deployment to create a pod by nginx image which will always run in one particular node only.

* To create a deployment that ensures an Nginx pod always runs on a specific node, you can use a node selector in your Kubernetes deployment YAML file. Here's an example:

apiVersion: apps/v1

kind: Deployment

metadata:

name: nginx-deployment

spec:

replicas: 1

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

containers:

- name: nginx

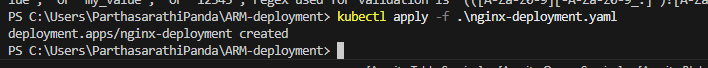
image: nginx:latest

nodeSelector:

kubernetes.io/hostname: <node-name>

* Replace <node-name> with the name of the node where you want the Nginx pod to run. This configuration ensures that the pod is scheduled on the specified node.

Check deployment



Check pods in that deployment

A black screen with white text

Description automatically generated

Check current Kubernetes version

A screenshot of a computer

Description automatically generated

Upgrade the cluster version.

A screenshot of a computer

Description automatically generated

Keep tracking on backend process. 1 new node will be created the workload will be transfer to new node. New node will be with new version. Then it will update the existing node. Then the workload will be transfer to the old updated node and the new node which was created for upgrade will be decom automatically.

Those internal process shifting workload pods to new (cordon, Drain) will be handled by Microsoft.

1. It will upgrade control plane

A screenshot of a computer

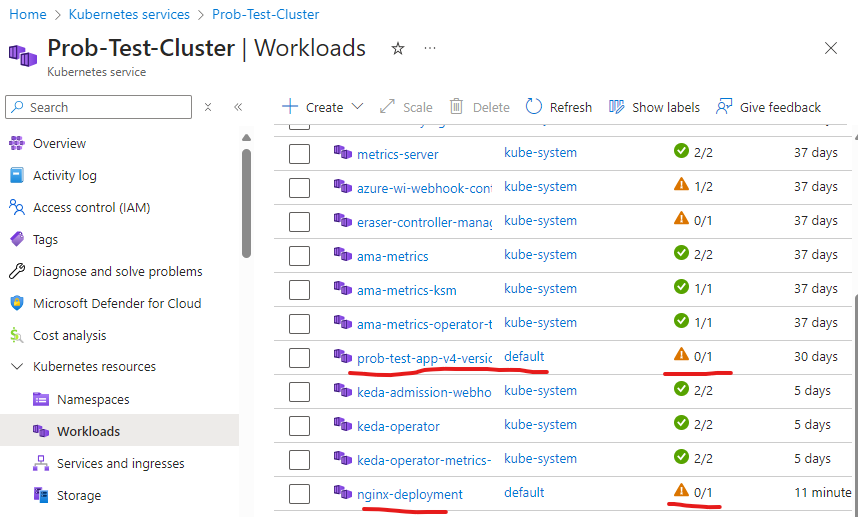
Description automatically generated

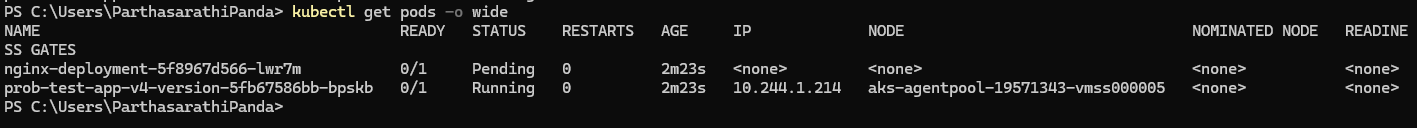
1. Spin up new node for update. We must have enough quota for this process to create new node with same size.

A screenshot of a computer

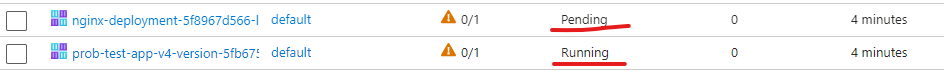
Description automatically generated

1. Shift workload to new node to update the existing node.





1. Note the nginx deployment pod is in pending state



1. The nginx pod did not choose the new node to run

A white screen with black text

Description automatically generated

1. The reason of this failure is we have mentioned that the pod of this deployment will run on selected node only.

A red line with white text

Description automatically generated

1. The new created node is deleted.

A screenshot of a computer

Description automatically generated

1. Nodepool is in upgrading state

A screenshot of a computer

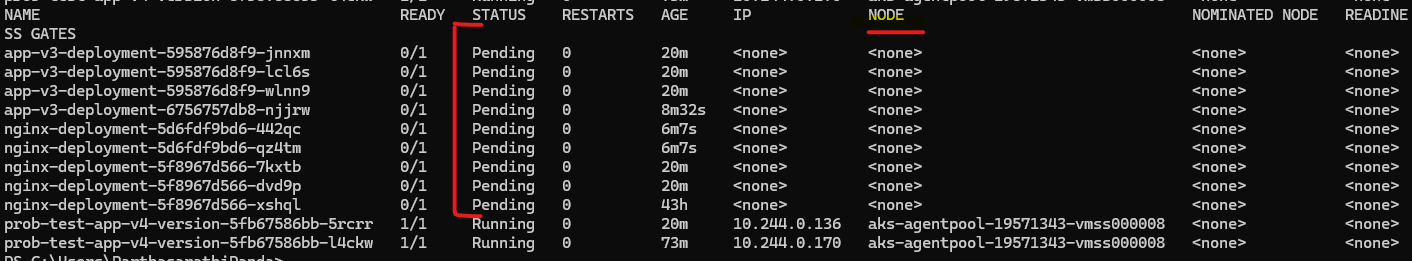
Description automatically generated

So the cluster upgrade is also in progress.

A screenshot of a computer

Description automatically generated

1. It got updated successfully.
2. I had stopped the aks and restarted. So, new node came up with name “aks-agentpool-19571343-vmss000008”. But, in my configuration I have mentioned the node selector “aks-agentpool-19571343-vmss000004”. So due to unavailability of that node 2 deployments pods did not run. Those are in pending state only.



1. Will change the configuration for one deployment. That is nginx-deployment and reapply the deployment.



Pods came to running state.

For Eriez the issue was with policy which was defined on min pod running count.

A screenshot of a computer error

Description automatically generated

What is Disruption?

* Somethin,  a break or interruption in the normal course or continuation of some activity, process, etc.

It’s basically interruption in ongoing /process.

What is Disruption in AKS?

There are two types of disruption in AKS which can make the pod unavailable.

Pods does not get terminate automatically. It get terminate due to either human deletion by admin or if the pod runs into any criticality such as error, memory or CPU issue. Etc..

Let’s deploy app v3 which have issue to run the pod so the pod will never run.