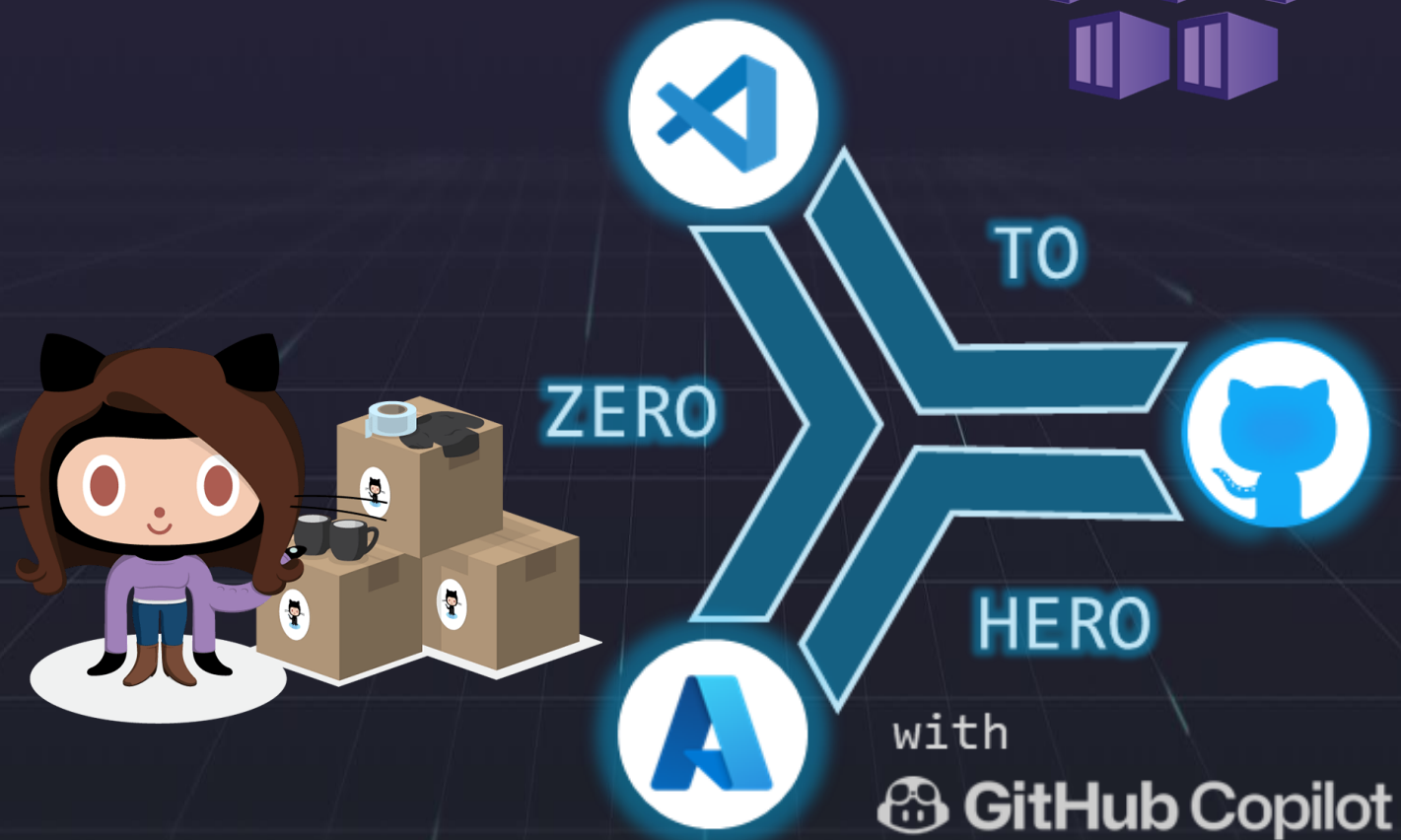


# Docker & Kubernetes



[aka.ms/zero2hero](https://aka.ms/zero2hero)

Please note that the views expressed in this doc are just my own



[github.com/vieiraae](https://github.com/vieiraae)



[linkedin.com/in/vieira](https://linkedin.com/in/vieira)

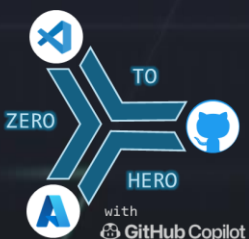


# HERO Goals

1. Create a custom NGINX container
2. Create a container registry and Kubernetes cluster
3. Build the container
4. Create k8s manifest
5. Deploy the container



Docker &  
Kubernetes





# GitHub Copilot Superpowers

- ✓ Execute complex tasks with simple prompts to your personal assistant
- ✓ Context aware environment to keep you in the flow
- ✓ Natural language powers your coding experience
- ✓ Proven impact on your productivity & happiness:

**74%** focus on more satisfying work

**88%** feel more productive

**96%** are faster with repetitive tasks



<https://github.com/features/copilot>

# 0

# Environment Setup



- ☐ Activate GitHub Copilot
- ☐ Join the waitlist (if needed)



- ☐ Install Visual Studio Code
- ☐ Add the extensions
  - ☐ GitHub Copilot
  - ☐ GitHub Copilot Chat
  - ☐ Azure Tools



- ☐ Install Azure CLI (or use the Azure Cloud Shell)

# 1

# Create the Container

❑ Type in the chat:



*provide the steps and code for a nginx docker container with a custom configuration file pointing to a welcome HTML page that displays "welcome to my HERO web server" in a HeroContainer folder*



GitHub Copilot



Here are the steps to create an Nginx Docker container with a custom configuration file pointing to a welcome HTML page that displays "welcome to my HERO web server" in a HeroContainer folder:

1. Create a new directory for your project and navigate into it:

```
mkdir HeroContainer
cd HeroContainer
```

2. Create a new file called **Dockerfile** and add the following code:

```
FROM nginx
COPY nginx.conf /etc/nginx/conf.d/default.conf
COPY index.html /usr/share/nginx/html/HeroContainer/index.html
```

3. Create a new file called **nginx.conf** and add the following code:

```
server {
    listen 80;
    server_name localhost;
    root /usr/share/nginx/html/HeroContainer;
    index index.html;
```

- ❑ Go to "File -> Open Folder " to open the HeroContainer
- ❑ Create the Dockerfile, nginx.conf and index.html files with the generated content

# 2

## Create Azure Resources

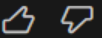
❑ Type in the chat:



*use the terminal to create an Azure Resource Group named hero-container-rg located on west europe with a new Azure Kubernetes Service named heroaks attached to a new Azure Container Registry named heroacr1*



GitHub Copilot



Sure! Here's the command to create an Azure Resource Group named **hero-container-rg** located in West Europe:

```
az group create --name hero-container-rg --location westeurope
```

And here's the command to create an Azure Container Registry named **heroacr1**:

```
az acr create --resource-group hero-container-rg --name heroacr1 --sku Basic
```

Finally, here's the command to create an Azure Kubernetes Service named **heroaks** attached to the **heroacr1** container registry:

```
az aks create --resource-group hero-container-rg --name heroaks --node-count 1
```

This will create a new AKS cluster with one node and attach it to the **heroacr1** container registry. You can now deploy your containerized application to the AKS cluster using the **kubect1** command-line tool.

❑ Run the commands in the terminal (bash or PowerShell)  
⚠ the registry name must be unique

# 3

## Build the Container

❑ Type in the chat:



*use the terminal to build my container in the cloud without using docker and using my Azure Container Registry instead with the name hero-container and tag v1*



GitHub Copilot

To build your container in the cloud without using Docker and using your Azure Container Registry instead with the name **hero-container** and tag **v1**, you can use the **az acr build** command. Here's an example command:

```
az acr build --registry heroacr1 --image hero-container:v1 .
```

This command will build a new container image from the current directory (.) and push it to your Azure Container Registry named **heroacr1** with the name **hero-container** and tag **v1**.

- ❑ Run the commands in the terminal (bash or PowerShell)
- ❑ (Optional) browse the Container Registry and locate the container image that was just built



### heroacr1 | Repositories

Container registry



Refresh



Manage Deleted Repositories



Overview



Activity log



Access control (IAM)



Tags



Quick start



Events



Repositories ↑↓

hero-container

# 4 Create k8s manifest

❑ Type in the chat:



*create a Kubernetes manifest file with a service and a deployment named hero-container-app with my container image and using the acrLoginServer as heroacr1.azurecr.io*



GitHub Copilot



Here's an example Kubernetes manifest file that creates a deployment and a service named **hero-container-app** using your container image from your Azure Container Registry **heroacr1.azurecr.io**:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hero-container-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: hero-container-app
  template:
    metadata:
      labels:
        app: hero-container-app
    spec:
      containers:
        - name: hero-container
          image: heroacr1.azurecr.io/hero-container:v1
          ports:
            - containerPort: 80
          imagePullSecrets:
            - name: acr-auth
---
apiVersion: v1
kind: Service
metadata:
  name: hero-container-app
```

- ❑ Copy the content to a new file named deploy.yaml
- ❑ (Optional) Open the deploy.yaml file and test GitHub Copilot code suggestions



# 5 Deploy the Container

❏ Type in the chat:



*use the terminal to get the aks credentials, run the deploy.yaml manifest file and output the service endpoint*



GitHub Copilot

Here are the commands to get the AKS credentials, deploy the `deploy.yaml` manifest file, and output the service address:

```
az aks get-credentials --resource-group hero-container-rg --name heroaks

kubectl apply -f deploy.yaml

kubectl get service hero-container-app -o jsonpath='{.status.loadBalancer.ingress[0].ip}'
```

The first command gets the AKS credentials for your Kubernetes cluster named `heroaks` in the resource group `hero-container-rg`.

The second command deploys the `deploy.yaml` manifest file to your Kubernetes cluster.

The third command gets the external IP address of the `hero-container-app` service and outputs it to the console. You can use this IP address to access your application.

- ❏ Run the commands in the terminal (bash or PowerShell)
- ❏ Open the command palette and type: Simple Browser: Show and then enter `http://` + the IP returned in the last command

← → ↺ `http://20.31.211.8`

# Welcome to my HERO web server

# Reviewing



## GitHub Copilot

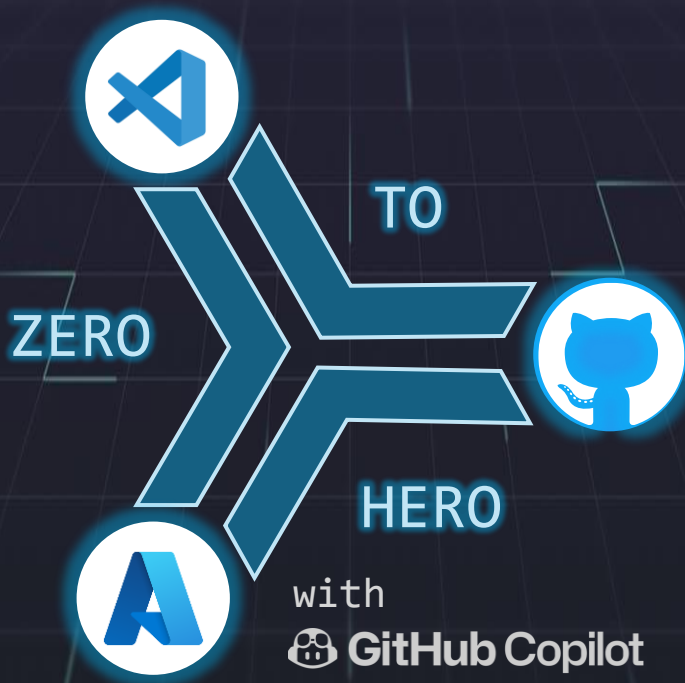
- 👍 Generated the Dockerfile, nginx configuration and custom HTML
- 👍 Generated commands to create a Resource Group, Container Registry and AKS
- 👍 Generated command to build the container in the cloud using the Container Registry
- 👍 Generated the Kubernetes manifest for the deployment and service
- 👍 Generated the commands to get AKS credentials, apply the manifest and output the service endpoint to test the end result



[aka.ms/zero2hero](https://aka.ms/zero2hero)



Thumbs up  
to pump  
for more



[github.com/vieiraae](https://github.com/vieiraae)



[linkedin.com/in/vieira](https://linkedin.com/in/vieira)