############################# Pre-requisite ####################################

# # An AKS cluster on Azure

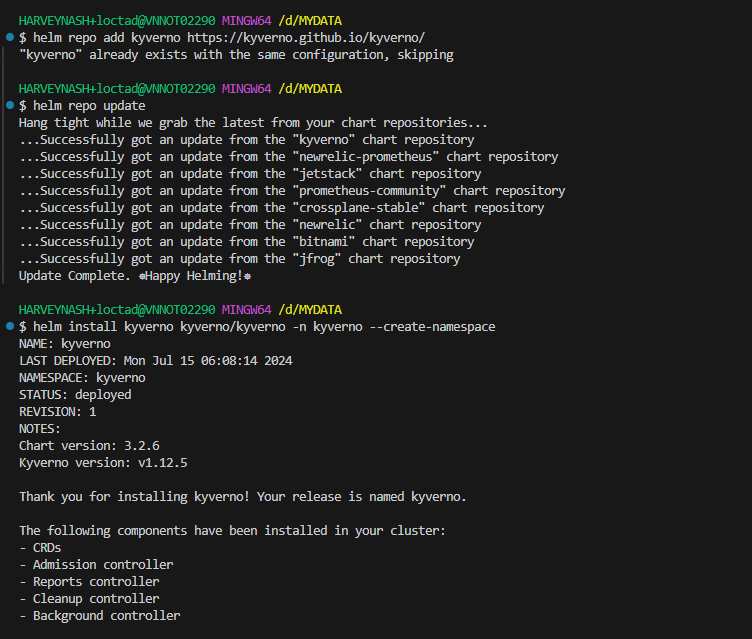
############################# ###############################################

**Install Kyverno on AKS cluster**

*helm repo add kyverno https://kyverno.github.io/kyverno/*

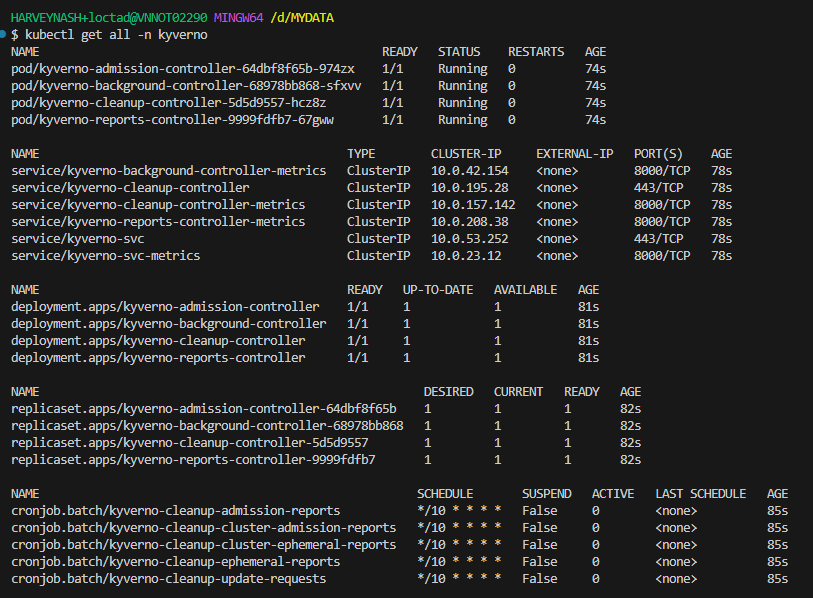
*helm repo update*

*helm install kyverno kyverno/kyverno -n kyverno --create-namespace*



Run the following command to see all of the resources in the Kyverno namespace.

*kubectl get all -n kyverno*



**1. Create a Kyverno policy blocks a Kubernetes resource from being deployed with default namespace.**

Now that Kyverno is installed, let’s take a look at policy configurations.

Luckily, Kyverno already has a ton of policies that are available.You can see a ton of the policies [Disallow Default Namespace | Kyverno](https://kyverno.io/policies/best-practices/disallow-default-namespace/disallow-default-namespace/)

*Notes:*

*validationFailureAction: Audit*

*Behavior: When set to audit, Kyverno allows the resource to be created or updated even if it violates one or more policies. However, Kyverno logs the violation and reports it through its policy reports. This mode is useful for monitoring and tracking policy violations without impacting the normal operation of your cluster.*

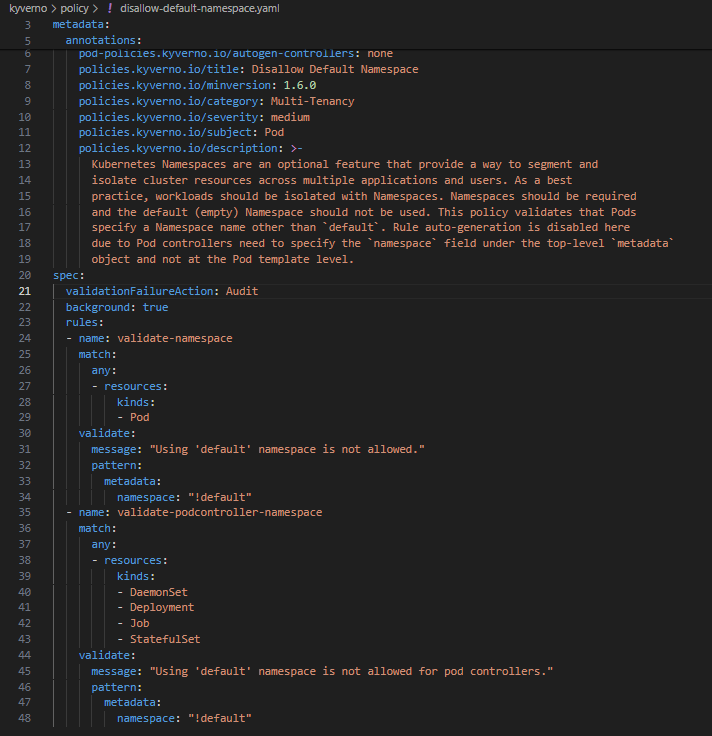
*Use Cases: This setting is particularly useful in a testing or development environment where you want to observe the effects of policies without enforcing them. It's also helpful when you're rolling out new policies and want to see their potential impact before making them mandatory.*

*validationFailureAction: Enforce*

*Behavior: Setting validationFailureAction to enforce makes Kyverno block the creation or updating of a resource that violates the policies. The request will be rejected, and the user will receive an error message explaining which policy was violated and why.*

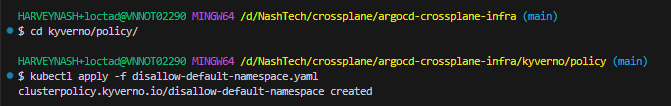
*Use Cases: This setting is crucial for environments where adherence to policies is mandatory for security, compliance, or governance reasons. Enforcing policies ensures that only compliant resources are allowed to run in your cluster, reducing the risk of security vulnerabilities or misconfigurations.*

Open up a new file and call it disallow-default-namespace.yaml, and paste in the following policy.



Apply the Manifest to your Kubernetes cluster.

kubectl apply -f disallow-default-namespace.yaml



**Create a Kubernetes deployment**

Next, create a new YAML file called nginx.yaml and paste in the following



Notice how it’s using **the default namespace**.

Apply the nginx.yaml Manifest and you should see an output similar to the below.

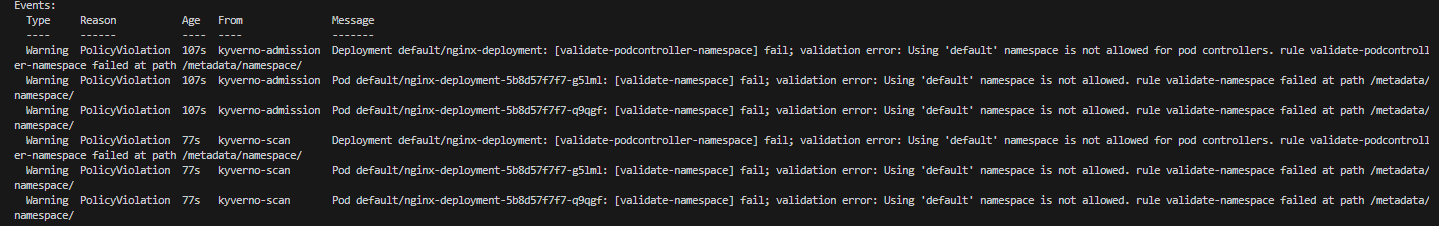
kubectl apply -f nginx.yaml



Because we set validationFailureAction: Audit

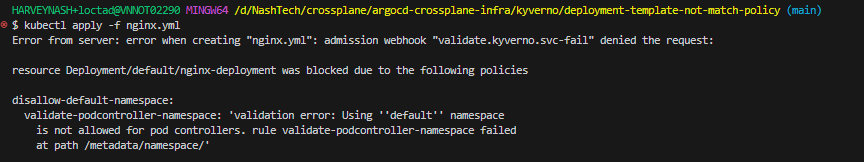
Behavior: When set to audit, Kyverno allows the resource to be created or updated even if it violates one or more policies. However, Kyverno logs the violation and reports it through its policy reports.

kubectl describe cpol disallow-default-namespace



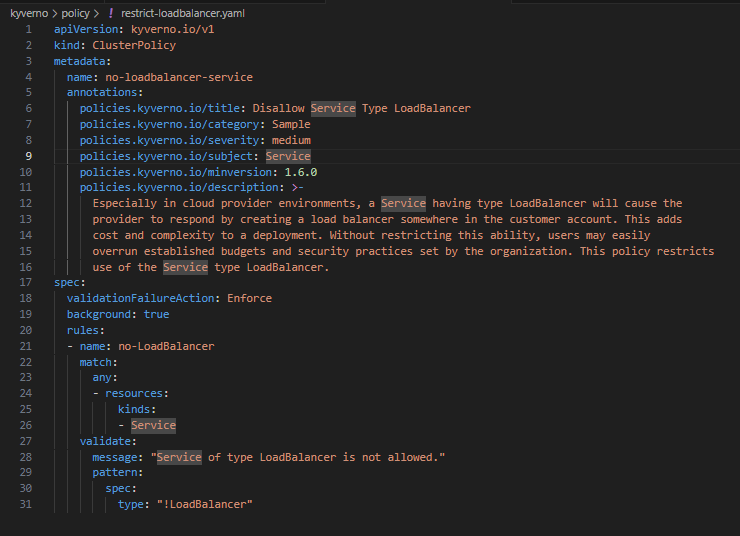
Update disallow-default-namespace.yaml change validationFailureAction: Audit to Enforce

Re-apply it and create a deployment.



**2. Create a Kyverno policy blocks a Kubernetes resource from being deployed with service LoadBalancer**

Open up a new file and call it restrict-loadbalancer.yaml, and paste in the following policy.



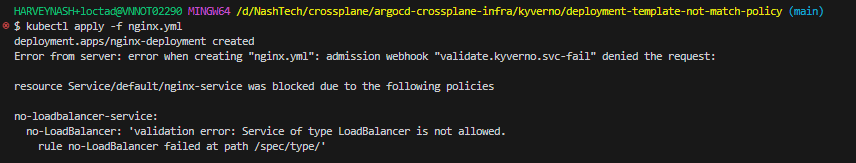
Apply it: kubectl apply -f restrict-loadbalancer.yaml



Update nginx.yml with service type is LoadBalancer file

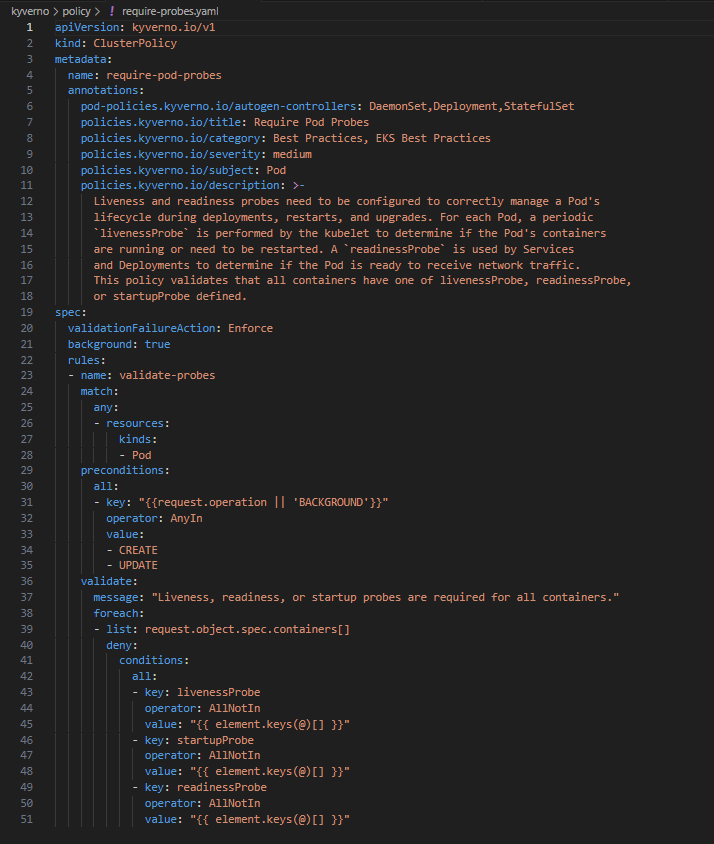


Apply it:

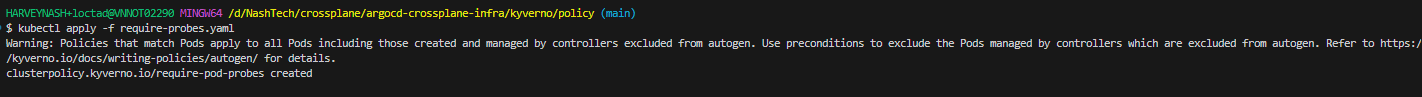


***3. Create a Kyverno policy require a Kubernetes resource from being deployed without Probes***

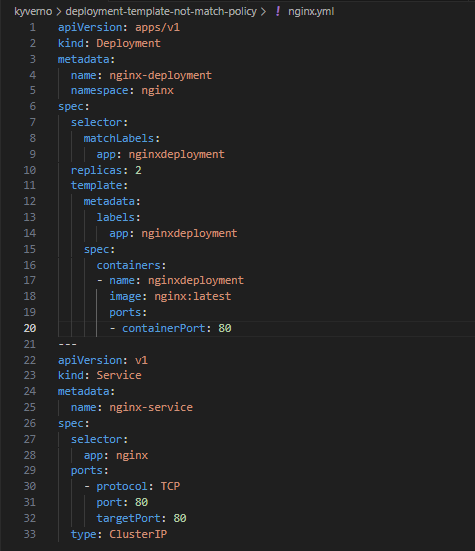
*Open up a new file and call it require-probes.yaml, and paste in the following policy.*



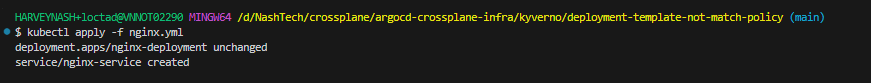
*Apply it: kubectl apply -f require-probes.yaml*



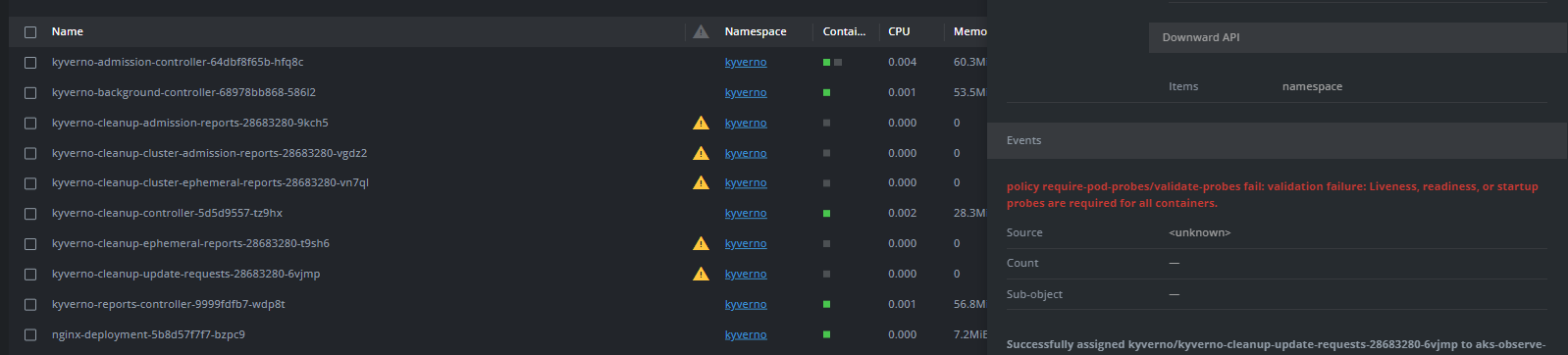
*Update nginx.yml without liveness and readness probes*

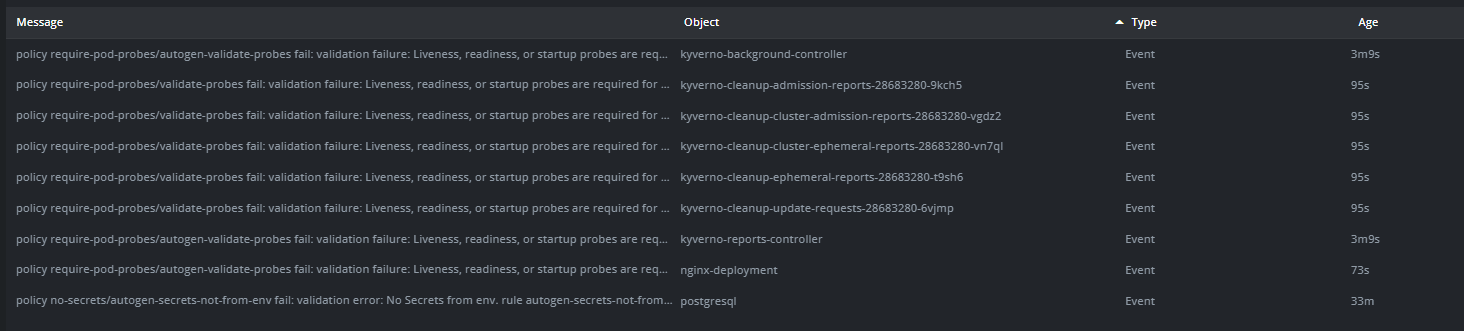


*Apply it:*



*Then you can access to Cluster to see the event like*





*Or you can run this command to see the detail of pod*

*kubectl describe pod nginx -n nginx*

