May 5th – 7th,2022

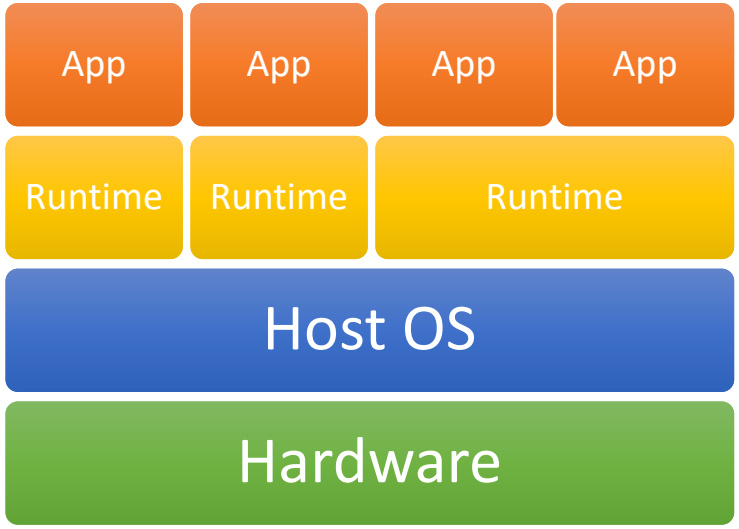
#GlobalAzure

Nuestro patrocinadores

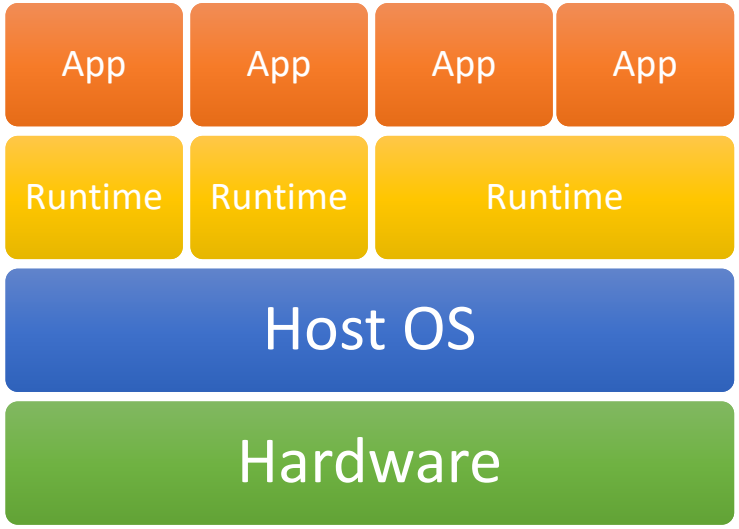
Colabora

ALL COMMANDS AND EVENTS IN THIS SHOW -- EVEN THOSE BASED ON REAL ONES -- ARE ENTIRELY FICTIONAL. ALL CRAPY SLIDES ARE EXPLAINED.....POORLY. THE FOLLOWING SESSION CONTAINS COARSE LANGUAGE AND DUE TO ITS CONTENT IT SHOULD NOT BE VIEWED BY ANYONE

Traditional environment



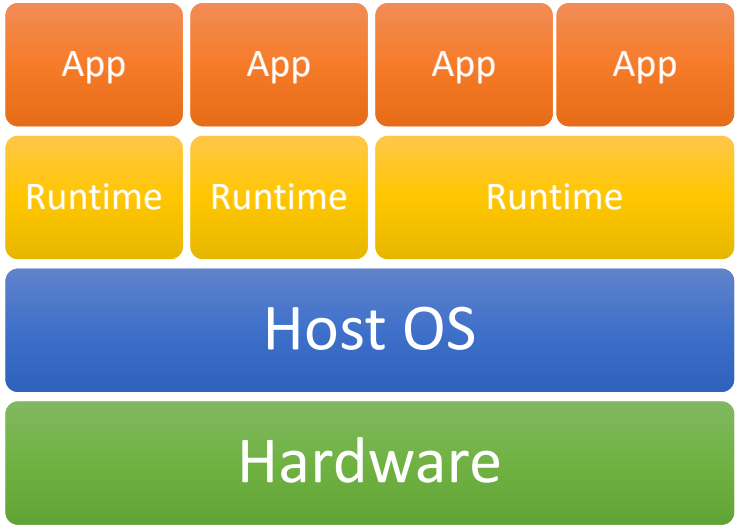
Traditional environment

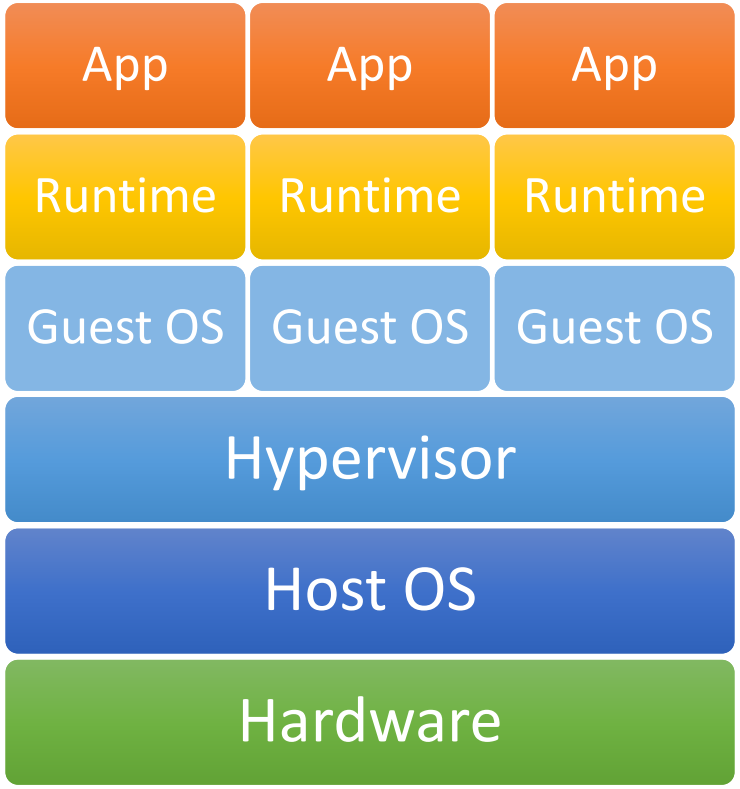


Problems:

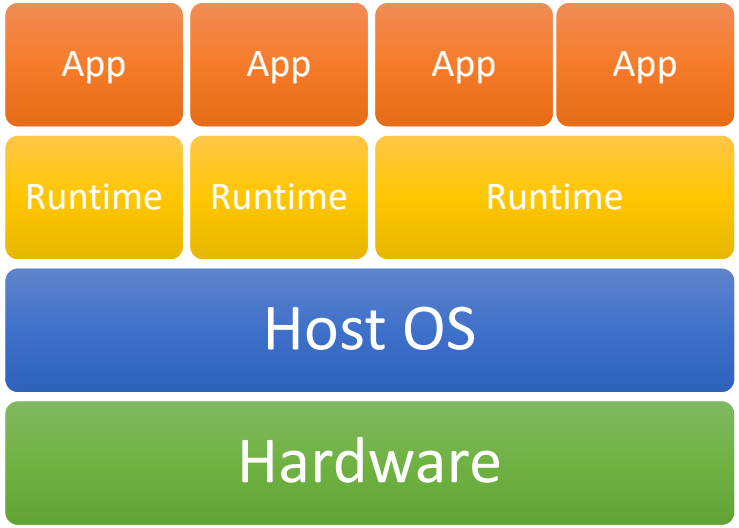
- Shared environment variables
- Can not have different versions of the same runtime
- Can not scale applications automatically as the resource is static

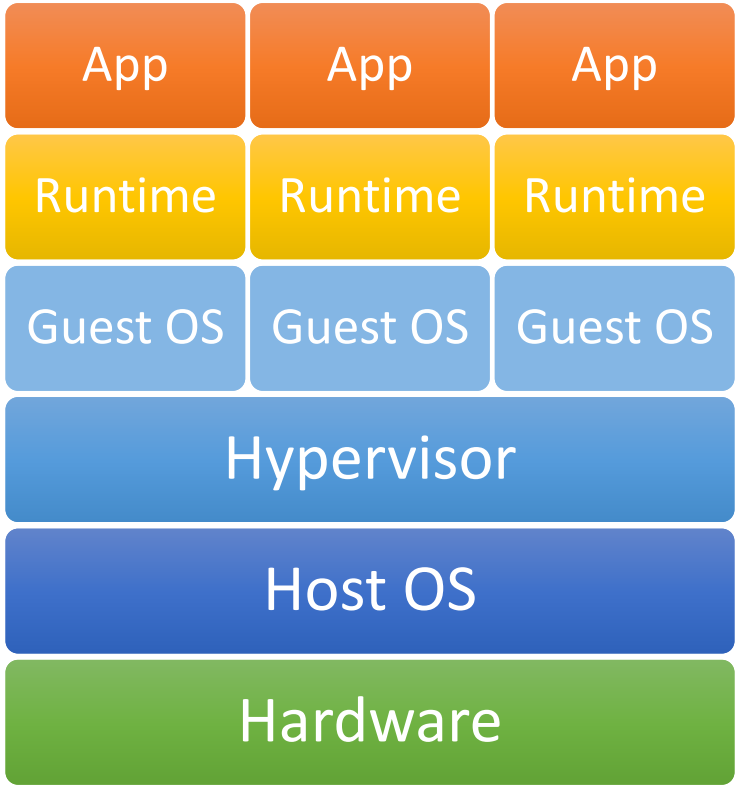
Virtual environment





Virtual environment

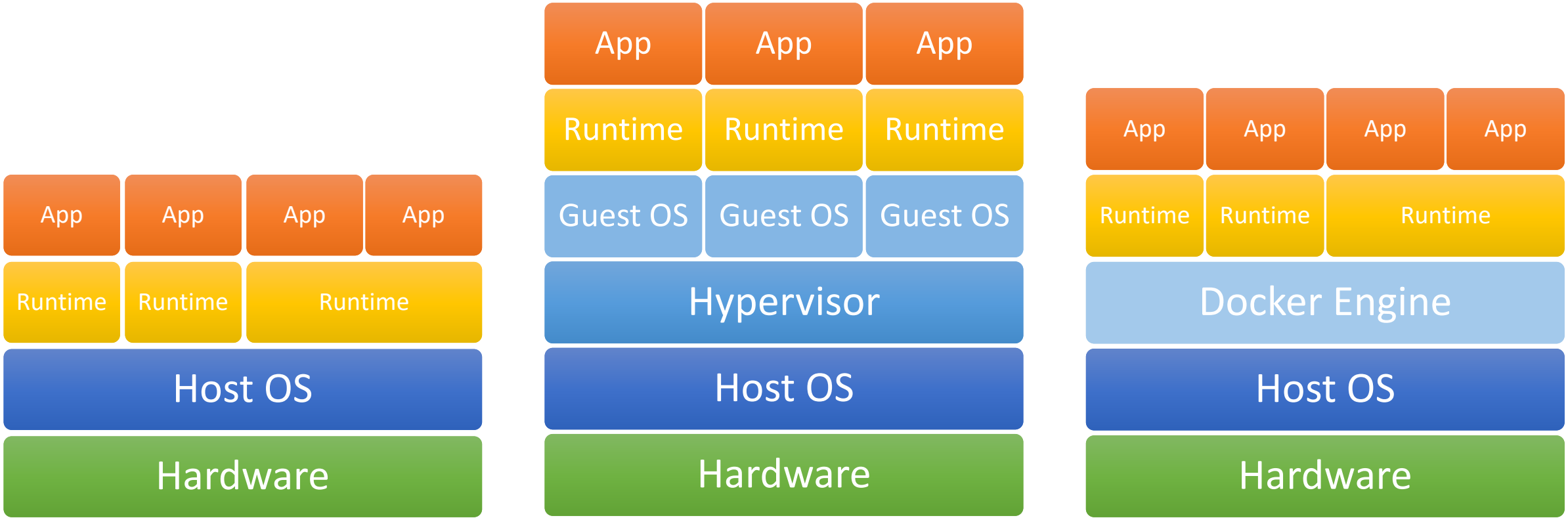




Problems:

- 1 VM per App = overkill
- Shared environments variables in the same VM
- Slow scalation as the VM resources are static
- Resources are wasted

Containers environment

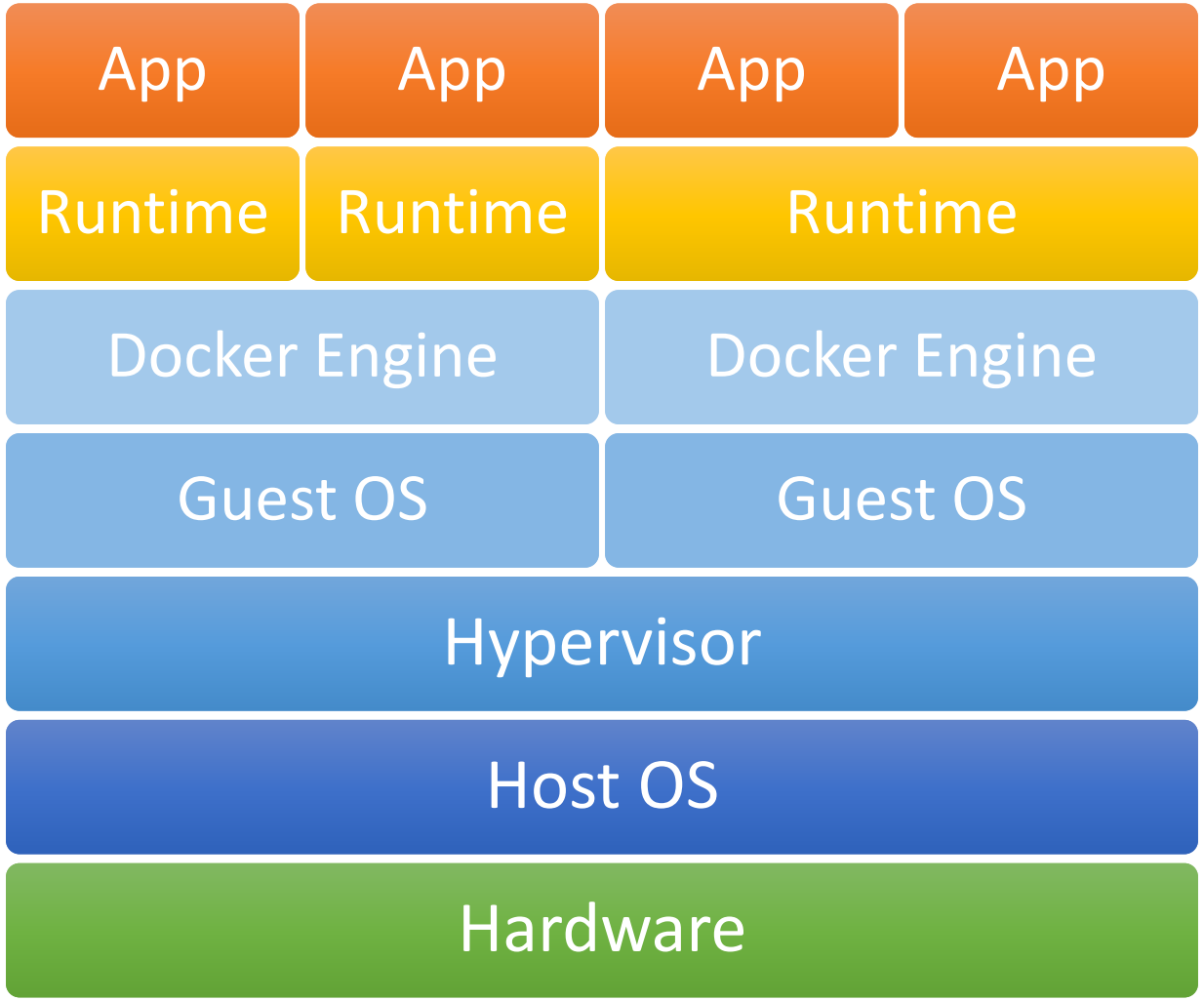


```
dotnet new sln -n ContainerNet6
dotnet new webapi -o MyApi
dotnet sln add MyApi/MyApi.csproj
```

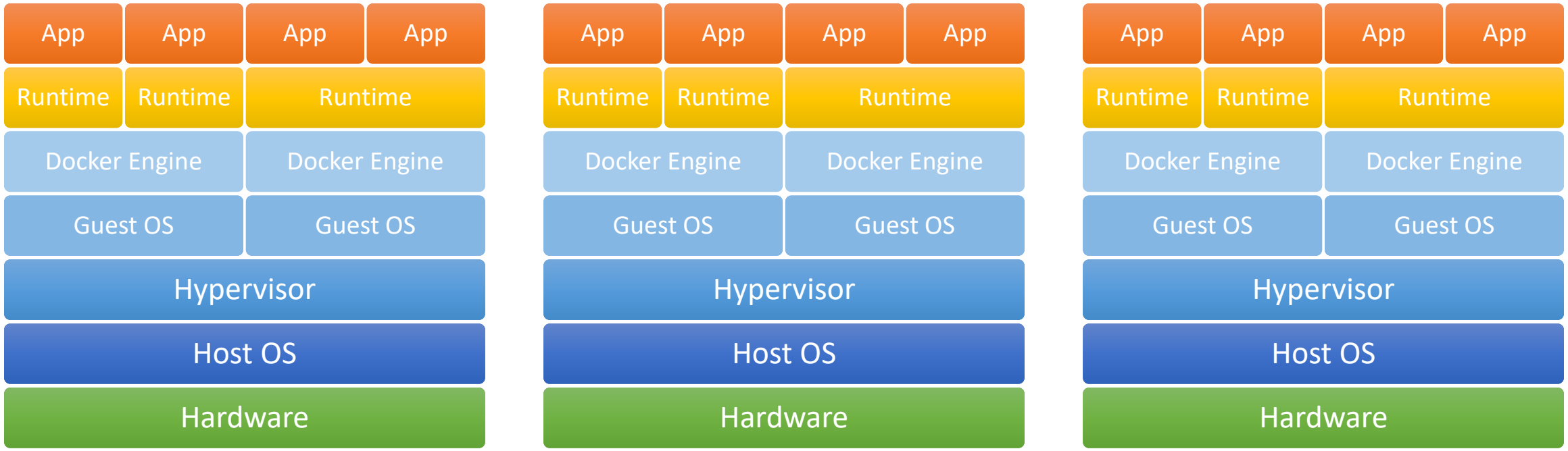
Dockerfile

```
FROM mcr.microsoft.com/dotnet/sdk:6.0 AS build
WORKDIR /src
COPY . .
RUN dotnet publish "MyApi/MyApi.csproj" -c Release -o /app

FROM mcr.microsoft.com/dotnet/aspnet:6.0
WORKDIR /app
COPY --from=build /app ./
ENTRYPOINT ["dotnet", "MyApi.dll"]
```

Actual environment



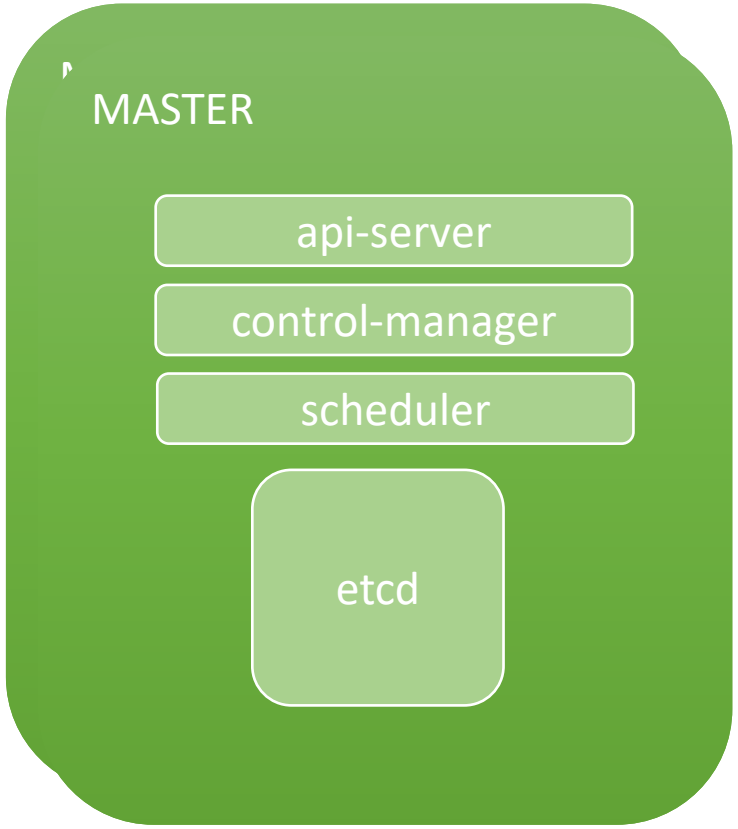
Actual environment

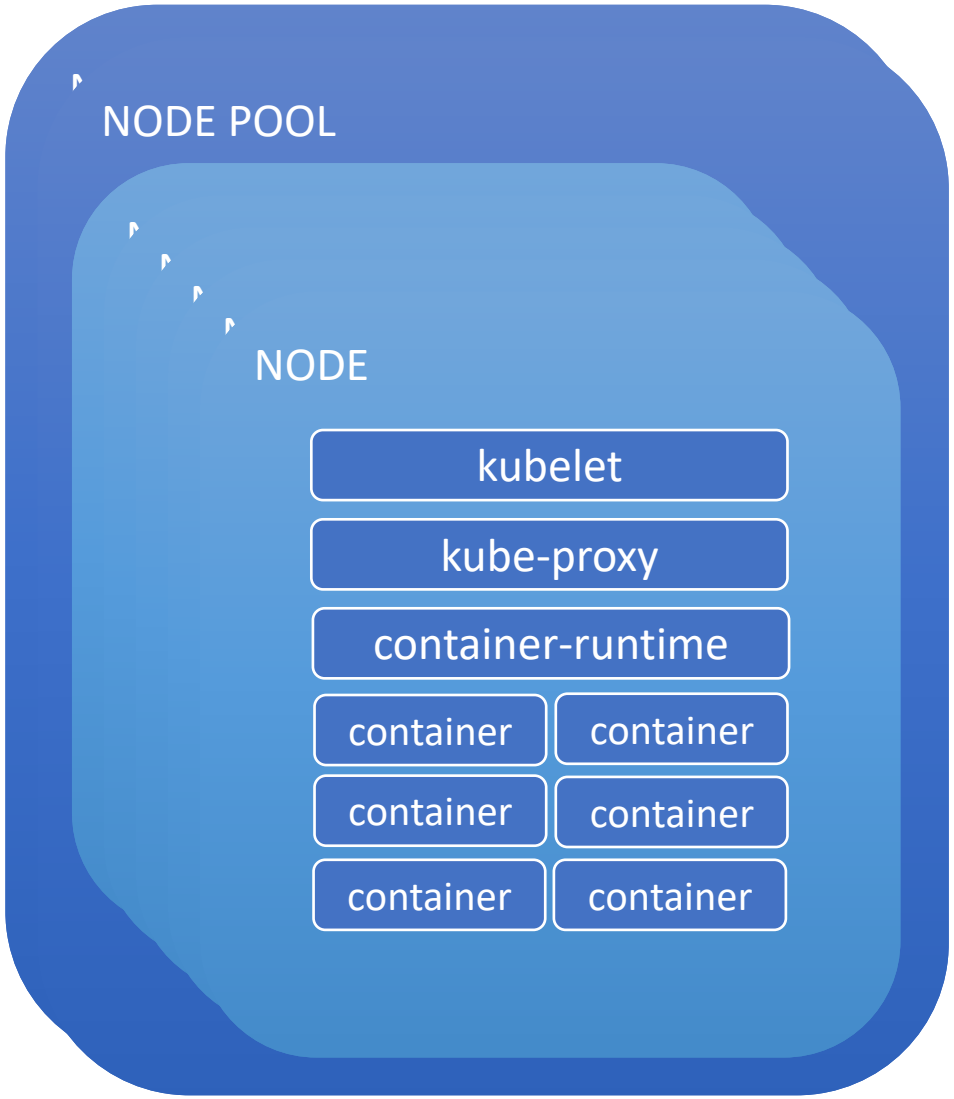


Kubernetes = K8S

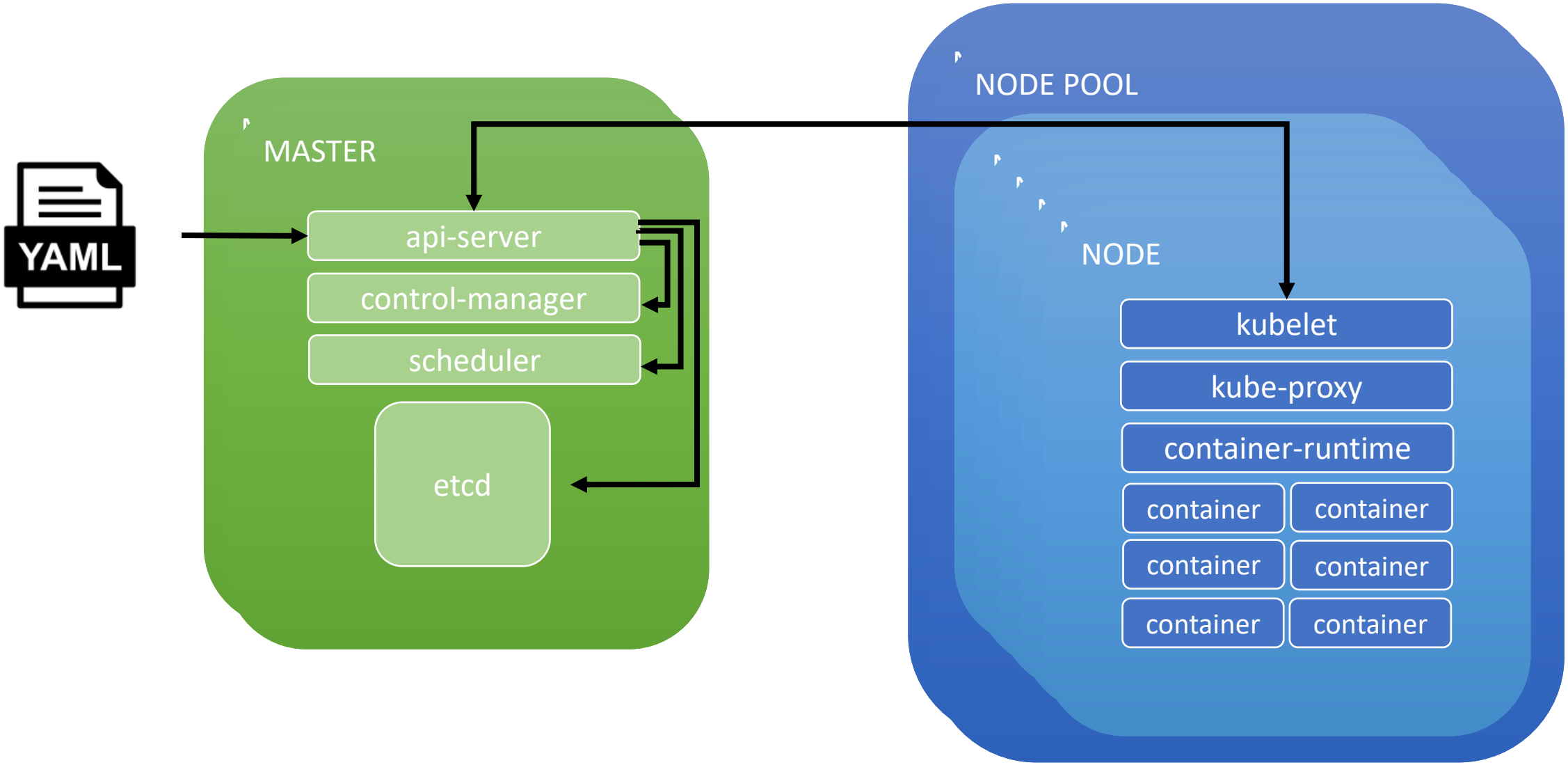
- **Service Discovery** and **Load Balancing**: Kubernetes can expose a container using the DNS name or using their own IP address.
- **Self-healing**: Auto healing is a great feature that Kubernetes provides — it restarts, kills, and replaces containers that fail.
- Automated **Roll outs** and **Rollbacks**: Micro service systems could include hundreds, if not thousands, of services which can be hard or impossible to spin up manually. With this feature, you’re able to specify the desired state of a given application (deployment) and Kubernetes will do the work to make sure to achieve this state.
- **Secret/Config** Management: This allows you to store config and sensitive data like passwords, tokens and SSH keys.
- Auto **Resource Management**: Specify the resource, RAM and CPU, needed for your deployments, and Kubernetes will distribute containers to relevant nodes, and fit them for optimal use of machine resources.
- **Storage Orchestration**: Kubernetes allows you to automatically mount a storage system of your choice, such as local storage or from public cloud providers.

K8S


MASTER
api-server
control-manager
scheduler
etcd


NODE POOL
NODE
kubelet
kube-proxy
container-runtime
container
container
container
container
container
container

K8S


YAML
MASTER
api-server
control-manager
scheduler
etcd
NODE POOL
NODE
kubelet
kube-proxy
container-runtime
container
container
container
container
container
container

Azure Kubernetes Service = AKS

- **Managed Kubernetes Cluster:** Azure Kubernetes Service (AKS) offers serverless Kubernetes, an integrated continuous integration and continuous delivery (CI/CD) experience, and enterprise-grade security and governance. Unite your development and operations teams on a single platform to rapidly build, deliver, and scale applications with confidence.

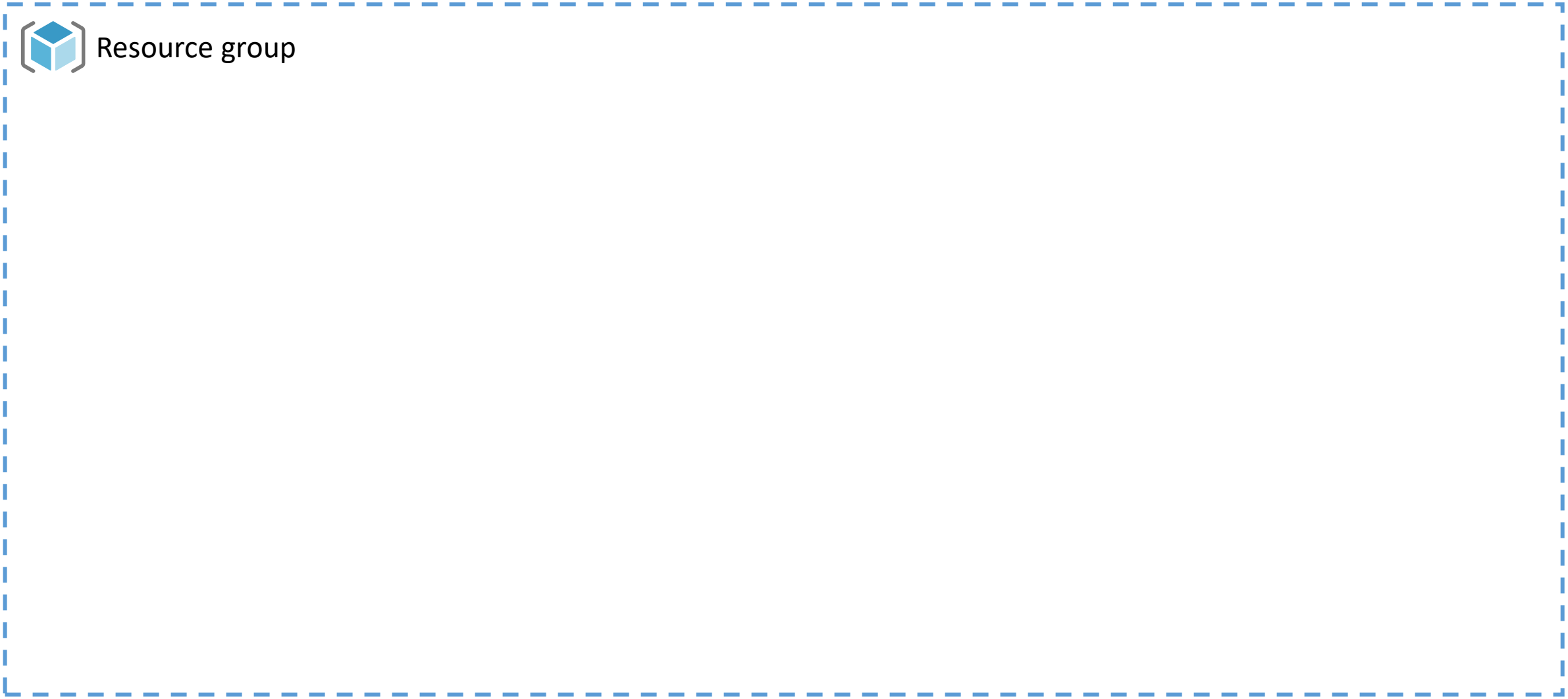
- **Elastic provisioning** of capacity without the need to manage the infrastructure and with the ability to add event-driven autoscaling and triggers.

- Faster end-to-end development experience through **Azure Kubernetes tools**.

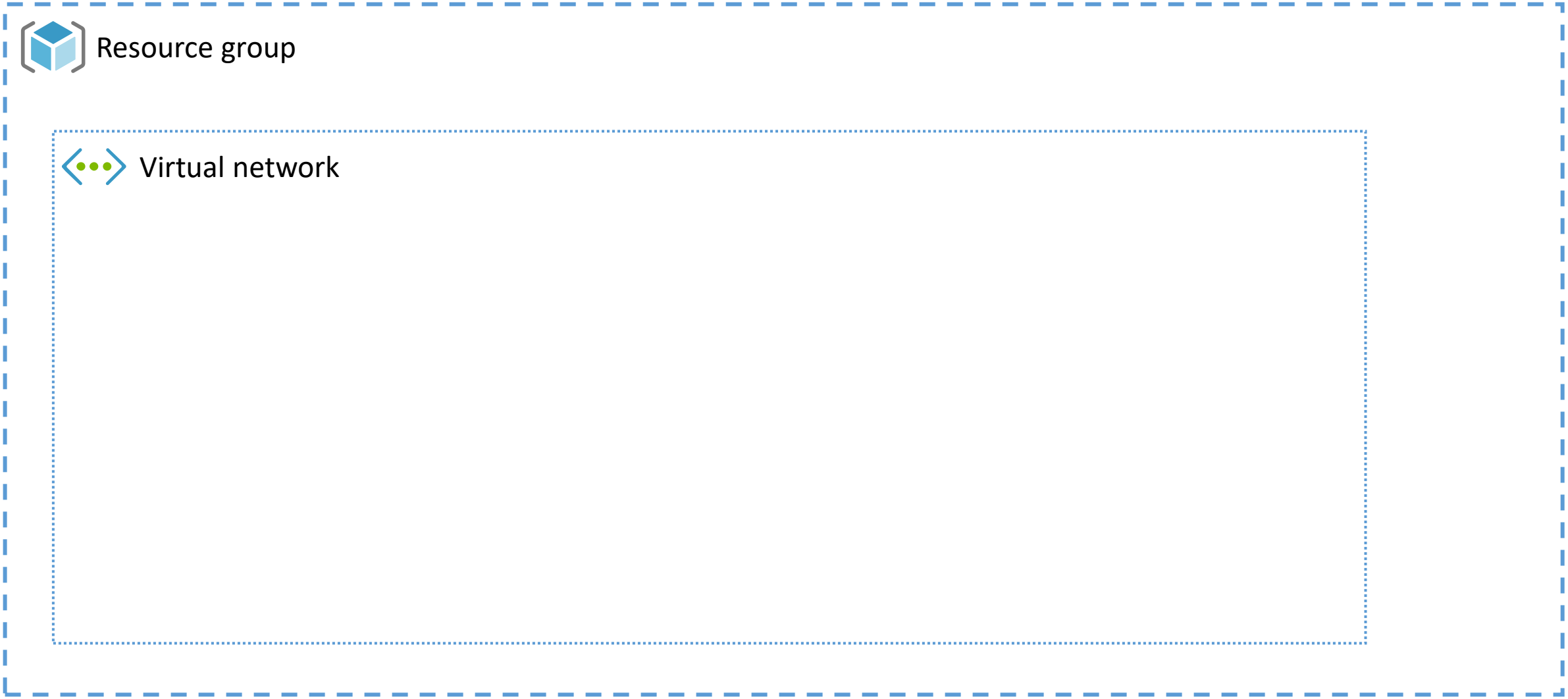
- **Most comprehensive authentication and authorization** capabilities using Azure Active Directory, and dynamic rules enforcement across multiple clusters with Azure Policy.

- Availability in **more regions** than any other cloud provider.

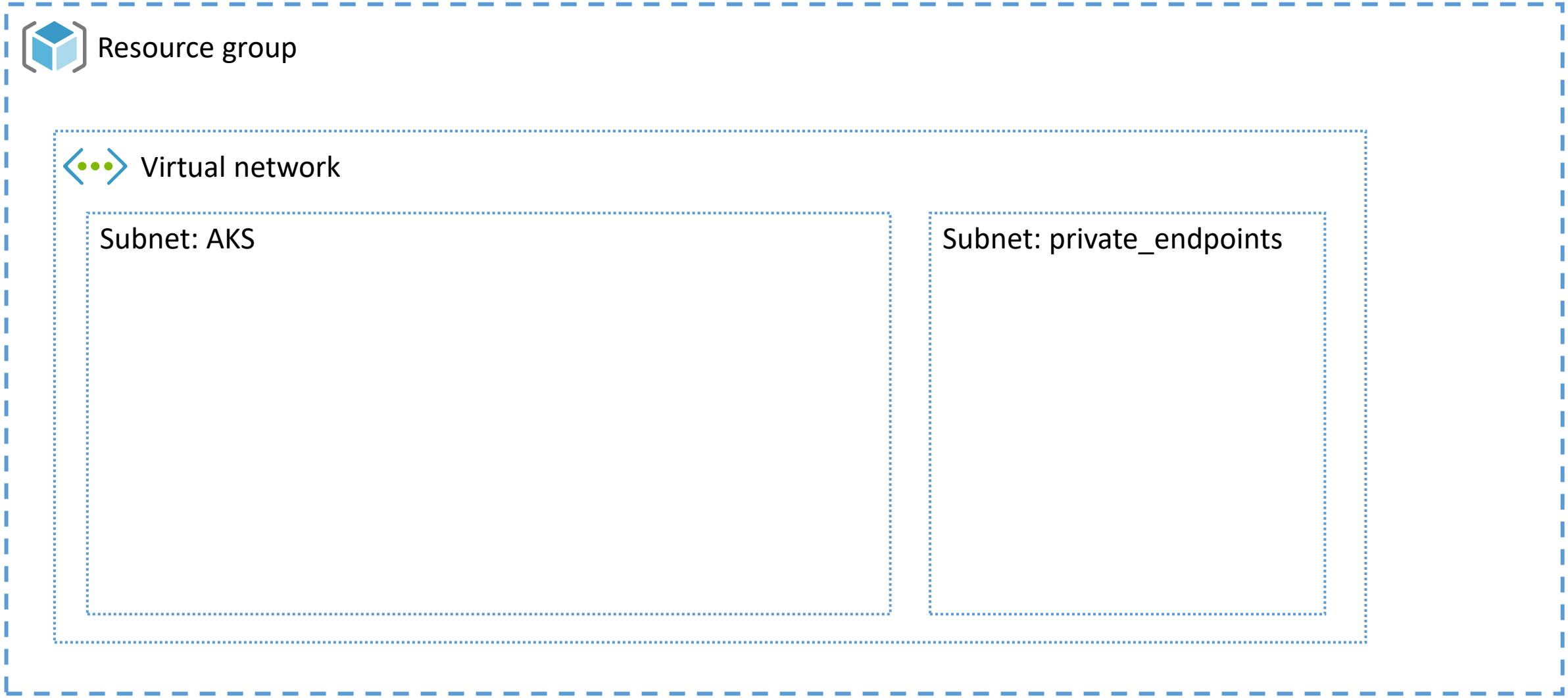
AKS



AKS



AKS


Resource group
Virtual network
Subnet: AKS
Subnet: private_endpoints

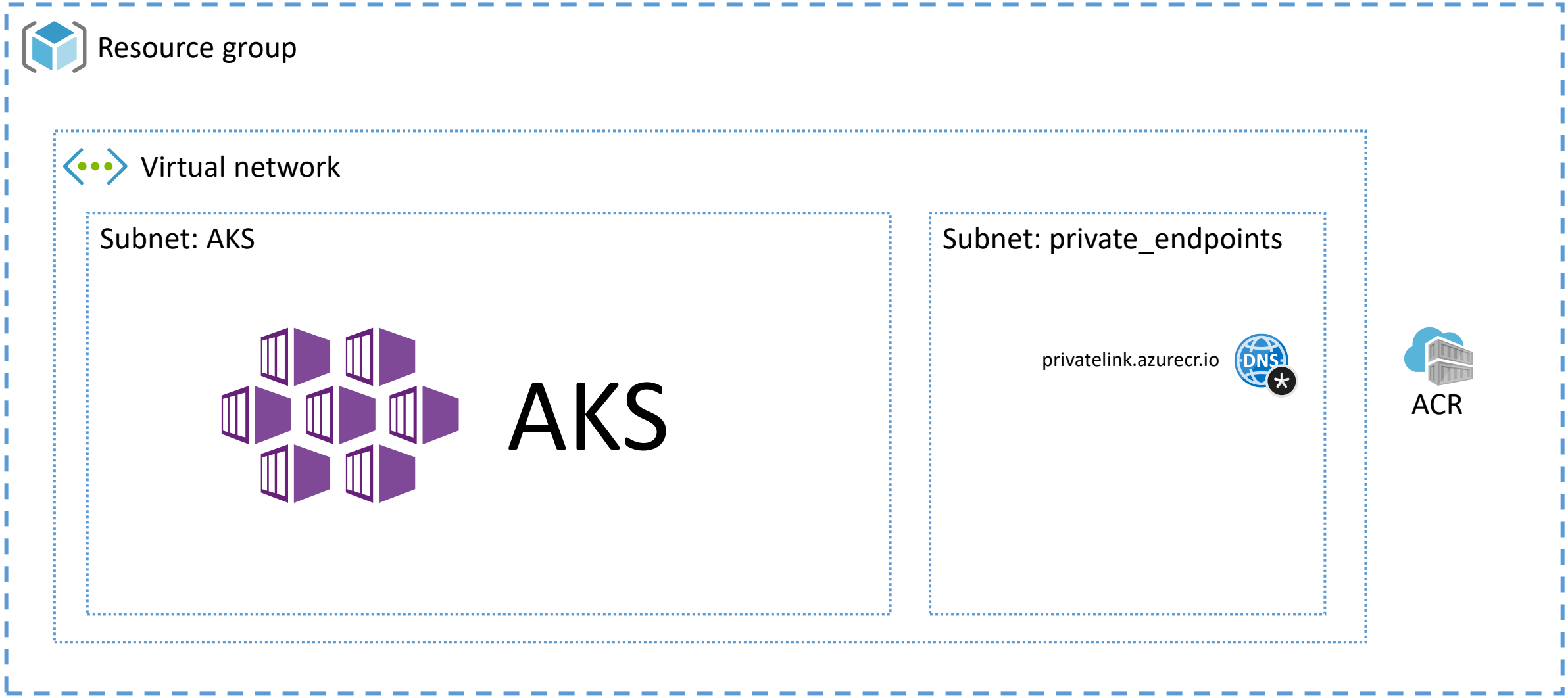
AKS


Resource group
Virtual network
Subnet: AKS
AKS
Subnet: private_endpoints

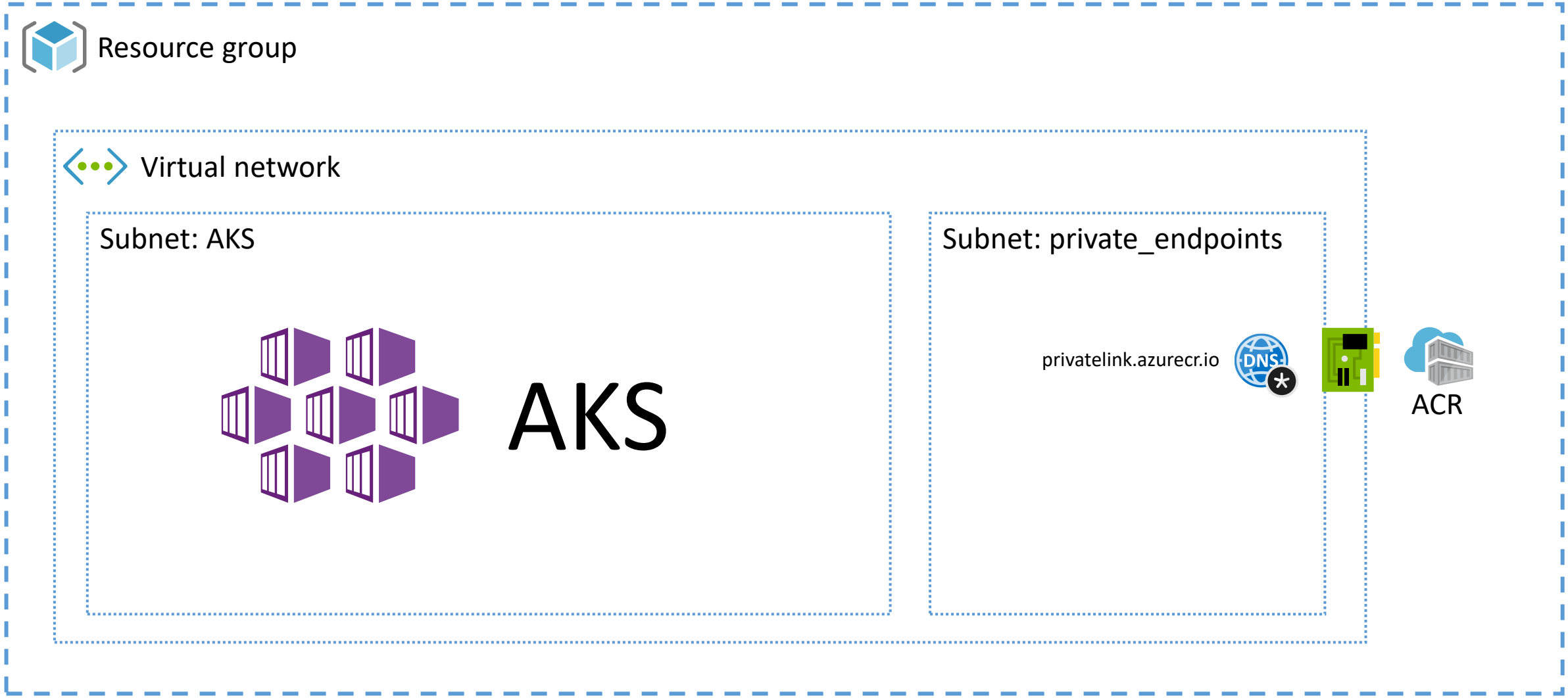
AKS


Resource group
Virtual network
Subnet: AKS
AKS
Subnet: private_endpoints
ACR

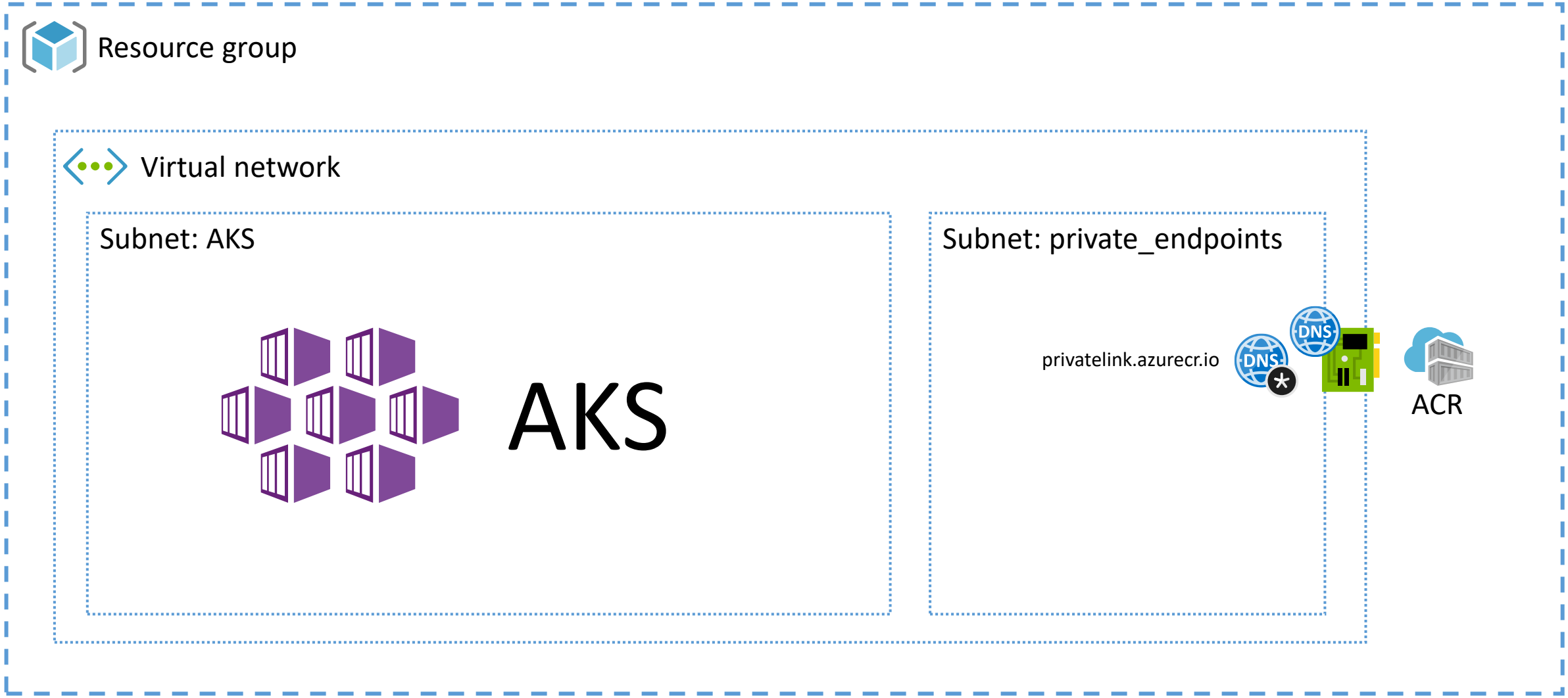
AKS


Resource group
Virtual network
Subnet: AKS
AKS
Subnet: private_endpoints
privatelink.azurecr.io
DNS
ACR

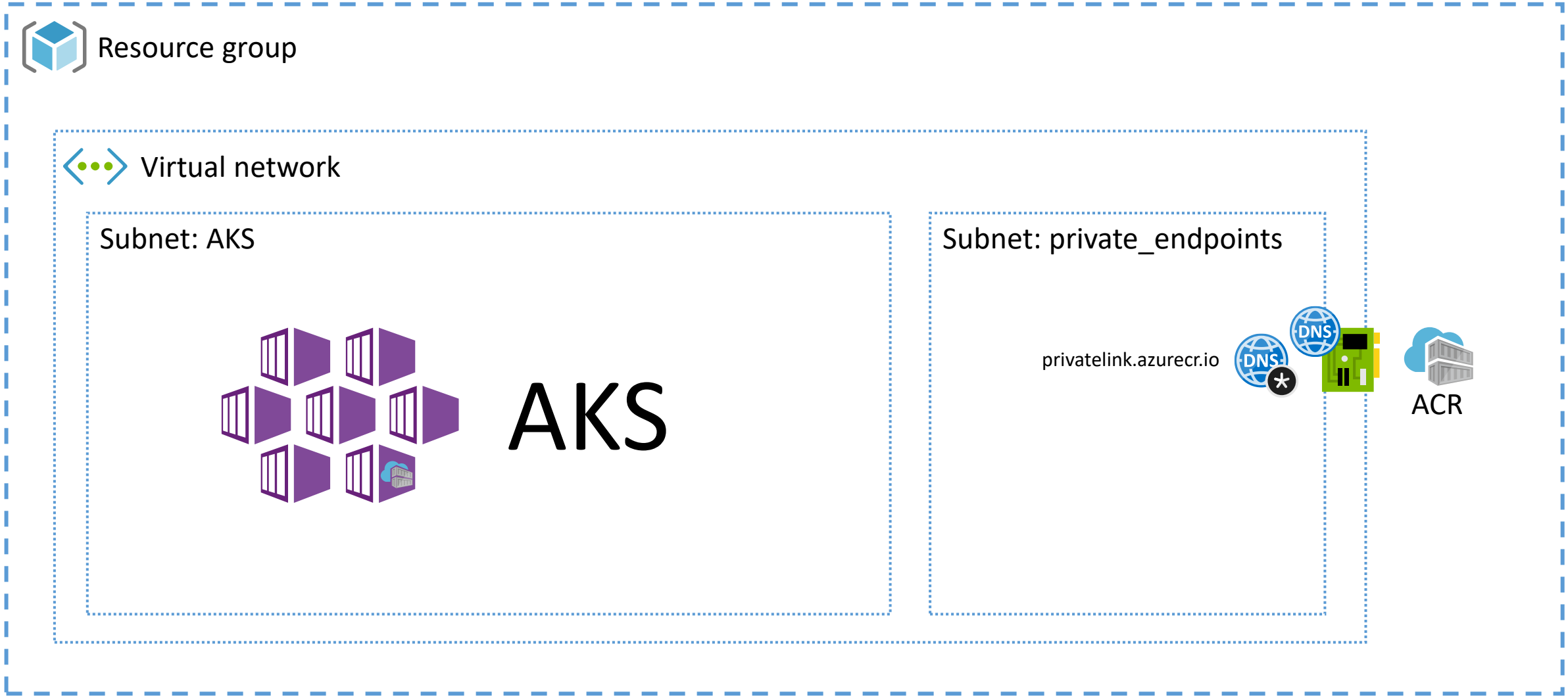
AKS


Resource group
Virtual network
Subnet: AKS
AKS
Subnet: private_endpoints
privatelink.azurecr.io
DNS
ACR

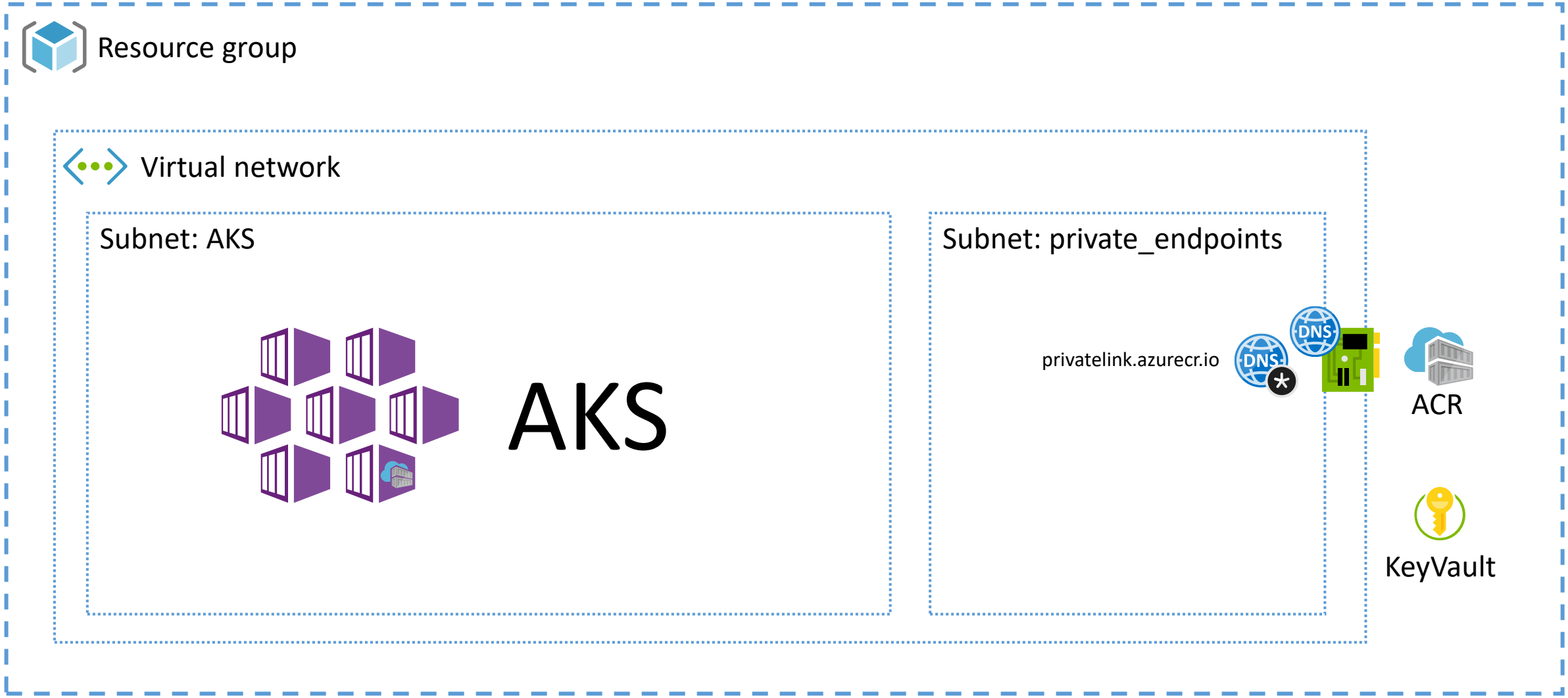
AKS


Resource group
Virtual network
Subnet: AKS
AKS
Subnet: private_endpoints
privatelink.azurecr.io
DNS
DNS
ACR

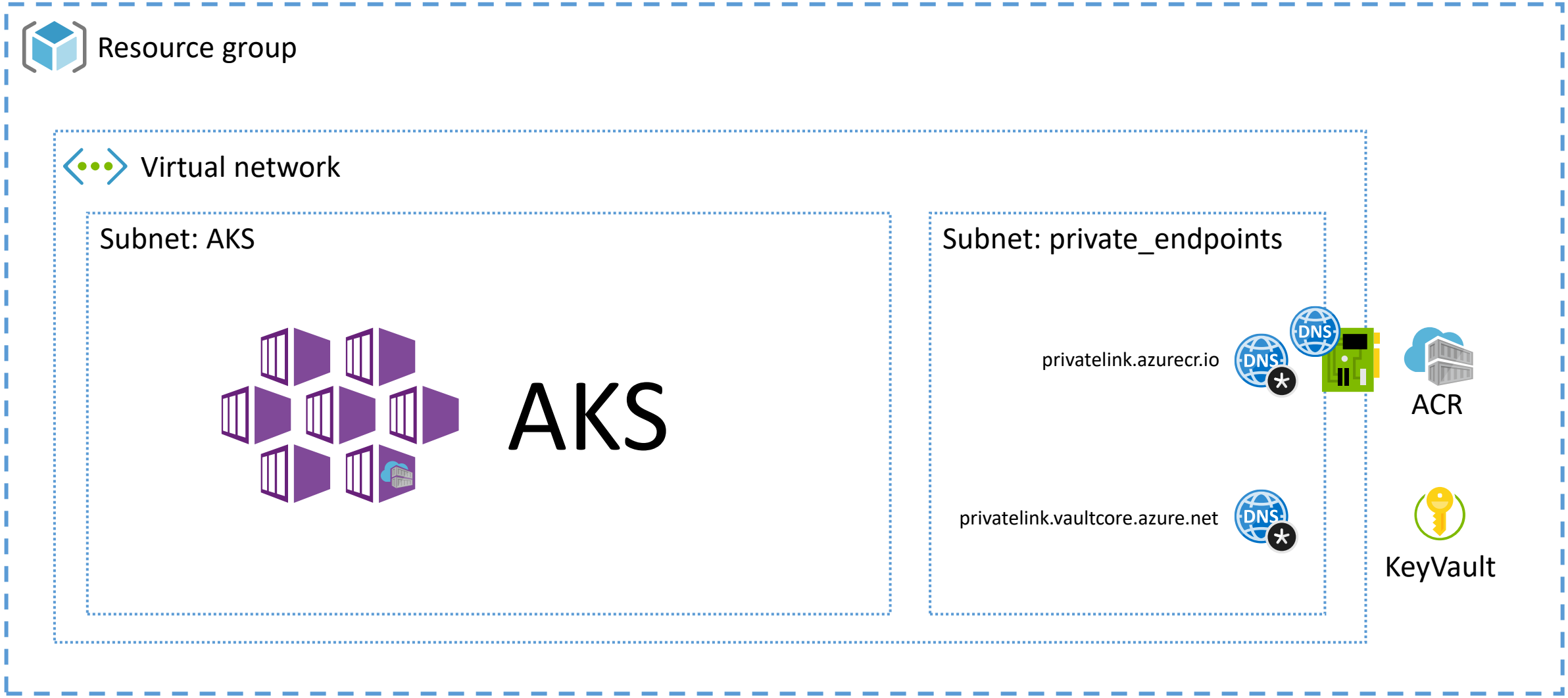
AKS


Resource group
Virtual network
Subnet: AKS
AKS
Subnet: private_endpoints
privatelink.azurecr.io
DNS
DNS
ACR

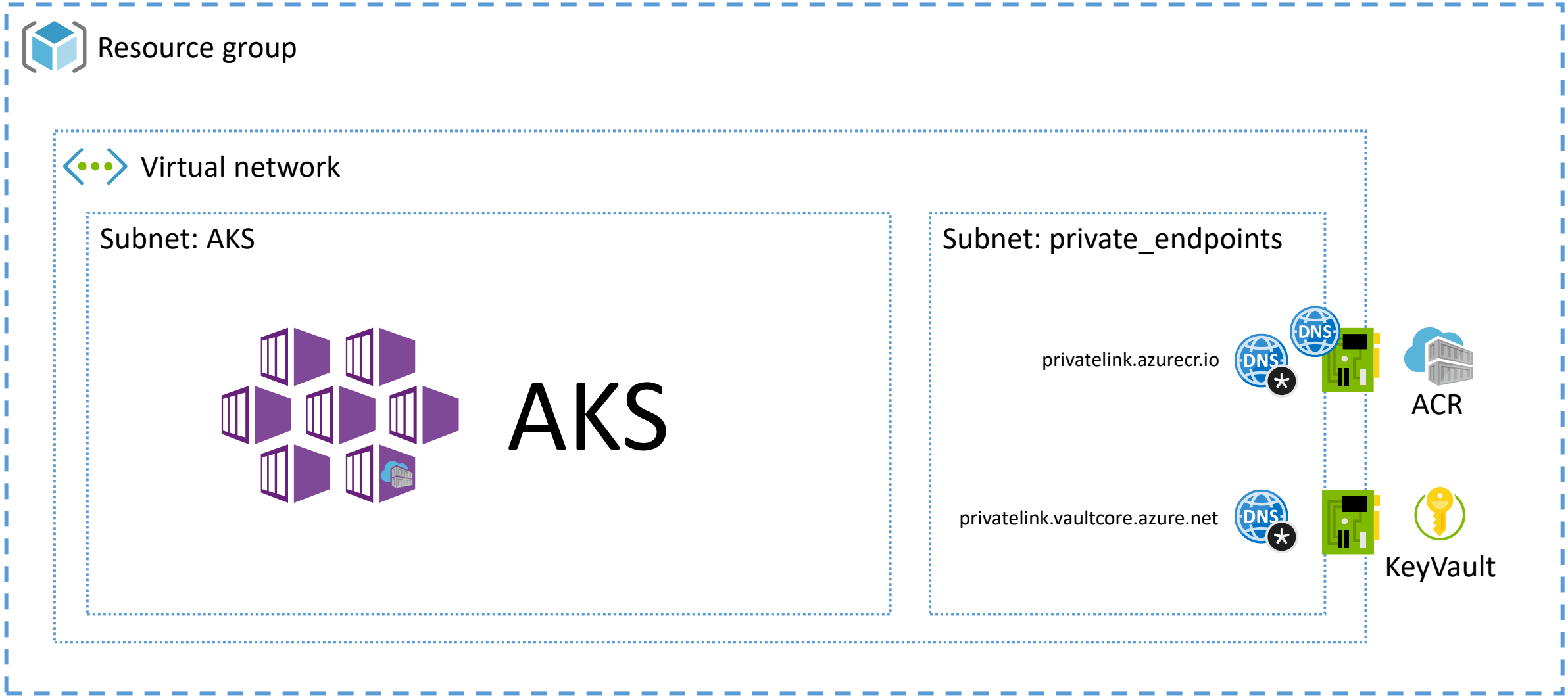
AKS


Resource group
Virtual network
Subnet: AKS
AKS
Subnet: private_endpoints
privatelink.azurecr.io
DNS
DNS
ACR
KeyVault

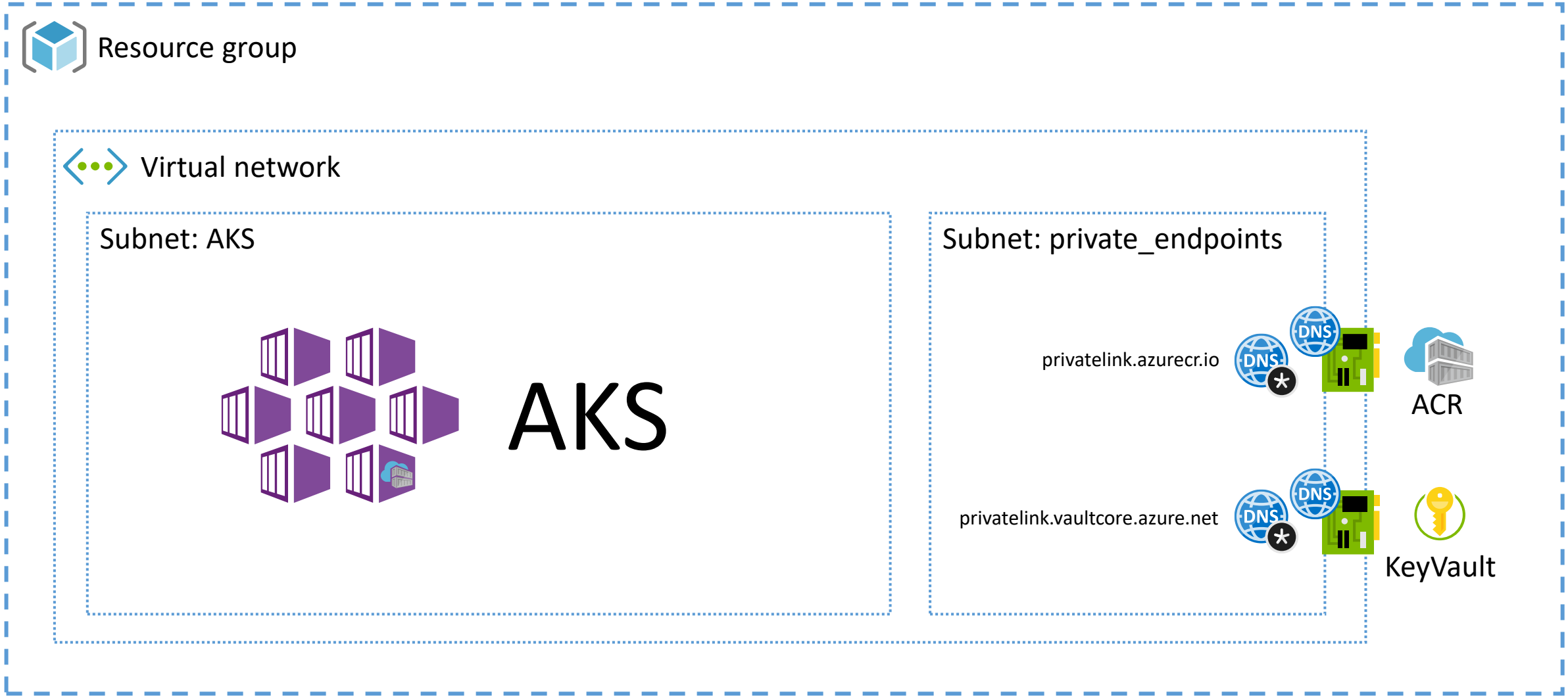
AKS


Resource group
Virtual network
Subnet: AKS
AKS
Subnet: private_endpoints
privatelink.azurecr.io
privatelink.vaultcore.azure.net
ACR
KeyVault

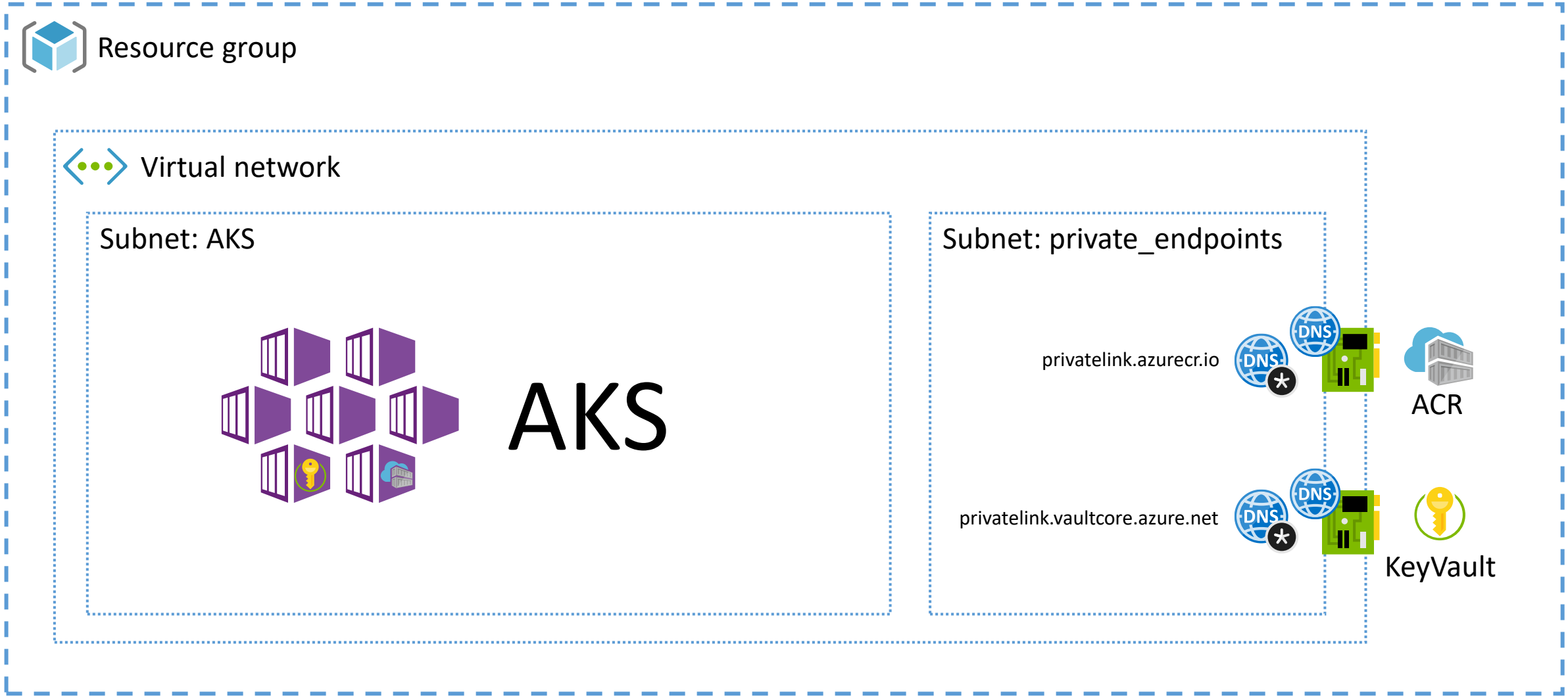
AKS


Resource group
Virtual network
Subnet: AKS
AKS
Subnet: private_endpoints
privatelink.azurecr.io
DNS
DNS
ACR
privatelink.vaultcore.azure.net
DNS
KeyVault

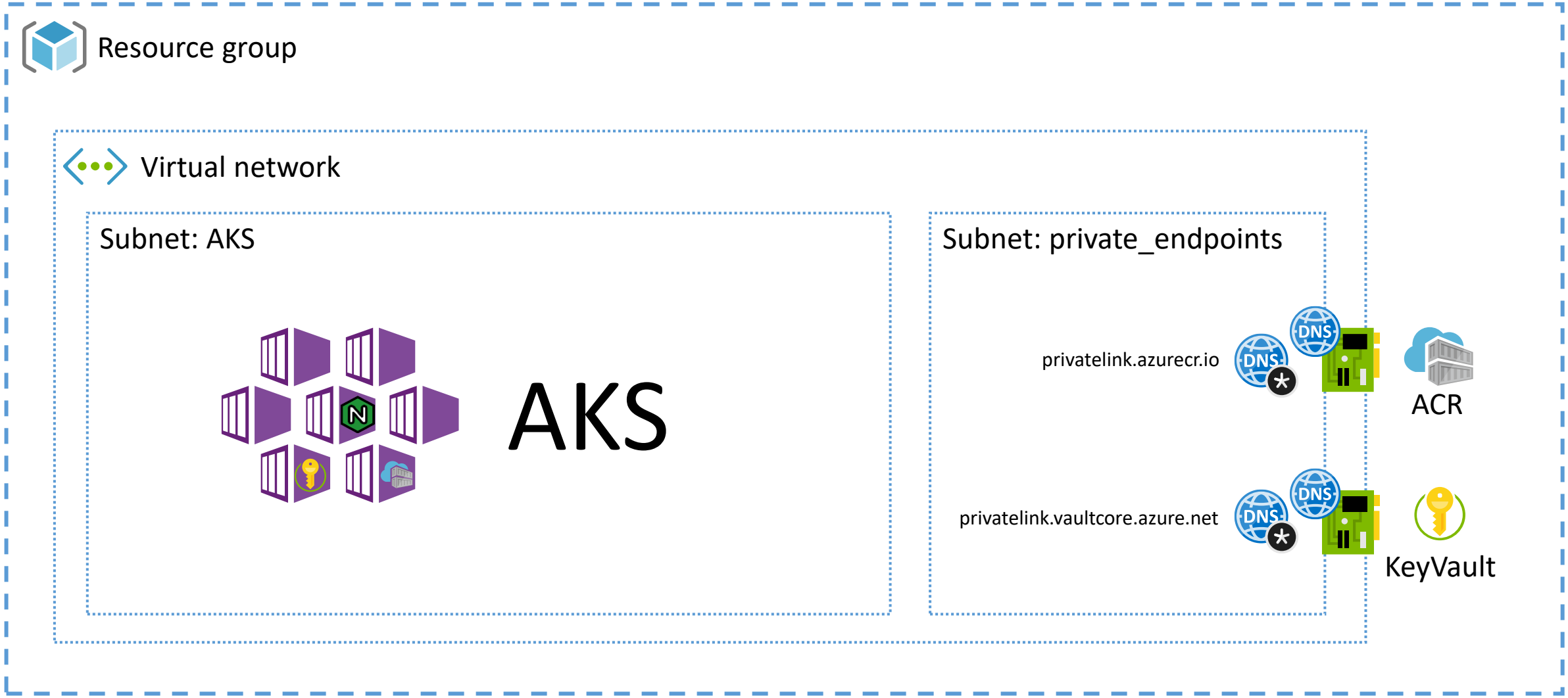
AKS


Resource group
Virtual network
Subnet: AKS
AKS
Subnet: private_endpoints
privatelink.azurecr.io
DNS
DNS
ACR
privatelink.vaultcore.azure.net
DNS
DNS
KeyVault

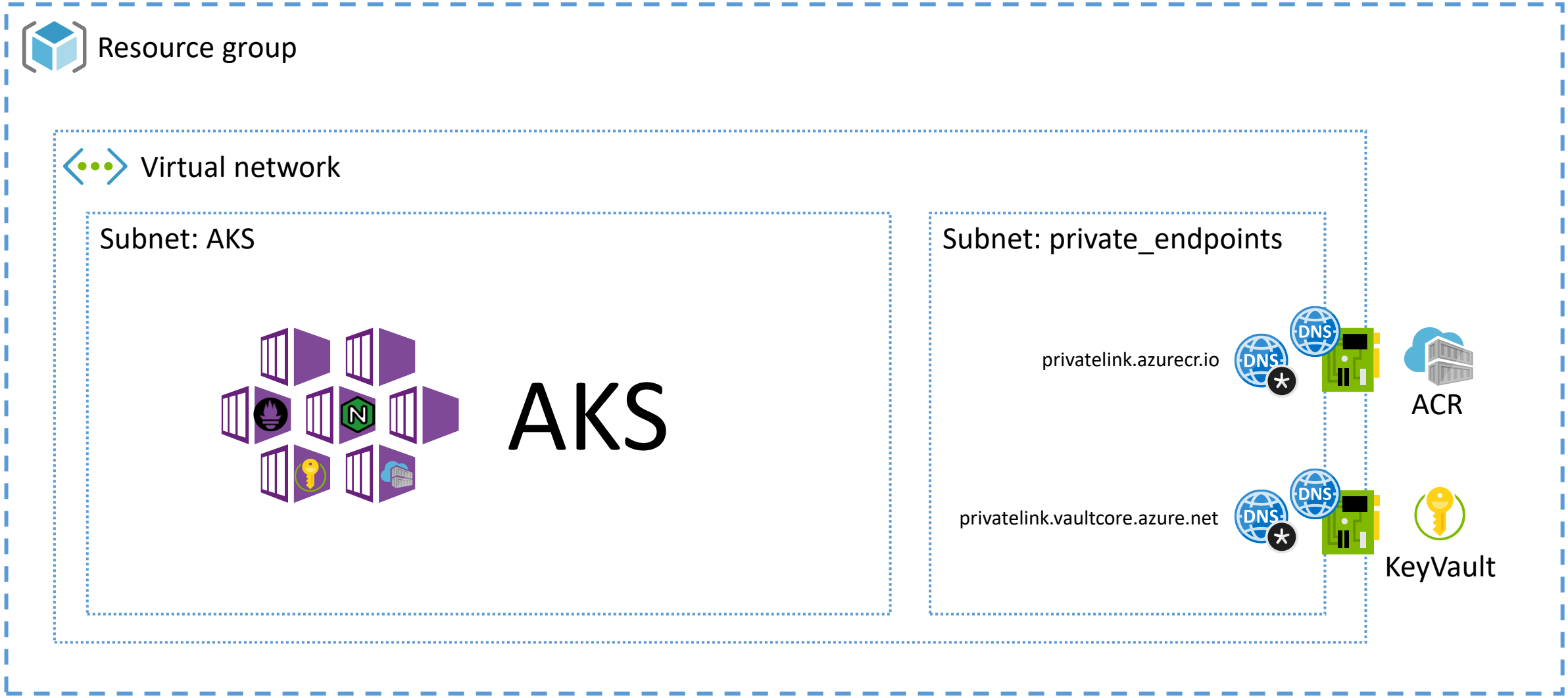
AKS


Resource group
Virtual network
Subnet: AKS
AKS
Subnet: private_endpoints
privatelink.azurecr.io
DNS
DNS
ACR
privatelink.vaultcore.azure.net
DNS
DNS
KeyVault

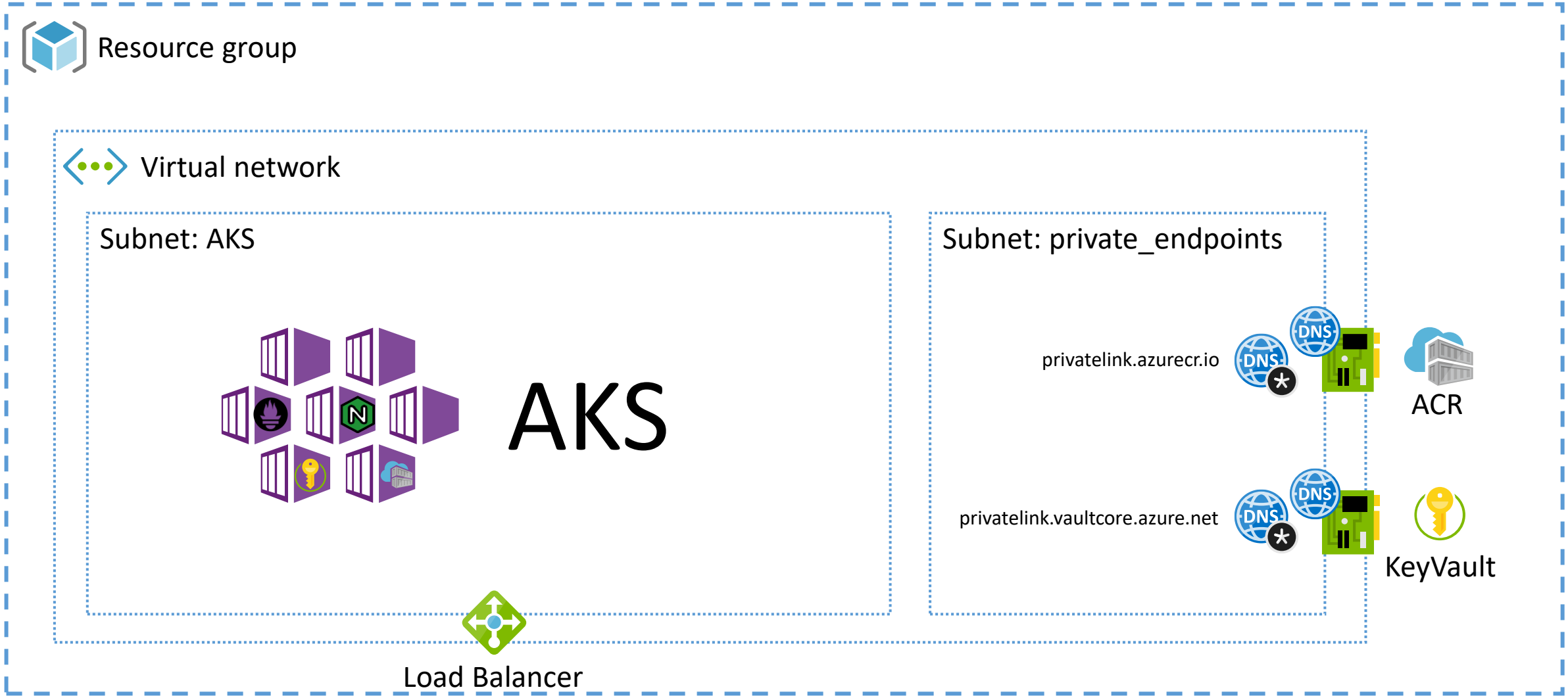
AKS


Resource group
Virtual network
Subnet: AKS
AKS
Subnet: private_endpoints
privatelink.azurecr.io
ACR
privatelink.vaultcore.azure.net
KeyVault

AKS


Resource group
Virtual network
Subnet: AKS
AKS
Subnet: private_endpoints
privatelink.azurecr.io
DNS
DNS
ACR
privatelink.vaultcore.azure.net
DNS
DNS
KeyVault

AKS


Resource group
Virtual network
Subnet: AKS
AKS
Subnet: private_endpoints
privatelink.azurecr.io
ACR
privatelink.vaultcore.azure.net
KeyVault
Load Balancer

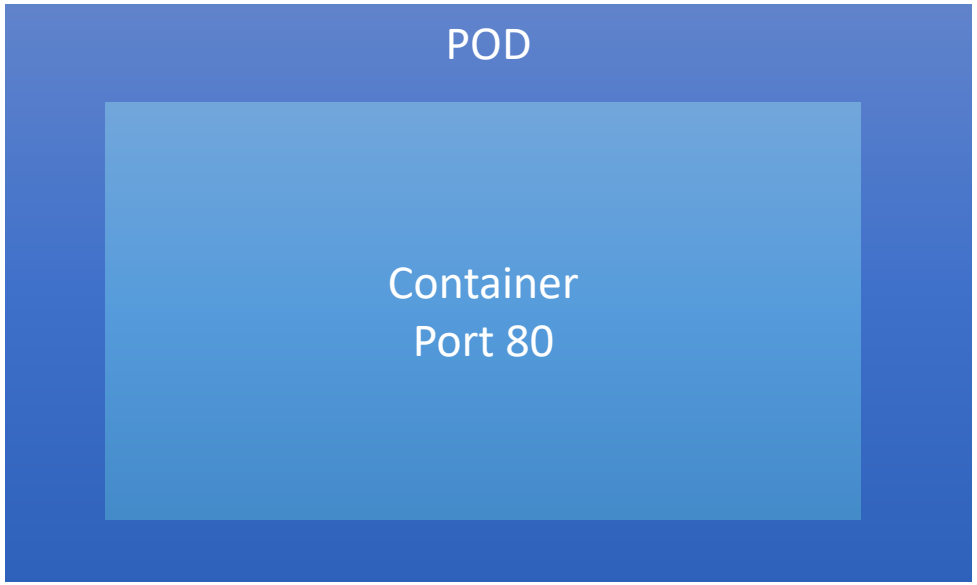
K8S Deployment

K8S Deployment

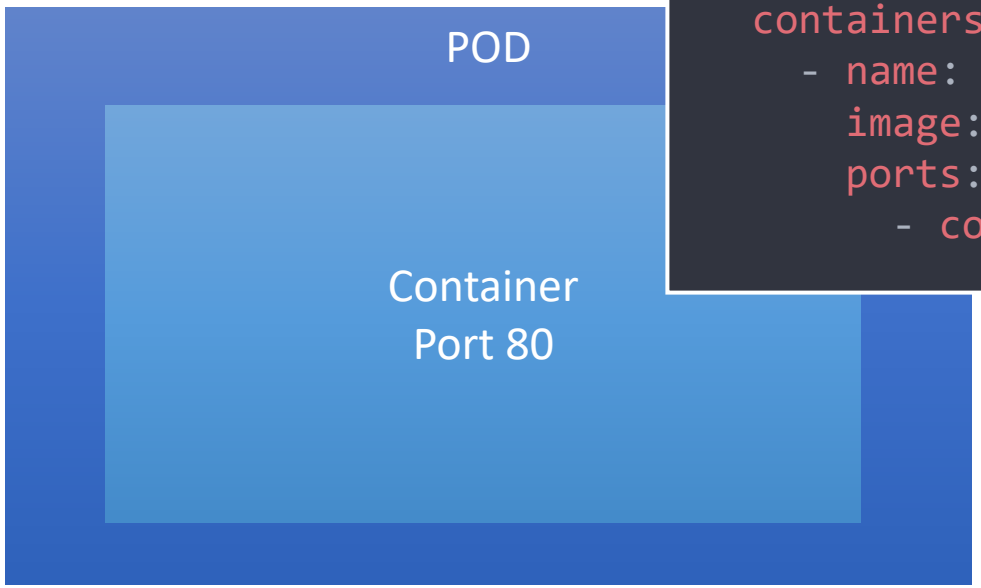
```
docker login $n.azurecr.io -u $user -p $password
docker tag myapi:0.1 fergab22.azurecr.io/myapi:0.1
docker push fergab22.azurecr.io/myapi:0.1
```

Container
Port 80

K8S Deployment

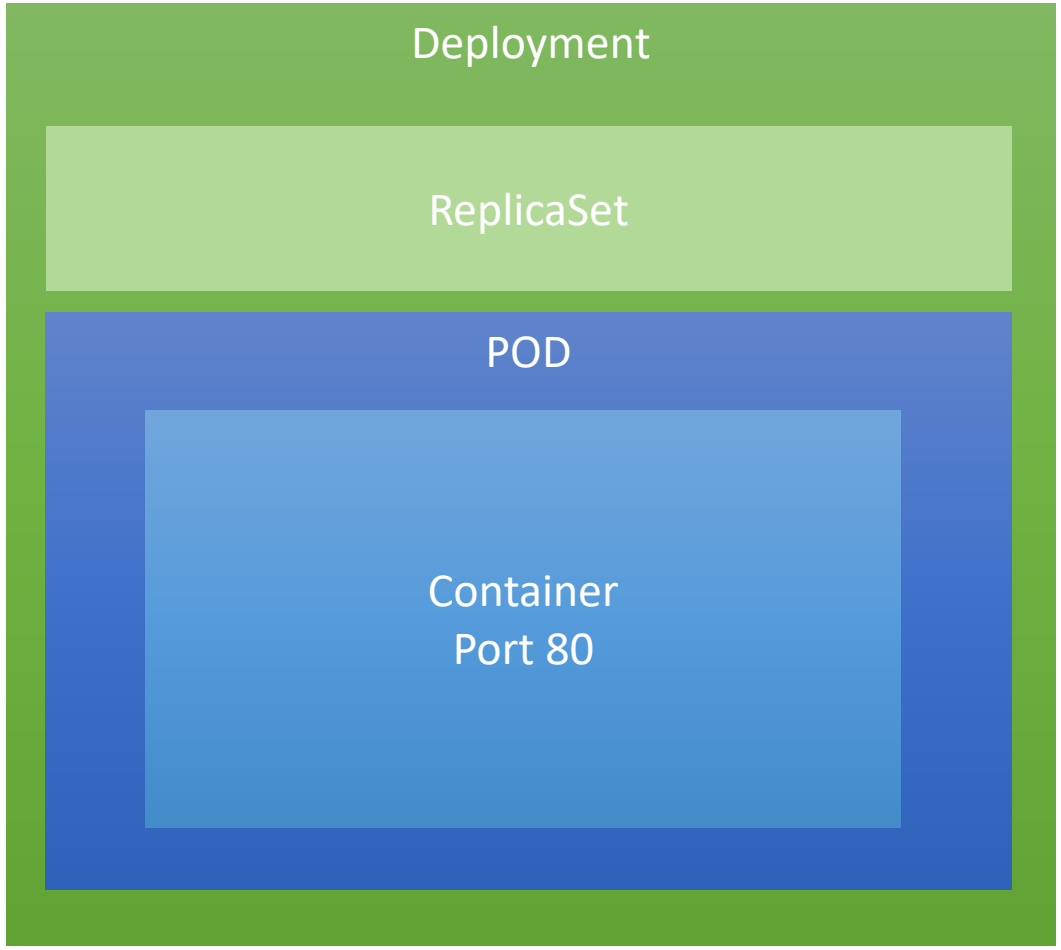


K8S Deployment

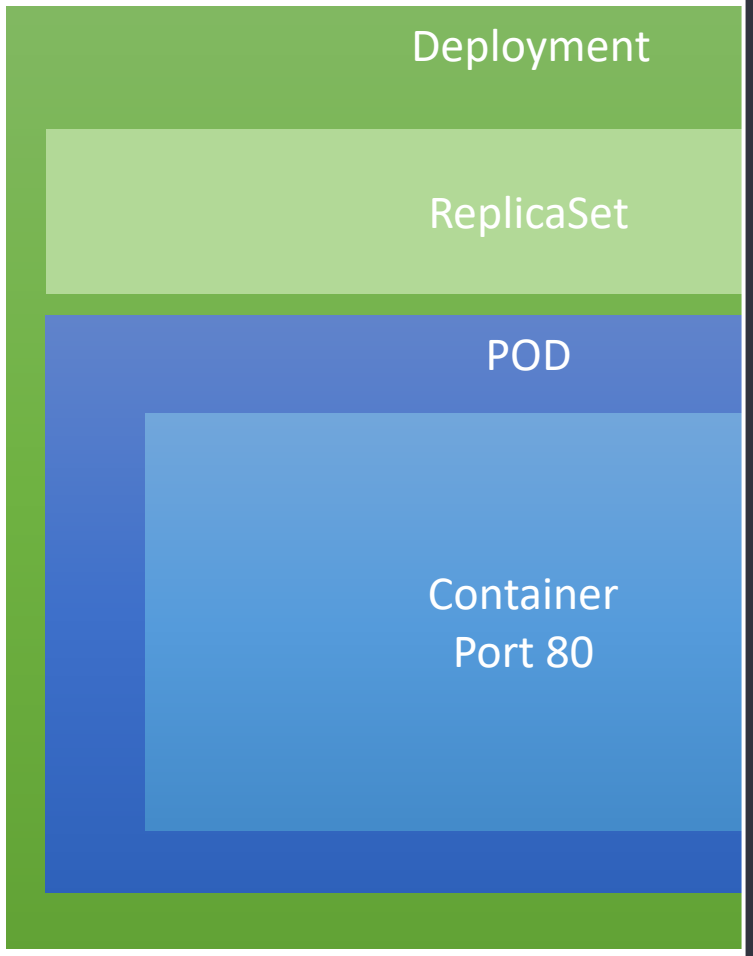



```yaml
kind: Pod
apiVersion: v1
metadata:
  name: my-api
spec:
  containers:
    - name: my-api
      image: fergab22.azurecr.io/myapi:0.1
      ports:
        - containerPort: 80
```

K8S Deployment

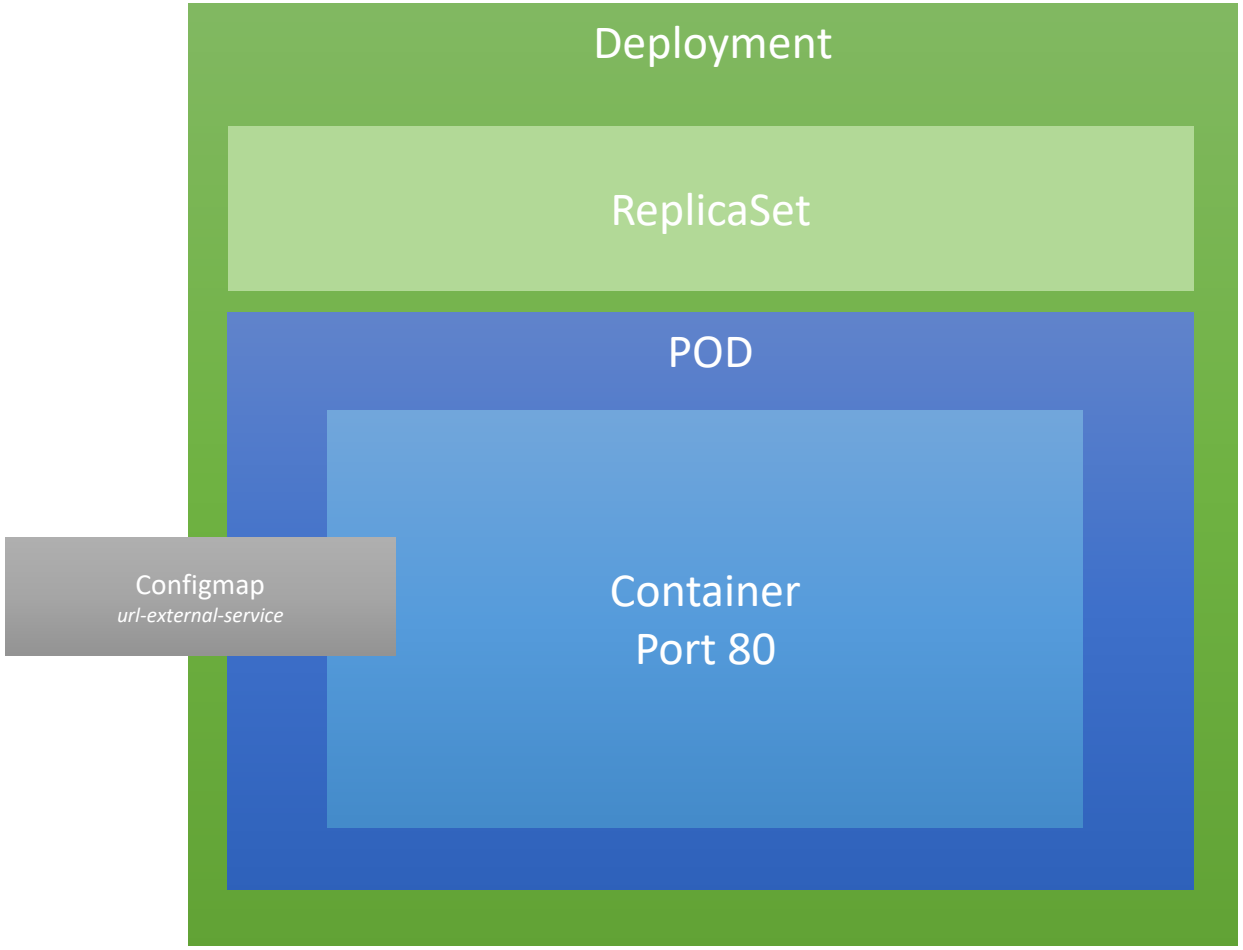


K8S Deployment

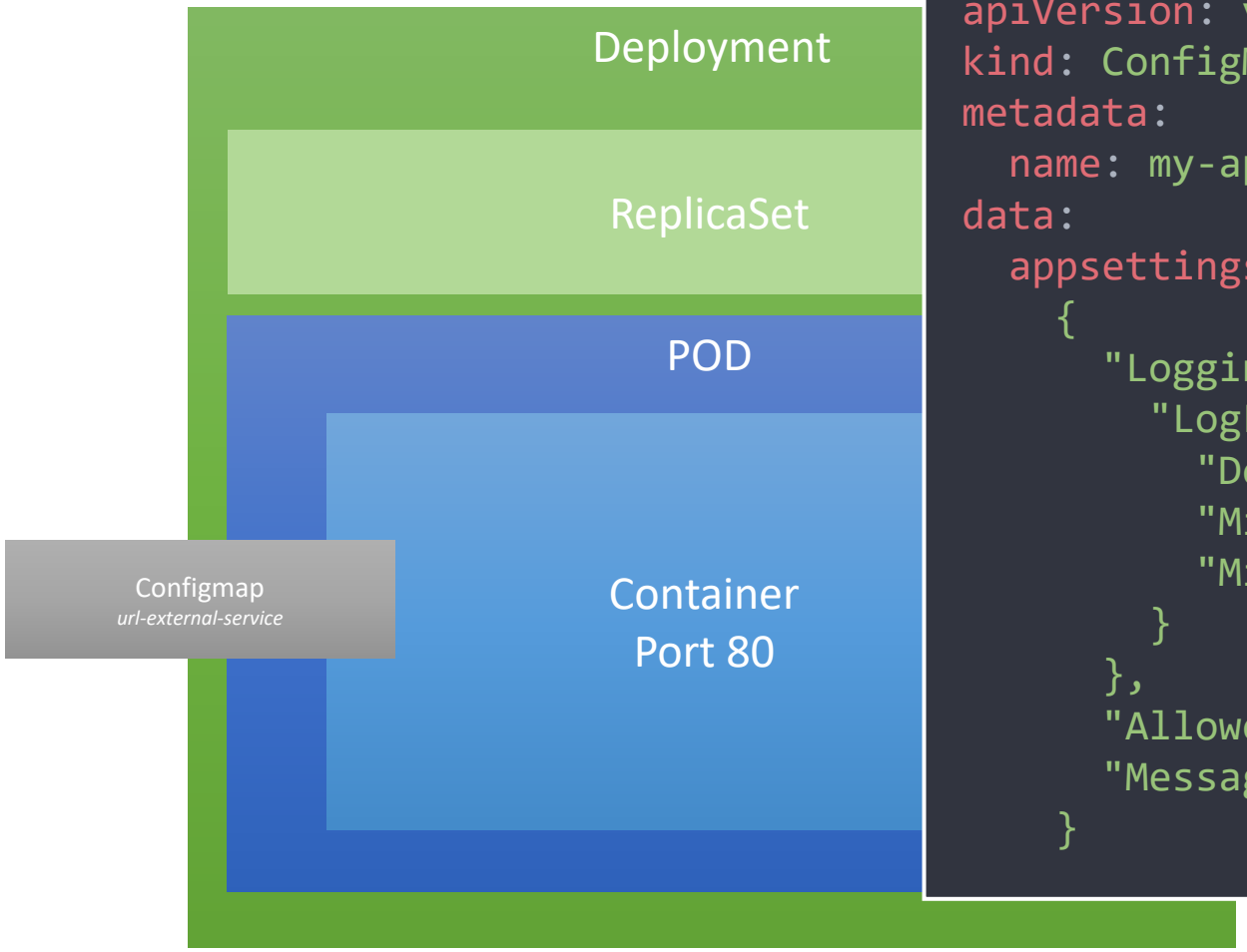



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-api
spec:
  selector:
    matchLabels:
      app: my-api
  replicas: 1
  template:
    metadata:
      labels:
        app: my-api
    spec:
      containers:
      - name: my-api
        image: fergab22.azurecr.io/myapi:0.1
        ports:
          - containerPort: 80
```

K8S Deployment

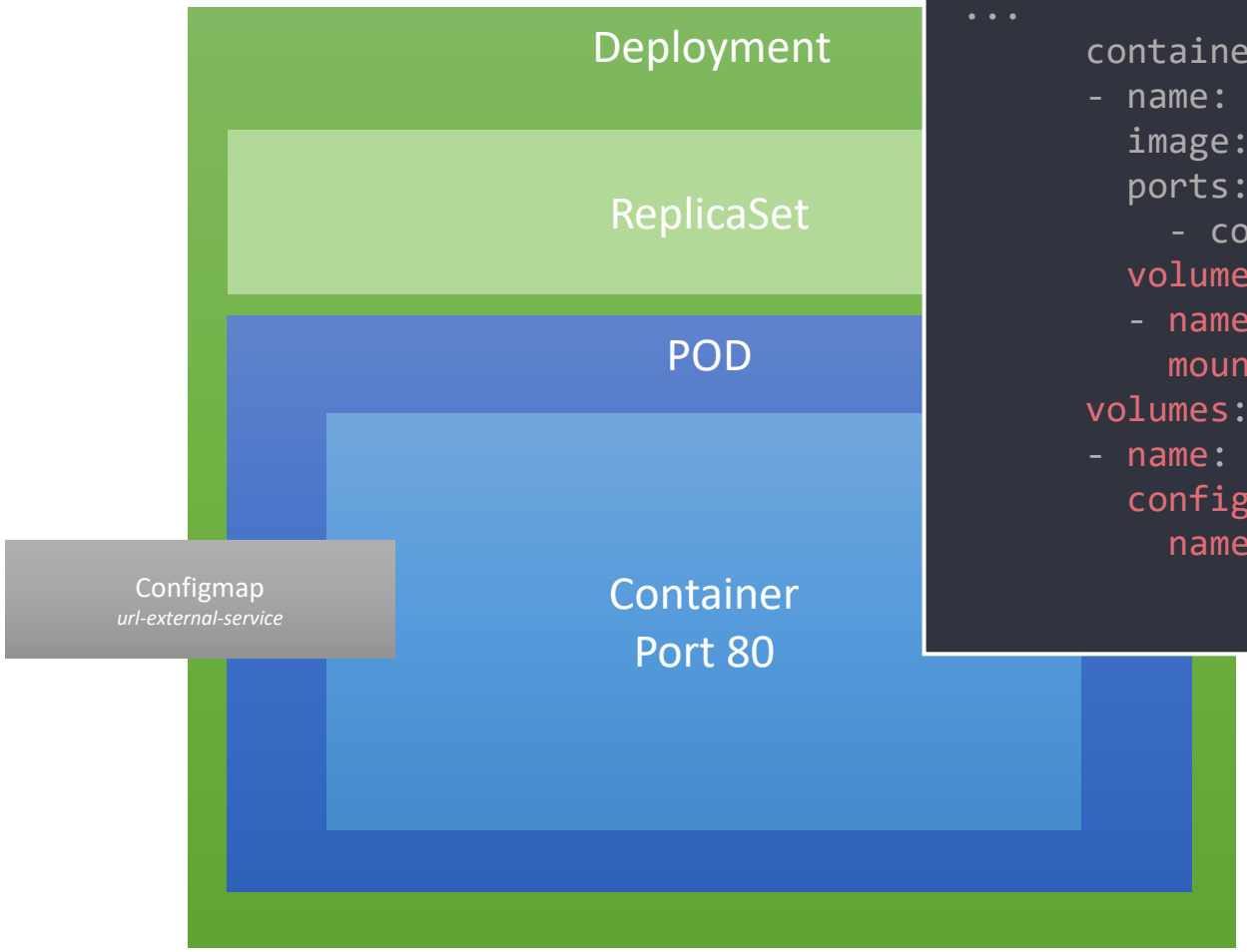


K8S Deployment



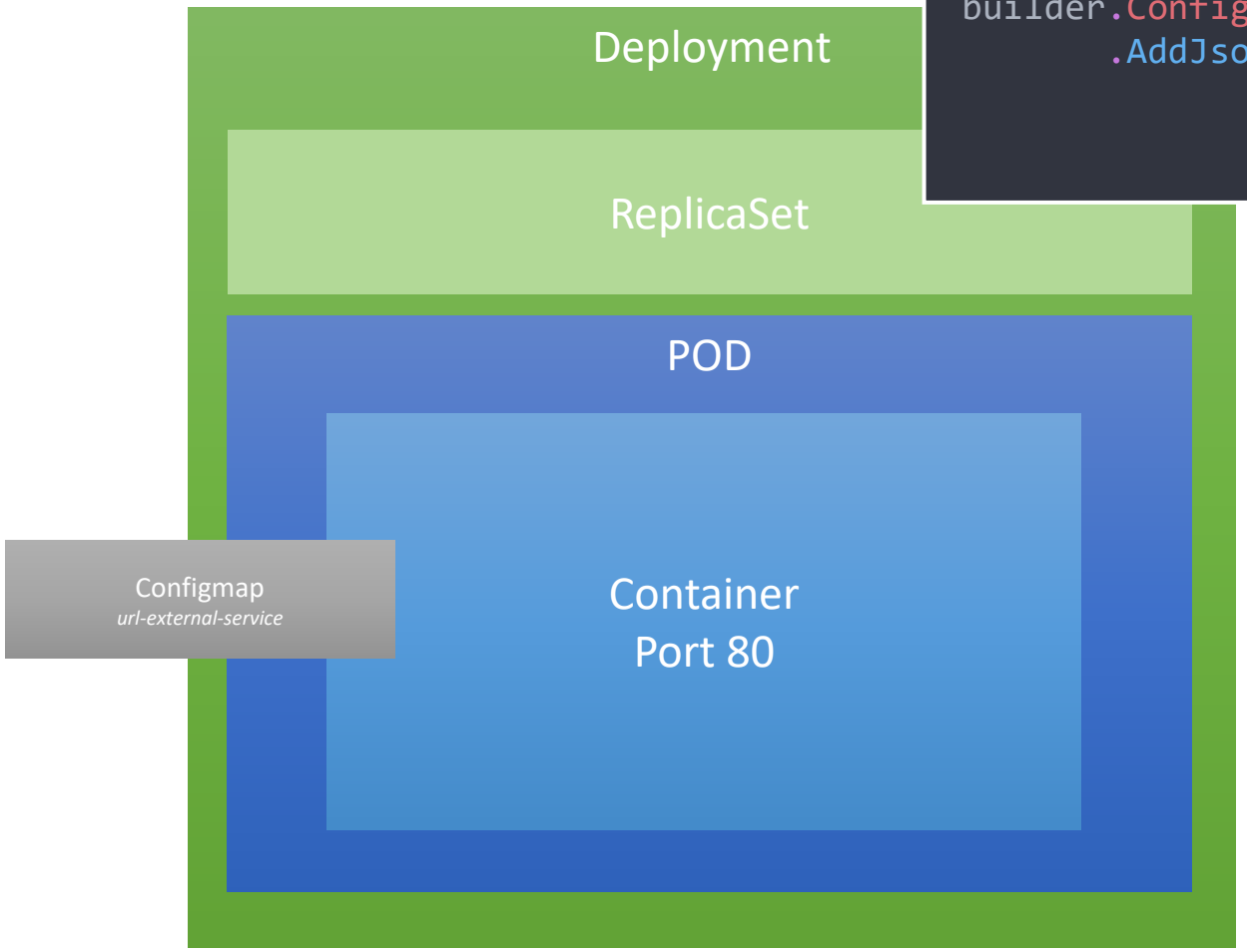

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: my-app-config
data:
  appsettings.json: |-
    {
      "Logging": {
        "LogLevel": {
          "Default": "Information",
          "Microsoft": "Warning",
          "Microsoft.Hosting.Lifetime": "Information"
        }
      },
      "AllowedHosts": "*",
      "Message": "Hello world!"
    }
```

K8S Deployment



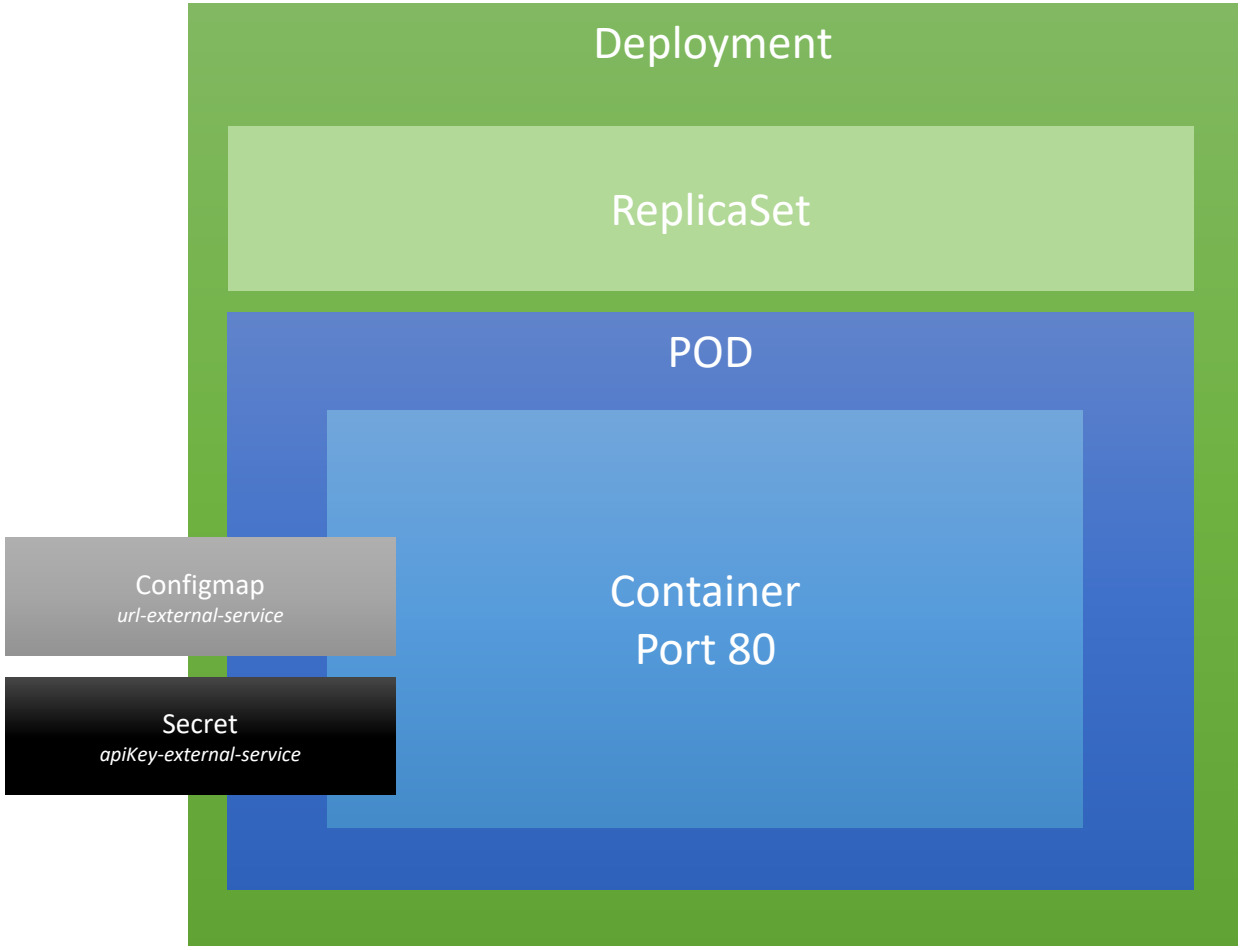

```yaml
...
      containers:
      - name: my-api
        image: fergab22.azurecr.io/my-api:1.0.0
        ports:
          - containerPort: 80
        volumeMounts:
        - name: appsettings-volume
          mountPath: /app/config
      volumes:
      - name: appsettings-volume
        configMap:
          name: my-app-config
```

K8S Deployment

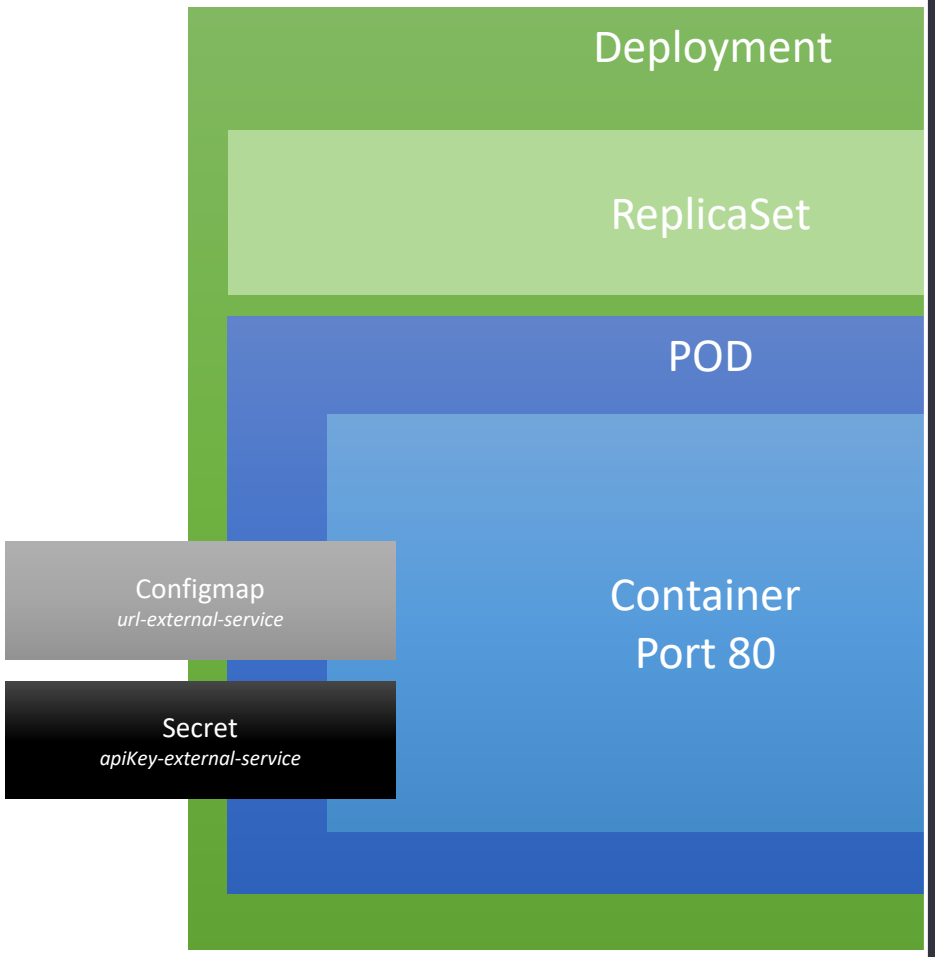



```
builder.Configuration
        .AddJsonFile("config/appsettings.json",
                     optional: true,
                     reloadOnChange: true);
```

K8S Deployment



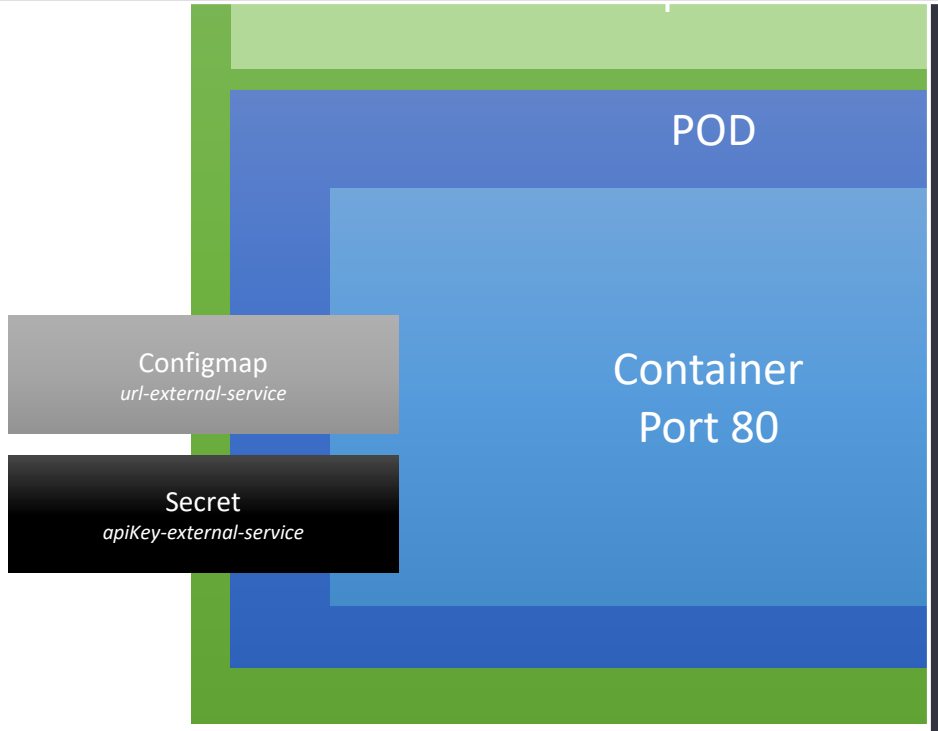
K8S Deployment




```yaml
apiVersion: secrets-store.csi.x-k8s.io/v1
kind: SecretProviderClass
metadata:
  name: azure-kv-secret
spec:
  provider: azure
  parameters:
    useVMManagedIdentity: "true"
    userAssignedIdentityID: f32*****-****-****-****-**********12
    keyvaultName: fergab22
    objects:  |
      array:
        - |
          objectName: TestSecret
          objectType: secret
    tenantId: dd7*****-****-****-****-**********fc
  secretObjects:
  - secretName: my-key-ring
    type: Opaque
    data:
    - key: testSecret
      objectName: TestSecret
```

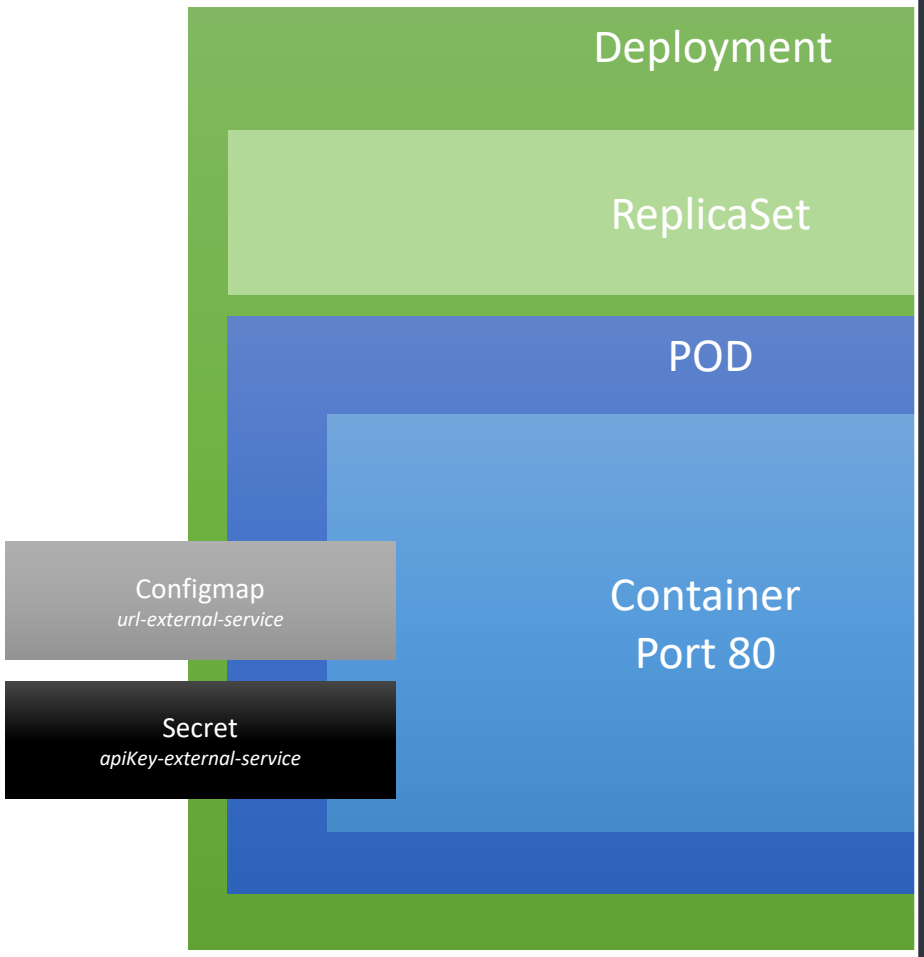
K8S Deployment

```
az aks show \
  -g $rg \
  -n $n \
  --query identityProfile.kubeletidentity.clientId \
  -o tsv
```



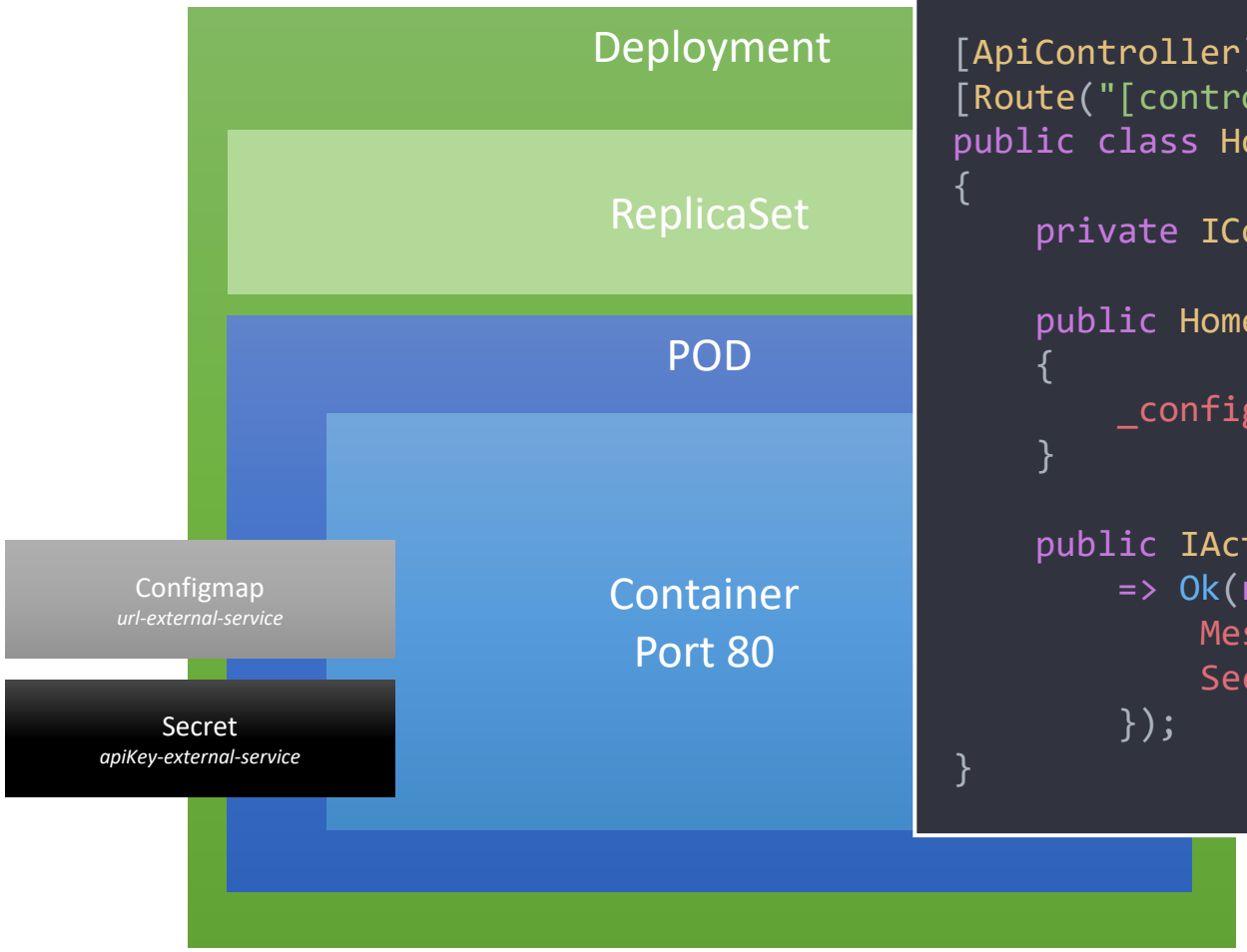

```
ersion: secrets-store.csi.x-k8s.io/v1
: SecretProviderClass
ata:
me: azure-kv-secret
:
provider: azure
parameters:
  useVMManagedIdentity: "true"
  userAssignedIdentityID: f32*****-****-****-****-**********12
  keyvaultName: fergab22
  objects: |
    array:
      - |
        objectName: TestSecret
        objectType: secret
  tenantId: dd7*****-****-****-****-**********fc
secretObjects:
- secretName: my-key-ring
  type: Opaque
  data:
  - key: testSecret
    objectName: TestSecret
```

K8S Deployment



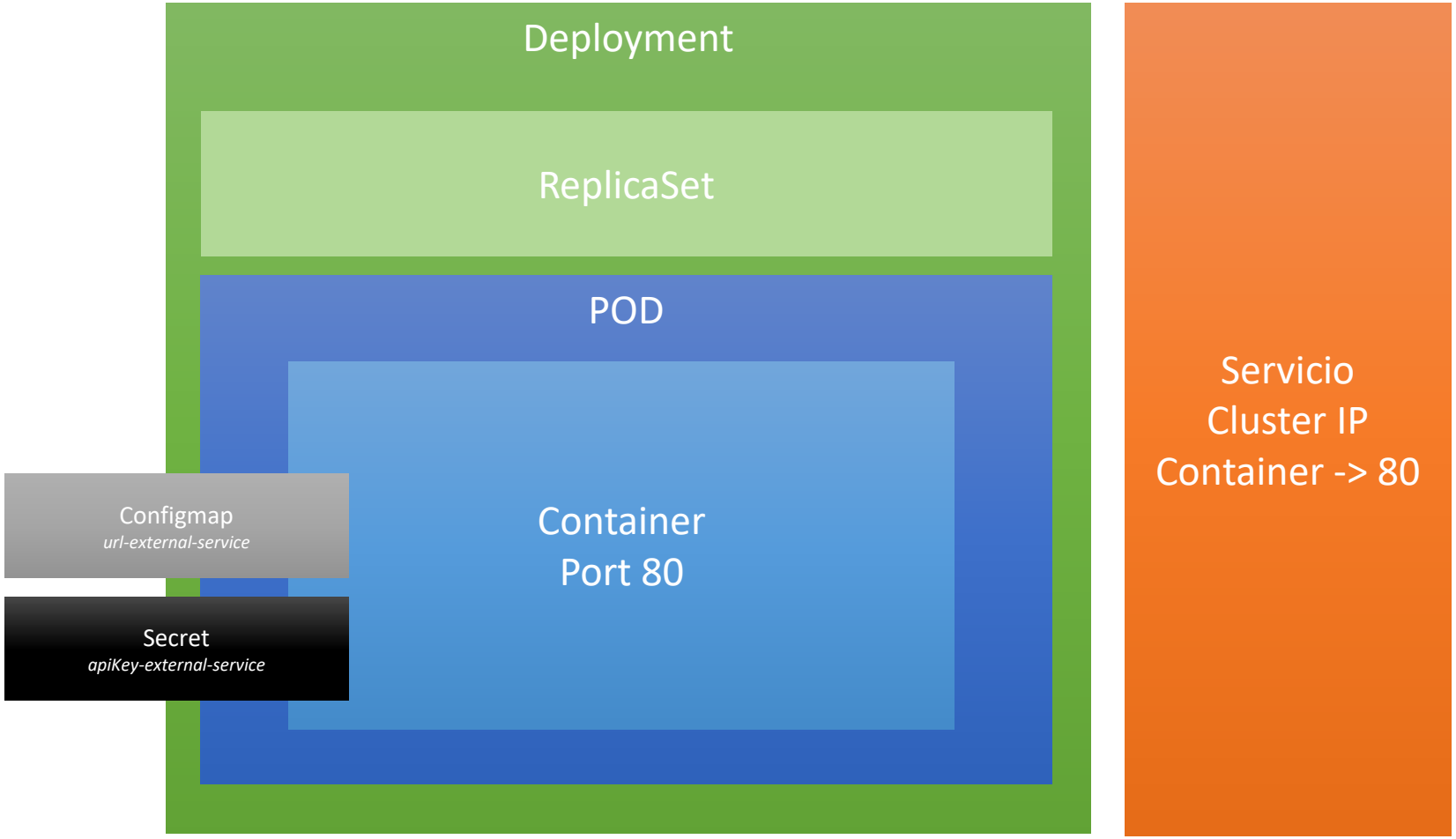

```yaml
...
      containers:
      - name: my-api
        image: fergab221.azurecr.io/my-api:1.0.0
        ports:
          - containerPort: 80
        env:
        - name: TEST_SECRET
          valueFrom:
            secretKeyRef:
              name: my-key-ring
              key: testSecret
        volumeMounts:
        - name: secrets-store01
          mountPath: "/mnt/secrets-store"
          readOnly: true
      volumes:
      - name: secrets-store01
        csi:
          driver: secrets-store.csi.k8s.io
          readOnly: true
          volumeAttributes:
            secretProviderClass: "azure-kv-secret"
```

K8S Deployment

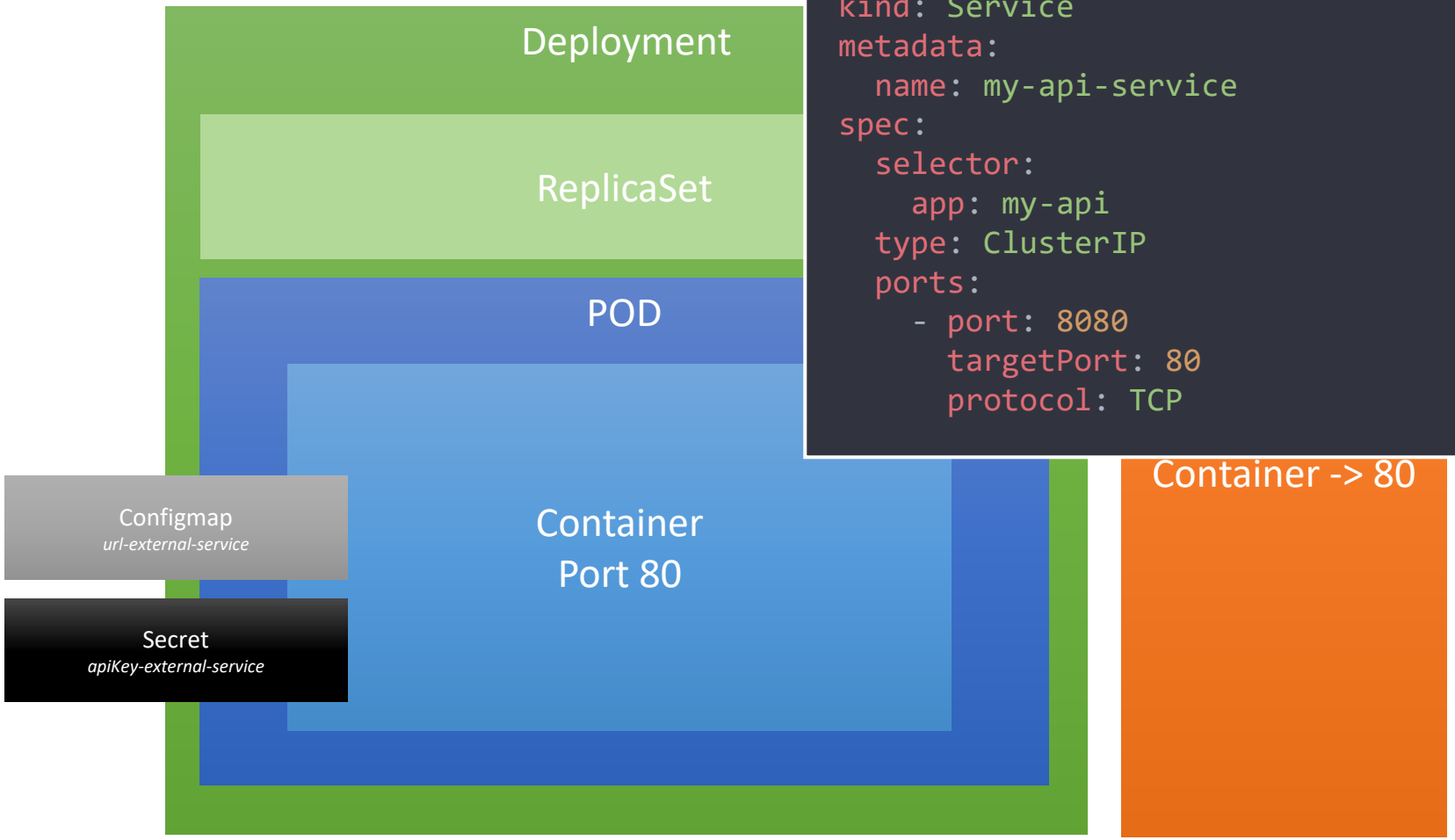



```csharp
using Microsoft.AspNetCore.Mvc;

[ApiController]
[Route("[controller]")]
public class HomeController : ControllerBase
{
    private IConfiguration _configuration;

    public HomeController(IConfiguration configuration)
    {
        _configuration = configuration;
    }

    public IActionResult Get()
        => Ok(new {
            Message = _configuration["Message"],
            Secret = _configuration["TEST_SECRET"]
        });
}
```

K8S Deployment



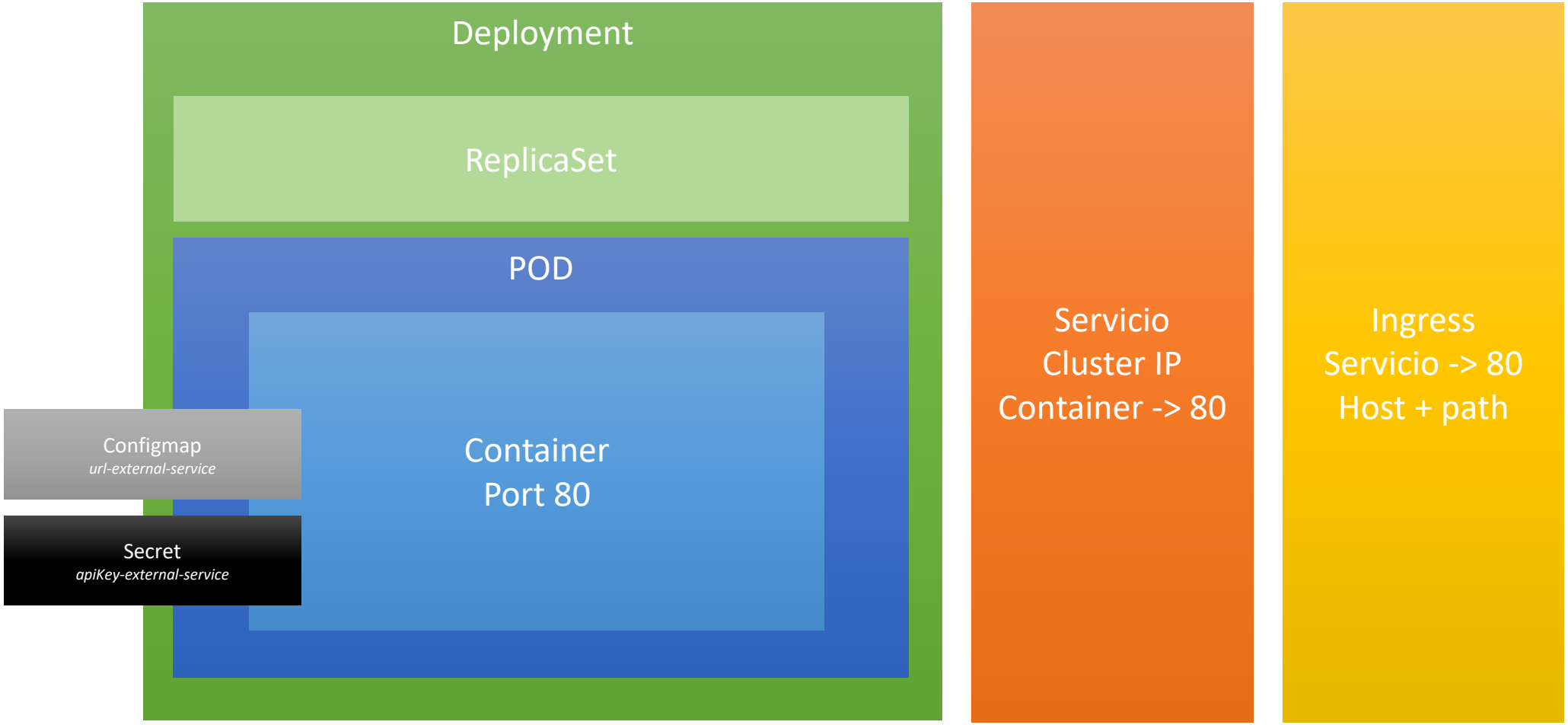
K8S Deployment



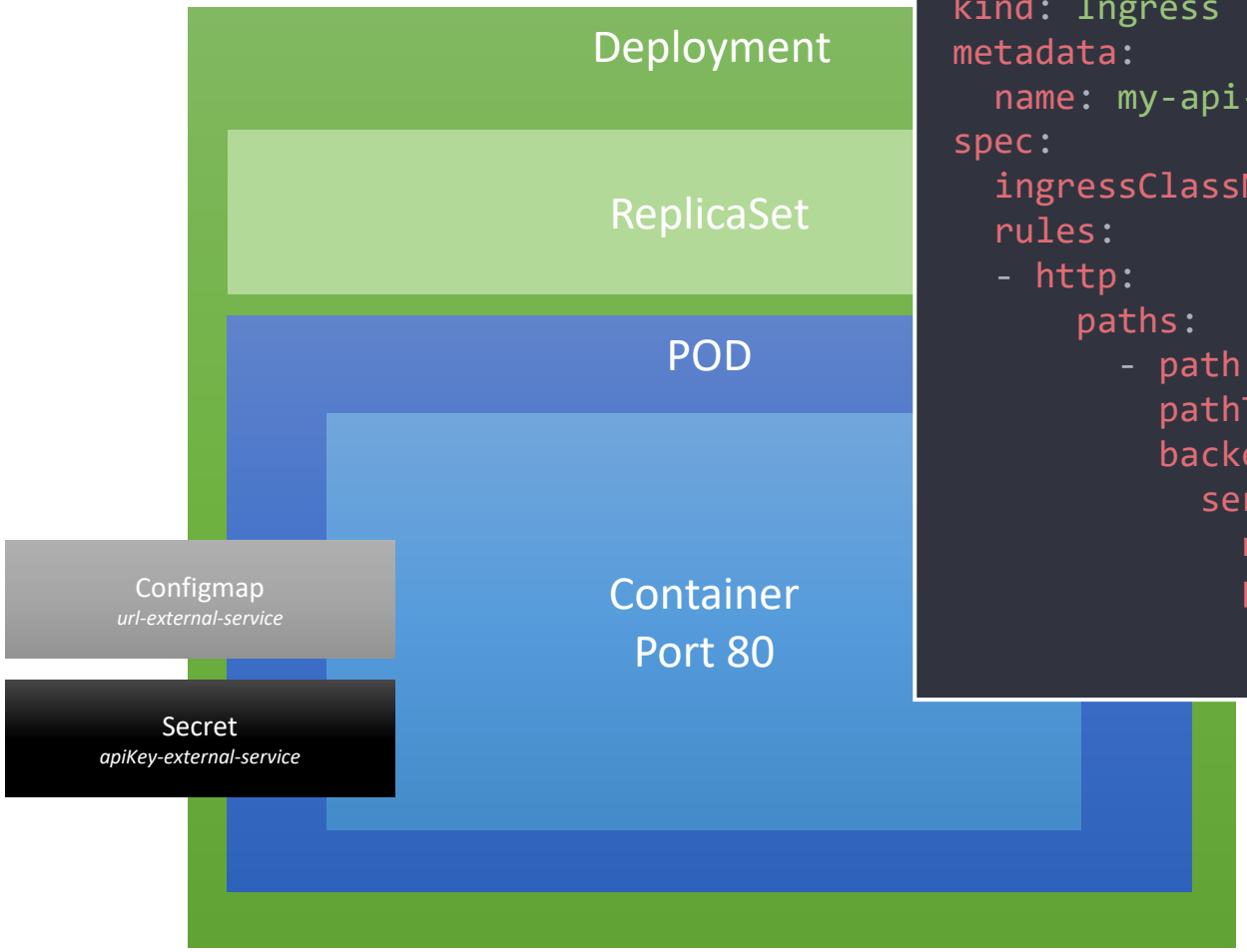

```yaml
apiVersion: v1
kind: Service
metadata:
  name: my-api-service
spec:
  selector:
    app: my-api
  type: ClusterIP
  ports:
    - port: 8080
      targetPort: 80
      protocol: TCP
```

Container -> 80

K8S Deployment



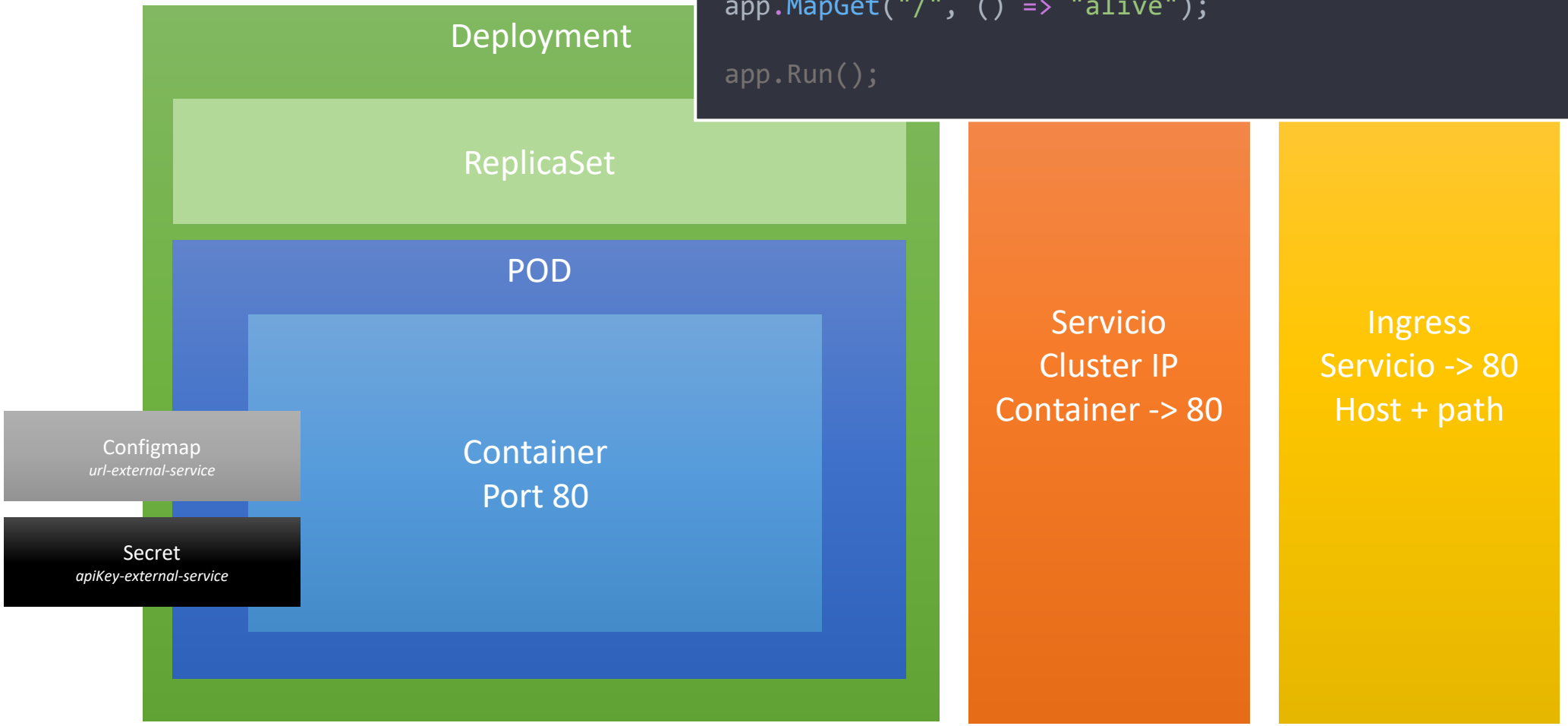
K8S Deployment




```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-api-ingress
spec:
  ingressClassName: nginx
  rules:
  - http:
      paths:
        - path: /
          pathType: Prefix
          backend:
            service:
              name: my-api-service
              port:
                number: 8080
```

K8S Deployment

```
app.MapControllers();
app.MapGet("/", () => "alive");

app.Run();
```



```
az group delete -n $n
```

```
thank you!
```

Nuestro patrocinadores

Colabora

BY COMMUNITY • FOR COMMUNITY

SPAIN

#GLOBALAZURE