2025

# Infrastructure and CI/CD setup

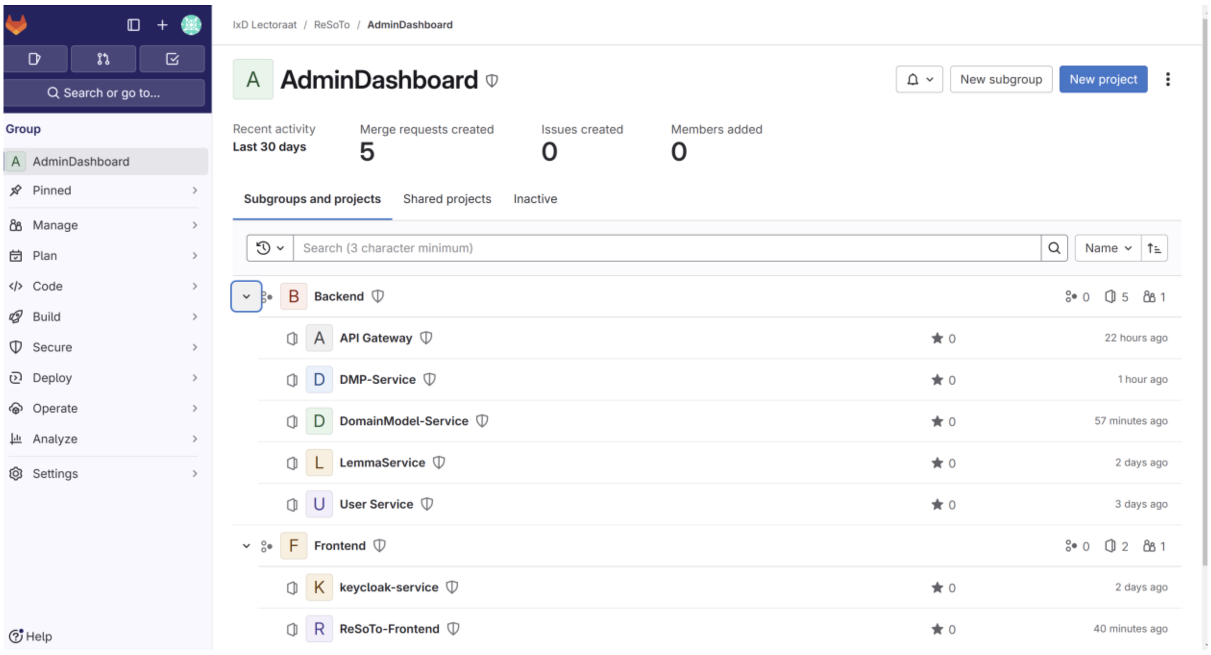
RESOTO DASHBOARD

## Table of Contents

# Repositories and CI/CD Pipelines

## Repositories

All repositories are in the gitfhict in IxD Lectoraat/ReSoTo/AdminDashboard.



IxD Lectoraat / ReSoTo / AdminDashboard

A AdminDashboard

| Recent activity Last 30 days | Merge requests created | Issues created | Members added |
|---|---|---|---|
| | 5 | 0 | 0 |

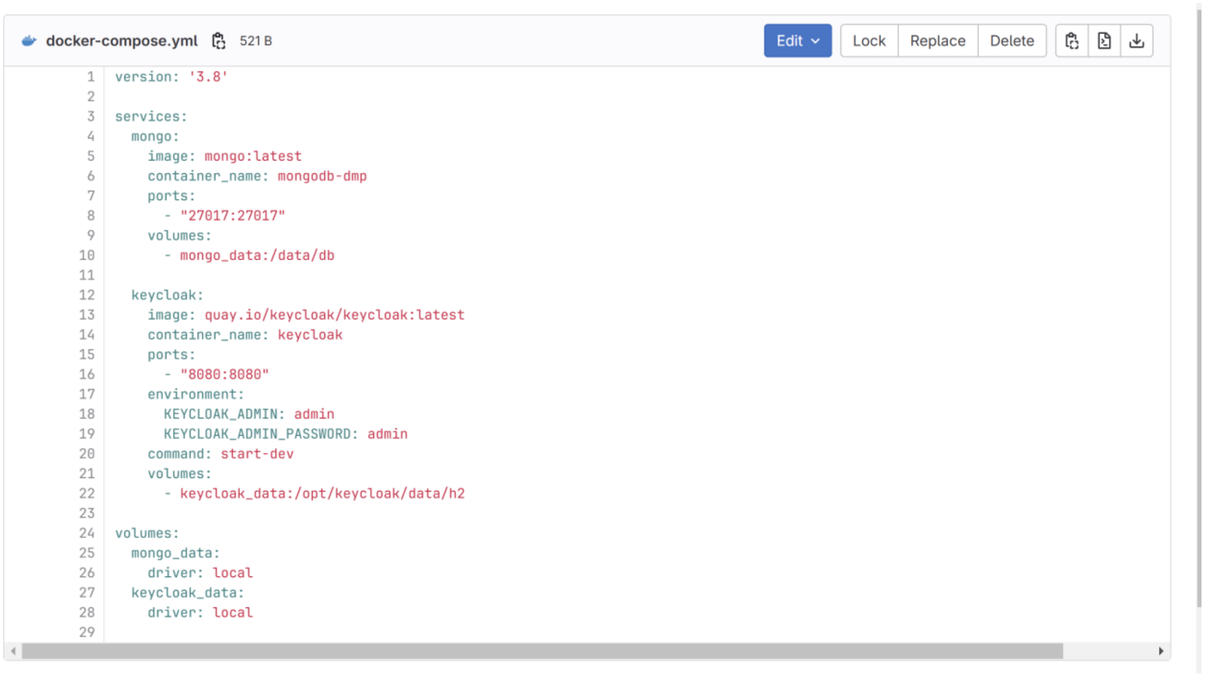
Subgroups and projects | Shared projects | Inactive

| | | | |
|---|---|---|---|
| B Backend | | | 0 · 5 · 1 |
| A API Gateway | ★ 0 | 22 hours ago | |
| D DMP-Service | ★ 0 | 1 hour ago | |
| D DomainModel-Service | ★ 0 | 57 minutes ago | |
| L LemmaService | ★ 0 | 2 days ago | |
| U User Service | ★ 0 | 3 days ago | |
| F Frontend | | | 0 · 2 · 1 |
| K keycloak-service | ★ 0 | 2 days ago | |
| R ReSoTo-Frontend | ★ 0 | 40 minutes ago | |

We have divided them in Frontend and Backend. All microservices and API Gateway are in the backend, whereas in the frontend folder, you can see Keycloak and the frontend itself.

## Application Setup

Inside the Keycloak service repository, there is a docker-compose.yml to setup the keycloak and the databases.



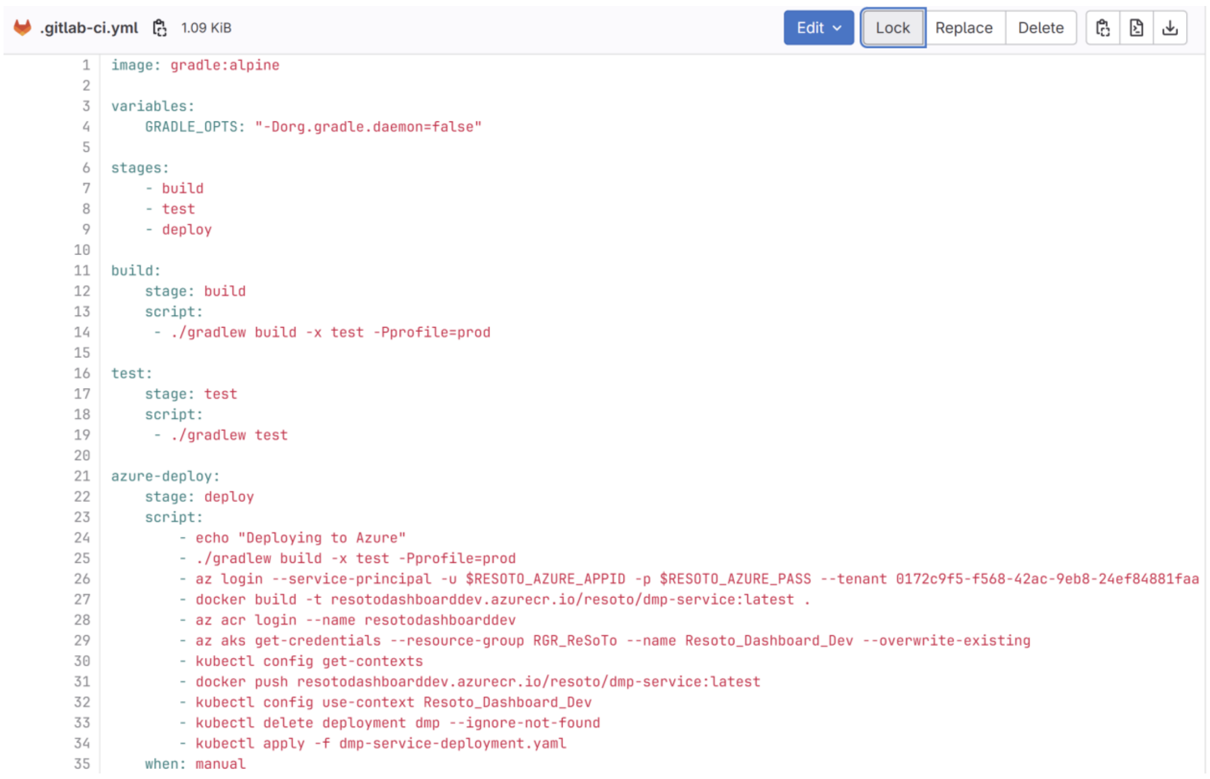
```yaml
version: '3.8'

services:
  mongo:
    image: mongo:latest
    container_name: mongodb-dmp
    ports:
      - "27017:27017"
    volumes:
      - mongo_data:/data/db

  keycloak:
    image: quay.io/keycloak/keycloak:latest
    container_name: keycloak
    ports:
      - "8080:8080"
    environment:
      KEYCLOAK_ADMIN: admin
      KEYCLOAK_ADMIN_PASSWORD: admin
    command: start-dev
    volumes:
      - keycloak_data:/opt/keycloak/data/h2

volumes:
  mongo_data:
    driver: local
  keycloak_data:
    driver: local

```

When developing locally, the services have some external dependencies and by running this script, you ensure that you have them initialized and running in Docker.

## CI/CD Pipelines

All pipelines run on local runners that you need to configure for yourself. The pipelines for all services follow the same logic.

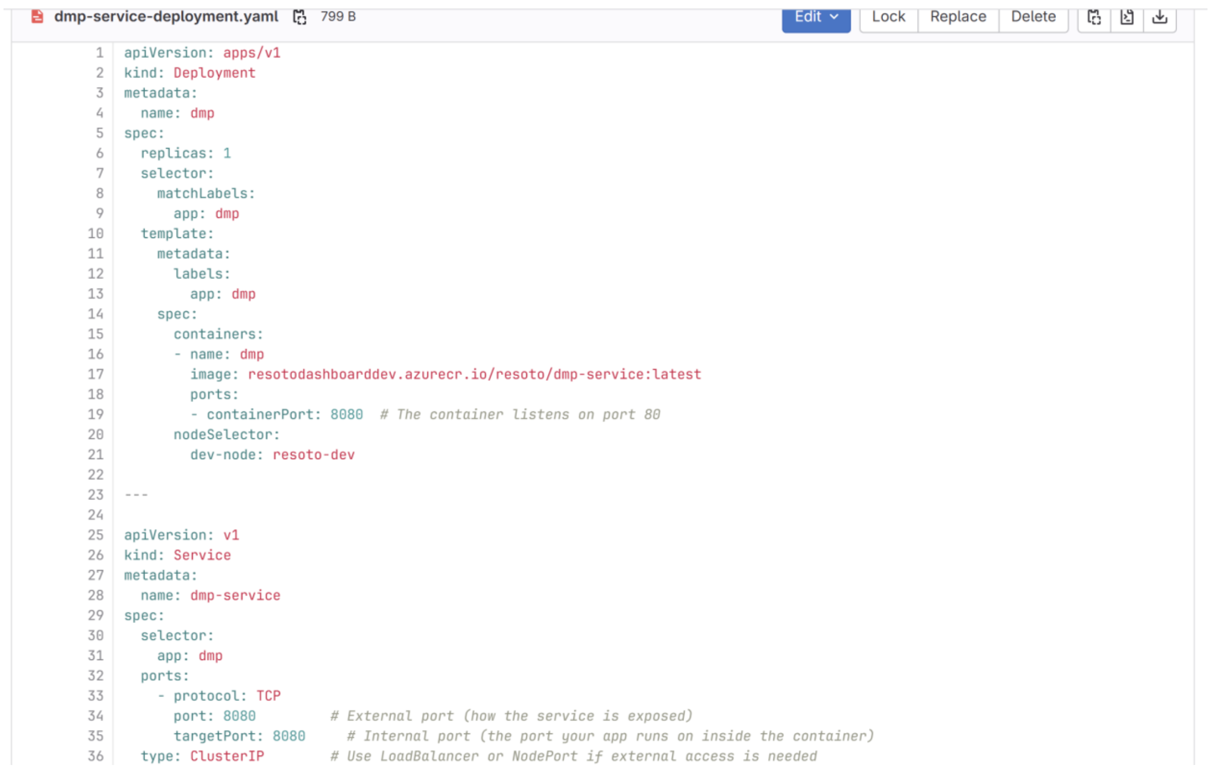
DMP Service pipeline:



```yaml
image: gradle:alpine

variables:
    GRADLE_OPTS: "-Dorg.gradle.daemon=false"

stages:
    - build
    - test
    - deploy

build:
    stage: build
    script:
     - ./gradlew build -x test -Pprofile=prod

test:
    stage: test
    script:
     - ./gradlew test

azure-deploy:
    stage: deploy
    script:
        - echo "Deploying to Azure"
        - ./gradlew build -x test -Pprofile=prod
        - az login --service-principal -u $RESOTO_AZURE_APPID -p $RESOTO_AZURE_PASS --tenant 0172c9f5-f568-42ac-9eb8-24ef84881faa
        - docker build -t resotodashboarddev.azurecr.io/resoto/dmp-service:latest .
        - az acr login --name resotodashboarddev
        - az aks get-credentials --resource-group RGR_ReSoTo --name Resoto_Dashboard_Dev --overwrite-existing
        - kubectl config get-contexts
        - docker push resotodashboarddev.azurecr.io/resoto/dmp-service:latest
        - kubectl config use-context Resoto_Dashboard_Dev
        - kubectl delete deployment dmp --ignore-not-found
        - kubectl apply -f dmp-service-deployment.yaml
    when: manual
```

The pipelines are configured in a YAML file and have 3 steps: Build, Test and Deploy. Gradle is used for building and testing, Docker for containerization, and Azure CLI for deploying to Azure.

The deploy stage begins by logging into Azure using a service principal with credentials stored in environment variables ($RESOTO_AZURE_APPID and $RESOTO_AZURE_PASS). The service principle is managed by Jeffrey Cornelissen. The pipeline then builds the Docker image with the latest code and logs into Azure Container Registry (ACR) to push the image. After pushing the image, it retrieves AKS credentials to interact with the Kubernetes cluster. The pipeline deletes any existing dmp deployment to ensure a clean setup, then applies the dmp-service-deployment.yaml to create a new deployment, specifying the image, replicas, and port configurations.



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: dmp
spec:
  replicas: 1
  selector:
    matchLabels:
      app: dmp
  template:
    metadata:
      labels:
        app: dmp
    spec:
      containers:
      - name: dmp
        image: resotodashboarddev.azurecr.io/resoto/dmp-service:latest
        ports:
        - containerPort: 8080  # The container listens on port 80
      nodeSelector:
        dev-node: resoto-dev

---

apiVersion: v1
kind: Service
metadata:
  name: dmp-service
spec:
  selector:
    app: dmp
  ports:
    - protocol: TCP
      port: 8080         # External port (how the service is exposed)
      targetPort: 8080     # Internal port (the port your app runs on inside the container)
  type: ClusterIP        # Use LoadBalancer or NodePort if external access is needed
```

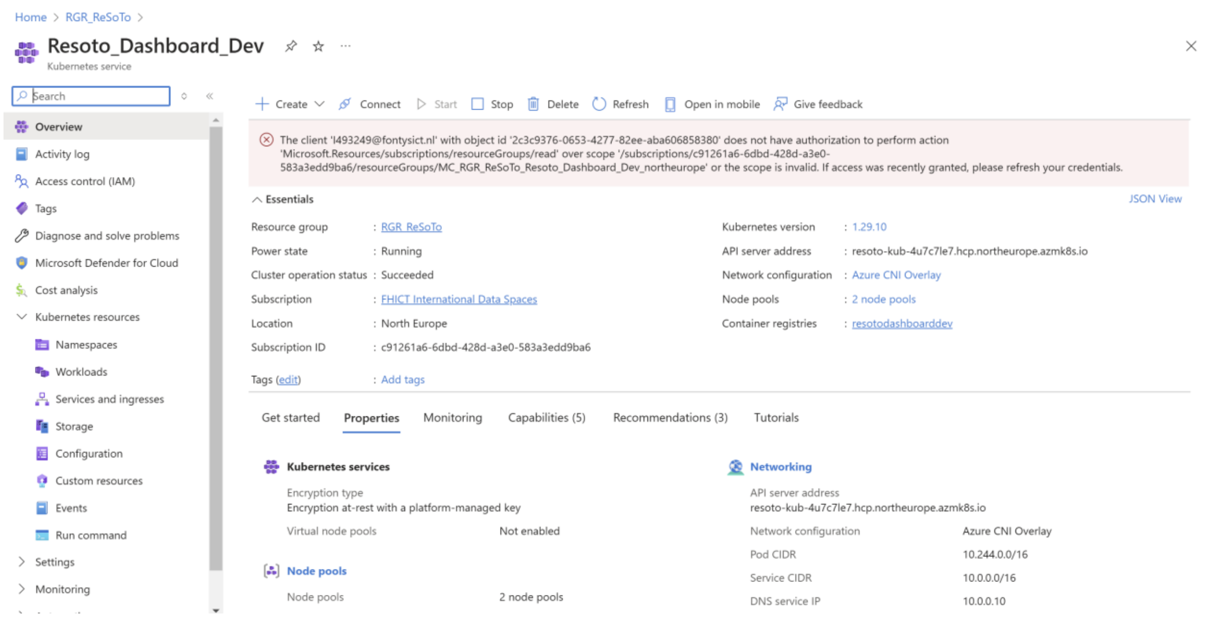
# Infrastructure in Azure

## Resources

All resources are in the RGR_ReSoTo resource group in Azure when you log in with your i-account.

### Kubernetes Cluster

All microservices, API gateway and frontend are being deployed to the cluster you see below via the pipelines.



The client 'I493249@fontysict.nl' with object id '2c3c9376-0653-4277-82ee-aba606858380' does not have authorization to perform action 'Microsoft.Resources/subscriptions/resourceGroups/read' over scope '/subscriptions/c91261a6-6dbd-428d-a3e0-583a3edd9ba6/resourceGroups/MC_RGR_ReSoTo_Resoto_Dashboard_Dev_northeurope' or the scope is invalid. If access was recently granted, please refresh your credentials.

Essentials

| | | | |
|---|---|---|---|
| Resource group | : RGR_ReSoTo | Kubernetes version | : 1.29.10 |
| Power state | : Running | API server address | : resoto-kub-4u7c7le7.hcp.northeurope.azmk8s.io |
| Cluster operation status | : Succeeded | Network configuration | : Azure CNI Overlay |
| Subscription | : FHICT International Data Spaces | Node pools | : 2 node pools |
| Location | : North Europe | Container registries | : resotodashboarddev |
| Subscription ID | : c91261a6-6dbd-428d-a3e0-583a3edd9ba6 | | |
| Tags (edit) | : Add tags | | |

**Kubernetes services**

Encryption type
Encryption at-rest with a platform-managed key

| | |
|---|---|
| Virtual node pools | Not enabled |

**Node pools**

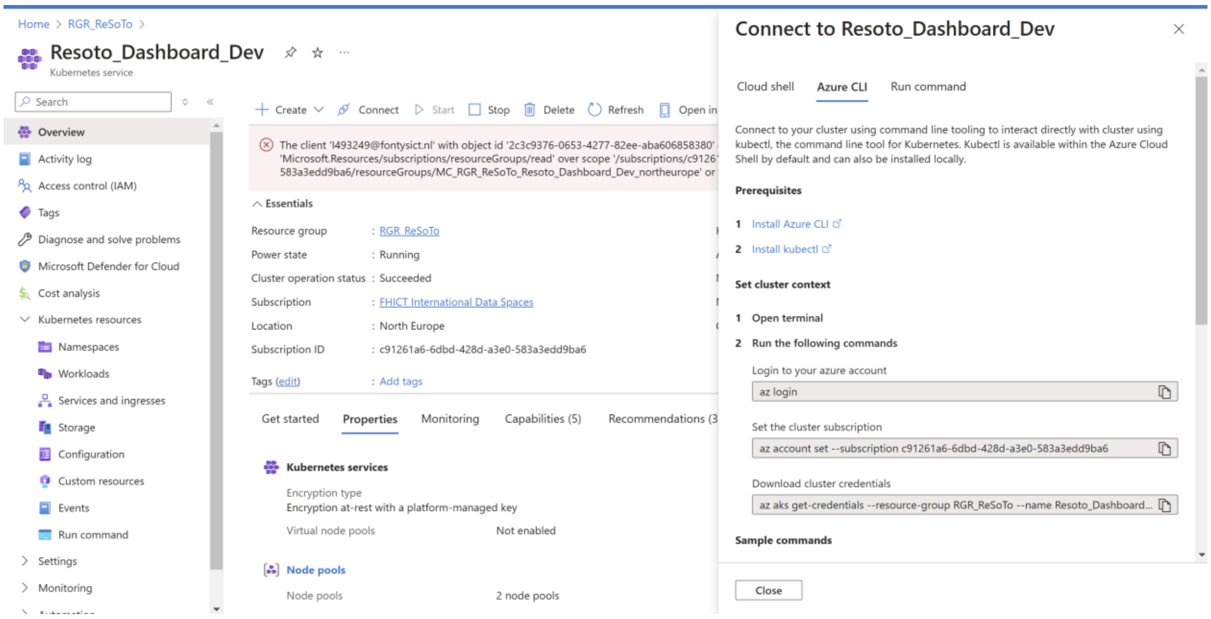
| | |
|---|---|
| Node pools | 2 node pools |

**Networking**

API server address
resoto-kub-4u7c7le7.hcp.northeurope.azmk8s.io

| | |
|---|---|
| Network configuration | Azure CNI Overlay |
| Pod CIDR | 10.244.0.0/16 |
| Service CIDR | 10.0.0.0/16 |
| DNS service IP | 10.0.0.10 |

To connect to the cluster, you can follow the steps that appear when clicking "Connect".



Connect to Resoto_Dashboard_Dev

Connect to your cluster using command line tooling to interact directly with cluster using kubectl, the command line tool for Kubernetes. Kubectl is available within the Azure Cloud Shell by default and can also be installed locally.

**Prerequisites**

1 Install Azure CLI
2 Install kubectl

**Set cluster context**

1 Open terminal
2 Run the following commands

Login to your azure account

```
az login
```

Set the cluster subscription

```
az account set --subscription c91261a6-6dbd-428d-a3e0-583a3edd9ba6
```

Download cluster credentials

```
az aks get-credentials --resource-group RGR_ReSoTo --name Resoto_Dashboard...
```

**Sample commands**

If you were to start it from 0, please follow these steps:

1. Install Ingres controller
2. Apply Ingres configuration (ingres-deployment-host-route.yaml)
3. Install certificate manager

helm repo add jetstack https://charts.jetstack.io

helm repo update

helm install cert-manager jetstack/cert-manager --namespace cert-manager --create-namespace --set crds.enabled=true

4. Apply the Clusterissuer

kubectl apply -f clusterissuer.yaml

5. Apply certificate

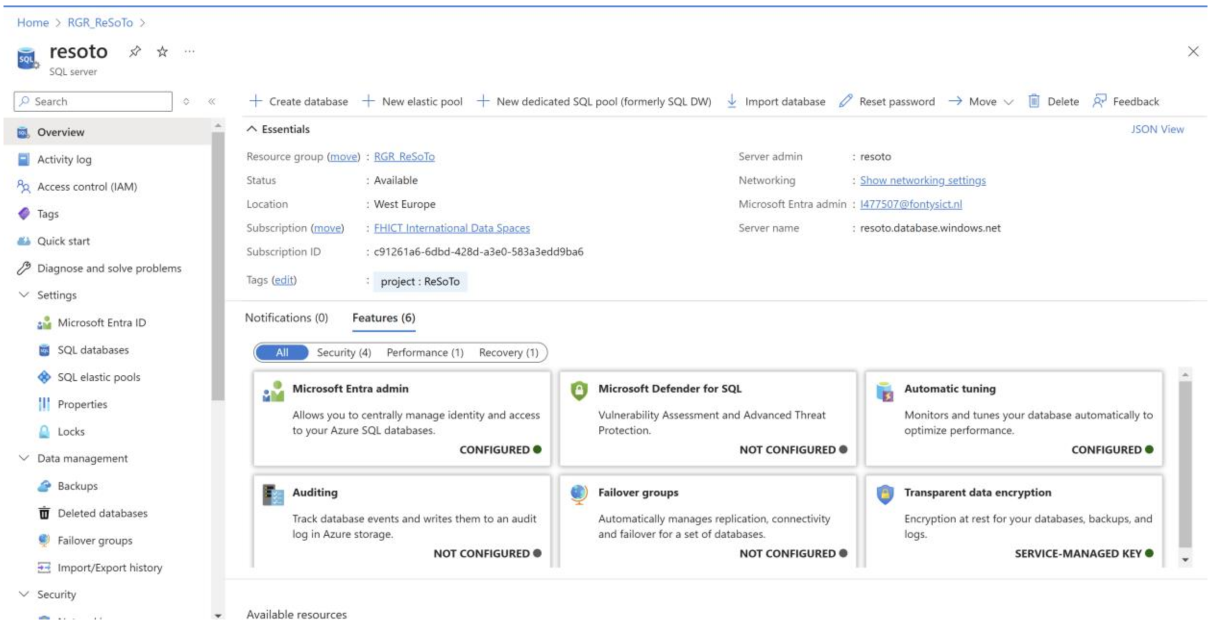
kubectl apply -f certificate.yaml

6. Reapply the Ingres controller

The files is present in the repository in the application_setup in API Gateway.

## SQL Server

We have created an SQL server as well with a users database inside of it.



Resource group (move) : RGR_ReSoTo
Status : Available
Location : West Europe
Subscription (move) : FHICT International Data Spaces
Subscription ID : c91261a6-6dbd-428d-a3e0-583a3edd9ba6
Tags (edit) : project : ReSoTo
Server admin : resoto
Networking : Show networking settings
Microsoft Entra admin : I477507@fontysict.nl
Server name : resoto.database.windows.net

Microsoft Entra admin
Allows you to centrally manage identity and access to your Azure SQL databases.
CONFIGURED ●

Microsoft Defender for SQL
Vulnerability Assessment and Advanced Threat Protection.
NOT CONFIGURED ●

Automatic tuning
Monitors and tunes your database automatically to optimize performance.
CONFIGURED ●

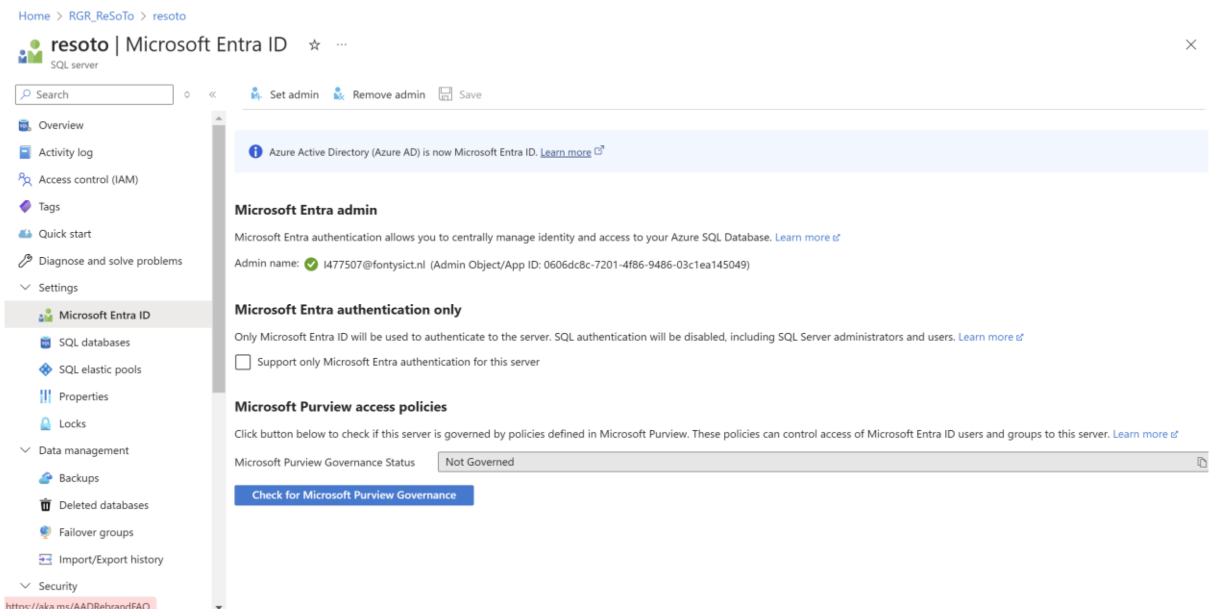
Auditing
Track database events and writes them to an audit log in Azure storage.
NOT CONFIGURED ●

Failover groups
Automatically manages replication, connectivity and failover for a set of databases.
NOT CONFIGURED ●

Transparent data encryption
Encryption at rest for your databases, backups, and logs.
SERVICE-MANAGED KEY ●

To be able to login to the server, you should follow the following steps:

1. Set yourself as an admin via the Microsoft Entra Id in the server:


Home > RGR_ReSoTo > resoto

resoto | Microsoft Entra ID
SQL server

Set admin · Remove admin · Save

Azure Active Directory (Azure AD) is now Microsoft Entra ID. Learn more

**Microsoft Entra admin**

Microsoft Entra authentication allows you to centrally manage identity and access to your Azure SQL Database. Learn more

Admin name: ✓ l477507@fontysict.nl (Admin Object/App ID: 0606dc8c-7201-4f86-9486-03c1ea145049)

**Microsoft Entra authentication only**

Only Microsoft Entra ID will be used to authenticate to the server. SQL authentication will be disabled, including SQL Server administrators and users. Learn more

☐ Support only Microsoft Entra authentication for this server

**Microsoft Purview access policies**

Click button below to check if this server is governed by policies defined in Microsoft Purview. These policies can control access of Microsoft Entra ID users and groups to this server. Learn more

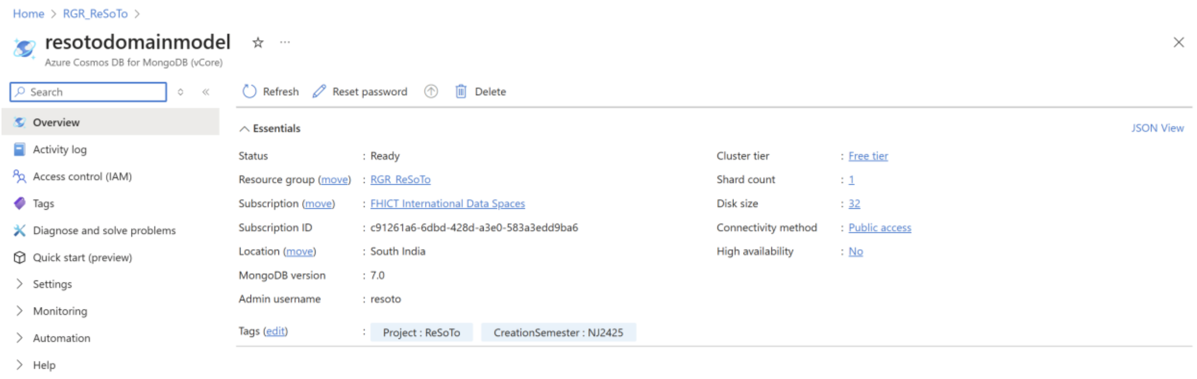
Microsoft Purview Governance Status: Not Governed

2. Open SQL Management Studio and use the server name with SQL Server Authentication with the following credentials:

   Username: resoto

   Password: Amazing123*()

## MongoDB


Home > RGR_ReSoTo >

resotodomainmodel
Azure Cosmos DB for MongoDB (vCore)

Refresh · Reset password · Delete

Essentials

| | | | |
|---|---|---|---|
| Status | : Ready | Cluster tier | : Free tier |
| Resource group (move) | : RGR_ReSoTo | Shard count | : 1 |
| Subscription (move) | : FHICT International Data Spaces | Disk size | : 32 |
| Subscription ID | : c91261a6-6dbd-428d-a3e0-583a3edd9ba6 | Connectivity method | : Public access |
| Location (move) | : South India | High availability | : No |
| MongoDB version | : 7.0 | | |
| Admin username | : resoto | | |
| Tags (edit) | : Project : ReSoTo · CreationSemester : NJ2425 | | |

To log in use the credentials:

Username: resoto

Password: RstAdmin01

## Keycloak

To setup Keycloak, please go to the Keycloak repository and follow the steps in the README file.