

[day-6] [CI - master-slave - security - Email] ②

- * Jenkins ! [jenkins-setup-CI.sh]
- * Jenkins plug-in (First-job maven compile)
- * Poll SCM & web-hook & trigger SCM
- * Compile
- * Code Review - checking unfinished method
job - Review
 - ↳ maven plugin for [pmd & checkstyle] [Find Bugs]
 - ↳ DevOpsClassCodes
 - ↳ Build after other projects (Compile)
 - ↳ Build - invoke top level targets
Goals - P metrics pmd:pmd
- Workspace → review/ws/target/pmd.xml
 - Result → [Avoid Unused private fields] ↑ generated Report
- pmd.xml to HTML (jenkins pmd plugin for html generates)
- Post-build Actions (publish pmd Analysis results)
PMD Result | *** / pmd.xml
 - ↑ in workspace
- PMD warnings (After Post Actions)
 - Kill memory or RAM (during program mistake detector)
- * Sonar - Quality Gates (decision making for finding bugs)
 - ↳ code Review or Analyzer.
- { Sonar-Qube instead PMD }
- * Unit-tests - test each & every Units, whether code works or not.
 - Code Level testing
- Build after
 - ↳ review
 - Trigger build stable (success review)

- * Compile
- * CodeReview → pmd.xml → pmd.html
- * test

Workspace → targets

↳ surefire-plugin

(Run all test using this

plugin)
<maven-surefire-plugin>

<include> *.java</include>

— surefire-reports

↳ testReports.xml

* Post Build Actions

publish

* publish junit test result report

XMLS Report: 'my targets/surefire-reports/*.xml'

Unit-test → Latest Test Result

↳ All tests.

* CROOVY → {to write Jenkins jobs}

* metrics-check (code coverage) (tested as part of Unit-testing)

Build after: unit test.

Build steps: Goals: Cobertura: Cobertura
-Dcobertura1.result.xml

target → site | Cobertura → coverage.xml

XML File

Post Build Actions:

Publish Cobertura Coverage Report

xml : **/target/site/cobertura/cob.xml

* ↳ coverage reports.

Code Coverage: → Not Covered properly.

• Master Branch as a job running

* Package:

Build Action: maven package.

③ L target / addressbook, war

L .m2 (user home)

L .war file will be copied here.

* Pipeline Build (Run jobs in sequence,

SE

— See flow of your jobs)

+ Symbol

L my-ci

Continuous Integration

L Build pipeline view.

CI

— Select Initial job: compile

Compile — review — Unit-test — Package.

↓
Poll scm.

Yellow	- In progress
Green	- Success
Blue	- will Run
Red	- Fail

* Pipeline as Code.

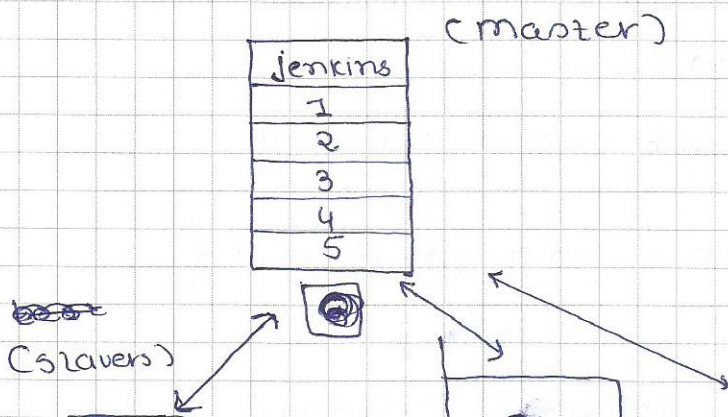
L

Master / Slave Config.

One Jenkins — n-number of people.

L may go down / heavy use.

* Jenkins Master / Slave Architecture.



* (Slave)

master

(should always there)

↳ Another Machine Ubuntu. (slave-name)

* java available on slave-machine

* Labs → Setup - user.sh

vi s.sh

> chmod 755 s.sh

↳ user 'devops' user creation
password 'jdokvans'

> ssh devops @ 18.221.68.57.0

> sudo su -

> cd /tmp /home/devops
mkdir savedata.

> manage → manage Nodes & clouds.

master →

> manage jenkins → Global Security

> Agents

↳ Random (For master & slave)

* Add slave. → Name 'slave1'

pwd → Path.

Remote root directory : Pwd

Labels : (Ravi)

Launch method: Launch agents via ssh

Host: internal IP address

Credentials:

← Add
jenkins
Registry

: Add username

: password

Host key Verification → Non Verifying

□ master

devops devops $\xrightarrow{\text{user}}$ slave data

(5)

*

slave job

↳ Build (execute shell)

command touch ravi

General

☒ Restrict where this project can be run

Label Expression ['ravi']

cd workspace → Ravi (Folder ravi file created)

master

→ jenkins

slave

(change color)

→ Only java install

→ install maven on slave.

* Node property.

↳ tool Locations

↳ ~~git~~ git

↳ env compile

↳ java (java-home)

↳ maven (maven-home)

[vi | etc | profile]

[Inside container]

↳ jenkins

* jenkins can download & Run ins