Keretapi Tanah Melayu Berhad GitHub Implementation

Sprint 1 – Day 2

30th May 2023

Sprint 1 – Day 2 Agenda

DevOps Principals

Time		Activities & Deliverables
From	To	

9:00	13:00	• Introduction to Cloud Native & DevOps
		• Introduction to Modern ICT Organization (People, Process & Technology)
		• Introduction to DevOps
		  • Git, Code Repository
		  • Continuous Integration
		  • Continuous Testing
		  • Continuous Delivery / Continuous Deployment
		• GitHub DevOps Platform

13:00	14:30	• Lunch

14:30	16:30	• GitHub Features Demo
		  • Agile Process / Project Management with GitHub
		  • Git code repository with GitHub
		  • CI / CD Workflow & Pipeline with GitHub Actions

.who am i ()

#Hanafiah Hassan

<Bachelor of Science in Computer Science>
<Bachelor of Arts in Mathematics>
<Minor degree in Statistics>
<University of Missouri, St. Louis, USA>

<Associate Degree>
<State University of New York, Buffalo and Institut Teknologi MARA>

#Managing Director

<Enovade Sdn Bhd>
<Cloud Connect Sdn Bhd>

#previous employment

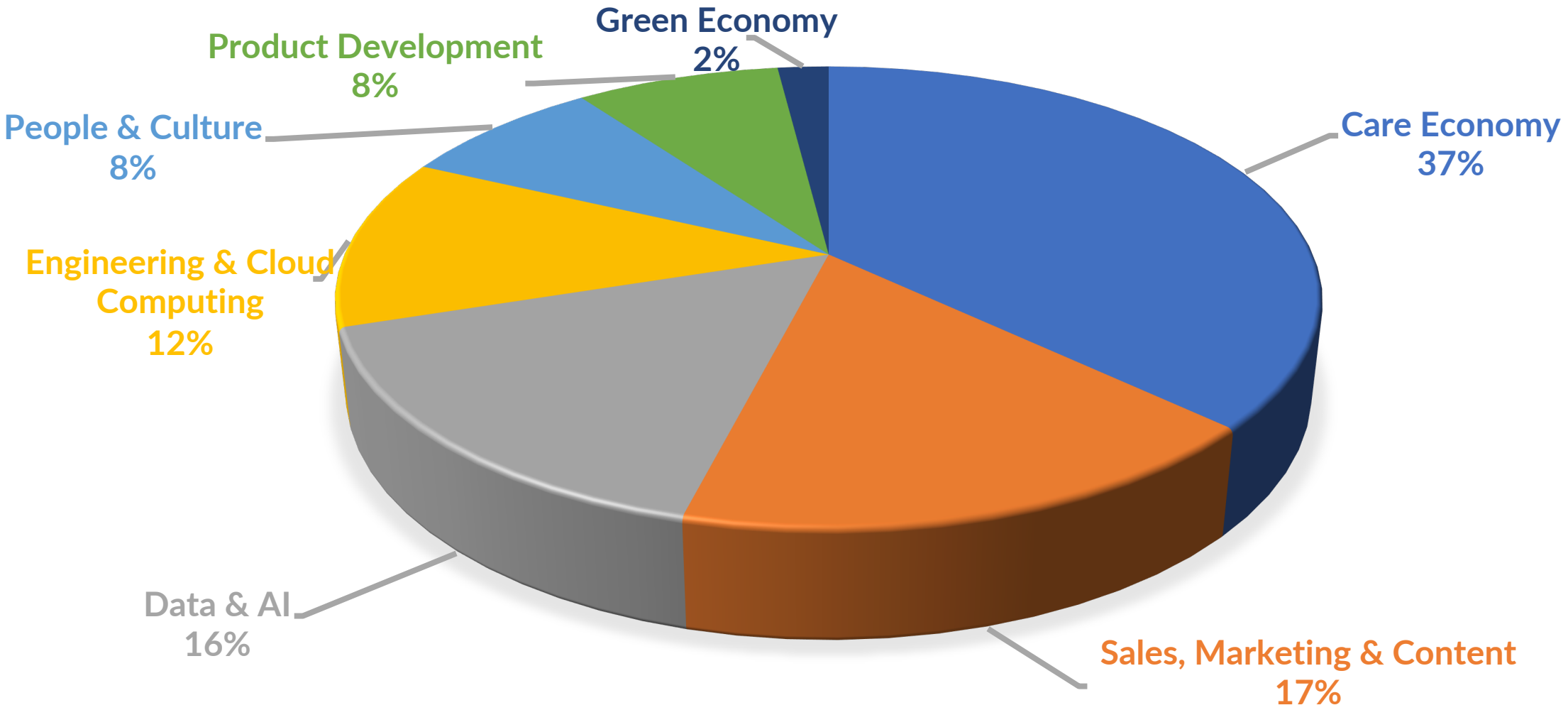
<Copernicus USA>
<Microsoft Malaysia>
<Microsoft Asia>
<Dimension Data>

Trend Semasa

ICT Trends & Prediction

Jobs Of Tomorrow

World Economic Forum Report January 2020



Professions of the Future

Data & AI

Emerging Jobs

1. Artificial Intelligence Specialist
2. Data Scientist
3. Data Engineer
4. Big Data Developer
5. Data Analyst
6. Analytics Specialist
7. Data Consultant
8. Insights Analyst
9. Business Intelligence Developer
10. Analytics Consultant

Top 10 Skills

1. Data Science
2. Data Storage Technologies
3. Development Tools
4. Artificial Intelligence
5. Software Development Life Cycle (SDLC)
6. Management Consulting
7. Web Development
8. Digital Literacy
9. Scientific Computing
10. Computer Networking

World Economic Forum Report - January 2020
Jobs of Tomorrow: Mapping Opportunity in the New Economy

N Rank **Scale of Opportunity:** Small-scale Large-scale
Skill Type: Tech Disruptive Tech Baseline Business

Professions of the Future

Engineering & Cloud Computing

Emerging Jobs

1. Site Reliability Engineer / Cloud Computing /
2. Python Developer / Engineering /
3. Full Stack Engineer / Engineering /
3. Javascript Developer / Engineering /
5. Back End Developer / Engineering /
6. Frontend Engineer / Engineering /
6. Software Developer Dotnet / Engineering /
8. Platform Engineer / Cloud Computing /
9. Development Specialist / Engineering /
10. Cloud Engineer / Cloud Computing /
10. DevOps Engineer / Cloud Computing /
12. Cloud Consultant / Cloud Computing /
13. DevOps Manager / Cloud Computing /
14. Technology Analyst / Engineering /

Top 10 Skills

1. Development Tools
2. Web Development
3. Data Storage Technologies
4. Software Development Life Cycle (SDLC)
5. Computer Networking
6. Human Computer Interaction
7. Technical Support
8. Digital Literacy
9. Business Management
10. Employee Learning & Development

World Economic Forum Report - January 2020
Jobs of Tomorrow: Mapping Opportunity in the New Economy

Source
LinkedIn.

(N) Rank **Scale of Opportunity:** Small-scale Large-scale
Skill Type: Tech Disruptive Tech Baseline Business

Professions of the Future

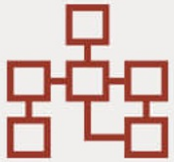
Product Development

1. Product Owner
2. Quality Assurance Tester
3. Agile Coach
4. Software Quality Assurance Engineer
5. Product Analyst
6. Quality Assurance Engineer
6. Scrum Master
8. Digital Product Manager
9. Delivery Lead

1. Software Testing
2. Software Development Life Cycle (SDLC)
3. Development Tools
4. Project Management
5. Business Management
6. Data Storage Technologies
7. Web Development
8. Manufacturing Operations
9. Digital Literacy
10. Leadership

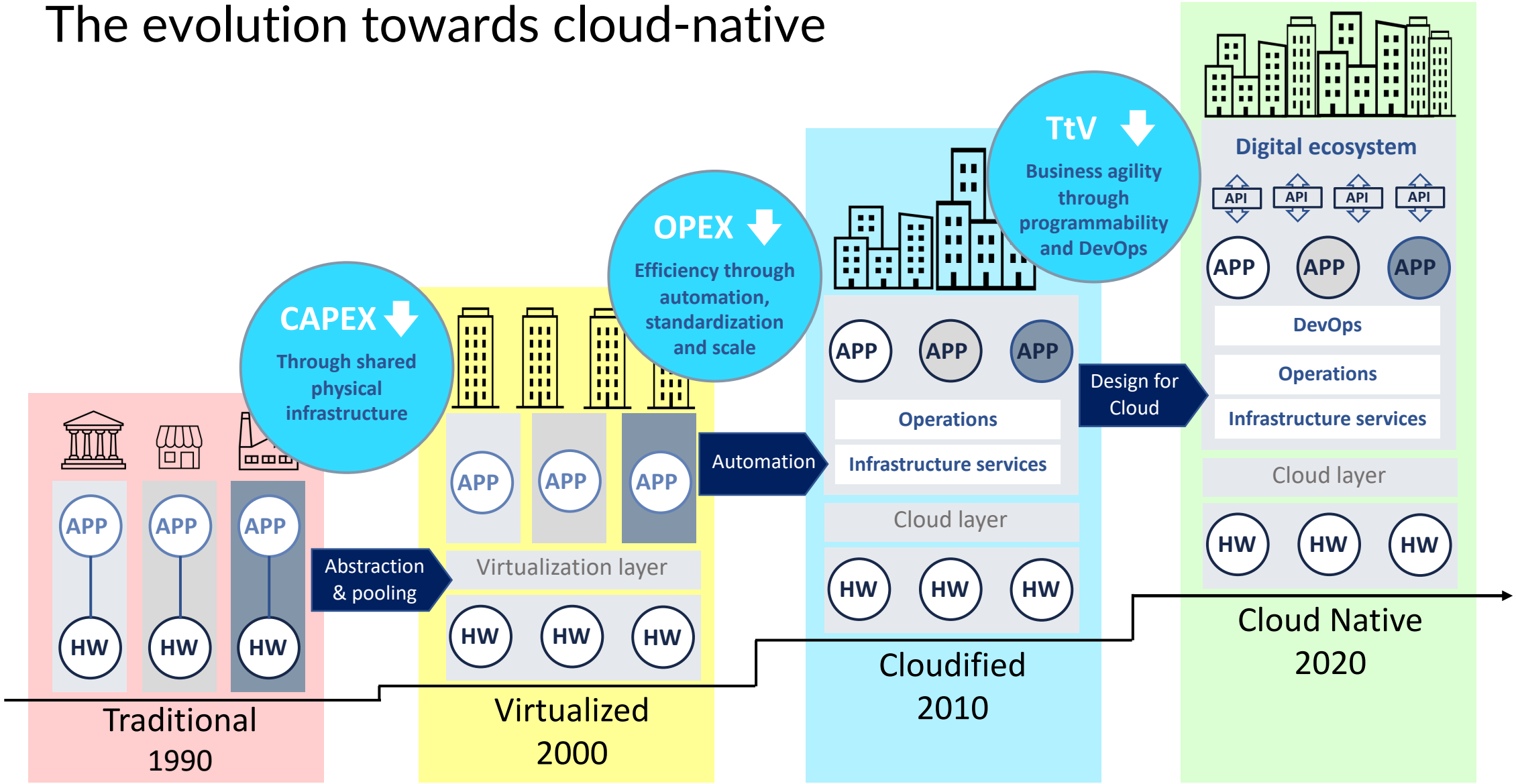
World Economic Forum Report - January 2020
Jobs of Tomorrow: Mapping Opportunity in the New Economy

Sejarah Evolusi *Cloud Computing*

	Development Process	Application Architecture	Deployment & Packaging	Application Infrastructure
~ 1980 ~ 1990	Waterfall 	Monolithic 	Physical Server 	Datacenter 
~ 2000	Agile 	N-Tie 	Virtual Servers 	Hosted 
~ 2010	DevOps 	Microservices 	Containers 	Cloud 

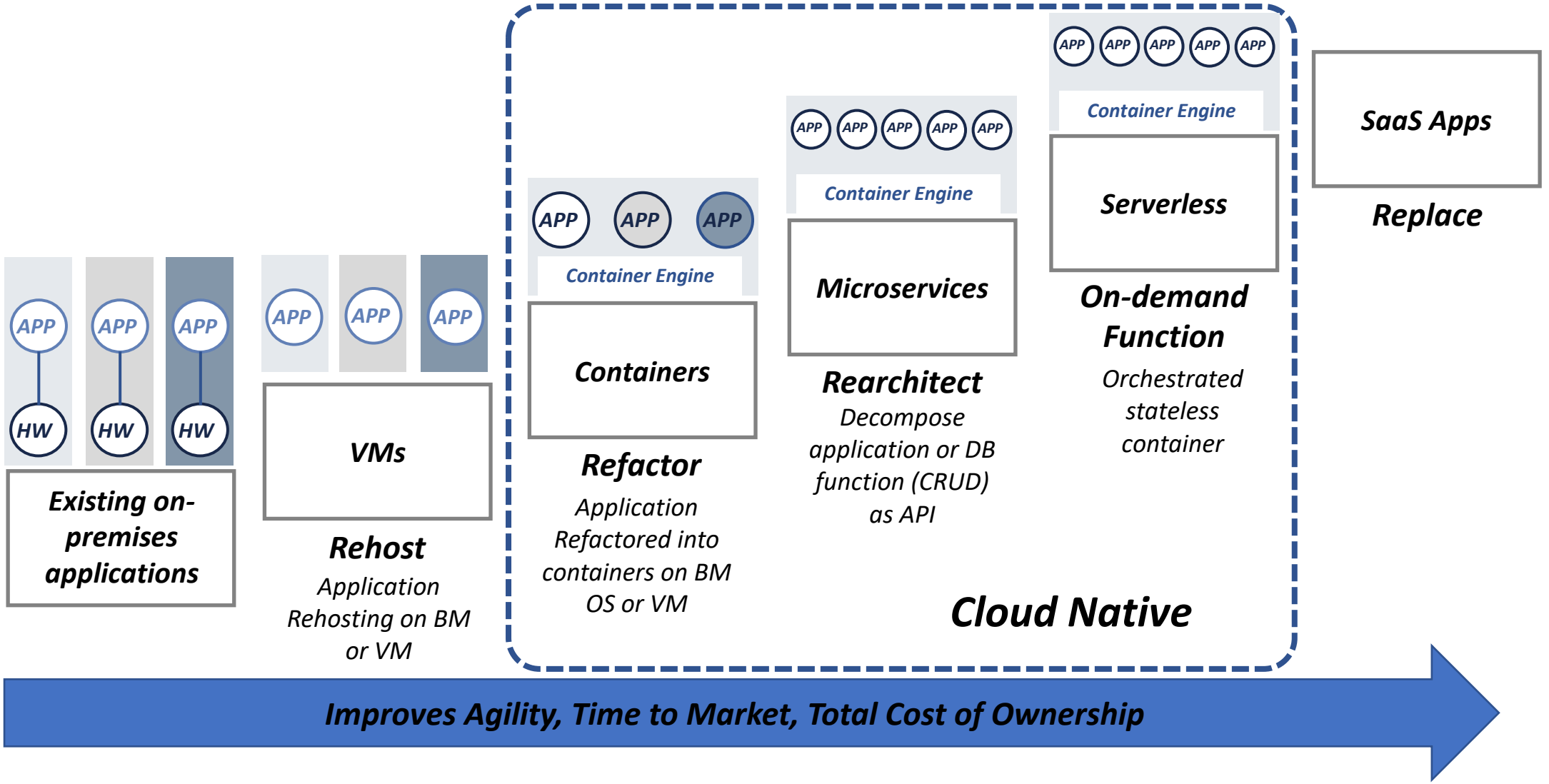
Cloud journey

The evolution towards cloud-native



Pemodenan Aplikasi

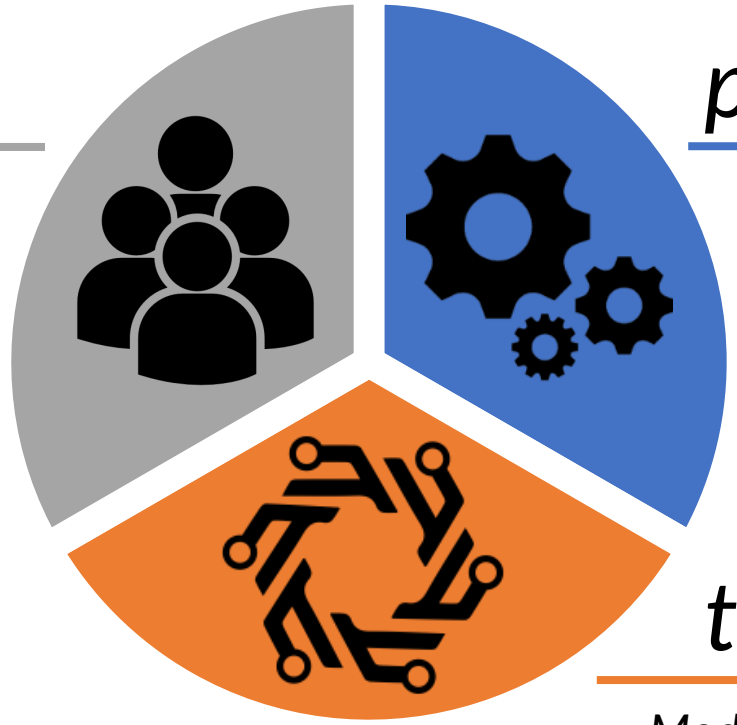
Evolusi ke arah pemodenan aplikasi



Moden Organisasi ICT

people

- *DevOps*
- *Front-end, back-end, and full-stack developer*
- *Cloud Consultant / Engineer*
- *QA Engineer / Tester*
- *Platform Engineer*
- *Scrum Master*
- *Agile Coach*



process

- ***DevOps** - Continuous Integration (**CI**), Continuous Delivery (**CD**) & Continuous Testing*
- ***Agile Framework***

technology

- *Modern **Infrastructure***
- *Modern **Web Application***
- *Cloud-native*

Cabaran Pembangunan Infrastruktur

- Terlebih budget dan masa
- Tiada standard/piawai dalam menguruskan dan menyimpan ***source code***
- Kerja berpasukan kurang efektif
- Masa yang lama diambil untuk penambahbaikan dan *bug fix*
- Tidak menepati keperluan pengguna (*user requirements)*
- Ujian aplikasi tidak menyeluruh
- *Project deployment* yang kurang lancar
- Penggunaan teknologi lama

Trend #1 – Teknologi Infra Moden

{ *Cloud Native* }

Modern People

DevOps

software development (Dev) and IT operations (Ops). Built, test, and release software

Front-end Developer

developer that codes the front end of a website or web application

Back-end Developer

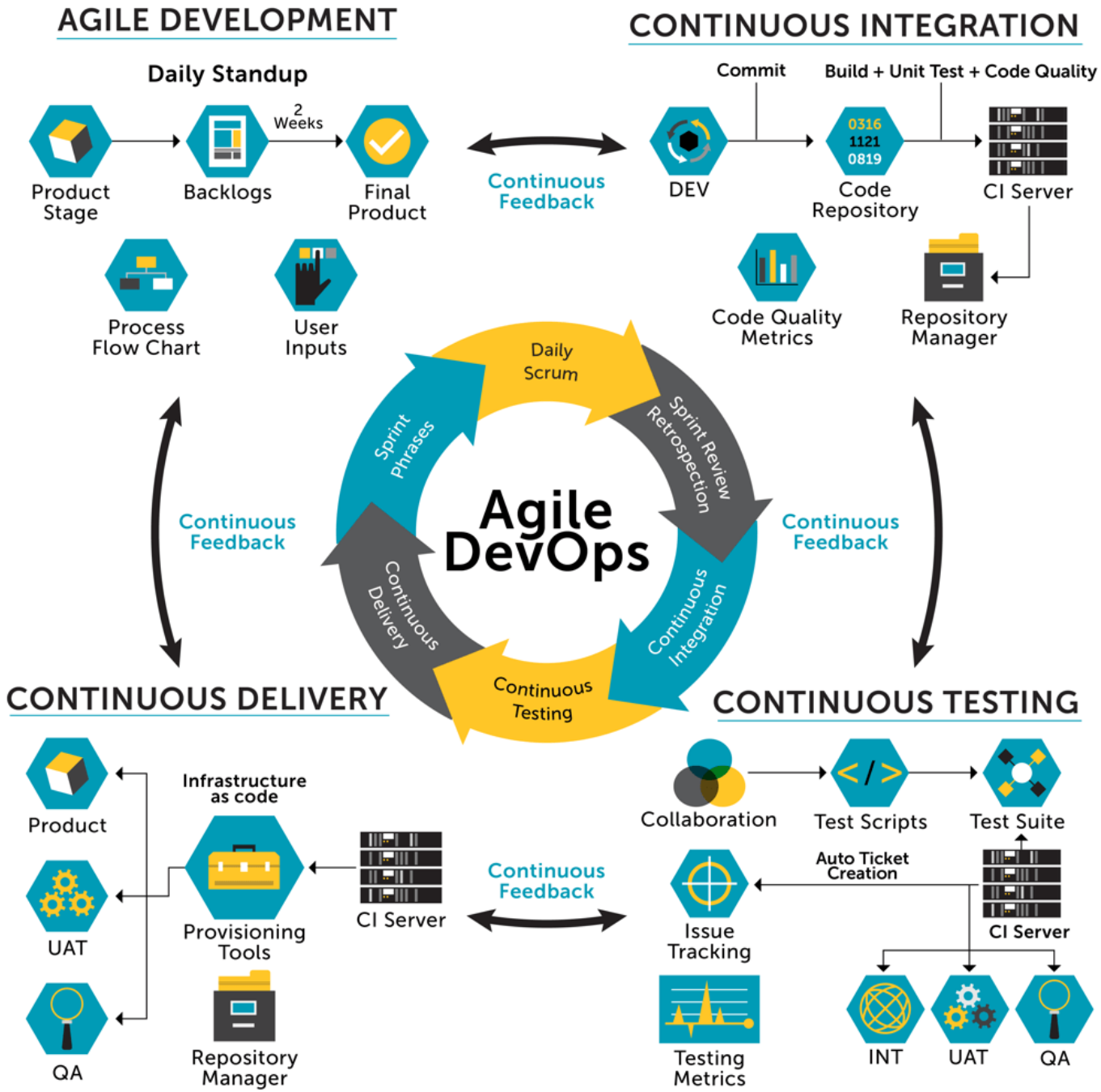
creates the logical back-end and core computational logic of a website, software or information system

Full-stack Developer

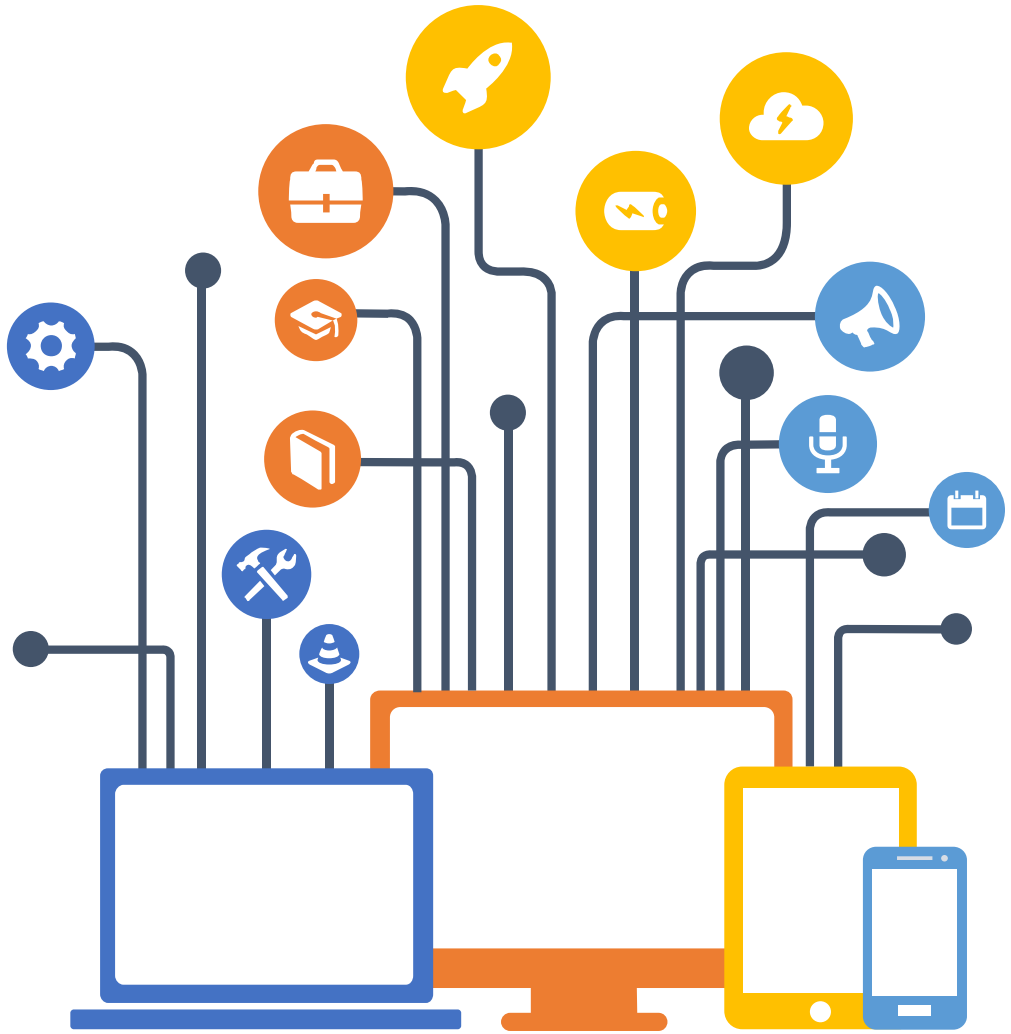
Front-end + Back-end Developers

Modern Process

- **AGILE Development methodology**
  - a practice that promotes **continuous iteration** of development and testing throughout the software development lifecycle of the project
- **Continuous integration (CI)**
  - **Software development practice** in which small adjustments to the underlying code in an application are tested every time a team member makes changes
- **Continuous delivery (CD)**
  - an **extension of continuous integration** - focuses on **automating** the **software delivery** process
- **Continuous Testing**
  - the process of executing automated tests as part of the software delivery pipeline in order to obtain feedback on the business risks associated with a software release candidate as rapidly as possible.



Modern Technology



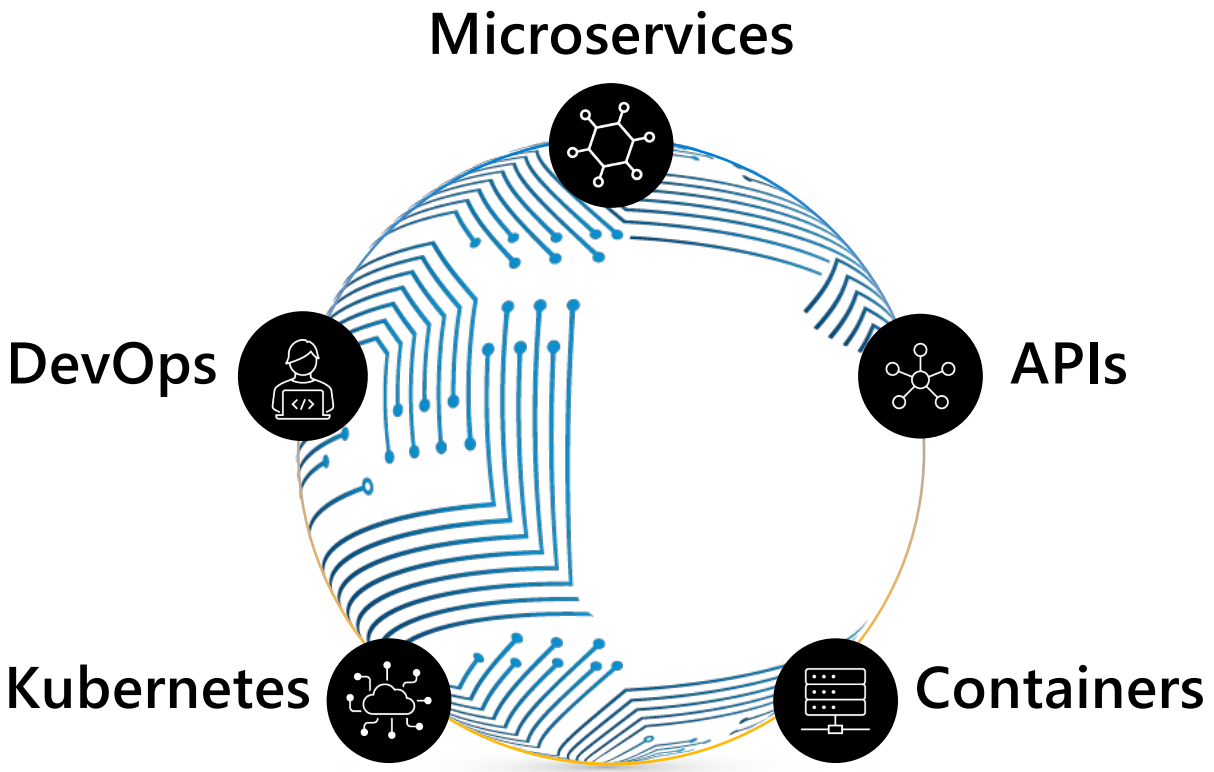
- Cloud Native
- Modern Infrastructure
- Modern Web Application

Apa itu *Cloud Native?*

- *Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, hybrids and edge clouds.*
- *Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.*
- *These techniques enable loosely coupled systems that are resilient, manageable, and observable*

Apa itu *Cloud Native*?

Package application code and dependencies in containers, deploy as microservices-API and manage them using DevOps processes and tools



Apa itu *CNCF?*

- *The Cloud Native Computing Foundation (CNCF) hosts critical components of the global technology infrastructure.*
- *CNCF brings together the world's top developers, end users, and vendors.*
- *CNCF is part of the nonprofit Linux Foundation.*
- *CNCF seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source and vendor-neutral technologies.*
- *Website : cncf.io*

CLOUD NATIVE
COMPUTING FOUNDATION

Trend #1 – Teknologi Infra Moden
{ *Cloud Native Landscape* }

https://landscape.cncf.io

Trend #2 – Teknologi Infra Moden

{ *Autonomic Computing* }

Apa itu *Autonomic Computing?*

- *the self-managing characteristics of distributed computing resources, adapting to unpredictable changes while hiding intrinsic complexity to operators and users – Wikipedia*
- *Autonomic computing is named after autonomic nervous system, where the particular systems can manage themselves by self-governing operation of the entire system*

Komponen *Autonomic Computing?*

01 Increased Responsiveness

Automatic configuration of components

02 Business Resiliency

Automatic discovery, and correction of faults

03 Operational Efficiency

Automatic monitoring and control of resources to ensure the optimal functioning

04 Secure Information & Resources

Proactive identification and protection from arbitrary attacks

Modern Technology

Modern Infrastructure

Azure Functions

Serverless

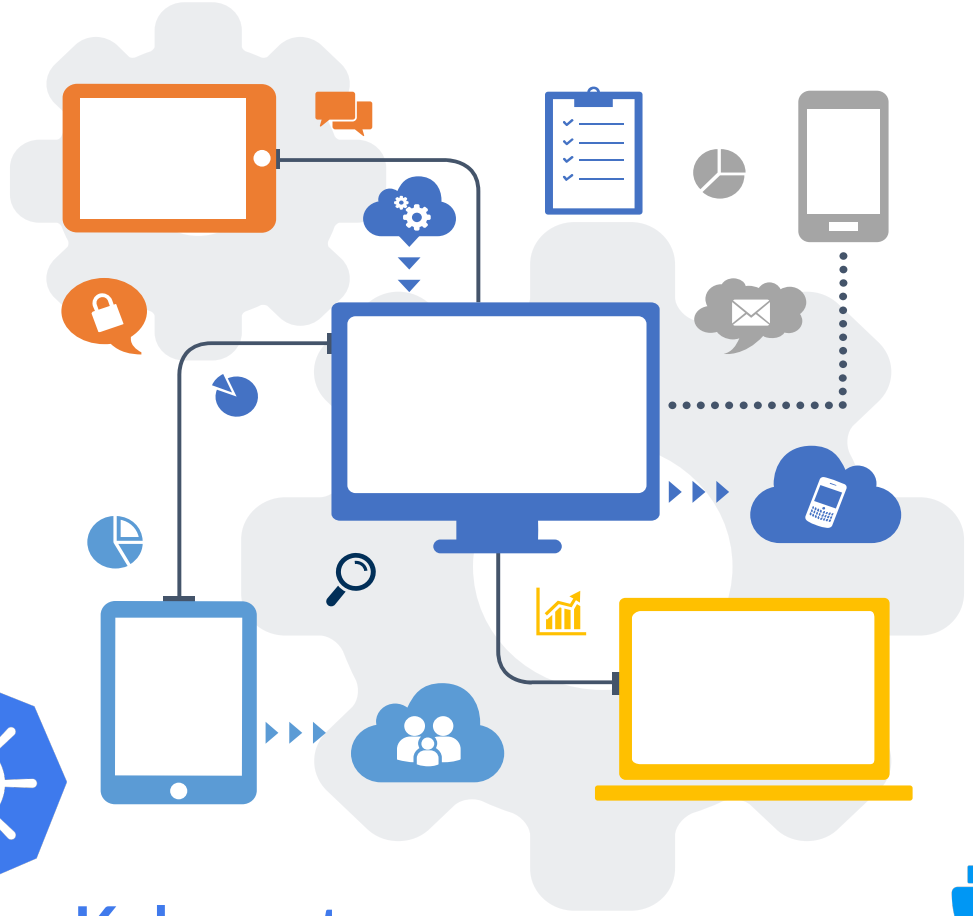
backend services on an as-used basis. **Function as a service**.

Microservices

split your application into a set of smaller, interconnected services, **API** services

Kubernetes

platform for automating deployment, scaling, and operations of application containers



Azure Kubernetes Service

Cloud Native

build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds
https://cncf.io

Container

Package Software into Standardized Units for Development, Shipment and Deployment

docker

Azure Container

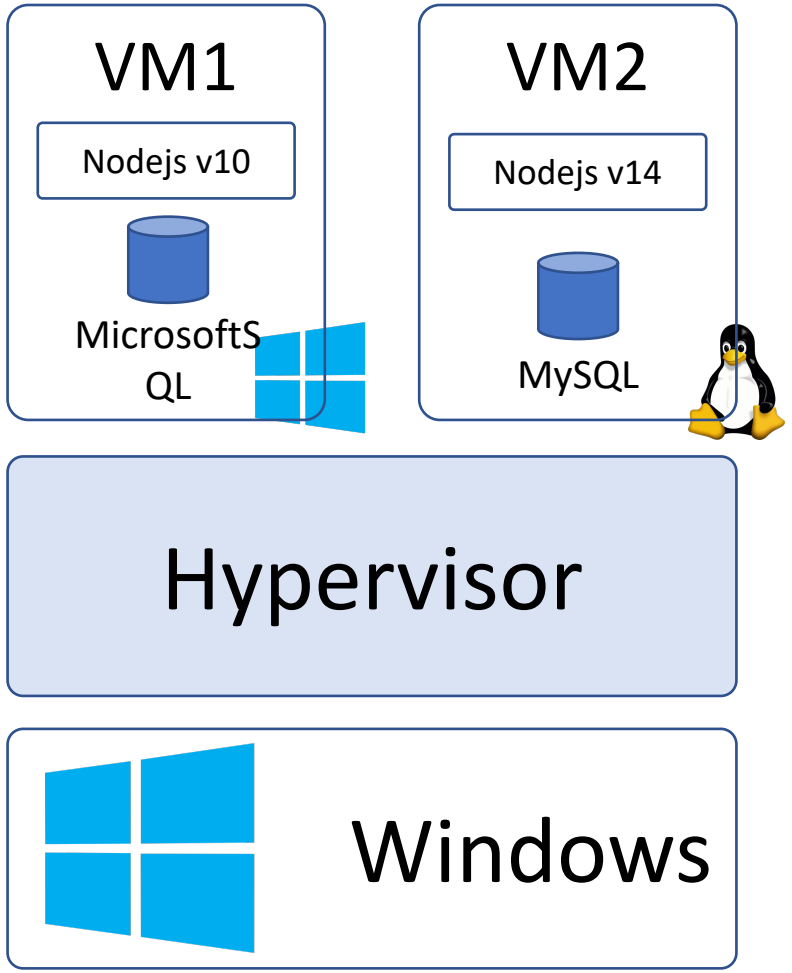
Real world example – MySQL Container

- *Project Name: MySQL Database Server.*
- *Solution: Using Docker to create MySQL Database Server*
- *Technology: Docker Container*

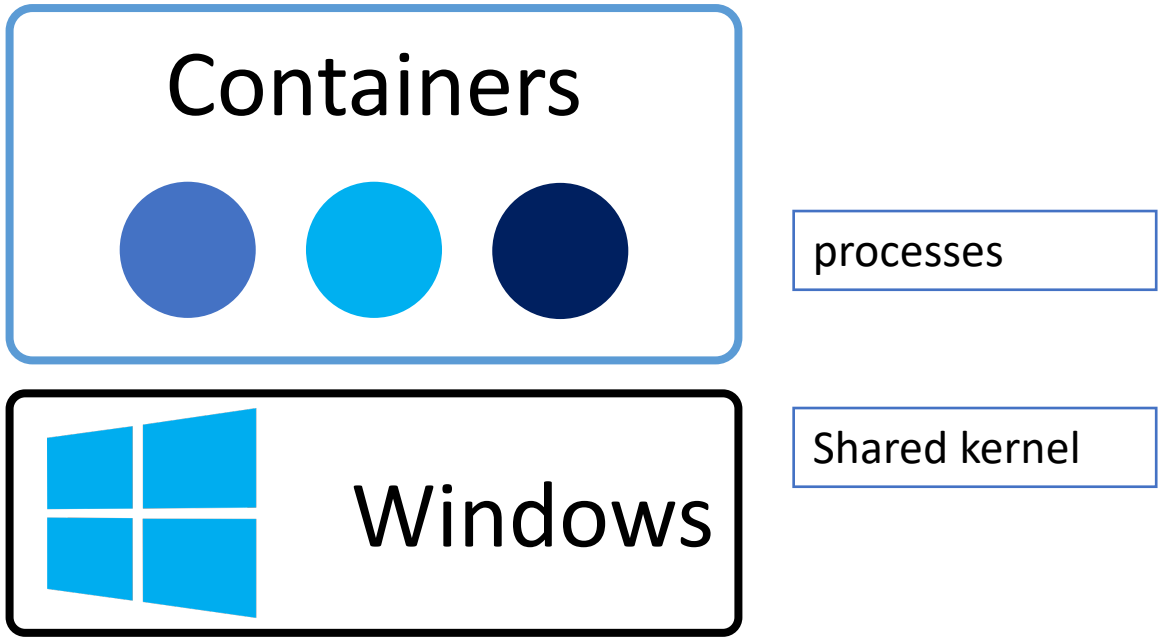
What is container?

- everything required to make a piece of software run is packaged into isolated containers.
- Unlike VMs, containers do not bundle a full operating system - only libraries and settings required to make the software work are needed.
- efficient, lightweight, self-contained systems and guarantees that software will always run the same, regardless of where it's deployed

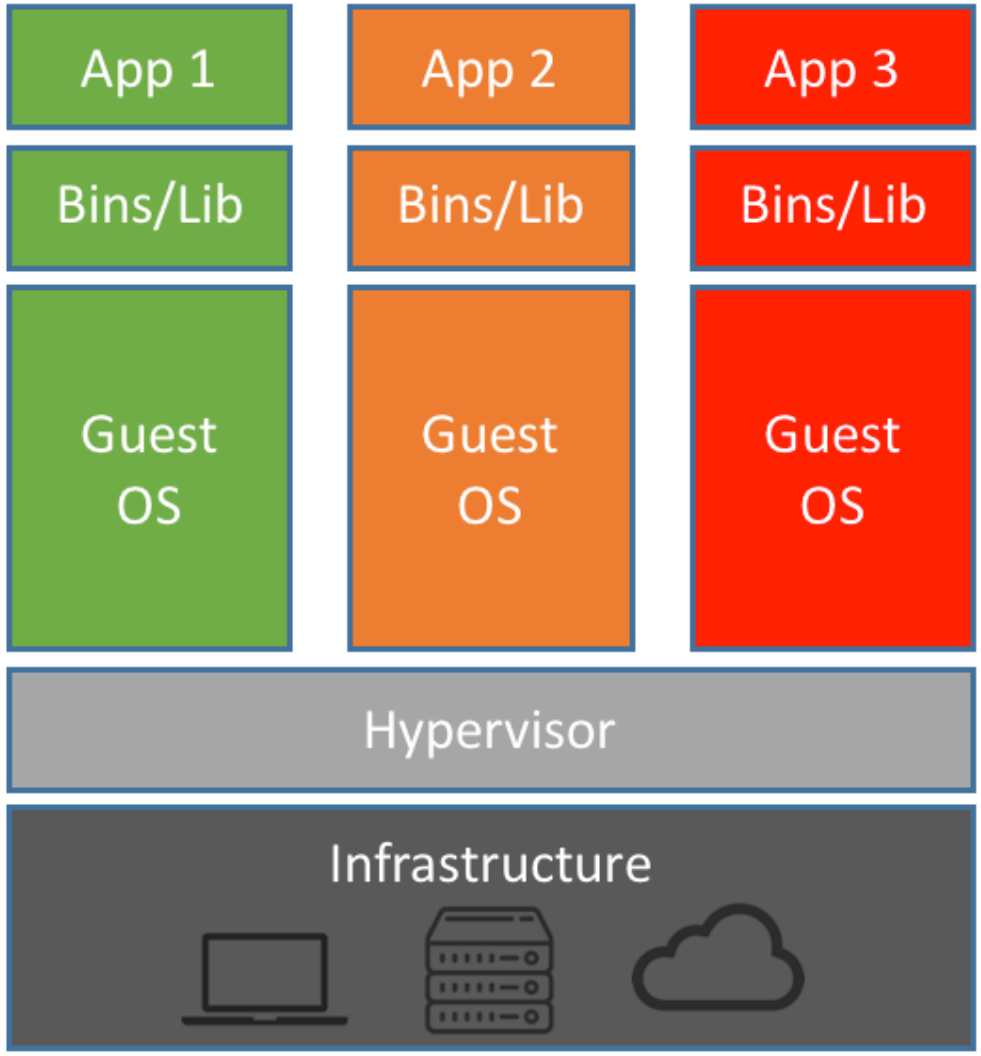
Hypervisor – Virtual Mcahine (VM)



Container

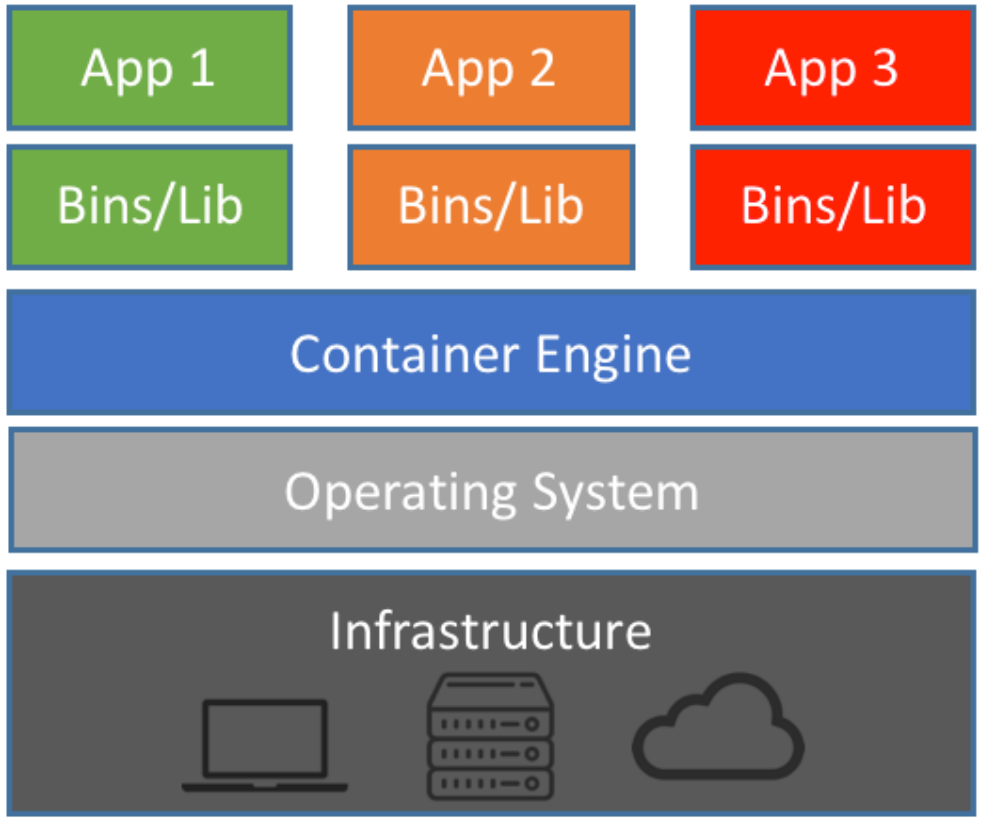


VM vs Container



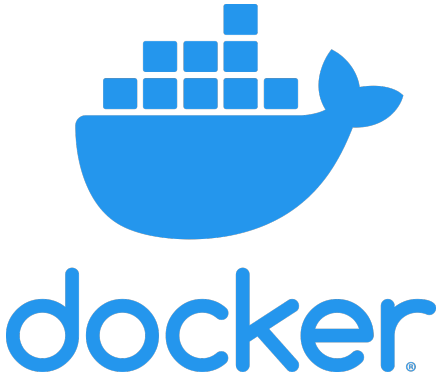
Machine Virtualization

- Containers are isolated, but share OS and, where appropriate, bin/libraries
- Containers don't include an OS.



Containers

What is Docker?



Software container platform

A platform for building, running & shipping applications.

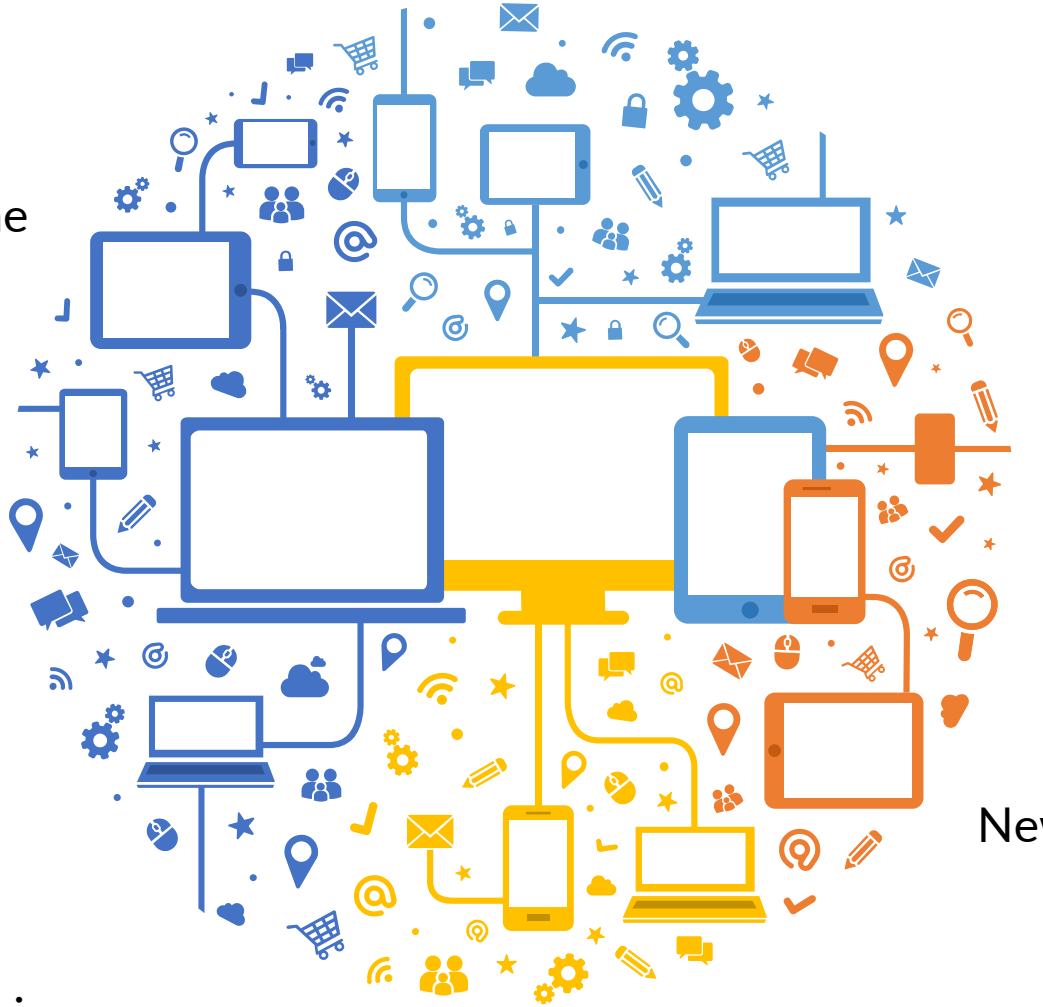
Why Docker?

Consistently

building, running & shipping applications.

Modern Technology

Modern Web Application



01

Javascript / NodeJS

Emerging of Javascript and the wide adoption of NodeJS

02

Microservices / API

Microservices & API are the crucial components of web development

03

Headless CMS

Back-end only content management system (CMS) using API

05

WebAssembly (Wasm)

Future Web technology to enable high-performance, memory-safe, secure applications on web pages

04

JAMStack

New Web application framework that utilizing Javascript, templated markup and API

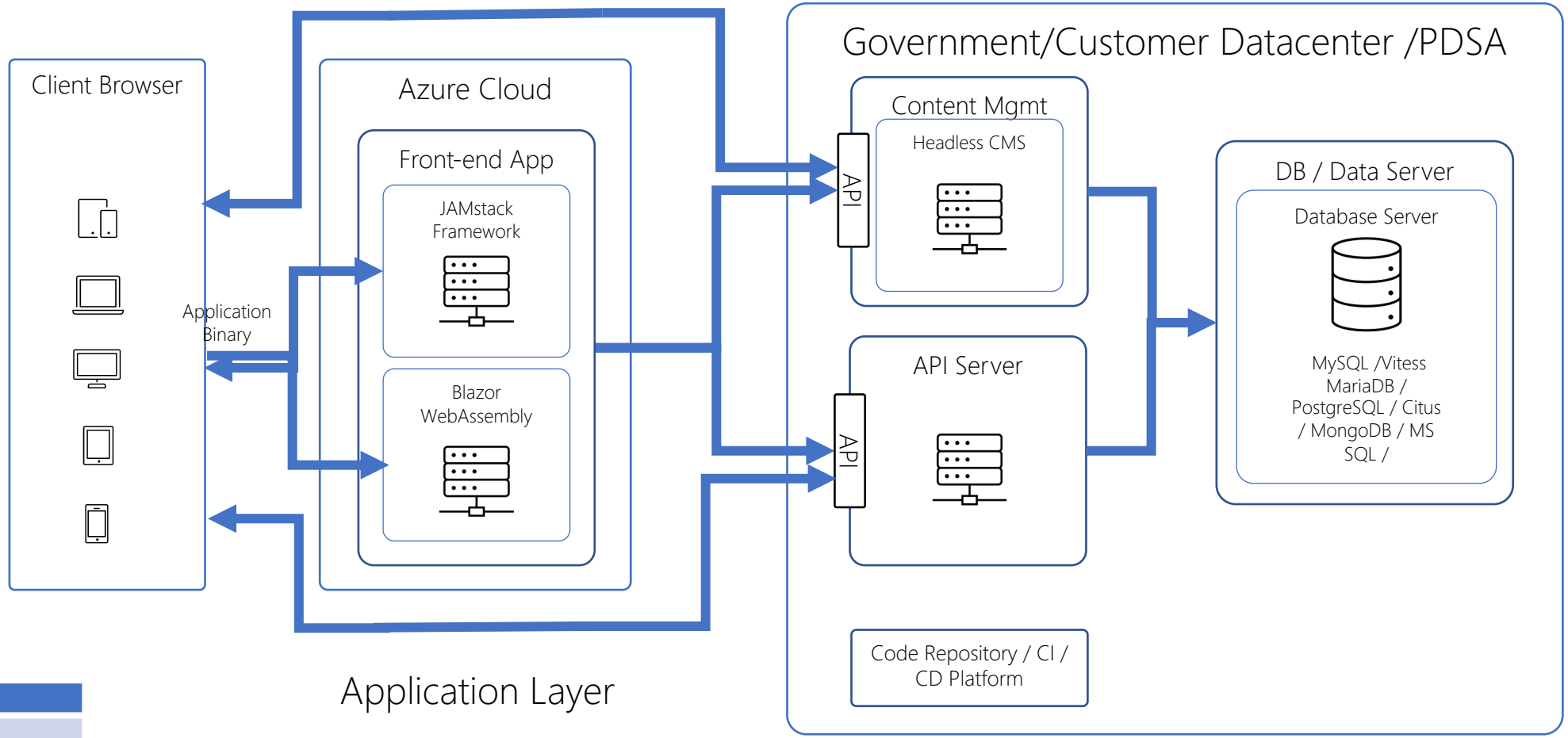
Real world example – JAMStack Application

- *Project Name: JAMStack App.*
- *Solution: Using Javascript to build JAMStack Application*
- *Technology: Javascript, Vue.js & Nuxt.js*

Rearchitecting Data Classification Model

Data Classification (Akta 88 – Akta Rahsia Rasmi 92)

No.	Type	Data Location
1	Top Secret (Rahsia Besar)	On premise
2	Secret (Rahsia)	On premise
3	Confidential (Sulit)	In country
4	Limited (Terhad)	In country
5	Open (Terbuka)	Any location



Our Experiences

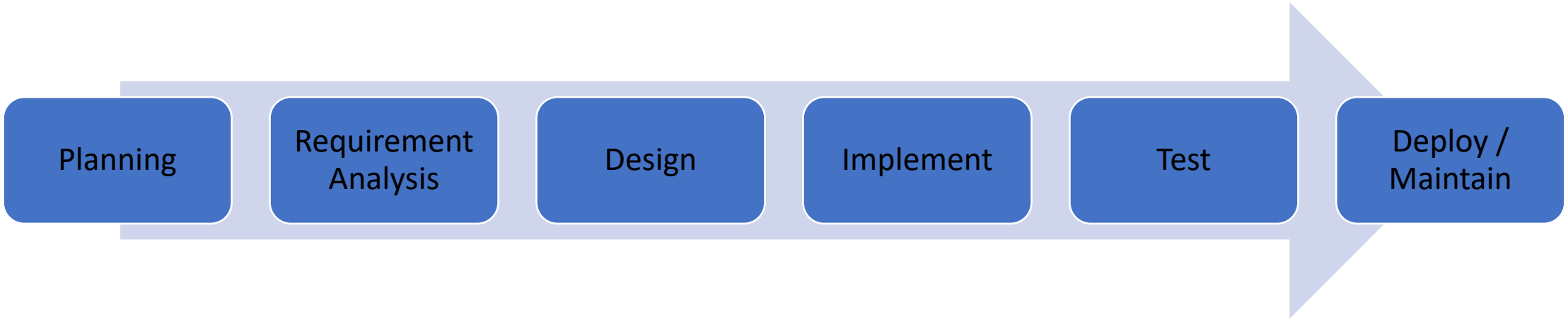
Type	Client
Private Sector	DHL Express
Private Sector	Shell Malaysia Exploration & Production
Public Sector	SIRIM
Public Sector	Ministry of Education

Azure VMs, Azure App Service, Azure Web Apps, Azure Static Web Apps, Azure Web App for Containers, Azure Container Instances, Azure Container Registry, Azure Container Apps, Azure Application Gateway, Azure DDoS Protection, Azure Key Vault

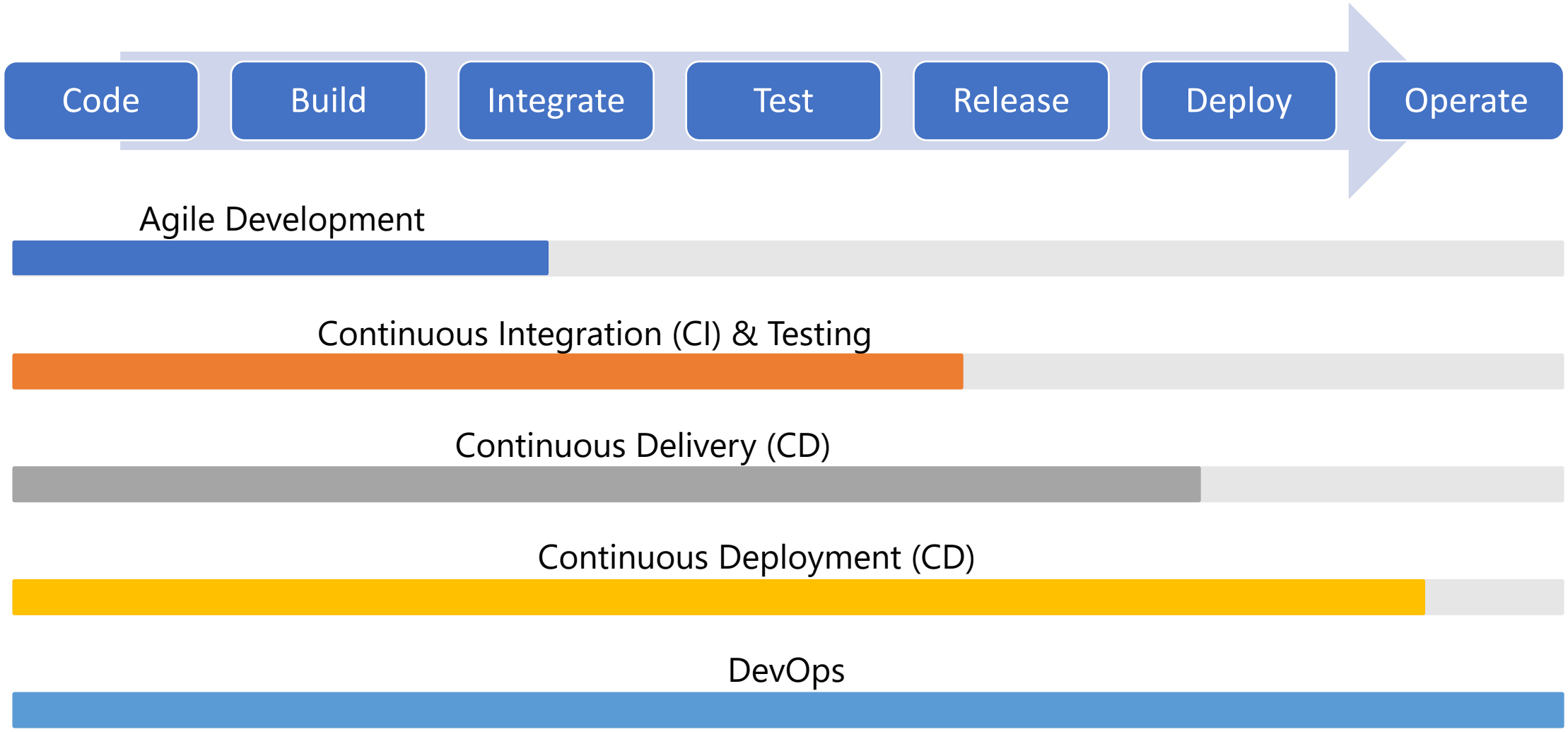
KTMB GitHub Implementation : Agile Software Development Workshop

KTMB GitHub Implementation : DevOps Principals Briefing Workshop

Software Development Life Cycle (SDLC)

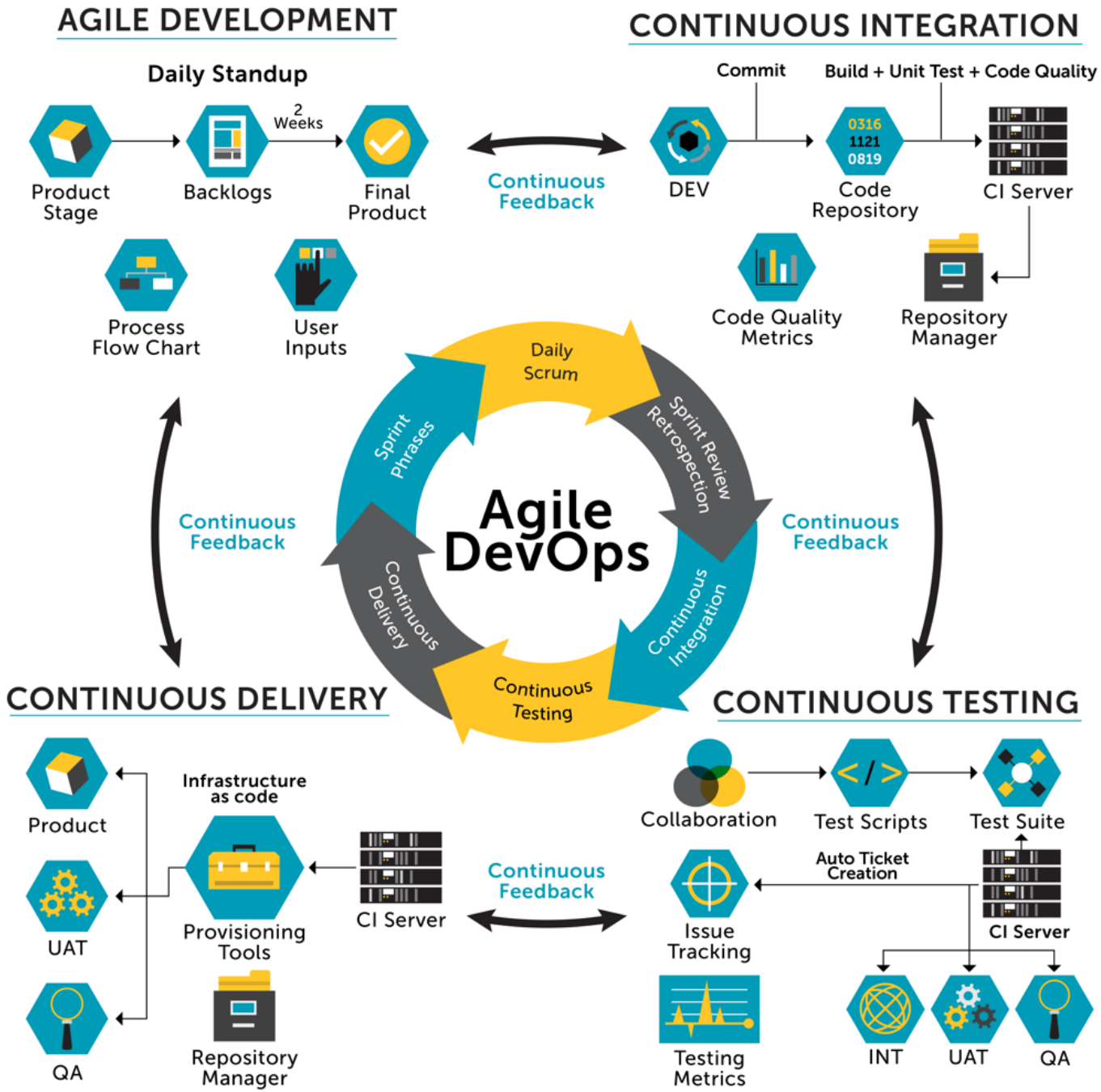


Modern Proses



Modern Proses

- AGILE Development methodology
  - a practice that promotes continuous iteration of development and testing throughout the software development lifecycle of the project
- Continuous integration (CI)
  - Software development practice in which small adjustments to the underlying code in an application are tested every time a team member makes changes
- Continuous delivery / deployment (CD)
  - an extension of continuous integration - focuses on automating the software delivery process
- Continuous Testing
  - the process of executing automated tests as part of the software delivery pipeline in order to obtain feedback on the business risks associated with a software release candidate as rapidly as possible.



DevOps

- DevOps – Development + Operations
- DevOps is the union of people, processes, and technology to deliver continuous value to users.

What is Agile?

- In software development, Agile practices involve discovering requirements and developing solutions through the collaborative effort of self-organizing and cross-functional teams and their customer(s)/end user(s).
- It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages flexible responses to change.
- It was popularized by the 2001 - publication of Manifesto for Agile Software Development.

-Wikipedia

Agile vs Waterfall

- One of the differences between agile software development methods and waterfall is the approach to quality and testing.
- In the **waterfall model**, work moves through Software Development Lifecycle (SDLC) phases—with one phase being completed before another can start (sequential) —hence the testing phase is separate and follows a build phase
- In **agile** software development, however, testing is completed in the same iteration as programming.

- Wikipedia

Agile vs Waterfall

- Waterfall works well for small projects with clear end goals, while Agile is best for large projects that require more flexibility.
- In Waterfall, clients aren't typically involved, whereas in Agile, client feedback is crucial.

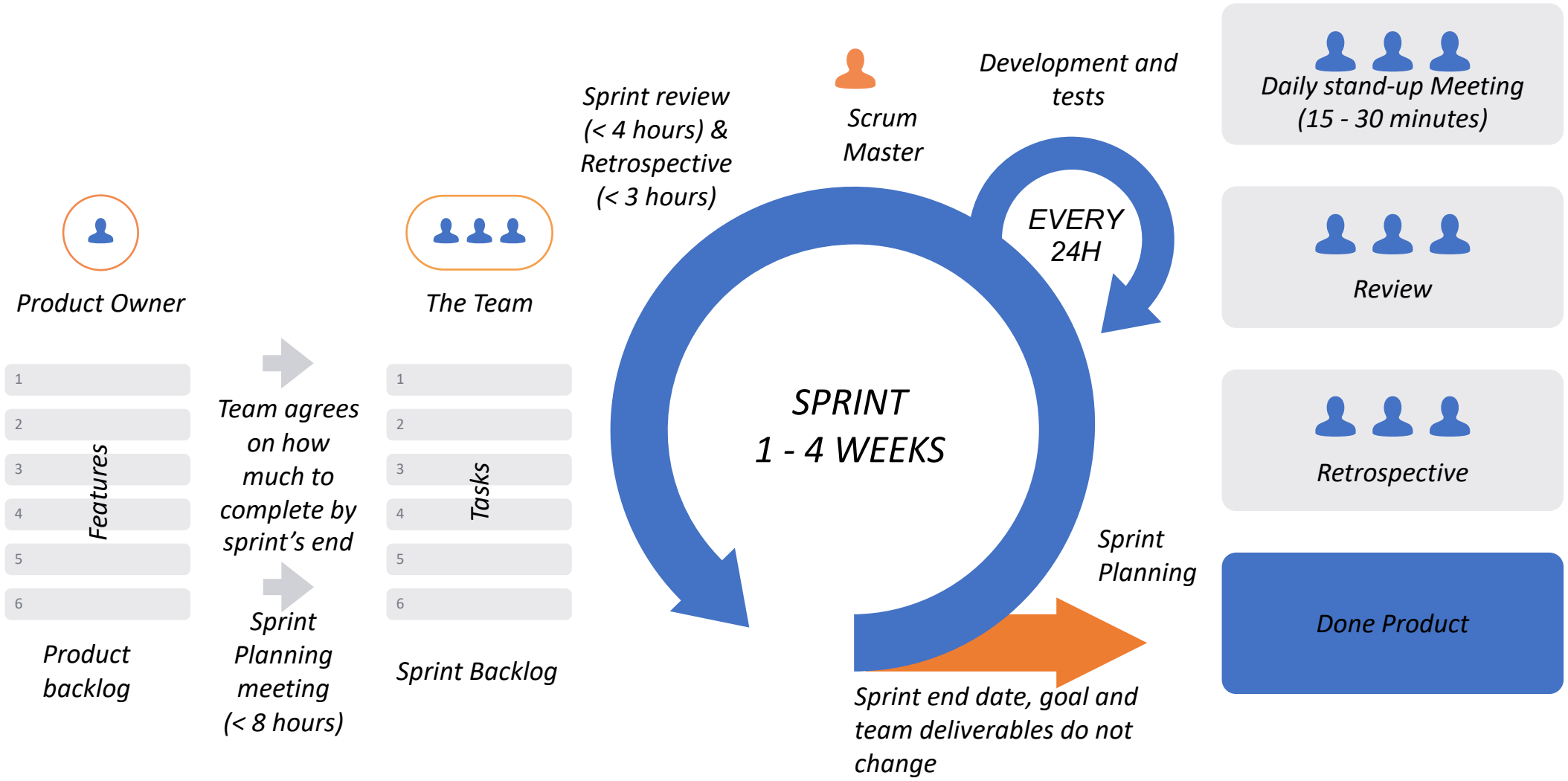
Type Agile Methodology

- Scrum,
- Extreme Programming (XP), and
- Kanban

What is Scrum?

- Scrum is a framework utilizing an agile mindset for developing, delivering, and sustaining complex products, with an initial emphasis on software development
- Scrum is a lightweight, iterative and incremental framework for managing complex software development work.

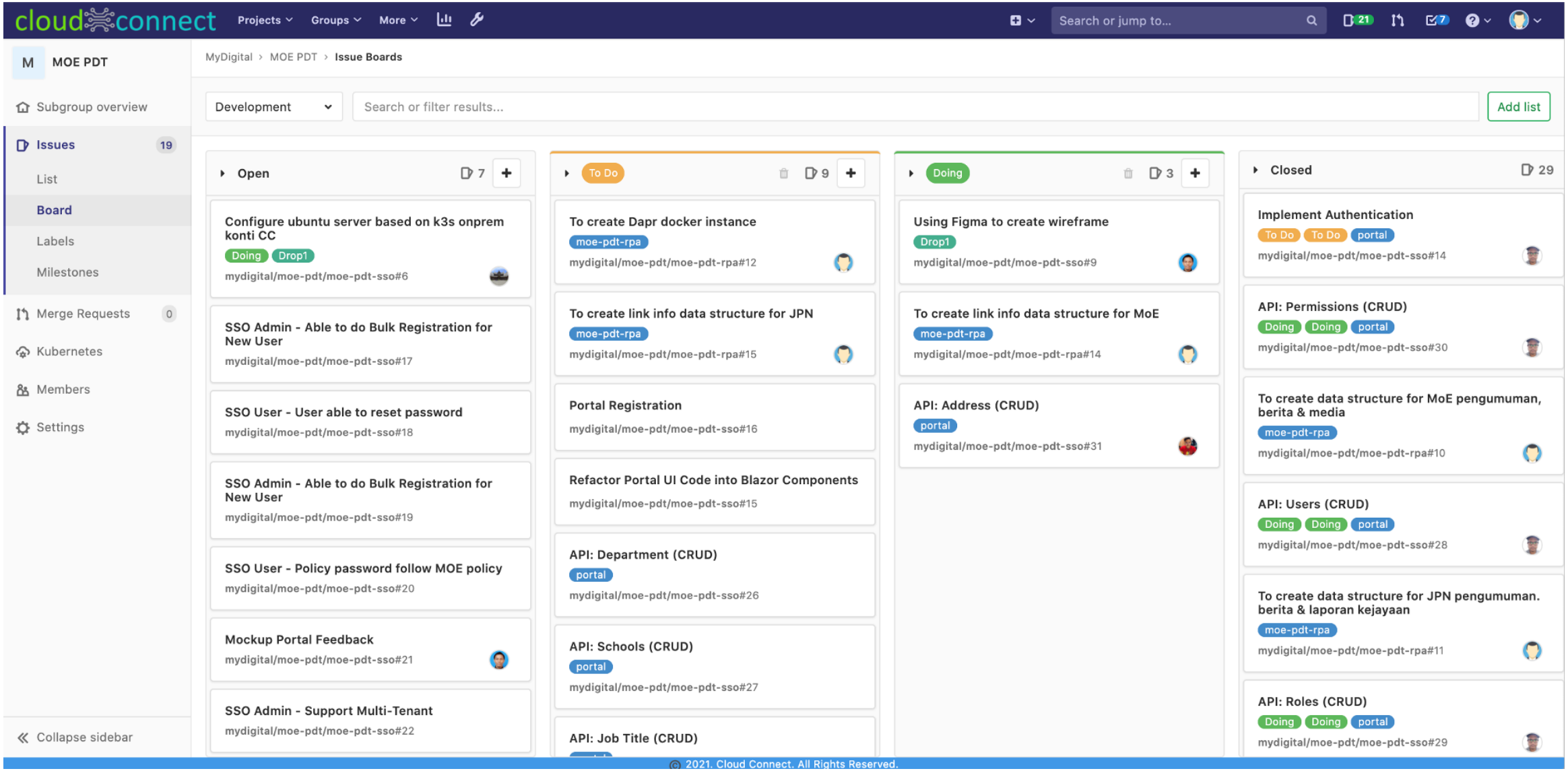
SCRUM


Product Owner
The Team
Sprint review (< 4 hours) & Retrospective (< 3 hours)
Scrum Master
Development and tests
Daily stand-up Meeting (15 - 30 minutes)
EVERY 24H
Review
Retrospective
Features
Tasks
Team agrees on how much to complete by sprint's end
SPRINT 1 - 4 WEEKS
Sprint Planning
Done Product
Product backlog
Sprint Planning meeting (< 8 hours)
Sprint Backlog
Sprint end date, goal and team deliverables do not change

What is Kanban?

- Kanban project management is a type of Agile methodology that seeks to improve the project management process through workflow visualization using a tool called a Kanban board.
- A Kanban board is composed of columns that depict a specific stage in the project management process, with cards or sticky notes representing tasks placed in the appropriate stage. As the project progresses, the cards will move from column to column on the board until they are completed.

Kanban


cloud connect
Projects
Groups
More
Search or jump to...
MOE PDT
Subgroup overview
Issues 19
List
Board
Labels
Milestones
Merge Requests 0
Kubernetes
Members
Settings
Collapse sidebar
MyDigital › MOE PDT › Issue Boards
Development
Search or filter results...
Add list
Open 7
Configure ubuntu server based on k3s onprem konti CC
Doing Drop1
mydigital/moe-pdt/moe-pdt-sso#6
SSO Admin - Able to do Bulk Registration for New User
mydigital/moe-pdt/moe-pdt-sso#17
SSO User - User able to reset password
mydigital/moe-pdt/moe-pdt-sso#18
SSO Admin - Able to do Bulk Registration for New User
mydigital/moe-pdt/moe-pdt-sso#19
SSO User - Policy password follow MOE policy
mydigital/moe-pdt/moe-pdt-sso#20
Mockup Portal Feedback
mydigital/moe-pdt/moe-pdt-sso#21
SSO Admin - Support Multi-Tenant
mydigital/moe-pdt/moe-pdt-sso#22
To Do 9
To create Dapr docker instance
moe-pdt-rpa
mydigital/moe-pdt/moe-pdt-rpa#12
To create link info data structure for JPN
moe-pdt-rpa
mydigital/moe-pdt/moe-pdt-rpa#15
Portal Registration
mydigital/moe-pdt/moe-pdt-sso#16
Refactor Portal UI Code into Blazor Components
mydigital/moe-pdt/moe-pdt-sso#15
API: Department (CRUD)
portal
mydigital/moe-pdt/moe-pdt-sso#26
API: Schools (CRUD)
portal
mydigital/moe-pdt/moe-pdt-sso#27
API: Job Title (CRUD)
Doing 3
Using Figma to create wireframe
Drop1
mydigital/moe-pdt/moe-pdt-sso#9
To create link info data structure for MoE
moe-pdt-rpa
mydigital/moe-pdt/moe-pdt-rpa#14
API: Address (CRUD)
portal
mydigital/moe-pdt/moe-pdt-sso#31
Closed 29
Implement Authentication
To Do To Do portal
mydigital/moe-pdt/moe-pdt-sso#14
API: Permissions (CRUD)
Doing Doing portal
mydigital/moe-pdt/moe-pdt-sso#30
To create data structure for MoE pengumuman, berita & media
moe-pdt-rpa
mydigital/moe-pdt/moe-pdt-rpa#10
API: Users (CRUD)
Doing Doing portal
mydigital/moe-pdt/moe-pdt-sso#28
To create data structure for JPN pengumuman. berita & laporan kejayaan
moe-pdt-rpa
mydigital/moe-pdt/moe-pdt-rpa#11
API: Roles (CRUD)
Doing Doing portal
mydigital/moe-pdt/moe-pdt-sso#29
© 2021, Cloud Connect. All Rights Reserved.

What is Extreme programming (XP)?

- Extreme programming (XP) is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements.
- As a type of agile software development methodology, it advocates frequent "releases" in short development cycles.
- It is used to improve software quality and responsive to customer requirements.
- Suitable for:
  - Small projects
  - Projects involving new technology or Research projects

{ Introduction to GitHub }

Apa itu *GitHub?*

- *Founded in Feb 2008*
- *Acquired by Microsoft in 2018 (USD 7.5 billion)*
- *73+ million users*
- *200+ million public repositories*
- *4+ million organizations*
- *GitHub Enterprise – self-hosted / self-managed*

Apa itu *GitHub?*

- *Offers the distributed version control and source code management functionality of Git.*
- *A complete DevOps platform, delivered as a single application for the entire software development and operations lifecycle.*
- *GitHub enables teams to apply Agile practices and principles to organize and manage their work*
- *provides everything you need to Manage, Plan, Create, Verify, Package, Release, Configure, Monitor, and Secure your applications*

GitHub *Features*

Collaborative Coding

Automation & CI / CD

Security

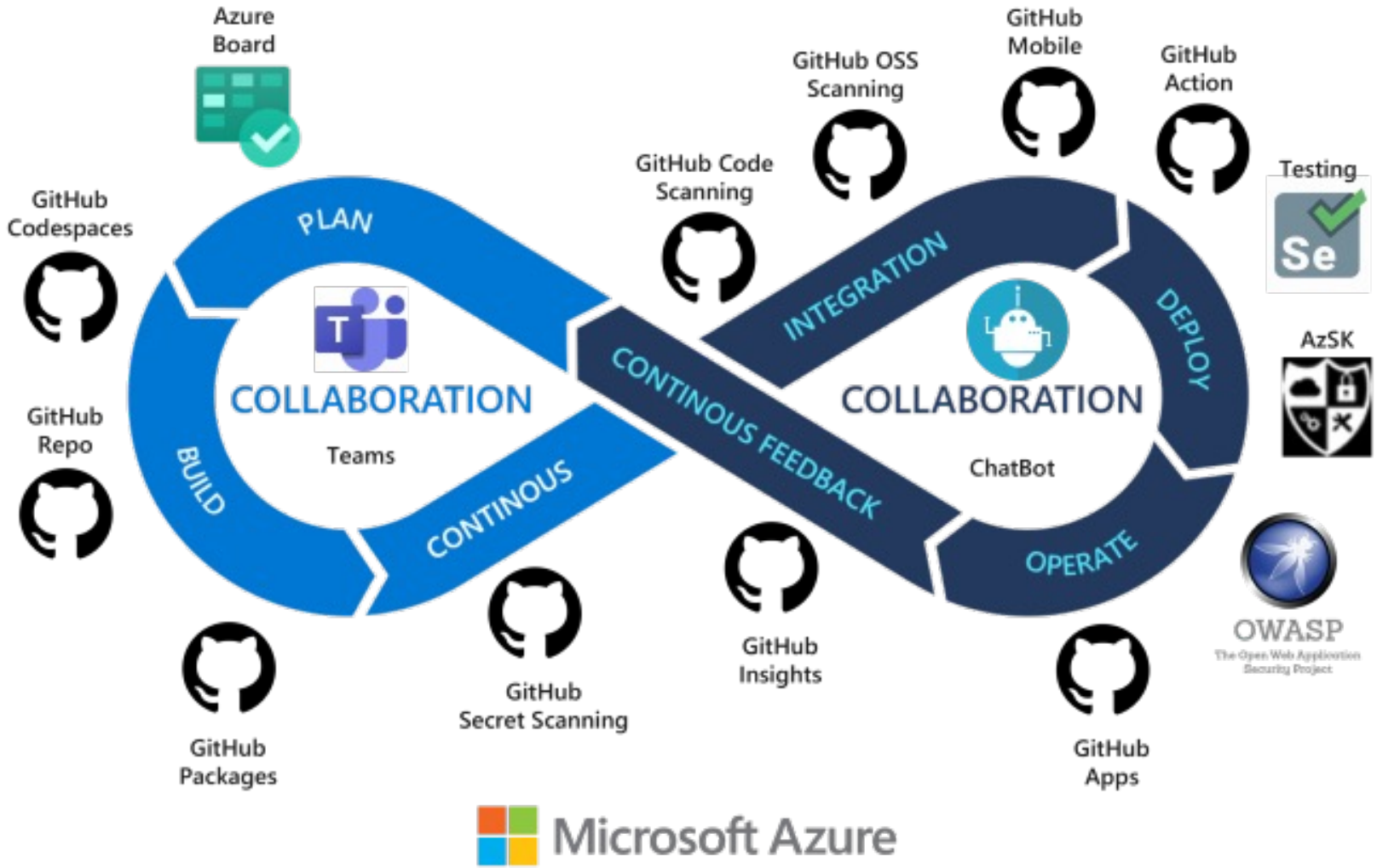
Client Apps

Project Management

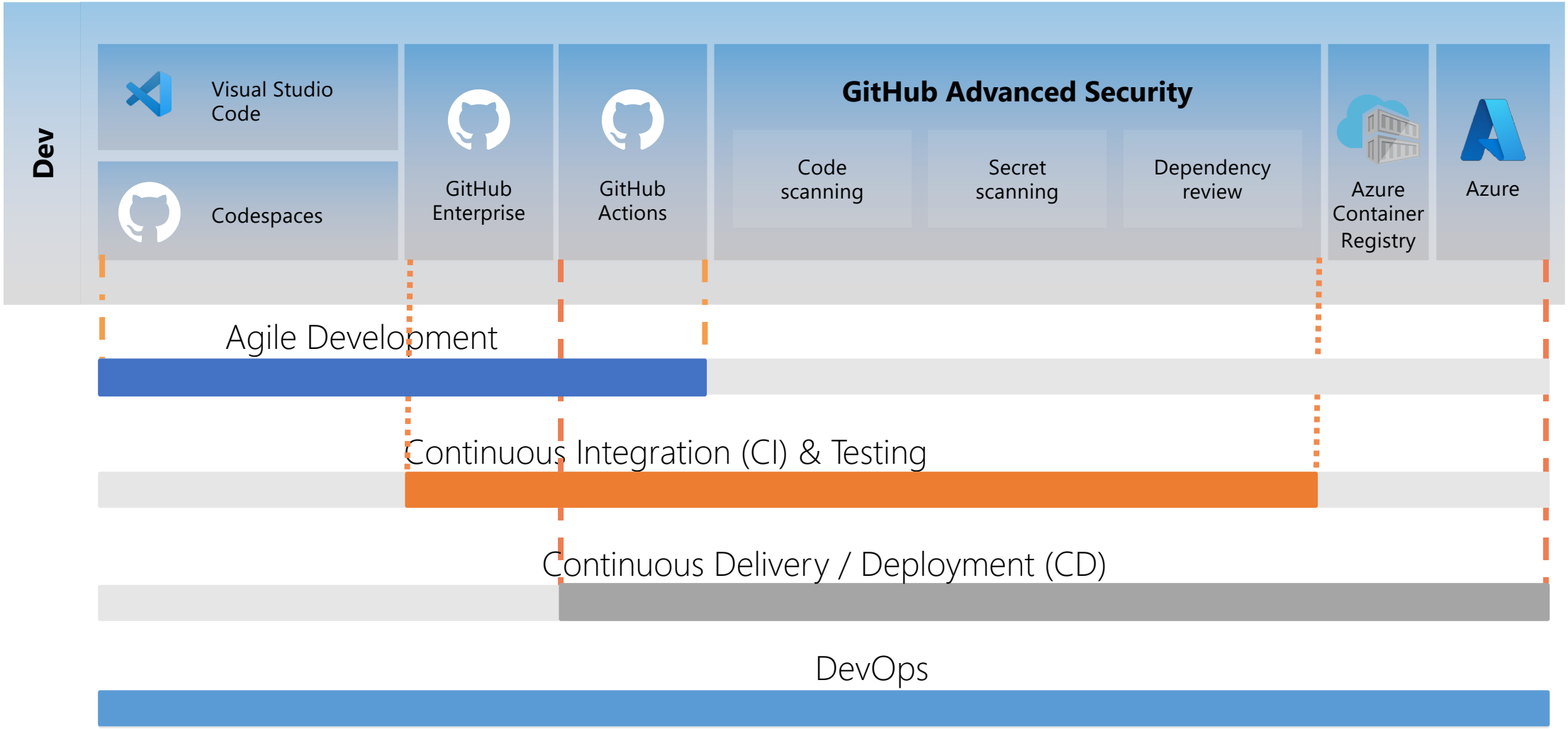
Team Administration

Community

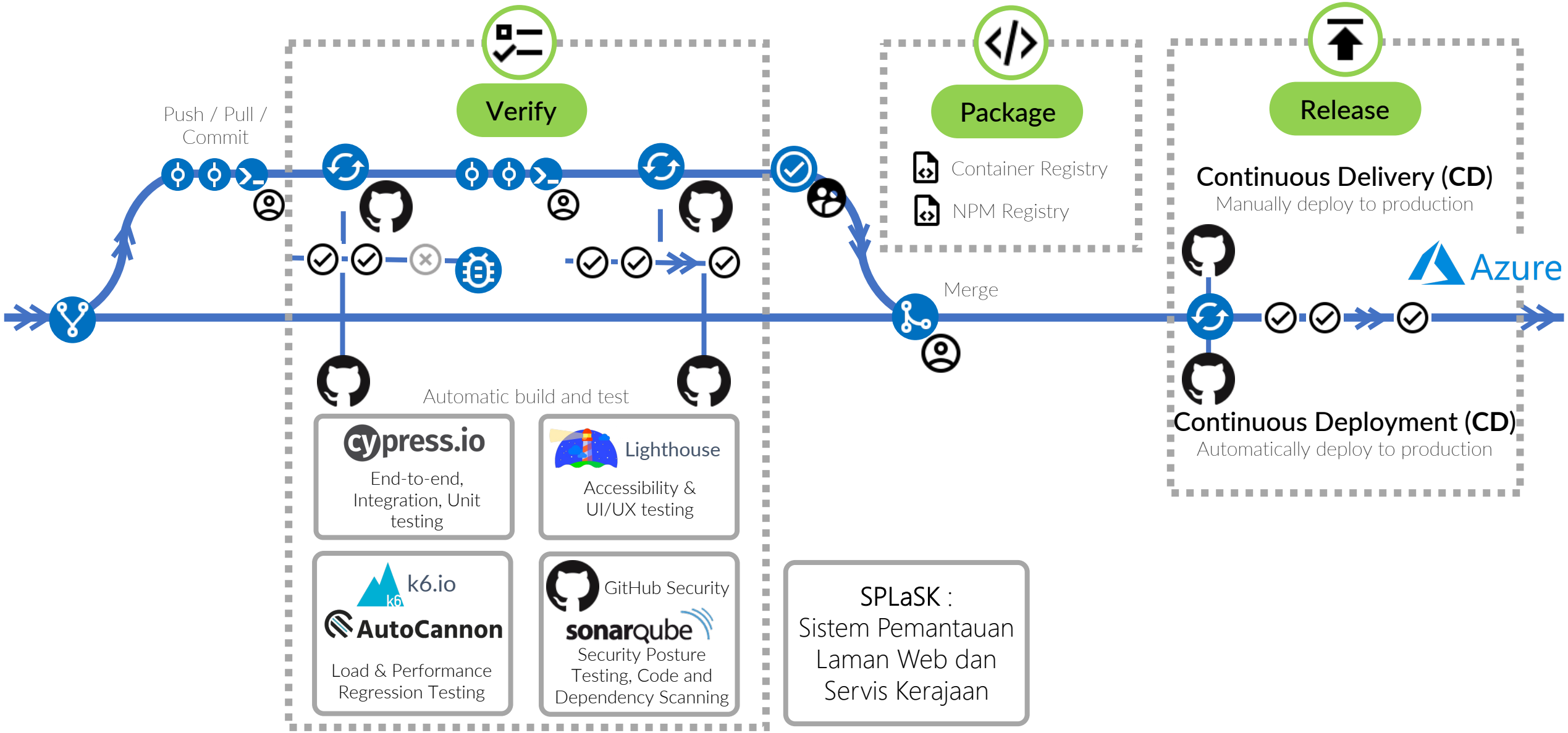
GitHub – Agile DevOps Platform



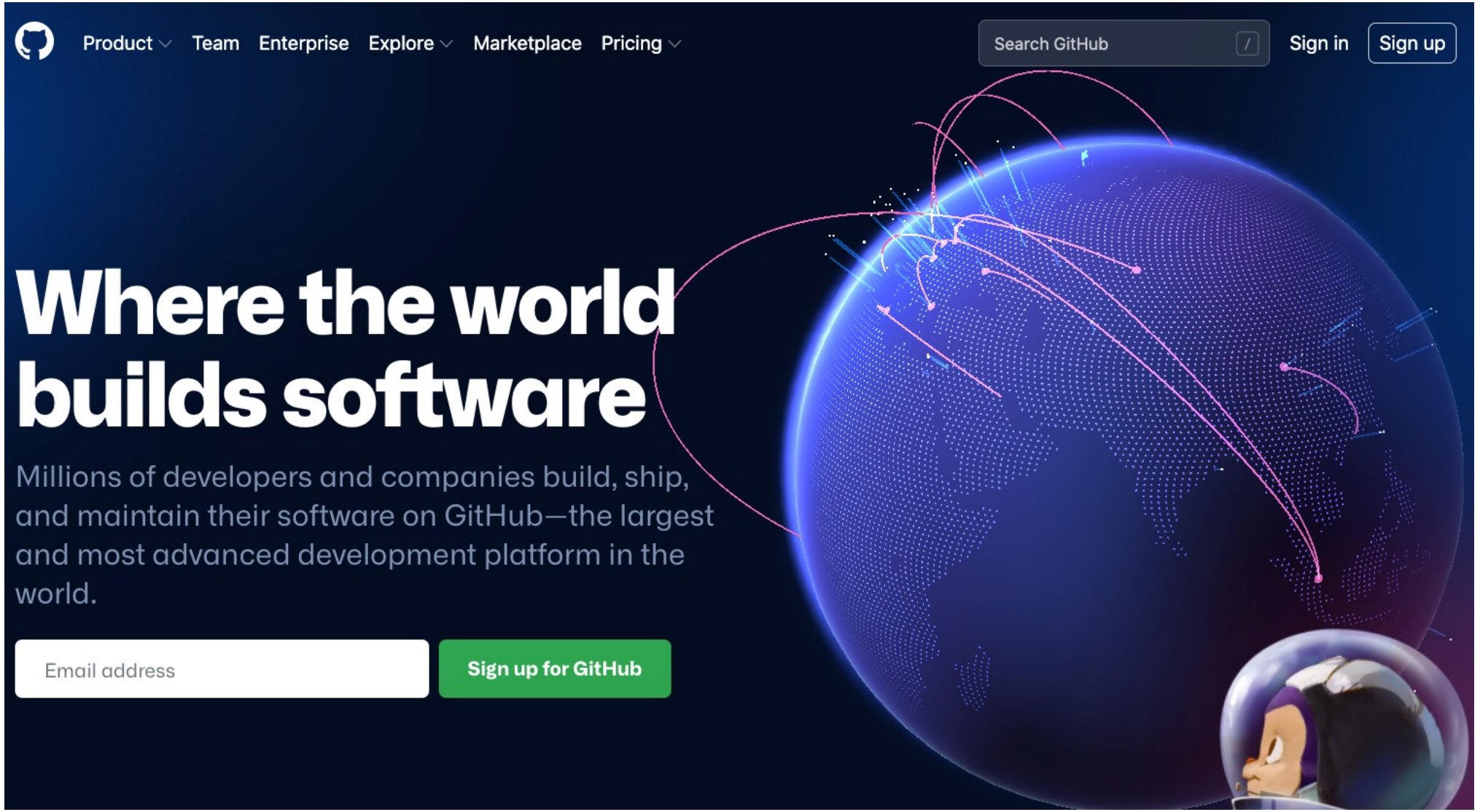
GitHub – DevOps Platform



Modern Process using GitHub



github.com


Product
Team
Enterprise
Explore
Marketplace
Pricing
Search GitHub
/
Sign in
Sign up
Where the world builds software
Millions of developers and companies build, ship, and maintain their software on GitHub—the largest and most advanced development platform in the world.
Email address
Sign up for GitHub

{ Latihan *GitHub Registration & Usage*}

Latihan 1 - Pendaftaran & Penggunaan GitHub

{ Introduction to Git }

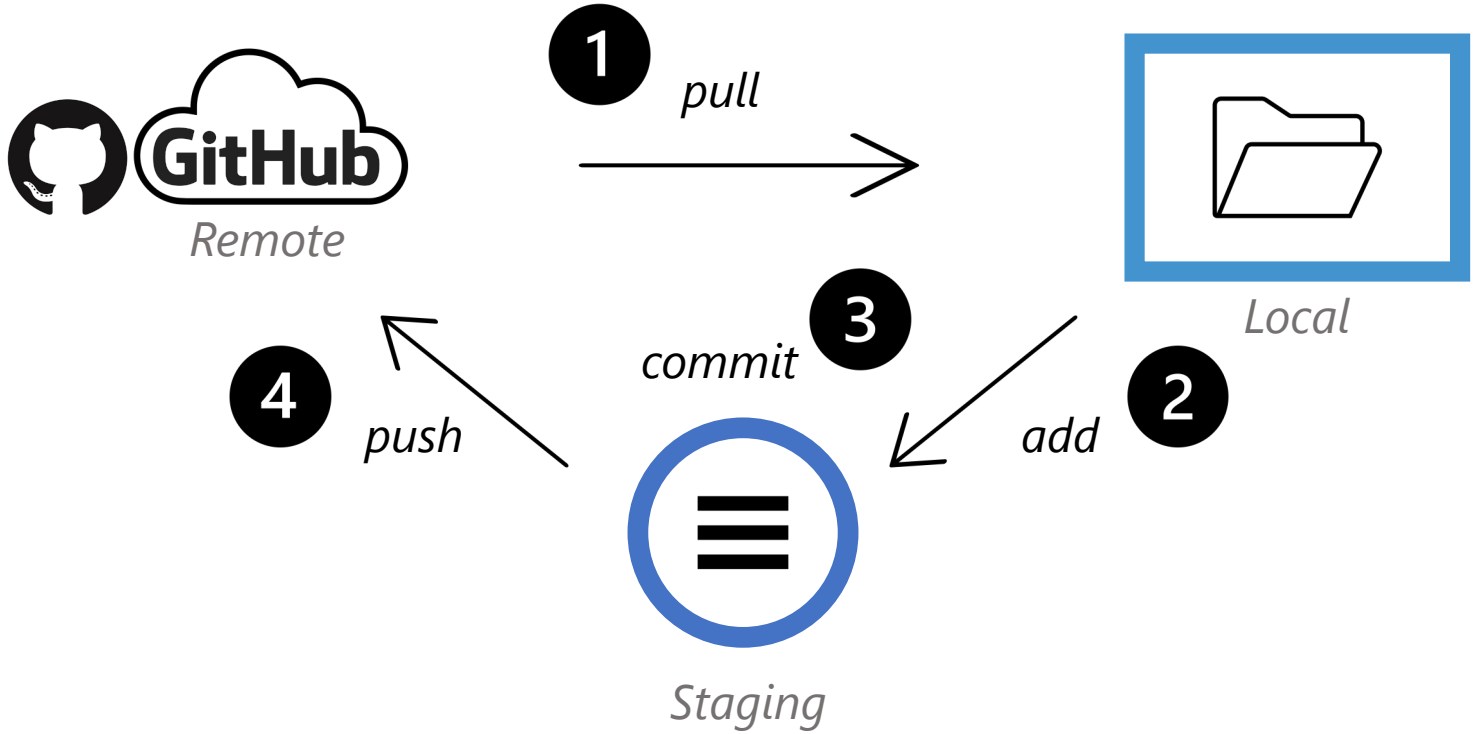
Git

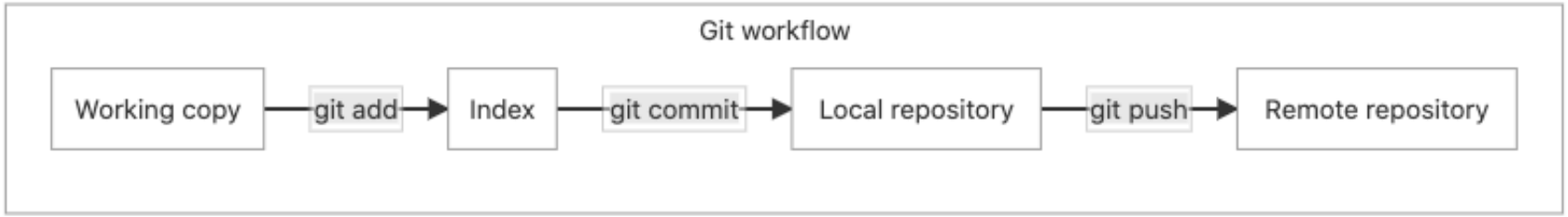
- Git - is a distributed version-control system for tracking changes in source code during software development
- Code Repository – is an archive with the code as well as the hosting facility for these software archives, where you can also have the project's technical documentation, web pages, snippets, patches, etc. which can be accessed publicly (open-source) or privately.

Kenapa Git?

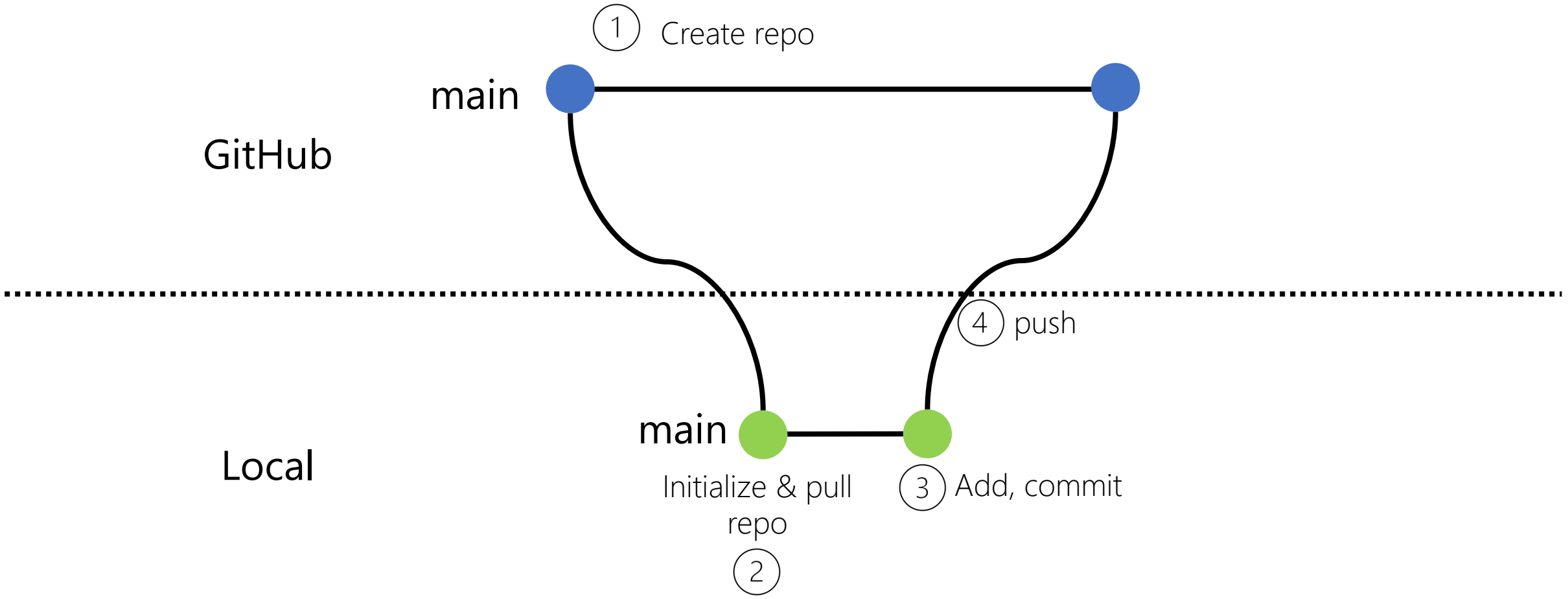
- *Keeps your code safe*
- *Offers version control option to make sure all changes that were done to your code are tracked, and you know who did something to your code. You can also revert back to the previous version of the code before "Everything went bad."*
- *Simplifies the process of unifying changes from developers' collaboration*
- *Provides and promotes teamwork principles since several developers can work together on the same projects, modules, and even code lines*
- *Prepares your code for release to production*
- *Keep the statistics and analytics of the changes in the code*

Langkah-Langkah Git





Skop Latihan



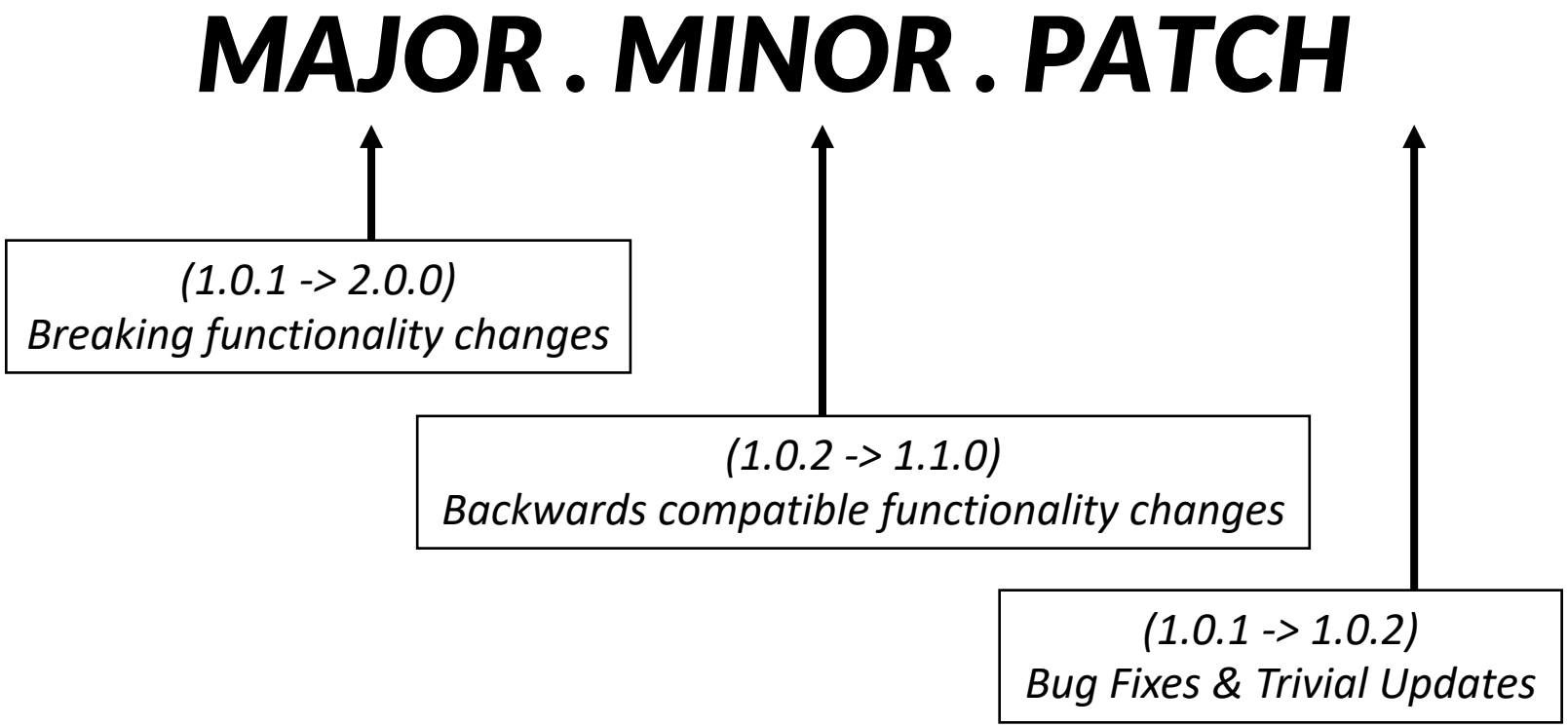
{ Latihan *Git Installation & Git Usage* }

Latihan 2 - Pemasangan dan Penggunaan Git

{ *Git Versioning & Tagging* }

Git Versioning

- *Semantic versioning structure*



Git Versioning

- *Once released, version cannot be changed*
- *Any changes need to be a new release*
- *Double check version number before releasing*
- *Major release should be reviewed and released by Release Manager*
- *Minor & Patch releases should be reviewed and released by Reviewer*

Git Tagging

- *2 types:*
  - *Lightweight tags*
    - *Tags that can be set and removed as needed.*
    - *A basic named pointer to a commit*

      > git tag <tag-name> [commit]
  - *Annotated Tags*
    - *An unchangeable part of Git history - stored as full objects in the Git database*
    - *Full object in the git database that is checksummed; contain the tagger name, email, and date; have a tagging message*

      > git tag –a <tag-name> -m “message” [commit]

Semantic Releases

Semantic Versioning + Annotated Tagging

= Semantic Release

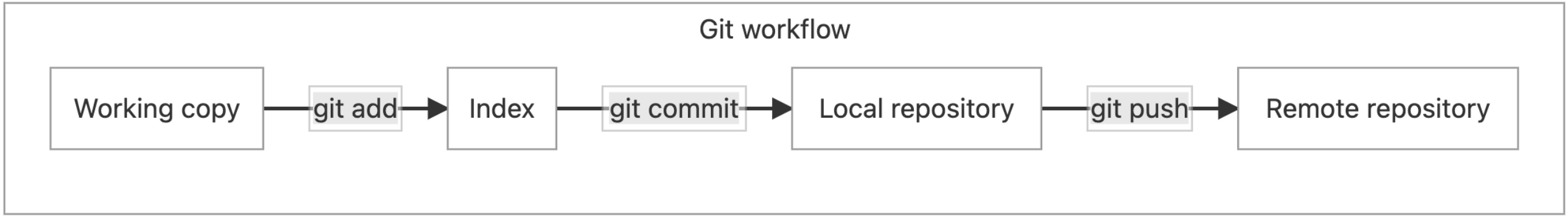
- *Benefits:*
  - *Signifies important changes in your repository*
  - *Communicates degree of changes between tags*
  - *Straightforward way to track history of changes*
  - *CI/CD*

{ Git Branch Management }

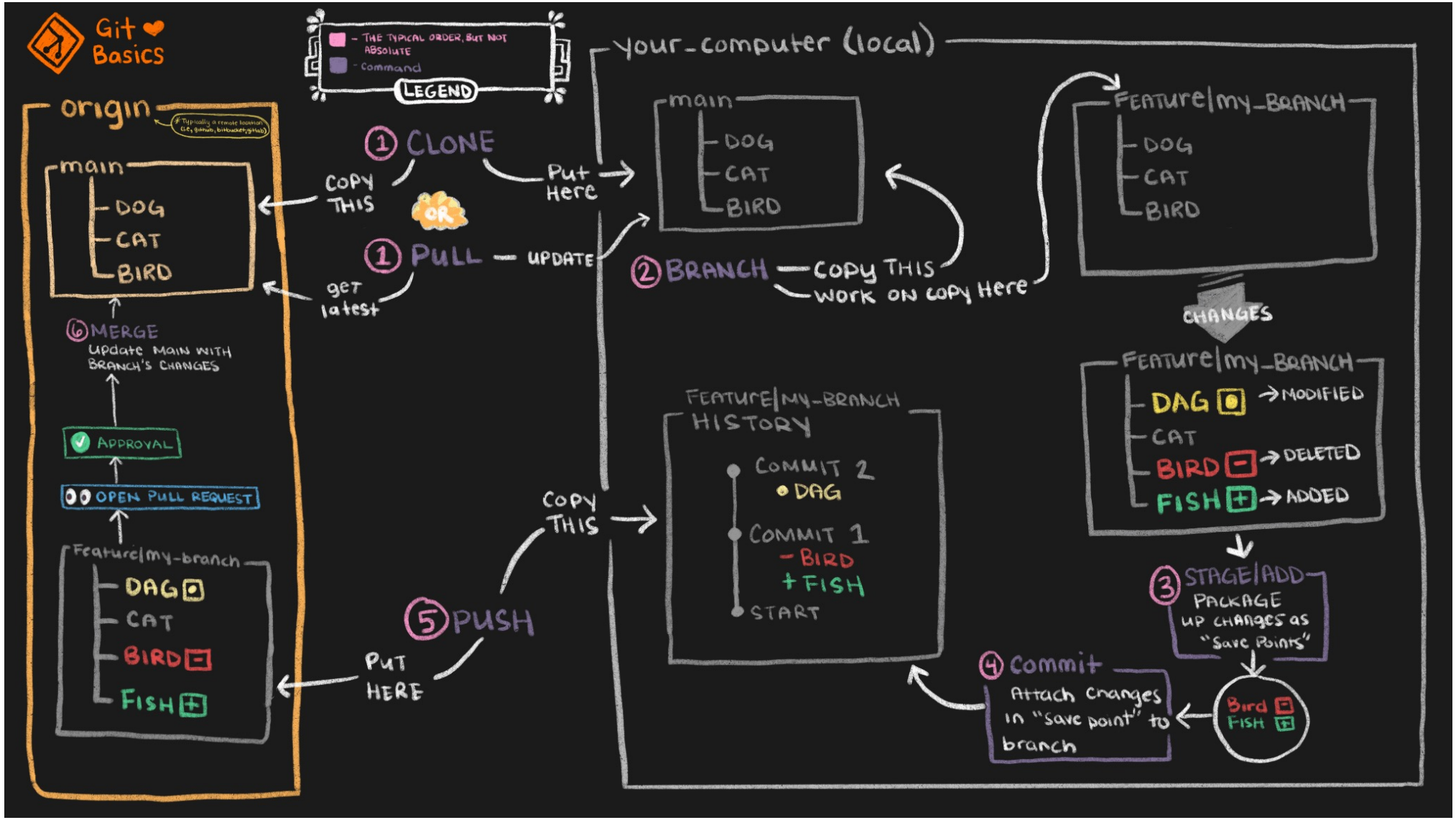
Git *Branching*

- *Branching means you diverge from the main line of development and continue to do work without messing with that main line.*
- *Git branches are effectively a pointer to a snapshot of your changes.*
- *A branch is essentially is a unique set of code changes with a unique name.*
- *The main branch — the one where all changes eventually get merged back into, and is called master.*

Git Workflow



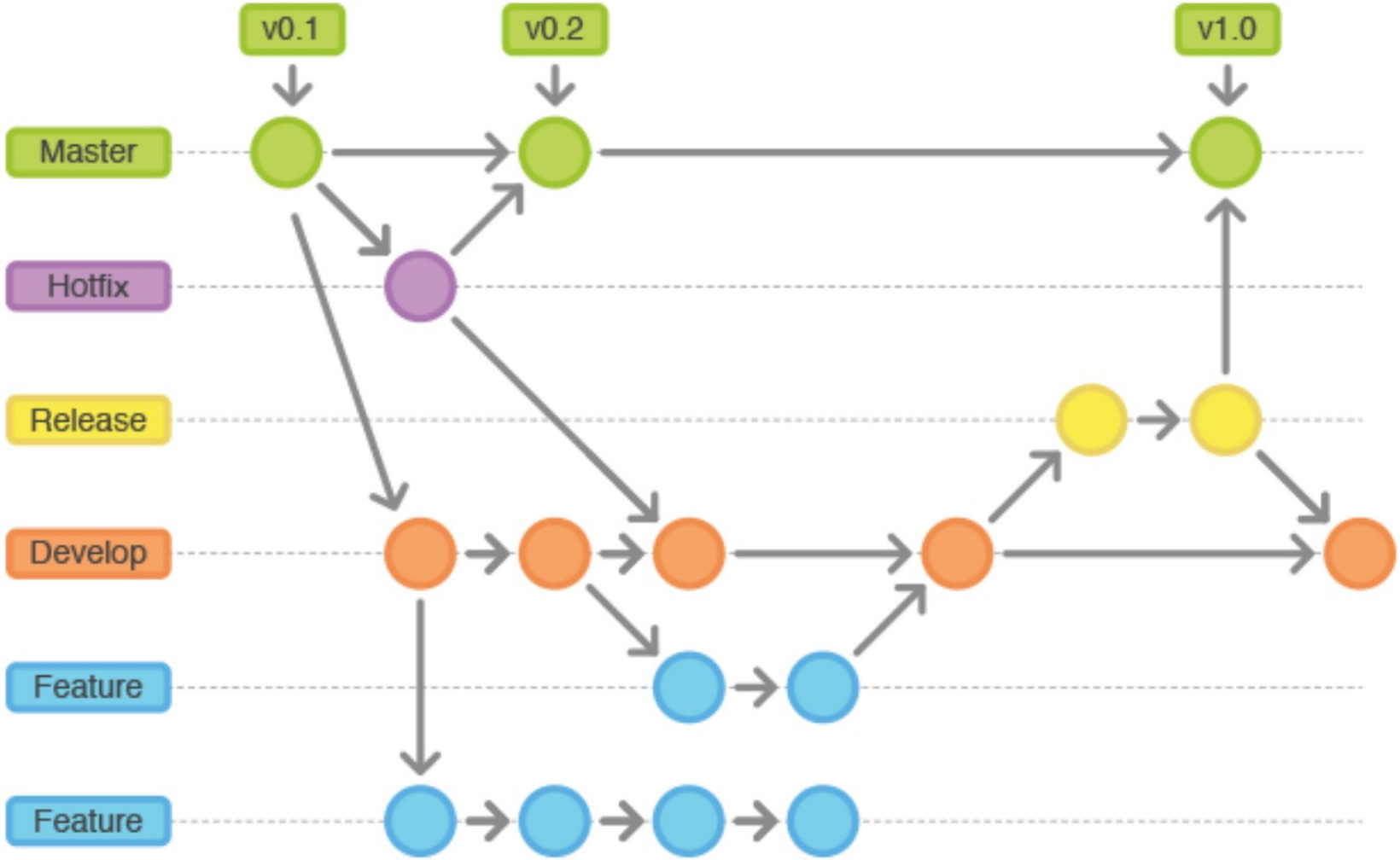
Proses Git


Git ♥ Basics
- THE TYPICAL ORDER, BUT NOT ABSOLUTE
- command
LEGEND
origin
Typically a remote location (Git, GitHub, BitBucket, GitHub)
main
DOG
CAT
BIRD
1 CLONE
COPY THIS
Put Here
OR
1 PULL — UPDATE
get latest
your-computer (local)
main
DOG
CAT
BIRD
2 BRANCH — COPY THIS
work on copy Here
feature/my-BRANCH
DOG
CAT
BIRD
CHANGES
feature/my-BRANCH
DAG → MODIFIED
CAT
BIRD → DELETED
FISH → ADDED
3 STAGE/ADD
PACKAGE UP CHANGES as "Save Points"
Bird
FISH
4 Commit
Attach changes in "save point" to branch
FEATURE/MY-BRANCH HISTORY
COMMIT 2
• DAG
COMMIT 1
- BIRD
+ FISH
START
COPY THIS
5 PUSH
PUT HERE
6 MERGE
Update MAIN WITH BRANCH'S CHANGES
APPROVAL
OPEN PULL REQUEST
feature/my-branch
DAG
CAT
BIRD
FISH

Jenis-jenis *Git workflow*

- *Git Flow*
- *GitHub Flow*
- *GitLab Flow*

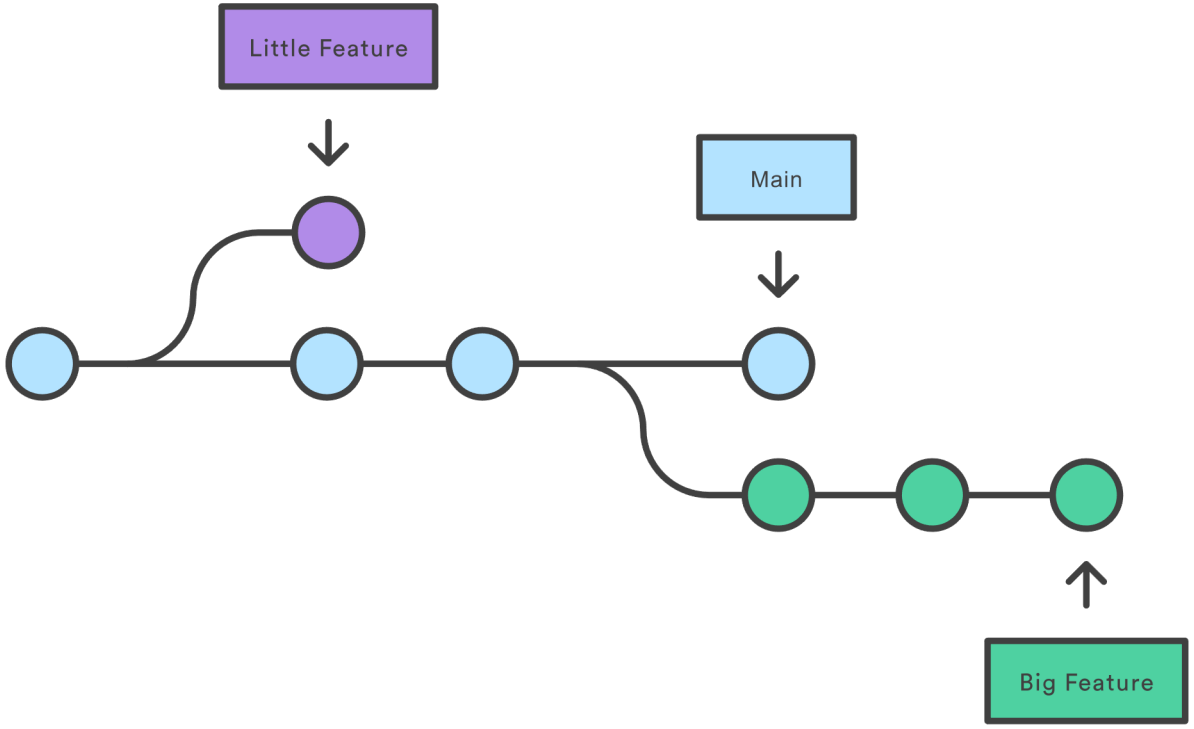
Git Flow

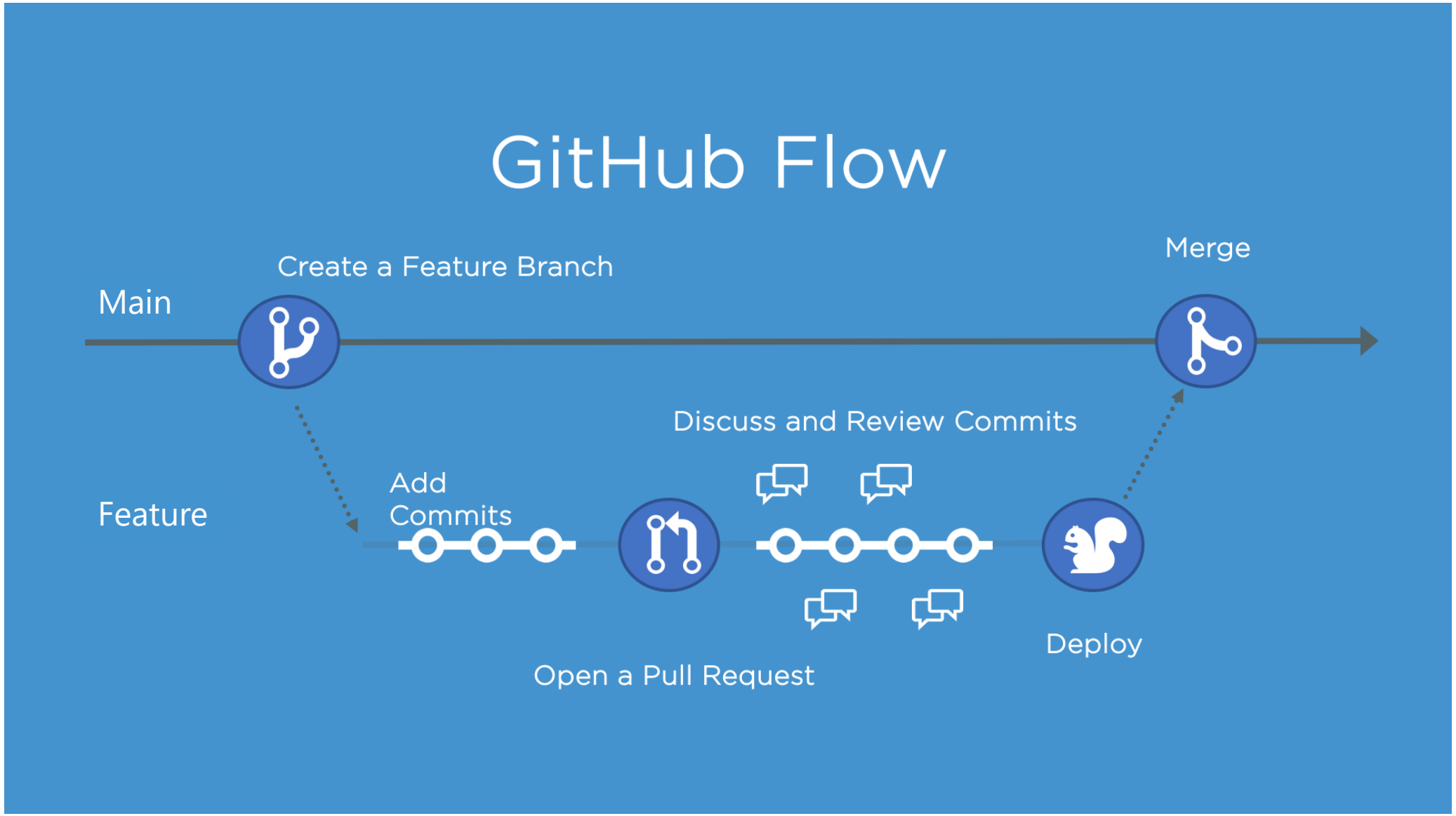


GitHub Flow

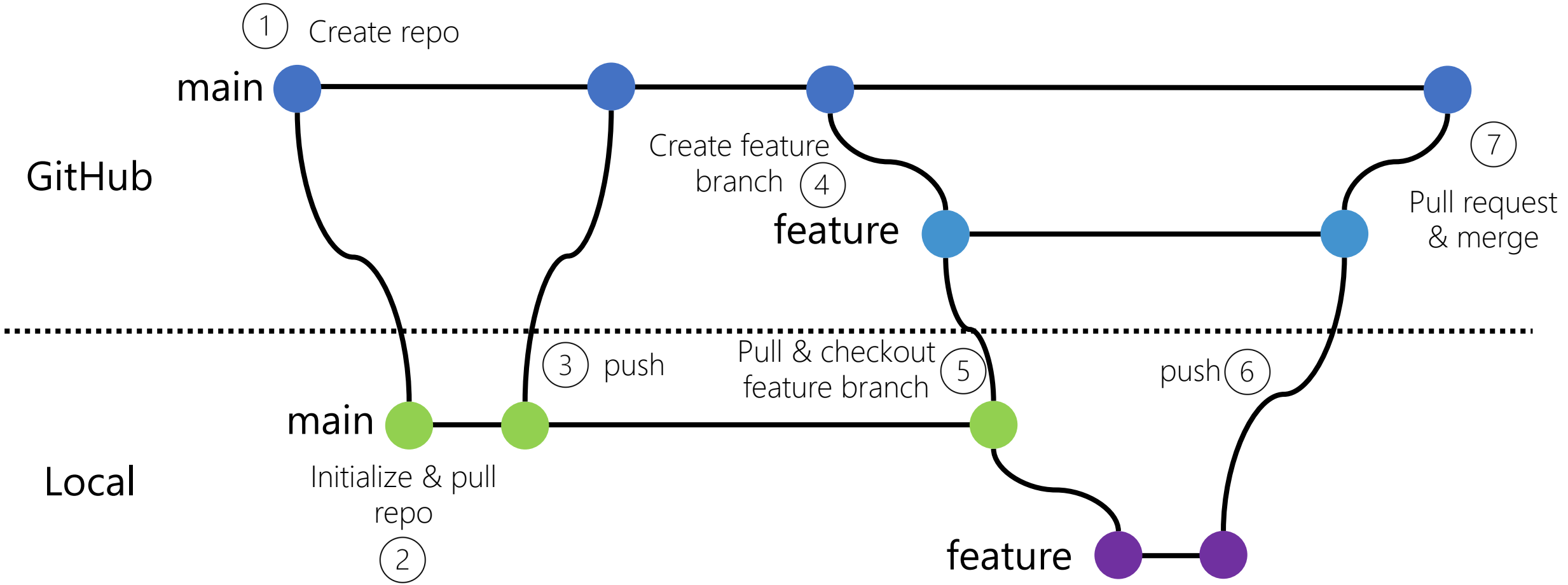
GitHub Flow

- GitHub flow is a lightweight, branch-based workflow.
- Project repo - *github.com/yourname/projectname*.



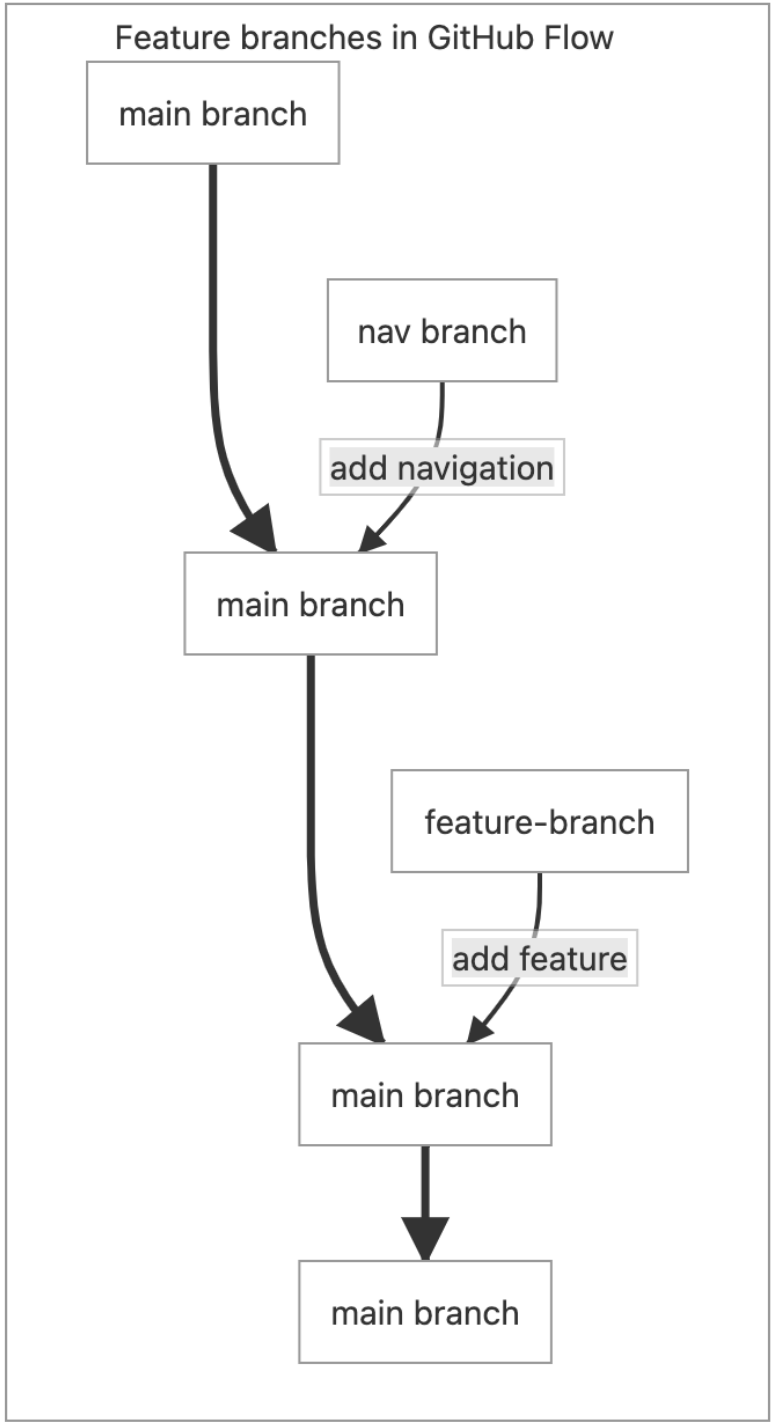

GitHub Flow
Create a Feature Branch
Merge
Main
Discuss and Review Commits
Add
Commits
Feature
Deploy
Open a Pull Request

Skop Latihan



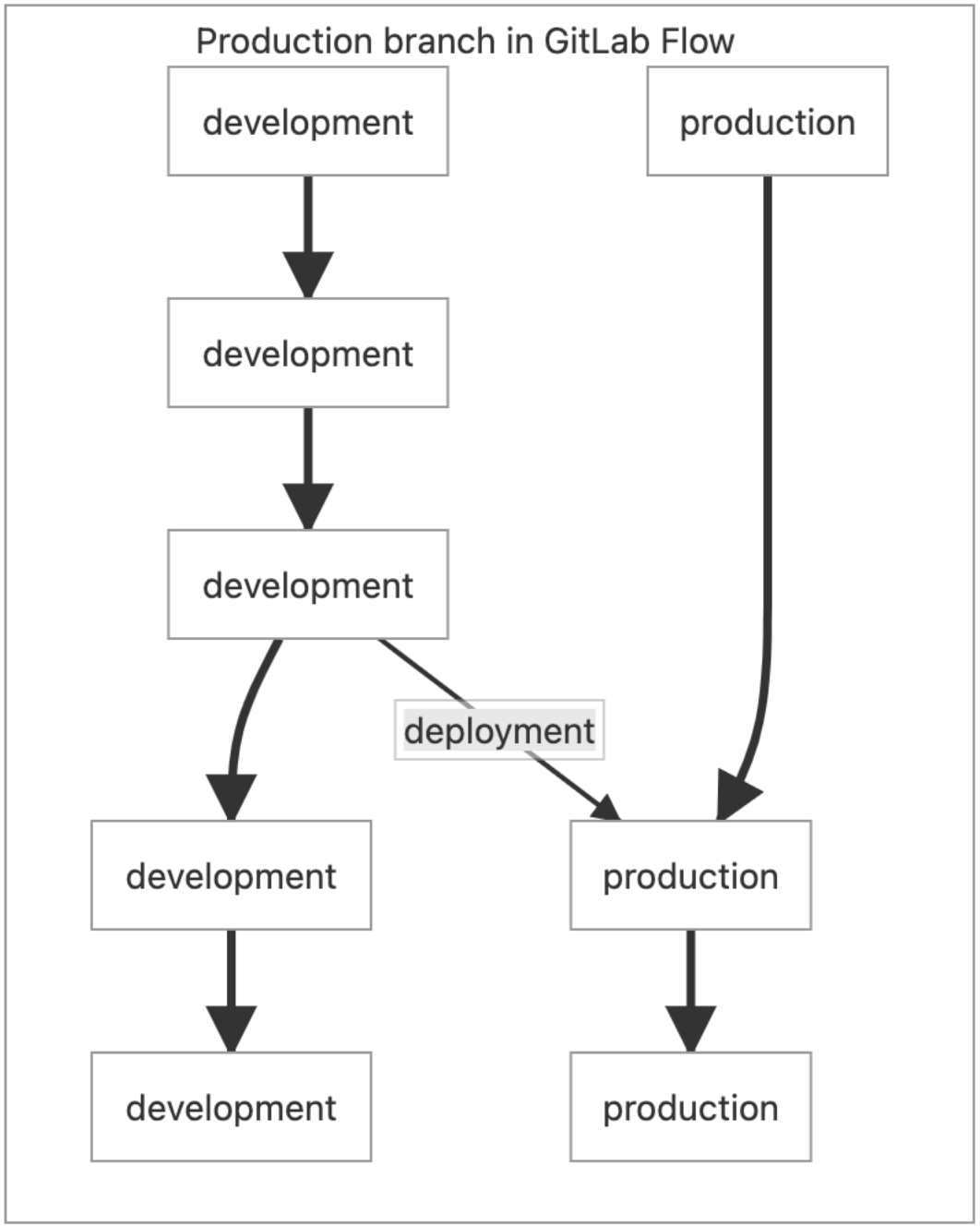
Main

Feature



GitLab Flow

- *GitHub flow assumes you can deploy to production every time you merge a feature branch*
- *there are some cases where this is not possible, such as:*
  - *You don't control the timing of a release. For example, an iOS application that is released when it passes App Store validation.*
  - *You have deployment windows - for example, workdays from 10 AM to 4 PM when the operations team is at full capacity - but you also merge code at other times.*
- *make a production branch that reflects the deployed code.*
- *You can deploy a new version by merging development into the production branch*


Production branch in GitLab Flow
development
production
development
development
deployment
development
production
development
production

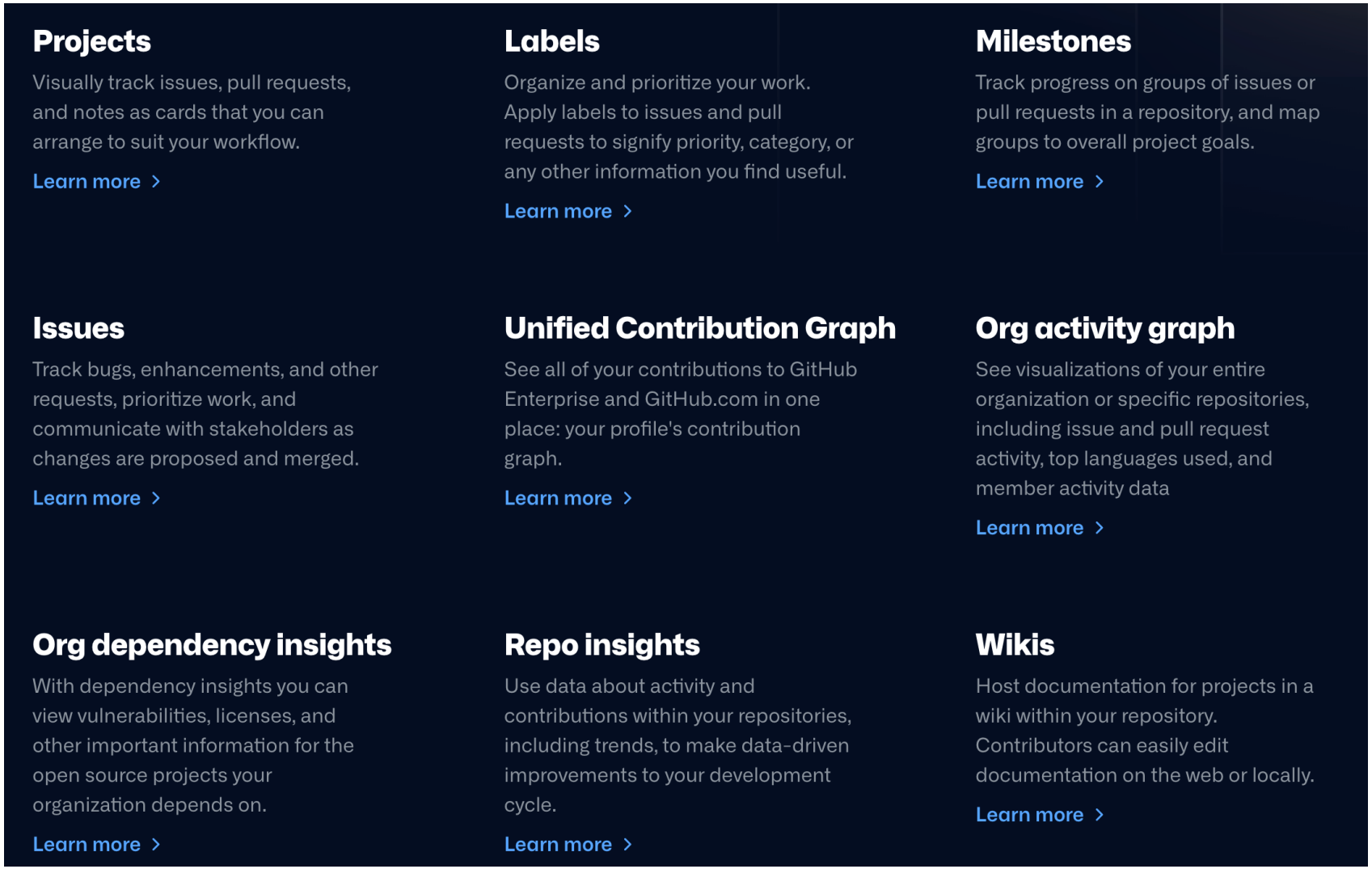
{ Latihan *Git Branching & GitHub Flow* }

Latihan 3 - Git Branching dan GitHub Flow

GitHub Project Management

- *Keep feature requests, bugs, and more organized with GitHub Issues — engineered for software teams.*
- *Coordinate initiatives big and small with project tables, boards, and tasks lists.*
- *Track what you deliver down to the commit.*

GitHub Project Management



Projects

Visually track issues, pull requests, and notes as cards that you can arrange to suit your workflow.

Learn more ›

Labels

Organize and prioritize your work. Apply labels to issues and pull requests to signify priority, category, or any other information you find useful.

Learn more ›

Milestones

Track progress on groups of issues or pull requests in a repository, and map groups to overall project goals.

Learn more ›

Issues

Track bugs, enhancements, and other requests, prioritize work, and communicate with stakeholders as changes are proposed and merged.

Learn more ›

Unified Contribution Graph

See all of your contributions to GitHub Enterprise and GitHub.com in one place: your profile's contribution graph.

Learn more ›

Org activity graph

See visualizations of your entire organization or specific repositories, including issue and pull request activity, top languages used, and member activity data

Learn more ›

Org dependency insights

With dependency insights you can view vulnerabilities, licenses, and other important information for the open source projects your organization depends on.

Learn more ›

Repo insights

Use data about activity and contributions within your repositories, including trends, to make data-driven improvements to your development cycle.

Learn more ›

Wikis

Host documentation for projects in a wiki within your repository. Contributors can easily edit documentation on the web or locally.

Learn more ›

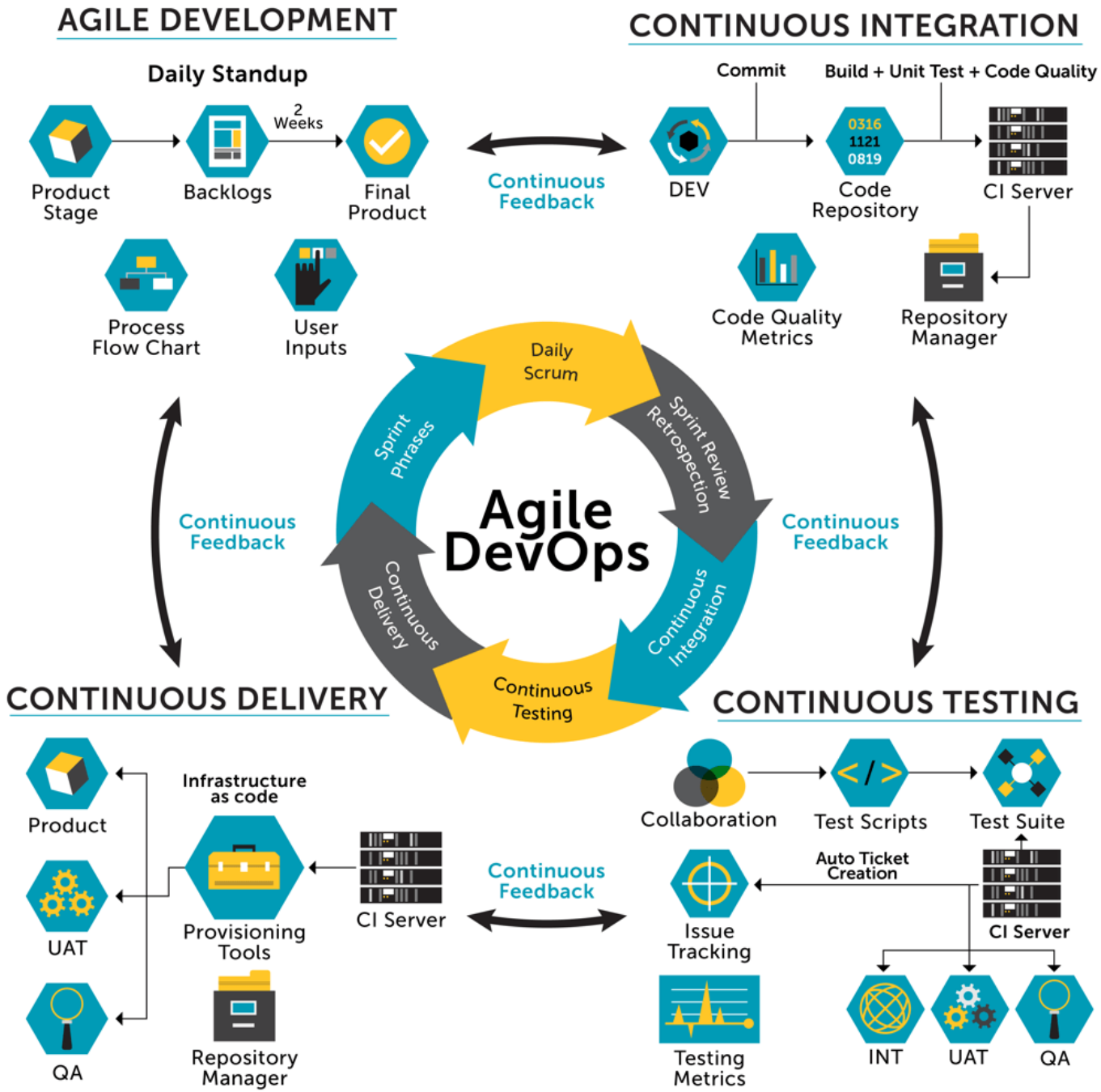
{ *Demo - GitHub Projects, Boards, Views, Issues* }

{ Introduction to GitHub Actions / Workflows / Runners / Security }

{ DevOps Continuous Integration, Continuous Testing, Continuous Delivery / Deployment }

Modern Proses

- AGILE Development methodology
  - a practice that promotes continuous iteration of development and testing throughout the software development lifecycle of the project
- Continuous integration (CI)
  - Software development practice in which small adjustments to the underlying code in an application are tested every time a team member makes changes
- Continuous delivery / deployment (CD)
  - an extension of continuous integration - focuses on automating the software delivery process
- Continuous Testing
  - the process of executing automated tests as part of the software delivery pipeline in order to obtain feedback on the business risks associated with a software release candidate as rapidly as possible.



What is CI/CD?

- In software engineering, CI/CD or CICD is the combined practices of continuous integration and either continuous delivery or continuous deployment.
- CI/CD bridges the gaps between development and operation activities and teams by enforcing automation in building, testing and deployment of applications.
- Modern day DevOps practices involve continuous development, continuous testing, continuous integration, continuous deployment and continuous monitoring of software applications throughout its agile development life cycle.
- The CI/CD practice, or CI/CD pipeline, forms the backbone of modern day DevOps operations.

- Wikipedia

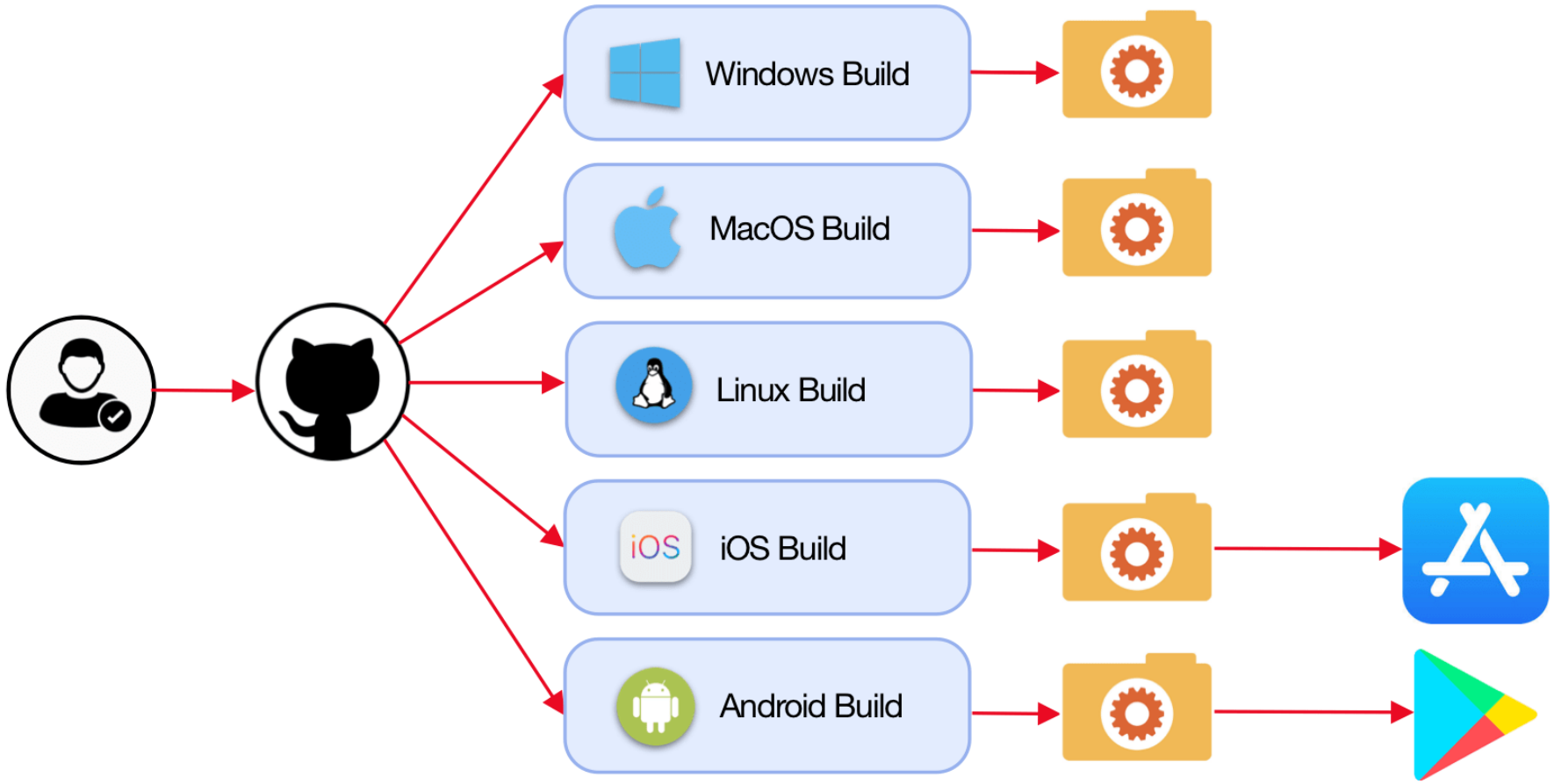
Continuous integration (CI)

- Software development practice in which small adjustments to the underlying code in an application are tested every time a team member makes changes.

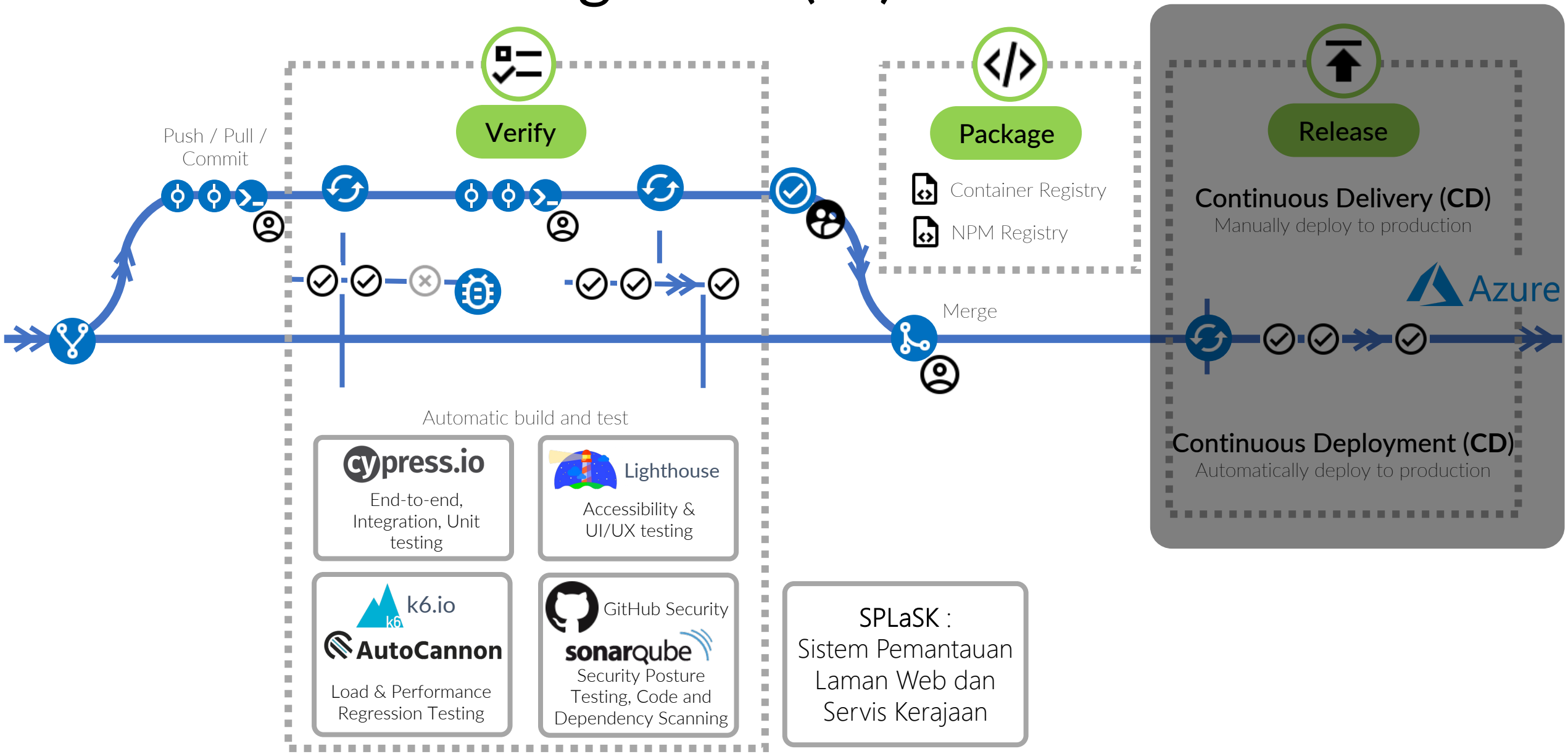
GitHub CI Tool

GitHub Actions

GitHub CI – Build Process

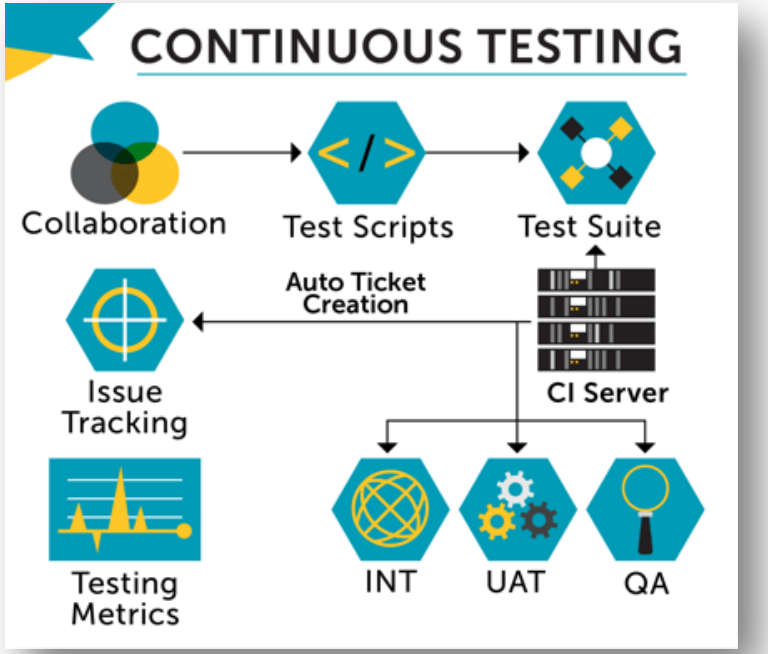


Continuous integration (CI)



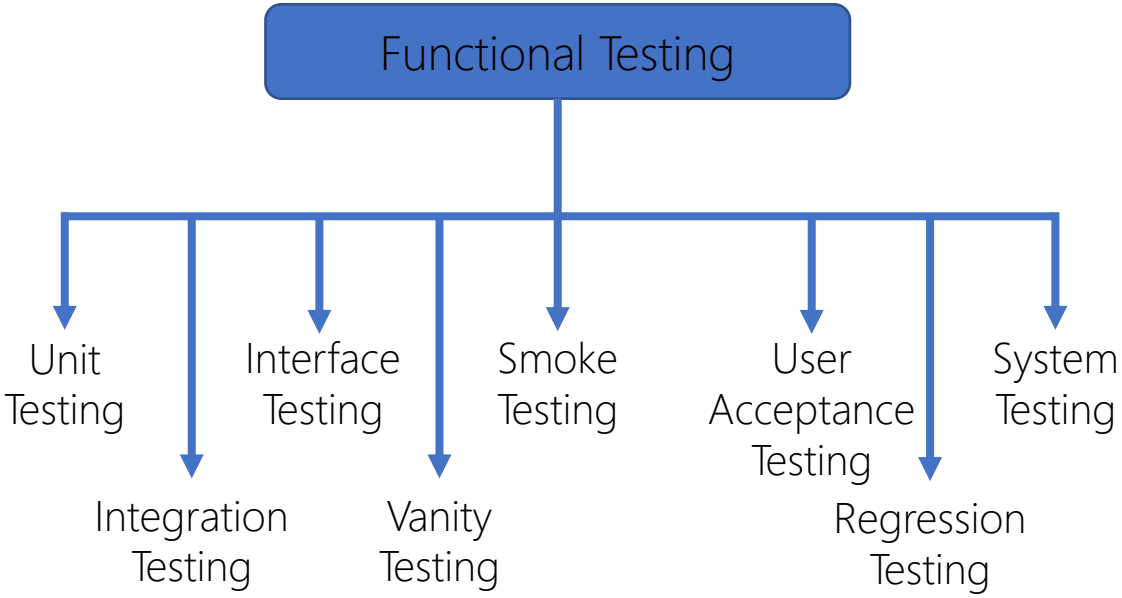
Continuous Testing

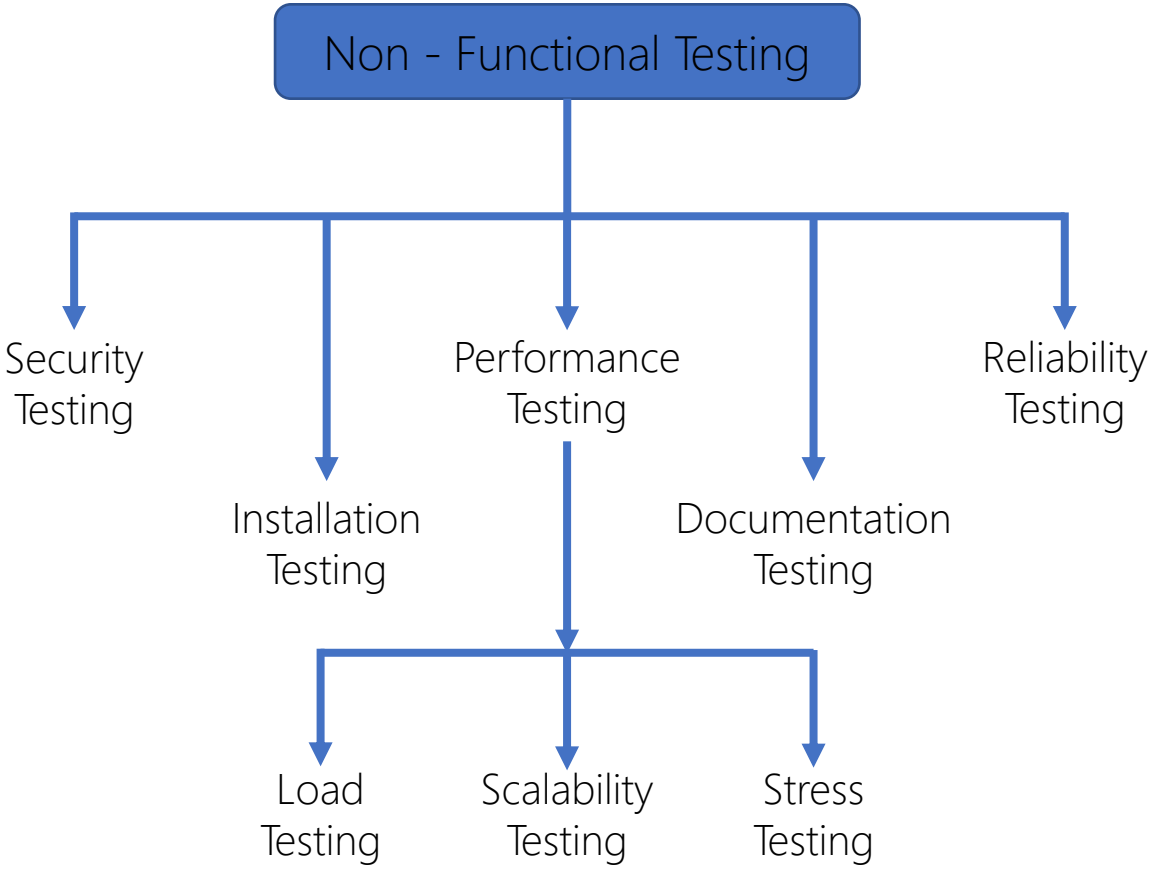
- the process of executing automated tests as part of the software delivery pipeline in order to obtain feedback on the business risks associated with a software release candidate as rapidly as possible.



Continuous Testing

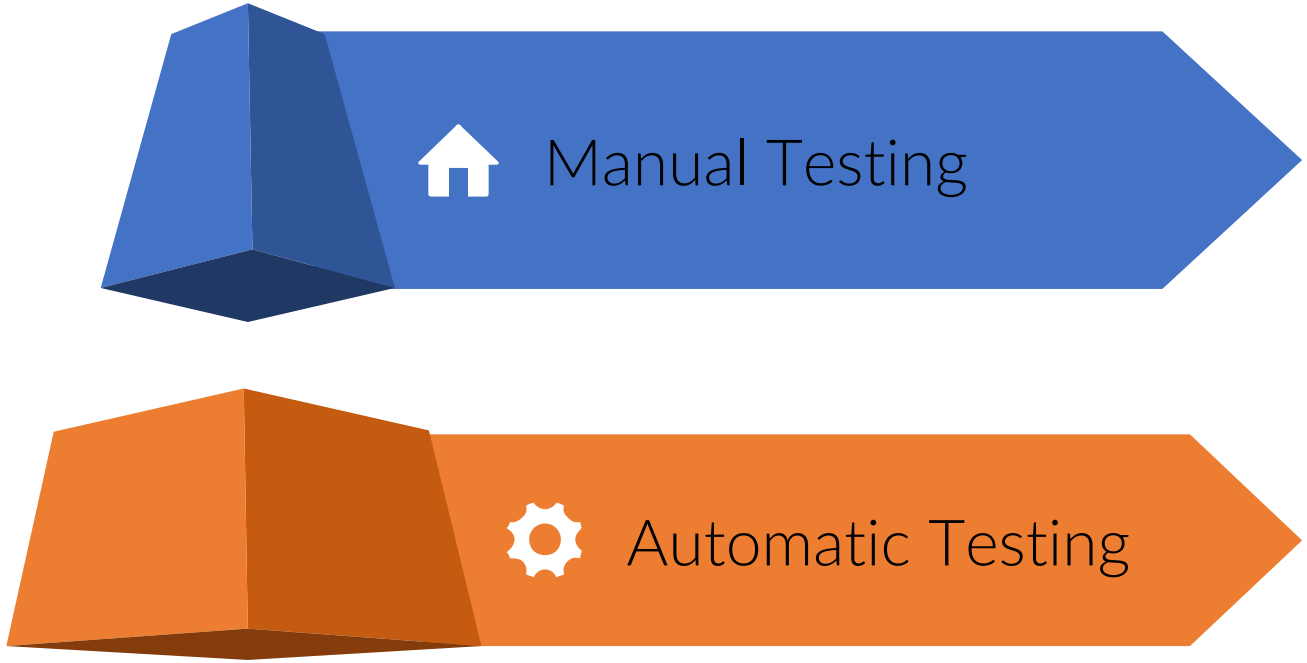
Software Testing Types





Continuous Testing

Testing Execution Approach



Continuous Testing

Manual Testing Process

Step 06: Defect Fix

Step 05: Defect Logging

Step 04: Test Case Execution

Step 01: Requirement Analysis

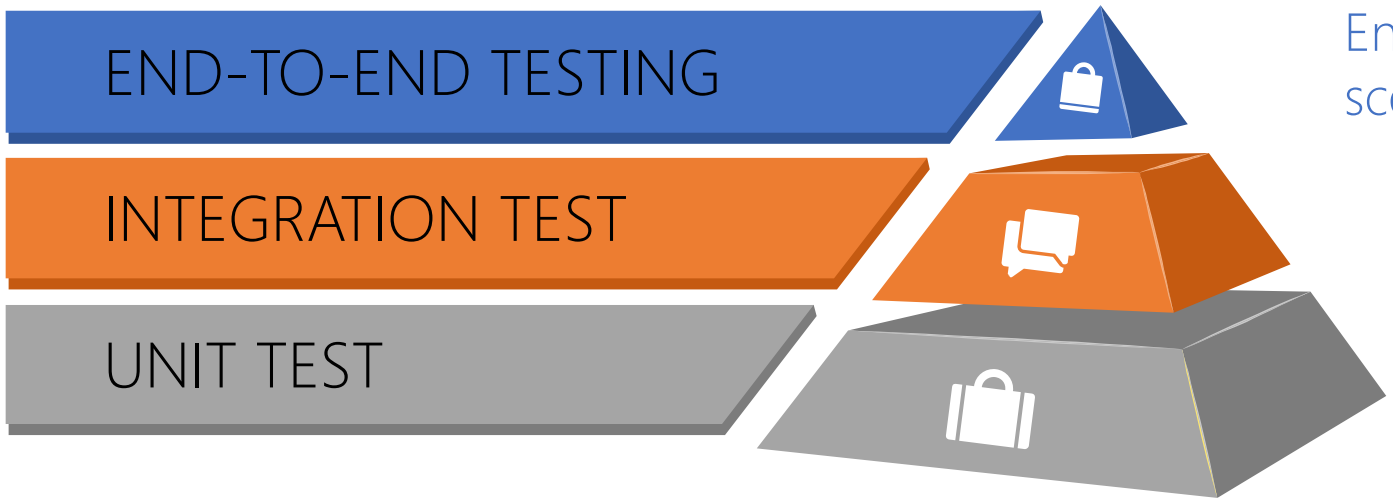
Step 02: Test Plan Creation

Step 03: Test Case Creation

Automatic Testing using cypress.io

- What is cypress.io?
  - Cypress is a next generation front end testing tool built for the modern web applications.
- Type of testings:
  - End-to-end tests
  - Integration tests
  - Unit tests

cypress.io



Entire application is tested n a real-world scenario

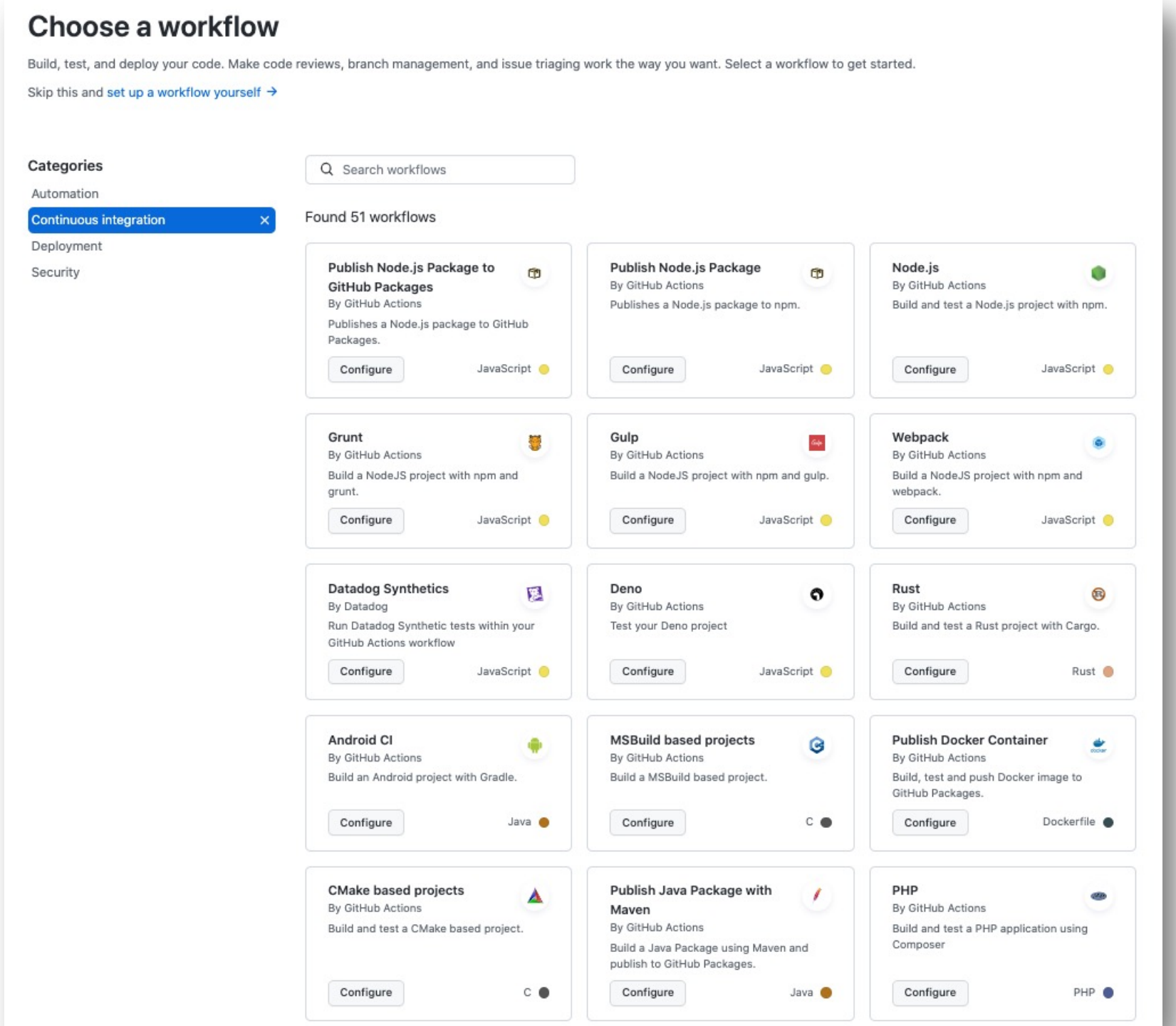
Program units are combined and tested as groups in multiple ways

Testing code in isolated small pieces

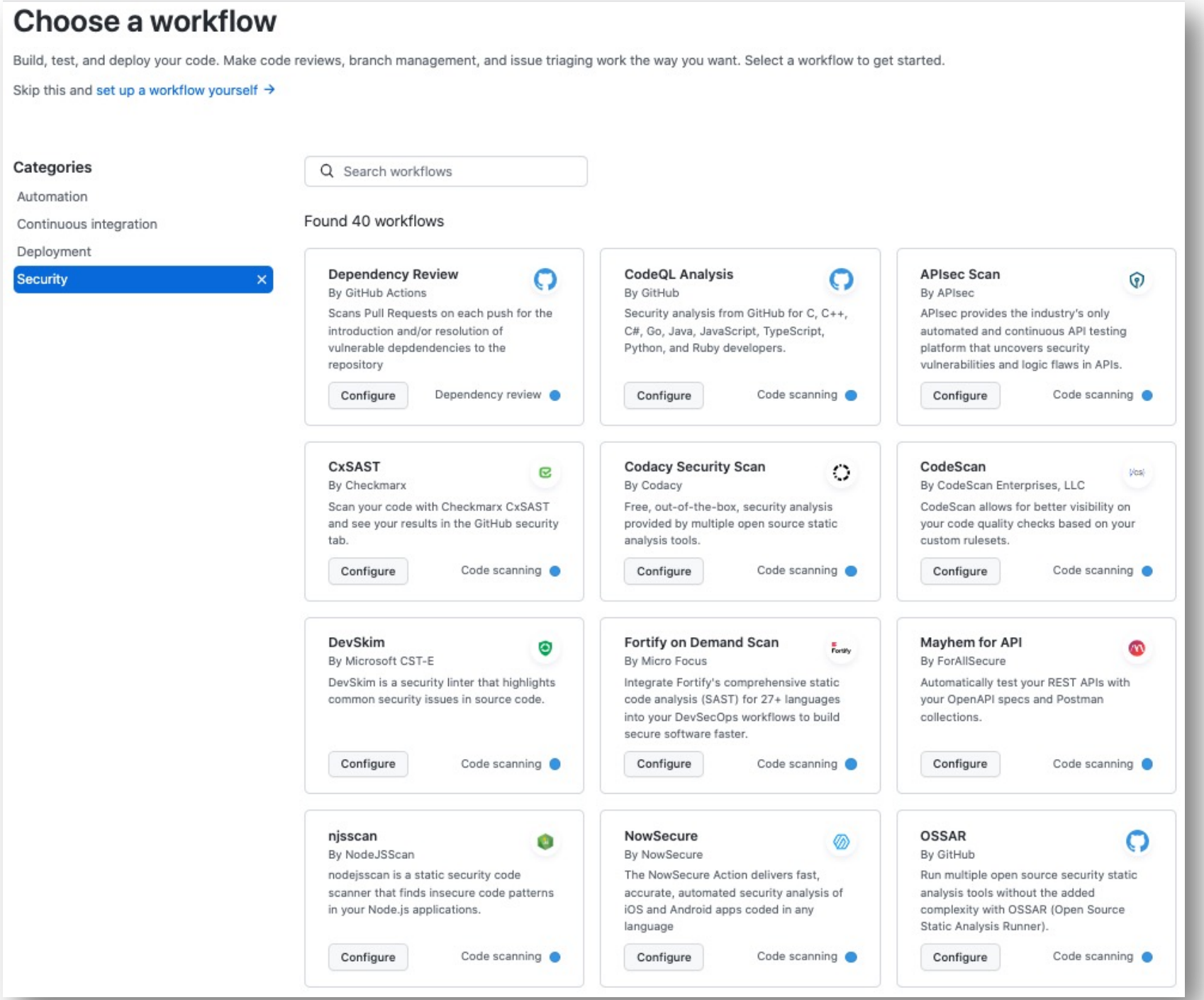
GitHub Actions - CI

- Create CI using GitHub Actions workflow.
- A workflow is a configurable automated process made up of one or more jobs.
- Create a YAML file to define the workflow configuration.

GitHub CI


Choose a workflow
Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow to get started.
Skip this and set up a workflow yourself →
Categories
Automation
Continuous integration
Deployment
Security
Search workflows
Found 51 workflows
Publish Node.js Package to GitHub Packages
By GitHub Actions
Publishes a Node.js package to GitHub Packages.
Configure
JavaScript
Publish Node.js Package
By GitHub Actions
Publishes a Node.js package to npm.
Configure
JavaScript
Node.js
By GitHub Actions
Build and test a Node.js project with npm.
Configure
JavaScript
Grunt
By GitHub Actions
Build a NodeJS project with npm and grunt.
Configure
JavaScript
Gulp
By GitHub Actions
Build a NodeJS project with npm and gulp.
Configure
JavaScript
Webpack
By GitHub Actions
Build a NodeJS project with npm and webpack.
Configure
JavaScript
Datadog Synthetics
By Datadog
Run Datadog Synthetic tests within your GitHub Actions workflow
Configure
JavaScript
Deno
By GitHub Actions
Test your Deno project
Configure
JavaScript
Rust
By GitHub Actions
Build and test a Rust project with Cargo.
Configure
Rust
Android CI
By GitHub Actions
Build an Android project with Gradle.
Configure
Java
MSBuild based projects
By GitHub Actions
Build a MSBuild based project.
Configure
C
Publish Docker Container
By GitHub Actions
Build, test and push Docker image to GitHub Packages.
Configure
Dockerfile
CMake based projects
By GitHub Actions
Build and test a CMake based project.
Configure
C
Publish Java Package with Maven
By GitHub Actions
Build a Java Package using Maven and publish to GitHub Packages.
Configure
Java
PHP
By GitHub Actions
Build and test a PHP application using Composer
Configure
PHP

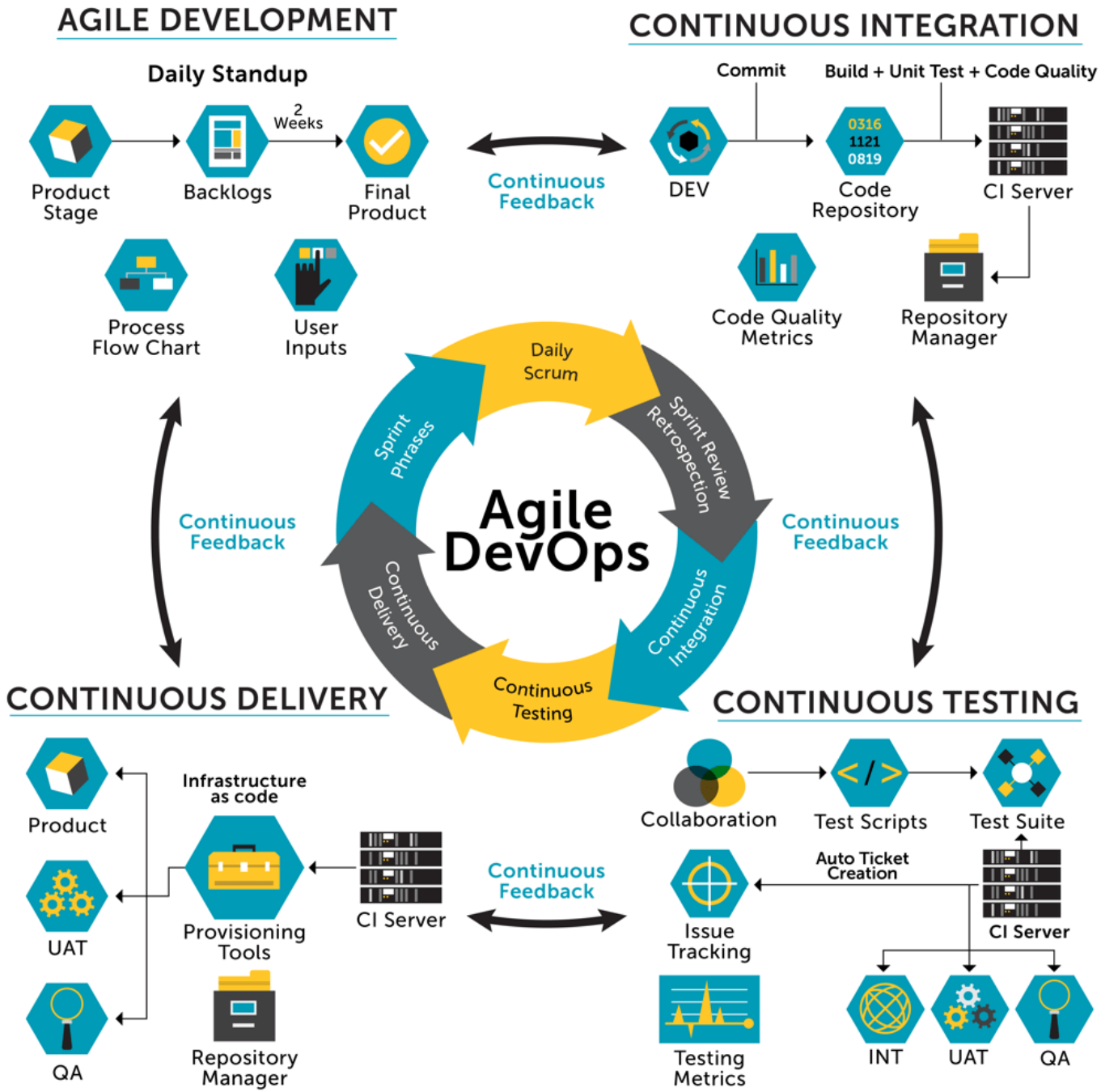
GitHub CI - Security



{ GitHub CI Demo }

Modern Proses

- AGILE Development methodology
  - a practice that promotes continuous iteration of development and testing throughout the software development lifecycle of the project
- Continuous integration (CI)
  - Software development practice in which small adjustments to the underlying code in an application are tested every time a team member makes changes
- Continuous delivery / deployment (CD)
  - an extension of continuous integration - focuses on automating the software delivery process
- Continuous Testing
  - the process of executing automated tests as part of the software delivery pipeline in order to obtain feedback on the business risks associated with a software release candidate as rapidly as possible.



Continuous Delivery / Deployment (CD)

- an extension of continuous integration - focuses on automating the software delivery and deployment process
- Continuous Delivery - Manually deploy to production
- Continuous Deployment - Automatically deploy to production

Continuous Delivery (CD)

- Continuous Delivery is when your code is always ready to be released but isn’t pushed to production unless you make the decision to do so. It is a manual step.
- With Continuous Deployment, any updated working version of the app is automatically pushed to production.

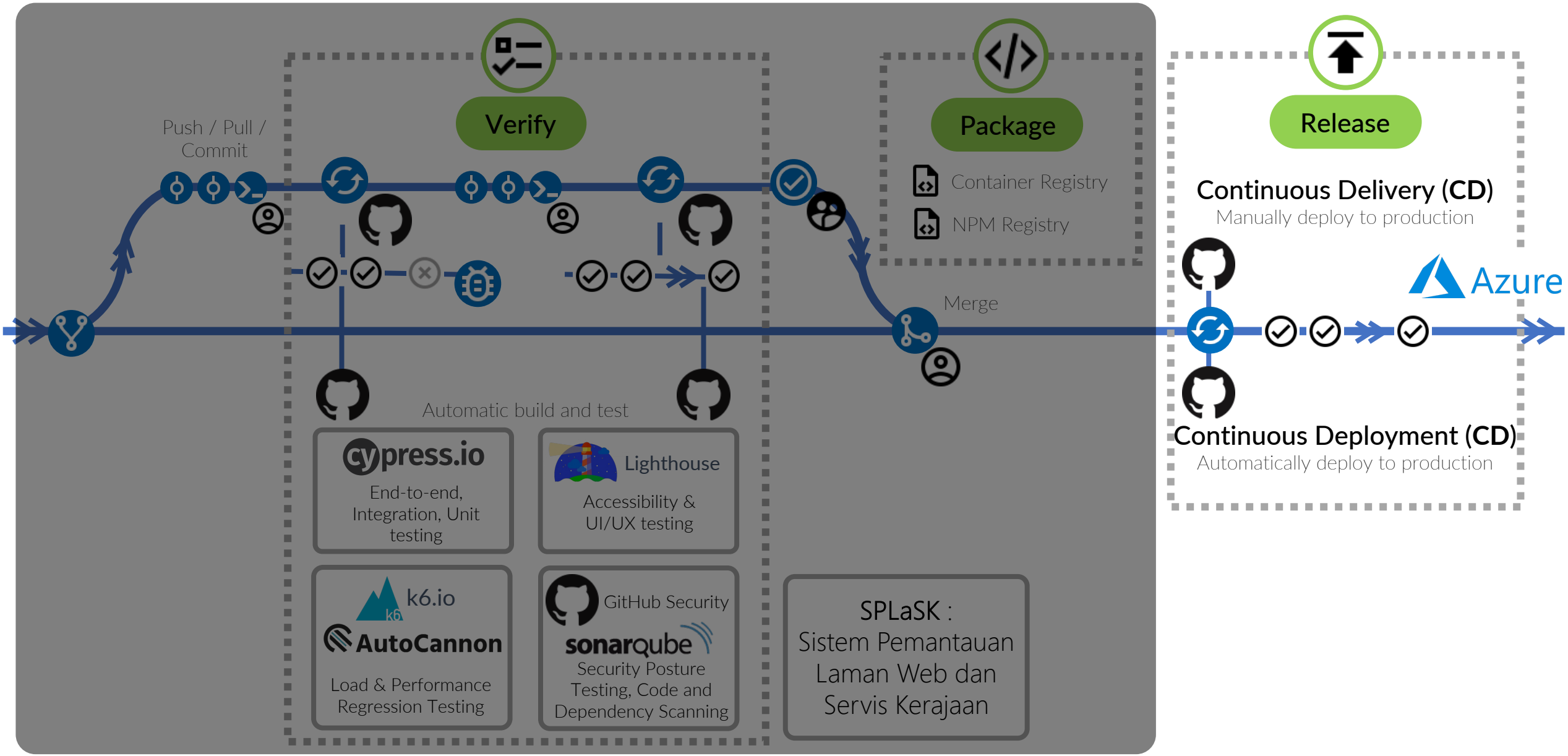
Continuous Deployment (CD)

- Continuous deployment is the process by which qualified changes in software code or architecture are deployed to production as soon as they are ready.

GitHub CD Tool

GitHub Actions

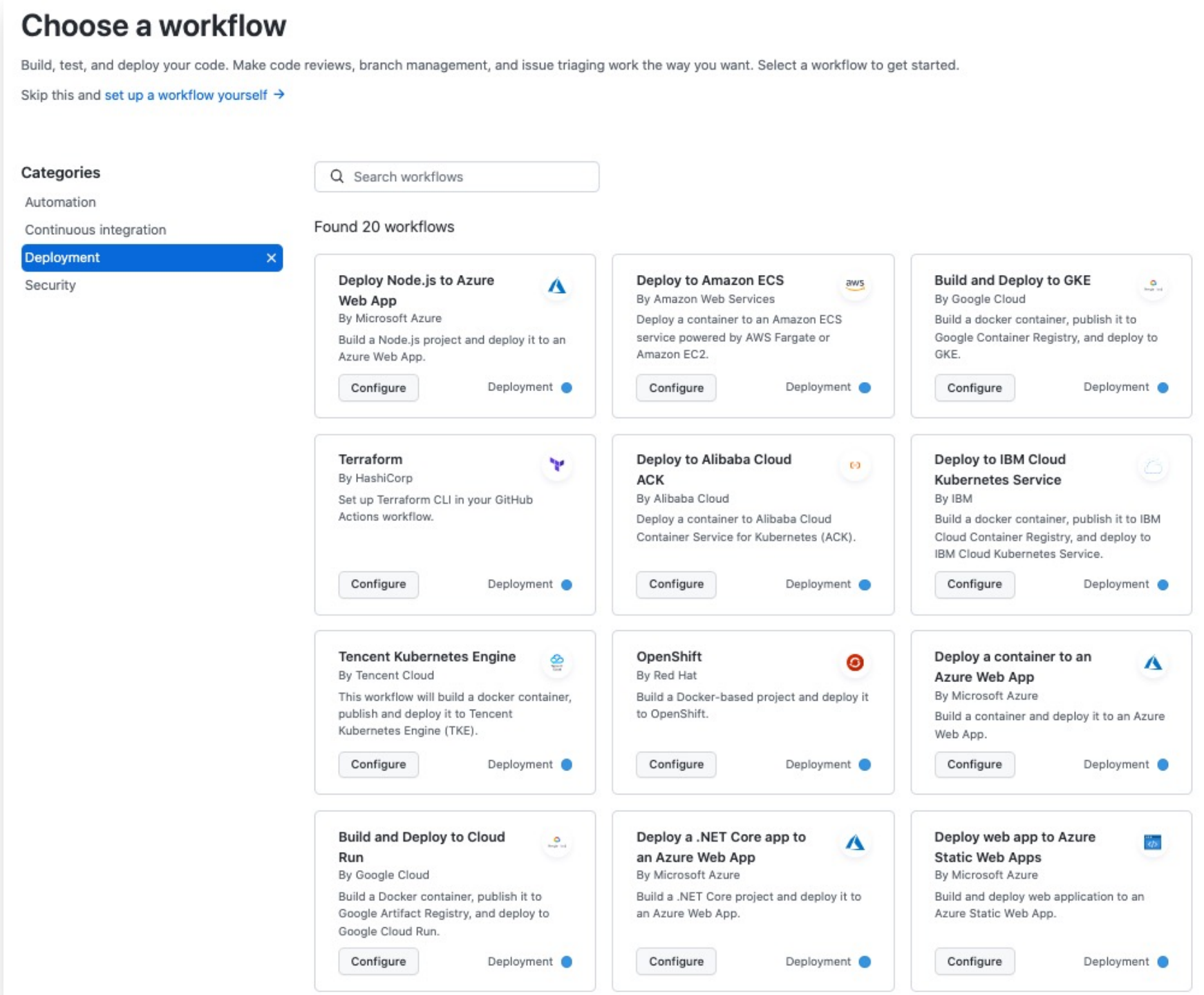
Continuous integration (CI)



GitHub Actions - CI

- Create CI using GitHub Actions workflow.
- A workflow is a configurable automated process made up of one or more jobs.
- Create a YAML file to define the workflow configuration.

GitHub CD


Choose a workflow
Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow to get started.
Skip this and set up a workflow yourself →
Categories
Automation
Continuous integration
Deployment
Security
Search workflows
Found 20 workflows
Deploy Node.js to Azure Web App
By Microsoft Azure
Build a Node.js project and deploy it to an Azure Web App.
Configure
Deployment
Deploy to Amazon ECS
By Amazon Web Services
Deploy a container to an Amazon ECS service powered by AWS Fargate or Amazon EC2.
Configure
Deployment
Build and Deploy to GKE
By Google Cloud
Build a docker container, publish it to Google Container Registry, and deploy to GKE.
Configure
Deployment
Terraform
By HashiCorp
Set up Terraform CLI in your GitHub Actions workflow.
Configure
Deployment
Deploy to Alibaba Cloud ACK
By Alibaba Cloud
Deploy a container to Alibaba Cloud Container Service for Kubernetes (ACK).
Configure
Deployment
Deploy to IBM Cloud Kubernetes Service
By IBM
Build a docker container, publish it to IBM Cloud Container Registry, and deploy to IBM Cloud Kubernetes Service.
Configure
Deployment
Tencent Kubernetes Engine
By Tencent Cloud
This workflow will build a docker container, publish and deploy it to Tencent Kubernetes Engine (TKE).
Configure
Deployment
OpenShift
By Red Hat
Build a Docker-based project and deploy it to OpenShift.
Configure
Deployment
Deploy a container to an Azure Web App
By Microsoft Azure
Build a container and deploy it to an Azure Web App.
Configure
Deployment
Build and Deploy to Cloud Run
By Google Cloud
Build a Docker container, publish it to Google Artifact Registry, and deploy to Google Cloud Run.
Configure
Deployment
Deploy a .NET Core app to an Azure Web App
By Microsoft Azure
Build a .NET Core project and deploy it to an Azure Web App.
Configure
Deployment
Deploy web app to Azure Static Web Apps
By Microsoft Azure
Build and deploy web application to an Azure Static Web App.
Configure
Deployment

{ GitHub CD Demo }

Thank You