

**COMSATS University Islamabad, Attock Campus****Course: TOCS**

<b>Course</b>	TOCS
<b>Instructor</b>	Dr. Farhan Adil

**Group Members Details**

<b>Registration No</b>	<b>Name</b>
Sp20-bcs-008	Shahid iqbal

## 1. Docker Containerization:

### 1. Create a Simple Web Application:

Create a simple web application, for example, a "Hello World" Flask application or any other sample website template.

### 2. Write a Dockerfile:

- Create a `Dockerfile` in the root of your project to define the containerization process.

```
``Dockerfile

FROM python:3.8-alpine

WORKDIR /app

COPY . .

RUN pip install -r requirements.txt # Install dependencies if needed

CMD ["python", "app.py"]

``
```

### 3. Build and Push Docker Image:

- Build the Docker image and push it to Docker Hub.

```
``bash

docker build -t your-dockerhub-username/your-web-app:latest .

docker push your-dockerhub-username/your-web-app:latest

``
```

## 2. Jenkins CI/CD Pipeline:

### 1. Install Jenkins:

- Install Jenkins on your local machine or a dedicated server. Follow the official [Jenkins installation guide](<https://www.jenkins.io/doc/book/installing/>) for your specific environment.

## 2. Set Up Jenkins Job:

- Open Jenkins and create a new freestyle project.
- Configure the job to pull the source code from your version control system (e.g., GitHub).

## 3. Configure Jenkins Job:

- Set up the Git repository URL and credentials.
- Configure the build step to build the Docker image using the Dockerfile.
- Implement a post-build step to push the Docker image to Docker Hub.

## 4. Define CI/CD Pipeline Stages:

- Add the following stages to your pipeline:
  - **\*\*Build:\*\*** Build the Docker image.
  - **\*\*Test:\*\*** Implement your testing steps.
  - **\*\*Deploy:\*\*** Deploy the Docker image to AWS EC2 instance.
- Implement rollback functionality in case of a failed deployment.

## 5. Configure Email Notifications:

- Add email notification post-build actions to notify the team in case of a failed deployment.

## 3. AWS EC2 Instance:

### 1. Create an AWS EC2 Instance:

- Log in to your AWS Management Console.
- Launch an EC2 instance (Ubuntu Server) and configure security groups to allow traffic on port 80.
- Configure necessary security groups, key pairs, and network settings.

### 2. Install Docker on EC2 Instance:

- Connect to your EC2 instance using SSH.
- Install Docker on the EC2 instance:

```
``bash
sudo apt update
sudo apt install docker.io
sudo systemctl start docker
sudo systemctl enable docker
``
```

### 3. Pull and Run Docker Image:

- Pull the Docker image from Docker Hub and run it on the EC2 instance.

```
``bash
docker pull your-dockerhub-username/your-web-app:latest
docker run -d -p 80:80 your-dockerhub-username/your-web-app:latest
``
```

## 4. Pipeline Automation:

### 1. Trigger Pipeline Automatically:

- Configure Jenkins to trigger the pipeline automatically whenever changes are pushed to the version control system.
- Use webhooks or polling to detect changes in your Git repository.

### 2. Implement Deployment Steps:

- Modify the Jenkins job to SSH into the AWS EC2 instance and update the running Docker container.
- You can use SSH plugins in Jenkins to execute commands on the remote server.

## AWS Console CMD

```
sudo apt update
```

```
sudo apt upgrade
```

```
sudo apt install default-jdk
```

```
java -version
```

```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
```

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

```
sudo apt update
```

```
sudo apt install jenkins
```

```
sudo systemctl start jenkins
```

```
sudo systemctl enable jenkins
```

```
sudo systemctl status jenkins
```

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

## #Enable Port in SecurityGroup

### #add ssh in rules

http://51.20.140.172:8080 go to this open RemoteJenkins

## ##NoW JenkinsPart

LocalDesktop Part

```
chmod 400 MyUbuntuServer.pem
```

```
ssh -i MyUbuntuServer.pem ubuntu@ec2-51-20-140-172.eu-north-1.compute.amazonaws.com
```

### **#ADD WEBHOOK**

```
http://ec2-51-20-140-172.eu-north-1.compute.amazonaws.com:8080/github-webhook/
```

### **#add PublishOver SSH IN JENKINS**

#### **#add ssh for interactions**

#### **#go to send artifact over ssh**

### **#give permission**

```
sudo chown -R ubuntu:ubuntu /home/ubuntu/python
```

```
sudo chmod -R u+w /home/ubuntu/python
```

### **#Termianl**

Install Jenkins

### **Plugins:**

Publish Over SSH

DockerPlugins

Email Extension

### **# Add Docker Credential in Jenkins Server**

### **#Add Github Webhook**

### **#Allow Port in Security Group of Instance**

### **#in repo Add Jenkins File and Docker file**

### **#Commands Ubuntu**

```
sudo apt update
```

```
sudo apt install docker.io
```

```
sudo systemctl start docker
```

```
sudo systemctl enable docker
```

### **# Add Docker's official GPG key:**

```
sudo apt-get update
```

```
sudo apt-get install ca-certificates curl gnupg
```

```
sudo install -m 0755 -d /etc/apt/keyrings
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o  
/etc/apt/keyrings/docker.gpg
```

```
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

### **# Add the repository to Apt sources:**

```
echo \
```

```
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]  
https://download.docker.com/linux/ubuntu \
```

```
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
```

```
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update
```

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

```
sudo usermod -aG docker jenkins
```

```
sudo usermod -aG docker ubuntu
```

```
sudo systemctl restart jenkins
```

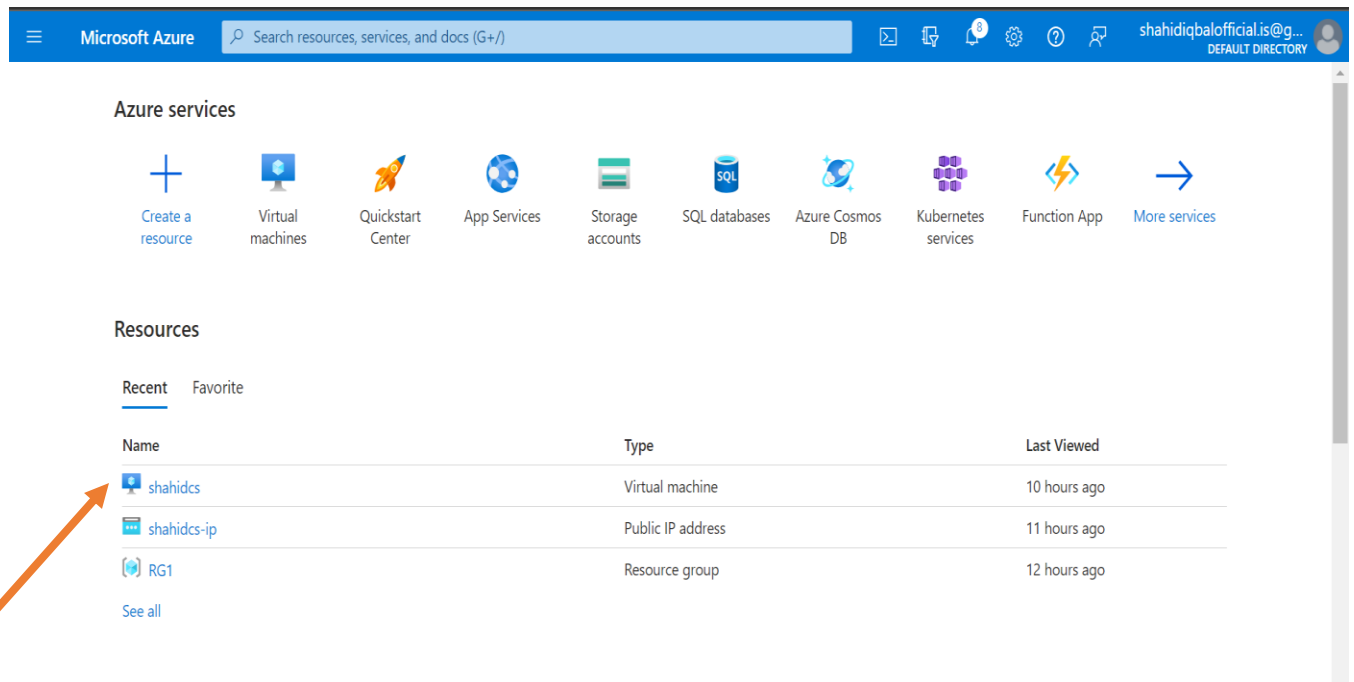
```
sudo lsof -i :80
```

# or

```
sudo netstat -tulnp | grep :80
```

```
sudo reboot
```

## create instance on Azure



Microsoft Azure Search resources, services, and docs (G+)

shahidiqbalofficial.is@g...  
DEFAULT DIRECTORY

### Azure services

Create a resource Virtual machines Quickstart Center App Services Storage accounts SQL databases Azure Cosmos DB Kubernetes services Function App More services

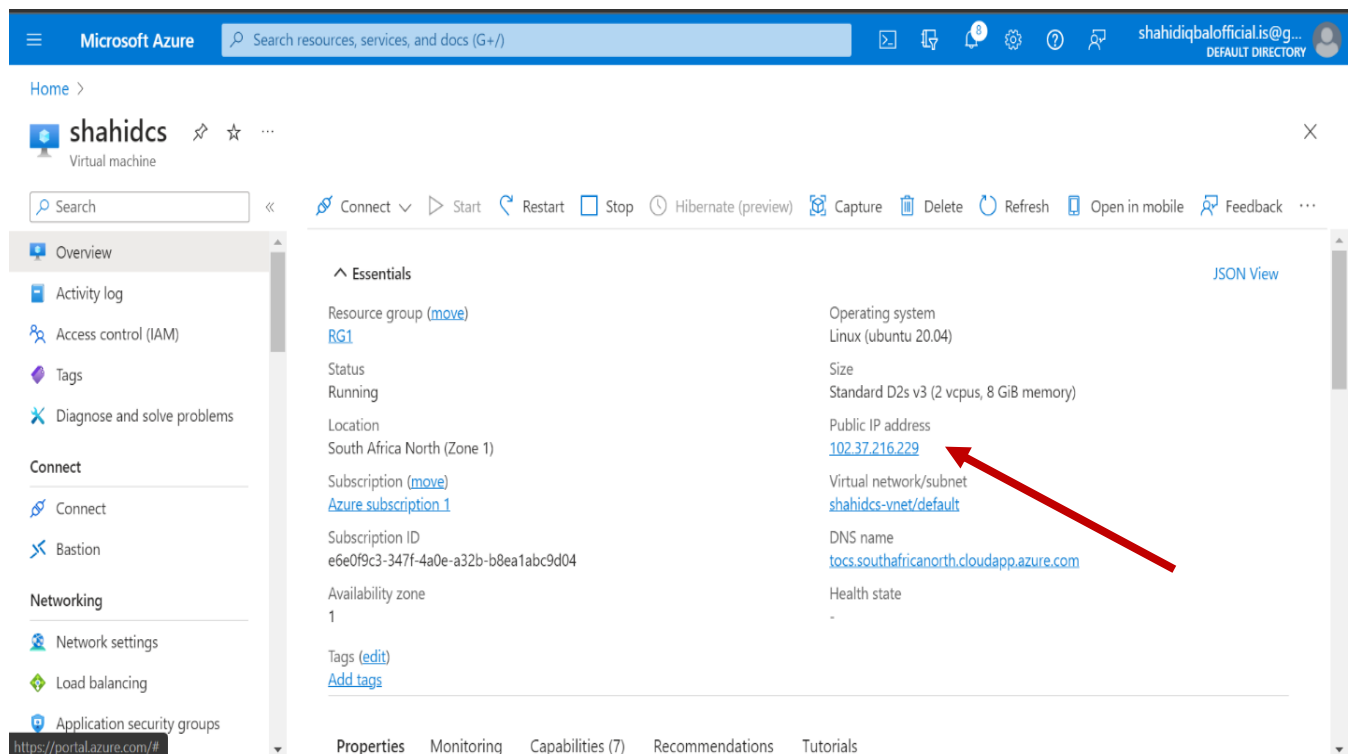
### Resources

Recent Favorite

Name	Type	Last Viewed
shahidcs	Virtual machine	10 hours ago
shahidcs-ip	Public IP address	11 hours ago
RG1	Resource group	12 hours ago

See all

## User Details, instance:(shahidcs)



Microsoft Azure Search resources, services, and docs (G+)

shahidiqbalofficial.is@g...  
DEFAULT DIRECTORY

Home >

### shahidcs

Virtual machine

Search

Connect Start Restart Stop Hibernate (preview) Capture Delete Refresh Open in mobile Feedback

#### Overview

- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems

#### Connect

- Connect
- Bastion

#### Networking

- Network settings
- Load balancing
- Application security groups

#### Essentials

Resource group (move) [RG1](#)

Status: Running

Location: South Africa North (Zone 1)

Subscription (move) [Azure subscription 1](#)

Subscription ID: e6e0f9c3-347f-4a0e-a32b-b8ea1abc9d04

Availability zone: 1

Tags (edit) [Add tags](#)

Operating system: Linux (ubuntu 20.04)

Size: Standard D2s v3 (2 vcpus, 8 GiB memory)

Public IP address: [102.37.216.229](#)

Virtual network/subnet: [shahidcs-vnet/default](#)

DNS name: [tocs.southafricanorth.cloudapp.azure.com](#)

Health state: -

[JSON View](#)

<https://portal.azure.com/#>

Properties Monitoring Capabilities (7) Recommendations Tutorials



## Here we are connecting our instance

 [Connect](#)  [Start](#)  [Restart](#)  [Stop](#)  [Hibernate \(preview\)](#)  [Capture](#)  [Delete](#)  [Refresh](#)  [Open in mobile](#)  [Feedback](#) ...

**Properties** Monitoring Capabilities (7) Recommendations Tutorials

### Virtual machine

Computer name	shahidcs
Operating system	Linux (ubuntu 20.04)
Image publisher	canonical
Image offer	0001-com-ubuntu-server-focal
Image plan	20_04-lts-gen2
VM generation	V2
VM architecture	x64
Agent status	Ready
Agent version	2.9.1.1
Hibernation	Disabled
Host group	-
Host	-

### Networking

Public IP address	102.37.216.229 ( Network interface shahidcs962_z1 )
Public IP address (IPv6)	-
Private IP address	10.0.0.4
Private IP address (IPv6)	-
Virtual network/subnet	shahidcs-vnet/default
DNS name	tocs.southafricanorth.cloudapp.azure.com

### Size

Size	Standard D2s v3
vCPUs	2
RAM	8 GiB

## Admin name (azureuser)

Microsoft Azure Search resources, services, and docs (G+/)

Home > shahidcs

### shahidcs | Connect

Virtual machine

Search

Refresh Troubleshoot More Options Feedback

Connecting using  
Public IP address | 102.37.216.229

Admin username  
azureuser

Port (change)  
22 [Check access](#)

Just-in-time policy  
Unsupported by plan

Recommended

SSH using Azure CLI

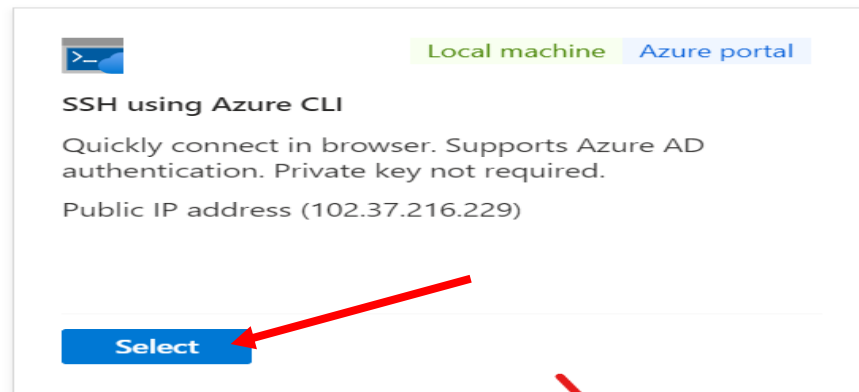
Quickly connect in browser. Supports Azure AD authentication. Private key not required.

Most common

Native SSH


No additional software needed. Private key required for connection. Best for those with existing SSH tools.

## Here we select (SSH using Azure CLI)



### SSH using Azure CLI

Connect from the Azure portal

 Connect from your local machine

#### 1 Configure prerequisites for SSH using Azure CLI

Azure needs to configure some features in order to connect to the VM.

##### ✓ Prerequisites configured

##### ✓ System assigned managed identity

Azure will configure a system-assigned managed identity in order to enable the Azure AD login extension. [Learn more](#)

##### ✓ Azure AD SSH Login Extension

The Azure Active Directory based SSH Login extension will securely connect to the VM using Azure AD instead of SSH or a username and password. [Learn more](#)

##### ✓ Virtual machine user or administrator login

A virtual machine administrator login role on the resource group will allow login to the virtual machine via CloudShell. [Learn more](#)

##### ✓ Port 22 access

Port 22 is accessible on this virtual machine for all configured IPs. [Learn more](#)

 Change the port for connecting to this virtual machine on the

Bash

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.  
See <https://ubuntu.com/esm> or run: `sudo pro status`

New release '22.04.3 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Dec 27 06:35:48 2023 from 4.175.91.72

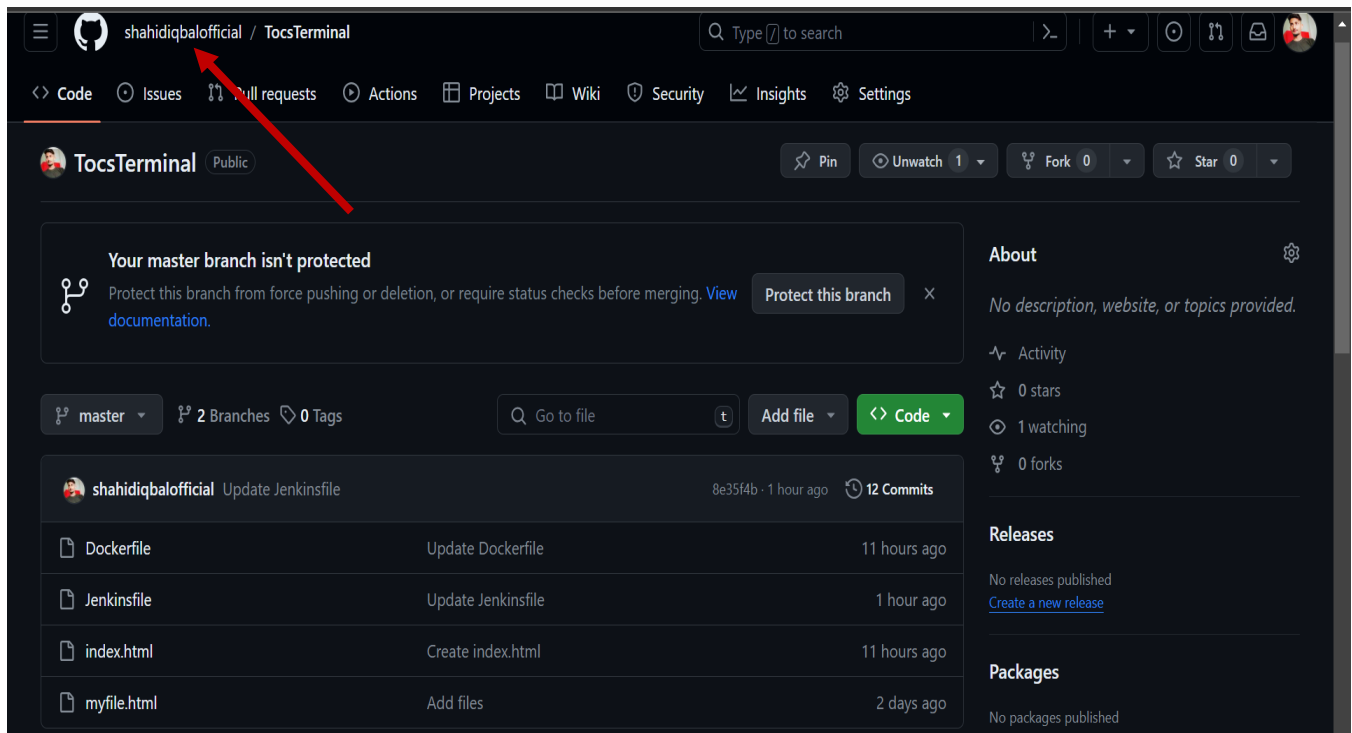
shahidiqbalofficial.is@gmail.com@shahidcs:~\$

shahidiqbalofficial.is@gmail.com@shahidcs:~\$ whoami

shahidiqbalofficial.is@gmail.com

shahidiqbalofficial.is@gmail.com@shahidcs:~\$

## Github Account shahidiqbalofficial



### Jenkins files code

```

pipeline {
    agent any

    stages {
        stage('Build') {
            steps {
                script {
                    dockerImage = docker.build("shahidiqbal008/distance-converter:${env.BUILD_ID}")
                }
            }
        }
    }

    stage('Push') {
        steps {
            script {
                docker.withRegistry('https://registry.hub.docker.com', 'docker-hub-credentials') {

```

```

        dockerImage.push()
    }
}
}
}
stage('Test') {
    steps {
        sh 'ls -l index.html'
    }
}
stage('Deploy') {
    steps {
        script {
            sshPublisher(
                publishers: [
                    sshPublisherDesc(
                        configName: "shahidcs",
                        transfers: [sshTransfer(
                            execCommand: ""
                                docker pull shahidiqbal008/distance-converter:${env.BUILD_ID}
                                docker stop distance-converter-container || true
                                docker rm distance-converter-container || true
                                docker run -d --name distance-converter-container -p 80:80
shahidiqbal008/distance-converter:${env.BUILD_ID}
                                ""
                        ))
                    ]
                )

            boolean isDeploymentSuccessful = sh(script: 'curl -s -o /dev/null -w "%{http_code}"
http://102.37.216.229:80', returnStdout: true).trim() == '200'

```

```

if (!isDeploymentSuccessful) {
    def previousSuccessfulTag = readFile('previous_successful_tag.txt').trim()
    sshPublisher(
        publishers: [
            sshPublisherDesc(
                configName: "shahidcs",
                transfers: [sshTransfer(
                    execCommand: ""
                        docker pull shahidiqbal008/distance-converter:${previousSuccessfulTag}
                        docker stop distance-converter-container || true
                        docker rm distance-converter-container || true
                        docker run -d --name distance-converter-container -p 80:80
shahidiqbal008/distance-converter:${previousSuccessfulTag}
                    ""
                ))]
            )
        } else {
            writeFile file: 'previous_successful_tag.txt', text: "${env.BUILD_ID}"
        }
    }
}

post {
    failure {
        mail(
            to: 'shahidiqbalofficial.is@gmail.com',
            subject: "Failed Pipeline: ${env.JOB_NAME} [${env.BUILD_NUMBER}]",
            body: ""Something is wrong with the build ${env.BUILD_URL}
            Rolling back to the previous version
            Regards,
            Jenkins
            ""
        )
    }
}

```

## Jenkins Process

Dashboard > Manage Jenkins > System >

### E-mail Notification

SMTP server

smtp.gmail.com

Default user e-mail suffix ?

Next

Email testing.....

Dashboard > Manage Jenkins > System >

☒ Use SMTP Authentication ?

User Name

shahidiqbalofficial.is@gmail.com

Password



Concealed

Change Password

☒ Use SSL ?

☒ Use TLS

SMTP Port ?

465

## Next

Dashboard &gt; Manage Jenkins &gt; System &gt;

## Publish over SSH

Jenkins SSH Key ?

Passphrase ?



Concealed

Change Password

Path to key ?

Key ?

```
-----BEGIN RSA PRIVATE KEY-----
MIIG4wIBAAKCAYEApd7A7JQ0g4LiN+dYf6O0OAWMZ7Geydtl2Is8FnofMpValP
qLXI4KQKQuqCEyncKIUTCv685THzghHp0qrdLL+zdy9jM9YXNa12k5m9BEMpmtA5
PFyfrhR7HrXm1yetWW940DD4ouQriYKjhZ1R6O4amPNefz6Ltr+/vG5UTfmxknQ9
tohNluBPC3PDVchGQgK+YhyDERnTSqijlmjTiDXhKqsrKL22ez+VN8Wo7gq8EaE
OK+OD4sMeuEbqbeJ0b7ZjCj/MtEZOP1ijkFwgPUsN7REu2XUAAg125qM9iEjh4zN
0qaHYSqVvoOg/Q20mfeH0Xz+Ey63eVGv8l3lCXCGgep0tEC7nLbKYHuFzYeFBPcZ
```

Save

Apply

## Next

Dashboard &gt; Manage Jenkins &gt; System &gt;

## SSH Server

Name ?

shahidcs

Hostname ?

tocs.southafricanorth.cloudapp.azure.com

Username ?

azureuser

Remote Directory ?

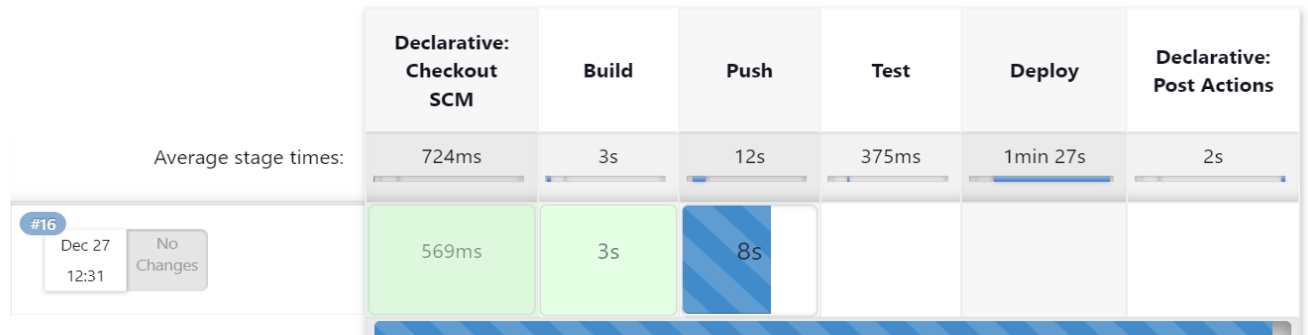
☐ Avoid sending files that have not changed ?

Save

Apply

## Building process

### Stage View



## Console output

```

Started by user shahid iqbal
Obtained Jenkinsfile from git https://github.com/shahidiqbalofficial/TocsTerminal
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Terminal
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Terminal/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/shahidiqbalofficial/TocsTerminal # timeout=10
Fetching upstream changes from https://github.com/shahidiqbalofficial/TocsTerminal
> git --version # timeout=10
> git --version # 'git version 2.25.1'
> git fetch --tags --force --progress -- https://github.com/shahidiqbalofficial/TocsTerminal
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 8e35f4b6dbe90484a5782fedc09c0a23f8e4aad4 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10

```




```

> git checkout -f 8e35f4b6dbe90484a5782fedc09c0a23f8e4aad4 # timeout=10
Commit message: "Update Jenkinsfile"
> git rev-list --no-walk 8e35f4b6dbe90484a5782fedc09c0a23f8e4aad4 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] script
[Pipeline] {
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker build -t shahidiqbal008/distance-converter:16 .
#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 94B done
#1 DONE 0.0s

```



.....And.....

```

#2 [internal] load .dockerignore
#2 transferring context: 2B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/library/nginx:alpine
#3 DONE 2.0s

#4 [1/2] FROM
docker.io/library/nginx:alpine@sha256:a59278fd22a9d411121e190b8cec8aa57b306aa3332459197777583beb728f59
#4 DONE 0.0s

#5 [internal] load build context
#5 transferring context: 6.01kB 0.0s done
#5 DONE 0.0s


#6 [2/2] COPY . /usr/share/nginx/html
#6 CACHED

#7 exporting to image
#7 exporting layers done
#7 writing image sha256:3ca22ed56508fdd704808e5574f78403af61b067753361de047541d5ff245188 done
#7 naming to docker.io/shahidiqbal008/distance-converter:16 0.0s done
#7 DONE 0.0s

```



```
$ docker login -u shahidiqbal008 -p ***** https://registry.hub.docker.com
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /var/lib/jenkins/workspace/Terminal@tmp/3c2990f3-4921-4e8f-9bc2-19da5acfe06c/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
```




Login Succeeded

```
[Pipeline] {
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker tag shahidiqbal008/distance-converter:16 registry.hub.docker.com/shahidiqbal008/distance-converter:16
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker push registry.hub.docker.com/shahidiqbal008/distance-converter:16
The push refers to repository [registry.hub.docker.com/shahidiqbal008/distance-converter]
8b5df497b8ce: Preparing
```



```
[Pipeline] sh
+ ls -l index.html
-rw-r--r-- 1 jenkins jenkins 254 Dec 26 21:13 index.html
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy)
```



# Hello, Docker World!

## Stay Happy!

Name: shahid iqbal

Age: 21

This is a simple web application containerized with Docker. Please provide your details:

Name:

Shahid Iqbal

Registration Number:

sp20-bcs-008

Submit

Created by shahid iqbal