

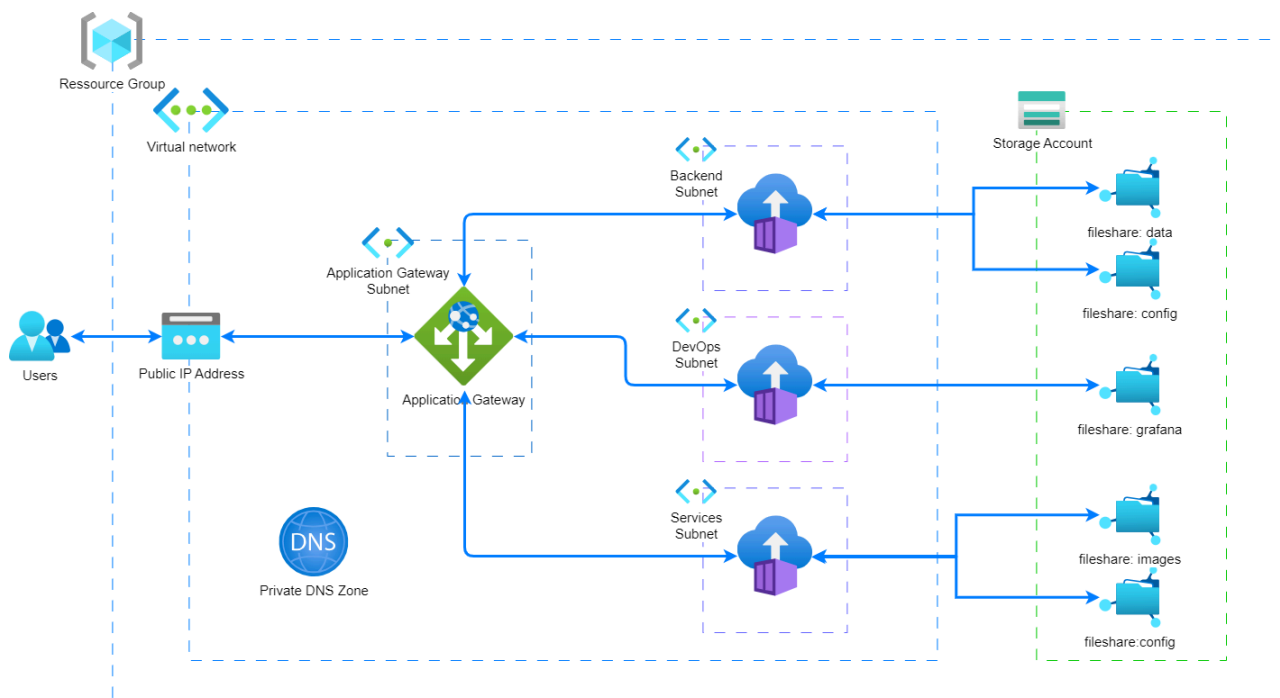
Azure Microservice Miljø med Bicep scripts

Version	Dato	Forfatter	Bemærkninger
1.2	15-05-2024	HNRK	Nyt deployment diagram
1.1	11-12-2023	HNRK	Rettelser
1.0	25-05-2023	HNRK	Første udkast

Introduktion

Denne guide beskriver hvordan et mindre microservice produktionsmiljø kan oprettes i Azure med en lille samling *Bicep* scripts.

Miljøet kan ses i diagrammet herunder:



Dette base-miljø indeholder følgende infrastrukturkomponenter til at understøtte en microservice-løsning:

- Et **virtuelt netværk** (VNET) med 4 subnetværk
- En **privat DNS zone** til at lokalisere miljøets komponenter/services
- En **offentlig IP adresse** til at tilgå hele løsningen fra internettet
- En **Azure Applikations Gateway** til at fordele trafikken fra den offentlige IP adresse på de 3 subnetværk
- En **Azure Storage Account** som base for 5 file shares (anvendt til container volumens)
- En **Azure Container Gruppe** med DevOps-komponenterne: **Loki** og **Grafana**
- En **Azure Container Gruppe** med Backend-komponenterne **MongoDB** og **RabbitMQ**

**Bemærk:**

Diagrammet herover indeholder en ekstra **Azure Container Gruppe** kaldet *Services*. Denne komponent er ikke en del af løsningen som leveres med Bicep-scriptene.

A. Opbygning af miljøet

Deployment-løsningen består af et hoved BICEP-script, kaldet **auctionsGO.bicep**. Fra dette script konfigureres og tilpasses de specifikke komponentparametre (så som jeres egne specifikke resourcenavne mm.). *Denne tilpasning udføres i toppen af scriptet.* Dokumentation for de enkelte komponentparametre findes i selve hoved-scriptet og er ikke gengivet her!

Når scriptet eksekveres, kalder det ud til nogle **submodul** Bicep-scripts for at oprette de enkelte komponenter i løsningen.

**Bemærk**

Nogle submodul-scripts opretter mere end én resource.

Modulscriptene er opbygget med følgende indehold:

Modul	filnavn	Beskrivelse
Hoved	auctionsGO.bicep	Top-niveau script kalder de andre modulscripts, beskrevet herunder. Opretter til sidst i forløbet en Azure Application Gateway baseret på output fra de andre moduler.
Network	networkGO.bicep	Opretter følgende komponenter:
		- Virtuelt netværk med 3 subnetværk:
		- <i>applicationGatewaySubnet</i> : prefix 10.0.0.0/24
		- <i>goBackendSubnet</i> : prefix 10.0.1.0/24
		- <i>goDevopsSubnet</i> : prefix 10.0.2.0/24
		- Statisk Public IP Address med DNSLabel
		- Privat DNS Zone (DNS Server) for det virtuelle netværk
Storage	storageGO.bicep	Oprette en Azure Storage konto med et unikt genereret id.
		Opretter også 3 file shares som kan anvendes til container volumes. De 3 shares navngives i hoved-scriptet.
Backend	backendGO.bicep	Azure Container Gruppe, som ligger på subnetværket <i>goBackendSubnet</i> og med IP-adressen 10.0.1.4. Gruppen indeholder følgende container-images:
		- MongoDB som lytter på port 27017
		- RabbitMQ som lytter på portene 15672 og 5672
DevOps	devopsGO.bicep	Azure Container Gruppe, som ligger på subnetværket <i>goDevopsSubnet</i> og med IP-adressen 10.0.2.4. Gruppen indeholder følgende container-images:
		- Loki
		- Grafana



Bemærk

Applikationerne i container-grupperne, følger muligvis **IKKE** jeres egen struktur eller krav, så de skal tilpasses derefter.

B. Fremgangsmetode

For at oprette ovennævnte miljø skal de følgende trin udføres med **az-cli-værktøjet** fra en terminal, (meget gerne Bash):

1. Start med at logge ind med:

```
$ az login
```

2. Opret en miljøvariabel med navnet på en ressourcegruppe til jeres produktionsmiljø i Azure (vælg jeres eget navn!):

```
$ export ResourceGroupName=AuktionshusetGOrg
```

3. Opret ressourcegruppen i **Azure**:

```
$ az group create --name $ResourceGroupName --location northeurope
```

4. Åben filen **auctionsGO.bicep** og tilpas variablerne i området markeret med:

```
// --- Konfigurerbare parametre ---
```

5. Kør top-scriptet med kommandoen herunder. Det tager ca. 5-10 minutter første gang det køres:

```
$ az deployment group create --resource-group $ResourceGroupName \
  --template-file auctionsGO.bicep --verbose
```

C. Noter

Note 1 - Application Gateway

Det er kun muligt at sende HTTP og HTTPS data igennem Application Gateway'en, så derfor er det ikke muligt at tilgå de services udefra, som ikke understøtter denne protokol. Dvs. f.eks. at **MongoDB** kan ikke tilgås med MongoDB Compass.

Note 2 - MongoDB

Har man behov for at tilgå MongoDB kan man fra Azure Portalen navigere ind i **backend-containergruppen** og vælge **Containers** fra menupunktet **Settings**. Herefter vælges **mongodb** fra listen af containers øverst og derefter **Connect**-fanebladet nederst. Vælg herefter at starte `"/bin/bash"` terminal, efterfulgt af et klik på **Connect-knappen**.

Det er nu muligt at køre Mongo Shell (MongoDBs klient værktøj) fra denne terminal, med kommandoen **mongosh**. Følg denne link for hurtig guide til *mongosh*:

- [Mongo shell commands to get you through the day](#)
- [MongoDB Create a User](#)
- [db.auth\(\)](#) (login i mongosh)

Note 3 - Økonomi

En løsning med en Applikation gateway og 3 container grupper kan godt koste omkring 30-50kr. per dag. Så det er en rigtig god idé ikke at køre sit **Azure Produktionsmiljø** hele tiden.

For at starte og stoppe miljøet er der vedlagt 2 bash-scripts: [startup.sh](#) og [shutdown.sh](#) i BICEP-løsningen. Husk at tilpasse dem så de kalder de rigtige ressourcer i jeres egen løsning.