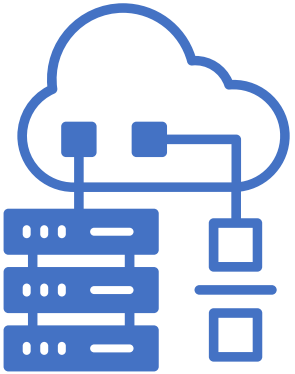


Infrastructure-as-Code Med Bicep



Infrastruktur som kode (IaC)



- **Definition:** *Infrastruktur-som-Kode* (IaC) refererer til praksissen med at administrere og levere IT-infrastruktur ved hjælp af de samme **principper og teknikker som softwareudvikling**.
- **Formål:** Automatisering af infrastrukturforvaltning, ensartet og reproducerbar implementering, **fejlreduktion** og **versionsstyring**.
- **Fordele:**
 - Hurtigere implementering
 - Mindre fejl og ensartet konfiguration
 - Effektiv ressourceudnyttelse
 - Let at versionere og vedligeholde

Hvad er Azure Bicep?



- Azure Bicep er et domænespecifikt sprog (DSL) til infrastruktur som kode (IaC) på Azure.
- Formålet er at erstatte JSON-skabeloner med et mere læseligt og vedligeholdbart sprog.
- Bygger på JSON, men med forbedringer i læsbarhed og genbrugelighed.

[VS Code Azure Bicep extension](#)

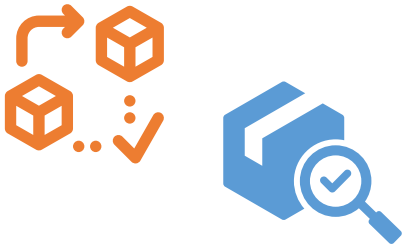
Hvorfor bruge Azure Bicep?



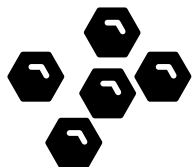
- **Øget læsbarhed:** Bicep-brugere oplever forbedret kodeforståelse sammenlignet med JSON.



- **Genbrugelighed:** Moduler og parametre letter genbrug af kodeskabeloner.



- **Valideringscheck:** Indbyggede kontrolmekanismer for at reducere fejl under *deployments*. Hurtigere fejlfinding med mere forståelige fejlmeddelelser.



- **Modulært:** Opdeling af komplekse skabeloninstallationer i mindre modulfiler og henviser til dem i en hovedskabelon.



- **Aktivt community:** Støttes af et voksende miljø af brugere og bidragydere.

Azure Bicep-kodeeksempel

// Eksempel på Azure Bicep-kode for oprettelse af en Storage-konto

```
param storageAccountName string
param location string = 'East US'
```

Parametre: Brug af parametre til at gøre koden mere fleksibel.

```
resource storageAccount 'Microsoft.Storage/storageAccounts@2021-06-01' = {
  name: storageAccountName
  location: location
  sku: {
    name: 'Standard_LRS'
  }
  kind: 'StorageV2'
}
```

Ressource-definition: Enkel syntax til at definere en Storage-konto.

```
output storageAccountConnectionString string = storageAccount.primaryConnectionString
```

Output: Definition af en output-variabel til at hente forbindelsesstrengen til den oprettede Storage-konto.

Bicep: Container Group eksempel - PART 1

main.bicep

```
param name string = 'qgtcatalogservice'

param location string = resourceGroup().location

param image string = 'hnrk/catalog-svc:latest'

param port int = 80

param cpuCores int = 1

param memoryInGb int = 1

@allowed([
  'Always'
  'Never'
  'OnFailure'
])
param restartPolicy string = 'Always'
```



(fortsætter ...)

Bicep: Container Group eksempel – PART 2

main.bicep

```
resource containerGroup 'Microsoft.ContainerInstance/containerGroups@2021-09-01' = {  
  name: name  
  location: location  
  properties: {  
    containers: [  
      ...  
    ]  
    osType: 'Linux'  
    restartPolicy: restartPolicy  
    ipAddress: {  
      type: 'Public'  
      ports: [  
        {  
          port: port  
          protocol: 'TCP'  
        }  
      ]  
    }  
  }  
}  
  
output containerIPv4Address string = containerGroup.properties.ipAddress.ip
```

```
{  
  name: name  
  properties: {  
    image: 'hnrkjnsn/catalog:1.0'  
    environmentVariables: []  
    ports: [  
      {  
        port: port  
        protocol: 'TCP'  
      }  
    ]  
    resources: {  
      requests: {  
        cpu: cpuCores  
        memoryInGb: memoryInGb  
      }  
    }  
  }  
}
```

localhost

Deployment med Azure-cli (az)

Fra en terminal:

```
$ az group create --name exampleRG --location eastus  
$ az deployment group create --resource-group exampleRG --template-file main.bicep
```

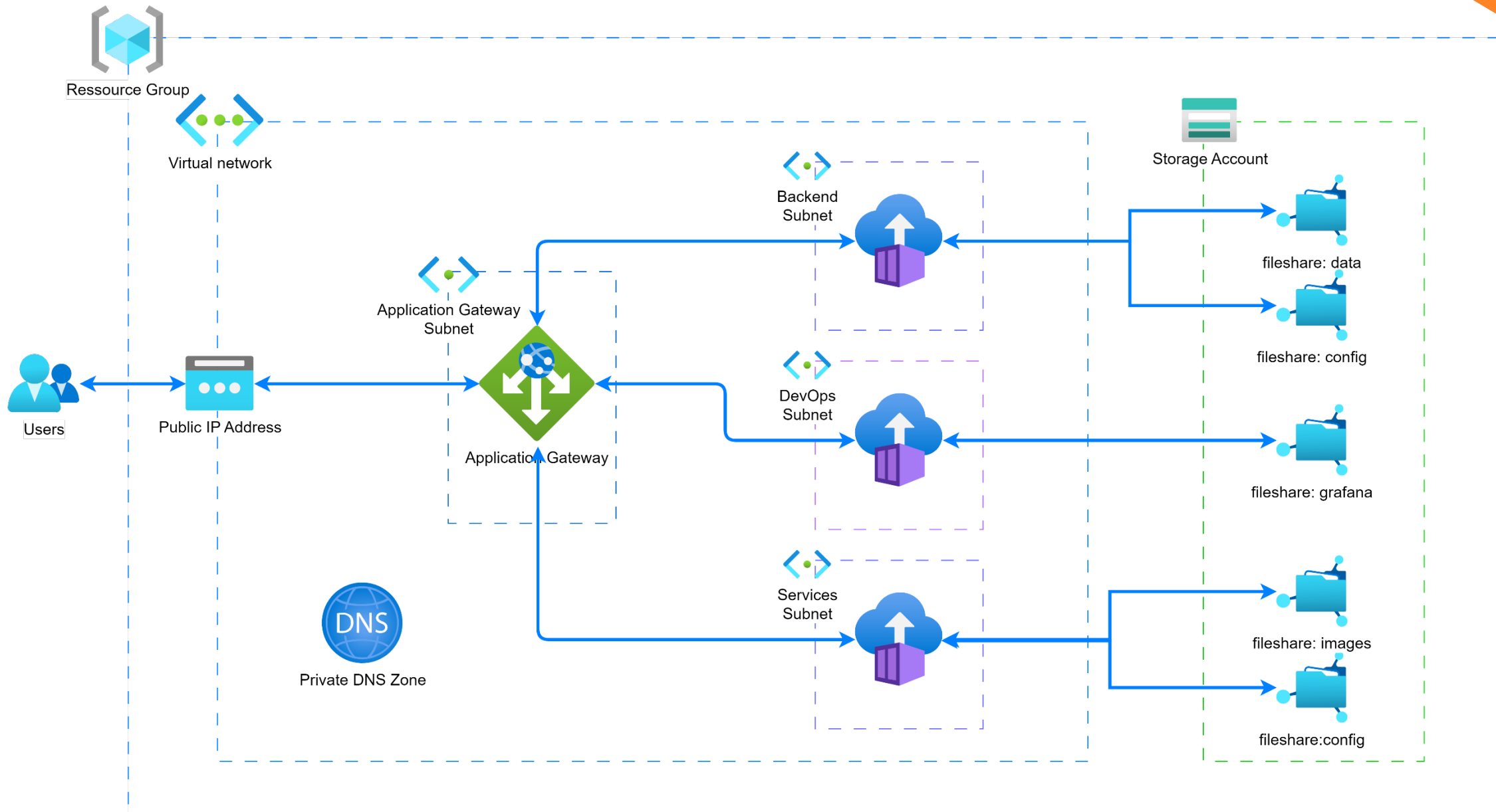
Review

```
$ az resource list --resource-group exampleRG
```

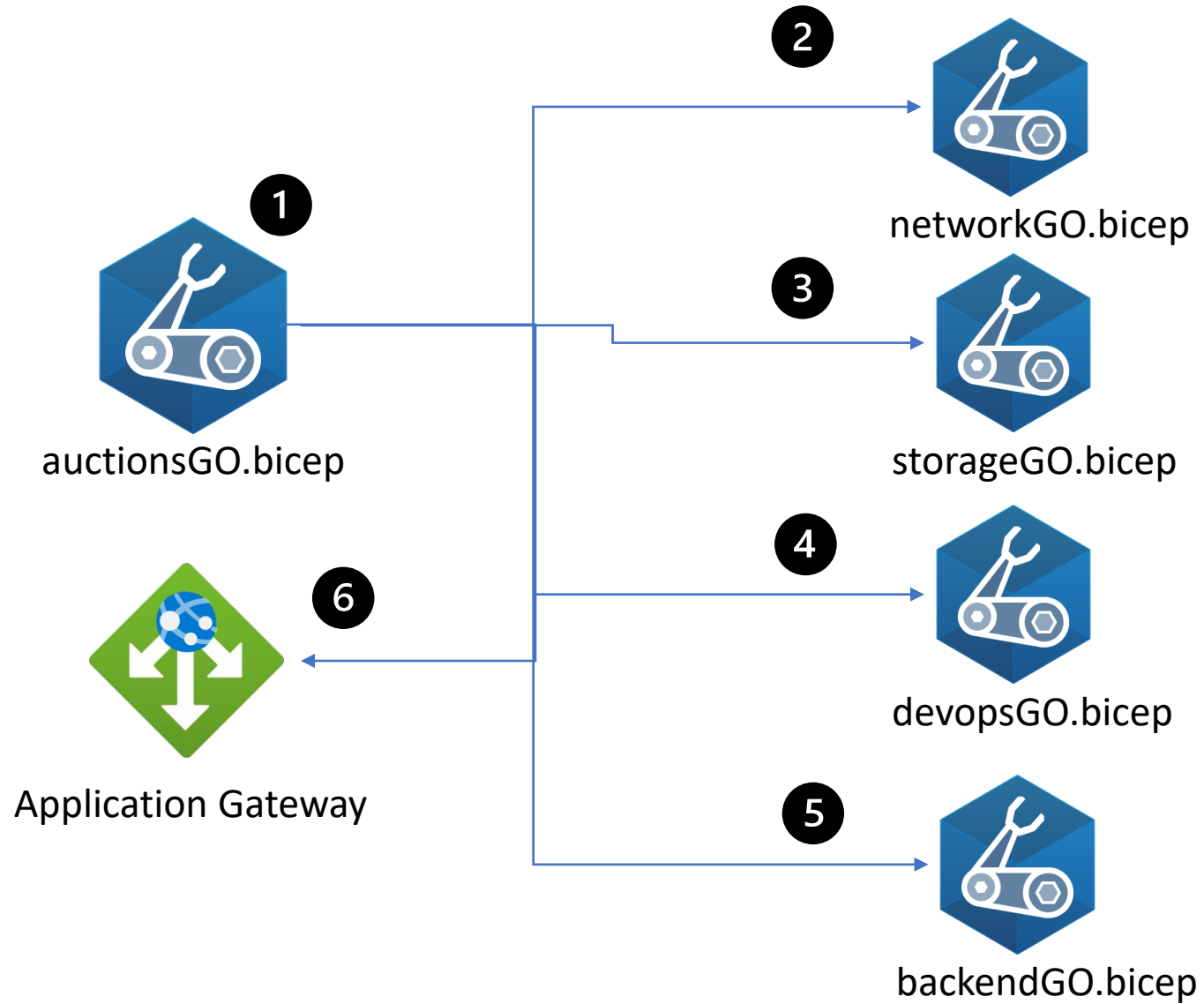
```
$ az container logs --resource-group exampleRG --name qgtcatalogservice
```


Auktionshuset

Deployment



Bicep scripts



Start script

```
#!/bin/bash
```

```
export RESGROUP=go-auctionhouse-rg  
export GATEWAYNAME=go-auctionhouse-appgateway
```

```
echo "Starting Azure container groups ..."
```

```
az container start --name auktionsHusetServicesGroup --resource-group $RESGROUP  
az container start --name auktionsHusetBackendGroup --resource-group $RESGROUP  
az container start --name auktionsHusetDevOpsGroup --resource-group $RESGROUP
```

```
echo "Starting Azure Application Gateway ..."
```

```
az network application-gateway start -g $RESGROUP -n $GATEWAYNAME
```

Shutdown script

```
#!/bin/bash
```

```
RESGROUP=AuktionsHusetRG
```

```
GATEWAYNAME=goauctionsAppGateway
```

```
az container stop --name auktionsHusetServicesGroup --resource-group $RESGROUP
```

```
az container stop --name auktionsHusetBackendGroup --resource-group $RESGROUP
```

```
az container stop --name auktionsHusetDevOpsGroup --resource-group $RESGROUP
```

```
az network application-gateway stop -g $RESGROUP -n $GATEWAYNAME
```