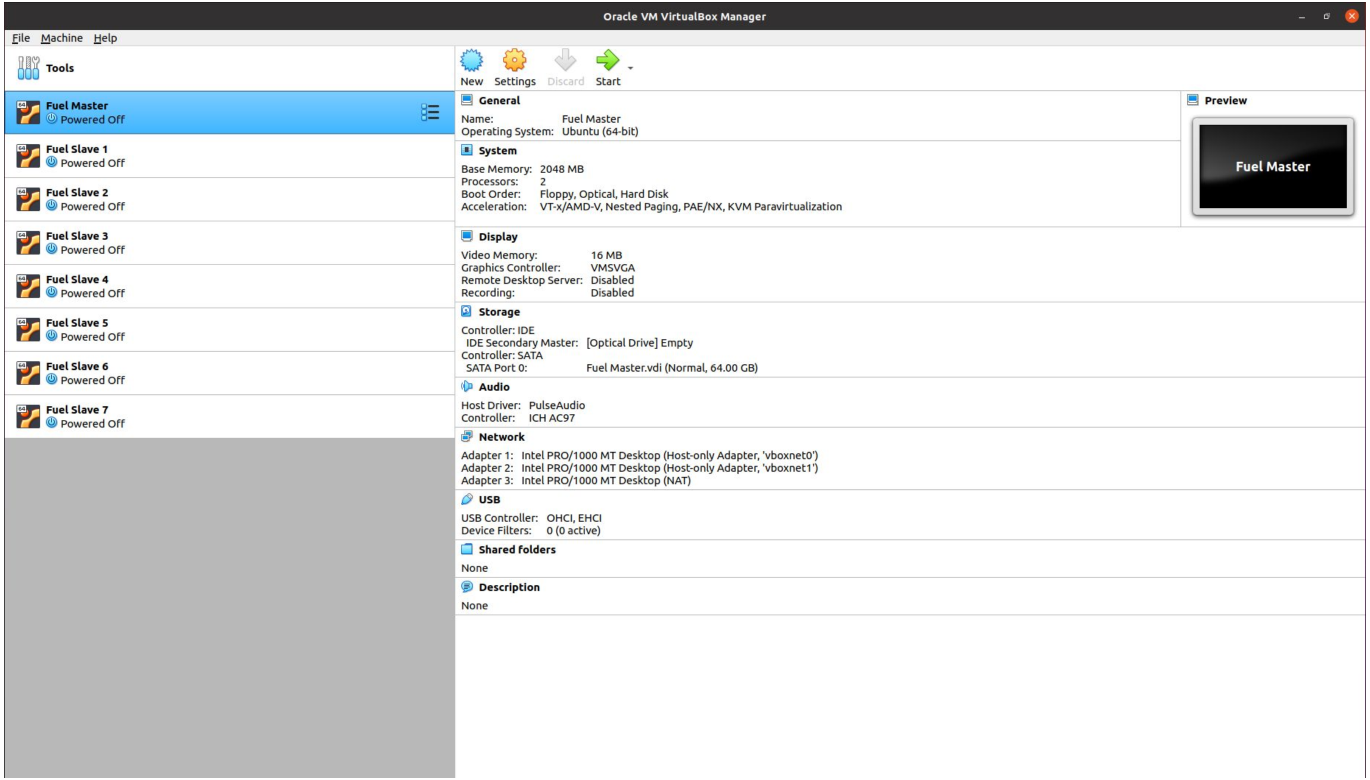
- Deployed on a single system using multiple virtual servers on Oracle Virtualbox due to network constraint to communicate among us. These virtual servers are connect through multiple local networks which can be configured on virtual box.
- Open Source Tool - Fuel is used to deploy different components of openstack cloud on different nodes respective to their assigned functionality.
- Openstack is installed on all these helper nodes
- There is a single master node which has full deployment of openstack and detects these other nodes as there in the same network and the discussed architecture is build using fuel
- After final deployment, we can access the horizon dashboard from master node for creating and managing out project instance.
- Then we launch a VM on created personal cloud to install the web server that connects to our web application

Roles of Different Nodes

- The deployed nodes are as follows :
    - 2 Controller nodes (Keystone, Neutron etc)
    - 1 Compute node (Nova)
    - 1 Compute node (Nova)
    - 3 Object Storage (Ceph OSD / Swift)
- The master node has the fuel implementation and all the metadata to control the openstack deployment.
- The controller node runs the Identity service, Image service, management portions of Compute, management portion of Networking, various Networking agents, and the dashboard. 2 controllers are added so that the VM is still functional in case of single failure
- The Image store node contains bootable disk images to launch the VM
- The Object store nodes contain the disk attached to VM which is used for storing the data from web server and this disk is replicated among the deployed 3 nodes for high availability so that the VM can still access its disk even when a storage node is down.

Setting up the Openstack Environment

Virtual Servers



Fuel Master
Powered Off

Fuel Slave 1
Powered Off

Fuel Slave 2
Powered Off

Fuel Slave 3
Powered Off

Fuel Slave 4
Powered Off

Fuel Slave 5
Powered Off

Fuel Slave 6
Powered Off

Fuel Slave 7
Powered Off

General

```
Name:              Fuel Master
Operating System:  Ubuntu (64-bit)
```

System

```
Base Memory:   2048 MB
Processors:    2
Boot Order:    Floppy, Optical, Hard Disk
Acceleration:  VT-x/AMD-V, Nested Paging, PAE/NX, KVM Paravirtualization
```

Display

```
Video Memory:           16 MB
Graphics Controller:    VMSVGA
Remote Desktop Server:  Disabled
Recording:              Disabled
```

Storage

```
Controller: IDE
 IDE Secondary Master:  [Optical Drive] Empty
Controller: SATA
 SATA Port 0:           Fuel Master.vdi (Normal, 64.00 GB)
```

Audio

```
Host Driver:  PulseAudio
Controller:   ICH AC97
```

Network

```
Adapter 1:  Intel PRO/1000 MT Desktop (Host-only Adapter, 'vboxnet0')
Adapter 2:  Intel PRO/1000 MT Desktop (Host-only Adapter, 'vboxnet1')
Adapter 3:  Intel PRO/1000 MT Desktop (NAT)
```

USB

```
USB Controller:  OHCI, EHCI
Device Filters:  0 (0 active)
```

Shared folders

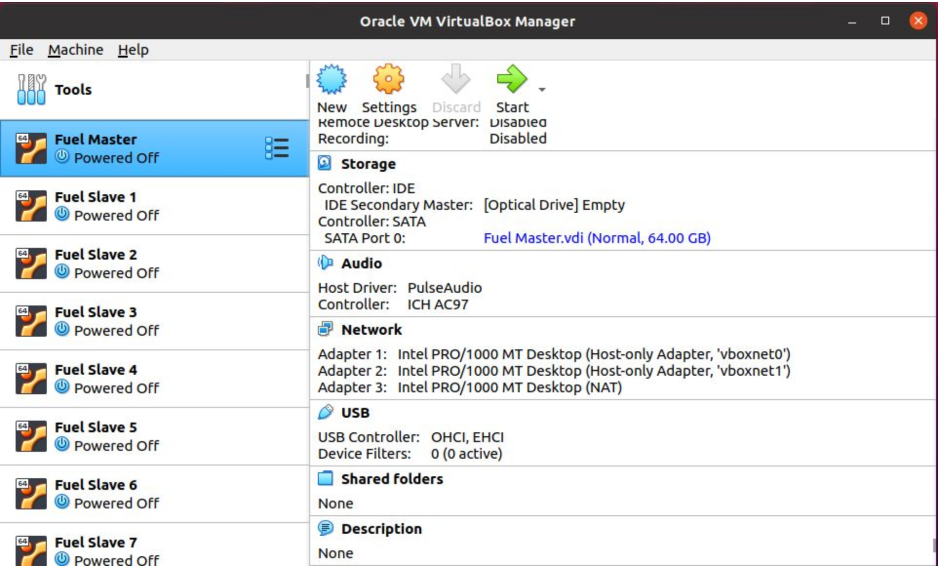
None

Description

None

Local Network

- Only the master node has a NAT adapter with DHCP enabled so that it is has open access across internet given a public static IP. Other nodes have only local network and only the master node can interact with them.



Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Secondary Master: [Optical Drive] Empty
Controller: SATA
SATA Port 0: Fuel Master.vdi (Normal, 64.00 GB)

Audio

Host Driver: PulseAudio
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (Host-only Adapter, 'vboxnet0')
Adapter 2: Intel PRO/1000 MT Desktop (Host-only Adapter, 'vboxnet1')
Adapter 3: Intel PRO/1000 MT Desktop (NAT)

USB

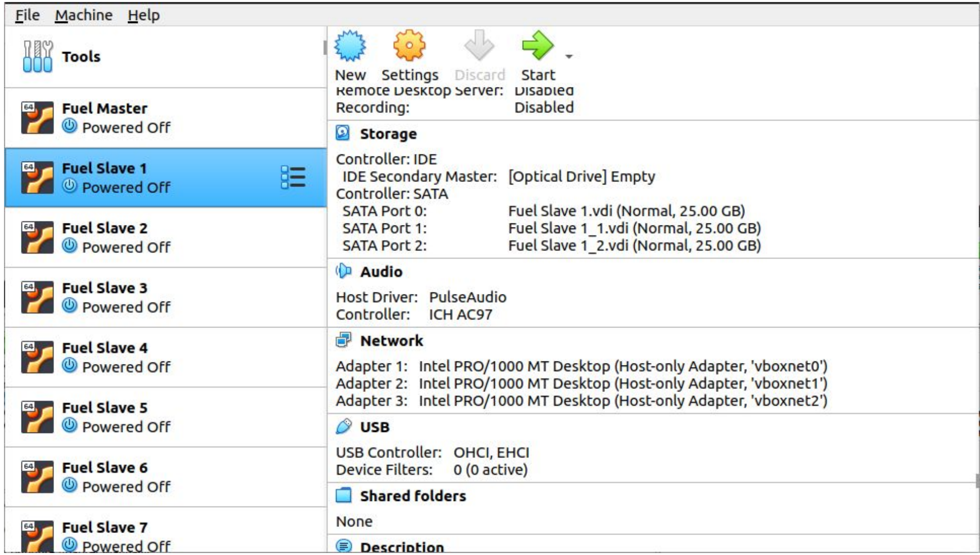
USB Controller: OHCI, EHCI
Device Filters: 0 (0 active)

Shared folders

None

Description

None



Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Secondary Master: [Optical Drive] Empty
Controller: SATA
SATA Port 0: Fuel Slave 1.vdi (Normal, 25.00 GB)
SATA Port 1: Fuel Slave 1_1.vdi (Normal, 25.00 GB)
SATA Port 2: Fuel Slave 1_2.vdi (Normal, 25.00 GB)

Audio

Host Driver: PulseAudio
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (Host-only Adapter, 'vboxnet0')
Adapter 2: Intel PRO/1000 MT Desktop (Host-only Adapter, 'vboxnet1')
Adapter 3: Intel PRO/1000 MT Desktop (Host-only Adapter, 'vboxnet2')

USB

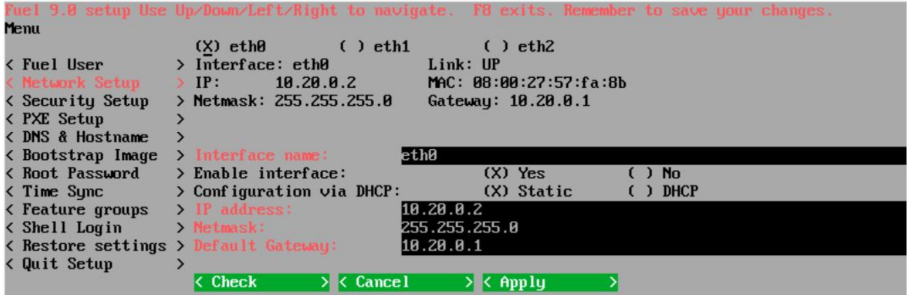
USB Controller: OHCI, EHCI
Device Filters: 0 (0 active)

Shared folders

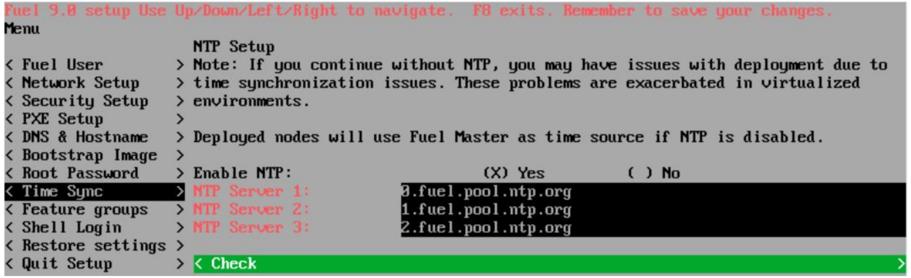
None

Description

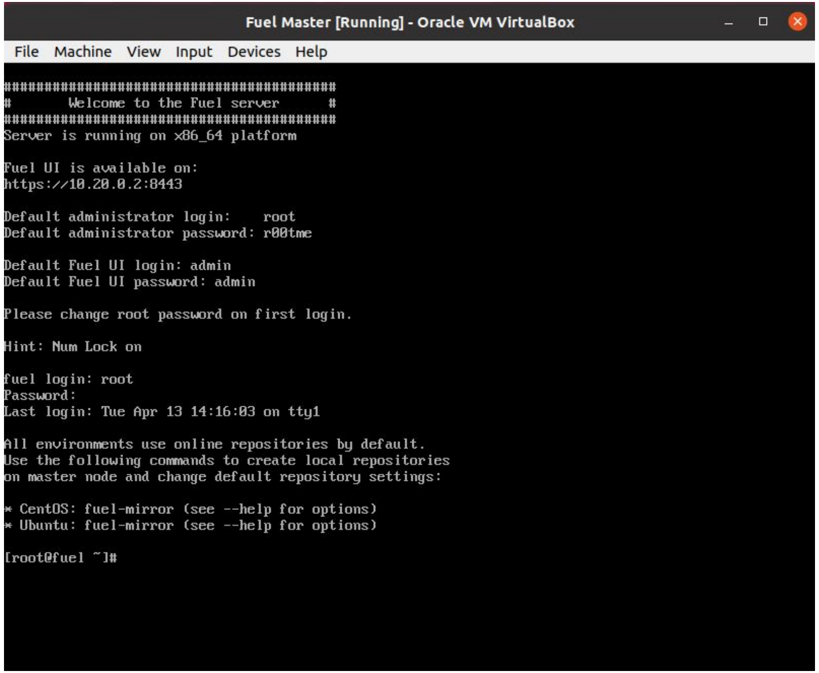
Fuel Tool (Open Source)



```
Fuel 9.0 setup Use Up/Down/Left/Right to navigate.  F8 exits. Remember to save your changes.
Menu
                      (X) eth0         ( ) eth1         ( ) eth2
< Fuel User          > Interface: eth0              Link: UP
< Network Setup      > IP:       10.20.0.2          MAC: 08:00:27:57:fa:8b
< Security Setup     > Netmask: 255.255.255.0       Gateway: 10.20.0.1
< PXE Setup          >
< DNS & Hostname     >
< Bootstrap Image    > Interface name:        eth0
< Root Password      > Enable interface:                (X) Yes          ( ) No
< Time Sync          > Configuration via DHCP:          (X) Static       ( ) DHCP
< Feature groups     > IP address:            10.20.0.2
< Shell Login        > Netmask:               255.255.255.0
< Restore settings   > Default Gateway:       10.20.0.1
< Quit Setup         >
                      < Check        > < Cancel       > < Apply        >
```



```
Fuel 9.0 setup Use Up/Down/Left/Right to navigate.  F8 exits. Remember to save your changes.
Menu
                      NTP Setup
< Fuel User          > Note: If you continue without NTP, you may have issues with deployment due to
< Network Setup      > time synchronization issues. These problems are exacerbated in virtualized
< Security Setup     > environments.
< PXE Setup          >
< DNS & Hostname     > Deployed nodes will use Fuel Master as time source if NTP is disabled.
< Bootstrap Image    >
< Root Password      > Enable NTP:                      (X) Yes          ( ) No
< Time Sync          > NTP Server 1:          0.fuel.pool.ntp.org
< Feature groups     > NTP Server 2:          1.fuel.pool.ntp.org
< Shell Login        > NTP Server 3:          2.fuel.pool.ntp.org
< Restore settings   >
< Quit Setup         > < Check                                                            >
```

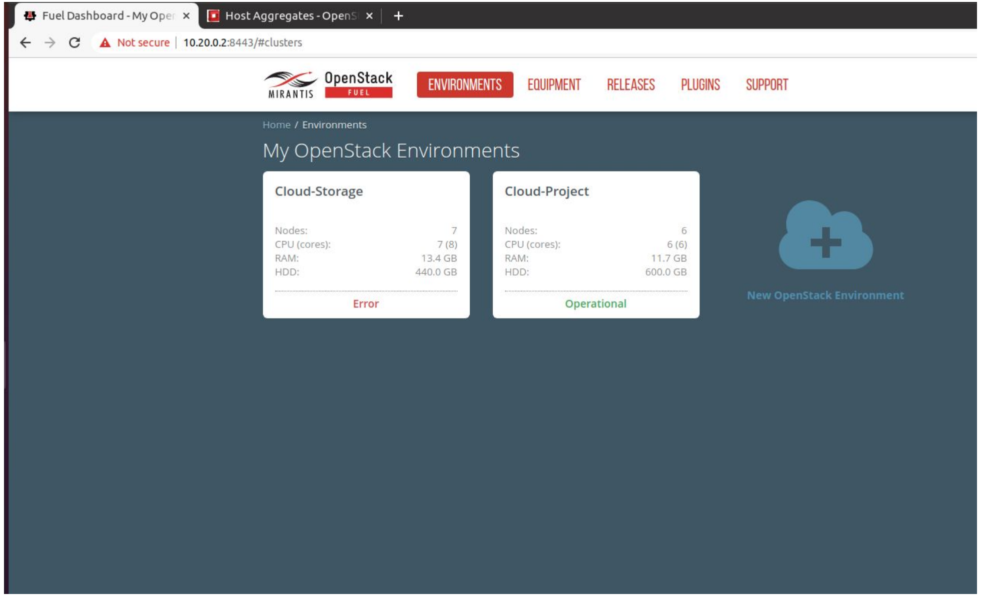


Fuel Master [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
#########################################
#       Welcome to the Fuel server      #
#########################################
Server is running on x86_64 platform

Fuel UI is available on:
https://10.20.0.2:8443

Default administrator login:    root
Default administrator password: r00tme

Default Fuel UI login: admin
Default Fuel UI password: admin

Please change root password on first login.

Hint: Num Lock on

fuel login: root
Password:
Last login: Tue Apr 13 14:16:03 on tty1

All environments use online repositories by default.
Use the following commands to create local repositories
on master node and change default repository settings:

* CentOS: fuel-mirror (see --help for options)
* Ubuntu: fuel-mirror (see --help for options)

[root@fuel ~]#
```

Fuel UI

- After Installing Fuel on the master node, the network parameters are set accordingly and the image is bootstrapped to other slave nodes.
- We can manage the nodes deployment to the openstack cloud from the Fuel UI accessible through the provides public IP address from a system on the same local network.
- Fuel can detect the nodes on the same network when booted and can use them to deploy openstack resources



My OpenStack Environments

Cloud-Storage

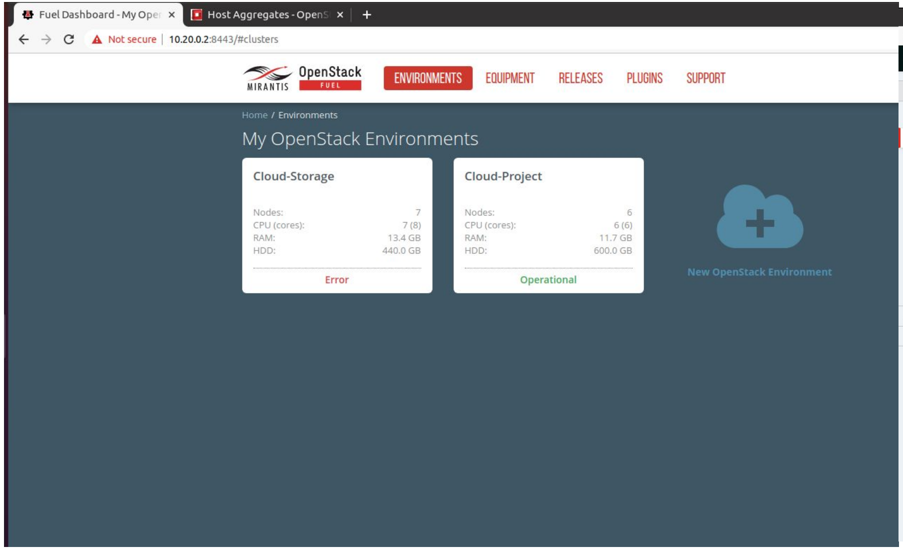
Nodes:	7
CPU (cores):	7 (8)
RAM:	13.4 GB
HDD:	440.0 GB

Error

Cloud-Project

Nodes:	6
CPU (cores):	6 (6)
RAM:	11.7 GB
HDD:	600.0 GB

Operational



My OpenStack Environments

Cloud-Storage

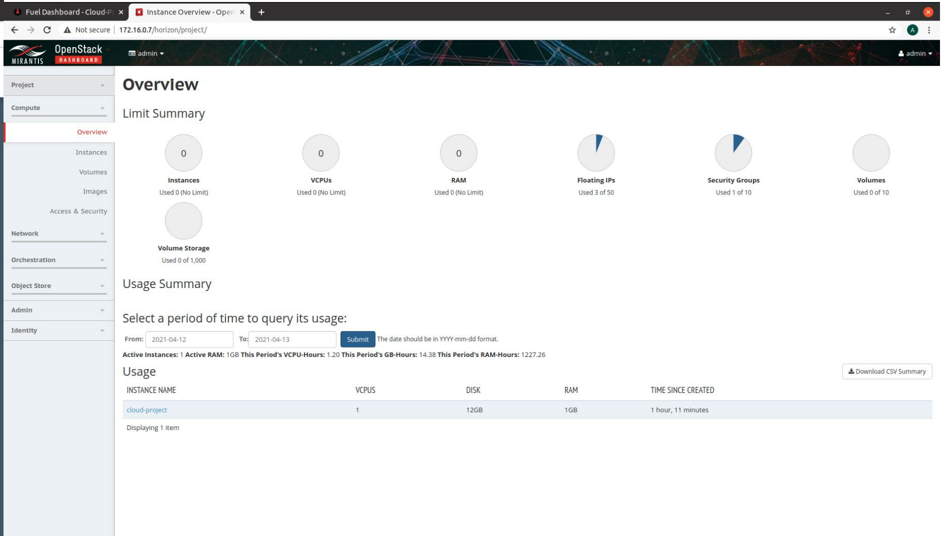
Nodes:	7
CPU (cores):	7 (8)
RAM:	13.4 GB
HDD:	440.0 GB

Error

Cloud-Project

Nodes:	6
CPU (cores):	6 (6)
RAM:	11.7 GB
HDD:	600.0 GB

Operational



Overview

Limit Summary

Instances	VCPUs	RAM	Floating IPs	Security Groups	Volumes	Volume Storage
0	0	0				
Used 0 (No Limit)	Used 0 (No Limit)	Used 0 (No Limit)	Used 3 of 50	Used 1 of 10	Used 0 of 10	Used 0 of 1,000

Usage Summary

Select a period of time to query its usage:

From: 2021-04-12 To: 2021-04-13 Submit The date should be in YYYY-mm-dd format.

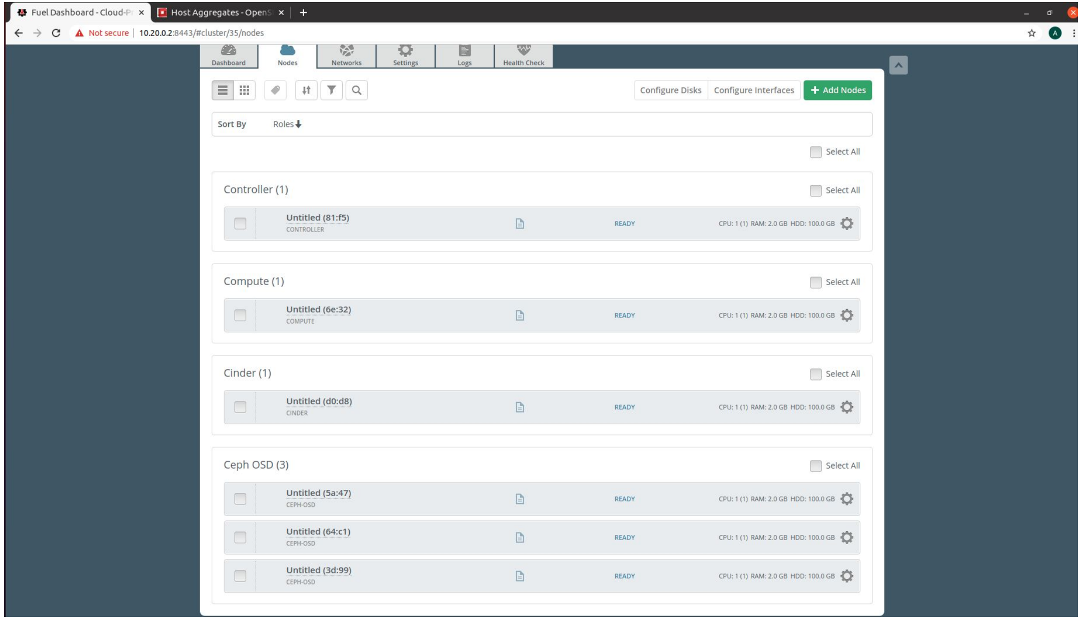
Active Instances: 1 **Active RAM:** 1GB **This Period's VCPU-Hours:** 1.20 **This Period's GB-Hours:** 14.38 **This Period's RAM-Hours:** 1227.26

Usage

INSTANCE NAME	VCPUS	DISK	RAM	TIME SINCE CREATED
cloud-project	1	12GB	1GB	1 hour, 11 minutes

Displaying 1 item

Fuel UI



Controller (1)

Untitled (81:f5) CONTROLLER	READY	CPU: 1 (1) RAM: 2.0 GB HDD: 100.0 GB

Compute (1)

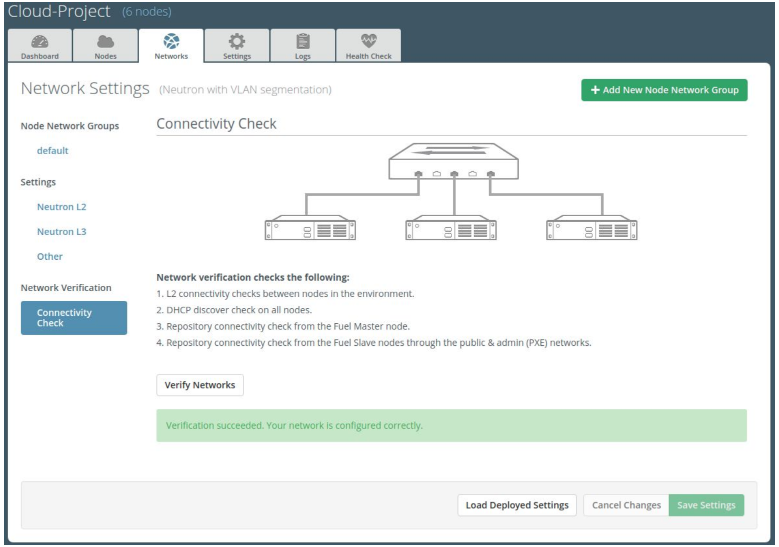
Untitled (6e:32) COMPUTE	READY	CPU: 1 (1) RAM: 2.0 GB HDD: 100.0 GB

Cinder (1)

Untitled (d0:d8) CINDER	READY	CPU: 1 (1) RAM: 2.0 GB HDD: 100.0 GB

Ceph OSD (3)

Untitled (5a:47) CEPH-OSD	READY	CPU: 1 (1) RAM: 2.0 GB HDD: 100.0 GB
Untitled (64:c1) CEPH-OSD	READY	CPU: 1 (1) RAM: 2.0 GB HDD: 100.0 GB
Untitled (3d:99) CEPH-OSD	READY	CPU: 1 (1) RAM: 2.0 GB HDD: 100.0 GB



Network Settings (Neutron with VLAN segmentation)

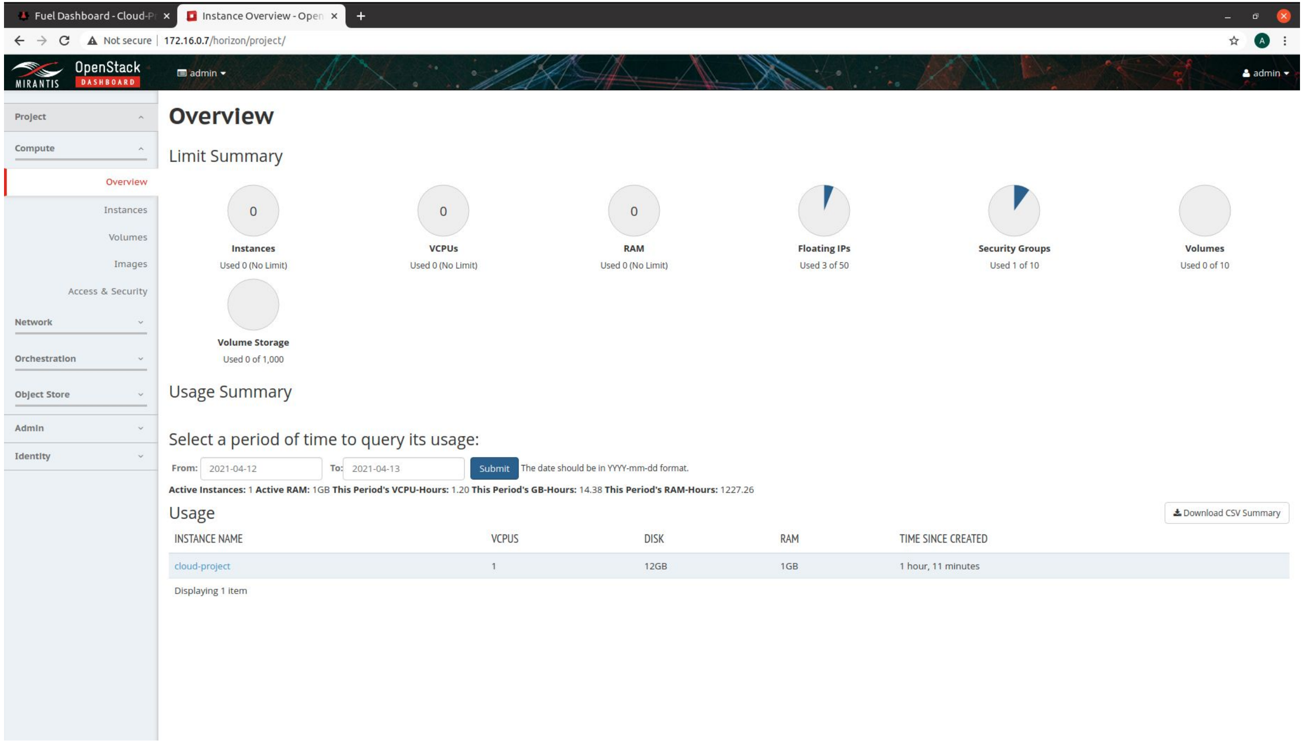
Connectivity Check

Network verification checks the following:

1. L2 connectivity checks between nodes in the environment.
2. DHCP discover check on all nodes.
3. Repository connectivity check from the Fuel Master node.
4. Repository connectivity check from the Fuel Slave nodes through the public & admin (PXE) networks.

Verification succeeded. Your network is configured correctly.

Openstack Dashboard



Overview

Limit Summary

0	0	0			
Instances	VCPUs	RAM	Floating IPs	Security Groups	Volumes
Used 0 (No Limit)	Used 0 (No Limit)	Used 0 (No Limit)	Used 3 of 50	Used 1 of 10	Used 0 of 10

Volume Storage
Used 0 of 1,000

Usage Summary

Select a period of time to query its usage:

From: 2021-04-12 To: 2021-04-13 Submit The date should be in YYYY-mm-dd format.

Active Instances: 1 **Active RAM:** 1GB **This Period's VCPU-Hours:** 1.20 **This Period's GB-Hours:** 14.38 **This Period's RAM-Hours:** 1227.26

Usage

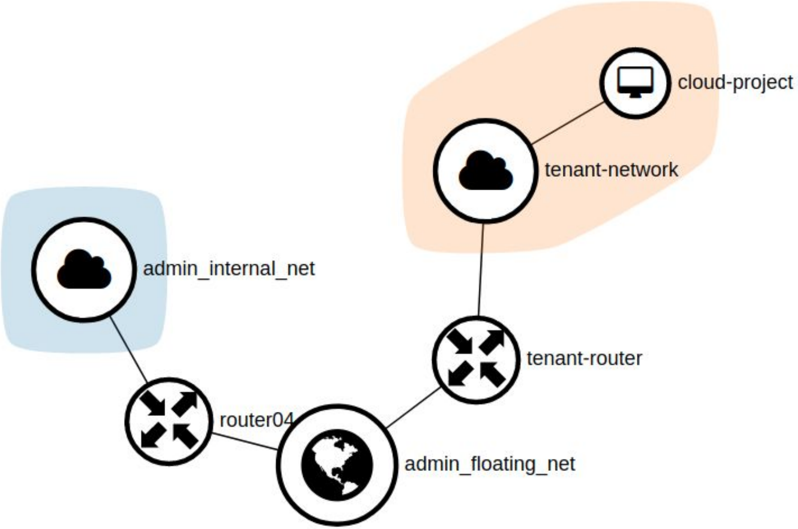
Download CSV Summary

INSTANCE NAME	VCPUS	DISK	RAM	TIME SINCE CREATED
cloud-project	1	12GB	1GB	1 hour, 11 minutes

Displaying 1 item

- The deployed openstack model can be accessed directly from the fuel UI and we can also deploy multiple instances of openstack on the cluster.
- We then create the networks and other requirements to launch the VM on the openstack

Launching the VM



- We first create networks, subnets and routers and generate floating ips to assign to the VM
- Then we generate security rules and key pairs generated by keystone for ssh into the VM after launching
- We then create a ubuntu disk image using glance and cinder to store the image
- We then launch the VM on this network and assign a public floating ip
- We then ssh into the VM from localhost and deploy our web application on the openstack VM

End
Demo