CSC 547 - Cloud Computing - Spring 2018

Container as a Service(Caas)

Team 0

Bhushan Thakur
Harshini Kadakol
Pavithra Iyer
Gokul Yelchuru
Kashish Aggarwal
Ragavi Raman

Problem Statement

- At present, the company offers Infrastructure as a Service (IaaS) cloud options.
- Investigate Container as a Service(CaaS) in AWS.
- Build cost effective prototype of CaaS solution for the company in sandbox provided.

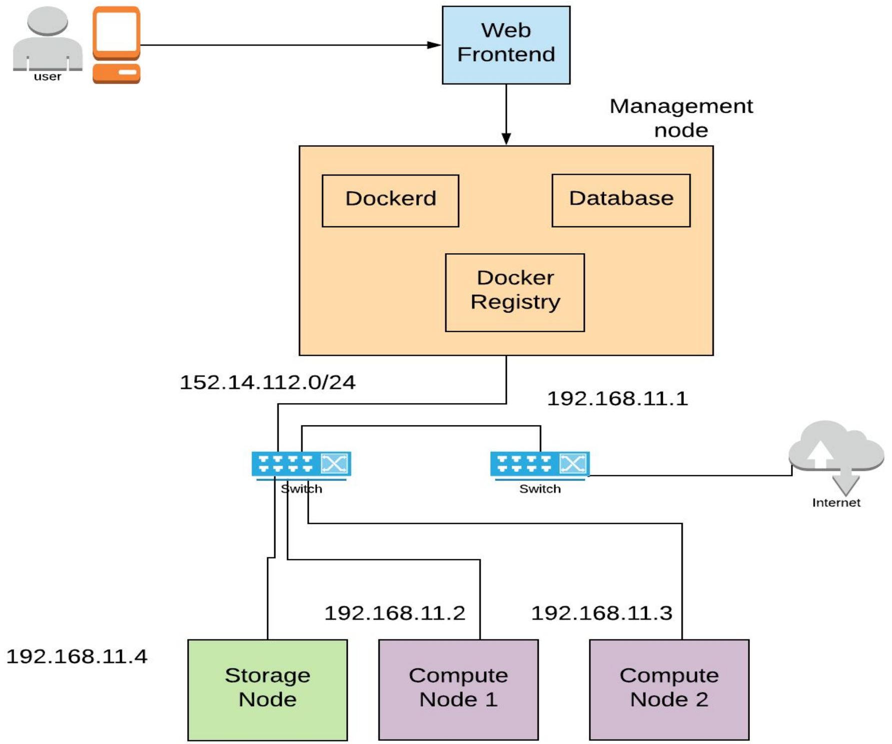
Functional Design Requirements

- User authentication
- Scheduling, Load Balancing and Image Reservation based on availability
- Ability of the user to manage his/her reservation
- Ability of the admin to monitor the cloud setup
- Choose between customer or cloud provider’s location for storage

Non-functional Design Requirements

- Cost
  - $.50/hr for a core, No network charge
- Usability
  - User Friendly, Well designed Web Interface
- Security
  - Private and Public IP addresses
  - IPtables rules at management node
- Performance
  - REST API based application
  - Avoid Resource over-provisioning - controlled through load balancing

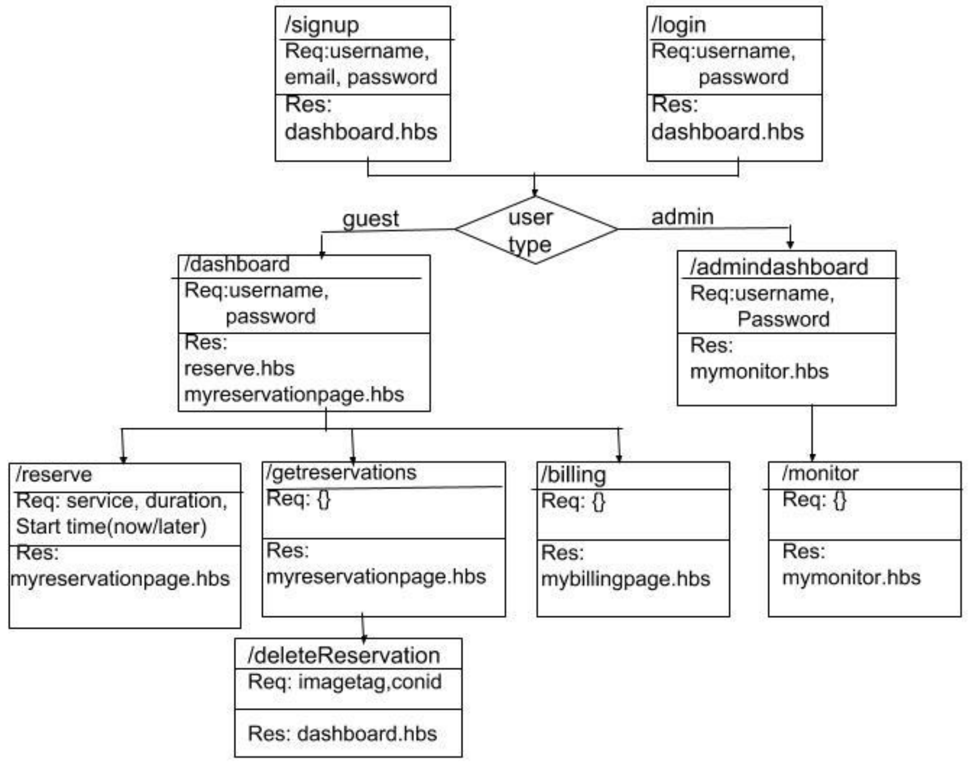
System Design and Architecture



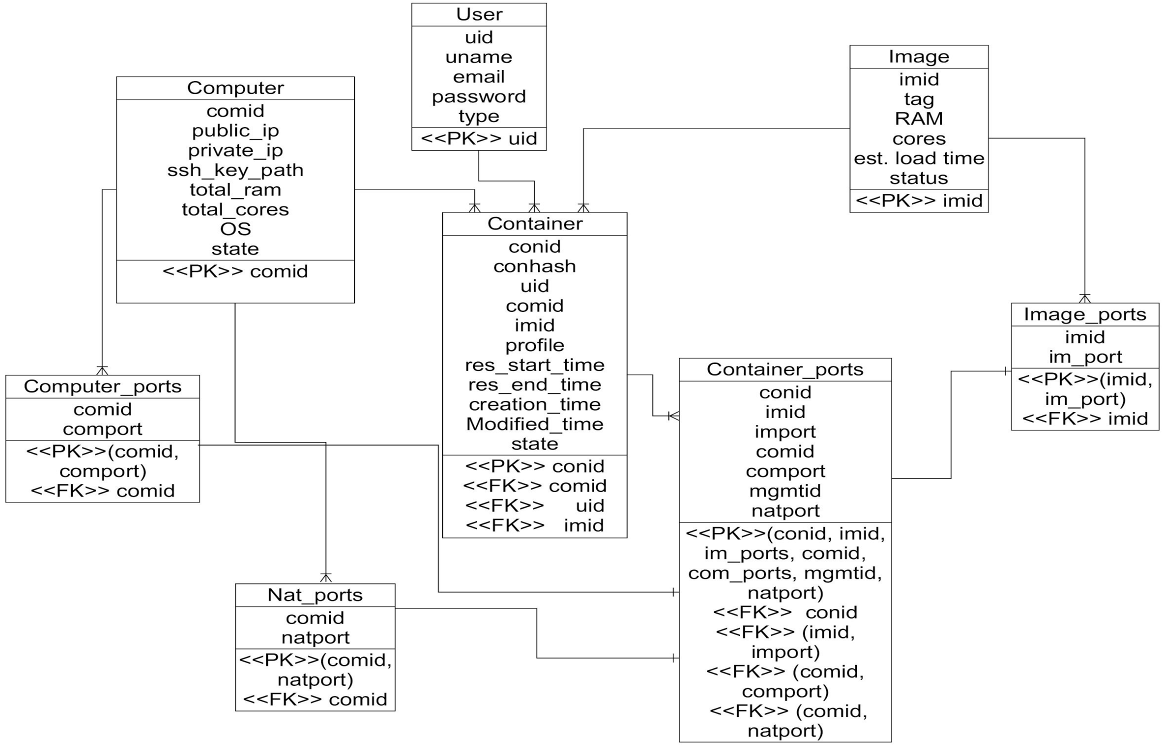
System Environment

	Dell T610	Dell T610	Dell T3400	Dell T3400
OS	Ubuntu 16.04	Ubuntu 16.04	Ubuntu 16.04	Ubuntu 16.04
RAM	23GB	23GB	3.8GB	3.8GB
Storage	900GB	900GB	150GB	150GB
Processor	2.40GHz	2.40GHz	2.40GHz	2.40GHz

System Functional Diagram



Database Design



Design Aspects: Load Balancing/Scheduling

- DB stores hardware requirements for different image services.
- First idle compute node will be selected.
- Otherwise, the compute node with minimum ram usage will be selected. (Ram_in_use/Total_RAM)

Note: Available resources for different compute nodes are compared only for the reservation interval requested.

Design Aspects: Port Mapping/Networking

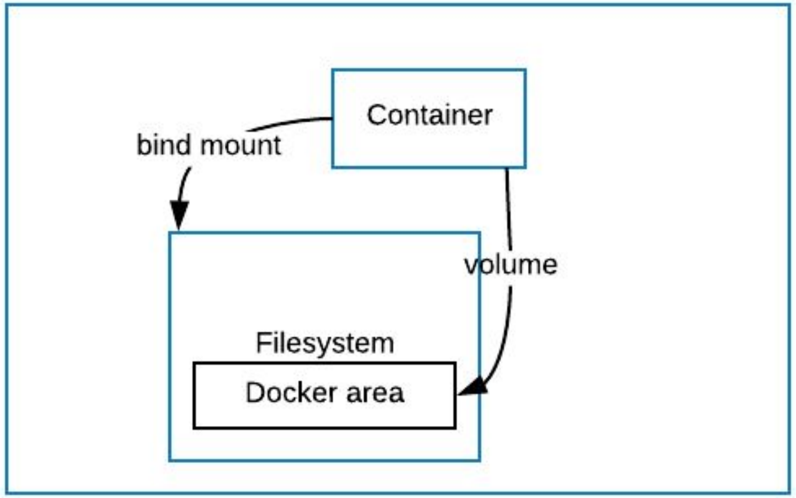
- Expose the container port by mapping it to compute node port
- iptable rule in management node

```
iptables -t nat -A PREROUTING -d <mgmtip> -p tcp --dport
<natport(mgmtnode)> -j DNAT --to-destination <compute node private
ip>:<compute node port>
```

- User hits at Management node IP and NAT PORT

Design Aspects: Storage (In Progress)

1. Different storage requirements for different customer profiles.
2. Dedicated storage node for storing data at cloud provider's location.
3. Exploring docker-machine to mount remote storage on local compute node.
4. Start container with the bind mount on your local system.



Schedule

Task	Start date	End date
Research about Docker and understand the working and installation	3/17/18	3/23/18
Initial UI design plan	3/23/18	3/27/18
Determine the system requirements and the cloud network setup	3/27/18	4/1/18
Install Ubuntu 16.04 and Docker on lab machines	4/2/18	4/7/18
Set up network functionalities on lab machines	4/7/18	4/8/18
Setting up database and docker registry on digital ocean	4/2/18	4/9/18
Decide the services offered to the customer	4/7/18	4/11/18

Schedule

Task	Start Date	End Date
User container reservations	4/13/18	4/18/18
User login and signup	4/17/18	4/19/18
Show and delete reservations	4/19/18	4/22/18
Container health check (monitoring)	4/19/18	4/22/18
Network and scheduling components	4/15/18	4/20/18
User Billing	4/22/18	4/23/18
Research on storage requirements	4/13/18	-
Verification and validation	4/22/18	-
Performance comparison with AWS	4/21/18	-
Migration to lab machine	4/22/18	-
Documentation	3/20/18	-

Tasks and Responsibility

Task	Members
Research about Docker and understand the working and installation	All members
UI design	Bhushan
Determine the system requirements and the cloud network setup	All members
Install Ubuntu 16.04 and Docker on lab machines	Ragavi, Gokul, Harshini, Pavithra
Set up network functionalities on lab machines	Gokul, Pavithra
Setting up docker registry on digital ocean	Bhushan, Pavithra
Decide the 5 services offered	Ragavi, Harshini
Set up database schemas	Bhushan, Pavithra, Kashish, Harshini
User container reservations	Bhushan, Pavithra
User login and signup	Harshini, Bhushan

Tasks and Responsibility

Task	Members
Show and delete reservations	Pavithra
User and admin dashboard functionalities	Bhushan
Container health check (monitoring)	Pavithra, Bhushan
Network components	Gokul
Scheduling components	Ragavi, Kashish
Research on storage requirements	Kashish, Gokul
User Billing	Pavithra
Verification and validation	All members
Migration to lab machine	All members
Performance comparison with AWS	Harshini, Ragavi
Documentation	All members

Verification and Validation

TestCase#	TestCase	Expected Observation
Network Setup		
T1	Proper configuration of Private and Public IPs	Management node should access compute nodes via private IP. Compute nodes not visible to the internet
Signup and Login		
T2	User enters registration details and submits	Registration success and addition of user details in user table
T3	User logs in with registered credentials	Login Successful and redirected to Dashboard
Create a Reservation		
T4	User gives valid image,start_time,duration that do not overlap and profile and hits reservation Button in the Dashboard	container reservation successful, user gets IP address and port number
T5	All compute nodes down , User creates reservation request and hits reserve in the Dashboard	User receives “no resources available” message

Verification and Validation

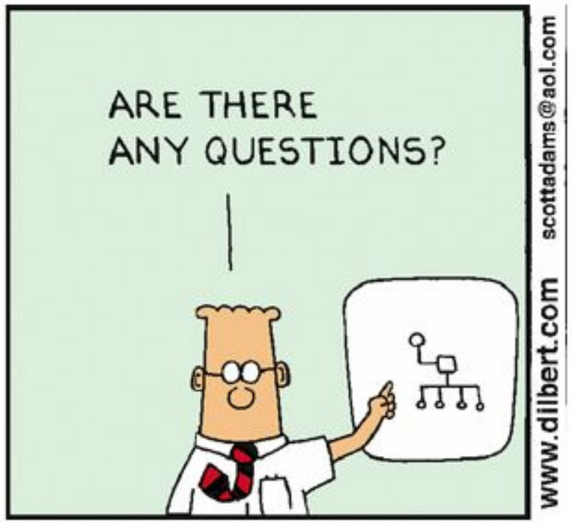
TestCase#	TestCase	Expected Observation
Scheduling and Load Balancing		
T6	Assign custom loads on different compute nodes for a create reservation request	The correct host is allocated for a container request
Container Service		
T7	User to access the container using provided public address and port	Port mapping successful and user is able to access required service
Image Repository		
T8	Local Image Registry setup with images of service provided	Docker module in compute node able to pull the right image and use it to spin a container
Delete Reservation		
T9	Reservation time for the user ends / User cancels his reservation in between	Container stopped, all configurations/data removed and displays "Deletion successful" message in Dashboard
Monitoring		
T10	Admin uses Dashboard to monitor/ do health checks on the containers	Admin is able to see which containers are running, resources used by each compute node etc

Results/Progress

Task	Status
Research about Docker and understand the working and installation	Completed
UI design plan	Completed
Determine the system requirements and the cloud network setup	Completed
Install Ubuntu 16.04 and Docker on lab machines	Completed
Set up network functionalities on lab machines	Completed
Setting up database and docker registry on digital ocean	Completed
User container reservations	Completed
User login and signup	Completed

Results/Progress

Task	Status
Decide the services offered to the customer	Completed
Show and delete reservations	Completed
Container health check (monitoring)	Completed
Network and scheduling components	Completed
User Billing	Completed
Research on storage requirements	In Progress
Verification and validation	In Progress
Performance comparison with AWS	In Progress
Migration to lab machine	In Progress
Documentation	In Progress


ARE THERE ANY QUESTIONS?
scottadams@aol.com
www.dilbert.com