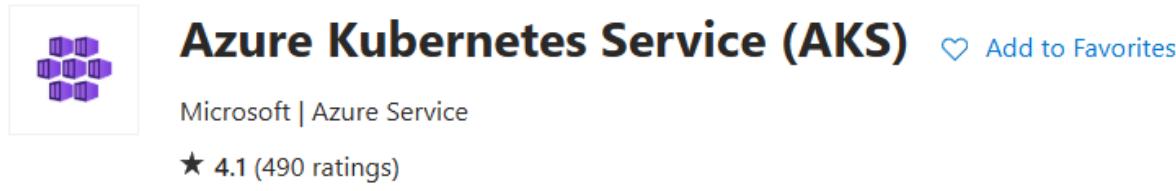


Azure Kubernetes

1. In this we will use Kubernetes to create a cluster and deploy our container onto that cluster.
2. In the marketplace search for Kubernetes and click on create.

Azure Kubernetes Service (AKS)

Microsoft



3. Choose your resource group, and give it a name and leave rest of the setting as they are.

Subscription * ⓘ MSDN Platforms Subscription

Resource group * ⓘ NewRG [Create new](#)

Cluster details

Cluster preset configuration * Dev/Test
To quickly customize your Kubernetes cluster, choose one of the preset configurations above. You can modify these configurations at any time.
[Compare presets](#)

Kubernetes cluster name * ⓘ appcluster121

Region * ⓘ (Europe) North Europe

Availability zones ⓘ None

AKS pricing tier ⓘ Free

4. In the integration section you need to integrate it with your container registry. Move to the review page and create your cluster.

Connect your AKS cluster with additional services.

Azure Container Registry

Connect your cluster to an Azure Container Registry to enable seamless deployments from a private image registry.

[Learn more ↗](#)

Container registry

appregistry1212



[Create new](#)

✓ Your deployment is complete



Deployment name: microsoft.aks-1741170495783
Subscription: [MSDN Platforms Subscription](#)
Resource group: [NewRG](#)

Start time: 5/3/2025, 3:59:21 pm

Correlation ID: 4c3ed9d9-1e2a-4872-ae62-b73547bcd250

✓ Deployment details

✗ Next steps

[Go to resource](#)

Now, instead of running your application and database in **Azure Container Instances (ACI)**, you want to deploy them onto this **Kubernetes cluster**.

To do this, you have **two YAML configuration files**:

1. Application container deployment

- This file defines a **Kubernetes pod** that runs your application.
- It specifies the **container name** and uses the **latest version** of your image.
- The image is pulled from **Azure Container Registry (ACR)**.
- A **load balancer service** is also deployed to expose the application on **port 80**.

```
✓ app.yml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: appservice
5  spec:
6    selector:
7      app: myapp
8    ports:
9      - protocol: TCP
10     port: 80
11     targetPort: 80
12   type: LoadBalancer
13 ---
14  apiVersion: apps/v1
15  kind: Deployment
16  metadata:
17    name: myapp
18  spec:
19    selector:
20      matchLabels:
21        app: myapp
22    template:
23      metadata:
24        labels:
25          app: myapp
26    spec:
27      containers:
```

2. Database container deployment

- Similar to the application deployment, this YAML file defines a **pod** for the **MySQL database**.
- It references the **database container image** (e.g., appsqlimage).
- Again, a **load balancer service** is created to expose the database publicly.
- However, exposing a **database to the internet** is generally **not recommended**, but it's being done here for demonstration purposes to show how data can be accessed from the application.

```

Y mysql.yml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: mysql
5  spec:
6    selector:
7      app: mysql
8    ports:
9      - protocol: TCP
10     port: 3306
11     targetPort: 3306
12     type: LoadBalancer
13   ---
14  apiVersion: apps/v1
15  kind: Deployment
16  metadata:
17    name: mysql
18  spec:
19    selector:
20      matchLabels:
21        app: mysql
22    strategy:
23      type: Recreate
24    template:
25      metadata:
26        labels:
27          app: mysql

```

5. Now the web application that we have to display the data from our database. We are going to publish it again and reupload it to the VM. But this time we need to change the server's name to MySQL as you can see in the snapshot.
6. Now just create a publish profile and upload it back to the VM. Also, delete the previous version of publish profile from the VM.

```

1   using MySql.Data.MySqlClient;
2   using sqlapp.Models;
3
4
5   namespace sqlapp.Services
6   {
7
8       public class ActivityService
9       {
10
11           private MySqlConnection _connection
12           {
13               get;
14               set;
15           }
16
17           public List<Activity> GetActivities()
18           {
19               List<Activity> _activity_lst = new List<Activity>();
20               string _statement = "SELECT Id,Operationname,Status,Eventcategory,Resourcetype,Resource from log";
21               MySqlConnection _connection = GetConnection();

```

- Once you have copied the publish profile and the docker file inside the publish profile we need to go to the terminal.

C:\tmp2\				/home/linuxadmin/publish/				
Name	Size	Type	Changed	Name	Size	Changed	Rights	Owner
..		Parent directory	05-03-2025 16:13:34	..		05-03-2025 16:14:38	rwxr-x---	linuxad...
publish		File folder	05-03-2025 16:13:34	runtimes		05-03-2025 16:14:54	rwxrwxr-x	linuxad...
Dockerfile	1 KB	File	05-03-2025 15:09:01	wwwroot		05-03-2025 16:15:53	rwxrwxr-x	linuxad...
				appsettings.Develop...	1 KB	05-03-2025 10:46:38	rw-rw-r--	linuxad...
				appsettings.json	1 KB	05-03-2025 10:46:38	rw-rw-r--	linuxad...
				BouncyCastle.Crypto....	3,241 KB	19-10-2021 17:23:34	rw-rw-r--	linuxad...
				Dockerfile	1 KB	05-03-2025 15:09:01	rw-rw-r--	linuxad...
				Google.Protobuf.dll	400 KB	26-10-2022 19:16:18	rw-rw-r--	linuxad...
				K4os.Compression.LZ...	66 KB	06-01-2023 22:46:12	rw-rw-r--	linuxad...
				K4os.Compression.LZ...	79 KB	06-01-2023 22:46:12	rw-rw-r--	linuxad...
				K4os.Hash.xxHash.dll	13 KB	08-11-2022 23:38:14	rw-rw-r--	linuxad...
				Microsoft.Win32.Syst...	23 KB	15-11-2019 14:06:56	rw-rw-r--	linuxad...
				MySql.Data.dll	1,167 KB	13-04-2023 06:01:20	rw-rw-r--	linuxad...
				sqlapp.deps.json	21 KB	05-03-2025 16:13:20	rw-rw-r--	linuxad...
				sqlapp.dll	46 KB	05-03-2025 16:13:19	rw-rw-r--	linuxad...
				sqlapp.exe	148 KB	05-03-2025 16:13:19	rw-rw-r--	linuxad...
				sqlapp.pdb	34 KB	05-03-2025 16:13:19	rw-rw-r--	linuxad...
				sqlapp.runtimeconfig...	1 KB	05-03-2025 15:13:25	rw-rw-r--	linuxad...
				sqlapp.staticwebasset...	129 KB	05-03-2025 15:13:26	rw-rw-r--	linuxad...
				System.Configuration...	372 KB	20-11-2017 23:39:04	rw-rw-r--	linuxad...
				System.Drawing.Com...	141 KB	19-09-2018 01:08:14	rw-rw-r--	linuxad...
				System.IO.Pipelines.dll	78 KB	13-04-2022 23:19:36	rw-rw-r--	linuxad...
				System.Security.Crypt...	25 KB	19-07-2017 15:31:34	rw-rw-r--	linuxad...
				System.Security.Perm...	91 KR	15-11-2019 14:26:44	rw-rw-r--	linuxad...

- I have also deleted the sqlapp repository because we are going to create a new one again.

The screenshot shows the Azure Container Registry (ACR) interface. On the left, there's a sidebar with various options: Networking, Properties, Locks, Services (with 'Repositories' selected), Webhooks, and Geo-replications. In the main area, there's a message bar that says 'New to ACR, Artifact streaming helps pull images faster'. Below that is a search bar with the placeholder 'Search to filter repositories ...'. Underneath the search bar, the word 'Repositories' is followed by an up-and-down arrow icon. Then, the repository 'appsqlimage' is listed.

9. Here you can see that we have removed sqlapp registry from the docker.

```
linuxadmin@VM1:~/publish$ sudo docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
sqlapp              latest   623937800d78  55 minutes ago  226MB
appregistry1212.azurecr.io/sqlapp    latest   623937800d78  55 minutes ago  226MB
appregistry1212.azurecr.io/appsqlimage  latest   3757f909f7bc  4 hours ago   797MB
appsqlimage         latest   3757f909f7bc  4 hours ago   797MB
linuxadmin@VM1:~/publish$ sudo docker image rmi 623937 --force
Untagged: appregistry1212.azurecr.io/sqlapp:latest
Untagged: appregistry1212.azurecr.io/sqlapp@sha256:1deaf1dc9917e7394a844cb9bb6d599f620054940d03ee4016e8b7f59fb8c323
Untagged: sqlapp:latest
Deleted: sha256:623937800d78608be013b14589fc285065c58a7bf6d039b2f936d6876a62cb77
linuxadmin@VM1:~/publish$ sudo docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
appregistry1212.azurecr.io/appsqlimage  latest   3757f909f7bc  4 hours ago   797MB
appsqlimage         latest   3757f909f7bc  4 hours ago   797MB
linuxadmin@VM1:~/publish$
```

10. Now we are going to use the commands to build and publish our image.

```

linuxadmin@VM1:~/publish$ sudo docker build -t sqlapp .
[+] Building 1.6s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 150B
=> [internal] load metadata for mcr.microsoft.com/dotnet/aspnet:6.0
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/3] FROM mcr.microsoft.com/dotnet/aspnet:6.0@sha256:e70c493f8af7f95bf459cb2b15c7e7a6173228929c2b7a9a6836b19377890e78
=> [internal] load build context
=> => transferring context: 260.86kB
=> CACHED [2/3] WORKDIR /app
=> [3/3] COPY . .
=> exporting to image
=> => exporting layers
=> => writing image sha256:fb68165757b2aee3101d8e7159f3fe2ff539cc7399679293f45997b6d75a2916
=> => naming to docker.io/library/sqlapp
linuxadmin@VM1:~/publish$ sudo docker login appregistry1212.azurecr.io -u appregistry1212 -p plvAXQngcXXH60SmTtKNXmVxr+7EY/WsJNaox4QkK+A
CRBDCzZ3
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
Login Succeeded
linuxadmin@VM1:~/publish$ sudo docker tag sqlapp appregistry1212.azurecr.io/sqlapp
linuxadmin@VM1:~/publish$ sudo docker push appregistry1212.azurecr.io/sqlapp
Using default tag: latest
The push refers to repository [appregistry1212.azurecr.io/sqlapp]
5bf8a0575be7: Pushing [=====] 17.86MB
70f3da3aba93: Pushing 1.536kB
7f7140d2a118: Pushing [=====] 2.701MB/20.34MB
59361b3806d7: Pushing 2.56kB
a190071c1c72: Pushing [=====] 17.02MB/70.72MB

```

11. Then in the portal open the Kubernetes cluster and open the workloads. Then on create and choose Apply a YAML.

Name	Namespace	Ready	Age	CPU	Memory
coredns	kube-system	✓ 2/2	20 minutes	0%	4%
coredns-autoscaler	kube-system	✓ 1/1	20 minutes	0%	2%
konnektivity-agent	kube-system	✓ 2/2	20 minutes	0%	1%
metrics-server	kube-system	✓ 2/2	20 minutes	0%	8%
azure-wi-webhook-controller-manager	kube-system	✓ 2/2	18 minutes	2%	4%
eraser-controller-manager	kube-system	✓ 1/1	18 minutes	0%	3%
ama-metrics	kube-system	✓ 2/2	15 minutes	0.07%	1%
ama-metrics-ksm	kube-system	✓ 1/1	15 minutes	0%	0.34%
ama-metrics-operator-targets	kube-system	✓ 1/1	15 minutes	0%	0.85%

12. Go onto MySQL first and copy the code from line number 14, click on add

YAML **JSON**

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: mysql
5  spec:
6    selector:
7      matchLabels:
8        app: mysql
9    strategy:
10      type: Recreate
11    template:
12      metadata:
13        labels:
14          app: mysql
15      spec:
16        containers:
17          - image: appregistry1212.azurecr.io/appsqlimage:latest
18            name: mysql
19            ports:
20              - containerPort: 3306
21            name: mysql
```

13. Here you can see that MySQL deployment has been completed.

Deployments						
	Name	Namespace	Ready	Age	CPU	Memory
<input type="checkbox"/>	coredns	kube-system	✓ 2/2	24 minutes	0% 	4% 
<input type="checkbox"/>	coredns-autoscaler	kube-system	✓ 1/1	24 minutes	0% 	2% 
<input type="checkbox"/>	connectivity-agent	kube-system	✓ 2/2	24 minutes	0% 	1% 
<input type="checkbox"/>	metrics-server	kube-system	✓ 2/2	24 minutes	0% 	8% 
<input type="checkbox"/>	azure-wi-webhook-controller-manager	kube-system	✓ 2/2	22 minutes	2% 	4% 
<input type="checkbox"/>	eraser-controller-manager	kube-system	✓ 1/1	22 minutes	0% 	3% 
<input type="checkbox"/>	ama-metrics	kube-system	✓ 2/2	19 minutes	0.07% 	1% 
<input type="checkbox"/>	ama-metrics-ksm	kube-system	✓ 1/1	19 minutes	0% 	0.34% 
<input type="checkbox"/>	ama-metrics-operator-targets	kube-system	✓ 1/1	19 minutes	0% 	0.85% 
<input type="checkbox"/>	mysql	default	✓ 1/1	14 seconds	-	-

14. Then come to services and ingresses choose apply a YAML here as well then copy the code from line 1-12 from the MySQL YAML file.

appcluster121 | Services and ingresses ⭐ ...

Kubernetes service

Search Create Delete Refresh Show labels Give feedback

- Monitor
- Diagnose and solve problems
- Microsoft Defender for Cloud (preview)
- Cost analysis
- Resource visualizer
- Kubernetes resources
 - Namespaces
 - Workloads
 - Services and ingresses**
 - Storage
 - Configuration
 - Custom resources
 - Events
 - Run command

Service
Create a service to expose an application running on a set of pods.

Ingress
Create an Ingress to route traffic to a Service with the Application Routing add-on.

Starter application
Create a basic web or single-image application.

Apply a YAML
A blank box where a YAML definition can be specified to deploy to the cluster.

Namespace	Status
default	Ok
kube-system	Ok
kube-system	Ok
kube-system	Ok
azure-wi-webhook-webhook-service	Ok
network-observability	Ok
ama-metrics-ksm	Ok
ama-metrics-operator-targets	Ok

YAML **JSON**

```

1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: mysql
5  spec:
6    selector:
7      app: mysql
8    ports:
9      - protocol: TCP
10     port: 3306
11     targetPort: 3306
12   type: LoadBalancer

```

15. Here you can see that you have MySQL running as your service. You also get the external IP to connect with it.

Services Ingresses

Filter by namespace

<input type="checkbox"/>	Name	Namespace	Status	Type	Cluster IP	External IP	Ports	Age ↓
<input type="checkbox"/>	kubernetes	default	✓ Ok	ClusterIP	10.0.0.1		443/TCP	27 minutes
<input type="checkbox"/>	kube-dns	kube-system	✓ Ok	ClusterIP	10.0.0.10		53/UDP,53/TCP	26 minutes
<input type="checkbox"/>	metrics-server	kube-system	✓ Ok	ClusterIP	10.0.101.217		443/TCP	26 minutes
<input type="checkbox"/>	azure-wi-webhook-webhook-service	kube-system	✓ Ok	ClusterIP	10.0.166.217		443/TCP	25 minutes
<input type="checkbox"/>	network-observability	kube-system	✓ Ok	ClusterIP	10.0.111.31		10093/TCP	21 minutes
<input type="checkbox"/>	ama-metrics-ksm	kube-system	✓ Ok	ClusterIP	10.0.206.49		8080/TCP	21 minutes
<input type="checkbox"/>	ama-metrics-operator-targets	kube-system	✓ Ok	ClusterIP	10.0.143.237		80/TCP	21 minutes
<input type="checkbox"/>	mysql	default	✓ Ok	LoadBalancer	10.0.86.26	4.208.4.17	3306:31287/TCP	1 minute

16. You need to do the same thing for the other file which is app.yml below are the snapshots.

YAML **JSON**

```

1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: myapp
5  spec:
6    selector:
7      matchLabels:
8        app: myapp
9    template:
10      metadata:
11        labels:
12          app: myapp
13      spec:
14        containers:
15          - image: appregistry1212.azurecr.io/sqlapp:latest
16            name: myapp
17            ports:
18              - containerPort: 80
19
```

YAML **JSON**

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: appservice
5  spec:
6    selector:
7      app: myapp
8    ports:
9      - protocol: TCP
10     port: 80
11     targetPort: 80
12   type: LoadBalancer
```

17. Here you can see that you have the external IP for both the services.

Services Ingresses

Filter by namespace
All namespaces Service name : All Add label filter

<input type="checkbox"/>	Name	Namespace	Status	Type	Cluster IP	External IP	Ports	Age ↓	...
<input type="checkbox"/>	kubernetes	default	Ok	ClusterIP	10.0.0.1		443/TCP	30 minutes	...
<input type="checkbox"/>	kube-dns	kube-system	Ok	ClusterIP	10.0.0.10		53/UDP,53/TCP	29 minutes	...
<input type="checkbox"/>	metrics-server	kube-system	Ok	ClusterIP	10.0.101.217		443/TCP	29 minutes	...
<input type="checkbox"/>	azure-wi-webhook-webhook-service	kube-system	Ok	ClusterIP	10.0.166.217		443/TCP	28 minutes	...
<input type="checkbox"/>	network-observability	kube-system	Ok	ClusterIP	10.0.111.31		10093/TCP	24 minutes	...
<input type="checkbox"/>	ama-metrics-ksm	kube-system	Ok	ClusterIP	10.0.206.49		8080/TCP	24 minutes	...
<input type="checkbox"/>	ama-metrics-operator-targets	kube-system	Ok	ClusterIP	10.0.143.237		80/TCP	24 minutes	...
<input type="checkbox"/>	mysql	default	Ok	LoadBalancer	10.0.86.26	4.208.4.17	3306:31287/TCP	4 minutes	...
<input type="checkbox"/>	appservice	default	Ok	LoadBalancer	10.0.147.157	172.205.93.113	80:30536/TCP	36 seconds	...

18. Here is the home page for Sql app which has no data record. We will be using the same application to load the data inside of it. We need to copy the external IP of mysql from the services.

△ Not secure 172.205.93.113

sqlapp Home Privacy

Information in the Activity Log

ID	Operation name	Status	Event category	Resource type	Resource
----	----------------	--------	----------------	---------------	----------

Not secure 172.205.93.113

sqlapp Home Privacy

Information in the Activity Log

Id	Operation name	Status	Event category	Resource type	Resource
1	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update SQL database	Succeeded	Administrative	Informational
2	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Create Deployment	Started	Administrative	Informational
3	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Create Deployment	Accepted	Administrative	Informational
4	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Registers the Microsoft SQL Database Resource Provider	Started	Administrative	Informational
5	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Registers the Microsoft SQL Database Resource Provider	Succeeded	Administrative	Informational
6	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update SQL server	Started	Administrative	Informational
7	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	'audit' Policy action.	Succeeded	Policy	Warning
8	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	'auditIfNotExist' Policy action.	Started	Policy	Informational
9	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update SQL server	Accepted	Administrative	Informational
10	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update SQL server firewall rules	Started	Administrative	Informational
11	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update SQL server firewall rules	Started	Administrative	Informational
12	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update Server Connection Policy Create	Started	Administrative	Informational
13	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update SQL database	Started	Administrative	Informational
14	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update Server Connection Policy Create	Succeeded	Administrative	Informational
15	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update SQL server	Succeeded	Administrative	Informational
16	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	'auditIfNotExist' Policy action.	Started	Policy	Informational
17	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update SQL database	Accepted	Administrative	Informational
18	99fe9c3a-e36e-44e0-acd4-58272ab10c7e	Update SQL server firewall rules	Succeeded	Administrative	Informational

19. Once you are done with the lab just delete all of the resources.