**Set Up AWS EC2 Instance**

* Name: gheskio
* AMI: Amazon Linux AMI 2017.09.1 (HVM), SSD Volume Type (64-bit)
  + - * The Amazon Linux AMI is an EBS-backed, AWS-supported image.
      * The default image includes AWS command line tools, Python, Ruby, Perl, and Java.
      * The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
* Instance: i-08141982e75b5aeca (gheskio)
* Instance type: ts.micro
* Public DNS: ec2-52-15-156-17.us-east-2.compute.amazonaws.com
* My Key Name: gheskio\_app (stored on my personal laptop)
  + *NOTE*: cannot have multiple key pairs. Can manage through user\_data but, given low data security requirements of test app server, will simply share the same key pair. Added security can be introduced by only allowing inbound from specific IPs.
* Username: ec2-user
* Security Group: gheskio\_security\_group
  + HTTP - Port 80 (All IPs)
  + HTTPS - Port (All IPs)
  + SSH - Port 22 (All IPs)

**Connect to Instance via SSH**

* Convert Private Key - PuTTYgen
  + Use PuTTYgen to convert .pem file (saved when instance created [above]) to .ppk
* Connect - PuTTY
  + Host Name: ec2-user@ec2-52-15-156-17.us-east-2.compute.amazonaws.com
    - * (user\_name@public\_dns\_name)
  + Connection Type: SSH
  + Port: 22
  + Category -> Connection -> SSH -> Auth
    - Browse -> select .ppk file (created above) -> Open
  + Save Session: [[give it a name]]
  + Open
  + FIRST TIME: check fingerprint

**Set Up Server on Instance**

* + Upgraded to Java 8
    - Root access: sudo -i
    - Check the Java version: java -version
    - Install 1.8: sudo yum install java-1.8.0 -openjdk-devel
    - Make 1.8 the default: sudo /usr/sbin/alternatives --config java
  + Set JAVA\_HOME environment variables
    - In /etc/profile [root permission w/ sudo -i] directory vi .bashrc
      * Unsure exactly which library to point to "export JAVA\_HOME=/usr/lib/jvm/default-java"
      * Helpful vi commands:
        + "i" = insert mode
        + "esc" = command mode
        + ":" then "q!" = Quit without saving
        + [Shift-z-z] = Quit and save
  + Apache Tomcat
    - Explore available packages: yum search tomcat
    - Install: yum install tomcat8-webapps tomcat8-docs-webapp tomcat8-admin-webapps
    - Set up users
      * Navigate to: cd usr/share/tomcat8/conf
      * Open: nano tomcat-users.xml
      * [[TBD - assign users that can write to the server]]
    - Make server accessible
      * Set the TCP Port Number
        + /conf/server.xml

Connector port = "8080"

* + - * Enabling Directory Listing
        + /conf/web.xml

<param-name>listings</param-name>

<param-value>true</param-value>

* + - * Enabling Automatic Reload
        + Not Applicable
    - To start Tomcat: sudo service tomcat8 start
      * Test server: http://ec2-52-15-156-17.us-east-2.compute.amazonaws.com:8080/hello/HelloHome.html
    - To stop Tomcat: sudo service tomcat8 stop
  + MySQL
    - Install: yum install mysql-server
    - Mysql installation set up: run usr/libexec/mysql55/mysql\_secure\_installation
      * Password: gheskioag
    - Mysql -u root -pgheskioag (log in into current mysql server as user “root”, can replace “root” with any other username)
    - Some examples of mysql command
      * show databases;
      * use [schema\_name];
      * show table;
      * describe [table\_name];
      * select \* from [table\_name];
      * load data infile “file path/file.csv” into table “table\_name” fields terminated by “,” lines terminated by “\n”; (mysql only accepts files stored in /var/lib/mysql-files for now due to set-up. If you upload file in other directory, you will get permission denied. We may want to re-visit this in the future).
      * grant all on [schemaName].[tableName] to ‘username’@’userhost’; (grant privilege to user)
      * flush privileges; (used with above command)
    - Resources
      * <http://nahan.github.io/2016/03/15/setup-tomcat-mysql-aws/>
      * <https://www.ntu.edu.sg/home/ehchua/programming/howto/Tomcat_HowTo.html>
  + Create testing servlet and set up mysql-connector-java
    - Set up mysql-connector-java
      * yum install mysql-connector-java-5.1\*
      * Back out to root directory -> find -name mysql-connector\*
      * cd usr/share/java
      * cp mysql-connector-java-5.1.12.jar /usr/share/tomcat8/lib
    - Set up servlet files
      * <https://www.ntu.edu.sg/home/ehchua/programming/howto/Tomcat_HowTo.html>
      * javac -cp .:/usr/share/tomcat8/lib/tomcat8-servlet-3.1-api.jar QuerryServlet.java
      * in querybook.html -> change locahost to public DNS address (localhost is supposed to work but not sure why it doesn’t, probably the website was meant for hosting this server on localhost).
      * in QueryServlet.java -> change “myuser” to “root” and “xxxx” to “gheskioag”
    - Testing
      * Go to: <http://ec2-52-15-156-17.us-east-2.compute.amazonaws.com:8080/hello/querybook.html>
      * Select any author and Voila!
  + Set up server with the app
    - Move UploadServlet.java and SerialQRecord.java to /usr/share/tomcat8/webapps/gheskio (gheskio app directory on tomcat)
    - Grab "javax.servlet-api-3.0.1.jar" on the net
    - Compile java class with required dependency: javac -cp "javax.servlet-api-3.0.1.jar" \*.java
    - Make sub folder (to accommodate java package) => ..classes/org/gheskio/classes/ => copy UploadServlet.class and SerialQRecord.class in here
    - Testing: Go back to main ..classes/ and run: java -classpath ".:javax.servlet-api-3.0.1.jar:/usr/share/tomcat8/lib/jtds.1.3.1.jar:/usr/share/tomcat8/lib/mysql-connector-java-5.1.12.jar" org.gheskio.queue.UploadServlet -j "jdbc:mysql://localhost:3306/gheskio" -u root -p gheskioag -t anything\_prob -f emptyfile.txt (emptyfile.txt is an empty text file in the same folder)
    - Set up database and permission in MySql server => log into mysql server => run: GRANT ALL PRIVILEGES on gheskio.\* to ''@'%' with grant option;
  + Edit Chris’s codes to accommodate MySql instead of MsSql
    - In UploadServlet.java (line 39) => change “ ’ as datatime2(7)))” to “ ‘ as datetime)”
    - Recompile UploadServlet.java into UploadServlet.class
  + Test simple upload into mysql server:
    - Log into mysql => create database gheskio => create table abc
    - Create testfile.txt containing one line of text: aa|ss|zz|1521920617|ww|tt|bb|hh|ss
    - java -classpath ".:javax.servlet-api-3.0.1.jar:/usr/share/tomcat8/lib/jtds.1.3.1.jar:/usr/share/tomcat8/lib/mysql-connector-java-5.1.12.jar" org.gheskio.queue.UploadServlet -j "jdbc:mysql://localhost:3306/gheskio" -u root -p gheskioag -t abc -f testfile.txt
    - terminal should return “inserted: aa|ss|zz|1521920617|ww|tt|bb|hh|ss “
  + Run the app on Android Studio
  + Make upload from the app
    - Find out how to upload using http request (not running UploadServlet directly in the server like above) => include find out to parse UploadServlet arguments from the http request => maybe need another Servlet? Maybe do it in web.xml file?
    - Try to upload using html http request from the browser first? (use some test file???)
    - Explore Chris’s codes to find where POST request is called => change the http path to the correct path
    - Run the app and let it connect and post to the server

Updates 5/5/2018

**Handle server codes:**

* Give yourself editing privilege:
  + sudo -i
* Change directory to Gheskio app:
  + cd /usr/share/tomcat8/webapps/gheskio/WEB-INF/classes
* Test upload using PostTest (already compiled):
  + java -cp . org.gheskio.queue.PostTest testfile.txt <http://ec2-52-15-156-17.us-east-2.compute.amazonaws.com:8080/gheskio/upload>
* Test upload using curl: (can see HTML response of POST request, right database log-in parameters were set up)
  + curl -X POST -d @testfile.txt <http://ec2-52-15-156-17.us-east-2.compute.amazonaws.com:8080/gheskio/upload> -v
* Edit and recompile PostTest (from …/classes folder)
  + cd org/gheskio/queue
  + vim PostTest.java
  + :x to save and exit, :q! to NOT save and exit
  + Javac -cp . SimpleAuth.java PostTest.java (=> to recompile PostTest.java into a class)
  + cd ../../..
  + Test your changes
* Test upload manually within the Tomcat server by passing data directly to UploadServlet (only this works so far)
  + java -classpath ".:javax.servlet-api-3.0.1.jar:/usr/share/tomcat8/lib/jtds.1.3.1.jar:/usr/share/tomcat8/lib/mysql-connector-java-5.1.12.jar" org.gheskio.queue.UploadServlet -j "jdbc:mysql://localhost:3306/gheskio" -u root -p gheskioag -t abc -f testfile.txt
  + You should see two more lines added to table ‘abc’ in database ‘gheskio’
* Edit and recompile UploadServet (from ../classes folder)
  + cd org/gheskio/queue
  + vim UploadServer.java
  + :x to save and exit, :q! to NOT save and exit
  + javac -cp "javax.servlet-api-3.0.1.jar" SerialQRecord.java UploadServlet.java (=> to recompile UploadServlet.java into a class)
  + cd ../../..
  + Test your changes

**Handle mySql database:**

* Open a new AWS EC2 command prompt
* Log into mySql database:
  + mysql -u root -pgheskioag
* Change to ‘gheskio’ database:
  + use gheskio;
* Check content of ‘abc’ table:
  + select \* from abc;