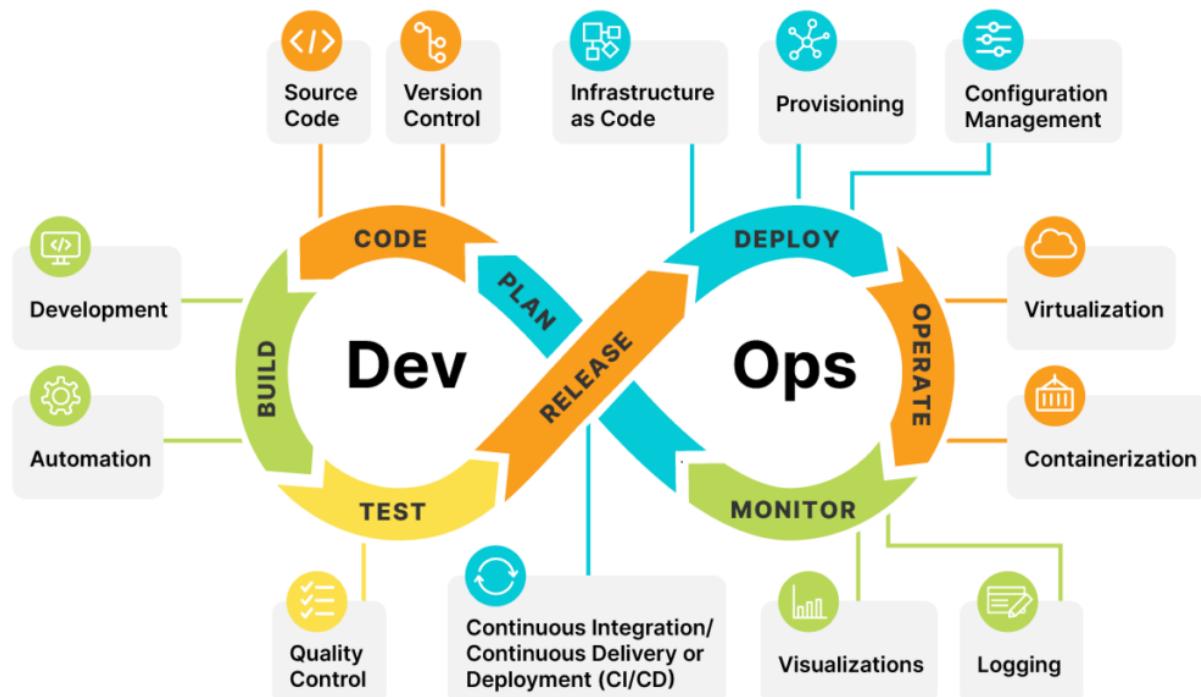
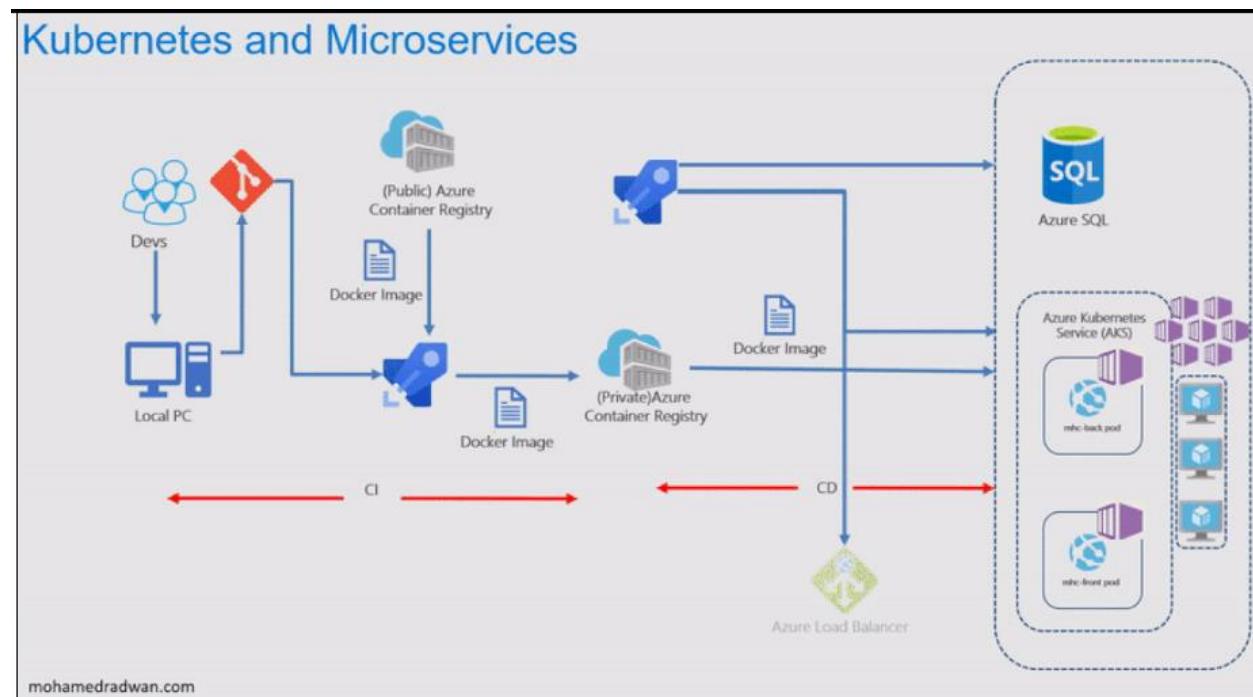


Azure Kubernetes Service (AKS) is the quickest way to use Kubernetes on Azure. **Azure Kubernetes Service (AKS)** manages your hosted Kubernetes environment, making it quick and easy to deploy and manage containerized applications without container orchestration expertise. It also eliminates the burden of ongoing operations and maintenance by provisioning, upgrading, and scaling resources on demand, without taking your applications offline. Azure DevOps helps in creating Docker images for faster deployments and reliability using the continuous build option.

One of the biggest advantages to use AKS is that instead of creating resources in cloud you can create resources and infrastructure inside Azure Kubernetes Cluster through Deployments and Services manifest files.



The architecture



<https://www.azuredevopslabs.com/labs/vstsextend/kubernetes/#exercise-2-configure-release-pipeline>

Phase I: Set up DevOps with our Project

We will generate the code from GitHub for the MyHealthApplication App and load it to our Azure DevOps organization.

Click the following link to access the DemoGenerator tool

<http://azuredevopsdemogenerator.azurewebsites.net/?TemplateId=77372&Name=AKS>

NOTE: Extensions needed incase they are not available on DevOps

1. Kubernetes: <https://marketplace.visualstudio.com/items?itemName=tsuyoshiushio.k8s-endpoint>
2. Replace Tokens:
<https://marketplace.visualstudio.com/items?itemName=qetza.replacetokens>

On portal.azure.com Register the following Resource Providers under your active Azure subscription by selecting Home > Subscriptions > Select your subscription eg Azure Pass Sponsorship > on the left-hand navigation menu select Resource Providers(see page 5)

The homepage to create a new a project. The template should indicate Azure Kubernetes Service

Selected Template : Azure Kubernetes Service

New Project Name : MHC

Select Organization : manassehmwangi

Note:
If you're using Demo Generator for the first time, please make the following setting in your organization settings

Security

- Policies (highlighted with a red box)
- Permissions

Application connection policies

- On Alternate authentication credentials
- On Third-party application access via OAuth (highlighted with a red box)
- On SSH authentication

About this Template

Azure Kubernetes Service

Use this template to learn how to deploy a Docker-based ASP.NET Core web application to a Kubernetes cluster running in Azure Kubernetes Service (AKS) with Azure Pipelines.

Lab URL: Deploying to Azure Container Service (AKS)

What's new?

- New templates in Microsoft Learning and Cloud Adoption Framework platform
- Switch Directory option for switching between organization directories. Click on Sign In once you switch the organization.

Create Project

After run create project this should be the result

Feedback:

To provide feedback, you can email us here .
Like our tool? We would appreciate if you share feedback on social

Boards

On Third-party application access via OAuth (highlighted with a red box)

Process

On SSH authentication

Verifying if all required extension(s) are installed and enabled

All required extensions are installed/enabled in your Azure DevOps Organization.

- Kubernetes extension
- Replace Tokens

Create Project

Congratulations! Your project is successfully provisioned.

Navigate to project

Like the tool? Share your feedback

Project MHC created
3 team(s) created
Board-Column, Swimlanes, Styles updated
Work items created
Build definition created
Release definition created

The homepage for the project

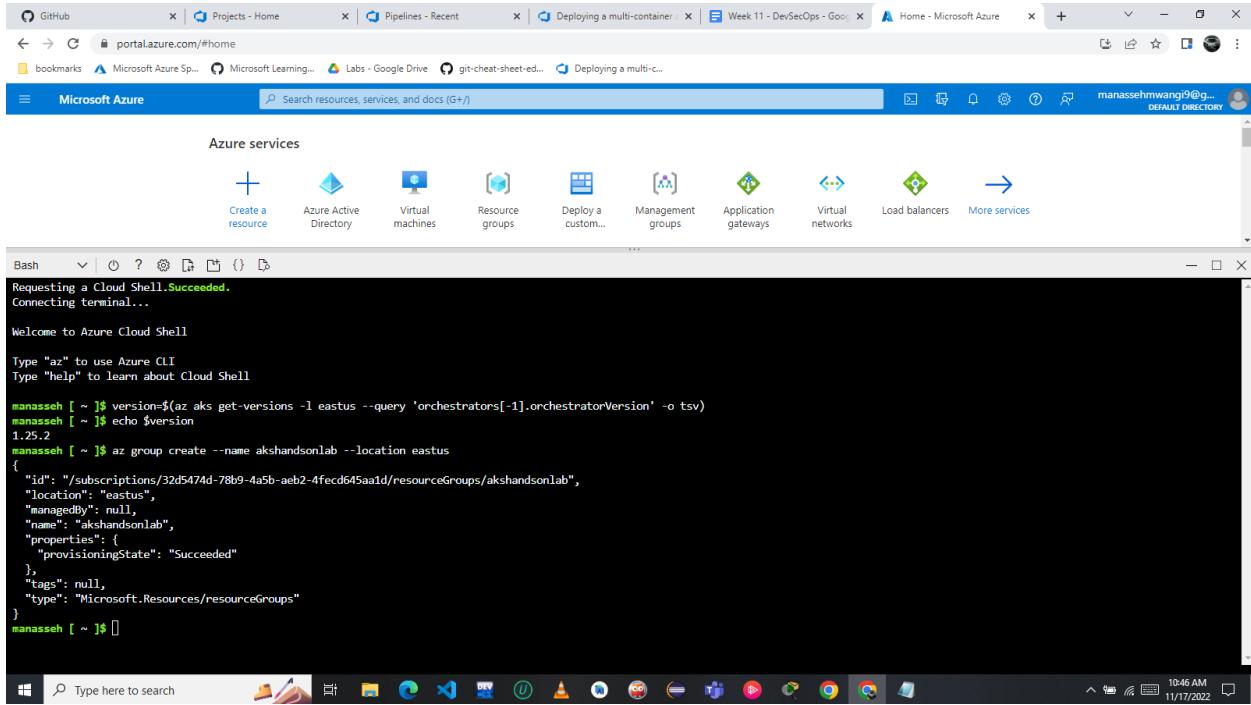
The screenshot shows the Azure DevOps Project Summary page for the project 'MHC'. The left sidebar contains navigation links for Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. The main content area displays the 'About this project' section, which states 'Generated by Azure DevOps Demo Generator'. To the right, there is a 'Project stats' section showing metrics for the last 7 days: 73 work items created, 9 work items completed, 0 pull requests opened, 23 commits by 3 authors, 0% builds succeeded, and 0% deployments succeeded. Below this is a 'Members' section with one user listed. The bottom of the screen shows the Windows taskbar with various pinned icons.

There are two pipelines as indicated we will start with the MyHealth.aks.build

The screenshot shows the Azure DevOps Pipelines page for the project 'MHC'. The left sidebar has a 'Pipelines' section selected. The main content area displays a 'Recently run pipelines' table with two entries: 'MyHealth.AKS.build-YAML' and 'MyHealth.AKS.build', both of which have 'No runs yet'. A 'New pipeline' button is located in the top right corner. The bottom of the screen shows the Windows taskbar with various pinned icons.

Get the latest available Kubernetes version in your preferred region into a bash variable.

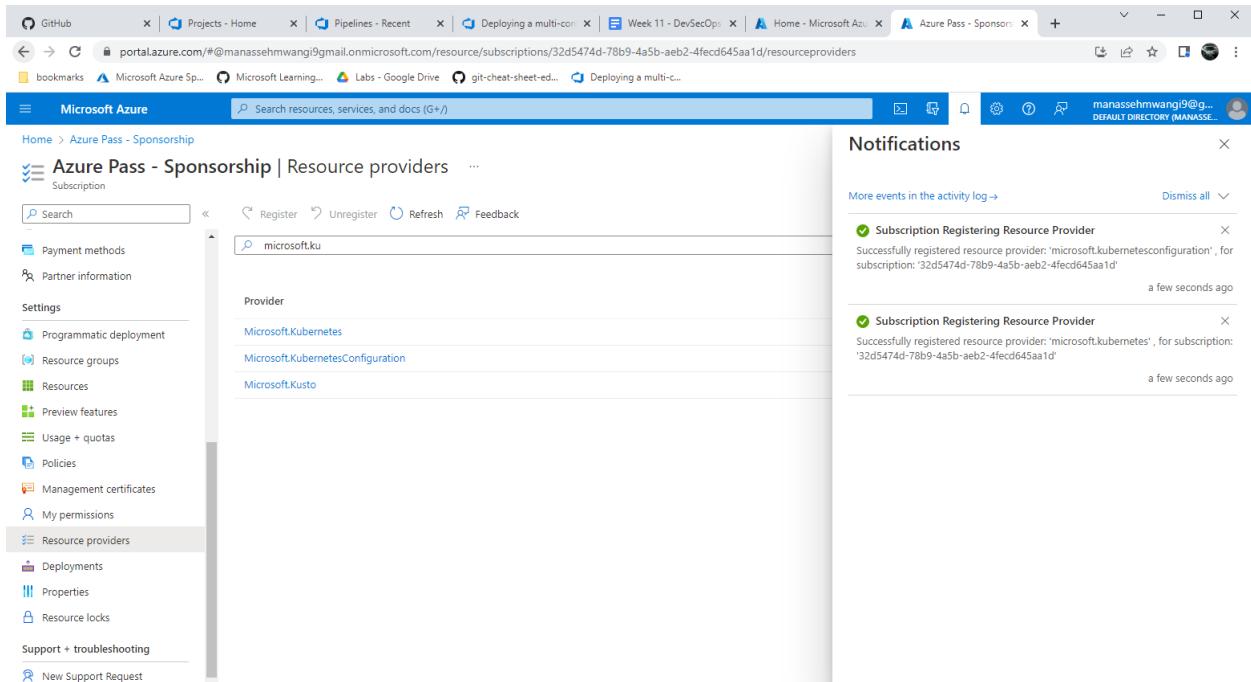
Create a Resource Group



The screenshot shows a Microsoft Edge browser window with multiple tabs open. The active tab is 'Bash' in the Microsoft Azure Cloud Shell. The terminal output shows the creation of a new resource group named 'akshandsonlab' in the 'eastus' location. The command used was 'az group create --name akshandsonlab --location eastus'. The output indicates success, showing the JSON representation of the newly created resource group.

```
Bash
Requesting a Cloud Shell...Succeeded.
Connecting terminal...
Welcome to Azure Cloud Shell
Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell
manasseh [ ~ $] version=$(az aks get-versions -l eastus --query 'orchestrators[-1].orchestratorVersion' -o tsv)
manasseh [ ~ $] echo $version
1.25.2
manasseh [ ~ $] az group create --name akshandsonlab --location eastus
{
  "id": "/subscriptions/32d5474d-78b9-4a5b-aeb2-4fecfd645aa1/resourceGroups/akshandsonlab",
  "location": "eastus",
  "managedBy": null,
  "name": "akshandsonlab",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
manasseh [ ~ $]
```

Register the following Resource Providers under your active Azure subscription



The screenshot shows the 'Resource providers' section of the Azure Pass - Sponsorship portal. The left sidebar lists various settings and providers. The main area displays a search bar and a list of registered providers, including 'Microsoft.Kubernetes', 'Microsoft.KubernetesConfiguration', and 'Microsoft.Kusto'. A notifications panel on the right shows two recent events: 'Subscription Registering Resource Provider' entries for each of the three providers, indicating successful registration.

Notifications

- More events in the activity log → Dismiss all
- Subscription Registering Resource Provider ×
Successfully registered resource provider: 'microsoft.kubernetesconfiguration', for subscription: '32d5474d-78b9-4a5b-aeb2-4fecfd645aa1'd
a few seconds ago
- Subscription Registering Resource Provider ×
Successfully registered resource provider: 'microsoft.kubernetes', for subscription: '32d5474d-78b9-4a5b-aeb2-4fecfd645aa1'
a few seconds ago

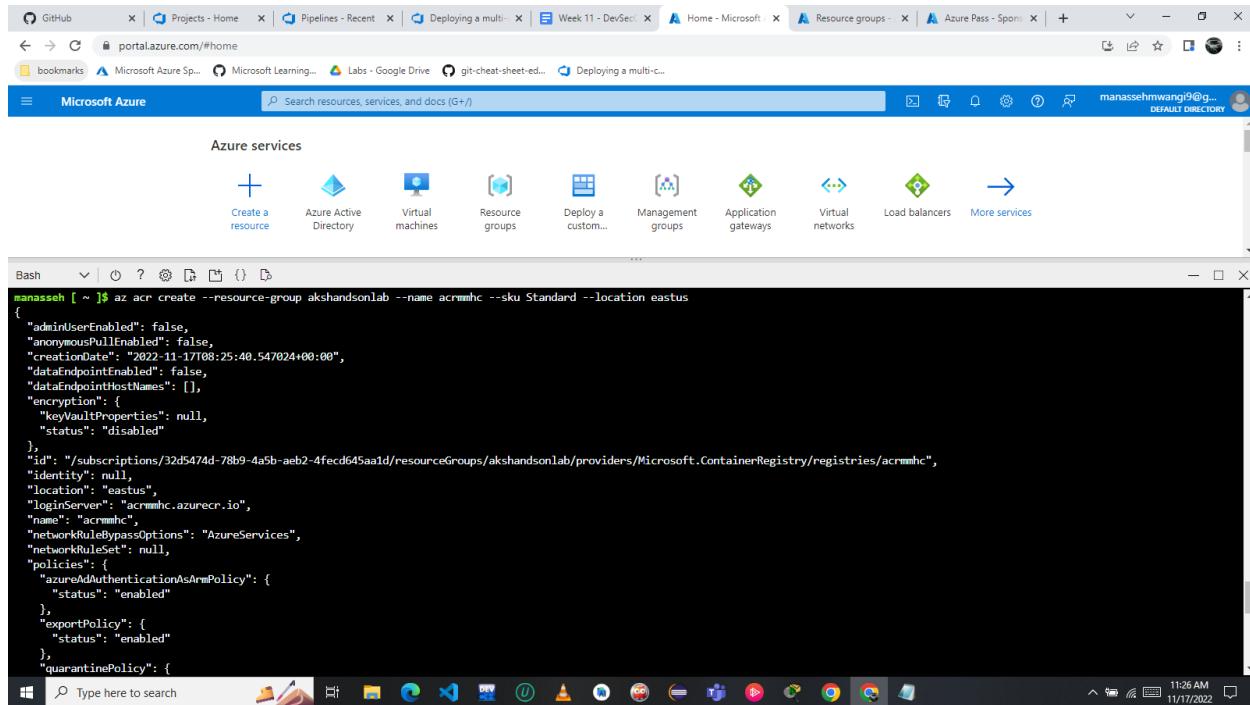
Microsoft.OperationalInsights, Microsoft.OperationsManagement, Microsoft.ContainerService, Microsoft.Kubernetes, Microsoft.KubernetesConfiguration, Microsoft.Sql

The screenshot shows the Azure Pass - Sponsorship | Resource providers page. The left sidebar lists various service categories like Payment methods, Partner information, Settings, Resource providers, Deployments, Properties, Resource locks, Support + troubleshooting, and New Support Request. The main area displays a search bar with the query "microsoft.contai". Below it, a table lists registered providers: Microsoft.ContainerInstance, Microsoft.ContainerRegistry, and Microsoft.ContainerService. On the right, a Notifications panel shows six recent events all labeled "Subscription Registering Resource Provider" with green checkmarks, indicating successful registration for the specified providers.

Create AKS using the latest version available

The screenshot shows the Azure portal with the Azure services dashboard. The "Resource groups" button is highlighted. Below, a terminal window titled "Bash" is open, displaying the command-line output of creating an AKS cluster named "aksclustermm". The command used was "az aks create --resource-group akshandsonlab --name aksclustermm --enable-addons monitoring --kubernetes-version \$version --generate-ssh-keys --location eastus". The output includes SSH key generation instructions and AAD role propagation details. The terminal also shows the user's environment variables and the current working directory as "/home/manasseh". The bottom of the screen shows the Windows taskbar with various pinned icons.

Deploy Azure Container Registry(ACR)

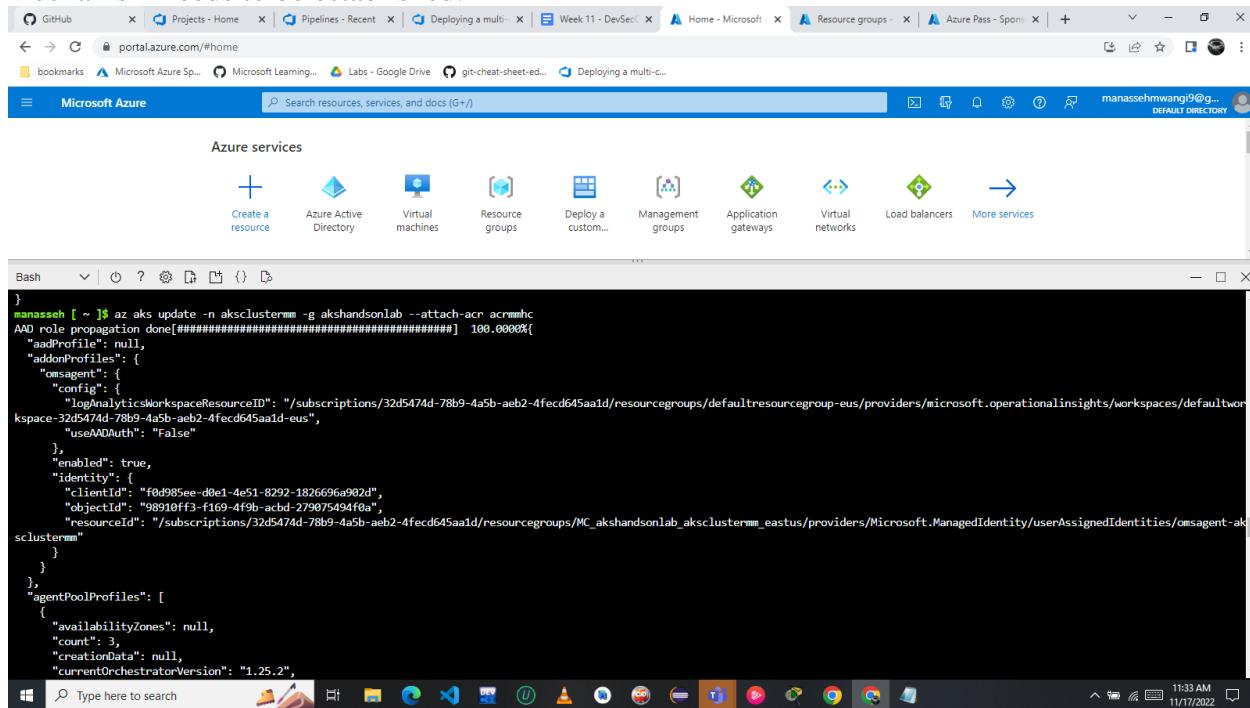


A screenshot of the Microsoft Azure portal. At the top, there are multiple browser tabs open, including GitHub, Projects - Home, Pipelines - Recent, Deploying a multi..., Week 11 - DevSec..., Home - Microsoft, Resource groups, and Azure Pass - Sponsor. The main navigation bar shows 'Microsoft Azure' and a search bar. Below the navigation is a 'Azure services' section with icons for Create a resource, Azure Active Directory, Virtual machines, Resource groups, Deploy a custom..., Management groups, Application gateways, Virtual networks, Load balancers, and More services. A terminal window is open in the center, showing a Bash session. The command entered is:

```
manasseh [ ~ ]$ az acr create --resource-group akshandonlab --name acrmhc --sku Standard --location eastus
```

The output of the command is a JSON object representing the newly created ACR resource group, including details like id, location, name, and policies.

Authenticate with Azure Container Registry from Azure Kubernetes Service : When you're using Azure Container Registry (ACR) with Azure Kubernetes Service (AKS), an authentication mechanism needs to be established.



A screenshot of the Microsoft Azure portal, similar to the previous one but with a different terminal history. The terminal window shows a Bash session with the following commands:

```
}
```

```
manasseh [ ~ ]$ az aks update -n aksClustermm -g akshandonlab --attach-acr acrmhc
```

The output indicates that AAD role propagation is being done. The terminal then continues with more configuration details for the AKS cluster, including OMS agent and identity settings.

Create an Azure SQL server

A screenshot of a Microsoft Azure terminal window titled "Bash". The terminal shows the command \$ az sql server create -l eastus -g akshandsonlab -n sqlsvrmmm -u sqladmin -p \$Mansemedi being run. The output is a JSON object representing the newly created Azure SQL server resource.

```
manasseh [ ~ ]$ az sql server create -l eastus -g akshandsonlab -n sqlsvrmmm -u sqladmin -p $Mansemedi
{
  "administratorLogin": "sqladmin",
  "administratorLoginPassword": null,
  "administrators": null,
  "federatedClientId": null,
  "fullyQualifiedDomainName": "sqlsvrmmm.database.windows.net",
  "id": "/subscriptions/32d5474d-78b9-4a5b-aeb2-4fed645aa1d/resourceGroups/akshandsonlab/providers/Microsoft.Sql/servers/sqlsvrmmm",
  "identity": null,
  "keyId": null,
  "kind": "v12.0",
  "location": "eastus",
  "minimalTlsVersion": null,
  "name": "sqlsvrmmm",
  "primaryUserAssignedIdentityId": null,
  "privateEndpointConnections": [],
  "publicNetworkAccess": "Enabled",
  "resourceGroup": "akshandsonlab",
  "restrictOutboundNetworkAccess": "Disabled",
  "state": "Ready",
  "tags": null,
  "type": "Microsoft.Sql/servers",
  "version": "12.0",
  "workspaceFeature": null
}
```

Create a database in the SQL Server

A screenshot of a Microsoft Azure terminal window titled "Bash". The terminal shows the command \$ az sql db create -g akshandsonlab -s sqlsvrmmm -n mhcdb --service-objective S0 being run. The output is a JSON object representing the newly created Azure SQL database resource.

```
{
  "type": "Microsoft.Sql/servers",
  "version": "12.0",
  "workspaceFeature": null
}
manasseh [ ~ ]$ az sql db create -g akshandsonlab -s sqlsvrmmm -n mhcdb --service-objective S0
{
  "autoPauseDelay": null,
  "catalogCollation": "SQL_Latin1_General_CI_AS",
  "collation": "SQL_Latin1_General_CI_AS",
  "createMode": null,
  "creationDate": "2022-11-17T08:43:17.563000+00:00",
  "currentBackupStorageRedundancy": "Geo",
  "currentServiceObjectiveName": "S0",
  "currentSku": {
    "capacity": 10,
    "family": null,
    "name": "Standard",
    "size": null,
    "tier": "Standard"
  },
  "databaseId": "3e2b0029-a7d5-4c91-bc8b-be958a6a1a51",
  "defaultSecondaryLocation": "westus",
  "earliestRestoreDate": null,
  "edition": "Standard",
  "elasticPoolId": null,
  "elasticPoolName": null,
  "failoverGroupId": null,
  "federatedClientId": null,
```

Ensure your environment look like this:

The screenshot shows the Microsoft Azure portal interface. The user is in the 'Resource groups' section of the 'akshandsonlab' resource group. The left sidebar lists several resource groups: 'akshandsonlab', 'cloud-shell-storage-westeurope', 'DefaultResourceGroup-EU5', 'MC_akshandsonlab_aksclustermm_eastus', and 'NetworkWatcherRG'. The main pane shows the 'Subscription (move)' and 'Azure Pass - Sponsorship' sections, followed by a table of resources. The table includes columns for Name, Type, and Location. Resources listed include 'acrmmhc' (Container registry, East US), 'aksclustermm' (Kubernetes service, East US), 'mhcdb' (sqlsvrmmm/mhcdb) (SQL database, East US), and 'sqlsvrmmm' (SQL server, East US). At the bottom right of the main pane, there is a 'Give feedback' link.

Select the mhcdb SQL database and make a note of the Server name.

The screenshot shows the Microsoft Azure portal interface. The user is in the 'mhcdb (sqlsvrmmm/mhcdb)' database blade under the 'akshandsonlab' resource group. The left sidebar lists various management options: Overview, Activity log, Tags, Diagnose and solve problems, Getting started, Query editor (preview), Compute + storage, Connection strings, Properties, Locks, Data management (Replicas, Sync to other databases), Integrations (Azure Synapse Link, Stream analytics (preview)), and a search bar at the bottom. The main pane shows the 'Essentials' section with details such as Resource group (move) : akshandsonlab, Status : Online, Location : East US, Subscription (move) : Azure Pass - Sponsorship, Subscription ID : 32d5474d-78b9-4a5b-aeb2-4fecc645aa1d, and Server name : sqlsvrmmm.database.windows.net. Below the essentials, there are sections for 'Getting started', 'Start working with your database', 'Configure access', 'Connect to application', 'See connection strings', 'Start developing', and 'Open Azure Data Studio'. The status bar at the bottom indicates the time as 11:44 AM on 11/17/2022.

Click on “Set server Firewall” and enable “Allow Azure services …” option.

The screenshot shows the Microsoft Azure portal interface. The left sidebar navigation includes 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Diagnose and solve problems', 'Quick start', 'Settings' (with options like 'Azure Active Directory', 'SQL databases', 'SQL elastic pools', 'DTU quota', 'Properties', 'Locks'), 'Data management' (with options like 'Backups', 'Deleted databases', 'Failover groups', 'Import/Export history'), and 'Search' bar. The main content area is titled 'sqlsvrmmm | Networking'. It displays 'Virtual networks' and 'Firewall rules'. Under 'Firewall rules', there is a table with columns 'Rule', 'Virtual network', 'Subnet', 'Address range', 'Endpoint status', 'Resource group', 'Subscription', and 'State'. A note says 'Connections from the IP addresses configured in the Firewall rules section below will have access to this database. By default, no public IP addresses are allowed.' Below the table, there is a section for 'Exceptions' with a checked checkbox for 'Allow Azure services and resources to access this server'. At the bottom are 'Save' and 'Discard' buttons. The top navigation bar shows multiple tabs and the user's email 'manassehwmwang19@gmail.com'.

Navigate to the resource group, select the created container registry and make a note of the Login server name

The screenshot shows the Microsoft Azure portal interface. The left sidebar navigation includes 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Quick start', 'Events', 'Settings' (with options like 'Access keys', 'Encryption', 'Identity', 'Networking', 'Microsoft Defender for Cloud', 'Locks'), 'Services' (with options like 'Repositories', 'Webhooks', 'Replications', 'Tasks'), and 'Search' bar. The main content area is titled 'acrmhc - Microsoft Azure'. It displays 'Essentials' information for a container registry named 'acrmhc'. The details include: Resource group (move) : akshansonlab, Location : East US, Subscription (move) : Azure Pass - Sponsorship, Subscription ID : 32d5474d-78b9-4a5b-aeb2-4fec645aa1d, Login server : acrmhc.azurecr.io, Creation date : 11/17/2022, 11:25 AM GMT+, SKU : Standard, Provisioning state : Succeeded, Soft Delete (Preview) : Disabled. There is also a 'Copied' message. Below this are sections for 'Usage' (Included in SKU: 100 GB, Used: 0.00 GB, Additional storage: 0.00 GB) and 'ACR Tasks' (Build, Run, Push and Patch containers in Azure with ACR Tasks. Tasks supports Windows, Linux and ARM with QEMU). At the bottom, there is a 'Container security integrations' section for 'Microsoft Defender for Cloud' (Vulnerability management, runtime protection, and hardening recommendations for your containers and container environments). The top navigation bar shows multiple tabs and the user's email 'manassehwmwang19@gmail.com'.

Phase III: Configure Build Pipeline

Navigate to Pipelines and configure the MyHealth.AKS.build > Click Edit

Select the correct subscription on each and the container registry Run, Build, Push and Lock services

The screenshot shows the Azure DevOps Pipelines interface for the 'MyHealth.AKS.build' pipeline. The left sidebar shows project navigation with 'MHC' selected. The main area displays the pipeline structure:

- Phase 1** (Run on agent):
 - Get sources (AKS master)
 - Replace tokens in appsettings.json (Replace Tokens)
 - Replace tokens in mhc-aks.yaml (Replace Tokens)
 - Run services (Docker Compose)**: This task is currently selected, indicated by a blue border. It has three sub-options: Build services, Push services, and Lock services.
 - Build services (Docker Compose)
 - Push services (Docker Compose)
 - Lock services (Docker Compose)
 - Copy Files (Copy files)

On the right, the 'Run services' task configuration pane is open:

- Task version: 0.*
- Display name: Run services
- Container Registry Type: Azure Container Registry
- Azure subscription: Azure Pass - Sponsorship (32d5474d-78b9-4a5b-aeb2-4fec645aa1d)
- Azure Container Registry: (dropdown menu)
- Docker Compose File: docker-compose.ci.build.yml
- Additional Docker Compose Files: (empty)
- Environment Variables: (empty)

Change the variables to correspond with the ones used on Azure portal

The screenshot shows the Azure DevOps Variables configuration for the 'MyHealth.AKS.build' pipeline. The left sidebar shows project navigation with 'MHC' selected. The main area displays the 'Variables' tab:

Name	Value	Settable at queue
ACR	*****	
SQLpassword	*****	
SQLserver	*****	
SQLUser	*****	
system.collectionId	aa3094c6-f99d-42cd-8b97-79b40d95ae8b	
system.debug	false	<input checked="" type="checkbox"/>
system.definitionId	1	
system.teamProject	MHC	

Before saving there is another button for variables , edit the credentials

The screenshot shows the Azure DevOps interface for a pipeline named 'MyHealth.AKS.build-YAML'. On the left, a sidebar lists project navigation options like Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Artifacts. The main area displays the YAML configuration file for the pipeline:

```
master AKS / azure-pipelines.yml

Settings
13 - task: replaceTokens@4
14   displayName: 'Replace tokens in appsettings.json'
15   inputs:
16     - rootDirectory: '${build.sourcesdirectory}/src/MyHealth.Web'
17     - targetFiles: appsettings.json
18     - escapeType: none
19
20 Settings
21 - task: replaceTokens@3
22   displayName: 'Replace tokens in mhc-aks.yaml'
23   inputs:
24     - targetFiles: 'mhc-aks.yaml'
25     - escapeType: none
26     - tokenPrefix: '_'
27     - tokenSuffix: '_'
28
29 Settings
30 - task: DockerCompose@0
31   displayName: 'Run services'
32   inputs:
33     - containerRegistryType: 'Azure Container Registry'
34     - azureSubscription: 'ARM'
35     - azureContainerRegistry: 'dockerreg0912.azurecr.io'
36     - dockerComposeFile: 'docker-compose.ci.build.yml'
37     - action: 'Run services'
38     - detached: false
39
40 Settings
41 - task: DockerCompose@0
```

To the right of the code editor, a panel titled 'Docker Compose' is open, showing configuration for the Docker Compose task. It includes fields for Container Registry Type (set to Azure Container Registry), Azure subscription (set to 'Azure Pass - Sponsorship (32d5474d-78...)'), and Docker Compose File (set to 'docker-compose.ci.build.yml'). There are also sections for Additional Docker Compose Files and Environment Variables.

After clicking save

The screenshot shows the Azure DevOps Pipelines dashboard. The left sidebar remains the same, showing the 'Pipelines' section is currently selected. The main area displays a list of pipelines under the heading 'All pipelines':

- MyHealth.AKS.build-YAML #20221122.1 • Update azure-pipelines.yml for Azure Pipelines
- MyHealth.AKS.build No runs yet

A blue 'New pipeline' button is visible in the top right corner of the main dashboard area.

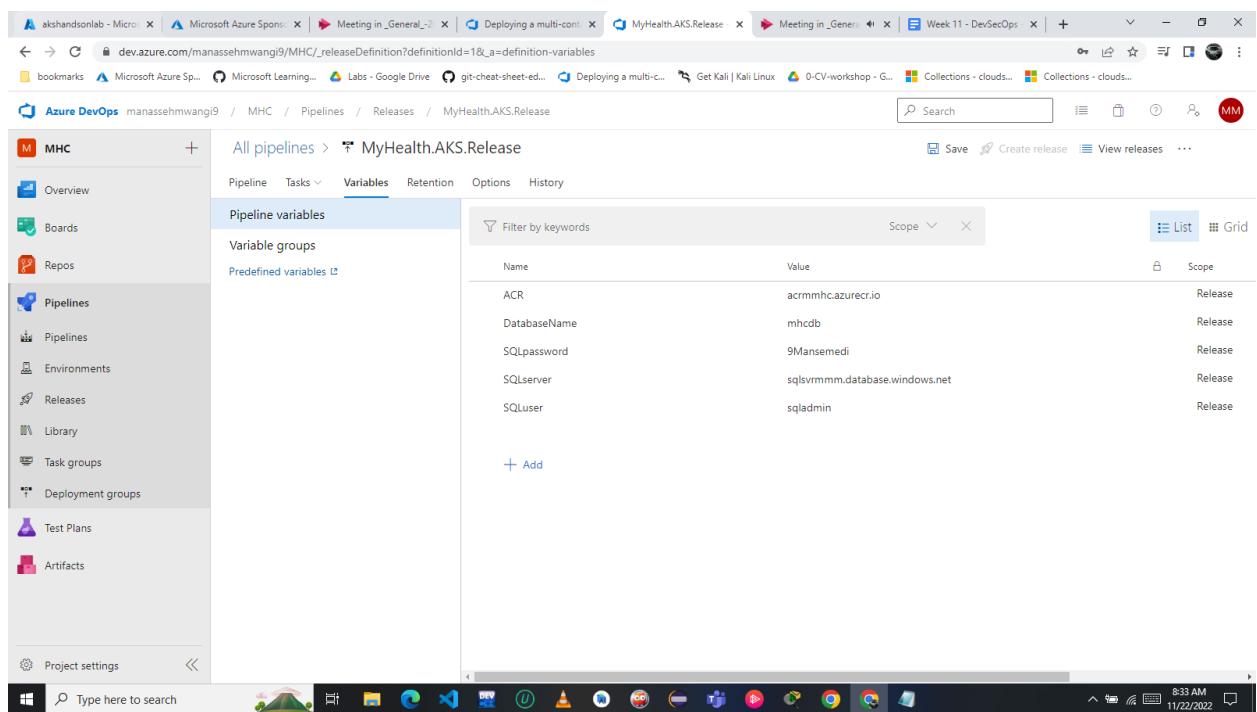
Configure Release pipeline

Navigate to **Pipelines | Releases**. Select **MyHealth.AKS.Release** pipeline and click **Edit**.

The screenshot shows the Azure DevOps Pipelines interface. On the left, there's a sidebar with project navigation. The main area displays the 'All pipelines > MyHealth.AKS.Release' screen. It shows an artifact named 'MyHealth.AKS.build' and a single stage named 'Dev'. The 'Dev' stage is expanded, showing two jobs and three tasks. To the right, there's a 'Stage' configuration panel where the stage name is set to 'Dev' and the owner is listed as 'Manasseh Mwangi'. The bottom of the screen shows a Windows taskbar with various icons.

Select Dev stage and click **View stage tasks** to view the pipeline tasks
update the **Azure Subscription** value from the dropdown

The screenshot shows the 'Tasks' tab for the 'Dev' stage of the 'MyHealth.AKS.Release' pipeline. The tasks listed are 'DB deployment' (Run on agent) and 'AKS deployment' (Run on agent). The 'DB deployment' task is selected, and its details are shown in the center. A modal dialog is open for the 'Execute Azure SQL : DacpacTask' task, specifically for the 'Container registry' configuration. It shows the 'Container registry type' as 'Azure Container Registry', the 'Azure subscription' as 'Azure Pass - Sponsorship (32d5474d-78b9-4a5b-aeb2-4fec645aa1d)', and the 'Azure container registry' as 'acrmmhc'. Other tabs like 'Variables', 'Retention', 'Options', and 'History' are also visible on the pipeline page.

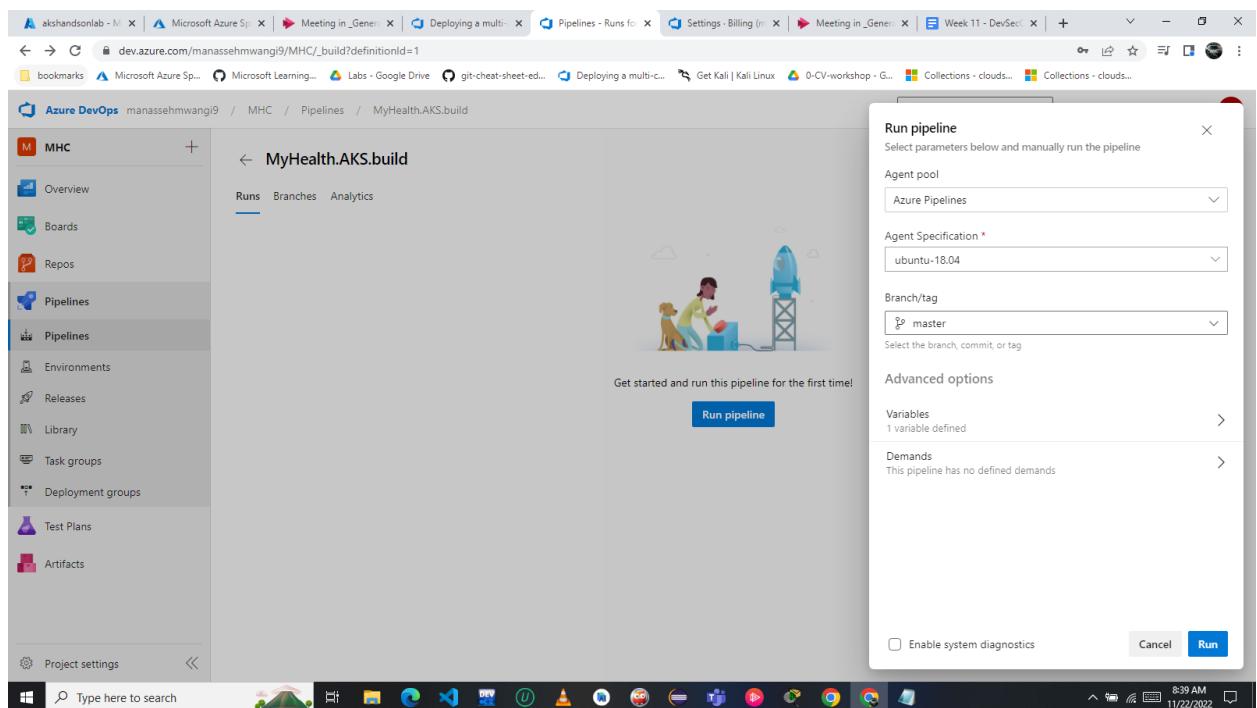


A screenshot of the Azure DevOps interface showing the 'Pipeline variables' section for a pipeline named 'MyHealth.AKS.Release'. The left sidebar shows project navigation with 'MHC' selected. The main area displays a table of predefined variables:

Name	Value	Scope
ACR	acrmmhc.azurecr.io	Release
DatabaseName	mhcdb	Release
SQLPassword	9Mansemedi	Release
SQLServer	sqlsvrmm.database.windows.net	Release
SQLUser	sqladmin	Release

Buttons for 'Save', 'Create release', and 'View releases' are at the top right. A search bar and filter options are also present.

Now its time to run the pipeline



A screenshot of the Azure DevOps interface showing the 'Run pipeline' dialog for a pipeline named 'MyHealth.AKS.build'. The left sidebar shows project navigation with 'MHC' selected. The dialog contains the following fields:

- Agent pool:** Azure Pipelines
- Agent Specification:** ubuntu-18.04
- Branch/tag:** master
- Advanced options:** Variables (1 variable defined), Demands (This pipeline has no defined demands)

At the bottom are 'Run pipeline' and 'Cancel' buttons, along with a checkbox for 'Enable system diagnostics'.

The pipeline services are running

Azure DevOps Pipelines log for build #20221122.1:

```
165 ci-build_1 | Installing Microsoft.AspNetCore.Server.Kestrel 1.0.3.
166 ci-build_1 | Installing Microsoft.VisualStudio.Web.BrowserLink 1.0.1.
167 ci-build_1 | Installing Microsoft.AspNetCore.Server.IISIntegration 1.0.2.
168 ci-build_1 | Installing Microsoft.AspNetCore.StaticFiles 1.0.2.
169 ci-build_1 | Installing Microsoft.AspNetCore.Diagnostics.EntityFrameworkCore 1.0.2.
170 ci-build_1 | Installing Microsoft.AspNetCore.Session 1.0.2.
171 ci-build_1 | Installing Microsoft.EntityFrameworkCore.InMemory 1.0.3.
172 ci-build_1 | Generating MSBuild file /src/src/MyHealth.Web/obj/MyHealth.Web.csproj.nuget.g.props.
173 ci-build_1 | Generating MSBuild file /src/src/MyHealth.Web/obj/MyHealth.Web.csproj.nuget.g.targets.
174 ci-build_1 | Restore completed in 3.74 sec for /src/src/MyHealth.Web/MyHealth.Web.csproj.
175 ci-build_1 | Restoring packages for /src/src/MyHealth.Web/MyHealth.Web.csproj...
176 ci-build_1 | Generating MSBuild file /src/test/MyHealth.API.IntegrationTests/obj/MyHealth.API.IntegrationTests.csproj.nuget.g.props.
177 ci-build_1 | Generating MSBuild file /src/test/MyHealth.API.IntegrationTests/obj/MyHealth.API.IntegrationTests.csproj.nuget.g.targets.
178 ci-build_1 | Restore completed in 3.74 sec for /src/test/MyHealth.API.IntegrationTests/MyHealth.API.IntegrationTests.csproj.
179 ci-build_1 | Restoring packages for /src/src/MyHealth.Web/MyHealth.Web.csproj...
180 ci-build_1 | Installing Microsoft.NETCore.Jit 1.0.2.
181 ci-build_1 | Installing Microsoft.NETCore.Runtime.CoreCLR 1.0.2.
182 ci-build_1 | Installing Microsoft.NETCore.DotNetHostPolicy 1.0.1.
183 ci-build_1 | Installing Microsoft.NETCore.App 1.0.0.
184 ci-build_1 | Installing Microsoft.Extensions.SecretManager.Tools 1.0.0.
185 ci-build_1 | Installing NUGITify 1.5.1.
186 ci-build_1 | Installing BundlerMinifier.Core 2.2.301.
187 ci-build_1 | Restore completed in 253.94 ms for /src/src/MyHealth.Web/MyHealth.Web.csproj.
188 ci-build_1 | Restore completed in 1.25 sec for /src/src/MyHealth.Web/MyHealth.Web.csproj.
189 ci-build_1 | Microsoft (R) Build Engine version 15.3.409.57025 for .NET Core
190 ci-build_1 | Copyright (C) Microsoft Corporation. All rights reserved.
191 ci-build_1 |
192 ci-build_1 | MyHealth.Model -> /src/src/MyHealth.Model/bin/Release/netstandard1.6/MyHealth.Model.dll
193 ci-build_1 | MyHealth.Model -> /src/src/MyHealth.Model/obj/Docker/publish/
```

Azure DevOps Pipelines log for build #20221122.1:

```
1 ##[warning]The ubuntu-18.04 environment is deprecated, consider switching to ubuntu-20.04(ubuntu-latest), or ubuntu-22.04 instead. For more details see https://aka.ms/runner-deprecation-warning
2 Pool: Azure Pipelines
3 Image: ubuntu-18.04
4 Agent: Hosted Agent
5 Started: Today at 8:40 AM
6 Duration: 2m 20s
7
8 ▶ Job preparation parameters
9 ▶ 1 queue time variable used
10 □ 1 artifact produced
11 Job live console data:
12 Finishing: Phase 1
```

When you go back to the Azure portal and check repo in the container registry this should show

Microsoft Azure

myhealth.web - Container registry

Repository

myhealth.web

Last updated date: 11/22/2022, 8:42 AM GMT+3

Tag count: 2

Manifest count: 1

Tags	Digest	Last modified
latest	sha256:7a0cc353bcd25d3a0f99f19fd753c69823755e...	11/22/2022, 8:42 AM GMT+3
2	sha256:7a0cc353bcd25d3a0f99f19fd753c69823755e...	11/22/2022, 8:42 AM GMT+3

Switch back to the Azure DevOps portal. Select the **Releases** tab in the **Pipelines** section and double-click on the latest release

Azure DevOps

manassehmwangi9 / MHC / Pipelines / Releases / MyHealth.AKS.Release / Release-1

Release

Continuous deployment for Manasseh Mwangi on 11/22/2022, 8:42 AM

Stages

Dev

Succeeded

1 warning on 11/22/2022, 8:45 AM

The screenshot shows the Azure DevOps Pipelines interface for a project named 'MyHealth.AKS.Release'. The pipeline stage is 'Release-1' and the task is 'Dev'. The 'Logs' tab is selected, showing a successful 'DB deployment' step. The log details are as follows:

Task	Status	Duration
Initialize job	succeeded	6s
Download artifact - MyHealth.AKS.build - deploy	succeeded	2s
Execute Azure SQL : DacpacTask	succeeded	2m 19s
Finalize Job	succeeded	<1s

The pipeline also includes an 'AKS deployment' step with one warning.

To access the application, run the below command. you should see an **External-IP**

The screenshot shows the Microsoft Azure portal with the 'Azure services' section open. A Cloud Shell terminal is running, displaying the output of several commands:

```
Bash
Requesting a Cloud Shell.Succeeded.
Connecting terminal...
Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

manasseh [ ~ ]$ az aks get-credentials --resource-group akshandsonlab --name aksclustermm
Merged "aksclustermm" as current context in /home/manasseh/.kube/config
manasseh [ ~ ]$ az aks get-credentials --resource-group akshandsonlab --name aksclustermm
Merged "aksclustermm" as current context in /home/manasseh/.kube/config
manasseh [ ~ ]$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
mhc-back-5c56d9b547-61klp  1/1   Running   0          12m
mhc-front-5f9fffb6997-cnhdr  0/1   CrashLoopBackOff  6 (2m18s ago)  12m
manasseh [ ~ ]$ kubectl get service mhc-front --watch
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
mhc-front   LoadBalancer  10.0.198.207  52.146.65.113  80:30131/TCP  12m
```

Or you can navigate to your kubernetes cluster and select services and progress
You should see an external IP Adress

The screenshot shows the Microsoft Azure portal interface. The left sidebar is collapsed, and the main area displays the 'Services and ingresses' section for the 'aksclusterm' Kubernetes service. The 'Services' tab is selected. A table lists several services with their details:

Name	Namespace	Status	Type	Cluster IP	External IP	Ports	Age
kubernetes	default	Ok	ClusterIP	10.0.0.1		443/TCP	5 days
kube-dns	kube-system	Ok	ClusterIP	10.0.0.10		53/UDP, 53/TCP	5 days
metrics-server	kube-system	Ok	ClusterIP	10.0.18.179		443/TCP	5 days
mhc-back	default	Ok	ClusterIP	10.0.81.176		6379/TCP	17 minutes
mhc-front	default	Ok	LoadBalancer	10.0.198.207	52.146.65.113	80:30131/TCP	17 minutes

This is the application that have been deployed

The screenshot shows a web browser window with multiple tabs open. The active tab displays the website 'HEALTHCLINIC.BIZ'. The page has a teal background and features the text 'Take care of your loved ones in a simpler form' and 'Our technology lets you follow the care and treatment of people who need it most.' Below this text are several small circular icons. The browser's taskbar at the bottom shows various pinned and open applications.