**Table of Contents**

1. Objective
2. Preparation
3. Installation Procedure
   1. Clone Git
   2. Ansible Ping Test
   3. Running Ansible
   4. MySQL Remote Connection
   5. Icinga Database
   6. Icinga Plugins
   7. Starting Icinga Daemon

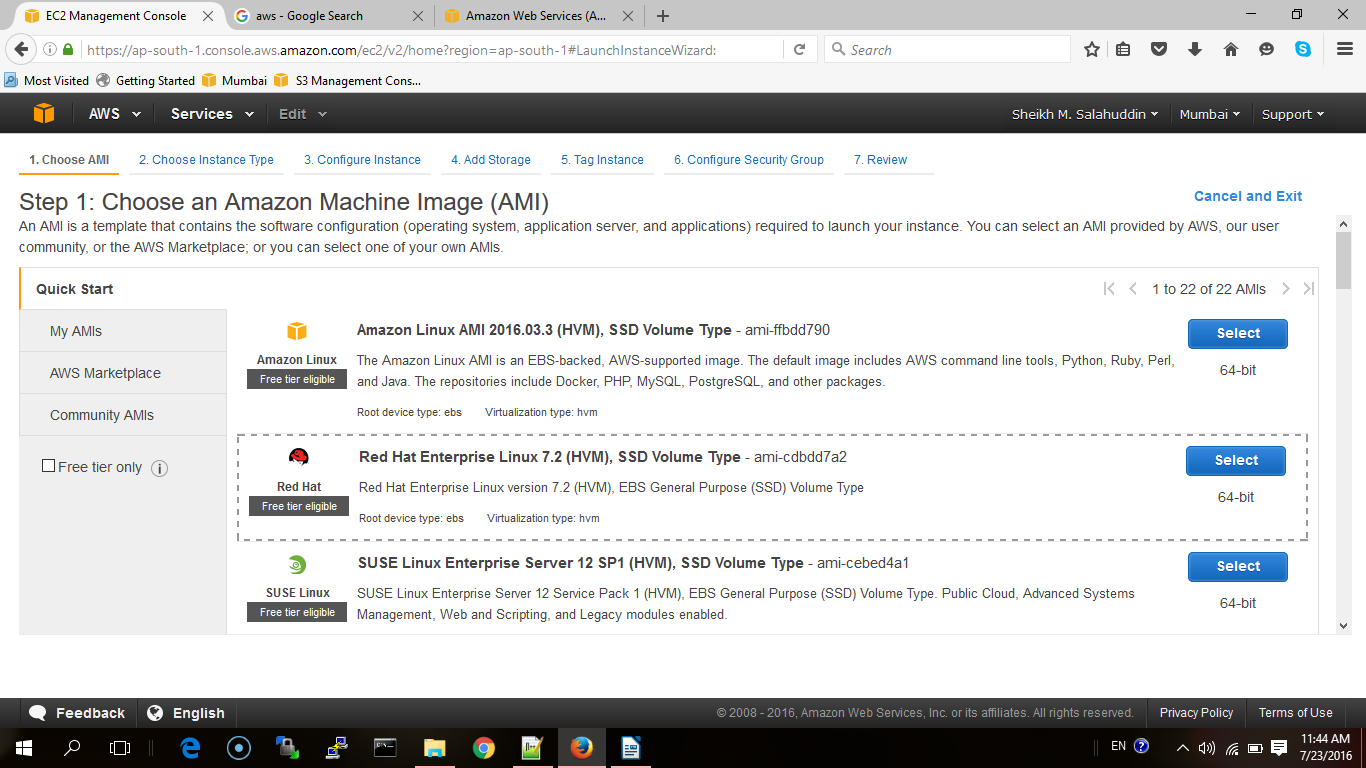
# **Overall Objective**

You are required to setup a monitoring solution using Icinga. Icinga should monitor the Apache Web Server and the Mysql Database server and send logs to Amazon S3 dynamically using Bash Scripting.

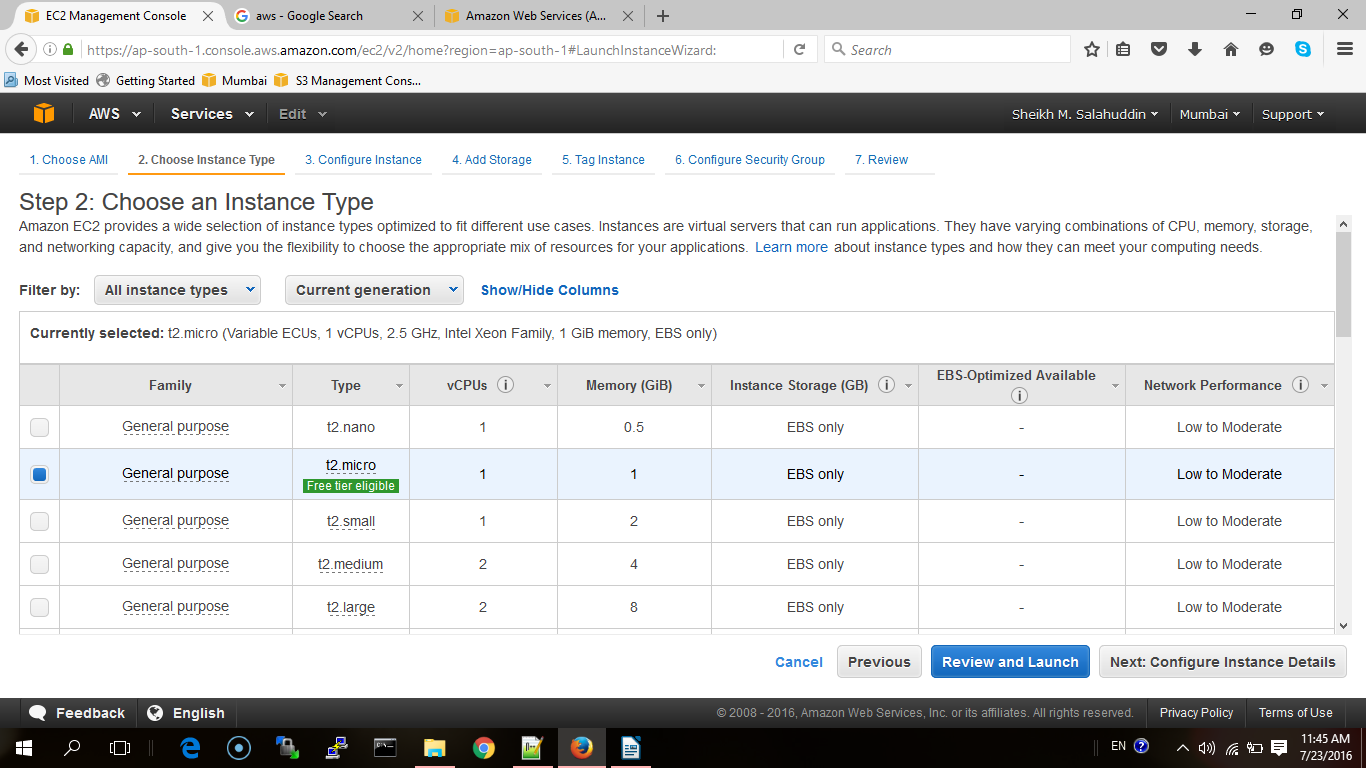
**Preparation**

1) Get an Account on Amazon Web Services (AWS)

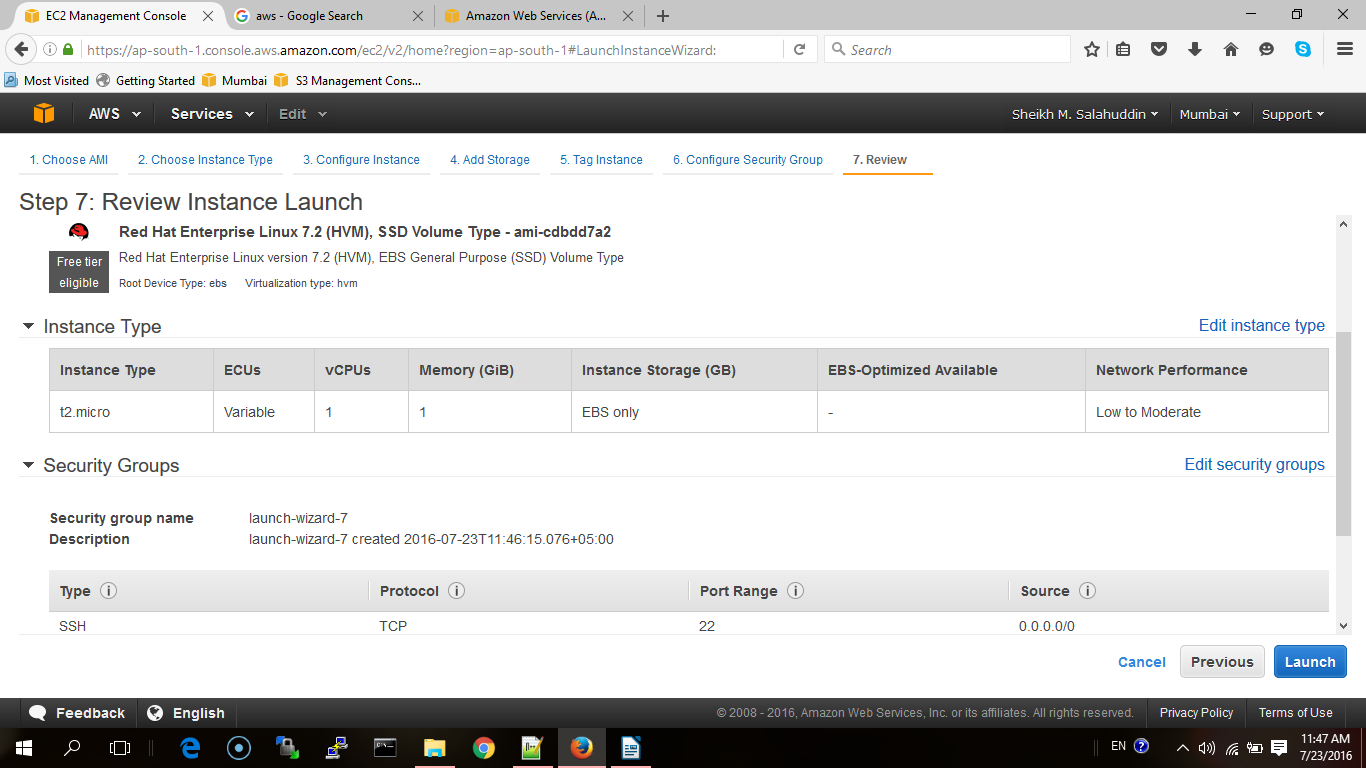
2) Chose AMI (Amazone Machine Image) of Redhat (Available for free tier)

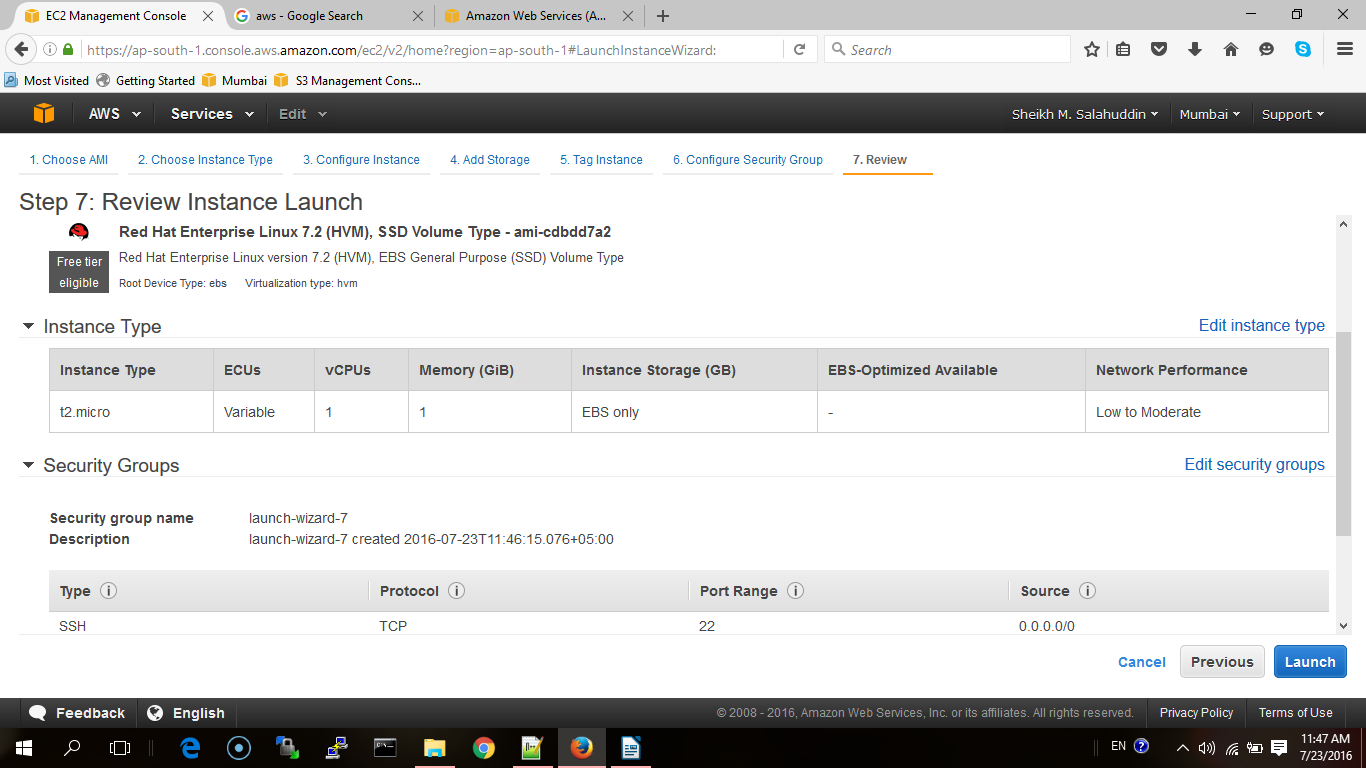


3) Review and Launch



4) Launch the machine, create a new key pair and download it.

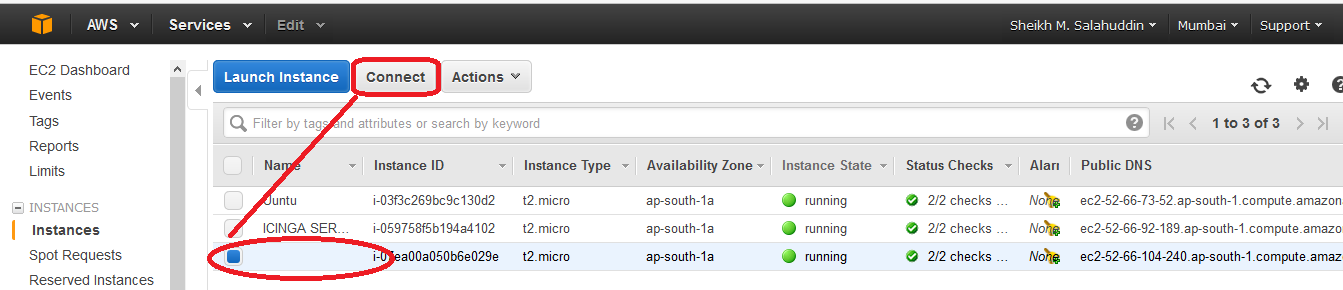


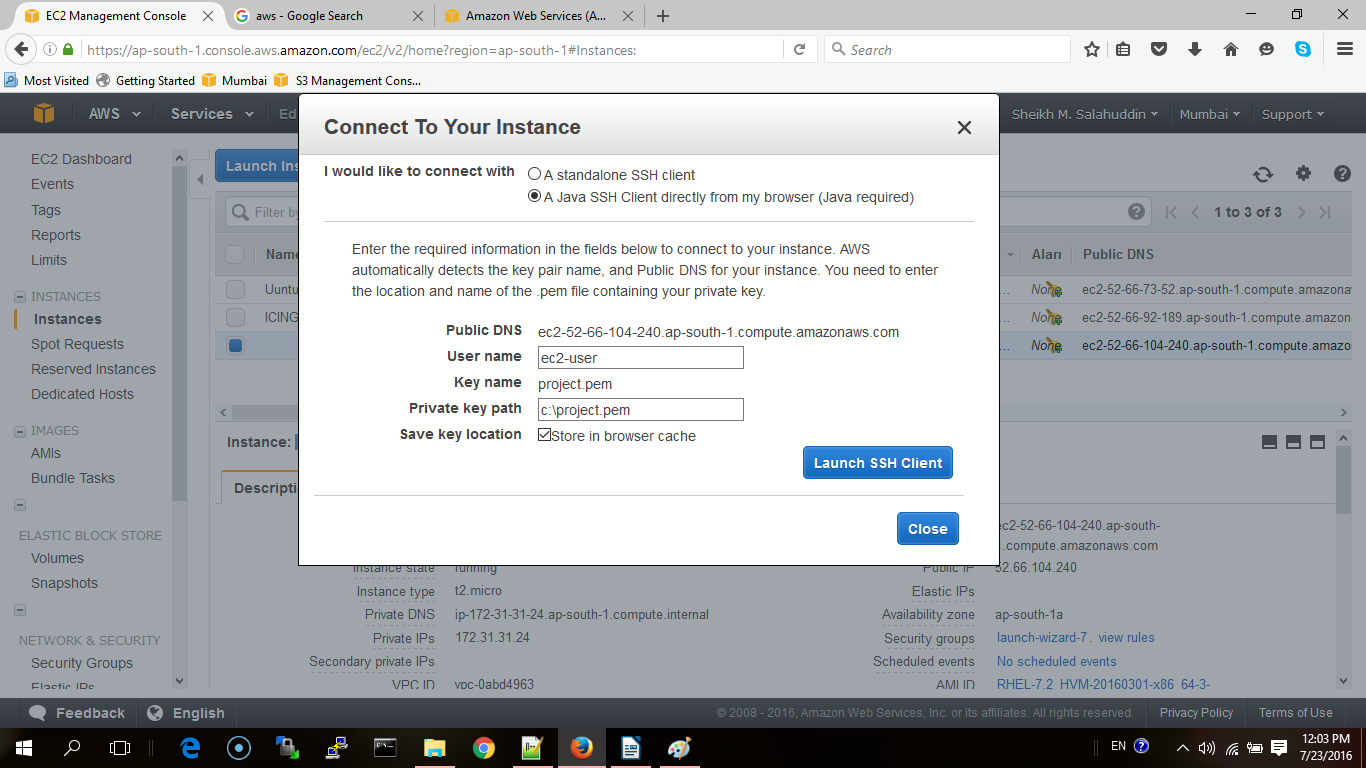


5) Launch the instance. It will take a few minutes before it's available for use.

6) Go to Management Console, select the instance and connect

(Note: if you want to use browser's java client than use firefox because chrome does not offer browser client.)





7) Install important packages. These packages will be used during the installation for different purposes.

$ sudo yum install wget nano git gcc mysql -y

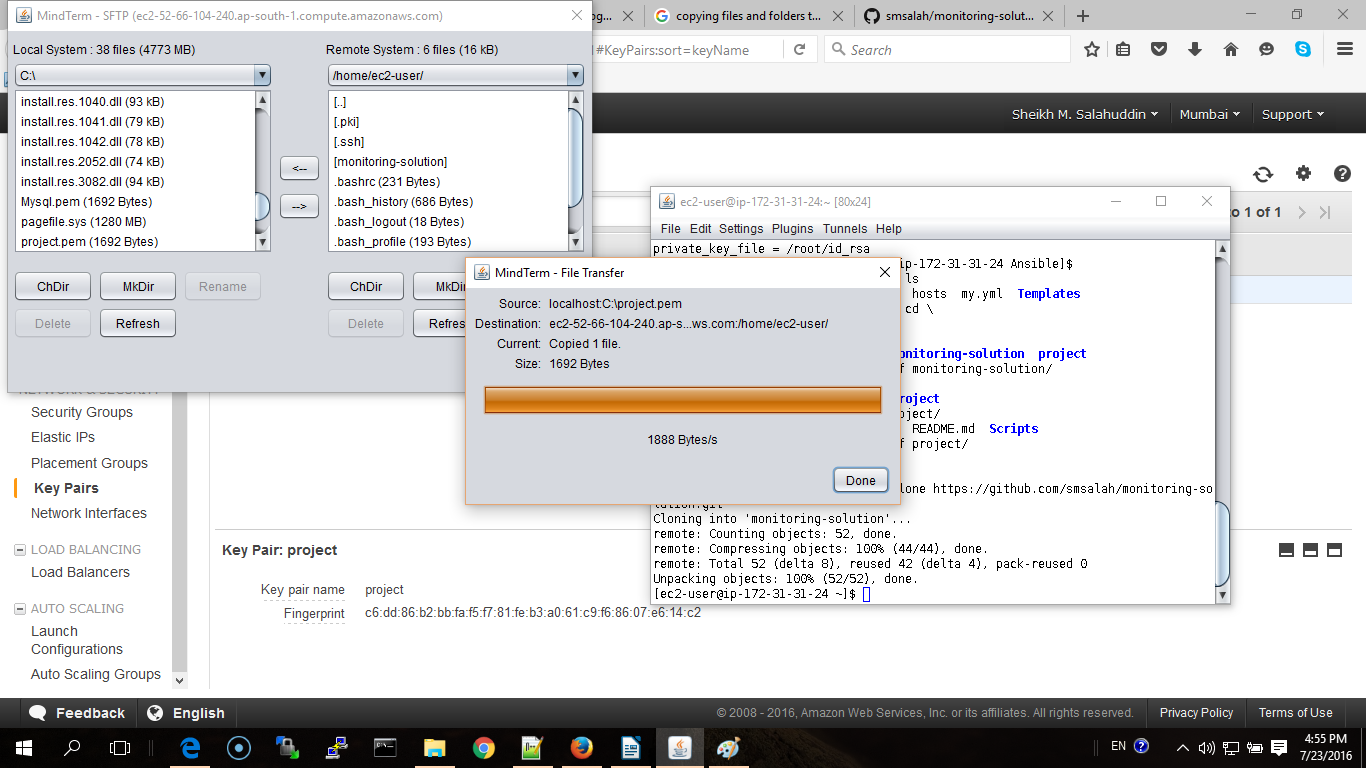
8) Our instance of redhat in amazon already has a user; ec2-user and is added in /etc/sudoer. In any case please make sure that user of your instance is the member of /etc/sudoer  
  
9) Install EPEL repository (<https://support.rackspace.com/how-to/install-epel-and-additional-repositories-on-centos-and-red-hat/>). Here are the commands  
  
$ wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm  
$ sudo rpm -Uvh epel-release-latest-7\*.rpm

10) Install Ansible (<http://docs.ansible.com/ansible/intro_installation.html> ).   
$ sudo yum install ansible  
  
11) Configure S3 Amazon proper configured (with a valid Access Key);  
  
12) Install docker repository  
$ sudo tee /etc/yum.repos.d/docker.repo <<-EOF  
[dockerrepo]  
name=Docker Repository  
baseurl=<https://yum.dockerproject.org/repo/main/centos/7>   
enabled=1  
gpgcheck=1  
gpgkey=<https://yum.dockerproject.org/gpg>   
EOF  
  
13) Install icinga repository  
$ sudo yum install <https://packages.icinga.org/epel/7/release/noarch/icinga-rpm-release-7-1.el7.centos.noarch.rpm>

**Installation**

**1) Import private key**

Import the key pair (project.pem) to the user home. Remember that we downloaded the key when we are launching the instance.



This key willl be used in ansible.cfg and hosts files. The key enables the user to run ansible using ssh on localhost.

**2) key must not be publicly viewable for SSH to work.**

$ chmod 400 project.pem

**3) Test ssh to localhost**

$ ssh -i project.pem localhost  
(If you successfully login then

$exit

**4) Clone my git repository to access the necessary files**

$ git clone https://github.com/smsalah/monitoring-solution.git

**5) change the folder to Ansible**

a) $ pwd

b) $ /home/ec2-user

c) cd monitoring-solution/Ansible

**Important Note:**  
- Make sure th check ansible.cfg hosts files  
- Change remote user name and private key file as per your requirement in ansible.cfg  
- Change ansible\_ssh\_private\_key\_file value as per your requirement.  
  
  
**6) Before starting installation procedure, test ansible connection to localhost**  
  
$ ansible all -m ping  
  
If eerything is ok you will get the result as follows  
  
$ localhost | SUCCESS => {  
 "changed": false,  
 "ping": "pong"  
}

**7) Run ansible-playbook using configuration file my.yml**

$ ansible-playbook -v my.yml

my.yml performs some important tasks.

a) Configuration of the local environment

b) Creation of a user and configure ssh. (Not in our case as we already have a user created by AMI instance

c) Installation of important packages such as docker and icinga etc

e)Installation of two docker containers running Apache Server and MySQL server separately.

f) Preparation of ICINGA daemon.

g) Install cron entries.

**8) Connect MySQL From the Host**

a) First connect to mysql container

$ sudo docker exec -i -t mydb /bin/bash  
root@4e6299889d89:/# mysql -p

Enter password: (enter your password. In our case It is 'root')

b) Update root privileges   
mysql> GRANT ALL ON \*.\* to root@'%' IDENTIFIED BY 'root';  
mysql> FLUSH PRIVILEGES;  
mysql> quit  
root@5afda1dcfec3:/# exit  
exit

c) Get the container id

$ sudo docker ps

d) user the container id to get the host ip address

$ docker inspect <container id>

e) Finally connect the mysql container from the host.  
$ mysql -P 3306 -u root -p -h 172.17.0.2  
Enter password: (root in our case)  
  
**9) Creating Icinga Database and its tables**  
  
a) Connect to the container  
$ docker exec -i -t mydb /bin/bash  
$ cd /etc/mysql  
$ mysql -proot -uroot < icinga.sql  
$ mysql -Dicinga -proot -uroot < icinga\_schema.sql  
$ exit  
  
**10) Install Icinga Plugins**  
  
Our ansible playbook downloads the tar package of plugins inside /tmp. To install it run   
the following commands:  
$ cd /tmp/  
$ tar xzvf monitoring-plugins-2.1.2.tar.gz ; cd monitoring-plugins-2.1.2/  
$ ./configure --prefix=/tmp/ && make && make install   
$ cd /tmp/libexec  
$ find . -print | sudo cpio -dumpv /usr/lib64/nagios/plugins

**11) Starting Icinga Daemon**  
  
Ansible playbook creates two configuration files and put it inside `/etc/icinga/objects`. These  
files are configured to provide basic monitoring for the apache and mysql (use `check\_http` and `check\_mysql` plugins).  
  
$ sudo icinga -d /etc/icinga/icinga.cfg  
  
To test the Daemon run the following command:  
  
$ sudo icinga --show-scheduling /etc/icinga/icinga.cfg