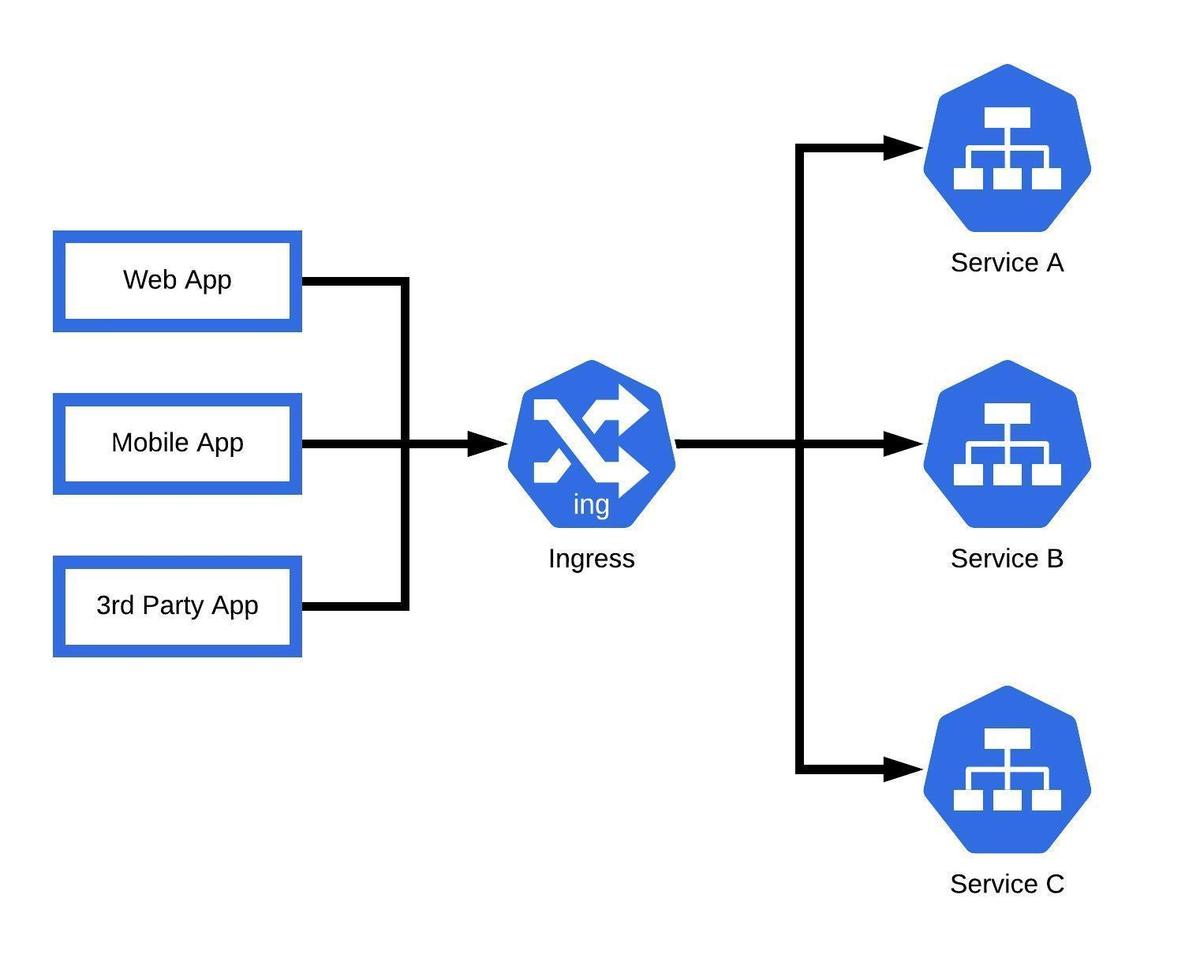
**Set up Ingress on Minikube with the NGINX Ingress Controller**

The objective of exposing an application using Ingress in Minikube is to provide a stable and centralized entry point for accessing multiple services within the Kubernetes cluster.

Solution:



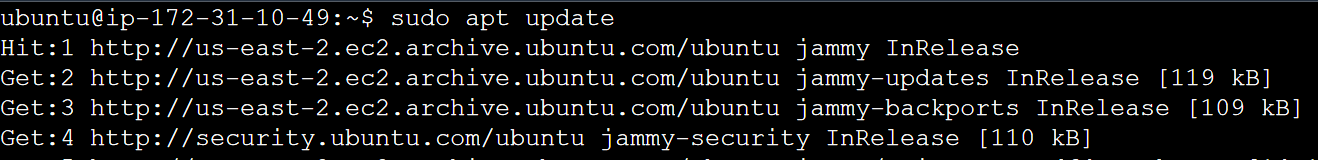
Requirements:

1. Docker
2. Kubectl
3. Minikube
4. EC2 or Local machine

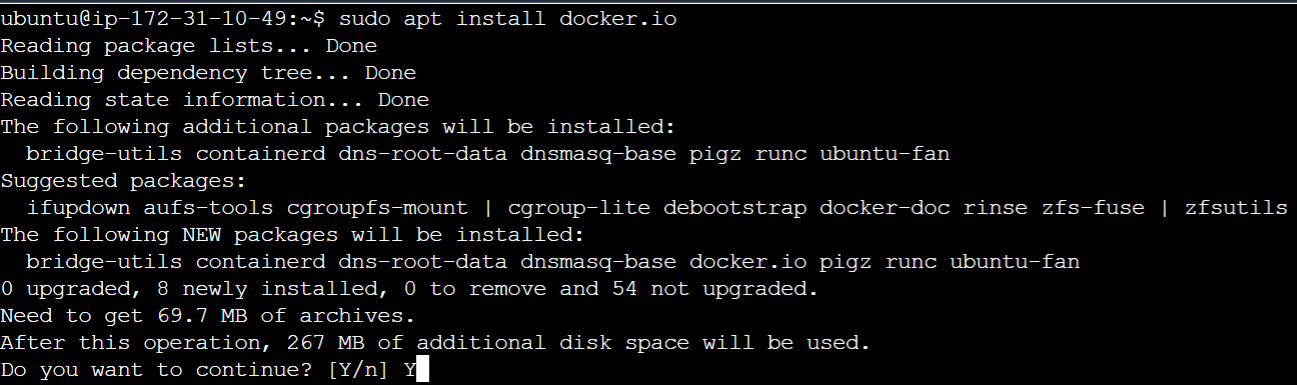
**Docker:**

* Install docker for start Minikube using docker driver.
* Use this below command to install docker in Ubuntu

**sudo apt update**



**sudo apt install docker.io**



* Once docker installation done we need configure user to access docker without root permission.

**sudo usermod –aG docker $USER && newgrp docker**

**Kubectl:**

* Using curl command to download kubectl

**curl -o kubectl** [**https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/linux/amd64/kubectl**](https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/linux/amd64/kubectl)

* Use the below command to add executable permission for kubectl.

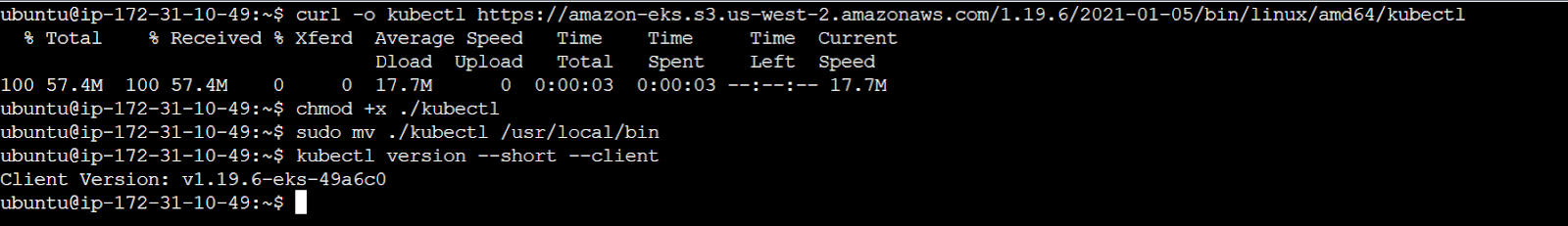
**chmod +x ./kubectl**

* Next move our kubectl into bin directory for access kubectl from anywhere.

**sudo mv ./kubectl /usr/local/bin**

* Check the kubectl client version

**kubectl version --short –client**

****

**Minikube:**

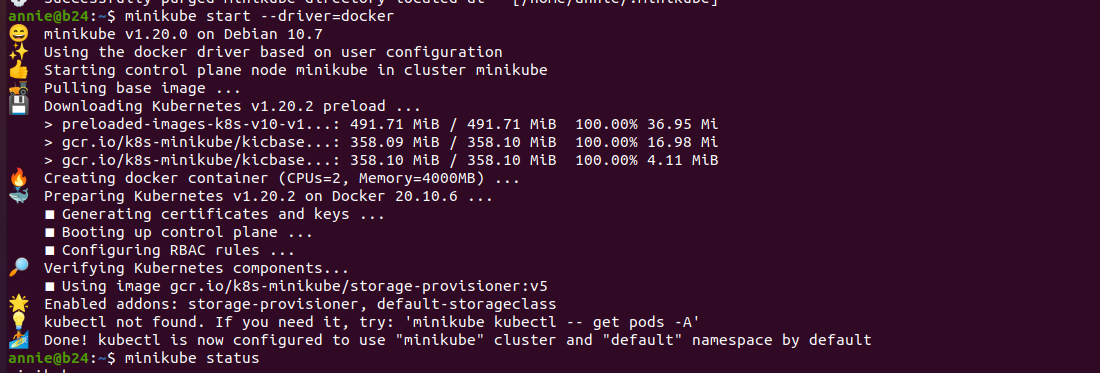
* If you are using EC2 that should contain minimum 2 CPU and 4 GB RAM to start Minikube.
* So choose t2.medium and larger instance.
* If you are using local VM that also need the same requirements to start Minikube.

**curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64**

**sudo install minikube-linux-amd64 /usr/local/bin/Minikube**

* Once installation done start Minikube cluster using the below command.

**minikube start –driver=docker**

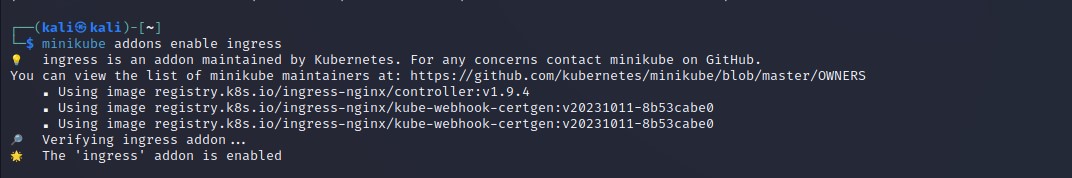


* Use minikube status command to check cluster status.

**Ingress Setup in Minikube:**

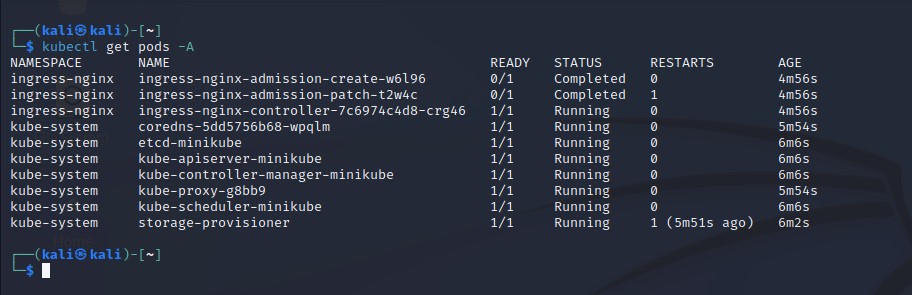
* Minikube has already contains many add on feature to extend minikube capabilities.
* Use this below command to activate ingress controller in minikube.

**minikube addon enable ingress**



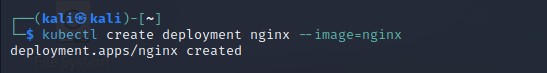
* Once ingress enabled check pods using Kubectl and ingress pods are running.

**kubectl get pods -A**

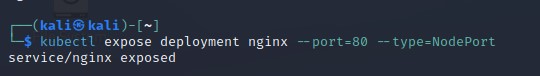


* Next we can create a deployment and expose the service.

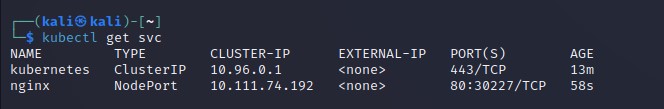
**kubectl create deployment nginx –image=nginx**



**kubectl expose deployment nginx –port=80 –type=NodePort**

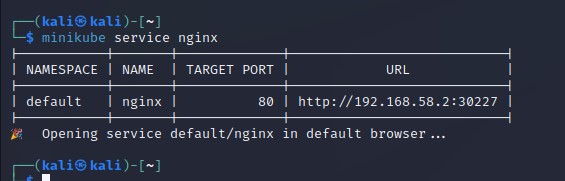


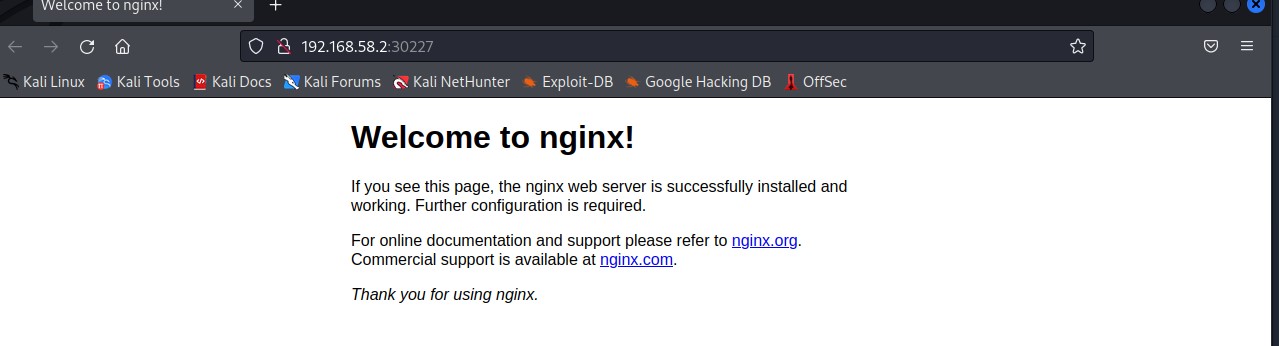
* Use **kubectl get svc** command to know about exposed services.



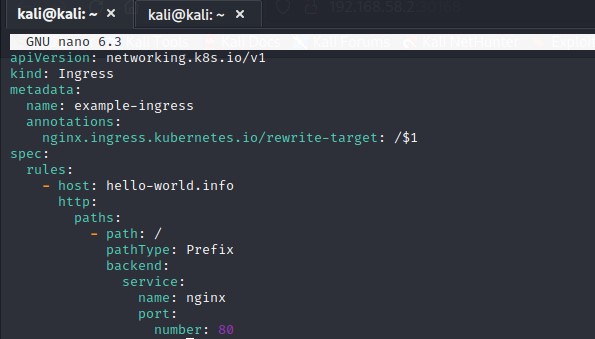
* Use minikube service command to open the exposed service in default web browser.

**minikube service nginx**

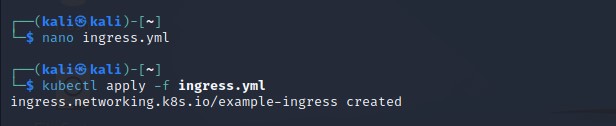


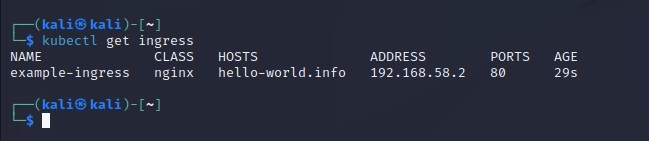


* Now we need to create an ingress.yaml to expose our service using ingress controller.



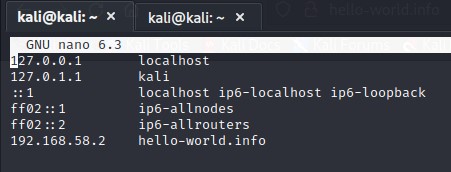
* In this case I used hello-world.info as my domain for exposing our service.
* Next apply that ingress file to expose our service into ingress.



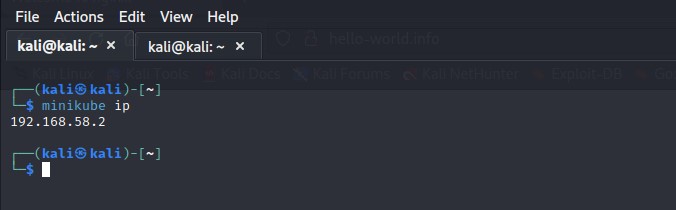


* We need to add our host name and ip address in the hosts file located in /etc.

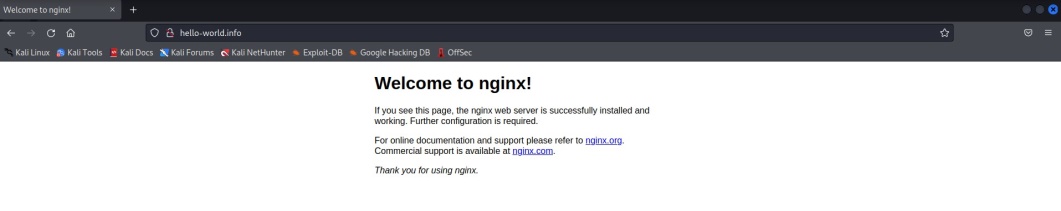




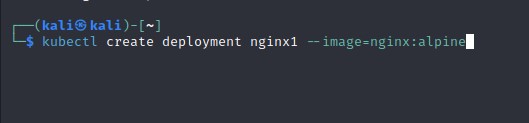
* Find minikube ip address using **minikube ip** command.

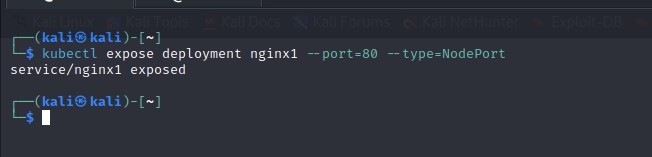


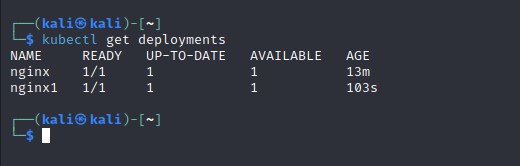
* Now check our domain that was set in ingress.yml config.

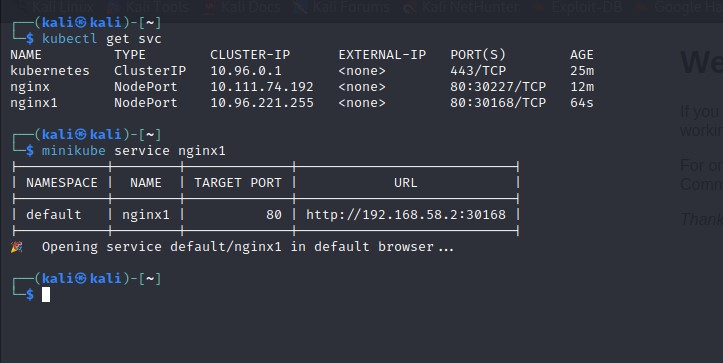


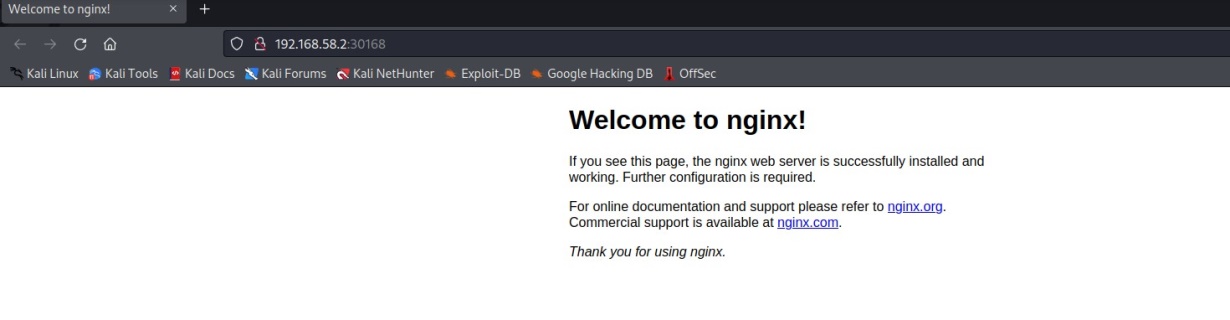
* Next we need to deploy other application for testing our same ingress works with different application.



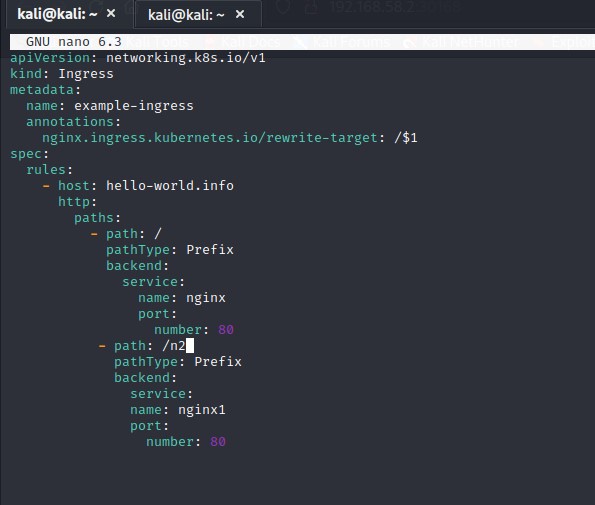




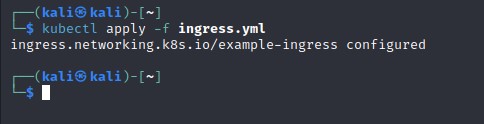




* Now we need to add some extra configuration on the same ingress.yml for exposing this new service as ingress.



* Apply the changes using kubectl command.



* Final output looks like the below image that 2 different applications that accessed by the same ingress and path that separated the applications.

