

# Inhoudsopgave

<b>1</b>	<b>Voorwoord</b>	<b>2</b>
<b>2</b>	<b>Defensie</b>	<b>2</b>
2.1	Stageplaats	3
<b>3</b>	<b>Opdrachten</b>	<b>6</b>
3.1	Backup-script moderniseren	6
3.2	Procedure upgrade mariadb	7
3.3	High availability bij databanken	8
3.4	MS SQL logins	9
<b>4</b>	<b>Eindreflectie</b>	<b>10</b>
4.1	Voldoet de stage aan hetgeen je verwacht had?	10
4.2	Overwelke opdracht ben je het meest trots?	10
4.3	Is dit het toekomstbeeld dat je voor ogen hebt?	10
4.4	Vind je van jezelf dat je genoeg ervaring hebt opgedaan om nu in het werkveld te stappen?	10
	<b>Verklarende woordenlijst</b>	<b>11</b>
	<b>Referenties</b>	<b>12</b>
<b>A</b>	<b>Stagedagboek</b>	<b>13</b>
A.1	Week 1	13
A.2	Week 2	14
A.3	Week 3	14
A.4	Week 4	15
A.5	Week 5	16
A.6	Week 6	17
A.7	Week 7	18
A.8	Week 8	18
A.9	Week 9	18
A.10	Week 10	19
A.11	Week 11	20
A.12	Week 12	21
A.13	Week 13	22
A.14	Week 14	22
<b>B</b>	<b>Procedure: Upgrade Mariadb</b>	<b>23</b>
<b>C</b>	<b>MariaDB: Replication installation guide</b>	<b>24</b>
<b>D</b>	<b>Kleine opdrachtjes</b>	<b>27</b>
<b>E</b>	<b>Script: bu_script</b>	<b>29</b>
<b>F</b>	<b>Script: check_users</b>	<b>34</b>



Figuur 1: Filip Leopold Lodewijk Maria Van België, de zevende koning der belgen (Wikipedia, 2022b)

## 1 Voorwoord

Dit stageverslag werd geschreven in het kader van de stage die ik liep bij Defensie van 21/02/2022 tot 27/05/2022. Dit was een zeer interessante periode omdat ik hier veel bijgeleerd heb.

Zoals vele andere dingen in het leven, kon ik dit niet alleen en daarom wil ik graag een aantal mensen bedanken. Steven beeckman, bedankt om mij zo goed te ontvangen bij Defensie, ook voor de vragen die ik had voor zowel de stage als bachelorproef kon ik steeds bij jou terecht. Tom De Leeuw, mijn stagementor, bedankt om mij te begeleiden en te helpen doorheen mijn stage. Wim De bruyn, mijn stagebegeleider, bedankt voor de opbouwende feedback en begeleiding tijdens de stage. Tot slot, Britt, die mij ondertussen al 3 jaar aanzet om het beste uit mezelf te halen.

Ik wens u veel leesplezier.

Pieter Van Keer  
Dendermonde, 27 mei, 2022

## 2 Defensie

Defensie, of ook wel het Belgisch leger, staat in voor de bescherming van de strategische belangen van België. De opperbevelhebber van Defensie is koning Filip (zie Figuur 1), Minister van Defensie is Ludivine Dedonder (zie Figuur 2) en de chef Defensie, de stafchef van de Generale staf is admiraal Michel Hofman (zie Figuur 3) (Wikipedia, 2022a)

Defensie telt 26.179 personeelsleden, verdeeld over vier componenten: Landcomponent, Luchtcomponent, Marine en Medische Component. Daarnaast werken ze ook op de verschillende algemene directies en stafdepartementen, onder leiding van de chef Defensie: admiraal Michel Hofman. Zie figuur 4

De personeelsleden van Defensie oefenen meer dan 300 verschillende beroepen uit. Dat maakt van Defensie een van de meest diverse werkgevers in het land, met zowel Nederlandstalige (54 %) als Franstalige (46 %) profielen. Militairen vormen de bulk van het personeelsbestand. Sommige functies vereisen



Figuur 2: Ludivine Dedonder, minister van Defensie (Wikipedia, 2022c)



Figuur 3: Admiraal Michel Hofman, chef Defensie (Nato, 2020)

echter geen uniform: bijna 6 % ervan wordt ingevuld door burgerpersoneel. (Defensie, 2022b)

## 2.1 Stageplaats

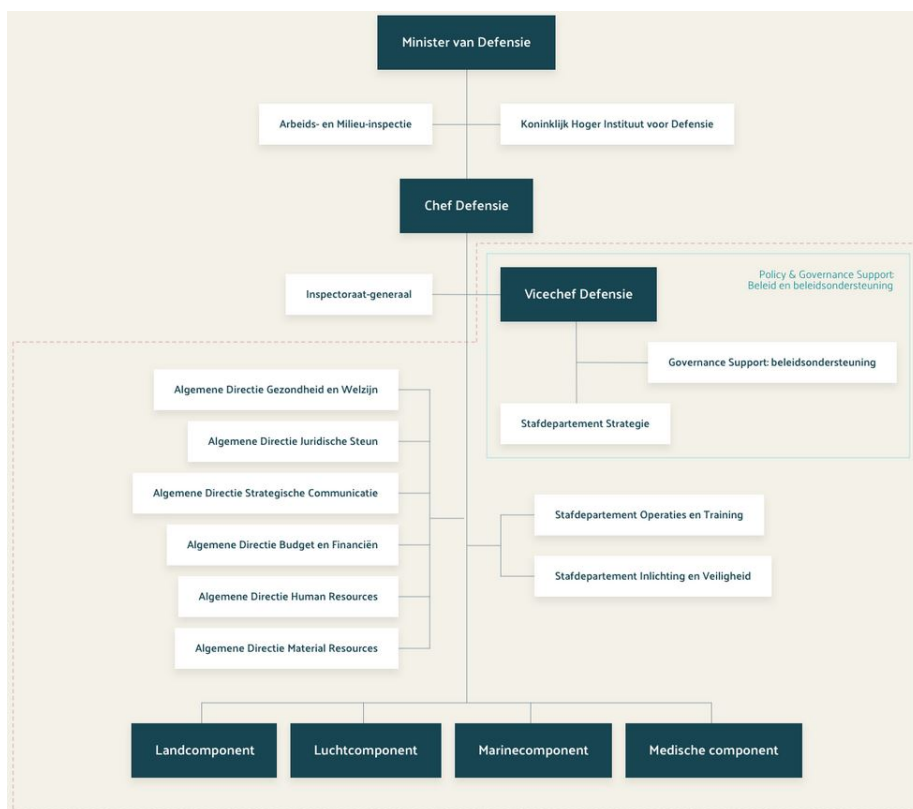
Defensie heeft verschillende kwartieren, mijn stage vond plaats binnen de eenheid CC V&C op het kwartier Majoor Housiau gelegen te Peutie. Zie Figuur 5

Mijn stage was bij de afdeling Systems binnen het Departement IT&S. Zie figuur 6

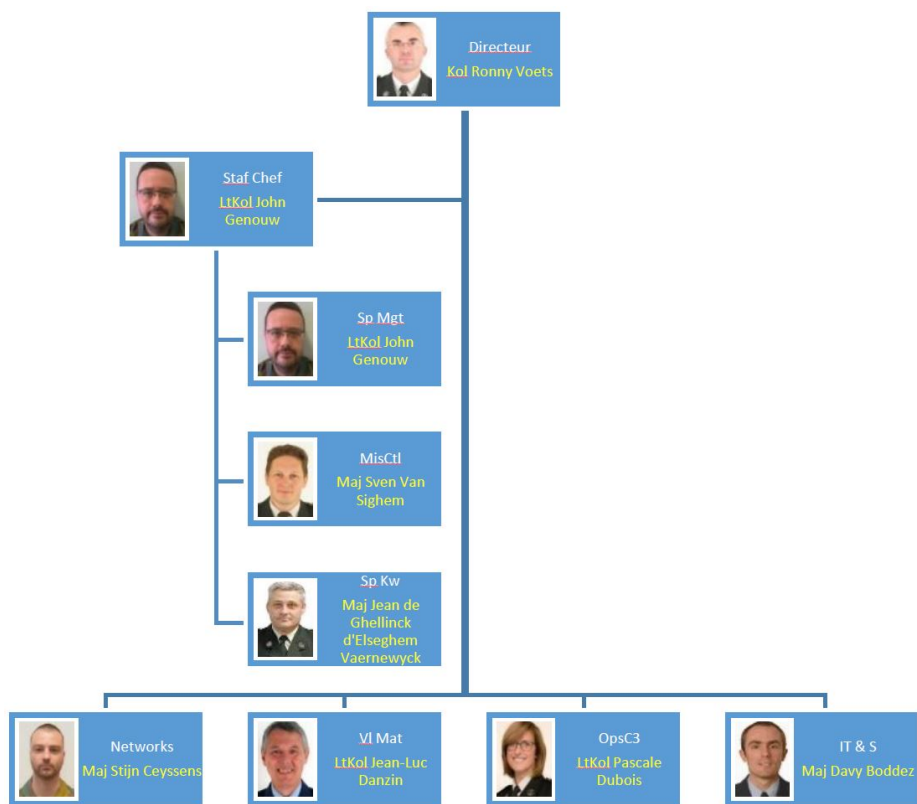
Binnen de afdeling Systems zijn er 6 diensten:

- linux
- windows
- oracle databanken
- Microsoft sql server
- prodctl
- techbu

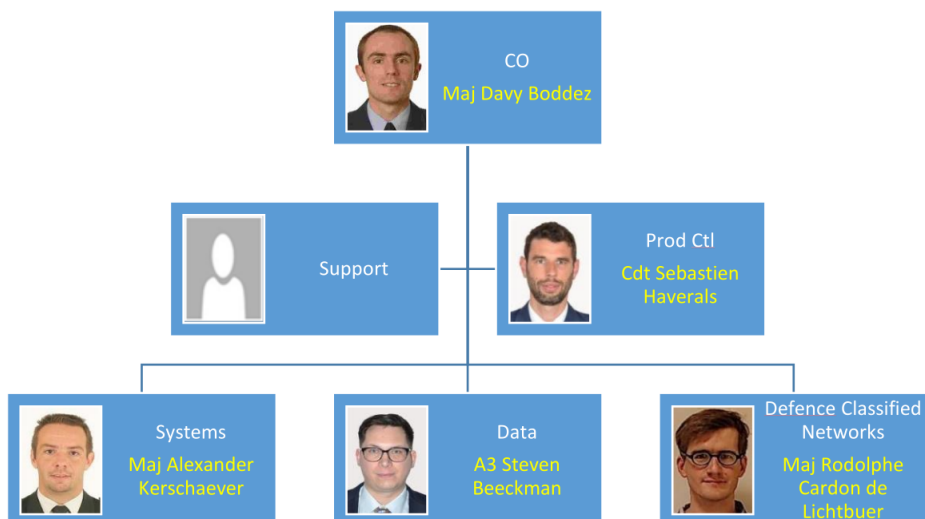
Mijn stage was binnen de dienst linux, onder leiding van Tom De Leeuw. Hier kwam ik in een team van drie personen die zich bezigheden met het beheer van de linux servers. Specifiek staat dit team in voor de monitoring, provisioning, patching van de linuxservers (databank-, applicatieservers) binnen Defensie. Ik heb gekozen om bij Defensie stage te lopen omdat ik ambitie heb om later bij



Figuur 4: Organigram van Defensie (Defensie, 2022b)



Figuur 5: Organigram van de eenheid CC V&C (Defensie, 2022a)



Figuur 6: Organigram van het departement IT&S (Defensie, 2022a)

Defensie te gaan werken. Via deze weg kan ik al eens proeven hoe het er op de werkvloer aan toe gaat.

## 3 Opdrachten

### 3.1 Backup-script moderniseren

Binnen Defensie gebruikt men een bashscript genaamd “bu\_script.sh” om een backup te nemen van de databanken die bestaan op een machine en deze weg te schrijven op het netwerk. Ik kreeg de opdracht om dit script te moderniseren. Om de opdracht een beetje te vereenvoudigen moest ik de backup’s lokaal wegschrijven. Het script is te vinden in Bijlage E

#### 3.1.1 Beginsituatie

Het script bestaat maar het is verouderd. Met het script is het mogelijk om meerdere instances van Mariadb/mysql te backupperen. Het script is zeer lang en moeilijk leesbaar.

#### 3.1.2 Doel

- moderniseer het script
- maak het script meer flexibel door gebruik te maken van meer variabelen
- genereer een rapport van de uitvoering van het script
- schrijf de backup lokaal weg onder de map “~/backup”

#### 3.1.3 Plan van Aanpak

1. lees het script door en probeer de denkwijze te begrijpen van de persoon die het script geschreven heeft
2. verwijder delen die niet meer van toepassing zijn
3. voeg meer commentaar toe aan het script zodat iemand die het script later zal doornemen sneller weet wat er gebeurt
4. gebruik meer functies aan zodat het script meer leesbaar wordt
5. test het script

#### 3.1.4 Uitwerking

Ik ben begonnen met het script door te nemen en de logica proberen te verstaan. Tijdens dat ik dit deed heb ik extra commentaar geschreven om duidelijk te maken wat er juist gebeurde in het script. Na een kort gesprek met Tom hebben we besloten om het script enkel te beperken voor server die maar 1 instance hebben van mariadb/mysql. Hierdoor werd het script al veel meer leesbaar. Door extra hulpfuncties toe te voegen is de leesbaarheid ook verbeterd. Tijdens het testen heb ik er nog enkele bugs uitgehaald maar het script werkt zoals gevraagd.

### 3.1.5 Eindresultaat

Het eindresultaat is een script dat makkelijker te lezen/verstaan en flexibel is. Het script gaat een rapport genereren van de uitvoering en de backup was te vinden in de map “~/backup”. Er is mogelijkheid om in de toekomst het script uit te breiden zodat dit een melding gaat genereren in Zabbix (monitoring tool).

### 3.1.6 Business doelstellingen

Voor de business was het belangrijk dat het script een moderne update kreeg. Dit was zeker het geval.

### 3.1.7 Persoonlijke doelstellingen

Voor mij persoonlijk was dit een geslaagde opdracht. Het script is nu meer flexibel en dit zal er voor zorgen dat men dit in de toekomst nog kan uitbreiden.

## 3.2 Procedure upgrade mariadb

Tegen 2024 moeten alle dbserver binnen Defensie draaien op Mariadb 10.6 op RHEL 8. Om deze transitie rustig te laten verlopen is er nood aan een procedure die men kan volgen tijdens zo een upgrade van een machine.

### 3.2.1 Beginsituatie

De dbserver bij Defensie draaien op mariadb 5.5 op RHEL 7.

### 3.2.2 Doel

Schrijf een procedure die men kan volgen om een machine te upgraden naar Mariadb 10.6 op RHEL 8. De procedure moet aandacht schenken aan volgende punten:

- het besturingssysteem van de nieuwe machine is Red Hat Enterprise Linux (RHEL) 8
- er gaat geen data verloren
- de juiste mensen inlichten dat er onderhoud zal doorgevoerd worden op hun dbserver

### 3.2.3 Plan van Aanpak

Voor deze opdracht zal ik iteratief werken:

- de eerste iteratie zal ik op papier (in grote lijnen) schetsen wat er moet gebeuren
- de tweede iteratie zal ik dit in een word-document gieten en laten nalezen door Tom
- de derde en volgende iteratie(s) zal ik de procedure meer specifiek maken

- eens de procedure duidelijk is, ga ik te werk om bepaalde hulpscripts te schrijven. Deze moeten er voor zorgen dat de persoon die de transitie doet zo weinig mogelijk manueel werk heeft.
- schrijf het script om users te checken
- als laatste zal ik de procedure grondig testen.

#### **3.2.4 Uitwerking**

Nadat de procedure (zie Bijlage C) opgesteld was en nagekeken door Tom ben ik begonnen aan het script `check_users` (zie Bijlage F).

#### **3.2.5 Eindresultaat**

#### **3.2.6 Business doelstellingen**

#### **3.2.7 Persoonlijke doelstellingen**

### **3.3 High availability bij databanken**

Onderzoek hoe men binnen defensie high availability implementeren? Als er een db faalt en niet beschikbaar is moet de applicatie zonder problemen verderwerken. Volgende punten moeten zeker vermeld zijn:

- is er een licentie voor nodig?
- is de software ondersteund door Red Hat?
- zijn er open source beperkingen?
  - enkel voor commercieel gebruik?
  - beperking voor militaire doeleinden?

#### **3.3.1 Beginsituatie**

Binnen Defensie is er momenteel maar 1 databank achter een bepaalde applicatie. Dit zorgt er natuurlijk voor dat als die databank faalt, de applicatie ook niet meer zal werken.

#### **3.3.2 Doel**

Zorg ervoor dat er meerdere databanken met dezelfde data achter een bepaalde applicatie zitten.

#### **3.3.3 Plan van Aanpak**

Voor deze opdracht zal ik iteratief werken:

- de eerste iteratie zal ik op papier (in grote lijnen) schetsen wat er moet gebeuren alsook ideeën opschrijven van productie die men mogelijk kan gebruiken
- de tweede iteratie zal ik dit in een word-document gieten en laten nalezen door Tom



- ik ga een master-slave opstelling testen
- ik ga proberen om de master-slave opstelling om te vormen naar een master-master

Nadat het document volledig is ga ik een handleiding maken om dit te implementeren en een eenvoudige opstelling testen met drie servers (twee db servers en 1 een proxyserver).

#### **3.3.4 Uitwerking**

Ik heb een installatie handleiding gemaakt (Zie bijlage C)

#### **3.3.5 Eindresultaat**

#### **3.3.6 Business doelstellingen**

#### **3.3.7 Persoonlijke doelstellingen**

### **3.4 MS SQL logins**

#### **3.4.1 Beginsituatie**

De Jboss Application servers connecteren zich met MSSQL server met SQL logins en dus via SQL server authentication.

#### **3.4.2 Doel**

Om het systeem veiliger te maken wil de dienst MS SQL overschakelen naar Windows authenticatie bij het inloggen op een databank.

#### **3.4.3 Plan van Aanpak**

Ik ga beginnen met mezelf bekend te maken met het onderwerp. Ik ga een virtuele machine vragen waar ik op kan testen. Daarna zal ik een oplossing proberen zoeken op het probleem, en als laatste zal ik deze testen.

#### **3.4.4 Uitwerking**

Ik had een virtuele machine gevraagd die ik kon gebruiken als testomgeving. Blijkbaar is het niet zo simpel om voor mij een virtuele machine te voorzien dus hebben ze mij toegang gegeven op een bestaande databank. Zo had ik een omgeving om dingen te testen. Ik heb een gesprek gehad met Donovan om een oplossing te zoeken op het probleem.

#### **3.4.5 Eindresultaat**

Het is mogelijk om om windows authenticatie te gaan gebruiken voor de JBOSS applicaties, maar het is niet onbelangrijk om aandacht te schenken aan het volgende. Het wachtwoord van de gebruiker die je in AD aanmaakt zal na een bepaalde tijd moeten veranderd worden (afgedwongen door de policy van AD), dit is een grote administratieve taak als je dit moet doen voor alle servers. Om aan Windows authenticatie te doen kam men gebruik maken van dezelfde driver dat men nu ook gebruikt.

#### **3.4.6 Business doelstellingen**

Het systeem zal veiliger zijn met Windows Authenticatie omdat er dan gebruikers gebruikt worden die gedefinieerd zijn in AD.

### **4 Eindreflectie**

- 4.1 Voldoet de stage aan hetgeen je verwacht had?**
- 4.2 Overwelke opdracht ben je het meest trots?**
- 4.3 Is dit het toekomstbeeld dat je voor ogen hebt?**
- 4.4 Vind je van jezelf dat je genoeg ervaring hebt opgedaan om nu in het werkveld te stappen?**

## Verklarende woordenlijst

**ACL** Access control list. 11

**AD** Active Directory. 9

**CC V&C** Competentie centrum Vliegend materieel & Communicatie- en informatiesystemen (Defensie, 2022a). 3, 5

**chef Defensie** De chef Defensie, ook CHOD (Chief of Defence) of stafchef genoemd, is het hoofd van de Belgische Defensie en van de vier componenten. Hij is de hoogste bevelhebber (na de koning) en hoeft alleen aan de bevoegde voogdijminister verantwoording af te leggen. (Wikipedia, 2021a). 2, 3

**db** Database. 8, 12

**eenheid** Een militaire eenheid bestaat uit een commando- en staf-element, uit uitvoerende militaire eenheden van een lager niveau en steuneenheden. Een militaire eenheid staat onder commando van een (onder)officier van voldoende rang (Wikipedia, 2021c). 3, 5

**IT&S** Information Technology & Services (Defensie, 2022a). 3, 5

**kwartier** Tijdelijke verblijfplaats voor militairen (Wikipedia, 2021b). 3

**LVM** Linux Volume Manager. 11, 14

**MS SQL** Microsoft SQL Server is een relationeel databasebeheersysteem ontwikkeld door Microsoft. 9, 14–17

**RDP** Remote desktop protocol. 11

**SELinux** Security Enhanced Linux. 11

**voogdijminister** In België is een voogdijminister een minister die verantwoordelijk is voor een bepaalde instelling, organisatie of dienst. 31

## Referenties

- Defensie. (2022a, februari 10). *Onthaalbrochure CCVE&C*.
- Defensie. (2022b). *Onze organisatie*. Verkregen 21 maart 2022, van <https://www.mil.be/nl/over-defensie/#onze-organisatie>
- Nato. (2020, juli 15). *Admiral Michel Hofman*. Verkregen 20 april 2022, van [https://www.nato.int/cps/en/natohq/who\\_is\\_who\\_161196.htm](https://www.nato.int/cps/en/natohq/who_is_who_161196.htm)
- Wikipedia. (2021a, augustus 24). *Chef Defensie (België)*. [https://nl.wikipedia.org/wiki/Chef\\_Defensie\\_\(Belgi%C3%AB\)](https://nl.wikipedia.org/wiki/Chef_Defensie_(Belgi%C3%AB))
- Wikipedia. (2021b, februari 25). *Kwartier (militair)*. Verkregen 21 maart 2022, van [https://nl.wikipedia.org/wiki/Kwartier\\_\(militair\)](https://nl.wikipedia.org/wiki/Kwartier_(militair))
- Wikipedia. (2021c, januari 19). *Militaire eenheid*. Verkregen 21 maart 2022, van [https://nl.wikipedia.org/wiki/Militaire\\_eenheid](https://nl.wikipedia.org/wiki/Militaire_eenheid)
- Wikipedia. (2022a, maart 11). *Defensie van België*. Verkregen 20 april 2022, van [https://nl.wikipedia.org/wiki/Defensie\\_van\\_Belgi%C3%AB](https://nl.wikipedia.org/wiki/Defensie_van_Belgi%C3%AB)
- Wikipedia. (2022b, maart 22). *Filip van België*. Verkregen 20 april 2022, van [https://nl.wikipedia.org/wiki/Filip\\_van\\_Belgi%C3%AB](https://nl.wikipedia.org/wiki/Filip_van_Belgi%C3%AB)
- Wikipedia. (2022c, januari 14). *Ludivine Dedonder*. Verkregen 20 april 2022, van [https://nl.wikipedia.org/wiki/Ludivine\\_Dedonder](https://nl.wikipedia.org/wiki/Ludivine_Dedonder)

## A Stagedagboek

### A.1 Week 1

#### A.1.1 Maandag 21/02/2022

*Eerste stagedag.* Kennismaking met **Tom De Leeuw**, Tom staat in voor het linuxgedeelte. Laptop ontvangen, vooral gepraat over mogelijkheden van de stage.

Werkwijze voor telewerken:

1. Surf naar <https://portal.connect.mil.be> en meld aan met **itsme**
2. Kies voor vpn. Eens de verbinding opstaat kan je beginnen met skype.

#### A.1.2 Dinsdag 22/02/2022

werkuren: 8:00 - 16:00

Zelfstandig research gedaan naar LVM. [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/8/html/configuring\\_and\\_managing\\_logical\\_volumes/index](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/configuring_and_managing_logical_volumes/index)

Na de research enkele kleine opdrachtjes uitgevoerd op mijn eigen devserver binnen Defensie:

1. aanmaken logical volume
  - maak logical volume aan van 300mb
  - mount logical volume op `/var/data01`
  - herstart server en kijk of het nog bestaat.
2. aanpassen logical volume
  - vergroot logical volume
  - verklein logical volume
3. volume groups
  - maak 2 volume groups van 100-200mb

Achteraf als afsluiter van de dag moest ik een grafische omgeving installeren op mijn devserver en proberen om via windows RDP de server te kunnen overnemen.

#### A.1.3 Woensdag 23/02/2022

werkuren: 8:00 - 16:00

Zelfstandig research gedaan naar Redhat Satellite, Foreman en Katello. Ook naar filesecurity: ACL's, SELinux.

Verworven kennis toegepast in demo-omgeving. Samen met Tom de Satellite omgeving ontdekt. Uitleg gekregen hoe alles werkt.

#### **A.1.4 Donderdag 24/02/2022**

werkuren: 8:00 - 16:00

Kennis opgefrist ivm Mariadb, mysql en appstreams in RHEL. Manueel mariadb geïnstalleerd op devmachine, db en users aangemaakt. Research over puppet en aan de hand van puppet mariadb proberen installeren en users en db's aan te maken. Meeting bijgewoond met externe mensen.

### **A.2 Week 2**

#### **A.2.1 Maandag 28/02/2022**

werkuren: 08:00-16:00

Afgewerkt waar ik vorige donderdag mee bezig was. Bezig aan het moderniseren van het script `bu_script.sh`.

#### **A.2.2 Dinsdag 01/03/2022**

werkuren: 8:00 - 16:00

`bu_script.sh` afgewerkt. Eenvoudige zelfgeschreven versie. Begin aan voorbereiding theorie voor donderdag.

#### **A.2.3 Woensdag 02/03/2022**

werkuren: 8:00 - 16:00

Vorbereiding voor Donderdag

- research gedaan in verband met Zabbix (monitoring tool)
- git kennis opgefrist
- toegang gekregen tot de gitlab servers van Defensie.
- hello world project geschreven in Puppet op mijn dev machine

#### **A.2.4 Donderdag 03/03/2022**

werkuren: 8:00 - 16:00

Introductie tot Zabbix en puppet door Donovan. Daarna zelf via puppet zabbix proberen installeren op devmachine.

### **A.3 Week 3**

#### **A.3.1 Maandag 07/03/2022**

werkuren: 08:00-16:00

- denk oefening:
  - hoe kan je het `bu_script.sh` uitbreiden zodat er een melding komt in Zabbix wanneer een backup faalt?
  - zoek uit wat er misgegaan is met de devmachine (te weinig geheugen).

### **A.3.2 Dinsdag 08/03/2022**

werkuren: 8:00 - 16:00

Introductie in Jboss applicatie servers door Christophe. Christophe heeft me alles goed uitgelegd en we hebben een testapplicatie gedeployed op mijn devmachine. Ik heb (onder begeleiding van Christophe) een applicatie mogen vernieuwen in de acceptatieomgeving. (netmanto)

### **A.3.3 Woensdag 09/03/2022**

werkuren: 8:00 - 16:00

De introductie in Jboss Applicatie servers verder gezet en afgewerkt. We hebben nu niet alles maar redelijk veel overlopen van jboss servers. Ook is het nu duidelijk hoe deze gemonitord worden (Jboss Operations Networks).

Een eerste stagegesprek.

### **A.3.4 Donderdag 10/03/2022**

Werkuren: 08:30 - 16:30

Werkdag in Peutie.

De opdracht gekregen om een oplossing te zoeken voor volgende problemen:

- hoe high availability implementeren voor een db die aan een applicatie hangt?
- maak een procedure om Mariadb 5.5 op RHEL 7.x to upgraden naar Mariadb 10.6 op RHEL 8 zonder verlies van gegevens.

Ook de toegangsbadge in orde gemaakt samen met Donovan.

## **A.4 Week 4**

### **A.4.1 Maandag 14/03/2022**

werkuren: 08:00-16:00

Verder gewerkt aan de denkoefeningen op een iteratieve manier. Alles mooi in een word-documentje gegoten.

### **A.4.2 Dinsdag 15/03/2022**

werkuren: 8:00 - 16:00

Verder gewerkt aan de denkoefeningen op een iteratieve manier. Er is nood aan validatie van gebruikers en databanken. Hiervoor een concept bedacht.

### **A.4.3 Woensdag 16/03/2022**

werkuren: 8:00 - 16:00

Verder gewerkt aan de denkoefeningen.

Upgrade mariadb:

- validatie voor users bedacht. gestart aan een script om dit uit te voeren maar ik zat vast gaandeweg dus nu ga ik eens kijken dat er geen andere manier is om dit te doen.

db high availability:

- iteratie afgewerkt. Nu wachten op feedback.

Stageverslag: structuur opgemaakt.

#### **A.4.4 Donderdag 17/03/2022**

Werkuren: 08:15 - 16:15

Fysiek in Peutie.

Normaal was er een introductie van MS SQL gepland vandaag maar deze is verzet naar een later moment. Ter vervanging heb ik verschillende praktische oefeningen gedaan ivm LVM op mijn devmachine. Ook dit allemaal besproken met Tom.

### **A.5 Week 5**

#### **A.5.1 Maandag 21/03/2022**

werkuren: 08:00-16:00

Verder gewerkt aan de denkoefening om high availability te implementeren bij databanken. Tom heeft voor mij een aantal virtuele machines laten maken zodat ik hierop kan testen.

Stageverslag verder aangevuld.

#### **A.5.2 Dinsdag 22/03/2022**

Geen stage.

Jobevent van HoGent

#### **A.5.3 Woensdag 23/03/2022**

werkuren: 8:00 - 16:00

Verder gewerkt aan de denkoefening om high availability te implementeren bij databanken: Testopstelling gemaakt.

- databank master
- databank slave
- reverse proxy

Zie Figuur 7, Dit is het netwerkdiagram van de opstelling.

Er is nog uitbreiding mogelijk: Hoe van replica een master maken?

#### **A.5.4 Donderdag 24/03/2022**

Fysiek in Peutie

werkuren: 8:00 - 16:00

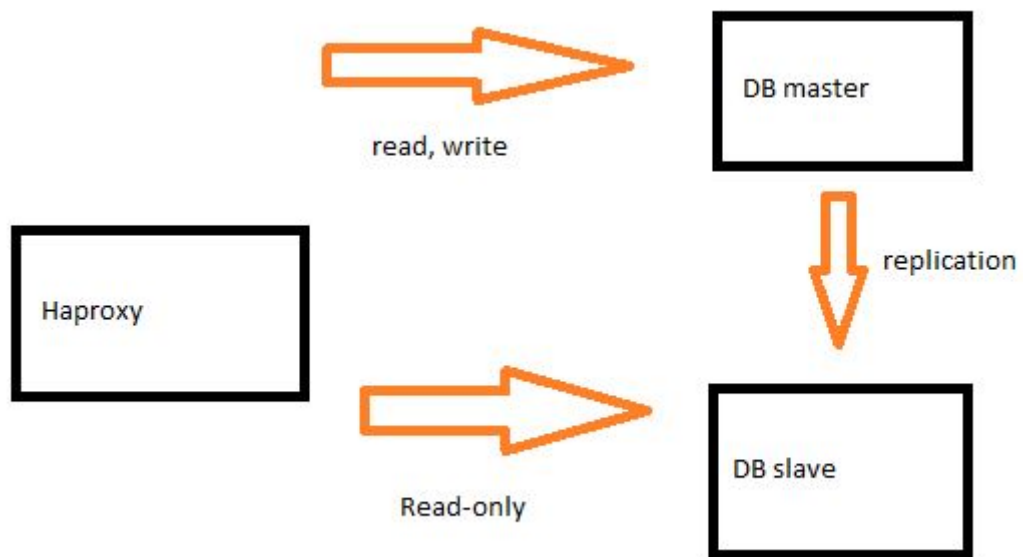
Introductie van *MS SQL*

Onderzoekopdracht besproken. ik zal hierover nog een document ontvangen.

Tom heeft mij een rondleiding gegeven in de serverroom van in Peutie.

In de namiddag was er een drink binnen defensie waar de kolonel een toespraak heeft gehouden. Het was tof om deze bij te wonen.





Figuur 7: netwerk diagram

## A.6 Week 6

### A.6.1 Maandag 28/03/2022

werkuren: 08:00-16:00

Gewerkt aan de opdracht voor MS SQL. Vooral research gedaan en mezelf bekend gemaakt met het onderwerp. Ook een virtuele machine gevraagd om dingen te testen.

### A.6.2 Dinsdag 29/03/2022

werkuren: 08:00-16:00

Bijlagen toegevoegd aan stageverslag. Verder onderzoek gedaan voor MS SQL. Gewacht op de virtuele machine.

Ad hoc een taak gekregen van Tom: schrijf een patch voor een databank. zie bijlage D.1

### A.6.3 Woensdag 30/03/2022

werkuren: 08:00-16:00

Blijkbaar is het niet simpel om voor mij een virtuele machine te voorzien dus we hebben het opgelost door een named instance te maken op een bestaande server. Hierdoor heb ik enkele dingen kunnen testen in MS SQL.

### A.6.4 Donderdag 31/03/2022

werkuren: 08:00-16:00

Telewerken.

Verder gewerkt aan de opdracht van MS SQL. Begonnen aan een script te schrijven voor de conversie van de logins/users te doen. Dit is moeilijker dan ik gedacht had.

## **A.7 Week 7**

### **A.7.1 Maandag 04/04/2022**

werkuren: *08:00-16:00*

Verder gewerkt aan de opdracht van MS SQL, niet gemakkelijk. Men kan zeggen dat ik vast zit. Johan was niet online dus ik kon hem niet bereiken om hulp te vragen.

### **A.7.2 Dinsdag 05/04/2022**

werkuren: *08:00-16:00*

Verder gewerkt aan de opdracht. Zit nog steeds vast, Johan niet online.

### **A.7.3 Woensdag 06/04/2022**

werkuren: *08:00-16:00*

Hulp gevraagd aan Johan maar nog geen antwoord gekregen. Voor de rest verder gewerkt aan de opdracht van MS SQL.

### **A.7.4 Donderdag 07/04/2022**

werkuren: *08:00-16:00*

Telewerken.

Verder gewerkt aan MS SQL.

## **A.8 Week 8**

### **A.8.1 Maandag 11/04/2022**

Afwezig

### **A.8.2 Dinsdag 12/04/2022**

Afwezig

### **A.8.3 Woensdag 13/04/2022**

Afwezig

### **A.8.4 Donderdag 14/04/2022**

Afwezig

## **A.9 Week 9**

### **A.9.1 Maandag 19/04/2022**

geen stage. (paasmaandag)

### **A.9.2 Dinsdag 20/04/2022**

werkuren: 08:00-16:00

Verder gewerkt aan de opdracht van MS SQL. Ik zit hierin vast en ik heb nog geen contact kunnen maken met Johan. Tom had vandaag geen tijd om mij te helpen maar zal morgen contact opnemen met mij.

### **A.9.3 Woensdag 21/04/2022**

werkuren: 10:00-17:00

Stand van zaken voor de opdracht van MS SQL besproken met Tom. Johan heeft deze week nog verlof en zal vanaf volgende week terug aanwezig zijn. Contact gehad met Steven om een moment in te plannen om de mogelijkheden na de stage te bespreken.

### **A.9.4 Donderdag 22/04/2022**

werkuren: 08:00-16:00

Begonnen aan het script om de users te checken voor de opdracht mysql migratie.

## **A.10 Week 10**

### **A.10.1 Maandag 25/04/2022**

werkuren: 08:15-16:15

Contact gehad met Johan en Donovan over de opdracht van MS SQL. Meer info gezocht over de jtcds driver. ook naar de jdbc driver gekeken.

### **A.10.2 Dinsdag 26/04/2022**

werkuren: 08:15-16:15

Verder gewerkt aan de oefening van MS SQL: meer info gezocht over de jtcds en jdbc drivers. niet gevonden wat ik wou vinden.

Om mijn gedachten eens te verzetten heb ik nog eens gekeken naar het script 'check\_users.py' en heb ik de output een aangepast zodat er nu een waarschuwing staat en verder gescriven aan het stageverslag.

Voor de opdracht High availability voor mariadb: Research gedaan over hoe we in plaats van master-slave naar master-master kunnen gaan.

### **A.10.3 Woensdag 27/04/2022**

werkuren: 10:00-17:30

Verder gezocht naar een oplossing voor de jtcds driver. Steeds niet gevonden, morgen ga ik hierover met Donovan proberen praten om samen een oplossing te zoeken. Nagedacht over hoe ik een testopstelling ga opzetten voor een master-master opstelling bij de opdracht High availability voor mariadb. Stageverslag verder aangevuld.

#### **A.10.4 Donderdag 28/04/2022**

werkuren: *8:15-16:15*

Fysiek in Peutie, JTDS driver:

- Geen onderzoekwerk mogelijk want binnen het netwerk van Defensie is het internet nog niet 100

Opdracht High Availability voor mariadb:

- Netwerkdigram aangepast zodat dit een master-master oplossing is. Nog niet getest.

Upgrade mariadb:

- Nagedacht hoe we een script kunnen schrijven die de check voor de databases doet.
- Check\_users. Python show grants uitvoeren voor elke user. <<https://codefather.tech/blog/shell-command-python/>>
- Check\_users:
  - verder gewerkt aan het script, logica geschreven om grants uit te lezen rechtstreeks van de db.
  - input veranderd dat het script de users ook rechtstreeks uit de db gaat halen.

### **A.11 Week 11**

#### **A.11.1 Maandag 02/05/2022**

werkuren: *8:30-16:30*

Een moment afgesproken met Donovan waar hij een scripting opdracht zal uitleggen en waar ik enkele vragen kan stellen voor een lopende opdracht (mysql) en waar ik enkele vragen kan stellen ivm met mijn bachelorproef. Dit zal waarschijnlijk donderdag plaatsvinden in Peutie.

Verder gewerkt aan het check\_users script.

#### **A.11.2 Dinsdag 03/05/2022**

werkuren: *8:30-16:30*

Frustrerende dag achter de rug. Verder gewerkt aan het check\_users script maar ik zit vast bij het vergelijken van de 2 puppet bestanden. Documentatie verder aangevuld.

#### **A.11.3 Woensdag 04/05/2022**

werkuren: *8:15-16:15*

Check\_users script even op kant gelegd. Beginnen nadenken hoe ik het check\_database script zal schrijven. Stageverslag nog extra aangevuld.

#### **A.11.4 Donderdag 05/05/2022**

werkuren: 8:00-16:00

Fysiek in Peutie.

Samengezeten met Donovan omdat ik enkele vragen had voor hem ivm met een aantal opdrachten en ook over mijn bachelorproef.

Samen met Donovan en Steven nog een gesprek gehad om te praten over de bachelorproef. Steven gaat mij in contact brengen met personen die bij Smals werken om mij een beetje uitleg te geven hoe te daar te werk gaan. Een moment afgesproken met Tom om mijn opdrachten eens te bekijken dit zal volgende donderdag plaatsvinden (12/05/2022)

check\_database:

- research gedaan naar welke commando's ik moet gebruiken om de nodige data te verkrijgen uit de db.
- code geschreven voor file structuur op te zetten.

mssql:

- alternatieve methodes gezocht.

#### **A.12 Week 12**

##### **A.12.1 Maandag 09/05/2022**

werkuren: 8:15-16:15

Aan stageverslag gewerkt. Verder opgezocht voor mssql

##### **A.12.2 Dinsdag 10/05/2022**

werkuren: 8:15-16:15

Enkele bugs uit de handleiding gehaald om replicatie te configureren. Status van de opdrachten toegevoegd in de readme

Aan het script `check_databases` gewerkt.

##### **A.12.3 Woensdag 11/05/2022**

Afwezig wegens een evenement van Telenet waar ik naartoe ben gegaan met toestemming van Tom.

##### **A.12.4 Donderdag 12/05/2022**

werkuren: 8:00-16:00

Fysiek in Peutie

Verder geschreven aan het `check_databases` script.

Samengezeten met Tom om mijn opdrachten eens te overlopen + enkele vragen gesteld over het `check_databases` script.

Een besluit beginnen schrijven voor de opdracht van mssql

### **A.13 Week 13**

**A.13.1 Maandag 16/05/2022**

**A.13.2 Dinsdag 17/05/2022**

**A.13.3 Woensdag 18/05/2022**

**A.13.4 Donderdag 19/05/2022**

### **A.14 Week 14**

**A.14.1 Maandag 23/05/2022**

**A.14.2 Dinsdag 24/05/2022**

**A.14.3 Woensdag 25/05/2022**

**A.14.4 Donderdag 26/05/2022**

## B Procedure: Upgrade Mariadb

This procedure can be used to upgrade Mariadb 5.5 on RHEL 7 to Mariadb 10.6 on RHEL 8.

1. Take vmware-snapshot of vm old db
2. Deploy new vm
  - OS: RHEL 8
  - Puppet Modules: `mysqldb`
3. Validate users (script)
  - Check if all existing users are defined in puppet
4. Validate databases (script)
  - Check if all existing databases are defined in puppet + check parameters
5. Create databases and users with grants via puppet
6. Plan a moment for the upgrade
7. Notify the right people "DB will be under maintenance"
8. Freeze the database
  - Stop the application
  - Edit firewall so no external connection can be made
9. Export data from old vm
10. Import the data into new vm
11. Notify the right people "db is ready to be used."
12. If go (upgrade works)
  - delete snapshot old vm
  - delete old vm
13. If no go (upgrade failed)
  - restore snapshot old vm
  - delete snapshot old vm
  - delete new vm

## C MariaDB: Replication installation guide

### C.1 Steps

#### C.1.1 Database replication

1. install MariaDB
2. start and config MariaDB on primary server
3. config master firewall
4. start and config MariaDB on replica server
5. test MariaDB

#### C.1.2 Reverse proxy

1. install proxy
2. config proxy
3. test proxy

### C.2 Install MariaDB

The repository is already there so we just have to install some packages.

```
1 sudo dnf install MariaDB-server MariaDB-backup
```

### C.3 Start/config primary server

Edit config file with following:

```
1 [mariadb]
2 log_bin
3 server_id=1
4 log-basename=master
5 binlog-format=mixed
```

*server\_id* is a unique value.

Start the dbserver:

If already running, restart dbserver to apply config.

```
1 systemctl enable mariadb --now
```

There should be at least 1 user created for replication:

Create user accounts:

Each account should be created on the master so replication can make these accounts on the slaves.



### C.3.1 Replication user

```
1 CREATE USER 'repl'@'slave_ip' IDENTIFIED BY 'test';
```

Grant required privileges.

```
1 GRANT REPLICATION SLAVE
2 ON *.* TO repl@'slave_ip';
```

## C.4 Start/config replica server

```
1 [mariadb]
2 log_bin
3 server_id=2
4 log-basename=master
5 binlog-format=mixed
```

*server\_id* is a unique value.

```
1 SHOW SLAVE STATUS;
2 +-----+-----+-----+-----+
3 | File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
4 +-----+-----+-----+-----+
5 | master1-bin.000096 |      568 |              |                  |
6 +-----+-----+-----+-----+
```

Take notes of the filename and position

```
1 CHANGE MASTER TO
2 MASTER_HOST='master_ip',
3 MASTER_USER='replication_user',
4 MASTER_PASSWORD='test',
5 MASTER_PORT=3306,
6 MASTER_LOG_FILE='master-bin.000096',
7 MASTER_LOG_POS=568,
8 MASTER_CONNECT_RETRY=10;
```

With fresh master, you don't need to specify 'MASTER\_LOG\_FILE' and 'MASTER\_LOG\_POS'

### C.4.1 Start replication

```
1 START SLAVE;
2 SHOW SLAVE STATUS;
```

If replication is running properly, both 'Slave\_IO\_Running' and 'Slave\_SQL\_Running' should be 'Yes'.

## C.5 Test replication

Use the following url to test replication: <https://mariadb.com/docs/deploy/topologies/primary-replica/enterprise-server-10-3/test-es/>

## C.6 Install proxy

I chose to go with 'haproxy'.  
To install this proxy just use dnf.

```
1 sudo dnf install haproxy
```

## C.7 Config proxy

### C.7.1 Edit configuration file

```
1 # /etc/haproxy/haproxy.cfg
2
3 defaults
4 mode tcp
5 option mysql-check user haproxy_health
6 frontend frontend
7 # read-requests will arrive at port 3100
8 bind *:3100
9 # write-requests will arrive at port 3200
10 bind *:3200
11 # if write send to masters
12 use_backend db_masters if { dst_port 3200 }
13 # if read sent to slaves
14 default_backend db_slaves
15
16 backend db_masters
17 server master1 10.8.131.120:3306
18
19 backend db_slaves
20 server slave1 10.8.131.121:3306
21 server master1 10.8.131.120:3306
```

### C.7.2 (re)start service

```
1 sudo systemctl enable haproxy --now
```

### C.7.3 SELinux

```
1 setsebool haproxy_connect_any 1
```

## C.8 Test proxy

To test proxy, use port 3200 to write data and port 3100 to query data.

## D Kleine opdrachtjes

Tijdens mijn stage heb ik een aantal kleine opdrachtjes gekregen die niet echt passen tussen de grote opdrachten in sectie 3. Daarom ga ik deze hier beschrijven.

### D.1 Schrijf een patch voor een databank

In het tekstbestand *output\_old.txt* en *output\_new.txt* staan namen van tabellen in een databank. (hieronder is dit een dataset om te testen)

Het is de bedoeling om een bestand (*patch\_command.txt*) te genereren met sql statements die alle tabellen die enkel in het bestand *output\_old.txt* te vinden zijn te verwijderen.

De inhoud van het tekstbestand *output\_old.txt* (testdata) is:

```
1 test
2 een
3 twee
```

De inhoud van het tekstbestand *output\_new.txt* (testdata) is:

```
1 ja
2 deutz
3 oke
4 twee
```

De inhoud van het bestand dat we gegenereerd hebben (*patch\_command.txt*) is:

```
1
2 drop table test;
3 drop table een;
```

#### D.1.1 Script

```
1 #!/bin/bash
2
3 # Variables
4
5 old_tables="$(cat output_old.txt)"
6 new_tables="$(cat output_new.txt)"
7 drop=-1
8 output="patch_command.txt"
9
10
11 # main
12
13 # create outputfile
14 if [ ! -e $output ]
15 then
16     touch $output
17 fi
18
19 # empty outputfile
20 echo "" > $output
21
22
23 # iterate tables and generate sql
24 for old_table in $old_tables
```

```
25 do
26     drop=1
27     for new_table in $new_tables
28     do
29         if [ "$old_table" = "$new_table" ]
30         then
31             drop=0
32         fi
33     done
34     if [ $drop = 1 ]
35     then
36         echo "drop table $old_table;" >> $output
37     fi
38 done
```

## E Script: bu\_script

```
1  #!/bin/bash
2
3
4  set -o errexit # abort on nonzero exitstatus
5  set -o nounset # abort on unbound variable
6  set -o pipefail # 'dont hide errors within pipes
7
8  #-----
9  # Variables
10 #-----
11
12 # TESTING PURPOSE VARIABLES
13 # remove after development
14 CONFIG_bu_root='/home/pieter/bu_dir'
15 CONFIG_file_cnf="$CONFIG_bu_root/files/cnf_files"
16 CONFIG_file_db_exclude="$CONFIG_bu_root/files/exclude_db"
17 CONFIG_file_bu_log="$CONFIG_bu_root/files/bu.log"
18 CONFIG_file_bu_report="$CONFIG_bu_root/files/bu_report.log"
19 CONFIG_backup_dir="$CONFIG_bu_root/data"
20 CONFIG_mail_address="pieter.vankeer@mil.be"
21 CONFIG_backup_dest="/home/pieter/backup"
22
23 # mysql backup user.
24 CONFIG_mysql_dump_username='backup'
25 CONFIG_mysql_dump_password='test'
26 CONFIG_mysql_dump_host='localhost'
27
28 CONFIG_mysql_standard_socket='/var/lib/mysql/mysql.sock'
29
30 errorflag=0
31 #-----
32 # Functions
33 #-----
34
35 # fix_file_format: Make sure file format is LF
36 fix_file_format(){
37     dos2unix "$CONFIG_file_cnf"
38     dos2unix "$CONFIG_file_db_exclude"
39 }
40
41 # prepreplog: prepare the report file. (empty file, create header...)
42 # @args: $CONFIG_file_bu_report
43 prepreplog(){
44
45     # make sure file is empty
46     cat /dev/null > "$1"
47
48     # prepare report file with header
49     {
50         echo "<-----Backup server $(hostname) at $(date +%Y-%m-%d)----->"
51         echo " "
52         echo "Backup start at $2 "
53     } >> "$1"
54 }
55
56 # initiate_server_backup: start the process to backup the db server.
57 initiate_server_backup(){
58     # test connection
59     RC=$(test_server $CONFIG_mysql_standard_socket)
```

```

60 RC=$?
61
62 # test backup destination
63 test_backup_dest "$CONFIG_backup_dest"
64
65 # if connection with server was successful (see test_server())
66 # then: green light for backup
67 # else: do not perform anything.
68 if [ $RC -eq 0 ];
69 then
70     # get all the db's on the server
71     ALLDB=$(mysql --user="${CONFIG_mysql_dump_username}" --password
        ="${CONFIG_mysql_dump_password}" --host="${CONFIG_mysql_dump_host}"
        --socket="${CONFIG_mysql_standard_socket}" -B -e "show databases" |
        grep -v "Database")
72
73     EXCLUDES=$(cat $CONFIG_file_db_exclude)
74
75     # for each dbname in ALLDB: check if not excluded and backup
76     # BU=1 means db is not excluded so perform backup.
77     # BU=0 means db is excluded so do not backup db
78     for dbname in $ALLDB
79     do
80         # check if dbname needs to be excluded
81         if is_excluded $dbname -eq 0
82         then
83             DUMP=$CONFIG_backup_dir/'/'$(date +%Y%m%d)'_'$dbname'.
            sql'
84             backup_db $CONFIG_mysql_standard_socket $dbname "$DUMP"
            > /dev/null
85             fi
86         done
87     fi
88 }
89
90 # test_server: check if connection can be established
91 # @args: socket to connect to the server.
92 # @return: exit code of the connection to the server
93 test_server(){
94     # test connection and write errors to report
95     mysqlshow --user="${CONFIG_mysql_dump_username}" --password="${
        CONFIG_mysql_dump_password}" --host="${CONFIG_mysql_dump_host}" --
        socket="$1" 2>> $CONFIG_file_bu_report
96     # was connection successful?
97     RC=$?
98     if [ $RC -ne 0 ]; then
99         errorflag=1
100     fi
101     return $RC
102 }
103
104 # test_backup_dest: check if backup destination is ready to receive the
        files.
105 # @args: destination folder to write backup
106 test_backup_dest(){
107     # Normally you would like to write the backup to a remote server/
        destination but for testing purposes I keep the backup locally.
108
109     if [ ! -e "$1" ]
110     then
111         mkdir -p "$1"
112     fi

```

```

113 }
114
115 # is_excluded: check is dbname is excluded from backup
116 # @args: dbname
117 # @return: 1 if excluded, 0 if not excluded
118 is_excluded(){
119     excluded=0
120     for exclude in $EXCLUDES
121     do
122         if [ "$1" = "$(echo "$exclude" | cut -f3 -d'/')" ]
123         then
124             # db needs te be excluded
125             excluded=1
126         fi
127     done
128     return $excluded
129 }
130
131 # backup_db: make backup of specific db
132 # @args: mysql socket, dbname, place to dump files
133 backup_db(){
134     # record time when db backup started
135     BUstart=$(date +%s)
136
137     # write in report
138     {
139         echo " "
140         echo "$dbname"
141     } >> $CONFIG_file_bu_report
142
143
144     mysqldump --user="${CONFIG_mysql_dump_username}" --password="${CONFIG_mysql_dump_password}" --host="${CONFIG_mysql_dump_host}" --socket="$1" --routines --databases "$2" 2>>$CONFIG_file_bu_report |
145     gzip > "$3.gz" 2>> $CONFIG_file_bu_report
146     RCbu=$?
147
148     # if db backup failed
149     # then: remove files.
150     # else: move files to backup destination
151     if [ $RCbu -ne 0 ]; then
152         rm -f $CONFIG_backup_dir/"*" 2>> $CONFIG_file_bu_report
153         errorflag=1
154     else
155         move_file "$DUMP.gz" $CONFIG_backup_dest > /dev/null
156         RCmove=$?
157
158         # if movefile was successfull
159         if [ $RCmove -eq 0 ]; then
160             drop_old_backup "$CONFIG_backup_dir/$(date +%Y%m%d)_$dbname.sql.gz"
161             RCdrop=$?
162             if [ $RCdrop -ne 0 ]; then
163                 errorflag=1
164             fi
165         else
166             errorflag=1
167         fi
168
169         # record time when db backup ended
170         BUend=$(date +%s)
171         BUdiff=$((BUend - BUstart))
172         {

```

```

171         echo "The backup for $dbname was done in = $(echo - | awk -
           v "S=$BUdiff" '{printf "%dh:%dm:%ds",S/(60*60),S%(60*60)/60,S%60}')"
           "
172         echo " "
173     } >> $CONFIG_file_bu_report
174
175 fi
176 }
177
178 # Move file to backup destination server
179 # @args: file to copy, destination for file
180 # @return: 0 if value of $RC1 and $RC2 both equal 0, else 1
181 move_file(){
182     cp -r "$1" "$2" 2>> $CONFIG_file_bu_report
183     RC1=$?
184     rm -f "$1" 2>> $CONFIG_file_bu_report
185     RC2=$?
186     if [ $RC1 -eq 0 ] && [ $RC2 -eq 0 ]; then
187         return 0
188     else
189         return 1
190     fi
191 }
192
193 # drop old backup in backup destination
194 # @args: filename of the new backup
195 # @return: 0 if remove operation was successful, >0 is not succesfull
196 drop_old_backup(){
197     rm -f "$1" 2>> $CONFIG_file_bu_report
198     RC=$?
199     return $RC
200 }
201
202 # process_runtime: calculate runtime and give feedback in report.
203 # @args: ENDSEC, STARTSEC
204 process_runtime() {
205     # Calculate runtime.
206     DIFF=$((ENDSEC - STARTSEC))
207
208     # Write feedback in report.
209     echo " " >> "$CONFIG_file_bu_report"
210     echo "Backup runtime = $(echo - | awk -v "S=$DIFF" '{printf "%dh:%
           dm:%ds",S/(60*60),S%(60*60)/60,S%60}')" >> "$CONFIG_file_bu_report"
211 }
212
213 # distribute_backup_feedback: send feedback to the world!
214 # @args: errorflag (are there errors?)
215 distribute_backup_feedback(){
216     send_mail "$1"
217
218     # Area to expand the distribution of feedback to zabbix,...
219 }
220
221 # send mail
222 # @args: bu error True or false
223 #
224 send_mail(){
225     if [ "$1" -eq 0 ]; then
226         mail -s "Backup done on $(hostname)" $CONFIG_mail_address <
           $CONFIG_file_bu_report
227     else
228         mail -s "Backup error on $(hostname)" $CONFIG_mail_address <

```



```

229     $CONFIG_file_bu_report
230     fi
231 }
232 #-----
233 # Main
234 #-----
235
236 # make sure file is in right format (CRLF --> LF)
237 fix_file_format
238
239 # record time when backup starts
240 STARTSEC="$(date +%s)"
241
242 # prepare the report file
243 preprelog $CONFIG_file_bu_report "$(date '+%Y-%m-%d %T')"
244
245 # perform server backup
246 initiate_server_backup
247
248 # record time when backup ends
249 ENDSEC="$(date +%s)"
250
251 # calculate runtime
252 process_runtime "$ENDSEC" "$STARTSEC"
253
254 # append report in bu_log
255 cat $CONFIG_file_bu_report >> $CONFIG_file_bu_log
256
257 # distribute feedback
258 distribute_backup_feedback $errorflag

```

## F Script: check\_users

```
1 # check users: are all existing users defined in puppet?
2
3 # imports
4 import os
5 import pandas as pd
6 from datetime import date
7 import yaml
8 import re
9
10 # functions
11 def is_ignored_user(line):
12     # check if user needs to be ignored+
13     is_ignored = False
14     arr = line.split("|")
15     for user in ignore_users:
16         if arr[1] == user:
17             is_ignored = True
18     return is_ignored
19
20
21 def exec_query(query):
22     return os.system("mysql -e " + query)
23
24
25 def get_users():
26
27     # open files
28     tmp_file = open(
29         "tmp_users.txt",
30         "w",
31     )
32     test_input = open(
33         "user_input.txt",
34         "r",
35     )
36     input_lines = test_input.readlines()
37     # TODO uncomment test purpose
38     # input_lines = exec_query("select user, host from mysql.user;")
39     user_lines = []
40     # filter usefull lines and prepare for processing
41     for line in input_lines:
42         if line[:1] != "+":
43             line = line.strip("\n").strip("|").replace(" ", "") + "\n"
44             if not is_ignored_user(line):
45                 user_lines.append(line)
46
47     tmp_file.writelines(user_lines)
48     tmp_file.close()
49
50     data = pd.read_csv(
51         "tmp_users.txt",
52         sep="|",
53         header=0,
54     )
55
56     # cleanup
57     os.remove("tmp_users.txt")
58
59     return data
60
```

```

61
62 def generate_puppet(users, grants):
63     output_file = open(
64         "generated_puppet.txt",
65         "w",
66     )
67
68     for userrow in users.itertuples():
69         strpuppet = (
70             userrow[2]
71             + "\n ensure: present"
72             + "\n password: "
73             + userrow[3]
74             + "\n host: "
75             + userrow[1]
76             + "\n grants:\n"
77         )
78         for grantrow in grants.itertuples():
79             username = grantrow[1].strip("`")
80             if userrow[2] != username or userrow[1] != grantrow[2].
strip("`"):
81                 continue
82
83             # Als de privileges toe te passen zijn op alle db's dan
84             # hanteren we een algemene schrijfwijze
85             if grantrow[4] == "*.*":
86                 strpuppet = (
87                     strpuppet
88                     + " "
89                     + grantrow[1].strip("`")
90                     + ":"
91                     + "\n ensure: present"
92                     + "\n db: *"
93                     + "\n privileges:"
94                 )
95             else:
96                 strpuppet = (
97                     strpuppet
98                     + " "
99                     + grantrow[1].strip("`")
100                    + "@"
101                    + grantrow[4].split(".")[0]
102                    + ":"
103                    + "\n ensure: present"
104                    + "\n db: "
105                    + grantrow[4].split(".")[0]
106                    + "\n privileges:"
107                )
108             privileges = grantrow[3].split("|")
109             for privilege in privileges:
110                 strpuppet += "\n - " + privilege.strip()
111             strpuppet += "\n"
112             output_file.write(strpuppet)
113             output_file.close()
114
115 def get_grants(users):
116
117     # Testing purpose
118     grant_file = open("show_grants_example.txt", "r")
119     return process_grant_input(grant_file.readlines())
120     # for user in users.itertuples():

```

```

121     #     tmp_file = open(
122     #         "tmp_users.txt",
123     #         "w",
124     #     )
125     #     tmp_file = exec_query("show grants for " + user[2] + "@" +
126     #         user[1])
127     #     tmp_file.close()
128     #     return process_grant_input(tmp_file)
129
130 def process_grant_input(grant_input):
131     # TODO
132     tmp_file = open(
133         "tmp_grants.txt",
134         "w",
135     )
136     grant_lines = filter_grant_input(grant_input)
137
138     # process each line so that you get a useable panda.
139     # bron: https://www.geeksforgeeks.org/python-extract-string-between-two-substrings/
140     tmp_file.write("user,host,grants,priv_level,grant_option\n")
141     for line in grant_lines:
142         grant_opt = False
143         sub1 = "GRANT "
144         sub2 = " ON "
145         sub3 = " TO "
146         sub4 = " IDENTIFIED BY PASSWORD "
147         sub5 = " WITH "
148         idx1 = line.index(sub1)
149         idx2 = line.index(sub2)
150         idx3 = line.index(sub3)
151         idx4 = line.index(sub4)
152         grants = ""
153         priv_level = ""
154         user = ""
155         # for indexes between sub1 and sub2
156         for i in range(idx1 + len(sub1), idx2):
157             grants += line[i]
158         if "all" in grants:
159             grants = ["all"]
160         grants = grants.strip().replace(",", "|")
161
162         # for indexes between sub2 and sub3
163         for i in range(idx2 + len(sub2), idx3):
164             priv_level += line[i]
165         priv_level = priv_level.strip(" ")
166
167         if "WITH" not in line:
168             # for indexes between sub3 and end of line
169             for i in range(idx3 + len(sub3), idx4):
170                 user += line[i]
171         else:
172             grant_opt = True
173             # for indexes between sub3 and sub4
174             idx5 = line.index(sub5)
175             for i in range(idx3 + len(sub3), idx4):
176                 user += line[i]
177             user.replace("~", " ")
178
179         # write to tmp file
180         if grant_opt:

```

```

181         grant_opt = "Y"
182     else:
183         grant_opt = "N"
184     tmp_file.write(
185         user.split("@")[0]
186         + ","
187         + user.split("@")[1]
188         + ","
189         + grants
190         + ","
191         + priv_level
192         + ","
193         + grant_opt
194         + "\n"
195     )
196     tmp_file.close()
197
198     # create panda
199     data = pd.read_csv(
200         "tmp_grants.txt",
201         sep=",",
202         header=0,
203     )
204
205     # cleanup
206     os.remove("tmp_grants.txt")
207
208     return data
209
210
211 def filter_grant_input(grant_input):
212     grant_lines = []
213     for line in grant_input:
214         # line is not header line or spacer
215         if line[:1] != "+" and "Grants for" not in line:
216             grant_lines.append(line.strip("\n").strip("|") + "\n")
217     return grant_lines
218
219
220 def chars_to_str(list):
221     # convert unwanted special characted to string
222     for line in list:
223         y = re.findall("[*][^ ]+", line)
224         idx = list.index(line)
225         if len(y) > 0:
226             list[idx] = re.sub("[*][^ ]+", "'" + y[0] + "'", line)
227         elif " * " in line or " %" in line:
228             list[idx] = line.replace("*", "'*'\n").replace("%", "'%\n'")
229     )
230     return list
231
232
233 def compare_puppet():
234     # open files
235     with open("input_puppet.txt", "r") as input_puppet_file, open(
236         "generated_puppet.txt", "r"
237     ) as generated_puppet_file:
238         # generate tmp files
239         tmp_file1 = open("tmp1.txt", "w")
240         tmp_file2 = open("tmp2.txt", "w")
241
242         # process tmp files

```

```

242     input_puppet_lines = input_puppet_file.readlines()
243     generated_puppet_lines = generated_puppet_file.readlines()
244     input_puppet_lines = chars_to_str(input_puppet_lines)
245     generated_puppet_lines = chars_to_str(generated_puppet_lines)
246     tmp_file1.writelines(input_puppet_lines)
247     tmp_file2.writelines(generated_puppet_lines)
248     tmp_file1.close()
249     tmp_file2.close()
250
251     tmp_file1 = open("tmp1.txt", "r")
252     tmp_file2 = open("tmp2.txt", "r")
253
254     # read yaml
255     input_puppet = yaml.safe_load(tmp_file1)
256     generated_puppet = yaml.safe_load(tmp_file2, Loader=yaml.
BaseLoader)
257
258     # remove tmp files
259
260     print(generated_puppet)
261
262     flag = False
263
264     return flag
265
266
267 def generate_output(flag):
268     output_file = open("output.txt", "w")
269
270     output_file.write("Report generated on: " + str(date.today()))
271
272     output_file.close()
273
274
275 # main
276 ignore_users = ["root", "backup", "snow_dbdetect"]
277 users = get_users()
278 grants = get_grants(users)
279 generate_puppet(users, grants)
280 is_puppet_ok = compare_puppet()
281 generate_output(is_puppet_ok)

```