



Einddocument Project Hosting

Verslag

Project Hosting IT-Factory

Team CCS1 – Brent Druyts, Siebe Van
Rompay, Timo Van Litsenborg, Emil Dudayev,
Ilias Latifine (2 CCS 1 & 2 CCS 2)

Academiejaar 2021-2022

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

Inhoud

| | | |
|------------|---|-----------|
| 1 | INLEIDING | 5 |
| 2 | TECHNISCHE BESCHRIJVING | 6 |
| 2.1 | Schematische voorstelling | 6 |
| 2.2 | Werking onderdelen..... | 7 |
| 2.2.1 | Hoe kan de gebruiker ons bereiken? (DNS) | 7 |
| 2.2.2 | Bestanden uploaden (SFTP)..... | 7 |
| 2.2.3 | Websites hosten (webserver) | 8 |
| 2.2.4 | Userinterface | 8 |
| 2.2.5 | Data van de websites opslaan (databases) | 11 |
| 2.2.6 | Beveiliging | 12 |
| 2.2.7 | Beschikbaarheid..... | 12 |
| 2.2.8 | Configuratie synchronisatie | 12 |
| 2.2.9 | Monitoring..... | 12 |
| 2.2.10 | Back-ups..... | 13 |
| 3 | PROCEDURES EN HANDLEIDINGEN | 15 |
| 3.1 | Disaster recovery procedures | 15 |
| 3.1.1 | Zabbix monitoring server kapot | 15 |
| 3.1.2 | Haproxy DNS/load balancing container kapot | 15 |
| 3.1.3 | Een web container kapot..... | 15 |
| 3.1.4 | Meerdere web containers kapot | 15 |
| 3.1.5 | Een database kapot..... | 15 |
| 3.1.6 | Meerdere databases kapot | 15 |
| 3.1.7 | De volledige webserver kapot | 16 |
| 3.1.8 | SFTP-server kapot..... | 16 |
| 3.1.9 | Puppet master kapot | 16 |
| 3.1.10 | Volledig datacenter kapot..... | 16 |
| 3.2 | Handleiding voor IT-medewerkers..... | 17 |
| 3.2.1 | Docker structuur..... | 17 |
| 3.2.2 | DNS structuur..... | 20 |
| 3.2.3 | Hoe is onze domeinnaam geregistreerd? | 20 |
| 3.2.4 | SFTP | 20 |
| 3.2.5 | Puppetmaster..... | 22 |
| 3.2.6 | Script om de websites aan te maken: | 23 |
| 3.2.7 | Backup | 24 |
| 3.2.8 | Monitoring..... | 25 |
| 3.2.9 | OS Security..... | 25 |
| 3.2.10 | Password safe..... | 28 |
| 3.2.11 | Alle admin wachtwoorden veranderen in een uur | 28 |
| 3.2.12 | NTP-server en clients..... | 29 |
| 3.3 | Handleiding voor gebruikers | 29 |
| 3.3.1 | Hoe maak ik een account aan? | 30 |
| 3.3.2 | Hoe maak ik een domeinnaam aan? | 32 |
| 3.3.3 | Hoe maak ik een database aan?..... | 32 |
| 3.3.4 | Hoe kan ik files uploaden?..... | 32 |
| 3.3.5 | Hoe kan ik contact opnemen met support? | 33 |
| 4 | OPS REPORT CARDS | 34 |
| 4.1 | Gekozen Ops Report Cards..... | 34 |
| 4.2 | Geïmplementeerde Ops Report Cards..... | 36 |
| 4.3 | Documenten voor Ops Report Cards..... | 42 |
| 4.3.1 | De 3 empowering policies | 42 |
| 4.3.2 | OpsDoc..... | 43 |

| | | |
|------------|----------------------------------|-----------|
| 5 | OPTIONEEL | 53 |
| 5.1 | Andere documentatie | 53 |
| 5.1.1 | SLA | 53 |
| 6 | BESLUIT | 57 |

1 INLEIDING

In dit document staat ons eindverslag van ons project. Wat is ons project juist? Studenten van het tweede jaar APP/AI moeten een website maken en deze online beschikbaar stellen. Zo'n website online zetten noemen we hosting. Een hosting bij een privaat bedrijf zoals Combell is niet zo duur maar vraagt wel veel configuratiewerk.

Met ons project willen we een eenvoudige interface maken waarmee gebruikers met enkele klikken een website bij ons kunnen hosten.

In dit document hebben we ons project uitgelegd. We gaan eerst een technische beschrijving geven van ons project. Dan gaan we de verschillende onderdelen bespreken en uitleggen. Vervolgens gaan we de disaster recovery procedures uitleggen. Dit zijn de procedures die in werking treden wanneer er een ramp zich voordoet in het datacenter.

Daarna hebben we een handleiding geschreven waarin we aan onze mede IT'ers uitleggen wat we hebben gebouwd. Dit is een technische en complexe uitleg bedoeld voor mensen met een sterke IT kennis. Dan hebben we een eenvoudige handleiding geschreven voor onze gebruikers.

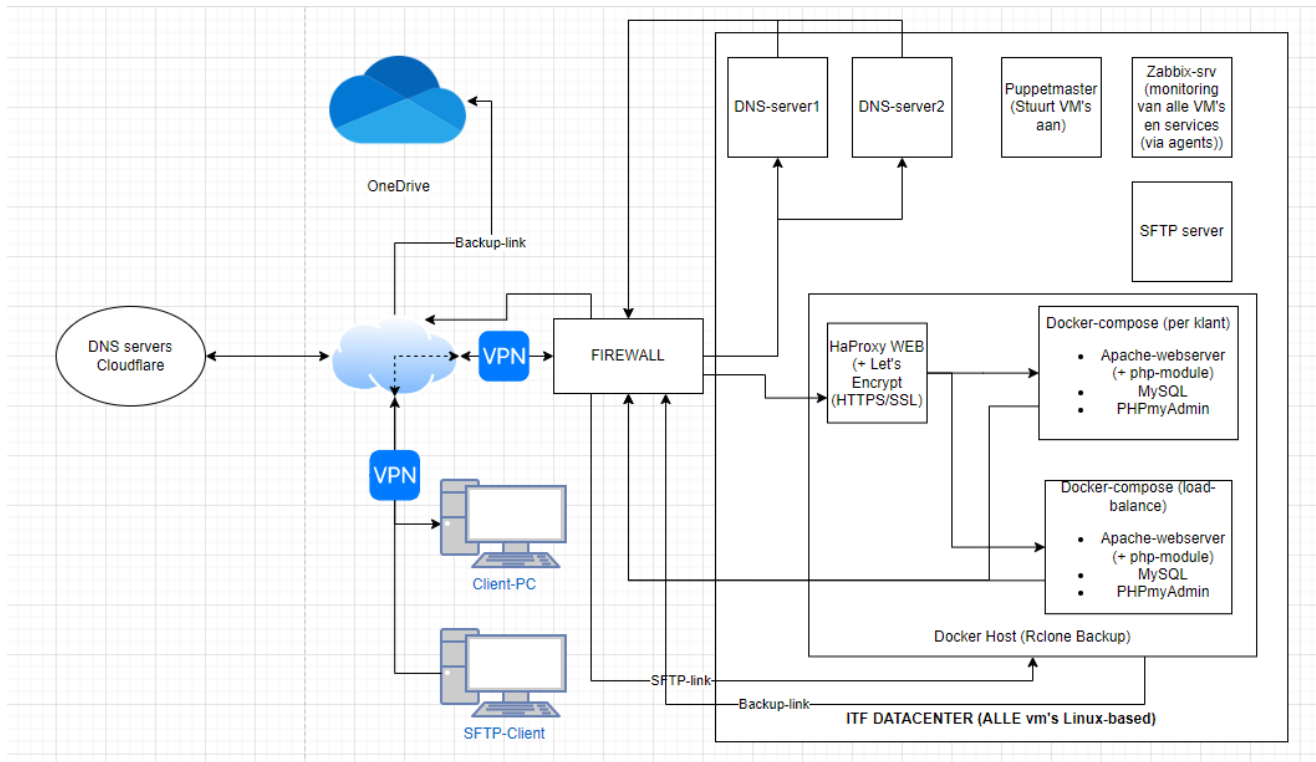
Vervolgens hebben we het over onze Ops report cards die we gekozen en geïmplementeerd hebben. Tot slot bespreken we onze testscenario's en onze plannen voor een volgende versie.

Veel leesplezier!

2 TECHNISCHE BESCHRIJVING

In dit hoofdstuk beschrijven we de technische samenstelling en werking van de verschillende onderdelen van ons project.

2.1 Schematische voorstelling



Schema 30/05/2022 13:00

Hoe werkt ons project? Ons project staat in het ITF-datacenter en is enkel toegankelijk via een VPN-verbinding. Als een gebruiker een website bij ons gehost heeft en hij wil deze website bekijken dan surft hij naar de website www.GekozenNaam.teamccs1.be.

Er wordt een verbinding gemaakt met de DNS servers van Cloudflare die het IP-adres teruggeven van de Docker Host. Het verzoek voor de website komt aan op de Docker Host bij de Haproxy DNS die het verzoek verder stuurt naar de juiste website cluster. Deze website cluster kiest dan welke web container er gebruikt wordt. Vervolgens krijgt de gebruiker zijn/haar website te zien.

De Puppetmaster/NTP server/Password safe (linksboven in het datacenter) zorgt dat de configuratie consistent en up to date blijft. Deze zorgt er ook voor dat de tijd op alle machines hetzelfde is. Daarnaast staan ook alle login-gegevens van alle machines hierop opgeslagen in een password safe.

De Zabbix server (rechtsboven in het datacenter) zorgt voor de monitoring. Dit zorgt ervoor als er een probleem opduikt dat we hier direct een melding van krijgen.

Op elke machine worden ook regelmatig back-ups genomen die worden opgeslagen in OneDrive.

2.2 Werking onderdelen

2.2.1 Hoe kan de gebruiker ons bereiken? (DNS)

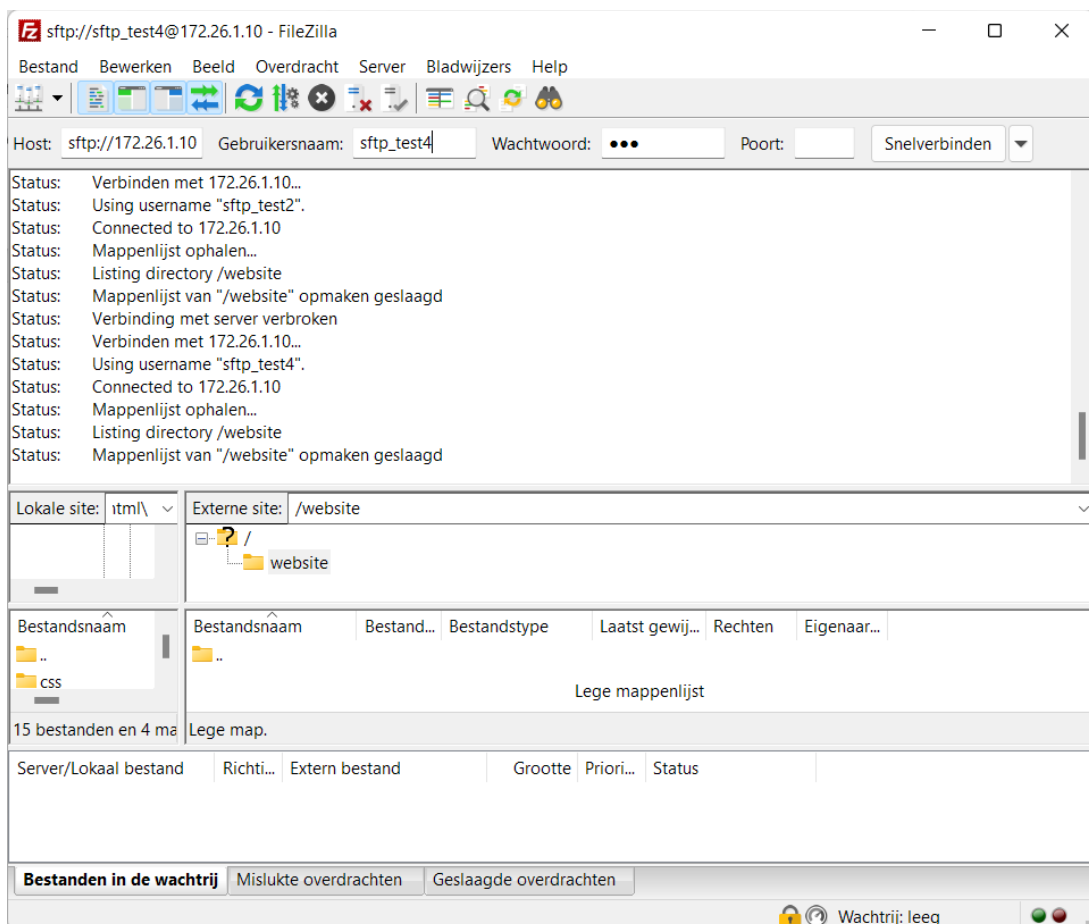
Hoe kan een gebruiker die een website bij ons host zijn website bereiken? Wanneer hij in zijn browser de URL www.GekozenNaam.teamccs1.be ingeeft dan wordt er verbinding gemaakt met de servers van Cloudflare die dan het IP-adres doorsturen van de webserver en dan krijgt de gebruiker zijn website te zien.

2.2.2 Bestanden uploaden (SFTP)

Elke gebruiker zal aanmeldgegevens krijgen om toegang te hebben tot de file-storage van de webserver. Deze toegang geschiedt over het met SSL beveiligde SFTP-protocol, dat op SSH gebaseerd is.

Met deze inloggegevens kan dan eenvoudig m.b.v. een programma zoals FileZilla, een verbinding opgezet worden. Dankzij de chroot-configuratie, krijgt de gebruiker enkel toegang tot zijn persoonlijke map, zonder dat het mogelijk is om hoger in de mappenstructuur te navigeren.

De bestanden in deze mappen worden tot slot gesynchroniseerd naar de locaties waar de bestanden bereikbaar zijn voor de webserver, zodat de inhoud aan surfers en klanten kan worden getoond, zonder dat alle gegevens zomaar voor iedereen toegankelijk zijn.

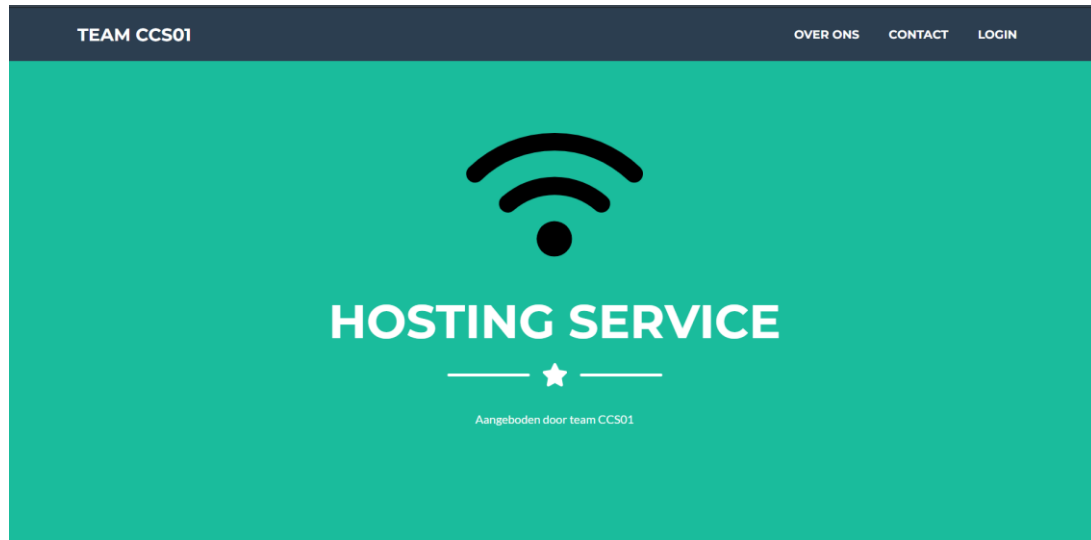


2.2.3 Websites hosten (webservers)

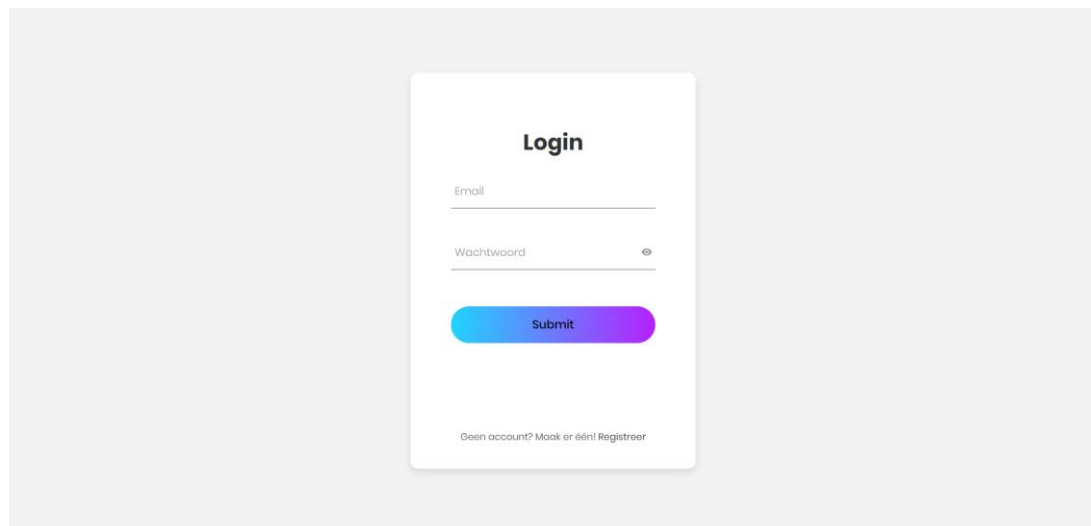
Wanneer de klant zijn/haar bestanden heeft geüpload wordt er een web container (kleine webserver) aangemaakt. De bestanden van de klant worden hierin geplaatst en dan kan de klant naar zijn website surfen.

2.2.4 Userinterface

De userinterface is een php pagina waar de klant de mogelijkheid heeft om een account te creëren, in te loggen en een domein kan aanmaken waarvan hij de gewenste php versie kan selecteren. Wanneer de gebruiker nog geen account heeft/niet ingelogd is zal hij verwelkomd worden op de homepage.



Door rechts vanboven op de login link te klikken wordt de gebruiker doorverwezen naar een inlogpagina.



Hier wordt het wachtwoord en het email adres van de gebruiker gevraagd. Wanneer de gebruiker zijn gegevens invult en op de submit knop drukt wordt het volgende php script gebruikt ter validatie.

```

81 <?php
82 if(isset($_POST['submit']))
83 {
84     $email = $_POST['mail'];
85     $ww = $_POST['pass'];
86
87     $servername = "localhost";
88     $username = "userinterface";
89     $password = "itf";
90     $dbname = "users";
91
92     // Create connection
93     $conn = new mysqli($servername, $username, $password, $dbname);
94     // Check connection
95     if ($conn->connect_error) {
96         die("Connection failed: " . $conn->connect_error);
97     }
98
99     $sql = "SELECT password FROM user WHERE email = '$email'";
100     $result = $conn->query($sql);
101
102     if ($result->num_rows > 0) {
103         // output data of each row
104         while($row = $result->fetch_assoc()) {
105
106             $hash = $row['password'];
107
108             if (password_verify($ww, $hash)) {
109                 header('Location: /index.php#domein');
110             }
111
112             else {
113                 ?>
114                 <script>alert("Het ingevoerde wachtwoord is fout.")</script>
115                 <?php
116                     }
117                 }
118             } else {
119                 ?>
120                 <script>alert("Het ingevoerde email adres is fout.")</script>
121                 <?php
122                     }
123                 $sql = "SELECT id FROM user WHERE email = '$email'";
124                 $result = $conn->query($sql);
125
126                 if ($result->num_rows > 0) {
127                     while($row = $result->fetch_assoc()) {
128                         $id = $row['id'];
129                     }
130                 }
131                 $conn->close();
132             }
133             $var_value = $id;
134             $_SESSION['varname'] = $var_value;
135             ?>

```


De code op lijn 82 zorgt ervoor dat het script begint te werken wanneer er op de submit knop gedrukt wordt. De code van lijn 84 tot 97 zal een connectie met de database creëren en controleren of deze gelukt is. Op lijn 99 wordt er een sql query gespecificeerd, lijn 100 zal dit opslaan en in een variable zetten.

Vervolgens wordt de hash van het wachtwoord uit de database gehaald en onder het \$hash variable opgeslagen. Op lijn 108 wordt er gecontroleerd of de opgeslagen hash overeenkomt met de hash van het door de user ingegeven wachtwoord, als dit het geval is wordt de gebruiker doorverbonden naar het gedeelte waar hij een domein aan kan maken. Het script dat op lijn 121 begint vraagt het id op van de gebruiker. Dit zet hij in een session variable zodat de user gegevens meegegeven kunnen worden aan de homepage.

Wanneer de user nog geen account heeft kan hij naar de registreer pagina gaan. Hier krijgt hij de mogelijkheid een account te creëren. Het onderstaande script stuurt de ingevulde gegevens naar de database.

```
if(isset($_POST['submit']))
{
    $email = $_POST['email'];
    $naam = $_POST['gebrnaam'];
    $ww = $_POST['ww'];
    $secpas = password_hash($ww, PASSWORD_DEFAULT);
    $sql = "INSERT INTO user (naam, email, password) VALUES ('$naam', '$email', '$secpas')";
    if (mysqli_query($conn, $sql)) {
        echo "New record has been added successfully !";
    } else {
        echo "Error: " . $sql . ":-" . mysqli_error($conn);
    }
    mysqli_close($conn);
    header('Location: /login/login.php');
}
```

De variable secpas is een gehashte versie van het wachtwoord, deze wordt meegegeven in de database zodat er geen wachtwoorden zichtbaar zijn in leesbare tekst. Het header commando zal de gebruiker doorsturen naar de login pagina waar hij kan inloggen met zijn nieuw account.

Wanneer de user ingelogd is zal hij op de homepage de mogelijkheid krijgen om een domeinnaam te selecteren met de bijhorende PHP-versie. Deze zullen met het volgende script aan de database toegevoegd worden. Er zal ook gecontroleerd worden of de gebruikersnaam nog niet in gebruik is.

```

<?php
}

if(isset($_POST['submit']))
{
    $servername='db2';
    $username='userinterface';
    $password='itf';
    $dbname = "users";
    $conn=mysqli_connect($servername,$username,$password,$dbname);

    $naam = $_POST['dn'];
    $versie = $_POST['phpversie'];
    $sql = "SELECT naam FROM `domein` WHERE naam = '$naam'";
    $result = $conn->query($sql);
    if ($result->num_rows > 0) {
        // output data of each row
        while($row = $result->fetch_assoc()) {
            if ($row['naam'] == $naam ) {
                >
                <script>alert("De gekozen domeinnaam is al in gebruik")</script>
            }
        }
    }
    else {
        $sql = "INSERT INTO domein (naam, user_id, versie) VALUES ('$naam', '$var_value', '$versie')";
        if (mysqli_query($conn, $sql)) {
            echo "<script>alert('Het domein " . $naam . " met PHP versie " . $versie . " is succesvol aangemaakt!')</script>";
            echo "<meta http-equiv='refresh' content='0'; url='http://172.26.1.4/'>";
        }
    }
}

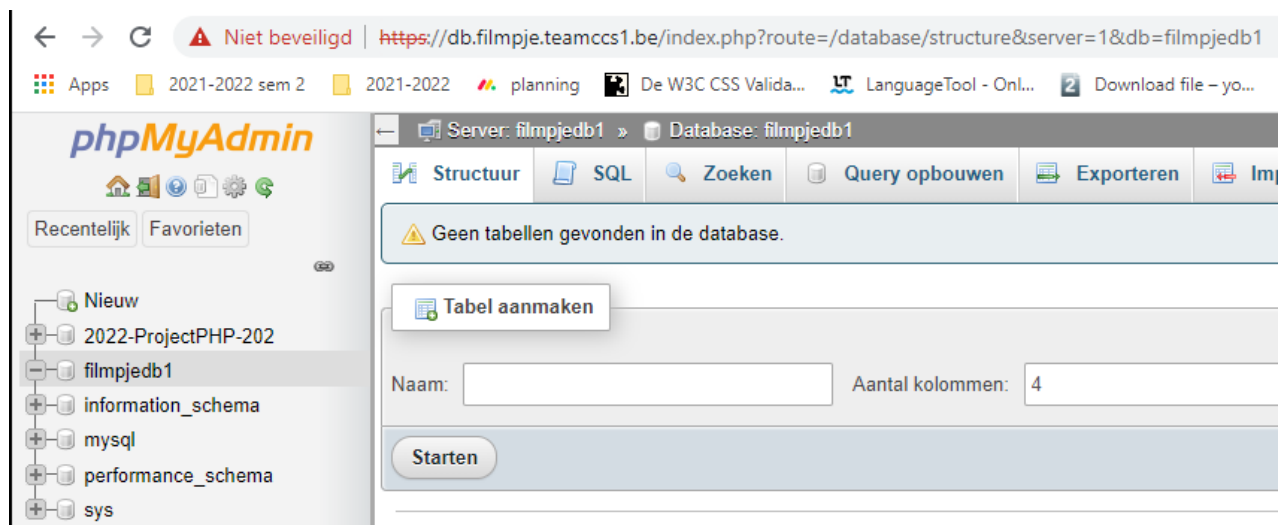
mysqli_close($conn);
}

```

2.2.5 Data van de websites opslaan (databases)

Wanneer de klant een sub-domein aanmaakt dan wordt er automatisch ook een database aangemaakt met de zelfde naam als de website aangemaakt. Dus als ik een website www.bakkerdruyts.teamccs1.be aanmaak dan wordt er een database met url db.bakkerdruyts.be aangemaakt.

Elke gebruiker heeft een SQL container die beschikbaar is het het adres db.bakkerdruyts.teamccs1.be. Dan kan de gebruiker inloggen met gebruikersnaam en als wachtwoord gebruikersnaampassword. Dit is een standaard wachtwoord dat de gebruiker onmiddellijk moet veranderen.



2.2.6 Beveiliging

Hoe is ons systeem beveiligd? We hebben meerdere lagen van beveiliging toegepast om ons systeem zo goed mogelijk te beveiligen. We hebben een password policy toegepast die ervoor zorgt dat de gebruikers en het IT-personeel wachtwoorden van minstens 10 karakters moeten gebruiken.

<https://www.xmodulo.com/set-password-policy-linux.html>

We hebben er ook voor gezorgd dat alle gebruikers enkel in hun mappenstructuur kunnen en enkel rechten hebben op hun bestanden. Vervolgens hebben we ook onze verschillende toestellen beveiligd door het Operating System zo goed mogelijk te beveiligen aan de hand van 5 topics/tools.

We maken hier bijvoorbeeld gebruik van Host firewall rules, fail2ban, banner, ...

2.2.7 Beschikbaarheid

Om ervoor te zorgen dat de websites van onze gebruikers bijna altijd beschikbaar zijn hebben we meerdere principes van redundantie toegepast. Elke website draait op twee web containers. Als 1 container faalt is er een andere die nog wel werkt.

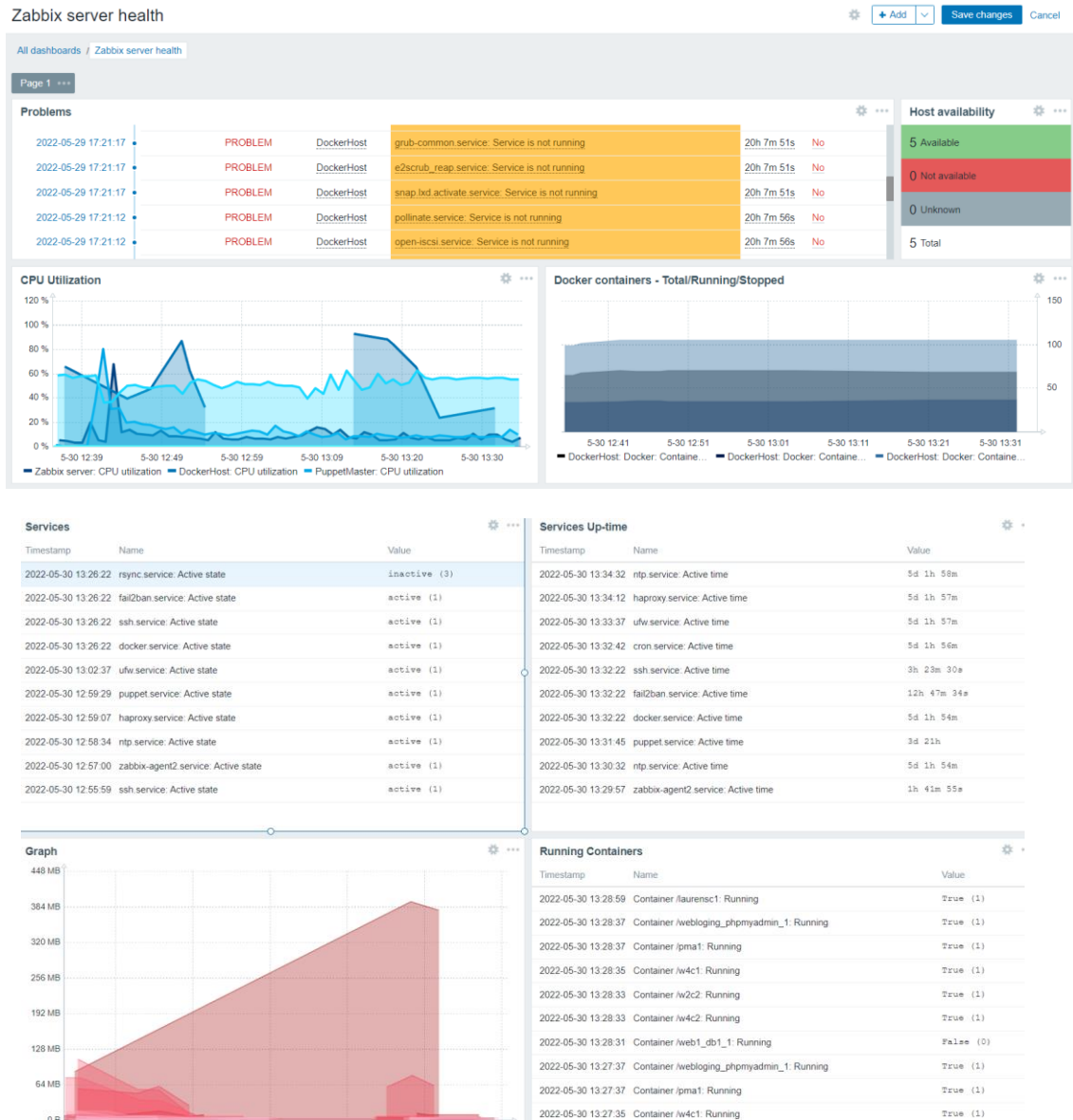
2.2.8 Configuratie synchronisatie

Als we de configuratie op meerdere toestellen veranderen gaan we niet met elk toestel verbinden en dan de verandering manueel ingeven. De verandering in de configuratie wordt gesynchroniseerd via een configuratie server (Puppetmaster). Alle configuratie bestanden staan op deze server en deze server kan de configuratie op elke toestel veranderen op elk moment.

2.2.9 Monitoring

Om een overzicht te hebben van alle websites en eventuele toestellen die niet meer werken hebben we gebruik gemaakt van Zabbix. Met deze monitoring tool hebben we een goed overzicht over alle toestellen en over alle websites en kunnen we snel ingrijpen als dat nodig is.








2.2.10 Back-ups

Als er een gebruiker per ongeluk zijn bestanden verwijderd of het datacenter ligt plat. Dan kunnen we alle bestanden en toestellen snel terug opzetten dankzij onze back-ups in de Cloud. We back-uppen alle bestanden van de gebruikers en alle configuratie van onze systemen naar de Cloud zodat we op elk moment een back-up hebben van alle bestanden en deze kunnen terugzetten wanneer nodig. De back-ups worden elk uur genomen en worden opgeslagen in een business OneDrive. Als we dit project verder af maken, zouden we gebruik maken van een betalende versie van OneDrive zodat er meer opslag ruimte beschikbaar is.

Mijn bestanden > ProjectHosting > Backup

| Naam ▾ | Gewijzigd ▾ | Gewijzigd door ▾ | Bestandsgrootte ▾ | Delen |
|--|--------------------------|------------------|-------------------|-------|
|  Client | Ongeveer een uur gele... | Siebe Van Rompay | 12 items | Privé |
|  LBW | 5 minuten geleden | Siebe Van Rompay | 2 items | Privé |
|  Master | Ongeveer een uur gele... | Siebe Van Rompay | 4 items | Privé |

```
GNU nano 4.8 /tmp/crontab.t1K8UH/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 * * * * rclone copy /etc/puppetlabs/code/environments/production/ Backup:/ProjectHosting/Backup/Master >/dev/null 2>&1
```

```
0 * * * * rclone copy /home/ Backup:/ProjectHosting/Backup/Client >/dev/null 2>&1
```

```
0 * * * * rclone copy /home/ccs1/backups Backup:/ProjectHosting/Backup/LBW/backups >/dev/null 2>&1
0 * * * * rclone copy /home/ccs1/webconfig/ Backup:/ProjectHosting/Backup/LBW/webconfig >/dev/null 2>&1
```

3 PROCEDURES EN HANDLEIDINGEN

In dit hoofdstuk bespreken we onze plannen als er zich een ramp voordoet. Dan bespreken we een handleiding over het volledig systeem voor onze IT collega's en tot slot hebben we het in dit hoofdstuk ook nog over een handleiding voor de gebruikers met een basis IT kennis.

3.1 Disaster recovery procedures

Deze procedures zijn opgesplitst per onderdeel daarna staat er nog een procedure voor moet het volledig datacenter kapotgaan.

3.1.1 Zabbix monitoring server kapot

Wanneer de Zabbix server stuk gaat, dan lezen we de docker logs op de Zabbix server VM en nemen we de nodige acties om dit op te lossen. We zullen dit hooguit 4 keer moeten doen doordat de Zabbix server wordt gestart aan de hand van 4 docker containers.

3.1.2 Haproxy DNS/load balancing container kapot

Wanneer de Haproxy DNS container stuk gaat dan verwijderen we deze container. Op de webserver staat een back-up Haproxy DNS container klaar om opgezet te worden. We activeren de back-up Haproxy DNS container en we kunnen terug verder. Als de back-up Haproxy container goed werkt maken we hier een back-up van.

3.1.3 Een web container kapot

Als er 1 web container stuk gaat, is dit geen ramp want dan is er nog een web container die dan alle verzoeken overneemt. Normaal gezien zien we dit dan op de Zabbix monitoring te zien zijn en dan wanneer er weinig verbindingen zijn naar de resterende web container (s 'nachts) dan verwijderen we alle web containers en maken we twee nieuwe werkende containers.

3.1.4 Meerdere web containers kapot

Wanneer er meerdere web containers kapot gaan krijgen we een alert op onze Zabbix monitoring. Dan verwijderen we de kapotte containers en maken 2 nieuwe containers via de configuratie die op de Puppetmaster en de bestanden van de klant die geback-upt staan in de Cloud.

3.1.5 Een database kapot

Als er een database stuk gaat dan verwijderen we de kapotte database. We maken een nieuwe database container aan met de configuratie die op de Puppetmaster staat en met de back-up van de database die in de Cloud staat.

3.1.6 Meerdere databases kapot

Wanneer er meerdere databases kapotgaan geeft dit aan dat er een fout in de webserver zit en niet alleen in de containers. Bij deze fout gaan we de webserver back-uppen. In de achtergrond gaan we dan een nieuwe webserver creëren met de back-up van de configuratie die op de Puppetmaster staat.

3.1.7 De volledige webserver kapot

Als de volledige webserver kapotgaat dan verwijderen we de kapotte webserver. Vervolgens maken we een nieuwe webserver met de configuratie die op de Puppetmaster staat en met de bestanden van de klanten die geback-upt staan in de Cloud.

3.1.8 SFTP-server kapot

Wanneer de SFTP-server kapotgaat dan verwijderen we de kapotte SFTP-server en dan maken we een nieuwe SFTP server met de configuratie die op de Puppetmaster staat en met de bestanden van de gebruikers die geback-upt staan in de Cloud. Het kan dan zijn dat er gebruikers zijn die bestanden hebben geüpload die nog niet geback-upt zijn.

In dat geval sturen we een mail naar de gebruikers om te laten weten dat als ze de afgelopen 24 uur bestanden hebben geüpload dat ze die beter nog eens uploaden om zeker te zijn dat de laatste versie van hun bestanden op de SFTP-server staat.

3.1.9 Puppet master kapot

Als de Puppetmaster kapotgaat dan is dit een probleem dat niet meteen merkbaar is voor de klanten. Als dit gebeurt dan verwijderen we de kapotte Puppetmaster en dan maken we een nieuwe Puppetmaster met de configuratie die geback-upt staat in de Cloud.

3.1.10 Volledig datacenter kapot

Wanneer het volledig datacenter kapotgaat dan sturen we eerst een mail naar al onze gebruikers dat hun websites een tijd niet beschikbaar zijn maar dat we al hun bestanden geback-upt hebben in de Cloud. Vervolgens maken we eerst een nieuwe Puppetmaster met de configuratie die geback-upt staat in de Cloud. Daarna maken we alle andere VM's aan met de Puppetmaster. Dan sturen we een mail naar de klanten waarin staat dat we terug succesvol operationeel zijn.

3.2 Handleiding voor IT-medewerkers

In dit hoofdstuk gaan we een handleiding schrijven voor mensen met een sterke IT-kennis.

3.2.1 Docker structuur

Voor elke website hebben we twee docker containers draaien op onze docker host. Deze twee containers worden geloadbalanced door een Haproxy container die ook de requests naar de juiste website cluster stuurt.

```

global

defaults
    mode http
    timeout connect 5000ms
    timeout client 50000ms
    timeout server 50000ms

frontend stats
    bind *:8404
    stats enable
    stats uri /
    stats refresh 10s

frontend myfrontend
    bind *:80
    bind :::443 ssl crt /usr/local/etc/haproxy/www.websitel.teamccsl.be.pem crt /usr/

    http-request redirect scheme https unless { ssl_fc }
    use_backend websitelcluster if { ssl_fc_sni www.websitel.teamccsl.be }
    use_backend pmal if { ssl_fc_sni db.websitel.teamccsl.be }

backend websitelcluster
    server wlcl wlcl:80 check
    server wlc2 wlc2:80 check

backemd pmal
    server pmal pmal:80 check

```

Haproxy config

De instellingen bij "defaults" zijn standaard instellingen. "frontend stats" dient om de status van de webcontainers te controleren. "frontend myfrontend" dient om de juiste website te selecteren. De requests komen binnen op poort 443 waar alle SSL-certificaten aan gekoppeld worden. Vervolgens worden de http-verbindingen die er binnenkomen omgezet naar https-verbindingen. Dan worden de requests naar de juiste backend cluster gestuurd.

De requests voor www.website1.teamccs1.be worden naar de website1cluster gestuurd en de requests voor db.website1.teamccs1.be worden naar de pmal cluster gestuurd.


```
GNU nano 4.8
FROM haproxy:1.7
```

Haproxy docker file

```
version: "3"
services:
  hpl:
    build: .
    container_name: hpl
    volumes:
      - ./:/usr/local/etc/haproxy/
    ports:
      - "80:80"
      - "8404:8404"
      - "443:443"
networks:
  default:
    external:
      name: external-example
```

Haproxy docker-compose file

Dit bestand gaat de Haproxy container bouwen en deployen. Bij "build" wordt er gebruik gemaakt van de bovenstaande Haproxy docker file. De container naam wordt "hp1". De map /home/ccs1/webconfig/hp wordt gekopieerd in de container naar de map /usr/local/etc/haproxy. Dit is nodig om de SSL-certificaten in de container te plaatsen.

Daarna zetten we poort 80 en 443 open voor de webverbindingen en poort 8404 voor de monitoring. Tot slot maken we gebruik van een netwerk "external-example" om ervoor te zorgen dat alle containers in hetzelfde netwerk zitten en met elkaar kunnen communiceren.

```

version: "3"

services:
  wlc1:
    build: ./
    container_name: wlc1
    volumes:
      - ./app_rsync:/var/www/html
    ports:
      - "8080:80"

  wlc2:
    container_name: wlc2
    build: ./
    volumes:
      - ./app_rsync:/var/www/html
    ports:
      - "8081:80"

  db1:
    # container_name: db1
    image: mysql
    environment:
      MYSQL_ROOT_PASSWORD: test
      MYSQL_DATABASE: 2022-ProjectPHP-202
      MYSQL_USER: homestead
      MYSQL_PASSWORD: secret
    volumes:
      - ./mysql/data:/var/lib/mysql
  phpmyadmin:
    container_name: pmal
    image: phpmyadmin/phpmyadmin
    environment:
      PMA_HOST: db1
    depends_on:
      - db1

networks:
  default:
    external:
      name: external-example

```

Docker Compose website 1

```

FROM php:7.4-apache
COPY website1.conf /etc/apache2/sites-available/000-default.conf
RUN a2enmod rewrite
RUN a2ensite 000-default.conf
RUN docker-php-ext-install mysqli && docker-php-ext-enable mysqli
RUN apt-get update && apt-get upgrade -y

```

Dockerfile website 1

3.2.2 DNS structuur

Hoe werkt onze DNS structuur? Wanneer een gebruiker is verbonden via een VPN-verbinding en deze gebruiker wil surfen naar www.website1.teamccs1.be. Dan wordt de DNS query gestuurd naar de DNS name servers van Cloudflare deze geven dan het IP-adres van de webserver terug.

Vervolgens komen we op de webserver terecht op de Haproxy container die zegt dat je naar de cluster van website 1 moet. De cluster van website 1 kiest dan willekeurig een van de twee web containers die beschikbaar zijn. Dan krijg je website 1 te zien.

3.2.3 Hoe is onze domeinnaam geregistreerd?

Onze domeinnaam teamccs1.be staat geregistreerd bij Combell maar we gebruiken de name servers van Cloudflare omdat deze de mogelijkheid hebben om op een eenvoudige manier met API-calls SSL certificaten aan te vragen.

3.2.4 SFTP

De SFTP-server wordt geprovisioneerd m.b.v. Puppet, waardoor de configuratie hiervan centraal kan geschieden.

```
node 'lm-puppetclient.puppet' {
  class { 'ssh::server':
    options => {
      'Match Group sftp_users' => {
        'ChrootDirectory' => '/home/%u',
        'ForceCommand' => 'internal-sftp -d /website',
        'PasswordAuthentication' => 'yes',
        'AllowTcpForwarding' => 'no',
        'X11Forwarding' => 'no',
      },
      'PermitRootLogin' => 'no',
      'Port' => [22],
      'Subsystem' => 'sftp internal-sftp',
      'PrintMotd' => 'no',
    },
  },
}
```

In bovenstaand screenshot (uit site.pp op de Puppetmaster) staat de configuratie voor de SSH-daemon, die verantwoordelijk is voor sftp. Merk op dat alle gebruikers uit de groep 'sftp_user' *gechroot* worden. Dit wil zeggen dat ze bij het aanmelden enkel toegang verkrijgen tot hun eigen hoofdmap en subdirectories hiervan, zonder dat de mogelijkheid bestaat om aan bovenliggende bestanden te geraken. Bovendien is de sftp-server gescheiden van de andere onderdelen van ons systeem.

```

$file = file("/home/ccs1/sshfs_userfiles/users.txt").chomp
$users = split($file, '[,]')

group { 'sftp_users':
    ensure => 'present',
}

$users.each |String $user| {
    $namepass = split($user, '[:]')

    user { "${namepass[0]}":
        groups => 'sftp_users',
        shell => '/usr/sbin/nologin',
        password => Sensitive("${namepass[1]}"),
    }

    file { "/home/${namepass[0]}":
        ensure => 'directory',
        owner => 'root',
        group => "${namepass[0]}",
        mode => '0755',
    }

    file { "/home/${namepass[0]}/website":
        ensure => 'directory',
        owner => "${namepass[0]}",
        group => 'sftp_users',
        mode => '0755',
    }

    file { "/home/${namepass[0]}/website/public":
        ensure => 'directory',
        owner => "${namepass[0]}",
        group => "${namepass[0]}",
        mode => '0755',
    }
}
}

```

In dit screenshot wordt duidelijk hoe de sftp-users worden aangemaakt vanuit Puppet. Alle gebruikersnamen worden door een Python-script uit de database gehaald, in combinatie met het gehashte paswoord. Hierna worden gekeken of de users reeds bestaan - en indien niet - ook aangemaakt. Let op de parameter 'shell', die ervoor zorgt dat deze users niet rechtstreeks via ssh toegang kunnen verkrijgen.

```

PS C:\Users\Laurens> ssh laumatt@172.26.1.12
ALERT! You are entering into a secured area!
- This service is restricted to authorized users only
- This service is being monitored to detect improper use and other illicit activity
- Your IP, Login Time, Username are being monitored
- Do not expect privacy when using this service

Unauthorized access will be fully investigated and we will take legal actions!
laumatt@172.26.1.12's password:
This service allows sftp connections only.
Connection to 172.26.1.12 closed.

```

Nadien wordt de directorystructuur aangemaakt voor de gebruiker, volgens de regels die chroot-toegang vereist. De chroot-map moet namelijk 'root' als eigenaar hebben, en mag niet beschrijfbaar zijn voor de gebruiker. Schrijven is pas mogelijk in de subfolder 'website'.

De folder 'public' wordt aangemaakt, waarin voor eenvoudige websites zich de zogenaamde 'document root' bevindt. Voor grotere PHP-projecten, mag deze folder door de eigen 'public'-folder overschreven worden.

Standaard bevindt zich in deze folder ook een default pagina van teamccs1 uit.

3.2.5 Puppetmaster

Om de configuratie zoveel mogelijk vanop 1 centrale plaats te kunnen regelen, maken we, zoals hierboven al duidelijk is geworden, gebruik van Puppet. Hierbij komt de gewenste staat van configuratie van het apparaat in een zogenaamde manifest-file op de puppetserver. Kort door de bocht, zal deze server de gewenste staat dan via een beveiligde verbinding doorsturen naar deze apparaten (de puppetclients), die hierop de nodige configuratiewijzigingen zullen doorvoeren.

```
ccs1@Team01-SFTP:~$ sudo /opt/puppetlabs/bin/puppet agent -t
Info: Using environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales
Info: Loading facts
Info: Caching catalog for lm-puppetclient.puppet
Info: Applying configuration version '1653933798'
Notice: Applied catalog in 1.00 seconds
```

(Voorbeeld toepassen van config op master -> client)

3.2.6 Script om de websites aan te maken:

In de achtergrond is een script constant aan het kijken of er nieuwe websites moeten aangemaakt worden op de webserver. Wanneer dit script in de database achter de gebruikerswebsite een nieuwe website ziet dan wordt het script uitgevoerd.

```
ccs1@loadbalancingwebcontainers1:~/webconfig/scripts$ ls -l
total 96
-rwxr-xr-x 1 root root 476 May 24 13:09 AddDNSRecord.sh
-rwxr-xr-x 1 root root 371 May 24 13:53 AddLinesToHapoxy.sh
-rw-r--r-- 1 root root 9313 May 28 14:49 AddWebsite.py
-rw-r--r-- 1 root root 1438 May 26 14:14 AddWebsite.pybak
-rwxr-xr-x 1 root root 453 May 24 08:50 bootscript1.sh
-rw----- 1 root root 176 May 24 09:16 cloudflare.ini
-rw-r--r-- 1 root root 69 May 28 13:30 createddomains.txt
-rwxr-xr-x 1 root root 901 May 24 13:50 CreateNewDockerComposeFiles.sh
-rwxr-xr-x 1 root root 85 May 24 13:50 DeployNewWebsites.sh
-rwxr-xr-x 1 root root 283 May 23 10:17 downscript1.sh
-rw-r--r-- 1 root root 234 May 27 13:17 haproxy0.cfg
-rw-r--r-- 1 root root 234 May 27 12:55 haproxy1.cfg
-rw-r--r-- 1 root root 1551 May 28 13:37 haproxy3.cfg
-rw-r--r-- 1 root root 1673 May 28 13:37 haproxy4.cfg
-rwxrwxrwx 1 root root 168 May 24 19:40 ResetPasswordscript.sh
-rwxr-xr-x 1 root root 86 May 24 11:39 RestartHapoxy.sh
-rw-r--r-- 1 root root 102 May 24 10:12 serverlist.txt
-rwxrwxrwx 1 root root 334 May 26 15:04 SSLscript.sh
-rwxr-xr-- 1 root root 270 May 27 15:53 sync.sh
```

Het script AddWebsite.py gaat dan een nieuwe website aanmaken. Dit script bevat meer dan 200 lijnen code dus voor dit document ga ik de algemene werking van het script uitleggen.

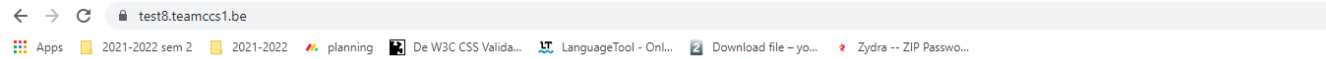
Wanneer een nieuw domein wordt aangevraagd dan wordt er gekeken welke gebruiker dit domein heeft aangevraagd, deze waarde wordt opgeslagen in de variabele "user".

Vervolgens worden de nodige mappen waar de bestanden van de gebruiker in moeten komen aangemaakt. Dan wordt er een DNS record aangemaakt bij de name servers van Cloudflare voor de domeinen www.domeinnaam.teamccs1.be en db.domeinnaam.teamccs1.be. Op het eerste domein komt de website van de gebruiker en op het tweede domein komt de toegang tot de database die gekoppeld is aan de website.

Daarna wordt de juiste Dockerfile gekozen aan de hand van de PHP-versie die de gebruiker gekozen heeft. Dan wordt er een configuratie bestand voor de website aangemaakt. Vervolgens wordt er een Docker Compose bestand aangemaakt dat de web containers van de gebruiker moet aanmaken.

Dan worden de nodige SSL-certificaten aangevraagd en gekoppeld met de website. Daarna wordt de website toegevoegd aan de configuratie van Hapoxy en wordt de nieuwe configuratie van Hapoxy ingeladen.

Daarna wordt er een welkom pagina in de web container geplaatst. Tot slot worden de web containers gestart en dan krijgt de gebruiker deze pagina te zien wanneer hij naar zijn domein surft.



Welkom

Als je deze pagina ziet heb je succesvol een domein aangemaakt bij Teamccs1. Het is de bedoeling dat hier jouw website komt te staan die je zelf gemaakt hebt. We hopen dat je van onze hosting service geniet.

Vragen?

Als je vragen hebt kan je terecht op onze mailbox: support@teamccs1.be.

3.2.7 Backup

De back-ups worden genomen via Rclone. Deze tool is er speciaal voor gemaakt om eenvoudig back-ups te nemen. Op elk toestel wordt Rclone geïnstalleerd en geconfigureerd voor OneDrive en vervolgens ingesteld zodat enkel de belangrijke documenten van de gebruiker en onszelf geback-upt zullen worden naar OneDrive. Om de back-ups te automatiseren is de Rclone commando om de back-ups te maken in de crontab gezet die deze elk uur zal uitvoeren.

Mijn bestanden > ProjectHosting > Backup

| Naam | Gewijzigd | Gewijzigd door | Bestandsgrootte | Delen |
|--------|--------------------------|------------------|-----------------|-------|
| Client | Ongeveer een uur gele... | Siebe Van Rompay | 12 items | Privé |
| LBW | 5 minuten geleden | Siebe Van Rompay | 2 items | Privé |
| Master | Ongeveer een uur gele... | Siebe Van Rompay | 4 items | Privé |

```
GNU nano 4.8 /tmp/crontab.mDqIW0/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow command
0 * * * * rclone copy /etc/puppetlabs/code/environments/production/ Backup:/ProjectHosting/Backup/Master >/dev/null 2>&1
```

```
0 * * * * rclone copy /home/ Backup:/ProjectHosting/Backup/Client >/dev/null 2>&1
```

```
0 * * * * rclone copy /home/ccs1/backups Backup:/ProjectHosting/Backup/LBW/backups >/dev/null 2>&1
0 * * * * rclone copy /home/ccs1/webconfig/ Backup:/ProjectHosting/Backup/LBW/webconfig >/dev/null 2>&1
```

3.2.8 Monitoring

Monitoring wordt gedaan door Zabbix, een open-source monitoring tool dat gebruikt wordt voor het monitoren van netwerkservern en applicaties en helpt de gebruikers om de prestatie knelpunten in het netwerk te identificeren. De tool biedt verschillende functies die het proces van monitoring en het oplossen van problemen vergemakkelijken.

De Zabbix tool biedt de mogelijkheid van real-time monitoring. Alle items kunnen onmiddellijk in grafieken worden weergegeven.

We hebben voor Zabbix 6.0 gekozen omdat vanaf 6.0, een template ter beschikking hebt in het webinterface om docker gegevens te monitoren. De server is opgezet aan de hand van 4 [containers](#) in een aparte virtual machine gedeceideerd voor monitoring.

Deze containers zijn:

- MySQL Database (Zabbix moet metrics opslaan in een database)
- Java Gateway (Zabbix is een applicatie dat Java gebruikt)
- Zabbix Server
- Zabbix Web Interface

Met deze containers kan je het webinterface van Zabbix gebruiken, maar je gaat nog geen hosts hebben omdat er nergens een agent op staat. De agents moeten geconfigureerd worden op de systemen die je wilt weergeven in je web interface. We hebben voor Zabbix-agent2 gekozen aangezien je agent2 nodig hebt voor docker monitoring en voor Systemd gerelateerde monitoring.

- https://www.zabbix.com/container_images
- <https://www.zabbix.com/integrations/systemd>
- <https://hub.docker.com/r/zabbix/zabbix-agent2>

3.2.9 OS Security

Voor de beveiliging van ons systeem maken we gebruik van 5 tools/topics.

Als eerste gebruiken we fail2ban dit is een tool die brute force attacks op de ssh verbinding zal voorkomen door IP's te bannen voor een bepaalde tijd (bantime) na een bepaald aantal keren fout wachtwoord in te vullen (maxretry).

Dit is mogelijk door een file aan te maken '/etc/fail2ban/jail.local', in deze file kopiëren we de content van '/etc/fail2ban/jail.config' en voegen we bij '[sshd]' een extra lijn toe.

(https://www.fail2ban.org/wiki/index.php/Main_Page)


```
[sshd]
# To use more aggressive sshd modes set filter parameter "mode"
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage
#mode = normal
enabled = true
port = ssh
logpath = %(sshd_log)s
backend = %(sshd_backend)s
```

Als tweede gebruiken we een script dat alle bestanden bekijkt op mogelijke privilege escalation (bekijkt of er foute rechten zijn op bepaalde bestanden). Heb het script binnen gehaald met het wget commando via deze site (<https://pentestmonkey.net/tools/unix-privesc-check/unix-privesc-check-1.4.tar.gz>). Daarna unzip ik het bestand en gebruik ik het in een script dat elke lijn en één extra waar het woord 'WARNING' staat in een bestand schrijft.

```
#!/bin/bash
./unix-privesc-check-1.4.sh standard 2> /dev/null | grep "WARNING" -A 1 > privesc-result.txt
```

(<https://pentestmonkey.net/tools/audit/unix-privesc-check>)

Als derde hebben we ook gekozen om op elke server hostfirewall regels toe te passen aan de hand van UFW. We zijn vertrokken vanuit het least privilege principe (standaard niets toelaten). Hierna hebben we dan een voor een bepaalde poorten opengezet die deze verschillen per server want elke server heeft andere toepassingen.

Hier een klein overzicht van welke poorten er open staan op welke server.

| Server | Poorten |
|----------------------------------|--|
| Team01-PuppetMaster, Team01-SFTP | Poort 22 (ssh) Poort 8140 (Puppet) Poort 10050 (Zabbix Agent) Poort 10051 (Zabbix Trapper) Poort 123 udp (NTP) |
| Team01-DockerHost | Poort 22 (ssh) Poort 8140 (Puppet) Poort 10050 (Zabbix Agent) Poort 10051 (Zabbix Trapper) Poort 123 udp (NTP) |

| | |
|-------------------------|---|
| | Poort 80 (HTTP) Poort 443 (HTTPS) |
| Team01-DNSmaster/-slave | Poort 22 (ssh) Poort 10050 (Zabbix Agent) Poort 10051 (Zabbix Trapper) Poort 123 udp (NTP) Poort 53 (DNS) |
| Team01-ZabbixServer | Poort 22 (ssh) Poort 10050 (Zabbix Agent) Poort 10051 (Zabbix Trapper) Poort 123 udp (NTP) |

De vierde toepassing die we gebruiken is ervoor zorgen dat de ingegeven commando's niet worden opgeslagen in het bestand '~/.bash_history'. Dit doen we door eerst het bestand '~/.bash_history' te verwijderen en dan een symbolic link te maken tussen '/dev/null' en '~/.bash_history'. Dit zorgt ervoor dat alles gediscard zal worden.

Dit hebben we dan op elke vm toegepast.

```
sudo rm ~/.bash_history
ln -s /dev/null ~/.bash_history
```

Tot slot maken we ook gebruik van een 'Banner' deze zal te zien zijn voor je ingelogd bent op de server en zal informatie geven aan de persoon die probeert in te loggen.

```
PS C:\Users\ilias> ssh ccs1@172.26.1.30
ALERT! You are entering into a secured area!
- This service is restricted to authorized users only
- This service is being monitored to detect improper use and other illicit activity
- Your IP, Login Time, Username are being monitored
- Do not expect privacy when using this service

Unauthorized access will be fully investigated and we will take legal actions!
ccs1@172.26.1.30's password:
```

3.2.10 Password safe

Op de Puppetmaster staat een passwordmanager ingesteld die alle wachtwoorden van ons systeem bevat. De passwordmanager die we gebruikt hebben is keepassxc. Deze passwordmanager is beveiligd met een sterk wachtwoord dat nergens genoteerd staat.

```
ccsl@lm-puppetmaster:~$ sudo keepassxc-cli ls password
[sudo] password for ccsl:
Insert password to unlock password:
DNSmaster
DNSslave
DockerHost
DRP
PuppetMaster
SFTP
ZabbixServer
ccsl@lm-puppetmaster:~$
```

3.2.11 Alle admin wachtwoorden veranderen in een uur

Om alle wachtwoorden te veranderen op alle machines maken we gebruik van het volgend script.

```
GNU nano 4.
172.26.1.1
172.26.1.2
172.26.1.3
172.26.1.4
172.26.1.6
172.26.1.7
172.26.1.11
172.26.1.12
172.26.1.30
```

Serverlist.txt

Dit bestand bevat de IP adressen van al onze gebruikte VMs.

```
#!/bin/bash
IFS= read -rp "Enter username: " username
for server in $(cat serverlist.txt)
do echo "Server IP is: $server"
ssh "$server" "passwd '$username'"
done
```

```

New password: 00010loadbalancingweb
ccsl@loadbalancingwebcontainers1:~/web
Enter username: ccsl
Server IP is: 172.26.1.1
ccsl@172.26.1.1's password:
Current password: itf
New password: admin123!
Retype new password: admin123!
passwd: password updated successfully
Changing password for ccsl.

```

Bovenstaande afbeelding is het script dat uitgevoerd wordt. Er wordt eerst om een gebruikersnaam gevraagd. Vervolgens gaat het programma in de for loop kijken naar de IP-adressen in het bestand serverlist.txt. Voor elk IP-adres in dit bestand wordt er een SSH-verbinding gemaakt naar het IP. Dan wordt er gevraagd om het wachtwoord dat bij de gebruiker en het IP-adres hoort in te geven.

Daarna wordt er om het huidig wachtwoord gevraagd. Dit is om te verwijderen dat je twee keer hetzelfde wachtwoord ingeeft. Vervolgens wordt er om het nieuwe wachtwoord gevraagd en er wordt gevraagd om het nieuwe wachtwoord te bevestigen. Dan komt er de boodschap dat het wachtwoord succesvol veranderd is.

3.2.12 NTP-server en clients

Om een gelijke tijd te hebben op alle VM's binnen ons systeem hebben we de Puppetmaster gesynchroniseerd met meerdere officiële NTP-servers. Daarna hebben we alle andere VM's gesynchroniseerd met de Puppetmaster.

```

ccsl@lm-puppetmaster:~$ date
Mon May 30 15:24:17 CEST 2022
ccsl@lm-puppetmaster:~$ █

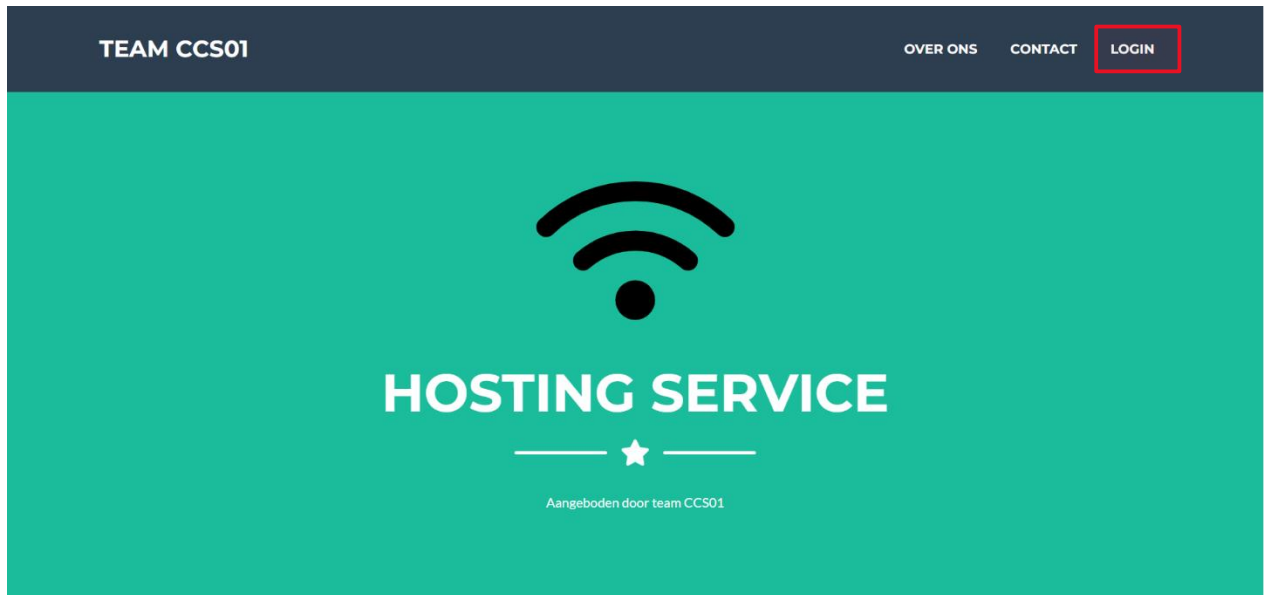
```

3.3 Handleiding voor gebruikers

In dit hoofdstuk staat een eenvoudige handleiding geschreven voor al onze gebruikers.

3.3.1 Hoe maak ik een account aan?

Surf naar <https://www.teamccs1.be/>, klik dan op de knop login.



U heeft nog geen logingegevens dus zal u zich moeten registreren, klik op de knop registreren.

The image shows a login form within a light gray rounded rectangle. At the top of the form, the word 'Login' is written in a large, bold, black font. Below it are two input fields. The first is labeled 'Email' in a light gray font. The second is labeled 'Wachtwoord' (Password) in a light gray font and has a small eye icon to its right, indicating a password field. Below these fields is a large, rounded button with a blue-to-purple gradient, containing the word 'Submit' in black text. At the bottom of the form, there is a red rectangular box containing the text 'Geen account? Maak er één! Registreer' in a light gray font.

Vul hier al u gegevens in en volg zeker de richtlijnen, klik op "Submit".

Maak een account

Email

Email

Gebruikersnaam (geen spaties gebruiken)

Naam

Wachtwoord (Geen : en , gebruiken)



Wachtwoord

Submit

Je wordt nu teruggestuurd naar het login scherm. Vul hier het gekozen email en wachtwoord in en druk op "Submit".

Login

Email

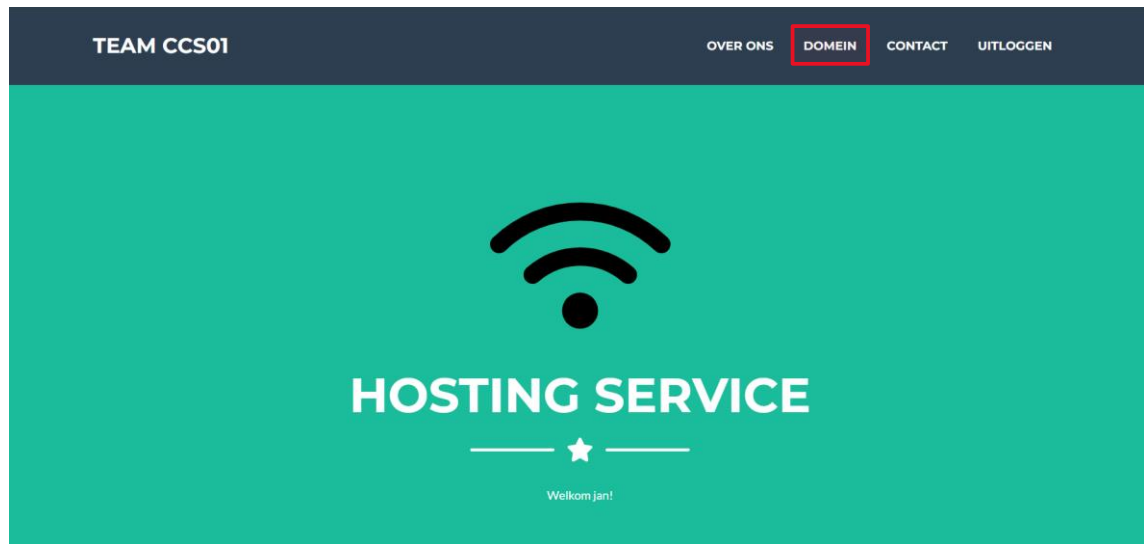
Wachtwoord



Submit

Geen account? Maak er één! Registreer

Nu ben je ingelogd en kan je verder met het aanmaken van een domeinnaam.



3.3.2 Hoe maak ik een domeinnaam aan?

Als u op de knop "Domein" hebt geklikt, of naar onder scrolt. Dan kan u een domeinnaam ingeven en een PHP-versie kiezen. Klik dan op "Submit" en u domein wordt aangemaakt, het kan ongeveer 15 minuten duren totdat je domein up and running is.

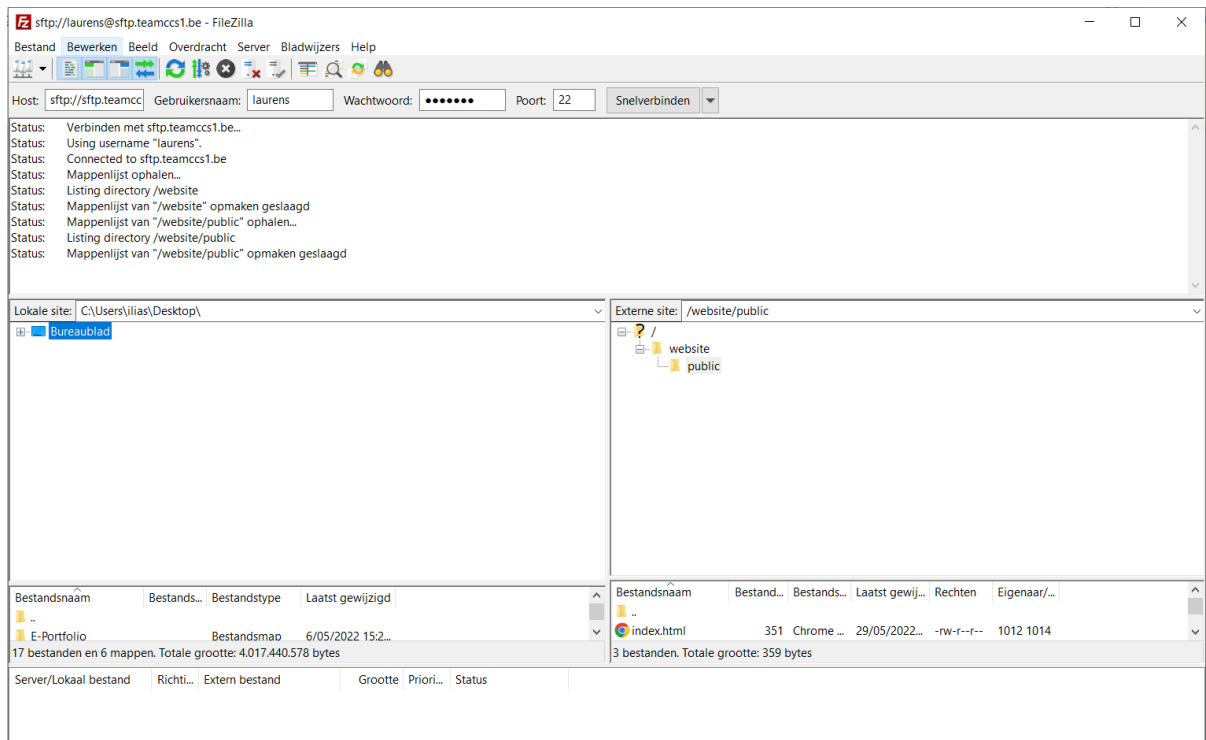
3.3.3 Hoe maak ik een database aan?

Dit gebeurt automatisch, je kan de database bereiken door te surfen naar `db.domeinnaam.teamccs1.be`.

3.3.4 Hoe kan ik files uploaden?

Maak gebruik van een SFTP-client software zoals bv. FileZilla.

Login met uw gegevens: Host = `sftp://sftp.teamccs1.be`, gebruikersnaam = uw eerder gekozen gebruikersnaam, wachtwoord = uw eerder gekozen wachtwoord.



Plaats nu u bestanden binnen de map website.

3.3.5 Hoe kan ik contact opnemen met support?

Klik vanboven op het vakje 'Contact', en vul verder het hulp formulier in.

U ontvangt geen bevestigingsmail maar uw hulp formulier wordt wel verstuurd.

Wij helpen u zo spoedig mogelijk verder.

4 OPS REPORT CARDS

In dit hoofdstuk staan de Ops report cards die we gekozen en geïmplementeerd. Via deze link staan alle Ops report cards: <https://www.opsreportcard.com/>

4.1 Gekozen Ops Report Cards

A. Public Facing Practices

2. Are "the 3 empowering policies" defined and published?

Met deze Ops report cards hebben ze het over de volgende drie dingen:

- Hoe kunnen de gebruikers ons contacteren?
- Wat is een ernstig probleem?
- Hoe gaan wij de gebruikers helpen als ze een probleem hebben?

3. Does the team record monthly metrics?

- We gaan op regelmatige basisgegevens verzamelen over ons systeem om de gebruikerservaring te verbeteren

B. Modern Team Practices

5. Do you have a password safe?

- Een veilige plaats waar we onze wachtwoorden opslaan

C. Operational Practices

11. Does each service have an OpsDoc?

- We gaan procedures maken voor elke service die we aanbieden en elke handeling die we moeten doen

12. Does each service have appropriate monitoring?

- Over elk toestel gaan we een monitoring service bouwen om snel te kunnen reageren als er iets misgaat

D. Automation Practices

16. Do you use configuration management tools like cfengine/puppet/chef?

- Alle verschillende virtuele machines gaan we beheren met puppet. Zodat we de configuratie van alle virtuele machines kunnen beheren vanop 1 VM.

E. Fleet Management Processes

21. Can you automatically patch software across your entire fleet?

- Via crontab word er om de 5 dagen geüpdate naar de laatste versie van de software. Dit gebeurt op elke machine.

F. Disaster Preparation Practices

23. Can your servers keep operating even if 1 disk dies?

- Om een goede beschikbaarheid te voorzien gaan we load balancing en redundantie toepassen tussen verschillende docker containers in een webserver en tussen 2 webserver. Dus elke website verdelen we over 2 docker containers, dus er mag 1 docker container stuk gaan en dan kan de klant nog steeds aan zijn/haar website.

25. Are your backups automated?

- Om ervoor te zorgen dat er geen data verloren gaat gaan we op regelmatige tijdstippen automatische back-ups nemen

27. Do machines in your data center have remote power / console access?

- Om te verbinden met onze vm's gebruiken we een VPN-verbinding

G. Security Practices

31. Can a user's account be disabled on all systems in 1 hour?

- Alle gebruikers gaan we opslaan in 1 database, dus als er een gebruiker gedeactiveerd moet worden kan dit snel gebeuren

32. Can you change all privileged (root) passwords in 1 hour?

- Om alle wachtwoorden binnen een uur te kunnen veranderen gaan we een bestand creëren waar alle plaatsen staan waar er een wachtwoord

We hebben gekozen voor deze OPS-report cards omdat deze de meeste waarde toevoegen aan het project en het minste tijd in beslag nemen om deze op een efficiënte manier te implementeren.

4.2 Geïmplementeerde Ops Report Cards

A. Public Facing Practices

2. Are "the 3 empowering policies" defined and published?

De 3 empowering policies staan uitgelegd in het punt 4.3.

3. Does the team record monthly metrics?

Maandelijks rapporten zijn ingesteld op basis van ons hoofd dashboard, hier staan onze favoriete statistieken.

Scheduled reports Create report

| | | | | | | | | |
|---|------------------------------|---------------|----------------|-----------|---------|----------|---------|-------------------------|
| Name | Show | Created by me | Status | Any | Enabled | Disabled | Expired | Filter |
| | | | | | | | | |
| <input type="checkbox"/> Name | Owner | Repeats | Period | Last sent | Status | Info | | |
| <input type="checkbox"/> CCST Monthly Recap | Admin (Zabbix Administrator) | Monthly | Previous month | Never | Enabled | | | Displaying 1 of 1 found |

B. Modern Team Practices

5. Do you have a password safe?

We hebben een password safe geconfigureerd op de Puppetmaster. Deze password safe is beveiligd met een sterk wachtwoord. Deze kluis bevat alle logingegevens voor de systeem accounts. Met deze accounts bedoelen we accounts die gebruikt worden voor onderhoud.

```
ccsl@lm-puppetmaster:~$ sudo keepassxc-cli ls password
Insert password to unlock password:
DNSmaster
DNSslave
DockerHost
DRP
PuppetMaster
SFTP
ZabbixServer
ccsl@lm-puppetmaster:~$
```

<https://www.mankier.com/1/keepassxc-cli>

De logingegevens van onze gebruikers worden veilig opgeslagen in de gebruikers database.

C. Operational Practices



11. Does each service have an OpsDoc?

De OpsDoc zijn uitgeschreven bij punt 4.3.2.

12. Does each service have appropriate monitoring?

We hebben ervoor gekozen om niet alle services, maar de belangrijkste te monitoren. Voor ons zijn dat:

- Zabbix-agent2.service
- Ntp.service
- Puppet.service
- Rsync.service
- Ssh.service
- Haproxy.service
- Cron.service
- Ufw.service
- Docker.service
- Fail2ban.service

| Services   | | |
|--|-------------------------------------|--------------|
| Timestamp | Name | Value |
| 2022-05-30 13:26:22 | rsync.service: Active state | inactive (3) |
| 2022-05-30 13:26:22 | fail2ban.service: Active state | active (1) |
| 2022-05-30 13:26:22 | ssh.service: Active state | active (1) |
| 2022-05-30 13:26:22 | docker.service: Active state | active (1) |
| 2022-05-30 13:02:37 | ufw.service: Active state | active (1) |
| 2022-05-30 12:59:29 | puppet.service: Active state | active (1) |
| 2022-05-30 12:59:07 | haproxy.service: Active state | active (1) |
| 2022-05-30 12:58:34 | ntp.service: Active state | active (1) |
| 2022-05-30 12:57:00 | zabbix-agent2.service: Active state | active (1) |
| 2022-05-30 12:55:59 | ssh.service: Active state | active (1) |

Services Up-time



| Timestamp | Name | Value |
|---------------------|------------------------------------|-------------|
| 2022-05-30 13:34:32 | ntp.service: Active time | 5d 1h 58m |
| 2022-05-30 13:34:12 | haproxy.service: Active time | 5d 1h 57m |
| 2022-05-30 13:33:37 | ufw.service: Active time | 5d 1h 57m |
| 2022-05-30 13:32:42 | cron.service: Active time | 5d 1h 56m |
| 2022-05-30 13:32:22 | ssh.service: Active time | 3h 23m 30s |
| 2022-05-30 13:32:22 | fail2ban.service: Active time | 12h 47m 34s |
| 2022-05-30 13:32:22 | docker.service: Active time | 5d 1h 54m |
| 2022-05-30 13:31:45 | puppet.service: Active time | 3d 21h |
| 2022-05-30 13:30:32 | ntp.service: Active time | 5d 1h 54m |
| 2022-05-30 13:29:57 | zabbix-agent2.service: Active time | 1h 41m 55s |

Dat wil niet zeggen dat we de andere services niet kunnen monitoren. We hebben specifiek deze services in onze monitoring dashboard gezet.

D. Automation Practices

16. Do you use configuration management tools like cfengine/puppet/chef?

Ons project wordt geautomatiseerd door gebruik van Puppet, alle machines en configuratie van de machines worden hierdoor aangestuurd.

Op de Puppet master staat een manifest die de juiste configuratie doorstuurt naar alle Puppet clients.

E. Fleet Management Processes

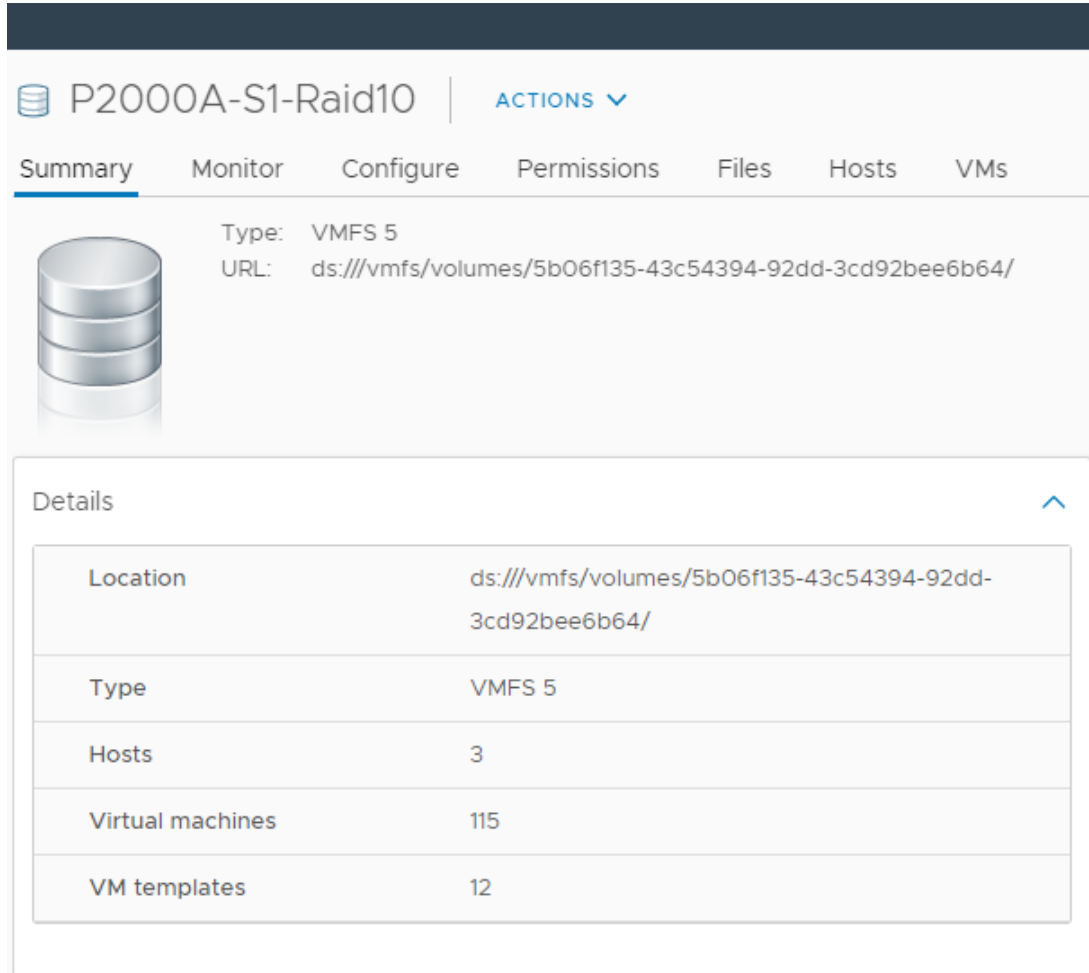
21. Can you automatically patch software across your entire fleet?

De software op de machines wordt automatisch geüpdate om de 5 dagen. Via crontab op elke machine worden deze commando's uitgevoerd.

F. Disaster Preparation Practices

23. Can your servers keep operating even if 1 disk dies?

Al onze virtuele machines staan opgeslagen op een opslag die gebruik maakt van RAID10.



P2000A-S1-Raid10 | ACTIONS ▾

Summary | Monitor | Configure | Permissions | Files | Hosts | VMs

Type: VMFS 5
URL: ds:///vmfs/volumes/5b06f135-43c54394-92dd-3cd92bee6b64/

Details

| | |
|------------------|---|
| Location | ds:///vmfs/volumes/5b06f135-43c54394-92dd-3cd92bee6b64/ |
| Type | VMFS 5 |
| Hosts | 3 |
| Virtual machines | 115 |
| VM templates | 12 |

Door gebruik te maken van deze opslag kunnen er meerdere harde schijven kapotgaan zonder dat we onze data kwijt zijn.

25. Are your backups automated?

De back-ups zijn geautomatiseerd via een Rclone command in crontab. Crontab voert elke uur het Rclone commando uit om van alle belangrijke documenten een back-up te nemen. Dit gebeurt op al onze systemen zodat we de bestanden van de gebruikers en onze eigen bestanden opslaan.

Mijn bestanden > ProjectHosting > Backup

| Naam | Gewijzigd | Gewijzigd door | Bestandsgrootte | Delen |
|--------|--------------------------|------------------|-----------------|-------|
| Client | Ongeveer een uur gele... | Siebe Van Rompay | 12 items | Privé |
| LBW | 5 minuten geleden | Siebe Van Rompay | 2 items | Privé |
| Master | Ongeveer een uur gele... | Siebe Van Rompay | 4 items | Privé |

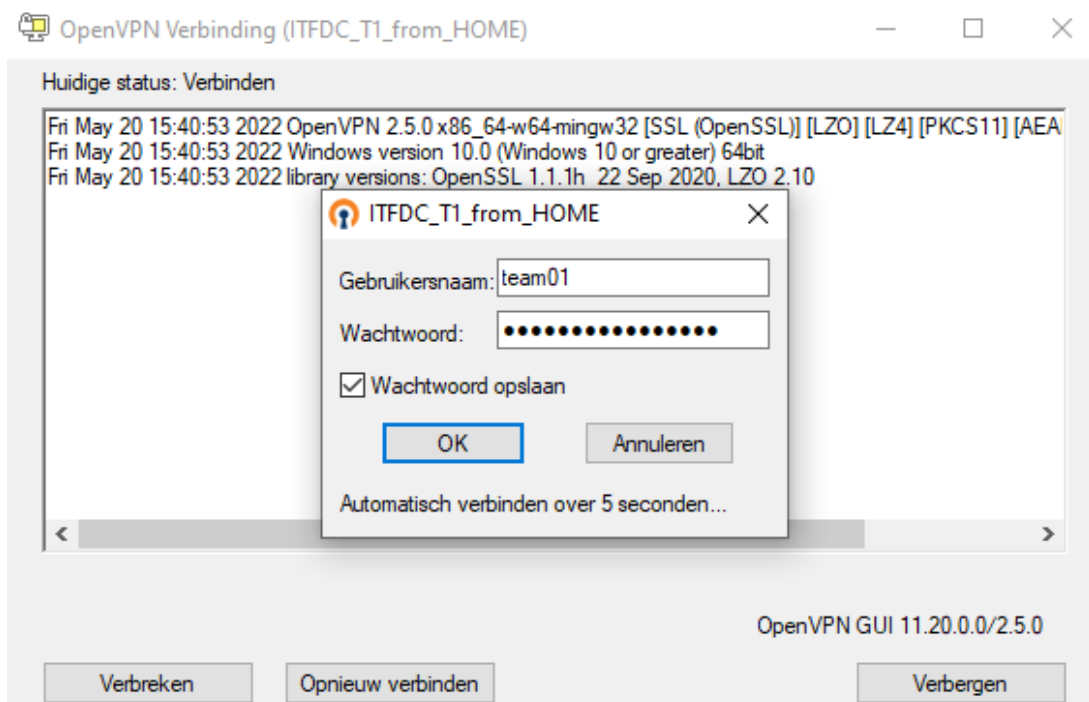
```
GNU nano 4.8 /tmp/crontab.mDqIW0/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 * * * * rclone copy /etc/puppetlabs/code/environments/production/ Backup:/ProjectHosting/Backup/Master >/dev/null 2>&1
```

```
0 * * * * rclone copy /home/ Backup:/ProjectHosting/Backup/Client >/dev/null 2>&1
```

```
0 * * * * rclone copy /home/ccs1/backups Backup:/ProjectHosting/Backup/LBW/backups >/dev/null 2>&1
0 * * * * rclone copy /home/ccs1/webconfig/ Backup:/ProjectHosting/Backup/LBW/webconfig >/dev/null 2>&1
```

27. Do machines in your data center have remote power / console access?

We hebben geen toegang tot het fysieke datacenter waar al de servers staan dus werken we met een VPN-verbinding. Deze VPN-verbinding geeft ons toegang tot het datacenter en zorgt ervoor dat enkel mensen met de juiste VPN sleutel toegang hebben tot het netwerk waarin onze vms staan.



Vervolgens maken we gebruik van een SSH verbinding vanuit onze pc's naar onze vm's om onderhoud te kunnen doen op de vm's.

```

login as: ccs1
ccs1@172.26.1.1's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-109-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri 20 May 2022 03:39:30 PM CEST

System load:  0.0               Processes:            231
Usage of /:   25.3% of 30.63GB   Users logged in:     1
Memory usage: 12%              IPv4 address for ens160: 172.26.1.1
Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

31 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

*** System restart required ***
Last login: Fri May 20 14:45:18 2022 from 192.168.213.2
ccs1@dnsserver1:~$

```

G. Security Practices

31. Can a user's account be disabled on all systems in 1 hour?

Dit is mogelijk. In de database heeft elke user een 'user_enabled'-waarde. Indien deze op 0 wordt gezet, zal deze wijziging binnen de minuut door een Python-script worden geïnterpreteerd, waarna aan de puppet-master wordt doorgegeven dat deze gebruiker wordt uitgeschakeld.

```

$dis_file = file("/home/ccs1/sshfs_userfiles/dis_users.txt").chomp
$dis_users = split($dis_file, '[,]')

$dis_users.each |String $dis_user| {
    user { "$dis_user":
        shell => '/bin/false',
        password => Sensitive("!"),
    }
}

```

32. Can you change all privileged (root) passwords in 1 hour?

Ook dit is mogelijk. Hiervoor hebben we een script ontworpen, dat zich op de DockerHost bevindt. Met dit script maak je automatisch een ssh-verbinding naar alle nodige virtuele machines, waarbij je een voor een bij elke machine automatisch wordt gevraagd om het paswoord te wijzigen. Je kan er zowel voor kiezen om elke vm hetzelfde paswoord te geven, of juist elke machine een verschillend paswoord toe te kennen.

4.3 Documenten voor Ops Report Cards

4.3.1 De 3 empowering policies

Hoe kunnen gebruikers ons aanspreken als ze met een probleem zitten?

Het is de bedoeling dat de gebruikers contact opnemen via het e-mailadres support@teamccs1.be. Deze mail zal beantwoord binnen de 7 werkdagen. Met een antwoord bedoelen wij het bevestigen dat de email van de gebruiker is aangekomen en dat er aan een antwoordt gewerkt wordt.

Wij (teamccs1) kunnen niet garanderen dat het probleem of vraag van de gebruiker binnen enige tijd beantwoord wordt. Wij doen ons best om de gebruiker zo snel mogelijk verder te helpen maar wij kunnen geen oplossing garanderen.

Als het probleem/vraag niet via mail beantwoord kan worden dan bieden wij ook ondersteuning via videocall. Dit gebeurt enkel en alleen in overeenstemming met de gebruiker en een medewerker van teamccs1, op een tijdstip dat ze beide beschikbaar zijn.

Wat is een noodgeval?

Een noodgeval is een probleem dat bij ons geprioriteerd kan worden volgens 3 kleurcodes.

- code rood: Wanneer er meer dan 5 gebruikers geen toegang meer hebben tot onze login website of tot hun gemaakte website die bij ons gehost wordt door hetzelfde probleem.
- code oranje: Wanneer er meer dan 2 gebruikers geen toegang meer hebben tot onze login website of tot hun gemaakte website die bij ons gehost wordt door hetzelfde probleem.
- code geel: 1 gebruiker geen toegang meer heeft tot onze login website of tot hun gemaakte website die bij ons gehost wordt door hetzelfde probleem.

Hoe en wanneer bieden we ondersteuning?

We bieden email ondersteuning van maandag tot vrijdag van 9:00 tot 17:00. Elke email wordt binnen de 7 werkdagen bevestigd aan de gebruiker dat de email goed ontvangen is en dat er gezocht wordt naar een oplossing.

Als het probleem niet via email kan opgelost worden dan is er videocall ondersteuning beschikbaar elke weekdag van maandag tot vrijdag van 10:00 tot 16:00. Of er een videocall nodig is of niet wordt volledig bepaald door onze medewerkers. Wij garanderen geen maximumtijd waarin een probleem opgelost moet zijn.

Wij garanderen geen oplossing voor elk probleem. We doen ons best om de problemen van de gebruikers zo snel mogelijk op te lossen.

4.3.2 OpsDoc

4.3.2.1 DNS servers

- Overzicht

In ons system zijn er twee DNS servers aanwezig omdat de configuratie server werkt met host names en niet met IP adressen. Dus de configuratieserver moet een hostname kunnen resolvable naar een IP adres.

- Configuratie

Hieronder staan alle commando's om de DNS servers vanaf een standaard linux machine op te bouwen.

DNS server 1

```
sudo hostnamectl set-hostname DNSmaster1
sudo nano /etc/netplan/00-installer-config.yaml
pas ip adres aan naar 172.26.1.1
sudo apt-get update -y
sudo apt-get upgrade -y
sudo apt-get install bind9 -y
sudo apt-get install dnsutils -y
sudo nano /etc/systemd/resolved.conf
voeg lijn toe
DNS=172.26.1.1
sudo nano /etc/bind/named.conf.local
voeg code toe
zone "puppet" {

    notify yes;
    allow-transfer { 172.26.1.2; };
    allow-update {none;};
    type master;
    file "/etc/bind/puppet";
};
```

```
Sudo nano /etc/bind/puppet
; BIND data file for local loopback interface
;
$TTL 604800
@ IN SOA DNSserver.puppet. root.puppet. (
        3 ; Serial
        604800 ; Refresh
        86400 ; Retry
        2419200 ; Expire
        604800 ) ; Negative Cache TTL
;
@ IN NS DNSserver.puppet.

DNSserver IN A 172.26.1.1
lm-puppetmaster IN A 172.26.1.11
lm-puppetclient IN A 172.26.1.12
webserver IN A 172.26.1.3
```

Deze configuratie gaat eerst de naam van de machine veranderen naar DNSmaster1. Daarna wordt het IP adres statisch ingesteld, dan wordt de laatste versie van bind 9 geïnstalleerd. Vervolgens zorgt de aanpassing in de resolv.conf ervoor dat alle DNS queries naar zichzelf worden gestuurd. Dan wordt er een zone bestand gemaakt om alle hostnames in ons netwerk te linken aan IP adressen.

DNS slave 1

DNS server 1

```
sudo hostnamectl set-hostname DNSslave1
sudo nano /etc/netplan/00-installer-config.yaml
pas ip adres aan naar 172.26.1.2
sudo apt-get update -y
sudo apt-get upgrade -y
sudo apt-get install bind9 -y
sudo apt-get install dnsutils -y
sudo nano /etc/systemd/resolved.conf
voeg lijn toe
DNS=172.26.1.2
sudo nano /etc/bind/named.conf.local
voeg code toe
zone "puppet" {
    type slave;
    file "auto_puppet";
    masters { 172.26.1.1; };
};
```

Deze configuratie is bijna hetzelfde als de DNS server 1. Alleen staat er hier in de named.conf.local dat het om een "slave" gaat en niet om een master. Dan wil zeggen dat deze server alle zone bestanden van de DNS server 1 gaat kopiëren.

Vervolgens voeren we deze 2 commando's uit op beide servers om de configuratie te herladen en dan is de configuratie van de DNS servers klaar.

```
sudo systemctl restart bind9.service
sudo service systemd-resolved restart
```

- Disaster recovery
Wanneer er een van de twee DNS servers stuk gaat worden alle DNS verzoeken automatisch naar de andere DNS server verstuurd en blijft het systeem werken. Wanneer ze beide stuk gaan dan moeten ze opnieuw opgebouwd worden met bovenstaande configuratie.

4.3.2.2 SFTP server

```
node 'lm-puppetclient.puppet' {
  class { 'ssh::server':
    options => {
      'Match Group sftp_users' => {
        'ChrootDirectory' => '/home/%u',
        'ForceCommand' => 'internal-sftp -d /website',
        'PasswordAuthentication' => 'yes',
        'AllowTcpForwarding' => 'no',
```

```

        'X11Forwarding' => 'no',
      },
      'PermitRootLogin'      => 'no',
      'Port'                 => [22],
      'Subsystem'            => 'sftp internal-sftp',
      'Banner'               => '/home/ccs1/scripts/banner.txt',
    },
  },
}
$file = file("/home/ccs1/sshfs_userfiles/users.txt").chomp
$users = split($file, '[,]')

group { 'sftp_users':
  ensure => 'present',
}

$users.each |String $user| {
  $namepass = split($user, '[:]')

  user { "${namepass[0]}":
    groups => 'sftp_users',
    shell => '/usr/sbin/nologin',
    password => Sensitive("${namepass[1]}"),
  }
  file { "/home/${namepass[0]}":
    ensure => 'directory',
    owner => 'root',
    group => "${namepass[0]}",
    mode => '0755',
  }
  file { "/home/${namepass[0]}/website":
    ensure => 'directory',
    owner => "${namepass[0]}",
    group => 'sftp_users',
    mode => '0755',
  }
  file { "/home/${namepass[0]}/website/public":
    ensure => 'directory',
    owner => "${namepass[0]}",
    group => "${namepass[0]}",
    mode => '0755',
  }
}

$dis_file = file("/home/ccs1/sshfs_userfiles/dis_users.txt").chomp
$dis_users = split($dis_file, '[,]')

$dis_users.each |String $dis_user| {
  user { "$dis_user":
    shell => '/bin/false',
    password => Sensitive("!"),
  }
}
}

```

Met bovenstaande code wordt de sftp-server, die onderdeel is van de SSH-daemon, geprovisioneerd vanuit de manifest file die zich op de puppemaster bevindt. Deze configuratie wordt continu gemonitord op wijzigingen, en worden bijgevolg meteen bijgewerkt.

De `'class::ssh-server'` spreekt de `sshd_config` op het apparaat aan. Hierin geven we aan dat voor de groep `'sftp_users'` het volgende moet gelden:

- De root-map wordt gewijzigd naar `/home/<username>`
- Bij aanmelding op de server met bijvoorbeeld FileZilla, moet automatisch naar de submap `'website'` worden gegaan
- Aanmelden met een paswoord is mogelijk, maar portforwarding, of forwarding van een GUI zijn geblokkeerd

Verder zijn er nog enkele algemene opties meegegeven:

- Root kan NOOIT rechtstreeks aanmelden
- Er wordt gebruik gemaakt van port 22 voor SSH-verkeer
- Sftp draait op de `'non-default'` internal-sftp module
- Bij het aanmelden moet de banner worden getoond die zich op het opgegeven pad bevindt

Hierna worden de actieve users opgehaald door een Pythonscript en in een tekstbestand gezet, samen met het gehashte paswoord.

Puppet kan deze strings automatisch omzetten naar arrays die eenvoudig kunnen worden gesplitst en verder gebruikt in de komende configuratie.

Voordat de users en hun mappen kunnen worden toegevoegd, vergewissen we er ons van dat de groep `'sftp_users'` wordt aangemaakt, zodat de configuratie die hierboven staat, daarnaar kan refereren.

Tot slot worden dan de user, met bijbehorend paswoord, aangemaakt. Let hierbij op de shell, die ervoor zorgt dat deze users onmogelijk naar een CLI kunnen aanmelden.

Hierna worden de onderliggende mappen aangemaakt.

Op het allerlaatste kijken we nog na welke users als `'disabled'` in de database staan. Van deze users wijzigen we de shell, naar een die ook sftp-aanmeldingen onmogelijk maakt. Daarbovenop blokkeren we het account, door het paswoord naar `'!'` te wijzigen.

4.3.2.3 Webserver

- Overzicht:
De webserver is het hart van ons systeem. Op deze server staat de gebruikerswebsite waarmee alle gebruikers websites kunnen aanmaken. De websites en databases van alle gebruikers staan ook op deze server.
- Configuratie
In deze server zit een groot deel configuratie werk daarom gaan we deze configuratie beknopt uitleggen.

We veranderen eerst de hostname en IP adres naar Webserver en 172.26.1.3. Vervolgens installeren we Docker en Docker Compose. Dan maken we een basis Dockerfile, Docker Compose, Haproxy config files voor de Haproxy loadbalancer.

```
GNU nano 4.8
FROM haproxy:1.7
```

Dockerfile

```
GNU nano 4.8
version: "3"

services:
  hpl:
    build: .
    container_name: hpl
    volumes:
      - ./:/usr/local/etc/haproxy/
    ports:
      - "80:80"
      - "8404:8404"
      - "443:443"
  networks:
    default:
      external:
        name: external-example
```

Docker Compose file

```
GNU nano 4.8
global
defaults
  mode http
  timeout connect 5000ms
  timeout client 50000ms
  timeout server 50000ms

frontend stats
  bind *:8404
  stats enable
  stats uri /
  stats refresh 10s

frontend myfrontend
  bind :80

  acl ACL_website1 hdr(host) -i website1.teamccsl.be www.website1.teamccsl.be
  use_backend website1cluster if ACL_website1

  acl ACL_website2 hdr(host) -i website2.teamccsl.be www.website2.teamccsl.be
  use_backend website2cluster if ACL_website2

backend website1cluster

  server w1c1 w1c1:80 check
  server w1c2 w1c2:80 check

backend website2cluster

  server w2c1 w2c1:80 check
  server w2c2 w2c2:80 check
```

Haproxy config file

Deze configuratie gaat een Haproxy container maken die zowel als DNS server gaat handelen voor de verschillende websites van de verschillende

gebruikers en ook de verzoeken voor deze websites gaat verdelen over 2 web containers per website.

- Disaster Recovery:
Zou er een web container van een gebruiker stuk gaan dan is dit geen ramp want dan is er nog een web container die dan overneemt. Als beide web containers stuk gaan dan zien we dit op onze monitoring en dan maken we nieuwe web containers aan.

Als er een database van een gebruiker stuk gaat. Dan zien we dit ook op onze monitoring en dan maken we zo snel mogelijk een nieuwe database aan voor die gebruiker en dan zetten we zijn data terug.

Moest de Haproxy container stuk gaan dan zien we dit op de monitoring en dan activeren we de back-up Haproxy container. Als de hele webserver stuk gaat dan moeten we deze volledig terug opnieuw configureren met de data die geback-up staan in de Cloud.

- SLA: Service Level Agreement
Wij garanderen geen percentage uptime omdat het datacenter niet door ons beheerd wordt en we dus ook geen controle hebben over de fysieke machines. Moest er zich een ramp voordoen dan doen we er alles aan om al onze websites zo snel mogelijk terug online te krijgen.

4.3.2.4 Zabbix server

- Overzicht

De Zabbix server geeft ons een duidelijke samenvatting van onze belangrijkste servers en services. Onze Zabbix dashboard is zodanig ingesteld zodat we ook een samenvatting kunnen zien van de problemen die we hebben op onze servers. Aan de hand van deze statistieken kunnen we op verschillende plaatsen ingrijpen waar nodig is.

- Configuratie

Zodra de server up en running is, is het de bedoeling dat we Zabbix Agents gaan toevoegen aan de verschillende VMs die we willen monitoren. Nadat er agents zijn toegevoegd, moet de config file `/etc/zabbix/zabbix-agent2.config` aangepast worden.

3 punten moeten aangepast worden:

1. Server (Ip van Zabbix server)

```
Server=172.26.1.6
```

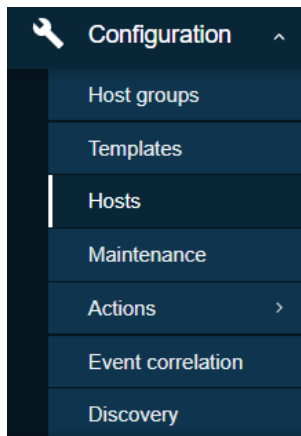
2. Server-Active (Ip van Zabbix server)

```
ServerActive=172.26.1.6
```

3. Hostname (Dit mag je zelf kiezen, deze gaan we later nodig hebben)

```
Hostname=DockerHost
```

Vergeet niet de Zabbix service te herstarten (`sudo systemctl restart zabbix-agent2.service`). Daarna moeten we de VM waar de host op staat nog toevoegen in de webinterface.



Hier moeten we dezelfde hostname meegeven die we hebben gekozen in de Zabbix Agent config file. Group en Ip zijn verplicht in te vullen. De Group kan je zelf kiezen. De Ip is je VM server waar je Zabbix Agent2 op is geconfigureerd.

New host ✕

Host IPMI Tags Macros Inventory Encryption Value mapping

* Host name

Visible name

Templates

* Groups

| Interfaces | Type | IP address | DNS name | Connect to | Port | Default |
|------------|------|------------|----------------------|---|-------|---|
| Agent | | 127.0.0.1 | <input type="text"/> | <input checked="" type="radio"/> IP <input type="radio"/> DNS | 10050 | <input checked="" type="radio"/> Remove |

[Add](#)

Description

Monitored by proxy

Enabled ☒

4.3.2.5 Back-ups

- Overzicht

De back-ups zijn geautomatiseerd om alle belangrijke bestanden elk uur opnieuw op te slaan. Dit geldt zowel voor alle bestanden van de klant als voor de belangrijke bestanden voor onze eigen machines. Hierdoor kunnen we indien onze systemen plat zouden liggen snel alles terug zetten naar de laatste versie.

- Configuratie

Voor het nemen van deze back-ups hebben we Rclone gebruikt. Deze hebben we ingesteld om bestanden te kopiëren naar OneDrive. Via crontab zorgen we ervoor dat het commando om bestanden te kopiëren naar OneDrive elk uur word uitgevoerd.

Via 'rclone config' hebben we een remote connection aan gemaakt met de naam 'Backup' die gelinkt is naar onze OneDrive.

```
ccs1@lm-puppetmaster:~$ rclone config
Current remotes:

Name                Type
====                ==
Backup              onedrive

e) Edit existing remote
n) New remote
d) Delete remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
e/n/d/r/c/s/q>
```

Het commando voor het kopiëren van de bestanden naar OneDrive word elk uur uitgevoerd via crontab.

```
GNU nano 4.8 /tmp/crontab.uxGM18/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 * * * * rclone copy /etc/puppetlabs/code/environments/production/ Backup:/ProjectHosting/Backup/Master >/dev/null 2>&1

# m h dom mon dow   command
0 * * * * rclone copy /home/ Backup:/ProjectHosting/Backup/Client >/dev/null 2>&1

# m h dom mon dow   command
0 * * * * rclone copy /home/ccs1/backups Backup:/ProjectHosting/Backup/LBW/backups >/dev/null 2>&1
0 * * * * rclone copy /home/ccs1/webconfig/ Backup:/ProjectHosting/Backup/LBW/webconfig >/dev/null 2>&1
```

De bestanden komen dan in de OneDrive.

Mijn bestanden > ProjectHosting > Backup

| Naam | Gewijzigd | Gewijzigd door | Bestandsgrootte | Delen |
|--------|--------------------------|------------------|-----------------|-------|
| Client | Ongeveer een uur gele... | Siebe Van Rompay | 12 items | Privé |
| LBW | 5 minuten geleden | Siebe Van Rompay | 2 items | Privé |
| Master | Ongeveer een uur gele... | Siebe Van Rompay | 4 items | Privé |

- Disaster Recovery

De bestanden worden opgeslagen naar de Cloud en dus niet naar onze eigen systemen. Hierdoor kunnen we altijd aan onze bestanden als onze eigen systemen platliggen. Enkel als de Cloud waar onze bestanden opstaan down gaat kunnen we niet aan onze back-ups.

- SLA: Service Level Agreement

Wij kunnen niet garanderen dat alle bestanden die geüpload worden altijd geback-upt zijn. Enkel als de bestanden er langer dan 1 uur opstaan zullen deze opgeslagen worden.

4.3.2.6 Configuratie server (Puppetmaster)

```
ccsi@lm-puppetmaster:/etc/puppetlabs/code/environments/production/manifests$ sudo /opt/puppetlabs/bin/puppetserver ca list --all
Signed Certificates:
  lm-puppetclient.puppet      (SHA256)  2B:74:67:8D:B2:25:25:00:41:82:0C:86:5C:94:61:F2:E9:19:A2:53:2F:4F:A4:
03:54:63:16:56:34:5A:D7:55    alt names: ["DNS:lm-puppetclient.puppet"]
  test.puppet                 (SHA256)  9A:9C:D3:09:D8:ED:D3:35:CC:F6:AC:5D:14:9E:A8:86:F2:06:D4:4E:BB:6A:B3:
A2:0F:19:DF:D9:7A:BD:CB:83    alt names: ["DNS:test.puppet"]
  webserver.puppet            (SHA256)  92:12:4E:35:A0:05:B3:C2:3B:0F:CD:B5:D4:B9:4D:59:1C:88:C3:85:BB:0F:F4:
5C:A6:E9:2A:32:E0:52:BA:D4    alt names: ["DNS:webserver.puppet"]
  lm-puppetmaster.puppet      (SHA256)  EA:E4:2C:0A:04:B8:0F:BA:EC:5F:E8:EB:AB:23:73:4E:0B:68:20:6A:1A:52:CE:
22:82:89:35:F0:EE:A5:DB:B6    alt names: ["DNS:puppet", "DNS:lm-puppetmaster.puppet"] authorization extensions
: [pp_cli_auth: true]
```

5 OPTIONEEL

In dit hoofdstuk staan alle testen die we hebben uitgevoerd. Daarna hebben we een aantal dingen besproken die we willen verbeteren naar volgende versies. Tot slot staat er nog overige documentatie.

5.1 Andere documentatie

5.1.1 SLA

| | |
|--------------------------|------------|
| Documenteigenaar: | Teamccs1 |
| Ingangsdatum: | 10/04/2022 |

Versies

| Versie | Datum | Omschrijving | Auteur |
|--------|------------|-------------------------|--------------|
| 1.0 | 10/04/2022 | Service Level Agreement | Brent Druyts |
| 1.1 | 21/04/2022 | Bedrijfsnaam aangepast | Brent Druyts |

Goedkeuring

(Door gebruik te maken van onze diensten, stemmen alle goedkeurders in met alle voorwaarden en bepalingen die in deze overeenkomst worden beschreven.)

| Goedkeurders | Rol | Ingangsdatum |
|--------------|---|--------------|
| Bedrijf | Teamccs1 | 10/04/2022 |
| Klant | Alle klanten die gebruik maken van de diensten van Teamccs1 | 10/04/2022 |

Overzicht

Deze overeenkomst vertegenwoordigt een Service Level Agreement ("SLA") tussen Teamccs1 en alle klanten die van de diensten van Teamccs1 gebruikmaken voor het leveren van IT-diensten die nodig zijn om de websites van alle klanten te hosten.

Deze overeenkomst blijft geldig totdat deze wordt vervangen door een herziene overeenkomst. Deze overeenkomst schetst de parameters van alle IT-services zoals deze worden begrepen door de belanghebbenden. Deze overeenkomst vervangt de huidige processen en procedures niet, tenzij dit hierin uitdrukkelijk wordt vermeld.

Doelen

Het doel van deze overeenkomst is ervoor te zorgen dat de juiste elementen en toezeggingen aanwezig zijn om consistente IT-serviceondersteuning en levering aan alle klanten van Teamccs1 te bieden.

Het doel van deze overeenkomst is het verkrijgen van wederzijdse overeenstemming voor IT-dienstverlening tussen Teamccs1 en al hun klanten.

De doelstellingen van deze overeenkomst zijn:

- Zorg voor een duidelijke verwijzing naar eigendom, verantwoordelijkheid, rollen en/of verantwoordelijkheden van de dienst
- Presenteer een duidelijke, beknopte en meetbare beschrijving van de dienstverlening aan de klant
- Duidelijk aangeven wanneer en in welke vorm er ondersteuning mogelijk is

Stakeholders

De volgende dienstverlener(s) en klant(en) zullen worden gebruikt als basis van de overeenkomst en vertegenwoordigen de belanghebbenden die bij deze SLA zijn betrokken:

IT-dienstverlener(s): Teamccs1 ("Aanbieder")

IT-klant(en): Alle mensen die gebruik maken van de diensten van Teamccs1, 2^{de} en 3^{de} jaar studenten van APP/AI van de Thomas More Geel

Periodieke herziening

Deze overeenkomst is geldig vanaf de ingangsdatum die hierin wordt beschreven en is geldig tot nader order. Deze overeenkomst moet minimaal eenmaal per jaar worden herzien. In plaats van een beoordeling gedurende een bepaalde periode blijft de huidige overeenkomst echter van kracht.

Teamccs1 is verantwoordelijk voor het faciliteren van regelmatige beoordelingen van dit document. De inhoud van dit document kan worden gewijzigd, op voorwaarde dat de belanghebbenden in onderling overleg zijn overeengekomen en aan alle betrokken partijen worden gecommuniceerd. De documenteigenaar zal alle daaropvolgende revisies opnemen en indien nodig wederzijdse overeenkomsten / goedkeuringen verkrijgen.

Zakelijke relatiemanager: Teamccs1
 Beoordelingsperiode: halfjaarlijks (6 maanden)
 Vorige beoordelingsdatum: 10/04/2022
 Volgende beoordelingsdatum: 10/10/2022

Serviceovereenkomst

De volgende gedetailleerde serviceparameters vallen onder de verantwoordelijkheid van de serviceprovider bij de voortdurende ondersteuning van deze overeenkomst.

Servicebereik

De volgende diensten vallen onder deze overeenkomst:

- Gecontroleerde e-mailondersteuning
- Halfjaarlijkse systeemgezondheidscontrole

Klantvereisten

De verantwoordelijkheden en/of vereisten van de klant ter ondersteuning van deze overeenkomst omvatten:

- Betaling van alle supportkosten op het afgesproken interval
- Redelijke beschikbaarheid van klantvertegenwoordiger(s) bij het oplossen van een service gerelateerd incident of verzoek
- Zich houden aan de voorwaarden die in dit document staan opgesteld

Vereisten voor serviceproviders

De verantwoordelijkheden en/of vereisten van de dienstverlener ter ondersteuning van deze overeenkomst omvatten:

- Responstijden voor vergaderingen in verband met service gerelateerde incidenten
- Passende melding aan alle klanten voor al het geplande onderhoud
- Responstijden voor het oplossen van problemen

Service veronderstellingen

Aannames met betrekking tot in-scope services en/of componenten zijn onder meer:

- Wijzigingen in diensten zullen worden gecommuniceerd en gedocumenteerd aan alle belanghebbenden
- Moest er een langdurig defect zijn dan worden alle klanten op de hoogte gebracht

Service Management

Effectieve ondersteuning van in-scope services is het resultaat van het handhaven van consistente serviceniveaus. De volgende secties bevatten relevante details over de beschikbaarheid van services, monitoring van in-scope services en gerelateerde componenten.

Beschikbaarheid van de dienst

Dekkingsparameters die specifiek zijn voor de dienst(en) die onder deze overeenkomst vallen, zijn als volgt:

- E-mailondersteuning: bewaakt om 09:00 uur tot 17.00 uur Maandag tot vrijdag
 - E-mails die buiten kantooruren worden ontvangen, worden verzameld, maar er kan geen actie worden gegarandeerd tot de volgende werkdag
- We doen ons best om en zo goed mogelijke performantie te voorzien maar we kunnen geen vast percentage of uren uptime verzekeren

Serviceaanvragen

Ter ondersteuning van de services die in deze overeenkomst worden beschreven, zal de serviceprovider binnen de volgende termijnen reageren op service gerelateerde incidenten en/of verzoeken die door de Klant zijn ingediend:

- 0-8 uur (tijdens kantooruren) voor problemen die zijn geclassificeerd als hoge prioriteit
- Binnen 48 uur voor problemen die zijn geclassificeerd als gemiddelde prioriteit
- Binnen 7 werkdagen voor problemen die zijn geclassificeerd als lage prioriteit

Hulp op afstand wordt verleend in overeenstemming met de bovenstaande tijdschema's, afhankelijk van de prioriteit van het ondersteuningsverzoek.

Besluit SLA

Betwistingen over dit document worden onderling geregeld tussen alle belanghebbende partijen. Als er een fout staat in dit document mag u altijd via email contact met ons opnemen, dank u wel.

6 **BESLUIT**

De afgelopen weken hebben we hard gewerkt om dit project tot een goed eind te brengen. We hadden graag meerdere features geïmplementeerd maar door het werk dat erbij kwam door het samenvoegen van de verschillende onderdelen hebben we dit niet kunne doen. Met ons project kunne we manueel een PHP-project implementeren en volledig automatisch een standaard pagina laten aanmaken voor elk nieuw domein.

Onze back-ups van onze data gaat volledig automatisch en gebeurt elk uur. Alle nodige bestanden worden geback-upt naar een Onedrive. In het begin hadden we niet gedacht dat we de automatische back-ups en het automatisch aanvragen van een SSL-certificaat zou lukken. We zijn trots op de oplossingen die we voor deze oplossingen gevonden hebben.

In dit project hebben we geleerd om samen te werken met een groot team. We hebben ook geleerd om een groot project onder te verdelen in kleine taken. We hebben elkaar geholpen als er iemand vast zat en we hebben via een online Trello bord onze vooruitgang bij gehouden.

Door goed teamwerk en doorzettingsvermogen hebben we dit project tot een goed eind gebracht.