# Deploy Container applications with AKS

Sivaiah Yakkanti

# Agenda

- Overview of AKS
- Overview of ACR
- Overview of Container
- Deployments
- Services
- Ingress
- Quick Demo

# Overview of AKS

Azure Kubernetes Service (AKS) is a managed Kubernetes service that you can use to deploy and manage containerized applications

AKS reduces the complexity and operational overhead of managing Kubernetes by shifting that responsibility to Azure

It can run in on-premises, public cloud and hybrid infra architecture

Benefits

Scalability – auto scale nodes, pods
High Availability – distribute app in multiple nodes
Resource Isolation – resource limits, quotas and namespaces
Load Balancing – traffic distribute evenly across nodes
Management etc…

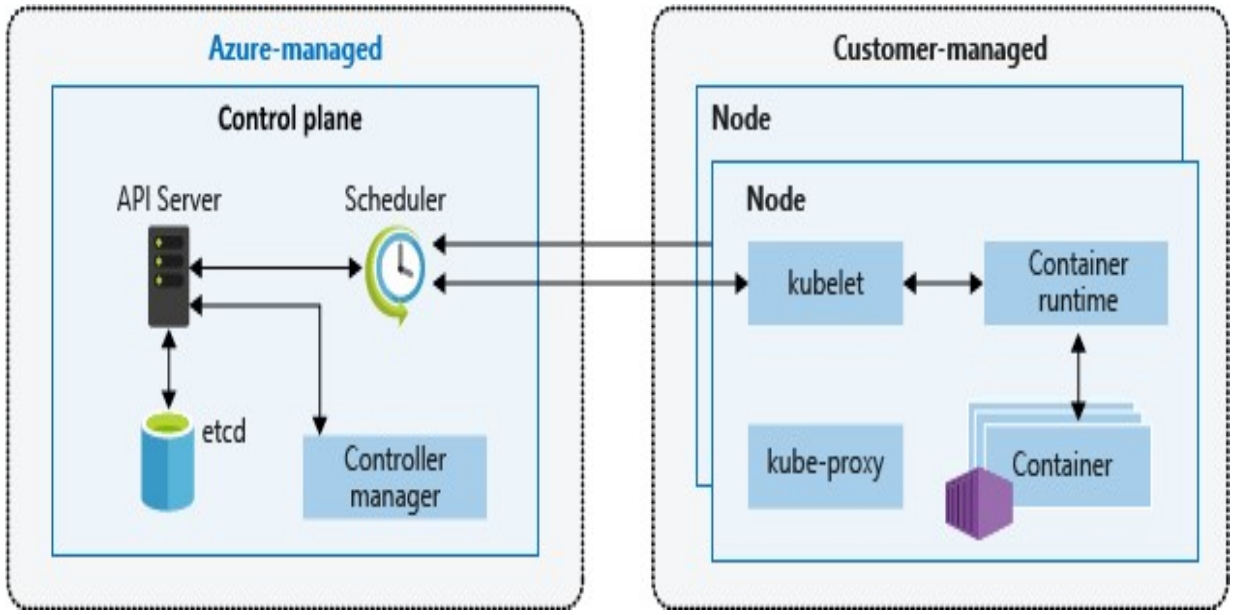
# Azure Kubernetes Services Architecture

Control Plane (Master Node ) Components

- API Server
- Scheduler
- Controller Manager
- Etcd

Date Plane (Worker Node ) Components

- Kubelet
- Kube-Proxy
- Container runtime
  - such as Docker, containerd, or CRI-O.



# Overview of Azure Container Registry

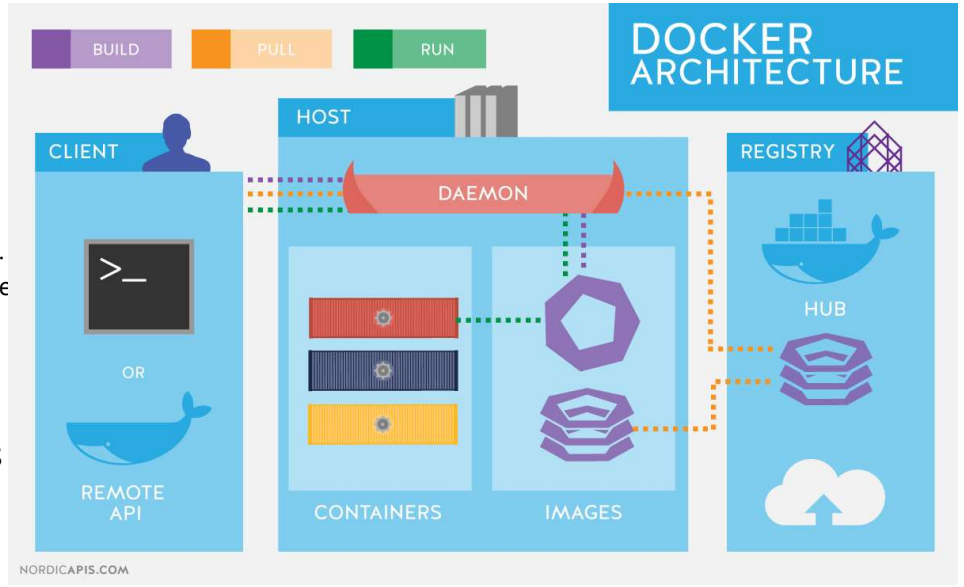
ACR is a storage and distribution service for Docker images.

It allows users to store, share, and retrieve Docker images.

Docker Hub is the official public registry maintained by Docker, but users can also set up private registries "Azure Container Registry" for their organizations

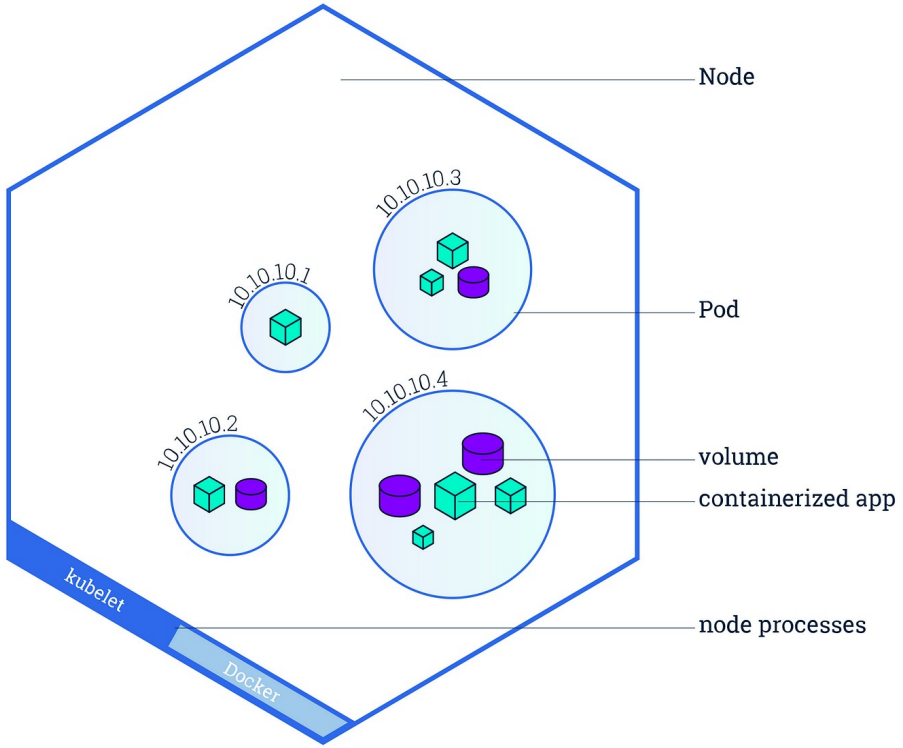
# Container

- A Dockerfile is a text file that contains instructions for building a Docker image.
  - FROM: Specifies the base image (in this case, Node.js version 14) to build upon.
  - WORKDIR: Sets the working directory within the container for subsequent instructions.
  - COPY: Copies package.json and package-lock.json to the working directory and installs dependencies using npm install.
  - COPY: Copies the application code into the container.
  - EXPOSE: Exposes port 3000 to allow external access to the application.
  - CMD: Specifies the command to run the application when the containe starts.
- A container is a runtime instance of a Docker image.
- It encapsulates an application and its dependencies, running in isolation from other containers and the host system.
- Containers are created from Docker images using the docker run command and can be started, stopped, and managed with Docker commands



# Deployments

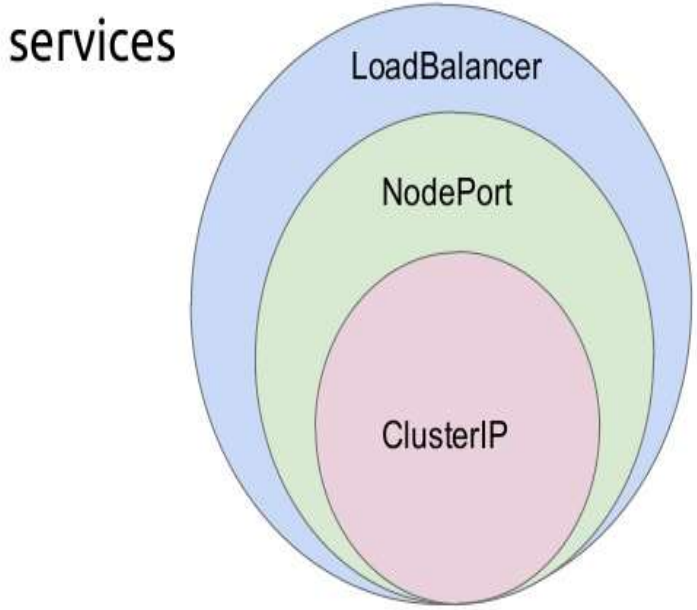
- Kubernetes provides a distributed platform for containerized applications. You build and deploy your own applications and services into a Kubernetes cluster and let the cluster manage the availability and connectivity.
  - Update a Kubernetes manifest file.
  - Deploy an application in Kubernetes.
  - Test the application.



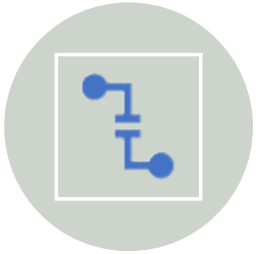
# Services

Kubernetes Services are used to logically group pods and provide network connectivity by allowing direct access to them through a specific IP address or DNS name on a designated port.

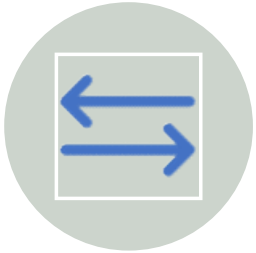
This allows you to expose your application workloads to other services within the cluster or to external clients without having to manually manage the network configuration for each pod hosting a workload.



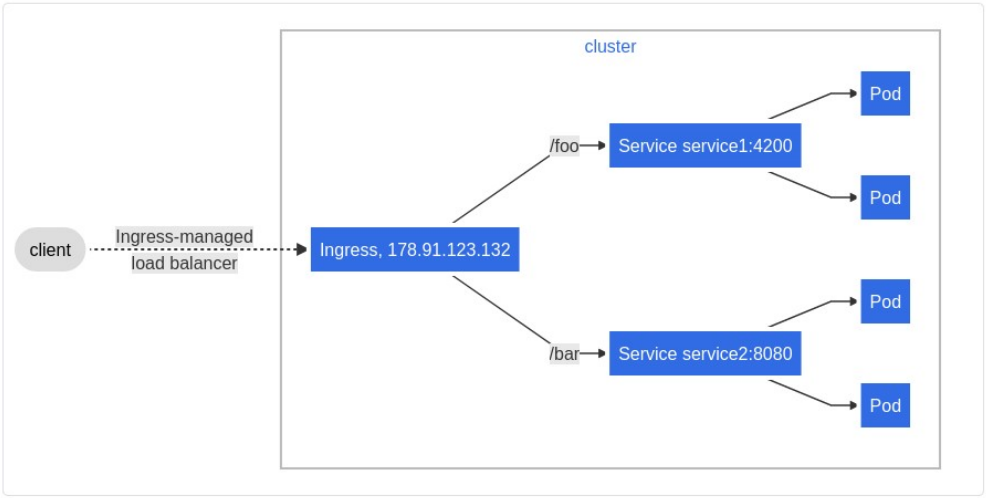
# Ingress



Ingress exposes HTTP and HTTPS routes from outside the cluster to services within the cluster. Traffic routing is controlled by rules defined on the Ingress resource.



Here is a simple example where an Ingress sends all its traffic to one Service:



# Quick Demo

# Q & A

# Thank you

Sivaiah
Yakkanti