SETUP Example AKS and Nginx

Step 1: Set Up Your Azure Environment

1. **Install Azure CLI**: If you don't already have it, install the Azure CLI on your machine.
   - **Windows**: Azure CLI installation instructions
   - **macOS**: `brew update && brew install azure-cli`
   - **Linux**: Follow instructions here
2. **Log in to Azure**:

   ```
   az login
   ```

   - This command logs you into your Azure account using a web browser. It prompts you to open a URL and enter a code to authenticate.

Step 2: Create a Resource Group

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

- `az group create`: Creates a new resource group in Azure.
- `--name myResourceGroup`: Specifies the name of the resource group.
- `--location eastus`: Specifies the Azure region where the resource group will be created (e.g., East US).

Step 3: Create an AKS Cluster

```
az aks create --resource-group myResourceGroup --name myAKSCluster --node-count 1 --enable-addons monitoring --ge
```

- `az aks create`: Creates a new Azure Kubernetes Service (AKS) cluster.
- `--resource-group myResourceGroup`: Specifies the resource group to create the cluster in.
- `--name myAKSCluster`: Specifies the name of the AKS cluster.
- `--node-count 1`: Sets the number of nodes (VMs) in the cluster.
- `--enable-addons monitoring`: Enables monitoring addon for the cluster.
- `--generate-ssh-keys`: Automatically generates SSH keys for authentication.

Step 4: Install Kubectl

- If `kubectl` is not already installed, this command installs it:

  ```
  az aks install-cli
  ```

  - `az aks install-cli`: Installs the `kubectl` CLI, which is used to manage Kubernetes clusters.

Step 5: Connect to Your AKS Cluster

```
az aks get-credentials --resource-group myResourceGroup --name myAKSCluster
```

- `az aks get-credentials`: Retrieves the access credentials for the AKS cluster.
- `--resource-group myResourceGroup`: Specifies the resource group of the cluster.

- `--name myAKSCluster`: Specifies the name of the AKS cluster.

Step 6: Create a Kubernetes Deployment for Your Application

1. **Create a YAML file**:
   - Example content for a simple Nginx deployment is provided. This YAML file describes the desired state of the application deployment in Kubernetes.
2. **Apply the deployment**:

   ```
   kubectl apply -f deployment.yaml
   ```

   - `kubectl apply`: Applies the configuration from the specified file to the Kubernetes cluster.
   - `-f deployment.yaml`: Specifies the path to the deployment YAML file.

Step 7: Expose Your Application

1. **Create a service**:
   - Example content for a service YAML file is provided. This YAML file defines how to expose the deployment to external traffic.
2. **Apply the service**:

   ```
   kubectl apply -f service.yaml
   ```

   - `kubectl apply`: Applies the configuration from the specified file to the Kubernetes cluster.
   - `-f service.yaml`: Specifies the path to the service YAML file.

Step 8: Verify Your Deployment and Service

1. **Check the status of your pods**:

   ```
   kubectl get pods
   ```

   - `kubectl get pods`: Lists all the pods running in the Kubernetes cluster, showing their status.
2. **Check the status of your service**:

   ```
   kubectl get service nginx-service
   ```

   - `kubectl get service`: Lists all the services in the Kubernetes cluster.
   - `nginx-service`: Specifies the name of the service to check.

Step 9: Access Your Application

- After obtaining the external IP address of the service, open a web browser and navigate to that IP address to access your deployed application.

Additional Steps (Optional)

- **Scaling your application**:

  ```
  kubectl scale --replicas=5 deployment/nginx-deployment
  ```

  - `kubectl scale`: Changes the number of replicas for a deployment.
  - `--replicas=5`: Sets the desired number of replicas.
  - `deployment/nginx-deployment`: Specifies the deployment to scale.
- **Updating your application**:

```
kubectl set image deployment/nginx-deployment nginx=nginx:1.17.4
```

- `kubectl set image`: Updates the image for a container in a deployment.
- `deployment/nginx-deployment`: Specifies the deployment to update.
- `nginx=nginx:1.17.4`: Sets the new image version for the Nginx container.

By following these detailed instructions, you should be able to create an AKS cluster and deploy an application to it. If you need further clarification or run into any issues, feel free to ask!

Azure-Samples/aks-load-testing