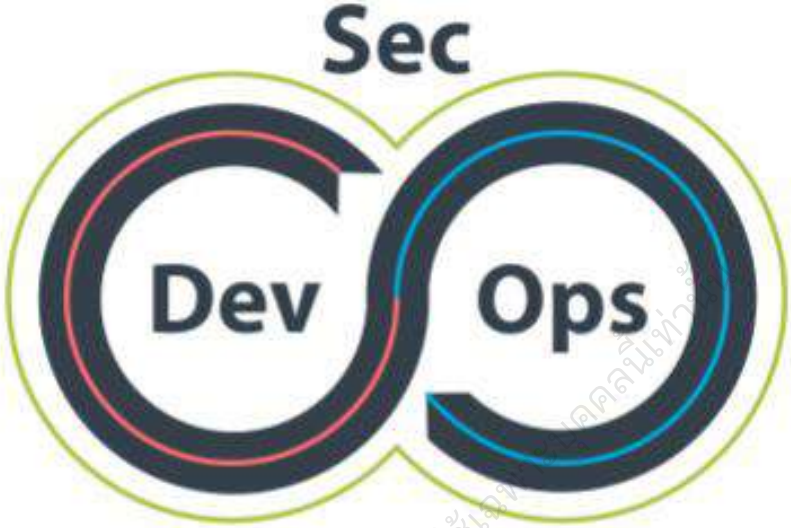


Opsta

Skooldio

Transformation and Technologies

Jirayut Nimsaeng (Dear)

CEO & Founder, Opsta (Thailand) Co.,Ltd.

Skooldio Online Course

#whoami

Jirayut Nimsaeng (Dear)

Jirayut has been involved in DevSecOps, Container, Cloud Technology and Open Source for over 10 years. He has experienced and succeeded in transforming several companies to deliver greater values and be more agile.

- **He is Founder and CEO of Opsta (Thailand) Co.,Ltd.**
- **He is Cloud/DevSecOps Transformation Consultant and Solution Architecture**
- **He is the first Certified Kubernetes Administrator (CKA) in Thailand and 146th of the world.**
- **He is first Thai Google Cloud Developer Expert (GDE) in Thailand**
- **Google Cloud Certified - Professional Cloud Architect and Associate Cloud Engineer**

Skooldio Opsta

Agenda (1)

- Introduction
  - DevSecOps Trend
  - What is DevSecOps?
  - DevSecOps and Agile
- DevSecOps Flow & Components
- Version Control System (VCS)
  - Git
- CI/CD & Artifacts
  - Continuous Integration
  - Continuous Delivery
- Modern Infrastructure
  - Cloud
  - Docker
  - Kubernetes
- Infrastructure Automation
  - Ansible
  - Terraform

Agenda (2)

- Monitoring
- Performance Load Testing
- Communication
- Automation Security
  - Pre-commit Stage
  - Acceptance Stage
  - Deployment Stage
- How to start DevSecOps
- Roadmap & Assessment
- DevSecOps Design
- People
- Demo
- Wrap up

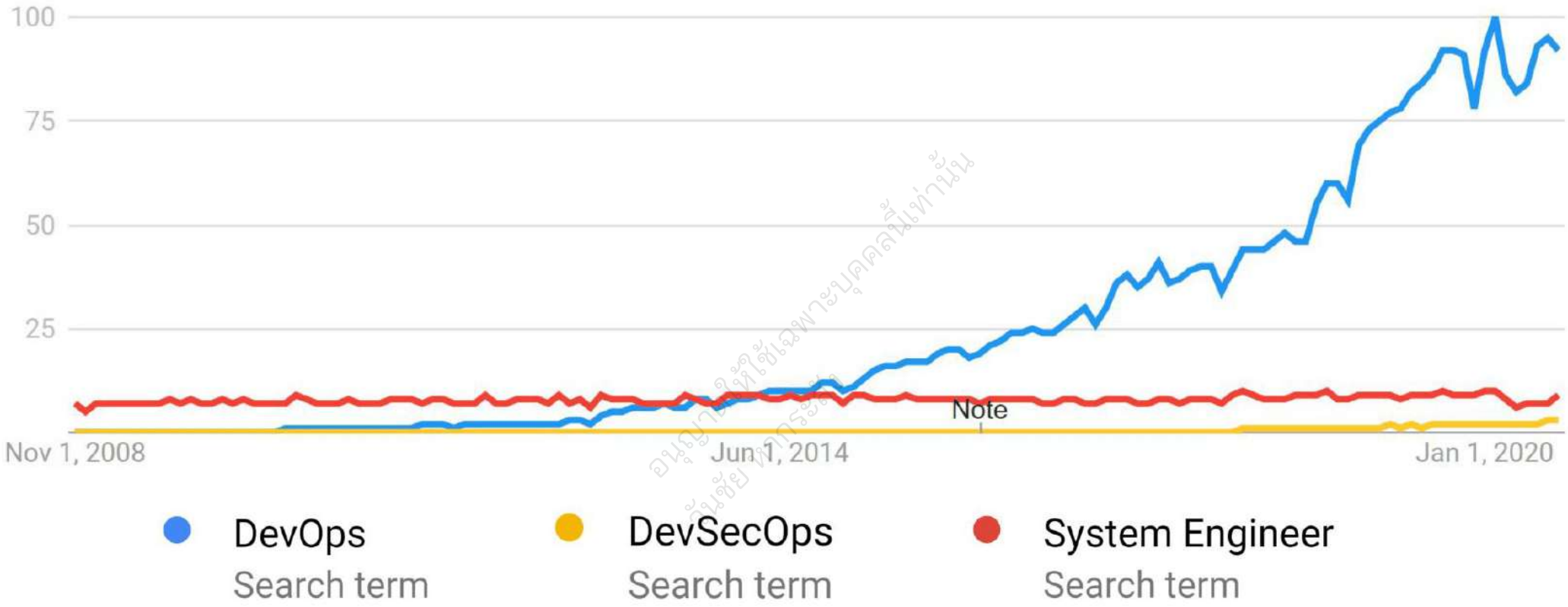
What is DevSecOps?

Technologies

Skooldio

Opsta

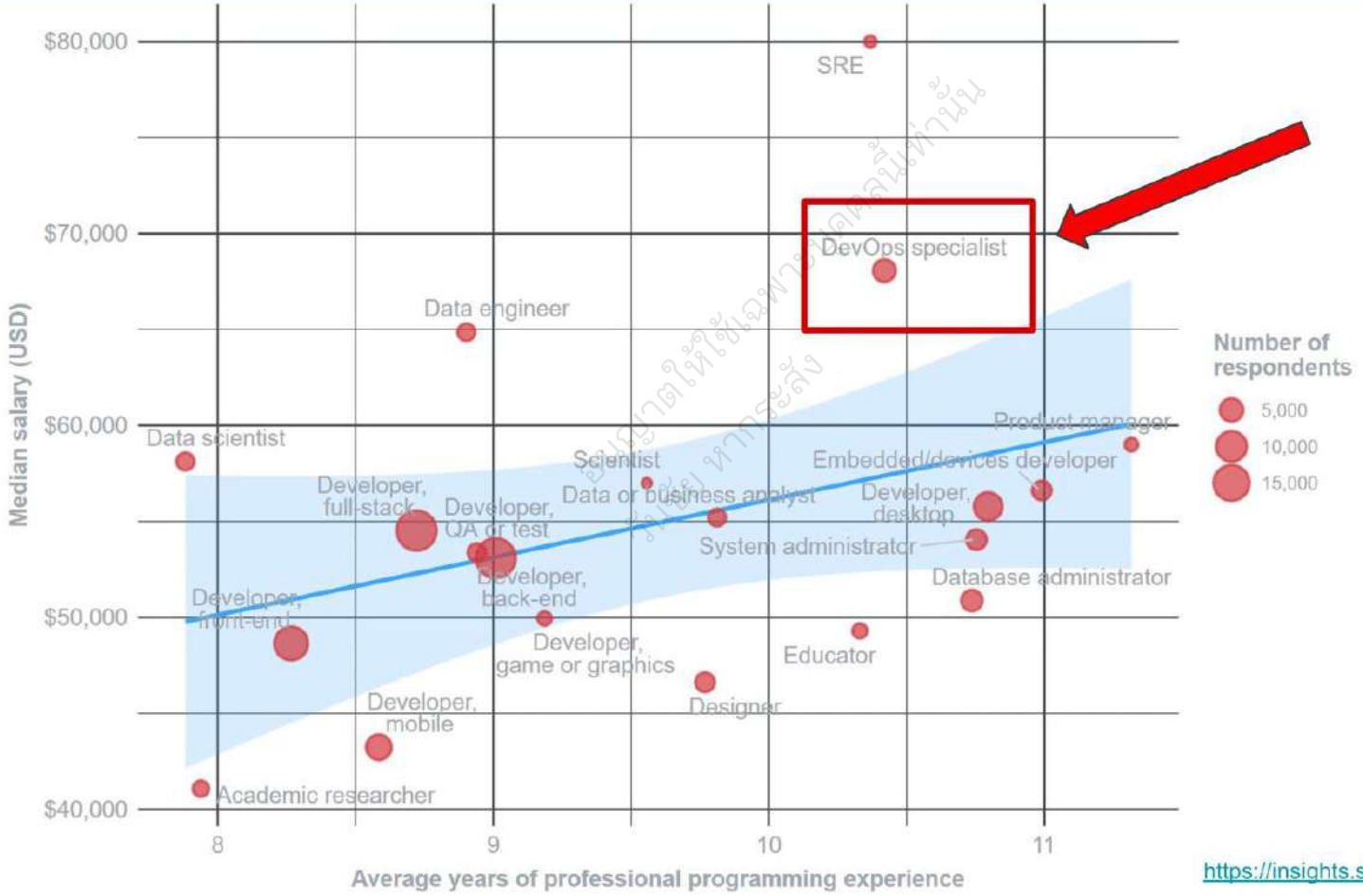
DevOps & DevSecOps Trend



https://trends.google.co.th/trends/explore?date=2008-10-13%202020-08-31&q=DevOps,System%20Engineer,DevSecOps

Skooldio Opsta

Global DevOps Salary 2020

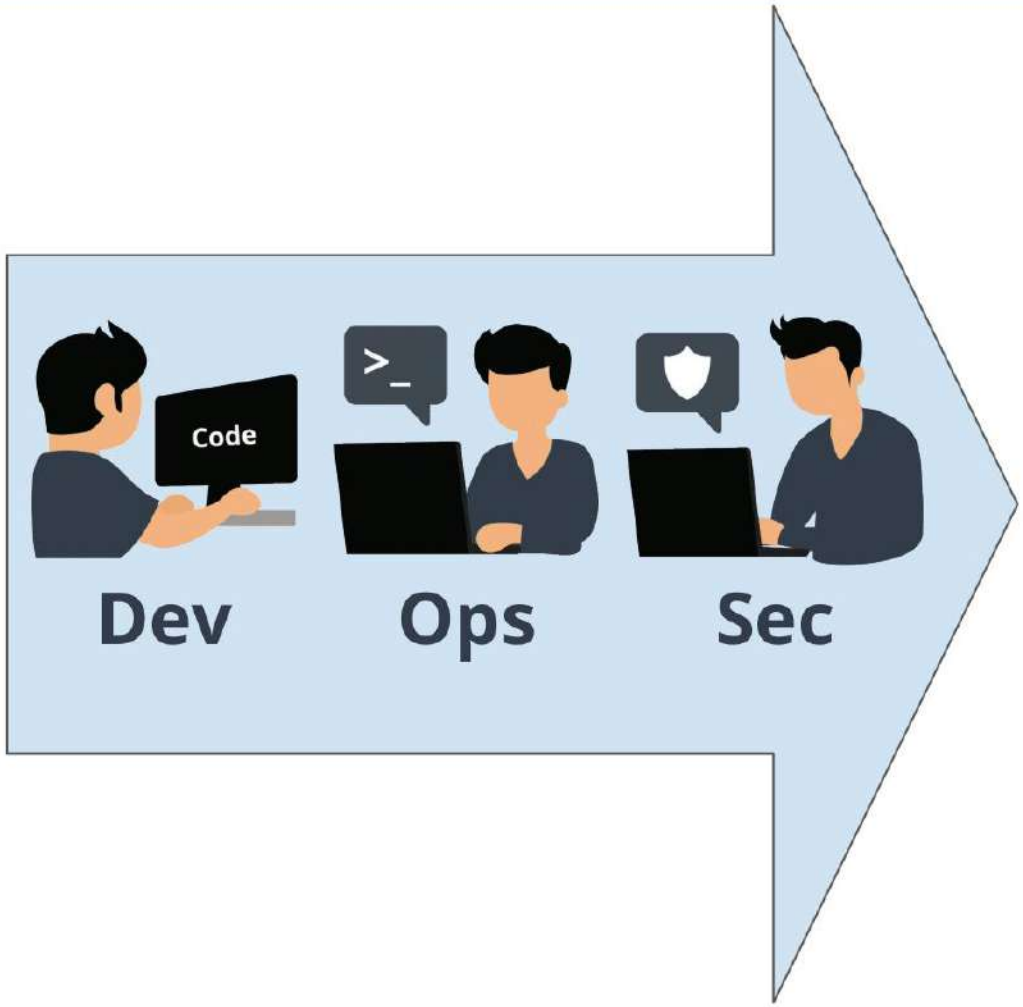


https://insights.stackoverflow.com/survey/2020

Skooldio Opsta

Business in IT Industy

Business in IT Industy

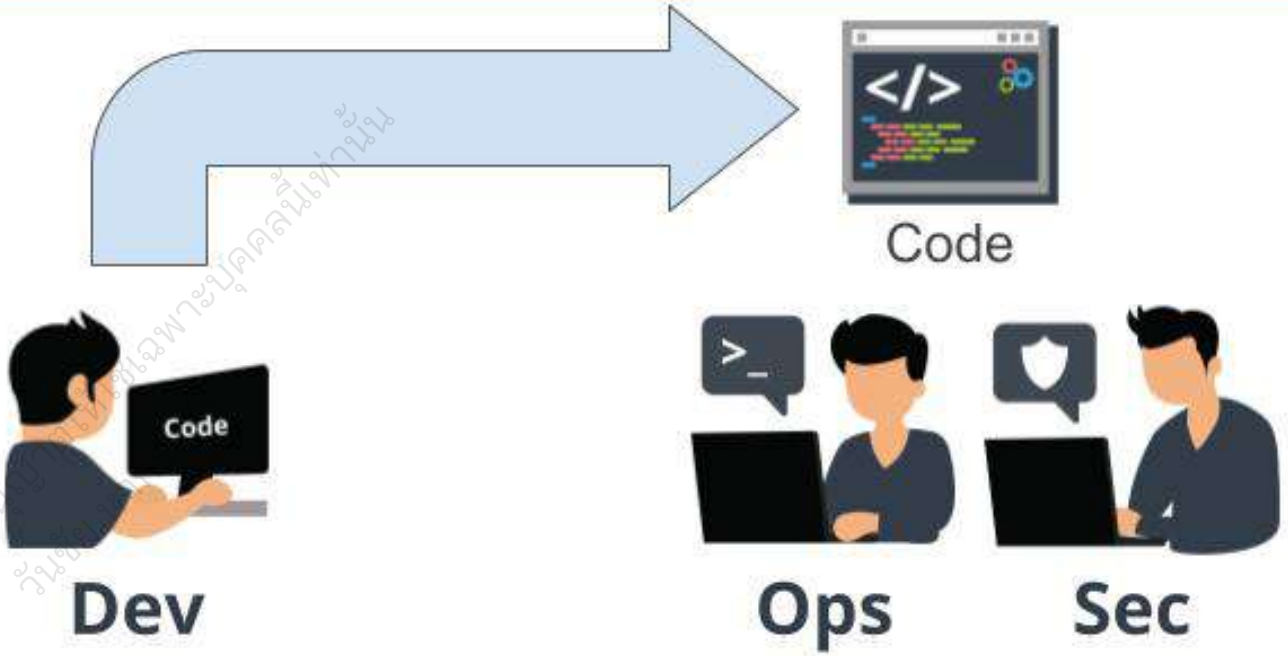


Ideal Development Cycle

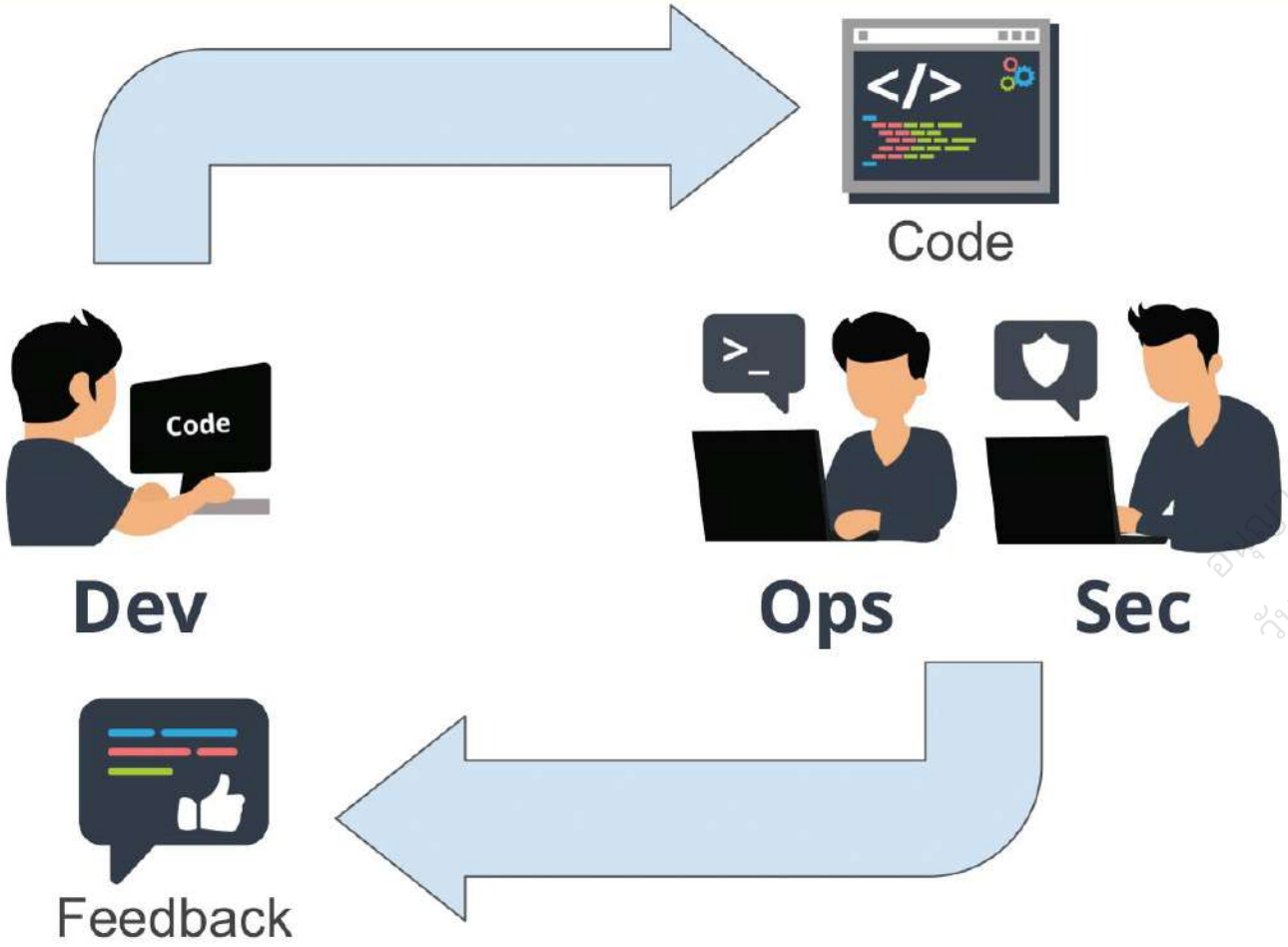
Dev

Ops Sec

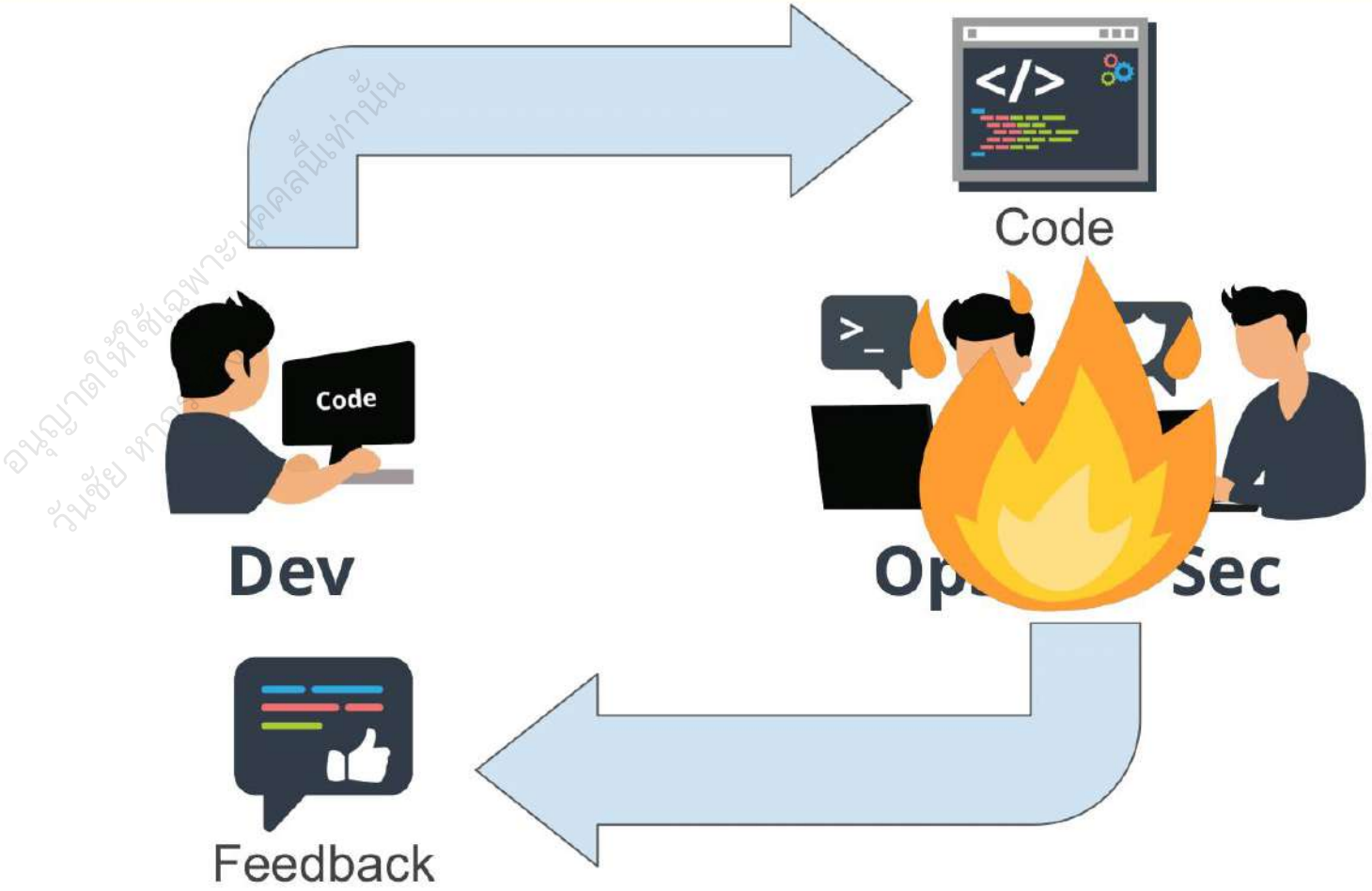
Ideal Development Cycle



Ideal Development Cycle

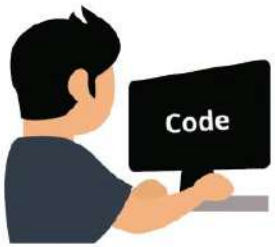


Reality



Silo

The Wall



Dev

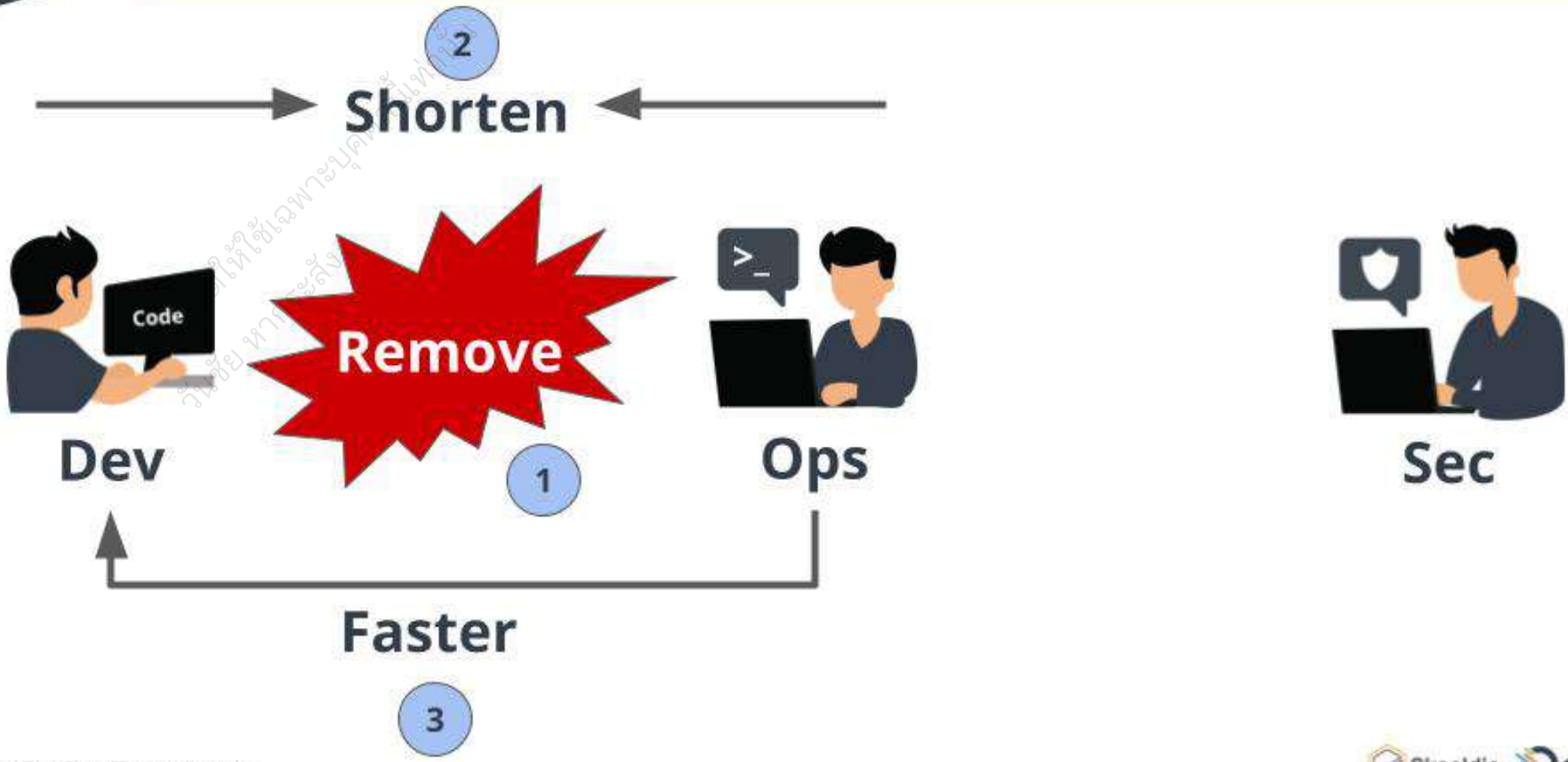
Ops

Sec

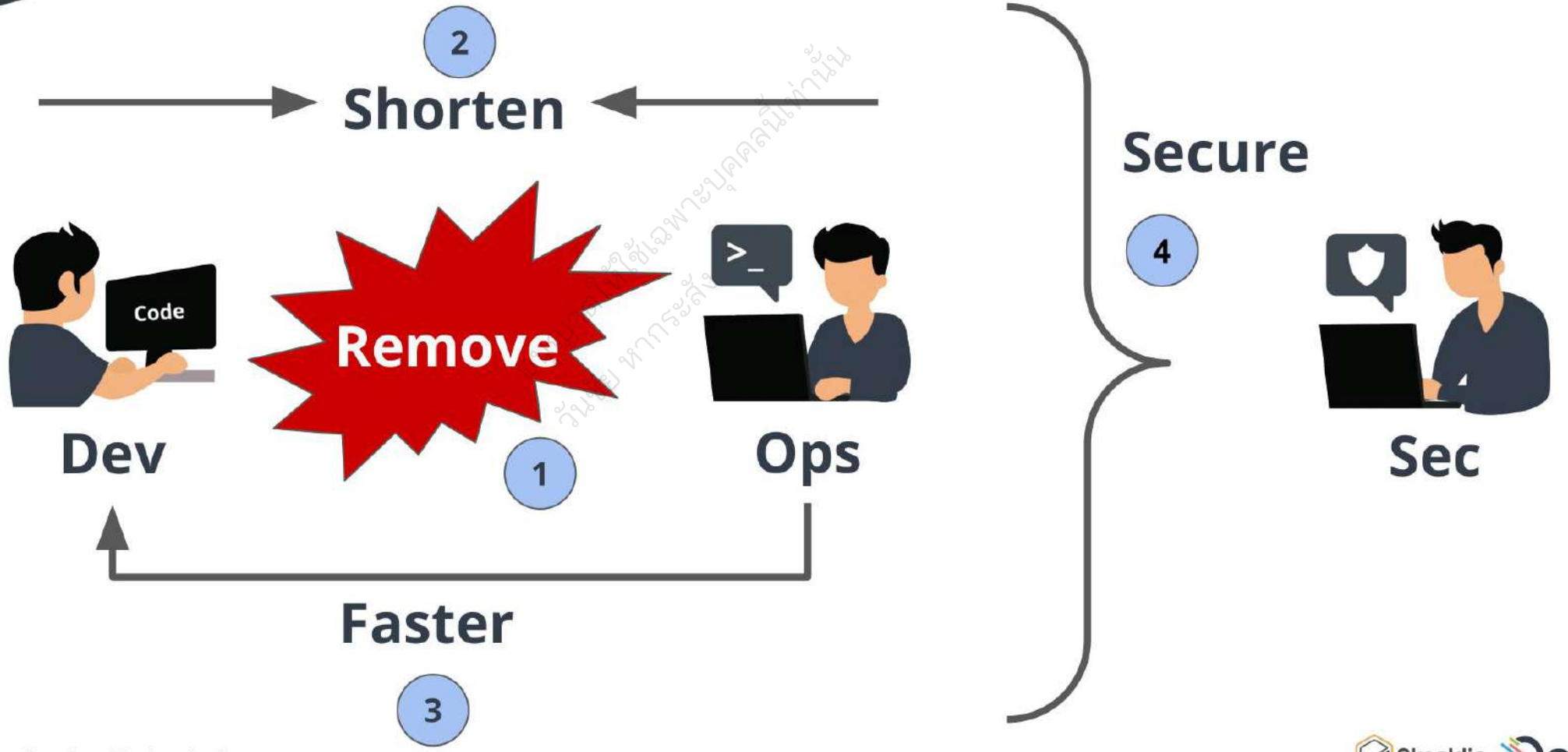
DevSecOps Culture

DevSecOps Culture

DevSecOps Culture



DevSecOps Culture



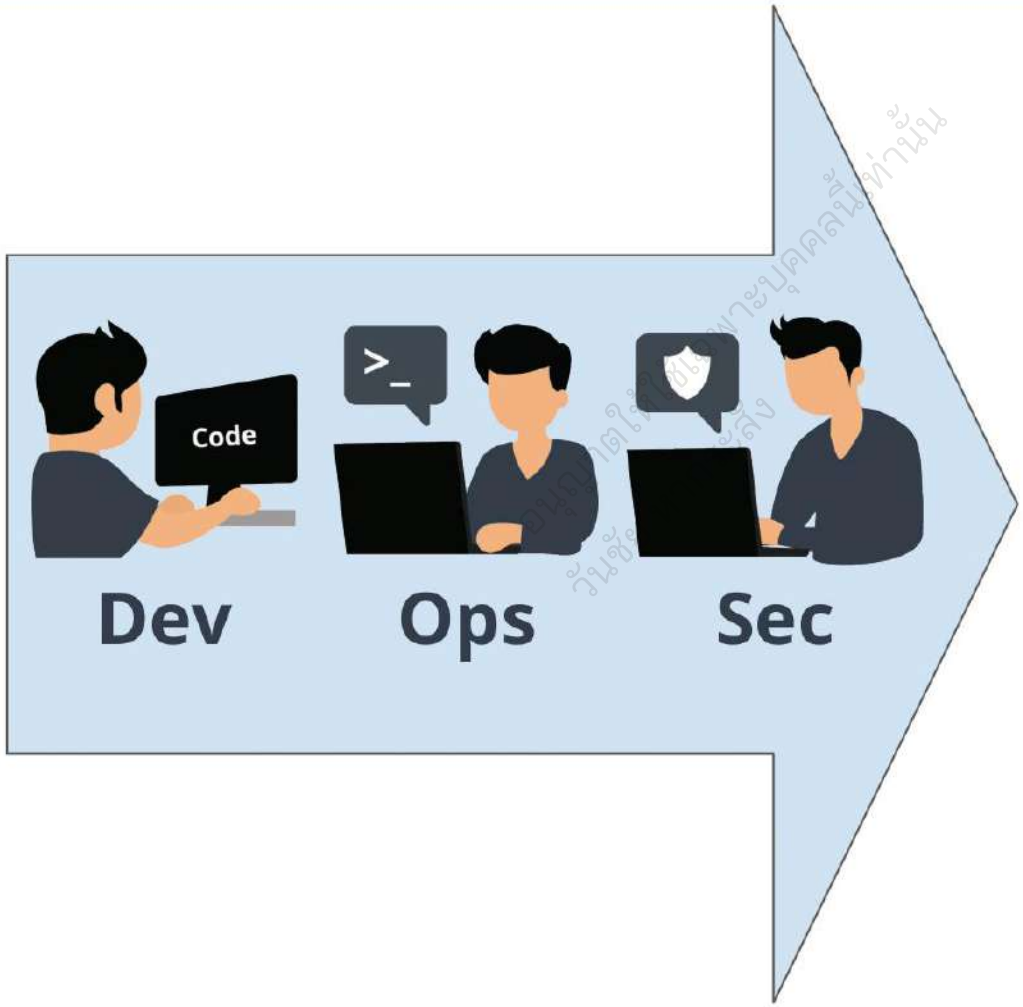
DevSecOps and Agile

Technologies

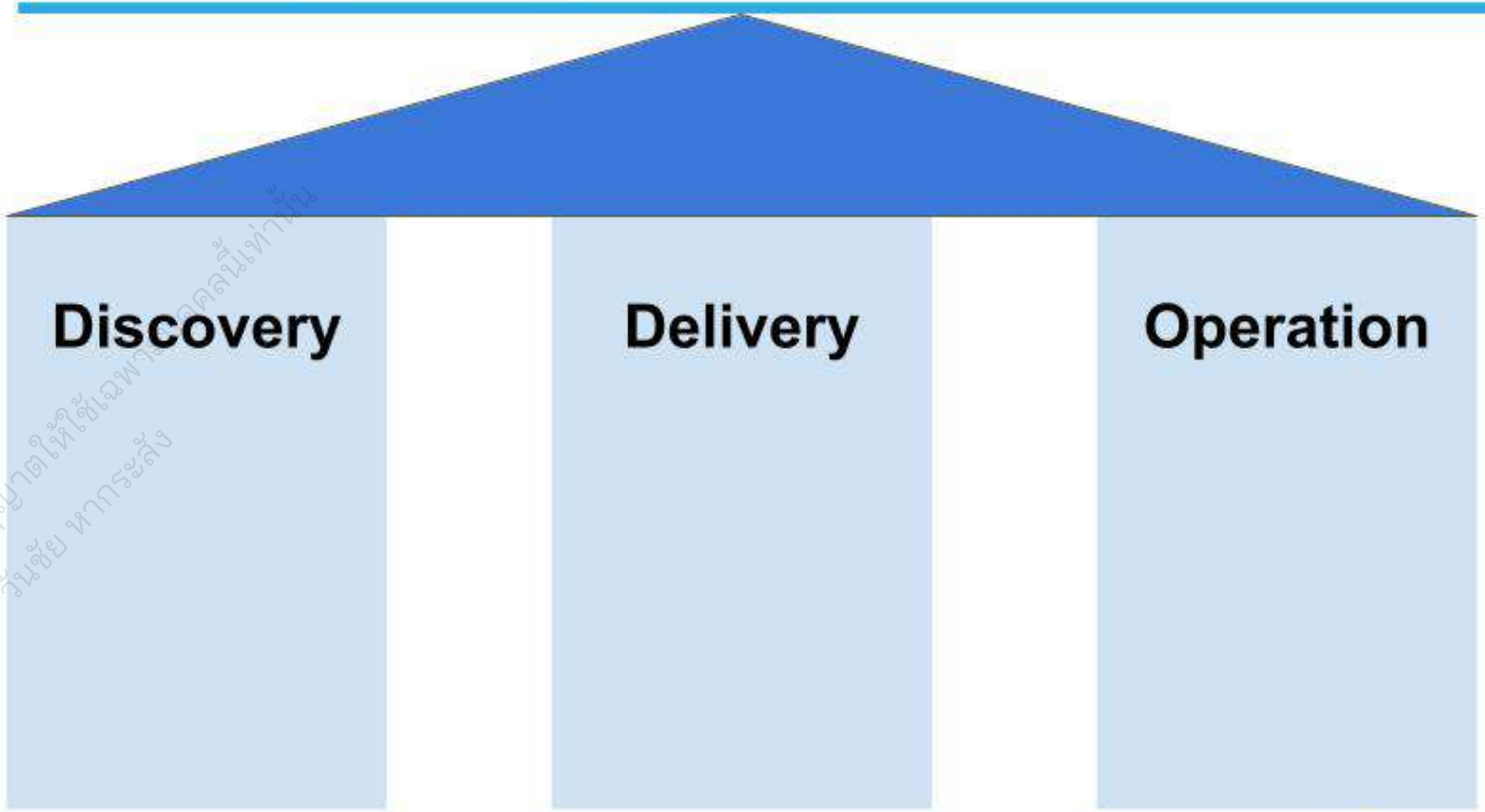
Skooldio

Opsta

Back to business again



Making an IT product



Making an IT product

Discovery

Design Thinking

Delivery

Scrum
Kanban

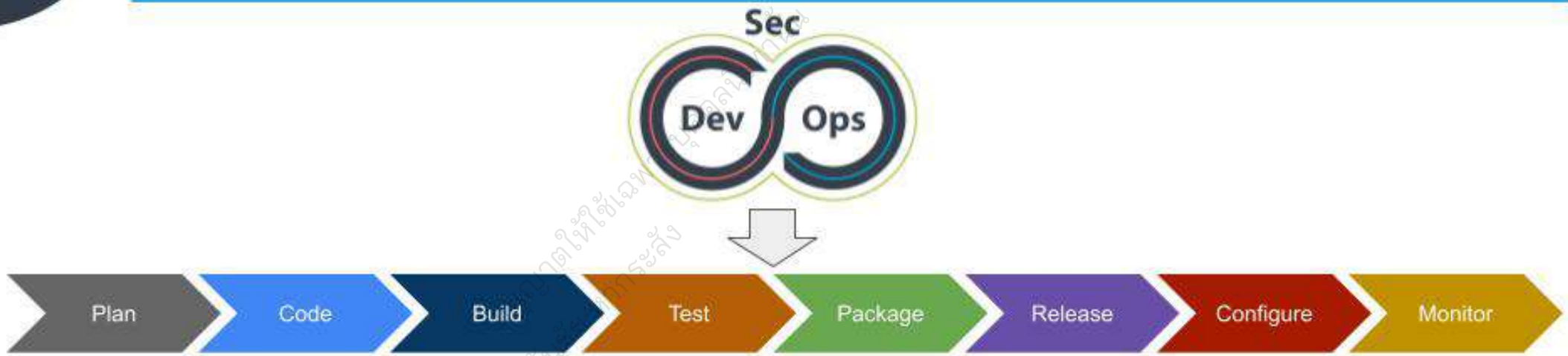
Operation

DevOps
DevSecOps

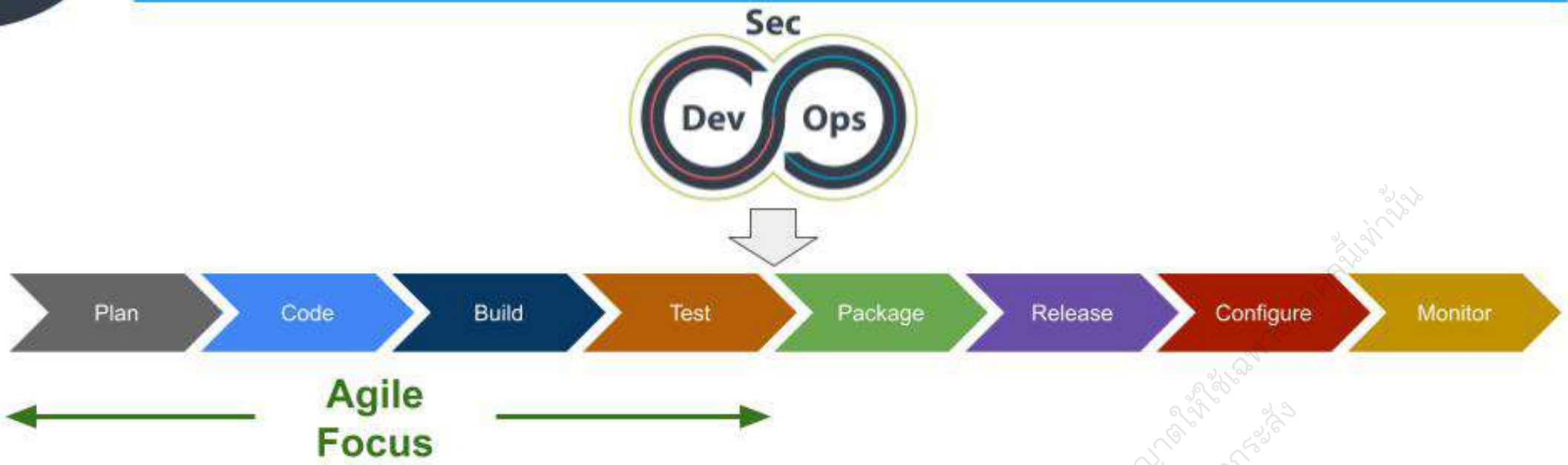
DevSecOps and Agile

Skooldio Opsta

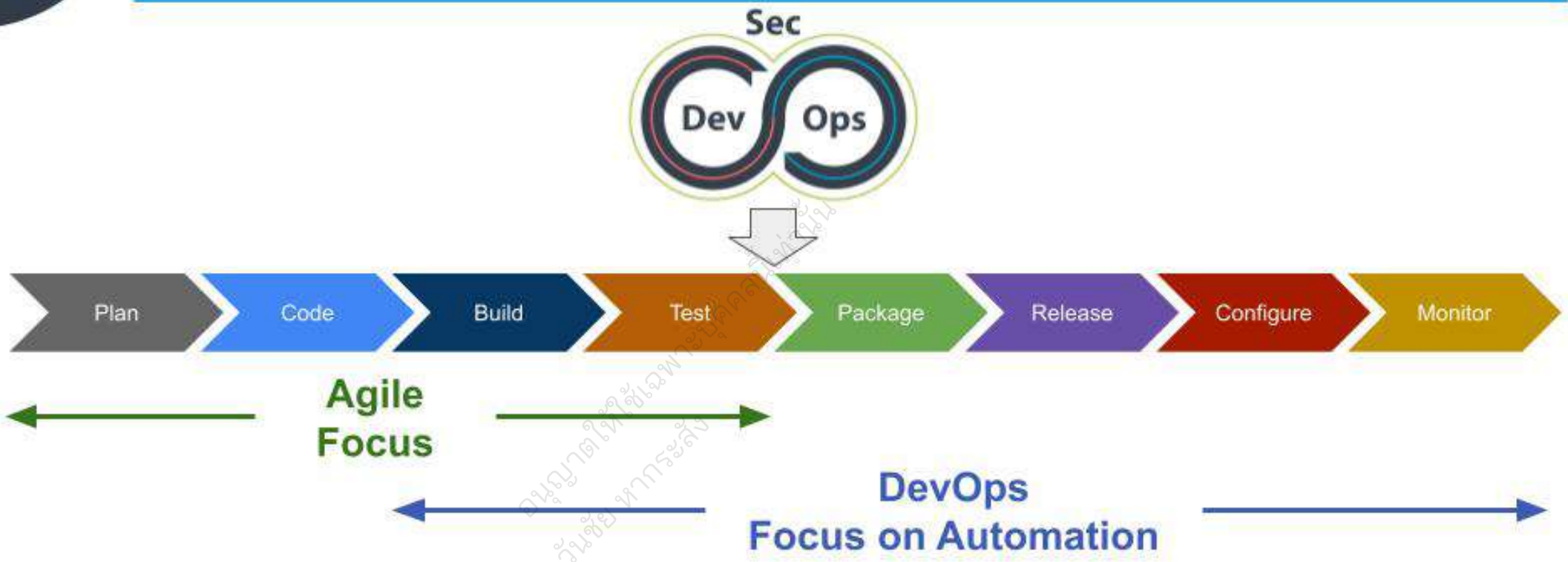
DevSecOps and Agile



DevSecOps and Agile

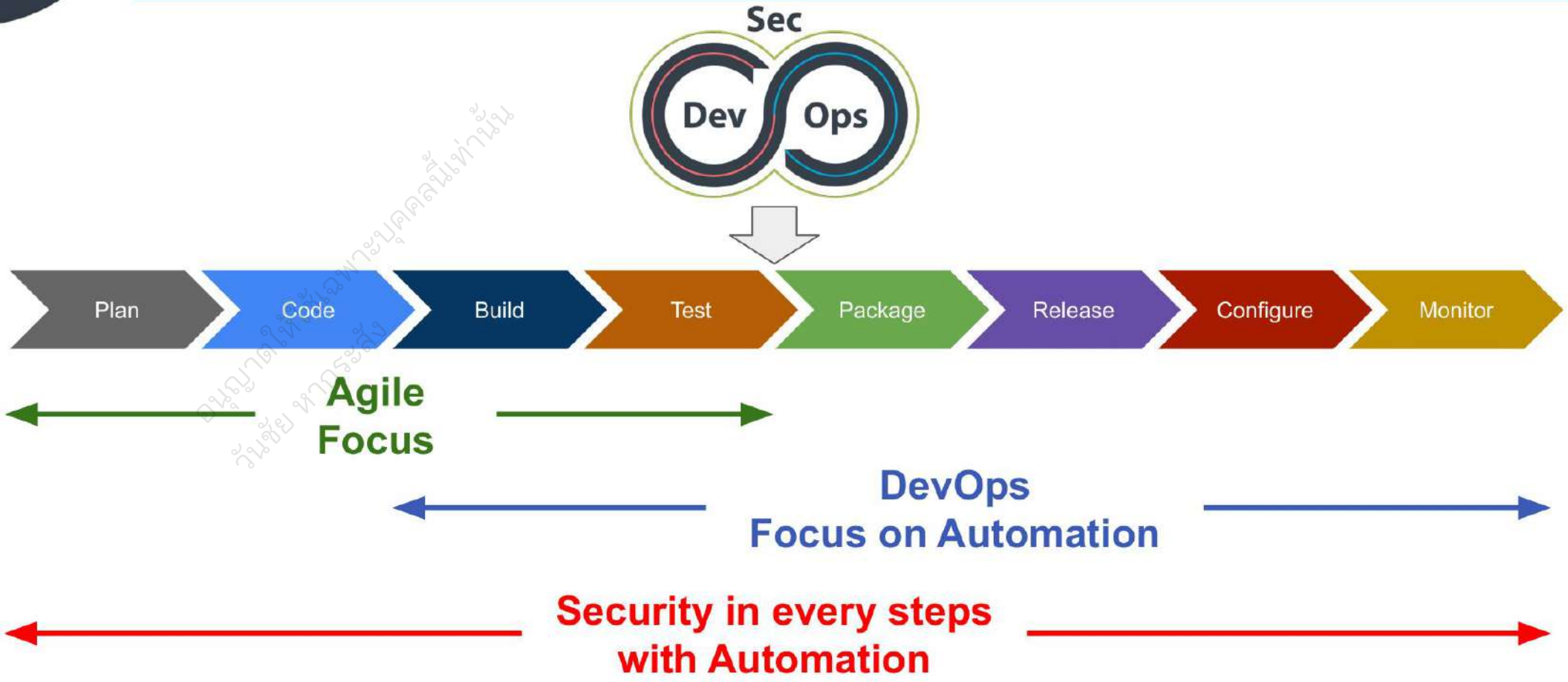


DevSecOps and Agile



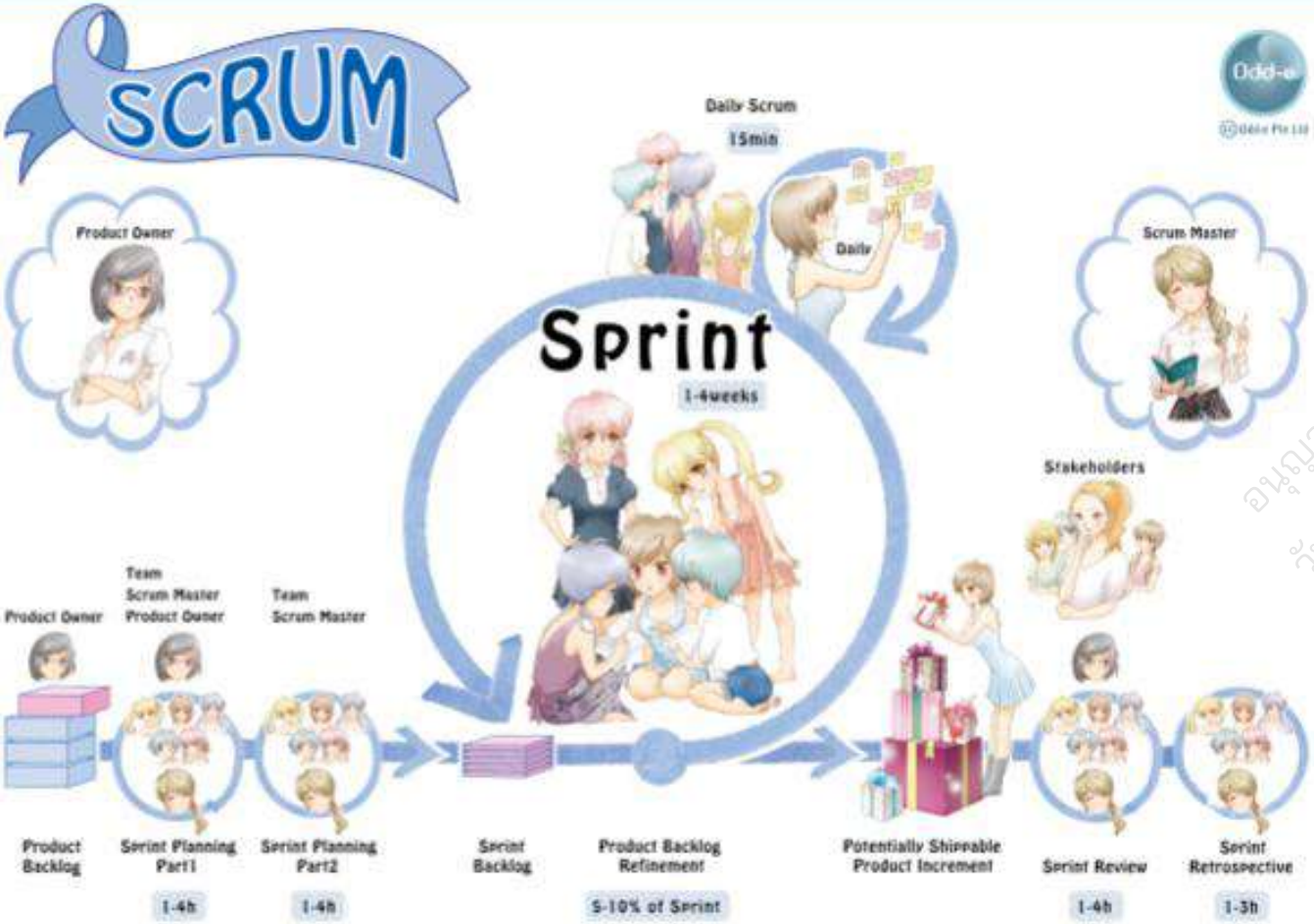
Skooldio Opsta

DevSecOps and Agile



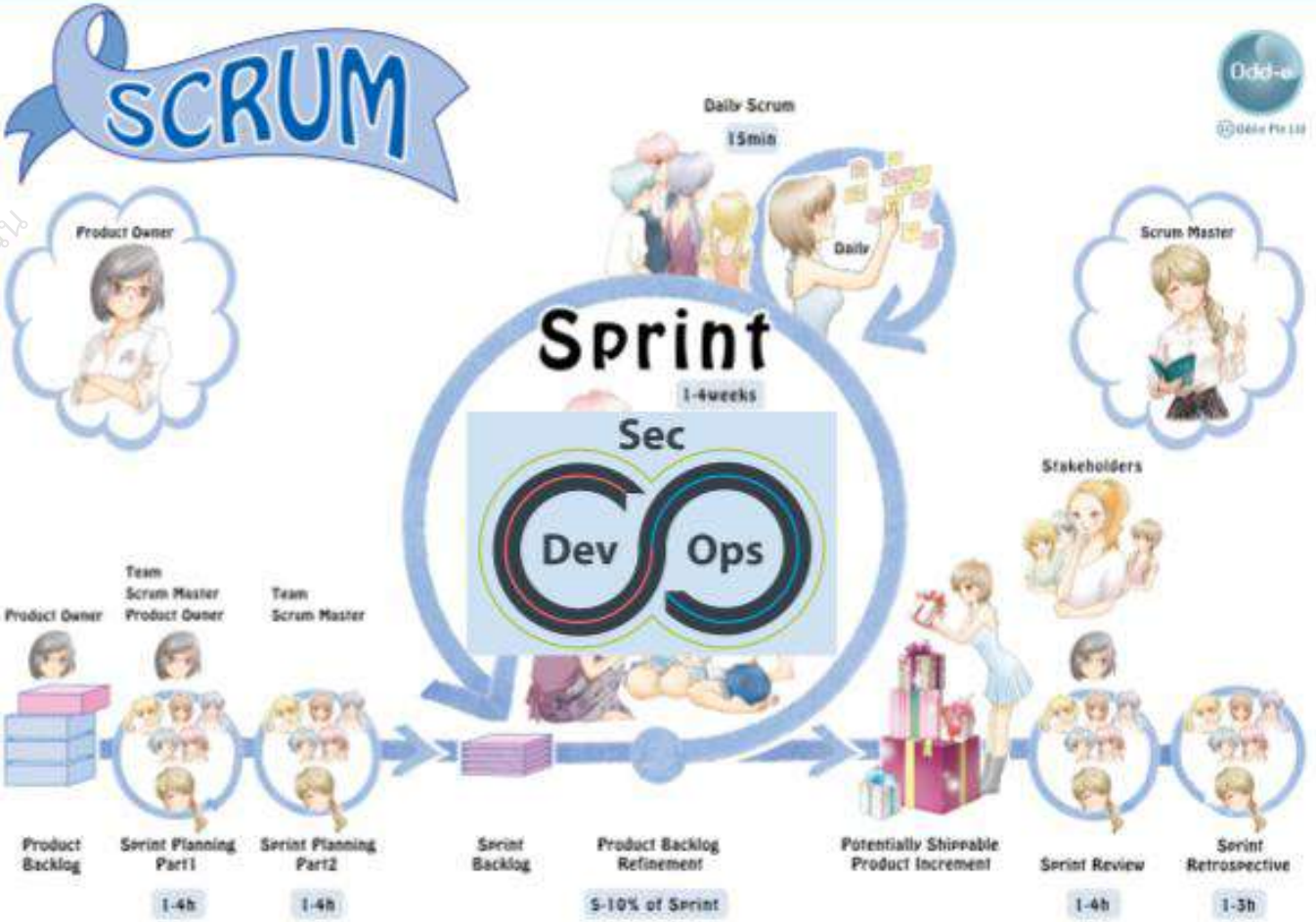
Skooldio Opsta

DevSecOps and Scrum



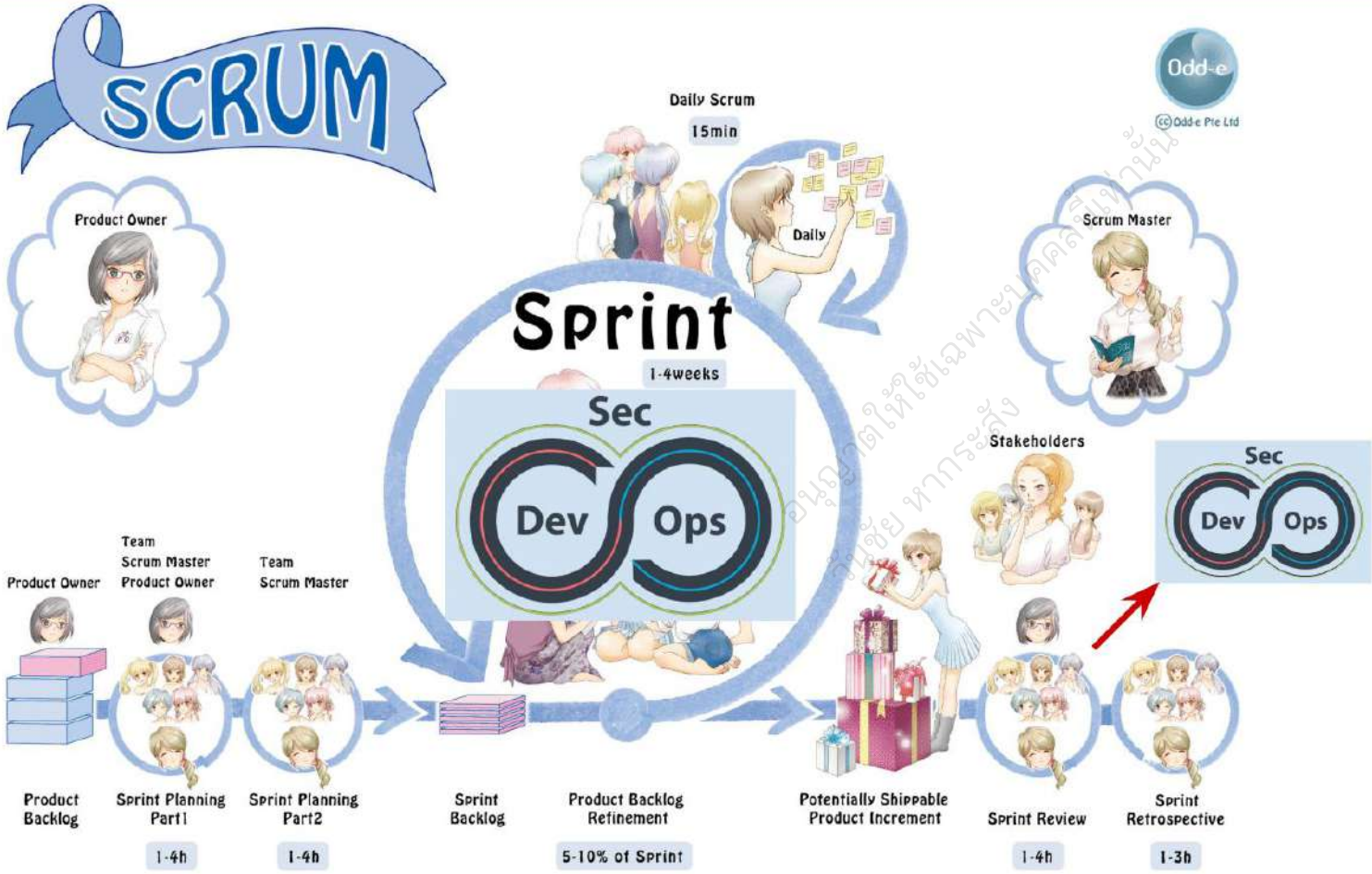
Skooldio Opsta

DevSecOps and Scrum



Skooldio Opsta

DevSecOps and Scrum



Skooldio Opsta

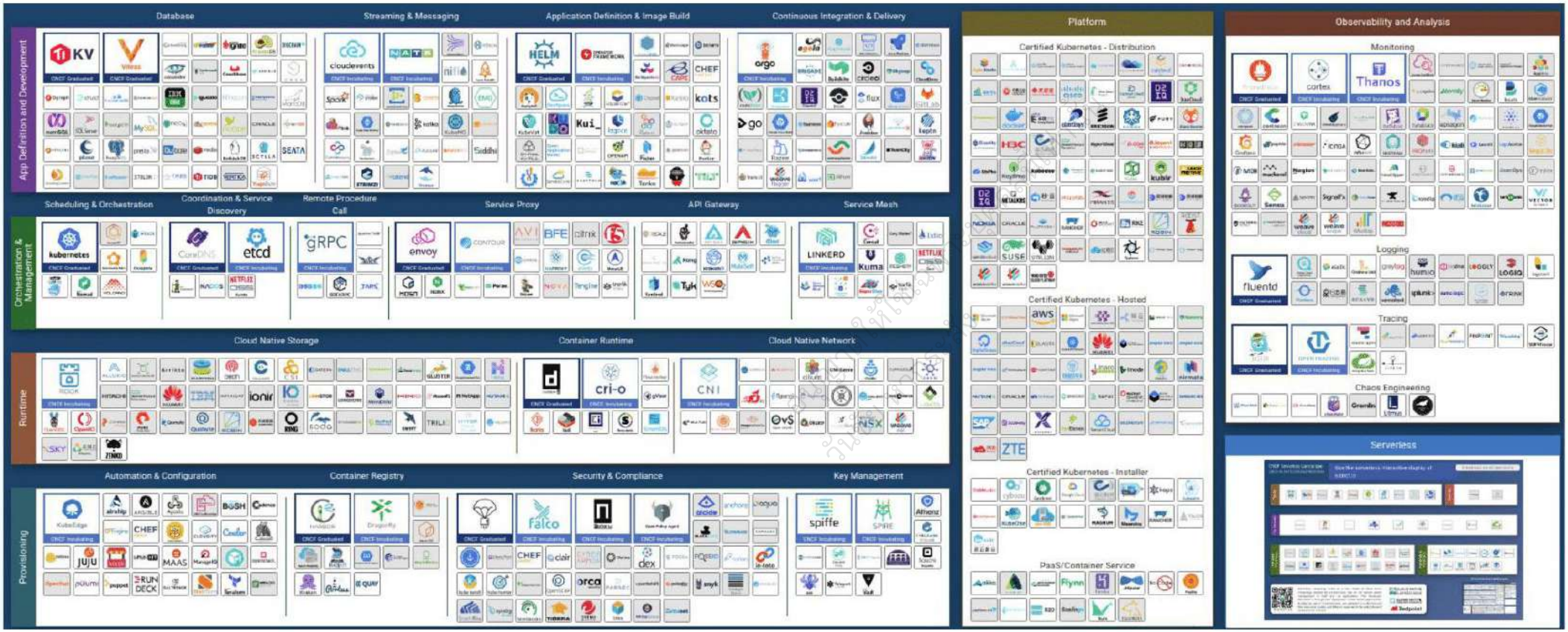
DevSecOps Flow & Components

Technologies

Skooldio

Opsta

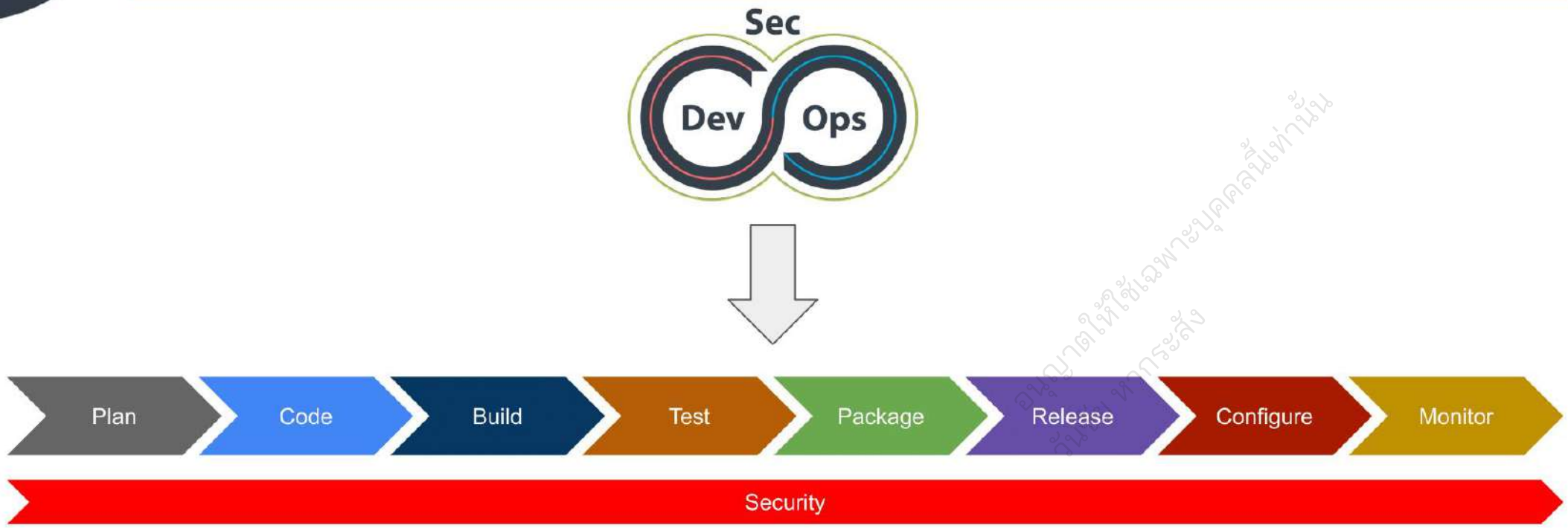
CNCF Landscape / DevSecOps Technologies



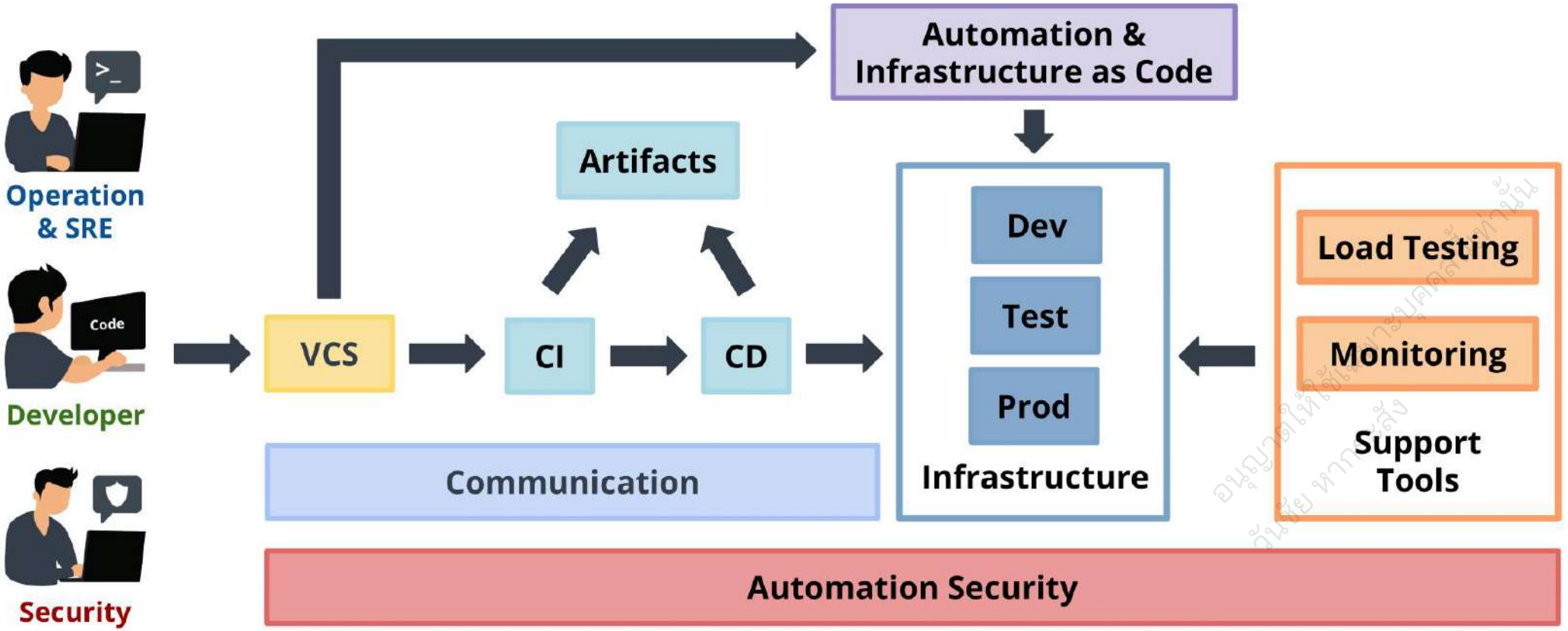
https://landscape.cncf.io

Skooldio Opsta

DevSecOps Flow



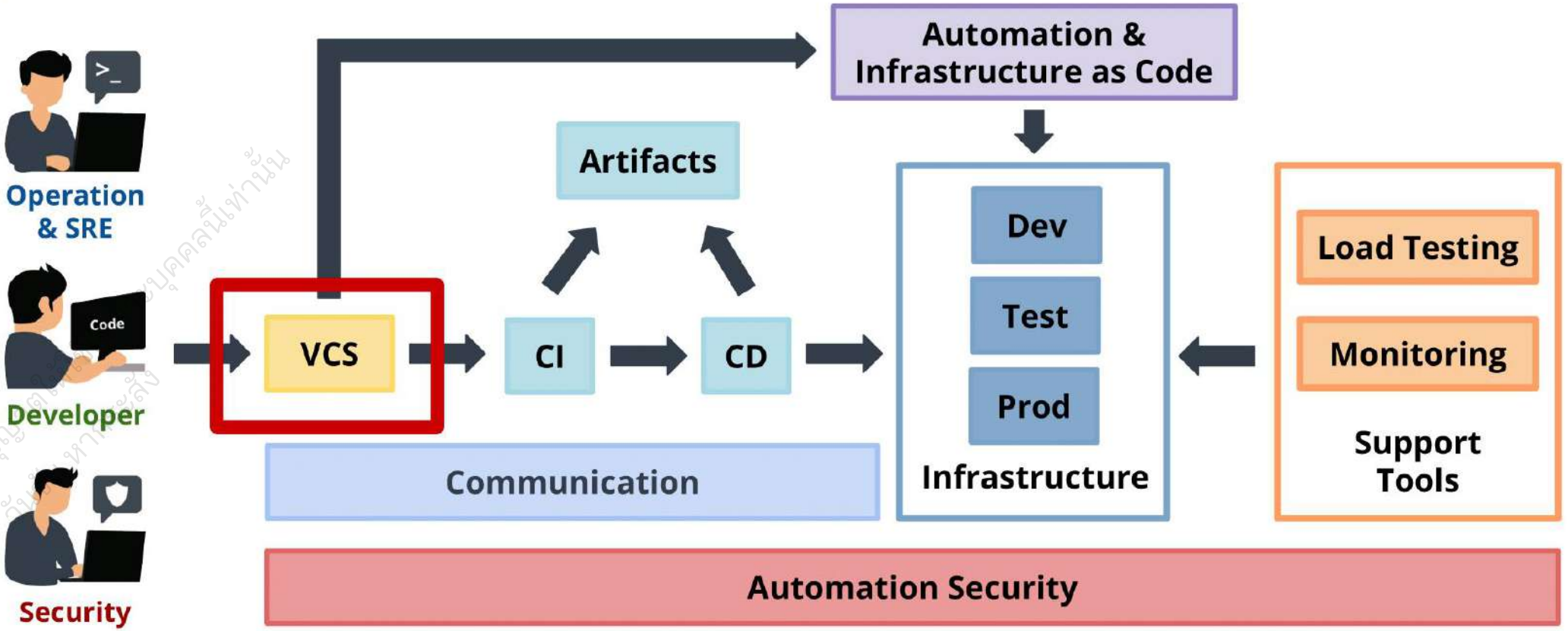
Generic DevSecOps Flow & Components



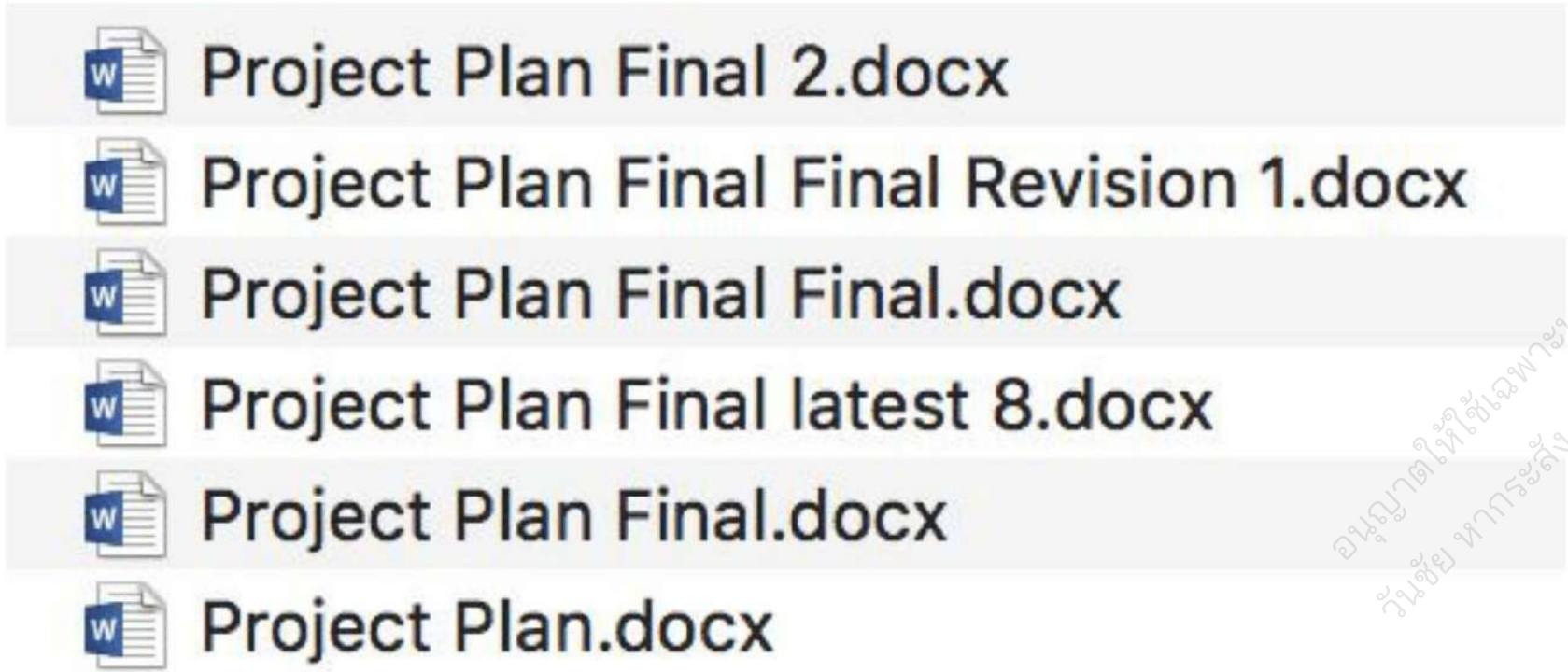
Skooldio Opsta

Version Control System (VCS)

Version Control System (VCS)



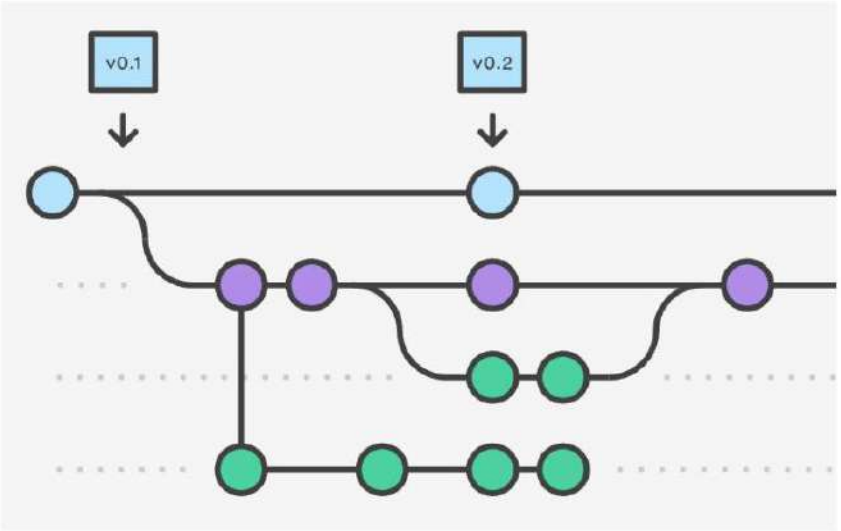
Classic Problem



Version Control System (VCS)

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

- Good for text file
- Not recommend for binary file



Version Control Software

git

Visual Studio
Team Foundation Server

mercurial

Git Version Control

Technologies

Skooldio

Opsta

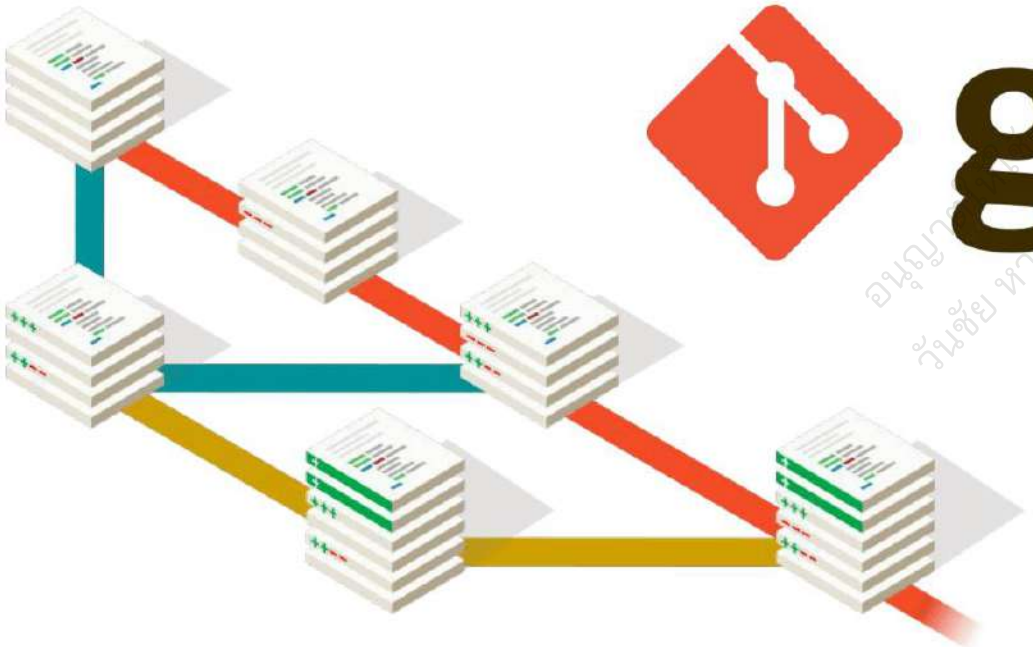
Popular Git Software

Server

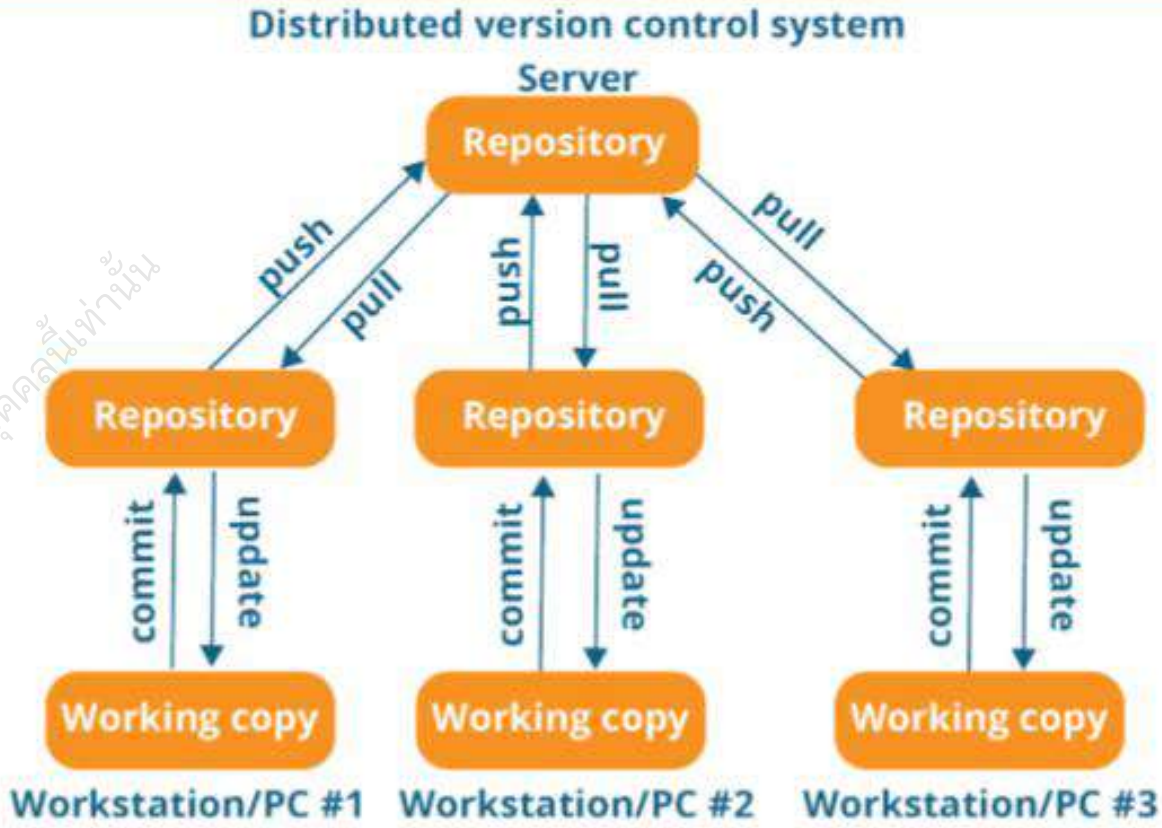
Client

Git Version Control

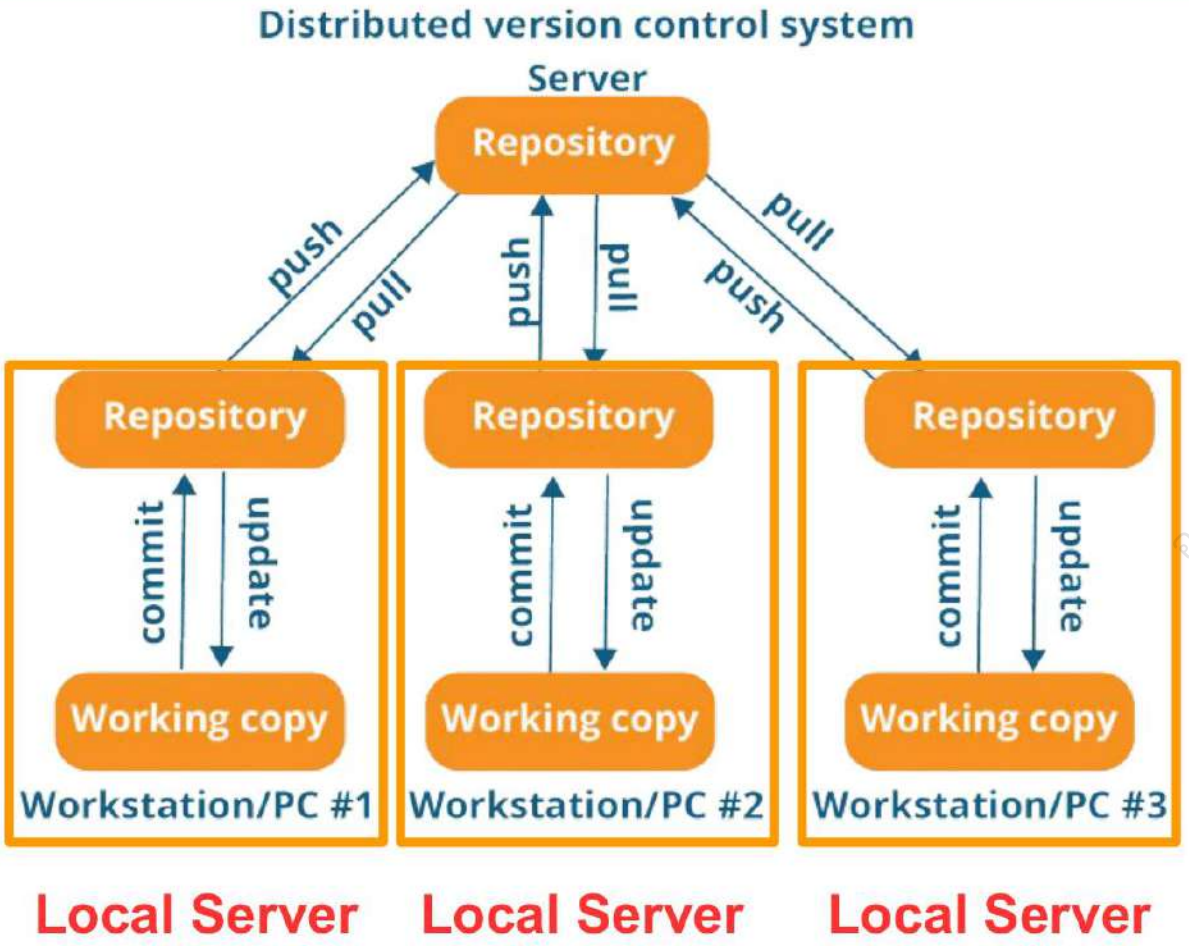
Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.



Git Architecture

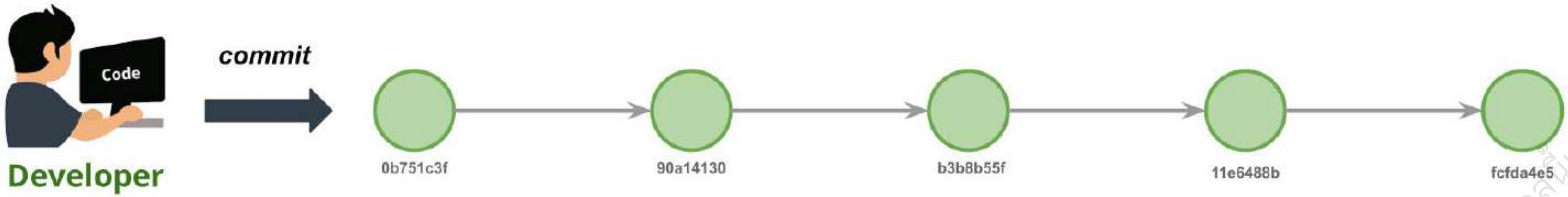


Git Architecture



Git Features: History

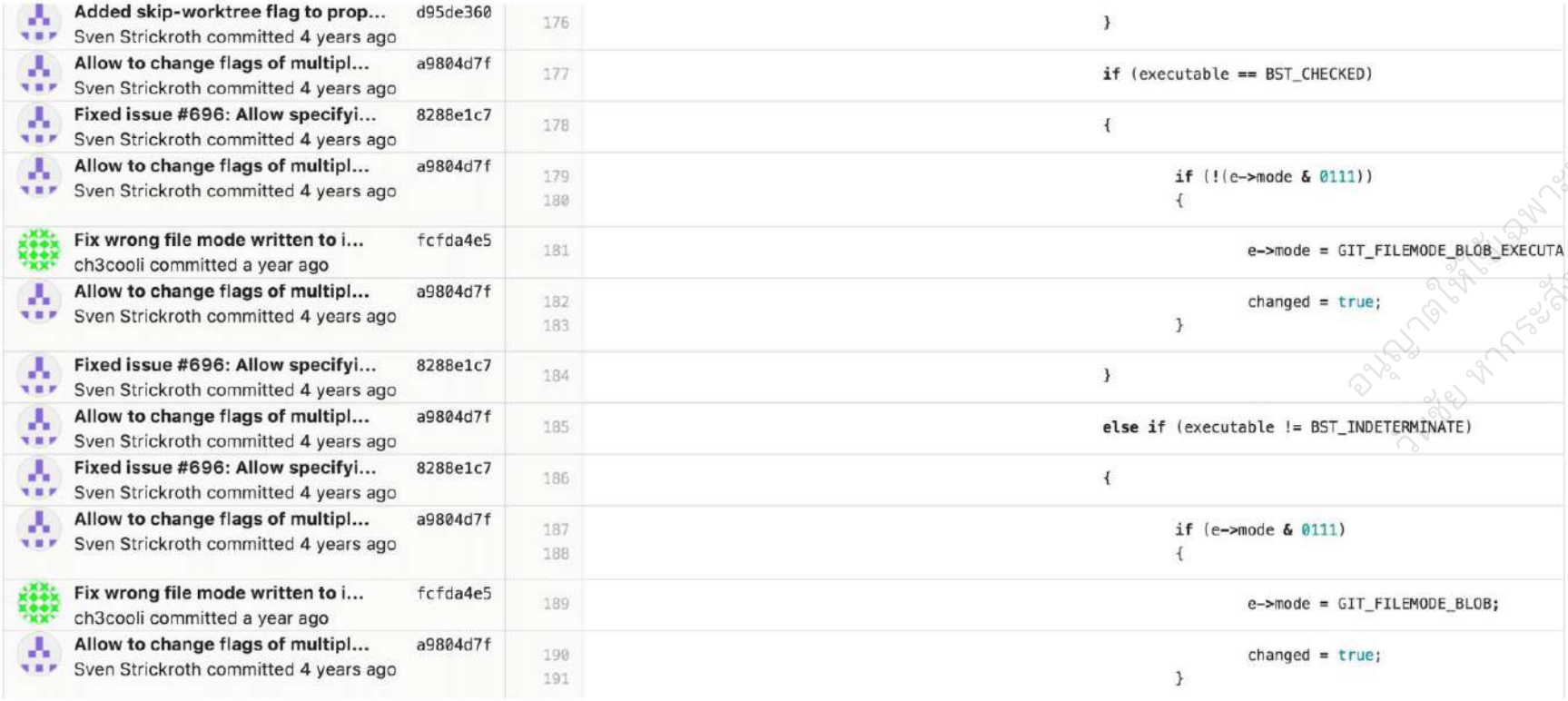
See what exactly be change and when.



16 Jan, 2016 1 commit		
Fix wrong file mode written to index when mark/unmark as symlink ... ch3cooli committed a year ago	fcfda4e5	Browse Files
12 Jan, 2016 1 commit		
Refactoring: Reduce indention ... Sven Strickroth committed a year ago	11e6488b	Browse Files
10 Jan, 2016 1 commit		
Fix possible division by zero ... Sven Strickroth committed a year ago	b3b8b55f	Browse Files
08 Jan, 2016 4 commits		
Refactor: Use smart pointers ... Sven Strickroth committed a year ago	90a14130	Browse Files
Initialize SYS_IMAGE_LIST after initializing OLE ... Sven Strickroth committed a year ago	0b751c3f	Browse Files
Fix typo ... Sven Strickroth committed a year ago	3fd7fb7a	Browse Files
Remove compilation warning ... Jiří Engelthaler committed a year ago	1e4021bd	Browse Files

Git Features: History

Really exact. Line by line history.



Added skip-worktree flag to prop... Sven Strickroth committed 4 years ago	d95de360	176	}
Allow to change flags of multipl... Sven Strickroth committed 4 years ago	a9804d7f	177	if (executable == BST_CHECKED)
Fixed issue #696: Allow specifyi... Sven Strickroth committed 4 years ago	8288e1c7	178	{
Allow to change flags of multipl... Sven Strickroth committed 4 years ago	a9804d7f	179 180	if (!(e->mode & 0111)) {
Fix wrong file mode written to i... ch3cooli committed a year ago	fcfda4e5	181	e->mode = GIT_FILEMODE_BLOB_EXECUTA
Allow to change flags of multipl... Sven Strickroth committed 4 years ago	a9804d7f	182 183	changed = true; }
Fixed issue #696: Allow specifyi... Sven Strickroth committed 4 years ago	8288e1c7	184	}
Allow to change flags of multipl... Sven Strickroth committed 4 years ago	a9804d7f	185	else if (executable != BST_INDETERMINATE)
Fixed issue #696: Allow specifyi... Sven Strickroth committed 4 years ago	8288e1c7	186	{
Allow to change flags of multipl... Sven Strickroth committed 4 years ago	a9804d7f	187 188	if (e->mode & 0111) {
Fix wrong file mode written to i... ch3cooli committed a year ago	fcfda4e5	189	e->mode = GIT_FILEMODE_BLOB;
Allow to change flags of multipl... Sven Strickroth committed 4 years ago	a9804d7f	190 191	changed = true; }

Git Features: Code Review

Allow micro level collaboration between team member.

src/Git/GitAdminDir.h

```
43  43      static CString GetSuperProjectRoot(const CString& path);
44  44
45  45      static bool GetAdminDirPath(const CString &projectTopDir, CString& adminDir);
    46  +   static bool GetWorktreeAdminDirPath(const CString &projectTopDir, CString& adminDir);
```

Sven Strickroth @mrtux commented a month ago — Owner

I'm not sure whether this is a good name... Also, please put the `&` to the type.

Rick Burgstaler @rburgstaler commented a month ago

I am open to a different name. I was just rolling with what @ch3cooli started in this branch.

As a reference libgit2 likes to refer to the `GetAdminDirPath()` as `commondir` and refer to `GetWorktreeAdminDirPath()` as `gitdir` see here. One idea would be to adopt that convention. Maybe that would be something like renaming `GetAdminDirPath()` to `GetCommonDirPath()` and rename `GetWorktreeAdminDirPath()` to `GetGitDirPath()`

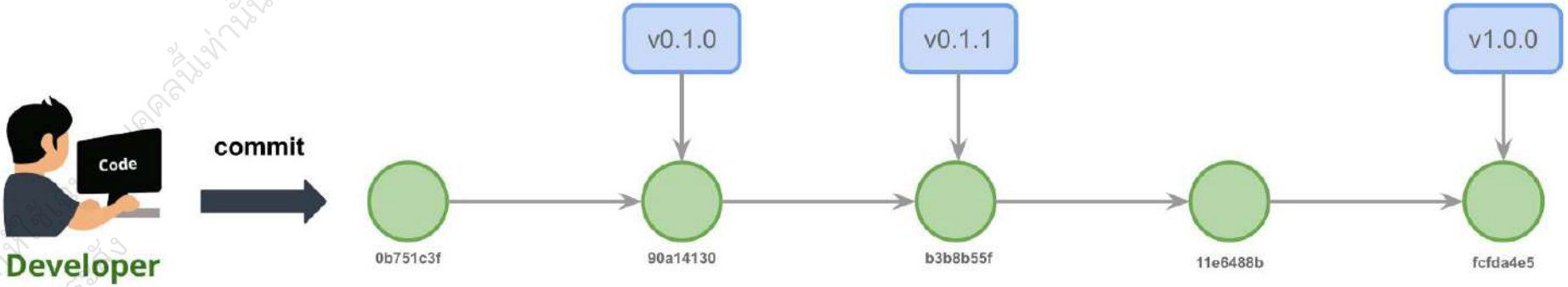
Edited a month ago

Sven Strickroth @mrtux commented 3 weeks ago — Owner

I rebased the worktree branch, can you please include that commit here?

Git Features: Tag

Tagging is generally used to capture a point in history that is used for a marked version release (i.e. v1.0.1)



REL_2.3.0.0_EXTERNAL Released TortoiseGit 2.3.0.0
d2068cc0 · Prepare release and bump version numbers to 2.3.0.0 · 4 months ago

REL_2.2.0.0_EXTERNAL Released TortoiseGit 2.2.0
7d0cf461 · Prepare release and bump version numbers to 2.2.0.0 · 7 months ago

REL_2.1.0.0_EXTERNAL Released TortoiseGit 2.1.0.0
010b3875 · Prepare release and bump version numbers to 2.1.0.0 · 10 months ago

REL_2.0.0.0_EXTERNAL Released TortoiseGit 2.0.0.0
920ed288 · Prepare release and bump version numbers to 2.0.0.0 · 11 months ago

REL_1.8.16.0_EXTERNAL Released TortoiseGit 1.8.16.0
9a490362 · Prepare release and bump version numbers to 1.8.16.0 · a year ago

Git Branching System

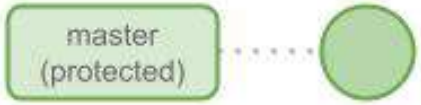
Technologies

Git Features: Branching

Git Branching means you diverge from the main line of development and continue to do work without messing with that main line

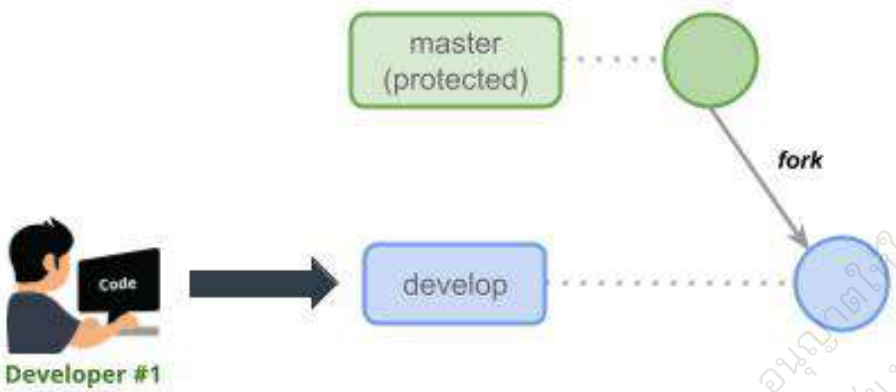
Git Features: Branching

Git Branching means you diverge from the main line of development and continue to do work without messing with that main line



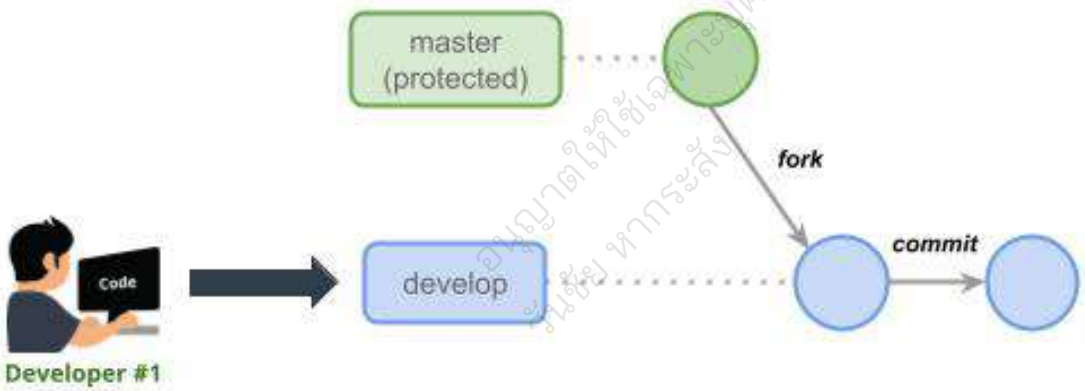
Git Features: Branching

Git Branching means you diverge from the main line of development and continue to do work without messing with that main line



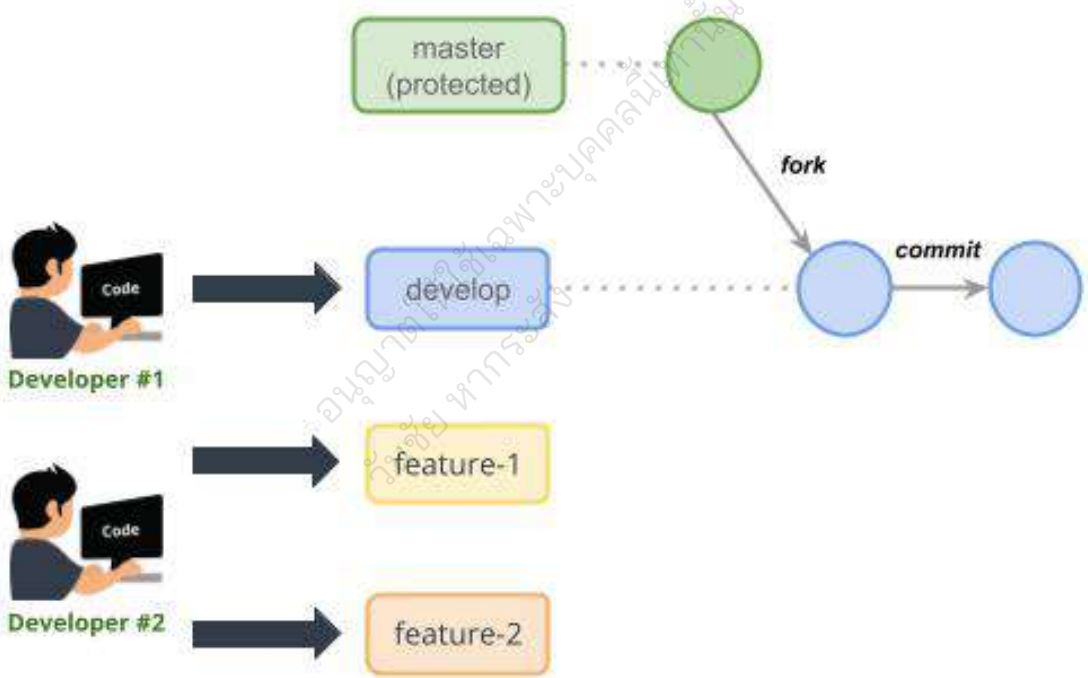
Git Features: Branching

Git Branching means you diverge from the main line of development and continue to do work without messing with that main line



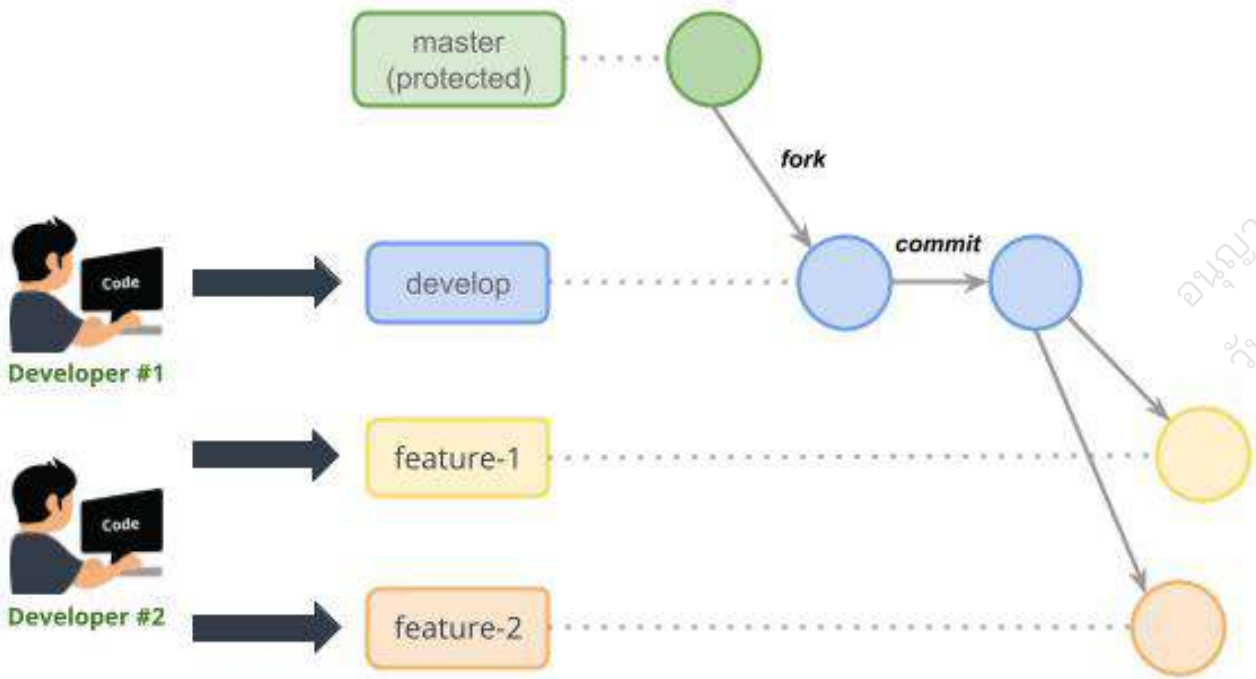
Git Features: Branching

Git Branching means you diverge from the main line of development and continue to do work without messing with that main line



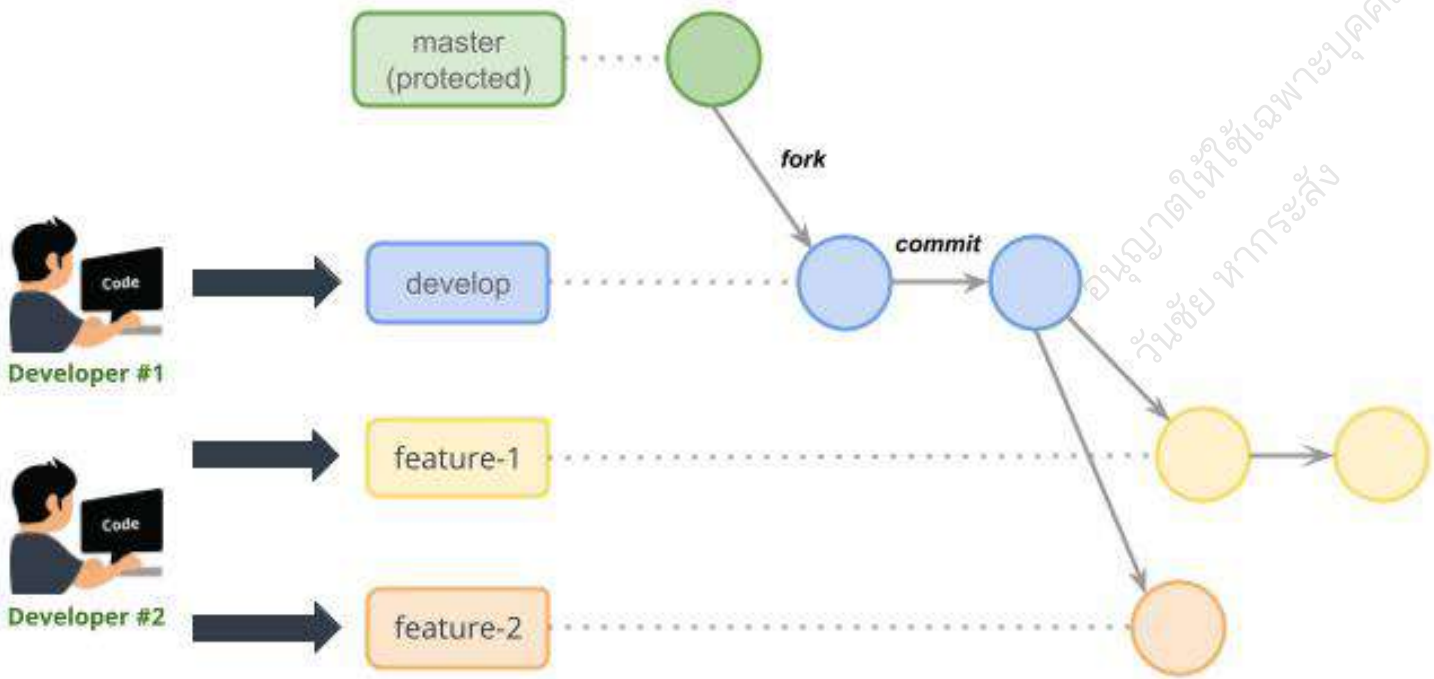
Git Features: Branching

Git Branching means you diverge from the main line of development and continue to do work without messing with that main line



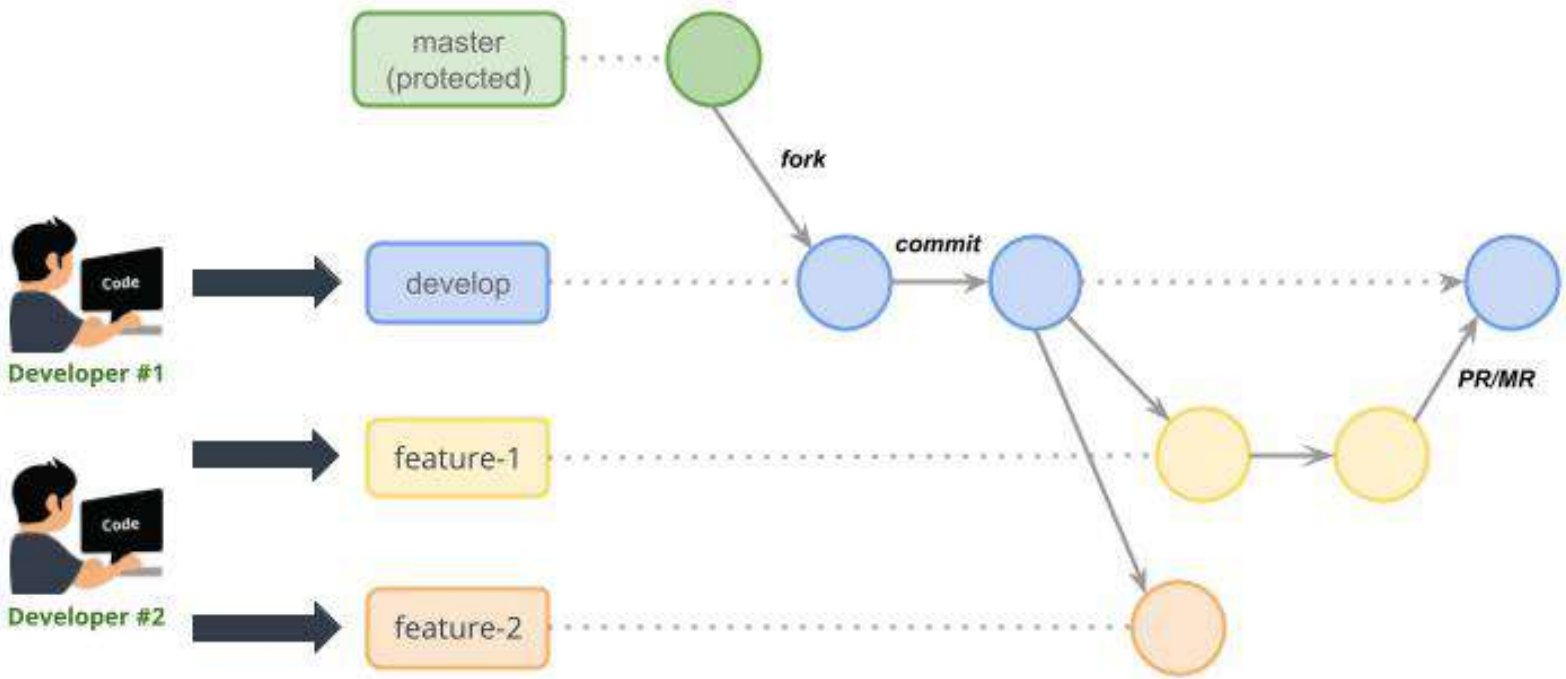
Git Features: Branching

Git Branching means you diverge from the main line of development and continue to do work without messing with that main line



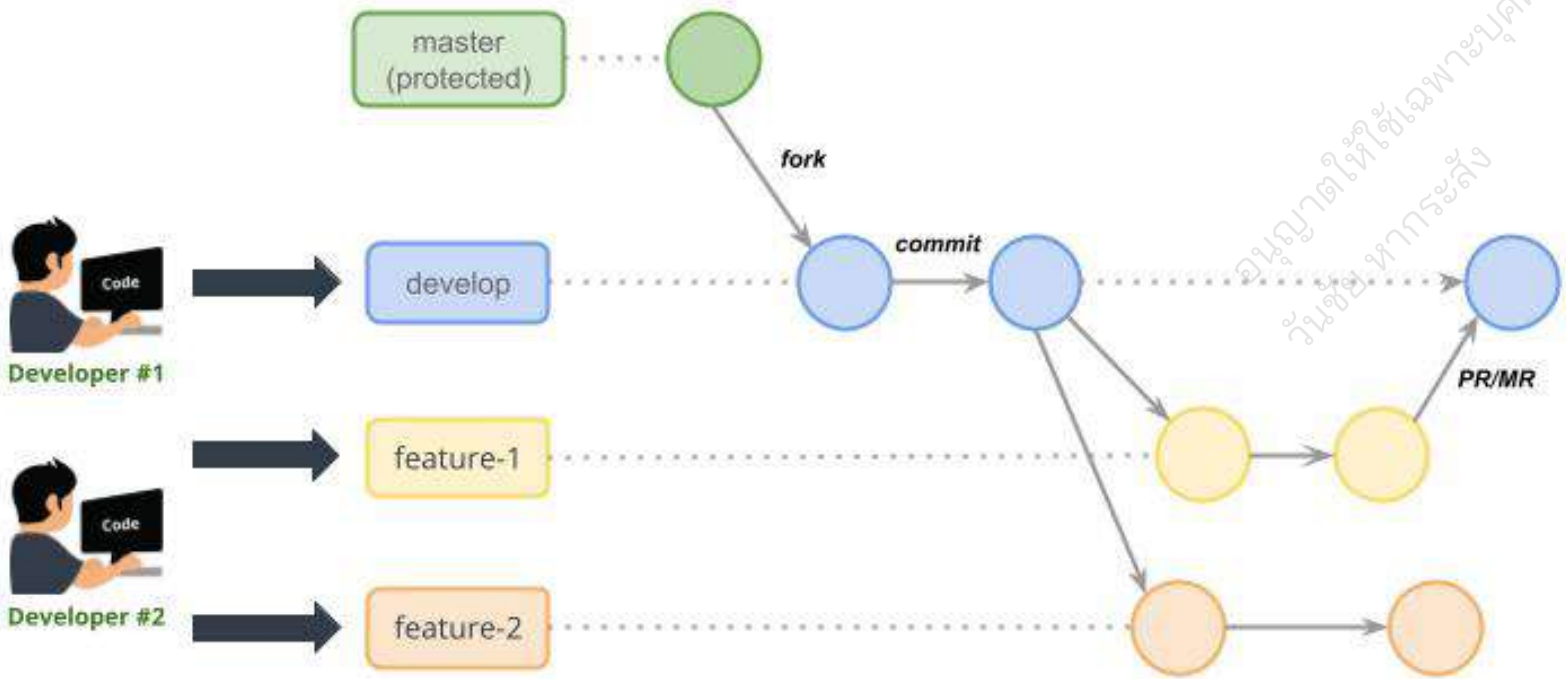
Git Features: Branching

Git Branching means you diverge from the main line of development and continue to do work without messing with that main line



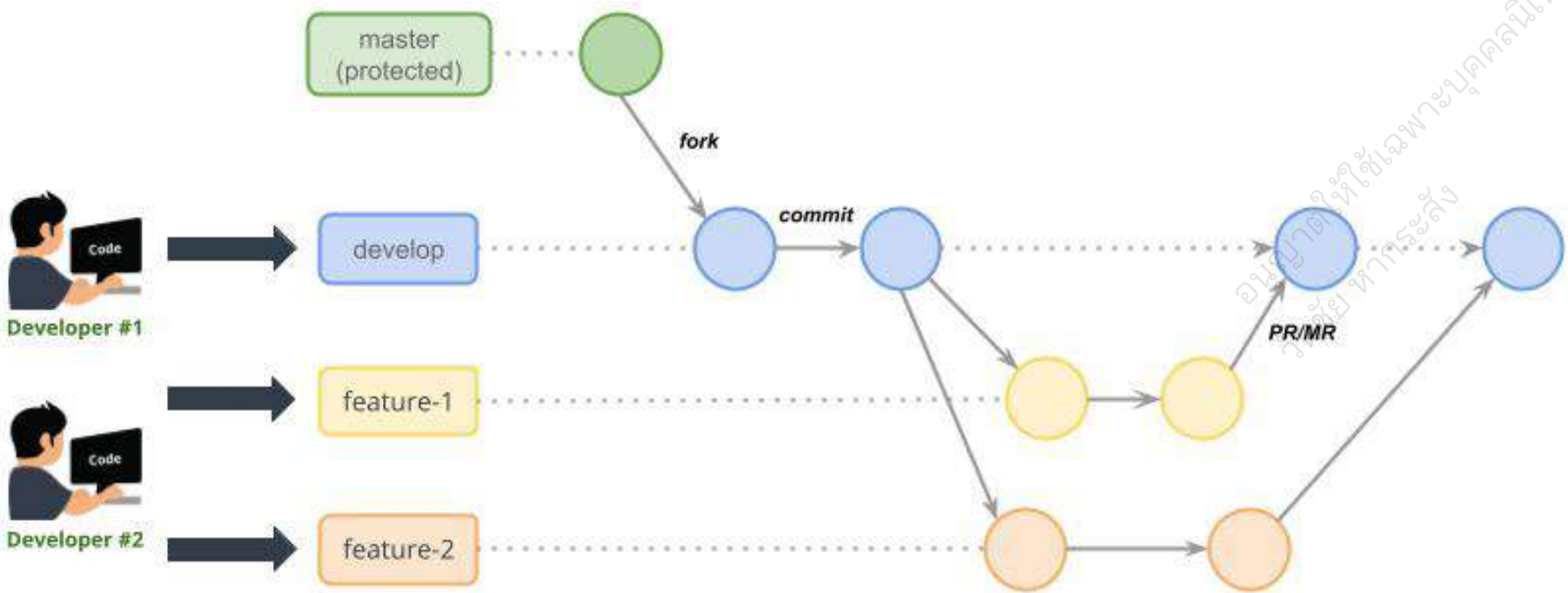
Git Features: Branching

Git Branching means you diverge from the main line of development and continue to do work without messing with that main line



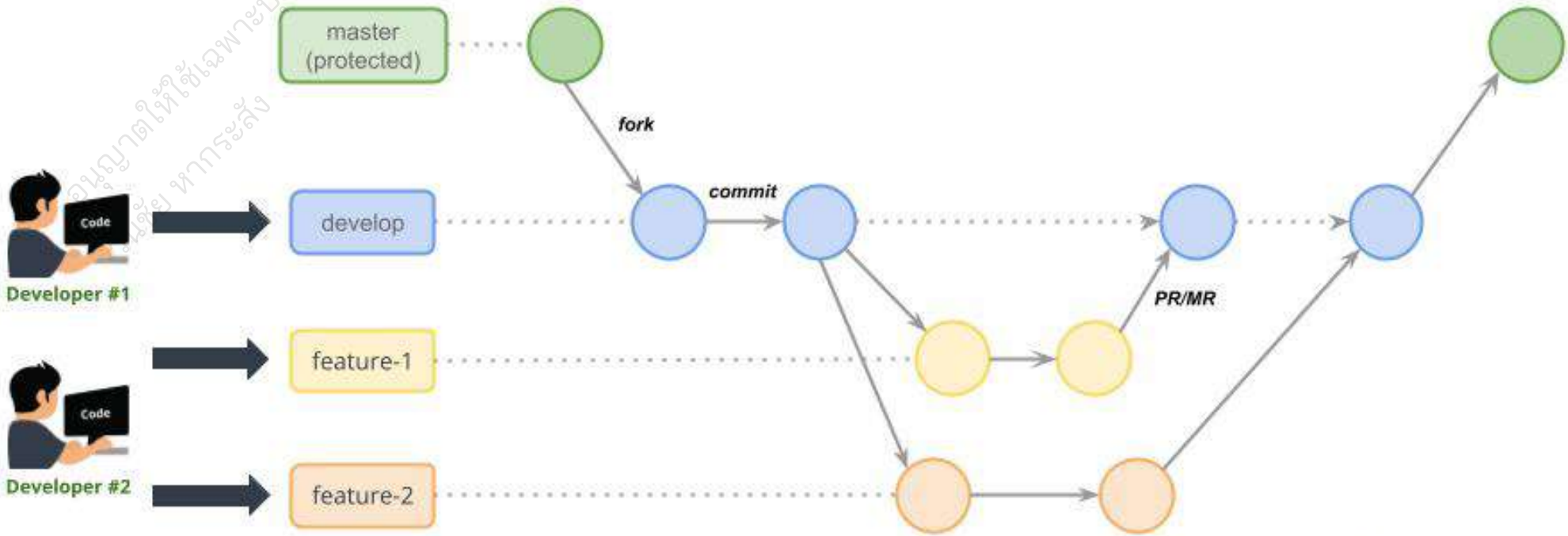
Git Features: Branching

Git Branching means you diverge from the main line of development and continue to do work without messing with that main line



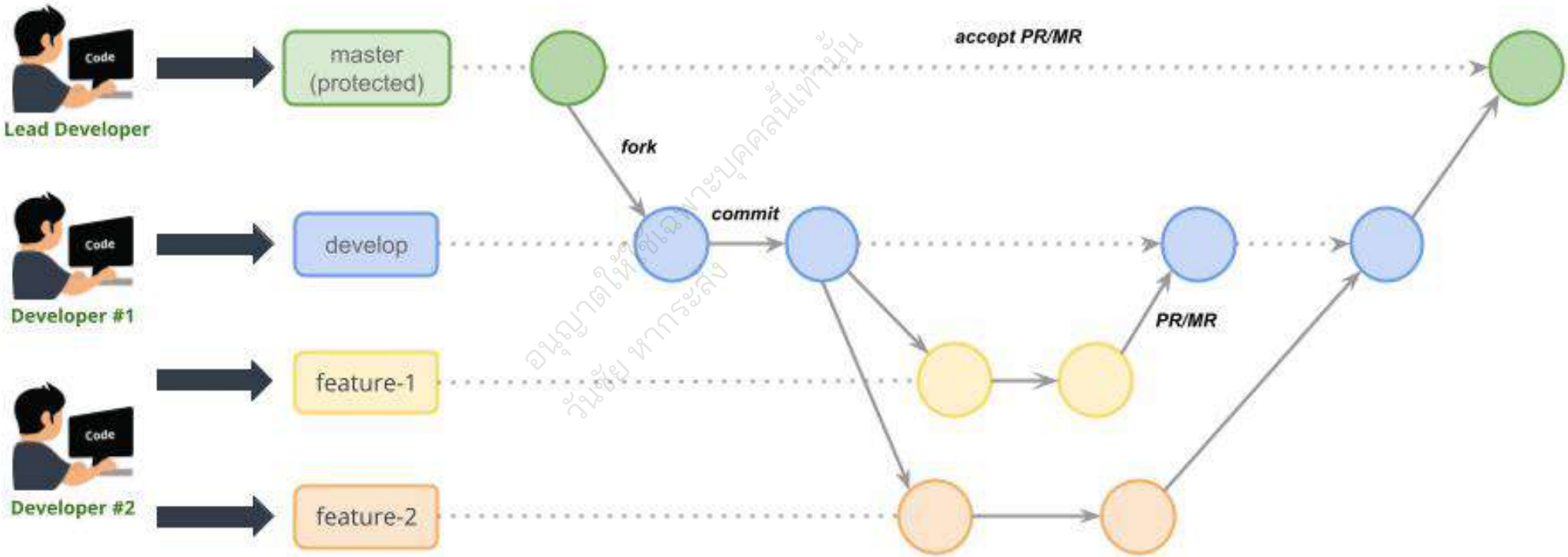
Git Features: Branching

Git Branching means you diverge from the main line of development and continue to do work without messing with that main line



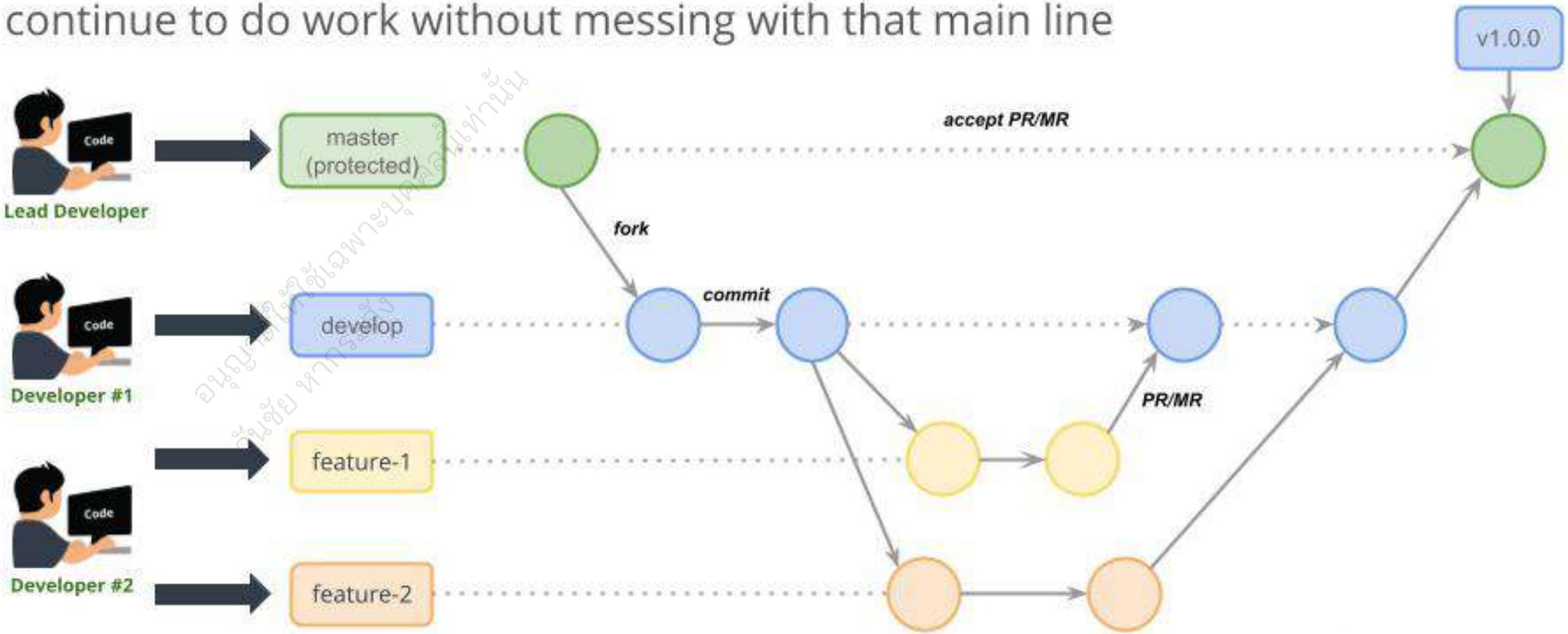
Git Features: Branching

Git Branching means you diverge from the main line of development and continue to do work without messing with that main line



Git Features: Branching

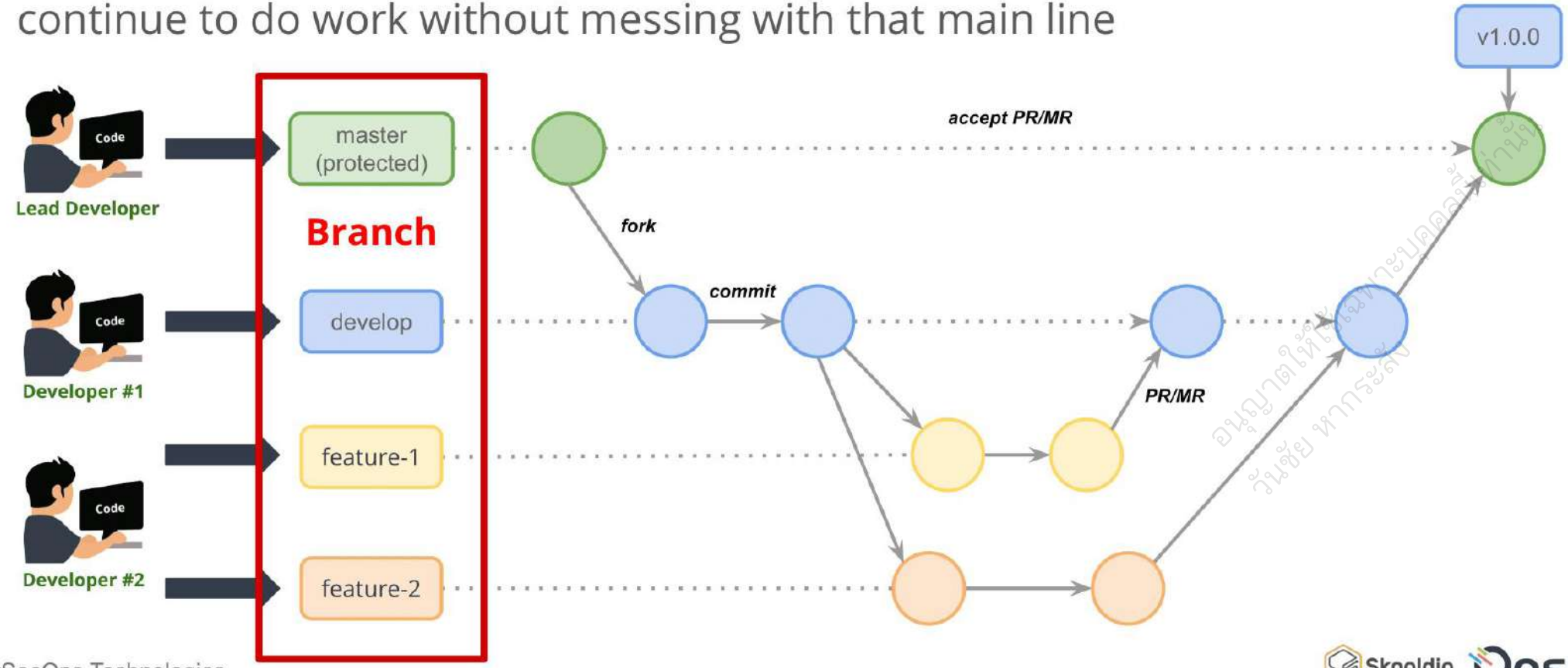
Git Branching means you diverge from the main line of development and continue to do work without messing with that main line



Skooldio Opsta

Git Features: Branching

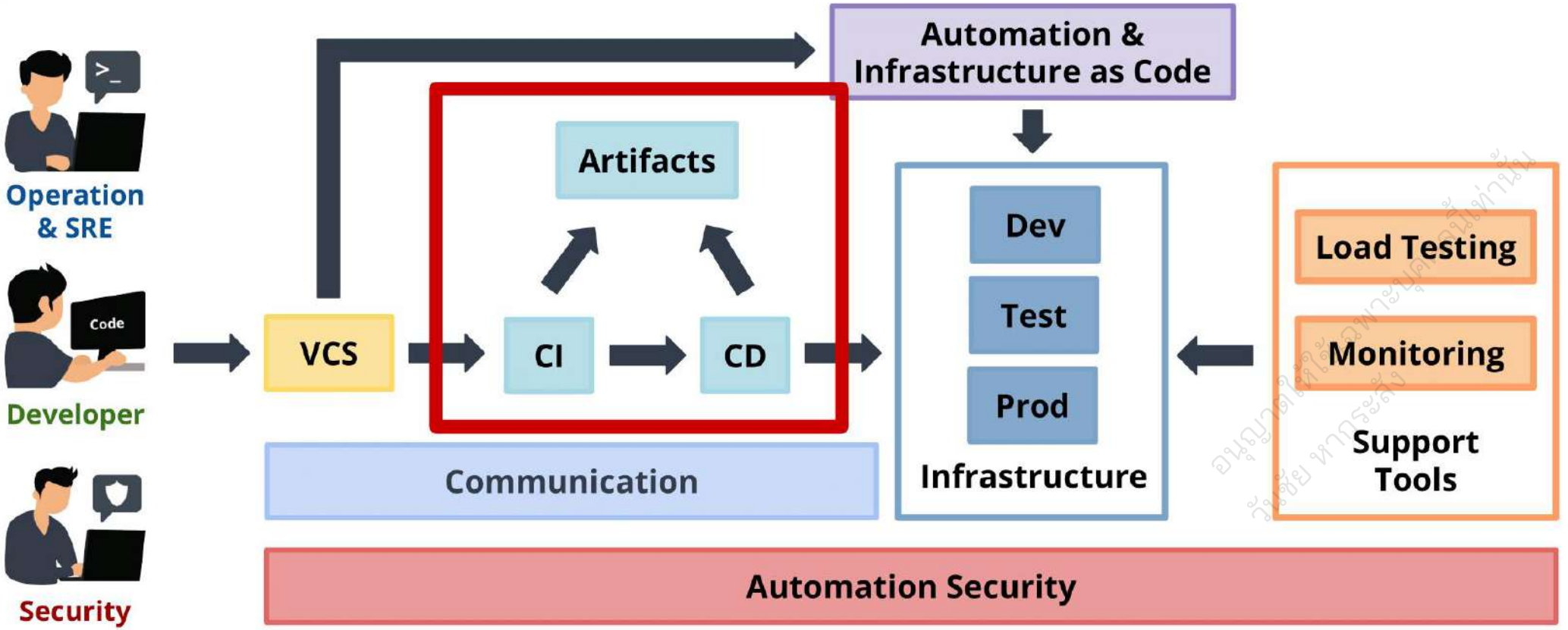
Git Branching means you diverge from the main line of development and continue to do work without messing with that main line



Skooldio Opsta

CI/CD and Artifacts

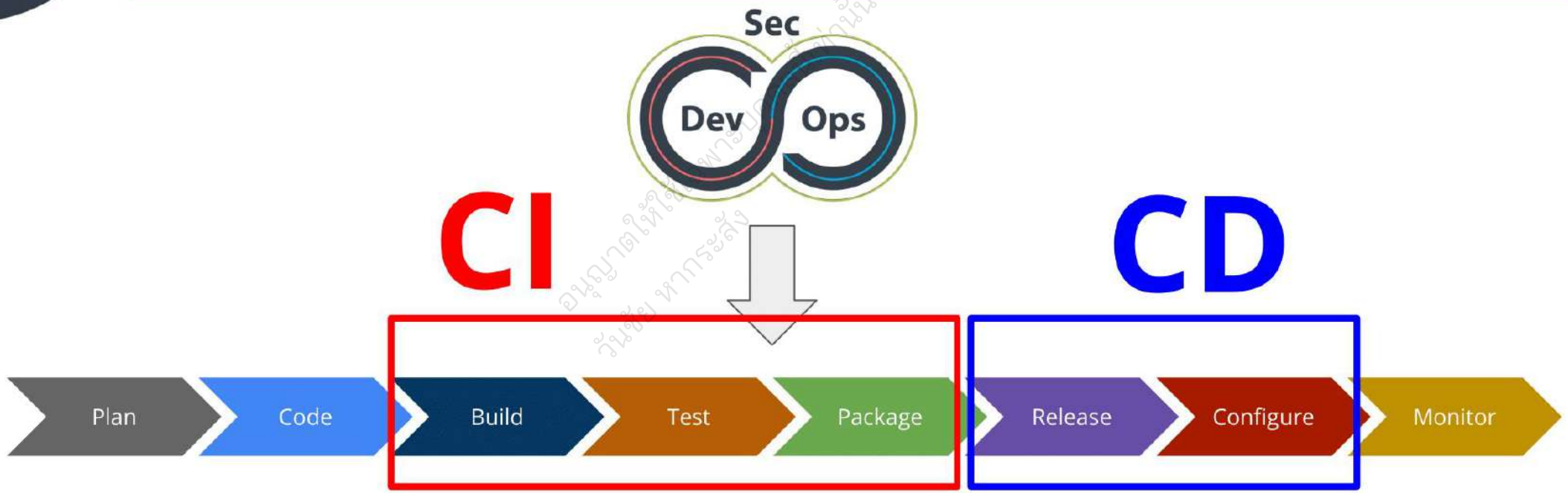
CI/CD and Artifacts



Continuous Integration (CI) and Continuous Delivery/Deployment (CD)

CI/CD Flow

CI/CD Flow



Continuous Integration

Technologies

Skooldio

Opsta

Type of Testing

Build → Test → Package → Release → Configure

- Unit Test
- Integration Test
- Functional Test
- Performance Test
- Acceptance Test
- UI Test

Test Automation Tools (1)

Build | Test | Package | Release | Configure

Unit Test

Browser Simulation Test

Test Automation Tools (2)

Build | Test | Package | Release | Configure

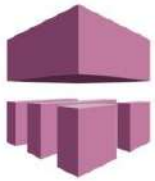
Behaviour Driven Development (BDD)

cucumber

Jasmine

appium

Cloud-Based Test



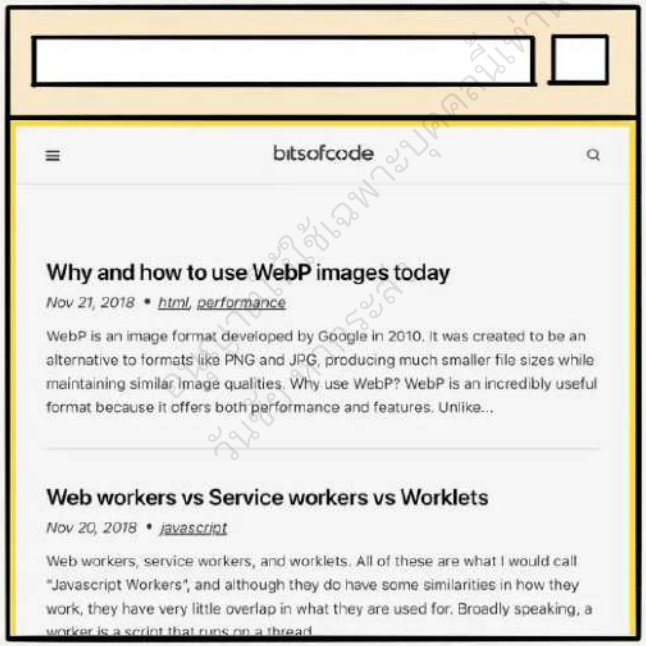
SAUCELABS

Firebase Test Lab

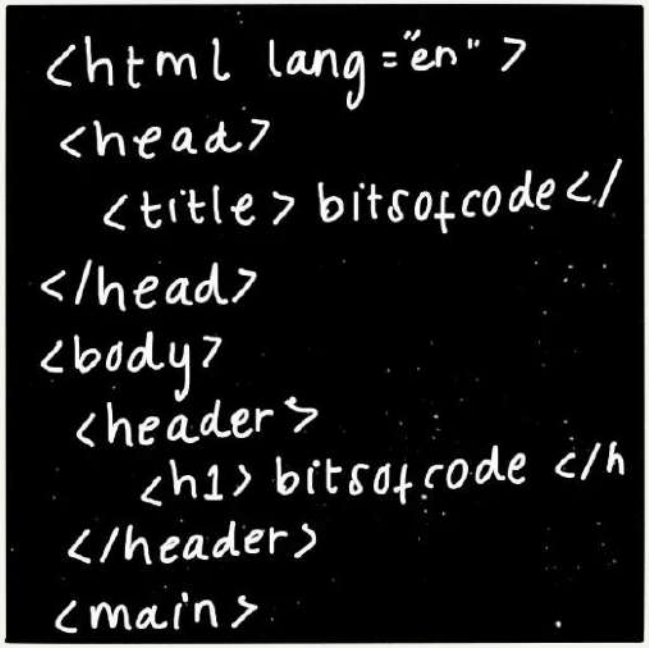
App Center

BrowserStack

Headless Browser



headful



headless

Type of Artifacts

Build → Test → **Package** → Release → Configure

docker

Gradle

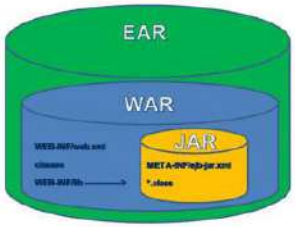
npm

python

debian

rpm

nuget



HTML

<xml />

Go

sbt

Packaging Tools (Artifacts Server)

Build → Test → **Package** → Release → Configure

JFrog Artifactory

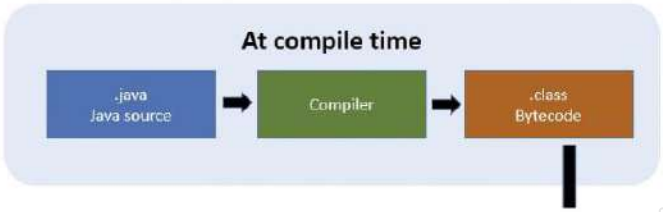
GitLab

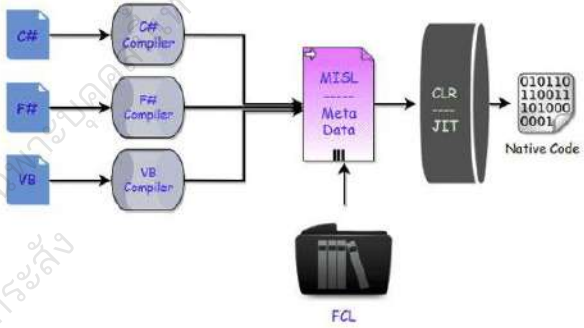
Nexus

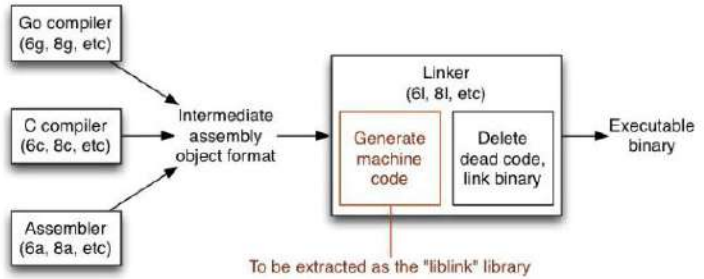
HARBOR™

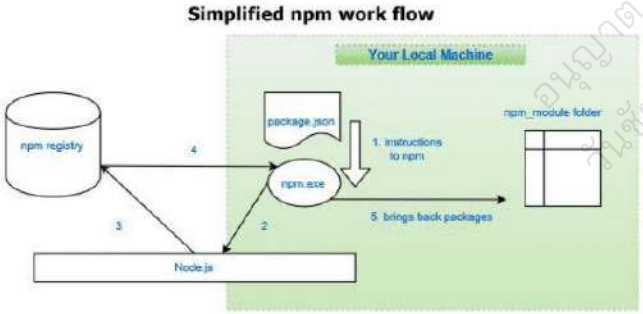
Build

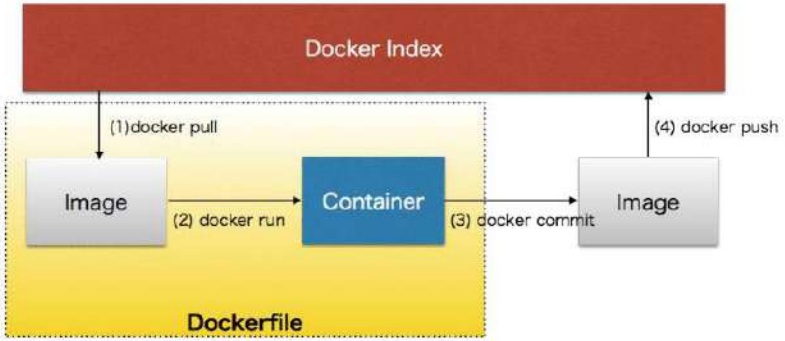
Build | Test | Package | Release | Configure







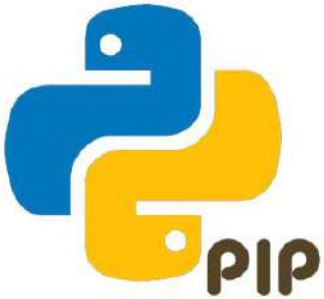


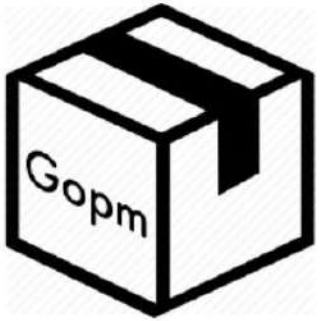


Skooldio Opsta

Package Manager

Build | Test | Package | Release | Configure





Type of Testing

Build → Test → Package → Release → Configure

Unit Test

Integration Test

Functional Test

Performance Test

Acceptance Test

UI Test

Test Automation Tools (1)

Build | Test | Package | Release | Configure

Unit Test

JUnit 5 | nunit | Jest | nose is nicer testing for python

Browser Simulation Test

Selenium | Puppeteer | cypress | MOCHA

Test Automation Tools (2)

Build | Test | Package | Release | Configure

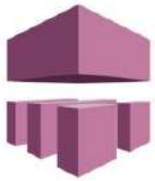
Behaviour Driven Development (BDD)

cucumber

Jasmine

appium

Cloud-Based Test



SAUCELABS

Firebase Test Lab

App Center

BrowserStack

Test Automation Tools (2)


```
*** Test Cases ***
Valid Login
    Given browser is opened to login page
    When user "demo" logs in with password "mode"
    Then welcome page should be open
```

Test Automation Tools (2)

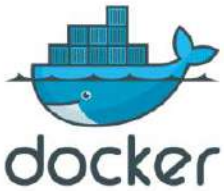
Build → Test → Package → Release → Configure

```
*** Keywords ***
Open Browser To Login Page
    Open Browser    ${LOGIN URL}    ${BROWSER}
    Maximize Browser Window
    Set Selenium Speed ${DELAY}
    Login Page Should Be Open

Login Page Should Be Open
    Title Should Be Login Page

Go To Login Page
    Go To    ${LOGIN URL}
    Login Page Should Be Open
```

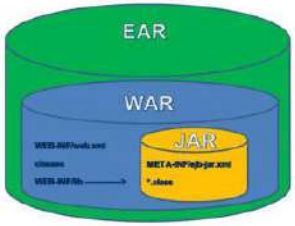
```
Input Username
    [Argument]    ${LOGIN URL}
    Input Text    username_field    ${username}

Input Password
    [Argument]    ${password}
    Input Text    password_field    ${password}

Submit Credentials
    Click Button    login_button

Welcome Page Should Be Open
    Location Should Be    ${WELCOME URL}
    Title Should Be    Welcome Page
```

Type of Artifacts

Build → Test → **Package** → Release → Configure



debian



Go

sbt

Packaging Tools (Artifacts Server)

Build → Test → **Package** → Release → Configure

JFrog Artifactory

GitLab

Nexus

HARBOR™

Skooldio Opsta

Continuous Delivery and Deployment

DevSecOps Technologies

Skooldio

Opsta

Release Strategies

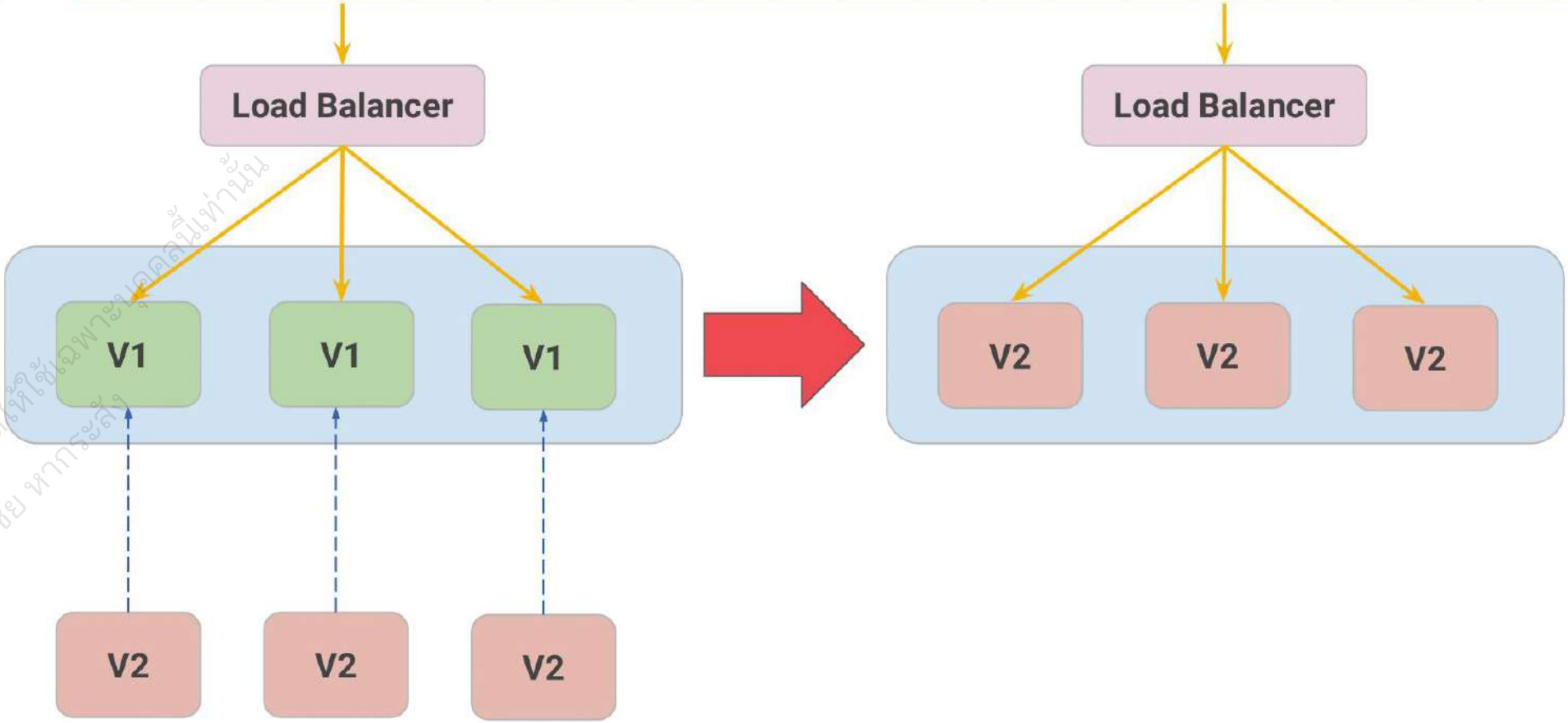
Deployment

- Recreate / Replace
- Rolling Deployment
- Blue-Green or Red-Black Deployment
- Canary Deployment

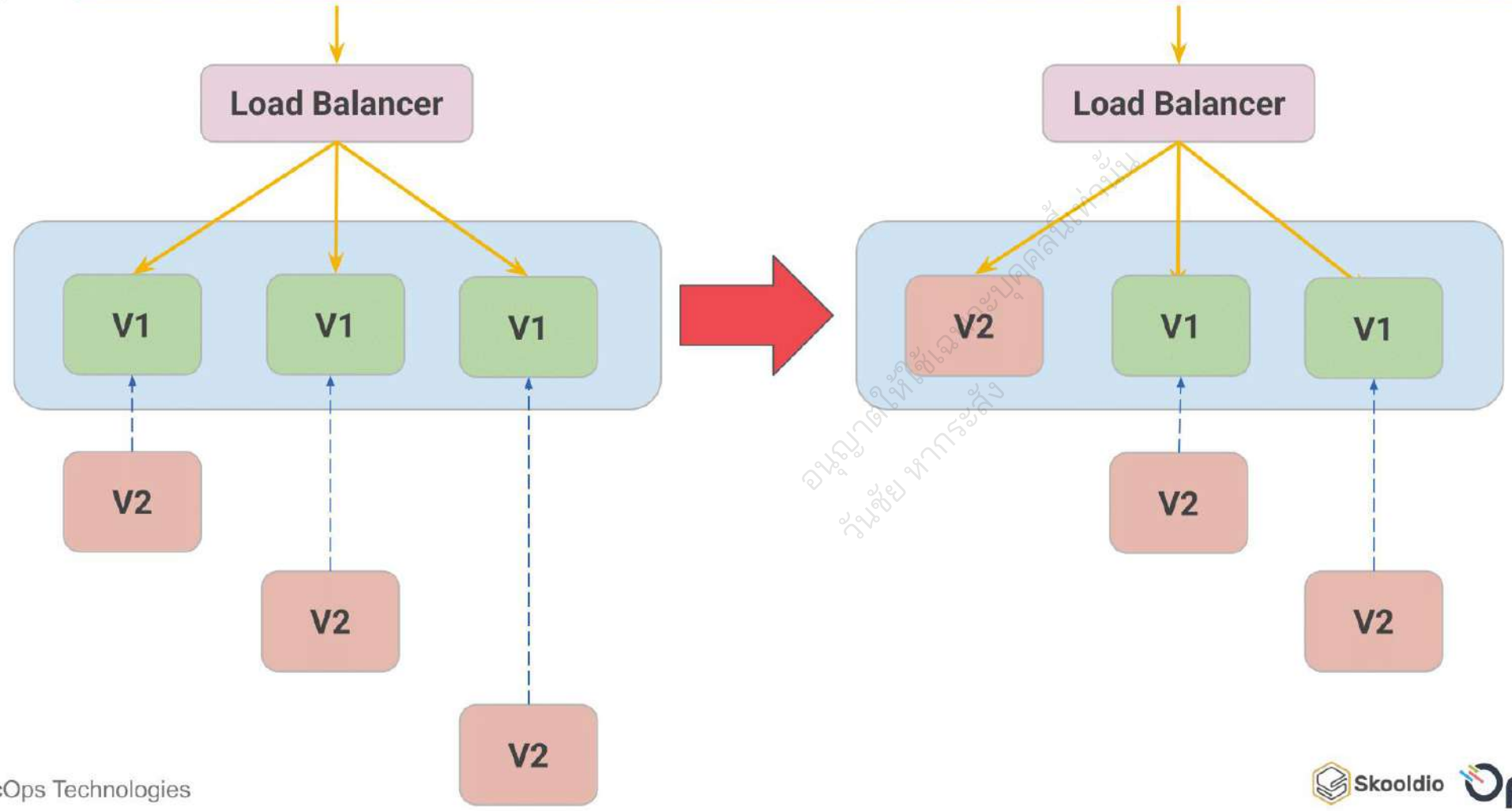
Feature

- A/B Testing
- Environment Separation
- Feature Toggles or Feature Flag
- User Targeted Testing

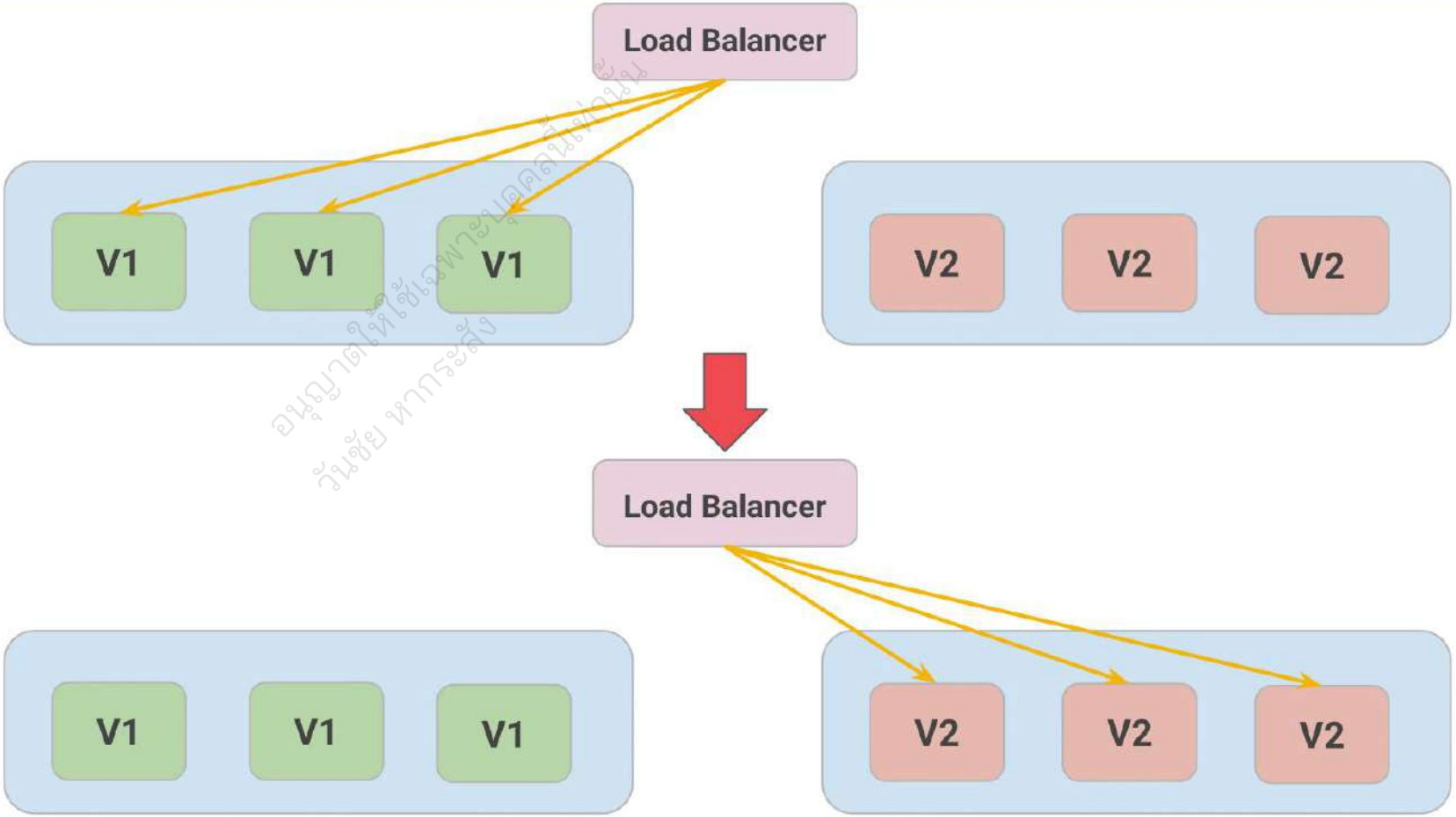
Recreate / Replace Deployment



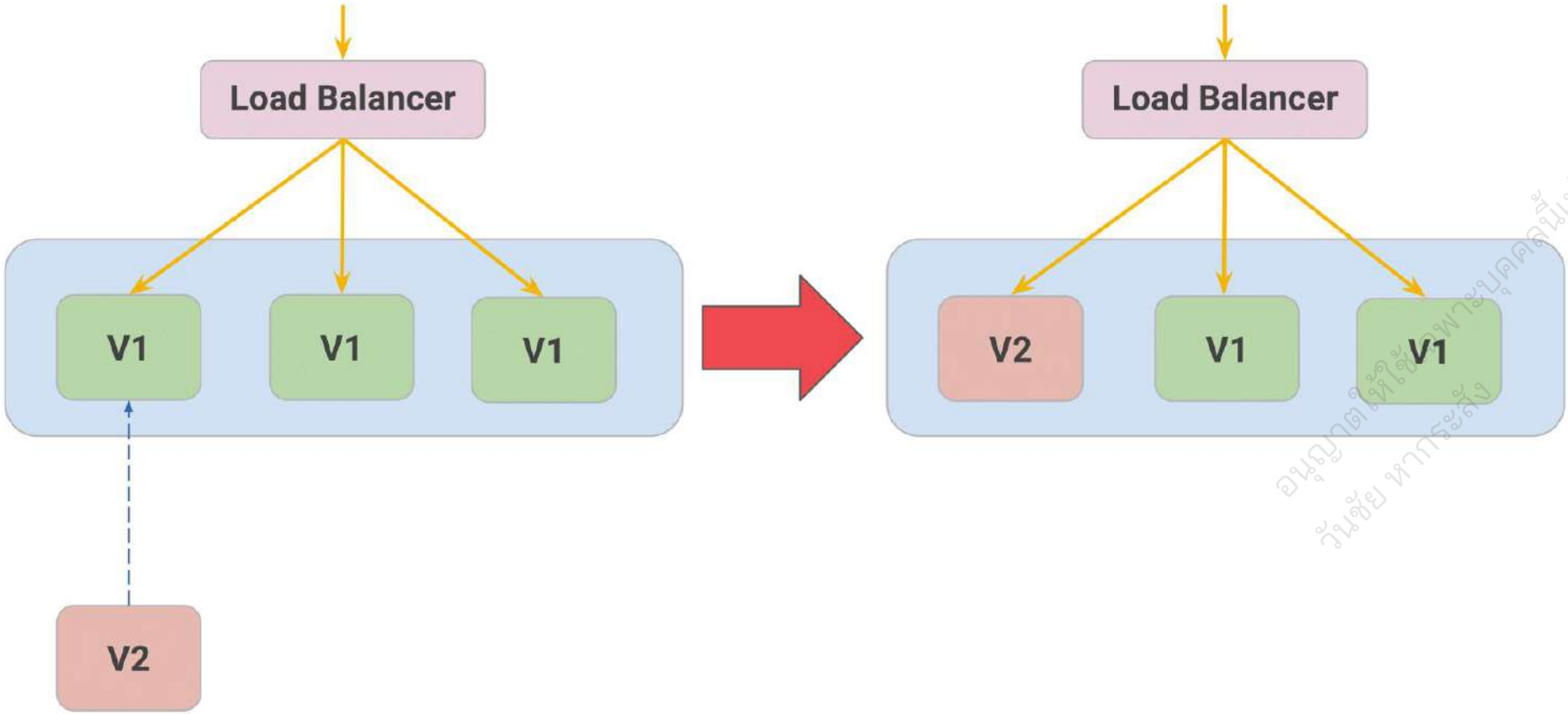
Rolling Deployment



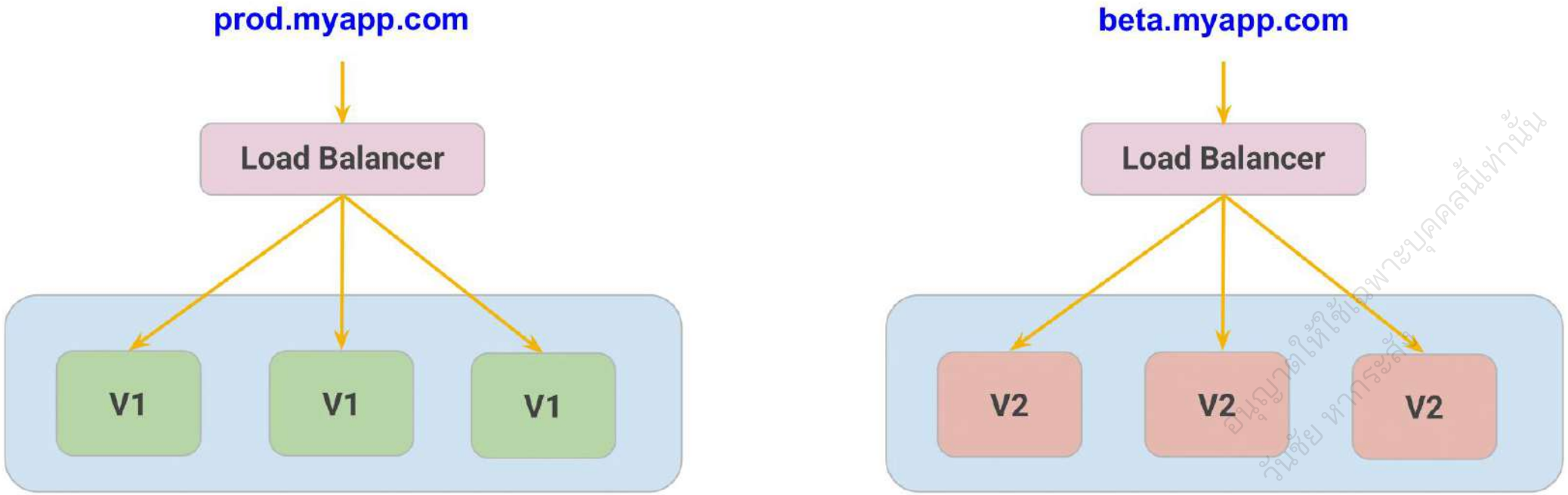
Blue-Green / Red-Black Deployment



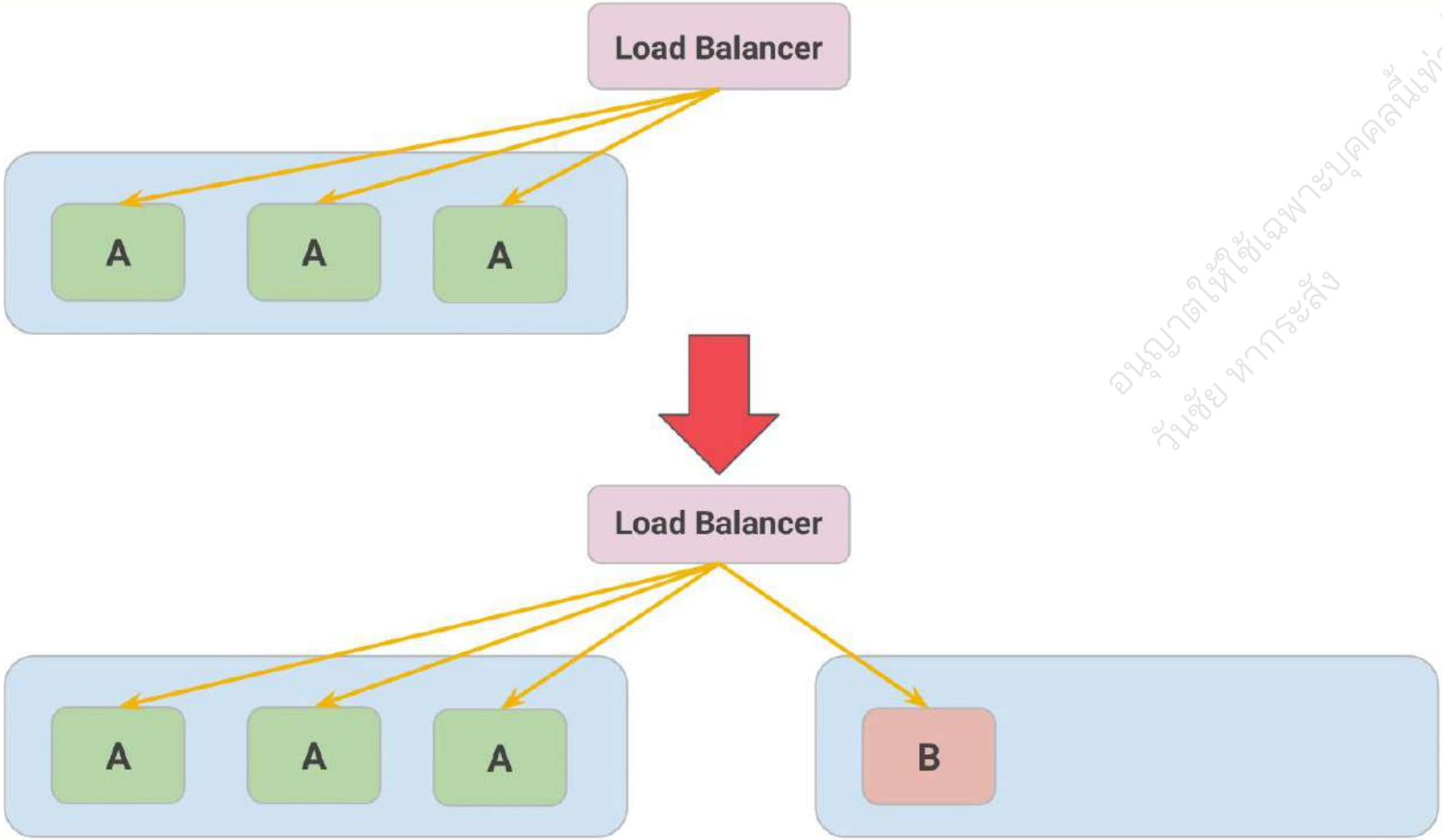
Canary Deployment



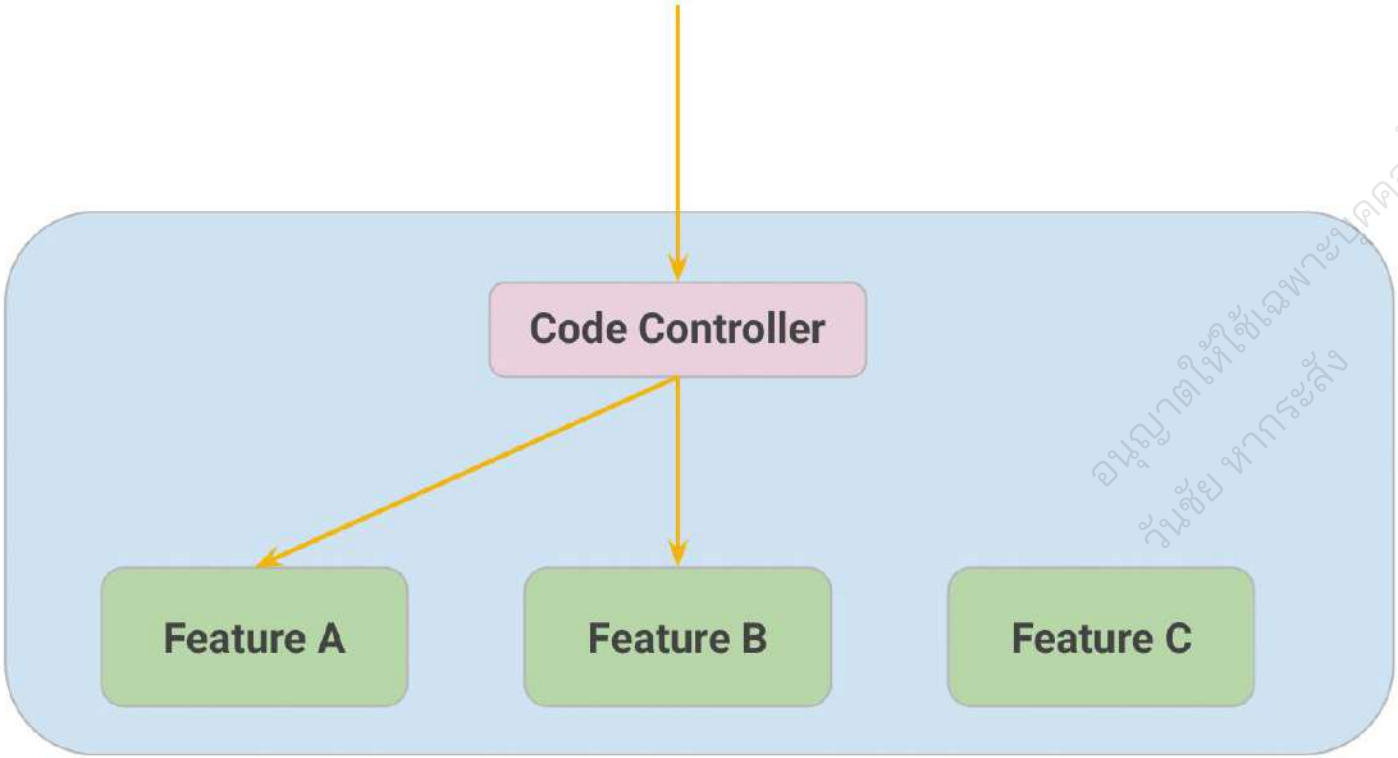
Environment Separation



A/B Testing

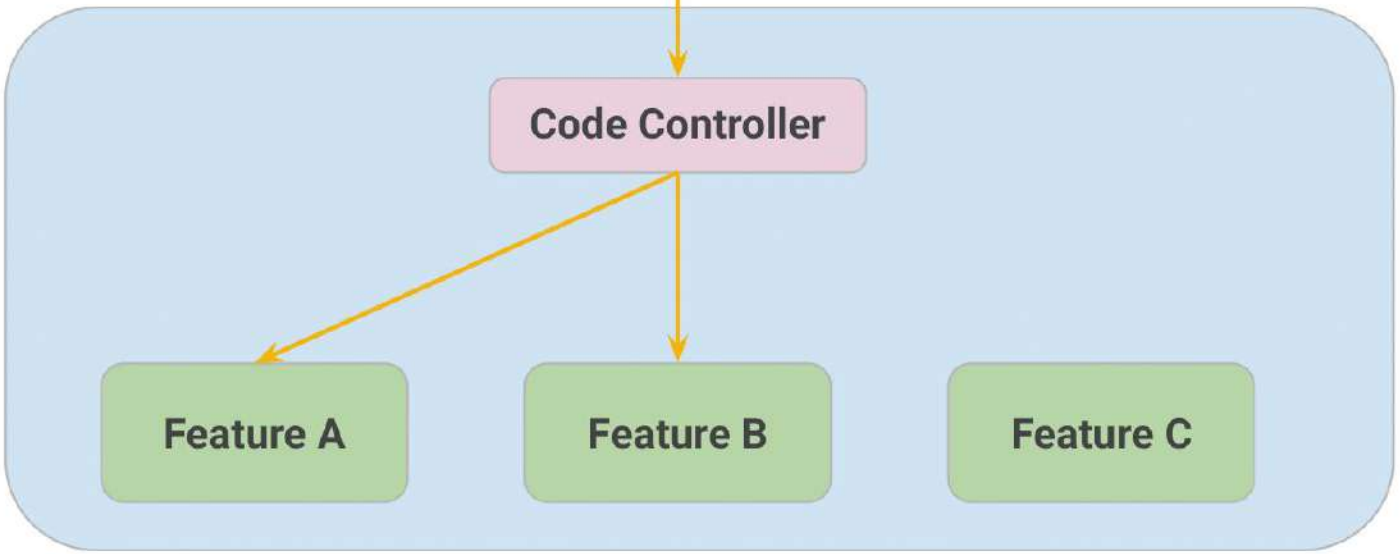


Feature Toggle or Feature Flag



User Targeted Testing

- Sticky Session
- User Agent
- Source IP Address
- Header
- Cookie
- User Specific

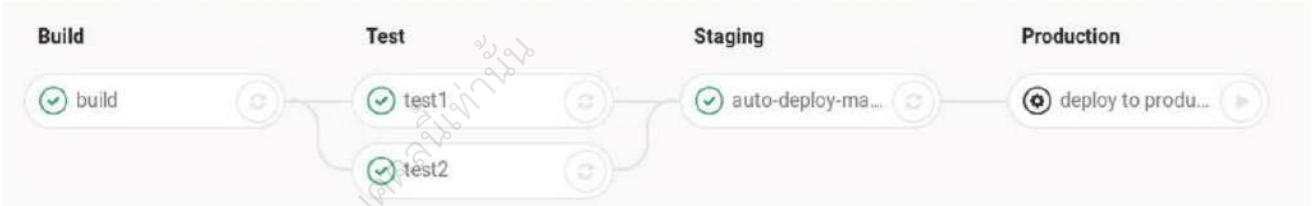


Popular CI/CD Tools

On-Premise Open Source	      
On-Premise Commercial	 
On-Cloud	      

Skooldio Opsta

Pipeline as Code




```
test:
  script:
  - apt-get update -qy
  - apt-get install -y nodejs
  - bundle install --path /cache
  - bundle exec rake db:create RAILS_ENV=test
  - bundle exec rake test

staging:
  type: deploy
  script:
  - gem install dpl
  - dpl --provider=heroku --app=gitlab-ci-ruby-test-staging --api-key=$HEROKU_STAGING_API_KEY
  only:
  - master

production:
  type: deploy
  script:
  - gem install dpl
  - dpl --provider=heroku --app=gitlab-ci-ruby-test-prod --api-key=$HEROKU_PRODUCTION_API_KEY
  only:
  - tags
```

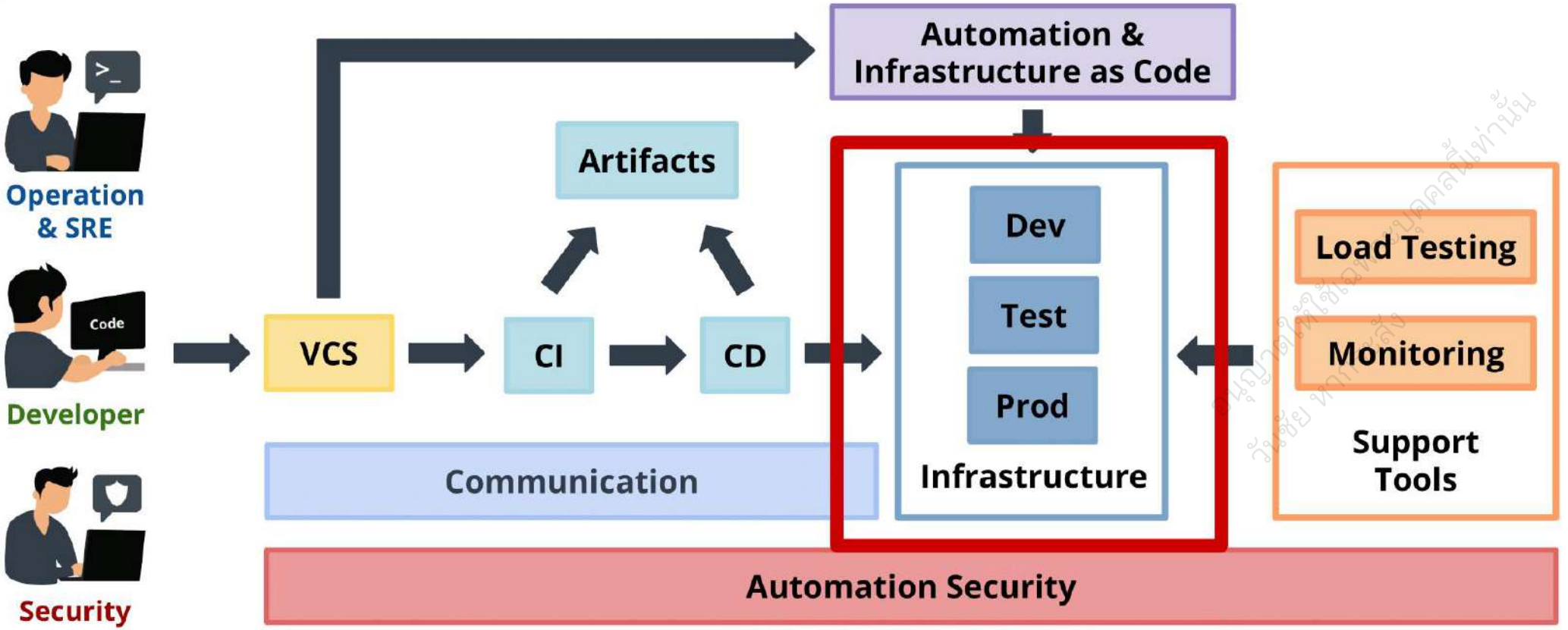
Modern Infrastructure

Technologies

Skooldio

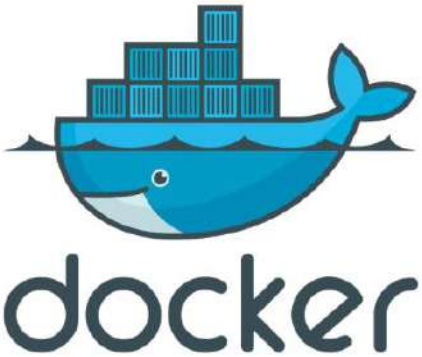
Opsta

Modern Infrastructure



Modern Infrastructure

Cloud



Cloud

DevSecOps & Cloud Characteristics

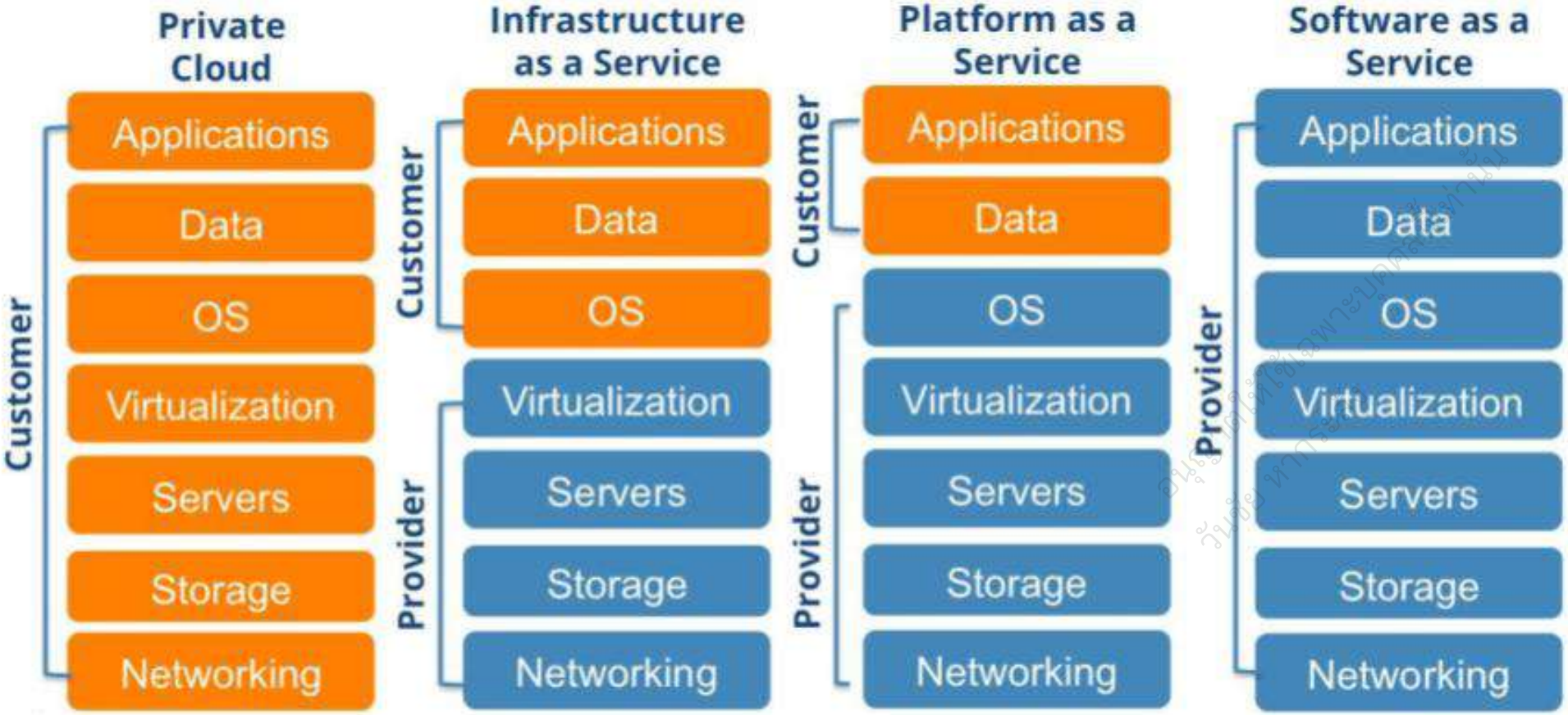
- Broad network access
- Multi-tenancy and resource pooling
- On-demand self-service
- Rapid elasticity and scalability
- Measured service

DevSecOps & Cloud Characteristics

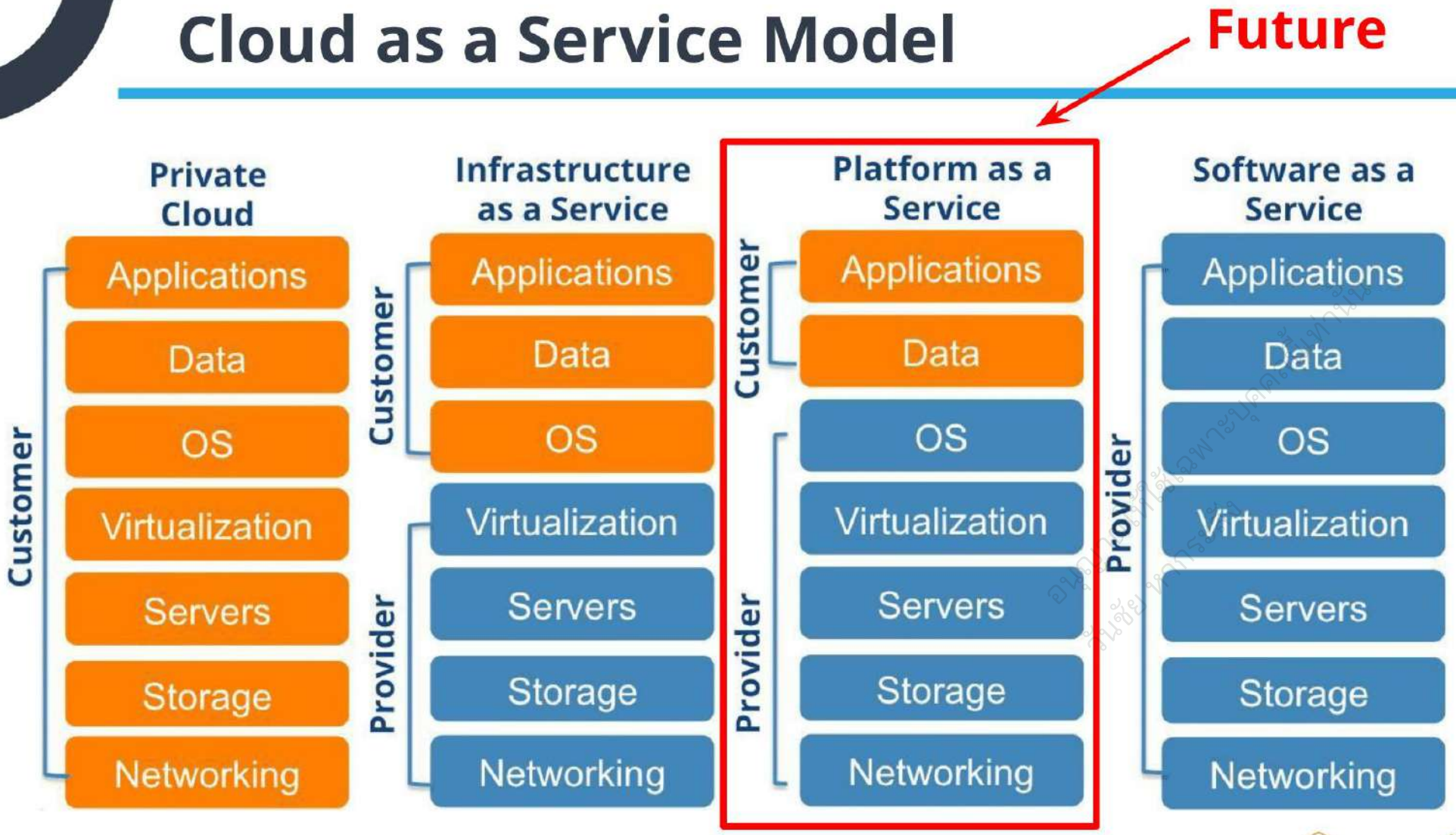
- Broad network access
- Multi-tenancy and resource pooling
- On-demand self-service
- Rapid elasticity and scalability
- Measured service

DevSecOps Core Value

Cloud as a Service Model



Cloud as a Service Model



Type of Cloud

Private Cloud

vmware®
NUTANIX™
openstack®

Public Cloud

amazon web services
Azure
Google Cloud
Alibaba Cloud
HUAWEI
Tencent Cloud

Hybrid Cloud

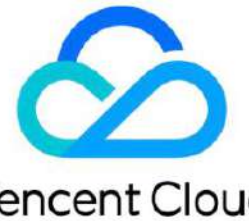
AWS Outposts
Anthos
Azure Stack

Skooldio Opsta

Type of Cloud

Hybrid Cloud

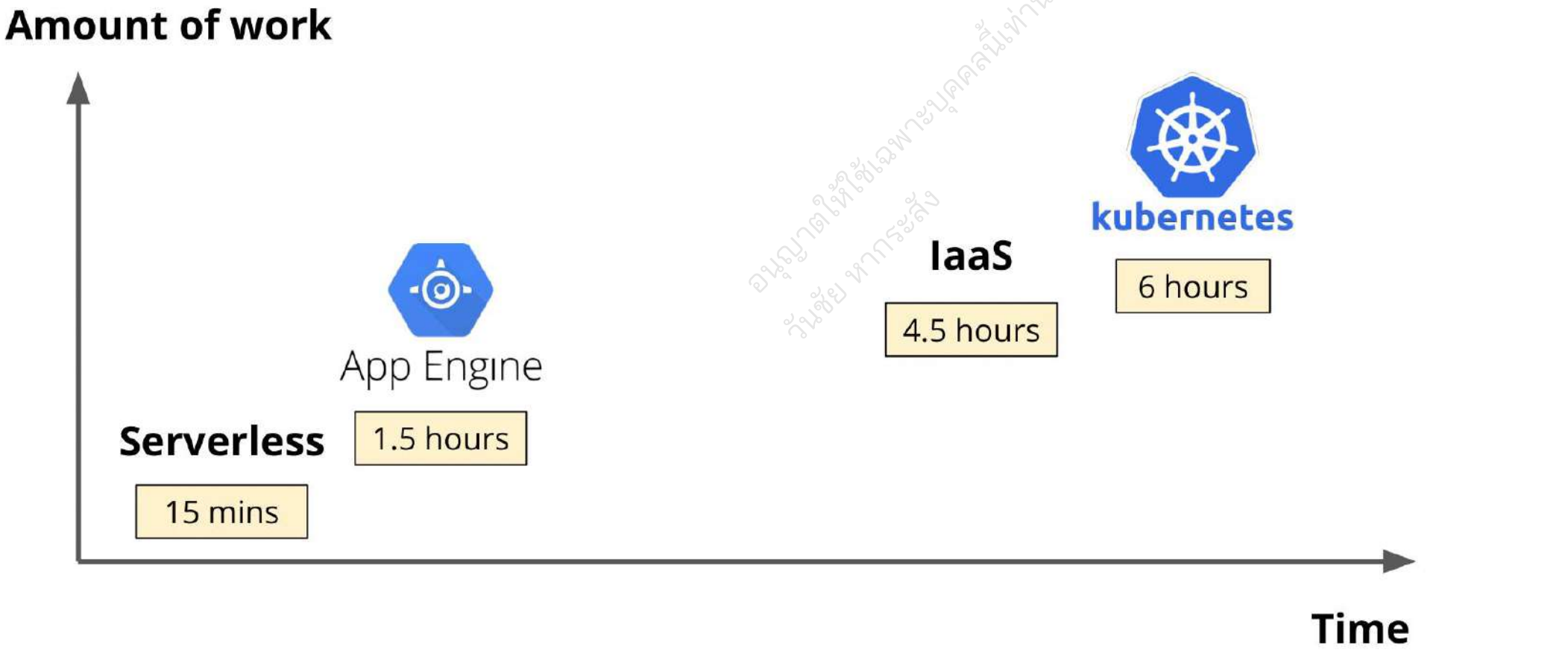
Multi Cloud



PaaS / Serverless / BaaS

Platform as a Service (PaaS)	HEROKU, kubernetes, App Engine
Serverless	AWS Lambda, Google Cloud Functions, Cloud Run, Kn, serverless
Backend as a Service (BaaS)	Firebase

Compare time-to-result



Why Serverless and why not?

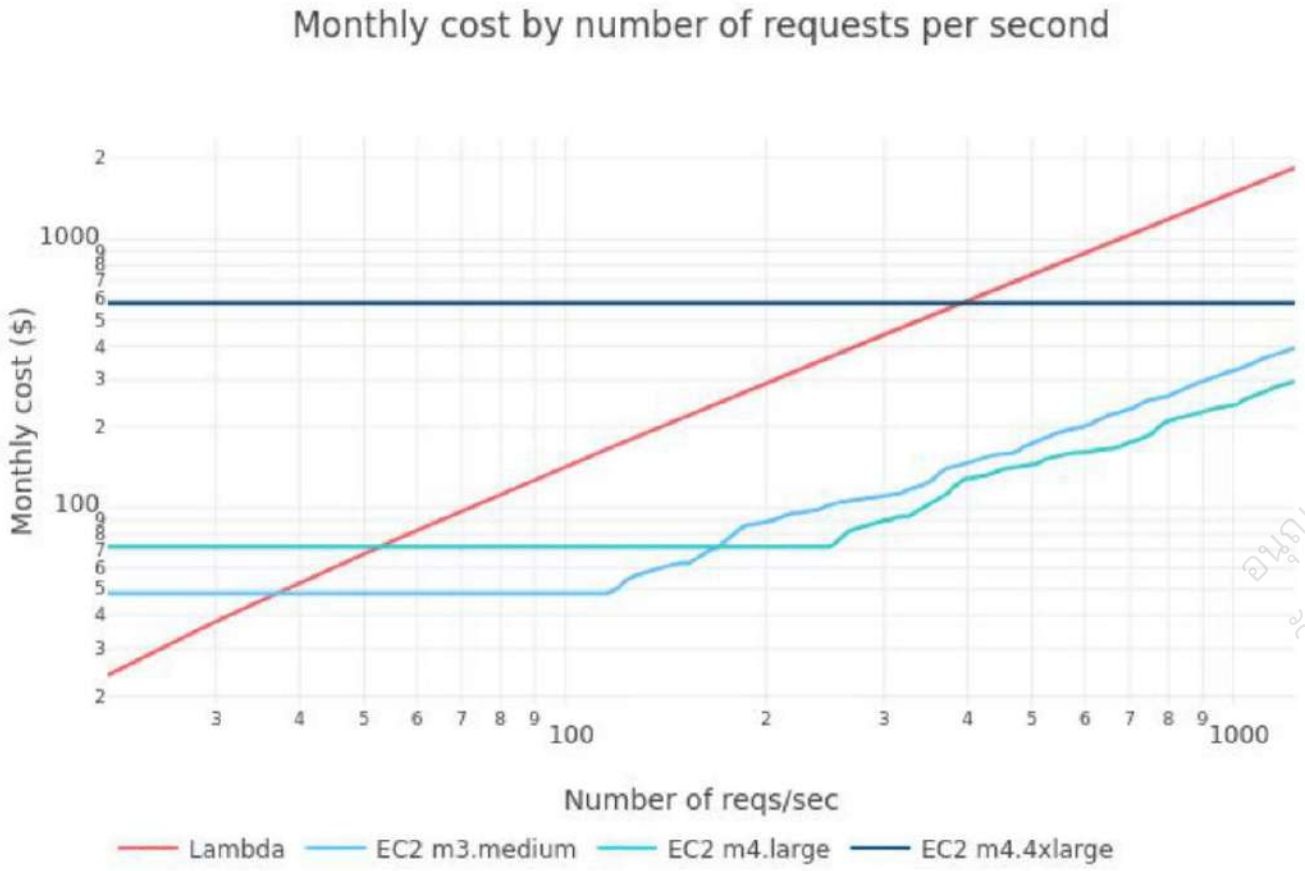
Pros

- Time-to-result
- Scale automatically
- Easy to maintain
- Focus on business
- Cheap for low traffic

Cons

- Vendor lock-in
- Cold start problem
- Hard to troubleshoot on complex system
- Expensive for high traffic
- Security concern
- CI/CD challenge

Server and Serverless Cost



https://www.bbva.com/en/economics-of-serverless/

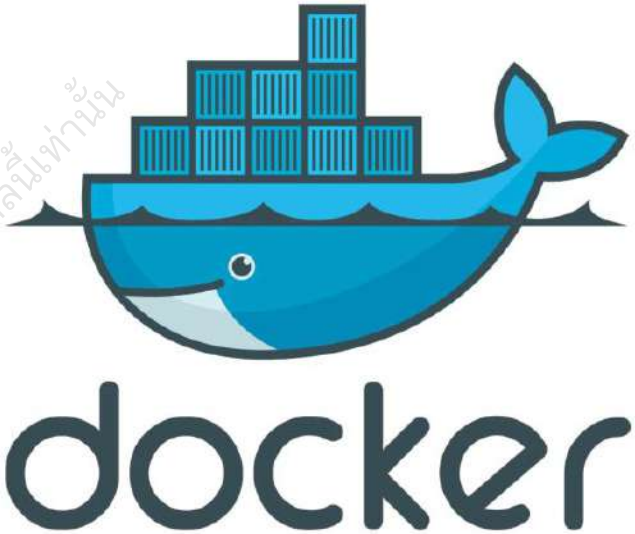
Docker

Technologies

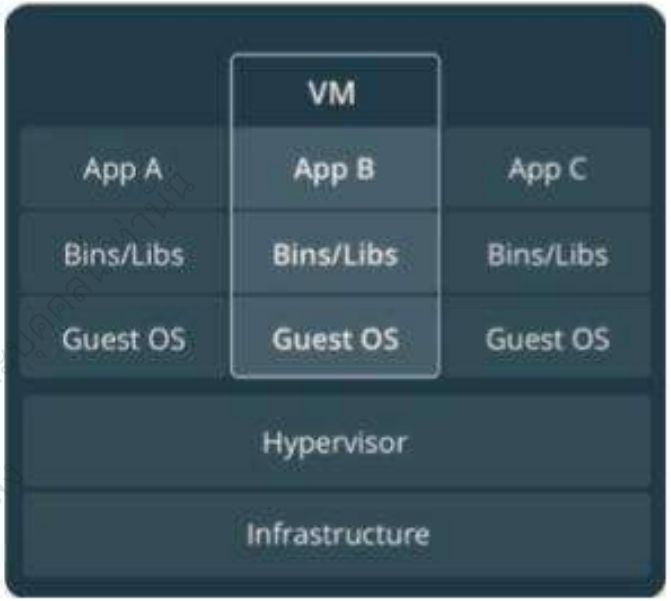
Skooldio

Opsta

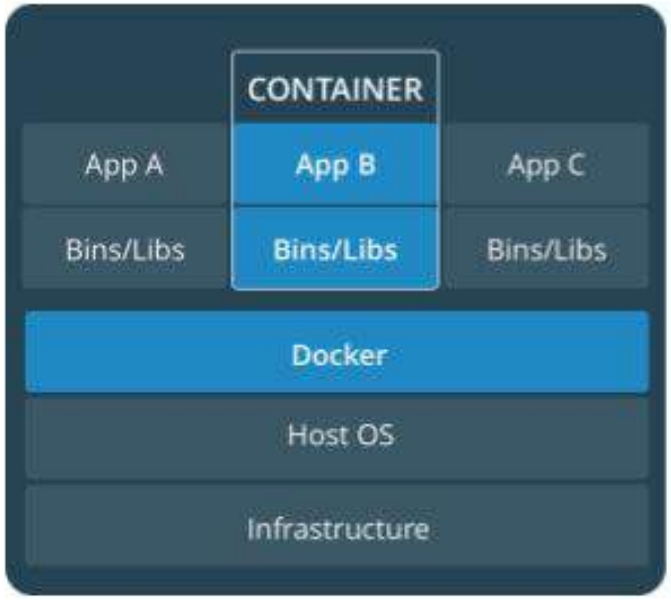
What is?



Docker Container

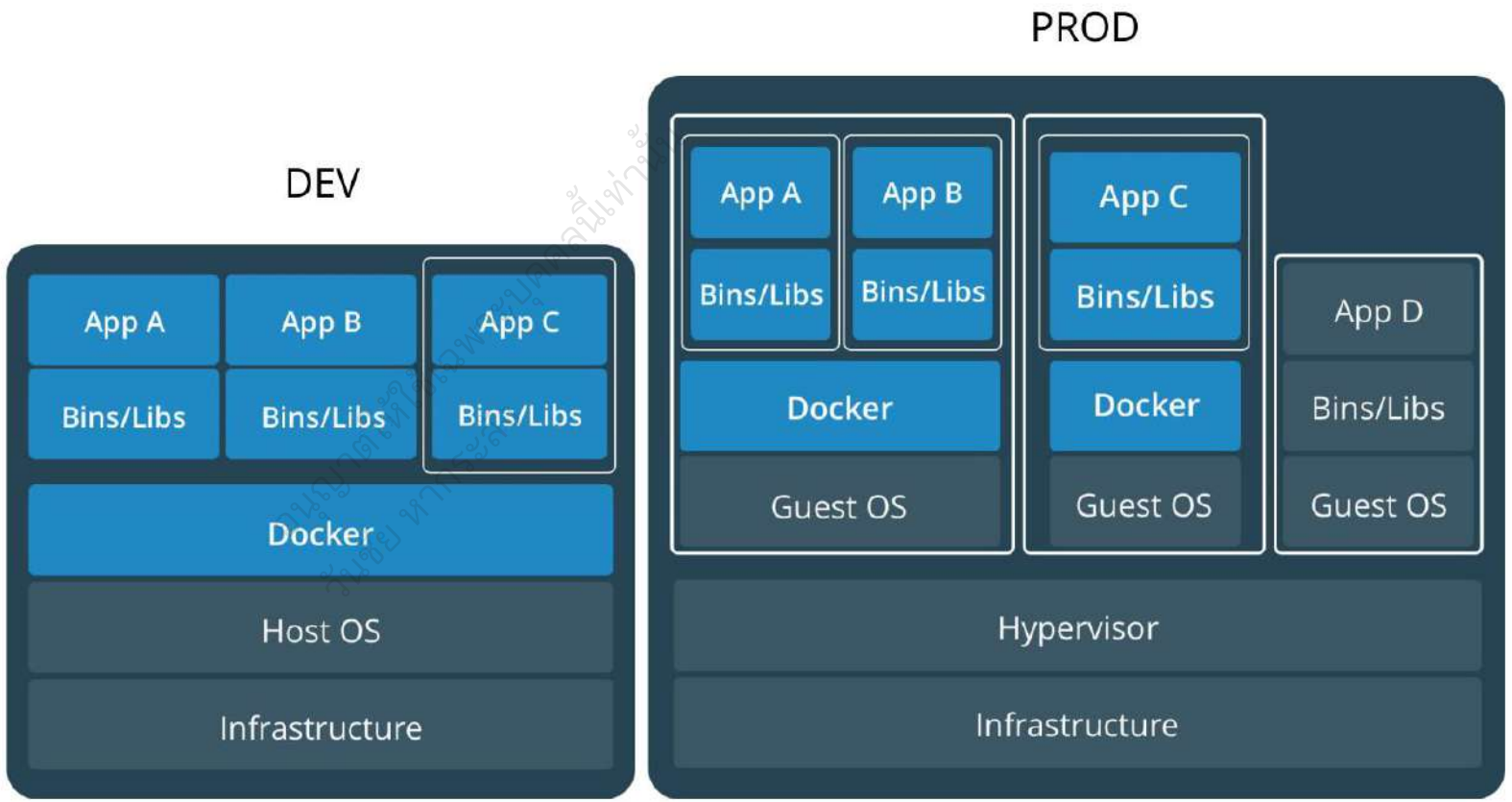


VM are an infrastructure level construct to turn one machine into many servers



Containers are an app level construct

VM and Container together

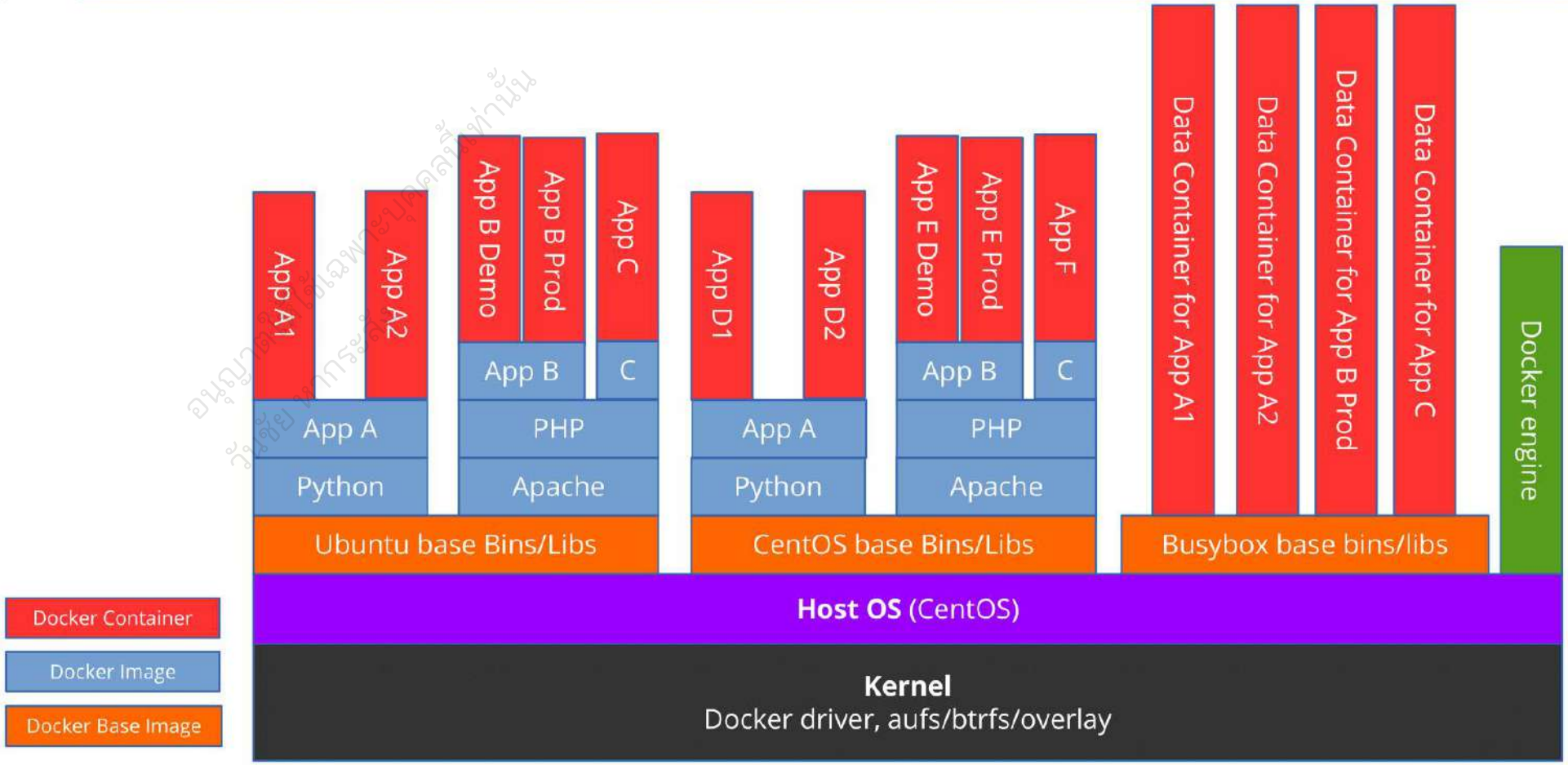


Where Docker can run

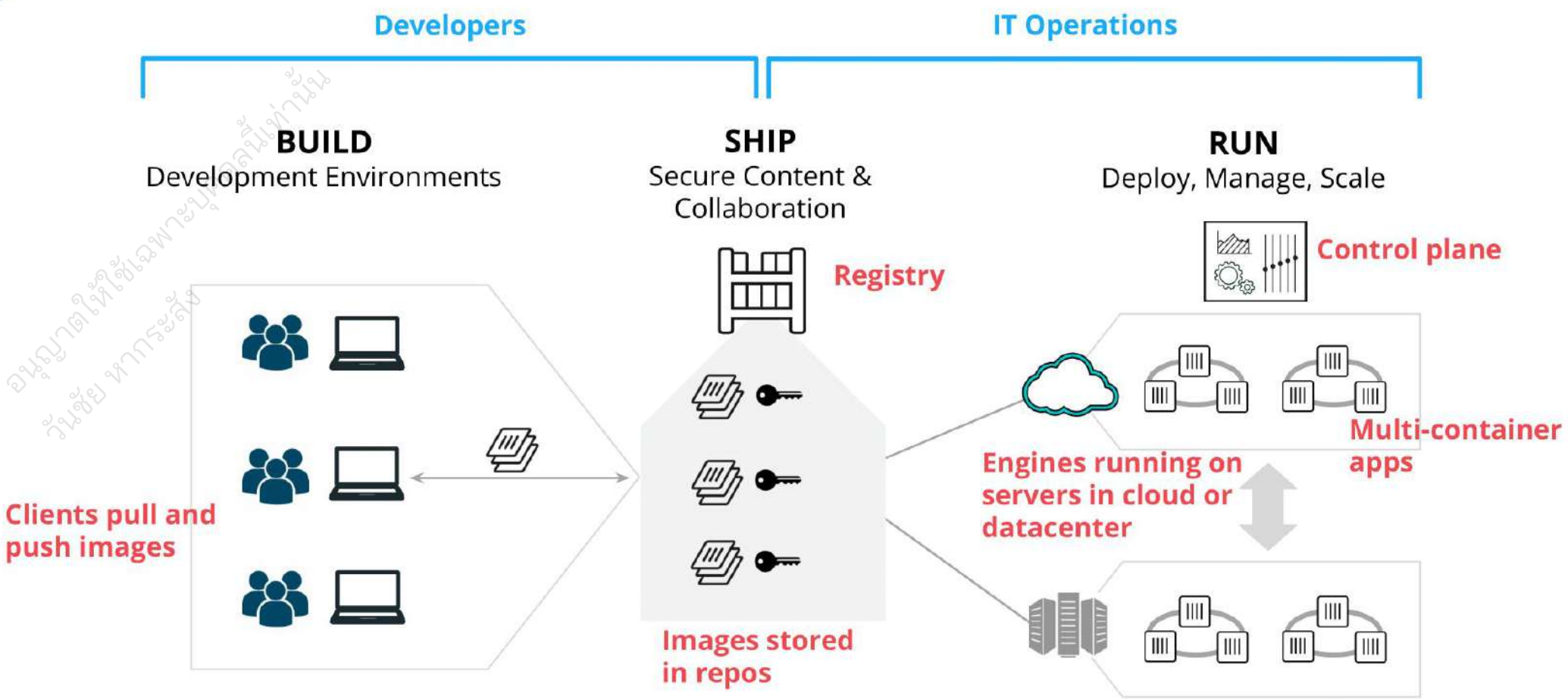
Native

VM

Docker Layer



Build, Ship, Run



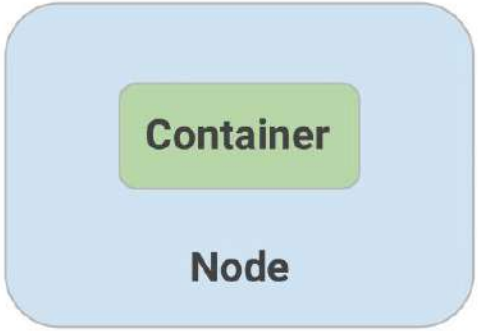
Skooldio Opsta

Kubernetes

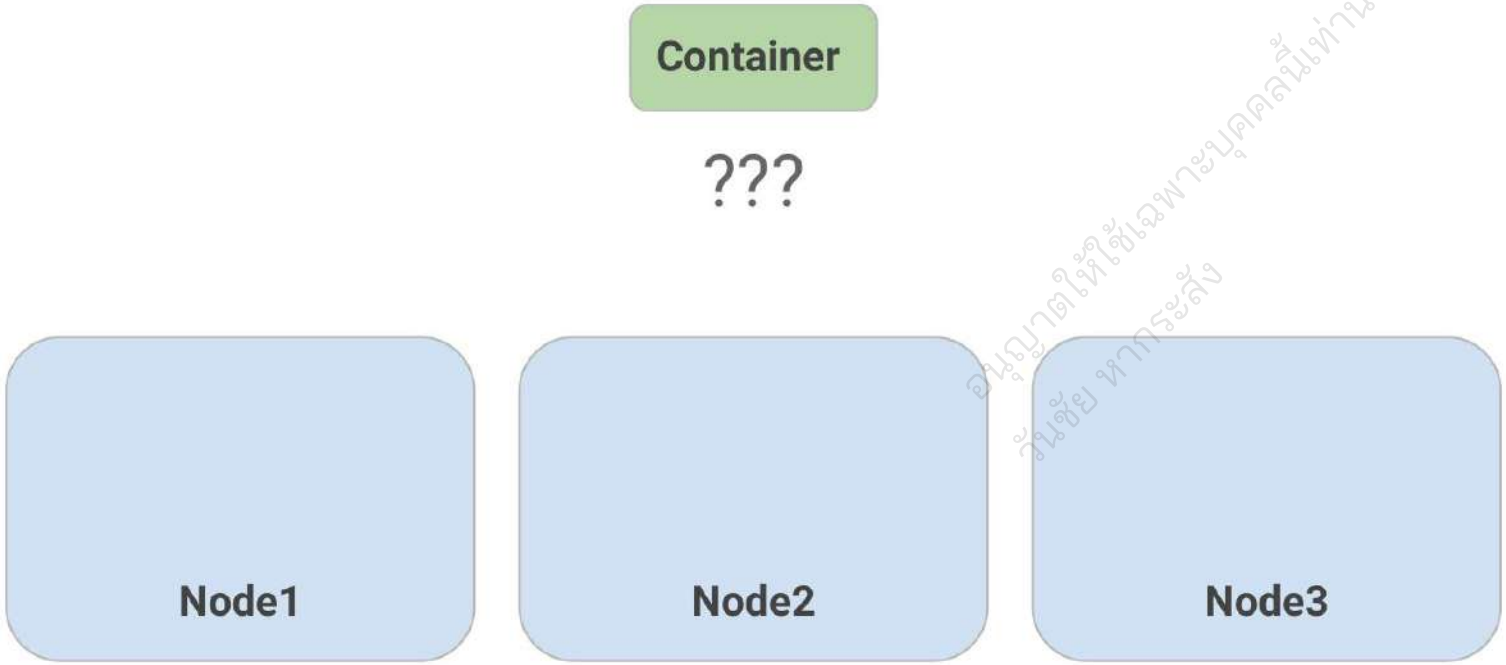
What is?

kubernetes

One Server



Multiple Servers



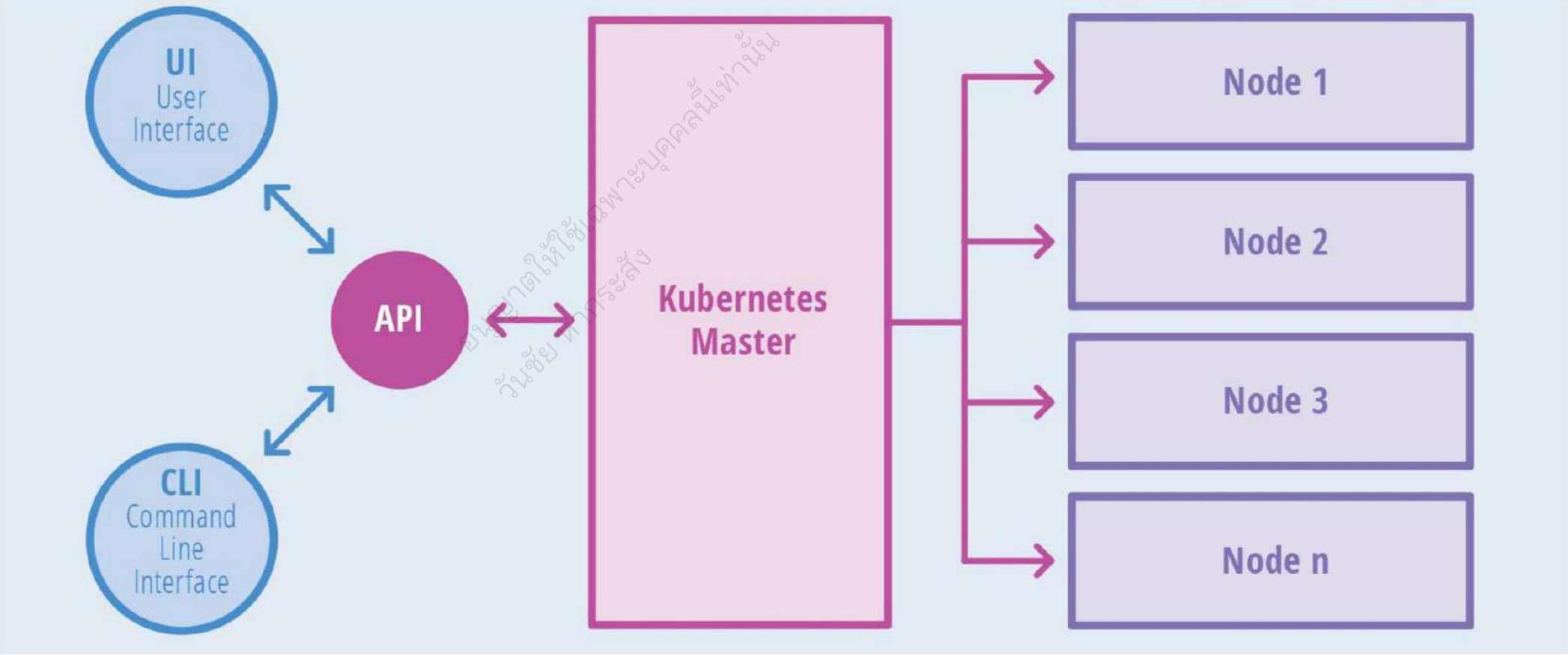
Container Cluster Solution

HashiCorp
Nomad

MESOS

RANCHER®

Kubernetes Architecture



Distribution of Kubernetes

Installation Tools

minikube

KUBESPRAY

kops

K3S

MicroK8s

Commercial

OPENSHIFT

VMware Tanzu

RANCHER

Public Cloud

Azure Container Service

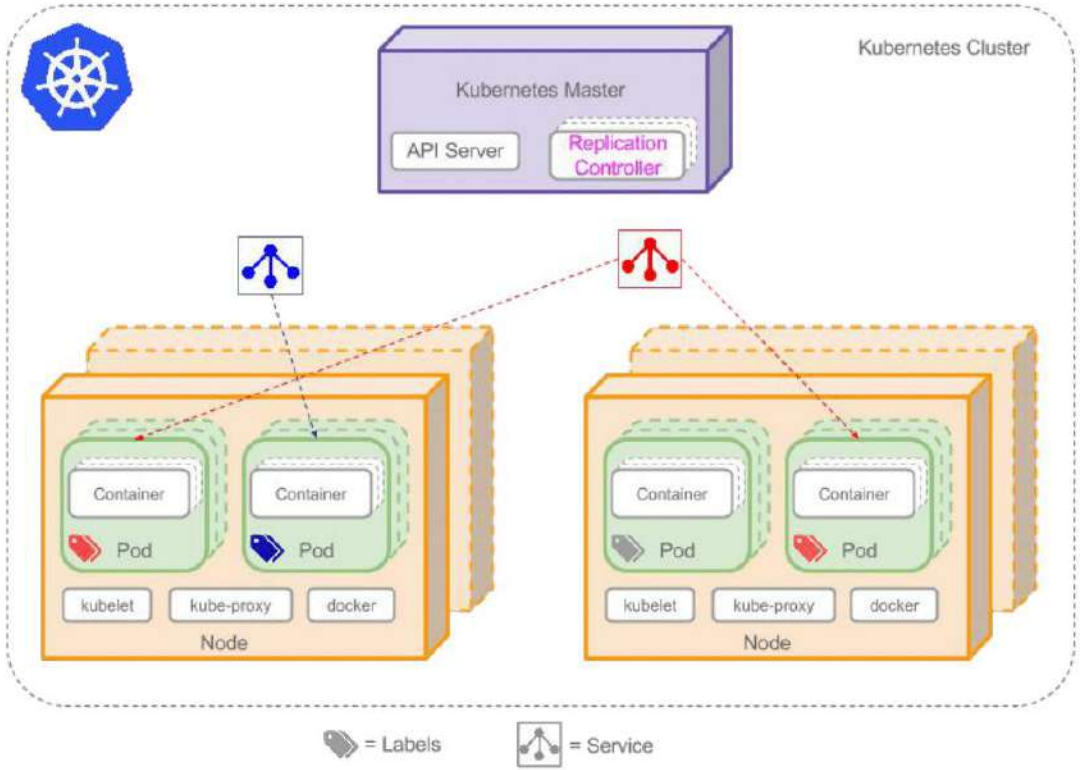
Amazon EKS

IBM Cloud Container Service

Skooldio Opsta

Kubernetes Concepts

- Pods
- Deployments
- Services
- Batch Execution
- Rolling Update
- Storage Orchestration
- Service Discovery
- Scheduling
- Autoscaling
- Secret and Configuration Management



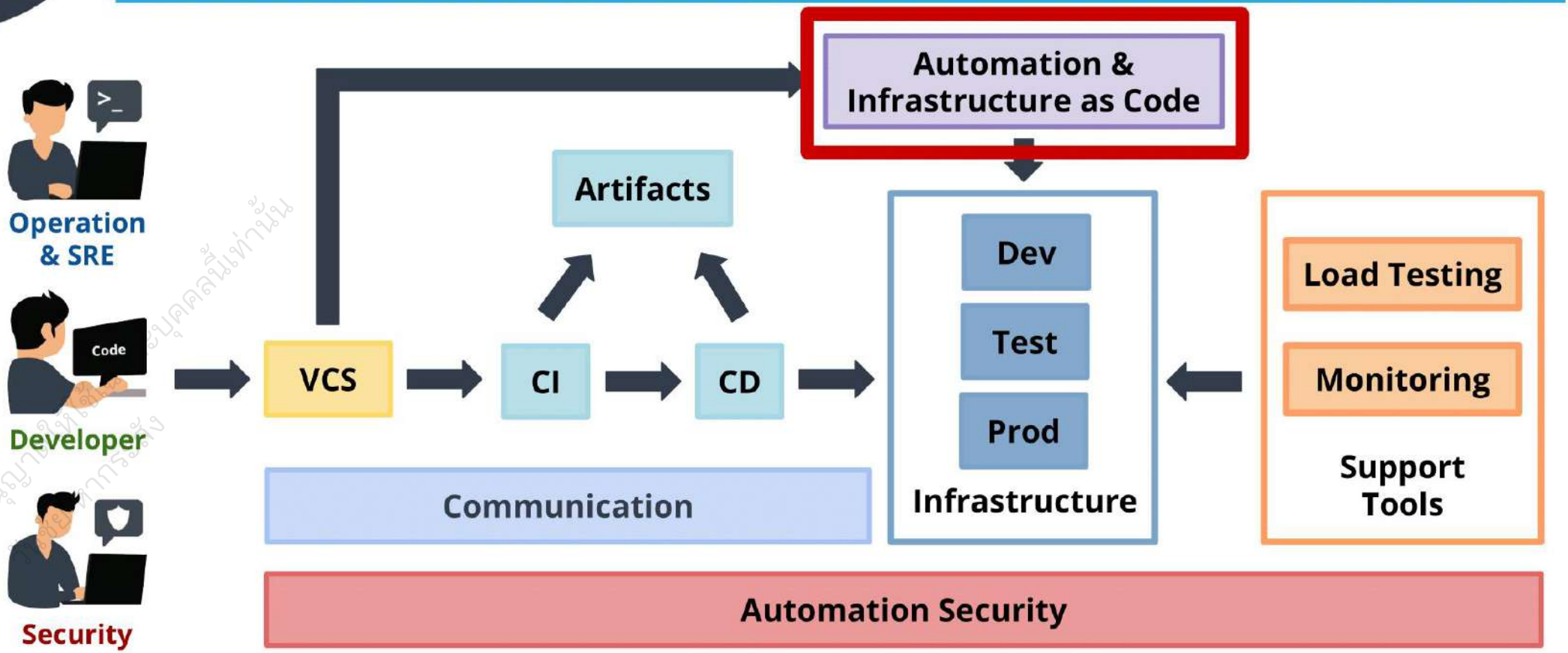
Infrastructure Automation

Technologies

Skooldio

Opsta

Infrastructure Automation



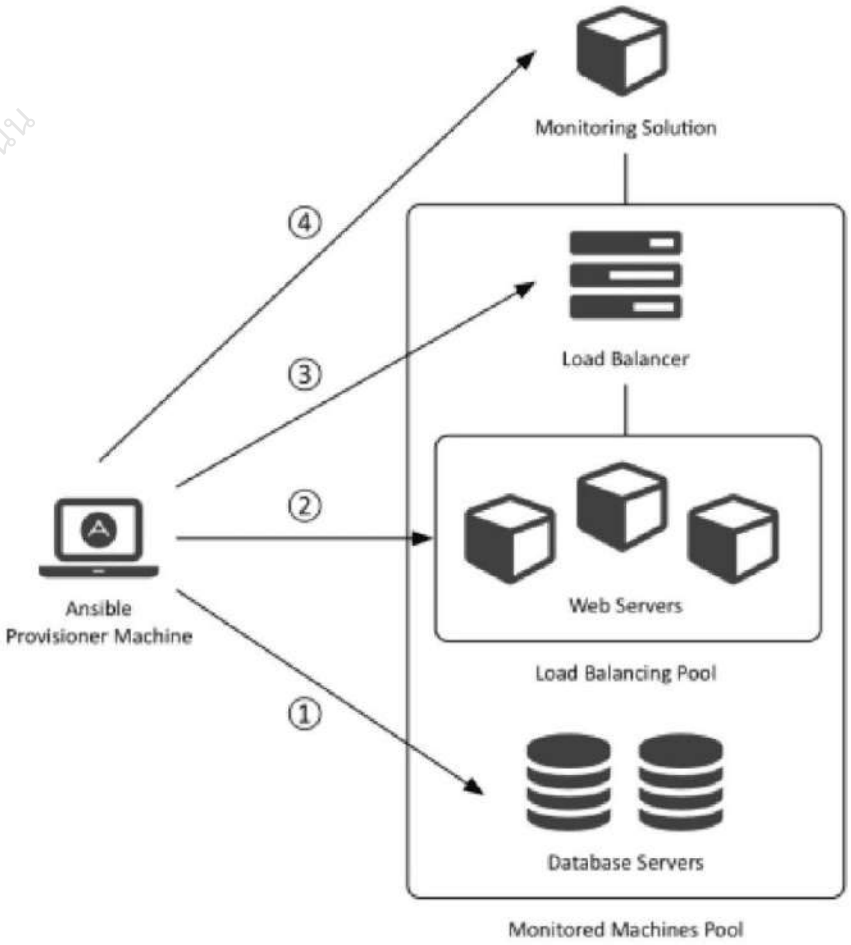
Automation Technologies

Infrastructure as Code

```yaml
---
service_name: nginx-opsta
os_instance:
  image: xenial-server-cloudimg-amd64
  key_name: dear
  flavor: m1.small
  network: opsta-network
  security_groups: default
  volume_size: 16
  project_name: opsta
  count: 3

docker_image:
  nginx:
    image: nginx:1.11.3-alpine
    name: nginx
    hostname: nginx
    published_ports:
      - "80:80"
      - "443:443"
```

Ansible Orchestration

Ansible is an open-source automation tool, or platform, used for IT tasks such as configuration management, application deployment, intraservice orchestration, and provisioning



Ansible Tasks

```
- name: ensure apache is at the latest version
  yum: name=httpd state=latest

- name: ensure apache is running (and enable it at boot)
  service: name=httpd state=started enabled=yes

- name: configure hosts file
  lineinfile:
    dest: /etc/hosts
    state: present
    line: "127.0.0.1 prod-01.example.com"
```

Ansible Galaxy Roles

geerlingguy

filebeat

Filebeat for Linux.

elasticsearch filebeat logging monitoring system web

build passing

4.2 / 5 Score 729699 Downloads

Last Imported: 3 days ago

geerlingguy

homebrew

Homebrew for macOS

brew cask development homebrew packaging system

build passing

5 / 5 Score 621702 Downloads

Last Imported: 6 days ago

dev-sec

os-hardening

This role provides numerous security-related configurations, providing all-round base protection.

hardening security system

build failing

4.9 / 5 Score 173835 Downloads

Last Imported: 3 days ago

geerlingguy

firewall

Simple iptables firewall for most Unix-like systems.

firewall iptables networking security system tcp

build passing

5 / 5 Score 113093 Downloads

Last Imported: 2 days ago

willshersystems

sshd

OpenSSH SSH daemon configuration

aix centos debian freebsd networking openbsd openssh redhat server ssh sshd system ubuntu

build passing

4.8 / 5 Score 225555 Downloads

Last Imported: 2 days ago

geerlingguy

elasticsearch

Elasticsearch for Linux.

efk elasticsearch elk logging lucene monitoring system web

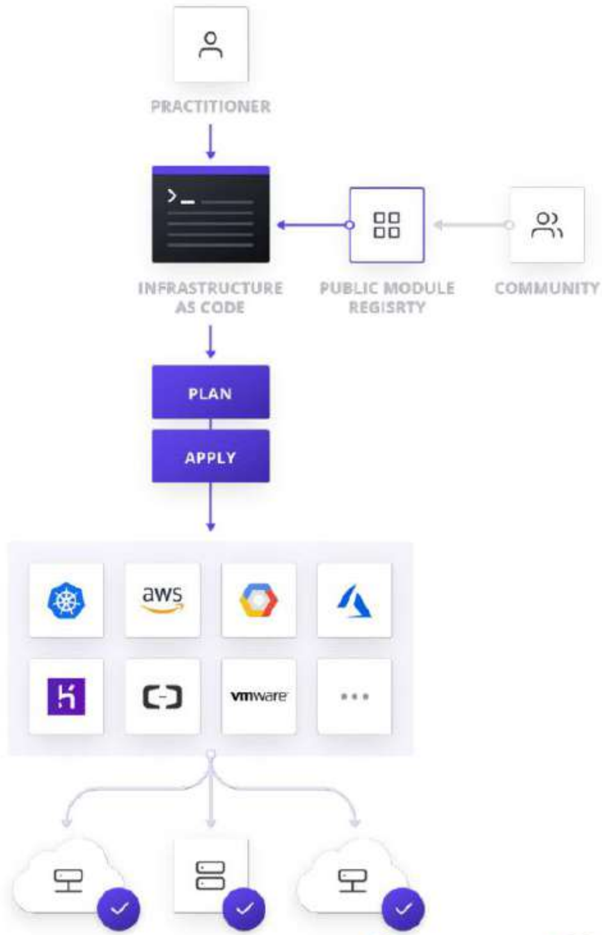
build passing

5 / 5 Score 302129 Downloads

Last Imported: 21 minutes ago

Terraform

Terraform use Infrastructure as Code to provision and manage any cloud, infrastructure, or service



HashiCorp Terraform

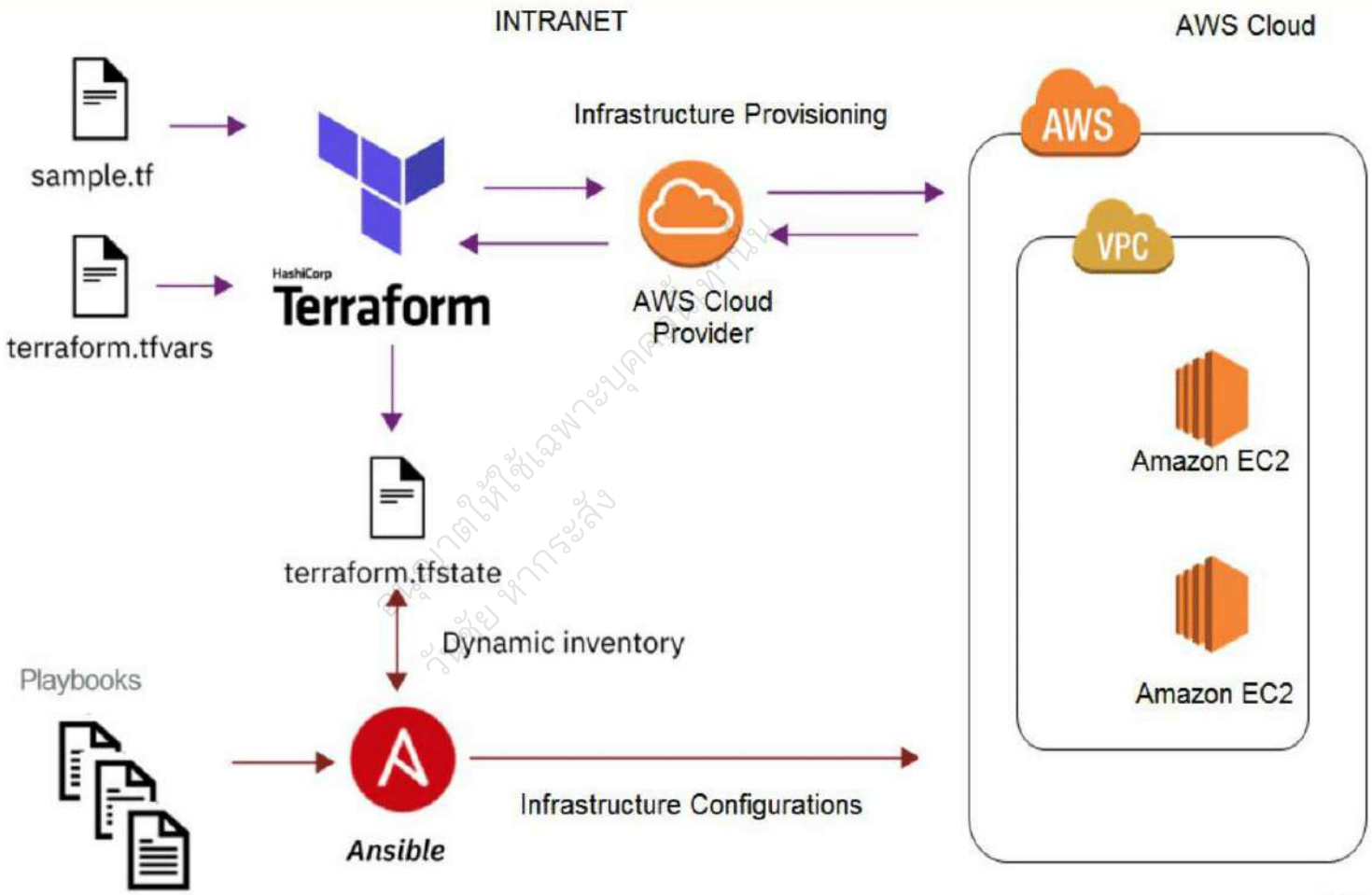
Terraform IaC

```
provider "aws" {
  profile    = "default"
  region     = "us-east-1"
}

resource "aws_instance" "example" {
  ami           = "ami-b374d5a5"
  instance_type = "t2.micro"
}
```

Terraform Plan

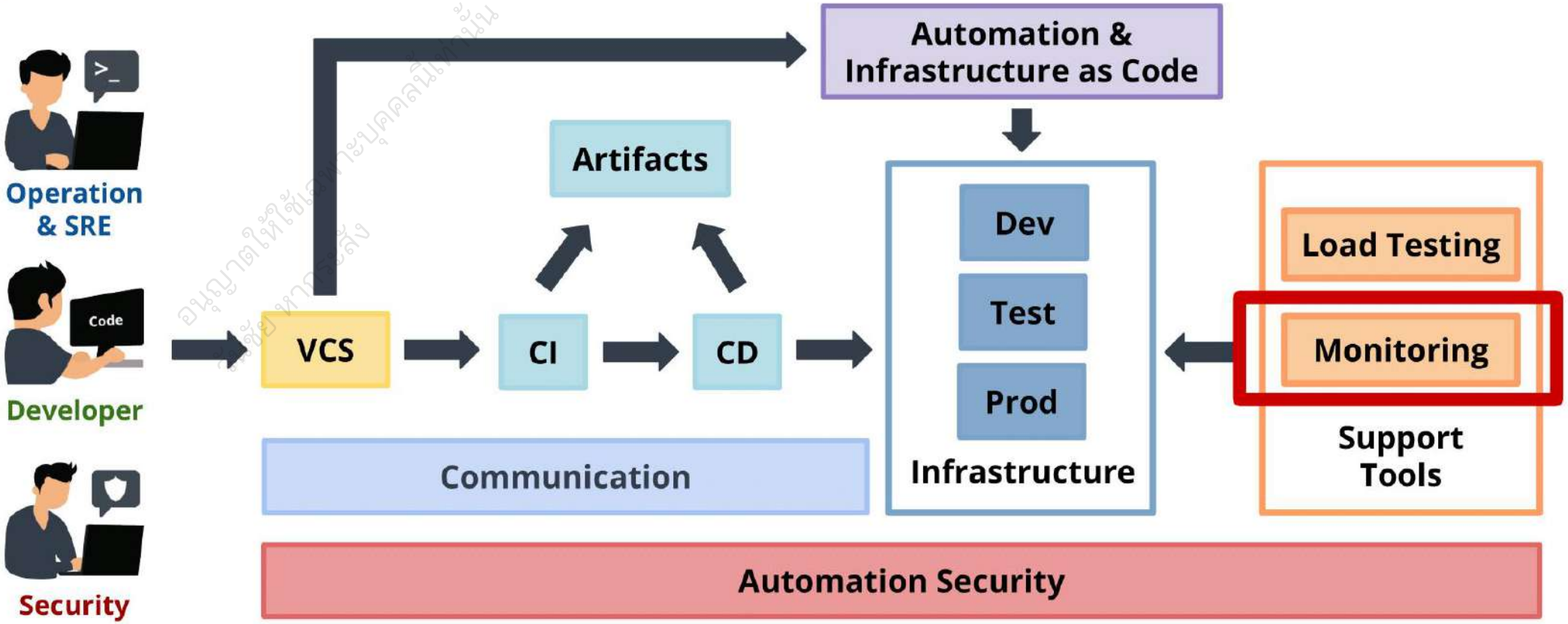
```
$ terraform apply
aws_instance.example: Refreshing state... [id=i-0bbf06244e44211d1]

An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
-/+ destroy and then create replacement

Terraform will perform the following actions:

  # aws_instance.example must be replaced
-/+ resource "aws_instance" "example" {
      ~ ami                          = "ami-2757f631" -> "ami-b374d5a5" # forces re
      ~ arn                          = "arn:aws:ec2:us-east-1:130490850807:instance
      ~ associate_public_ip_address  = true -> (known after apply)
      ~ availability_zone            = "us-east-1c" -> (known after apply)
      ~ cpu_core_count               = 1 -> (known after apply)
      ~ cpu_threads_per_core         = 1 -> (known after apply)
      - disable_api_termination      = false -> null
      - ebs_optimized                = false -> null
        get_password_data            = false
      + host_id                      = (known after apply)
      ~ id                           = "i-0bbf06244e44211d1" -> (known after apply)
      ~ instance_state               = "running" -> (known after apply)
        instance_type                = "t2.micro"
```

Terraform + Ansible

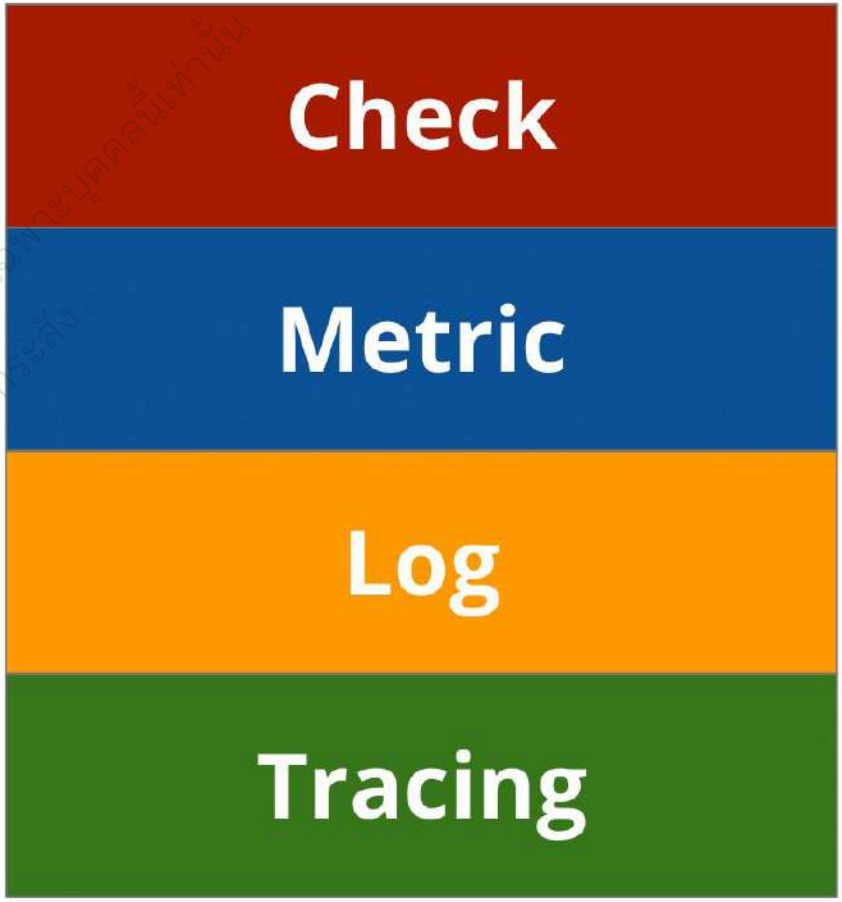


Monitoring

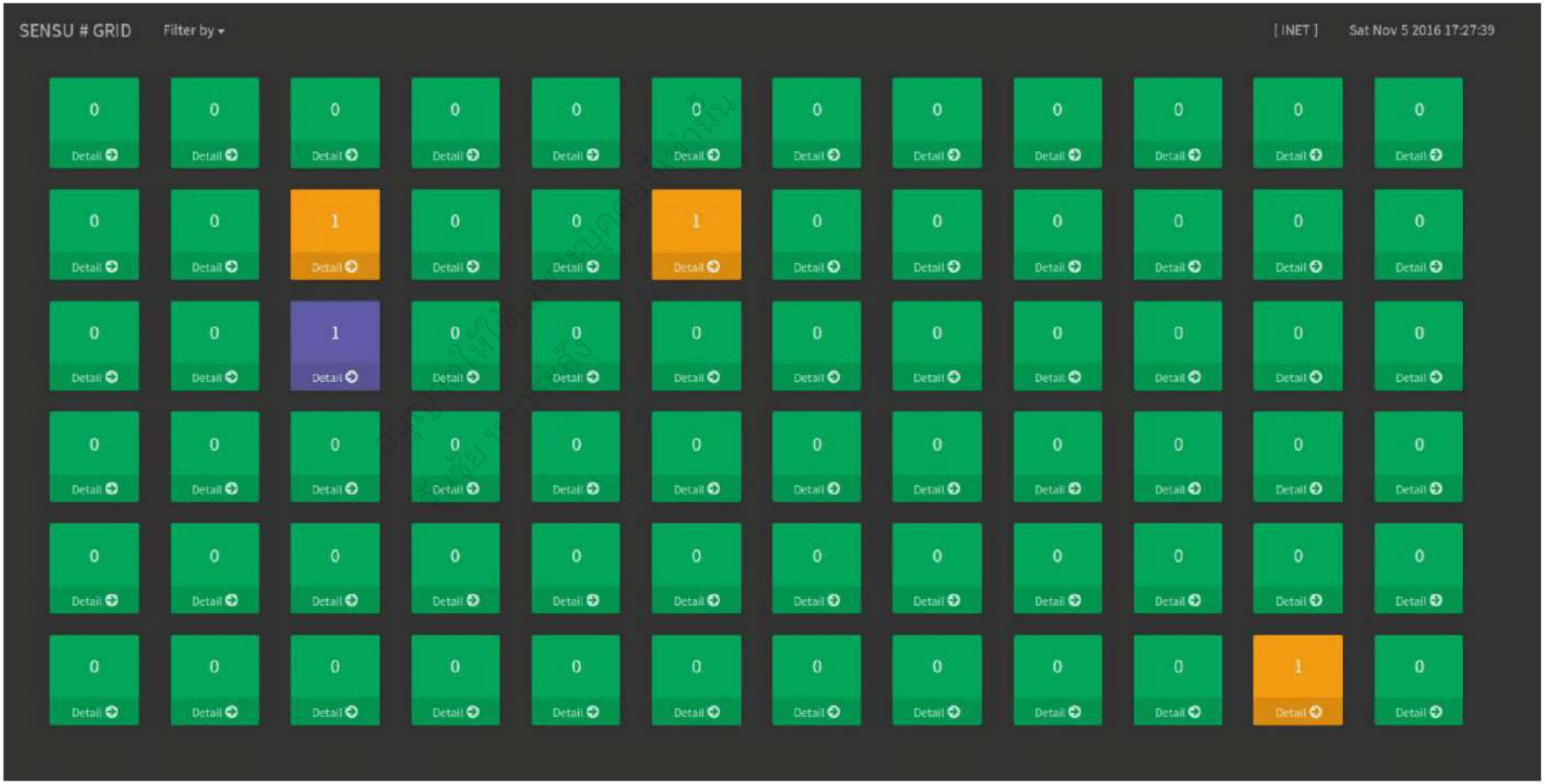
Monitoring



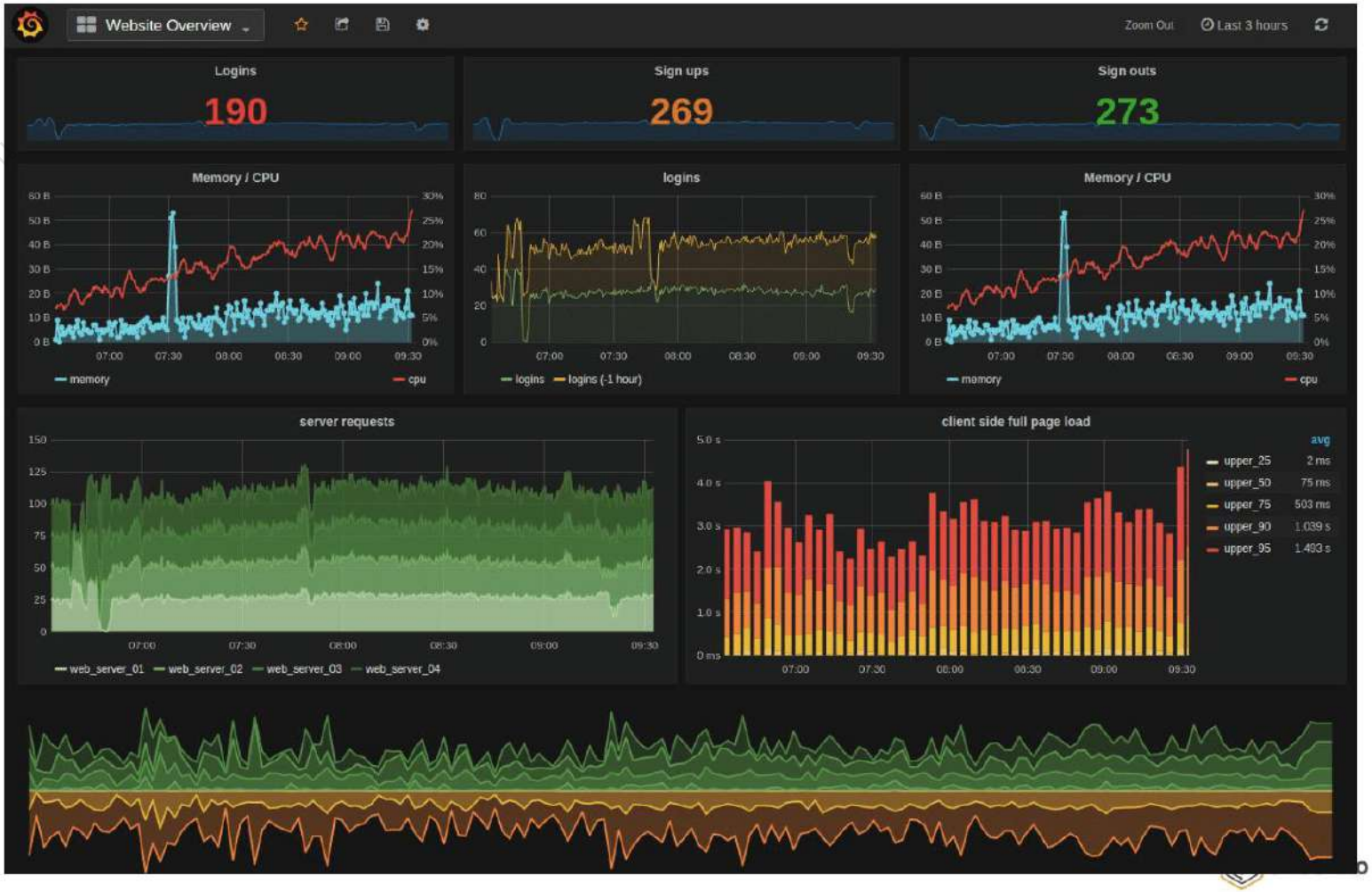
Type of Monitoring Data



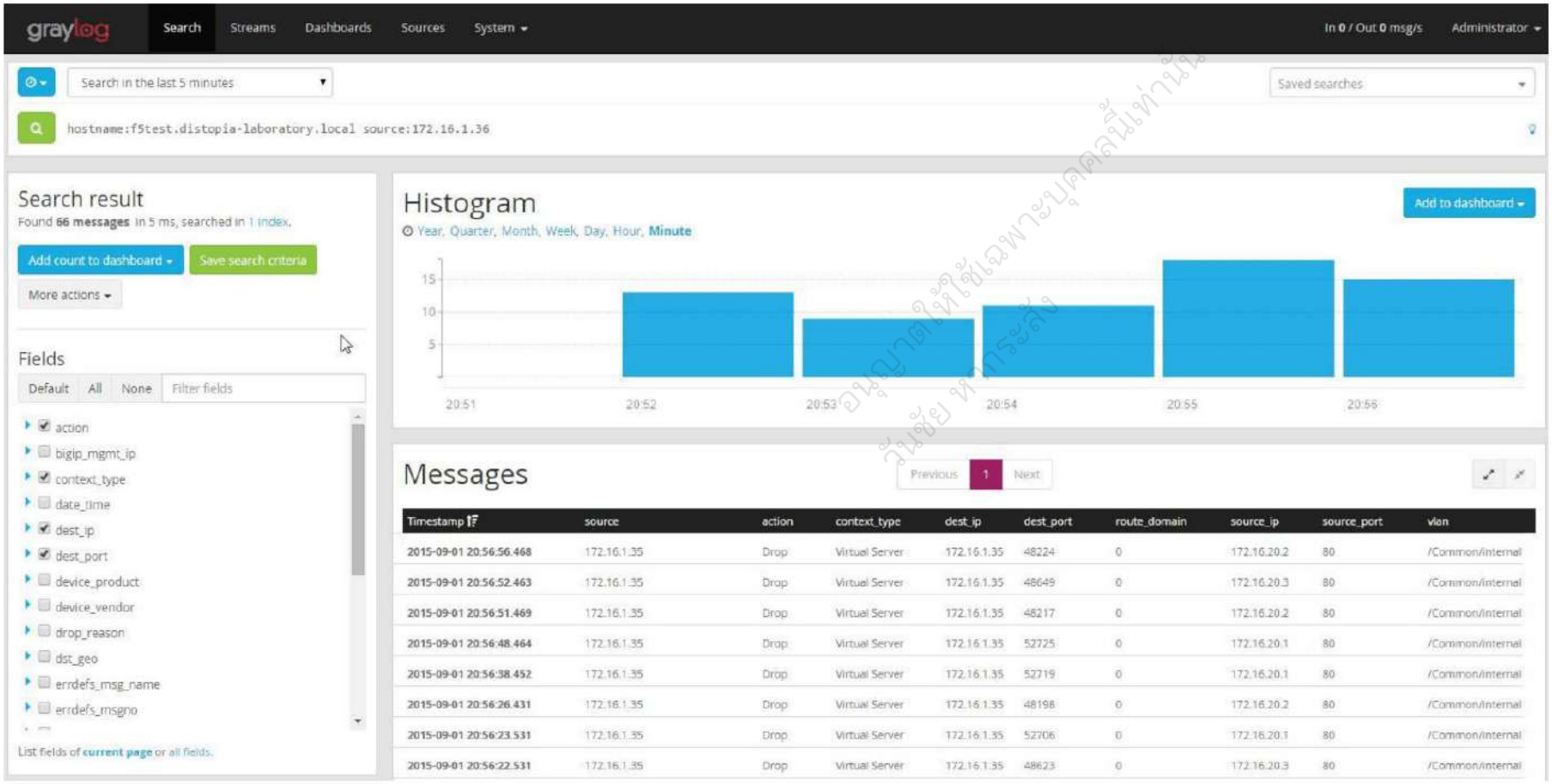
Check



Metric

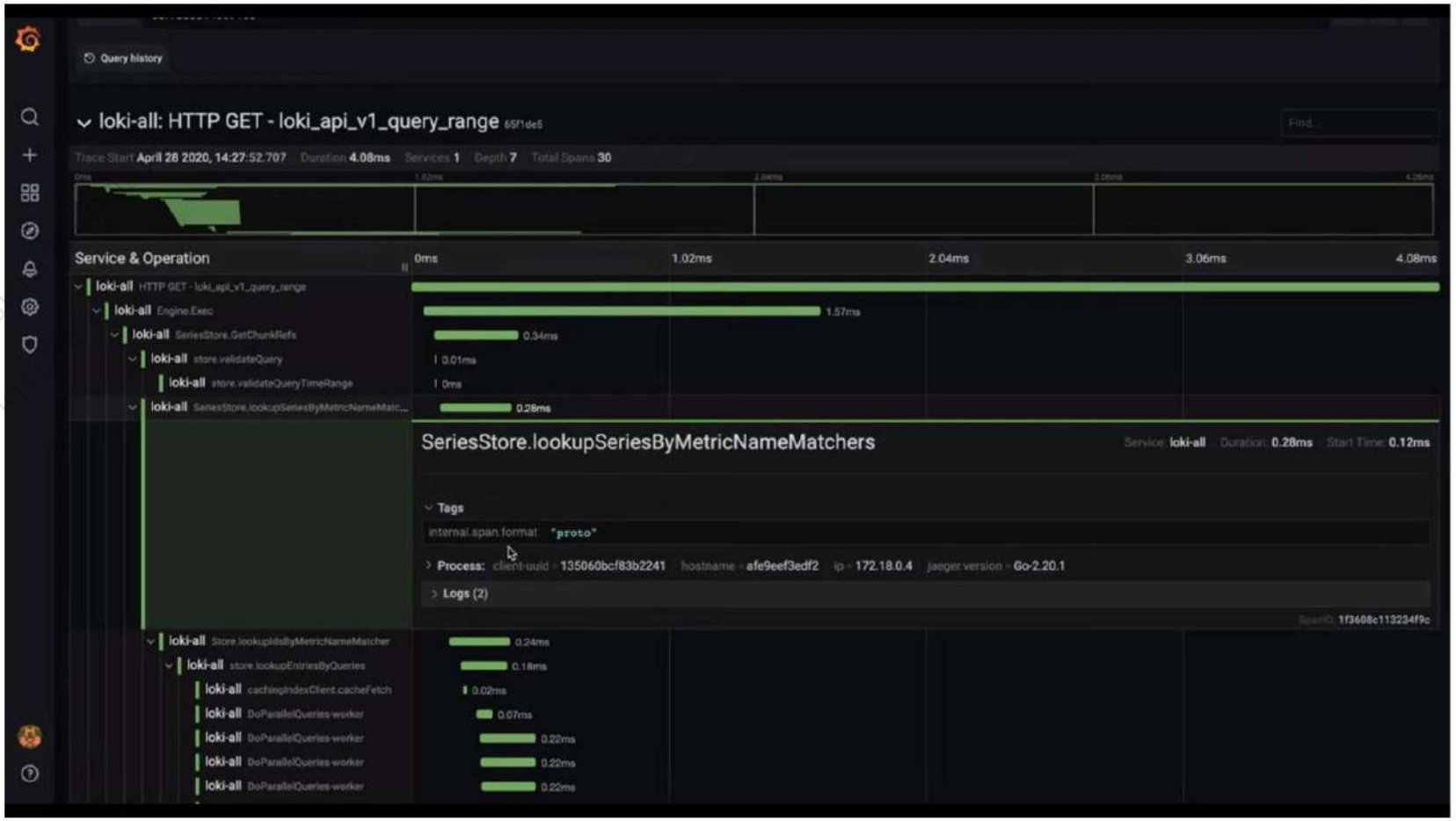


Log

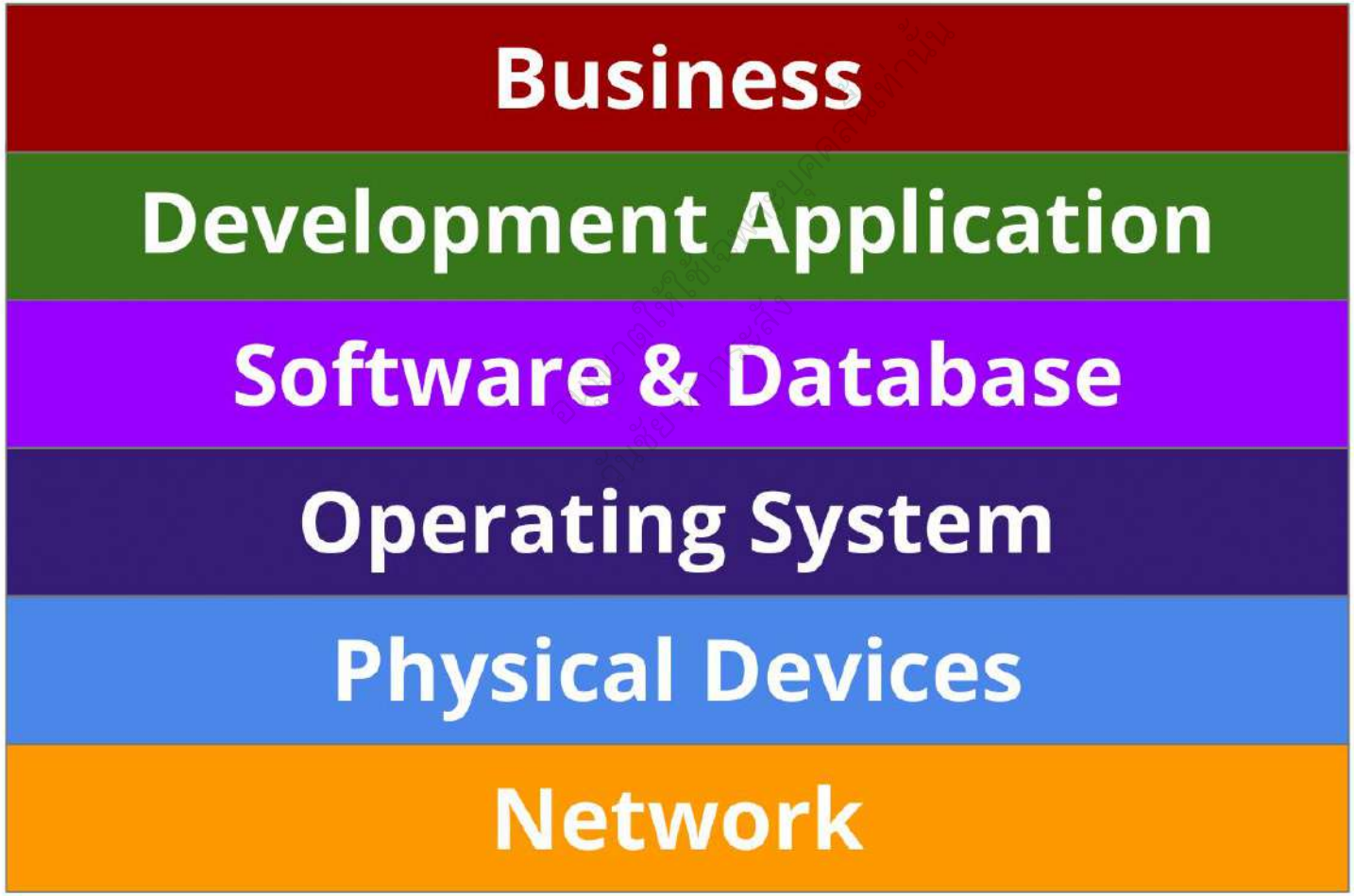


Timestamp	source	action	context_type	dest_ip	dest_port	route_domain	source_ip	source_port	vlan
2015-09-01 20:56:56.468	172.16.1.35	Drop	Virtual Server	172.16.1.35	48224	0	172.16.20.2	80	/Common/internal
2015-09-01 20:56:52.463	172.16.1.35	Drop	Virtual Server	172.16.1.35	48649	0	172.16.20.3	80	/Common/internal
2015-09-01 20:56:51.469	172.16.1.35	Drop	Virtual Server	172.16.1.35	48217	0	172.16.20.2	80	/Common/internal
2015-09-01 20:56:48.464	172.16.1.35	Drop	Virtual Server	172.16.1.35	52725	0	172.16.20.1	80	/Common/internal
2015-09-01 20:56:38.452	172.16.1.35	Drop	Virtual Server	172.16.1.35	52719	0	172.16.20.1	80	/Common/internal
2015-09-01 20:56:26.431	172.16.1.35	Drop	Virtual Server	172.16.1.35	48198	0	172.16.20.2	80	/Common/internal
2015-09-01 20:56:23.531	172.16.1.35	Drop	Virtual Server	172.16.1.35	52706	0	172.16.20.1	80	/Common/internal
2015-09-01 20:56:22.531	172.16.1.35	Drop	Virtual Server	172.16.1.35	48623	0	172.16.20.3	80	/Common/internal

Tracing



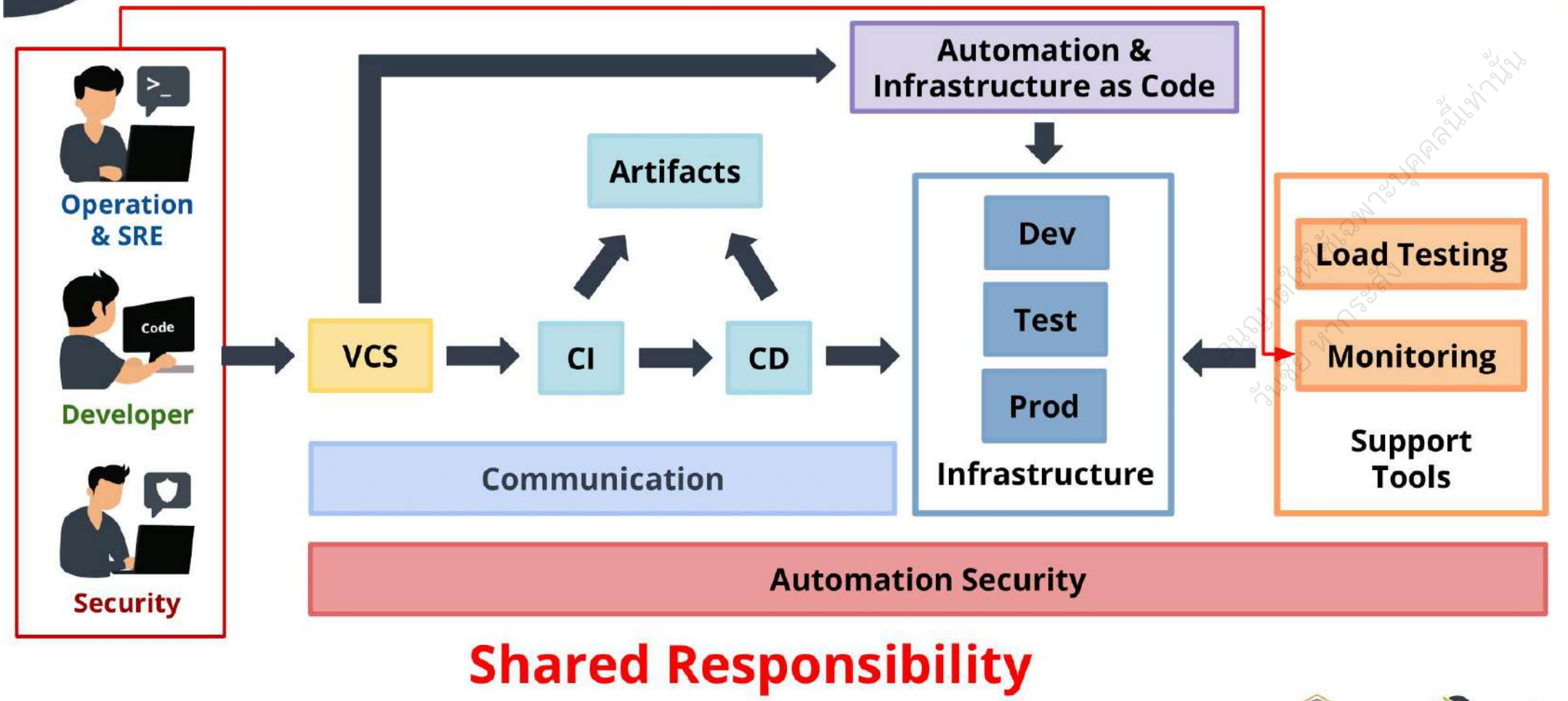
Monitoring Stack



Modern Monitoring Features

- Dashboard
- Alert
- High Availability
- **Scaling**
- **API**
- **Monitor as a Code**
- **Automated**
- **Monitor as Self Service**

Monitoring as Self Service



Shared Responsibility

Modern Monitoring Tools

StatsD

sensu

flapjack

Grafana loki

Grafana

Prometheus

uchiwa

Thanos

graylog

collectd

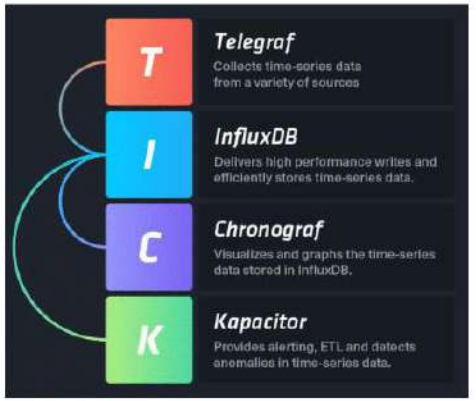
VICTORIA METRICS

elastic

beats

graphite

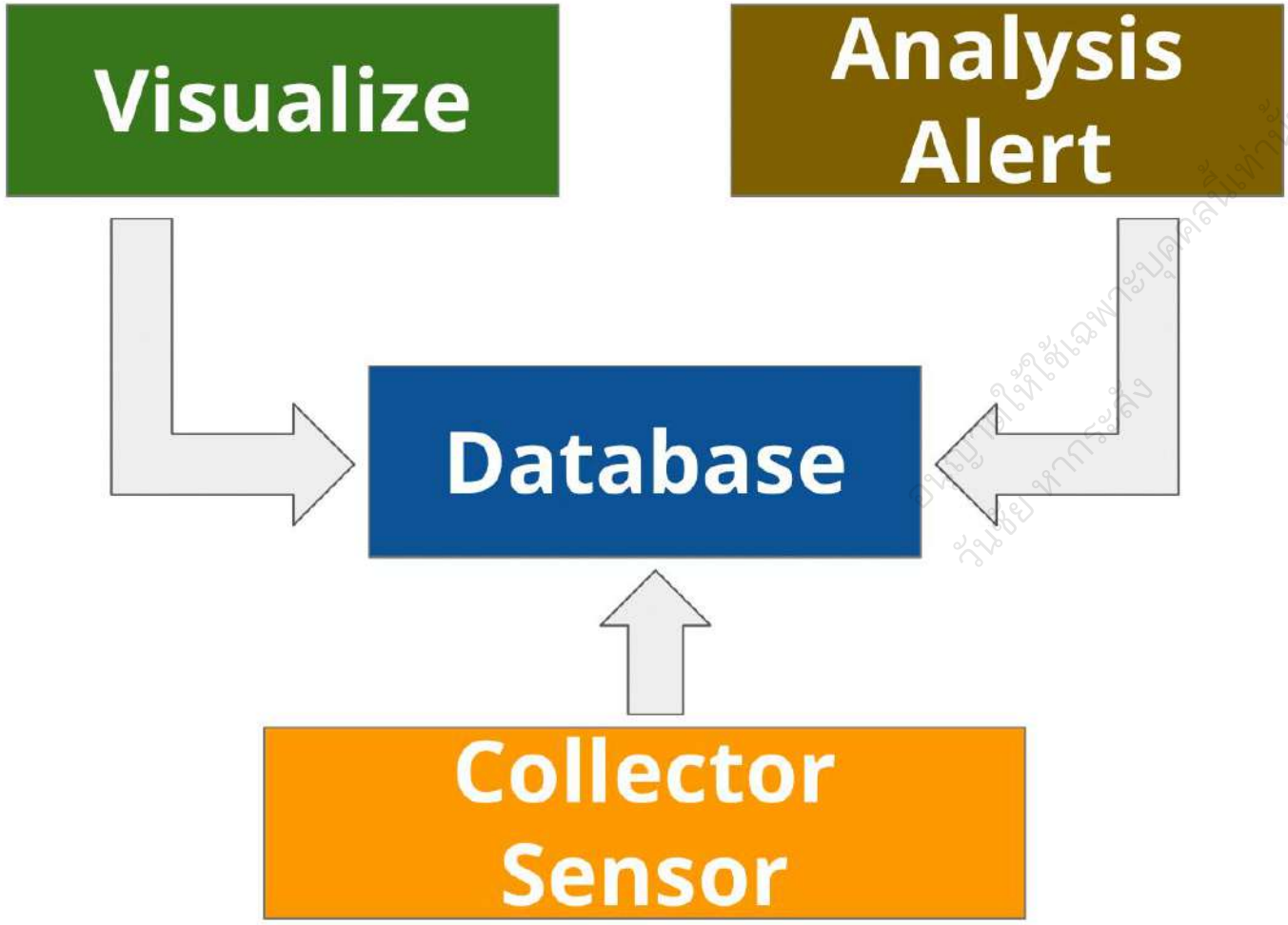
fluentd



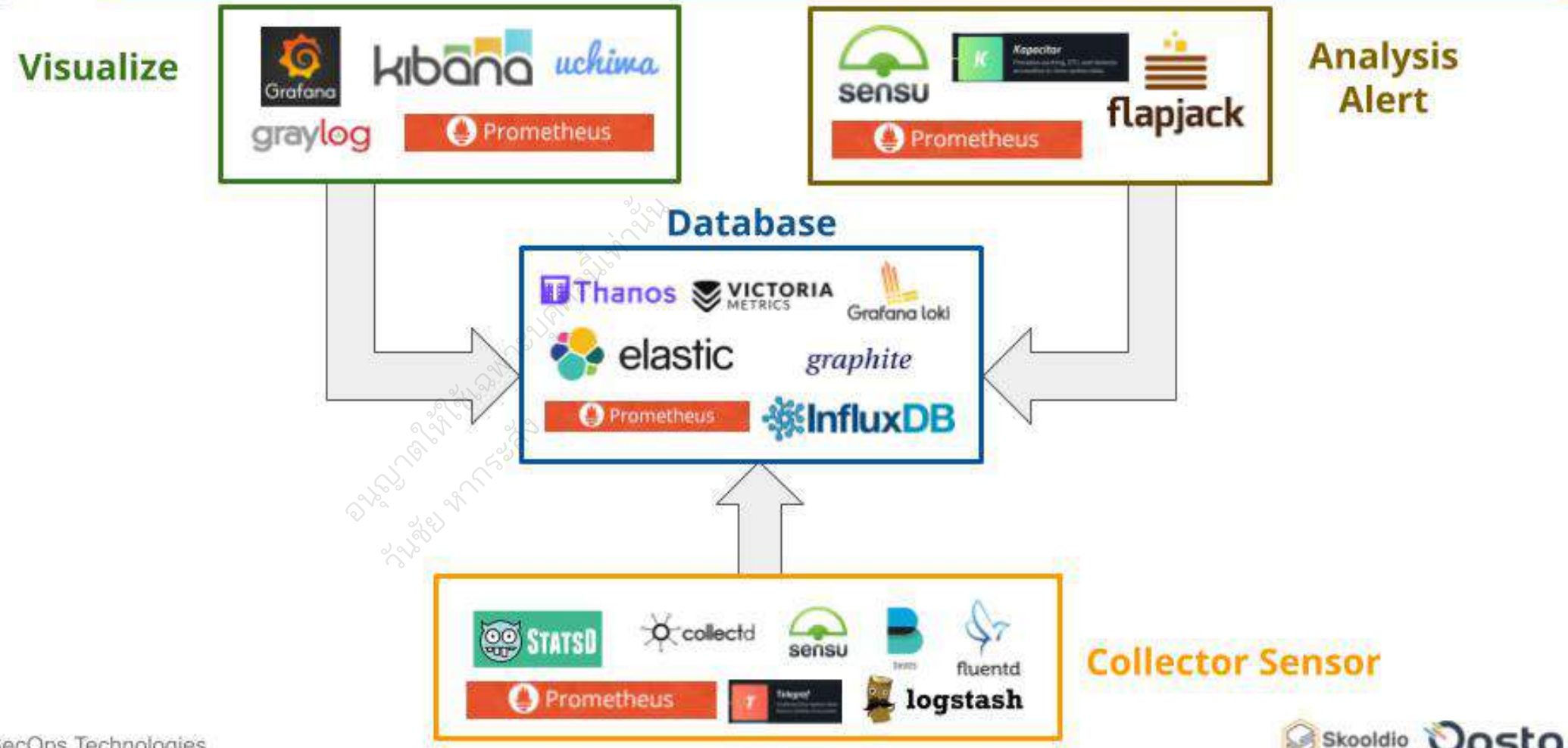
kibana

logstash

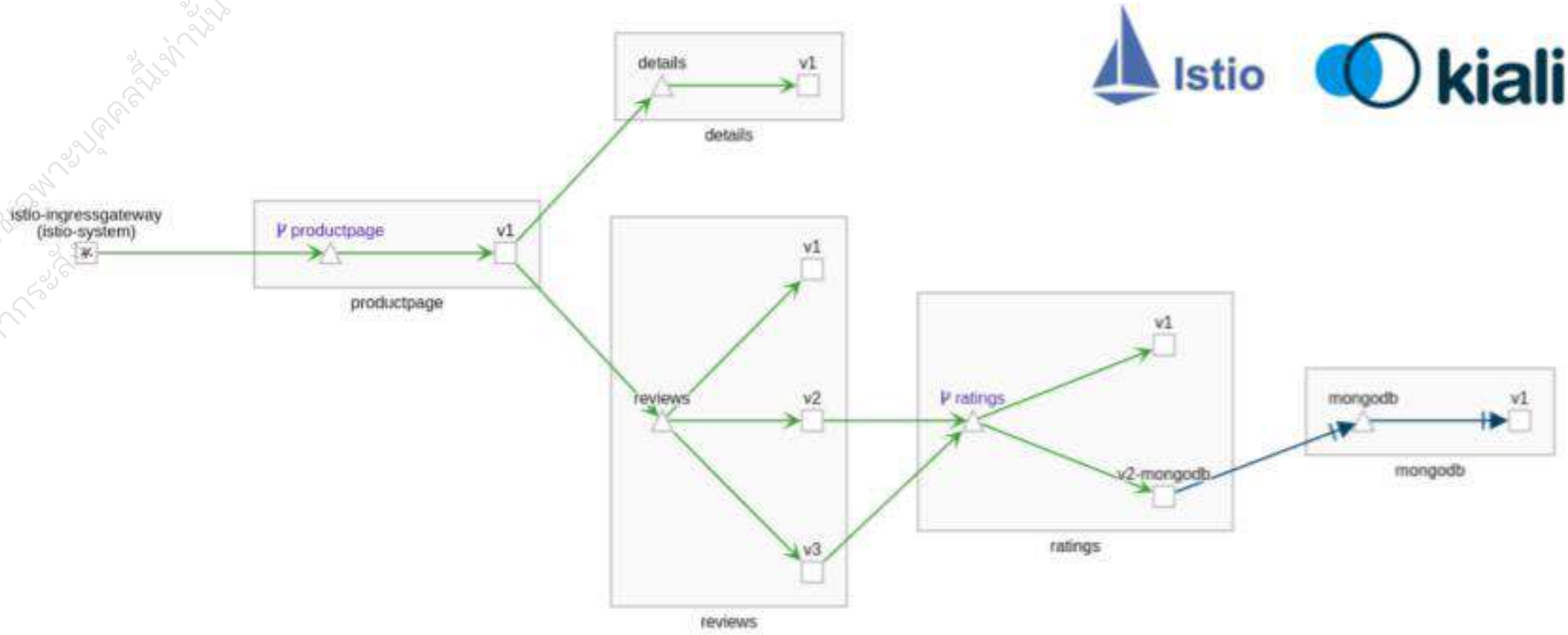
Monitoring Components



Monitoring Components



Application Monitor Tracing



Application Performance Monitoring (APM)

Open Source

Commercial

Skooldio Opsta

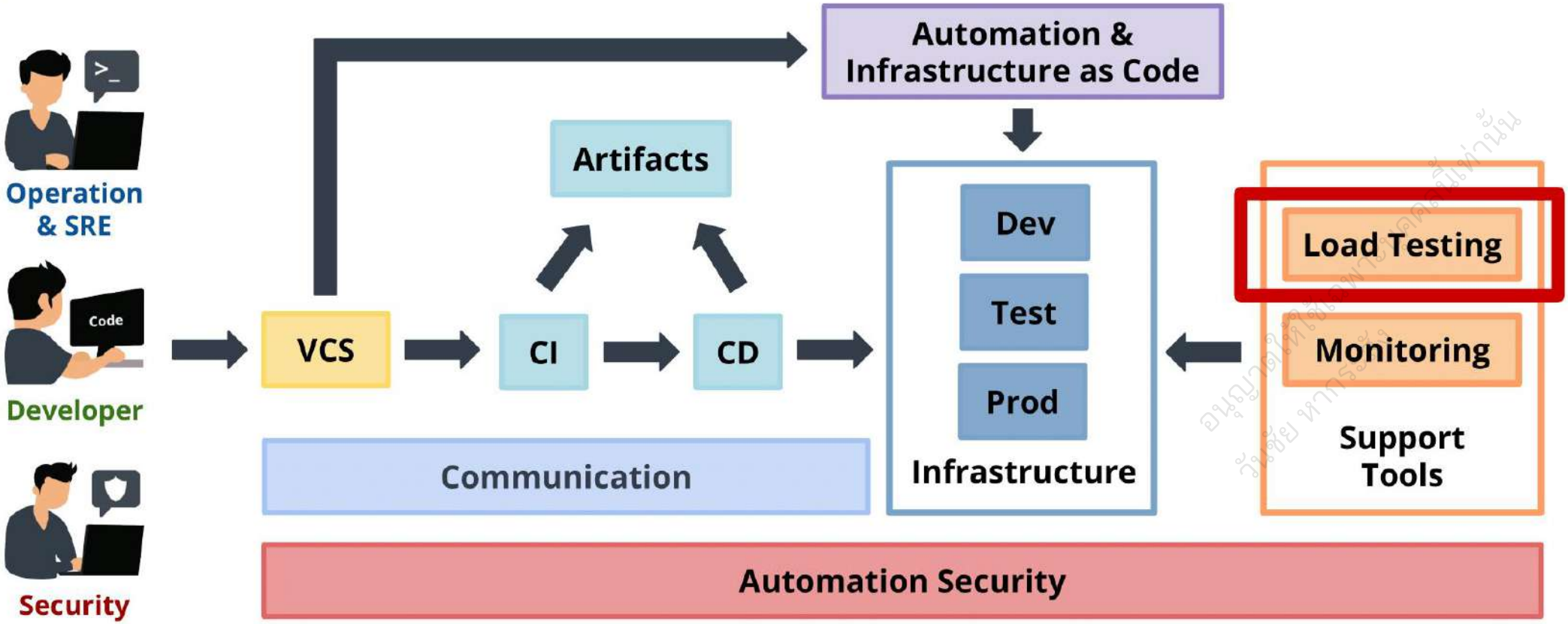
Performance Load Testing

Technologies

Skooldio

Opsta

Performance Load Testing



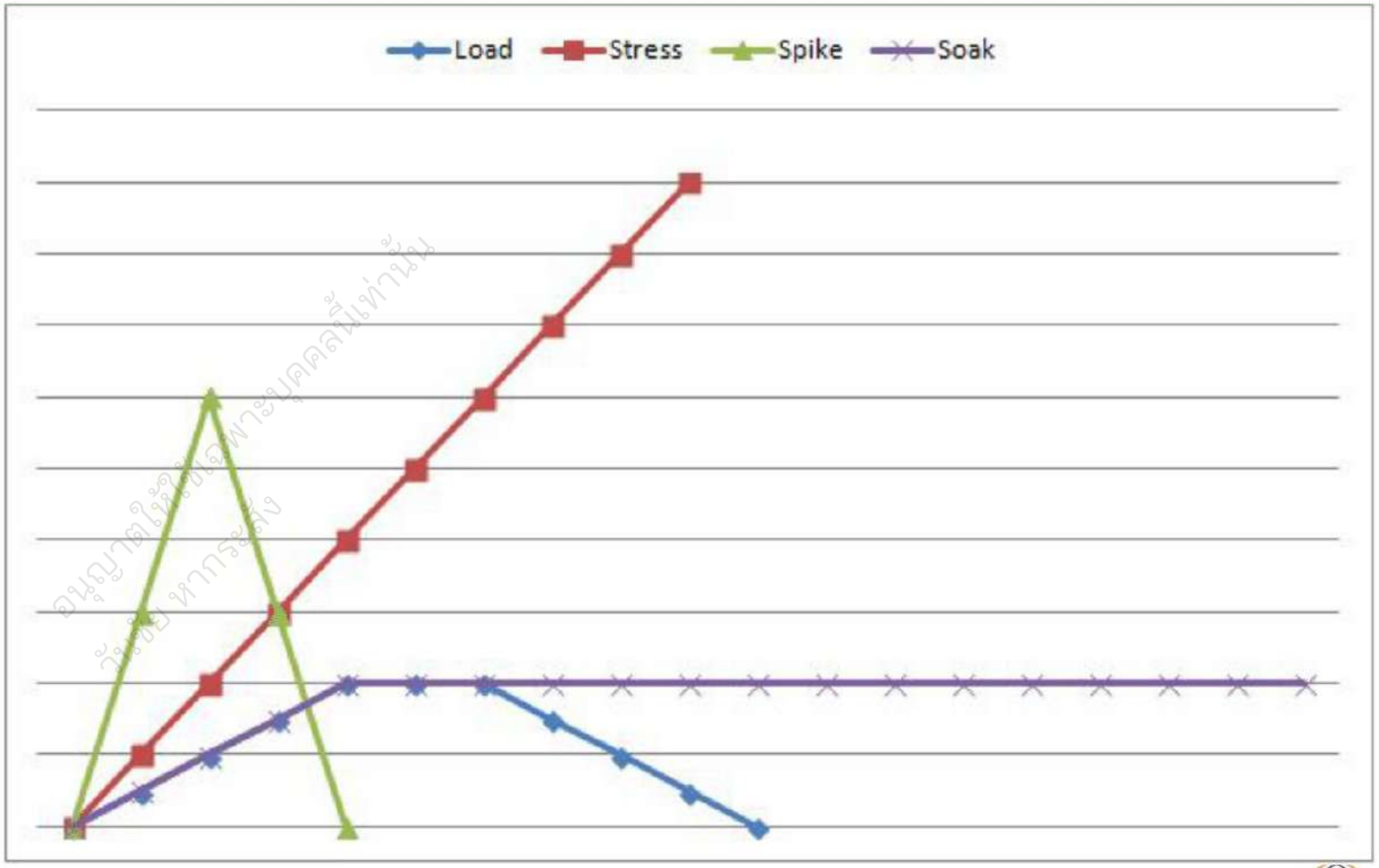
Performance Testing Tools

ApacheBench

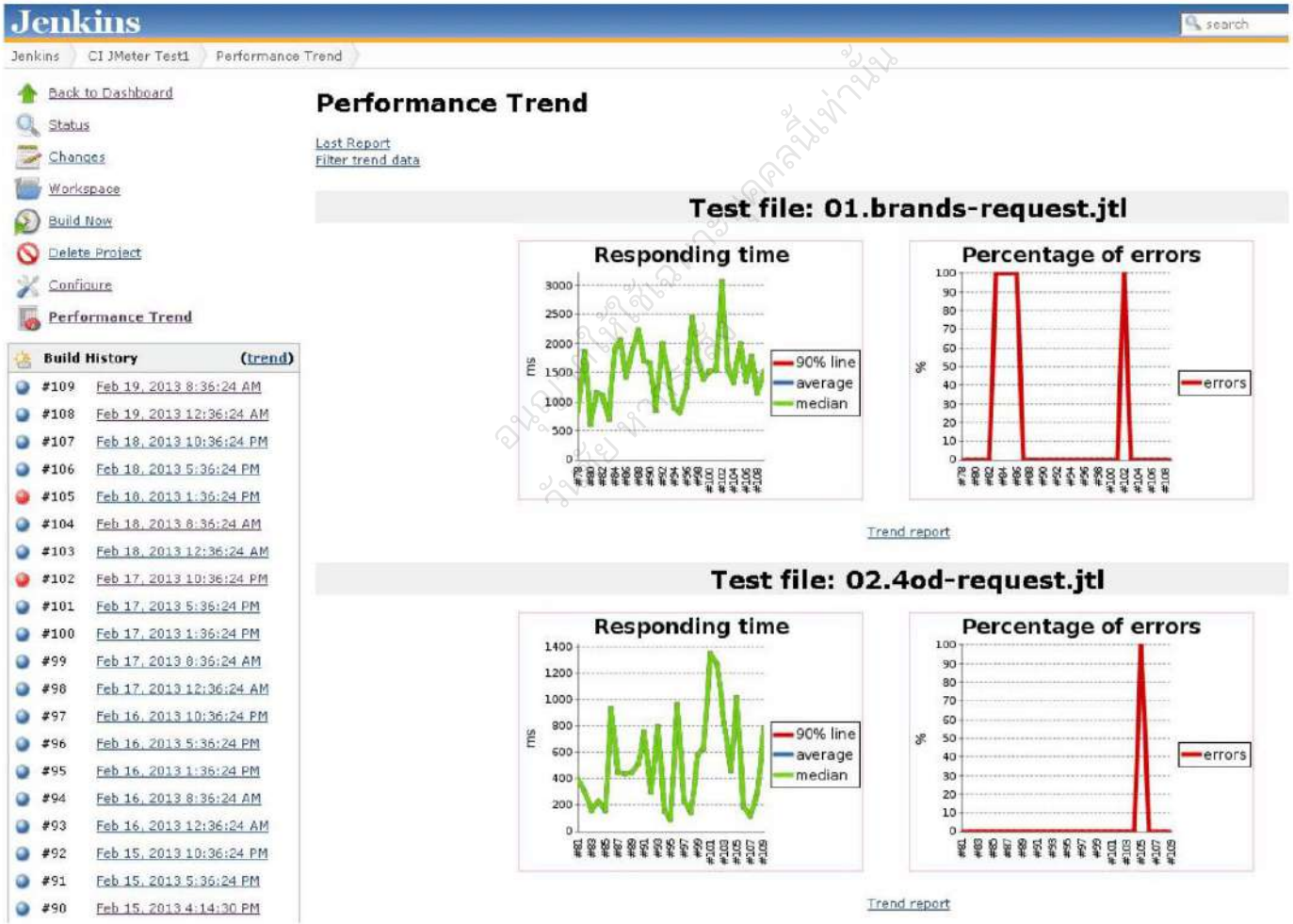
Type of Performance Testing

- Load Testing
- Capacity Testing
- Stress Testing
- Soak Testing
- Spike Testing

Comparison



Automated Performance Testing



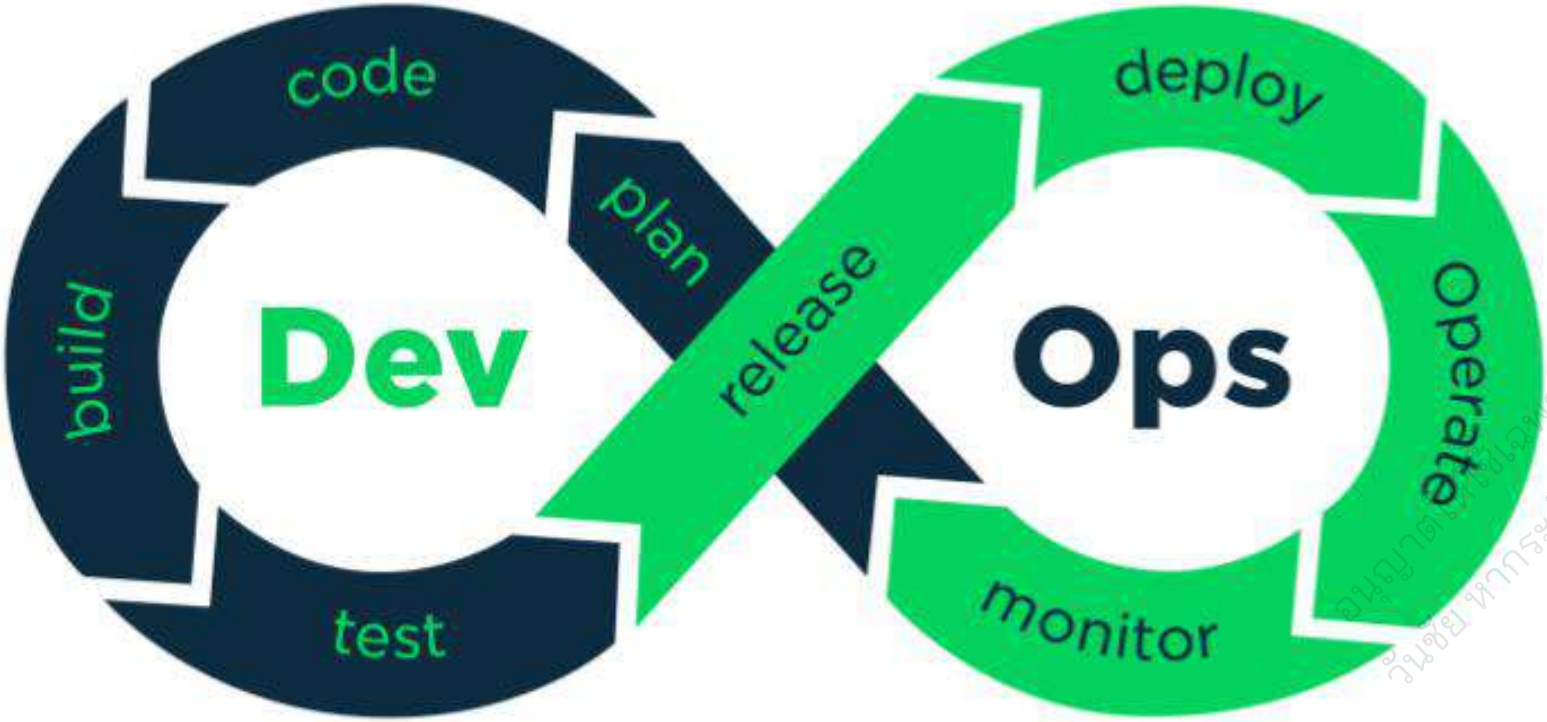
Automation Security

Technologies

Skooldio

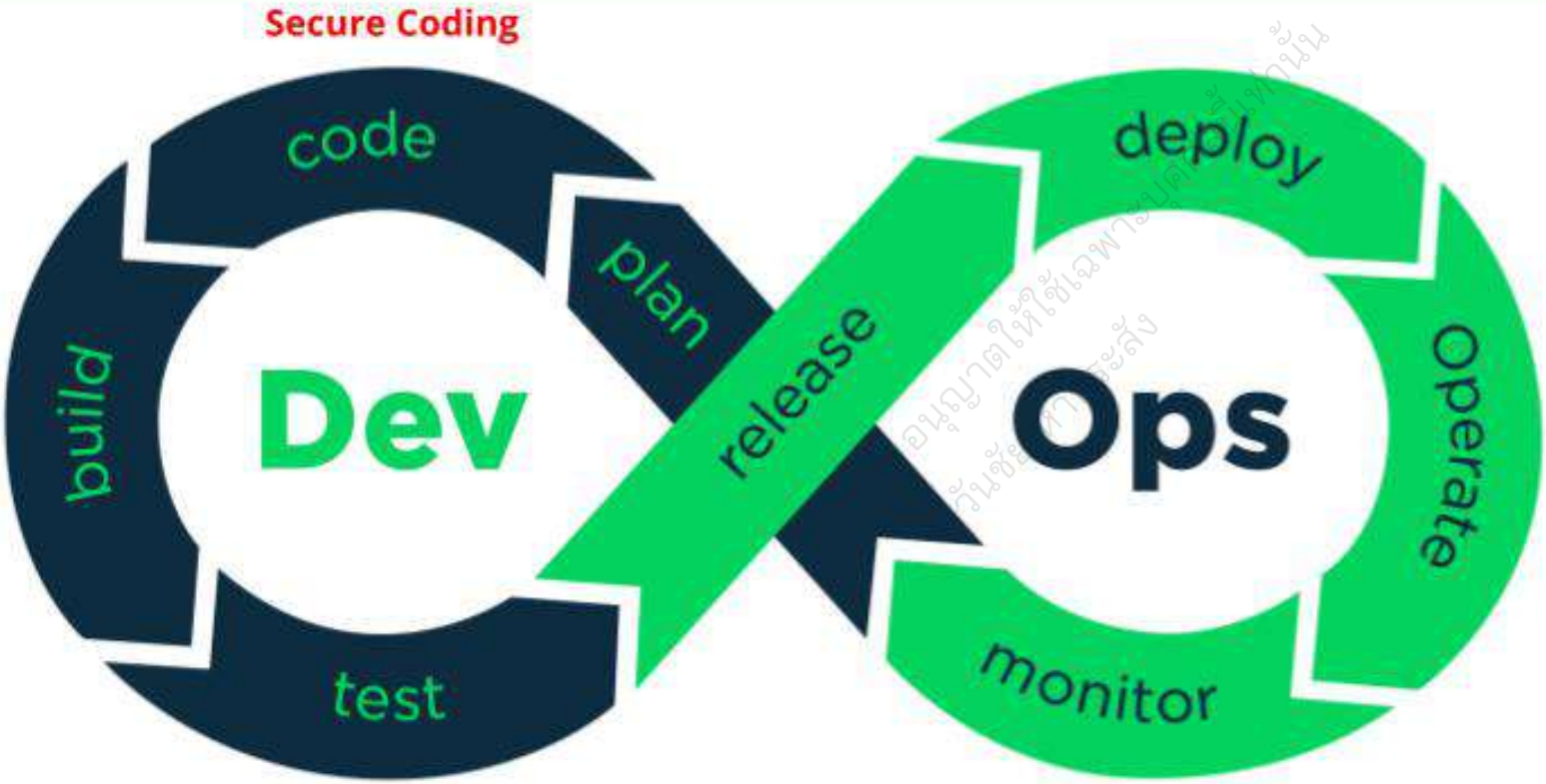
Opsta

Security Automation in every steps

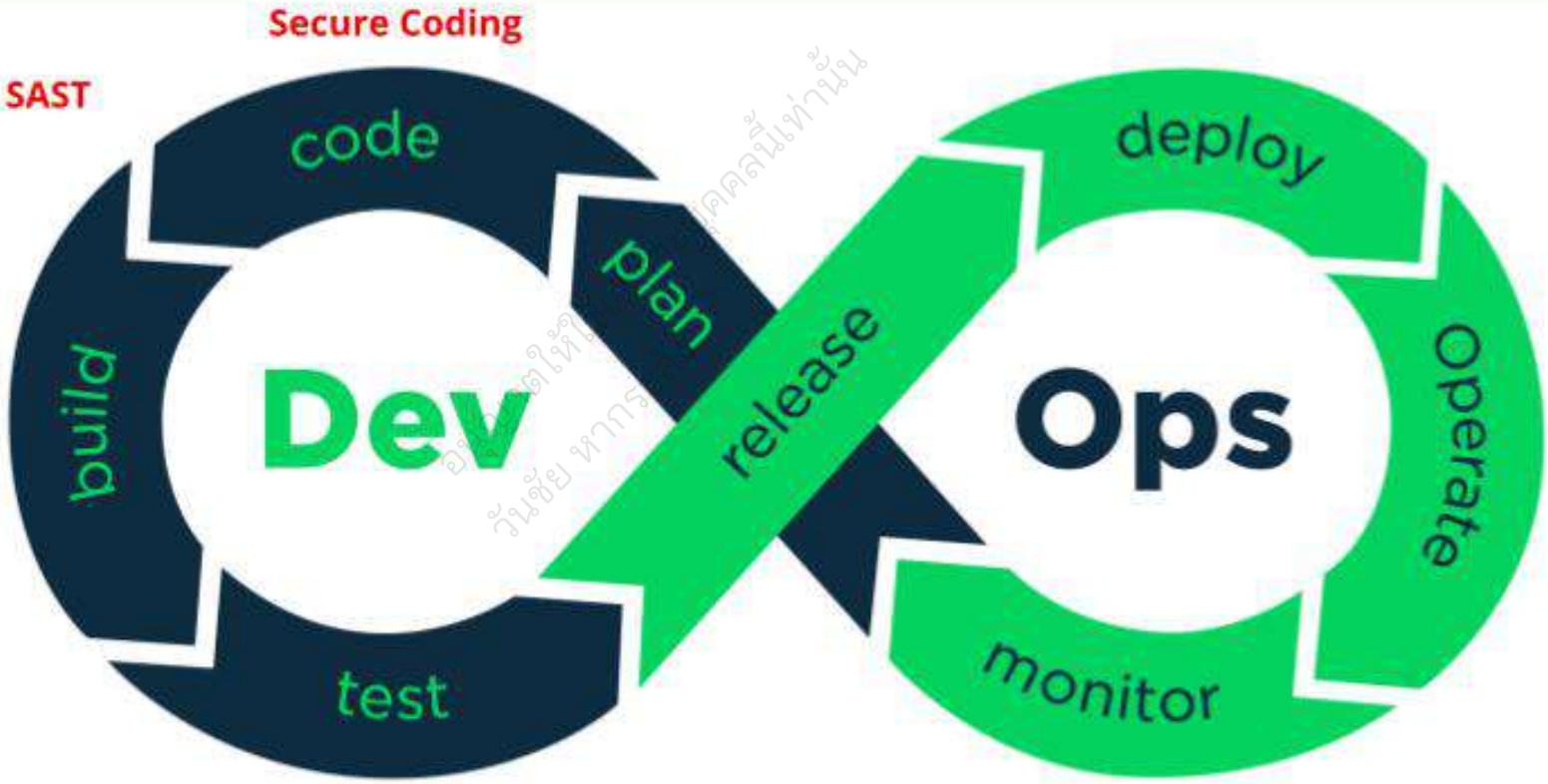


Skooldio Opsta

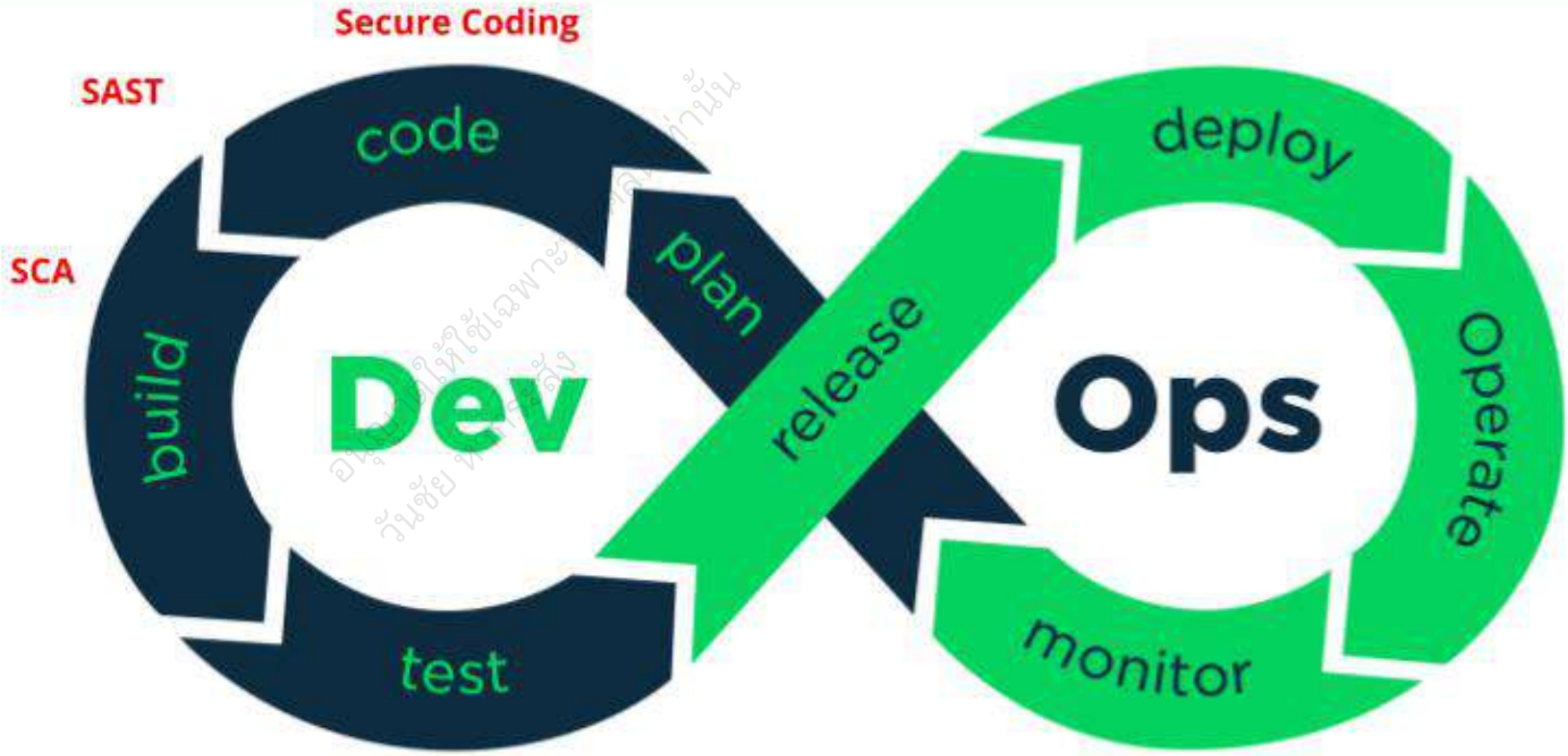
Security Automation in every steps



Security Automation in every steps

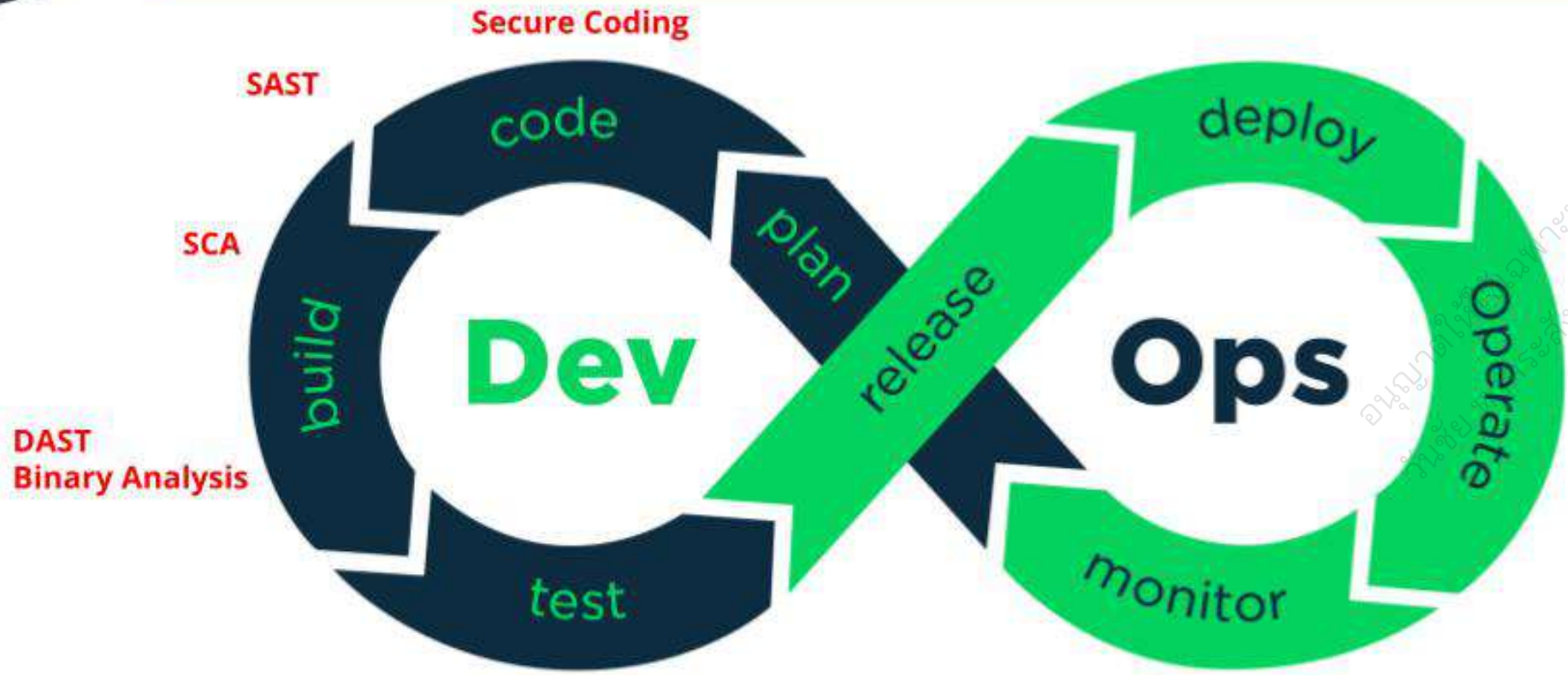


Security Automation in every steps



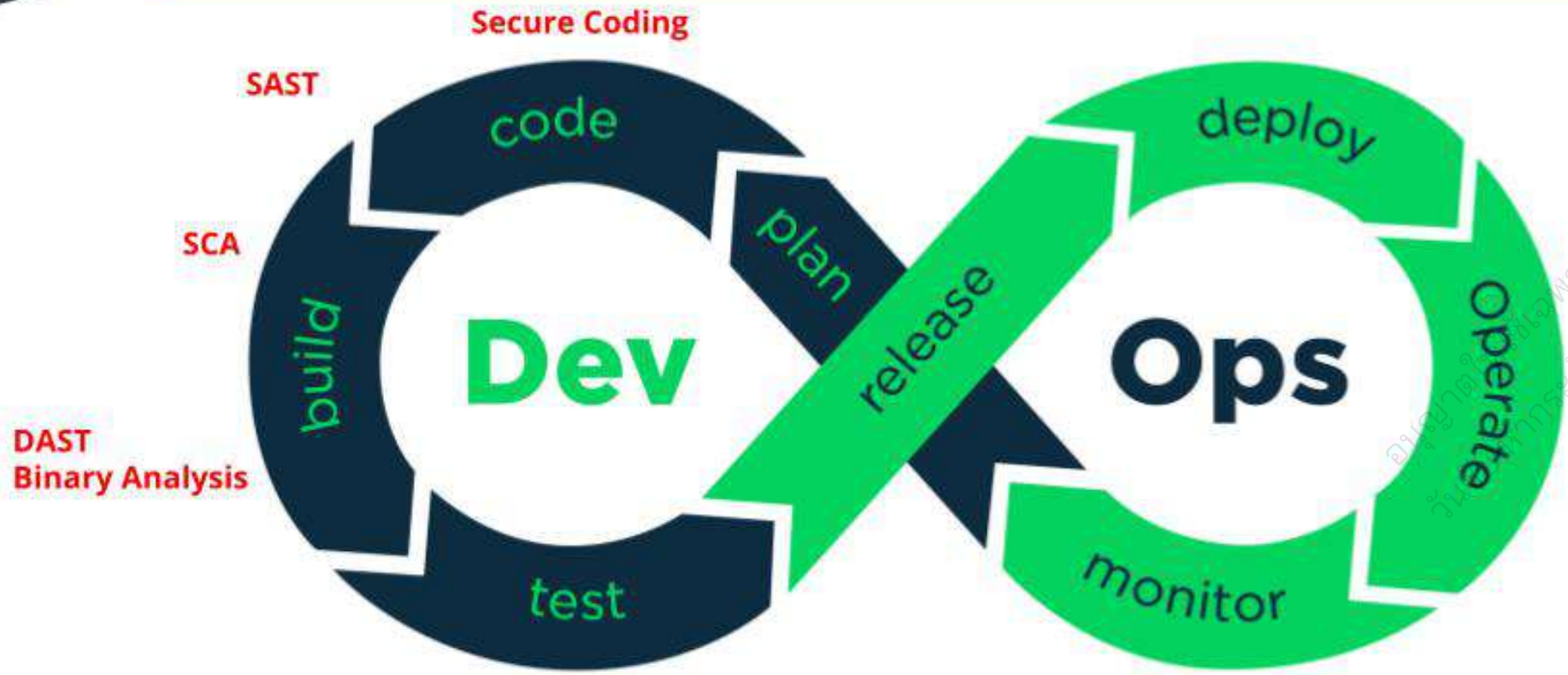
Skooldio Opsta

Security Automation in every steps



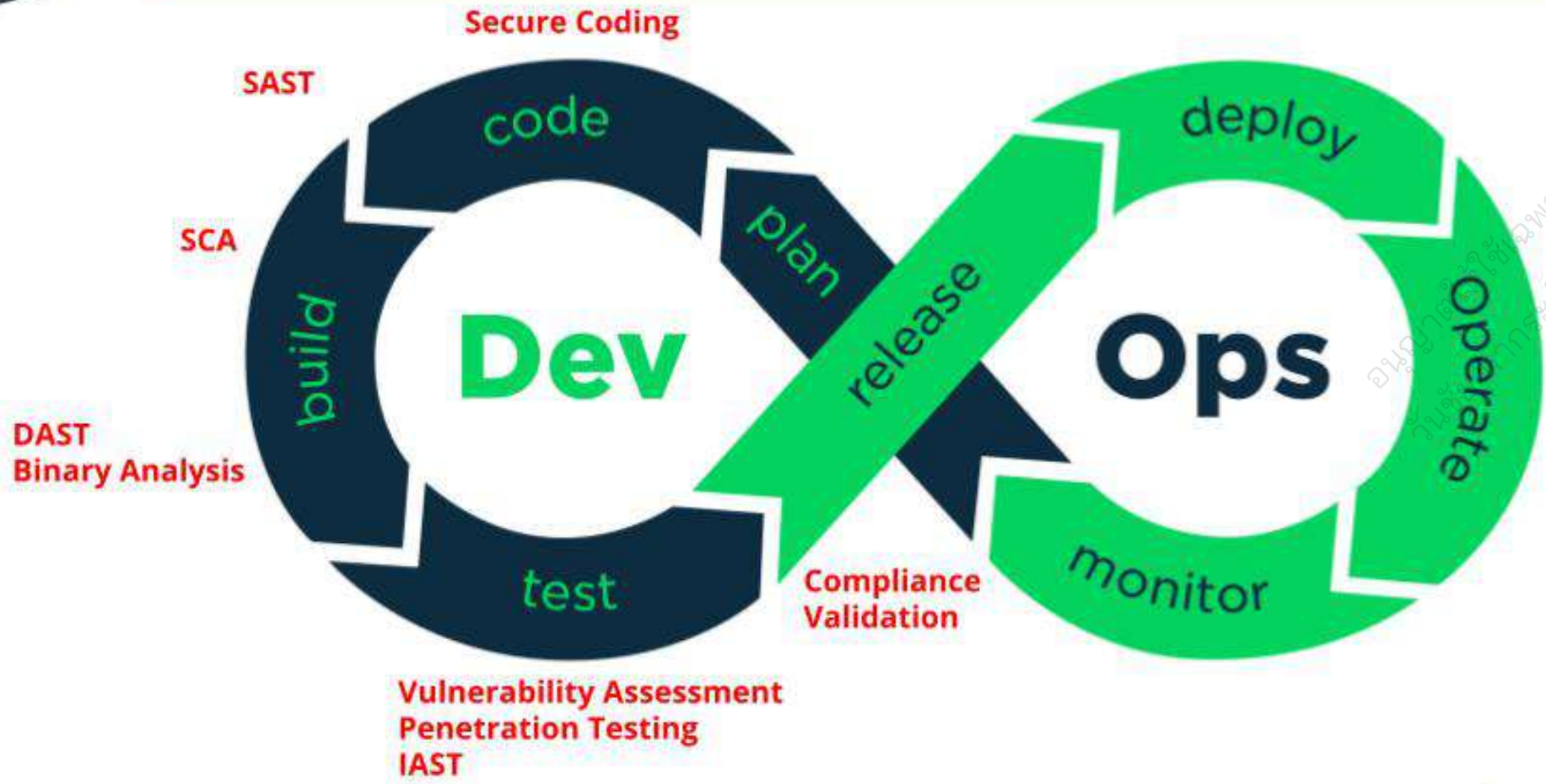
Skooldio Opsta

Security Automation in every steps



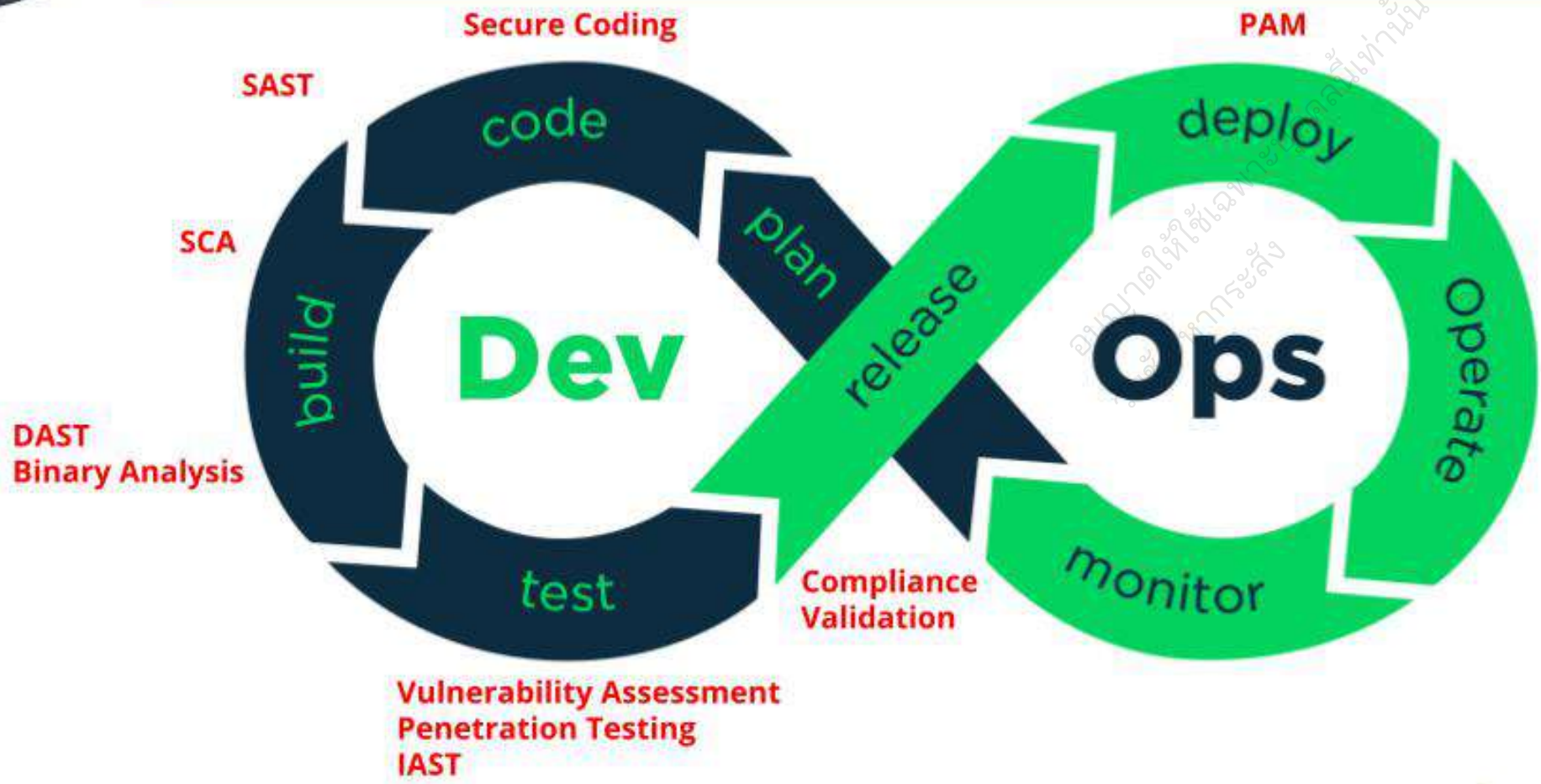
Skooldio Opsta

Security Automation in every steps



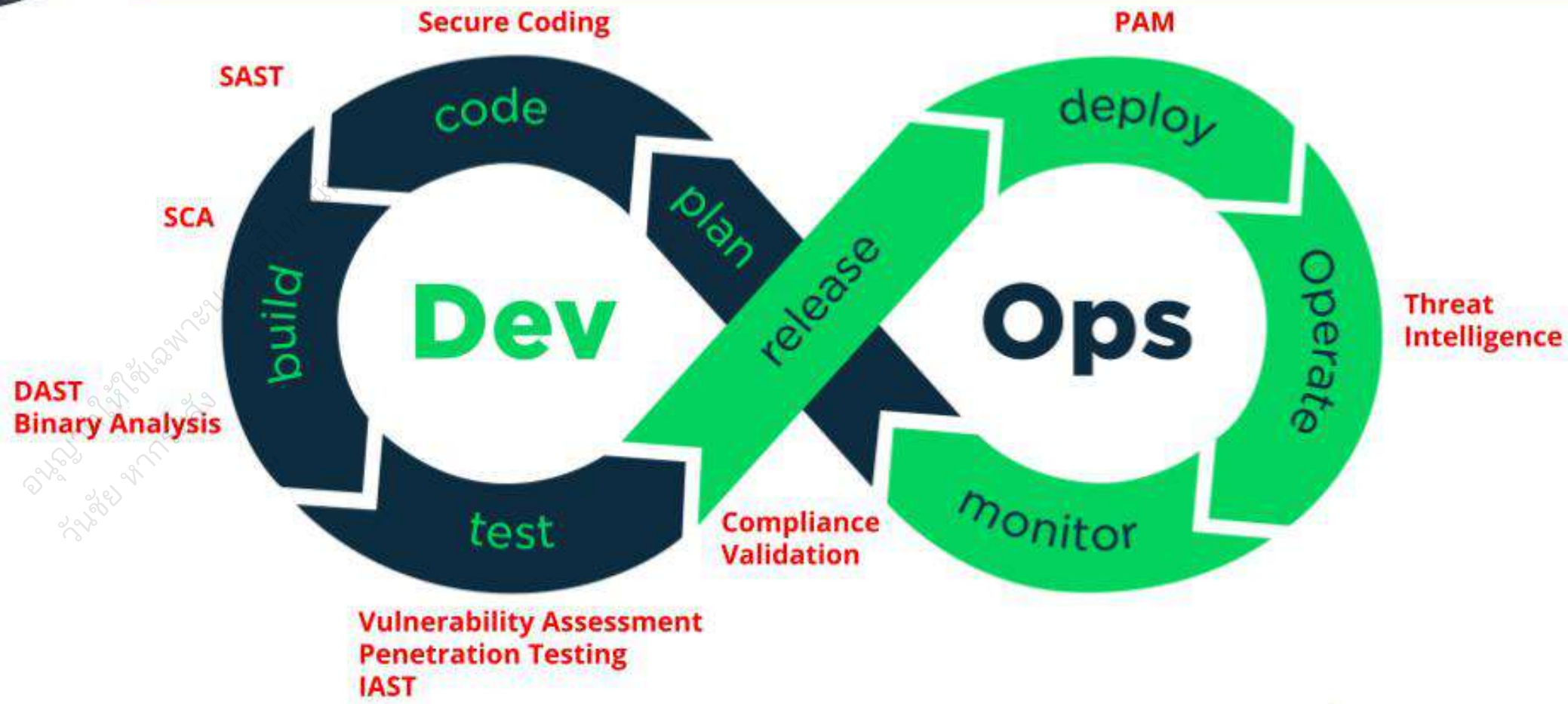
Skooldio Opsta

Security Automation in every steps



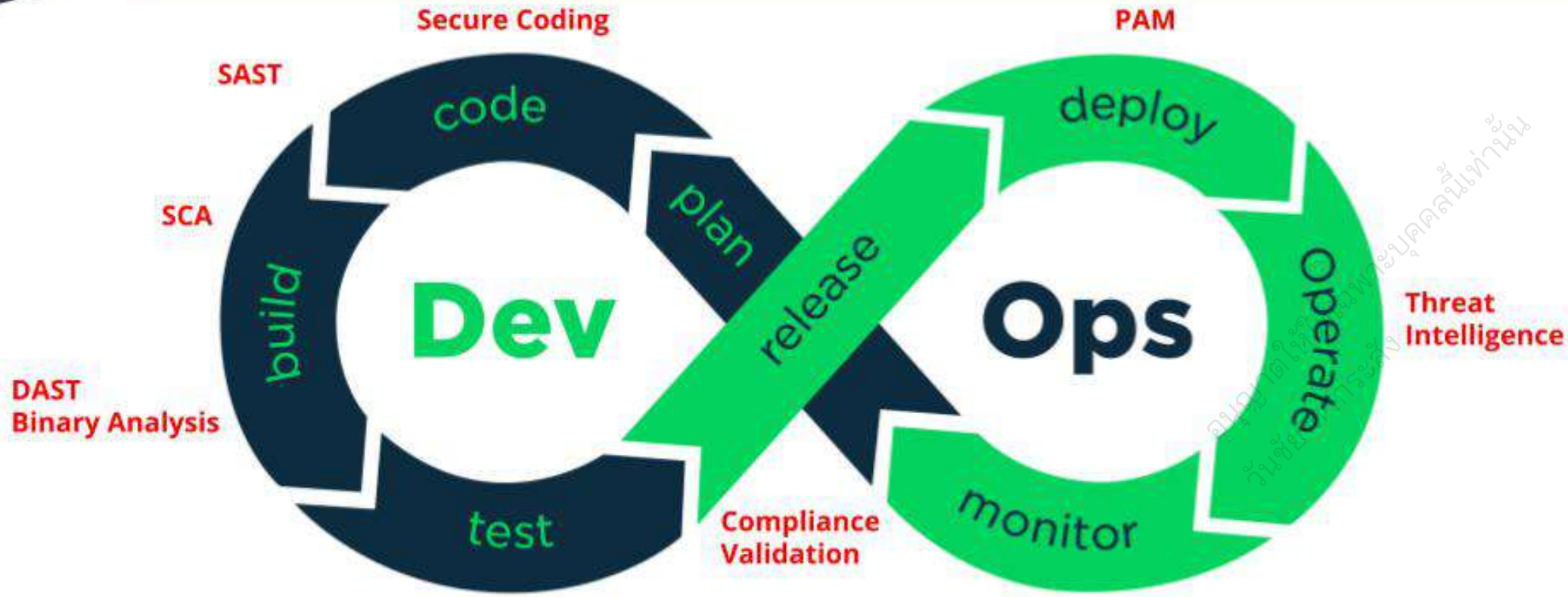
Skooldio Opsta

Security Automation in every steps



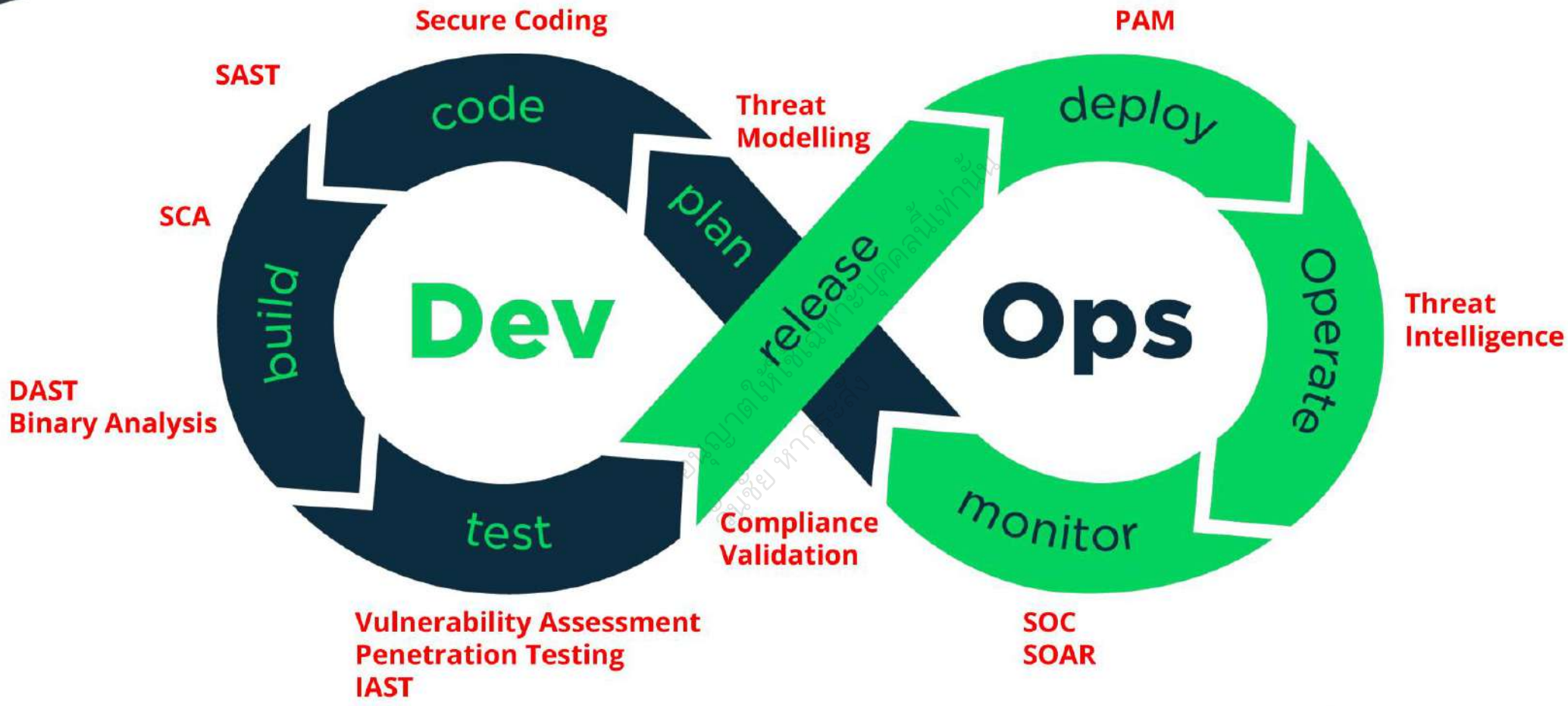
Skooldio Opsta

Security Automation in every steps



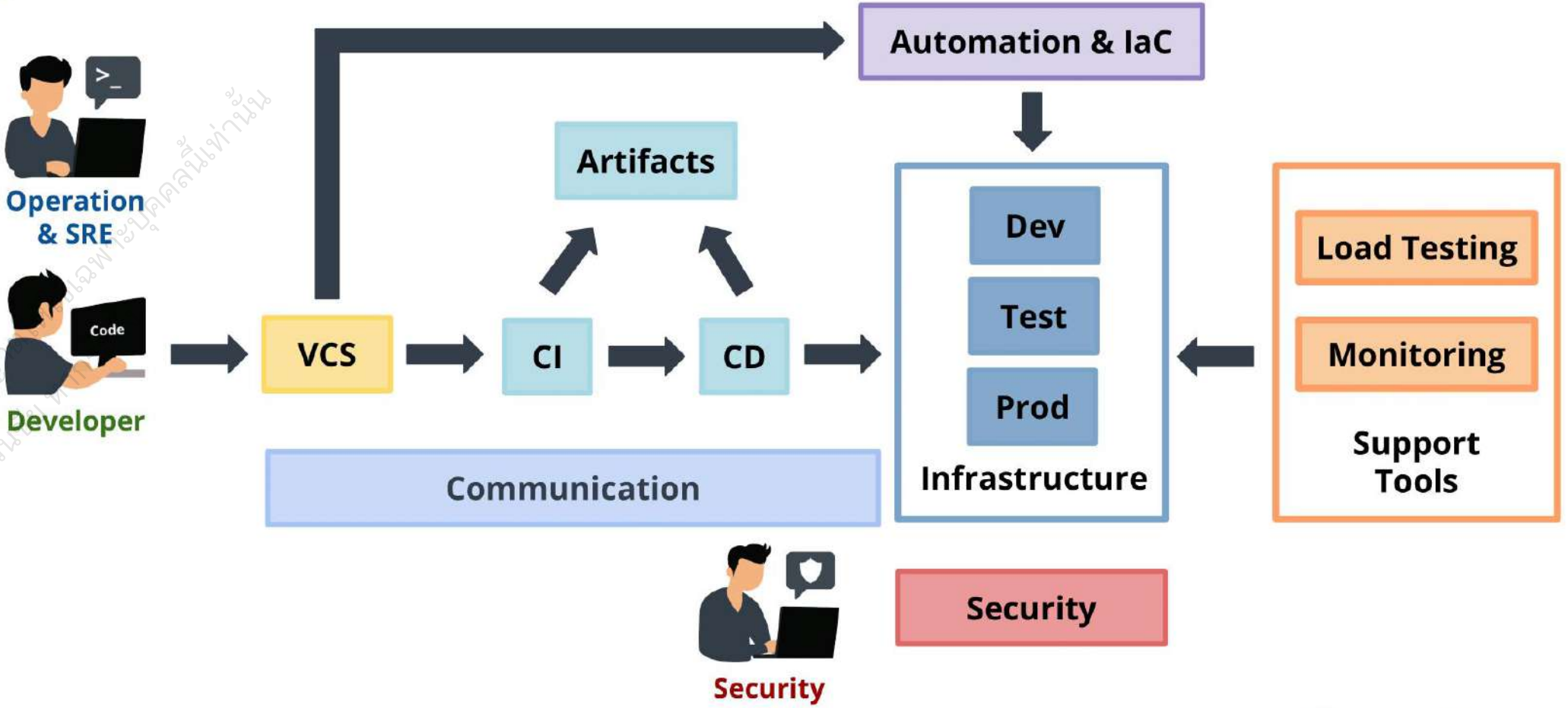
Skooldio Opsta

Security Automation in every steps

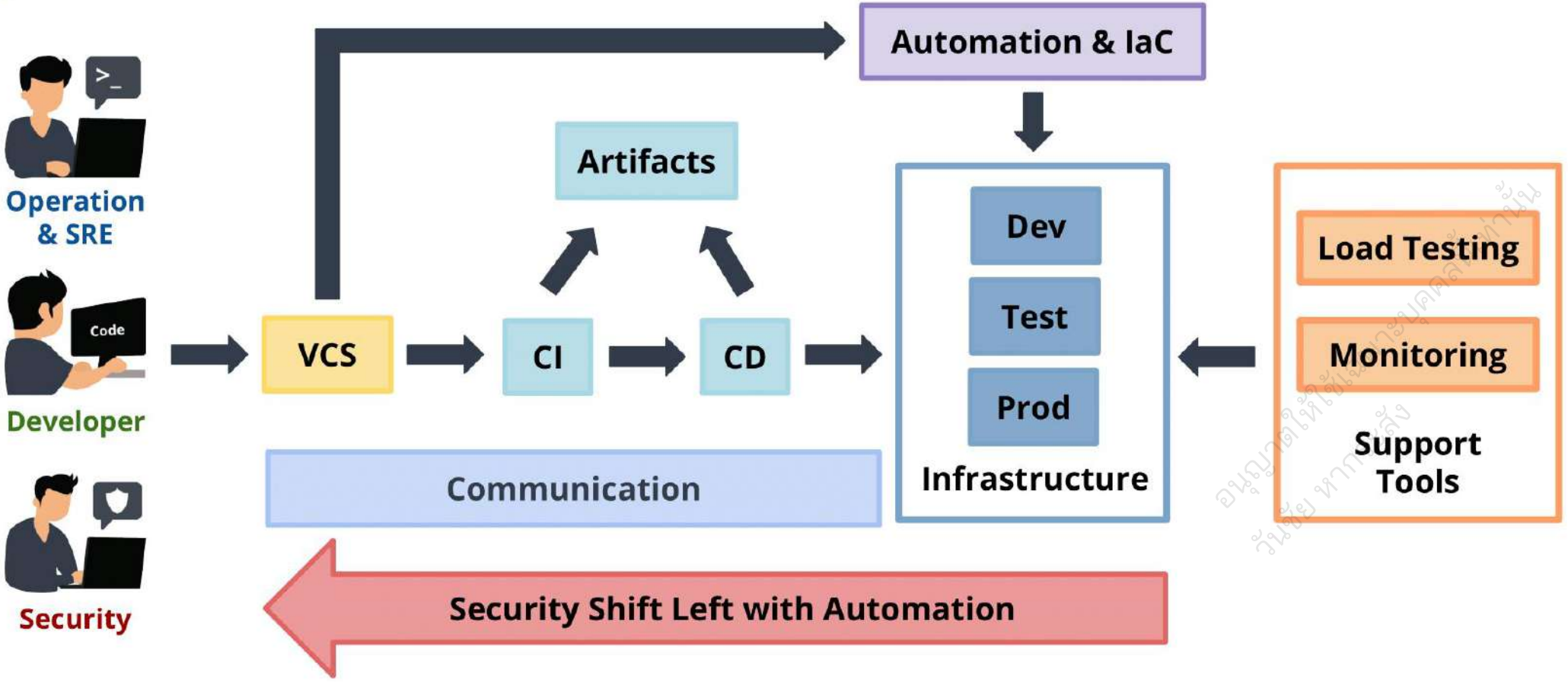


Skooldio Opsta

DevOps Flow



DevSecOps Flow



DevSecOps Tools

Skooldio Opsta

DevSecOps Tools

GitLab
DEPENDENCY-CHECK
clair
CYBERARK CONJUR
OpenVAS
Falco
splunk>
sonarqube
IQ
JFrog Xray
anchore
LogRhythm
HashiCorp Vault
ZAP
DATADOG

Multi-purpose Commercial

snyk
TREND MICRO
Checkmarx
paloalto NETWORKS
Check Point SOFTWARE TECHNOLOGIES LTD
aqua
MICRO FOCUS
RAPID7

Skooldio Opsta

Security Stages on DevOps Flow

Security Stages on DevOps Flow

Security Stages on DevOps Flow

Security Stages on DevOps Flow

Precommit Stage

Secure Coding

Secure coding is the **practice** of writing software that's **protected** from **vulnerabilities**. Some examples below refer from OWASP Secure Coding Practices

- Input Validation
- Authentication and Password Management
- Session Management
- Access Control
- Cryptographic Practices
- Error Handling and Logging
- Communication Security
- and much more...

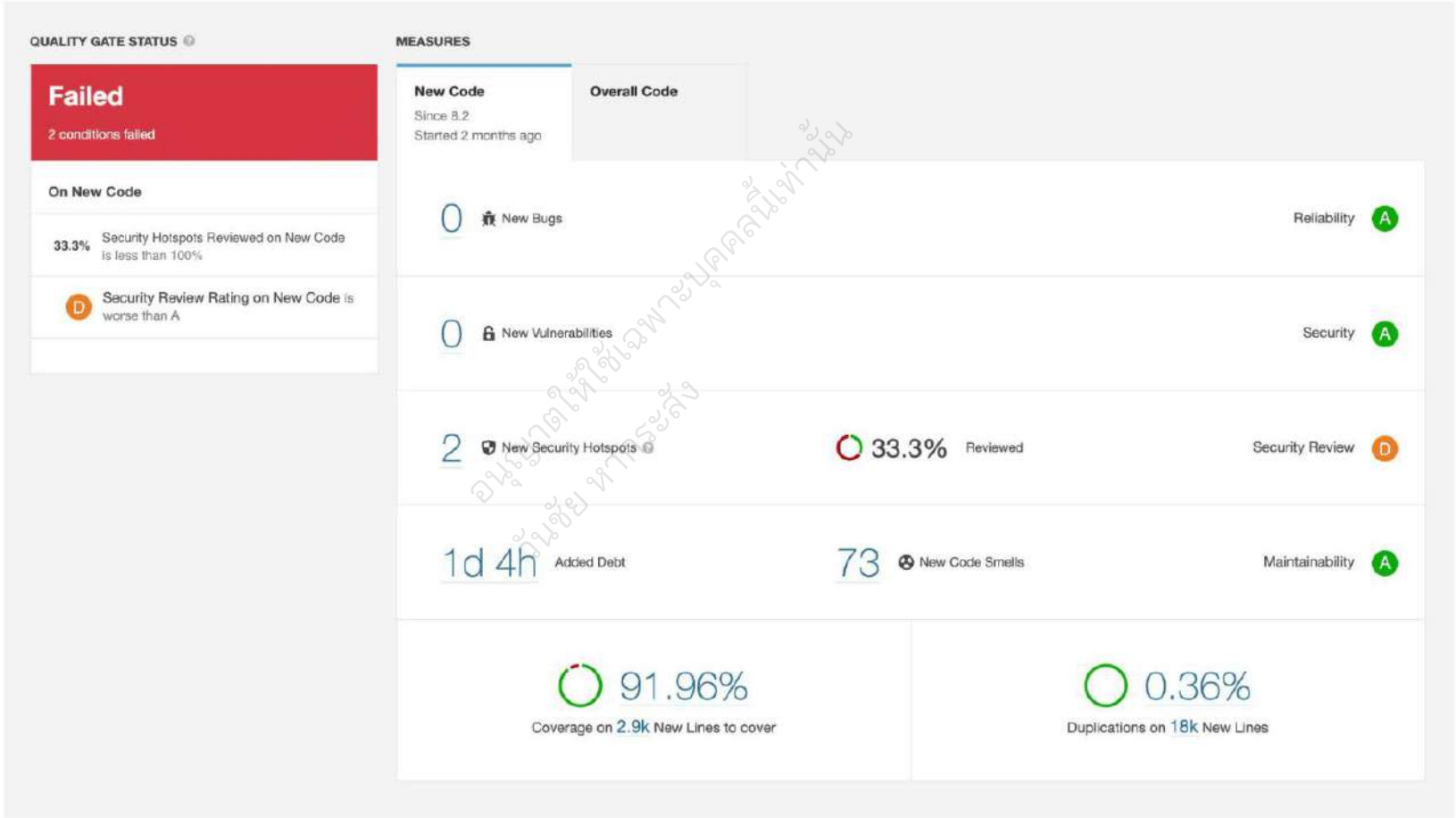
SAST (Static Application Security Testing)

SAST is a testing methodology that **analyzes source code** to find **security vulnerabilities**. SAST scans an application before the code is compiled. It's also known as **white box testing**.

sonarqube

MICRO FOCUS Fortify

SAST (Static Application Security Testing)



SCA (Software Composition Analysis)

SCA scans source code to **inventory** all **open-source components** to eliminate **vulnerabilities** those **listed** in the National Vulnerability Database **(NVD)** and **compatibility** issues with open-source **licenses**.

SCA (Software Composition Analysis)

Dependency-Check Results

SEVERITY DISTRIBUTION

6 | 16 | 37 | 1 | 1

Search

File Name	Vulnerability	Severity	Weakness
+ jackson-databind-2.8.11.3.jar	NVD CVE-2018-19360	Critical	CWE-502
+ jackson-databind-2.8.11.3.jar	NVD CVE-2018-19361	Critical	CWE-502
− jackson-databind-2.8.11.3.jar	NVD CVE-2018-19362	Critical	CWE-502

File Path	/Users/steve/.m2/repository/com/fasterxml/jackson/core/jackson-databind/2.8.11.3/jackson-databind-2.8.11.3.jar
SHA-1	844df5aba5a1a56e00905b165b12bb34116ee858
SHA-256	5582d55d615ea5ec09563558144c22cac46a06739a8561d7db4ff5c41532d6bc
Description	FasterXML jackson-databind 2.x before 2.9.8 might allow attackers to have unspecified impact by leveraging failure to block the jboss-common-core class from polymorphic deserialization.

+ jackson-databind-2.8.11.3.jar	NVD CVE-2019-12086	High	CWE-200
+ jquery-2.1.4.min.js	NVD CVE-2019-11358	Medium	CWE-79
+ jquery-2.2.4.min.js	NVD CVE-2015-9251	Medium	CWE-79
+ jquery-2.2.4.min.js	NVD CVE-2019-11358	Medium	CWE-79

« ‹ ... 1 2 3 4 5 ... › »

3 of 7

Acceptance Stage

Software Security Testing

Software security testing is the process of assessing and testing a system to discover security risks and vulnerabilities of the system and its data.

- **Penetration Testing** - The system undergoes analysis and attack from simulated malicious attackers.
- **Vulnerability Assessment** - The system is scanned and analyzed for security issues.
- **Fuzz Testing** - is a brute-force reliability testing technique wherein you create and inject random data into a file or API in order to intentionally cause errors and then see what happens

DAST

- **DAST (Dynamic Application Security Testing)** tools automate security tests for a variety of real-world threats. DAST is a **black-box testing** method to **identify vulnerabilities** in their applications from an **external perspective** to better simulate threats most easily accessed by hackers outside their organization

VA Scan and DAST

ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	2
Low	6
Informational	0

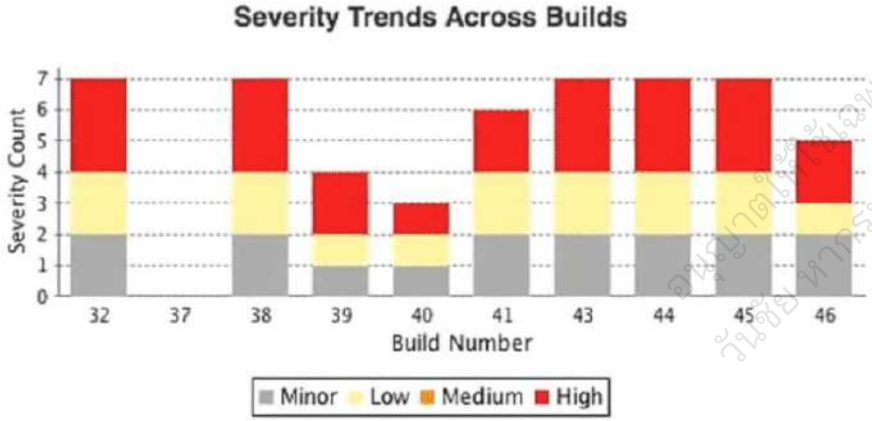
Alert Detail

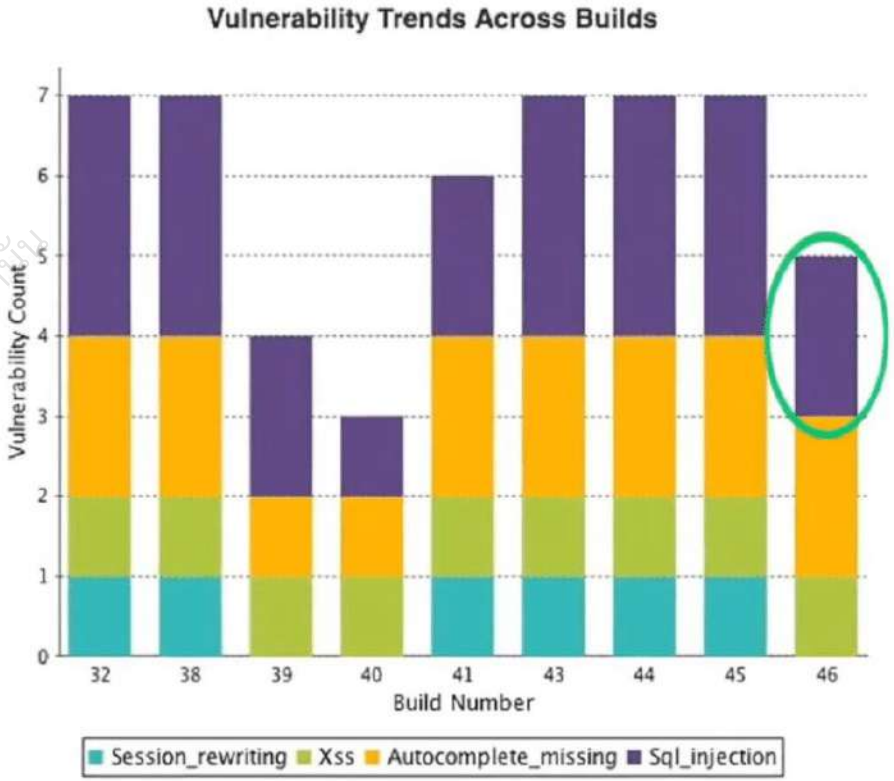
Medium (Medium)	X-Frame-Options Header Not Set
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.
URL	https://public-firing-range.appspot.com/address/location/documentwrite
Method	GET
Parameter	X-Frame-Options
URL	https://public-firing-range.appspot.com/cors/alloworigin/dynamicAllowOrigin
Method	GET
Parameter	X-Frame-Options
URL	https://public-firing-range.appspot.com/angular/angular_body_alt_symbols_raw/1.6.0?q=test
Method	GET
Parameter	X-Frame-Options

IAST (Interactive Application Security Testing)

IAST instruments applications by deploying **agents** and **sensors** in **running applications** and continuously analyzing all application interactions initiated by manual tests, automated tests, or a combination of both to **identify vulnerabilities in real time**

IAST (Interactive Application Security Testing)





Infrastructure as Code (IaC) Security

IaC Security test and **monitor** your **infrastructure as code** such as Ansible, Terraform modules and Kubernetes YAML, JSON, and Helm charts to **detect** configuration issues that could open your deployments to attack and **malicious behavior**.

Container Image Security

Container security software is used to **secure** multiple **components** of **containerized** applications or files, along with their infrastructure and connected networks. Testing capabilities will assist in developing security policies, **discover zero-day vulnerabilities**, and simulate attacks from common threat sources.

Container Image Security

Policy

Anchore Policy Evaluation Summary

Show 10 entries Search:

Repo Tag	Stop Actions	Warn Actions	Go Actions	Final Action
docker.io/library/ubuntu:latest	0	14	0	WARN

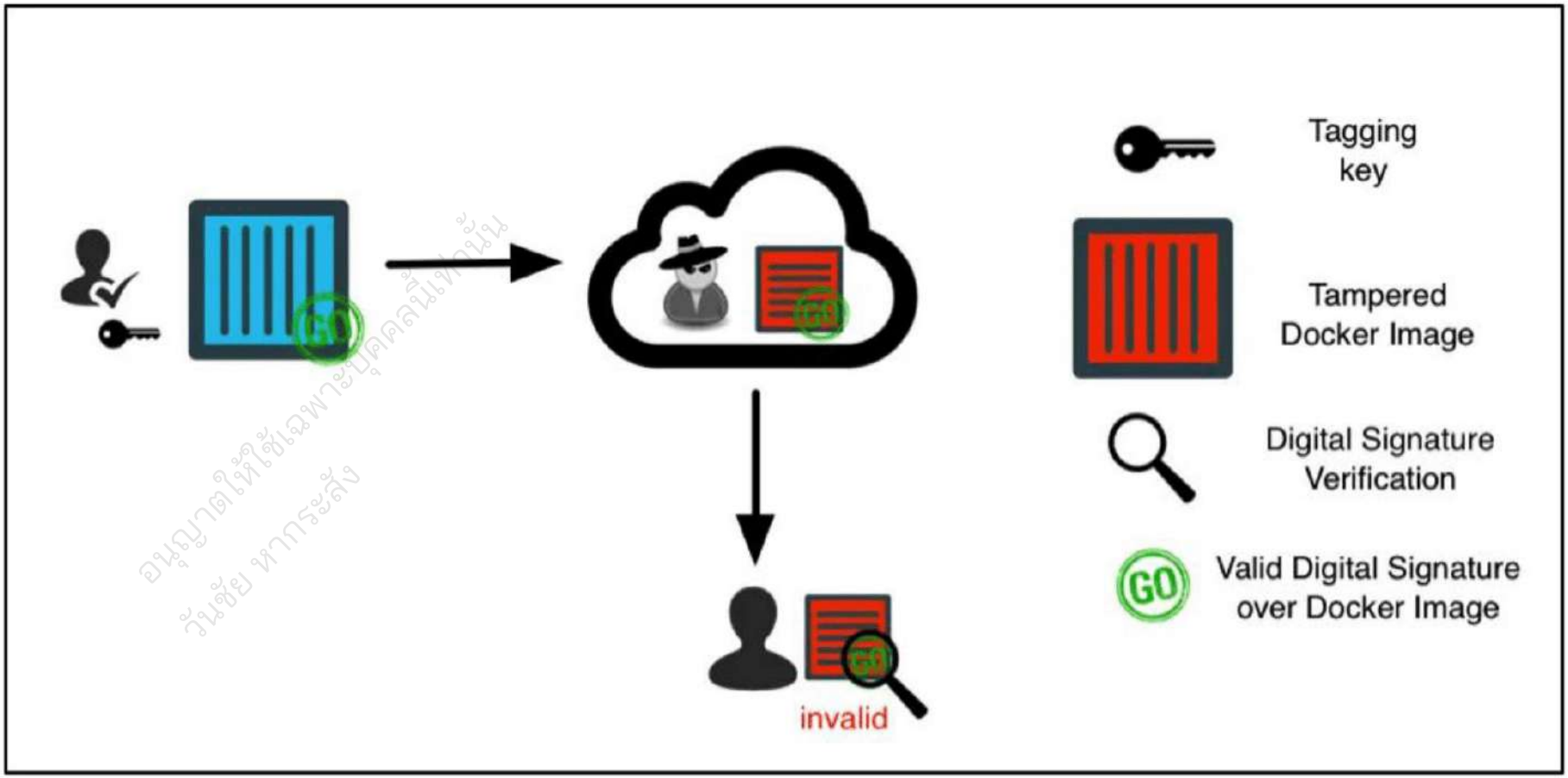
Showing 1 to 1 of 1 entries Previous 1 Next

Anchore Policy Evaluation Report

Show 10 entries Search:

Image Id	Repo Tag	Trigger Id	Gate	Trigger	Check Output	Gate Action	Whitelisted
f975c50357489439eb9145dbfa16bb7cd06c02c31aa4df45c77de4d2baa4e232	docker.io/library/ubuntu:latest	b38090bac771995c5af3fc8c033b7d3d	dockerfilecheck	nohealthcheck	Dockerfile does not contain any HEALTHCHECK instructions	WARN	false
f975c50357489439eb9145dbfa16bb7cd06c02c31aa4df45c77de4d2baa4e232	docker.io/library/ubuntu:latest	CVE-2018-6829+gnupg	anchoresec	vulnmedium	MEDIUM Vulnerability found in package - gnupg (CVE-2018-6829 - http://people.ubuntu.com/~ubuntu-security/cve/CVE-2018-6829)	WARN	false
f975c50357489439eb9145dbfa16bb7cd06c02c31aa4df45c77de4d2baa4e232	docker.io/library/ubuntu:latest	CVE-2018-6829+gpgv	anchoresec	vulnmedium	MEDIUM Vulnerability found in package - gpgv (CVE-2018-6829 - http://people.ubuntu.com/~ubuntu-security/cve/CVE-2018-6829)	WARN	false

Signed Container Image



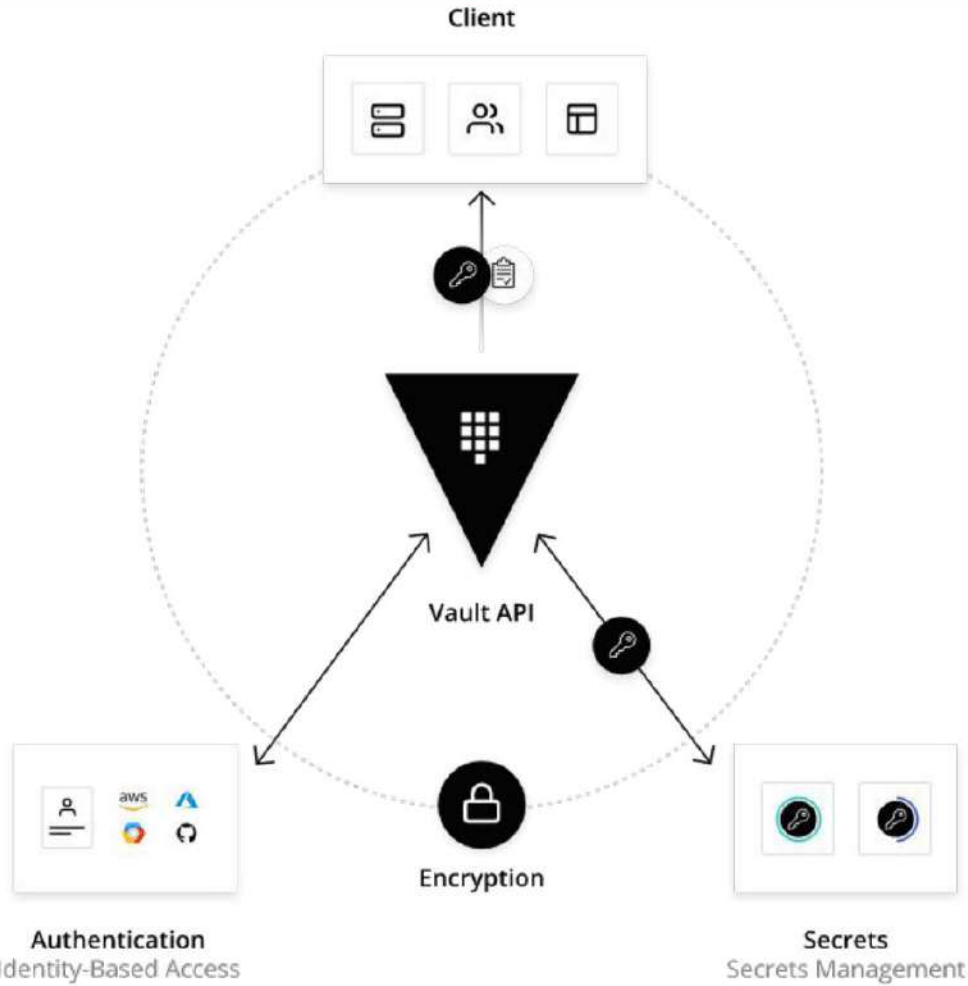
Privileged Access Management (PAM)

PAM software allows companies to secure their privileged **credentials** in a **centralized**, secure vault (a password safe). Additionally, these solutions control **who has access to**, and therefore **who can use**, the privileged credentials based on access policies (including user permissions and specific timeframes), often recording or **logging** user activity while using the credentials.

CYBERARK
CONJUR

HashiCorp
Vault

Privileged Access Management (PAM)



Production Stage

Automation Security Baseline

Automation Security Baseline build **standard hardening** steps into your recipes instead of using scripts or manual checklists. This includes minimizing the attack surface by removing all packages that aren't needed and that have known problems, and changing default configurations to be safe.

CIS. Center for Internet Security®

NIST Security Configuration CHECKLISTS http://checklists.nist.gov CSD

OpenSCAP

Automation Security Baseline Tools

Cloud Security Automation

- **Monitoring** - it is necessary that you monitor the workflow of all the tasks in your infrastructure.
- **Evaluation** - give you insights into which tasks can be automated like repetitive tasks, resource provisioning, deployments, creating security rules, etc.
- **In-depth analysis** - analyze the collected information in depth by differentiating it on the basis of severity as high, medium or low risk.
- **Reporting** - The automation processes should be configured to generate the reports to present the overview of the changes before or after.
- **Remediations** - implement remediation and improve overall security posture.

RASP (Run-time Application Security Protection)

RASP works **inside** the **application**. It's **plugged** into an application or its **runtime** environment and can control application execution. RASP lets an app run **continuous security checks** on itself and **respond to live attacks** by terminating an attacker's session and alerting defenders to the attack.

WAF (Web Application Firewall)

WAF or Web Application Firewall helps protect web applications by **filtering** and **monitoring** HTTP **traffic** between a web application and the Internet. It typically protects web applications from attacks such as cross-site forgery, cross-site-scripting (XSS), file inclusion, and SQL injection, among others. A WAF is a protocol **layer 7** defense (in the OSI model)

Security Monitoring

Security monitoring, sometimes referred to as **"security information monitoring (SIM)"** or **"security event monitoring (SEM),"** involves **collecting** and **analysing** information to **detect suspicious behavior** or unauthorised system changes on your network, defining which types of behavior should trigger alerts, and taking action on alerts as needed.

splunk> LogRhythm DATADOG

How to start DevSecOps

Technologies

Skooldio

Opsta

Impediment and Pain Point

Problem	Priority (1 is highest)	
Test ผ่าน แต่เกิดปัญหาตอนขึ้น Production		
ใช้เวลานานและยุ่งยากในขั้นตอนการสร้าง Environment ใหม่		
Program & Application มี bug เยอะ (Reliable)		
ความปลอดภัยของ Application ที่สร้างขึ้น (Security)		
ขึ้นระบบหรือ deploy แต่ละครั้งช้า ติดขั้นตอนภายในบริษัทนานเกิน		
มีความรู้ไม่เพียงพอ ทำ DevSecOps ช้า หรือคนทำไม่เพียงพอ		
ระบบล่มโดยไม่ทราบสาเหตุ หรือหาสาเหตุได้ช้า		
ไม่แน่ใจว่าระบบจะรับโหลดหนักๆ ไหวหรือเปล่า		
ไม่มีเอกสาร หรือวิธีการ หรือขาด Standard ในการทำ DevSecOps		

When we should do DevSecOps

- Team wants to deploy **more than 1 time** per day
- Team wants to solve **“work in my laptop”** problem
- Team wants more **reliable** deliver process
- Team wants to **rollback quickly**
- Team wants to **quickly troubleshooting** application problem
- Team have to **frequently and quickly spawn new environments**

Some DevSecOps Requirements

- Team has **time** to improve code to be Cloud Native Application
- Team can invest at least **CPU 20 cores, memory 64GB** and **HDD 1TB** to build **one application**
- Team ready to **accept failure on production** with **no blame culture**
- Team ready to **change the development process** to make it faster
- You need application that can run on **Linux** for **Kubernetes**

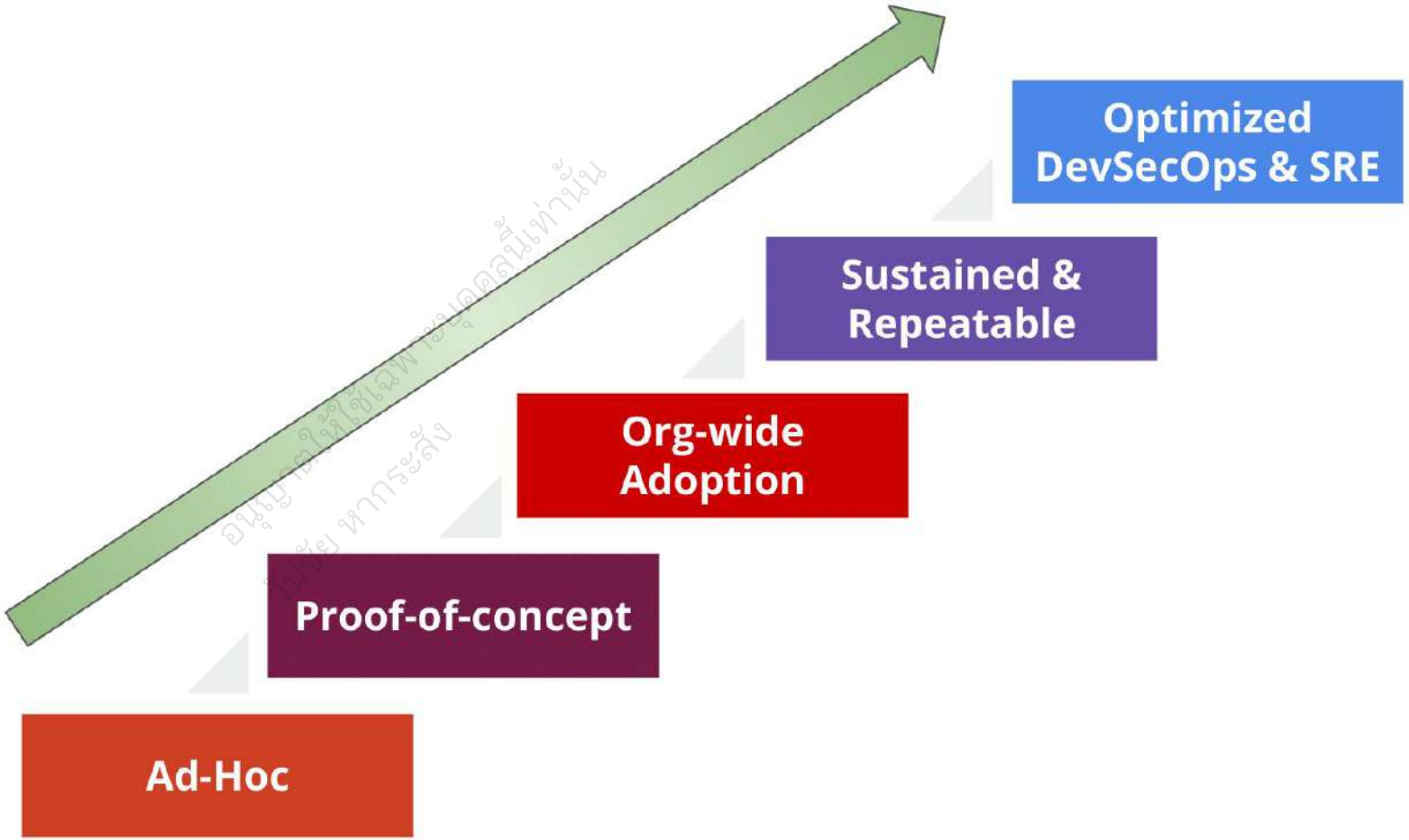
DevSecOps Maturity

Technologies

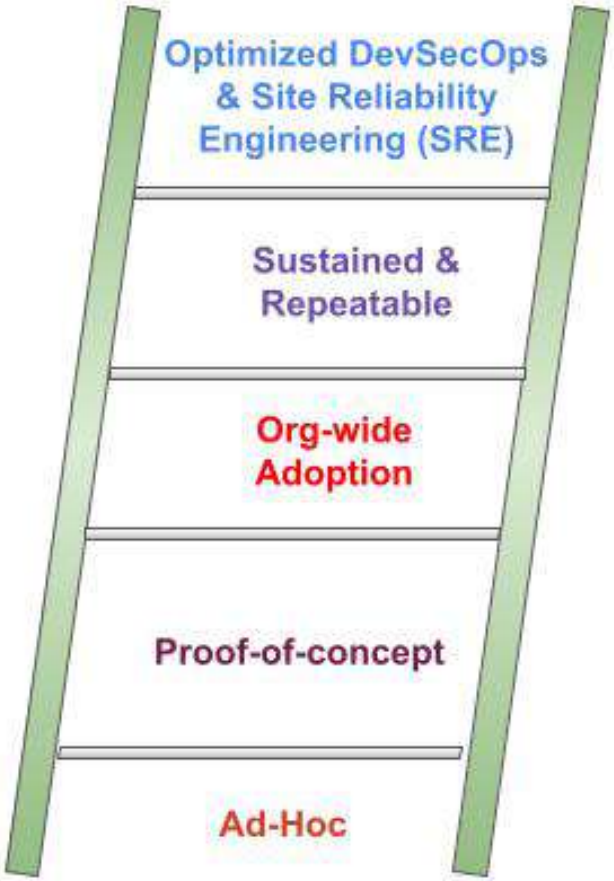
Skooldio

Opsta

DevSecOps Maturity Model



DevSecOps Maturity Levels Explain



- Lessons learnt using continuous feedback is incorporated back to improve the DevSecOps implementation
- Appropriate mix of tools & frameworks used for optimized outcomes
- Development processes updated to optimize DevSecOps outcomes

- Sustained DevSecOps implementation across different teams
- DevSecOps Governance
- Tracking & reporting

- DevSecOps rolled out for all the teams
- Mentoring sessions across the organization
- DevSecOps implementation governed & reported

- DevSecOps implementation planned
- Team mentored
- Tools & processes chosen
- 3-4 teams chosen
- Implementation done; lessons learnt

- Team across organization doing ad-hoc implementation
- Different tools & framework & processes used across different team

DevSecOps Adoption and Metrics

Strategies for Scaling DevSecOps (1)

- **Training Center (sometimes referred to as a DOJO)**
  Where people are taken out of their normal work routines to learn new tools or technologies, practices, and even culture for a period of time
- **Communities of Practice**
  Where groups that share common interests in tooling, language, or methodologies are fostered within an organization to share knowledge and expertise with each other, across teams, and around the organization.
- **Center of Excellence**
  Where all expertise lives and then consults out to others.

https://services.google.com/fh/files/misc/state-of-devops-2019.pdf

Strategies for Scaling DevSecOps (2)

- **Proof of Concept but Stall**
  A Proof of Concept (PoC) project, where a central team is given the freedom to build in whatever way they feel is best, often by breaking organizational norms (and often formal rules). However, the effort stalls after the PoC.
- **Proof of Concept as a Template**
  Starting with a small Proof of Concept (PoC) project and then replicating this pattern in other groups, using the first as a pattern.
- **Proof of Concept as a Seed**
  Starting with a small Proof of Concept (PoC), then spreading PoC knowledge to other groups to share the knowledge and practices learned.

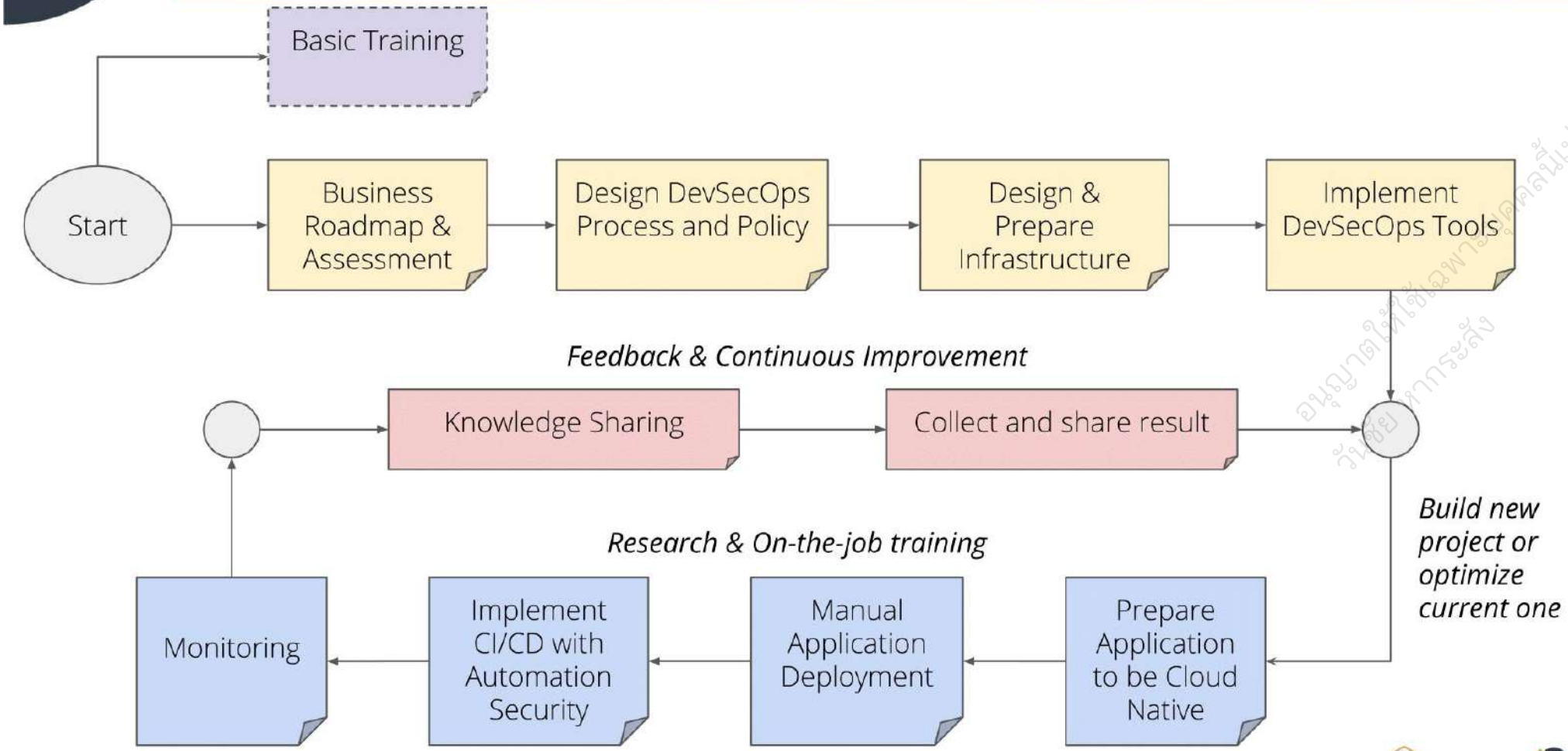
https://services.google.com/fh/files/misc/state-of-devops-2019.pdf

Strategies for Scaling DevSecOps (3)

- **Big Bang**
  Where the whole organization transforms to DevSecOps methodologies all at once, often with top-down directive.
- **Bottom-up or Grassroots**
  Where small teams close to the work pull together resources to transform and then informally share their success throughout the organization and scale without any formal organizational support
- **Mashup**
  Where the org implements several approaches described above, often only partially executed or with insufficient resources or prioritization to enable success.

https://services.google.com/fh/files/misc/state-of-devops-2019.pdf

DevSecOps Adoption Flow



Skooldio Opsta

Performance Metrics

https://services.google.com/fh/files/misc/state-of-devops-2019.pdf

Skooldio Opsta

Performance Metrics

SOFTWARE DEVELOPMENT

Lead Time

https://services.google.com/fh/files/misc/state-of-devops-2019.pdf

Skooldio Opsta

Performance Metrics

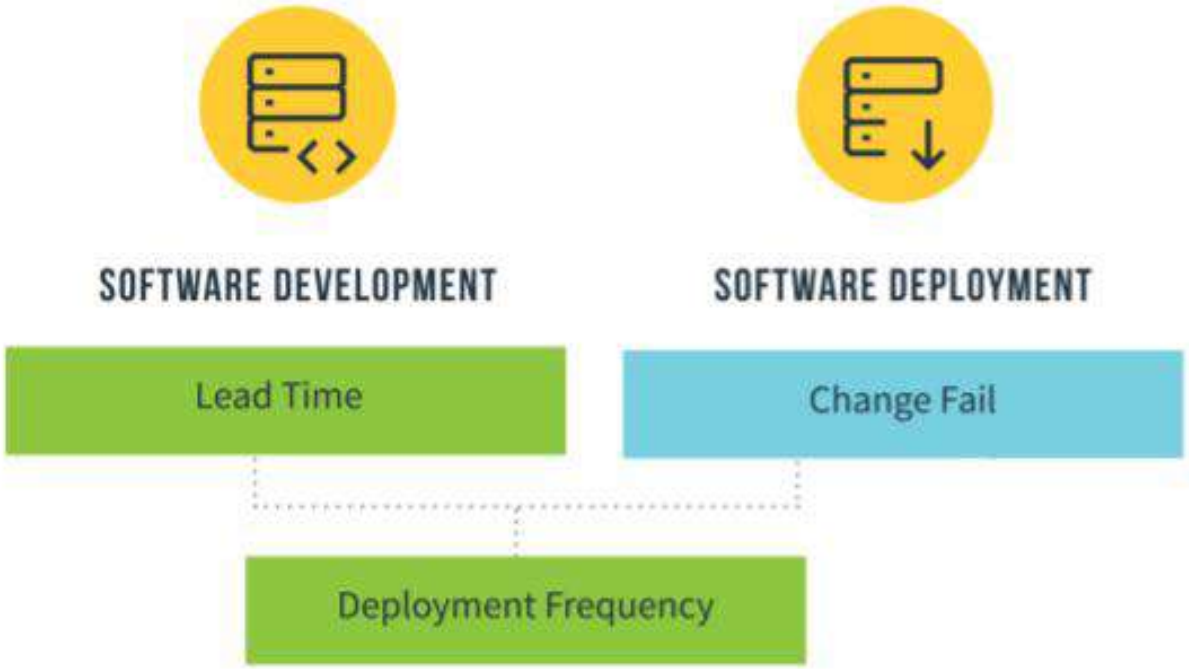
SOFTWARE DEVELOPMENT

Lead Time

Deployment Frequency

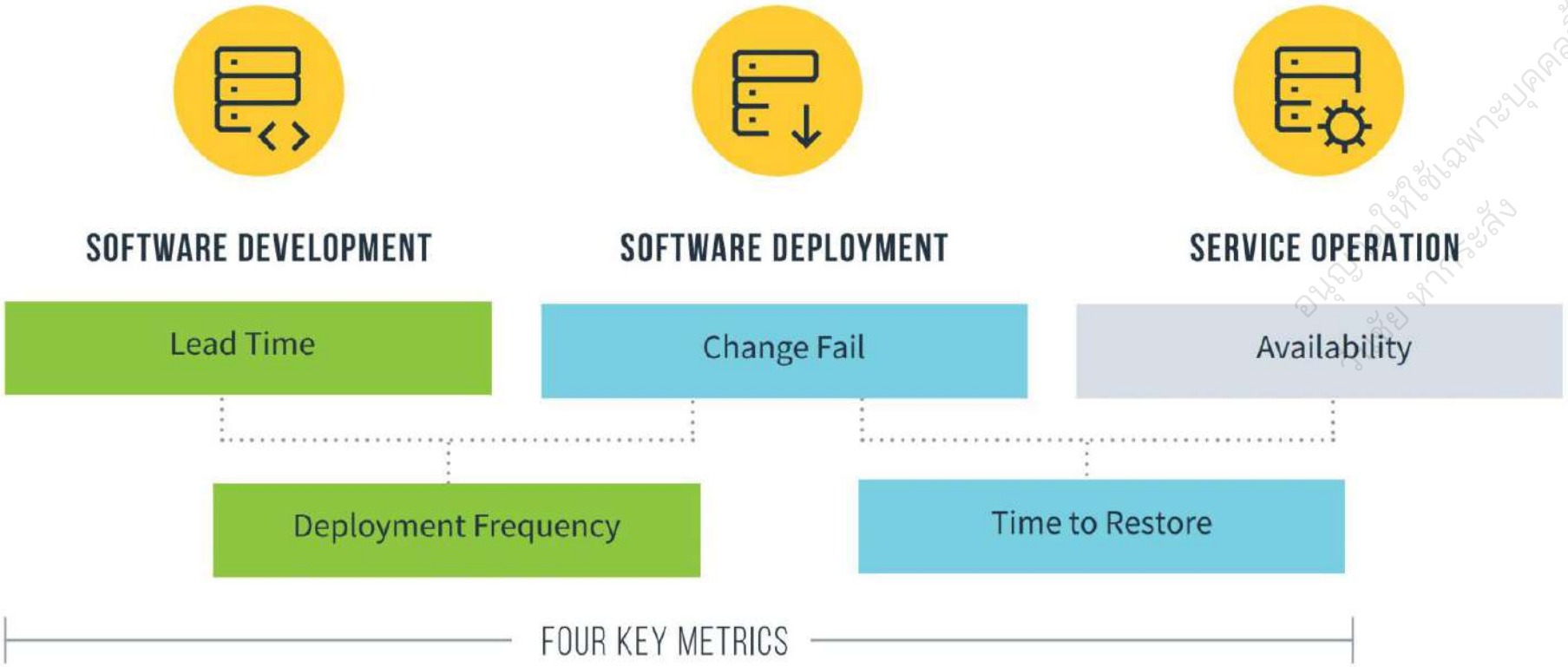
https://services.google.com/fh/files/misc/state-of-devops-2019.pdf

Performance Metrics



https://services.google.com/fh/files/misc/state-of-devops-2019.pdf

Performance Metrics



https://services.google.com/fh/files/misc/state-of-devops-2019.pdf

Software Delivery Performance

Aspect of Software Delivery Performance*	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per day and once per week	Between once per week and once per month	Between once per month and once every six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one day	Between one day and one week	Between one week and one month	Between one month and six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day[a]	Less than one day[a]	Between one week and one month
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0-15%[b,c]	0-15%[b,d]	0-15%[c,d]	46-60%

https://services.google.com/fh/files/misc/state-of-devops-2019.pdf

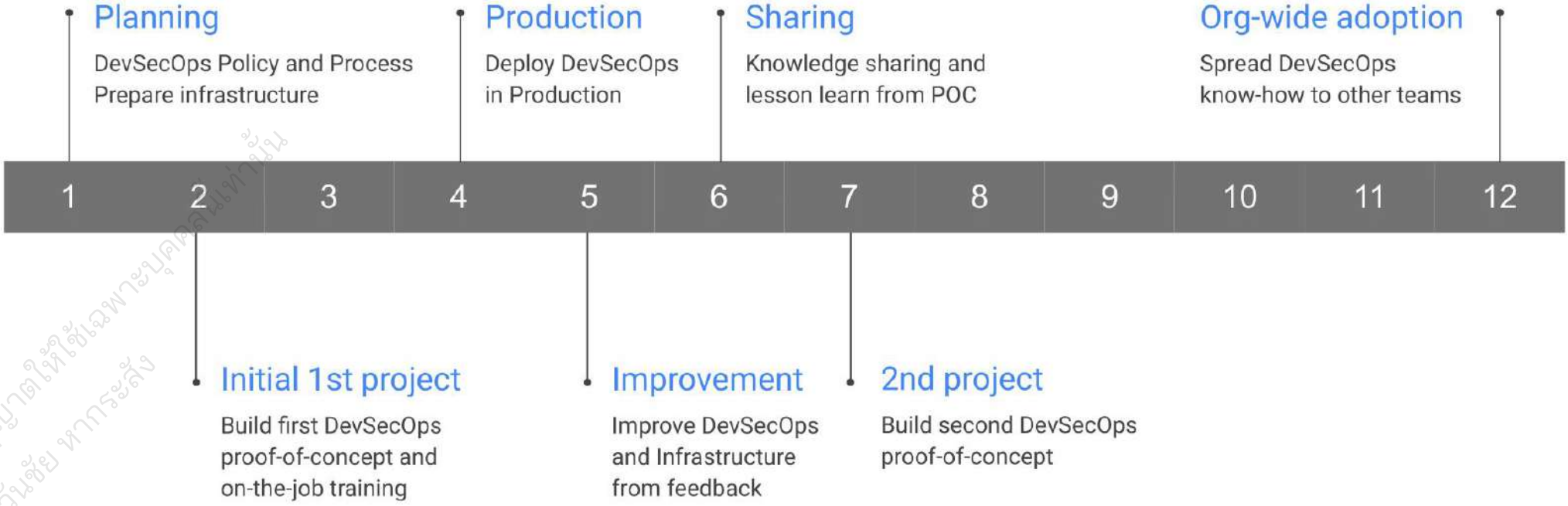
Assessment and Roadmap

Technologies

Skooldio

Opsta

DevSecOps Sample Business Roadmap

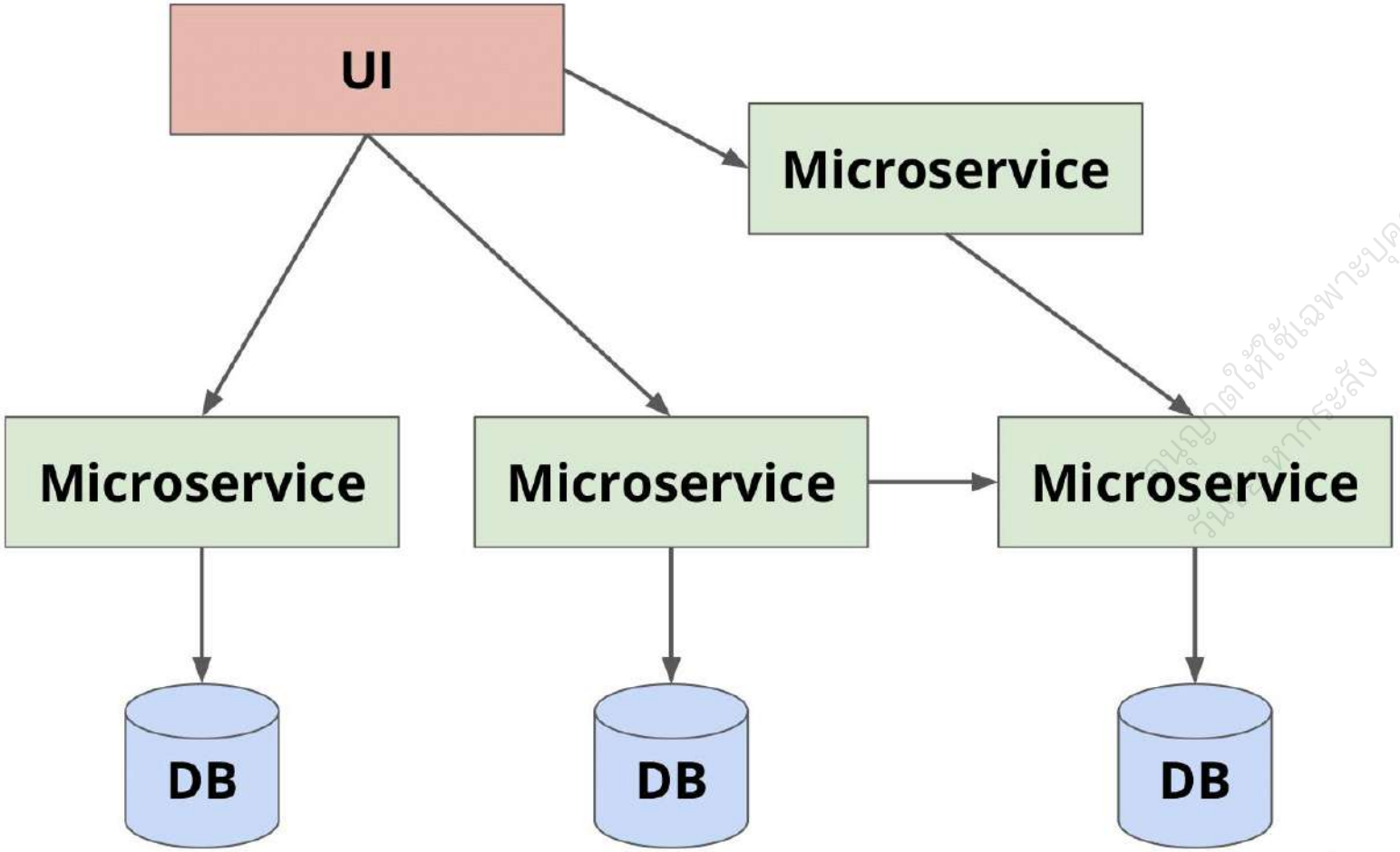


Skooldio Opsta

Pilot Project Details

No	Questions	Sample Answer	Answer
1	*Application Name*	Authentication	
2	*Application Summary*	ระบบการยืนยันตัวตน	
3	*Developer*	In-House, Outsource	
4	*User*	บุคคลทั่วไป, แผนกบัญชี, middleware	
5	*Programming Languages and Frameworks*	Frontend: Javascript React Backend: Python Django Mobile: Native iOS and Android	
6	*Databases*	MariaDB 10.3 and Redis 5.x	
7	*Infrastructure*	VMWare, GCP, AWS, Azure	
8	*Environments*	Dev For Developer SIT For In-House Tester UAT For Customer PRD For Production	
9	*Workload*	หน้าแรก 100 CCU per minute หน้า search 50 CCU per minute	
10	*SLA*	99.95%	

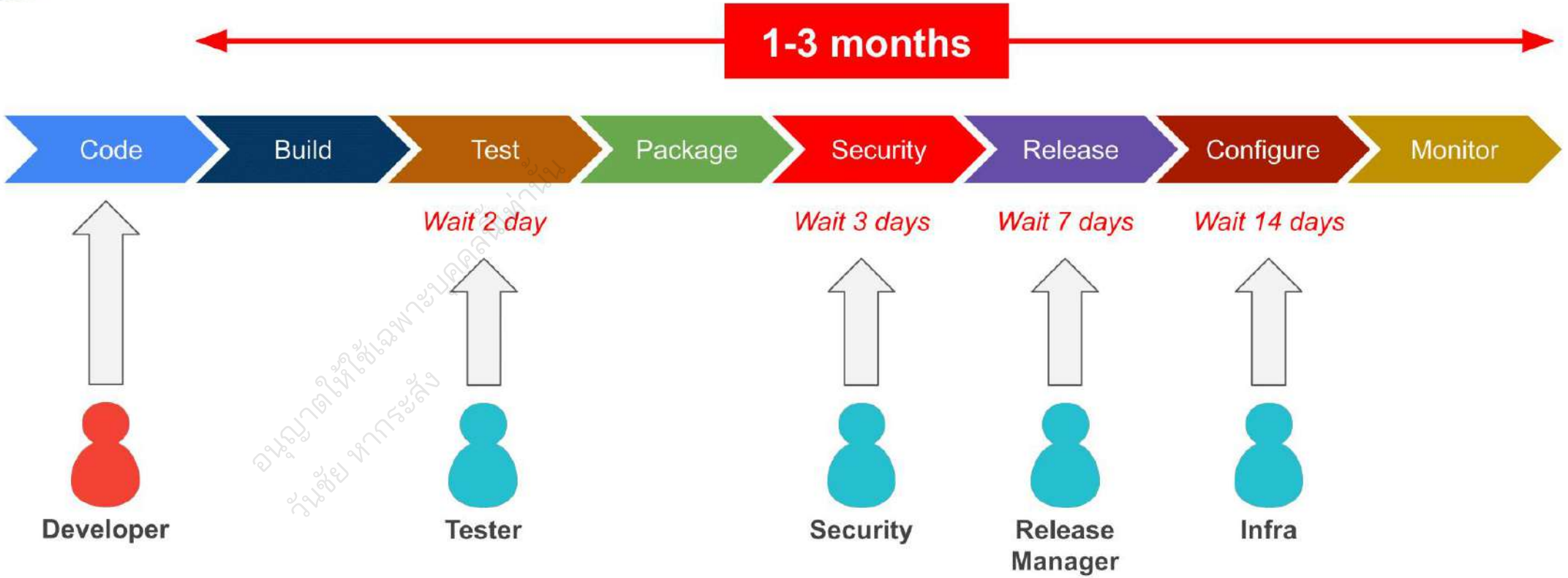
Pilot Project Sample Architecture



DevSecOps Metric Measurement

No	Metrics Name	Sample	October 2020	January 2021
1	Deployment frequency and rate	3-5 per day		
2	Cycle / Lead Time	30 mins		
3	Mean time to resolution (MTTR)	Half-day to few days		
4	Rollback rate (%)	30%		
5	Commit Rate / Volume Change	3-5 commit/person/day		
6	Number of defects on production (%)	20%		
7	Number of unsuccessful builds (%)	20%		

Sample Current Deploy Process



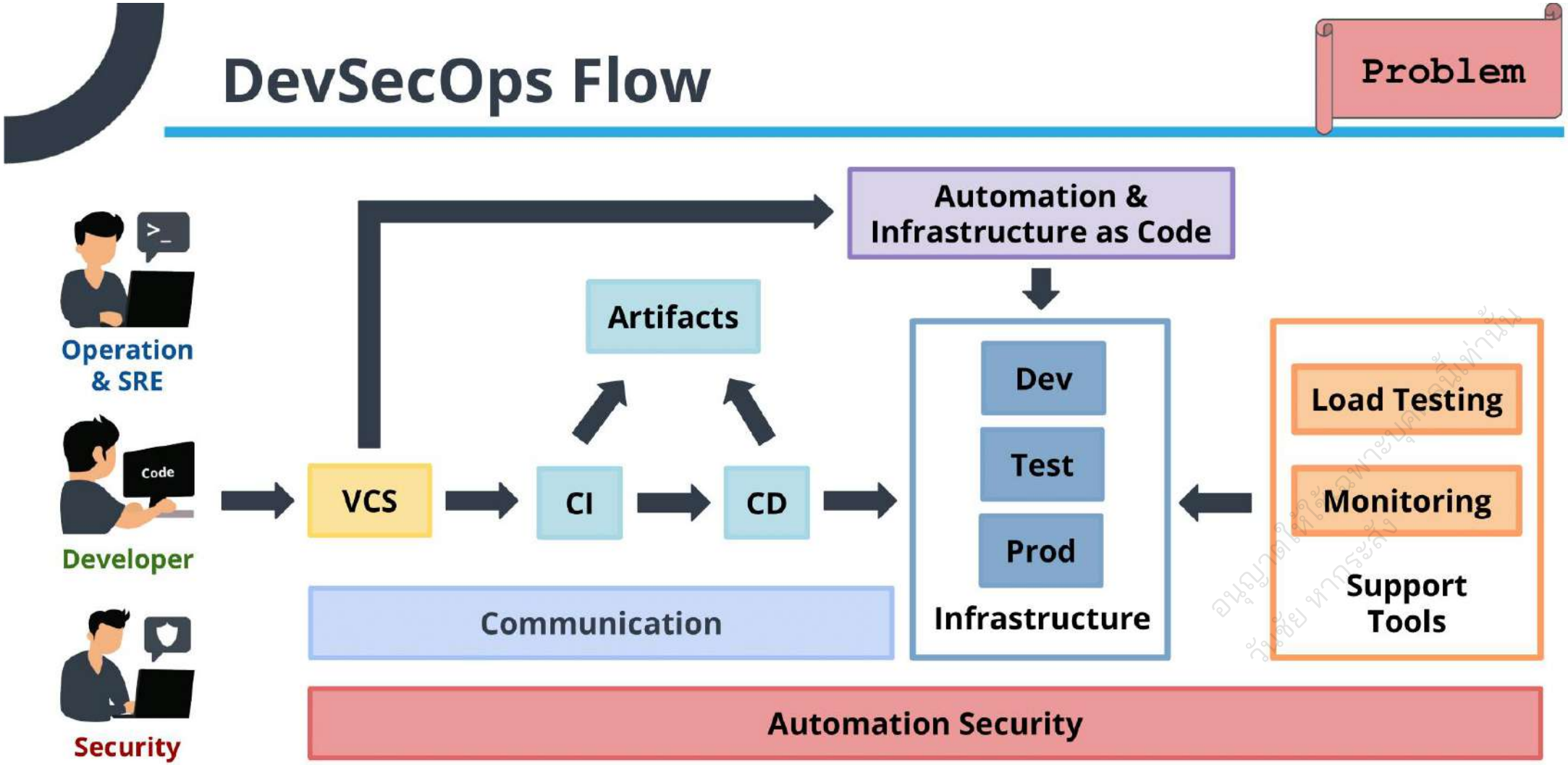
DevSecOps Design

Technologies

Skooldio

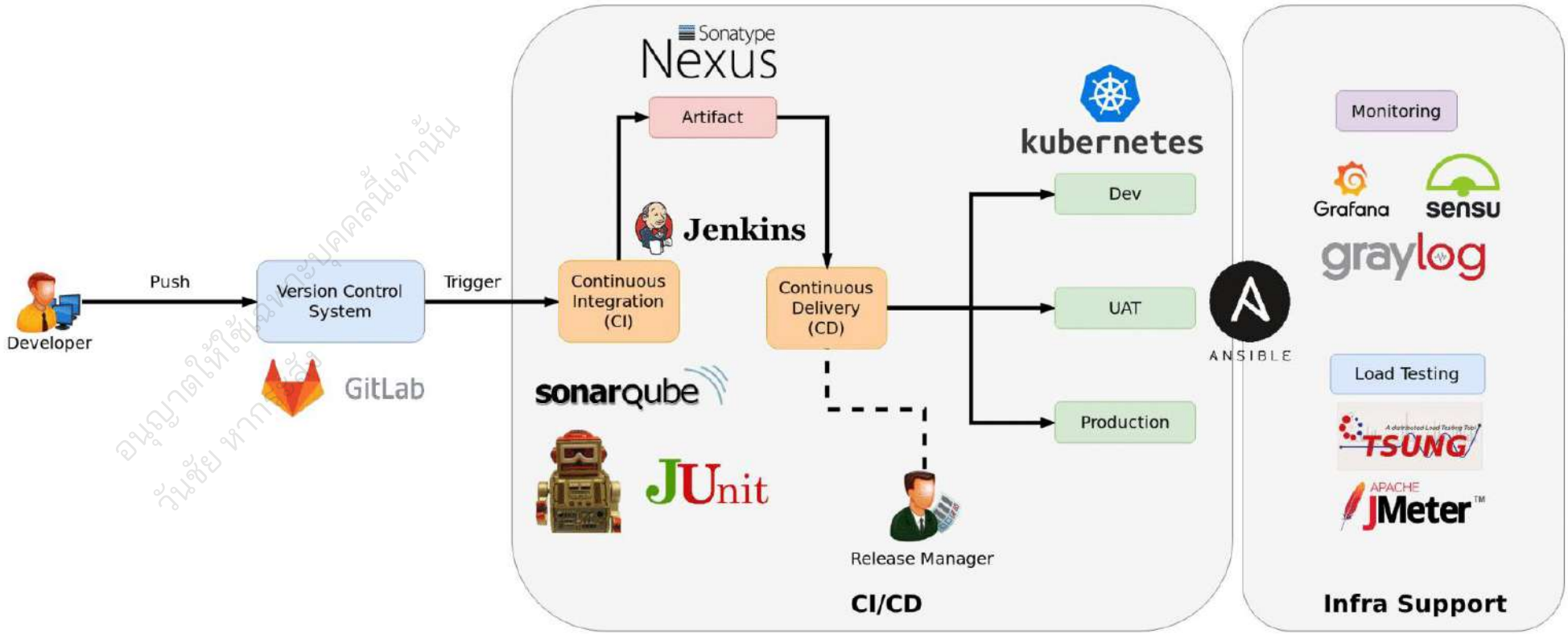
Opsta

DevSecOps Flow



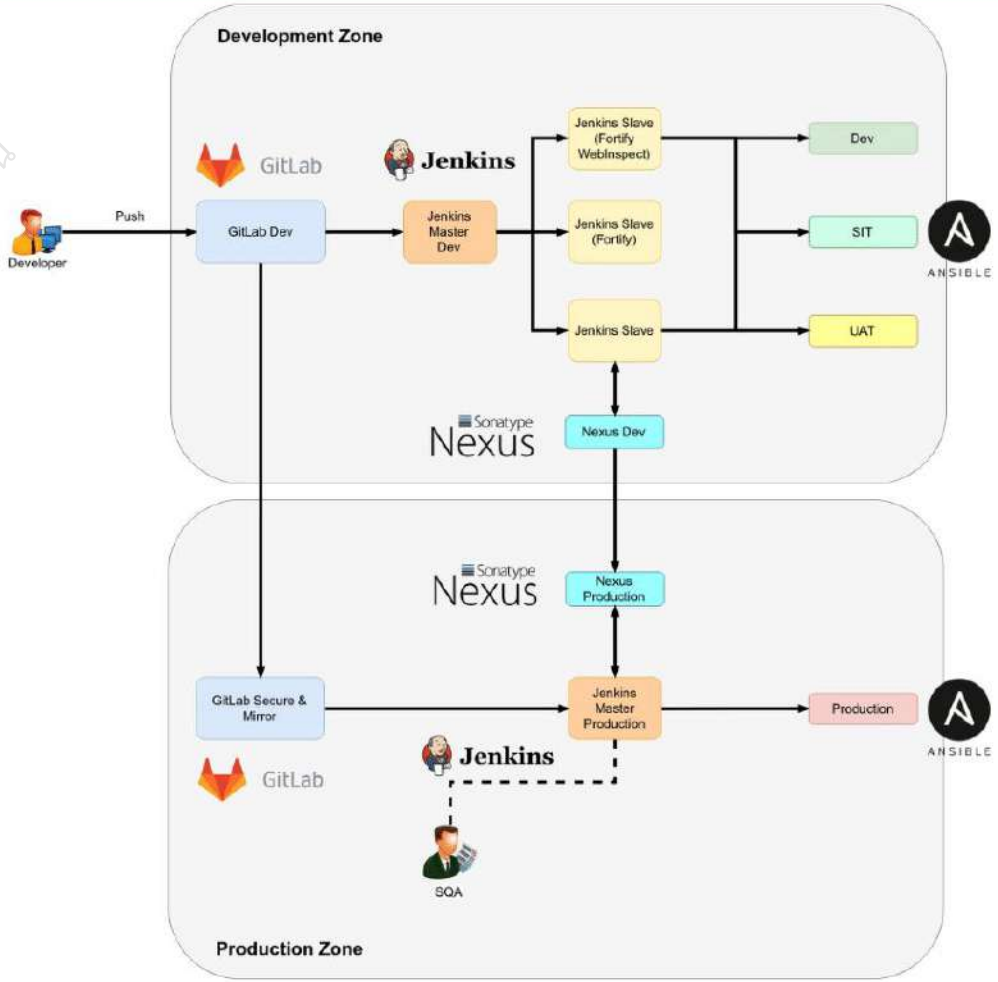
Sample Startup Architecture

Solution



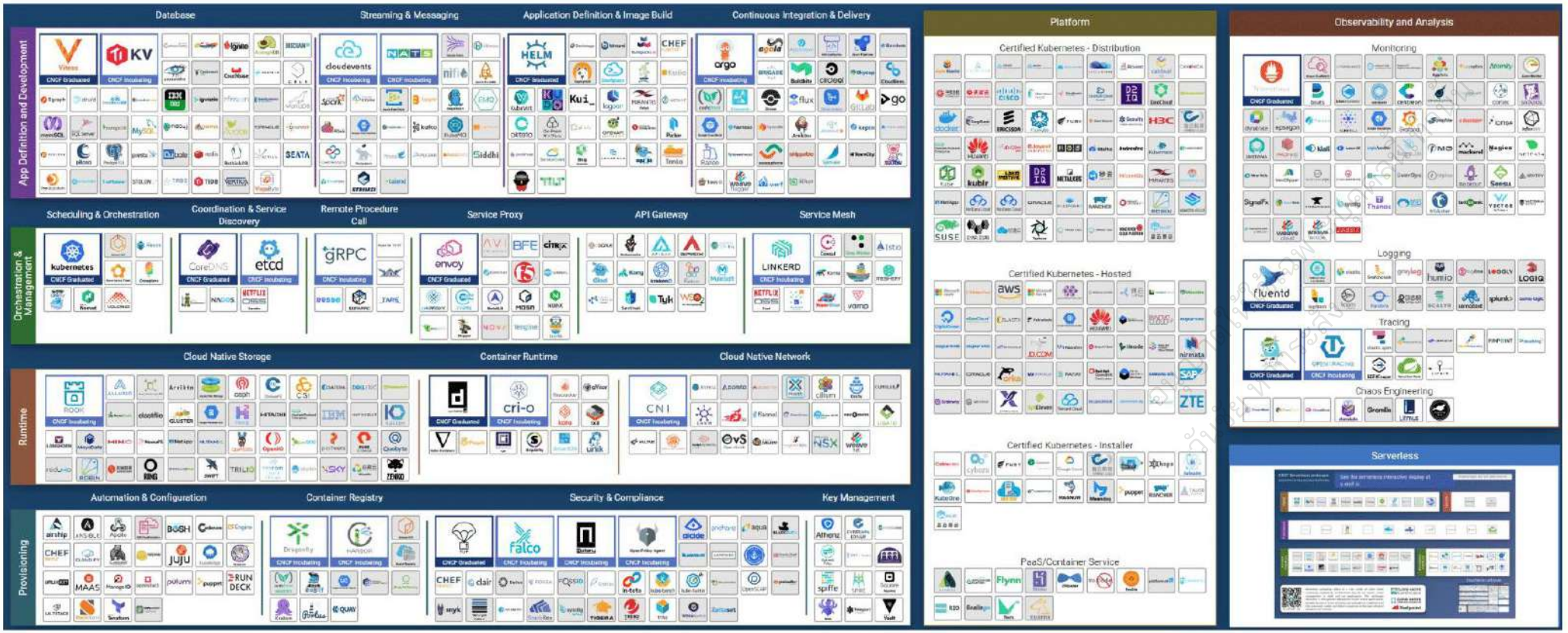
Sample Enterprise Architecture

Solution



Choose DevSecOps Technologies

Problem



https://landscape.cncf.io

Skooldio Opsta


Solution
Manifesto for Agile Software Development
Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

Choosing Git Branching Strategy

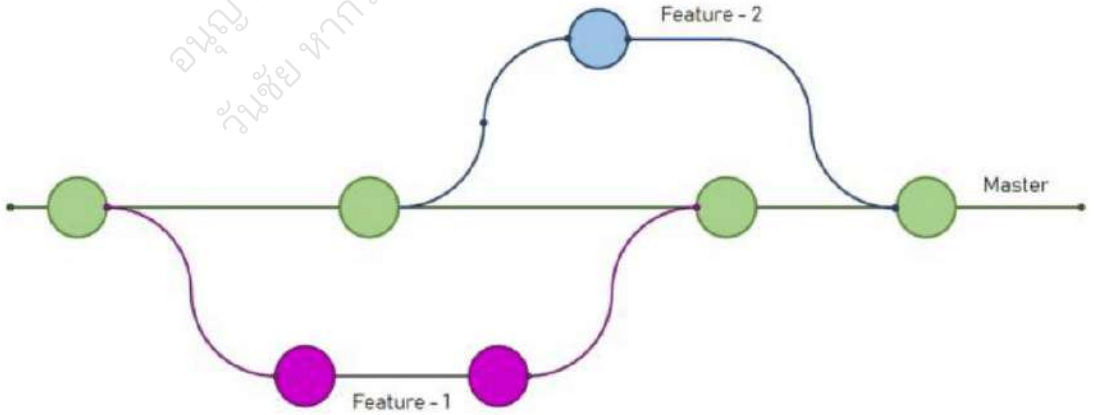
Problem

By Concept

- Environment Branch Strategy
- Feature Branch Strategy
- Scheduled Release Strategy

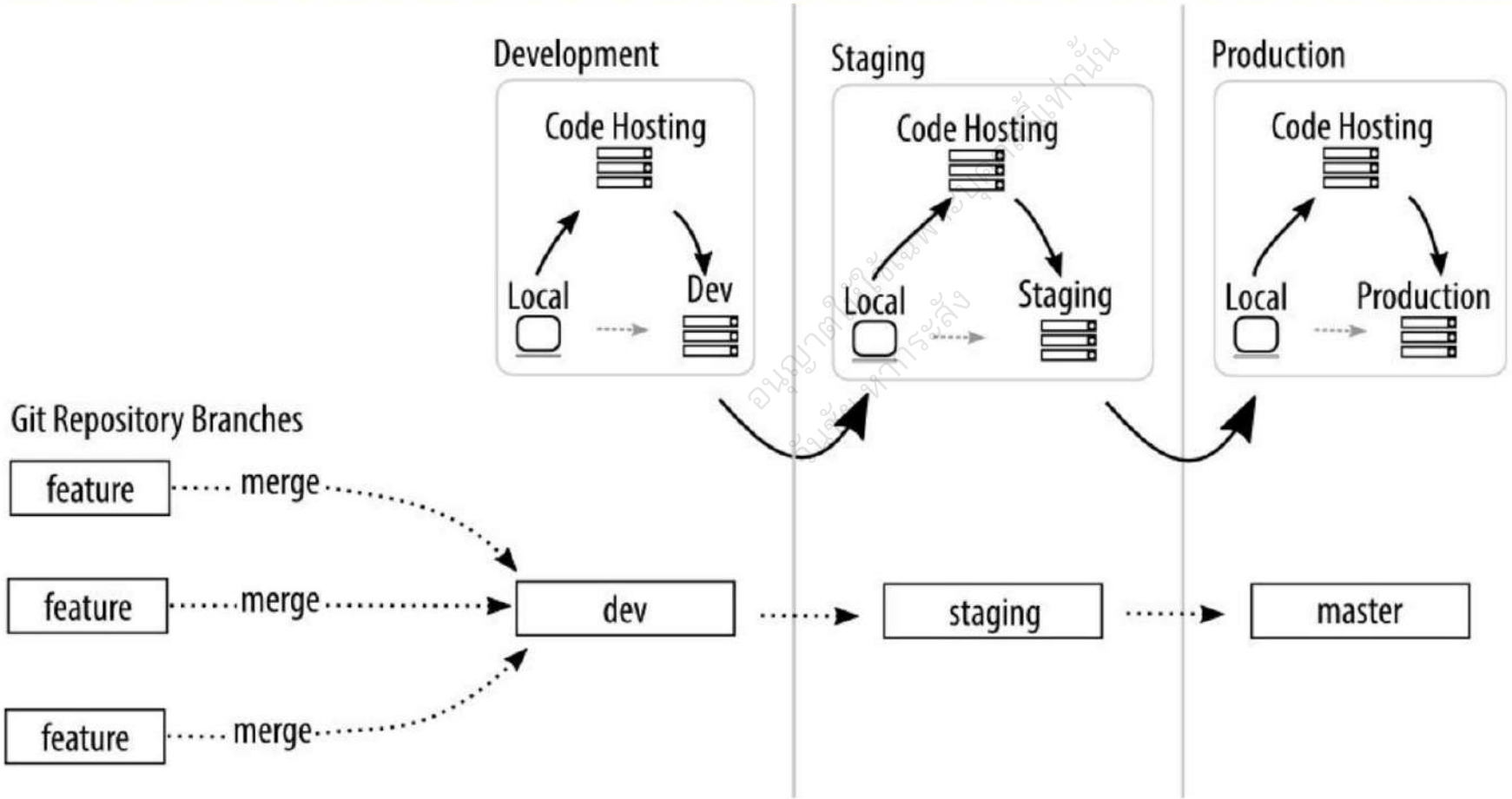
Full Proposing Idea

- GitLab Flow
- GitHub Flow
- Git Flow



Environment Branch Strategy

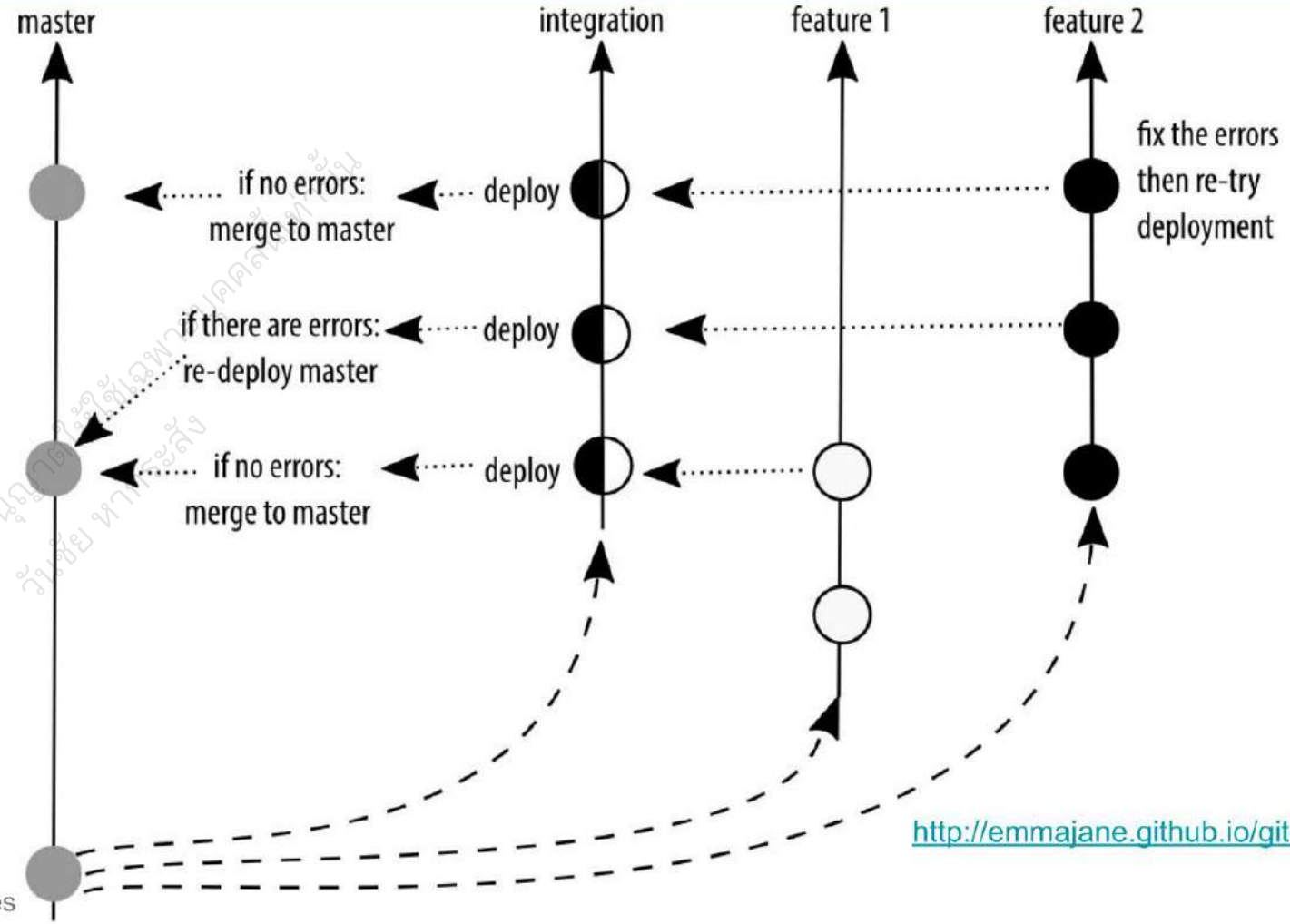
Solution



http://emmajane.github.io/git-adventure-mapping/#/4/5

Feature Branch

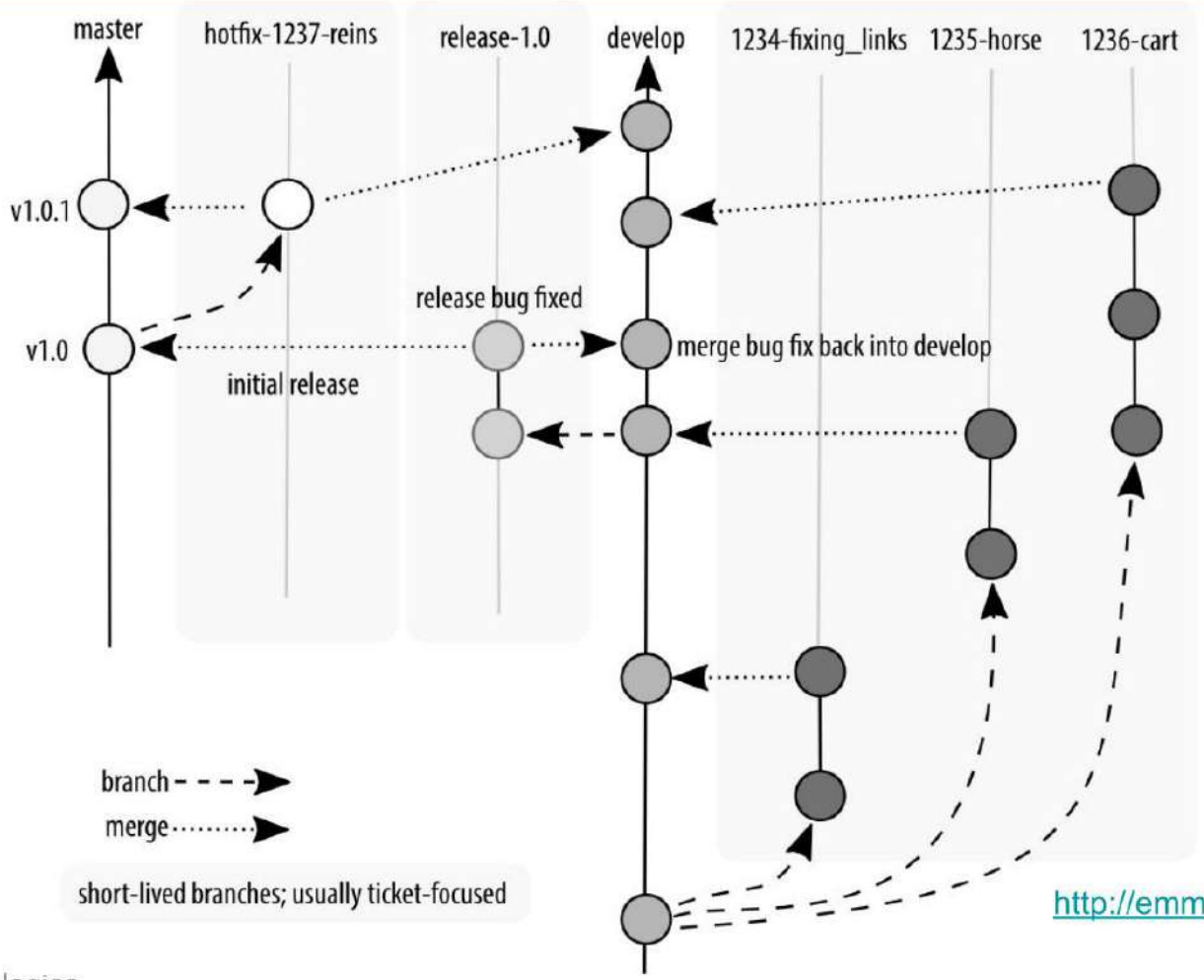
Solution



http://emmajane.github.io/git-adventure-mapping/#/4/6

Skooldio Opsta

Scheduled Release



http://emmajane.github.io/git-adventure-mapping/#/4/7

Deployment Strategies

Problem

- Recreate / Replace
- Rolling Deployment
- Blue-Green or Red-Black Deployment
- Canary Deployment

Compare Deployment Strategies

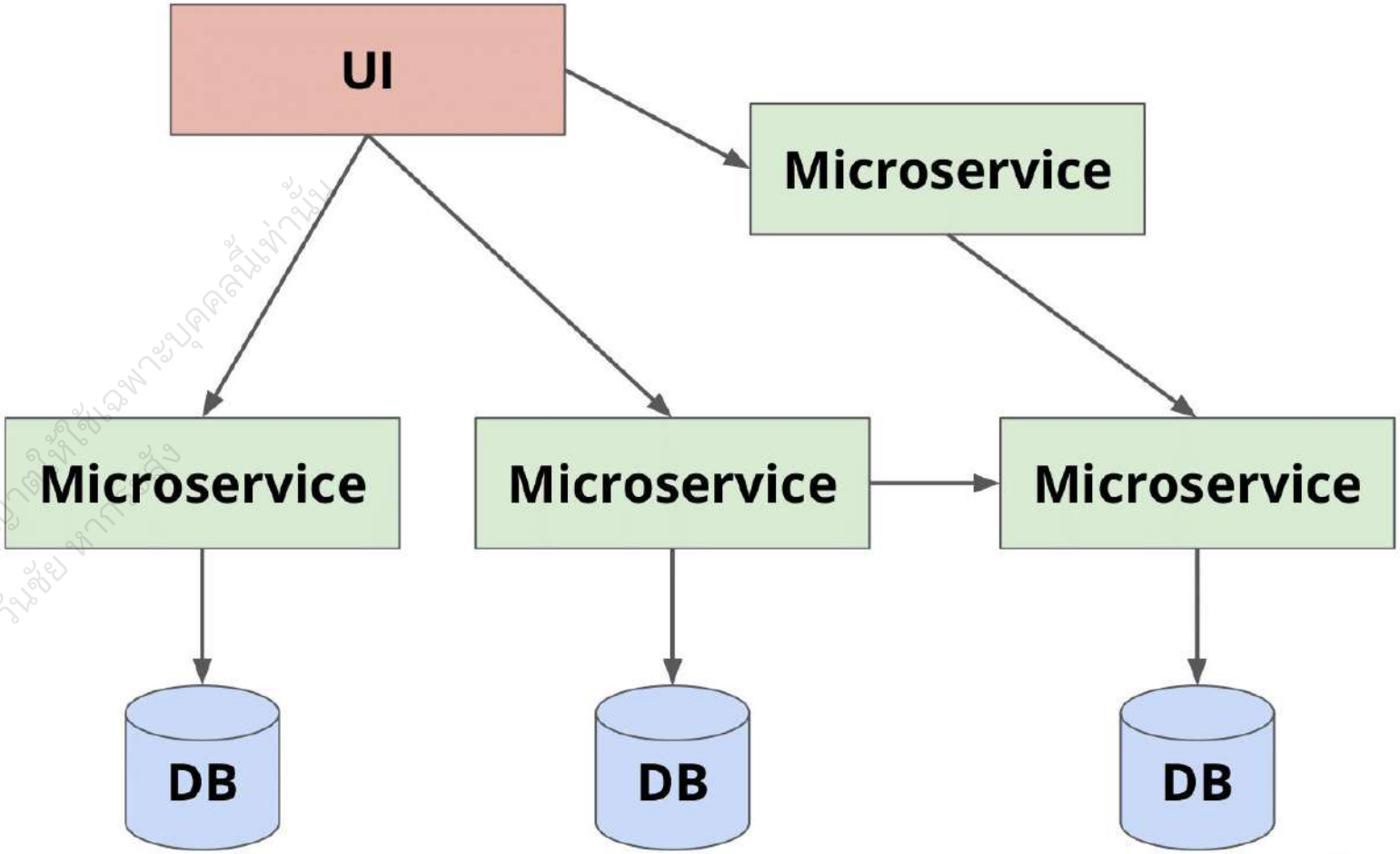
Solution

Strategy	Zero Downtime	Targeted Users	Cost	Rollback Time	Impact on User	Setup Complexity
Recreate	✗	✗	★	★★★	★★★	★
Rolling	✓	✗	★	★★★	★	★★
Blue-Green	✓	✗	★★★	★	★	★★★
Canary	✓	✓	★★	★★	★	★★★

Problem

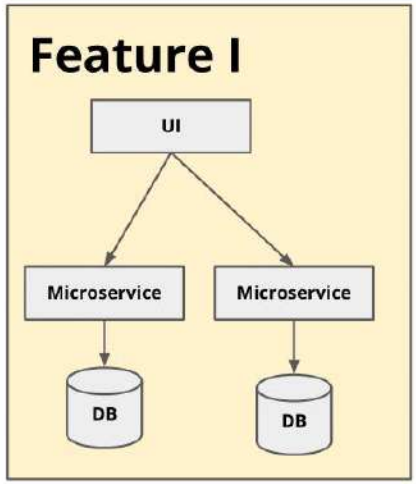
Microservices

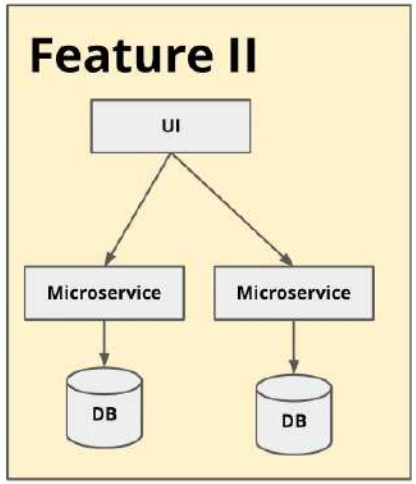
Problem

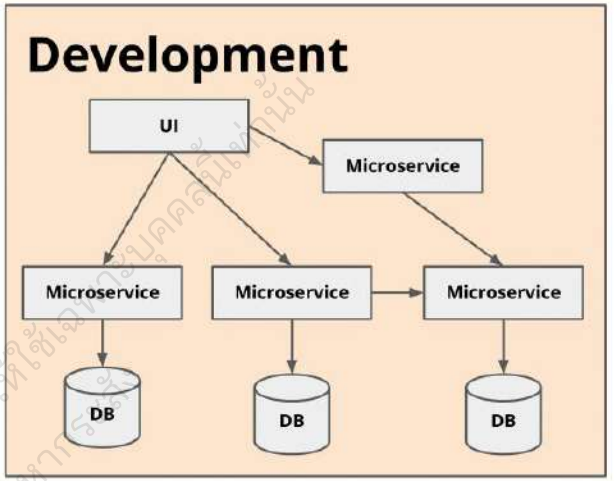


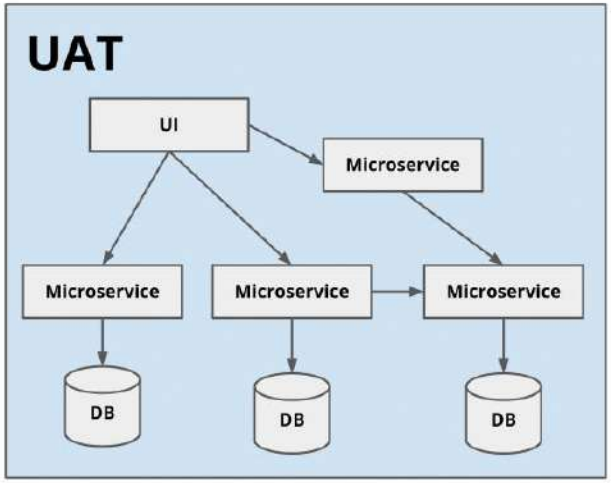
I need...

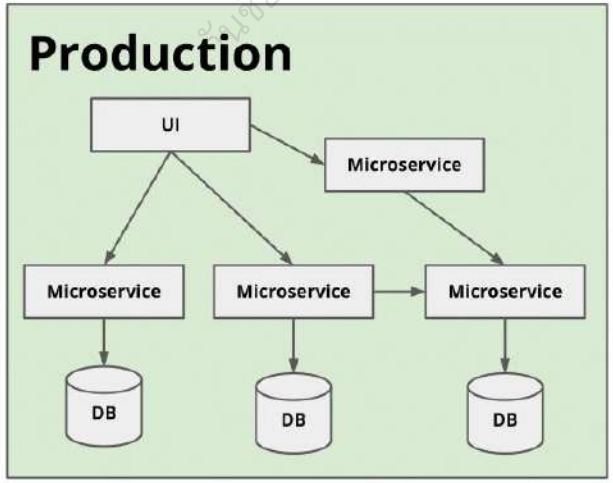
Problem



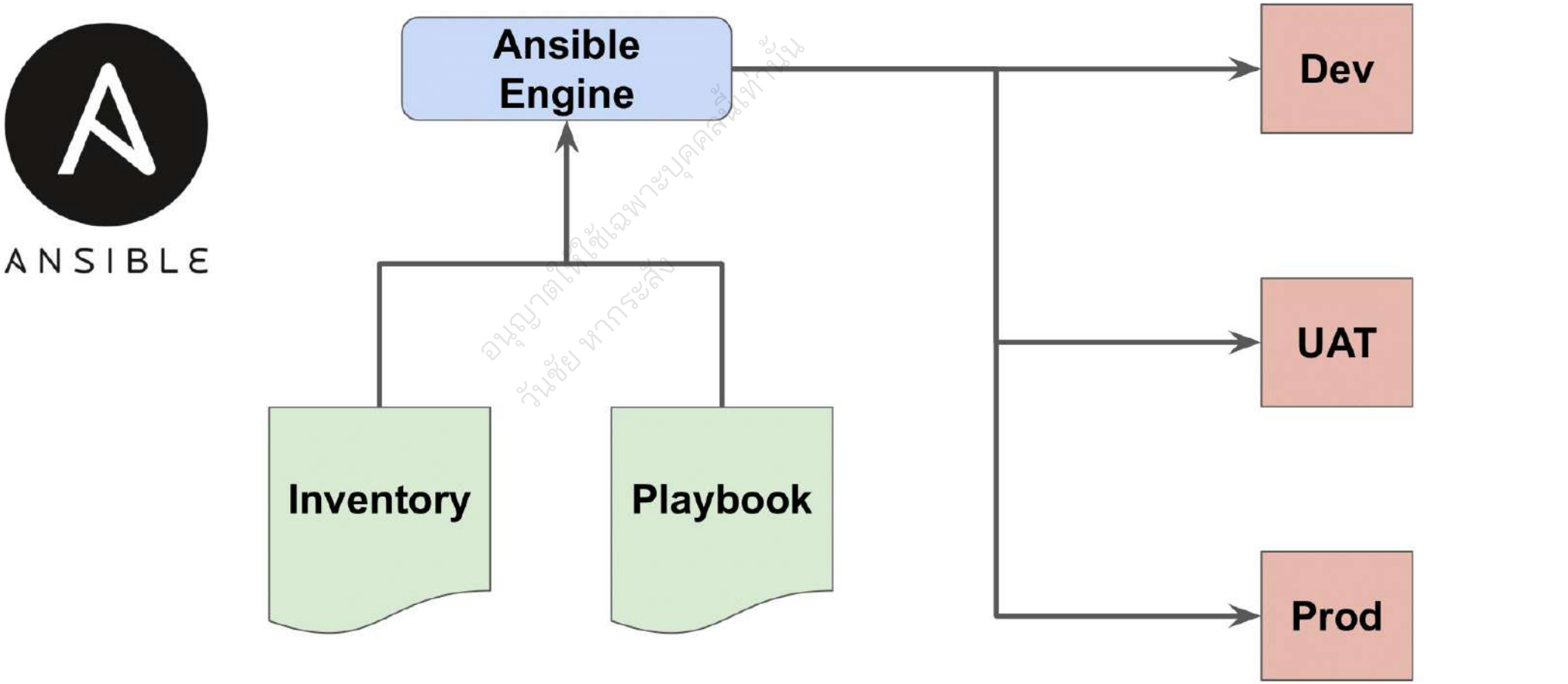






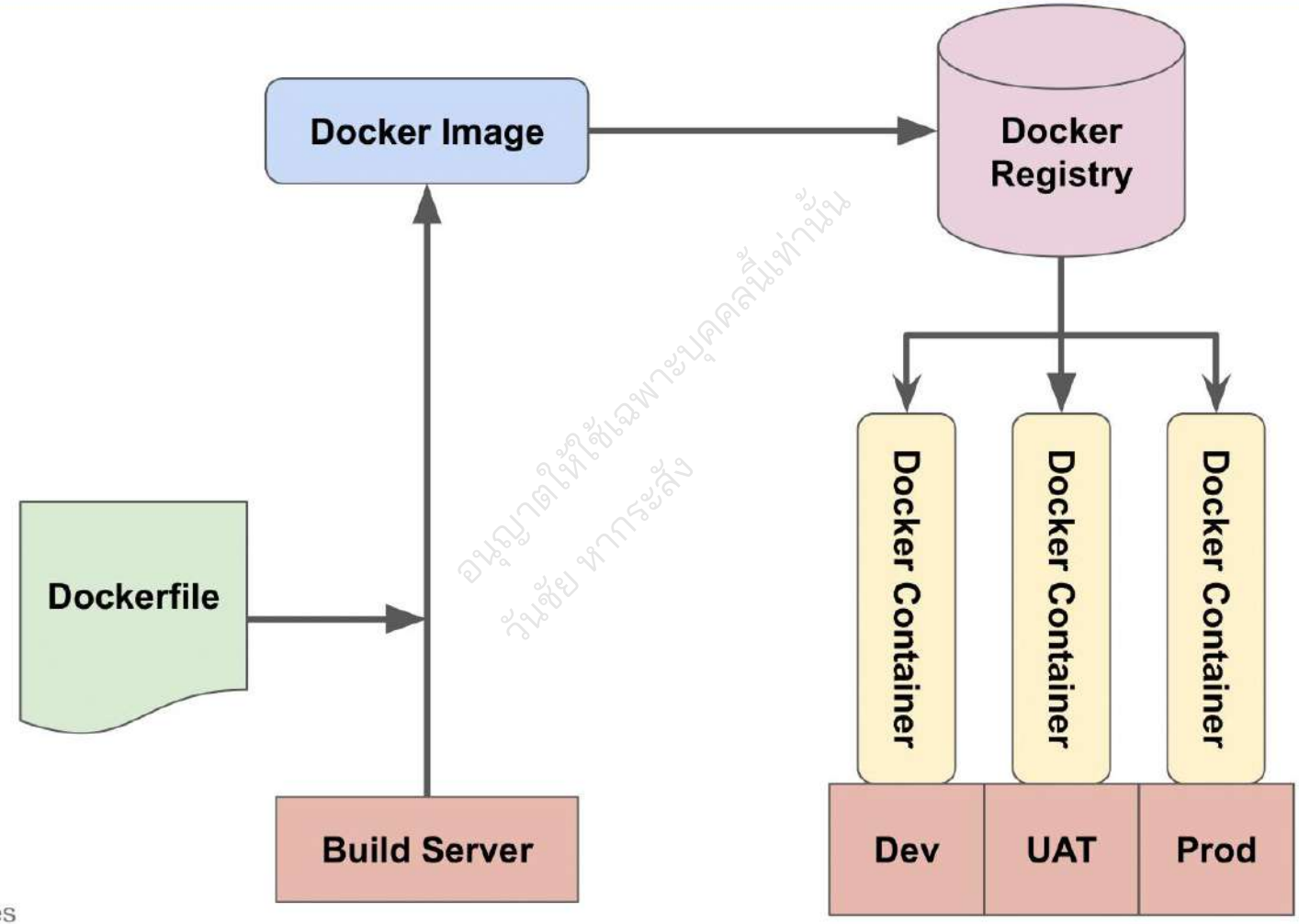


Ansible Provisioning



Docker, build once run anywhere

Solution



The Twelve-Factor App (1)

Also known as **Cloud Native Application**

- **Codebase**
  One codebase tracked in revision control, many deploys
- **Dependencies**
  Explicitly declare and isolate dependencies
- **Config**
  Store config in the environment
- **Backing services**
  Treat backing services as attached resources
- **Build, release, run**
  Strictly separate build and run stages
- **Processes**
  Execute the app as one or more stateless processes

https://12factor.net/

Skooldio Opsta

The Twelve-Factor App (2)

- **Port binding**

  Export services via port binding
- **Concurrency**

  Scale out via the process model
- **Disposability**

  Maximize robustness with fast startup and graceful shutdown
- **Dev/prod parity**

  Keep development, staging, and production as similar as possible
- **Logs**

  Treat logs as event streams
- **Admin processes**

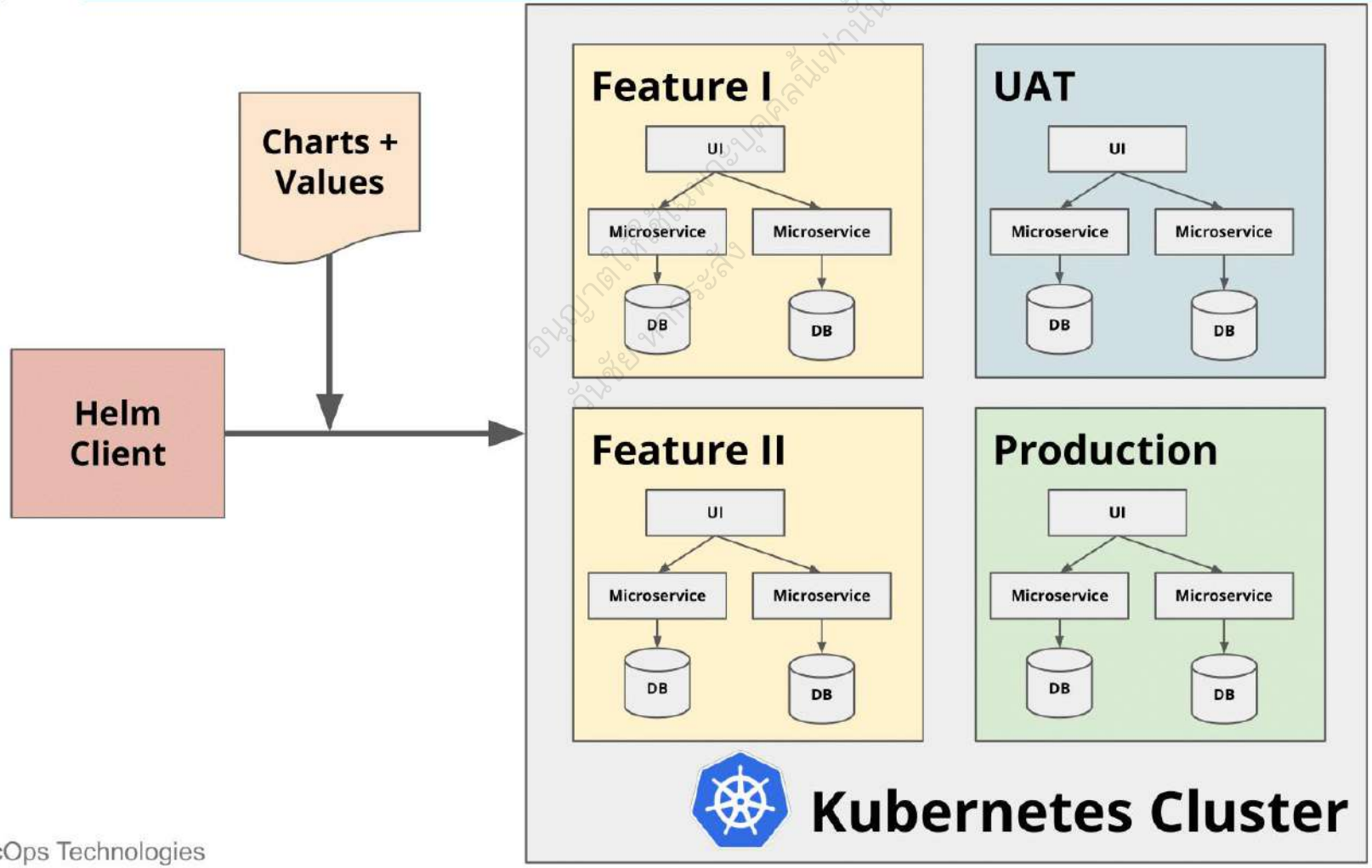
  Run admin/management tasks as one-off processes

https://12factor.net/

Skooldio Opsta

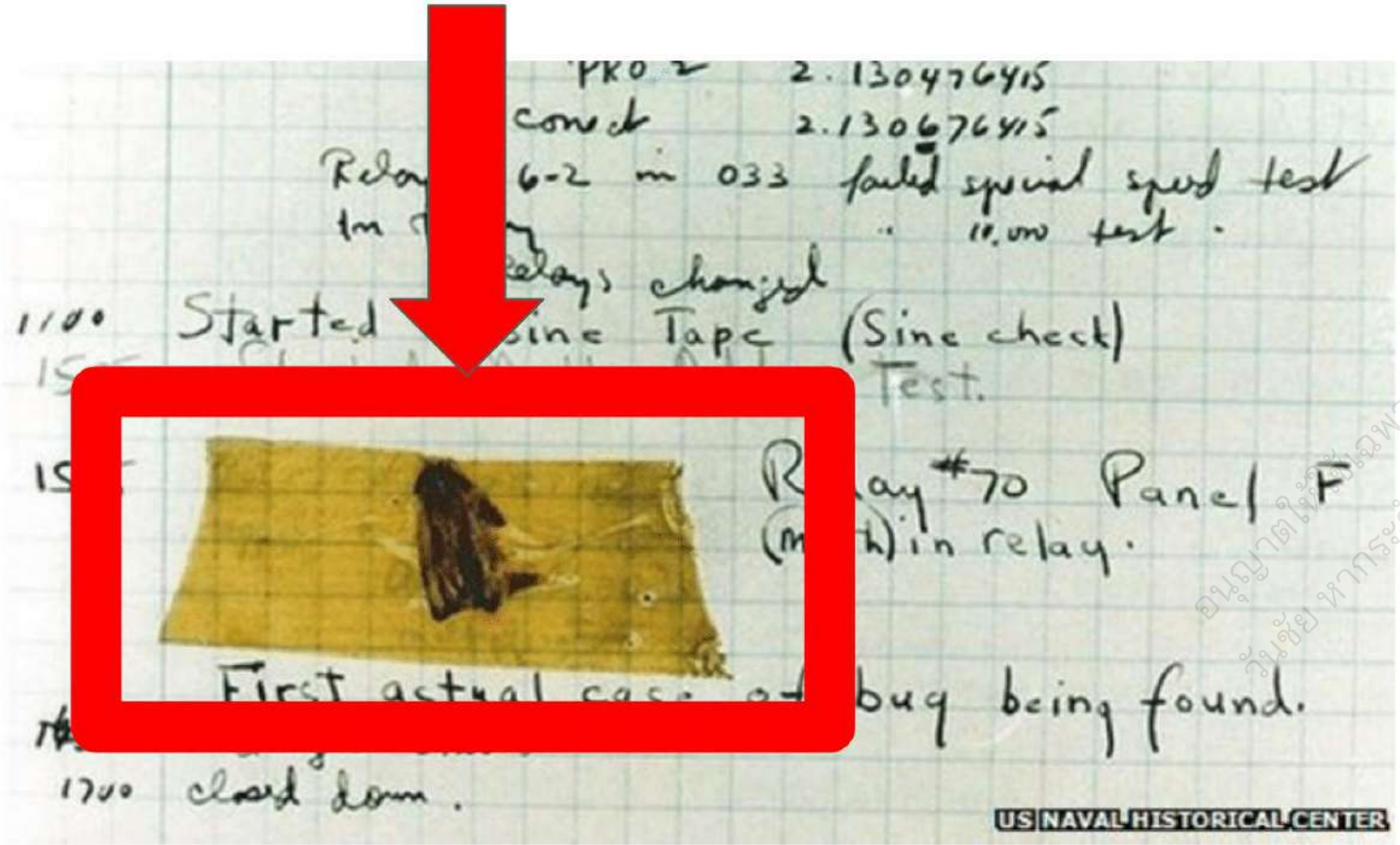
Helm + Kubernetes

Solution



This is a BUG

Problem



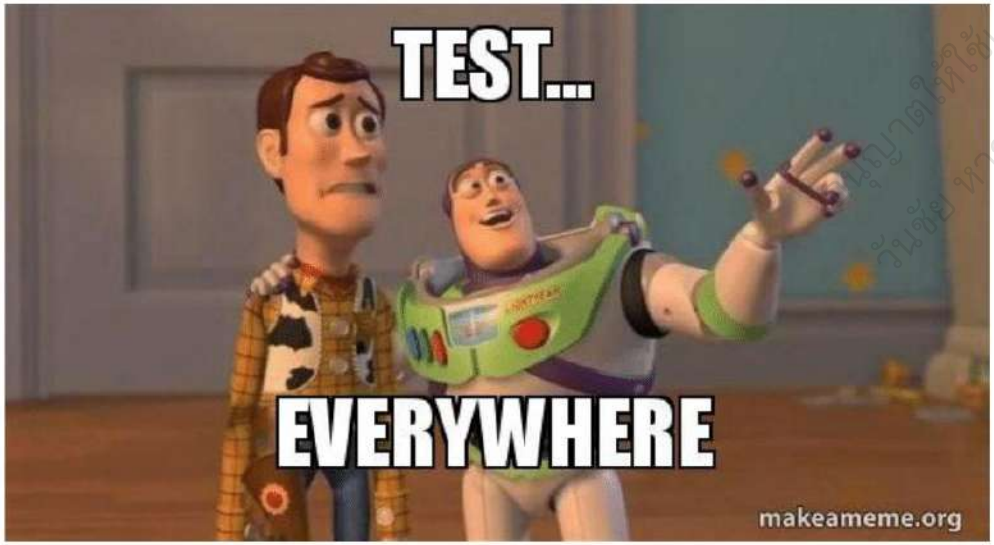
Unit Test

Integration Test

Problem

Performance Test

Acceptance Test



Review Code

Solution

Jenkinsfile

Edit | View file @ 8ca8c007

```
@@ -6,18 +6,19 @@ properties([
  ])
])

-def label = "petclinic-${UUID.randomUUID().toString()}"
-podTemplate(label: label, cloud: 'kubernetes', containers: [
+def label = "petclinic"
+podTemplate(label: label, cloud: 'kubernetes', idleMinutes: 360, containers: [
  // Don't use alpine version. It having problem with forking JVM such as running surefire and junit testing
-  containerTemplate(name: 'java', image: 'openjdk:8u171-jdk-stretch', ttyEnabled: true, command: 'cat'),
-  containerTemplate(name: 'docker', image: 'docker', ttyEnabled: true, command: 'cat'),
-  containerTemplate(name: 'helm', image: 'lachlanevenson/k8s-helm', ttyEnabled: true, command: 'cat'),
-  containerTemplate(name: 'git', image: 'paasmule/curl-ssl-git', ttyEnabled: true, command: 'cat'),
-  containerTemplate(name: 'jmeter', image: 'opsta/jmeter', ttyEnabled: true, command: 'cat'),
+  containerTemplate(name: 'java', image: 'openjdk:8u181-jdk-stretch', ttyEnabled: true, command: 'cat'),
+  containerTemplate(name: 'docker', image: 'docker:18.06.1-ce', ttyEnabled: true, command: 'cat'),
+  containerTemplate(name: 'helm', image: 'lachlanevenson/k8s-helm:v2.10.0', ttyEnabled: true, command: 'cat'),
+  containerTemplate(name: 'git', image: 'paasmule/curl-ssl-git:latest', ttyEnabled: true, command: 'cat'),
+  containerTemplate(name: 'jmeter', image: 'opsta/jmeter:latest', ttyEnabled: true, command: 'cat'),
  containerTemplate(name: 'robot', image: 'ppodgorsek/robot-framework:3.2.0', ttyEnabled: true, command: 'cat')
],
volumes: [
  hostPathVolume(mountPath: '/var/run/docker.sock', hostPath: '/var/run/docker.sock'),
+  hostPathVolume(mountPath: '/root/.m2', hostPath: '/tmp/jenkins/.m2')
```

Automated Test with CI/CD

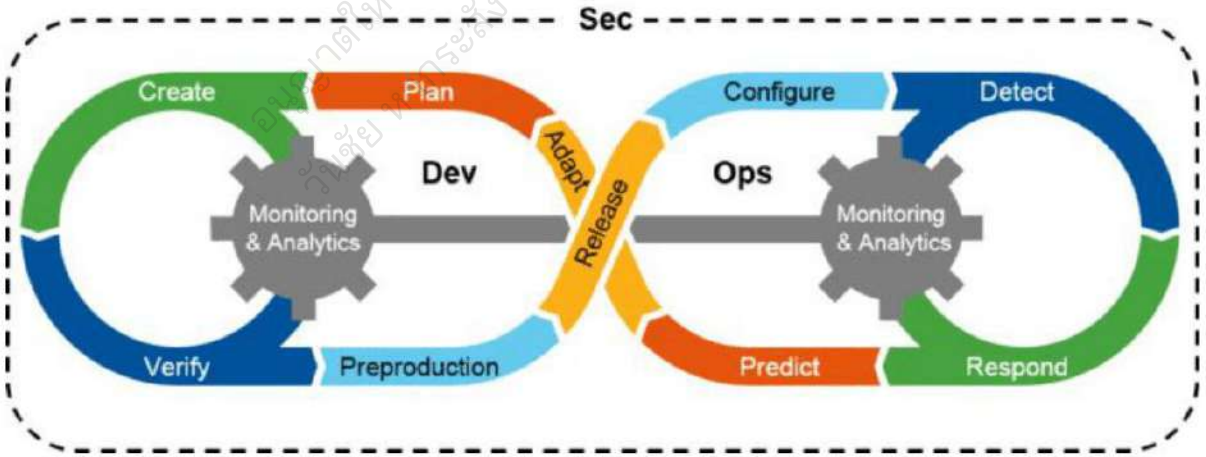
Solution

Run Unit Test	SonarQube analysis	Quality Gate	Build Artifact	Build docker image and push to registry	Deploy Application	Run User Acceptance Test	Run Performance Test
1min 9s	33s	371ms	29s	10s	10s	41s	50s
1min 9s	33s	450ms (paused for 1s)	29s	10s	11s	42s	50s
1min 8s	33s	158ms (paused for 1s)	29s	9s	9s	40s failed	
1min 10s	34s	507ms (paused for 1s)	30s	10s	9s	42s failed	

Problem

Code Smells	Bugs	Vulnerabilities

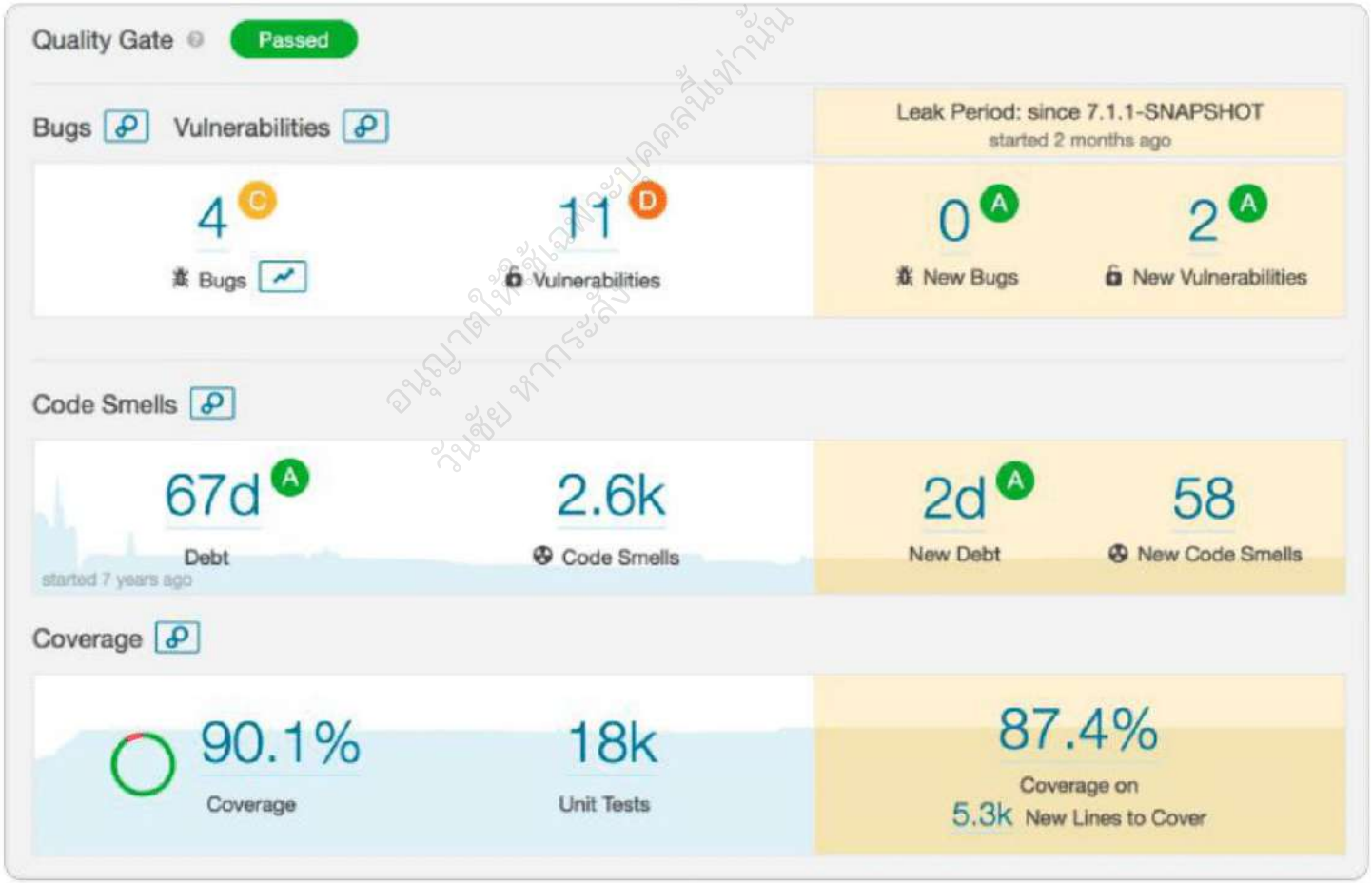
DevSecOps



Skooldio Opsta

Code Analysis

Solution



Skooldio Opsta

Automated Security Check

Solution

DependencyCheck Result

Warnings Trend

All Warnings	New Warnings	Fixed Warnings
153	138	0

Summary

Total	High Priority	Normal Priority	Low Priority
153	24	111	18

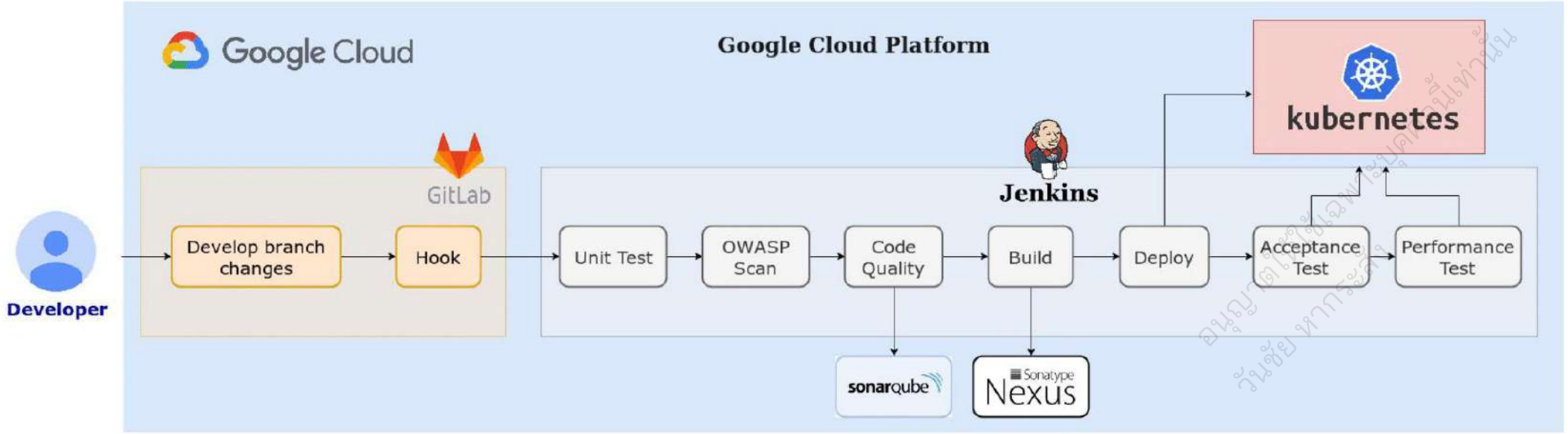
Details

Files | **Categories** | Types | Warnings | Details | New | High | Normal | Low

Category	Total	Distribution
CWE-119 Improper Restriction of Operations within the Bounds of a Memory Buffer	5	
CWE-134 Uncontrolled Format String	1	
CWE-189 Numeric Errors	2	
CWE-20 Improper Input Validation	7	
CWE-200 Information Exposure	5	
CWE-22 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	4	
CWE-264 Permissions, Privileges, and Access Controls	4	
CWE-287 Improper Authentication	2	
CWE-310 Cryptographic Issues	2	
CWE-399 Resource Management Errors	7	
CWE-59 Improper Link Resolution Before File Access ('Link Following')	4	
CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	14	
CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	2	
CWE-94 Improper Control of Generation of Code ('Code Injection')	10	
Total	**153**	

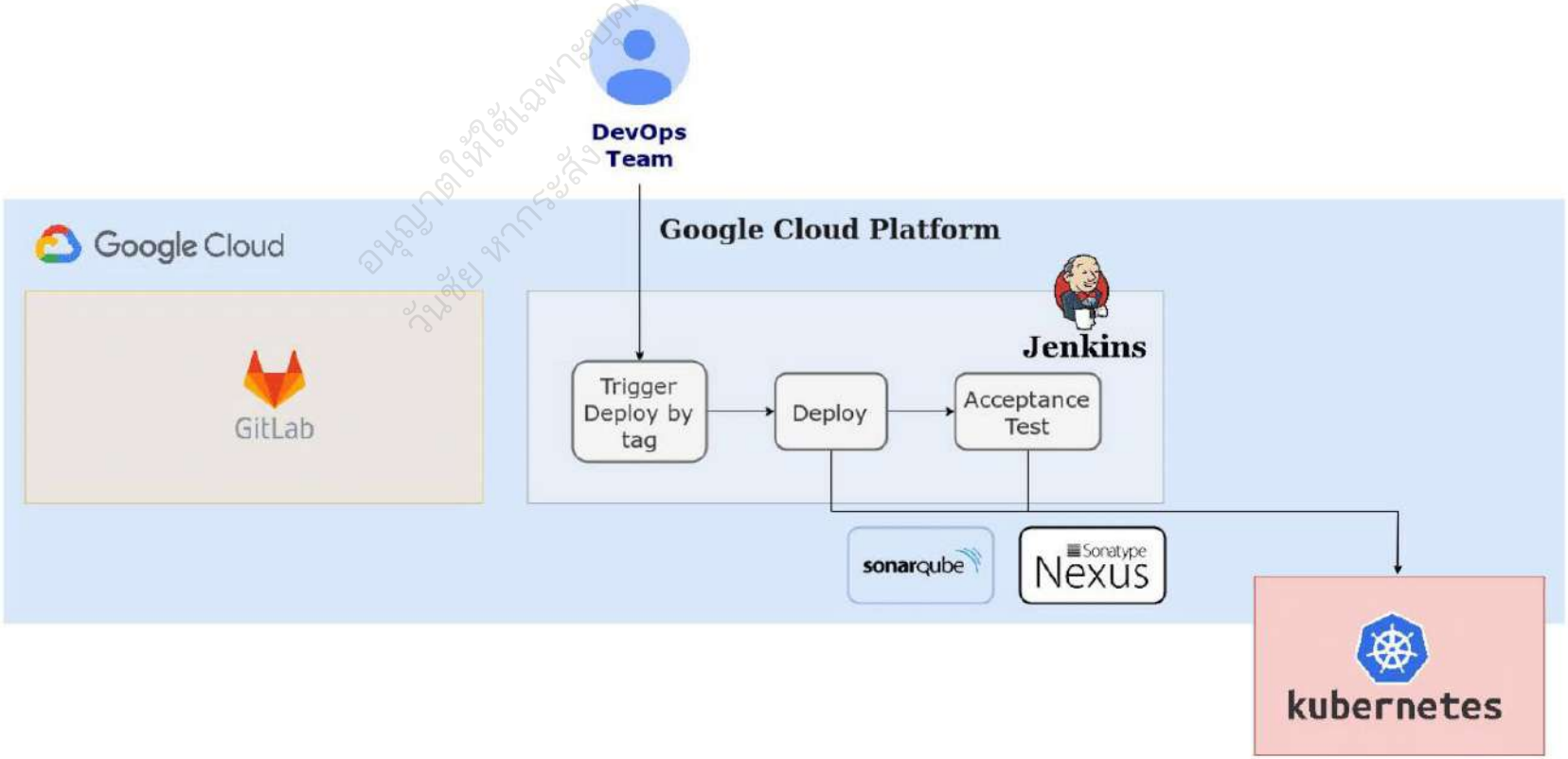
DevSecOps Process and Policy (1)

Development Environment



DevSecOps Process and Policy (2)

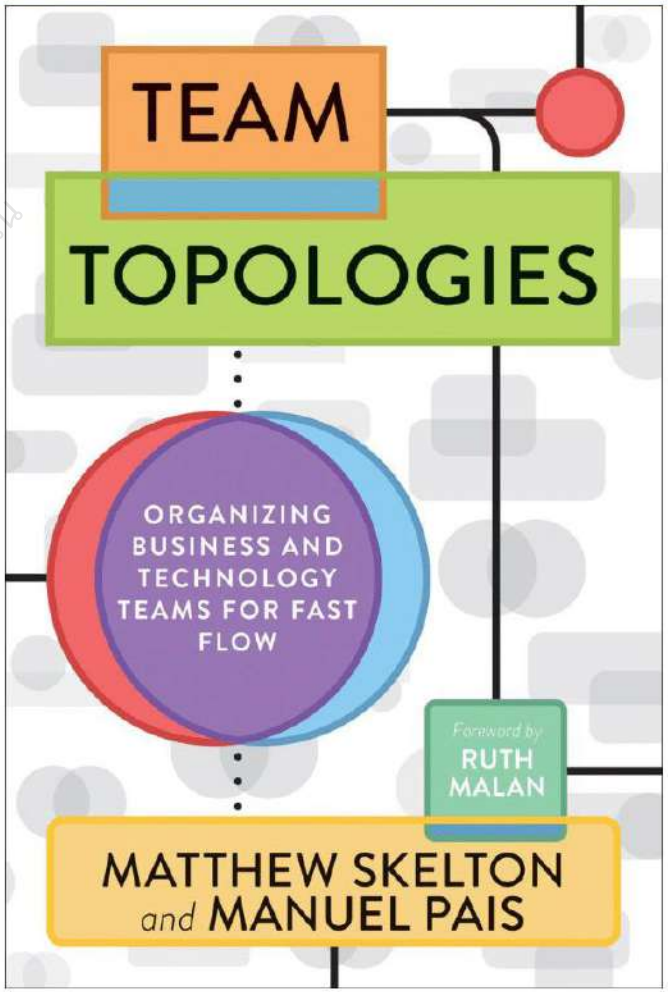
Production Environment

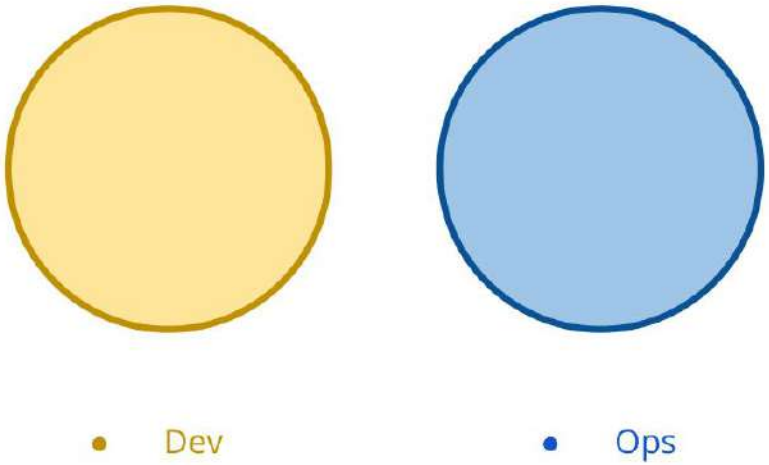


Skooldio Opsta

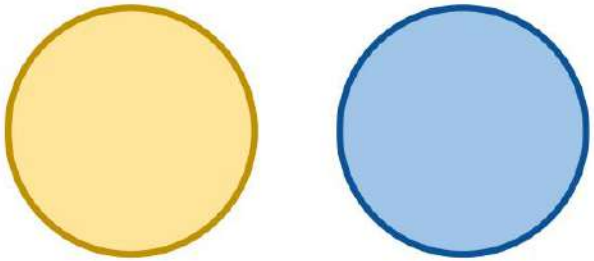
People

Team Topologies

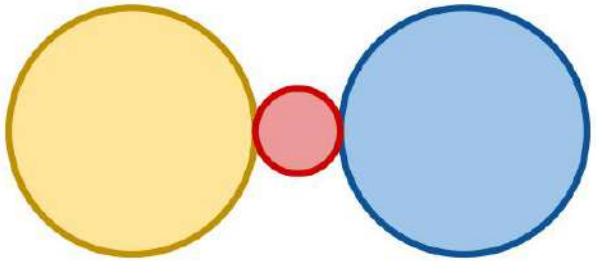




DevOps Anti-Types

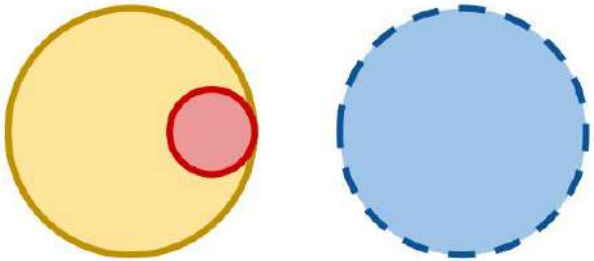


Dev and Ops Silos

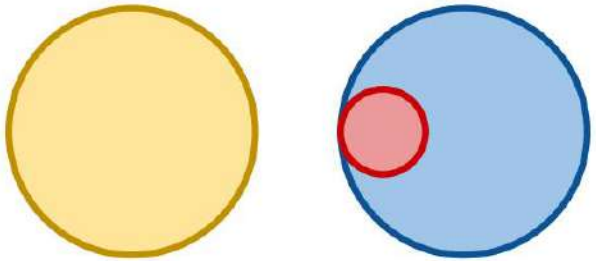


DevOps Team Silo

- Dev
- DevOps
- Ops

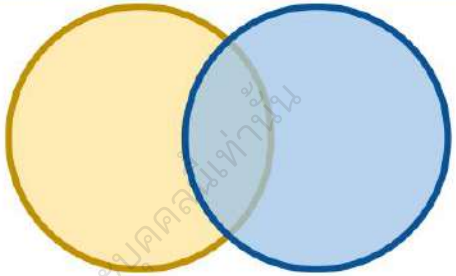


Dev Don't Need Ops

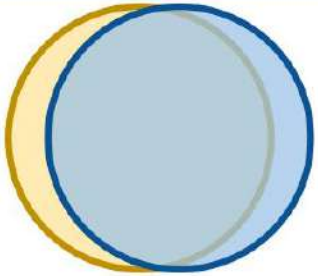


Rebranded SysAdmin

DevOps Team Topologies

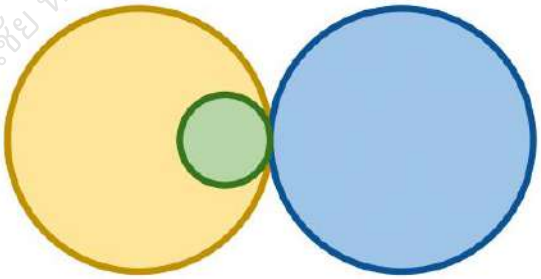


Dev and Ops Collaboration

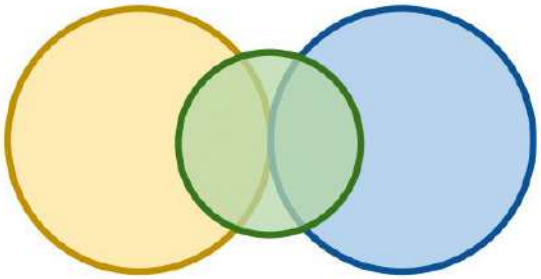


Fully Shared Ops Responsibilities

- Dev
- DevOps
- Ops



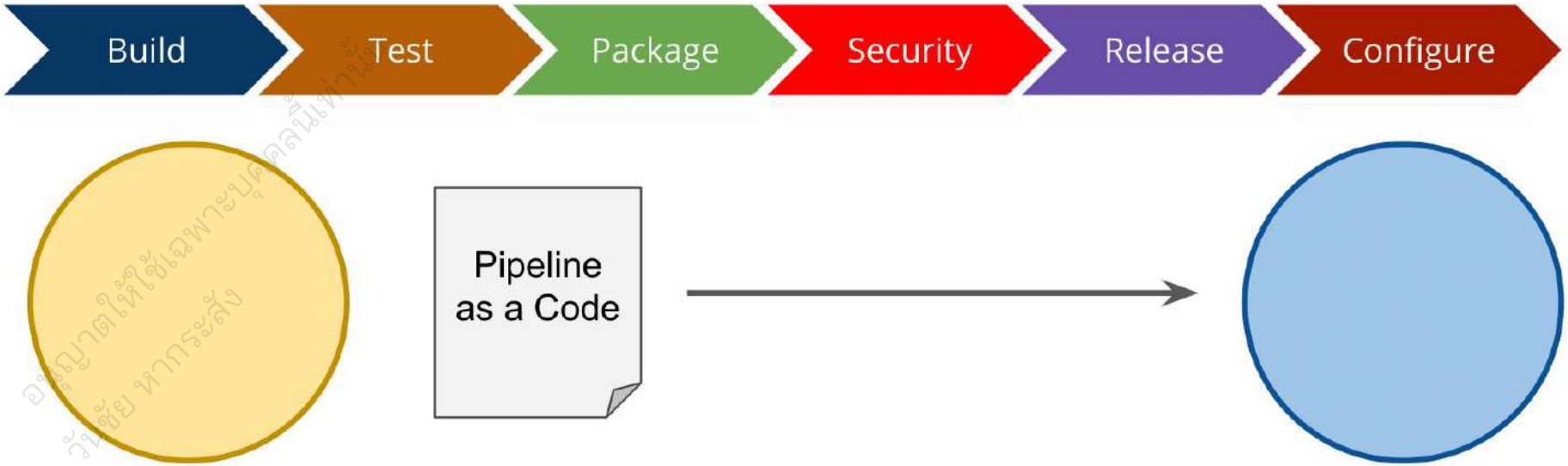
Ops as Infrastructure-as-a-Service (Platform)



DevOps Advocacy Team

Skooldio Opsta

CI/CD Responsibility



Developer

- Structure
- Build
- Test
- Deploy

Operation

- Review
- Optimize
- Security

Site Reliability Engineering (SRE)

Skooldio

Opsta

Site Reliability Engineering (SRE)

SRE

A set of **practices** with an emphasis of strong engineering capabilities that **implement** the DevSecOps **practices**, and sets a job role + team

DevSecOps

A set of **principles & culture guidelines** that helps to **breakdown** the **silos** between development and operations / networking / security

SRE Implements DevSecOps

Site Reliability Engineers spend less than 50% of their time performing operational work to allow them to spend more time on **improving infrastructure** and **task automation**

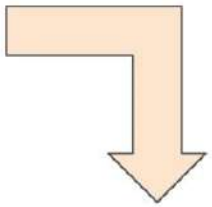
SRE is good for **product-based**

Service Levels

Product Management with SREs define service levels for the systems as part of product design

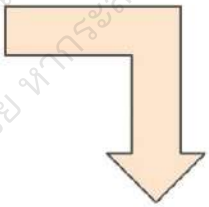
Service Level Indicator (SLI)

- Quantitative measure of an attribute of the service, such as throughput, latency
- A directly measurable & observable by the users – not an internal metric (response time vs. CPU utilization)
- Could be a way to represent the user's experience



Service Level Objective (SLO)

- A threshold beyond which an improvement of the service is required
- The point at which the users may consider opening up support ticket, the "pain threshold", e.g., YouTube buffering
- Driven by business requirements, not just current performance



Service Level Agreement (SLA)

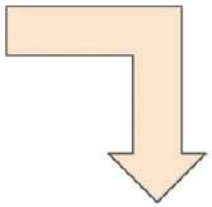
- A business contract the service provider + the customer • Loss of service could be related to loss of business
- Typically, an SLA breach would mean some form of compensation to the client

Sample Service Levels

Product Management with SREs define service levels for the systems as part of product design

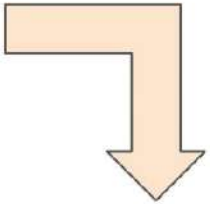
Service Level Indicator (SLI)

- The latency of successful HTTP responses (HTTP 200)



Service Level Objective (SLO)

- The latency of 95% of the responses must be less than 200ms



Service Level Agreement (SLA)

- Customer compensated if the 95th percentile latency exceeds 300ms

Error Budget

Error Budgets tend to bring balance between SRE and application development by mitigating risks. An Error Budget is unaffected until the system availability will fall within the SLO. The Error Budget can always be adjusted by managing the SLOs or enhancing the IT operability. The ultimate goal remains application reliability and scalability.

Availability level	Downtime per year	Downtime per quarter	Downtime per month	Downtime per week	Downtime per day	Downtime per hour
90%	36.52 days	9.13 days	3.04 days	16.80 hours	2.40 hours	6.00 minutes
95%	18.26 days	4.57 days	1.52 days	8.40 hours	1.20 hours	3.00 minutes
99%	3.65 days	21.91 hours	7.30 hours	1.68 hours	14.40 minutes	36.00 seconds
99.5%	1.83 days	10.96 hours	3.65 hours	50.40 minutes	7.20 minutes	18.00 seconds
99.9%	8.77 hours	2.19 hours	43.83 minutes	10.08 minutes	1.44 minutes	3.60 seconds
99.95%	4.38 hours	1.10 hours	21.91 minutes	5.04 minutes	43.20 seconds	1.80 seconds
99.99%	52.59 minutes	13.15 minutes	4.38 minutes	1.01 minutes	8.64 seconds	0.36 seconds
99.999%	5.26 minutes	1.31 minutes	26.30 seconds	6.05 seconds	0.86 seconds	0.04 seconds

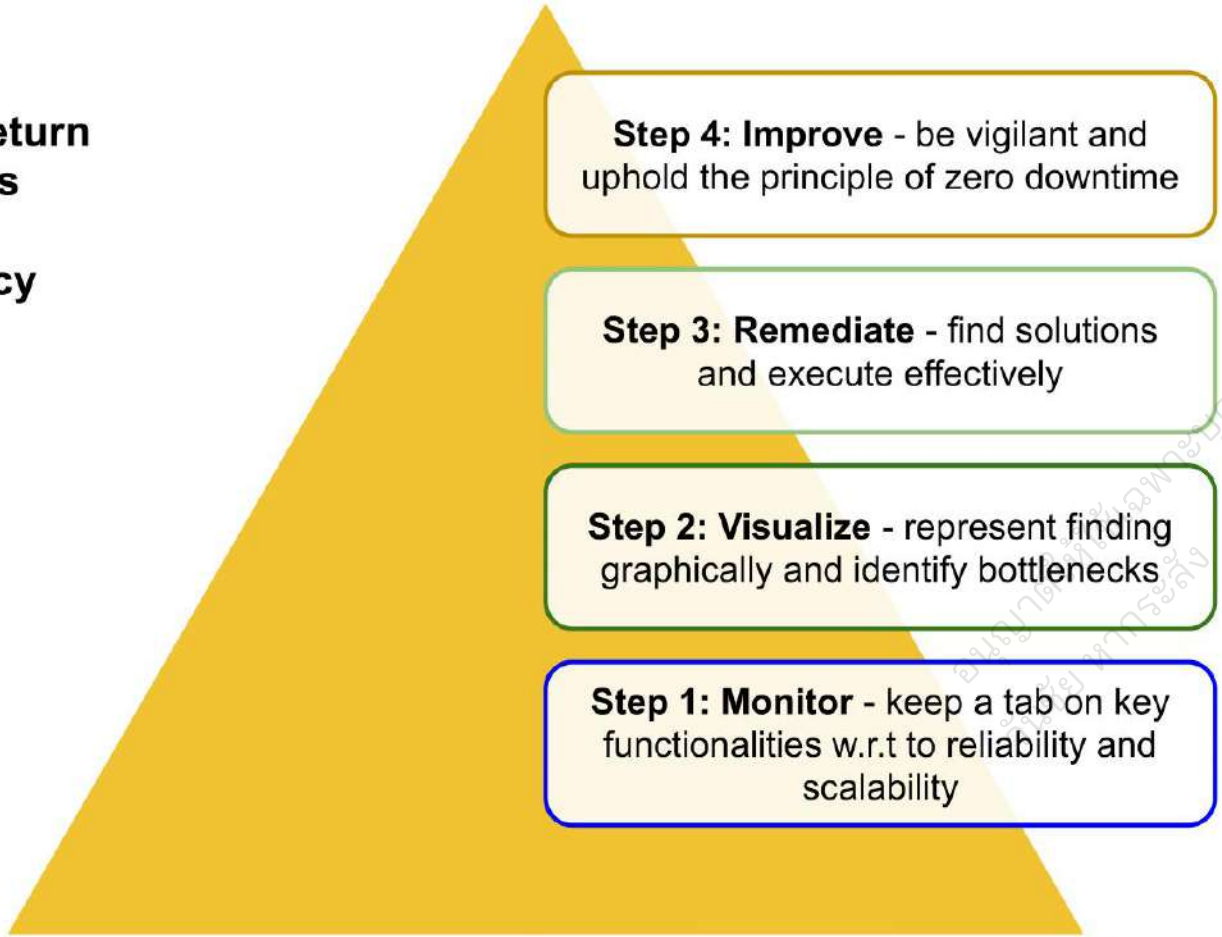
Calculations are based on the average Gregorian year: 365.2425 days

https://availability.sre.xyz

Steps of SRE

Service Level Objective (SLO)

- **The 95% of request must return successful HTTP responses (HTTP 200)**
- **The 95% of response latency must be less than 200ms**



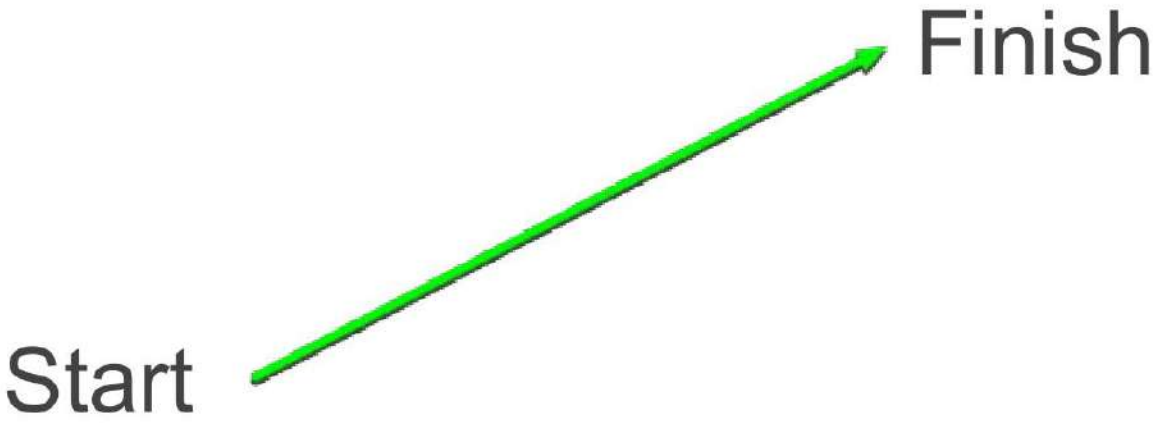
https://www.cncf.io/blog/2020/07/17/site-reliability-engineering-sre-101-with-devops-vs-sre/

Wrap Up

It’s about the

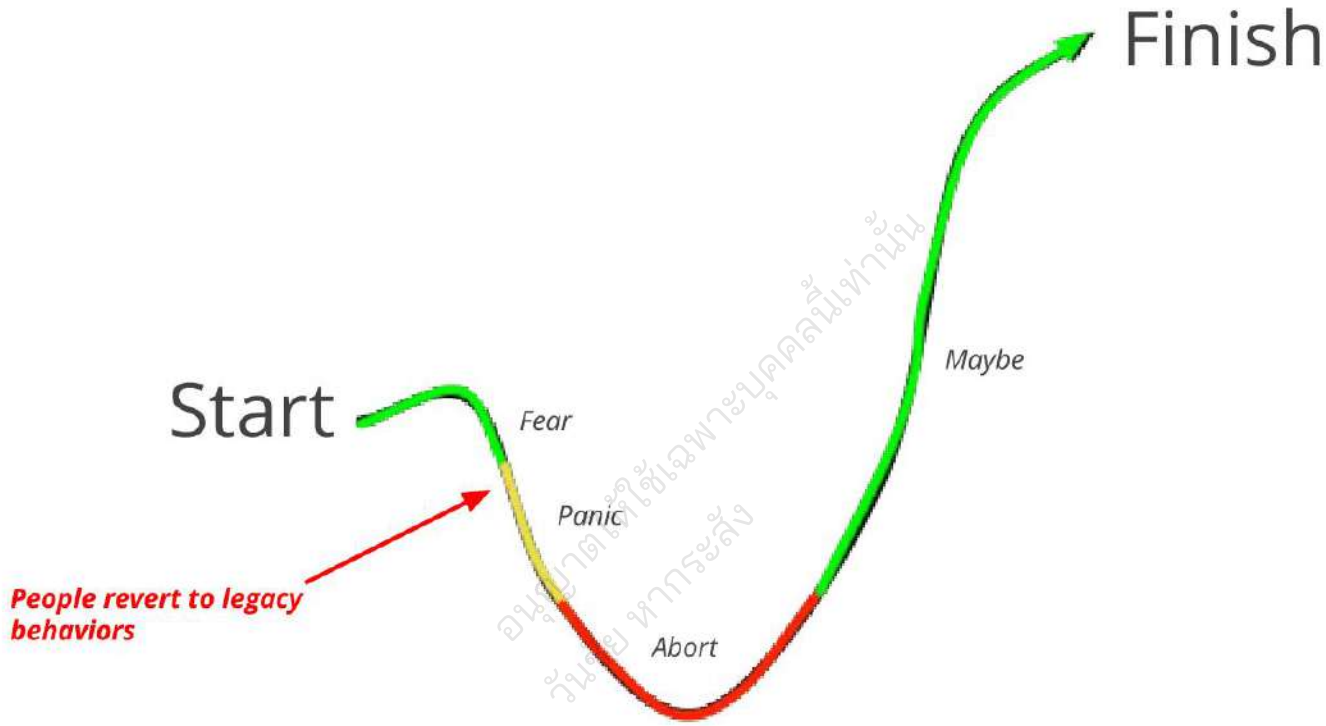
IMPROVEMENT

Big Bang Transformation



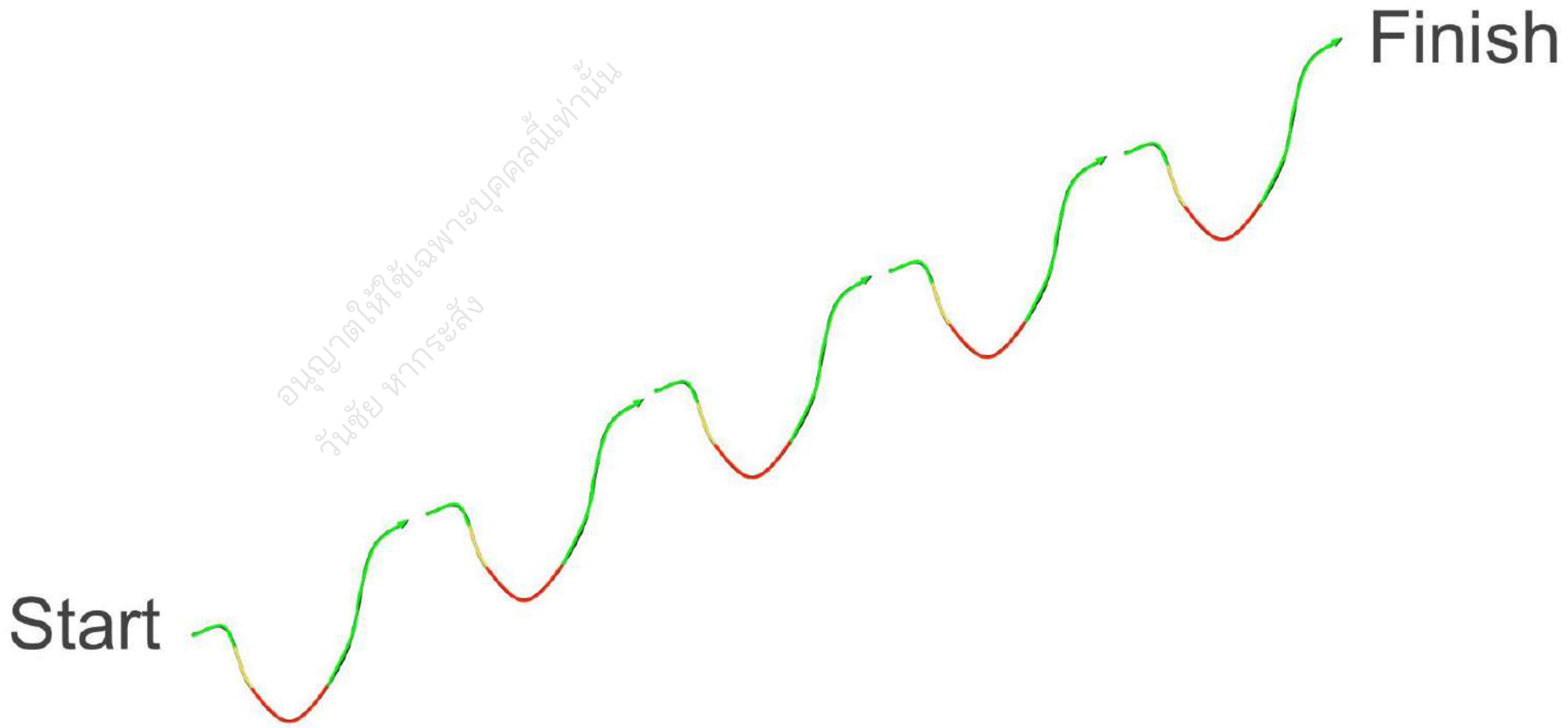
Dream

Big Bang Transformation



Reality

Many Small Transformation



It’s about the

CONTINUOUS

IMPROVEMENT