**Created by: Tushendra Kumar Mishra**

**SIMPLILEARN CAPSTONE PROJECT: Hangout Point**

**TASK TO DO:**

Set up the Jenkins server in master or slave architecture

Use the Jenkins plugins to perform the computation part on the Docker containers

Create Jenkins pipeline script

Use the GIT web hook to schedule the job on check-in or poll SCM

Build an image using the artifacts and deploy them on containers

Remove the container stack after completing the job

**GIT\_HUB URL:**

**APP Code url:** [**https://github.com/Tushendra/gs-spring-boot-docker**](https://github.com/Tushendra/gs-spring-boot-docker)

**Project Url:** [**https://github.com/Tushendra/simplilearn-capstone-1/tree/master**](https://github.com/Tushendra/simplilearn-capstone-1/tree/master)

**DOCKER\_HUB URL:**

[**https://hub.docker.com/repository/registry-1.docker.io/tushendra/tushrepo1/**](https://hub.docker.com/repository/registry-1.docker.io/tushendra/tushrepo1/)

**PLUGINS\_INSTALLED IN JENKINS:**

**Ansible**

**Docker**

**TOOLS USED:**

1. **Jenkins**
2. **Git**
3. **Git\_Hub**
4. **Azure Cloud**
5. **Terraform**
6. **Docker**
7. **Ansible**

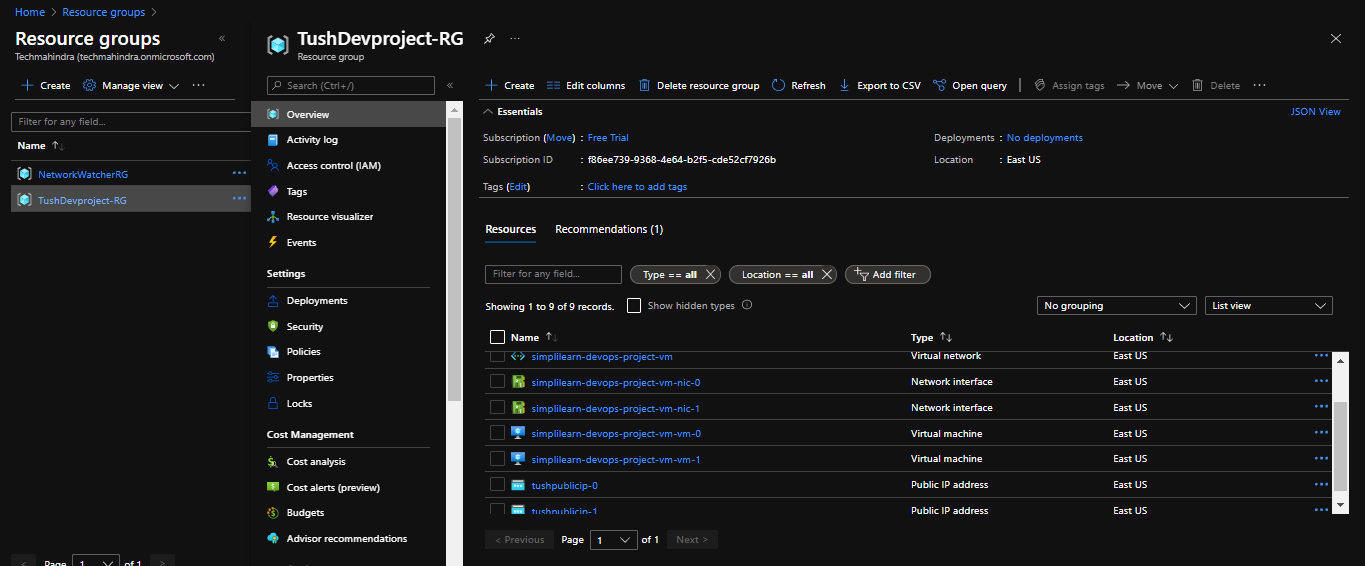
**HOW IMPLEMENTED:**

**Steps:**

**STEP\_1:**

**Created 2 VMs on Azure**

**In one Resource Group**

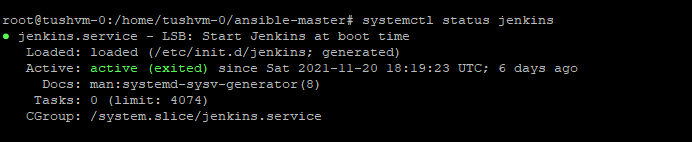
****

**STEP\_2:**

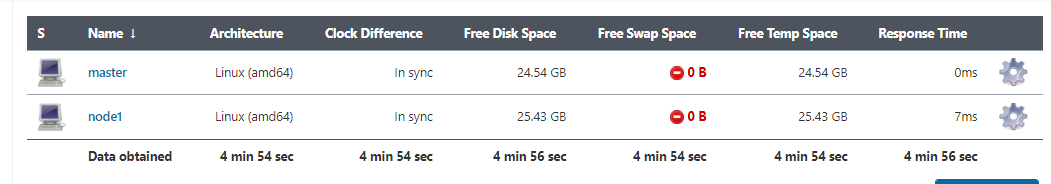
**VM\_0:- Simplilearn-devops-project-vm-vm-0**

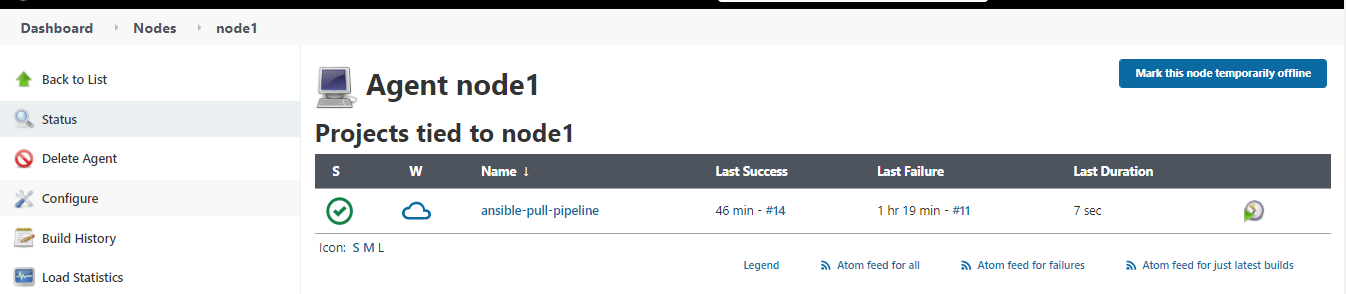
**VM\_1:- Simplilearn-devops-project-vm-vm-1**

**VM\_0 used as master for Jenkins**

****

**VM\_1 used as slave for ansible**

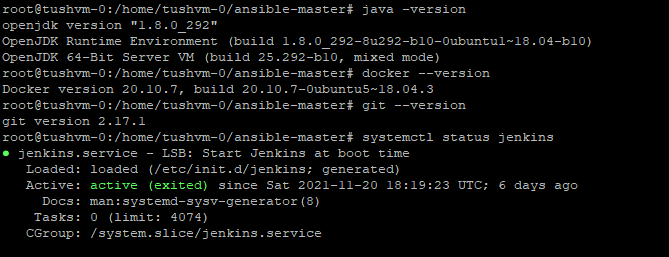
****

****

**STEP\_3:**

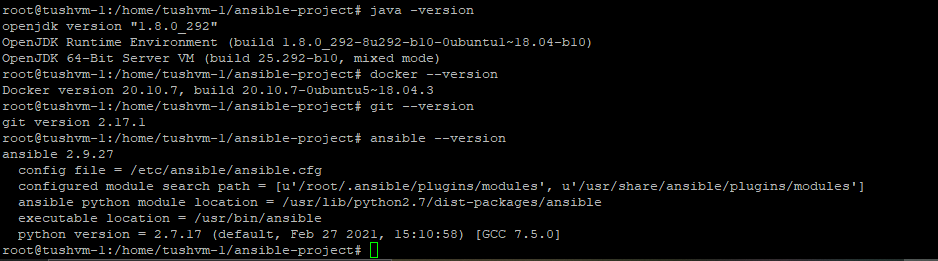
**Installed application on vm\_0**

1. **Docker**
2. **Jenkins**
3. **Git**
4. **Jdk**

****

**Installed application on vm\_1**

1. **Docker**
2. **Git**
3. **Jdk**
4. **Ansible**
5. **Python**

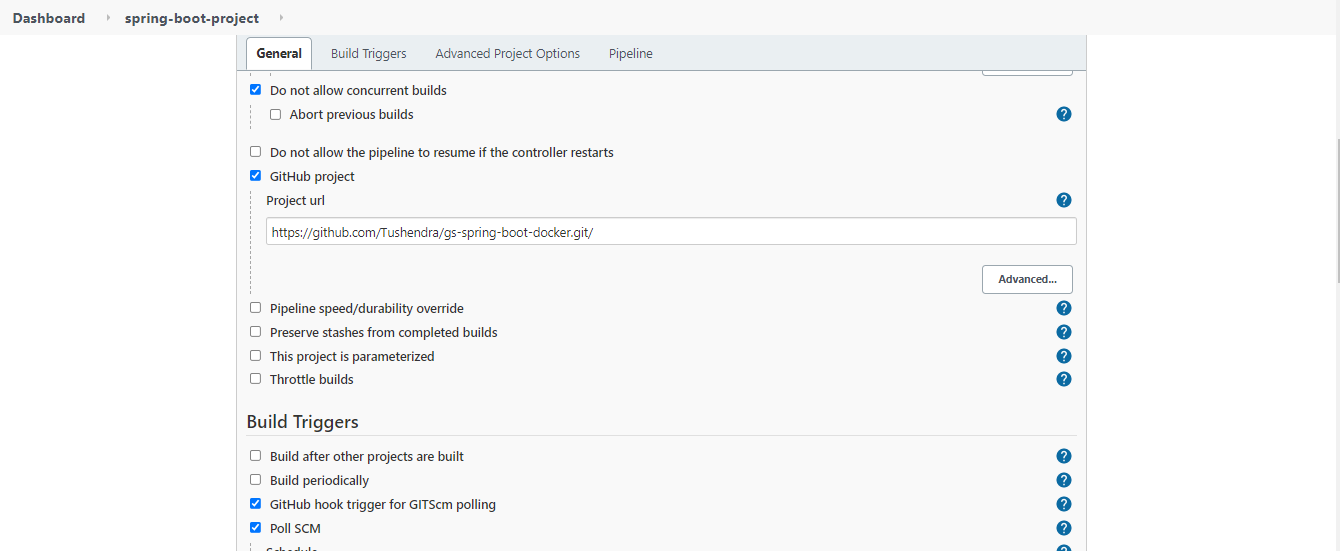
****

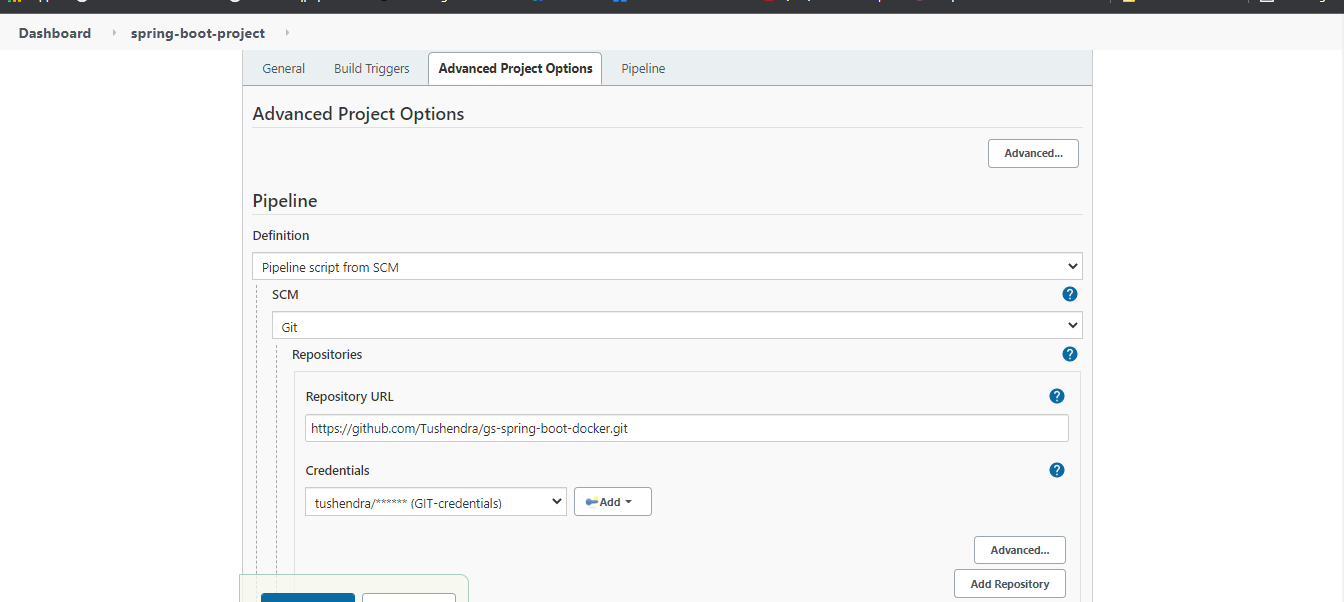
**Step\_4:**

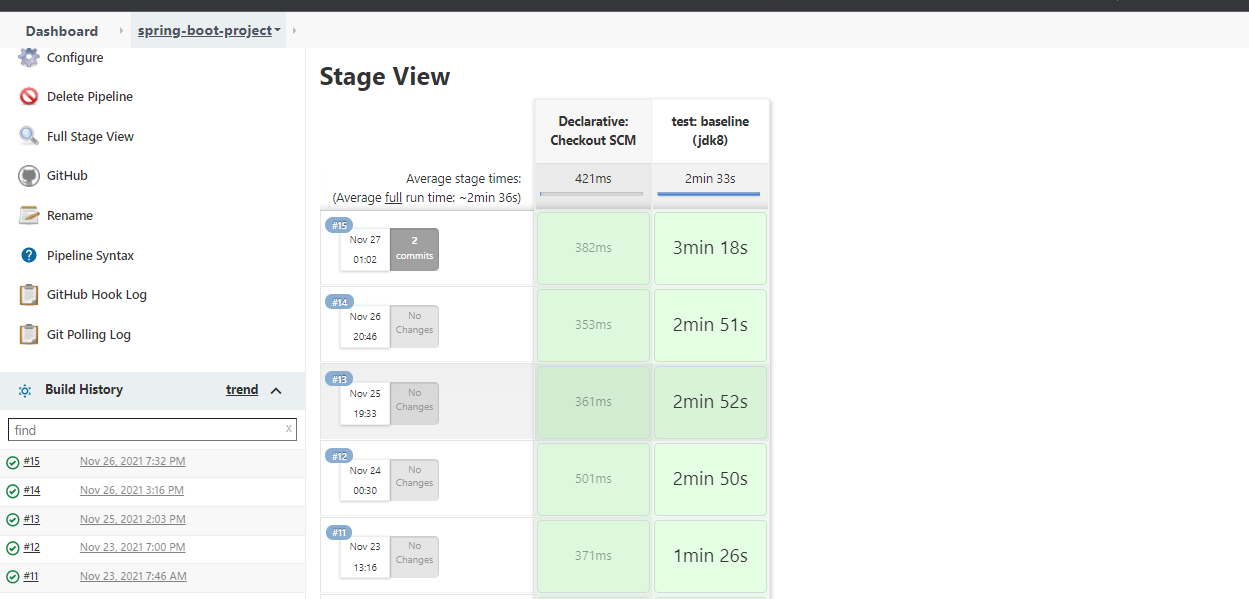
**Created First pipeline for building an image**

**Polling from Git\_Hub**

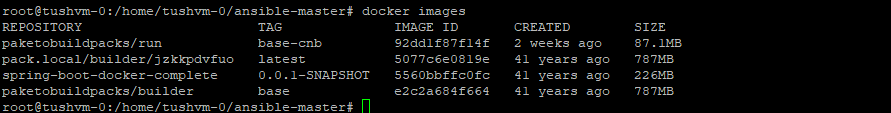
**Running on Master**

****

****

****

**Images Created/Build:**

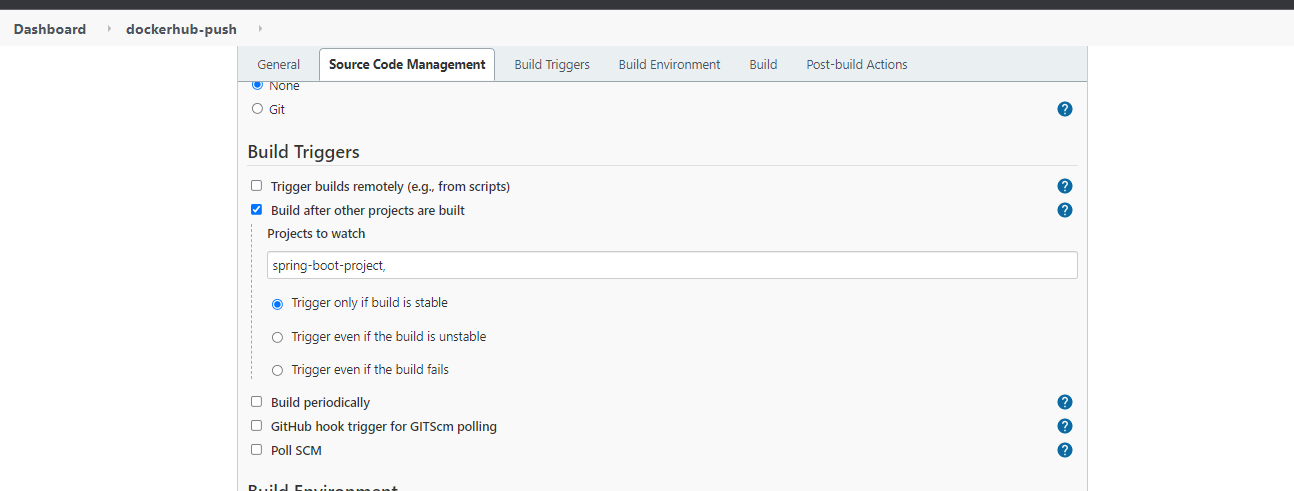
****

**Main image with all the dependencies is :**

**Complete:0.0.1-SNAPSHOT**

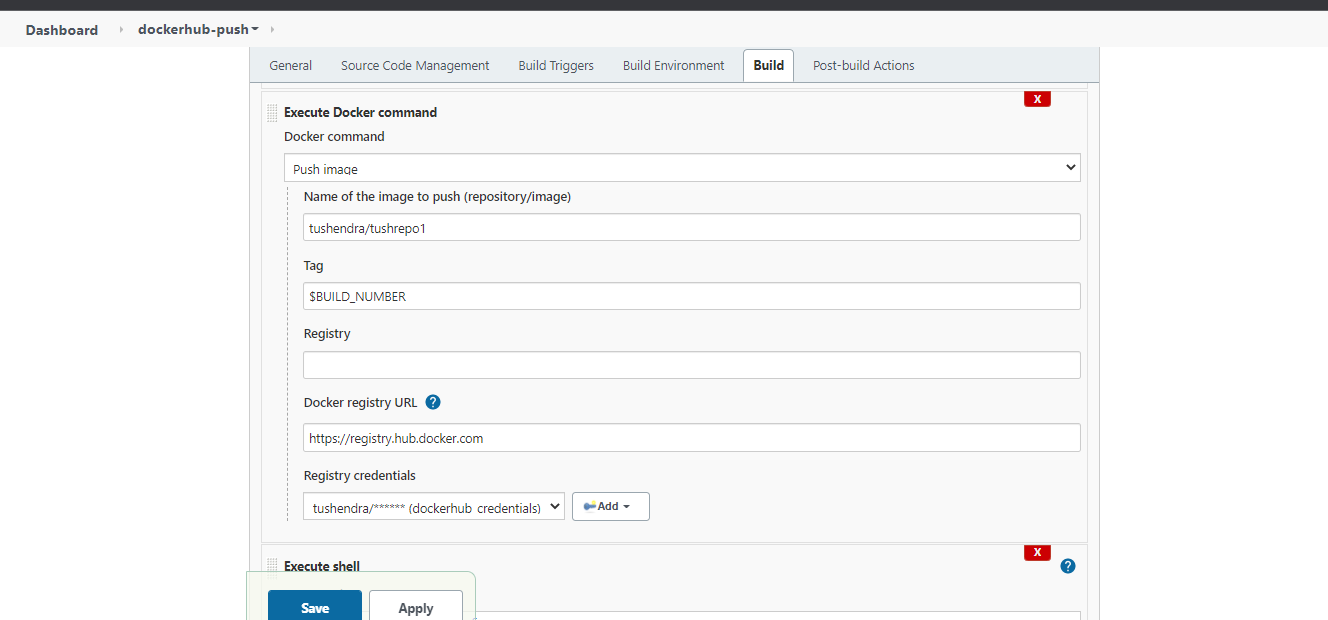
**STEP\_5:**

**Created another pipeline for pushing the image to Docker\_Hub**

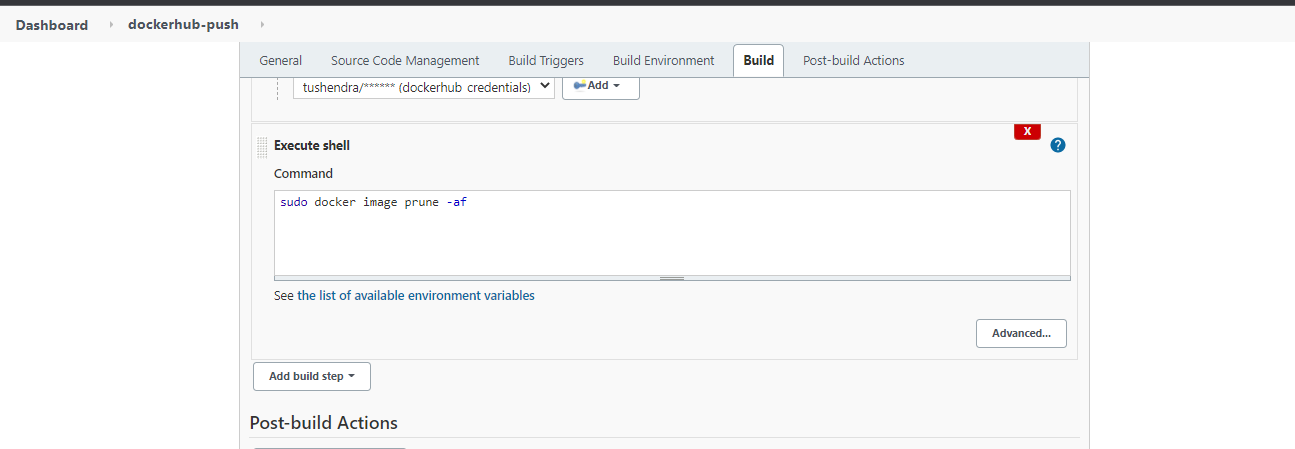
****

****

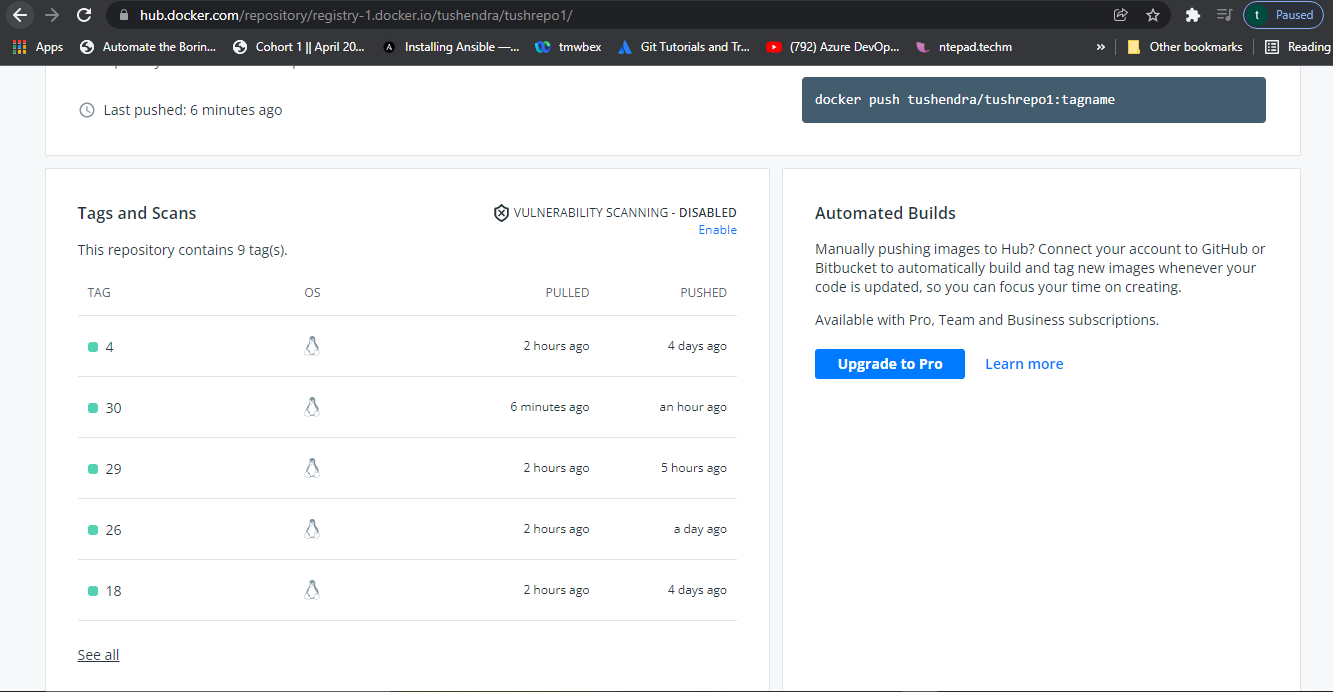
**Configuration to push to docker-hub**

****

**Now to clear all the images stack from master :**

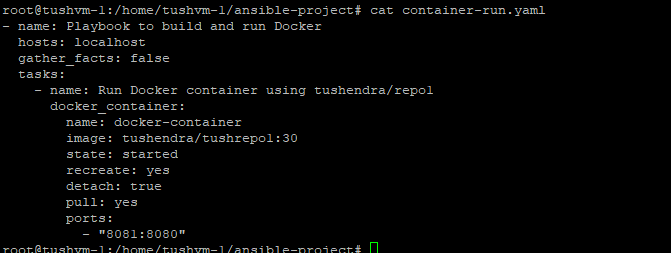
****

**Docker\_Hub Repo:**

****

**STEP\_7:**

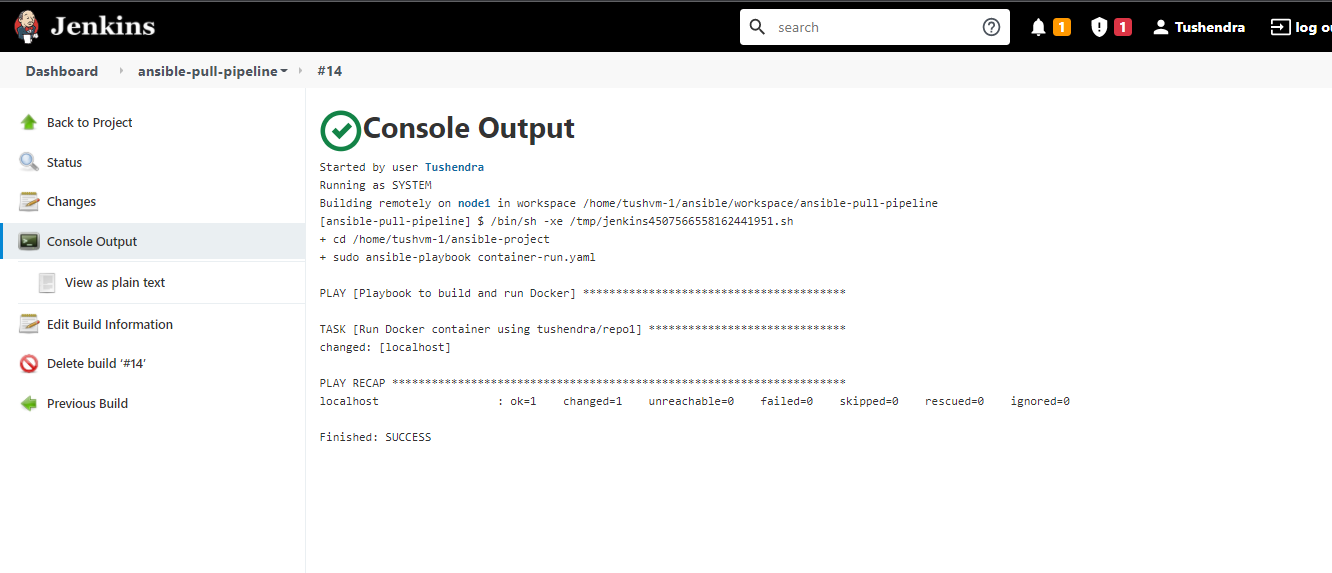
**Created Ansible playbook on node server : VM\_1**

****

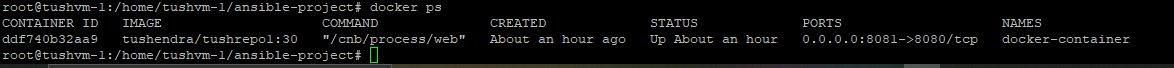
**Created pipeline to run the playbook from Jenkins:**

**Playbook will pull the image from Docker\_Hub and run the container**

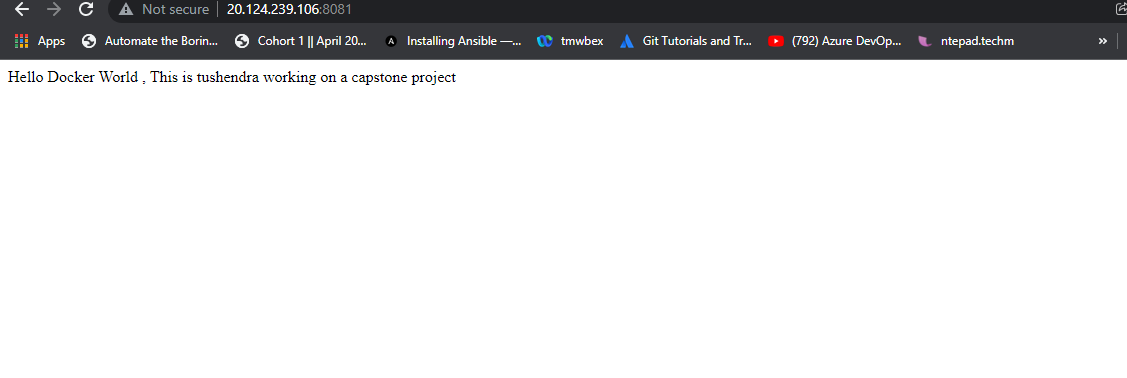
**Out of that image.**

****

**Container is running :**

****

**App is also visible on port 8081:**

****

**Execution Successful here…**

**Terraform code :**

**terraform {**

**required\_version = ">= 1.0.0"**

**required\_providers {**

**azurerm = {**

**source = "hashicorp/azurerm"**

**version = ">= 2.0" # Optional but recommended in production**

**}**

**}**

**}**

**# Configure the Microsoft Azure Provider**

**provider "azurerm" {**

**features {}**

**}**

**# Create Resource Group**

**resource "azurerm\_resource\_group" "TushRG" {**

**location = "eastus"**

**name = "TushDevproject-RG"**

**}**

**# Create Virtual Machine**

**resource "azurerm\_virtual\_network" "main" {**

**name = var.vm-name**

**address\_space = ["10.0.0.0/16"]**

**location = azurerm\_resource\_group.TushRG.location**

**resource\_group\_name = azurerm\_resource\_group.TushRG.name**

**}**

**resource "azurerm\_subnet" "internal" {**

**name = "internal"**

**resource\_group\_name = azurerm\_resource\_group.TushRG.name**

**virtual\_network\_name = azurerm\_virtual\_network.main.name**

**address\_prefixes = ["10.0.1.0/24"]**

**}**

**resource "azurerm\_network\_interface" "main" {**

**count = 2**

**name = "${var.vm-name}-nic-${count.index}"**

**location = azurerm\_resource\_group.TushRG.location**

**resource\_group\_name = azurerm\_resource\_group.TushRG.name**

**ip\_configuration {**

**name = "testconfiguration1"**

**subnet\_id = azurerm\_subnet.internal.id**

**private\_ip\_address\_allocation = "Dynamic"**

**public\_ip\_address\_id = element(azurerm\_public\_ip.public-ip[\*].id, count.index)**

**}**

**}**

**resource "azurerm\_public\_ip" "public-ip" {**

**count = 2**

**name = "${var.public-ip}-${count.index}"**

**resource\_group\_name = azurerm\_resource\_group.TushRG.name**

**location = azurerm\_resource\_group.TushRG.location**

**allocation\_method = "Static"**

**}**

**resource "azurerm\_virtual\_machine" "main" {**

**count = 2**

**name = "${var.vm-name}-vm-${count.index}"**

**location = azurerm\_resource\_group.TushRG.location**

**resource\_group\_name = azurerm\_resource\_group.TushRG.name**

**network\_interface\_ids = [azurerm\_network\_interface.main[count.index].id]**

**vm\_size = "Standard\_DS1\_v2"**

**# Uncomment this line to delete the OS disk automatically when deleting the VM**

**# delete\_os\_disk\_on\_termination = true**

**# Uncomment this line to delete the data disks automatically when deleting the VM**

**# delete\_data\_disks\_on\_termination = true**

**storage\_image\_reference {**

**publisher = "Canonical"**

**offer = "UbuntuServer"**

**sku = "18.04-LTS"**

**version = "latest"**

**}**

**storage\_os\_disk {**

**name = "os-disk${count.index}"**

**caching = "ReadWrite"**

**create\_option = "FromImage"**

**managed\_disk\_type = "Standard\_LRS"**

**}**

**os\_profile {**

**computer\_name = "${var.computername}-${count.index}"**

**admin\_username = "${var.computername}-${count.index}"**

**admin\_password = var.vmpassword**

**}**

**os\_profile\_linux\_config {**

**disable\_password\_authentication = false**

**}**

**tags = {**

**environment = "Dev"**

**}**

**}**