

**Lab Manual- AKS Troubleshooting Part 2**

**Prepared for**:

**Date:** 18th Dec 2023

**Prepared by:**

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

**Contributor:**

Contents

[1. Objective 3](#_Toc157448119)

[2. Install and Set Up kubectl on Windows 3](#_Toc157448120)

[3. Create 1 Node Azure Kubernetes Cluster 3](#_Toc157448121)

[4. Connect the Kubernetes Cluster from azure PowerShell 4](#_Toc157448122)

[5. Create a Deploy Kubernest Dashboard from helm chart 5](#_Toc157448123)

[6. Access the Kubernetes Dashboard ( Local PowerShell) 6](#_Toc157448124)

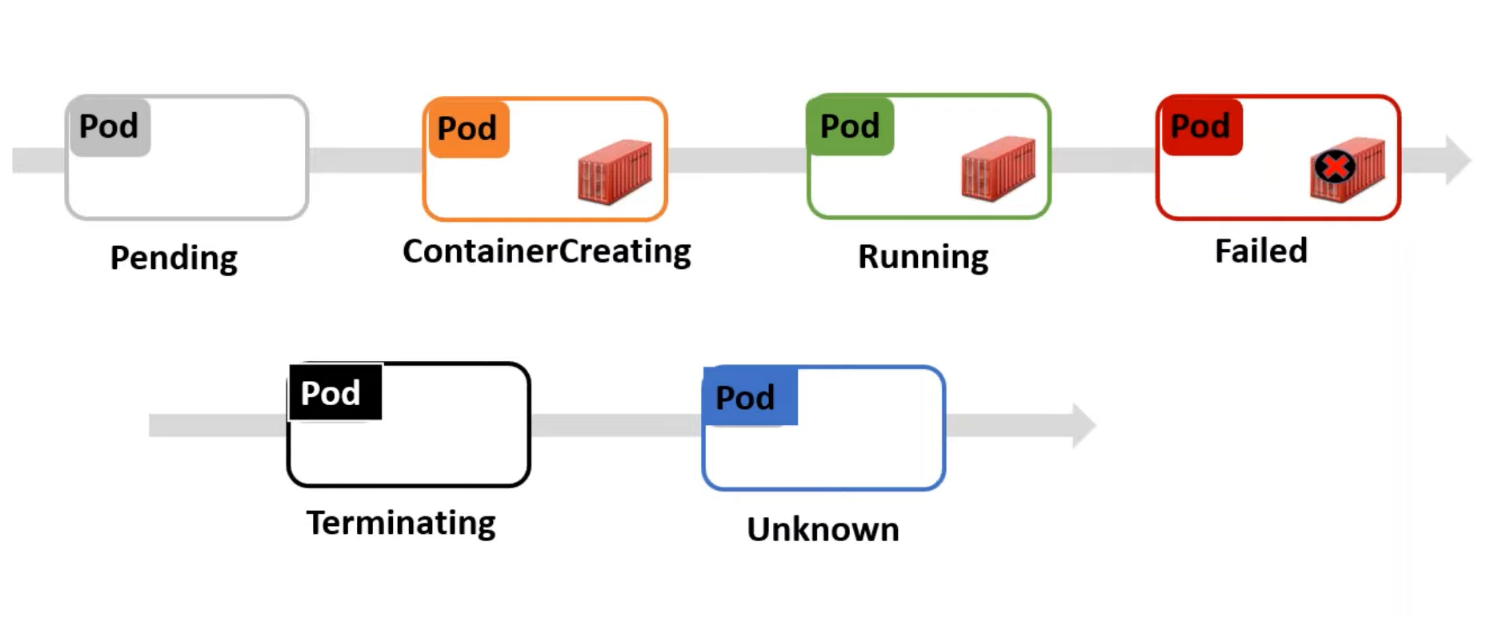
[7. Deploy Sample NGNIX Pod 10](#_Toc157448125)

[8. Check NGNIX Pod in Kubernetes Dashboard 12](#_Toc157448126)

# Objective

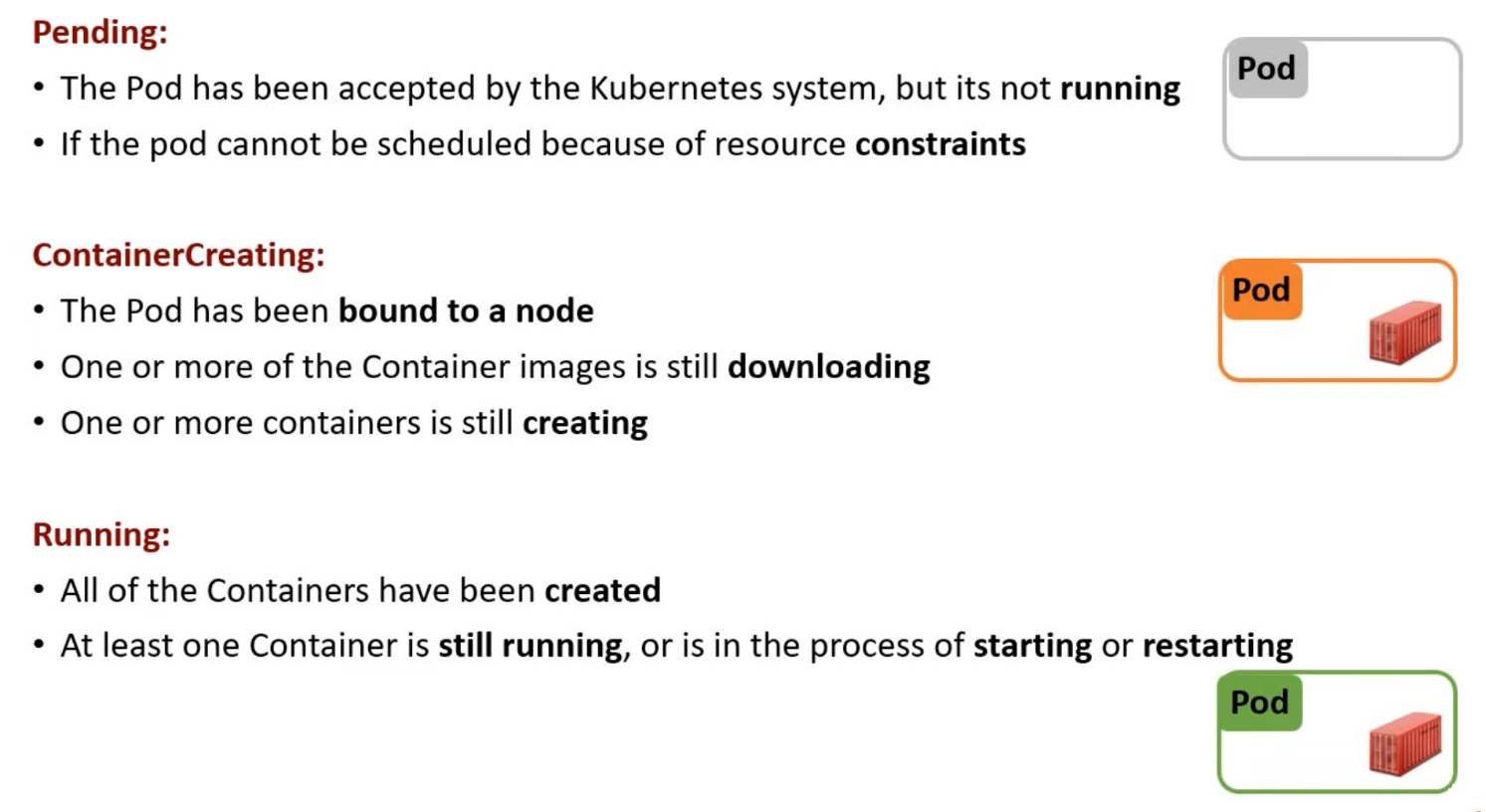
# POD Lifecycle

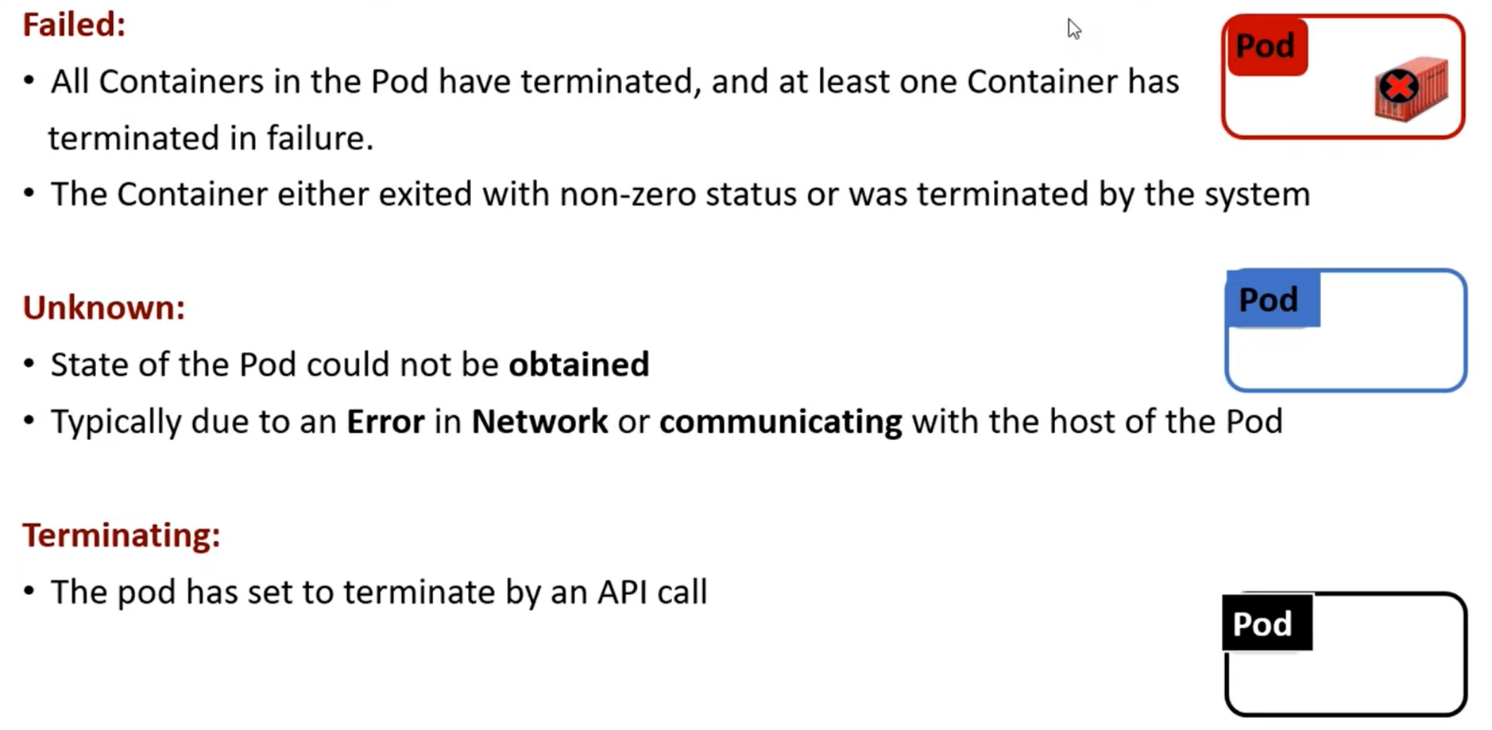
This page describes the lifecycle of a Pod. Pods follow a defined lifecycle, starting in the Pending [phase](https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#pod-phase), moving through Running if at least one of its primary containers starts OK, and then through either the Succeeded or Failed phases depending on whether any container in the Pod terminated in failure.



Here are the possible values for phase:

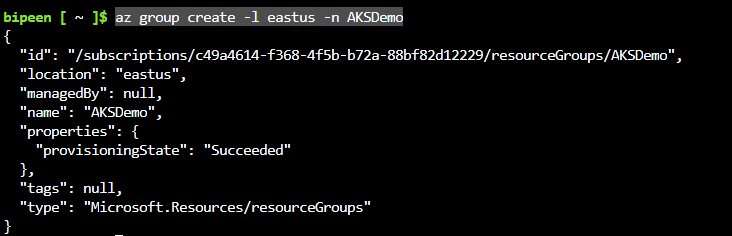
| **Value** | **Description** |
| --- | --- |
| Pending | The Pod has been accepted by the Kubernetes cluster, but one or more of the containers has not been set up and made ready to run. This includes time a Pod spends waiting to be scheduled as well as the time spent downloading container images over the network. |
| Running | The Pod has been bound to a node, and all of the containers have been created. At least one container is still running, or is in the process of starting or restarting. |
| Succeeded | All containers in the Pod have terminated in success, and will not be restarted. |
| Failed | All containers in the Pod have terminated, and at least one container has terminated in failure. That is, the container either exited with non-zero status or was terminated by the system. |
| Unknown | For some reason the state of the Pod could not be obtained. This phase typically occurs due to an error in communicating with the node where the Pod should be running. |





# Create 1 Node Azure Kubernetes Cluster

az group create -l eastus -n AKSDemo



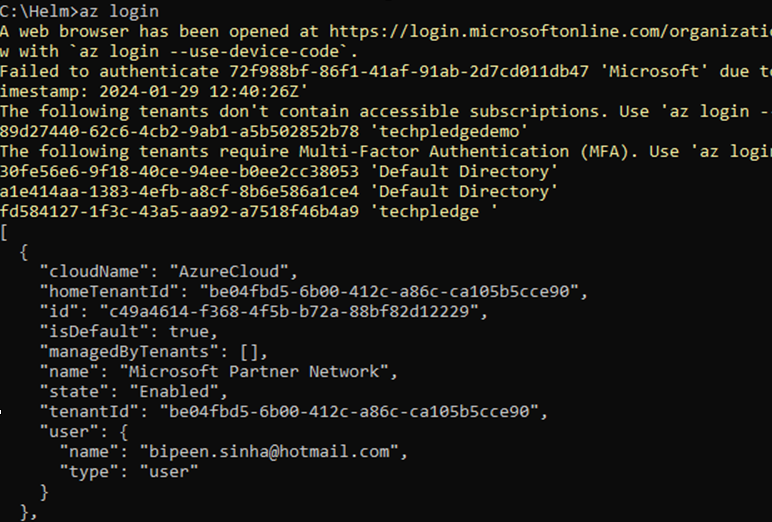
To create an AKS cluster, use the [az aks create](https://learn.microsoft.com/en-us/cli/azure/aks" \l "az-aks-create) command. The following example creates a cluster named BipeenClus01 with one node and generate SSH-key. It will automatically use 3 Nodes when you don’t specify node count

az aks create --resource-group AKSDemo --name BipeenClus01 --node-count 1 --generate-ssh-keys

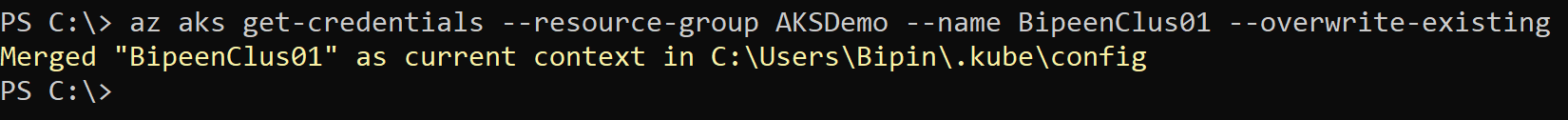


# Connect the Kubernetes Cluster from azure PowerShell

Az login



az aks get-credentials --resource-group AKSDemo --name BipeenClus01 --overwrite-existing



# Create a Deploy Kubernest Dashboard from helm chart

# Health Checks

We want to check if our pod services are healthy. Because processes can get into weird states, we want to make sure all of our containers in a pod are actually behaving normally. You get to define 'normally' in your pod manifest, and then that check will run against every container in your pod:

## Liveness Probe

"Is this container alive?"

### Insert example here, e.g.

apiVersion: v1

kind: Pod

metadata:

  name: kuard

spec:

  containers:

  - name: kuard

    image: gcr.io/kuar-demo/kuard-amd64:1

    ports:

    - containerPort: 8080

    livenessProbe:

      httpGet:

        path: /healthy

        port: 8080

      initialDelaySeconds: 5

      timeoutSeconds: 1

      periodSeconds: 10

      failureThreshold: 3

    readinessProbe:

      httpGet:

        path: /ready

        port: 8080

    protocol: TCP

kubectl apply -f kuard-pod-health.yaml

kubectl port-forward kuard 8080:8080