Inlämningsuppgift 2 - Houssam Boughdadi

Förutsättningar

Applikation

Jag kommer använda en simplifierad variant av applikationen som vi jobbat tillsammans med på Kodfredagars lektioner (Med InMemoryMongoDB). Den ligger i ett privat repo på mitt github men skriptet kan enkelt byta repo med bara 1 variabel ändring i huvudskriptet.

Installerade program och extentsions

- VS code
- Git bash
- .NET SDK
- Azure CLI (+ Inloggad på Azure via VS code, förekommer inte i denna Artikel)
- GitHub CLI (+ Inloggad på git bash "gh auth login" och följa instruktionerna)
- ARM Template Viewer (Extention för VS Code, för att se ARM template produkter)

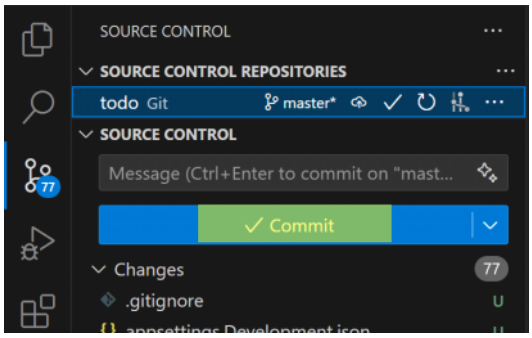
Utveckla applikationen

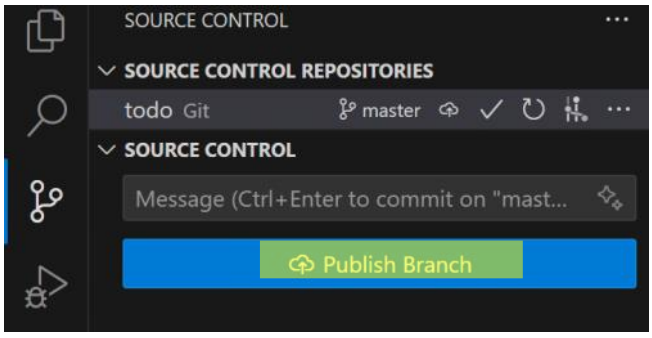
För att få en applikation till skriptet så behöver den utvecklas. Detta kommer göras i VS code här

1. I en tom mapp (mappens namn bör vara applikationens namn) öppna gitbash terminal

```
dotnet new webapp
dotnet new gitignore
git init
```

2. Publicera skriptet genom att Commit och Publish till ditt github (privat funkar)

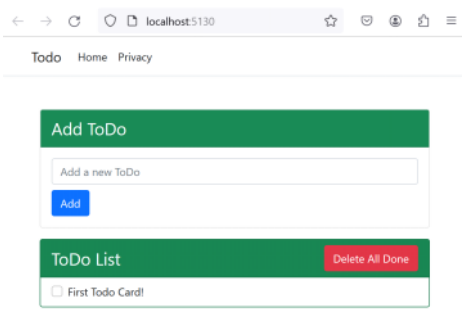




3. Fortsätt utveckla applikationen till nöje, iallafall till första Release. För att kolla applikationen under utvecklingens gång, kör följande kommando.

```
dotnet run watch
```

En klar webbapplikation kan se ut såhär



Provisionering

Inom själva provisioneringen så har jag gjort ett shellscript som kallar på andra shellscript med parametrar för att kunna modulärt byta ut olika delar av scriptet som behöver utvecklas eller ändras. Huvudskriptet ligger i "root" av hela provisioneringsmappen och de andra "pusselbitarna" som den kallar på är mer ihopsamlade inom olika mappar.

provision.sh (Huvudskriptet)

Huvudskriptet sätter namnet på repot i github och under vems namn repot ligger på som variabler som kan ändras beroende på vilken applikation man vill ha igång på sitt VM.

Skriptet sen ber ett annat skript att göra en Resource Group och returnerna vilket namn gruppen fått (Finns även implementationen att ta emot Custom namn för Resource Gruppen som går att skicka in som en parameter)

När Resource Group är gjord så körs ett tredje skript som ska deploya provisionen. Det skriptet tar in argumenten applikationsnamnet och namnet på resource group som blivit hämtad i tidigare steg. Skriptet kommer returnera Publika IP som når appen efter att den blivit deployed.

Huvudskriptet kallar på github cli kommando som bygger artifakterna och när en actions-runner är igång så plockas artifakterna upp till applikationen.

Slutligen skiver skriptet ut i terminalen vilket IP appen nås via (kan ta någon minut för applikationen att dyka upp)

Huvudskriptet finns skriven här:

```
#!/bin/bash
app_name="APPLICATION-NAME"
gh_user="REPOSITORY-OWNER"
resource_group=$(./ResourceGroup/resource_group_provision.sh)
public_ip=$(./Deployment/deployment_provision.sh $resource_group $app_name)
gh workflow run cicd.yaml --repo $gh_user/$app_name --ref master
echo "APP IP: $public_ip"
```

(Klicka på filpatherna för att ta dig till dess kod)

ResourceGroup\resource_group_provision.sh

Detta skript tar emot ett argument och sätter namnet på Resource gruppen till det eller tar inte emot något argument och därmed generera ett namn för Resource gruppen. Namnet blir en siffra som är lika med antalet exiterande Resource grupper + 1 för att inte få överlappande av siffror/namn på Resource grupper om flera grupper existerar med detta system av namngivning.

Efter att den gjort Resource Group så enkapslar det svaret i en response variabel för att förhindra problem av returnering i terminalen vid använding av echo. Sedan returnerar/echo skriptet variabeln för Resource Group namnet.

```
#!/bin/bash
if [ -z "$1" ]; then

    resource_groups=$(az group list --query "[].name" -o tsv)

    new_rg_number=$(( $(echo "$resource_groups" | wc -w) + 1))
    resource_group="$new_rg_number"
else
    resource_group="$1"
fi
response=$(az group create --location swedencentral --name $resource_group)
echo $resource_group
```

Deployment/deployment_provision.sh

Drifta Applikationen

Kodsnuttar

Tankar och Framtida Utvecklingsmöjligheter

IMPLEMENTERA ATT KUNNA VÄLJA REPO UTAN ATT BEHÖVA ÄGA DEN