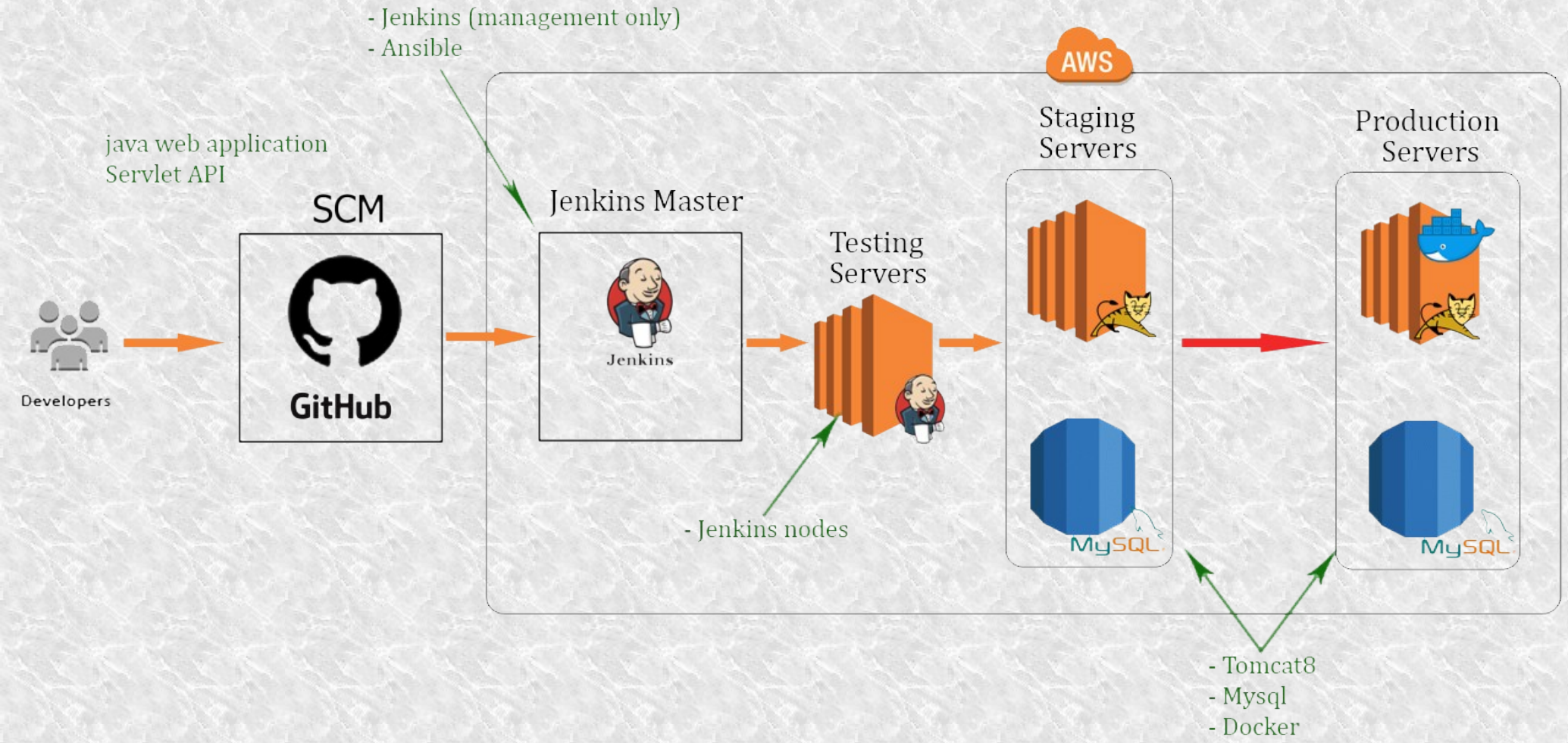
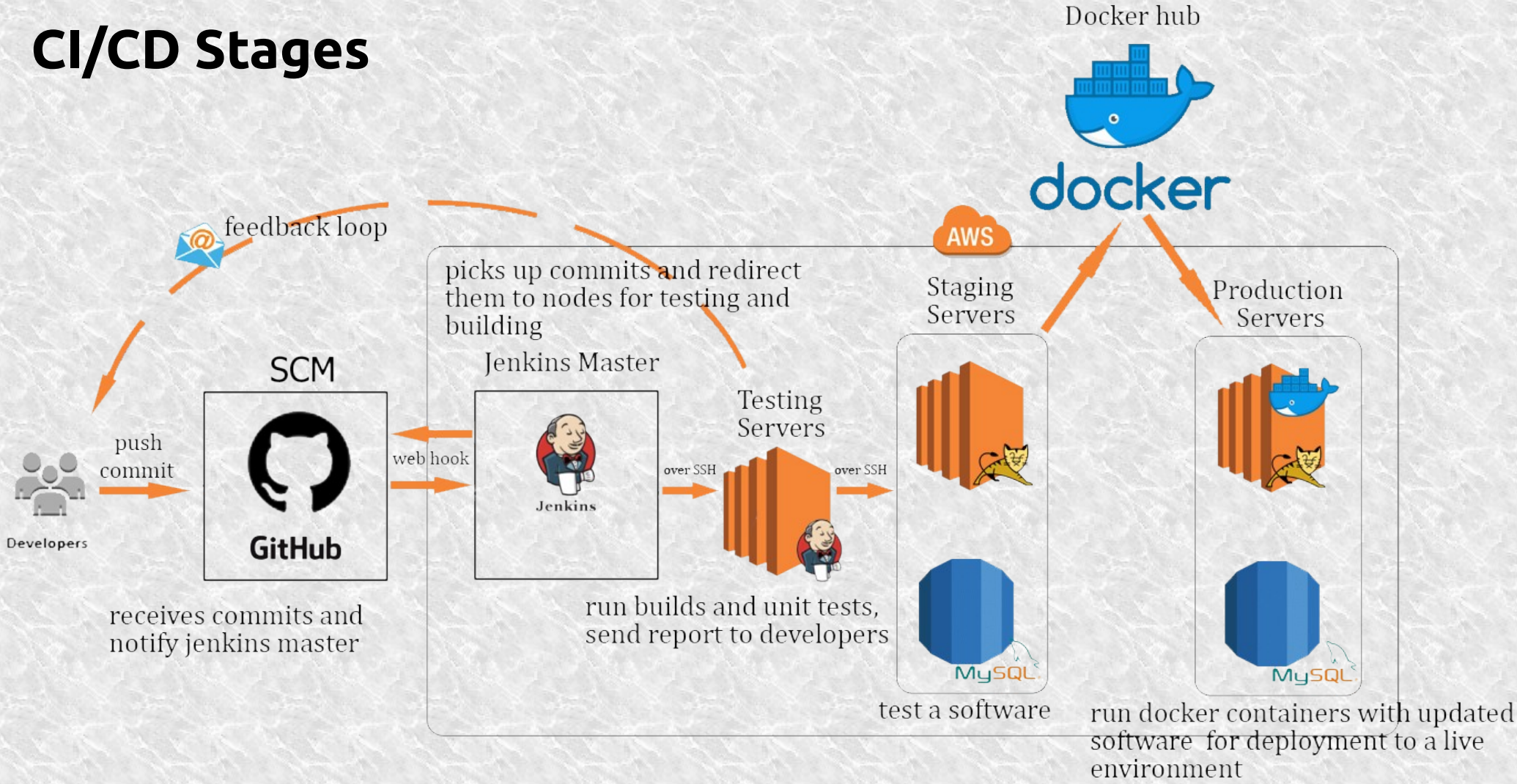
Project Task

1. Create book store management web application
2. Apply MySQL database to accessing and managing data
3. Focus on performance optimization

CI/CD Pipeline



CI/CD Stages


Docker hub
docker
feedback loop
AWS
picks up commits and redirect them to nodes for testing and building
Staging Servers
Production Servers
SCM
Jenkins Master
Testing Servers
push commit
web hook
over SSH
over SSH
Developers
GitHub
Jenkins
MySQL
MySQL
receives commits and notify jenkins master
run builds and unit tests, send report to developers
test a software
run docker containers with updated software for deployment to a live environment

Servers Configuration with Ansible Playbooks

1. Install Ansible on Jenkins Master server
2. Run Ansible Playbook for Jenkins Master to configure required software on it
3. Setup required software on all servers by running Ansible Playbooks


```
---
- name: Install JENKINS on localhost
  hosts: localhost
  connection: local
  become: yes

  tasks:
  - name: Install jenkins apt repository key
    apt_key: url=https://pkg.jenkins.io/debian-stable/jenkins.io.key state=present

  - name: Configure the jenkins repository on localhost
    apt_repository: repo='deb https://pkg.jenkins.io/debian-stable binary/' state=present

  - name: Install Java (openjdk-8-jdk)
    apt: name=openjdk-8-jdk state=latest

  - name: Install jenkins
    apt: name=jenkins state=latest update_cache=yes

  - name: Run jenkins' service and put it to autorun
    service: name=jenkins state=started enabled=yes

  - name: check if initialAdminPassword file exists
    stat: path=/var/lib/jenkins/secrets/initialAdminPassword
```

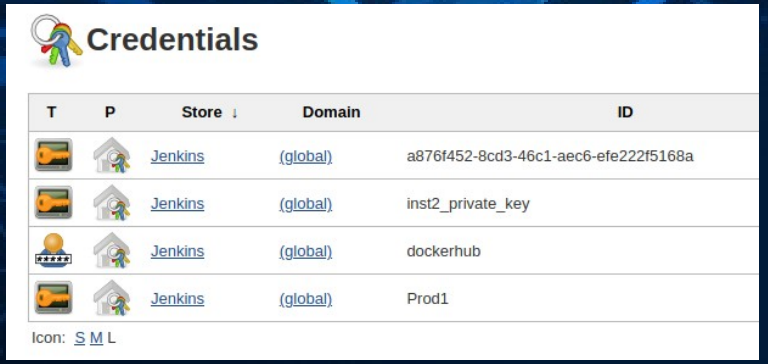
```
ubuntu@inst1:~/ansible/playbooks$ cat playbook_install_docker.yml
---
- name: Install Docker engine
  hosts: TEST_SERVERS,STAGING_SERVERS,PROD_SERVERS
  become: yes

  tasks:
  # ===== for Ubuntu only ======
  - name: Install transport-https for Ubuntu
    apt: name=apt-transport-https state=latest

  - name: Install Docker apt repository key
    apt_key: url=https://download.docker.com/linux/ubuntu/gpg state=present

  - name: Configure the Docker repository on localhost
    apt_repository: repo='deb https://download.docker.com/linux/ubuntu bionic stable' state=present

  - name: Update apt and install docker-ce
    apt: update_cache=yes name=docker-ce state=latest

  - name: Add user to docker group
    user:
      name: 'ubuntu'
      groups: docker
      append: yes

  - name: Unconditionally reboot the machine with all defaults
    reboot:
```

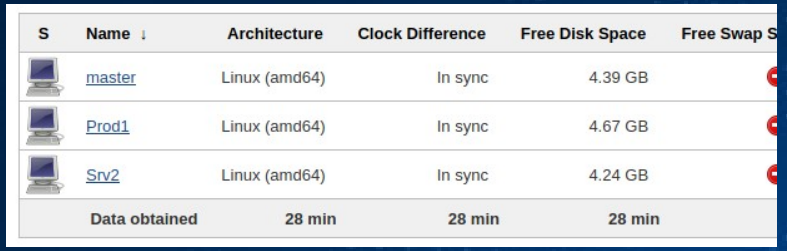
```
ubuntu@inst1:~/ansible/playbooks$ cat playbook_install_tomcat.yml
---
- name: Install Tomcat server
  hosts: STAGING_SERVERS,PROD_SERVERS
  become: yes

  tasks:
  - name: Check LINUX Version
    debug: var=ansible_os_family

  - block:        # ===== Block for Ubuntu ======
    - name: Install Java for Ubuntu
      apt: name=default-jre state=latest

    - name: Install Tomcat for ubuntu
      apt: name=tomcat8 state=latest

    - name: Start Tomcat service for ubuntu
      service: name=tomcat8 state=started enabled=yes

    when: ansible_os_family == "Debian"

  - block:        # ===== Block for RedHat ======

    - name: Install tomcat for RedHat
      yum: name=tomcat state=latest

    - name: Start tomcat service for RedHat
      service: name=tomcat state=started enabled=yes

    when: ansible_os_family == "RedHat"
```

Jenkins Configuration

1. Setup plugins for Jenkins
2. Add required credentials for GitHub, Nodes etc.
3. Create and configure the pipeline

Builds run on jenkins nodes only !



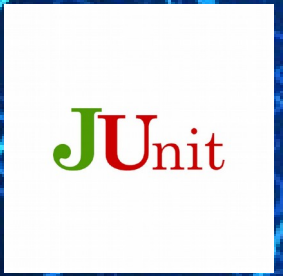


Building And Unit Testing

- Unit testing - JUNIT
- Maven builds war archive
- Feedback Loops

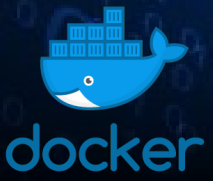
Testing Environment

- Publishing release on Staging servers
- Tests
- Sending Docker image with new version of release to Docker Hub

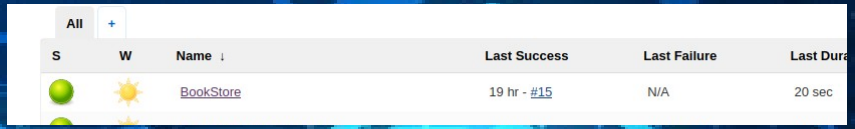


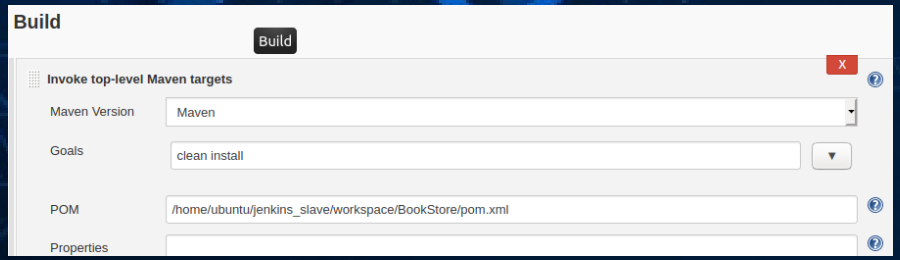
Deploying To Production

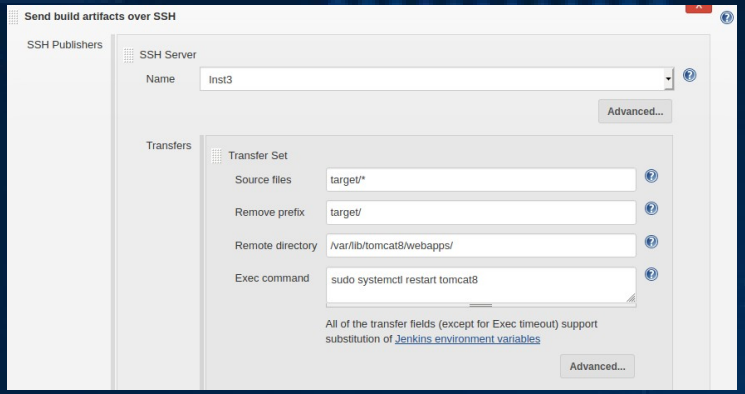
- Run Docker container with the desired version of software on production servers

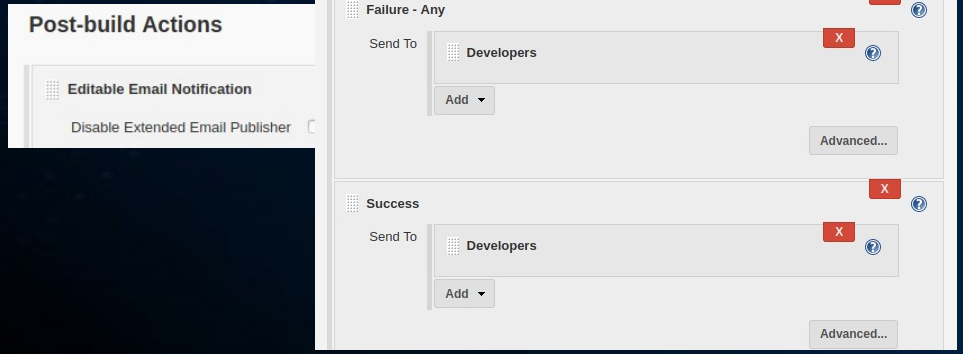


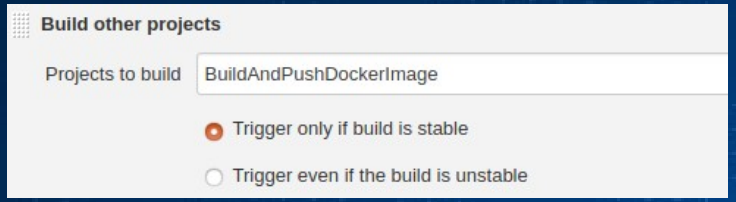
Jobs Implementation











Jobs Implementation

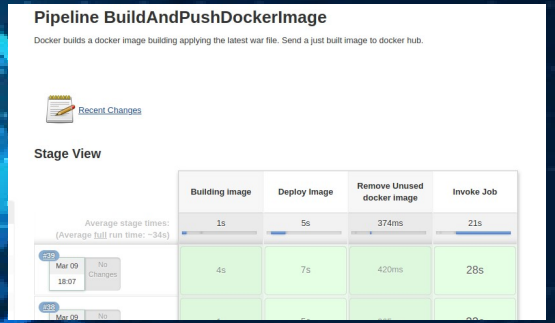
Console Output

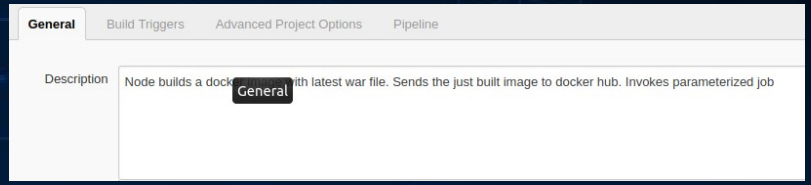
```
Started by user Student S T
Running as SYSTEM
Building remotely on Srv2 (TEST_SERVERS) in workspace /home/ubuntu/jenkins_slave/workspace/BookStore
using credential a876f452-8cd3-46c1-aec6-efe222f5168a
 > git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url git@github.com:serjr777/applet_deploy.git # timeout=10
Fetching upstream changes from git@github.com:serjr777/applet_deploy.git
 > git --version # timeout=10
using GIT_SSH to set credentials
 > git fetch --tags --progress -- git@github.com:serjr777/applet_deploy.git +refs/heads/*:refs/remotes/origin/* #
timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
 > git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
Checking out Revision 6d87ae67bb36eef36f45014c043f51abe2db3293 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 6d87ae67bb36eef36f45014c043f51abe2db3293 # timeout=10
Commit message: "Merge branch 'master' of github.com:serjr777/applet_deploy"
 > git rev-list --no-walk 6d87ae67bb36eef36f45014c043f51abe2db3293 # timeout=10
No emails were triggered.
[BookStore] $ /home/ubuntu/jenkins_slave/tools/hudson.tasks.Maven_MavenInstallation/Maven/bin/mvn -f /home/ubuntu
/jenkins_slave/workspace/BookStore/pom.xml clean install
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------< Bookstore:Bookstore >-------------------------
[INFO] Building App Maven Webapp 1.0-SNAPSHOT
```

```
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running ua.bookstore.BookTest
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.111 s - in ua.bookstore.BookTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-war-plugin:3.2.2:war (default-war) @ Bookstore ---
[INFO] Packaging webapp
[INFO] Assembling webapp [Bookstore] in [/home/ubuntu/jenkins_slave/workspace/BookStore/target/bookstore]
[INFO] Processing war project
```

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  6.070 s
[INFO] Finished at: 2020-03-09T16:07:12Z
[INFO] ------------------------------------------------------------------------
SSH: Connecting from host [inst2]
SSH: Connecting with configuration [Inst3] ...
SSH: EXEC: STDOUT/STDERR from command [sudo systemctl restart tomcat8] ...
SSH: EXEC: completed after 5,209 ms
SSH: Disconnecting configuration [Inst3] ...
SSH: Transferred 1 file(s)
Email was triggered for: Success
Sending email for trigger: Success
Sending email to: serjr777@gmail.com
Triggering a new build of BuildAndPushDockerImage
Finished: SUCCESS
```

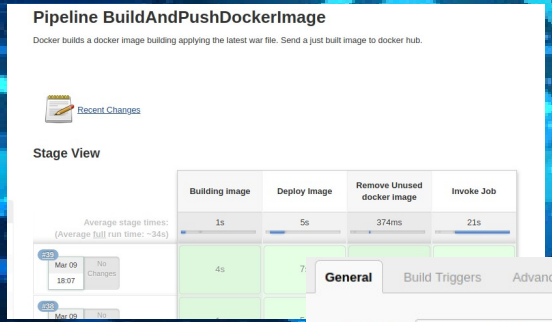
Jobs Implementation

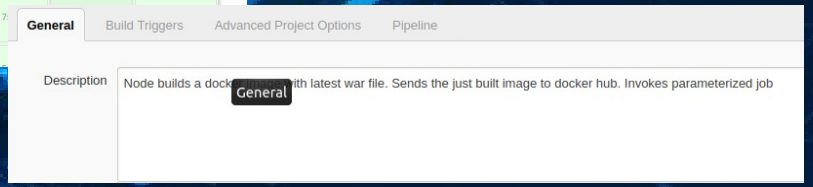





```
pipeline {
  environment {
    registry = "serjr777/bookstore"
    registryCredential = 'dockerhub'
    dockerImage = ''
  }
  agent { label 'TEST_SERVERS' }
  stages {
    stage('Building image') {
      steps{
        script {
          dockerImage = docker.build(registry + ":$BUILD_NUMBER", "/home/ubuntu/jenkins_slave/workspace")
        }
      }
    }

    stage('Deploy Image') {
      steps{
        script {
           docker.withRegistry( '', registryCredential ) {
           dockerImage.push()
          }
        }
      }
    }
    stage('Remove Unused docker image') {
      steps{
        sh "docker rmi $registry:$BUILD_NUMBER"
      }
    }

    stage('Invoke Job') {
      steps{
        build job: 'DeployToProduction',
        parameters: [
            string(name: 'bs_version', value: String.valueOf("${currentBuild.previousBuild.getNumber()}"))
        ]
      }
    }

  }
}
```

Jobs Implementation

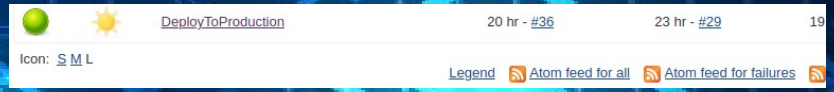


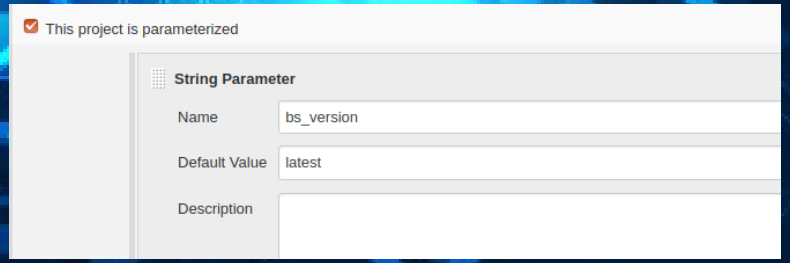


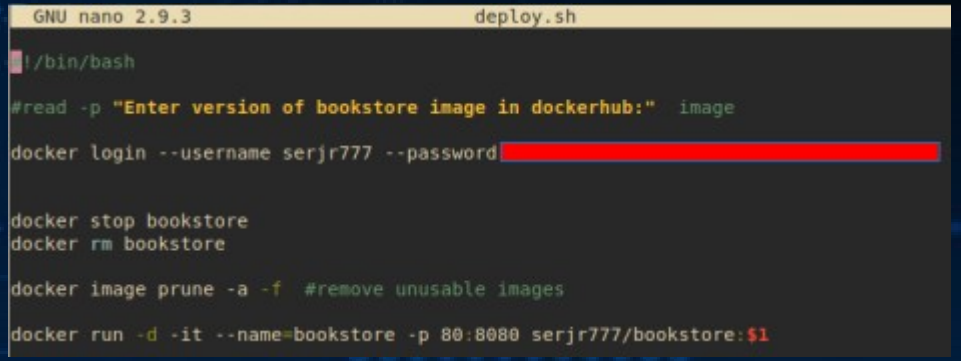

```
  GNU nano 2.9.3                          Dockerfile

# we are extending everything from tomcat:8.0 image ...
FROM tomcat:8.0

MAINTAINER sergey
# COPY path-to-your-application-war path-to-webapps-in-docker-tomcat
COPY BookStore/target/bookstore.war /usr/local/tomcat/webapps/
```

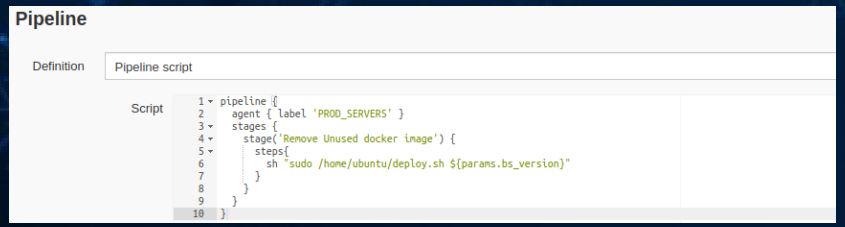
```
pipeline {
  environment {
    registry = "serjr777/bookstore"
    registryCredential = 'dockerhub'
    dockerImage = ''
  }
  agent { label 'TEST_SERVERS' }
  stages {
    stage('Building image') {
      steps{
        script {
          dockerImage = docker.build(registry + ":$BUILD_NUMBER", "/home/ubuntu/jenkins_slave/workspace")
        }
      }
    }

    stage('Deploy Image') {
      steps{
        script {
            docker.withRegistry( '', registryCredential ) {
            dockerImage.push()
          }
        }
      }
    }
    stage('Remove Unused docker image') {
      steps{
        sh "docker rmi $registry:$BUILD_NUMBER"
      }
    }

    stage('Invoke Job') {
      steps{
        build job: 'DeployToProduction',
        parameters: [
            string(name: 'bs_version', value: String.valueOf("${currentBuild.previousBuild.getNumber()}"))
        ]
      }
    }

  }
}
```

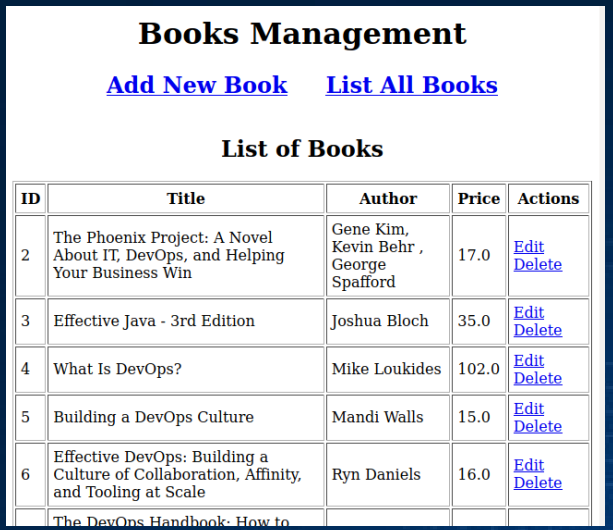
Jobs Implementation












Pros and Cons


Questions


Thank you !