

## 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:	<i>your-key-pair</i>

## 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 passwd = ***** (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)

The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, 'Services', 'Resource Groups', and a search bar. Below this is a left-hand navigation pane with categories like 'EC2 Dashboard', 'INSTANCES', 'IMAGES', 'ELASTIC BLOCK STORE', and 'NETWORK & SECURITY'. The main content area displays a table of EC2 instances. One instance, 'grails-gumball', is highlighted. Below the table, the details for this instance are shown, including its ID, state (running), type (t2.micro), and availability zone (us-west-2b). A red arrow points from the text 'Public IP of your Running EC2 Instance' to the 'IPv4 Public IP' field, which is highlighted with a red box and contains the value '54.213.56.182'.

**Public IP of your Running EC2 Instance**

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm
grails-gumball	i-00c9d7d69be0321ab	t2.micro	us-west-2b	running	2/2 checks ...	None

Instance: **i-00c9d7d69be0321ab (grails-gumball)** Public DNS: **ec2-54-213-56-182.us-west-2.compute.amazonaws.com**


Description		Status Checks	Monitoring	Tags
Instance ID	i-00c9d7d69be0321ab	Public DNS (IPv4)	ec2-54-213-56-182.us-west-2.compute.amazonaws.com	
Instance state	running	IPv4 Public IP	<b>54.213.56.182</b>	
Instance type	t2.micro	IPv6 IPs	-	
Elastic IPs		Private DNS	ip-172-31-23-148.us-west-2.compute.internal	
Availability zone	us-west-2b	Private IPs	172.31.23.148	
Security groups	<a href="#">grails. view inbound rules.</a> <a href="#">view outbound rules</a>	Secondary private IPs		
Scheduled events	No scheduled events	VPC ID	vpc-9da61ce5	


Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Welcome to Grails

54.213.56.182:8080

# @ > BOOKS EDU AGILE TOOLS SJSU LABS CODE AWS JDK GO DATA PY APIS K8S

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

# @ - > BOOKS EDU AGILE TOOLS SJSU LABS CODE AWS JDK GO DATA PY APIS K8S >>

# 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



Insert Quarter

Turn Crank

**Gumball Machine Running**

**If this is "Empty",  
Your MySQL Connection  
Is Incorrect!**

**Insert Quarter  
Then Turn Crank.**

**Inventory  
Should Decrease.**

```
ssh — ec2-user@ip-172-31-23-148:~ — ssh • -bash — 111x31
~/Desktop/ABS/281/lab0/gumball-v1 — -bash
~/ADMIN/AWS/ssh — -bash
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
Session ID: E258E54CCD7DBF8A8082F022414C27BA

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

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
```

**Start Up Grails App  
Running on Port 8080**

## 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>