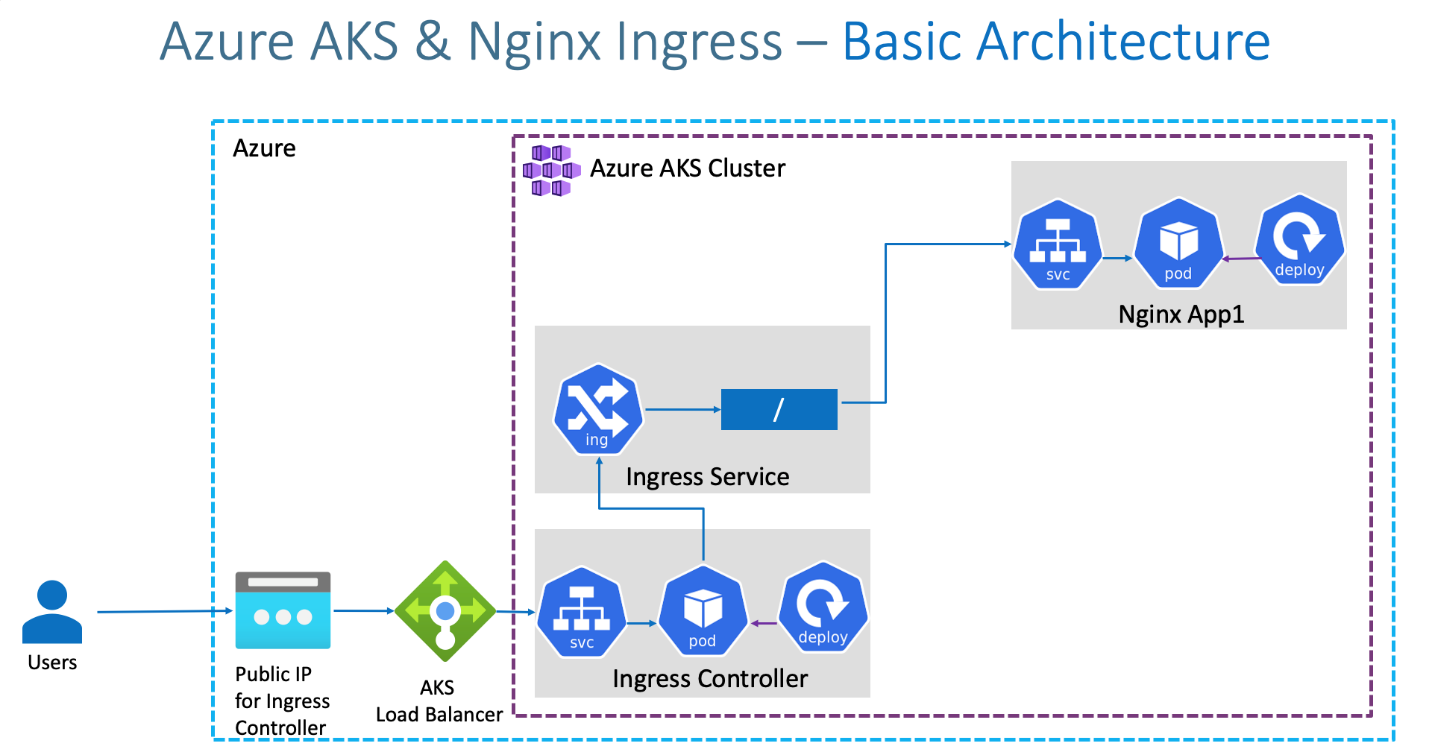
**Ingress - Basics**



### What are we going to learn?

* We are going to create a **Static Public IP** for Ingress in Azure AKS
* Associate that Public IP to **Ingress Controller** during installation.
* We are going to create a namespace ingress-basic for Ingress Controller where all ingress controller related things will be placed.
* In future, we install **cert-manager** for SSL certificates also in same namespace.
* **Caution Note:** This namespace is for Ingress controller stuff, ingress resource we can create in any other namespaces and not an issue. Only condition is create ingress resource and ingress pointed application in same namespace (Example: App1 and Ingress resource of App1 should be in same namespace)
* Create / Review Ingress Manifest
* Deploy a simple Nginx App1 with Ingress manifest and test it
* Clean-Up or delete application after testing

## Step-02: Create Static Public IP

# Get the resource group name of the AKS cluster

az aks show --resource-group aks-rg1 --name aksdemo1 --query nodeResourceGroup -o tsv

# TEMPLATE - Create a public IP address with the static allocation

az network public-ip create --resource-group <REPLACE-OUTPUT-RG-FROM-PREVIOUS-COMMAND> --name myAKSPublicIPForIngress --sku Standard --allocation-method static --query publicIp.ipAddress -o tsv

# REPLACE - Create Public IP: Replace Resource Group value

az network public-ip create --resource-group MC\_aks-rg1\_aksdemo1\_centralus --name myAKSPublicIPForIngress --sku Standard --allocation-method static --query publicIp.ipAddress -o tsv

* Make a note of Static IP which we will use in next step when installing Ingress Controller

# Make a note of Public IP created for Ingress

52.154.156.139

## Step-03: Install Ingress Controller

# Install Helm3 (if not installed)

brew install helm

# Create a namespace for your ingress resources

kubectl create namespace ingress-basic

# Add the official stable repository

helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx

helm repo add stable https://kubernetes-charts.storage.googleapis.com/

helm repo update

# Customizing the Chart Before Installing.

helm show values ingress-nginx/ingress-nginx

# Use Helm to deploy an NGINX ingress controller

helm install ingress-nginx ingress-nginx/ingress-nginx \

--namespace ingress-basic \

--set controller.replicaCount=2 \

--set controller.nodeSelector."beta\.kubernetes\.io/os"=linux \

--set defaultBackend.nodeSelector."beta\.kubernetes\.io/os"=linux \

--set controller.service.externalTrafficPolicy=Local \

--set controller.service.loadBalancerIP="REPLACE\_STATIC\_IP"

# Replace Static IP captured in Step-02

helm install ingress-nginx ingress-nginx/ingress-nginx \

--namespace ingress-basic \

--set controller.replicaCount=2 \

--set controller.nodeSelector."beta\.kubernetes\.io/os"=linux \

--set defaultBackend.nodeSelector."beta\.kubernetes\.io/os"=linux \

--set controller.service.externalTrafficPolicy=Local \

--set controller.service.loadBalancerIP="52.154.156.139"

# List Services with labels

kubectl get service -l app.kubernetes.io/name=ingress-nginx --namespace ingress-basic

# List Pods

kubectl get pods -n ingress-basic

kubectl get all -n ingress-basic

# Access Public IP

http://<Public-IP-created-for-Ingress>

# Output should be

404 Not Found from Nginx

# Verify Load Balancer on Azure Mgmt Console

Primarily refer Settings -> Frontend IP Configuration

## Step-04: Review Application k8s manifests

* 01-NginxApp1-Deployment.yml
* 02-NginxApp1-ClusterIP-Service.yml
* 03-Ingress-Basic.yml

## Step-05: Deploy Application k8s manifests and verify

# Deploy

kubectl apply -f kube-manifests/

# List Pods

kubectl get pods

# List Services

kubectl get svc

# List Ingress

kubectl get ingress

# Access Application

http://<Public-IP-created-for-Ingress>/app1/index.html

http://<Public-IP-created-for-Ingress>

# Verify Ingress Controller Logs

kubectl get pods -n ingress-basic

kubectl logs -f <pod-name> -n ingress-basic

## Step-06: Clean-Up Apps

# Delete Apps

kubectl delete -f kube-manifests/