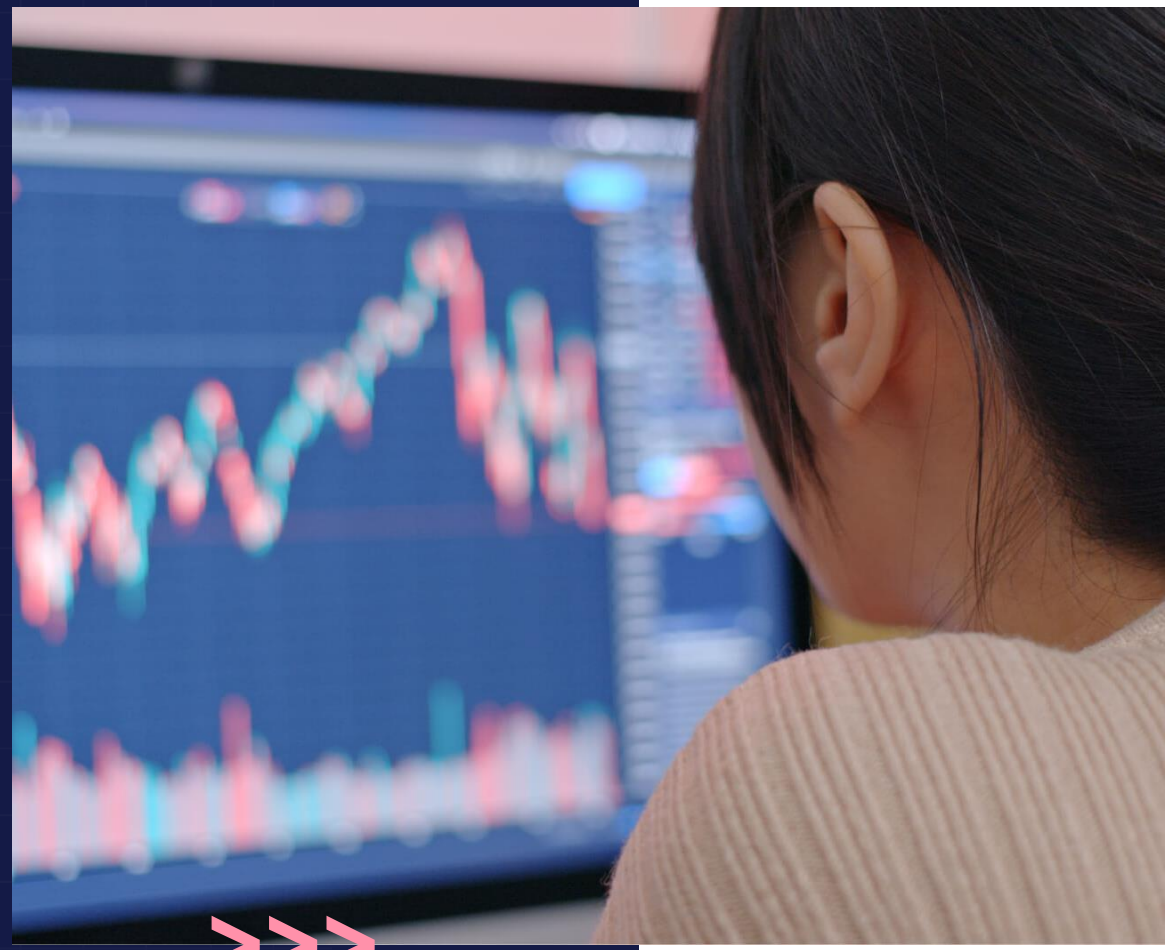
{mthree}

Jenkins

Site Reliability Engineering



Objectives

In this module, we will do a quick overview of Jenkins and its use in creating pipelines for managing DevOps style projects.

Learning Objectives

By the end of this module, you will be able to:

- Describe what Jenkins does
- Describe a Jenkins pipeline code
- Describe tests Jenkins can do
- Identify different job types
- Know what Jenkins can do
- Enable feedback
- Run a Jenkins pipeline

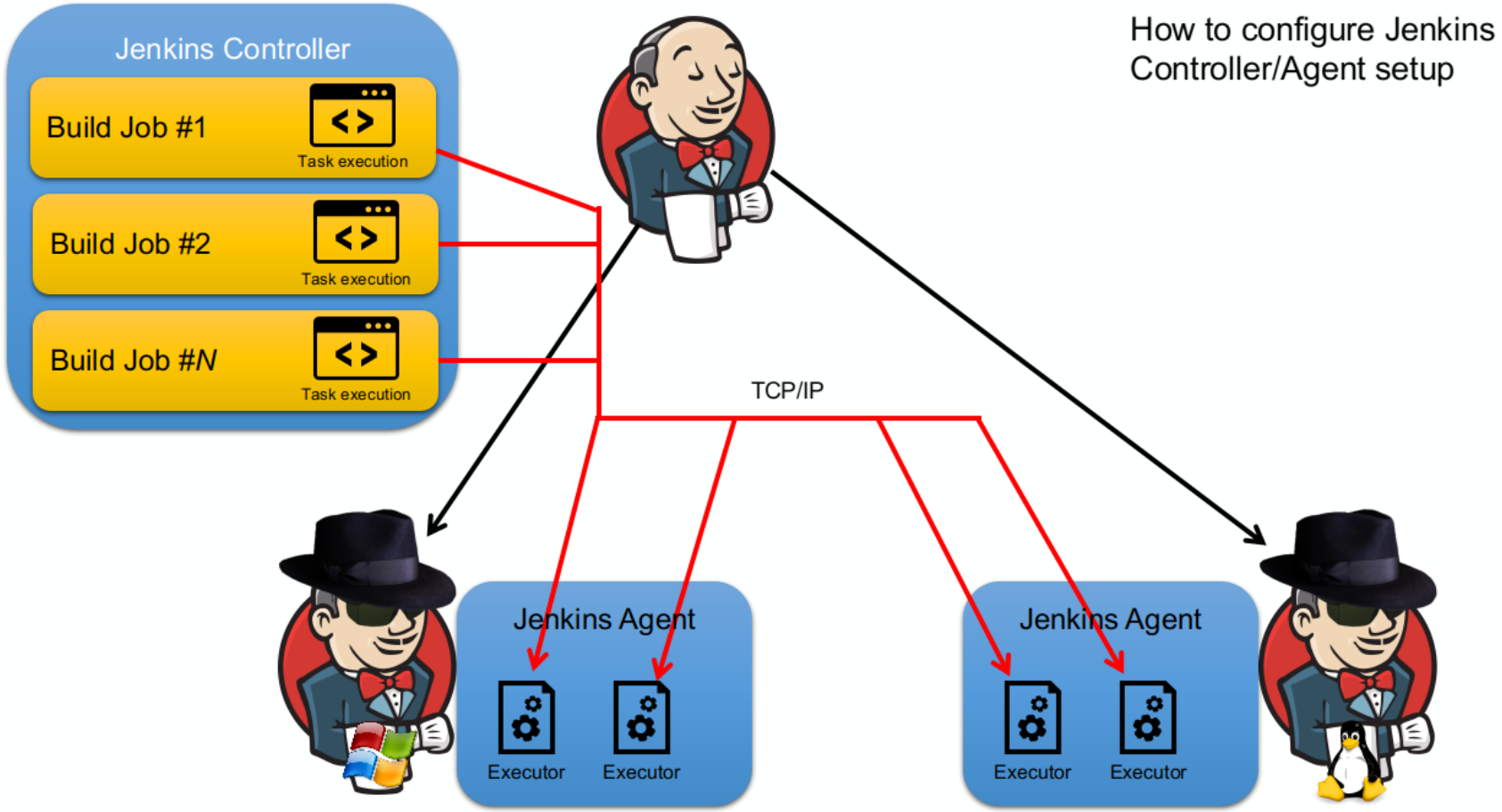
{mthree}

What is Jenkins?

- Free to use, open source automation server
- Initially designed for the continuous integration (CI) cycle
- Now it can cross the entire SDLC, including deployment
- Links into SCM to identify changes
- Can run almost anything
  - >>> Cannot run graphical services

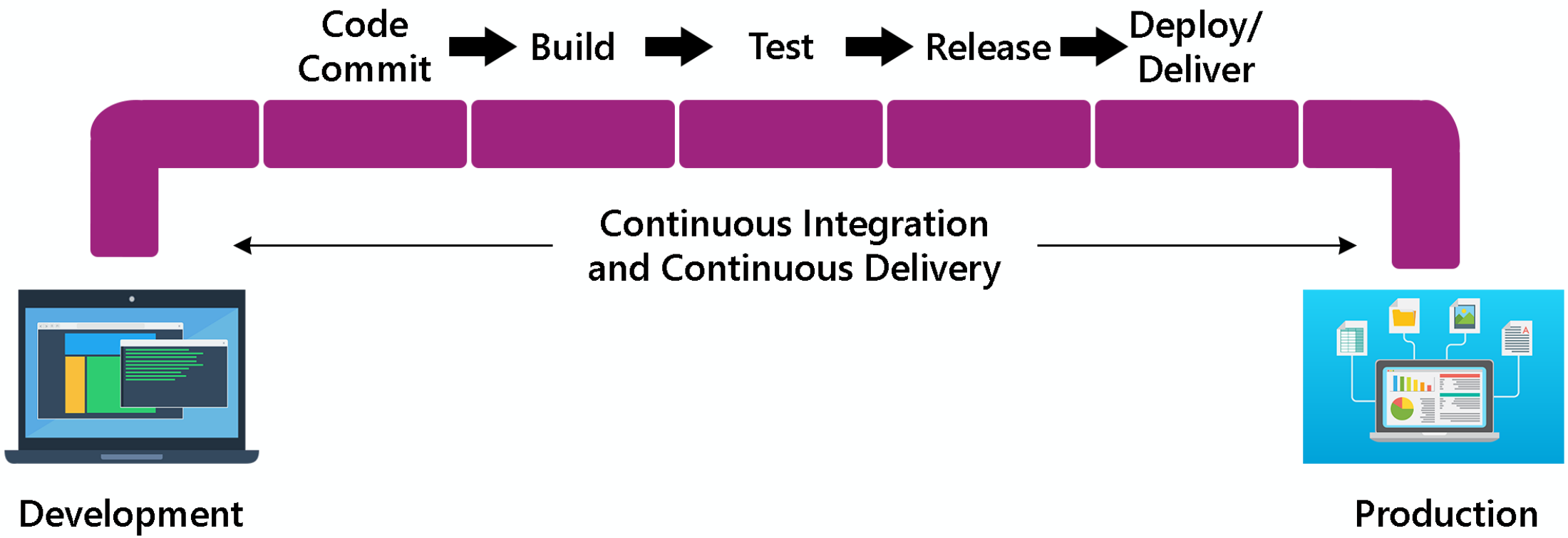
{mthree}

Scaling Jenkins



{mthree}

Pipelines



{mthree}

Code Pipeline

- Jenkins uses Groovy language to code pipelines
- File is normally placed in the root of the SCM repository
  - >>> Normally named Jenkinsfile
  - >>> If you use a different name, make sure to specify the name in the pipeline configuration
- Requires Pipeline plugin and dependencies to be installed

DevOps – Jenkins Practical

- The Orderbook application in Git:
  >>> https://github.com/The-Software-Guild/sre-orderbook/blob/master/Jenkinsfile

- Here we want you to understand what the code is doing

- Add your Job to build your application pipeline

{mthree}

Pipeline Creation – step by step

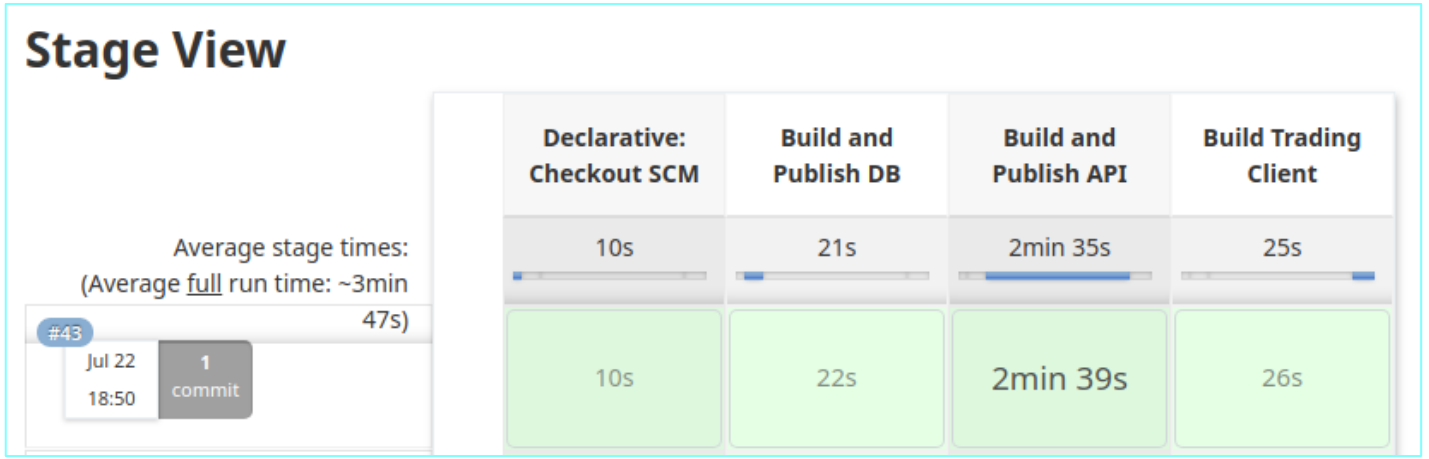
- This will show the single pipeline way of creating the pipeline
- Steps required
  - >>> GitHub Personal Access Token creation
  - >>> Jenkins credential addition or update
  - >>> Create Jenkins pipeline task

- **NOTE**: We have a standard naming convention

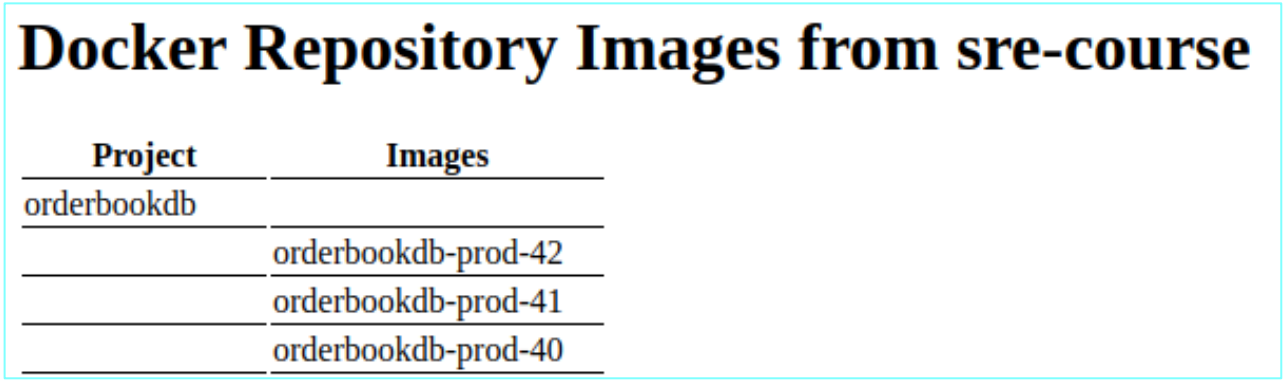
{mthree}

The Running Pipeline

- The pipeline view



- Viewing the container versions in the repository
  >>> http://ecrlist.computerlab.online/index.php



Docker Repository Images from sre-course

Project	Images
orderbookdb	
	orderbookdb-prod-42
	orderbookdb-prod-41
	orderbookdb-prod-40

{mthree}

Activity: Ready for Production

- If our application release works as expected in Dev:
  - >>> What could we do to make our release available to a production environment?
  - >>> Could we simply refer to the Dev containers in the software repository?
  - >>> Could we rebuild the containers from Git but name them prod?
  - >>> Could we make a copy of the Dev containers?

Answer – step by step

- We would make a copy
  - >>> Ideally, we promote the Dev package into a Prod repository

- Why?
  - >>> We should not use the Dev packages as they could change
  - >>> We need something to be in a Prod environment for segregation of duty
    - ~ Reduce accidental deletion or change
  - >>> We should not rebuild from Git
    - ~ Code base may have changed already - new sprint/release
    - ~ Too slow - we should not recompile unless we've genuinely lost the original package

{mthree}

Summary Q & A

{mthree}

References

- Pipeline Syntax
- Using a Jenkinsfile
- Pipeline Examples
- Blue Ocean

{mthree}