Jenkins Primer

Continuous Integration/Development with Jenkins

Delivered by: Noureddin Sadawi, PhD

About the Speaker

Name: Dr Noureddin Sadawi

Qualification: PhD Computer Science

Experience: Several areas including but not limited to Docker, Machine Learning and Data Science, Python and much more

Definitions

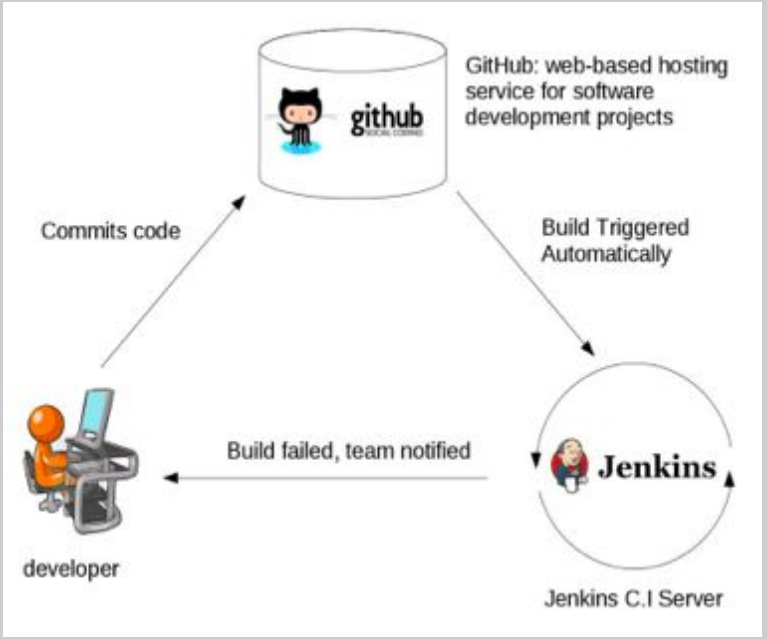
- **Continuous Integration:**
  - The practice of merging development work with the main branch constantly
- **Continuous Delivery:**
  - Continuous delivery of code to an environment once the code is ready to ship
  - This could be staging or production. The idea is the product is delivered to a user base, which can be QAs or customers for review and inspection
- **Continuous Deployment:**
  - The deployment or release of code to production as soon as it is ready

Jenkins

- Continuous integration systems are a vital part of any Agile team because they help enforce the ideals of Agile development
- Jenkins, a continuous build tool, enables teams to focus on their work by automating the build, artifact management, and deployment processes
- Jenkins' core functionality and flexibility allow it to fit in a variety of environments and can help streamline the development process for all stakeholders involved

What is Continuous Integration

- Developers commit code to a shared repository on a regular basis
- Version control system is being monitored. When a commit is detected, a build will be triggered automatically
- If the build is not successful, developers will be notified immediately

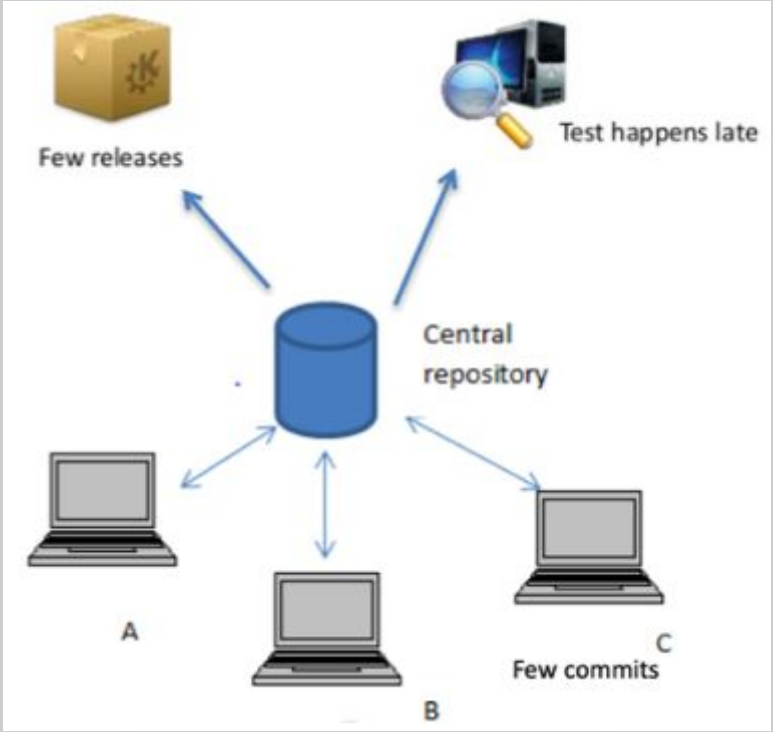


Why Do we Need it?

- Detect problems or bugs, as early as possible, in the development life cycle
- Since the entire code base is integrated, built and tested constantly, the potential bugs and errors are caught earlier in the life cycle which results in better quality software

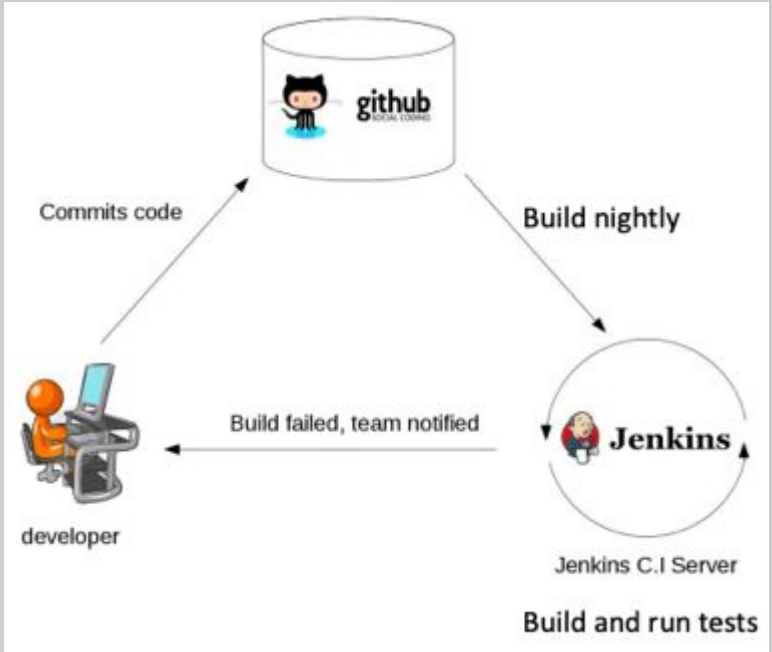
No CI

- No build servers
- Developers commit on a regular basis
- Changes are integrated and tested manually
- Fewer releases



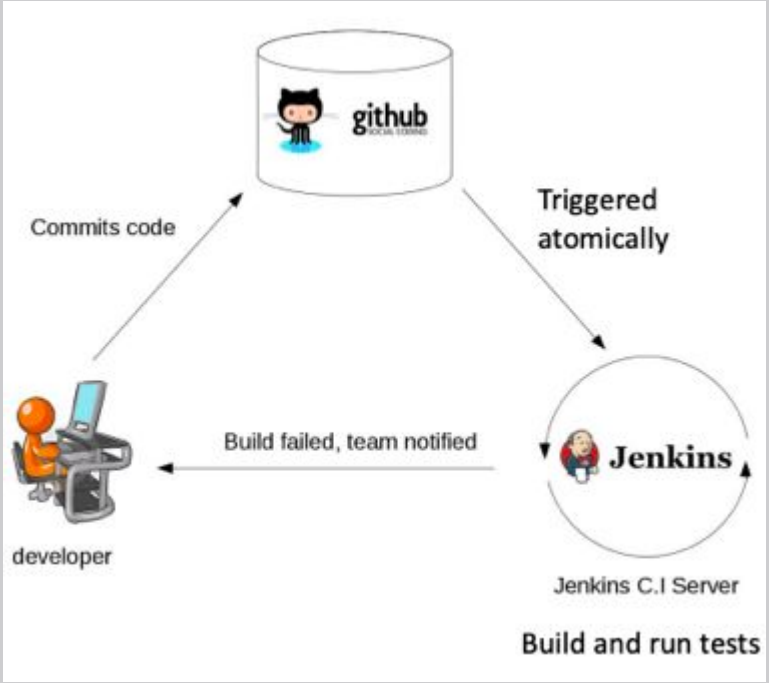
Scheduled Builds

- Automated builds are scheduled on a regular basis
- Build script compiles the application and runs a set of automated tests
- Developers now commit their changes regularly
- Build servers would alert the team members in case of build failure



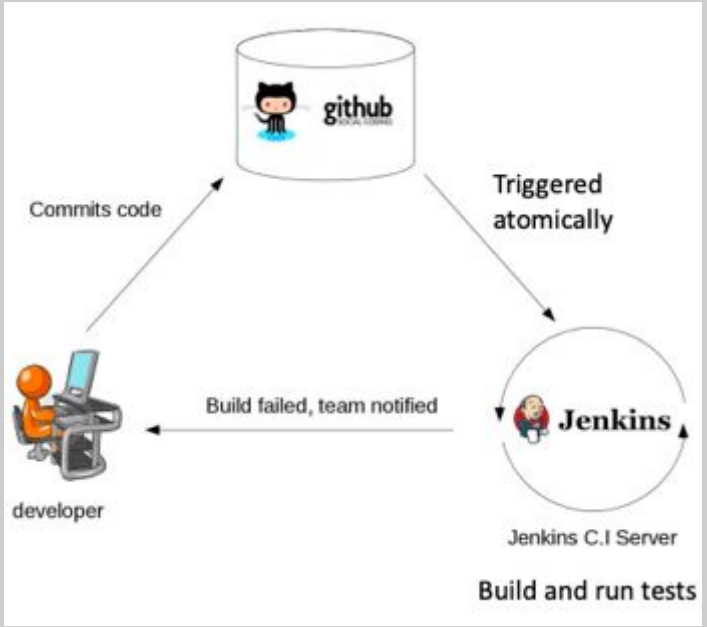
Build when Code is Committed 1/2

- A build is triggered whenever new code is committed to the central repository
- Broken builds are usually treated as a high priority issue and are fixed quickly



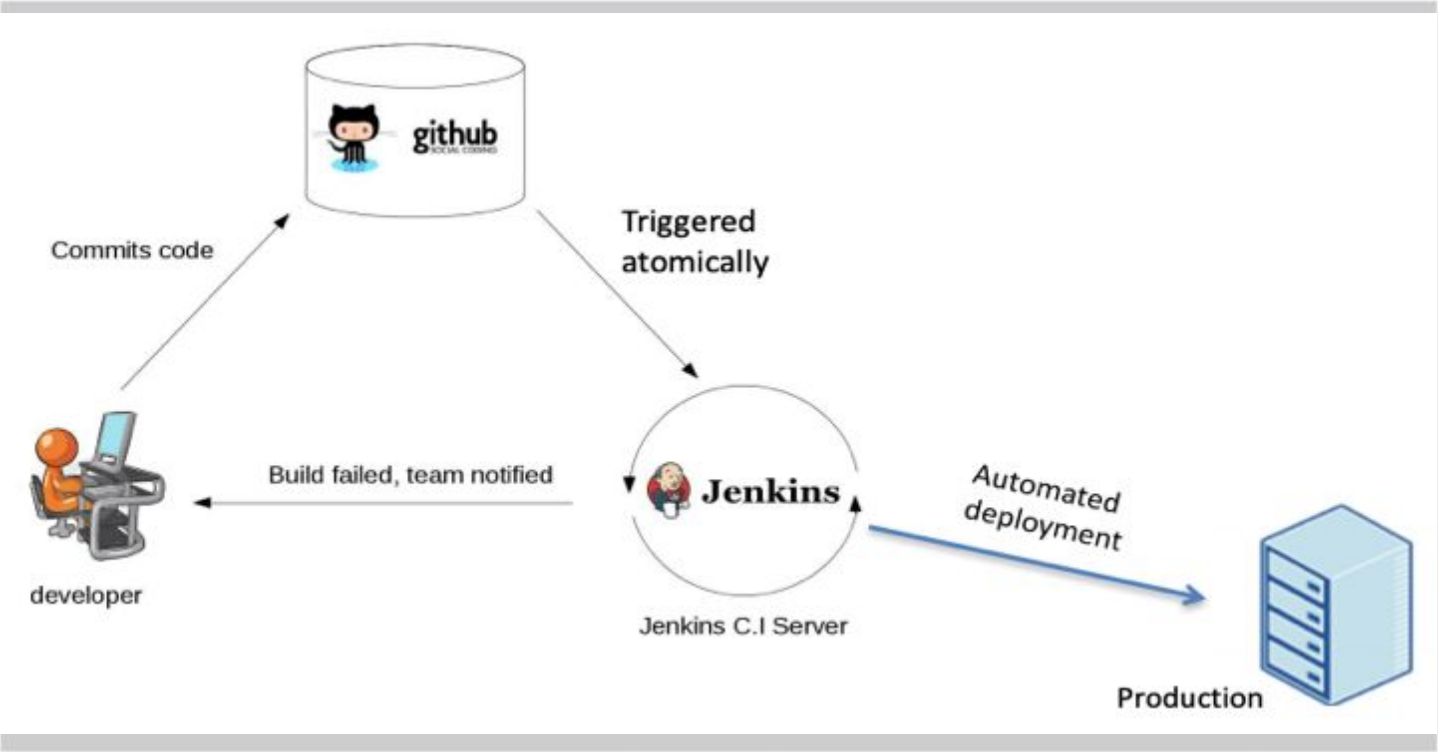
Build when Code is Committed 2/2

- Automated code quality and code coverage metrics are now run along with unit tests to continuously evaluate the code quality
- Is the code coverage increasing?
- Do we have fewer and fewer build failures?



Build, run code quality and code coverage metrics along with tests

Automated Deployment



Jenkins

- Jenkins is a continuous integration and build server
- It is used to manually, periodically, or automatically build software development projects
- It is an open source Continuous Integration tool written in Java
- Jenkins is used by teams of all different sizes, for projects with various languages

Why Jenkins?

- Easy to use
- Great extensibility (high configurability)
  - Support different version control systems
  - Code quality metrics
  - Build notifiers
  - UI customization
- There is a large support community and thorough documentation
- It is easy to write plugins

Jenkins Online Book

This is a helpful resource:

https://www.jenkins.io/doc/book/

Jenkins behind Proxy

More details here:

https://wiki.jenkins.io/display/JENKINS/JenkinsBehindProxy

Cucumber

- Cucumber is a test automation tool following the principles of Behavioural Driven Design and living documentation (BDD)
- Specifications are written in a concise human readable form and executed in continuous integration
- Have a look at this project to generate pretty HTML reports: https://github.com/damianszczepanik/cucumber-reporting

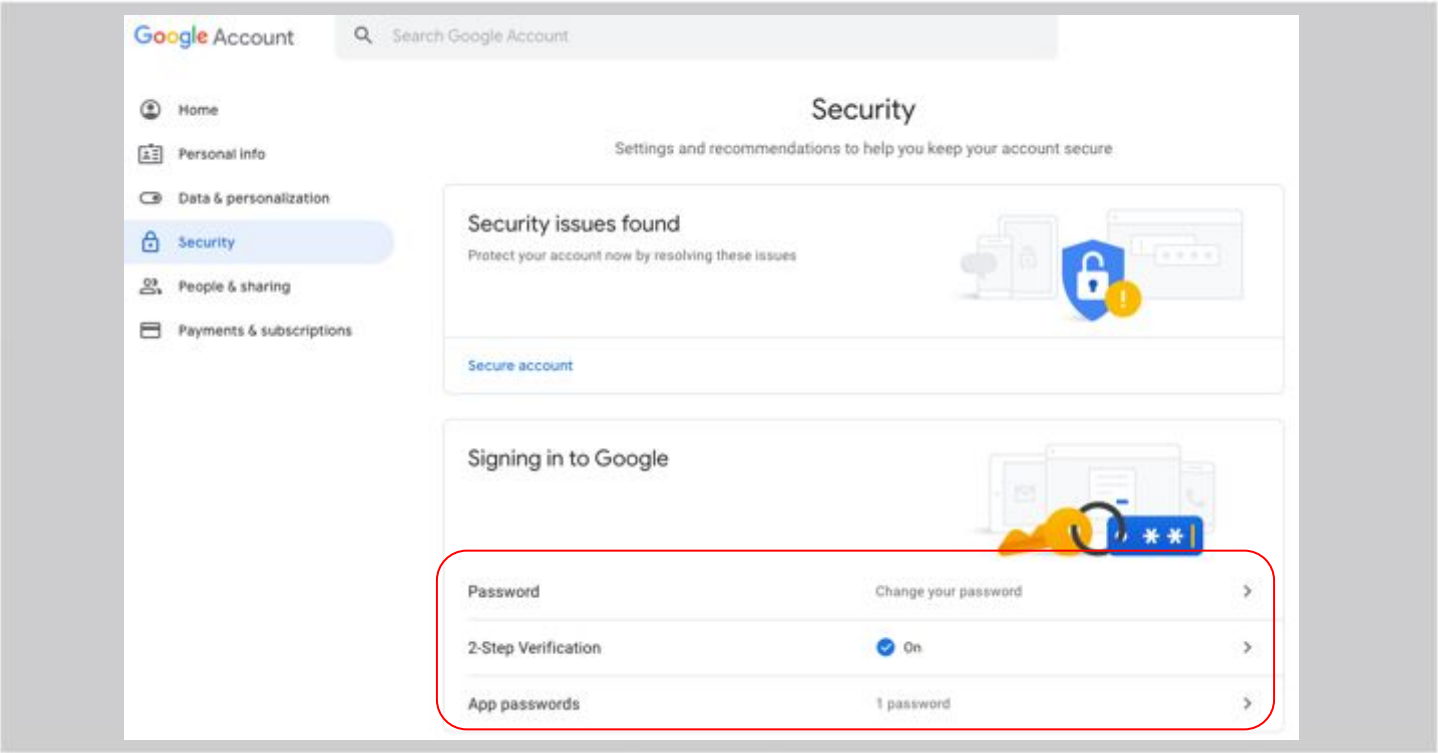
Video Links

- Windows 10 install java 8: https://youtu.be/4x0RnjaPb4g
- Start Jenkins on Windows 10: https://youtu.be/S6nNSQhkUsw
- Install Chocolatey on Windows 10: https://youtu.be/EMboJJTTqZY
- Install .NET Framework version 4.6.1 and MSBuild (using Chocolatey) on Windows 10: https://youtu.be/LeBx1v4lbYw
- Install Git, NUnit and NuGet on Windows 10 using Chocolatey: https://youtu.be/9jO-U1XcSIs
- Install and Configure MSBuild plugin on Jenkins (Please notice no need for the “” around the MSBuild path on Jenkins): https://youtu.be/ps8JAZN-csM

Sending Emails (via GMail)

- Create an App password
- Go to your Google Account.
- On the left navigation panel, choose Security (provide your password if needed).
- Setup 2-Step Verification if you do not have it
- On the 'Signing in to Google' panel, choose **App passwords**. ...
- At the bottom, choose Select **app** and choose the **app** that you're using.
- Choose Select device and choose the device that you're using.
- Choose Generate.

Sending Emails (via GMail)



Sending Emails .. On Jenkins

- Go to Manage Jenkins -> Configure System -> E-mail Notification
  - SMTP server = smtp.gmail.com
  - Click Advanced
  - Tick “Use SMTP Authentication”
  - Add your GMail ID and the App password you have created (not your gmail password)
  - Tick “Use SSL”
  - Port number = 465
  - You can tick “Test configuration by sending test e-mail”
- In your project you can use “Post-Build Actions” -> Email Notification

Email & Docker

- Go to https://hub.docker.com/r/maildev/maildev
  - This is a SMTP server for viewing and testing emails during development
- Run the container using: *$docker run -p 1080:80 -p 1025:25 maildev/maildev*
- Page might say *-p 1080:80 -p 1025:25* That is a typo
- Open your web-browser: localhost:1080
- Go to Manage Jenkins -> Configure System -> Extended E-mail Notification
- SMTP server = localhost
- SMTP port = 1025

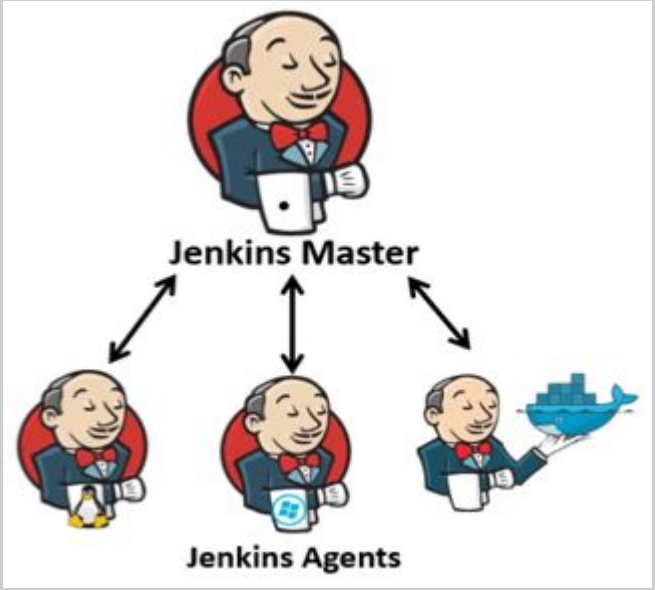
Build Triggers

- Start a job when an event happens
- We have seen the Github Webhook
- Now we can see: Build periodically
- Click on the '?' on the right to check out the syntax

Github Webhook for a Jenkins Project

- In your Jenkins project settings, check the “GitHub hook trigger for GITScm polling” under Build Triggers
- On your project’s github repo:
  - Go to "settings" in the right corner.
  - Click on "webhooks"
  - Click "Add webhook"
  - Write the Payload URL as:
    https://jenkins-url:8080/github-webhook/

Jenkins Agents



[Set up master and agent machines on Windows]

Jenkins Agents

Architecting for Scale:
https://www.jenkins.io/doc/book/architecting-for-scale/

Different ways to configure agents:

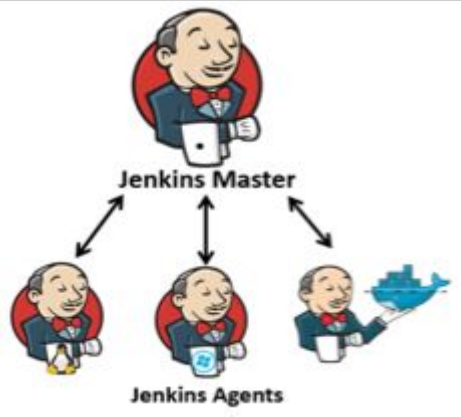
- Have Master start Slave agent via SSH
- Have Master start Slave agent on Windows
- Write your own script to launch Slave agent
- Launch Slave agent via Java Web Start

Let us have an example using Java Web Start

Jenkins Agents

A nice tutorial on how to have Master start Slave agent via SSH is here:

https://www.howtoforge.com/tutorial/ubuntu-jenkins-master-slave/

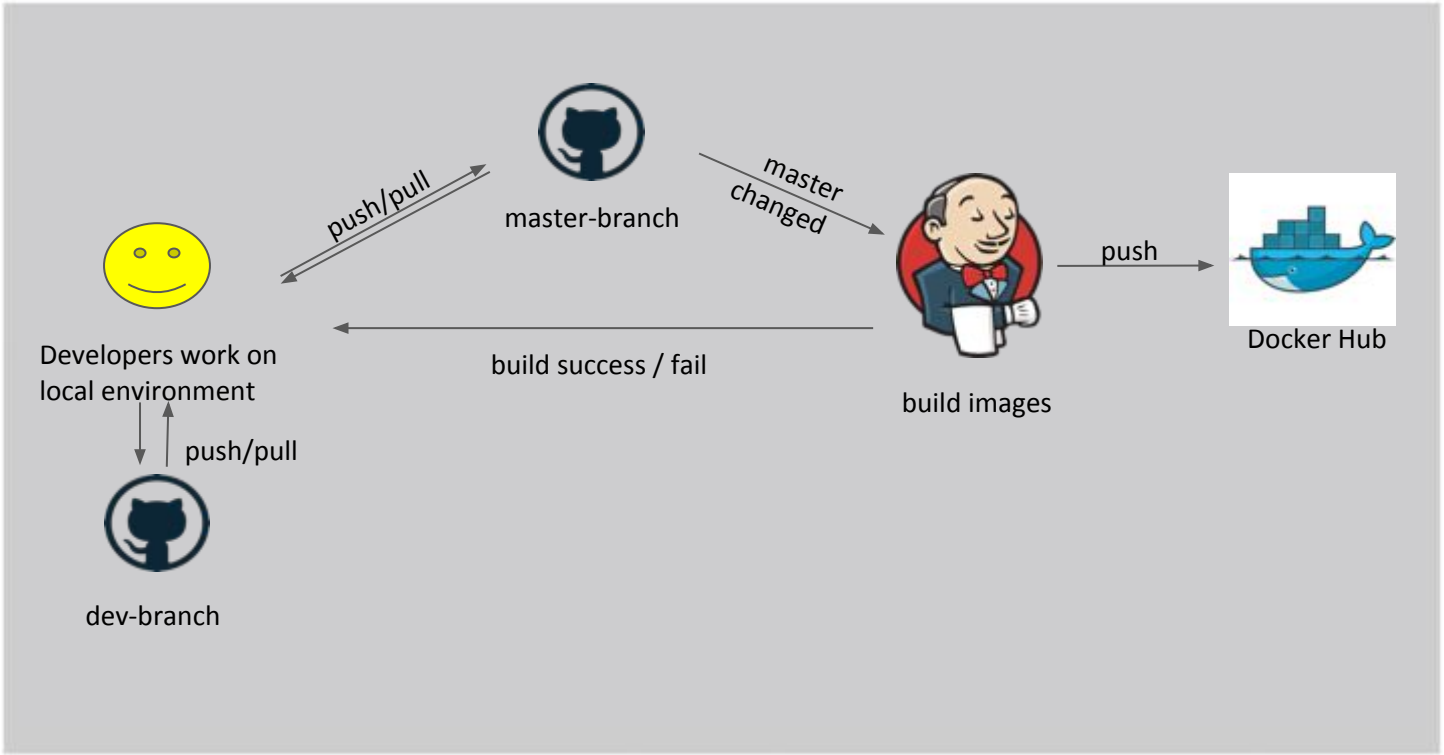


Blue Ocean Plugin for UI

- Sophisticated visualizations of CD pipelines for fast and intuitive comprehension of software pipeline status
- Pipeline editor that makes automating CD pipelines approachable by guiding the user through an intuitive and visual process to create a pipeline
- Personalization of the Jenkins UI to suit the role-based needs of each member of the DevOps team
- Pinpoint precision when intervention is needed and/or issues arise. The Blue Ocean UI shows where in the pipeline attention is needed, facilitating exception handling and increasing productivity
- Native integration for branch and pull requests enables maximum developer productivity when collaborating on code with others in GitHub and Bitbucket

https://plugins.jenkins.io/blueocean/

Jenkins & Docker



Full Project with Docker

- Here we will have a fully working NodeJS project
- We will dockerise it (create a docker image for it)
  - And run a container to make sure all is working
- Use Jenkins to build the image and upload it to Docker Hub

Enough Talking ..
Let’s do it!


thank you
danke
謝謝
ngiyabonga
рахмат
teşekkür ederim
gracias
merci
obrigado
bedankt
dziękuję
go raibh maith agat
mochchakkeram
sukriya
kop khun krap
terima kasih
감사합니다
ευχαριστώ
arigatō
dakujem
grazie
takk
tapadh leat
спасибо
Баярлалаа
dank je
mauruuru
hvala
sagolun
taiku
ありがとう
мерси
faafetai lava
vinaka
blagodaram
misaotra
matondo
paldies
grazzi
mersi
barka
welalin
tack
spas
kiitos
dankie
dhanyavad
köszönöm
enkosi
chnorakaloutioun
grafias ago
gracies
sulpay
sobodi
dekuji
mesi
najis tuke
kam sah hamnida
rahmet
tanemirt
dhanyavadagalu
shukriya
trugarez
хвала
asante
manana
obrigada
mamnun
chokrane
djiere dieuf
xiexie
diolch