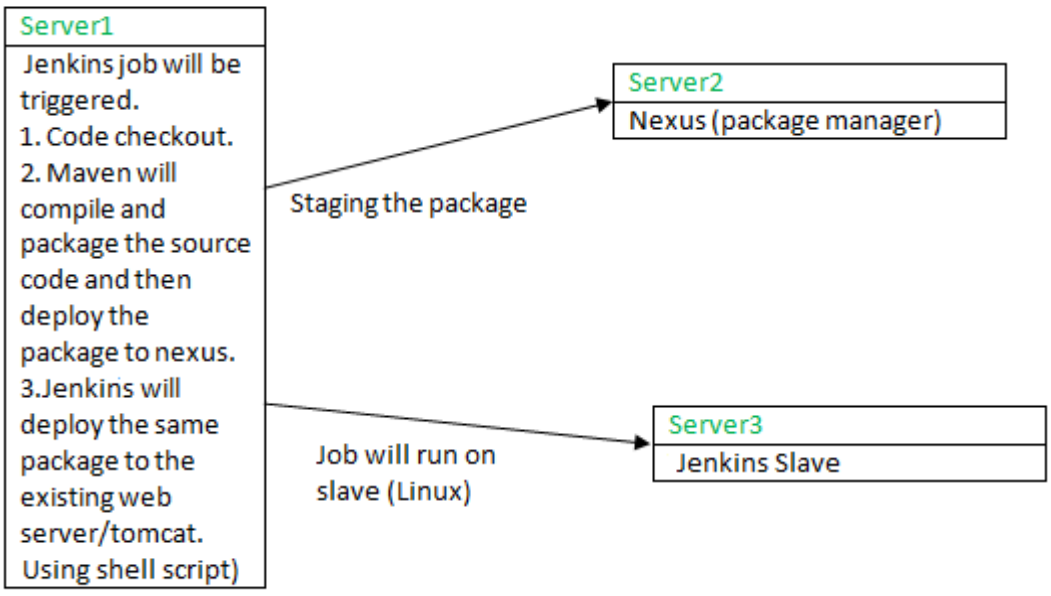
Phase-1: Implementation

Phases	Targeted Implementations	Tools Required	Number of instances	Status/duration	Supporting Documents
Phase-1.1	Install the required tools to check out the code from GitHub and then build.	Java, Tomcat, Maven, Git, Jenkins	1 RHEL	2 hrs	1.AWS-TOMCAT-JENKINS.doc, 2.Java-Tomcat-maven-git-Jenkins.sh
Phase-1.2	Implement as to get the code from GitHub--->build with maven-->deploy to nexus-->deploy to TomCat server.	Above tools + Nexus	2 RHEL	2 hrs	3.NexusSetup.sh
Phase-1.3	Jenkins master and slaves configuration.	Above tools	3 RHEL	Completed in one day	4.AWS-Jenkins-Slaves-Configuration.doc
Phase-1.4	Documentation of Phase-1 tasks	Google Docs/sheets	-	Completed in one day	**DevOps-Setup-Phase1-FinalDocs.zip**

Required Tools & servers:

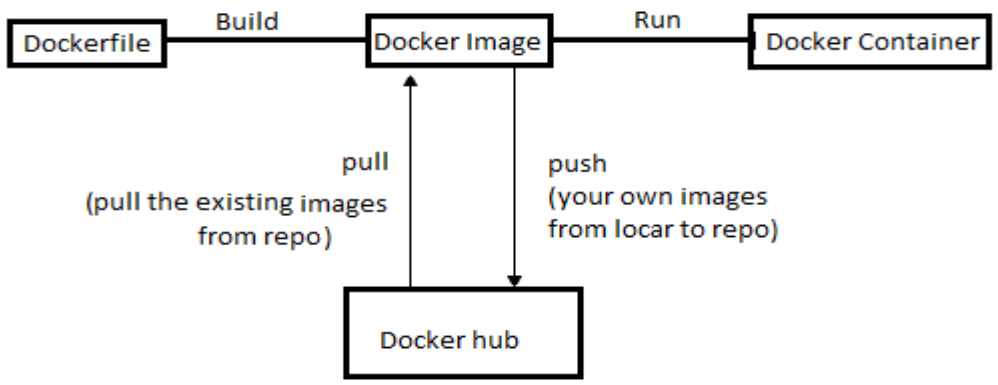
Server1	Server2	Server3
Java Tomcat Maven Git Jenkins	Nexus	Slave

Communication b/w servers:



Phases	Targeted Implementations	Tools Required	Number of instances	Status/duration	Supporting Documents (Follow the same sequence to refer docs)
Phase-2.1	Deploy using Ansible with jenkins free style job as wells as pipeline script.	Java, Tomcat, Maven, Git, Jenkins, Ansible	Total 3 RHELs. Tools on 1st RHEL = Java, Tomcat, Maven, Git, Jenkins, Ansible. Tools on 2nd RHEL = java, tomcat. 3rd RHEL = Jenkins slave	Completed in four days	2.1.1. AWS-ANSIBLE.txt 2.1.2. AWS-ANSIBLE.doc
Phase-2.2	Docker (containerization)	Java, Tomcat, Maven, Git, Jenkins, Docker	Total 3 RHELs. Tools on 1st RHEL= Java, Tomcat, Maven, Git, Jenkins. Tools on 2nd RHEL = java, tomcat. 3rd RHEL = Jenkins slave, docker, git, java.	Completed in three days	2.2.1. AWS-Docker-Introduction.sh 2.2.2. Docker-Practice-1.txt 2.2.3. Docker-Practice-1.doc 2.2.4. Docker-Practice-2.sh 2.2.5. Docker-Practice-2.doc
Phase-2.3	Documentation of Phase-2 tasks	Doc	-	Completed in one day	DevOps-Setup-Phase2-FinalDocs.zip

Docker workflow:



Phase-3: Implementation

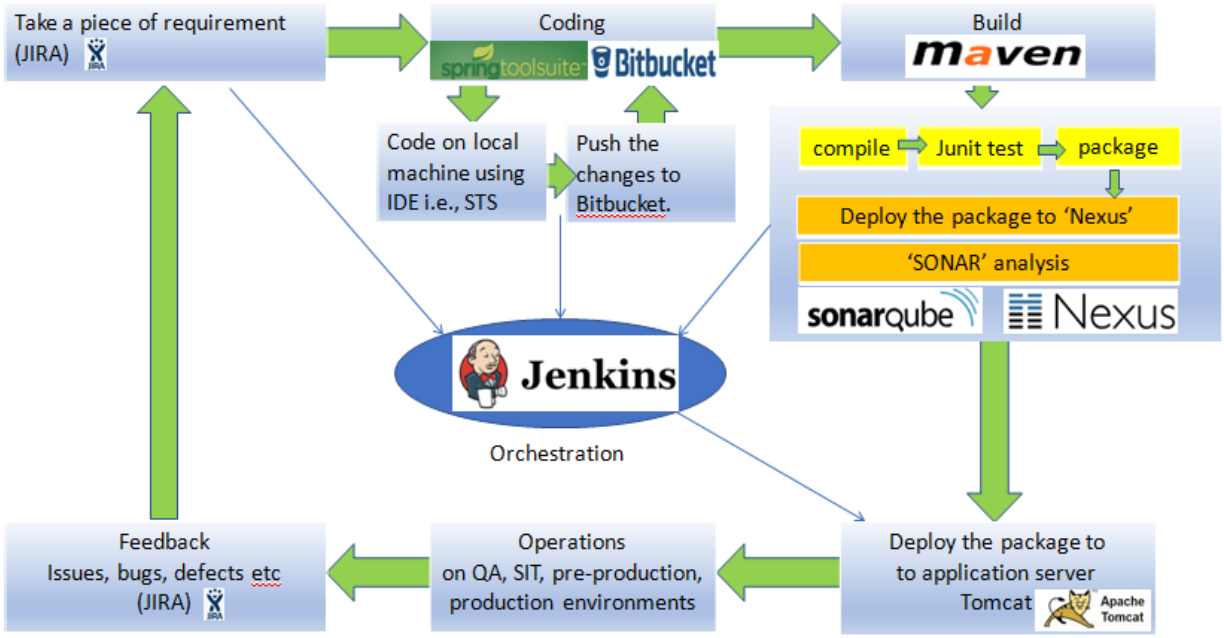
Phases	Targeted Implementations	Tools Required	Number of instances	Status/duration	Supporting Documents
Phase-3.1	JIRA Installation with DB	JIRA, PostgreSQL	1 RHEL	2 hrs	Phase-3.1
Phase-3.2	Practical JIRA with Jenkins+GitHub	Java, Tomcat, Maven, Git, Jenkins, JIRA	2 RHEL	Completed in three days	Phase-3.2
Phase-3.3	SonarQuebe with DB	SONAR, PostgreSQL	1 Ubuntu	2 hrs	Phase-3.3
Phase-3.4	Practical SONAR with Jenkisn+GitHub(Jenkins + Maven + Sonar integration)	Java, Tomcat, Maven, Git, Jenkins, SONAR	2 RHEL	Completed in four days	Phase-3.4

Phase-4: Implementation

Phases	Targeted Implementations	Tools Required	Number of instances	Status/duration	Supporting Documents
Phase-4.1	Bitbucket installation with DB	Bitbucket, PostgreSQL	1 Ubuntu	Completed in one day	AWS-Bitbucket-PostgreSQL.sh, AWS-Bitbucket-PostgreSQL-Installation.doc
Phase-4.2	Integration of the tools: Bitbucket + JIRA	-	-	Completed in one day	AWS-Bitbucket-JIRA-PostgreSQL.doc
Phase-4.3	Integration of the tools &CI Setup: Bitbucket + JIRA, Bitbucket + Jenkins, JIRA + Jenkins	Jenkins,Git, Bitbucket, JIRA, Maven, tomcat	1 Ubuntu, 2 RHEL	Completed in one day	AWS-Jenkins-Bitbucket-JIRA-CI-CD-Practice.doc
Phase-4.4	Documentation of Phase-5 tasks	Google Docs/sheets	-	Completed in one day	**DevOps-Setup-Phase4-Docs-v0.2.zip**

Simple work-flow:

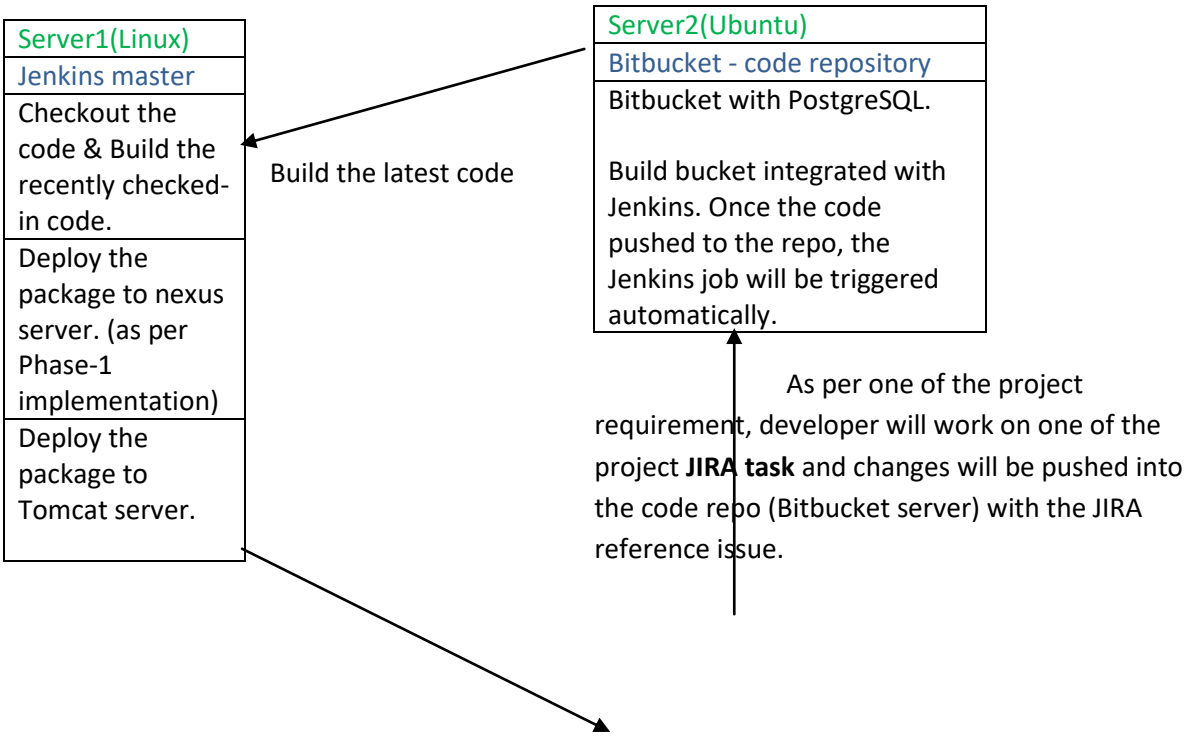
Requirement (in terms of JIRA task issue) →Coding in local machine and then push to remote server like bitbucket→Automatic Jenkins job build on new changes→Deploy the package to tomcat server.→ if any issues open a bug/defect issue in JIRA to the dev team. Repeat the steps. i.e., CI --- Continuous Integration.



Required Tools & servers:

Server/Type	Server1(RHEL)	Server2 (Ubuntu)	Server3(RHEL)
Server Activity	Jenkins master	Bitbucket - code repository	JIRA – bug tracker OR issue management OR project management tool
Tools need to be installed	Java, Tomcat, Maven, Git, Jenkins	Bitbucket PostgreSQL	JIRA PostgreSQL

Communication between servers:



After manual test or automated selenium, the test engineer will open a defect or bug in JIRA to fix the error if any error occurred while testing.	Server3(Linux) JIRA – bug tracker JIRA with PostgreSQL and this server integrated to Bitbucket server (server2) and Jenkins server (server1). Epics, Stories, Tasks, Bugs, defects are managed in JIRA projects.

(See the below image, in case the above tables are not proper in your word document)

Phase-5: Implementation

Phases	Targeted Implementations	Tools Required	Number of instances	Status/duration	Supporting Documents
Phase-5.1	Selenium Grid Installation	Java, Selenium JAR	Local Windows	Completed in one day	Jenkins+Maven+Selenium-OnLocalWindows.doc
Phase-5.2	Selenium + Maven + Jenkins with sample code web application automatic testing.	Java, Maven, Jenkins	Local Windows	Completed in one day	Refer the same above doc
Phase-5.3	AWS-Jenkins-Slaves-Linux-And-Windows-Configuration	Java	2RHEL, 1 Windows - AWS	Completed in one day	AWS-Jenkins-Slaves-Linux-And-Windows-Configuration.doc
Phase-5.4	Selenium Grid Installation on AWS windows	Java, Selenium jar	1 Windows	Completed in one day	Jenkins+Maven+Selenium-OnAWS.doc
Phase-5.5	Selenium + Maven + Jenkins on AWS Linux, Ubuntu, Windows instances	Java, Jenkins, Maven, Selenium Grid	2RHEL, 1Windows	Completed in one day	Refer the same above doc
Phase-5.6	Documentation of Phase-5 tasks	Google Docs/sheets	-	Completed in one day	**DevOps-Setup-Phase5-Docs-v0.3.zip**

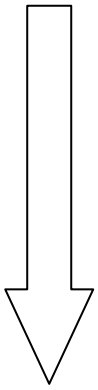
Example work-flow:

Original Code check-in for web app → Prepare Selenium script to test the web pages in different browsers → Trigger a local maven build OR trigger a Jenkins job as to Build the original code →

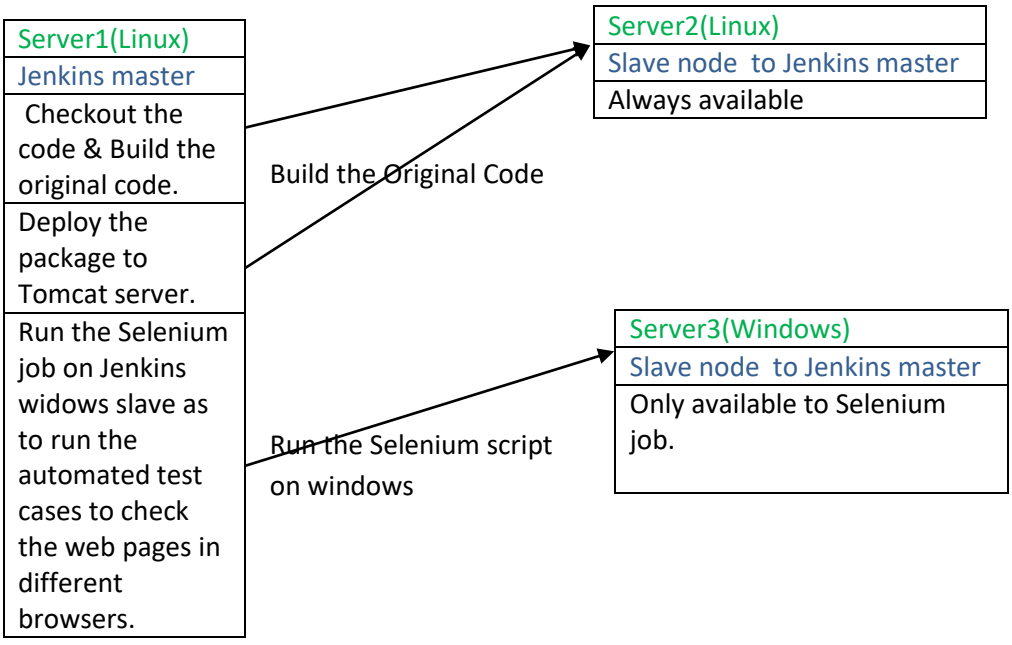
deploy the package to Tomcat server → build the Selenium job to run the automated test cases to check the web pages in different browsers.

Required Tools & servers:

Server/Type	Server1(RHEL)	Server2 (RHEL)	Server3(Windows)
Server Activity	Jenkins master	Slave node to Jenkins master	Slave node to Jenkins master
Tools need to be installed	Java Tomcat Maven Git Jenkins	Java Git Maven Tomcat	Java Git Maven Selenium Grid All required browsers



Communication between servers:



Phase-6: Implementation

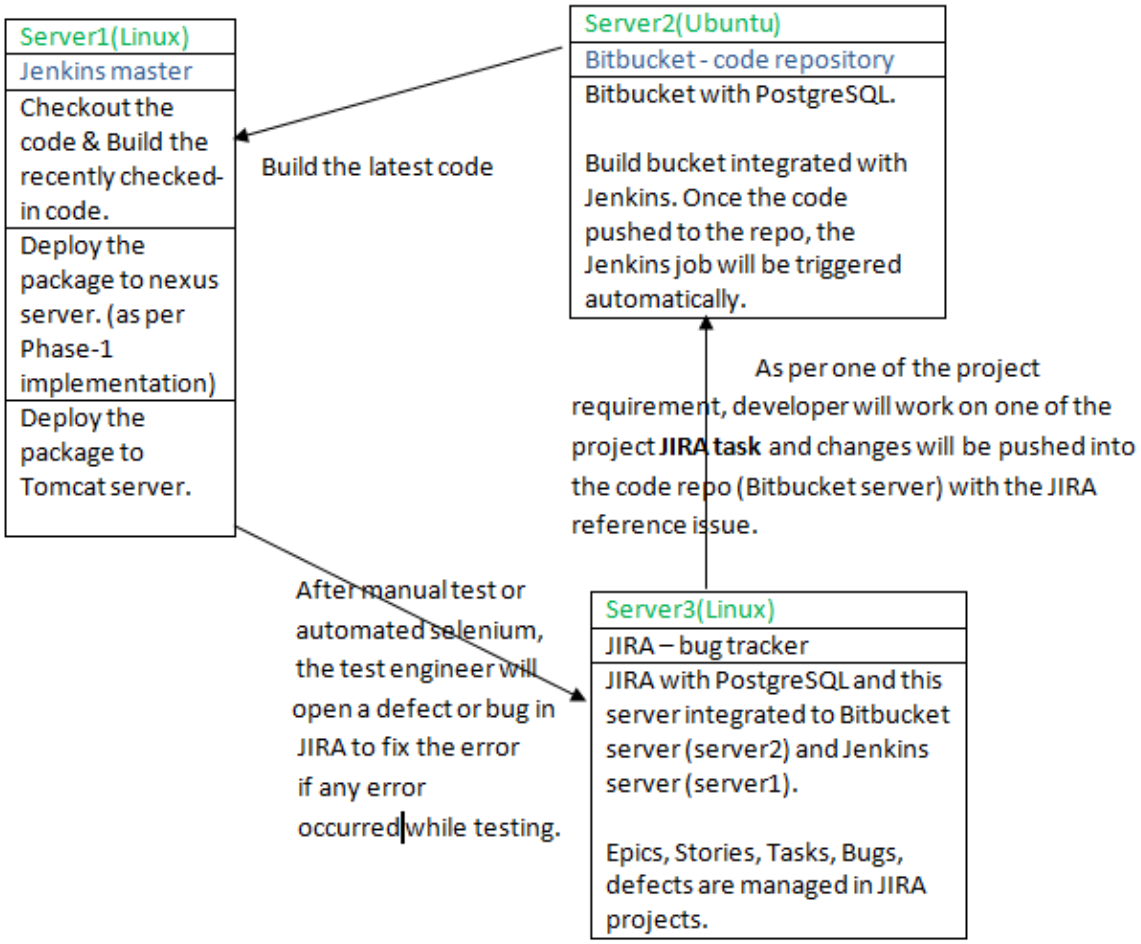
Phases	Targeted Implementations	Tools Required	Number of instances	Status/duration	Supporting Documents
Phase-6.1	LDAP with Jenkins	Java, Jenkins, LDAP	1 RHEL, 1 Ubuntu	**Phase-6.1**	LDAP with Jenkins
Phase-6.2	LDAP with JIRA	JIRA, PostgreSQL, LDAP	1 RHEL, 1 Ubuntu	**Phase-6.2**	LDAP with JIRA
Phase-6.3	LDAP with Bitbucket	Bitbucket, PostgreSQL, LDAP	2 Ubuntu	**Phase-6.3**	LDAP with Bitbucket
Phase-6.4	LDAP with SonarQube	Sonar, PostgreSQL, LDAP	1 RHEL, 1 Ubuntu	**Phase-6.4**	LDAP with SonarQube
Phase-6.5	LDAP with Splunk	Splunk, LDAP	1 RHEL, 1 Ubuntu	**Phase-6.5**	LDAP with Splunk
Phase-6.6	LDAP with Nagios	Nagios, LDAP	1 RHEL, 1 Ubuntu	**Phase-6.6**	LDAP with Nagios
Phase-6.7	LDAP with Nexus	Nexus, LDAP	1 RHEL, 1 Ubuntu	**Phase-6.7**	LDAP with Nexus

Phase-7&8: Implementation

Phases	Targeted Implementations	Tools Required	Number of instances	Status/duration	Supporting Documents
Phase-7.1	Splunk		1 RHEL, 1 Ubuntu	**Phase-7.1**	Splunk
Phase-7.2	Monitoring tools setup NAGIOS or any similar tools		1 RHEL, 1 Ubuntu	**Phase-7.2**	Monitoring tools setup NAGIOS or any similar tools
Phase-	Jenkins + Ansible		1 RHEL, 1	**Phase-7.3**	Jenkins + Ansible

7.3	integration		Ubuntu		integration
Phase-7.4	Jenkins + Docker integration		1 RHEL, 1 Ubuntu	**Phase-7.4**	Jenkins + Docker integration
Phase-7.5	Documentation of Phase-6 tasks	Docs/sheets		**Phase-7.5**	Documentation of Phase-6 tasks
Phase-8.1	Automatic provisioning AWS CLI or TERRAFORM or ANSIBLE			**Phase-8.1**	Automatic provisioning AWS CLI or TERRAFORM or ANSIBLE
Phase-8.2	auto scaling or load balancing			**Phase-8.2**	auto scaling or load balancing
Phase-8.3	AWS Important services			**Phase-8.3**	AWS Important services
Phase-8.4	Documentation of Phase-8 tasks	Docs/sheets		**Phase-8.4**	Documentation of Phase-8 tasks

Communication between servers:



Phase-5: Implementation

Phases	Targeted Implementations	Tools Required	Number of instances	Status/duration	Supporting Documents
Phase-5.1	Selenium Grid Installation	Java, Selenium JAR	Local Windows	Completed in one day	Jenkins+Maven+Selenium-OnLocalWindows.doc
Phase-5.2	Selenium + Maven + Jenkins with sample code web application automatic testing.	Java, Maven, Jenkins	Local Windows	Completed in one day	Refer the same above doc
Phase-5.3	AWS-Jenkins-Slaves-Linux-And-Windows-Configuration	Java	2RHEL, 1 Windows - AWS	Completed in one day	AWS-Jenkins-Slaves-Linux-And-Windows-Configuration.doc
Phase-5.4	Selenium Grid Installation on AWS windows	Java, Selenium jar	1 Windows	Completed in one day	Jenkins+Maven+Selenium-OnAWS.doc
Phase-5.5	Selenium + Maven + Jenkins on AWS	Java, Jenkins,	2RHEL, 1Windows	Completed in one day	Refer the same above doc

	Linux, Ubuntu, Windows instances	Maven, Selenium Grid			
Phase-5.6	Documentation of Phase-5 tasks	Google Docs/sheets	-	Completed in one day	**DevOps-Setup-Phase5-Docs-v0.3.zip**

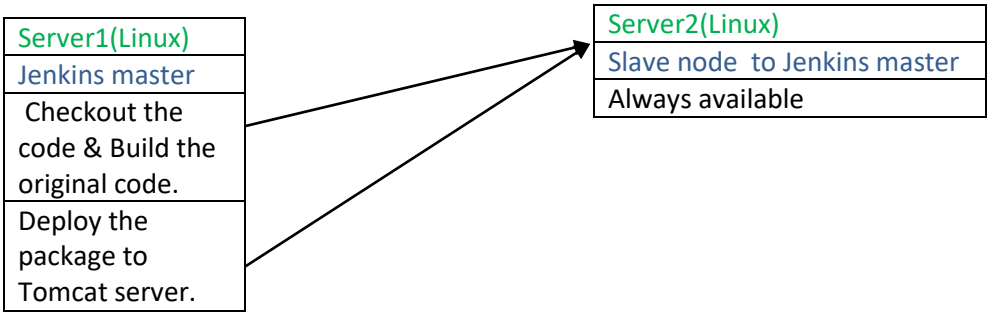
Example work-flow:

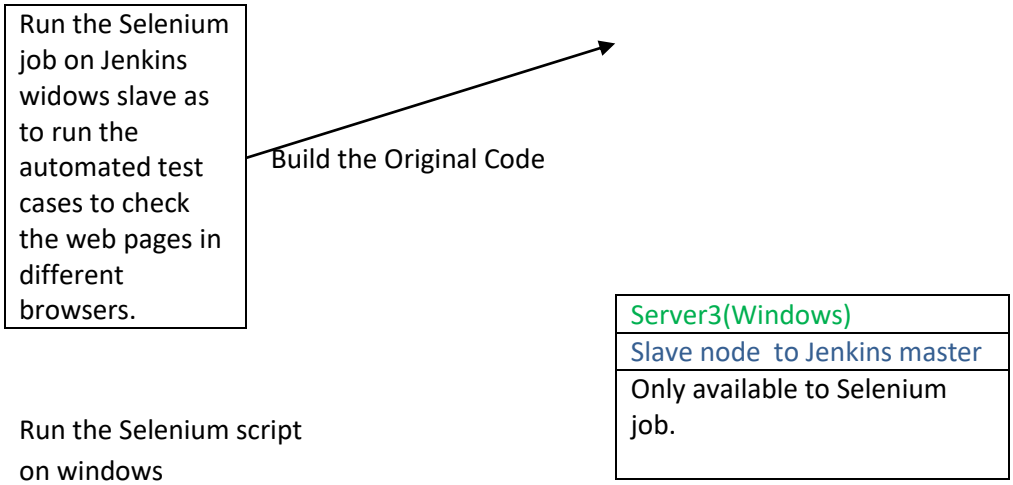
Original Code check-in for web app → Prepare Selenium script to test the web pages in different browsers → Trigger a local maven build OR trigger a Jenkins job as to Build the original code → deploy the package to Tomcat server → build the Selenium job to run the automated test cases to check the web pages in different browsers.

Required Tools & servers:

Server/Type	Server1(RHEL)	Server2 (RHEL)	Server3(Windows)
Server Activity	Jenkins master	Slave node to Jenkins master	Slave node to Jenkins master
Tools need to be installed	Java Tomcat Maven Git Jenkins	Java Git Maven Tomcat	Java Git Maven Selenium Grid All required browsers

Communication between servers:




Run the Selenium job on Jenkins widows slave as to run the automated test cases to check the web pages in different browsers.
Build the Original Code
Server3(Windows)
Slave node to Jenkins master
Only available to Selenium job.
Run the Selenium script on windows