

# Azure Kubernetes Service (AKS)



Moti Malka, DevOps  
Lead



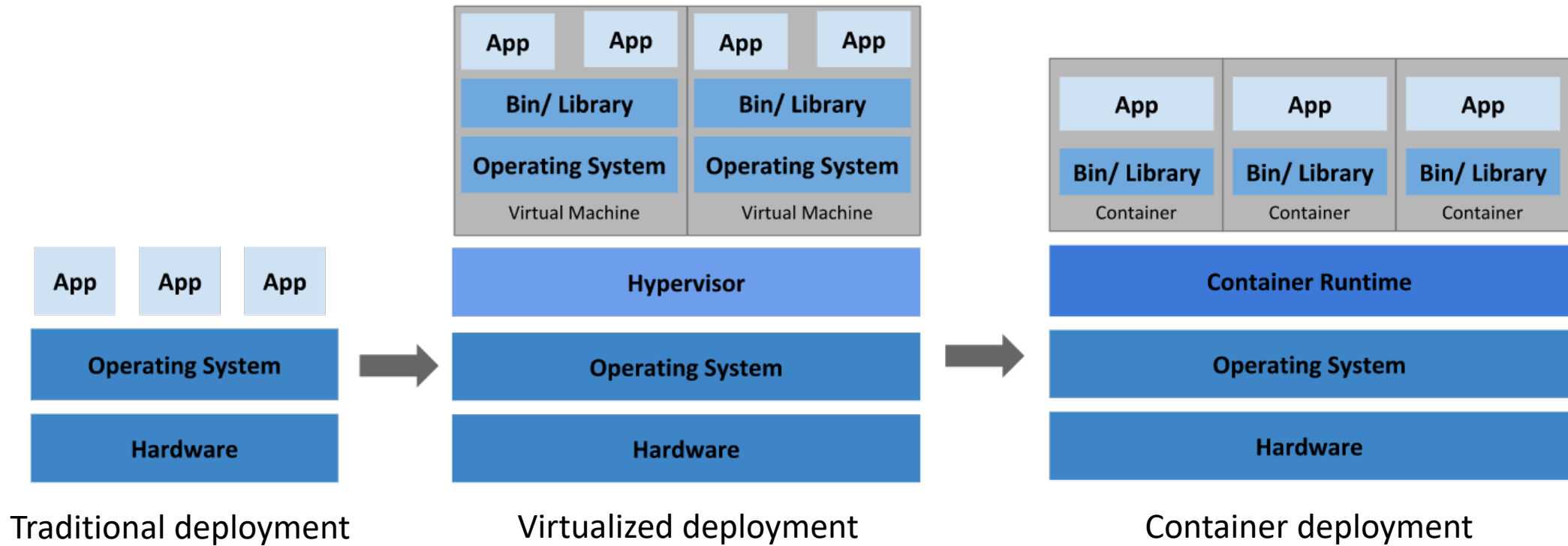
# Agenda

- » Containers Overview
- » Kubernetes Overview
- » Containerized application deployment on azure
- » AKS Overview
- » Interacting with AKS
- » AKS + IaC
- » Scale with AKS + Demo
- » AKS Features
- » AKS add-on

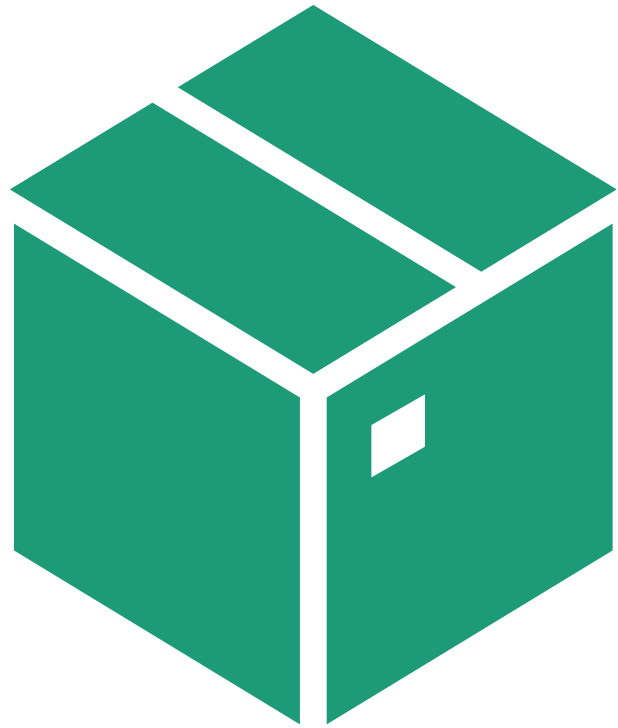
## Do You Really Need Kubernetes?

- 
- K8S is helpful if you are dealing with many containers.
  - You have a DevOps team to manage it.
  - Cost: Kubernetes is NOT Serverless.





Going back in time

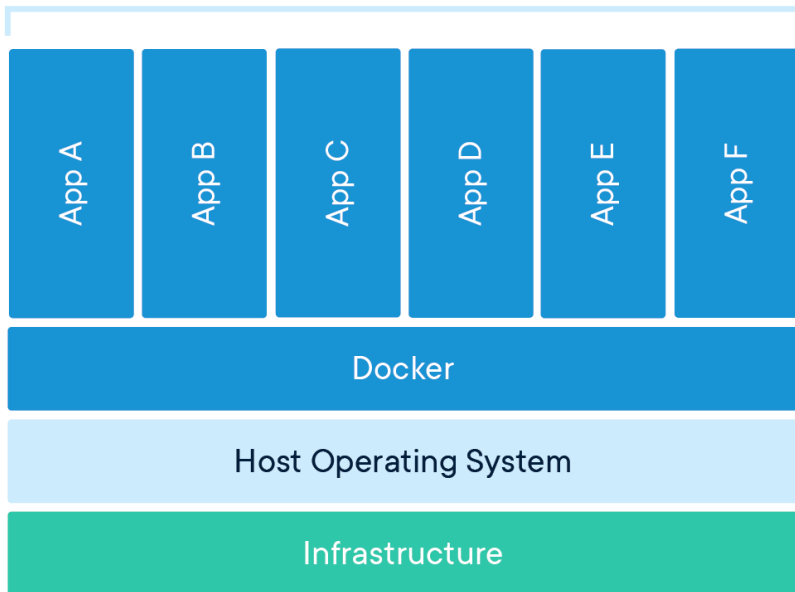


WHAT ARE CONTAINERS,  
AND WHAT PROBLEMS  
DOES IT SOLVE?





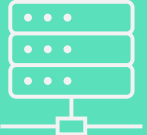

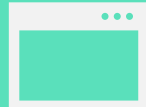

# Container Overview

"Containers are an operating system virtualization technology used to package applications and their dependencies and run them in isolated environments. They provide a lightweight method of packaging and deploying applications in a standardized way across many different types of infrastructure\OS."

## Containerized Applications



# The complexity





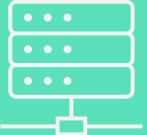

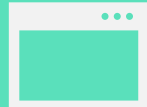

Static Website	?	?	?	?	?	?	?	?
Web Frontend	?	?	?	?	?	?	?	?
Background Workers	?	?	?	?	?	?	?	?
API	?	?	?	?	?	?	?	?
Functions	?	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?	?
	Desktop 	Test 	Production 	Cloud 	Data Center 	Mainframe 	Windows Server 	Edge Device 

# CONTAINERS REDUCES THE COMPLEXITY

Static Website
Web Frontend
Background Workers
API
Functions
Queue



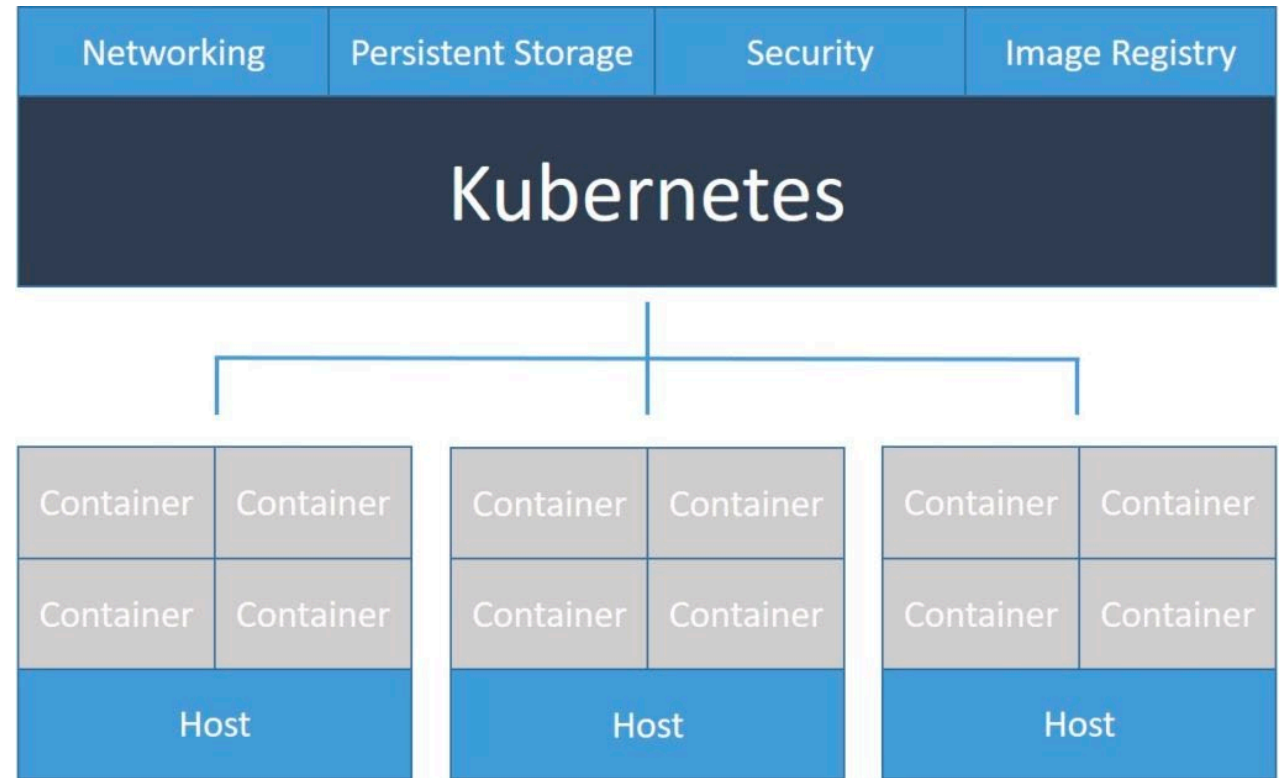
This Photo by Unknown Author is licensed under [CC BY-NC](#)

Desktop	Test	Production	Cloud	Data Center	Mainframe	Windows Server	Edge Device
							



# Kubernetes Overview

"Kubernetes is an open-source container orchestration platform that automates many of the manual processes involved in deploying, managing, and scaling containerized applications."



# Kubernetes Components

## Control Plane (Master Node) components:

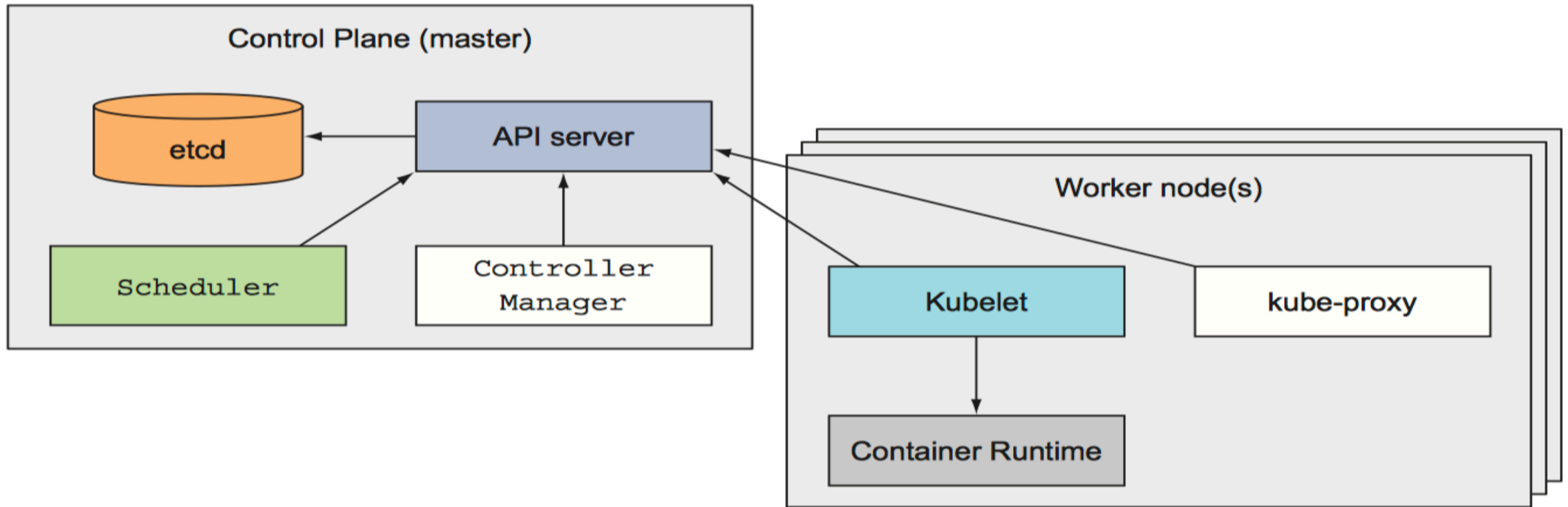
- API Server.
- Scheduler.
- Controller Manager.
- Etcd.

## Data Plane (Worker Node) components:

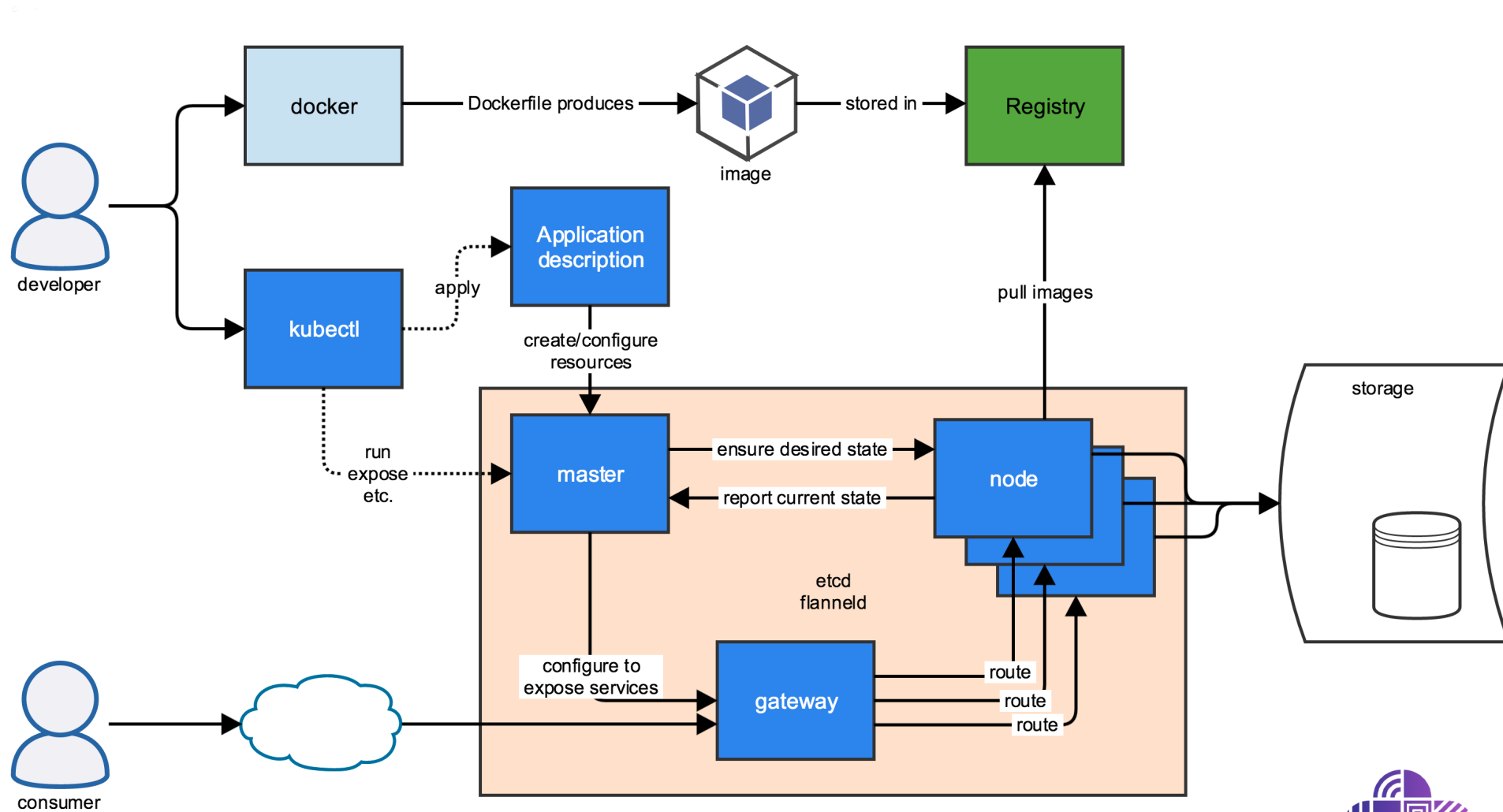
- Kubelet.
- Kube-Proxy.
- Container Runtime.



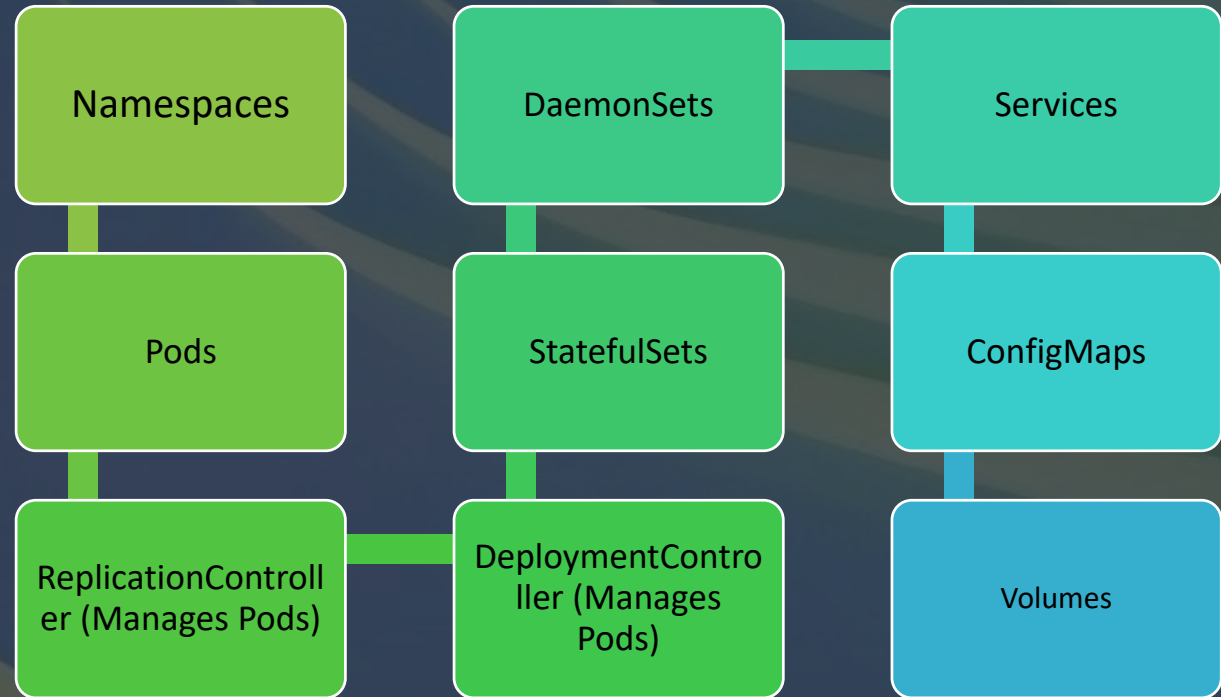
# Kubernetes Components



# Kubernetes Flow



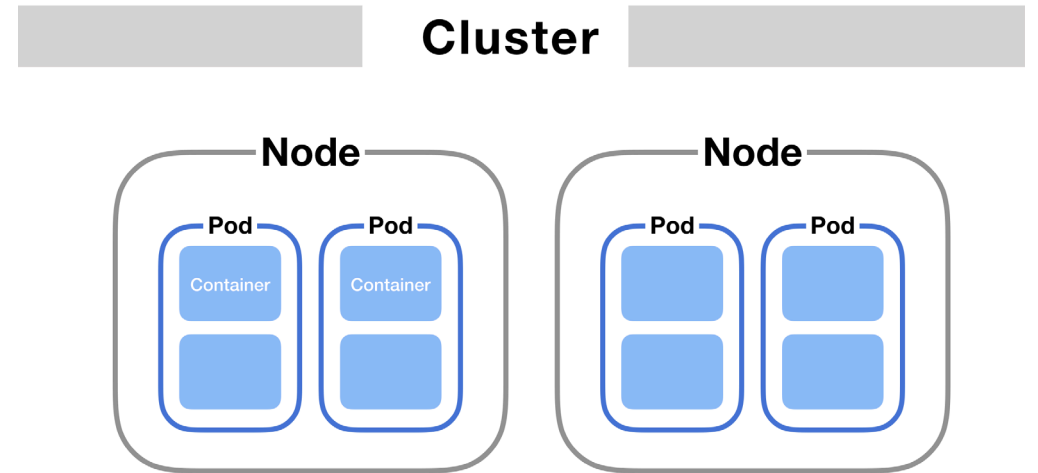
# Kubernetes Resources\Objects

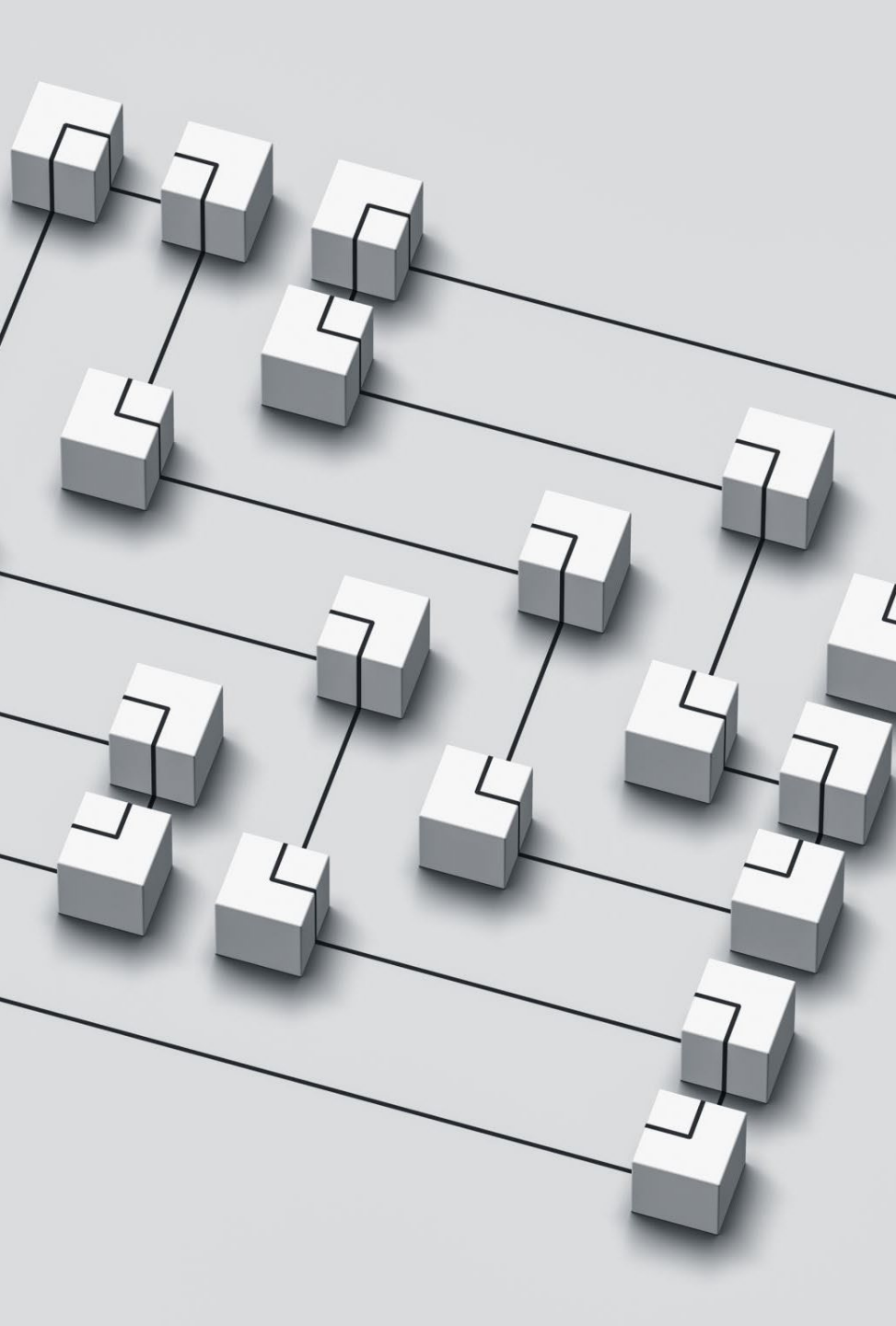


# Nodes

A node (worker) is a machine where containers (workloads) are deployed.

- **Kubelet:**  
is responsible for the running state of each node, ensuring that all containers on the node are healthy.
- **Kube-proxy:**  
is responsible for routing traffic to the appropriate container based on IP and port number of the incoming request





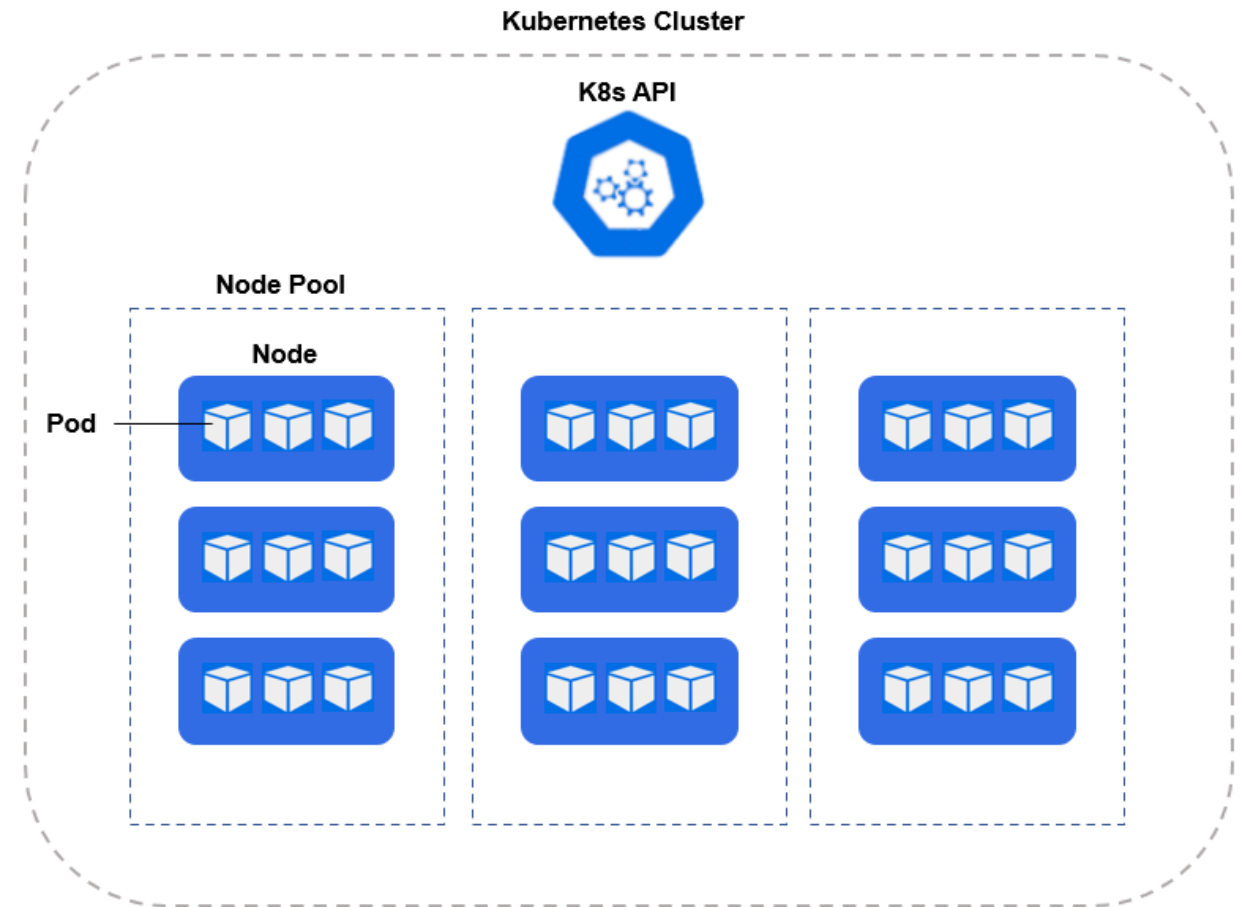
# Namespaces

---

- Namespaces provide a mechanism for isolating groups of resources within a single cluster
- Many users spread across multiple teams
- Projects
- Separating environments like development, test, and production

# Pod

- Pods are the smallest deployable units of computing that you can create and manage in Kubernetes.
- In general, we manage pod by using:
  - Deployment
  - StatefulSet
  - DaemonSet



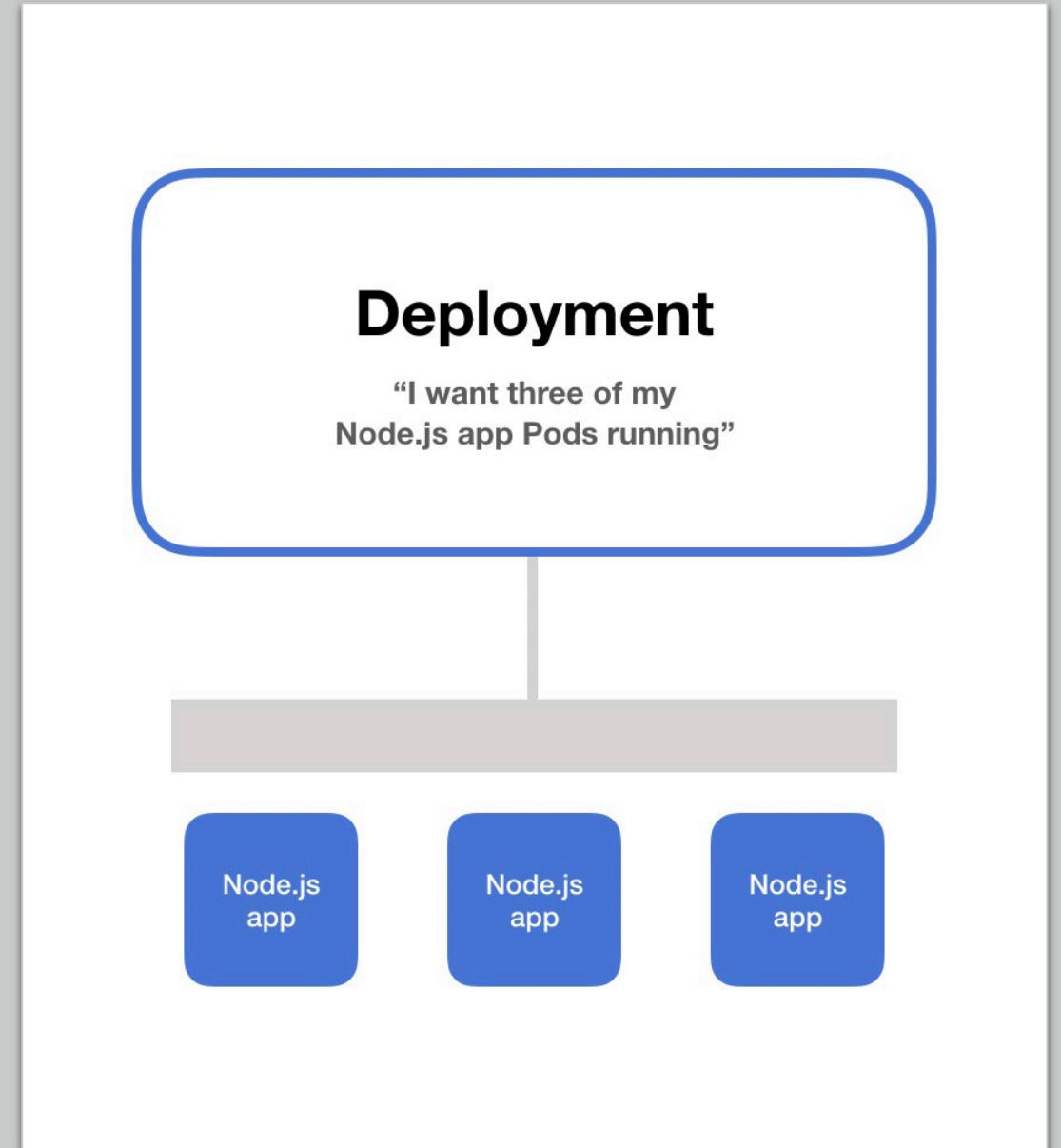


# Pod Phase

Value	Description
Pending	The Pod has been accepted by the Kubernetes cluster, but one or more of the containers has not been set up and made ready to run. This includes time a Pod spends waiting to be scheduled as well as the time spent downloading container images over the network.
Running	The Pod has been bound to a node, and all of the containers have been created. At least one container is still running, or is in the process of starting or restarting.
Succeeded	All containers in the Pod have terminated in success, and will not be restarted.
Failed	All containers in the Pod have terminated, and at least one container has terminated in failure. That is, the container either exited with non-zero status or was terminated by the system.
Unknown	For some reason the state of the Pod could not be obtained. This phase typically occurs due to an error in communicating with the node where the Pod should be running.

# Deployment

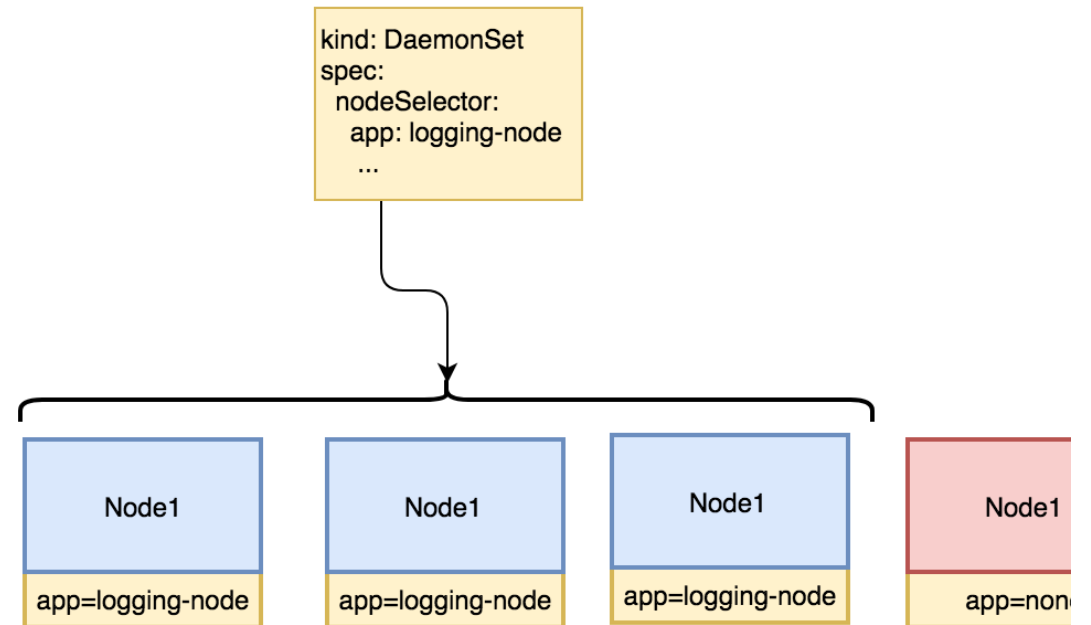
- A **Deployment** is used to tell Kubernetes how to create or modify instances of the pods that hold a containerized application.
- **Deployments** can scale the number of replica pods, enable the rollout of updated code in a controlled manner, or roll back to an earlier deployment version if necessary.



# DaemonSets

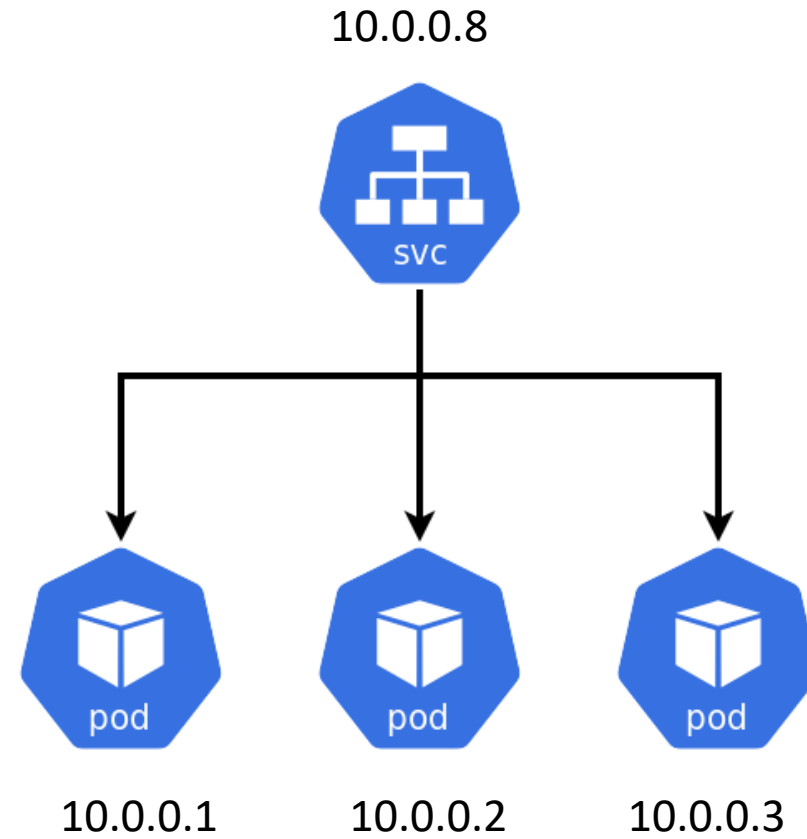
A DaemonSet ensures that all (or some) Nodes run a copy of a Pod, for example:

- running a logs collection daemon on every node
- running a node monitoring daemon on every node



# Services

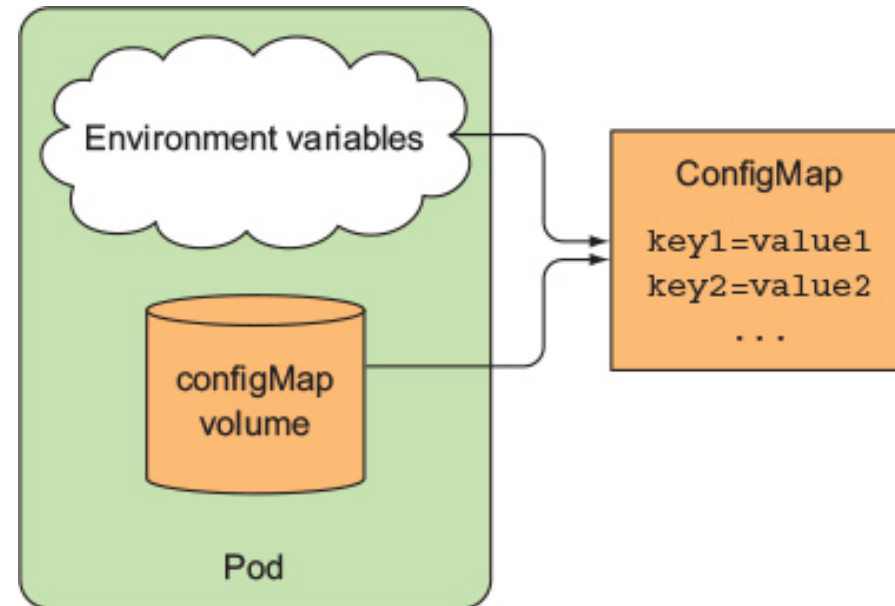
expose an application running on a set of Pods as a network service



# ConfigMaps

---

- A ConfigMap used to store non-confidential data in key-value pairs.
- Pods can consume ConfigMaps as:
  - environment variables
  - command-line arguments
  - as configuration files in a volume.

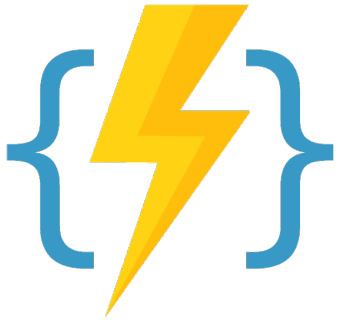




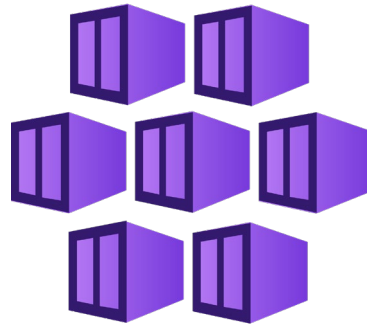
**Web App**



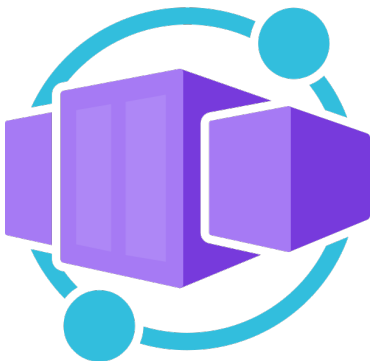
**Container Instance**



**Azure Function**



**AKS**

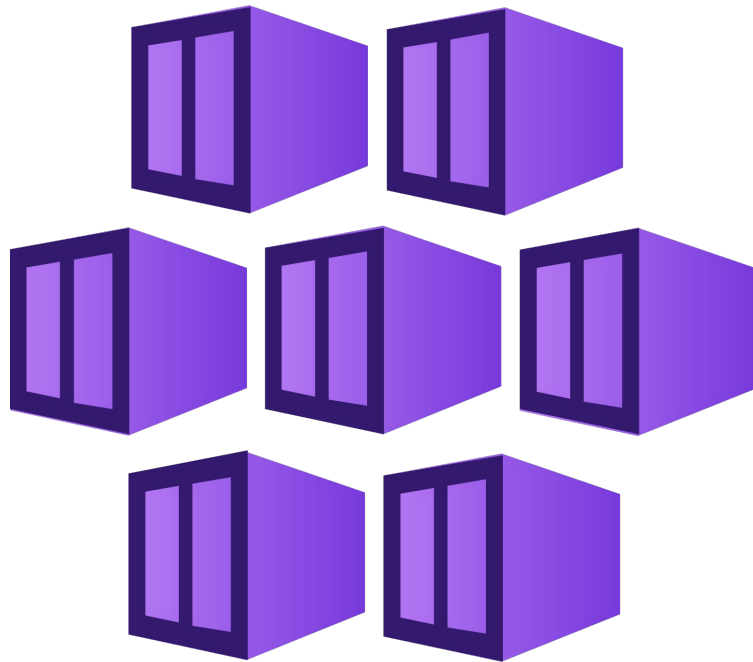


**Container Apps**



**VM**

containerized  
applications  
deployment  
in azure cloud



# AKS Overview



Managed Kubernetes service (means it let's you quickly deploy and manage Kubernetes cluster on azure).

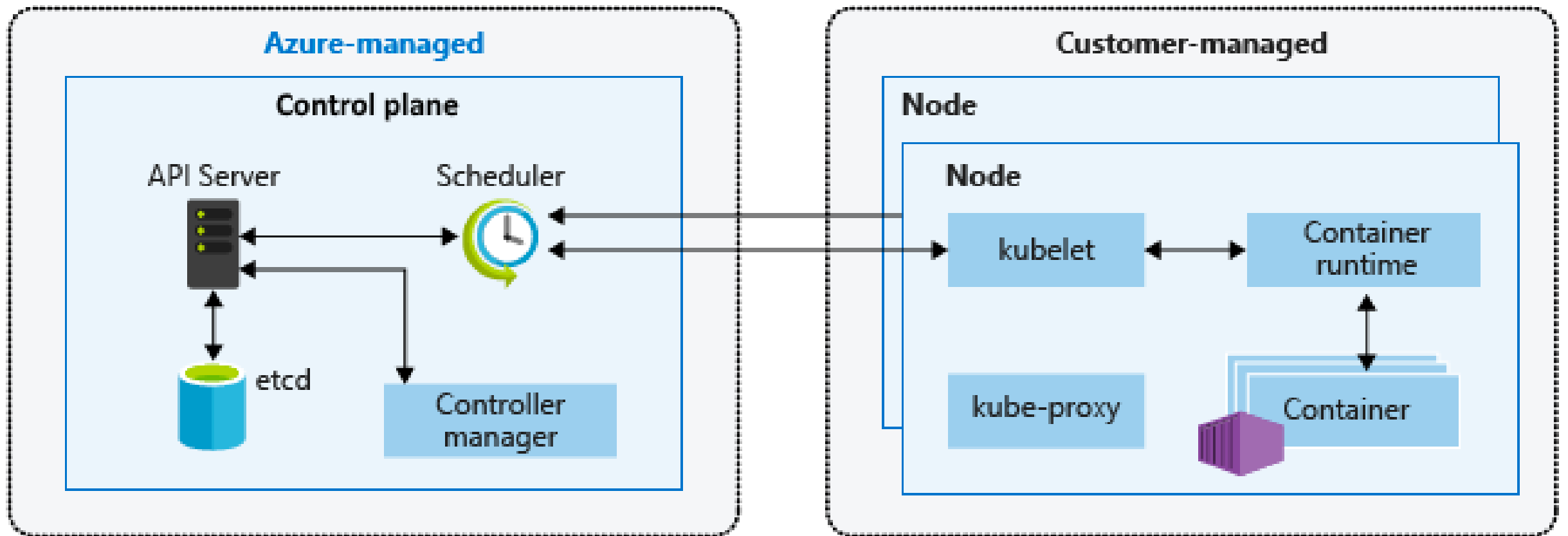


Kubernetes master nodes are managed by azure, while worker nodes are managed by customer.



it reduce the setup & operational complexity of Kubernetes for production workloads

# AKS cluster architecture





# AKS Features

## Manage K8S Control plane.

- Master Nodes.
- Etcd.
- Backup.
- Security.

## Cluster Configuration.

- VM Configuration: CPU, RAM, Type of Storage.
- Node Pools.

## Identity, Access Control & Security:

- [Azure Active Directory](#).
- [RBAC](#)
- Virtual Networks, Subnets.
- K8S Version Upgrades, OS Security Patches.

## Scaling:

- Cluster/Node pools scale (in & out).
- Application Scale

# AKS Features

## Monitoring & Logging.

- Azure Monitor.
- Log Analytics.
- Container Insight.

## Storage

- Azure Disks.
- Azure Storage Account.

# Interacting with AKS

## Azure Portal UI

## CLI

- Azure CLI
- Azure Cloud Shell
- API

## IaC (infrastructure as code)

- Azure Resource Manager (ARM) Template
- [Terraform](#)
- Pulumi

# Why IaC?

Manage any  
infrastructure:

Azure  
AWS  
GCP  
K8S  
1000+ provider

Track your  
infrastructure:

Plan & approval  
Save state of a live environment

Automate  
changes:

Declarative Configuration

Standardize  
configurations:

Modules  
Best Practices.

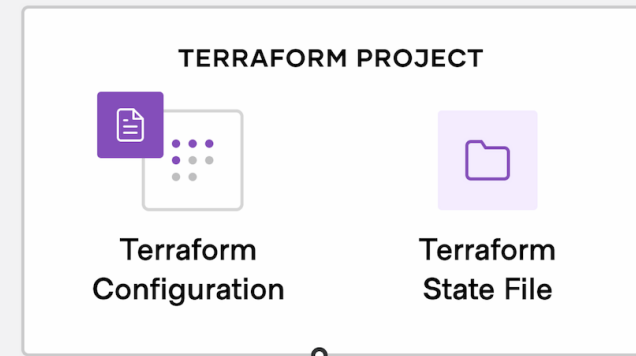
Security:

Limit access.  
Environment Scan

# Terraform: Infrastructure as code

## Write

Define infrastructure in configuration files



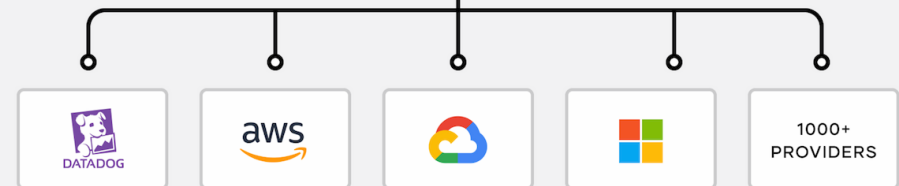
## Plan

Review the changes  
Terraform will make to  
your infrastructure

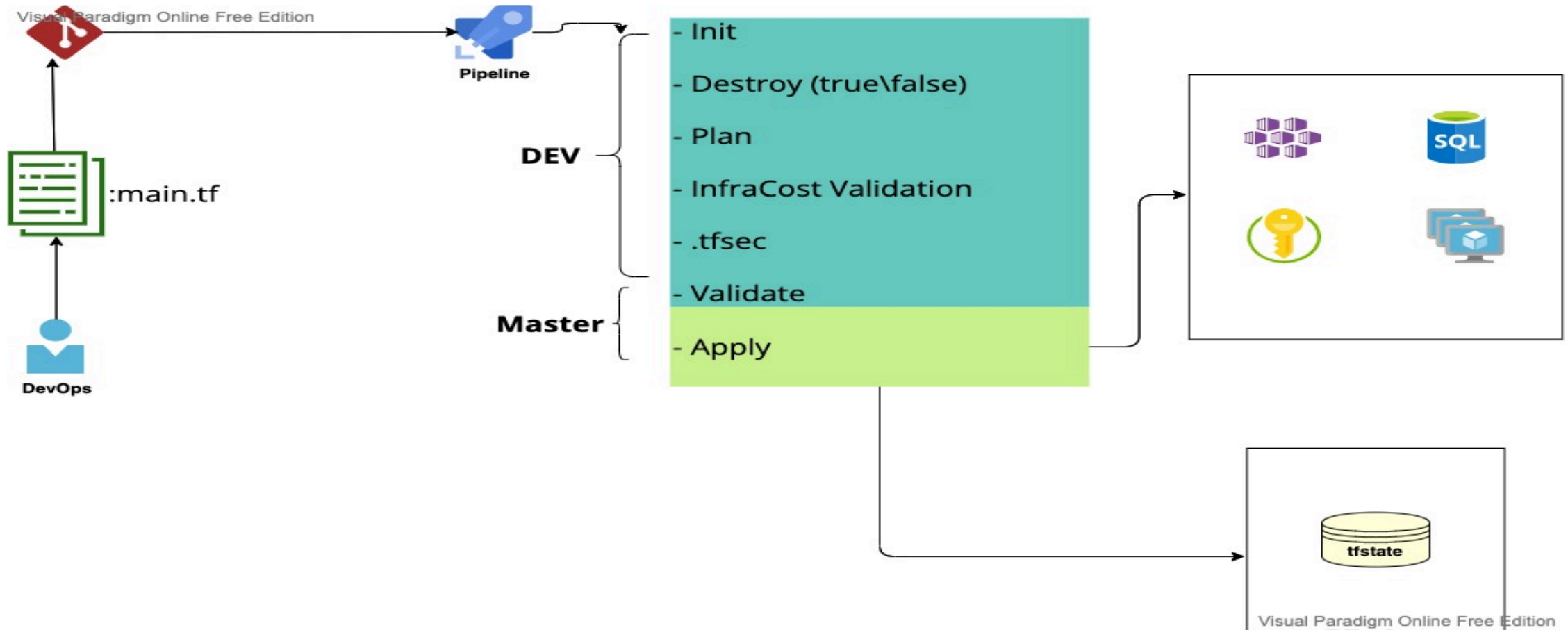
```
$ terraform plan
...
Terraform will perform
the following actions
```

## Apply

Terraform provisions  
your infrastructure and  
updates the state file.



# IaC Flow

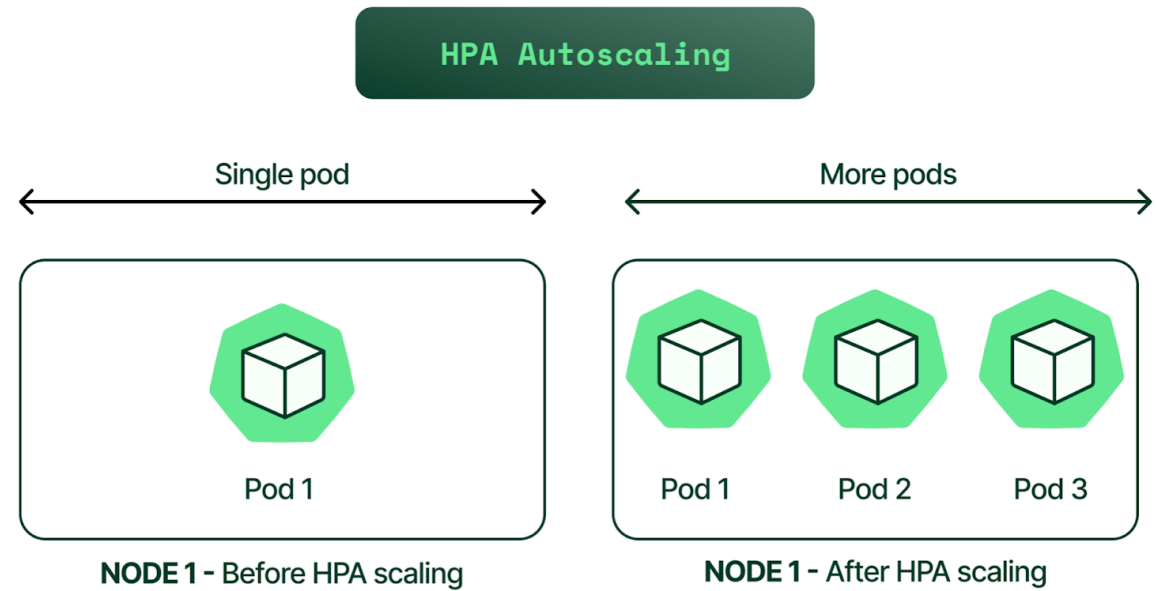
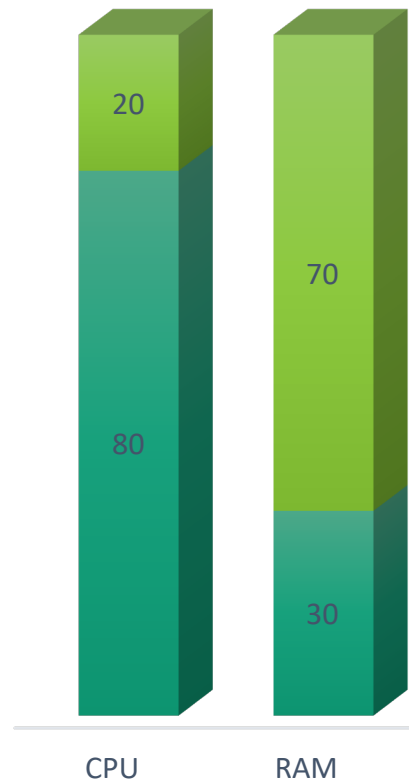


# AKS Scale pods Or Nodes

- **Manually scale pods or nodes.**
- **Horizontal pod auto scaler (HPA).**
- **Vertical pod auto scaler (VPA)**
- **Cluster auto scaler.**

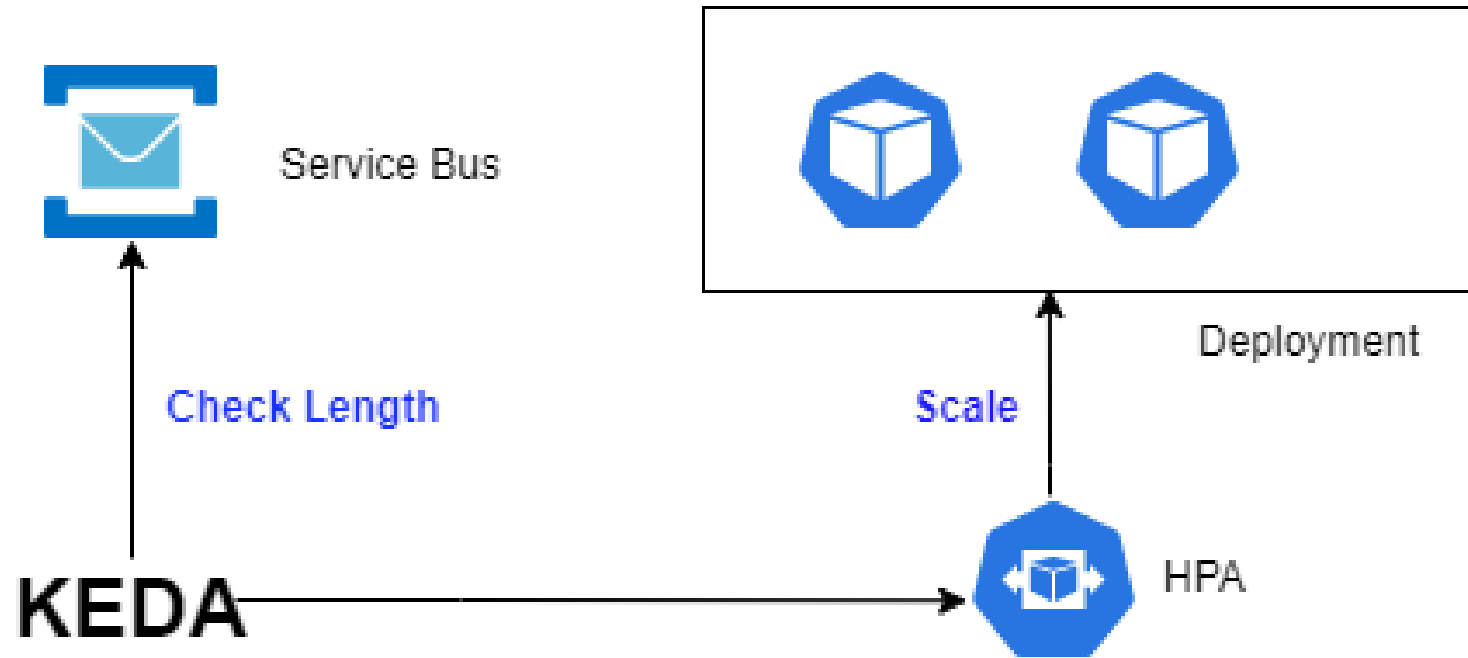


# Horizontal Pod Autoscaler





# KEDA: Scale Demo



# KEDA: Kubernetes Event-driven Autoscaling



- “Scaling of any container in Kubernetes based on the number of events needing to be processed”
- [Azure Application Insights](#): Scale applications based on Azure Application Insights metrics.
- [Azure Blob Storage](#): Scale applications based on the count of blobs in a given Azure Blob Storage container.
- [Azure Service Bus](#): Scale applications based on Azure Service Bus Queues or Topics.
- [Azure Log Analytics](#): Scale applications based on Azure Log Analytics query results.

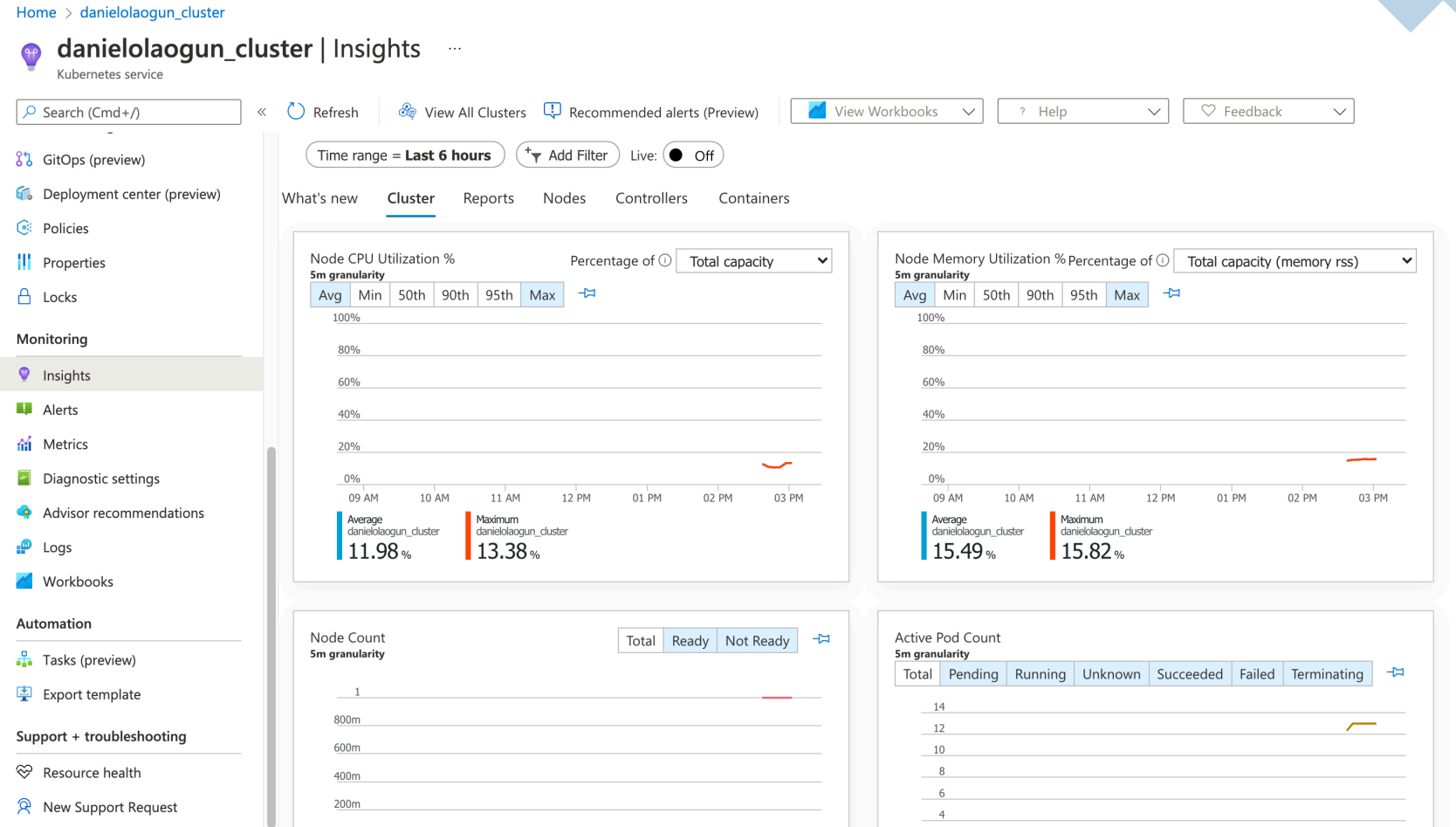
# AKS Add-ons

- **Monitoring:** Use Container Insights monitoring with your AKS cluster.
- **Azure Policy:** Use Azure Policy for AKS, which enables at-scale enforcements and safeguards on your clusters in a centralized, consistent manner.
- **Ingress appgw:** Use Application Gateway Ingress Controller with your AKS cluster.
- **azure-keyvault-secrets-provider:** Use the Azure Keyvault Secrets Provider add-on.

# AKS Monitoring

## Container Insights:

- information and analytics
- node resource utilization
- failed pods



# AKS Monitoring

## Logs:

- Collect logs from nodes, pod, control plane, and containers.
- Viewed and analyzed on Dashboard.
- Configure alert rules

The screenshot displays the Azure portal interface for monitoring an AKS cluster named 'danielolaogun\_cluster'. The left sidebar shows navigation options like 'Cluster configuration', 'Networking', and 'Monitoring'. The 'Monitoring' section is expanded, showing 'Logs' as the active view. The main area shows a 'New Query 1\*' editor with a Kusto query for 'Readiness status per Node'. The query filters for nodes that are ready within the last hour. Below the query editor, a 'Results' section shows a 'Chart' view with a bar chart titled 'Completed' showing a 'ReadyCount' of 1.0 over time from 7:20 PM to 8:10 PM. The chart indicates that all nodes in the cluster are consistently ready.

```
1 // Readiness status per Node
2 // For all your cluster view count of all the nodes by readiness.
3 // To create an alert for this query, click '+ New alert rule'
4 //Customize startDateTime, endDateTime to select custom time range
5 let endDateTime = now();
6 let startDateTime = ago(1h);
7 let trendBinSize = 1m;
8 KubeNodeInventory
9 | where TimeGenerated <= endDateTime
10 | where TimeGenerated >= startDateTime
11 | distinct ClusterName, Computer, _ResourceId, TimeGenerated
```

Results: **Completed** (00:04.0, 59 records)

Chart: ReadyCount vs TimeGenerated [UTC]

Legend: danielolaogun\_cluster

# AKS Policy

---

Control what end-users can do on the cluster.

---

Ensure that clusters are in compliance (governance and legal requirements)

---

Track Misconfiguration & Security issue.





Azure Kubernetes Service (AKS)



Report Compliance

Get Policies



Azure Policy add-on

Compliance State



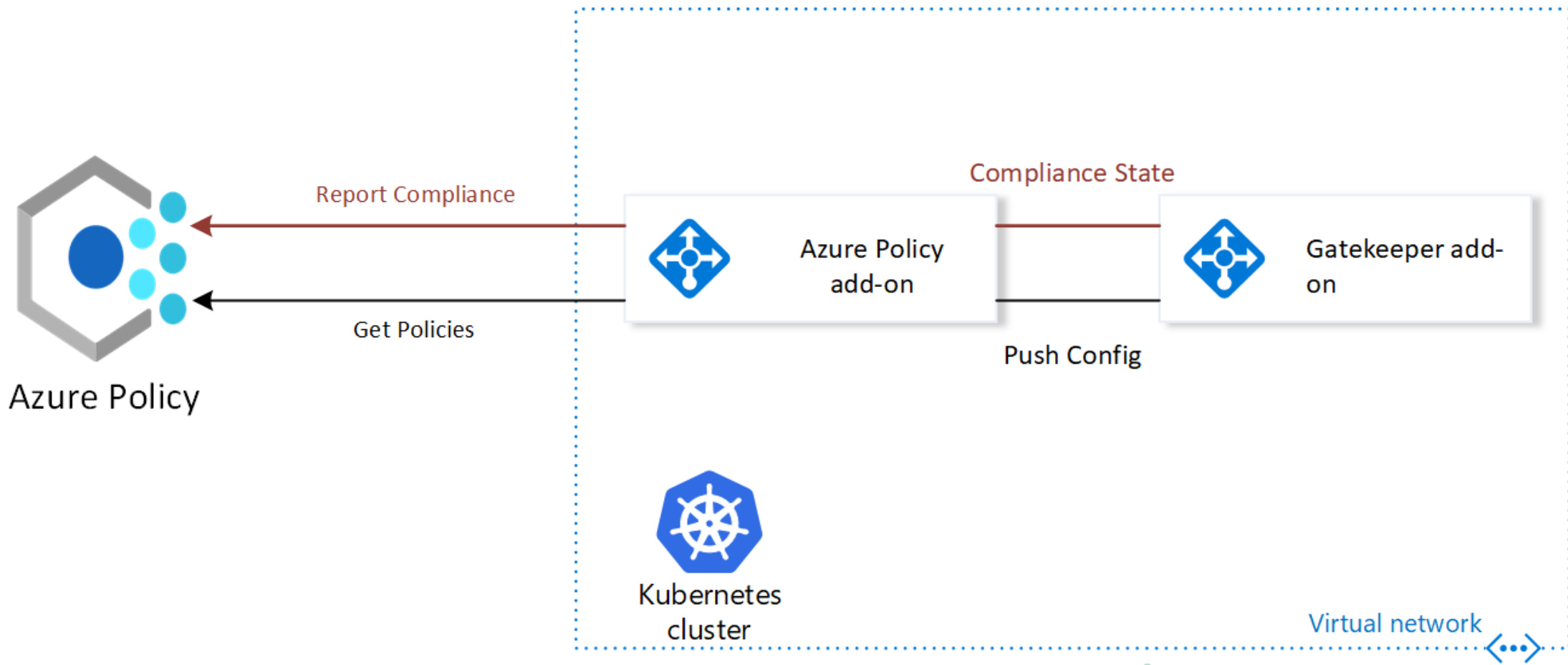
Gatekeeper add-on

Push Config



Kubernetes cluster

Virtual network



# Azure Policy built-in definitions

- Use images from trusted registries.
- Run containers with a read-only root file system.
- Enforce container CPU and memory resource limits to prevent resource attacks in a Kubernetes cluster.
- Ensure that the required annotations are attached.
- Restrict access to the Kubernetes Service Management API by granting API.



# Policy | Definitions



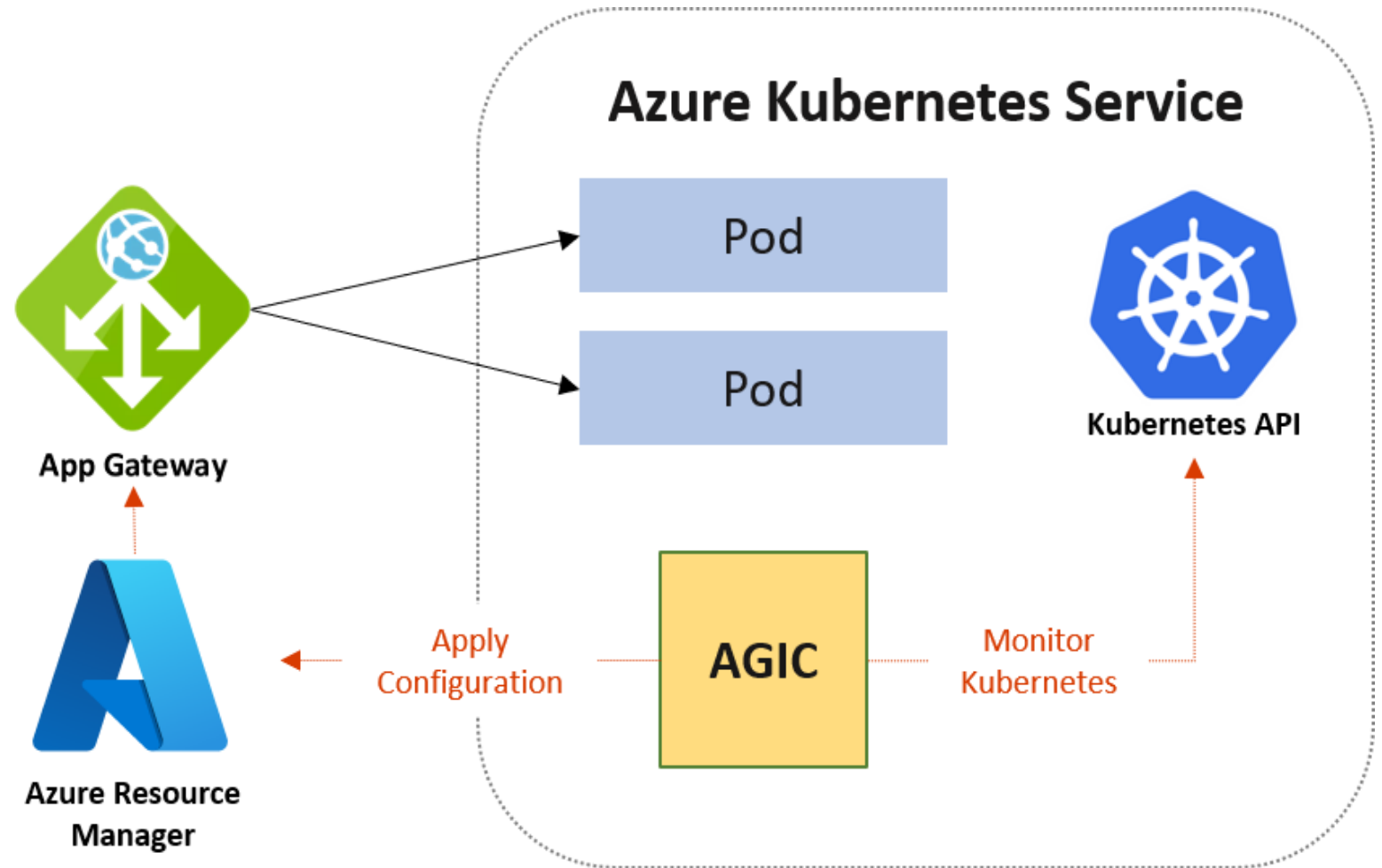
» + Initiative definition + Policy definition Export definitions Refresh

Scope: 4 selected ...
 Definition type: All definition types ▼
 Type: All types ▼
 Category: 1 categories ▼
 Search: Filter by name or ID...

Name	Definition location	Policies	Type	Definition type	Category	
Kubernetes cluster pod security restricted standards for Linux-based workloads		8	Built-in	Initiative	Kubernetes	...
Kubernetes cluster pod security baseline standards for Linux-based workloads		5	Built-in	Initiative	Kubernetes	...
[Preview]: Azure Policy Add-on for Kubernetes service (AKS) should be installed and enabled on your clusters			Built-in	Policy	Kubernetes	...
[Preview]: Deploy GitOps to Kubernetes cluster			Built-in	Policy	Kubernetes	...
Kubernetes cluster pod hostPath volumes should only use allowed host paths			Built-in	Policy	Kubernetes	...
Kubernetes cluster pods should only use allowed volume types			Built-in	Policy	Kubernetes	...
Enforce HTTPS ingress in Kubernetes cluster			Built-in	Policy	Kubernetes	...
Kubernetes clusters should not allow container privilege escalation			Built-in	Policy	Kubernetes	...
Ensure services listen only on allowed ports in Kubernetes cluster			Built-in	Policy	Kubernetes	...
Enforce internal load balancers in Kubernetes cluster			Built-in	Policy	Kubernetes	...
Ensure containers listen only on allowed ports in Kubernetes cluster			Built-in	Policy	Kubernetes	...
Enforce labels on pods in Kubernetes cluster			Built-in	Policy	Kubernetes	...
Kubernetes cluster containers should not share host process ID or host IPC namespace			Built-in	Policy	Kubernetes	...
Kubernetes cluster containers should only use allowed AppArmor profiles			Built-in	Policy	Kubernetes	...
Kubernetes cluster containers should not use forbidden sysctl interfaces			Built-in	Policy	Kubernetes	...
Kubernetes cluster pods should only use approved host network and port range			Built-in	Policy	Kubernetes	...
Do not allow privileged containers in Kubernetes cluster			Built-in	Policy	Kubernetes	...
Kubernetes cluster containers should only use allowed seccomp profiles			Built-in	Policy	Kubernetes	...
Kubernetes cluster containers should only use allowed capabilities			Built-in	Policy	Kubernetes	...
Kubernetes cluster containers should run with a read only root file system			Built-in	Policy	Kubernetes	...
Kubernetes cluster pods and containers should only use allowed SELinux options			Built-in	Policy	Kubernetes	...
Ensure container CPU and memory resource limits do not exceed the specified limits in Kubernetes cluster			Built-in	Policy	Kubernetes	...
Kubernetes cluster pods and containers should only run with approved user and group IDs			Built-in	Policy	Kubernetes	...
Kubernetes cluster and FlexVolume volumes should not use allowed drivers			Built-in	Policy	Kubernetes	...

# Application Gateway Ingress Controller (AGIC)

- Load-balancer to expose cloud software to the Internet.
- Web Application Firewall



# AGIC features

URL routing

Cookie-based affinity (stickiness).

TLS termination.

End-to-end TLS.

Support for public, private, and hybrid web sites.

Integrated web application firewall (WAF)

# AKS Pricing

- AKS is Free.
- You only pay for cloud resources.
  - Worker nodes (not master)
  - Other Cloud resources Like load balancer, storage, etc.



+

# Thank You

■



Moti Malka

DevOps Lead

Phone: +972-77-5455028

Fax: +972-77-5455027

10 Hagavish St.

Cellcom Building, Floor 2

Netanya, Israel.

