

CI/CD



Contents

Terminology

Deployment process

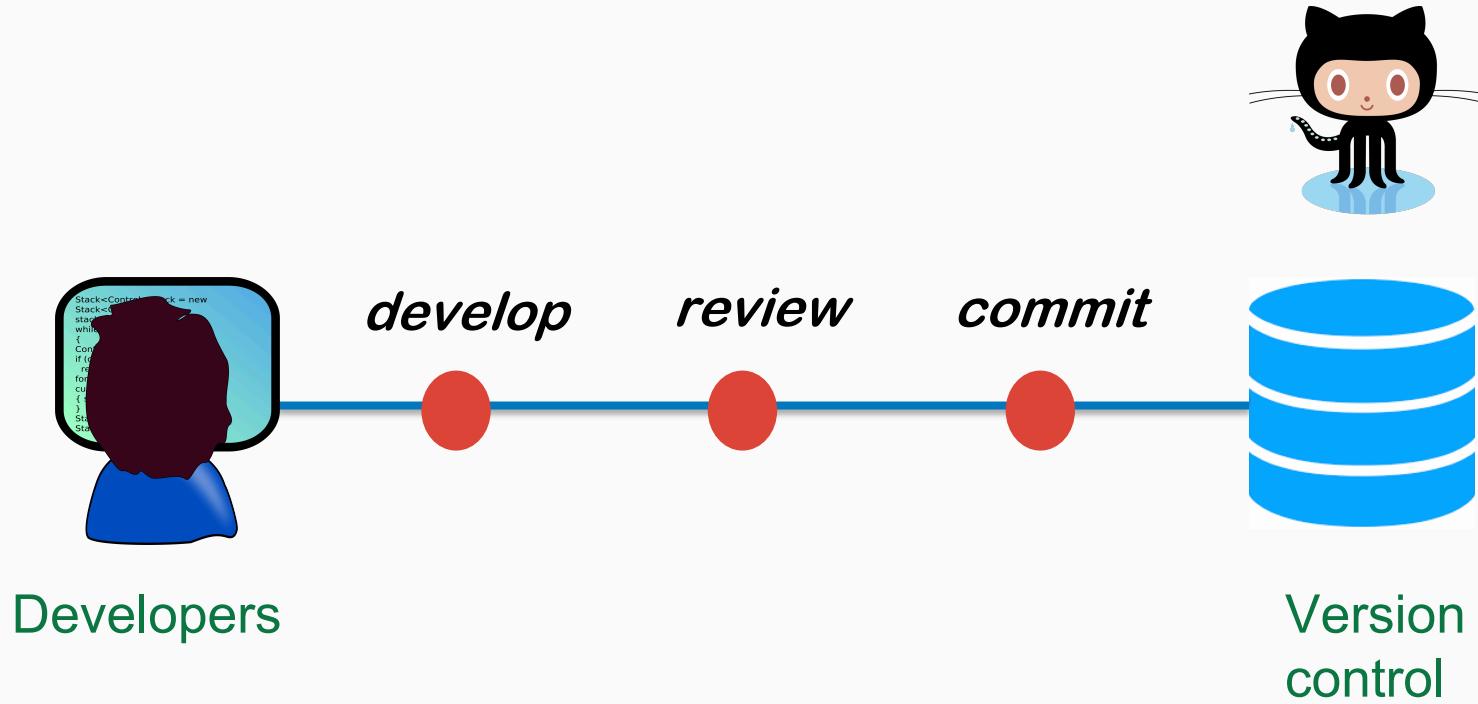
Pipeline example

Automation in pipeline

Jenkins

Contents

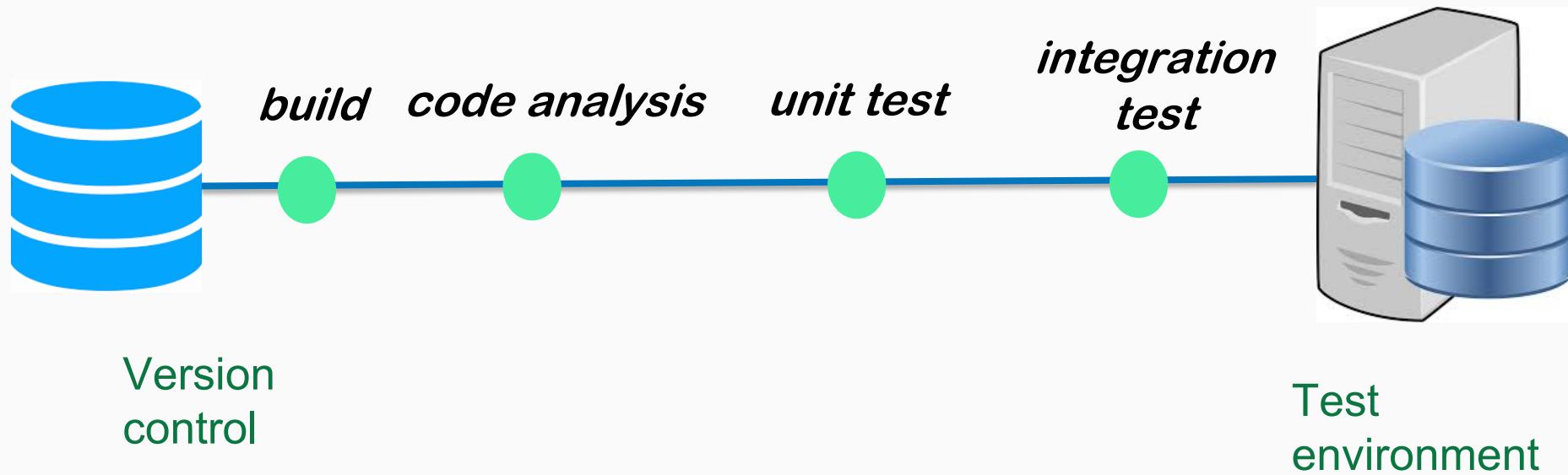


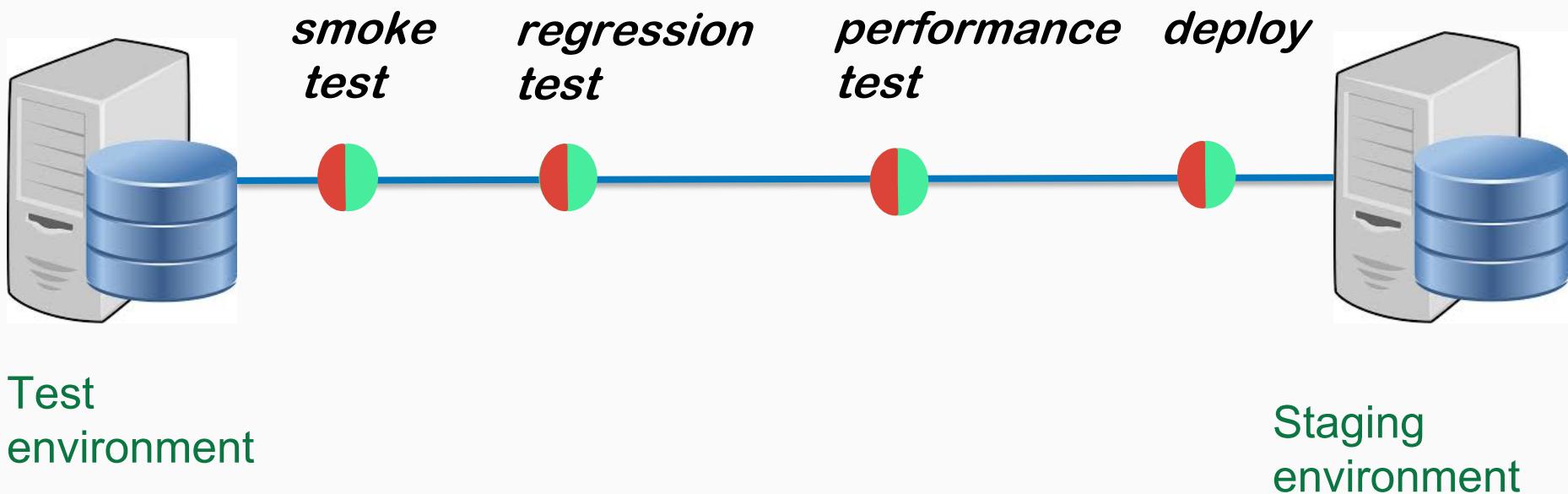


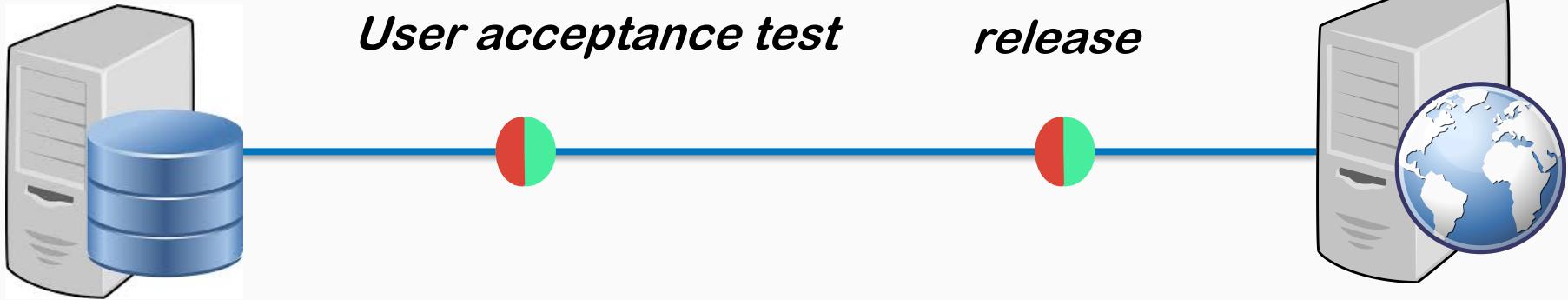


sonarqube

JUnit

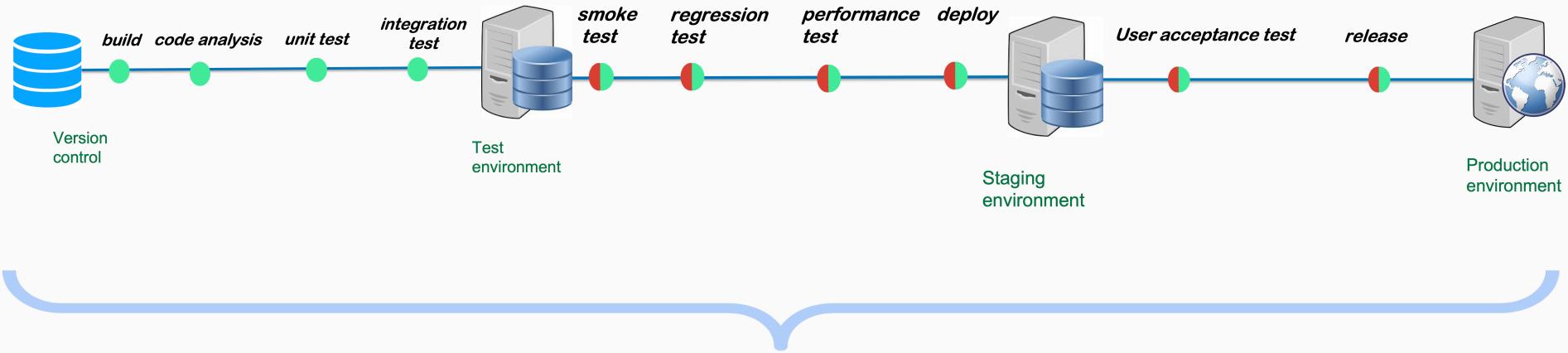






Test
environment

Production
environment

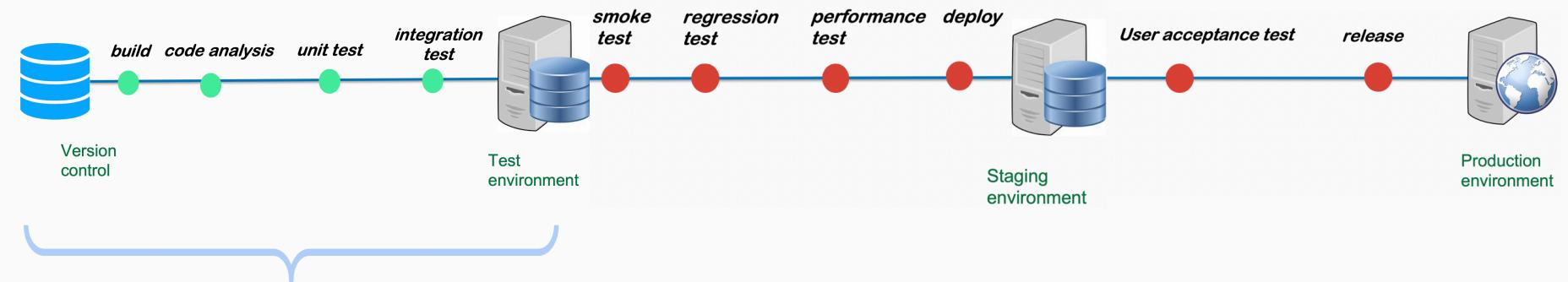


Production Pipeline

Pipeline

Pipeline is a set of processes that take the code from version control and compile, build, test and deploy to production in automated fashion.

The pipeline breaks down the software delivery process into stages. Each stage is made of different tasks which can be carried out in parallel. When all tasks in a stage passes, next stage is triggered.

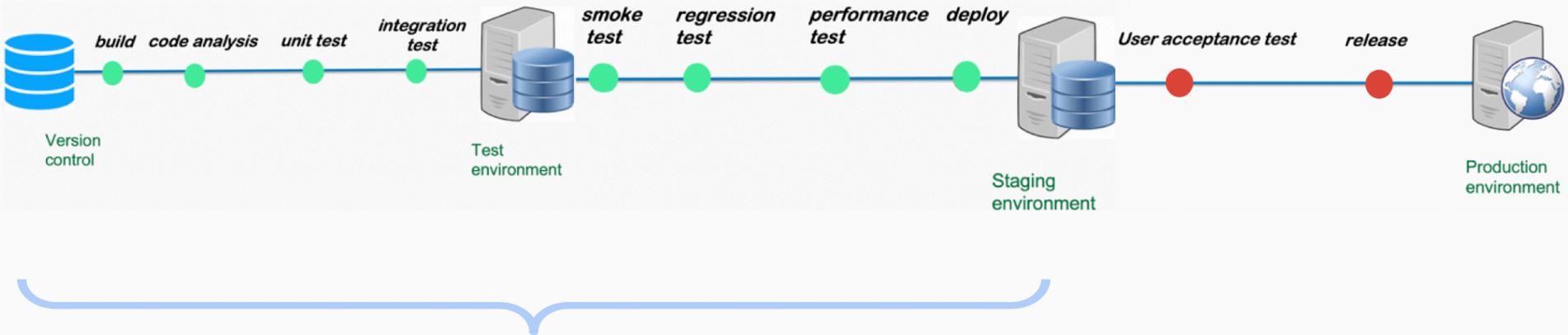


Continuous Integration

Continuous Integration

Continuous Integration is an automated build and execution of at unit and integration tests, performing code analysis.

The Continuous Integration process is comprised of automatic tools that assert the new code's correctness before integration. It reduces integration problems allowing to deliver software more rapidly by providing quick feedback every time new code is added to the source control. Usually Continuous Integration does not involve testing the functionality of the application.

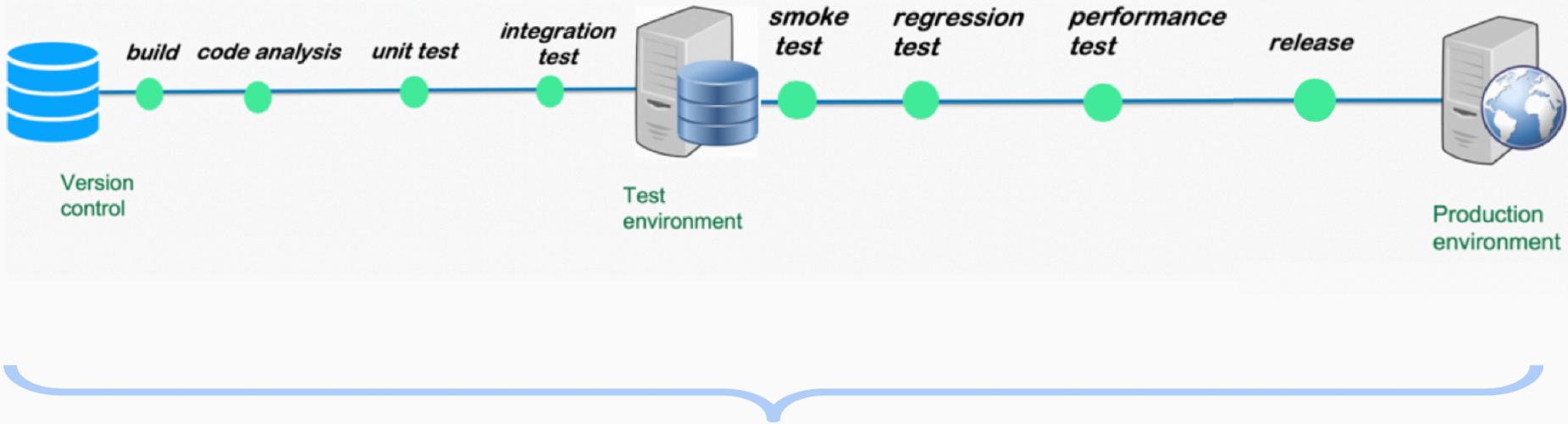


Continuous Delivery

Continuous Delivery

Continuous Delivery is an automated build and execution of unit and integration tests, performing code analysis, functional tests and also deploying to any supported platform any time. Each time a build or a set of code passes the tests, it's automatically deployed out to a staging environment.

In Continuous Delivery releasing to end users is a manual process. Continuous delivery involves human decision-making when it comes to deciding when to release the software to the customers.



Continuous Deployment

Continuous Deployment

Continuous Deployment

Continuous deployment means that every change that you make, goes through the pipeline, and if it passes all the tests, it automatically gets deployed into production.

When a developer checks in code, the automated processes take the code and move it through the entire lifecycle and if it passes each gate, it gets deployed directly to production. The delivery speeds are notably faster due to elimination of manual steps.

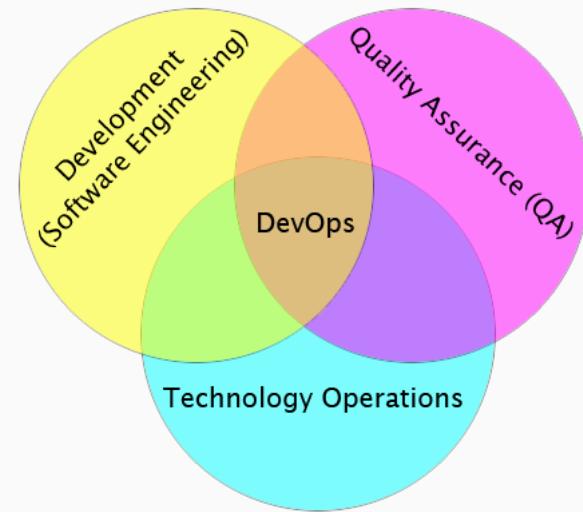
DevOps

Methodology (like agile, waterfall)

DevOps merges developer, tester and operations roles together. whoever writes the software is also responsible and deploying and maintaining it.

DevOps organizations break down the barriers between Operations and Engineering by cross-training each team in the other's skills.

DevOps focuses on culture highlighting roles that emphasize responsiveness and breaking down barriers between developers and operations teams.



Jenkins



1. CD/CD tool
2. Travis, TeamCity, Bamboo
3. Open source
4. Automates building, testing, packaging, staging, deploying the application

Integrates with different tools using plugins

Jenkins job

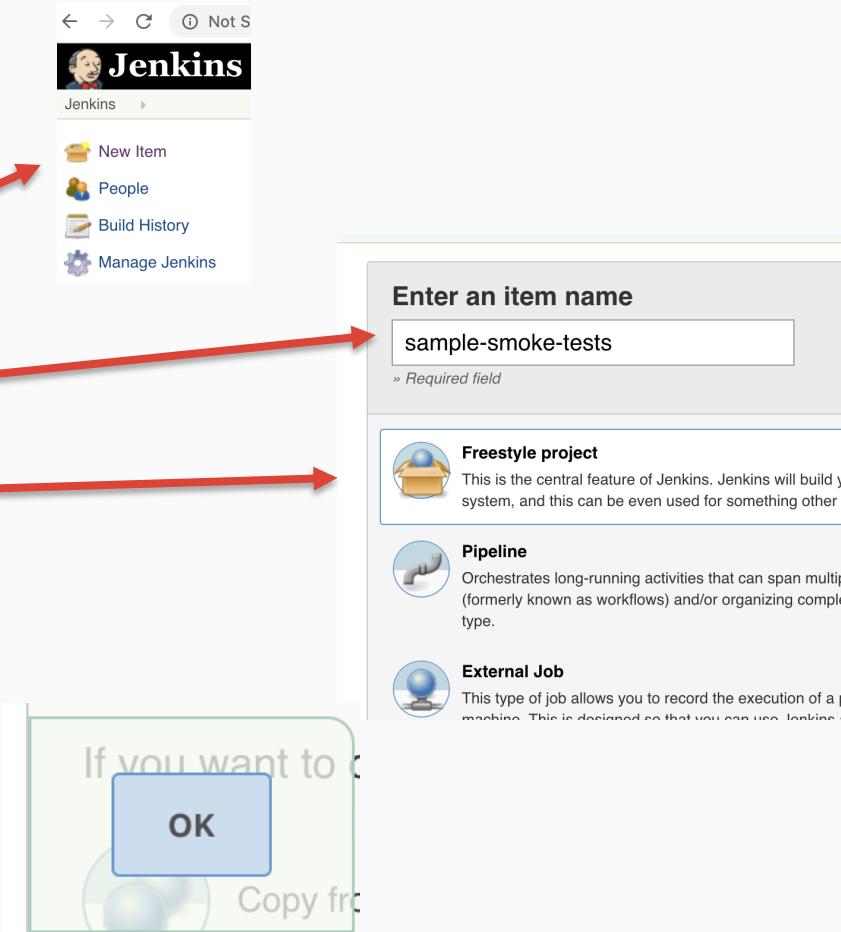


In Jenkins everything is done by creating a job

1. a task that Jenkins performs based its schedule
2. be made of several steps
3. can have a schedule or a trigger which determines when it runs
4. reports the results of the run automatically

Create new job on jenkins

1. Click on New Item on Jenkins home page
2. Enter name
3. Select freestyle project
4. Click OK



Enter source code information

1. Click on Source Code Management. You also scroll down on the page

2. Select Git

3. Enter your git repository url information

4. Select the branch to build

NOTE: For private repositories you will have to Credentials

Credentials - none - Add

The screenshot shows the Jenkins configuration interface for a Jenkins job. The top navigation bar has tabs: General, Source Code Management (which is selected and highlighted in black), Build Triggers, Build Environment, and Build. The main content area is titled "Source Code Management". Under "Source Code Management", the "Git" option is selected. The "Repositories" section contains one repository entry. The "Repository URL" field is populated with "https://github.com/CybertekSchool/demo-project.git". Below it, the "Credentials" dropdown is set to "- none -" and there is an "Add" button. To the right of the repository entry are "Advanced..." and "Add Repository" buttons. The "Branches to build" section contains one branch entry. The "Branch Specifier (blank for 'any')" field is populated with "*/*master". Below it is an "Add Branch" button. At the bottom of the configuration area, there is a "Repository browser" dropdown set to "(Auto)".

Build triggers

1. Click on Build triggers. You also scroll down on the page
2. Select Build periodically
3. Enter job schedule

The screenshot shows the Jenkins build triggers configuration screen. At the top, there are tabs: General, Source Code Management, **Build Triggers**, Build Environment, and Build. A red arrow points from the text "Click on Build triggers" to the **Build Triggers** tab. Another red arrow points from the text "Select Build periodically" to the "Build periodically" checkbox. A third red arrow points from the text "Enter job schedule" to the "Schedule" field. The "Build periodically" checkbox is checked. Below it, the "Schedule" field contains the value "H 5 * * *". A tooltip for the schedule field states: "Would last have run at Wednesday, April 29, 2020 5:00:00 AM".

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

Trigger builds remotely (e.g., from scripts)

Build after other projects are built

Build periodically

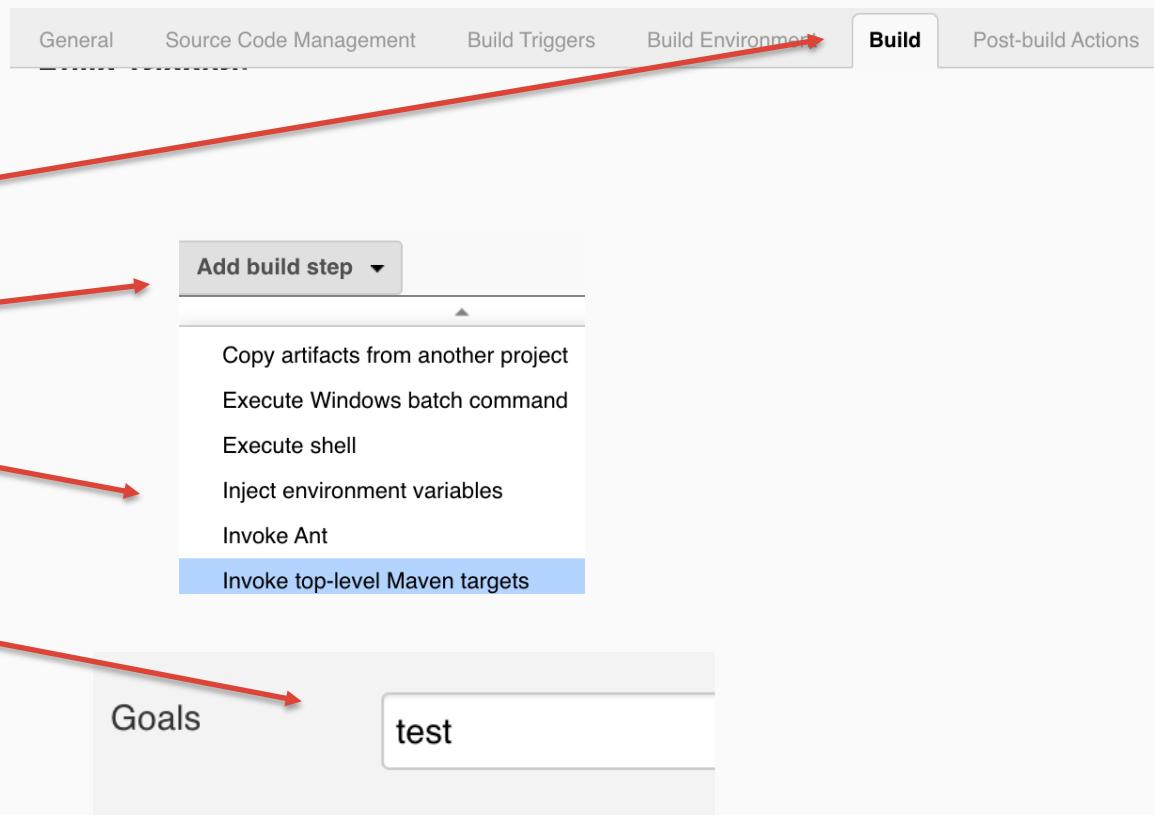
Schedule

H 5 * * *

Would last have run at Wednesday, April 29, 2020 5:00:00 AM

Build

1. Click on Build. You also scroll down on the page
2. Click on Add build step
3. Select Invoke top-level Maven targets
4. Enter maven goal that executes your test



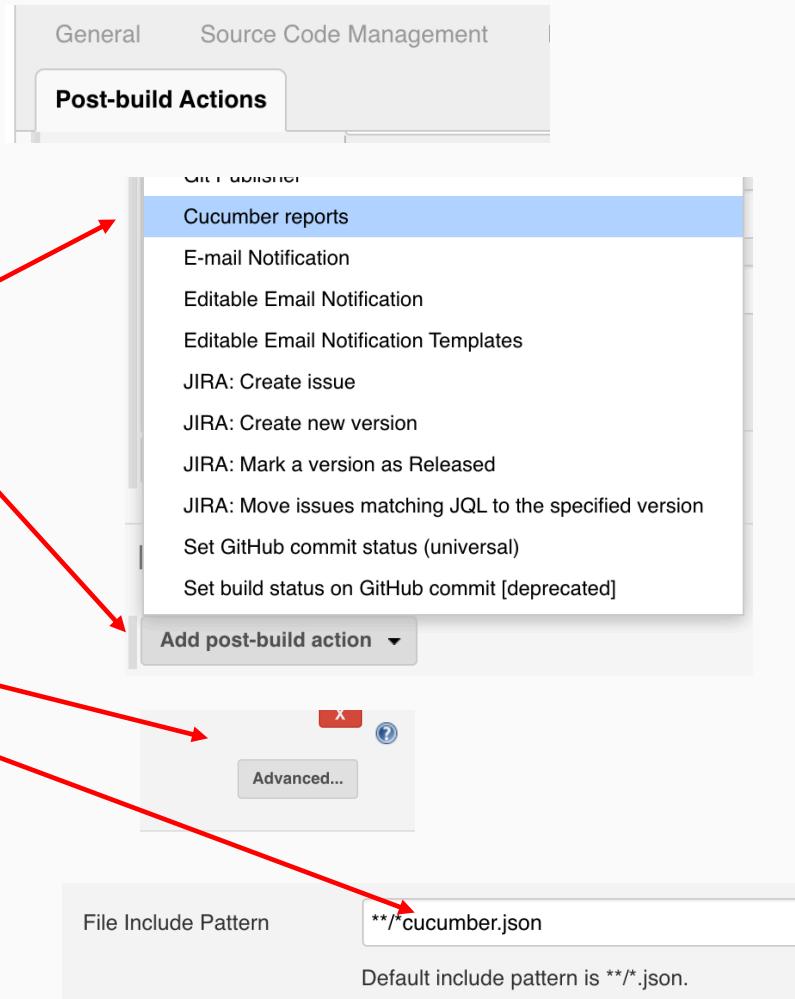
Cucumber reports

1. Click on Post-build Actions. You also scroll down on the page
2. Click on Add post-build action
3. Select Cucumber reports.
4. OPTIONAL STEPS. By default customize the path to json reports

- 4.1 Click on Advanced

- 4.2 Show path to cucumber json file

NOTE: Cucumber reports is a 3rd party plugin. If you do not see this plugin, if mean is not installed. Go to Jenkins plugin management to install



Install slack plugins

1. Go to Jenkins → Manage Jenkins → Manage Plugins

The screenshot shows the Jenkins 'Manage Plugins' page. At the top, there are tabs: 'Updates' (highlighted with a red arrow), 'Available', 'Installed', and 'Advanced'. Below these tabs is a search bar with the placeholder 'Filter:' containing the text 'slack'. A red arrow points from the search bar to the filter input. In the main list area, a modal window is open for the 'Slack Notification' plugin. The modal title is 'Slack Notification' and it contains the text: 'This plugin is a Slack notifier that can publish build status to Slack channels.' At the bottom of the modal, there is a large button with the text 'Download now and install after restart'.

2. Click Available

3. Enter search filter slack

4. Select plugin Slack Notification

5. Click on Download now and install after restart Update

6. Wait until Jenkins restarts

Jenkins link fix

1. Go to Jenkins → Manage Jenkins → Configure System
2. Scroll down to Jenkins Location
3. Copy the IP address and port number of your EC2 instance
4. Paste the value to the Jenkins URL field



Jenkins Location

Jenkins URL
System Admin e-mail address

A screenshot of the Jenkins configuration page under 'Administrative monitors configuration'. It shows the 'Jenkins Location' section with 'Jenkins URL' set to 'http://100.27.23.190:8081/'. A red arrow points from the text 'Paste the value to the Jenkins URL field' to this input field. Above the URL field, the text 'must be the same' is displayed in bold. The browser's address bar at the top also shows the URL '100.27.23.190:8081/configure'.

must be the same

Administrative monitors configuration

Jenkins Location

Jenkins URL
System Admin e-mail address

Lockable Resources Manager

Lockable Resources
Add Lockable Resource

GitHub

Github Servers
Add Github Server

Slack configuration 1

1. Go to Jenkins → Manage Jenkins → Configure System

2. Scroll down to Slack

3. Click Add → Jenkins

4. Select Kind → Secret text

5. Enter secret

6. Select ID

7. Click Add



Slack

- none - Add Jenkins

Domain: Global credentials (unrestricted)

Kind: Secret text

Scope: Global (Jenkins, nodes, items, all child)

Secret: (redacted)

ID: slack-token-b15

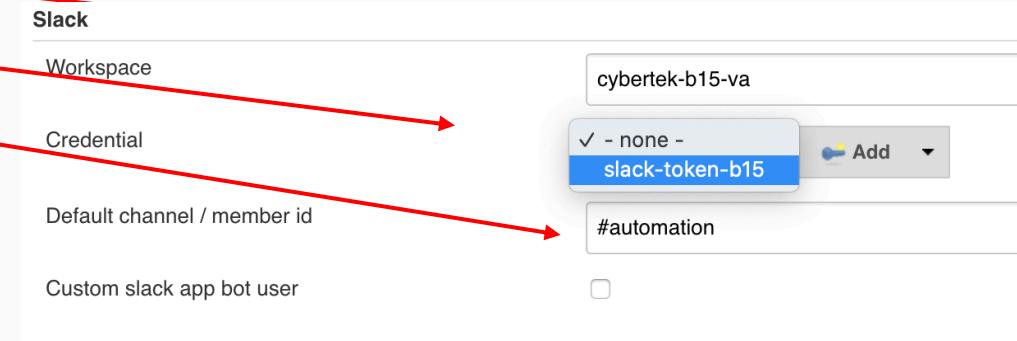
Description:

Add Cancel

A screenshot of the Jenkins 'Add Credential' dialog for Slack. The dialog has tabs for 'Domain' (set to 'Global credentials (unrestricted)') and 'Kind' (set to 'Secret text'). Under 'Scope', it says 'Global (Jenkins, nodes, items, all child)'. The 'Secret' field contains '.....' (redacted). The 'ID' field contains 'slack-token-b15'. There is a 'Description' field which is empty. At the bottom are 'Add' and 'Cancel' buttons. Red arrows from the numbered steps on the left point to each of these fields and buttons.

Slack configuration 2

1. Enter workspace
2. Select none → slack token
3. Select Default channel/member id



Add Slack notification

1. Click on Add post build action
2. Select Slack Notifications
3. Select All options that apply

The screenshot shows the Jenkins interface for adding a post-build action. A red arrow points from the first step to the 'Add post-build action' button. Another red arrow points from the second step to the 'Slack Notifications' option in the dropdown menu. A third red arrow points from the third step to the list of notification options below.

Slack Notifications

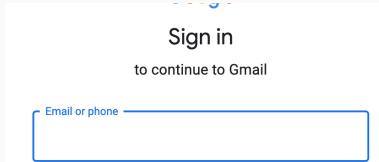
Add post-build action ▾

Slack Notifications

Notify Build Start	<input checked="" type="checkbox"/>
Notify Success	<input checked="" type="checkbox"/>
Notify Aborted	<input checked="" type="checkbox"/>
Notify Not Built	<input checked="" type="checkbox"/>
Notify Unstable	<input checked="" type="checkbox"/>
Notify Regression	<input checked="" type="checkbox"/>
Notify Every Failure	<input checked="" type="checkbox"/>
Notify Back To Normal	<input checked="" type="checkbox"/>

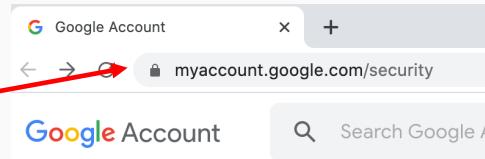
Configure security settings on Gmail 1

1. Login to your email account



2. Go to link

<https://myaccount.google.com/security>



3. Scroll down to Less secure app access



Less secure app access

To protect your account, apps and devices using less secure technology are blocked. To keep your account safe, Google automatically turn this setting OFF if it's no longer needed.

Off

[Turn on access \(not recommended\)](#)

4. Click on Turn on access



5. Change Allow less secure apps to ON



Allow less secure apps: ON



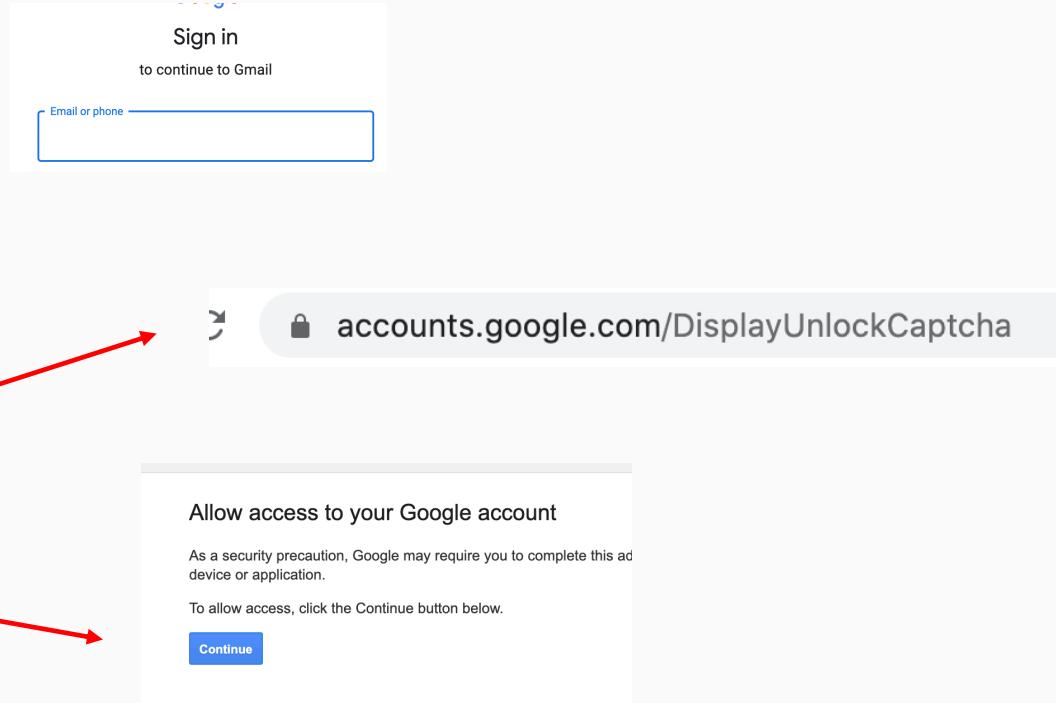
Configure security settings on Gmail 2

1. Login to your email account

2. Go to link

<https://www.google.com/accounts/DisplayUnlockCaptcha>

3. Click Continue



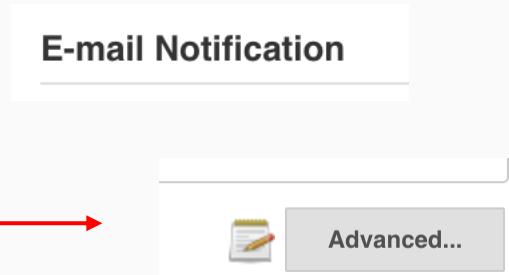
Email settings configuration (gmail specific) 1

1. Go to Jenkins → Manage Jenkins
→ Configure System

2. Scroll down to Email Notification

3. Click on Advanced

4. Enter your email username and
password



User Name	jenkins@cybertekschool.com
Password

Email settings configuration (gmail specific) 2

1. Go to Jenkins → Manage Jenkins → Configure System
2. Scroll down to Extended E-mail Notification
3. Click on Advanced
4. Enter your email username and password



Extended E-mail Notification

Advanced...

User Name: jenkins@cybertekschool.com

Password: [REDACTED]

A screenshot of the 'Extended E-mail Notification' configuration page. It shows a 'User Name' field containing 'jenkins@cybertekschool.com' and a 'Password' field containing a redacted password. A red arrow points from the list item 4 to the 'User Name' field.

Configure Email notification 3

1. Click on Post-build Actions. You also scroll down on the page
2. Click on Add post-build action
3. Select Editable E-mail notification
4. Enter receipt email

The screenshot shows the Jenkins configuration interface for a project. The top navigation bar includes tabs for General and Source Code Management, with Post-build Actions selected. A dropdown menu titled 'Add post-build action' is open, listing various actions. The 'Editable Email Notification' option is highlighted with a blue selection bar. Below the dropdown, the 'Project Recipient List' field contains the email addresses 'john.doe@someemail.com, jane.doe@someemail.com'. Red arrows from the numbered steps on the left point to the 'Post-build Actions' tab, the 'Add post-build action' dropdown, and the recipient list field respectively.

General Source Code Management

Post-build Actions

Editable Email Notification

Editable Email Notification Templates

JIRA: Create issue

JIRA: Create new version

JIRA: Mark a version as Released

JIRA: Move issues matching JQL to t

Set GitHub commit status (universal)

Set build status on GitHub commit [d

Slack Notifications

Add post-build action ▾

Project Recipient List

john.doe@someemail.com, jane.doe@someemail.com

Configure Email notification 4

1. Enter receipt email
2. Click on advanced settings
3. Select Add Trigger → Always

