

CS480 Software Engineering Assignment 7 - Deployment

Due Date

Wednesday, February 11, 2015

Score

5

Questions and Directions

This is a team assignment. Just creating one instance using your AWS account for your team. However, every team member should try the deployment process using the same instance.

In this assignment, you will be deploying your team project application in the cloud so that it is publicly accessible through the Internet.

To submit the assignment, please email the deployed URL to your project page, such as:

`http://ec2-54-67-30-104.us-west-1.compute.amazonaws.com:8080/cs480/home`

Please email to both yusun@cpp.edu and yuchingsu@cpp.edu

0. Build Your Project with Maven in Command-Line Mode

You should be able to run `mvn package` in your project folder and have the whole project built correctly. The final deployable artifact for your project should be the `jar` file in the target directory.

You cannot move on to the following steps unless you can successfully build your project as a jar file using Maven.

1. Deployment Prerequisite

- You need to have an AWS account. Apply it at <http://aws.amazon.com/free/>
- If you are using Windows, make sure to download the following 3 tools:
 - Putty (putty.exe) <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
 - PuttyGEN (puttygen.exe)
<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
 - WinSCP
<http://winscp.net/eng/index.php>
- If you are using Mac/Linux, all you need is the command line shell with `ssh` and `scp` command working (should be working by default)

- When performing this assignment with the university's WIFI, please use the WIFI "CPP" with your credential, rather than the "GUEST".

2. Start an EC2 Instance (Server) in AWS

Login to your AWS console at: <https://console.aws.amazon.com/console/home>

Start a new server with the following steps:

1. Go to EC2
2. Go to "Key Pairs", and click on "Create Key Pair". This creates the private key to access your server (it is a type of password to access and login to your server). Make sure you give it a name and then store the key file in a safe location in your local disk. **Your team is going to share the same key file. Please make sure to share the key file to all your team members.**
3. Go to "Instances", and click on "Launch Instance"
4. Select the very first image type **"Amazon Linux AMI 2014.09.2"**
5. Select the **"General Purpose" - t2.micro (Free tier eligible)**
6. For the rest of the configurations, just choose the default settings
7. Finally, you will need to select the key pair you just created to indicate using it to access the server
8. Go back to EC2/Instances, you should be able to see a new instance being initialized and eventually turned to be "running" with green light
9. When the instance is up and running, clicking on the instance will tell you the "Public DNS" name, something like *ec2-54-67-30-104.us-east-1.compute.amazonaws.com*

3. Access and Login to Your EC2 Instance Server through SSH

- If you are using Windows, please follow the guide at <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html> to access your server using Putty. Please follow the whole guide except the last part on "Transferring Files to Your Linux Instance Using the PuTTY Secure Copy Client" and "Transferring Files to Your Linux Instance Using WinSCP".
- If you are using Mac/Linux, please follow the guide at <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html> to access your server using ssh in your shell. Please follow the whole guide except the last part on "Transferring Files to Linux Instances from Linux Using SCP".

Note, the guide mentions using AWS CLI tools such as [describe-instances](#) and [ec2-describe-instances](#). Please ignore them, since you do not have to use this tool.

If you can successfully login to your server, you should be able to type the regular Linux command there, such as "ls", "date", "java", etc.

Don't close the terminal, since you will need to run other commands later.

4. Copy Your Project Jar File to the EC2 Instance Server and Run the Jar File

- If you are using Windows, please follow the guide at <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/putty.html> with the last part on "Transferring Files to Your Linux Instance Using WinSCP".
- If you are using Mac/Linux, please follow the guide at <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html> with the last part on "Transferring Files to Linux Instances from Linux Using SCP".

Note, it is recommended to simply put the **jar file** in the default home directory in the Linux machine, **/home/ec2-user/**

If you can successfully copy the jar file, you should be able to run the following command in the terminal (putty or ssh) to start your application remotely in the server:

```
java -jar cs480-1.0.jar
```

5. Open EC2 Instance Internet Port 8080

You need to open the port 8080 so that your server can be publicly accessible.

Go to the AWS console, and select the instance in EC2/Instances. You will see the detailed information about the instance as shown below:

Instance: **i-c3f38809** Public DNS: **ec2-54-67-30-104.us-west-1.compute.amazonaws.com**

Description	Status Checks	Monitoring	Tags	
Instance ID	i-c3f38809		Public DNS	ec2-54-67-30-104.us-west-1.compute.amazonaws.com
Instance state	running		Public IP	54.67.30.104
Instance type	t2.micro		Elastic IP	-
Private DNS	ip-172-31-7-184.us-west-1.compute.internal		Availability zone	us-west-1c
Private IPs	172.31.7.184		Security groups	launch-wizard-2 . view rules
Secondary private IPs			Scheduled events	No scheduled events
VPC ID	vpc-593acf3c		AMI ID	amzn-ami-hvm-2014.09.1.x86_64-ebs (ami-4b6f650e)

Click on the group in "Security Group" will lead you to the console to configure the "Inbound" and "Outbound".

Click on "Inbound", add a new Rule with "8080" port. (Don't delete the existing 22, just add a new port in a new rule). Then, save your changes.

6. Test Your Application with the Public URL

Make sure that your application is still running with step 4. Your final public URL to your project will be something like:

<http://ec2-54-67-30-104.us-west-1.compute.amazonaws.com:8080/>

You need to replace the domain name with the public DNS name of your instance server. If you can see the application up and running, congratulations, you are almost done!

7. Keep Your Application Running in the Background

Your application running with step 4 will stop if you close the terminal window. To keep running in the background all the time, you will need the following command:

```
nohup java -DXms512m -DXmx1024m -jar cs480-1.0.jar &
```

The Xms and Xmx configures the memory for the application. nohup helps keeping the application running after you close the terminal. See <http://en.wikipedia.org/wiki/Nohup> for the purpose of nohup command.

8. (Optional) Use 80 rather than 8080

If you feel the 8080 port in the URL is annoying, you can redirect all the requests from 80 to 8080, so that users can use your URL without the port number. To do so, you need to:

1. Open the 80 port in your security group as shown in step 5.
2. Follow the instructions at: <http://proghowto.com/iptables-redirect-port-80-to-port-8080>

9. (Optional) Use Your Own Domain Name

If you already have a registered domain name, such as john.com, you can easily connect it with your server. You just need to go to the registrar console (e.g., GoDaddy control console), and add a CNAME to the public DNS name of your server. Please google how to setup CNAME to an EC2 instance for more information. Or you can use an A record to do it as shown at:

<http://www.quora.com/If-I-bought-a-domain-name-from-Godaddy-but-plan-to-use-amazon-EC2-to-run-the-site-do-I-need-hosting-from-Godaddy>