Monitoring User Resource Utilization

Web Application for Monitoring Cloud Resource Utilization

Amsal Naeem

CIDSE ASU

anaeem1@asu.edu

Parneet Kaur

CIDSE ASU

pkaur8@asu.edu

Vanthana Sachdev

CIDSE ASU

vsachde2@asu.edu

*Abstract* — In a Cloud environment, each user allocates some resources according to his/her requirements. However, there is no guarantee that these resources are sufficient or efficiently used, i.e., there could be under or over utilization. The scope of the project is to develop a web application as an interface for the user to monitor the utilization of his/her allocated resources; based on observations, suggestions will be given to the user regarding how to maximize the use of their resources or which resources to add to the pool of resources to facilitate user task completion.

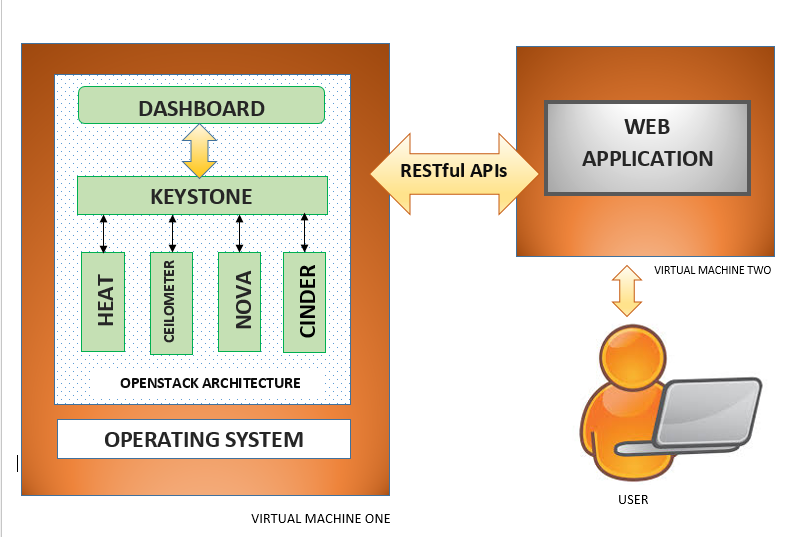
Keywords— Cloud platform, Devstack, Cloud services, Heat, Ceilometer, Web application, Python Ceilometerclient API

# Introduction

1. The problems to be addressed in this project.
   1. Monitoring resource utilization for each user.
   2. Based on statistics give suggestions to user regarding allocated resources.
   3. Provide interface to user to view the status of their allocated resources
2. Why are these problems important?
   1. Cloud infrastructure is meant to provide resources to the users. These resources need to be managed to ensure full and efficient utilization. The motivation behind this is to make sure that no resources go to waste. This will promote scalability and allow more users to take advantage of these resources.
3. Applied technologies and solutions to address these problems.
   1. Devstack (Cloud platform)
   2. Devstack/Openstack services. Predominantly Heat and Ceilometer
   3. Python Ceilometerclient API to extract data from cloud services
   4. The above extracted data is stored in SQL database for analysis and statistical report generation.
   5. Web application for user for user using HTML, CSS, php, Ubuntu Server (LAMP)
4. Expected outcomes of this projects.
   1. Web interface to monitor the utilization status of resources allocated on the Devstack/Openstack cloud platform.
   2. Users will be able to see the statistics of the resources that have been allocated to them on demand. A statistical Report, consisting of current resource utilization of demanded resources will also generated for the user to get an idea how much resources their application is consuming out of the demanded resources.
5. Project management plan
   1. (Completed) The tentative plan is to finish designing the Web Application , setting up the server LAMP (Parneet Kaur) and finish the setting up of the cloud infrastructure on Devstack, along with gathering statistics (Amsal Naeem and Vanthana Sachdev) by the Second Phase Deadline.
   2. We are presently working on populating the MySQL database with the metrics generated by python ceilometer client, making the front end of the application more interactive (Parneet). We are also researching on methods that can be used to visualize the data in a user friendly way, so that it is understandable by the user and how we can create multiple instances and create a meaning out of registering several users in the web portal. (Amsal, Vanthana, Parneet)
   3. Once we are completed with the above tasks, we will create a pictorial representation of resource utilization(for one user) and then create multiple users.
   4. The extraction of information and integration of results from Phase 2 with the front web application should be completed by the final deadline (Vanthana Sachdev, Parneet Kaur, Amsal Naeem).

# System Models

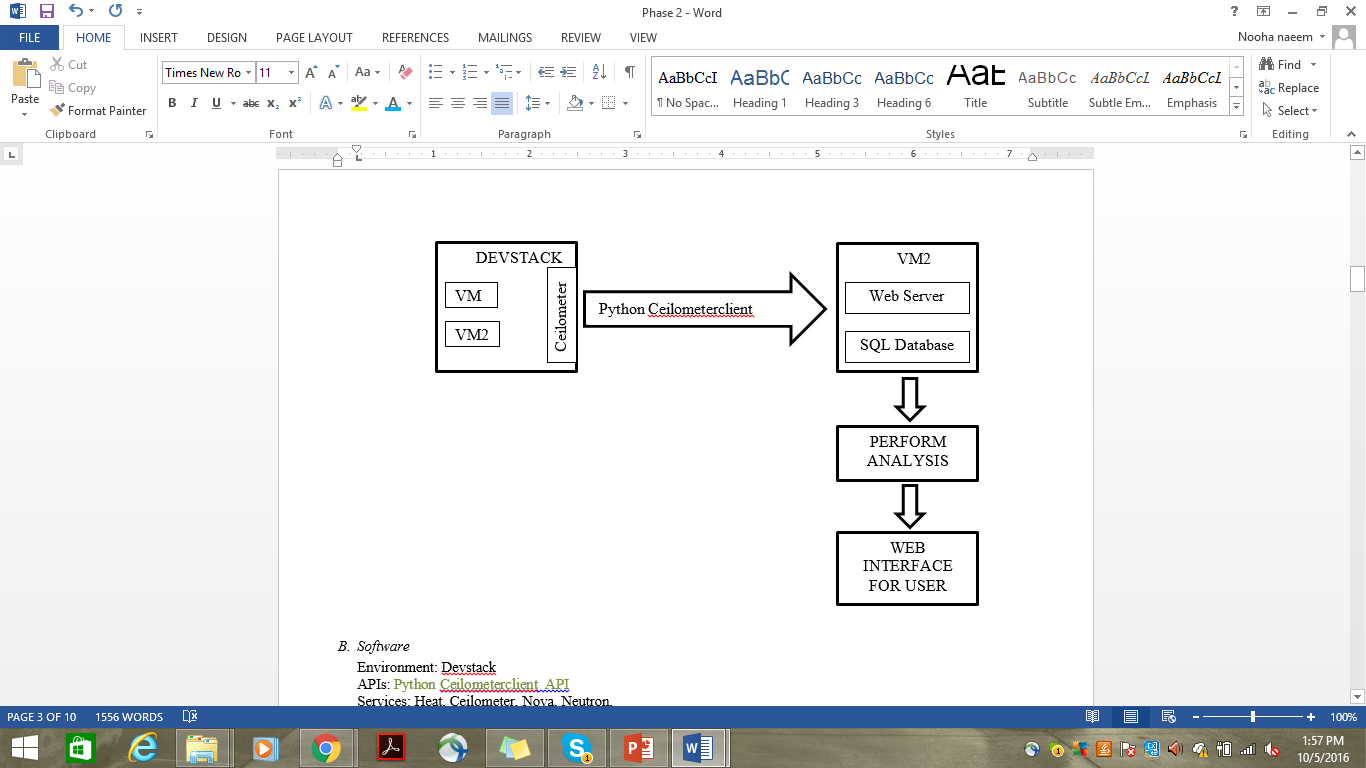
## System Model



The system model comprises of the following:

1. Cloud Environment: The Devstack platform will be used to set up the cloud environment. Heat, Nova, Ceilometer will be the main services used.

2. Web Application: The front end of the web portal will be a user interface created using HTML, CSS and JavaScript (on the front end) and on the server side, LAMP server is set up, and server side scripting will be done in php. A private database will be created to store the information of each user.



## Software

Environment: Devstack

APIs: Python Ceilometerclient API

Services: Heat, Ceilometer, Nova, Neutron.

Tools: Ububtu Server, SQL Database

Programming Language: Python, HTML, CSS, JavaScript, NodeJs

# Project Description

## Project Overview

The project goal is to provide users with a web interface to view the status of the allocated resources in a cloud. This information will help users analyze their application requirements in a better way. Devstack will be used as the cloud platform and its services will be used to gather metrics (displayed in the following diagram) for the resources. Out of the gathered data, only the data related to a particular user is made available on the web interface for the user. Also, based on the data, cloud management may generate a report suggesting the resource requirements for user application.

Resource Id

Metrics COLLECTION

image size

Samples

METERS

CPU util

memory

disk capacity

## Task 1 : Setting up Cloud environment

This task involves setting up the Devstack platform on the allocated virtual machines in the Thoth lab. Understanding Devstack architecture, configuring the required services, interacting with devstack services will also be part of this task

## Task 2: Gathering metrics

After having a good understanding of cloud, devstack services will be used to gather metrics. Mainly used services include heat, ceilometer, nova, neutron. Theses gathered metrics are stored in SQL database.

## Task 3: Web application

A web portal will be created for user to get the information of resources usage. The front end of the webportal will be a user interface created using HTML, CSS (on the front end) and on the server side, A LAMP Server is set up. We will be linking the front end of the web portal with the back end using php. A private database has been set up in MySQL. We are currently connecting the DB to the output we got from devstack. We are completed with the designing of web pages at the front end. Once we get more instances at the devstack, we will proceed with the web app.

Task 4: Integration.

This is the last and most important phase of the project. This involves integrating front end application and the under laying metrics collection on cloud. The monitored statistics that are stored in database are retrieved and after processing they will be displayed to user through the web interface.

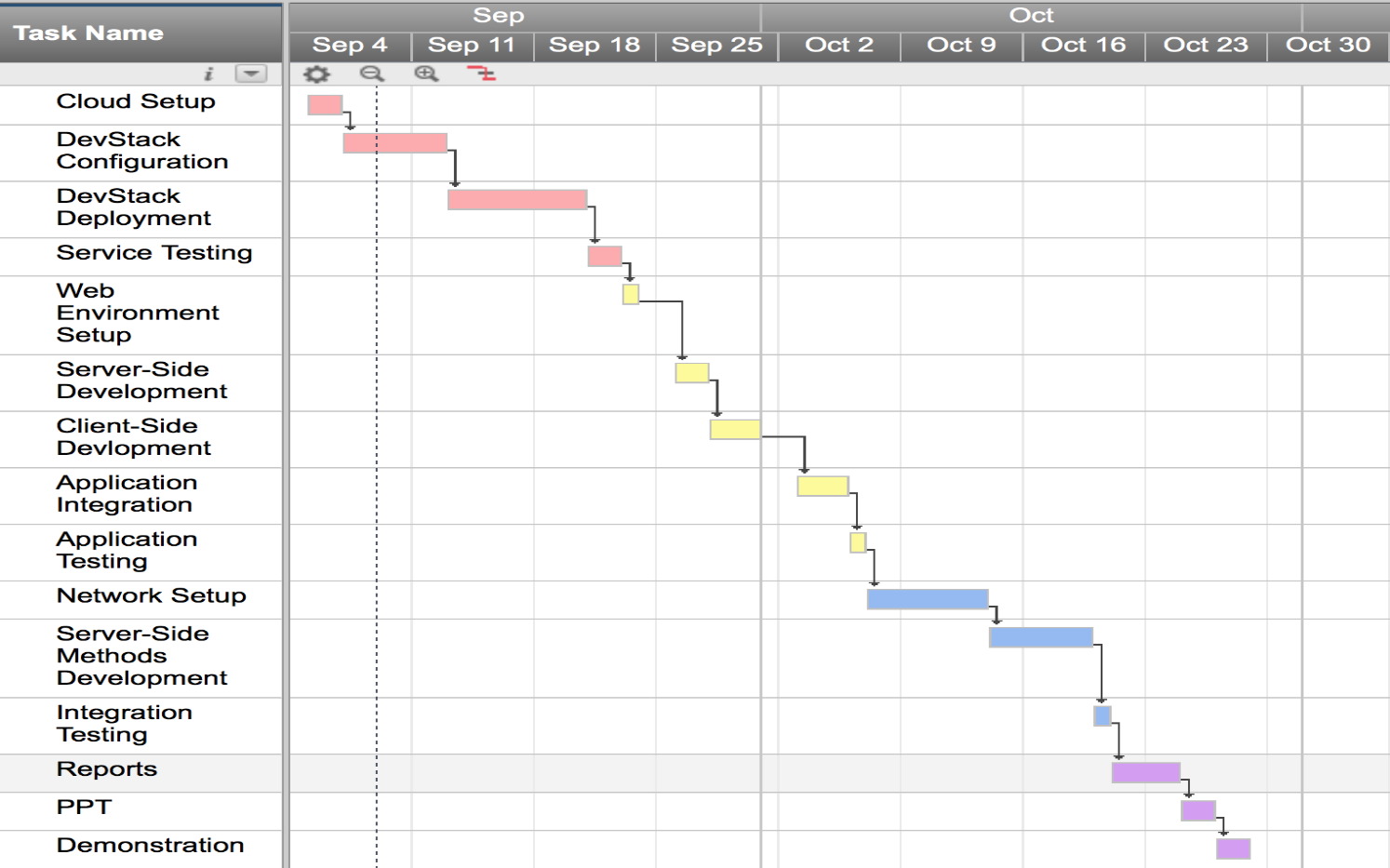
## Project Task Allocation

| **Task** | **Sub-Task** | **Team Member** | **Work Percentage** |
| --- | --- | --- | --- |
| Cloud Setup | Setup | Vanthana/Amsal | 7.5 |
| DevStack Configuration | Vanthana Sachdev | 10 |
| DevStack Deployment | Amsal Naeem | 10 |
| Services Testing | Amsal/Vanthana | 7.5 |
| Web Application Setup | Environment Setup | Parneet Kaur | 3 |
| Server-Side Development | Parneet Kaur | 8 |
| Client-Side Development | Parneet Kaur | 8 |
| Application Integration | Amsal /Vanthana | 3 |
| Software Testing | All | 3 |
| Connecting the Web application and cloud | Network Setup | All | 12.5 |
| Server-Side Methods Development | All | 12.5 |
| Integration Testing | All | 5 |
| Documentation | Reports | All | 3 |
| Power-point Presentation | All | 3 |
| Demo | All | 4 |

## Deliverables

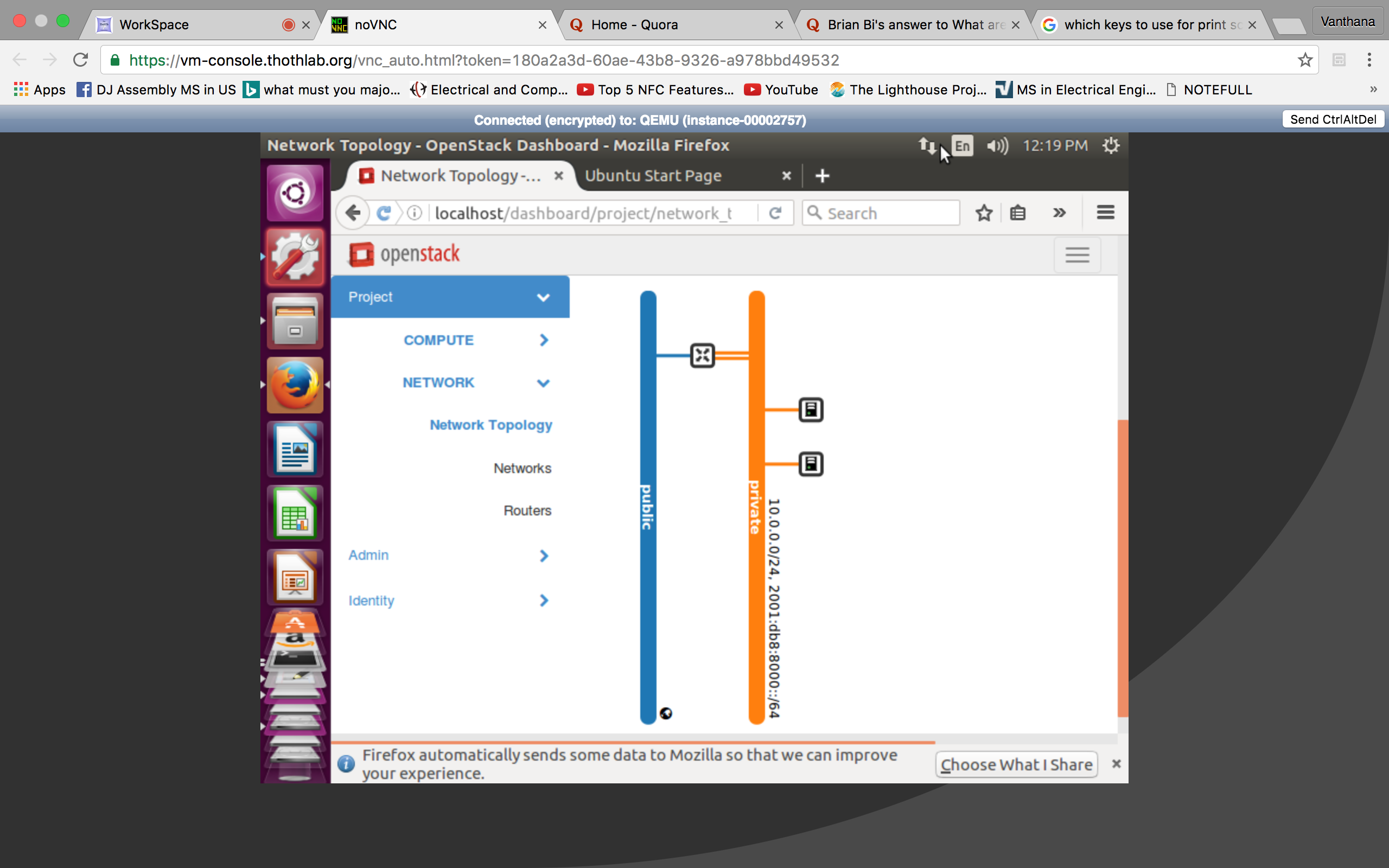
Web Portal to access cloud, backend monitoring service and a detailed Readme.

## Project Timeline

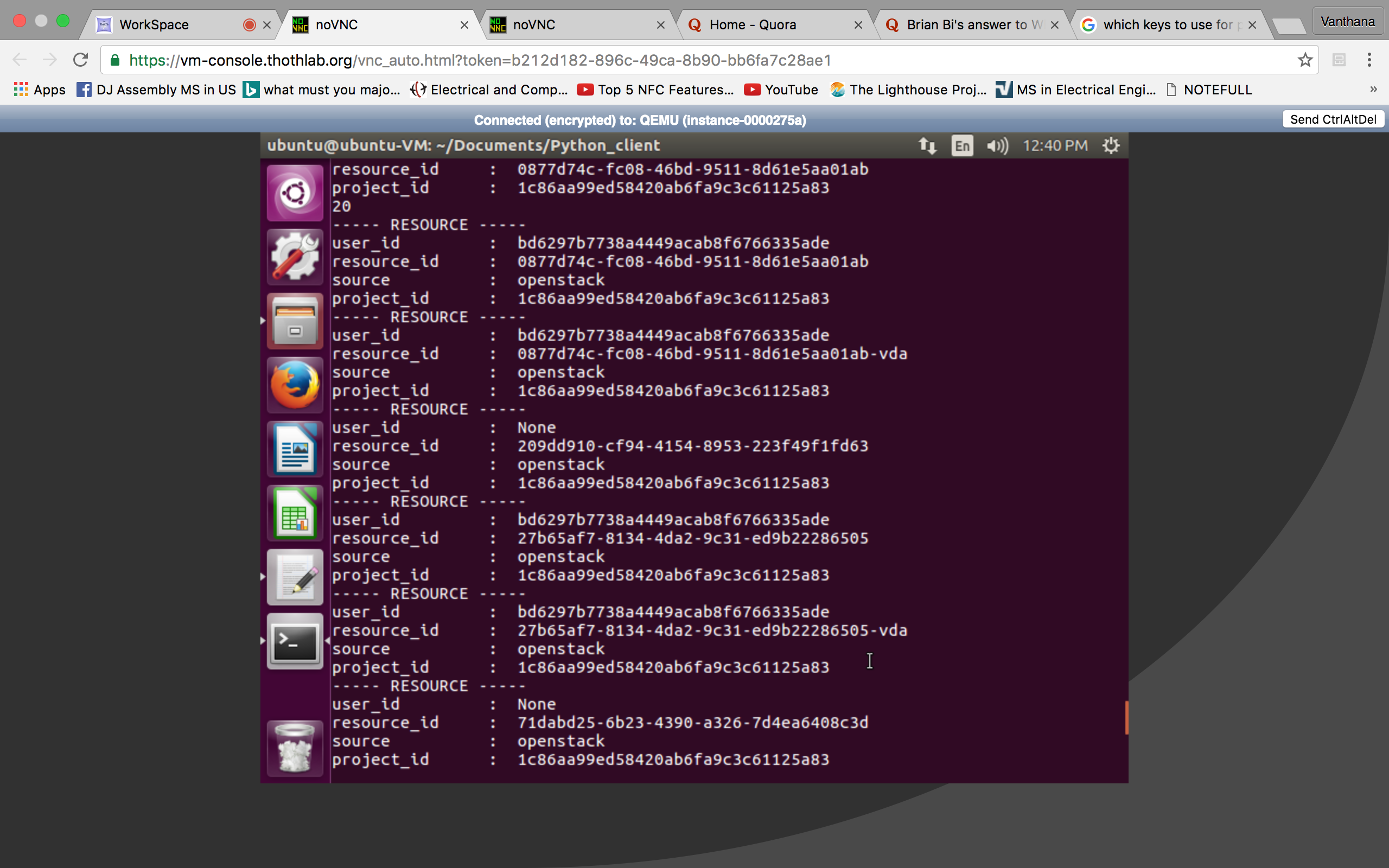


## Progress till Phase 2

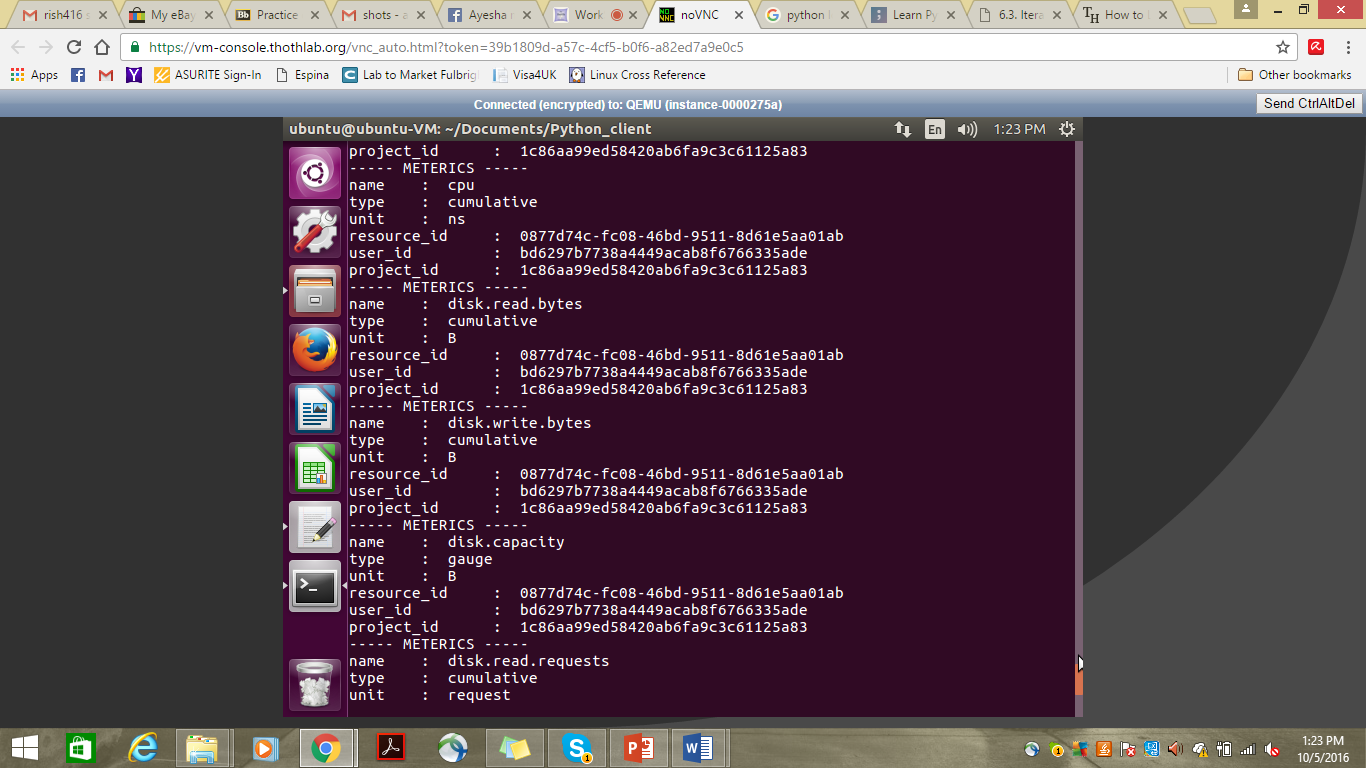
* + DevStack (the cloud environment) has been setup on the devstack1 machine provided on Thoth lab.
  + Following is the current network topology that we are working for now



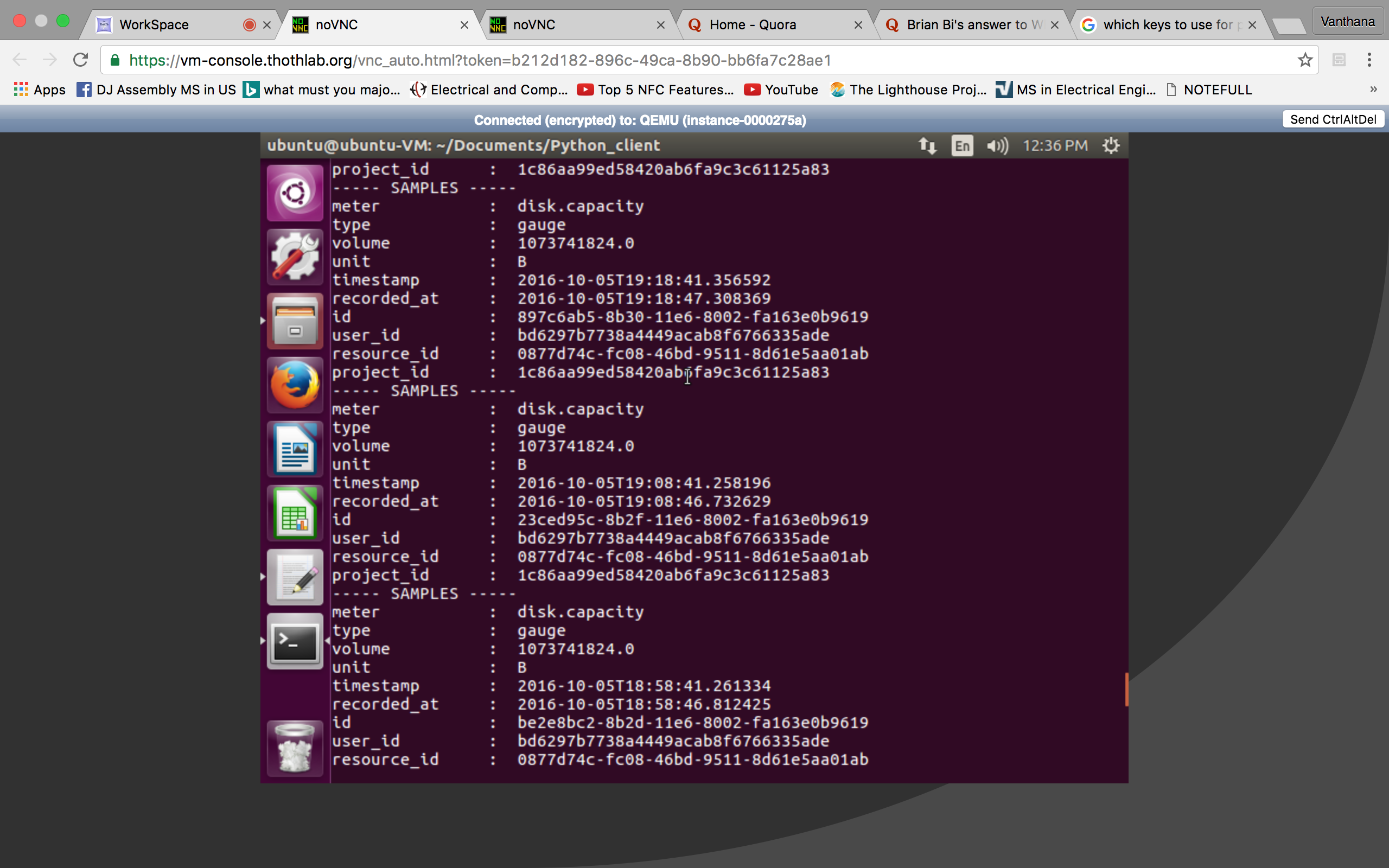
* + Ceilometer services were enabled in the local.conf file while doing stack.
  + Ceilometer services can be used on devstack1 through command line interface.
  + Using Python Ceilometerclient API on VM1 (on Thoth lab), ceilometer metrics from Devstack can be collected.
  + We collected some metrics as follows
    - We can get all the resources that are in the devstack



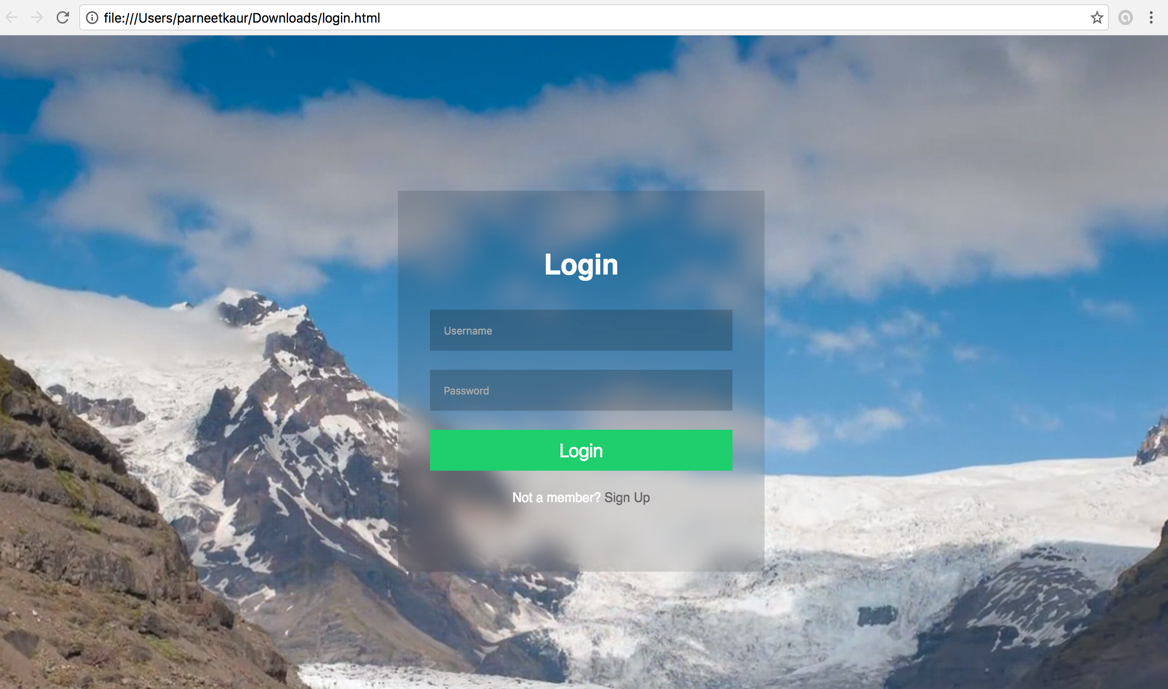
* + - For every resource we can get metrics associated with it using the resource id. Different resources have different meters associated with it, so depending on type of resource we collect the meters and samples for analysis.



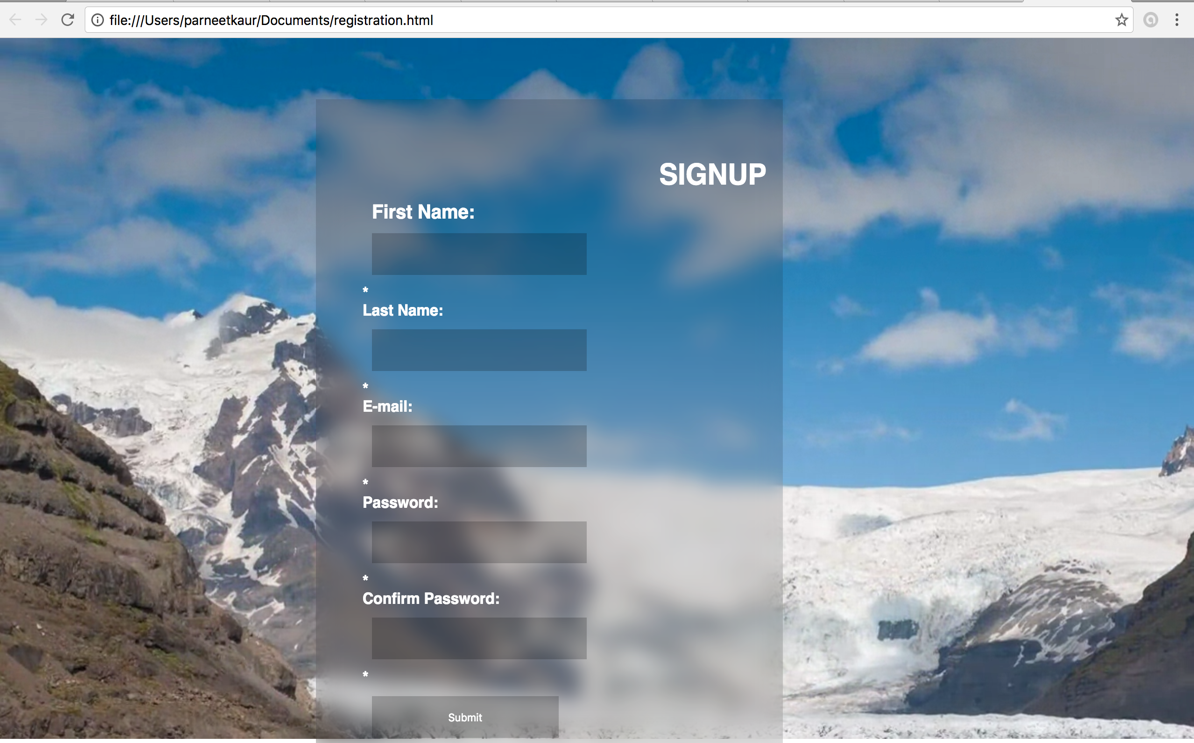
* + - For every meter we can get samples i.e. the meters saved at various timestamps. And using this data we can analyze the utilization over a period of time.



* + SQL database has been installed on VM2 (So we can work in parallel on front end).
  + We are able to connect python to SQL database and put the data in the database tables.
  + Web pages for the web portal have been designed.
  + Currently working on installing python Ceilometer API client on VM2 and then directly populating database with the devstack ceilometer metrics from the devstack machine.



* This is the front page of the web portal, where a user can login and view the details of resources’ utilization by him.
* If the user is registered, he will navigate to the page containing the pictorial information about his utilization of resources. Or if the user is a new user, he can click on sign up and get navigated to the page where he can register as a new user.



* This is the page where the user can register as a new user.

# Risk Management of the project

|  |  |  |
| --- | --- | --- |
| **Risk Description** | **Risk** | **Mitigation Strategy** |
| Network Connectivity | High | Periodically checking n/w connectivity |
| Machine Overload | High | Use machine with higher configuration |
| Data Loss | Medium | Implementing a cloud based data backup |

# Conclusion

The need of the hour is efficient and cost effective utilization of limited resources to build and complete tasks in a scalable manner, especially since most companies and institutions are offloading their work to Cloud. To address this problem, which is and will in the future gain more notoriety, we are building a Web Application to monitor resource utilization.

The future scope is extending the application to monitor resources on per hour basis.

##### Acknowledgment

We sincerely thank our professor, Dr. Dijiang Huang, for giving us the opportunity to work on a project utilizing an open-source cloud platform, which will aid our careers in the field of upcoming technology. We would also like to thank our Teaching Assistant, Mr. Ankur Chowdhary, for assisting us in our endeavor to build a good solution for monitoring cloud resources and for patiently answering all our questions and clearing all our confusions.

##### References

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.