DevOps

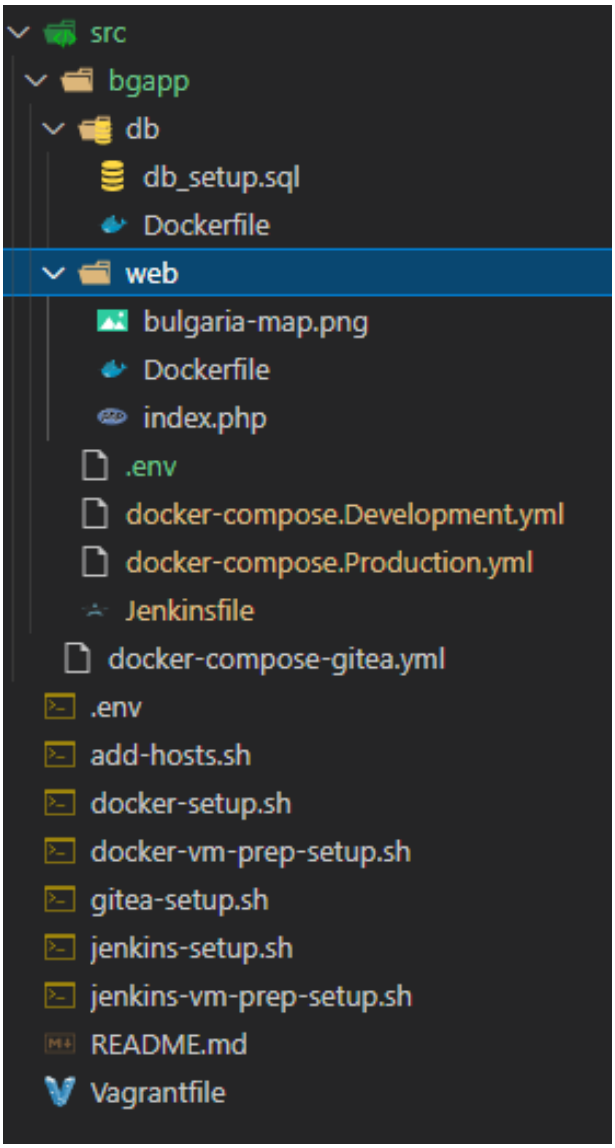
Containerization, CI/CD & Monitoring

February 2022

Advanced CI/CD with Jenkins

Home Work

Stefan Veselinov



Assignment

You are expected to create the following

- A setup with two virtual machines – one with **Jenkins** and another with **Docker** installed just like on the practice

** You can adjust their parameters in order to fit within your available resources*

- On the **Docker** machine you must deploy **Gitea** (as we did during the practice)
- On the **Jenkins** machine create a pipeline to build the **BGApp** application.

Preparation

- Host Machine - Windows 10 Pro 21H2 19044.1526
- Creating Project structure

  - src/ folder that is going to be shared throughout vm's (contains bgapp source code and all files for running and deploying the app)

  - root – Vagrant file and all .sh scripts for vm's setup

- Vagrantfile – create two vm's.
  - 1-st – Jenkins – vm that is going to run Jenkins Ui, controller of docker vm agent
  - 2-nd Jenkins- vm that is going to be a Jenkins agent, executes docker and docker compose commands, runs Gitea

On vagrant up, the two vm's are createt, provisioned with tooling, setting up the users permissions and deploy Gitea CMS

```
Vagrant.configure(2) do |config|
  # enable use of .env file
  config.env.enable
  config.ssh.insert_key = false

  config.vm.define ENV['JENKINS_VM_DEFINE_NAME'] do |jenkins|
    jenkins.vm.box=ENV['JENKINS_VAGRANT_BOX_NAME']
    jenkins.vm.hostname = ENV['JENKINS_VM_HOST_NAME']
    jenkins.vm.network "private_network", ip: ENV['JENKINS_VM_IP']
    jenkins.vm.network "forwarded_port", guest: 8080, host: 8080
    jenkins.vm.provision "shell", path: ENV['ADD_HOSTS_SCRIPT']
    jenkins.vm.provision "shell", path: ENV['JENKINS_VM_PROVISION_SCRIPT']
    jenkins.vm.provision "shell", path: ENV['JENKINS_VM_PROVISION_JENKINS_SCRIPT']
    jenkins.vm.provider :virtualbox do |vb|
      vb.memory = ENV['JENKINS_VB_MEMORY']
      vb.cpus = ENV['JENKINS_VB_CPUS']
      vb.customize ["modifyvm", :id, "--usb", "off"]
```

```
      vb.customize ["modifyvm", :id, "--usbehci", "off"]
    end
  end

  config.vm.define ENV['DOCKER_VM_DEFINE_NAME'] do |docker|
        docker.vm.box=ENV['DOCKER_VAGRANT_BOX_NAME']
        docker.vm.hostname = ENV['DOCKER_VM_HOST_NAME']
        docker.vm.network "private_network", ip: ENV['DOCKER_VM_IP']
        docker.vm.provision "shell", path: ENV['ADD_HOSTS_SCRIPT']
        docker.vm.provision "shell", path: ENV['DOCKER_VM_PROVISION_SCRIPT']
        docker.vm.provision "shell", path: ENV['DOCKER_VM_PROVISION_DOCKER_SCRIPT']
        docker.vm.provider :virtualbox do |vb|
          vb.memory = ENV['DOCKER_VB_MEMORY']
          vb.cpus = ENV['DOCKER_VB_CPUS']
          vb.customize ["modifyvm", :id, "--usb", "off"]
          vb.customize ["modifyvm", :id, "--usbehci", "off"]
        end
    end
end
```

- Vagrant .env file

```
export JENKINS_VAGRANT_BOX_NAME=shekeriev/debian-11
export JENKINS_VB_MEMORY=1024
export JENKINS_VB_CPUS=1
export JENKINS_VM_DEFINE_NAME=jenkins
export JENKINS_VM_HOST_NAME=jenkins.do1.lab
export JENKINS_VM_IP=192.168.99.100
export JENKINS_VM_PROVISION_SCRIPT=jenkins-vm-prep-setup.sh
export JENKINS_VM_PROVISION_JENKINS_SCRIPT=jenkins-setup.sh

export DOCKER_VAGRANT_BOX_NAME=shekeriev/debian-11
export DOCKER_VB_MEMORY=1024
export DOCKER_VB_CPUS=1
export DOCKER_VM_DEFINE_NAME=docker
export DOCKER_VM_HOST_NAME=docker.do1.lab
export DOCKER_VM_IP=192.168.99.101
export DOCKER_VM_PROVISION_SCRIPT=docker-vm-prep-setup.sh
export DOCKER_VM_PROVISION_DOCKER_SCRIPT=docker-setup.sh

export ADD_HOSTS_SCRIPT=add-hosts.sh
```

- ADD_HOSTS_SCRIPT

```
echo "* Add hosts ..."
echo "192.168.99.100 jenkins.do1.lab jenkins" >> /etc/hosts
echo "192.168.99.101 docker.do1.lab docker" >> /etc/hosts
```

- JENKINS_VM_PROVISION_SCRIPT

```
echo "* Install Software ..."
sudo apt-get -y install git
```

- JENKINS_VM_PROVISION_JENKINS_SCRIPT

```
echo "* Update System"
sudo apt-get -y update

echo "* Install Java"
sudo apt-get -y install fontconfig openjdk-11-jre

echo "* Get Jenkins Repo Keys"
sudo curl -fsSL https://pkg.jenkins.io/debian/jenkins.io.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null

echo "* Add Jenkins Repository"
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
```

```
echo "* Install Jenkins"
sudo apt-get -y update
sudo apt-get -y install jenkins

echo "* Add Passwd to Jenkins"
echo -e "Password1\nPassword1" | sudo passwd Jenkins
```

- DOCKER_VM_PROVISION_SCRIPT

```
echo "* Install Software ..."
sudo apt-get -y install git fontconfig openjdk-11-jre

echo "Setup Jenkins User"
sudo useradd -m -p $(echo Password1 | openssl passwd -1 -stdin) -s /bin/bash Jenkins
```

- DOCKER_VM_PROVISION_DOCKER_SCRIPT

```
echo "* Update System"
sudo apt-get -y update

echo "* Install Docker"
sudo apt-get -y install ca-certificates curl gnupg lsb-release

curl -fsSL https://download.docker.com/linux/debian/gpg \
      | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

echo \
      "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/debian \
        $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get -y update
sudo apt-get -y install docker-ce docker-ce-cli containerd.io

echo "* Add Docker Permissions"
sudo usermod -aG docker vagrant

echo "* Add Jenkins - Docker group"
echo 'jenkins ALL=(ALL)    NOPASSWD: ALL' | sudo EDITOR='tee -a' visudo
sudo usermod -aG docker jenkins

echo "* Adding Docker-Compose"
curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o
/usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose

echo "* Doker-Compose - Starting Gitea"
sudo docker-compose -f /vagrant/src/docker-compose-gitea.yml up –d
```

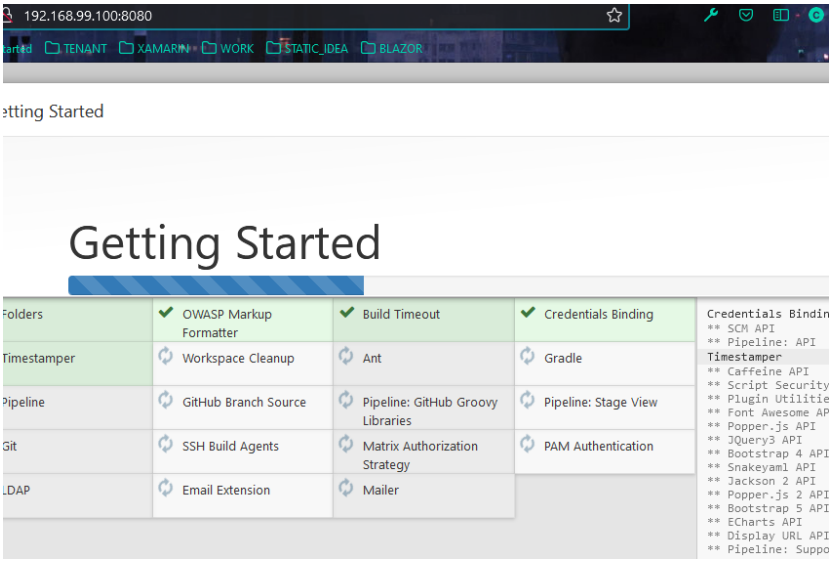
- Running Vagrant up, checking for src code in Docker vm

```
vagrant@docker:~$ cd /vagrant/
vagrant@docker:/vagrant$ pwd
/vagrant
vagrant@docker:/vagrant$ ls -la
total 30
drwxrwxrwx  1 vagrant vagrant 4096 Mar 10 08:36 .
drwxr-xr-x 19 root    root    4096 Mar 10 08:37 ..
-rwxrwxrwx  1 vagrant vagrant  153 Mar  6 07:57 add-hosts.sh
-rwxrwxrwx  1 vagrant vagrant 1237 Mar  6 13:55 docker-setup.sh
-rwxrwxrwx  1 vagrant vagrant  210 Mar  6 08:51 docker-vm-prep-setup.sh
-rwxrwxrwx  1 vagrant vagrant  735 Mar  6 13:53 .env
-rwxrwxrwx  1 vagrant vagrant  119 Mar  6 13:52 gitea-setup.sh
-rwxrwxrwx  1 vagrant vagrant  660 Mar  6 16:31 jenkins-setup.sh
-rwxrwxrwx  1 vagrant vagrant   71 Mar  6 16:31 jenkins-vm-prep-setup.sh
-rwxrwxrwx  1 vagrant vagrant 1383 Mar 10 08:27 README.md
drwxrwxrwx  1 vagrant vagrant    0 Mar  6 13:43 src
drwxrwxrwx  1 vagrant vagrant    0 Mar  6 16:41 .vagrant
-rwxrwxrwx  1 vagrant vagrant 1712 Mar  6 13:54 Vagrantfile
```

- Prep for making docker vm agent

```
vagrant@jenkins:/vagrant$ su - jenkins
Password:
jenkins@jenkins:~$ ssh-keygen -t rsa -m PEM
Generating public/private rsa key pair.
Enter file in which to save the key (/var/lib/jenkins/.ssh/id_rsa):
Created directory '/var/lib/jenkins/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/lib/jenkins/.ssh/id_rsa
Your public key has been saved in /var/lib/jenkins/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:SmsLZ0WnbRobpr2eM3+IZFLeU78A1LTonGPKccYMMFw jenkins@jenkins
The key's randomart image is:
+---[RSA 3072]----+
|      .o.E  o.    |
|       .o  ....   |
|         o.o .    |
|        ..X...    |
|       .oS.&o .   |
|      ..O+%o.. .  |
|     . B+*. o . . |
|      = .+o. . .  |
|       ..++..     |
+----[SHA256]-----+
jenkins@jenkins:~$ ssh-copy-id jenkins@docker.do1.lab
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/var/lib/j
The authenticity of host 'docker.do1.lab (192.168.99.101)' can't be estab
ECDSA key fingerprint is SHA256:7iJ3nMHok0dHOUluoaBIaKg1ddmSp4wBJ0qsA2eiq
Are you sure you want to continue connecting (yes/no/[fingerprint])?
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to
The authenticity of host 'docker.do1.lab (192.168.99.101)' can't be estab
ECDSA key fingerprint is SHA256:7iJ3nMHok0dHOUluoaBIaKg1ddmSp4wBJ0qsA2eiq
Are you sure you want to continue connecting (yes/no/[fingerprint])?

/usr/bin/ssh-copy-id: ERROR: Host key verification failed.

jenkins@jenkins:~$ hostname
jenkins
jenkins@jenkins:~$ exit
logout
vagrant@jenkins:/vagrant$
```

- Jenkins set up - Visiting http://192.168.99.100:8080/ and running try Jenkins installation process



- Creating Jenkins user ang logging in
- Creating ssh credentials for docker

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
8805c93f-a665-4616-898e-e62a33286ce5	jenkins (SSH-Uname/PrivateKey)	SSH Username with private key	SSH-Uname/PrivateKey

- Set up docker vm as agent

Labels ?

docker-node

Usage ?

Only build jobs with label expressions matching this node

Launch method ?

Launch agents via SSH

Host ?

docker.do1.lab

Credentials ?

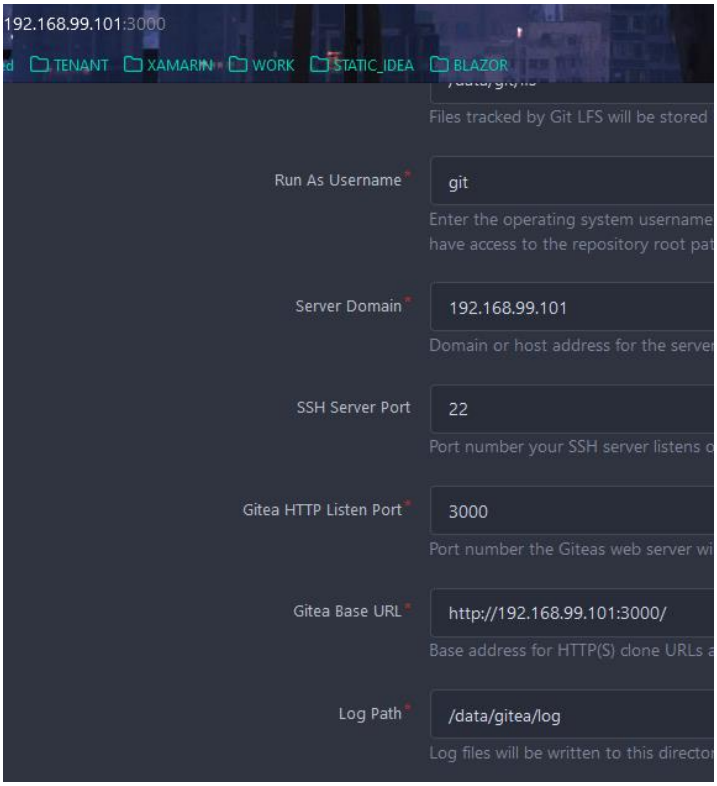
jenkins (SSH- Uname/Private key)

Add

Host Key Verification Strategy ?

Known hosts file Verification Strategy

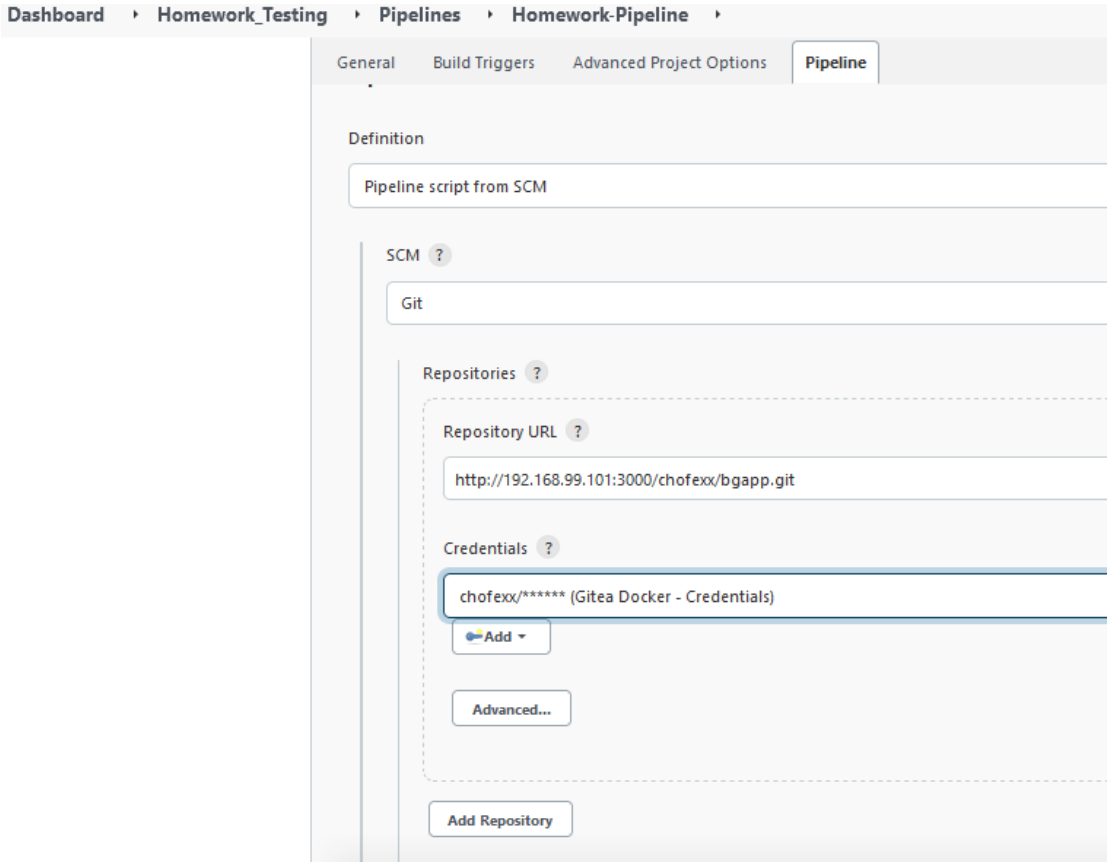
- Adding Gitea plugin
- Visiting http://192.168.99.101:3000/ and setting up Gitea



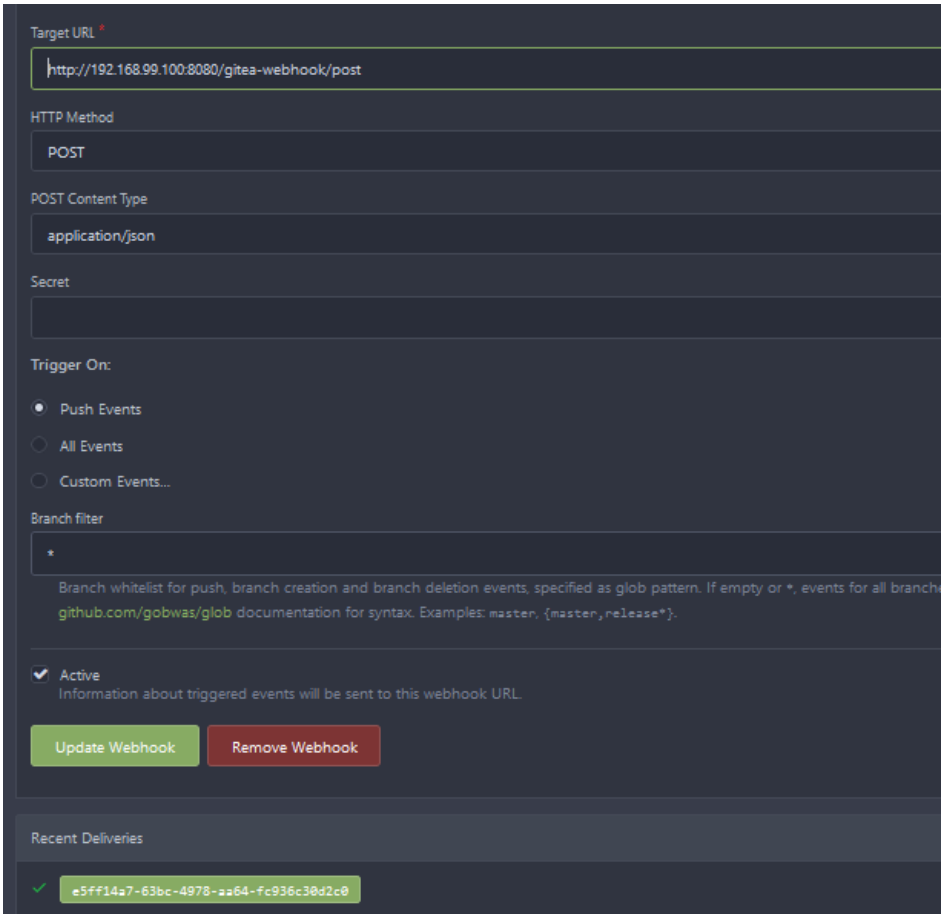
- Creating Gitea user
- Creating bgapp repository
- Docker vm – Commiting to Gitea repository

```
vagrant@docker:/vagrant/src/bgapp$ ls -la
total 5
drwxrwxrwx 1 vagrant vagrant    0 Mar  6 08:31 .
drwxrwxrwx 1 vagrant vagrant    0 Mar  6 13:43 ..
drwxrwxrwx 1 vagrant vagrant    0 Mar  6 08:08 db
-rwxrwxrwx 1 vagrant vagrant  424 Mar  6 08:30 docker-compose.Development.yml
-rwxrwxrwx 1 vagrant vagrant  397 Mar  6 08:30 docker-compose.Production.yml
-rwxrwxrwx 1 vagrant vagrant 1661 Mar  6 08:32 Jenkinsfile
drwxrwxrwx 1 vagrant vagrant    0 Mar  6 08:11 web
vagrant@docker:/vagrant/src/bgapp$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /vagrant/src/bgapp/.git/
vagrant@docker:/vagrant/src/bgapp$ git add *
vagrant@docker:/vagrant/src/bgapp$ git commit -m "Initial Commit from Jenkins Agent"
[master (root-commit) 10b450c] Initial Commit from Jenkins Agent
 Committer: Vagrant User <vagrant@docker.do1.lab>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 8 files changed, 208 insertions(+)
 create mode 100644 Jenkinsfile
 create mode 100644 db/Dockerfile
 create mode 100644 db/db_setup.sql
 create mode 100644 docker-compose.Development.yml
 create mode 100644 docker-compose.Production.yml
 create mode 100644 web/Dockerfile
 create mode 100644 web/bulgaria-map.png
 create mode 100644 web/index.php
vagrant@docker:/vagrant/src/bgapp$ git remote add origin http://192.168.99.101:3000/chofexx/bgapp.git
git push -u origin master
Username for 'http://192.168.99.101:3000': chofexx
Password for 'http://chofexx@192.168.99.101:3000':
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Compressing objects: 100% (12/12), done.
Writing objects: 100% (12/12), 15.25 KiB | 867.00 KiB/s, done.
Total 12 (delta 0), reused 0 (delta 0), pack-reused 0
remote: . Processing 1 references
remote: Processed 1 references in total
To http://192.168.99.101:3000/chofexx/bgapp.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
vagrant@docker:/vagrant/src/bgapp$
```

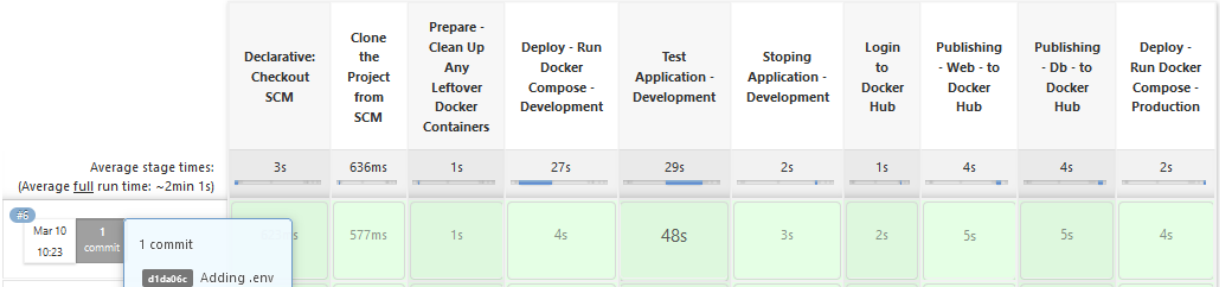
- Jenkins vm – Creating credentials (Username – Password) for Gitea
- Creating Credentials for Docker Hub
- Creating Jenkins job
    - enable - GitHub hook trigger for GITScm polling
    - enable – Poll scm
    - select Pipeline – pipeline script from SCM



- Enable Gitea repo Webhook

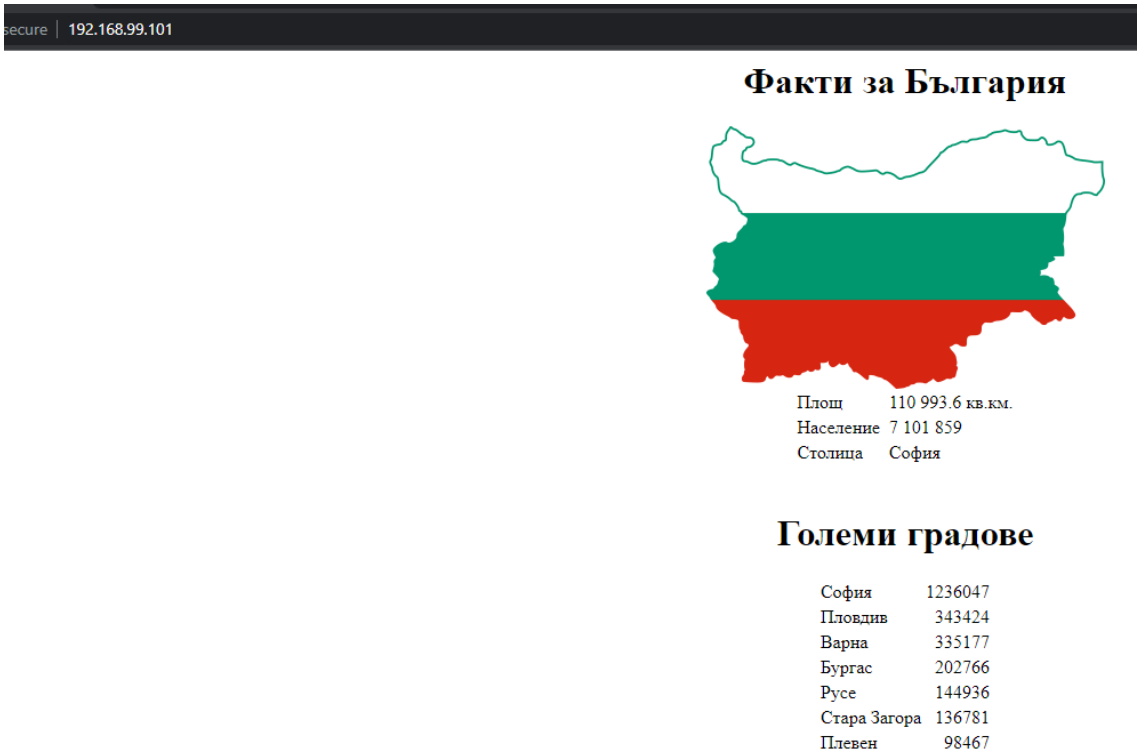


- Docker vm – adding and commiting .env file



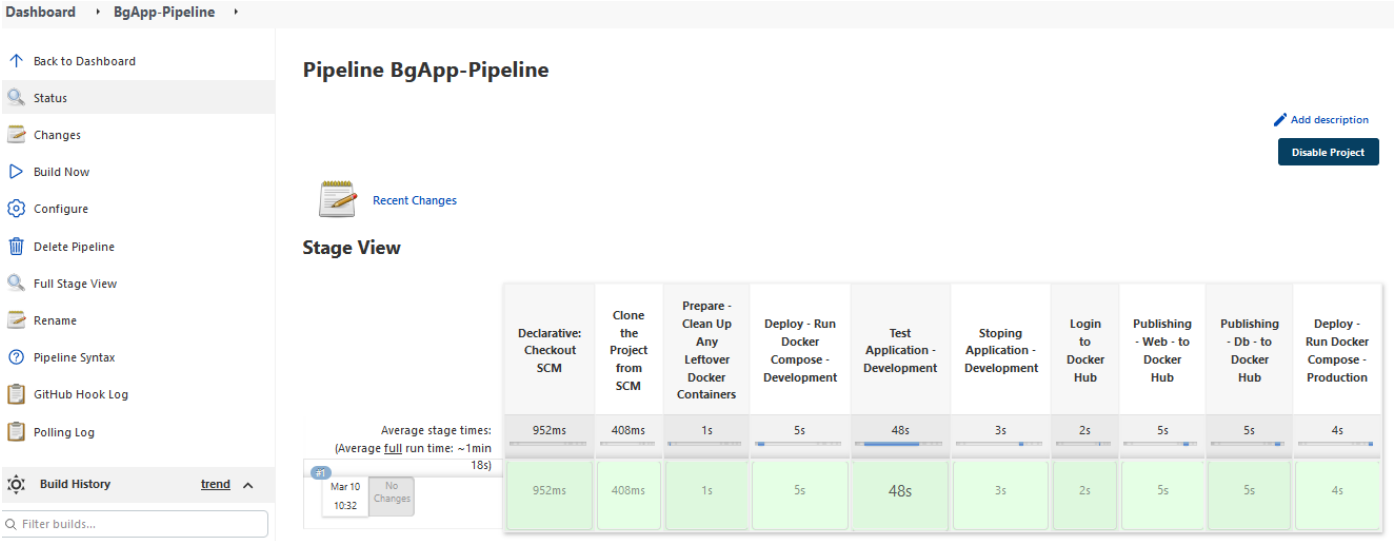
	Declarative: Checkout SCM	Clone the Project from SCM	Prepare - Clean Up Any Leftover Docker Containers	Deploy - Run Docker Compose - Development	Test Application - Development	Stoping Application - Development	Login to Docker Hub	Publishing - Web - to Docker Hub	Publishing - Db - to Docker Hub	Deploy - Run Docker Compose - Production
Average stage times: (Average full run time: ~2min 1s)	3s	636ms	1s	27s	29s	2s	1s	4s	4s	2s
#6 Mar 10 10:23 — 1 commit — 1 commit d1da06c Adding .env	[illegible]	577ms	1s	4s	48s	3s	2s	5s	5s	4s

- Checking app



- Docker vm – use of Jenkins CLI
  - Installing Jenkins CLI - wget http://192.168.99.100:8080/jnlpJars/jenkins-cli.jar
  - Getting the job - java -jar jenkins-cli.jar -s http://192.168.99.100:8080/ -auth admin:password get-job Homework_Testing/Pipelines/Homework-Pipeline > /vagrant/src/bgapp/bgapp-pipeline-job.xml
  - Commiting to repository
  - Deleting Jenkins items - java -jar jenkins-cli.jar -s http://192.168.99.100:8080/ -auth admin:password delete-job Homework_Testing
  - Creating job from exported xml - java -jar jenkins-cli.jar -s http://192.168.99.100:8080/ -auth admin:password create-job BgApp-Pipeline < bgapp-pipeline-job.xml
  - Building the job

```
vagrant@docker:/vagrant/src/bgapp$ java -jar jenkins-cli.jar -s http://192.168.99.100:8080/ -auth admin:password build BgApp-Pipeline -f -v
Started BgApp-Pipeline #1
Started from command line by admin
Obtained Jenkinsfile from git http://192.168.99.101:3000/chofexx/bgapp.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on docker-node in /home/jenkins/workspace/BgApp-Pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
The recommended git tool is: NONE
using credential 21802602-900d-44ea-913a-c985184365c3
Cloning the remote Git repository
Avoid second fetch
Checking out Revision 72ca8e68674c9e666a0681f7bcc0dd8ef9517e6b (refs/remotes/origin/master)
Commit message: "Adding Jenkins xml job"
First time build. Skipping changelog.
[Pipeline] }
```



Other Used Files

- Db Dockerfile

```
FROM mariadb:10.7

ADD ./db_setup.sql /docker-entrypoint-initdb.d/init.sql
```

- Web Dockerfile

```
FROM php:8.0-apache

RUN docker-php-ext-install pdo pdo_mysql
```

- docker-compose.Development.yml

```
version: "3.8"

services:

    web:

      image: bgapp_web

      build:

        context: ./web

        dockerfile: Dockerfile

      container_name: bgapp_web

      ports:

          - 8080:80

      volumes:

          - ./web/:/var/www/html/

      networks:

          - bgapp-app-dev-network

      depends_on:

          - db

    db:

      image: bgapp_db

      build:

        context: ./db

        dockerfile: Dockerfile

      container_name: bgapp_db
```

```
      networks:
          - bgapp-app-dev-network
      environment:
          MYSQL_ROOT_PASSWORD: "${DB_PASSWORD}"
networks:
    bgapp-app-dev-network:
```

- docker-compose.Production.yml

```
version: "3.8"
services:
    web:
        image: veselinovstf/bgapp-jenkins-web
        container_name: bgapp_web
        ports:
            - 80:80
        volumes:
            - ./web/:/var/www/html/
        networks:
            - bgapp-app-prod-network
        depends_on:
            - db
    db:
        image: veselinovstf/bgapp-jenkins-db
        networks:
            - bgapp-app-prod-network
        container_name: bgapp_db
        environment:
            MYSQL_ROOT_PASSWORD: "${DB_PASSWORD}"
networks:
    bgapp-app-prod-network:
```

- Jenkinsfile

```
pipeline
{
    agent
    {
        label 'docker-node'
    }

    environment
    {
        DOCKERHUB_CREDENTIALS=credentials('docker-hub-credentials')
    }
```

```
    stages
    {
        stage('Clone the Project from SCM')
        {
            steps
            {
                git branch: 'master', url: 'http://192.168.99.101:3000/chofexx/bgapp'
            }
        }

        stage("Prepare - Clean Up Any Leftover Docker Containers") {

            steps {
                sh '''
                    docker container rm -f bgapp_db || true
                    docker container rm -f bgapp_web || true
                '''
            }
        }

        stage("Deploy - Run Docker Compose - Development") {

            steps {
                sh '''
                    docker-compose -f docker-compose.Development.yml up -d --build
                '''
            }
        }

        stage("Test Application - Development") {

            steps {
                echo 'Test #1 - reachability'
                sh 'echo $(curl --write-out "%{http_code}" --silent --output /dev/null http://localhost:8080) | grep
200'

                echo 'Test #2 - Факти за България'
                sh "curl --silent http://localhost:8080 | grep 'Факти за България'"

                echo 'Test #3 - Db Connection - wait 60s and curl Пловдив'
                sh "sleep 40s && curl --silent --connect-timeout 60 http://localhost:8080 | grep 'Русе'"
            }
        }

        stage("Stoping Application - Development") {
```

```
            steps {
                sh '''
                   docker-compose -f docker-compose.Development.yml down
                '''
            }
        }

        stage('Login to Docker Hub')
        {
            steps
            {
                sh 'echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin'
            }
        }

        stage("Publishing - Web - to Docker Hub") {

            steps {
                sh 'docker image tag bgapp_web veselinovstf/bgapp-jenkins-web'
                sh 'docker push veselinovstf/bgapp-jenkins-web'
            }
        }

        stage("Publishing - Db - to Docker Hub") {

            steps {
                sh 'docker image tag bgapp_db veselinovstf/bgapp-jenkins-db'
                sh 'docker push veselinovstf/bgapp-jenkins-db'
            }
        }

        stage("Deploy - Run Docker Compose - Production") {

            steps {
                sh 'docker-compose -f docker-compose.Production.yml up -d'
            }
        }

    }
}
```