

AI-02077 Docker Desktop

Owner	© Rigel Arcayan
Tags	Project
Created time	@September 6, 2025 10:35 PM

Create a step by step instructions on how to use Docker Desktop in Azure

Here's a **step-by-step guide** to using **Docker Desktop with Azure**, especially useful if you're just getting started with containerized apps and cloud deployment.

Step-by-Step: Using Docker Desktop with Azure

Step 1: Install Docker Desktop

1. Go to Docker Desktop download page.
2. Download the installer for your OS (Windows/macOS).
3. Run the installer and follow the setup instructions.
4. After installation, launch Docker Desktop and ensure it's running.

Step 2: Enable Docker Compose

Docker Desktop includes Docker Compose by default. You can verify it by running:

```
docker compose version
```

Step 3: Install Azure CLI

1. Visit Azure CLI installation page.
2. Download and install the CLI for your OS.
3. After installation, verify with:

```
az version
```

Step 4: Log in to Azure

Use the CLI to log in:

```
az login
```

This opens a browser window for authentication. Once logged in, your terminal will show your subscription details.

Step 5: Create a Resource Group and Azure Container Registry (ACR)

```
az group create --name myResourceGroup --location southeastasia
```

```
az acr create --resource-group myResourceGroup \  
  --name myContainerRegistry \  
  --sku Basic
```

Replace `myContainerRegistry` with a unique name.

Step 6: Log in to ACR from Docker

```
az acr login --name myContainerRegistry
```

Or use Docker directly:

```
docker login mycontainerregistry.azurecr.io
```

Step 7: Build and Tag Your Docker Image

Navigate to your project folder with a `Dockerfile`, then run:

```
docker build -t mycontainerregistry.azurecr.io/myapp:latest .
```

Step 8: Push Image to Azure Container Registry

```
docker push mycontainerregistry.azurecr.io/myapp:latest
```

Step 9: Deploy to Azure (Optional)

You can deploy your container to Azure services like:

- **Azure Container Instances (ACI):**

```
az container create \  
  --resource-group myResourceGroup \  
  --name myapp \  
  --image mycontainerregistry.azurecr.io/myapp:latest \  
  --registry-login-server mycontainerregistry.azurecr.io \  
  --registry-username &lt;username> \  
  --registry-password &lt;password> \  
  --dns-name-label myappdns \  
  --ports 80
```

- **Azure App Service for Containers**
- **Azure Kubernetes Service (AKS)**

Optional: Use Docker Compose Locally

Create a `docker-compose.yml` file:

```
version: '3'  
services:  
  web:  
    image: mycontainerregistry.azurecr.io/myapp:latest
```

```
ports:  
- "80:80"
```

Run it with:

```
docker compose up
```

Would you like help creating a sample Dockerfile or Compose file for a web app?
Or would you prefer to deploy to Azure App Service instead of ACI?