

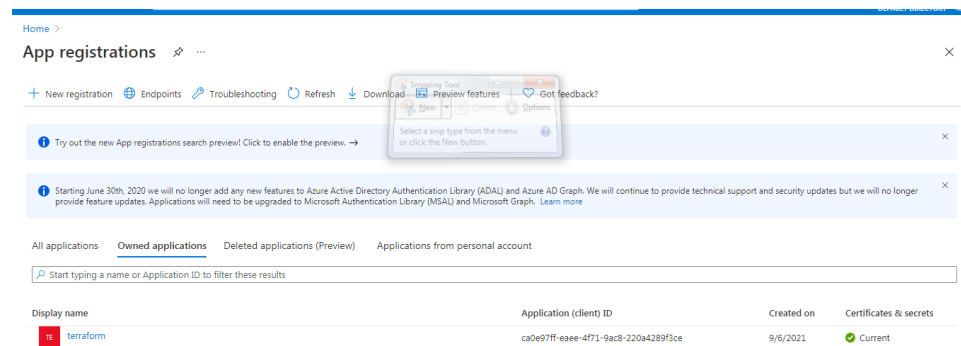
Project 1:

Task1:

```
PS C:\Users\labadmin> terraform --version
Terraform v1.0.6
on windows_amd64
PS C:\Users\labadmin> Set-ExecutionPolicy Unrestricted
Execution Policy Change
```

Terraform Version.

Generate Service Principle for Terraform in Azure:



Create required Vars.tf to create two VMs in Azure:

```

C: > project1 > variable.tf
3  #####
4
5  # Service Principal Variables
6
7
8  # Prefix and Tags
9
10 variable "prefix" {
11     description = "Prefix to append to all resource names"
12     type        = string
13     default     = "jenkins"
14 }
15
16 variable "tags" {
17     description = "Resource tags"
18     type        = map(string)
19     default     = {
20         "project"      = "jenkins"
21         "deployed_with" = "Terraform"
22     }
23 }
24
25 # Resource Group
26
27 variable "location" {
28     description = "Location of the resource group"
29     type        = string
30     default     = "West US"
31 }
32
33 # Vnet and Subnet
34
35 variable "vnet_address_range" {

```

Create main.tf to create two Linux VMs:

```

C: > project1 > linuxvm.tf
195 source_image_reference {
196     publisher      = var.publisher
197     offer          = var.offer
198     sku            = var.sku
199     version        = var.vm_image_version
200 }
201
202 connection {}
203 type = "ssh"
204 host = "${azurerm_public_ip.pip.1.ip_address}"
205 user = "vmadmin"
206 password = "Newone@13579"
207 {}
208
209
210
211 provisioner "file" {
212     source      = "slave.sh"
213     destination = "/home/vmadmin/slave.sh"
214 }
215
216 provisioner "remote-exec" {
217     inline = [
218         "sudo apt-get update -y",
219         "sudo apt-get install dos2unix",
220         "dos2unix slave.sh",
221         "chmod +x /home/vmadmin/slave.sh",
222         "/home/vmadmin/slave.sh",
223     ]
224 }
225

```

Use Terraform Provisioner to install JDK and Jenkins in VM1:

```
C: > project1 > master.sh
1  |! /bin/sh
2  sudo apt-get update -y
3  sudo apt-get install openjdk-8-jdk openjdk-8-jre -y
4  sudo chmod 777 /etc/environment
5  sudo echo 'JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64' >> /etc/environment
6  sudo echo 'JRE_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre' >> /etc/environment
7  wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
8  sudo sh -c 'echo deb https://pkg.jenkins.io/debian binary/ > /etc/apt/sources.list.d/jenkins.list'
9  sudo apt-get update
10 sudo apt-get install jenkins -y
```

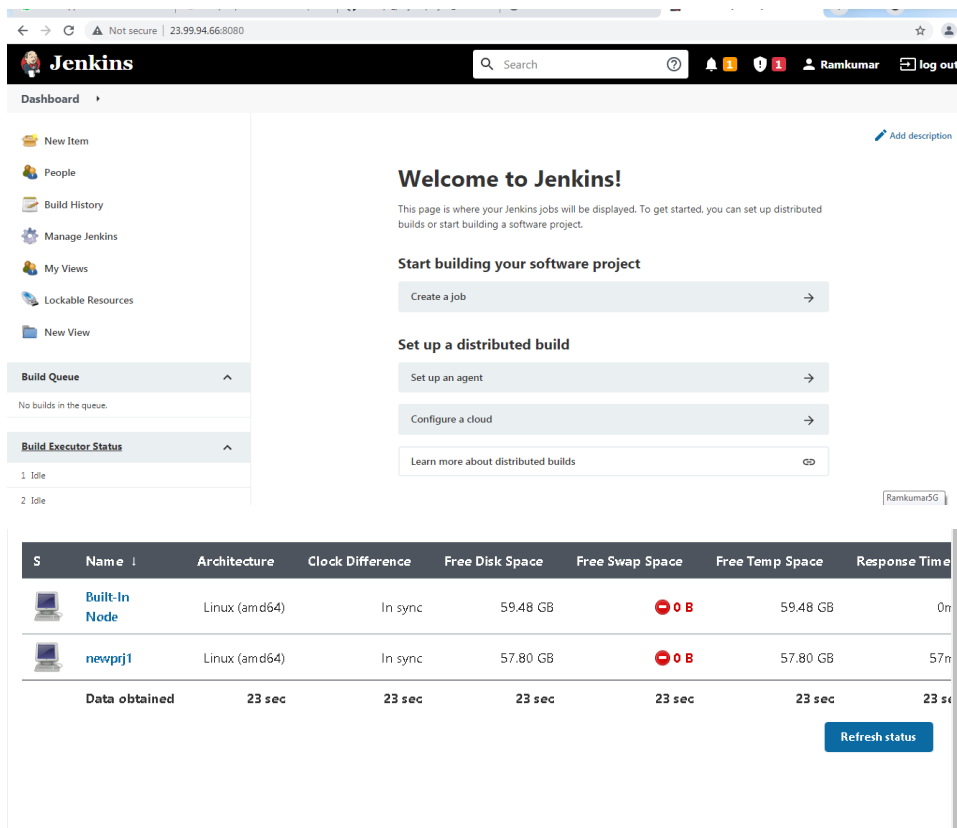
Use Terraform Provisioner to install JDK, Maven, Ansible, Docker, AzureCli and Git:

```
C: > project1 > slave.sh
1  |! /bin/sh
2  sudo apt-get update -y
3  sudo apt-get install openjdk-8-jdk openjdk-8-jre -y
4  sudo chmod 777 /etc/environment
5  sudo echo 'JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64' >> /etc/environment
6  sudo echo 'JRE_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre' >> /etc/environment
7  sudo apt-get install -y docker*
8  sudo service docker start
9  curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
10 sudo apt install python3-pip -y
11 sudo apt-get install software-properties-common -y
12 sudo apt-add-repository ppa:ansible/ansible -y
13 sudo apt-get update -y
14 sudo apt-get install ansible -y
15 sudo pip install ansible[azure]
16 sudo apt install maven -y
```

Init, Plan and Apply Terraform Script:

```
Progress: | 100% | (#####) ote-exec: Setting up libsisu-inject-java (0.3.3-1) ...
Progress: | 100% | (#####) ote-exec: Setting up libsisu-plexus-java (0.3.3-1) ...
Progress: | 100% | (#####) ote-exec: Setting up libsisu-core-java (3.6.3-1) ...
Progress: | 100% | (#####) ote-exec: Setting up maven (3.6.3-1) ...
Progress: | 100% | (#####) ote-exec: update-alternatives: using /usr/share/maven/bin/mvn to provide /usr/bin/mvn (mvn) in auto mode
Progress: | 100% | (#####) ote-exec:
azurerm_linux_virtual_machine.vms: Still creating... [7m40s elapsed]
azurerm_linux_virtual_machine.vms: creation complete after 7m41s [id=/subscriptions/4d9c19ca-b504-415c-a283-2e9b68b2d026/resourceGroups/jenkins-rg/providers/Microsoft.Compute/virtualMachines/mkfsbuild]
```

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

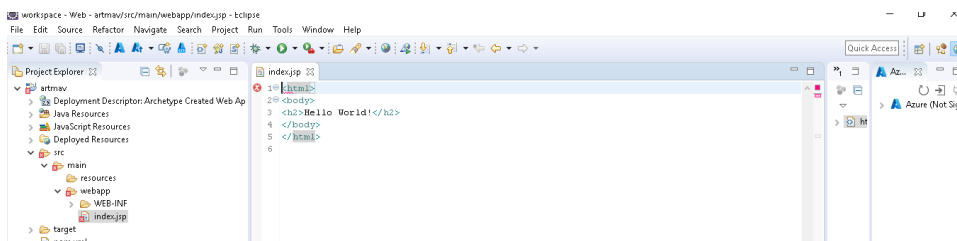


The screenshot shows the Jenkins Dashboard. On the left, there's a sidebar with navigation links: New Item, People, Build History, Manage Jenkins, My Views, Lockable Resources, and New View. Below these are sections for 'Build Queue' (showing 'No builds in the queue') and 'Build Executor Status' (showing two idle executors). The main area has a 'Welcome to Jenkins!' message, a 'Start building your software project' button, and a 'Set up a distributed build' section with buttons for 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'. At the bottom, there's a table with columns: S, Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, Free Temp Space, and Response Time. The table lists two executors: 'Built-In Node' and 'newprj1', both on Linux (amd64) and in sync. A 'Refresh status' button is at the bottom right.

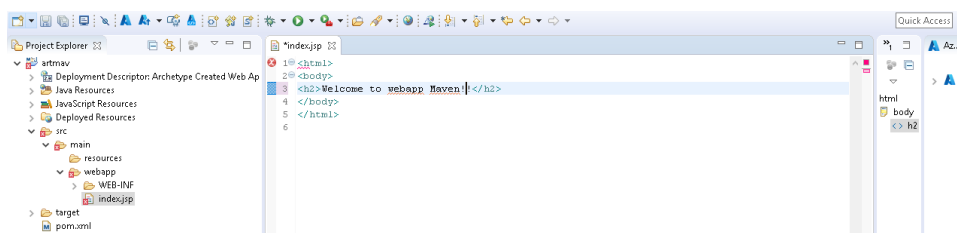
S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	59.48 GB	0 B	59.48 GB	0m
	newprj1	Linux (amd64)	In sync	57.80 GB	0 B	57.80 GB	57m
Data obtained		23 sec	23 sec	23 sec	23 sec	23 sec	23 sec

Part 2 – Phase1:

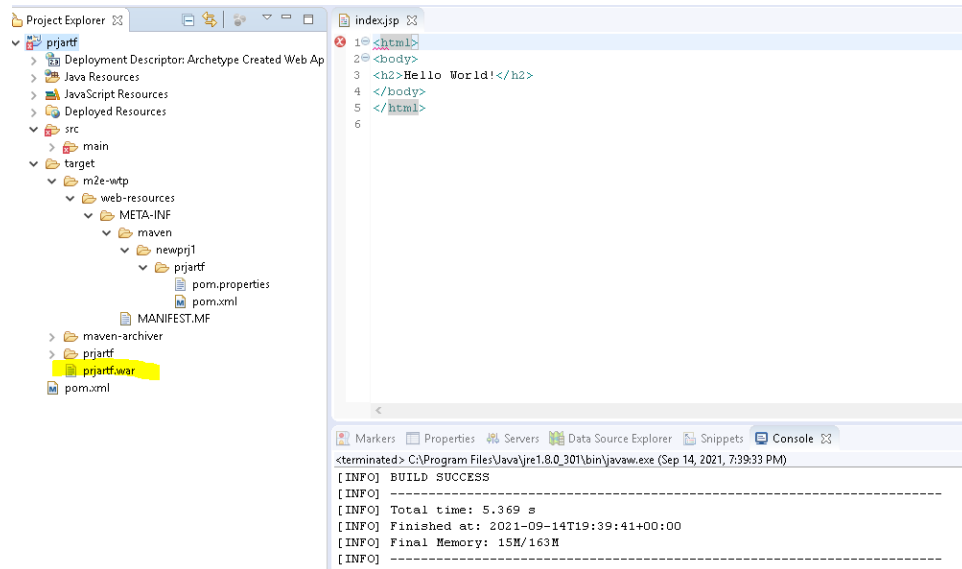
Create Maven Project with Archtype as web application in eclipse:



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

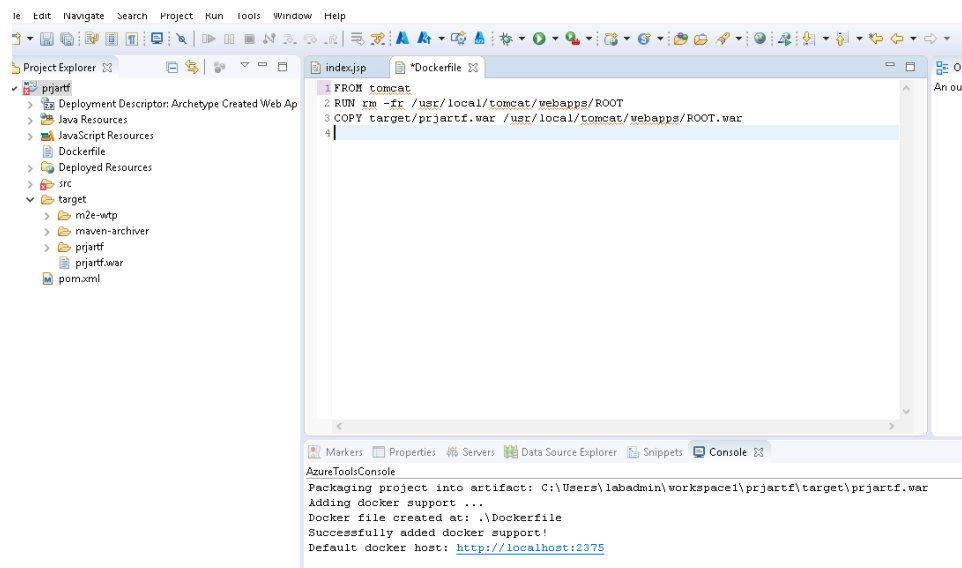


Run Maven clean install in eclipse to check the build and ckeck for .war file in target folder



Part 2- Phase 2

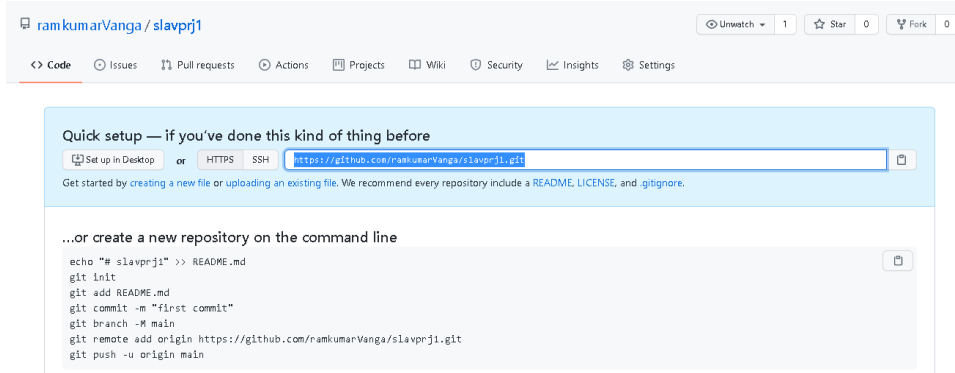
Gerate Dockerfile under project folder of your app & Modify FORM statement to use tomcat as base image



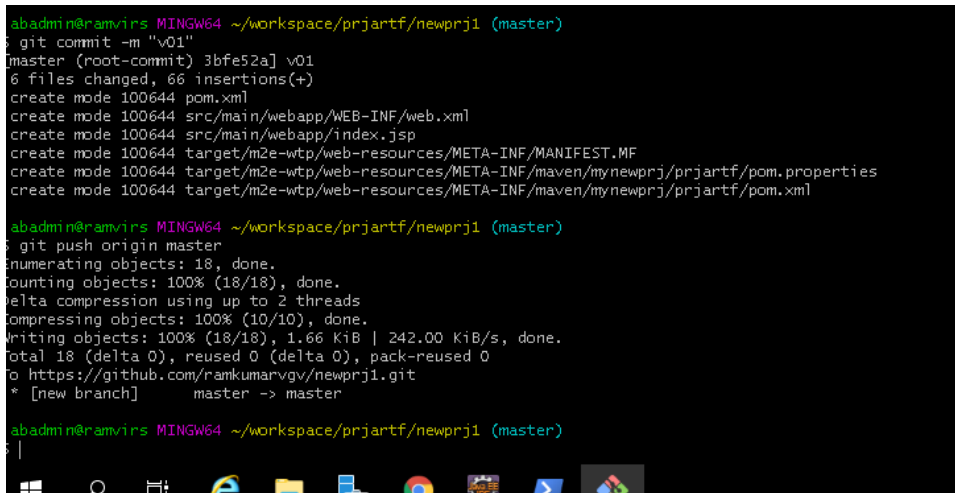
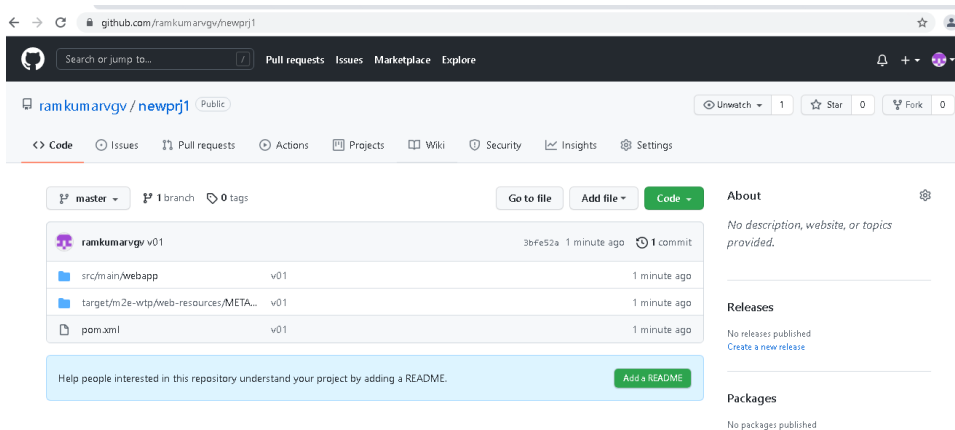
Part 2- Phase 3

Create a github repository and copy repo URL:

<https://github.com/ramkumarvgv/newprj1.git> - Repo URL



Commit and Push the code to remote repo



Modify ansible.cfg to use hosts file as inventory:

```

# config file for ansible -- https://ansible.com/
# =====

# nearly all parameters can be overridden in ansible-playbook
# or with command line flags.  ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory or /etc/ansible/ansible.cfg, whichever it
# finds first

[defaults]

# some basic default values...

inventory      = /etc/ansible/hosts
library        = /usr/share/my_modules/
module_utils   = /usr/share/my_module_utils/
remote_tmp     = ~/.ansible/tmp
local_tmp      = ~/.ansible/tmp
plugin_filters_cfg = /etc/ansible/plugin_filters.yml
forks          = 5
poll_interval  = 15
sudo_user      = root
ask_sudo_pass  = True
ask_pass       = True
transport      = smart
remote_port    = 22

```

install python-pip in ansible server

```

vmadmin@Jenkinsbuild:/etc/ansible$ sudo apt-get install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pip is already the newest version (20.0.2-5ubuntu1.6).
0 upgraded, 0 newly installed, 0 to remove and 160 not upgraded.
vmadmin@Jenkinsbuild:/etc/ansible$ python3-pip --version
python3-pip: command not found
vmadmin@Jenkinsbuild:/etc/ansible$

```

Create Repo in hub.docker.com:

labadmin123
Repositories
newprj123
Using 0 of 1 private repositories. [Get more](#)

General
Tags
Builds
Collaborators
Webhooks
Settings

Advanced Image Management
View all your images and tags in this repository, clean up unused content, recover untagged images. Available for Pro and Team accounts.
[View preview](#)

labadmin123/newprj123
This repository does not have a description

Last pushed: never

Docker commands
To push a new tag to this repository.
[Public View](#)

```
docker push labadmin123/newprj123:tagname
```

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

Maven

Maven installations

Add Maven

Maven

Name

mvn

MAVEN_HOME

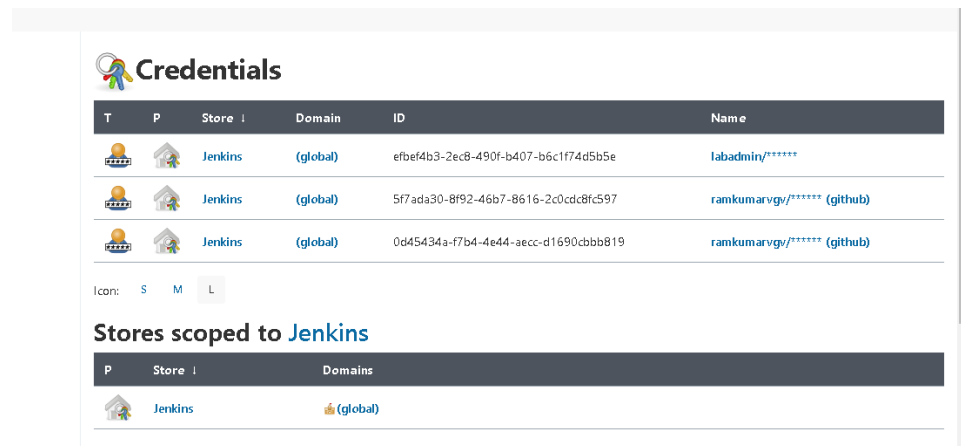
/usr/share/maven

/usr/share/maven is not a directory on the Jenkins controller (but perhaps it exists on some agents)

☐
Install automatically

Delete Maven

Configure Git credentials in Jenkins Vault



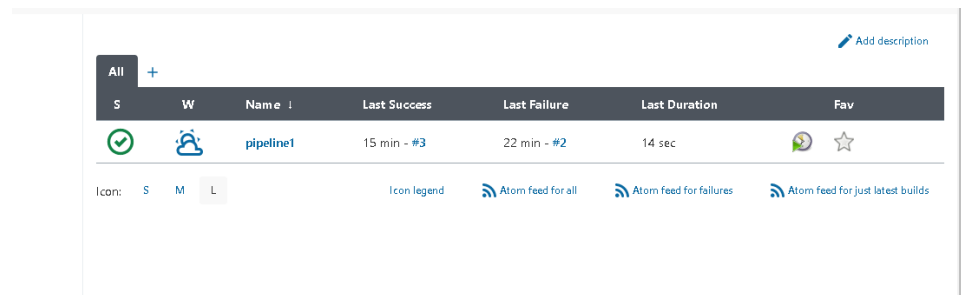
The screenshot shows the Jenkins 'Credentials' page. At the top, there's a header 'Credentials' with a Jenkins logo. Below it is a table with columns: T, P, Store, Domain, ID, and Name. There are three entries, all with 'Jenkins' as the Store and 'global' as the Domain. The first entry has ID 'efbef4b3-2ec8-490f-b407-b6c1f74d5b5e' and Name 'labadmin/*****'. The second and third entries have IDs '5f7ada30-8f92-46b7-b616-2c0cd8f597' and '0d45434a-f7b4-4e44-aecc-d1690cbbb819' respectively, and both have Name 'ramkumarvgv/***** (github)'. Below the table, there's a section 'Stores scoped to Jenkins' with a table showing 'Jenkins' and 'global' domains.

T	P	Store	Domain	ID	Name
		Jenkins	(global)	efbef4b3-2ec8-490f-b407-b6c1f74d5b5e	labadmin/*****
		Jenkins	(global)	5f7ada30-8f92-46b7-b616-2c0cd8f597	ramkumarvgv/***** (github)
		Jenkins	(global)	0d45434a-f7b4-4e44-aecc-d1690cbbb819	ramkumarvgv/***** (github)

Stores scoped to Jenkins

P	Store	Domains
	Jenkins	(global)

Create Pipeline1 using Freestyle project in Jenkins



The screenshot shows the Jenkins 'Pipeline1' project page. At the top, there's a header 'Pipeline1' with a Jenkins logo. Below it is a table with columns: S, W, Name, Last Success, Last Failure, Last Duration, and Fav. There is one entry with a green checkmark icon, a Jenkins icon, Name 'pipeline1', Last Success '15 min - #3', Last Failure '22 min - #2', Last Duration '14 sec', and Fav icon. Below the table, there's a section 'Icon legend' with links for 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'.

S	W	Name	Last Success	Last Failure	Last Duration	Fav
		pipeline1	15 min - #3	22 min - #2	14 sec	

Icon legend

- Atom feed for all
- Atom feed for failures
- Atom feed for just latest builds

In SCM stage Pull code form Remote Repo



The screenshot shows the Jenkins 'SCM' configuration page. At the top, there's a radio button for 'None' and a selected radio button for 'Git'. Below it is a section 'Repositories' with a 'Repository URL' field containing 'https://github.com/ramkumarvgv/newprj2'. There's a 'Credentials' dropdown menu showing 'ramkumarvgv/***** (github)' and an 'Add' button. Below the 'Repositories' section is a section 'Branches to build' with a 'Branch Specifier (blank for 'any')' field containing '*/main'. There are 'Advanced...' and 'Add Repository' buttons at the bottom right.

☐ None
☒ Git

Repositories

Repository URL
https://github.com/ramkumarvgv/newprj2

Credentials
ramkumarvgv/***** (github)

Advanced...
Add Repository

Branches to build

Branch Specifier (blank for 'any')
*/main

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

Invoke top-level Maven targets

Maven Version

mvn

Goals

clean install

Advanced...

Add build step

Post-build Actions

Add post-build action

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

Docker Build and Publish

Repository Name

labadmin123 / newprj123

Tag

latest

Docker Host URI

Server credentials

- none -

Add

Docker registry URL

Dashboard
> pipeline1
> #6 labadmin123/newprj123:latest labadmin123/newprj123:latest

```

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/labadmin123/newprj123]
c7029473806f: Preparing
4831bcd1167f: Preparing
977cfbc0f0fa: Preparing
4e4de253c94d: Preparing
3991808a925b: Preparing
d402f4f1b906: Preparing
00ef5416d927: Preparing
8555e663f65b: Preparing
d00da3cd7763: Preparing
4e61e63529c2: Preparing
799760671c38: Preparing
d402f4f1b906: Waiting
00ef5416d927: Waiting
8555e663f65b: Waiting
d00da3cd7763: Waiting
4e61e63529c2: Waiting
799760671c38: Waiting
977cfbc0f0fa: Layer already exists
4831bcd1167f: Layer already exists
c7029473806f: Layer already exists
3991808a925b: Layer already exists
4e4de253c94d: Layer already exists
d402f4f1b906: Layer already exists
4e61e63529c2: Layer already exists
d00da3cd7763: Layer already exists
00ef5416d927: Layer already exists
799760671c38: Layer already exists
8555e663f65b: Layer already exists
latest: digest: sha256:1d156a5100463d7fdecb0b0872efb14f502858401e4aa41e70986fcdc9185e43 size: 2630
Finished: SUCCESS

```

Advanced Image Management

View all your images and tags in this repository. clean up unused content, recover untagged images. Available for Pro and Team accounts.

View preview

labadmin123 / newprj123

This repository does not have a description

Last pushed: a minute ago

Docker commands

Public view

To push a new tag to this repository,

```
docker push labadmin123/newprj123:tagname
```

Tags and Scans

VULNERABILITY SCANNING - DISABLED
Enable

This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
latest		a minute ago	a minute ago

See all

Automated Builds

Manually pushing Images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available on Pro and Team plans.

Upgrade to ProLearn more

Create Pipeline2 using Freestyle project in Jenkins

Jenkins

Search

1

1

labadmin

Dashboard » pipeline2 »

GeneralSource Code ManagementBuild TriggersBuild EnvironmentBuildPost-build Actions

Description

[Plain text] Preview

☐ Commit agent's Docker container

☐ Define a Docker template

☐ Discard old builds

☐ GitHub project

☐ Use custom workspace

GitLab Connection

In SCM stage pull code form Remote Repo

← → ↻ ⚠ Not secure 104.40.68.150:8080/job/pipeline2/configure ☆ 👤

Dashboard » pipeline2 »

GeneralSource Code ManagementBuild TriggersBuild EnvironmentBuildPost-build Actions

Repositories

Repository URL

https://github.com/ramkumarvgv/newprj2

Credentials

ramkumarvgv/***** (github)

Add

Advanced...
Add Repository

Branches to build

Branch Specifier (blank for 'any')

*/main

Add Branch

In Build Stage Step 1: Run ansible Playbook1

Dashboard > pipeline2 >

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

Invoke Ansible Playbook X

Ansible installation

ansible

Playbook path ?

playbook1.yml

Inventory

☐ Do not specify Inventory

☐ File or host list

☐ Inline content

Host subset ?

Credentials ?

- none - Add

In Build Stage Step 2: Call Shell Script

Dashboard > pipeline2 >

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

☐ sudo ? Advanced...

Execute shell X ?

Command

```
ip=$(echo `az vm list-ip-addresses -g jenkins-rg -n dook | head -7 | tail -1 | cut -c 12-19`)  
echo "ip" >> ~/ip.txt  
ssh-keyscan -f ~/ip.txt >> ~/.ssh/known_hosts  
echo "\n [linux] " >> /etc/ansible/hosts  
echo "\n $ip " >> /etc/ansible/hosts  
echo "\n [linux:vars] " >> /etc/ansible/hosts  
echo "ansible_user=labadmin" >> /etc/ansible/hosts  
echo "ansible_password=Newone@13579" >> /etc/ansible/hosts
```

See the list of available environment variables

Advanced...

Invoke Ansible Playbook X

Ansible installation

ansible

In Build Stage Step 3: Run Ansible Playbook3(docker.yml)

Dashboard > pipeline2 >

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

Invoke Ansible Playbook X

Ansible installation

ansible

Playbook path ?

docker.yml

Inventory

☐ Do not specify Inventory

☐ File or host list

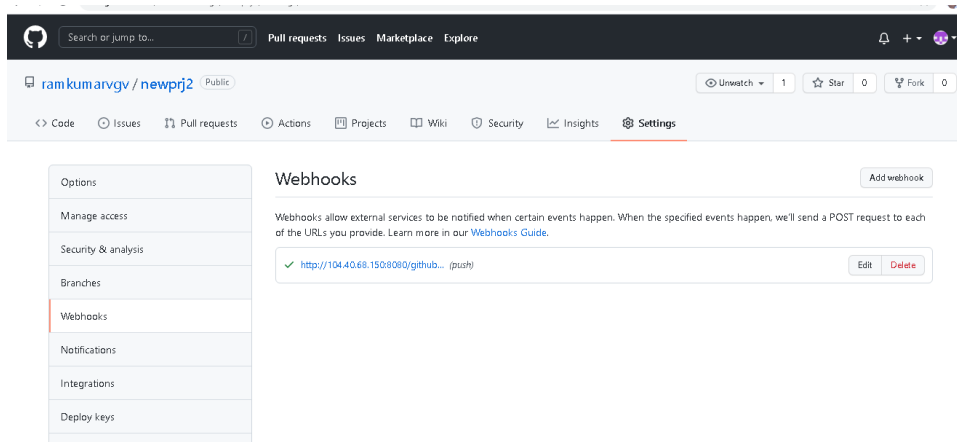
☐ Inline content

Host subset ?

Credentials ?

- none - Add

In Github repo configure webhook for Jenkins



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

