

CI/CD Workshop



Jenkins



Configuration Jenkins



Configuration !!

Manage Jenkins -> Global Tool Configuration

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there is a sidebar with several options: New Item, People, Build History, Manage Jenkins (which is circled with a red number 1), My Views, and Credentials. Below this is a Build Queue section indicating "No builds in the queue." At the bottom is a Build Executor Status section. On the right, the main content area is titled "Manage Jenkins" and lists several configuration options: Configure System, Configure Global Security, Configure Credentials (which is circled with a red number 2), Global Tool Configuration (which is also circled with a red circle), and Reload Configuration from Disk.

Jenkins

New Item

People

Build History

1

Manage Jenkins

My Views

Credentials

Build Queue

No builds in the queue.

Build Executor Status

Manage Jenkins

Configure System

Configure global settings and paths.

Configure Global Security

Secure Jenkins; define who is allowed to access/use

Configure Credentials

Configure the credential providers and types

2

Global Tool Configuration

Configure tools, their locations and automatic instal

Reload Configuration from Disk

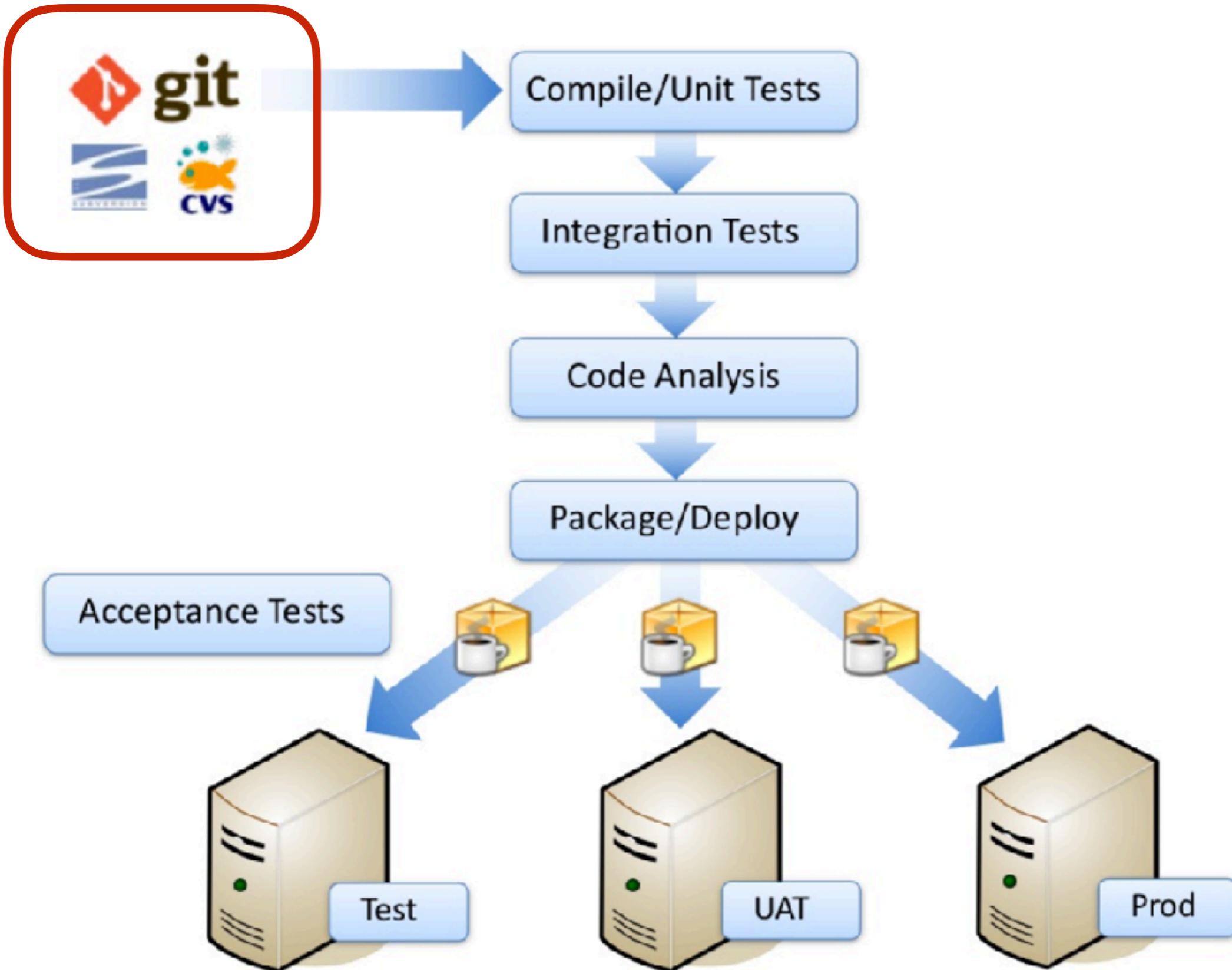
Discard all the loaded data in memory and reload ev

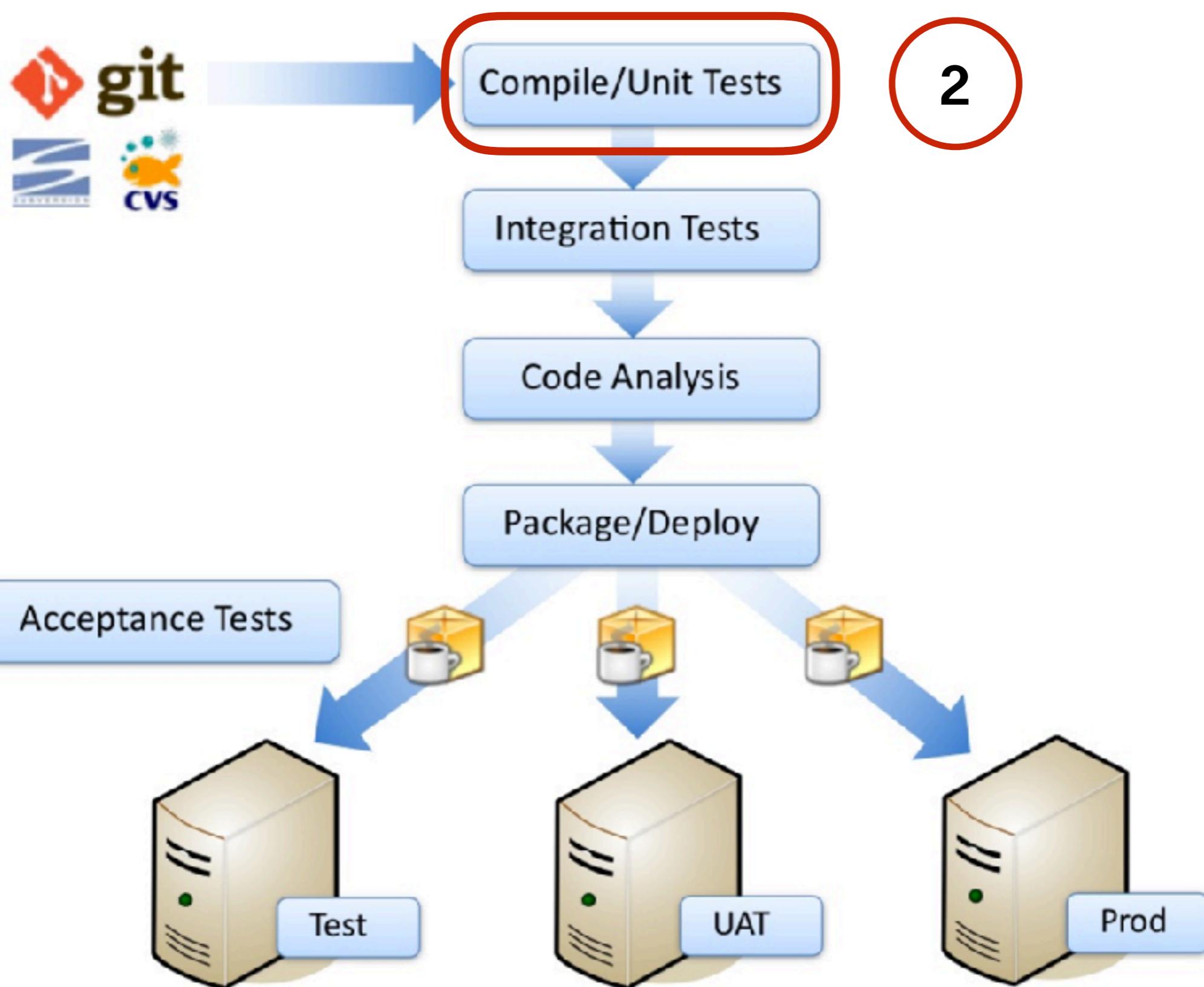


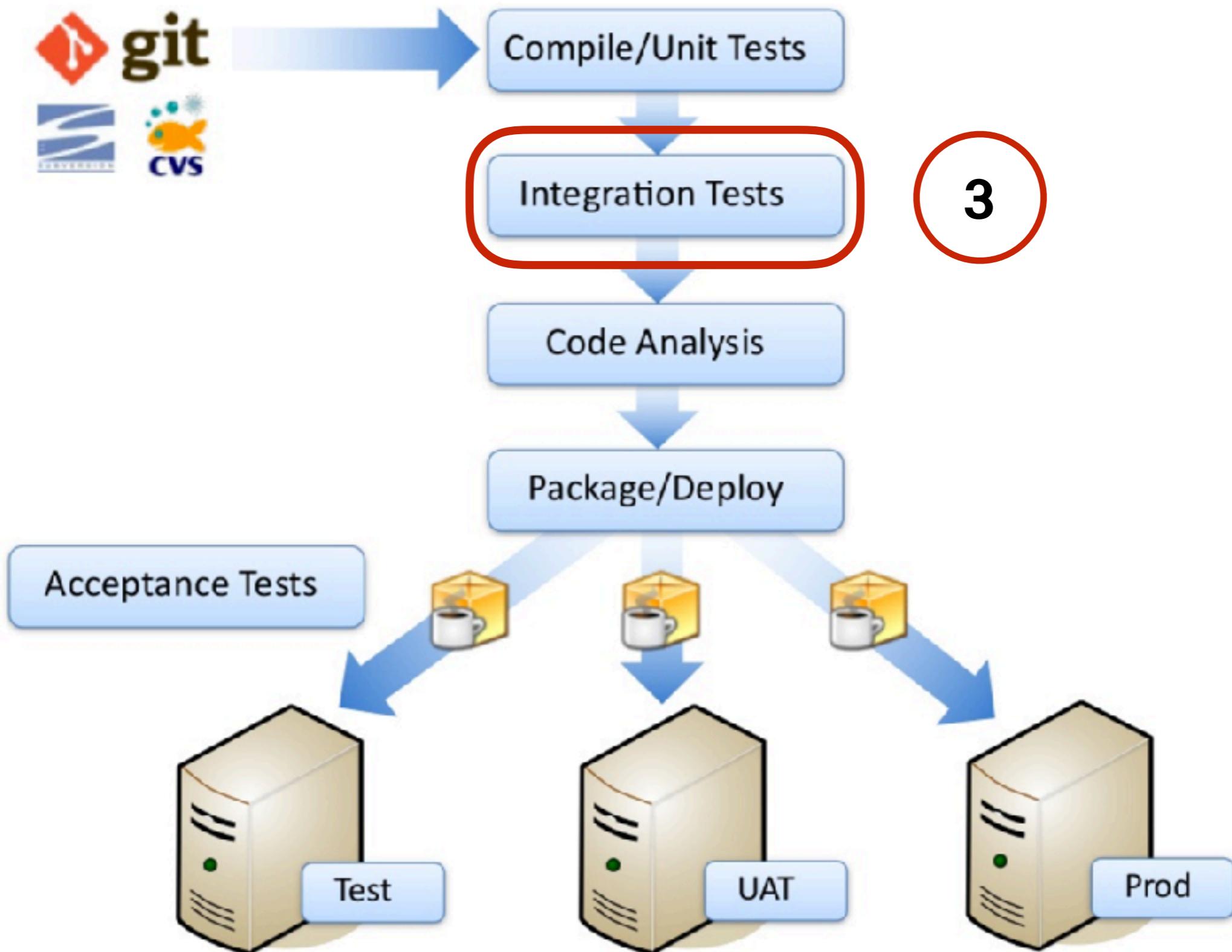
Let's start with Jenkins

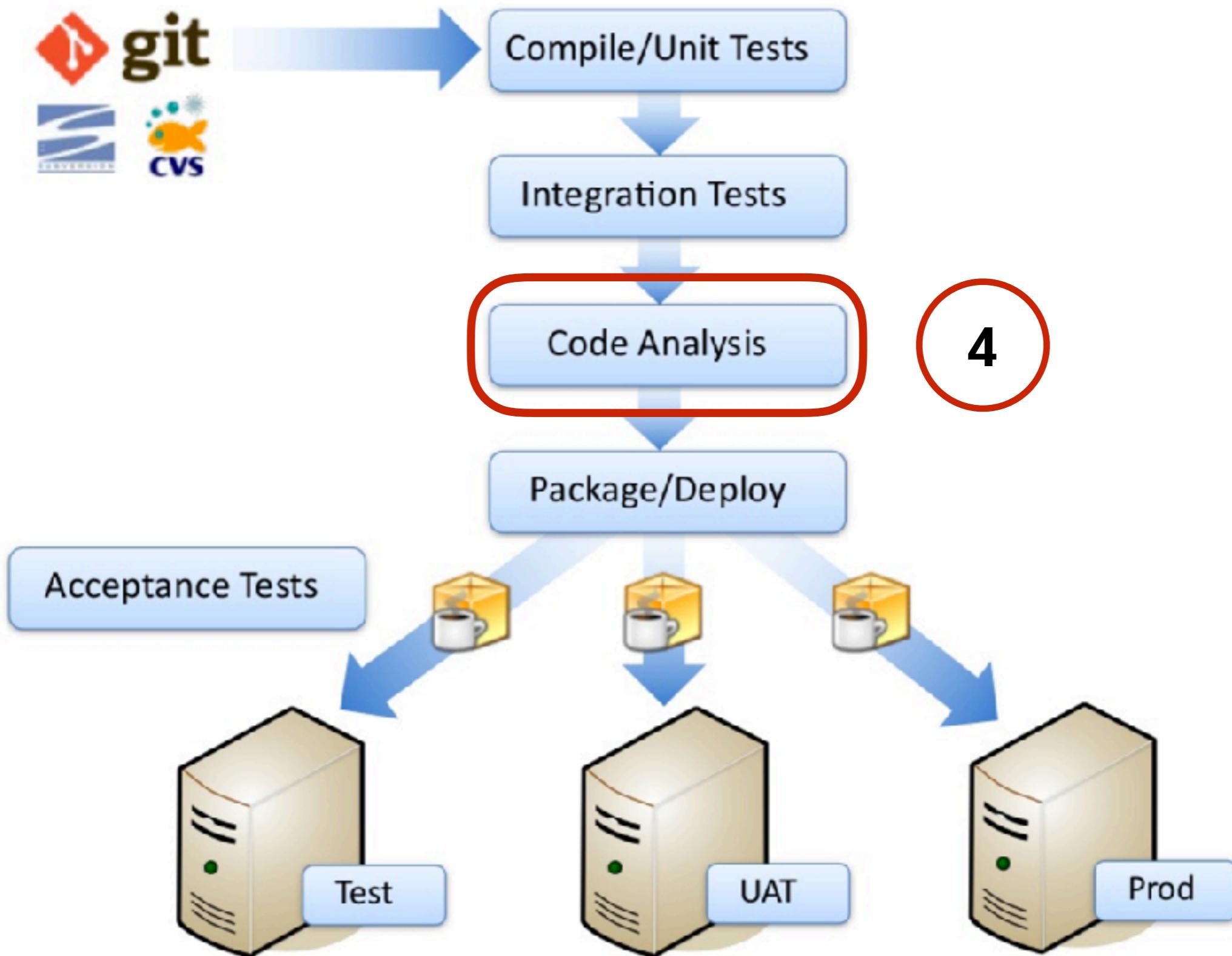


1









Code analysis with SonarQube



Setup SonarQube

Download from sonarqube.org

The screenshot shows the official SonarQube website. At the top, there's a navigation bar with links for FEATURES, DOWNLOADS, ROADMAP, COMMUNITY, and BLOG. The main headline reads "The leading product for CONTINUOUS CODE QUALITY". Below this, there are three cards: "Code Smells" (radioactive symbol icon), "Bugs" (bug icon), and "Vulnerabilities" (padlock icon). A "Used by more than 80,000 organizations" badge is visible. At the bottom, there are links for "ONLINE DOWNLOAD" and "USE ONLINE", along with a list of supported languages: Java, JavaScript, C#, C/C++, COBOL, and "AND MORE".



Install SonarQube Scanner plugin

Filter:

Updates Available Installed Advanced

Install ↓	Name	Version
<input checked="" type="checkbox"/>	SonarQube Scanner for Jenkins	2.6.1
<input type="checkbox"/>	Mashup Portlets Additional Dashboard Portlets: Generic JS Portlet (lets you pull in arbitrary content via JS), Recent Changes Portlet (shows the SCM changes for a given job), SonarQube Portlets (show SonarQube statistics directly in Jenkins) and Test Results Portlet (shows the test results for a given job).	1.0.8

[Install without restart](#) [Download now and install after restart](#) [Update information obtained](#)



Download SonarQube Scanner

Scanners / Analyzing Source Code / Analyzing with SonarQube Scanner

Analyzing with SonarQube Scanner

Created by OLD - Evgeny Mandrikov, last modified by Julien Henry on May 12, 2017

By SonarSource – GNU LGPL 3 – Issue Tracker – Sources

Download SonarQube Scanner 3.0.3

Compatible with SonarQube 5.6+ (LTS)

[Linux 64 bit](#) [Windows 64 bit](#) [Mac OS X 64 bit](#) [Any*](#)

*This package expects that a JVM is already installed on the system - with same Java requirements as the SonarQube server.

Table of Contents

- Features
- Installation
- Use
- Troubleshooting
- Going Further

<https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner>



Config SonarQube Server

Manage Jenkins -> Configure System

SonarQube servers

Environment variables

Enable injection of SonarQube server configuration as build environment variables
If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

SonarQube installations

Name: sonar server

Server URL: http://localhost:9000

Default is http://localhost:9000

Server version: 5.3 or higher

Configuration fields depend on the SonarQube server version.

Server authentication token:

SonarQube authentication token. Mandatory when anonymous access is disabled.

SonarQube account login:

SonarQube account used to perform analysis. Mandatory when anonymous access is disabled. No longer used since SonarQube 5.3.

SonarQube account password:

SonarQube account used to perform analysis. Mandatory when anonymous access is disabled. No longer used since SonarQube 5.3.

Advanced...

Delete SonarQube

Add SonarQube

Save Apply



Config SonarQube Scanner

Manage Jenkins -> Global Tool Configuration

SonarQube Scanner for MSBuild

SonarQube Scanner for MSBuild installations

Add SonarQube Scanner for MSBuild

List of SonarQube Scanner for MSBuild installations on this system

SonarQube Scanner

SonarQube Scanner installations

SonarQube Scanner

Name

SONAR_RUNNER_HOME

Install automatically

Required

Delete SonarQube Scanner

Add SonarQube Scanner

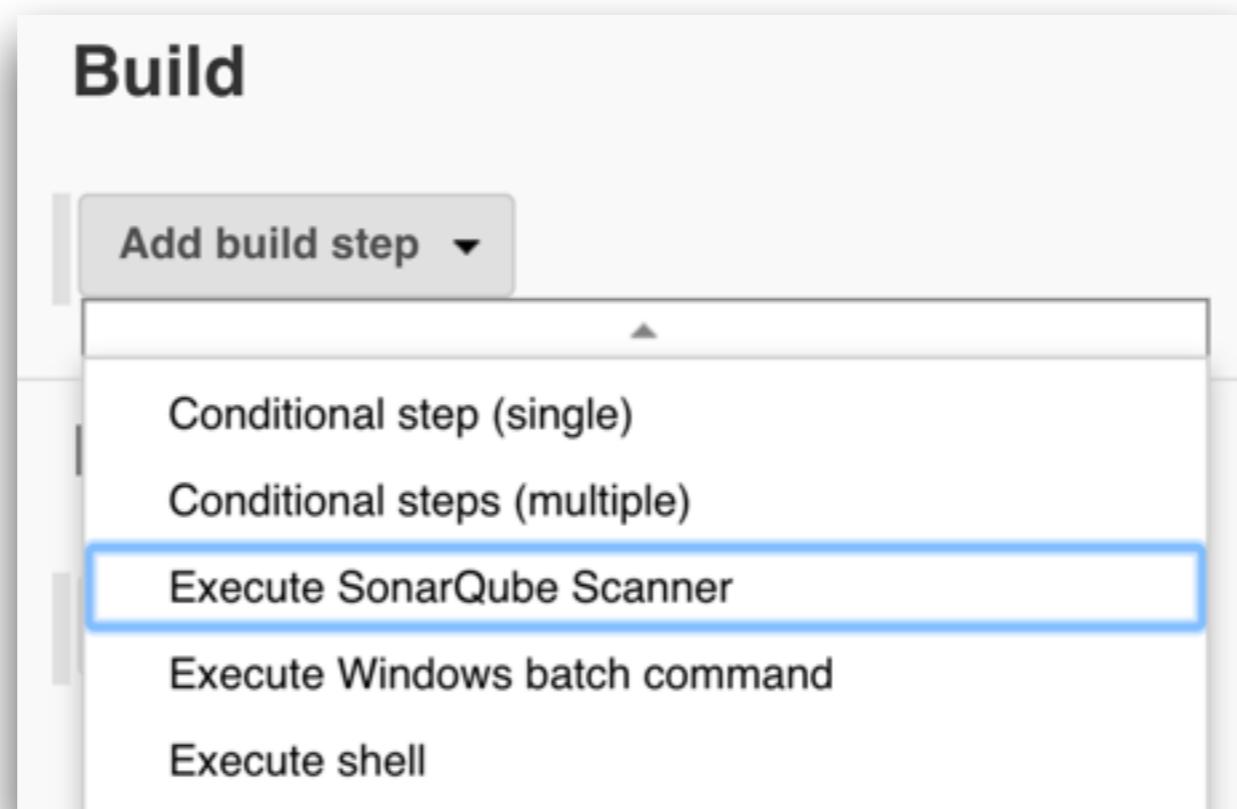
List of SonarQube Scanner installations on this system

The screenshot displays the Jenkins Global Tool Configuration interface. The 'SonarQube Scanner' section is currently selected. It contains fields for 'Name' (which is required) and 'SONAR_RUNNER_HOME', along with an 'Install automatically' checkbox. A 'Delete SonarQube Scanner' button is located to the right of the 'SONAR_RUNNER_HOME' field. Below the configuration form is a list of existing installations. The entire interface has a clean, modern design with a white background and light gray borders for the input fields.



Using SonarQube Scanner

Choose execute SonarQube Scanner



Using SonarQube Scanner

Config in your job

Build

Execute SonarQube Scanner

Task to run (?)

JDK (?)
JDK to be used for this SonarQube analysis

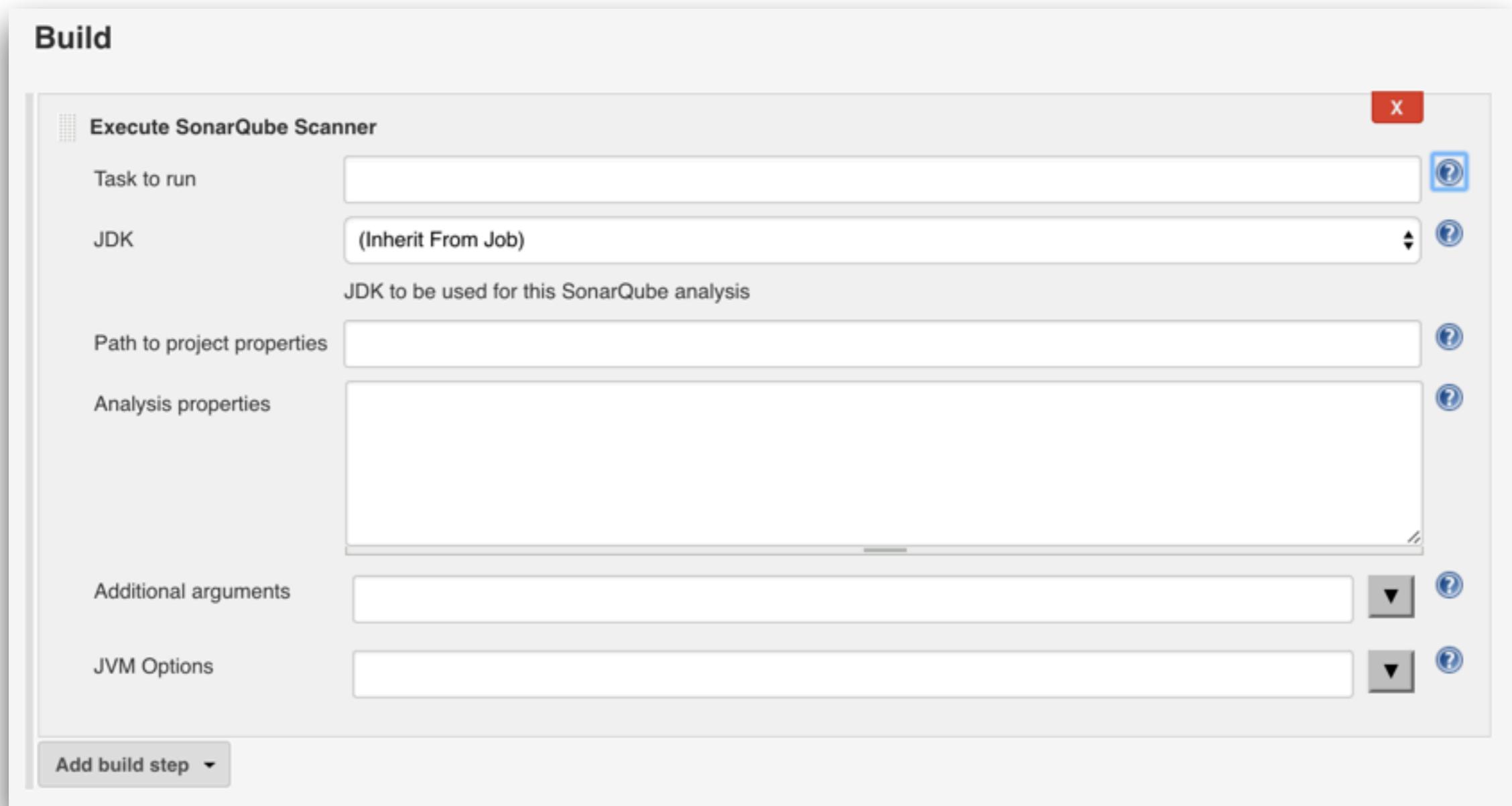
Path to project properties (?)

Analysis properties (?)

Additional arguments (?)

JVM Options (?)

Add build step ▾



Add sonar-project.properties configuration file of project

sonar-project.properties

```
# must be unique in a given SonarQube instance
sonar.projectKey=my:project
# this is the name and version displayed in the SonarQube UI. Was
sonar.projectName=My project
sonar.projectVersion=1.0

# Path is relative to the sonar-project.properties file. Replace '.
# This property is optional if sonar.modules is set.
sonar.sources=.

# Encoding of the source code. Default is default system encoding
#sonar.sourceEncoding=UTF-8
```

<https://github.com/SonarSource/sonar-scanning-examples>



Result Jenkins

The screenshot shows the Jenkins interface for the project "4_sonarqube". The left sidebar contains links: Back to Dashboard, Status, Changes, Workspace, Build Now, Delete Project, Configure, and SonarQube. The main content area is titled "Project 4_sonarqube". It features a red-bordered box containing SonarQube integration links: "SonarQube", "Workspace", and "Recent Changes". Below this is the "SonarQube Quality Gate" section, which displays "my project OK" and "server-side processing: Success". The "Upstream Projects" section lists "2_compile_unittest". On the left, there is a "Build History" table with four entries:

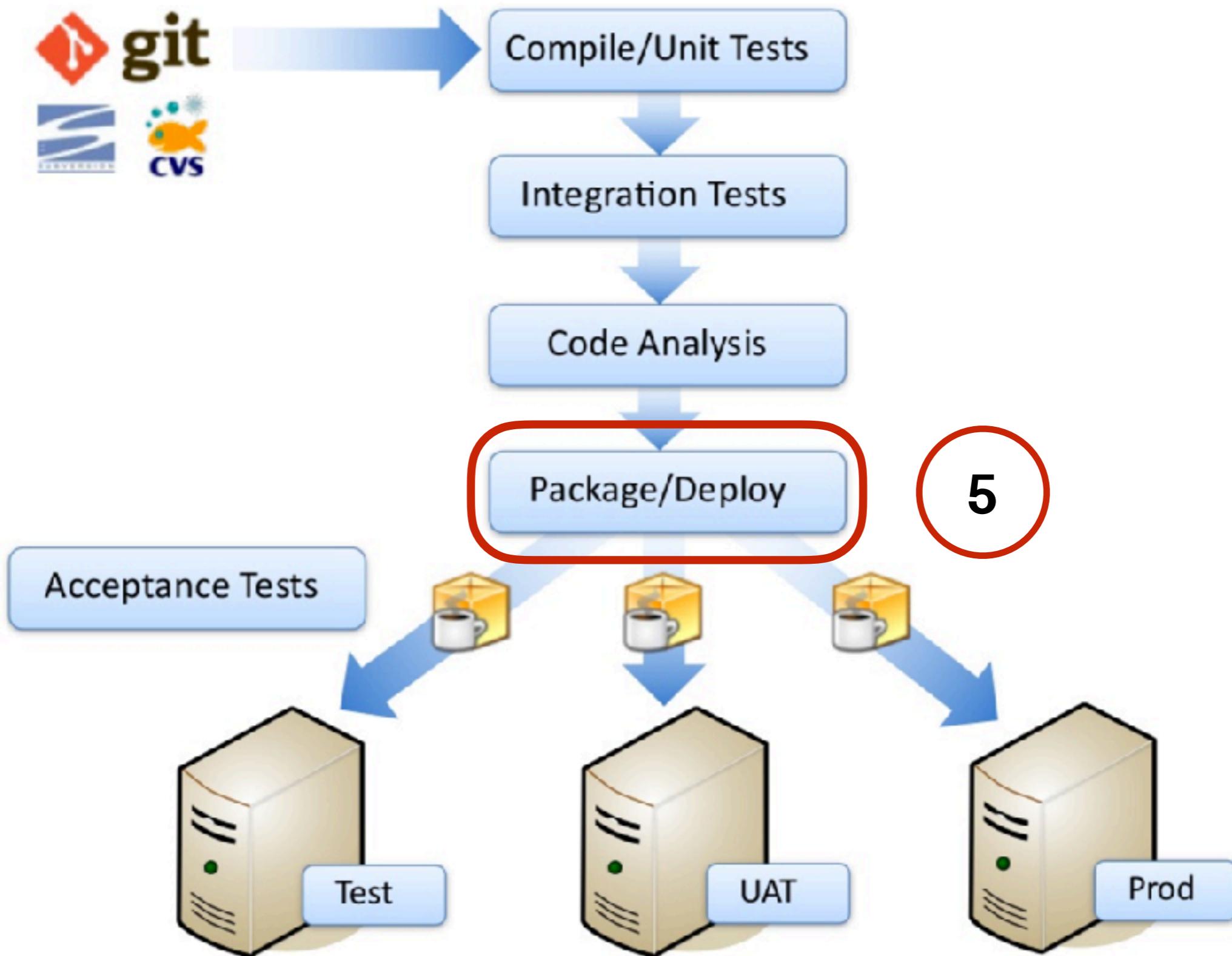
#	Build Status	Date
#4	Success	Jun 15, 2017 10:33 PM
#3	Failure	Jun 15, 2017 10:18 PM
#2	Failure	Jun 15, 2017 10:17 PM



See result in SonarQube

PROJECTS				
QG	NAME ▲	VERSION	LOC	BUGS
★	✓ my project	1.0	172	0
1 results				



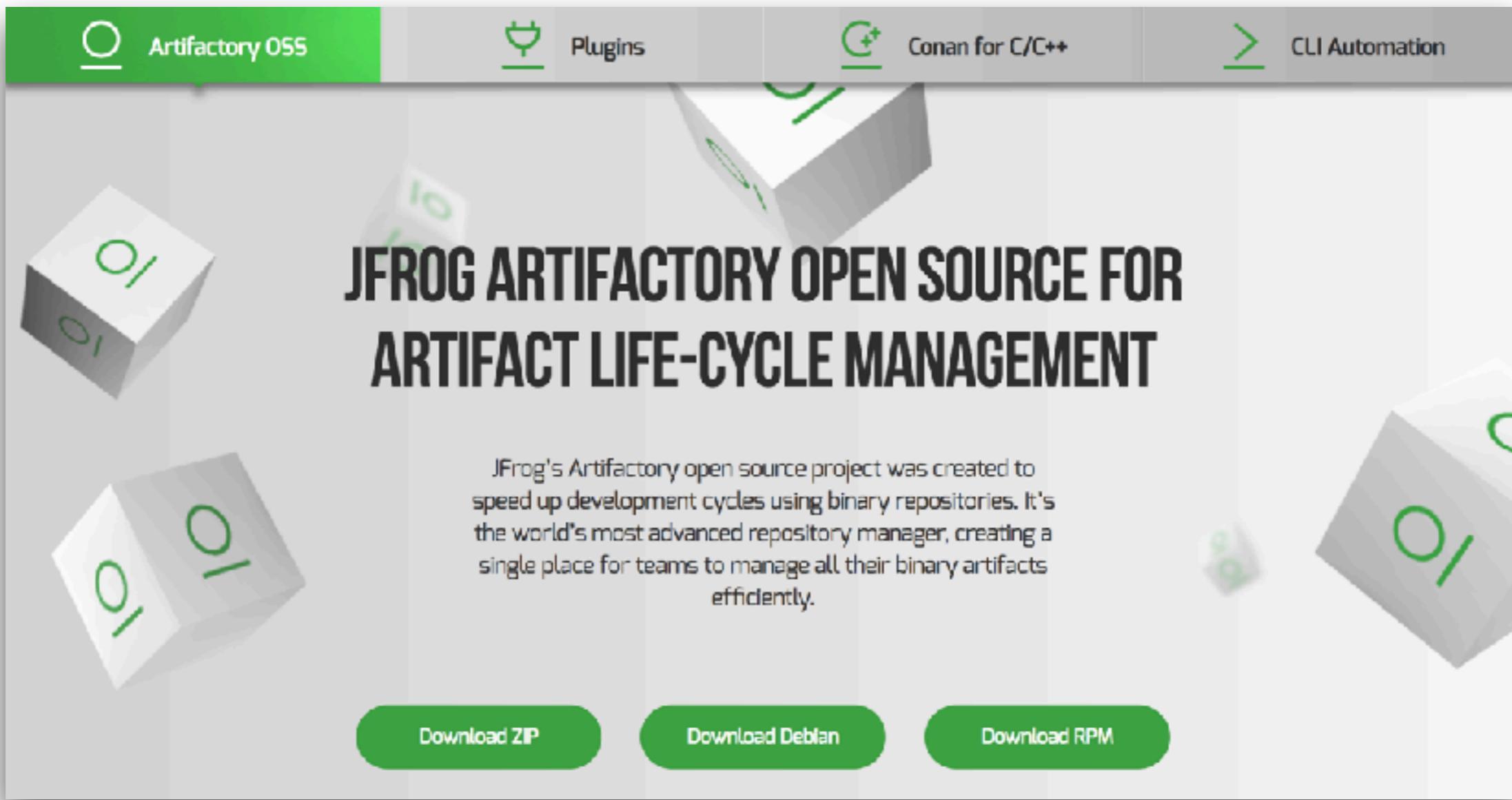


Keep artifact files with JFrog artifactory



Setup JFrog artifactory

Download from jfrog.com



The screenshot shows the JFrog Artifactory OSS download page. At the top, there's a navigation bar with four items: "Artifactory OSS" (selected), "Plugins", "Conan for C/C++", and "CLI Automation". Below the navigation bar, the main heading reads "JFROG ARTIFACTORY OPEN SOURCE FOR ARTIFACT LIFE-CYCLE MANAGEMENT". A descriptive paragraph explains that the project speeds up development cycles using binary repositories and is the world's most advanced repository manager. At the bottom, there are three download buttons: "Download ZIP", "Download Debian", and "Download RPM".

JFrog's Artifactory open source project was created to speed up development cycles using binary repositories. It's the world's most advanced repository manager, creating a single place for teams to manage all their binary artifacts efficiently.

[Download ZIP](#) [Download Debian](#) [Download RPM](#)



Login as admin

Welcome to JFrog Artifactory! X

Username * **admin**

Password * **password**

Remember me Log In



Create local repository

The screenshot shows the JFrog Artifactory web interface. The top navigation bar includes the logo, a search icon, 'Welcome, admin', and 'Help'. On the left, there's a vertical sidebar with icons for Home, Artifacts, Groups, Artifactory, and JFrog. The main content area is titled 'Local Repositories' and displays a table with one repository entry:

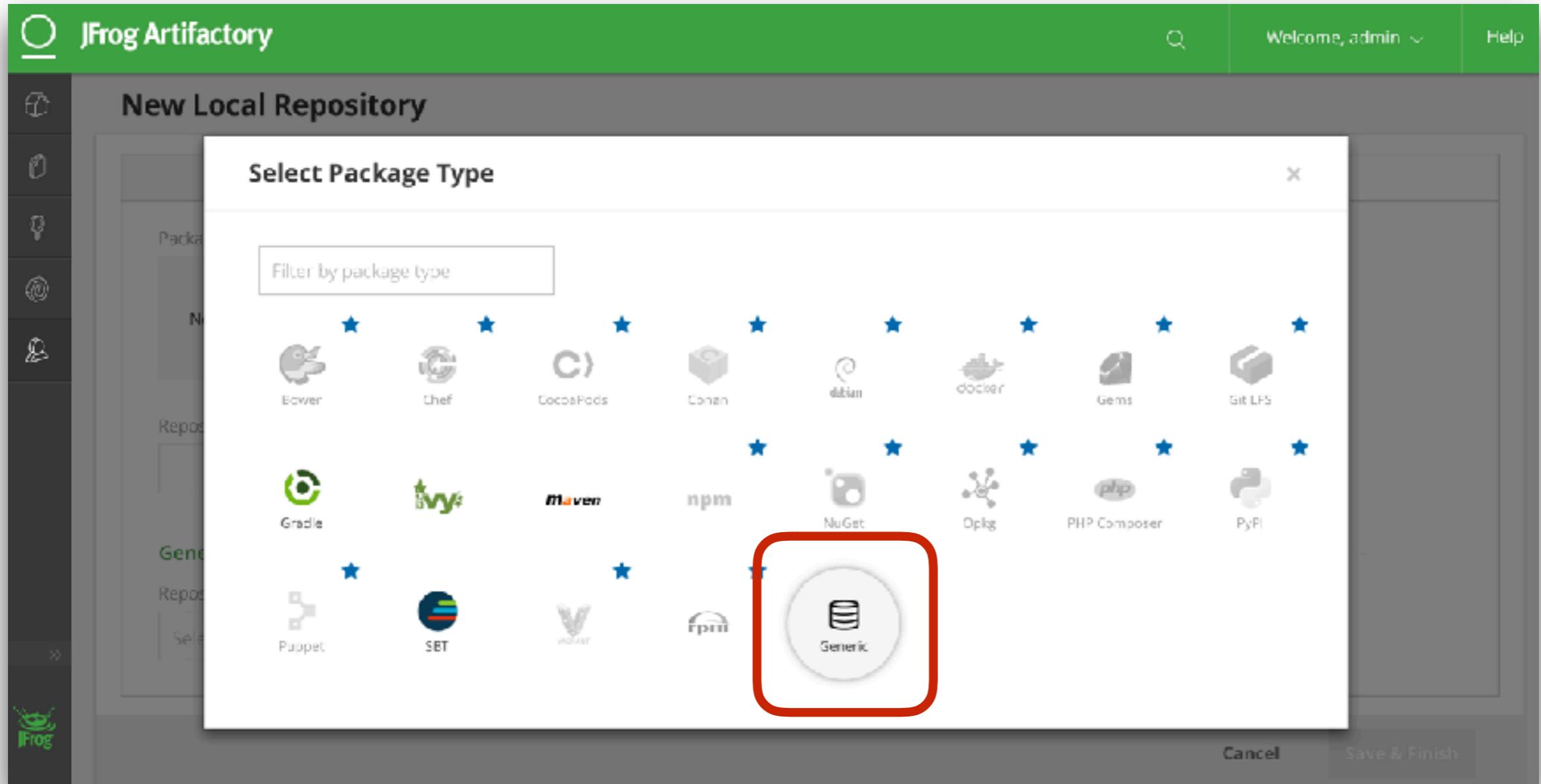
Repository Key	Type	Reca
example-repo-local	Generic	

Below the table is a 'Filter by Repository Key' input field. To the right, a 'Create Repositories' dropdown menu is open, listing:

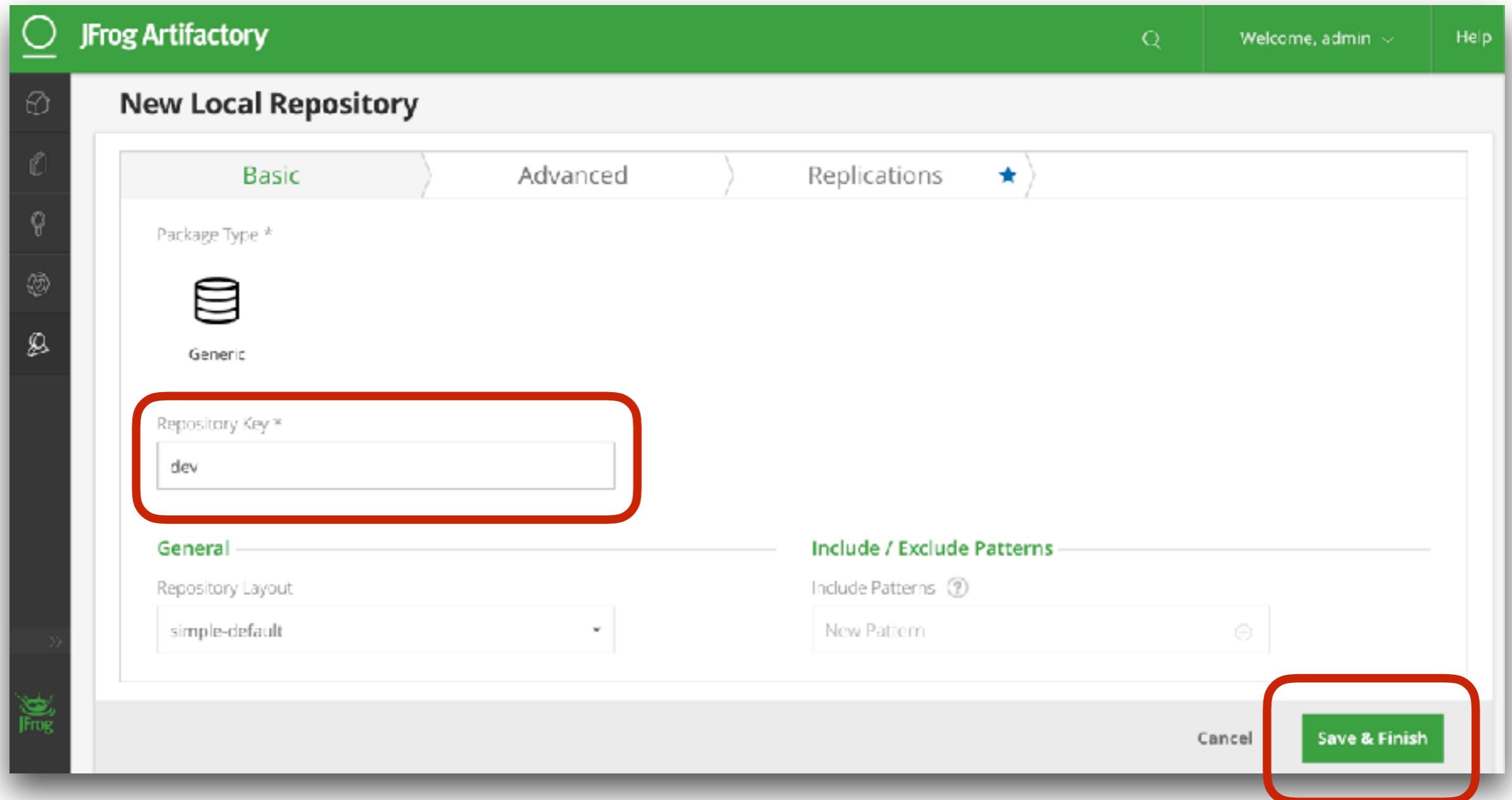
- Quick Setup
- Local Repository** (highlighted with a red box)
- Remote Repository
- Virtual Repository
- Distribution Repository
- Add User
- Add Group
- Add Permission
- Edit Profile
- Log Out



Choose Generic type



Fill in name and save



Install Artifactory plugin

The screenshot shows the Jenkins plugin manager interface. At the top, there is a search bar with the placeholder text "Filter: artifactory". Below the search bar, there are four tabs: "Updates", "Available" (which is selected), "Installed", and "Advanced". In the main list area, there is one item listed:

Install ↓	Name	Version
<input checked="" type="checkbox"/>	Artifactory Plugin Integrates Artifactory to Jenkins	2.11.0

At the bottom of the screen, there are three buttons: "Install without restart", "Download now and install after restart" (which is highlighted in blue), and "Update information obtained".



Config Artifactory Server

Manage Jenkins -> Configure System

Artifactory

Enable Push to Bintray [?](#)

Use the Credentials Plugin [?](#)

Artifactory servers

Artifactory

Server ID [?](#)

URL [?](#)

Default Deployer Credentials

Username [?](#)

Password [?](#)

[Advanced...](#)

Found Artifactory 5.3.2 [Test Connection](#)

Use Different Resolver Credentials [Delete](#)

[Add Artifactory Server](#)

List of Artifactory servers that projects will want to deploy artifacts and build info to

Enable Build-Info proxy for Docker images

[Save](#) [Apply](#)



Config job to upload artifactory

Choose Generic-Artifactory Integration

Build Environment

- Delete workspace before build starts
- Provide Configuration files
- Abort the build if it's stuck
- Add timestamps to the Console Output
- Ant/Ivy-Artifactory Integration
- Create Delivery Pipeline version
- Generic-Artifactory Integration
- Gradle-Artifactory Integration
- Maven3-Artifactory Integration
- Use secret text(s) or file(s)



Config job to upload artifactory

Artifactory Configuration

Download and upload by Specs Legacy patterns (deprecated)

Upload Details

Artifactory upload server `http://localhost:8081/artifactory`

Target Repository `dev` ? Different Value Refresh Repositories

Override default credentials

Published Artifacts
`target/*.war=>${BUILD_NUMBER}`

target/*.war=>\${BUILD_NUMBER}

Upload properties ?



Result in Artifactory server

The screenshot shows the JFrog Artifactory interface. On the left, there's a sidebar with various icons. The main area is titled "Artifact Repository Browser". It displays a tree view of artifacts under a repository named "dev". A red box highlights the artifact structure under "dev": it contains three folders labeled 12, 13, and 14, each containing a file named "demo.war". Below this is another folder named "example-repo-local". To the right of the tree view, there are tabs for "General", "Effective Permissions", and "Properties". The "General" tab is selected, showing the following information:

Info	
Name:	dev
Package Type:	Generic
Repository Path:	dev/
Repository Layout:	simple-default
Artifact Count / Size:	Show
Created:	15-06-17 23:08:37 ICT



Result Jenkins

Jenkins  search

Back to Dashboard Status Changes Workspace Build Now Delete Project Configure Artifactory Build Info

Project 5_create_war_file

 [Artifactory Build Info](#)

 [Workspace](#)

 [Recent Changes](#)

Upstream Projects

 [2_compile_unittest](#)

Permalinks

- [Last build \(#14\), 3 min 16 sec ago](#)
- [Last stable build \(#14\), 3 min 16 sec ago](#)
- [Last successful build \(#14\), 3 min 16 sec ago](#)
- [Last failed build \(#8\), 21 min ago](#)
- [Last unsuccessful build \(#8\), 21 min ago](#)
- [Last completed build \(#14\), 3 min 16 sec ago](#)

Build History trend —

#	Build Number	Date	Action
14	#14	Jun 15, 2017 11:44 PM	
13	#13	Jun 15, 2017 11:44 PM	
12	#12	Jun 15, 2017 11:40 PM	
11	#11	Jun 15, 2017 11:38 PM	
10	#10	Jun 15, 2017 11:28 PM	
9	#9	Jun 15, 2017 11:27 PM	



Deploy artifact to Apache Tomcat



Install Deploy to container plugin

The screenshot shows a software interface for managing plugins. At the top, there is a search bar labeled "Filter:" with the text "Deploy to container". Below the search bar, there are four tabs: "Updates", "Available" (which is selected), "Installed", and "Advanced". A table below the tabs lists a single plugin entry:

Install ↓	Name	Version
<input checked="" type="checkbox"/>	Deploy to container Plugin	1.10

At the bottom of the interface, there are two buttons: "Install without restart" and "Download now and install after restart". To the right of these buttons, the text "Update information obtained: 2 hr 16 mi" is displayed.

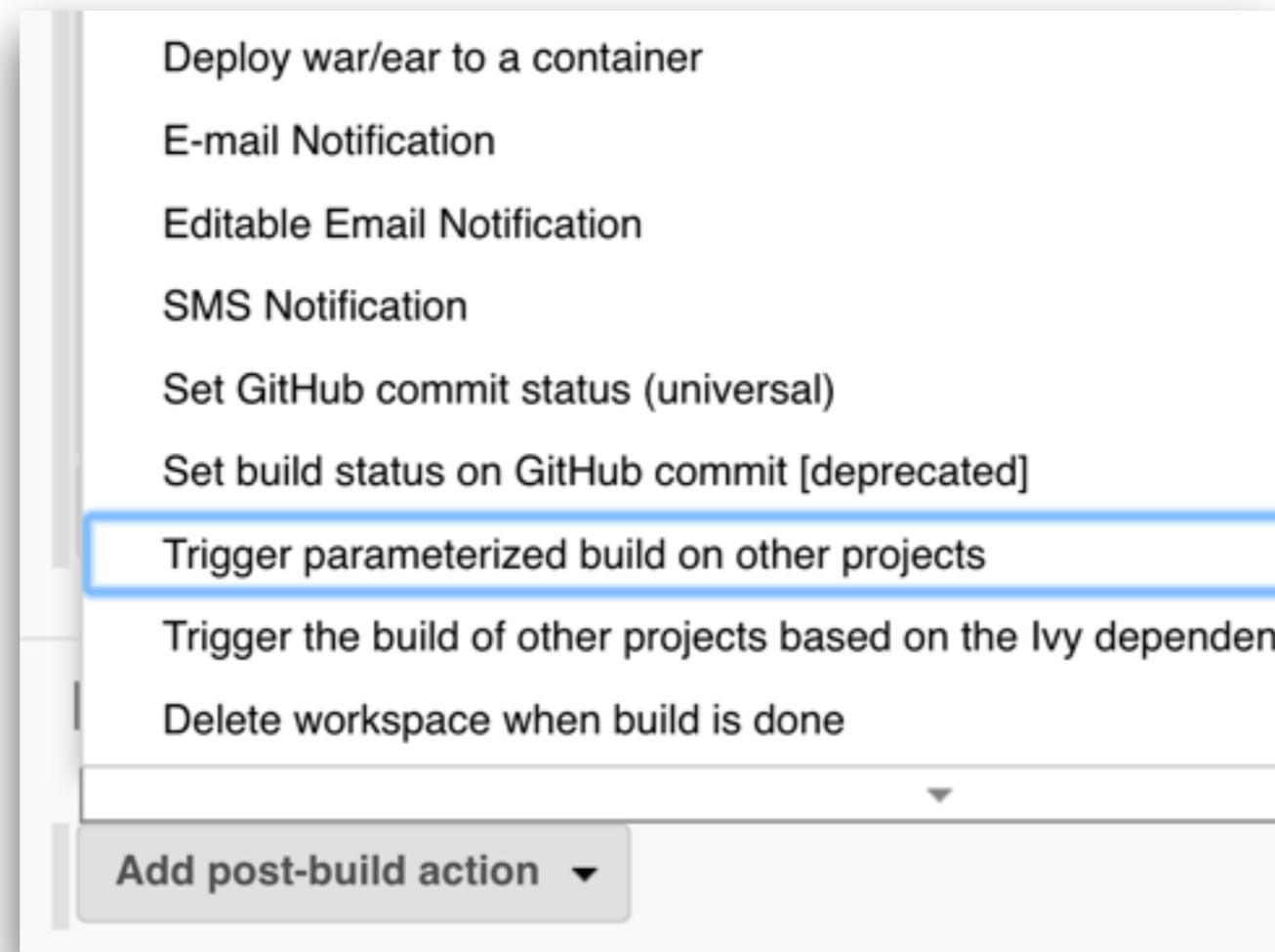


Create new job to deploy

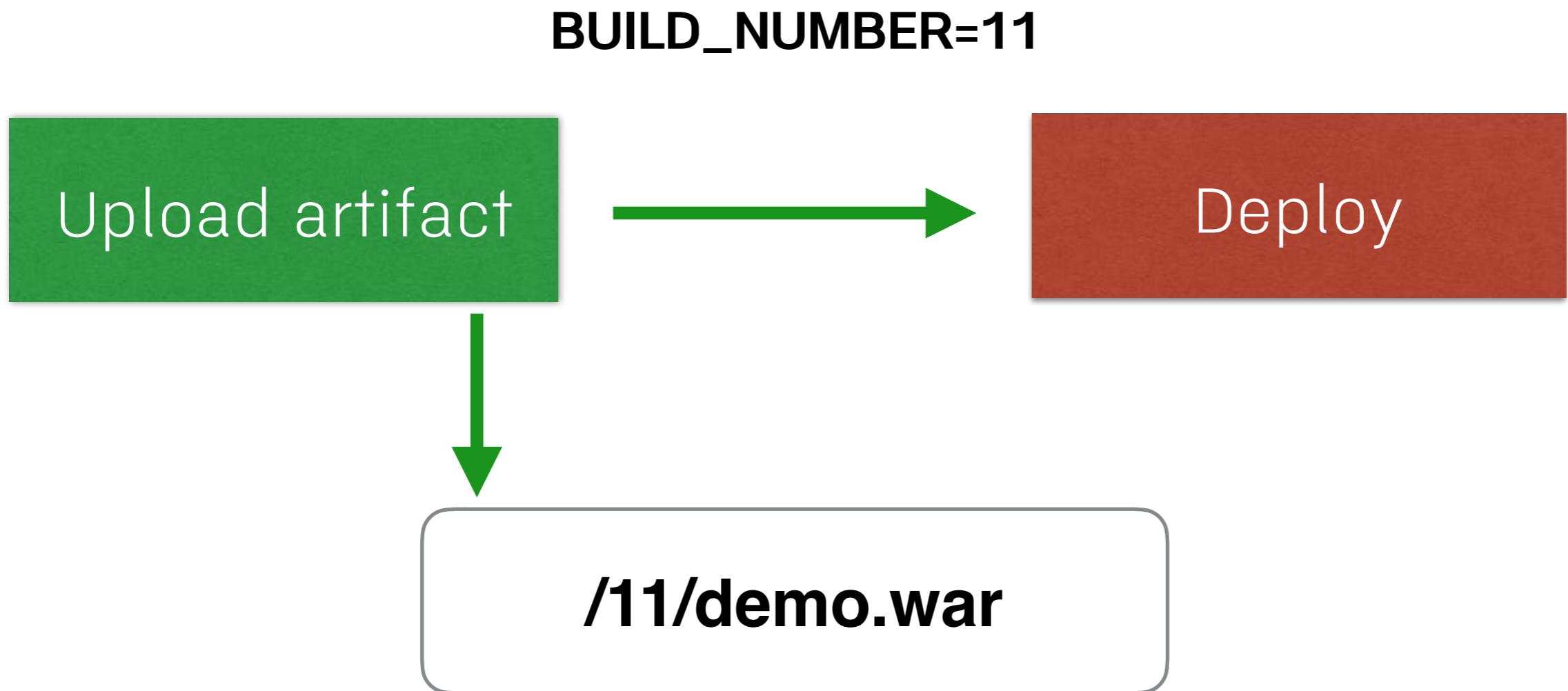
The screenshot shows the Jenkins interface for a project named "6_deploy_to_tomcat". The top navigation bar includes the Jenkins logo, a search bar, and a back breadcrumb trail: Jenkins > 6_deploy_to_tomcat. The left sidebar contains links: Back to Dashboard, Status, Changes, Workspace, Build Now, Delete Project, Configure, and Build History. The main content area displays the project name "Project 6_deploy_to_tomcat" and two links: "Workspace" and "Recent Changes". Below these are sections for "Permalinks" and "trend".



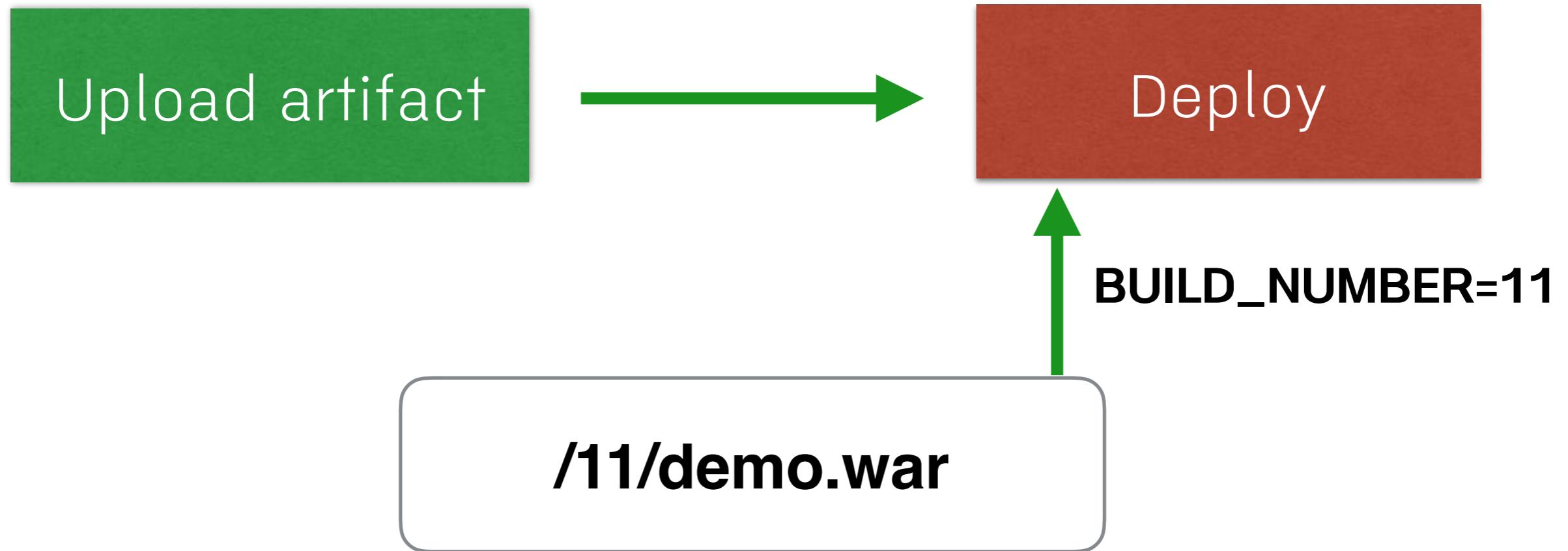
Trigger with parameterized



Trigger with parameterized



Trigger with parameterized



Trigger with parameterized

Post-build Actions

X

Trigger parameterized build on other projects

Build Triggers

Projects to build

6_deploy_to_tomcat, Blank project name in the list

Trigger when build is

Stable

Trigger build without parameters

Predefined parameters

Parameters **BUILD_NUMBER=\${BUILD_NUMBER}**

BUILD_NUMBER=\${BUILD_NUMBER}

Add Parameters ▾

TARGET JOB

The screenshot shows the Jenkins 'Post-build Actions' configuration for a job. Under 'Build Triggers', there is a 'Projects to build' field containing '6_deploy_to_tomcat,' which is highlighted with a red box and accompanied by the error message 'Blank project name in the list'. Below it, the 'Trigger when build is' dropdown is set to 'Stable'. There is also a checkbox for 'Trigger build without parameters' which is unchecked. In the 'Predefined parameters' section, there is a 'Parameters' field containing 'BUILD_NUMBER=\${BUILD_NUMBER}' which is also highlighted with a red box. At the bottom left, there is a button labeled 'Add Parameters ▾'. The word 'TARGET JOB' is displayed prominently in red at the top right of the configuration area.



Download from Artifactory server



Add String parameter

6_deploy_to_tomcat >

General Source Code Management Build Triggers Build Environment Build Post-build Actions

This project is parameterized

String Parameter

Name	BUILD_NUMBER
Default Value	\$(BUILD_NUMBER)
Description	

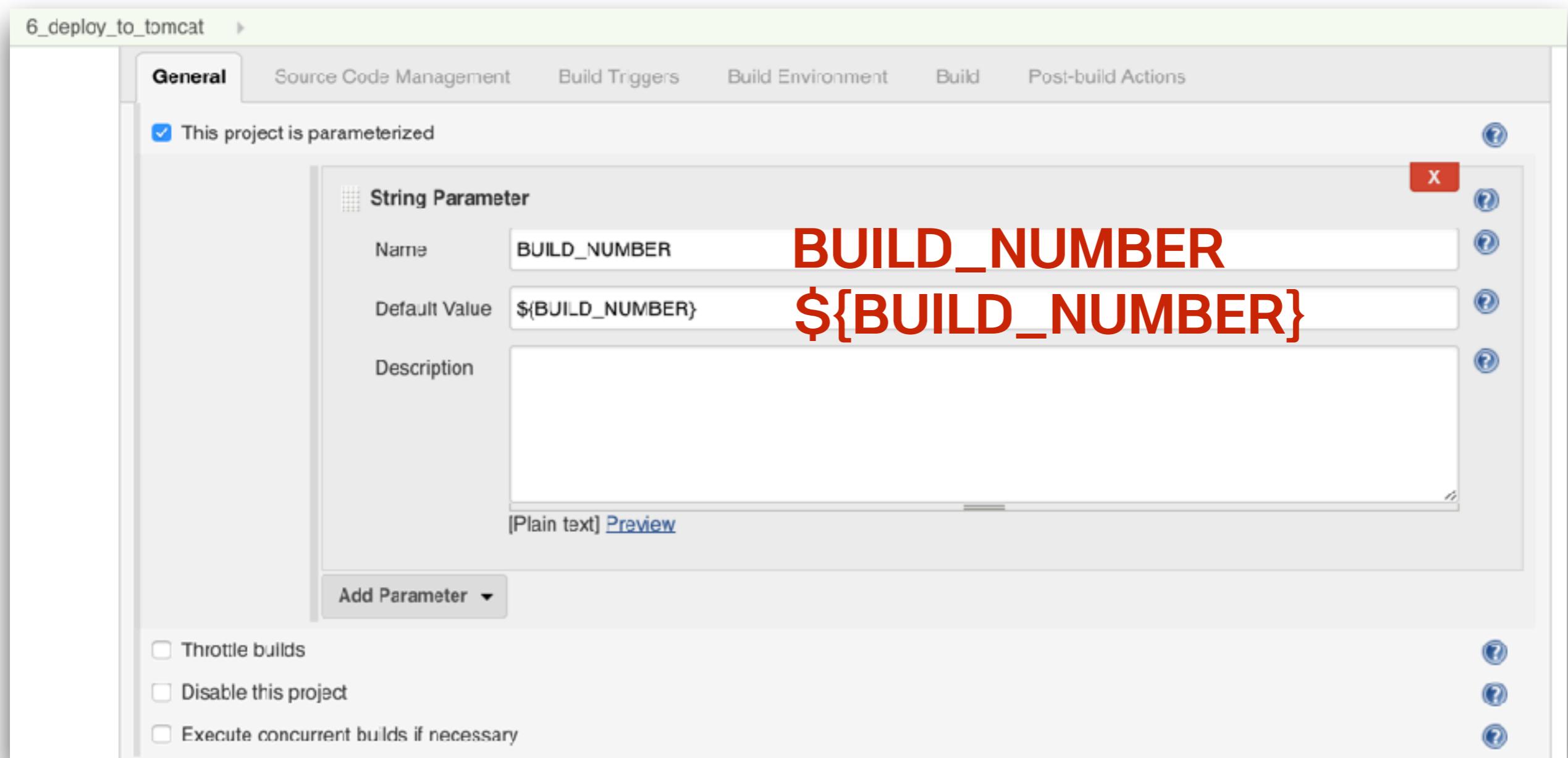
[Plain text] [Preview](#)

Add Parameter ▾

Throttle builds

Disable this project

Execute concurrent builds if necessary



Download artifact file

wget http://localhost:8081/artifactory/dev/

\$BUILD_NUMBER/demo.war



Wget for windows

<http://gnuwin32.sourceforge.net/packages/wget.htm>

Wget for Windows

Wget: retrieve files from the WWW

Version

1.11.4

Description

GNU Wget is a free network utility to retrieve files from the World Wide Web using HTTP and FTP, the two most widely used Internet protocols. It works non-interactively, thus enabling work in the background, after having logged off.

The recursive retrieval of HTML pages, as well as FTP sites is supported -- you can use Wget to make mirrors of archives and home pages, or traverse the web like a WWW robot (Wget understands /robots.txt).

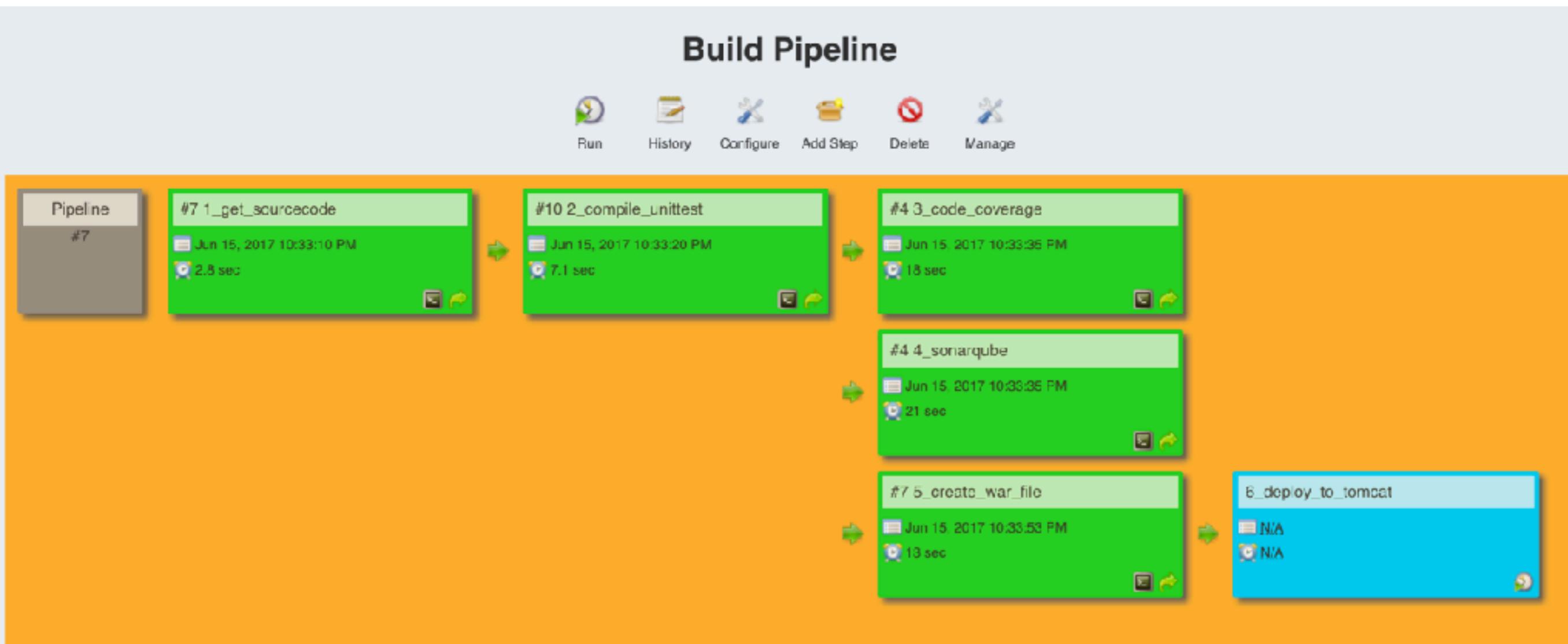
Wget works exceedingly well on slow or unstable connections, keeping getting the document until it is fully retrieved. Re-getting files from where it left off works on servers (both HTTP and FTP) that support it. Matching of wildcards and recursive mirroring of directories are available when retrieving via FTP. Both HTTP and FTP retrievals can be time-stamped, thus Wget can see if the remote file has changed since last retrieval and automatically retrieve the new version if it has.

Wget supports proxy servers, which can lighten the network load, speed up retrieval and provide access behind firewalls. If you are behind a firewall that requires the use of a socks style gateway, you can get the socks library and compile wget with support for socks.

Most of the features are configurable, either through command-line options, or via initialization file .wgetrc. Wget allows you to install a global startup file (etc/wgetrc by default) for site settings.



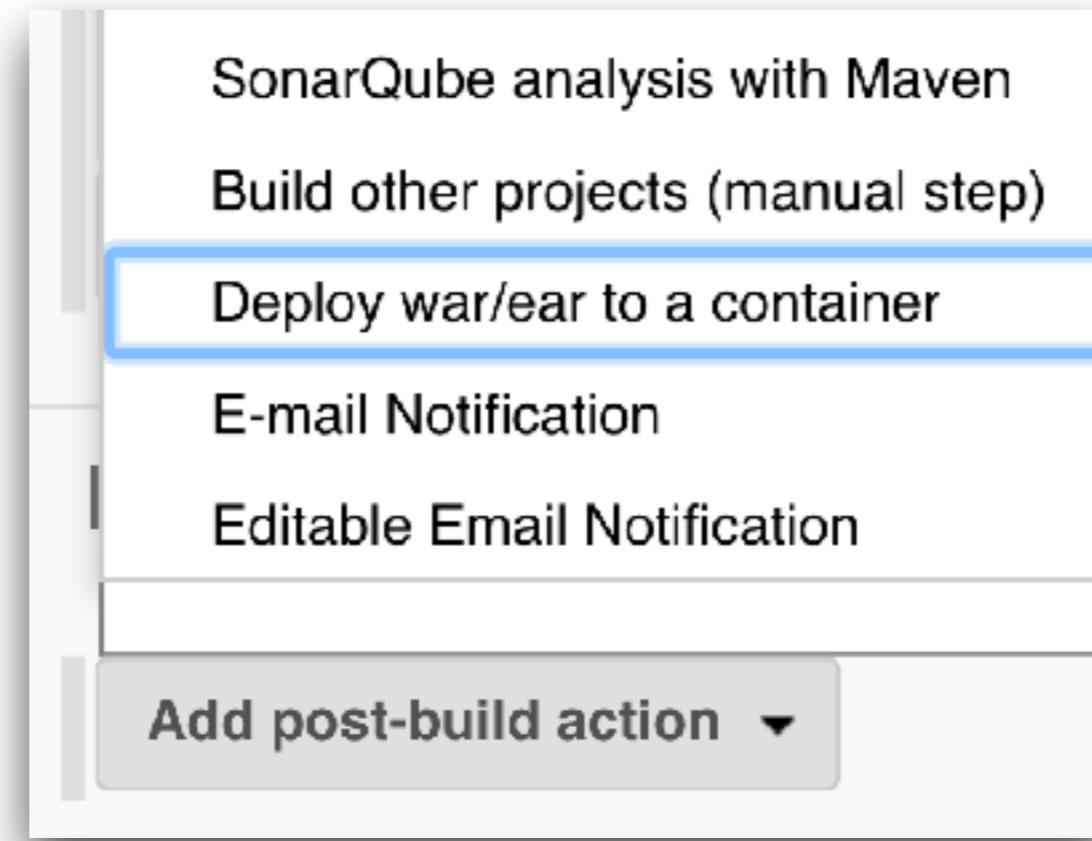
My build pipeline



Deploy WAR file to Apache Tomcat



Add post-build action



Deploy war/ear to a container

Post-build Actions

Deploy war/ear to a container

WAR/EAR files	demo.war	demo.war
Context path	/demo	/demo
Containers	Add Container ▾	
Deploy on failure	<input type="checkbox"/>	



Choose Tomcat container

Post-build Actions

Deploy war/ear to a container

WAR/EAR files: demo.war

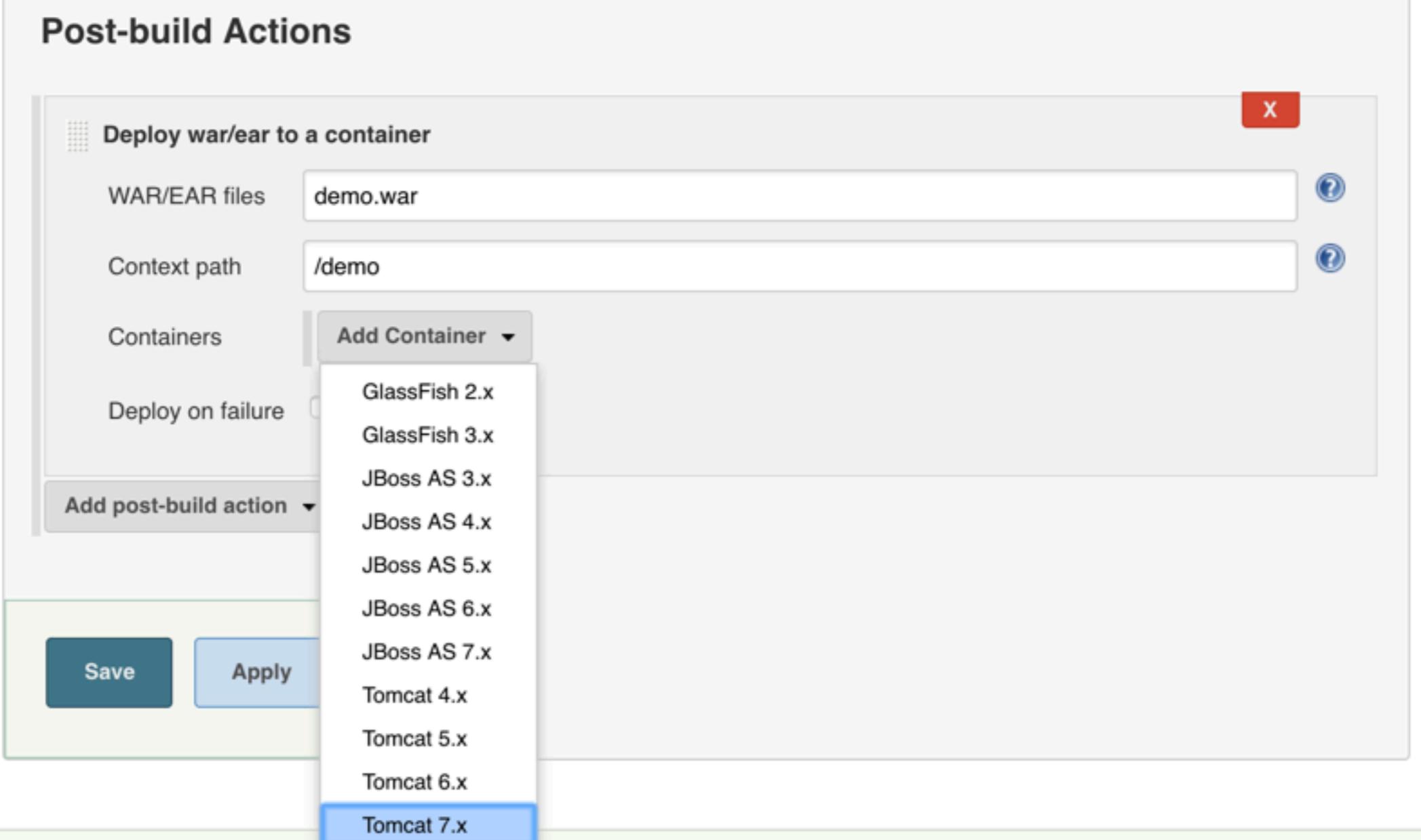
Context path: /demo

Containers: Add Container ▾

- GlassFish 2.x
- GlassFish 3.x
- JBoss AS 3.x
- JBoss AS 4.x
- JBoss AS 5.x
- JBoss AS 6.x
- JBoss AS 7.x
- Tomcat 4.x
- Tomcat 5.x
- Tomcat 6.x
- Tomcat 7.x

Add post-build action ▾

Save Apply



Config tomcat in jenkins

Deploy war/ear to a container

WAR/EAR files: demo.war

Context path: /demo

Containers:

Tomcat 7.x

Manager user name: [redacted]

Manager password: [redacted]

Tomcat URL: [redacted]

Add Container ▾

Deploy on failure:



Install Apache Tomcat



Edit tomcat user

\$TOMCAT_HOME/conf/tomcat-users.xml

```
<tomcat-users>
    <role rolename="manager"/>
    <role rolename="manager-script"/>
    <role rolename="manager-gui"/>
    <user username="deploy" password="deploy"
          roles="manager,manager-gui,manager-script"/>
</tomcat-users>
```



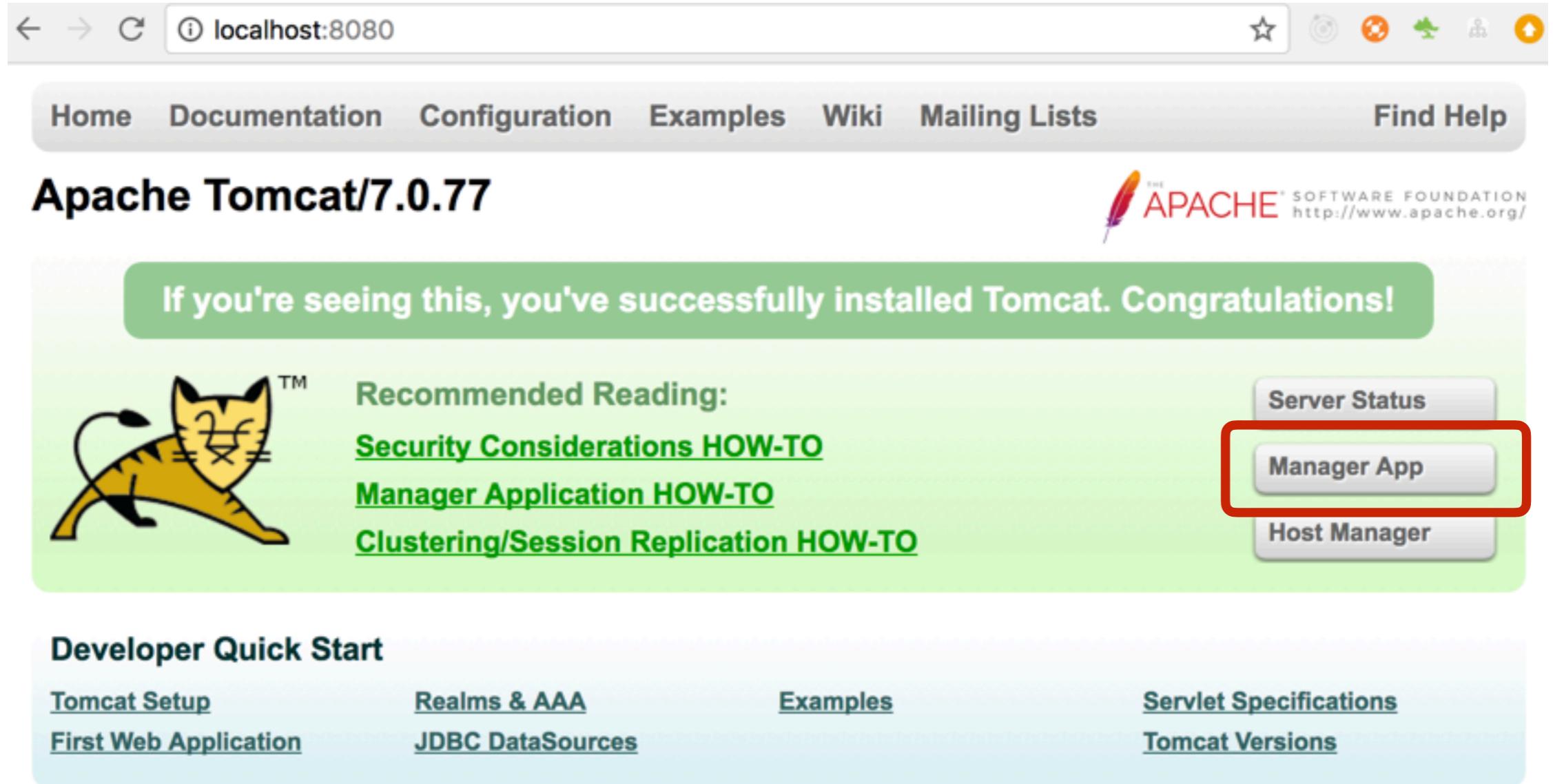
Start Tomcat Server

The screenshot shows a web browser window with the URL `localhost:8080` in the address bar. The page content is as follows:

- Apache Tomcat/7.0.77**
- APACHE SOFTWARE FOUNDATION <http://www.apache.org/>**
- If you're seeing this, you've successfully installed Tomcat. Congratulations!**
- 
- Recommended Reading:**
 - [Security Considerations HOW-TO](#)
 - [Manager Application HOW-TO](#)
 - [Clustering/Session Replication HOW-TO](#)
- Server Status**
- Manager App**
- Host Manager**
- Developer Quick Start**
- [Tomcat Setup](#)
- [First Web Application](#)
- [Realms & AAA](#)
- [JDBC DataSources](#)
- [Examples](#)
- [Servlet Specifications](#)
- [Tomcat Versions](#)



Try to login



A screenshot of a web browser displaying the Apache Tomcat 7.0.77 homepage. The URL in the address bar is `localhost:8080`. The page title is "Apache Tomcat/7.0.77". At the top, there is a navigation bar with links to Home, Documentation, Configuration, Examples, Wiki, and Mailing Lists, along with a "Find Help" button. To the right of the navigation bar is the Apache Software Foundation logo. A green callout box contains the text: "If you're seeing this, you've successfully installed Tomcat. Congratulations!". Below this, there is a cartoon cat logo with the text "TM" next to it. To the right of the cat is a section titled "Recommended Reading:" with links to "Security Considerations HOW-TO", "Manager Application HOW-TO", and "Clustering/Session Replication HOW-TO". On the far right, there are three buttons: "Server Status", "Manager App", and "Host Manager", with "Manager App" being highlighted by a red rectangle. At the bottom, there is a "Developer Quick Start" section with links to "Tomcat Setup", "First Web Application", "Realms & AAA", "JDBC DataSources", "Examples", "Servlet Specifications", and "Tomcat Versions".



Config jenkins again

Deploy war/ear to a container

WAR/EAR files demo.war

Context path /demo

Containers

Tomcat 7.x

Manager user name deploy

Manager password

Tomcat URL http://localhost:8080

Add Container ▾

Deploy on failure

deploy
deploy
http://localhost:8080



Try to build !!

Jenkins > 6_deploy_to_tomcat > #5

[Back to Project](#)
[Status](#)
[Changes](#)
Console Output
[View as plain text](#)
[Edit Build Information](#)
[Delete Build](#)
[Parameters](#)
[Previous Build](#)

Console Output

```
Started by upstream project "5_create_war_file" build number 19
originally caused by:
  Started by user somkiat
Building in workspace /Users/somkiat/data/slide/ci-
cd/swpark/software/keep/workspace/6_deploy_to_tomcat
[6_deploy_to_tomcat] $ /bin/sh -xe
/var/Folders/t5/8kg23s_97z9dw14tfcl06dqw000gn/T/hudson2611444727417694445.sh
+ wget http://localhost:8081/artifactory/dev/19/demo.war
--2017-06-15 01:07:44-- http://localhost:8081/artifactory/dev/19/demo.war
Resolving localhost... ::1, 127.0.0.1
Connecting to localhost|::1|:8081... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10879 (11K) [application/java-archive]
Saving to: 'demo.war.2'

OK ..... 100% 76.3M=0s

2017-06-15 01:07:44 (76.3 MB/s) - 'demo.war.2' saved [10879/10879]

Deploying /Users/somkiat/data/slide/ci-
cd/swpark/software/keep/workspace/6_deploy_to_tomcat/demo.war to container Tomcat 7.x Remote
[/Users/somkiat/data/slide/ci-cd/swpark/software/keep/workspace/6_deploy_to_tomcat/demo.war] is
not deployed. Doing a fresh deployment.
Deploying [/Users/somkiat/data/slide/ci-
cd/swpark/software/keep/workspace/6_deploy_to_tomcat/demo.war]
Finished: SUCCESS
```

Check result on Tomcat

The screenshot shows the Apache Tomcat Web Application Manager interface at localhost:8080/manager/html. The page includes the Tomcat logo and the Apache Software Foundation logo.

Tomcat Web Application Manager

Message: OK

Manager

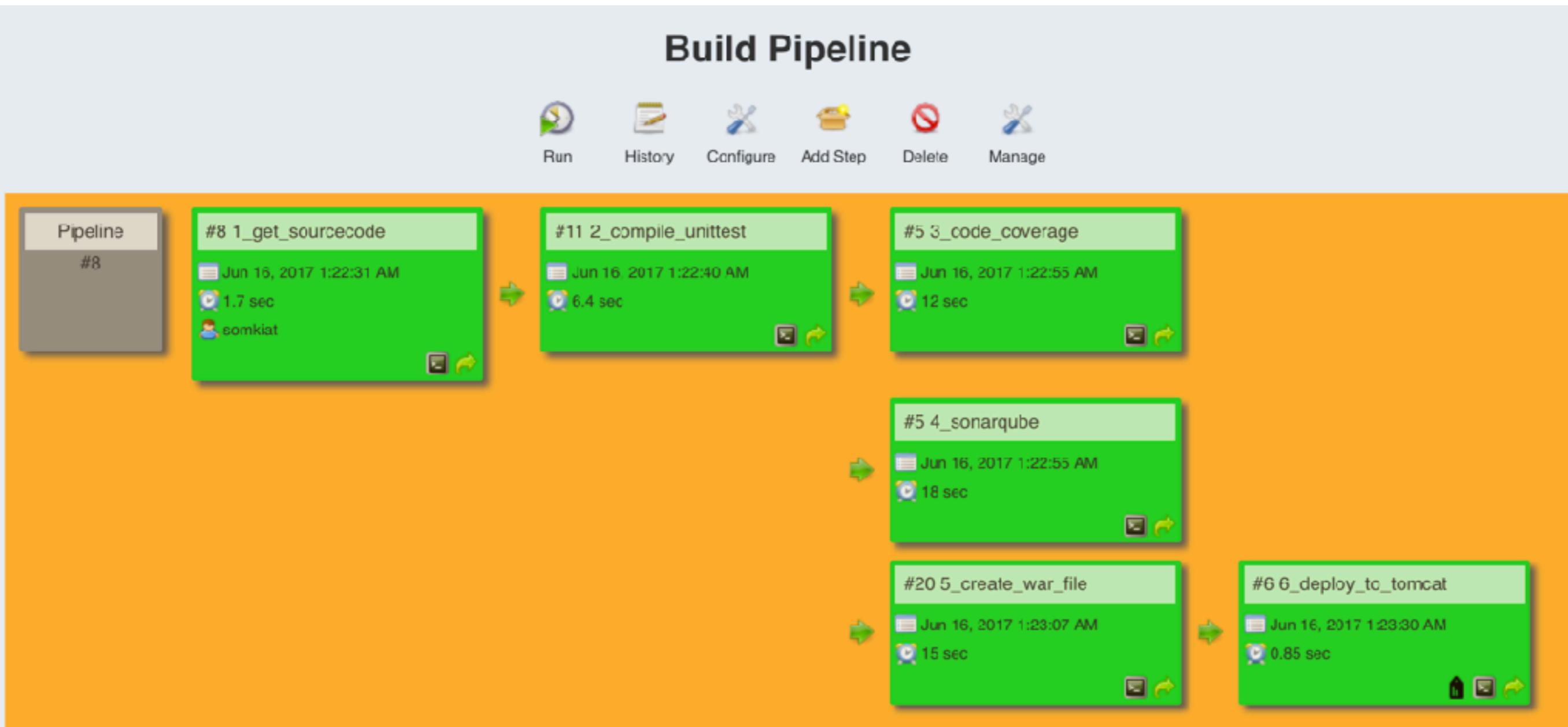
[List Applications](#) [HTML Manager Help](#) [Manager Help](#) [Server Status](#)

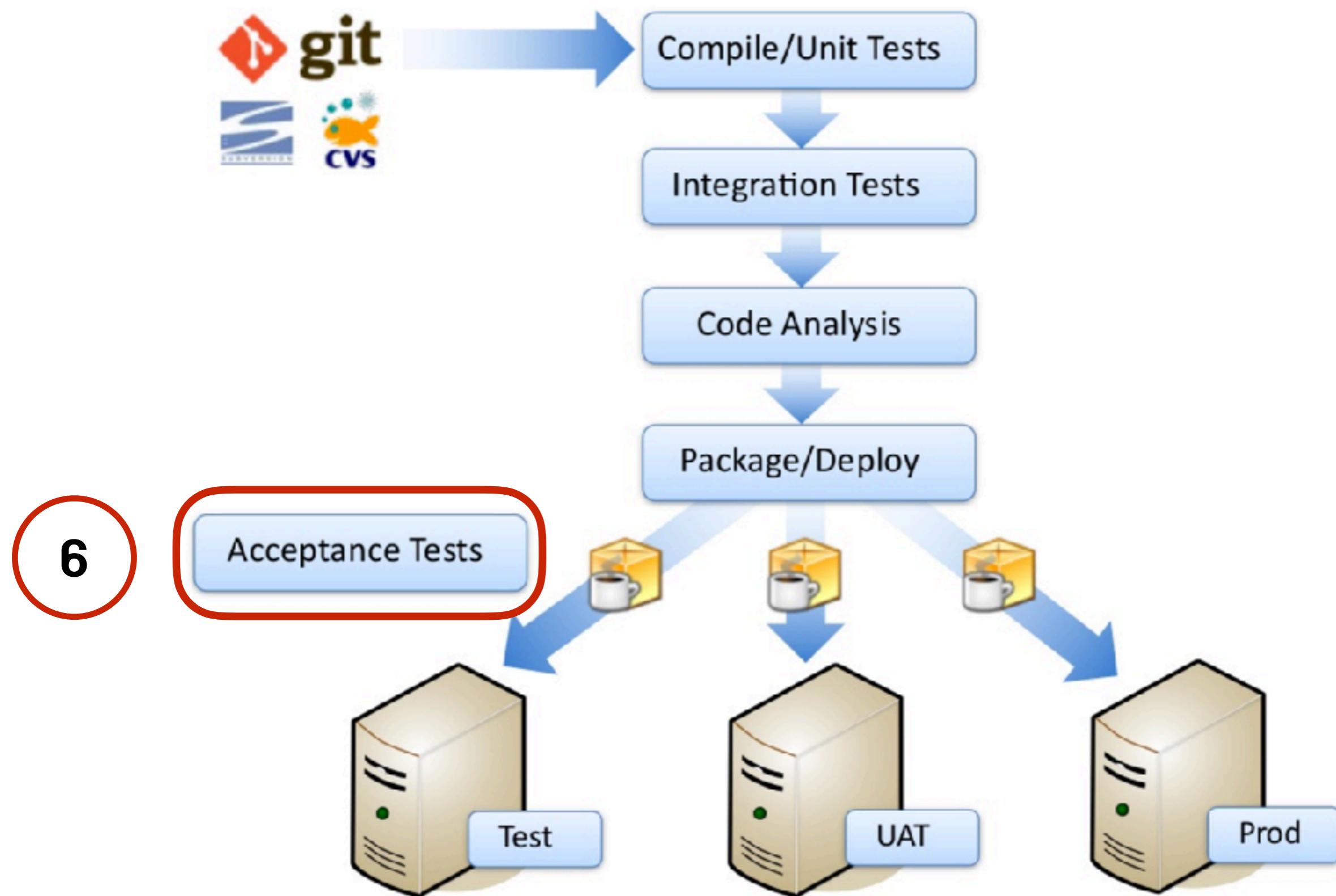
Applications

Path	Version	Display Name	Running	Sessions	Commands
/	None specified	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/demo	None specified	Grade Converter	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/docs	None specified	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes



My build pipeline





Acceptance test with Robotframework



Install Robot Framework plugin

Filter:

Updates Available Installed Advanced

Install ↓	Name	Version
<input checked="" type="checkbox"/> Robot Framework plugin	Shows Robot Framework test results in project	1.6.4

Install without restart Download now and install after restart

Update information obtained: 9 hr 31 min ago



Install Python 2.7.13

<https://www.python.org/>

The screenshot shows the Python.org website's download page. At the top, there's a search bar with a magnifying glass icon and a 'GO' button. Below the header, a navigation menu includes 'About', 'Downloads', 'Documentation', 'Community', 'Success Stories', and 'News'. A sidebar on the left contains a snippet of Python code and links to 'All releases', 'Source code', 'Windows', 'Mac OS X', 'Other Platforms', 'License', and 'Alternative Implementations'. The main content area features a heading 'Download for Mac OS X' with two large buttons for 'Python 3.6.1' and 'Python 2.7.13'. Below this, a message states: 'Not the OS you are looking for? Python can be used on many operating systems and environments.' followed by a link 'View the full list of downloads.'



Try ...

\$python

\$pip



Configuration for Windows

```
$set PYTHON_HOME=c:/Python27
```

```
$set PATH=.;%PYTHON_HOME%;
```

```
%PYTHON_HOME%/Scripts;%PATH%
```



Configuration for Mac

```
$easy_install pip
```



Try again !!

\$python

\$pip



Install Robot framework

\$pip install robotframework

\$pip install robotframework-selenium2library



Try to run

\$pybot

```
[ ERROR ] Expected at least 1 argument, got 0.
```

```
Try --help for usage information.
```



Create new job on Jenkins

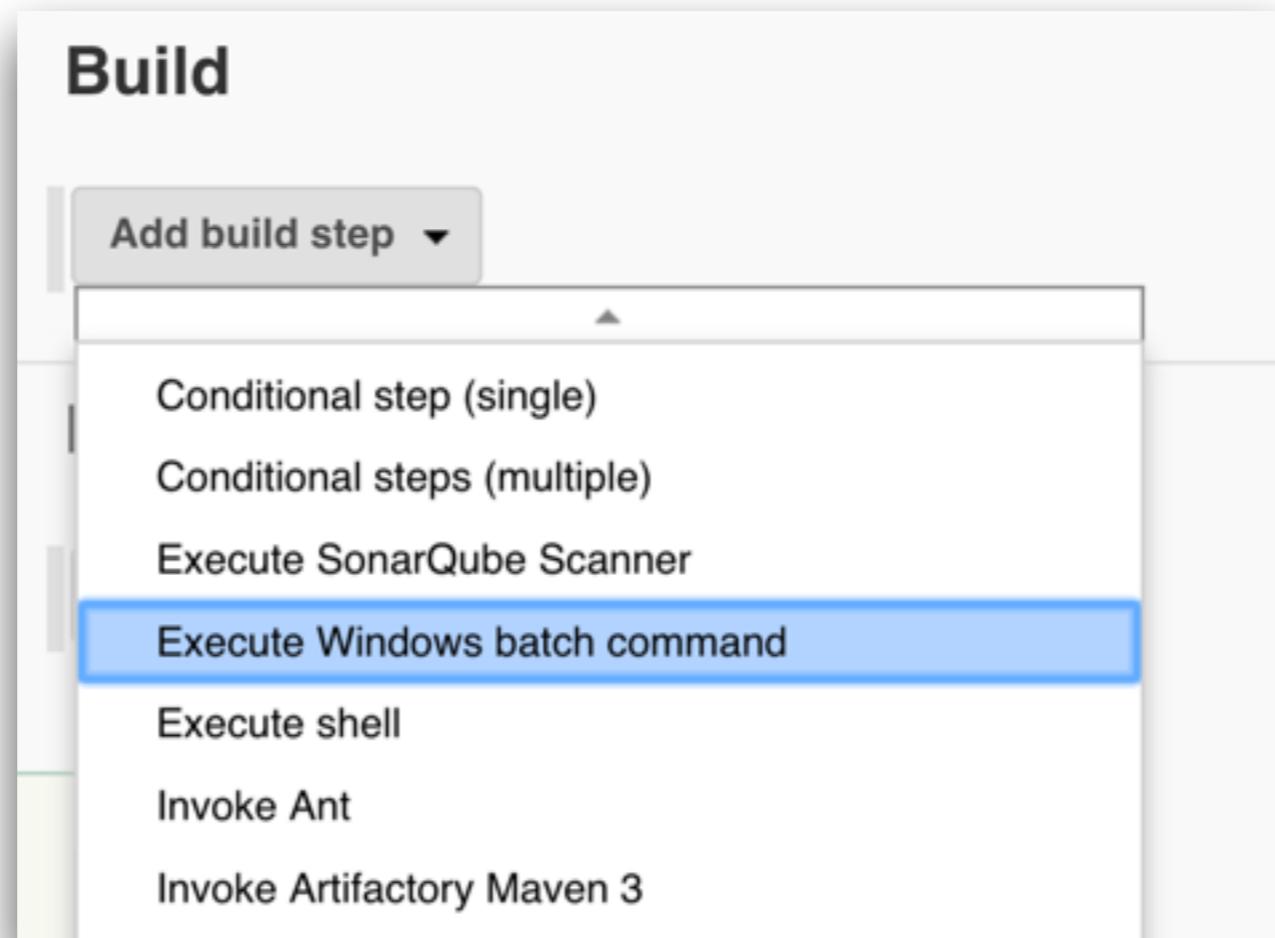
Enter an item name

» Required field

 **Freestyle project**
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.



Config job to run robot



Config job to run robot

Build

Execute shell

Command `pybot *.robot`

pybot *.robot

See [the list of available environment variables](#)

Advanced...



Add report in Post-build

The screenshot shows a list of Jenkins post-build actions. The 'Publish Robot Framework test results' option is highlighted with a blue background, indicating it is selected or being demonstrated. Other options listed include: Publish JUnit test result report, Publish Javadoc, Record fingerprints of files to track usage, Git Publisher, SonarQube analysis with Maven, Build other projects (manual step), Deploy war/ear to a container, E-mail Notification, and Editable Email Notification. At the bottom left, there is a button labeled 'Add post-build action ▾'.

- Publish JUnit test result report
- Publish Javadoc
- Publish Robot Framework test results**
- Record fingerprints of files to track usage
- Git Publisher
- SonarQube analysis with Maven
- Build other projects (manual step)
- Deploy war/ear to a container
- E-mail Notification
- Editable Email Notification

Add post-build action ▾



Add report in Post-build

Post-build Actions

Publish Robot Framework test results X ?

Directory of Robot output

Path to directory containing robot xml and html files (relative to build workspace) Advanced...

Thresholds for build result ?

! % Entry must be percentage value between 0-100

! % Entry must be percentage value between 0-100

Use thresholds for critical tests only

Add post-build action ▾



Try to build !!

Jenkins search Somkiat | log out

Jenkins > 7_acceptance_testing > ENABLE AUTO REFRESH

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Delete Project](#)

[Configure](#)

[Robot Results](#)

[Build History](#)

[trend](#)

find

#1 Jun 16, 2017 7:44 AM

[RSS for all](#) [RSS for failures](#)

Project 7_acceptance_testing

[add description](#)

[Disable Project](#)

Robot Framework Tests Trend (all tests)

Number of test cases

Passed Failed

Build number

Critical tests

	Total	Failed	Passed	Pass %
Critical tests	5	5	0	0.0

All tests

	Total	Failed	Passed	Pass %
All tests	5	5	0	0.0

Zoom to changes Show only failed Show only critical all Max builds

[Show bigger image](#)

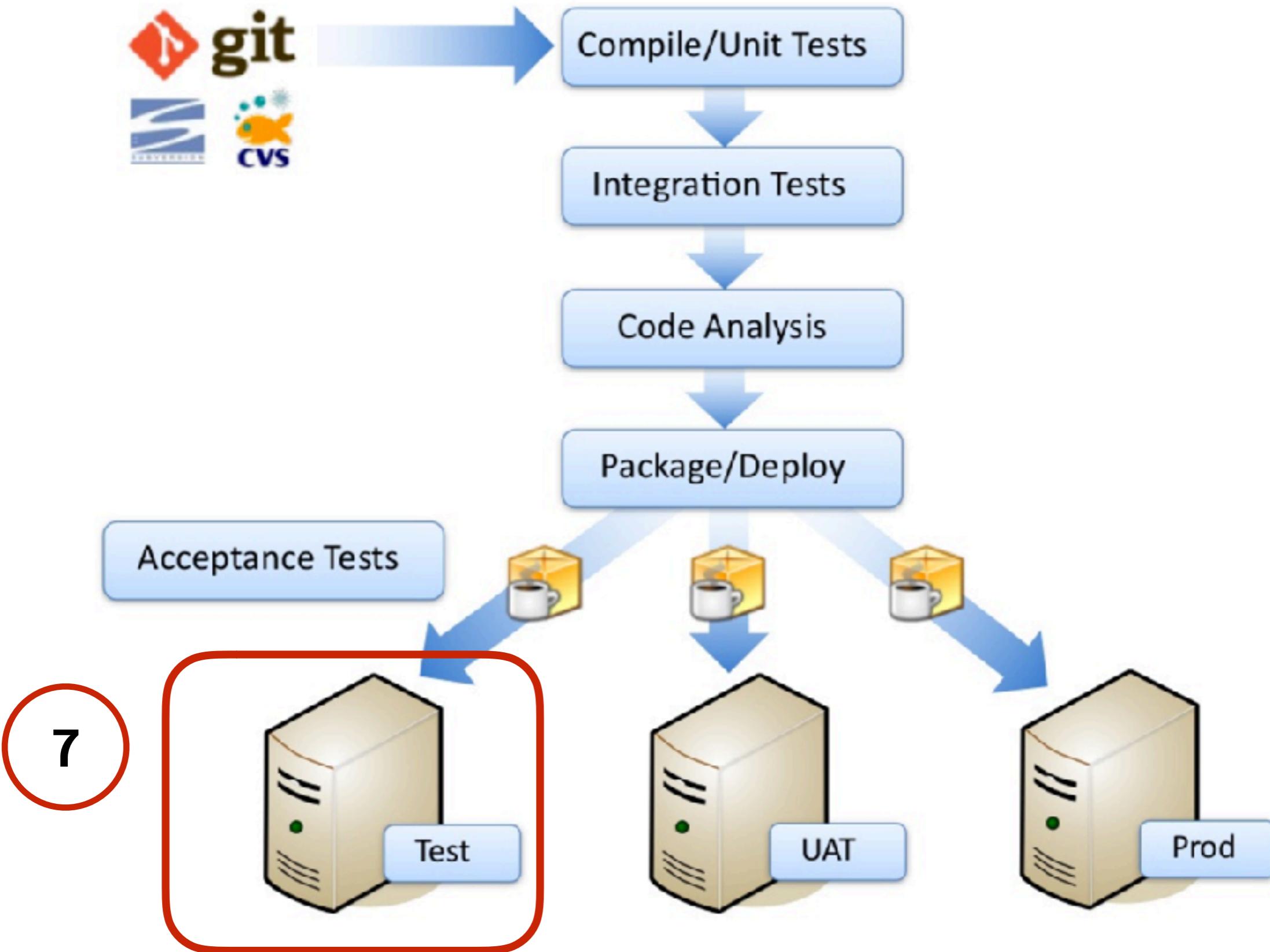
Latest Robot Results:

- [Browse results](#)
- [Open report.html](#)
- [Open log.html](#)



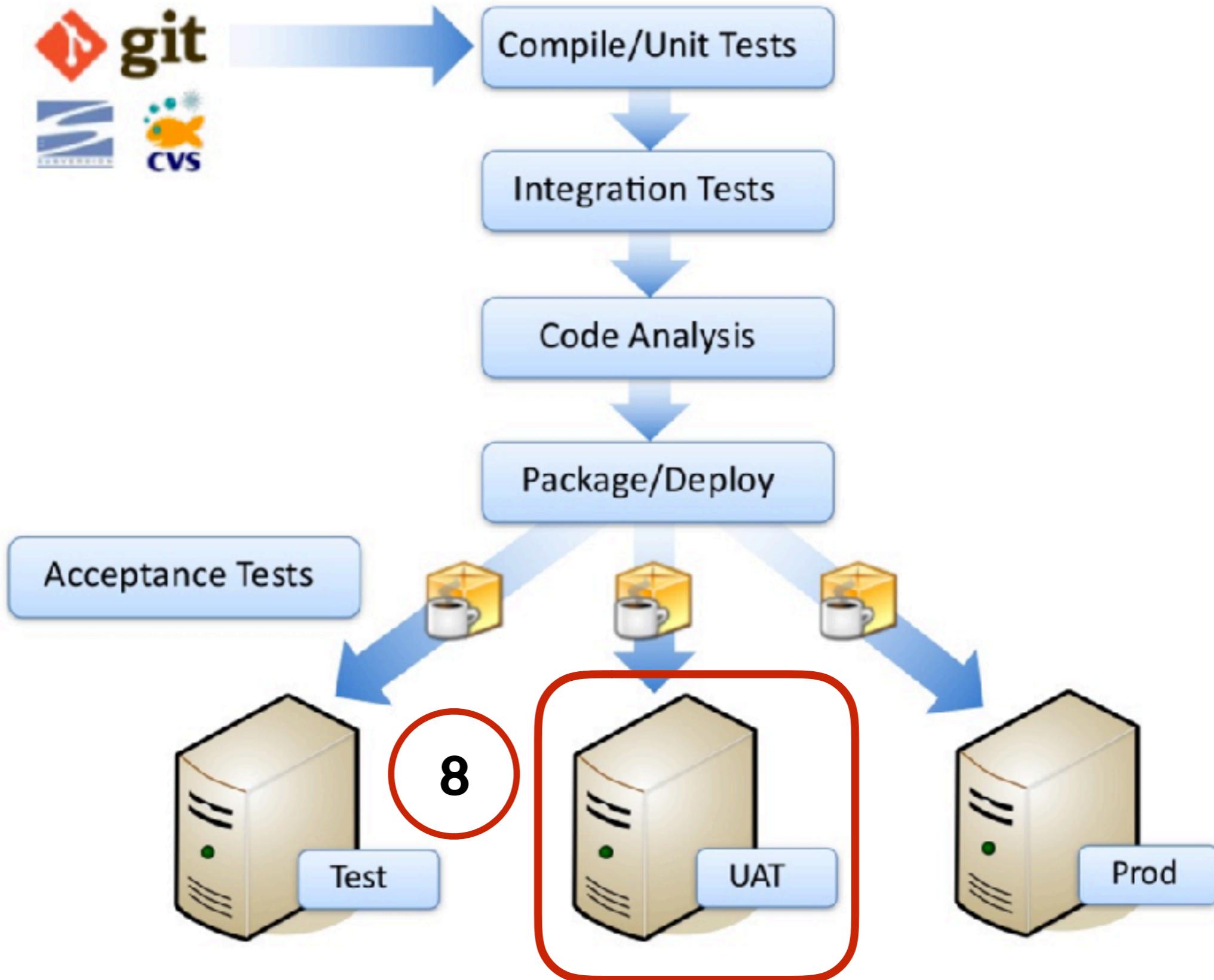
My build pipeline





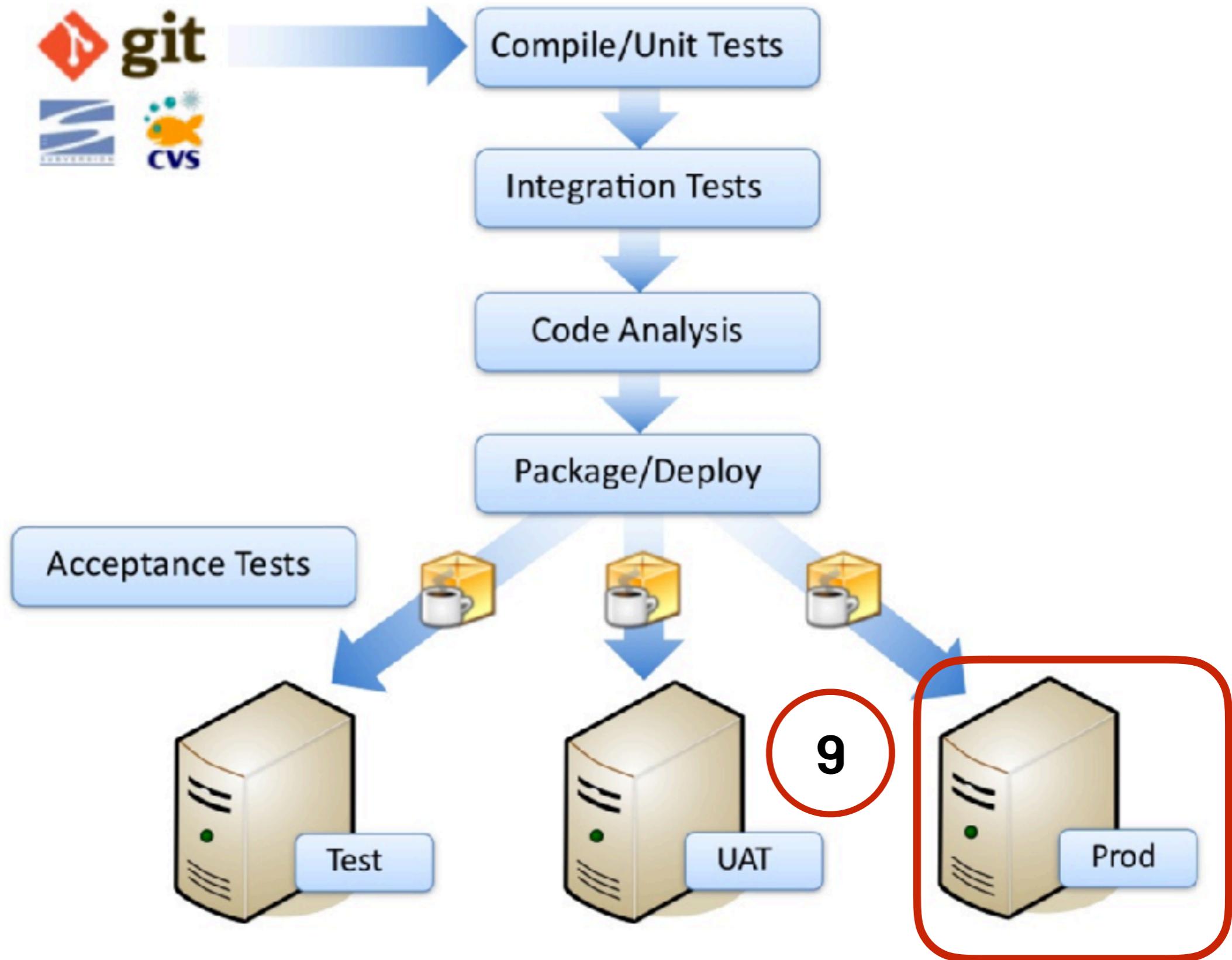
Deploy to Test ?





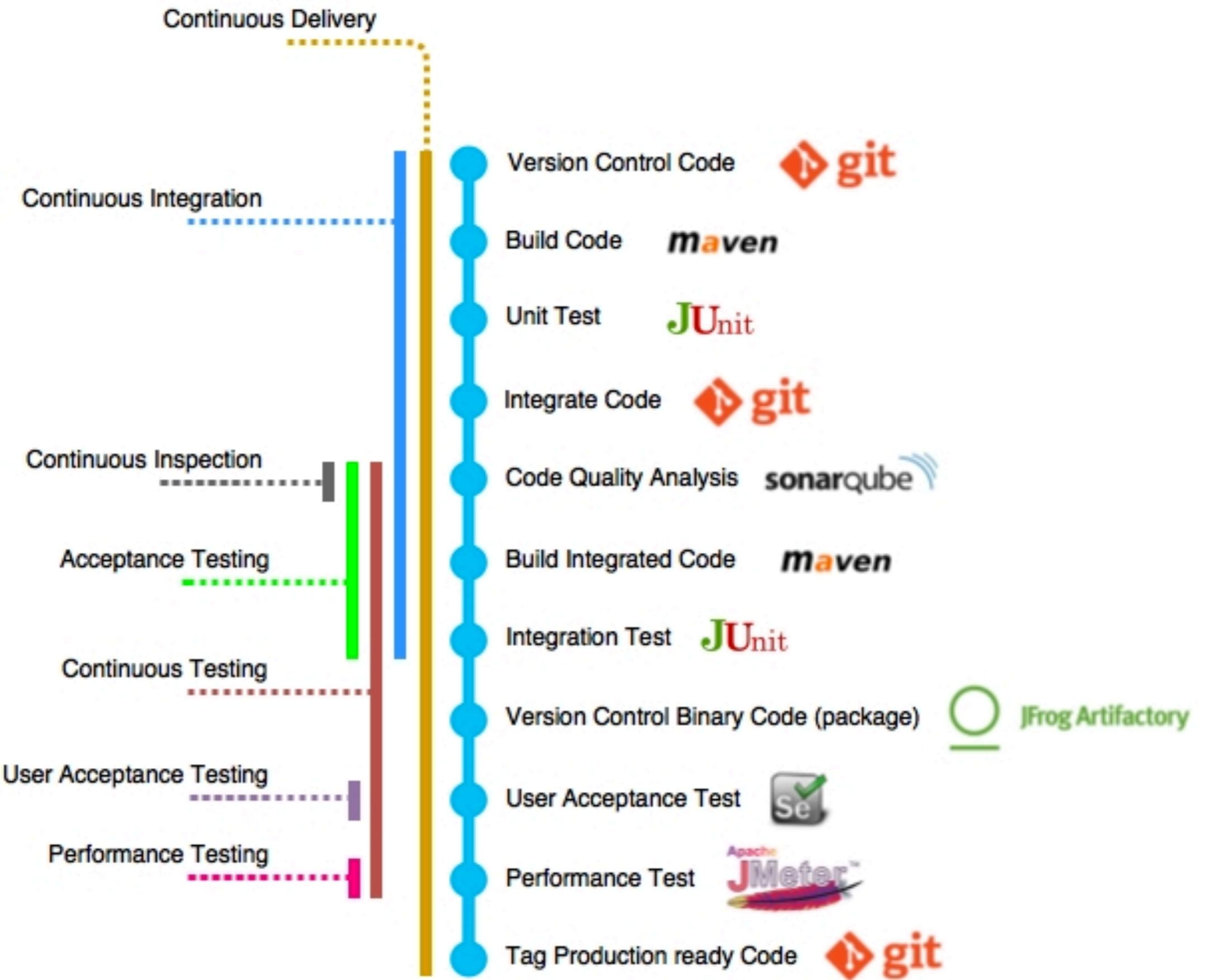
Deploy to UAT ?





Deploy to Production ?





What's next ?

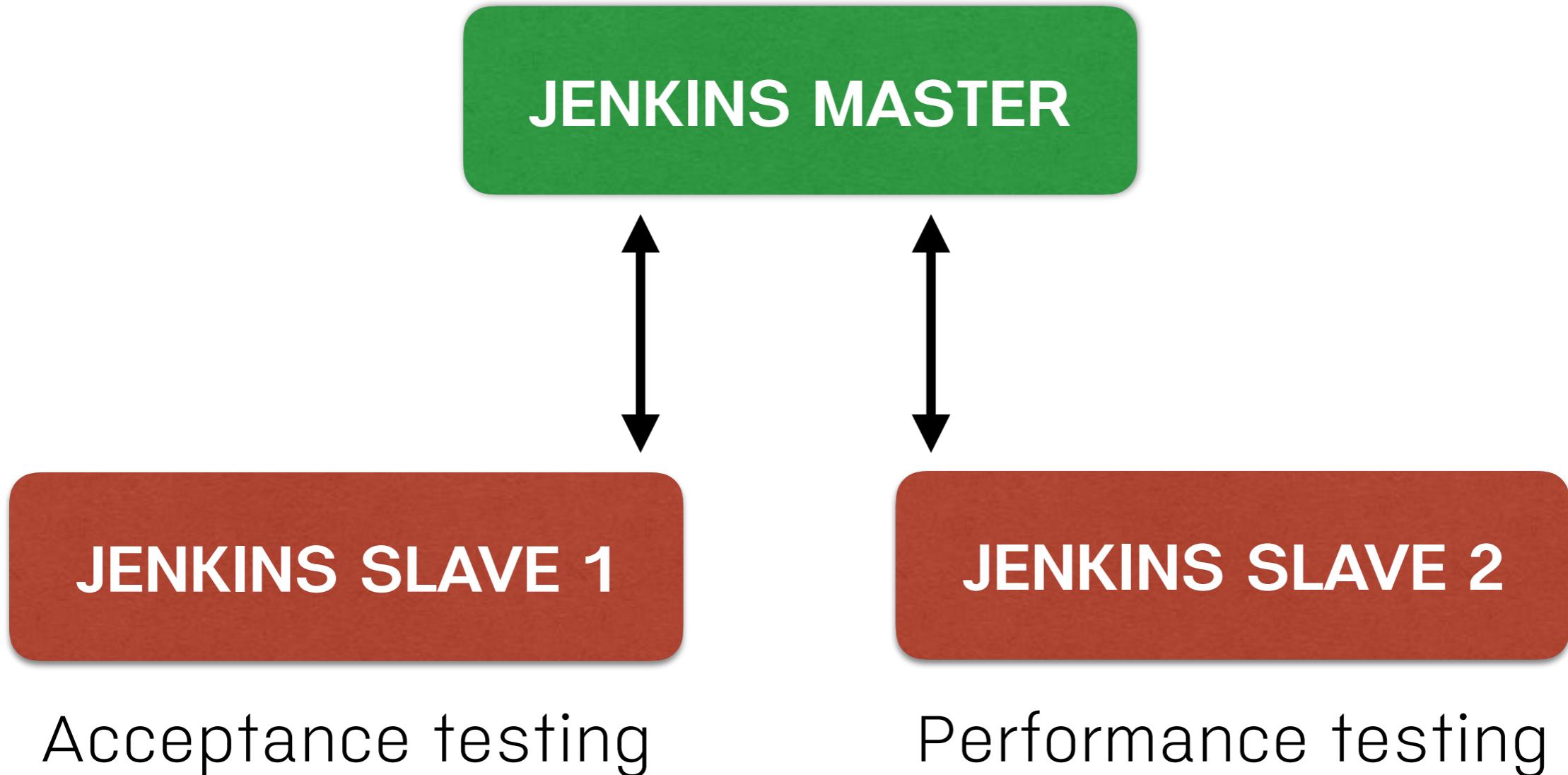


บริษัท สยามชนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

Performance test with JMeter

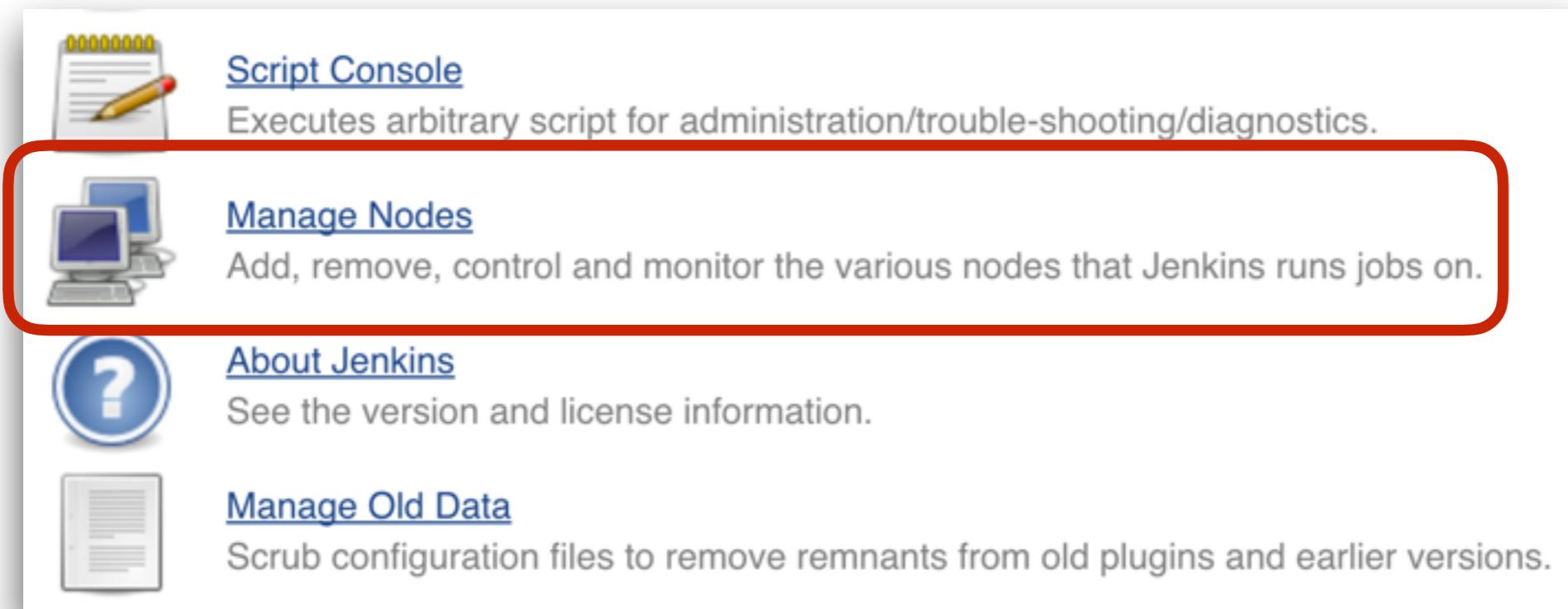


Jenkins master-slave



Add slave node

Manage Jenkins -> Manage Nodes



Current node

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time	
	master	Mac OS X (x86_64)	In sync	21.31 GB	1.26 GB	21.31 GB	0ms	
	Data obtained	22 min	22 min	22 min	22 min	22 min	22 min	<button>Refresh status</button>



New node

The screenshot shows the Jenkins web interface for creating a new node. The top navigation bar includes the Jenkins logo, a search bar, and user information for 'Somkiat'. The left sidebar has links for 'Back to Dashboard', 'Manage Jenkins', and 'New Node', with 'New Node' highlighted by a red box. The main content area shows a 'Node name' field containing 'testing'. A radio button labeled 'Permanent Agent' is selected, with a descriptive text explaining it adds a plain, permanent agent to Jenkins. Below this is an 'OK' button. On the left, there are two collapsed sections: 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing '1 Idle' and '2 Idle').

Jenkins

search

Somkiat | log out

Jenkins > Nodes >

Back to Dashboard

Manage Jenkins

New Node

Configure

Node name testing

Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

OK

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle



Edit node

Jenkins  Somkiat | log out

Jenkins > Nodes > testing

[Back to List](#) [Status](#) [Delete Agent](#) [Configure](#) [Build History](#) [Load Statistics](#) [Script Console](#) [Log](#) [System Information](#) [Disconnect](#)

Name: testing [?](#)

Description: [?](#)

of executors: 1 [?](#)

Remote root directory: /Users/somkiat/data/slide/ci-cd/swpark/node_testing [?](#)

Labels: testing [?](#)

Usage: Only build jobs with label expressions matching this node [?](#)

Launch method: Launch agent via execution of command on the master [?](#)

Launch command: java -jar /Users/somkiat/data/slide/ci-cd/swpark/slave.jar [?](#)

Availability: Keep this agent online as much as possible [?](#)

Build Executor Status [-](#)

1 Idle

Node Properties

Environment variables
 Tool Locations

[Save](#)

List of nodes

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time	
	master	Mac OS X (x86_64)	In sync	21.28 GB	1.02 GB	21.28 GB	0ms	
	testing	Mac OS X (x86_64)	In sync	21.28 GB	1.02 GB	21.28 GB	4025ms	
Data obtained	1 min 27 sec		1 min 27 sec	1 min 27 sec	1 min 26 sec	1 min 27 sec	1 min 27 sec	
								Refresh status

