Using Node on the Engineering Servers

Intro

OSU Engineering IT services provide servers where you can run your code and host web-pages for this course.

Step 1: OSU VPN

The first thing to do is to get on to *the OSU VPN*. If you are not connected to the VPN you will not be able to access your site. You can find information on connecting to the VPN here (http://oregonstate.edu/helpdocs/osu-applications/offered-apps/virtual-private-network-vpn).

Step 2: SSH into Flip

After you are logged into OSU VPN, you need to SSH into one of Engineering's FLIP servers using a tool like Putty (Windows) (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html) or Terminal (Mac/Linux). You might have done this for one of your earlier classes, but if not, OSU EECS IT Services (http://it.engineering.oregonstate.edu/software) provides instructions here (http://it.engineering.oregonstate.edu/accessing-unix-server-using-putty-ssh). Instead of logging into `access.engr.oregonstate.edu` enter `flip3.engr.oregonstate.edu` in the "Host Name (or IP address)" field. You can also use `flip2.engr.oregonstate.edu` or `flip1.engr.oregonstate.edu` (choose one and preferably stick to it for the entire course). You need to use your ONID username and password for the SSH.

Step 3: Getting Files

You should make a new directory in your OSU engineering space. Then from within that directory run the command `git clone https://github.com/wolfordj/CS290-Server-Side-Examples.git ./`. This will clone the class files from the class Git repository into that directory.

Step 4-1: Making Required Changes

Next you will need to tweak some files to work with the engineering setup because you are now using shared resources that others can access.

In particular there is a file called `dbcon.js.template`. Rename this to `dbcon.js`. Note that `dbcon.js.template` is present in multiple directory. You need to be in the `$PROJECT/diagnostic/` directory, where `$PROJECT` is the place where git clone was run

Then you need to edit the file, it should look like the following code with the square brackets and their contents replaced with the appropriate content:

```
var mysql = require('mysql');
var pool = mysql.createPool({
  connectionLimit : 10,
  host            : 'classmysql.engr.oregonstate.edu',
  user            : '[cs290_yourengrusername]',
  password        : '[last-4-digits-of-your-osu-id-number]',
  database        : '[cs290_yourengrusername]'
});

module.exports.pool = pool;
```

These are the default settings for all students who were enrolled when the term started. An example might look like this:

```
var mysql = require('mysql');
var pool = mysql.createPool({
  connectionLimit : 10,
  host            : 'classmysql.engr.oregonstate.edu',
  user            : 'cs290_smithj',
  password        : '1234',
  database        : 'cs290_smithj'
});

module.exports.pool = pool;
```

With that file renamed with the proper info added we are almost ready to go.

Your flip account may need to manually install Express, Express-handlebars, mysql if you have not done it before. Under the diagnostic folder, run `npm install express`, `npm install express-handlebars`, `npm install mysql` will install these for you.

Step 4-2: Changing your MySQL Password

In this step, you may want to change your MySQL password to be something other than the last 4 digits of your ID to make it more secure. You can skip this step if you just want to use the default password. You can access the administration panel at https://tools.engr.oregonstate.edu/phpMyAdmin/index.phpu (https://tools.engr.oregonstate.edu/phpMyAdmin/index.php) where you can change your password. Please use "cs290_your ONID user name" as user name and the last four digits of your OSU ID number as the initial db password, and choose server name 'classmysql'. *This site can only be accessed while you are VPNed into the OSU network*. Note that if you change the password, you need to make the change in the `dbcon.js` as well. This can also be useful for troubleshooting database problems as it is a graphical interface that will let you inspect the contents of your database when you might otherwise be unable to access it. If you forget the password after you change it, please email the IT support and they will reset to the default.

Step 5: Ports and Persistence

Since this is running on a shared machine, everyone cannot use port 3000. If you look at the `diagnostic.js` file you will see that the port 3000 was replaced with `process.argv[2]`. This means it will take the argument immediately after the filename when you start Node.js and use that as the port. So running `node diagnostic.js 5345` will start node running on port 5345. Everyone will need to use a unique port otherwise you will get an error that the port is in use. Try a number between 3000 and 65535 till you find one that works for you (Hint: Try some random permutation of the last 4 or 5 digits of your OSU ID).

Step 6: forever

Finally is the topic of persistence. If you want to use `forever` there will be a few changes. You need to install it on a per application basis. So to install it for the `diagnostic` folder, navigate to in that folder and run `npm install forever`. Note that this does *not* use the global flag.

Then you will need to run forever by accessing the forever binary directly. So from within the diagnostic directory you would run `forever` using the command `./node_modules/forever/bin/forever`. So when before you would run `forever start diagnostic.js`, now you will run `./node_modules/forever/bin/forever start diagnostic.js 5345`. So we specify the path to the forever binary and also add a port number as an argument. You could view the page being served by visiting `http://flipX.engr.oregonstate.edu:5345` while you are VPNed into the OSU network.

The Many Flips

And as a closing note, if you log into access.engr.oregonstate.edu you will *randomly* be put on flip1, flip2 or flip3. You can see which flip you are using the command `hostname` then you can switch flips by using the command `ssh flipX` where X is 1, 2 or 3. You need to be sure to log into the same flip every time because the node instance will only be running on one of them. So if you ever run `forever list` and don't see your instance running it means you are probably on the wrong flip.

Activity

You should be able to run the diagnostic and access the page it serves while VPNed onto the server. It will display a message that "MySQL is working". That means your node is set up successfully.

Review

The steps on this page are essential prerequisites for all other assignments. You might return to it from time to time so be sure to bookmark it. The standard sequence of steps to be followed are: using the VPN, SSH-ing into FLIP, making sure you are on the right flip, forever running from the application directory and configuring MySQL. If you have trouble with this section, be sure to consult Piazza for relevant discussions.