Gorilla Logic Challenge

By Justo Paredes

Paredes.justo.jp@.gmail.com

This solution is deployed using AWS Cloud Services

Tools:

GIT for SCM

Jenkins to CI/CD pipeline.

Ansible to create infrastructure and start the app

Cloudformation create infrastructure and AWS ELB

Docker to start a mysql db image

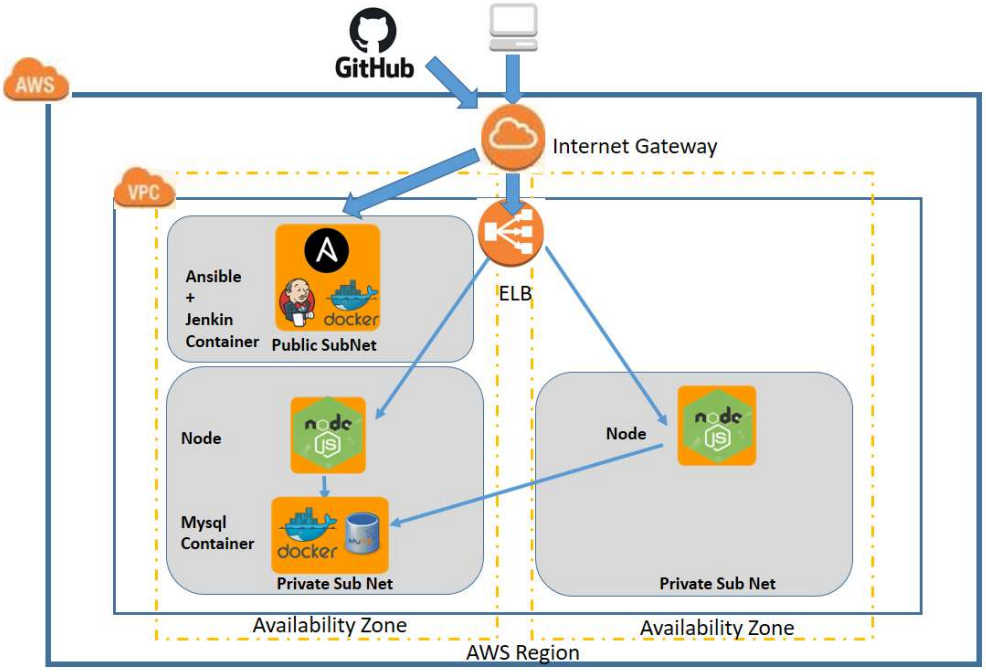
Pre-Requisites

Aws Account

One Ansible control server on EC2 Instance

Aws ssh Key

Diagram


GitHub
AWS
Internet Gateway
VPC
Ansible
+
Jenkin
Container
Public SubNet
ELB
Node
Mysql
Container
Private Sub Net
Node
Private Sub Net
Availability Zone
Availability Zone
AWS Region


Git Push code
Jenkins Start Pipeline
• Build
Ansible Playbook start app

Environment Set UP

1.-Cloning Public Git Reporitory using GitHub Desktop and them create your public repository

https://github.com/paredesjustojp/jpchallenge.git

2.- Deploy Jenkins

- For this challenge, we are going to use the Ansible control server to deploy a dockerized Jenkins Master Node.

  $docker run -p 8080:8080 -p 50000:50000 jenkins/jenkins:lts

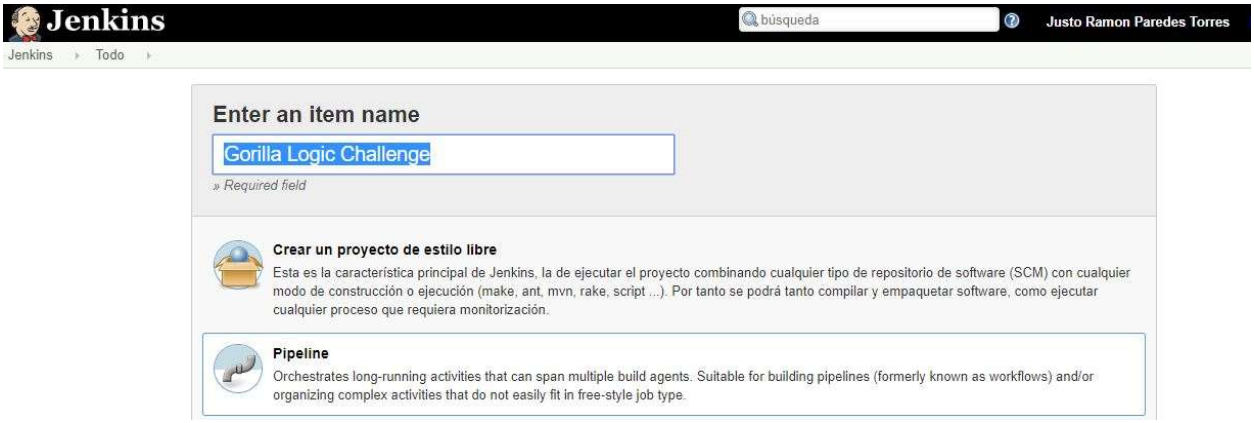
  Get Admin password

  $docker exec -i 1837d93f8f74 cat /var/jenkins_home/secrets/initialAdminPassword

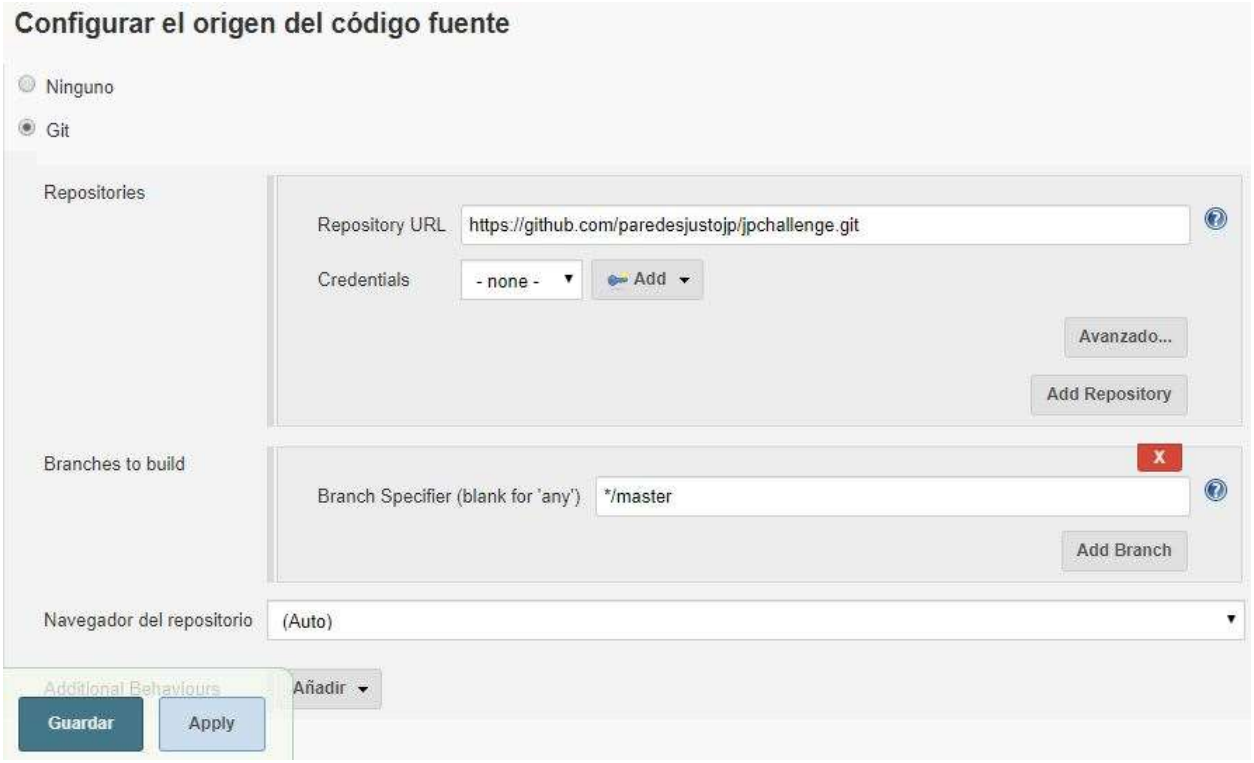
- Install GitHub, node and Ansible plugins

3.- Integrate Jenkins with GitHub

- In Jenkins Main Page select New task and then Pipeline



- Select GIT as SCM and put the public repository



- Go to GitHub Account and configure a Jenkin Webhook to pull de code to your Jenkins Server
- Trigger a commit and verify your pipeline output


```
Started by GitHub push by paredesjustojp
Running as SYSTEM
Building remotely on Gorilla1 (NPM1) in workspace /home/ec2-user/jenkins/workspace/Gorilla
No credentials specified
 > /usr/bin/git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
 > /usr/bin/git config remote.origin.url https://github.com/paredesjustojp/jpchallenge.git # timeout=10
Fetching upstream changes from https://github.com/paredesjustojp/jpchallenge.git
 > /usr/bin/git --version # timeout=10
 > /usr/bin/git fetch --tags --force --progress -- https://github.com/paredesjustojp/jpchallenge.git
 +refs/heads/*:refs/remotes/origin/* # timeout=10
 > /usr/bin/git rev-parse refs/remotes/origin/master^{commit} # timeout=10
 > /usr/bin/git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision 496b4c3dc0017256f2562dff1ab2ad2217075047 (refs/remotes/origin/master)
 > /usr/bin/git config core.sparsecheckout # timeout=10
 > /usr/bin/git checkout -f 496b4c3dc0017256f2562dff1ab2ad2217075047 # timeout=10
Commit message: "Delete jenkinsfile"
 > /usr/bin/git rev-list --no-walk 7864408a10b637e76644df8935a8e6821286f73e # timeout=10
[Gorilla] $ /bin/sh -xe /tmp/jenkins6182864858591377329.sh
+ npm install
npm WARN eslint-plugin-compat@3.5.1 requires a peer of eslint@^3.0.0 || ^4.0.0 || ^5.0.0 || ^6.0.0 but none is
installed. You must install peer dependencies yourself.
npm WARN TimeOff_Management@1.0.0 No repository field.
```

4.- Deploy ELB and Node ec2 Instances

- From your ansible server execute:

  $ansible-playbook cloudformation.yml

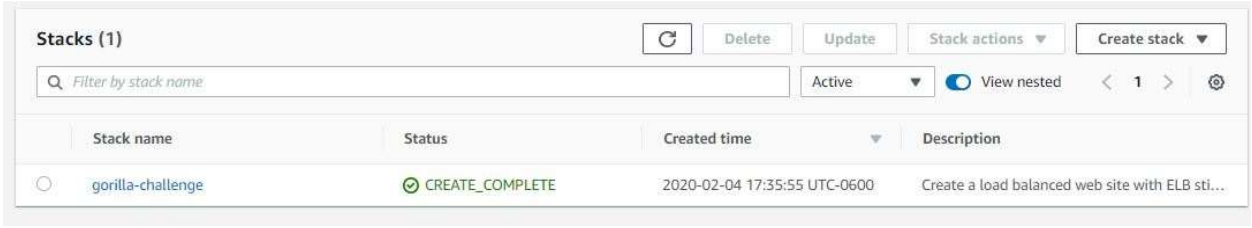
  You can find this playbook in the Public GitHub Repository /Ansible

ec2-user@ip-172-31-26-165:/etc/ansible

```yaml
- name: desplique de infratesructura
  hosts: localhost
  tasks:
    - name: create  cloudformation stack
      cloudformation:
        stack_name: "gorilla-challenge"
        state: "present"
        region: "eu-west-1"
        disable_rollback: true
        template_url: "https://postnetcr.s3.amazonaws.com/postnet/ELB_BCK.json"
        template_parameters:
          VpcId: "vpc-fd65539b"
          NodeImageId: "ami-04ff679b4bca0bf38"
          Subnets: "subnet-0b51316d,subnet-6427542c,subnet-c3329099"
          KeyName: "key_ssh"
          SSHLocation : "0.0.0.0/0"
        tags:
          Stack: "ansible-cloudformation"
~
~
```

Output

```
[ec2-user@ip-172-31-26-165 ansible]$ ansible-playbook cloudformation.yml -vvv

PLAYBOOK: cloudformation.yml ****************************************************************

PLAY [desplique de infratesructura] *********************************************************

TASK [Gathering Facts] **********************************************************************
task path: /etc/ansible/cloudformation.yml:1
ok: [localhost]

TASK [create  cloudformation stack] *********************************************************
task path: /etc/ansible/cloudformation.yml:4
```



Test ELB

ELB for Gorilla Logic Challenge

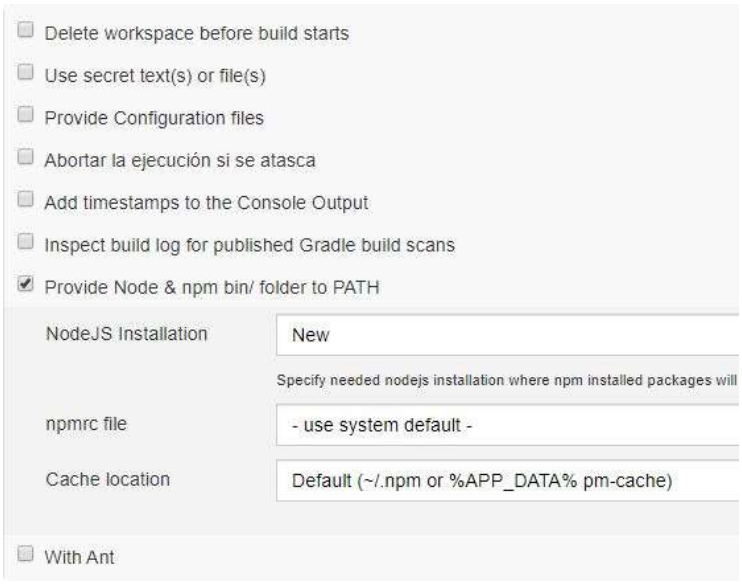
This Step, use a Cloudformation Tempate Previosly uploaded to S3

https://postnetcr.s3.amazonaws.com/postnet/ELB_BCK.json

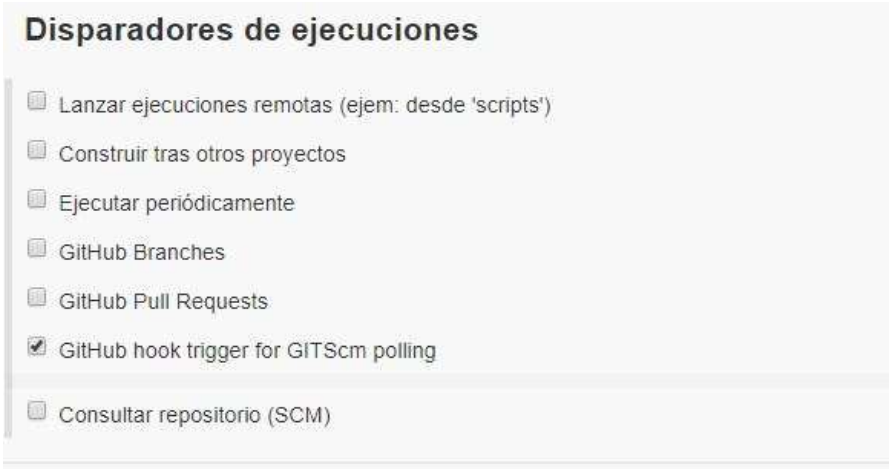
You can find this template in the Public GitHub Repository /Ansible

4.-Configure Open Project to build the app

Set the Node Path



Enable GitHub Commit Triggers



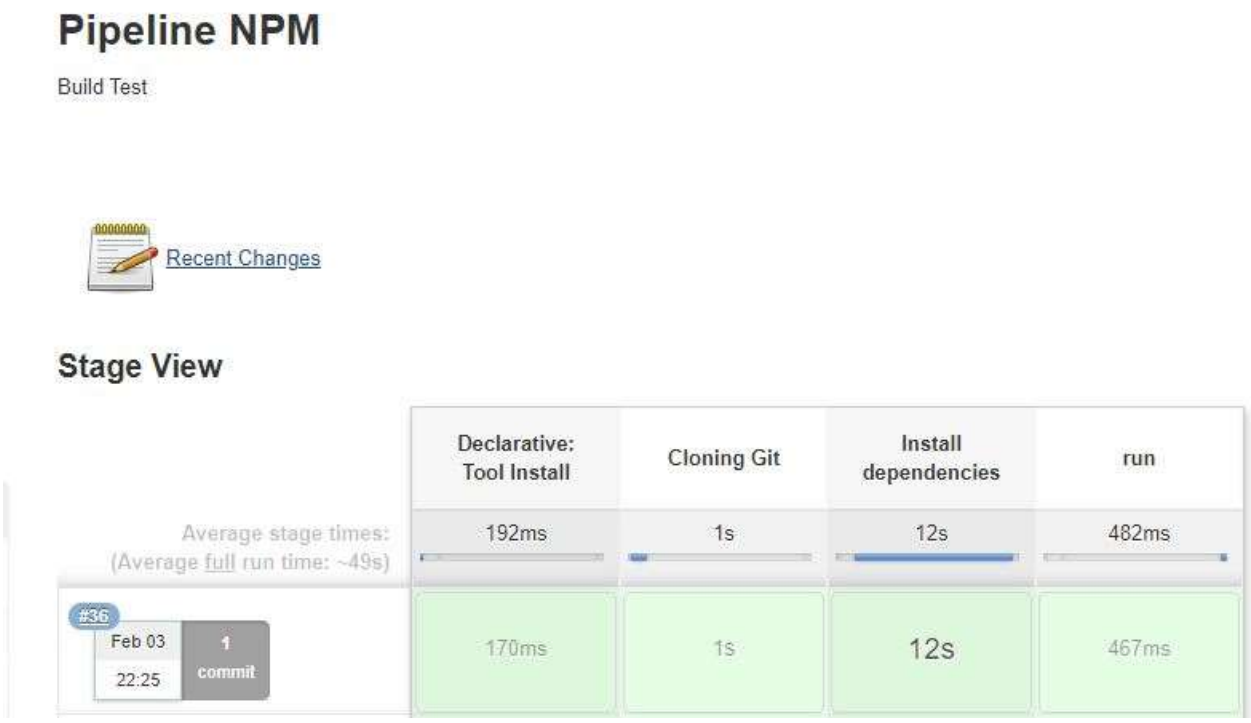
Set commands to buid the app

Alternatively you can configure a Script Pipeline

```
pipeline {
  agent any

  tools {nodejs "New"}

  stages {

    stage('Cloning Git') {
      steps {
        git 'https://github.com/paredesjustojp/jpchallenge.git'
      }
    }

    stage('Install dependencies') {
      steps {
        sh 'npm install'
        sh 'npm rebuild'
      }
    }

    stage('deploy') {
      steps {
        echo 'go to your ansible server and execute ansible-playbook npm.yml'
      }
    }
  }
}
```

You can find this playbook in the Public GitHub Repository

Output



5.- Once the app is build and approved , go to ansible server and execute

$ansible-playbook npm.yml

You can find this template in the Public GitHub Repository /Ansible

```
[ec2-user@ip-172-31-26-165 ansible]$ ansible-playbook npm.yml

PLAY [webservers] **************************************************

TASK [Gathering Facts] *********************************************
ok: [172.31.21.11]
ok: [172.31.25.131]

TASK [staring npm] *************************************************
```

5.Configure ELB to listen over 3000 port and test

Go to your AWS Account and configure ELB targets over port 3000