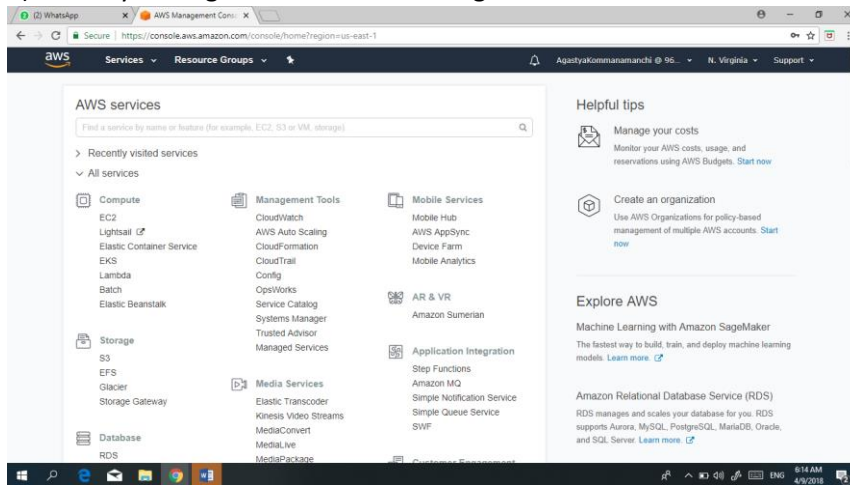**Solution Dockers and Microservices Lab1**
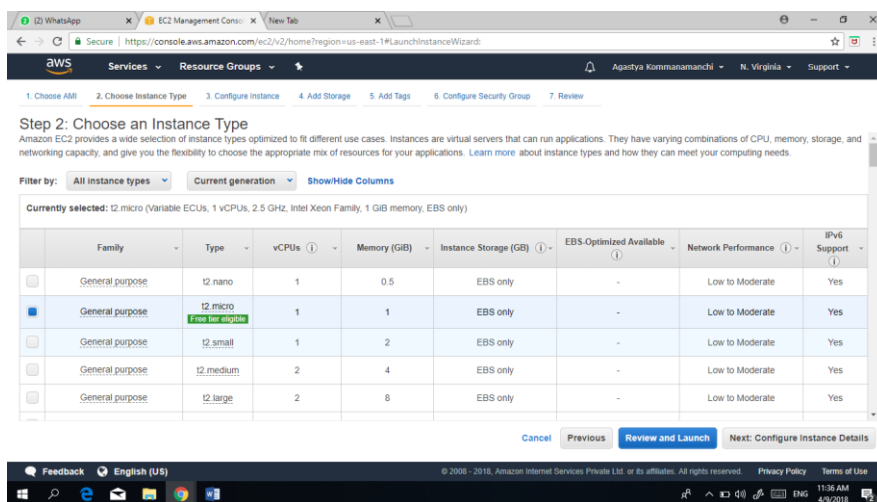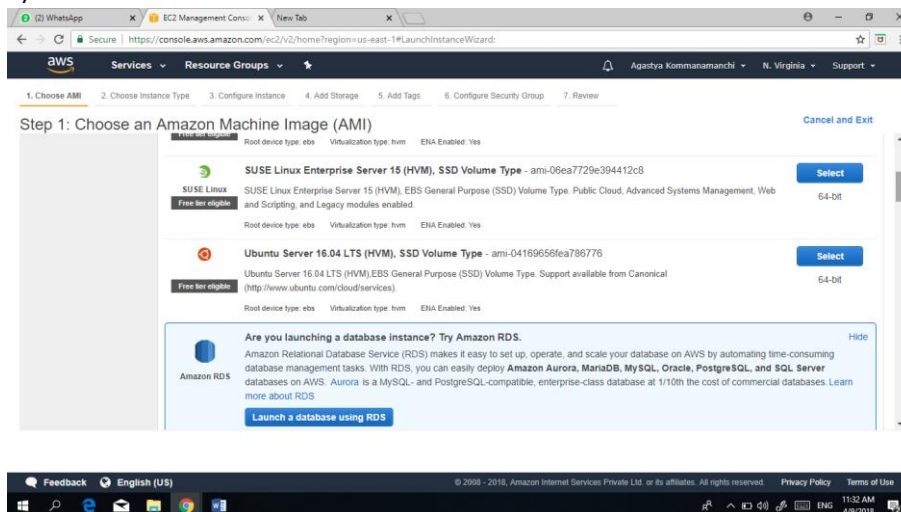
PartA

1)Ensure your region is set to North Virginia

2)Create an EC2 instance using seven step workflow

a)"T2 Small" instance with ubuntu16.04LTS

Solution Dockers and Microservices Lab1

c)Download a New pem file

Solution Dockers and Microservices Lab1

C)SSH into the instance
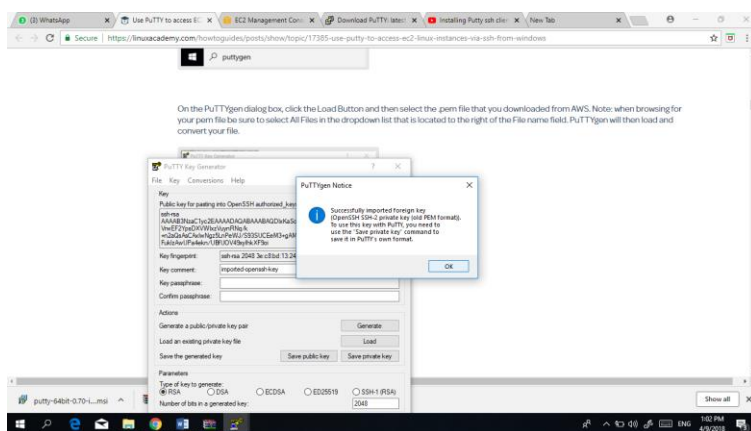
4) Create a new SG port to open port 80 and assign to the EC2 instance
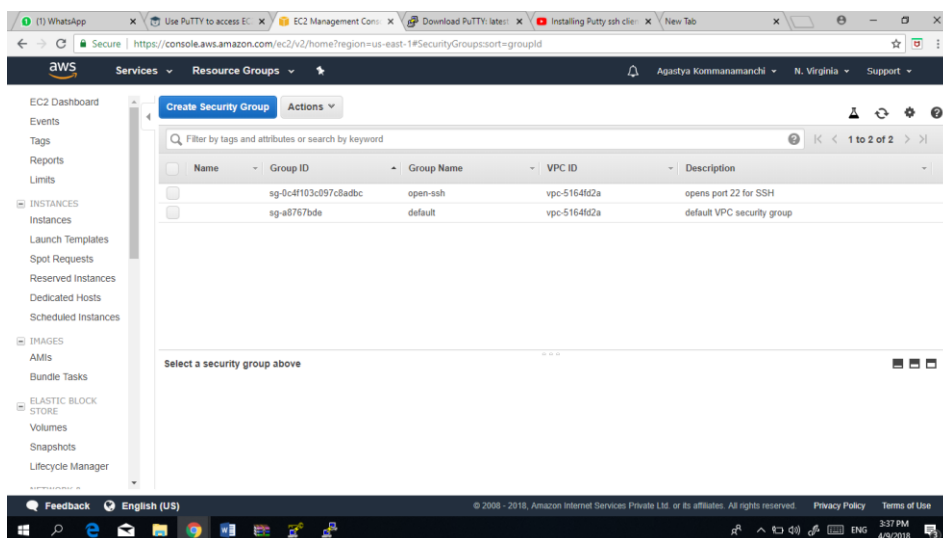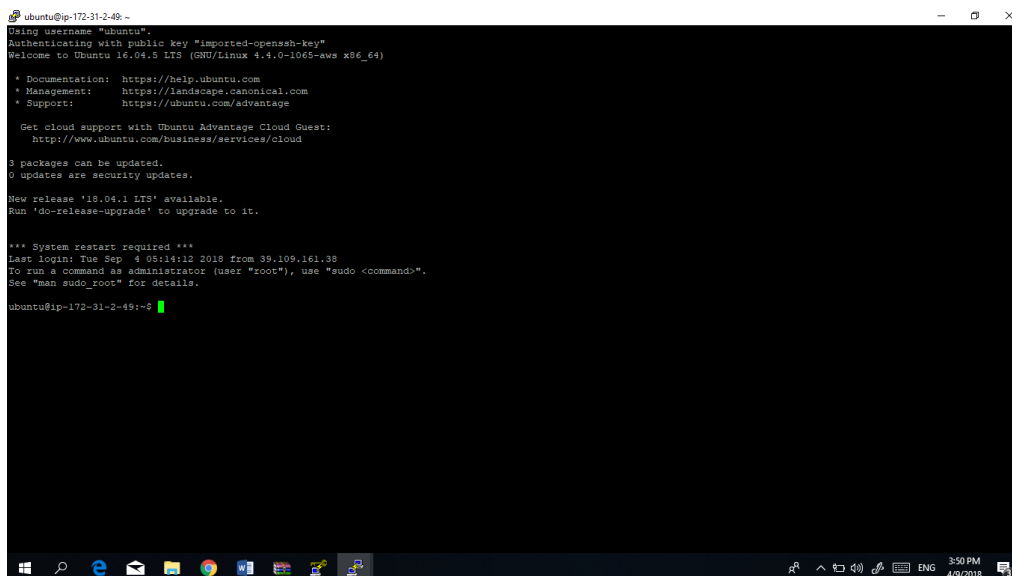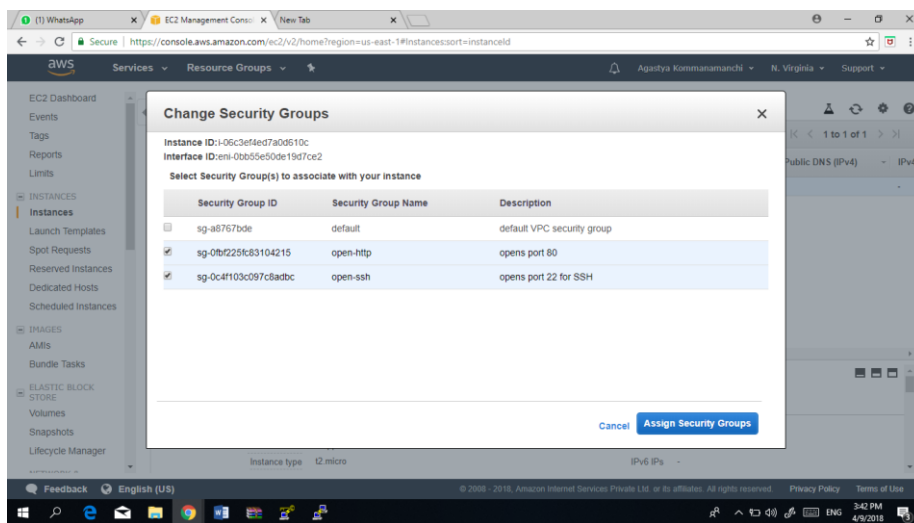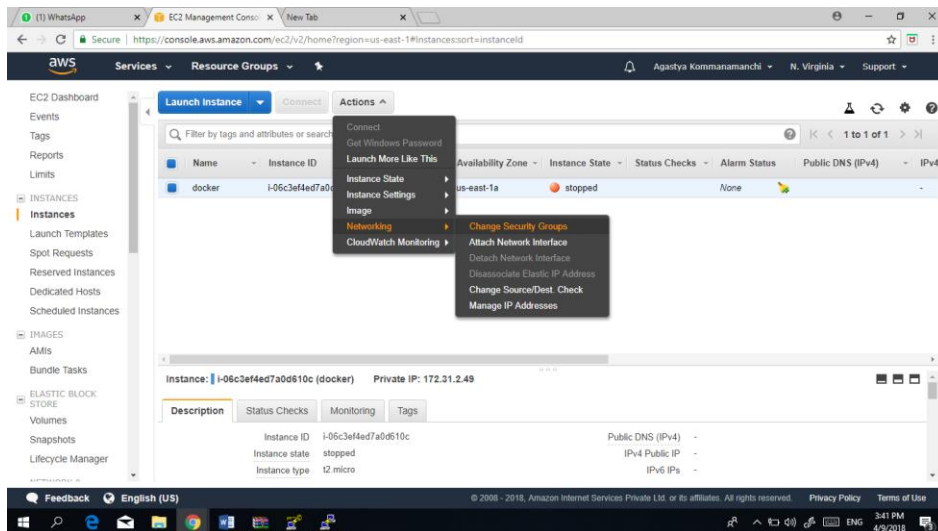
Solution Dockers and Microservices Lab1

Solution Dockers and Microservices Lab1

4) Install Docker from scratch Commands used from reference 1

- sudo apt update
- sudo apt install docker.io

sudo docker version

sudo usermod -a -G docker ubuntu

sudo chown ubuntu:ubuntu -R /opt

Solution Dockers and Microservices Lab1

#After SSH type the below command (notice there is no sudo)

docker version

Part B

1) Run a tomcat instance with the following settings **(reference 2)**
   - docker images
   - docker ps -a
   - docker run --rm busybox /bin/echo "Hello World"

After downloading busybox image

Solution Dockers and Microservices Lab1

- docker images
- docker ps -a

```
ubuntu@ip-172-31-2-49: ~                                                    —  □  ×
ubuntu@ip-172-31-2-49:~$ docker run --rm busybox /bin/echo "Hello world"
Hello world
ubuntu@ip-172-31-2-49:~$ docker images
REPOSITORY          TAG            IMAGE ID          CREATED          SIZE
busybox             latest         e1ddd7948a1c      4 weeks ago      1.16 MB
ubuntu@ip-172-31-2-49:~$ docker ps -a
CONTAINER ID       IMAGE          COMMAND           CREATED          STATUS          PORTS          NAMES
ubuntu@ip-172-31-2-49:~$ docker ps -a
CONTAINER ID       IMAGE          COMMAND           CREATED          STATUS          PORTS          NAMES
ubuntu@ip-172-31-2-49:~$ docker images
REPOSITORY          TAG            IMAGE ID          CREATED          SIZE
busybox             latest         e1ddd7948a1c      4 weeks ago      1.16 MB
ubuntu@ip-172-31-2-49:~$ docker -d -p 80:8080 tomcat:jre8
unknown shorthand flag: 'd' in -d
See 'docker --help'.

Usage:  docker COMMAND

A self-sufficient runtime for containers

Options:
      --config string      Location of client config files (default "/home/ubuntu/.docker")
  -D, --debug              Enable debug mode
      --help               Print usage
  -H, --host list          Daemon socket(s) to connect to (default [])
  -l, --log-level string   Set the logging level ("debug", "info", "warn", "error", "fatal") (default "info")
      --tls                Use TLS; implied by --tlsverify
      --tlscacert string   Trust certs signed only by this CA (default "/home/ubuntu/.docker/ca.pem")
      --tlscert string     Path to TLS certificate file (default "/home/ubuntu/.docker/cert.pem")
      --tlskey string      Path to TLS key file (default "/home/ubuntu/.docker/key.pem")
      --tlsverify          Use TLS and verify the remote
  -v, --version            Print version information and quit

Management Commands:
  checkpoint  Manage checkpoints
  container   Manage containers
  image       Manage images
  network     Manage networks
  node        Manage Swarm nodes
  plugin      Manage plugins
  secret      Manage Docker secrets
  service     Manage services
  stack       Manage Docker stacks
  swarm       Manage Swarm
```

b) Should be JRE8

```
ubuntu@ip-172-31-2-49: ~                                                    —  □  ×
  restart   Restart one or more containers
  rm        Remove one or more containers
  rmi       Remove one or more images
  run       Run a command in a new container
  save      Save one or more images to a tar archive (streamed to STDOUT by default)
  search    Search the Docker Hub for images
  start     Start one or more stopped containers
  stats     Display a live stream of container(s) resource usage statistics
  stop      Stop one or more running containers
  tag       Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
  top       Display the running processes of a container
  unpause   Unpause all processes within one or more containers
  update    Update configuration of one or more containers
  version   Show the Docker version information
  wait      Block until one or more containers stop, then print their exit codes

Run 'docker COMMAND --help' for more information on a command.

ubuntu@ip-172-31-2-49:~$ docker run -d -p 80:8080 tomcat:jre8
Unable to find image 'tomcat:jre8' locally
jre8: Pulling from library/tomcat
55cbf04beb70: Pull complete
1607093a898c: Pull complete
9a8ea045c926: Pull complete
1290813abd9d: Pull complete
8a6b982ad6d7: Pull complete
abb029e68402: Pull complete
d068d0a738e5: Pull complete
42ee47bb0c52: Pull complete
ae9c86laed25: Pull complete
60bba9d0dc8d: Pull complete
091c11d70257: Pull complete
869034383fc7: Pull complete
Digest: sha256:eab533908d6c7760a84a5edbb4490b648484941dfa708a1c5f4286db8caab825
Status: Downloaded newer image for tomcat:jre8
7035b3e774612e2f767e69273ed49344184fdc10a997cc5929b0f3bbeafb8905
ubuntu@ip-172-31-2-49:~$ docker ps -a
CONTAINER ID       IMAGE          COMMAND           CREATED          STATUS          PORTS                    NAMES
7035b3e77461       tomcat:jre8    "catalina.sh run"  5 minutes ago    Up 5 minutes    0.0.0.0:80->8080/tcp    infallible_chandrasekhar
ubuntu@ip-172-31-2-49:~$ docker images
REPOSITORY          TAG            IMAGE ID          CREATED          SIZE
tomcat              jre8           690cb3b9c7d1      12 days ago      463 MB
busybox             latest         e1ddd7948a1c      4 weeks ago      1.16 MB
ubuntu@ip-172-31-2-49:~$
```
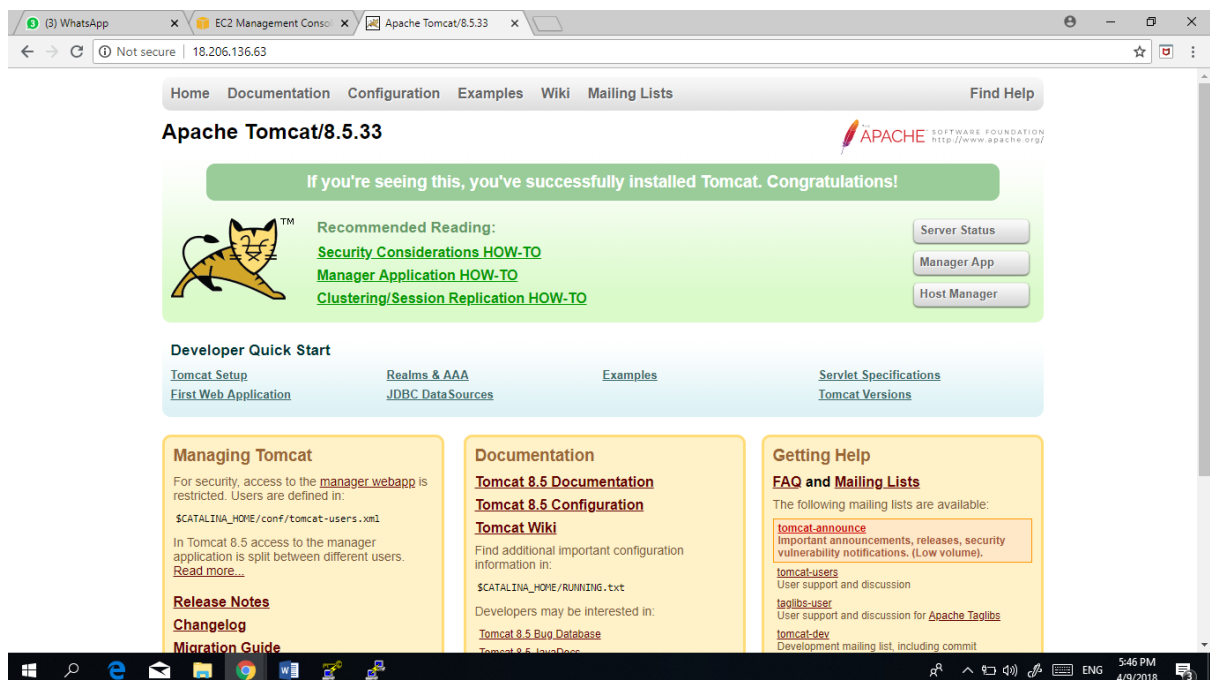
c) Default tomcat port 8080 should be mapped to port 80 of host

Solution Dockers and Microservices Lab1

d) Be able to access the default tomcat page from the browser

Removing the docker images and containers created

Below are the commands used

- docker rm [container id]
- docker rmi [image id]
- docker build -t [your image name without spaces] .

Part C

1. Stop the previous container running only Tomcat

2.

2. To create a custom image of a java web app using Tomcat8 (reference 3)

a) Download the HelloWorld.war in the EC2 instance

b) Create a file by the name "Dockerfile", this file does not have any extension

c) Create the custom image using the appropriate command

Solution Dockers and Microservices Lab1

3. Launch a container from your custom image and access the application from the browser using the URL http:// 34.236.237.141/ HelloWorld

Welcome!

If you are reading this message then the installation has gone well and the application is running. Congratulations!!

Login

Type in your first name

Password

The password is hard coded as admin123

Go ahead, try it!

Application version - v1
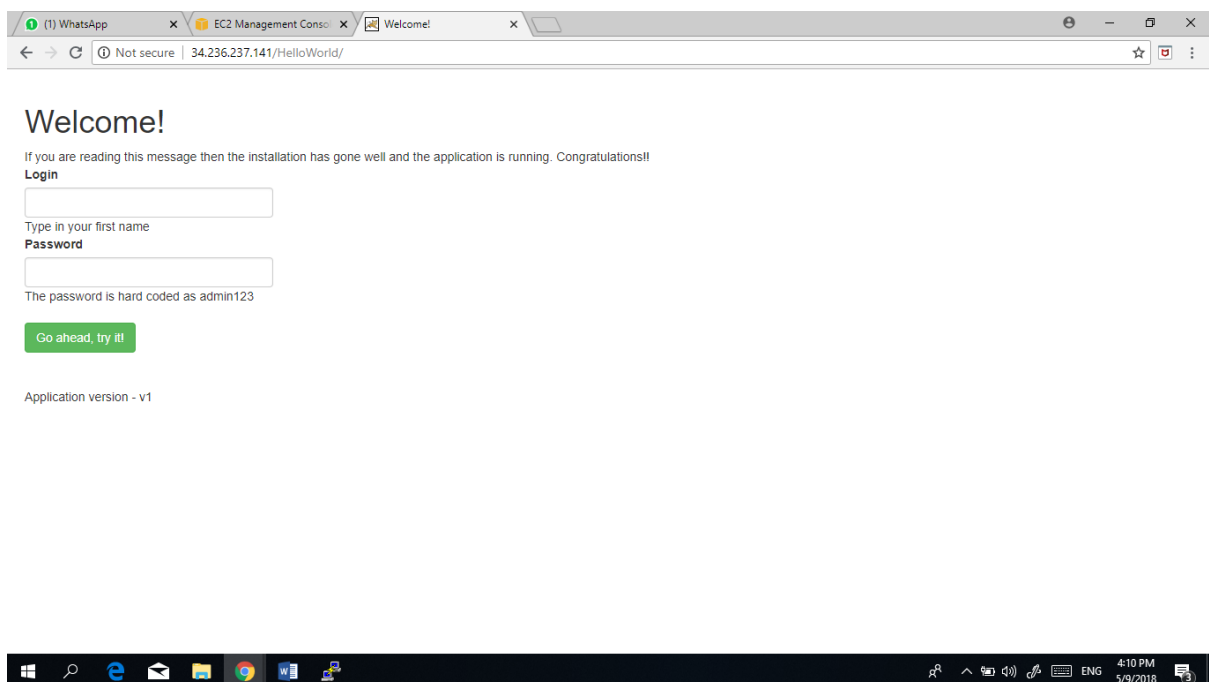
Solution Dockers and Microservices Lab1
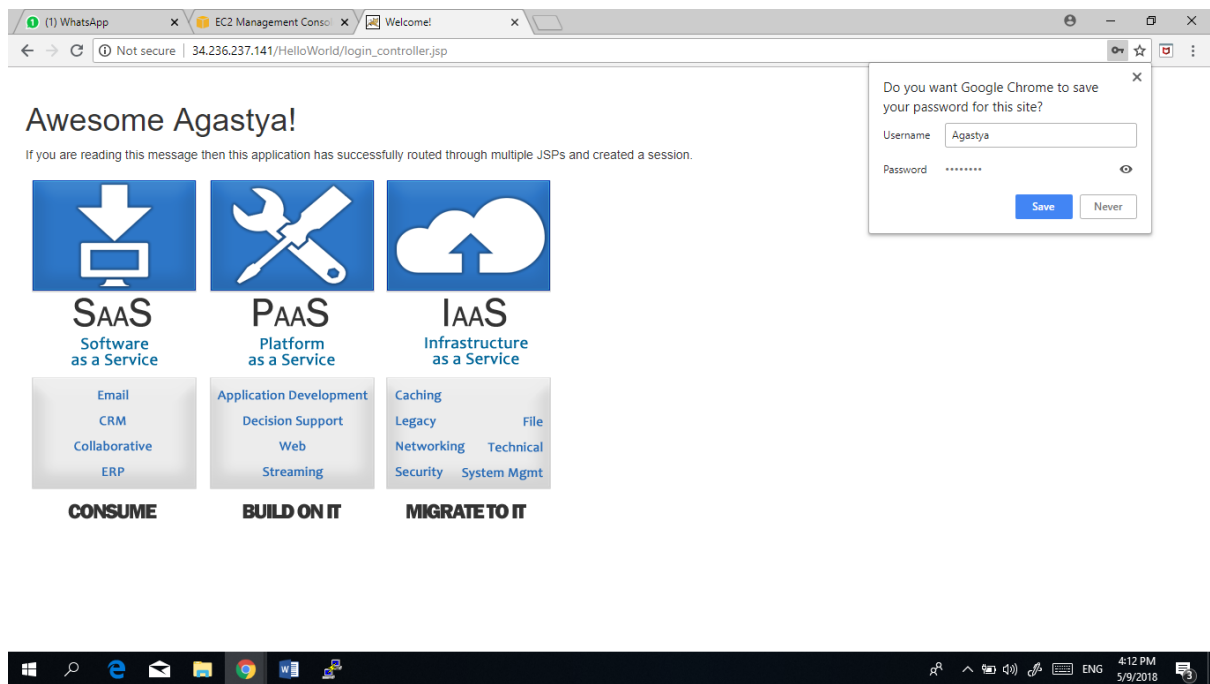
Part D

. Stop the previous container that is running the Java web application in Tomcat

3. Create a custom image with a static website running in httpd (reference 4)

Solution Dockers and Microservices Lab1

4.

Solution Dockers and Microservices Lab1

Part E(Advanced)

1)PULL Debian

1.

a)update and upgrade

Solution Dockers and Microservices Lab1

Adding user

Debian base=user setup +Debian

Docker file

Solution Dockers and Microservices Lab1

Adding port HTTPS

Solution Dockers and Microservices Lab1

Final Result

Lab Maintenance