

DevOps Engineering on AWS: Lab 6 - Application Pipeline - v1.5

==================================================================================================================

Using this command reference.

==================================================================================================================

1. Locate the section you need. Each section in this file matches a section in the lab instructions.

2. Replace items in angle brackets - < > - with appropriate values. For example, in this command you would replace the value - <JobFlowID> - (including the angle brackets) with the parameter indicated in the lab instructions:

elastic-mapreduce --list <JobFlowID>. You can also use find and replace to change bracketed parameters in bulk.

3. Do NOT enable the Word Wrap feature in Windows Notepad or the text editor you use to view this file.

++++1. Task: Initialize Continuous Integrations Framework

==================================================================================================================

1.2 Initialize Dependencies

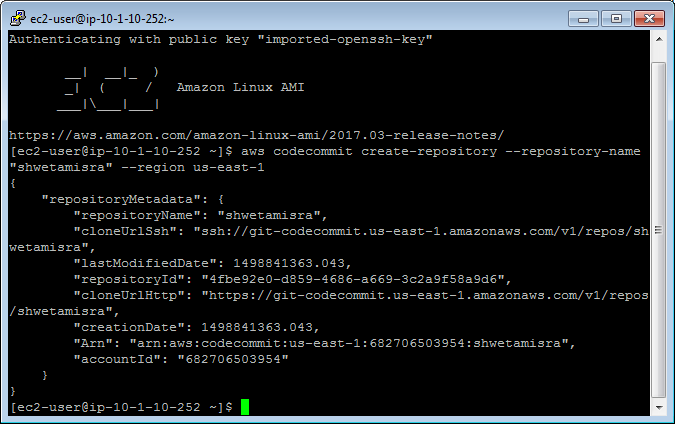
==================================================================================================================

1.2.2 Create the CodeCommit repository

aws codecommit create-repository --repository-name <repo name> --region us-east-1

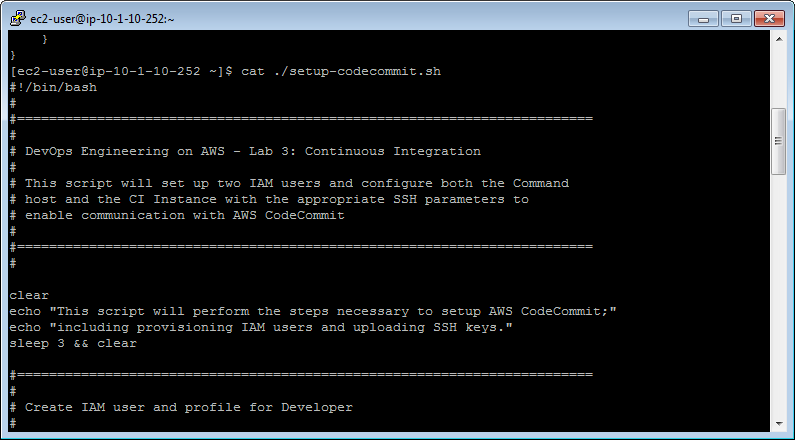
aws codecommit create-repository --repository-name "shwetamisra" --region us-east-1

1.2.4 View the script and initialize permissions for AWS CodeCommit



cat ./setup-codecommit.sh

./setup-codecommit.sh

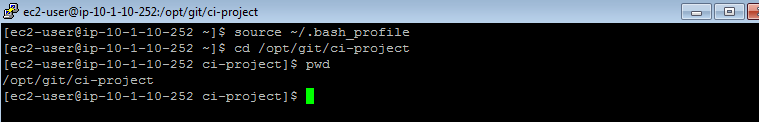


1.2.5 Reload the bash profile

source ~/.bash\_profile

1.2.6 Navigate to the ci-project directory

cd /opt/git/ci-project

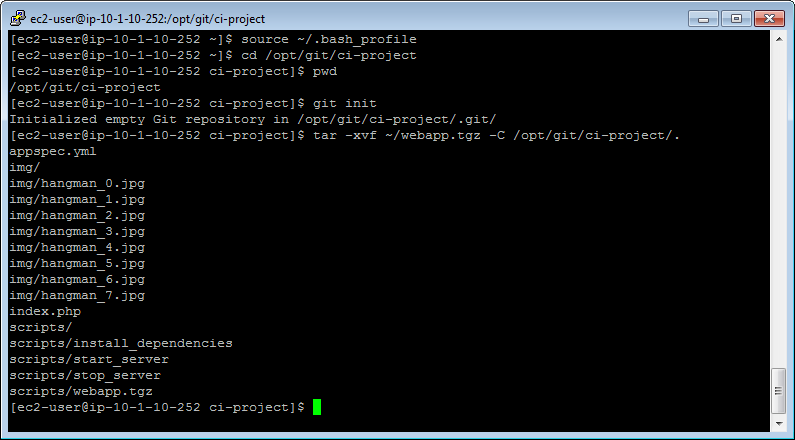


1.2.7 Initialize the local Git repository

git init

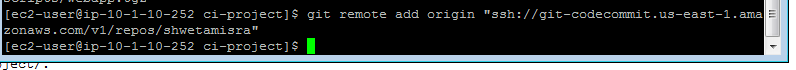
1.2.8 Extract the sample app binaries

tar -xvf ~/webapp.tgz -C /opt/git/ci-project/.



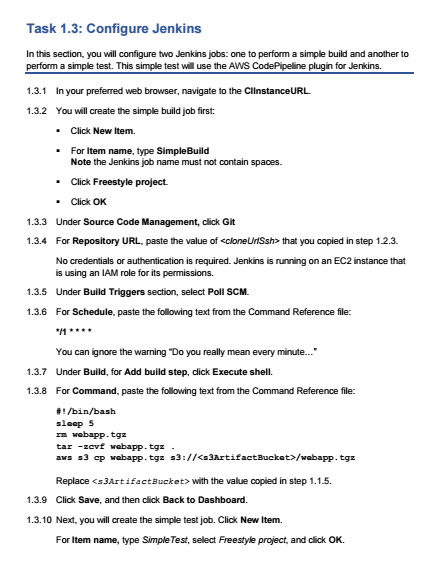
1.2.9 Register remote Git repo

git remote add origin <clonseUrlSsh>

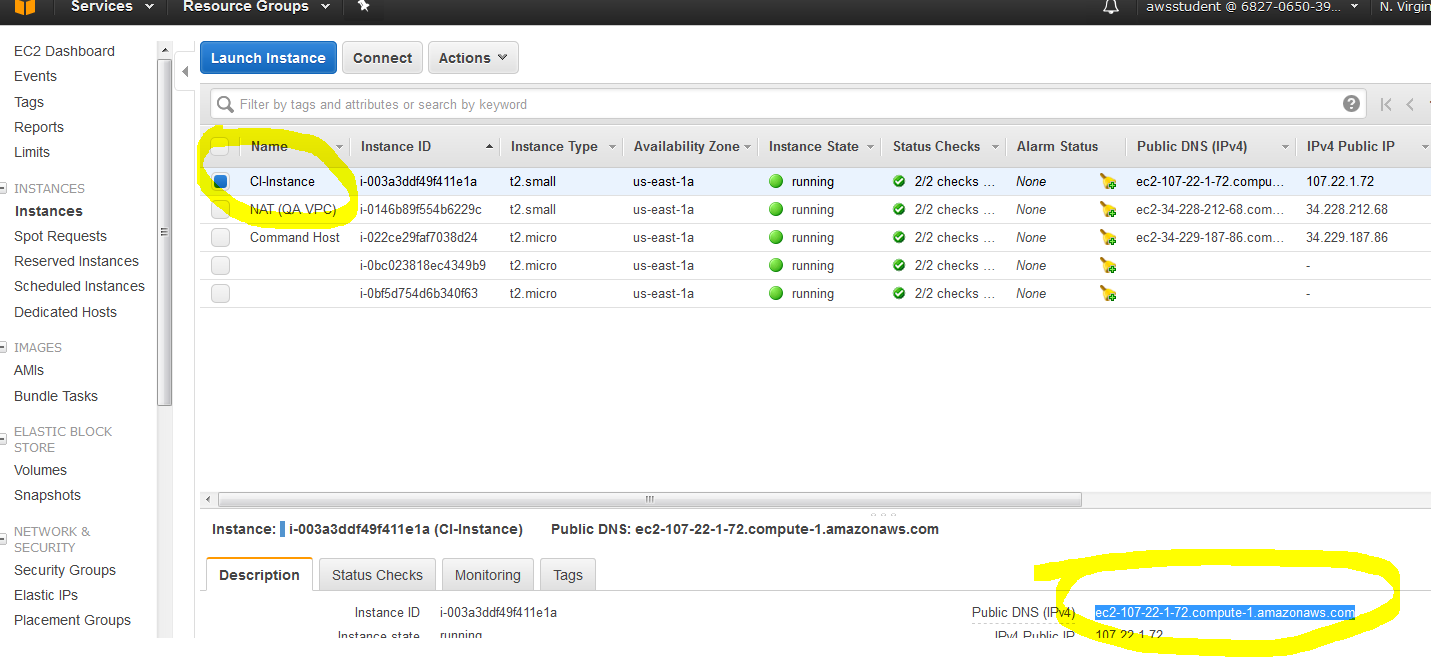


output:

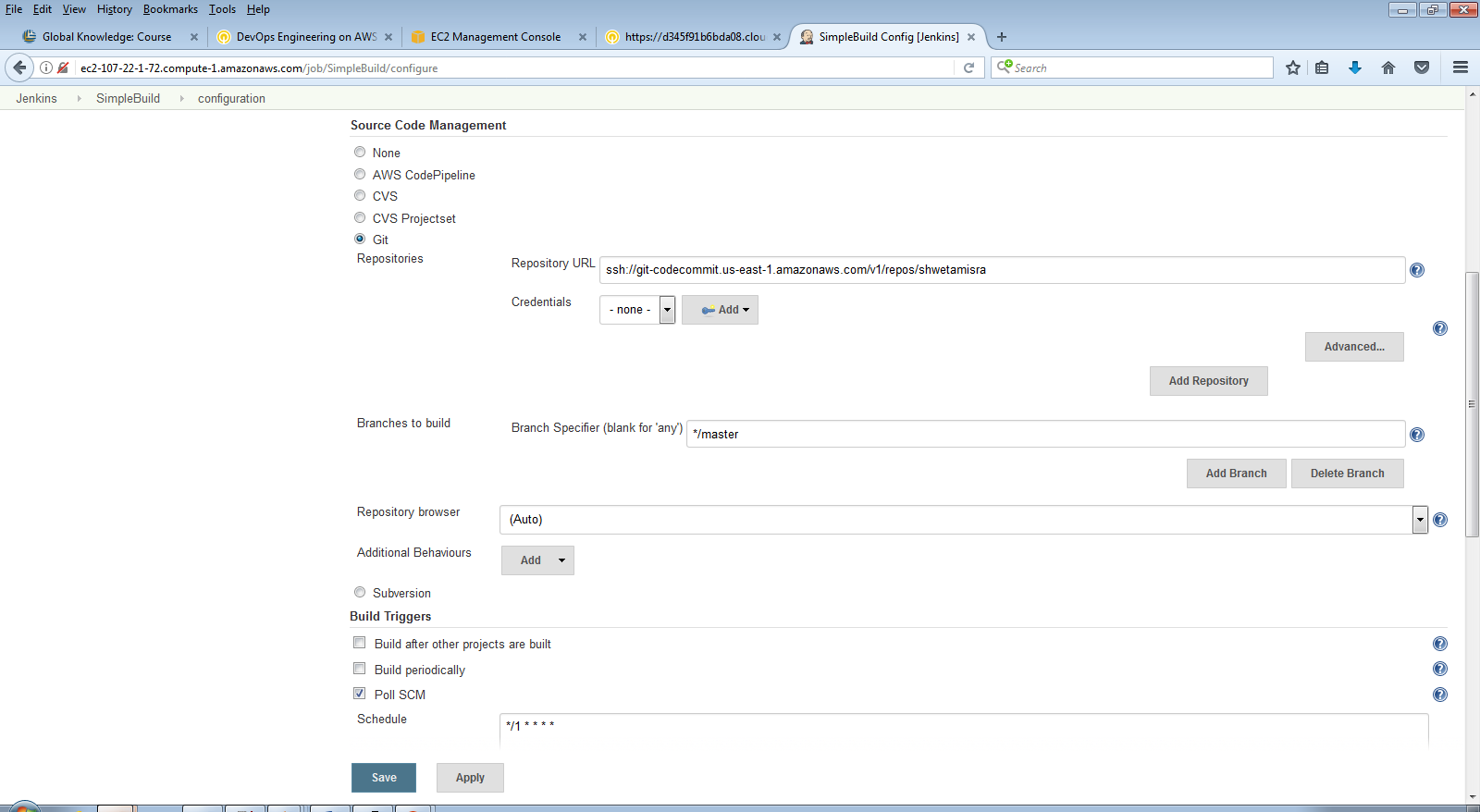
|  |  |  |  |
| --- | --- | --- | --- |
| CIInstancePublicIP | 107.22.1.72 | The public IP address of the CI / Jenkins instance |  |
| s3ArtifactBucket | qls-147617-57efcf940f0d7552-s3artifactbucket-1oho032vciz6w | S3 Bucket for Storing Deployment Artifacts |  |
| CIInstanceURL | <http://ec2-107-22-1-72.compute-1.amazonaws.com> | URL for the newly created Continuous Integrations instance |  |
| CommandHostPublicIP | 34.229.187.86 | The public IP address of the Command Host |  |
| CIInstancePrivateIP | 10.1.10.64 | The private IP address of the CI / Jenkins instance |  |
| QaElbEndpointUrl | <http://qls-14761-QaWebApp-1S539G9QJH74C-607031401.us-east-1.elb.amazonaws.com> | The URL for our Elastic Load Balancer. |  |

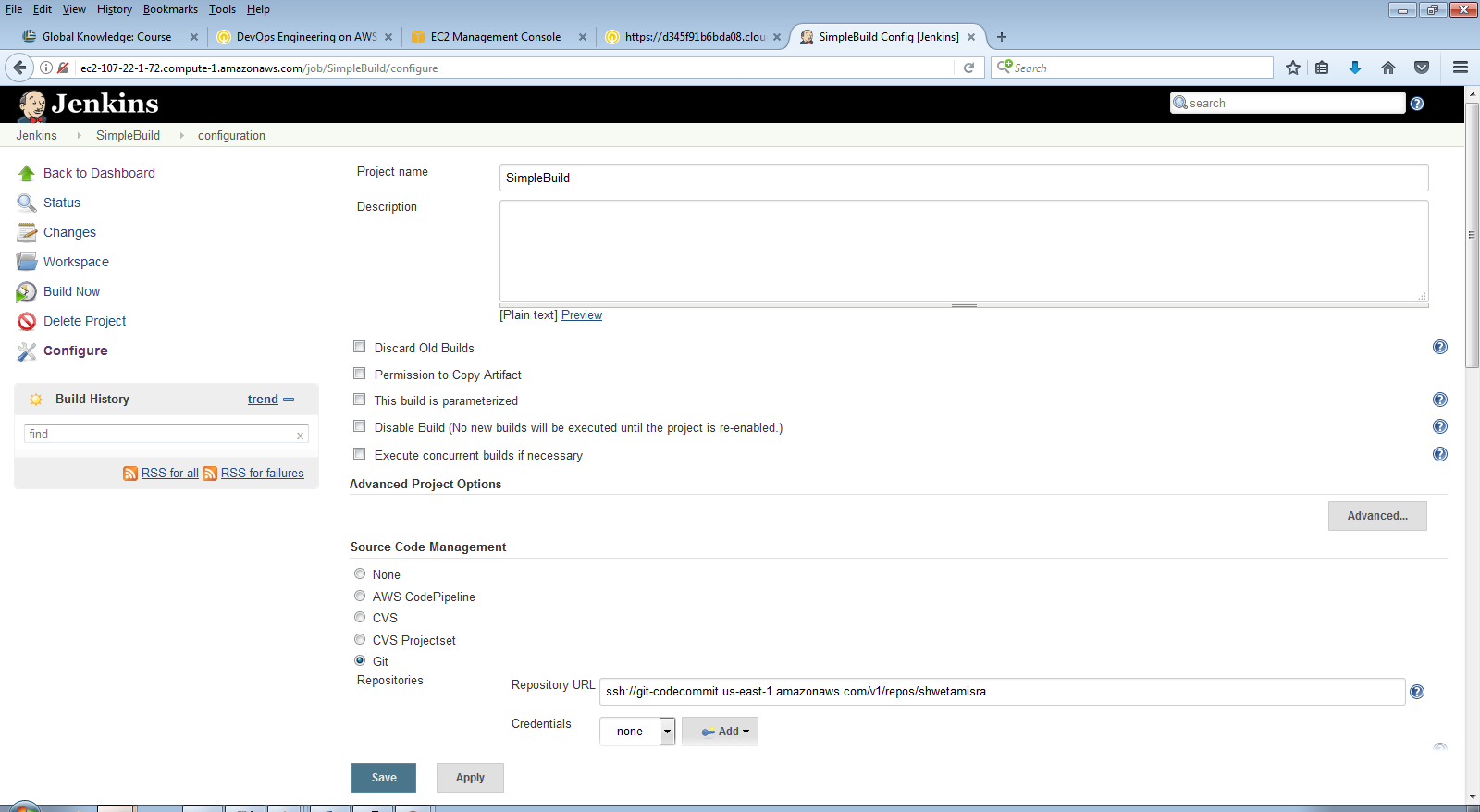


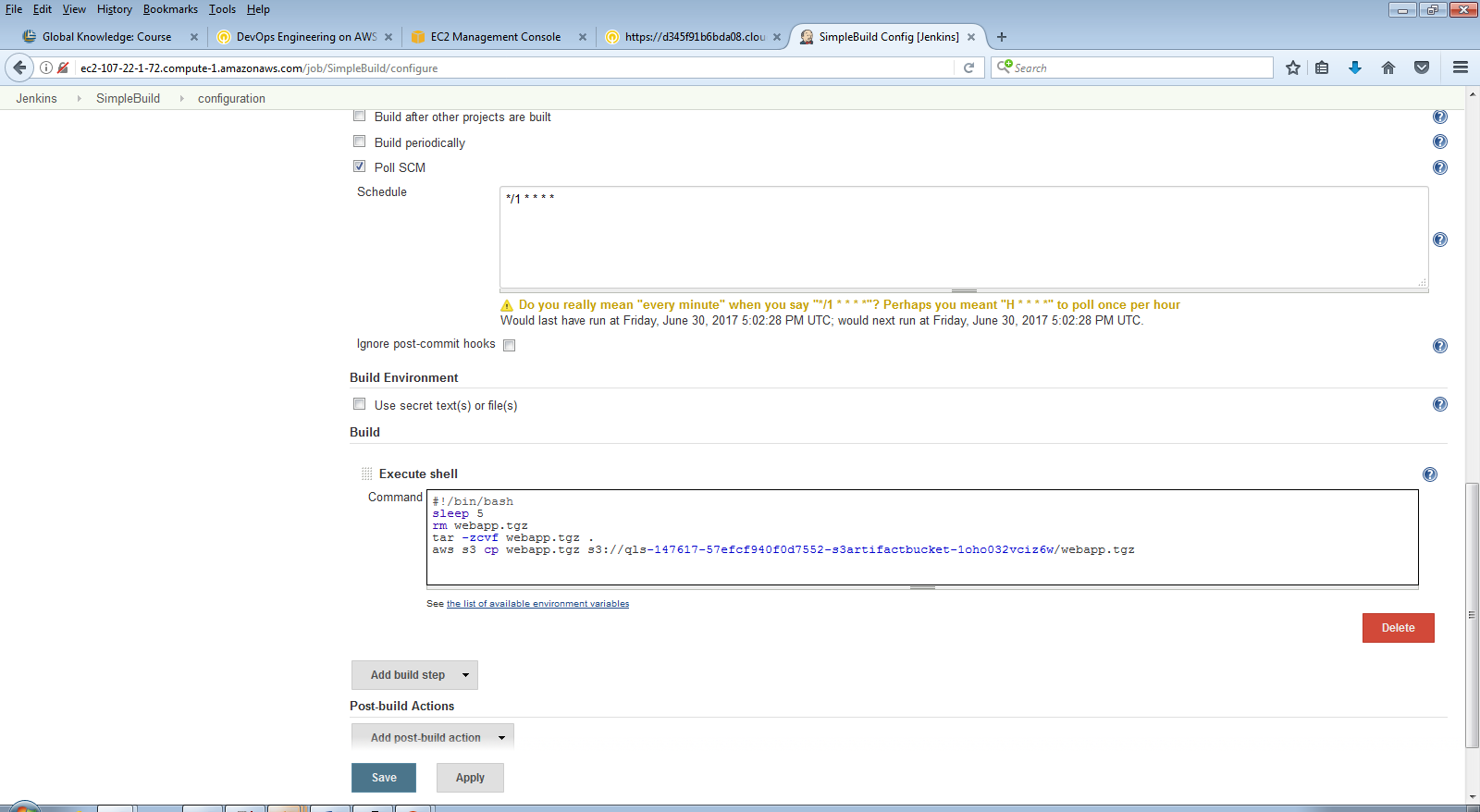
Output:



1.3 Configure Jenkins







=====================================

1.3.5 Set the schedule

\*/1 \* \* \* \*

1.3.7 Shell script for Simple Build job

#!/bin/bash

sleep 5

rm webapp.tgz

tar -zcvf webapp.tgz .

aws s3 cp webapp.tgz s3://<s3ArtifactBucket>/webapp.tgz

1.3.13 Set the schedule

\* \* \* \* \*

1.3.14 Shell script for Simple Test job

#!/bin/bash

destUrl=http://<CommandHostPublicIP>

verify=$(curl $destUrl/index.php | grep "404 Not Found")

if [[ "$verify" == "<title>404 Not Found</title>" ]]; then

echo "Test target does not exist. Check it out and try again!"

echo

exit 1

fi

result=$(ab -n 1000 -c 25 $destUrl/index.php | grep "longest request")

if [[ $result > 200 ]]; then

echo "Performance is terrible. Start over!"

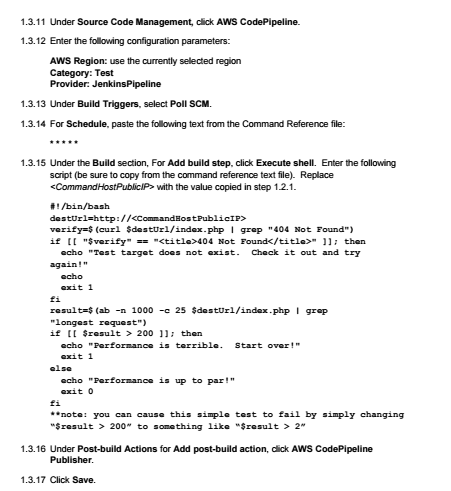
exit 1

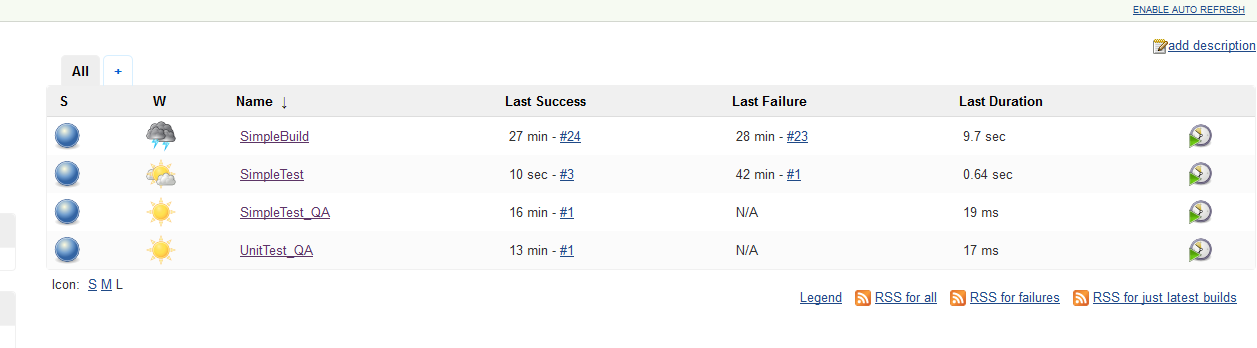
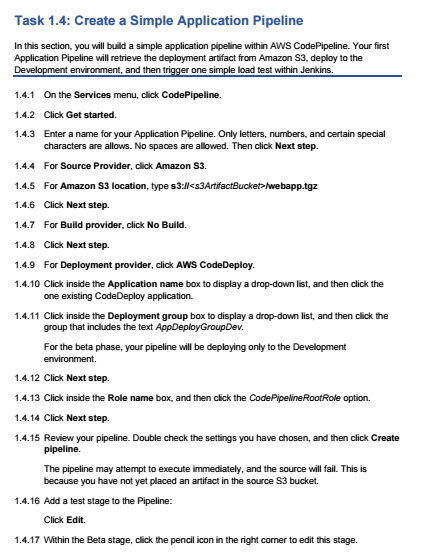
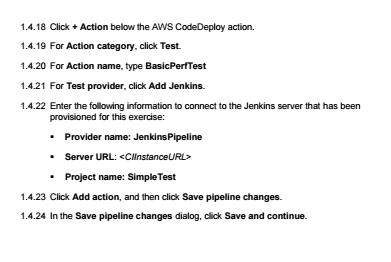
else

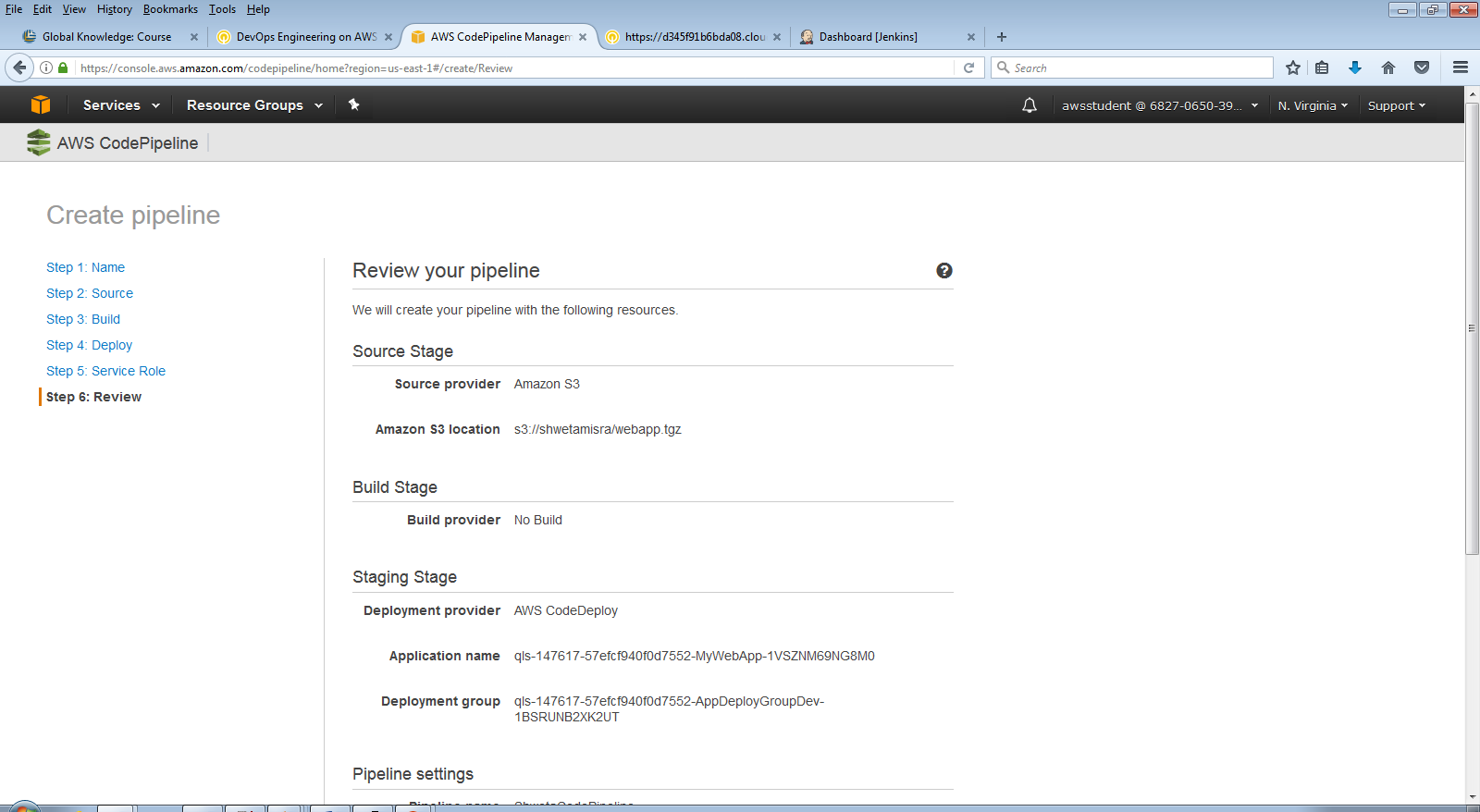
echo "Performance is up to par!"

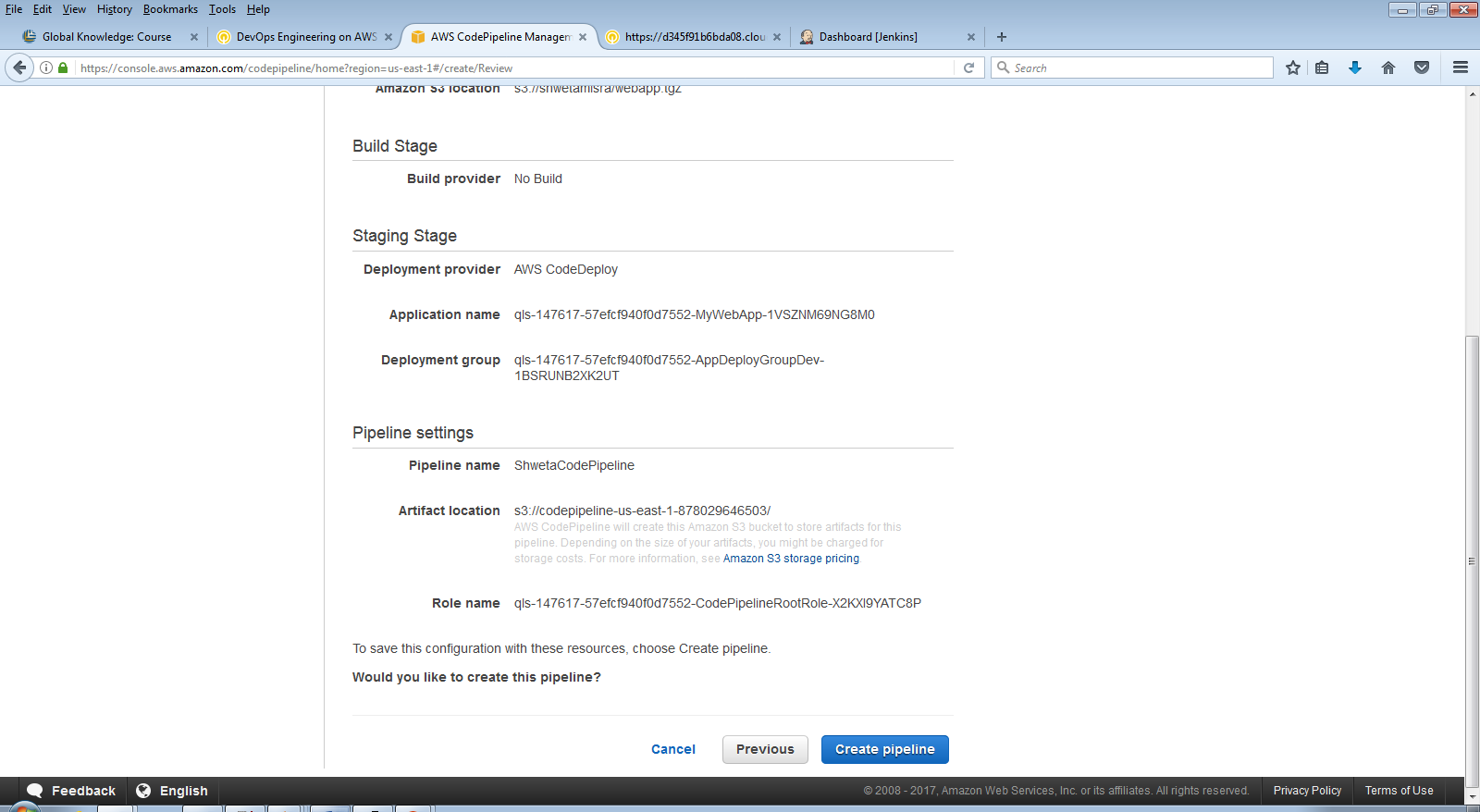
exit 0

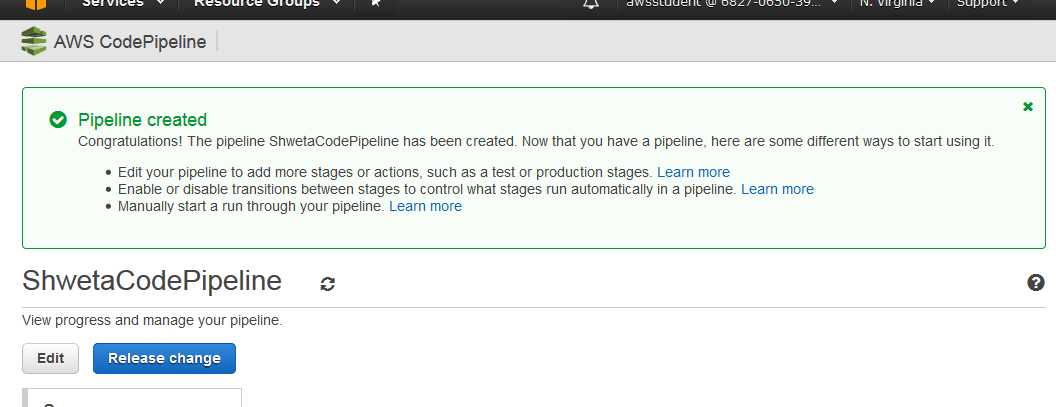
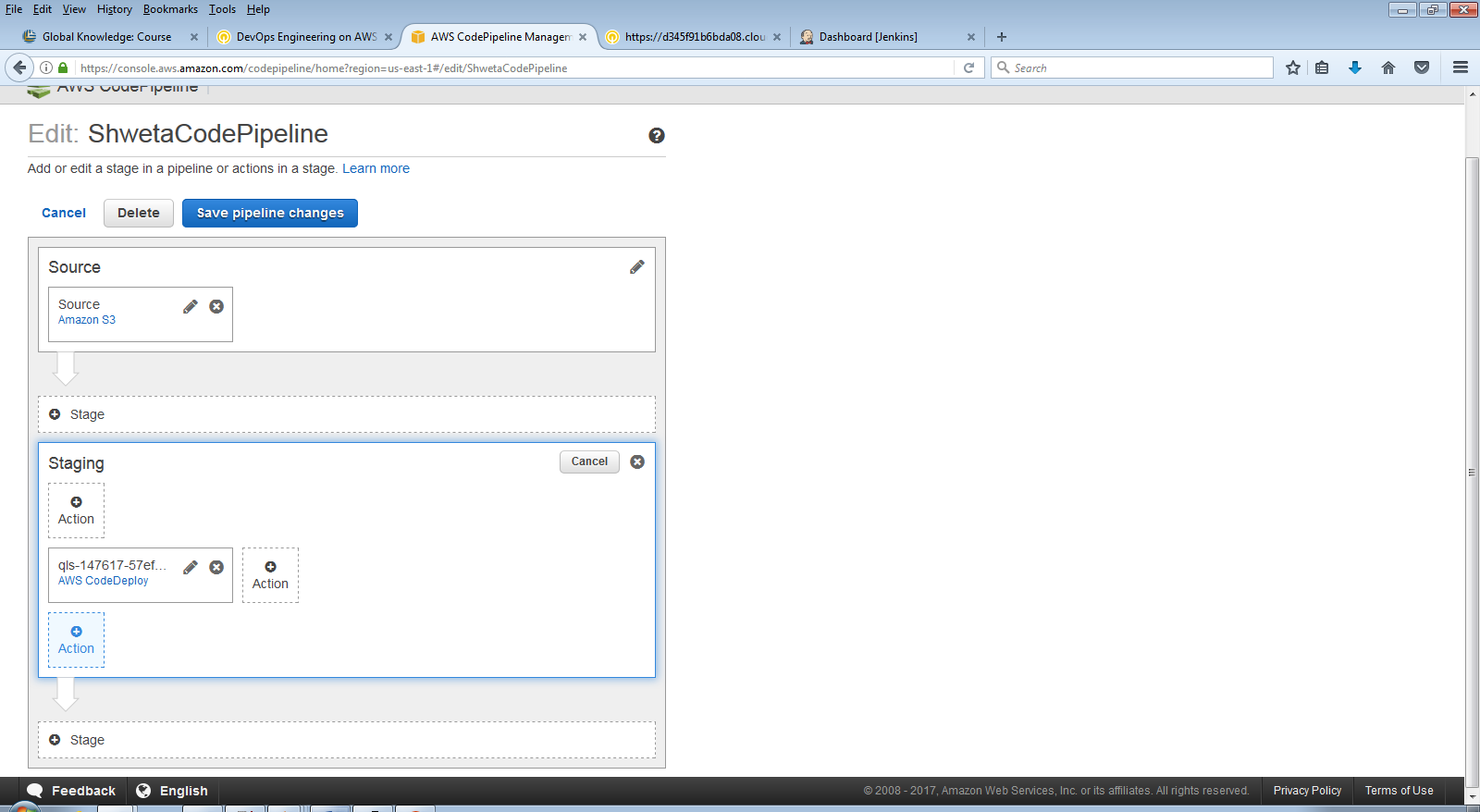
fi

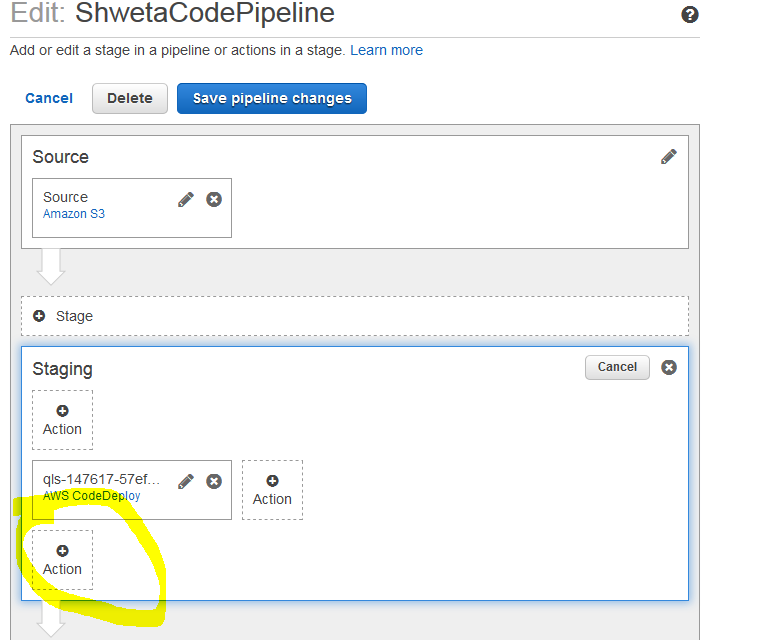






\ 

==================================================================================================================

1.4 Create a Simple Application Pipeline

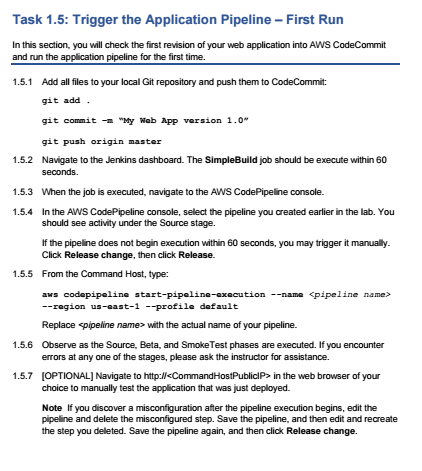
==================================================================================================================

1.4.5 Source Provider location

s3://<s3ArtifactBucket>/webapp.tgz

s3://qls-147617-57efcf940f0d7552-s3artifactbucket-1oho032vciz6w/webapp.tgz

=============================================================================



=====================================

1.5 Trigger the Application Pipeline - First Run

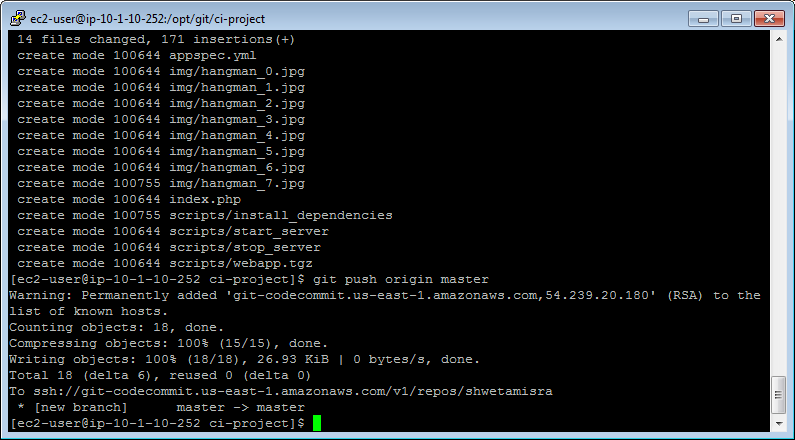
==================================================================================================================

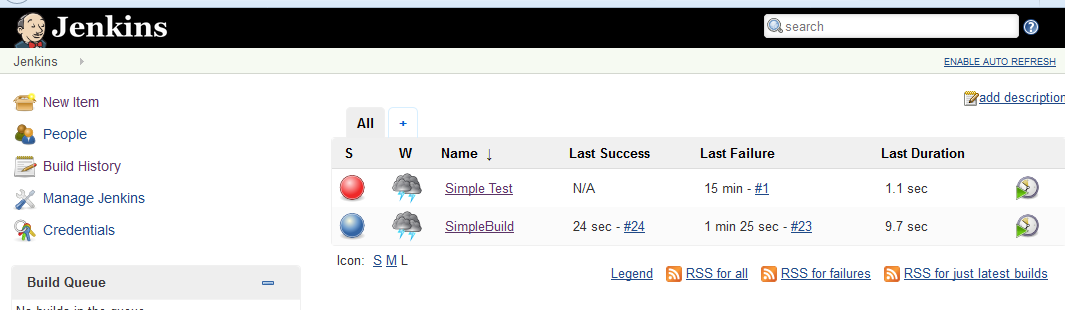
1.5.1 Git commit and push

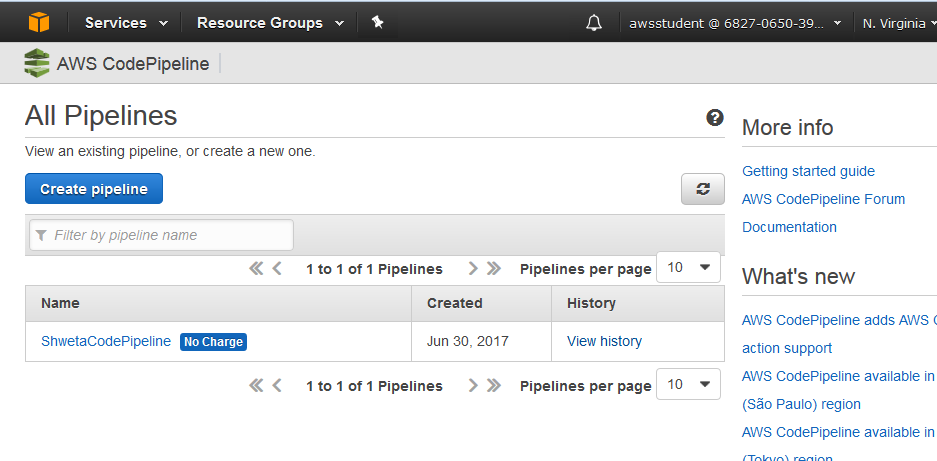
git add .

git commit -m "My Web App version 1.0"

git push origin master







1.5.5 [OPTIONAL] Manually trigger pipeline

aws codepipeline start-pipeline-execution --name <pipeline name> --region us-east-1 --profile default

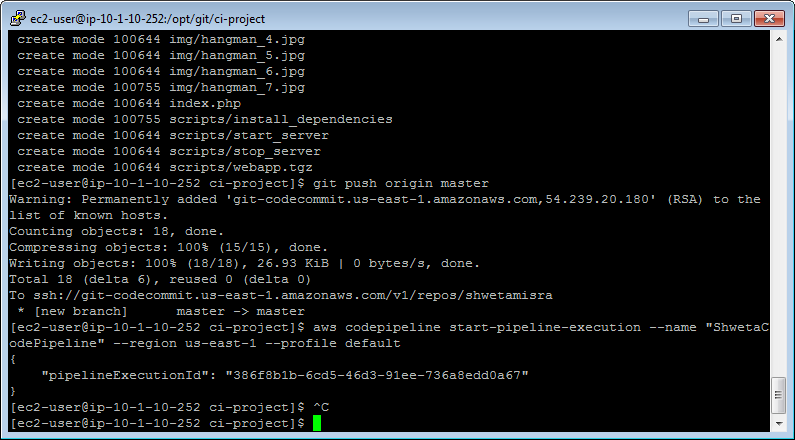
aws codepipeline start-pipeline-execution --name "ShwetaCodePipeline" --region us-east-1 --profile default

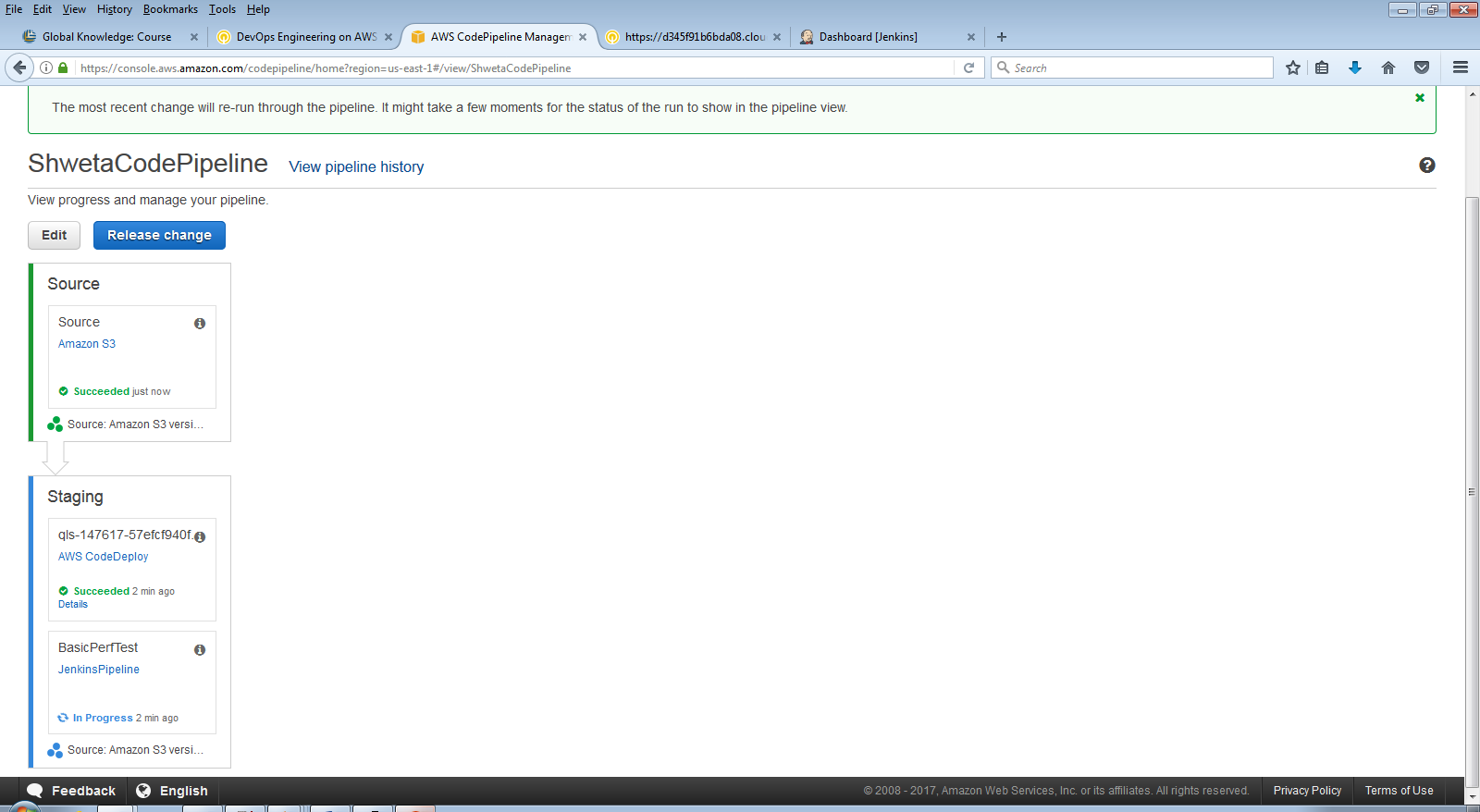
CodePipeline" --region us-east-1 --profile default

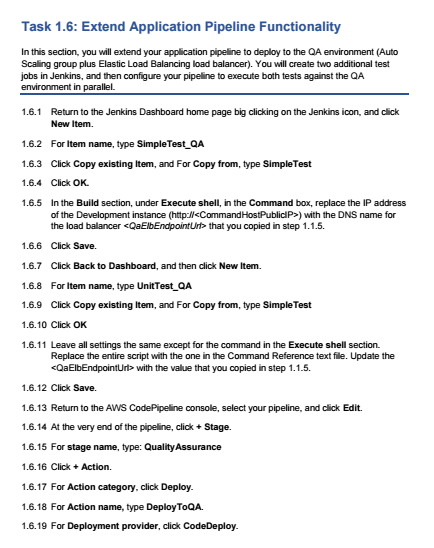
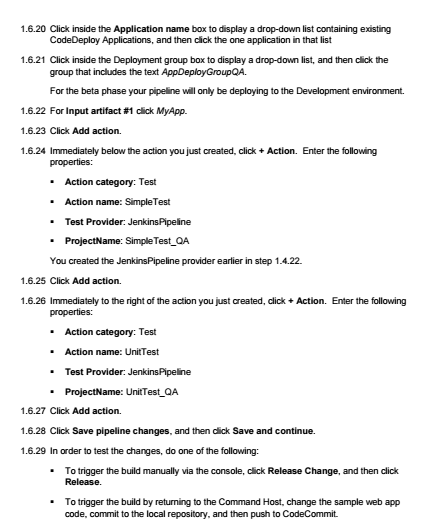
{

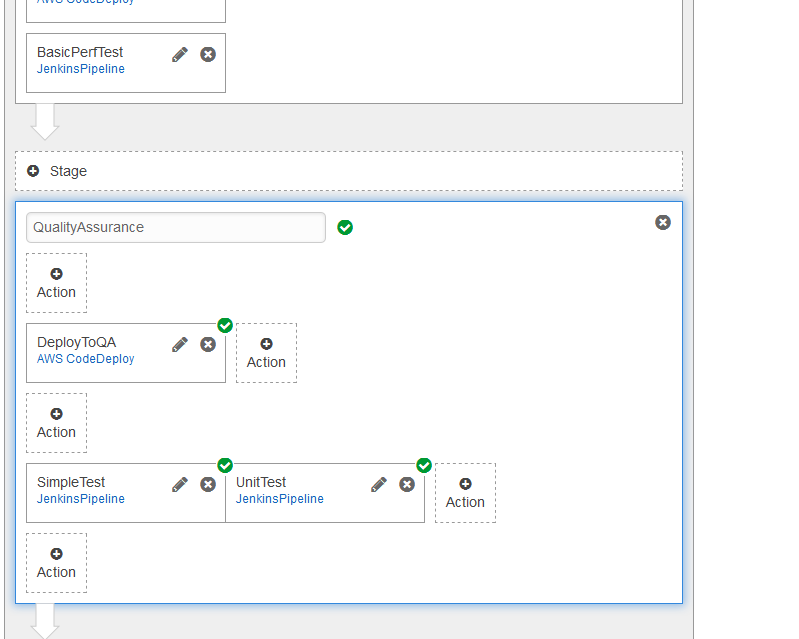
"pipelineExecutionId": "386f8b1b-6cd5-46d3-91ee-736a8edd0a67"

}







==================================================================================================================

1.6 Extend Applicaiton Pipeline Functionality

==================================================================================================================

1.6.11 UnitTest\_QA Script for Jenkins

#!/bin/bash

WebAppUrl="<QaElbEndpointUrl>"

# Do not modify anything below this line!

# ========================================================================================

letters=""

rInt=$(cat /dev/urandom | tr -dc '0-9' | fold -w 1 | head -n 1)

echo "Testing word/phrase # $rInt. Lets see how it goes."

echo

counter=0

while true; do

counter=$[$counter + 1]

rChar=$(cat /dev/urandom | tr -dc 'A-Z' | fold -w 1 | head -n 1)

letters=$letters$rChar

output=$(curl "$WebAppUrl/index.php?letters=$letters&n=$rInt")

testWin=$(echo $output | grep "You win")

testLose=$(echo $output | grep "YOU ARE HANGED")

if [ -n "$testWin" ] || [ -n "$testLose" ]; then

if [ -n "$testWin" ]; then

echo "It looks like we won!!"

break

else

echo "It looks like we did NOT win. Let us try again!"

fi

else

echo "No luck...guess we try again."

echo "Letters guessed so far: $letters" && sleep 2

echo

fi

done

echo "We tested $counter times. Exiting."

echo

