Curriculum Runbook

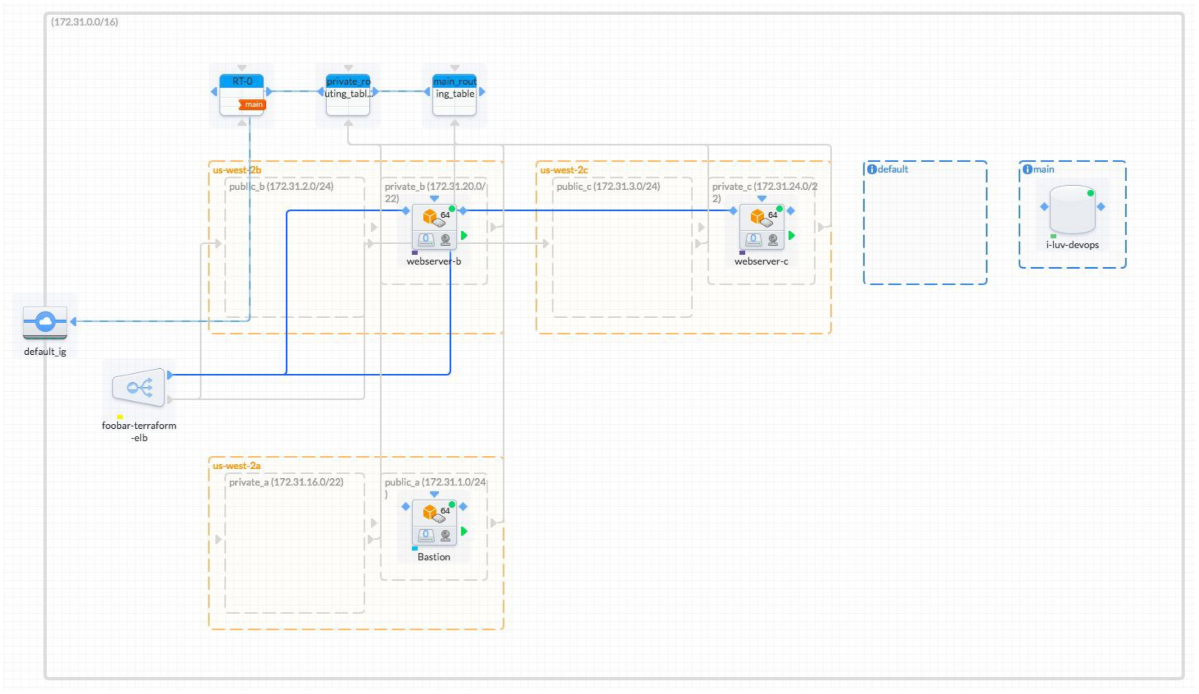
Short Description

The Curriculum Runbook displays the classes available at CSUN. We do this by dumping a database and web services onto dedicated servers.

Required Software

The software required are Amazon Web Services, Ansible, Terraform, AWS CLI, Git, OpenSSH, MariaDB (community-fork of the MYSQL relational database software). For the web servers, you'd need nginx, composer, php54, php54-fpm, php54-ldap, php54-mbstring, php54-mcrypt, php54-mysqlnd, php54-pdo

Architecture Diagram



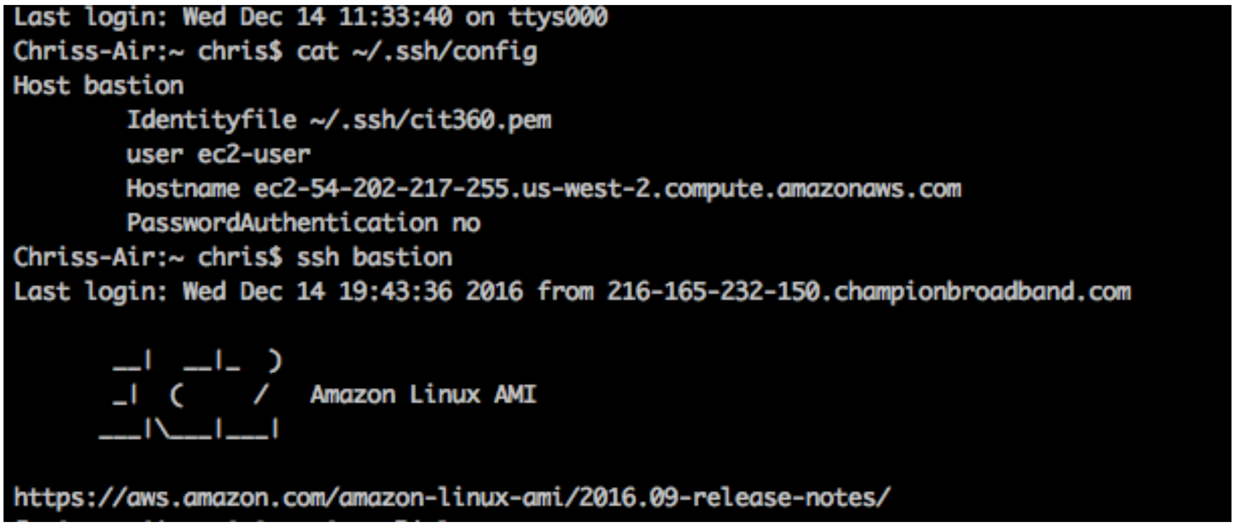
Deployment

After creating the virtual infrastructure via Terraform, it is now time to SSH into our Bastion host. For convenience, it is recommended to create an "ssh_config" or "config" file in our .ssh folder. For reference, it should be as followed:

```
Host bastion
    Identityfile ~/.ssh/cit360.pem
    user ec2-user
    Hostname *insert AWS EC2 instance*
```

PasswordAuthentication no

We then run $ ssh bastion It will ask for permission to add to known hosts, then type yes.



```
Last login: Wed Dec 14 11:33:40 on ttys000
Chriss-Air:~ chris$ cat ~/.ssh/config
Host bastion
        Identityfile ~/.ssh/cit360.pem
        user ec2-user
        Hostname ec2-54-202-217-255.us-west-2.compute.amazonaws.com
        PasswordAuthentication no
Chriss-Air:~ chris$ ssh bastion
Last login: Wed Dec 14 19:43:36 2016 from 216-165-232-150.championbroadband.com

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2016.09-release-notes/
```

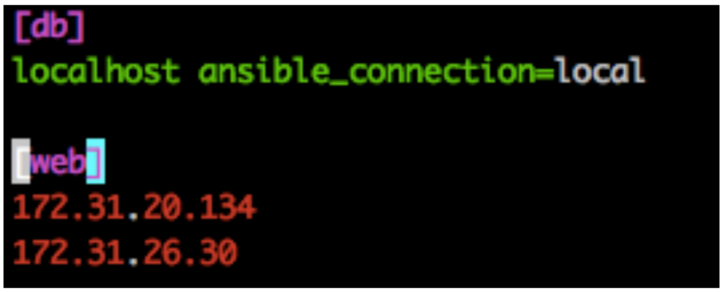
We also need to import the pem file we created with cit360 public key using secure copy on mac. To run it, we use

$ scp ***-i*** *~/path/to/pem/file ~/path/to/pem/file ec2-user@public-dns-hostname:~/path/to/.ssh*

because we'll reuse the key to connect into our webservers. Then we installed ansible on our Bastion host . We can install our ansible folders one of two ways but for simplicity we'll use secure copy with a directly flag*:*
$ scp ***—I*** *~/path/to/pem/file -r ~/path/ansible ec2-user@public-dns-hostname:~/*

When everything is finished uploading, installing ansible is done through *$ sudo pip install ansible*.

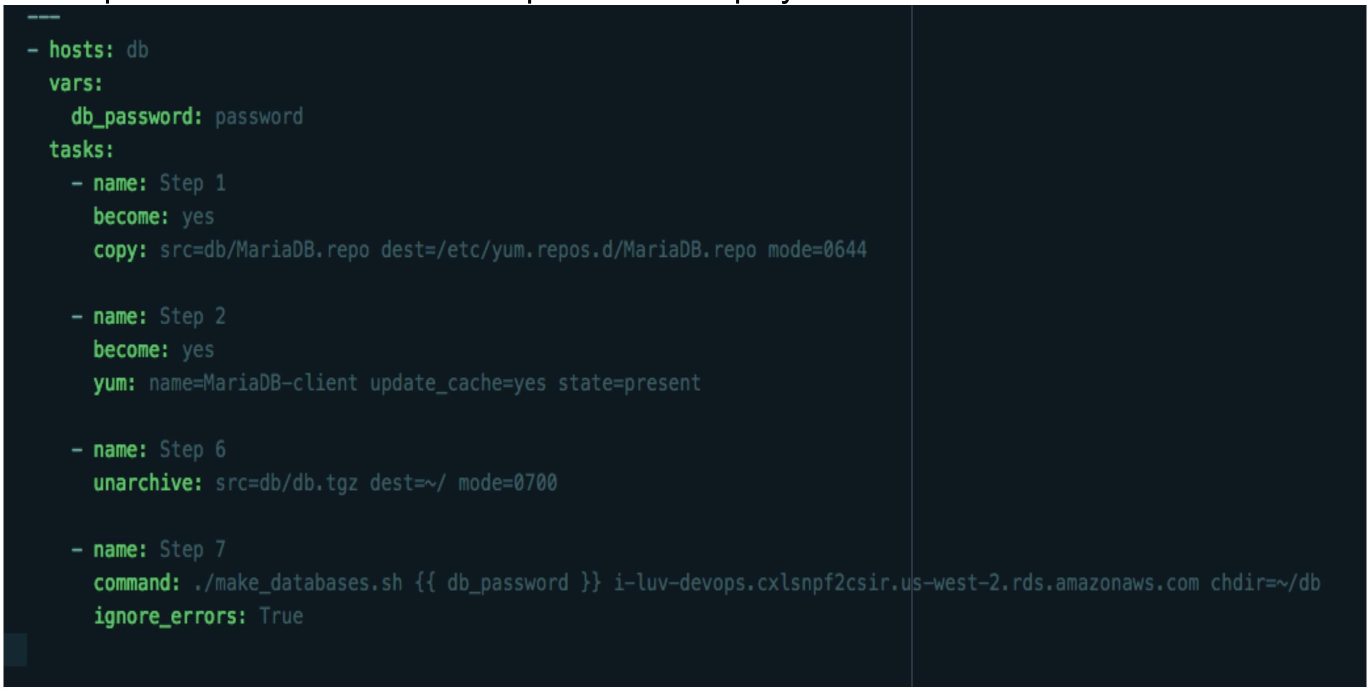
Then we update the hosts.ini file.
The hosts.ini file allows Ansible to know which nodes to run the playbooks off of and should be configured as shown. However, your IP addresses for [web] will need to be changed for you.



```
[db]
localhost ansible_connection=local

[web]
172.31.20.134
172.31.26.30
```

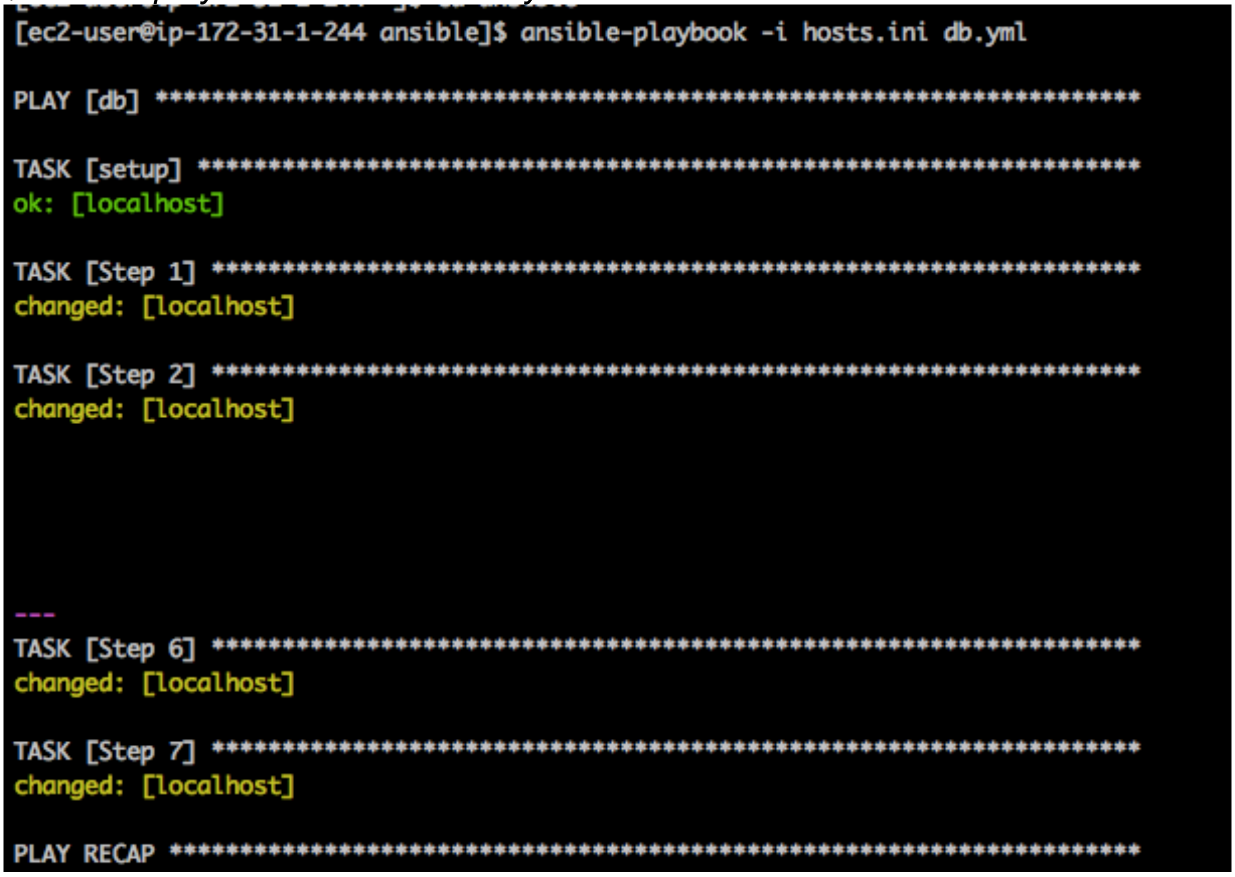
Next we update the playbooks
Because we just uploaded everything we're using into the bastion host, we edit our *db.yml* playbook to only download the MariaDB client instead of server. We also take away some steps to use the only required steps in the playbook

It is important to add the RDS endpoint into the playbook when needed



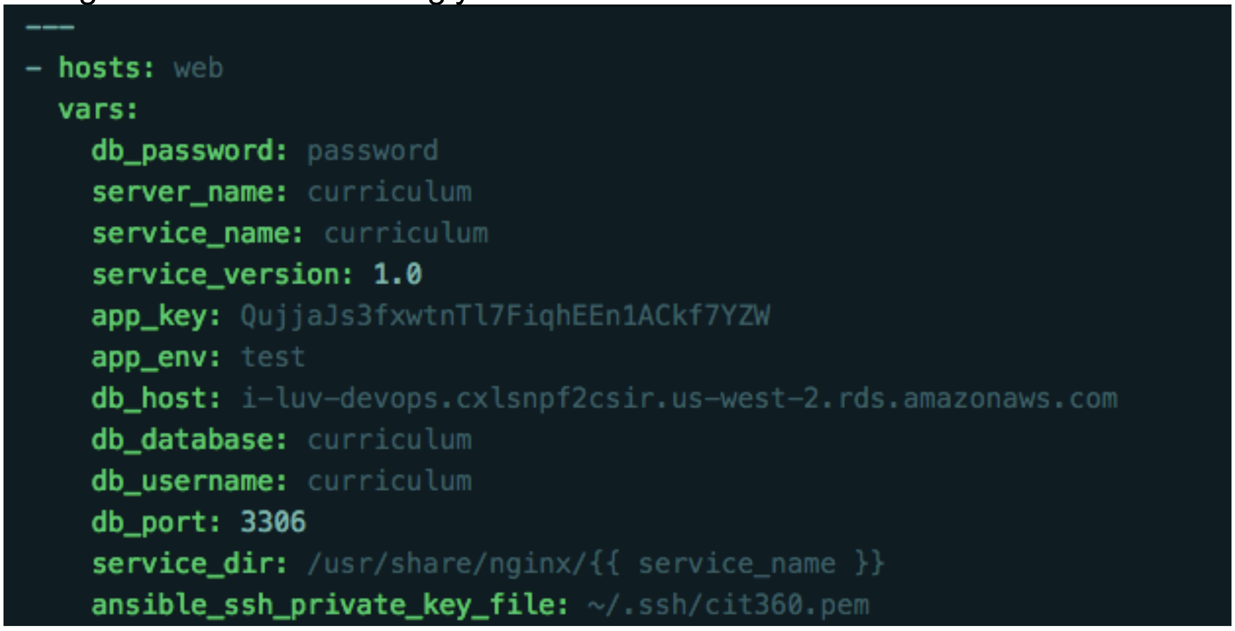
```
---
- hosts: db
  vars:
    db_password: password
  tasks:
    - name: Step 1
      become: yes
      copy: src=db/MariaDB.repo dest=/etc/yum.repos.d/MariaDB.repo mode=0644

    - name: Step 2
      become: yes
      yum: name=MariaDB-client update_cache=yes state=present

    - name: Step 6
      unarchive: src=db/db.tgz dest=~/ mode=0700

    - name: Step 7
      command: ./make_databases.sh {{ db_password }} i-luv-devops.cxlsnpf2csir.us-west-2.rds.amazonaws.com chdir=~/db
      ignore_errors: True
```

Then we run ansible playbook on our bastion host.
$ ansible-playbook –i hosts.ini db.yml



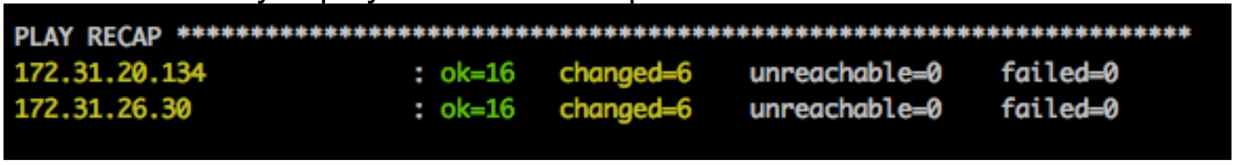
```
[ec2-user@ip-172-31-1-244 ansible]$ ansible-playbook -i hosts.ini db.yml

PLAY [db] ***************************************************************

TASK [setup] ************************************************************
ok: [localhost]

TASK [Step 1] ***********************************************************
changed: [localhost]

TASK [Step 2] ***********************************************************
changed: [localhost]




---
TASK [Step 6] ***********************************************************
changed: [localhost]

TASK [Step 7] ***********************************************************
changed: [localhost]

PLAY RECAP **************************************************************
```

Next we update our web.yml playbooks to use our db host address, our cit360.pem file for authentication, install newer php 5.4 on our webservers, and change directories accordingly



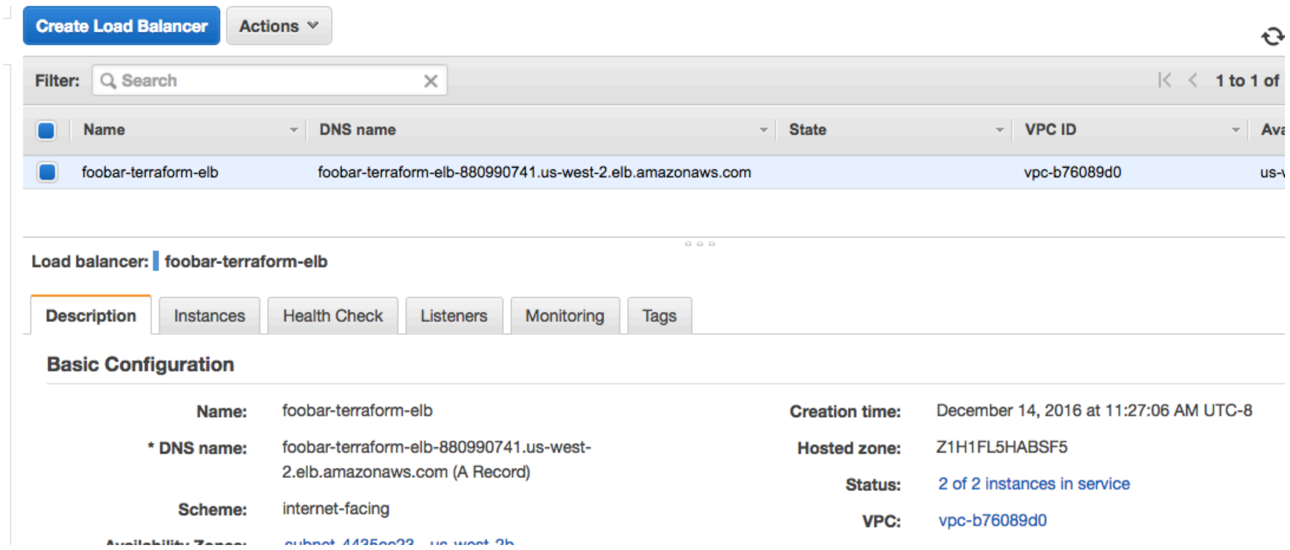
```
---
- hosts: web
  vars:
    db_password: password
    server_name: curriculum
    service_name: curriculum
    service_version: 1.0
    app_key: QujjaJs3fxwtnTl7FiqhEEn1ACkf7YZW
    app_env: test
    db_host: i-luv-devops.cxlsnpf2csir.us-west-2.rds.amazonaws.com
    db_database: curriculum
    db_username: curriculum
    db_port: 3306
    service_dir: /usr/share/nginx/{{ service_name }}
    ansible_ssh_private_key_file: ~/.ssh/cit360.pem
```

We run the web.yml playbook and see it passed



```
PLAY RECAP *********************************************************************
172.31.20.134              : ok=16   changed=6    unreachable=0    failed=0
172.31.26.30               : ok=16   changed=6    unreachable=0    failed=0
```

Then we look up our load balancer on the AWS console and see if our instances our healthy, under the EC2 Dashboard



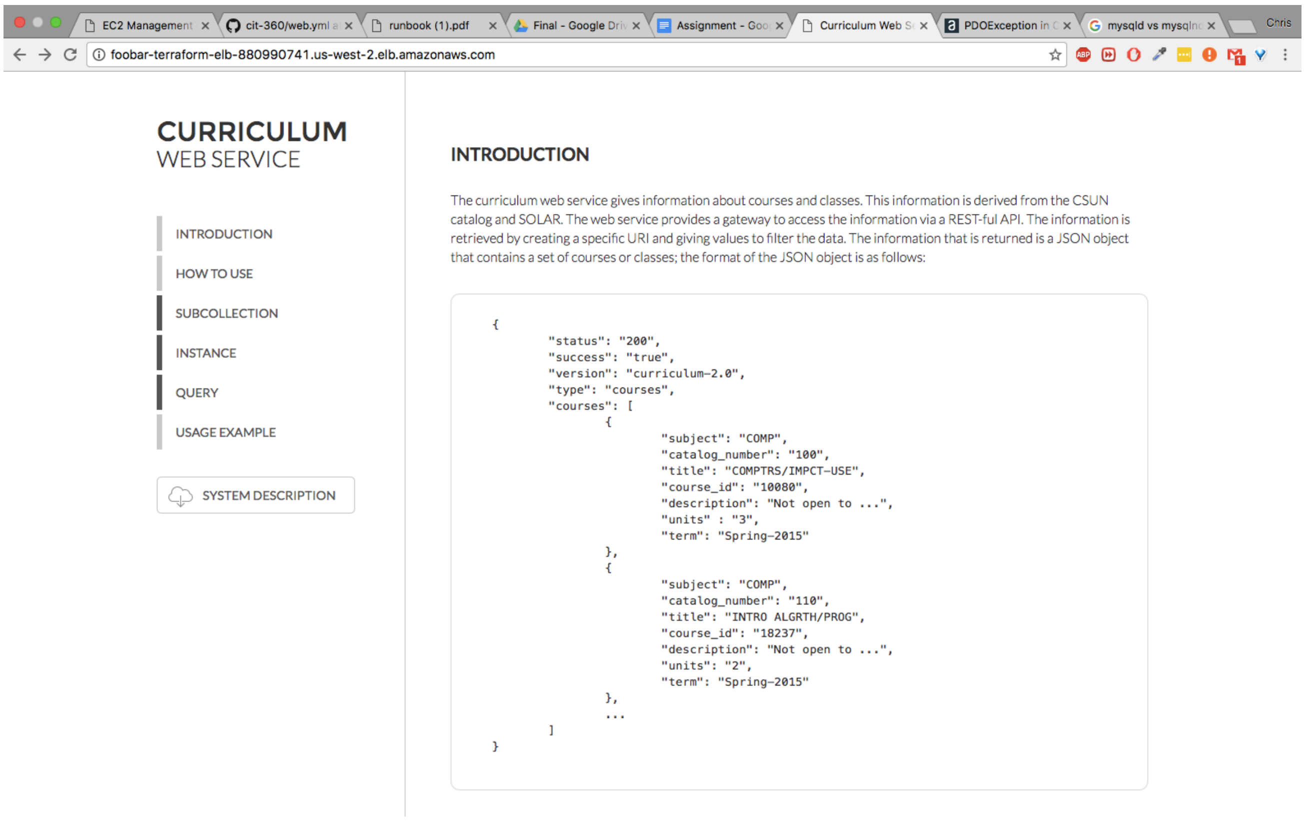
Name	DNS name	State	VPC ID	Ava
foobar-terraform-elb	foobar-terraform-elb-880990741.us-west-2.elb.amazonaws.com		vpc-b76089d0	us-

Load balancer: foobar-terraform-elb

Basic Configuration

Name:	foobar-terraform-elb	Creation time:	December 14, 2016 at 11:27:06 AM UTC-8
* DNS name:	foobar-terraform-elb-880990741.us-west-2.elb.amazonaws.com (A Record)	Hosted zone:	Z1H1FL5HABSF5
Scheme:	internet-facing	Status:	2 of 2 instances in service
		VPC:	vpc-b76089d0

We type the address for our load balancer to go see the contents of our website



INTRODUCTION

The curriculum web service gives information about courses and classes. This information is derived from the CSUN catalog and SOLAR. The web service provides a gateway to access the information via a REST-ful API. The information is retrieved by creating a specific URI and giving values to filter the data. The information that is returned is a JSON object that contains a set of courses or classes; the format of the JSON object is as follows:

```
{
        "status": "200",
        "success": "true",
        "version": "curriculum-2.0",
        "type": "courses",
        "courses": [
                {
                        "subject": "COMP",
                        "catalog_number": "100",
                        "title": "COMPTRS/IMPCT-USE",
                        "course_id": "10080",
                        "description": "Not open to ...",
                        "units" : "3",
                        "term": "Spring-2015"
                },
                {
                        "subject": "COMP",
                        "catalog_number": "110",
                        "title": "INTRO ALGRTH/PROG",
                        "course_id": "18237",
                        "description": "Not open to ...",
                        "units": "2",
                        "term": "Spring-2015"
                },
                ...
        ]
}
```

Issues

Title: *Permission Denied SSH Into Instance*
Description: *Unable to SSH into instance with the permission denied error.*
Remediation Steps:
Your private key file must be protected from read and write operations from any other users. If your private key can be read or written to by anyone but you, then SSH ignores your key and you see the following warning message below.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@
Permissions 0777 for '.ssh/my_private_key.pem' are too open.
It is required that your private key files are NOT accessible by others. T
his private key will be ignored.
bad permissions: ignore key: .ssh/my_private_key.pem P
ermission denied (publickey).
```

If you see a similar message when you try to log in to your instance, examine the first line of the error message to verify that you are using the correct public key for your instance. The above example uses the private key .ssh/my_private_key.pem with file permissions of 0777, which allow anyone to read or write to this file. This permission level is very insecure, and so SSH ignores this key. To fix the error, execute the following command, substituting the path for your private key file.

```
$ chmod 0400 .ssh/my_private_key.pem
```

Title: PDO Exception
Description: Pages/links not working properly
Remediation Steps: Ensure that php54-pdo and/or php54/mysqlnd are installed on the webservers

Make sure to type in the required modules in the *web.yml* playbook