

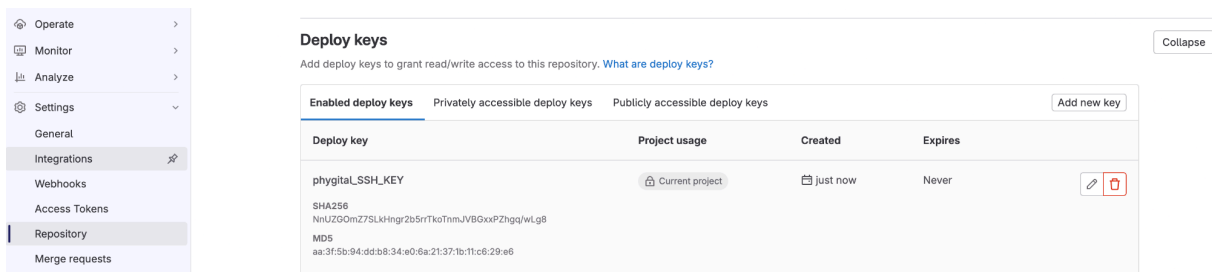
# CONNECTIVITY

## Connectie naar een GitLab private repository

- met ofwel HTTPS of SSH. de voorkeur heeft de SSH keypairs
- je maakt aparte SSH keys aan die gebruikt kunnen worden om toegang te krijgen tot de repo

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

- je zorgt ervoor dat zowel op de remote als de lokale repo de public keys hetzelfde zijn
- vervolgens voeg je de aangemaakte public key toe in het gitlab in "deploy keys"



Deploy key	Project usage	Created	Expires
phygital_SSH_KEY SHA256 NnUZG0mZ7SLkHngR2b5rTkoTnmJVB6xxPZhga/wLg8 MD5 aa:3f5b:94:dd:b8:34:e0:6a:21:37:1b:11:c6:29:e6	Current project	just now	Never

- erna gebruik je de SSH optie voor de rep aan te spreken zoals in het volgende voorbeeld

```
git clone git@gitlab.com:kdg-ti/integratieproject-1/202324/8_mf_i/dotnet.git
```

## Connectie Applicatie naar Databank

ook hier 2 opties: publiek IP of private IP

optie: publiek IP

via een publiek IP krijg je een publiek accessible IP adres die je kan gebruiken in je connectionstring naar de databank

om de connectie toe te laten van de applicatie naar de databank moet je in de instellingen een network instellen



Connections

SUMMARY

**NETWORKING**

SECURITY

CONNECTIVITY TESTS

Choose how you want your source to connect to this instance, then define which networks are authorized to connect. [Learn more](#)

You can use the Cloud SQL Proxy for extra security with either option. [Learn more](#)

### Instance IP assignment



Private IP

Assigns an internal, Google-hosted VPC IP address. Requires additional APIs and permissions. Can't be disabled once enabled. [Learn more](#)



Public IP

Assigns an external, internet-accessible IP address. Requires using an authorized network or the Cloud SQL Proxy to connect to this instance. [Learn more](#)

### Authorized networks

You can specify CIDR ranges to allow IP addresses in those ranges to access your instance. [Learn more](#)



You have not authorized any external networks to connect to your Cloud SQL instance. External applications can still connect to the instance through the Cloud SQL Proxy. [Learn more](#)

ADD A NETWORK

optie Private IP

om de private IP optie te gebruiken moeten er eerst een aantal andere dingen gebeuren om dit toe te laten in zowel Compute Engine als Cloud SQL

eerst en vooral moet de API optie van service networking geactiveerd worden, deze API biedt een automatische configuratie van het netwerk waarin de VM(s) opereert / opereren.

vervolgens moet je binnen je project de juiste IAM permissies aan zetten om de communicatie toe te laten

create a new VPC (=virtual private cloud) network

een VPC is een privaat network waarbinnen je project draait.  
dit biedt betere administratie, overview en bovenal veiligheid

om de connectie tussen de VM en de Cloud SQL te doen werken zorg je ervoor dat deze elkaar kunnen “zien” binnen het netwerk door IP subnet en ranges toegang te geven via firewalls

je kan zowel custom (zelf de subnets bepalen), alsook automatisch per region

## [←](#) Create a VPC network

Name \*



Lowercase letters, numbers, hyphens allowed

Description



Maximum transmission unit (MTU)

1460



### VPC network ULA internal IPv6 range ?

Enabling this feature will assign a /48 from Google-defined ULA prefix fd20::/20.

☐ Enabled

☒ Disabled

## Subnets

Subnets let you create your own private cloud topology within Google Cloud. Click Automatic to create a subnet in each region, or click Custom to manually define the subnets. [Learn more](#)

### Subnet creation mode ?

☒ Custom

☐ Automatic

### New subnet



Name \*



Lowercase letters, numbers, hyphens allowed

Description



Region \*



### IP stack type

☒ IPv4 (single-stack)

☐ IPv4 and IPv6 (dual-stack) ?

## Connectie andere cloud componenten

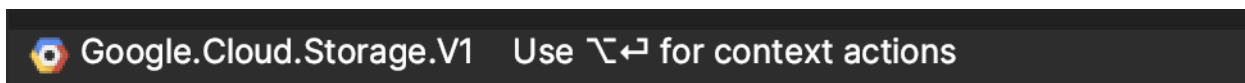
Een cloud storage bucket wordt gebruikt als een opslag voor verschillende soorten gegevens, waaronder media zoals afbeeldingen, video's, enzovoort.

In samenwerking met een cloud engine VM kan een cloud storage bucket worden gebruikt om media op te slaan en te delen tussen verschillende virtuele machines of andere services in de cloud.

Hoe dit juist in zijn werkt gaat is door de bucket steeds aan te spreken wanneer er iets opgeslagen moet worden of opgehaald moet worden.

Dit kan door de Cloud Storage te mounten als een Drive met behulp van Cloud Storage FUSE, of door de bucket aan te spreken vanuit het publieke IP adres. Opnieuw is het hier nodig om de juiste IAM permissies op te stellen zodat enkel de Cloud Engine service account dit kan doen

Om dit te implementeren binnen de .NET applicatie heb je een NuGet package nodig:



Vervolgens gebruik je een interface om de opslag te implementeren, alsook een service in de Program.cs

## Hoe de Google Cloud Metadata server gebruiken?

De Google Cloud Metadata server bevat alle metadata van een project. Metadata zijn dingen zoals login-credentials voor de databank, startup script en andere informatie die over het hele project beschikbaar moet zijn en bovenal makkelijk op te vragen is.

Omdat de vooraf gedefinieerde metadata keys van Compute Engine hetzelfde zijn voor elke VM, kun je je script hergebruiken zonder het voor elke VM te hoeven bijwerken.

Vandaar dat er een centraal punt wordt ingesteld waar je deze informatie kan opslaan en opvragen, dit is de Metadata server.

de data kan vanaf de VM makkelijk opgevraagd worden op basis van curl commando's. je hebt wel steeds de metadata key waarde nodig om dit te doen

## Links

[gitlab with SSH keys tutorial](#)

[Google Cloud Storage in Dotnet  
metadata querying](#)

# Risico Analyse

## 1. Downtime Google Cloud:

Risico: Google Cloud ervaart ongeplande downtime, waardoor de applicatie niet beschikbaar is voor gebruikers.

Mitigatie: In deze context moeilijk te voorzien, maar met de reputatie van Google is dit het risico met de kleinste kans dat het voorkomt. Een optie om ook een 2de cloud provider te voorzien. Maar dat is in deze context niet mogelijk.

## 2. Applicatie wordt getroffen door een cyber aanval:

Risico: Door een verkeerde configuratie vinden hackers een way in en brengen de applicatie down.

Mitigatie: Goed doordachte firewall regels, centrale en goed beveiligde storage van credentials en gebruik van sterke wachtwoorden.

## 3. Onverwachte piek in applicatie trafiek

Risico: Door ongekend succes is er een hoge trafiek naar de applicatie. De load balancer kan het verkeer niet langer verwerken en de applicatie crasht.

Mitigatie: Voldoende extra VMs voorzien zodat het piekverkeer opgevangen kan worden. Zonder dat er ook te diep gesneden wordt in de beschikbare credits. Dus voorzie meerdere kleinere (qua CPU performantie) VMS die matchen met de computing power die nodig is om een simpele applicatie als deze te runnen

## 4. Databank crasht

Risico: Een applicatie kan crashen, maar een databank kan dat ook. Het is misschien het belangrijkste onderdeel van het hele project. Aangezien de belangrijke data die de lokale instellingen willen vergaren daar opgeslagen is. De applicatie kan makkelijk opnieuw uitgerold worden. Een databank opnieuw opbouwen vraagt tijd

Mitigatie: Voldoende en frequente backups maken van de databank. Liefst dagelijks.

## 5. Over budget met Google Cloud credits

Risico: De applicatie stopt met werken door het opgebruiken van de credits

Mitigatie: Voldoende Budget notificaties zetten zodat het gebruik makkelijk kan opgevolgd worden

## 6. Menselijke fout

Risico: Ikzelf, als verantwoordelijke voor de deployment, maak een fout die de finale uitrol van de applicatie in gedrang kan brengen.

Mitigatie: Back up everything, heb een back up plan en documenteer zo veel mogelijk zodat je snel terug op het punt bent waar het fout ging. Waar ik dan liefst niet dezelfde fout 2 keer maak.



