# My Journey to Kubernetes

**Agastyo Satriaji Idam**
Infrastructure Engineer @Ruangguru

kubernetes

# My 4 Phases of Learning Kubernetes

1. Unconscious Incompetence
2. Conscious Incompetence
3. Conscious Competence
4. Unconscious Incompetence

# Unconscious Incompetence

Getting acquainted with Kubernetes

# How I first met Kubernetes at 

?

# Working as a backend/devops engineer at Universitas Islam Indonesia I was tasked to...

- Migrate monolithic system to microservices architecture
- Change application/service deployment platform from VM to container
- Find the right tool for managing container orchestration

# The year was 2016, Kubernetes wasn't the only cool kid, there are others...

HashiCorp
Nomad

Apache
MESOS™

kubernetes


# The reason I finally chose Kubernetes

- Kubernetes popularity is on the rise, more companies and organizations are using it
- It has the most comprehensive features compare to other solutions
- It was born and battle-tested at Google as an Open Source fork of Borg
- Exclusively build to manage containerized workloads
- Growing communities

kubernetes

# How I learn Kubernetes?

# My go to learning resources

- Official documentation at kubernetes.io/docs/home
- Official tutorials at kubernetes.io/docs/tutorials
- Kelsey Hightower's Kubernetes The Hard Way guide
- Kelsey Hightower's Kubernetes Up & Running book
- Phone a friend at Prismapp & Sale Stock

# Conscious Incompetence

Struggling with Kubernetes

# Operating Bare-Metal K8s at


# Solutions I worked with

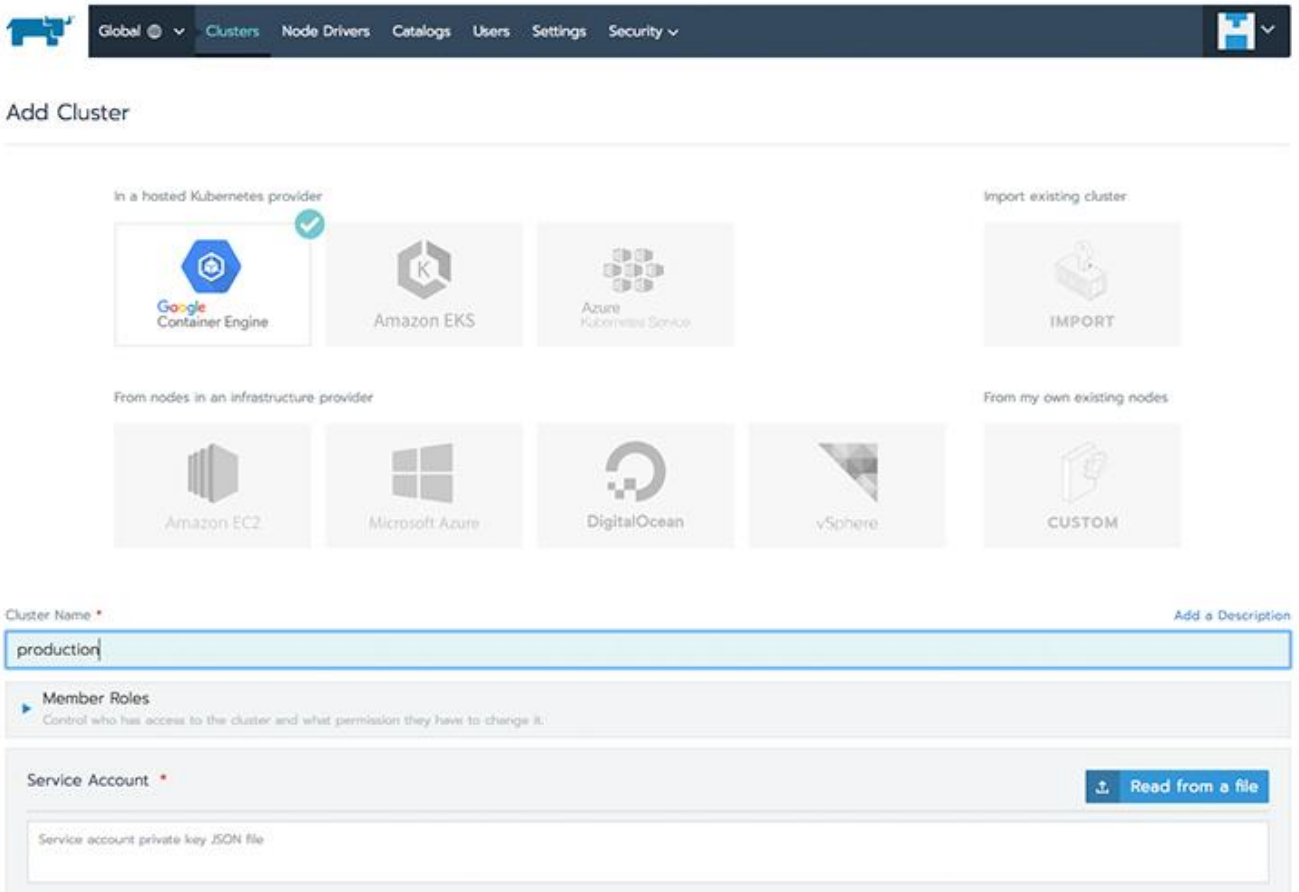
- Kubernetes The Hard Way => Failed
- Kubeadm => Failed
- Rancher => Success

More selection of tools for bare-metal & on-premise K8s:

https://github.com/ramitsurana/awesome-kubernetes#installers

kubernetes

# Rancher Kubernetes Cluster Installation Dashboard



# Challenges in operating Kubernetes

# The challenges I faced when implementing Kubernetes for the first time...

- Guiding developers to the right way
- Logging and monitoring
- Kubernetes cluster maintenance
- Securing Kubernetes cluster
- Keeping up with the continuously expanding Kubernetes landscape

# Failing in production

# Things I failed to anticipate

- Weak logging & monitoring setup
- No planned workload sizing
- Trusting Kubernetes self-healing & auto-scaling features a bit too much
- Failed to correctly implement microservices best practices

# Conscious Competence

Confidence with Kubernetes

# Operating GKE cluster at ruang guru

# A little insight on how Ruangguru Kubernetes clusters look like...

- Multiple GKE clusters, each with its own specific functionality
- The core cluster has more than 1.400 Deployments, 600 StatefulSets, and 8.000 Pods running
- Currently we have 5 different stacks used to develop our microservices, which consist of Go, Python, ReasonML, NodeJS, Ruby, & PHP

# What the experience is like compare to working with bare-metal...

**The Good:**

- Easier installation & setup
- Easier cluster maintenance & update
- Less to none networking concerns
- Nodepool autoscaling
- Preemptible instances to optimize cost

**The Bad:**

- Limited cluster customization
- Limited control to Kubernetes master components

# Anticipating Failures

# What we do at Ruangguru

- RED and USE monitoring using Prometheus + Grafana for both VMs and containers
- Log everything from every applications and services using Loki
- Implement service mesh using Istio to support more robust traffic management, security policy, & observability
- Utilize IaC for more controlled, faster, and less error-prone infrastructures update and provisioning
- Secure sensitive secrets such as access token, password, etc. using Vault

# Unconscious Competence

Taming Kubernetes

# Building Tools for Kubernetes

**Developing**

**Rogu + Newton:**

Ruangguru's internal Kubernetes deployment tools.

```
$ rogu --help
Ruangguru deployment tools.

Usage:
  rogu [command]

Available Commands:
  config             Manage service configuration
  create-version     Create a version
  db                 Manage database deployment
  dependency         manage dependecies
  deploy             Deploy a from current directory
  deploy-version     Deploy a specific version
  deployment-history get deployment history for specific service
  frontend           Run frontend project
  gen-proto          generate files from proto files in dependencies as defined in service.yaml
  help               Help about any command
  init-member        init rogu cli member info
  install-protoc     Install protoc and plugin needed
  list-version       show list of version
  loadtest           Perform loadtest.
  redeploy           redeploy last deployment version of a service
  rollback           Rollback to specific hash on deployment-history
  run                Run service localy with remote configs
  scaling            Manage service scaling
  secret             manage secret cofiguration
  set-config         set config variables for specific service
  status             get status of a service at current time
  sync               sync rogu from source
  synclibs           sync libs dependencies (go only) from shared-libs
  test               Integration test device and api service
  update             update rogu to latest version
  version            Print client version

Flags:
  -h, --help   help for rogu

Use "rogu [command] --help" for more information about a command.
```

# Newton Deployment System

- Manage service build & deployment
- Manage service versioning
- Manage service database dependency
- Manage service scaling profile
- Simplify service configuration
- Orchestrate load testing

kubernetes

# Rogu CLI

- Centralized CLI developers
- Single command to deploy & scale service running in Kubernetes
- Single command to launch load testing
- Shared libraries management
- Other ton of useful commands

kubernetes

# Tips for Your K8s Journey

# Some personal advices

- Try Kubernetes The Hard Way
- Complete your monitoring homework since Day-0
- Learn and use IaC tools (Terraform, Ansible, etc.)
- Learn Go programming
- Join Community

# That’s all folks!

Agastyo Satriaji Idam

@satriajidam on the internet

agastyo@ruangguru.com

kubernetes