

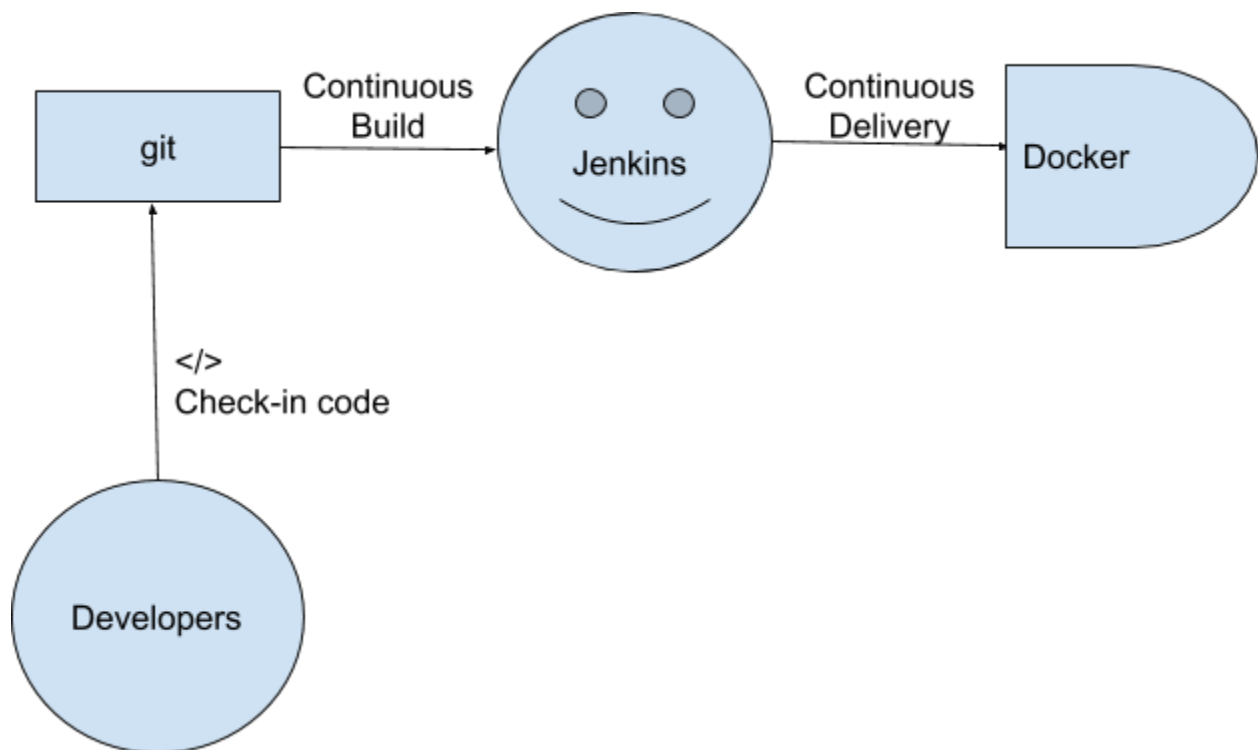
CI/CD PIPELINE USING GIT, JENKINS & DOCKER ON AWS EC2 INSTANCES

INTRODUCTION

The following document's steps taken to create a CI/CD Pipeline for a Node.js application using GIT for source control management, Jenkins as a build and deployment tool, Docker as the target environment in AWS environment

PROCESS FLOW

Whenever code is available in git, jenkins pulls the code and performs the build, once build is completed it generates artifacts. These artifacts are deployed to a target environment which could be a VM, docker container or kubernetes cluster.



PREREQUISITES

- Have an AWS account

Create two EC2 linux instances on AWS console(remember to turn on Password Authentication in /etc/ssh/sshd_config). One for the Jenkins Server and the other for the Docker Host.

INSTALL JENKINS ON JENKINS SERVER

```
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
```

```
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
yum -y install jenkins
```

Start Jenkins

```
service jenkins start
```

Set Jenkins to start at boot

```
chkconfig jenkins on
```

Accessing Jenkins

http://YOUR-SERVER-PUBLIC-IP:8080

Configure Jenkins Console

The default Username is admin

Grab the default password

Password Location: /var/lib/jenkins/secrets/initialAdminPassword

Change admin password

Admin > Configure > Password

Configure java path

Manage Jenkins > Global Tool Configuration > JDK

Create another admin user id

INSTALL DOCKER ON DOCKER HOST

#Install docker and start docker services

```
yum install docker -y
```

```
docker --version
```

start docker services

```
service docker start
```

```
service docker status
```

#Create a user called dockeradmin

```
useradd dockeradmin
```

```
passwd dockeradmin
```

#add a user to docker group to manage docker

```
usermod -aG docker dockeradmin
```

Integration between Docker-host and Jenkins

-Install "publish Over SSH"

Manage Jenkins > Manage Plugins > Available > Publish over SSH

-Enable connection between Docker-host and Jenkins

Manage Jenkins > Configure System > Publish Over SSH > SSH Servers

SSH Servers: - Name: docker-host

Hostname:<ServerIP>

username: dockeradmin

Advanced > choose Use password authentication, or use a different key

password: *****

Create the Jenkins job

From Jenkins home page select "New Item"

Enter an item name: Nodejs

Source Code Management:

Repository: <https://gitlab.com/neealink/fml-voip.git>

Branches to build : */master

Poll SCM : - * * * * *

Build Environment:

Check> Send files or execute commands over SSH after the build runs

SSH Publishers

SSH Server Name: docker-host

Transfers > Transfer set

Source files: **/**

Remove prefix: webapp/target

Remote directory: .

Exec command:

cd /home/dockeradmin; docker build -t nodejs-image .; docker run -d --name nodejs-container -p 3000:3000 nodejs-image;

Check>Provide Node & npm bin/ folder to PATH

NodeJS Installation: Node-Build

Nprmc file: -use system default-

Build:

```
Command> npm install  
npm install
```

Apply and Save. Build!!

N/B- Jenkins cannot manage deployments very well, cannot deploy new container with same name

docker: Error response from daemon: Conflict. The container name "/nodejs-container" is already in use by container "ecb4755a182b29b299820be3329018ef9ccf390bcacc4a3dc48c441d9c54ef5f". You have to remove (or rename) that container to be able to reuse that name.

Areas of Improvement- To use a different deployment Tool such as Ansible, set up ansible environment, integrate ansible with jenkins, write ansible playbook to deploy on container