Dokumentasjon : Komme i gang med Azure og Go (AKS)

1. Jeg installerte OG fra <https://go.dev/dl/>
2. Azure CLI var allerede installert, men om ikke, kan det lastes ned herfra:  
   <https://learn.microsoft.com/en-us/cli/azure/install-azure-cli> , dette kan deretter bekreftes med «az version»
3. jeg opprettet en mappe som heter «aks-terraform-GO og la denne i : C:\Users\vlorj\OneDrive\Dokumenter\aks-terraform-GO
4. Inne i mappen skrev jeg : «go mod init aks-terraform-og» , denne gjør:  
   - Oppretter et nytt Go-modul i mappen du står i.

* Den lager en fil som heter go.mod.
* go.mod fungerer som en "pakke­definisjon" for prosjektet ditt, litt som package.json i Node.js eller requirements.txt i Python.
* Setter modulens navn til aks-terraform-go.
* Dette blir identiteten til prosjektet ditt.

1. Installer nødvendige **Azure SDK**-pakker :   
   - go get [github.com/Azure/azure-sdk-for-go/sdk/azcore@latest](mailto:github.com/Azure/azure-sdk-for-go/sdk/azcore@latest)  
   - go get [github.com/Azure/azure-sdk-for-go/sdk/azidentity@latest](mailto:github.com/Azure/azure-sdk-for-go/sdk/azidentity@latest)  
   - go get [github.com/Azure/azure-sdk-for-go/sdk/resourcemanager/resources/armresources@latest](mailto:github.com/Azure/azure-sdk-for-go/sdk/resourcemanager/resources/armresources@latest)  
   - go get [github.com/Azure/azure-sdk-for-go/sdk/resourcemanager/containerservice/armcontainerservice@latest](mailto:github.com/Azure/azure-sdk-for-go/sdk/resourcemanager/containerservice/armcontainerservice@latest)  
   - go mod tidy
2. Opprettet en main.go fil, og la inn dette:

package main

import (

"context"

"fmt"

"log"

"github.com/Azure/azure-sdk-for-go/sdk/azcore/to"

"github.com/Azure/azure-sdk-for-go/sdk/azidentity"

"github.com/Azure/azure-sdk-for-go/sdk/resourcemanager/resources/armresources"

"github.com/Azure/azure-sdk-for-go/sdk/resourcemanager/containerservice/armcontainerservice"

)

func main() {

cred, err := azidentity.NewDefaultAzureCredential(nil)

if err != nil {

log.Fatalf("Kunne ikke autentisere: %v", err)

}

ctx := context.Background()

subscriptionID := "<din-subscription-id>"

// Opprett resource group

rgClient, \_ := armresources.NewResourceGroupsClient(subscriptionID, cred, nil)

\_, \_ = rgClient.CreateOrUpdate(ctx, "aks-demo-rg2", armresources.ResourceGroup{

Location: to.Ptr("Norway East"),

}, nil)

fmt.Println("✅ Resource Group 'aks-demo-rg2' opprettet!")

// Opprett AKS cluster

aksClient, \_ := armcontainerservice.NewManagedClustersClient(subscriptionID, cred, nil)

cluster := armcontainerservice.ManagedCluster{

Location: to.Ptr("Norway East"),

Properties: &armcontainerservice.ManagedClusterProperties{

DNSPrefix: to.Ptr("aksdemocluster2"),

AgentPoolProfiles: []\*armcontainerservice.ManagedClusterAgentPoolProfile{

{

Name: to.Ptr("systempool"),

Count: to.Ptr(int32(1)),

VMSize: to.Ptr("Standard\_D2as\_v5"),

Mode: to.Ptr(armcontainerservice.AgentPoolModeSystem),

},

},

},

Identity: &armcontainerservice.ManagedClusterIdentity{

Type: to.Ptr(armcontainerservice.ResourceIdentityTypeSystemAssigned),

},

}

poller, \_ := aksClient.BeginCreateOrUpdate(ctx, "aks-demo-rg2", "aks-demo-cluster2", cluster, nil)

resp, \_ := poller.PollUntilDone(ctx, nil)

fmt.Printf("✅ AKS Cluster opprettet: %s\n", \*resp.Name)

}

1. logge inn hvis det ikke er blitt gjort : az login
2. også kjøre programmet : go run main.go