Introduction:

In this Lab, you will be deploying a Java Stack on the Cloud using the Grails Framework and Sample App.

Key Steps:

1. Install Java JDK 1.8 on a Free-Tier Amazon Linux AMI Instance
2. Install Tomcat 7 on AWS EC2 Instance
3. Install MySQL on same Grails Instance
4. Update Grails App DB Config and Deploy WAR App to Tomcat
5. Clean-Up (Stop and Terminate EC2 Instance)

Part I. Install Java 8 JDK and Tomcat 7

Step 1: Launch EC2 Free-Tier Instance

```
Type:                       m3.medium    (note that is not free-tier eligible!)
VPC:                        default
Subnet:                     public
Auto Assigned Public IP:    enabled
Create new SG:              grails
Open Ports:                 22, 80, 8080
Key Pair:                   your-key-pair
```

Step 2: SSH into EC2 Instance via Public IP

```
ssh -i <your-key-pair.pem> ec2-user@<your-host-ip>
```

Step 3: Install Java 8

```
sudo yum install java-1.8.0-openjdk-devel

** Select Java 8 Option for: **

sudo /usr/sbin/alternatives --config java
sudo /usr/sbin/alternatives --config javac

NOTE: JAVA_HOME = /usr/lib/jvm/java-1.8.0-openjdk.x86_64
```

Step 4: Install Tomcat 7

```
wget http://apache.mirrors.hoobly.com/tomcat/tomcat-7/v7.0.82/bin/apache-tomcat-7.0.82.tar.gz
tar zxpvf apache-tomcat-7.0.82.tar.gz
sudo mv apache-tomcat-7.0.82 /usr/share
```

To configure Tomcat to launch automatically create a file called “tomcat” in the directory /etc/rc.d/init.d/ with the following contents:

```
!/bin/sh
# Tomcat init script for Linux.
#
# chkconfig: 2345 96 14
# description: The Apache Tomcat servlet/JSP container.
JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk.x86_64
CATALINA_HOME=/usr/share/apache-tomcat-7.0.82
export JAVA_HOME CATALINA_HOME
exec $CATALINA_HOME/bin/catalina.sh $*
```

Set Init Script Permissions

```
sudo chmod 755 /etc/rc.d/init.d/tomcat
sudo chkconfig --level 2345 tomcat on
```

Manual Run of Tomcat

```
/etc/rc.d/init.d/tomcat start
/etc/rc.d/init.d/tomcat stop
```

Step 5: Config Tomcat Users

File: **/usr/share/apache-tomcat-7.0.82/conf/tomcat-users.xml**
Password: <Chose your Password>

```
<tomcat-users>
        <role rolename="manager-script"/>
        <role rolename="manager-jmx"/>
        <role rolename="manager-status"/>
        <role rolename="admin-gui"/>
        <role rolename="manager-gui"/>
        <user username="tomcat" password="**********"
          roles="manager-gui,manager-status,admin-gui"/>
        <user username="tomcattools" password="**********"/>
</tomcat-users>
```

Part II. Install MySQL, Create CMPE281 Database and Deploy Hello Grails Application

Step 6: Install MySQL on Same Tomcat EC2 Instance

REF: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-LAMP.html
NOTE: Only Install MySQL from Instructions Above.

```
sudo yum install -y mysql56-server
sudo service mysqld start
sudo chkconfig mysqld on

sudo mysql_secure_installation
```

Default root password = none (hit enter)
*Set root passwrd = ***** (choose your own)*
Remove Anonymous Users = Y
Disallow root Remote Logins = Y
Remove Test Databases = Y

```
Note:  sudo service mysqld stop  (to stop mysql)
```

Step 7: On MySQL Command Line, Create DB & Install Tables

REF: https://dev.mysql.com/doc/refman/5.6/en/mysql.html

```
mysql --user=user_name --password=your_password db_name

mysql --user=root --password
password: ***** (enter your password)

mysql> create database cmpe281 ;
mysql> use cmpe281;
mysql> show tables ;

CREATE TABLE gumball (
  id bigint(20) NOT NULL AUTO_INCREMENT,
  version bigint(20) NOT NULL,
  count_gumballs int(11) NOT NULL,
  model_number varchar(255) NOT NULL,
  serial_number varchar(255) NOT NULL,
  PRIMARY KEY (id),
  UNIQUE KEY serial_number (serial_number)
) ;

insert into gumball ( id, version, count_gumballs, model_number, serial_number )
values ( 1, 0, 1000, 'M102988', '1234998871109' ) ;

select * from gumball ;
```

Part II (Cont.) -- Update Grails Project & Deploy to Tomcat

Install SDK MAN (Locally)

Follow Instructions Here: http://sdkman.io/

```
curl -s "https://get.sdkman.io" | bash
source "$HOME/.sdkman/bin/sdkman-init.sh"
sdk version
```

Install Groovy & Grails

Note: assuming you already have Java JDK 7 or 8 Installed

```
sdk ls groovy
sdk ls grails
sdk install groovy 2.4.7
sdk install grails 3.2.11
sdk current

grails --version

| Grails Version: 3.2.5
| Groovy Version: 2.4.7
| JVM Version: 1.8.0_151
```

Config Grails Database Connection for Production

Update your **grails-app/conf/application.yml** file to connect to your MySQL DB.

Generate and Deploy Application WAR file

In your Grails Project Root Folder, Run Command:

```
grails war
```

Deploy Generated WAR file in:

```
build/libs (folder)
```

Note: To Deploy, SCP War file to EC2 Instance and Copy into Tomcat's "webapps" folder.

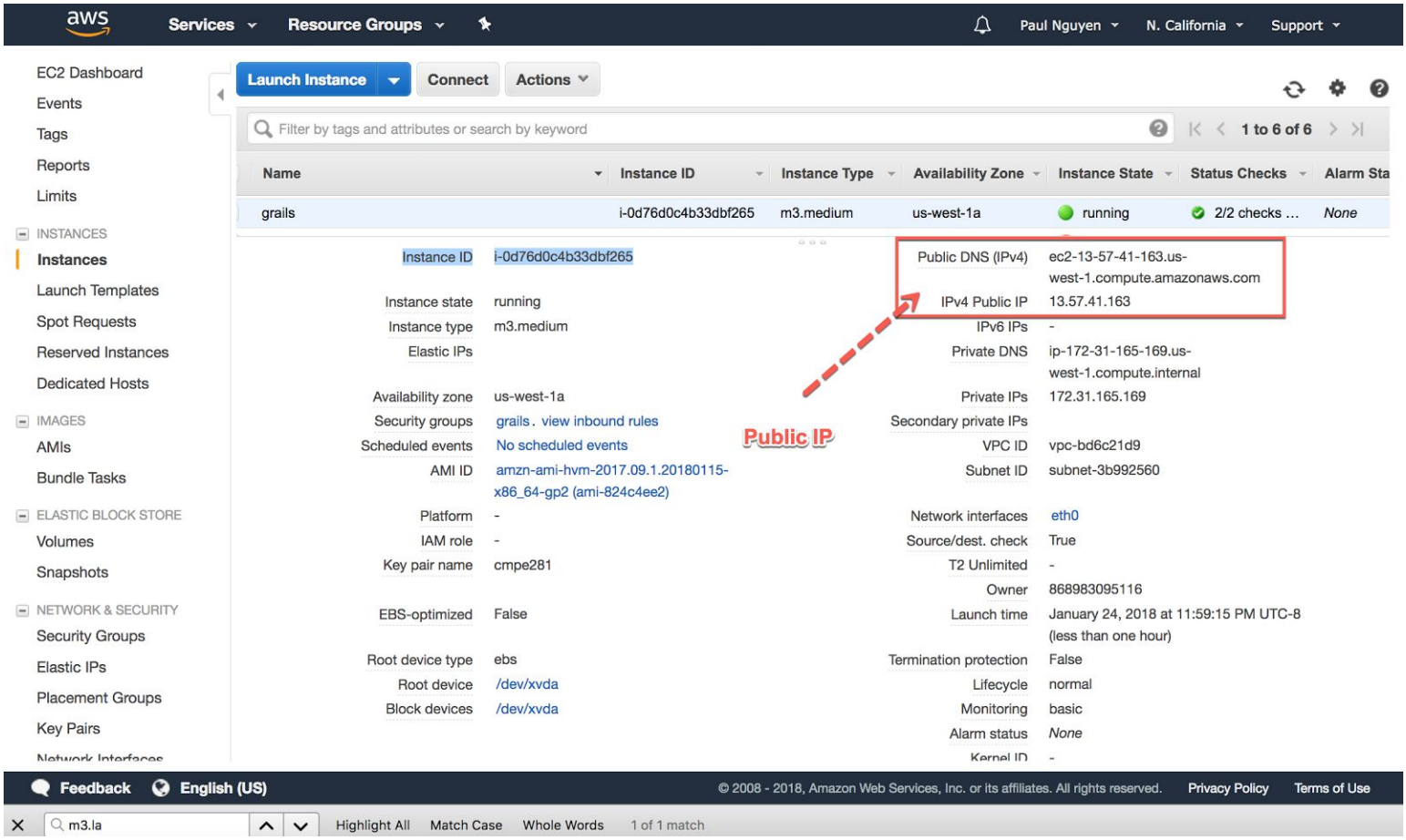
Copy (SCP) file "**gumball-v1-1.0.war**" to your EC2 Instance. For example:

```
scp -i .<your-key-pair.pem> gumball-v1.war ec2-user@<your-host-ip>:/tmp
```

This should be in the folder:

```
/usr/share/apache-tomcat-7.0.82/webapps
```

Grails App should now be running in your AWS EC2 Instance. (For Example)



Tomcat Web Application Manager

Message:	OK - Started application at context path /gumball-v1-1.0

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Applications

Path	Version	Display Name	Running	Sessions	Commands
/	*None specified*	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	*None specified*	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/examples	*None specified*	Servlet and JSP Examples	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/gumball-v1-1.0	*None specified*		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/host-manager	*None specified*	Tomcat Host Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	*None specified*	Tomcat Manager Application	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Grails App Running

Deploy

Deploy directory or WAR file located on server

Context Path (required):

XML Configuration file URL:

WAR or Directory URL:

Deploy

WAR file to deploy

Select WAR file to upload Browse... No file selected.

Deploy

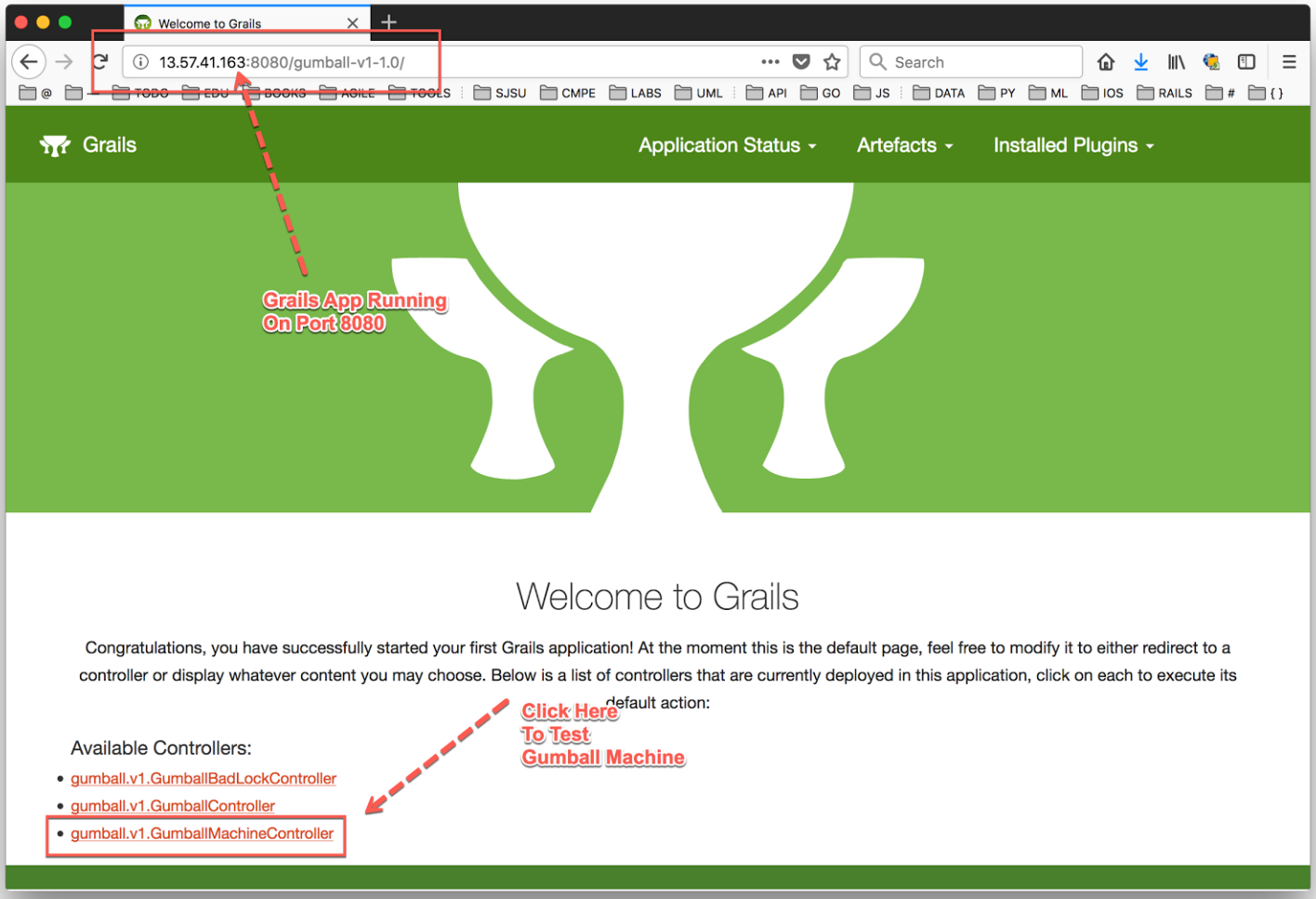
Diagnostics

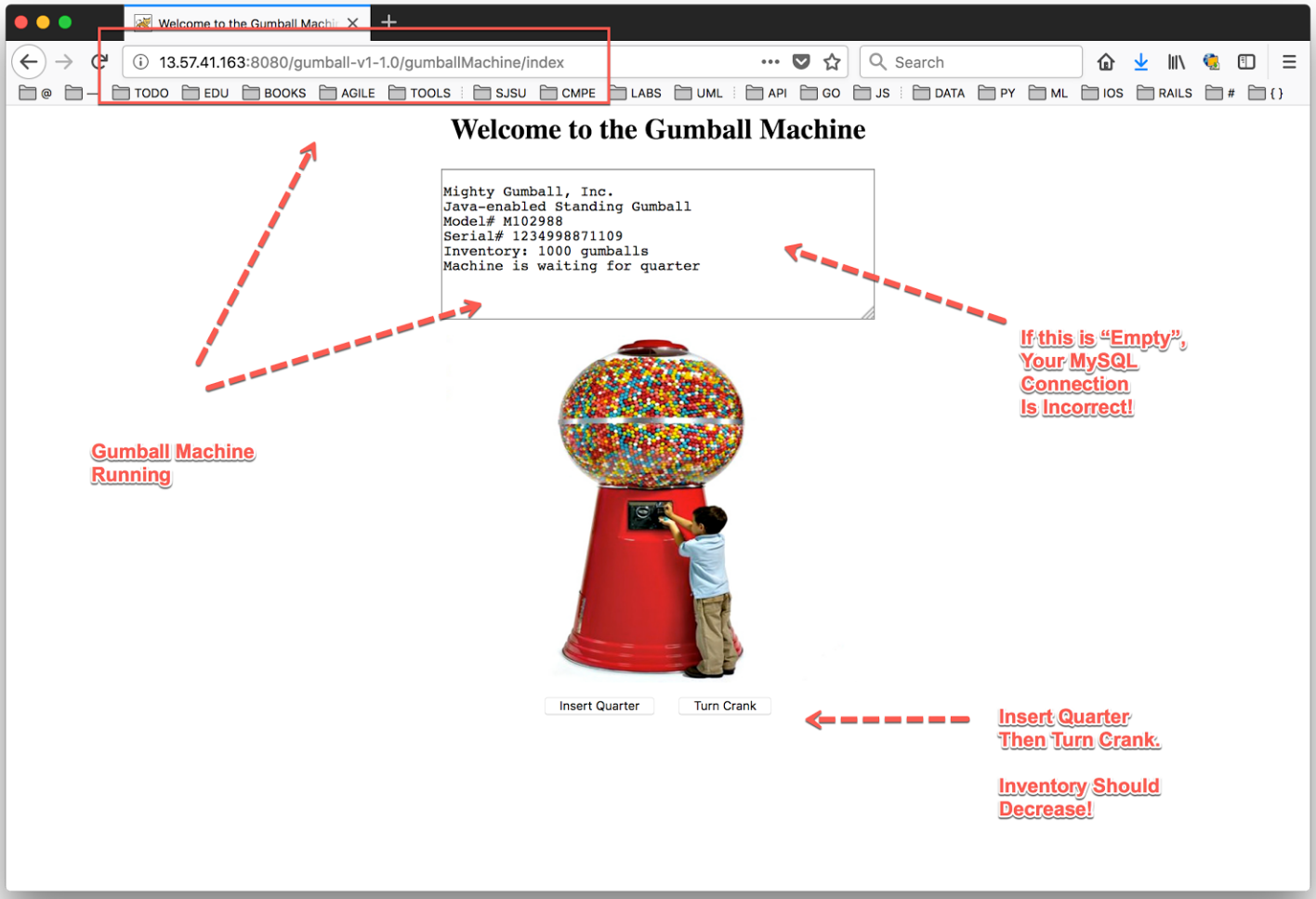
Check to see if a web application has caused a memory leak on stop, reload or undeploy

Find leaks This diagnostic check will trigger a full garbage collection. Use it with extreme caution on production systems.

Server Information

Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture	Hostname	IP Address
Apache Tomcat/7.0.82	1.8.0_151-b12	Oracle Corporation	Linux	4.9.76-3.78.amzn1.x86_64	amd64	ip-172-31-165-169	172.31.165.169


Welcome to Grails
13.57.41.163:8080/gumball-v1-1.0/
Search
Grails
Application Status
Artefacts
Installed Plugins
Grails App Running
On Port 8080
Welcome to Grails
Congratulations, you have successfully started your first Grails application! At the moment this is the default page, feel free to modify it to either redirect to a controller or display whatever content you may choose. Below is a list of controllers that are currently deployed in this application, click on each to execute its default action:
Click Here
To Test
Gumball Machine
Available Controllers:
gumball.v1.GumballBadLockController
gumball.v1.GumballController
gumball.v1.GumballMachineController


Welcome to the Gumball Machine
13.57.41.163:8080/gumball-v1-1.0/gumballMachine/index
Search
TODO EDU BOOKS AGILE TOOLS SJSU CMPE LABS UML API GO JS DATA PY ML IOS RAILS
Welcome to the Gumball Machine
Mighty Gumball, Inc.
Java-enabled Standing Gumball
Model# M102988
Serial# 1234998871109
Inventory: 1000 gumballs
Machine is waiting for quarter
Insert Quarter
Turn Crank
Gumball Machine Running
If this is "Empty", Your MySQL Connection Is Incorrect!
Insert Quarter Then Turn Crank.
Inventory Should Decrease!

Part III. Clean-Up (Terminate EC2 Instance)

Terminate Your AWS EC2 Instance.

Make sure to stop and terminate your EC2 instance when completed with the lab to avoid AWS charges.

Reference Lab Documents:

- https://github.com/paulnguyen/cmpe281/blob/master/aws/4-aws-tomcat-and-mysql.md
- https://readlearncode.com/cloud/amazon-free-usage-tier-installing-tomcat-7-on-an-ec2-linux-instance
- http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-LAMP.html
- https://dev.mysql.com/doc/refman/5.6/en/mysql.html