Continues Integration Automation Using Jenkins

Java and Open Source Competency

Tech Mahindra's Automation Program

TABLE OF CONTENT (Day-1)

Continuous Integration - Overview

Background:

Continuous Integration (CI) is a fundamental best practice of modern software development CI requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

Overview:

- Build automation in Jenkins achieved by running automated scripts (Batch / Ant / Maven) for projects.
- Scripts for most of the common tasks is auto generated and requires just writing script for uncommon, complex tasks only.
- Build for Windows, iOS, Android, Java (Ant, Maven, or Gradle), or Linux using the same domain-specific languages.
- Live console view in the web with real time status of each build.
- Easy customization - Edit in the web and leverage existing knowledge of popular script languages (Shell Script, Batch Script)

Continuous Integration - Benefits

Benefits:

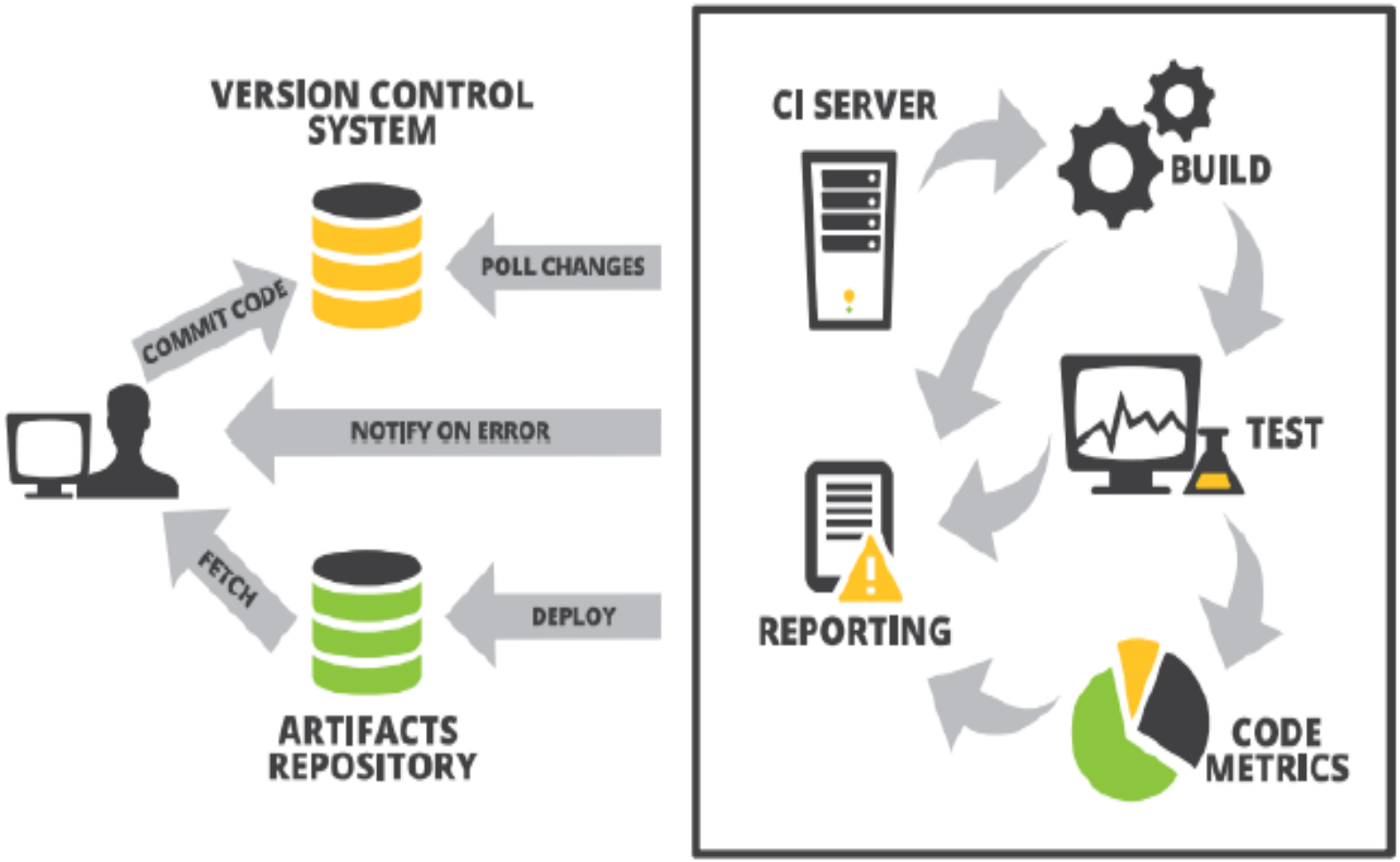
- Streamlined, consistent and reliable Builds.
- Saves time
- Easy to scale up or scale out build components across machines to get more build power
- Reduced dependency on individual resources
- Immediate Bug detection

Continuous Integration – Tools Recommended

- **Jenkins** – Open source CI tool
- **Scripts Automation using Shell Scripts, Batch Scripts** – Automating tasks using simple scripts
- **Build tools like Maven, Ant -** Open source tools for managing build structure
- **Subversion, CVS, Git** - Capability to work with various configuration management systems

Case Studies : CITI, Swiss Re, D+H, GE, ACRA, AT&T, Nissan, CMIC

CI Architecture



Aspects of CI

Tech Mahindra

Source Code Repository	• A source code repository is a file archive and web hosting facility and file are viewed and accessed publicly or privately. • **Tools:** CVS, SVN, GIT
Automated Build	• Build automation is the process of automating the creation of a software build and the associated processes including - compiling computer source code into binary code, packaging binary code, and running automated tests • **Tools:** Scripts, Ant, Maven
CI Server	• Continuous integration (CI) is the practice, in software engineering, of merging all developer working copies to a shared mainline several times a day • **Tools:** Jenkins
Automated Tests	• Test automation is the use of special software (separate from the software being tested) to control the execution of tests and the comparison of actual outcomes with predicted outcomes • **Tools:** Selenium
Automated Deployment	• Continuous Delivery (CD) is a software engineering approach in which teams keep producing valuable software in short cycles and ensure that the software can be reliably released at any time • **Tools:** Jenkins

Jenkins - Overview

- Jenkins is an open source continuous integration tool written in Java programming language for testing and reporting on isolated changes in a larger code base in real time.
- The software enables developers to find and solve defects in a code base rapidly and to automate testing of their builds.
- CI is a development practice that requires developers/testers to integrate their work frequently into a shared repository several times a day.
- Each integration is performed by automated build
- Building / Testing of software projects continuously
- Monitoring of externally executed jobs
- Hudson was first released by Sun Microsystems in the year of 20025
- Jenkins is a fork of a project called Hudson, which is trademarked by Oracle and is currently being developed parallel to Jenkins.

Jenkins - Feature List

- Easy install, upgrade and configure.
- Check out source code:
  - CVS, Subversion, Clear Case, Mercurial, Accurev, Perforce, Git
- Perform builds using
  - Ant, Maven, shell script, NAnt
- Record and publish results.
- HTTP API.
- Open Source and Commercial Tools integration:
  - Find Bugs, Cobertura, Emma
- Plugin support, update centre, 400+ plugins
- Authentication Using
  - LDAP, AD, Open ID, SQL
- Testing using
  - Selenium, Sauce, Windmill
- Master + slave nodes
- Start Master-only, add slaves later
- Slaves can be on different platforms
- Allows build/testing on different OS

Jenkins - Installation

Pre-requisites:

- To run Jenkins on a Windows / Linux machine need to install the Java 6/ 7 and above

Downloading Jenkins:

- Download Jenkins war file from http://jenkins-ci.org/

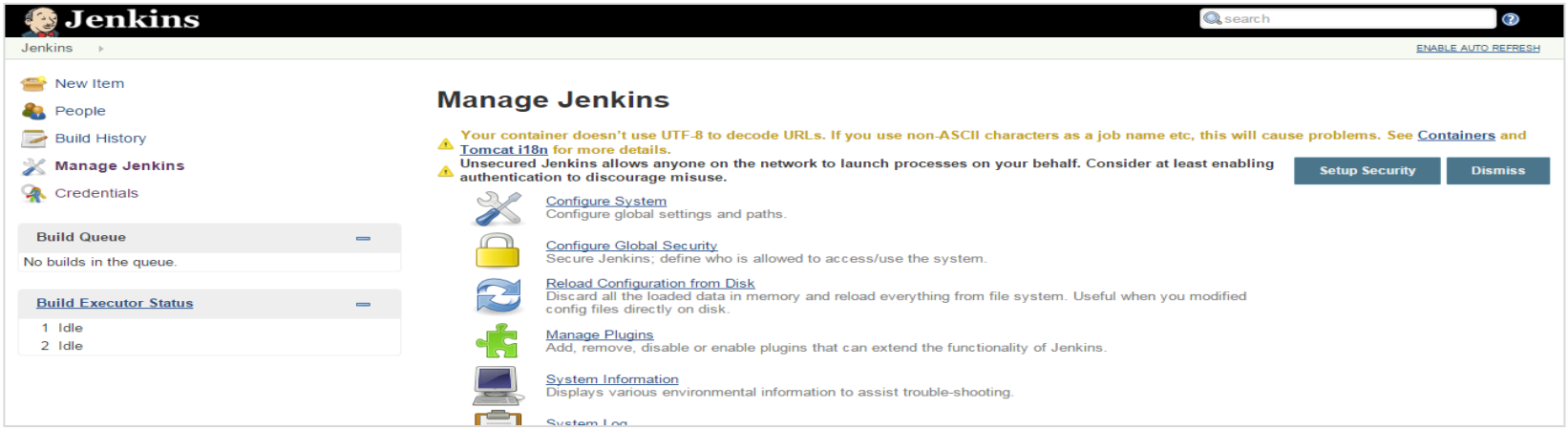
Running Jenkins:

- Execute war file using
  C:/>java –jar jenkins.war

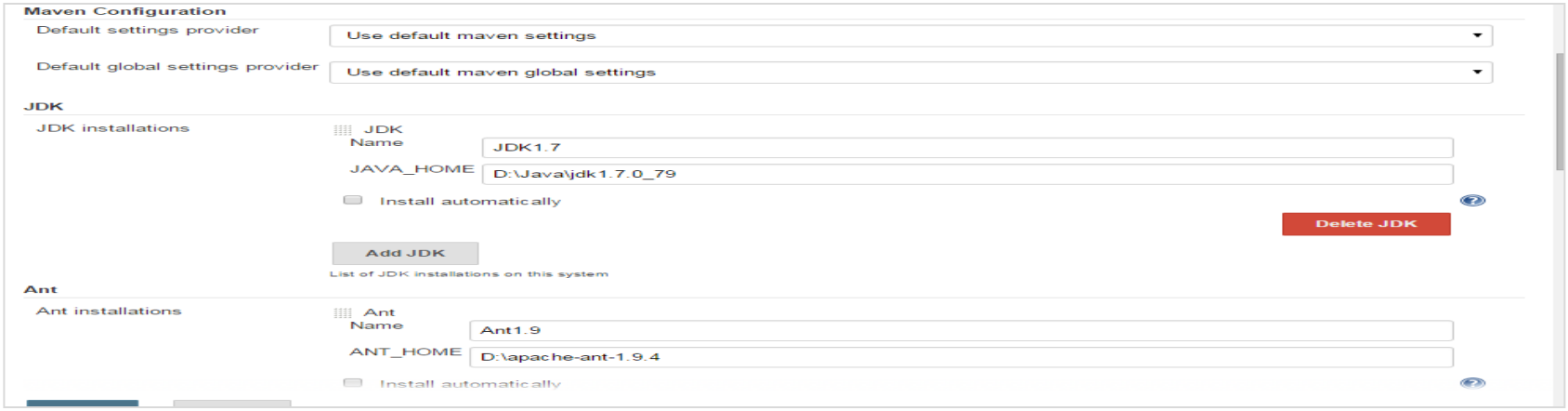
  (OR)
- Install tomcat and deploy the jenkins.war file in [Tomcat]/webapps folder and run the tomcat server by running the file in [tomcat]/bin/startup.bat file
- Open the Jenkins browser using URL in any browser
  http://localhost:8080/jenkins

Jenkins - Configuration

Manage Jenkins page - Central one-stop-shop for all Jenkins configuration. From this page configure Jenkins server, install and upgrade plugins, keep track of system load, manage distributed build servers can be performed.

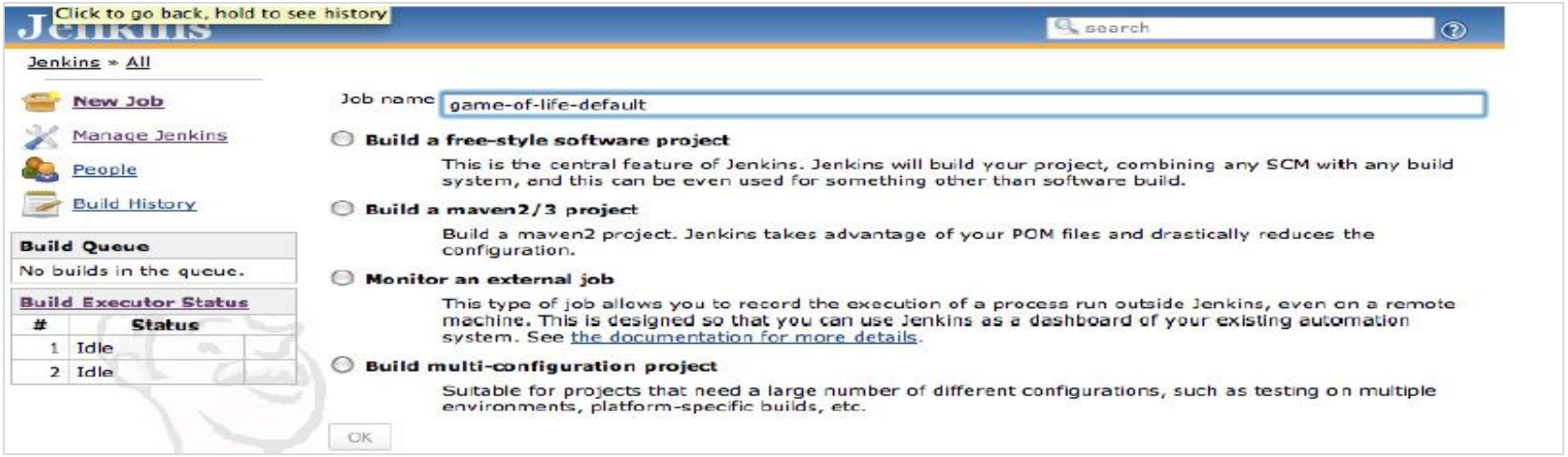


Configure Jenkins screen – Configuring configurations JDK, Ant, Maven, Global Properties, Notifications, Git and security options.

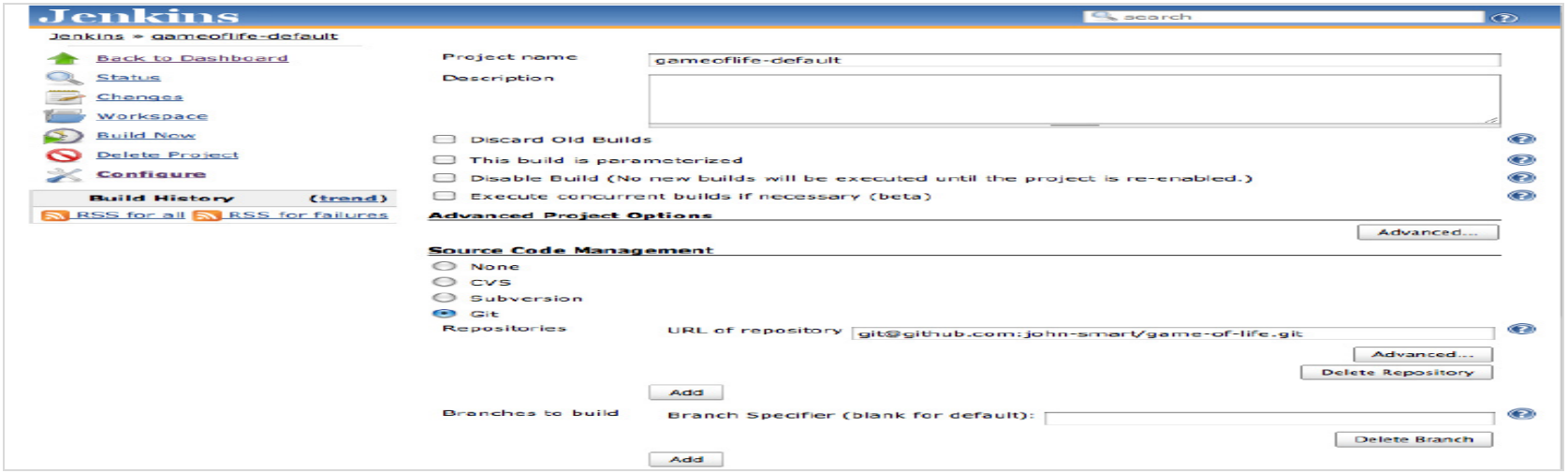


Jenkins - Setting First Job

First Job - Build jobs are at the heart of the Jenkins build process



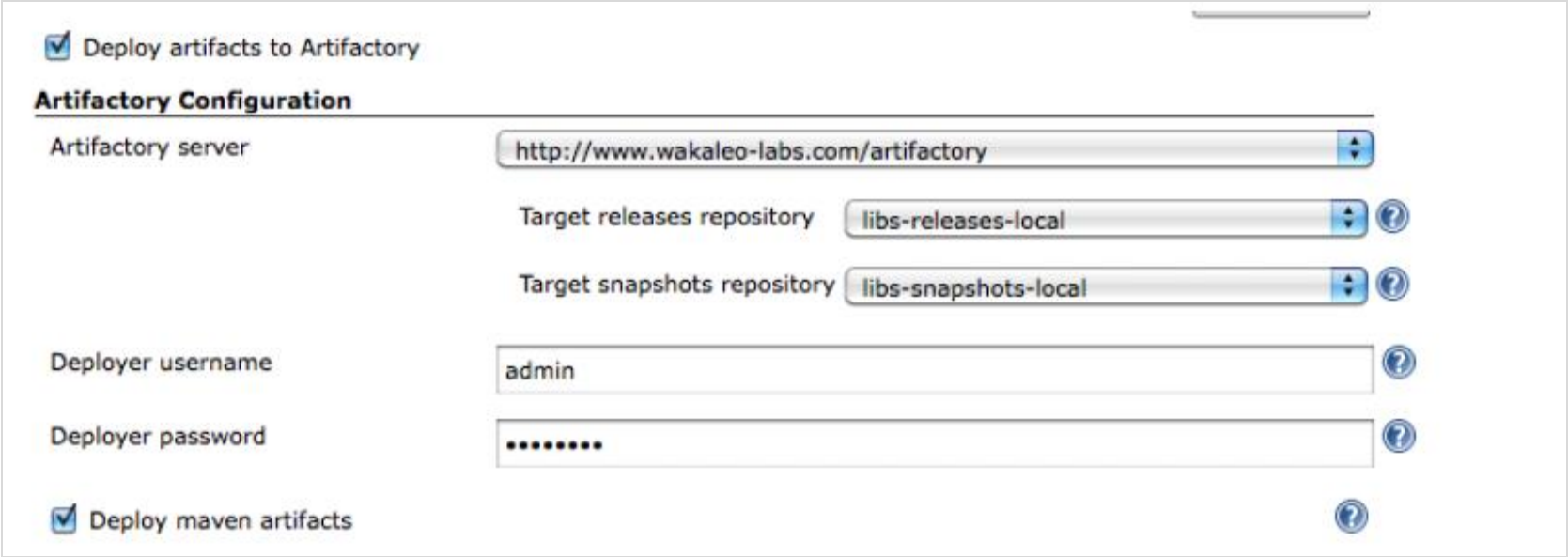
Configure Jenkins Job- Screen to configure the job where to get the source code and how to execute scripts in sequential process.



Jenkins – Configurations (Email)

E-Mali configuration

Artifactory Configuration

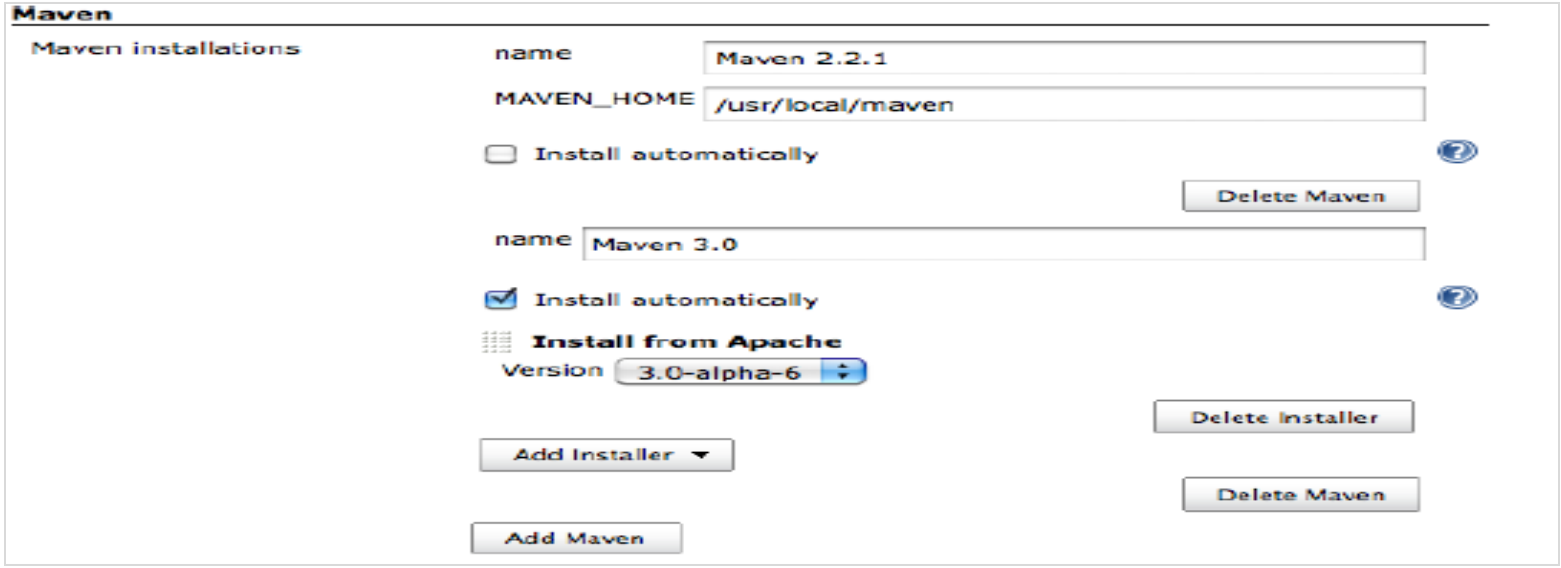


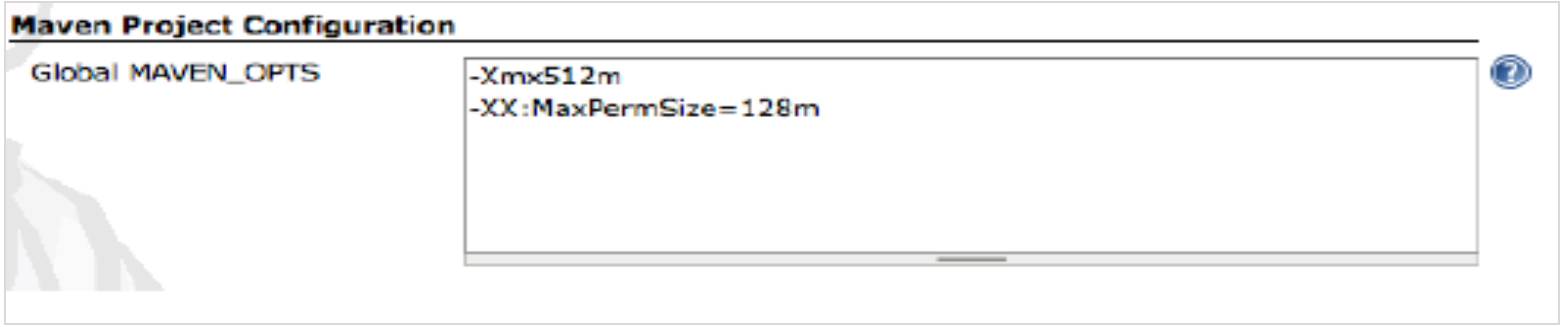
Jenkins – Configurations (Build Tools)

Tech Mahindra

Build tools are the bread-and-butter of any build server, and Jenkins is no exception. Jenkins supports three principal build tools: Ant, Maven, and the basic shell-script

Maven:





Jenkins – Configurations (Build Tools)

Build tools are the bread-and-butter of any build server, and Jenkins is no exception. Jenkins supports three principal build tools: Ant, Maven, and the basic shell-script

Ant:

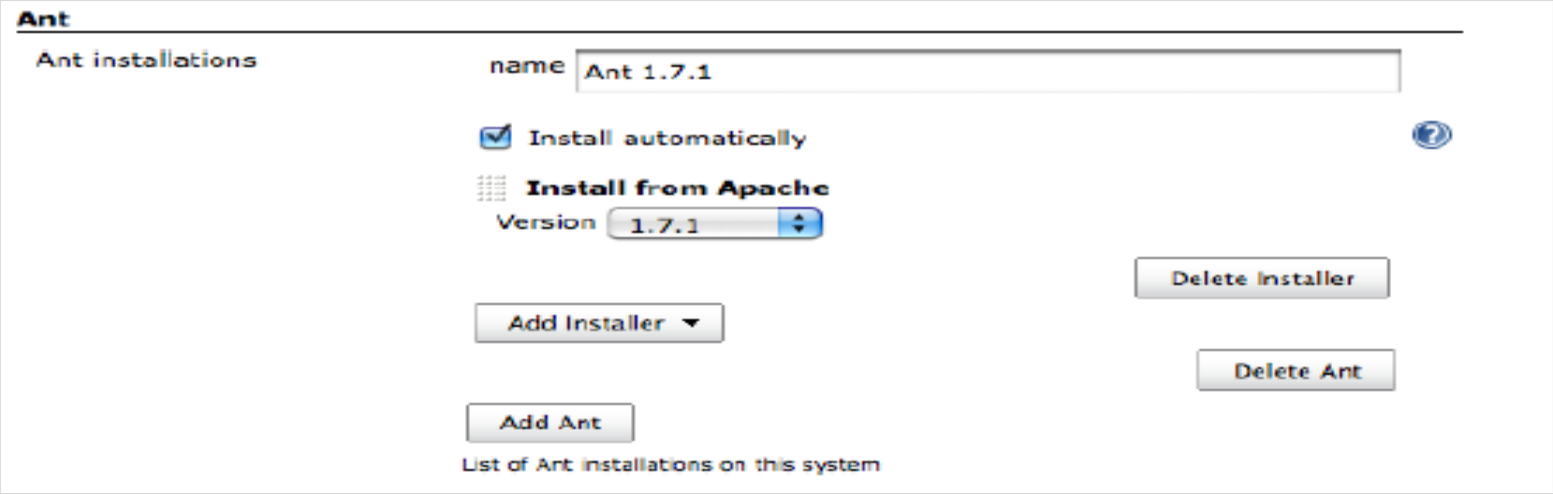


TABLE OF CONTENT (Day-2)

Jenkins - SonarQube Process Flow

Tech Mahindra

Sonar

- Install Sonar and Sonar Runner
- Create Users and Groups in Sonar server using administration login
- Assign projects to the sonar by using Sonar runner
- Assign Rules for the project
- Assign Users and Groups for the project to access

Jenkins

- Install Jenkins
- Add Sonar plugin in Jenkins plugin
- Create people OR assign LDAP to the Jenkins
- Configure Sonar and Sonar Runner in Jenkins
- Assign jobs to the people for execution
- Create Job and assign Sonar runner / ANT / Maven task to execute sonar rules

Sonar Reports

- Login to Sonar server with user details
- View the project code review report in Sonar Dashboard

Jenkins - SonarQube Configurations and Running

Tech Mahindra

Sonar

Sonar installations

Name	Sonar8080
Disable	Check to quickly disable Sonar on all jobs.
Server URL	http://localhost:8080/sonar Default is http://localhost:9000
Sonar account login	admin Sonar account used to perform analysis. Mandatory since Sonar 3.4 when anonymous access is disabled.
Sonar account password	••••• Sonar account used to perform analysis. Mandatory since Sonar 3.4 when anonymous access is disabled.
Database URL	Do not set if default embedded database.
Database login	Default is sonar.
Database password	Default is sonar.
Database driver	Do not set if you use the default embedded database on localhost.
Version of sonar-maven-plugin	If not specified, then sonar:sonar will be used.
Additional properties	Additional properties to be passed to the mvn executable (example : -Dsome.property=some.value)

Sonar Server Configuration Details where the SonarQube is running from Jenkins Configuration

Sonar Runner

Sonar Runner installations

Sonar Runner

Name: Sonar Runner

SONAR_RUNNER_HOME: D:\appservers\sonar\sonar-runner-2.3

Install automatically

Delete Sonar Runner

Add Sonar Runner

List of Sonar Runner installations on this system

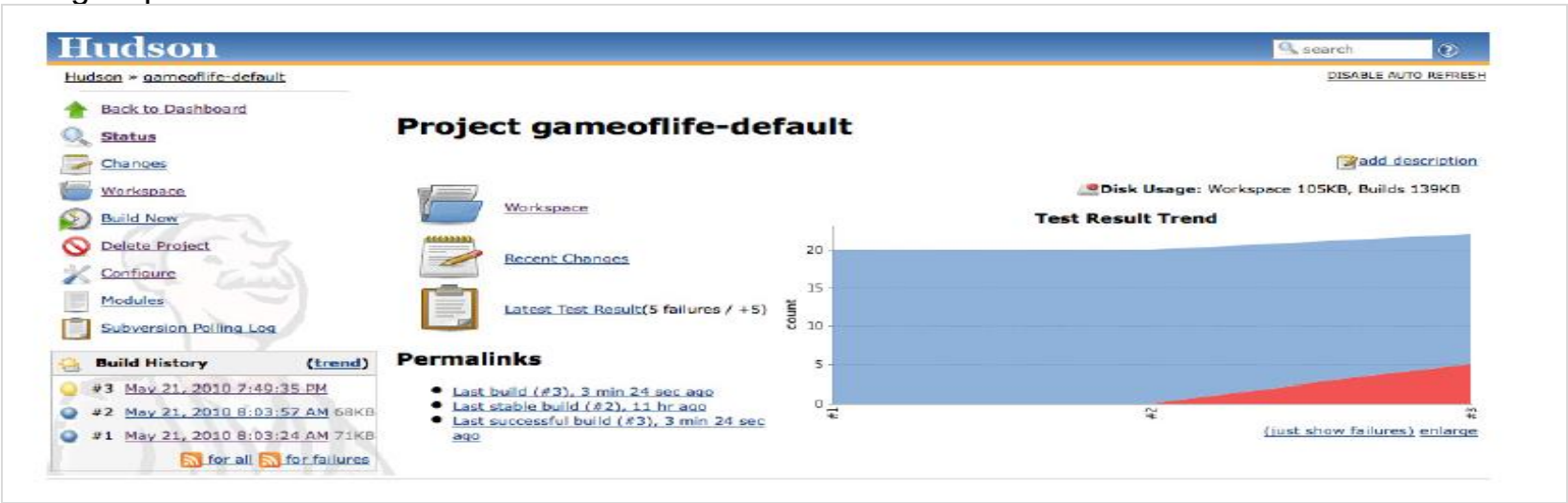
Sonar Runner Configuration where the Sonar Runnier is available from Jenkins Configuration

Jenkins – Testing Configuration

Tech Mahindra

Automated Unit Testing

Testing Report



Jenkins - Code Coverage

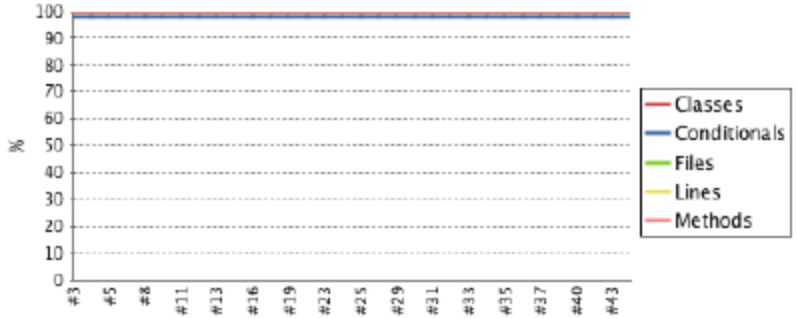
Code Coverage Report:

Code Coverage

Cobertura Coverage Report >

com.wakaleo.gameoflife.domain

Trend



Package Coverage summary

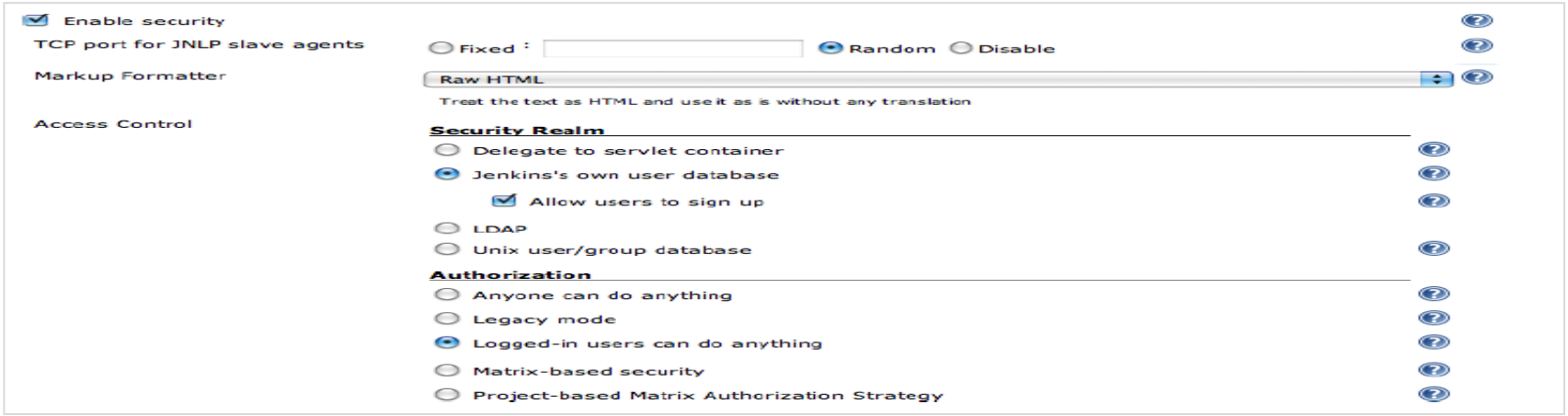
Name	Classes	Conditionals	Files	Lines	Methods
com.wakaleo.gameoflife.domain	100% 5/5	98% 53/54	100% 5/5	100% 108/108	100% 36/36

Coverage Breakdown by File

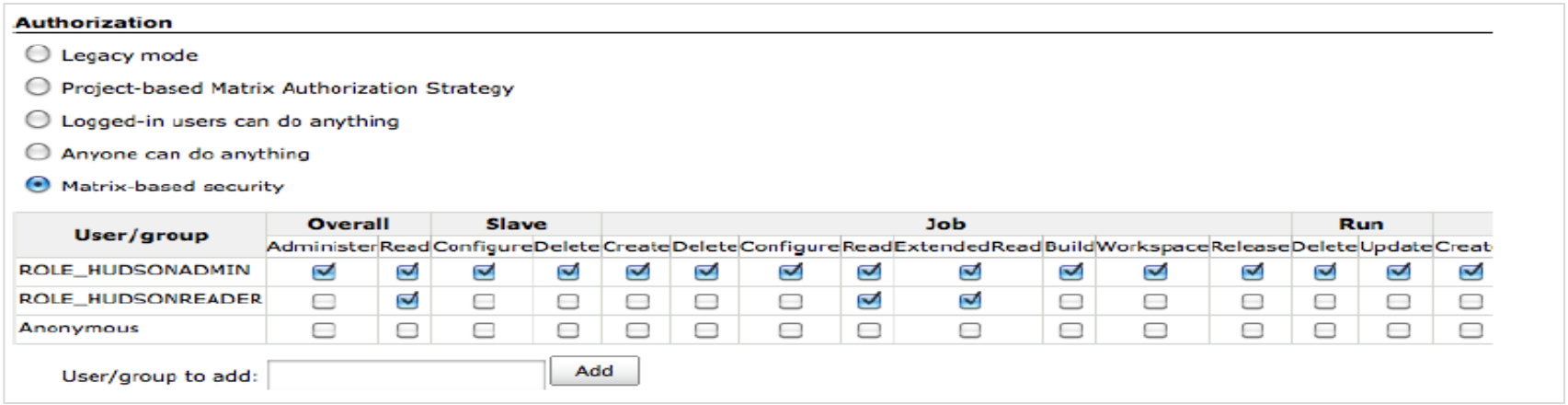
Name	Classes	Conditionals	Lines	Methods
Grid.java	100% 1/1	100% 30/30	100% 40/40	100% 16/16
Cell.java	100% 1/1	75% 3/4	100% 14/14	100% 5/5
GridWriter.java	100% 1/1	100% 4/4	100% 7/7	100% 2/2
Universe.java	100% 1/1	100% 12/12	100% 34/34	100% 9/9
GridReader.java	100% 1/1	100% 4/4	100% 13/13	100% 4/4

Jenkins - Security

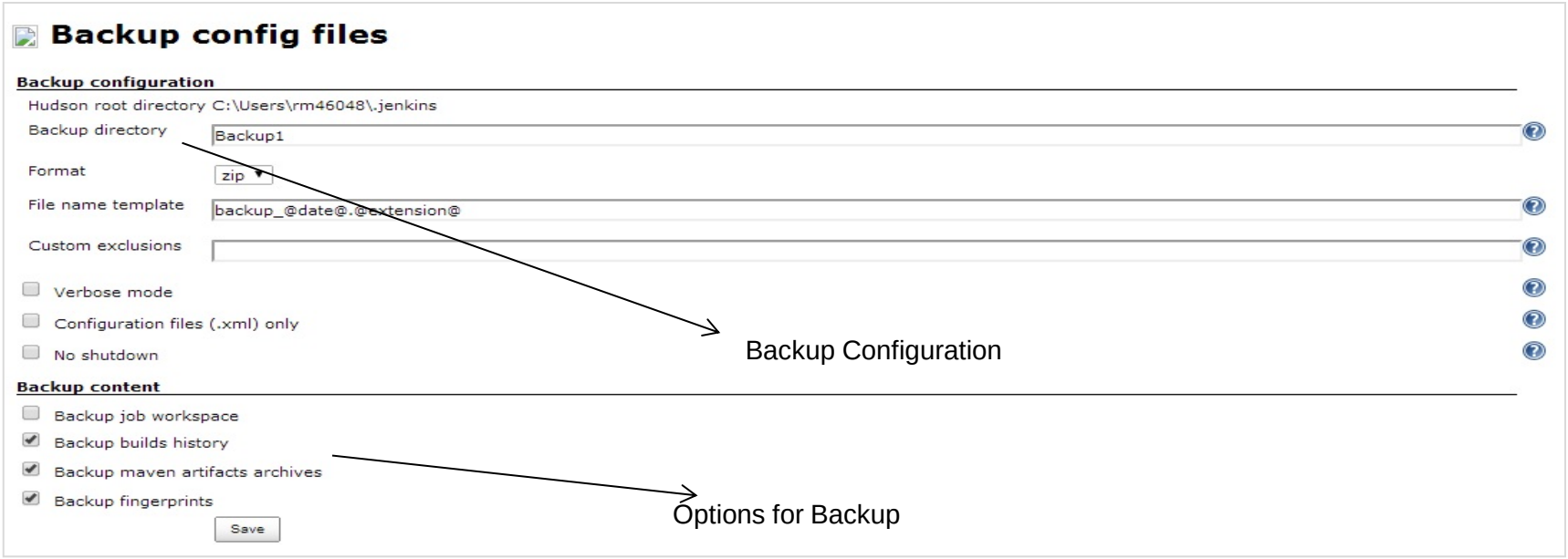
The first section, Security Realms, determines where Jenkins will look for users during authentication, and includes options such as using users stored in an LDAP server, using the underlying Unix user accounts:



Authorization:



Jenkins - Backup

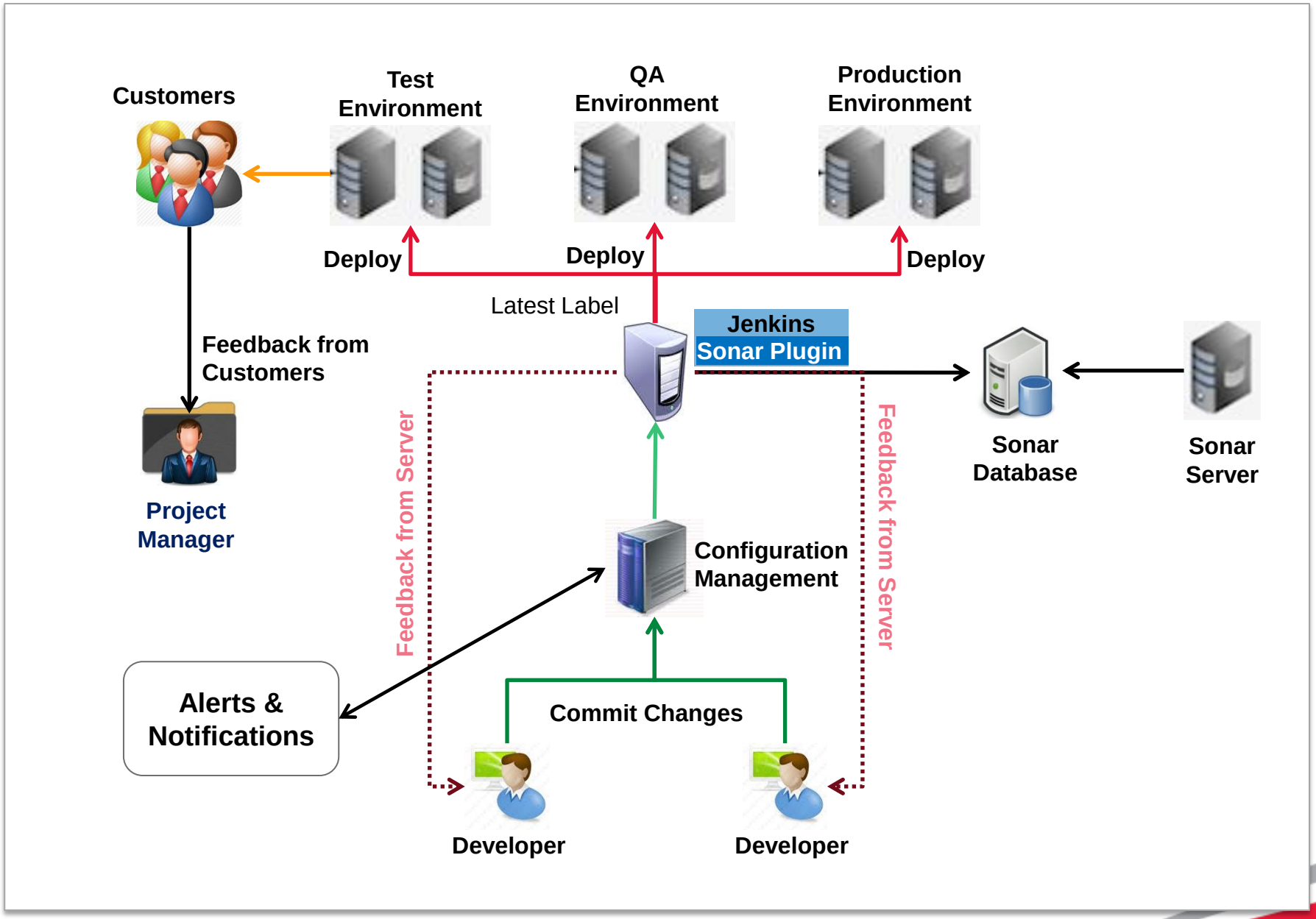


Backup Configuration

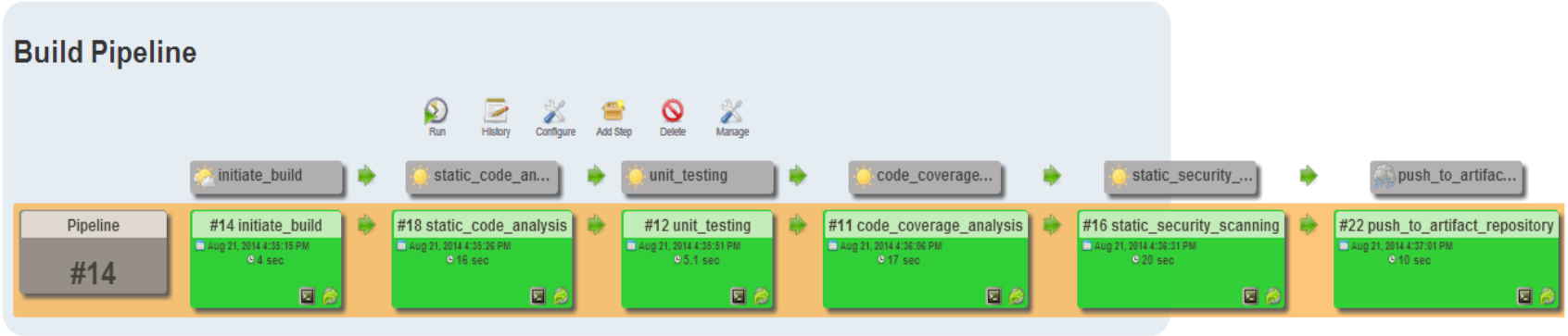
Options for Backup

Backup and Restore options

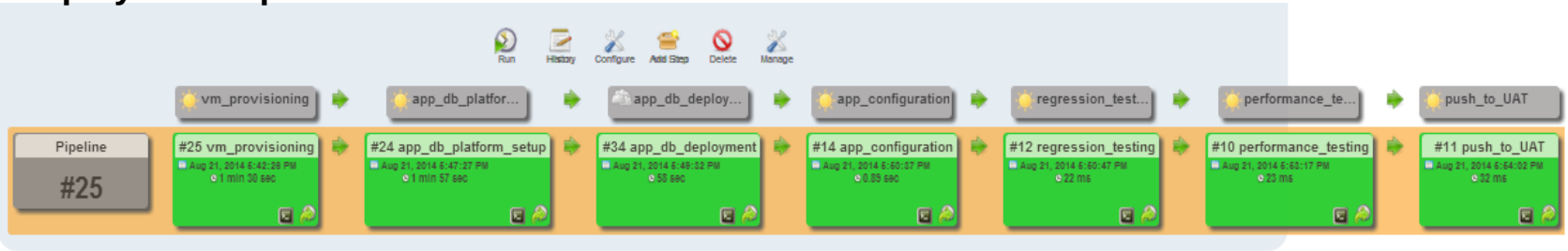
Automated Deployment



Automated Deployment – Build Pipeline



Deployment Pipeline for Test Environment



Deployment Pipeline for Staging Environment

Jenkins - Distributed Builds

Jenkins uses a master/slave architecture to manage distributed builds.

In a nutshell, the master's job is to handle scheduling build jobs, dispatching builds to the slaves for the actual execution, monitor the slaves and recording and presenting the build results.

Even in a distributed architecture, a master instance of Jenkins can also execute build jobs directly.

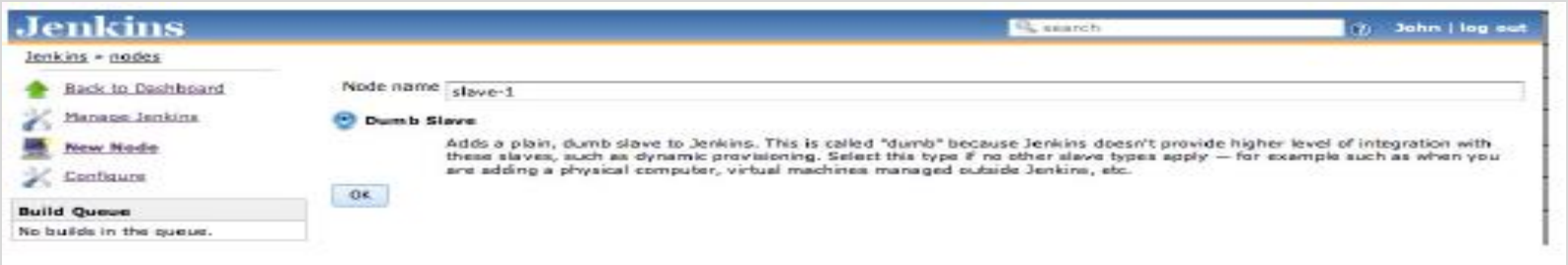
The job of the slaves is to do as they are told, which involves executing build jobs dispatched by the master.

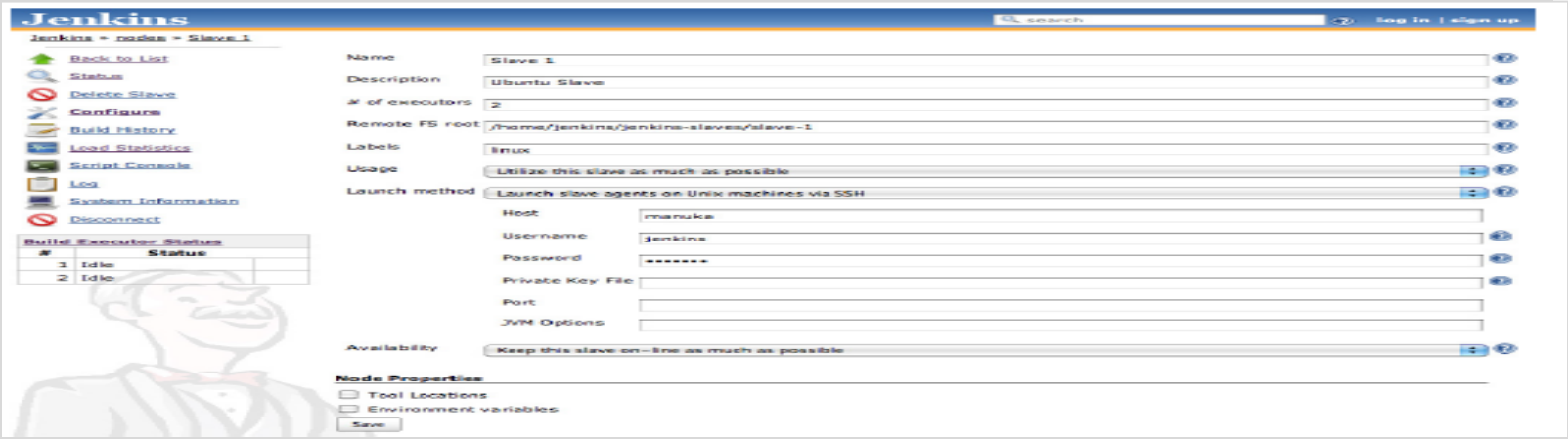
Jenkins - Distributed Builds (Contd..)

Tech Mahindra

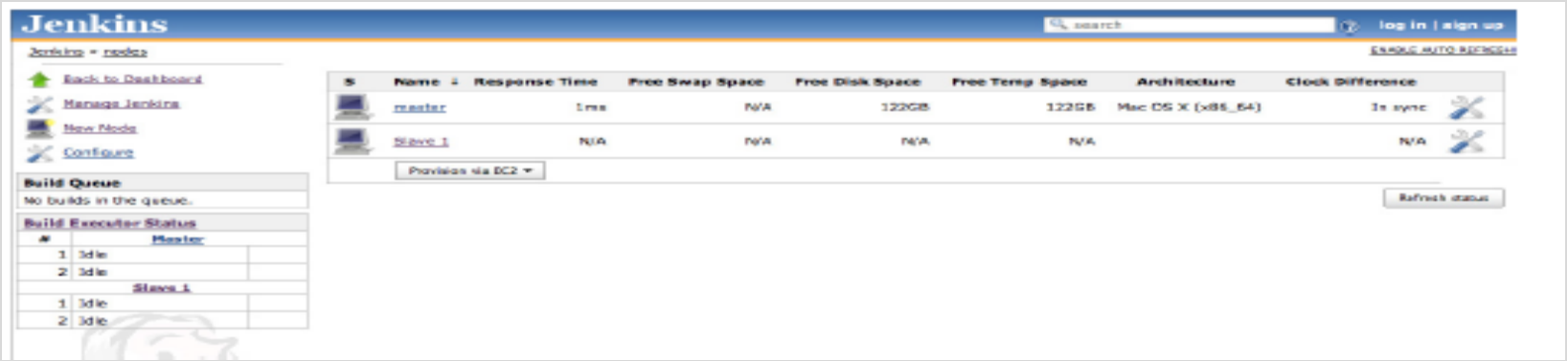
Several different strategies when it comes to managing Jenkins slave nodes, depending on target operating systems and other architectural considerations. The following are the different strategies.

- Master starts the slave agents via SSH
- Starting the slave agent manually using Java Web Start
- Installing the slave agent as a Window service
- Starting the slave agent directly from the command line on the slave machine from the command line

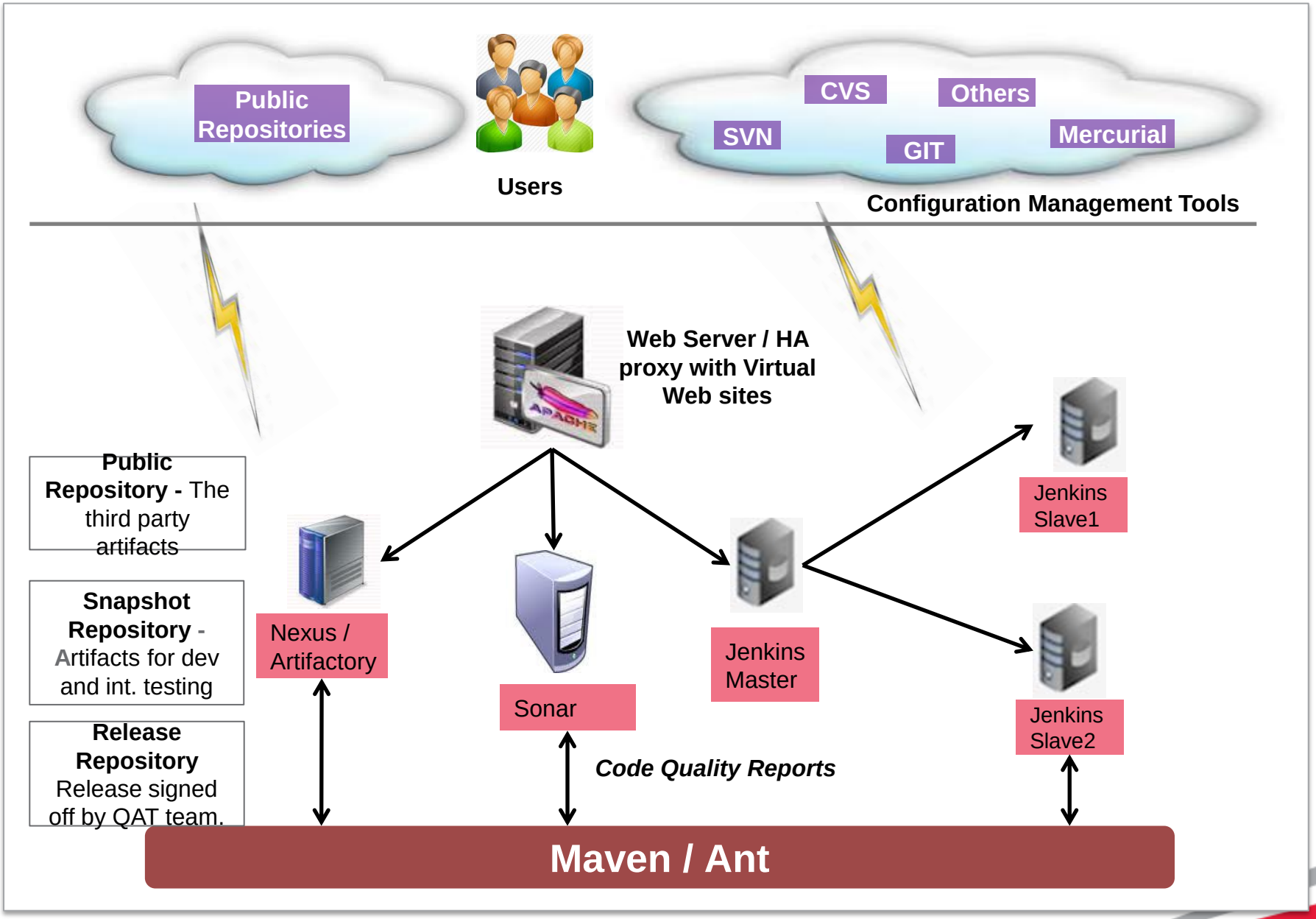




Jenkins - Distributed Builds (Contd..)



Jenkins - Master Slave Architecture



Tools and Accelerators

Tech Mahindra

M-IDE : Developer Workbench

- **Configuration Management**
  - Subversion, CVS
- **Build Management**
  - Maven for Large projects
  - Ant for Smaller projects
- **Code Quality**
  - PMD, CheckStyle, FindBugs using Developer IDE
  - Sonar using Developer IDE and a centralized Sonar Server
- **Unit Testing**
  - Unit Testing
  - EclEmma (Code Coverage)
  - eCobertura (Code Coverage)

M-CI : CI Workbench

- **Continuous Integration**
  - Jenkins
- **Configuration Management**
  - Subversion
  - CVS
- **Project Structure and Build Management**
  - Maven
- **Code Quality Assessment**
  - Sonar
- **Test Automation and Execution**
  - JUnit, Selenium, TPTP Test Cases
  - EMMA for Code Coverage
- **Defect Tracking**
  - Bugzilla

M-TEST : Tester Workbench

- **Test Planning**
  - TPTP for Test Case Suite Management
- **Test Case Preparation and Execution**
  - Selenium , TPTP, TestNG
  - WebLoad and JMeter for Load Testing
  - TPTP for Code Profiling
- **Defect Tracking**
  - Bugzilla

Reusable Components

- Exception Handling and Logging
- LDAP Integration, Email Integration
- Document Management – Jackrabbit
- Barcode Generator, Captua Generator
- Ehcache + Terracotta for cache distribution
- Image Scanner, Image Viewer – Java Applet
- MQ crusting scripts on Glass Fish Server.
- Free lance calendar

QoS Aids

- Architecture Review Templates, Capacity Planning Templates, High Availability solutions, Performance Engineering Methodology and Tools, Security Assessment
- Agile Development Templates

Reference Implementations

- Spring-Hibernate, JSF-Spring-Hibernate, GWT-Spring-Hibernate LDAP Integration
- jBPM and BRMS, Mule

Thank You

Disclaimer

Tech Mahindra, herein referred to as TechM provide a wide array of presentations and reports, with the contributions of various professionals. These presentations and reports are for informational purposes and private circulation only and do not constitute an offer to buy or sell any securities mentioned therein. They do not purport to be a complete description of the markets conditions or developments referred to in the material. While utmost care has been taken in preparing the above, we claim no responsibility for their accuracy. We shall not be liable for any direct or indirect losses arising from the use thereof and the viewers are requested to use the information contained herein at their own risk. These presentations and reports should not be reproduced, re-circulated, published in any media, website or otherwise, in any form or manner, in part or as a whole, without the express consent in writing of TechM or its subsidiaries. Any unauthorized use, disclosure or public dissemination of information contained herein is prohibited. Unless specifically noted, TechM is not responsible for the content of these presentations and/or the opinions of the presenters. Individual situations and local practices and standards may vary, so viewers and others utilizing information contained within a presentation are free to adopt differing standards and approaches as they see fit. You may not repackage or sell the presentation. Products and names mentioned in materials or presentations are the property of their respective owners and the mention of them does not constitute an endorsement by TechM. Information contained in a presentation hosted or promoted by TechM is provided “as is” without warranty of any kind, either expressed or implied, including any warranty of merchantability or fitness for a particular purpose. TechM assumes no liability or responsibility for the contents of a presentation or the opinions expressed by the presenters. All expressions of opinion are subject to change without notice.

CI Workbench Open Source Tools