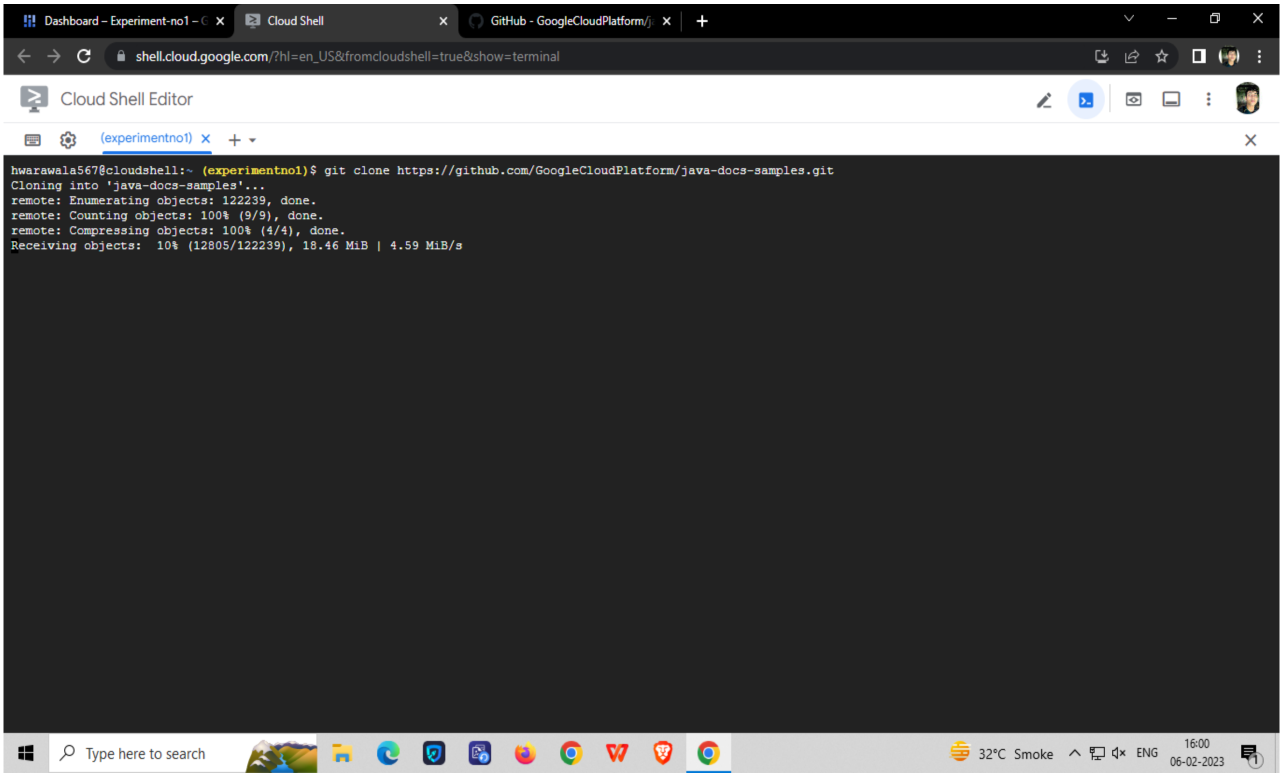
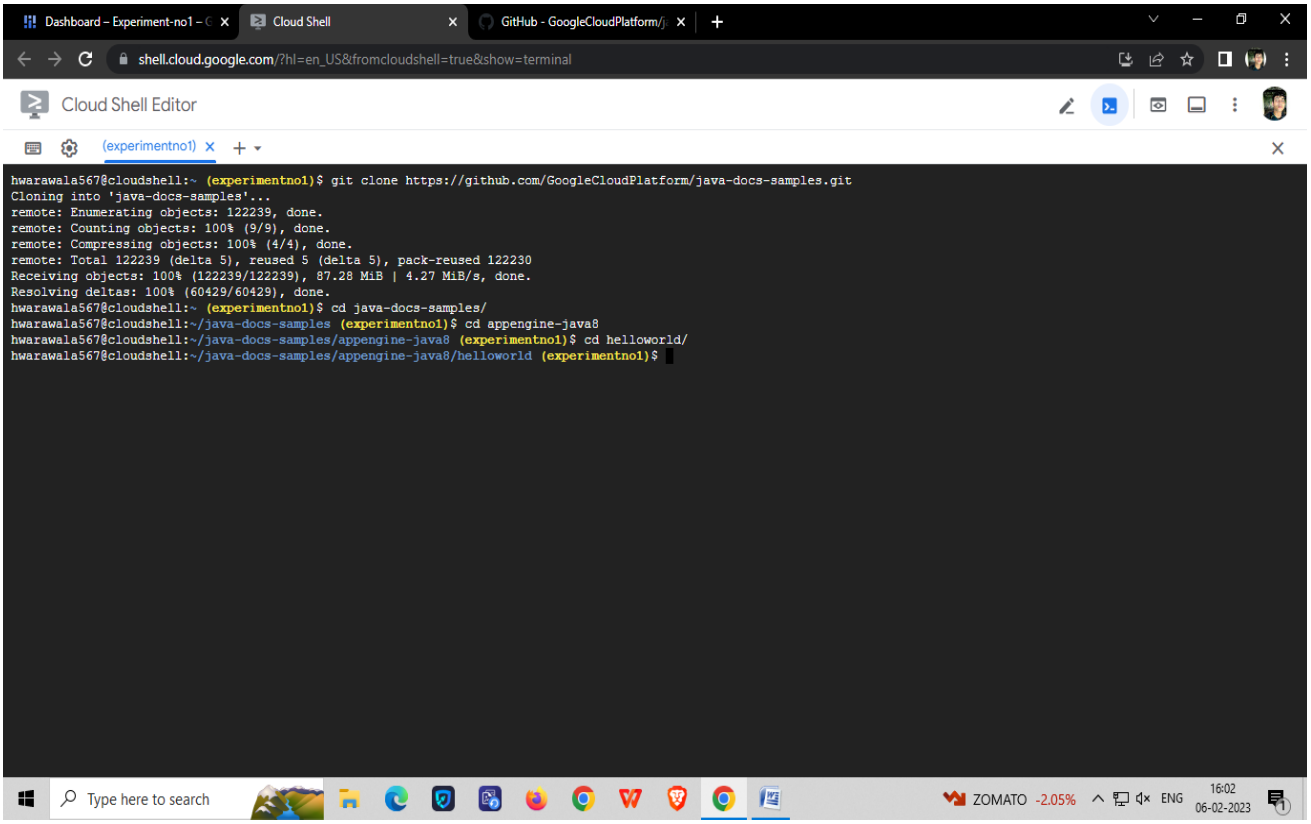
Experiment No. 1

Install Google App Engine. Create hello world app and other simple web applications using python /java.



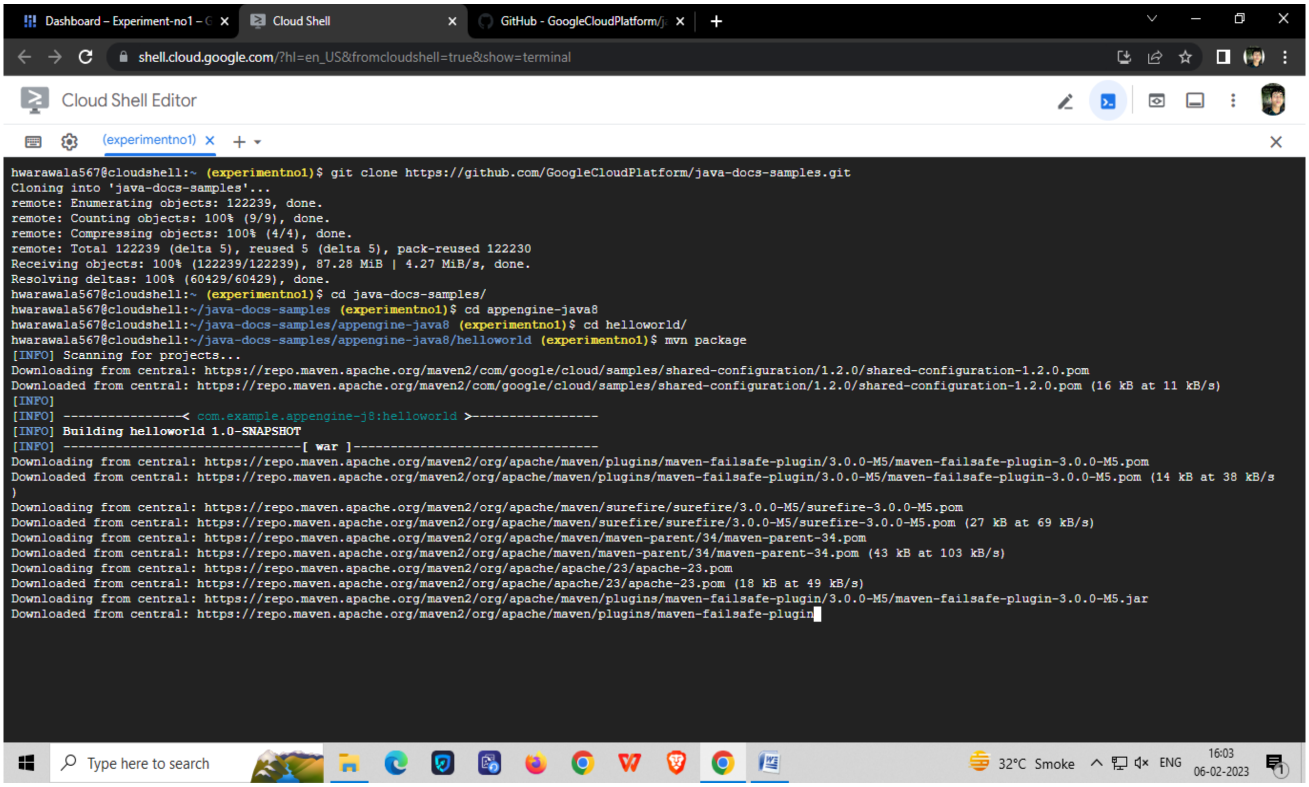
Dashboard – Experiment-no1 – G
Cloud Shell
GitHub - GoogleCloudPlatform/j
shell.cloud.google.com/?hl=en_US&fromcloudshell=true&show=terminal
Cloud Shell Editor
(experimentno1)
Type here to search
32°C Smoke
ENG
16:00
06-02-2023

```
hwarawala567@cloudshell:~ (experimentno1)$ git clone https://github.com/GoogleCloudPlatform/java-docs-samples.git
Cloning into 'java-docs-samples'...
remote: Enumerating objects: 122239, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (4/4), done.
Receiving objects:  10% (12805/122239), 18.46 MiB | 4.59 MiB/s
```

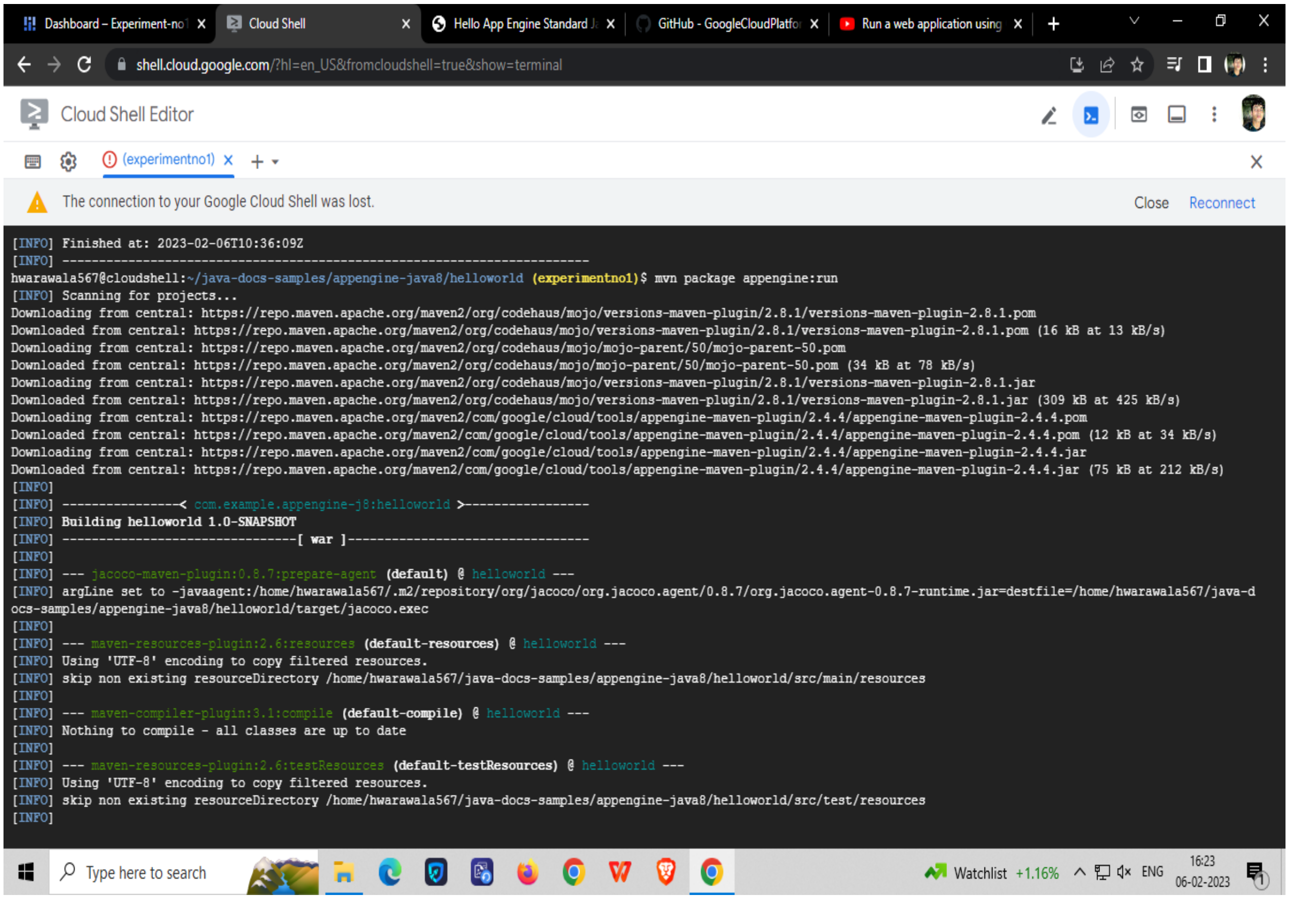


Dashboard – Experiment-no1 – G
Cloud Shell
GitHub - GoogleCloudPlatform/j
shell.cloud.google.com/?hl=en_US&fromcloudshell=true&show=terminal
Cloud Shell Editor
(experimentno1)
Type here to search
ZOMATO -2.05%
ENG
16:02
06-02-2023

```
hwarawala567@cloudshell:~ (experimentno1)$ git clone https://github.com/GoogleCloudPlatform/java-docs-samples.git
Cloning into 'java-docs-samples'...
remote: Enumerating objects: 122239, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 122239 (delta 5), reused 5 (delta 5), pack-reused 122230
Receiving objects: 100% (122239/122239), 87.28 MiB | 4.27 MiB/s, done.
Resolving deltas: 100% (60429/60429), done.
hwarawala567@cloudshell:~ (experimentno1)$ cd java-docs-samples/
hwarawala567@cloudshell:~/java-docs-samples (experimentno1)$ cd appengine-java8
hwarawala567@cloudshell:~/java-docs-samples/appengine-java8 (experimentno1)$ cd helloworld/
hwarawala567@cloudshell:~/java-docs-samples/appengine-java8/helloworld (experimentno1)$
```

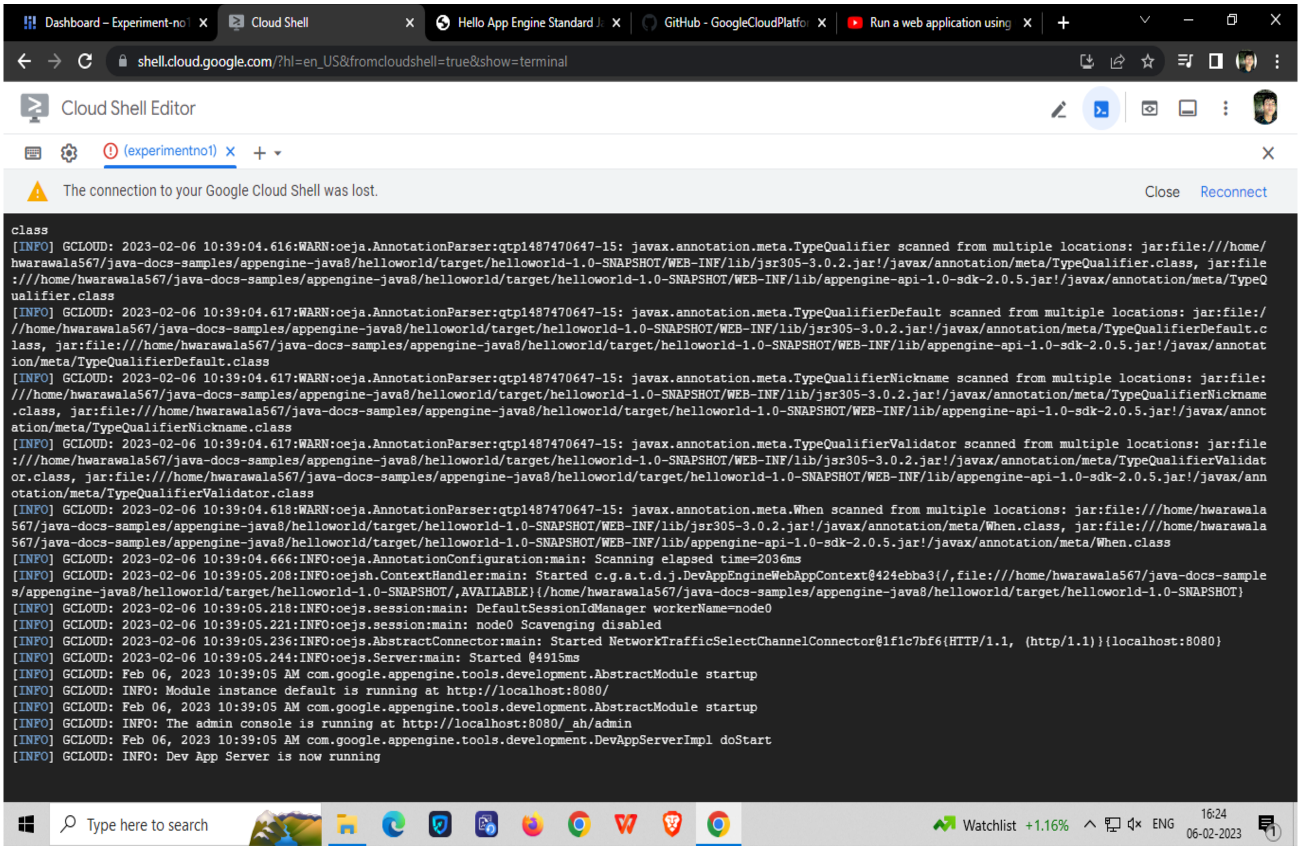

Dashboard – Experiment-no1 – C
Cloud Shell
GitHub - GoogleCloudPlatform/j
shell.cloud.google.com/?hl=en_US&fromcloudshell=true&show=terminal
Cloud Shell Editor
(experimentno1)
Type here to search
32°C Smoke
ENG
16:03
06-02-2023

```
hwarawala567@cloudshell:~ (experimentno1)$ git clone https://github.com/GoogleCloudPlatform/java-docs-samples.git
Cloning into 'java-docs-samples'...
remote: Enumerating objects: 122239, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 122239 (delta 5), reused 5 (delta 5), pack-reused 122230
Receiving objects: 100% (122239/122239), 87.28 MiB | 4.27 MiB/s, done.
Resolving deltas: 100% (60429/60429), done.
hwarawala567@cloudshell:~ (experimentno1)$ cd java-docs-samples/
hwarawala567@cloudshell:~/java-docs-samples (experimentno1)$ cd appengine-java8
hwarawala567@cloudshell:~/java-docs-samples/appengine-java8 (experimentno1)$ cd helloworld/
hwarawala567@cloudshell:~/java-docs-samples/appengine-java8/helloworld (experimentno1)$ mvn package
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/com/google/cloud/samples/shared-configuration/1.2.0/shared-configuration-1.2.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/com/google/cloud/samples/shared-configuration/1.2.0/shared-configuration-1.2.0.pom (16 kB at 11 kB/s)
[INFO]
[INFO] ---------------< com.example.appengine-j8:helloworld >----------------
[INFO] Building helloworld 1.0-SNAPSHOT
[INFO] --------------------------------[ war ]---------------------------------
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-failsafe-plugin/3.0.0-M5/maven-failsafe-plugin-3.0.0-M5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-failsafe-plugin/3.0.0-M5/maven-failsafe-plugin-3.0.0-M5.pom (14 kB at 38 kB/s
)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire/3.0.0-M5/surefire-3.0.0-M5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/surefire/surefire/3.0.0-M5/surefire-3.0.0-M5.pom (27 kB at 69 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/34/maven-parent-34.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/34/maven-parent-34.pom (43 kB at 103 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/apache/23/apache-23.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/23/apache-23.pom (18 kB at 49 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-failsafe-plugin/3.0.0-M5/maven-failsafe-plugin-3.0.0-M5.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-failsafe-plugin
```


Dashboard – Experiment-no1
Cloud Shell
Hello App Engine Standard J
GitHub - GoogleCloudPlatfor
Run a web application using
shell.cloud.google.com/?hl=en_US&fromcloudshell=true&show=terminal
Cloud Shell Editor
(experimentno1)
Close
Reconnect
Type here to search
Watchlist +1.16%
ENG
16:23
06-02-2023

The connection to your Google Cloud Shell was lost.

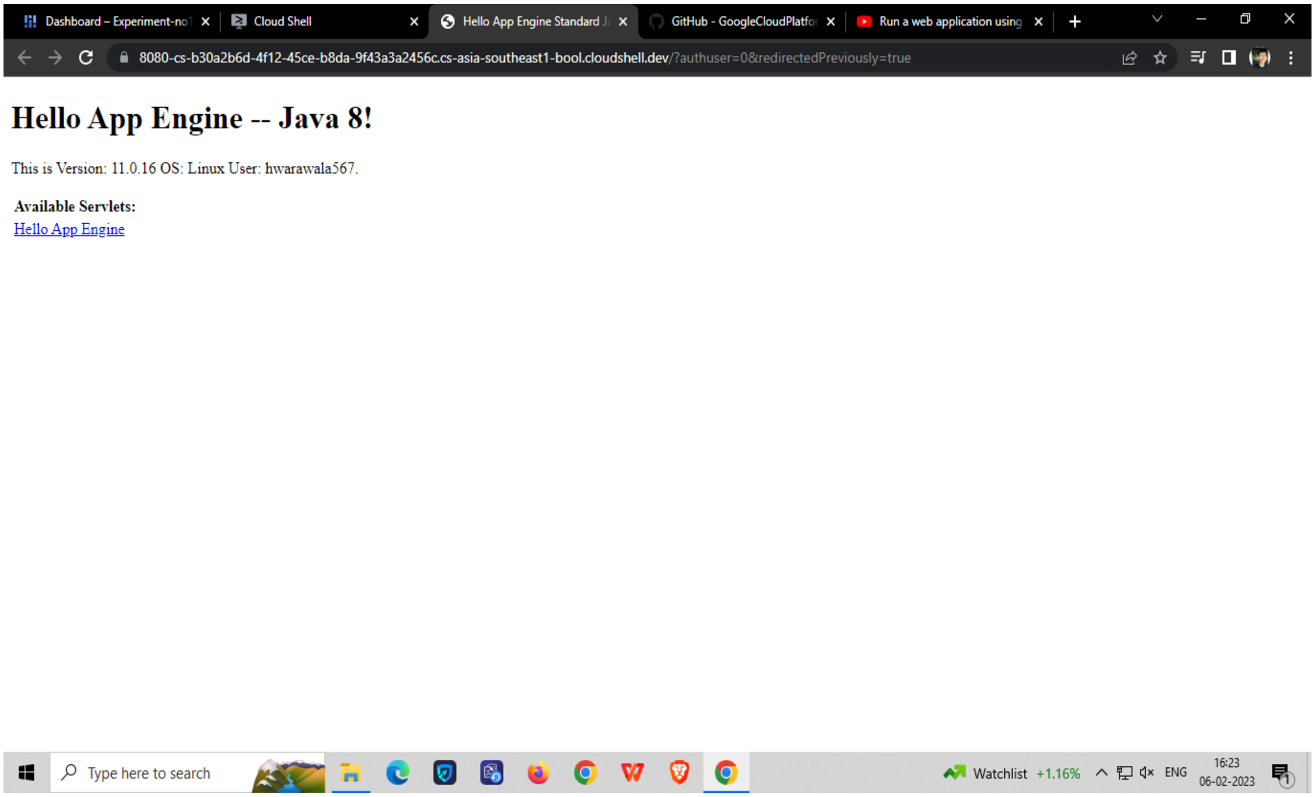
```
[INFO] Finished at: 2023-02-06T10:36:09Z
[INFO] ------------------------------------------------------------------------
hwarawala567@cloudshell:~/java-docs-samples/appengine-java8/helloworld (experimentno1)$ mvn package appengine:run
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/mojo/versions-maven-plugin/2.8.1/versions-maven-plugin-2.8.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/mojo/versions-maven-plugin/2.8.1/versions-maven-plugin-2.8.1.pom (16 kB at 13 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/mojo/mojo-parent/50/mojo-parent-50.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/mojo/mojo-parent/50/mojo-parent-50.pom (34 kB at 78 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/mojo/versions-maven-plugin/2.8.1/versions-maven-plugin-2.8.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/mojo/versions-maven-plugin/2.8.1/versions-maven-plugin-2.8.1.jar (309 kB at 425 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/com/google/cloud/tools/appengine-maven-plugin/2.4.4/appengine-maven-plugin-2.4.4.pom
Downloaded from central: https://repo.maven.apache.org/maven2/com/google/cloud/tools/appengine-maven-plugin/2.4.4/appengine-maven-plugin-2.4.4.pom (12 kB at 34 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/com/google/cloud/tools/appengine-maven-plugin/2.4.4/appengine-maven-plugin-2.4.4.jar
Downloaded from central: https://repo.maven.apache.org/maven2/com/google/cloud/tools/appengine-maven-plugin/2.4.4/appengine-maven-plugin-2.4.4.jar (75 kB at 212 kB/s)
[INFO]
[INFO] ---------------< com.example.appengine-j8:helloworld >----------------
[INFO] Building helloworld 1.0-SNAPSHOT
[INFO] --------------------------------[ war ]---------------------------------
[INFO]
[INFO] --- jacoco-maven-plugin:0.8.7:prepare-agent (default) @ helloworld ---
[INFO] argLine set to -javaagent:/home/hwarawala567/.m2/repository/org/jacoco/org.jacoco.agent/0.8.7/org.jacoco.agent-0.8.7-runtime.jar=destfile=/home/hwarawala567/java-d
ocs-samples/appengine-java8/helloworld/target/jacoco.exec
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ helloworld ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/hwarawala567/java-docs-samples/appengine-java8/helloworld/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ helloworld ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ helloworld ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/hwarawala567/java-docs-samples/appengine-java8/helloworld/src/test/resources
[INFO]
```



Dashboard – Experiment-no1
Cloud Shell
Hello App Engine Standard J
GitHub - GoogleCloudPlatfo
Run a web application using
shell.cloud.google.com/?hl=en_US&fromcloudshell=true&show=terminal
Cloud Shell Editor
(experimentno1)

The connection to your Google Cloud Shell was lost.

```
class
[INFO] GCLOUD: 2023-02-06 10:39:04.616:WARN:oeja.AnnotationParser:qtp1487470647-15: javax.annotation.meta.TypeQualifier scanned from multiple locations: jar:file:///home/
hwarawala567/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/jsr305-3.0.2.jar!/javax/annotation/meta/TypeQualifier.class, jar:file
:///home/hwarawala567/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/appengine-api-1.0-sdk-2.0.5.jar!/javax/annotation/meta/TypeQ
ualifier.class
[INFO] GCLOUD: 2023-02-06 10:39:04.617:WARN:oeja.AnnotationParser:qtp1487470647-15: javax.annotation.meta.TypeQualifierDefault scanned from multiple locations: jar:file:/
//home/hwarawala567/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/jsr305-3.0.2.jar!/javax/annotation/meta/TypeQualifierDefault.c
lass, jar:file:///home/hwarawala567/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/appengine-api-1.0-sdk-2.0.5.jar!/javax/annotat
ion/meta/TypeQualifierDefault.class
[INFO] GCLOUD: 2023-02-06 10:39:04.617:WARN:oeja.AnnotationParser:qtp1487470647-15: javax.annotation.meta.TypeQualifierNickname scanned from multiple locations: jar:file:
///home/hwarawala567/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/jsr305-3.0.2.jar!/javax/annotation/meta/TypeQualifierNickname
.class, jar:file:///home/hwarawala567/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/appengine-api-1.0-sdk-2.0.5.jar!/javax/annot
ation/meta/TypeQualifierNickname.class
[INFO] GCLOUD: 2023-02-06 10:39:04.617:WARN:oeja.AnnotationParser:qtp1487470647-15: javax.annotation.meta.TypeQualifierValidator scanned from multiple locations: jar:file
:///home/hwarawala567/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/jsr305-3.0.2.jar!/javax/annotation/meta/TypeQualifierValidat
or.class, jar:file:///home/hwarawala567/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/appengine-api-1.0-sdk-2.0.5.jar!/javax/ann
otation/meta/TypeQualifierValidator.class
[INFO] GCLOUD: 2023-02-06 10:39:04.618:WARN:oeja.AnnotationParser:qtp1487470647-15: javax.annotation.meta.When scanned from multiple locations: jar:file:///home/hwarawala
567/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/jsr305-3.0.2.jar!/javax/annotation/meta/When.class, jar:file:///home/hwarawala
567/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/WEB-INF/lib/appengine-api-1.0-sdk-2.0.5.jar!/javax/annotation/meta/When.class
[INFO] GCLOUD: 2023-02-06 10:39:04.666:INFO:oeja.AnnotationConfiguration:main: Scanning elapsed time=2036ms
[INFO] GCLOUD: 2023-02-06 10:39:05.208:INFO:oejsh.ContextHandler:main: Started c.g.a.t.d.j.DevAppEngineWebAppContext@424ebba3{/,file:///home/hwarawala567/java-docs-sample
s/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT/,AVAILABLE}{/home/hwarawala567/java-docs-samples/appengine-java8/helloworld/target/helloworld-1.0-SNAPSHOT}
[INFO] GCLOUD: 2023-02-06 10:39:05.218:INFO:oejs.session:main: DefaultSessionIdManager workerName=node0
[INFO] GCLOUD: 2023-02-06 10:39:05.221:INFO:oejs.session:main: node0 Scavenging disabled
[INFO] GCLOUD: 2023-02-06 10:39:05.236:INFO:oejs.AbstractConnector:main: Started NetworkTrafficSelectChannelConnector@1f1c7bf6{HTTP/1.1, (http/1.1)}{localhost:8080}
[INFO] GCLOUD: 2023-02-06 10:39:05.244:INFO:oejs.Server:main: Started @4915ms
[INFO] GCLOUD: Feb 06, 2023 10:39:05 AM com.google.appengine.tools.development.AbstractModule startup
[INFO] GCLOUD: INFO: Module instance default is running at http://localhost:8080/
[INFO] GCLOUD: Feb 06, 2023 10:39:05 AM com.google.appengine.tools.development.AbstractModule startup
[INFO] GCLOUD: INFO: The admin console is running at http://localhost:8080/_ah/admin
[INFO] GCLOUD: Feb 06, 2023 10:39:05 AM com.google.appengine.tools.development.DevAppServerImpl doStart
[INFO] GCLOUD: INFO: Dev App Server is now running
```


Dashboard – Experiment-no1
Cloud Shell
Hello App Engine Standard J
GitHub - GoogleCloudPlatfo
Run a web application using
8080-cs-b30a2b6d-4f12-45ce-b8da-9f43a3a2456c.cs-asia-southeast1-bool.cloudshell.dev/?authuser=0&redirectedPreviously=true
Type here to search
Watchlist +1.16%
ENG
16:23
06-02-2023

Hello App Engine -- Java 8!

This is Version: 11.0.16 OS: Linux User: hwarawala567.

Available Servlets:
Hello App Engine

Conclusion : Hence study and implementation of a Platform as a Service using Google App Engine completed successfully.

Experiment No. 2

Use GAE launcher to launch the web applications.

Aim :
To Install Google App Engine. Create hello world app and other simple web applications using python/java.
Procedure:
Use **Eclipse** to create a **Google App Engine** (GAE) **Java** project (hello world example), run it locally, and deploy it to Google App Engine account.

Tools used:
1. JDK 1.6
2. Eclipse 3.7 + Google Plugin for Eclipse
3. Google App Engine Java SDK 1.6.3.1

P.S Assume JDK1.6 and Eclipse 3.7 are installed.
1. Install Google Plugin for Eclipse

Read this guide – how to install Google Plugin for Eclipse. If you install the Google App Engine Java SDK together with "**Google Plugin for Eclipse**", then go to step 2, Otherwise, get the Google App Engine Java SDK and extract it.
2. Create New Web Application Project

In Eclipse toolbar, click on the Google icon, and select "**New Web Application Project…**"

Figure – New Web Application Project

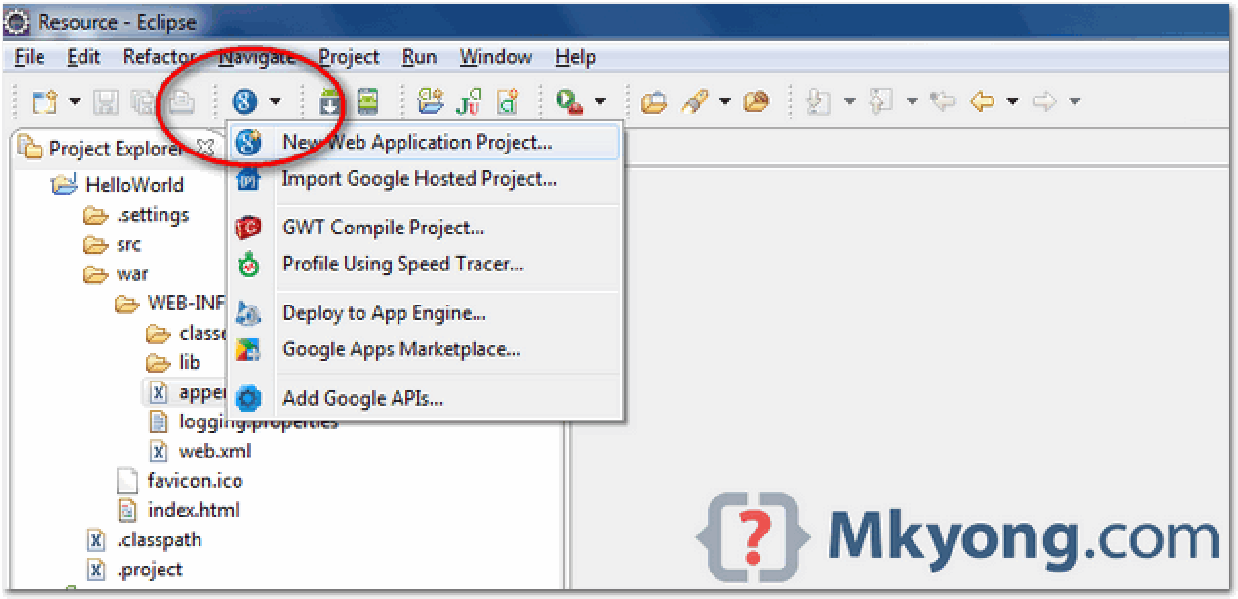
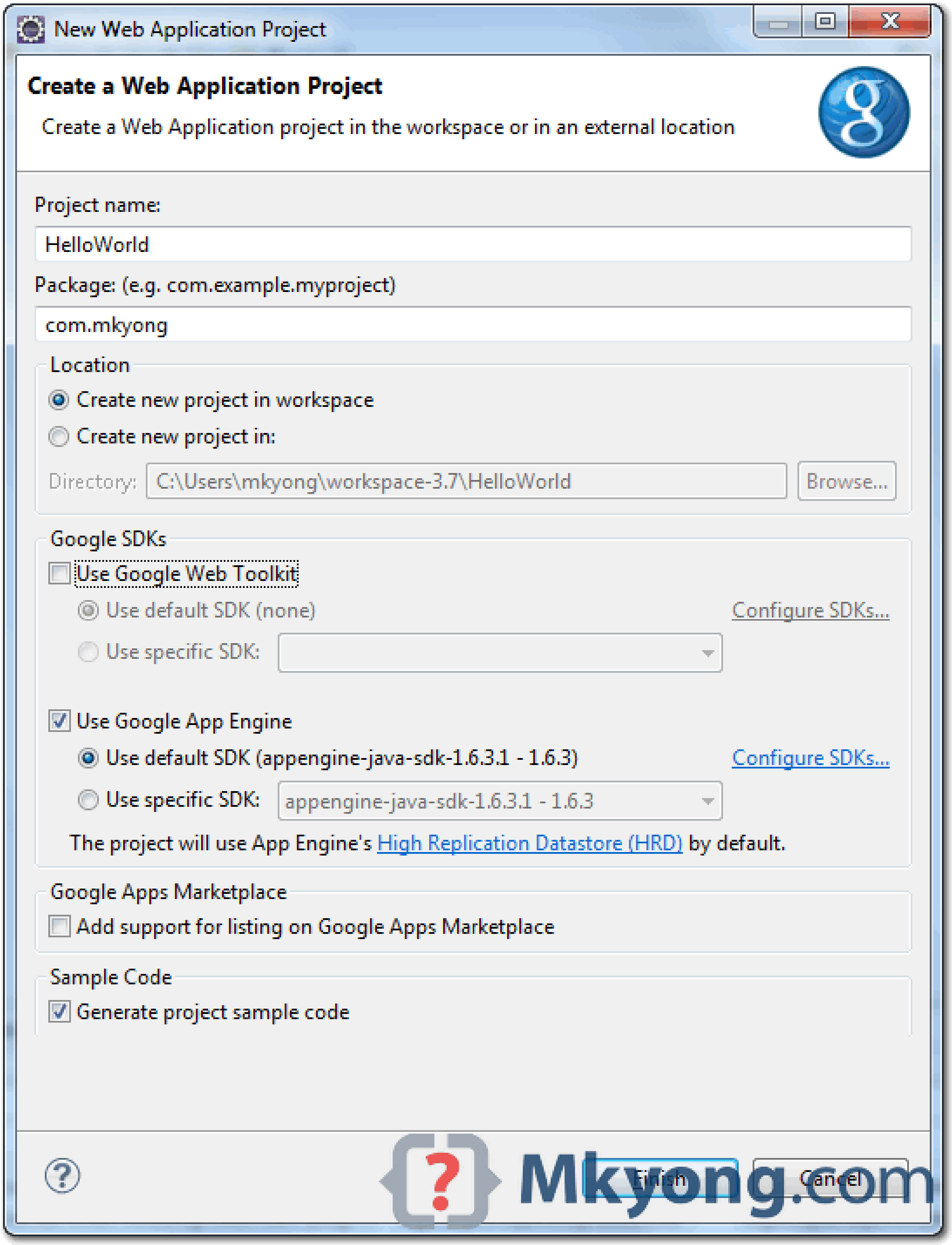


*Figure – Deselect the "**Google Web ToolKit**", and link your GAE Java SDK via the "**configure SDK**" link.*



Create a Web Application Project

Create a Web Application project in the workspace or in an external location

Project name: HelloWorld

Package: (e.g. com.example.myproject) com.mkyong

Location

● Create new project in workspace

○ Create new project in:

Directory: C:\Users\mkyong\workspace-3.7\HelloWorld

Google SDKs

□ Use Google Web Toolkit

● Use default SDK (none)

○ Use specific SDK:

☑ Use Google App Engine

● Use default SDK (appengine-java-sdk-1.6.3.1 - 1.6.3)

○ Use specific SDK: appengine-java-sdk-1.6.3.1 - 1.6.3

The project will use App Engine's High Replication Datastore (HRD) by default.

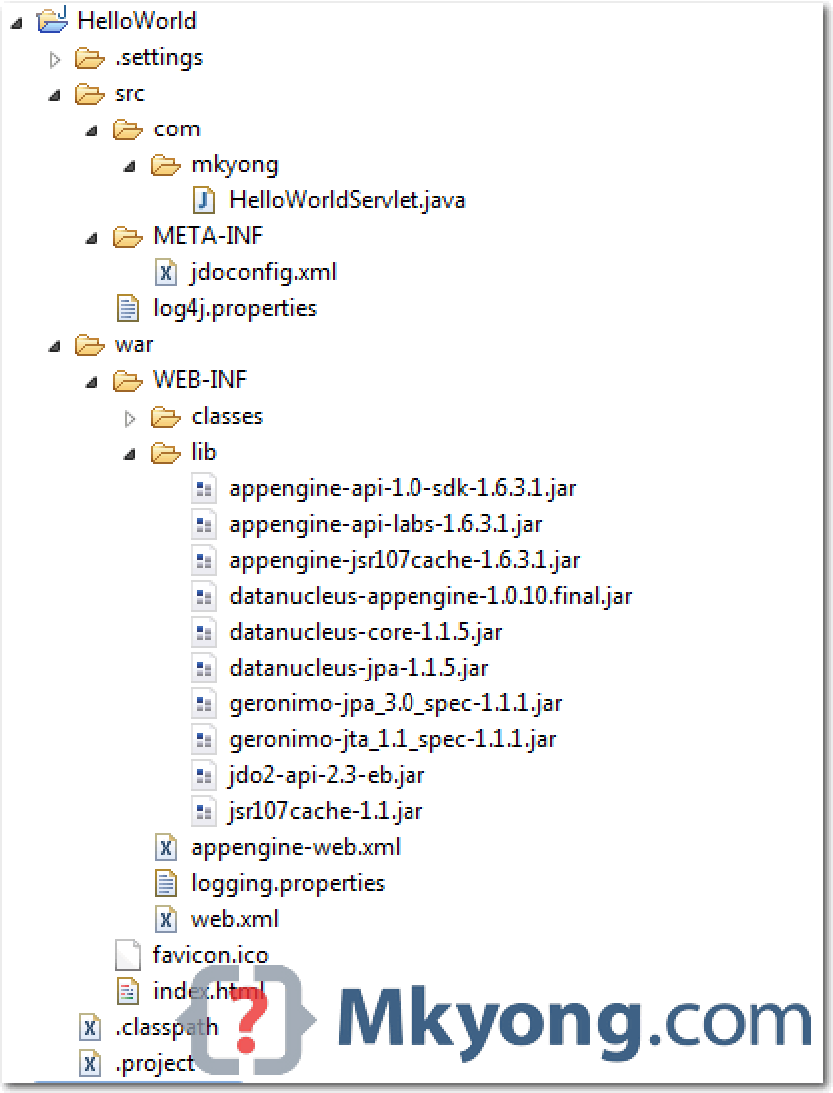
Google Apps Marketplace

□ Add support for listing on Google Apps Marketplace

Sample Code

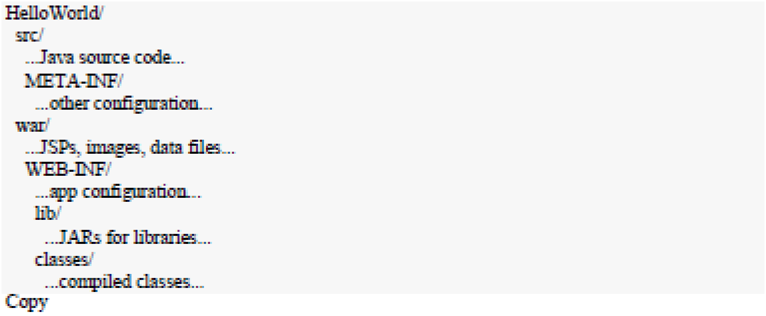
☑ Generate project sample code

*Figure – Deselect the "**Google Web ToolKit**", and link your GAE Java SDK via the "**configure SDK**" link*



```
HelloWorld
  .settings
  src
    com
      mkyong
        HelloWorldServlet.java
    META-INF
      jdoconfig.xml
    log4j.properties
  war
    WEB-INF
      classes
      lib
        appengine-api-1.0-sdk-1.6.3.1.jar
        appengine-api-labs-1.6.3.1.jar
        appengine-jsr107cache-1.6.3.1.jar
        datanucleus-appengine-1.0.10.final.jar
        datanucleus-core-1.1.5.jar
        datanucleus-jpa-1.1.5.jar
        geronimo-jpa_3.0_spec-1.1.1.jar
        geronimo-jta_1.1_spec-1.1.1.jar
        jdo2-api-2.3-eb.jar
        jsr107cache-1.1.jar
      appengine-web.xml
      logging.properties
      web.xml
    favicon.ico
    index.html
  .classpath
  .project
```

Nothing special, a standard Java web project structure.



```
HelloWorld/
  src/
    ...Java source code...
    META-INF/
      ...other configuration...
  war/
    ...JSPs, images, data files...
    WEB-INF/
      ...app configuration...
      lib/
        ...JARs for libraries...
      classes/
        ...compiled classes...
```

The extra is this file "appengine-web.xml", Google App Engine need this to run and deploy the application.

File : appengine-web.xml

File : appengine-web.xml

```
<?xml version="1.0" encoding="utf-8"?>
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">
<application></application>
<version>1</version>
<!-- Configure java.util.logging -->
<system-properties>
<property name="java.util.logging.config.file" value="WEB-INF/logging.properties"/>
</system-properties>
</appengine-web-app>
```

Copy

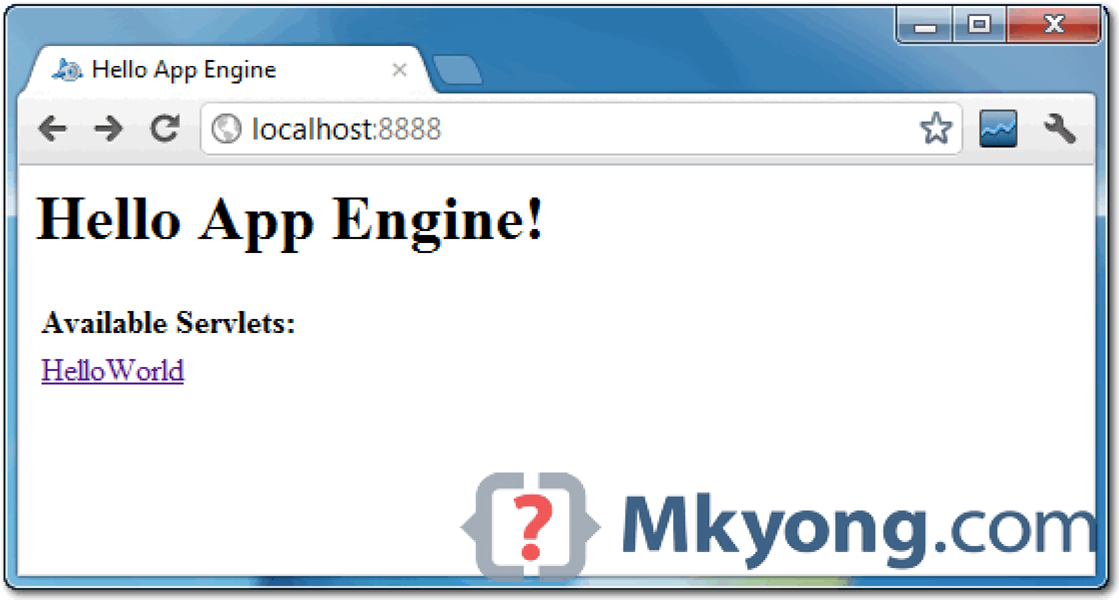
4. Run it local

Right click on the project and run as “**Web Application**“.

Eclipse console :

Copy

Access URL http://localhost:8888/, see output



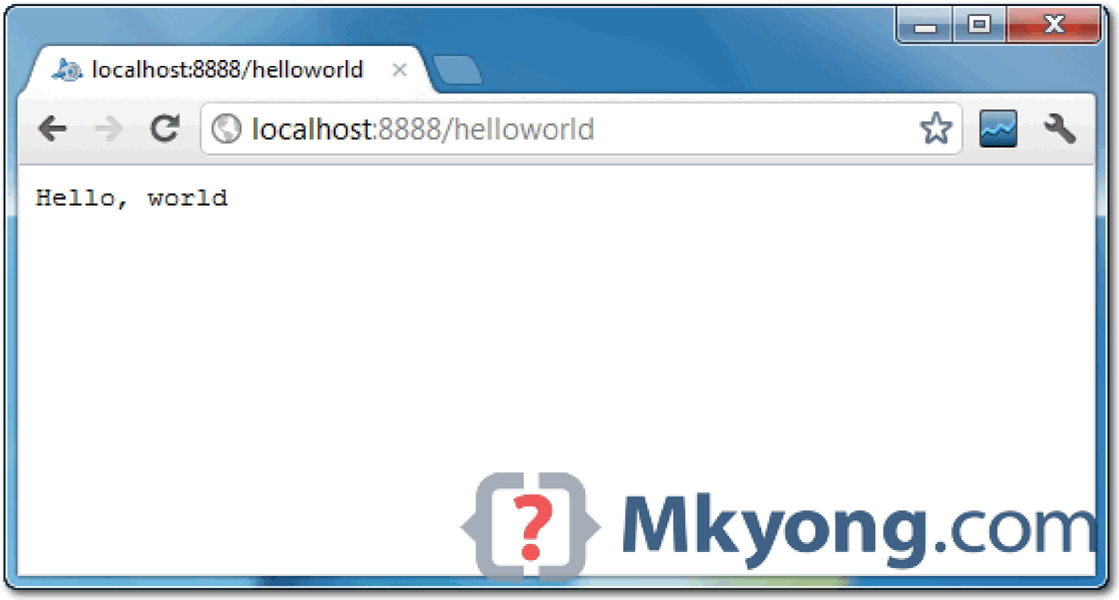
Hello App Engine!

Available Servlets:

HelloWorld

and also the hello world servlet – http://localhost:8888/helloworld

```
//...
INFO: The server is running at http://localhost:8888/
30 Mac 2012 11:13:01 PM com.google.appengine.tools.development.DevAppServerImpl start
INFO: The admin console is running at http://localhost:8888/_ah/admin
```



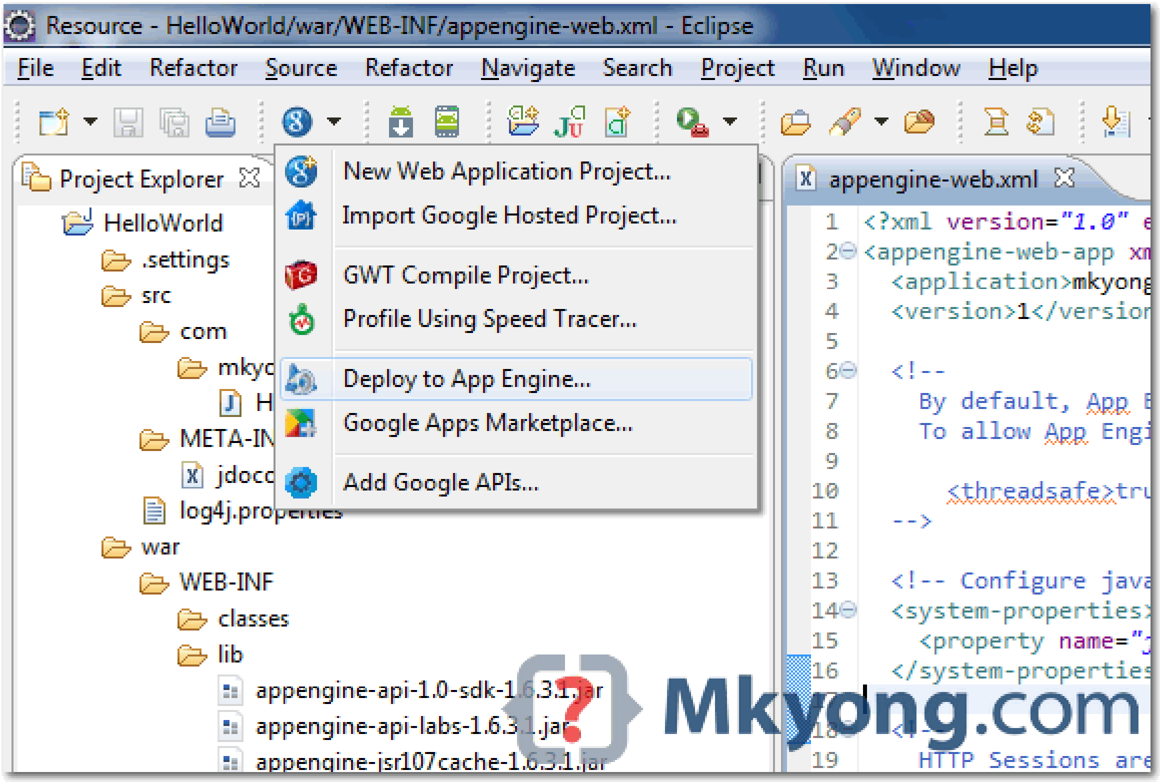
```
Hello, world
```

5. Deploy to Google App Engine
Register an account on https://appengine.google.com/, and create an application ID for your web application.
In this demonstration, I created an application ID, named “mkyong123”, and put it in appengine-web.xml.

File : appengine-web.xml

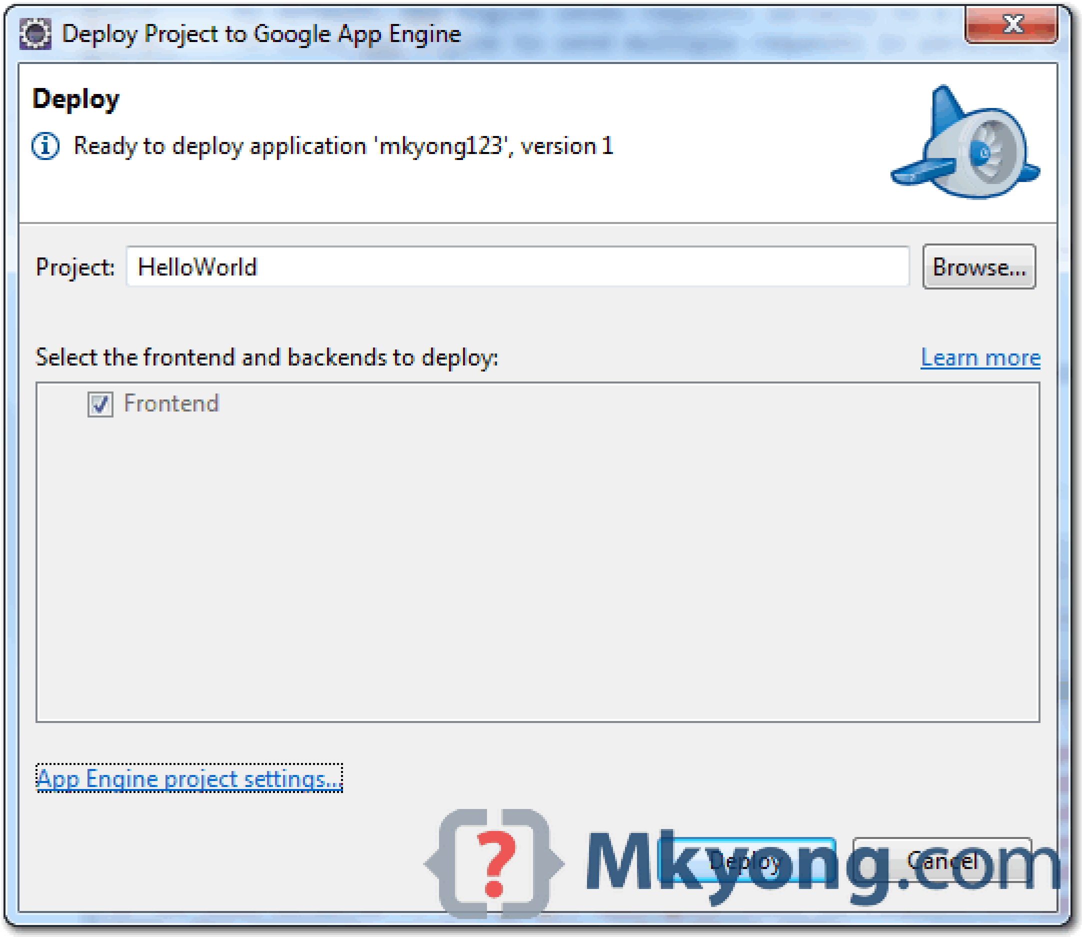
```
<?xml version="1.0" encoding="utf-8"?>
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">
<application>mkyong123</application>
<version>1</version>
<!-- Configure java.util.logging -->
<system-properties>
<property name="java.util.logging.config.file" value="WEB-INF/logging.properties"/>
</system-properties>
</appengine-web-app>
```

Copy
To deploy, see following steps:
Figure 1.1 – Click on GAE deploy button on the toolbar.


Resource - HelloWorld/war/WEB-INF/appengine-web.xml - Eclipse
File Edit Refactor Source Refactor Navigate Search Project Run Window Help
Project Explorer
HelloWorld
.settings
src
com
mkyo
H
META-IN
jdoco
log4j.properties
war
WEB-INF
classes
lib
appengine-api-1.0-sdk-1.6.3.1.jar
appengine-api-labs-1.6.3.1.jar
appengine-jsr107cache-1.6.3.1.jar
New Web Application Project...
Import Google Hosted Project...
GWT Compile Project...
Profile Using Speed Tracer...
Deploy to App Engine...
Google Apps Marketplace...
Add Google APIs...
appengine-web.xml
Mkyong.com

```
<?xml version="1.0" e
<appengine-web-app xm
  <application>mkyong
  <version>1</version

  <!--
    By default, App E
    To allow App Engi

      <threadsafe>tru
  -->

  <!-- Configure java
  <system-properties>
    <property name="j
  </system-properties

  <!--
    HTTP Sessions are
```

Figure 1.2 – Sign in with your Google account and click on the Deploy button.


Deploy Project to Google App Engine
Deploy

Ready to deploy application 'mkyong123', version 1

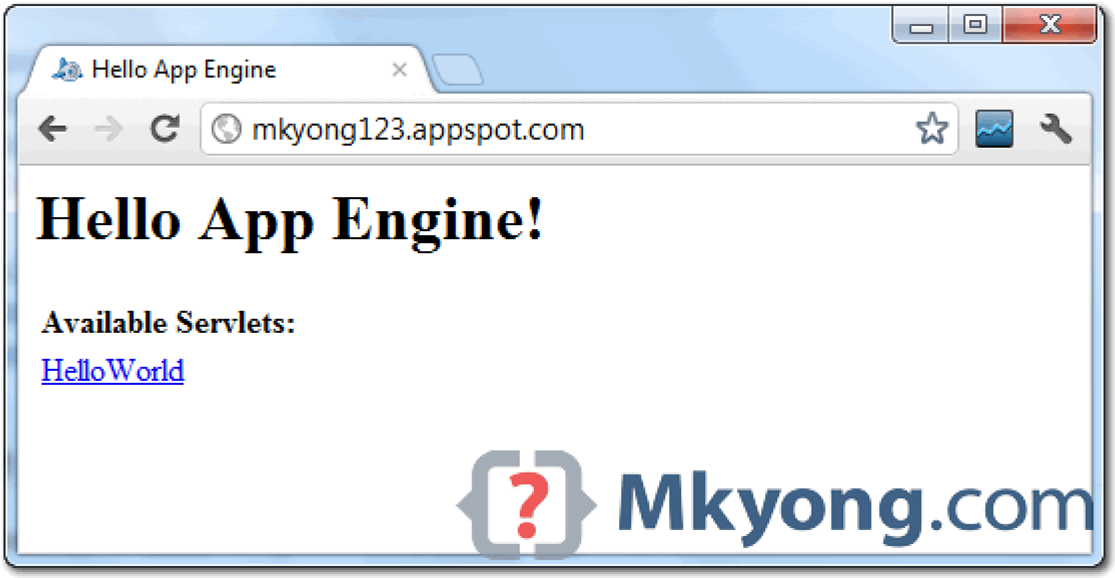
Project: HelloWorld

Select the frontend and backends to deploy:

☑ Frontend

App Engine project settings...
Deploy
Cancel

Figure 1.3 – If everything is fine, the hello world web application will be deployed to this URL – http://mkyong123.appspot.com/



Hello App Engine!

Available Servlets:

HelloWorld

Figure 1.3 – If everything is fine, the hello world web application will be deployed to this URL – http://mkyong123.appspot.com/

Done.

References

1. Google App Engine – Getting Started: Java
2. Google app engine Python hello world example using Eclipse

Viva Questions:

1. What is GAE?
2. What is ASP?
3. What is JSP?
4. Expalin the procedure to create the GAE
5. What is SDK?

Result:

Thus the GAE is installed and executed the hello world application.

Experiment No. 3

Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.

Aim:

To Simulate a cloud scenario using CloudSim and run a scheduling algorithm
that is not present in CloudSim.

Steps:

How to use CloudSim in Eclipse

CloudSim is written in Java. The knowledge you need to use CloudSim is basic Java programming and some basics about cloud computing. Knowledge of programming IDEs such as Eclipse or NetBeans is also helpful. It is a library and, hence, CloudSim does not have to be installed. Normally, you can unpack the downloaded package in any directory, add it to the Java classpath and it is ready to be used. Please verify whether Java is available on your system.

To use CloudSim in Eclipse:

1. Download CloudSim installable files

from *https://code.google.com/p/cloudsim/downloads/list and unzip*

2. Open Eclipse
3. Create a new Java Project: File -> New
4. Import an unpacked CloudSim project into the new Java Project

The first step is to initialise the CloudSim package by initialising the CloudSim library, as follows

CloudSim.init(num_user, calendar, trace_flag)

5. Data centres are the resource providers in CloudSim; hence, creation of data centres is a second step. To create Datacenter, you need the DatacenterCharacteristics object that stores the properties of a data centre such as architecture, OS, list of machines, allocation policy that covers the time or spaceshared, the time zone and its price:

Datacenter datacenter9883 = new Datacenter(name, characteristics, new VmAllocationPolicySimple(hostList), s

6. The third step is to create a broker:

DatacenterBroker broker = createBroker();

7. The fourth step is to create one virtual machine unique ID of the VM, userId ID of the VM's owner, mips, number Of Pes amount of CPUs, amount of RAM, amount of bandwidth, amount of storage, virtual machine monitor, and cloudletScheduler policy for cloudlets:

Vm vm = new Vm(vmid, brokerId, mips, pesNumber, ram, bw, size, vmm, new CloudletSchedulerTimeShared())

8. Submit the VM list to the broker:

broker.submitVmList(vmlist)

9. Create a cloudlet with length, file size, output size, and utilisation model:

```
Cloudlet cloudlet = new Cloudlet(id, length, pesNumber, fileSize, outputSize, utilizationModel,
utilizationMode
```

10. Submit the cloudlet list to the broker:

```
broker.submitCloudletList(cloudletList)
```

Sample Output from the Existing Example:

```
Starting CloudSimExample1... Initialising...
Starting CloudSim version 3.0 Datacenter_0 is starting...
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>null
Broker is starting... Entities started.
: Broker: Cloud Resource List received with 1 resource(s) 0.0: Broker: Trying to Create VM #0
in Datacenter_0
: Broker: VM #0 has been created in Datacenter #2, Host #0

0.1: Broker: Sending cloudlet 0 to VM #0
400.1: Broker: Cloudlet 0 received
: Broker: All Cloudlets executed. Finishing... 400.1: Broker: Destroying VM #0

Broker is shutting down... Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down. Datacenter_0 is
shutting down...
Broker is shutting down... Simulation completed.
Simulation completed.

========== OUTPUT ==========
Cloudlet ID STATUS Data center ID VM ID Time Start Time Finish Time 0 SUCCESS 2 0 400
0.1 400.1
*****Datacenter:
Datacenter_0***** User id Debt
3 35.6
CloudSimExample1 finished!
```

RESULT:

The simulation was successfully executed.

Experiment No. 4

Find a procedure to transfer the files from one virtual machine to another virtual machine.

Steps:

1. You can copy few (or more) lines with *copy & paste* mechanism.

For this you need to share clipboard between host OS and guest OS, installing Guest Addition on both the virtual machines (probably setting *bidirectional* and restarting them). You *copy* from *guest OS* in the clipboard that is shared with the *host OS*.
Then you *paste* from the *host OS* to the second *guest OS*.

2. You can enable drag and drop too with the same method (Click on the machine, settings, general, advanced, drag and drop: set to *bidirectional*)

3. You can have common *Shared Folders* on both virtual machines and use one of the directory shared as buffer to copy.

Installing Guest Additions you have the possibility to set Shared Folders too. As you put a file in a shared folder from *host OS* or from *guest OS*, is immediately visible to the other. (Keep in mind that can arise some problems for date/time of the files when there are different clock settings on the different virtual machines).
If you use the same folder shared on more machines you can exchange files directly copying them in this folder.

4. You can use usual method to copy files between 2 different computer with client-server application. (e.g. scp with sshd active for linux, winscp... you can get some info about SSH servers e.g. here)

You need an active server (sshd) on the receiving machine and a client on the sending machine. Of course you need to have the authorization setted (via password or, better, via an automatic authentication method).
Note: many Linux/Ubuntu distribution install sshd by default: you can see if it is running with pgrep sshd from a shell. You can install with sudo apt-get install openssh-server.

5. You can mount part of the file system of a virtual machine via NFS or SSHFS on the other, or you can share file and directory with Samba.

You may find interesting the article Sharing files between guest and host without VirtualBox shared folders with detailed step by step instructions.

You should remember that you are dialling with a little network of machines with different operative systems, and in particular:
 Each virtual machine has its own operative system running on and acts as a physical machine.

 Each virtual machine is an instance of a program *owned* by an *user* in the hosting operative system and should undergo the restrictions of the *user* in the *hosting OS*.

E.g Let we say that Hastur and Meow are users of the hosting machine, but they did not allow each other to see their directories (no read/write/execute authorization). When each of them run a virtual machine, for the hosting OS those virtual machine are two normal programs owned by Hastur and Meow and cannot see the private directory of the other user. This is a restriction due to the *hosting OS*. It's easy to overcame it: it's enough to give authorization to read/write/execute to a directory or to chose a different directory in which both users can read/write/execute.
 Windows likes mouse and Linux fingers. :-)

I mean I suggest you to enable *Drag & drop* to be cosy with the Windows machines and the *Shared folders* or to be cosy with Linux.
When you will need to be fast with Linux you will feel the need of ssh-keygen and
to Generate once SSH Keys to copy files on/from a remote machine without writing password anymore. In this way it functions bash auto-completion remotely too!

PROCEDURE:

Steps:

1. Open Browser, type localhost:9869

2. Login using username: oneadmin, password: opennebula

3. Then follow the steps to migrate VMs
a. Click on infrastructure

b. Select clusters and enter the cluster name

c. Then select host tab, and select all host

d. Then select Vnets tab, and select all vnet

e. Then select datastores tab, and select all datastores

f. And then choose host under infrastructure tab

g. Click on + symbol to add new host, name the host then click on create.

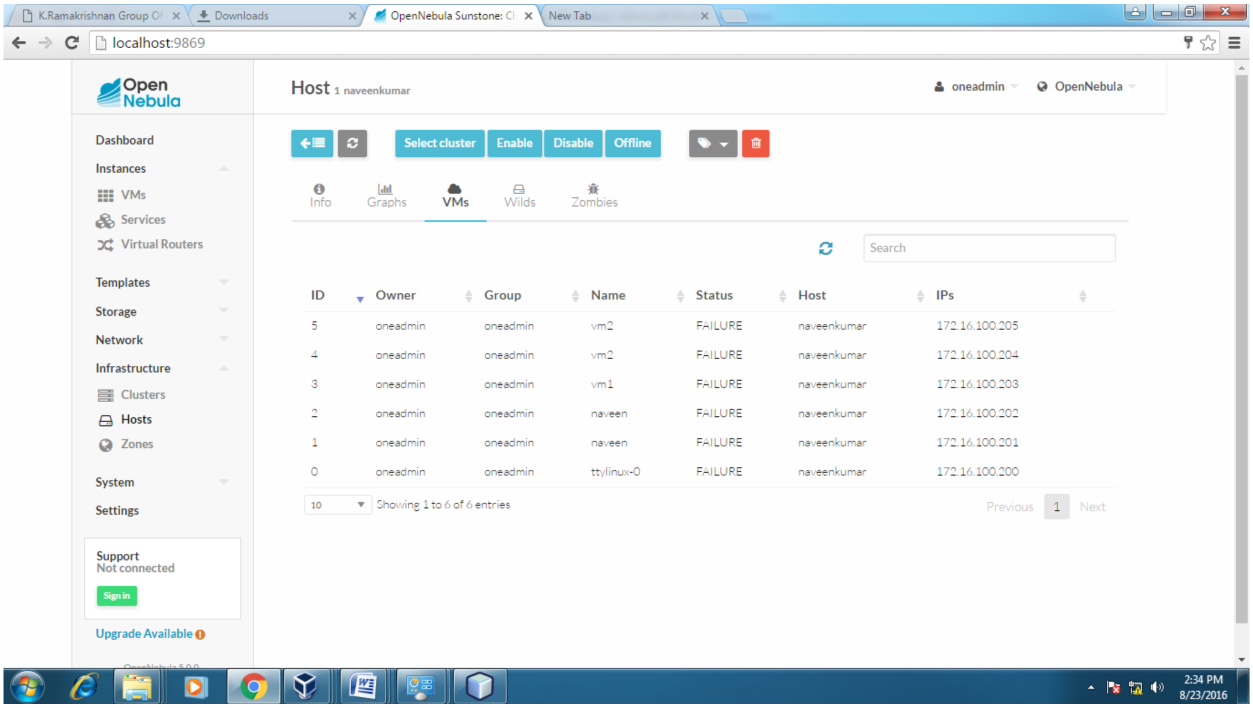
4. on instances, select VMs to migrate then follow the stpes
a. Click on 8th icon ,the drop down list display

b. Select migrate on that ,the popup window display

c. On that select the target host to migrate then click on migrate.

Before migration

Host:SACET

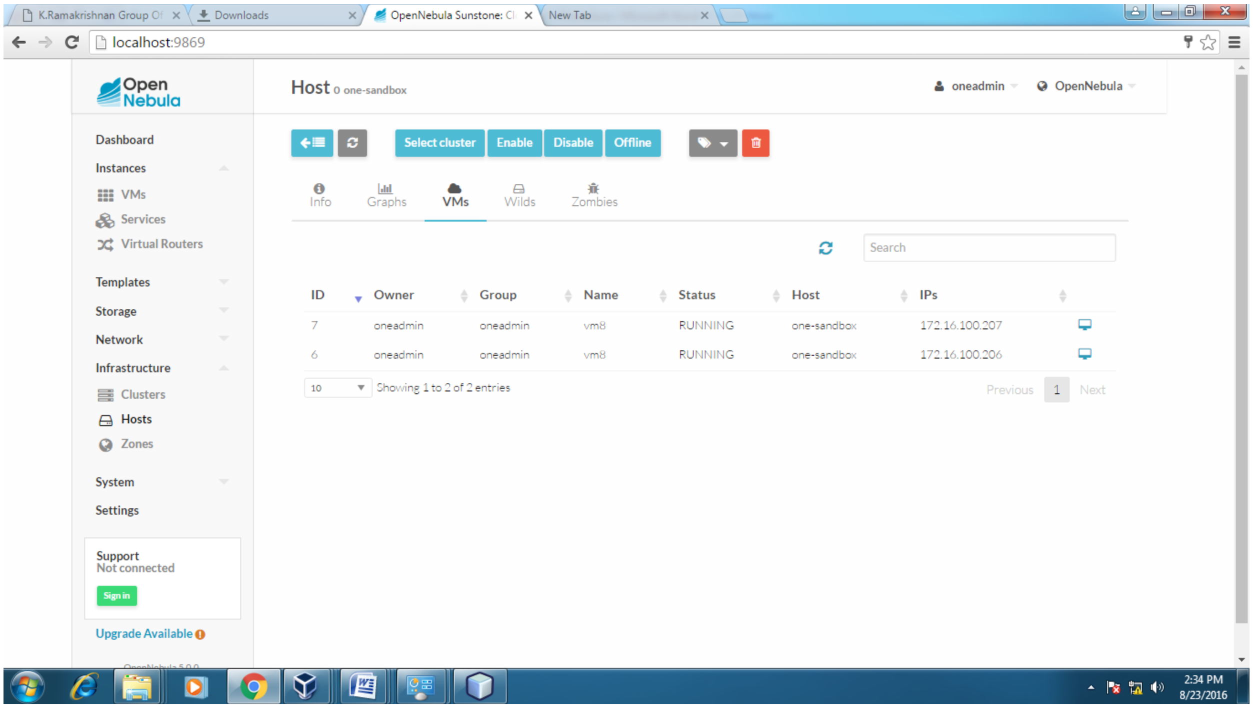


Host 1 naveenkumar

ID	Owner	Group	Name	Status	Host	IPs
5	oneadmin	oneadmin	vm2	FAILURE	naveenkumar	172.16.100.205
4	oneadmin	oneadmin	vm2	FAILURE	naveenkumar	172.16.100.204
3	oneadmin	oneadmin	vm1	FAILURE	naveenkumar	172.16.100.203
2	oneadmin	oneadmin	naveen	FAILURE	naveenkumar	172.16.100.202
1	oneadmin	oneadmin	naveen	FAILURE	naveenkumar	172.16.100.201
0	oneadmin	oneadmin	ttylinux-0	FAILURE	naveenkumar	172.16.100.200

Showing 1 to 6 of 6 entries

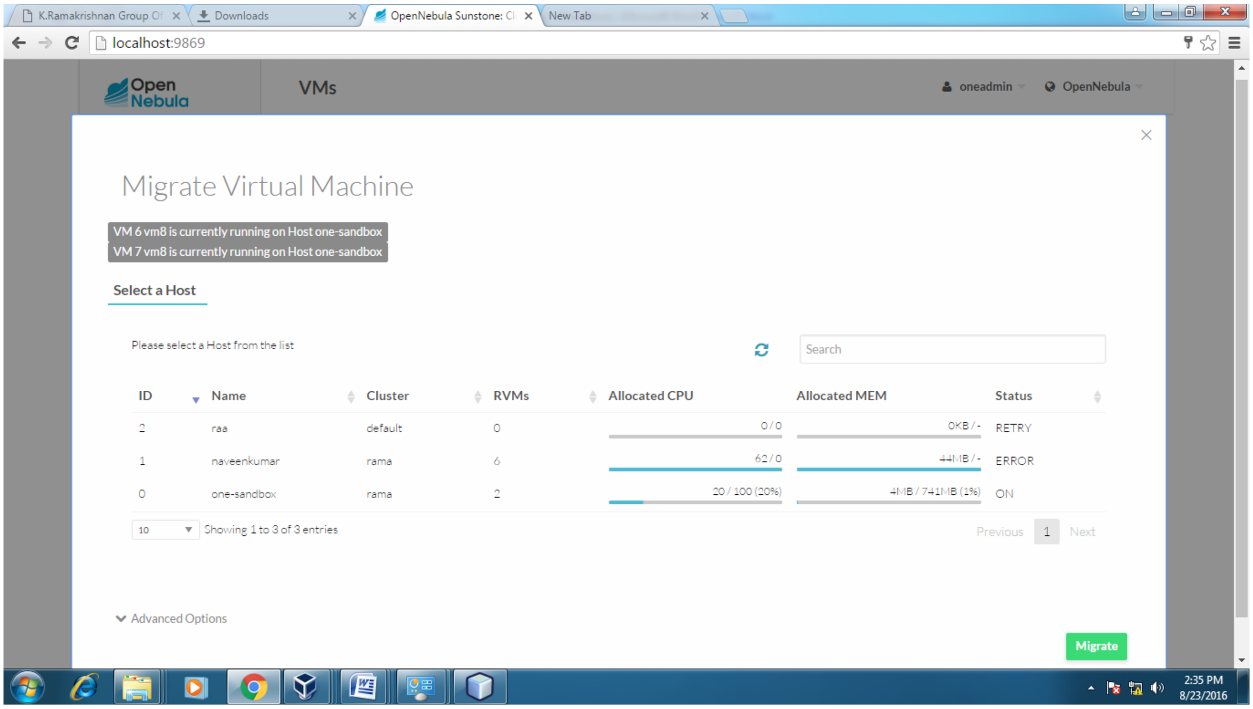
Host:one-sandbox



Host 0 one-sandbox

ID	Owner	Group	Name	Status	Host	IPs
7	oneadmin	oneadmin	vm8	RUNNING	one-sandbox	172.16.100.207
6	oneadmin	oneadmin	vm8	RUNNING	one-sandbox	172.16.100.206

Showing 1 to 2 of 2 entries



Migrate Virtual Machine

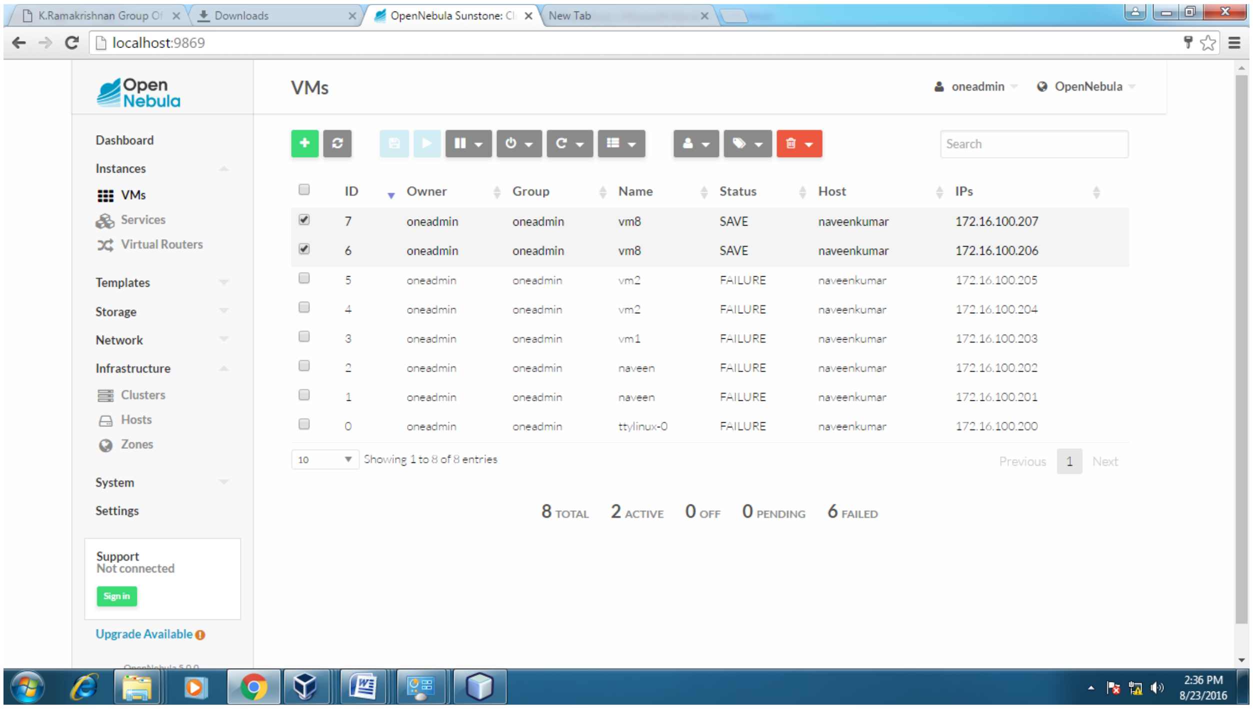
VM 6 vm8 is currently running on Host one-sandbox
VM 7 vm8 is currently running on Host one-sandbox

Select a Host

Please select a Host from the list

ID	Name	Cluster	RVMs	Allocated CPU	Allocated MEM	Status
2	raa	default	0	0 / 0	0KB / -	RETRY
1	naveenkumar	rama	6	62 / 0	44MB / -	ERROR
0	one-sandbox	rama	2	20 / 100 (20%)	4MB / 741MB (1%)	ON

10 Showing 1 to 3 of 3 entries

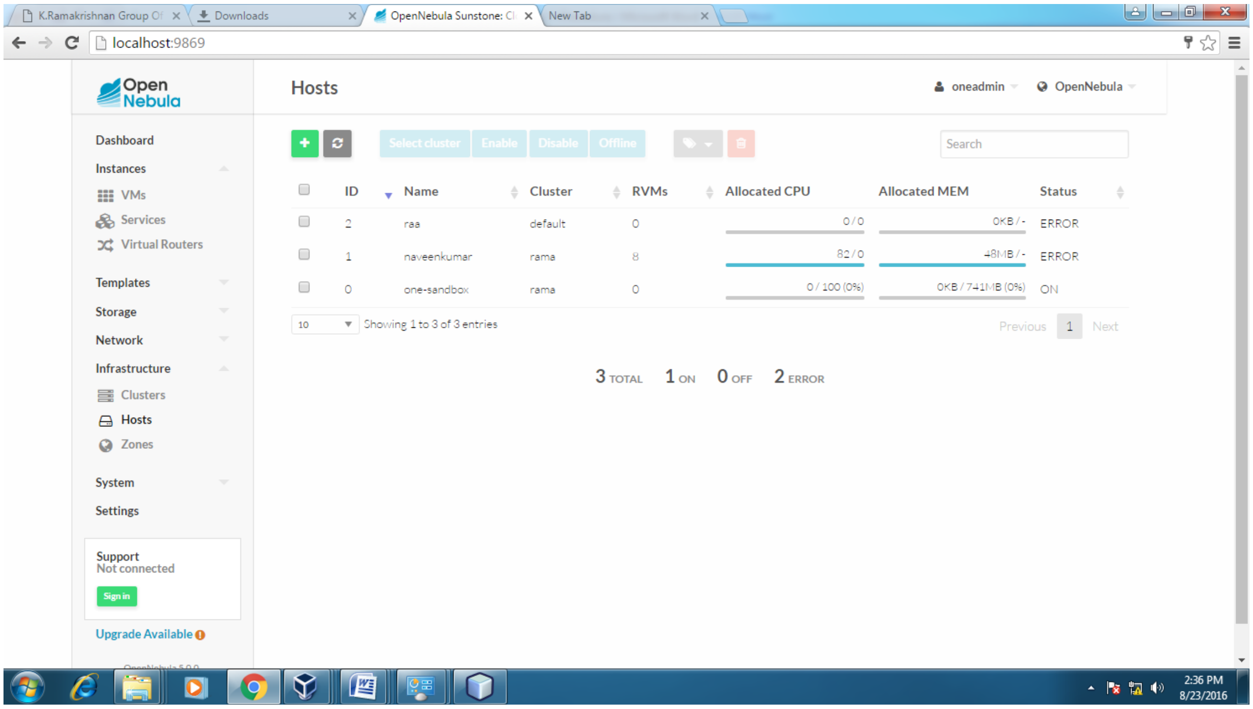


	ID	Owner	Group	Name	Status	Host	IPs
☑	7	oneadmin	oneadmin	vm8	SAVE	naveenkumar	172.16.100.207
☑	6	oneadmin	oneadmin	vm8	SAVE	naveenkumar	172.16.100.206
□	5	oneadmin	oneadmin	vm2	FAILURE	naveenkumar	172.16.100.205
□	4	oneadmin	oneadmin	vm2	FAILURE	naveenkumar	172.16.100.204
□	3	oneadmin	oneadmin	vm1	FAILURE	naveenkumar	172.16.100.203
□	2	oneadmin	oneadmin	naveen	FAILURE	naveenkumar	172.16.100.202
□	1	oneadmin	oneadmin	naveen	FAILURE	naveenkumar	172.16.100.201
□	0	oneadmin	oneadmin	ttylinux-0	FAILURE	naveenkumar	172.16.100.200

10 Showing 1 to 8 of 8 entries

8 TOTAL 2 ACTIVE 0 OFF 0 PENDING 6 FAILED

After Migration:



Hosts

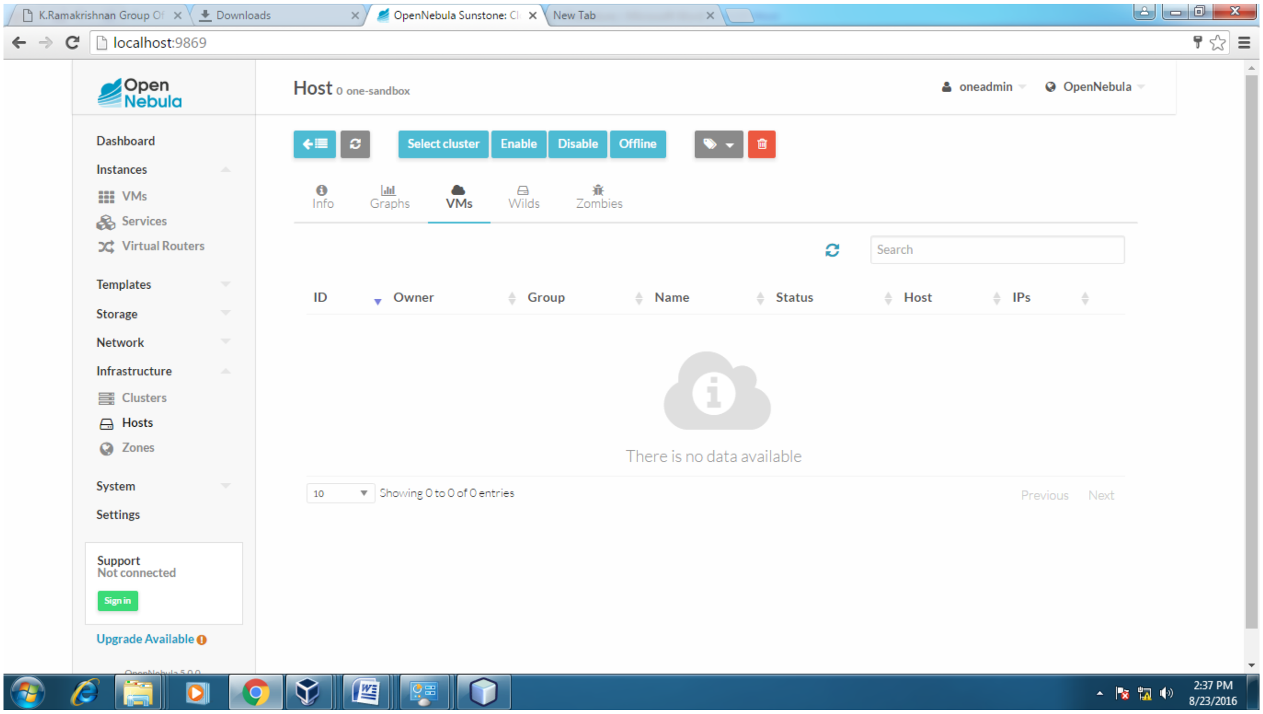
ID	Name	Cluster	RVMs	Allocated CPU	Allocated MEM	Status
2	raa	default	0	0 / 0	0KB / -	ERROR
1	naveenkumar	rama	8	82 / 0	48MB / -	ERROR
0	one-sandbox	rama	0	0 / 100 (0%)	0KB / 741MB (0%)	ON

10 Showing 1 to 3 of 3 entries

Previous 1 Next

3 TOTAL 1 ON 0 OFF 2 ERROR

Host:one-sandbox



Host 0 one-sandbox

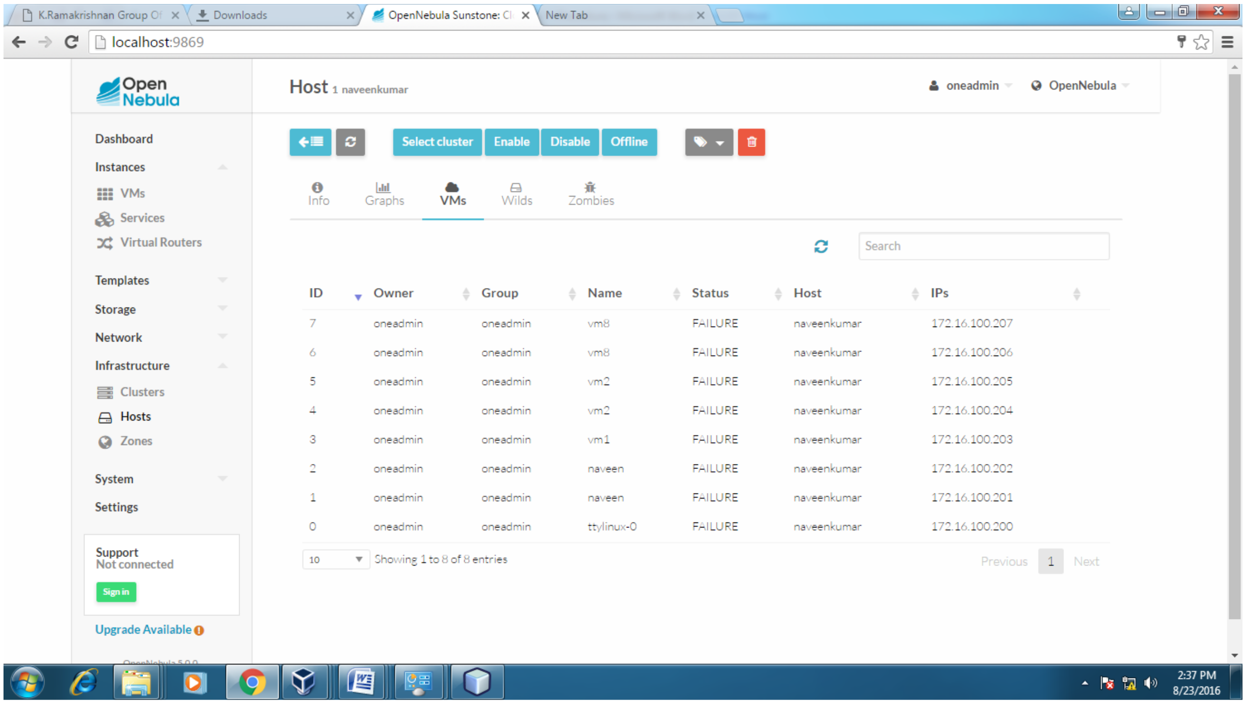
ID	Owner	Group	Name	Status	Host	IPs

There is no data available

10 Showing 0 to 0 of 0 entries

Previous Next

Host:SACET



Host 1 naveenkumar

ID	Owner	Group	Name	Status	Host	IPs
7	oneadmin	oneadmin	vm8	FAILURE	naveenkumar	172.16.100.207
6	oneadmin	oneadmin	vm8	FAILURE	naveenkumar	172.16.100.206
5	oneadmin	oneadmin	vm2	FAILURE	naveenkumar	172.16.100.205
4	oneadmin	oneadmin	vm2	FAILURE	naveenkumar	172.16.100.204
3	oneadmin	oneadmin	vm1	FAILURE	naveenkumar	172.16.100.203
2	oneadmin	oneadmin	naveen	FAILURE	naveenkumar	172.16.100.202
1	oneadmin	oneadmin	naveen	FAILURE	naveenkumar	172.16.100.201
0	oneadmin	oneadmin	ttylinux-0	FAILURE	naveenkumar	172.16.100.200

10 Showing 1 to 8 of 8 entries

APPLICATIONS:

Easily migrate your virtual machine from one pc to another.

Result:

Thus the file transfer between VM was successfully completed.

Experiment No. 5

Launch using Amazon Educate Account.

2. Objectives:

- Launch a web server with termination protection enabled
- Monitor Your EC2 instance
- Modify the security group that your web server is using to allow HTTP access
- Resize your EC2 instance to scale
- Explore Amazon EC2 limits
- Test termination protection
- Terminate your EC2 instance

3. Hardware / Software Required: Internet, AWS console

4. **Theory**:

Amazon EC2 is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing simple and intuitive to use.

With the web service interface of Amazon EC2, you can obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources. You can run application servers, blogs, batch processing, and more on the Amazon computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes so that you can quickly scale capacity both up and down as your computing requirements change. Amazon EC2 changes the economics of computing so that you pay for only the capacity that you actually use. Amazon EC2 provides developers the tools to build failure-resilient applications and isolate themselves from common failure scenarios.

Steps:-

Task 1: Launching your EC2 instance

In this task, you launch an EC2 instance with termination protection. Termination protection prevents you from accidentally terminating an EC2 instance. You also deploy your instance with a user data script in order to deploy a simple web server.

In the AWS Management Console on the Services menu, choose EC2.

In the left navigation pane, choose EC2 Dashboard to ensure that you are on the dashboard page.

Choose Launch instance, and then select Launch instance.

Step 1: Name your EC2 instance

Using tags, you can categorize your AWS resources in different ways (for example, by purpose, owner, or environment). This categorization is useful when you have many resources of the same type. You can quickly identify a specific resource based on the tags that you have assigned to it. Each tag consists of a key and a value, both of which you define.
When you name your instance, AWS creates a key-value pair. The key for this pair is Name, and the value is the name you enter for your EC2 instance.
8.In the Name and tags section, for Name, enter Web-Server
9.Choose the Add additional tags link.
10.From the Resource types dropdown list, ensure that both Instances and Volumes are selected.
Step 2: Choose an Amazon Machine Image (AMI)
An Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud. An AMI includes the following:
•A template for the root volume for the instance (for example, an operating system or an application server with applications)
•Launch permissions that control which AWS accounts can use the AMI to launch instances
•A block device mapping that specifies the volumes to attach to the instance when it is launched
The Quick Start list contains the most commonly used AMIs. You can also create your own AMI or select an AMI from the AWS Marketplace, an online store where you can sell or buy software that runs on AWS.
11.Locate the Application and OS Images (Amazon Machine Image) section. It is just below the Name and tags section.
12.In the AMI Machine Image (AMI) box, notice that Amazon Linux 2 AMI is selected by default. Keep this setting.
Step 3: Choose an instance type
Amazon EC2 provides a wide selection of instance types that are optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more instance sizes so that you can scale your resources to the requirements of your target workload.
In this step, you choose a t2.micro instance. This instance type has 1 virtual CPU and 1 GiB of memory.
13.Keep the default instance type, t2.micro.
Step 4: Configure a key pair
Amazon EC2 uses public key cryptography to encrypt and decrypt login information. To log in to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance.
In this lab, you do not log in to your instance, so you do not require a key pair.
14.In the Key pair (login) section, from the Key pair name - required dropdown list, choose

Proceed without a key pair.

Step 5: Configure the network settings

You use this pane to configure networking settings.

The virtual private cloud (VPC) indicates which VPC you want to launch the instance into. You can have multiple VPCs, including different ones for development, testing, and production.

15.In the Network settings section, choose Edit.

16.From the VPC - required dropdown list, choose Lab VPC.

The Lab VPC was created using an AWS CloudFormation template during the setup process of your lab. This VPC includes two public subnets in two different Availability Zones.

17.In the Network settings section, for Security group name - required, enter Web Server security group

A security group acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add rules to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group.

In this lab, you do not log in to your instance using SSH. Removing SSH access improves the security of the instance.

18.To delete the existing SSH rule, next to Security group rule 1, choose Remove.

Step 6: Add storage

Amazon EC2 stores data on a network-attached virtual disk called Amazon Elastic Block Store (Amazon EBS).

You launch the EC2 instance using a default 8 GiB disk volume. This is your root volume (also known as a boot volume).

19.In the Configure storage pane, keep the default storage configuration.

Step 7: Configure advanced details

20.Expand the Advanced details pane.

When you no longer require an EC2 instance, you can terminate it, which means that the instance stops, and Amazon EC2 releases the instance's resources. You cannot restart a terminated instance. If you want to prevent your users from accidentally terminating the instance, you can enable termination protection for the instance, which prevents users from terminating instances.

21.From the Termination protection dropdown list, choose Enable.

When you launch an instance in Amazon EC2, you have the option of passing user data to the instance. These commands can be used to perform common automated configuration tasks and even run scripts after the instance starts.

22.Copy the following commands, and paste them into theIn the User data text box.

```
#!/bin/bash
yum -y install httpd
systemctl enable httpd
systemctl start httpd
echo '<html><h1>Hello From Your Web Server!</h1></html>' > /var/www/html/index.html
```

The script does the following:

o Install an Apache web server (httpd)

o Configure the web server to automatically start on boot

o Activate the Web server

o Create a simple web page

Step 8: Launch an EC2 instance

Now that you have configured your EC2 instance settings, it is time to launch your instance.

23.In the Summary section, choose Launch instance.

24.Choose View all instances

The instance appears in a Pending state, which means that it is being launched. It then changes to Running, which indicates that the instance has started booting. There will be a short time before you can access the instance.

The instance receives a public Domain Name System (DNS) name that you can use to contact the instance from the Internet.

Next to your Web-Server, select the check box. The Details tab displays detailed information about your instance.

To view more information in the Details tab, drag the window divider upward.

Review the information displayed in the Details, Security and Networking tabs.

25.Wait for your instance to display the following:

Note: Refresh if needed.

o Instance State: Running

o Status Checks: 2/2 checks passed

Task 2: Monitoring your instance

Monitoring is an important part of maintaining the reliability, availability, and performance of your EC2 instances and your AWS solutions.

26.Choose the Status checks tab.

With instance status monitoring, you can quickly determine whether Amazon EC2 has detected any problems that might prevent your instances from running applications. Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues.

Notice that both the System reachability and Instance reachability checks have passed.

27.Choose the Monitoring tab.

This tab displays Amazon CloudWatch metrics for your instance. Currently, there are not many metrics to display because the instance was recently launched.

You can chose a graph to see an expanded view.

Amazon EC2 sends metrics to Amazon CloudWatch for your EC2 instances. Basic (5 minute) monitoring is enabled by default. You can enable detailed (1 minute) monitoring.

28.At the top of the page, choose the Actions dropdown menu. Select Monitor and troubleshoot Get system log.

The system log displays the console output of the instance, which is a valuable tool for diagnosing problems. It is especially useful for troubleshooting kernel problems and service configuration issues that could cause an instance to terminate or become unreachable before its SSH daemon can be started. If you do not see a system log, wait a few minutes and then try again.

29.Scroll through the output, and note that the HTTP package was installed from the user data that you added when you created the instance. The entries in the system log should be similar to the following example:

```
[ 26.760639] cloud-init[3280]: Installed:
[ 26.770051] cloud-init[3280]: httpd.x86_64 0:2.4.52-1.amzn2
[ 26.777748] cloud-init[3280]: Dependency Installed:
[ 26.781750] cloud-init[3280]: apr.x86_64 0:1.7.0-9.amzn2
[ 26.793739] cloud-init[3280]: apr-util.x86_64 0:1.6.1-5.amzn2.0.2
[ 26.796595] cloud-init[3280]: apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2
[ 26.805964] cloud-init[3280]: generic-logos-httpd.noarch 0:18.0.0-4.amzn2
[ 26.817765] cloud-init[3280]: httpd-filesystem.noarch 0:2.4.52-1.amzn2
[ 26.829760] cloud-init[3280]: httpd-tools.x86_64 0:2.4.52-1.amzn2
[ 26.833753] cloud-init[3280]: mailcap.noarch 0:2.1.41-2.amzn2
[ 26.845761] cloud-init[3280]: mod_http2.x86_64 0:1.15.19-1.amzn2.0.1
[ 26.849762] cloud-init[3280]: Complete!
```

30.To return to the Amazon EC2 dashboard, choose Cancel.

31.With your Web-Server selected, choose the Actions dropdown menu, and select Monitor and troubleshoot Get instance screenshot.

This option shows you what your EC2 instance console would look like if a screen were attached to it. It is essentially a command line interface.

If you are unable to reach your instance via SSH or RDP, you can capture a screenshot of your instance and view it as an image. This option provides visibility about the status of the instance and allows for quicker troubleshooting.

32.At the bottom of the page, choose Cancel.

Task 3: Updating your security group and accessing the web server

When you launched the EC2 instance, you provided a script that installed a web server and created a simple web page. In this task, you access content from the web server.

33.Select the check box next to the Amazon EC2 Web-Server that you created, and then choose the Details tab.

34.Copy the Public IPv4 address of your instance to your clipboard.

35.In your web browser, open a new tab, paste the IP address that you just copied, and then press Enter.

Question: Are you able to access your web server? Why not?

You are not currently able to access your web server because the security group is not permitting inbound traffic on port 80, which is used for HTTP web requests. This is a demonstration of how to use a security group as a firewall to restrict the network traffic that is allowed in and out of an instance.

To correct this issue, you now update the security group to permit web traffic on port 80.

36.Keep the browser tab open, but return to the EC2 Management Console tab.

37.In the left navigation pane, choose Security Groups.

38.Next to Web Server security group, select the check box.

39.Choose the Inbound rules tab.

The security group currently has no rules.

40.Choose Edit inbound rules, and then choose Add rule and configure the following options:

o Type: Choose HTTP.

o Source: Choose Anywhere-IPv4.

Note: Notice the "Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." While this is true and common best practice, this lab allows access from any IP address (Anywhere) to simplify both the security group configuration and testing of the website running on your EC2 instance.

41.Choose Save rules

42.Return to the web server browser tab with the public IPv4 address that you previously opened, and choose to refresh the page.

You should see the message Hello From Your Web Server!

Task 4: Resizing your instance - instance type and EBS volume

As your needs change, you might find that your instance is over utilized (too small) or under utilized (too large). If so, you can change the instance type. For example, if a t2.micro instance is too small for its workload, you can change it to an m5.medium instance. Similarly, you can change the size of a disk.

Stop your instance

Before you can resize an instance, you must stop it.

When you stop an instance, it is shut down. There is no charge for a stopped EC2 instance, but the storage charge for attached EBS volumes remains.

43.On the EC2 Management Console, in the left navigation pane, choose Instances.

The check box next to Web Server should already be selected.

44.At the top of the page, select the Instance state dropdown menu, and choose Stop instance.

45.In the Stop instance? pop-up window, choose Stop.

Your instance performs a normal shutdown and then stops running.

46.Wait for the Instance state to display Stopped.

Change the instance type

47.Select the check box next to your Web-Server. From the Actions dropdown menu, select Instance settings Change instance type, and then configure the following option:

o Instance type: Select t2.nano.

48.Choose Apply.

15

When the instance is started again, it is a t2.nano instance.

Note: You are restricted from using other instance types in this lab.

Resize the EBS volume

49.In the left navigation menu, choose Volumes.

50.Select the check box for the one volume that is listed, which is attached to your Web-Server instance.

51.In the Actions dropdown menu, select Modify Volume.

The disk volume currently has a size of 8 GiB. You now increase the size of this disk.

52.Change the Size (GiB) to 10

53.Choose Modify.

54.To confirm and increase the size of the volume, in the Modify pop-up window, choose Modify

Start the resized instance

You now start the instance again, which now has less memory but more disk space.

55.In left navigation pane, choose Instances. Next to your Web-Server, select the check box.

56.From the Instance state dropdown menu, choose Start instance.

Task 5: Exploring EC2 limits

Amazon EC2 provides different resources that you can use. These resources include images, instances, volumes, and snapshots. When you create an AWS account, there are default limits on these resources on a per-Region basis.

57.In the left navigation pane, choose Limits.

Note: There is a limit on the number of instances that you can launch in this Region. When launching an instance, the request must not cause your usage to exceed the current instance limit in that Region.

You can request an increase for many of these limits.

Task 6: Testing termination protection

You can delete your instance when you no longer need it. This is referred to as terminating your instance. You cannot connect to or restart an instance after it has been terminated.

In this task, you learn how to use termination protection.

58.In left navigation pane, choose Instances. Select the check box for your Web-Server.

59.At the top of the page in the Instance state dropdown menu, choose Terminate instance. From the Terminate instance? pop-up window, choose Terminate.

Note: At the top of the page, a message says Failed to terminate an instance: The instance 'ixxxxxxxxxxxx'
may not be terminated. Modify its 'disableApiTermination' instance attribute
and try again. This message is a safeguard to prevent the accidental termination of an instance.
If you really want to terminate the instance, you need to turn off the termination protection.

6. **Results**: (Students can paste their screen shots of the every task.)

7. **Conclusion**: We have created a EC2 instance and also configured it.

Experiment 6

Aim: Develop a static website and store in AWS S3 bucket using version control.

11. Objectives:

• Create a bucket in Amazon S3

• Add an object to a bucket

• Manage access permissions on an object and a bucket

• Create a bucket policy

 Use bucket versioning

3 Outcomes:

The learner will be able to **c**reate a bucket in Amazon S3. And upload objects.

4 **Hardware / Software Required:** Internet, AWS console

5 **Theory:**

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and from all industries can use it to store and protect any amount of data for a range of use cases, such as websites, mobile applications, backup and restore, archive, enterprise applications, Internet of Things (IoT) devices, and big data analytics. Amazon S3 provides easy-to-use management features so you can organize your data and configure finely tuned access controls to meet your specific business, organizational, and compliance requirements. Amazon S3 is designed for 99.999999999 percent (11 9's) of durability and stores data for millions of applications for companies all around the world.

Steps:-

Task 1: Creating a bucket

1.You are new to Amazon S3 and want to test the features and security of Amazon S3 as you configure the environment to hold the Amazon Elastic Compute Cloud (Amazon EC2) report data.

2.You know that every object in Amazon S3 is stored in a bucket, so creating a new bucket to hold the reports is the first thing on your task list.

3.In this task, you create a bucket to hold your Amazon EC2 report data .

4.Then examine the different bucket configuration options.

5.At the upper left of the AWS Management Console, on the Services menu, choose S3.

6.Choose Create bucket. Bucket names must be 3–63 characters long and consist of only lowercase letters, numbers, or hyphens. The bucket name must be globally unique across all of Amazon S3 regardless of account or Region, and you cannot change a bucket name after creating the bucket. As you enter a bucket name, a help box displays showing any violations of the naming rules. Refer to the Amazon S3 bucket naming rules in the Additional resources section at the end of the lab for more information.

7.In the General configuration section, enter the following as the Bucket name: report bucket(NUMBER)n the bucket name, replace (NUMBER) with a random number so that your bucket has a unique name.

•**Example bucket name: reportbucket98798**718

Leave Region at its default value.

By selecting a particular Region, you can optimize latency, minimize costs, or address regulatory requirements. Objects stored in a Region never leave that Region unless you explicitly transfer them to another Region.

8.Choose Create bucket

Task 2: Uploading an object to the bucket

Now that you have created a bucket for your report data, you are ready to work with objects.

An object can be any kind of file: a text file, a photo, a video, a .zip file, and so on. When you add an object to Amazon S3, you have the option to include metadata with the object and set permissions to control access to the object.

In this task, you test uploading objects to your reportbucket. You have a screen capture of a daily report and want to upload this image to your S3 bucket.

9.Right-click the following link: new-report.png. Choose Save link as, and save the file to your desktop.

10.In the S3 Management Console, find and select the bucket name that starts with reportbucket.

11.Choose Upload

This step launches an upload wizard. Use this wizard to upload files either by selecting them from a file chooser or by dragging them to the Amazon S3 window.

12.Choose Add files

13.Browse to and select the new-report.png file that you downloaded previously.

14.At the bottom of the page, choose Upload

Your file is successfully uploaded when the green bar indicating Upload succeeded appears.

15.In the Upload: status section in the upper right, choose Close

Task 3: Making an object public

Security is a priority in Amazon S3. Before you configure your EC2 instance to connect to the reportbucket, you want to test the bucket and object settings for security.

In this task, you configure permissions on your bucket and your object to test accessibility.

First, you attempt to access the object to confirm that it is private by default.

16.In the reportbucket overview page, on the Objects tab, locate the new-report.png object, and choose the new-report.png file name.

The new-report.png overview page opens. The navigation in the upper left updates with a link to return to the bucket overview page.

17.In the Object overview section, locate and copy the Object URL link.

The link should look similar to the following: https://reportbucket987987.s3-us-west-2.amazonaws.com/new-report.png

18.Open a new browser tab and paste the object URL link into the address field, and then press Enter.

You receive an Access Denied error because objects in Amazon S3 are private by default.

Now that you've confirmed that the default security of Amazon S3 is private, you test how to make the object publicly accessible.

19.Keep the browser with the Access Denied error open, and return to the web browser tab with the S3 Management Console.

20.You should still be on the new-report.png Object overview tab.

21.In the upper right, choose the Object actions dropdown menu, you will notice that Make public via ACL is greyed out.

22.In the upper left of the page, choose the reportbucket name in the navigation to go back to the main reportbucket overview page.

23.Choose the Permissions tab.

24.We need to allow the use of ACLs first. Under Object Ownership choose Edit.

25.Choose ACLs enabled.

26.Choose Bucket owner preferred.

27.Choose the check box next to I acknowledge that ACLs will be restored.

28.Choose Save Changes

29.Under Block public access (bucket settings), choose Edit to change the settings.

30.Clear the check box for the Block all public access option, and then leave all other options cleared.

Notice that all of the individual options remain cleared. When clearing the option for all public access, you must then select the individual options that apply to your situation and security objectives. You use access control lists (ACLs) and bucket policies later in the lab, so these options remain cleared in this task. In a production environment, it is recommended to use the least permissive settings possible. Refer to the Amazon S3 block public access link in the Additional resources section at the end of the lab for more information.

31.Choose Save changes

32.A dialogue box opens asking you to confirm your changes. Enter confirm in the field, and then choose Confirm

A message that says Successfully edited Block Public Access settings for this bucket. displays at the top of the window.

33.Choose the Objects tab.

34.Choose the new-report.png file name.

35.At the upper right on the new-report.png overview page, choose the Object actions dropdown menu, and select Make public.

Notice the warning: When public read access is enabled and not blocked by Block Public Access settings, anyone in the world can access the specified objects. This warning reminds you that if you make the object public, then everyone in the world will be able to read the object.

36.Choose Make public and you should see the green banner Successfully edited public access at the top of the window.

37.In the upper right, choose Close to return to the new-report.png object overview.

38.Return to the browser tab that displayed Access Denied for the new-report.png object, and refresh the page.

The new-report.png object now displays properly because it is publicly accessible.

39.Close the web browser tab that displays your new-report.png image, and return to the tab with the Amazon S3 Management Console.

In this example, you granted read access to just one specific object. If you would like to grant access to the entire bucket, you need to use a bucket policy, which this lab covers later.

In the next task, you work with your EC2 instance to confirm connectivity to the S3 bucket.

Task 4: Testing connectivity from the EC2 instance

In this task, you connect to your EC2 instance to test connectivity and security to the Amazon S3 reportbucket.

You should already be signed in to the AWS Management Console. If not, follow the steps in the Start Lab section to sign in to the AWS Management Console.

40.On the Services menu, choose EC2.

41.On the EC2 Dashboard, under the Resources section, choose Instances (running).

42.Select the check box for Bastion Host and choose Connect

43.In the Connect to instance window, select the Session Manager tab for the connection method.

With AWS Systems Manager Session Manager, you can connect to the bastion host instance without the need for specific ports to be open on your firewall or Amazon Virtual Private Cloud (Amazon VPC) security group. Refer to AWS Systems Manager Session Manager in the Additional resources section at the end of this lab for more information.

44.Choose Connect

A new browser tab or window opens with a connection to the bastion host instance.

You are now connected to the EC2 instance that holds the reporting application. Because Session Manager uses HTTPS port 443, it does not require you to open SSH port 22 to the outside world. You are satisfied with this security feature. Now you want to see how EC2 interacts with your S3 bucket.

45.In the bastion host session, enter the following command to change to the home directory (/home/ssm-user/):cd ~

The output returns you to the command prompt.

46.Enter the following command to verify that you are in the home directory:

```
pwd
```

The output should be as follows:

```
/home/ssm-user
```

You are now in the ssm-user's home directory where you will run all of the commands in this lab.

47.Enter the following command to list all of your S3 buckets.

```
aws s3 ls
```

The output should look similar to the following:

```
2020-11-11 22:34:46 reportbucket987987
```

You see the reportbucket you created and lab auto-generated buckets.

Note: During the creation of the lab environment, both an instance profile (which defines who you are for authentication) and a role (which defines what you can do after you authenticate) have been automatically added for the EC2 instance to allow the EC2 instance to list the S3 buckets and objects.

48.In the following command, change (NUMBER) at the end of the reportbucket name to the name of the bucket you created. Enter your adjusted command to list all the objects in your reportbucket.

```
aws s3 ls s3://reportbucket(NUMBER)
```

The command looks similar to the following: aws s3 ls s3://reportbucket987987

The output should look like the following:

```
2020-11-11 15:46:34 86065 new-report.png
```

There is currently only one object in your bucket. The object is called new-report.png.

49.Enter the following command to change directories into the reports directory.

```
cd reports
```

The output returns you to the command prompt.

50.Enter the following command to list the contents of the directory.

```
ls
```

The output shows some files created in your reports directory to test the application.

```
dolphins.jpg files.zip report-test.txt report-test1.txt report-test2.txt report-test3.txt whale.jpg
```

51.In the following command, change (NUMBER) at the end of the reportbucket name to the name of the bucket you created. Enter your adjusted command to see if you can copy a file to the S3 bucket.

```
aws s3 cp report-test1.txt s3://reportbucket(NUMBER)
```

The command looks similar to this: aws s3 cp report-test1.txt s3://reportbucket987987

The output indicates an upload failed error. This error occurs because you have read-only rights to the bucket and do not have the permissions to perform the PutObject action.

52.Leave this window open. and go back to browser tab with the AWS console.

In the next task, you create a bucket policy to add the PutObject permission.

Task 5: Creating a bucket policy

A bucket policy is a set of permissions associated with an S3 bucket. It is used to control access to an entire bucket or to specific directories within a bucket.

In this task, you use the AWS Policy Generator to create a bucket policy to enable read and write access from the EC2 instance to the bucket so that your reporting application can successfully write to Amazon S3.

53.Right-click the following link: sample-file.txt. Choose Save link as, and save the file to your desktop.

54.Return to the AWS Management Console, go to the Services menu, and select S3.

55.In the S3 Management Console tab, select the name of your bucket.

56.To upload the sample-file.txt file, choose Upload and use the same upload process that you used in task 2.

57.On the reportbucket overview page, choose the sample-file.txt file name. The samplefile. txt overview page opens.

58.Under the Object overview section, locate and copy the Object URL link.

59.In a new browser tab, paste the link into the address field, and then press Enter.

Once again, your browser displays an Access Denied message. You need to configure a bucket policy to grant access to all objects in the bucket without having to specify permissions on each object individually.

60.Keep this browser tab open, but return to the tab with the S3 Management Console.

61.Select Services and select IAM. In the left navigation, choose Roles.

62.In the Search field, enter EC2InstanceProfileRole

22

This is the role that the EC2 instance uses to connect to Amazon S3.

63.Select EC2InstanceProfileRole. In the Summary section, copy the Role ARN to a text file to use in a later step.

It should look similar to the following:

arn:aws:iam::596123517671:role/EC2InstanceProfileRole

64.Choose Services and S3, and return to the S3 Management Console.

65.Choose the reportbucket.

You should see the two objects you uploaded. If not, navigate back to your bucket so that you see the list of objects you have uploaded.

66.Choose the Permissions tab.

67.In the Permissions tab, scroll to the Bucket policy section, and choose Edit

A blank Bucket policy editor displays. You can create bucket policies manually, or you can create them with the assistance of the AWS Policy Generator.

Amazon Resource Names (ARNs) uniquely identify AWS resources across all of AWS. A colon (:) separates each section of the ARN, and each section represents a specific piece of the path to the specified resource. The sections can vary slightly depending on the service being

referenced but generally follow the format:
arn:partition:service:region:account-id:resource
Amazon S3 does not require Region or account-id parameters in ARNs, so those sections are left blank. However, the colon (:) to separate the sections is still used, so it looks similar to
arn:aws:s3:::reportbucket987987
Refer to the Amazon Resource Names (ARNs) and AWS Service Namespaces documentation link in the Additional resources section at the end of the lab for more information.
68.Below the Policy examples and Policy generator buttons, find the Bucket ARN. Copy the Bucket ARN to a text file to use in a later step.
It looks like the following:
Bucket ARN
arn:aws:s3:::reportbucket987987
69.Choose Policy generator
A new web browser tab opens with the AWS Policy Generator.
AWS policies use the JSON format and are used to configure granular permissions for AWS services. You can manually write the policy in JSON, or you can use the AWS Policy Generator to create the policy with a user-friendly web interface.
In the AWS Policy Generator window, configure the following options:
• For Select Type of Policy, select S3 Bucket Policy.
• For Effect, select Allow.
• For Principal, paste the EC2 Role ARN that you copied to a text file in a previous step.
• For AWS Service, keep the default setting of Amazon S3.
• For Actions, select GetObject and PutObject.
The GetObject action grants permission for objects to be retrieved from Amazon S3. Refer to the Additional resources section at the end of the lab for links to more information about the actions available for use in Amazon S3 policies.
• For Amazon Resource Name (ARN), enter *
70.Choose Add Statement. The details of the statement you configured are added to a table below the button. You can add multiple statements to a policy.
23
71.Choose Generate Policy.
A new window displays the generated policy in JSON format. It should look similar to the following:

```
{
"Version": "2012-10-17",
"Id": "Policy1604361694227",
"Statement": [
{
"Sid": "Stmt1604361692117",
"Effect": "Allow",
"Principal": {
"AWS": "arn:aws:iam::416159072693:role/EC2InstanceProfileRole"
},
"Action": [
"s3:GetObject",
"s3:PutObject"
],
"Resource": "*"
}
]
}
```

72.Copy the policy that you created to your clipboard.

73.Close the web browser tab, and return to the S3 Management Console tab with the Bucket policy editor.

74.Paste the bucket policy that you created into the Bucket policy editor.

75.In the Bucket policy editor, update the Resource value replacing * with the Bucket ARN you saved earlier followed by /*:

The updated Resource line in the lab policy should be similar to the following example:

```
"Resource": "arn:aws:s3:::reportbucket987987/*"
```

Confirm that /* appears after your bucket name as the Resource line in this sample shows.

76.Choose Save changes.

77.Return to the AWS Systems Manager (Systems Manager) window. If your session has timed out, reconnect to Systems Manager using the previous steps in the lab.

78.Enter the following command to verify that you are in the /home/ssm-user/reports directory.

```
pwd
```

The output should be as follows:

```
/home/ssm-user/reports
```

79.In the command below, replace (NUMBER) with the number you used to create your bucket. Enter your adjusted command to list all objects in your reportbucket.

```
aws s3 ls s3://reportbucket(NUMBER)
```

24

The command should look similar to the following: aws s3 ls s3://reportbucket987987

The output should look similar to the following:

```
sh-4.2$ aws s3 ls s3://reportbucket987987
2020-11-02 23:20:27 86065 new-report.png
2020-11-02 23:57:03 90 sample-file.txt
```

80.Enter the following command to list the contents of the reports directory.

```
ls
```

The output returns a list of files.

81.In the command below, replace (NUMBER) with the number you used to create your bucket. Enter your adjusted command to try copying the report-test1.txt file to the S3 bucket.

```
aws s3 cp report-test1.txt s3://reportbucket(NUMBER)
```

The command should look like the following: aws s3 cp report-test1.txt s3://reportbucket987987

The output returns the following:

```
upload: ./report-test1.txt to s3://reportbucket987987/report-test1.txt
```

82.In the command below, replace (NUMBER) with the number you used to create your bucket. Enter your adjusted command to see if the file successfully uploaded to Amazon S3.

```
aws s3 ls s3://reportbucket(NUMBER)
```

The output should look similar to the following:

```
2020-11-11 18:20:23 86065 new-report.png
2020-11-11 18:32:18 31 report-test1.txt
2020-11-11 18:20:22 90 sample-file.txt
```

You have successfully uploaded (PutObject) a file from the EC2 instance to your S3 bucket.

83. In the command below, replace (NUMBER) with the number you used to create your bucket. Enter your adjusted command to retrieve (GetObject) a file from Amazon S3 to the EC2 instance.

```
aws s3 cp s3://reportbucket(NUMBER)/sample-file.txt sample-file.txt
```

The output should look similar to the following:

```
download: s3://reportbucket987987/sample-file.txt to ./sample-file.txt
```

84.Enter the following command to see if the file is now in the /reports directory.

```
ls
```

The output should look similar to the following:

```
dolphins.jpg files.zip report-test1.txt report-test2.txt report-test3.txt sample-file.txt
```

You now see the sample-file.txt in your file list. Congratulations! You have successfully uploaded and retrieved a file from Amazon EC2 to the S3 bucket.

85.Return to the browser tab that displayed the Access Denied error for the sample-file.txt, and refresh the page.

The page still displays an error message because the bucket policy gave rights to only the principal called EC2InstanceProfileRole.

86.Go to the AWS Policy Generator, and add another statement to the bucket policy allowing everyone (*) read access (GetObject). Take a moment to generate this policy. This policy allows the EC2InstanceProfileRole to have access to the bucket while giving everyone access to read the objects via the browser.

Below is an expample of the above:

```
{
"Sid": "Stmt1604428842806",
"Effect": "Allow",
"Principal": "*",
"Action": "s3:GetObject",
"Resource": "arn:aws:s3:::reportbucket987987/*"
}
```

87.To test if your policy works, go to your browser with the Access Denied error and refresh it. If you can read the text, then congratulations! Your policy was successful.

If not, look at the following policy for help. The modified policy should look like the following policy. Notice that there are two statements: one with the EC2InstanceProfileRole and one where the principal is "*" for everyone.

If you had trouble generating the policy on your own, you can copy the policy below and paste it into the BucketPolicy Editor. Remember to replace the existing EC2InstanceProfileRole ARN in the policy below with the EC2InstanceProfileRole ARN you copied in a previous step. Ensure that you replace the reportbucket example ARN with the bucket you created and the /* appears at the end of the Bucket ARN. See the last line of the policy as an example.

```
{ "Version": "2012-10-17",
"Id": "Policy1604428844058",
"Statement": [
{
"Sid": "Stmt1604428821481",
"Effect": "Allow",
"Principal": {
"AWS": "arn:aws:iam::285058481724:role/EC2InstanceProfileRole"
},
"Action": [
"s3:GetObject",
"s3:PutObject"
],
"Resource": "arn:aws:s3:::reportbucket987987/*"
},
{
"Sid": "Stmt1604428842806",
"Effect": "Allow",
"Principal": "*",
```

```
"Action": "s3:GetObject",
"Resource": "arn:aws:s3:::reportbucket987987/*"
} ]}
```

88.Leave the tab open with the sample-file.txt displayed. You return to this tab in the next task.

In this task, you created a bucket policy to allow specific access rights to your bucket. In the next section, you explore how to keep copies of files to prevent against accidental deletion.

Task 6: Exploring versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures.

For auditing and compliance reasons, you need to enable versioning on your reportbucket. Versioning should protect the reports in the reportbucket against accidental deletion. You are curious to see if this works as advertised. In this task, you enable versioning and test the feature by uploading a modified version of the sample-file.txt file from the previous task.

89.You should be on the S3 bucket Permissions tab from the previous task. If you are not, choose the link to the bucket at the upper left of the screen to return to the bucket overview page.

90.On the reportbucket overview page, choose the Properties tab.

91.Under the Bucket Versioning section, click Edit select Enable then click Save changes.

Versioning is enabled for an entire bucket and all objects within the bucket. It cannot be enabled for individual objects.

There are also cost considerations when enabling versioning. Refer to the Additional resources section at the end of the lab for links to more information.

92.Right-click this link, and save the text file to your computer using the same name as the text file in the previous task: sample-file.txt

Although this file has the same name as the previous file, it contains new text.

93.In the Amazon S3 Management Console, on the reportbucket, choose the Objects tab.

Under the Objects section, find Show versions.

94.Choose Upload and use the same upload process that you used in tasks 2 and 5 to upload the new sample-file.txt file.

95.Go to the browser tab that has the contents of the sample-file.txt file.

96.Make a note of the contents on the page, and then refresh the page.

Notice that new lines of text appear.

If a version is not otherwise specified, Amazon S3 always returns the latest version of an object.

You can also obtain a list of available versions in the Amazon S3 Management Console.

97.Close the web browser tab with the contents of the text file.

98.In the Amazon S3 Management Console, choose the sample-file.txt file name. The samplefile.

txt overview page opens.

99.Choose the Versions tab, and then select the check box for the bottom version, which reads null. (This is not the latest version.)

100.Click Open.

27

You should now see the original version of the file using the Amazon S3 Management Console.

However, if you try to access the older version of the sample-file.txt file using the object URL link, you will receive an access denied message. This message is expected because the bucket policy you created in the previous task allows permission to access only the latest version of the object. In order to access a previous version of the object, you need to update your bucket policy to include the s3:GetObjectVersion permission. The following bucket policy example includes the additional s3:GetObjectVersion action that allows you to access the older version using the link. You do not need to update your bucket policy with this example to complete this lab. You can try to do this on your own after you complete the task.

```
{
"Id": "Policy1557511288767",
"Version": "2012-10-17",
"Statement": [
{
"Sid": "Stmt1557511286634",
"Action": [
"s3:GetObject",
"s3:GetObjectVersion"
],
"Effect": "Allow",
"Resource": "arn:aws:s3:::reportbucket987987/*",
"Principal": "*"
}
]
}
```

101.Return to the AWS Management Console tab, and choose the link for the bucket name at the upper left to return to the bucket overview tab.

102.Locate the Show versions option, and toggle the button to on to show the versions.

Now you can view the available versions of each object and identify which version is the latest. Notice that the new-report.png object has only one version. The version ID is null because the object was uploaded before versioning was enabled on this bucket.

Also notice that you can now choose the version name link to navigate directly to that version of the object in the console.

103.Next to Show versions, toggle the button to off to return to the default object view.

104.Select the check box to the left of the sample-file.txt.

105.With the object selected, choose Delete

106.The Delete objects page appears.

107.At the bottom, in the Delete objects? section, enter delete and choose the Delete objects button to confirm deletion of the object.

108.In the upper right of the page, choose Close to return to the bucket overview.

The sample-file.txt object is no longer displayed in the bucket. However, if the object is deleted by mistake, you can use versioning to recover it.

109.Locate the Show versions option, and toggle the button to on to show the versions.

Notice that the sample-file.txt object is displayed again, but the most recent version is a Delete marker. The two previous versions are also listed. If versioning has been enabled on the bucket, objects are not immediately deleted. Instead, Amazon S3 inserts a delete marker, which becomes the current object version. The previous versions of the object are not removed. Refer to the Additional resources section at the end of the lab for links to more information about versioning.

110.Select the check box for the version of the sample-file.txt object with the Delete marker.

111.With the object selected, choose Delete

112.The Delete objects window appears

113.At the bottom in the Permanently delete objects? section, enter permanently delete and choose the Delete objects button to confirm deletion of the object.

114.On the upper right of the page, choose Close to return to the bucket overview.

115.Next to Show versions, toggle the button to off to return to the default object view.

Notice that the sample-file.txt object has been restored to the bucket. Removing the delete marker has effectively restored the object to its previous state. Refer to the Additional resources section at the end of the lab for links to more information about undeleting S3 objects.

Next, you delete a specific version of the object.

116.To delete a specific version of the object, locate the Show versions option, and toggle the button to on to show the versions.

You should see two versions of the sample-file.txt object.

117.Select the check box for the latest version of the sample-file.txt object.

118.With the object selected, choose Delete

119.The Delete objects window appears.

120.At the bottom in the Permanently delete objects? section, enter permanently delete and choose the Delete objects button.

121.On the upper right of the page, choose Close to return to the bucket overview.

Notice that there is now only one version of the sample-file.txt file. When deleting a specific version of an object, no delete marker is created. The object is permanently deleted. Refer to the Additional resources section at the end of the lab for links to more information about deleting object versions in Amazon S3.

122.Next to Show versions, toggle the button to off to return to the default object view.

123.Choose the sample-file.txt file name. The sample-file.txt overview page opens

124.Copy the Object URL link displayed at the bottom of the window.

125.In a new browser tab, paste the link into the address field, and then press Enter.
The browser page displays the text of the original version of the sample-file.txt object.

10. Result: (Students can paste their screen shots of the every task.)

11. Conclusion :

You have successfully created an S3 bucket to use to store report data from your EC2 instance. You created a bucket policy so that the EC2 instance can Put Objects and Get Object from the report bucket, and you successfully tested uploading and downloading files from the EC2 instance to test the bucket policy.

Experiment No. 7
Create Mysql database using amazon RDS

Hardware / Software Required: Internet, AWS console

Theory:
Amazon Relational Database Service (RDS) is a managed SQL database service provided by Amazon Web Services (AWS). Amazon RDS supports an array of database engines to store and organize data. It also helps with relational database management tasks, such as data migration, backup, recovery and patching.
Amazon RDS facilitates the deployment and maintenance of relational databases in the cloud. A cloud administrator uses Amazon RDS to set up, operate, manage and scale a relational instance of a cloud database. Amazon RDS is not itself a database; it is a service used to manage relational databases.
Task 1: Creating an Amazon RDS database
In this task, you create a MySQL database in your virtual private cloud (VPC). MySQL is a popular open-source relational database management system (RDBMS), so there are no software licensing fees.
1. On the **Services** menu, choose **RDS**.
2. Choose **Create database**
3. Under **Engine options**, select **MySQL**.
4. In the **Templates** section, select **Dev/Test**
5. In the **Settings** section, configure the following options:
DB instance identifier: inventory-db
Master username:
Master password:
Confirm password:
6. In the **DB instance class** section, configure the following options:
Select **Burstable classes (includes t classes)**.
Select **db.t3.micro**.
7. In the **Storage** section, for **Storage type**, select **General Purpose SSD (gp2)**.
8. In the **Connectivity** section, configure the following option:
Virtual private cloud (VPC): Lab VPC
9. In the **Connectivity** section, for **Existing VPC security groups**, choose the **X** on default
to remove this security group. Then choose the dropdown list, and select ***DB-SG*** *to add*
it.

10. Scroll to the **Additional configuration** section, and choose to expand it. Configure the following settings:

For **Initial database name**, enter

Clear (turn off) the **Enable Enhanced monitoring** option.

*This is the logical name of the database that the application will use At the bottom of the page, choose **Create database***

You should receive this message: **Creating database inventory-db**.

 Before you continue to the next task, the database instance status must Be **Available**. This process could take several minutes.

Task 2: Configuring web application communication with a database instance

This lab automatically deployed an Amazon Elastic Compute Cloud (Amazon EC2) instance with a running web application. You must use the IP address of the instance to connect to the application.

*11. On the **Services** menu, choose **EC2**.*

*12. In the left navigation pane, choose **Instances**.*

*13. In the center pane, there should be a running instance that is named **App Server**.*

*14. Select the check box for the **App Server** instance.*

*15. In the **Details** tab, copy the **Public IPv4 address** to your clipboard.*

***Tip:** If you hover over the IP address, a copy icon appears. To copy the displayed value, choose the*
icon.

16. Open a new web browser tab, paste the IP address into the address bar, and then press Enter.

The web application should appear. It does not display much information because the application is not yet connected to the database.

*17. Choose **Settings**.*

You can now configure the application to use the Amazon RDS database instance that you created earlier. You first retrieve the database endpoint so that the application knows how to connect to a database.

18. Return to the AWS Management Console, but do not close the application tab. (You will return to it soon.)

*19. On the **Services** menu, choose **RDS**.*

*20. In the left navigation pane, choose **Databases**.*

21. Choose inventory-db.

22. Scroll to the Connectivity & security section, and copy the Endpoint to your clipboard.

It should look similar to this example: inventory-db.crwxbgqad61a.rds.amazonaws.com

23. Return to the browser tab with the inventory application, and enter the following values:

*For **Endpoint**, paste the endpoint you copied earlier.*

*For **Database**, enter i*

*For **Username**, enter*

For ***Password****, enter*
Choose ***Save****.*
The application will now connect to the database, load some initial data, and display information.
24. You can use the web application to Add inventory, edit, and delete inventory information.
25. Insert new records into the table. Ensure that the table has 5 or more inventory records before submitting your work.
You have now successfully launched the application and connected it to the database.

6. Conclusion : Hense successfully created Mysql database using amazon RDS.

Experiment No. 8
To study AWS Identity and Access Management (IAM)

Hardware / Software Required: Internet, AWS console

Theory:

AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform.

IAM provides the infrastructure necessary to control authentication and authorization for your account. The IAM infrastructure includes the following elements:

- **Terms :** Learn more about IAM terms.
- **IAM Resources :** The user, group, role, policy, and identity provider objects that are stored in IAM. As with other AWS services, you can add, edit, and remove resources from IAM.
- **IAM Identities:** The IAM resource objects that are used to identify and group. You can attach a policy to an IAM identity. These include users, groups, and roles.
- **IAM Entities :** The IAM resource objects that AWS uses for authentication. These include IAM users and roles.
- **Principals :** A person or application that uses the AWS account root user, an IAM user, or an IAM role to sign in and make requests to AWS. The principal is authenticated as the AWS account root user or an IAM entity to make requests to AWS. As a best practice, do not use your root user credentials for your daily work. Instead, create IAM entities (users and roles). You can also support federated users or programmatic access to allow an application to access your AWS account.Principals include federated users and assumed roles.
- **Request** : When a principal tries to use the AWS Management Console, the AWS API, or the AWS CLI, that principal sends a *request* to AWS. The request includes the following information:
- **Actions or operations** – The actions or operations that the principal wants to perform. This can be an action in the AWS Management Console, or an operation in the AWS CLI or AWS API.
- **Resources** – The AWS resource object upon which the actions or operations are performed.
- **Principal** – The person or application that used an entity (user or role) to send the request. Information about the principal includes the policies that are associated with the entity that the principal used to sign in.
- **Environment data** – Information about the IP address, user agent, SSL enabled status, or the time of day.
- **Resource data** – Data related to the resource that is being requested. This can include information such as a DynamoDB table name or a tag on an Amazon EC2 instance.

- **Authentication :** A principal must be authenticated (signed in to AWS) using their credentials to send a request to AWS. Some services, such as Amazon S3 and AWS STS, allow a few requests from anonymous users. However, they are the exception to the rule.

To authenticate from the console as a root user, you must sign in with your email address and password. As an IAM user, provide your account ID or alias, and then your user name and password. To authenticate from the API or AWS CLI, you must provide your access key and secret key. You might also be required to provide additional security information. For example, AWS recommends that you use multi- factor authentication (MFA) to increase the security of your account. To learn more about the IAM entities that AWS can authenticate, see IAM users and IAM roles.

- **Authorization :** You must also be authorized (allowed) to complete your request. During authorization, AWS uses values from the request context to check for policies that apply to the request. It then uses the policies to determine whether to allow or deny the request. Most policies are stored in AWS as JSON documents and specify the permissions for principal entities. There are several types of policies that can affect whether a request is authorized. To provide your users with permissions to access the AWS resources in their own account, you need only identity-based policies. Resource-based policies are popular for granting cross- account access. The other policy types are advanced features and should be used carefully.

AWS checks each policy that applies to the context of your request. If a single permissions policy includes a denied action, AWS denies the entire request and stops evaluating. This is called an explicit deny. Because requests are denied by default, AWS authorizes your request only if every part of your request is allowed by the applicable permissions policies. The evaluation logic for a request within a single account follows these general rules:

- By default, all requests are denied. (In general, requests made using the AWS account root user credentials for resources in the account are always allowed.)
- An explicit allow in any permissions policy (identity-based or resource-based) overrides this default.
- The existence of an Organizations SCP, IAM permissions boundary, or a session policy overrides the allow. If one or more of these policy types exists, they must all allow the request. Otherwise, it is implicitly denied.
- An explicit deny in any policy overrides any allows.
- **Actions or operations :** After your request has been authenticated and authorized, AWS approves the actions or operations in your request. Operations are defined by a service, and include things that you can do to a resource, such as viewing, creating, editing, and deleting that resource. For example, IAM supports approximately 40 actions for a user resource, including the following actions:
  - CreateUser
  - DeleteUser
  - GetUser
  - UpdateUser
- To allow a principal to perform an operation, you must include the necessary actions in a policy that applies to the principal or the affected resource.

- **Recourses :** After AWS approves the operations in your request, they can be performed on the related resources within your account. A resource is an object that exists within a service. Examples include an Amazon EC2 instance, an IAM user, and an Amazon S3 bucket. The service defines a set of actions that can be performed on each resource. If you create a request to perform an unrelated action on a resource, that request is denied. For example, if you request to delete an IAM role but provide an IAM group resource, the request fails.
- **Task1 : Explore the users and groups**

In this task, you will explore the users and groups that have already been created for you in IAM. First, note the Region that you are in; for example, N. Virginia. The Region is displayed in the upper-right corner of the console page. You might need this information later in the lab.

- user-1
- user-2

44

hoose the Services menu, locate the Security, Identity, & Compliance services, and choose IAM.

1. In the navigation pane on the left, choose Users.
- The following IAM users have been created for you:
- user-3

2. Choose the name of user-1.
- This brings you to a summary page for user-1. The Permissions tab will be displayed.
- Notice that user-1 does not have any permissions.

3. Choose the Groups tab. Notice that user- 1 also is not a member of any groups.
4. Choose the Security credentials tab. Notice that user-1 is assigned a Console password. This allows the user to access the AWS Management Console.
5. In the navigation pane on the left, choose User groups. The following groups have already been created for you:
- EC2-Admin
- EC2-Support
- S3-Support

6. Choose the name of the EC2-Support group. This brings you to the summary page for the EC2-Support group.
7. Choose the Permissions tab. This group has a managed policy called AmazonEC2ReadOnlyAccess associated with it. Managed policies are prebuilt policies (built either by AWS or by your administrators) that can be attached to IAM users and groups. When the policy is updated, the changes to the policy are immediately applied against all users and groups that are attached to the policy.
8. Under Policy Name, choose the link for the **AmazonEC2ReadOnlyAccess** policy.
9. **Choose the {} JSON tab**.
- A policy defines what actions are allowed or denied for specific AWS resources. This policy is granting permission to List and Describe (view) information about Amazon Elastic Compute Cloud (Amazon EC2), Elastic Load Balancing, Amazon CloudWatch, and Amazon EC2 Auto Scaling. This ability to view resources, but not modify them, is ideal for assigning to a support role.
- Statements in an IAM policy have the following basic structure:

 Effect says whether to Allow or Deny the permissions.

 Action specifies the API calls that can be made against an AWS service (for example, cloudwatch:ListMetrics).

 Resource defines the scope of entities covered by the policy rule (for example, a specific Amazon Simple Storage Service [Amazon S3] bucket or Amazon EC2 instance; an asterisk [*] means any resource).

10. In the navigation pane on the left, choose
User groups.

 Task 2: Add users to groups

11. Choose the name of the **S3-Support group**.

12. Choose the **Permissions** tab. The S3- Support group has the AmazonS3ReadOnlyAccess policy attached.

13. Under Policy Name, choose the link for the **AmazonS3ReadOnlyAccess** policy.

14. Choose the **{} JSON** tab. This policy has permissions to Get and List for all resources in Amazon S3.

45

15. In the navigation pane on the left, choose
User groups.

16. Choose the name of the **EC2-Admin**
group.

17. Choose the **Permissions** tab. This group is different from the other two. Instead of a managed policy, the group has an inline policy, which is a policy assigned to just one user or group. Inline policies are typically used to apply permissions for specific situations.

18. Under **Policy Name**, choose the name of the **EC2-Admin-Policy** policy.

19. Choose the **JSON** tab. This policy grants permission to Describe information about Amazon EC2 instances, and also the ability to Start and Stop instances.

20. At the bottom of the screen, choose Cancel to close the policy.

You have recently hired *user-1* into a role where they will provide support for Amazon S3. You will add them to the *S3-Support* group so that they inherit the necessary permissions via the attached *AmazonS3ReadOnlyAccess* policy.

Ignore any "not authorized" errors that appear during this task. They are caused by your lab account having limited permissions and will not impact your ability to complete the lab.

Add user-1 to the S3-Support group

21. In the left navigation pane, choose User groups.

22. Choose the name of the S3-Support group.

23. On the Users tab, choose Add users.

24. Select user-1, and choose Add users. On the Users tab, notice that *user-1* has been added to the group.

Add user-2 to the EC2-Support group : You have hired ***user-2*** into a role where they will provide support for Amazon EC2. You will add them to the *EC2-Support* group so that they inherit the necessary permissions via the attached *AmazonEC2ReadOnlyAccess* policy.

25. Use what you learned from the previous steps to add *user-2* to the *EC2-Support* group. *user-2* should now be part of the *EC2-Support* group.

Add user-3 to the EC2-Admin group : You have hired *user-3* as your Amazon EC2 administrator to manage your EC2 instances. You will add them to the *EC2-Admin* group so that they inherit the necessary permissions via the attached *EC2-Admin-Policy*.

26. Use what you learned from the previous steps to add *user-3* to the *EC2-Admin* group.
user-3 should now be part of the *EC2-Admin* group.
27. In the navigation pane on the left, choose **User groups**. Each group should have a 1 in the Users column. This indicates the number of users in each group.

Task 3: Sign in and test users :

In this task, you will test the permissions of each IAM user in the console.

Get the console sign-in URL

28. In the navigation pane on the left, choose
Dashboard.

46

Notice the **Sign-in URL for IAM users in this account** section at the top of the page. The signin
URL looks similar to the following:
https://123456789012.signin.aws.amazon.com/console This link can be used to sign in to the AWS account that you are currently using.
29. Copy the sign-in link to a text editor.

Test user-1 permissions

30. Open a private or incognito window in your browser.
31. Paste the sign-in link into the private browser, and press **ENTER.** You will now sign-in as *user-1*, who has been
o **IAM user name:** user-1
o **Password:** Lab-Password1
hired as your Amazon S3 storage support staff.
32. Sign in with the following credentials:
33. Choose the **Services** menu, and choose **S3**.
34. Choose the name of one of your buckets, and browse the contents.
Because this user is part of the *S3-Support* group in IAM, they have permissions to view a list of Amazon S3 buckets and their contents. Now, test whether the user has access to Amazon EC2.
Choose the **Services** menu, and choose **EC2**.
35. In the left navigation pane, choose **Instances**.
36. You cannot see any instances. Instead, an error message says *you are not authorized to perform this operation*. This user has not been assigned any permissions to use Amazon EC2. You will now sign in as *user-2*, who has been hired as your Amazon EC2 support person.
37. First, sign out *user-1* from the console:
o In the upper-right corner of the page, choose **user-1**.
o Choose **Sign Out**. **Test**
user-2 permissions
o **IAM user name:** user-2
o **Password:** Lab-Password2
38. Paste the sign-in link into the private browser again, and press ENTER.
39. Sign in with the following credentials:
40. Choose the **Services** menu, and choose **EC2**.
41. In the navigation pane on the left, choose **Instances**.
You are now able to see an EC2 instance. However, you cannot make any changes to Amazon

EC2 resources because you have read-only permissions. If you cannot see an EC2 instance, then your Region might be incorrect. In the upper-right corner of the page, choose the Region name, and then choose the Region that you were in at the beginning of the lab (for example, **N. Virginia**).

47

42. Select the **EC2 instance**.
43. Choose the **Instance state** menu, and then choose **Stop instance**.
44. To confirm that you want to stop the instance, choose **Stop**.

An error message appears and says that *You are not authorized to perform this operation*. This demonstrates that the policy only allows you to view information without making changes.

45. Next, check if *user-2* can access Amazon S3.
46. Choose the **Services** menu, and choose **S3**. An error message says *You don't have permissions to list buckets* because *user-2* does not have permissions to use Amazon S3.

You will now sign-in as *user-3,* who has been hired as your Amazon EC2 administrator.

47. First, sign out *user-2* from the console:

o In the upper-right corner of the page, choose **user-2**.

48. Choose **Sign Out**.

Test user-3 permissions

Paste the sign-in link into the private browser again, and press ENTER. Sign in with the following credentials:

IAM user name: user-3 **Password:** Lab-Password3

49. Choose the **Services** menu, and choose **EC2**.
50. In the navigation pane on the left, choose **Instances**.
51. An EC2 instance is listed. As an Amazon EC2 Administrator, this user should have permissions to *Stop* the EC2 instance.
52. If you cannot see an EC2 instance, then your Region might be incorrect. In the upper-right corner of the page, choose the Region name, and then choose the Region that you were in at the beginning of the lab (for example, **N. Virginia**).
53. Select the EC2 instance.
54. Choose the **Instance state** menu, and then choose **Stop instance**.
55. To confirm that you want to stop the instance, choose **Stop**.
56. This time, the action is successful because *user-3* has permissions to stop EC2 instances.

The **Instance state** changes to *Stopping* and starts to shut down.

57. Close your private browser window.

7. Conclusions : Hense AWS Identity and Access Management (IAM) has been studied