KubeCon | CloudNativeCon

North America 2018

Noobernetes 101

Karthik Gaekwad @iteration1
Neependra Khare @neependra

Karthik Gaekwad

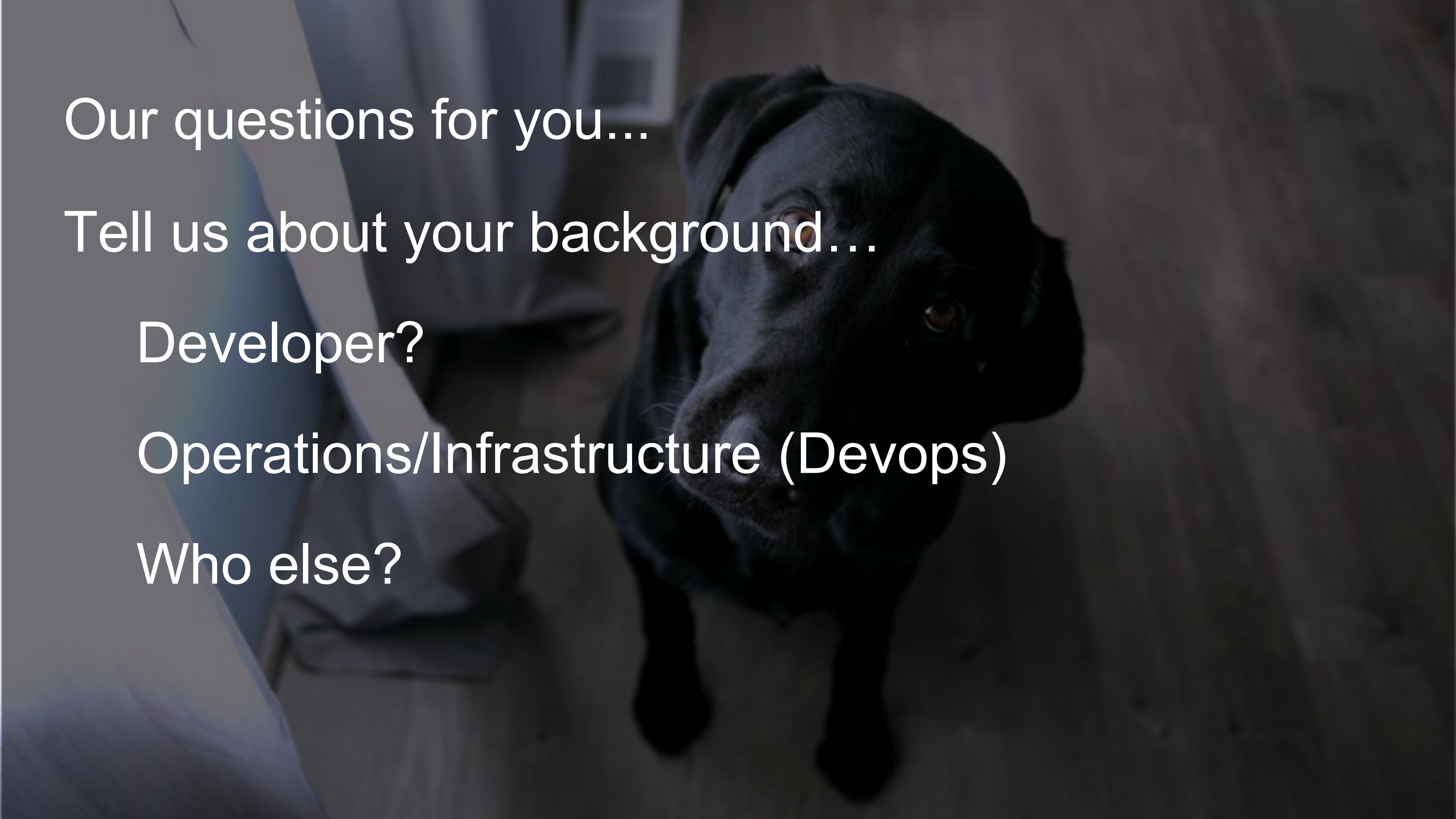
- I work at Oracle, but NOT a DBA
- https://cloudnative.oracle.com/
- Cloud Native evangelist at Oracle Cloud.
- Authored a Learning Kubernetes and a many other cloudnative courses on lynda.com/LinkedIn learning.
- Over 250k views.
- Organizes: Devopsdays Austin, Cloud Austin, All Day Devops, Container Days.
- Been in industry...for a while.

ORACLE Cloud Native Labs

ORACLE StackEngine Mentor Graphics NATIONAL INSTRUMENTS SIGNALSCIENCES

Neependra Khare

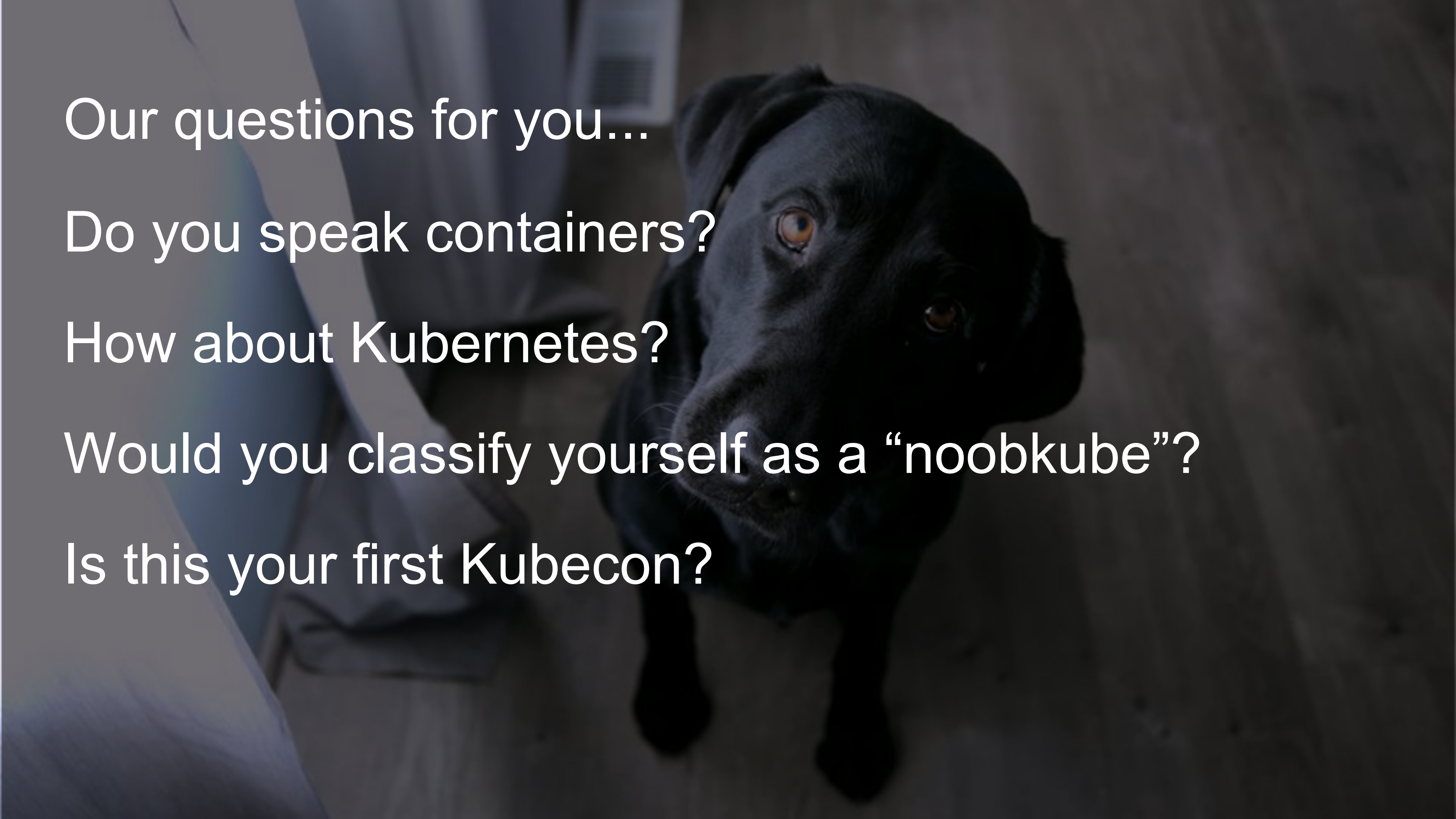
- Founder and Principal Consultant at CloudYuga
- Authored “Introduction to Kubernetes” Course on Edx for the CNCF, which has more than 55K+ participants
- Authored “Introduction to Cloud Infrastructure Technologies” on Edx for the Linux Foundation
- Authored Docker Cookbook in 2015
- Running Docker Meetup in Bangalore for last 5 years



Our questions for you...

Tell us about your background…

- Developer?
- Operations/Infrastructure (Devops)
- Who else?



Our questions for you...

Do you speak containers?

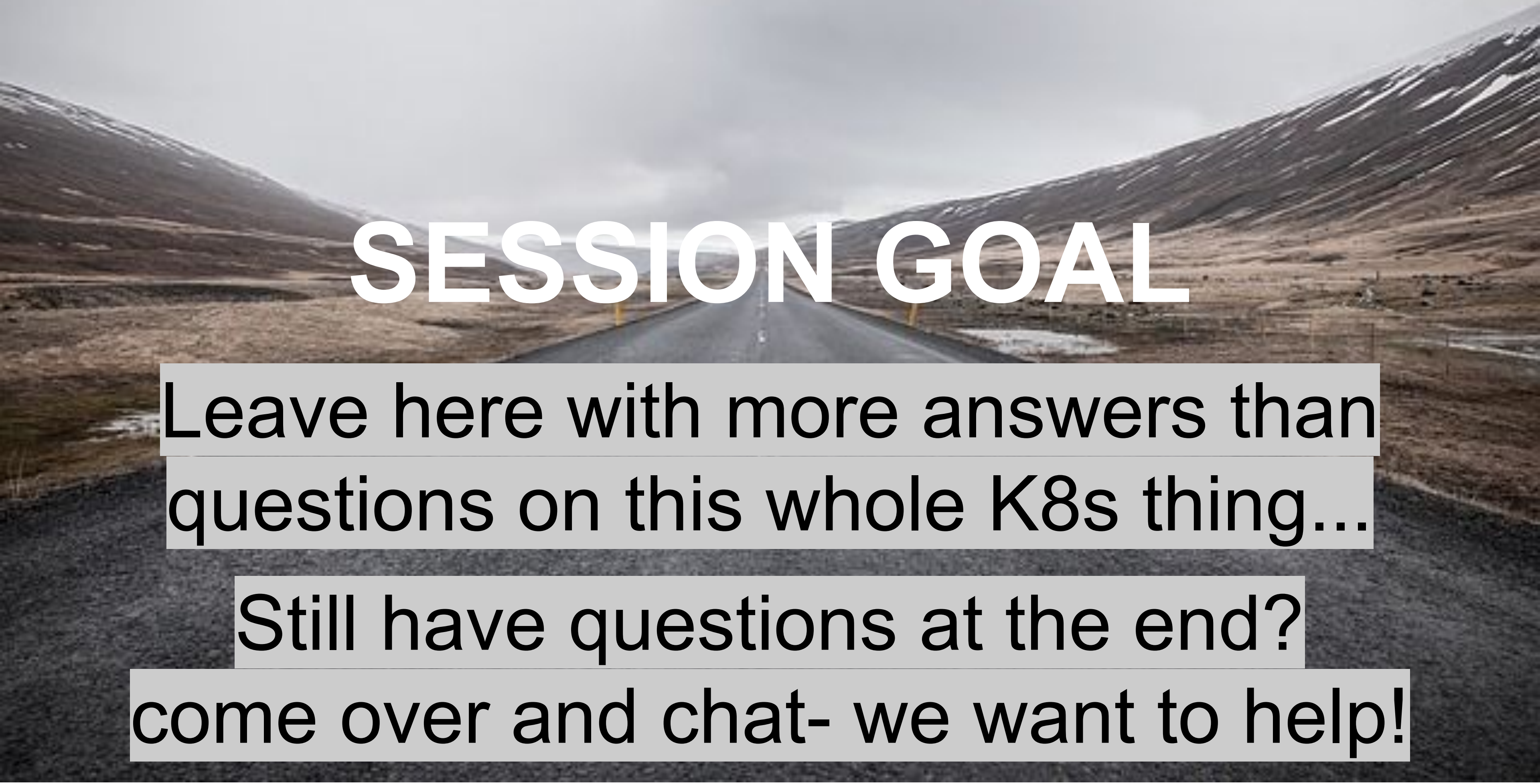
How about Kubernetes?

Would you classify yourself as a “noobkube”?

Is this your first Kubecon?

SESSION GOAL

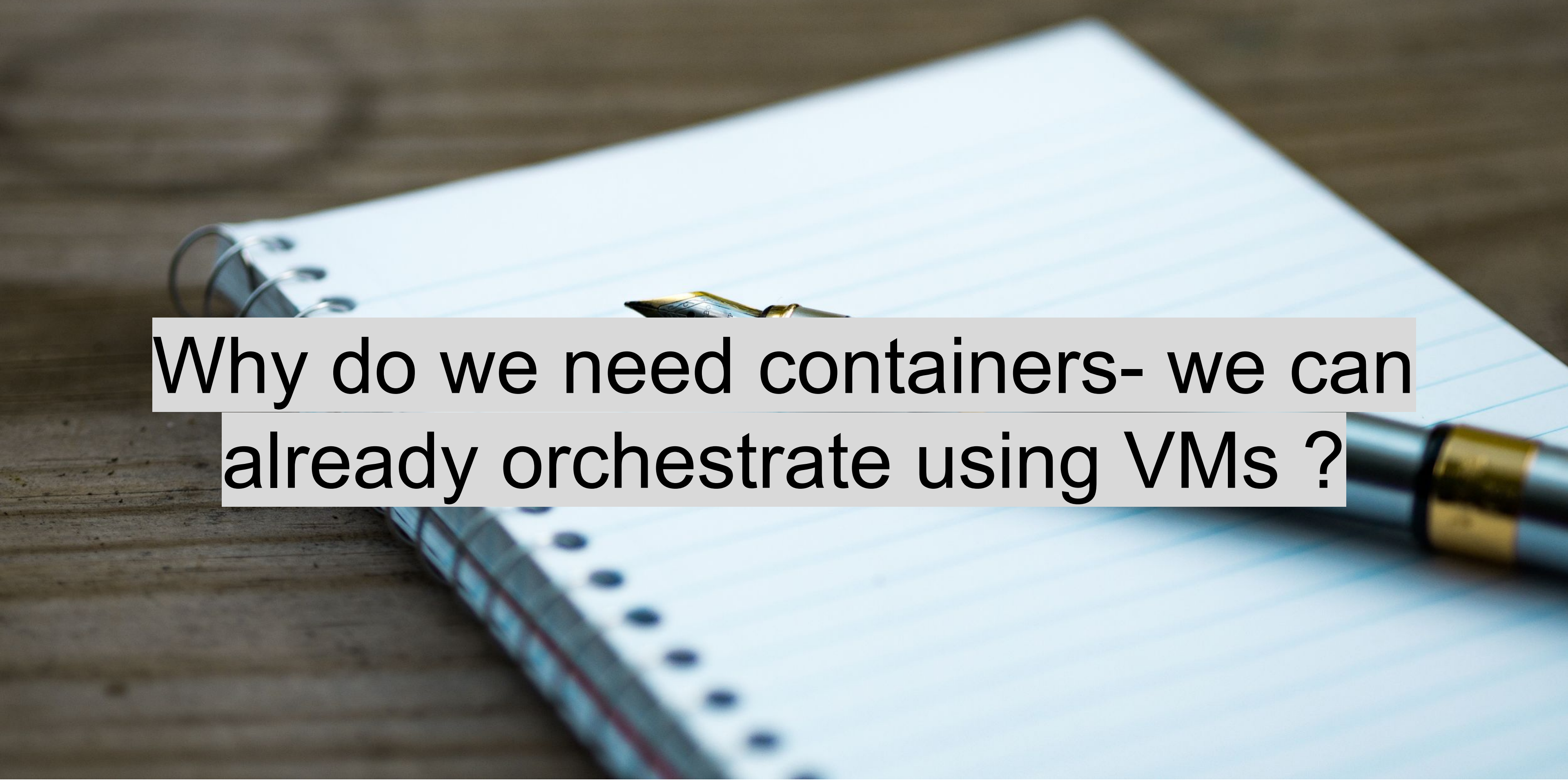
Talk through the most common Kubernetes questions we’ve gotten over time.



SESSION GOAL

Leave here with more answers than questions on this whole K8s thing...

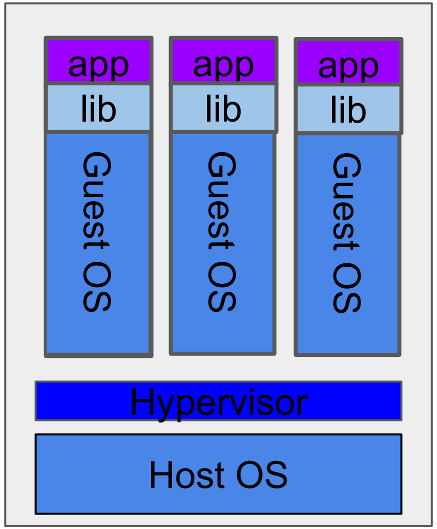
Still have questions at the end? come over and chat- we want to help!



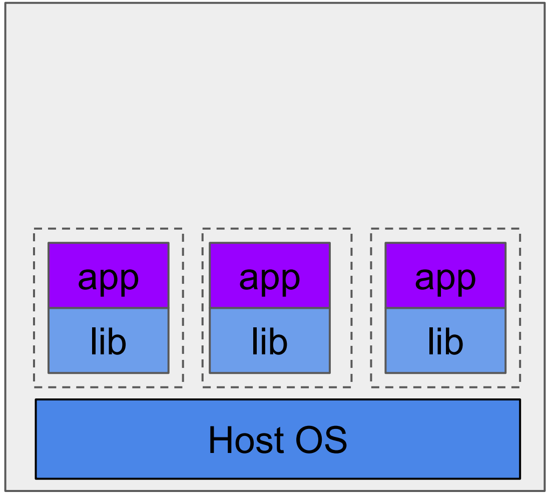
Why do we need containers- we can already orchestrate using VMs ?

VMs vs Containers

“Containers are lightweight VMs. Good for analogy but **NOT** true”



Virtual Machines



Containers

Why use Containers ?

- Lesser Cost
- More Density
- Immutable Infrastructure
- Easier to manage

Lightweight VMs ?

- Hyper-V Containers
- Kata Containers
- AWS Firecraker

On-prem Kubernetes

v/s

Cloud Hosted Kubernetes

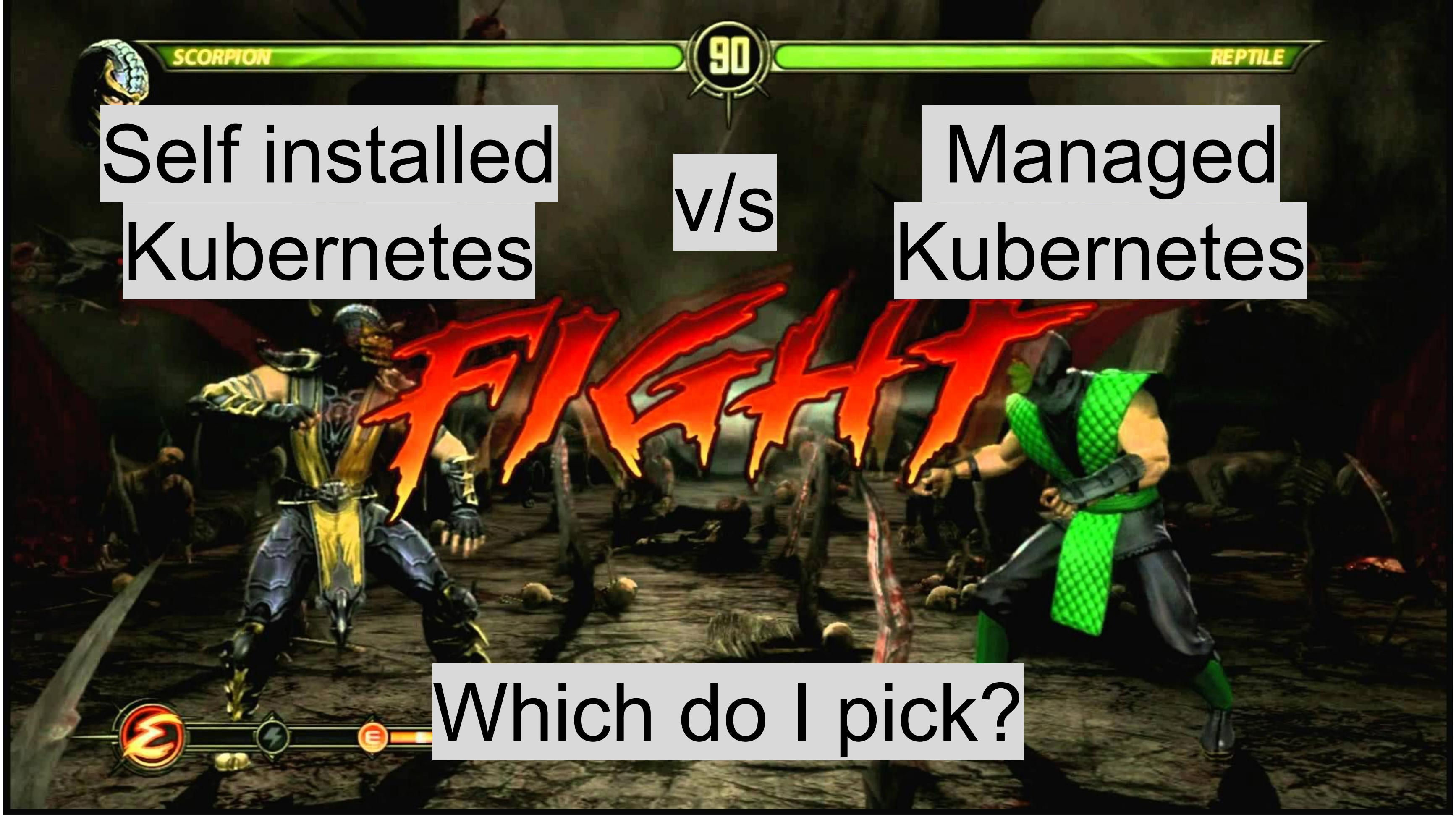
On-premise vs Cloud Hosted Kubernetes

On-Premise

- Capacity planning and Hardware provisioning needs to be done upfront
- A bit harder to expose applications to external world
- Only auto-scaling of application is supported
- Attaching external storage can be bit harder
- Full control of data

Cloud Hosted

- Hardware is provisioned on demand
- Setting up Load Balancers, Ingress etc are very straight forward
- Auto-scaling of application and nodes are supported
- Easily intergrate with cloud storage



Self installed Kubernetes

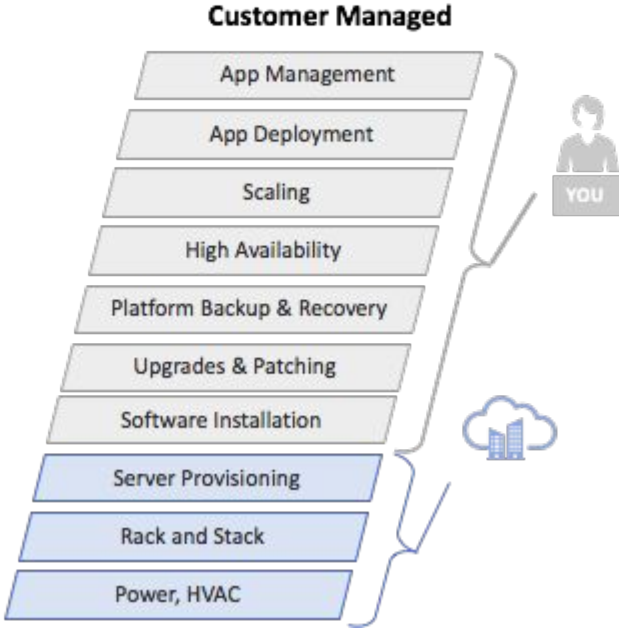
v/s

Managed Kubernetes

Which do I pick?

Self-installed K8s

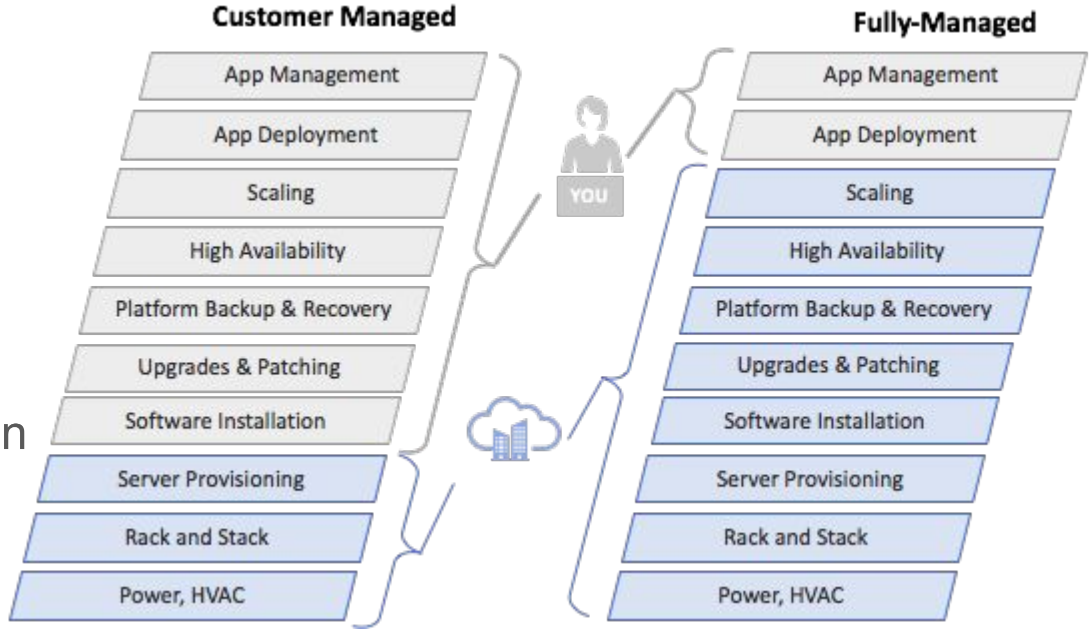
- Similar to using kops on AWS/K8s terraform installer
- 100% responsibility for all cluster.
- Typical Usecase:
  - On premise
  - Custom Cluster Requirements



YOU
HAVE
THE POWER!

Managed K8s offerings

- Cloud Provider manages Control plane.
- You manage Data plane.
- Benefits:
  - Quicker to get started
  - Lower Risk
  - Faster pace of innovation



FORM VOLTRON!!!

Teamwork = Dreamwork

Self installed v/s Managed K8s: Help me choose

- Managed K8s service

- Pros:

  –Offload control plane management to the provider.

  –Spend more time working on your apps, and less time on infrastructure.

  –Less maintenance headaches.

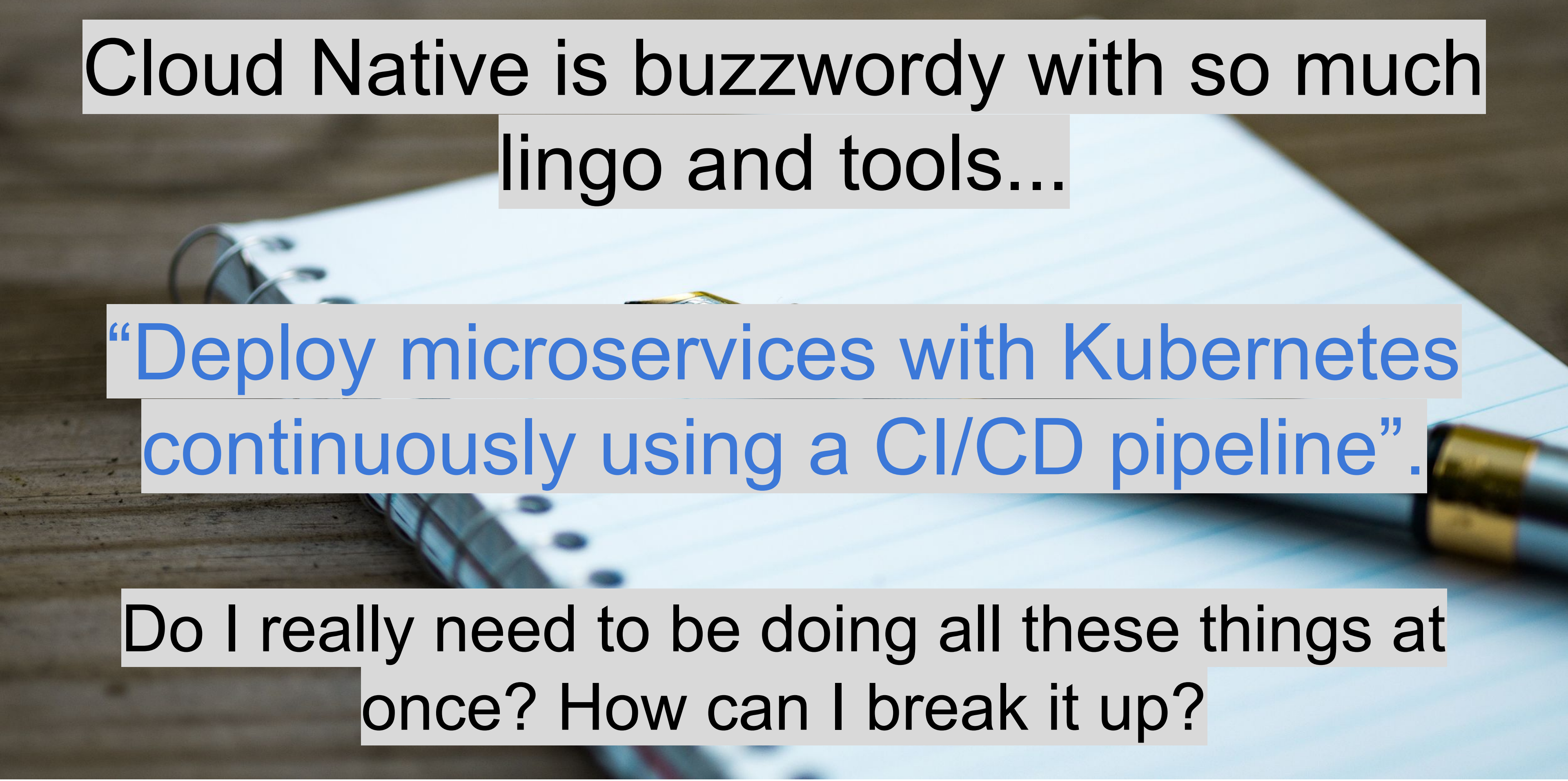
- Cons:

  –Not fully customizable (Regulatory usecases, specific networking, or on premise?)

  –Hidden costs.

Help me choose recipe

- New to K8s:
  - If (cloud)?
    - Use the managed K8s service by the cloud provider
    - Easier to get started, and play!
    - Learn the basics of K8s, understand the gotchas for your usecases.
    - Then pivot to self install if need be.
  - Else
    - Self Install

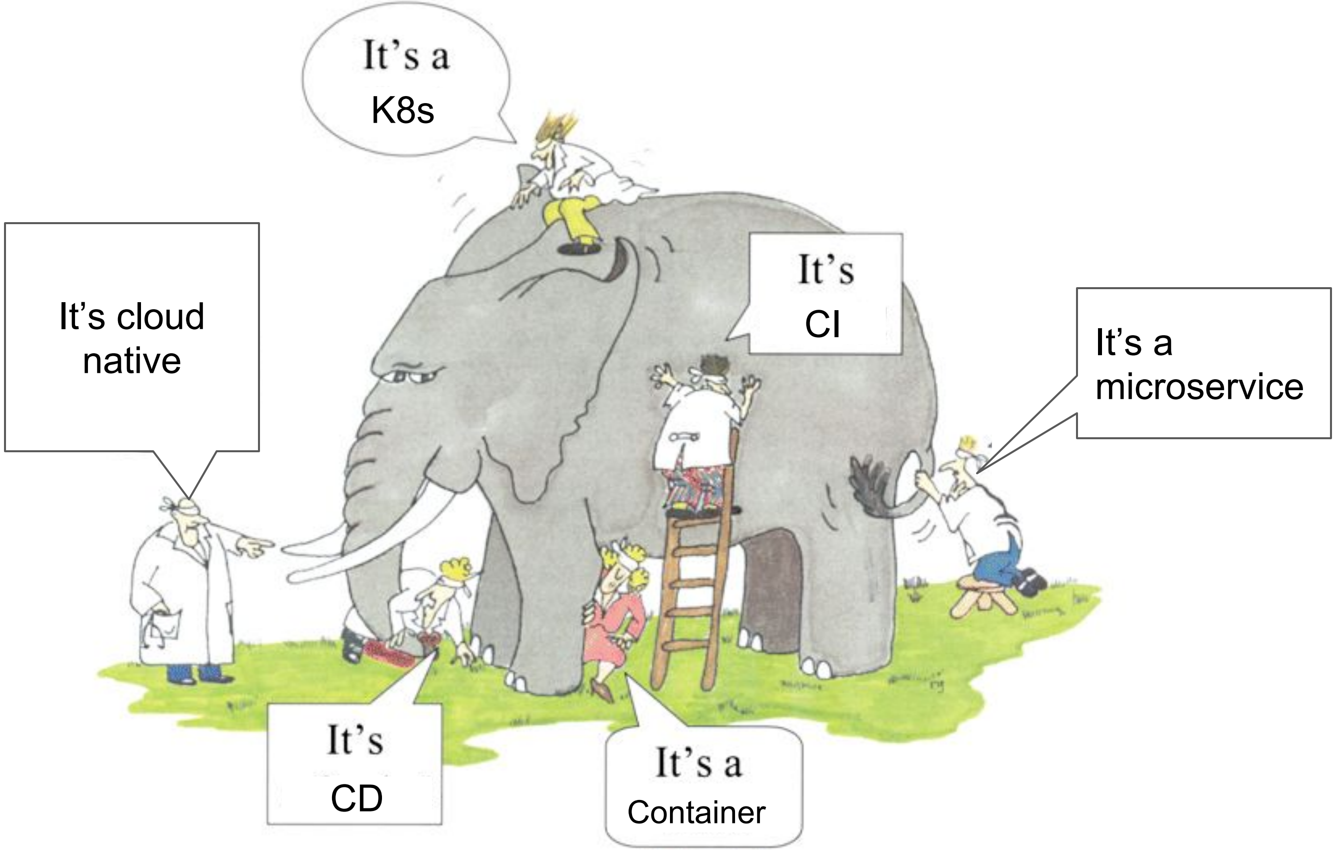


Cloud Native is buzzwordy with so much lingo and tools...

“Deploy microservices with Kubernetes continuously using a CI/CD pipeline”.

Do I really need to be doing all these things at once? How can I break it up?

The blind men and the elephant...



Kubernetes plays nicely with many things….

Kubernetes plays nicely with many things….

...Your role: decide on priority of what to do.

Strategy

- Have a plan for microservices
  - Collaborate with devs on K8s yaml
  - Common spec language for team
- CI for building your container images
- CI for pushing container images to a registry
- Then decide on a deployment strategy for K8s

How should I deploy applications in Kubernetes? (Helm/ksonnet/yamls)

Deploying applications on Kubernetes

Directly from Code

- Draft
- Skaffold

Using Manifests Files

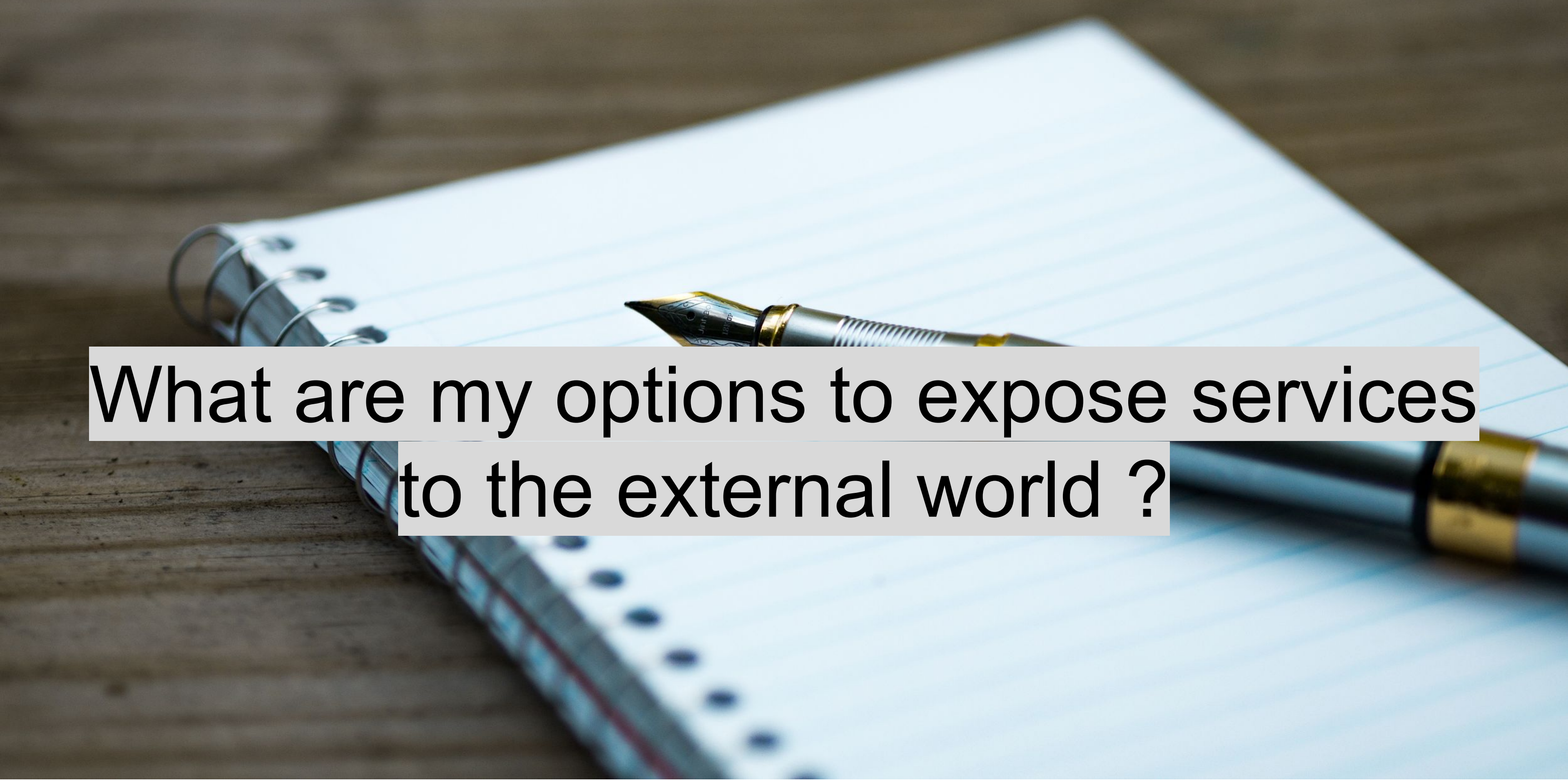
- Helm
  - Uses charts, which contains templatize *yaml* files
- Ksonnet
- Kustomize

GitOps and Kubernetes

kubectl + Git

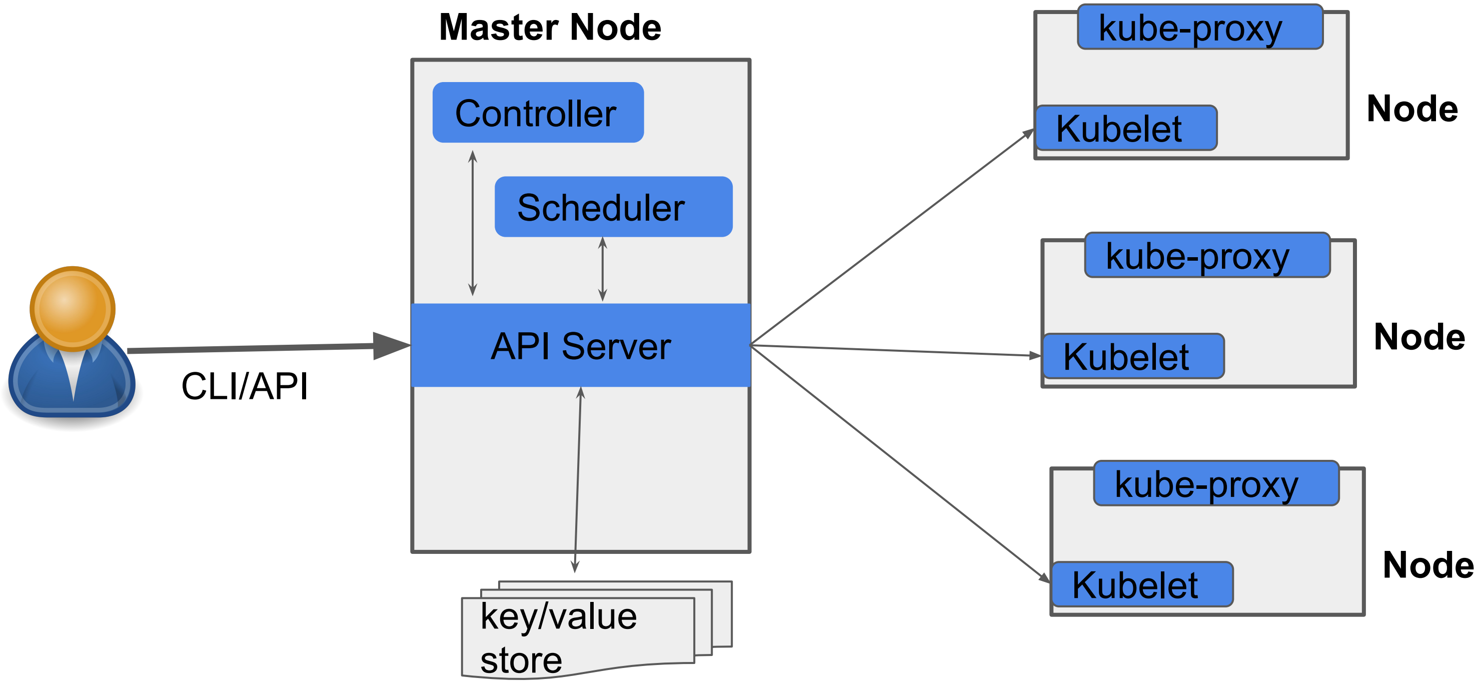
"Committing a chage on Git; deploys or updates the app on k8s."

- ArgoCD
- Jenkins X
- Weaveworks/flux
- Hasura/gitkube

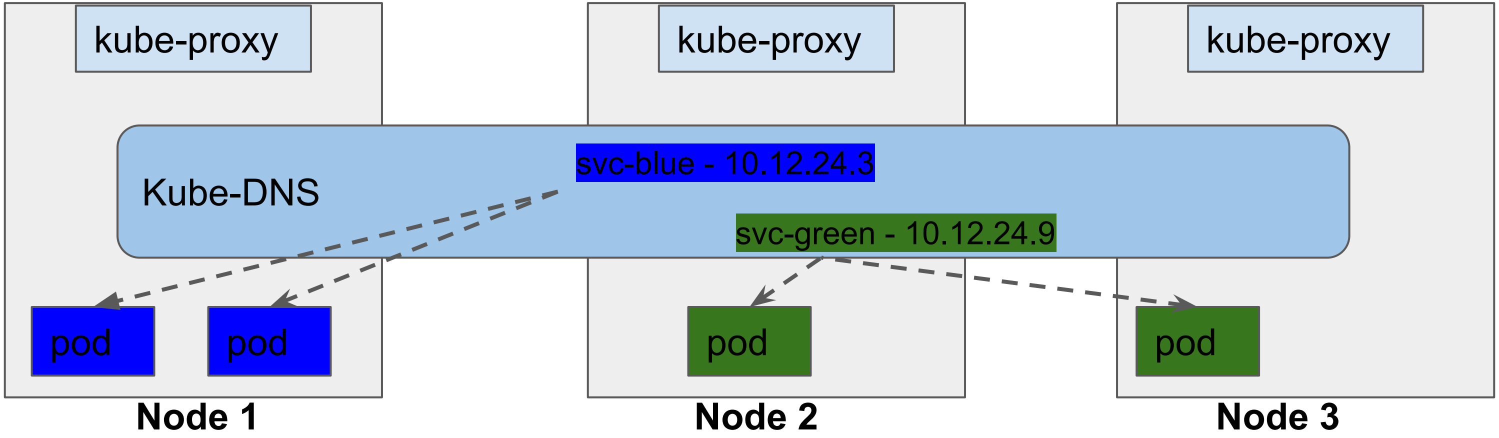


What are my options to expose services to the external world ?

Kubernetes Architecture - Quick Overview



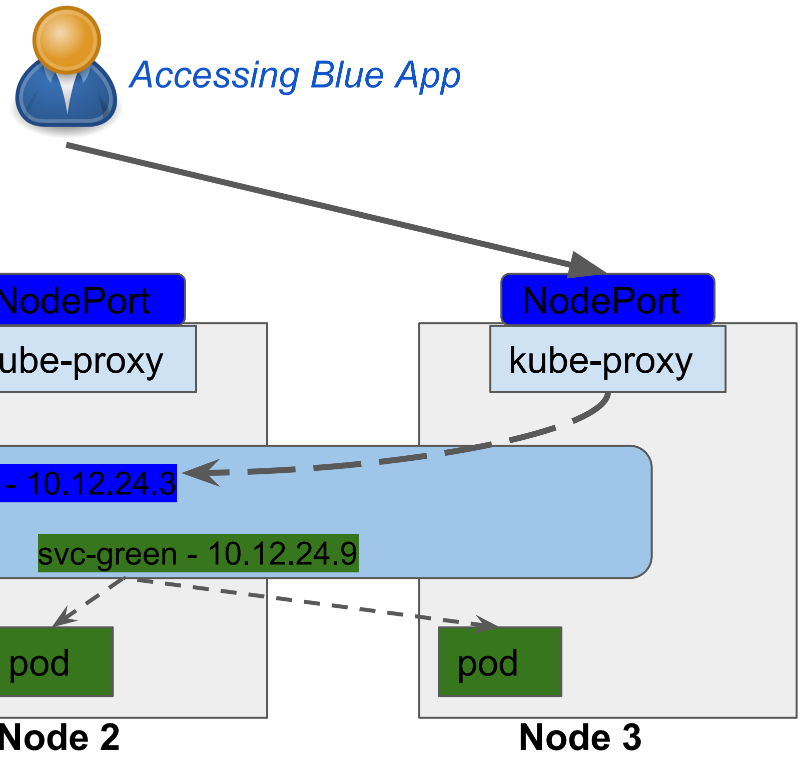
Accessing the Application - ClusterIP



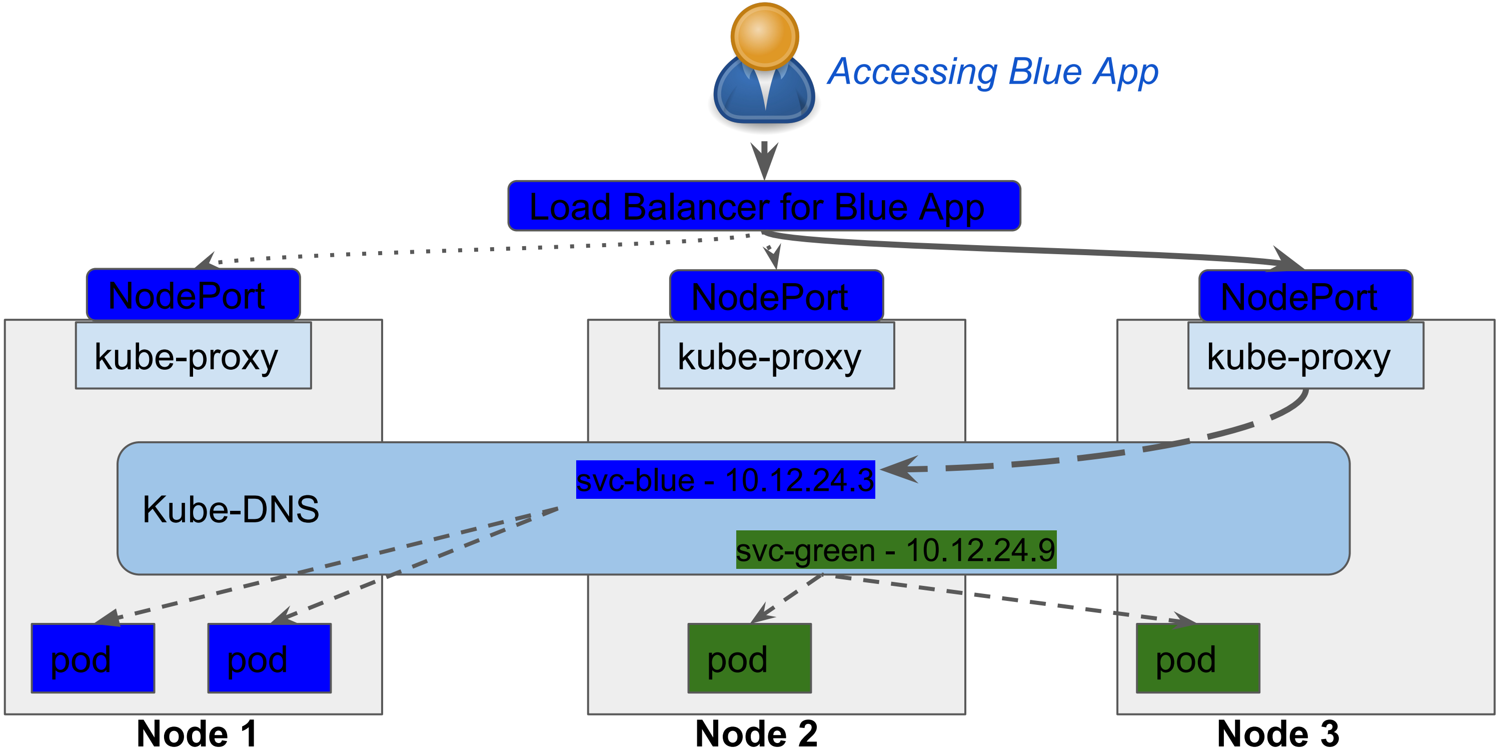
@iteration1

@neependra

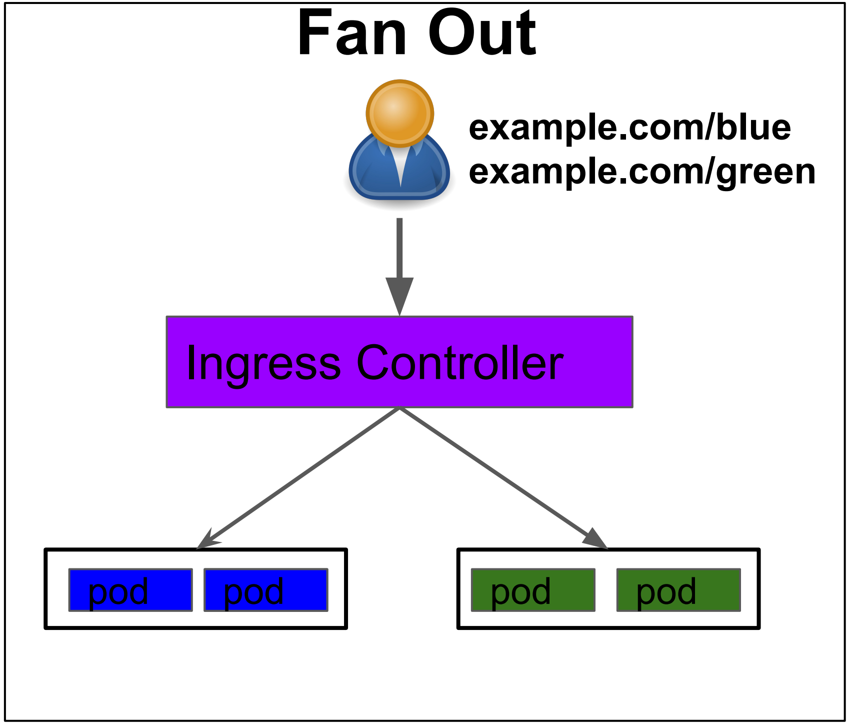
Accessing the Application - NodePort

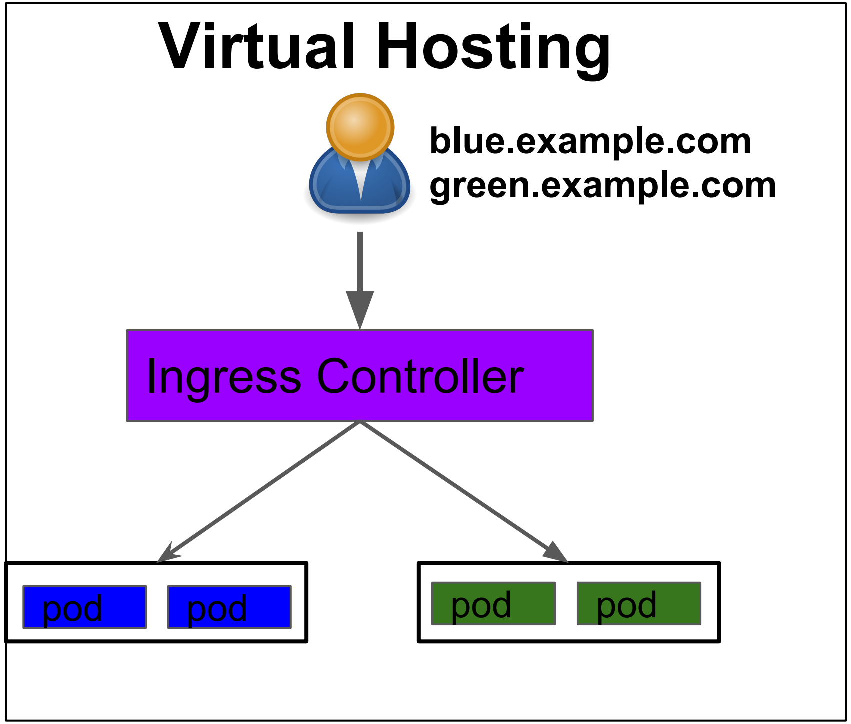


Accessing the Application - LoadBalancer

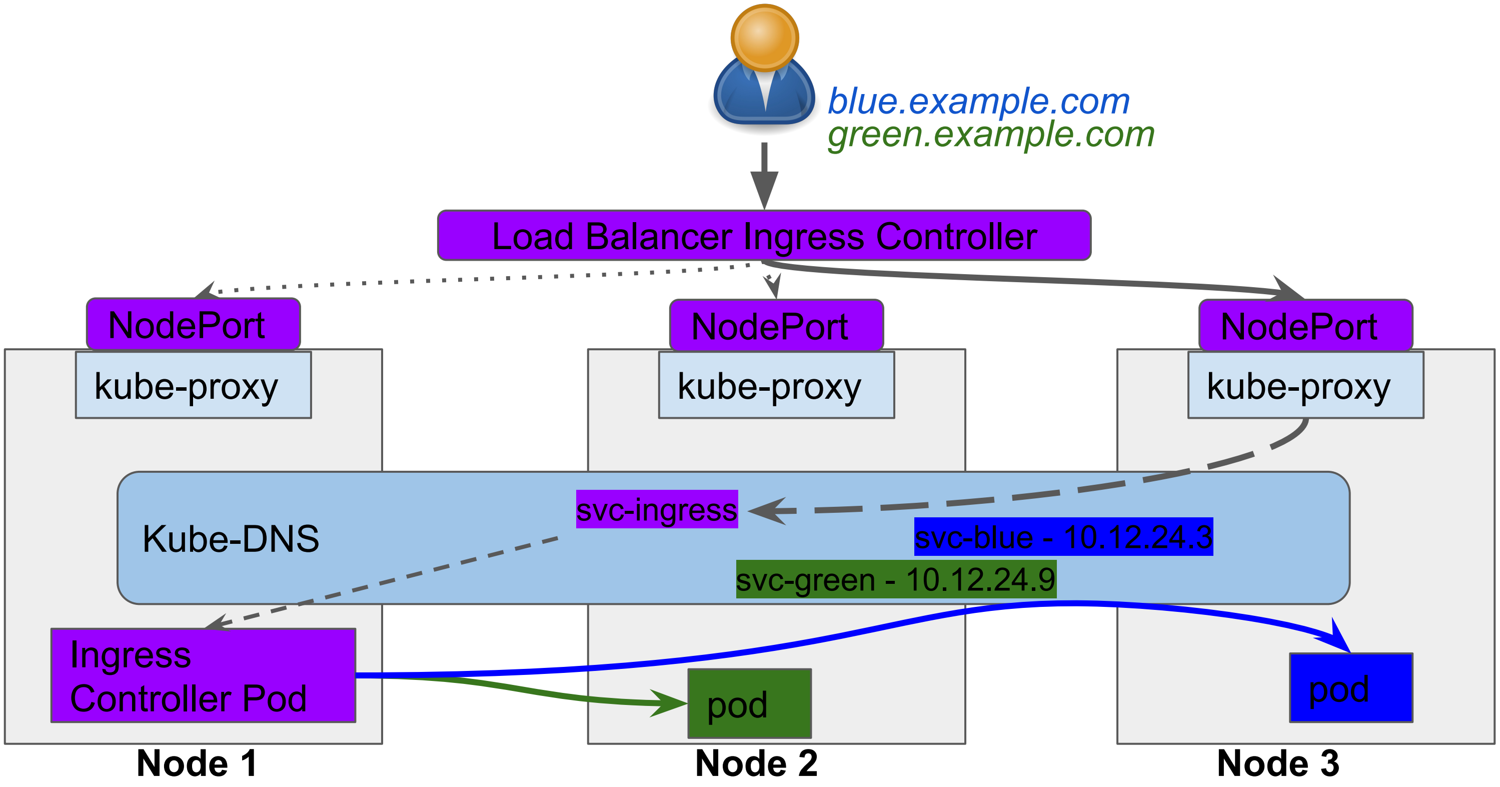


Accessing the Application - Ingress





Accessing the Application - Ingress



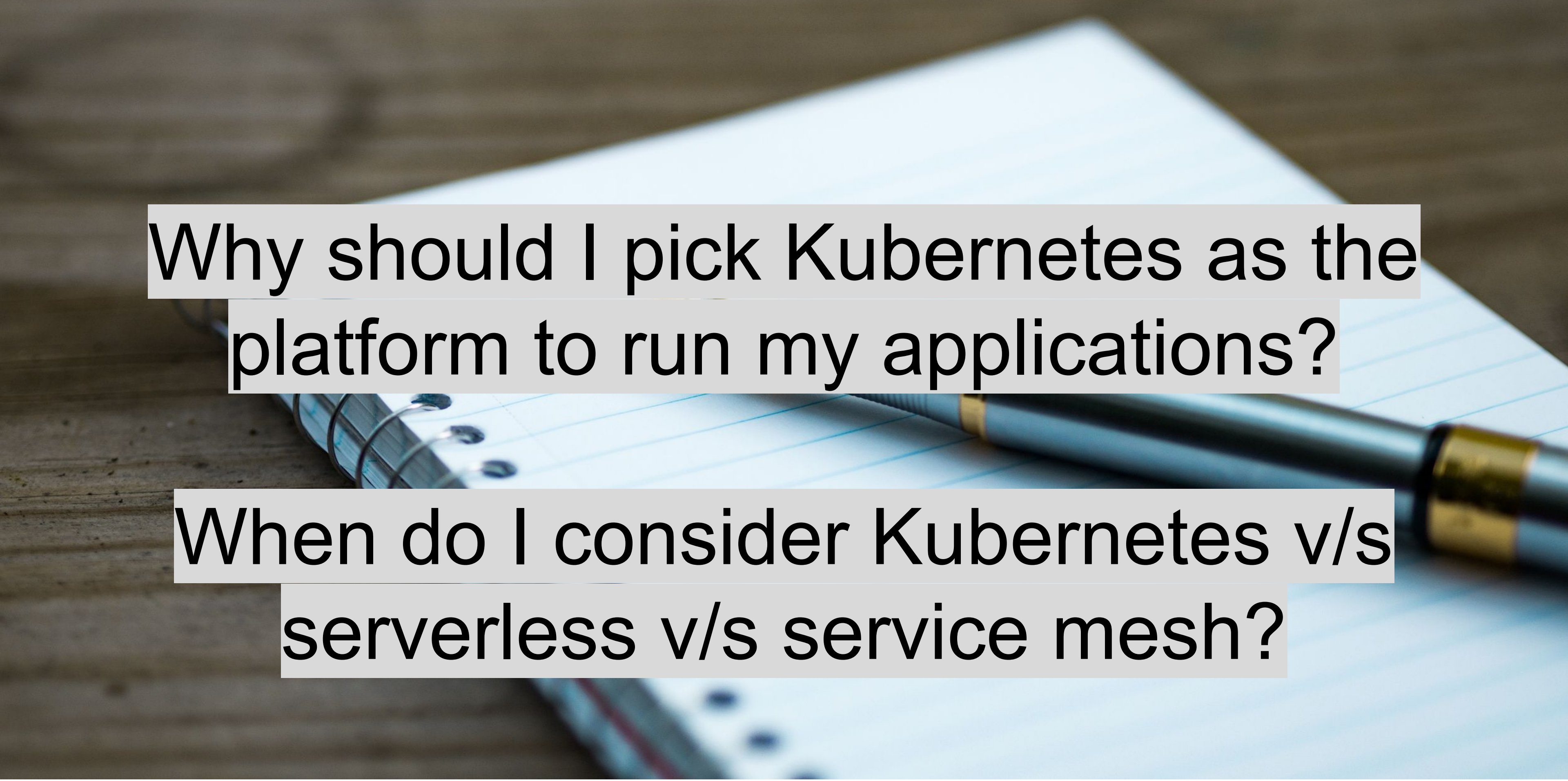
@iteration1

@neependra

Other ways to accessing the application

- Amazon’s Application Load Balancer
- Amazon’s Network Load Balancer
- Google Cloud Load Balancer

Image ref: https://aws.amazon.com/blogs/opensource/kubernetes-ingress-aws-alb-ingress-controller/



Why should I pick Kubernetes as the platform to run my applications?

When do I consider Kubernetes v/s serverless v/s service mesh?

Kubernetes v/s Service Mesh v/s Serverless

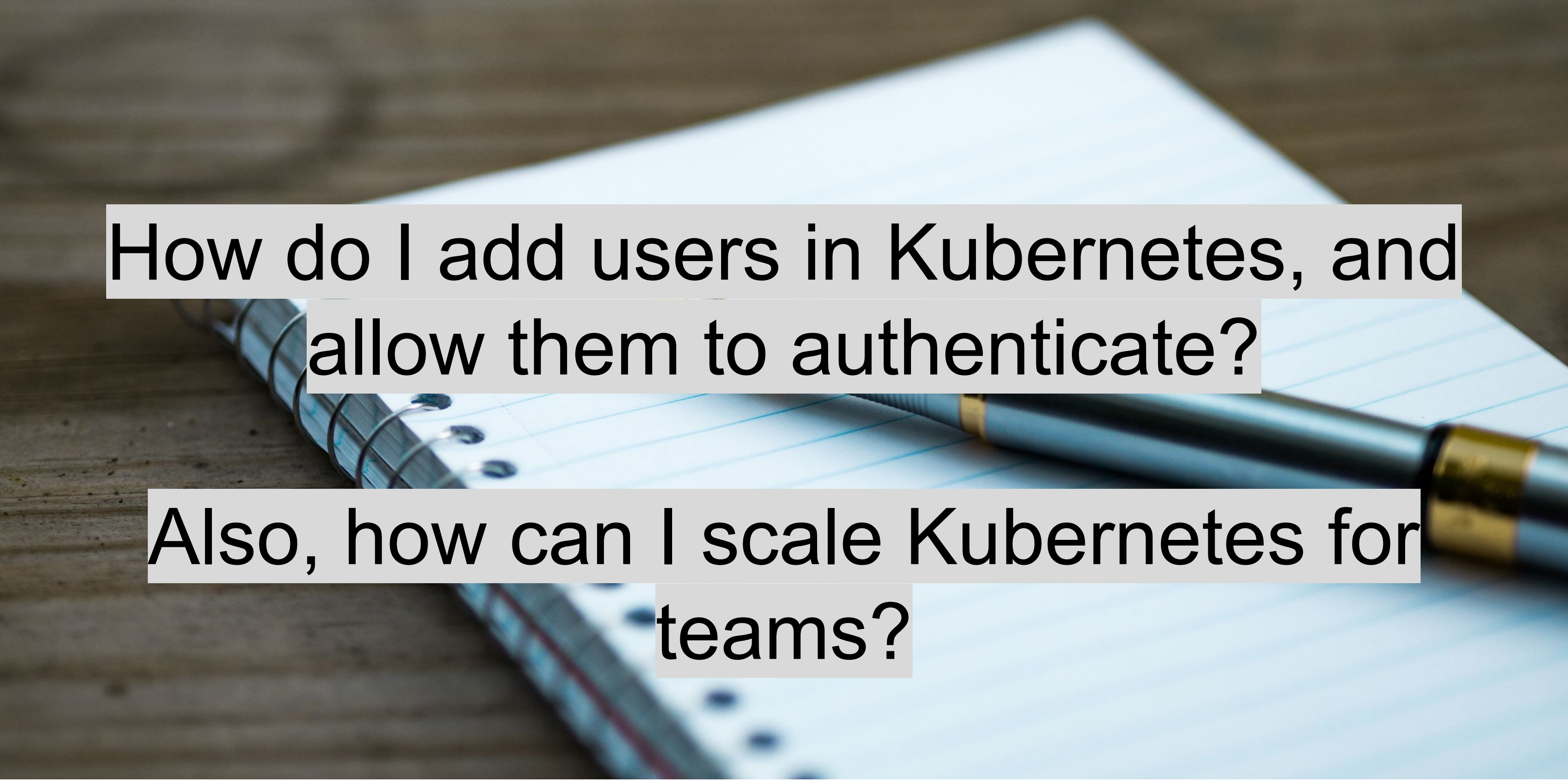
- General answer:
  - Depends on your usecase
- Real answer (question):
  - What problem does K8s solve for your org/team?
  - Are you ready to manage a new platform?
  - Consider your usecase..

Kubernetes v/s **Service Mesh** v/s Serverless

- Service Mesh:
  - IMHO: Subset of K8s.
  - Learn how to use K8s in production first before istio etc.
  - NEW and EVOLVING- build with caution, don’t over engineer

Kubernetes v/s Service Mesh v/s **Serverless**

- Fits well for small teams/dev teams.
- Lots of usecases for processing/API’s etc.
- Less management overheard of clusters and nodes.
- Better fit for teams that have usecases that don’t require orchestration, or need events.
- Good to ask “why can’t we use serverless?” if you don’t already use K8s today.



How do I add users in Kubernetes, and allow them to authenticate?

Also, how can I scale Kubernetes for teams?

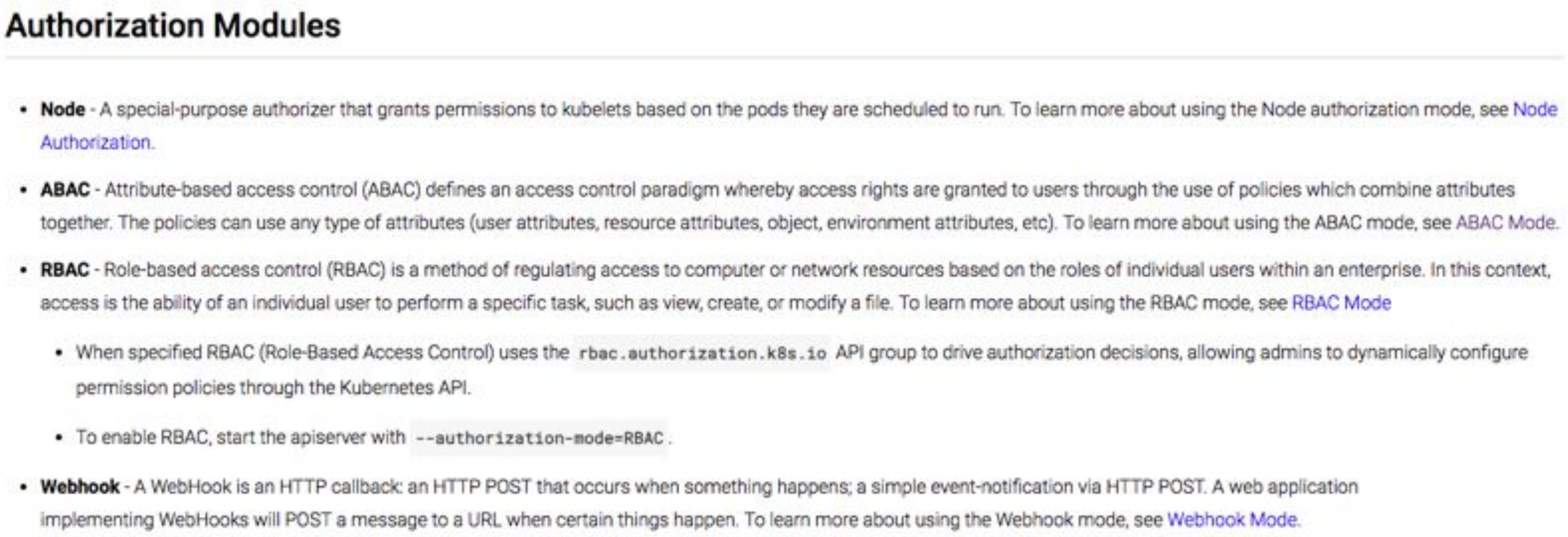
Authentication...

Many ways to Authenticate:

- Client Certs
- Static token file
- Service Account tokens
- OpenID
- Webhook Mode
- And more (https://kubernetes.io/docs/reference/access-authn-authz/authentication/)

Whatever you do, don’t yolo

Kubernetes Authorization



Authorization Modules

- **Node** - A special-purpose authorizer that grants permissions to kubelets based on the pods they are scheduled to run. To learn more about using the Node authorization mode, see Node Authorization.
- **ABAC** - Attribute-based access control (ABAC) defines an access control paradigm whereby access rights are granted to users through the use of policies which combine attributes together. The policies can use any type of attributes (user attributes, resource attributes, object, environment attributes, etc). To learn more about using the ABAC mode, see ABAC Mode.
- **RBAC** - Role-based access control (RBAC) is a method of regulating access to computer or network resources based on the roles of individual users within an enterprise. In this context, access is the ability of an individual user to perform a specific task, such as view, create, or modify a file. To learn more about using the RBAC mode, see RBAC Mode
  - When specified RBAC (Role-Based Access Control) uses the `rbac.authorization.k8s.io` API group to drive authorization decisions, allowing admins to dynamically configure permission policies through the Kubernetes API.
  - To enable RBAC, start the apiserver with `--authorization-mode=RBAC`.
- **Webhook** - A WebHook is an HTTP callback: an HTTP POST that occurs when something happens; a simple event-notification via HTTP POST. A web application implementing WebHooks will POST a message to a URL when certain things happen. To learn more about using the Webhook mode, see Webhook Mode.

https://kubernetes.io/docs/reference/access-authn-authz/authorization/

Authn and Authz

- Pro tip: Nobody uses ABAC anymore. Don’t be that person...

- RBAC is the defacto standard
  - Based on roles and role bindings
  - Good set of defaults: https://github.com/uruddarraju/kubernetes-rbac-policies
- Can use multiple authorizers together, but can get confusing.
  - 1st authorizer to authorize passes authz

Other Idea:

Larger teams…

- Namespaces:
    - Consider using namespaces to separate out your work for your environments
- Separate Clusters:
    - Multiple unique clusters for separation of concerns
        - Unique dev, test and production clusters
    - Treat as cattles not pets!

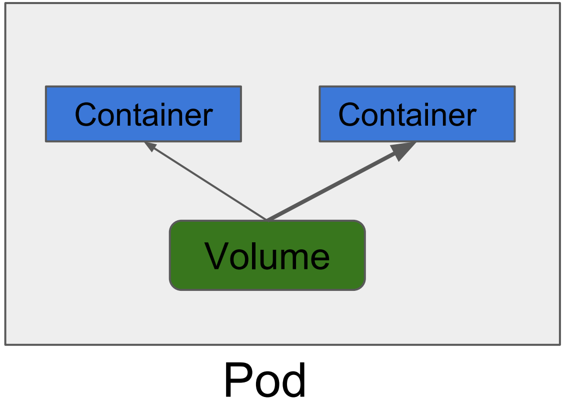
What are the best ways to handle secrets/certs/credentials ?

Credentials and secrets

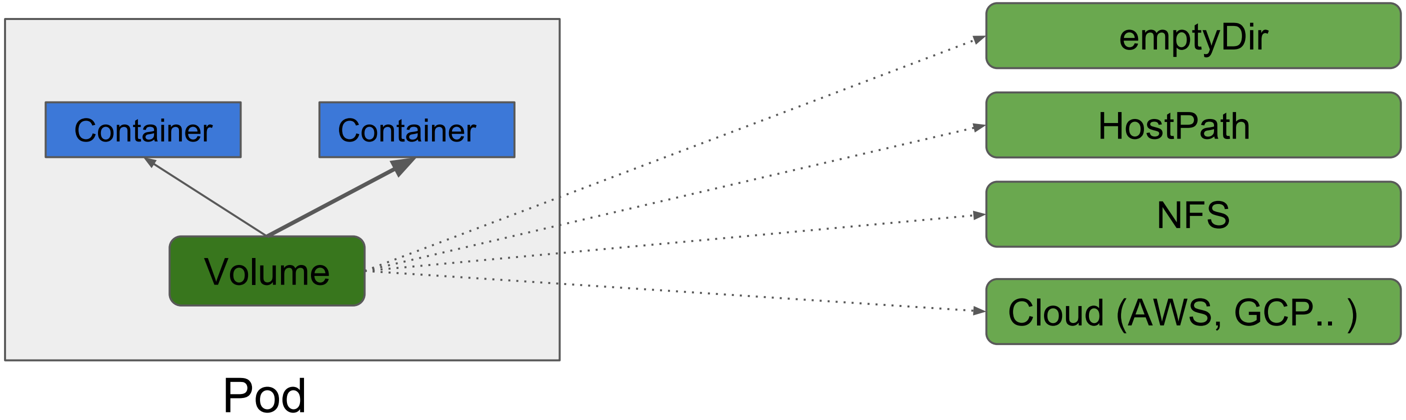
- Always Kubernetes secrets to store sensitive data.
- Look into: secrets encryption provider.
- Controls how etcd encrypts API data
- **--experimental-encryption-provider-config**
- https://kubernetes.io/docs/tasks/administer-cluster/encrypt-data/

How do I store data in Kubernetes ?

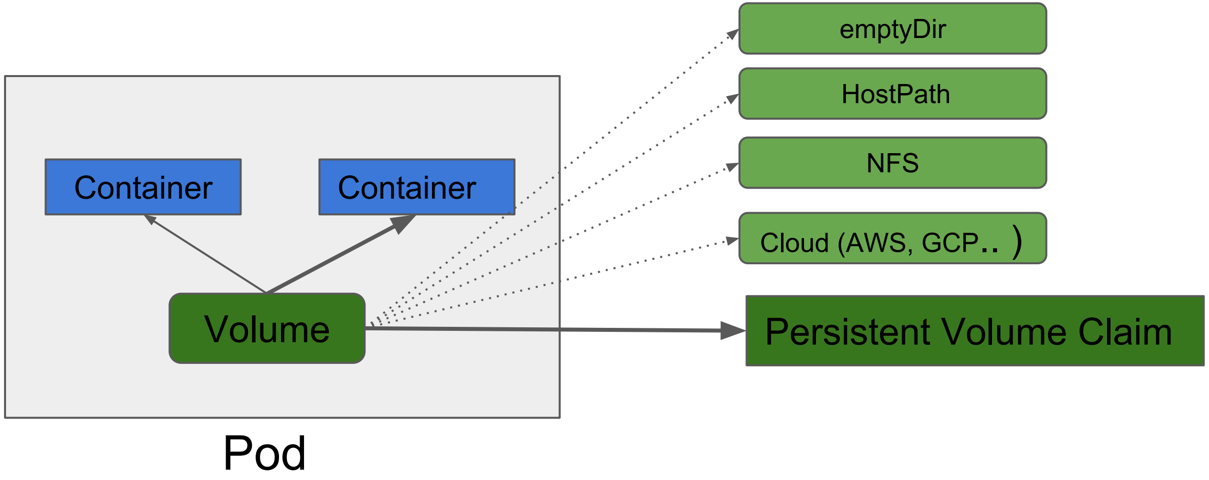
Volume Management in Kubernetes



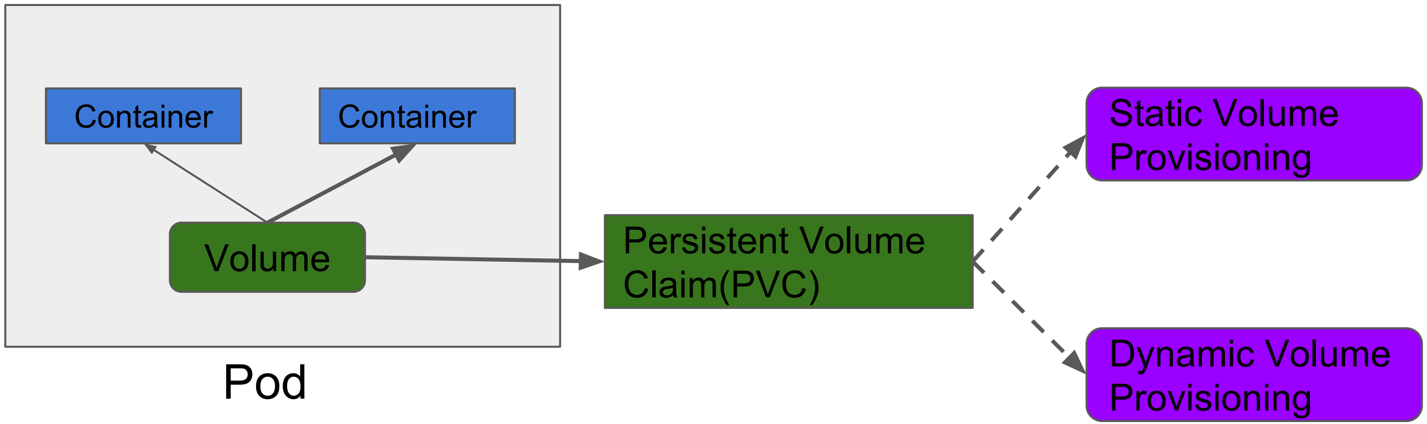
Volume Management in Kubernetes



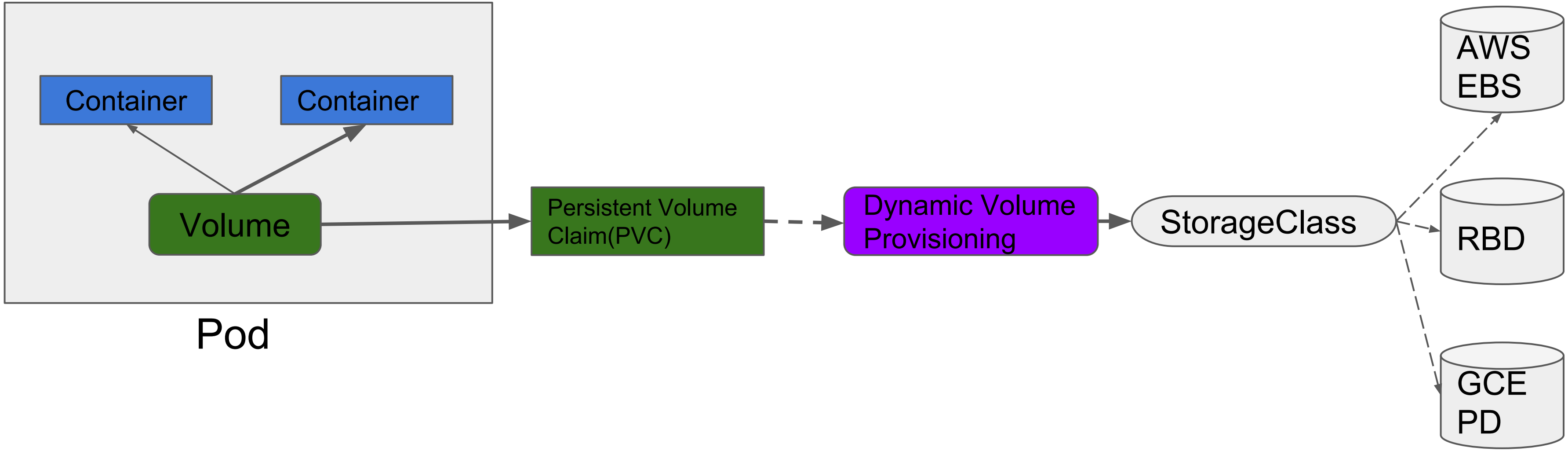
Volume Management in Kubernetes



Volume Management in Kubernetes


Container
Container
Volume
Pod
Persistent Volume Claim(PVC)
Static Volume Provisioning
Dynamic Volume Provisioning

Volume Management in Kubernetes



Where do I go from here?

Neependra Says...

- Take the free on Edx from CNCF, which I authored.

Home > All Subjects > Computer Science > Introduction to Kubernetes

Introduction to Kubernetes

Want to learn Kubernetes? Get an in-depth primer on this powerful system for managing containerized applications.

- Kubernetes Documentation
- Checkout CloudYuga’s self paced learning platform (https://pilot.cloudyuga.guru)

Karthik says...

- Join the Kubernetes Slack community (http://slack.k8s.io/)
- Check out https://cloudnative.oracle.com/
- Connect with me if you have questions!

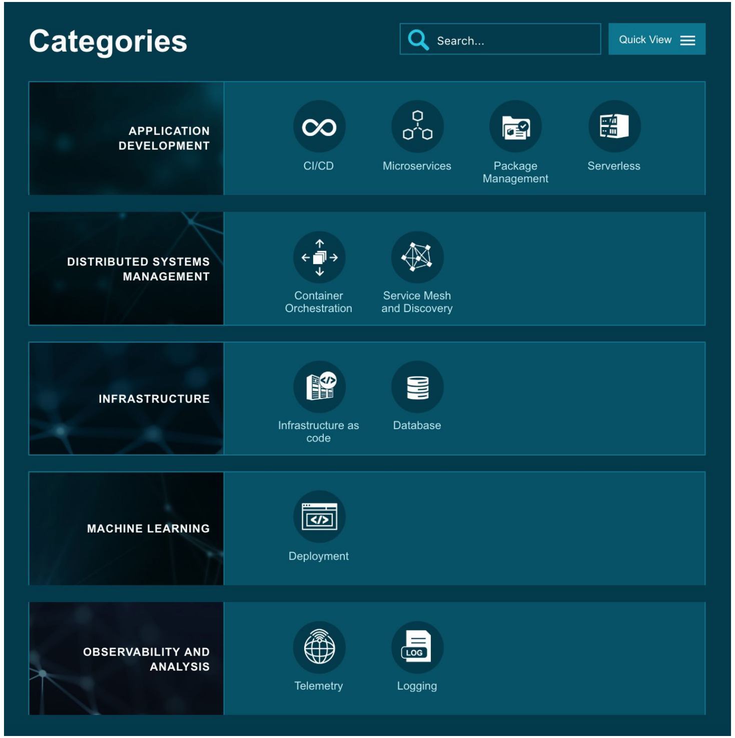
@iteration1/LinkedIn/Email

https://blogs.oracle.com/cloudnative/beginner-kubernetes

December 6, 2018

Kubernetes 101

Mickey Boxell
PRODUCT MANAGEMENT



Categories

Search...

Quick View

APPLICATION DEVELOPMENT: CI/CD, Microservices, Package Management, Serverless

DISTRIBUTED SYSTEMS MANAGEMENT: Container Orchestration, Service Mesh and Discovery

INFRASTRUCTURE: Infrastructure as code, Database

MACHINE LEARNING: Deployment

OBSERVABILITY AND ANALYSIS: Telemetry, Logging

KEEP CALM
AND
KUBE ON

@iteration1 @neependra