K8s 101 Workshop

Cloud CSE Team

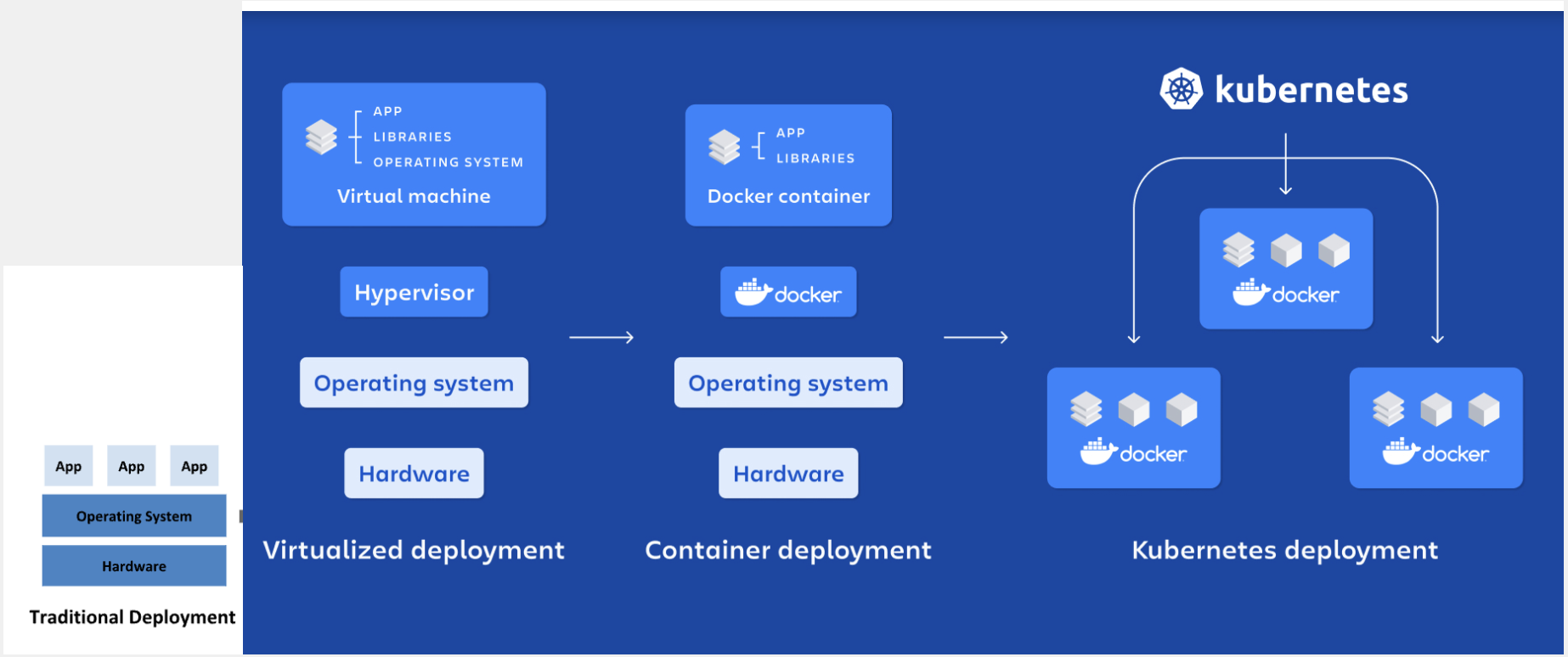
Andy Wang (wandy@fortinet.com) - Consulting Systems Engineer
Srija Reddy Allam (sallam@fortinet.com) - Cloud DevOps Architect

Goal of K8s 101 Workshop

- The Goal of this workshop is to introduce Kubernetes
- Get Familiarize with the K8s Terms and Concepts
- Understand difference between Managed and Self managed/unmanaged Kubernetes
- Deploy a Kubernetes Cluster with Linux VM’s in Azure
- Hands on experience to deploy, scale, upgrade and access an application.

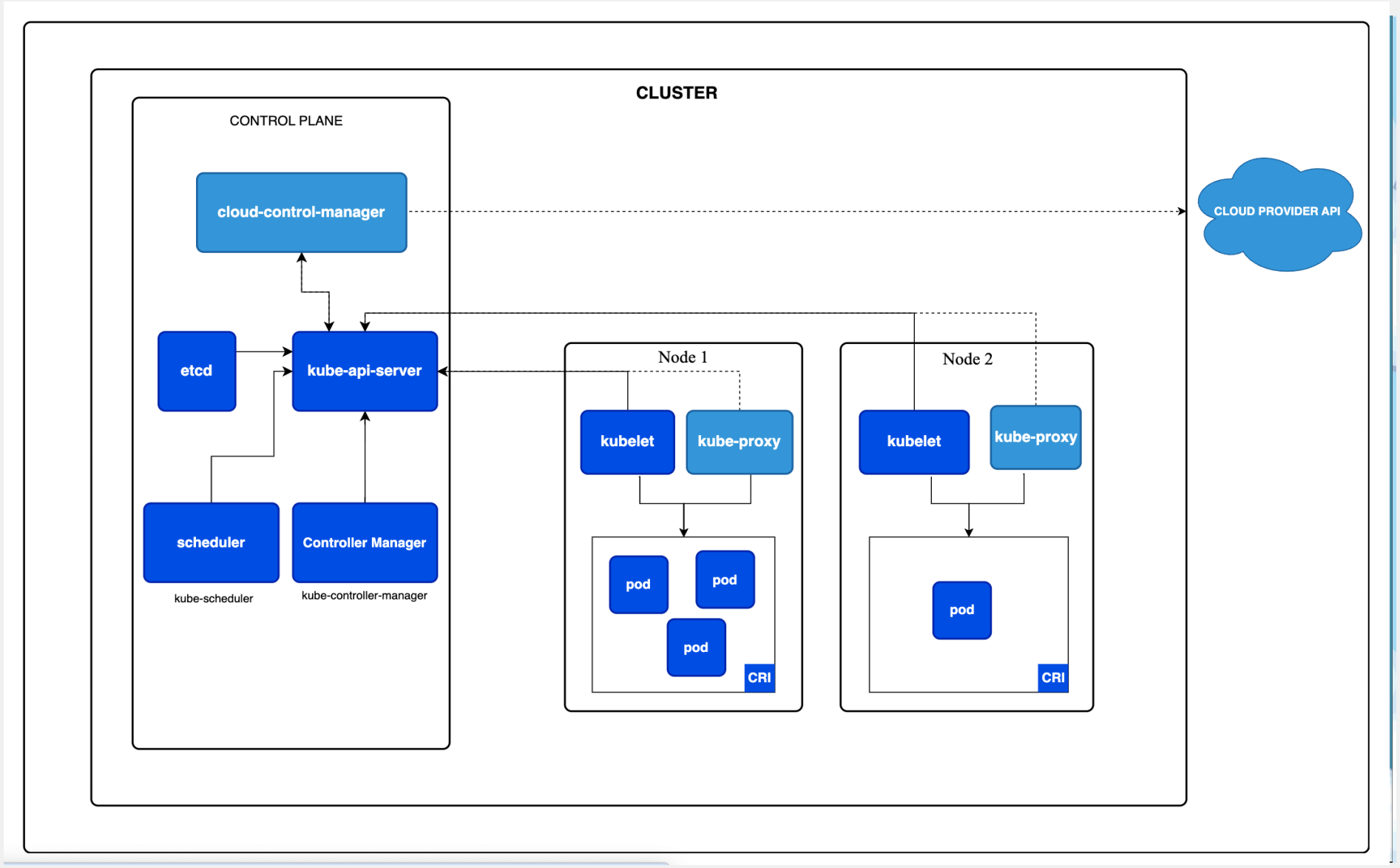
© Fortinet Inc. All Rights Reserved.

Going back in time..



© Fortinet Inc. All Rights Reserved.

Intro to K8s



© Fortinet Inc. All Rights Reserved.

Before we continue

- Workshop Link: https://tinyurl.com/fortik8intro
- Please deploy the resources before we get started:

https://fortinetcloudcse.github.io/k8s-101-workshop/02_quickstart_overview_faq.html

- TinyURL: https://tinyurl.com/fortik8sch1
- Please complete Task1 (Set up Azure Cloud Shell) and Task 2 (Run Terraform)

© Fortinet Inc. All Rights Reserved.

K8s Overview

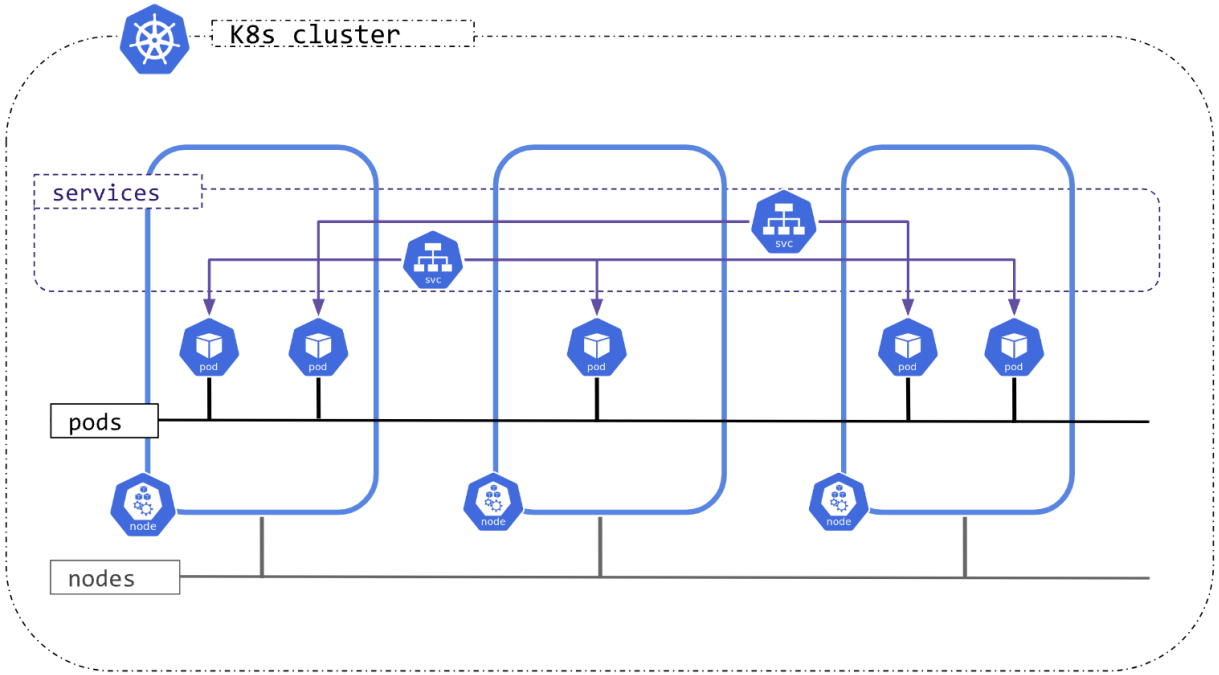
Kubernetes sometimes shortened to K8s with the 8 standing for the number of letters between the “K” and the “s”

Managed Kubernetes

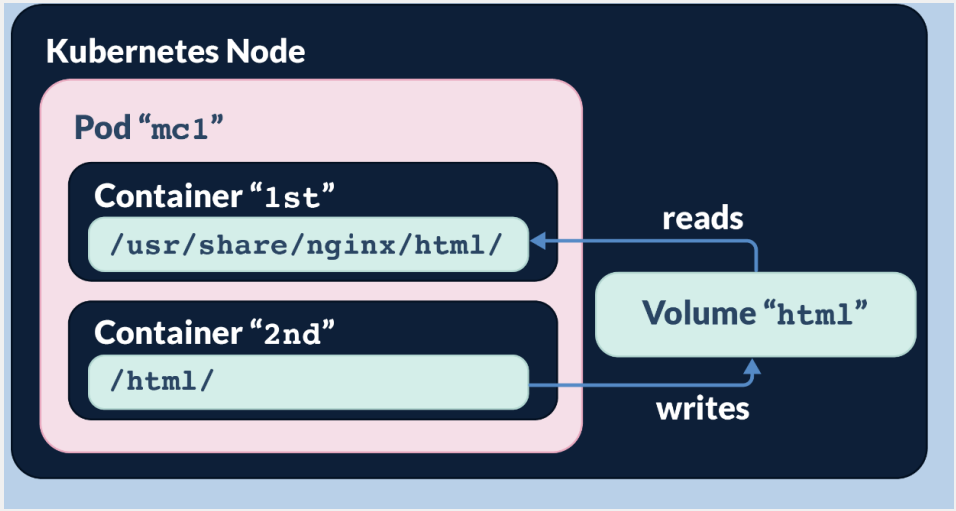
Feature	AKS (Azure Kubernetes Service)	GKE (Google Kubernetes Engine)	EKS (Amazon Elastic Kubernetes Service)
Cloud Provider	Microsoft Azure	Google Cloud Platform (GCP)	Amazon Web Services (AWS)
Managed Service	Fully managed	Fully managed	Fully managed
Infrastructure	Built on Azure infrastructure	Built on Google infrastructure	Built on AWS infrastructure
Scaling	Automated scaling	Automated scaling	Automated scaling
Monitoring	Built-in monitoring	Built-in monitoring	Built-in monitoring
Authentication/ Authorization	Integration with Azure Active Directory	Integration with Google Cloud IAM	Integration with AWS IAM
Container Support	Windows and Linux containers	Linux containers only	Linux containers only
Integration	Tight integration with Azure services	Seamless integration with GCP services	Seamless integration with AWS services
Updates	Automated updates	Automated updates	Automated updates
Support for Fargate	No	No	Yes (Allows running containers without managing the underlying infrastructure)
Pricing Model	Pay-as-you-go	Pay-as-you-go	Pay-as-you-go

© Fortinet Inc. All Rights Reserved.

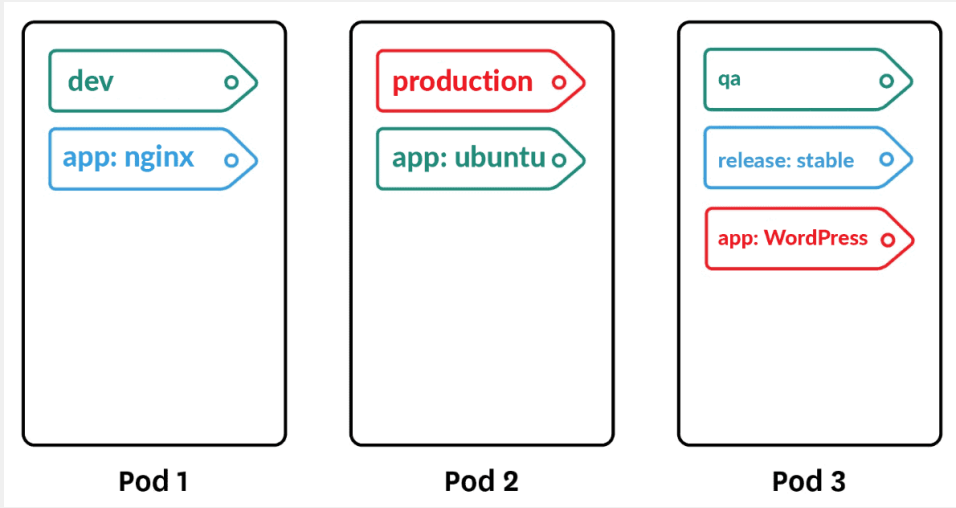
Node, Pod, SVC



Inside a Pod



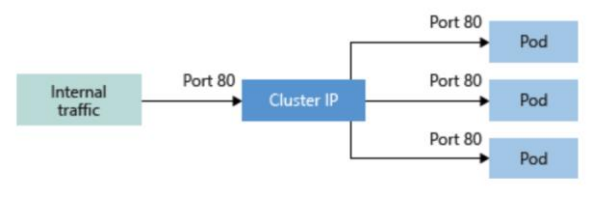
Labels



© Fortinet Inc. All Rights Reserved.

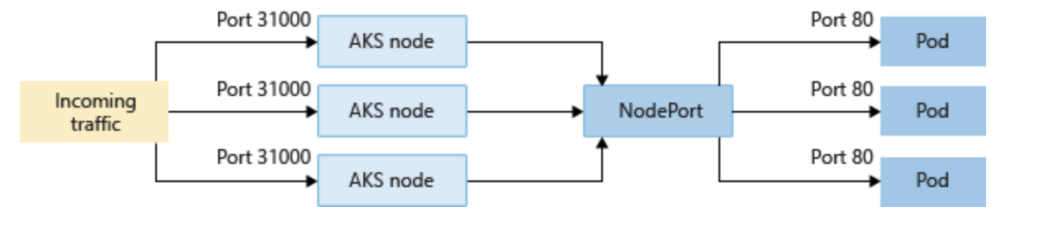
Service Types

- ClusterIP



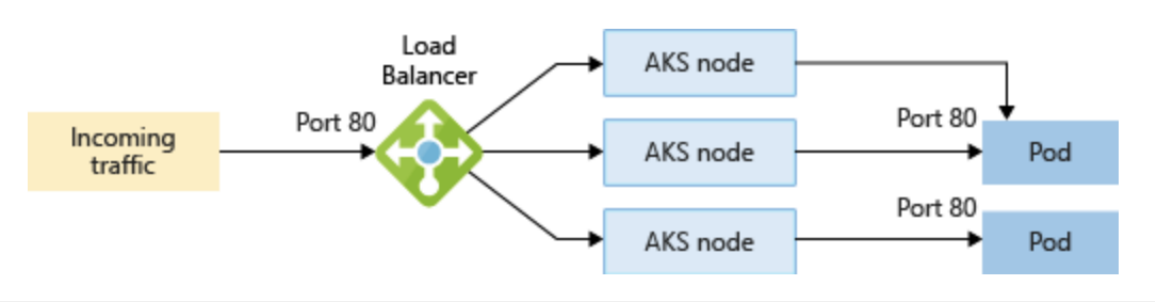

```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376
```

- NodePort




```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  type: NodePort
  selector:
    app.kubernetes.io/name: MyApp
  ports:
    - port: 80
      # By default and for convenience, the `targetPort` is set to
      # the same value as the `port` field.
      targetPort: 80
      # Optional field
      # By default and for convenience, the Kubernetes control plane
      # will allocate a port from a range (default: 30000-32767)
      nodePort: 30007
```

- LoadBalancer




```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-service
spec:
  selector:
    app.kubernetes.io/name: MyApp
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376
  clusterIP: 10.0.171.239
  type: LoadBalancer
status:
  loadBalancer:
    ingress:
    - ip: 192.0.2.127
```

© Fortinet Inc. All Rights Reserved.

Deployment and Replica Set

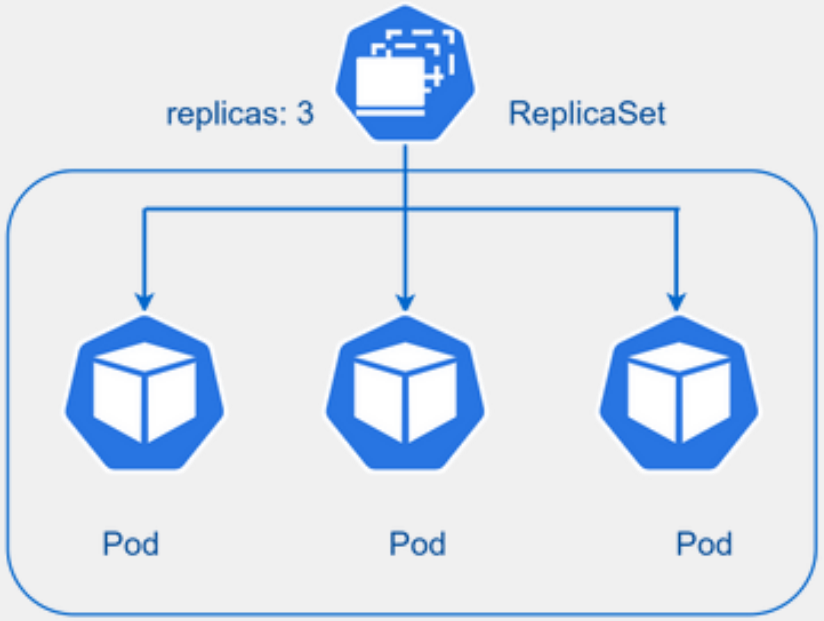


Fig: ReplicaSet

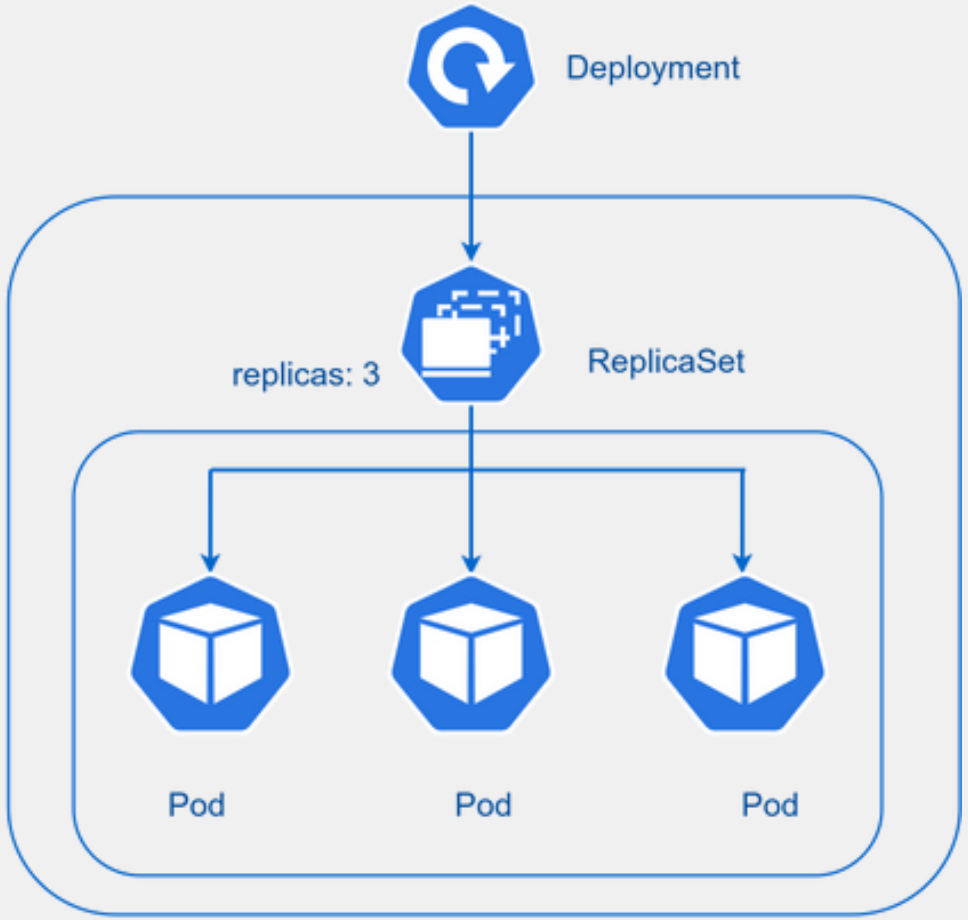


Fig: Deployment

#goglides

© Fortinet Inc. All Rights Reserved.

Why is K8s widely used?

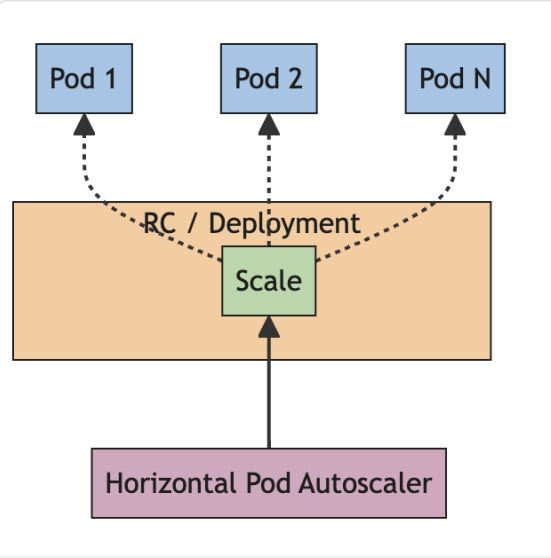
- **Scalability**:

Kubernetes is highly scalable, both in terms of application scalability and cluster scalability. It can handle large-scale deployments with thousands of containers across multiple nodes, enabling organizations to scale their applications effortlessly as demand grows.

- **High Availability**:

Kubernetes ensures high availability of applications by automatically rescheduling pods in case of node failures, performing health checks on containers, and enabling rolling updates without downtime. This ensures that applications are always available and responsive to user requests.

- **Horizontal Pod Autoscaler** aims to automatically scale the workload to meet the demand.



© Fortinet Inc. All Rights Reserved.

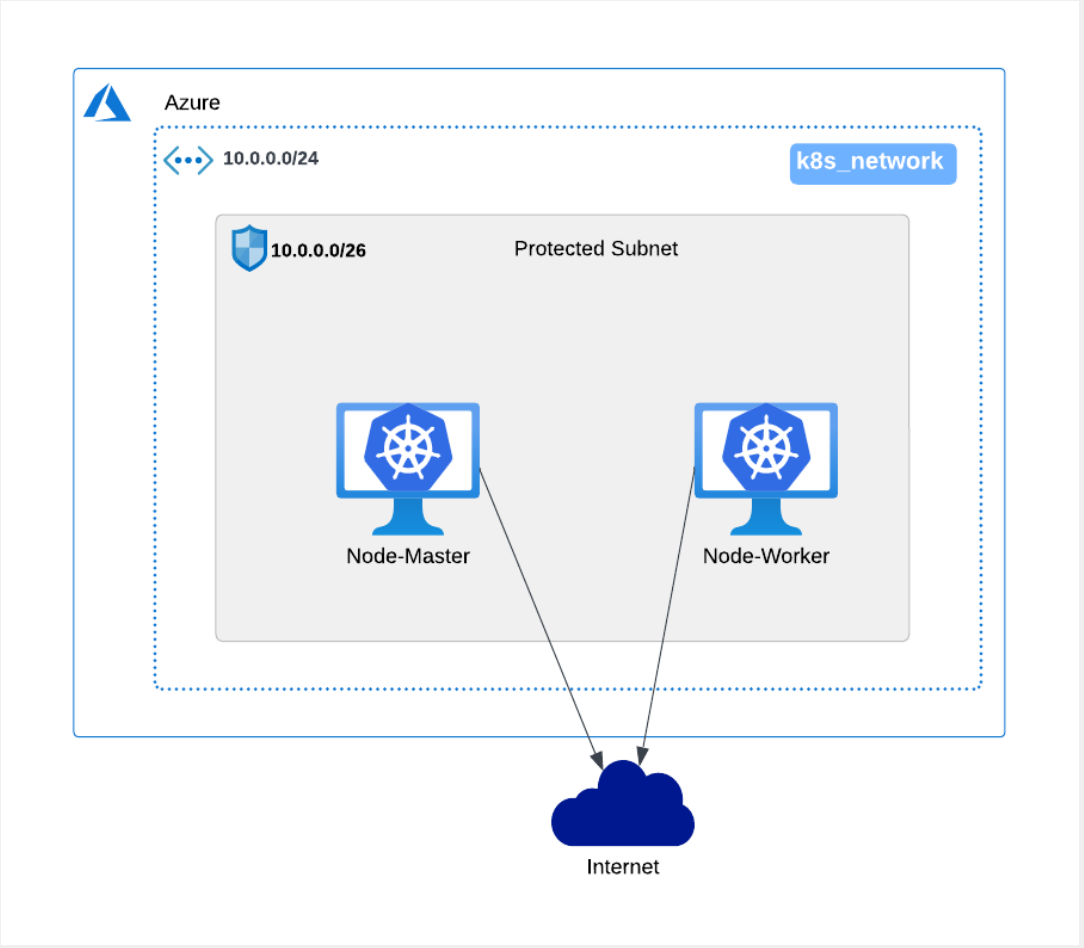
HPA example

```
cat << EOF | kubectl apply -f -
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: nginx-hpa
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: nginx-deployment
  minReplicas: 2
  maxReplicas: 10
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 50
```

© Fortinet Inc. All Rights Reserved.

Finally, Unmanaged/Self managed K8s cluster

- An unmanaged Kubernetes (K8s) cluster refers to a Kubernetes cluster that you set up and manage manually without relying on a managed Kubernetes service provided by a cloud provider or a third-party vendor.
- Usually not built manually.
- Will be part of autoscaling group on Public cloud platforms, CI/CD pipeline
- **Scaling application** automatically in case of large requests to the app with autoscaling



© Fortinet Inc. All Rights Reserved.

Concepts

- Pod: The smallest deployable units in Kubernetes, containing one or more containers with shared storage, network, and specifications for how to run the containers.
- ConfigMap and Secrets: Configuration data and sensitive information stored separately from the main application code.
- Deployments: Managing application lifecycles, ensuring consistent and reliable deployment and scaling.
- Autoscaling: Automatically adjusting the number of instances based on workload to maintain performance and efficiency.

© Fortinet Inc. All Rights Reserved.

Concepts

- Rolling Deployment Upgrades: Updating application versions gradually to minimize downtime and risk.
- Exposing application with Load balancer Service: Distributing incoming traffic across multiple instances to improve reliability and performance.
- Using Ingress Controller to route traffic through multiple services: Directing incoming traffic to specific services based on defined rules for more flexible and efficient routing.

© Fortinet Inc. All Rights Reserved.

Q & A

Intro to K8s

Lab Time

Thank You