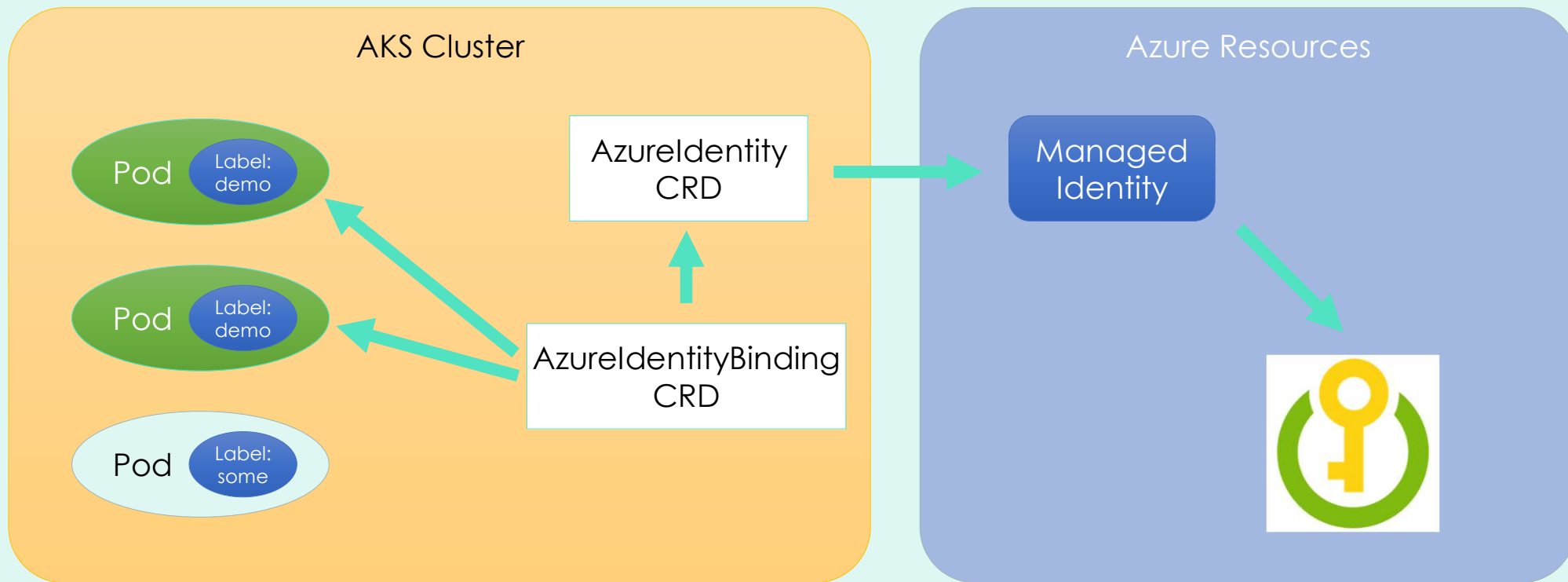


nova
net

nova
net

Key Vault FlexVolume

Olav Nybø

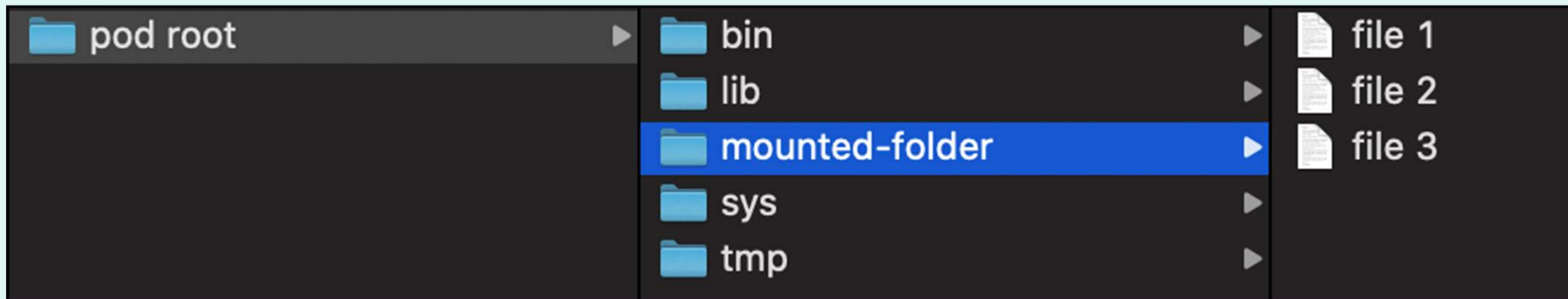


nova
net


```
private static async Task ShowKeyVaultSecret()
{
    var provider = new AzureServiceTokenProvider();
    var kv = new KeyVaultClient(new KeyVaultClient.AuthenticationCallback(provider.KeyVaultTokenCallback));
    try
    {
        var secret =
            await kv.GetSecretAsync($"https://midgard-key.vault.azure.net/", "test-secret");
        Console.WriteLine($"secret is: '{secret.Value}'");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Exception when calling GetSecretAsync: '{ex}'");
    }
}
```

Kubernetes - Volumes

- Fil abstraksjon
- Knytter en Pod mot persistent lager
- En driver sørger for å tilby data som om det er filer



Azure Key Vault - Key Value Store

 **midgard-key | Secrets**
Key vault

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Events (preview)

Settings

Keys


Secrets

Certificates

+ Generate/Import

↻ Refresh

↑ Restore Backup

 The secret 'another-secret' has been successfully created.

Name	Type
another-secret	test secret - can delete
discord-token	Discord token
midgard-deploy-sp-appid	Service Principal AppId
midgard-deploy-sp-secret	Service principal secret
sanity-token	Sanity api write token
sqlpassword	Password for the sqladmin user
test-secret	test secret

Key Vault flex volume installation

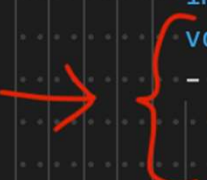
- <https://github.com/Azure/kubernetes-keyvault-flexvol/blob/master/deployment/kv-flexvol-installer.yaml>

Deployment - AAD Pod Identity

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: keyvault-reader
  labels:
    component: keyvault-reader
spec:
  replicas: 1
  selector:
    matchLabels:
      component: keyvault-reader
      aadpodidbinding: my-demo-pod
  template:
    metadata:
      labels:
        component: keyvault-reader
        aadpodidbinding: my-demo-pod
    spec:
      containers:
        - name: keyvault-reader
          image: midgard.azurecr.io/keyvault-reader:latest
```

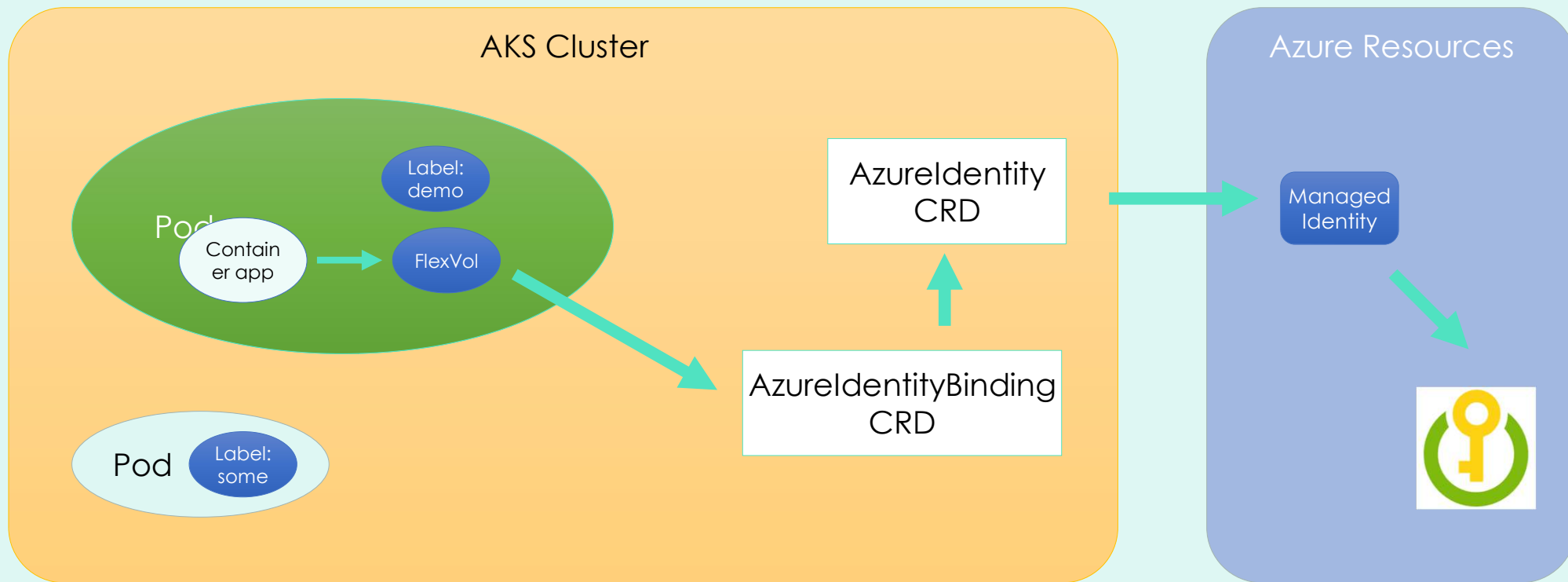

Deployment - Volume Mount

```
spec:
  replicas: 1
  selector:
    matchLabels:
      component: keyvault-reader
      aadpodidbinding: my-demo-pod
  template:
    metadata:
      labels:
        component: keyvault-reader
        aadpodidbinding: my-demo-pod
    spec:
      containers:
        - name: keyvault-reader
          image: midgard.azurecr.io/keyvault-reader:latest
          volumeMounts:
            - name: keyvault
              mountPath: /kvmnt
              readOnly: true
      volumes:
```



Deployment - Key Vault Flex Volume

```
spec:
  containers:
    - name: keyvault-reader
      image: midgard.azurecr.io/keyvault-reader:latest
      volumeMounts:
        - name: keyvault
          mountPath: /kvmnt
          readOnly: true
  volumes:
    - name: keyvault
      flexVolume:
        driver: "azure/kv"
        options:
          usepodidentity: "true"
          keyvaultname: "midgard-key"
          keyvaultobjectnames: "test-secret;another-secret" # list of KeyVault object names (semi-colon separated)
          keyvaultobjecttypes: secret;secret # list of KeyVault object types: secret, key or cert (semi-colon separated)
          tenantid: "187d6d88-2b74-4702-bb6b-0a894642f922" # the tenant ID of the KeyVault
```



nova
net

Demo - list files

Read Key Vault flex volume file from code

- .Net configuration packages
- Microsoft.Extensions.Configuration.*

Deployment - Key Vault Flex Volume

```
keyvault-reader.csproj src/keyvault-reader/keyvault-reader.csproj
1 <Project Sdk="Microsoft.NET.Sdk">
2
3   <PropertyGroup>
4     <OutputType>Exe</OutputType>
5     <TargetFramework>netcoreapp3.1</TargetFramework>
6     <RootNamespace>KeyvaultReader</RootNamespace>
7     <UserSecretsId>Midgard-Keyvault-Reader-Demo</UserSecretsId>
8   </PropertyGroup>
9
10  <ItemGroup>
11    <PackageReference Include="Microsoft.Azure.KeyVault" Version="3.0.5" />
12    <PackageReference Include="Microsoft.Azure.Services.AppAuthentication" Version="1.4.0" />
13    <PackageReference Include="Microsoft.Extensions.Configuration" Version="3.1.3" />
14    <PackageReference Include="Microsoft.Extensions.Configuration.EnvironmentVariables" Version="3.1.3" />
15    <PackageReference Include="Microsoft.Extensions.Configuration.Json" Version="3.1.3" />
16    <PackageReference Include="Microsoft.Extensions.Configuration.KeyPerFile" Version="3.1.3" />
17    <PackageReference Include="Microsoft.Extensions.Configuration.UserSecrets" Version="3.1.3" />
18  </ItemGroup>
19
20 </Project>
21
```

Deployment - Key Vault Flex Volume

```
static async Task Main(string[] args)
{
    var cfg = new ConfigurationBuilder()
        .SetBasePath(Directory.GetCurrentDirectory())
        .AddJsonFile("appsettings.json", optional: true, reloadOnChange: true)
        .AddEnvironmentVariables()
        .AddKeyPerFile("/kvmnt", optional: true)
        .AddUserSecrets<Program>(optional: true)
        .Build();

    var c = 1;
    while (true)
    {
        await Task.Delay(2000);
        Console.WriteLine($"trying: {c++}");
        ShowSecret(cfg);
    }
}

1 reference
private static void ShowSecret(IConfiguration cfg)
{
    Console.WriteLine($"secret is: '{cfg["test-secret"]}'");
    Console.WriteLine($"another secret is: '{cfg["another-secret"]}'");
}
```