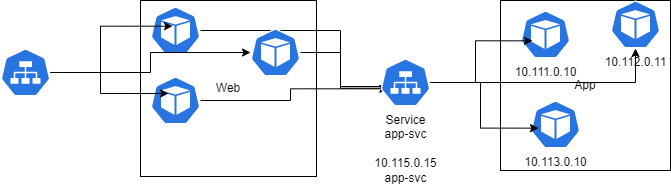
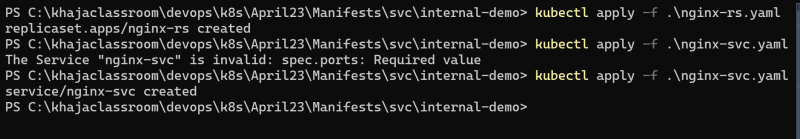
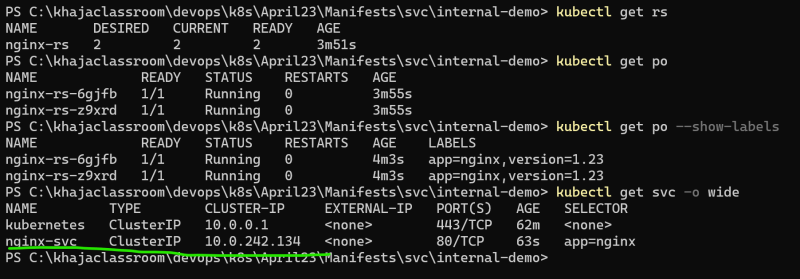
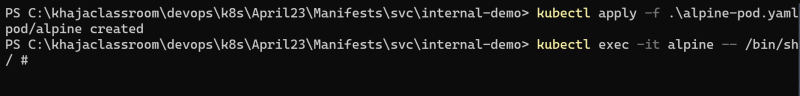
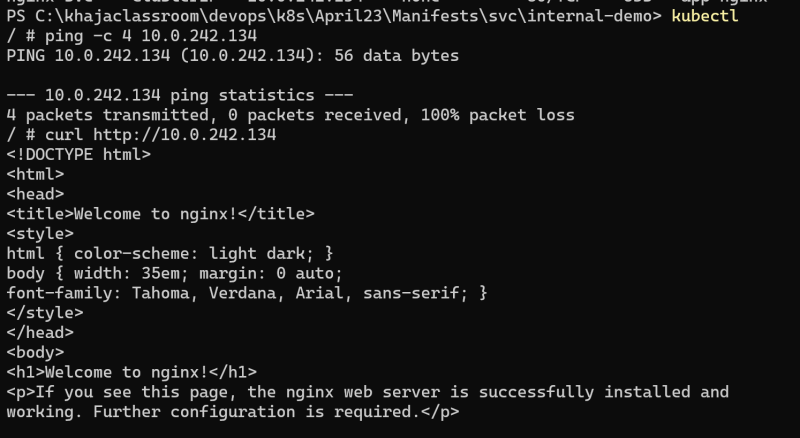
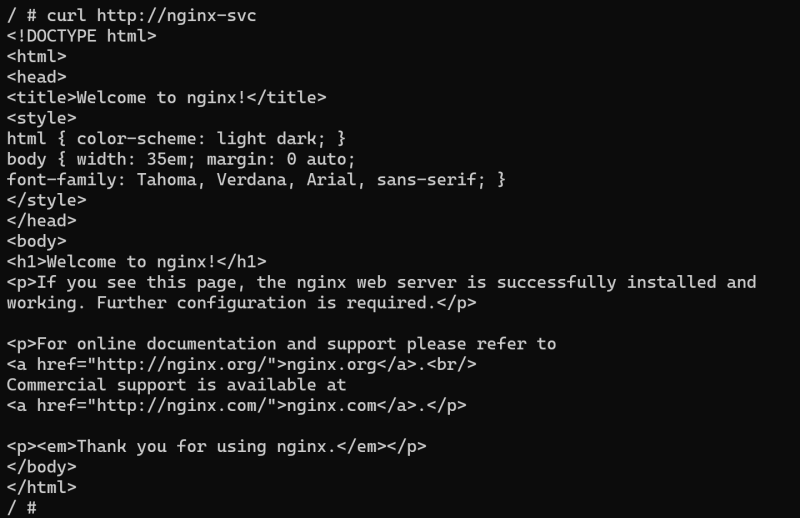
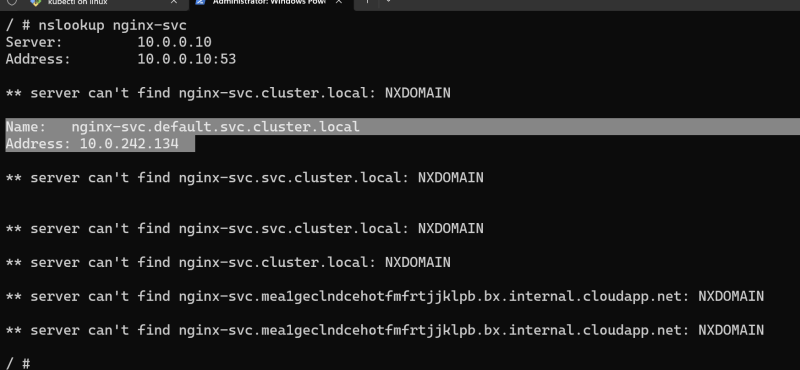
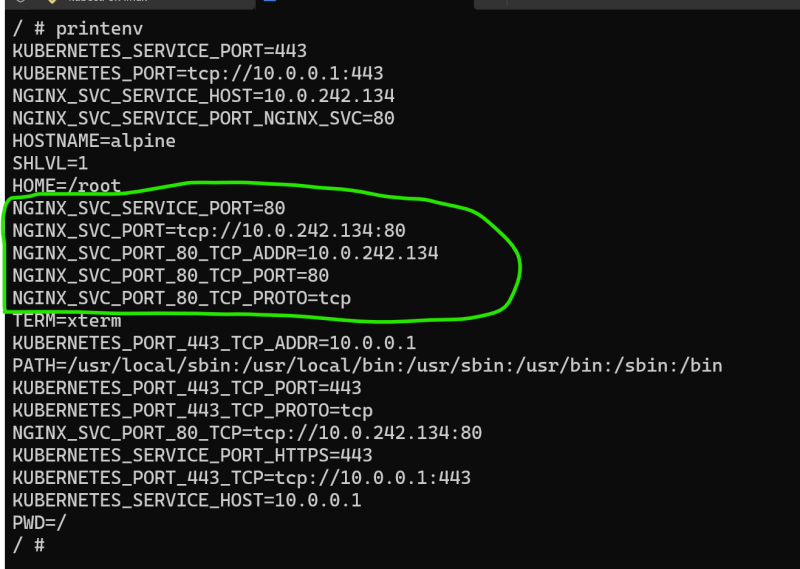
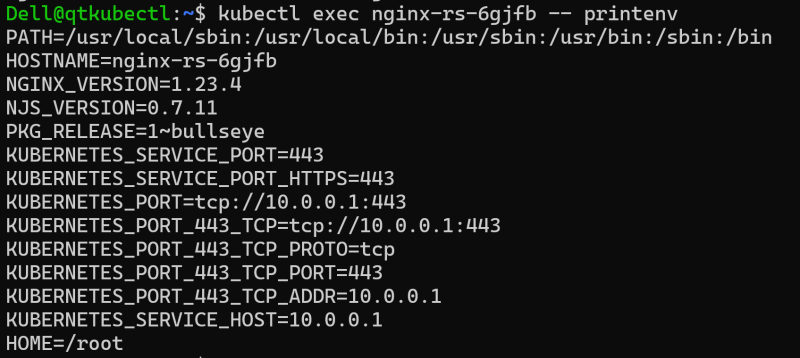
**K8s Service**

**Exposing Applications running in cluster to externally as well as internally when scaled**

* Every pod gets a unique ip and name.
* Connecting from one pod to other on the basis of name/ip might not be a good idea as pods are controlled by replicasets or other controllers
* K8s has a service which helps us in connecting to pods with similar behaviour but by using labels.
* Each service gets a ip address and this is virtual ip which helps in forwarding traffic to one of the pod based on labels. This ip is called as cluster ip
* Services can be exposed to external world  
  
* Service is similar to layer 4 load balancer
* <https://kubernetes.io/docs/concepts/services-networking/service/> for official docs

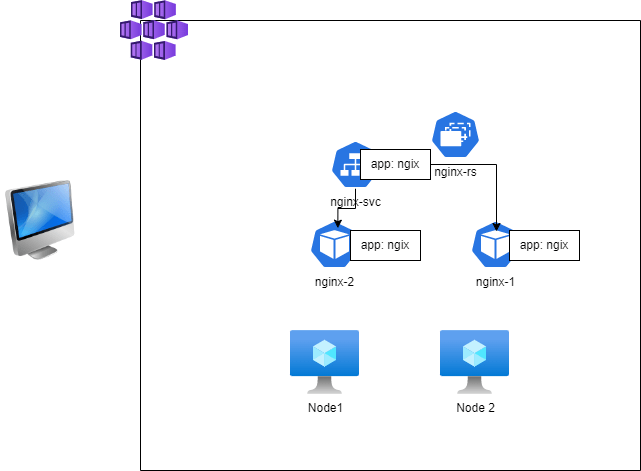
**Internal Communication using k8s service**

* Consider the following
  + We have an alpine pod which needs to consume nginx
  + but nginx is a replica set and there can be n replicas
* Let’s create a nginx-rs  
    
  
* Create an alpine pod and login into that  
  
* ping nginx-svc by its ip address and try accessing the web page using curl  
  
* access nginx-svc by using name  
  
* now do nslookup based on name  
  
* Look into environment variables in alpine pod (Alpine was created post nginx service creation)  
  

Look into environment variables in nginx pods (These were created prior to nginx service)  


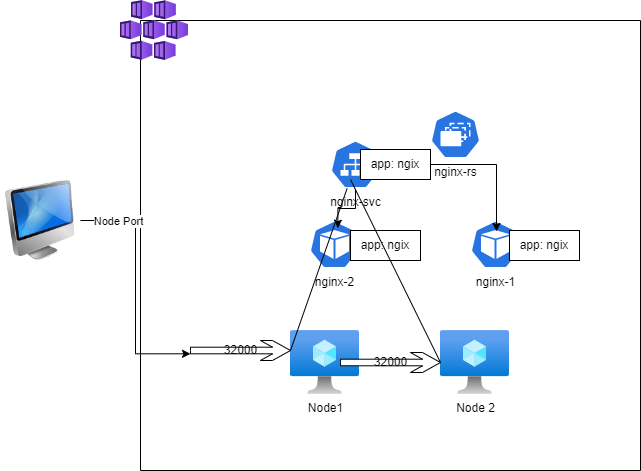
<https://github.com/asquarezone/KubernetesZone/commit/6b48735bf6627279d4339217f56ca04cff92b76d> for internal communication manifests

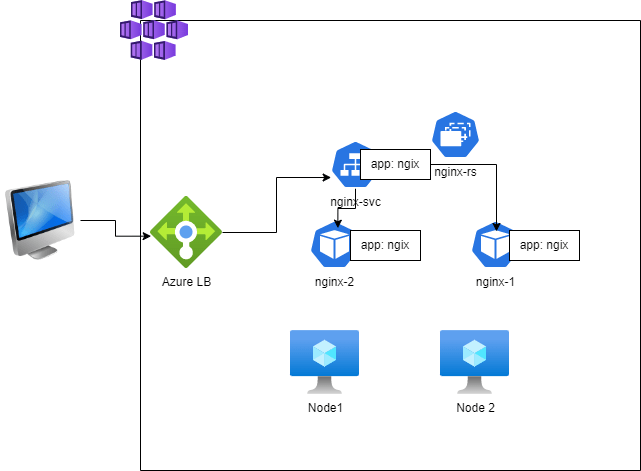
**External Communication using k8s service**

Some user external to k8s cluster wants to access nginx  


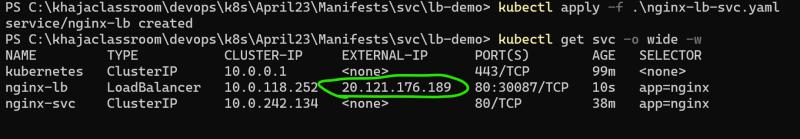
Kubernetes has the service publishing types <https://kubernetes.io/docs/concepts/services-networking/service/#publishing-services-service-types>

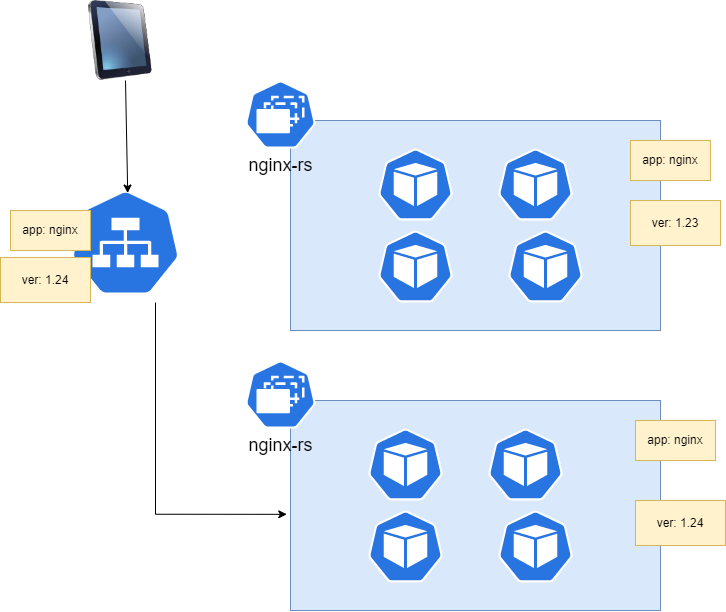
Cluster ip: internal communication

Node Port: k8s will expose the application on a port on every node in k8s cluster.  


Load Balancer: This is generally used with managed k8s clusters  


ExternalName: Creates a CNAME record that can be used in your DNS Servers

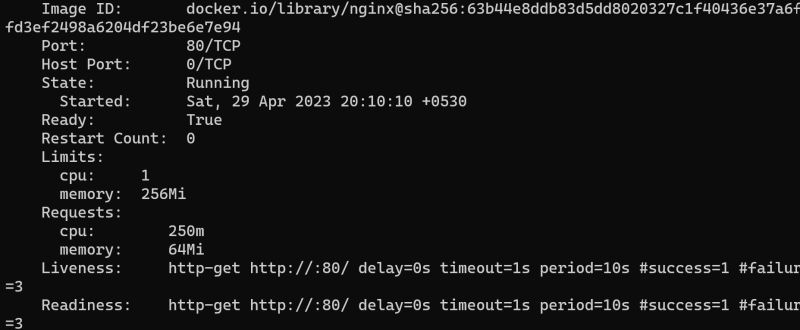
* We have created a manifest with loadBalancer  
  
* <https://github.com/asquarezone/KubernetesZone/commit/aec7b6a9e647177b76c453fa739da7be5205ead2> for the spec

When upgrading to newer versions of Pods ensure right set of labels are present on k8s service selector  


**Health Checks/Probes for containers in k8s Pods**

* <https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-startup-probes/> for official docs
* K8s supports 3 kinds of checks
  + liveness probe: if this check fails Kubernetes will restart the container.
  + readiness probe: if this check fails the pod will be removed from service (pod will not get requests from service)
  + startup probe: This checks for startup and until startup is ok, the other checks will be paused.
* Probes or checks can be performed by
  + exec: run any linux/windows command which returns status/exit code.
  + http: we send http request to the application. based on status codes we can decide <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>
  + grpc: This communicates over grpc
  + tcp: send tcp request
* <https://github.com/asquarezone/KubernetesZone/commit/cac117360161d61012649baf1401ec6ae87d9cbb> for the manifests with health checks

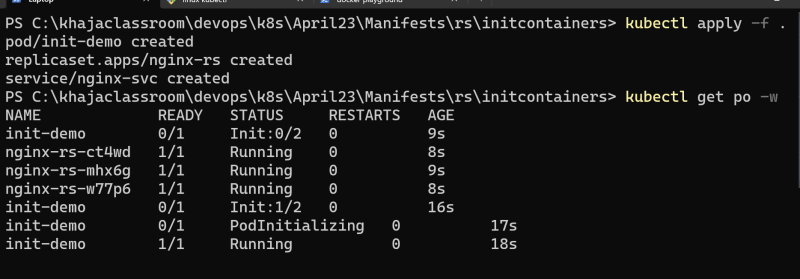
**Run Pods with specific Resources (CPU/Memory)**

* <https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/> for the official docs on Resource Limits
* <https://github.com/asquarezone/KubernetesZone/commit/910a088cab6e157f79468910494515d8141c220c> for the manifests with requests and limits  
  

**Container Types in Pods**

* We have 3 types of containers
  + containers: these are why we write pod spec
  + init containers: <https://kubernetes.io/docs/concepts/workloads/pods/init-containers/>
  + ephemeral containers

**Init Containers**

* <https://github.com/asquarezone/KubernetesZone/commit/41e66c454ef62cc5f9c50e72a0c8fa672ee2a727> for the manifests added  
  

**Ephemeral Containers**

<https://kubernetes.io/docs/concepts/workloads/pods/ephemeral-containers/> for official docs

**Node Usecases**

* How to schedule a Pod on a Particular node
* How to stop assigning more pods to a node
* How to move all the pods running a node to other ndoe

**headless service spec**

---

apiVersion: v1

kind: Service

metadata:

name: nginx-svc

spec:

selector:

app: nginx

type: ClusterIP

clusterIP: None

ports:

- name: nginx-svc

port: 80

targetPort: 80

protocol: TCP

Headless service returns the ips of the pods returned by selector. This is used in stateful sets