CI and CD

Jenkins

https://www.facebook.com/somkiatspns

Somkiat Puisungnoen

Somkiat | Home

Somkiat Puisungnoen

Update Info 2 | View Activity Log 10+ | ...

Timeline | About | Friends 2,604 | Photos | More

สยามชำนาญกิจ

SIAM CHAMNAN KIT

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

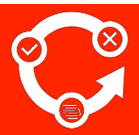
Jenkins

Agenda

Concept of **C**ontinuous **I**ntegration
Concept of **C**ontinuous **D**elivery
Build pipeline

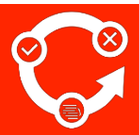
Agenda

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


Source Code
Version Control System
Development Team
Source Code Build
Publish Reports
Staric Code Analysis
Run Automates Functional Tests
Continuous Integration
Run Automated Unit Tests
Provision and Deploy to Test Environment
Code Coverage Analysis
Set-up Test 'Fixtures'
Built Artifact
1
2
3
4
5
6
7
8
9
10
11
12

Jenkins

Bamboo

TeamCity

go™

Visual Studio Team Foundation Server

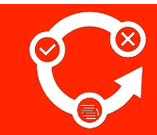
Hudson

Travis

wercker

circleci

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



CI is a practice

Discipline to **integrate frequently**

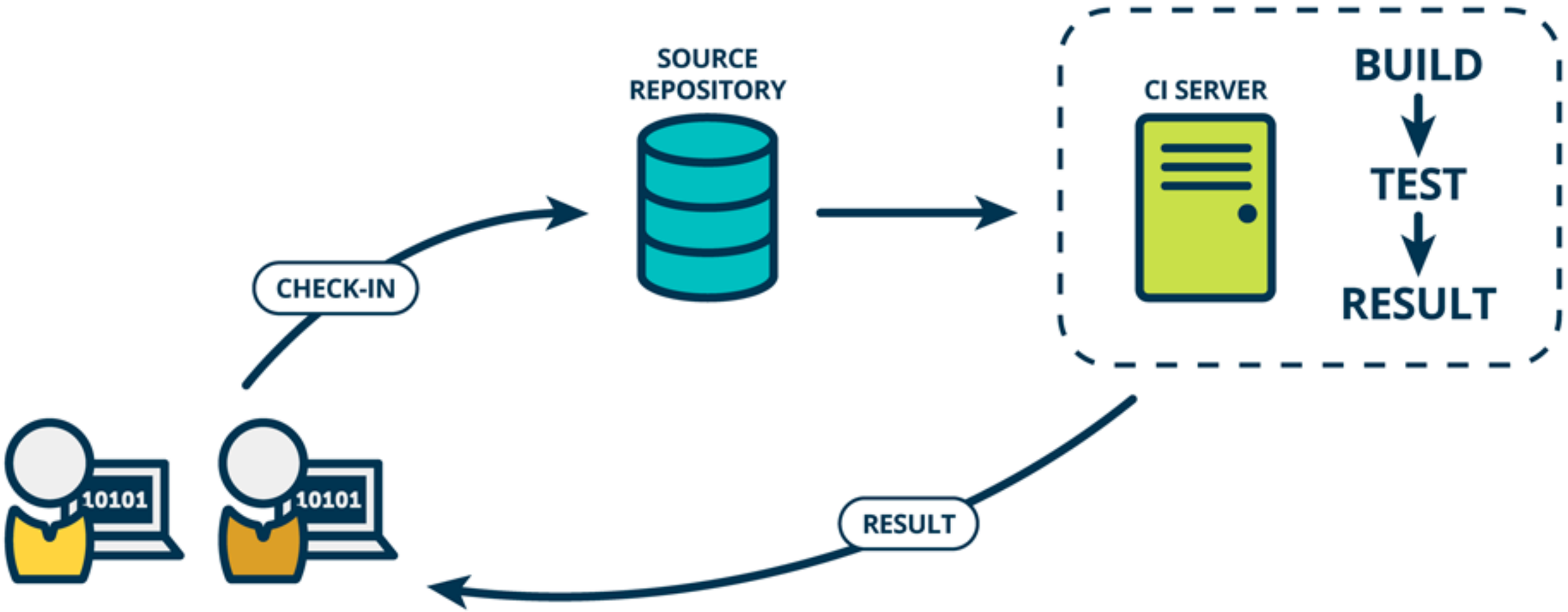
CI is a practice

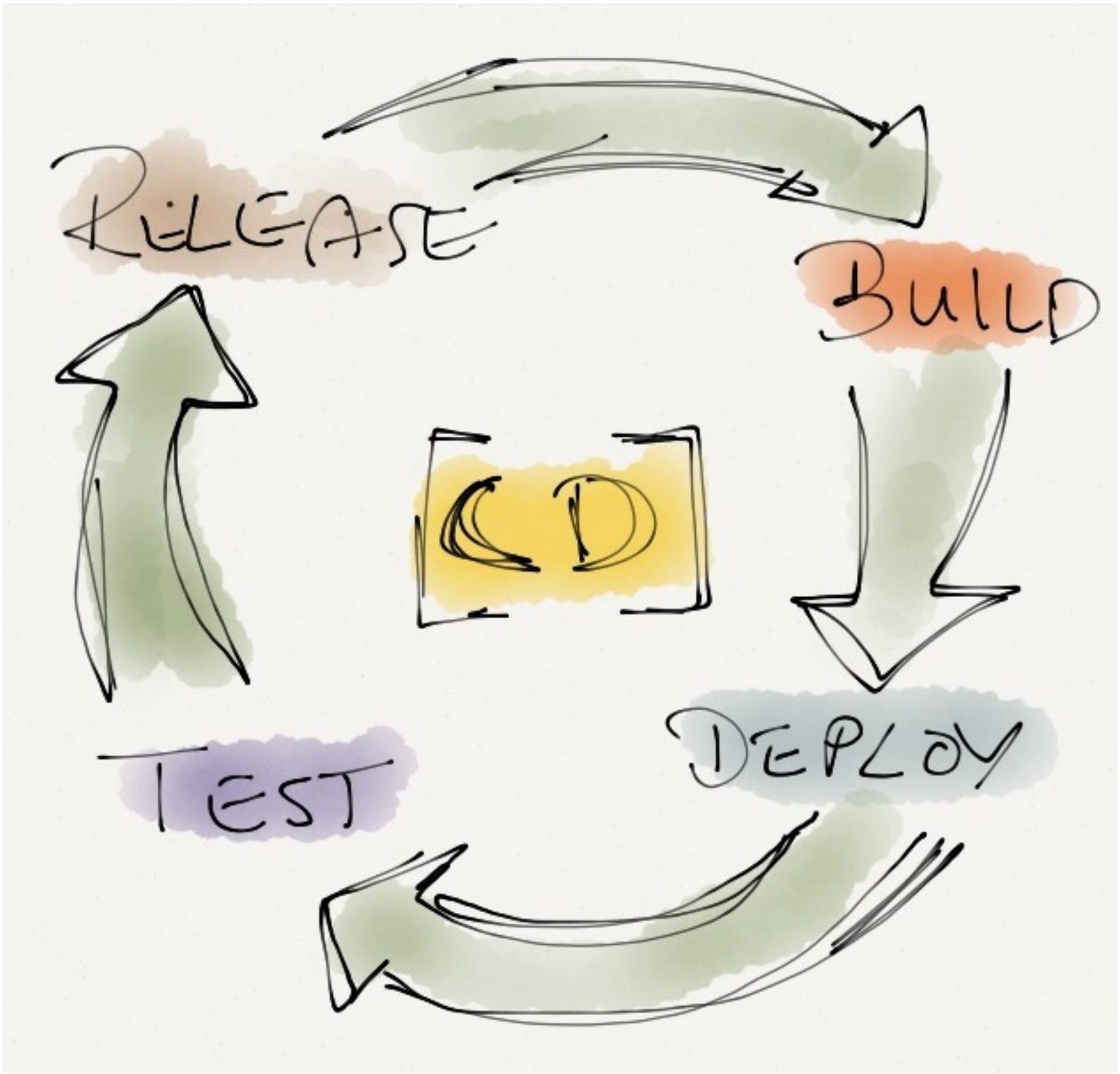
Strive to make **small change**

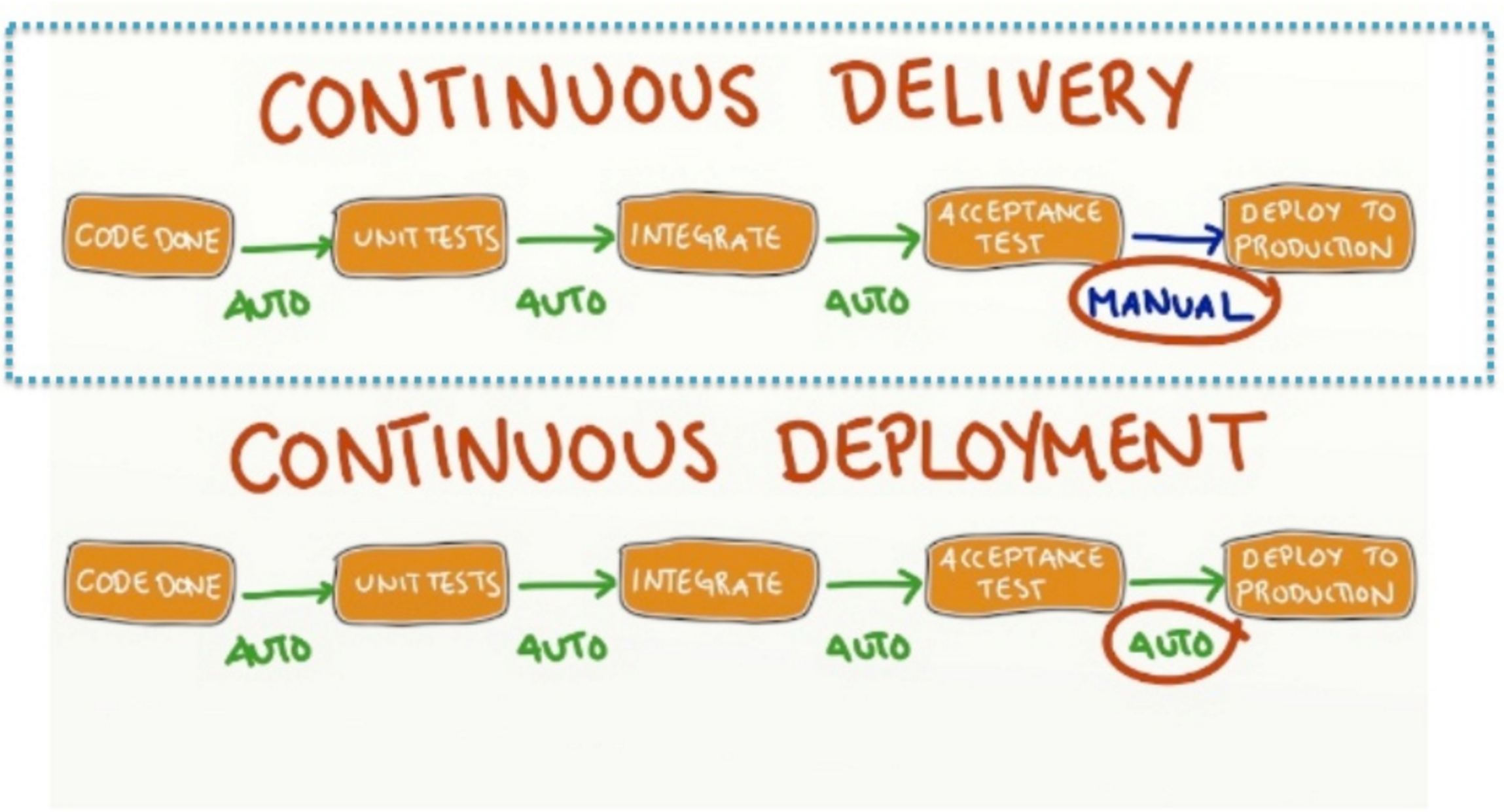
CI is a practice

Strive for **fast feedback**

Continuous Integration

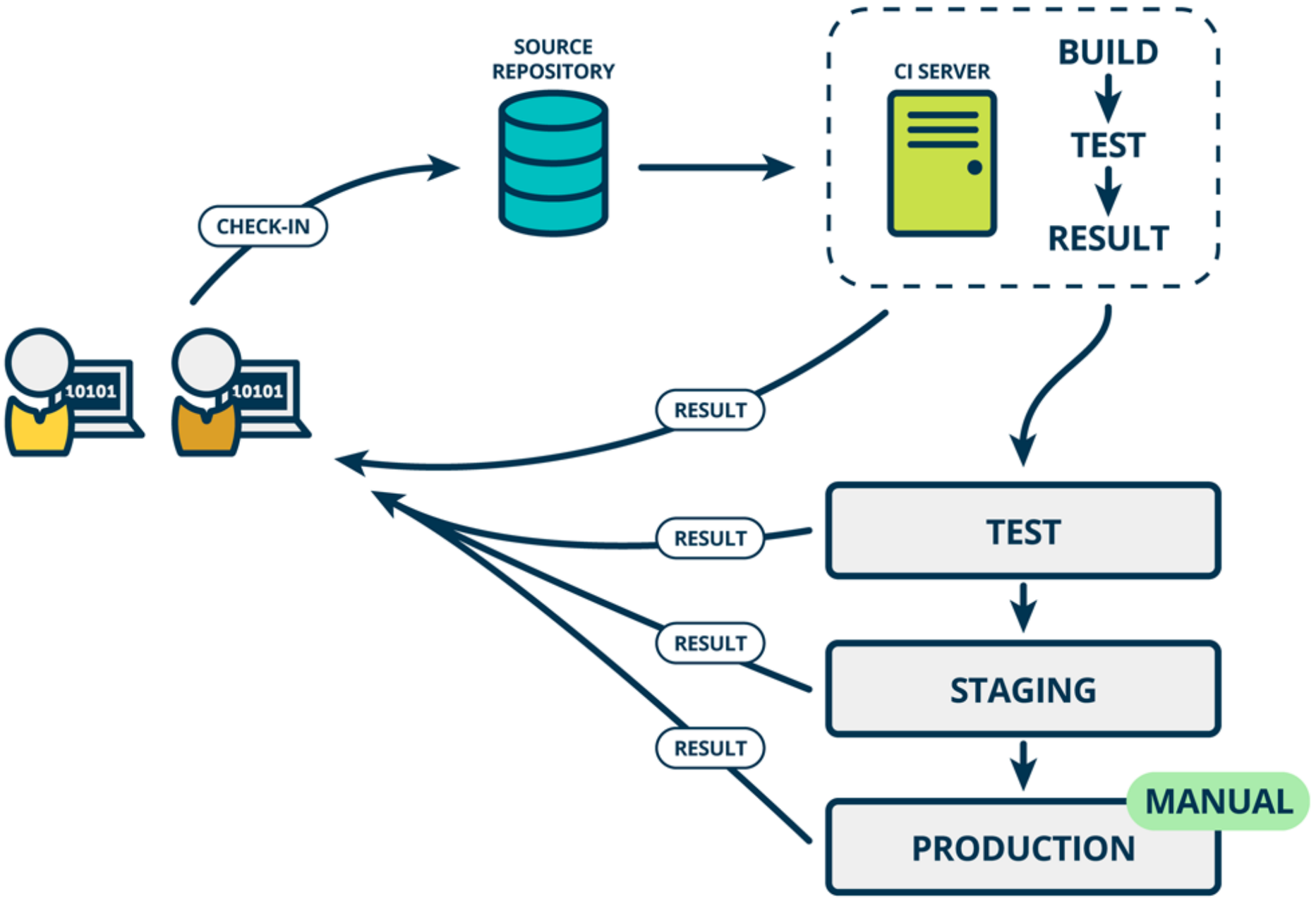



RELEASE
BUILD
TEST
DEPLOY

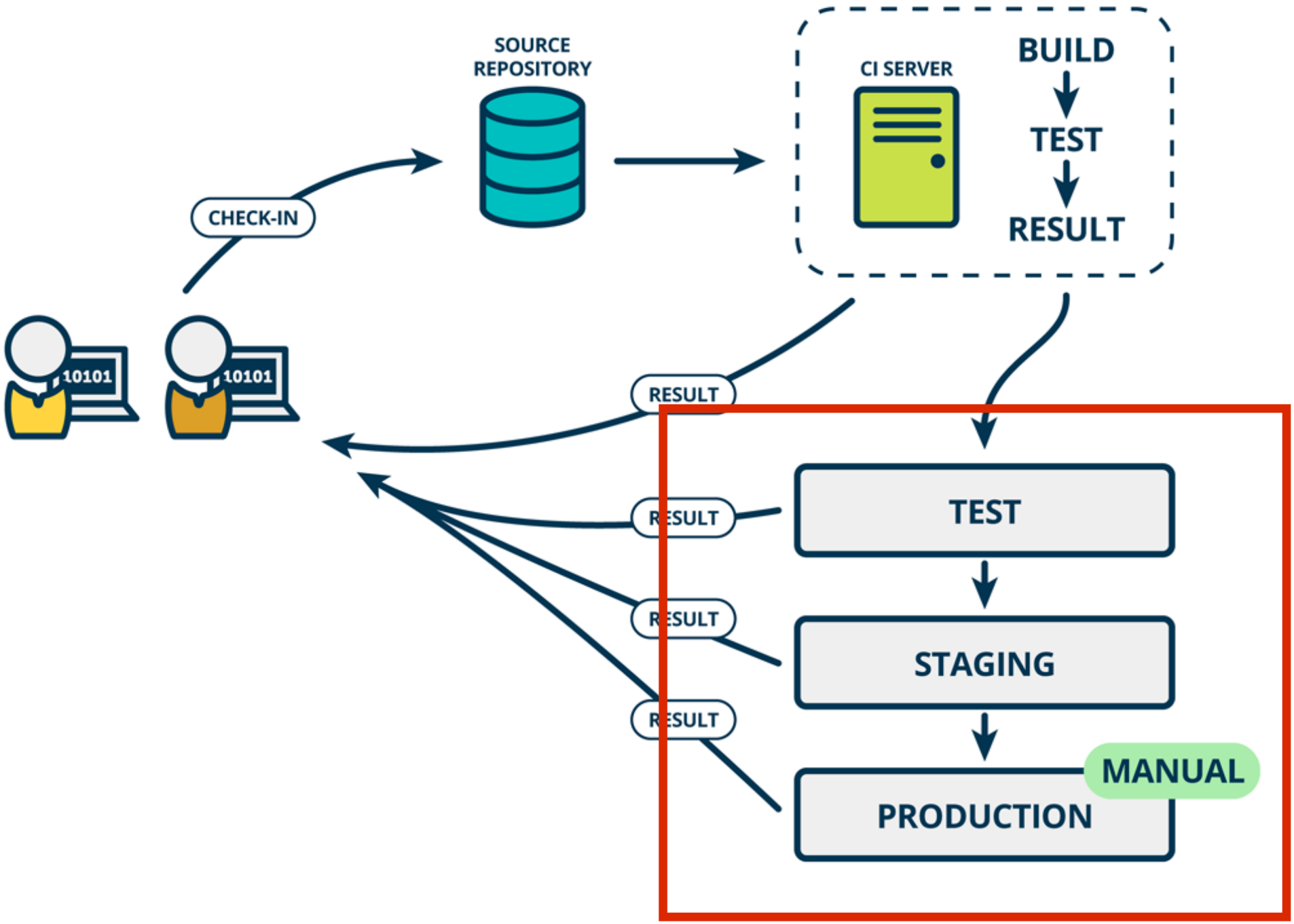


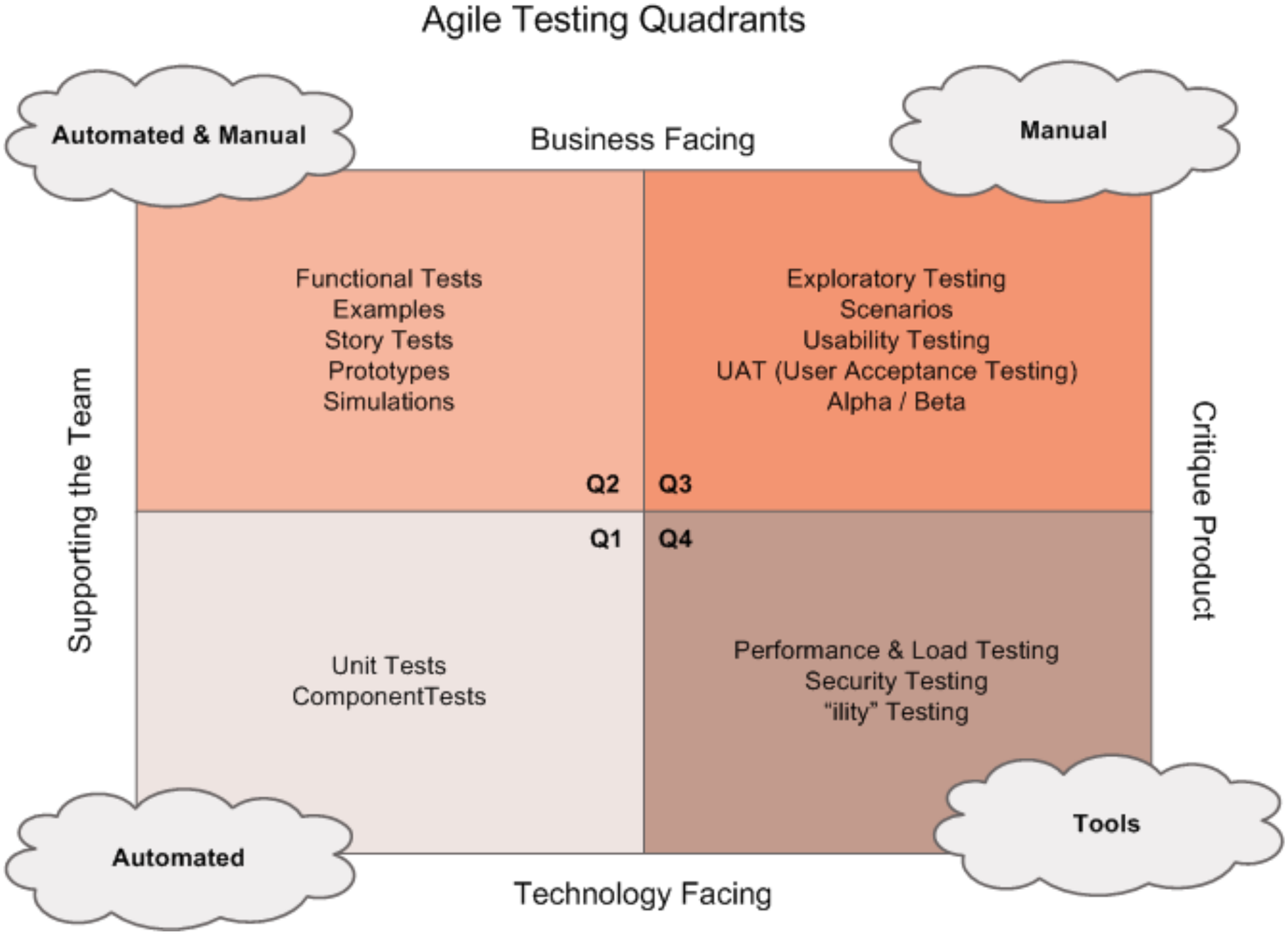
http://blog.crisp.se/2013/02/05/yassalsundman/continuous-delivery-vs-continuous-deployment

Continuous Delivery



Rising of DevOps




Agile Testing Quadrants
Automated & Manual
Business Facing
Manual
Functional Tests
Examples
Story Tests
Prototypes
Simulations
Exploratory Testing
Scenarios
Usability Testing
UAT (User Acceptance Testing)
Alpha / Beta
Q2
Q3
Supporting the Team
Critique Product
Q1
Q4
Unit Tests
ComponentTests
Performance & Load Testing
Security Testing
“ility” Testing
Automated
Tools
Technology Facing


Are you too busy to improve?
No thanks!
We are too busy
Created by Håkan Forss @hakanforss http://hakanforss.wordpress.com
BY NC

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**

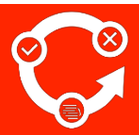
7. Testing in a clone of the **production environment**

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

9. **Everyone can see** what’s happening

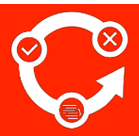
10. Automate **Deployment**

Good habits with CI



Developer should work in the private workspace

Rebase frequently
from the mainline



Check-in/push frequently

Frequent **build**

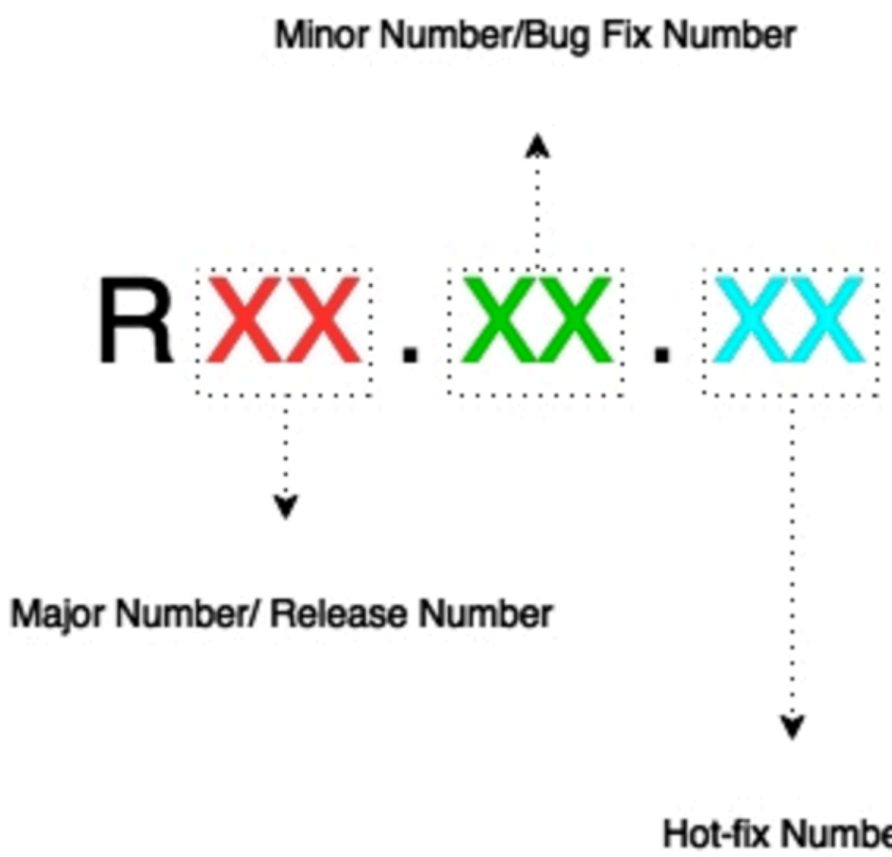
Automate the **testing** as much as possible

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

Automate the **deployment**

Have a **labeling strategy**
fore releases

Instant **notification**


Minor Number/Bug Fix Number
RXX.XX.XX
Major Number/ Release Number
Hot-fix Number

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.

Microsoft cloudbees OSL OPEN SOURCE LAB

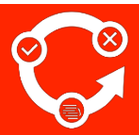
rackspace redhat

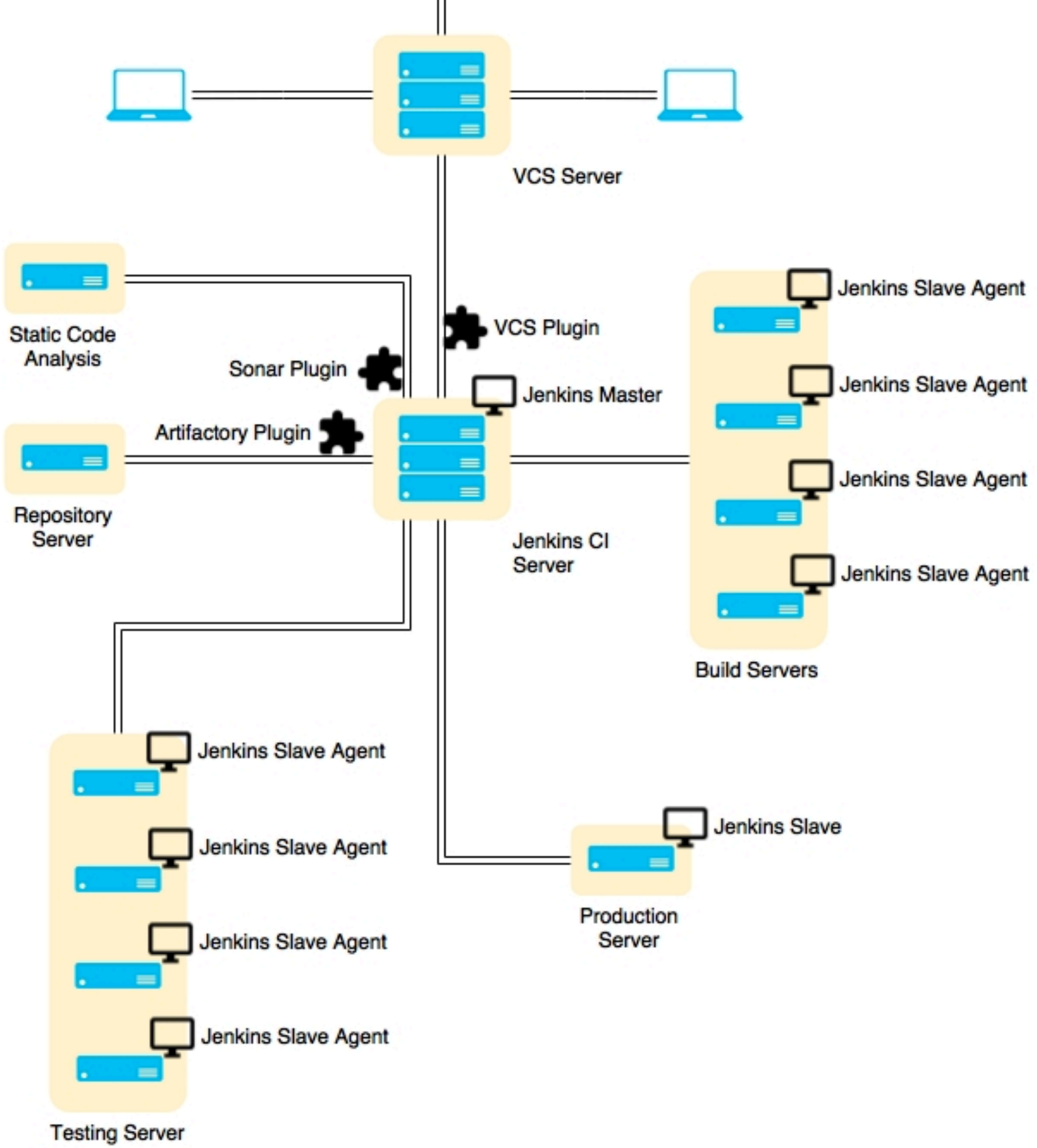
We thank the following organizations for their support of the Jenkins project through free and/or open source licensing programs.

Atlassian Datadog JFrog Mac Cloud PagerDuty XMission

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

Jenkins => Centralize CI Server




VCS Server
Static Code Analysis
Sonar Plugin
VCS Plugin
Jenkins Master
Artifactory Plugin
Repository Server
Jenkins CI Server
Jenkins Slave Agent
Jenkins Slave Agent
Jenkins Slave Agent
Jenkins Slave Agent
Build Servers
Jenkins Slave Agent
Jenkins Slave Agent
Jenkins Slave Agent
Jenkins Slave Agent
Testing Server
Jenkins Slave
Production 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

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.

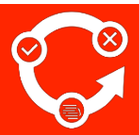
Changelog | Past Releases

Download Jenkins 2.65 for:

- Arch Linux
- Docker
- FreeBSD
- Gentoo

Start Jenkins

$java -jar jenkins.war

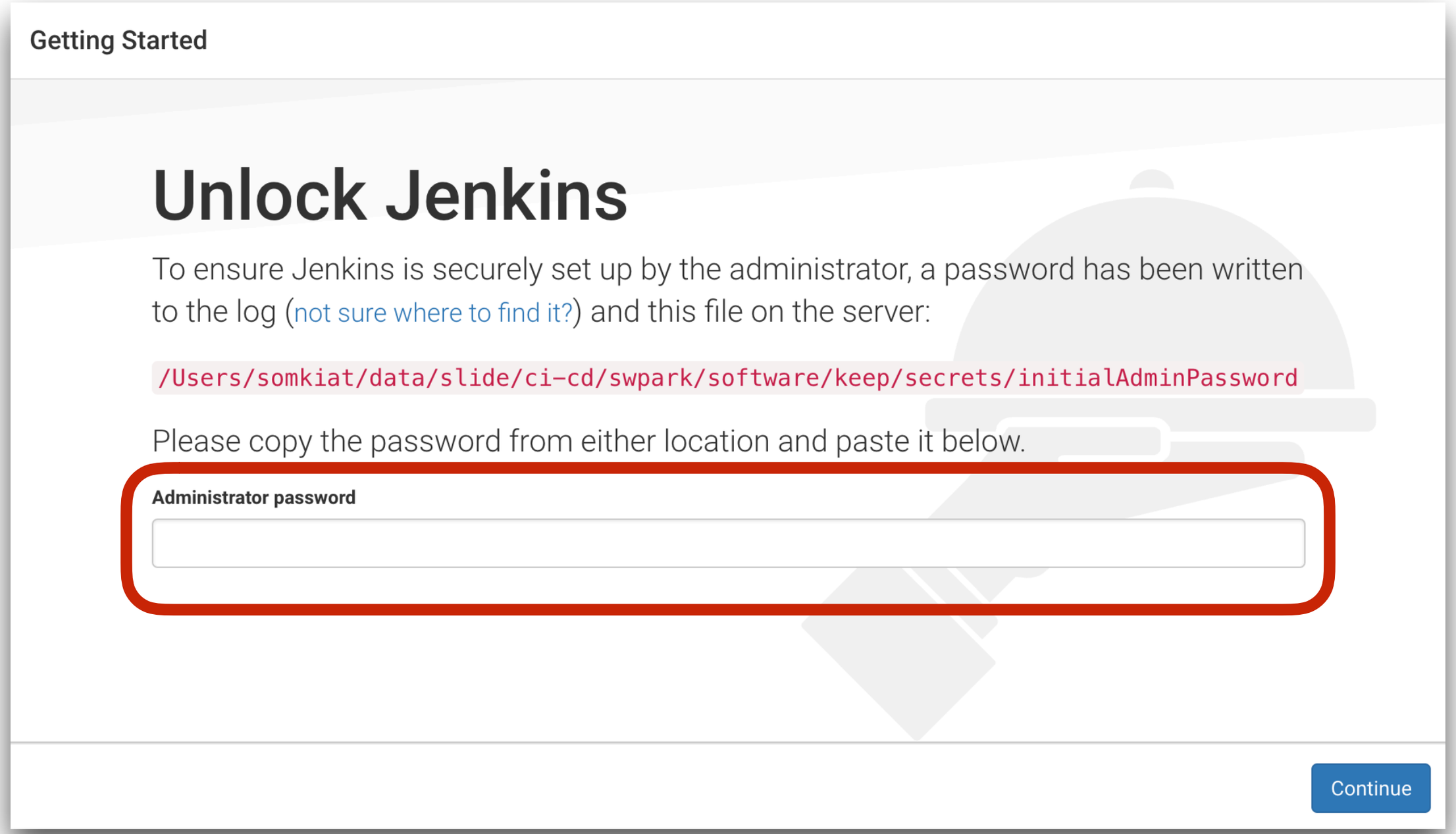


Change Port of Jenkins (8080)

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

Open in browser

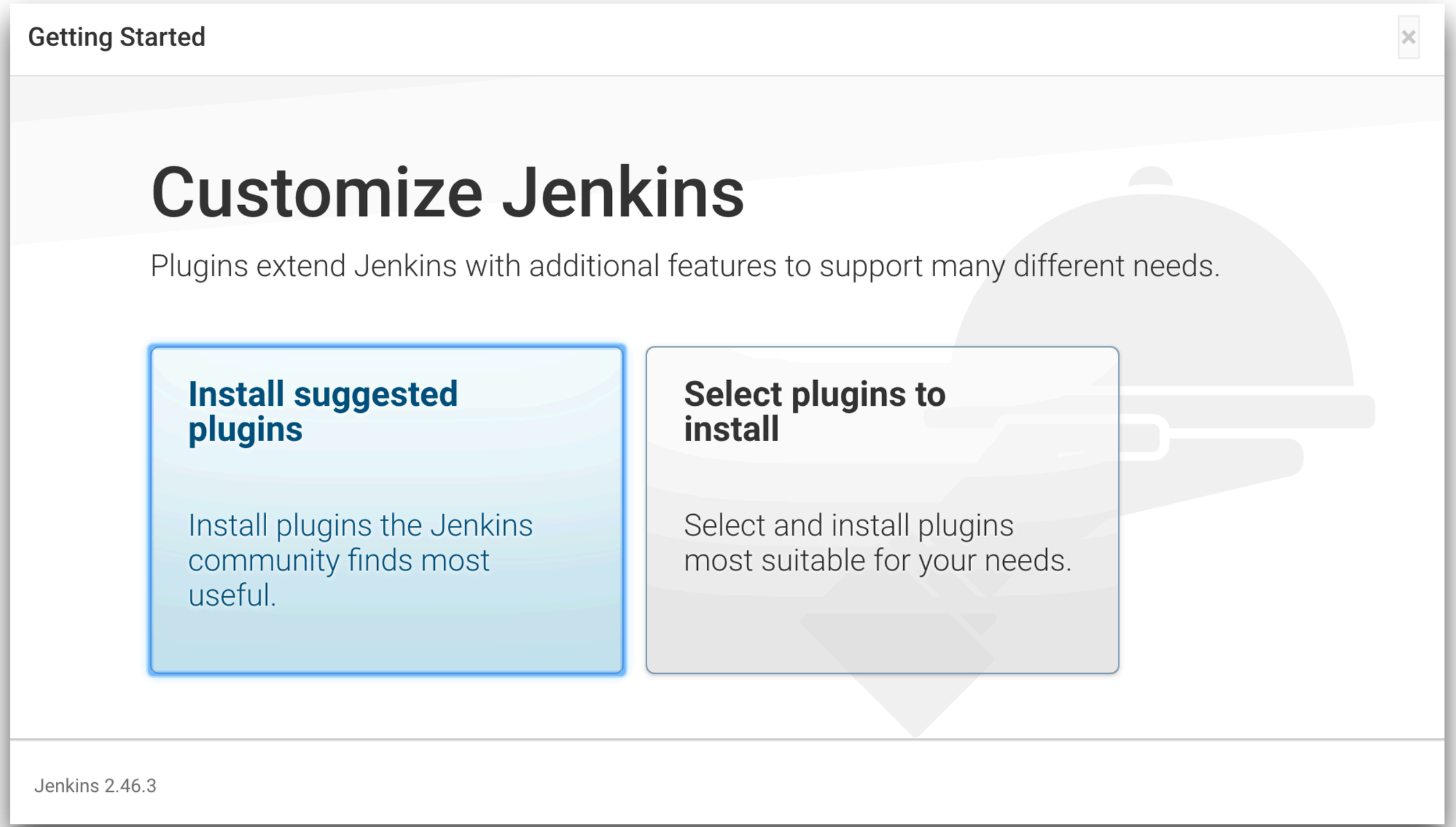
http://localhost:8080



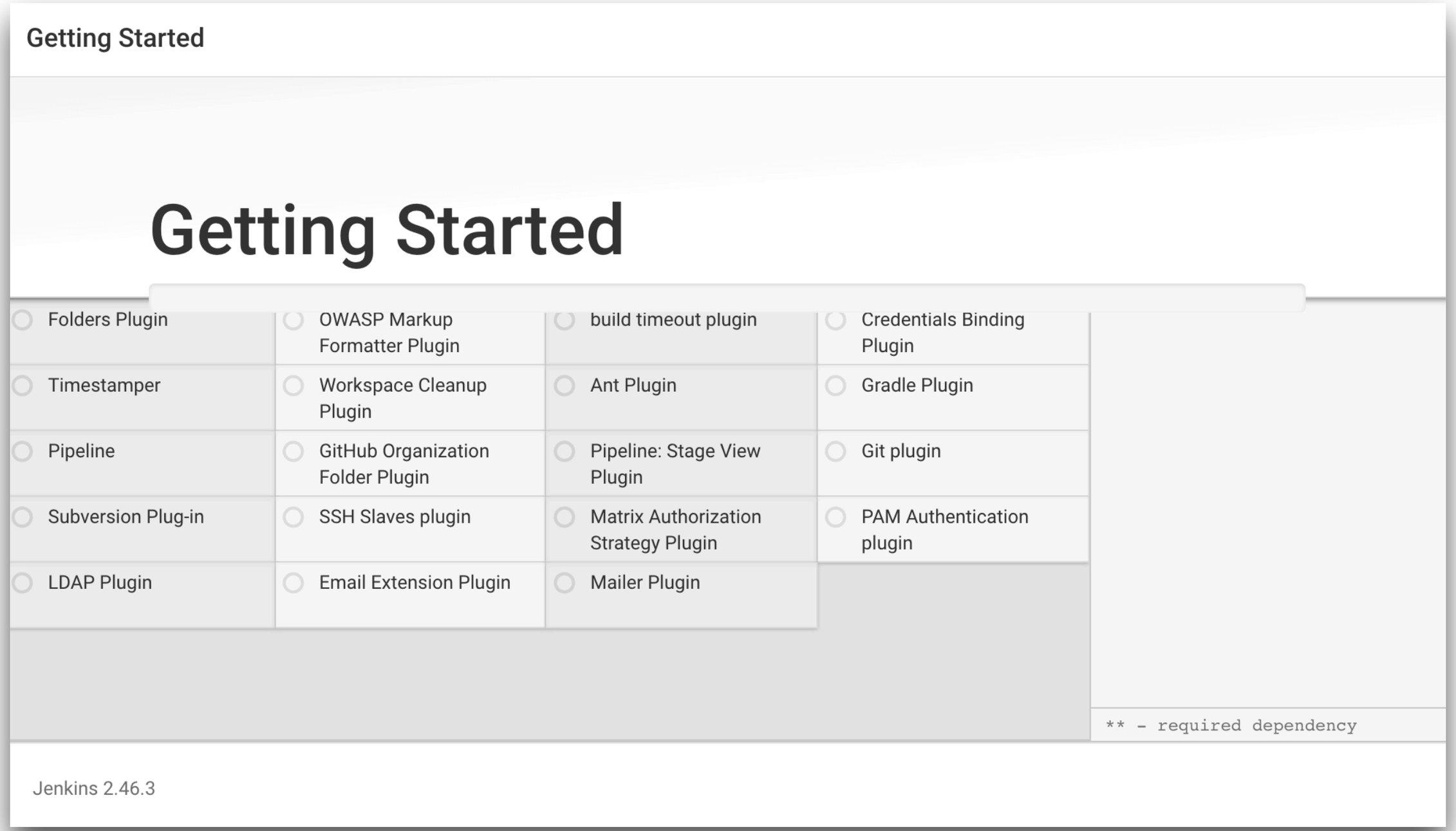
Copy password from console

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

Jenkins initial setup is required. An admin user has been creat
.
Please use the following password to proceed to installation:

a4b3a5231b8048419192d0c5afd3fce8

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

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

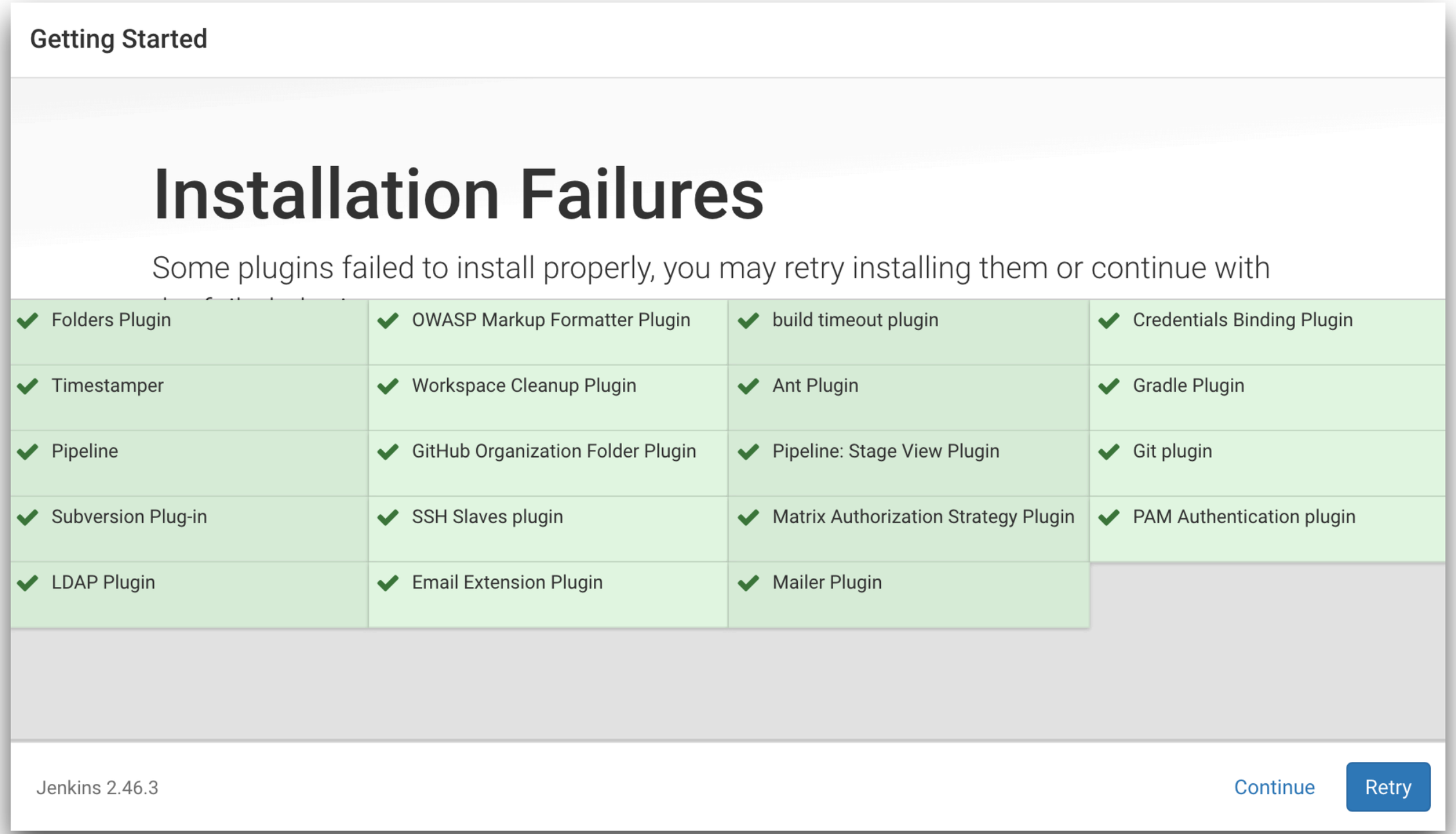
Custom your plug-ins



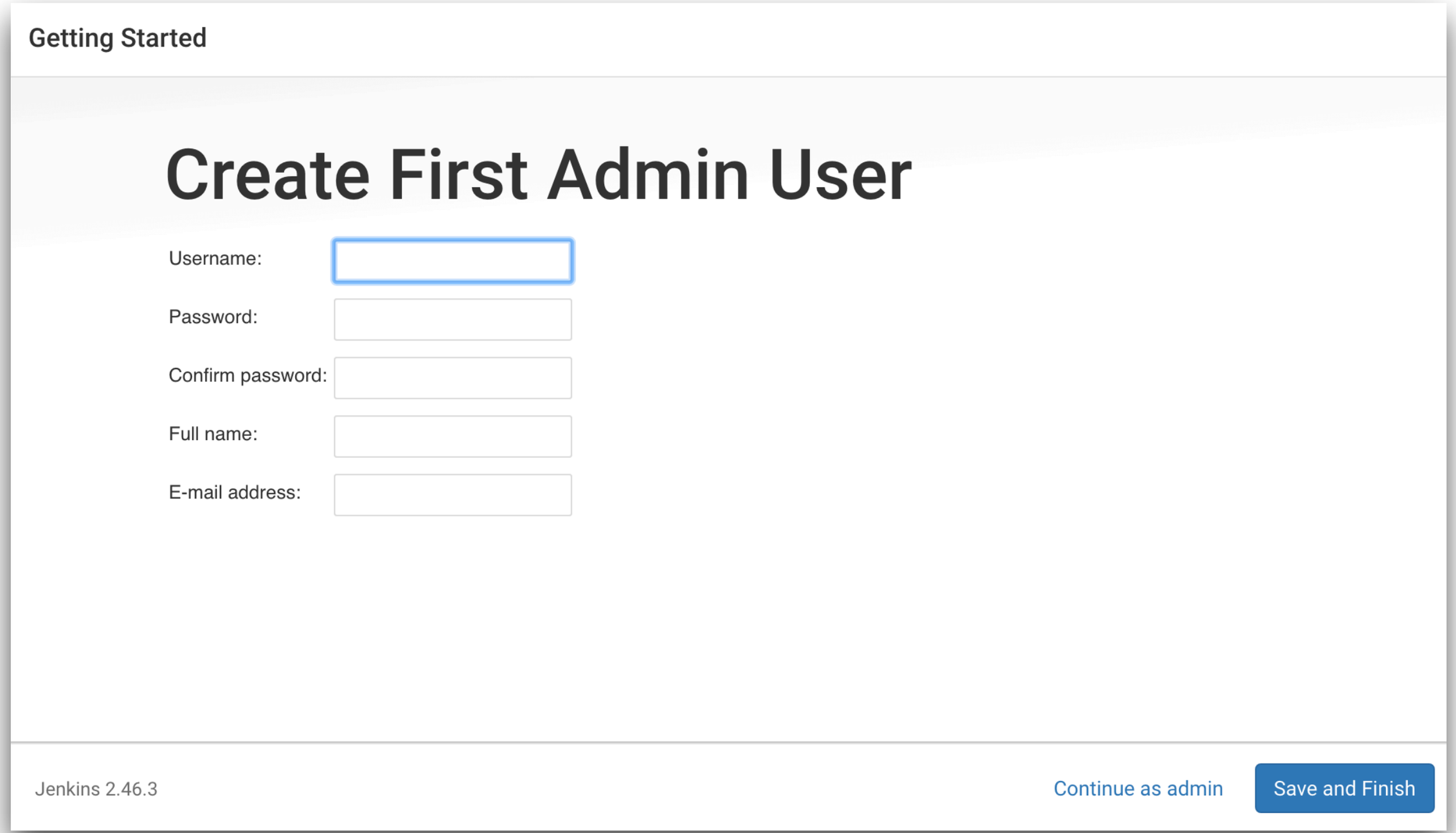
Waiting ...


Getting Started
Getting Started
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
** - required dependency
Jenkins 2.46.3

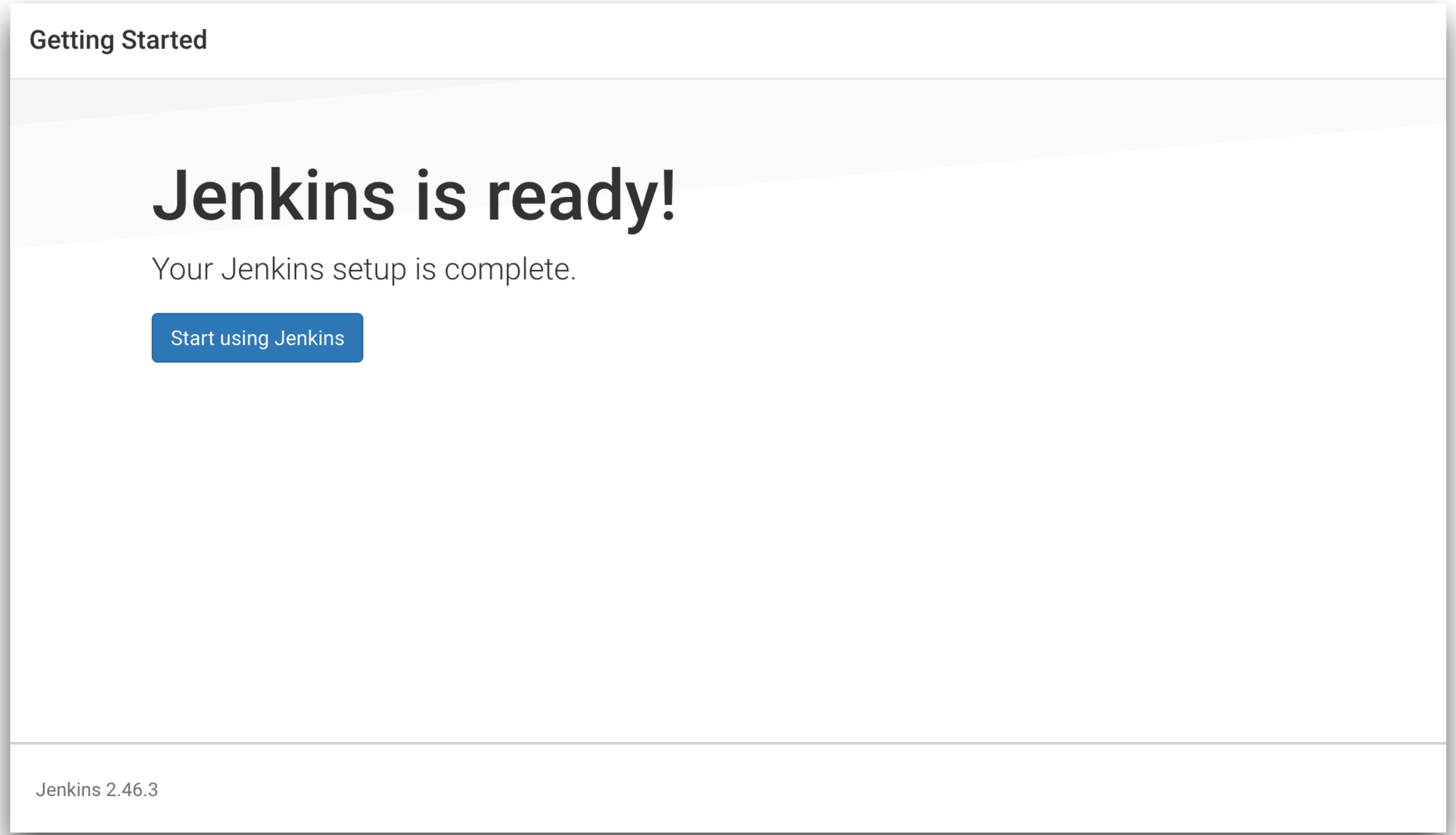
Success


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

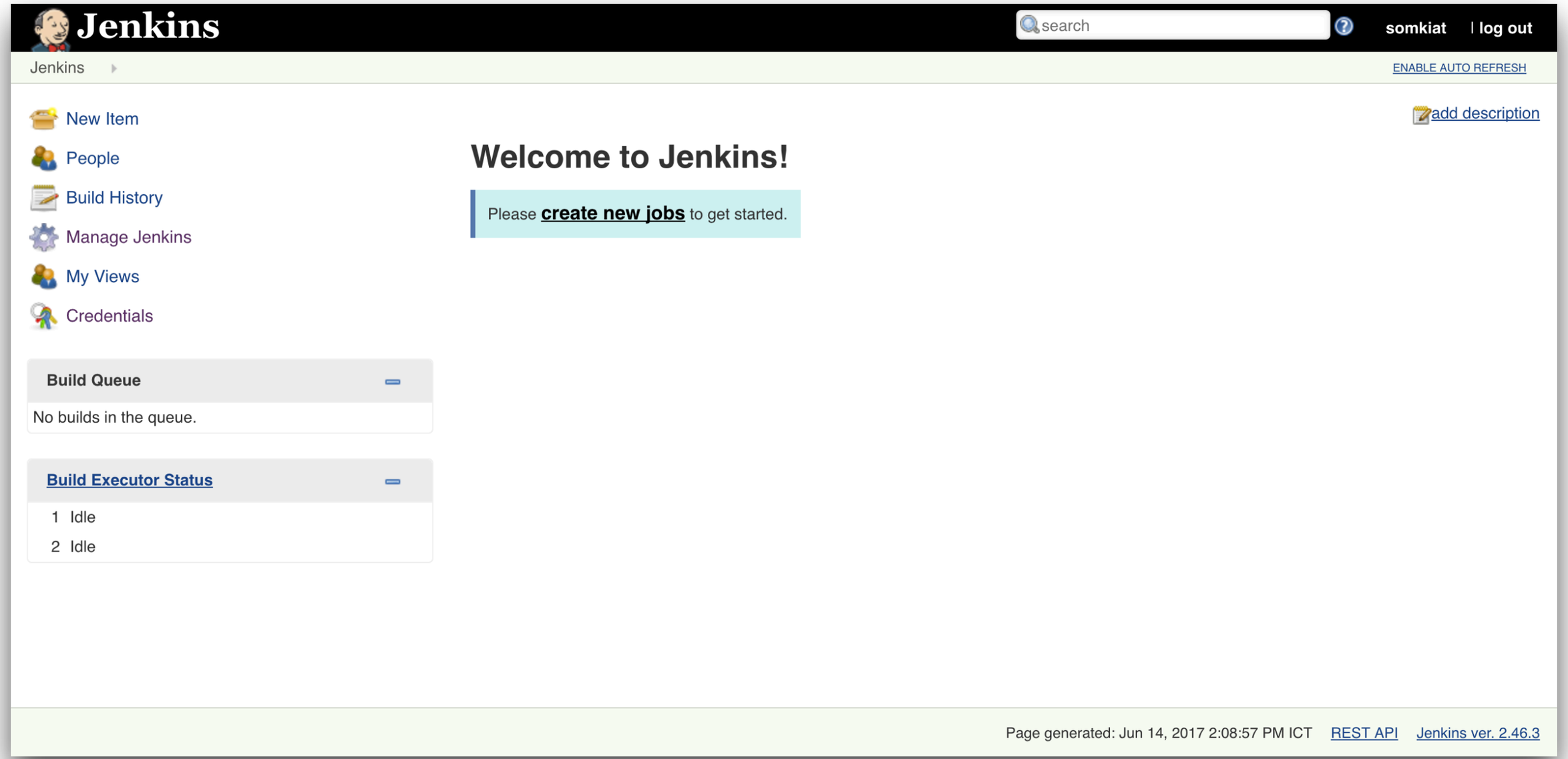
Create new user



Ready !!

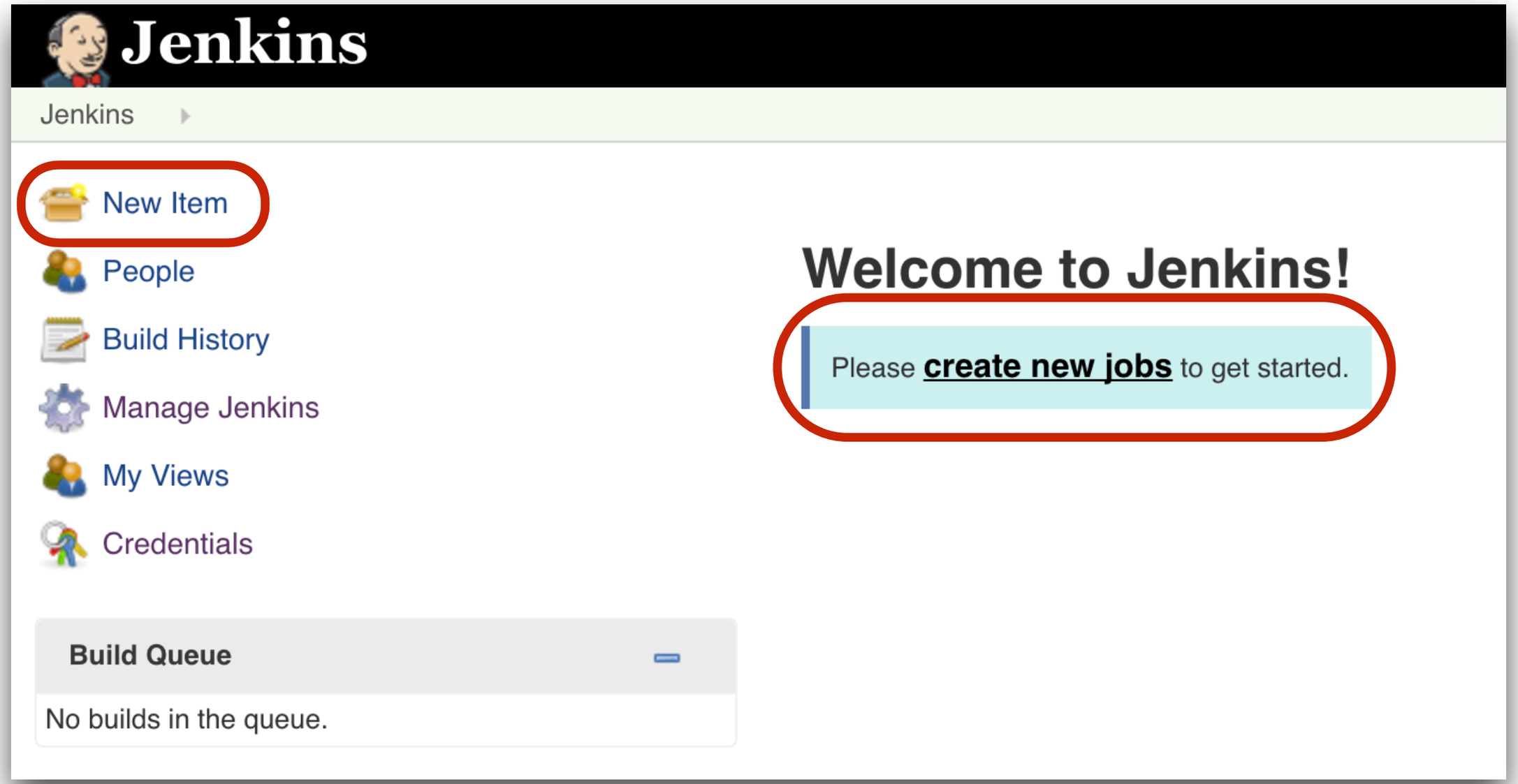

Getting Started
Jenkins is ready!
Your Jenkins setup is complete.
Start using Jenkins
Jenkins 2.46.3

Welcome to Jenkins

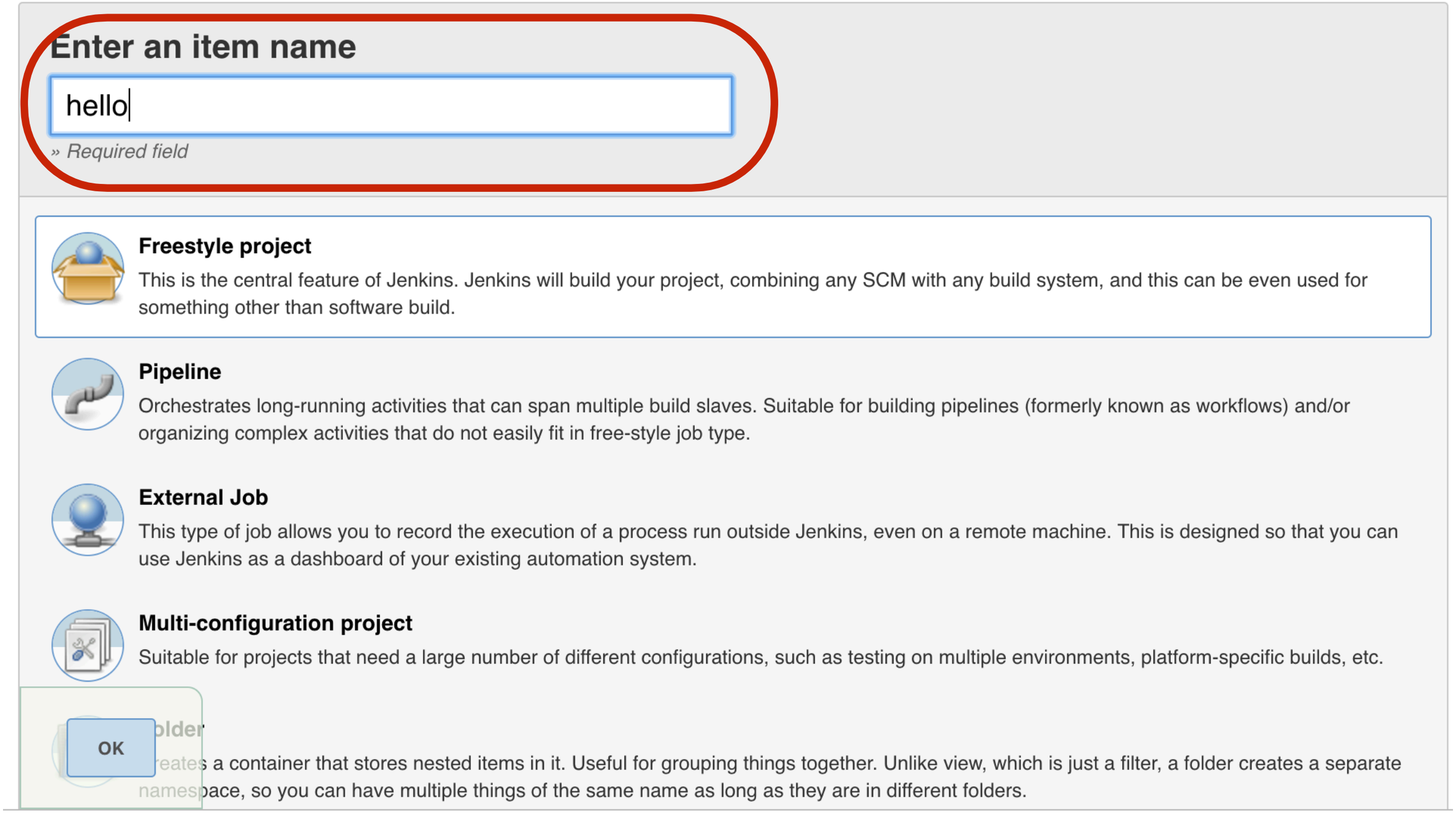


Create first Jenkins job

1. Create new job



2. Fill in name



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.

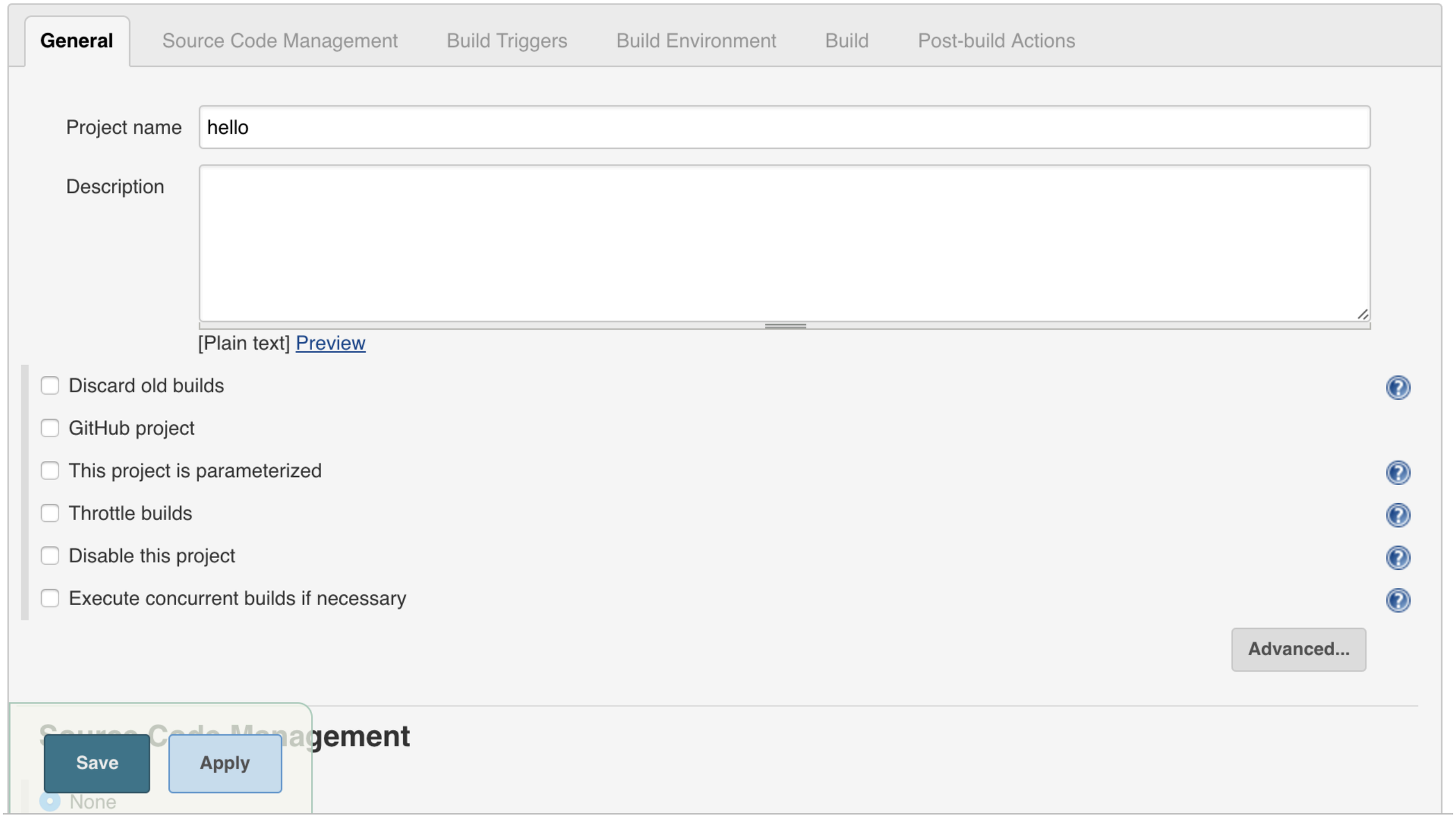
External Job
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.

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

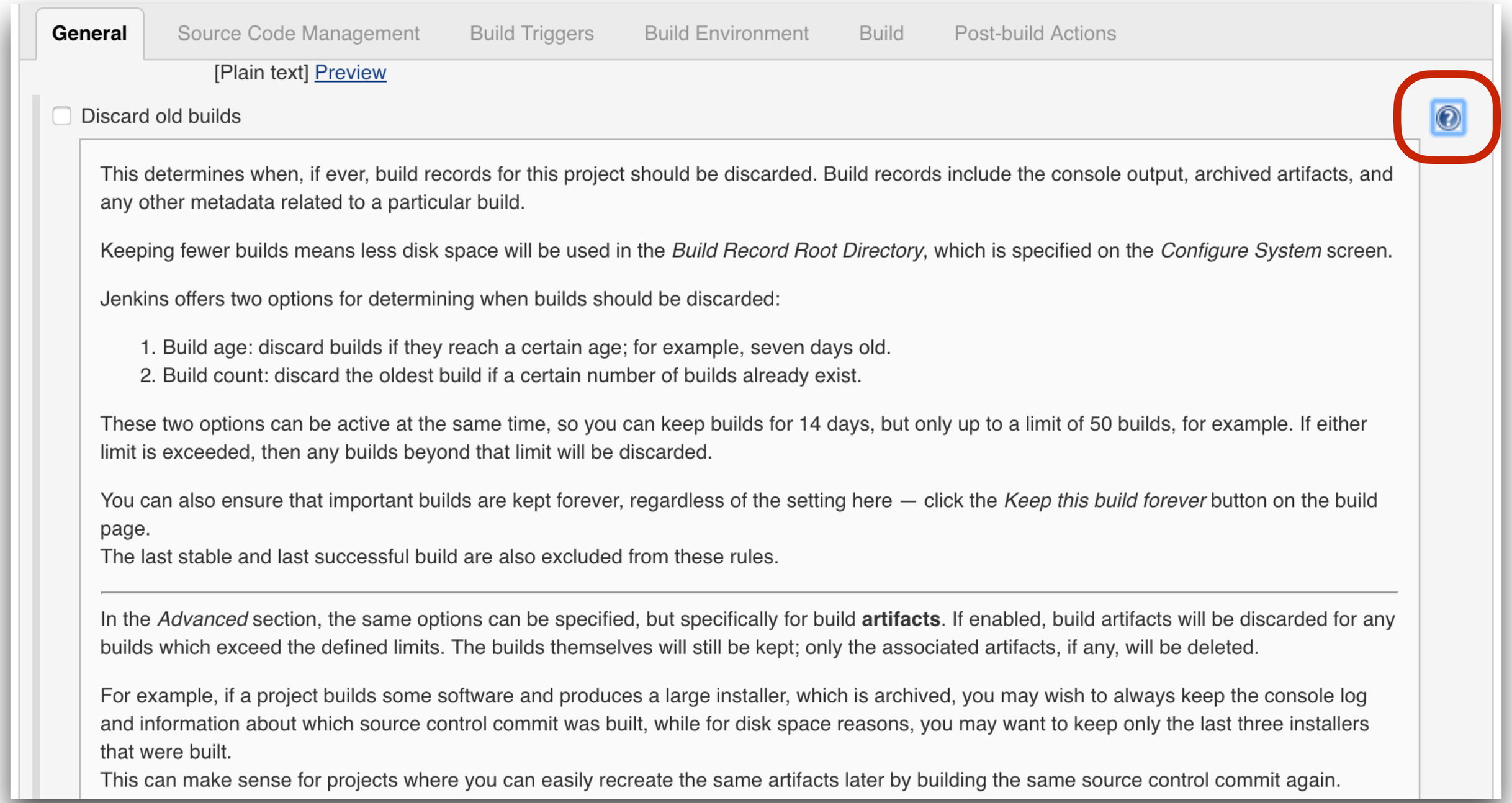
older
reates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

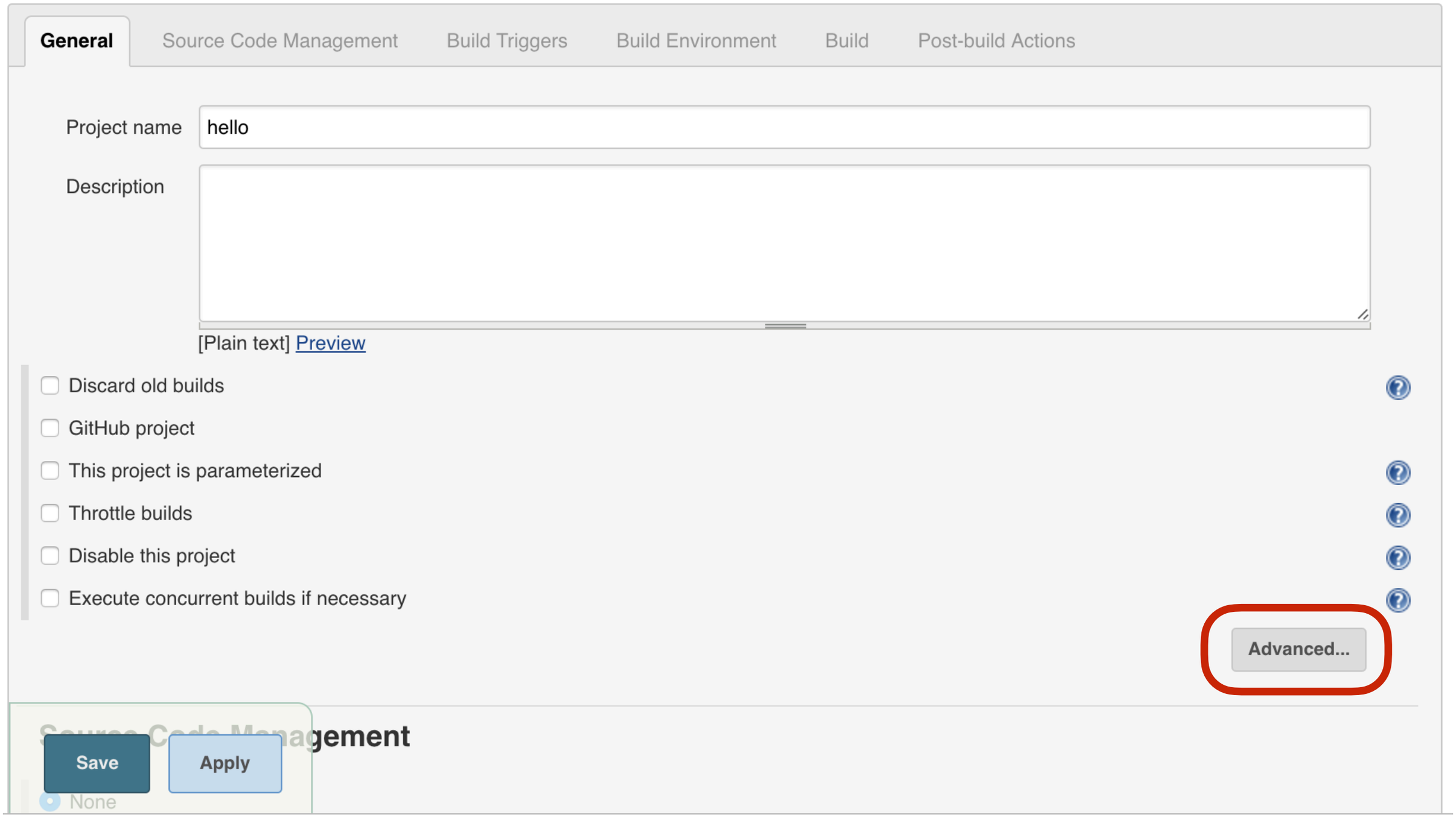
4. Config in General



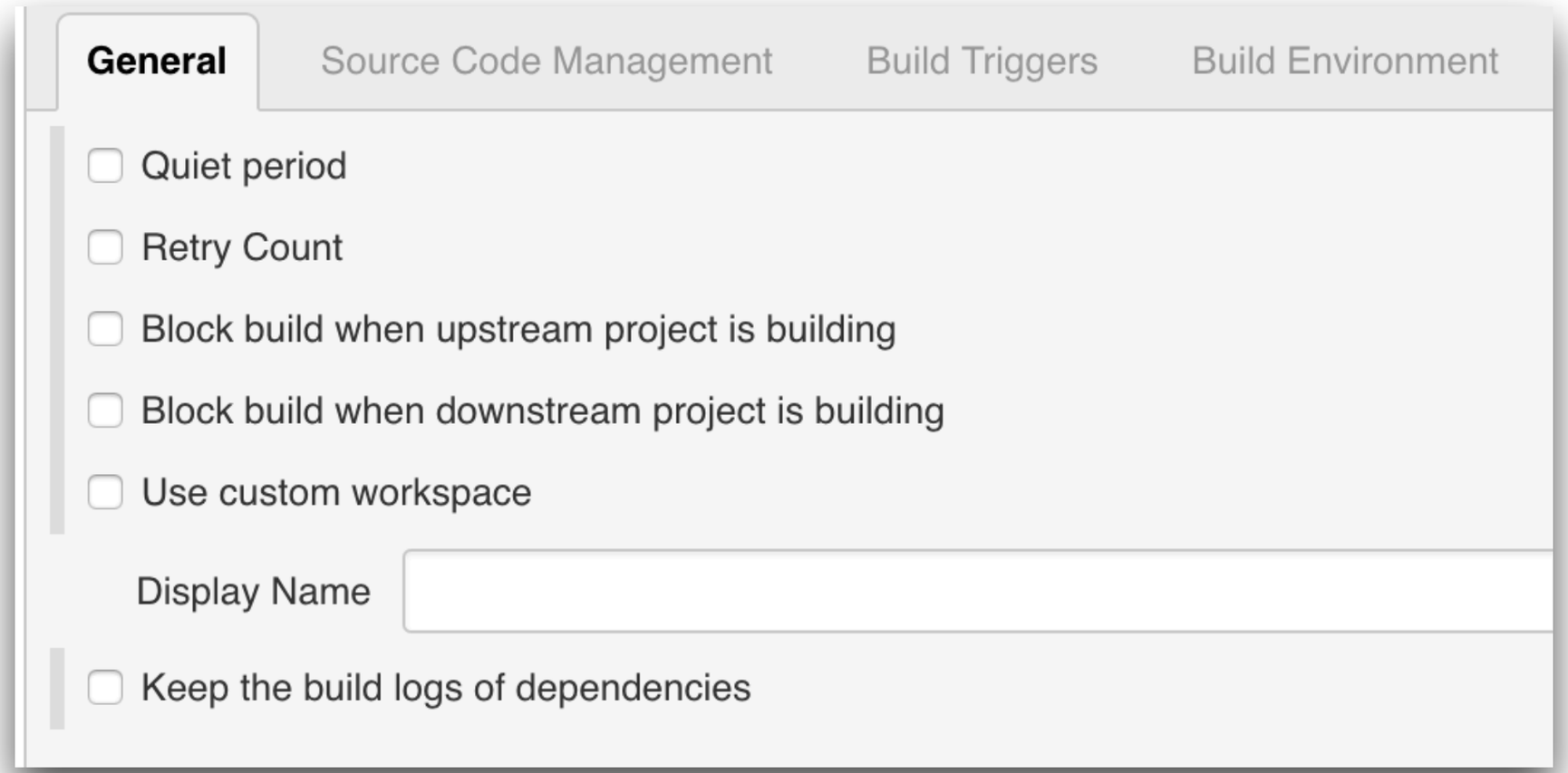
See help !!


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.

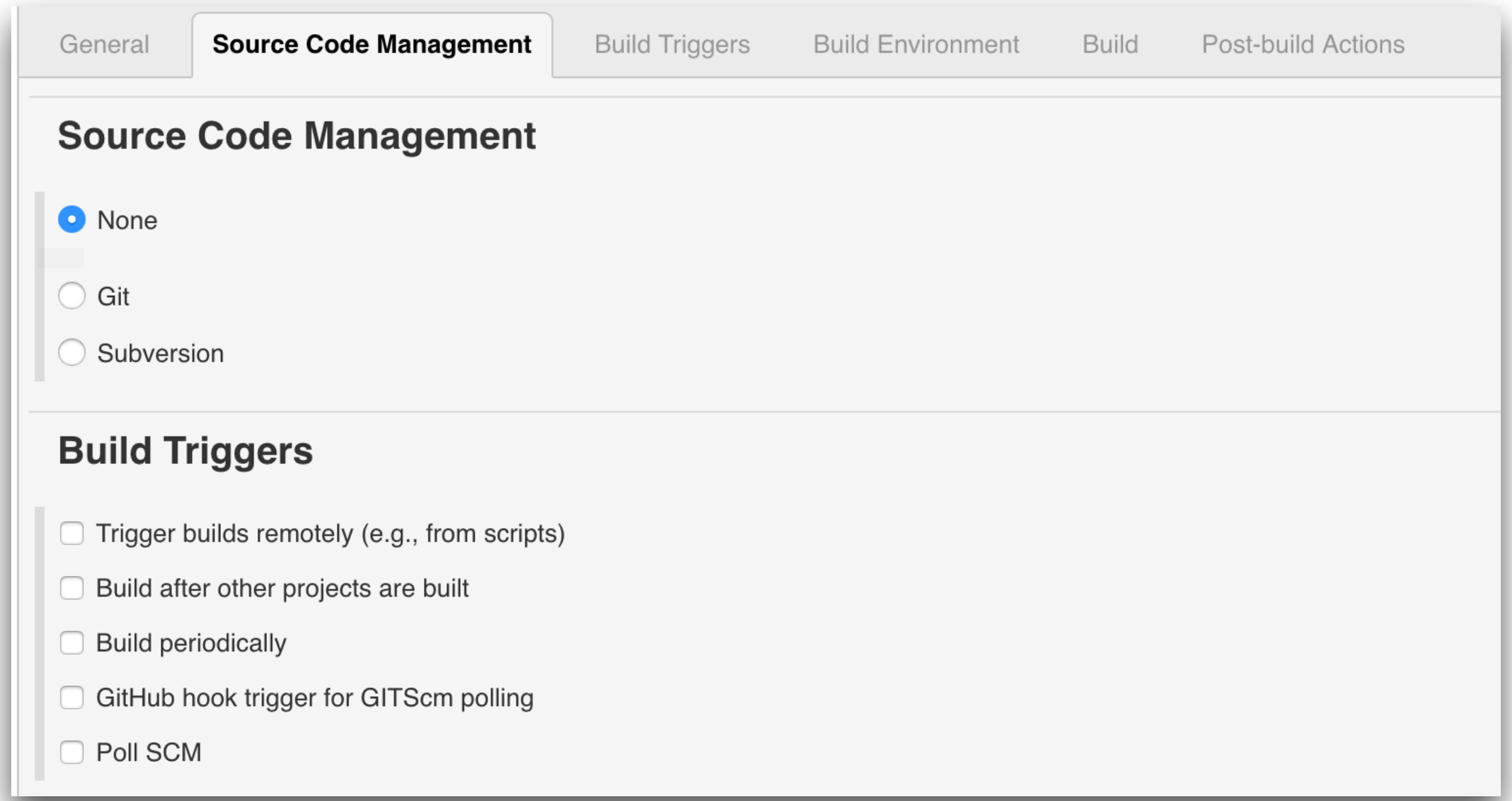
5. Advance project options



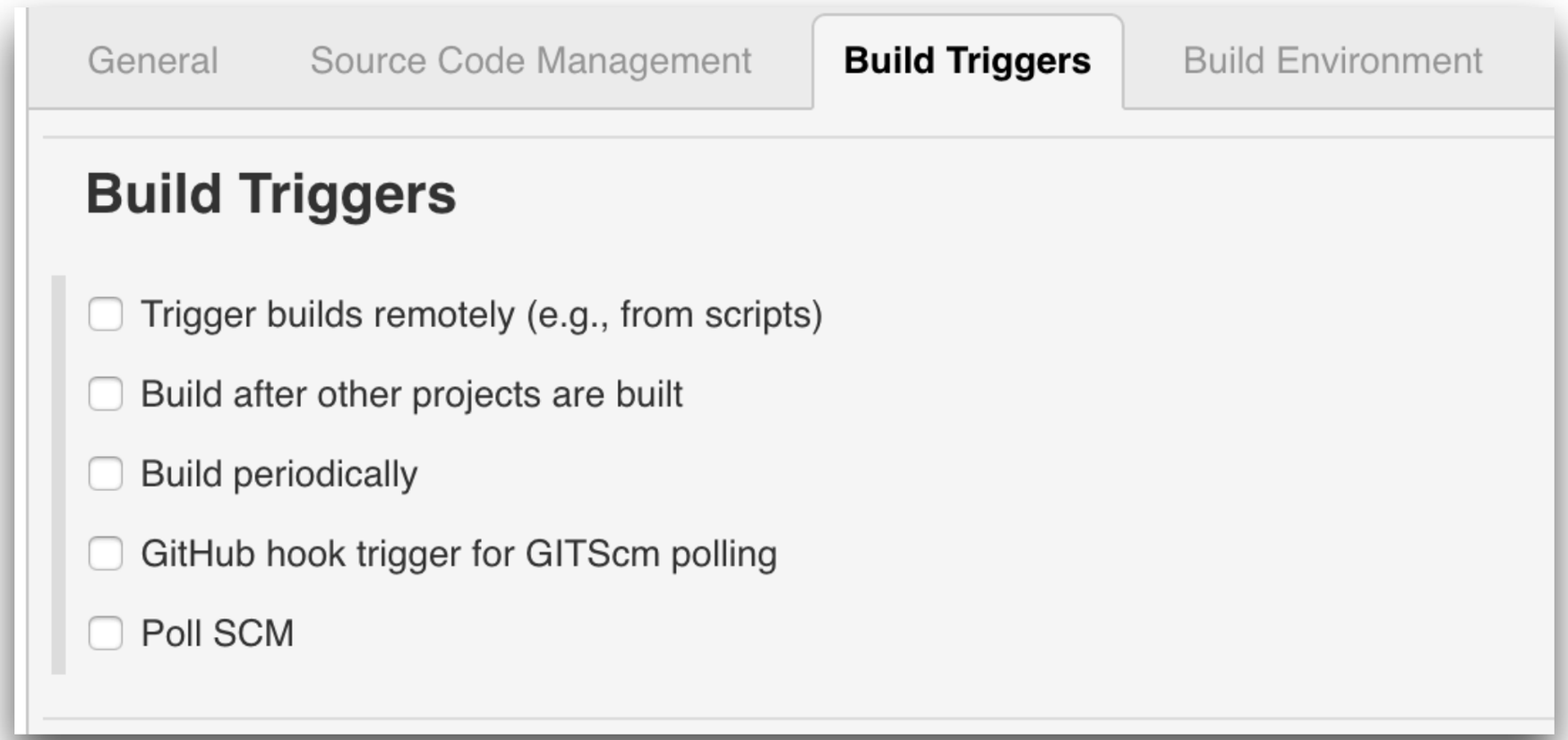
Advance project options



7. Source code management



8. Build Triggers



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
- [x] Build periodically

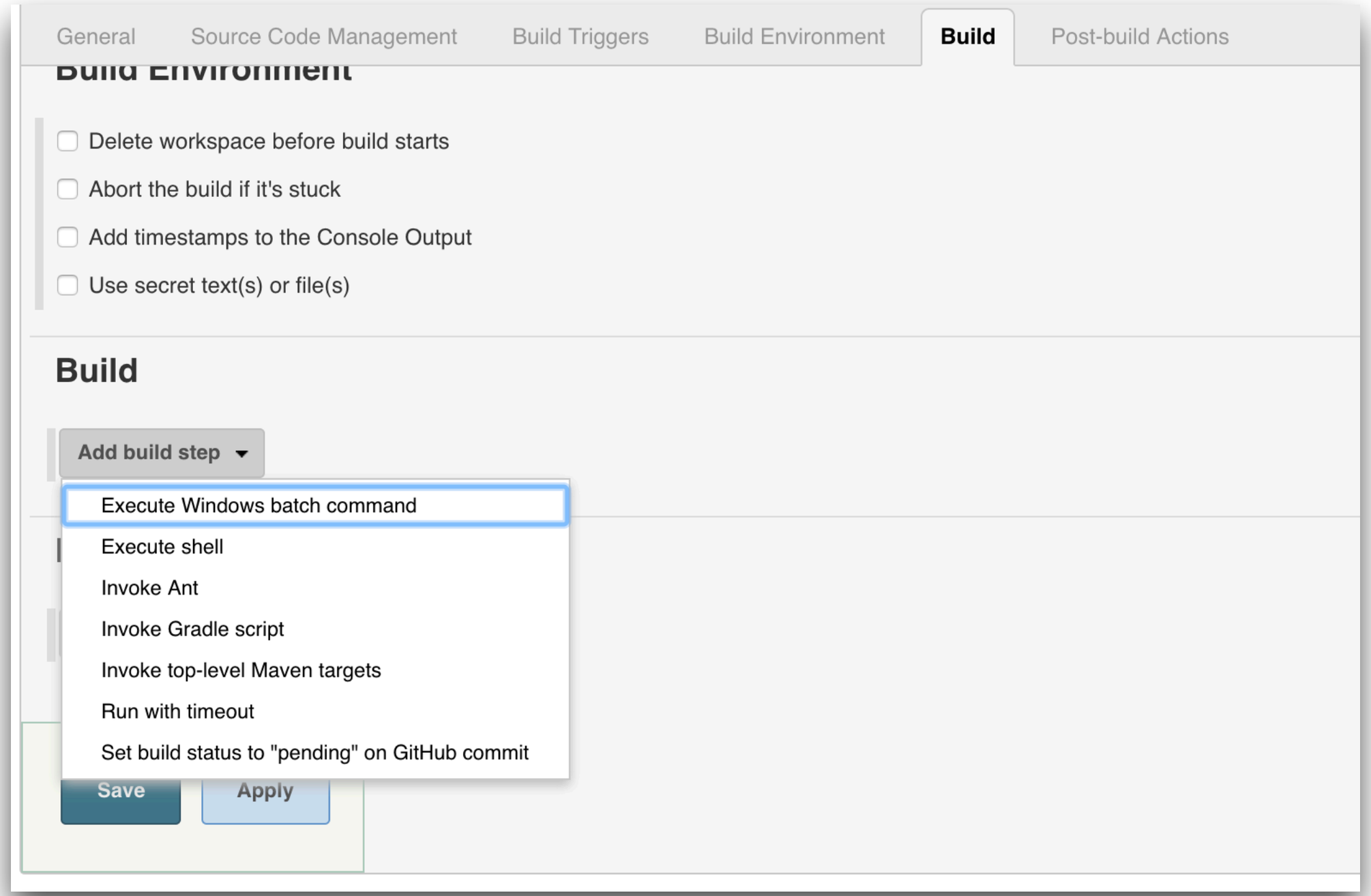
Schedule **H 23 * * ***

⚠ No schedules so will never run

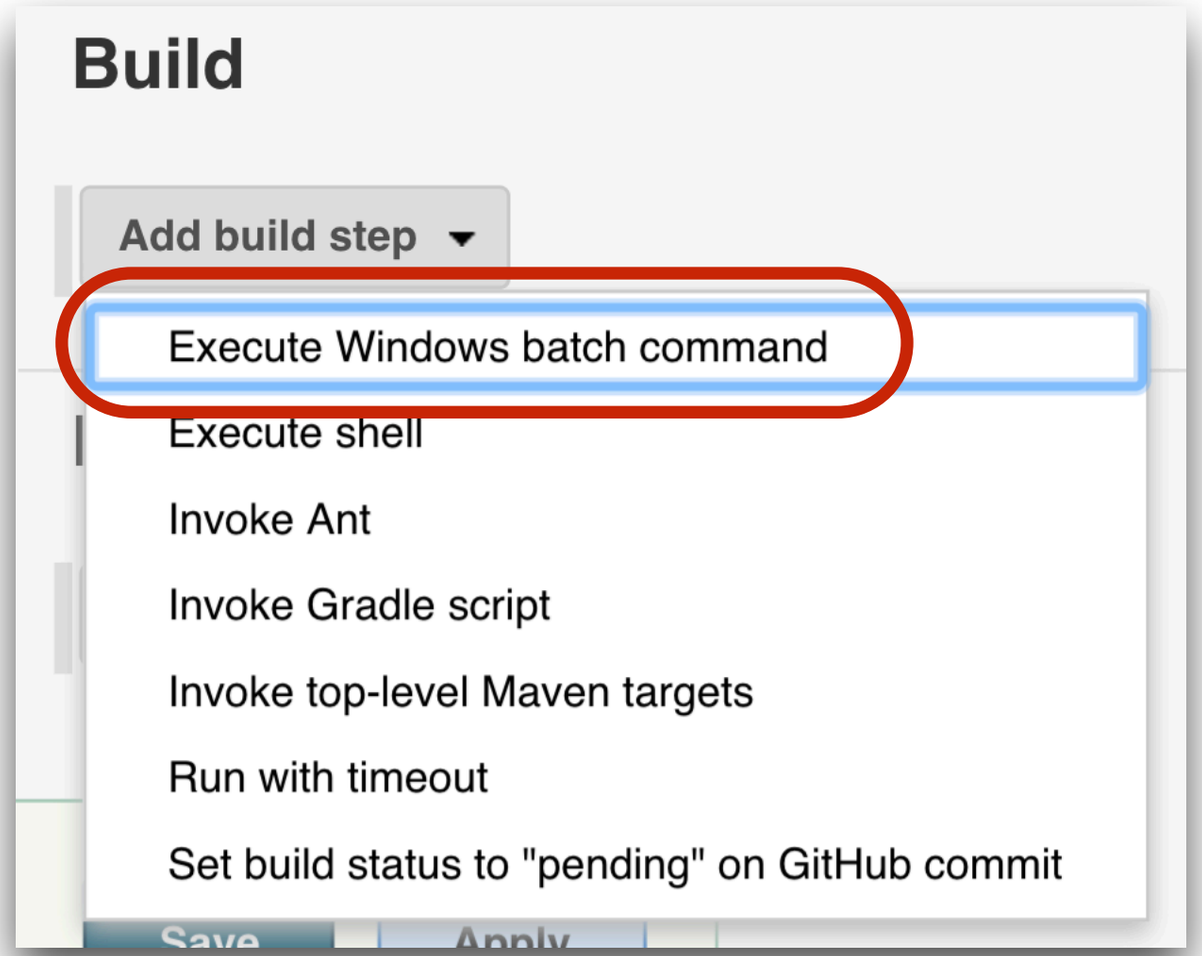
- [ ] GitHub hook trigger for GITScm polling
- [ ] Poll SCM

Run every day around 11.59 PM

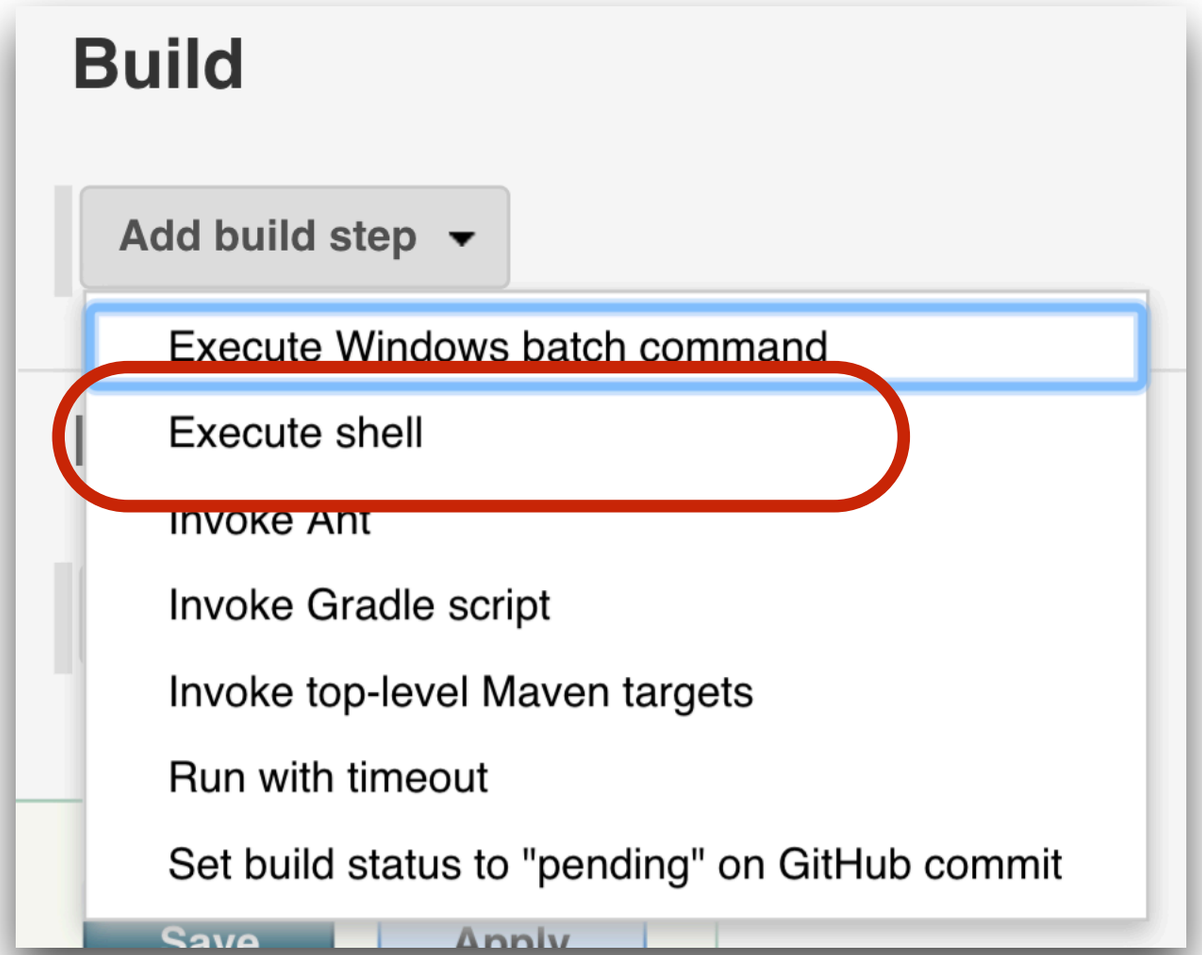
9. Add a Build step



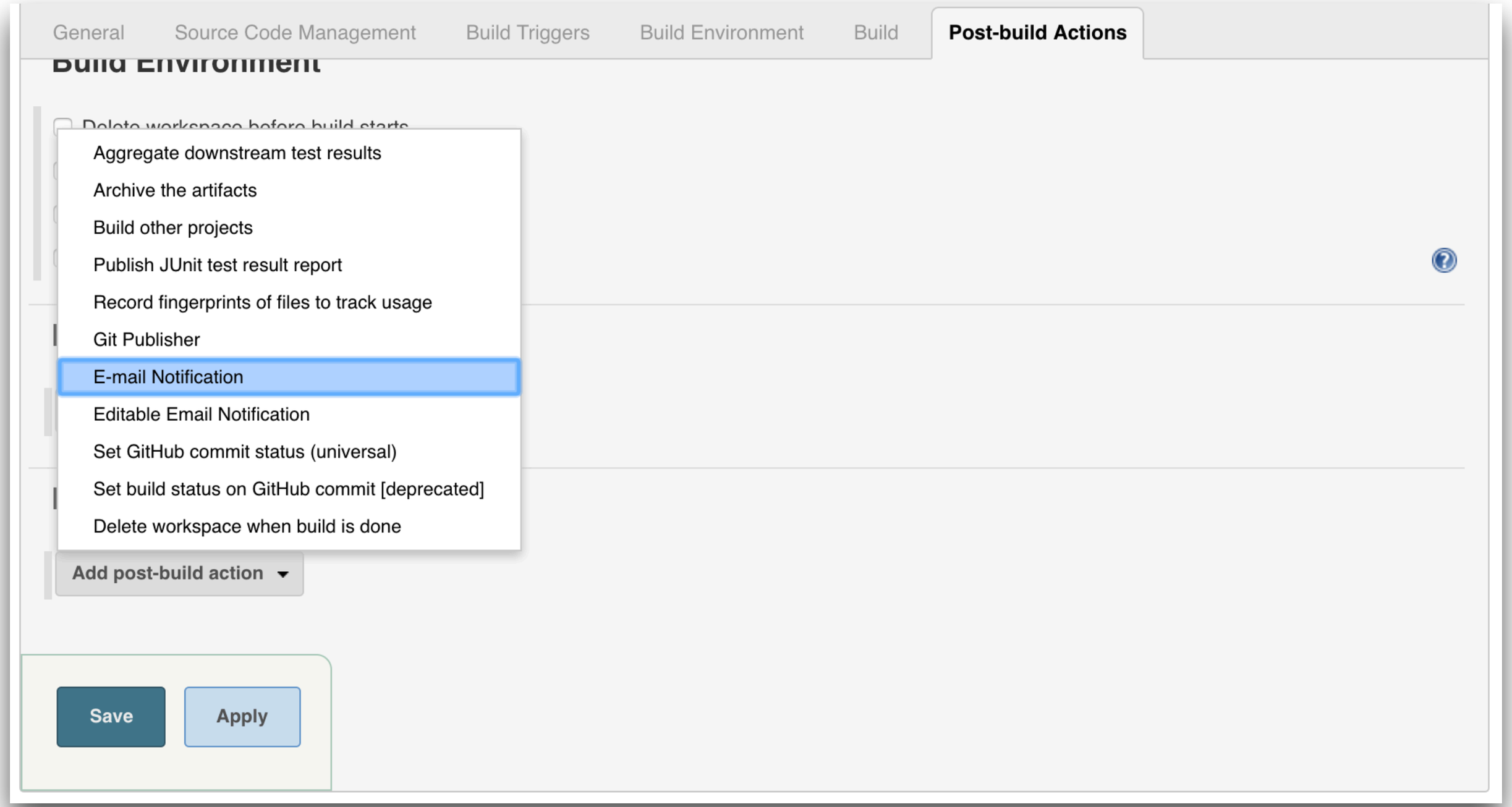
For Windows


Build
Add build step
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Run with timeout
Set build status to "pending" on GitHub commit
Save
Apply

For UNIX/Mac

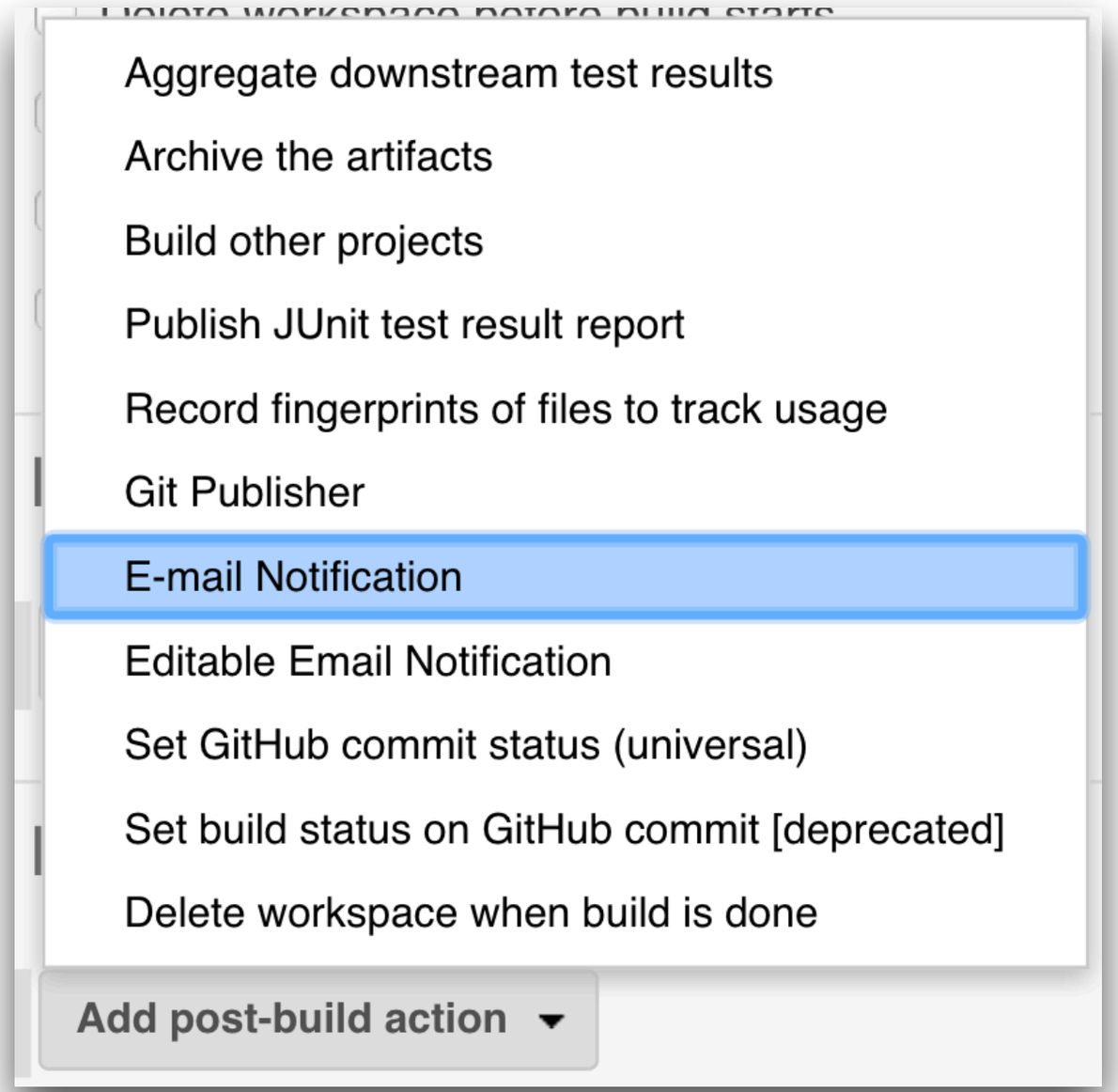

Build
Add build step
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Run with timeout
Set build status to "pending" on GitHub commit
Save
Apply

10. Post-build actions



Email notification

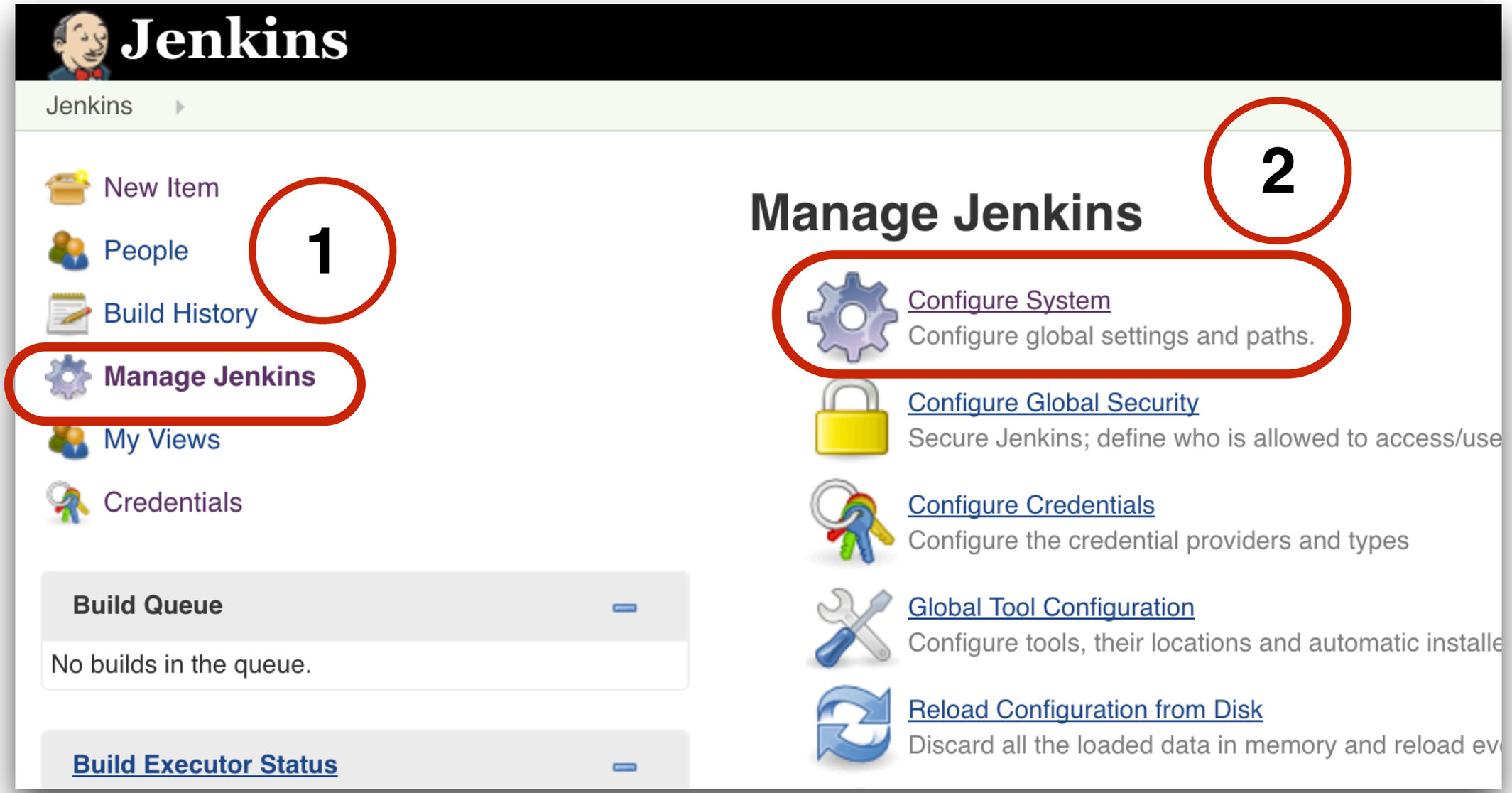
Email notification



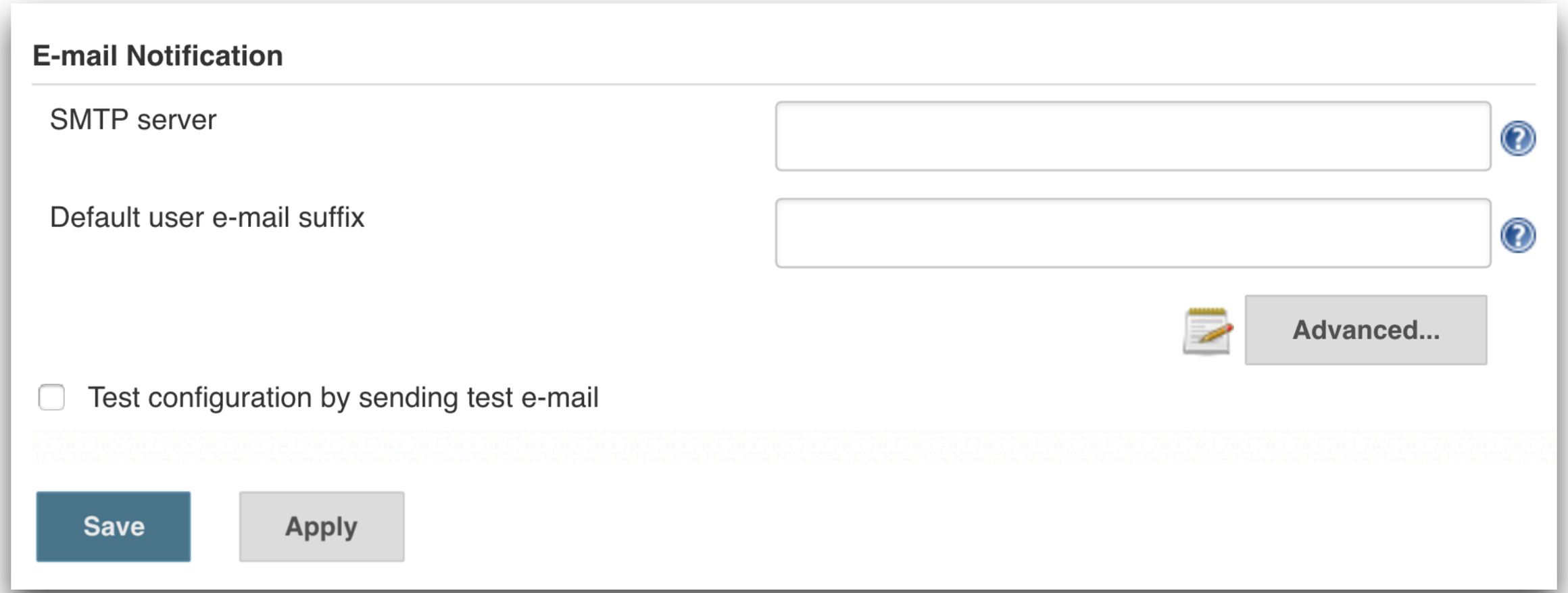
Add Recipients

Configure SMTP server

Manage Jenkins -> Configure System



Configure SMTP server



Configure SMTP server

E-mail Notification

SMTP server

Default user e-mail suffix

Use SMTP Authentication

Use SSL

SMTP Port

Reply-To Address

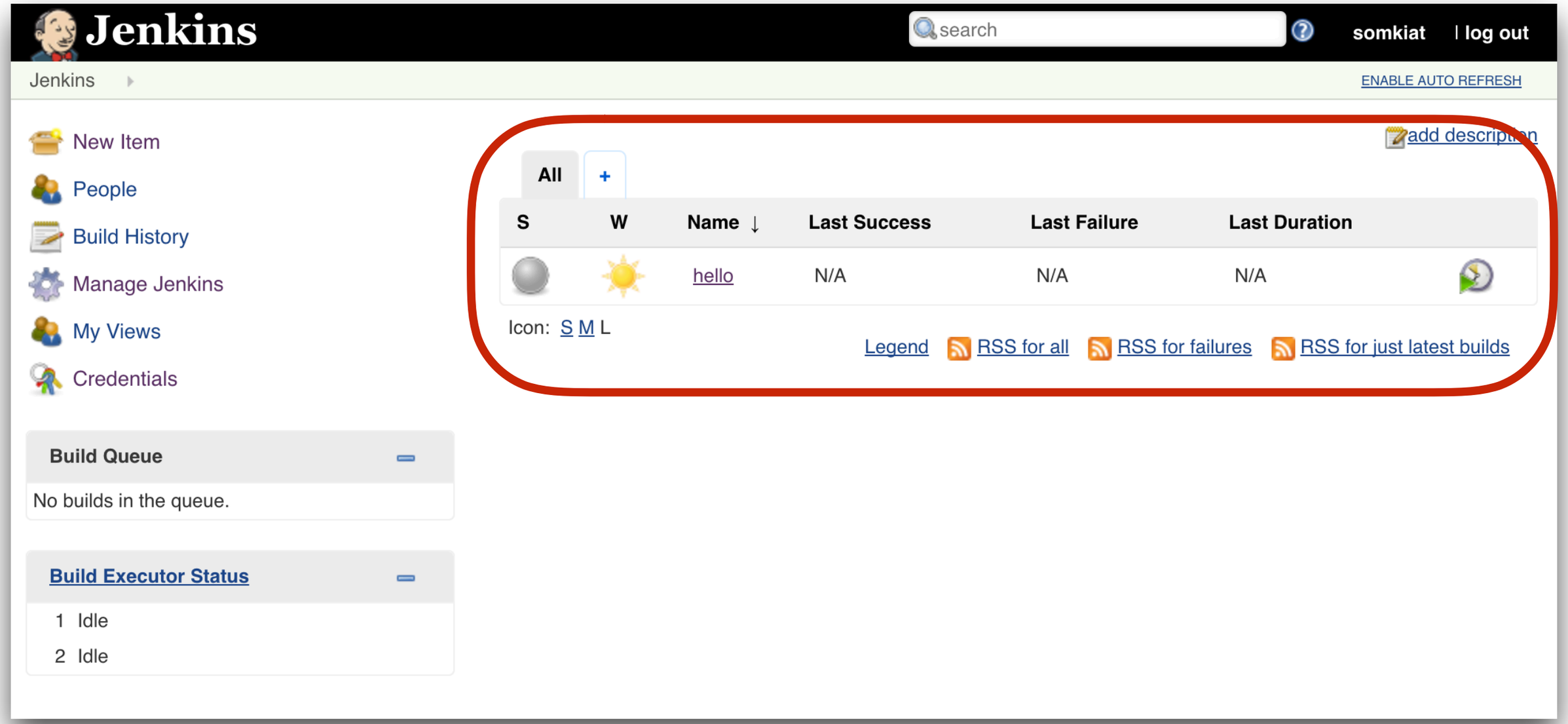
Charset

UTF-8

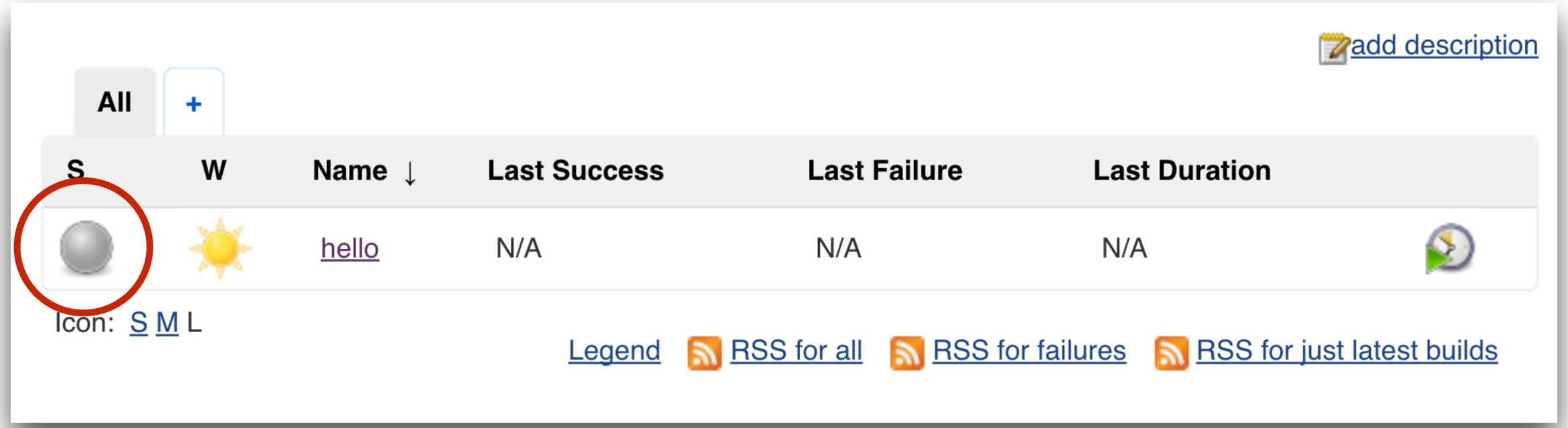
Test configuration by sending test e-mail

Save Apply

11. Run your job !!



Job status

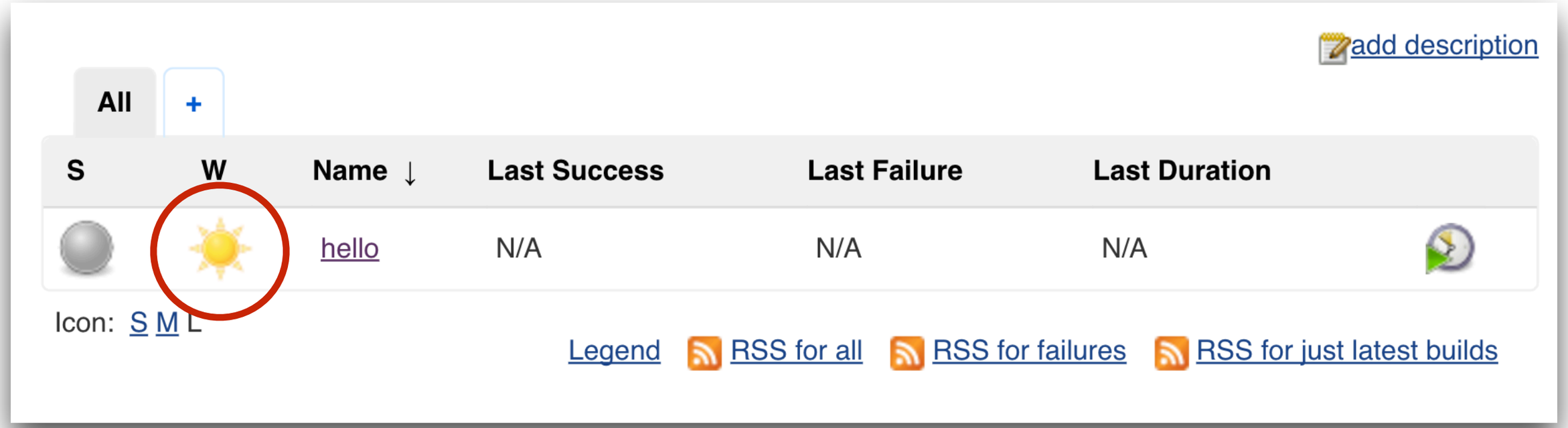


Blue = success

Red = failure

Gray = disabled/never execute

Job health

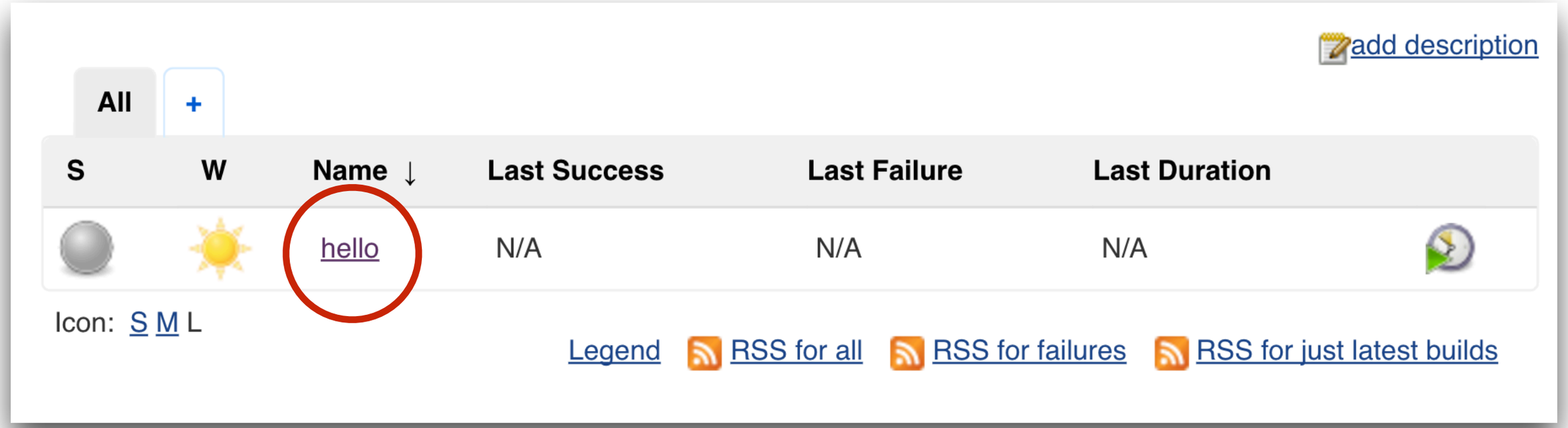


Sunny = 100% success rate

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

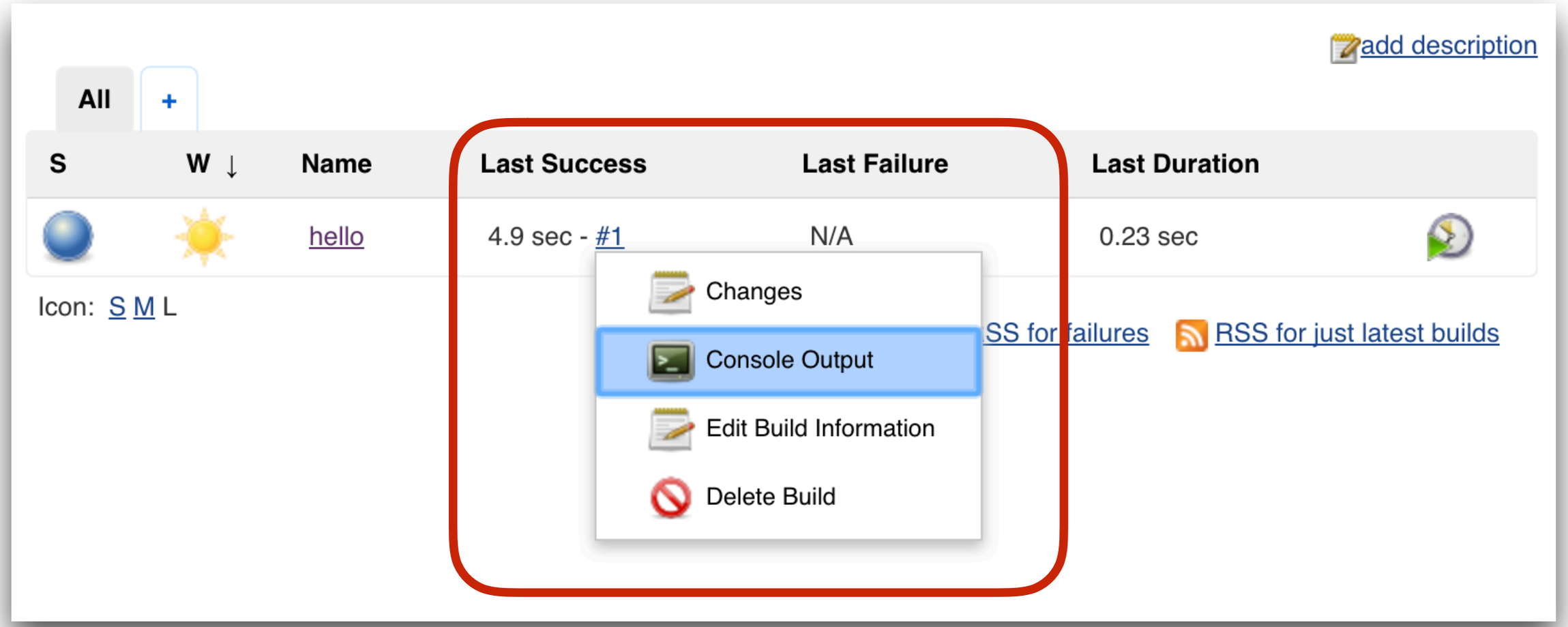
All +

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

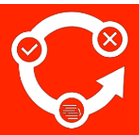


Console output

Console Output

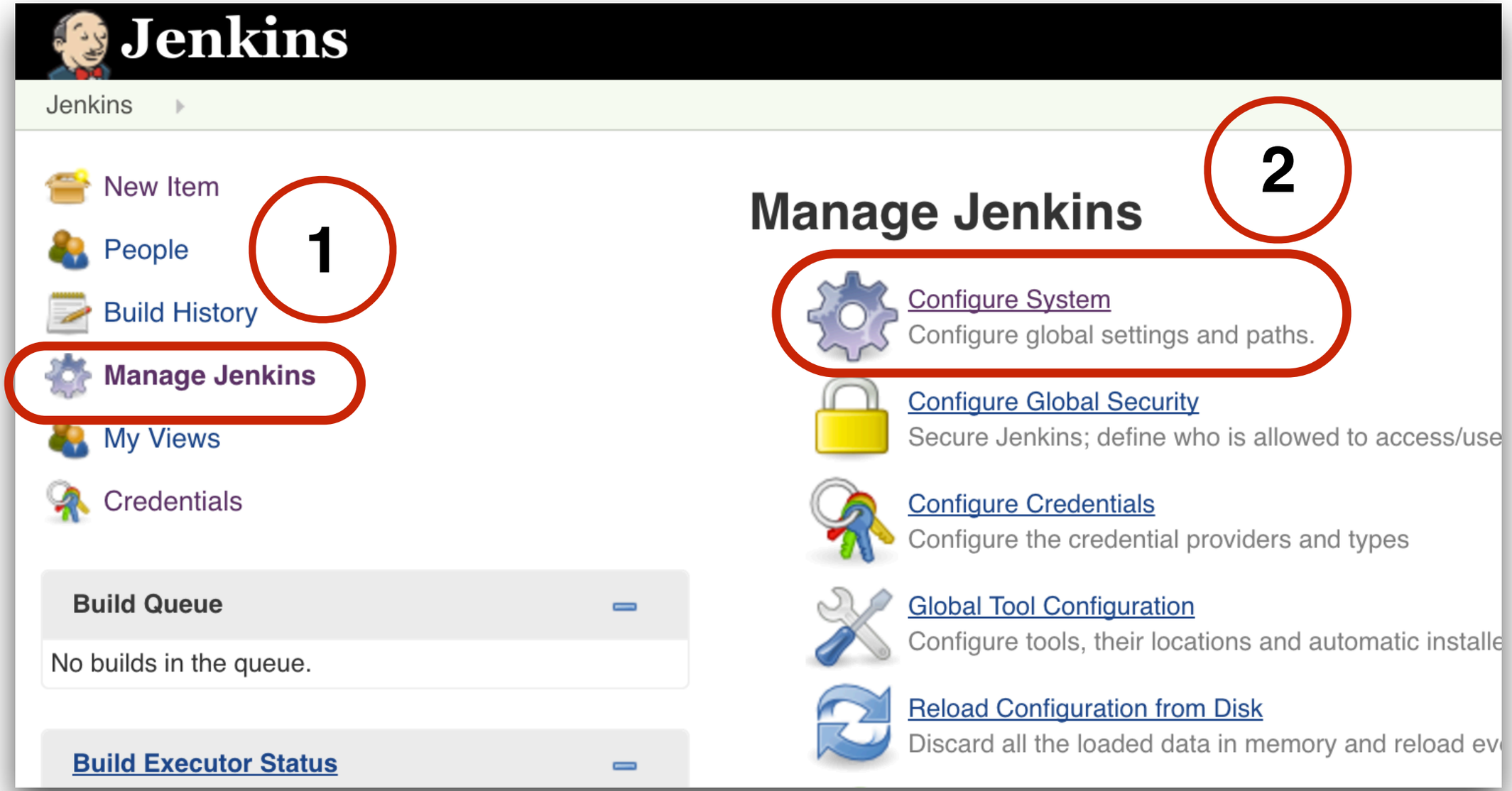
```
Started by user somkiat
Building in workspace /Users/somkiat/data/slide/ci-cd/swpark/
Finished: SUCCESS
```

Jenkins Home directory

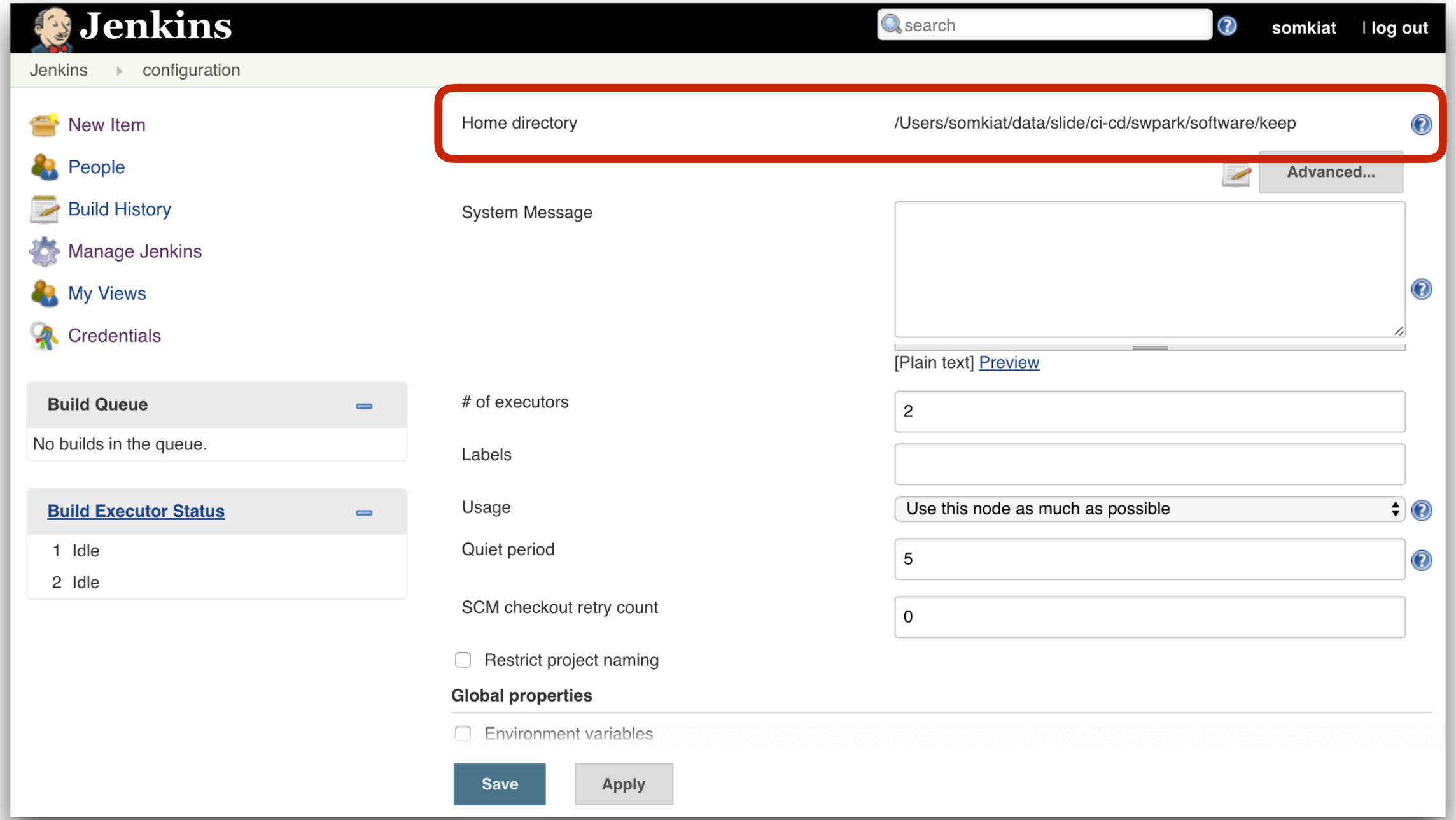


Default Jenkins Home

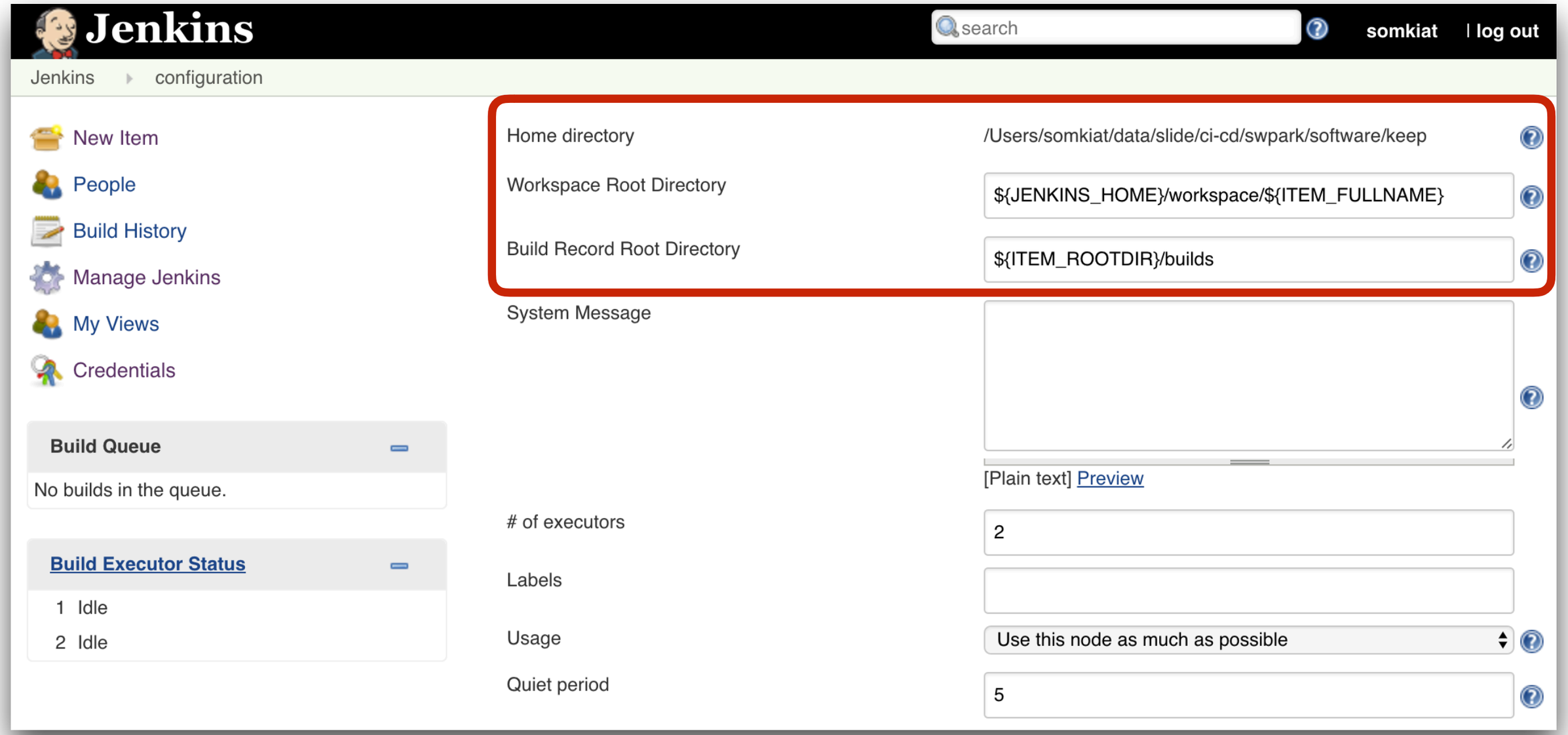
Manage Jenkins -> Configure System



Default Jenkins Home



Default Jenkins Home



Change Jenkins's Home

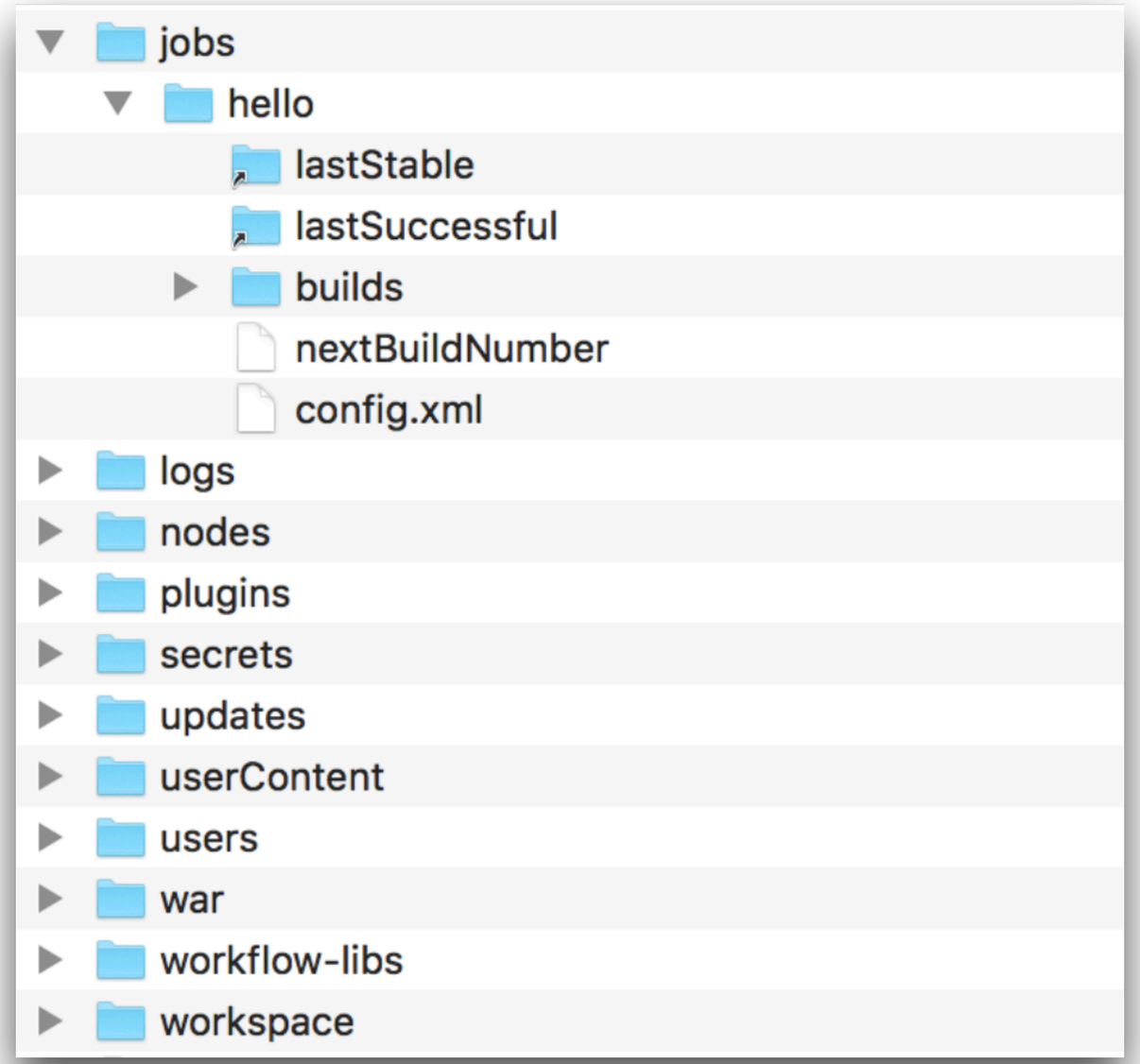
For Windows

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

For UNIX/MAC

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

Jenkins home structure



Jenkins Job

config.xml

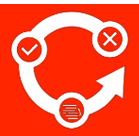
+ jenkins job configuration

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

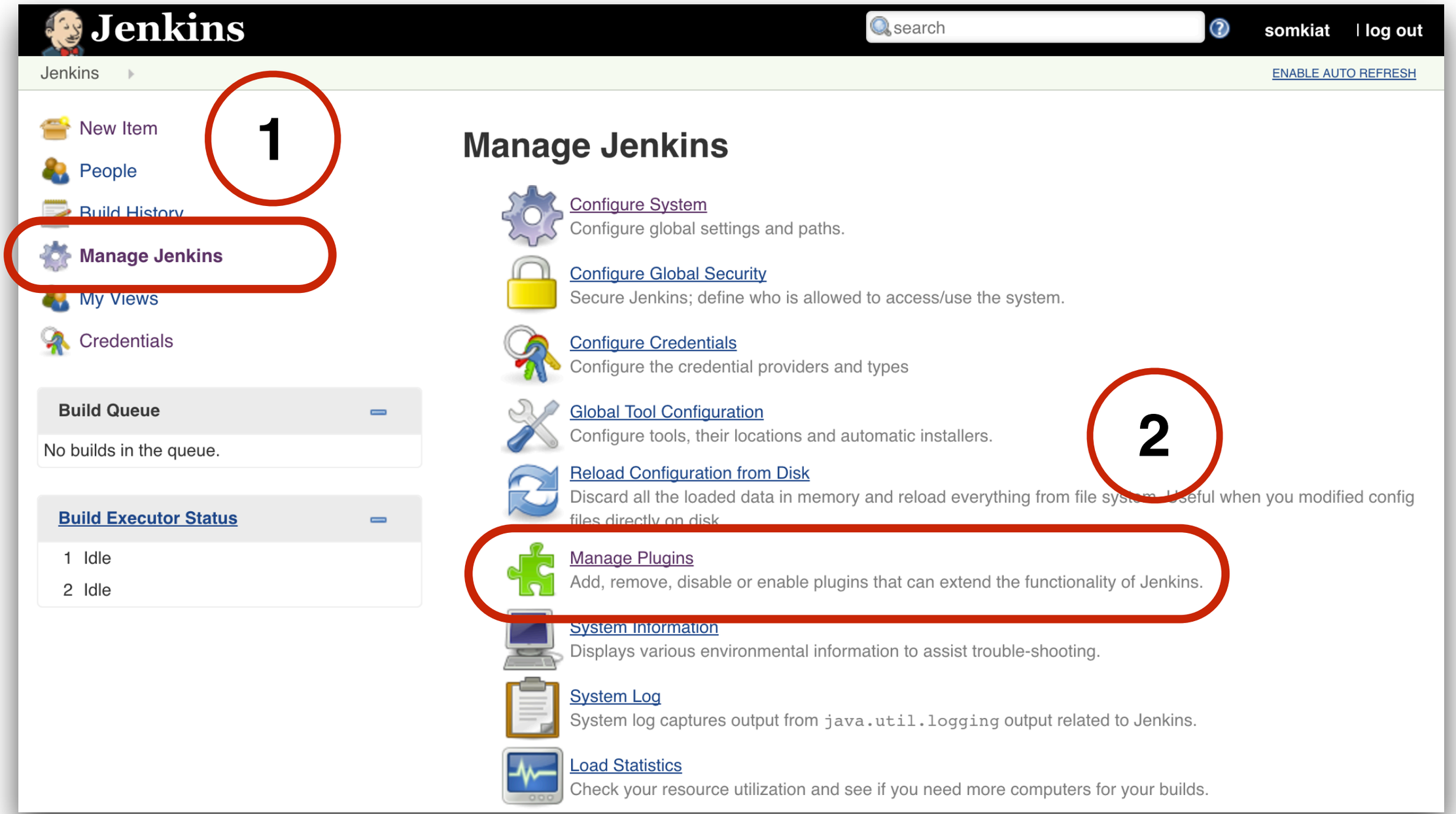
https://jenkins.io/doc/book/managing/plugins/

Find your plugins

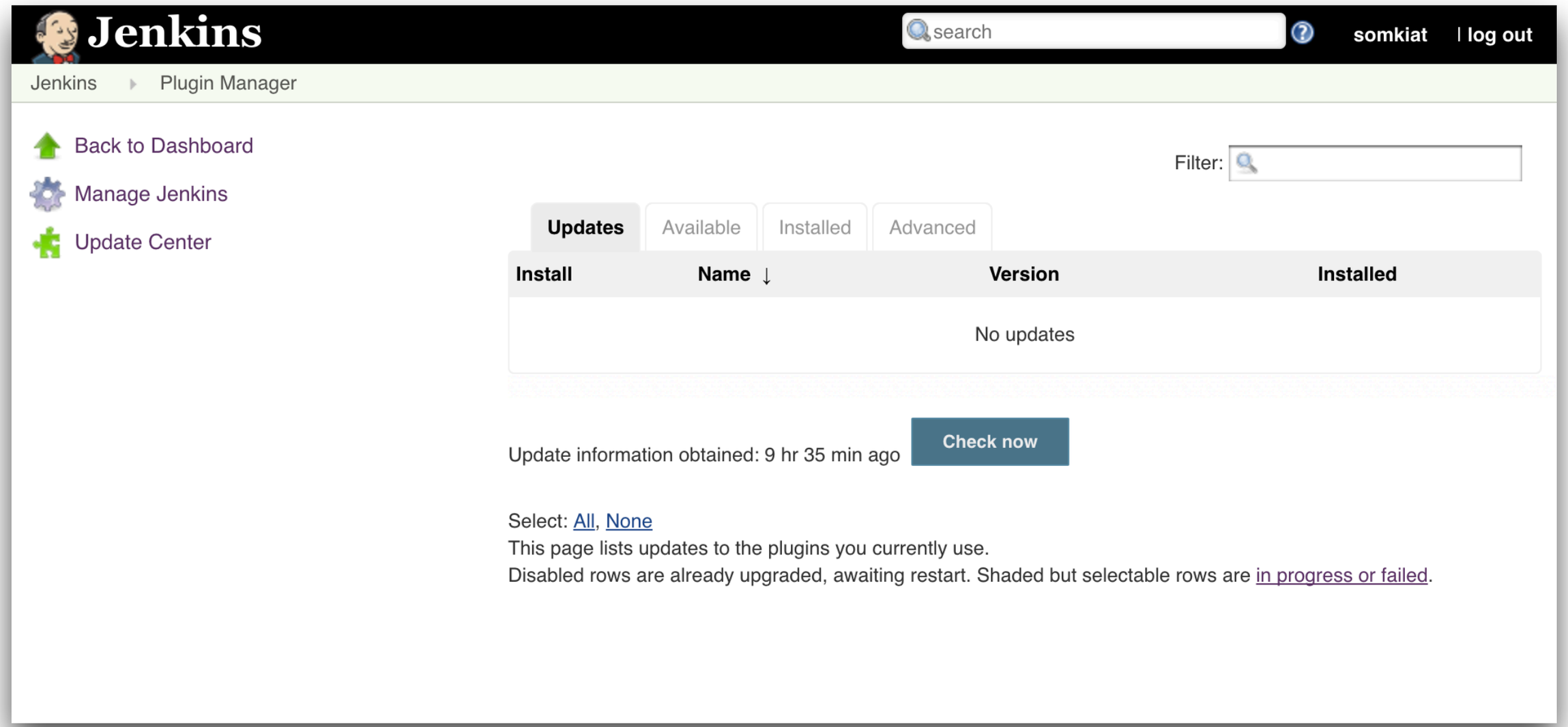
https://plugins.jenkins.io/

Manage Plugins

Manage Jenkins -> Manage Plugins



Plugin manager



Plugin manager

Filter:

Updates | **Available** | Installed | Advanced

Install ↓	Name	Version
.NET Development		
☐	CCM Plug-in This plug-in generates the trend report for CCM, an open source static code analysis program.	3.1
☐	FxCop Runner plugin	1.1
☐	MSBuild Plugin	1.27
☐	MSTest plugin Generates test reports for MSTest.	0.19
☐	MSTestRunner plugin	1.3.0
☐	NAnt Plugin	1.4.3
☐	NCover plugin	0.3
☐	PowerShell plugin	1.3
☐	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
☐	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

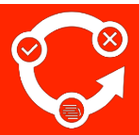
List of updates for the plugins installed on the current Jenkins instance

Available tab

List of all the plugins available from Jenkins community

Installed tab

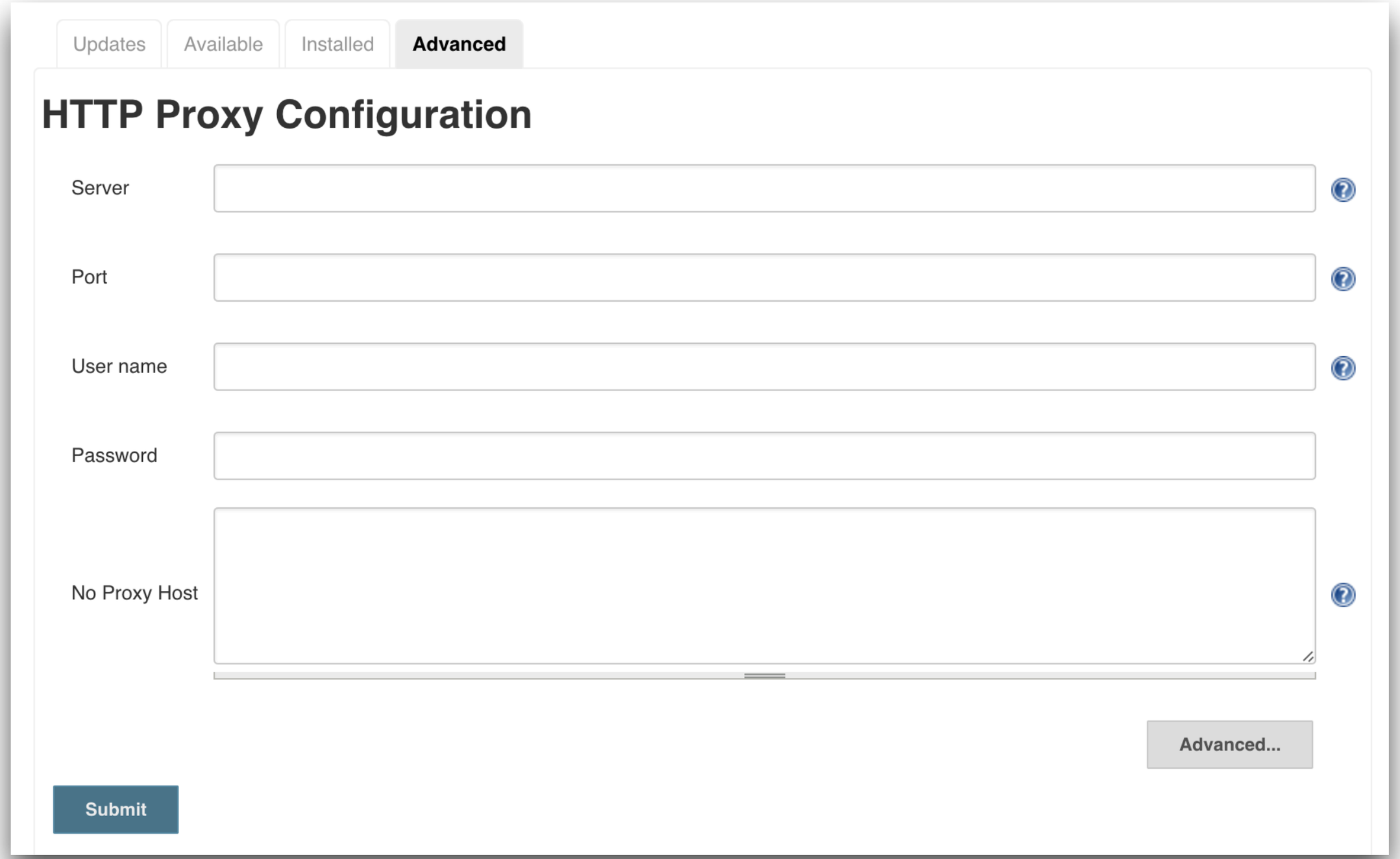
List of all the plugins installed
on the current Jenkins instance



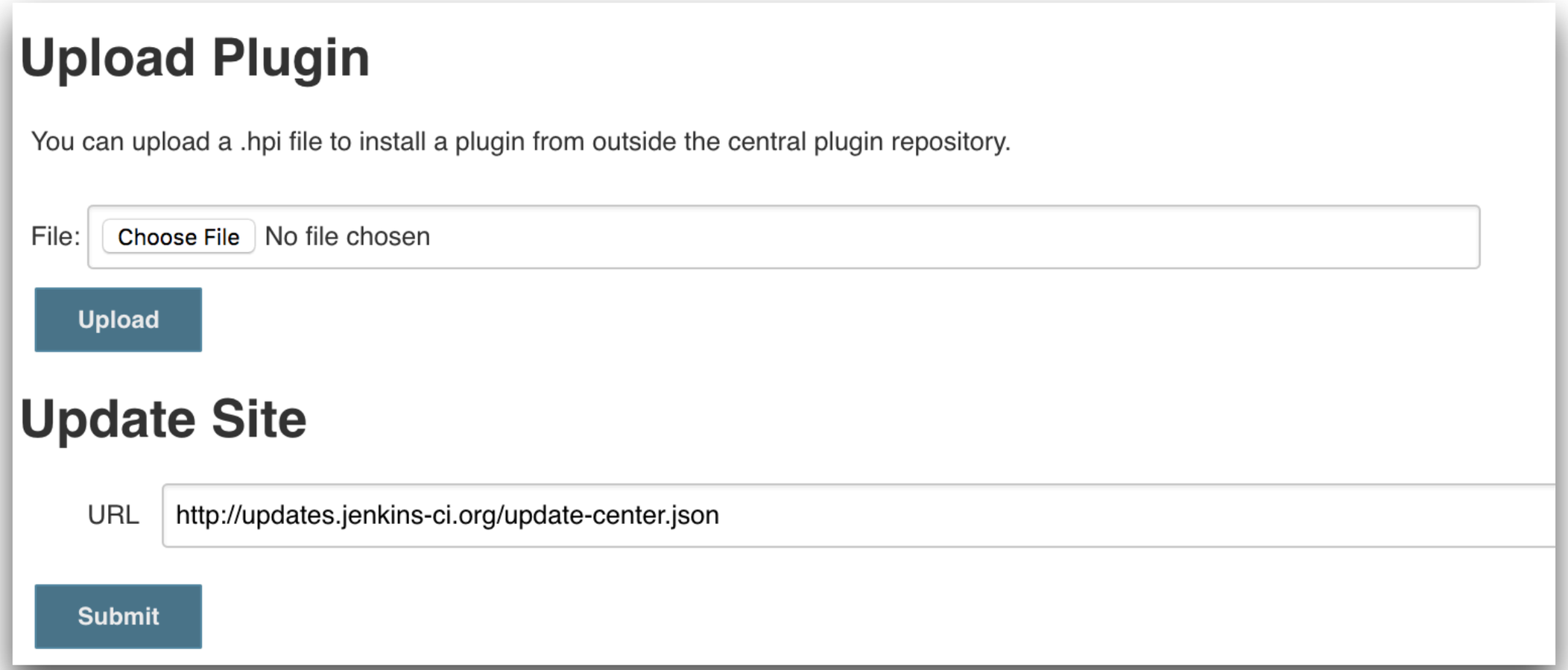
Advanced tab

Configure internet settings
and update Jenkins plugins manually

Advanced tab



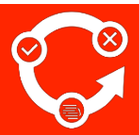
Advanced tab



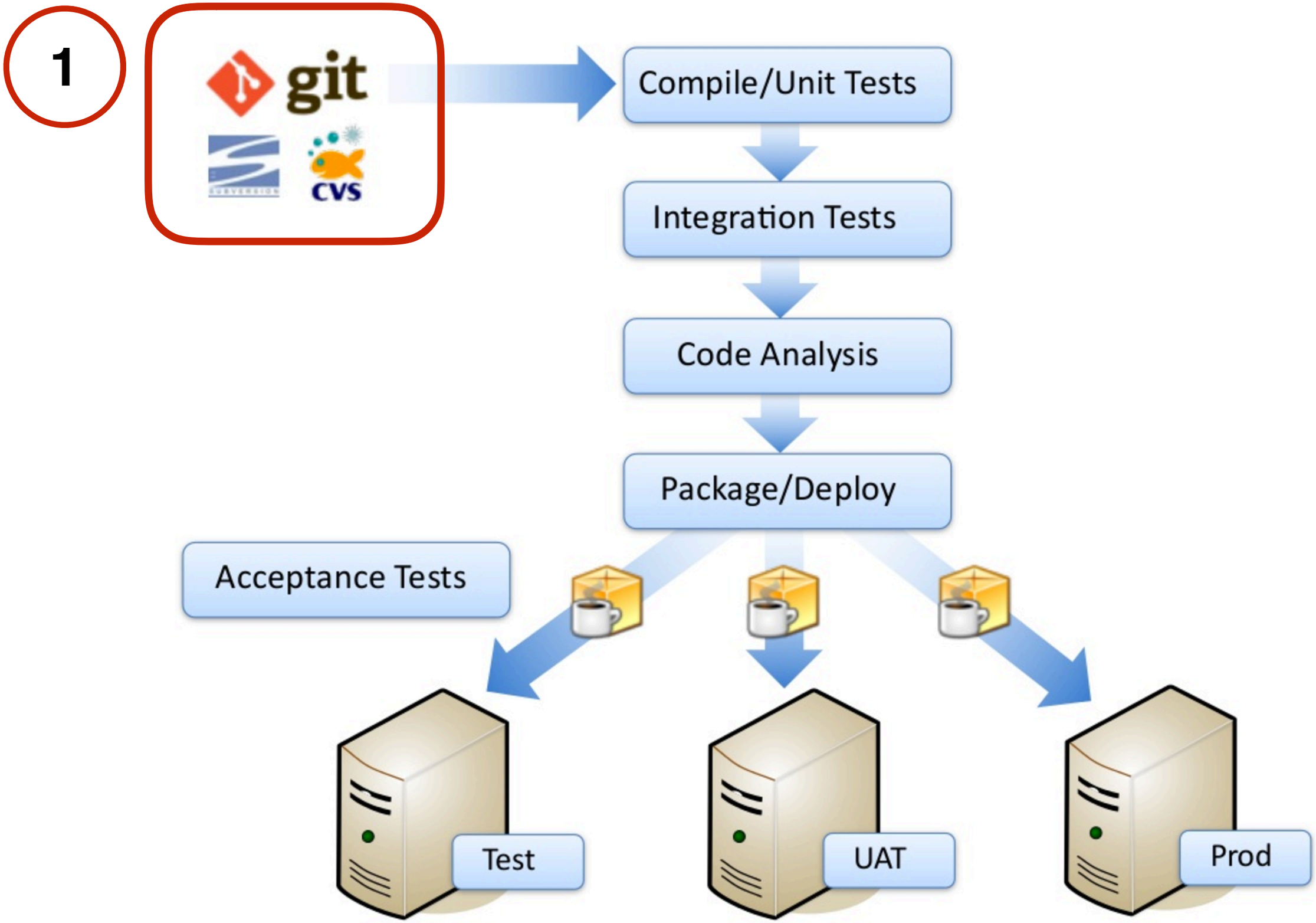
https://updates.jenkins-ci.org/download/plugins/

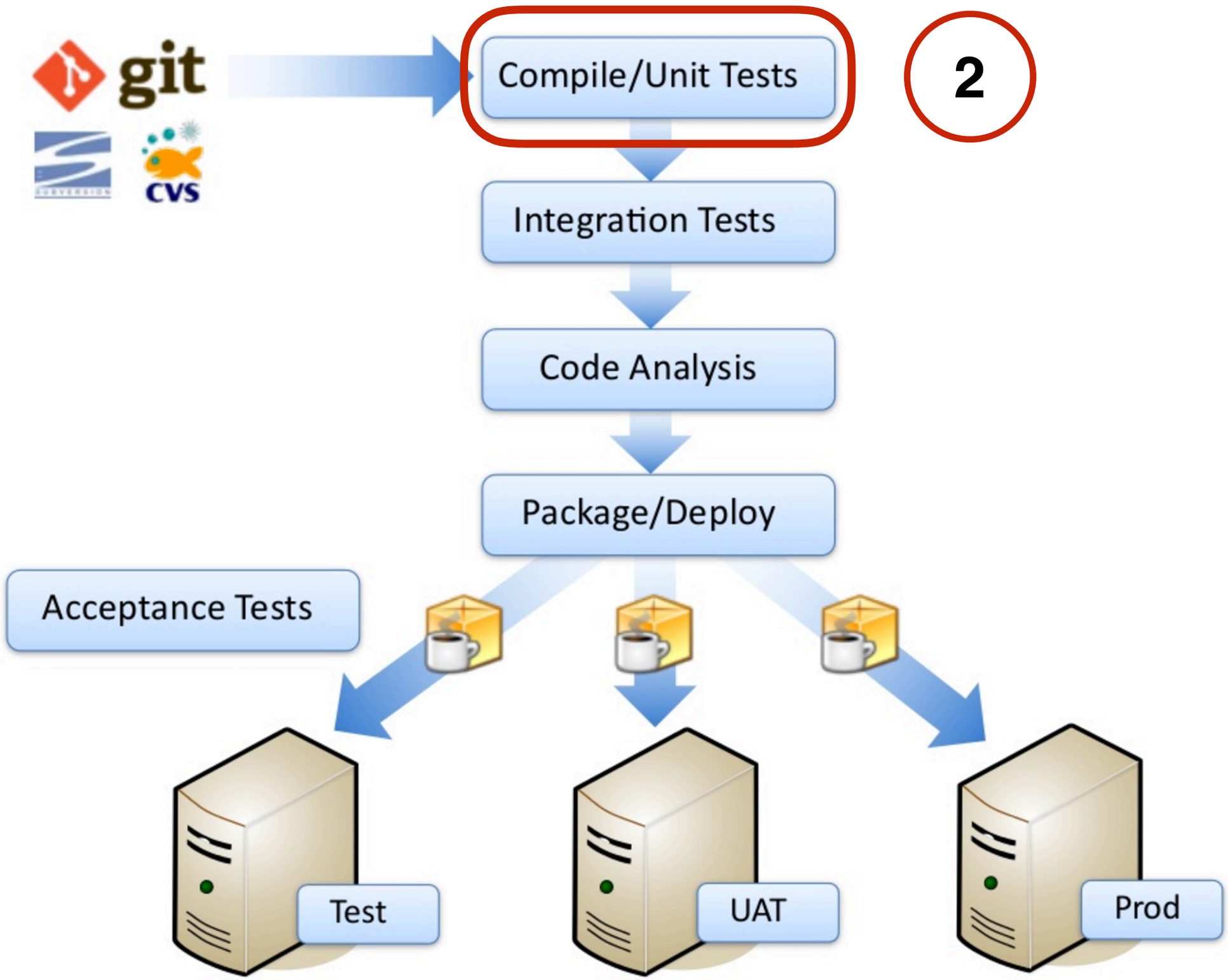
Install Jenkins plugins

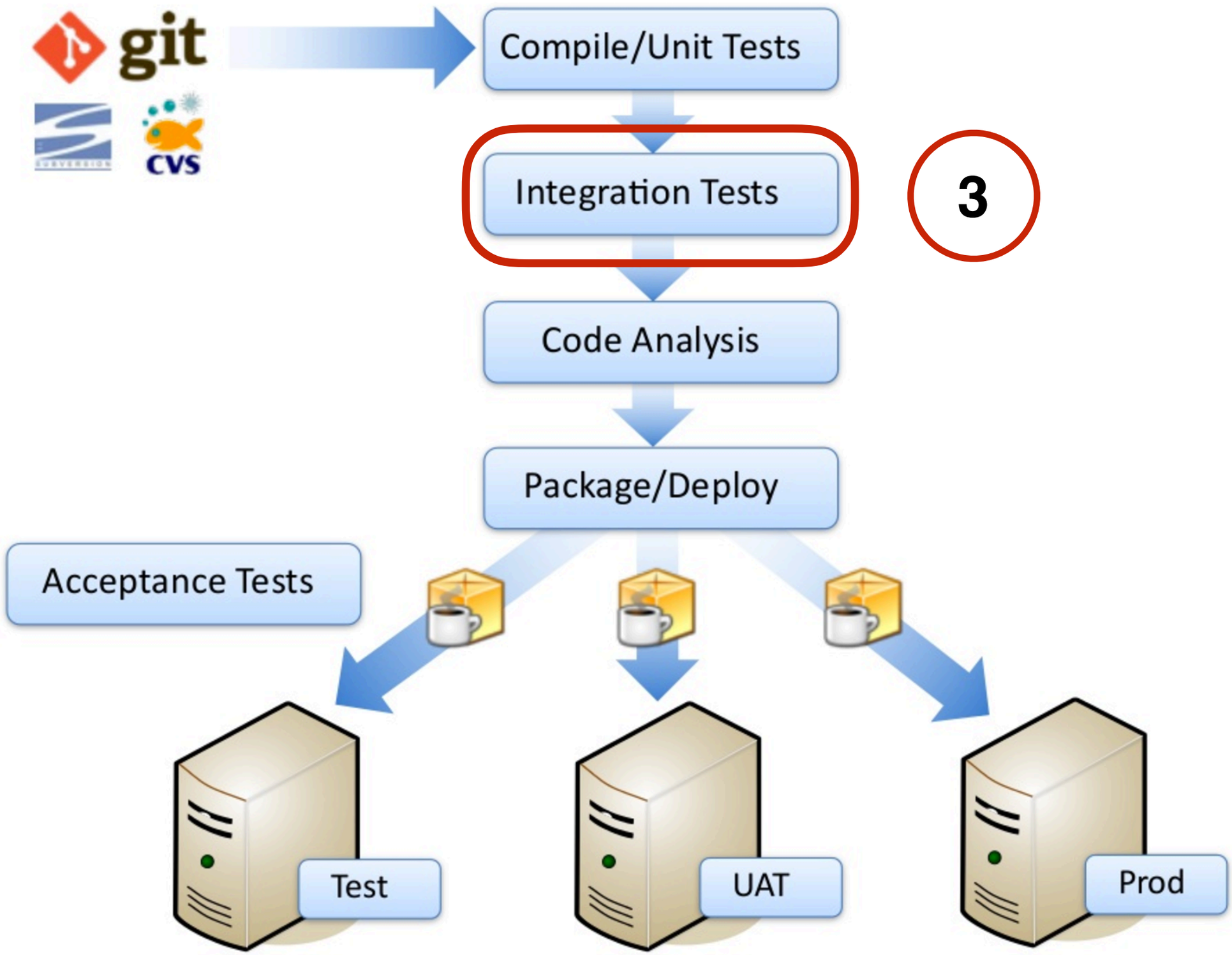
Jenkins driven by **Plugins**

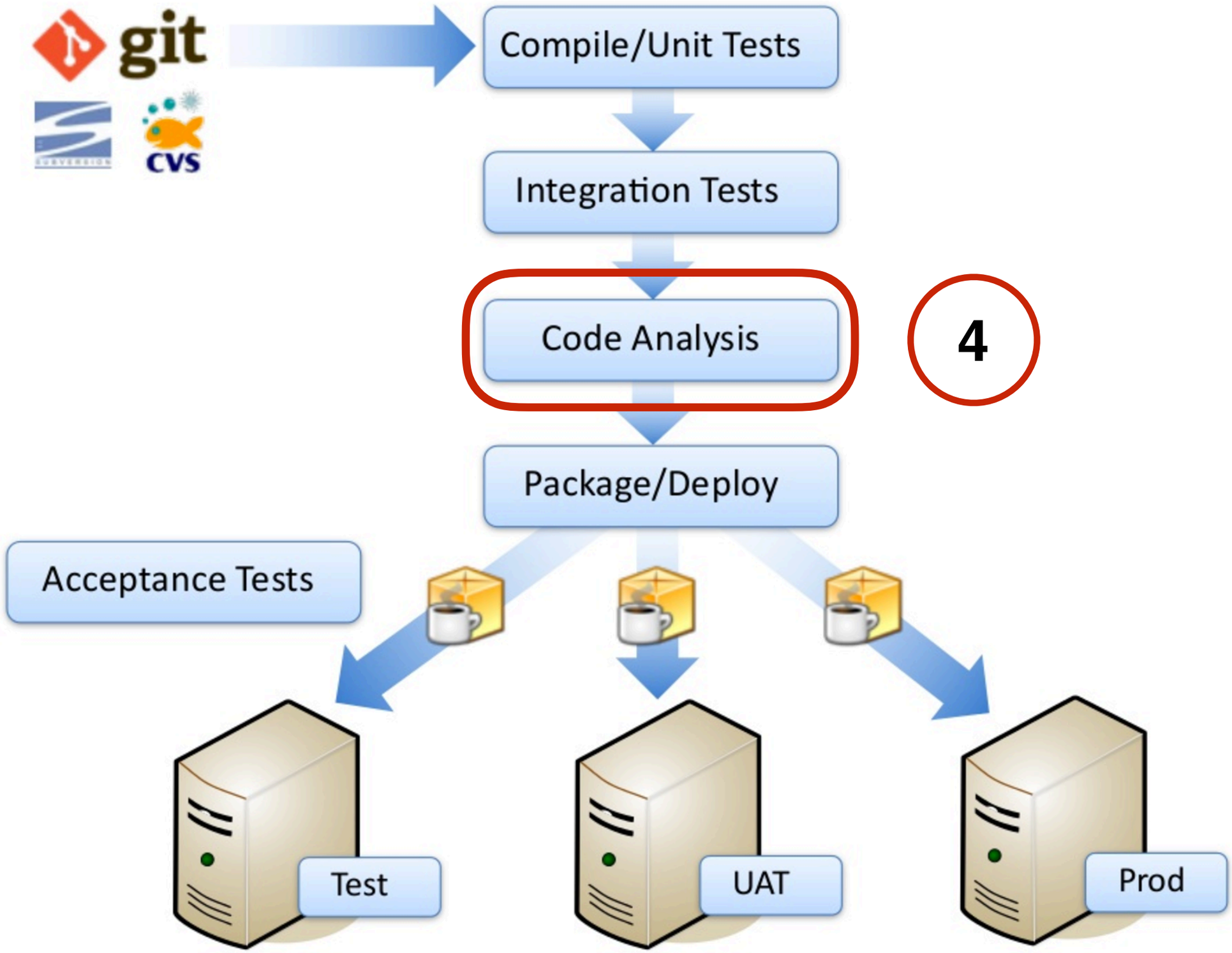


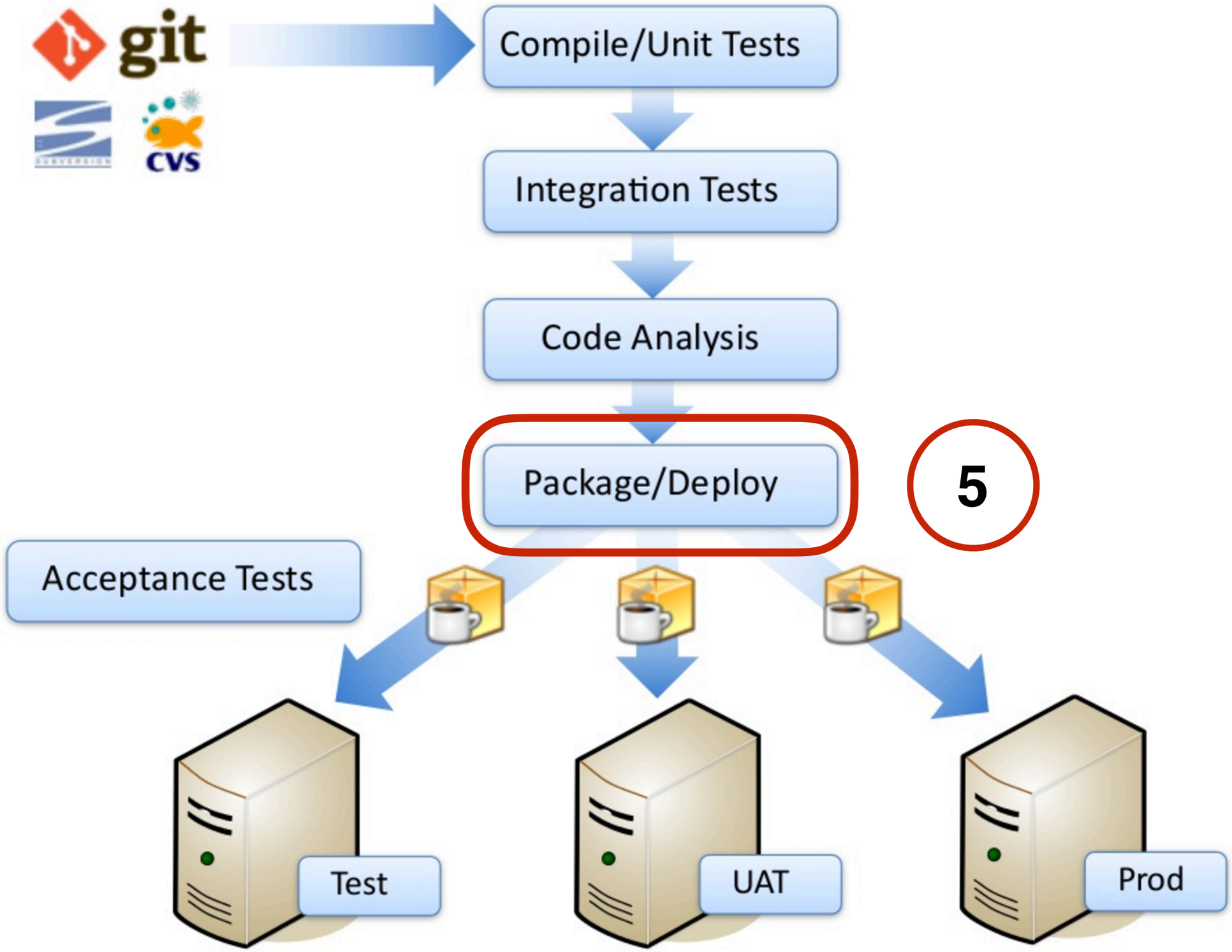
Let’ start with Jenkins

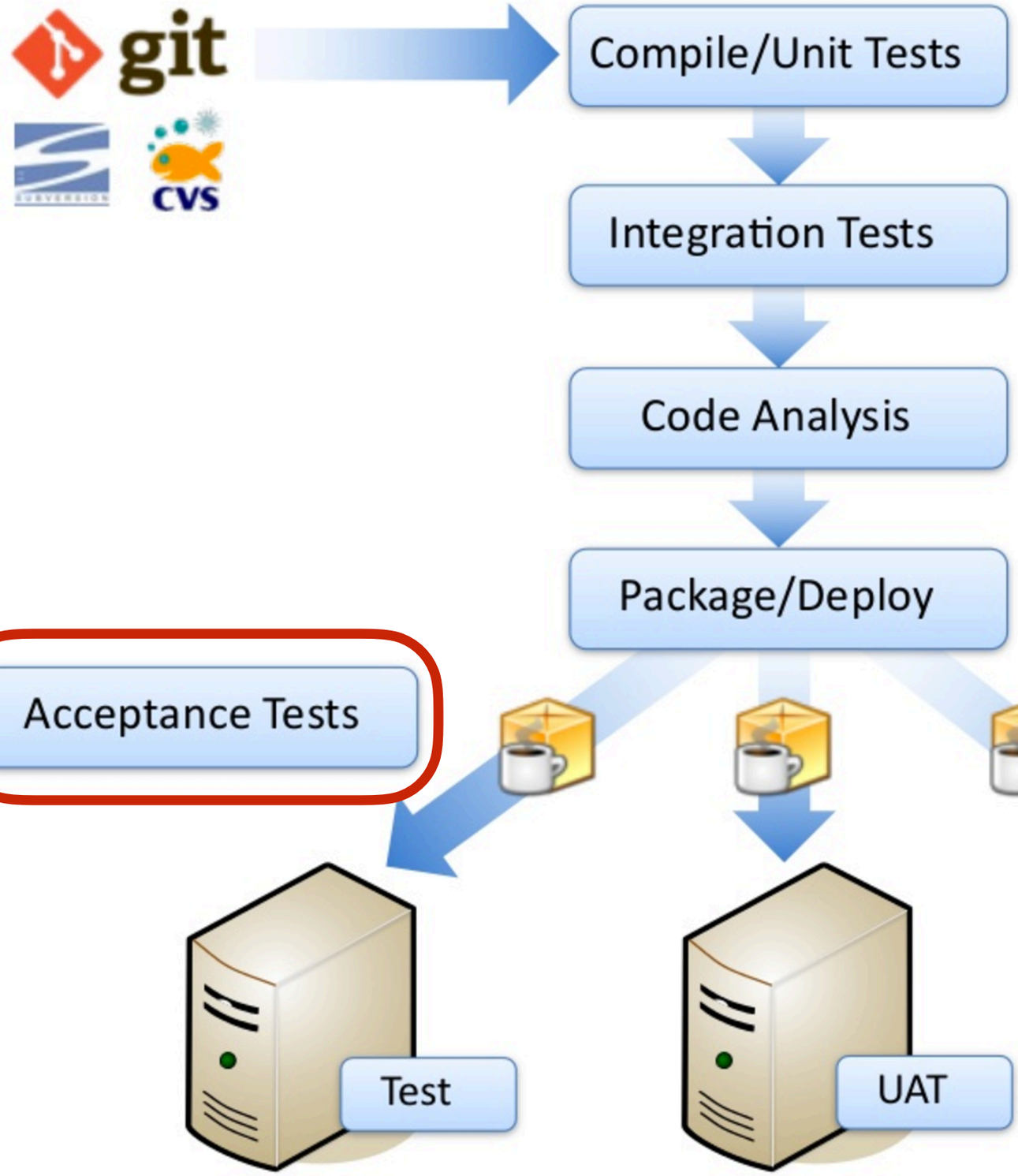

1
git
CVS
Compile/Unit Tests
Integration Tests
Code Analysis
Package/Deploy
Acceptance Tests
Test
UAT
Prod


git
SVN
CVS
Compile/Unit Tests
2
Integration Tests
Code Analysis
Package/Deploy
Acceptance Tests
Test
UAT
Prod

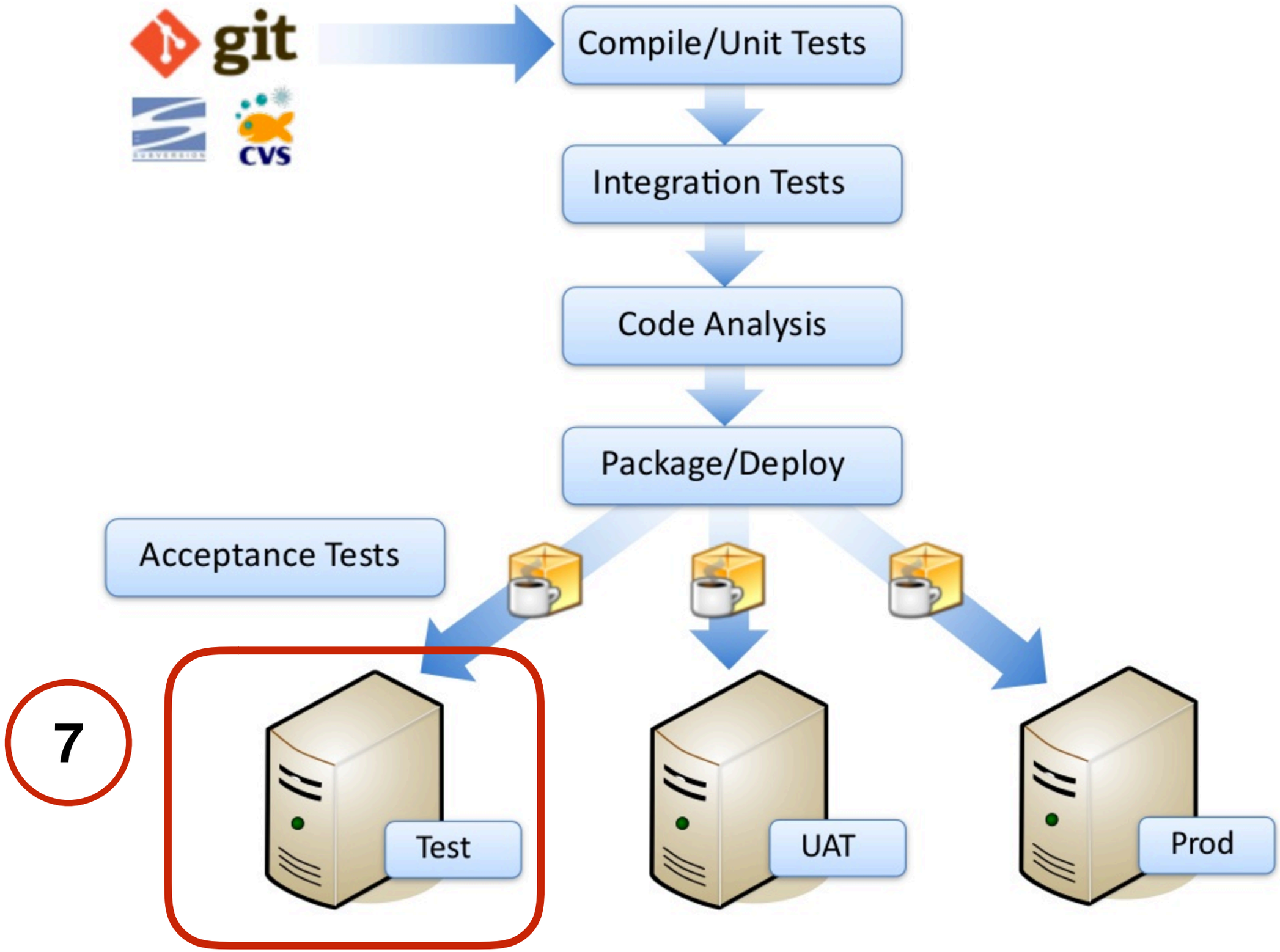

git
CVS
Compile/Unit Tests
Integration Tests
Code Analysis
Package/Deploy
3
Acceptance Tests
Test
UAT
Prod

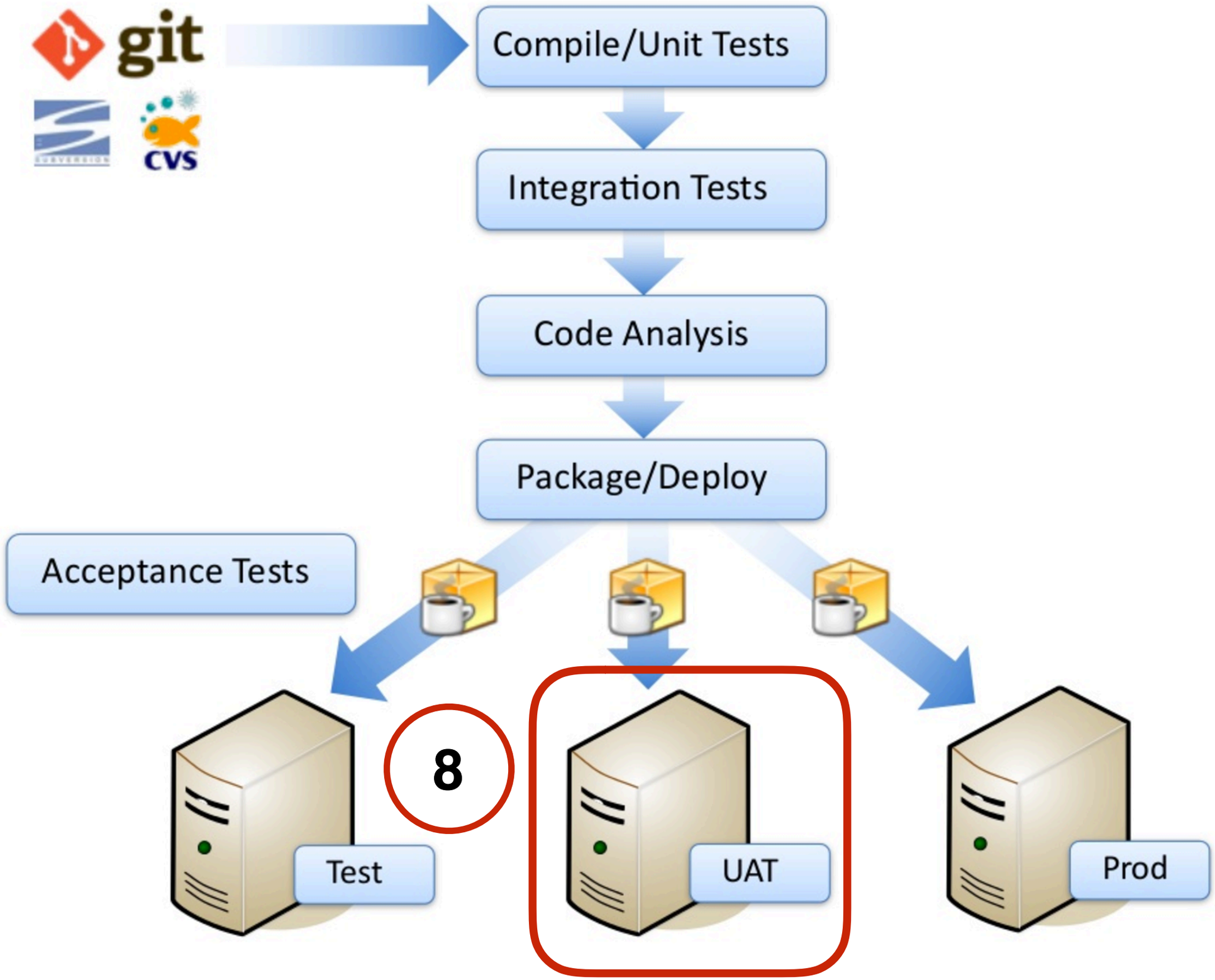

git
SVN
CVS
Compile/Unit Tests
Integration Tests
Code Analysis
Package/Deploy
4
Acceptance Tests
Test
UAT
Prod

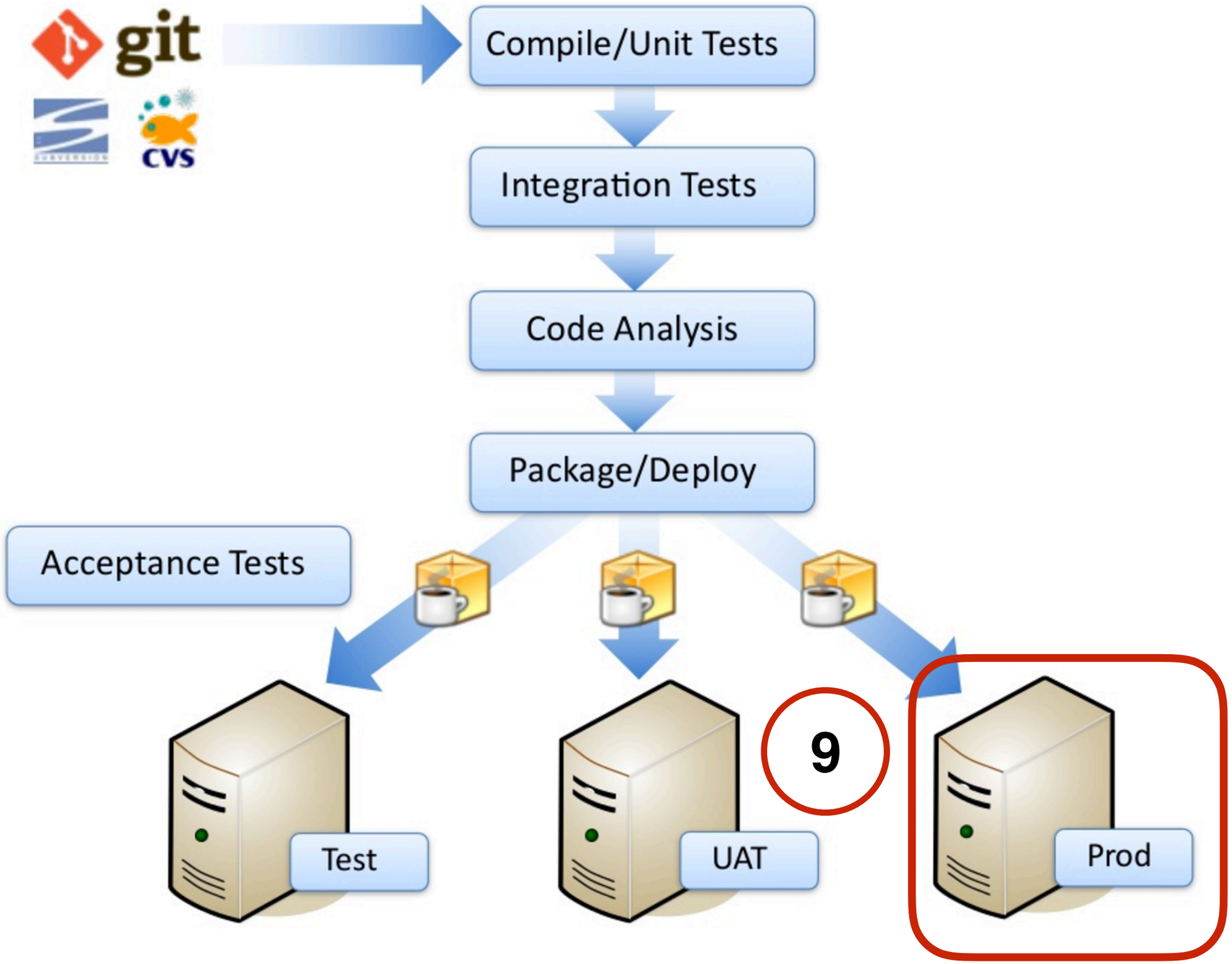

git
CVS
Compile/Unit Tests
Integration Tests
Code Analysis
Package/Deploy
5
Acceptance Tests
Test
UAT
Prod


Compile/Unit Tests
Integration Tests
Code Analysis
Package/Deploy
Acceptance Tests
Test
UAT
Prod

6


git
CVS
Compile/Unit Tests
Integration Tests
Code Analysis
Package/Deploy
Acceptance Tests
7
Test
UAT
Prod


git
CVS
Compile/Unit Tests
Integration Tests
Code Analysis
Package/Deploy
Acceptance Tests
8
Test
UAT
Prod


git
CVS
Compile/Unit Tests
Integration Tests
Code Analysis
Package/Deploy
Acceptance Tests
Test
UAT
Prod
9

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: robot

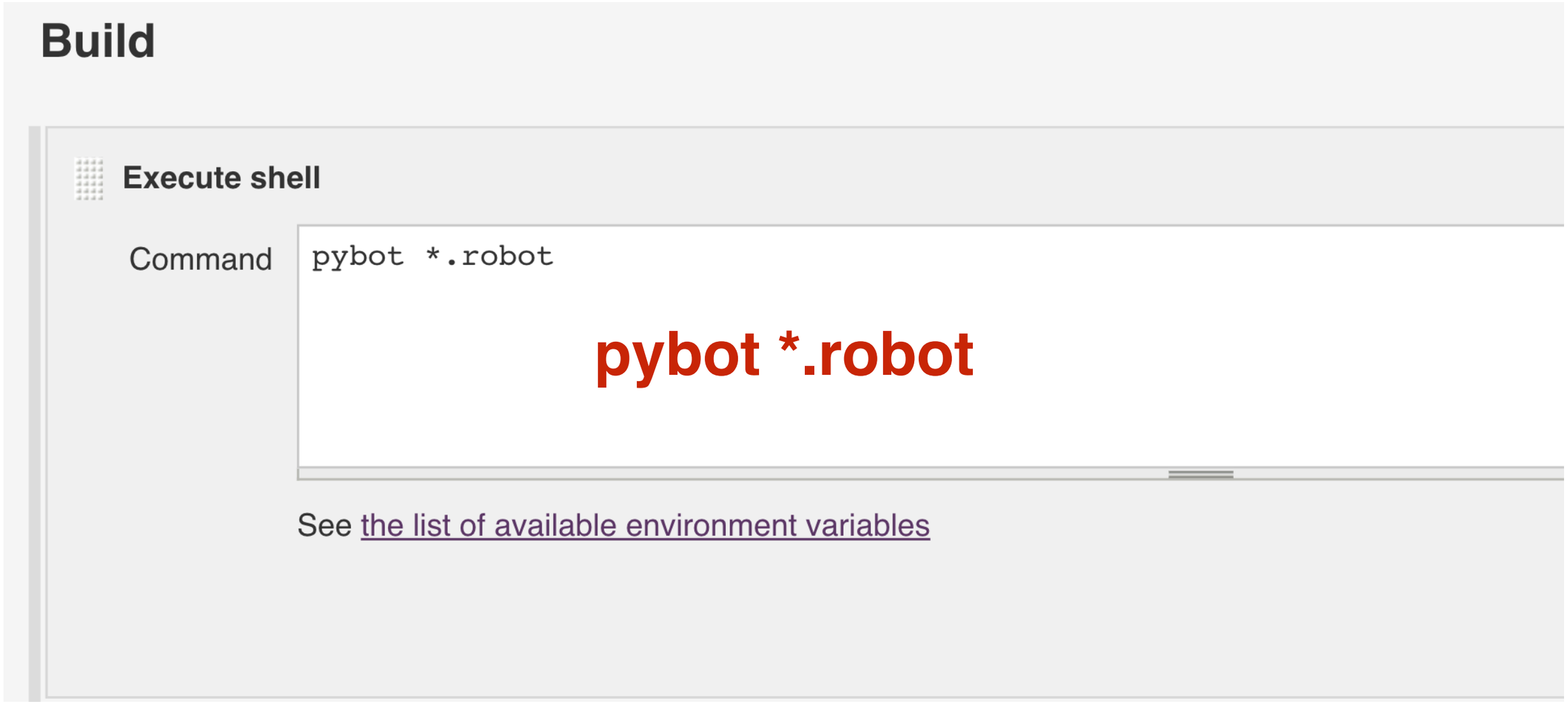
Updates | **Available** | Installed | Advanced

Install ↓	Name	Version
☐	Robot Framework Shows Robot Framework test results in project	1.6.4

Install without restart | Download now and install after restart

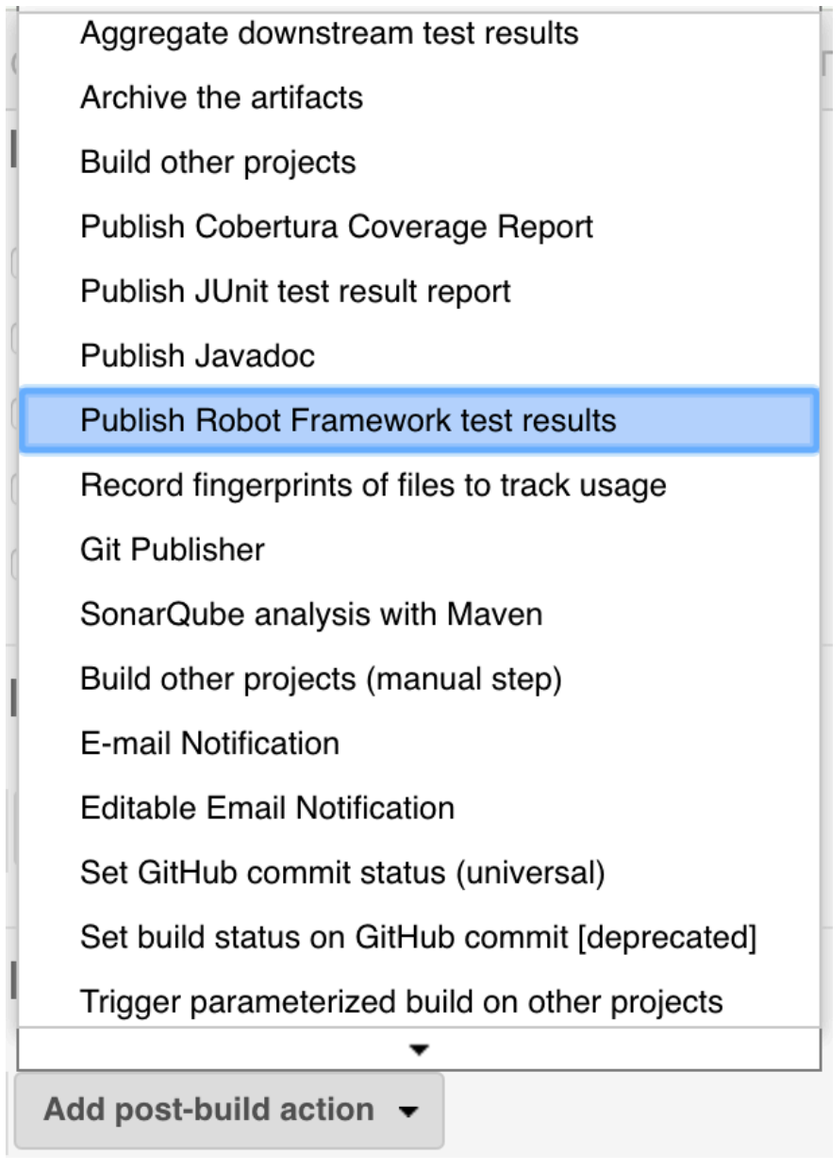
Update information obtained:

Add build step to run with Robot



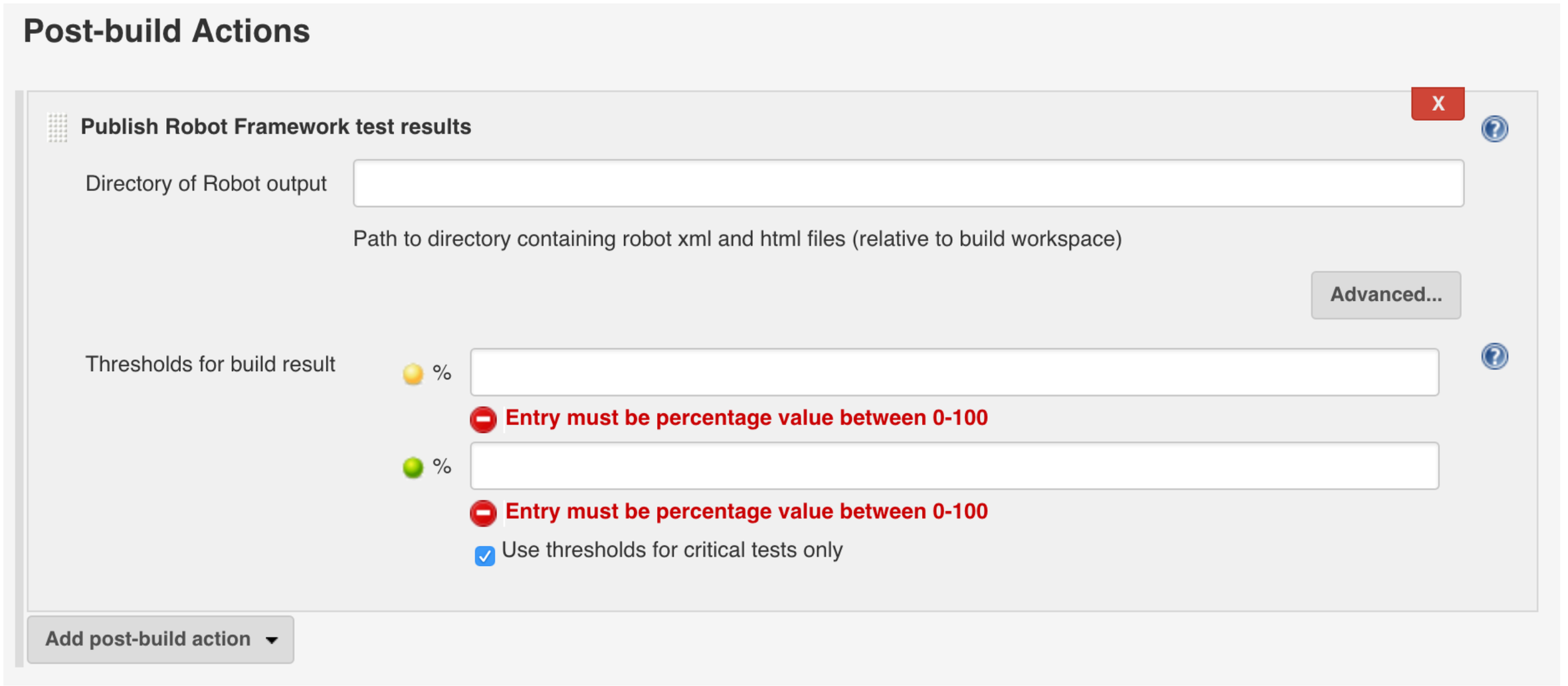
Add Robotframework report

Add post build action =>Publish Robot Framework



Add Robotframework report

Configuration your report



Report

add description

Disable Project

Workspace

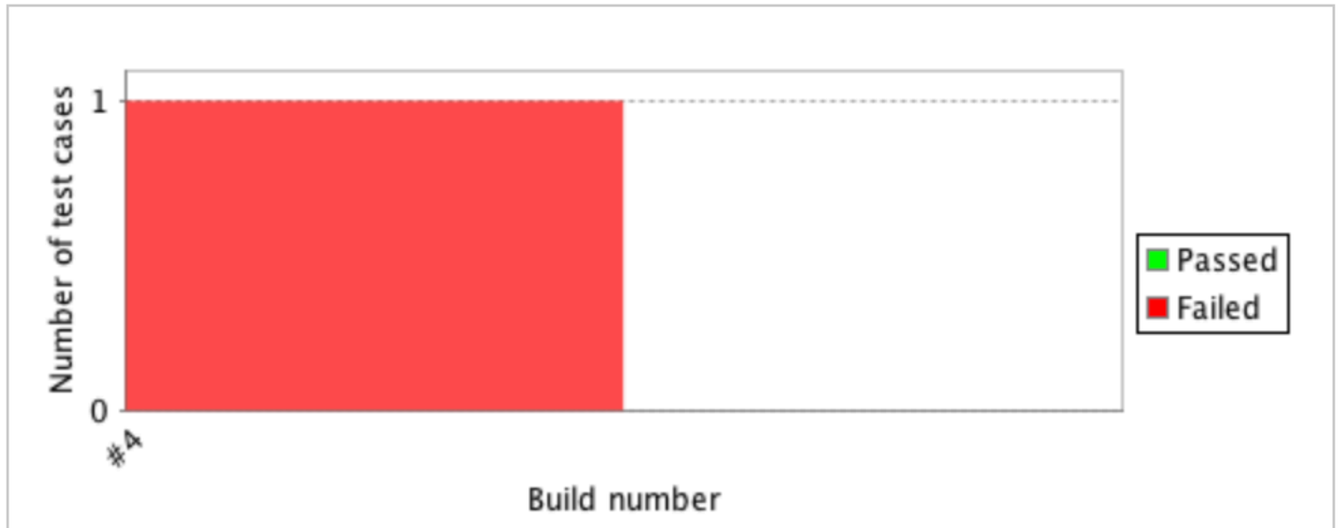
Recent Changes

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)



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

Show bigger image