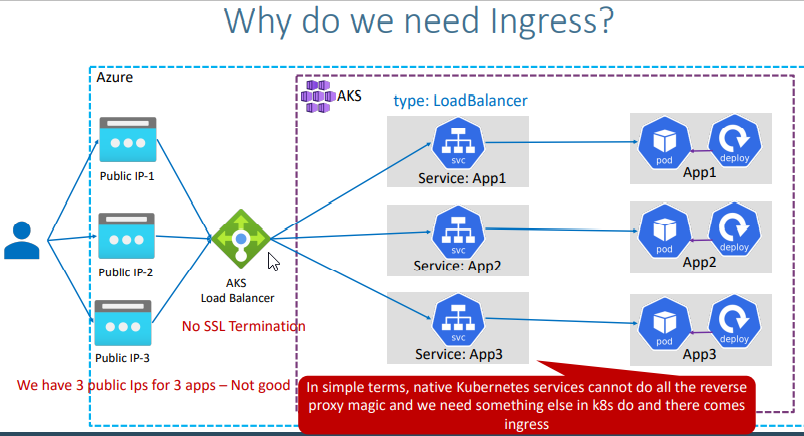
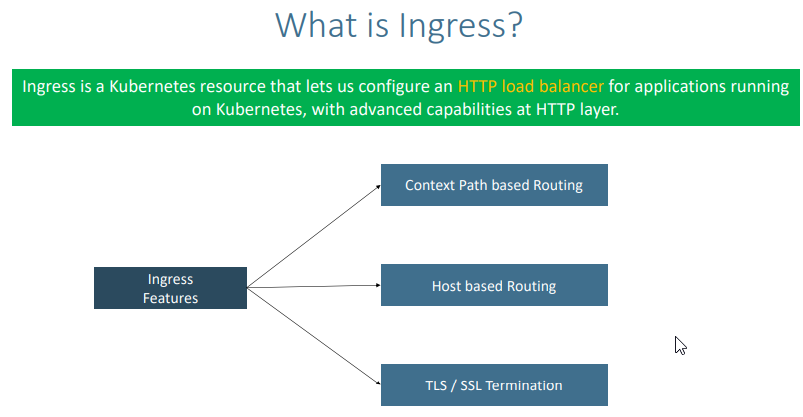
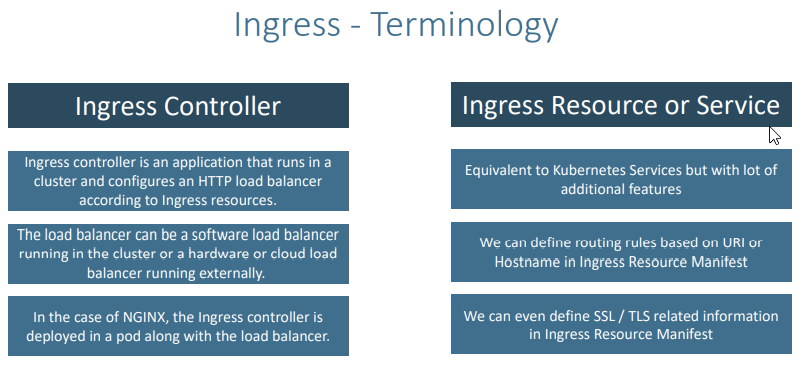
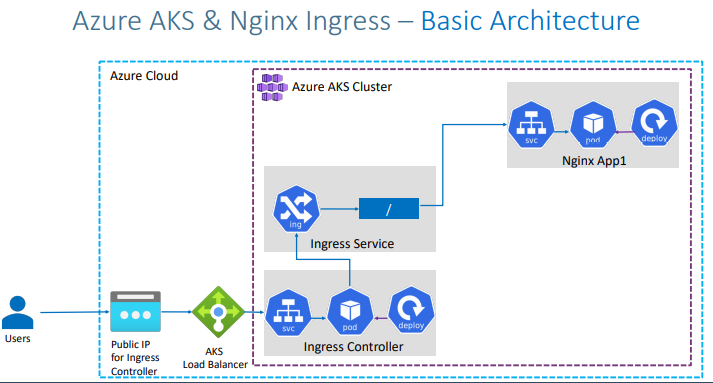
**Ingress – Basic**

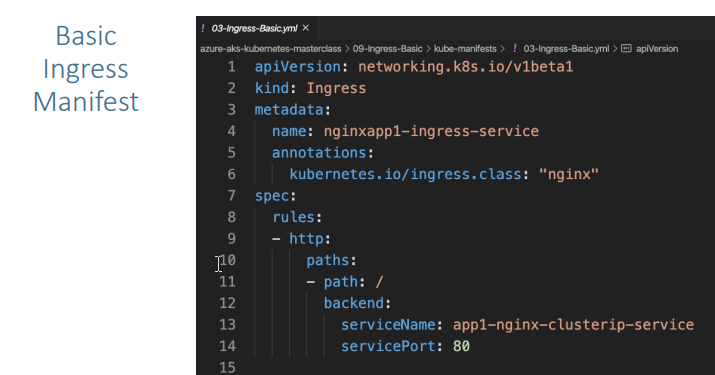






Inshot ingress controller is nothing but a controller which will be running as a pod and deployment in Kubernetes which plays a key role in handling the ingress resources ( Kubernetes manifest for your application as ingress resource it manages the Kubernetes manifest)





**What are we going to learn?**

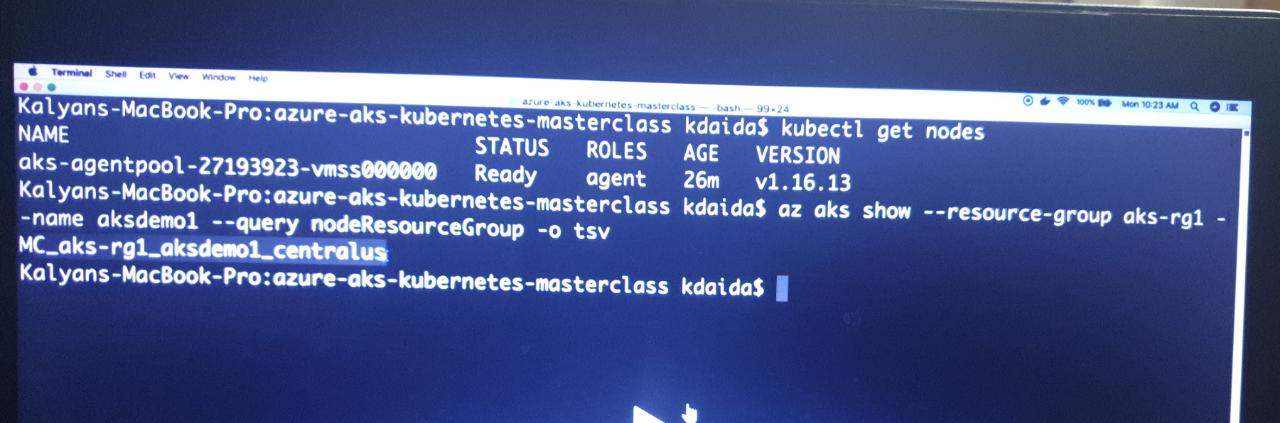
* We are going to create a **Static Public IP** for Ingress in Azure AKS
* Associate that Public IP to **Ingress Controller** during installation.
* We are going to create a namespace ingress-basic for Ingress Controller where all ingress controller related things will be placed.
* In future, we install **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

from the above query we can know the node resource group

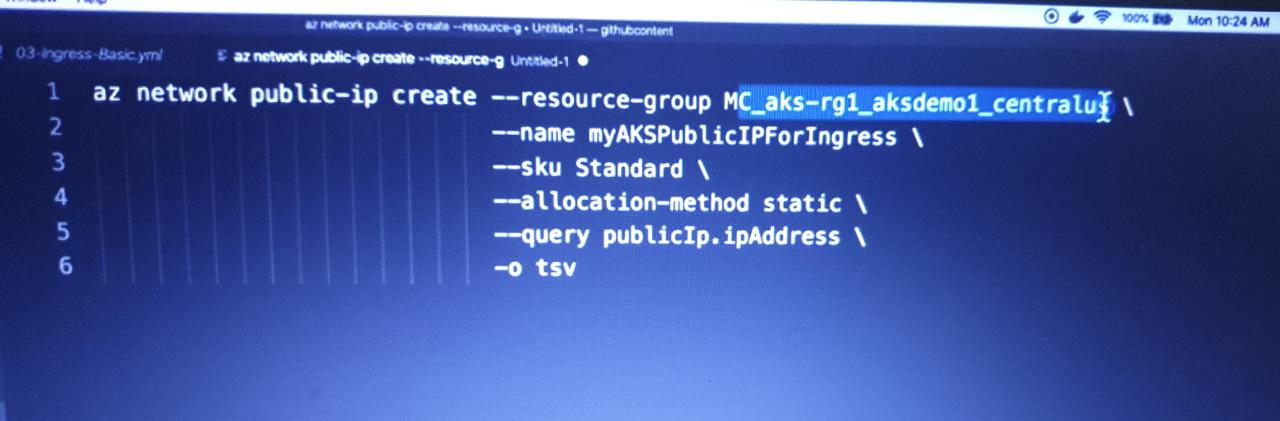


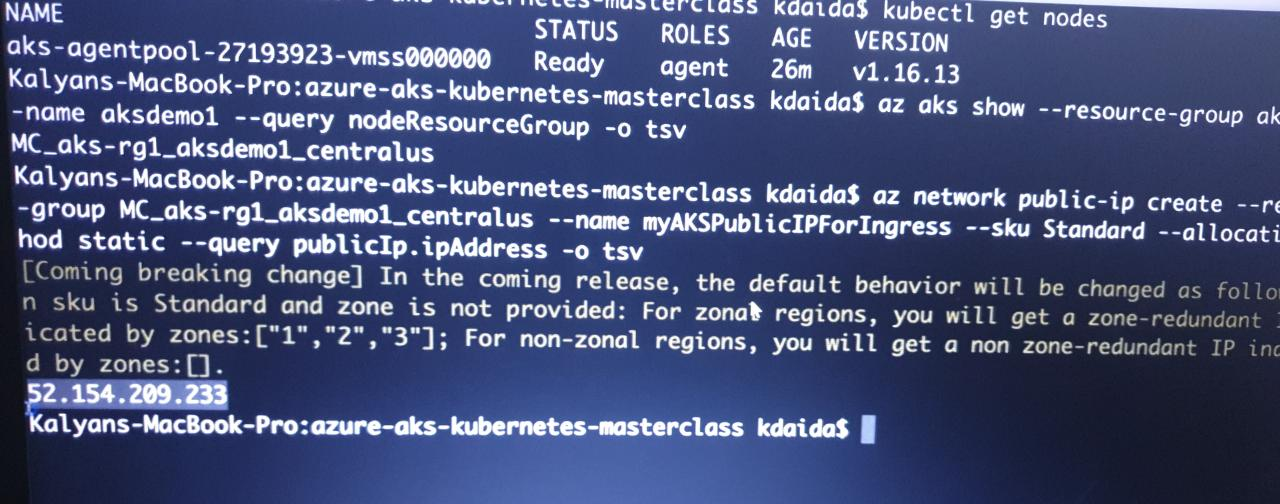
# TEMPLATE - Create a public IP address with the static allocation

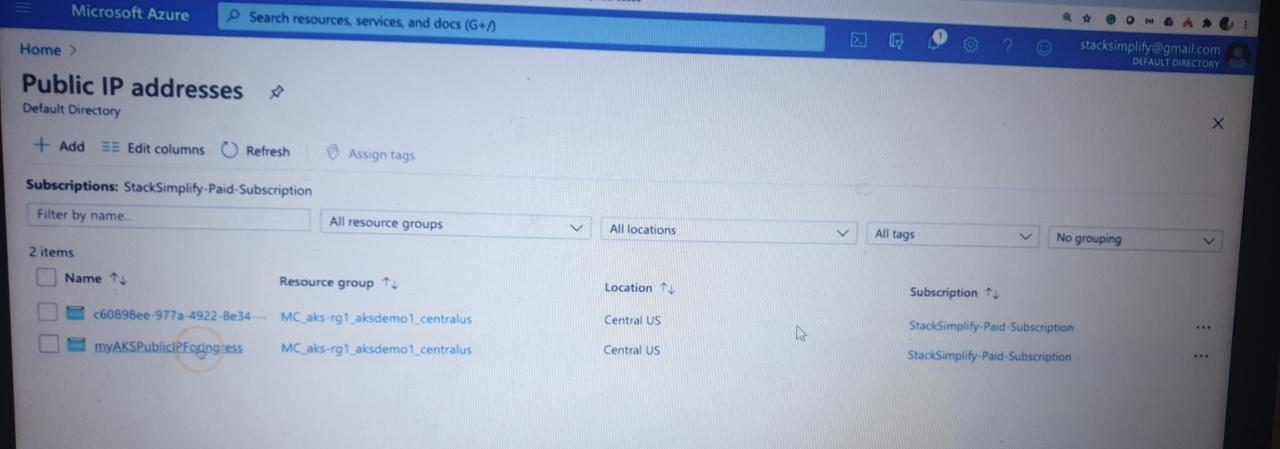
az network public-ip create --resource-group <REPLACE-OUTPUT-RG-FROM-PREVIOUS-COMMAND> --name myAKSPublicIPForIngress --sku Standard --allocation-method static --query publicIp.ipAddress -o tsv

# 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

refer the document <https://helm.sh/docs/intro/install/>

# Create a namespace for your ingress resources

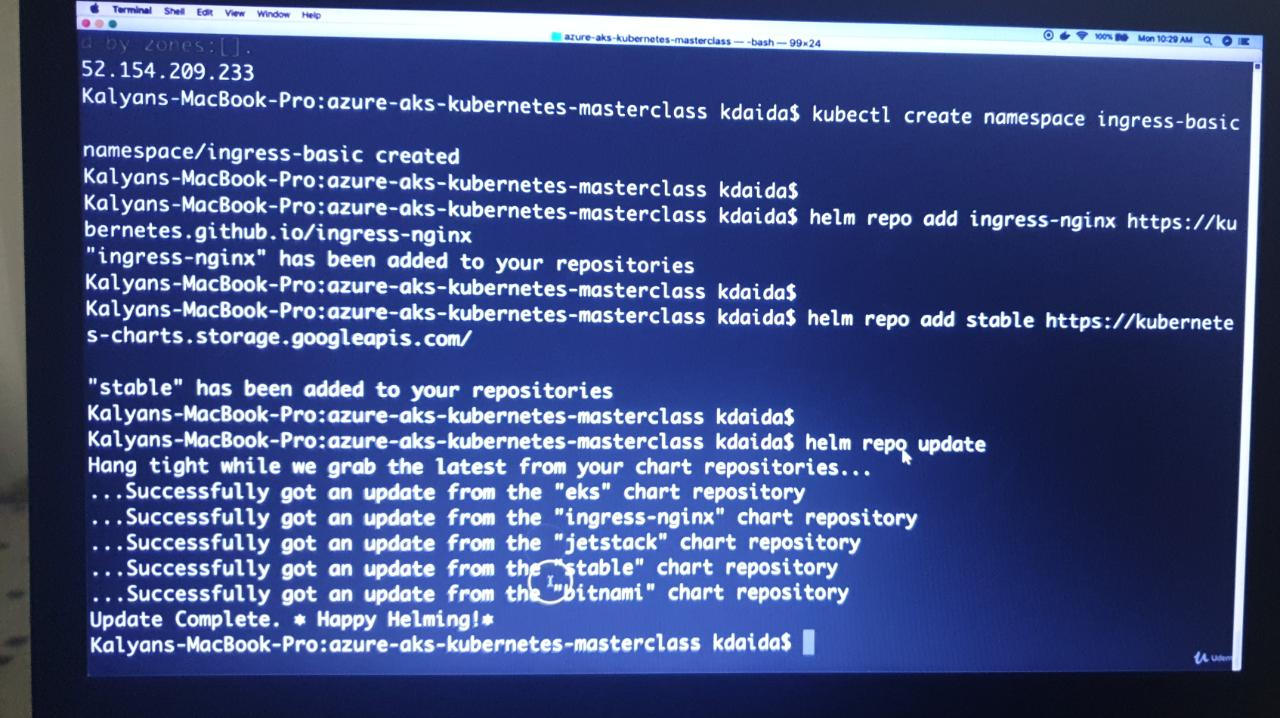
kubectl create namespace ingress-basic

# 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 (https://github.com/sravan-kumar41/azure-aks-kubernetes-masterclass/blob/master/09-Ingress-Basic/FOR-Review-purpose-values.yml)

|  |
| --- |
| # nginx configuration |
|  | ## Ref: https://github.com/kubernetes/ingress/blob/master/controllers/nginx/configuration.md |
|  | ## |
|  | controller: |
|  | image: |
|  | repository: us.gcr.io/k8s-artifacts-prod/ingress-nginx/controller |
|  | tag: "v0.34.1" |
|  | digest: sha256:0e072dddd1f7f8fc8909a2ca6f65e76c5f0d2fcfb8be47935ae3457e8bbceb20 |
|  | pullPolicy: IfNotPresent |
|  | # www-data -> uid 101 |
|  | runAsUser: 101 |
|  | allowPrivilegeEscalation: true |
|  |  |
|  | # Configures the ports the nginx-controller listens on |
|  | containerPort: |
|  | http: 80 |
|  | https: 443 |
|  |  |
|  | # Will add custom configuration options to Nginx https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/configmap/ |
|  | config: {} |
|  |  |
|  | ## Annotations to be added to the controller config configuration configmap |
|  | ## |
|  | configAnnotations: {} |
|  |  |
|  | # Will add custom headers before sending traffic to backends according to https://github.com/kubernetes/ingress-nginx/tree/master/docs/examples/customization/custom-headers |
|  | proxySetHeaders: {} |
|  |  |
|  | # Will add custom headers before sending response traffic to the client according to: https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/configmap/#add-headers |
|  | addHeaders: {} |
|  |  |
|  | # Optionally customize the pod dnsConfig. |
|  | dnsConfig: {} |
|  |  |
|  | # Optionally change this to ClusterFirstWithHostNet in case you have 'hostNetwork: true'. |
|  | # By default, while using host network, name resolution uses the host's DNS. If you wish nginx-controller |
|  | # to keep resolving names inside the k8s network, use ClusterFirstWithHostNet. |
|  | dnsPolicy: ClusterFirst |
|  |  |
|  | # Bare-metal considerations via the host network https://kubernetes.github.io/ingress-nginx/deploy/baremetal/#via-the-host-network |
|  | # Ingress status was blank because there is no Service exposing the NGINX Ingress controller in a configuration using the host network, the default --publish-service flag used in standard cloud setups does not apply |
|  | reportNodeInternalIp: false |
|  |  |
|  | # Required for use with CNI based kubernetes installations (such as ones set up by kubeadm), |
|  | # since CNI and hostport don't mix yet. Can be deprecated once https://github.com/kubernetes/kubernetes/issues/23920 |
|  | # is merged |
|  | hostNetwork: false |
|  |  |
|  | ## Use host ports 80 and 443 |
|  | ## Disabled by default |
|  | ## |
|  | hostPort: |
|  | enabled: false |
|  | ports: |
|  | http: 80 |
|  | https: 443 |
|  |  |
|  | ## Election ID to use for status update |
|  | ## |
|  | electionID: ingress-controller-leader |
|  |  |
|  | ## Name of the ingress class to route through this controller |
|  | ## |
|  | ingressClass: nginx |
|  |  |
|  | # labels to add to the pod container metadata |
|  | podLabels: {} |
|  | # key: value |
|  |  |
|  | ## Security Context policies for controller pods |
|  | ## |
|  | podSecurityContext: {} |
|  |  |
|  | ## See https://kubernetes.io/docs/tasks/administer-cluster/sysctl-cluster/ for |
|  | ## notes on enabling and using sysctls |
|  | ### |
|  | sysctls: {} |
|  | # sysctls: |
|  | # "net.core.somaxconn": "8192" |
|  |  |
|  | ## Allows customization of the source of the IP address or FQDN to report |
|  | ## in the ingress status field. By default, it reads the information provided |
|  | ## by the service. If disable, the status field reports the IP address of the |
|  | ## node or nodes where an ingress controller pod is running. |
|  | publishService: |
|  | enabled: true |
|  | ## Allows overriding of the publish service to bind to |
|  | ## Must be <namespace>/<service\_name> |
|  | ## |
|  | pathOverride: "" |
|  |  |
|  | ## Limit the scope of the controller |
|  | ## |
|  | scope: |
|  | enabled: false |
|  | namespace: "" # defaults to .Release.Namespace |
|  |  |
|  | ## Allows customization of the configmap / nginx-configmap namespace |
|  | ## |
|  | configMapNamespace: "" # defaults to .Release.Namespace |
|  |  |
|  | ## Allows customization of the tcp-services-configmap |
|  | ## |
|  | tcp: |
|  | configMapNamespace: "" # defaults to .Release.Namespace |
|  | ## Annotations to be added to the tcp config configmap |
|  | annotations: {} |
|  |  |
|  | ## Allows customization of the udp-services-configmap |
|  | ## |
|  | udp: |
|  | configMapNamespace: "" # defaults to .Release.Namespace |
|  | ## Annotations to be added to the udp config configmap |
|  | annotations: {} |
|  |  |
|  | ## Additional command line arguments to pass to nginx-ingress-controller |
|  | ## E.g. to specify the default SSL certificate you can use |
|  | ## extraArgs: |
|  | ## default-ssl-certificate: "<namespace>/<secret\_name>" |
|  | extraArgs: {} |
|  |  |
|  | ## Additional environment variables to set |
|  | extraEnvs: [] |
|  | # extraEnvs: |
|  | # - name: FOO |
|  | # valueFrom: |
|  | # secretKeyRef: |
|  | # key: FOO |
|  | # name: secret-resource |
|  |  |
|  | ## DaemonSet or Deployment |
|  | ## |
|  | kind: Deployment |
|  |  |
|  | ## Annotations to be added to the controller Deployment or DaemonSet |
|  | ## |
|  | annotations: {} |
|  |  |
|  | # The update strategy to apply to the Deployment or DaemonSet |
|  | ## |
|  | updateStrategy: {} |
|  | # rollingUpdate: |
|  | # maxUnavailable: 1 |
|  | # type: RollingUpdate |
|  |  |
|  | # minReadySeconds to avoid killing pods before we are ready |
|  | ## |
|  | minReadySeconds: 0 |
|  |  |
|  |  |
|  | ## Node tolerations for server scheduling to nodes with taints |
|  | ## Ref: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/ |
|  | ## |
|  | tolerations: [] |
|  | # - key: "key" |
|  | # operator: "Equal|Exists" |
|  | # value: "value" |
|  | # effect: "NoSchedule|PreferNoSchedule|NoExecute(1.6 only)" |
|  |  |
|  | ## Affinity and anti-affinity |
|  | ## Ref: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/#affinity-and-anti-affinity |
|  | ## |
|  | affinity: {} |
|  | # # An example of preferred pod anti-affinity, weight is in the range 1-100 |
|  | # podAntiAffinity: |
|  | # preferredDuringSchedulingIgnoredDuringExecution: |
|  | # - weight: 100 |
|  | # podAffinityTerm: |
|  | # labelSelector: |
|  | # matchExpressions: |
|  | # - key: app |
|  | # operator: In |
|  | # values: |
|  | # - ingress-nginx |
|  | # topologyKey: kubernetes.io/hostname |
|  |  |
|  | # # An example of required pod anti-affinity |
|  | # podAntiAffinity: |
|  | # requiredDuringSchedulingIgnoredDuringExecution: |
|  | # - labelSelector: |
|  | # matchExpressions: |
|  | # - key: app |
|  | # operator: In |
|  | # values: |
|  | # - ingress-nginx |
|  | # topologyKey: "kubernetes.io/hostname" |
|  |  |
|  | ## terminationGracePeriodSeconds |
|  | ## wait up to five minutes for the drain of connections |
|  | ## |
|  | terminationGracePeriodSeconds: 300 |
|  |  |
|  | ## Node labels for controller pod assignment |
|  | ## Ref: https://kubernetes.io/docs/user-guide/node-selection/ |
|  | ## |
|  | nodeSelector: {} |
|  |  |
|  | ## Liveness and readiness probe values |
|  | ## Ref: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#container-probes |
|  | ## |
|  | livenessProbe: |
|  | failureThreshold: 5 |
|  | initialDelaySeconds: 10 |
|  | periodSeconds: 10 |
|  | successThreshold: 1 |
|  | timeoutSeconds: 1 |
|  | port: 10254 |
|  | readinessProbe: |
|  | failureThreshold: 3 |
|  | initialDelaySeconds: 10 |
|  | periodSeconds: 10 |
|  | successThreshold: 1 |
|  | timeoutSeconds: 1 |
|  | port: 10254 |
|  |  |
|  | # Path of the health check endpoint. All requests received on the port defined by |
|  | # the healthz-port parameter are forwarded internally to this path. |
|  | healthCheckPath: "/healthz" |
|  |  |
|  | ## Annotations to be added to controller pods |
|  | ## |
|  | podAnnotations: {} |
|  |  |
|  | replicaCount: 1 |
|  |  |
|  | minAvailable: 1 |
|  |  |
|  | # Define requests resources to avoid probe issues due to CPU utilization in busy nodes |
|  | # ref: https://github.com/kubernetes/ingress-nginx/issues/4735#issuecomment-551204903 |
|  | # Ideally, there should be no limits. |
|  | # https://engineering.indeedblog.com/blog/2019/12/cpu-throttling-regression-fix/ |
|  | resources: |
|  | # limits: |
|  | # cpu: 100m |
|  | # memory: 90Mi |
|  | requests: |
|  | cpu: 100m |
|  | memory: 90Mi |
|  |  |
|  | autoscaling: |
|  | enabled: false |
|  | minReplicas: 1 |
|  | maxReplicas: 11 |
|  | targetCPUUtilizationPercentage: 50 |
|  | targetMemoryUtilizationPercentage: 50 |
|  |  |
|  | autoscalingTemplate: [] |
|  | # Custom or additional autoscaling metrics |
|  | # ref: https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/#support-for-custom-metrics |
|  | # - type: Pods |
|  | # pods: |
|  | # metric: |
|  | # name: nginx\_ingress\_controller\_nginx\_process\_requests\_total |
|  | # target: |
|  | # type: AverageValue |
|  | # averageValue: 10000m |
|  |  |
|  | ## Enable mimalloc as a drop-in replacement for malloc. |
|  | ## ref: https://github.com/microsoft/mimalloc |
|  | ## |
|  | enableMimalloc: false |
|  |  |
|  | ## Override NGINX template |
|  | customTemplate: |
|  | configMapName: "" |
|  | configMapKey: "" |
|  |  |
|  | service: |
|  | enabled: true |
|  |  |
|  | annotations: {} |
|  | labels: {} |
|  | # clusterIP: "" |
|  |  |
|  | ## List of IP addresses at which the controller services are available |
|  | ## Ref: https://kubernetes.io/docs/user-guide/services/#external-ips |
|  | ## |
|  | externalIPs: [] |
|  |  |
|  | # loadBalancerIP: "" |
|  | loadBalancerSourceRanges: [] |
|  |  |
|  | enableHttp: true |
|  | enableHttps: true |
|  |  |
|  | ## Set external traffic policy to: "Local" to preserve source IP on |
|  | ## providers supporting it |
|  | ## Ref: https://kubernetes.io/docs/tutorials/services/source-ip/#source-ip-for-services-with-typeloadbalancer |
|  | # externalTrafficPolicy: "" |
|  |  |
|  | # Must be either "None" or "ClientIP" if set. Kubernetes will default to "None". |
|  | # Ref: https://kubernetes.io/docs/concepts/services-networking/service/#virtual-ips-and-service-proxies |
|  | # sessionAffinity: "" |
|  |  |
|  | # specifies the health check node port (numeric port number) for the service. If healthCheckNodePort isn’t specified, |
|  | # the service controller allocates a port from your cluster’s NodePort range. |
|  | # Ref: https://kubernetes.io/docs/tasks/access-application-cluster/create-external-load-balancer/#preserving-the-client-source-ip |
|  | # healthCheckNodePort: 0 |
|  |  |
|  | ports: |
|  | http: 80 |
|  | https: 443 |
|  |  |
|  | targetPorts: |
|  | http: http |
|  | https: https |
|  |  |
|  | type: LoadBalancer |
|  |  |
|  | # type: NodePort |
|  | # nodePorts: |
|  | # http: 32080 |
|  | # https: 32443 |
|  | # tcp: |
|  | # 8080: 32808 |
|  | nodePorts: |
|  | http: "" |
|  | https: "" |
|  | tcp: {} |
|  | udp: {} |
|  |  |
|  | ## Enables an additional internal load balancer (besides the external one). |
|  | ## Annotations are mandatory for the load balancer to come up. Varies with the cloud service. |
|  | internal: |
|  | enabled: false |
|  | annotations: {} |
|  |  |
|  | extraContainers: [] |
|  | ## Additional containers to be added to the controller pod. |
|  | ## See https://github.com/lemonldap-ng-controller/lemonldap-ng-controller as example. |
|  | # - name: my-sidecar |
|  | # image: nginx:latest |
|  | # - name: lemonldap-ng-controller |
|  | # image: lemonldapng/lemonldap-ng-controller:0.2.0 |
|  | # args: |
|  | # - /lemonldap-ng-controller |
|  | # - --alsologtostderr |
|  | # - --configmap=$(POD\_NAMESPACE)/lemonldap-ng-configuration |
|  | # env: |
|  | # - name: POD\_NAME |
|  | # valueFrom: |
|  | # fieldRef: |
|  | # fieldPath: metadata.name |
|  | # - name: POD\_NAMESPACE |
|  | # valueFrom: |
|  | # fieldRef: |
|  | # fieldPath: metadata.namespace |
|  | # volumeMounts: |
|  | # - name: copy-portal-skins |
|  | # mountPath: /srv/var/lib/lemonldap-ng/portal/skins |
|  |  |
|  | extraVolumeMounts: [] |
|  | ## Additional volumeMounts to the controller main container. |
|  | # - name: copy-portal-skins |
|  | # mountPath: /var/lib/lemonldap-ng/portal/skins |
|  |  |
|  | extraVolumes: [] |
|  | ## Additional volumes to the controller pod. |
|  | # - name: copy-portal-skins |
|  | # emptyDir: {} |
|  |  |
|  | extraInitContainers: [] |
|  | ## Containers, which are run before the app containers are started. |
|  | # - name: init-myservice |
|  | # image: busybox |
|  | # command: ['sh', '-c', 'until nslookup myservice; do echo waiting for myservice; sleep 2; done;'] |
|  |  |
|  | admissionWebhooks: |
|  | enabled: true |
|  | failurePolicy: Fail |
|  | port: 8443 |
|  |  |
|  | service: |
|  | annotations: {} |
|  | # clusterIP: "" |
|  | externalIPs: [] |
|  | # loadBalancerIP: "" |
|  | loadBalancerSourceRanges: [] |
|  | servicePort: 443 |
|  | type: ClusterIP |
|  |  |
|  | patch: |
|  | enabled: true |
|  | image: |
|  | repository: docker.io/jettech/kube-webhook-certgen |
|  | tag: v1.2.2 |
|  | pullPolicy: IfNotPresent |
|  | ## Provide a priority class name to the webhook patching job |
|  | ## |
|  | priorityClassName: "" |
|  | podAnnotations: {} |
|  | nodeSelector: {} |
|  | tolerations: [] |
|  | runAsUser: 2000 |
|  |  |
|  | metrics: |
|  | port: 10254 |
|  | # if this port is changed, change healthz-port: in extraArgs: accordingly |
|  | enabled: false |
|  |  |
|  | service: |
|  | annotations: {} |
|  | # prometheus.io/scrape: "true" |
|  | # prometheus.io/port: "10254" |
|  |  |
|  | # clusterIP: "" |
|  |  |
|  | ## List of IP addresses at which the stats-exporter service is available |
|  | ## Ref: https://kubernetes.io/docs/user-guide/services/#external-ips |
|  | ## |
|  | externalIPs: [] |
|  |  |
|  | # loadBalancerIP: "" |
|  | loadBalancerSourceRanges: [] |
|  | servicePort: 9913 |
|  | type: ClusterIP |
|  |  |
|  | serviceMonitor: |
|  | enabled: false |
|  | additionalLabels: {} |
|  | namespace: "" |
|  | namespaceSelector: {} |
|  | # Default: scrape .Release.Namespace only |
|  | # To scrape all, use the following: |
|  | # namespaceSelector: |
|  | # any: true |
|  | scrapeInterval: 30s |
|  | # honorLabels: true |
|  |  |
|  | prometheusRule: |
|  | enabled: false |
|  | additionalLabels: {} |
|  | # namespace: "" |
|  | rules: [] |
|  | # # These are just examples rules, please adapt them to your needs |
|  | # - alert: TooMany500s |
|  | # expr: 100 \* ( sum( nginx\_ingress\_controller\_requests{status=~"5.+"} ) / sum(nginx\_ingress\_controller\_requests) ) > 5 |
|  | # for: 1m |
|  | # labels: |
|  | # severity: critical |
|  | # annotations: |
|  | # description: Too many 5XXs |
|  | # summary: More than 5% of the all requests did return 5XX, this require your attention |
|  | # - alert: TooMany400s |
|  | # expr: 100 \* ( sum( nginx\_ingress\_controller\_requests{status=~"4.+"} ) / sum(nginx\_ingress\_controller\_requests) ) > 5 |
|  | # for: 1m |
|  | # labels: |
|  | # severity: critical |
|  | # annotations: |
|  | # description: Too many 4XXs |
|  | # summary: More than 5% of the all requests did return 4XX, this require your attention |
|  |  |
|  | ## Improve connection draining when ingress controller pod is deleted using a lifecycle hook: |
|  | ## With this new hook, we increased the default terminationGracePeriodSeconds from 30 seconds |
|  | ## to 300, allowing the draining of connections up to five minutes. |
|  | ## If the active connections end before that, the pod will terminate gracefully at that time. |
|  | ## To efectively take advantage of this feature, the Configmap feature |
|  | ## worker-shutdown-timeout new value is 240s instead of 10s. |
|  | ## |
|  | lifecycle: |
|  | preStop: |
|  | exec: |
|  | command: |
|  | - /wait-shutdown |
|  |  |
|  | priorityClassName: "" |
|  |  |
|  | ## Rollback limit |
|  | ## |
|  | revisionHistoryLimit: 10 |
|  |  |
|  | # Maxmind license key to download GeoLite2 Databases |
|  | # https://blog.maxmind.com/2019/12/18/significant-changes-to-accessing-and-using-geolite2-databases |
|  | maxmindLicenseKey: "" |
|  |  |
|  | ## Default 404 backend |
|  | ## |
|  | defaultBackend: |
|  | ## |
|  | enabled: false |
|  |  |
|  | image: |
|  | repository: k8s.gcr.io/defaultbackend-amd64 |
|  | tag: "1.5" |
|  | pullPolicy: IfNotPresent |
|  | # nobody user -> uid 65534 |
|  | runAsUser: 65534 |
|  |  |
|  | extraArgs: {} |
|  |  |
|  | serviceAccount: |
|  | create: true |
|  | name: |
|  | ## Additional environment variables to set for defaultBackend pods |
|  | extraEnvs: [] |
|  |  |
|  | port: 8080 |
|  |  |
|  | ## Readiness and liveness probes for default backend |
|  | ## Ref: https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-probes/ |
|  | ## |
|  | livenessProbe: |
|  | failureThreshold: 3 |
|  | initialDelaySeconds: 30 |
|  | periodSeconds: 10 |
|  | successThreshold: 1 |
|  | timeoutSeconds: 5 |
|  | readinessProbe: |
|  | failureThreshold: 6 |
|  | initialDelaySeconds: 0 |
|  | periodSeconds: 5 |
|  | successThreshold: 1 |
|  | timeoutSeconds: 5 |
|  |  |
|  | ## Node tolerations for server scheduling to nodes with taints |
|  | ## Ref: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/ |
|  | ## |
|  | tolerations: [] |
|  | # - key: "key" |
|  | # operator: "Equal|Exists" |
|  | # value: "value" |
|  | # effect: "NoSchedule|PreferNoSchedule|NoExecute(1.6 only)" |
|  |  |
|  | affinity: {} |
|  |  |
|  | ## Security Context policies for controller pods |
|  | ## See https://kubernetes.io/docs/tasks/administer-cluster/sysctl-cluster/ for |
|  | ## notes on enabling and using sysctls |
|  | ## |
|  | podSecurityContext: {} |
|  |  |
|  | # labels to add to the pod container metadata |
|  | podLabels: {} |
|  | # key: value |
|  |  |
|  | ## Node labels for default backend pod assignment |
|  | ## Ref: https://kubernetes.io/docs/user-guide/node-selection/ |
|  | ## |
|  | nodeSelector: {} |
|  |  |
|  | ## Annotations to be added to default backend pods |
|  | ## |
|  | podAnnotations: {} |
|  |  |
|  | replicaCount: 1 |
|  |  |
|  | minAvailable: 1 |
|  |  |
|  | resources: {} |
|  | # limits: |
|  | # cpu: 10m |
|  | # memory: 20Mi |
|  | # requests: |
|  | # cpu: 10m |
|  | # memory: 20Mi |
|  |  |
|  | service: |
|  | annotations: {} |
|  |  |
|  | # clusterIP: "" |
|  |  |
|  | ## List of IP addresses at which the default backend service is available |
|  | ## Ref: https://kubernetes.io/docs/user-guide/services/#external-ips |
|  | ## |
|  | externalIPs: [] |
|  |  |
|  | # loadBalancerIP: "" |
|  | loadBalancerSourceRanges: [] |
|  | servicePort: 80 |
|  | type: ClusterIP |
|  |  |
|  | priorityClassName: "" |
|  |  |
|  | ## Enable RBAC as per https://github.com/kubernetes/ingress/tree/master/examples/rbac/nginx and https://github.com/kubernetes/ingress/issues/266 |
|  | rbac: |
|  | create: true |
|  | scope: false |
|  |  |
|  | # If true, create & use Pod Security Policy resources |
|  | # https://kubernetes.io/docs/concepts/policy/pod-security-policy/ |
|  | podSecurityPolicy: |
|  | enabled: false |
|  |  |
|  | serviceAccount: |
|  | create: true |
|  | name: |
|  |  |
|  | ## Optional array of imagePullSecrets containing private registry credentials |
|  | ## Ref: https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/ |
|  | imagePullSecrets: [] |
|  | # - name: secretName |
|  |  |
|  | # TCP service key:value pairs |
|  | # Ref: https://github.com/kubernetes/contrib/tree/master/ingress/controllers/nginx/examples/tcp |
|  | ## |
|  | tcp: {} |
|  | # 8080: "default/example-tcp-svc:9000" |
|  |  |
|  | # UDP service key:value pairs |
|  | # Ref: https://github.com/kubernetes/contrib/tree/master/ingress/controllers/nginx/examples/udp |
|  | ## |
|  | udp: {} |
|  | # 53: "kube-system/kube-dns:53" |

# 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="52.154.209.233"

# 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

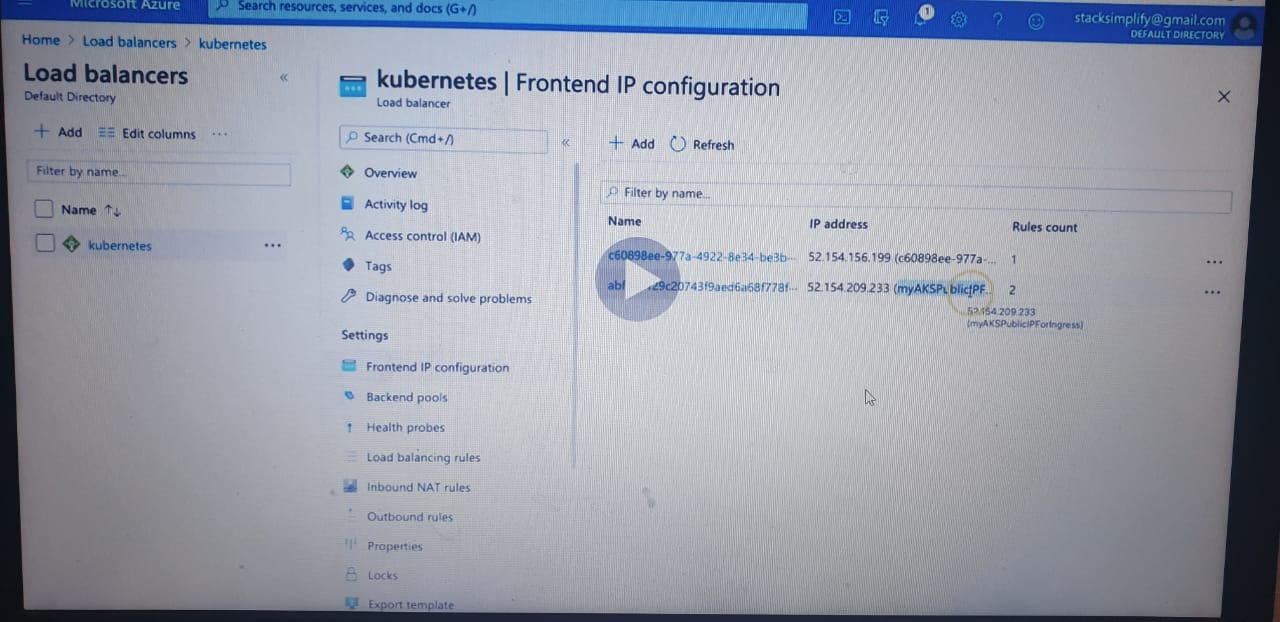
http://<Public-IP-created-for-Ingress>

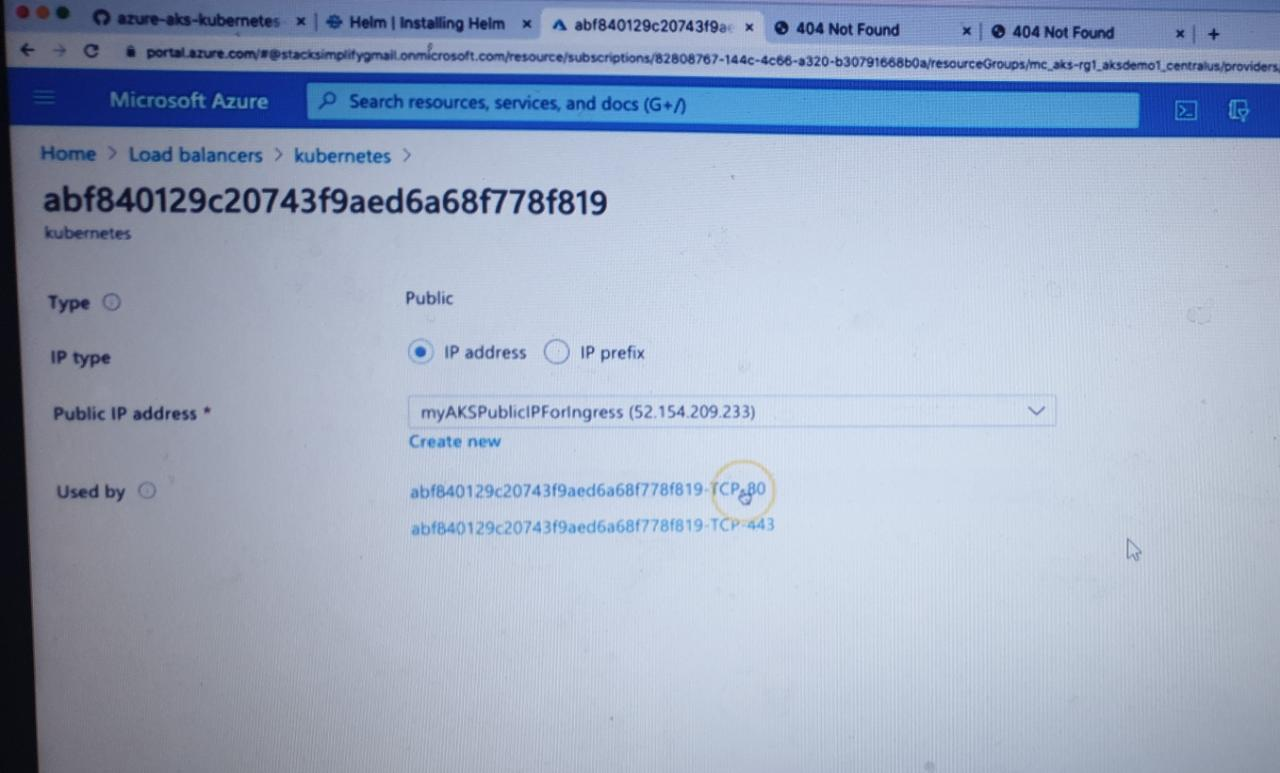
# Output should be

404 Not Found from Nginx

# Verify Load Balancer on Azure Mgmt Console

Primarily refer Settings -> Frontend IP Configuration





**Step-04: Review Application k8s manifests**

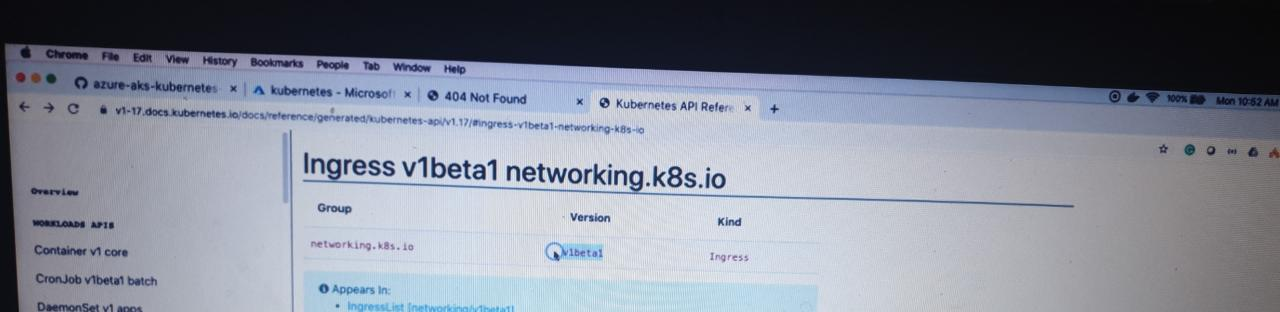
* 01-NginxApp1-Deployment.yml

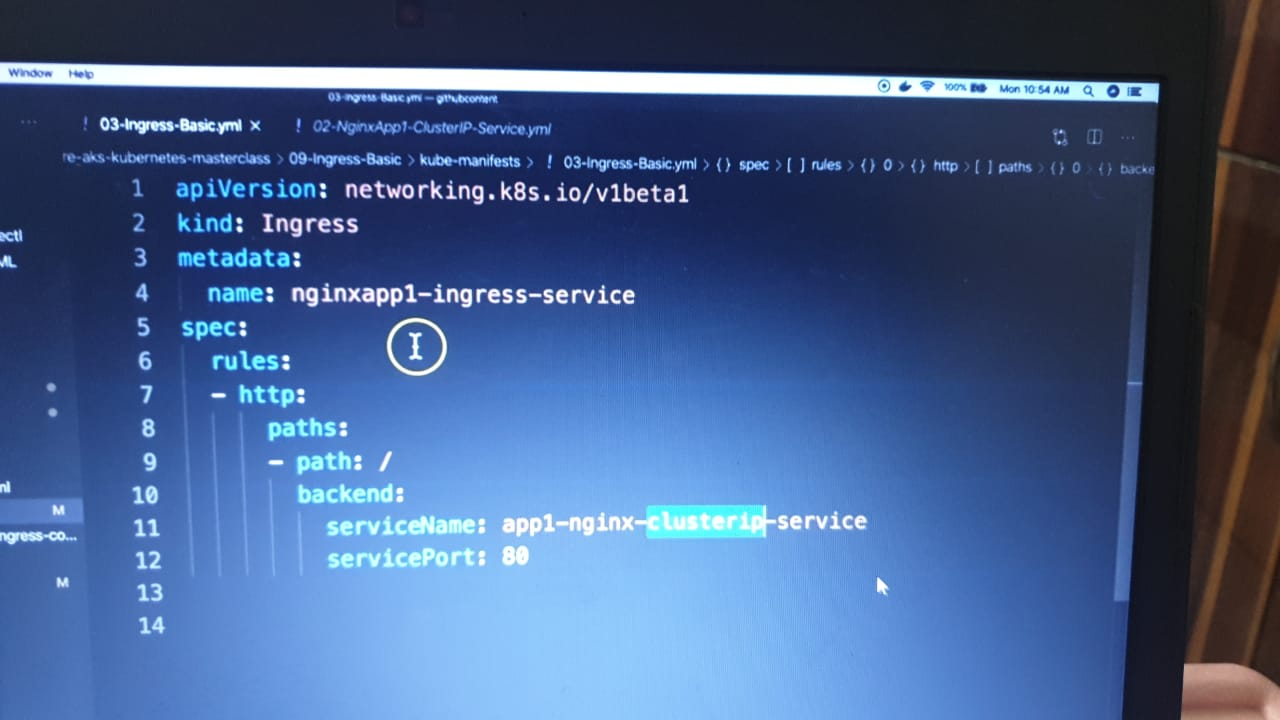


* 02-NginxApp1-ClusterIP-Service.yml



* 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/

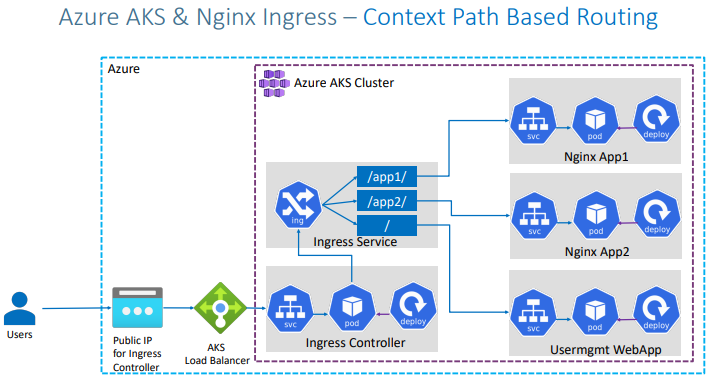
**Ingress Annotation Reference**

* <https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/annotations/>

**Other References**

* <https://github.com/kubernetes/ingress-nginx>
* <https://github.com/kubernetes/ingress-nginx/blob/master/charts/ingress-nginx/values.yaml>
* <https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.34.1/deploy/static/provider/cloud/deploy.yaml>
* <https://kubernetes.github.io/ingress-nginx/deploy/#azure>
* <https://helm.sh/docs/intro/install/>
* <https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.19/#ingress-v1-networking-k8s-io>

Ingress - Context Path based Routing





<https://github.com/sravan-kumar41/azure-aks-kubernetes-masterclass/tree/master/10-Ingress-Context-Path-Based-Routing> copy all the manifests

Review k8s Application Manifests

01-NginxApp1-Manifests

02-NginxApp2-Manifests

03-UserMgmtmWebApp-Manifests

Step-03: Review Ingress Service Manifests

04-IngressService-Manifests

Step-04: Deploy and Verify

# Deploy Apps

kubectl apply -R -f kube-manifests/ ( R is for recursive inside the folder > inside the folder) if we need to execute then mention R

# List Pods

kubectl get pods

# List Services

kubectl get svc

# List Ingress

kubectl get ingress

# Verify Ingress Controller Logs

kubectl get pods -n ingress-basic

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

Step-05: Access Applications

# Access App1

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

# Access App2

http://<Public-IP-created-for-Ingress>/app2/index.html

# Access Usermgmt Web App

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

Username: admin101

Password: password101

Step-06: Clean-Up Applications

# Delete Apps

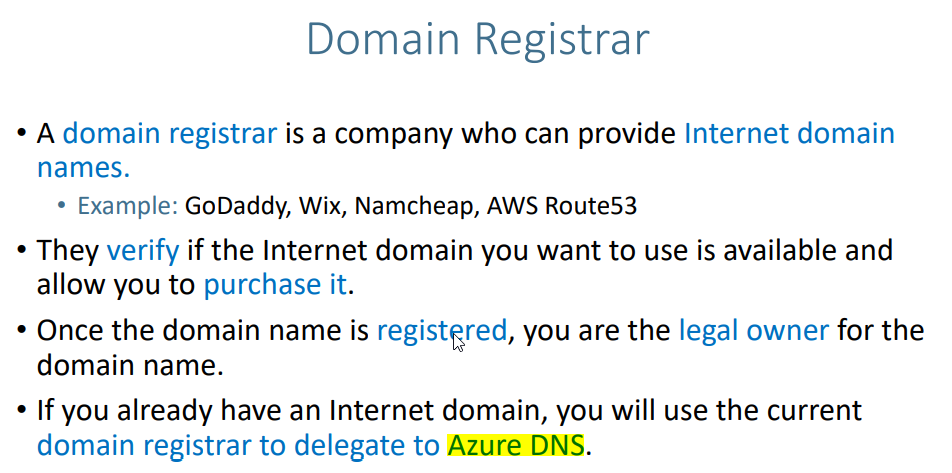
kubectl delete -f kube-manifests/

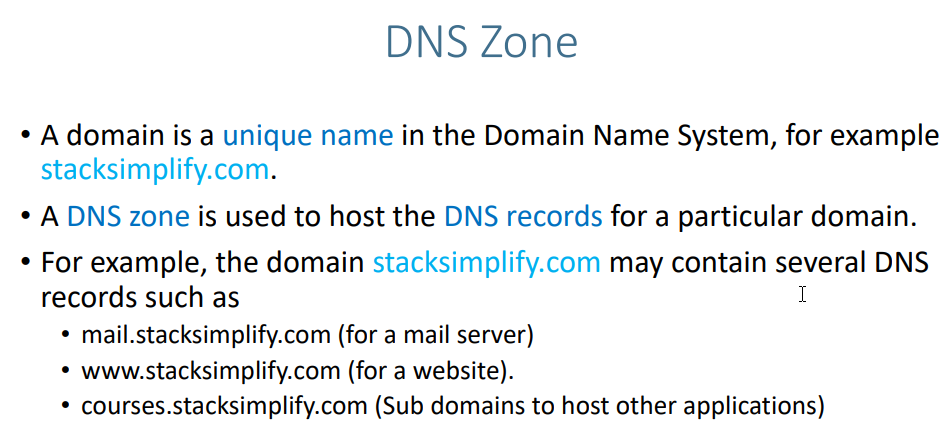
# Delete Azure Disk created for Usermgmt Web App

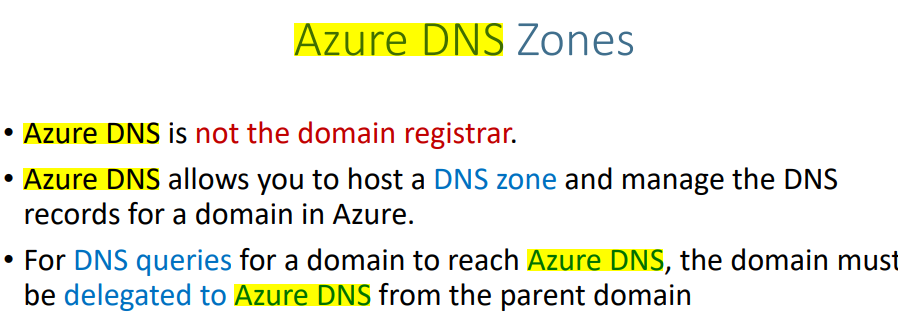
Go to All Services -> Azure Disks -> Delete disk

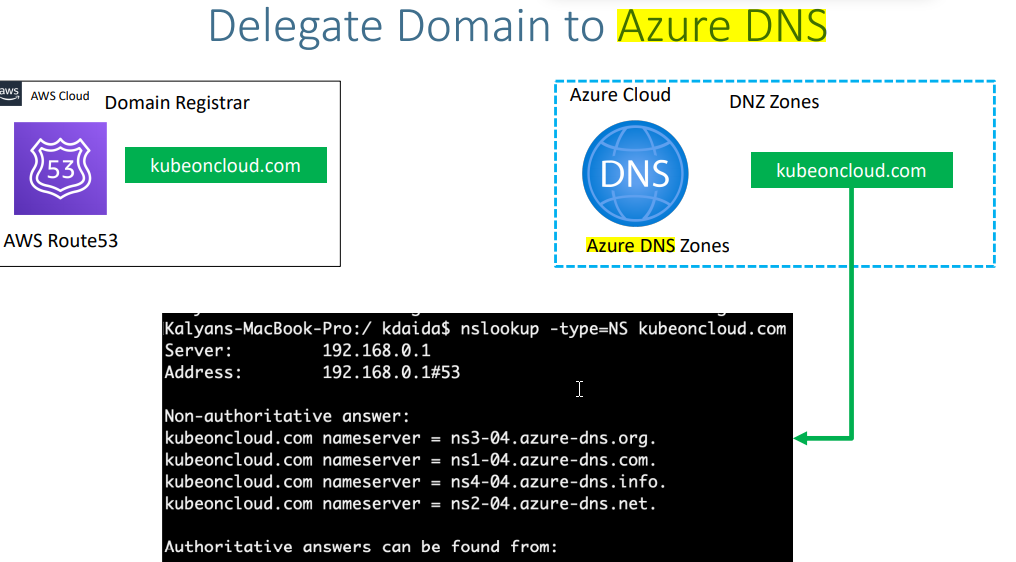
Azure DNS Zones -Delegate a DNS Domain to Azure DNS

Delegate a domain is nothing but for that respective domain underline name servers will be changed AWS Name servers to Azure name servers. Change the name servers whatever it is correctly pointing (like go daddy) to azure dns zones









**DNS Zones - Create DNS Zone**

* Go to Service -> **DNS Zones**
* **Subscription:** StackSimplify-Paid-Subscription (You need to have a paid subscription for this)
* **Resource Group:** dns-zones
* **Name:** kubeoncloud.com
* **Resource Group Location:** East US
* Click on **Review + Create**

**Step-03: Make a note of Azure Nameservers**

* Go to Services -> **DNS Zones** -> **kubeoncloud.com**
* Make a note of Nameservers in Overview page

ns1-04.azure-dns.com.

ns2-04.azure-dns.net.

ns3-04.azure-dns.org.

ns4-04.azure-dns.info.

**Step-04: Update Nameservers at your Domain provider (Mine is AWS)**

* **Verify before updation**

nslookup -type=SOA kubeoncloud.com

nslookup -type=NS kubeoncloud.com

* Go to AWS Route53 (This is my Domain Provider)
* Go to Services -> Route53 -> Registered Domains -> kubeoncloud.com
* Click on **Add or edit name servers**
* Update Azure Name servers here and click on **Update**
* Click on **Hosted Zones**
* Delete the hosted zone with name **kubeoncloud.com**
* **Verify after updation**

nslookup -type=SOA kubeoncloud.com 8.8.8.8 (service )

nslookup -type=NS kubeoncloud.com 8.8.8.8 (Name server location)

**External DNS with Ingress service**