**E91 Cloud Devops: Fall 2018**

**Assignment 5**

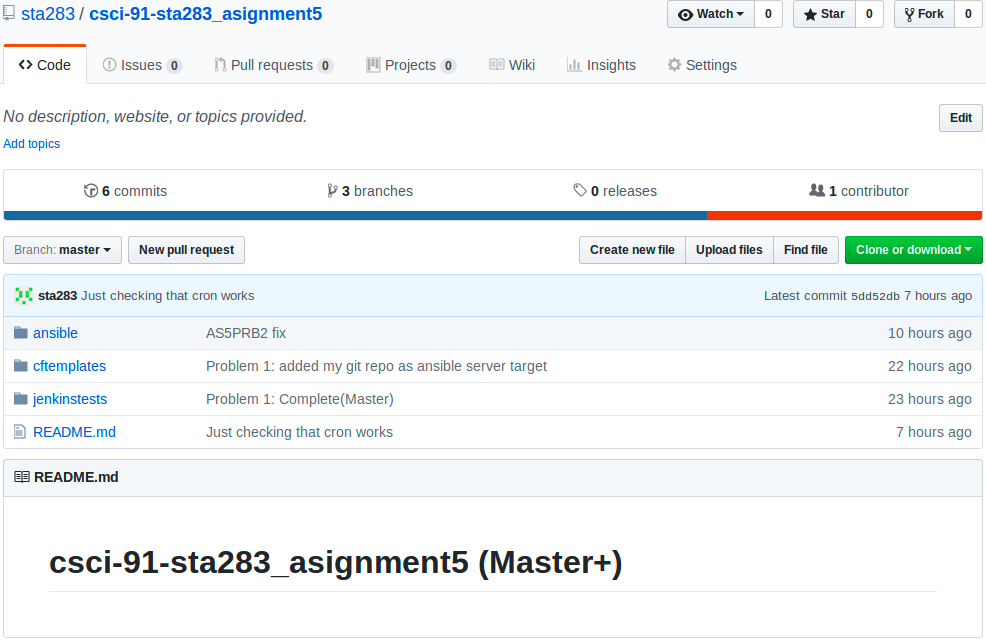
**Stephen Akaeze**

[**https://code.harvard.edu/sta283/csci-91-sta283\_asignment5**](https://code.harvard.edu/sta283/csci-91-sta283_asignment5)

Instruction Manual: <https://canvas.harvard.edu/courses/53026/files/6838145/download?verifier=sb3I0hyGCl8booSa2xovXS8a2f9ZMAc2Sahp4MBK&wrap=1>

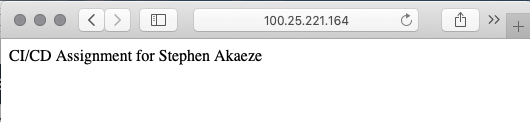
**Problem 1:**

1. As instructed by the manual, [csci-91-sta283\_asignment5](https://code.harvard.edu/sta283/csci-91-sta283_asignment5) repo was created comprising the master, dev and stage branches. Each branch is setup to contain the cftemplates, ansible and jenkinstests directories.
2. The ssh key pair was created using “ssh-keygen -t rsa” and saved in the local VM.
3. The ansible, cftemplates and jenkinstests directories were updated as instructed.
4. The CloudFormation templates were updated with the newly generated ssh key pair and the public key was added to my git repository
5. The AMI ami-058f0b6d904c90419 was solely used in my CloudFormation templates
6. The branches were pushed to update the remote repo at code.harvard.edu as shown below

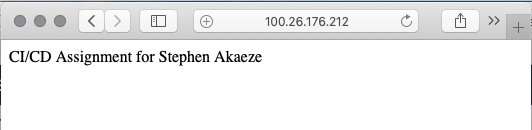


**Problem 2:**

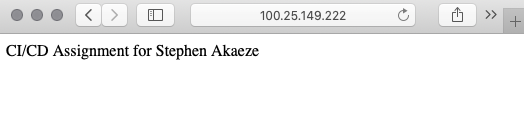
1. As instructed,
   1. My infrastructure (Ansible, Jenkins, Dev, Stage and Prod) was created using my CloudFormation templates (AnsibleServer.yml and AnsibleClient.yml files in the cftemplates directory).
   2. After creating my infrastructure, my ansible server could not clone my git repo due to private key issues. So, I manually deleted the ansible server’s faulty private key (/root/.ssh/id\_rsa) and copied the original key from my local VM to the ansible server. This fixed the problem.
   3. The ansible main host file was updated to reflect my infrastructure’s local IPs.
2. As instructed, all the checks and required files were updated. To verify, please view https://code.harvard.edu/sta283/csci-91-sta283\_asignment5
3. As instructed,
   1. The Dev, Stage and Prod EC2s cron jobs were setup using commands detailed at <https://code.harvard.edu/sta283/csci-91-sta283_asignment5/blob/master/Cron_setup.txt>
      1. Dev instance site



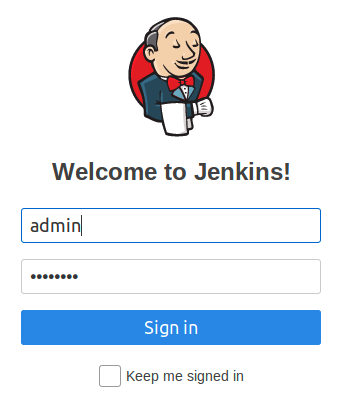
* + 1. Stage instance site

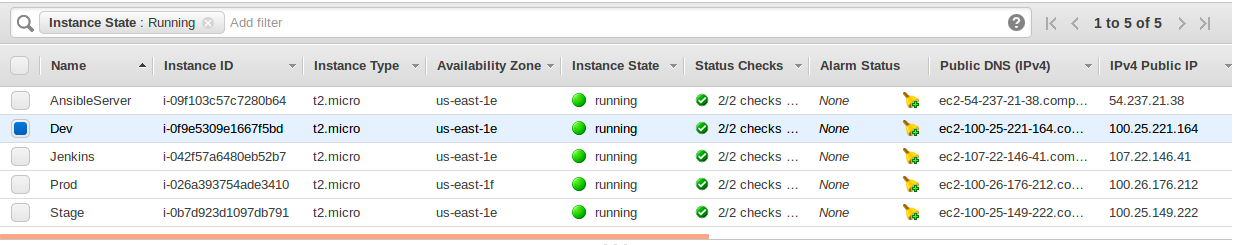


* + 1. Prod instance site



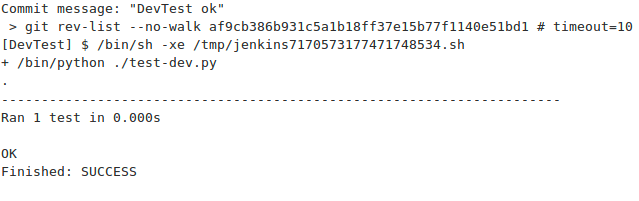
* 1. The Jenkins server was also configured as instructed as shown below



* 1. All Instances
     1. 

**Problem 3:**

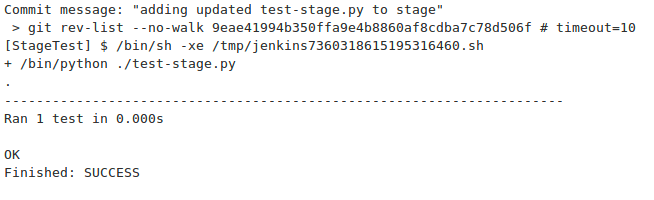
1. As instructed
   1. The Dev environment test job (DevTest) was setup and successfully executed every minute. Then, the “dev” branch index.html file was updated and was merged into the “stage” branch.
      1. The DevTest job



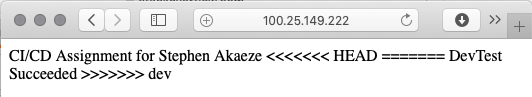
* + 1. The dev instance site output



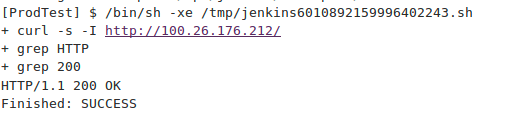
1. As instructed
   1. The stage environment test job (StageTest) was setup and successfully executed every minute. Then, the “stage” branch was merged into the “master” branch.
      1. The StageTest job



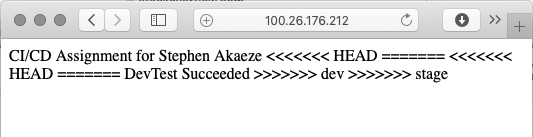
* + 1. The stage instance site output



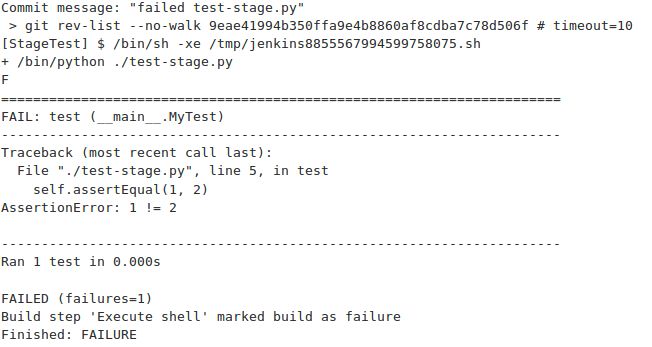
1. As instructed
   1. ProdTest was created as instructed and successfully executed every minute. Below are the results
      1. The ProdTest job



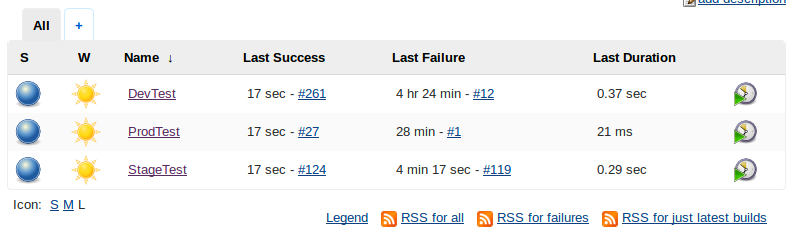
* + 1. The Prod instance site output



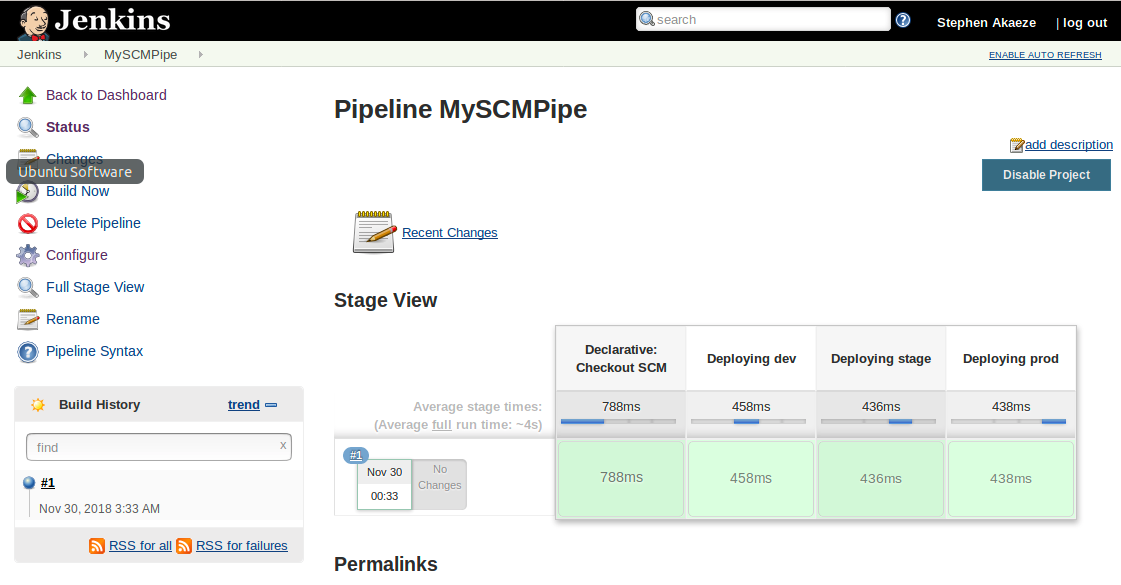
1. The information is provided above
2. After the “self.assertEqual(1,2)” update, the StageTest failed. All the previous outputs remained the same except the StageTest which failed as shown below



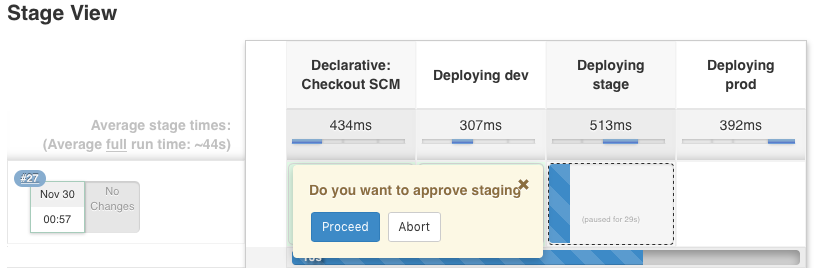
**Jenkins Console nippet**



1. Thoughts and Ideas
   1. If the index.html file is archived through the successful dev and stage, it becomes a fully and truly automated continuous integration and deployment setup. This can serve a notification feature to tell the developer about a successful CI/CD. This will minimize the need for a human intervention.
   2. Jenkins needs the ssh private key for the ansible server to send files to the respective servers.
2. Thoughts and ideas
   1. Yes, Jenkins can perform the push and merge after successful because it possesses the required ssh private key to perform the commands.
3. Thoughts and ideas
   1. Yes, Ansible can utilize webhooks to get notified on new code commits
4. As instructed, Jenpipe was created and successfully executed. It is located at <https://code.harvard.edu/sta283/csci-91-sta283_asignment5/blob/master/jenpipe>



1. As instructed Jenpipe2 was created and successfully executed. It is located at <https://code.harvard.edu/sta283/csci-91-sta283_asignment5/blob/master/jenpipe2>

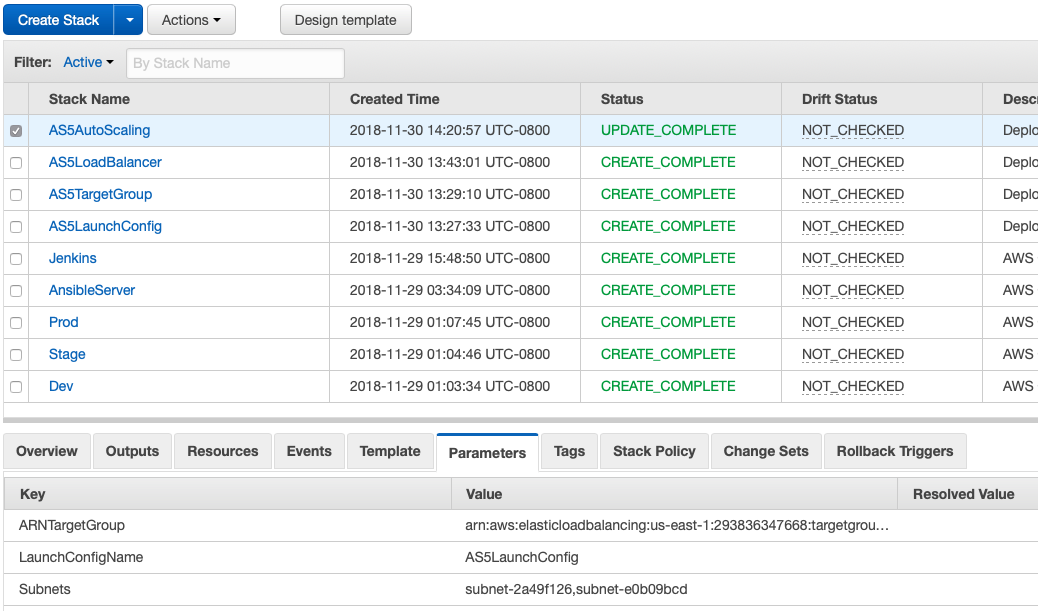


**Problem 4:**

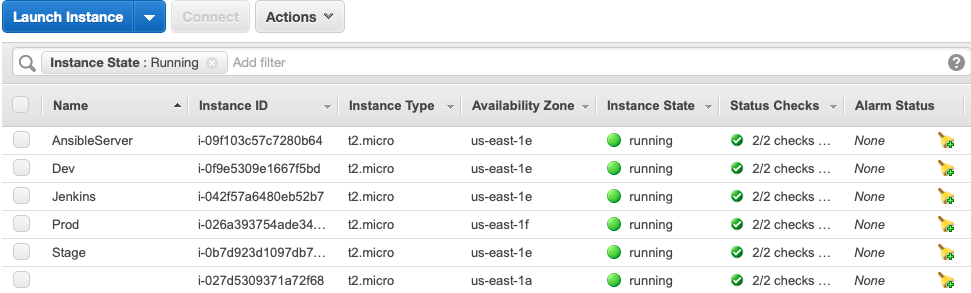
As instructed the environment was setup using the following files

1. **Launch config:** <https://code.harvard.edu/sta283/csci-91-sta283_asignment5/blob/master/cftemplates%E2%80%8B%E2%80%8B/AS5LaunchConfig.yml>
2. **Target group:** <https://code.harvard.edu/sta283/csci-91-sta283_asignment5/blob/master/cftemplates%E2%80%8B%E2%80%8B/AS5TargetGroup.yml>
3. **Load Balancer:** <https://code.harvard.edu/sta283/csci-91-sta283_asignment5/blob/master/cftemplates%E2%80%8B%E2%80%8B/ASLoadbalancer.yml>
4. **Auto Scaling:** <https://code.harvard.edu/sta283/csci-91-sta283_asignment5/blob/master/cftemplates%E2%80%8B%E2%80%8B/AS5AutoScale.yml>

Cloud formation snippet



Instance List for infrastructure



Load Balancer DNS name query

