* Heb je de usecase van je bedrijf (datalake etc) beter in kaart kunnen brengen?
  + Data van Microsoft 365 Customer Engagement of Power Platform wordt geëxporteerd naar CSV bestanden en in Azure Data Lake geplaatst (1 grote pool die opgesplitst moet worden per groep per jaar)
  + Minstens 1 keer per dag:
    - Opkuisen
    - Aanvullen
    - Opsplitsen
    - Ten slotte: Doorsturen naar de klant via SFTP of email (CSV bestanden of parquet files)
* Heb je al een idee van concrete vergelijkingscriteria? (stel een lijst op van meetbare/aantoonbare zaken)
  + Meetbaar:
    - Kostprijs
    - Performantie
    - Mogelijkheid tot debuggen (kan aangetoond worden met een voorbeeld?)
    - Verschil in implementatietijd? (kan verschillen door ervaring?)
  + Minder gemakkelijk meetbaar:
    - Moeilijkheidsgraad in opzet
      * Alles via source control
      * Azure Pipelines om te deployen
      * Moeilijker om te deployen 🡪 minder goeie score
    - Mogelijkheden van de tool (wat is de feature set?)
    - Onderhoudbaarheid
    - Testbaarheid
* Wat is je doelstelling tegen volgende meeting? (wat wil je over 2 weken bereikt hebben)

Er bestaan on-premises tools maar er worden ook cloud etl tools aangeboden, hierdoor is er geen pc nodig

Literatuurstudie

* Wat zijn ETL’s of ELT’s?
* Wat is het verschil tussen ETL en ELT?
* Welke verschillende soorten ETL tools bestaan er?
  + On-premises
  + Open-source
  + Cloud-based
  + Hybrid
* ??? Kort per soort ETL tool de meest populairste opties opnoemen
* ??? Uitleggen waarom bij Net IT er gekozen zal worden voor implementatiemogelijkheden binnen Azure
* ??? Verschillen uitleggen tussen de implementatiemogelijkheden binnen Azure

Vragen:

* Kan documentatie gebruikt worden als bron? Hoe kan dit gebruikt worden?
* Moeten/mogen er tussentitels in de literatuurstudie/stand van zaken zitten?
* Kan een online artikel zonder auteur gebruikt worden? Vaak is dit een bedrijf/organisatie die het artikel publiceerde.

Vragen voor Net IT:

* Waarom wordt de data naar de klant doorgestuurd?

Vragen 15/03:

# Feedback 01/03

zoals terraform bekijken

wat moet er in literatuurstudie?

data factory moet beschreven worden in de literatuurstudie

cli, pipelines etc moeten beschreven worden

elk onderwerp dat ik aanraak moet "gedocumenteerd" worden

specifieker gaan -> uitleggen wat hjet is (uitleggen wat een datalake is)

kan ik connecteren met andere datalakes ook?

auteur: microsoft, geraadpleegd op voor datum

er mogen tussentitels, moet leuk te lezen zijn

Welke IDEs?

Monitoring -> Azure monitoring

Workspace losgekoppeld van persoon

Verschillende soort computes uitzoeken

19/04 feedback

nog niet tastbaar genoeg, niet iedereen kent het (je kan queries doen maar wat wordt er mee bedoeld?)

Titel aanpassen, waar is ELT?

Goeie voorbeeldjes voorzien zodat de lezer goed mee is met wat er gebeurd

Automatisatiemogelijkheden, infrastructure as code?

elke afbeelding moet een voetnoot hebben

screenshots zijn te klein

Volledige implementaties in bijlage toevoegen, belangrijkste in bachelorproef

oefenpresentatie vragen !!!

Feedback Koen:

Details vergelijken met elkaar, ik heb de ene transformatie hier en die voer ik zo uit (in adf) en bij databricks doe ik het zo, die in detail gaan vergelijken

# Creation

## Databricks

Managed Resource group source: https://learn.microsoft.com/en-us/answers/questions/1332779/azure-databricks

# Source control

## ADF

* Data Factory service geeft geen repository voor het opslaan van JSON entiteiten voor wijzigigen, de enigste manier op wijzigingen op te slaan is via de **Publish All** knop en alle wijzigingen worden direct gepubliceerd naar data factory service
* Data Factory service is niet geoptimaliseerd voor samenwerking en version control
* Azure Resource Manager (ARM) template voor het deployen van Data Factory is niet inbegrepen
* Azure Data Factory laat het toe om een Git repository te configureren via Azure Repos of Github

Voordelen :

* Mogelijkheid om wijzigen bij te houden/controleren
* Mogelijkheid om wijzigingen die bugs hebben geïntroduceerd ongedaan te maken
* **Gedeeltelijk opslaan:** wijzigingen kunnen niet als concept opgeslaan worden en alle publicaties moeten datafactory-validatie doorstaan
* **Samenwerking en controle:** elke contributor kan bepaalde rechten krijgen, sommige team members zullen bijvoorbeeld geen changes via git kunnen maken terwijl andere team members die wel zullen kunnen
* **Betere CI/CD:** er kunnen release pipelines gebruikt worden
* **Betere performance:** een factory met git kan 10 keer sneller laden, dit doordat resources via Git gedownload worden

Elke Azure Repos Git repository dat is geassocieerd met data factory heeft een collaboration branch (default: main). Daarnaast kunnen gebruikers ookfeature branches aanmaken. Met behulp van ene pull request kunnen deze feature branches dan gemerged worden met de collaboration branch. Er kan enkel gepubliceerd worden naar Data Factory service vanaf de collaboration branch.

Standaard genereert data factory Resource Manager templates van de gepubliceerde factory naar een branch met de naam *adf\_publish*. (kan gewijzigd worden)

Data factory slaat geen secrets op in git. Aanpassingen in Linked Services waarin secrets zoals passwords zitten worden direct gepubliceerd naar Azure Data Factory service. Het gebruik van een Key Vault of MSI authentication maakt dus continuous integration en deployment makkelijker doordat er geen rekening meer gehouden moet worden met deze secrets tijdens Resource Manager deployment

Source: <https://learn.microsoft.com/en-us/azure/data-factory/source-control>

Azure Data Factory access geven aan andere gebruikers -> **Data Factory Contributor** role geven bij de **Resource Group** van de Data Factory

* Create, edit & delete data factories en child resources including datasets, linked services, pipelines, triggers, en integration runtimes
* Deploy Resource Manager templates
* Manage App Insight alerts voor data factory
* Aanmaken van support tickets

Permissions bij Azure Repos en GitHub :

* Afhankelijk van Data Factory permissions
* Gebruiker die enkel member is van de Reader role kan enkel Data Factory child resources wijzigen en commits uitvoeren op de repo maar niet publishen

Custom roles kan ook maar die skippen we ff