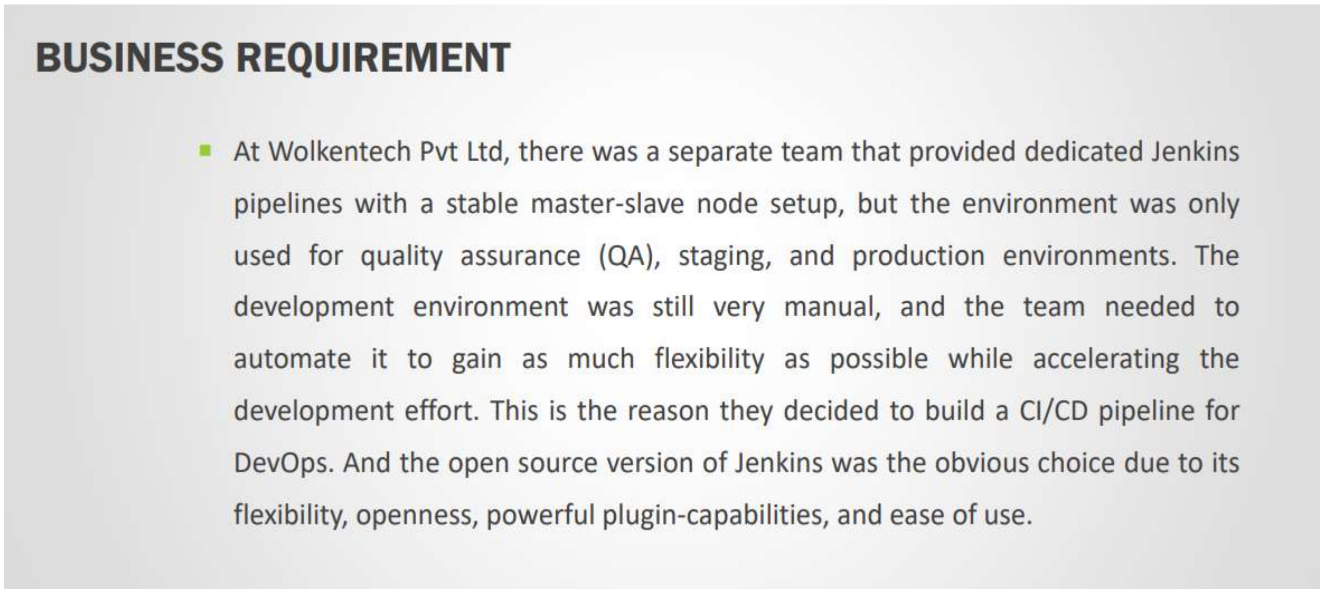
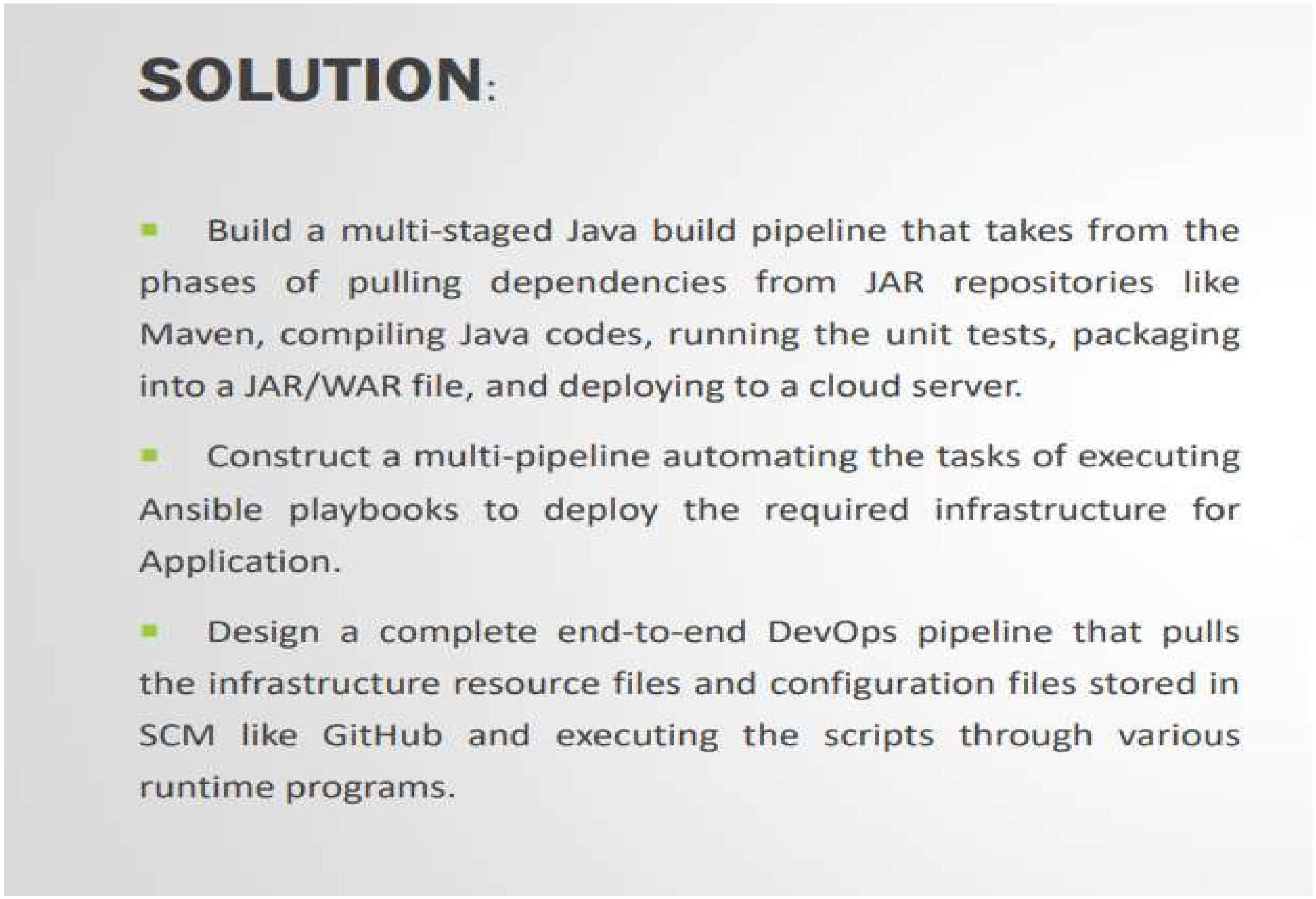
############ PROJECT1 ###############



BUSINESS REQUIREMENT

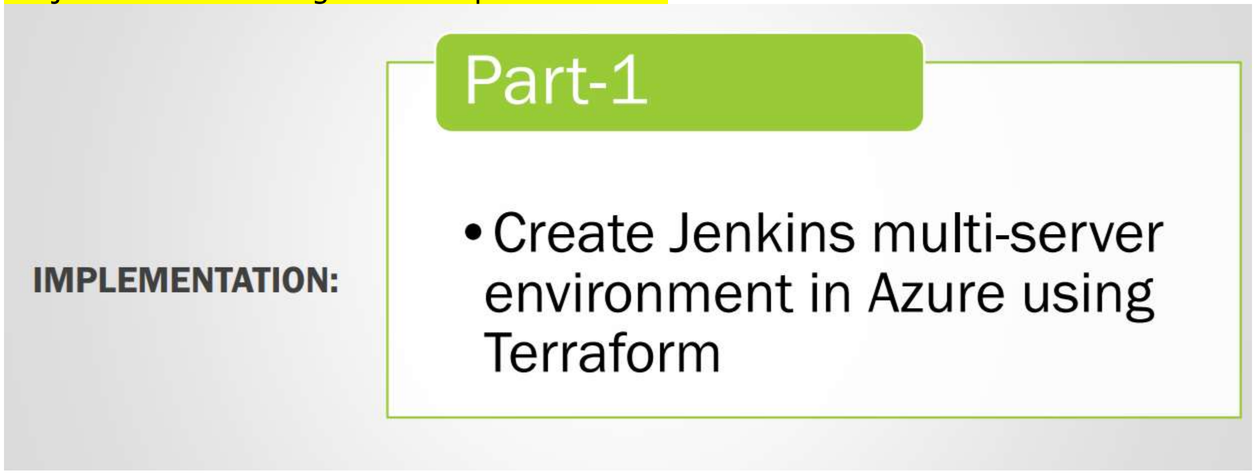
- At Wolkentech Pvt Ltd, there was a separate team that provided dedicated Jenkins pipelines with a stable master-slave node setup, but the environment was only used for quality assurance (QA), staging, and production environments. The development environment was still very manual, and the team needed to automate it to gain as much flexibility as possible while accelerating the development effort. This is the reason they decided to build a CI/CD pipeline for DevOps. And the open source version of Jenkins was the obvious choice due to its flexibility, openness, powerful plugin-capabilities, and ease of use.



SOLUTION:

- Build a multi-staged Java build pipeline that takes from the phases of pulling dependencies from JAR repositories like Maven, compiling Java codes, running the unit tests, packaging into a JAR/WAR file, and deploying to a cloud server.
- Construct a multi-pipeline automating the tasks of executing Ansible playbooks to deploy the required infrastructure for Application.
- Design a complete end-to-end DevOps pipeline that pulls the infrastructure resource files and configuration files stored in SCM like GitHub and executing the scripts through various runtime programs.

Project1 – Part1 – High level implementation

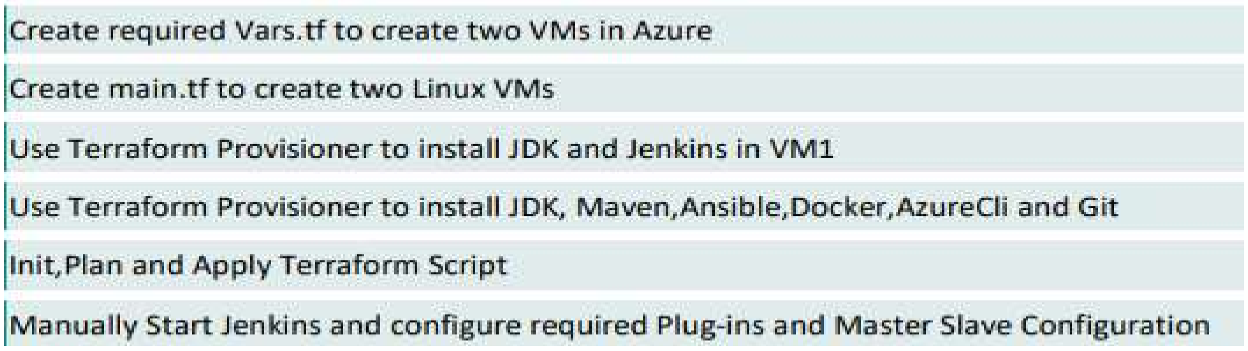


IMPLEMENTATION:

Part-1

- Create Jenkins multi-server environment in Azure using Terraform

Below are the detailed steps needed to perform



Create required Vars.tf to create two VMs in Azure

Create main.tf to create two Linux VMs

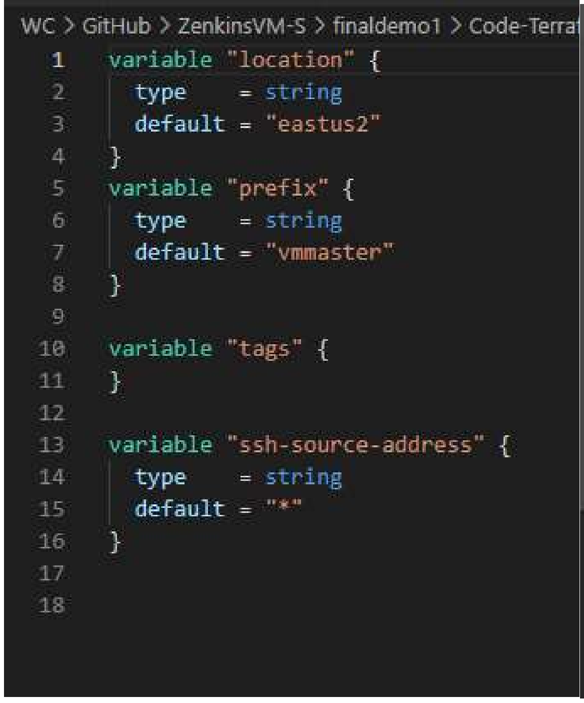
Use Terraform Provisioner to install JDK and Jenkins in VM1

Use Terraform Provisioner to install JDK, Maven,Ansible,Docker,AzureCli and Git

Init,Plan and Apply Terraform Script

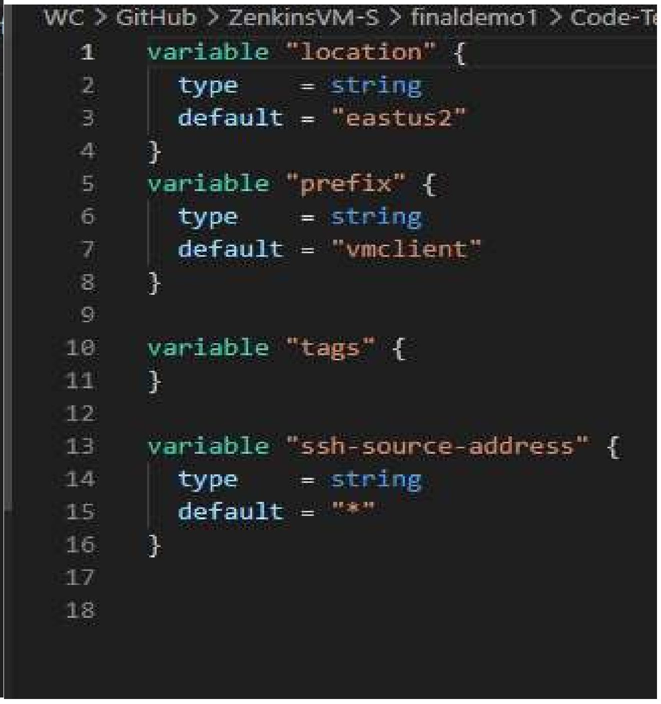
Manually Start Jenkins and configure required Plug-ins and Master Slave Configuration

Created required Vars.tf to create two VMs in Azure



WC > GitHub > ZenkinsVM-S > finaldemo1 > Code-Terra

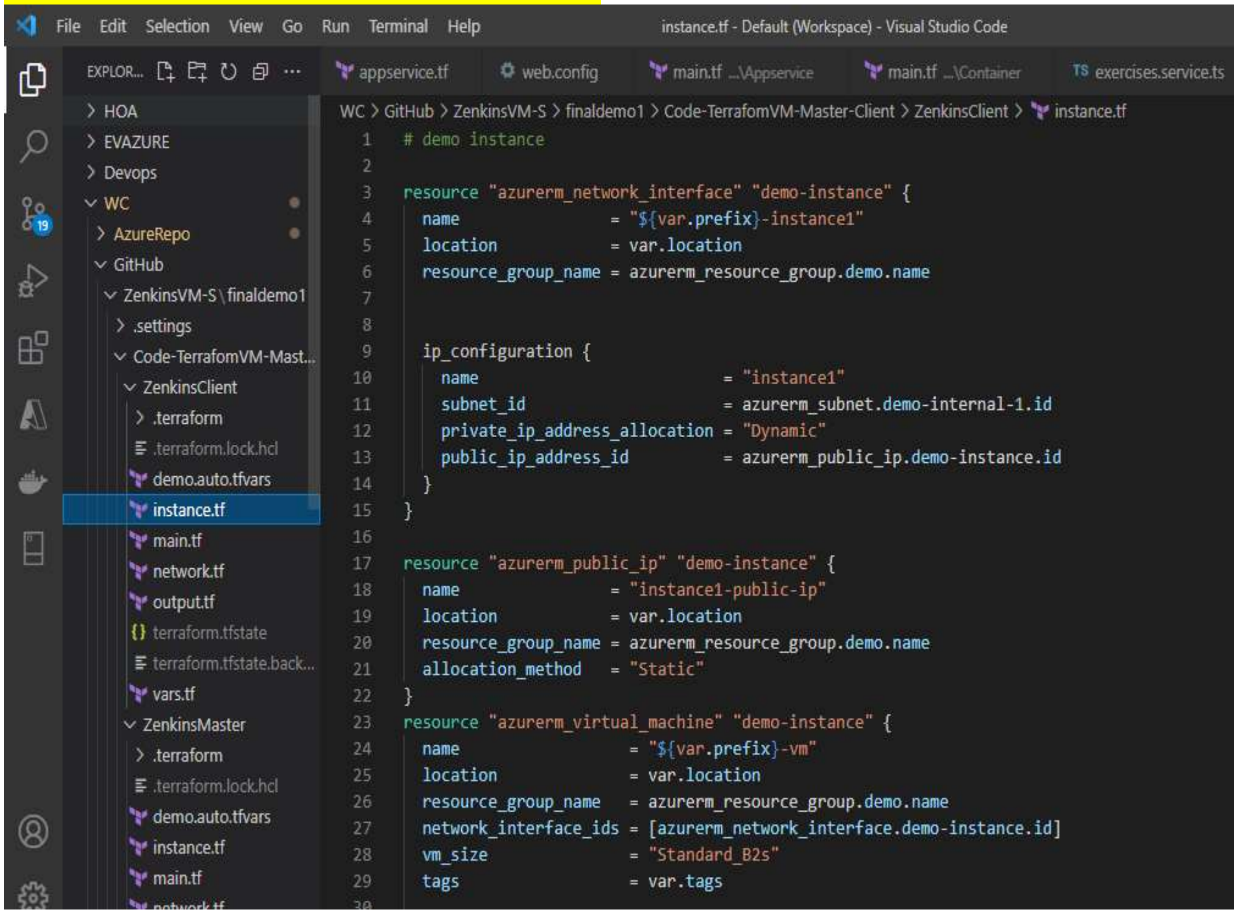
```
variable "location" {
  type    = string
  default = "eastus2"
}
variable "prefix" {
  type    = string
  default = "vmmaster"
}

variable "tags" {
}

variable "ssh-source-address" {
  type    = string
  default = "*"
}


```



WC > GitHub > ZenkinsVM-S > finaldemo1 > Code-Te

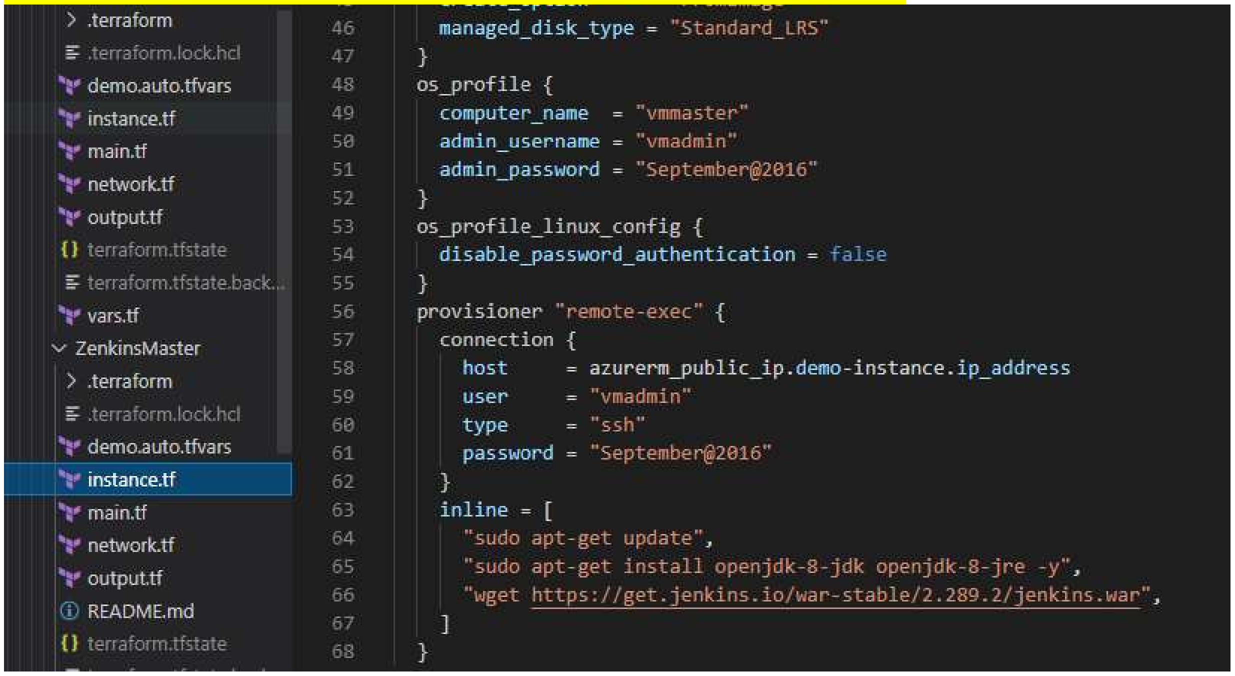
```
variable "location" {
  type    = string
  default = "eastus2"
}
variable "prefix" {
  type    = string
  default = "vmclient"
}

variable "tags" {
}

variable "ssh-source-address" {
  type    = string
  default = "*"
}


```

Created main.tf to create two Linux VMs



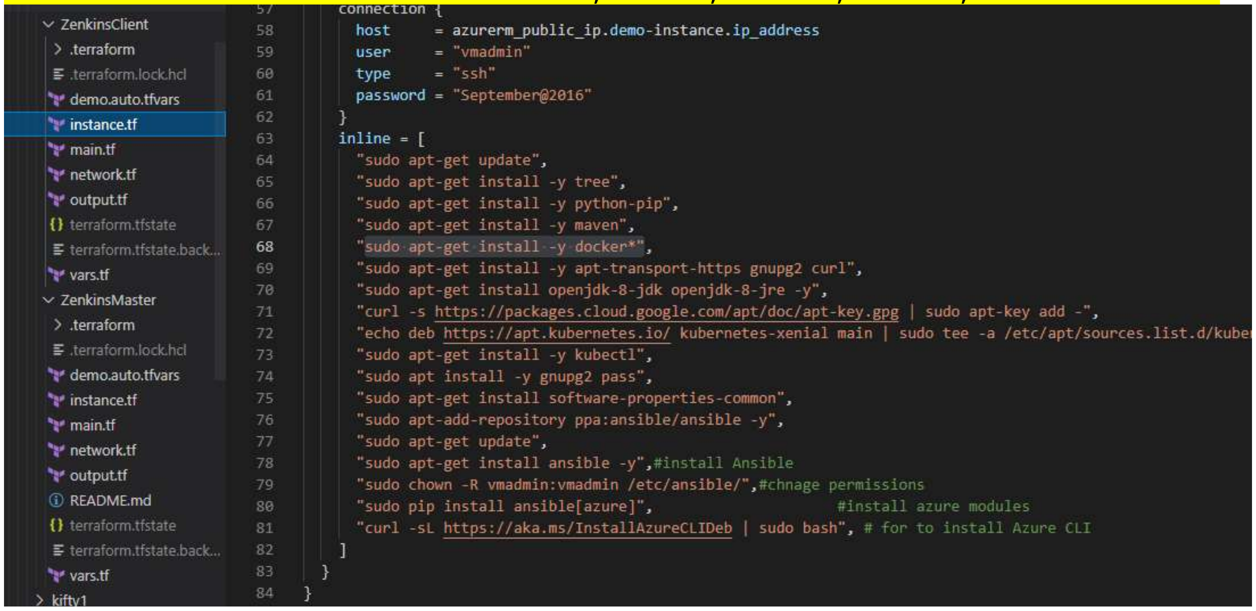
```
# demo instance

resource "azurerm_network_interface" "demo-instance" {
  name                = "${var.prefix}-instance1"
  location            = var.location
  resource_group_name = azurerm_resource_group.demo.name


  ip_configuration {
    name                          = "instance1"
    subnet_id                     = azurerm_subnet.demo-internal-1.id
    private_ip_address_allocation = "Dynamic"
    public_ip_address_id          = azurerm_public_ip.demo-instance.id
  }
}

resource "azurerm_public_ip" "demo-instance" {
  name                = "instance1-public-ip"
  location            = var.location
  resource_group_name = azurerm_resource_group.demo.name
  allocation_method   = "Static"
}
resource "azurerm_virtual_machine" "demo-instance" {
  name                  = "${var.prefix}-vm"
  location              = var.location
  resource_group_name   = azurerm_resource_group.demo.name
  network_interface_ids = [azurerm_network_interface.demo-instance.id]
  vm_size               = "Standard_B2s"
  tags                  = var.tags
```

Used Terraform Provisioner to install JDK and Jenkins in VM1



```
    managed_disk_type = "Standard_LRS"
  }
  os_profile {
    computer_name  = "vmmaster"
    admin_username = "vmadmin"
    admin_password = "September@2016"
  }
  os_profile_linux_config {
    disable_password_authentication = false
  }
  provisioner "remote-exec" {
    connection {
      host     = azurerm_public_ip.demo-instance.ip_address
      user     = "vmadmin"
      type     = "ssh"
      password = "September@2016"
    }
    inline = [
      "sudo apt-get update",
      "sudo apt-get install openjdk-8-jdk openjdk-8-jre -y",
      "wget https://get.jenkins.io/war-stable/2.289.2/jenkins.war",
    ]
  }
```

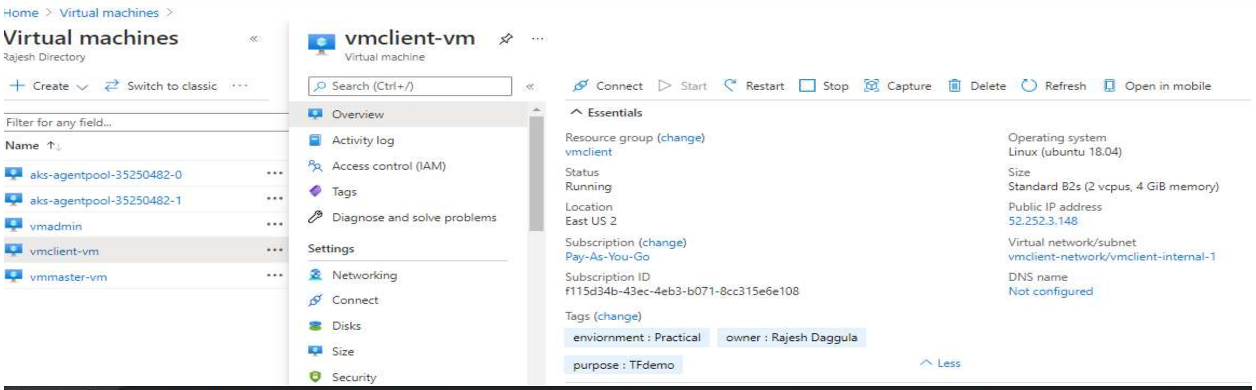
Use Terraform Provisioner to install JDK, Maven, Ansible, Docker, AzureCli and Git



```
    connection {
      host     = azurerm_public_ip.demo-instance.ip_address
      user     = "vmadmin"
      type     = "ssh"
      password = "September@2016"
    }
    inline = [
      "sudo apt-get update",
      "sudo apt-get install -y tree",
      "sudo apt-get install -y python-pip",
      "sudo apt-get install -y maven",
      "sudo apt-get install -y docker*",
      "sudo apt-get install -y apt-transport-https gnupg2 curl",
      "sudo apt-get install openjdk-8-jdk openjdk-8-jre -y",
      "curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -",
      "echo deb https://apt.kubernetes.io/ kubernetes-xenial main | sudo tee -a /etc/apt/sources.list.d/kube
      "sudo apt-get install -y kubectl",
      "sudo apt install -y gnupg2 pass",
      "sudo apt-get install software-properties-common",
      "sudo apt-add-repository ppa:ansible/ansible -y",
      "sudo apt-get update",
      "sudo apt-get install ansible -y",#install Ansible
      "sudo chown -R vmadmin:vmadmin /etc/ansible/",#chnage permissions
      "sudo pip install ansible[azure]",            #install azure modules
      "curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash", # for to install Azure CLI
    ]
  }
}
```

Init, Plan and Apply Terraform Script to create VM'S in Azure

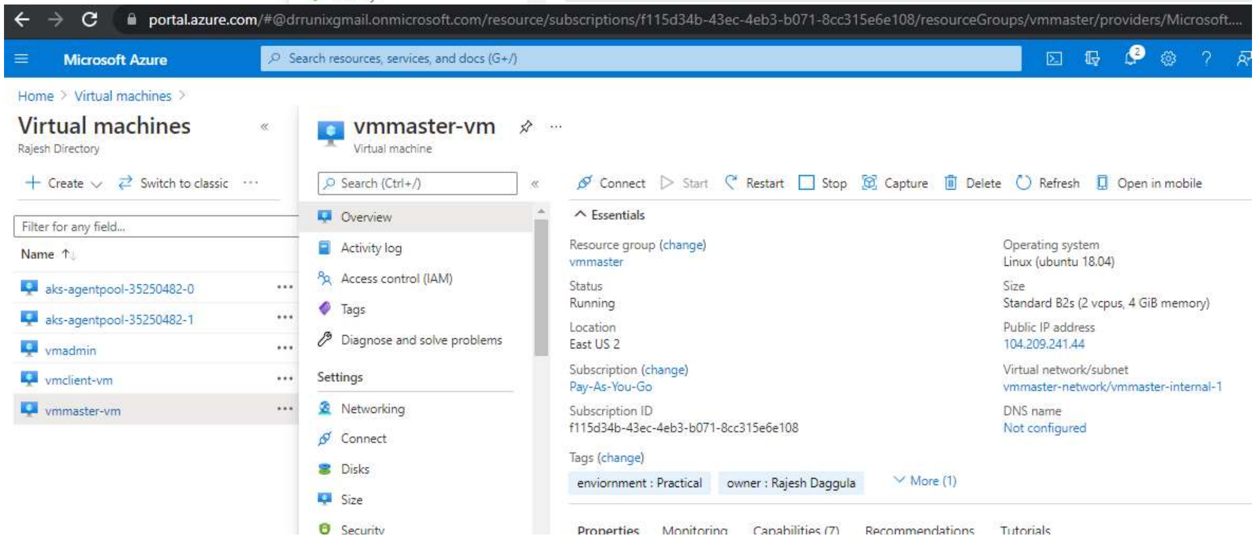
I already created VMS using terraform before. currently both VM'S are up and running in Azure.



aks-agentpool-35250482-0
aks-agentpool-35250482-1
vmadmin
vmclient-vm
vmmaster-vm

Resource group (change)	vmclient
Status	Running
Location	East US 2
Subscription (change)	Pay-As-You-Go
Subscription ID	f115d34b-43ec-4eb3-b071-8cc315e6e108
Operating system	Linux (ubuntu 18.04)
Size	Standard B2s (2 vcpus, 4 GiB memory)
Public IP address	52.252.3.148
Virtual network/subnet	vmclient-network/vmclient-internal-1
DNS name	Not configured

Tags (change)
envionment : Practical
owner : Rajesh Daggula
purpose : TFdemo



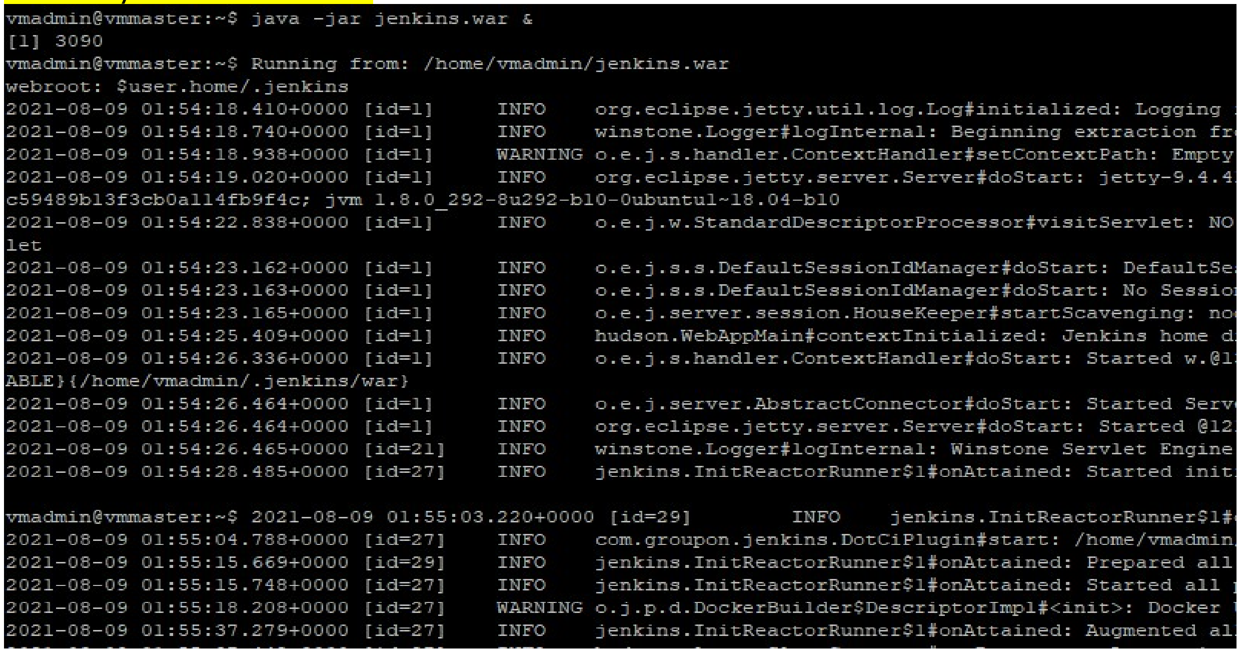
aks-agentpool-35250482-0
aks-agentpool-35250482-1
vmadmin
vmclient-vm
vmmaster-vm

Resource group (change)	vmmaster
Status	Running
Location	East US 2
Subscription (change)	Pay-As-You-Go
Subscription ID	f115d34b-43ec-4eb3-b071-8cc315e6e108
Operating system	Linux (ubuntu 18.04)
Size	Standard B2s (2 vcpus, 4 GiB memory)
Public IP address	104.209.241.44
Virtual network/subnet	vmmaster-network/vmmaster-internal-1
DNS name	Not configured

Tags (change)
envionment : Practical
owner : Rajesh Daggula

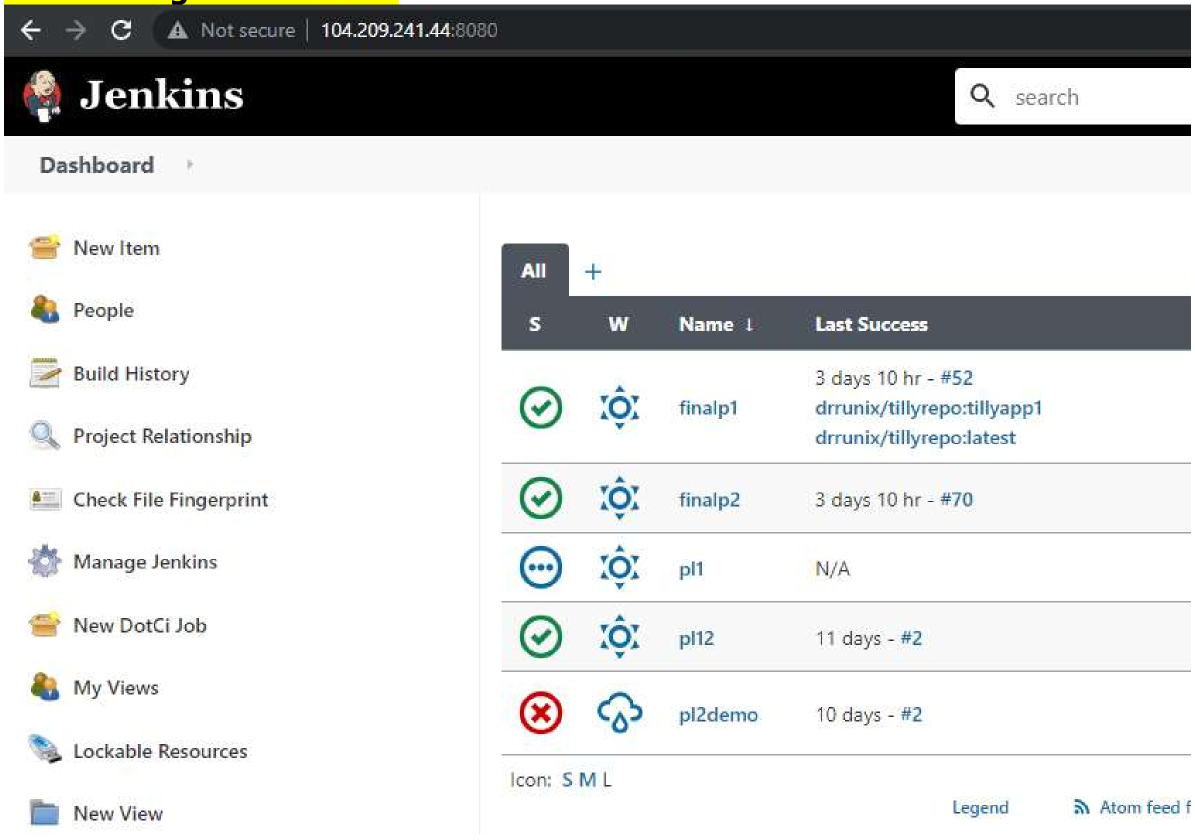
Manually Started Jenkins and configured required Plug-ins and Master Slave Configuration

Manually Started Jenkins



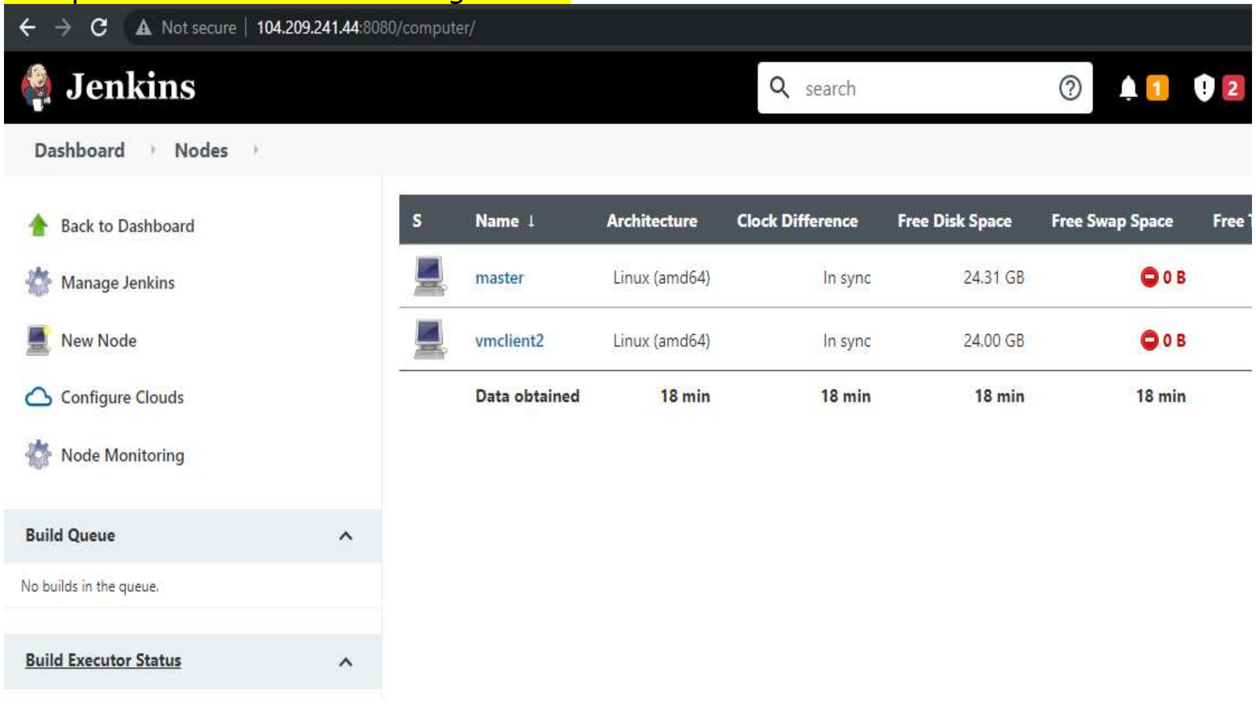
```
vmadmin@vmmaster:~$ java -jar jenkins.war &
[1] 3090
vmadmin@vmmaster:~$ Running from: /home/vmadmin/jenkins.war
webroot: $user.home/.jenkins
2021-08-09 01:54:18.410+0000 [id=1]     INFO    org.eclipse.jetty.util.log.Log#initialized: Logging
2021-08-09 01:54:18.740+0000 [id=1]     INFO    winstone.Logger#logInternal: Beginning extraction fr
2021-08-09 01:54:18.938+0000 [id=1]     WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty
2021-08-09 01:54:19.020+0000 [id=1]     INFO    org.eclipse.jetty.server.Server#doStart: jetty-9.4.4
c59489b13f3cb0a114fb9f4c; jvm 1.8.0_292-8u292-b10-0ubuntu1~18.04-b10
2021-08-09 01:54:22.838+0000 [id=1]     INFO    o.e.j.w.StandardDescriptorProcessor#visitServlet: NO
let
2021-08-09 01:54:23.162+0000 [id=1]     INFO    o.e.j.s.s.DefaultSessionIdManager#doStart: DefaultSe
2021-08-09 01:54:23.163+0000 [id=1]     INFO    o.e.j.s.s.DefaultSessionIdManager#doStart: No Sessio
2021-08-09 01:54:23.165+0000 [id=1]     INFO    o.e.j.server.session.HouseKeeper#startScavenging: no
2021-08-09 01:54:25.409+0000 [id=1]     INFO    hudson.WebAppMain#contextInitialized: Jenkins home d
2021-08-09 01:54:26.336+0000 [id=1]     INFO    o.e.j.s.handler.ContextHandler#doStart: Started w.@1
ABLE}{/home/vmadmin/.jenkins/war}
2021-08-09 01:54:26.464+0000 [id=1]     INFO    o.e.j.server.AbstractConnector#doStart: Started Serv
2021-08-09 01:54:26.464+0000 [id=1]     INFO    org.eclipse.jetty.server.Server#doStart: Started @12
2021-08-09 01:54:26.465+0000 [id=21]    INFO    winstone.Logger#logInternal: Winstone Servlet Engine
2021-08-09 01:54:28.485+0000 [id=27]    INFO    jenkins.InitReactorRunner$1#onAttained: Started init

vmadmin@vmmaster:~$ 2021-08-09 01:55:03.220+0000 [id=29]        INFO    jenkins.InitReactorRunner$1#
2021-08-09 01:55:04.788+0000 [id=27]    INFO    com.groupon.jenkins.DotCiPlugin#start: /home/vmadmin
2021-08-09 01:55:15.669+0000 [id=29]    INFO    jenkins.InitReactorRunner$1#onAttained: Prepared all
2021-08-09 01:55:15.748+0000 [id=27]    INFO    jenkins.InitReactorRunner$1#onAttained: Started all
2021-08-09 01:55:18.208+0000 [id=27]    WARNING o.j.p.d.DockerBuilder$DescriptorImpl#<init>: Docker
2021-08-09 01:55:37.279+0000 [id=27]    INFO    jenkins.InitReactorRunner$1#onAttained: Augmented al
```

Able to Login to Jenkins



S	W	Name ↓	Last Success
✓	☀	finalp1	3 days 10 hr - #52 drrunix/tillyrepo:tillyapp1 drrunix/tillyrepo:latest
✓	☀	finalp2	3 days 10 hr - #70
…	☀	pl1	N/A
✓	☀	pl12	11 days - #2
✗	🌧	pl2demo	10 days - #2

Completed Master Slave Configuration



104.209.241.44:8080/computer/

Dashboard > Nodes

- Back to Dashboard
- Manage Jenkins
- New Node
- Configure Clouds
- Node Monitoring

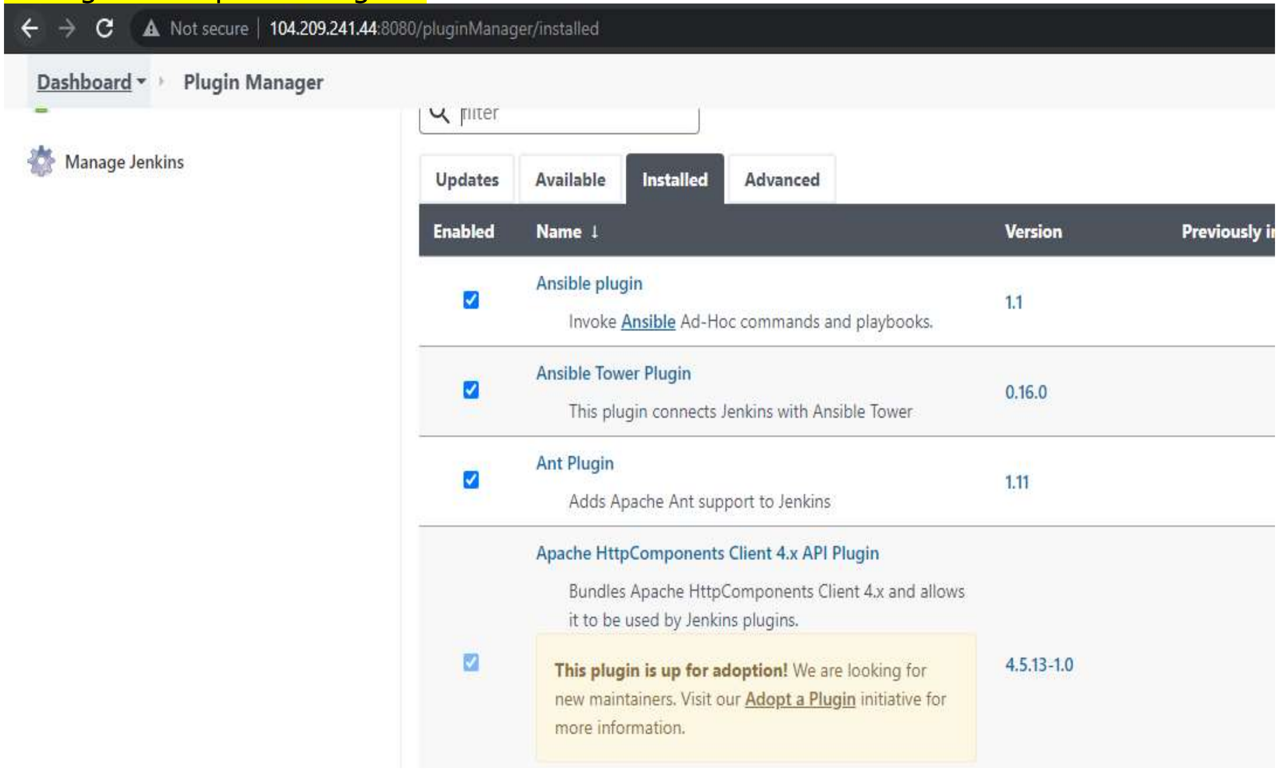
Build Queue

No builds in the queue.

Build Executor Status

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free
	master	Linux (amd64)	In sync	24.31 GB	0 B	
	vmclient2	Linux (amd64)	In sync	24.00 GB	0 B	
	Data obtained	18 min	18 min	18 min	18 min	

Configured required Plug-ins



104.209.241.44:8080/pluginManager/installed

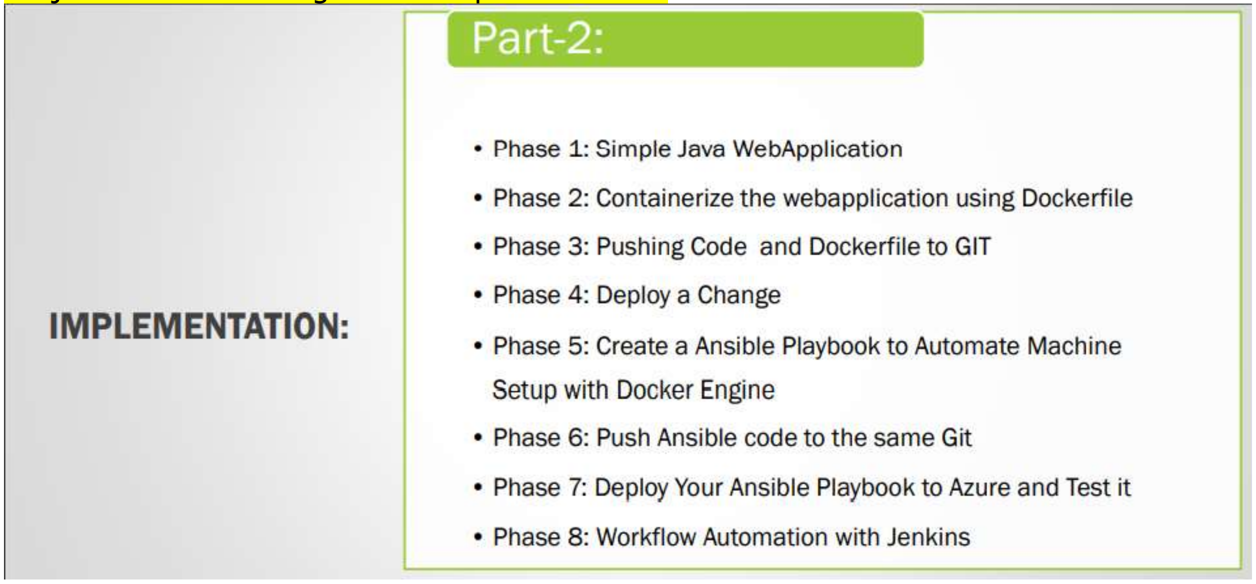
Dashboard > Plugin Manager

Manage Jenkins

Updates | Available | Installed | Advanced

Enabled	Name ↓	Version	Previously i
✓	Ansible plugin Invoke Ansible Ad-Hoc commands and playbooks.	1.1	
✓	Ansible Tower Plugin This plugin connects Jenkins with Ansible Tower	0.16.0	
✓	Ant Plugin Adds Apache Ant support to Jenkins	1.11	
✓	Apache HttpComponents Client 4.x API Plugin Bundles Apache HttpComponents Client 4.x and allows it to be used by Jenkins plugins. **This plugin is up for adoption!** We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.	4.5.13-1.0	

Project1 – Part2 – High level implementation



IMPLEMENTATION:

Part-2:

- Phase 1: Simple Java WebApplication
- Phase 2: Containerize the webapplication using Dockerfile
- Phase 3: Pushing Code and Dockerfile to GIT
- Phase 4: Deploy a Change
- Phase 5: Create a Ansible Playbook to Automate Machine Setup with Docker Engine
- Phase 6: Push Ansible code to the same Git
- Phase 7: Deploy Your Ansible Playbook to Azure and Test it
- Phase 8: Workflow Automation with Jenkins

Below are Detailed steps needed to perform in Part-2

Create Maven Project with Archtype as web application in eclipse

Modify Index.jsp under src/main/webcontent to display a custom message

Gerate Dockerfile under project folder of your app

Modify FORM statement to use tomcat as base image

Create a github repository and copy repo URL

In Eclipse convert the app in to a local repo from Team meanu share Project Option

Commit and Push the code to remote repo

In build server configure Ansible manually

Modify ansible.cfg to use hosts file as inventory

Create a playbook1 to create a vm in azure

Update the playbook2 to install Docker engine on the VM

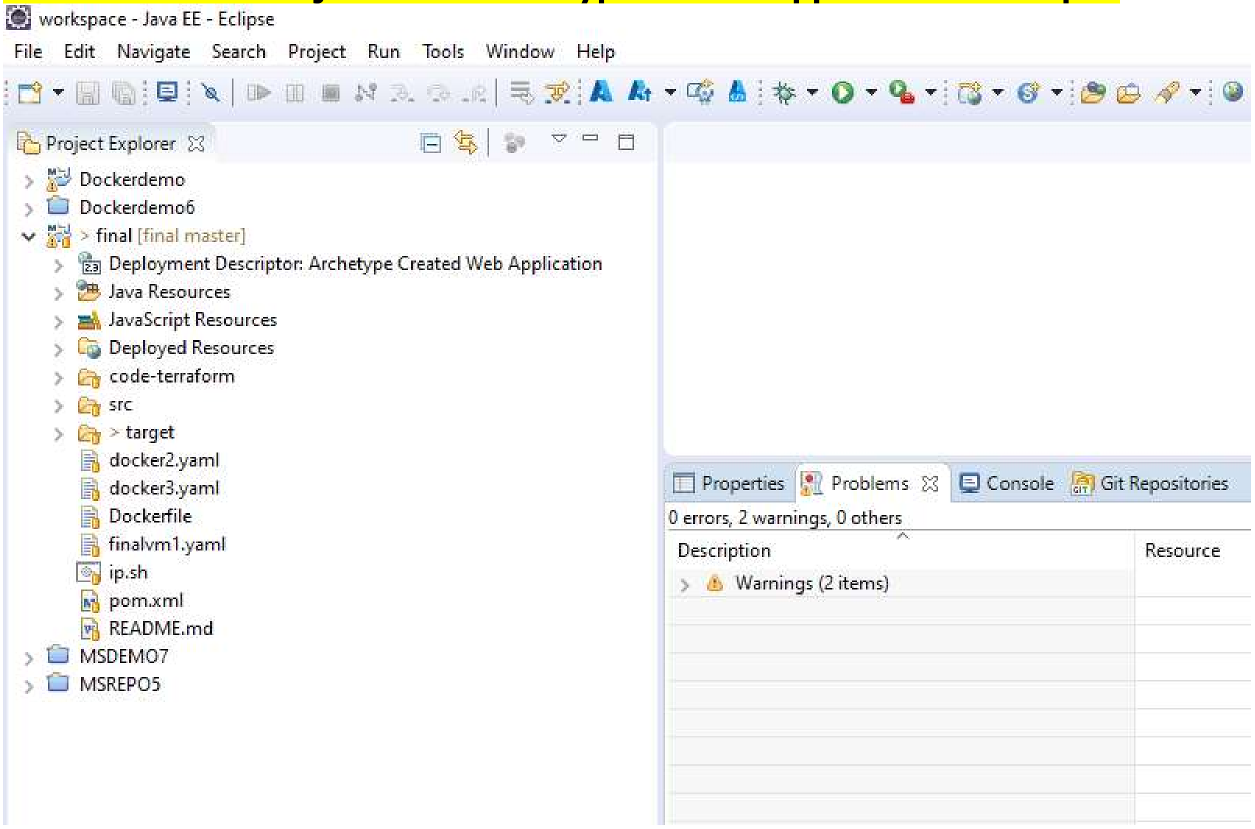
create a shell script to get VM ip and updating it in Inventory File

Push playbook1,playbook2 and shell scrip to Remote Git repo created in Phase3

Check for the change in remote Repo

From Build server run the playbook1,playbook2 and shell script to test for the required result

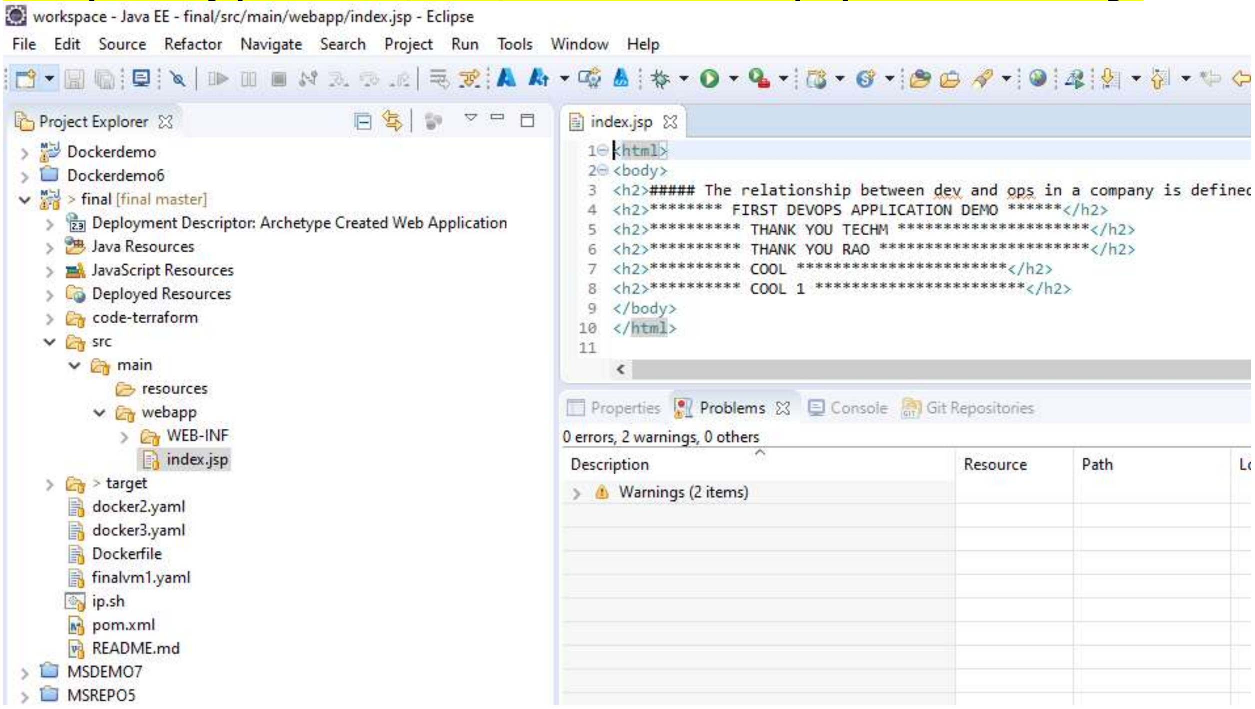
Created Maven Project with Archetype as web application in eclipse


workspace - Java EE - Eclipse
File Edit Navigate Search Project Run Tools Window Help
Project Explorer
Properties
Problems
Console
Git Repositories

```
Dockerdemo
Dockerdemo6
> final [final master]
    Deployment Descriptor: Archetype Created Web Application
    Java Resources
    JavaScript Resources
    Deployed Resources
    code-terraform
    src
    > target
    docker2.yaml
    docker3.yaml
    Dockerfile
    finalvm1.yaml
    ip.sh
    pom.xml
    README.md
MSDEMO7
MSREPO5
```

```
0 errors, 2 warnings, 0 others
Description                      Resource
> Warnings (2 items)
```

Modify Index.jsp under src/main/web content to display a custom message

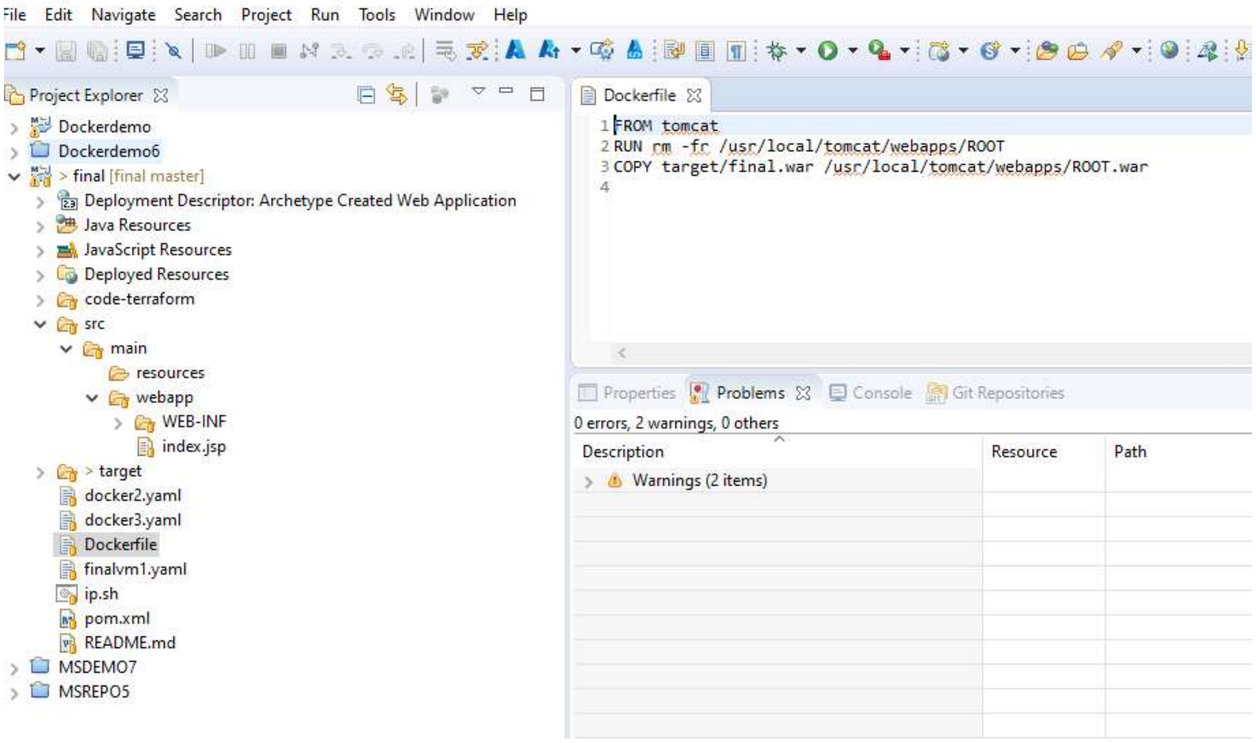

workspace - Java EE - final/src/main/webapp/index.jsp - Eclipse
File Edit Source Refactor Navigate Search Project Run Tools Window Help
Project Explorer
index.jsp
Properties
Problems
Console
Git Repositories

```
Dockerdemo
Dockerdemo6
> final [final master]
    Deployment Descriptor: Archetype Created Web Application
    Java Resources
    JavaScript Resources
    Deployed Resources
    code-terraform
    src
        main
            resources
            webapp
                WEB-INF
                index.jsp
    > target
    docker2.yaml
    docker3.yaml
    Dockerfile
    finalvm1.yaml
    ip.sh
    pom.xml
    README.md
MSDEMO7
MSREPO5
```

```
<html>
<body>
<h2>##### The relationship between dev and ops in a company is definec
<h2>******** FIRST DEVOPS APPLICATION DEMO ******</h2>
<h2>********** THANK YOU TECHM *********************</h2>
<h2>********** THANK YOU RAO ***********************</h2>
<h2>********** COOL ***********************</h2>
<h2>********** COOL 1 ***********************</h2>
</body>
</html>
```

```
0 errors, 2 warnings, 0 others
Description                      Resource     Path       Lo
> Warnings (2 items)
```

Generated Docker file under project folder of app and Modified FROM statement to use tomcat as base image

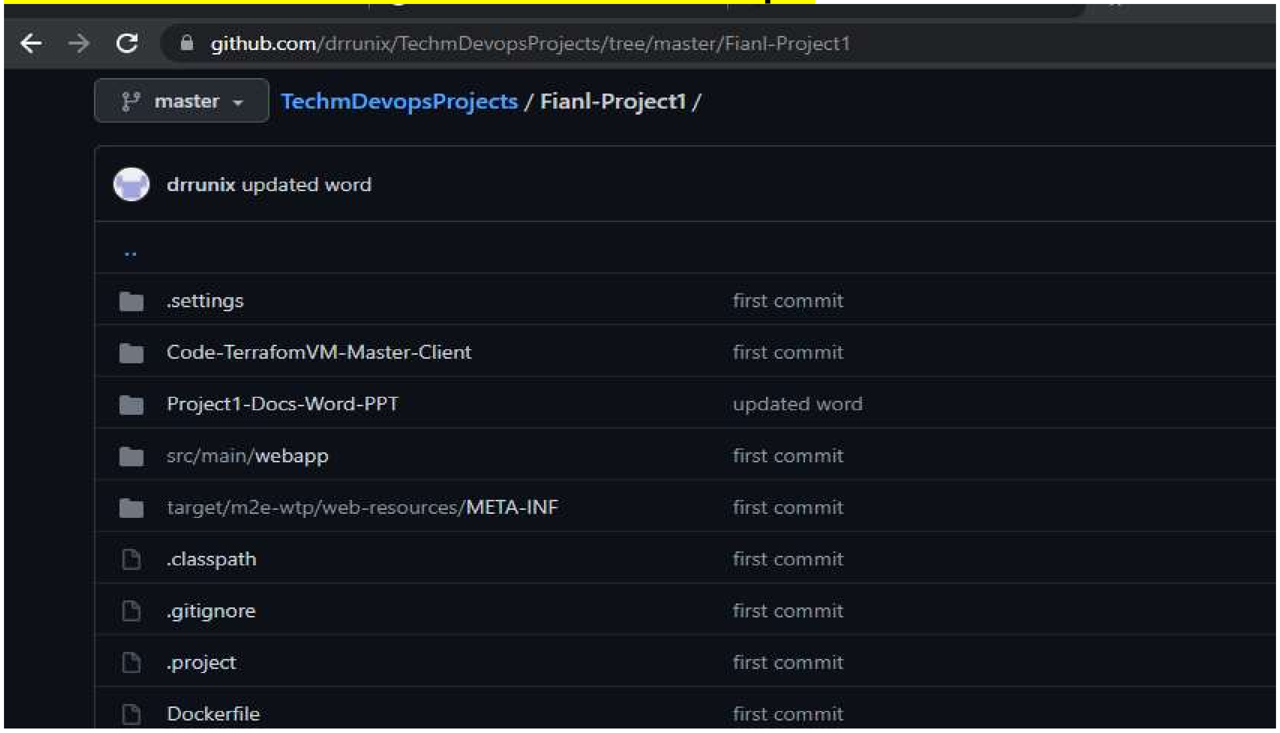


```
FROM tomcat
RUN rm -fr /usr/local/tomcat/webapps/ROOT
COPY target/final.war /usr/local/tomcat/webapps/ROOT.war
```

0 errors, 2 warnings, 0 others

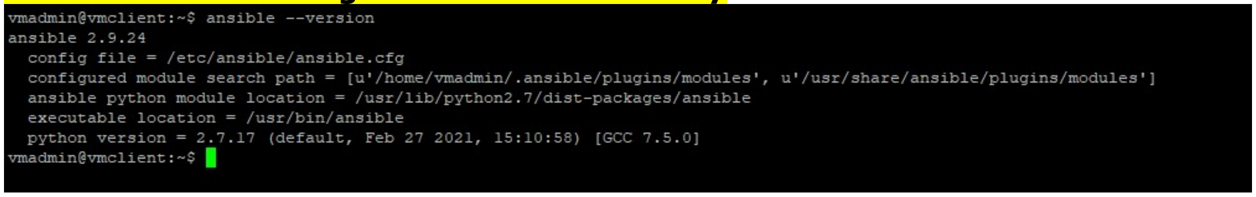
Description	Resource	Path
Warnings (2 items)		

Created a GitHub repository and copied repo URL, In Eclipse converted the app into a local repo from Team menu share Project Option and performed Commit and Pushed the code to the remote repo.


github.com/drrunix/TechmDevopsProjects/tree/master/Fianl-Project1
master
TechmDevopsProjects / Fianl-Project1 /
drrunix updated word

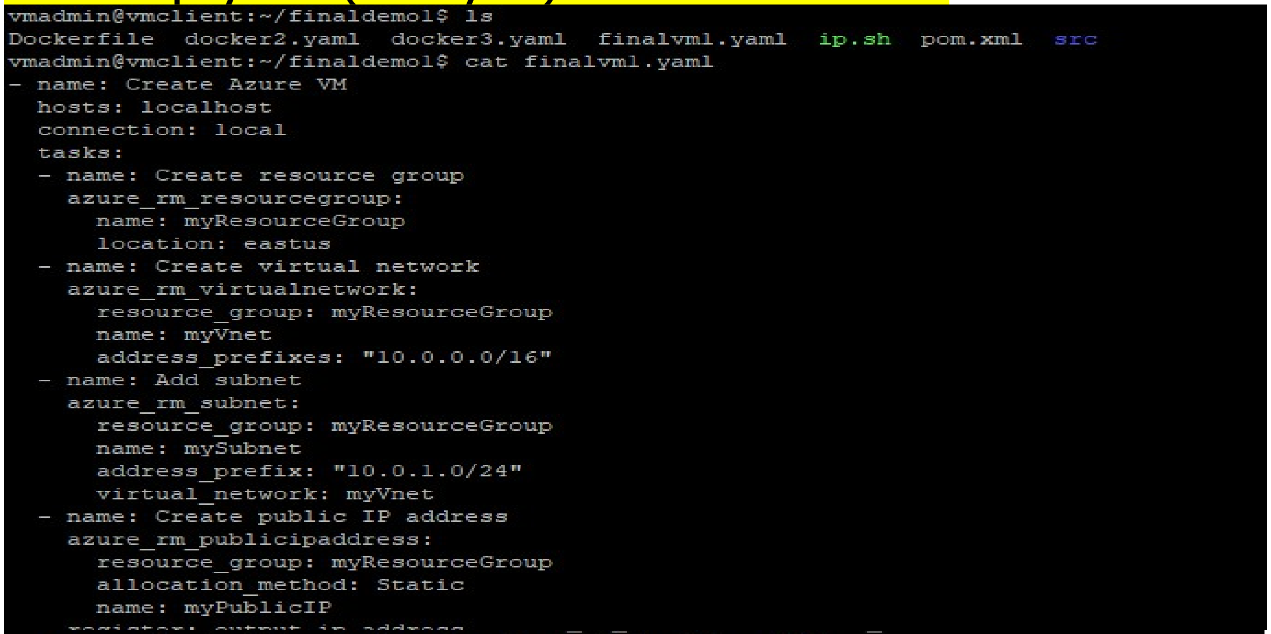
.settings	first commit
Code-TerrafomVM-Master-Client	first commit
Project1-Docs-Word-PPT	updated word
src/main/webapp	first commit
target/m2e-wtp/web-resources/META-INF	first commit
.classpath	first commit
.gitignore	first commit
.project	first commit
Dockerfile	first commit

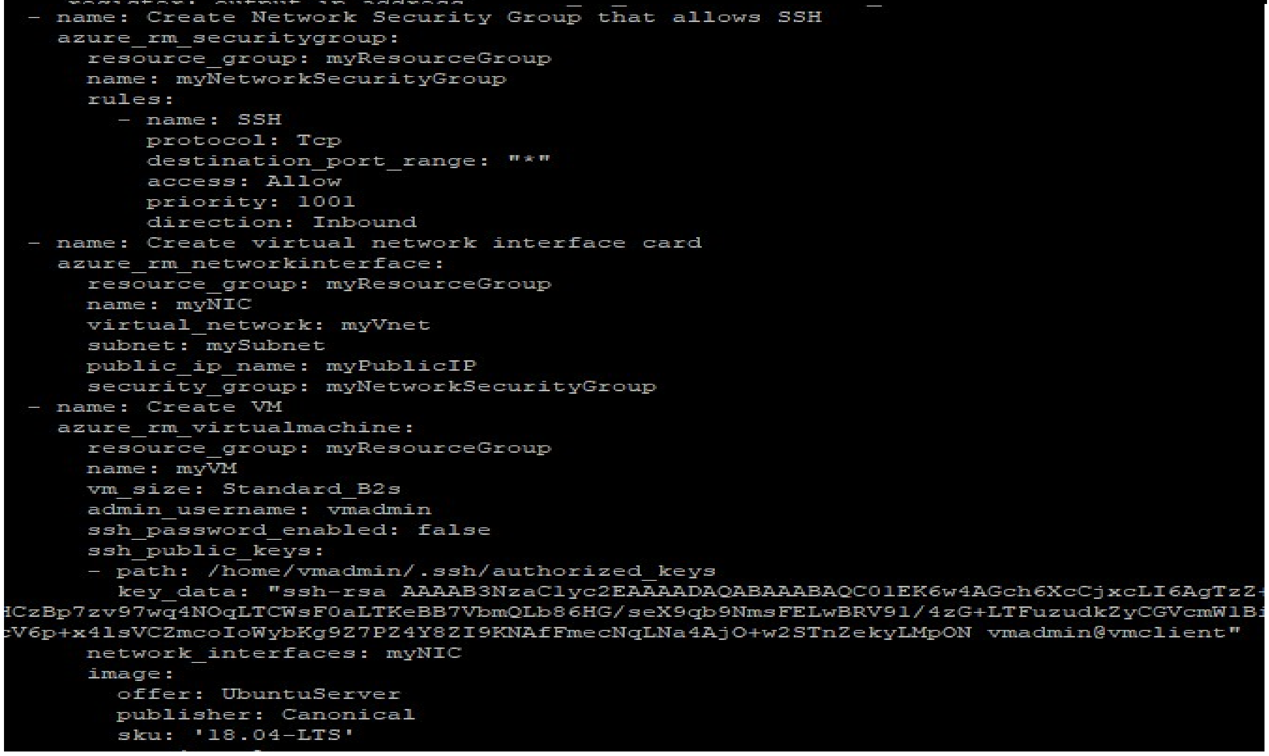
In build server configured Ansible manually



```
vmadmin@vmclient:~$ ansible --version
ansible 2.9.24
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/home/vmadmin/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.17 (default, Feb 27 2021, 15:10:58) [GCC 7.5.0]
vmadmin@vmclient:~$
```

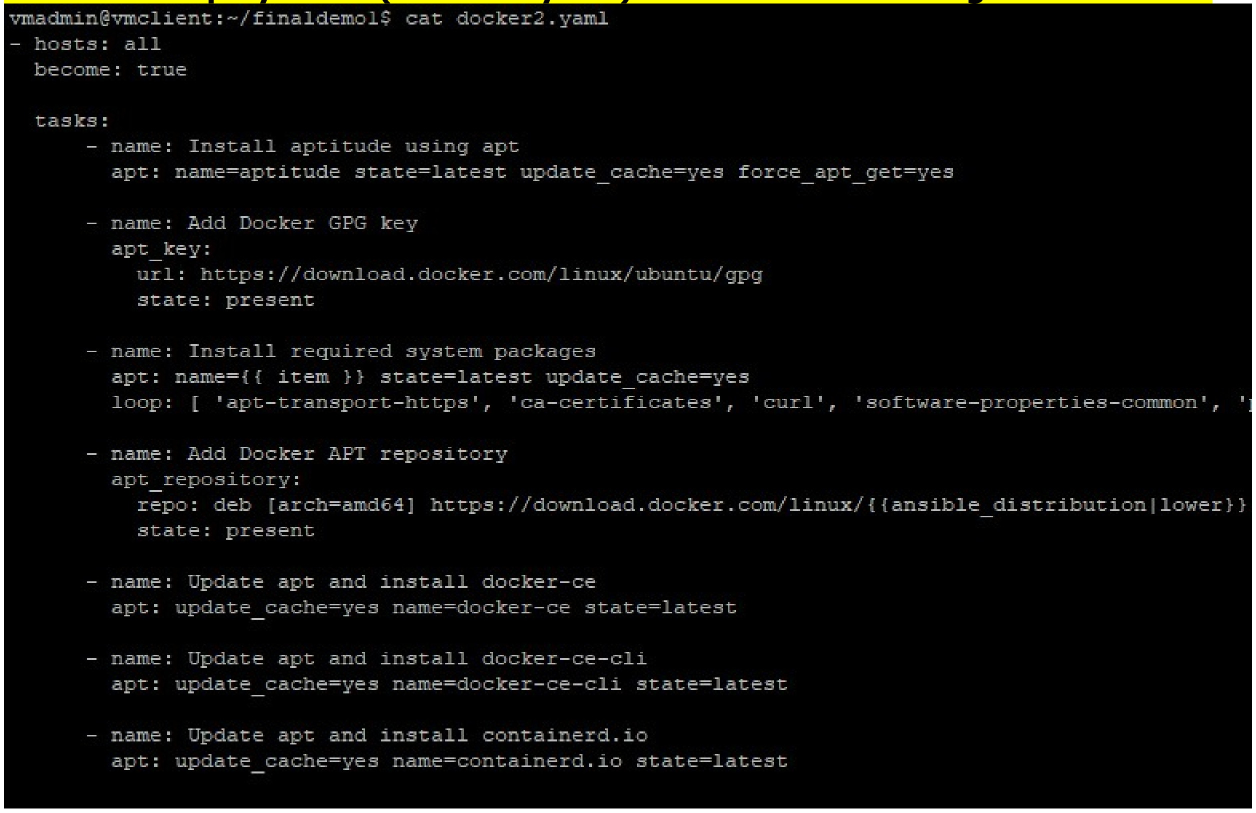
Created a playbook1(final1.yaml) to create a VM in azure





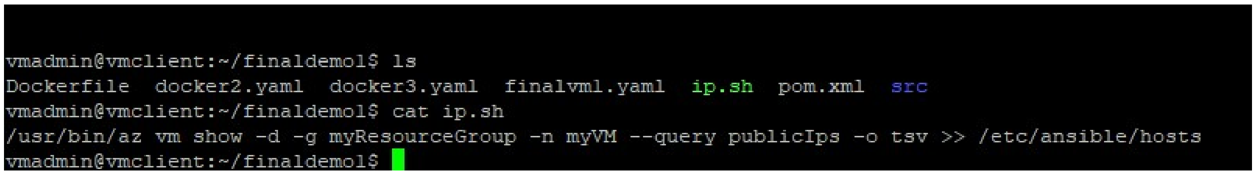
```
vmadmin@vmclient:~/finaldemo1$ ls
Dockerfile  docker2.yaml  docker3.yaml  finalvm1.yaml  ip.sh  pom.xml  src
vmadmin@vmclient:~/finaldemo1$ cat finalvm1.yaml
- name: Create Azure VM
  hosts: localhost
  connection: local
  tasks:
  - name: Create resource group
    azure_rm_resourcegroup:
      name: myResourceGroup
      location: eastus
  - name: Create virtual network
    azure_rm_virtualnetwork:
      resource_group: myResourceGroup
      name: myVnet
      address_prefixes: "10.0.0.0/16"
  - name: Add subnet
    azure_rm_subnet:
      resource_group: myResourceGroup
      name: mySubnet
      address_prefix: "10.0.1.0/24"
      virtual_network: myVnet
  - name: Create public IP address
    azure_rm_publicipaddress:
      resource_group: myResourceGroup
      allocation_method: Static
      name: myPublicIP
  - name: Create Network Security Group that allows SSH
    azure_rm_securitygroup:
      resource_group: myResourceGroup
      name: myNetworkSecurityGroup
      rules:
        - name: SSH
          protocol: Tcp
          destination_port_range: "*"
          access: Allow
          priority: 1001
          direction: Inbound
  - name: Create virtual network interface card
    azure_rm_networkinterface:
      resource_group: myResourceGroup
      name: myNIC
      virtual_network: myVnet
      subnet: mySubnet
      public_ip_name: myPublicIP
      security_group: myNetworkSecurityGroup
  - name: Create VM
    azure_rm_virtualmachine:
      resource_group: myResourceGroup
      name: myVM
      vm_size: Standard_B2s
      admin_username: vmadmin
      ssh_password_enabled: false
      ssh_public_keys:
      - path: /home/vmadmin/.ssh/authorized_keys
        key_data: "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC01EK6w4AGch6XcCjxcLI6AgTzZ
HCzBp7zv97wq4NOqLTCWsF0aLTKeBB7VbmQLb86HG/seX9qb9NmsFELwBRV91/4zG+LTFuzudkZyCGVcmW1Bi
cV6p+x41sVCZmcoIoWybKg9Z7PZ4Y8ZI9KNAfFmecNqLNa4AjO+w2STnZekyLMpON vmadmin@vmclient"
      network_interfaces: myNIC
      image:
        offer: UbuntuServer
        publisher: Canonical
        sku: '18.04-LTS'
```

Created the playbook2(Docker2.yaml) to install Docker engine on the VM



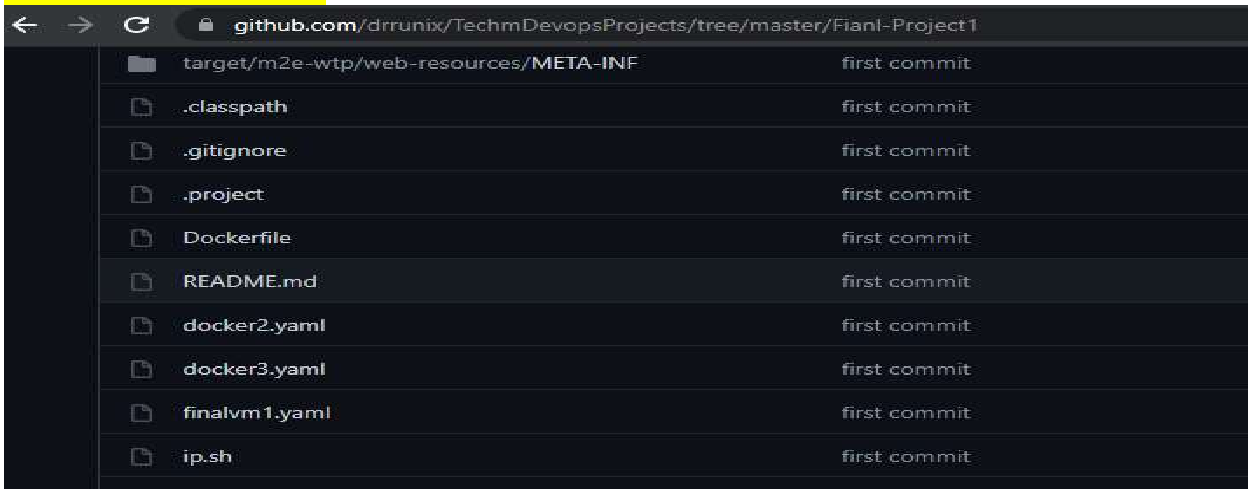
```
vmadmin@vmclient:~/finaldemo1$ cat docker2.yaml
- hosts: all
  become: true

  tasks:
      - name: Install aptitude using apt
        apt: name=aptitude state=latest update_cache=yes force_apt_get=yes

      - name: Add Docker GPG key
        apt_key:
          url: https://download.docker.com/linux/ubuntu/gpg
          state: present

      - name: Install required system packages
        apt: name={{ item }} state=latest update_cache=yes
        loop: [ 'apt-transport-https', 'ca-certificates', 'curl', 'software-properties-common', '

      - name: Add Docker APT repository
        apt_repository:
          repo: deb [arch=amd64] https://download.docker.com/linux/{{ansible_distribution|lower}}
          state: present

      - name: Update apt and install docker-ce
        apt: update_cache=yes name=docker-ce state=latest

      - name: Update apt and install docker-ce-cli
        apt: update_cache=yes name=docker-ce-cli state=latest

      - name: Update apt and install containerd.io
        apt: update_cache=yes name=containerd.io state=latest
```

Created a shell script(ip.sh) to get VM IP and updating it in Inventory File



```
vmadmin@vmclient:~/finaldemo1$ ls
Dockerfile  docker2.yaml  docker3.yaml  finalvm1.yaml  ip.sh  pom.xml  src
vmadmin@vmclient:~/finaldemo1$ cat ip.sh
/usr/bin/az vm show -d -g myResourceGroup -n myVM --query publicIps -o tsv >> /etc/ansible/hosts
vmadmin@vmclient:~/finaldemo1$
```

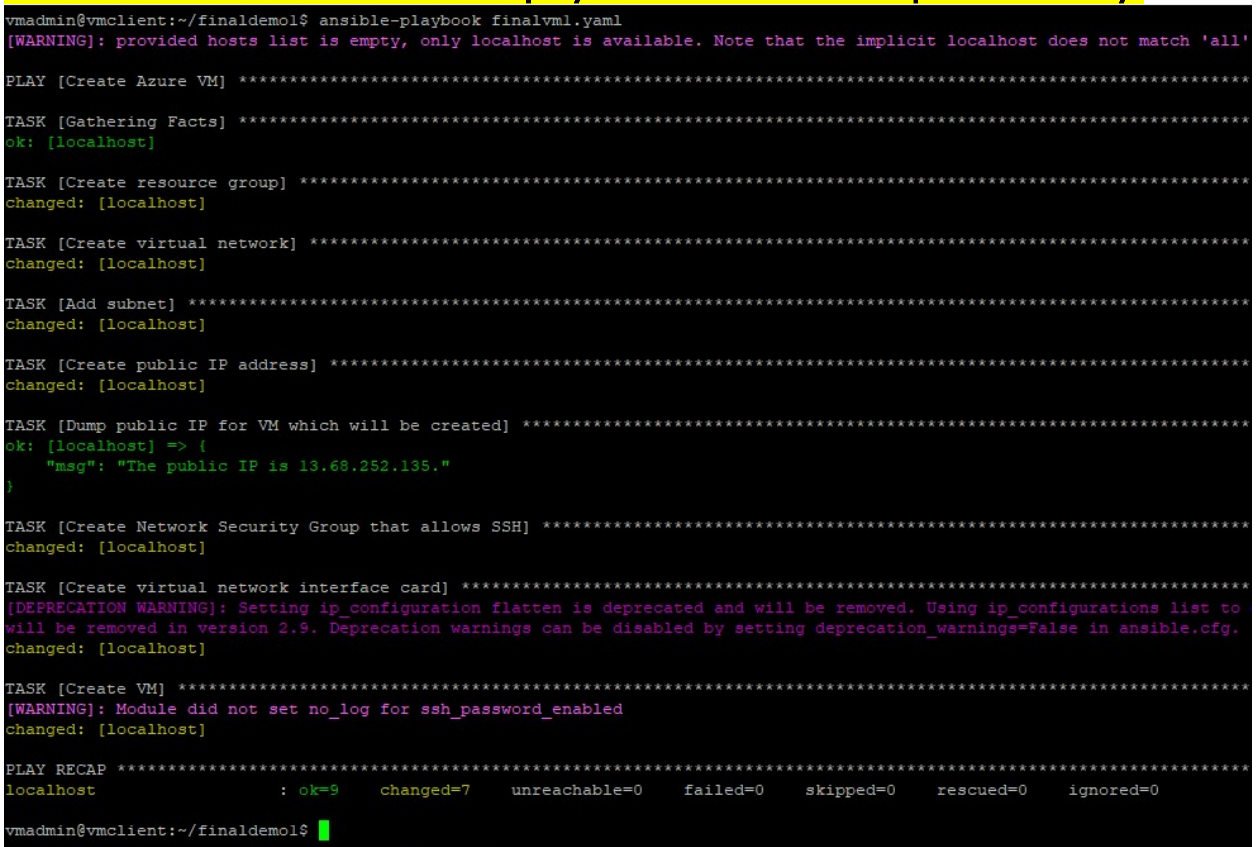
Push fianl1.yaml, docket2.yaml2 and shell script to Remote Git repo created in Phase3



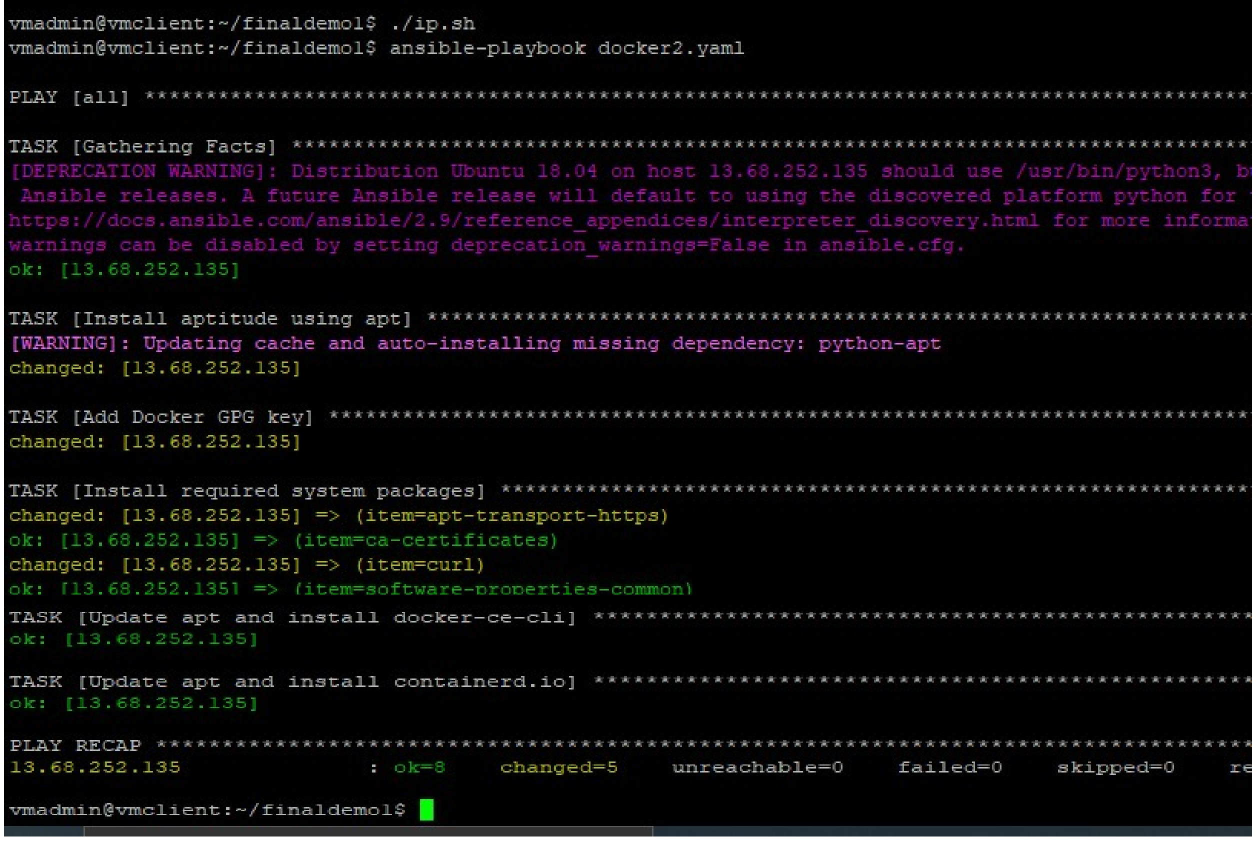
github.com/drrunix/TechmDevopsProjects/tree/master/Fianl-Project1

target/m2e-wtp/web-resources/META-INF	first commit
.classpath	first commit
.gitignore	first commit
.project	first commit
Dockerfile	first commit
README.md	first commit
docker2.yaml	first commit
docker3.yaml	first commit
finalvm1.yaml	first commit
ip.sh	first commit

From Build server able to run all 2 playbooks and one sh script successfully.

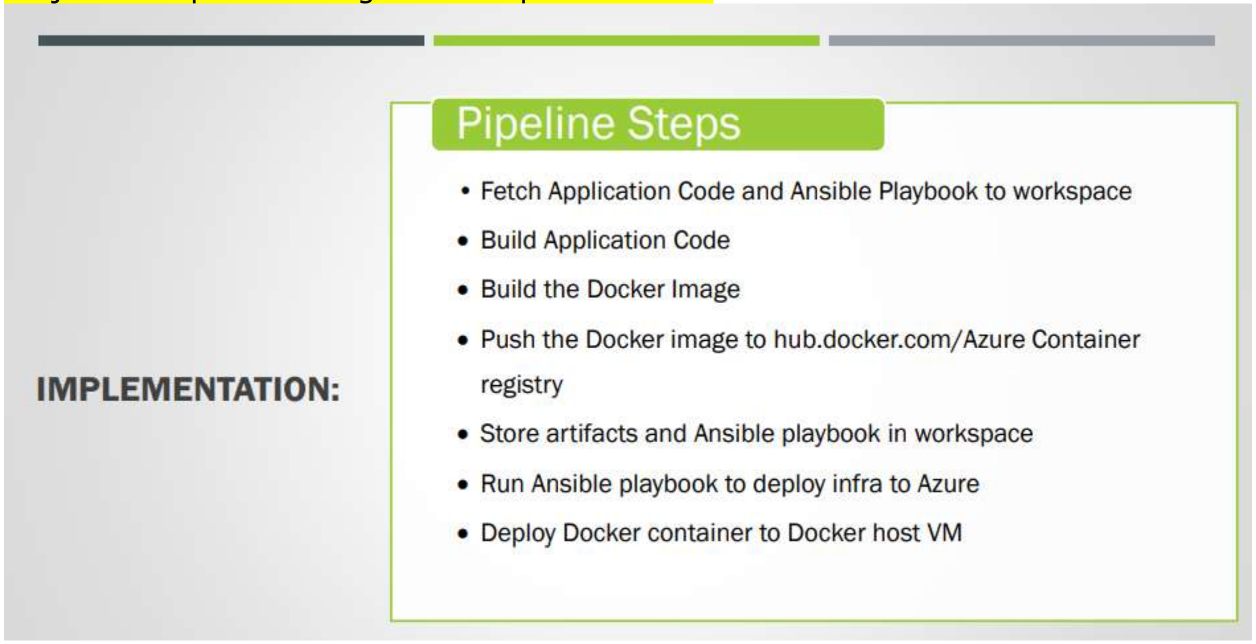


```
vmadmin@vmclient:~/finaldemo1$ ansible-playbook finalvm1.yaml
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'

PLAY [Create Azure VM] *********************************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [localhost]

TASK [Create resource group] ***************************************************************************************
changed: [localhost]

TASK [Create virtual network] **************************************************************************************
changed: [localhost]

TASK [Add subnet] **************************************************************************************************
changed: [localhost]

TASK [Create public IP address] ************************************************************************************
changed: [localhost]

TASK [Dump public IP for VM which will be created] *****************************************************************
ok: [localhost] => {
    "msg": "The public IP is 13.68.252.135."
}

TASK [Create Network Security Group that allows SSH] ***************************************************************
changed: [localhost]

TASK [Create virtual network interface card] ***********************************************************************
[DEPRECATION WARNING]: Setting ip_configuration flatten is deprecated and will be removed. Using ip_configurations list to
will be removed in version 2.9. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
changed: [localhost]

TASK [Create VM] ***************************************************************************************************
[WARNING]: Module did not set no_log for ssh_password_enabled
changed: [localhost]

PLAY RECAP *********************************************************************************************************
localhost                  : ok=9    changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

vmadmin@vmclient:~/finaldemo1$
```



```
vmadmin@vmclient:~/finaldemo1$ ./ip.sh
vmadmin@vmclient:~/finaldemo1$ ansible-playbook docker2.yaml

PLAY [all] ******************************************************************************************

TASK [Gathering Facts] ******************************************************************************
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 13.68.252.135 should use /usr/bin/python3, b
 Ansible releases. A future Ansible release will default to using the discovered platform python for
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more informa
warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
ok: [13.68.252.135]

TASK [Install aptitude using apt] *******************************************************************
[WARNING]: Updating cache and auto-installing missing dependency: python-apt
changed: [13.68.252.135]

TASK [Add Docker GPG key] ***************************************************************************
changed: [13.68.252.135]

TASK [Install required system packages] *************************************************************
changed: [13.68.252.135] => (item=apt-transport-https)
ok: [13.68.252.135] => (item=ca-certificates)
changed: [13.68.252.135] => (item=curl)
ok: [13.68.252.135] => (item=software-properties-common)
TASK [Update apt and install docker-ce-cli] *********************************************
ok: [13.68.252.135]

TASK [Update apt and install containerd.io] *********************************************
ok: [13.68.252.135]

PLAY RECAP ******************************************************************************
13.68.252.135              : ok=8    changed=5    unreachable=0    failed=0    skipped=0    re

vmadmin@vmclient:~/finaldemo1$
```

Project1 – Pipelines - High level implementation

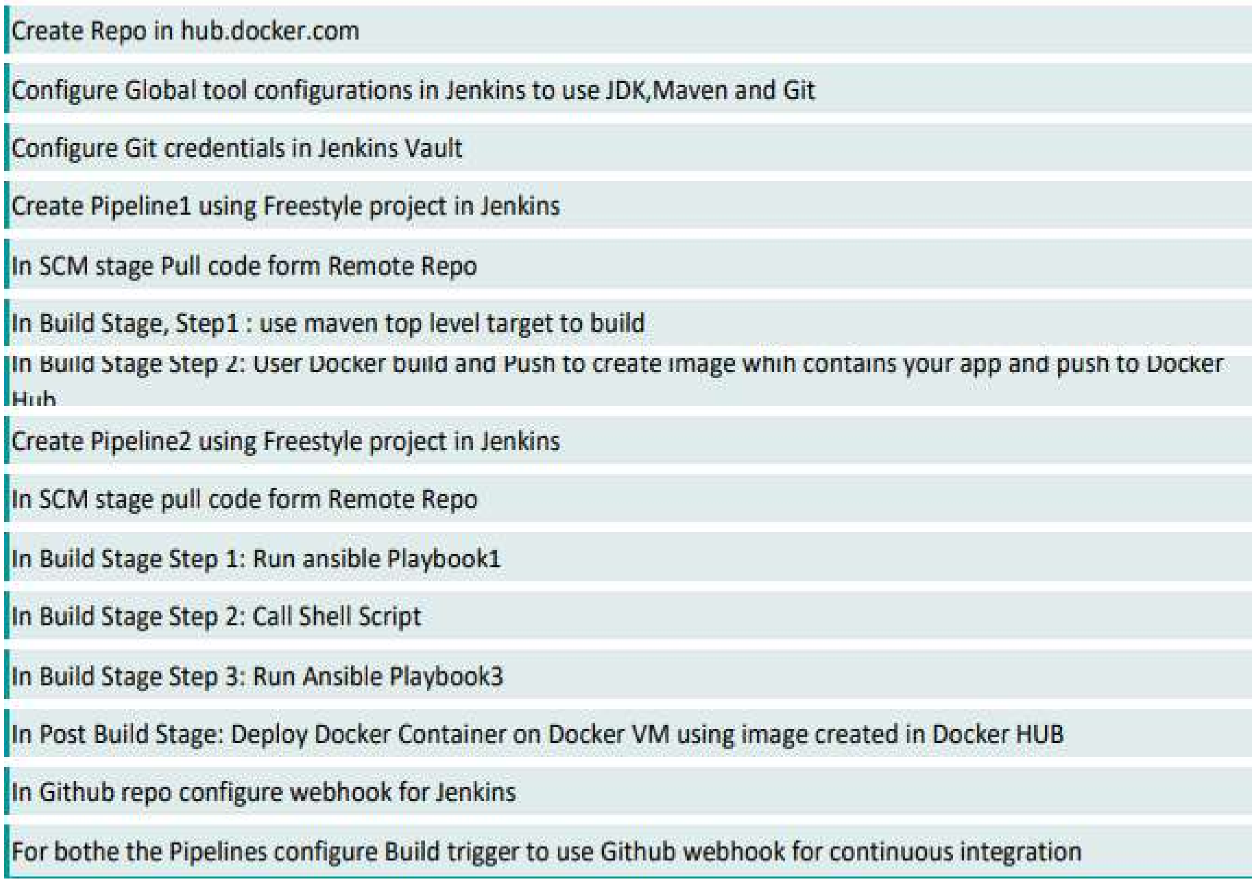


IMPLEMENTATION:

Pipeline Steps

- Fetch Application Code and Ansible Playbook to workspace
- Build Application Code
- Build the Docker Image
- Push the Docker image to hub.docker.com/Azure Container registry
- Store artifacts and Ansible playbook in workspace
- Run Ansible playbook to deploy infra to Azure
- Deploy Docker container to Docker host VM

Below are the detailed steps needed to perform



Create Repo in hub.docker.com

Configure Global tool configurations in Jenkins to use JDK,Maven and Git

Configure Git credentials in Jenkins Vault

Create Pipeline1 using Freestyle project in Jenkins

In SCM stage Pull code form Remote Repo

In Build Stage, Step1 : use maven top level target to build

In Build Stage Step 2: User Docker build and Push to create image whih contains your app and push to Docker Hub

Create Pipeline2 using Freestyle project in Jenkins

In SCM stage pull code form Remote Repo

In Build Stage Step 1: Run ansible Playbook1

In Build Stage Step 2: Call Shell Script

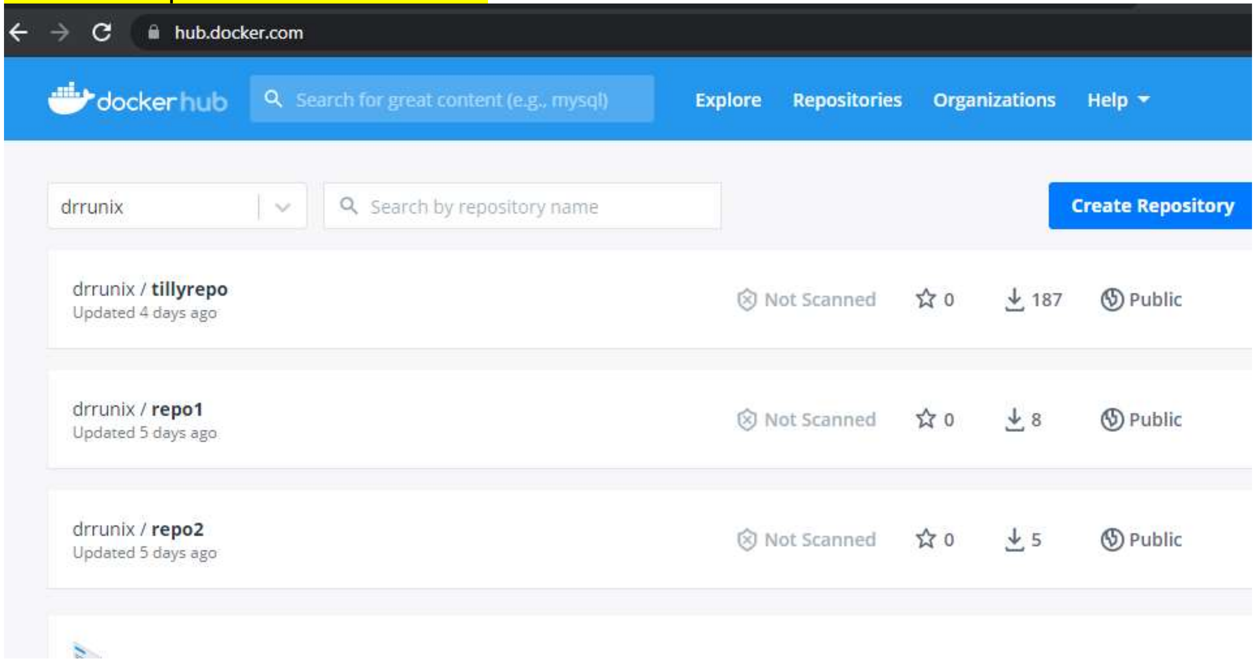
In Build Stage Step 3: Run Ansible Playbook3

In Post Build Stage: Deploy Docker Container on Docker VM using image created in Docker HUB

In Github repo configure webhook for Jenkins

For bothe the Pipelines configure Build trigger to use Github webhook for continuous integration

Created Repo in hub.docker.com



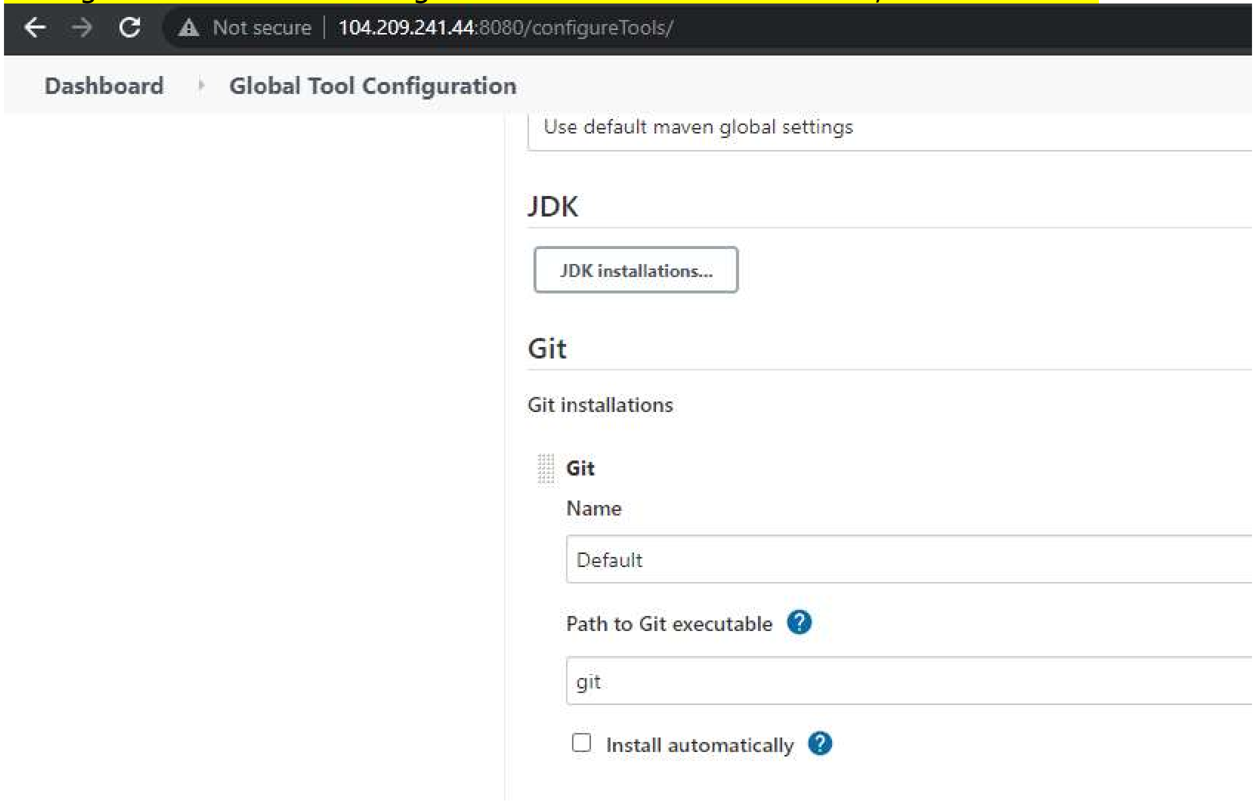
hub.docker.com

drrunix / **tillyrepo**
Updated 4 days ago — Not Scanned — 0 — 187 — Public

drrunix / **repo1**
Updated 5 days ago — Not Scanned — 0 — 8 — Public

drrunix / **repo2**
Updated 5 days ago — Not Scanned — 0 — 5 — Public

Configured Global tool configurations in Jenkins to use JDK,Maven and Git



Not secure | 104.209.241.44:8080/configureTools/

Use default maven global settings

JDK

Git

Git installations

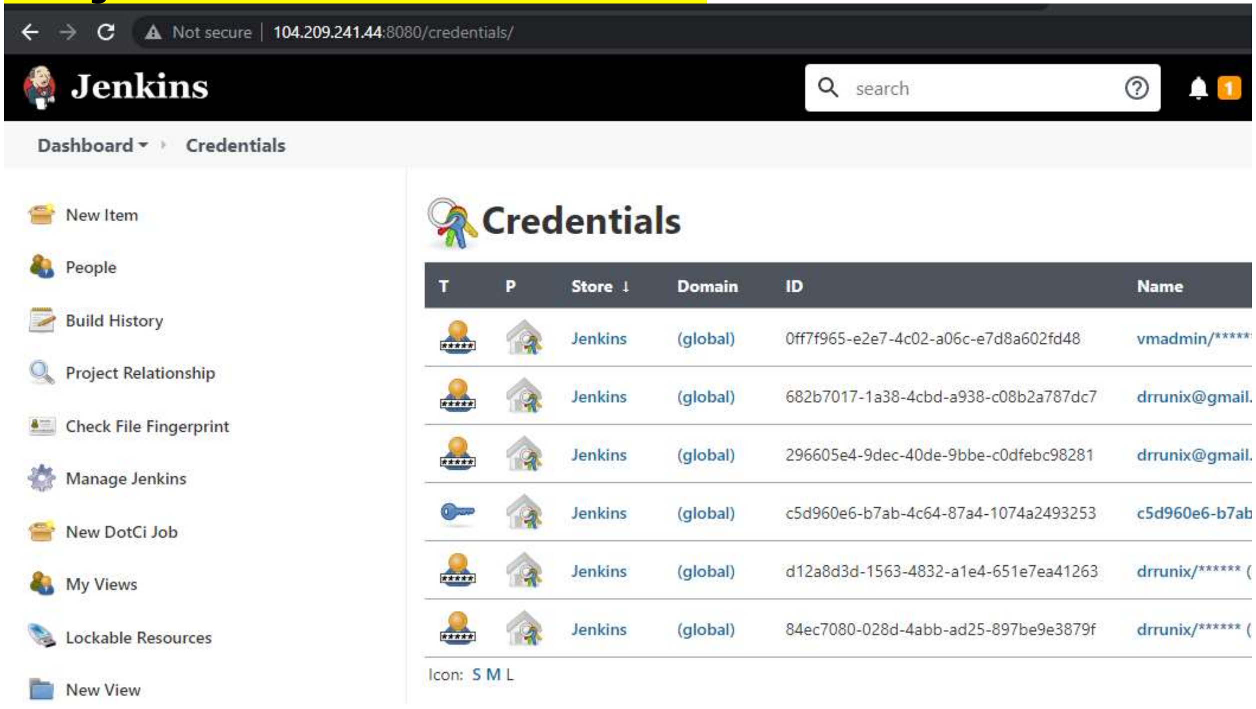
Git

Name: Default

Path to Git executable: git

☐ Install automatically

Configured Git credentials in Jenkins Vault



Not secure | 104.209.241.44:8080/credentials/

Jenkins

Dashboard › Credentials

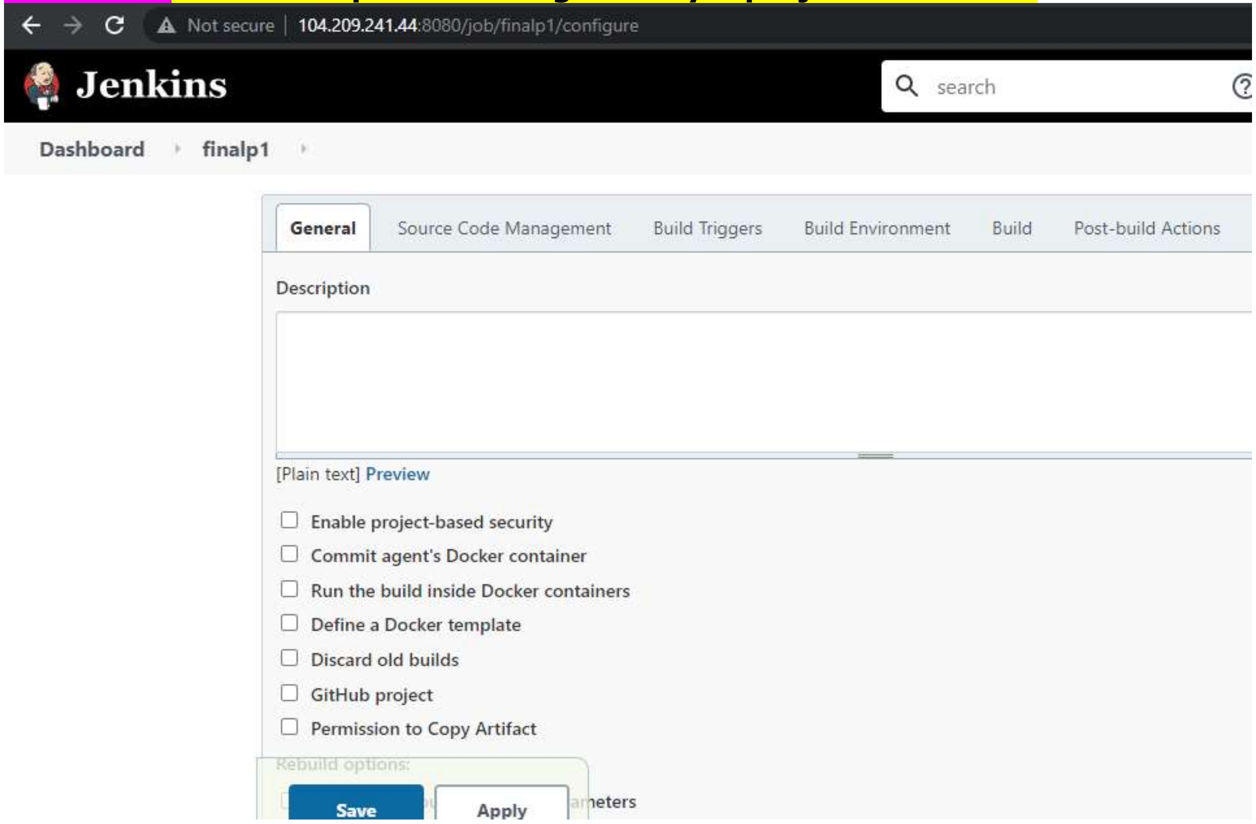
New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
New DotCi Job
My Views
Lockable Resources
New View

Credentials

T	P	Store ↓	Domain	ID	Name
		Jenkins	(global)	0ff7f965-e2e7-4c02-a06c-e7d8a602fd48	vmadmin/*****
		Jenkins	(global)	682b7017-1a38-4cbd-a938-c08b2a787dc7	drrunix@gmail.
		Jenkins	(global)	296605e4-9dec-40de-9bbe-c0dfebc98281	drrunix@gmail.
		Jenkins	(global)	c5d960e6-b7ab-4c64-87a4-1074a2493253	c5d960e6-b7ab
		Jenkins	(global)	d12a8d3d-1563-4832-a1e4-651e7ea41263	drrunix/****** (
		Jenkins	(global)	84ec7080-028d-4abb-ad25-897be9e3879f	drrunix/****** (

Icon: S M L

PIPELINE1: Created Pipeline1 using Freestyle project in Jenkins



Not secure | 104.209.241.44:8080/job/finalp1/configure

Jenkins

Dashboard › finalp1 ›

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

Description

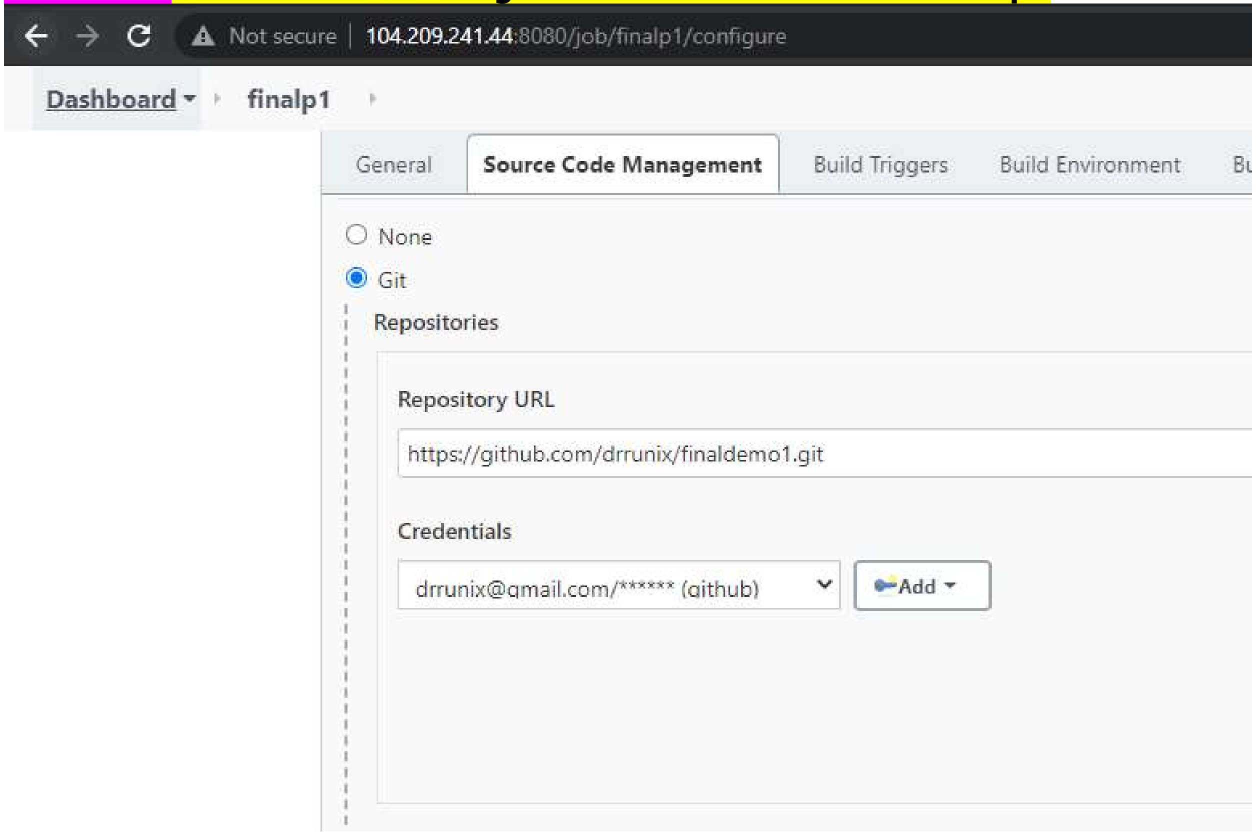
[Plain text] Preview

- [ ] Enable project-based security
- [ ] Commit agent's Docker container
- [ ] Run the build inside Docker containers
- [ ] Define a Docker template
- [ ] Discard old builds
- [ ] GitHub project
- [ ] Permission to Copy Artifact

Rebuild options:

Save Apply

PIPELINE1: **In Jenkins SCM stage Pulled code form Remote Repo**


Not secure | 104.209.241.44:8080/job/finalp1/configure

Dashboard ▸ finalp1

None

[✓ Git]

Repositories

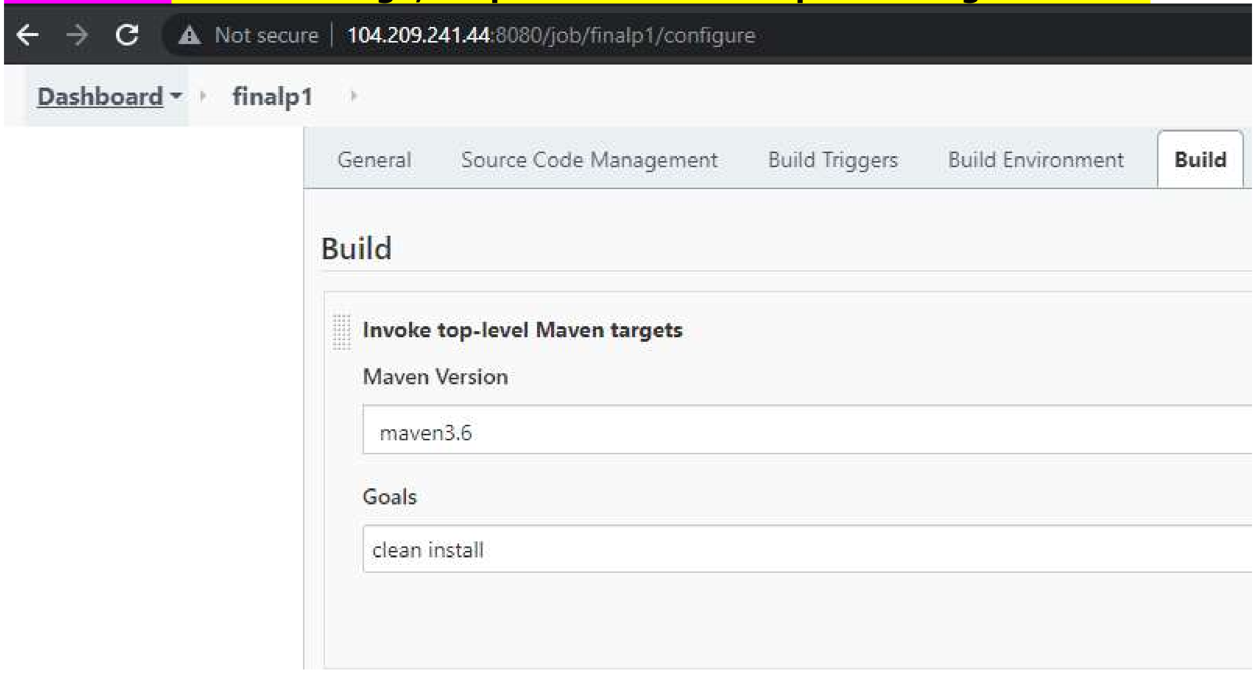
Repository URL

https://github.com/drrunix/finaldemo1.git

Credentials

drrunix@gmail.com/****** (github)

PIPELINE1: **In Build Stage, Step1: used maven top level target to build**


Not secure | 104.209.241.44:8080/job/finalp1/configure

Dashboard ▸ finalp1

Build

Invoke top-level Maven targets

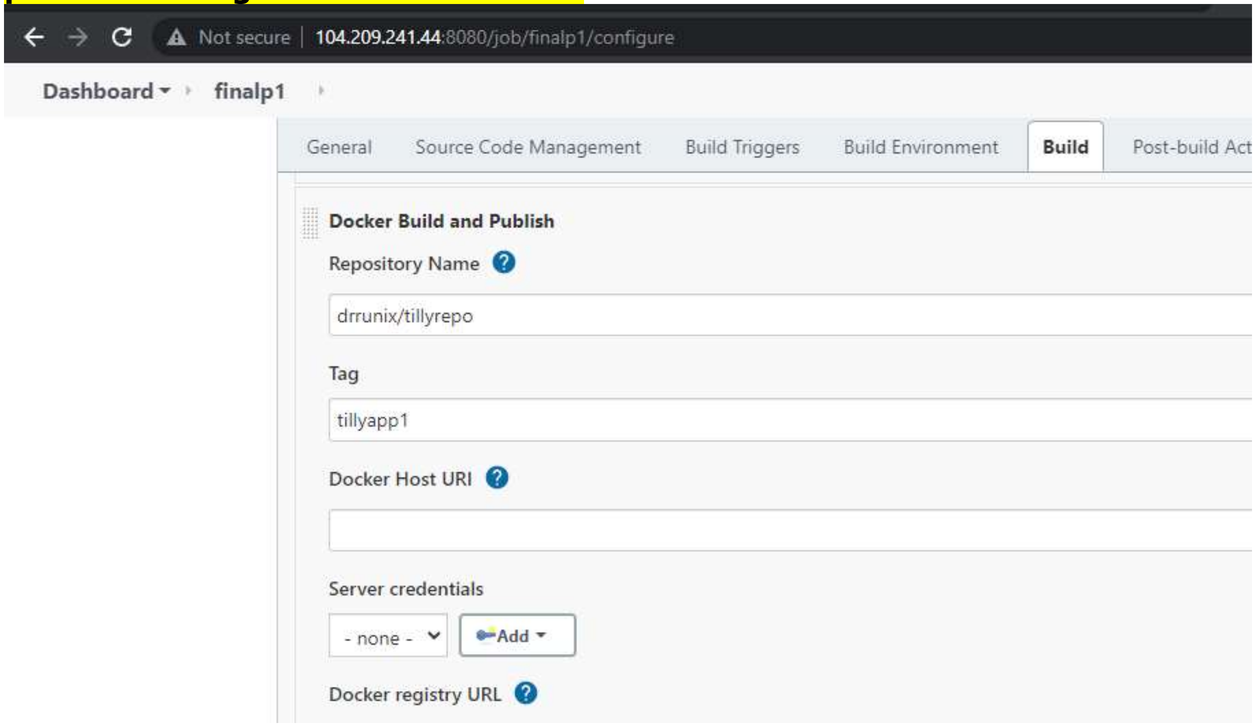
Maven Version

maven3.6

Goals

clean install

PIPELINE1:**in Build Stage Step 2: Used Docker build to create an image to push that image to the docker hub.**



Docker Build and Publish

Repository Name: drrunix/tillyrepo

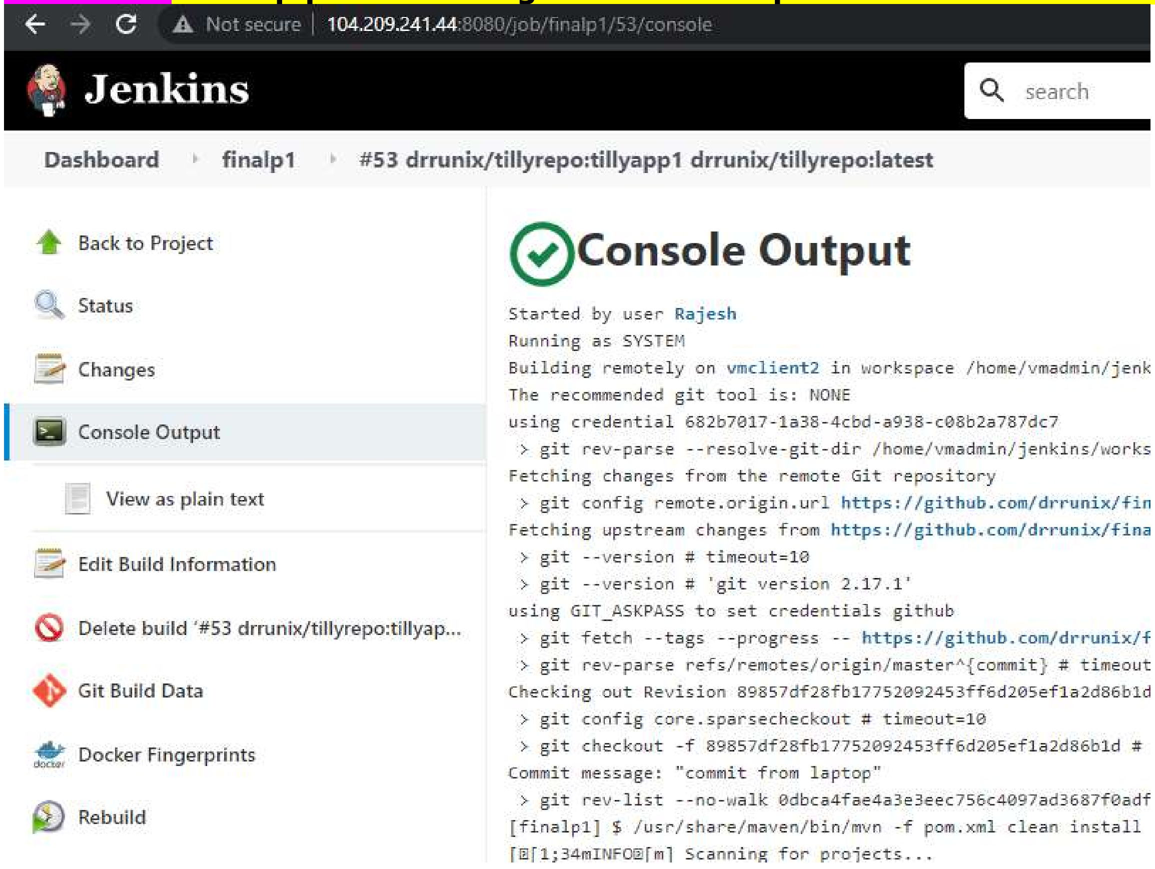
Tag: tillyapp1

Docker Host URI:

Server credentials: - none -

Docker registry URL

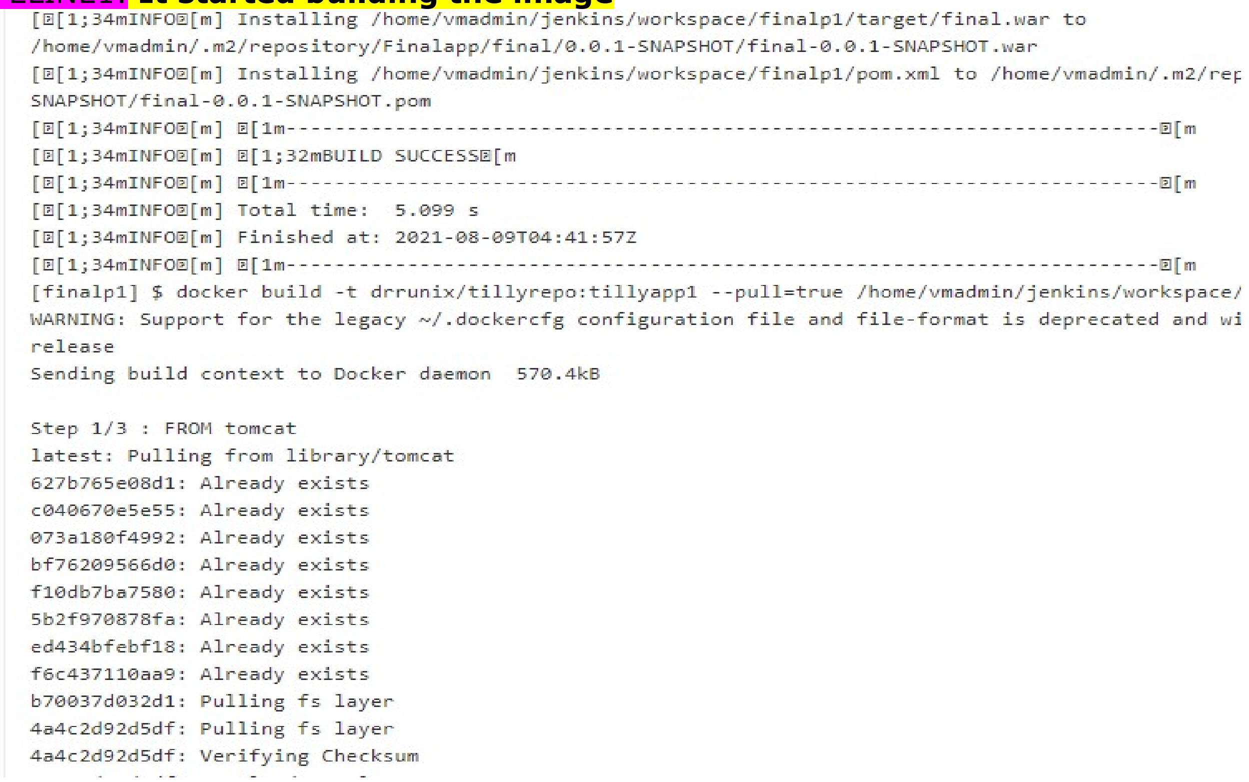
PIPELINE1: **Once pipeline1 configuration is complete and started build.**


Not secure | 104.209.241.44:8080/job/finalp1/53/console
Jenkins
search
Dashboard
finalp1
#53 drrunix/tillyrepo:tillyapp1 drrunix/tillyrepo:latest
Back to Project
Status
Changes
Console Output
View as plain text
Edit Build Information
Delete build '#53 drrunix/tillyrepo:tillyap...
Git Build Data
Docker Fingerprints
Rebuild

Console Output

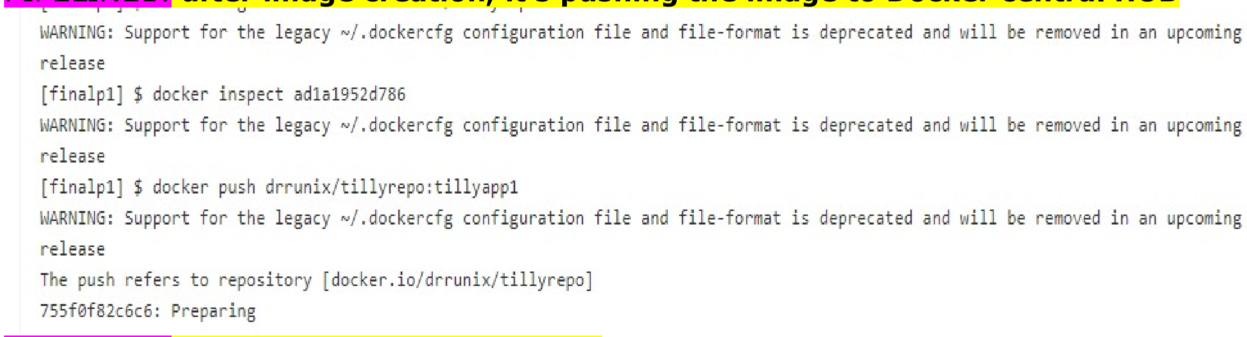
```
Started by user Rajesh
Running as SYSTEM
Building remotely on vmclient2 in workspace /home/vmadmin/jenk
The recommended git tool is: NONE
using credential 682b7017-1a38-4cbd-a938-c08b2a787dc7
 > git rev-parse --resolve-git-dir /home/vmadmin/jenkins/works
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/drrunix/fin
Fetching upstream changes from https://github.com/drrunix/fina
 > git --version # timeout=10
 > git --version # 'git version 2.17.1'
using GIT_ASKPASS to set credentials github
 > git fetch --tags --progress -- https://github.com/drrunix/f
 > git rev-parse refs/remotes/origin/master^{commit} # timeout
Checking out Revision 89857df28fb17752092453ff6d205ef1a2d86b1d
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 89857df28fb17752092453ff6d205ef1a2d86b1d #
Commit message: "commit from laptop"
 > git rev-list --no-walk 0dbca4fae4a3e3eec756c4097ad3687f0adf
[finalp1] $ /usr/share/maven/bin/mvn -f pom.xml clean install
[⌧[1;34mINFO⌧[m] Scanning for projects...
```

PIPELINE1: It started building the image



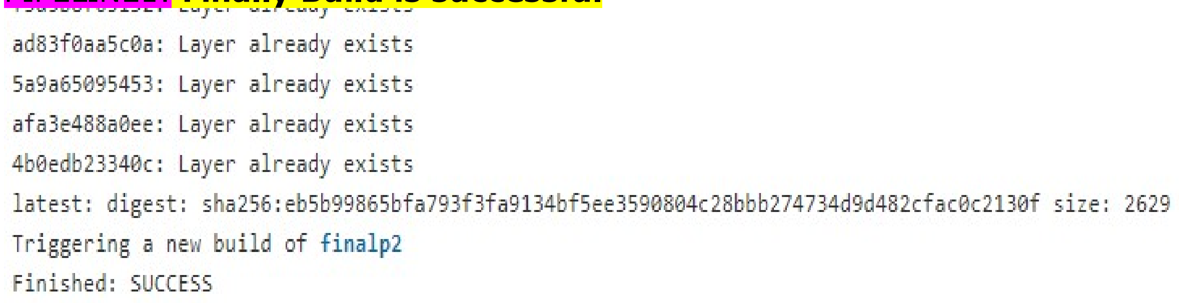
```
[⊡[1;34mINFO⊡[m] Installing /home/vmadmin/jenkins/workspace/finalp1/target/final.war to
/home/vmadmin/.m2/repository/Finalapp/final/0.0.1-SNAPSHOT/final-0.0.1-SNAPSHOT.war
[⊡[1;34mINFO⊡[m] Installing /home/vmadmin/jenkins/workspace/finalp1/pom.xml to /home/vmadmin/.m2/rep
SNAPSHOT/final-0.0.1-SNAPSHOT.pom
[⊡[1;34mINFO⊡[m] ⊡[1m------------------------------------------------------------------------⊡[m
[⊡[1;34mINFO⊡[m] ⊡[1;32mBUILD SUCCESS⊡[m
[⊡[1;34mINFO⊡[m] ⊡[1m------------------------------------------------------------------------⊡[m
[⊡[1;34mINFO⊡[m] Total time:  5.099 s
[⊡[1;34mINFO⊡[m] Finished at: 2021-08-09T04:41:57Z
[⊡[1;34mINFO⊡[m] ⊡[1m------------------------------------------------------------------------⊡[m
[finalp1] $ docker build -t drrunix/tillyrepo:tillyapp1 --pull=true /home/vmadmin/jenkins/workspace/
WARNING: Support for the legacy ~/.dockercfg configuration file and file-format is deprecated and wi
release
Sending build context to Docker daemon  570.4kB

Step 1/3 : FROM tomcat
latest: Pulling from library/tomcat
627b765e08d1: Already exists
c040670e5e55: Already exists
073a180f4992: Already exists
bf76209566d0: Already exists
f10db7ba7580: Already exists
5b2f970878fa: Already exists
ed434bfebf18: Already exists
f6c437110aa9: Already exists
b70037d032d1: Pulling fs layer
4a4c2d92d5df: Pulling fs layer
4a4c2d92d5df: Verifying Checksum
```

PIPELINE1: after image creation, it's pushing the image to Docker central HUB



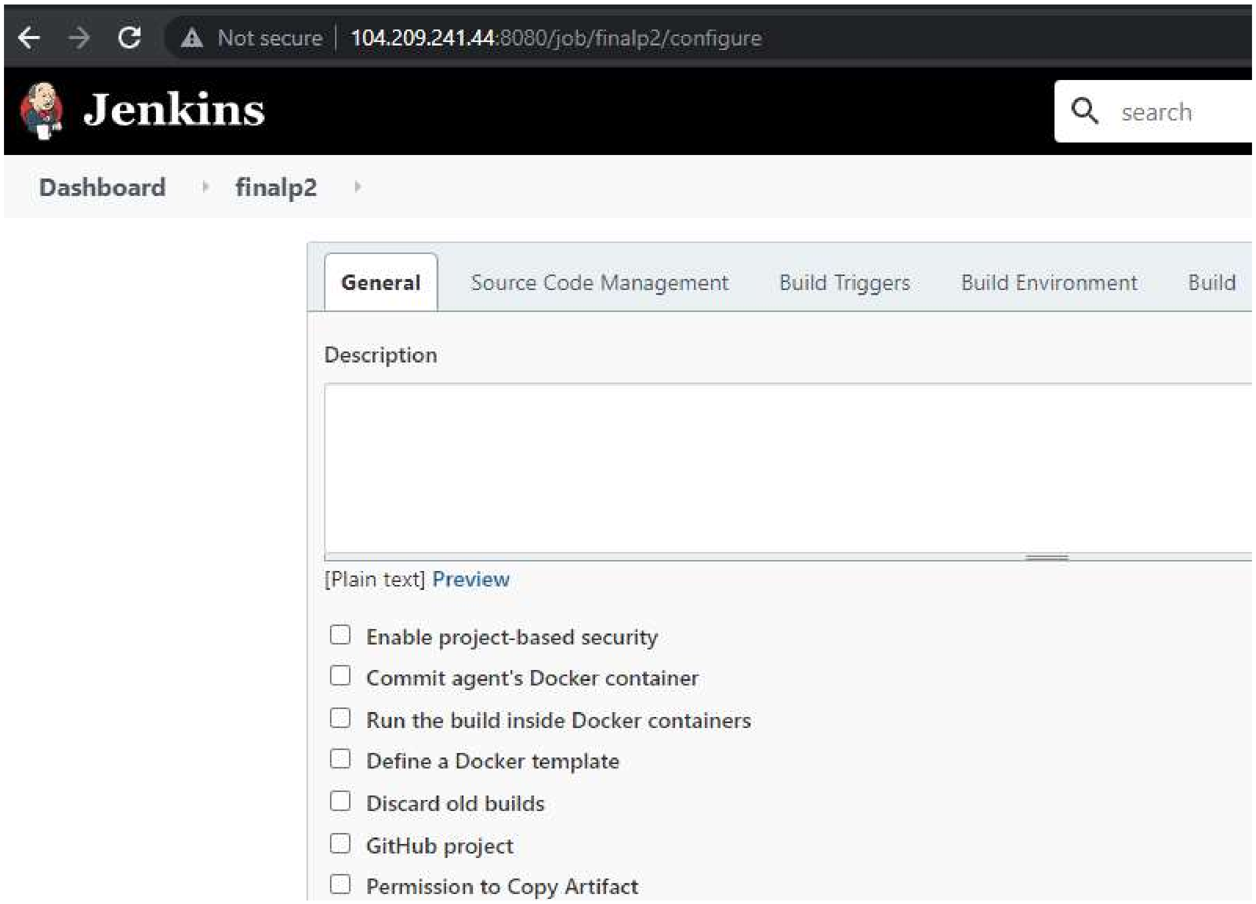
```
WARNING: Support for the legacy ~/.dockercfg configuration file and file-format is deprecated and will be removed in an upcoming
release
[finalp1] $ docker inspect ad1a1952d786
WARNING: Support for the legacy ~/.dockercfg configuration file and file-format is deprecated and will be removed in an upcoming
release
[finalp1] $ docker push drrunix/tillyrepo:tillyapp1
WARNING: Support for the legacy ~/.dockercfg configuration file and file-format is deprecated and will be removed in an upcoming
release
The push refers to repository [docker.io/drrunix/tillyrepo]
755f0f82c6c6: Preparing
```

PIPELINE1: Finally Build is successful



```
ad83f0aa5c0a: Layer already exists
5a9a65095453: Layer already exists
afa3e488a0ee: Layer already exists
4b0edb23340c: Layer already exists
latest: digest: sha256:eb5b99865bfa793f3fa9134bf5ee3590804c28bbb274734d9d482cfac0c2130f size: 2629
Triggering a new build of finalp2
Finished: SUCCESS
```

Created Pipeline2 using Freestyle project in Jenkins



104.209.241.44:8080/job/finalp2/configure

Jenkins

Dashboard › finalp2

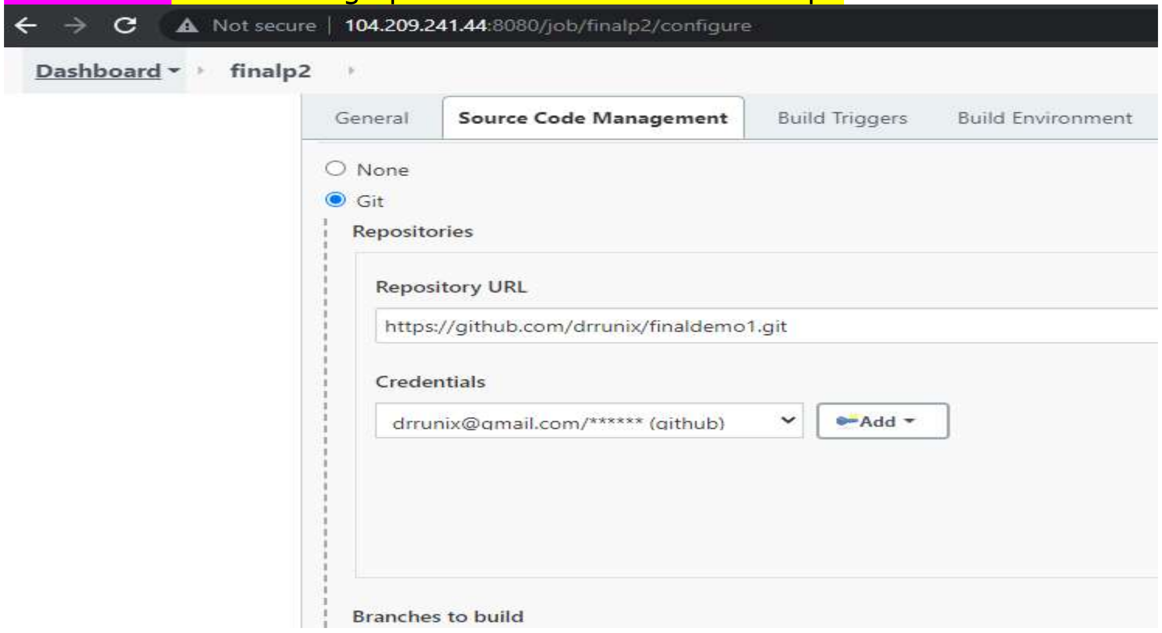
General | Source Code Management | Build Triggers | Build Environment | Build

Description

[Plain text] Preview

- [ ] Enable project-based security
- [ ] Commit agent's Docker container
- [ ] Run the build inside Docker containers
- [ ] Define a Docker template
- [ ] Discard old builds
- [ ] GitHub project
- [ ] Permission to Copy Artifact

PIPELINE2: In SCM stage pulled code form Remote Repo



104.209.241.44:8080/job/finalp2/configure

Dashboard › finalp2

General | Source Code Management | Build Triggers | Build Environment

None [✓ Git]

Repositories

Repository URL

https://github.com/drrunix/finaldemo1.git

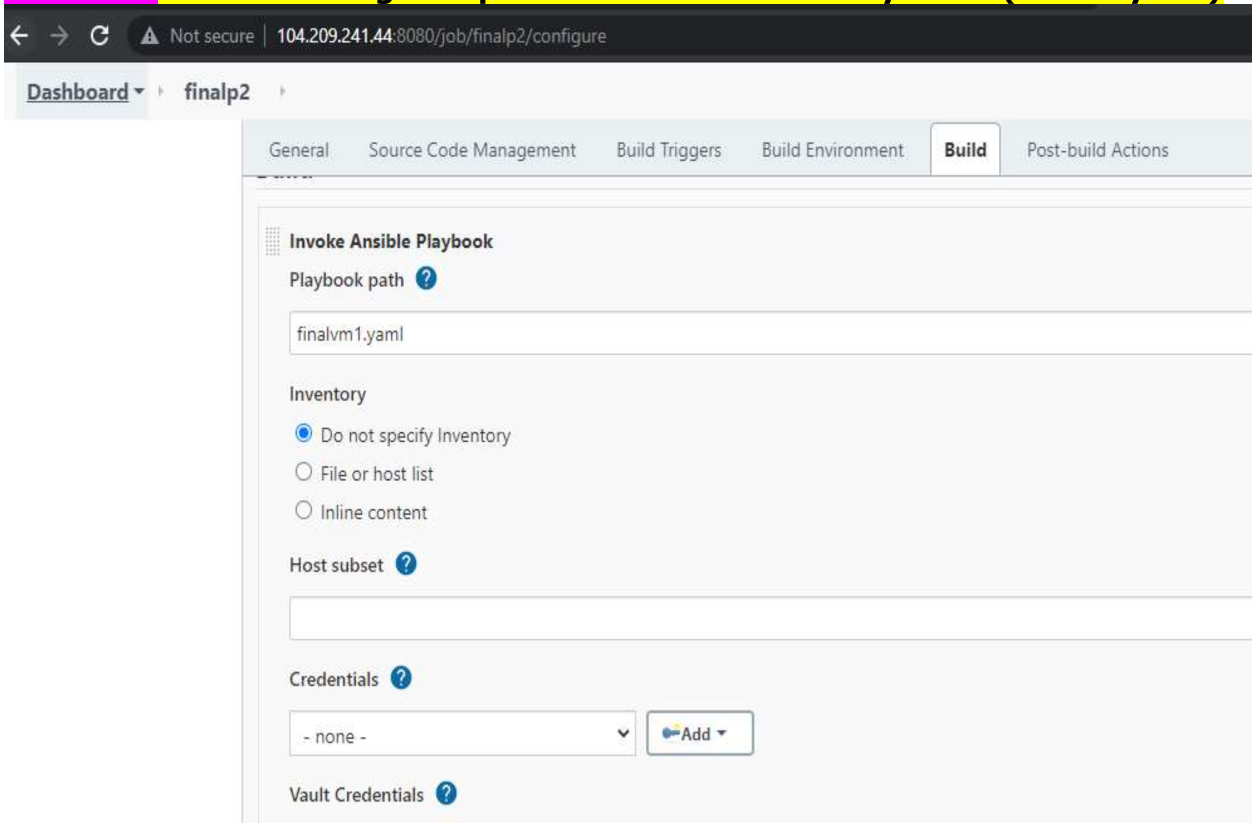
Credentials

drrunix@gmail.com/****** (github)

Add

Branches to build

PIPELINE2: In Build Stage Step 1: Selected ansible Playbook1(Final1.yaml)



Not secure | 104.209.241.44:8080/job/finalp2/configure

Dashboard ▾ › finalp2 ›

General | Source Code Management | Build Triggers | Build Environment | **Build** | Post-build Actions

Invoke Ansible Playbook

Playbook path: finalvm1.yaml

Inventory

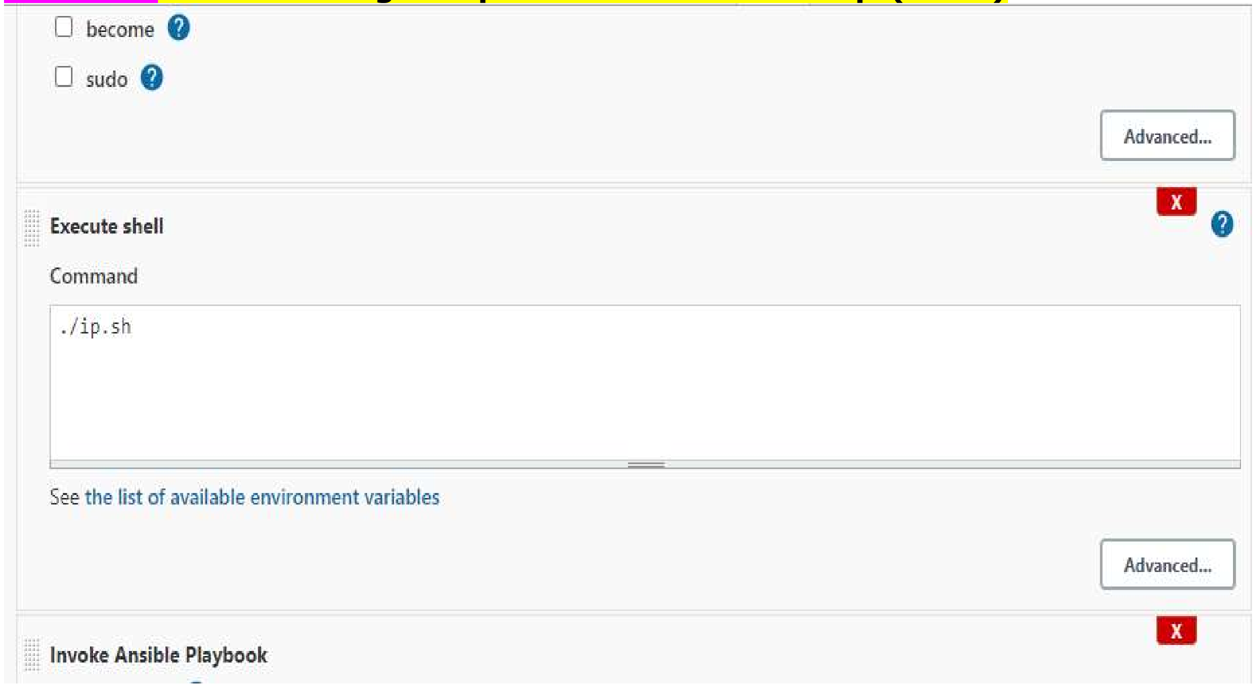
- (●) Do not specify Inventory
- ( ) File or host list
- ( ) Inline content

Host subset:

Credentials: - none - Add ▾

Vault Credentials

PIPELINE2: In Build Stage Step 2: Selected Shell Script(IP.sh)



☐ become

☐ sudo

Advanced...

Execute shell

Command

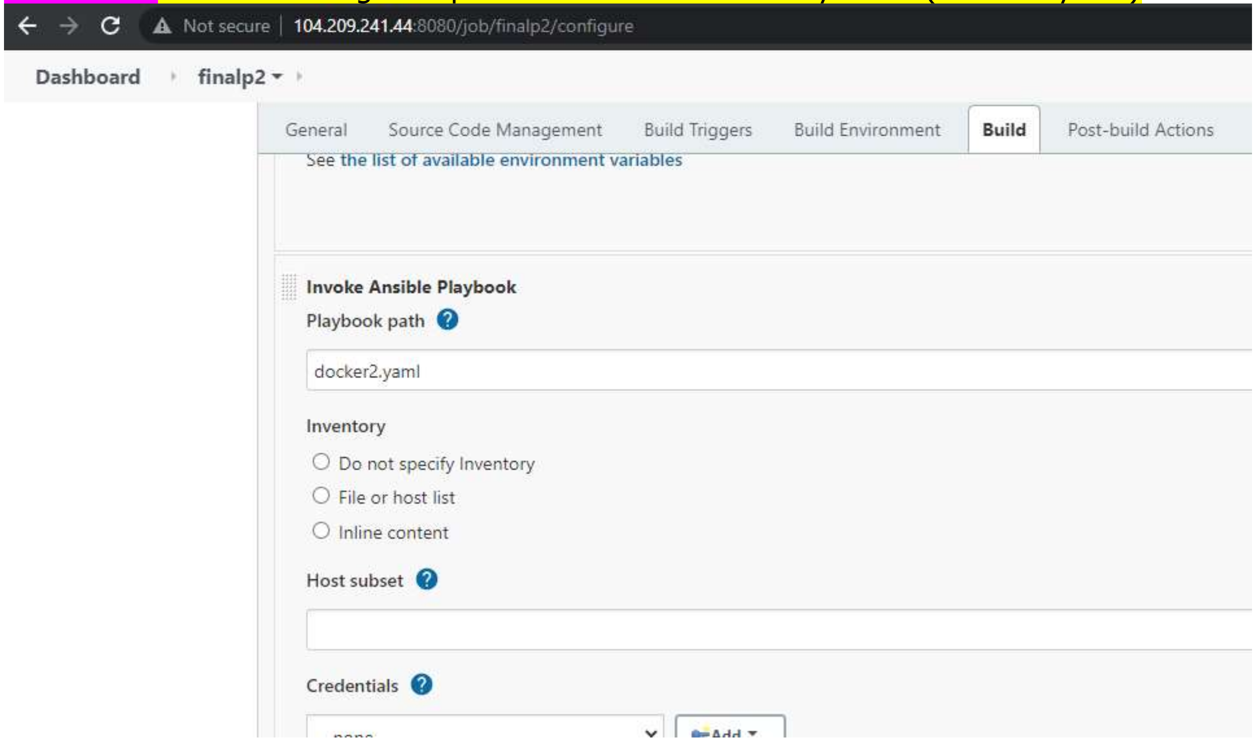
```
./ip.sh
```

See the list of available environment variables

Advanced...

Invoke Ansible Playbook

PIPELINE2: In Build Stage Step 3: Selected Ansible Playbook2(Docke2.yaml)



Not secure | 104.209.241.44:8080/job/finalp2/configure

Dashboard › finalp2 ›

General | Source Code Management | Build Triggers | Build Environment | **Build** | Post-build Actions

See the list of available environment variables

Invoke Ansible Playbook

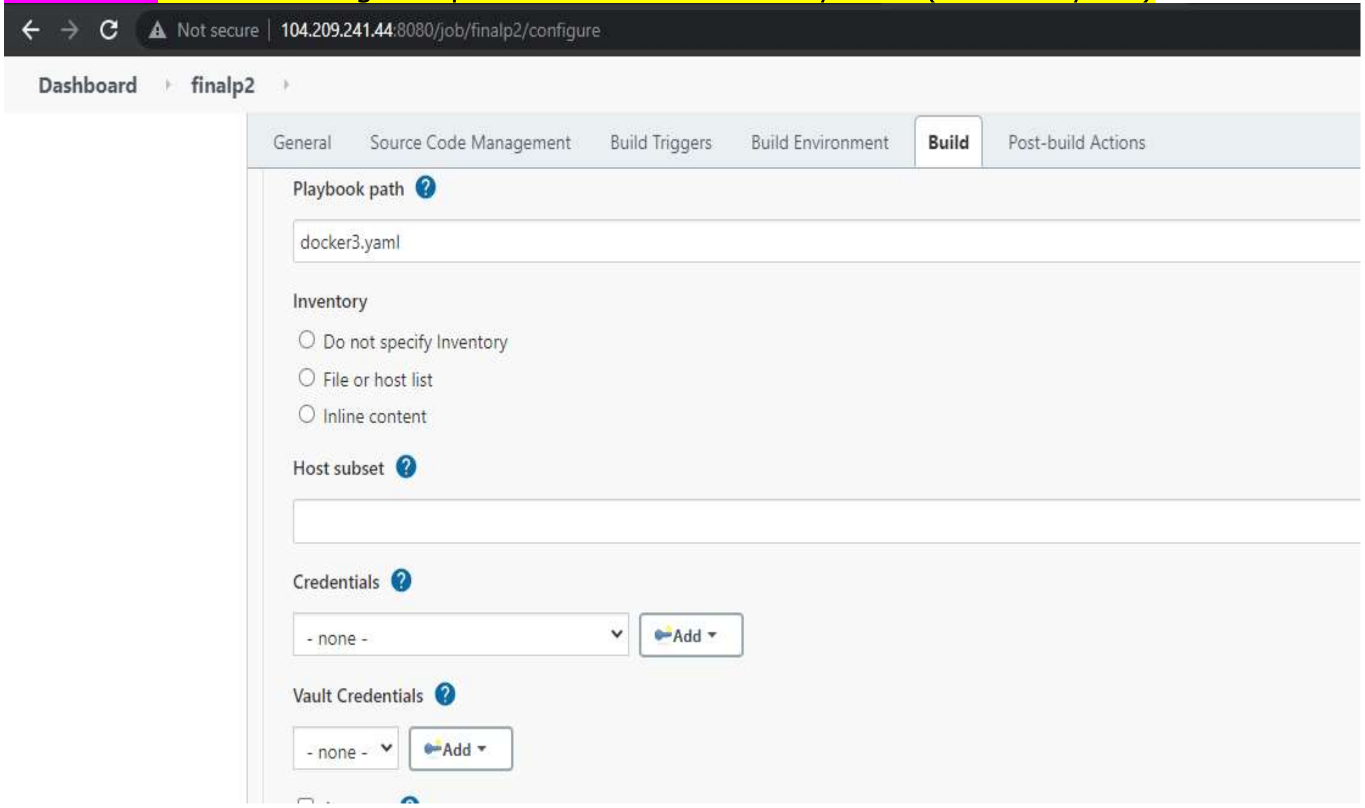
Playbook path: docker2.yaml

Inventory
- Do not specify Inventory
- File or host list
- Inline content

Host subset:

Credentials

PIPELINE2: In Build Stage Step 4: Selected Ansible Playbook3(Docker3.yaml)



Not secure | 104.209.241.44:8080/job/finalp2/configure

Dashboard › finalp2 ›

General | Source Code Management | Build Triggers | Build Environment | **Build** | Post-build Actions

Playbook path: docker3.yaml

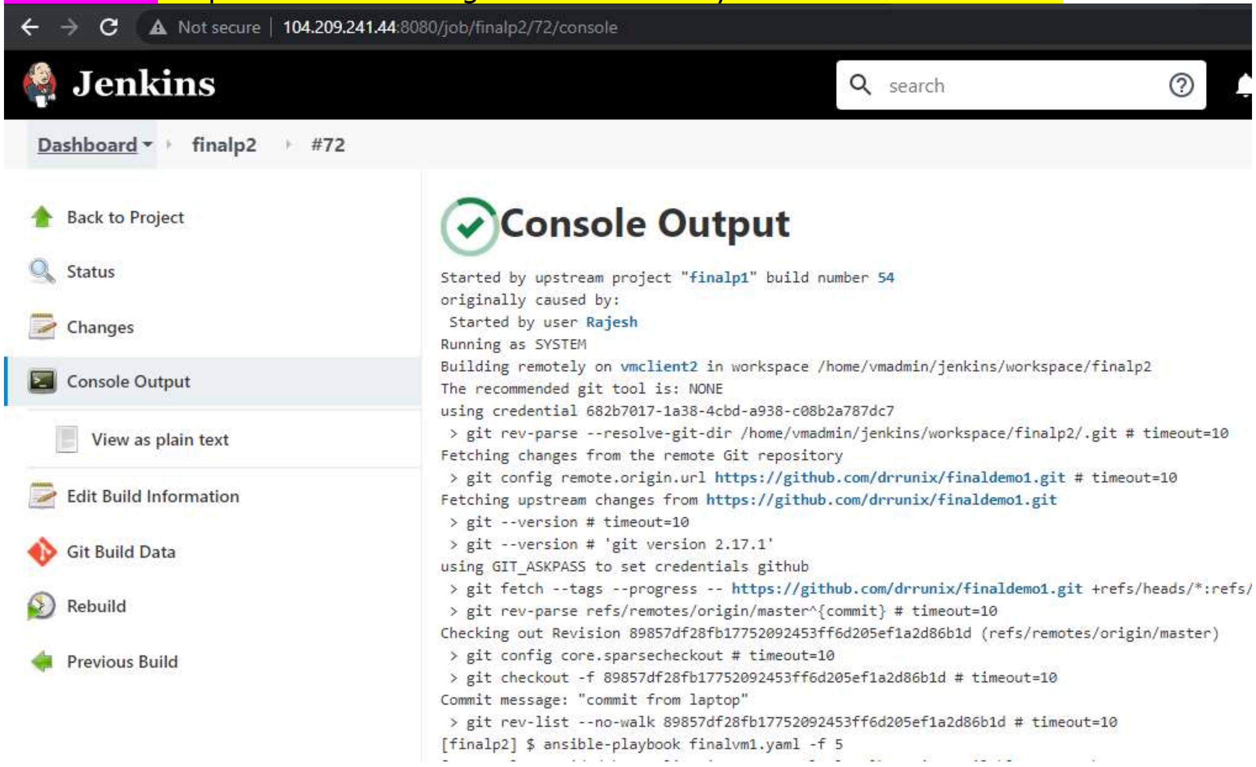
Inventory
- Do not specify Inventory
- File or host list
- Inline content

Host subset:

Credentials: - none - | Add

Vault Credentials: - none - | Add

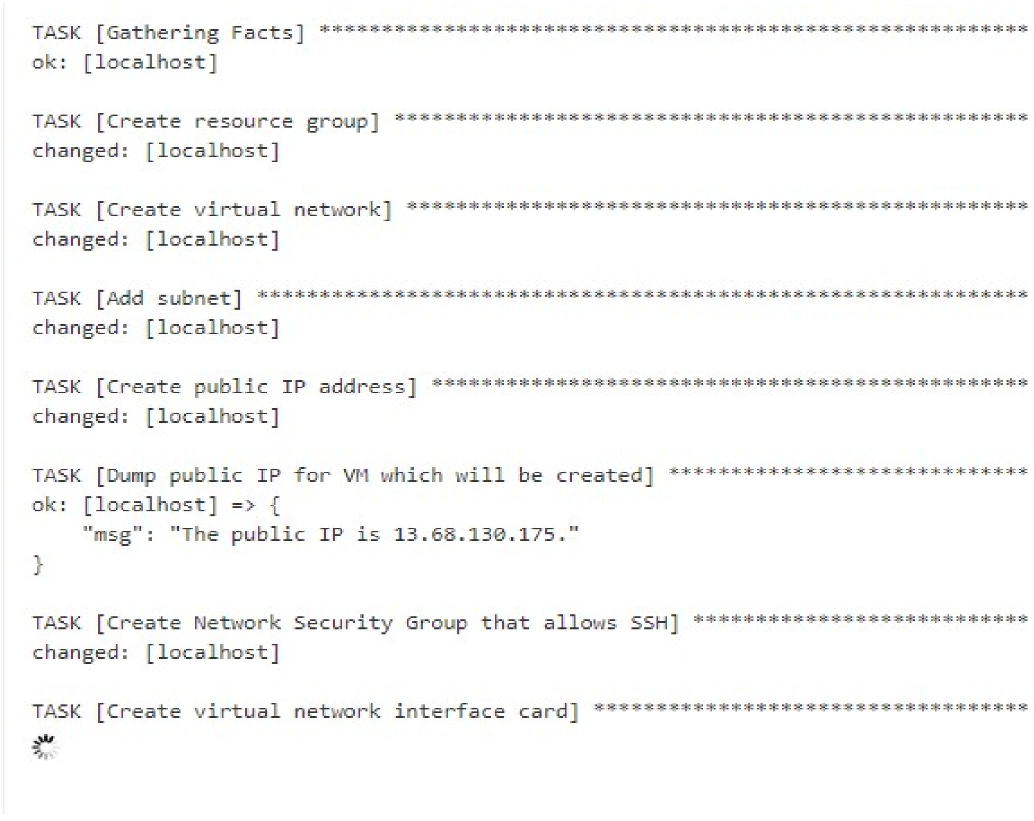
PIPELINE2: Pipeline2 was configured successfully and started the Build.



Console Output

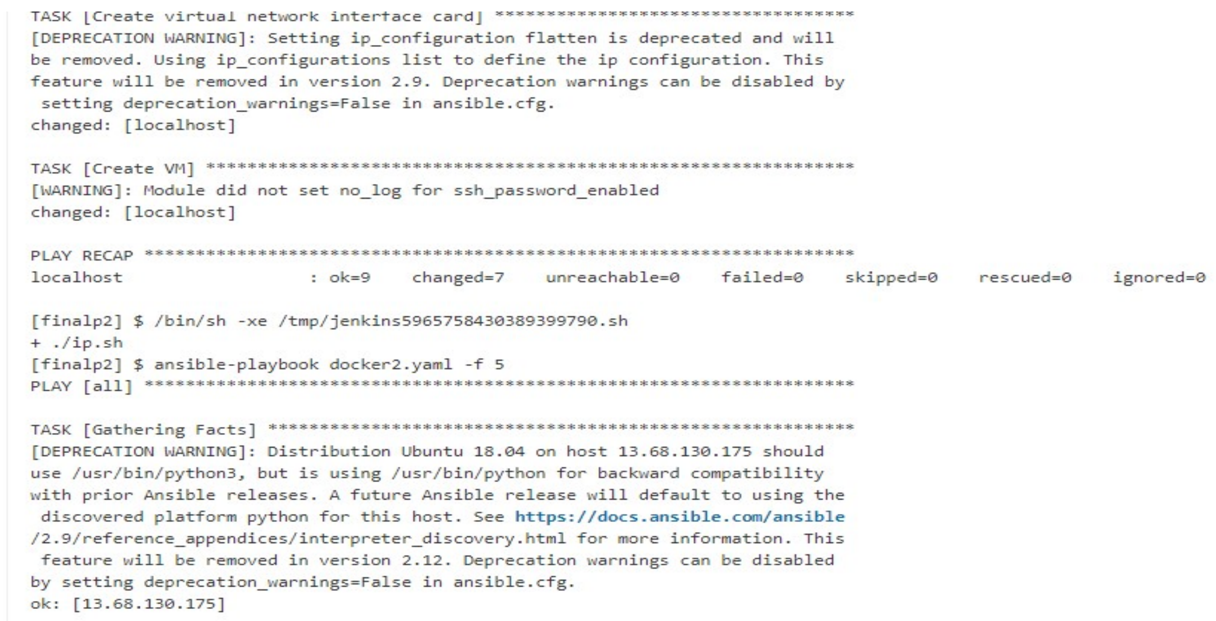
```
Started by upstream project "finalp1" build number 54
originally caused by:
 Started by user Rajesh
Running as SYSTEM
Building remotely on vmclient2 in workspace /home/vmadmin/jenkins/workspace/finalp2
The recommended git tool is: NONE
using credential 682b7017-1a38-4cbd-a938-c08b2a787dc7
 > git rev-parse --resolve-git-dir /home/vmadmin/jenkins/workspace/finalp2/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/drrunix/finaldemo1.git # timeout=10
Fetching upstream changes from https://github.com/drrunix/finaldemo1.git
 > git --version # timeout=10
 > git --version # 'git version 2.17.1'
using GIT_ASKPASS to set credentials github
 > git fetch --tags --progress -- https://github.com/drrunix/finaldemo1.git +refs/heads/*:refs/
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 89857df28fb17752092453ff6d205ef1a2d86b1d (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 89857df28fb17752092453ff6d205ef1a2d86b1d # timeout=10
Commit message: "commit from laptop"
 > git rev-list --no-walk 89857df28fb17752092453ff6d205ef1a2d86b1d # timeout=10
[finalp2] $ ansible-playbook finalvm1.yaml -f 5
```

PIPELINE2:Pipeline2 Creating VM on Azure using Playbook1**(Final1.yaml)**



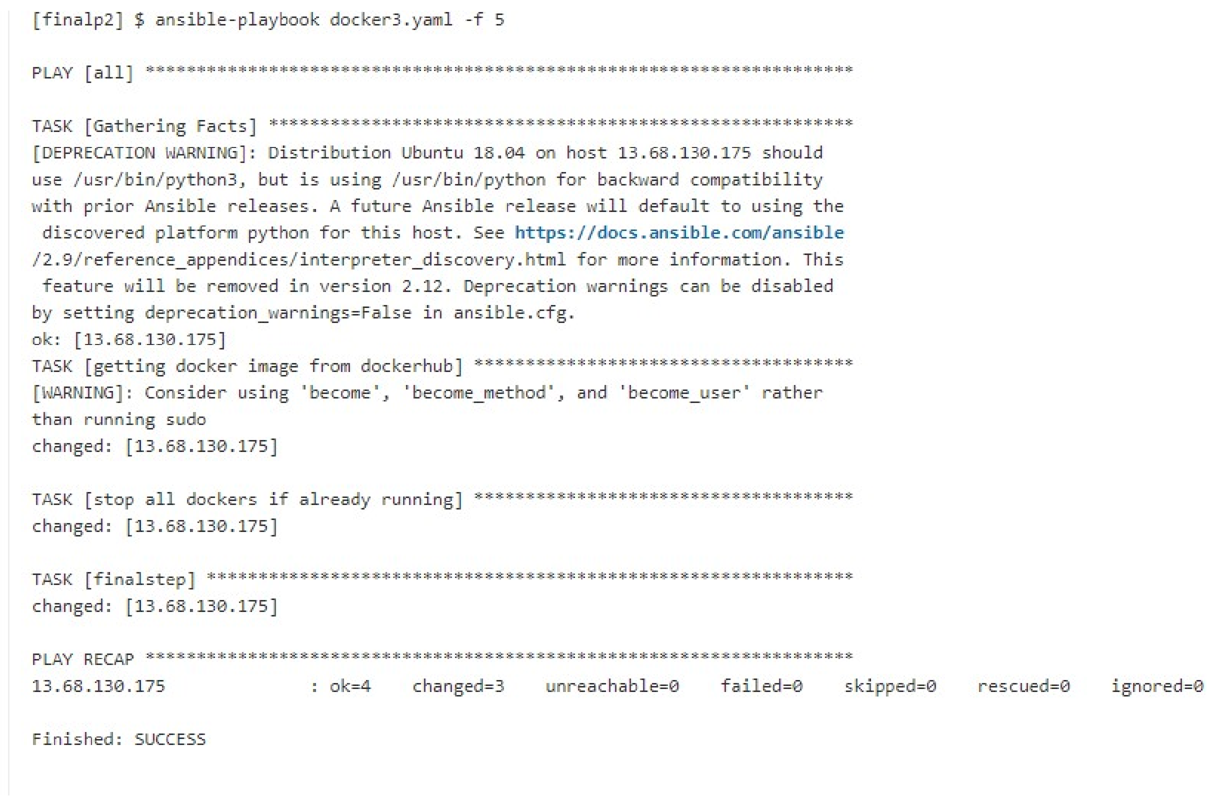
```
TASK [Gathering Facts] *********************************************************
ok: [localhost]

TASK [Create resource group] ***************************************************
changed: [localhost]

TASK [Create virtual network] **************************************************
changed: [localhost]

TASK [Add subnet] **************************************************************
changed: [localhost]

TASK [Create public IP address] ************************************************
changed: [localhost]

TASK [Dump public IP for VM which will be created] *****************************
ok: [localhost] => {
    "msg": "The public IP is 13.68.130.175."
}

TASK [Create Network Security Group that allows SSH] ***************************
changed: [localhost]

TASK [Create virtual network interface card] ***********************************
```

PIPELINE2: Pipeline created VM using ansible playbook1 and using shell script its updated ansible hosts file automatically, also using playbook2 its installing docker on newly created VM in Azure.



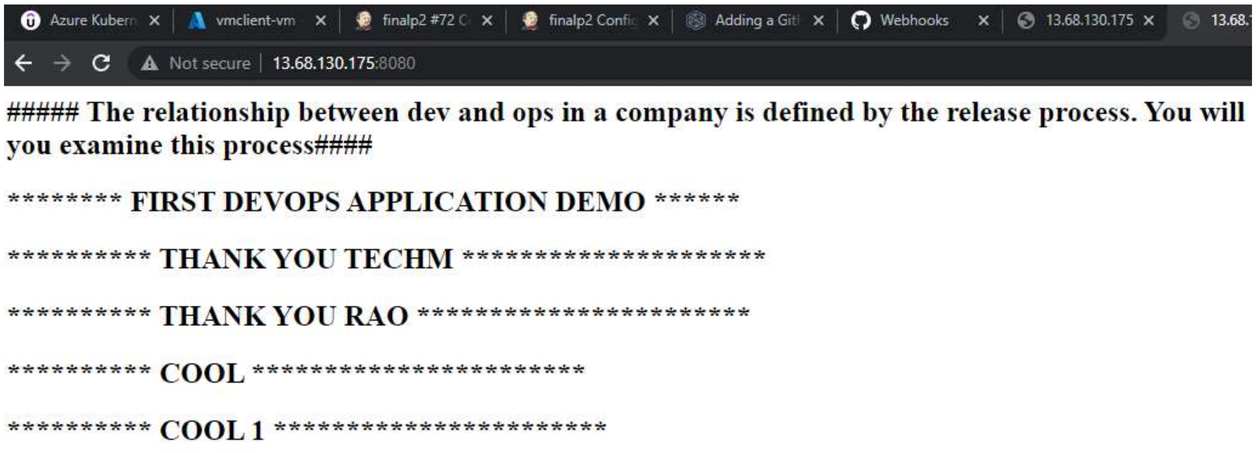
```
TASK [Create virtual network interface card] ***********************************
[DEPRECATION WARNING]: Setting ip_configuration flatten is deprecated and will
be removed. Using ip_configurations list to define the ip configuration. This
feature will be removed in version 2.9. Deprecation warnings can be disabled by
 setting deprecation_warnings=False in ansible.cfg.
changed: [localhost]

TASK [Create VM] ***************************************************************
[WARNING]: Module did not set no_log for ssh_password_enabled
changed: [localhost]

PLAY RECAP *********************************************************************
localhost                  : ok=9    changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[finalp2] $ /bin/sh -xe /tmp/jenkins5965758430389399790.sh
+ ./ip.sh
[finalp2] $ ansible-playbook docker2.yaml -f 5
PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 13.68.130.175 should
use /usr/bin/python3, but is using /usr/bin/python for backward compatibility
with prior Ansible releases. A future Ansible release will default to using the
 discovered platform python for this host. See https://docs.ansible.com/ansible
/2.9/reference_appendices/interpreter_discovery.html for more information. This
 feature will be removed in version 2.12. Deprecation warnings can be disabled
by setting deprecation_warnings=False in ansible.cfg.
ok: [13.68.130.175]
```

PIPELINE2: Finally Build is successful, Pipeline2 is able to successfully deploy the application by taking the image from Docker Hub.



```
[finalp2] $ ansible-playbook docker3.yaml -f 5

PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 13.68.130.175 should
use /usr/bin/python3, but is using /usr/bin/python for backward compatibility
with prior Ansible releases. A future Ansible release will default to using the
 discovered platform python for this host. See https://docs.ansible.com/ansible
/2.9/reference_appendices/interpreter_discovery.html for more information. This
 feature will be removed in version 2.12. Deprecation warnings can be disabled
by setting deprecation_warnings=False in ansible.cfg.
ok: [13.68.130.175]
TASK [getting docker image from dockerhub] *************************************
[WARNING]: Consider using 'become', 'become_method', and 'become_user' rather
than running sudo
changed: [13.68.130.175]

TASK [stop all dockers if already running] *************************************
changed: [13.68.130.175]

TASK [finalstep] ***************************************************************
changed: [13.68.130.175]

PLAY RECAP *********************************************************************
13.68.130.175              : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

Finished: SUCCESS
```

PIPELINE2 SUCCESS: After Build success, I got this IP 13.68.130.175
and when I try to browse the IP, I am getting the below message, which means my Pipeline is successful.



The relationship between dev and ops in a company is defined by the release process. You will you examine this process####

******** FIRST DEVOPS APPLICATION DEMO ******

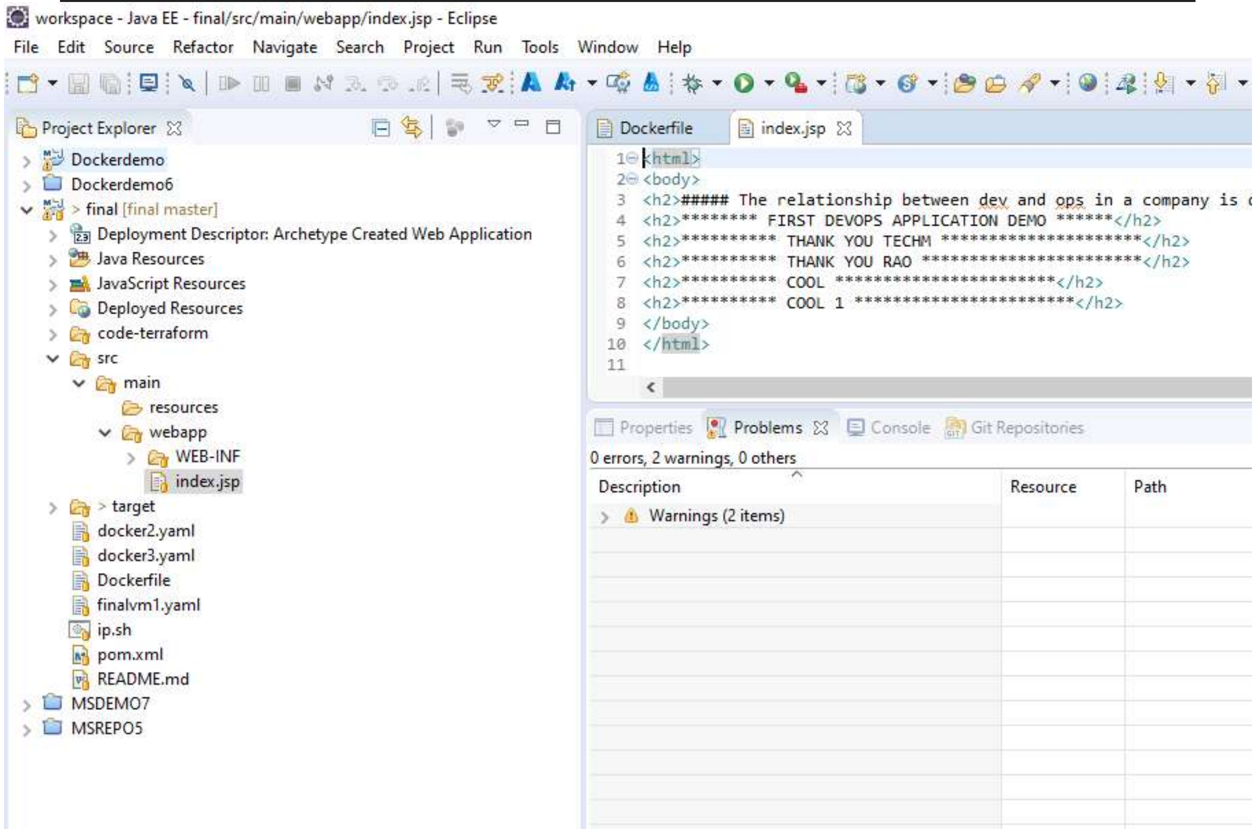
********** THANK YOU TECHM *********************

********** THANK YOU RAO **********************

********** COOL **********************

********** COOL 1 **********************

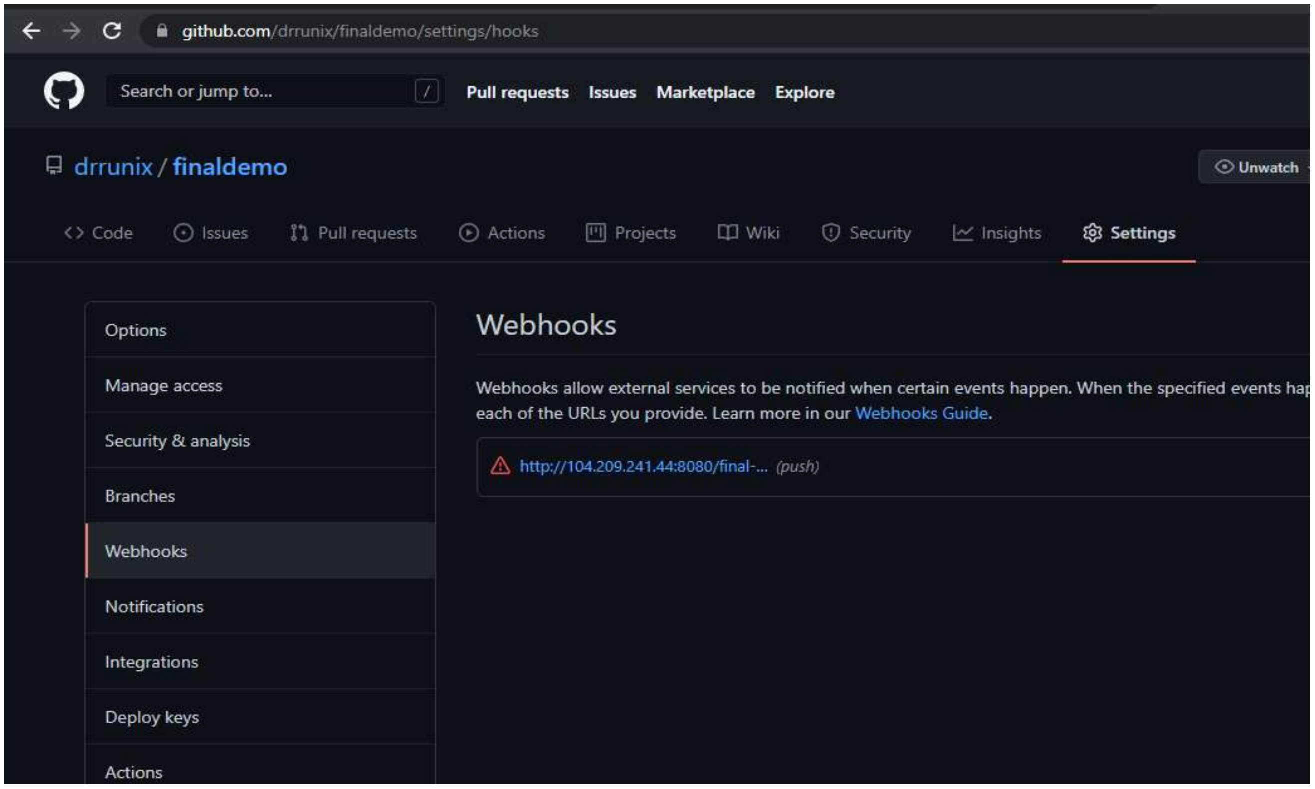
Below is the code on my local, it means my Project is a success



```
<html>
<body>
<h2>##### The relationship between dev and ops in a company is
<h2>******** FIRST DEVOPS APPLICATION DEMO ******</h2>
<h2>********** THANK YOU TECHM *********************</h2>
<h2>********** THANK YOU RAO **********************</h2>
<h2>********** COOL **********************</h2>
<h2>********** COOL 1 **********************</h2>
</body>
</html>
```

0 errors, 2 warnings, 0 others

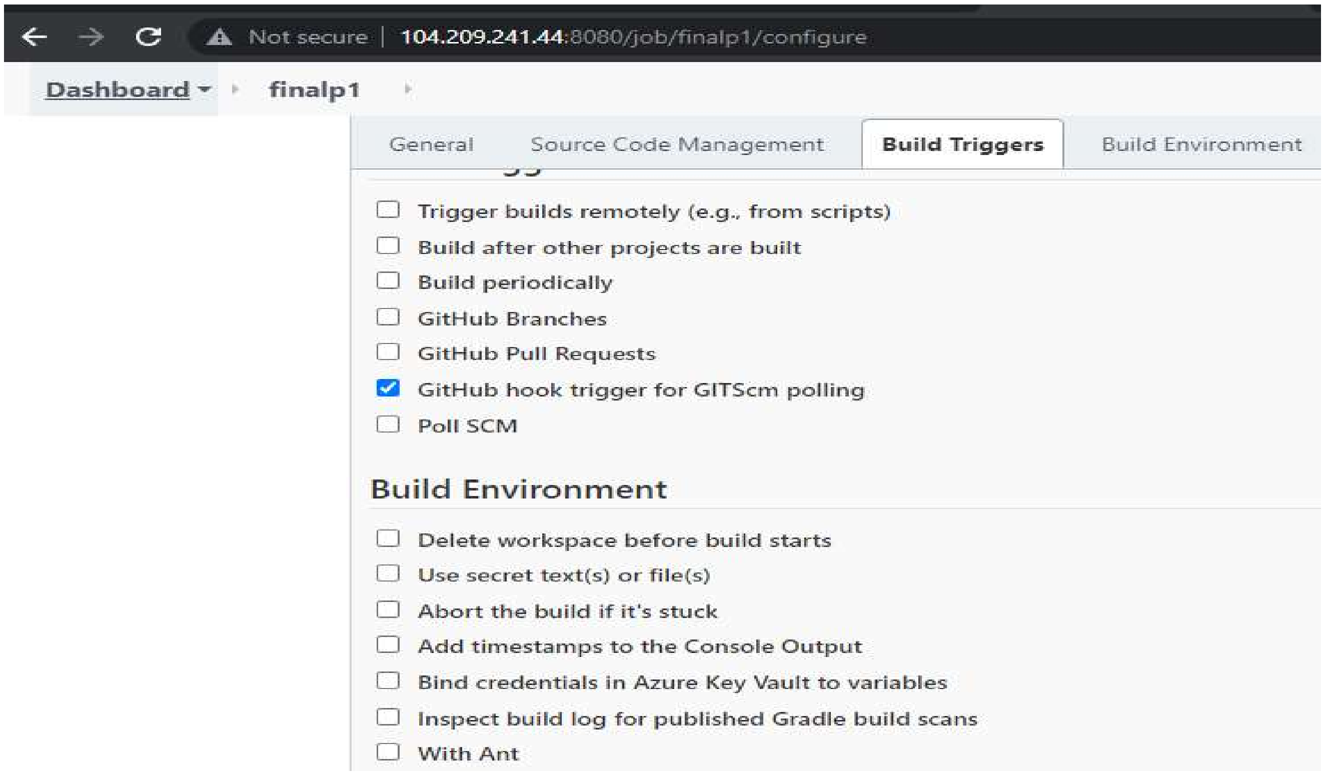
Final step: Configured the webhook on the GitHub repository, just to ensure that every time a developer commits a code to GitHub, our build will be triggered.


github.com/drrunix/finaldemo/settings/hooks

Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events hap each of the URLs you provide. Learn more in our Webhooks Guide.

http://104.209.241.44:8080/final-... (push)

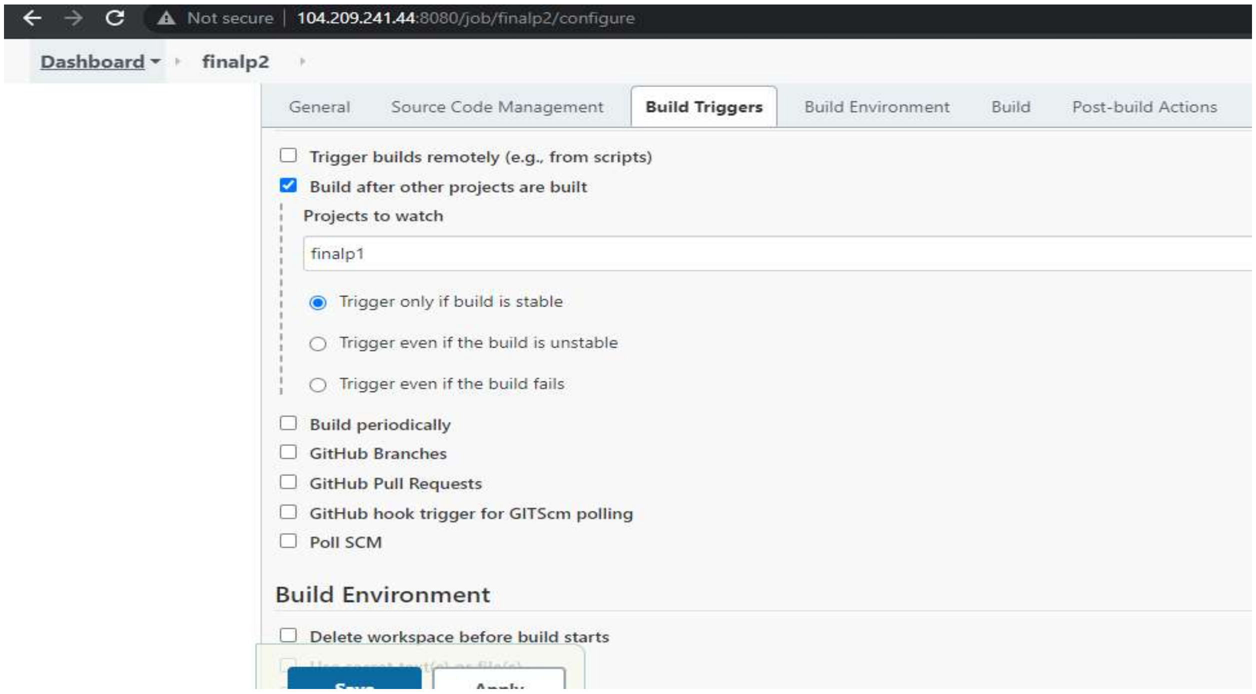

Not secure | 104.209.241.44:8080/job/finalp1/configure

- [ ] Trigger builds remotely (e.g., from scripts)
- [ ] Build after other projects are built
- [ ] Build periodically
- [ ] GitHub Branches
- [ ] GitHub Pull Requests
- [x] GitHub hook trigger for GITScm polling
- [ ] Poll SCM

Build Environment

- [ ] Delete workspace before build starts
- [ ] Use secret text(s) or file(s)
- [ ] Abort the build if it's stuck
- [ ] Add timestamps to the Console Output
- [ ] Bind credentials in Azure Key Vault to variables
- [ ] Inspect build log for published Gradle build scans
- [ ] With Ant

Additional step: Also Enabled Build after other projects are built option in pipeline2. so, once pipeline1 is successful and the second pipeline will get triggered.


104.209.241.44:8080/job/finalp2/configure

Dashboard › finalp2

☐ Trigger builds remotely (e.g., from scripts)
☑ Build after other projects are built
Projects to watch: finalp1
● Trigger only if build is stable
○ Trigger even if the build is unstable
○ Trigger even if the build fails
☐ Build periodically
☐ GitHub Branches
☐ GitHub Pull Requests
☐ GitHub hook trigger for GITScm polling
☐ Poll SCM

Build Environment

☐ Delete workspace before build starts

FINAL STEP – TESTING CI/CD

Now I am going to update the code on a local system and push code to GitHub. Once I push the code, Pipeline1 will be triggered automatically, and after that Pipeline2 will be executed automatically.

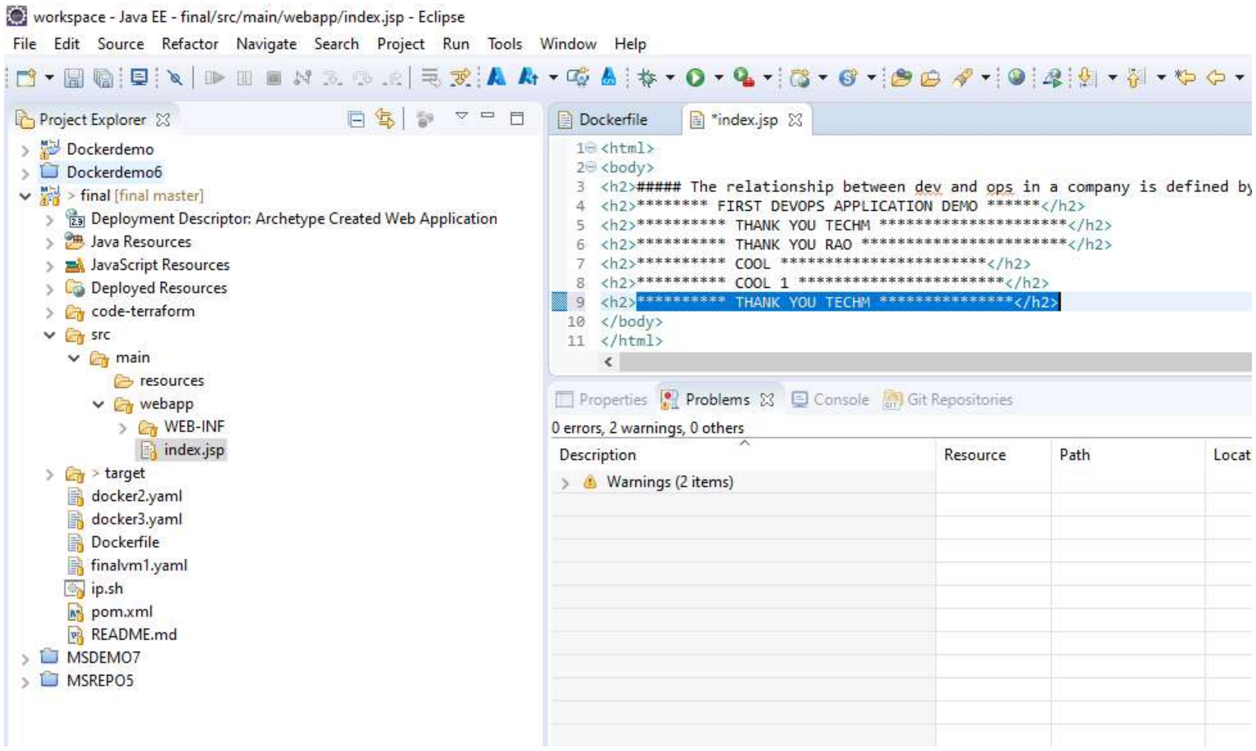
Below are the pipeline tasks.

Pipeline1 - it will take app code from GitHub, build a docker image and will push to docker hub, and it will trigger pipeline2.

Pipeline2 - it will deploy VM on Azure and install docker on that VM. Also, using Ansible playbooks, it will get an image from the docker hub and will deploy on that Azure VM using the playbook.

This is a continuous integration and deployment.

Updated code on my local



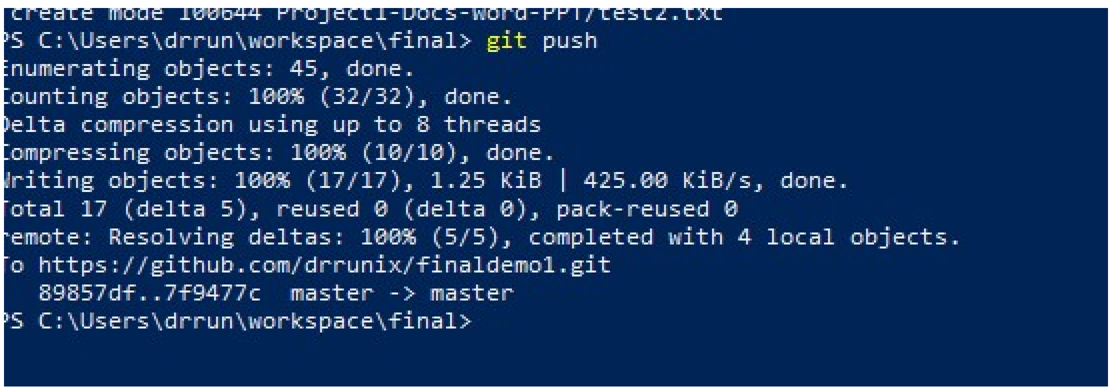
workspace - Java EE - final/src/main/webapp/index.jsp - Eclipse

```
<html>
<body>
<h2>##### The relationship between dev and ops in a company is defined by
<h2>******** FIRST DEVOPS APPLICATION DEMO ******</h2>
<h2>********** THANK YOU TECHM *********************</h2>
<h2>********** THANK YOU RAO ***********************</h2>
<h2>********** COOL ***********************</h2>
<h2>********** COOL 1 ***********************</h2>
<h2>********** THANK YOU TECHM ***************</h2>
</body>
</html>
```

0 errors, 2 warnings, 0 others

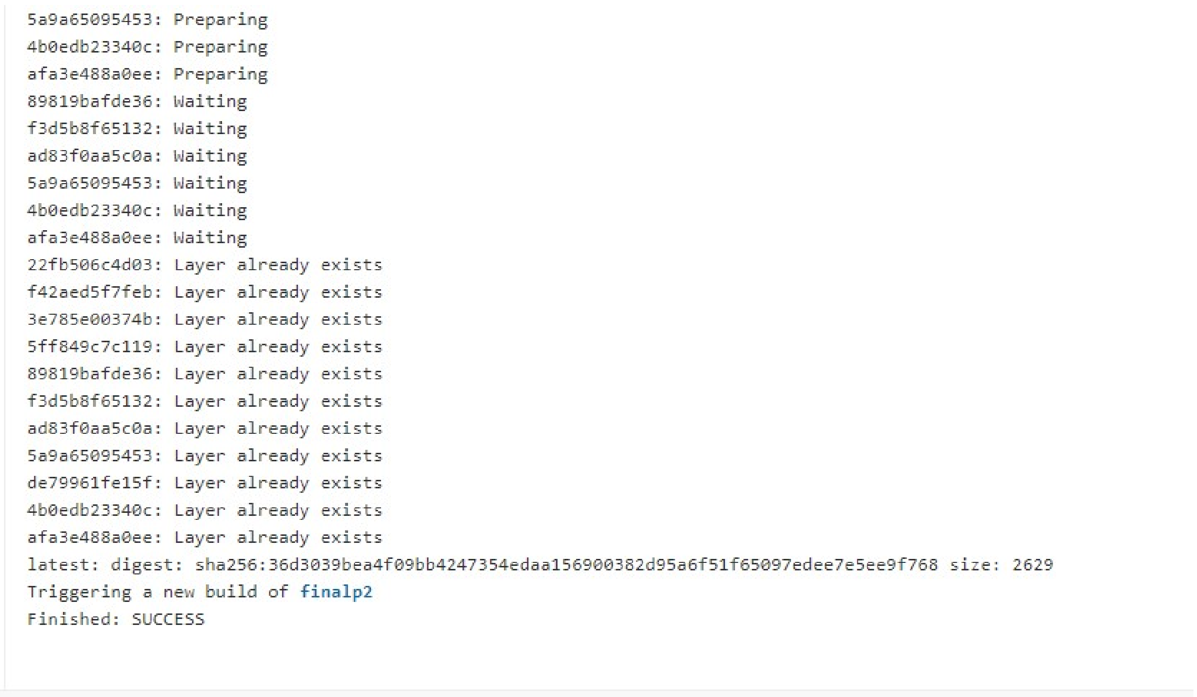
Description	Resource	Path	Locat
Warnings (2 items)			

Pushed the code



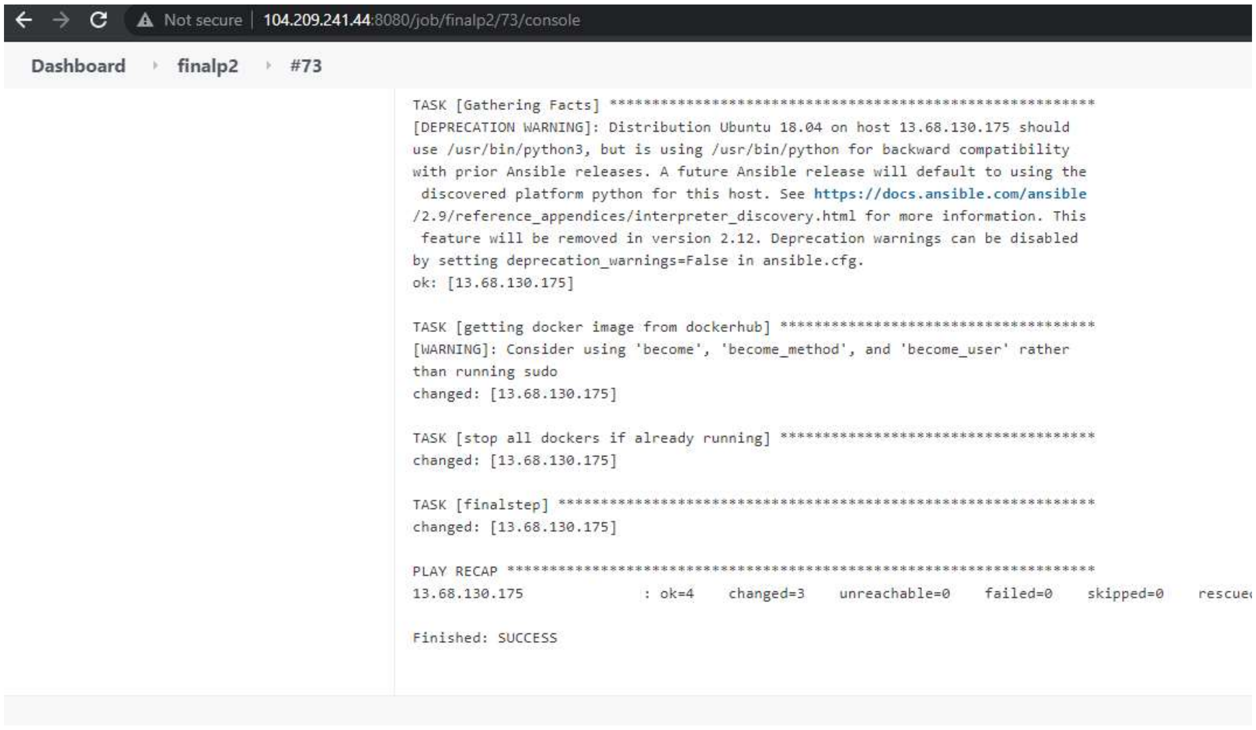
```
create mode 100644 Project1-Docs-Word-PPT/test2.txt
PS C:\Users\drrun\workspace\final> git push
Enumerating objects: 45, done.
Counting objects: 100% (32/32), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (17/17), 1.25 KiB | 425.00 KiB/s, done.
Total 17 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 4 local objects.
To https://github.com/drrunix/finaldemo1.git
   89857df..7f9477c  master -> master
PS C:\Users\drrun\workspace\final>
```

Pipeline1 - it will take app code from GitHub, build a docker image and will push to docker hub, and it will trigger pipeline2.

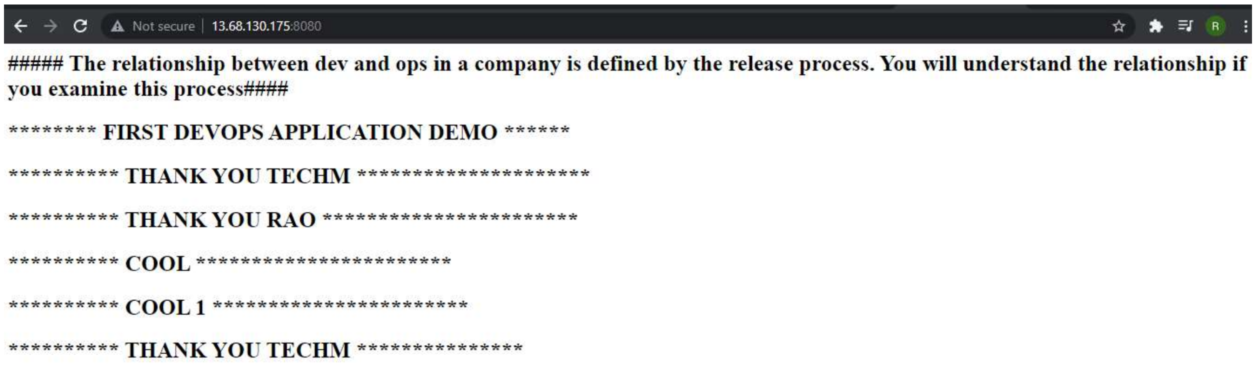


```
5a9a65095453: Preparing
4b0edb23340c: Preparing
afa3e488a0ee: Preparing
89819bafde36: Waiting
f3d5b8f65132: Waiting
ad83f0aa5c0a: Waiting
5a9a65095453: Waiting
4b0edb23340c: Waiting
afa3e488a0ee: Waiting
22fb506c4d03: Layer already exists
f42aed5f7feb: Layer already exists
3e785e00374b: Layer already exists
5ff849c7c119: Layer already exists
89819bafde36: Layer already exists
f3d5b8f65132: Layer already exists
ad83f0aa5c0a: Layer already exists
5a9a65095453: Layer already exists
de79961fe15f: Layer already exists
4b0edb23340c: Layer already exists
afa3e488a0ee: Layer already exists
latest: digest: sha256:36d3039bea4f09bb4247354edaa156900382d95a6f51f65097edee7e5ee9f768 size: 2629
Triggering a new build of finalp2
Finished: SUCCESS
```

Pipeline2 - it will deploy VM on Azure and install docker on that VM. Also, using Ansible playbooks, it will get an image from the docker hub and will deploy on that Azure VM using the playbook.



```
TASK [Gathering Facts] *********************************************************
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 13.68.130.175 should
use /usr/bin/python3, but is using /usr/bin/python for backward compatibility
with prior Ansible releases. A future Ansible release will default to using the
 discovered platform python for this host. See https://docs.ansible.com/ansible
/2.9/reference_appendices/interpreter_discovery.html for more information. This
 feature will be removed in version 2.12. Deprecation warnings can be disabled
by setting deprecation_warnings=False in ansible.cfg.
ok: [13.68.130.175]

TASK [getting docker image from dockerhub] *************************************
[WARNING]: Consider using 'become', 'become_method', and 'become_user' rather
than running sudo
changed: [13.68.130.175]

TASK [stop all dockers if already running] *************************************
changed: [13.68.130.175]

TASK [finalstep] ***************************************************************
changed: [13.68.130.175]

PLAY RECAP *********************************************************************
13.68.130.175              : ok=4    changed=3    unreachable=0    failed=0    skipped=0    rescue

Finished: SUCCESS
```



```
##### The relationship between dev and ops in a company is defined by the release process. You will understand the relationship if you examine this process####

********* FIRST DEVOPS APPLICATION DEMO ******

*********** THANK YOU TECHM ***********************

*********** THANK YOU RAO *************************

*********** COOL *************************

*********** COOL 1 *************************

*********** THANK YOU TECHM *****************
```

This is a continuous integration and deployment.

############### THE END ##################