

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

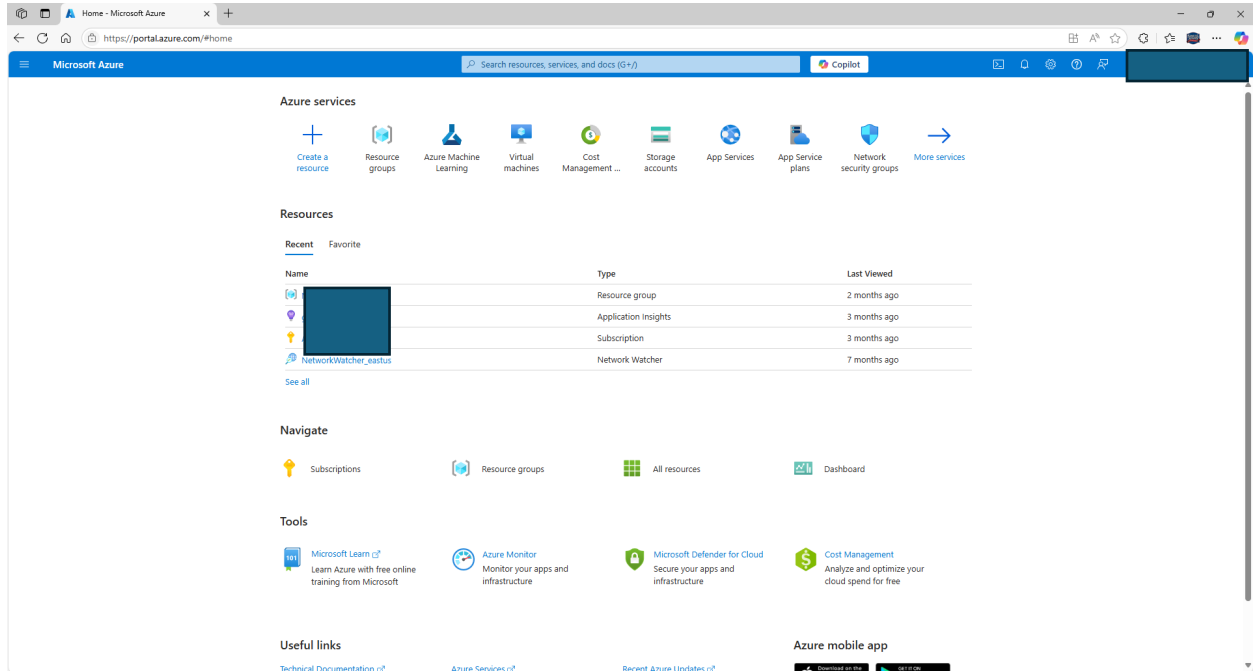
- **Problem Statement:** Deployed a scalable, containerized web application using Azure-managed Kubernetes infrastructure.
- **Challenges:** Set up secure access, managed cluster scaling, enabled public exposure of services, and configured monitoring for performance visibility.
- **Tools/Technologies:** 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`

The screenshot shows the Microsoft Azure portal interface. At the top, there's a search bar and a navigation menu. Below the navigation menu, there are sections for 'Azure services' (with icons for various services), 'Resources' (with a table of recent resources), and 'Navigate' (with links to Subscriptions, Resource groups, All resources, and Dashboard). The 'Resources' section contains a table with columns 'Name', 'Type', and 'Last Viewed'. The 'Navigate' section has a link to 'Resource groups'. In the bottom right corner, there's a 'Cloud Shell' icon, which is highlighted with a red box. Below the main content area, there's a terminal window titled 'Requesting a Cloud Shell.Succeeded. Connecting terminal...'. The terminal shows the command `az group create --name kubernetesjlp --location eastus` being entered, which is also highlighted with a red box.

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

The screenshot shows the 'Resource groups | Simplified view' page in the Microsoft Azure portal. The page has a header with 'Home > Resource groups | Simplified view'. Below the header, there's a 'Create' button and a 'Manage view' dropdown. There's also a 'Filter for any field...' section with filters for 'Subscription equals all', 'Resource Group equals all', and 'Location equals all'. The main content area shows a table with columns 'Name' and 'Subscription'. The table has two rows: one for 'kubernetesjlp' (highlighted with a red box) and another for 'kubernetesjlp' (with a red box around the 'kubernetesjlp' text). Below the table, it says 'Showing 1 - 2 of 2 results.' In the bottom right corner, there's a 'Give feedback' link. At the bottom of the page, there's a terminal window showing the command `az group create --name kubernetesjlp --location eastus` and its output, which is highlighted with a red box.

Name	Subscription
kubernetesjlp	...
kubernetesjlp	...

Resource groups - Microsoft Azure

Search resources, services, and docs (G+)

Copilot

Home >

Resource groups

Default Directory (gotphillshotmail.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	...	East US
<input type="checkbox"/>	kubernetesjlp	...	East US
<input type="checkbox"/>	MC_kubernetesjlp_myAKSCluster_eastus	...	East US
<input type="checkbox"/>	[REDACTED]	...	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 resource types, 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. A message states: '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 indicates the command is running.

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 and a search bar. 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.

Name	Type	Location
MSCI-eastus-myAKSCluster	Data collection rule	East US
myAKSCluster	Kubernetes service	East US

```
joshua [ ~ ] $ az aks get-credentials --resource-group kubernetesjlp --name myAKSCluster
```

Enter: `kubectl get nodes`

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the 'Microsoft Azure' logo and a search bar. 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.

Name	Type	Location
MSCI-eastus-myAKSCluster	Data collection rule	East US
myAKSCluster	Kubernetes service	East US

```
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

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the 'kubernetesjlp' resource group. The left sidebar shows a list of resource groups, including 'kubernetesjlp'. The main content area displays the 'Overview' tab for the 'kubernetesjlp' resource group. Below the 'Overview' tab, there is a table of resources. The table has columns for 'Name', 'Type', and 'Location'. The resources listed are 'MSCI-eastus-myAKSCluster' (Data collection rule, East US) and 'myAKSCluster' (Kubernetes service, East US). Below the table, there is a terminal window showing the following commands and output:

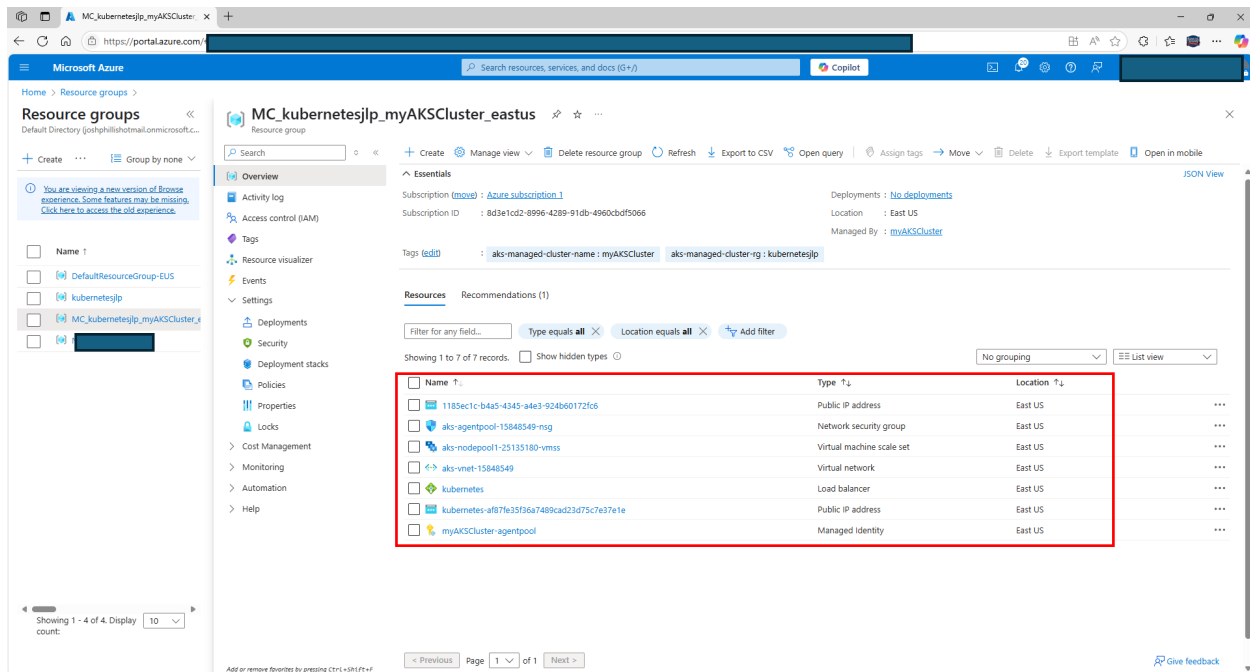
```
joshua [ ~ ] $ az aks get-credentials --resource-group kubernetesjlp --name myAKSCluster
Merged "myAKSCluster" as current context in /home/joshua/.kube/config
joshua [ ~ ] $ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
aks-nodepool11-33651397-vms000000   Ready    <none>   4m26s v1.31.8
aks-nodepool11-33651397-vms000001   Ready    <none>   4m18s v1.31.8
aks-nodepool11-33651397-vms000002   Ready    <none>   4m24s v1.31.8
joshua [ ~ ] $ kubectl create deployment myapp --image=nginx
deployment.apps/myapp created
joshua [ ~ ] $
```

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

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

The screenshot shows the Microsoft Azure portal interface. The top navigation bar includes the 'kubernetesjlp' resource group. The left sidebar shows a list of resource groups, including 'kubernetesjlp'. The main content area displays the 'Overview' tab for the 'kubernetesjlp' resource group. Below the 'Overview' tab, there is a table of resources. The table has columns for 'Name', 'Type', and 'Location'. The resources listed are 'MSCI-eastus-myAKSCluster' (Data collection rule, East US) and 'myAKSCluster' (Kubernetes service, East US). Below the table, there is a terminal window showing the following commands and output:

```
joshua [ ~ ] $ az aks get-credentials --resource-group kubernetesjlp --name myAKSCluster
Merged "myAKSCluster" as current context in /home/joshua/.kube/config
joshua [ ~ ] $ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
aks-nodepool11-33651397-vms000000   Ready    <none>   4m26s v1.31.8
aks-nodepool11-33651397-vms000001   Ready    <none>   4m18s v1.31.8
aks-nodepool11-33651397-vms000002   Ready    <none>   4m24s v1.31.8
joshua [ ~ ] $ kubectl create deployment myapp --image=nginx
deployment.apps/myapp created
joshua [ ~ ] $ kubectl expose deployment myapp --port=80 --type=LoadBalancer
service/myapp exposed
joshua [ ~ ] $
```



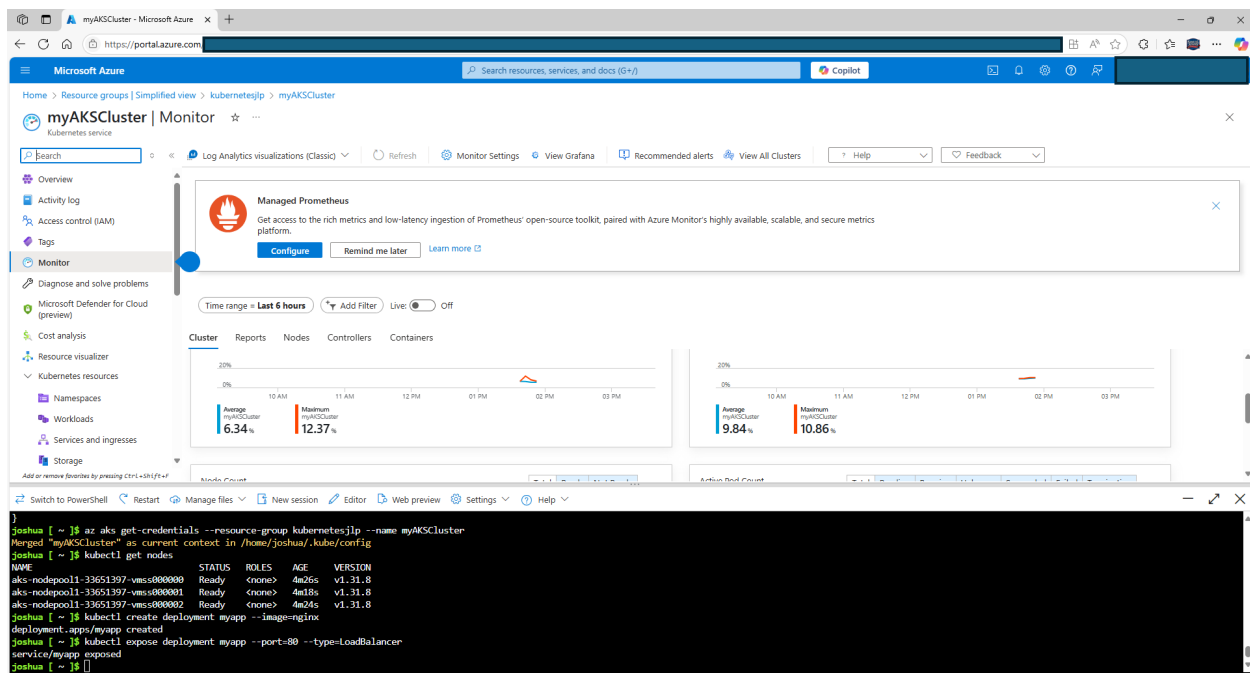
6. Monitor the Cluster

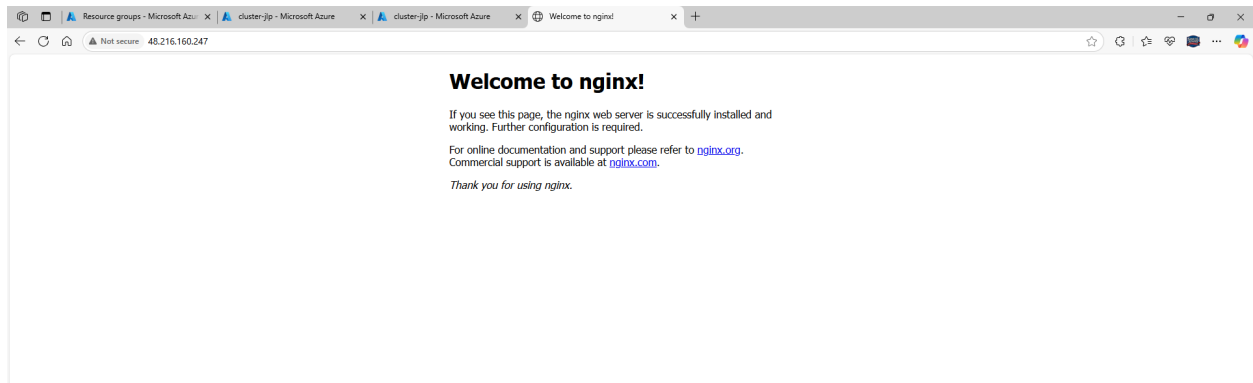
From portal, select **kubernetesjlp** Resource Group

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.