?

Kubernetes

Hello!

I am Yeshwanth Reddy

I am here because I like to talk Kubernetes.

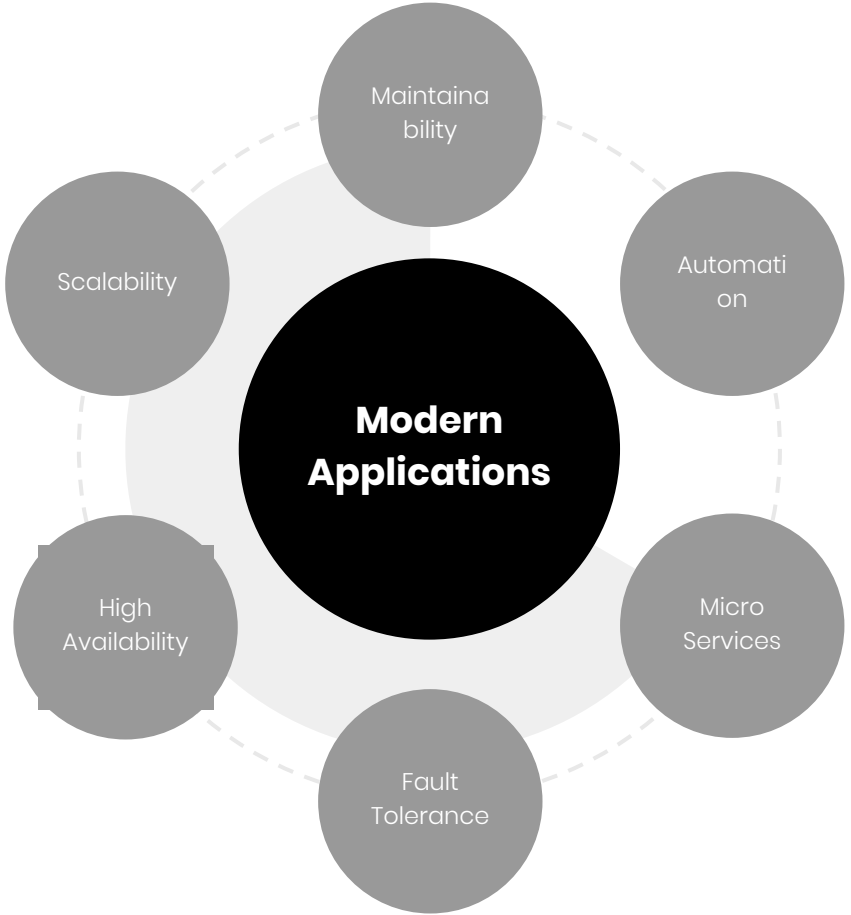
You can find me at:

@golazynani on Twitter

@LazyNani in Slack

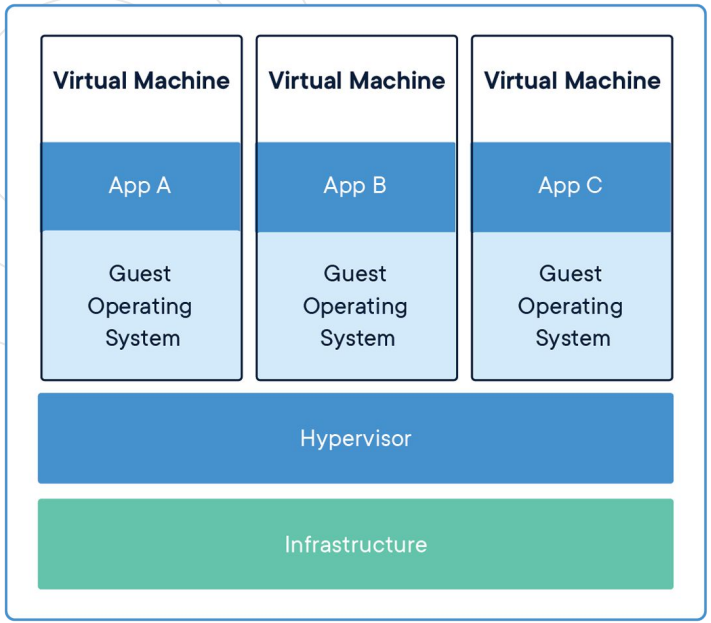
https://www.linkedin.com/in/golazynani/

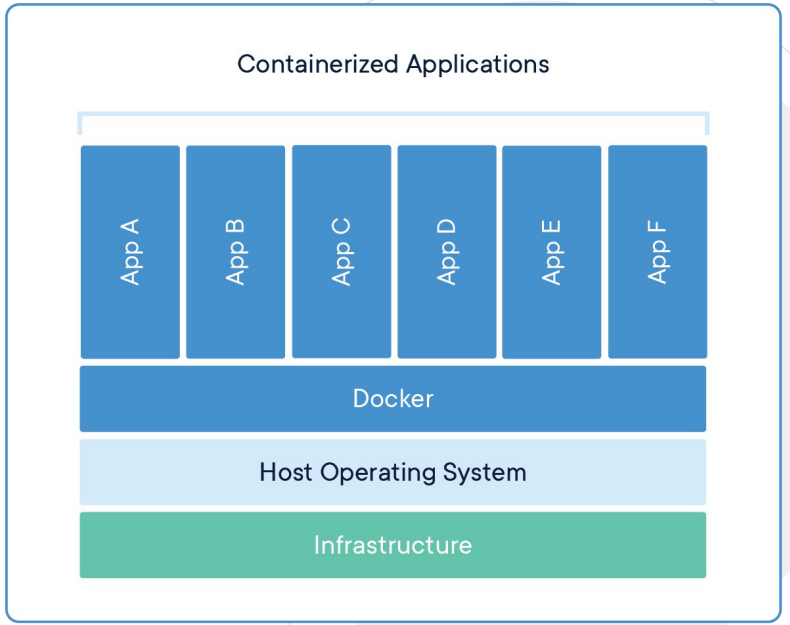
Modern Application Requirements



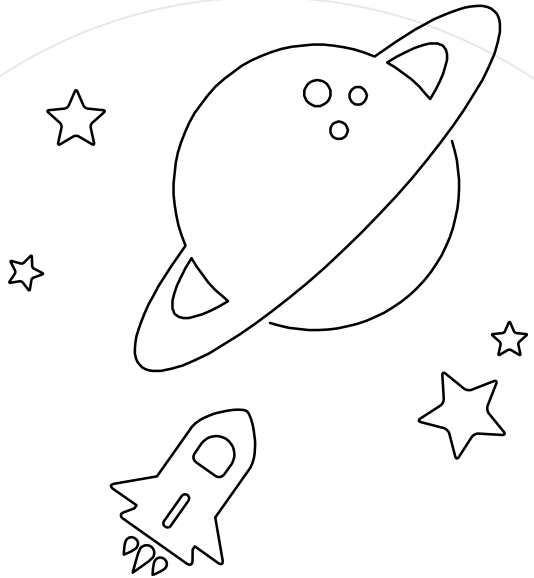
“ Why Containers?
Why Orchestrator?
Why Container Orchestrator?

Containers?





Containers and virtual machines have similar resource isolation and allocation benefits, but function differently because containers virtualize the operating system instead of hardware. Containers are more portable and efficient.



Enter the Kubernetes

Kubernetes is a :

Very active open source project

2k+ Contributers

75k+ Commits

Apache 2 licensed

Written in Go (or Golang)

Hosted by Cloud Native Computing Foundation(CNCF)

What does an Orchestrator DO?

For a Software

- Ensure that the software are spread across multiple servers for High Availability
- Rolling
- Scaling

This is what **Kubernetes** does as a Container Orchestrator

What Kubernetes is Not

- Kubernetes is not a traditional, all-inclusive PaaS (Platform as a Service) system. Since Kubernetes operates at the container level rather than at the hardware level, it provides some generally applicable features common to PaaS offerings, such as deployment, scaling, load balancing, logging, and monitoring
- Kubernetes is not monolithic, and these default solutions are optional and pluggable

K8s

1

The name **Kubernetes** originates from Greek, meaning *helmsman* or *pilot*, and is the root of *governor* and cybernetic. *K8s* is an abbreviation derived by replacing the 8 letters “ubernete” with “8”

Considered as:

Kubernetes has a number of features. It can be thought of as:

- a container platform
- a microservices platform
- a portable cloud platform and a lot more.

Few concepts of K8s

Pods

Container Runtime

Nodes

Cluster

Master - Slave Architecture

Desired vs Current State

What kind of Workloads can be run on K8s?

Short Q/A

Wanna See It Work?


Single Node
Kubernetes

Why Deployments?

Scaling

With a deployment, you can specify the number of replicas you want, and deployment will scale-up or scale-down pods to meet that number of replicas

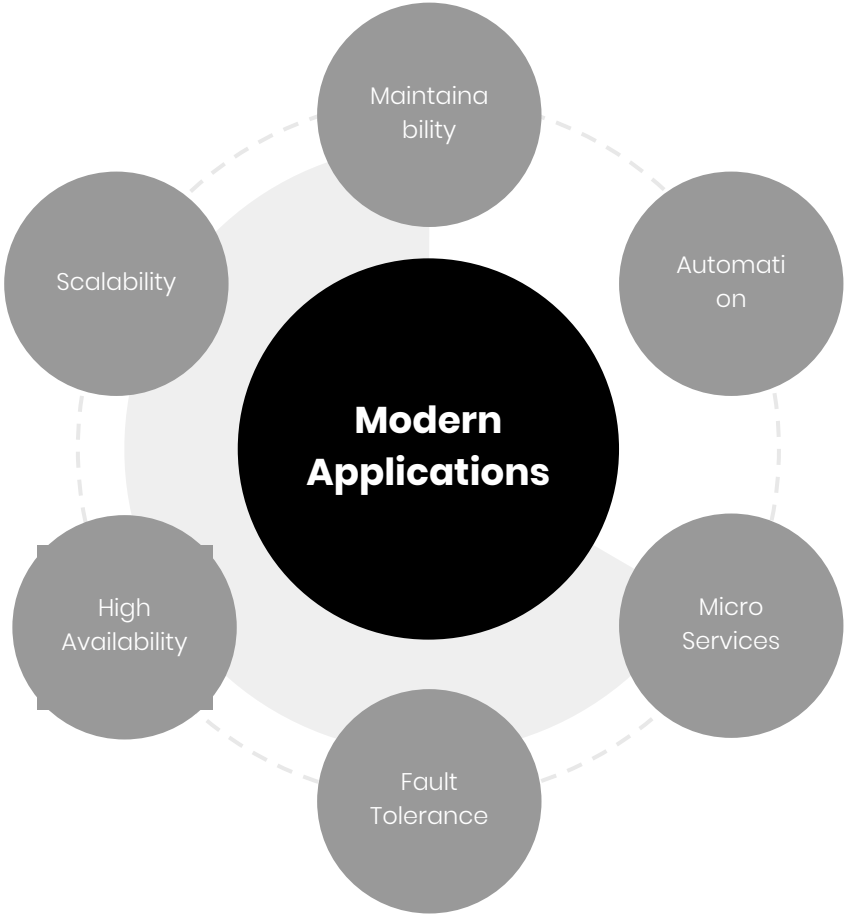
Rolling Updates

With deployments, changing the deployment image will gradually migrate the existing containers to the latest version of the container

Self-Healing

If a Pod is down for any reason, the deployment will immediately spin up a new one to replace it

Did we answer these requirements ?



Hope I made Sense

You can find my presentation at

https://github.com/imjuststarting/presentations/

Thanks!

Any questions?

You can find me at

- @golazynani on Twitter
- @LazyNani on Kubernetes Slack
- https://www.linkedin.com/in/golazynani/