

Steps to create a Kubernetes solution in Microsoft Azure

Step Description

- 1 Prepare tools and Azure account
- 2 Log in to Azure
- 3 Create resource group
- 4 Create AKS cluster
- 5 Get credentials
- 6 Verify cluster status
- 7 Deploy app
- 8 Monitor and scale

Designed and deployed a Kubernetes solution on Microsoft Azure using AKS (Azure Kubernetes Service).

Logged into the Azure Portal and utilized Azure Cloud Shell (Bash) to create a resource group in the East US region. Provisioned an AKS cluster named *myAKSCluster* with 3 nodes and auto-generated SSH keys. Connected to the cluster using `kubectl`, deployed a containerized NGINX application exposed on port 80 with a LoadBalancer service type. Retrieved the external IP for public access and verified deployment health. Monitored cluster metrics and logs via Azure Insights to ensure performance and availability.

Kubernetes Solution Deployment on Microsoft Azure

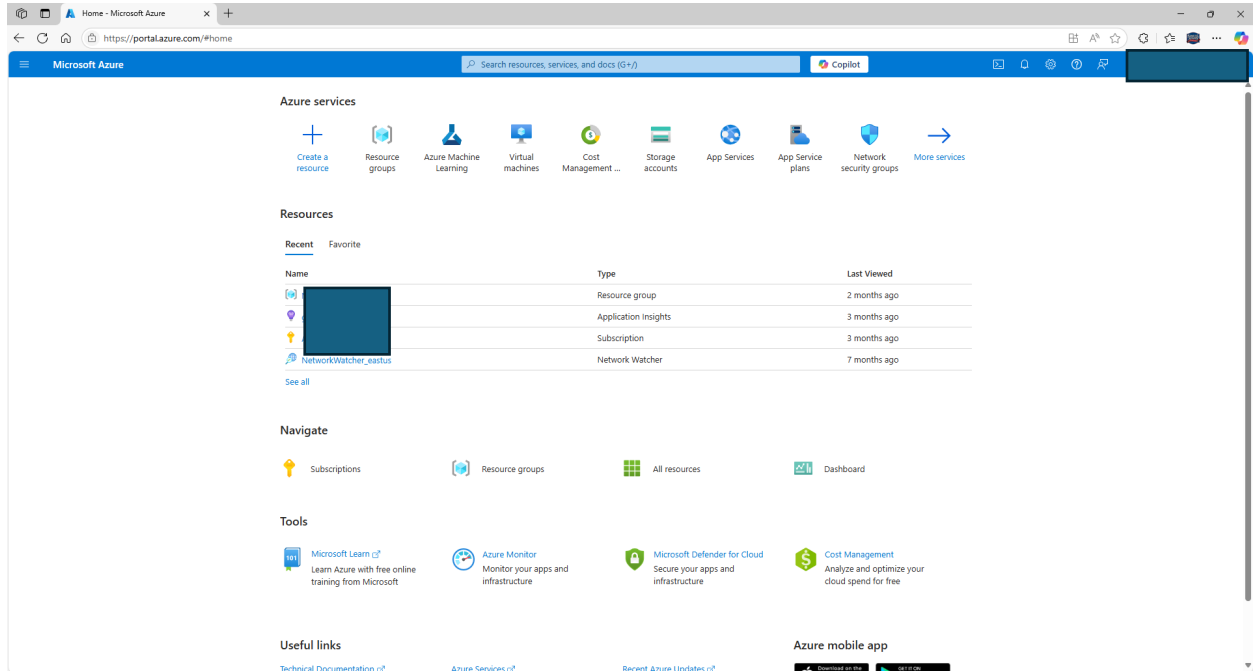
- **Problem Statement:** Deployed a scalable, containerized web application using Azure-managed Kubernetes infrastructure.
- **Challenges:** Set up secure access, managed cluster scaling, enabled public exposure of services, and configured monitoring for performance visibility.
- **Tools/Technologies:** Azure Portal, AKS, Azure Cloud Shell (Bash), `kubectl`, NGINX, Azure Monitor/Insights.
- **Outcomes:** Successfully provisioned a 3-node AKS cluster (*myAKSCluster*) with SSH access, deployed a LoadBalancer-exposed NGINX application, retrieved external IP for access, and validated performance through Azure Insights.

1. Log in to Azure

a. Username: joshphillis@hotmail.com

b. Password:

c. Microsoft Authenticator: 6 digit code



2. Create a Resource Group (Bash)

Select **Cloud Shell** icon

Enter: `az group create --name kubernetesjlp --location eastus`

Microsoft Azure

Search resources, services, and docs (G+/I)

Copilot

Azure services

- Create a resource
- Resource groups
- Azure Machine Learning
- Virtual machines
- Cost Management ...
- Storage accounts
- App Services
- App Service plans
- Network security groups
- More services

Resources

Recent Favorite

Name	Type	Last Viewed
NetworkWatcherRG	Resource group	2 months ago
gfunctionapp	Application Insights	3 months ago
Azure subscription 1	Subscription	3 months ago
NetworkWatcher_eastus	Network Watcher	7 months ago

Set all

Navigate

- Subscriptions
- Resource groups
- All resources
- Dashboard

Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help

Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Your Cloud Shell session will be ephemeral, so no files or system changes will persist beyond your current session.

joshua [~] \$ az group create --name kubernetesjlp --location eastus

https://portal.azure.com/#blade/HubsExtension/RecentResources.ReactView

Microsoft Azure

Search resources, services, and docs (G+/I)

Copilot

Home >

Resource groups | Simplified view

Create Manage view Refresh Export to CSV Assign tags

Filter for any field... Subscription equals all Resource Group equals all Location equals all Add filter

Name	Subscription
kubernetesjlp	Azure subscription 1

Showing 1 - 2 of 2 results.

Give feedback

Switch to PowerShell Restart Manage files New session Editor Web preview Settings Help

```
joshua [ ~ ] $ az group create --name kubernetesjlp --location eastus
{
  "id": "/subscriptions/8d3e1cd2-8996-4289-91db-4960cbdf9066/resourceGroups/kubernetesjlp",
  "location": "eastus",
  "managedBy": null,
  "name": "kubernetesjlp",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
joshua [ ~ ] $
```

Resource groups - Microsoft Azure

Search resources, services, and docs (G+)

Copilot

Home >

Resource groups

Default Directory (gotashillshotmail.onmicrosoft.com)

+ Create

Manage view

Refresh

Export to CSV

Open query

Assign tags

Group by none





You are viewing a new version of Browse experience. Some features may be missing. [Click here to access the old experience.](#)

Filter for any field...

Subscription equals all

Location equals all

+ Add filter

<input type="checkbox"/>	Name 1	Subscription	Location
<input type="checkbox"/>	 DefaultResourceGroup-EUS	... Azure subscription 1	East US
<input type="checkbox"/>	 kubernetesjlp	... Azure subscription 1	East US
<input type="checkbox"/>	 MC_kubernetesjlp_myAKSCluster_eastus	... Azure subscription 1	East US
<input type="checkbox"/>	 NetworkWatcherRG	... Azure subscription 1	East US

Showing 1 - 4 of 4. Display count:

10

Give feedback

3. Create the AKS Cluster (Bash) - At first, the script did not work. I had to adjust the naming.

Enter: az aks create \

--resource-group myResourceGroup \

--name myAKSCluster \

--node-count 3 \

--enable-addons monitoring \

--generate-ssh-keys

The screenshot displays the Microsoft Azure portal interface. The top navigation bar shows the 'kubernetesjlp' resource group. The left sidebar lists various services, including 'Activity log', 'Access control (IAM)', 'Tags', 'Resource visualizer', 'Events', 'Settings', 'Deployments', 'Security', 'Deployment stacks', 'Policies', 'Properties', 'Locks', and 'Cost Management'. The main content area shows the 'Resources' tab for the 'kubernetesjlp' resource group. It indicates 'Showing 0 to 0 of 0 records' and 'No resources match your filters'. Below the portal, a terminal window shows the command 'az aks create' with various flags, including '--resource-group kubernetesjlp', '--name myAKSCluster', '--node-count 3', '--enable-addons monitoring', and '--generate-ssh-keys'. The terminal output shows an error message about Log Analytics Workspace table availability.

```
az aks create \
--resource-group kubernetesjlp \
--name myAKSCluster \
--node-count 3 \
--enable-addons monitoring \
--generate-ssh-keys
```

The terminal output shows an error message: "put stream 'Microsoft-KubePodInventory' is not available for destination 'la-workspace'. Please ensure that the table exists in Log Analytics Workspace before creating or updating the rule." This message is repeated for several other tables, including 'Microsoft-KubePodInventory', 'Microsoft-KubePVInventory', 'Microsoft-KubeServices', and 'Microsoft-KubeInsightsMetrics'.

4. Connect to the Cluster

Enter: `az aks get-credentials --resource-group kubernetesjlp --name myAKSCluster`

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and a Copilot button. The main content area displays the 'kubernetesjlp' resource group. On the left, a sidebar lists various resource types. The 'Resources' tab is active, showing a table of resources. The table has columns for Name, Type, and Location. Two resources are listed: 'MSCI-eastus-myAKSCluster' (Data collection rule) and 'myAKSCluster' (Kubernetes service). Below the portal, a terminal window shows the command `az aks get-credentials --resource-group kubernetesjlp --name myAKSCluster` being entered.

Enter: `kubectl get nodes`

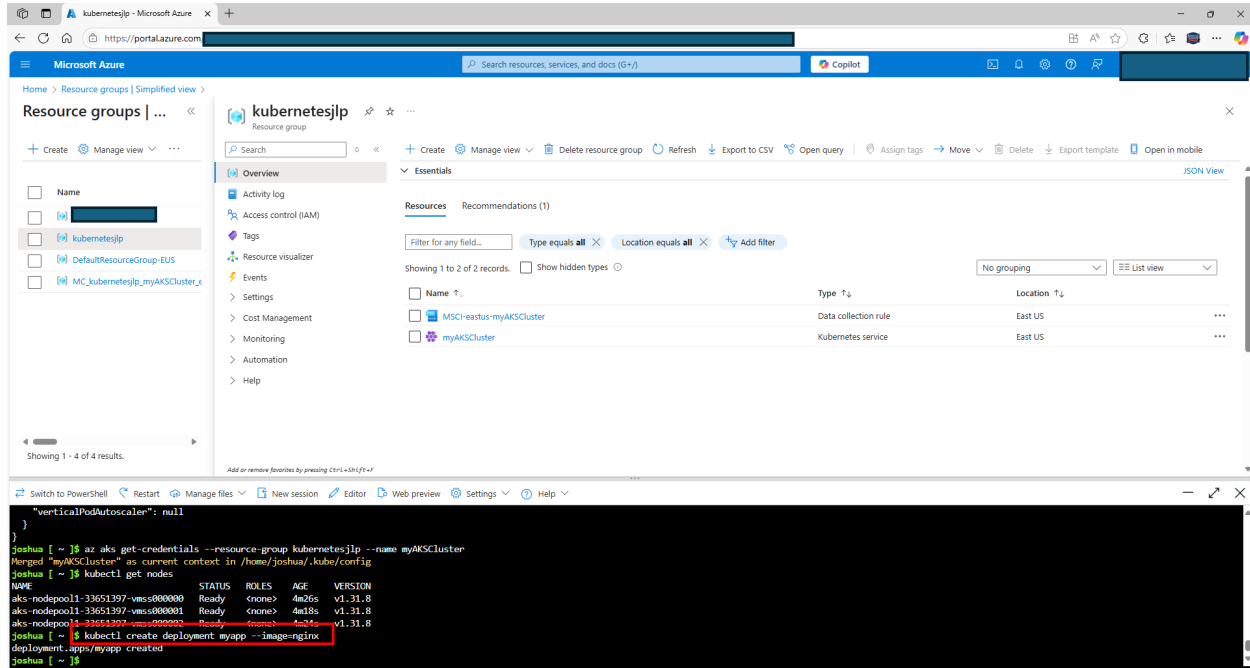
The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the Microsoft Azure logo, a search bar, and a Copilot button. The main content area displays the 'kubernetesjlp' resource group. On the left, a sidebar lists various resource types. The 'Resources' tab is active, showing a table of resources. The table has columns for Name, Type, and Location. Two resources are listed: 'MSCI-eastus-myAKSCluster' (Data collection rule) and 'myAKSCluster' (Kubernetes service). Below the portal, a terminal window shows the command `kubectl get nodes` being entered, followed by the output of the command:

```
joshua [ ~ ]$ az aks get-credentials --resource-group kubernetesjlp --name myAKSCluster
Merged "myAKSCluster" context into /home/joshua/.kube/config
joshua [ ~ ]$ kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
aks-nodepool1-33651397-vms8000000  Ready    <none>    4m26s  v1.31.8
aks-nodepool1-33651397-vms8000001  Ready    <none>    4m18s  v1.31.8
aks-nodepool1-33651397-vms8000002  Ready    <none>    4m24s  v1.31.8
```

5. Deploy Your Application

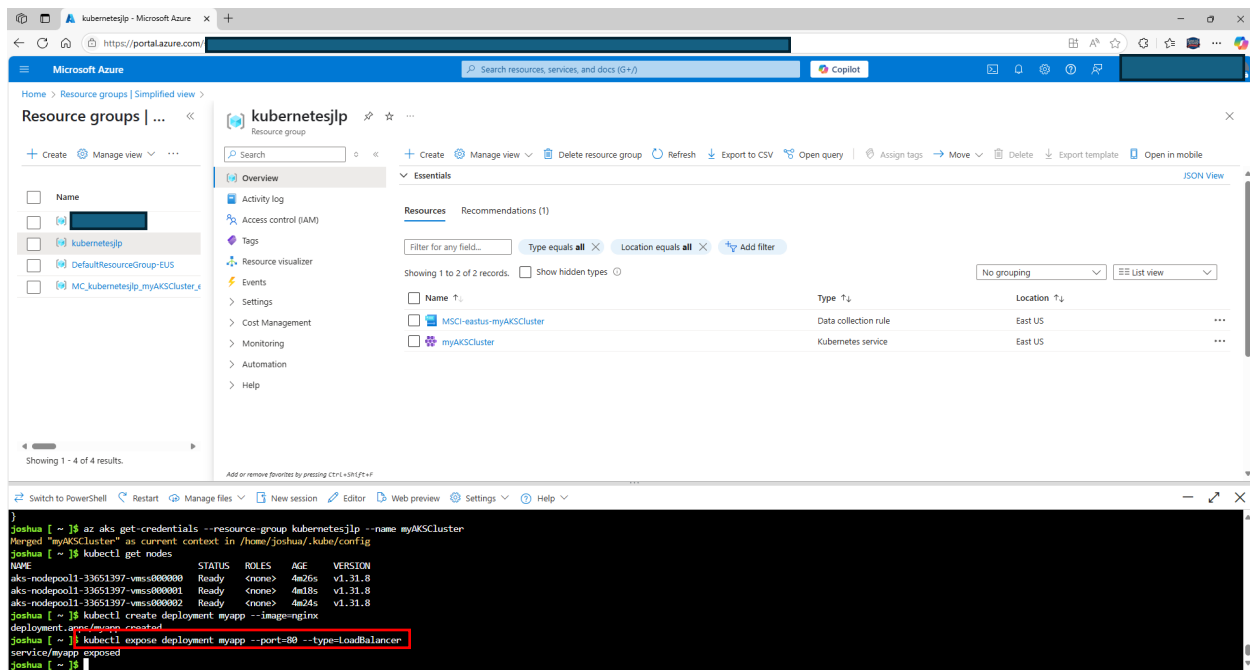
Enter: `kubectl create deployment myapp --image=nginx`

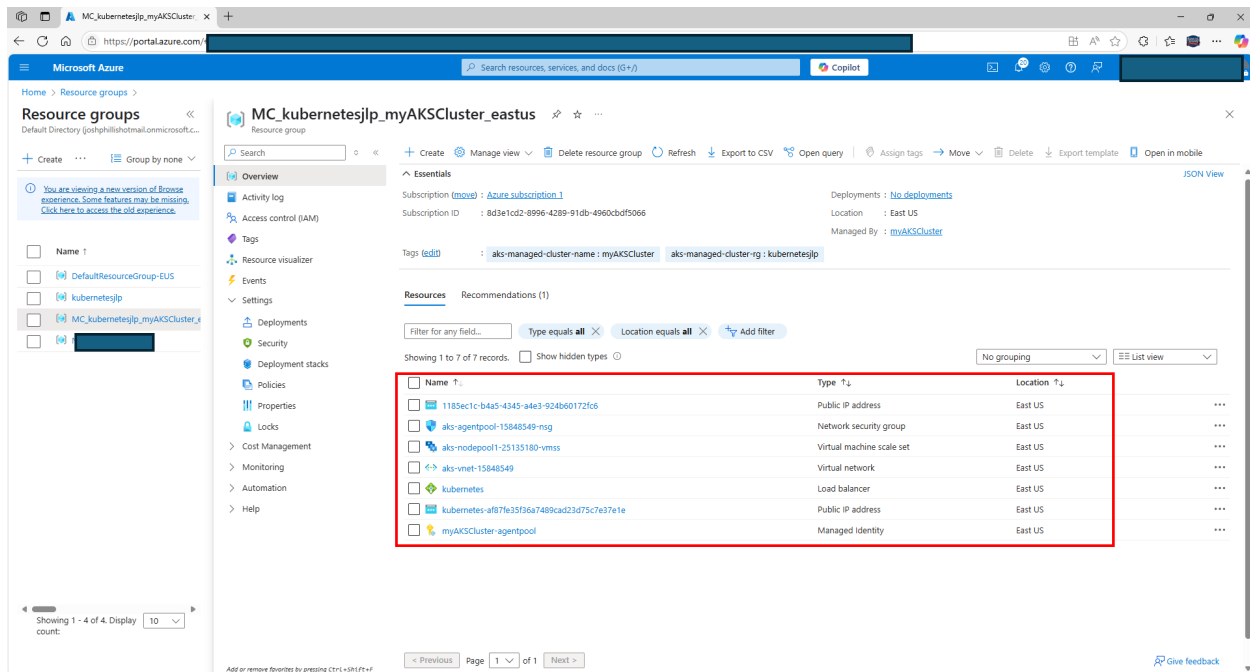
Output: **myapp** was created



Enter: `kubectl expose deployment myapp --port=80 --type=LoadBalancer`

Output: **myapp** exposed to **port 80** with a **Load Balancer**





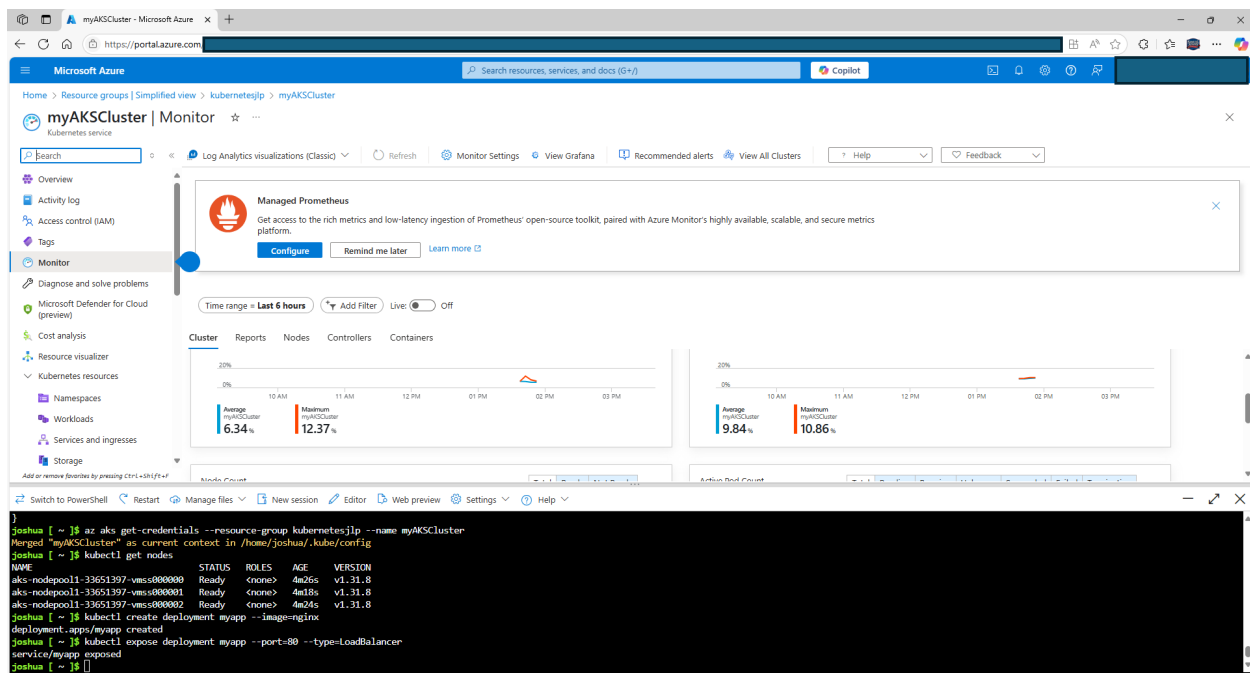
6. Monitor the Cluster

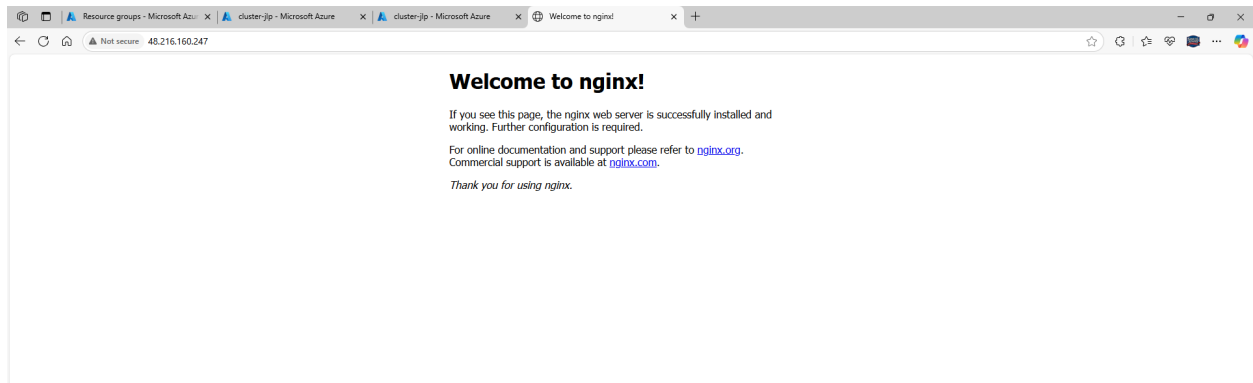
From portal, select **kubernetesjlp** Resource Group

Select **myAKSCluster**

Select **Monitor**

Select **Insights** to view Cluster metrics





Problem Statement:

The goal was to secure a Kubernetes application hosted on an Azure Kubernetes Service (AKS) cluster by enabling HTTPS to protect data in transit and enhance user trust.

Challenges:

While the AKS resource group and cluster were created via the Azure portal, automating TLS certificate issuance and renewal through Kubernetes required CLI interaction. Proper DNS setup and ingress configuration were necessary to route HTTPS traffic correctly. Certificate issuance could also fail if placeholder or invalid domains were used.

Tools & Techniques:

The NGINX Ingress Controller managed external access to the app. cert-manager integrated with Let's Encrypt automated certificate provisioning and renewal. DNS was configured to point the domain to the cluster's ingress IP. Most Kubernetes commands and resource management were performed through Bash using kubectl.

Outcomes:

The application became securely accessible over HTTPS with trusted certificates. Certificate lifecycle management was automated with minimal downtime. Traffic between clients and the ingress was encrypted, and a repeatable process for deploying TLS-secured applications in Kubernetes was established.