

# Gorilla Logic Challenge

By Justo Paredes

[Paredes.justo.jp@gmail.com](mailto: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

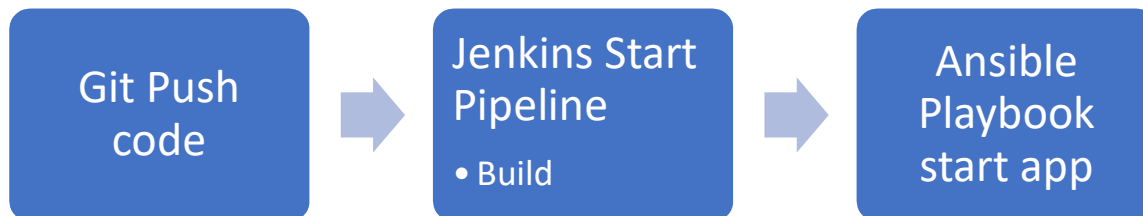
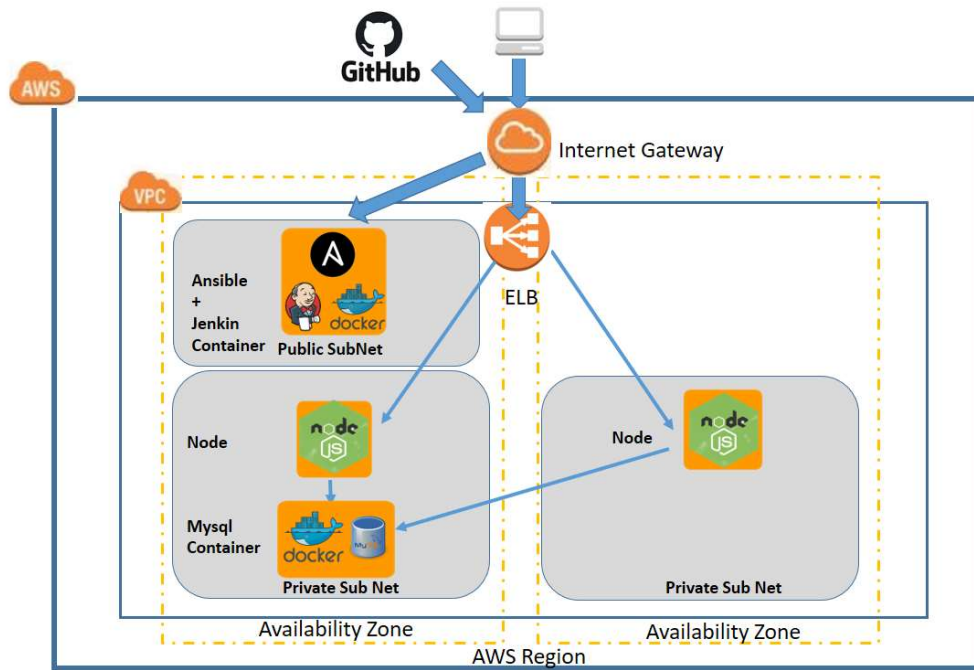
## Pre-Requisites

Aws Account

One Ansible control server on EC2 Instance

Aws ssh Key

## Diagram



## Environment Set UP

1.-Cloning Public Git Repository using GitHub Desktop and then create your public repository

<https://github.com/paredesjustoip/jpchallenge.git>

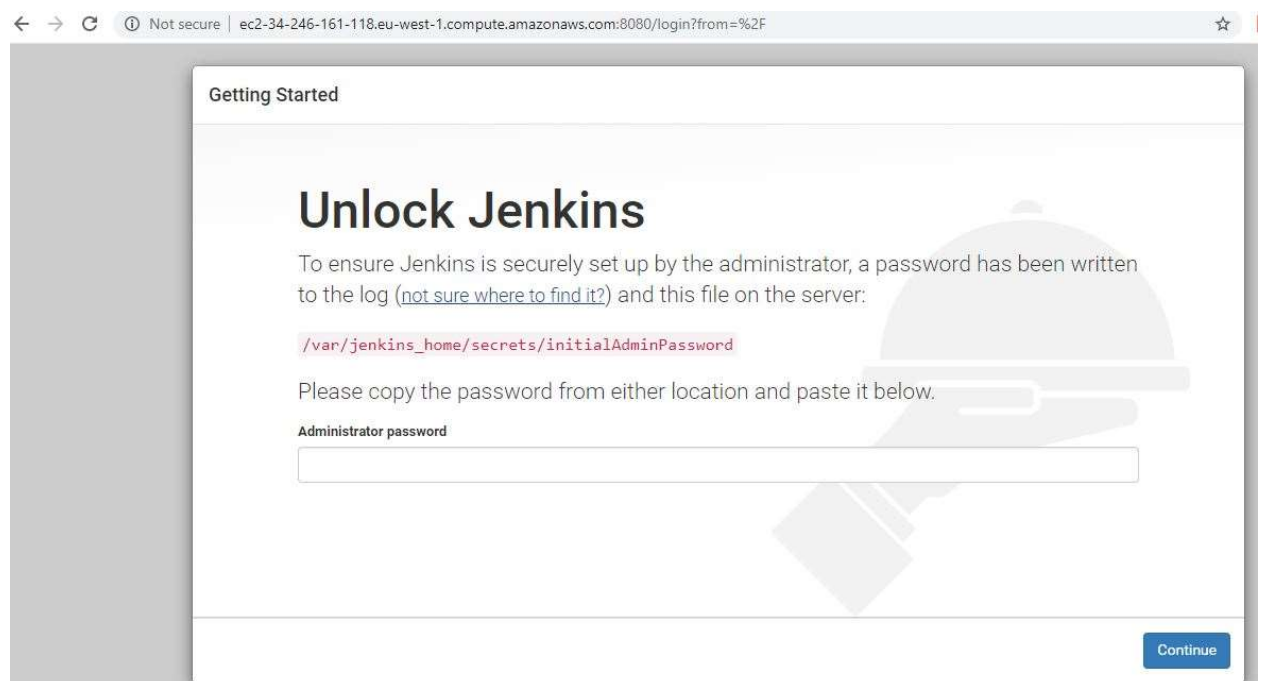
2.- Deploy Jenkins

- 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:its
```

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

**Jenkins** Justo Ramon Paredes Torres


Jenkins > Todo >

Enter an item name

» Required field

 **Crear un proyecto de estilo libre**

Esta es la característica principal de Jenkins, la de ejecutar el proyecto combinando cualquier tipo de repositorio de software (SCM) con cualquier modo de construcción o ejecución (make, ant, mvn, rake, script ...). Por tanto se podrá tanto compilar y empaquetar software, como ejecutar cualquier proceso que requiera monitorización.

 **Pipeline**

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

- Select GIT as SCM and put the public repository

### Configurar el origen del código fuente

☐ Ninguno  
☒ Git

Repositories

Repository URL

Credentials

Branches to build

Branch Specifier (blank for 'any')

Navegador del repositorio:

Additional Behaviours

- 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

## Salida de consola

```
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
- name: despligue 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 -vvvv
ansible-playbook 2.7.1
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/ec2-user/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/local/lib/python2.7/site-packages/ansible
  executable location = /usr/local/bin/ansible-playbook
  python version = 2.7.15 (default, Nov 24 2016, 22:14:04) [GCC 4.8.5 20160521 (Red Hat 4.8.5-16)]
Using /etc/ansible/ansible.cfg as config file
/usr/share/ansible does not meet host list requirements, check plugin documentation if this is unexpected
/usr/share/ansible does not meet script requirements, check plugin documentation if this is unexpected
Parsed /etc/ansible/inventory source with ini plugin

PLAYBOOK: cloudformation.yml *****
  1 plays in cloudformation.yml

PLAY [despligue de infratesructura] *****

TASK [Gathering Facts] *****
task path: /etc/ansible/cloudformation.yml:1
127.0.0.1: ESTABLISH LOCAL CONNECTION FOR USER: ec2-user
127.0.0.1: EXEC /bin/sh -c 'echo ~ec2-user && sleep 0'
127.0.0.1: EXEC /bin/sh -c '(id && echo ?) && echo -p *' && echo /home/ec2-user/.ansible/tmp/ansible-tmp-1540818161.98-344614944
6144444127*
127.0.0.1: PUT /home/ec2-user/.ansible/tmp/ansible-local-1540818161_?/tmp/ansible-tmp-1540818161.98-3446149446144444127
127.0.0.1: EXEC /bin/sh -c 'mkdir -p /home/ec2-user/.ansible/tmp/ansible-tmp-1540818161.98-3446149446144444127 /home/ec2-user
1540818161/ansible-tmp-1540818161.98-3446149446144444127 && sleep 0'
127.0.0.1: EXEC /bin/sh -c '/usr/bin/python2.7 /home/ec2-user/.ansible/tmp/ansible-tmp-1540818161.98-3446149446144444127/Ansib
127.0.0.1: EXEC /bin/sh -c 'rm -f -r /home/ec2-user/.ansible/tmp/ansible-tmp-1540818161.98-3446149446144444127 && echo null' &&
ok: [localhost]
META: ran handlers

TASK [create cloudformation stack] *****
task path: /etc/ansible/cloudformation.yml:4
127.0.0.1: ESTABLISH LOCAL CONNECTION FOR USER: ec2-user
127.0.0.1: EXEC /bin/sh -c 'echo ~ec2-user && sleep 0'
127.0.0.1: EXEC /bin/sh -c '(id && echo ?) && echo -p *' && echo /home/ec2-user/.ansible/tmp/ansible-tmp-1540818161.98-11336964
13336964127*
127.0.0.1: PUT /home/ec2-user/.ansible/tmp/ansible-local-1540818161_?/tmp/ansible-tmp-1540818161.98-1133696413336964127
127.0.0.1: EXEC /bin/sh -c 'mkdir -p /home/ec2-user/.ansible/tmp/ansible-tmp-1540818161.98-1133696413336964127 /home/ec2-user
1540818161/ansible-tmp-1540818161.98-1133696413336964127 && sleep 0'
127.0.0.1: EXEC /bin/sh -c '/usr/bin/python2.7 /usr/share/ansible/modules/cloud/amazon/cloudformation.py
127.0.0.1: PUT /home/ec2-user/.ansible/tmp/ansible-local-1540818161_?/tmp/ansible-tmp-1540818161.98-1133696413336964127
127.0.0.1: EXEC /bin/sh -c 'rm -f -r /home/ec2-user/.ansible/tmp/ansible-tmp-1540818161.98-1133696413336964127 && echo null' &&
ok: [localhost]
```

Stacks (1)			
<input type="text" value="Filter by stack name"/> <span>Active</span> <span>View nested</span> <span>&lt; 1 &gt;</span>			
Stack name	Status	Created time	Description
<a href="#">gorilla-challenge</a>	CREATE_COMPLETE	2020-02-04 17:35:55 UTC-0600	Create a load balanced web site with ELB sti...

Test ELB



## ELB for Gorilla Logic Challenge

This Step, use a Cloudformation Tempate Previously uploaded to S3

[https://postnetcr.s3.amazonaws.com/postnet/ELB\\_BCK.json](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



☐ Delete workspace before build starts

☐ Use secret text(s) or file(s)

☐ Provide Configuration files

☐ Abortar la ejecución si se atasca

☐ Add timestamps to the Console Output

☐ Inspect build log for published Gradle build scans

☒ Provide Node & npm bin/ folder to PATH

---

NodeJS Installation

Specify needed nodejs installation where npm installed packages will

npmrc file

Cache location

---

☐ With Ant

Enable GitHub Commit Triggers

## Disparadores de ejecuciones

☐ Lanzar ejecuciones remotas (ejem: desde 'scripts')

☐ Construir tras otros proyectos

☐ Ejecutar periódicamente

☐ GitHub Branches

☐ GitHub Pull Requests

☒ GitHub hook trigger for GITScm polling

---

☐ Consultar repositorio (SCM)

Set commands to build the app

### Ejecutar

Ejecutar línea de comandos (shell) X ?

Comando

Visualizar [la lista de variables de entorno disponibles](#)

Avanzado...



Alternatively you can configure a Script Pipeline

```
1  pipeline {
2    agent any
3
4    tools {nodejs "New"}
5
6    stages {
7
8      stage('Cloning Git') {
9        steps {
10          git 'https://github.com/paredesjustojp/jpchallenge.git'
11        }
12      }
13
14      stage('Install dependencies') {
15        steps {
16          sh 'npm install'
17          sh 'npm rebuild'
18        }
19      }
20
21      stage('deploy') {
22        steps {
23          echo 'go to your ansible server and execute ansible-playbook npm.yml'
24        }
25      }
26    }
27  }
```

You can find this playbook in the Public GitHub Repository

Output

## Pipeline NPM

Build Test



[Recent Changes](#)

### Stage View



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 [webserver] *****

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

### Register and deregister targets

#### Registered targets

To deregister instances, select one or more registered instances and then click Remove.

[Remove](#)

<input type="checkbox"/>	Instance	Name	Port	State	Security groups
<input type="checkbox"/>	i-0efd353cfdccd8ea4	Nodo2 Stack	3000	● running	gorilla-challenge-InstanceSecurityGroup-6lF...
<input type="checkbox"/>	i-002dfceef9e96cfc3	Nodo 1 Stack	3000	● running	gorilla-challenge-InstanceSecurityGroup-6lF...

#### Instances

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target registered on the specified port, you must specify a different port.

[Add to registered](#) on port

Not secure | goril-appli-1og0w72f61h70-861466151.eu-west-1.elb.amazonaws.com/login/

TimeOff.Management

# Login

Employee email:

Password

[Login](#)

[Forgot password?](#) | [Register new company](#)

© TimeOff.management 2014-2019

