Deployment - Docs

Table of Contents

Introduction

This documentation provides step-by-step instructions for deploying a containerized application to an Azure Kubernetes Service (AKS) cluster using an Azure Container Registry (ACR) with a PowerShell script. No technical expertise is required to follow the steps outlined in this guide.

Prerequisites

Before using this script, ensure the following requirements are met:

1. You have a Windows machine with PowerShell 5.1 or later installed.
2. You have Git installed on your machine. If not, download and install it from Git website.
3. You have Docker installed on your machine. If not, download and install it from Docker website.
4. Azure PowerShell module is installed. If not, run the following command in PowerShell to install it:

```
Install-Module -Name Az -AllowClobber -Scope CurrentUser
```

5. The powershell-yaml module is installed. If not, run the following command in PowerShell to install it:

```
Install-Module -Name powershell-yaml
```

Overview

This PowerShell script helps users deploy a containerized application to an Azure Kubernetes Service (AKS) cluster using an Azure Container Registry (ACR). The script uses a graphical user interface (GUI) to collect user inputs and automate deployment tasks.

Input Parameters

The script requires the following user inputs:

1. Resource Group: The Azure resource group where your resources are located.
2. Git Repository URL: The web address of the Git repository containing the application source code.
3. Azure Container Registry (ACR) Name: The name of your ACR.
4. Azure Kubernetes Service (AKS) Name: The name of your AKS cluster.
5. Networking Capabilities: Optional selections to enhance your application.
    - Storage Account (optional): Provides scalable cloud storage.
    - Redis (optional): Offers an in-memory data structure store.
    - Database (optional): Enables database management services.

Execution Flow

The script follows these steps:

1. Validates the required modules are installed.
2. Loads WPF and creates a window using XAML.
3. Fetches existing resource groups, ACRs, and AKS clusters from the Azure account.
4. Clones the Git repository containing the application source code.
5. Connects to the Azure account.
6. Builds and pushes the container image to the ACR.
7. Deploys the container image to the AKS cluster.
8. If selected, configures the application with the chosen networking capabilities (Storage Account, Redis, and/or Database).
9. Cleans up temporary files and displays a success message.

Required Azure Roles

The Azure account used with this script must have the following roles:

- Contributor role for managing resources (e.g., Resource Groups, ACRs, AKS clusters).
- Reader role for reading Azure resources' properties.
- AcrPush role for pushing images to the ACR.

Running the Script

To run the script:

1. Open PowerShell, and navigate to the folder containing the script.
2. Run the script using the command: .\script_name.ps1. A graphical user interface will appear.
3. Provide the necessary input parameters and click the Deploy button.

Troubleshooting and Tips

- Ensure the selected Azure account has sufficient permissions to perform all required operations. If you are unsure about your permissions, consult your Azure administrator.
- Check your internet connection to ensure you can access Azure services and the Git repository.
- If errors occur, read the PowerShell console output for detailed error messages and further guidance. You can also consult online forums or contact support if necessary.
- If you encounter issues with the GUI, verify that the required assemblies are loaded correctly. If issues persist, consider reaching out to support or a knowledgeable colleague for assistance.