

CI and CD



Jenkins



Somkiat Puisungnoen

Somkiat Puisungnoen

Update Info 2 View Activity Log 10+ ...

Timeline About Friends 2,604 Photos More ▾





សាស្ត្រពិភាក្សានៃប្រជាជាតិ



← → 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





บริษัท สยามชำนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

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



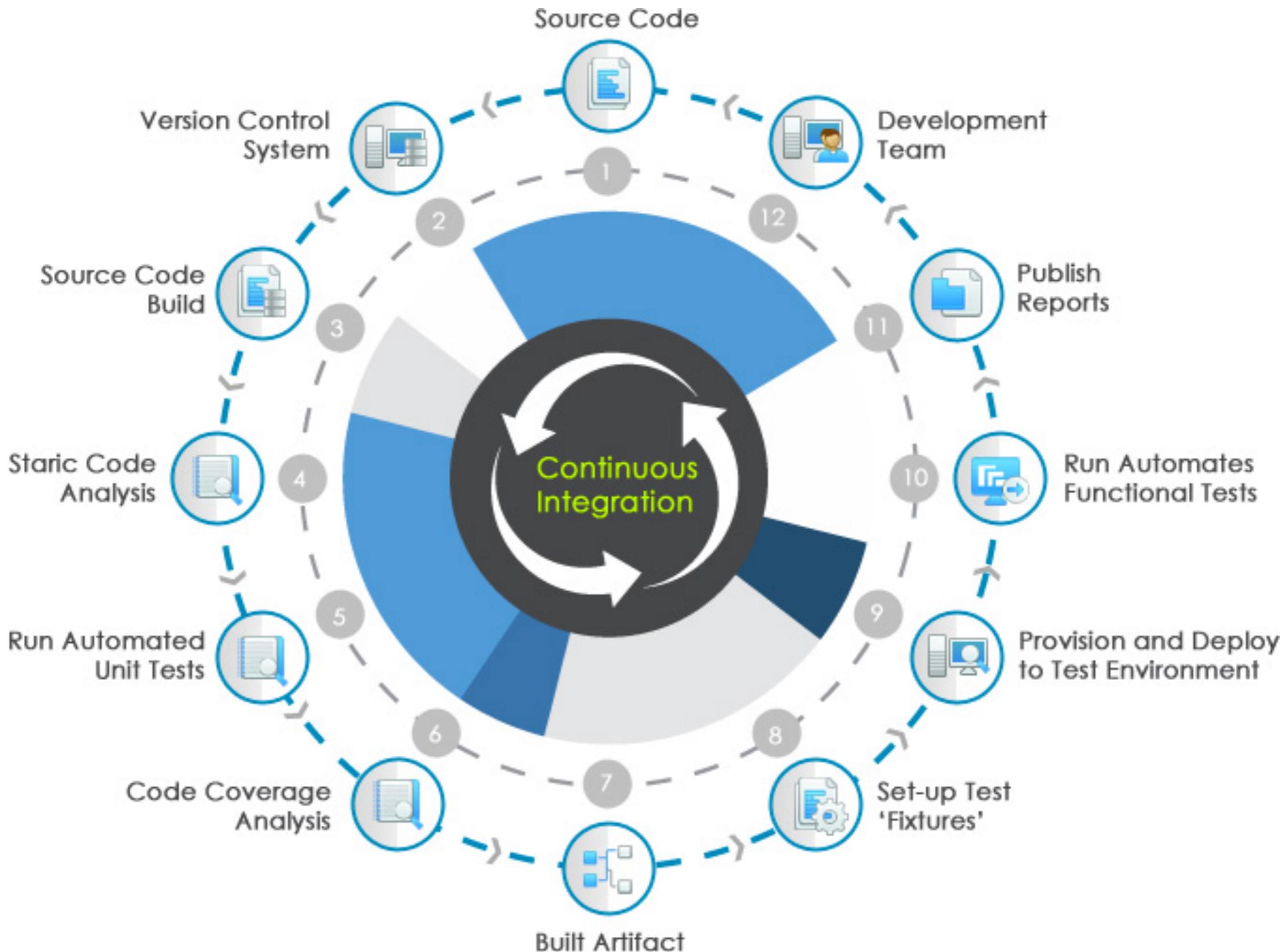
Agenda

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

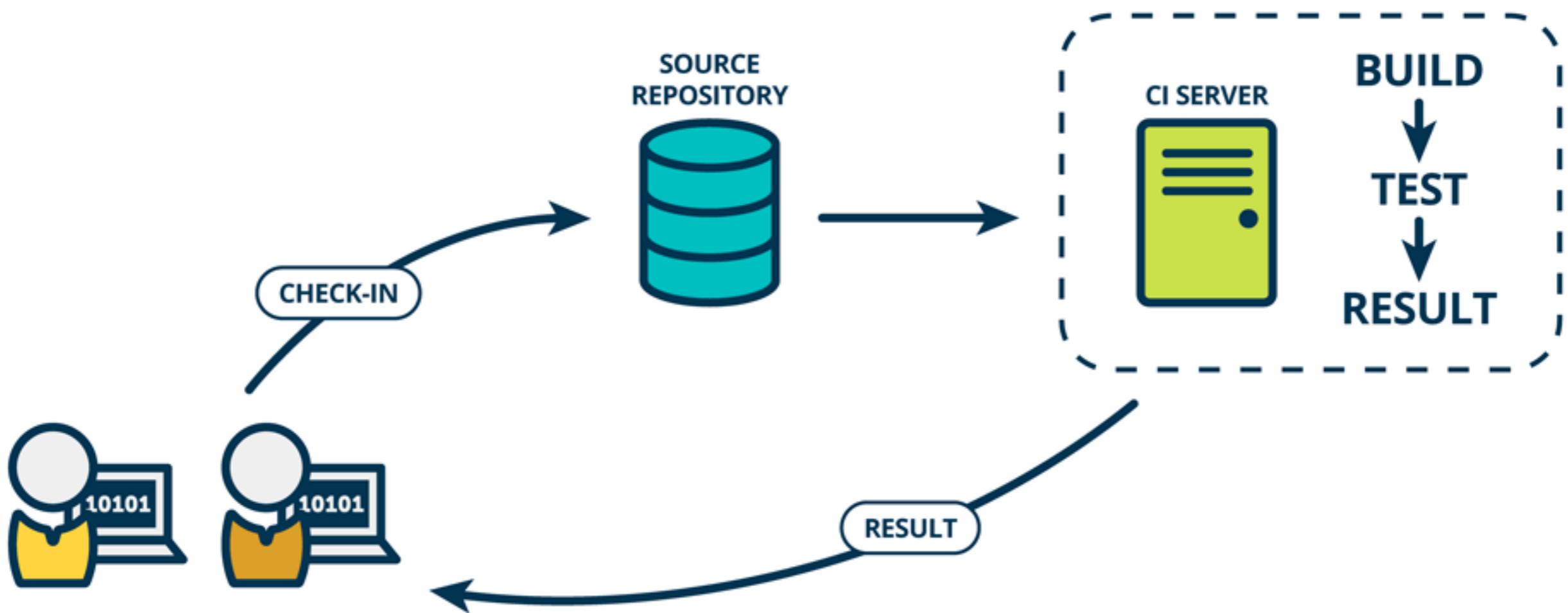


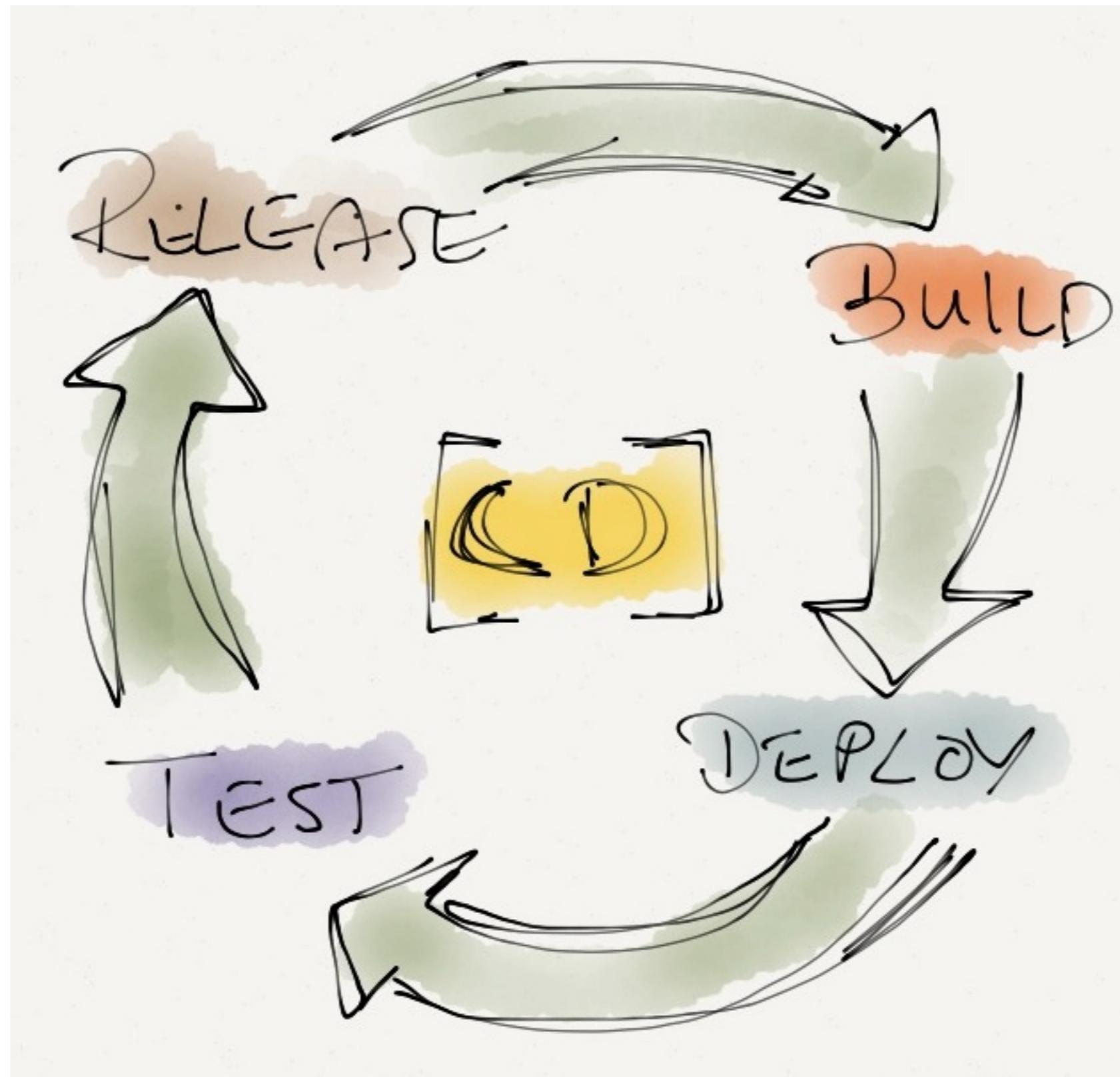
CI is a practice

Strive for **fast feedback**



Continuous Integration





CONTINUOUS DELIVERY



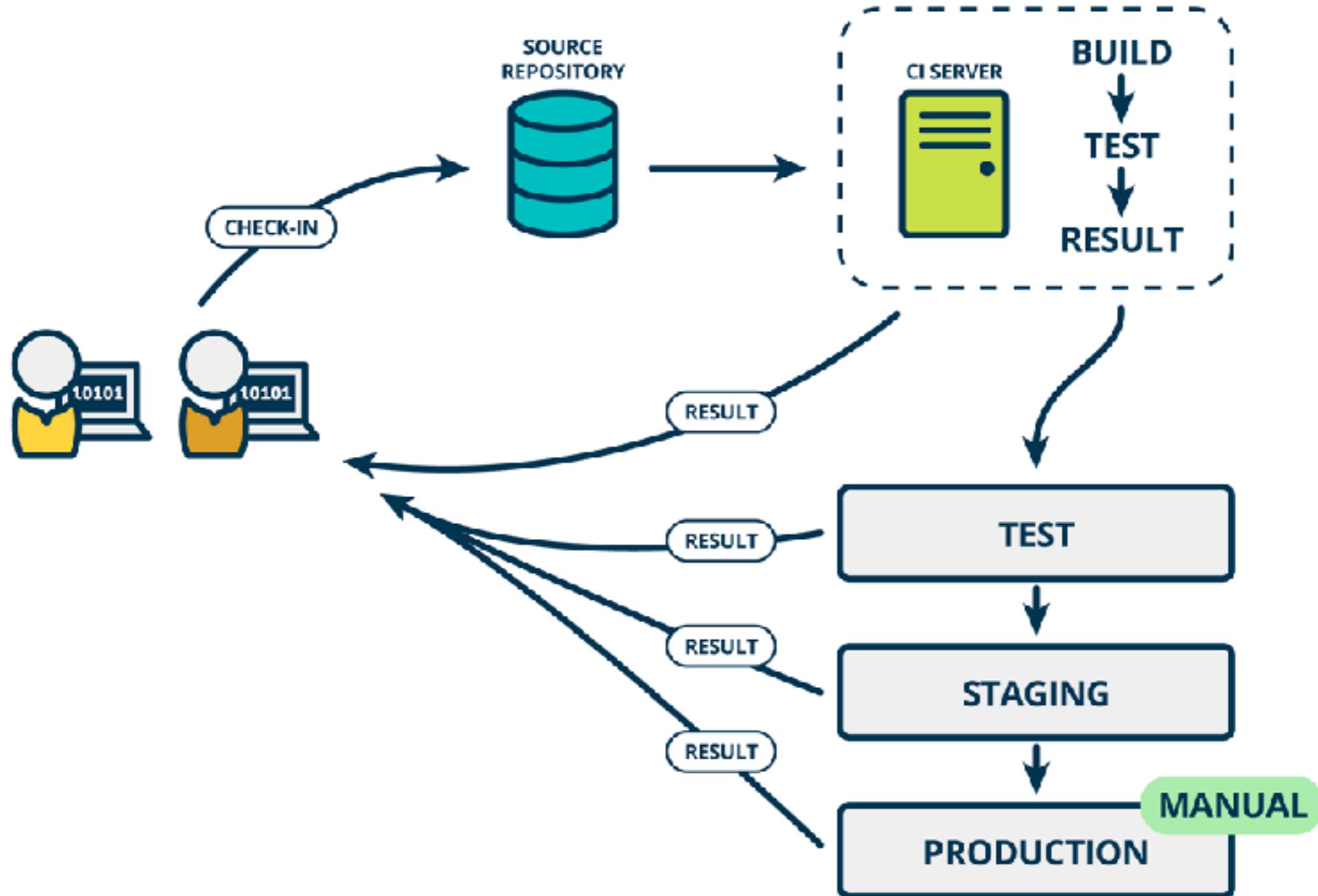
CONTINUOUS DEPLOYMENT



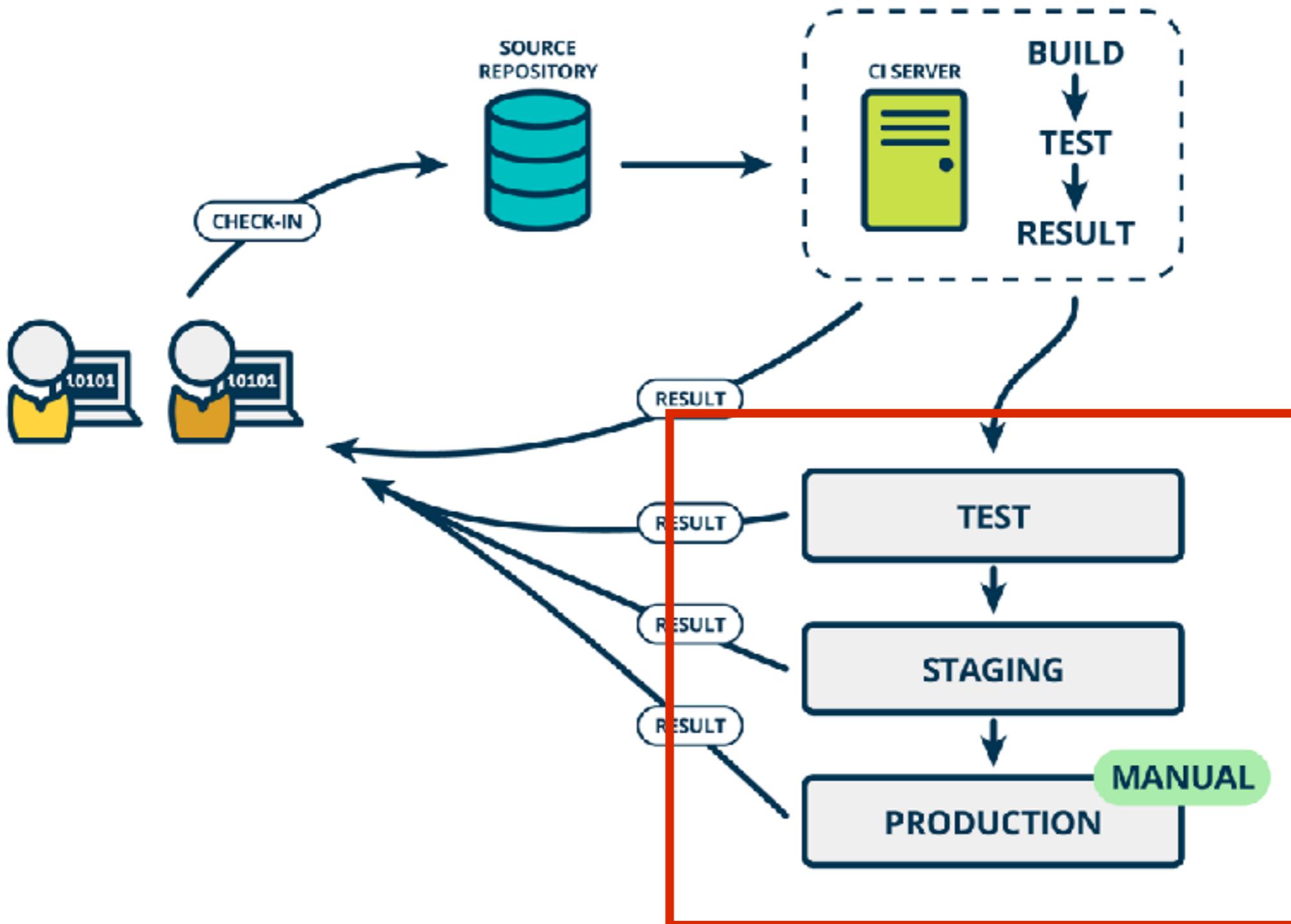
<http://blog.crisp.se/2013/02/05/yassalsundman/continuous-delivery-vs-continuous-deployment>



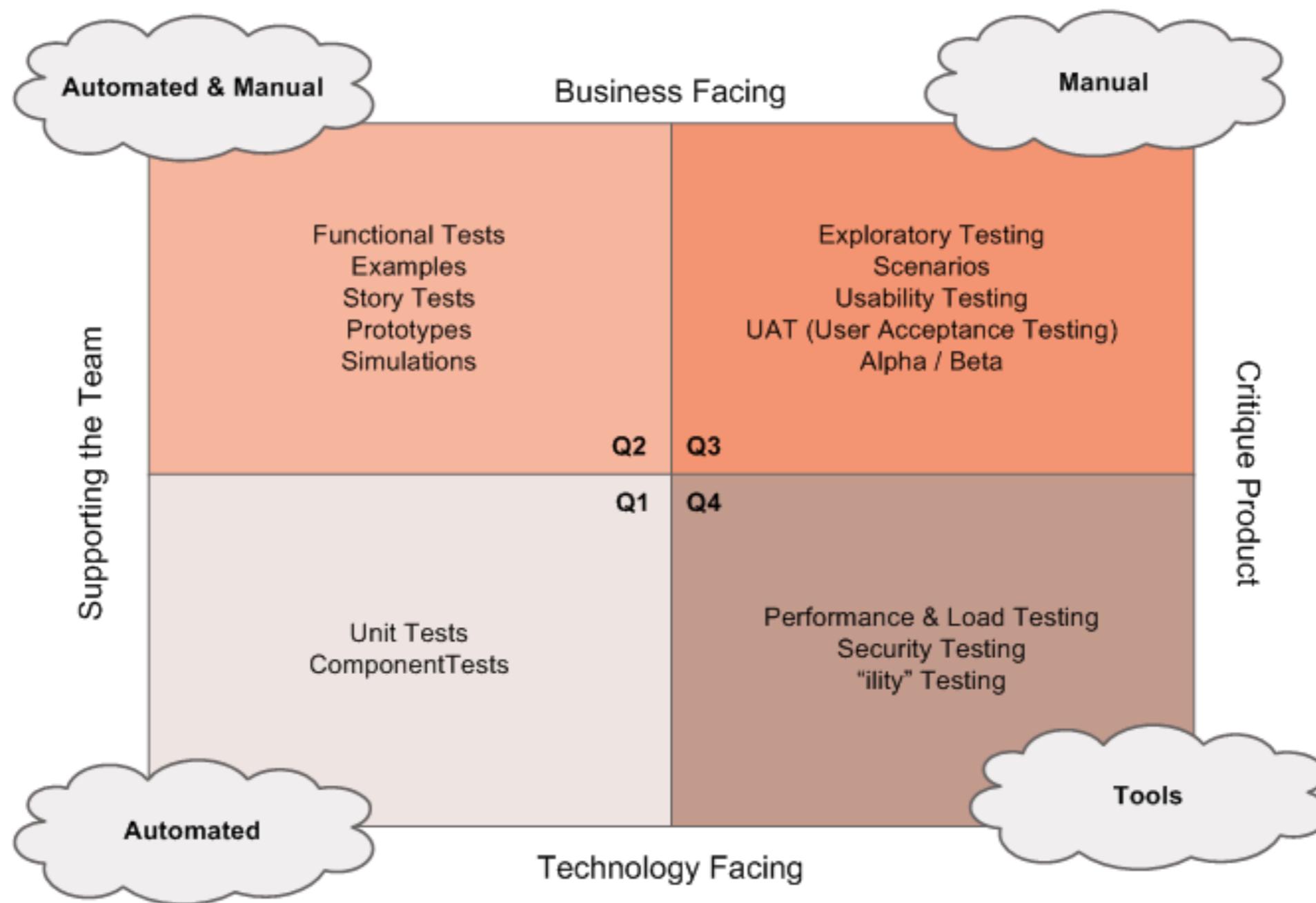
Continuous Delivery



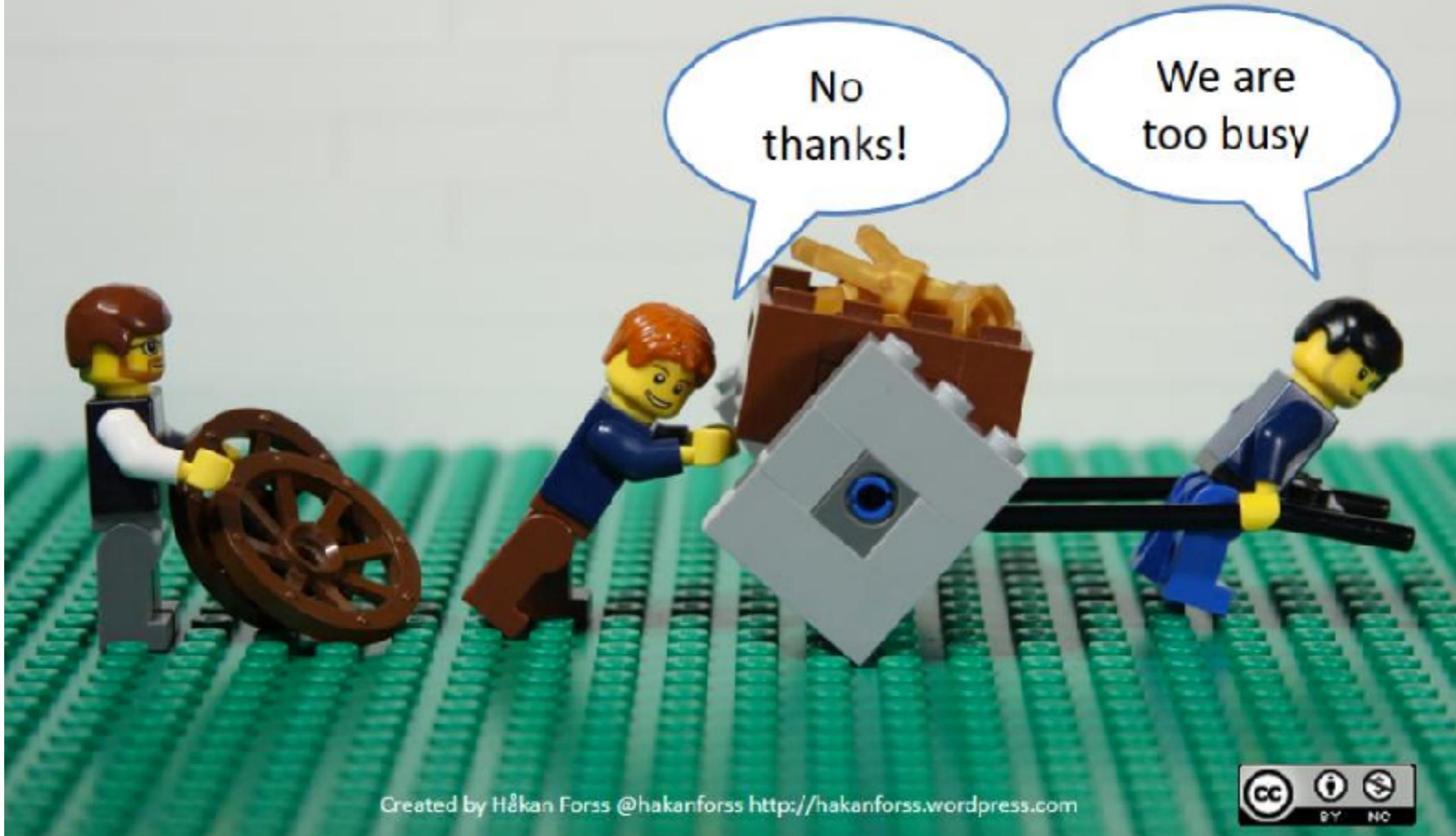
Rising of DevOps



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



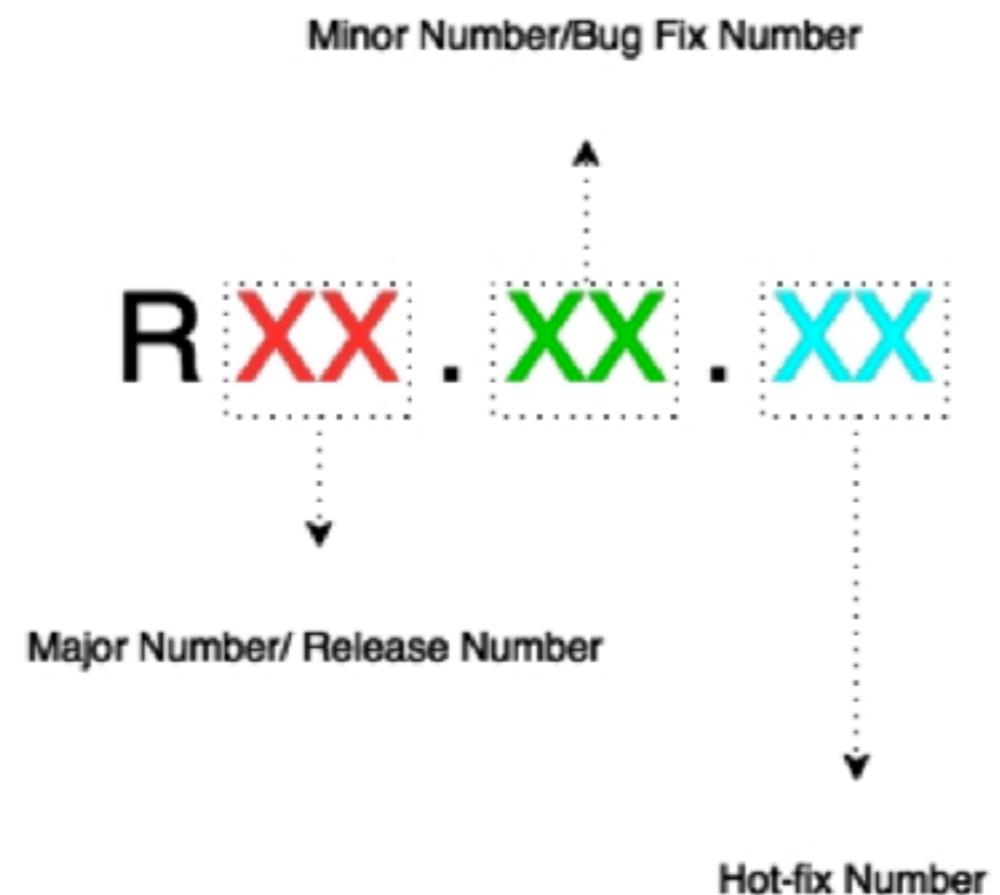
บริษัท สยามชนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

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



How to achieve CI ?

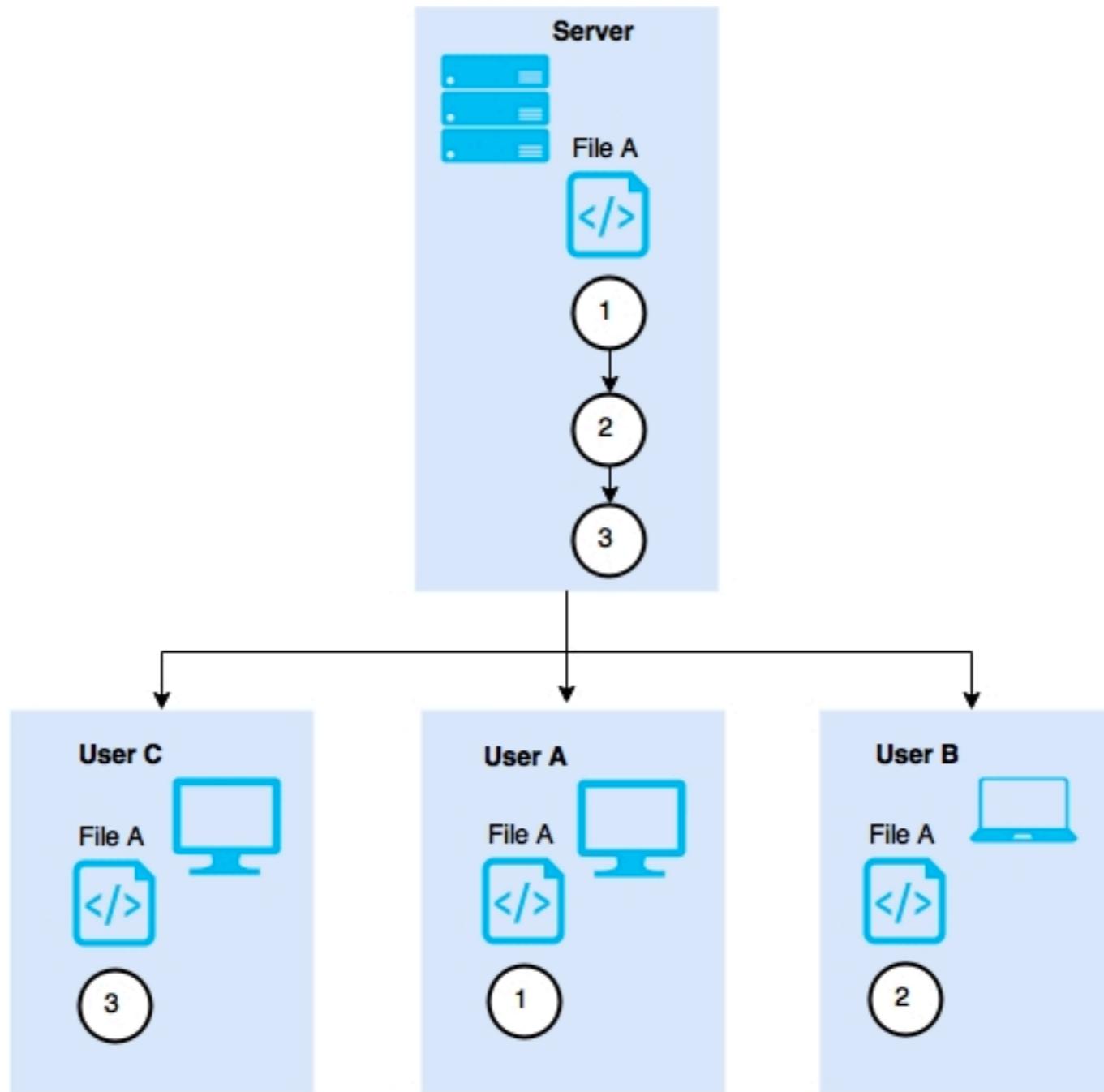


1. Use version control system

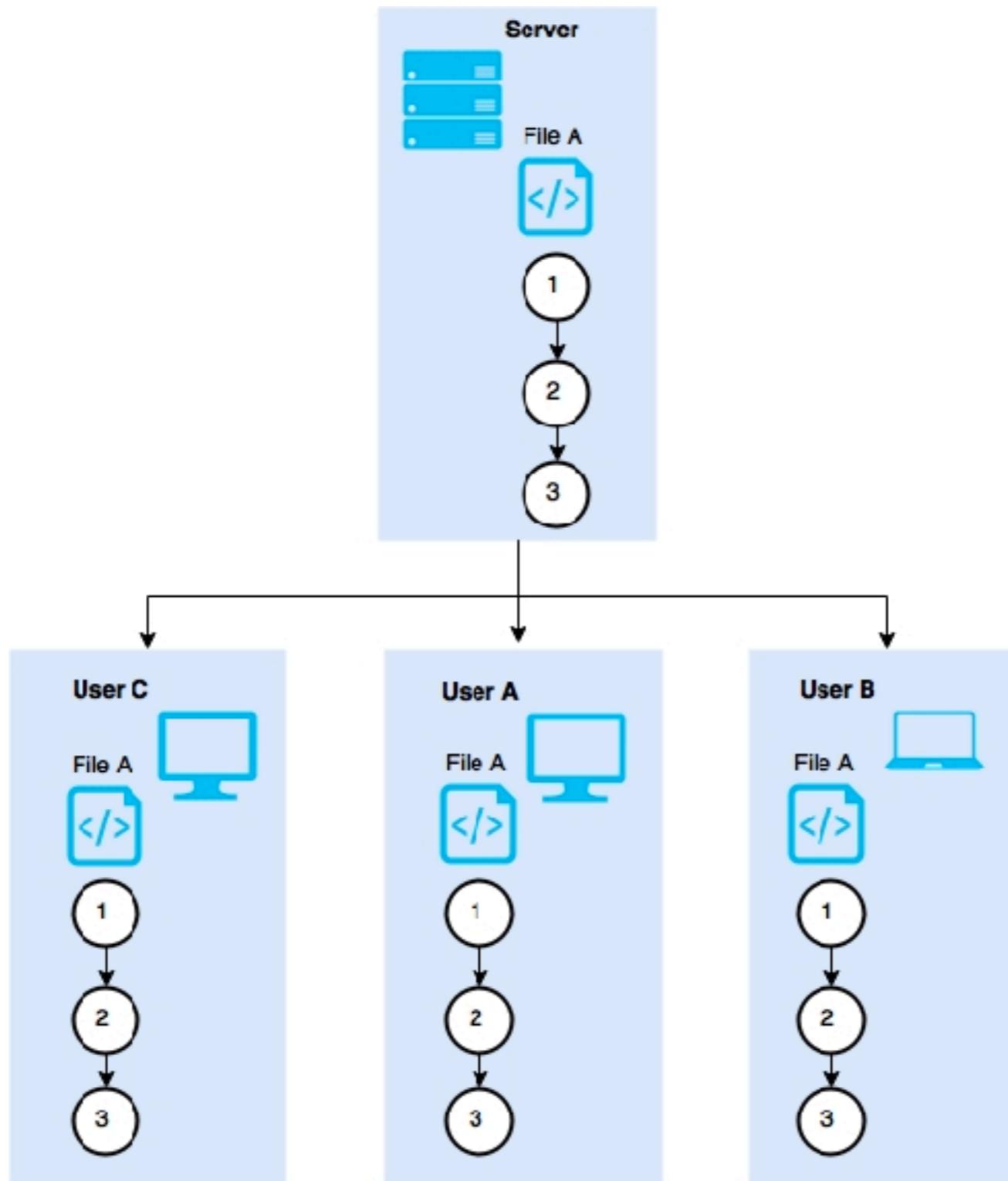
Local
Centralize
Distributed



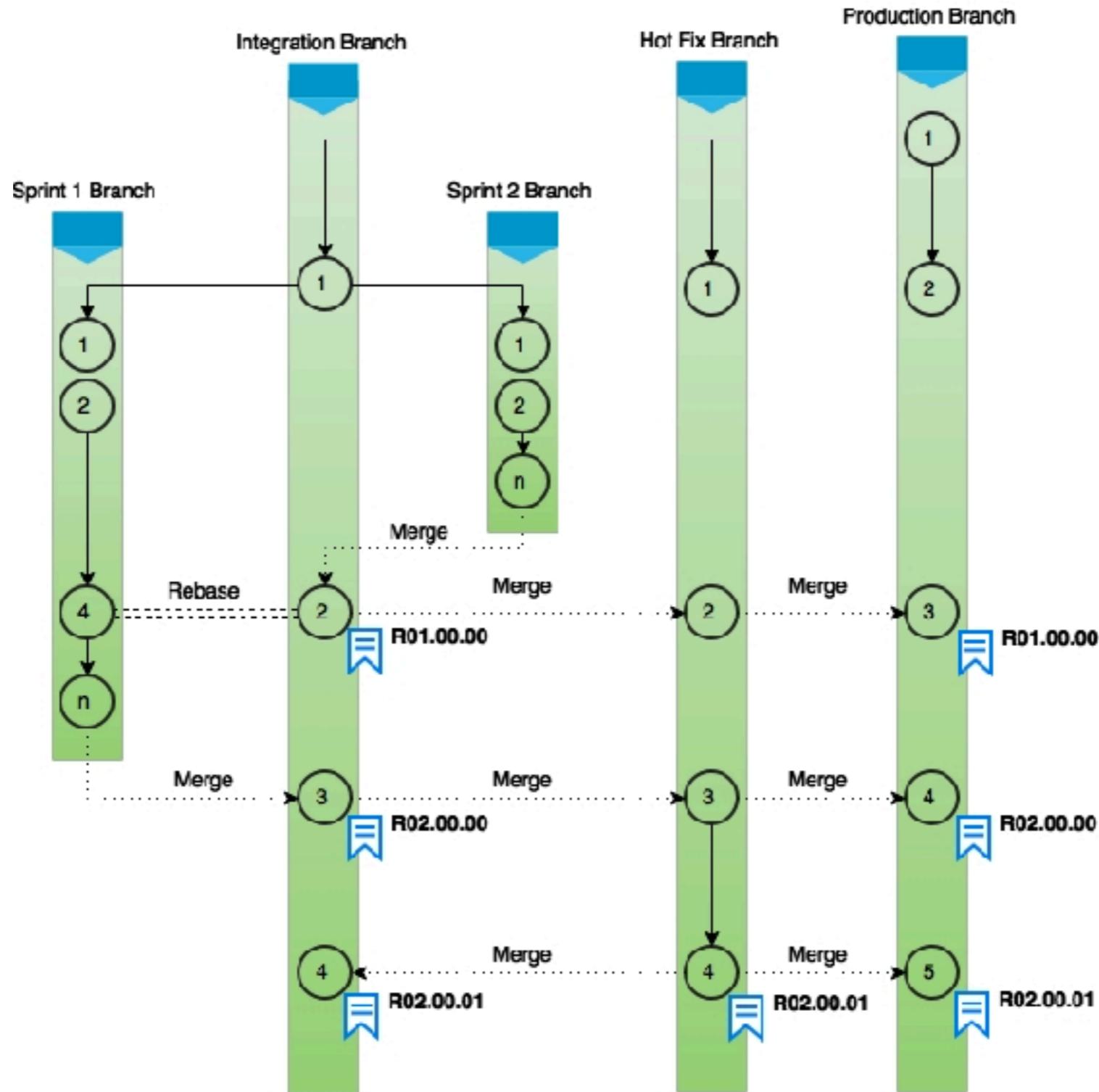
Centralize version control



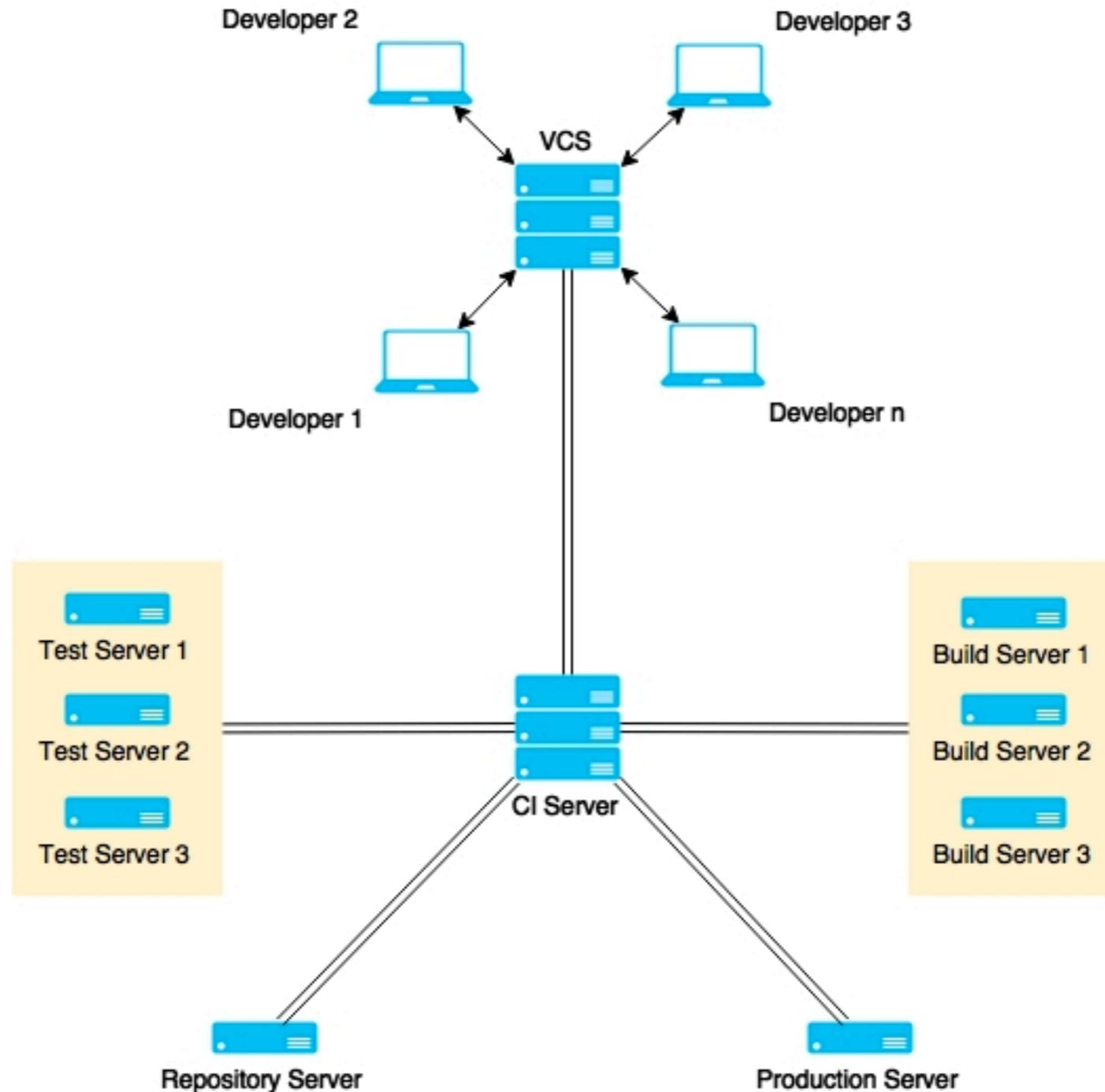
Distributed version control



Branch strategies

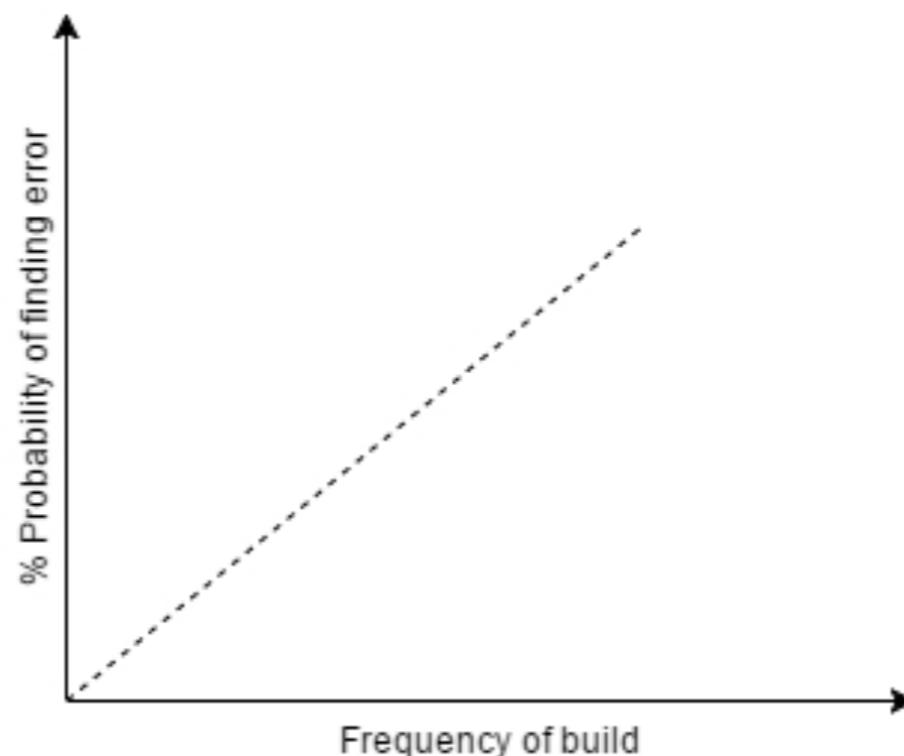


2. Use CI tool

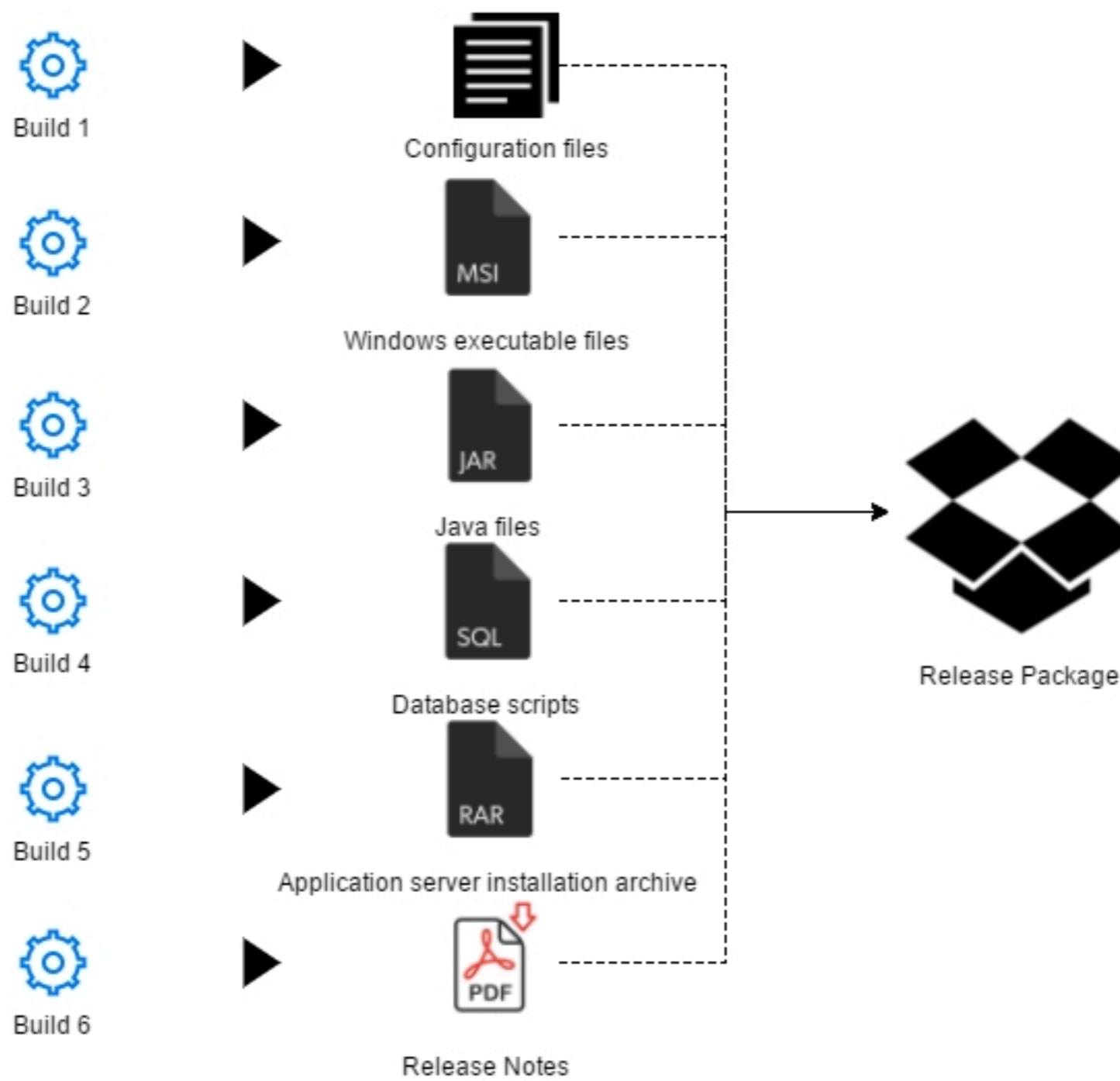


3. Create self-trigger build

Speed
Catching issues as early as possible



4. Automate the packing



5. Using build tools

- Javascript
 - Gulp, Grunt, Brocolli



- C#/.NET
 - Nant, MSBuild



- Java/JVM
 - Ant, Maven, Gradle, SBT, Leiningen



Use static code analysis



**Use static code analysis
Automate testing**



Use static code analysis
Automate testing
Automate deployment



Use static code analysis
Automate testing
Automate deployment
Backward traceability



Use static code analysis
Automate testing
Automate deployment
Backward traceability
Use defect tracking tool



"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.



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

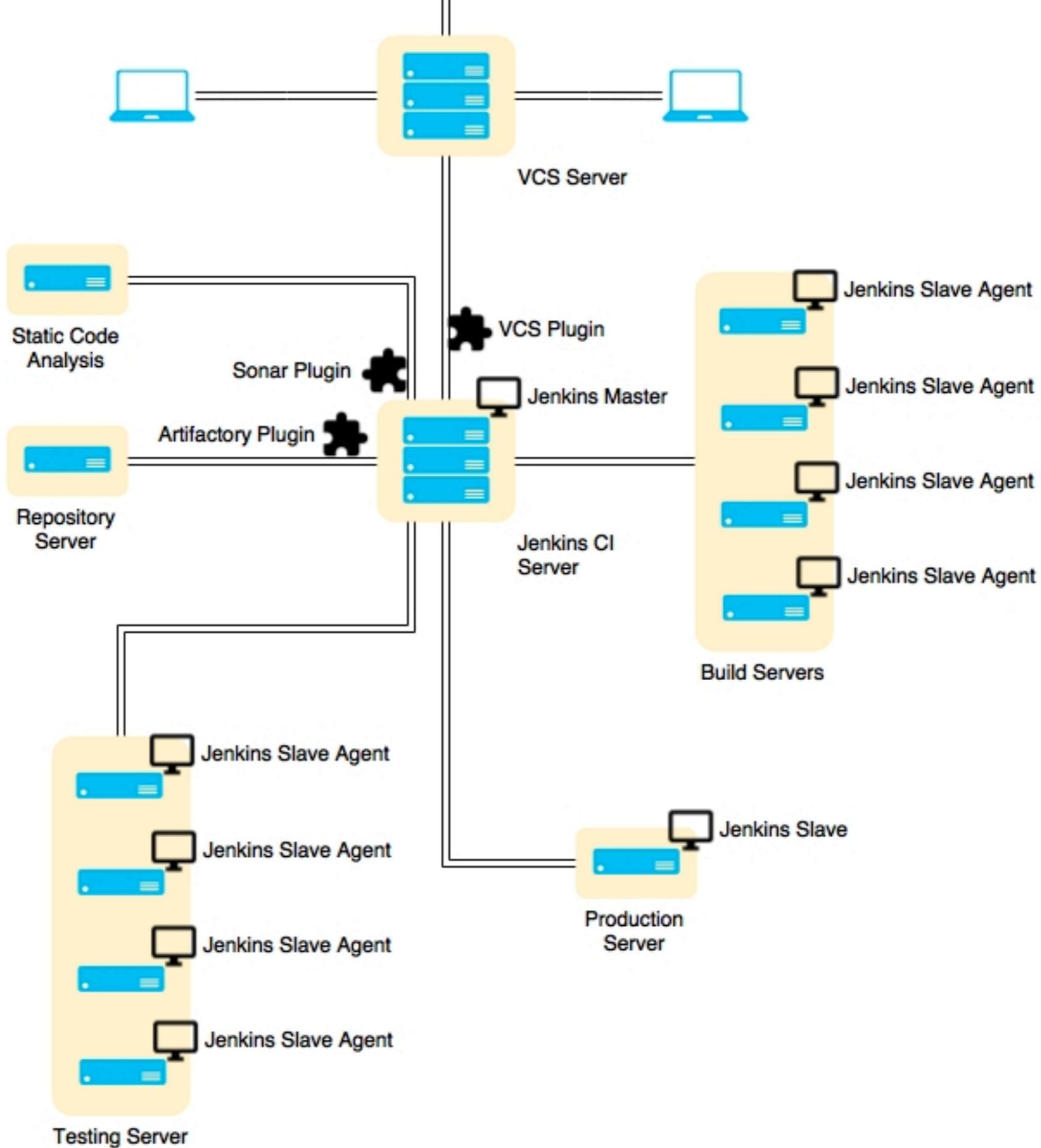
<https://wiki.jenkins-ci.org/pages/viewpage.action?pageId=58001258>



บริษัท สยามชนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

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 small body. It has a smiling face with simple features. It is wearing a blue suit jacket over a white shirt with a red bow tie. 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>



บริษัท สยามชนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

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 is a 'Getting Started' link. The main title 'Unlock Jenkins' is displayed prominently. Below it, a text block explains that a password has been written to the log and a specific file on the server. A red box highlights the 'Administrator password' input field, which is currently empty. A 'Continue' button is located at the bottom right of the form.

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

The screenshot shows the Jenkins dashboard. At the top, there is a navigation bar with the Jenkins logo, a search bar, and user information for 'somkiat'. Below the navigation bar, the main content area features a large 'Welcome to Jenkins!' message and a button to 'create new jobs'. On the left side, there is a sidebar with links for 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', and 'Credentials'. Below the sidebar, there are two collapsed sections: 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing '1 Idle' and '2 Idle'). At the bottom of the page, there is a footer with the text '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. On the left, there is a sidebar with the following items:

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

Below the sidebar is a "Build Queue" section with the message "No builds in the queue." To the right, a large "Welcome to Jenkins!" message is displayed, followed by a call-to-action: "Please create new jobs to get started." This message is also highlighted with a red circle.



2. Fill in name

Enter an item name

hello

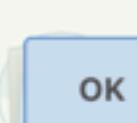
» Required field

 **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.



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

Save Apply

None



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...](#)

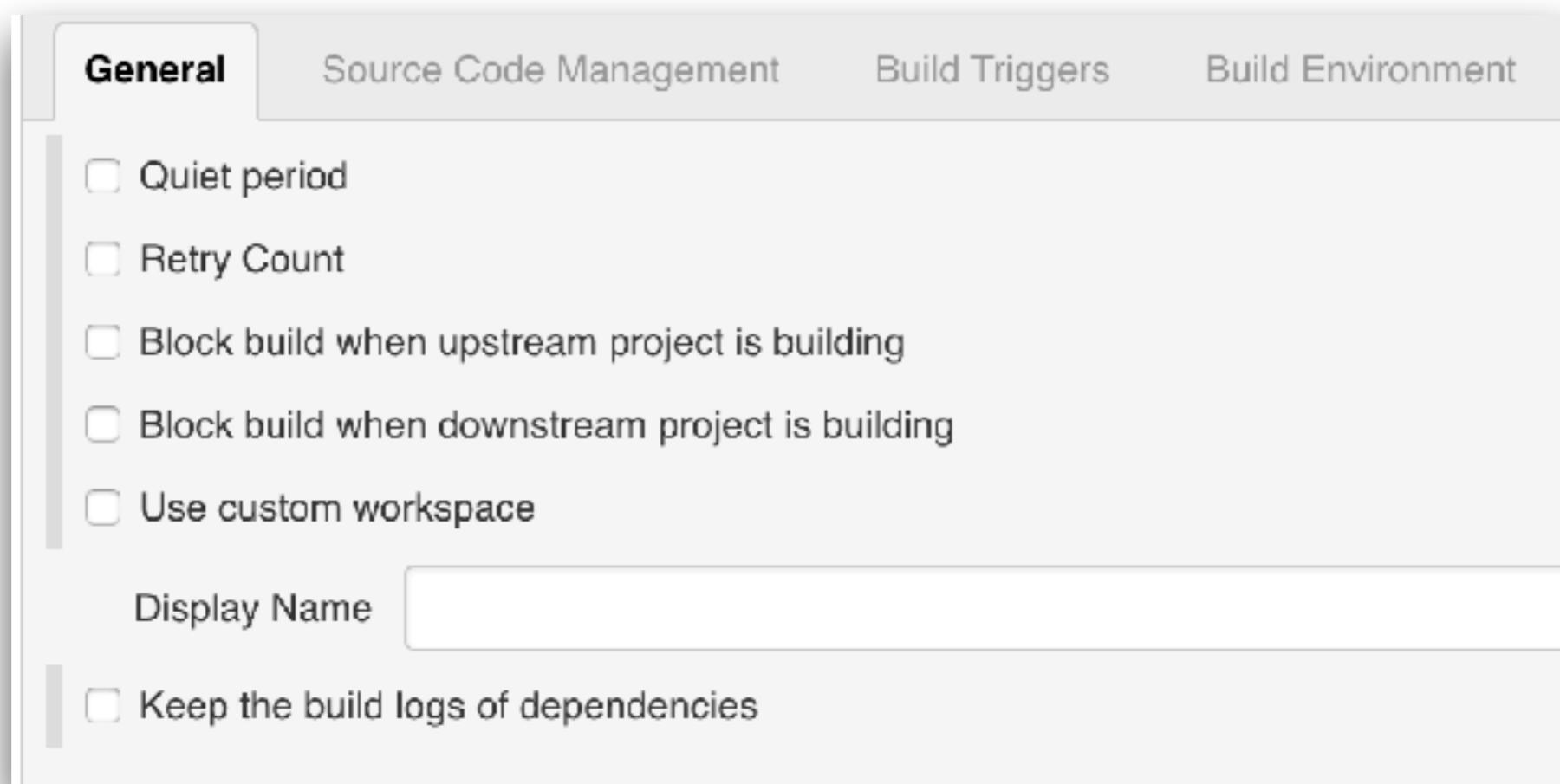
Source Code Management

Save Apply

None



Advance project options



7. Source code management

The screenshot shows a software interface with a navigation bar at the top containing tabs: General, Source Code Management, Build Triggers, Build Environment, Build, and Post-build Actions. The 'Source Code Management' tab is currently selected, indicated by a bold black font and a slightly darker background.

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

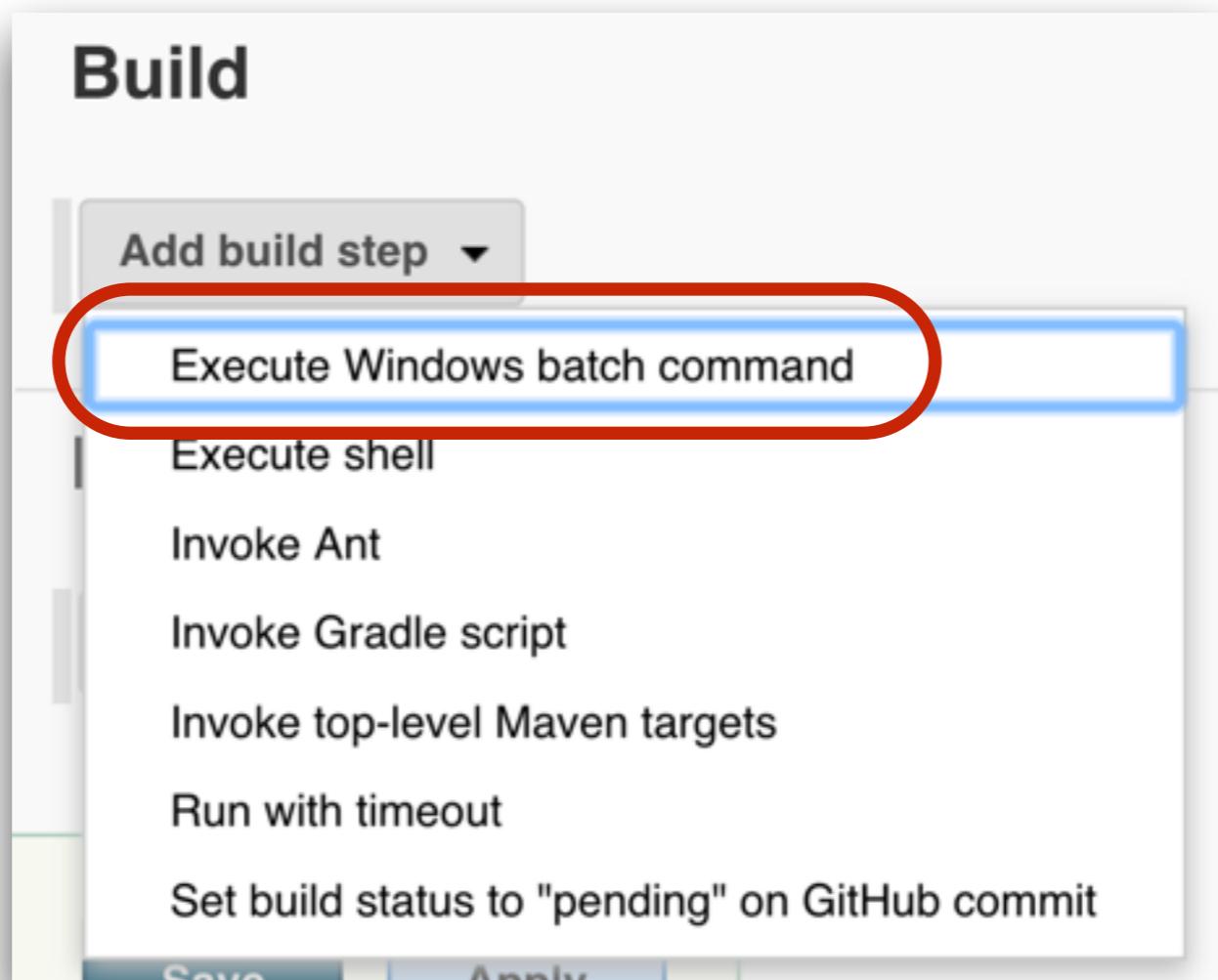


9. Add a Build step

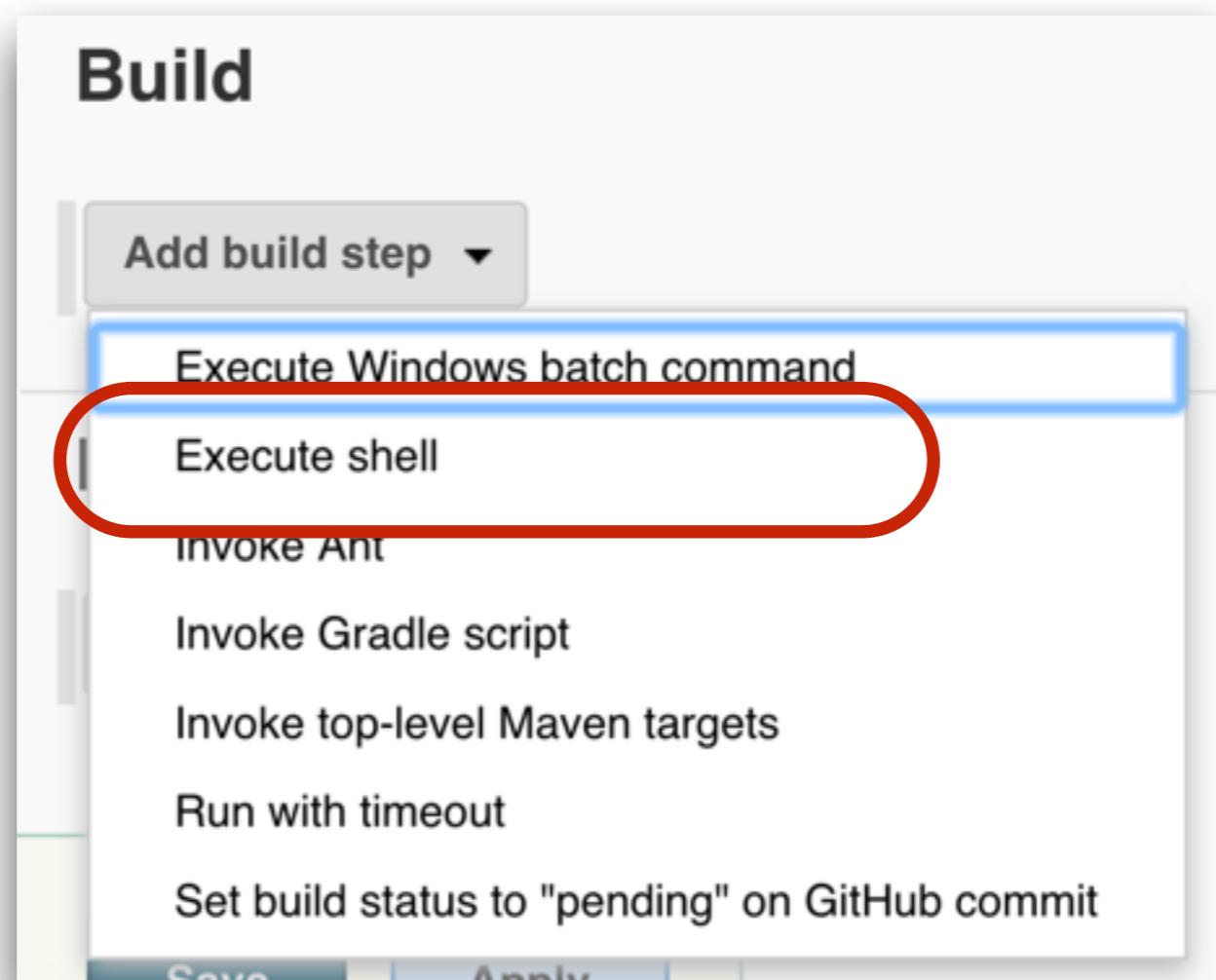
The screenshot shows a software interface for managing build configurations. At the top, there are tabs for General, Source Code Management, Build Triggers, Build Environment, **Build**, and Post-build Actions. Below these tabs, the word "BUILD ENVIRONMENT" is displayed in bold capital letters. Under the Build tab, there is a section titled "Build" with a sub-section titled "Add build step". A dropdown menu is open, 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, the 'Build Environment' section is visible. A dropdown menu is open over the 'E-mail Notification' option in the list of post-build actions. The list includes:

- 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** (highlighted with a blue background)
- Editable Email Notification
- Set GitHub commit status (universal)
- Set build status on GitHub commit [deprecated]
- Delete workspace when build is done

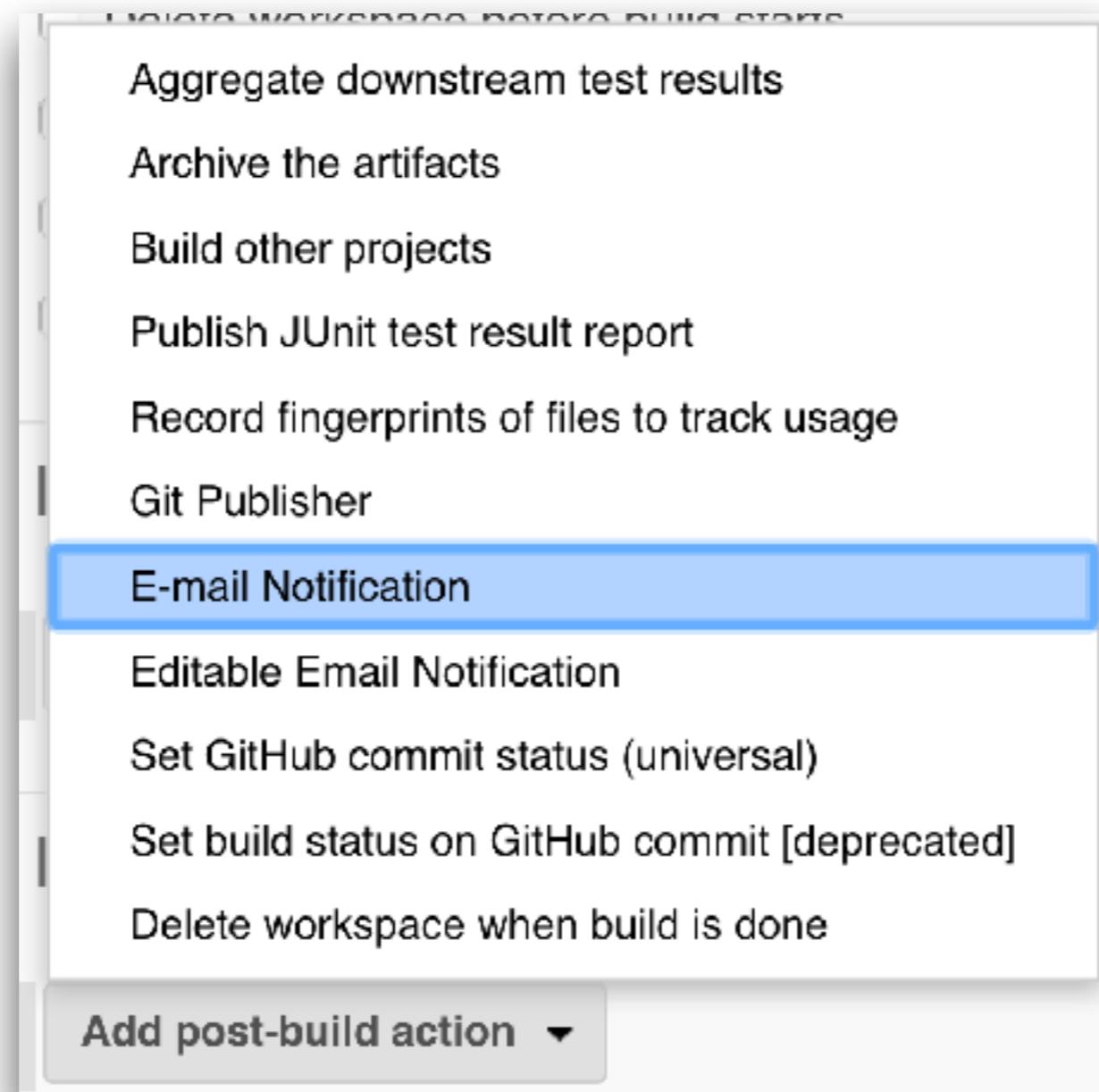
At the bottom left of the interface, there are two buttons: 'Save' and 'Apply'.



Email notification



Email notification



Add Recipients

Post-build Actions

E-mail Notification X ?

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's a sidebar with links: New Item, People, Build History, Manage Jenkins (which is circled with a red circle labeled '1'), My Views, and Credentials. Below the sidebar are two collapsed sections: Build Queue (No builds in the queue) and Build Executor Status. On the right, under the heading 'Manage Jenkins', there are several configuration options: 'Configure System' (circled 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.

Jenkins

New Item

People

Build History

1

Manage Jenkins

My Views

Credentials

Build Queue

No builds in the queue.

Build Executor Status

2

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
- [Global Tool Configuration](#)
Configure tools, their locations and automatic instal
- [Reload Configuration from Disk](#)
Discard all the loaded data in memory and reload ev



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

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



11. Run your job !!

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with links: New Item, People, Build History, Manage Jenkins, My Views, and Credentials. The main area displays a table of jobs. A single job, named "hello", is listed with a yellow sun icon. The table has columns: S, W, Name, Last Success, Last Failure, and Last Duration. Below the table, it says "Icon: S M L". At the bottom right of the main area, there are links for "Legend", "RSS for all", "RSS for failures", and "RSS for just latest builds". A red circle highlights the "hello" job entry in the table.

| S | W | Name | Last Success | Last Failure | Last Duration |
|---|---|-----------------------|--------------|--------------|---------------|
| | | hello | N/A | N/A | |

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

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



Job status

[add description](#)

All [+](#)

| 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

[add description](#)

All [+](#)

| S | W | Name ↓ | Last Success | Last Failure | Last Duration |
|---|------------------|-----------------------|--------------|--------------|---------------|
| | (circled in red) | hello | 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 |
|---|---|-----------------------|--------------|--------------|---------------|
| | | hello | N/A | N/A | N/A |

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

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



Build job

[add description](#)

All

| 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](#)



Jenkins build log

The screenshot shows the Jenkins dashboard for a job named "hello". The job status is "Success" (green circle) and the last success was 4.9 sec ago. A context menu is open over the "Last Success" entry, listing four options: "Changes", "Console Output" (which is highlighted in blue), "Edit Build Information", and "Delete Build". The "Console Output" option is currently selected.

| S | W ↓ | Name | Last Success | Last Failure | Last Duration |
|---|-----|-----------------------|--------------|--------------|---------------|
| | | hello | 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



Jenkins Home directory



Default Jenkins Home

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 labeled '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 oval labeled '2'), 'Configure Global Security', 'Configure Credentials', 'Global Tool Configuration', and 'Reload Configuration from Disk'. Each option has a brief description below it.

| Option | Description |
|--|---|
| 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 |
| Global Tool Configuration | Configure tools, their locations and automatic instal |
| Reload Configuration from Disk | Discard all the loaded data in memory and reload ev |



Default Jenkins Home

The screenshot shows the Jenkins configuration interface. On the left, there's a sidebar with links: New Item, People, Build History, Manage Jenkins, My Views, and Credentials. Below these are two expandable sections: Build Queue (No builds in the queue) and Build Executor Status (1 Idle, 2 Idle). The main content area has several configuration sections. One section, "Home directory", is highlighted with a red box and contains the path "/Users/somkiat/data/slides/ci-cd/swpark/software/keep". There's also a "System Message" section with a large empty box and a "[Plain text] Preview" section showing the value "2". Other sections include "# of executors" (set to 2), "Labels" (empty), "Usage" (set to "Use this node as much as possible"), "Quiet period" (set to 5), "SCM checkout retry count" (set to 0), and checkboxes for "Restrict project naming" and "Environment variables". At the bottom are "Save" and "Apply" buttons.

Jenkins configuration screen showing the Home directory set to `/Users/somkiat/data/slides/ci-cd/swpark/software/keep`.

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

System Message:

[Plain text] Preview: 2

of executors: 2

Labels:

Usage: Use this node as much as possible

Quiet period: 5

SCM checkout retry count: 0

Global properties:

- Restrict project naming
- Environment variables

Save Apply



Default Jenkins Home

The screenshot shows the Jenkins configuration page. On the left, there's a sidebar with links like New Item, People, Build History, Manage Jenkins, My Views, and Credentials. Below that are sections for Build Queue (empty) and Build Executor Status (2 Idle). The main area is titled "configuration" and contains several configuration items:

- Home directory:** /Users/somkiat/data/slide/ci-cc/swpark/software/keep (with a help icon)
- Workspace Root Directory:** \${JENKINS_HOME}/workspace/\${ITEM_FULLNAME} (with a help icon)
- Build Record Root Directory:** \${ITEM_ROOTDIR}/builds (with a help icon)
- System Message:** (empty text area with a help icon)
- # of executors:** 2 (with a help icon)
- Labels:** (empty text area with a help icon)
- Usage:** Use this node as much as possible (with a help icon)
- Quiet period:** 5 (with a help icon)

A red box highlights the workspace settings (Home directory, Workspace Root Directory, and Build Record Root Directory).



Change Jenkins's Home

For Windows

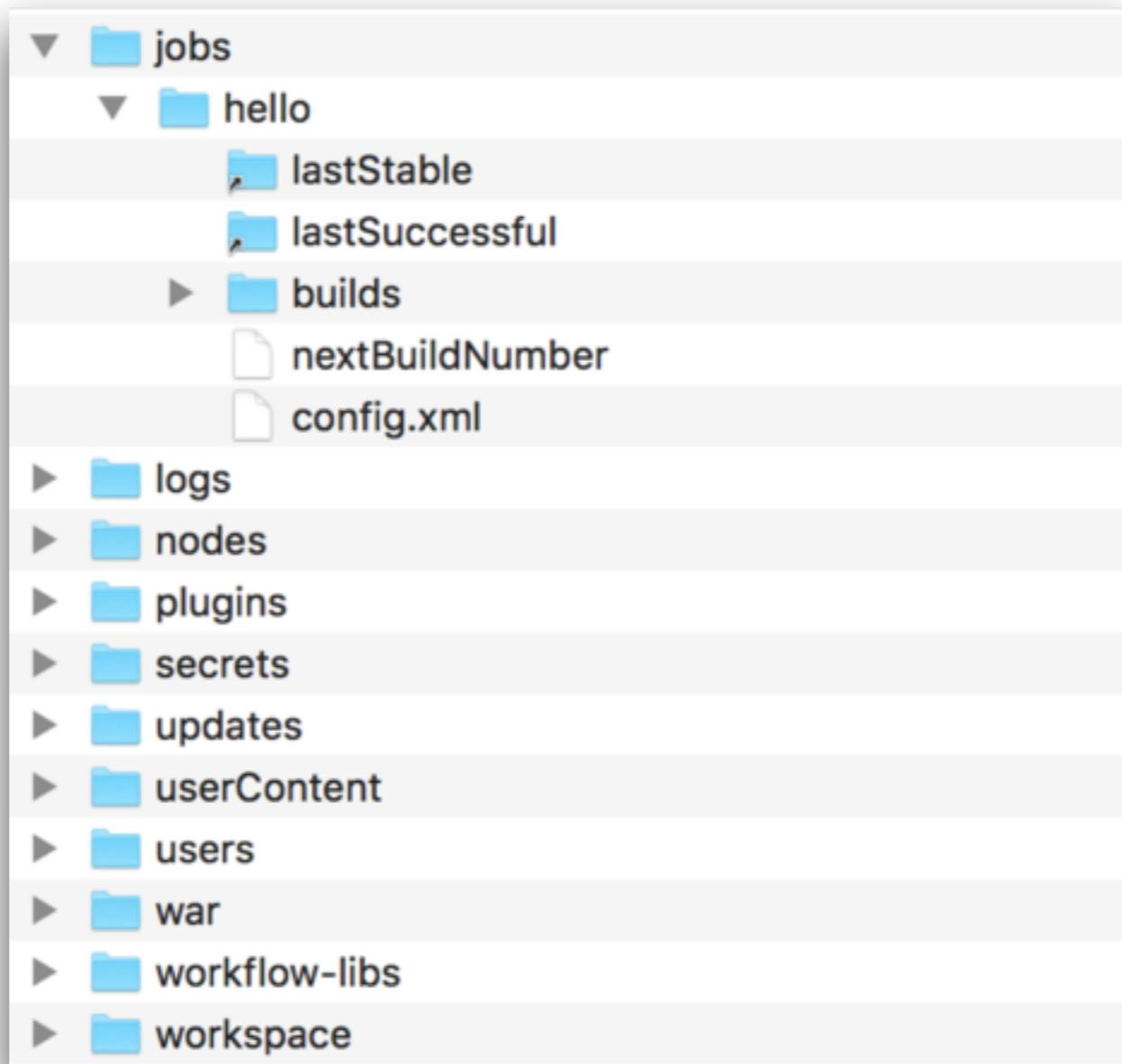
```
$set JENKINS_HOME=<your path>
```

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



Backup Jenkins



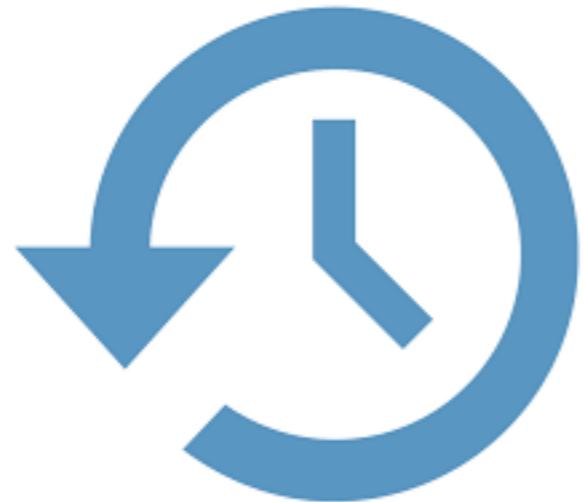
Backup ?

what to backup ?
when to backup ?
how to backup ?



Backup Jenkins

All files and folders in Jenkins Home



Backup Jenkins

Create Jenkins Job to backup Jenkins



Backup requirement ?

Backup with 7-zip
Backup Jenkins Home folder



1. Create new job

Enter an item name
backup_jenkins » 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, folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



2. Build trigger

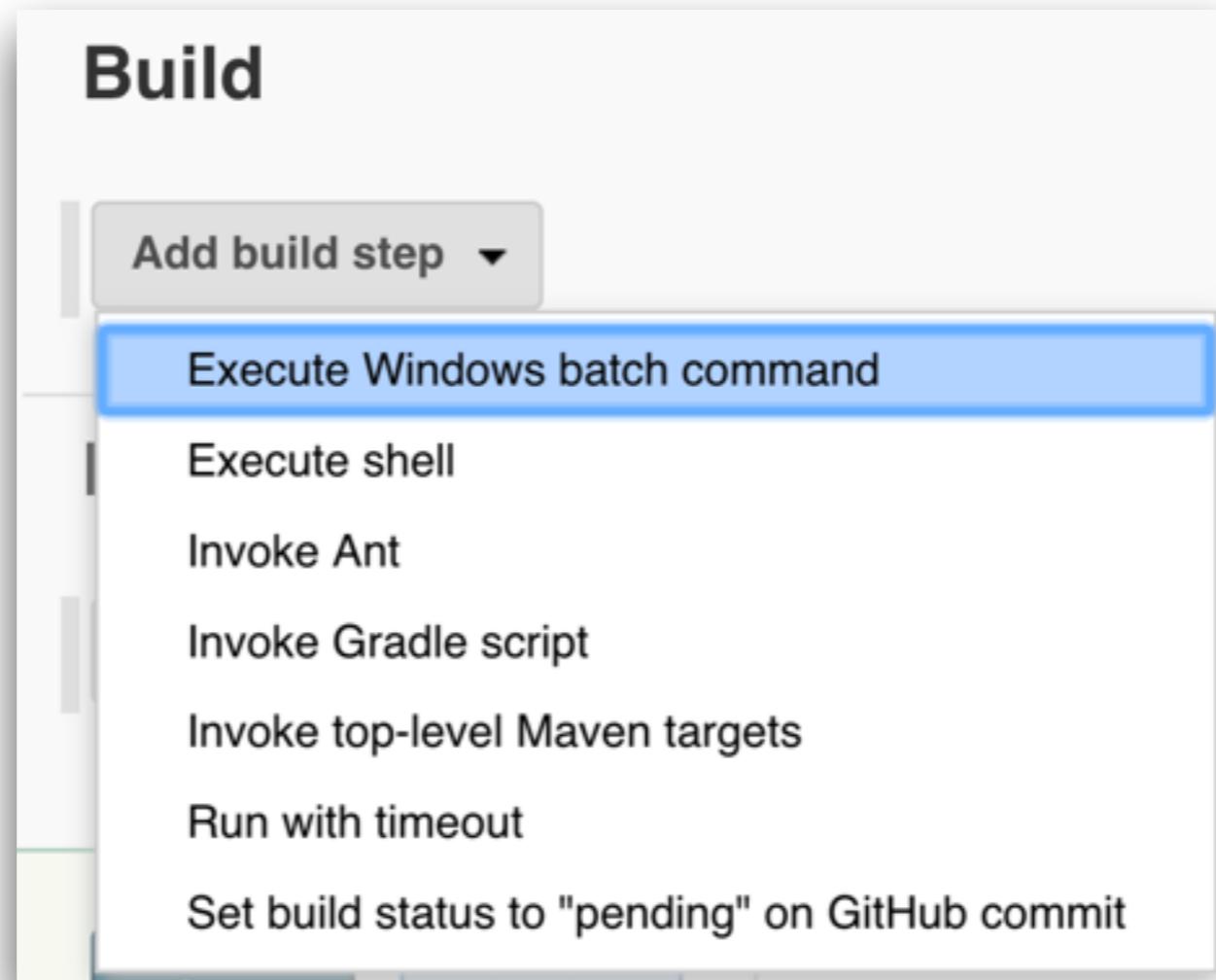
Using Build periodically => H 23 * * 7

The screenshot shows the 'Build Triggers' tab selected in the Jenkins configuration interface. Under the 'Build Triggers' section, the 'Build periodically' checkbox is checked, and the schedule is set to 'H 23 * * 7'. A tooltip below the schedule indicates the next run will be on Sunday, June 18, 2017, at 11:40:39 PM ICT. Other trigger options like 'Trigger builds remotely' and 'GitHub hook trigger for GITScm polling' are also listed.

| Trigger Type | Schedule | Description |
|--|------------|--|
| Trigger builds remotely (e.g., from scripts) | | Would last have run at Sunday, June 11, 2017 11:40:39 PM ICT; would next run at Sunday, June 18, 2017 11:40:39 PM ICT. |
| Build after other projects are built | | |
| Build periodically | H 23 * * 7 | Would last have run at Sunday, June 11, 2017 11:40:39 PM ICT; would next run at Sunday, June 18, 2017 11:40:39 PM ICT. |
| GitHub hook trigger for GITScm polling | | |
| Poll SCM | | |



3. Build to execute command



4.1 Write backup script

Windows

REM Store the current date inside a variable named "DATE"

```
for /f %%i in ('date /t') do set DATE=%%i
```

REM 7-Zip command to create an archive

```
"C:\Program Files\7-Zip\7z.exe" a -t7z
```

```
C:\Jenkins_Backup\Backup_%DATE%.7z C:\Jenkins\*
```



4.2 Write backup script

Mac OS

```
for /f %%i in ('date /t') do set DATE=%%i  
7z a -t7z $BACKUP\Backup_$DATE.7z $JENKINS_HOME\*
```



On Mac

Build

Execute shell

X **?**

Command

```
BACKUP=/Users/somkiat/data/slide/ci-cd/swpark/software/backup
DATE=`date +%Y-%m-%d`
7z a -t7z $BACKUP/Backup_${DATE}.7z $JENKINS_HOME/*
```

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

Advanced...

Add build step ▾



5. Save and build

The screenshot shows the Jenkins interface for a project named "backup_jenkins". The left sidebar contains links: "Back to Dashboard", "Status", "Changes", "Workspace", "Build Now" (which is highlighted with a red box), "Delete Project", and "Configure". Below this is a "Build History" section with a search bar and RSS links for "RSS for all" and "RSS for failures". The main content area is titled "Project backup_jenkins" and includes links to "Workspace" and "Recent Changes". At the bottom is a "Permalinks" section.

Jenkins

search

Project backup_jenkins

Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Project

Configure

Build History

trend

find

RSS for all RSS for failures

Workspace

Recent Changes

Permalinks



6. See build history

The screenshot shows the Jenkins interface for the project 'backup_jenkins'. The top navigation bar includes the Jenkins logo, a search bar, and user information ('somkiat' and 'log out'). A sidebar on the left provides links to 'Back to Dashboard', 'Status', 'Changes', 'Workspace', 'Build Now', 'Delete Project', and 'Configure'. The main content area displays the project name 'Project backup_jenkins' and two links: 'Workspace' and 'Recent Changes'. On the right, there are buttons for 'add description' and 'Disable Project'. A red box highlights the 'Build History' section on the left, which lists the most recent build (#2) from June 14, 2017, at 9:51 PM. The build details include 'Changes', 'Console Output' (which is selected and highlighted in blue), 'Edit Build Information', and 'Delete Build'.

Project backup_jenkins

Workspace

Recent Changes

Build History

trend =

find

#2 Jun 14, 2017 9:51 PM

- Changes
- Console Output
- Edit Build Information
- Delete Build

Permalinks

- Last build (#2), 1 min 59 sec ago
- Last stable build (#2), 1 min 59 sec ago
- Last successful build (#2), 1 min 59 sec ago
- Last completed build (#2), 1 min 59 sec ago



Console output

Jenkins > backup_jenkins > #2

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

Console Output

```
Started by user somkiat
Building in workspace /Users/somkiat/data/slide/ci-cd/swpark/software/keep/workspace/backup_jenkins
[backup_jenkins] $ /bin/sh -xe /var/folders/t5/8kg23e_97z9dw44tfsld6dqw000Cgn/T/hudson6420145093817228191.sh
+ BACKUP=/Users/somkiat/data/slide/ci-cd/swpark/software/backup
++ date +%Y-%m-%d
+ DATE=2017-06-14
+ 7z a -t7z /Users/somkiat/data/slide/ci-cd/swpark/software/backup/Backup_2017-06-14.7z /Users/somkiat/data/slide/ci-cd/swpark/software/keep/ccnfig.xml /Users/somkiat/data/slide/ci-cd/swpark/software/keep/hudson.model.UpdateCenter.xml /Users/somkiat/data/slide/ci-cd/swpark/software/keep/hudson.plugins.emailtext.ExtendedEmailPublisher.xml /Users/somkiat/data/slide/ci-cd/swpark/software/keep/hudson.plugins.git.GitTool.xml /Users/somkiat/data/slide/ci-cd/swpark/software/keep/identity.key.enc /Users/somkiat/data/slide/ci-cd/swpark/software/keep/jenkins.CLI.xml /Users/somkiat/data/slide/ci-cd/swpark/software/keep/jenkins.install.InstallUtil.installingPlugins /Users/somkiat/data/slide/ci-cd/swpark/software/keep/jenkins.install.InstallUtil.lastExecVersion /Users/somkiat/data/slide/ci-cd/swpark/software/keep/jenkins.install.UpgradeWizard.state /Users/somkiat/data/slide/ci-cd/swpark/software/keep/jobs /Users/somkiat/data/slide/ci-cd/swpark/software/keep/logs /Users/somkiat/data/slide/ci-cd/swpark/software/keep/nodeMonitors.xml /Users/somkiat/data/slide/ci-cd/swpark/software/keep/nodes /Users/somkiat/data/slide/ci-cd/swpark/software/keep/plugins /Users/somkiat/data/slide/ci-cd/swpark/software/keep/secret.key /Users/somkiat/data/slide/ci-cd/swpark/software/keep/secret.key.not-so-secret /Users/somkiat/data/slide/ci-cd/swpark/software/keep/secrets /Users/somkiat/data/slide/ci-cd/swpark/software/keep/updates /Users/somkiat/data/slide/ci-cd/swpark/software/keep/userContent /Users/somkiat/data/slide/ci-cd/swpark/software/keep/users /Users/somkiat/data/slide/ci-cd/swpark/software/keep/var /Users/somkiat/data/slide/ci-cd/swpark/software/keep/workspace
/Users/somkiat/data/slide/ci-cd/swpark/software/keep/workspace

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2015-05-21
p7zip Version 15.02 (locale=utf8,Utf16=on,HugeFiles=on,64 bits,4 CPUs x64)

scanning the drive:
703 folders, 2801 files, 203646776 bytes (195 MiB)

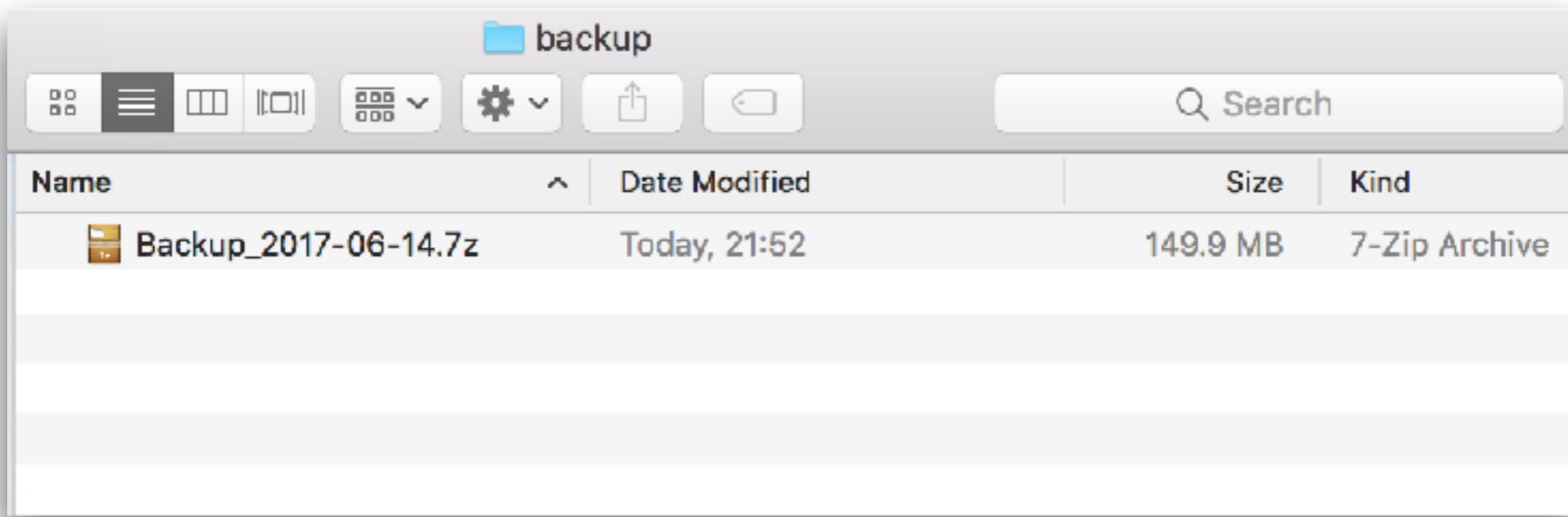
creating archive: /Users/somkiat/data/slide/ci-cd/swpark/software/backup/Backup_2017-06-14.7z

Items to compress: 3504

Files read from disk: 2726
Archive size: 149922651 bytes (143 MiB)
Everything is Ok
Finished: SUCCESS
```



See backup file



Question :: How to restore ?



Manage & Config plugins



Jenkins Plugins

Power from plugins
+1000 community

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



บริษัท สยามชนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

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), 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 red and white jetpack-like suit. 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 the placeholder text "Find plugins..." and a magnifying glass icon.

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



บริษัท สยามชนาญกิจ จำกัด และเพื่อนพ้องน้องพี่

Manage Plugins

Manage Jenkins -> Manage Plugins

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there is a sidebar with various links: New Item, People, Build History, Manage Jenkins (which is circled in red with the number 1), My Views, and Credentials. Below the sidebar are two sections: Build Queue (No builds in the queue) and Build Executor Status (1 Idle, 2 Idle). The main content area is titled "Manage Jenkins" and contains several configuration links. One link, "Manage Plugins", is highlighted with a red circle and the number 2, indicating it is the target of the tutorial. Other links include Configure System, Configure Global Security, Configure Credentials, Global Tool Configuration, Reload Configuration from Disk, System Information, System Log, and Load Statistics.

1

Manage Jenkins

Configure System
Configure global settings and paths.

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

Configure Credentials
Configure the credential providers and types

Global Tool Configuration
Configure tools, their locations and automatic installers.

Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful when you modified config files directly on disk.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

System Information
Displays various environmental information to assist trouble-shooting.

System Log
System log captures output from java.util.logging output related to Jenkins.

Load Statistics
Check your resource utilization and see if you need more computers for your builds.



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

Submit

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



Install Jenkins plugins



Periodic backup plugin

The screenshot shows the Jenkins plugin marketplace interface. A red circle highlights the number '1' in the top right corner, indicating one available update. Below it, a search bar contains the text 'periodic backup' with a magnifying glass icon. A red box highlights this search bar. At the top, there are tabs for 'Updates', 'Available' (which is selected), 'Installed', and 'Advanced'. The main table lists a single plugin:

| Install ↓ | Name | Version |
|--|------|---------|
| <input type="checkbox"/> Periodic Backup | | 1.3 |

Below the table are three buttons: 'Install without restart' (disabled), 'Download now and install after restart', and 'Check now'. To the right of the table, the text 'Update information obtained: 9 hr 52 min ago' is displayed.

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



Choose and install

Filter: periodic backup

Updates Available Installed Advanced

| Install ↓ | Name | Version |
|-------------------------------------|-----------------|---------|
| <input checked="" type="checkbox"/> | Periodic Backup | 1.3 |

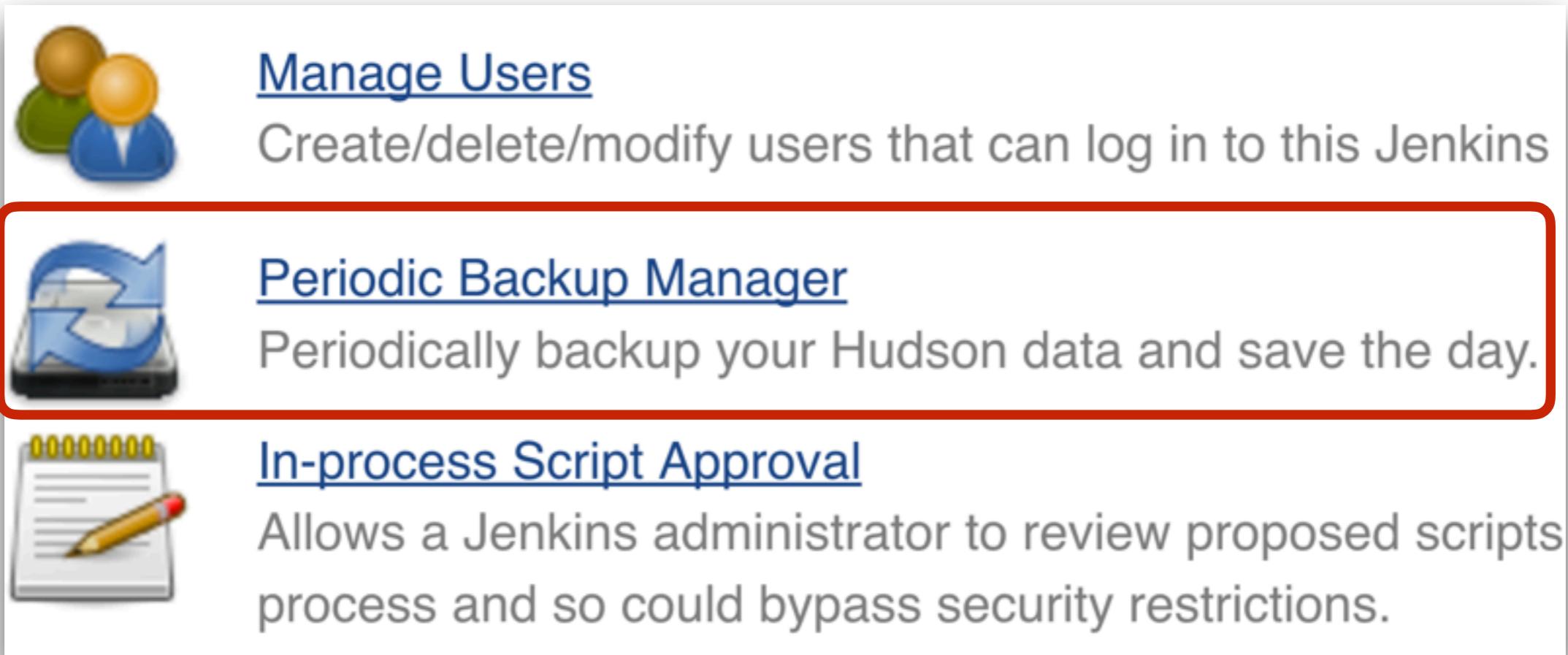
Install without restart Download now and install after restart

Update information obtained: 9 hr 52 min ago Check now



How to use ?

Manage Jenkins -> Periodic Backup Manager



[Manage Users](#)
Create/delete/modify users that can log in to this Jenkins

[Periodic Backup Manager](#)
Periodically backup your Hudson data and save the day.

[In-process Script Approval](#)
Allows a Jenkins administrator to review proposed scripts
process and so could bypass security restrictions.



Configuration !!

Jenkins Periodic Backup Manager

[Back to Dashboard](#)

[Restore](#)

[Backup Now!](#)

[Configure](#)

Root Directory: /Users/somkiat/data/slide/ci-cd/swpark/software/keep

Temporary Directory:

Backup schedule (cron):

Validate cron syntax

Maximum backups in location: 0

Store no older than (days): 0

File Management Strategy

ConfigOnly

FullBackup

Storage Strategy

Add Storage ▾

Backup Location

Add Location ▾

Save



Temporary Directory

Temporary folder of archive files
while restoring and backup



Backup schedule (cron)

The schedule that you want to backup

e.g. H 23 * * *



Maximum backups in location

Total number of backups
in backup location



Store no older than (days)

**Ensure any backup in any location
which is older than the number of days
is deleted automatically**



Configuration

| | | |
|-----------------------------|--|---------------------------------------|
| Root Directory | /Users/somkiat/data/slide/ci-cd/swpark/software/keep | |
| Temporary Directory | Users/somkiat/data/slide/ci-cd/swpark/software | (?) |
| Backup schedule (cron) | H 23 * * * | |
| | This cron is OK | <button>Validate cron syntax</button> |
| (?) | | |
| Maximum backups in location | 5 | (?) |
| Store no older than (days) | 15 | (?) |



File management strategy

File Management Strategy

- ConfigOnly
- FullBackup

ConfigOnly is backup only configuration



Jenkins driven by Plugins



Continuous Integration



Continuous Integration

Branching strategy

List of tools for Continuous Integration

Jenkins pipeline structure



Continuous Integration Design



Branching strategy

Master branch

Integration branch

Feature branch



Branching strategy

Master branch

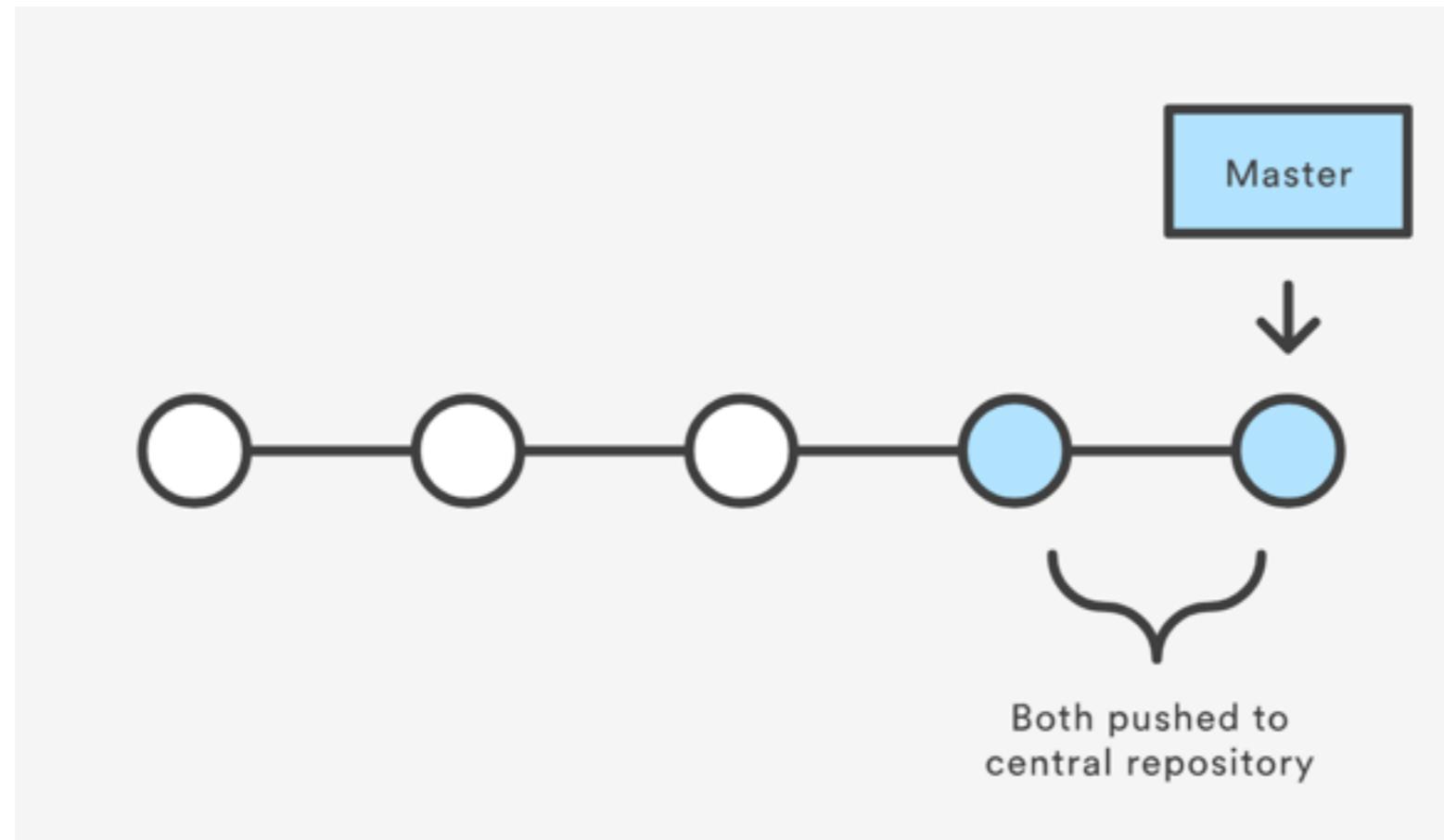
Master-develop branch

Integration branch

Feature branch



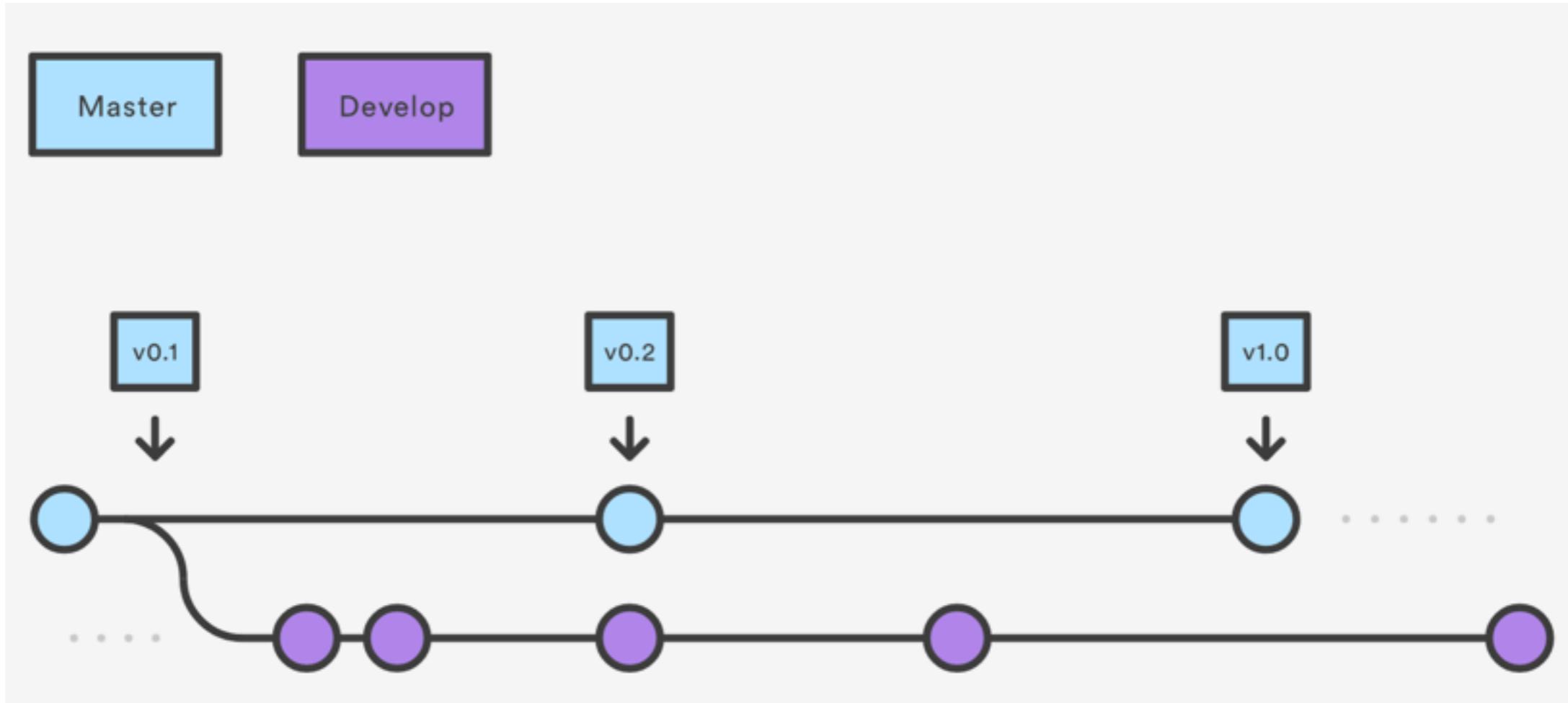
Master branch



<https://www.atlassian.com/git/tutorials/comparing-workflows>



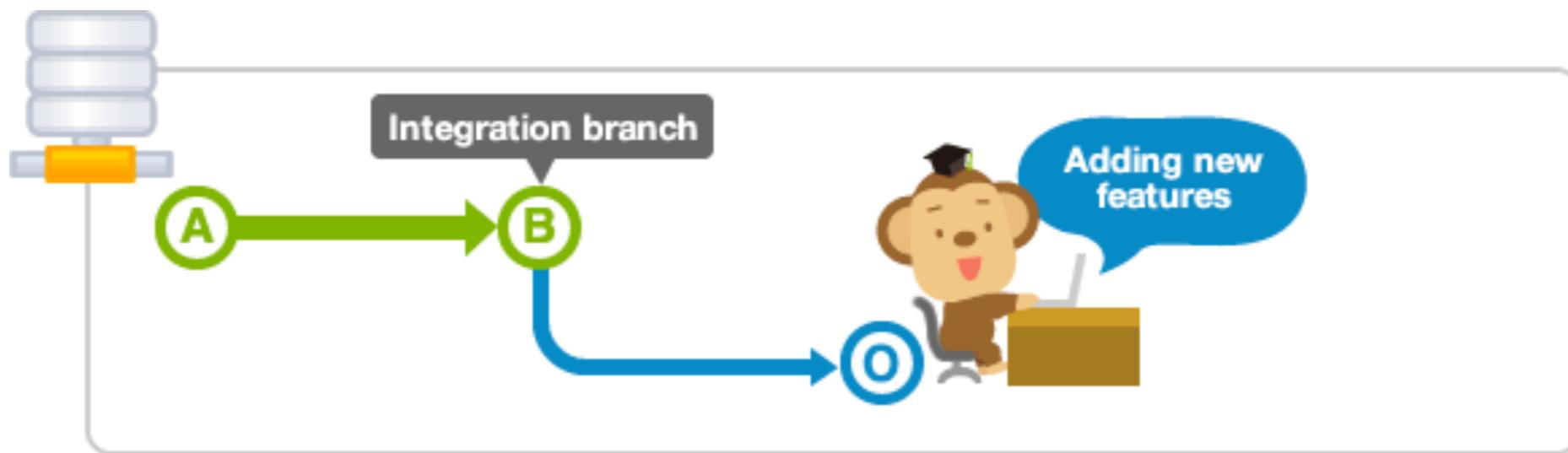
Master-develop branch



<https://www.atlassian.com/git/tutorials/comparing-workflows>



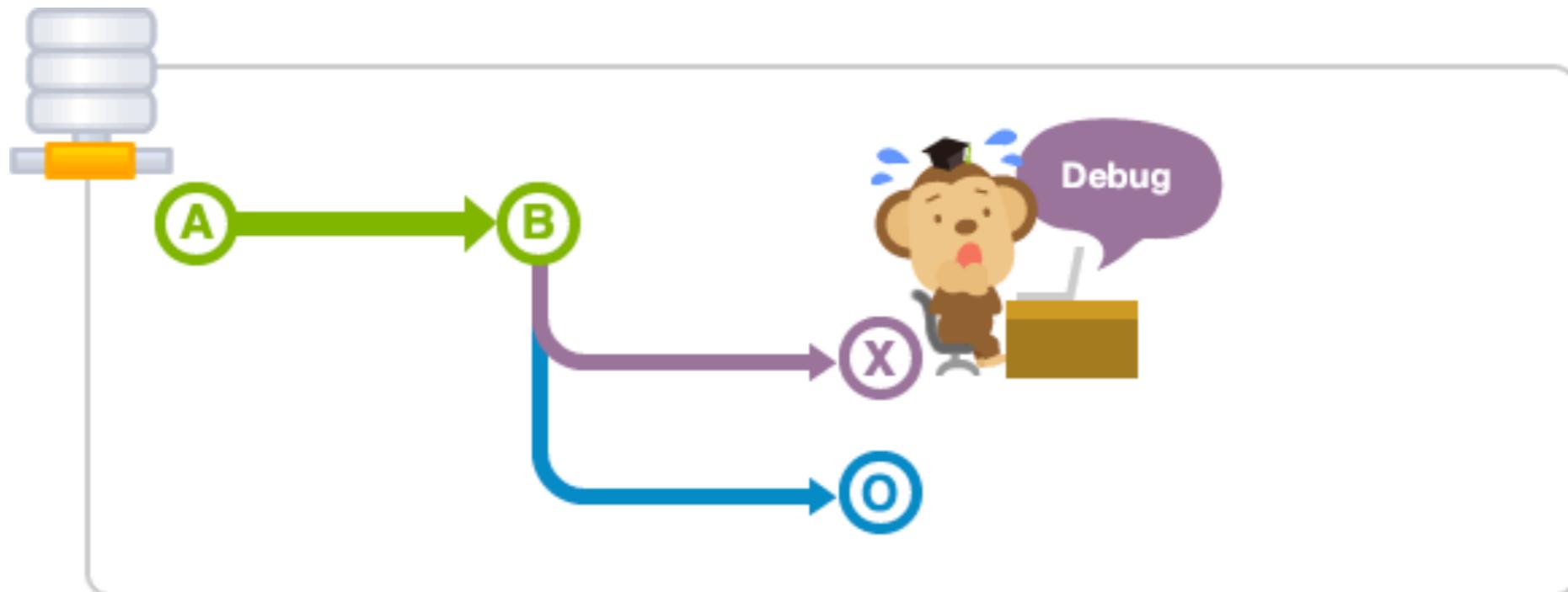
Integration branch



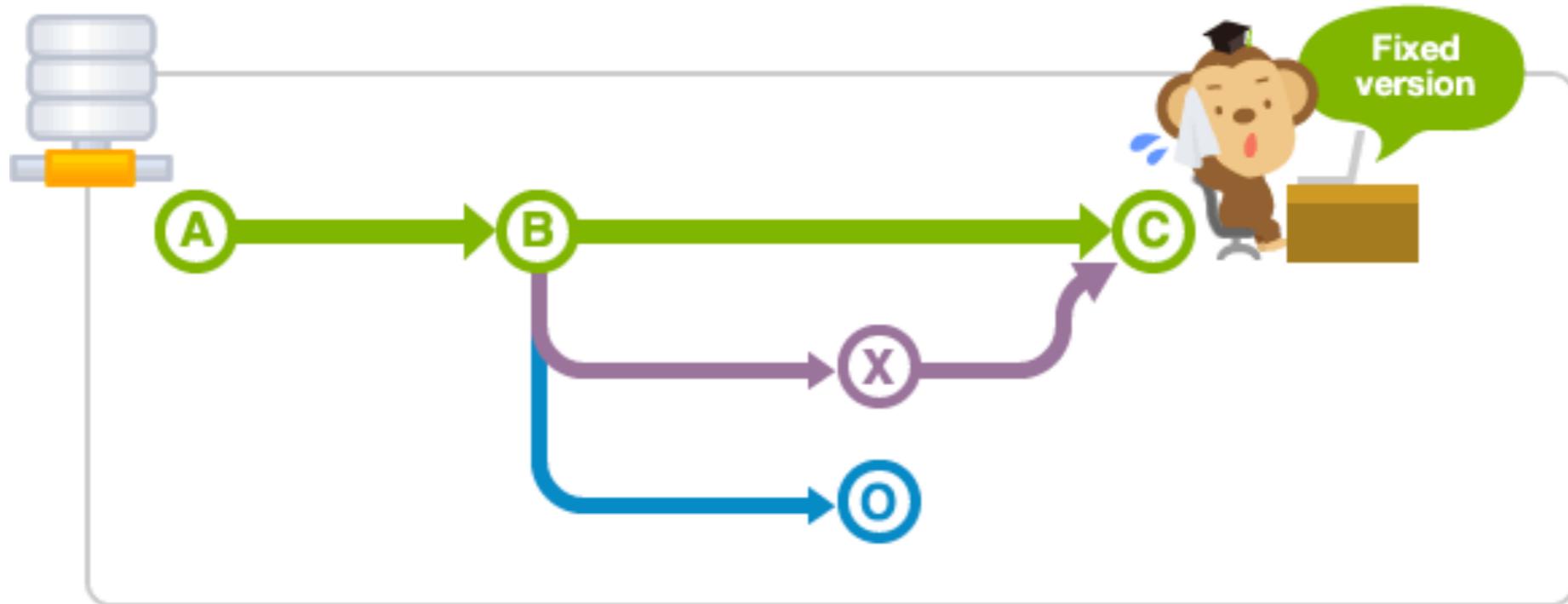
https://backlogtool.com/git-guide/en/stepup/stepup1_5.html



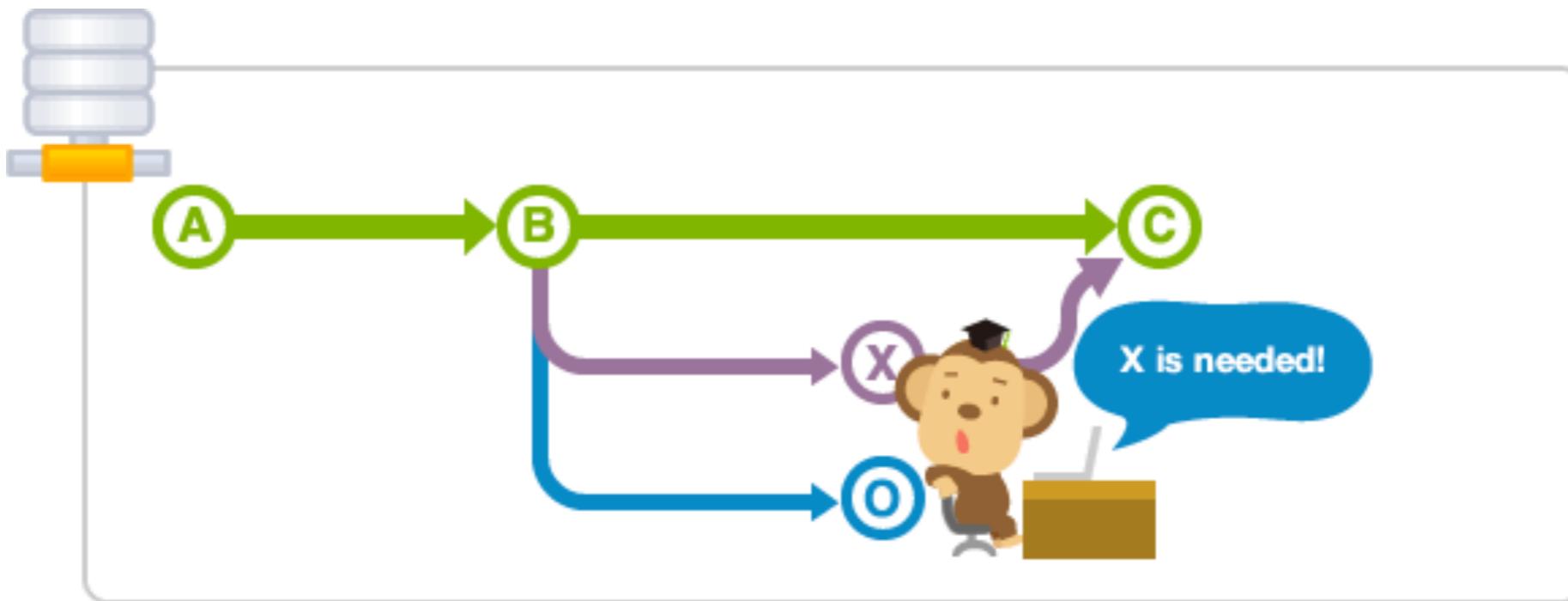
Integration branch



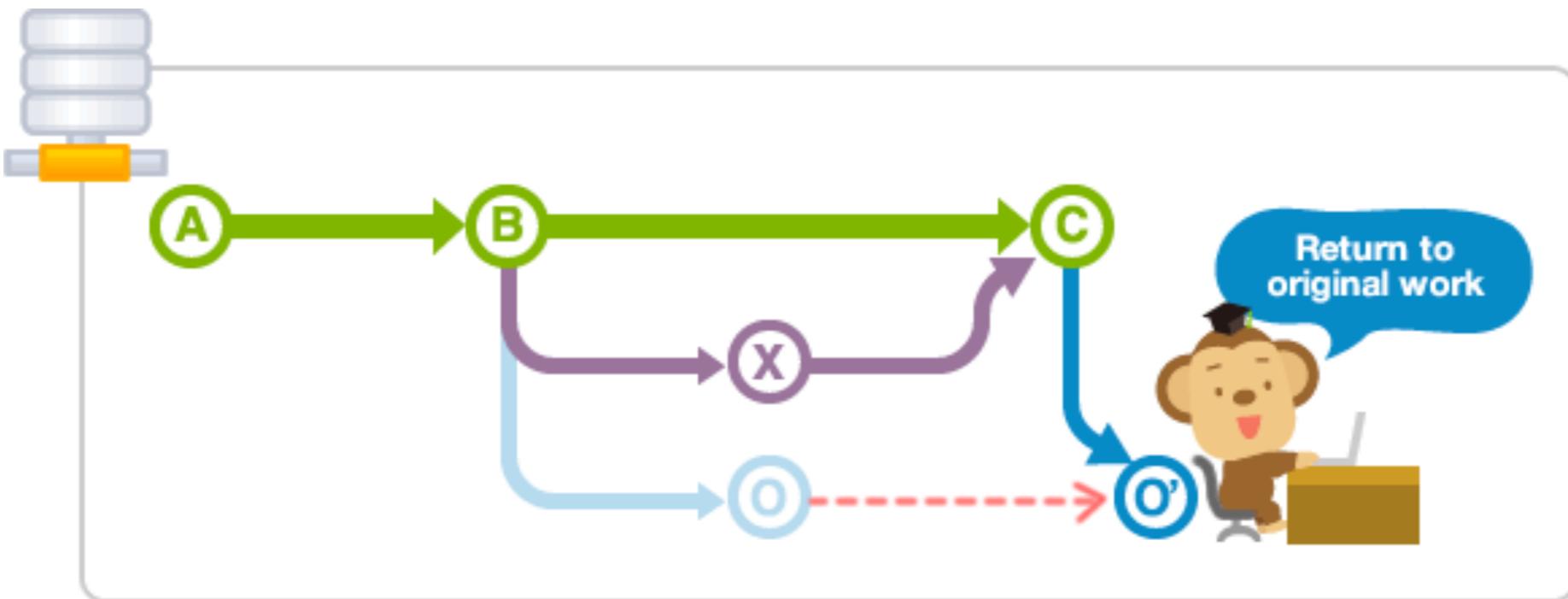
Integration branch



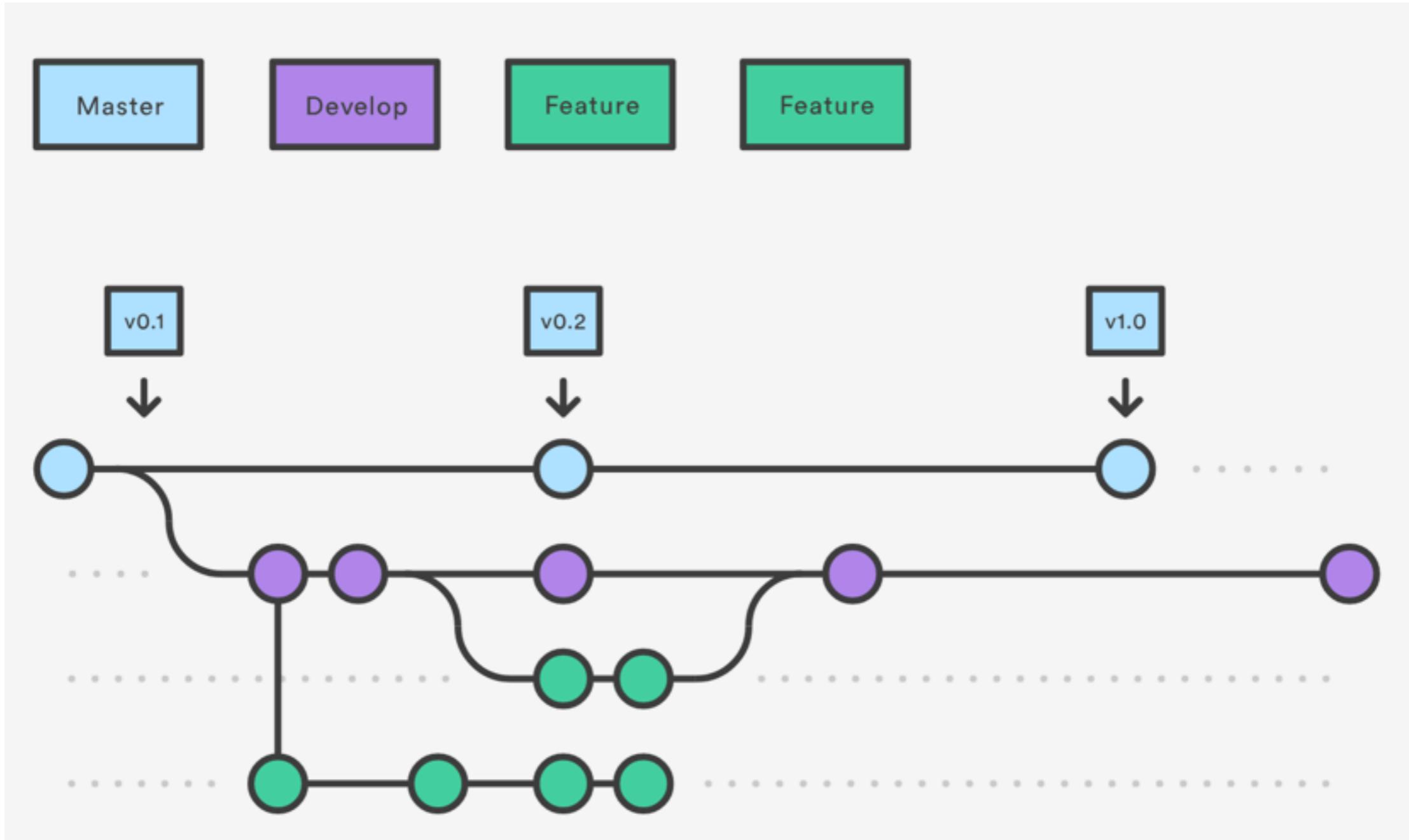
Integration branch



Integration branch



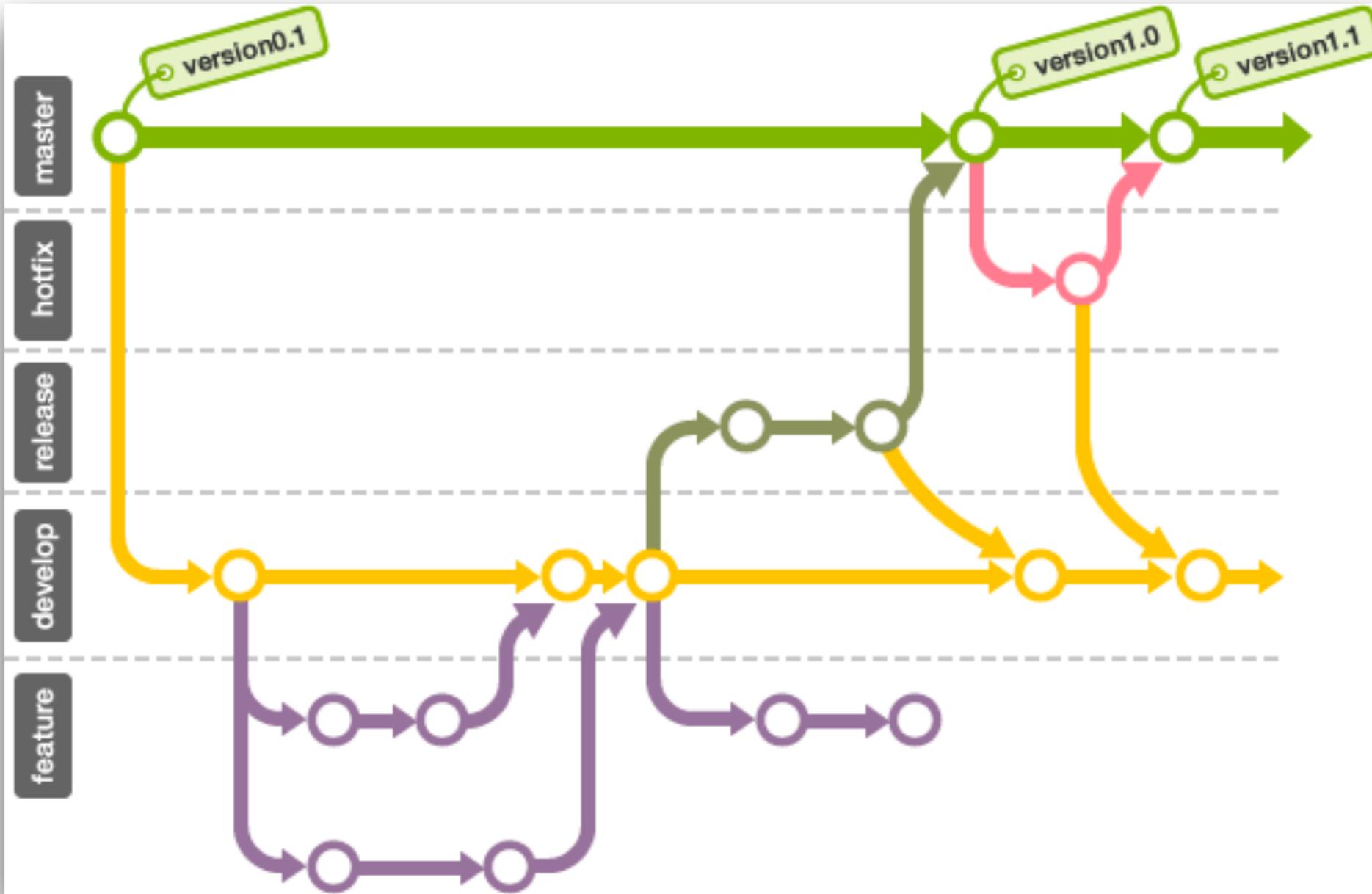
Feature branch



<https://www.atlassian.com/git/tutorials/comparing-workflows>



Gitflow model

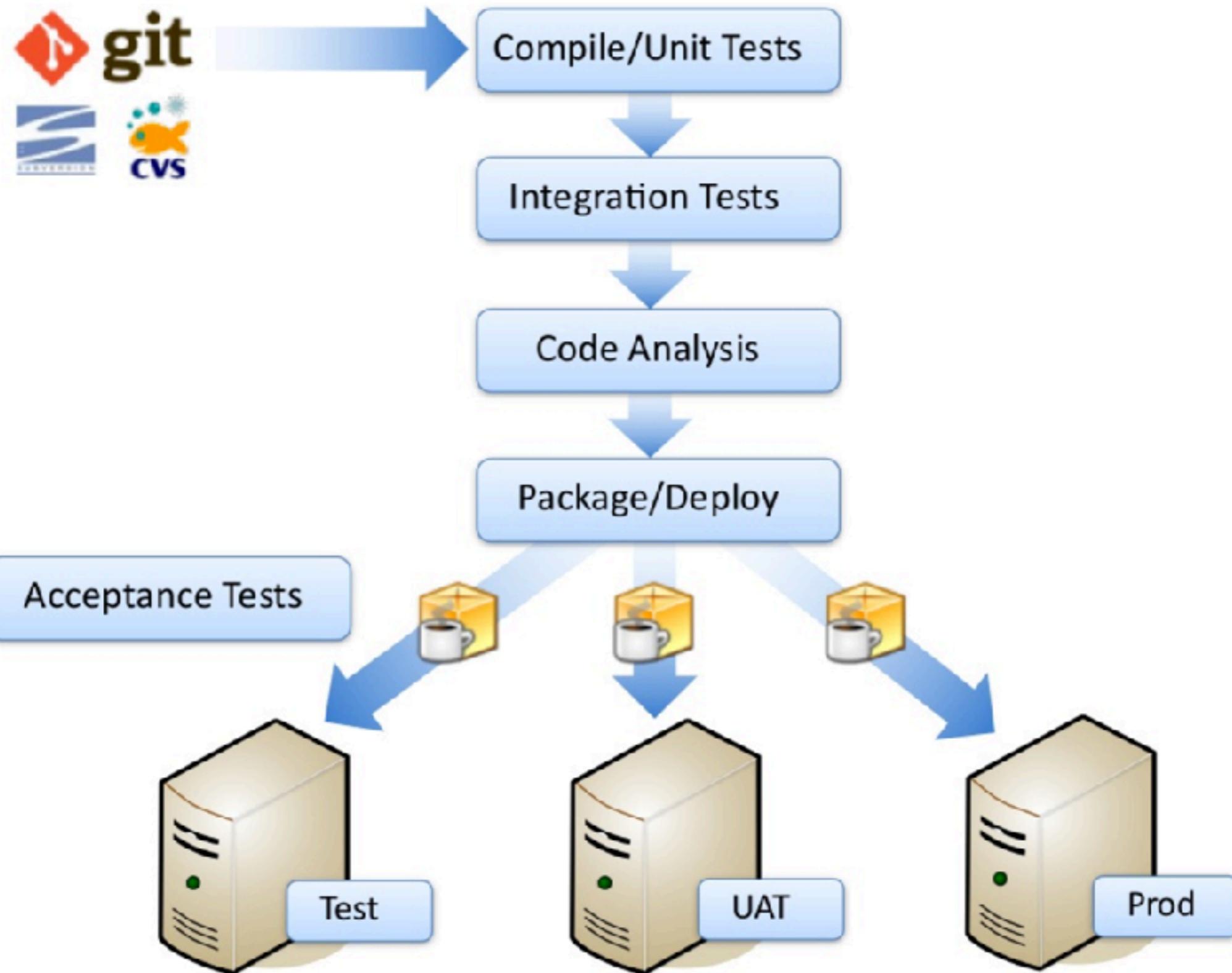


<http://nvie.com/posts/a-successful-git-branching-model/>



Build pipeline





Toolset for Continuous Integration



Example

| Technology | Description |
|------------------|--------------------------------|
| Java | Main programming language |
| Apache Maven | Build tool |
| JUnit | Unit and integration test tool |
| Cobertura | Code coverage tool |
| Robotframework | Acceptance test tool |
| Apache Tomcat | Java Web Server |
| Git | Version Control System |
| SonarQube | Static code analysis tool |
| Frog Artifactory | Keep artifact files |
| Jenkins | Continuous Integration Server |
| IntelliJ IDEA | IDE for Java development |



Setup Git

Download from <https://git-scm.com/>

The screenshot shows the official Git website (<https://git-scm.com/>). At the top left is the Git logo with the tagline "distributed even if your workflow isn't". A search bar is at the top right. The main content area features a diagram of five servers connected by red and yellow lines, representing a distributed workflow. To the left of the diagram, text describes Git as a free and open source distributed version control system designed for efficiency. Below this, another section highlights Git's ease of learning, fast performance, and unique features like cheap local branching and multiple workflows. A "Try Git" button with a user icon is present. The bottom half of the page is divided into several sections: "About" (with a gear icon), "Documentation" (with a book icon), "Downloads" (with a download arrow icon), and "Community" (with a speech bubble icon). On the right, a large box displays the "Latest source Release" (2.13.1) with "Release Notes (2017-06-05)" and a "Downloads for Mac" button.

git --distributed-even-if-your-workflow-isn't

Search entire site...

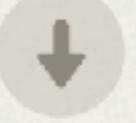
Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient staging areas, and **multiple workflows**.

 Learn Git in your browser for free with [Try Git](#).

 **About**
The advantages of Git compared to other source control systems.

 **Documentation**
Command reference pages, Pro Git book content, videos and other material.

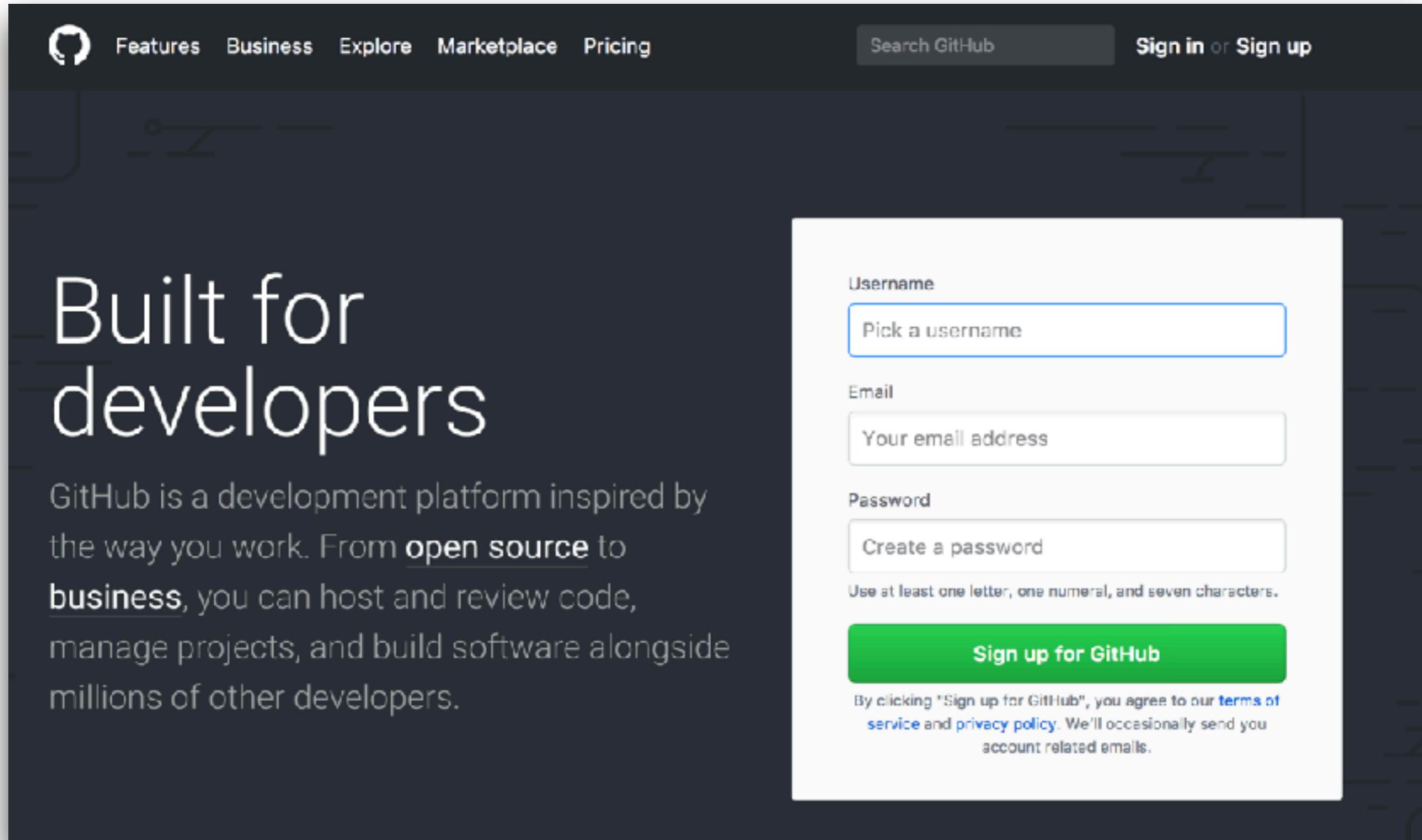
 **Downloads**
GUI clients and binary releases for all major platforms.

 **Community**
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release
2.13.1
Release Notes (2017-06-05)
Downloads for Mac



Create github.com account



The screenshot shows the GitHub homepage with a dark background. On the left, there's a large white text area containing the heading "Built for developers" and a descriptive paragraph about GitHub's purpose as a development platform for open source and business projects. At the top, there's a navigation bar with links for Features, Business, Explore, Marketplace, and Pricing. To the right of the navigation is a search bar labeled "Search GitHub" and a "Sign in or Sign up" button. A prominent white sign-up form is overlaid on the right side of the page. The form has three input fields: "Username" with placeholder "Pick a username", "Email" with placeholder "Your email address", and "Password" with placeholder "Create a password". Below the password field is a note: "Use at least one letter, one numeral, and seven characters." At the bottom of the form is a green "Sign up for GitHub" button. Below the button, a small note states: "By clicking 'Sign up for GitHub', you agree to our [terms of service](#) and [privacy policy](#). We'll occasionally send you account related emails."

Features Business Explore Marketplace Pricing

Search GitHub

Sign in or Sign up

Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside millions of other developers.

Username

Pick a username

Email

Your email address

Password

Create a password

Use at least one letter, one numeral, and seven characters.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#). We'll occasionally send you account related emails.



Setup Java

Download from <http://www.oracle.com/>



The screenshot shows the Oracle Java SE Downloads page. At the top, there's a navigation bar with the Oracle logo, a menu icon, a search bar, and user account options. Below the navigation, the breadcrumb path reads "Oracle Technology Network > Java > Java SE > Downloads". The main content area has tabs for "Overview", "Downloads" (which is selected), "Documentation", "Community", "Technologies", and "Training". On the left, a sidebar lists categories like Java SE, Java EE, Java ME, etc. The central part features two download cards: "Java Platform (JDK) 8u131" and "NetBeans with JDK 8". A callout box highlights the "Java Platform, Standard Edition" section, which contains information about Java SE 8u131, a note about MD5-signed JARs, and links for "Installation Instructions" and "Release Notes". To the right, there are sections for "Java SDKs and Tools" and "Java Resources", each with a list of links.

Java SE Downloads

Java Platform (JDK) 8u131

NetBeans with JDK 8

Java Platform, Standard Edition

Java SE 8u131

Java SE 8u131 includes important security fixes and bug fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release.

[Learn more](#)

Important planned change for MD5-signed JARs

Starting with the April Critical Patch Update releases, planned for April 18 2017, all JRE versions will treat JARs signed with MD5 as unsigned. [Learn more and view testing instructions](#).

For more information on cryptographic algorithm support, please check the JRE and JDK [Crypto Roadmap](#).

• Installation Instructions

• Release Notes

JDK DOWNLOAD

Java SDKs and Tools

- Java SE
- Java EE and Glassfish
- Java ME
- Java Card
- NetBeans IDE
- Java Mission Control

Java Resources

- Java APIs
- Technical Articles
- Demos and Videos
- Forums
- Java Magazine
- java.net
- Developer Training
- Tutorials
- Java.com



Setup Apache Maven

Download from maven.apache.org

The screenshot shows the Apache Maven Project website at <http://maven.apache.org>. The page features a large logo with the word "Maven" and a feather icon. The main navigation bar includes links for Apache, Maven, and Welcome to Apache Maven. A sidebar on the left lists various sections like Welcome, License, Download, Install, Configure, Run, IDE Integration, About Maven, What is Maven?, Features, FAQ, Support and Training, Documentation, Maven Plugins, Index (category), and Running Maven. The central content area is titled "Welcome to Apache Maven" and provides an overview of what Maven is and how it can be used. It highlights sections for Use (Download, Install, Run Maven), Extend (Write Maven Plugins), Contribute (Help Maven), Configure, Use Maven and Maven Plugins, Improve the Maven Repository, and Develop Maven.

Welcome to Apache Maven

Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

If you think that Maven could help your project, you can find out more information about in the "About Maven" section of the navigation. This includes an in-depth description of [what Maven is](#), a [list of some of its main features](#), and a set of frequently asked questions about [what Maven is](#).

This site is separated into the following sections, depending on how you'd like to use Maven:

| Use | Download, Install, Run Maven | Configure, Use Maven and Maven Plugins |
|------------|-------------------------------------|--|
| Extend | Write Maven Plugins | Improve the Maven Repository |
| Contribute | Help Maven | Develop Maven |



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".



Configuration Jenkins



Configuration !!

Manage Jenkins -> Global Tool Configuration

The screenshot shows the Jenkins Manage Jenkins interface. On the left sidebar, the 'Manage Jenkins' item is circled with a red circle labeled '1'. In the main content area, the 'Global Tool Configuration' item under the 'Configure System' section is also circled with a red circle labeled '2'.

Manage Jenkins

- New Item
- People
- Build History
- Manage Jenkins**
- My Views
- Credentials

Build Queue
No builds in the queue.

Build Executor Status

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

Global Tool Configuration
Configure tools, their locations and automatic instal

Reload Configuration from Disk
Discard all the loaded data in memory and reload ev



Java/JDK

 **Global Tool Configuration**

Maven Configuration

Default settings provider: Use default maven settings

Default global settings provider: Use default maven global settings

JDK

JDK installations

| | |
|---|---|
|  JDK | Name: <input type="text" value="JDK8"/> |
| JAVA_HOME: | <input type="text" value="your java home"/> |
| <input type="checkbox"/> Install automatically | ? |
| Delete JDK | |

[Add JDK](#)

List of JDK installations on this system



Apache Maven

Maven

Maven installations

| | |
|---|-------------------------------------|
|  Maven | <input type="checkbox"/> Name maven |
| MAVEN_HOME | your maven home |

Install automatically (?)

Delete Maven

Add Maven

List of Maven installations on this system



Git

Git

Git installations



Git

Name

Default

Path to Git executable

git



Install automatically



Delete Git

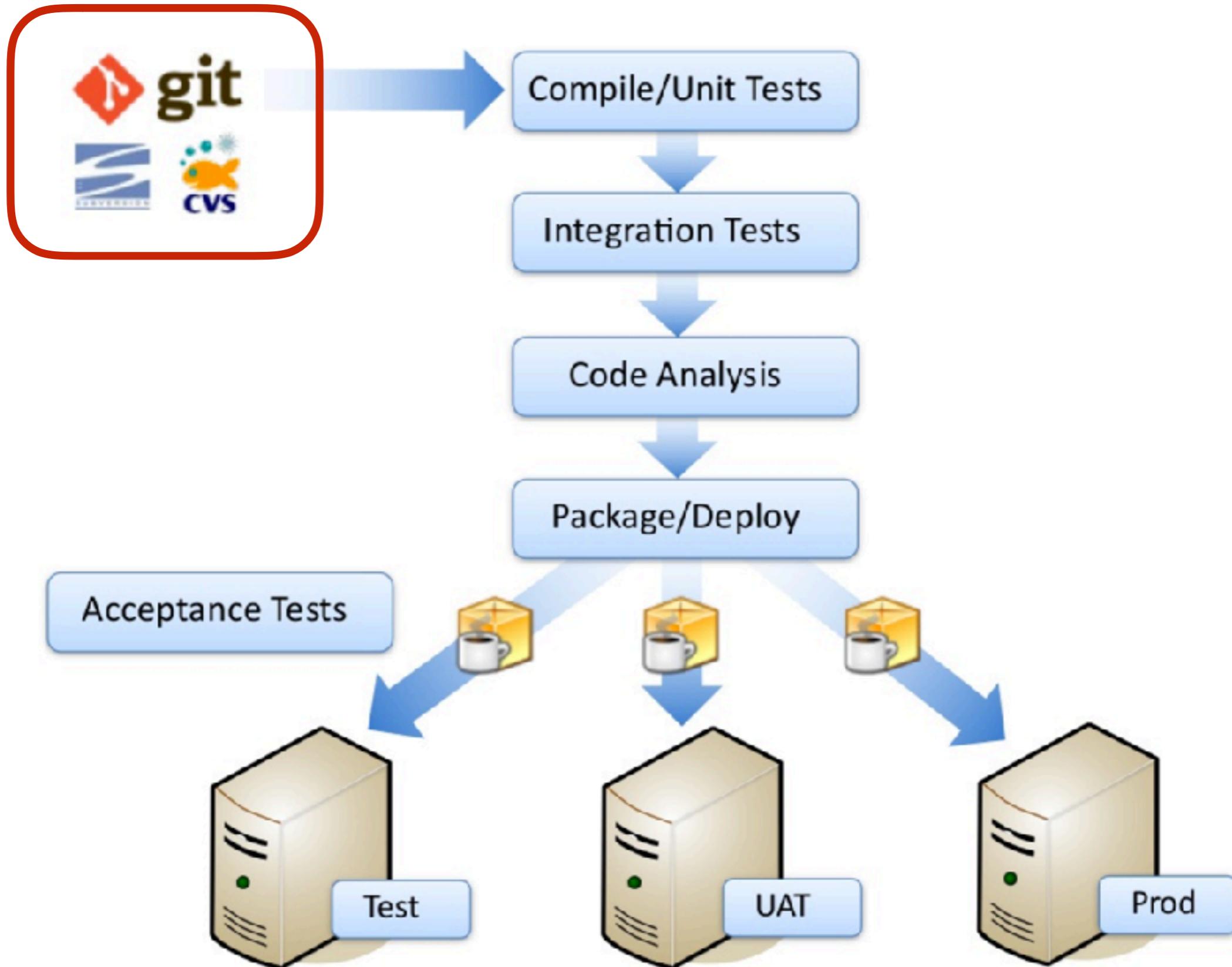
Add Git

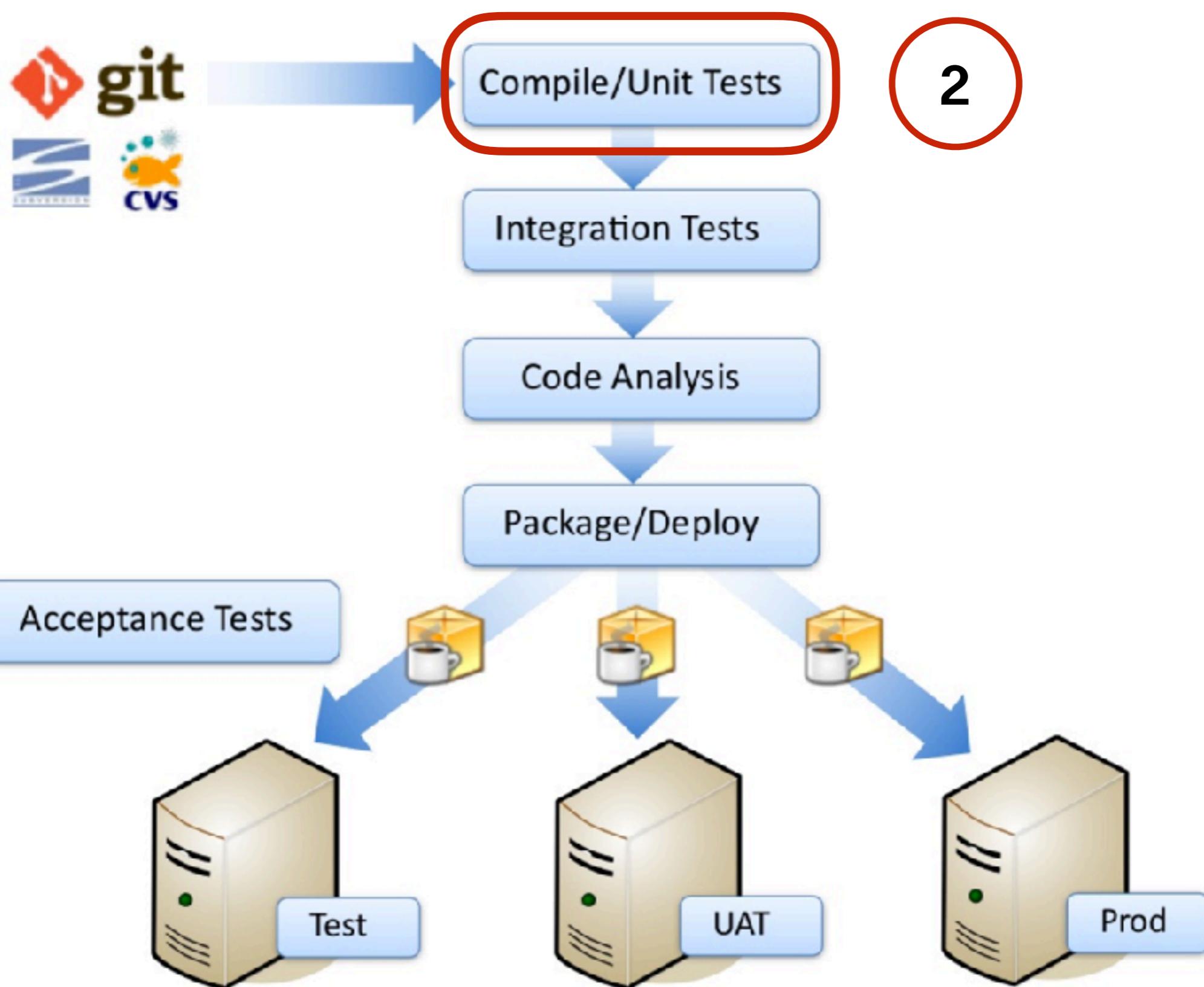


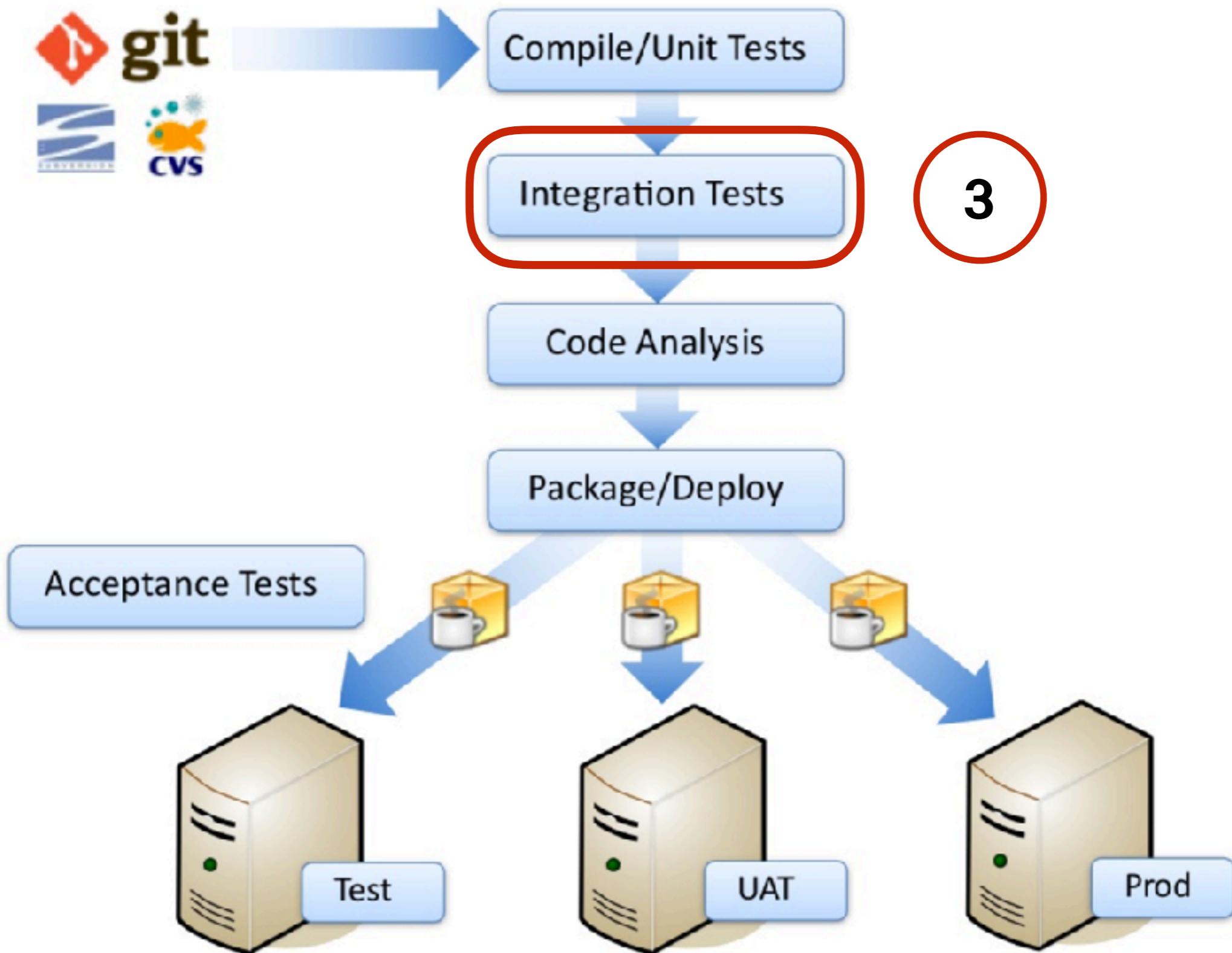
Let's start with Jenkins

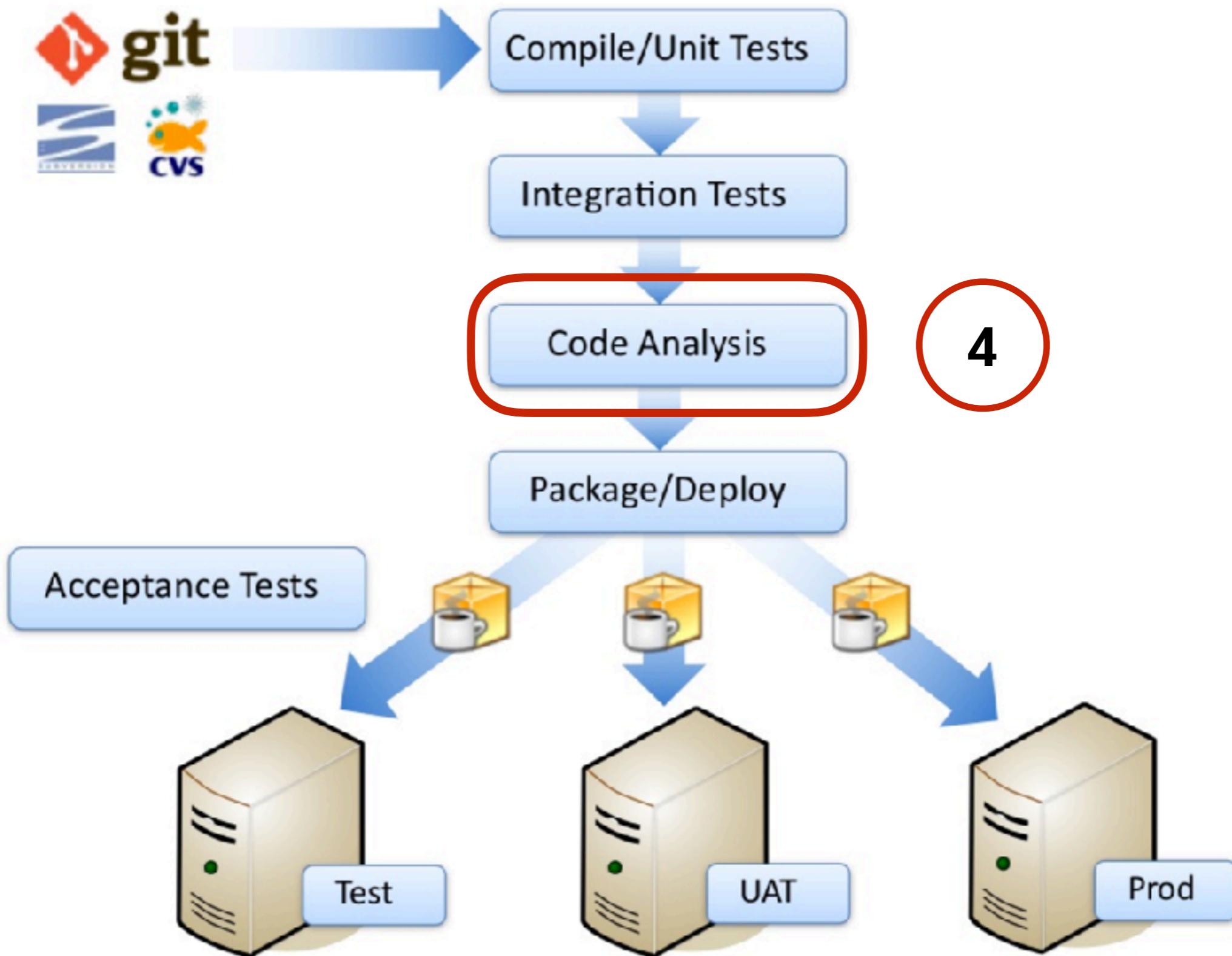


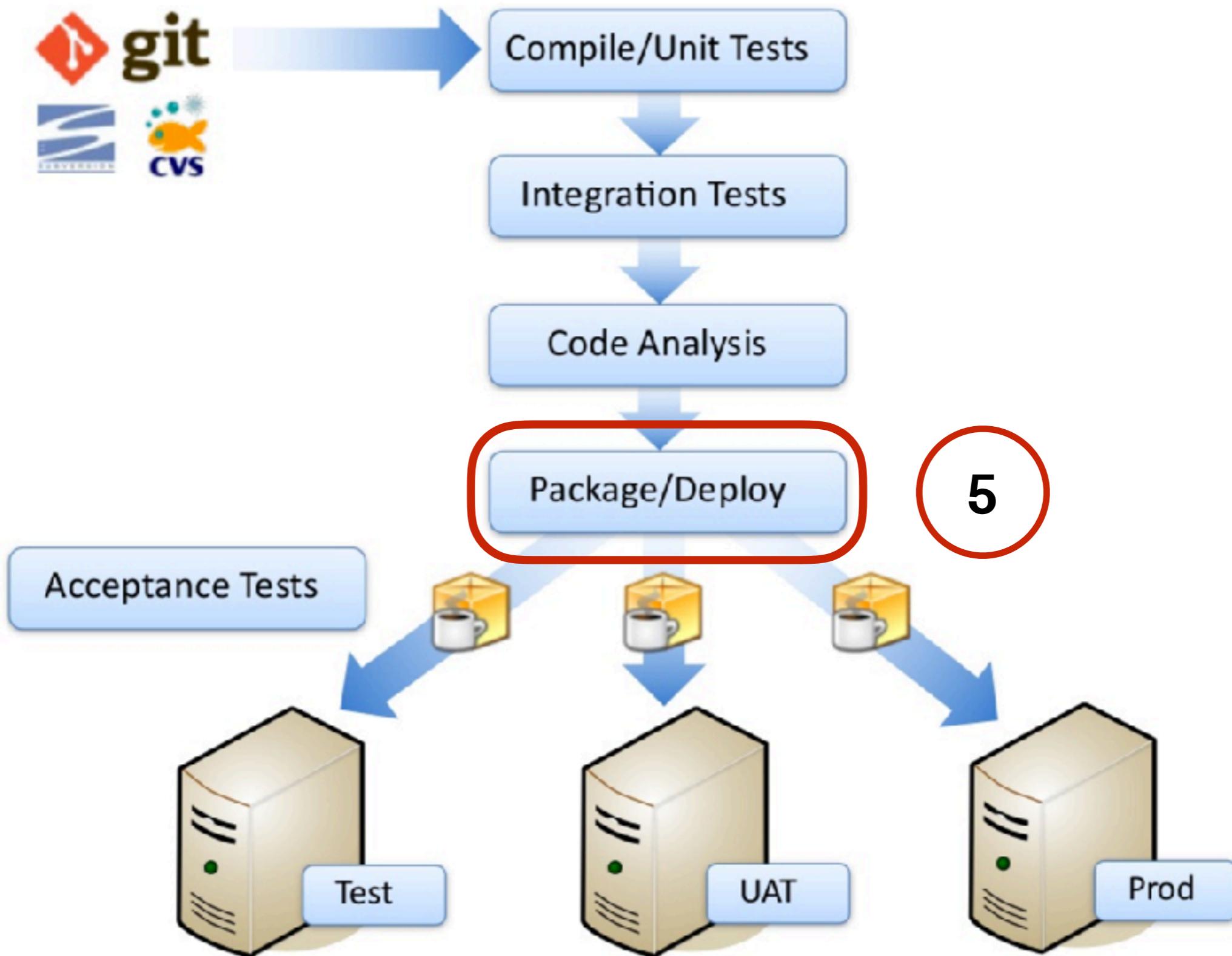
1

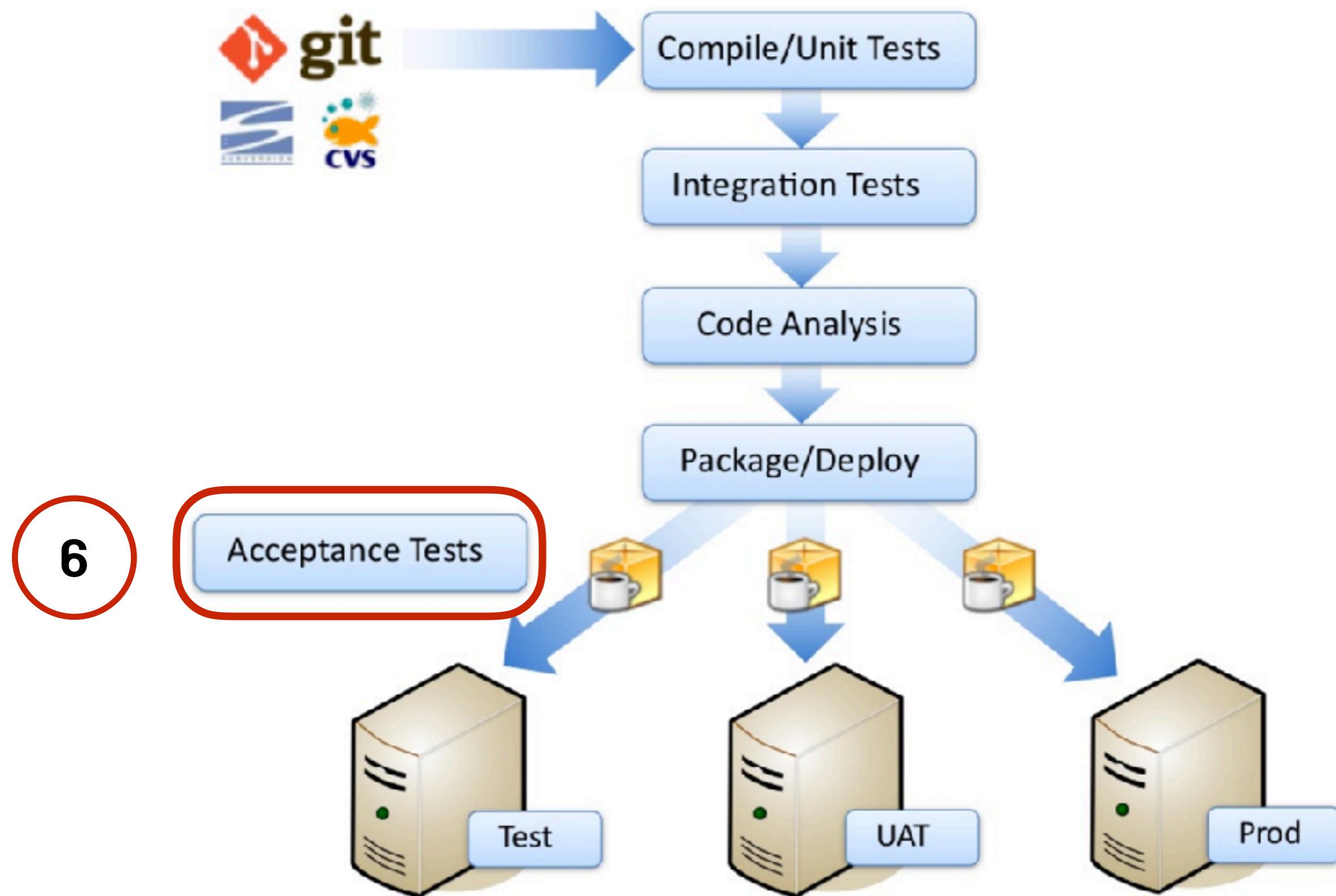


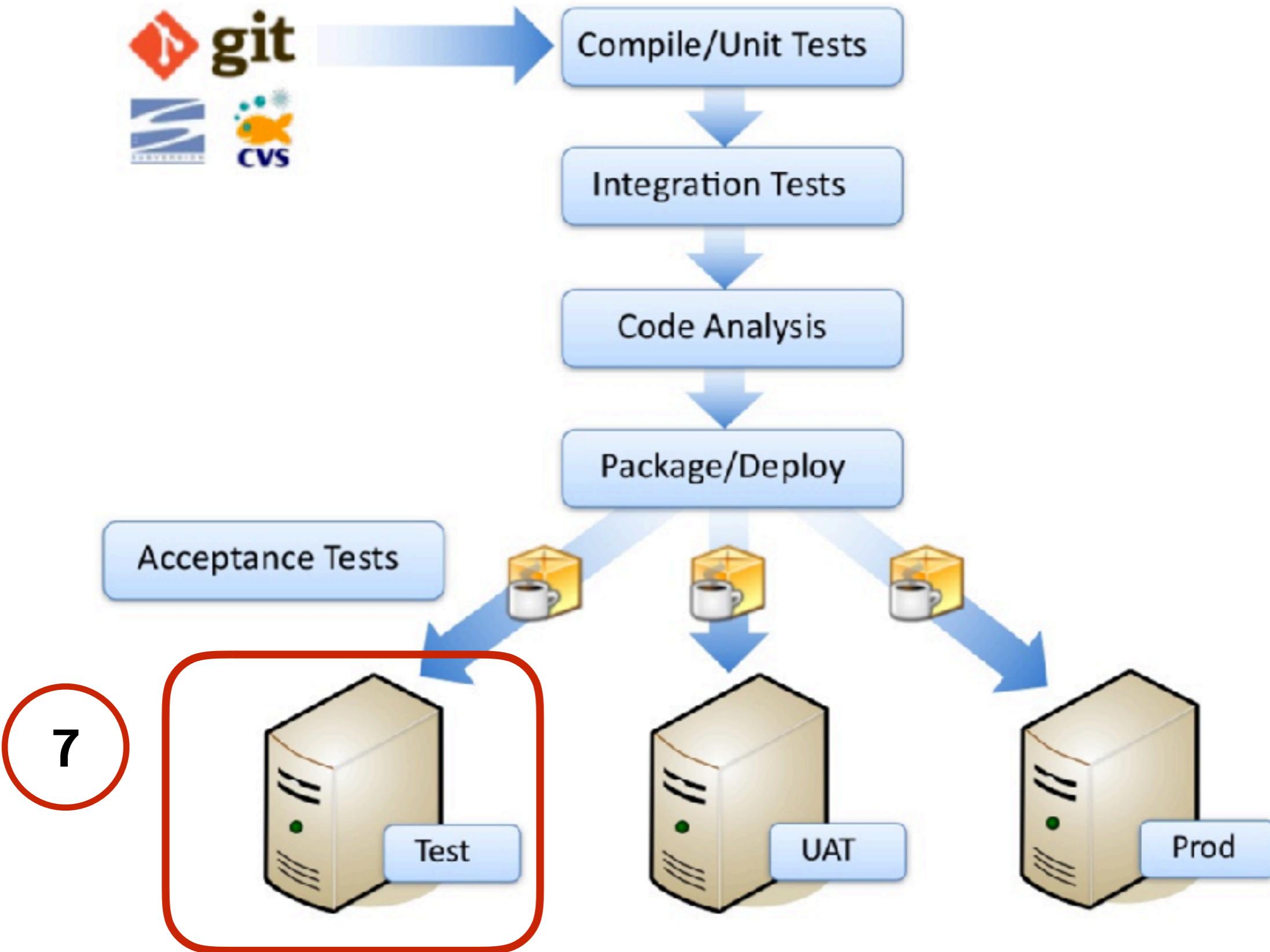


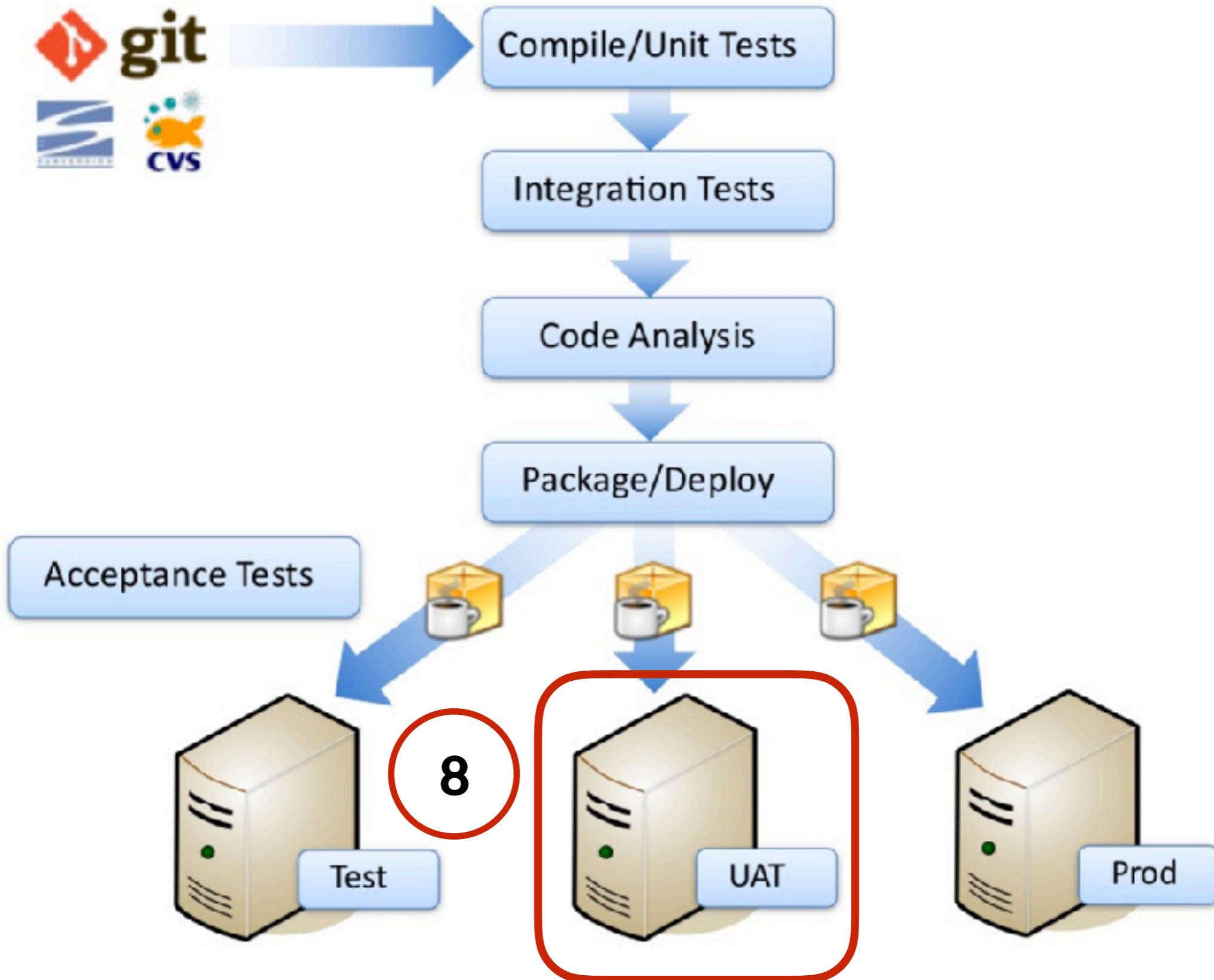


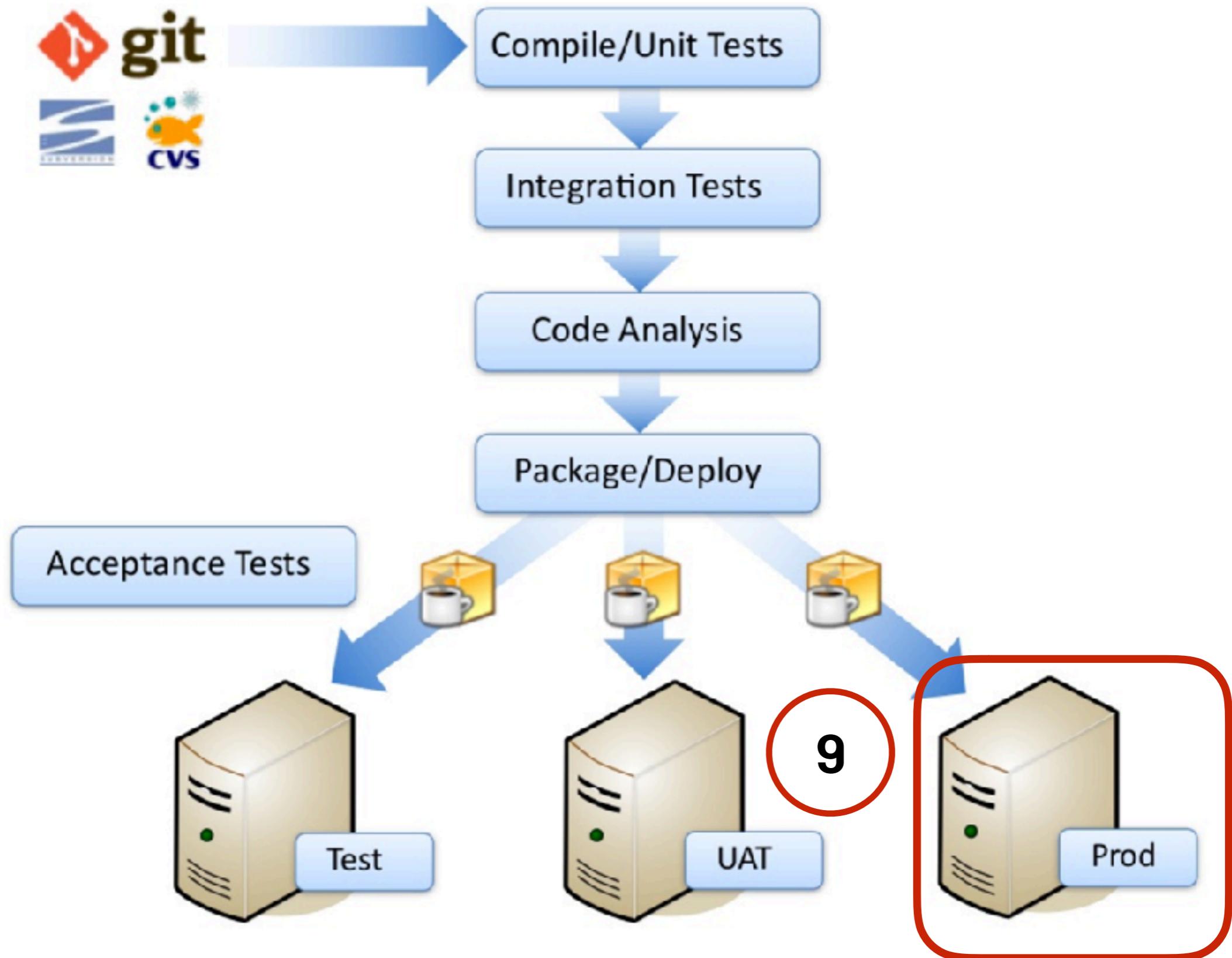












Workshop



บริษัท สยามชนาญกิจ จำกัด และเพื่อนพ้องน้องพี่