CS 548—Spring 2018
Enterprise Software Architecture and Design
Assignment One—Enterprise in the Cloud

In this assignment, you will set up the IT infrastructure that you will be using for deploying your applications later in the semester. You will set up this infrastructure in the Amazon Elastic Compute Cloud (EC2), which provides an infrastructure-as-a-service (IaaS) for configuring a custom software stack. In sum, the steps are as follows:

1. Launch an EBS-backed EC2 instance. Launch the instance from a bare Amazon Linux 64-bit AMI, without any additional software installed. Elastic Block Store (EBS) is essentially a virtual disk that Amazon provides for backing storage for EC2 instances.
2. Allocate a second volume in EBS, for storing a database. Attach this volume to the instance, and mount it as an external file system on your instance.
3. Install a database management system (postgresql) on the instance, using the external file system to store its data. *Use version 9.x of the Postgresql database management system, which is the version currently supported by AWS.* Configure postgresql to allow external access to certain users that you create, so you can administer the database server remotely. Create a database that can be accessed for querying and updating by a simple user with no administrative privileges. The steps to do this follow the demo videos, though you will need to provide the `--interactive` option to the createuser command:
   ```
   /usr/bin/createuser --interactive lord
   /usr/bin/createuser --interactive serf
   ```
4. Install OpenJDK and the Payara Open Source application server[1]. *Use the Java 8 JDK. Use version 174 Full Server of the Payara application server.* Configure the app server to run as a system service, with the limited privileges of a special user `glassfish` that you create, not as `root`. *Enable secure administration, so that the application console can be accessed from a remote host.* The default administrator password is blank, and the default master password for protecting the application server keys is "changeit". You should change both of these to more secure passwords. Use the `–savemasterpassword` option with the `change-master-password` command to save the master password on the file system, otherwise Payara will hang your machine boot when it tries to start the service.
5. Configure a connection pool for your database in the application server, authenticating with the credentials of the simple database user, and a JNDI resource that applications will use to access the database.

[1] Payara is a supported version of the Glassfish application server (v4.1). It unzips to a folder named `payara41`. You should rename this to `glassfish4` to follow the rest of the demo videos provided for this assignment. There is a new version of Glassfish, v5.x, that is the reference implementation for Java EE 8, but I don't want to use this until there is more experience with it, since it is not supported.

6. Deploy a simple `Messages` application that you will be provided with.
7. Save an EBS image of your customized instance as an AMI.

You should follow the detailed instructions provided in the podcast demos of setting up an EC2 instance[2], in the Canvas classroom for this course. *Note: Wherever the assignment specification differs from the demo podcasts, you should follow the assignment specification. The podcasts are only there for guidance on how to complete the specification*. As an addition to the information in the podcasts, follow these steps to ensure that Postgresql starts automatically when your EC2 instance starts:

1. In /etc/init.d/postgresql, make sure you have defined these variables:
```
# Set defaults for configuration variables
PGENGINE=/usr/lib64/pgsql93/bin
PGPORT=5432
PGDATA=/data
PGLOG=/data/pgstartup.log
```
2. Type this command in Bash to enable autostart of Postgresql:
`sudo chkconfig postgresql on`

You should download Payara directly to your local machine, and then use scp or winscp to copy this file to EC2.

In addition, do the following to save your personal information in the instance:

```
cd
echo "YOUR-NAME" > info.txt
echo "YOUR-CWID" >> info.txt
echo "YOUR-EMAIL" >> info.txt
```

For your submission, provide videos and a report in PDF format that describes what you did, including each of the following:

a. Screenshot of your AWS Console showing the running instance, including its external DNS.
b. Screenshot of your AWS Console showing the volumes you have allocated on EBS.
c. Output of Linux command "df" in instance after Step (2).
d. Output of "`ls –l /data`" after Step (3).
e. Output of "`more -10 /data/pg_hba.conf`" after Step (3).
f. Output of "`psql –U postgres –c '\du'` " after Step (3).
g. Output of "`java –version`" after Step (4).
h. Output of "`ls –l /usr/share/glassfish4`" after Step (4).
i. Screenshot of your Payara Admin Console, showing `Resources|JDBC|Connection Pools`, after Step (5).

[2] There is no point in installing the Oracle JDK, since it is derived from the same codebase as OpenJDK, which can be installed using yum in Amazon Linux.

j. The administrator password you chose for Payara. This password should be secure ("abc123" or "glassfish" or "c548" are not acceptable), but not ones that you use for any other business outside of this class.
k. Video of a demonstration of your deployment working. In this video, demonstrate the starting of the database server and the starting of the application server (the latter using asadmin), then your logging in to the application server administration console, deployment of the `Messages` application, and finally your running this application[3].

You should not leave the EC2 instance running, since you may incur bills of hundreds of dollars from Amazon if you do this. Instead, leave your instance stopped, and use IAM to provide graders with access to your EC2 console, so they can start the instance, and grant them access to your EC2 instance via ssh public keys. See the separate document detailing how to grant graders access to the instance.

You are also strongly encouraged to back up your instance as an Amazon machine image (AMI). This way, if the instance ever becomes corrupted in some way, you can instantiate a new version from the AMI that you have created. However you do not need to share the AMI with the graders, they will just be accessing the instance that you have stopped. Be sure to include, in the report for this and later assignment submissions, complete instructions on how to start the instance from your EC2 console. They should then be able to ssh to the instance, if they need to. Payara and postgresql should start automatically when the instance starts, if you have set it up right.

Make sure that your name appears at the beginning of the video. For example, display the contents of a file that provides your name. *Do not provide private information such as your email or cwid in the video*. Be careful of any "free" apps that you download to do the recording, there are known cases of such apps containing Trojan horses, including key loggers.

Your submission should be uploaded via the Canvas classroom, as a zip file. This zip file should have the same name as your Canvas userid. It should unzip to a folder with this same name, which should contain the files and subfolders with your submission.

It is important that you provide a document that documents your submission, included as a PDF document in your submission root folder. Name this document README.pdf. This should document the video(s) that you provide demonstrating that you have correctly set up your cloud environment.

---

[3] You may as well get used to undeploying your application before you shut down the server, since restarting a deployed app when Glassfish restarts is not reliable.