CS 79C Final Project **Date: 04/14/21**

How to cost optimize Jenkins jobs on Kubernetes with EC2 Spot Instances

I choose this project because of my familiarity with standalone Jenkins setup, configuration of free-style and pipelines projects. While preparing for CS 79C, I became familiar with Kubernetes, EC2 instances, etc. Hence combining all these areas would help me learn migration Jenkins to EC2 instances on Cloud9 and enhance my knowledge in these areas.

I will launch an Amazon EKS cluster with a managed node group running On-Demand instances for a Jenkins server. Then, I will set up another EKS managed node group running Spot Instances, and run a sample Jenkins build on those Spot Instances. Later I will also configure the build to retry in case Spot Instances are interrupted, when EC2 needs the capacity back

I will use Cloud9, Amazon EKS Cluster, EC2 Spot and On-Demand instances

Definitions:

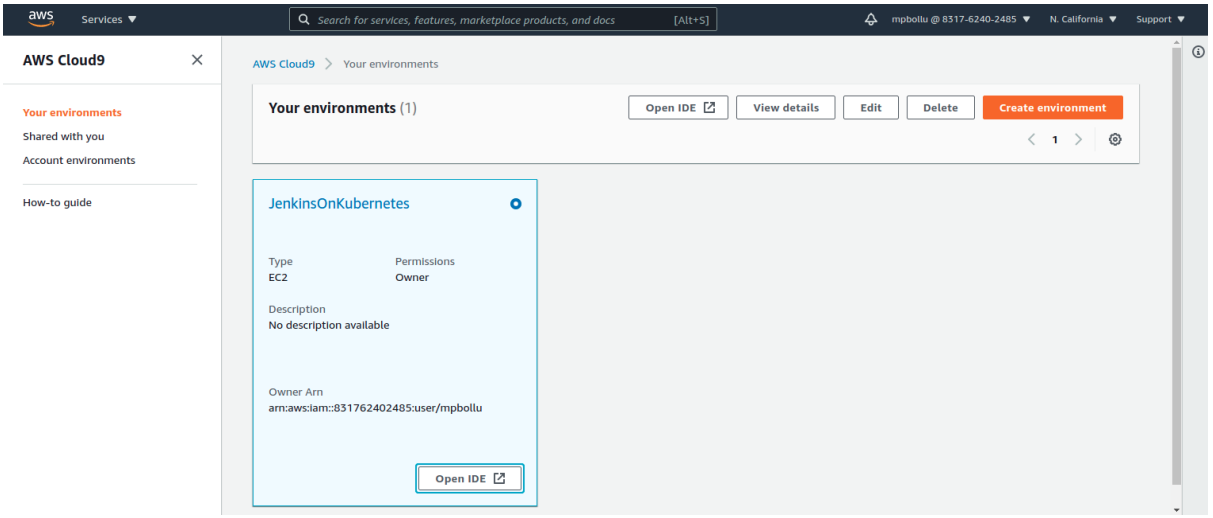
Cloud9: AWS Cloud9 allows you to write, run, and debug your code with just a browser.
Amazon EKS Cluster: An **Amazon EKS cluster** consists of a control plane and the **Amazon** EC2 or AWS Fargate compute that you run pods on.
EC2 Spot and On-Demand instances: A Spot Instance is an unused EC2 instance that is available for less than the On-Demand price.
Jenkins: Automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery.

Referred to Documentation:

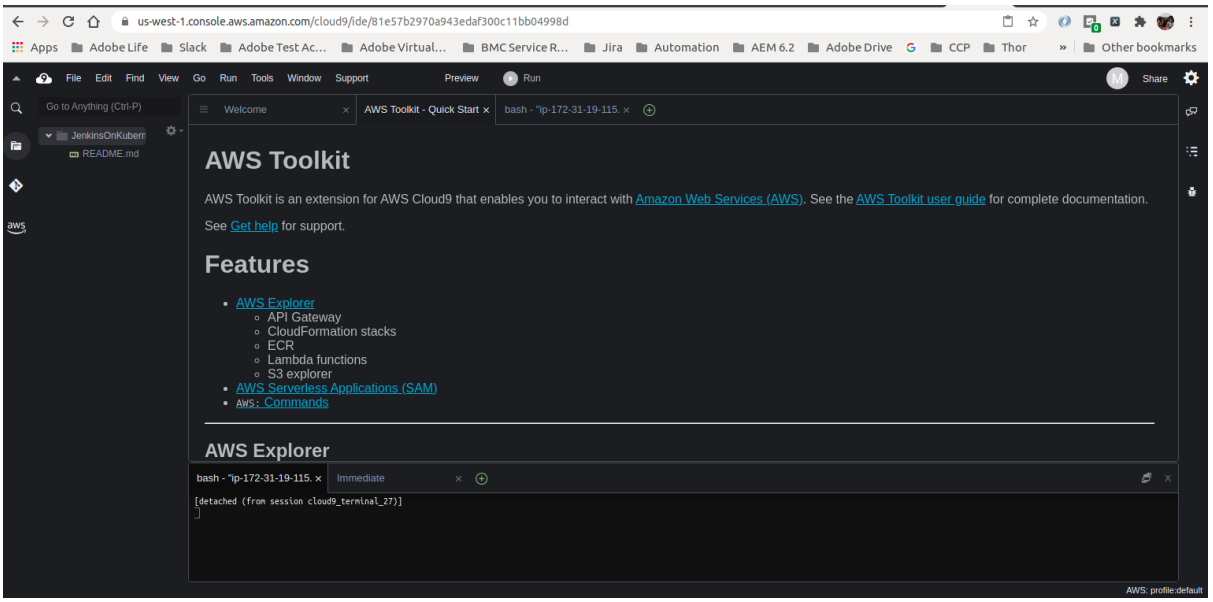
https://aws.amazon.com/getting-started/hands-on/cost-optimize-jenkins/?trk=gs_card

Step 1: Setup Cloud9

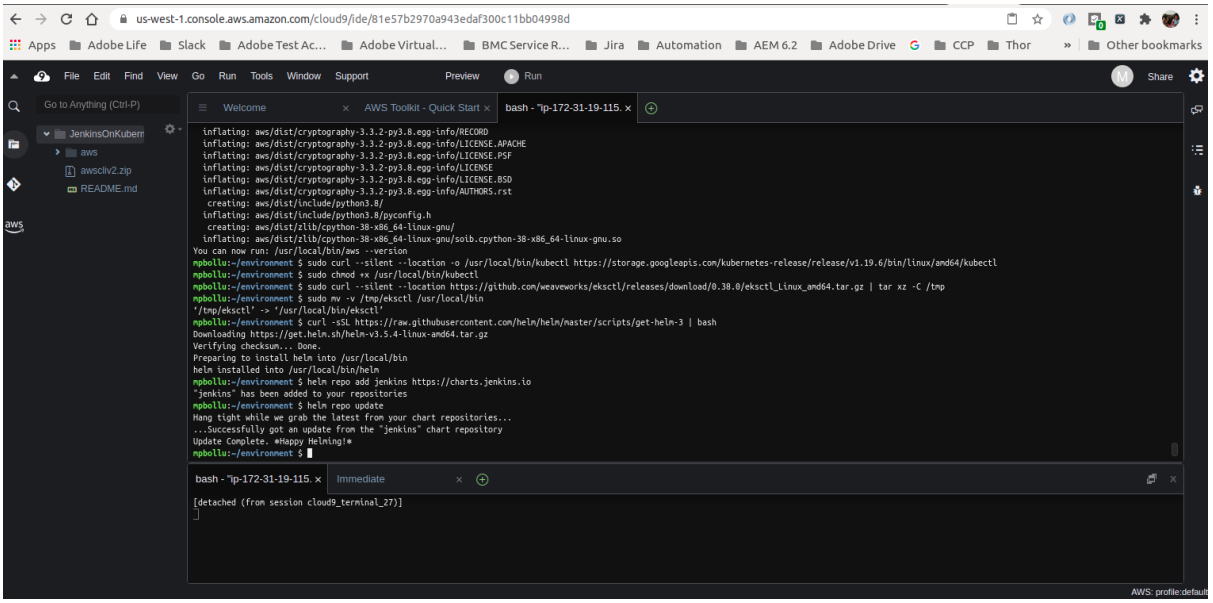
AWS Cloud9 environment created/running:



Open Cloud9 IDE:

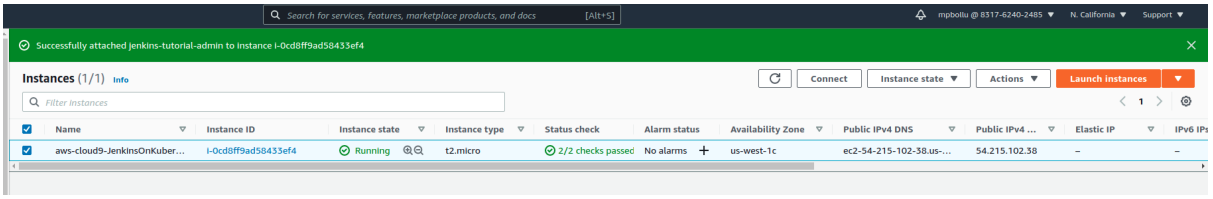


Step 2: Setup Prerequisites

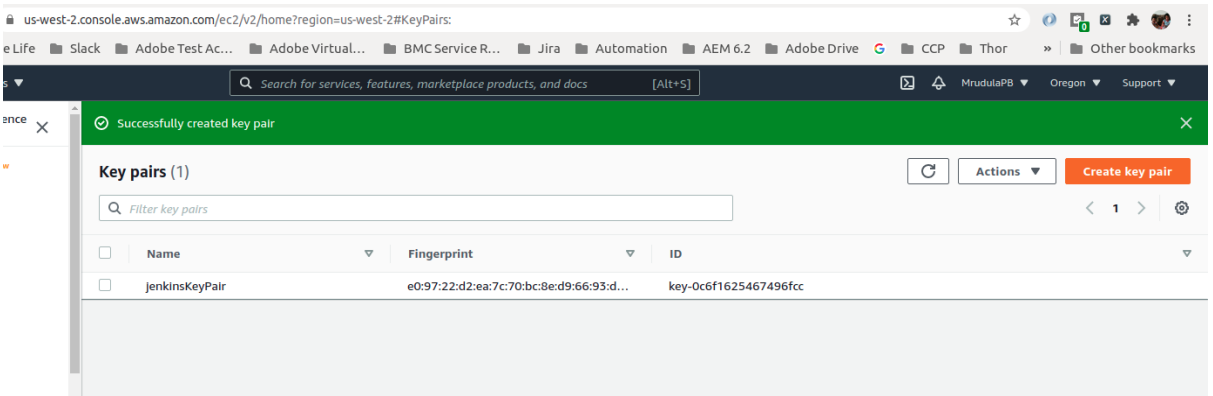


Step 3: Create an EKS cluster

Attach Admin Role to Cloud9 EC2 instance:

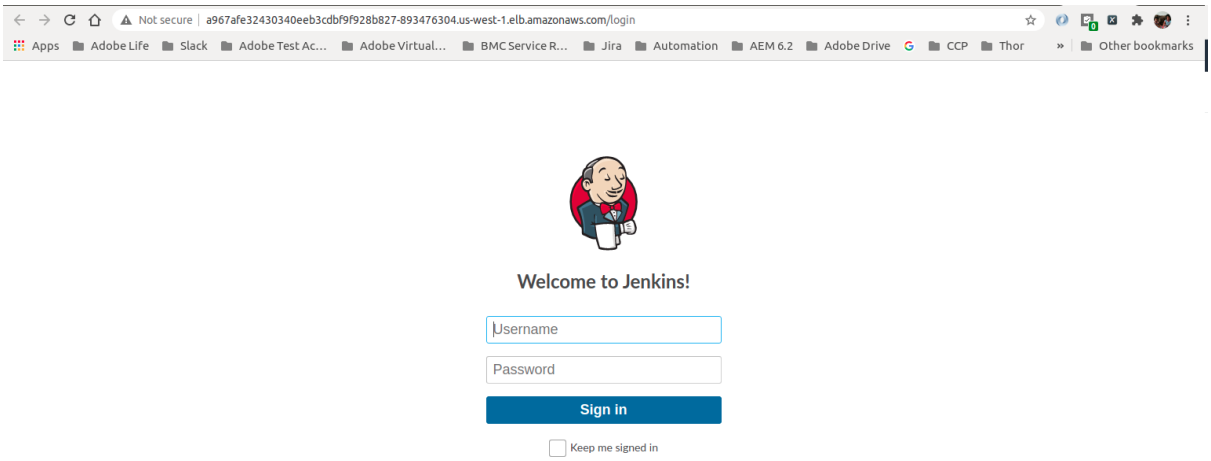


New KeyPair created and added IAM Role:



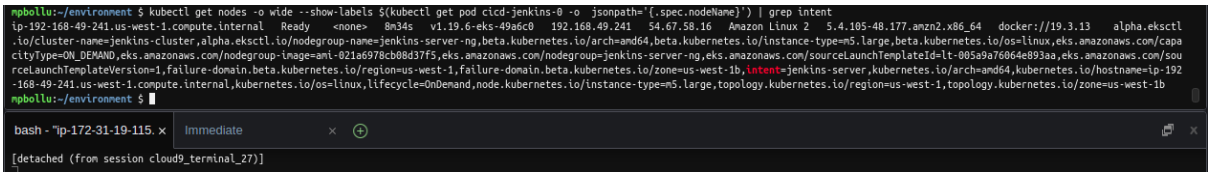
Step 4: Install Jenkins server on an On Demand nodegroup

Running Jenkins from UI:



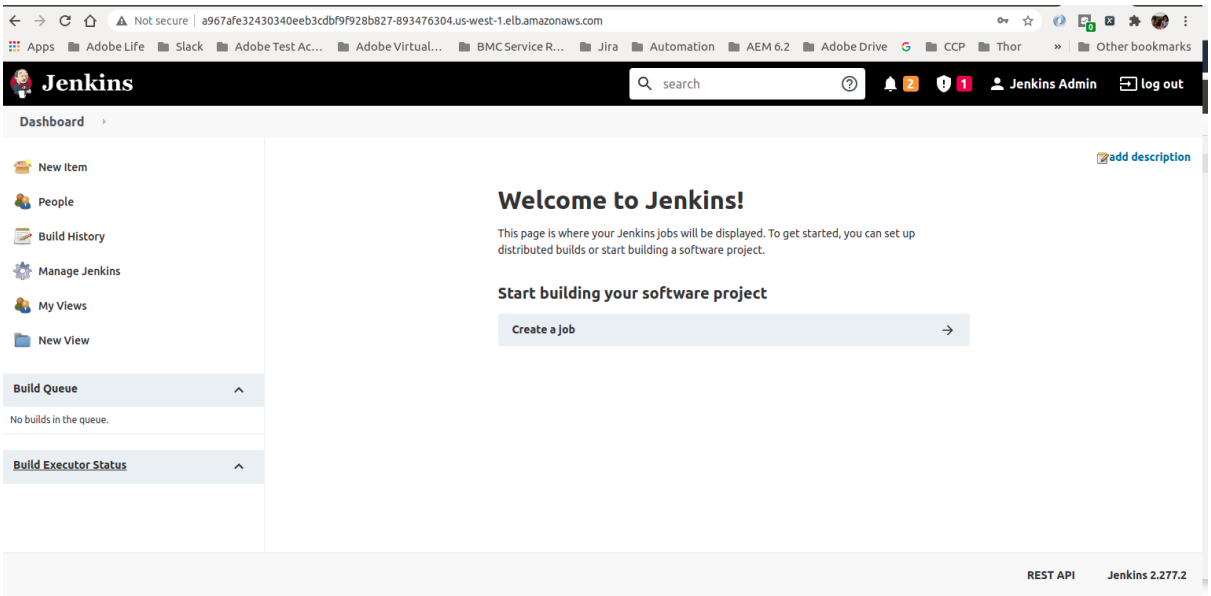
Jenking running on CLI:

intent=jenkins-server **indicating that the Jenkins is running on** ***jenkins-server-ng*** **nodegroup.**

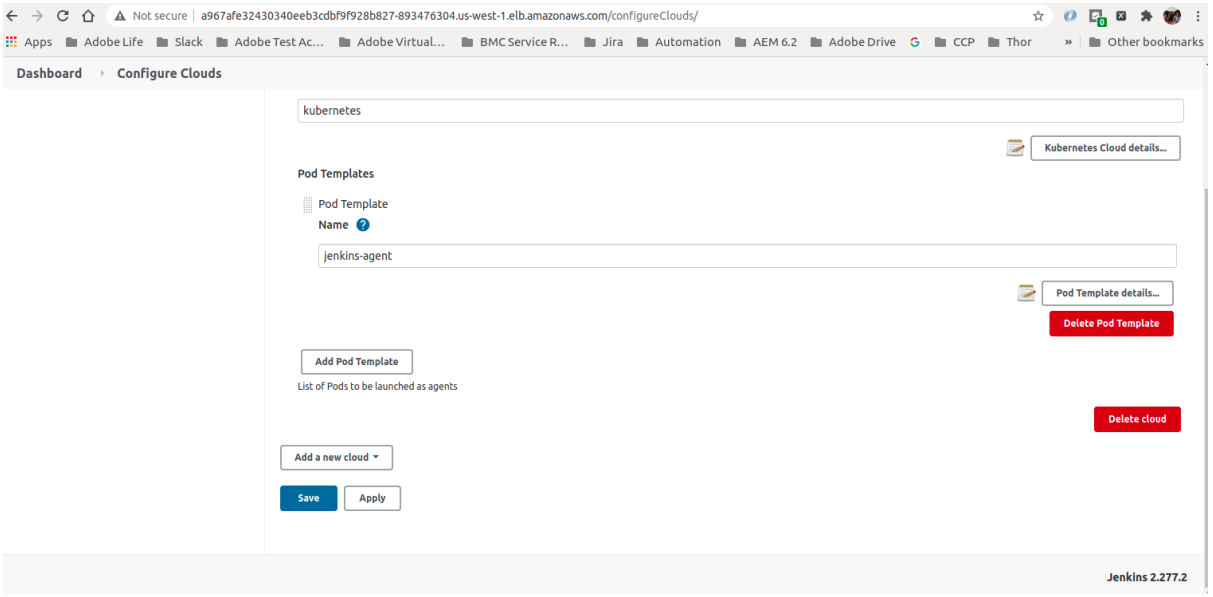


Step 5: Install Jenkins agents on a EC2 Spot nodegroup

Login into Jenkins as Admin:

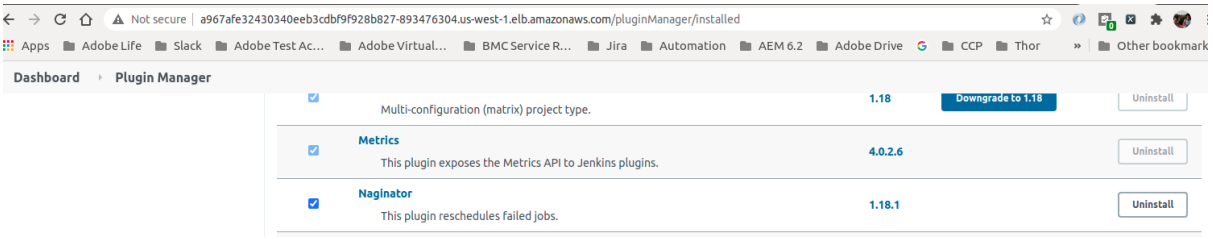


Pod Template:



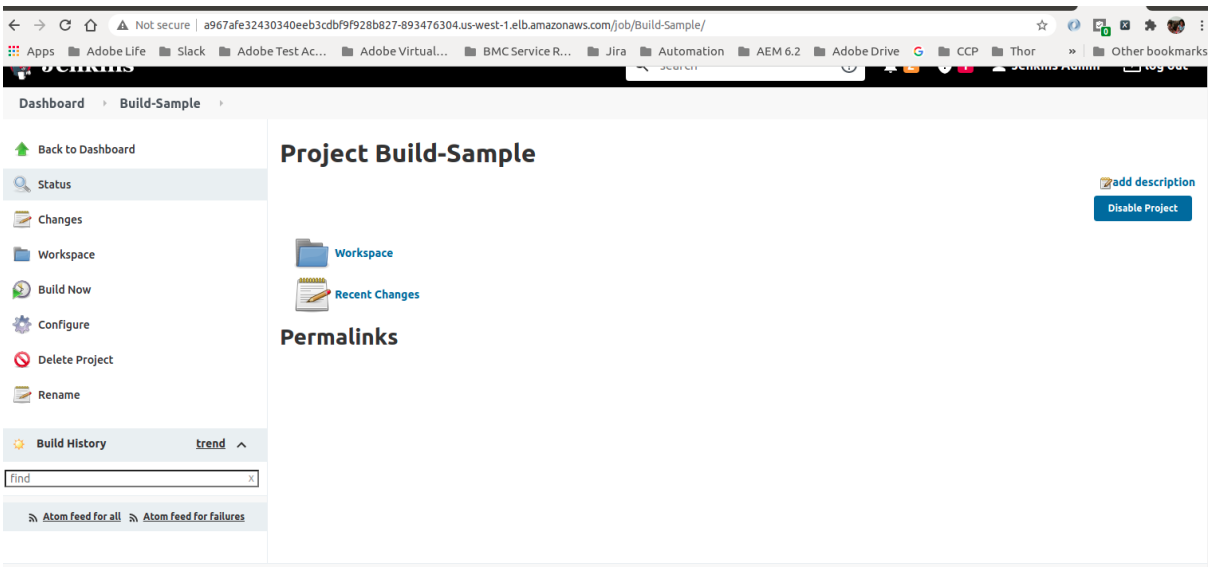
Step 6: Setup Jenkins plugins for retry

Naginator:



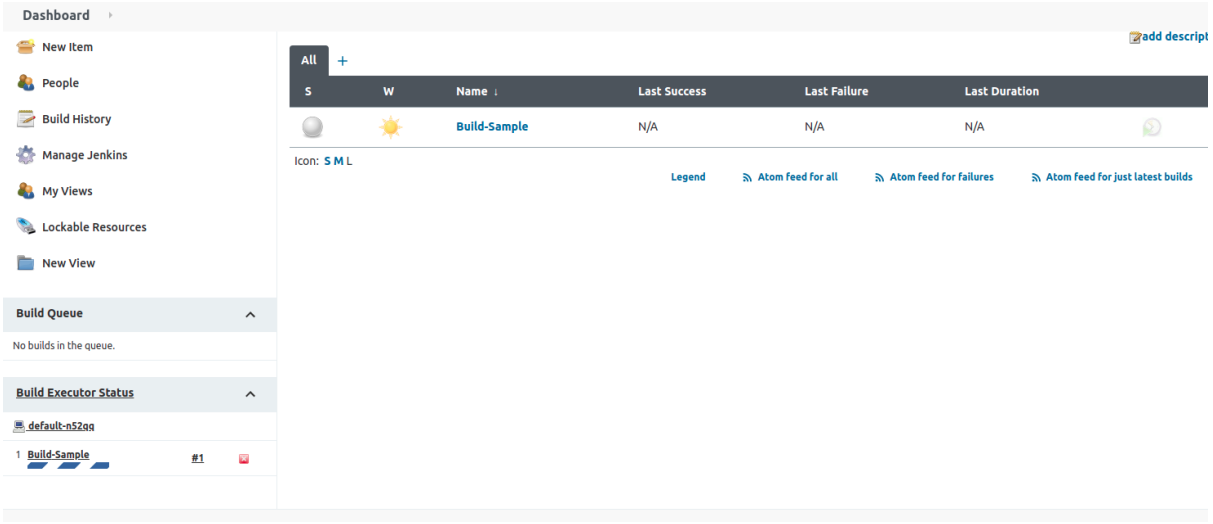
Step 7: Setup a sample Continuous Integration (CI) pipeline in Jenkins

Free-style Build-Sample Project created:

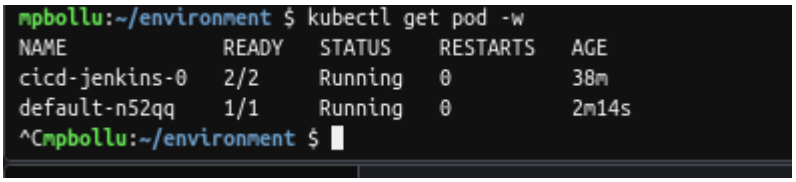


Step 8: Run the sample CI pipeline in Jenkins

Build Now Build-Sample:



Jenkins agent in Cloud9 IDE:



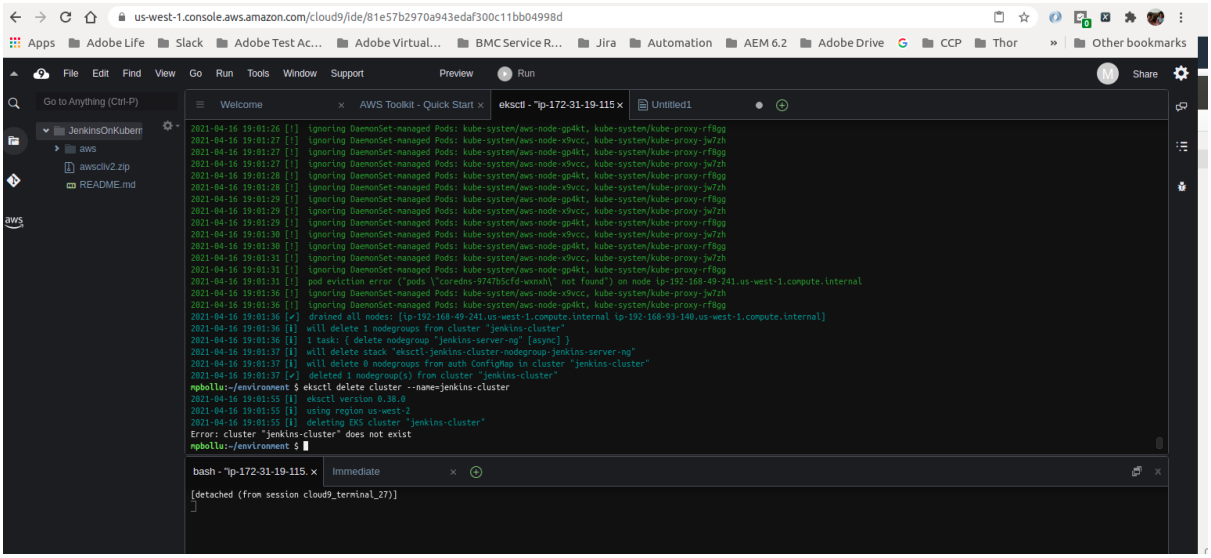
Build-Sample Console Output:

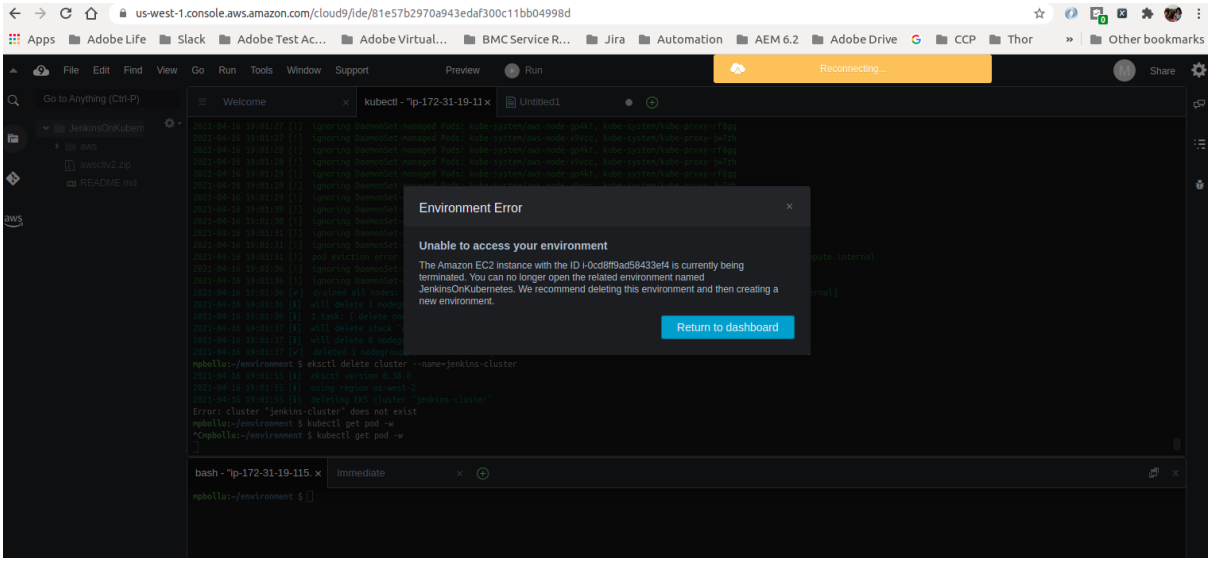
Dashboard › Build-Sample › #1

```
    name: "jnlp"
    resources:
      limits:
        memory: "512Mi"
        cpu: "512m"
      requests:
        memory: "512Mi"
        cpu: "512m"
    tty: false
    volumeMounts:
    - mountPath: "/home/jenkins"
      name: "workspace-volume"
      readOnly: false
    workingDir: "/home/jenkins"
  nodeSelector:
    kubernetes.io/os: "linux"
  restartPolicy: "Never"
  serviceAccountName: "default"
  volumes:
  - emptyDir:
      medium: ""
    name: "workspace-volume"

Building remotely on default-n52qq (cicd-jenkins-agent) in workspace /home/jenkins/workspace/Build-Sample
[Build-Sample] $ /bin/sh -xe /tmp/jenkins6029157892916056446.sh
+ sleep 3m
+ echo Job completed successfully
Job completed successfully
Finished: SUCCESS
```

Step 9: Cleanup




us-west-1.console.aws.amazon.com/cloud9/ide/81e57b2970a943edaf300c11bb04998d
Reconnecting...
Environment Error
Unable to access your environment
The Amazon EC2 instance with the ID i-0cd8ff9ad58433ef4 is currently being terminated. You can no longer open the related environment named JenkinsOnKubernetes. We recommend deleting this environment and then creating a new environment.
Return to dashboard