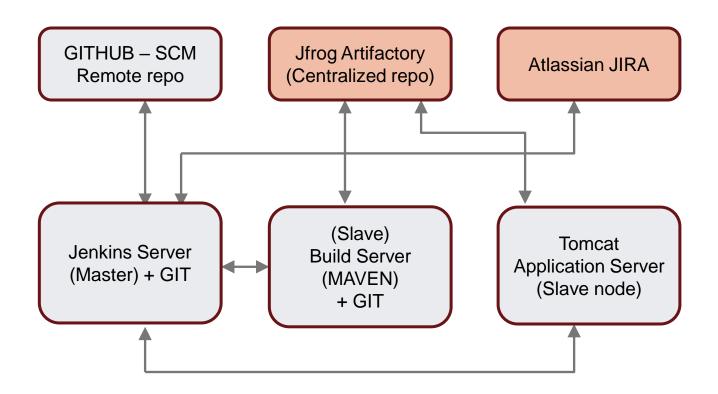# DEPLOYMENT SCENARIOS

Build Scenario -1

Build Scenario -1

Infrastructure setup:
- Jenkins installed on a Linux VM installed with GIT and Java JDK 1.8.
- Linux VM installed with MAVEN, GIT and Java JDL 1.8
- Linux VM installed with TOMCAT server, GIT and Java 1.8.
- Cloud VM instance installed with JIRA server.
- Cloud VM installed with Artifactory repository server.
- Install GIT, MAVEN and Tomcat Plug-in on the Jenkins server.
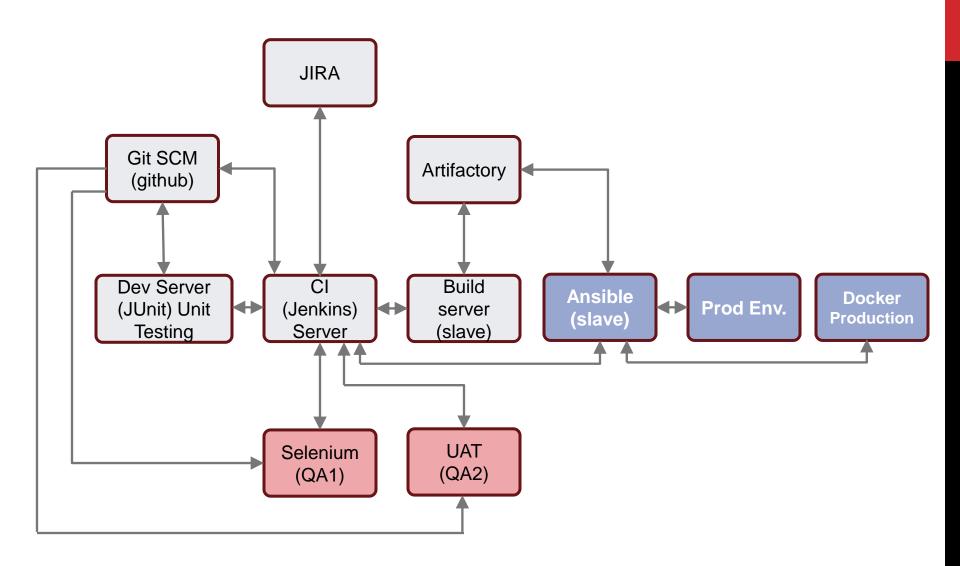- Install Jira and Artifactory plugin on Jenkins.

Jenkins JOB setup

Job 1:
- Create a build job on Jenkins. Configure it to run on MAVEN Build server.
- This Job should fetch the project repository from the remote repo to the local repo.
- Run MVN build goals with clean and Package arguments.
- Upload / export artifacts to the Artifactory server.

Job 2:
- Create a job in Jenkins and configure it to run on TOMCAT server that's been added as a slave.
- Download Artifacts from Artifactory server using wget or similar command and
- Restart the TOMCAT engine so as to extract the application package archive.

Build Scenario -2

Build Scenario -2

Prerequisites:
Artifactory server configured as a Generic repository
Jira Software Project tracking server
Tomcat App Server as Production environment
Jenkins server
Build server (Maven)
Selenium Server (QA1)
UAT test environment (QA2)
Ansible server (Deployment server)
Docker Production environment Server
Access to GitHub
Integrate Jira with Jenkins
Integrate Artifactory with Jenkins
Integrate GIT and Maven with Jenkins
Ansible server
Docker Host
Add Build Server(Maven) as a slave in Jenkins
Add App Server (Tomcat) as a slave in Jenkins
Add Ansible Server as Slave

Job1: Tasks:
Pull source code from GIT
Compile the code, create package.
Run unit test
Check the results
On successful completion, run downstream job

Job2: Tasks:
send notification abt job start
Pull source code from GIT
Compile the code, create package.
Deploy package
Run Selenium functional test
on failure create a JIRA ticket
On succefull completion run downstream job

Job3: Tasks:
Pull source code from GIT
Compile the code, create package.
Run test in QA2 environment for UAT
On succefull completion push package to Artifactory server

Build Scenario -2

Job4: Tasks:
Pull package from Artifactory server
run Ansible playbook to deploy package onto production serve
Run nagios check to test production URL retunrs success
Job5: Tasks:
Pull package from Artifactory server
Run Ansible playbook to deploy package to Production Docker
RUN nagios check to test production URL retunrs success