**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 MySQL on same Grails Instance
3. Update Grails App DB Config
4. Deploy WAR App to EC2 Instance and Start Up
5. Clean-Up (Stop and Terminate EC2 Instance)

---

# Part I. Install Java 8 JDK

### *Step 1: Launch EC2 Free-Tier Instance*

```
Type:                       t2.micro
AMI:                        Amazon Linux AMI 2018.03.0 (HVM)
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
```

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

### *Step 4: 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*
*Reload privilege tables now? = Y*

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

### *Step 5: 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 EC2 Instance

## 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 grails
sdk install grails 4.0.0
sdk current

grails --version

| Grails Version: 4.0.0
| JVM Version: 1.8.0_181
```

## Config Grails Database Connection for Production

Update your **grails-app/conf/application.yml** file to connect to your MySQL DB.  Note, make changes to the "***production***" database environment.

## 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.

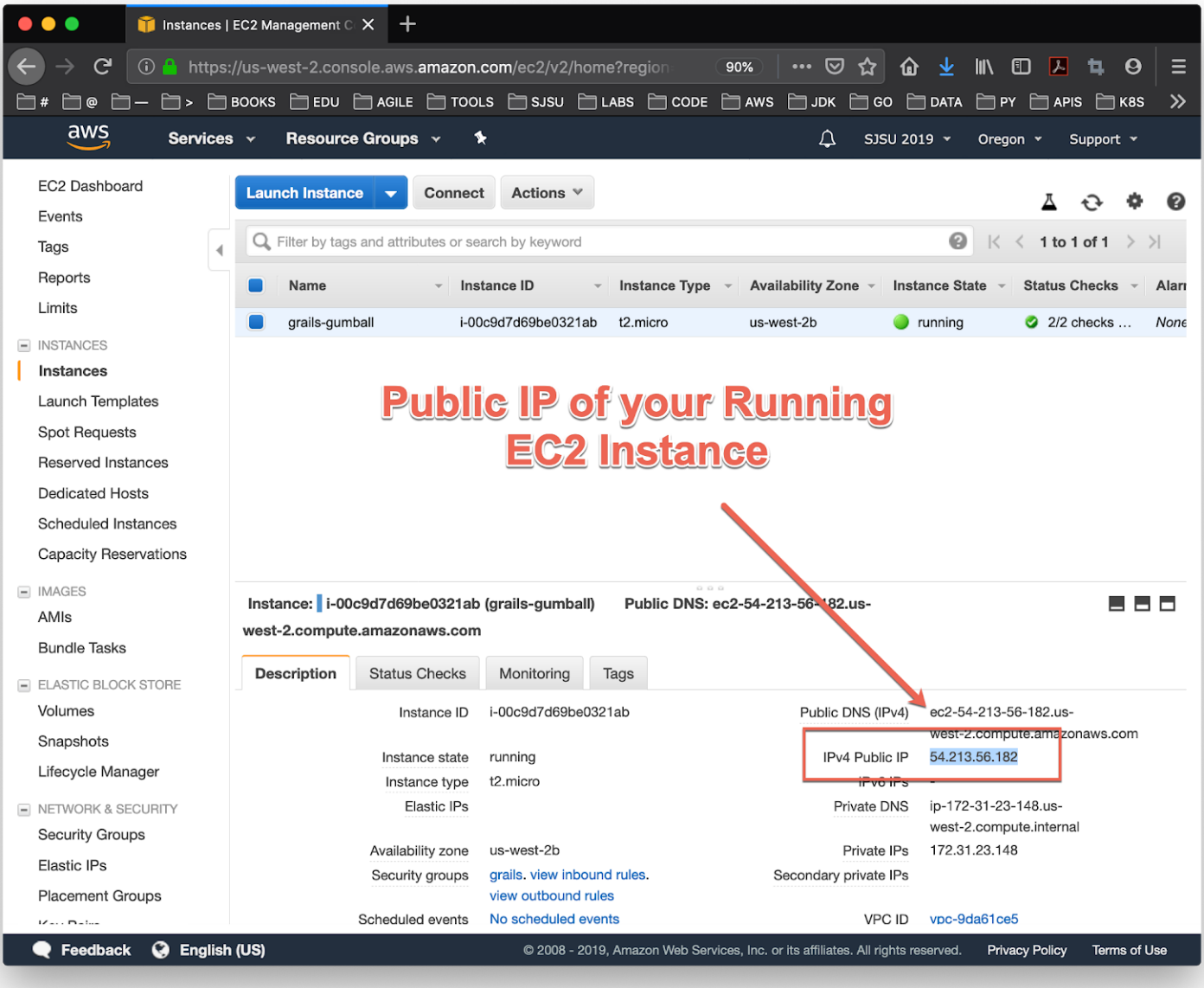
Copy (SCP) file "**gumball-v1-1.0.war**" to your EC2 Instance.  For example:

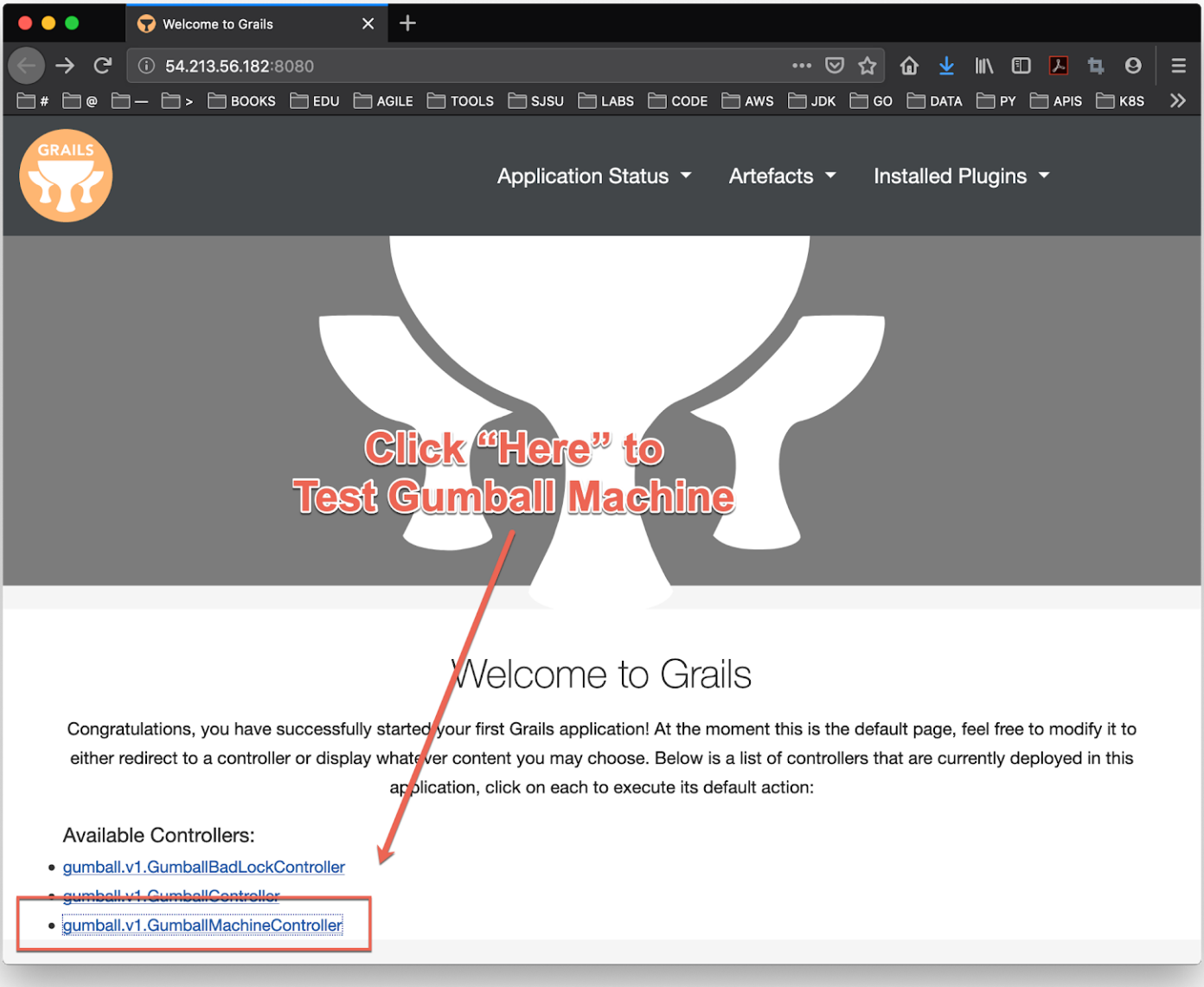
```
scp -i .<your-key-pair.pem> gumball-v1-1.0.war ec2-user@<your-host-ip>:/<path to your home dir>
```

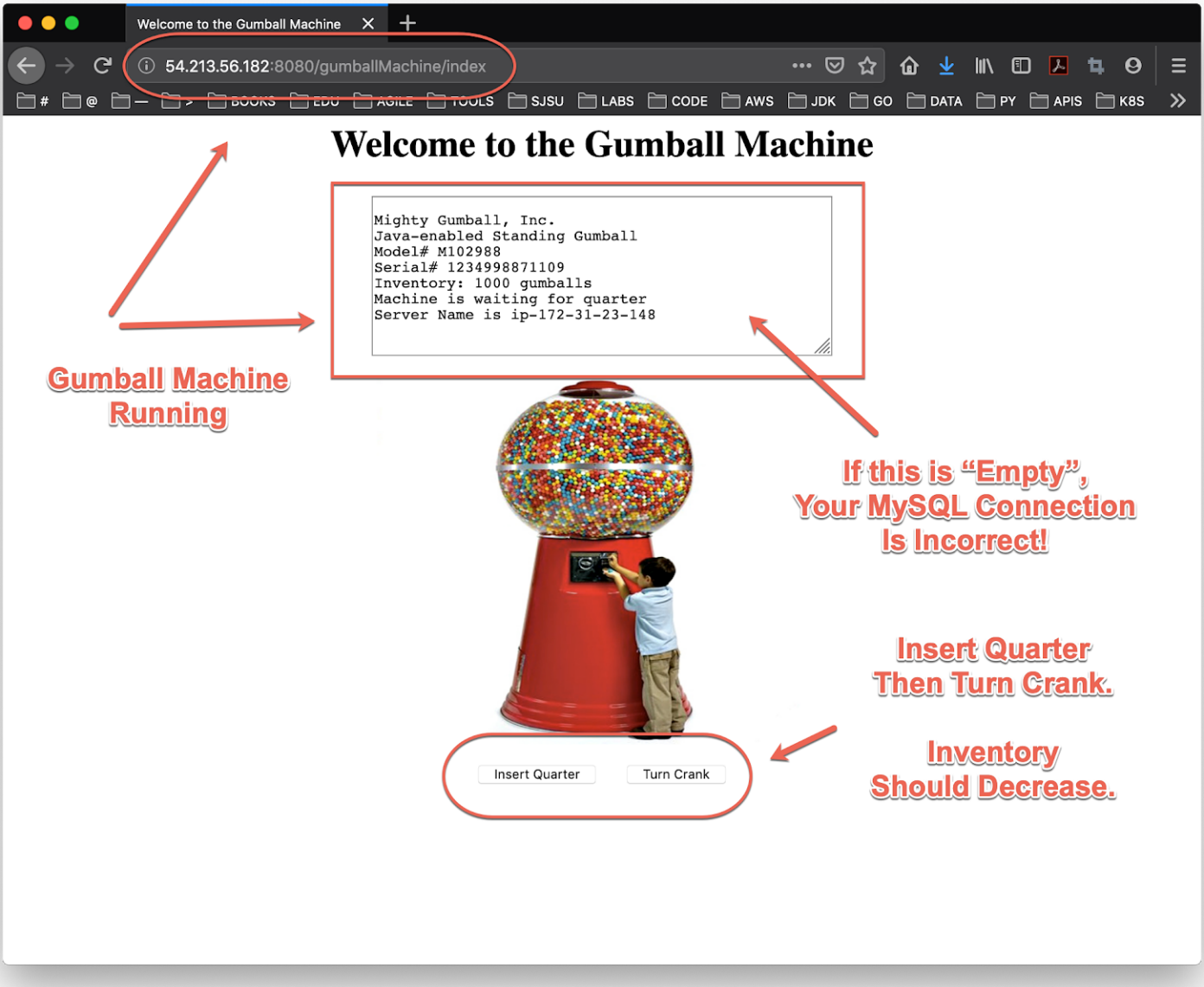
On the EC2 Terminal, run the following command:

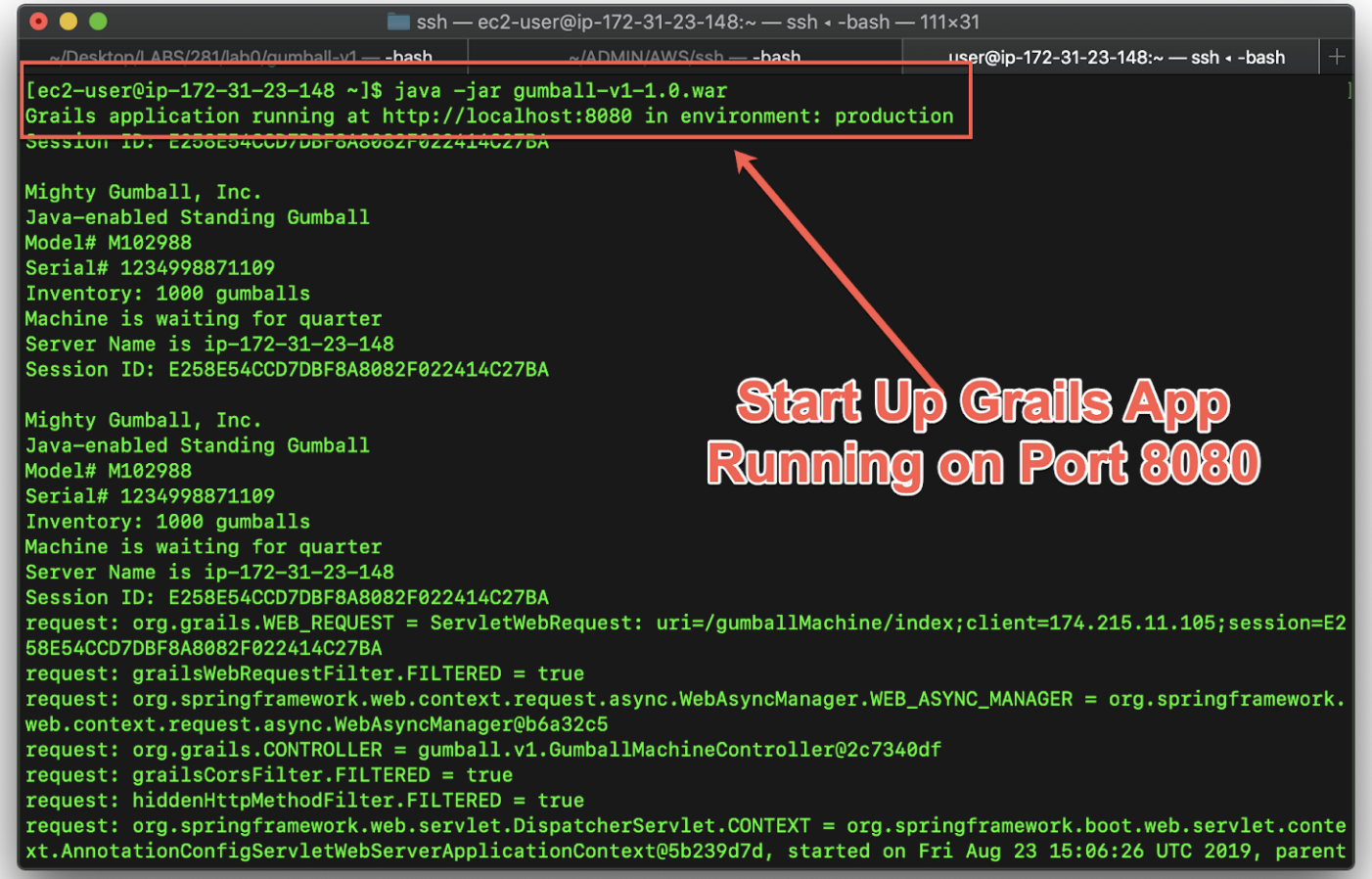
```
java -jar gumball-v1-1.0.war
```

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




Welcome to Grails
54.213.56.182:8080
BOOKS EDU AGILE TOOLS SJSU LABS CODE AWS JDK GO DATA PY APIS K8S
GRAILS
Application Status
Artefacts
Installed Plugins
Click “Here” to
Test Gumball Machine
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:
Available Controllers:
gumball.v1.GumballBadLockController
gumball.v1.GumballController
gumball.v1.GumballMachineController


Welcome to the Gumball Machine
54.213.56.182:8080/gumballMachine/index
Welcome to the Gumball Machine
Mighty Gumball, Inc.
Java-enabled Standing Gumball
Model# M102988
Serial# 1234998871109
Inventory: 1000 gumballs
Machine is waiting for quarter
Server Name is ip-172-31-23-148
Gumball Machine Running
If this is "Empty", Your MySQL Connection Is Incorrect!
Insert Quarter Then Turn Crank.
Inventory Should Decrease.
Insert Quarter
Turn Crank


ssh — ec2-user@ip-172-31-23-148:~ — ssh ◂ -bash — 111×31
user@ip-172-31-23-148:~ — ssh ◂ -bash
[ec2-user@ip-172-31-23-148 ~]$ java -jar gumball-v1-1.0.war
Grails application running at http://localhost:8080 in environment: production
Mighty Gumball, Inc.
Java-enabled Standing Gumball
Model# M102988
Serial# 1234998871109
Inventory: 1000 gumballs
Machine is waiting for quarter
Server Name is ip-172-31-23-148
Session ID: E258E54CCD7DBF8A8082F022414C27BA
Start Up Grails App
Running on Port 8080
Mighty Gumball, Inc.
Java-enabled Standing Gumball
Model# M102988
Serial# 1234998871109
Inventory: 1000 gumballs
Machine is waiting for quarter
Server Name is ip-172-31-23-148
Session ID: E258E54CCD7DBF8A8082F022414C27BA
request: org.grails.WEB_REQUEST = ServletWebRequest: uri=/gumballMachine/index;client=174.215.11.105;session=E2
58E54CCD7DBF8A8082F022414C27BA
request: grailsWebRequestFilter.FILTERED = true
request: org.springframework.web.context.request.async.WebAsyncManager.WEB_ASYNC_MANAGER = org.springframework.
web.context.request.async.WebAsyncManager@b6a32c5
request: org.grails.CONTROLLER = gumball.v1.GumballMachineController@2c7340df
request: grailsCorsFilter.FILTERED = true
request: hiddenHttpMethodFilter.FILTERED = true
request: org.springframework.web.servlet.DispatcherServlet.CONTEXT = org.springframework.boot.web.servlet.conte
xt.AnnotationConfigServletWebServerApplicationContext@5b239d7d, started on Fri Aug 23 15:06:26 UTC 2019, parent

# 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