CS 455 Database Management Systems **Fall 2015**

Project 1: To the Cloud!

Instructor: David Chiu *Be prepared for demos after Sep 28 (50pts)*

1 Preliminary

1. Before you do anything, you should form a team containing 3-4 people. When you've settled on a team, please email me with the names your members.
2. Next, you're required to join slack. Go to: `https://univpugetsound.slack.com/signup`
3. For your email, use your `pugetsound.edu` address. It's set up so that it only accepts users with our school's domain to keep out strangers.
4. If asked, the Team Name is `univpugetsound`
5. Once logged into the team page, join the `#cs455` channel. This is where general course discussion takes place.
6. You will receive an invitation from me to invite you to your group's private channel, where you'll have project-based communication.
7. It is recommended that you download their app (iOS and/or Android) to get notifications on your phone or tablet.

2 Introduction

In this preliminary project, you will learn the ins-and-outs of one of Amazon's most popular cloud services, Elastic Compute Cloud (EC2). Under EC2, users can allocate virtual machines (hosted on Amazon's hardware) on a pay-as-you-go basis. While you don't need to worry too much about the specifics of Amazon's pricing model for this class, it doesn't hurt for you to do some back-of-the-envelope calculations on costs. You can easily find their pricing information (*i.e.*, cost per instance-hour) online.

Why are you learning how to use the cloud in a database course? Good question! This is all in preparation for your term project, which is to build a web application over a database back-end. This is commonly known as the *3-Tier Architecture* in web development. We'll get to details of that in the next project phase. For now, we'll simply focus on getting a Linux server up and running.

Learning Outcomes

The goal of this project is to get you started with:

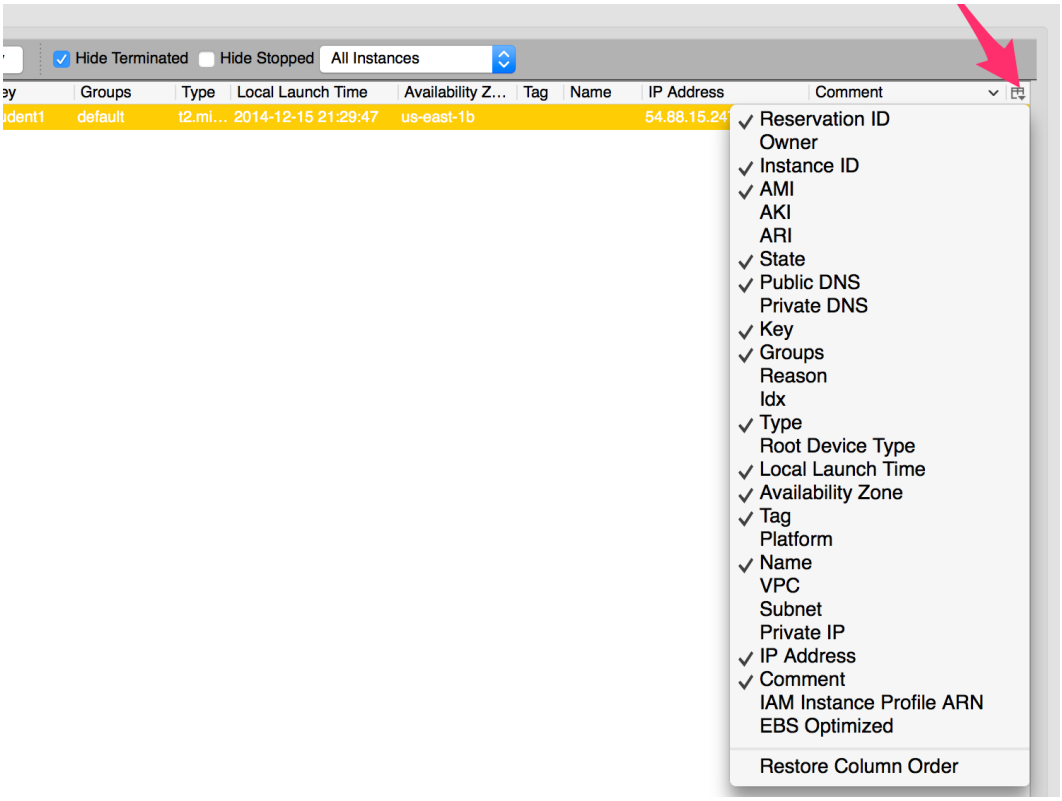
- Understand the concept of cloud computing and virtualization
- Use Amazon EC2 cloud instances
- Become familiar with common Linux commands
- Getting to know the Apache Web Server

Grading

This is a preliminary phase, with no requirements for write-up or any code submissions. You will be graded completely on your group's demo (Q&A session). This phase will only be worth 25pts.

3 Launching an EC2 Instance

1. Before you start, make sure you have Firefox installed on your machine.
2. Download the file located at `http://cs.pugetsound.edu/~dchiu/CS455/notes/AWS2015.txt`.
3. Next, go to `http://elasticfox-ec2tag.s3-website-ap-northeast-1.amazonaws.com`, and download the latest version of `elasticfox-ec2tag-*.xpi`. This file is a Firefox extension.
4. Open the `.xpi` file with Firefox. After restarting Firefox, click on "Tools" then click "ElasticFox-ec2tag".
5. Under "Credentials," input the account information given in `AWS2015.txt`.
6. Make sure `us-east-1` is selected under "Regions" on the top left corner.
7. Click on the "Instances" tab, and place a check-mark next to "Hide Terminated." You'll also notice that there are way too many columns showing. Let's hide the ones that you won't care about. Click on the right-most column button, and de-select the unused columns, shown below:



8. Now click on the "Images" tab. Change the selection to "Amazon AMIs." These AMIs are saved virtual machine images that can be run. In the search bar to the left, search for `ami-08249861`.

9. Right click on it and click "Launch Instances of this AMI"). In the dialog box that follows, make sure you do the following:

   (a) Under "Instance Type": Select `m3.medium`

   (b) Under "Number of Instance(s)": Input `1` for both Min and Max

   (c) Under "New Instance(s) Tag": input `Group <num>`

   (d) Under "Keypair": select your group's ID (this is important!).

   (e) Under "Availability Zone": select `us-east-1b`

   Leave the remaining fields untouched, and click "Launch."

10. This brings you back over to the "Instance" menu. Wait until your machine's the status says "Running."

11. **Important**: Note your instance ID, key, and public DNS!

12. **Super Important**: Yes, I'm aware that you can see and even control all other teams' instances. You must **tag** your group's instance with your group number! Please do not mess with other groups' instances (if terminated without their knowledge, other teams could be losing `all` their code). If you're caught doing this with malicious intent, you will automatically fail my course and be reported to Academic Misconduct for breaking the honor code.

4 Accessing Your EC2 Instance

1. First, you need to get an SSH client.

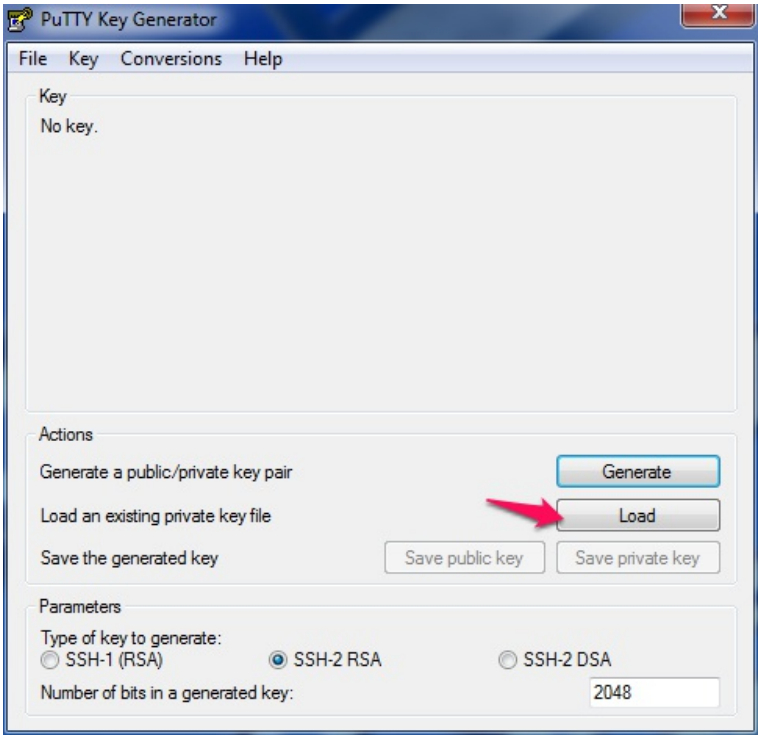
   - On Linux: Your terminal would do just fine.
   - On MacOS: Your terminal would work, but I also like iTerm2.
   - On Windows: Download PuTTY and PuTTYgen.

2. Make sure I sent you group a private key, with the filename `id_group<num>` where `<num>` is your group number. You will need to change the permissions of this file to read-only, *i.e.*, `chmod 400 path/to/id_group<num>`.

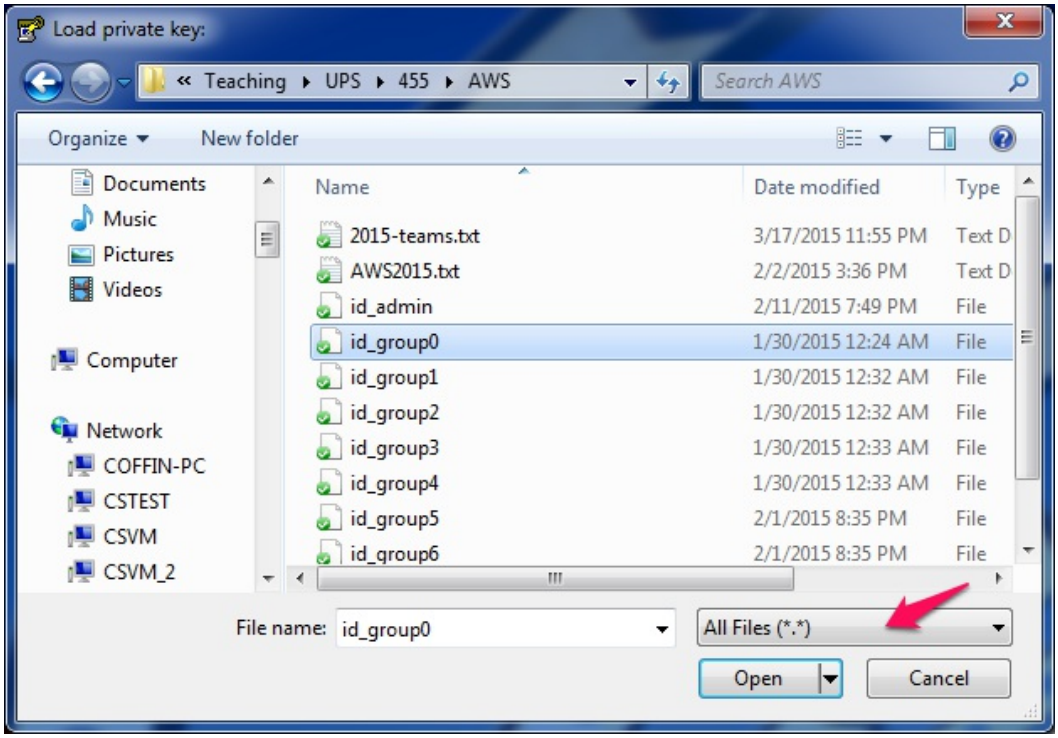
3. **On Mac/Linux:** Open up a terminal program and type in `ssh -i path/to/id_group<num> ec2-user@public_dns`, where `id_group<num>` is your group's private key, and `public_dns` is the hostname of your EC2 instance.

4. **On Windows:**

   (a) Open up PuTTYgen, and click "Load."

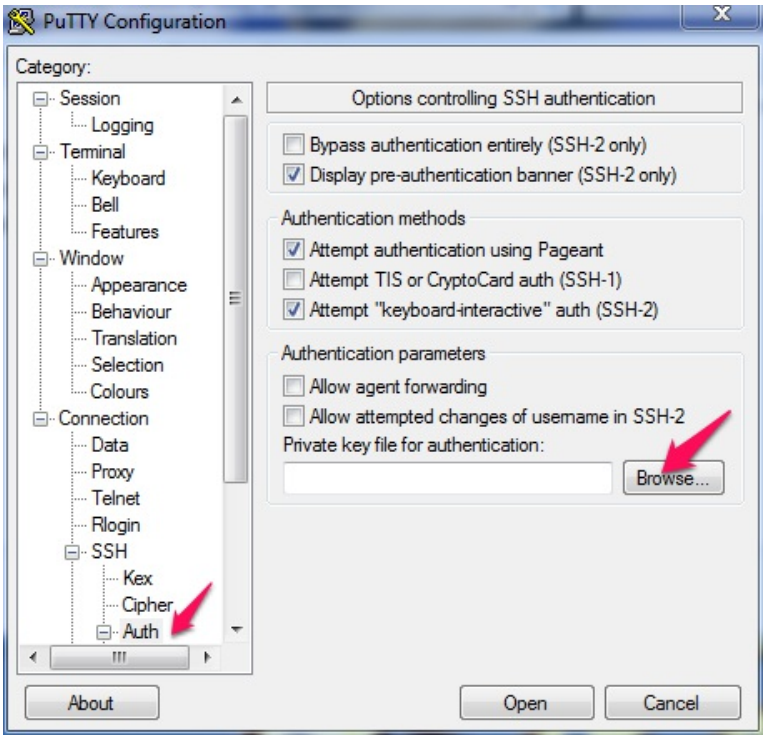


(b) Make sure "All Files" is selected on the bottom righthand corner, then open the private key file I gave you and your team.



(c) You should get a message that says the private key has been successfully loaded. Now click on the "Save private key" button, and put it somewhere safe where you'll remember. This is the only time that you'll use PuTTYgen.

(d) Now open up PuTTY. On the left-hand panel, navigate to "SSH → Auth." Then click "Browse..." and load the (`.ppk`) private key file you just generated and saved.



(e) Back in the "Session" menu on the left-hand panel, enter the hostname or IP address of your AWS instance. Under "Saved Sessions," type in "AWS" and click "Save." You only have to do this once.

(f) Double-click on "AWS," and PuTTY will ask you to "Login as:" You should enter `ec2-user`. That should be it!

(g) **Important:** The next time you open PuTTY, click on "AWS" then click the "Load" button. This will put the instance's hostname in the input box up on the top. Make sure this still matches your instance's Public DNS (or IP Address) on elasticfox, and click Open.

5. The connection might time-out or be refused at first if your instance had just launched. It takes Amazon several minutes to set up your instance's network.

6. Once you have logged in, play around and get used to the Linux environment.

7. Create a file in your instance called `testing.txt`, and type in the following: `''CS 455 is cool.''` Save the file.

8. Go back to ElasticFox, right click on your running instance and hit `Terminate`.

9. Re-launch the AMI image and log into the new instance using ssh. Note the `testing.txt` file you created is gone. Create it again, and this time hit `Stop` in ElasticFox. To restart it, click on the green

button, "Start an EBS backed instance." Log back into your instance. Note the `testing.txt` file is still there.

10. The previous step is an important lesson. If you make changes to your instance (*e.g.*, installing software, creating files, etc.), and terminate it, then the state will not be saved. You can instead "stop" the instance, which will save the state.

11. However, you still run into the risk of Amazon's physical servers crashing (or someone on your team terminating an instance by accident!), which would still erase all state in a stopped instance.

12. To circumvent this problem, you should install all the software you need, and get the machine to a state you feel comfortable with. Then you need to *Register* your AMI. We'll skip this step for now. Let's install some crucial software.

13. Now you can treat this cloud instance like your own private sandbox, and start playing around with it (there's nothing to break, because you can always revert back to the most recently registered AMI!). You may have noticed that the instance is quite bare-bone; you are free to configure and install any software of your choice through the `yum` package manager.

5 Learn to Use the Linux Command Line

To administer your server, you need to be somewhat familiar with common commands in Linux. In the very least, you should commit the following to memory:

- Superuser: `sudo`, `su`.
- User management: `useradd`, `passwd`
- Filesystem navigation: `cd`, `ls`, `pwd`, `find`.
- Filesystem manipulation: `cp`, `mv`, `mkdir`, `rm`, `rmdir`.
- Change file/directory permissions: `chmod`, `chown`, `chgrp`.
- Software package management: `yum`
- Process management: `ps`, `top`, `kill`
- Open files (read-only): `less`, `more`, `tail`; search within files using `grep`.
- Edit files: Start with the simple `nano` editor. More advanced editors include `emacs` and `vi`.
- Networking: `ssh`, `wget`
- Accessing documentation: `man`
- Get comfortable with your shell. This CentOS AMI defaults to the commonly-used `bash` (bourne-again shell). `bash` offers some nice features, such as:
  - The tilde character $\sim$ always stands for your home directory. For instance, you can navigate to your home directory with: `cd` $\sim$.
  - Tab completion: Try reading some file in your current directory with `less`. Type: `less` followed by the first couple of letters in your filename, then hit `<tab>`. If the filename is unambiguous, it the shell will auto-complete. If there is ambiguity, hit `<tab>` twice in quick succession, and it will show all options matching the letters you've entered.
  - Your command history can be accessed by hitting the up-arrow.

6 Installing Required Software

1. It is assumed that you are relatively familiar and comfortable with the Linux command line after the previous project.
2. Previously, I had you choose a specific AMI to boot up. What I didn't tell you is that the AMI is an image running CentOS 6, a popular and free distribution that spawned off Fedora Linux. You may have already gotten familiar with using `yum` to manage your software packages.
3. Using `yum`, install the Apache Web Server. Its package name is `httpd`.
4. After installation, start the web server: `sudo /sbin/service httpd start`

7 Getting to Know the Apache Web Server

1. To check if your installation of apache was successful, point your web browser to the hostname or IP address of your EC2 instance. On success, you should get a generic apache page saying something to the effect of `It Worked!`, on failure, your browser will probably time-out or tell you the page does not exist. *Do not proceed further* until you ensure apache is running properly. Check with me if you have questions.
2. The following exercises must be done as the `root` user, or with `sudo`.
3. Before starting, you need to know the following commands.
   - To start apache: `sudo /sbin/service httpd start`
   - To stop apache: `sudo /sbin/service httpd stop`
   - To restart apache: `sudo /sbin/service httpd restart`

   Your server should be running by default.
4. The behavior of your apache web server is largely controlled by its configuration file. On some systems, it is called `httpd.conf`, and on others, `apache.conf`. Find this file (see: `find` command) on your server, and open it up for reading (see: `less` command).
5. Look through this file. You'll quickly notice that any line starting with # symbol is a comment. Read through all the comments for each configuration key and value. Pay particular attention to the following keys, and try to understand what their values mean. If you have questions/doubts, ask me. Note that any change to the config file must be followed by an apache restart for it to take effect.
   - `ServerRoot`
   - `Listen`
   - `DocumentRoot`
   - `DirectoryIndex`
   - `ErrorLog` and `LogFormat`
6. You should now be able to answer the following questions:
   (a) What command can you type into the terminal to check if apache is current running?
   (b) What is the difference between `ServerRoot` and `DocumentRoot`?
   (c) What port does your web server listen to for HTTP connections from browsers?

(d) In what directory do you need to place all of your HTML and PHP documents for apache to serve them up? Test by navigating to this directory, and if your hunch is right, you should find that *It Worked!* HTML document.

(e) What file contains all the traffic logs? What about error logs?

(f) What is a directory index file? Why would it be nice to *have* one in each directory?

(g) How do you give every user on your Linux machine their own web space? Look into `public_html`

(h) How do you create a password protected directory (like I did for the class notes)? Look into `htpasswd` and `.htaccess files`.

(i) When the browser tries to access a page that doesn't exist, the HTTP protocol issues a 404 error code. There's a way to send the browser to a particular document upon issuing this error. How? Look into `ErrorDocument` in the config file.

(j) Apache implements the HTTP protocol. The protocol is extremely simple, with just a few commands. What is the difference between the `GET`, `POST`, and `HEAD` commands?

7. After getting to know apache a little better, it's time to create and serve up your first web page:

   - Navigate to the `DocumentRoot` directory, create a new file called `groupN.html`, where `N` is your group number.
   - Create a simple web page with all your names on it, and answers to all the questions above.
   - You may (but you're not required to) add some design and colors to your page using appropriate CSS elements.

8 File Uploads

1. **WARNING!!** I do not recommend you writing code directly on the EC2 server. Use version-control tools (git, cvs, svn) with your team members as you would usually, and upload the files to your server when they're ready for production. Good file upload tools:

2. MacOS: Get Cyberduck (free). Open a new connection, then choose the SFTP (SSH) protocol. Use the ec2-user with no password. On the bottom, click "Use public key authentication," and navigate to your private key.

3. Windows: Get WinSCP (free)

4. Linux: Use the `scp` command

9 Creating an AMI

1. After you've install all these tools, you should register your AMI. After all that work you put in, it'd be a real shame if you accidentally terminated your instance (which causes it to lose state), and had to start all over!

2. Creating your AMI allows you to essentially generate your own image. Here's the idea. Get your server to a stable state where all the necessary software have been installed and configured. Then the idea is to take a *snapshot* of your server's state, so that whenever you boot up a new instance using this snapshot (i.e., image), it will start right back from where you left off.

3. With your instance running, right-click on it and choose "Create Image (EBS AMI)." This can take some time: think about it, Amazon needs to take the entire operating system's state and save/compress it into a single file. After this is done, you must write down your new AMI image ID. It should now show up under the "Images" tab when you select the "My AMIs" filter.

Submission and Demo

- Make sure your instance is registered with a desired state.
- Write me on slack with your new AMI's ID so that I can start it up myself and check your work.
- I set up a meeting with each team separately for a demonstration. Every member should be prepared to answer questions about EC2 and demonstrate using Linux commands.

Grading

- **[5pts]** You have customized and registered an AMI.
- **[5pts]** The apache web server is installed, up and running.
- **[20pts]** You have created an HTML file with the correct answers to the questions posed in Section 7. 2pts per question.
- **[20pts]** All members demonstrate basic proficiency with Linux commands during team demo.