



# UPPSALA UNIVERSITET

Applied Cloud Computing

Uppsala University – Autumn 2020  
Assignment 2

---

Raheel Ali

## PART 1

**Are VMs slower than the physical machine? If yes, explain the reason. Are there alternative to VMs? How do you compare them with VMs?**

**Answer:** Yes, VMs are slower than the physical machine. The physical machine is better at instantiating classes and they do this through memory management hardware and VMs do this through the software which has partial hardware assistance.

Alternative to VMs are containers which can enable a lot more applications in to a physical server. With VMs the hardware is being virtualized to run multiple instances of OS whereas the containers provide a way to virtualize an OS due to which it can run single OS instance with multiple workloads.

## PART 2

### Task-0 Setting the environment for API access

#### 1.1 What version of the API are we using?

**Answer:** We are using API version 3.

#### 1.2 Explain how the communication works in OpenStack?

**Answer:** OpenStack is divided into different components that work together and these services are integrated by different API's. Horizon provides an end user and administrator interface to other services. Nova compute retrieves images and user requests on virtual machines, Neutron network provides virtual machines as a service, Swift provides object storage, Glance provides a catalog and a repository for images and Cinder block storage provides continuous storage for VMs hosted on the cloud.

#### 1.3 Can we use EC2 and S3 APIs to communicate with OpenStack?

**Answer:** Yes we can use EC2 and S3 API by using restful APIs to communicate OpenStack.

### Task-2 Single-machine contextualization

#### 2.1 Explain the output?

**Answer:** The output of the script is a Cowsay server printing Hello student.

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 175
Server: Werkzeug/1.0.1 Python/3.6.9
Date: Sun, 20 Sep 2020 15:09:13 GMT

< Hello student >
-----
      \   ^__^
       (oo)\_____)
            (_____)
                ||----w |
                ||     ||
```

Figure 1: Cowsay Program

## 2.2 What is contextualization?

**Answer:** Contextualization is a term used in cloud computing which refers to configuration of individual components of an application providing specific environment.

## 2.3 What language is use to prepare CloudInit configurations?

**Answer:** YAML is the language that is used to prepare CloudInit configurations.

## 2.4 What are the variants of CloudInit package?

**Answer:** Puppet or Chef are the variants of CloudInit package.

## 2.5 Can we run CloudInit scripts without booting an instance?

**Answer:** Yes, We can run the CloudInit scripts without booting an instance with the command `cloud-init init`. This command runs the modules from `cloud_init_modules`.

## 2.6 What limitation you can anticipate with the CloudInit package?

**Answer:** Limitations with CloudInit package is that it can only be booted in the start of the instance but cannot be rebooted again while instance is in running process.

## Task-4 Orchestration using Heat

### 4.1 What language is used with the Heat service to define the templates?

**Answer:** YAML language is used with the Heat service to define the templates.

### 4.2 What are the advantages of using templates rather than the APIs?

**Answer:** Advantage of using template rather than APIs is the ability to create and define custom template properties. In template we can change values of few variables for doing contextualizing and templates are independent of Cloud provider.

### 4.3 Explain the different sections in the templates?

**Answer:** There are three main sections of the templates. Resource allocations, parameters and output. Resource allocation section is used to define routers ports, security groups and scripts that are to be run on the instance. In parameters section we define what kind of image, flavor, key and network type are to be used. In output section we get attributes of the created instance.

## Task-5: Introduction to Linux Containers

### 5.1 In what category of virtualization do containers fall?

**Answer:** OS-level virtualization

### 5.2 What are the other frameworks that provide container technology. Write at least two name.

**Answer:**

- 1) Kubernetes
- 2) Azure Container Service

**5.3 Explain the provided Dockerfile. What does it do? How does it work? Write a brief (one line) description about each line in the Dockerfile.**

**Answer:** In this Dockerfile we create the application of cowsay through container. In the start we update all the required libraries, then install git, python 3, pip3, flask and cowsay. Afterwards we clone the github repository files and move the file to work directory of cowsay, we then use expose command to open the port 5000 and give the environment path using ENV. In the end using CMD command to run app.py program using python 3.

**5.4 Write a brief (one line) description about each command used in Step-2-2.**

**Answer:**

1) Edit the file sudo nano etc/docker/daemon.json and paste the following lines.

{ "mtu"=1400 }

2) Restart the system

sudo systemctl restart docker

3) Execute the docker build on the same path

docker build -t cowsay:latest .

4) To access the container

sudo docker run -it ubuntu

5) To output cowsay from the terminal

curl -i http://130.238.29.36:5000/cowsay/api/v1.0/saysomething

**5.5 What is dockerhub? Write a brief description of how can we use dockerhub for our newly build CSaaS container?**

**Answer:** We can say dockerhub acts like a GitHub where we pull and push our code, moreover it is a repository for the users to store data like container images. By creating and logging account then pushing through IP command we can make the dockerhub active for CSaaS container.

**5.6 Write a CloudInit script that contextualize a VM based on the steps (Step-1 and 2) mentioned in this task. Submit the script with your assignment report.**

**Answer:**

```

#cloud-config

apt_update: true
apt_upgrade: true
packages:
- python3-pip
- python3-dev
- build-essential
byobu_default: system

write_files:
- path: /home/ubuntu/Dockerfile
  content: |
    FROM ubuntu
    RUN apt-get update
    RUN apt-get -y upgrade
    RUN apt-get install -y git
    RUN apt-get install -y python3-pip
    RUN pip3 install --upgrade pip
    RUN pip3 install flask
    RUN apt-get install -y cowsay
    RUN git clone https://github.com/TDB-UU/csaas.git
    WORKDIR /csaas/cowsay
    EXPOSE 5000
    ENV PATH="${PATH}:/usr/games/"
    CMD ["python3", "app.py"]

- path: /etc/docker/daemon.json
  content: |
    {
      "mtu": 1400
    }

runcmd:
- sudo bash
- curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
- add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
- apt-get update
- apt-get install -y docker-ce
- cd container
- docker build -t cowsay:latest /home/ubuntu
- docker run -d -p 5000:5000 cowsay

```

---

Figure 2: CloudInit Script

## TASK 2

### Part 1

In the Raheel-OpenStack-API-Lab2-PartI-2020 submitted on student portal.

## Part 2

**Write half a page mini-essay where you highlight some of the key issues that can hinder cloud interoperability and also suggest possible solutions to those issues.**

There are some of the key issues that hinder cloud interoperability. Security issue has played the most important role in hindering cloud computing. Factors that influence security are data loss, phishing, botnet, availability and reliability, these factors can have a serious threat to organization or an application. [1]

Several techniques can be used to tackle this problem. Data confidentiality should not be comprised, that means the data stored on the cloud should be kept private. Data owner should only be allowed to have the control and authorization over the data. [2]

Another issue that occurs in cloud interoperability is that each cloud that is setup has its own way on how cloud applications interact with the cloud. This factor can have an adverse effect on the cloud resulting in vendor locking, which prohibits the ability of users to choose from alternative vendors/offering simultaneously. Sometimes cloud APIs makes issues for integrating cloud services with the legacy systems of the companies. There should be a smooth system between the cloud and local applications for the transfer and access of data. Poor interoperability can result in ambiguous data used by the cloud and will encounter difficulties in integration of the relevant APIs. [1] If the user decides to export or move it's data from one cloud to the other cloud then there can be compatibility issues due to which it would be difficult to transfer that data. The solution to these kind of some issues in my view is that all the cloud services and applications should be on same platform and up to date. The legacy systems should be updated according to meet new APIs. The cloud service providers should readily update their application to be sure they are protected and ensure to minimize these kind of issues. Reuse of good standards and gap analysis can be done to minimize these issues. There is a platform called cloud computing interoperability forum where the vendors can discuss their problems and come up with specific solutions. [3]

## Part 3

**In this part, you need to design and explain a highly available cloud native application architecture for CSaaS (CowSay-as-a-Service). Your task is to draw an architectural diagram and write a paragraph explaining your design choices.**

This architecture represents the communication of CowSay service, in this diagram when a user tries to access this service, the request will be served by Kubernetes controller. The cloud service launches a series of machines linked together. This machine runs kubernetes for the process. Two virtual machines are running with 4 nodes each which help to run the CowSay containers. These containers that can also share different kind of resources between themselves e.g. networking and shared resources. In this diagram there are different proxy servers, availability regions due to which if one region is not available than the other regions can be used for the communication of data.

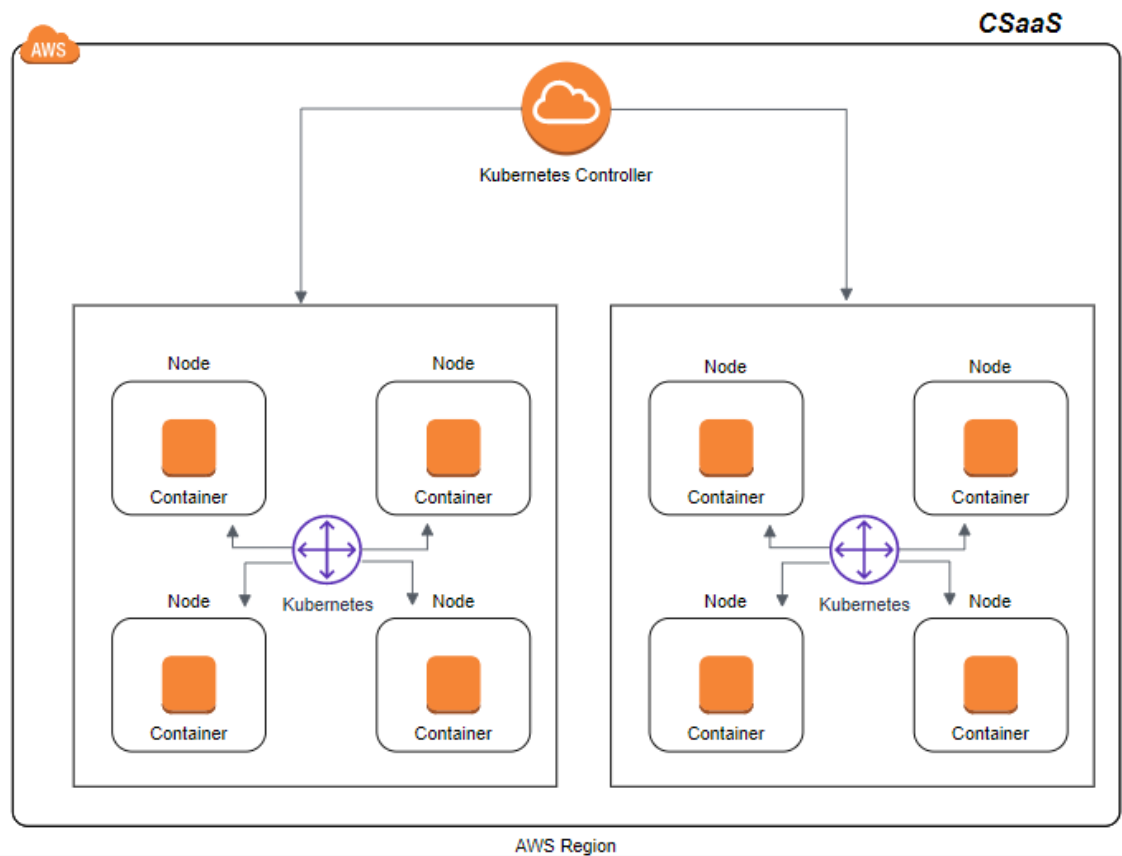


Figure 3: Architecture Diagram

## References

- [1] <http://sistemas-humano-computacionais.wdfiles.com/local-files/capitulo:sistemas-de-servico/Dillon2010.pdf>
- [2] [https://www.researchgate.net/publication/329906249\\_Security\\_Issues\\_and\\_Challenges\\_in\\_Cloud\\_Computing\\_A\\_Review](https://www.researchgate.net/publication/329906249_Security_Issues_and_Challenges_in_Cloud_Computing_A_Review)
- [3] <http://basconsulting.in/downloads/computer/cloud%20computing.pdfpage=21>