**Project 1 | CFS-DevOps Engineer | SA00570532 | Aashiq Ahamed SP**

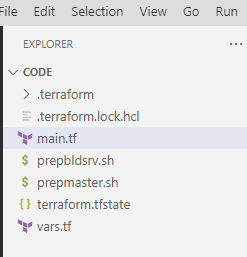
Task1:

Terraform script to create Jenkins-master and build-server in Azure along with provisioner block.

Used Visual Studio code with terraform and azure terraform extenions.

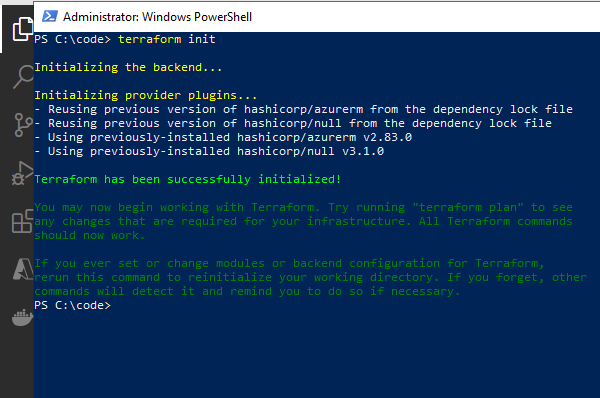
1. Main.tf – to Create Resource Group, Virtual Network, Subnet, Network security Group, Nic, public IP and Virtual machines
2. Vars.tf – holds variables for above main.tf resources.
3. Prepmaster.sh – script file to install java and Jenkins in Jenkins-master vm using Terraform
4. Prepbldsrv.sh – script file to install java, maven, Docker, Ansible and Azure CLI on buildserver.

All above files are available in <https://github.com/aashiqAhamedSP/project1_570532.git>



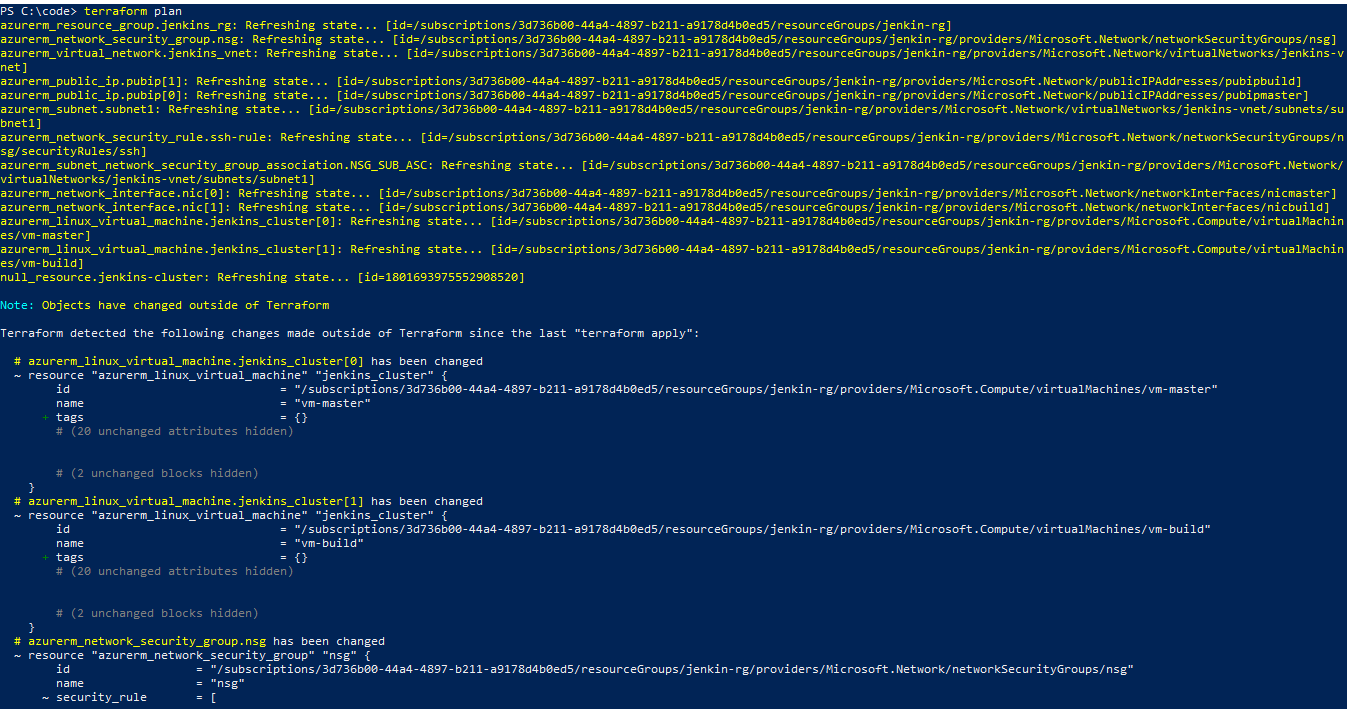
Task 2:

Running the terraform init after installing terraform in the windows machine.



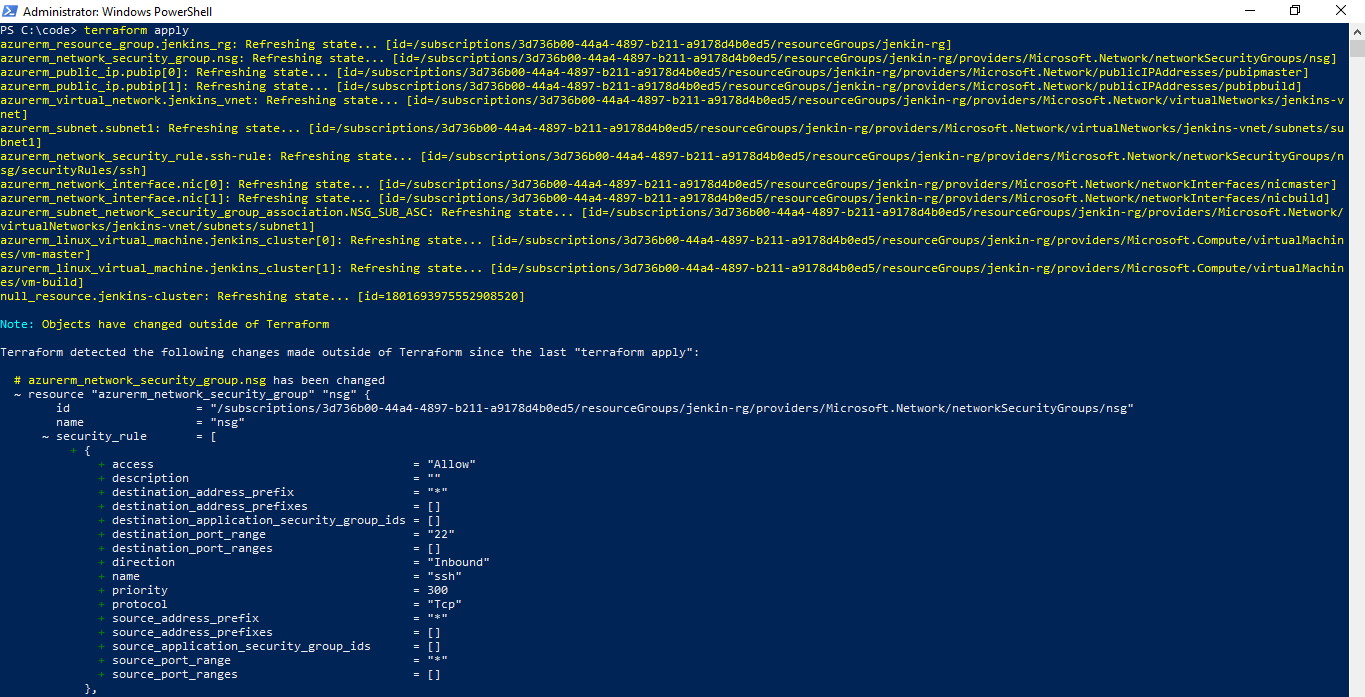
Task3:

Terraform plan

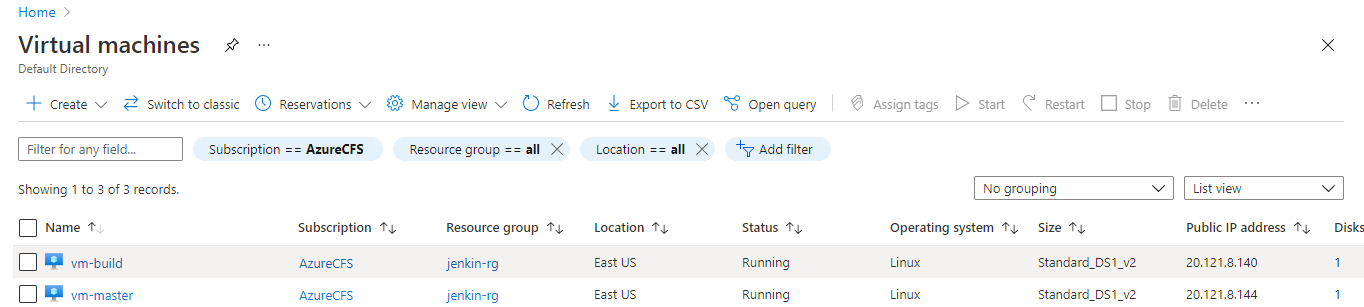


Task 4:

Terraform apply

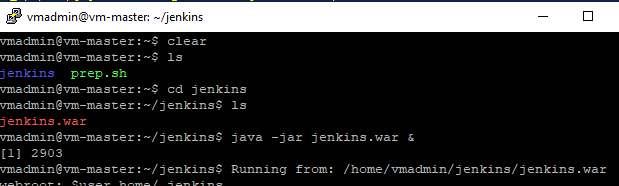


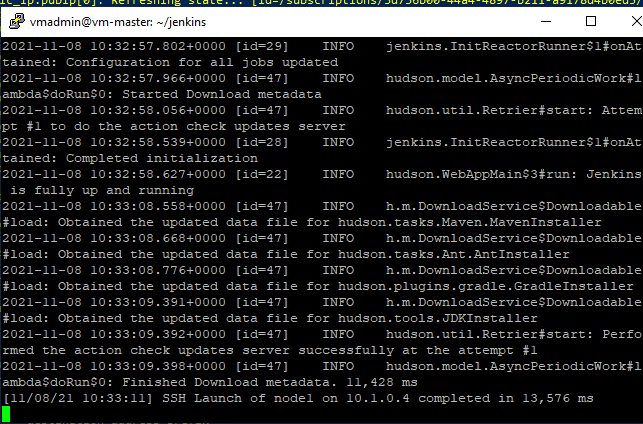
Task 5:



Task 6:

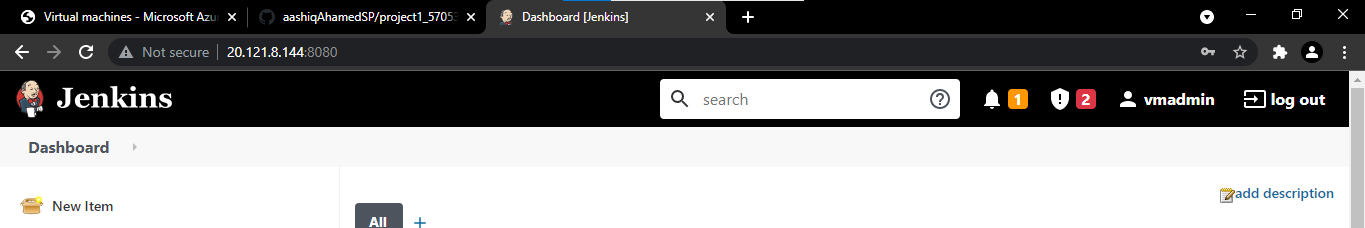
Log into vm-master and start Jenkins

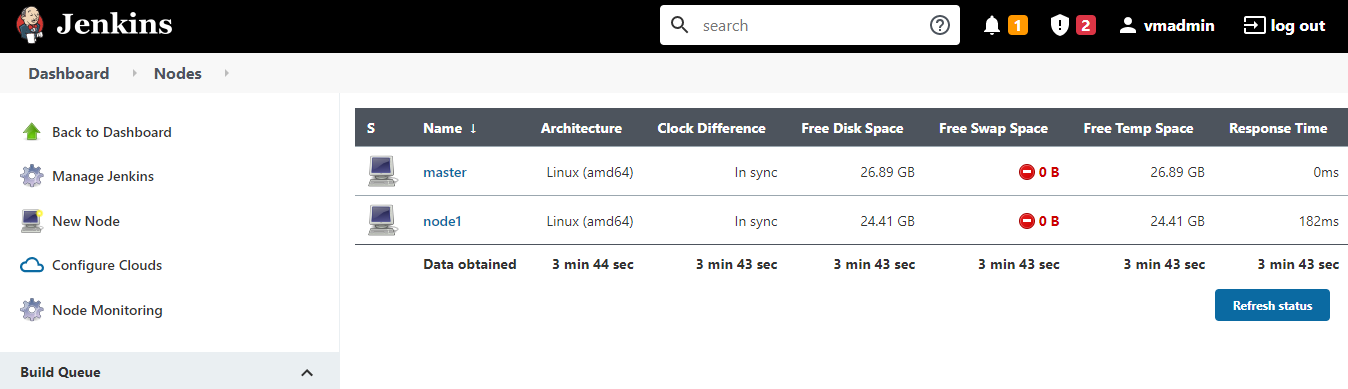




Task 7:

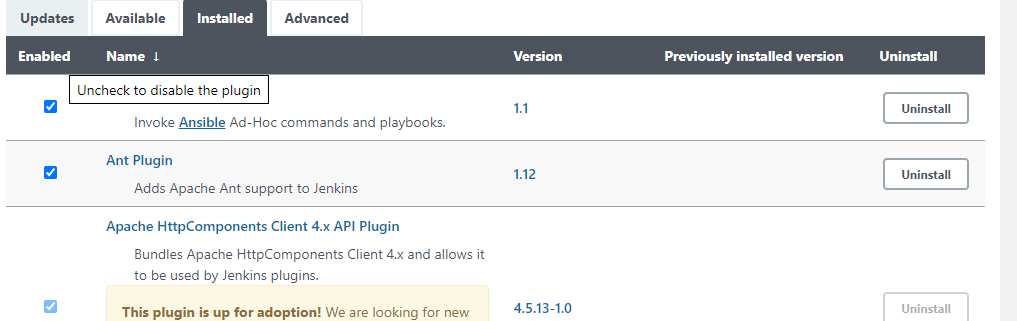
Log into Jenkins UI and configure build-server as node 1





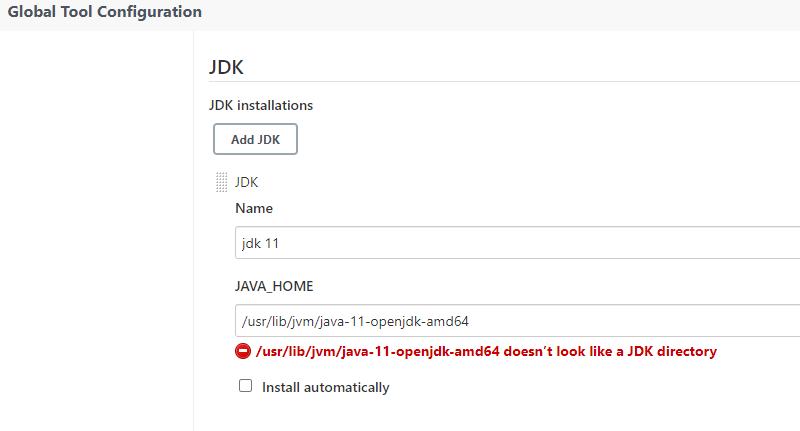
Task 8:

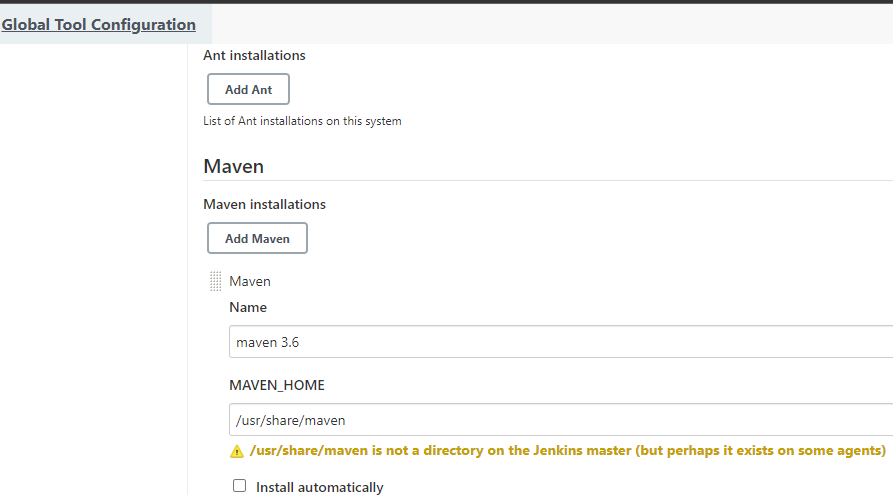
Install all necessary plugins Ansible, Github, maven, Docker build and plublish



Task 9:

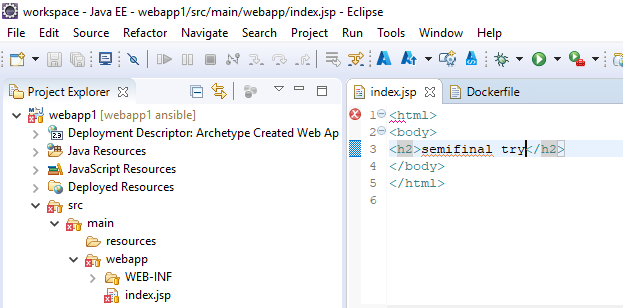
Manager global tools java as JDK 11 and Maven 3,6

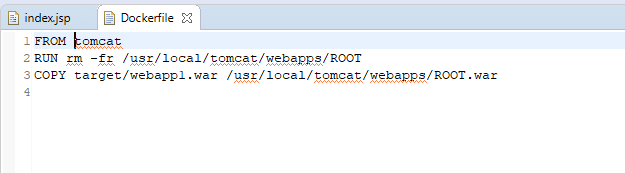




Task 10:

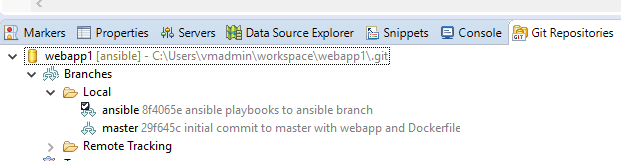
Create Maven project using Eclipse with custom index.jsp and Dockerfile





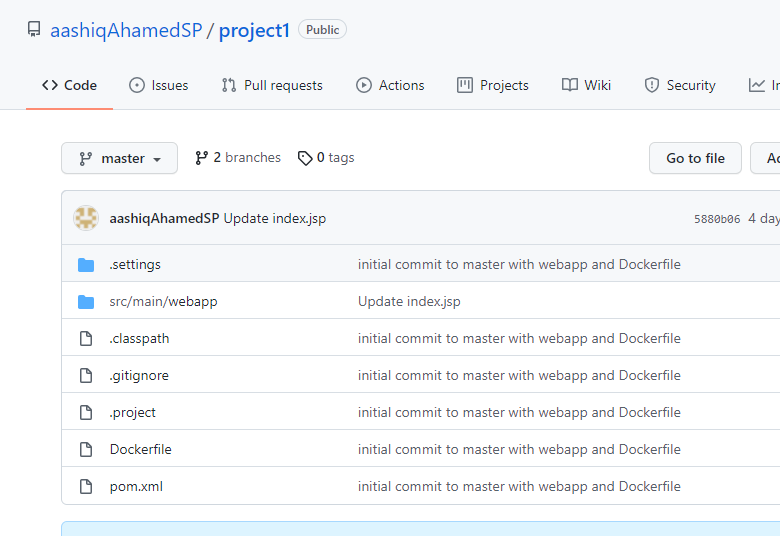
Task 11:

Create a local repo for maven project



Task 12:

Push the code to github repo

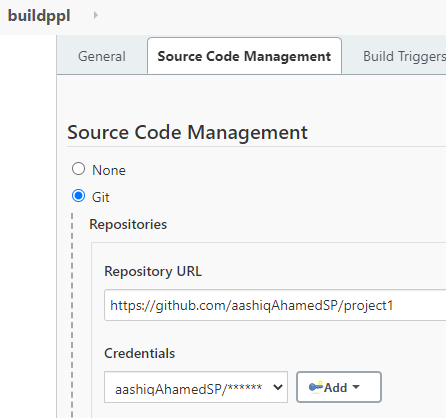


Task 13:

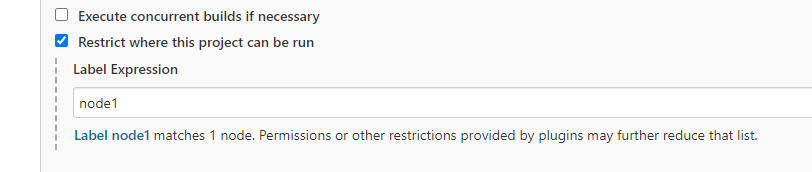
Create pipeline 1 in Jenkins

Buildppl:

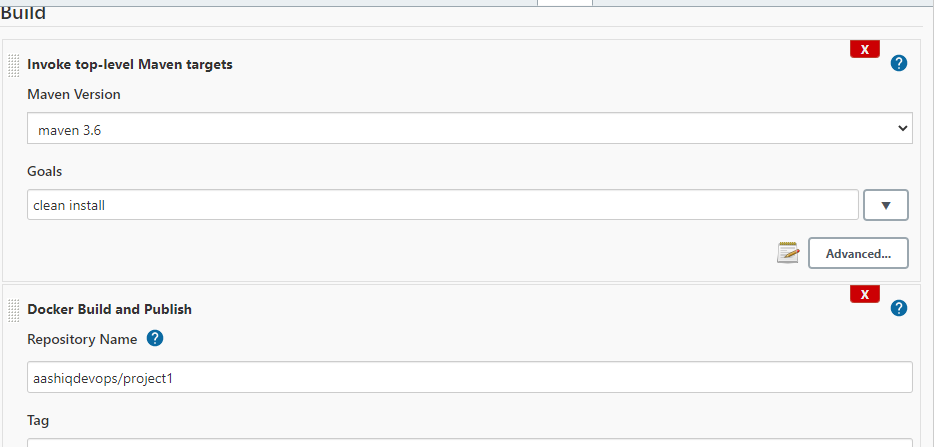
1. Get code from <https://github.com/aashiqAhamedSP/project1.git>



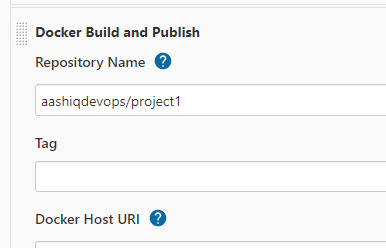
1. Restrict the build to Node1(Jenkins build-server)



1. Build webapp1.war using pom.xml by invoking top-level maven targets

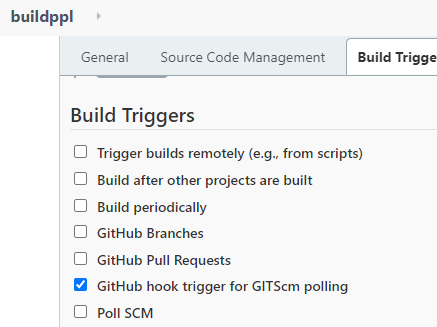


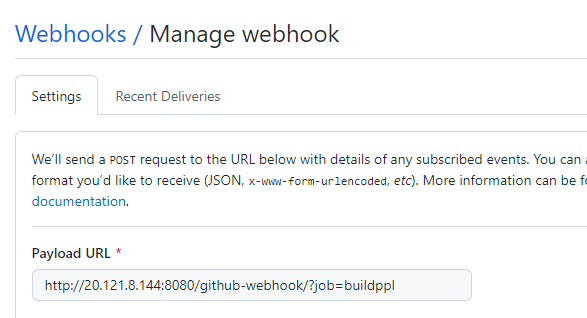
1. Build docker image and push it to Dockerhub



Task 14:

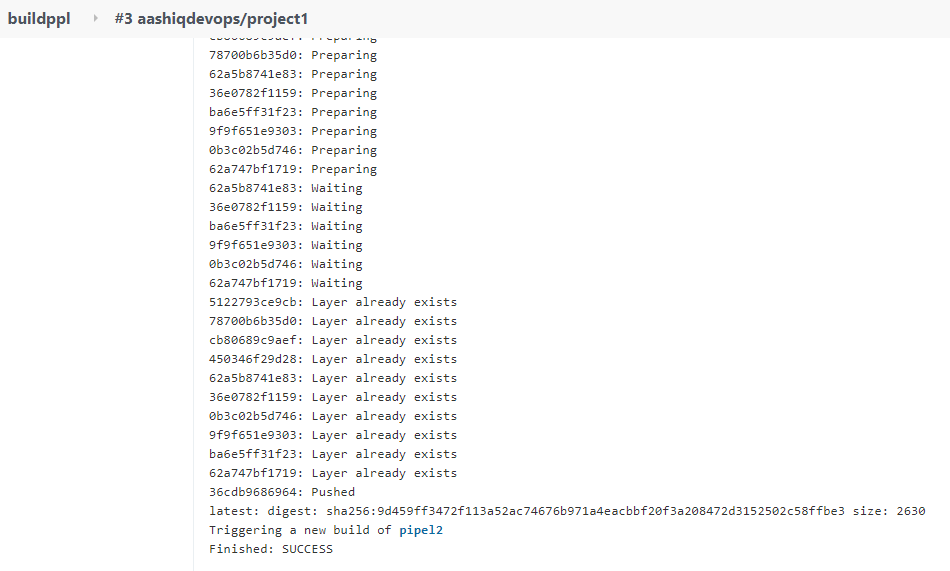
Configure webhook for triggering buildppl from guthub

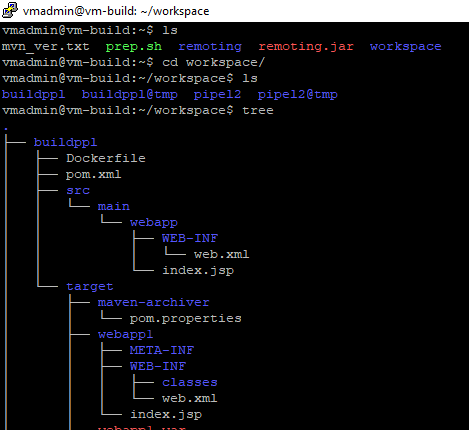




Task 15:

Build now and check





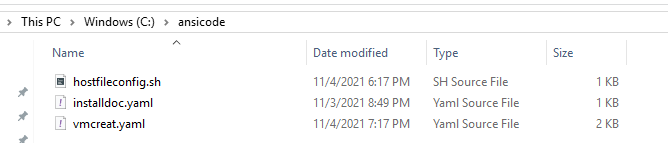
Task 16:

Create Ansible Playbook “vmcreat.yaml”

Vmcreat.yaml – this playbook will create Azure Linux VM, Public IP, new subnet in Jenkins Vnet, Nic and Network Security Group.

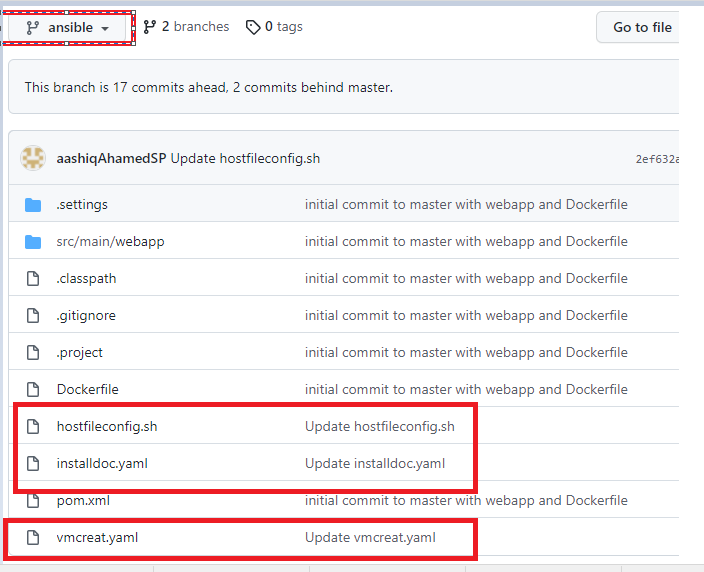
Installdoc.yaml – this playbook will install docker and all its dependency packages in newly created VM

Hostfileconfig.sh – this script file will configure ansible hosts and SSH settings into newly created VM.



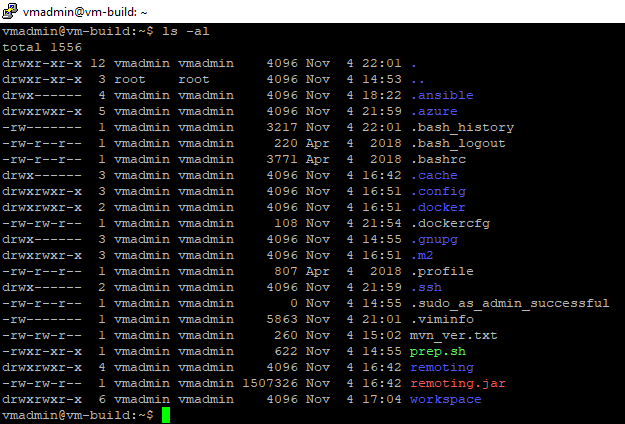
Task 17:

Create new branch “ansible ” in the github repo where maven project exists and push the playbooks and script file into Ansible branch

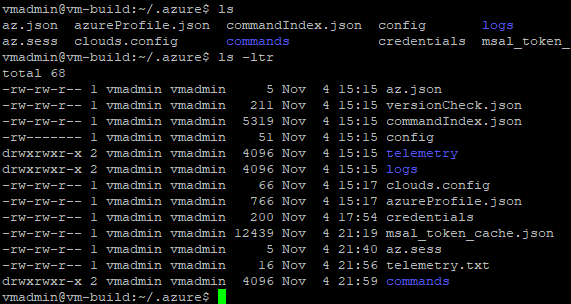


Task 18:

Using PIP installed azure module for ansible in build server

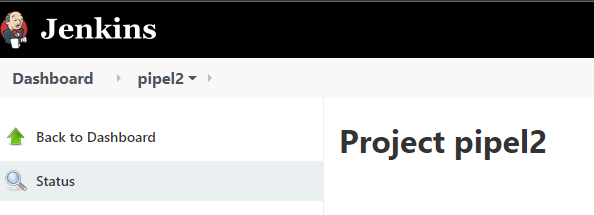


Configure a credentials file in .azure



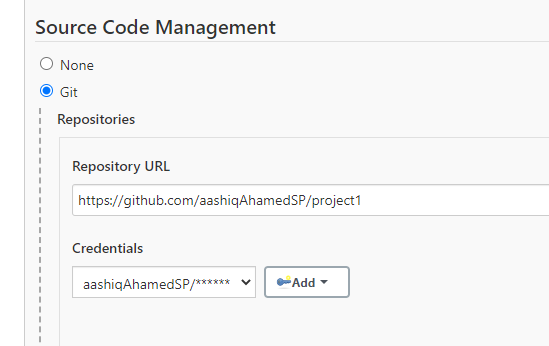
Task 19:

Create pipeline 2 in Jenkins to get these playbooks and script file from github Ansible branch and pull docker image pushed in pipleline 1 from docker hub and run container on newly created VM



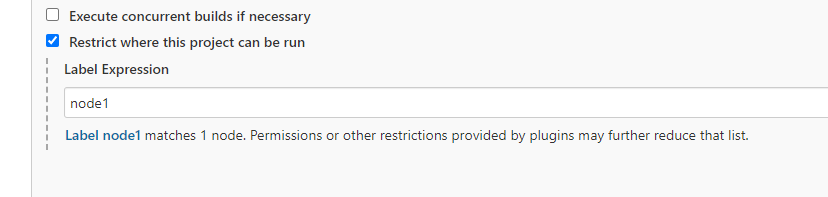
Task 20:

Configure github as SCM to pull Ansible code



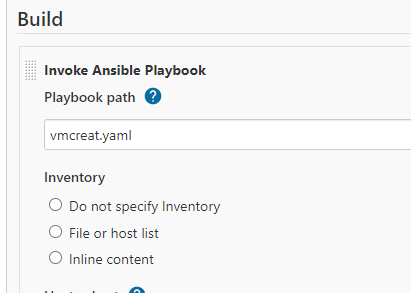
Task 21:

Restrict the build to node 1 build-server



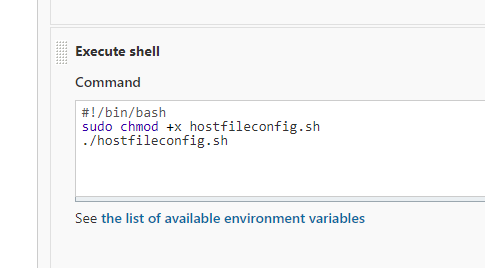
Task 22:

Configure a build “invoke ansible playbook” step to vmcreat.yaml



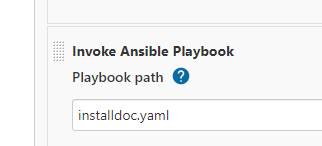
Task 23:

Configure another build step “execute shell” to run the script file



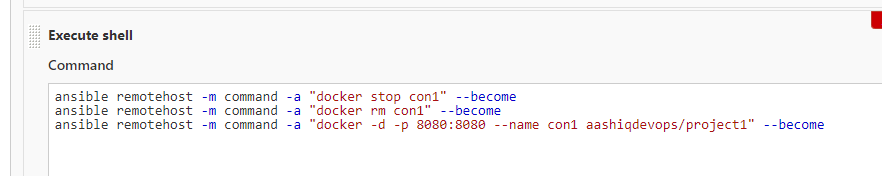
Task 24:

Configure another build “invoke ansible playbook” to run the installdoc.yaml playbook



Task 25:

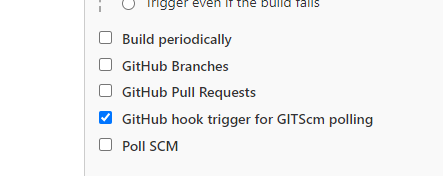
Configure a build step “execute shell” to pull the Docker image form Docker hub and run a container names con1 and also delete any containers of same name already available.



No specific tag has been used in pipeline1 “buildppl” while building and pushing Docker image so that latest tag can be used by default this helps in defaulting the image tag name in pipeline2 also.

Task 26:

Configure webhook build trigger for pipeline 2 also

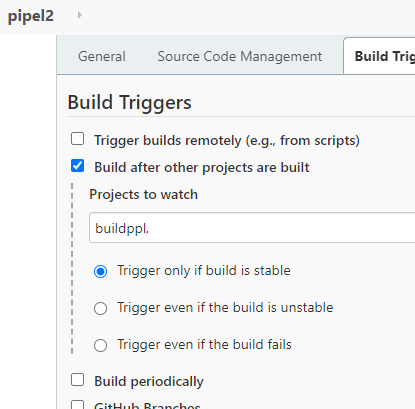


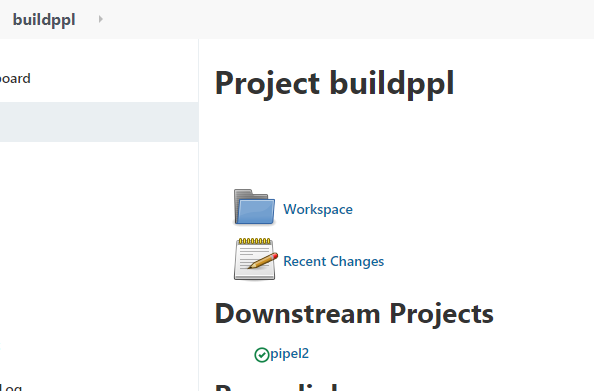
When a commit action performed with changes in master branch then pipeline1 “buildppl” will be triggered

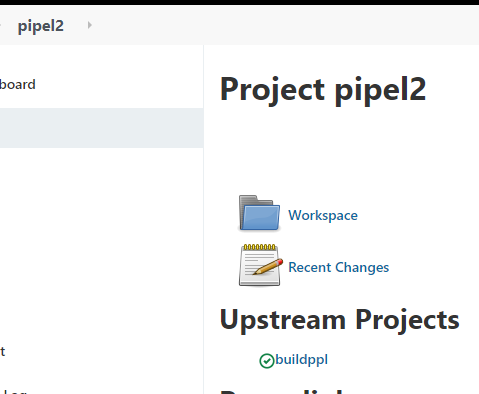
When a commit action performed with changes in ansible branch then pipeline2 “pipel2” will be triggered

Task 27:

Configure parent child trigger between pipeline1 and pipeline 2, after this setup whenever there is build completed for pipeline 1 then pipeline 2 also triggers immediately

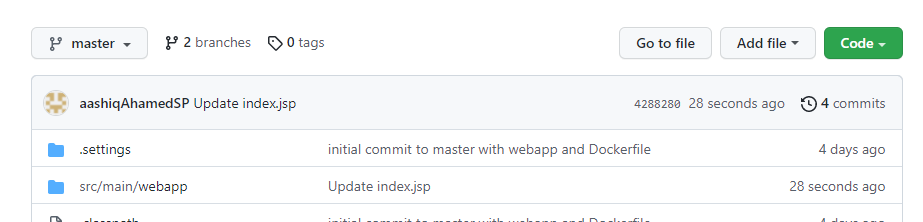




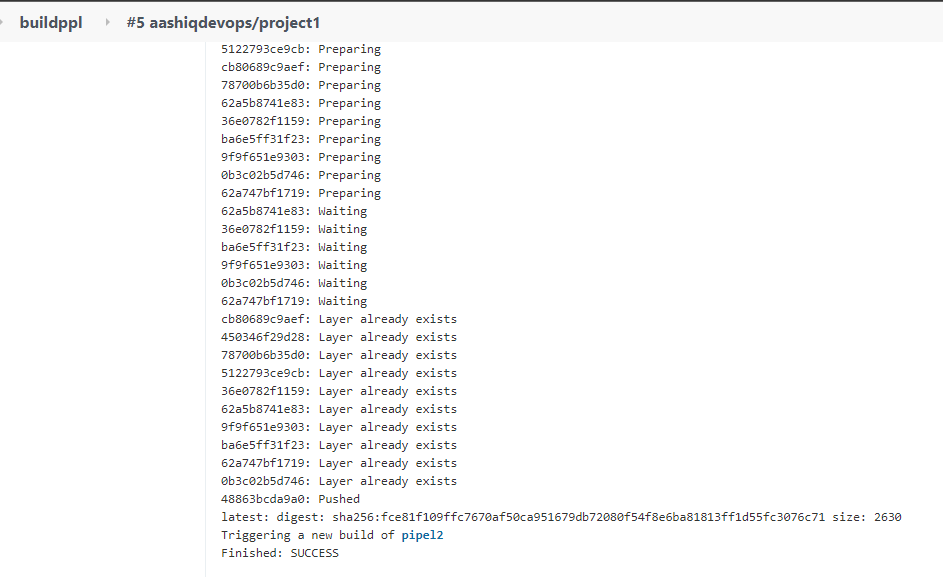


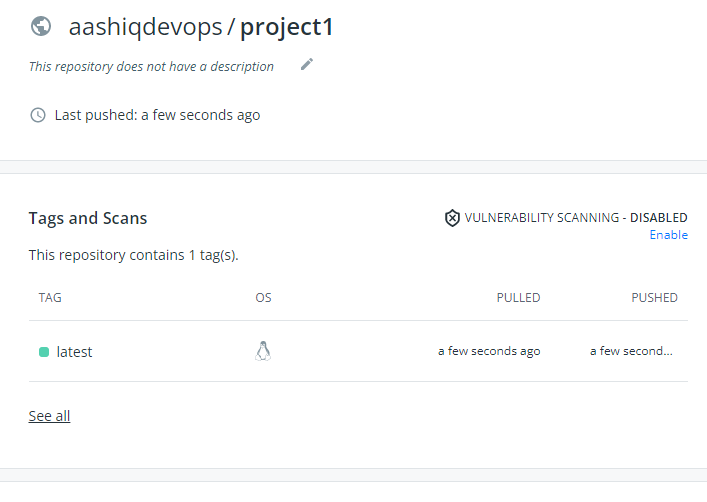
Task 28:

Commit a change in index.jsp of maven project through github



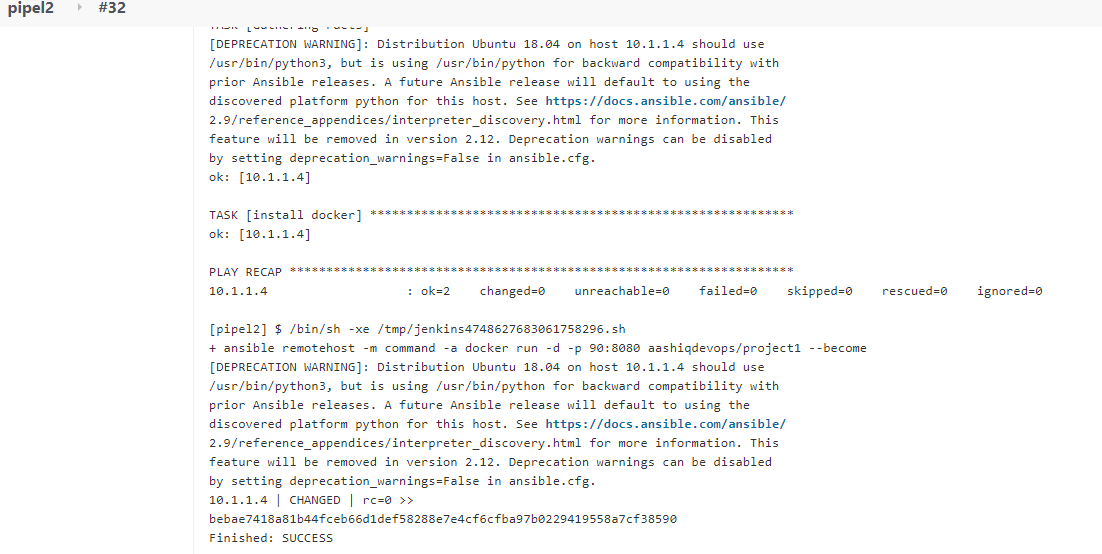
This will trigger pipeline1 in Jenkins and completed successfully

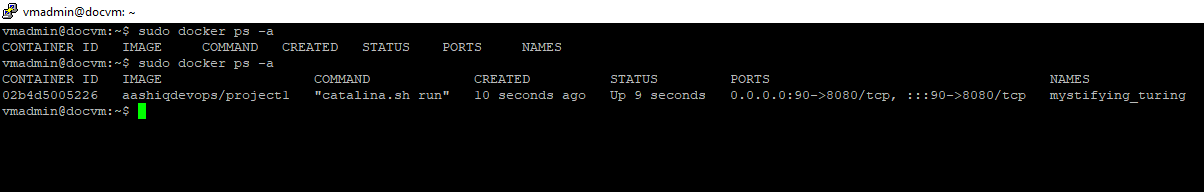




Once pipeline1 is done pipeline 2 triggers

Pipeline 2 triggered and finished





Task 29:

Check with the public IP of docvm

