

# CI and CD



# Jenkins



Somkiat Puisungnoen

Somkiat Puisungnoen

Update Info 2 View Activity Log 10+ ...

Timeline About Friends 2,604 Photos More ▾





← → C <https://www.facebook.com/somkiat.cc/>

 somkiat.cc 

Somkiat | Home 

**Page** Messages Notifications 1 Insights Publishing Tools Settings

  
somkiat.cc  
@somkiat.cc

**Home** About



# CI and CD



# Jenkins



# Agenda

Concept of Continuous Integration  
Concept of Continuous Delivery  
Build pipeline



# Agenda

All about Jenkins  
Installation and configuration  
Using plugins  
Setup build pipeline with Jenkins  
Build and deploy with Jenkins  
Perform testing with Jenkins  
Using metrics to improve quality



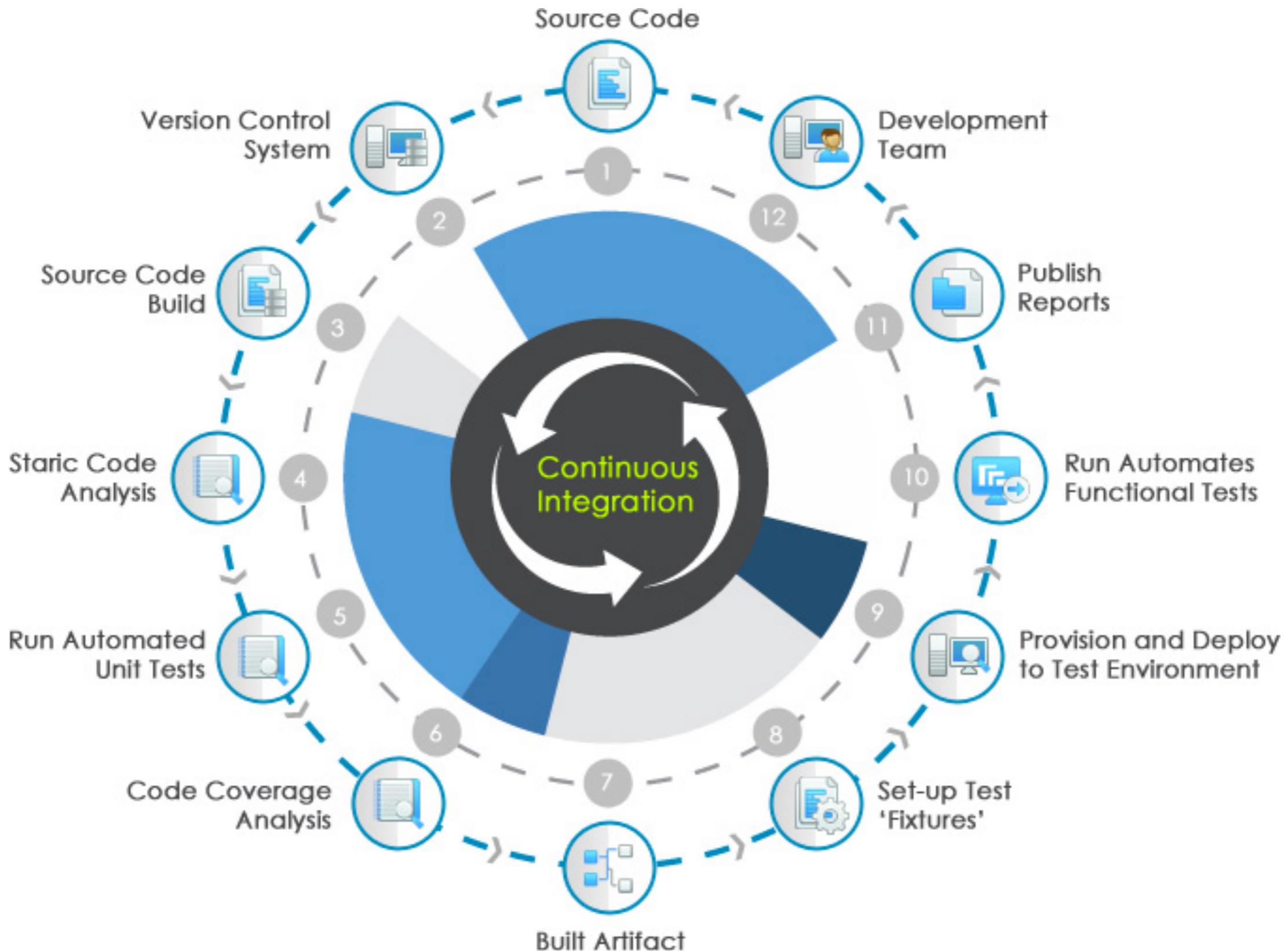
# Agenda

Backup and Restore data  
Understand the Master/Slave



# Continuous Integration







Jenkins

Bamboo



TeamCity

> go™



Hudson





Jenkins



Bamboo

CI is about what people do  
not about what tools they use



Hudson



# CI is a practice

**Discipline to integrate frequently**



# CI is a practice

Strive to make **small change**

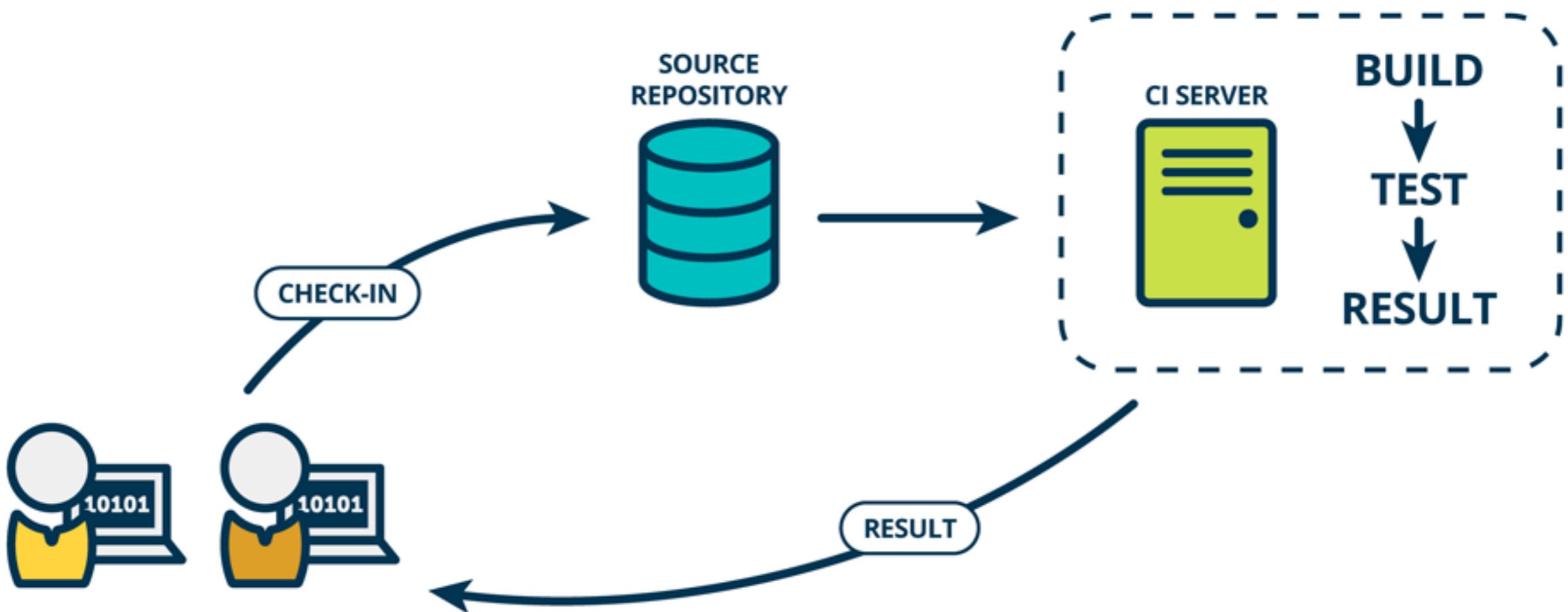


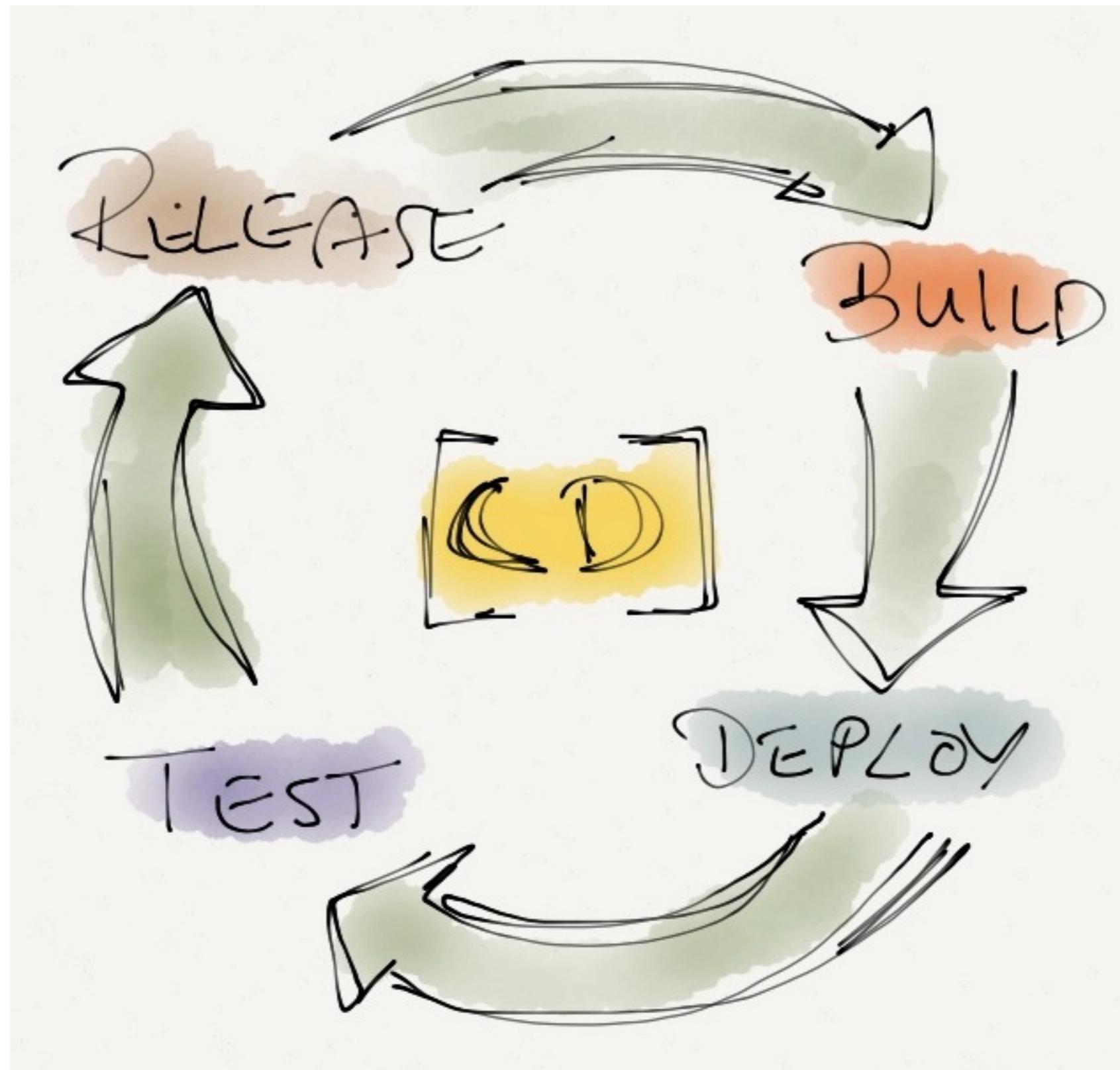
# CI is a practice

Strive for **fast feedback**



# Continuous Integration





# CONTINUOUS DELIVERY



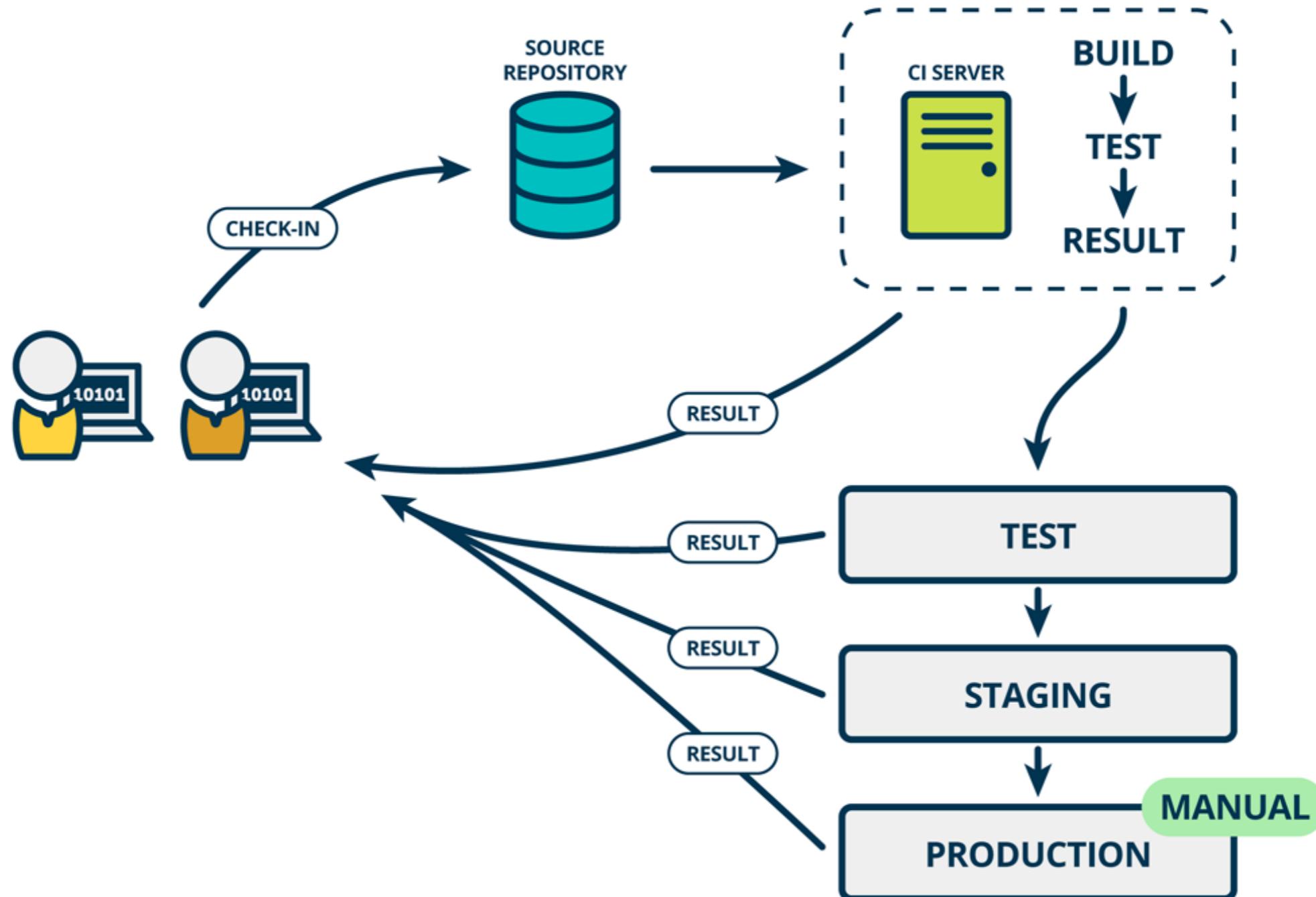
# CONTINUOUS DEPLOYMENT



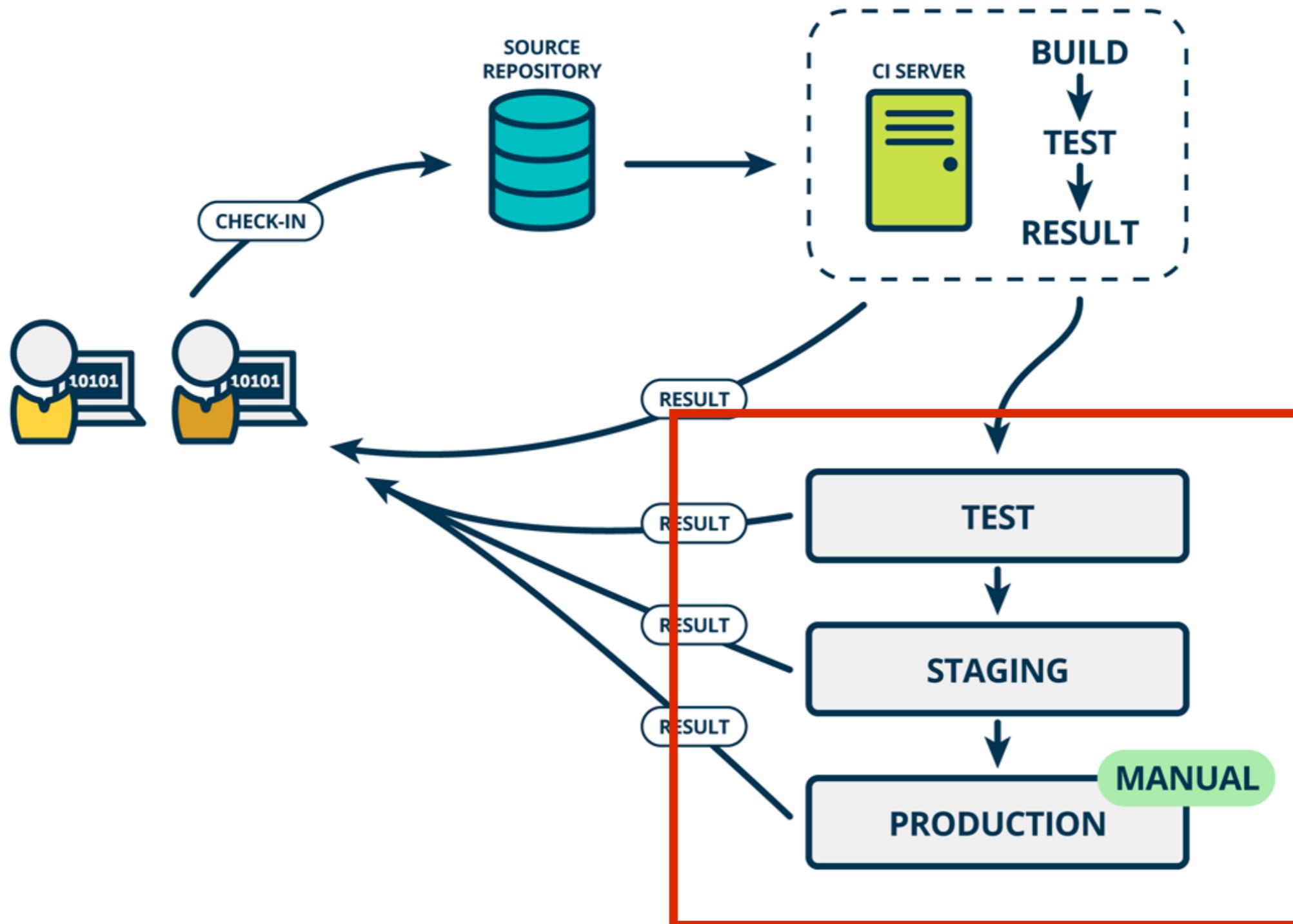
<http://blog.crisp.se/2013/02/05/yassalsundman/continuous-delivery-vs-continuous-deployment>



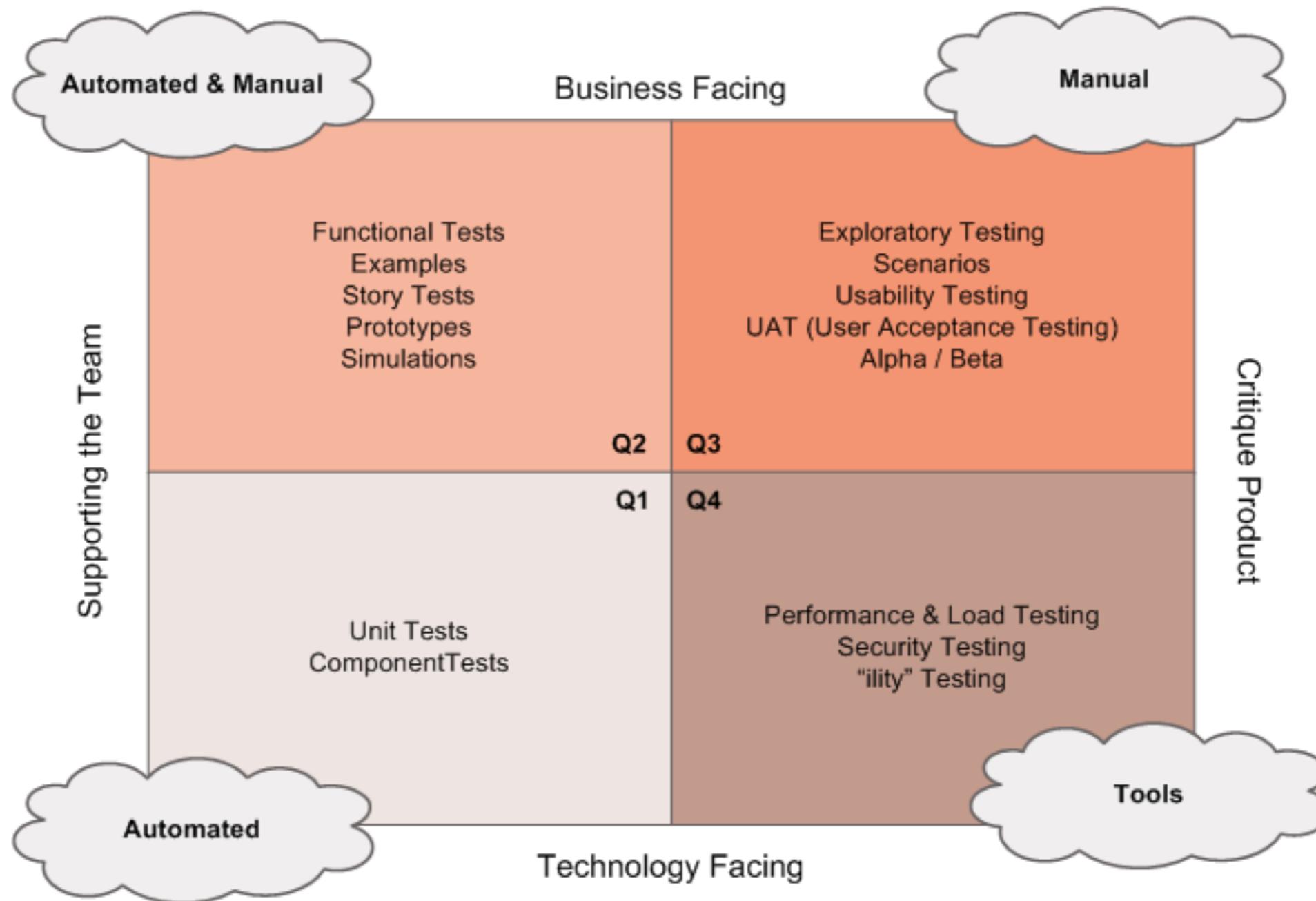
# Continuous Delivery



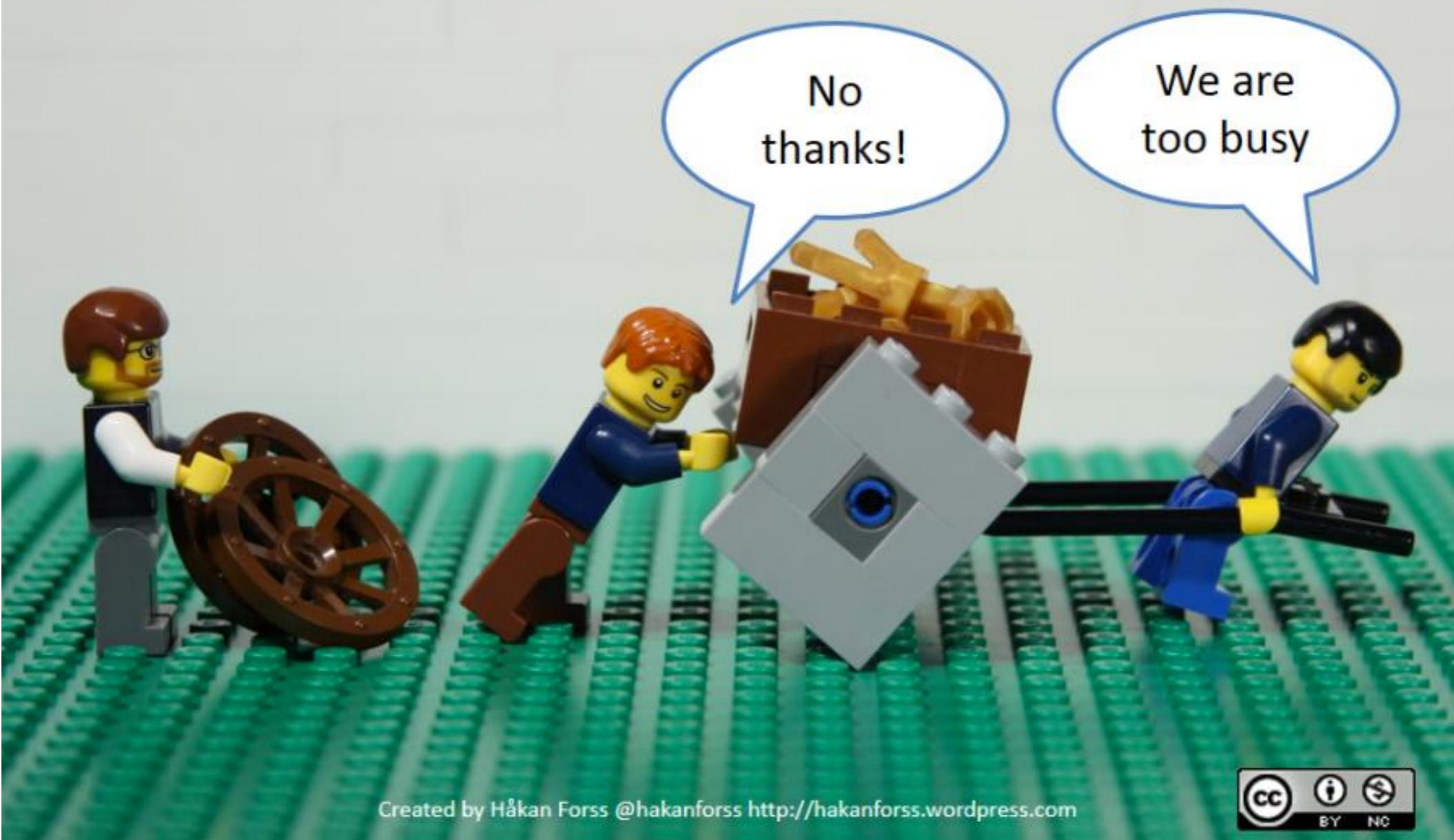
# Rising of DevOps



## Agile Testing Quadrants



# Are you too busy to improve?



# **Continuous Integration**

**is a software development practices**



# **1. Maintain a Single Source Repository**



## 2. Automate the Build



# 3. Make your build Self-Testing



**4. Everyone commit change  
to the mainline everyday**



# 5. Every commit should Build on an Integration Machine



# 6. Keep the build fast



# **7. Testing in a clone of the production environment**



**8. Make it easy for everyone  
to get the latest executable**



# 9. Everyone can see what's happening



# 10. Automate Deployment



# Good habits with CI



# **Developer should work in the private workspace**



# **Rebase frequently from the mainline**



# Check-in/push frequently



# Frequent build



# Automate the testing as much as possible



# **Don't check-in/push when the build is broken**



# Automate the deployment

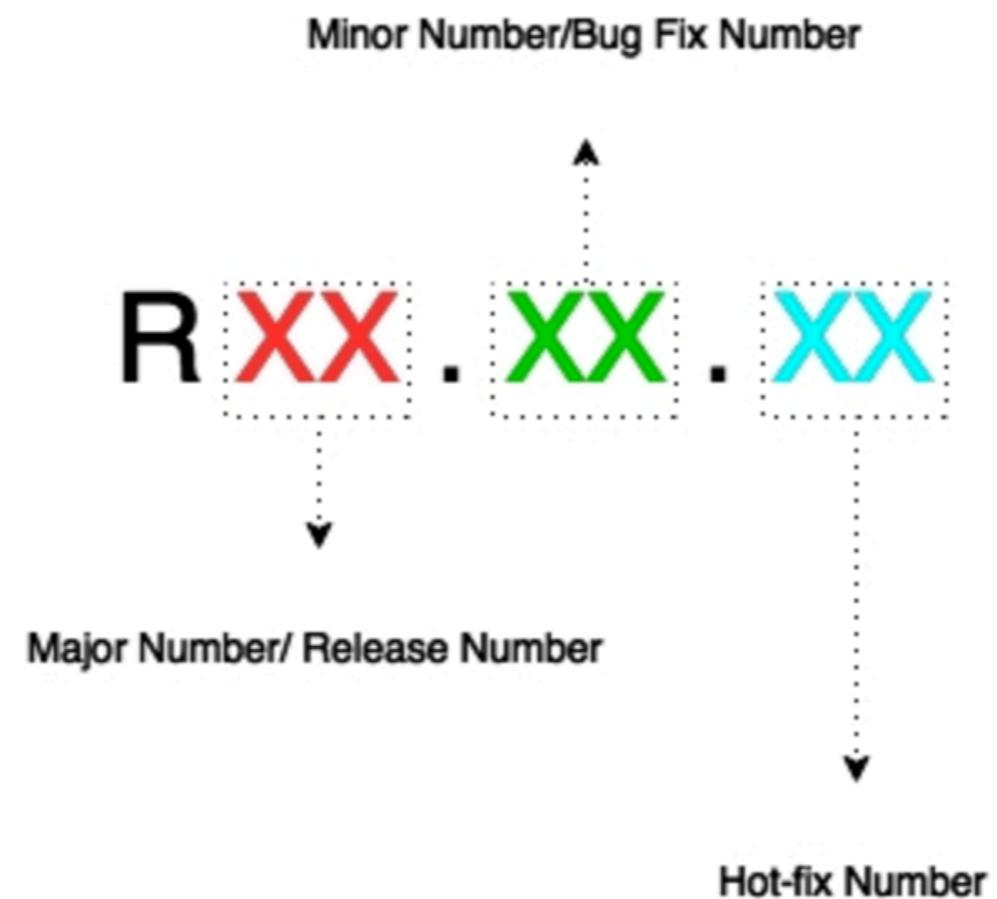


# Have a labeling strategy fore releases



# Instant notification





**Public Release Number**  
(New features require a  
new major number)

**Sprint number**  
the release was  
branched off

**Bug Fix Revision**  
(Incremental every time  
a bug-fix release is made)

v 1.12.1



**"Behind every successful agile project, there is a **Continuous Integration** server."**



# Let's start with Jenkins



# What is Jenkins ?

Application and framework  
manage and monitor  
the execution of **repeated tasks**



# Why Jenkins ?

Easy !!

Extensible

Scalable

Flexible

Open source

Community support

Lots of plugins

Cloud support



# Who use Jenkins ?

We thank the following organizations for their major commitments to support the Jenkins project.



redhat.

We thank the following organizations for their support of the Jenkins project through free and/or open source licensing programs.

Atlassian

Datadog

JFrog

Mac Cloud

PagerDuty

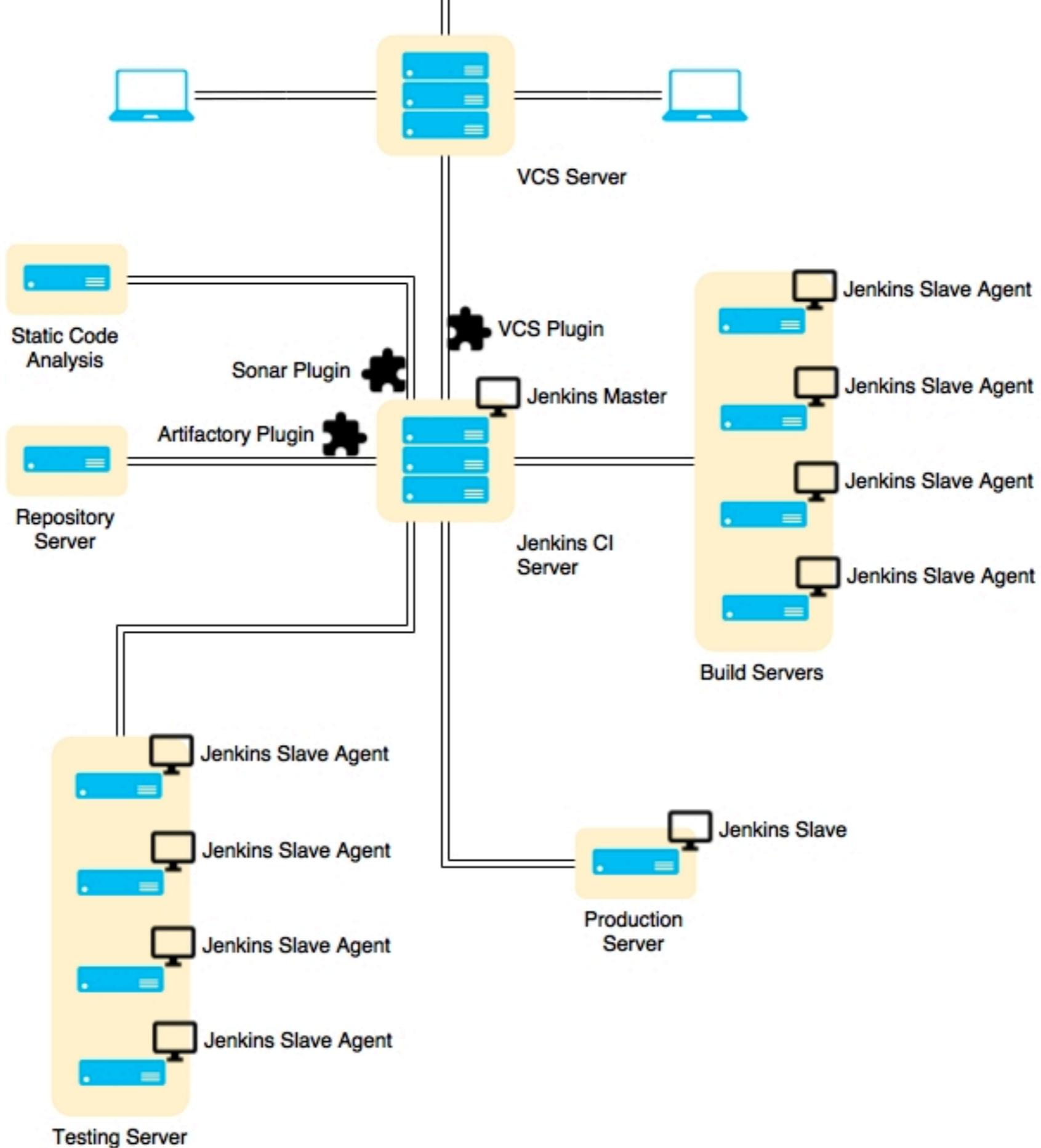
XMission

<https://wiki.jenkins-ci.org/pages/viewpage.action?pageId=58001258>



# Jenkins => Centralize CI Server





# Hardware requirements

## Jenkins master server

RAM +2GB and up to +60GB

More CPU

More disk space for job running

## Jenkins slave server

More RAM for connect to master server



# Setting up Jenkins



# Jenkins in container

Apache Tomcat

JBoss

Jetty

WebLogic

IBM Websphere

Glassfish



# Download



The Jenkins logo is a cartoon character with a large, round head and a wide, smiling mouth. It has short, dark hair and is wearing a red bow tie over a white shirt. It is dressed in a blue suit jacket and white cuffs. It is holding a white coffee cup in its right hand.

**Jenkins**

**Build great things at any scale**

The leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project.

[Documentation](#) [Download](#)

<https://jenkins.io/index.html>



# Use Long-term support (LTS)

## Getting started with Jenkins

The Jenkins project produces two release lines, LTS and weekly. Depending on your organization's needs, one may be preferred over the other.

Both release lines are distributed as `.war` files, native packages, installers, and Docker containers.

### Long-term Support (LTS)

LTS (Long-Term Support) releases are chosen every 12 weeks from the stream of regular releases as the stable release for that time period. [Learn more...](#)

[Changelog](#) | [Upgrade Guide](#) | [Past Releases](#)

 [Deploy Jenkins 2.46.3](#)

 [Deploy to Azure](#)

 [Download Jenkins 2.46.3 for:](#)

Docker

FreeBSD

### Weekly

A new release is produced weekly to deliver bug fixes and features to users and plugin developers.

[Changelog](#) | [Past Releases](#)

 [Download Jenkins 2.65 for:](#)

Arch Linux

Docker

FreeBSD

Gentoo



# Start Jenkins

```
$java -jar jenkins.war
```



# Change Port of Jenkins (8080)

```
$java -jar jenkins.war --httpPort=<port>
```



# Open in browser

http://localhost:8080

The screenshot shows the Jenkins 'Unlock Jenkins' setup page. At the top left, there's a 'Getting Started' link. Below it, the title 'Unlock Jenkins' is displayed in large, bold font. A text block explains that a password has been written to the log and a file on the server. It provides the file path: `/Users/somkiat/data/slide/ci-cd/swpark/software/keep/secrets/initialAdminPassword`. A note below says to copy the password from either location and paste it into the input field below. An 'Administrator password' input field is highlighted with a red border. At the bottom right, there's a 'Continue' button.

Getting Started

## Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

`/Users/somkiat/data/slide/ci-cd/swpark/software/keep/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

**Administrator password**

Continue



# Copy password from console

```
*****  
*****  
*****
```

Jenkins initial setup is required. An admin user has been created.

Please use the following password to proceed to installation:

a4b3a5231b8048419192d0c5afd3fce8

This may also be found at: /Users/somkiat/data/slide/ci-cd/swpa/initialAdminPassword

```
*****  
*****  
*****
```



# Custom your plug-ins

Getting Started

X

## Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

**Install suggested plugins**

Install plugins the Jenkins community finds most useful.

**Select plugins to install**

Select and install plugins most suitable for your needs.

Jenkins 2.46.3



# Waiting ...

## Getting Started

### Getting Started

<input type="radio"/> Folders Plugin	<input type="radio"/> OWASP Markup Formatter Plugin	<input type="radio"/> build timeout plugin	<input type="radio"/> Credentials Binding Plugin
<input type="radio"/> Timestamper	<input type="radio"/> Workspace Cleanup Plugin	<input type="radio"/> Ant Plugin	<input type="radio"/> Gradle Plugin
<input type="radio"/> Pipeline	<input type="radio"/> GitHub Organization Folder Plugin	<input type="radio"/> Pipeline: Stage View Plugin	<input type="radio"/> Git plugin
<input type="radio"/> Subversion Plug-in	<input type="radio"/> SSH Slaves plugin	<input type="radio"/> Matrix Authorization Strategy Plugin	<input type="radio"/> PAM Authentication plugin
<input type="radio"/> LDAP Plugin	<input type="radio"/> Email Extension Plugin	<input type="radio"/> Mailer Plugin	

\*\* - required dependency

Jenkins 2.46.3



# Success

Getting Started

## Installation Failures

Some plugins failed to install properly, you may retry installing them or continue with

✓ Folders Plugin	✓ OWASP Markup Formatter Plugin	✓ build timeout plugin	✓ Credentials Binding Plugin
✓ Timestamper	✓ Workspace Cleanup Plugin	✓ Ant Plugin	✓ Gradle Plugin
✓ Pipeline	✓ GitHub Organization Folder Plugin	✓ Pipeline: Stage View Plugin	✓ Git plugin
✓ Subversion Plug-in	✓ SSH Slaves plugin	✓ Matrix Authorization Strategy Plugin	✓ PAM Authentication plugin
✓ LDAP Plugin	✓ Email Extension Plugin	✓ Mailer Plugin	

Jenkins 2.46.3

[Continue](#)

[Retry](#)



# Create new user

Getting Started

## Create First Admin User

Username:

Password:

Confirm password:

Full name:

E-mail address:

Jenkins 2.46.3

[Continue as admin](#)

[Save and Finish](#)



# Ready !!

Getting Started

## Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

Jenkins 2.46.3



# Welcome to Jenkins

Jenkins  search  somkiat | log out [ENABLE AUTO REFRESH](#)

New Item  add description 

People 

Build History 

Manage Jenkins 

My Views 

Credentials 

**Welcome to Jenkins!**

Please [create new jobs](#) to get started.

**Build Queue**  
No builds in the queue.

**Build Executor Status**  
1 Idle  
2 Idle

Page generated: Jun 14, 2017 2:08:57 PM ICT [REST API](#) [Jenkins ver. 2.46.3](#)



# Create first Jenkins job



# 1. Create new job

The screenshot shows the Jenkins dashboard. At the top left is the Jenkins logo. The main header says "Jenkins". Below the header is a navigation menu with the following items:

- New Item (highlighted with a red oval)
- People
- Build History
- Manage Jenkins
- My Views
- Credentials

In the center, there is a large "Welcome to Jenkins!" message. Below it is a teal-colored box containing the text "Please create new jobs to get started." This text and the box are also highlighted with a red oval.

At the bottom left, there is a "Build Queue" section with the sub-section "No builds in the queue."



# 2. Fill in name

**Enter an item name**

hello

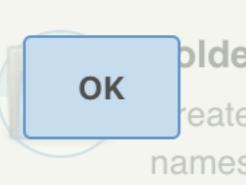
» Required field

 **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.

 **Pipeline**  
Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

 **External Job**  
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

 **Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

 **Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate space, so you can have multiple things of the same name as long as they are in different folders.



# 3. choose type

Enter an item name

hello

» 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.

## Pipeline

Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

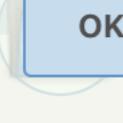
## External Job

This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

## Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK

 creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate space, so you can have multiple things of the same name as long as they are in different folders.



# 4. Config in General

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

Project name: hello

Description:

[Plain text] [Preview](#)

Discard old builds      [?](#)

GitHub project      [?](#)

This project is parameterized      [?](#)

Throttle builds      [?](#)

Disable this project      [?](#)

Execute concurrent builds if necessary      [?](#)

[Advanced...](#)

**Source Code Management**

**None**

**Save**      **Apply**



# See help !!

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

[Plain text] [Preview](#)

Discard old builds 

This determines when, if ever, build records for this project should be discarded. Build records include the console output, archived artifacts, and any other metadata related to a particular build.

Keeping fewer builds means less disk space will be used in the *Build Record Root Directory*, which is specified on the *Configure System* screen.

Jenkins offers two options for determining when builds should be discarded:

1. Build age: discard builds if they reach a certain age; for example, seven days old.
2. Build count: discard the oldest build if a certain number of builds already exist.

These two options can be active at the same time, so you can keep builds for 14 days, but only up to a limit of 50 builds, for example. If either limit is exceeded, then any builds beyond that limit will be discarded.

You can also ensure that important builds are kept forever, regardless of the setting here — click the *Keep this build forever* button on the build page.

The last stable and last successful build are also excluded from these rules.

---

In the *Advanced* section, the same options can be specified, but specifically for build **artifacts**. If enabled, build artifacts will be discarded for any builds which exceed the defined limits. The builds themselves will still be kept; only the associated artifacts, if any, will be deleted.

For example, if a project builds some software and produces a large installer, which is archived, you may wish to always keep the console log and information about which source control commit was built, while for disk space reasons, you may want to keep only the last three installers that were built.

This can make sense for projects where you can easily recreate the same artifacts later by building the same source control commit again.



# 5. Advance project options

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

Project name: hello

Description:

[Plain text] [Preview](#)

Discard old builds      [?](#)

GitHub project      [?](#)

This project is parameterized      [?](#)

Throttle builds      [?](#)

Disable this project      [?](#)

Execute concurrent builds if necessary      [?](#)

[Advanced...](#)

**Source Code Management**

Save      Apply

None



# Advance project options

General      Source Code Management      Build Triggers      Build Environment

- Quiet period
- Retry Count
- Block build when upstream project is building
- Block build when downstream project is building
- Use custom workspace

Display Name

- Keep the build logs of dependencies



# 7. Source code management

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

## Source Code Management

None  
 Git  
 Subversion

## Build Triggers

Trigger builds remotely (e.g., from scripts)  
 Build after other projects are built  
 Build periodically  
 GitHub hook trigger for GITScm polling  
 Poll SCM



# 8. Build Triggers

General    Source Code Management    **Build Triggers**    Build Environment

## Build Triggers

- Trigger builds remotely (e.g., from scripts)
- Build after other projects are built
- Build periodically
- GitHub hook trigger for GITScm polling
- Poll SCM



# Build periodically

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

## Build Triggers

- Trigger builds remotely (e.g., from scripts) ?
- Build after other projects are built ?
- Build periodically ?

Schedule

H 23 \* \* \*

⚠ No schedules so will never run

- GitHub hook trigger for GITScm polling ?
- Poll SCM ?

Run every day around 11.59 PM

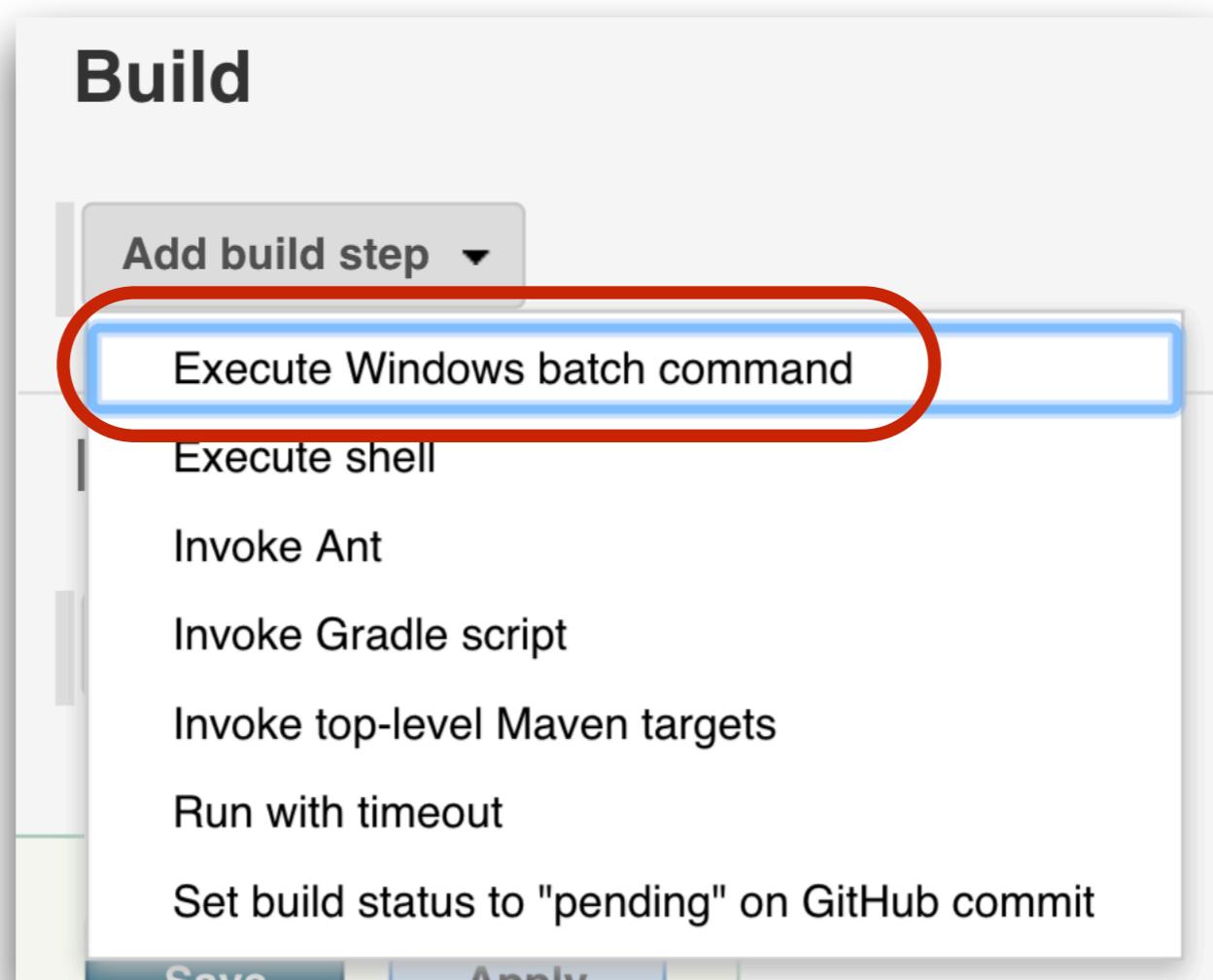


# 9. Add a Build step

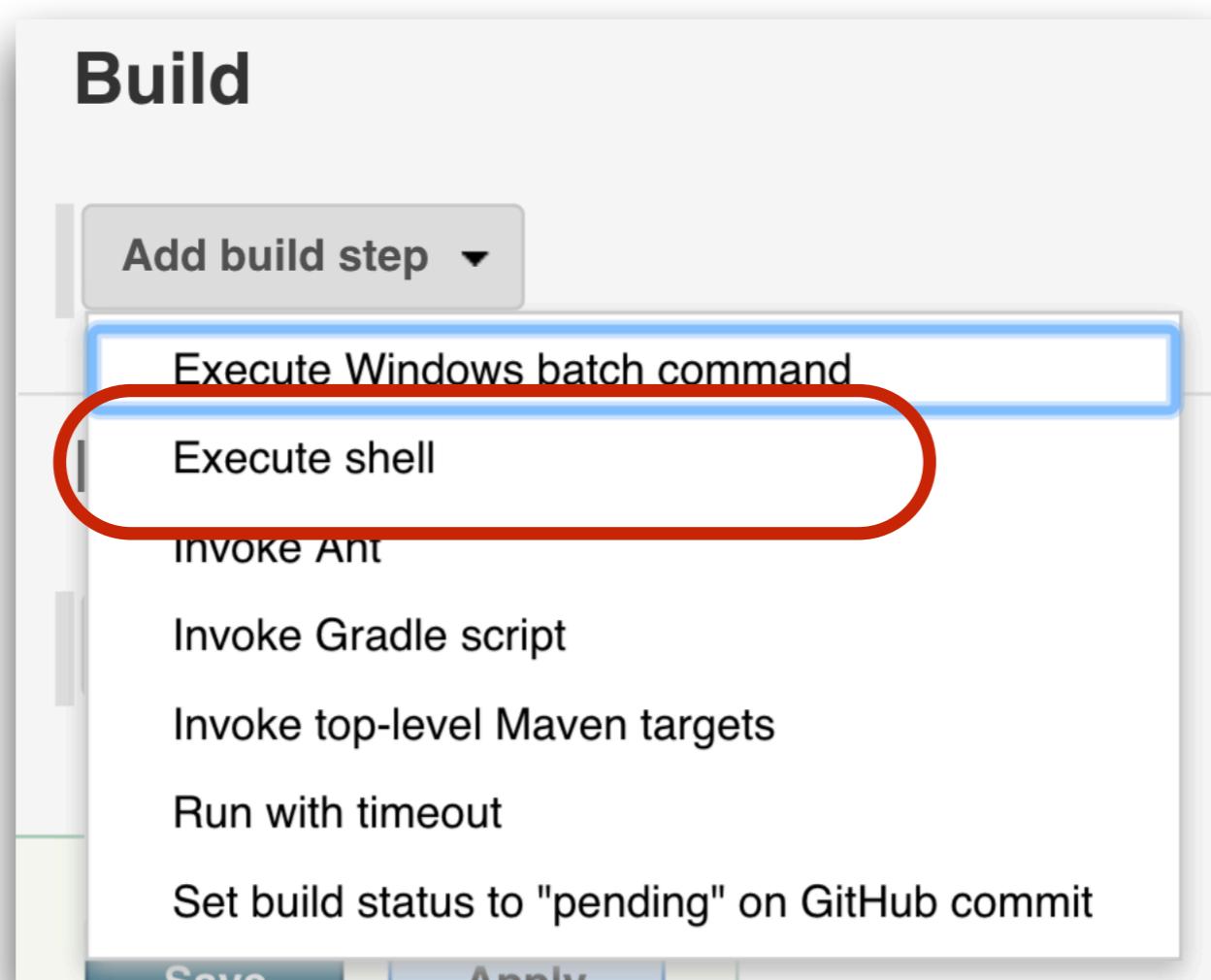
The screenshot shows a software interface for managing build configurations. At the top, there are tabs for General, Source Code Management, Build Triggers, Build Environment, **Build**, and Post-build Actions. The **Build** tab is currently active. Below the tabs, there's a section titled "BUILD ENVIRONMENT" containing four checkboxes: "Delete workspace before build starts", "Abort the build if it's stuck", "Add timestamps to the Console Output", and "Use secret text(s) or file(s)". Under the "Build" tab, there's a "Build" section with a "Add build step" button. A dropdown menu is open from this button, listing several options: "Execute Windows batch command" (which is highlighted with a blue border), "Execute shell", "Invoke Ant", "Invoke Gradle script", "Invoke top-level Maven targets", "Run with timeout", and "Set build status to "pending" on GitHub commit". At the bottom of the dropdown menu are two buttons: "Save" and "Apply".



# For Windows



# For UNIX/Mac



# 10. Post-build actions

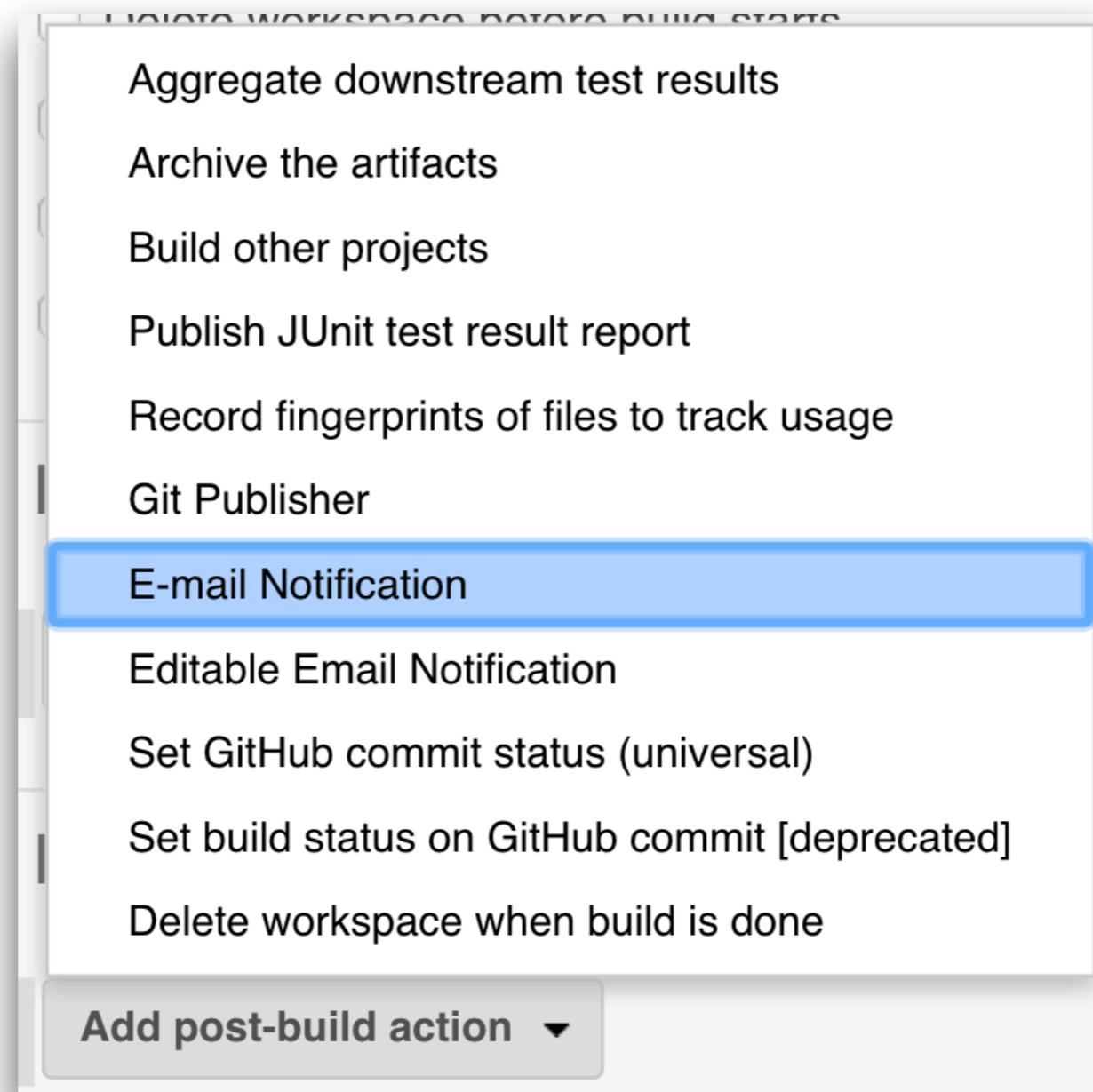
The screenshot shows a software interface for managing build configurations. The top navigation bar includes tabs for General, Source Code Management, Build Triggers, Build Environment, Build, and Post-build Actions. The Post-build Actions tab is currently active. Below the tabs, a section titled "BUILD ENVIRONMENT" is visible. On the left, a vertical sidebar lists several build actions: Delete workspace before build starts, Aggregate downstream test results, Archive the artifacts, Build other projects, Publish JUnit test result report, Record fingerprints of files to track usage, Git Publisher, E-mail Notification (which is highlighted with a blue background), Editable Email Notification, Set GitHub commit status (universal), Set build status on GitHub commit [deprecated], and Delete workspace when build is done. At the bottom left of the main area, there is a button labeled "Add post-build action ▾". In the bottom right corner of the main area, there are two buttons: "Save" and "Apply".



# Email notification



# Email notification



# Add Recipients

**Post-build Actions**

**E-mail Notification**

**Recipients**

Whitespace-separated list of recipient addresses. May reference build parameters like \$PARAM. E-mail will be sent when a build fails, becomes unstable or returns to stable.

Send e-mail for every unstable build

Send separate e-mails to individuals who broke the build

[?](#)

**Add post-build action ▾**



# Configure SMTP server

Manage Jenkins -> Configure System

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 highlighted with a red circle and the 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, under the heading 'Manage Jenkins', there are five configuration options: 'Configure System' (highlighted with a red circle and the number 2), 'Configure Global Security', 'Configure Credentials', 'Global Tool Configuration', and 'Reload Configuration from Disk'. Each option has a brief description below it.

Option	Description
<a href="#">Configure System</a>	Configure global settings and paths.
<a href="#">Configure Global Security</a>	Secure Jenkins; define who is allowed to access/use
<a href="#">Configure Credentials</a>	Configure the credential providers and types
<a href="#">Global Tool Configuration</a>	Configure tools, their locations and automatic instal
<a href="#">Reload Configuration from Disk</a>	Discard all the loaded data in memory and reload eve



# Configure SMTP server

**E-mail Notification**

---

SMTP server  

Default user e-mail suffix  

 **Advanced...**

Test configuration by sending test e-mail

---

**Save** **Apply**



# Configure SMTP server

**E-mail Notification**

---

SMTP server	<input type="text"/>	<a href="#">?</a>
Default user e-mail suffix	<input type="text"/>	<a href="#">?</a>
<input type="checkbox"/> Use SMTP Authentication	<input type="checkbox"/>	<a href="#">?</a>
Use SSL	<input type="checkbox"/>	<a href="#">?</a>
SMTP Port	<input type="text"/>	<a href="#">?</a>
Reply-To Address	<input type="text"/>	
Charset	<input type="text" value="UTF-8"/>	
<input type="checkbox"/> Test configuration by sending test e-mail		

---

**Action Buttons:**

**Save** **Apply**



# 11. Run your job !!

Jenkins search somkiat | log out

ENABLE AUTO REFRESH

New Item People Build History Manage Jenkins My Views Credentials

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration
		<a href="#">hello</a>	N/A	N/A	

Icon: [S](#) [M](#) [L](#)

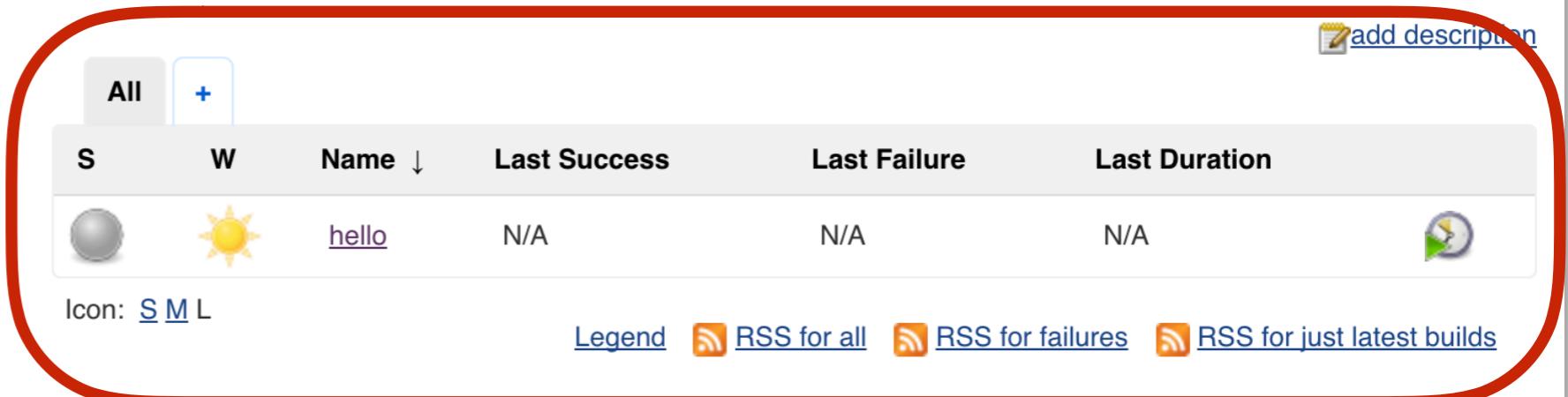
Legend [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

Build Queue

No builds in the queue.

Build Executor Status

1 Idle  
2 Idle



# Job status

The screenshot shows a Jenkins job status page. At the top, there are buttons for 'All' and '+'. To the right is a 'add description' button with a pencil icon. Below the buttons is a table header with columns: S, W, Name ↓, Last Success, Last Failure, and Last Duration. A row of data follows: a gray circle icon (circled in red), a yellow sun icon, the name 'hello', and three 'N/A' entries for the last success, failure, and duration. To the right of the row is a small green and yellow icon. Below the table, the text 'Icon: S M L' is displayed. At the bottom of the screenshot, there is a legend with three RSS feed icons and their corresponding links: 'RSS for all', 'RSS for failures', and 'RSS for just latest builds'.

S	W	Name ↓	Last Success	Last Failure	Last Duration
		hello	N/A	N/A	N/A

Icon: [S](#) [M](#) [L](#)

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)

**Blue** = success

**Red** = failure

**Gray** = disabled/never execute



# Job health

[!\[\]\(8c77d3b223dda6d93793a923f91fed36\_img.jpg\) add description](#)

All [+](#)

S	W	Name ↓	Last Success	Last Failure	Last Duration
		<a href="#">hello</a>	N/A	N/A	N/A 

Icon: [S](#) [M](#) [L](#)

[Legend](#)  [RSS for all](#)  [RSS for failures](#)  [RSS for just latest builds](#)

**Sunny** = 100% success rate

**Cloudy** = 60% success rate

**Raining** = 40% success rate



# Job name

[add description](#)

All

S	W	Name ↓	Last Success	Last Failure	Last Duration
		<a href="#">hello</a>	N/A	N/A	N/A

Icon: [S](#) [M](#) [L](#)

[Legend](#) [RSS for all](#) [RSS for failures](#) [RSS for just latest builds](#)



# Build job

[!\[\]\(c9eeb7e2f20788628e80272a2841f0be\_img.jpg\) add description](#)

All [!\[\]\(b41facf15a3e2841562827ebcb83c652\_img.jpg\)](#)

S	W	Name ↓	Last Success	Last Failure	Last Duration	
		<a href="#">hello</a>	N/A	N/A	N/A	

Icon: [S](#) [M](#) [L](#)

[Legend](#)  [RSS for all](#)  [RSS for failures](#)  [RSS for just latest builds](#)



# Jenkins build log

The screenshot shows the Jenkins interface for a build named "hello". The build was last successful 4.9 sec ago (#1) and has not failed. The "Last Duration" is 0.23 sec. A context menu is open over the successful build entry, listing options: Changes, Console Output (which is highlighted), Edit Build Information, and Delete Build. The "Console Output" option is currently selected.

S	W ↓	Name	Last Success	Last Failure	Last Duration
		<a href="#">hello</a>	4.9 sec - #1	N/A	0.23 sec

Icon: [S](#) [M](#) [L](#)

[Changes](#)  
[Console Output](#)  
[Edit Build Information](#)  
[Delete Build](#)

[SS for failures](#) [RSS for just latest builds](#)



# Console output



## Console Output

Started by user [somkiat](#)

Building in workspace /Users/somkiat/data/slide/ci-cd/swpark/

Finished: SUCCESS



# Jenkins Home directory



# Default Jenkins Home

Manage Jenkins -> Configure System

The screenshot shows the Jenkins home page. On the left, there is a sidebar with several links: 'New Item', 'People', 'Build History', 'Manage Jenkins' (which is highlighted with a red circle labeled '1'), 'My Views', and 'Credentials'. Below this is a 'Build Queue' section stating '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'. It contains five items: 'Configure System' (highlighted with a red circle labeled '2'), 'Configure Global Security', 'Configure Credentials', 'Global Tool Configuration', and 'Reload Configuration from Disk'. The 'Configure System' link has a detailed description below it.

Link	Description
<a href="#">Configure System</a>	Configure global settings and paths.
<a href="#">Configure Global Security</a>	Secure Jenkins; define who is allowed to access/use
<a href="#">Configure Credentials</a>	Configure the credential providers and types
<a href="#">Global Tool Configuration</a>	Configure tools, their locations and automatic instal
<a href="#">Reload Configuration from Disk</a>	Discard all the loaded data in memory and reload eve



# Default Jenkins Home

Jenkins configuration

New Item      Home directory /Users/somkiat/data/slide/ci-cd/swpark/software/keep      Advanced...

People      System Message

Build History      # of executors 2

Manage Jenkins      Labels

My Views      Usage Use this node as much as possible

Credentials      Quiet period 5

Build Queue      SCM checkout retry count 0

Build Executor Status      Restrict project naming

Global properties

Environment variables

Save      Apply

A red box highlights the "Home directory" section.



# Default Jenkins Home

Jenkins configuration

New Item    People    Build History    Manage Jenkins    My Views    Credentials

Build Queue: No builds in the queue.

Build Executor Status: 1 Idle, 2 Idle

System Message: [Plain text] [Preview](#)

Home directory: /Users/somkiat/data/slide/ci-cd/swpark/software/keep

Workspace Root Directory: \${JENKINS\_HOME}/workspace/\${ITEM\_FULLNAME}

Build Record Root Directory: \${ITEM\_ROOTDIR}/builds

# of executors: 2

Labels:

Usage: Use this node as much as possible

Quiet period: 5

The "Home directory", "Workspace Root Directory", and "Build Record Root Directory" fields are highlighted with a red border.



# Change Jenkins's Home

## For Windows

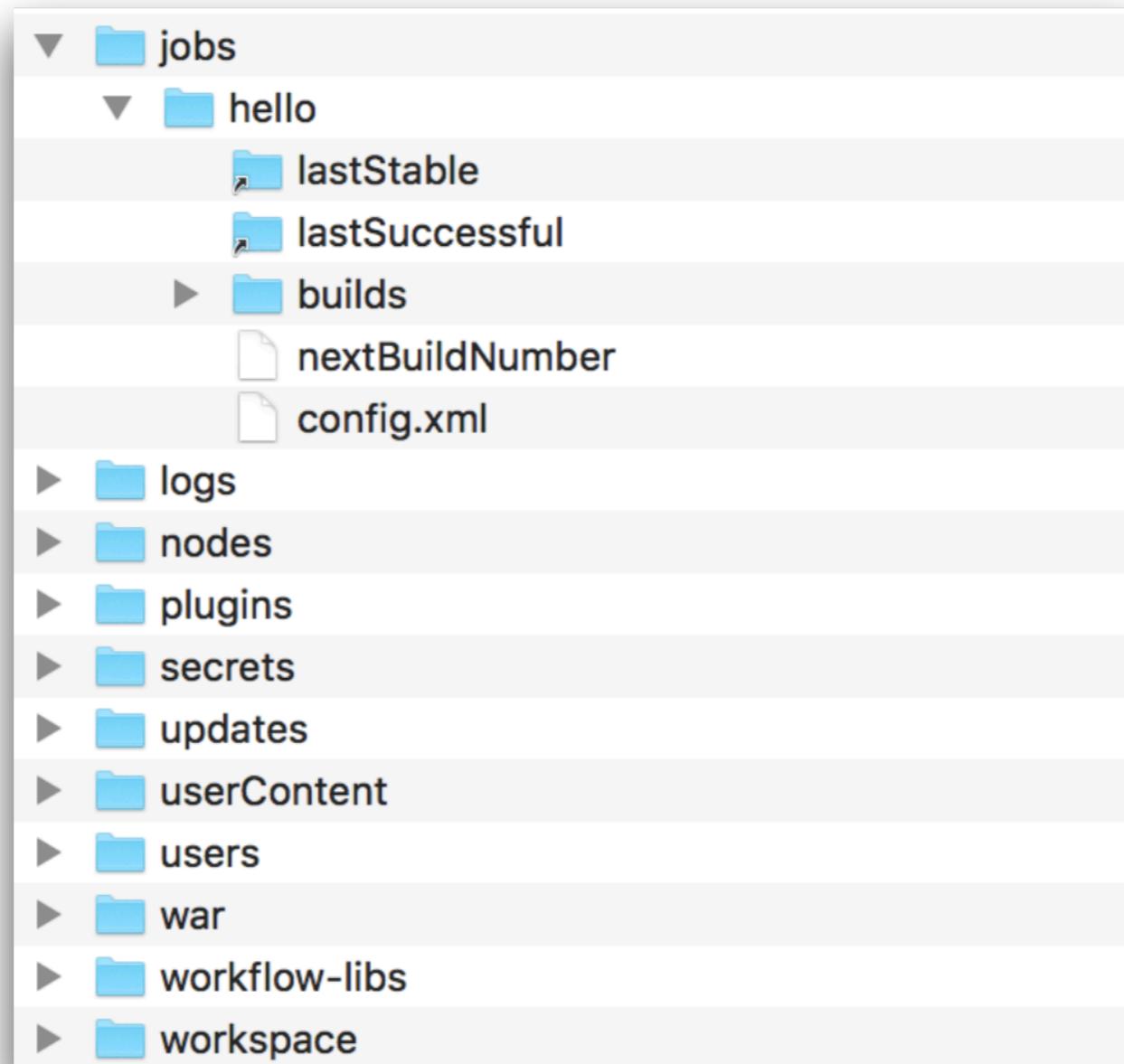
```
$set JENKINS_HOME=<your path>
```

## For UNIX/MAC

```
$export JENKINS_HOME=<your path>
```



# Jenkins home structure



# Jenkins Job

## **config.xml**

+ jenkins job configuration

## **workspace folder**

+ contains the output/content of build

## **builds folder**

+ log information of all builds



# Manage & Config plugins



# Jenkins Plugins

Power from plugins  
+1000 community

<https://jenkins.io/doc/book/managing/plugins/>



# Find your plugins



The screenshot shows the Jenkins Plugins Index page. At the top, there is a navigation bar with links: Jenkins, Blog, Documentation, Plugins (which is underlined in red), Use-cases ▾, Participate, Sub-projects ▾, and Resources ▾. Below the navigation bar is a large blue header section. On the left side of the header is a circular icon featuring a cartoon character wearing a white flight helmet and goggles, surrounded by red and grey mechanical components. To the right of the icon, the text "Plugins Index" is displayed in a large, white, sans-serif font. Below this title, a subtitle reads: "Discover the 1000+ community contributed Jenkins plugins to support building, deploying and automating any project." At the bottom of the blue header is a search bar with a red background. The search bar contains the text "Find plugins..." and a magnifying glass icon on the right end.

<https://plugins.jenkins.io/>



# Manage Plugins

Manage Jenkins -> Manage Plugins

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links: New Item, People, Build History, **Manage Jenkins**, My Views, and Credentials. The 'Manage Jenkins' link is circled with a red circle labeled '1'. Below the sidebar, there are two sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). On the right, under the heading 'Manage Jenkins', there are several configuration options: Configure System, Configure Global Security, Configure Credentials, Global Tool Configuration, Reload Configuration from Disk, Manage Plugins, System Information, System Log, and Load Statistics. The 'Manage Plugins' link is circled with a red circle labeled '2'.

Jenkins

search ? somkiat | log out

ENABLE AUTO REFRESH

New Item

People

Build History

**Manage Jenkins**

My Views

Credentials

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Configure System

Configure Global Security

Configure Credentials

Global Tool Configuration

Reload Configuration from Disk

**Manage Plugins**

System Information

System Log

Load Statistics

Manage Jenkins



# Plugin manager

Jenkins  somkiat | log out

search

Jenkins ▶ Plugin Manager

Back to Dashboard Manage Jenkins Update Center

Updates Available Installed Advanced

Install	Name ↓	Version	Installed
No updates			

Update information obtained: 9 hr 35 min ago [Check now](#)

Select: [All](#), [None](#)  
This page lists updates to the plugins you currently use.  
Disabled rows are already upgraded, awaiting restart. Shaded but selectable rows are [in progress or failed](#).



# Plugin manager

Filter:

Updates Available Installed Advanced

Install ↓	Name	Version
.NET Development		
<a href="#">CCM Plug-in</a>	<input type="checkbox"/> This plug-in generates the trend report for CCM, an open source static code analysis program.	3.1
<a href="#">FxCop Runner plugin</a>	<input type="checkbox"/>	1.1
<a href="#">MSBuild Plugin</a>	<input type="checkbox"/>	1.27
<a href="#">MSTest plugin</a>	<input type="checkbox"/> Generates test reports for MSTest.	0.19
<a href="#">MSTestRunner plugin</a>	<input type="checkbox"/>	1.3.0
<a href="#">NAnt Plugin</a>	<input type="checkbox"/>	1.4.3
<a href="#">NCover plugin</a>	<input type="checkbox"/>	0.3
<a href="#">PowerShell plugin</a>	<input type="checkbox"/>	1.3
<a href="#">Violation Comments to Bitbucket Server Plugin</a>	<input type="checkbox"/> Finds violations reported by code analyzers and comments Bitbucket Server (or Stash) pull requests (or commits) with them.	1.50
<a href="#">Violations plugin</a>	<input type="checkbox"/>	0.7.11

Install without restart   Download now and install after restart   Update information obtained: 9 hr 37 min ago   Check now



# Updated tab

List of updates for the plugins installed  
on the current Jenkins instance



# Available tab

List of all the plugins available  
from Jenkins community



# Installed tab

List of all the plugins installed  
on the current Jenkins instance



# Advanced tab

Configure internet settings  
and update Jenkins plugins manually



# Advanced tab

Updates Available Installed Advanced

## HTTP Proxy Configuration

Server  ?

Port  ?

User name  ?

Password

No Proxy Host  ?

**Advanced...**

**Submit**



# Advanced tab

## Upload Plugin

You can upload a .hpi file to install a plugin from outside the central plugin repository.

File:  Choose File No file chosen

**Upload**

## Update Site

URL  <http://updates.jenkins-ci.org/update-center.json>

**Submit**

<https://updates.jenkins-ci.org/download/plugins/>



# Install Jenkins plugins



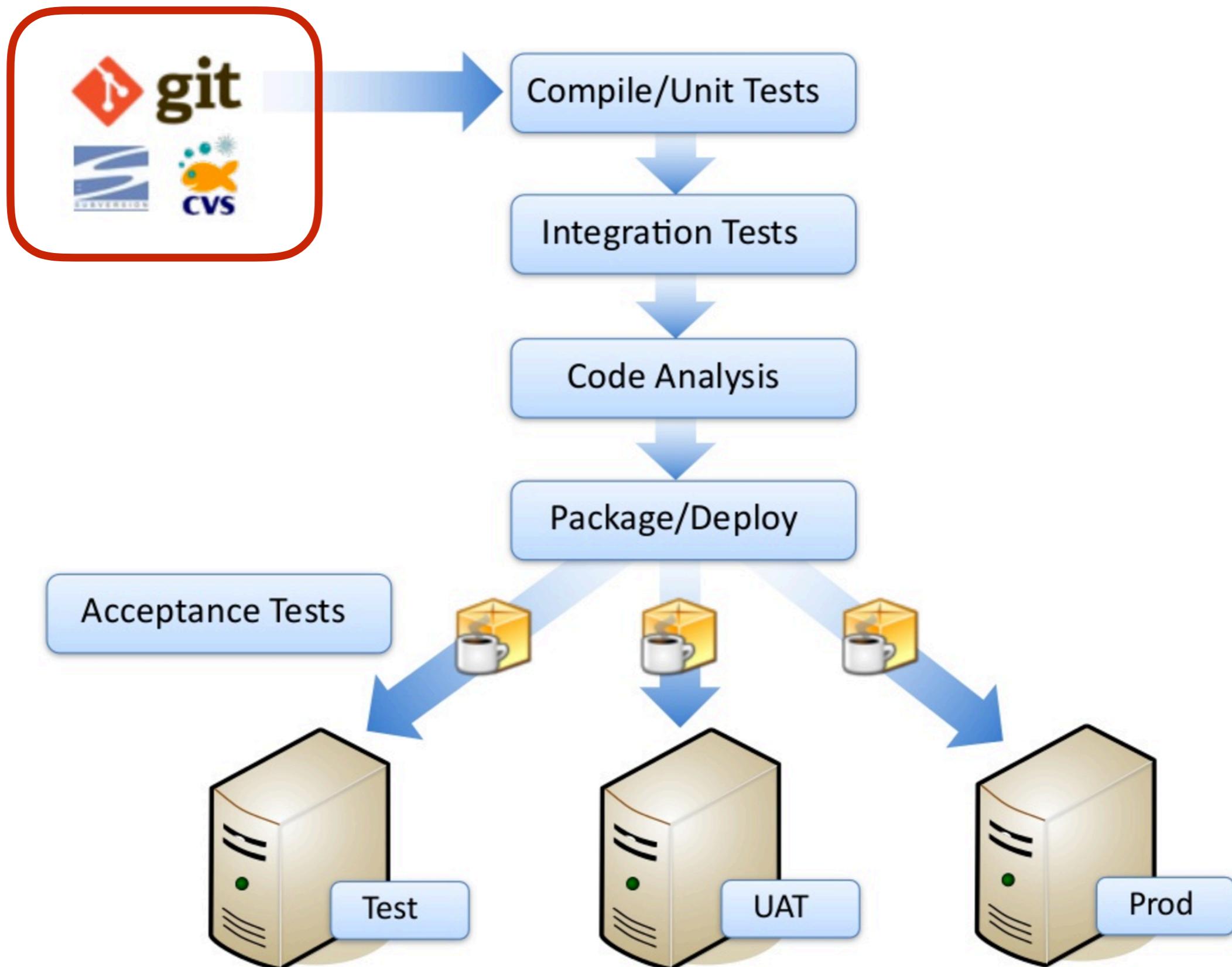
# Jenkins driven by Plugins

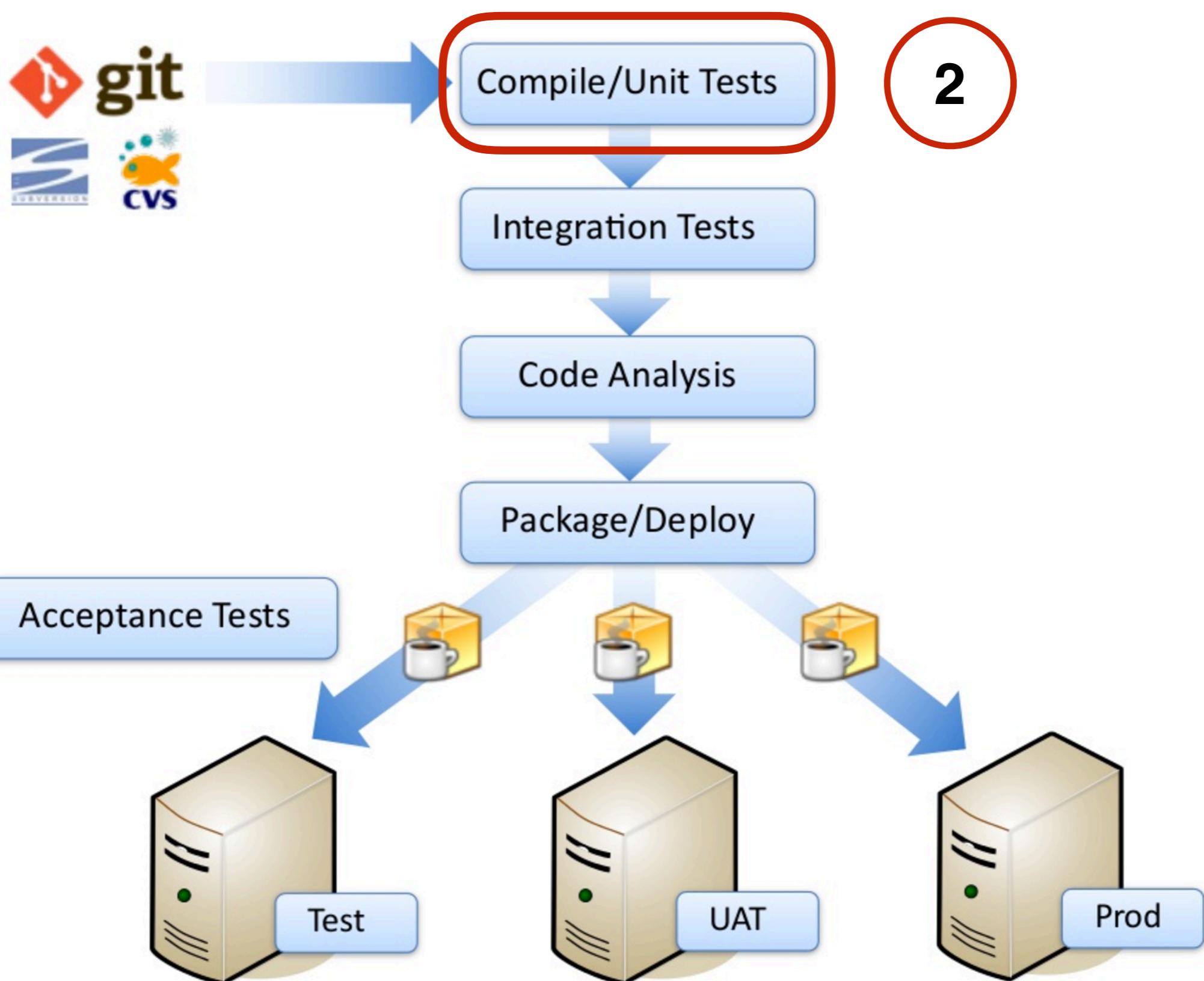


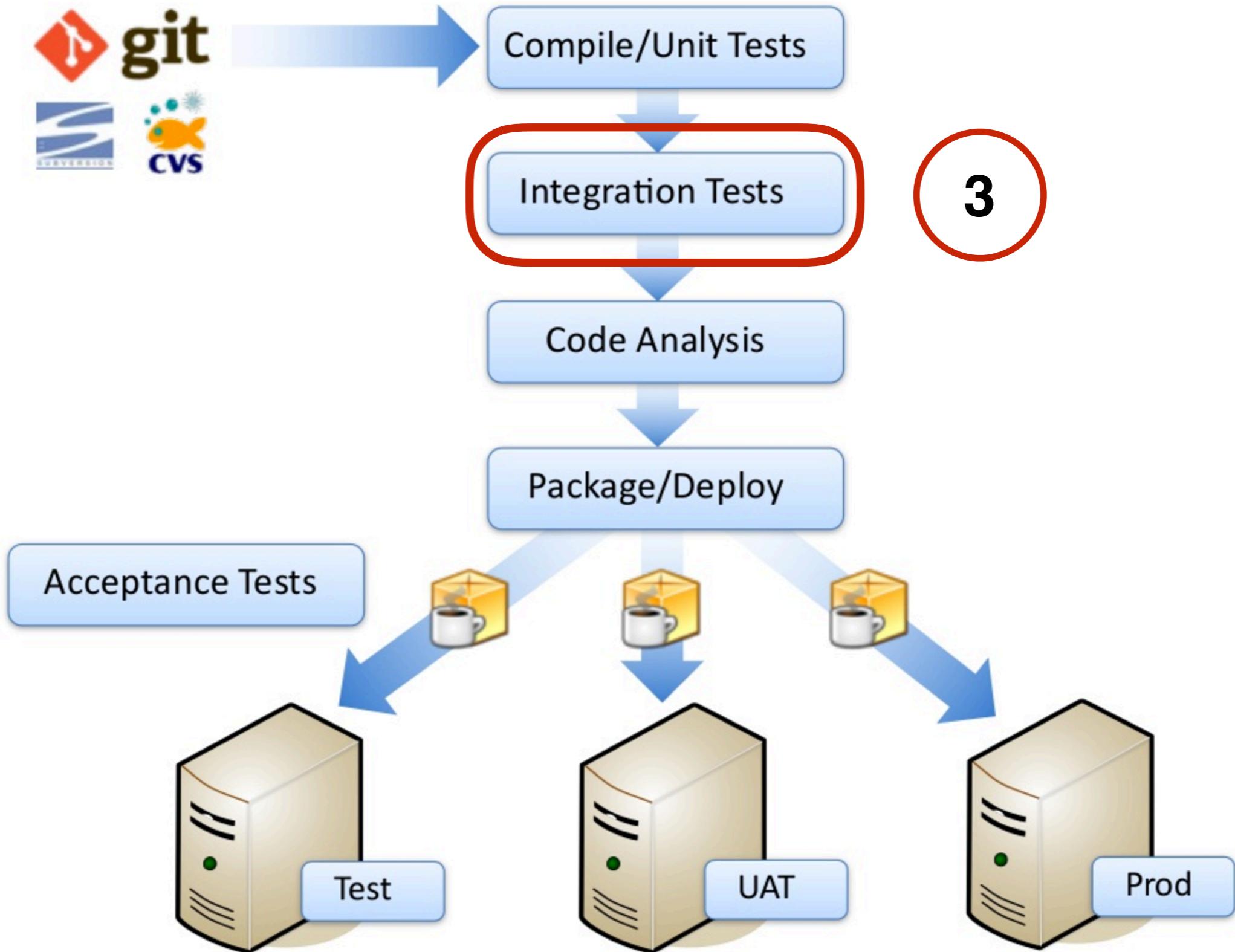
# Let's start with Jenkins

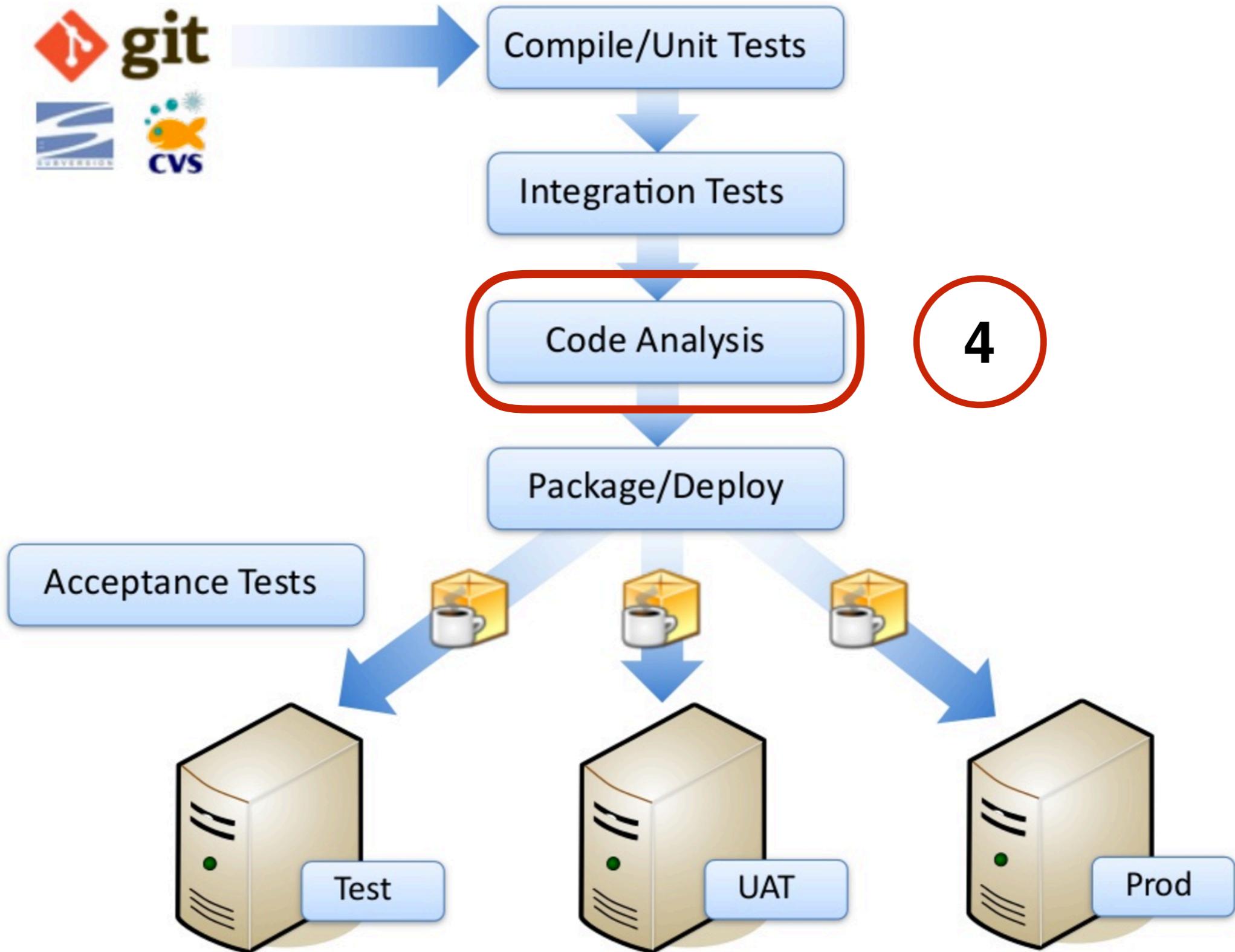


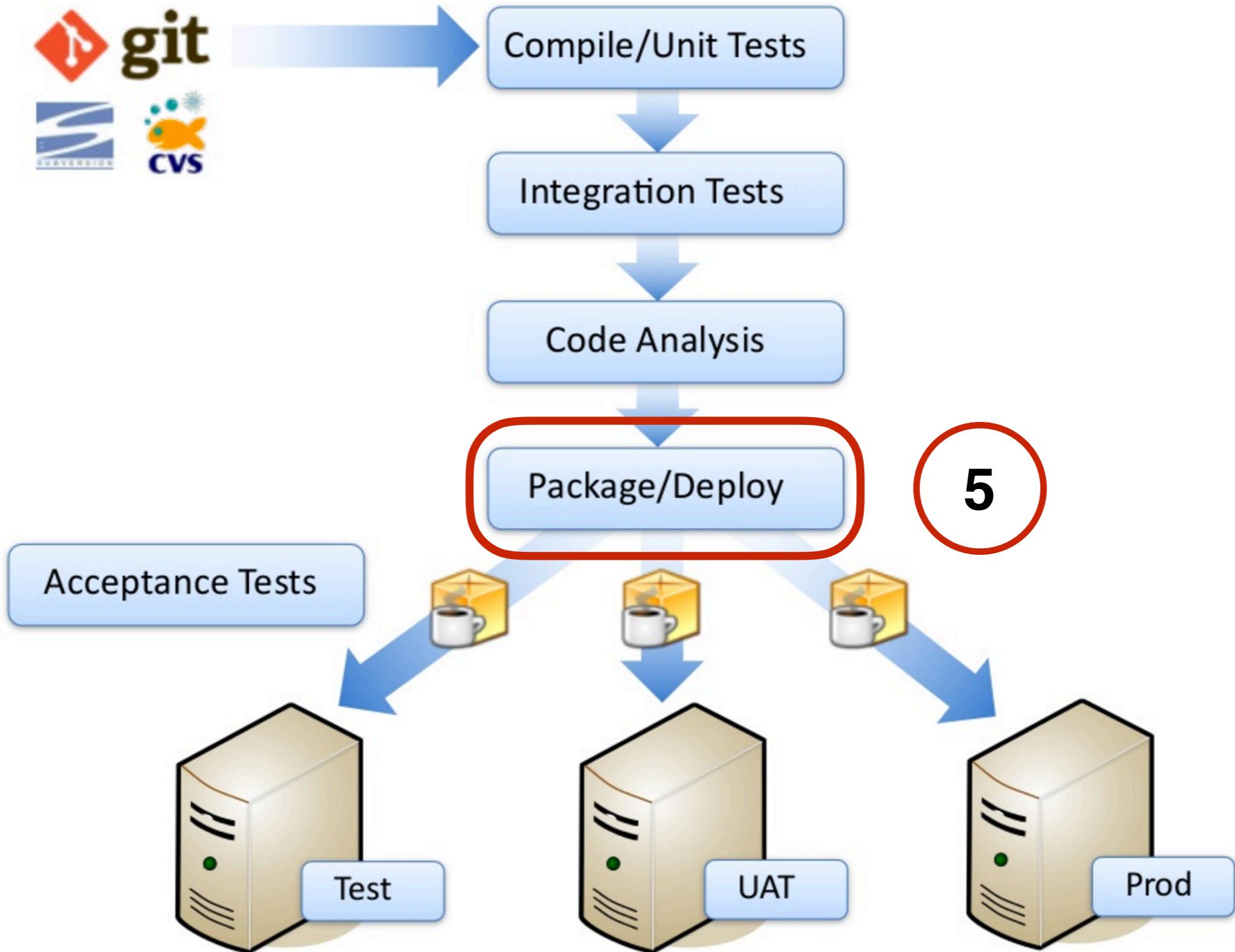
1

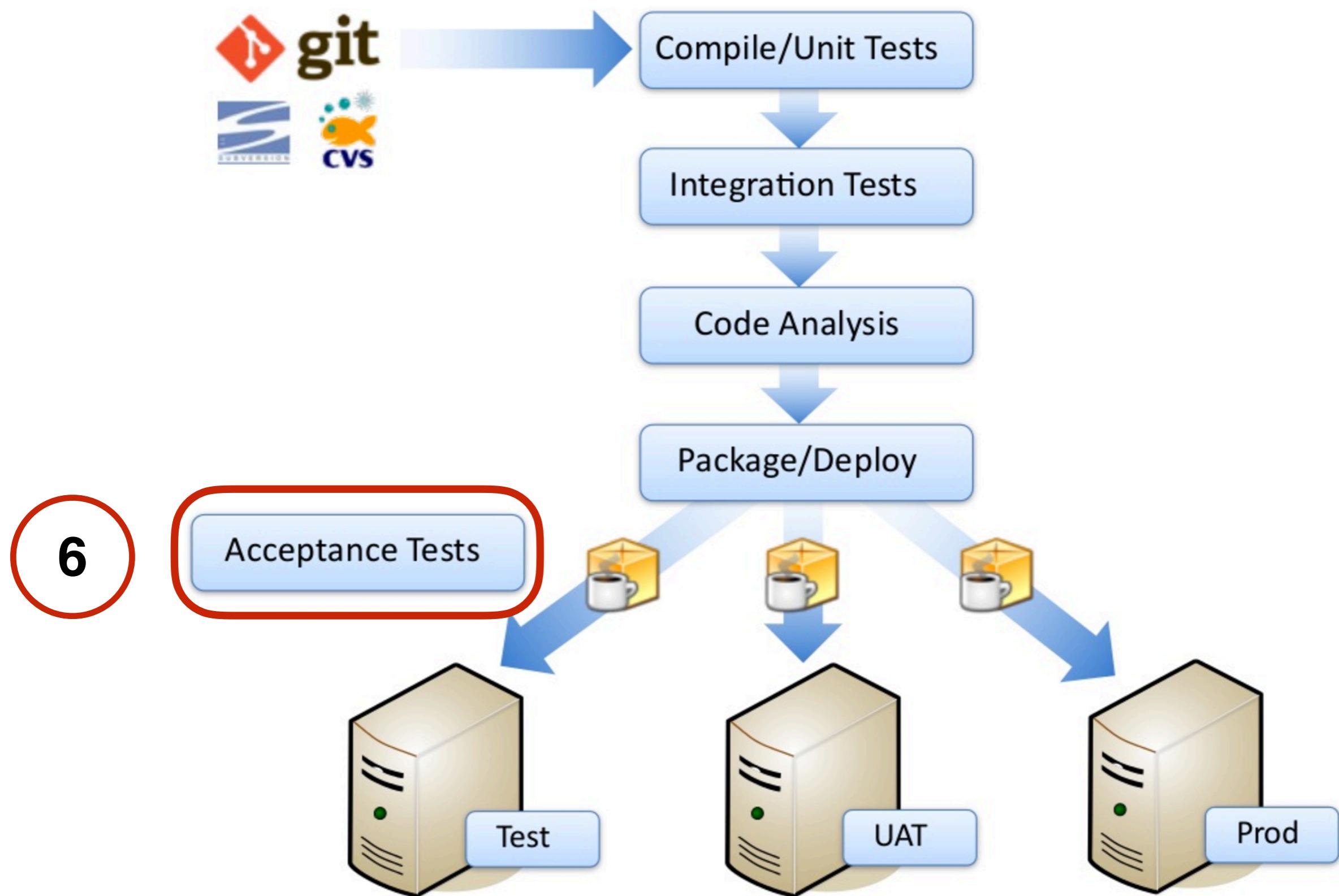


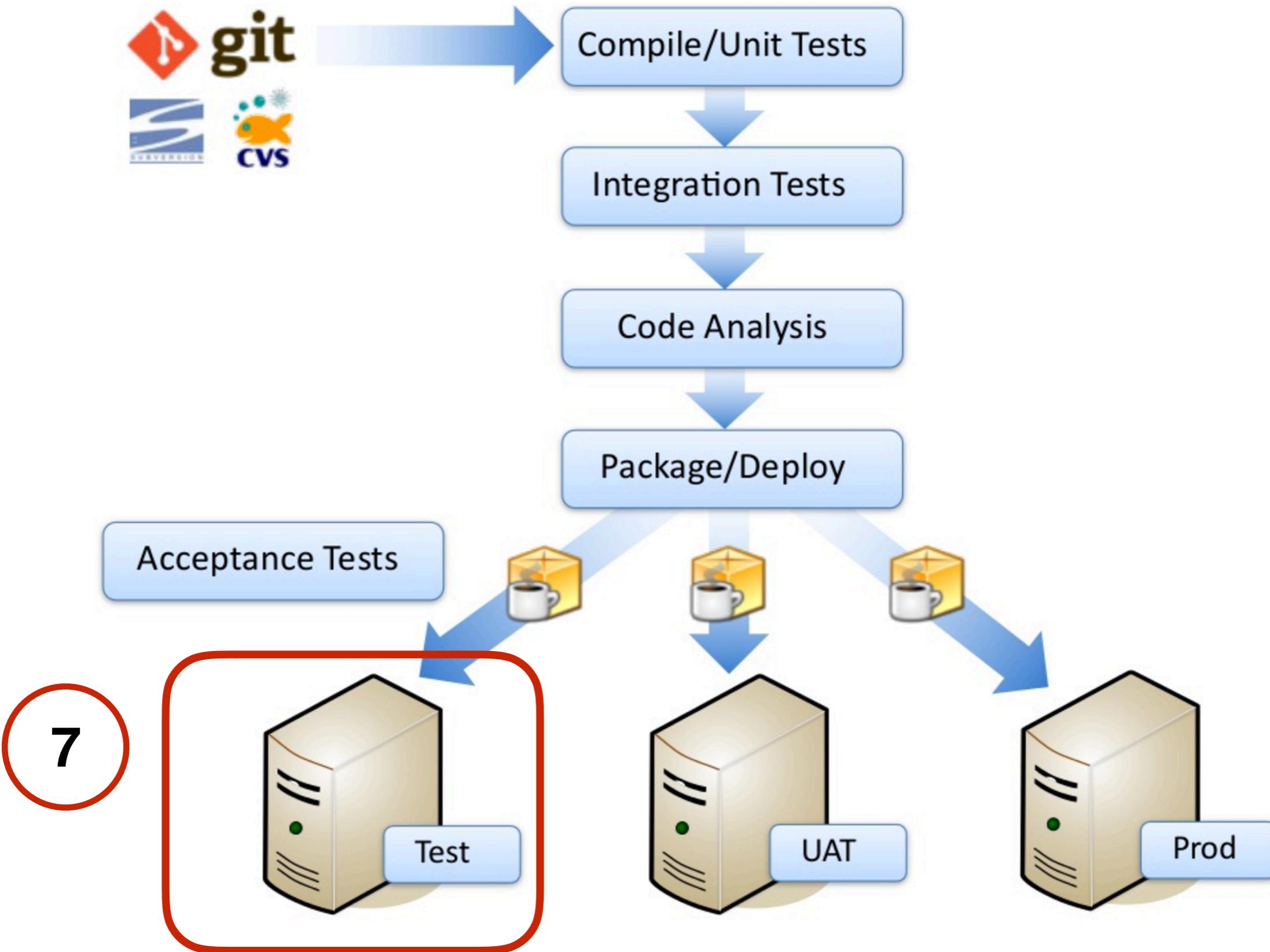


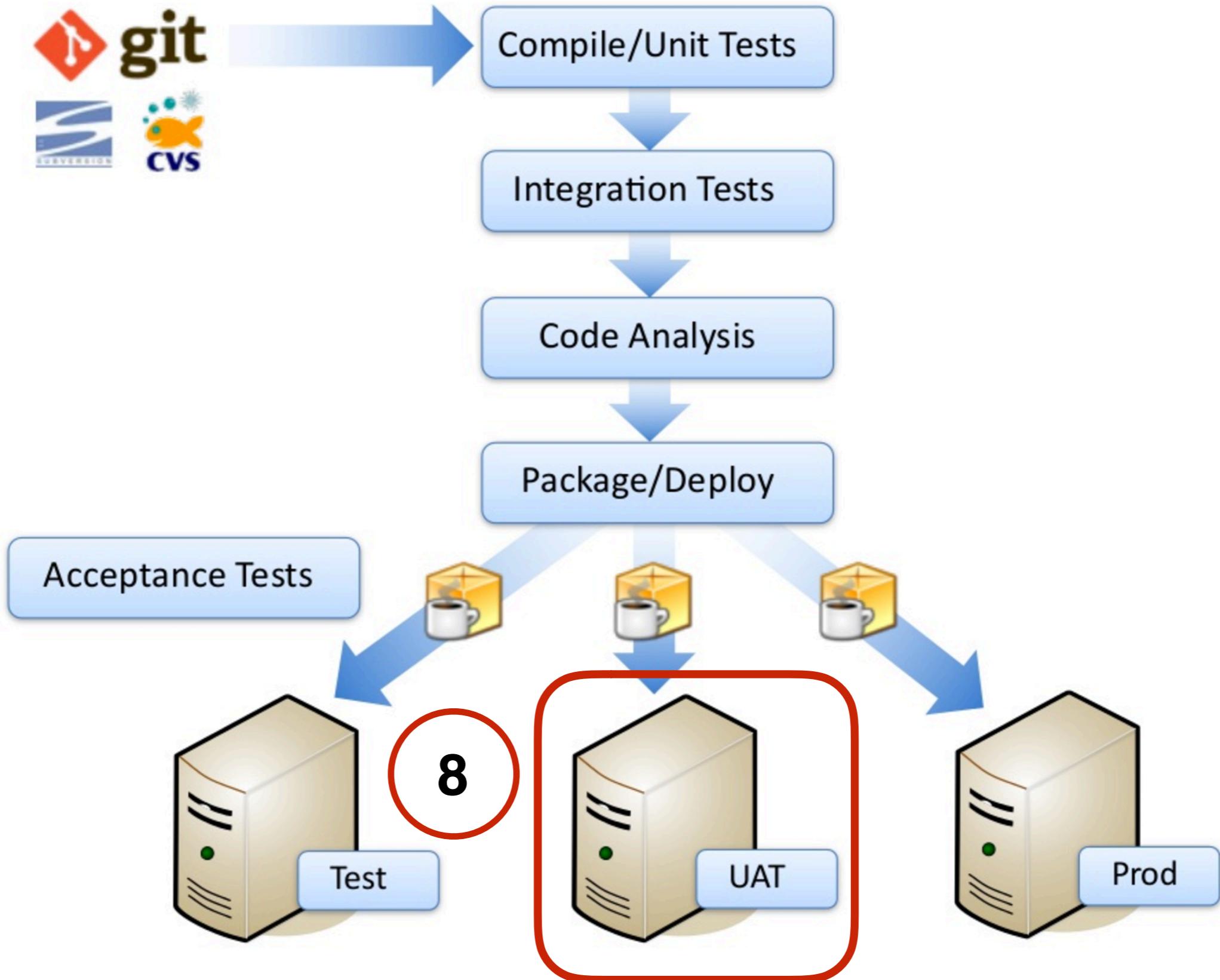


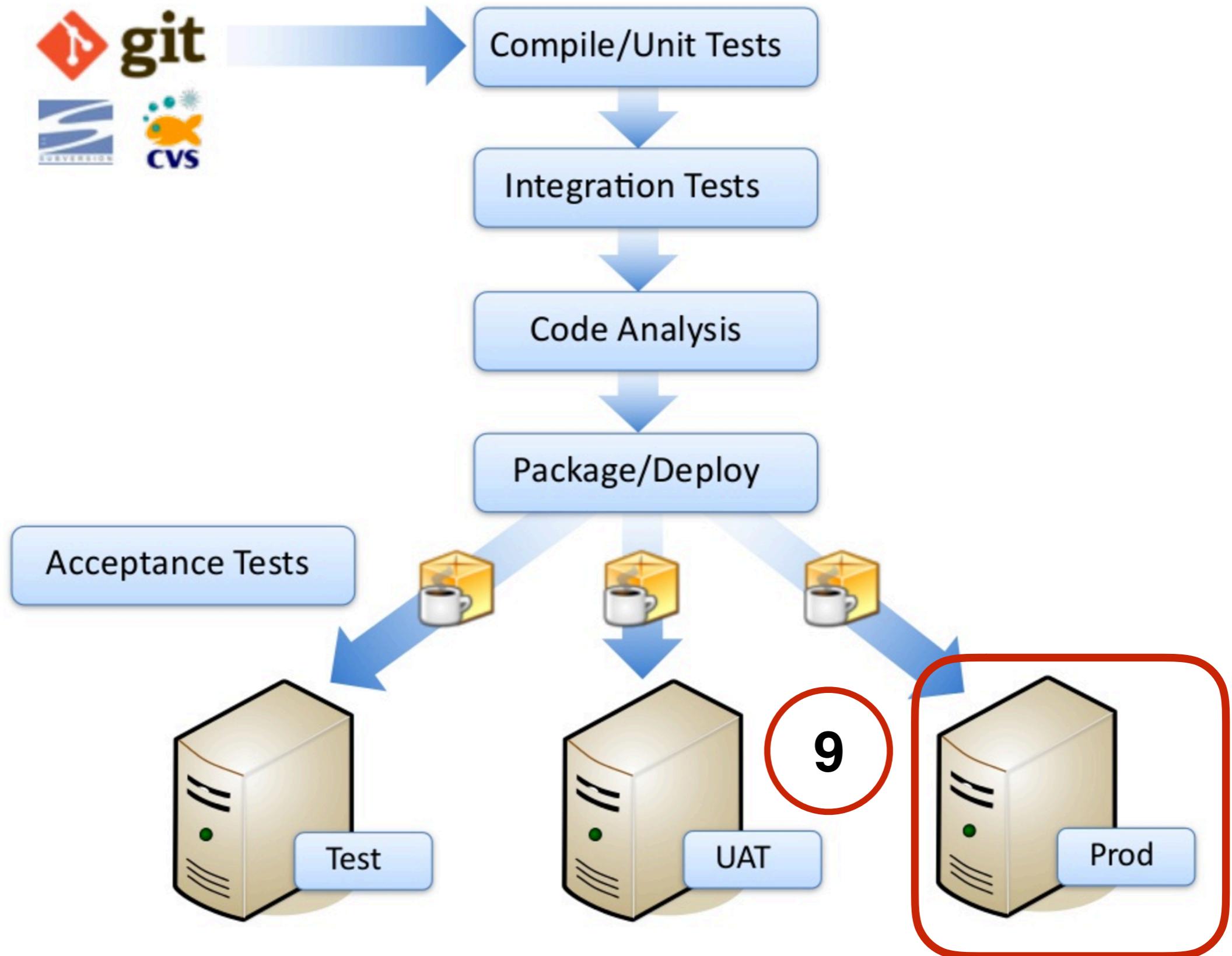












# Run acceptance tests

Try to test Web UI with Robotframework



# Install Robotframework

```
$pip install robotframework
```

```
$pip install robotframework-selenium2library
```



# Run Robotframework

```
$pybot
```

```
[ ERROR ] Expected at least 1 argument, got 0.  
Try --help for usage information.
```



# Robotframework plugin

Filter:

Updates Available Installed Advanced

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

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



# Add build step to run with Robot

## Build

### Execute shell

Command `pybot *.robot`

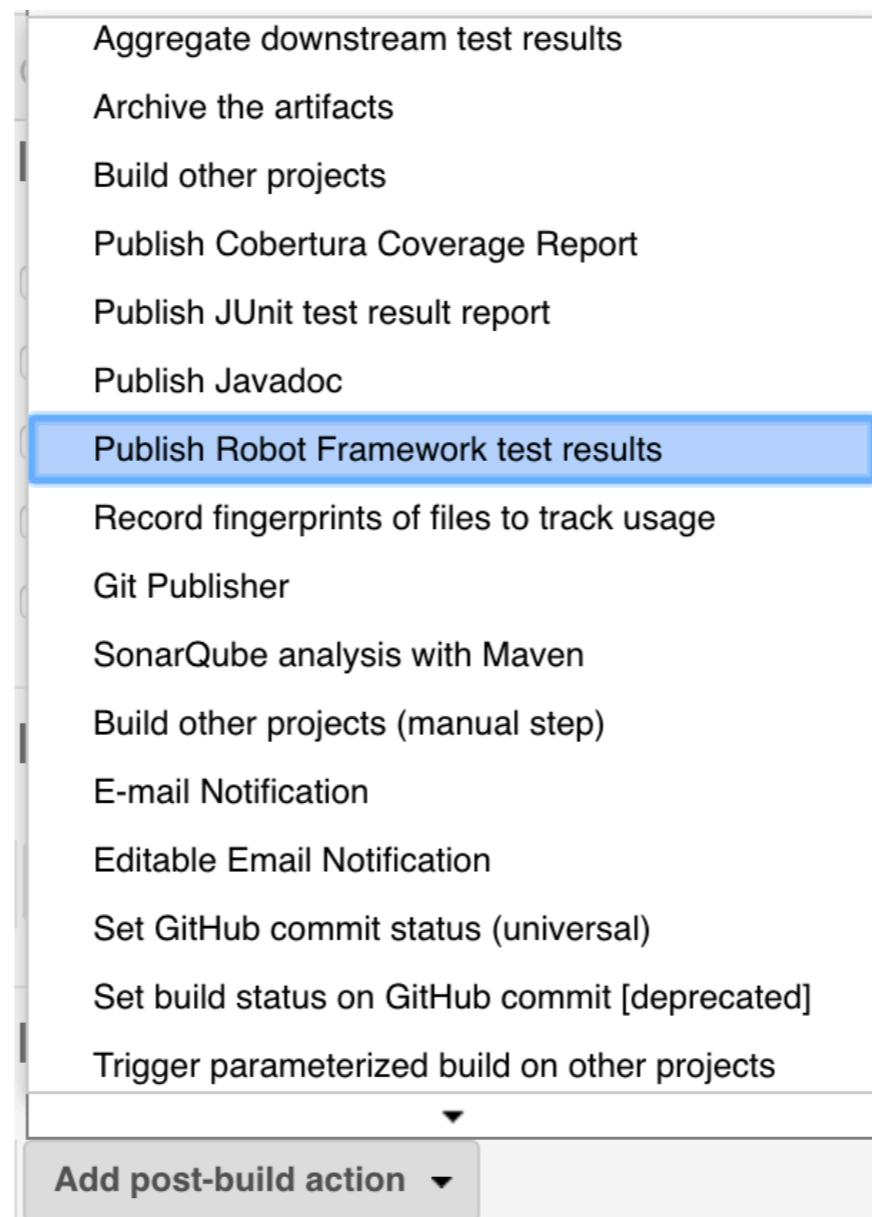
**pybot \*.robot**

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



# Add Robotframework report

Add post build action => Publish Robot Framework



# Add Robotframework report

## Configuration your report

**Post-build Actions**

**Publish Robot Framework test results**

Directory of Robot output

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

Thresholds for build result

Yellow circle %  **Entry must be percentage value between 0-100**

Green circle %  **Entry must be percentage value between 0-100**

Use thresholds for critical tests only

Add post-build action ▾



# Report

 [add description](#)[Disable Project](#)[Workspace](#)[Recent Changes](#)

## Latest Robot Results:

	Total	Failed	Passed	Pass %
Critical tests	1	1	0	0.0
All tests	1	1	0	0.0

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

Robot Framework Tests Trend (all tests)

