

**Lab Manual- AKS Taints and tolerations**

**Prepared for**:

**Date:** 18th Dec 2023

**Prepared by:**

Document Name: Lab Manual **Document Number** AZLabn916

**Contributor:**

Contents

[1. Objective 2](#_Toc157066608)

[2. Provide dedicated nodes using taints and tolerations 3](#_Toc157066609)

# Objective

As you manage clusters in Azure Kubernetes Service (AKS), you often need to isolate teams and workloads. Advanced features provided by the Kubernetes scheduler let you control:

* Which pods can be scheduled on certain nodes.
* How multi-pod applications can be appropriately distributed across the cluster.

This best practices article focuses on advanced Kubernetes scheduling features for cluster operators. In this article, you learn how to:

* Use taints and tolerations to limit what pods can be scheduled on nodes.
* Give preference to pods to run on certain nodes with node selectors or node affinity.
* Split apart or group together pods with inter-pod affinity or anti-affinity.
* Restrict scheduling of workloads that require GPUs only on nodes with schedulable GPUs.

***Taints, Tolerations, and Node Affinity: What are They?***

**Taints** are **labels** that you can apply to nodes in your AKS cluster. These labels indicate a node has specific characteristics, such as resource limitations or security requirements. Taints prevent Kubernetes from scheduling pods on a node that does not have a corresponding toleration. **Tolerations** are **settings** that you can apply to pods to allow them to be scheduled on nodes with specific taints. Using taints and tolerations, you can ensure that workloads are only scheduled on nodes meeting specific criteria.

**Node affinity**, on the other hand, is a way to influence scheduling decisions based on the labels that you apply to your nodes and pods. With node affinity, you can specify which nodes your workloads should be scheduled based on characteristics such as geographic location, hardware capabilities, or other custom labels.

# Provide dedicated nodes using taints and tolerations

Ngnixdemo.yaml

**apiVersion**: apps/v1

**kind**: Deployment

**metadata**:

**name**: nginx-deployment

**spec**:

**selector**:

**matchLabels**:

**app**: nginx

**replicas**: 2 *# tells deployment to run 2 pods matching the template*

**template**:

**metadata**:

**labels**:

**app**: nginx

**spec**:

**containers**:

- **name**: nginx

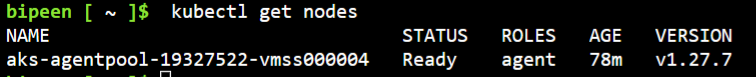
**image**: nginx:1.14.2

**ports**:

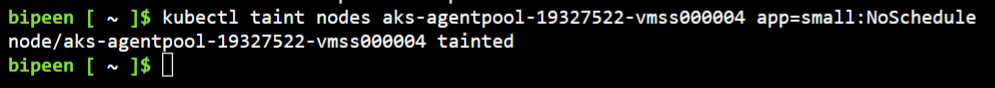
- **containerPort**: 80



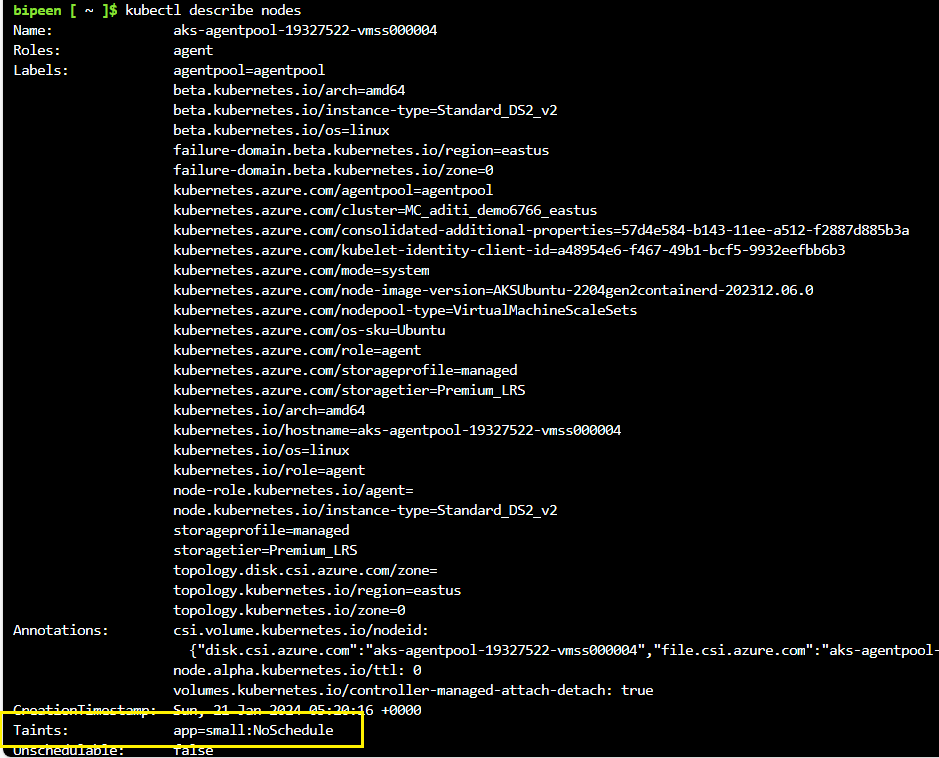
kubectl get nodes



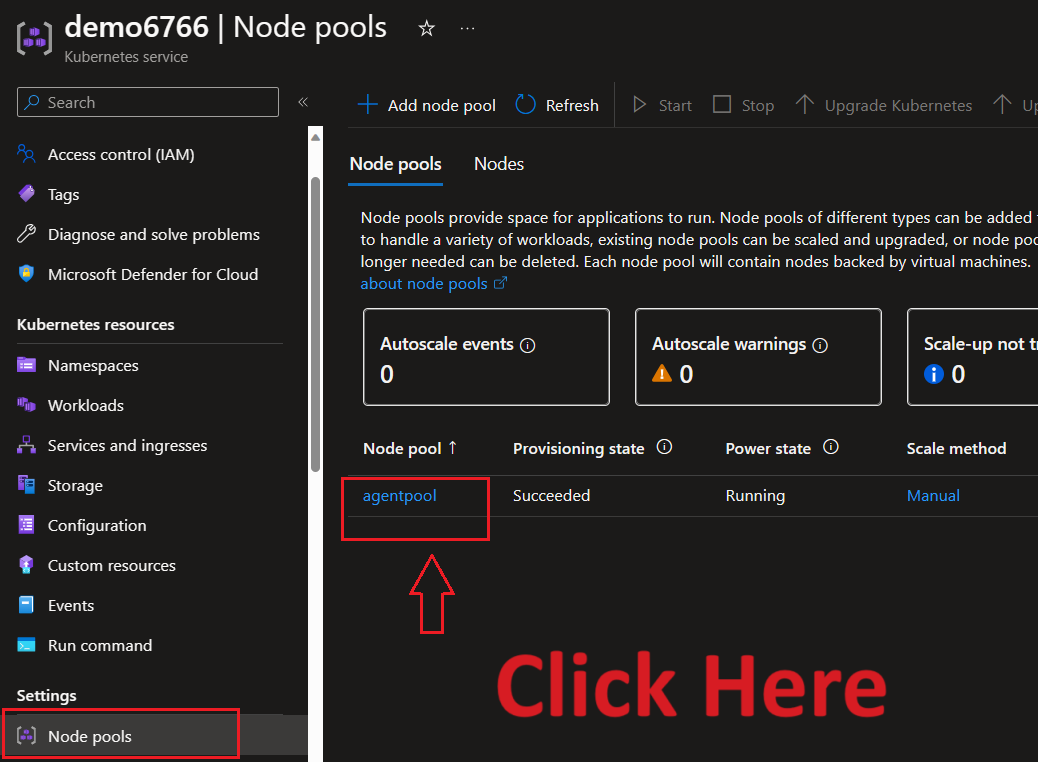
kubectl taint nodes aks-agentpool-19327522-vmss000004 app=small:NoSchedule

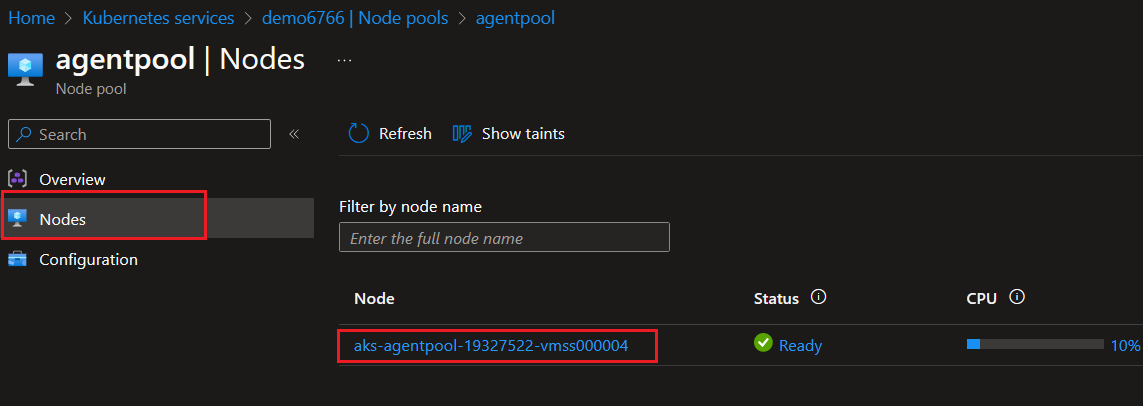


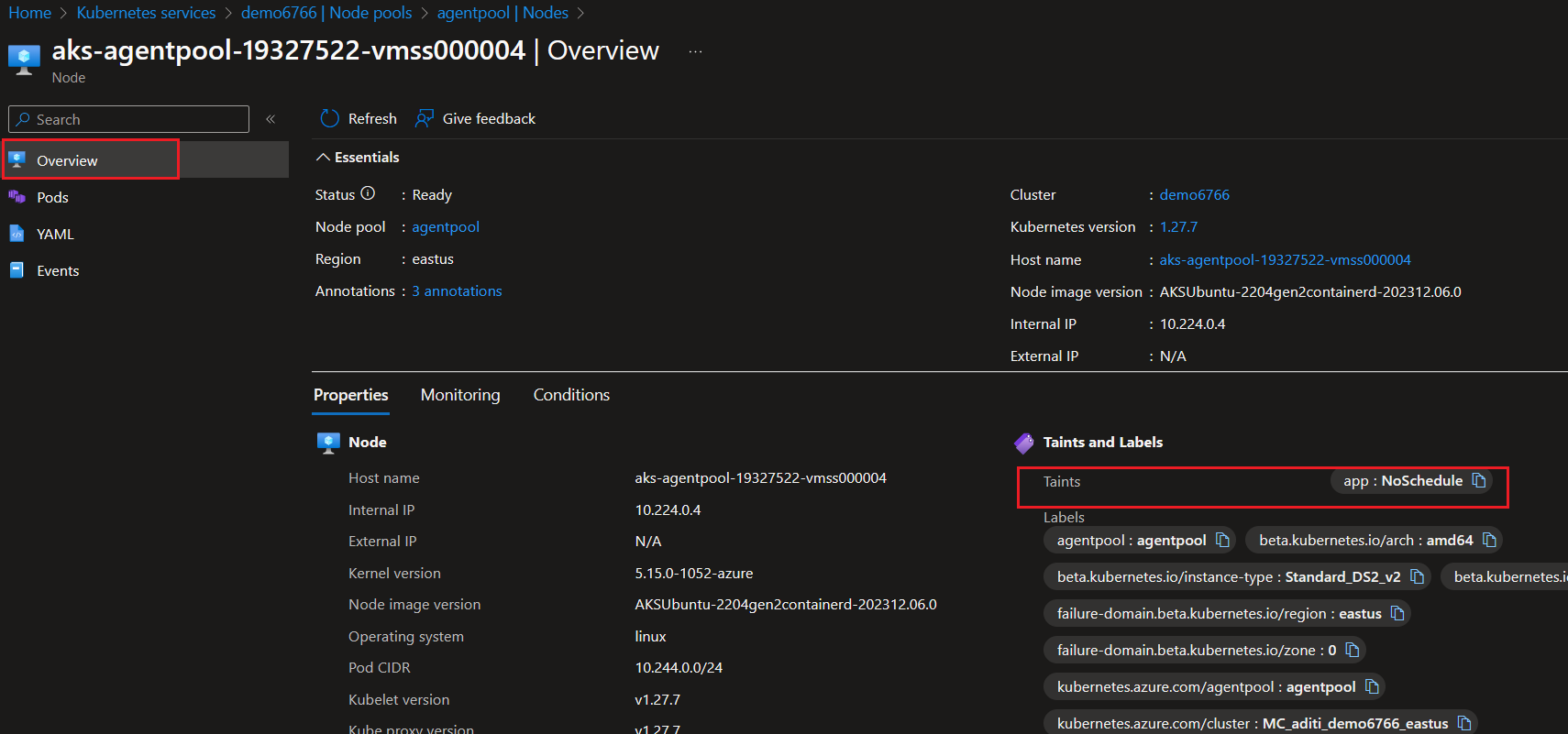
kubectl describe nodes



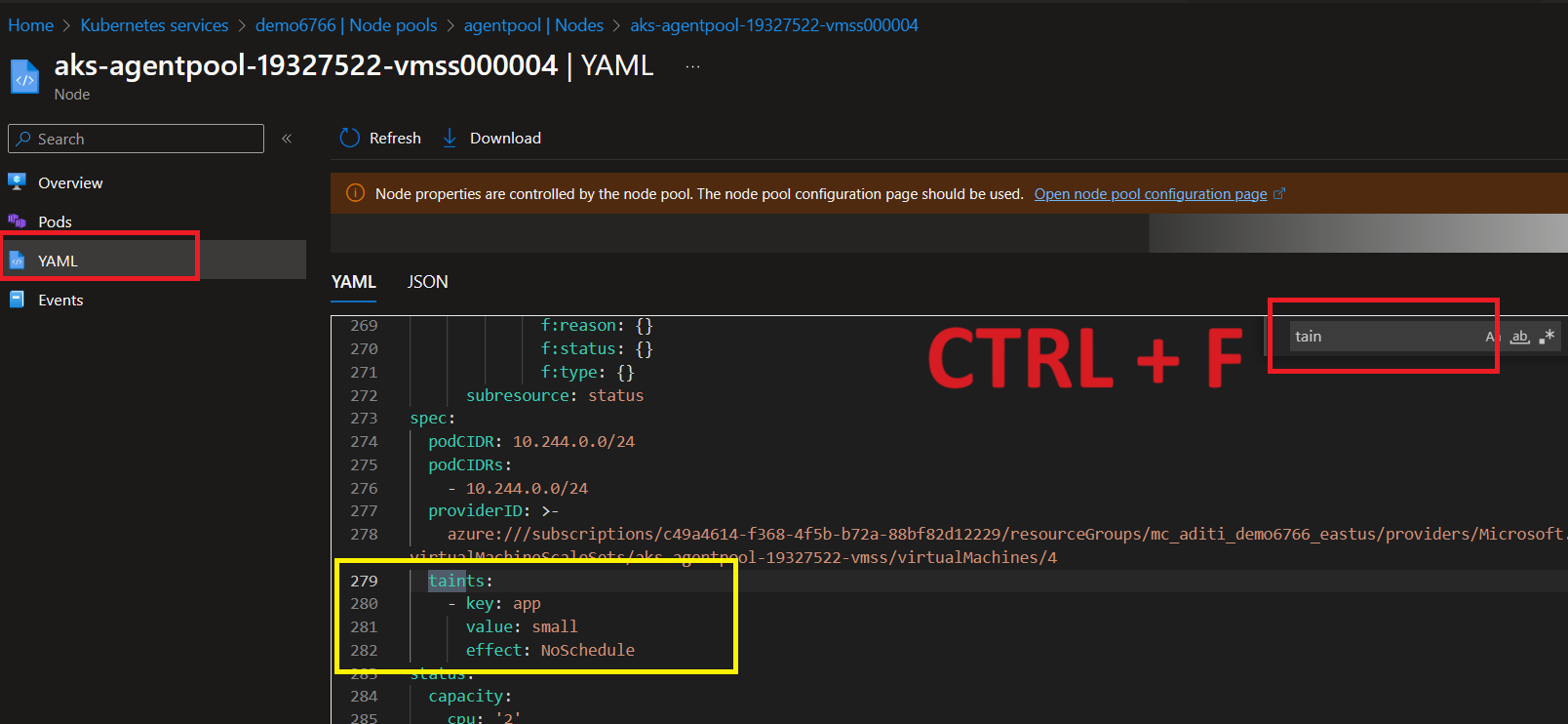
To Validate

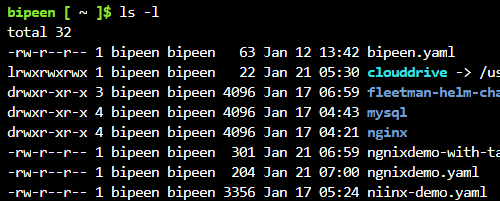






Click YAML and CTRL+F and search “Taints”

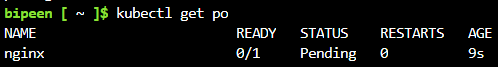




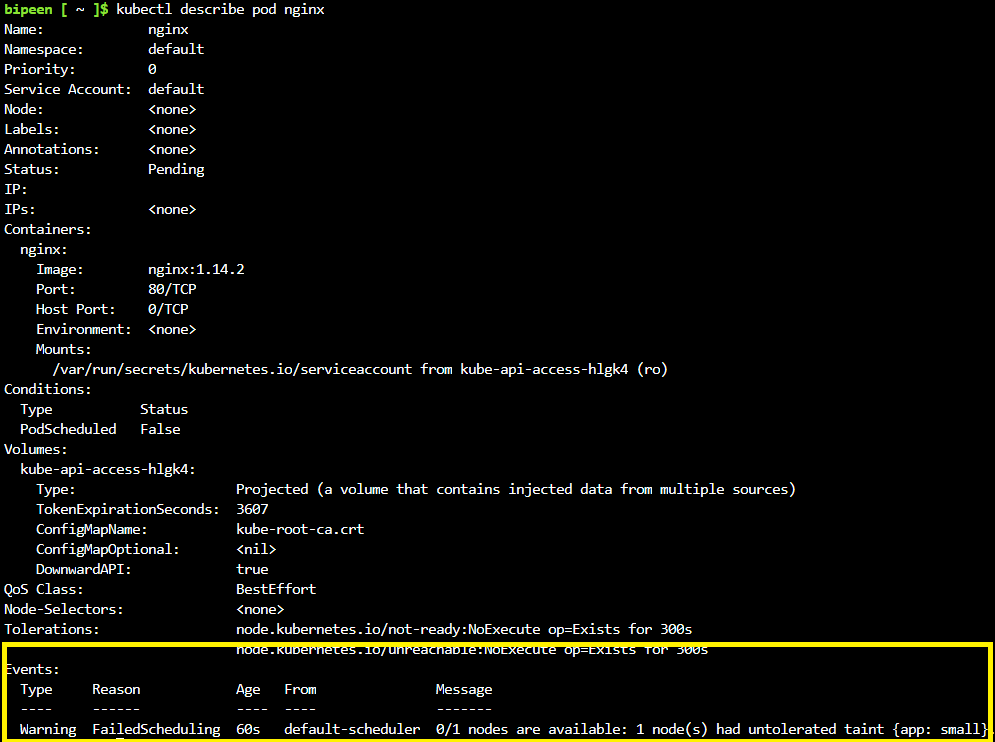
kubectl apply -f ngnixdemo.yaml



kubectl get po



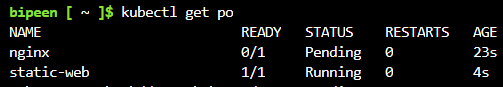
kubectl describe pod nginx



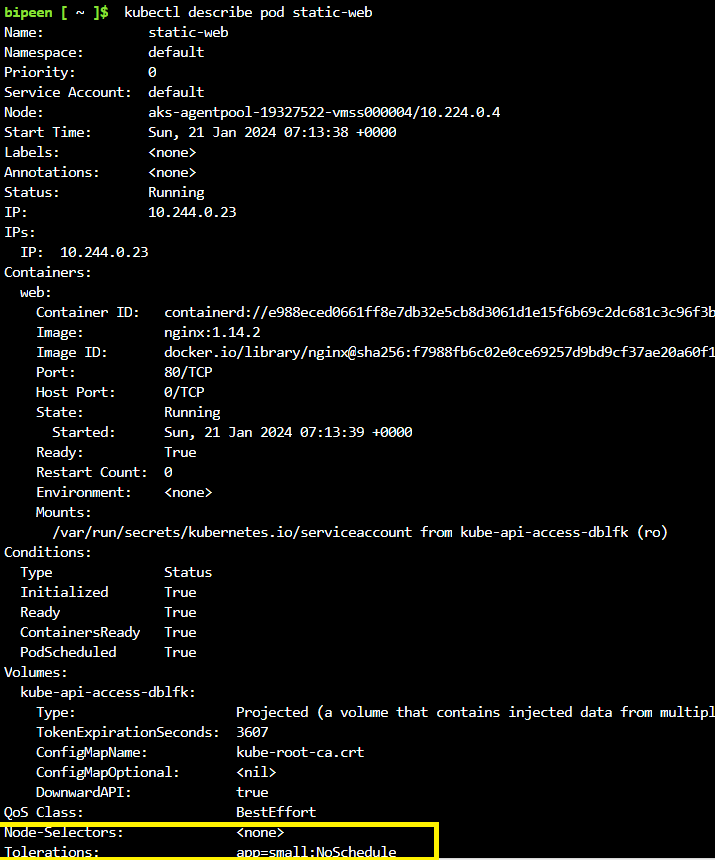
kubectl apply -f ngnixdemo-with-taints.yaml



kubectl get po



kubectl describe pod static-web

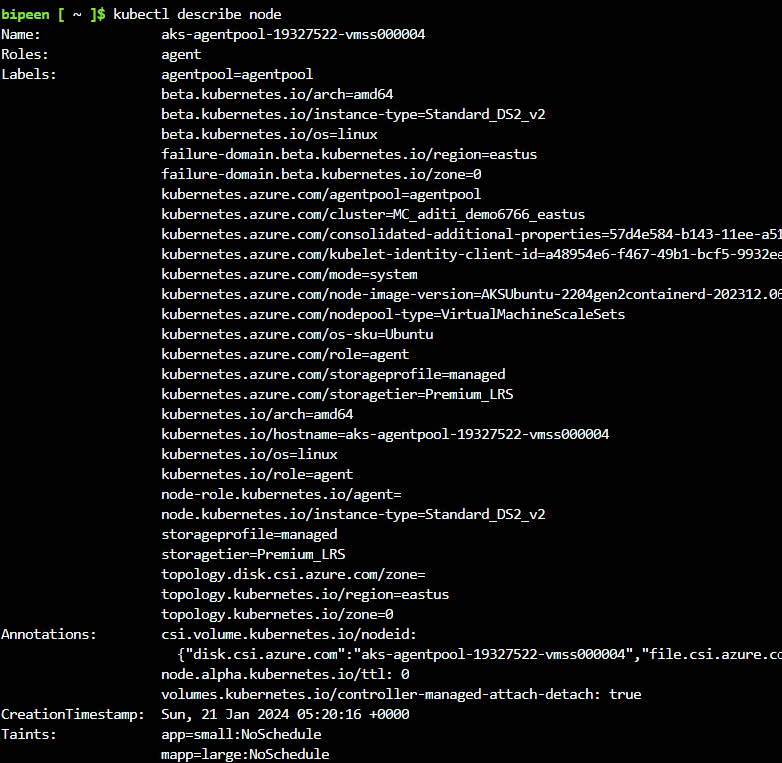


Now change the tolerance

kubectl taint nodes aks-agentpool-19327522-vmss000004 mapp=large:NoSchedule

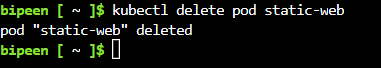


kubectl describe nodes

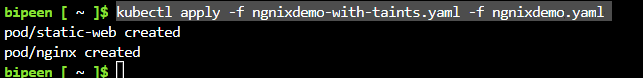


kubectl delete pod static-web

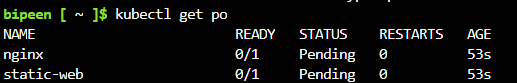
kubectl delete pod ngnix



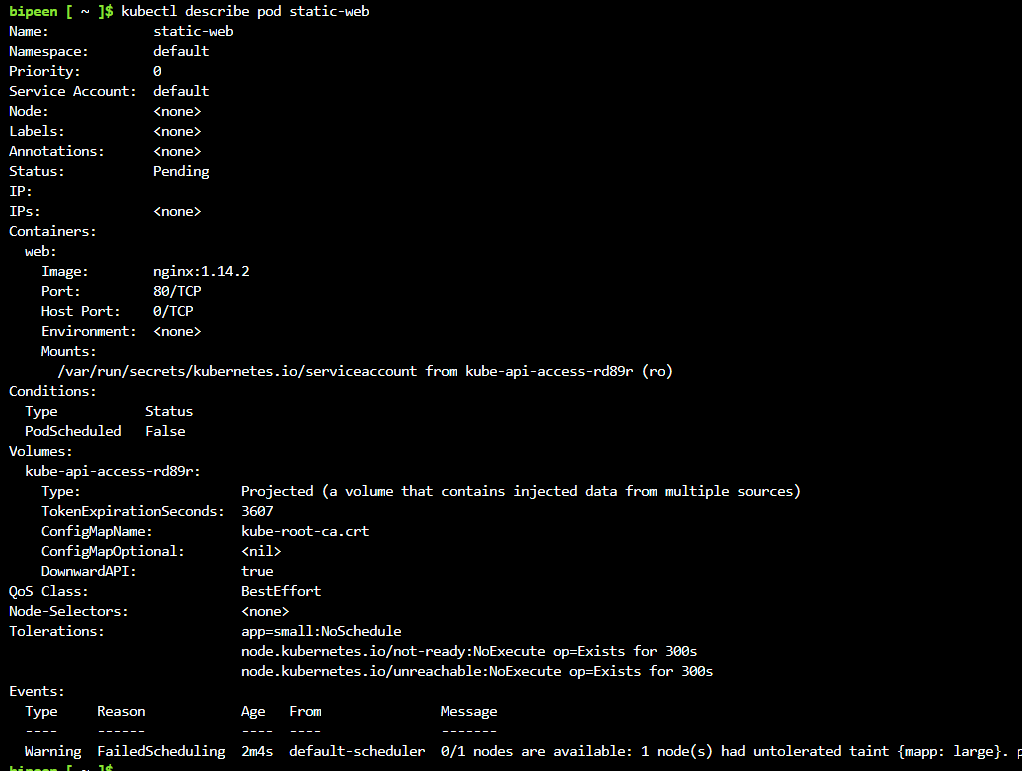
kubectl apply -f ngnixdemo-with-taints.yaml -f ngnixdemo.yaml



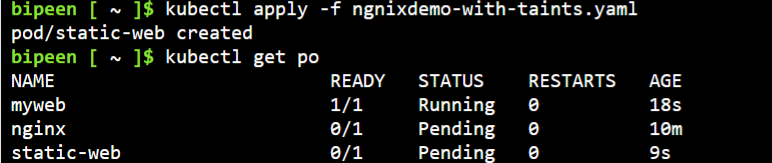
kubectl get pod



kubectl describe pod static-web



kubectl apply -f ngnixdemo-with-2taints.yaml



kubectl taint nodes aks-agentpool-19327522-vmss000004 mapp=large:NoSchedule-