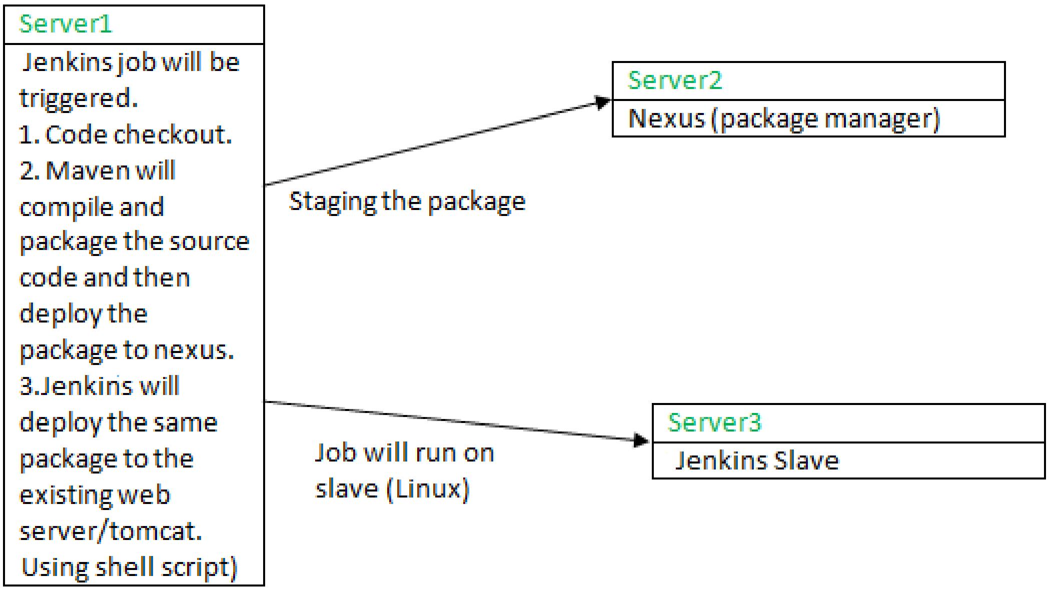
Phase-1: Implementation

Phases	Targeted Implementations	Tools Required	Number of instances	Status/duration
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
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
Phase-1.3	Jenkins master and slaves configuration.	Above tools	3 RHEL	Completed in one day
Phase-1.4	Documentation of Phase-1 tasks	Google Docs/sheets	-	Completed in one day

Required Tools & servers:

Server1	Server2	Server3
Java Tomcat Maven Git Jenkins	Nexus	Slave

Communication b/w servers:



Server1

Jenkins job will be triggered.
1. Code checkout.
2. Maven will compile and package the source code and then deploy the package to nexus.
3.Jenkins will deploy the same package to the existing web server/tomcat.
Using shell script)

Staging the package

Server2

Nexus (package manager)

Job will run on slave (Linux)

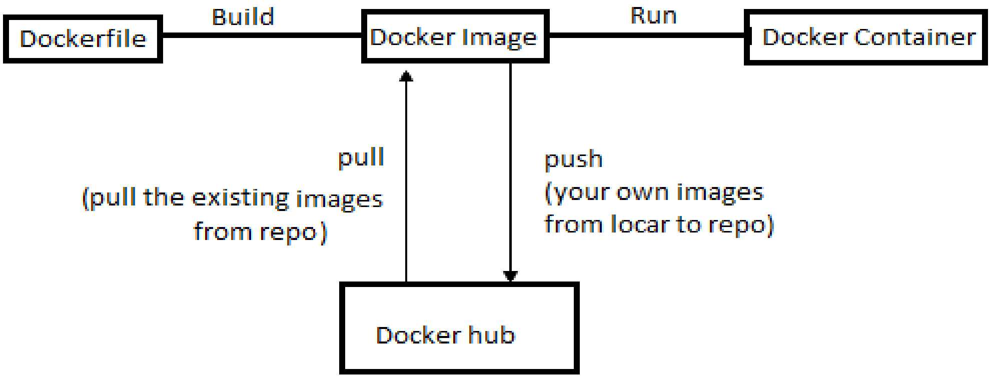
Server3

Jenkins Slave

Phase-2: Implementation

Phases	Targeted Implementations	Tools Required	Number of instances	Status/duration
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.	Completed in four days
			Tools on 1st RHEL = Java, Tomcat, Maven, Git, Jenkins, Ansible.	
			Tools on 2nd RHEL = java, tomcat.	
			3rd RHEL = Jenkins slave	
Phase-2.2	Docker (containerization)	Java, Tomcat, Maven, Git, Jenkins, Docker	Total 3 RHELs.	Completed in three days
			Tools on 1st RHEL= Java, Tomcat, Maven, Git, Jenkins.	
			Tools on 2nd RHEL = java, tomcat.	
			3rd RHEL = Jenkins slave, docker, git, java.	
Phase-2.3	Documentation of Phase-2 tasks	Doc	-	Completed in one day

Docker workflow:



Phase-3: Implementation

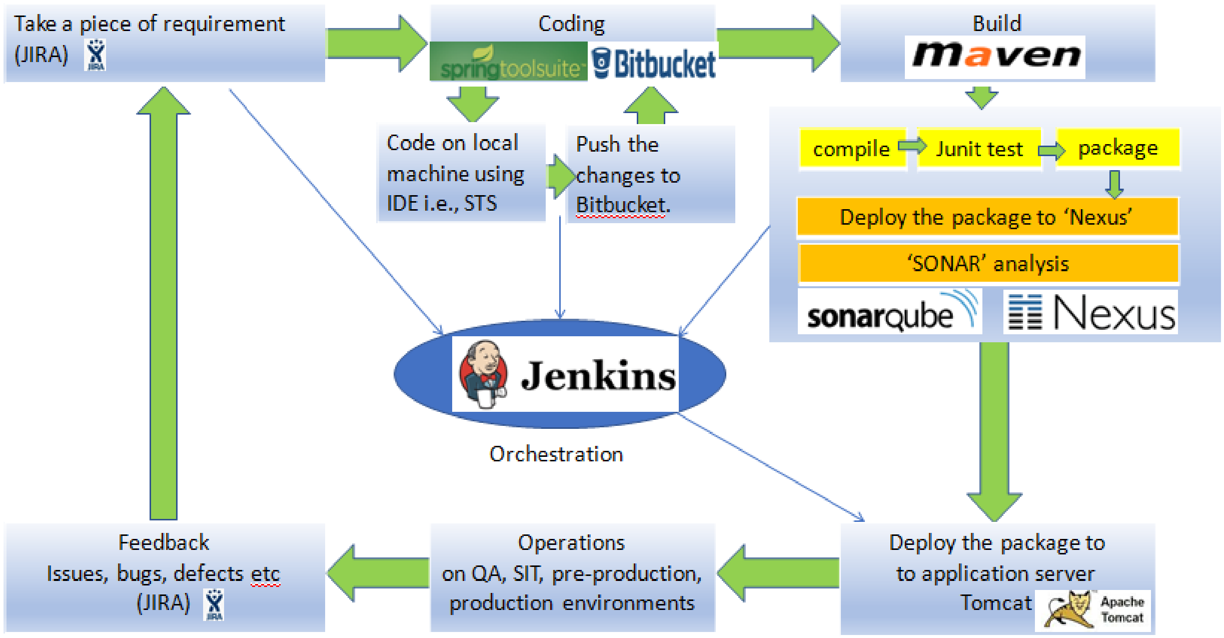
Phases	Targeted Implementations	Tools Required	Number of instances
Phase-3.1	JIRA Installation with DB	JIRA, PostgreSQL	1 RHEL
Phase-3.2	Practical JIRA with Jenkins+GitHub	Java, Tomcat, Maven, Git, Jenkins, JIRA	2 RHEL
Phase-3.3	SonarQuebe with DB	SONAR, PostgreSQL	1 Ubuntu
Phase-3.4	Practical SONAR with Jenkisn+GitHub(Jenkins + Maven + Sonar integration)	Java, Tomcat, Maven, Git, Jenkins, SONAR	2 RHEL

Phase-4: Implementation

Phases	Targeted Implementations	Tools Required	Number of instances
Phase-4.1	Bitbucket installation with DB	Bitbucket, PostgreSQL	1 Ubuntu
Phase-4.2	Integration of the tools: Bitbucket + JIRA	-	-
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
Phase-4.4	Documentation of Phase-5 tasks	Google Docs/sheets	-

Simple work-flow:

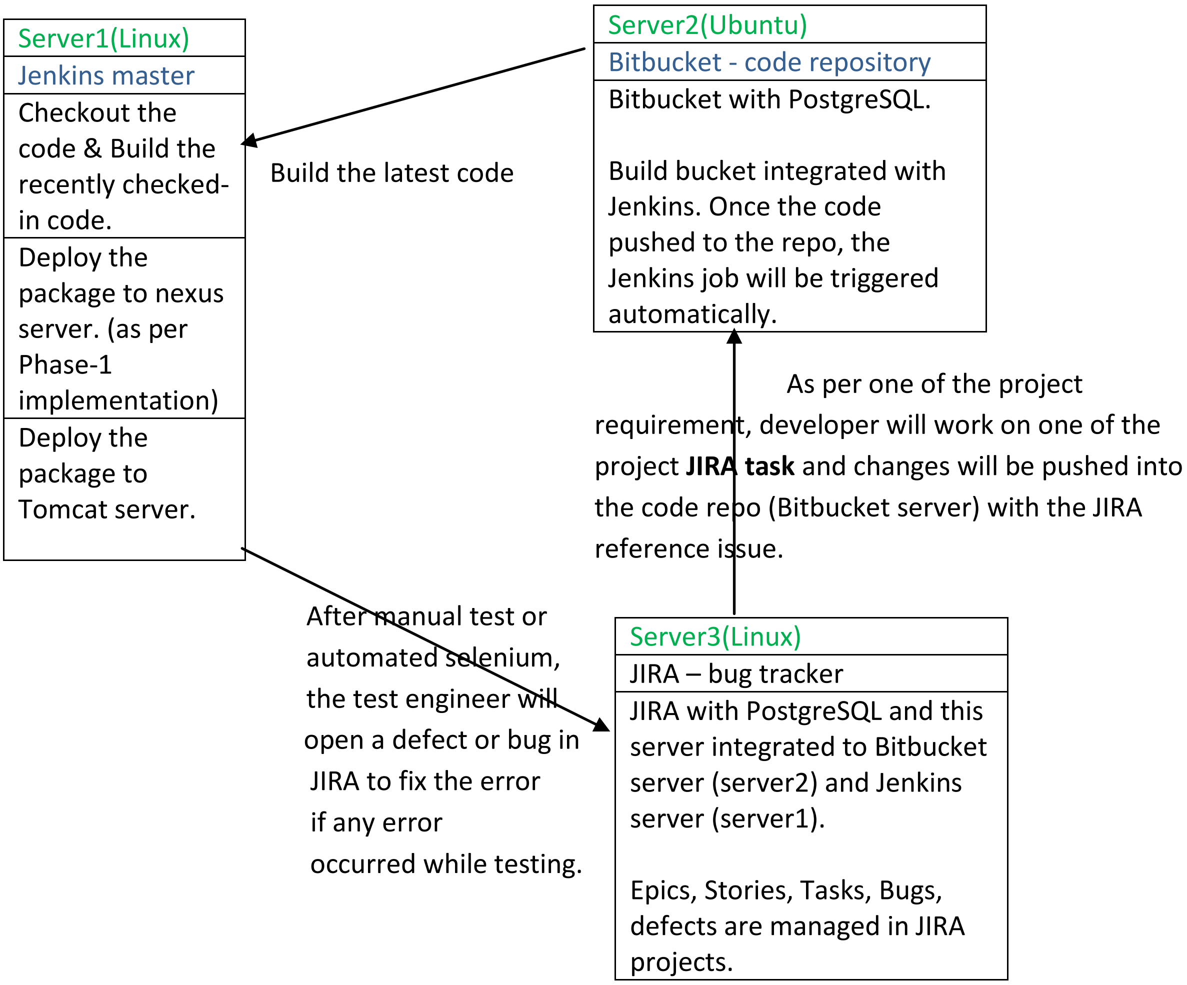
Requirement (in terms of JIRA task issue) →Coding in local machine and then push to remote server like bitbucket→Automatic Jenkins job build on new changes→Deploy the package to tomcat server.→ if any issues open a bug/defect issue in JIRA to the dev team. Repeat the steps. i.e., CI --- Continuous Integration.



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:



Server1(Linux)
Jenkins master
Checkout the code & Build the recently checked-in code.
Deploy the package to nexus server. (as per Phase-1 implementation)
Deploy the package to Tomcat server.

Build the latest code

Server2(Ubuntu)
Bitbucket - code repository
Bitbucket with PostgreSQL. Build bucket integrated with Jenkins. Once the code pushed to the repo, the Jenkins job will be triggered automatically.

As per one of the project requirement, developer will work on one of the project **JIRA task** and changes will be pushed into the code repo (Bitbucket server) with the JIRA reference issue.

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
Phase-5.1	Selenium Grid Installation	Java, Selenium JAR	Local Windows
Phase-5.2	Selenium + Maven + Jenkins with sample code web application automatic testing.	Java, Maven, Jenkins	Local Windows
Phase-5.3	AWS-Jenkins-Slaves-Linux-And-Windows-Configuration	Java	2RHEL, 1 Windows - AWS
Phase-5.4	Selenium Grid Installation on AWS windows	Java, Selenium jar	1 Windows
Phase-5.5	Selenium + Maven + Jenkins on AWS Linux, Ubuntu, Windows instances	Java, Jenkins, Maven, Selenium Grid	2RHEL, 1Windows
Phase-5.6	Documentation of Phase-5 tasks	Google Docs/sheets	-

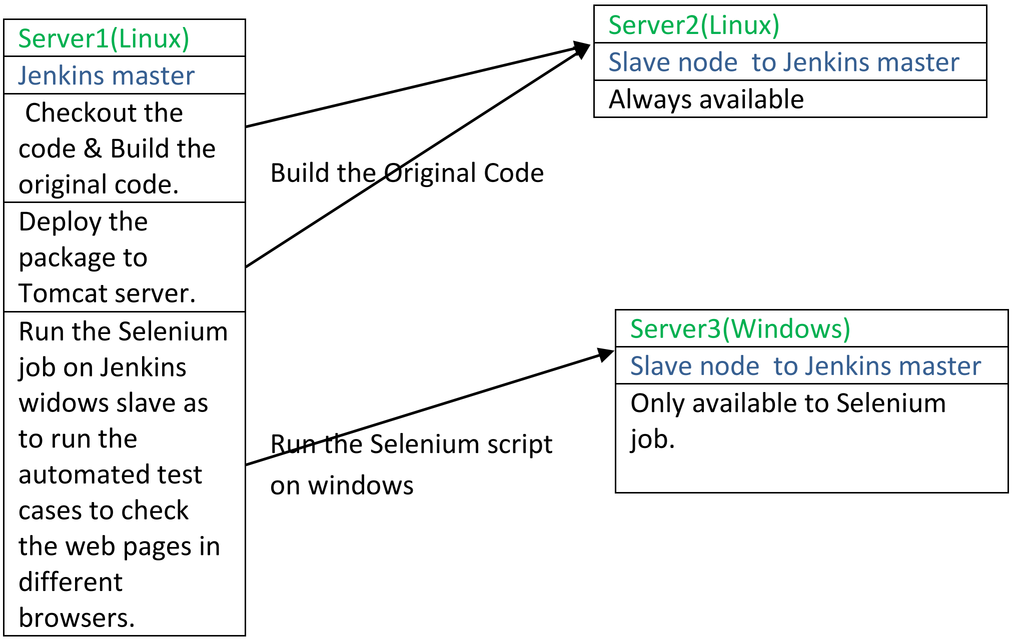
Example work-flow:

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:



Server1(Linux)

Jenkins master

Checkout the code & Build the original code.

Deploy the package to Tomcat server.

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

Run the Selenium script on windows

Server2(Linux)

Slave node to Jenkins master

Always available

Server3(Windows)

Slave node to Jenkins master

Only available to Selenium job.

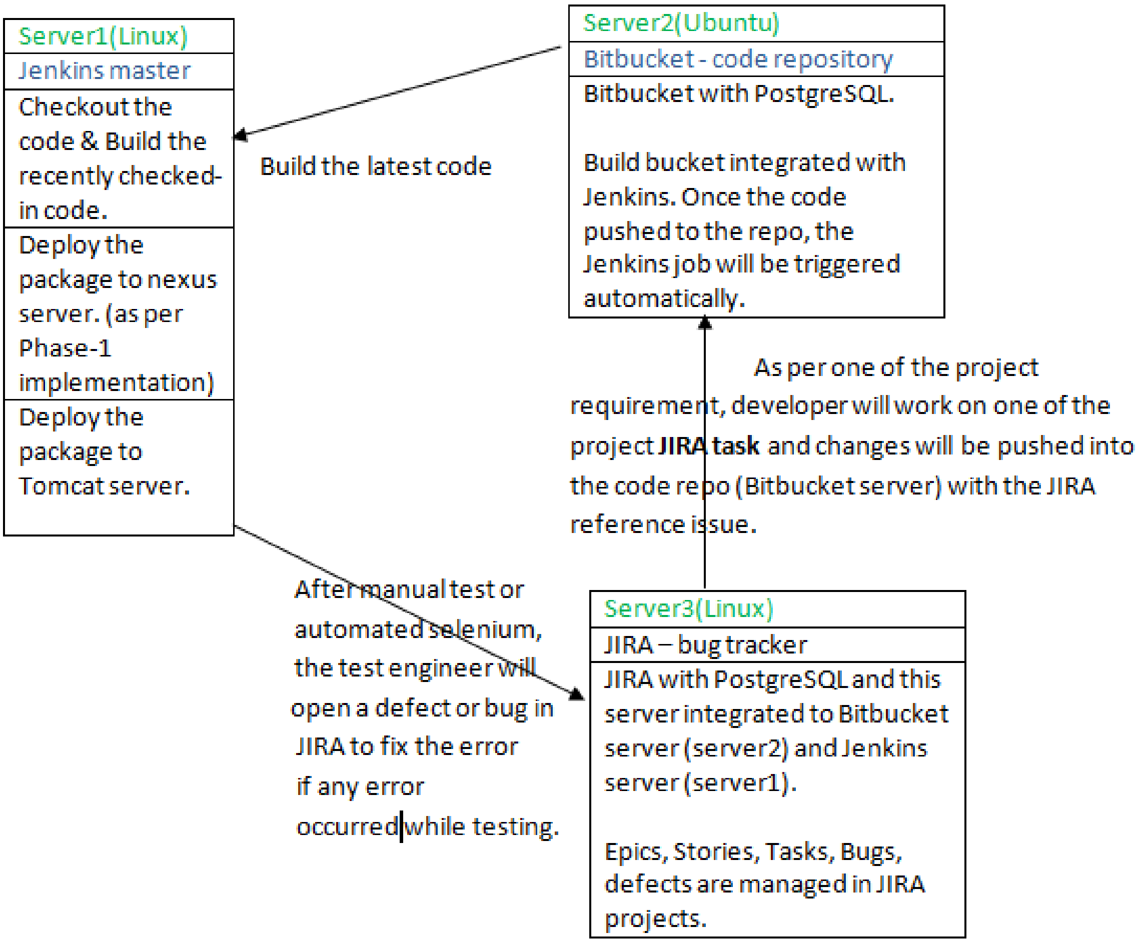
Phase-6: Implementation

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

Phase-7&8: Implementation

Phases	Targeted Implementations	Tools Required	Number of instances
Phase-7.1	Splunk		1 RHEL, 1 Ubuntu
Phase-7.2	Monitoring tools setup NAGIOS or any similar tools		1 RHEL, 1 Ubuntu
Phase-7.3	Jenkins + Ansible integration		1 RHEL, 1 Ubuntu
Phase-7.4	Jenkins + Docker integration		1 RHEL, 1 Ubuntu
Phase-7.5	Documentation of Phase-6 tasks	Docs/sheets	
Phase-8.1	Automatic provisioning AWS CLI or TERRAFORM or ANSIBLE		
Phase-8.2	auto scaling or load balancing		
Phase-8.3	AWS Important services		
Phase-8.4	Documentation of Phase-8 tasks	Docs/sheets	

Communication between servers:



Server1(Linux)

Jenkins master

Checkout the code & Build the recently checked-in code.

Deploy the package to nexus server. (as per Phase-1 implementation)

Deploy the package to Tomcat server.

Build the latest code

Server2(Ubuntu)

Bitbucket - code repository

Bitbucket with PostgreSQL.

Build bucket integrated with Jenkins. Once the code pushed to the repo, the Jenkins job will be triggered automatically.

As per one of the project requirement, developer will work on one of the project **JIRA task** and changes will be pushed into the code repo (Bitbucket server) with the JIRA reference issue.

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.

Build the Original Code

Server3(Windows)

Slave node to Jenkins master

Only available to Selenium job.

Run the Selenium script on windows