Linux Project Hosting

WLNinnovations

Naam:

Ward Boeckx
Lorzenzo Elias
Niek Smets

Bachelor in Elektronica-ICT
Option: Cloud & Cybersecurity

Academiejaar: 2022-2023

Campus : Geel

Inhoud

1 INLEIDING

In dit document staat ons verslag van het project hosting. Wat kan er verstaan worden onder "het project hosting"? We hebben de vraag gekregen van de klant om een hostingplatform op te zetten. Waarom? De klant heeft zelf een x-aantal klanten die een detailhandelszaak hebben. Hiervoor zou hij/zij graag een platform creëren waar deze zaken de kans krijgen om hun goederen ook online aan te bieden.

Ons project heeft dus als doel om een zo gebruiksvriendelijke interface te maken waar klanten, zonder enige kennis van IT, een website kunnen maken.

In dit document gaan we de opbouw van ons project verder uitleggen. We gaan het over de technische aspecten van ons project hebben, hoe klanten moeten inloggen, veiligheidsprocedures,.. Dit alles doen we door de vereiste van de klant en onze OPS report cards in het achterhoofd te houden.

Tot slot gaan we nog een handleiding schrijven die het voor nieuwe gebruikers zo toegankelijk mogelijk maakt om een nieuw account/nieuwe website aan te maken.

Dit alles met als doel hosting te vereenvoudigen.

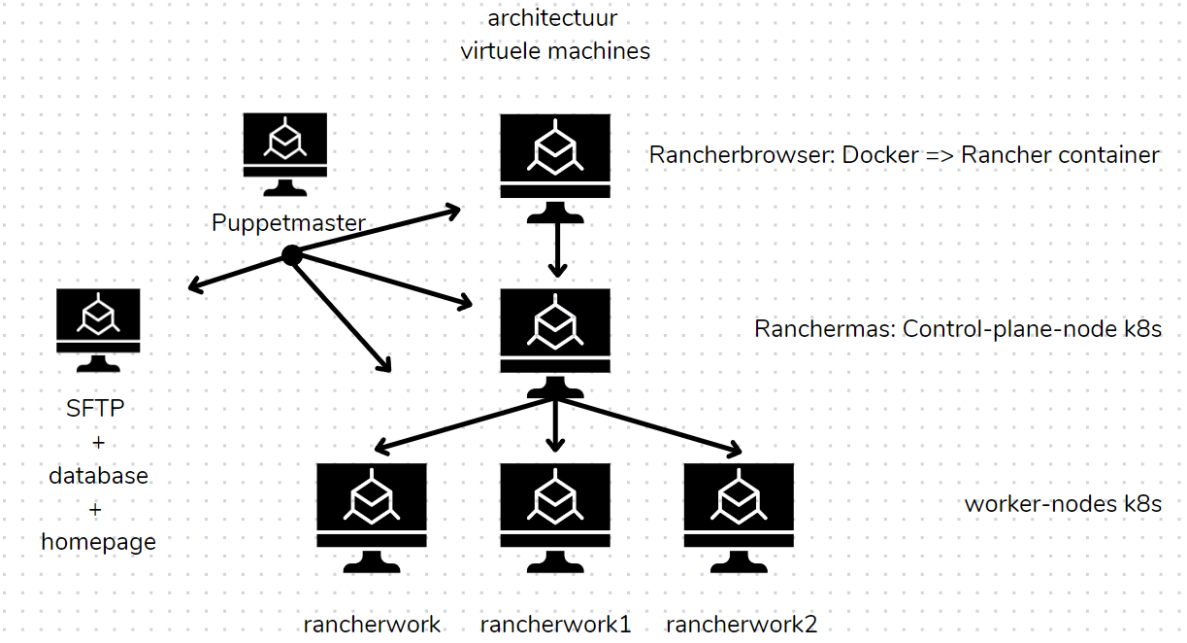
Veel leesplezier!

2 TECHNISCHE BESCHRIJVING

De technische samenstelling van ons project zal verder uitgelegd worden in dit hoofdstuk.

2.1 Schematische voorstelling

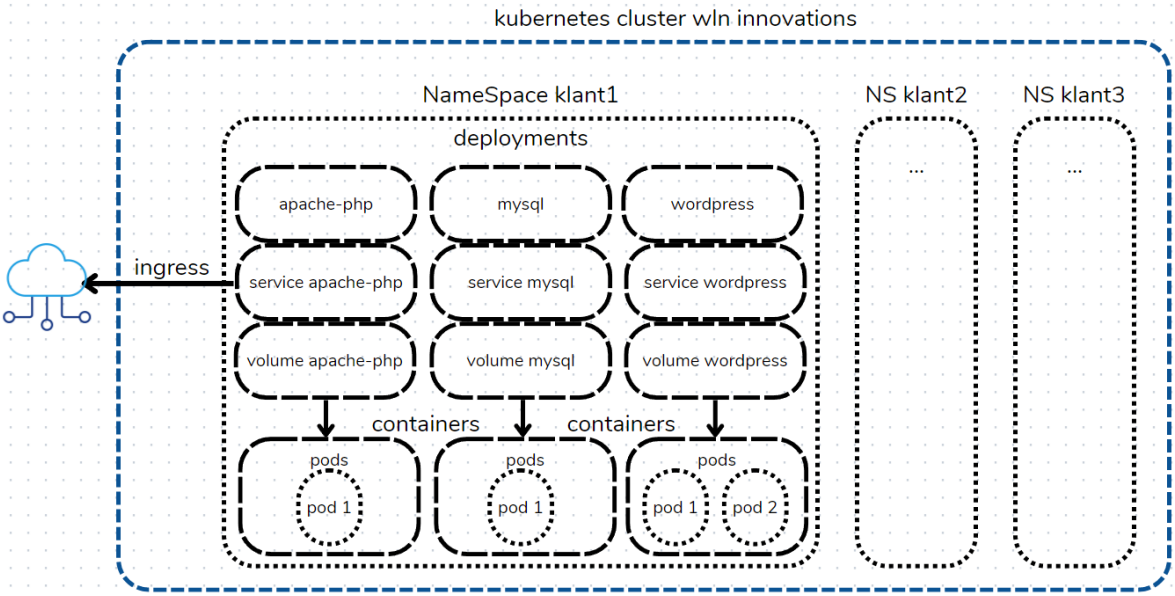
2.1.1 Architectuur virtuele machines



Figuur 1: architectuur virtuele machines (WLN Innovations)

Bij het opbouwen van onze machines en de Kubernetes cluster (wordt verder in het document nog verduidelijkt) hebben we gekozen om met Rancher te werken. Zoals zichtbaar in het schema hebben we een Rancherbrowser, deze draait de Docker container van Rancher. Indien er configuratie dient te gebeuren van Rancher, zal er naar het IP-adres van de Rancherbrowser gesurft worden. Hierdoor wordt het overzichtelijk dashboard van Rancher weergegeven. Zoals zichtbaar in het schema is er een Ranchermas geconfigureerd met drie worker-nodes. Rancher zal op zijn beurt aan load balancing doen door te zorgen dat elke work-node overeenkomstig belast wordt. De Puppetmaster zal ervoor zorgen dat al de virtuele machines onderhouden worden op regelmatige basis. Al deze virtuele machines worden als puppetclients geconfigureerd, zodat deze configuratie eenvoudig vanop 1 centrale locatie kan gebeuren. Verder hebben we nog een aparte virtuele machine waarop onze homepage staat. Wanneer een nieuwe klant zich aanmeldt, zal zijn/haar wachtwoord hier in de database worden opgeslagen en zal er automatisch een /home/klant map aangemaakt worden, waarheen de klant kan connecteren via SFTP. (Dillon, 2023)

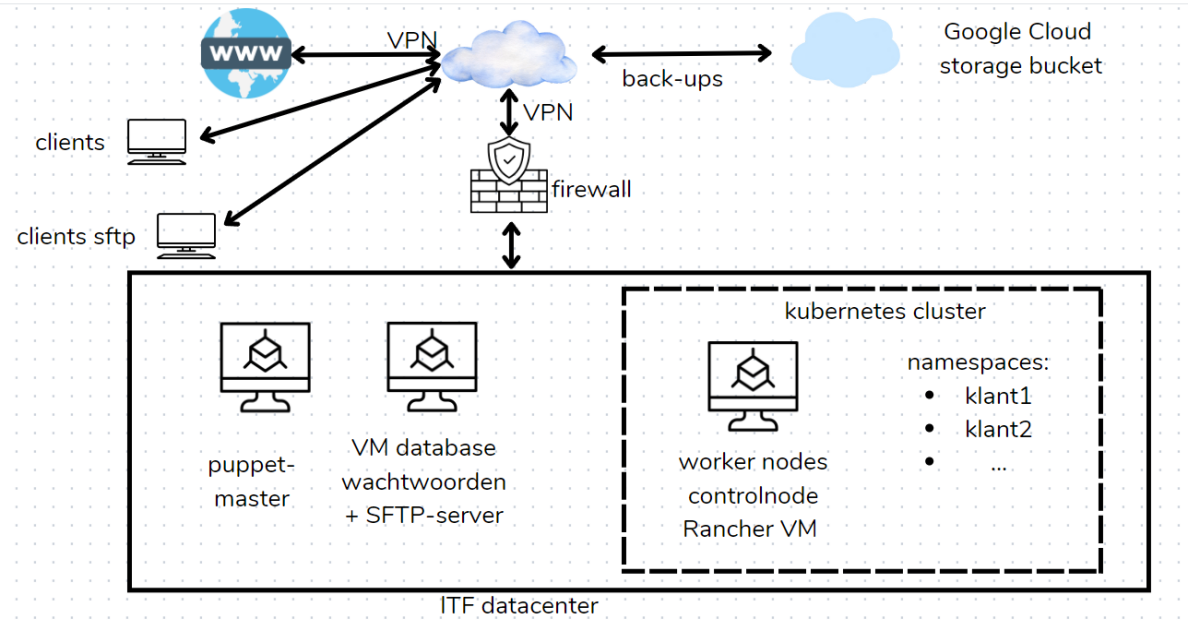
2.1.2 Opbouw Kubernetes cluster



Figuur 2:Opbouw Kubernetes cluster (WLN Innovations)

De Kubernetes cluster is opgebouwd met een duidelijke architectuur om de applicaties efficiënt en betrouwbaar te laten draaien. In de cluster wordt er per klant een Namespace gedeployed, deze fungeert als container, die er voor zorgt dat de applicaties van onze klant (apache-php, mysql, wordpress,…) gegroepeerd staan en eenvoudiger te beheren zijn. Om deze applicaties toegankelijk te maken voor de buitenwereld, wordt er een ingress geïmplementeerd. Deze fungeert als toegangspunt voor externe verzoeken naar de cluster. Het leidt inkomend verkeer naar de juiste services binnen de deployments door middel van configuratie. Hierdoor kunnen externe gebruikers veilig verbinding maken met de applicaties. (Authors, 2023)

2.1.3 Samenhang hosting



Figuur 3: Samenhang hosting (WLN Innovations)

Hoe zorgen al deze componenten nu voor een webhosting? Onze machines draaien op de servers in het datacenter van Thomasmore, een verbinding kan dus enkel mogelijk gemaakt worden door het gebruik van een VPN-verbinding.

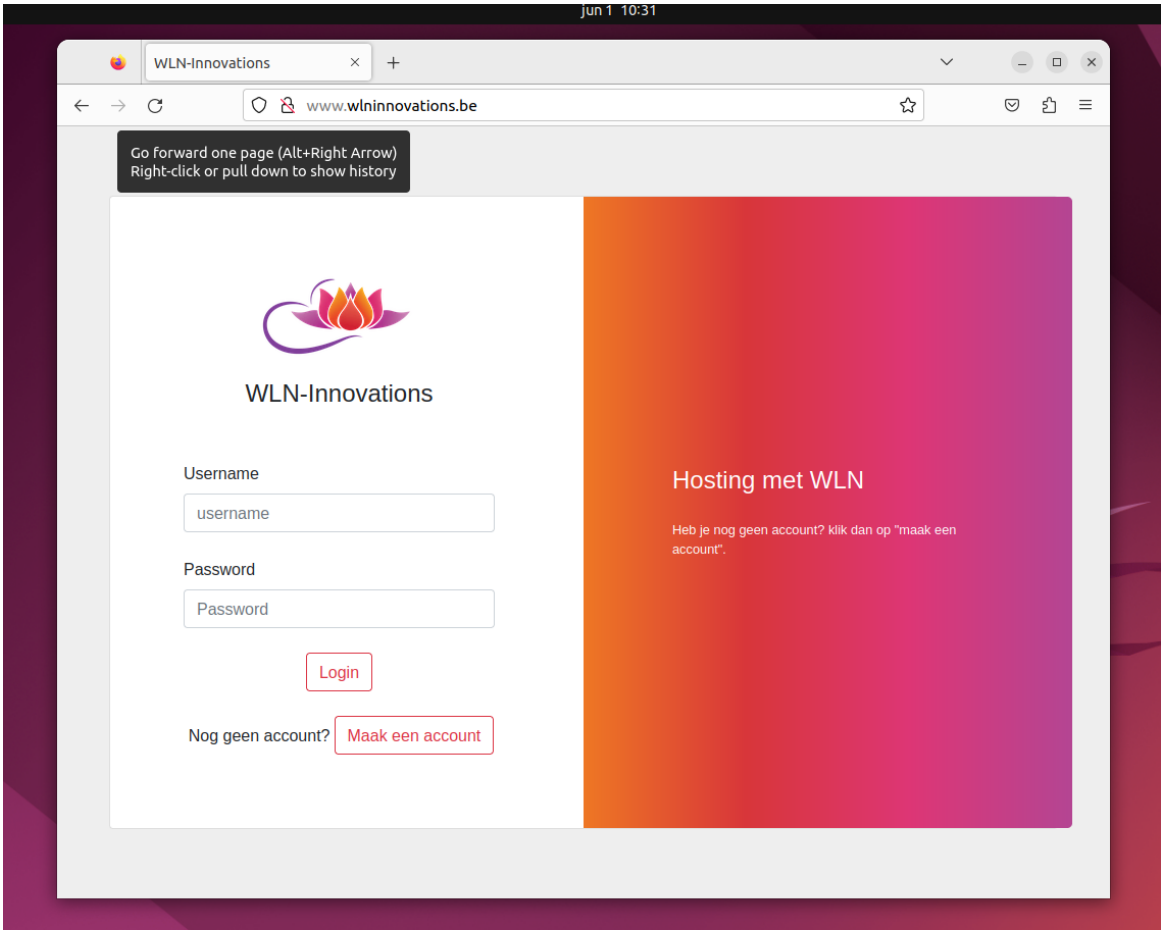
Als een klant naar de website www.wlninnovations.be surft en daar een nieuw account maakt, of inlogt met zijn gegevens, wordt deze doorverwezen naar bijvoorbeeld www.klant01.wlninnovations.be. Hierdoor wordt de klant doorverwezen naar het IP-adres van onze de load balancer samen met zijn Namespace. Rancher in samenwerking met Kubernetes weet dan welke services er moeten getoond worden aan de klant. Vervolgens krijgt de klant zijn/haar website te zien in de browser.

Om de machines up to date te houden wordt er gebruik gemaakt van Puppet, zo kan er gegarandeerd worden dat de configuratie up-to-date en consistent blijft. Buiten dit, staan alles inloggegevens van de machines ook nog opgeslagen in een password safe.

Ook wordt er gebruik gemaakt van de monitoring tool Grafana. Hierdoor is er een overzichtelijke weergave van CPU gebruik, geheugen,… Verder is er ook een Alertmanager, die ons een melding geeft wanneer er zich fouten voordoen en onder welke Namespace.

2.2 Werking onderdelen

2.2.1 Waar logt de klant in en hoe wordt hij doorverwezen?



Figuur 4: hoofdpagina www.wlninnovations.be (WLN Innovations)

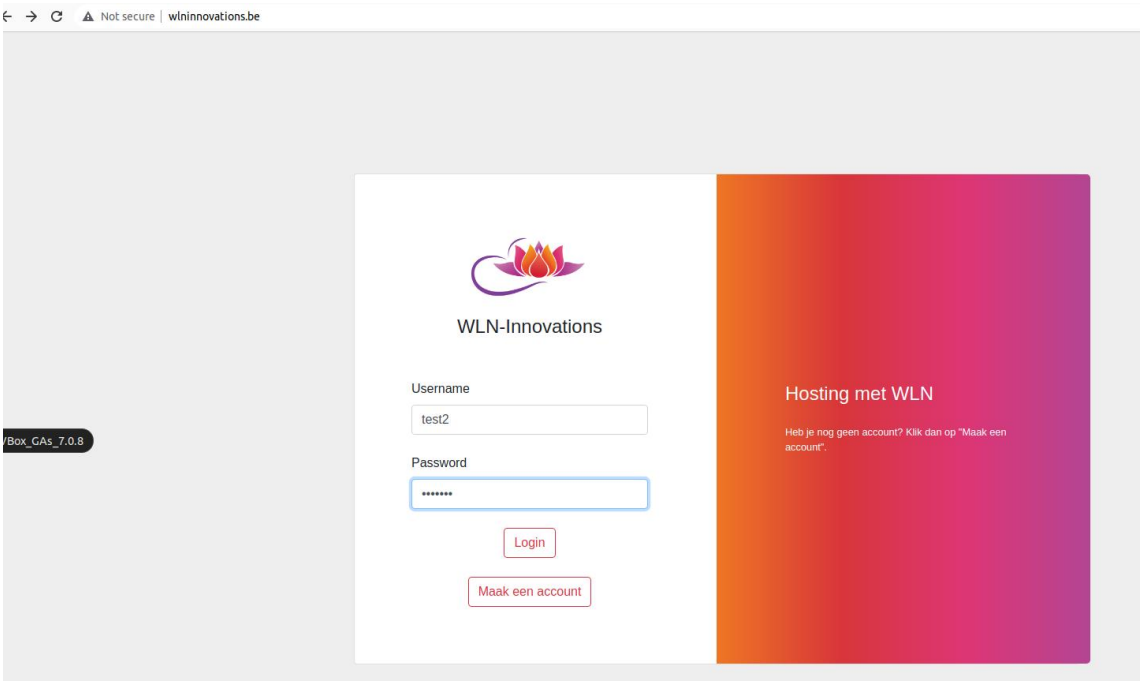
Wanneer de klant naar de pagina www.wlninnovations.be surft, komt hij op de hoofdpagina van het project uit. Hier zal hij/zij de keuze krijgen om zichzelf in te loggen met het eerder aangemaakt account of de kans krijgen om een account aan te maken.

```php
<?php
$servername = "mysql";
$username = "your_mysql_user";
$password = "your_mysql_password";
$dbname = "your_mysql_database";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

function getNamespaceFromCredentials($conn, $gebruikersnaam, $wachtwoord) {
    $stmt = $conn->prepare("SELECT namespace FROM users WHERE gebruikersnaam = ? AND wachtwoord = ?");
    $stmt->bind_param("ss", $gebruikersnaam, $wachtwoord);
    $stmt->execute();
    $stmt->bind_result($namespace);
    $stmt->fetch();
    $stmt->close();
    return $namespace;
}

function redirectUserToNamespace($namespace) {
    $mainDomain = "wlninnovations.be";

    $customerSiteURL = "www." . $namespace . "." . $mainDomain;

    header("Location: http://" . $customerSiteURL);
    exit;
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $gebruikersnaam = $_POST["gebruikersnaam"];
    $wachtwoord = $_POST["wachtwoord"];

    $namespace = getNamespaceFromCredentials($conn, $gebruikersnaam, $wachtwoord);

    if ($namespace) {
        redirectUserToNamespace($namespace);
    } else {
        $errorMessage = "Het wachtwoord en/of de gebruikersnaam is fout.";
    }
}

$conn->close();

?>
```

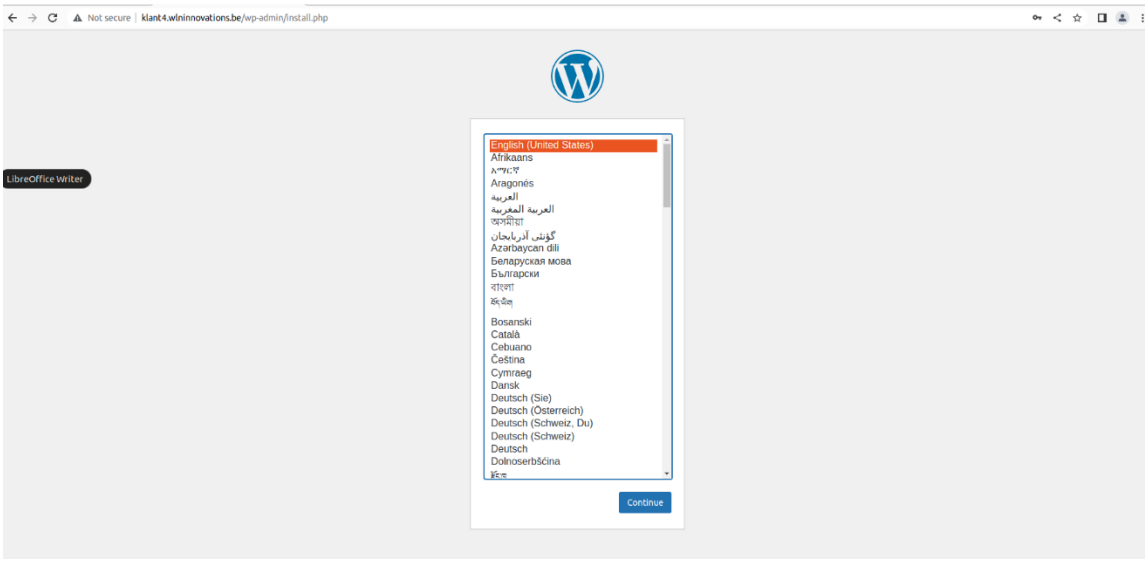
Figuur 5: script connectie database (WLN Innovations)

De moment dat de klant inlogt, zal er een connectie gemaakt worden met de database. Daarna zal er gekeken worden naar de inloggegevens die de klant heeft ingevuld en zal er naar de database gekeken worden of de klant zijn gebruikersnaam en wachtwoord wel effectief kloppen. Indien deze gegevens van de klant juist zijn, zal de klant doorgestuurd worden naar de webserver en daar op zijn beurt verder geholpen worden door Rancher. Indien de klant een foutieve combinatie ingeeft, zal er een het bericht “Het wachtwoord en/of de gebruikersnaam is fout” tevoorschijn komen.

Indien de klant dus inlogt, krijgen we het volgende te zien.



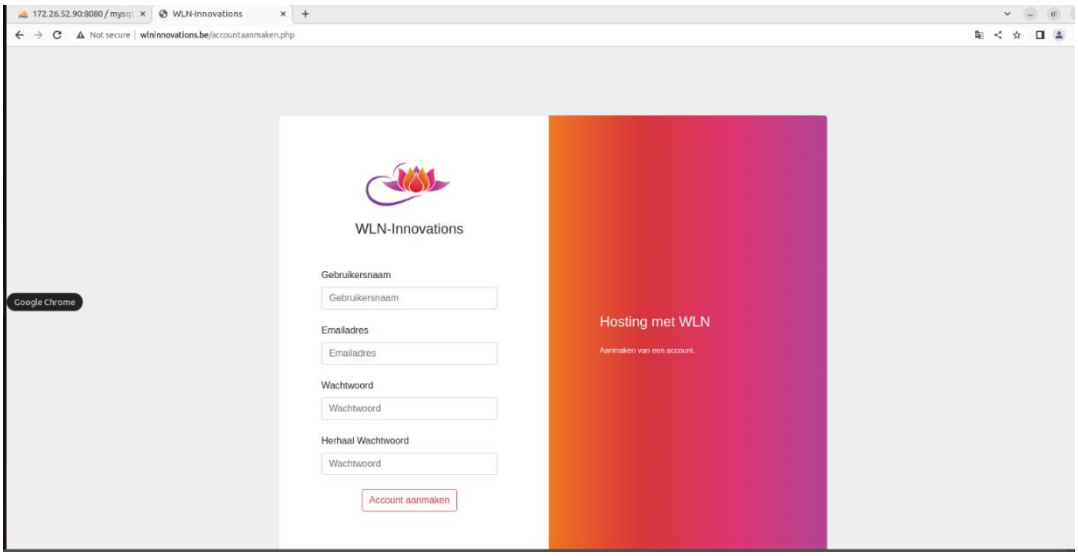
Figuur 6: aanmelden klanten (WLN Innovations)



Figuur 7: index.php Wordpress (WLN Innovations)

Bovenaan in de adresbalk is wordt er duidelijker aangetoond hoe het php-script in zijn werking gaat. Elke klant wordt bij het aanmaken van een account gekoppeld aan een waarde “klantX” waar “X” de variabele is. Dit zal gebruikt worden als subdomain voor de hoofdpagina www.wlninnovations.be. In dit geval wordt de klant doorgestuurd naar de begin pagina van Wordpress, hier kan de klant op zijn/haar beurt het designen van een website beginnen.

Als de gebruiker nog geen account heeft, kan hij op “maak een account” duwen, hier krijgt hij de mogelijkheid een account te creëren. De klant zal hier een gebruikersnaam, e-mailadres en wachtwoord moeten invoeren voor het aanmaken van een account. (WordPress, 2023)



Figuur 8: aanmaken account (WLN Innovations)

LinuxProjectHosting > newuser.php

```php
<?php
$servername = "mysql";
$username = "your_mysql_user";
$password = "your_mysql_password";
$dbname = "your_mysql_database";
$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT namespace FROM users ORDER BY namespace DESC LIMIT 1";
$result = $conn->query($sql);

$sql = "SELECT gebruikersnaam FROM users WHERE gebruikersnaam = '$gebruikersnaam'";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // Gebruikersnaam bestaat al, geef een melding
    echo "Gebruikersnaam bestaat al";
    $conn->close();
    exit;
}
if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();
    $lastNamespace = $row["namespace"];
    preg_match('/\d+/', $lastNamespace, $matches);
    $maxNamespace = intval($matches[0]);
    $newNamespace = "klant" . ($maxNamespace + 1);
} else {
    $newNamespace = "klant1";
}
$gebruikersnaam = $_POST["gebruikersnaam"];
$wachtwoord = $_POST["wachtwoord"];
$email = $_POST["email"];

$sql = "INSERT INTO users (namespace, gebruikersnaam, wachtwoord, email)
        VALUES ('$newNamespace', '$gebruikersnaam', '$wachtwoord', '$email')";

if ($conn->query($sql) === TRUE) {
    echo "User created successfully with namespace: $newNamespace";
} else {
    echo "Error creating user: " . $conn->error;
    $conn->close();
    exit;
}

$conn->close();
```

Figuur 9: php-script authenticatie (WLN Innvations)

Bij het bovenstaande php-script, is opnieuw te zien hoe er via lijn 2 tot en met 10, opnieuw connectie wordt gemaakt met de database. Daarna zal er eerst gekeken worden of er al klanten zijn met dezelfde gebruikersnaam, indien ja, zal er een melding "gebruikersnaam bestaat al" gegeven worden. Indien nee, wordt er bij de volgende stap gekeken naar de bestaande namespaces. Als er nog geen namespaces aanwezig zijn, en dus ook geen gebruikers, zal de eerste gebruiker een namespace "klant1" toegewezen krijgen. Na dat dit gebeurd is, zullen de gegevens in de SQL database opgeslagen worden.

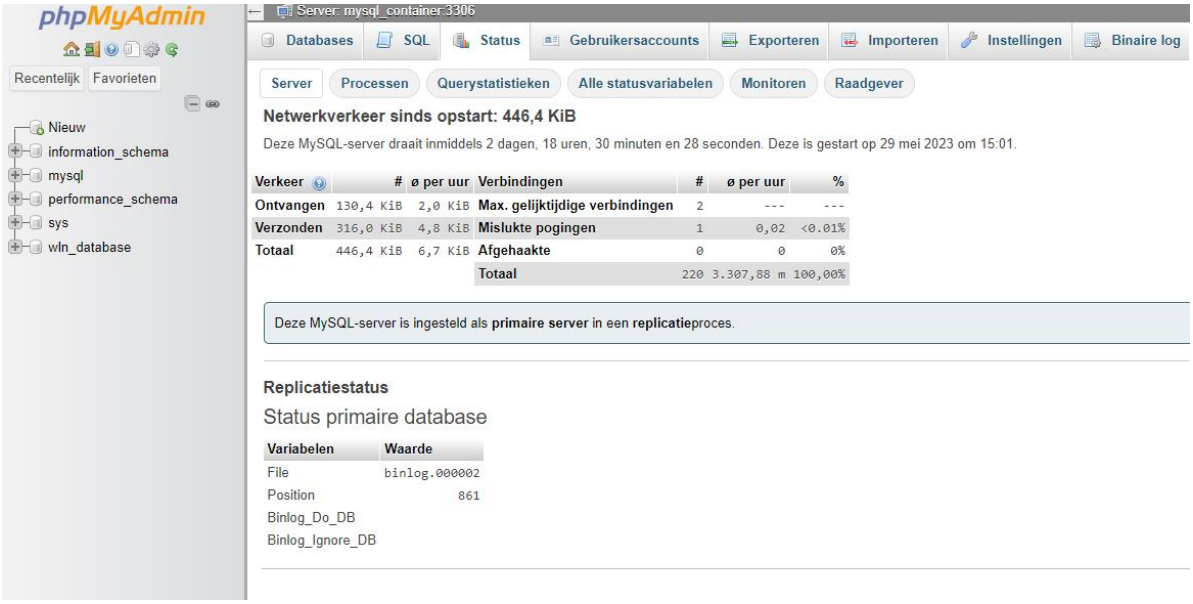
2.2.2 Hosting van de website

Zoals eerder vermeld zal na de registratie van de nieuwe klant in de database, een nieuwe Namespace in Rancher worden aangemaakt met de bijbehorende services die de klant nodig heeft.

2.2.3 De data van de klant opslaan

Wanneer er een nieuwe Namespace aangemaakt wordt voor een klant, wordt er ook voor gezorgd dat deze klant een eigen database heeft die ook voor de klant toegankelijk is. Deze database zal de klant dan bijvoorbeeld kunnen bezoeken op het adres www.klant01.db.wlninnovations.be.

Voor veiligheidsredenen hebben we gekozen om voor elke klant een aparte database op te zetten, deze zal beschikbaar zijn via phpMyAdmin.



Figuur 10: database op PhpMyAdmin (WLN Innovations)

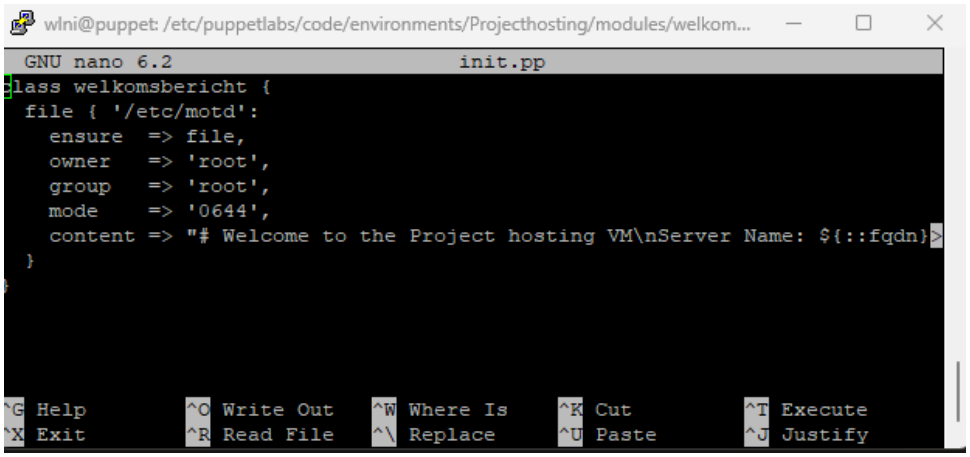
2.2.4 Beschikbaarheid van de websites

Om de klanten te garanderen dat hun website altijd beschikbaar zal zijn, zijn er meerdere lagen van redundanties toegepast. Zo hebben we, zoals eerder beschreven in het document, er voor gezorgd dat onze Kubernetes cluster is gedeployed met Rancher. Dit zorgt ervoor dat Rancher door middel van load balancing gaat verdelen en er dus bijna altijd ruimte zal zijn om de websites te draaien. Als één container faalt, is er altijd een andere die klaar staat om dit op te vangen.

2.2.5 Synchronisatie en configuratie

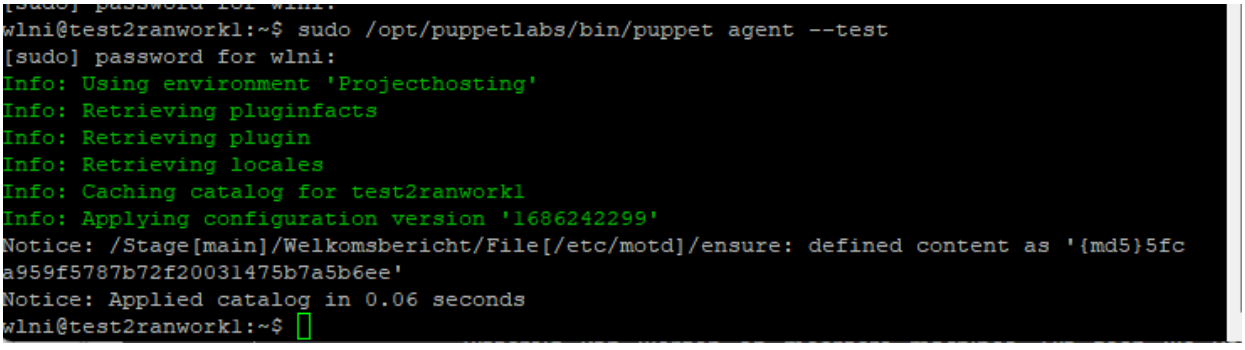
Wanneer er updates of een nieuwe configuratie moet gebeuren waarvan het de bedoeling is dat we deze op elke VM gaan draaien, wordt er niet handmatig ingelogd op elke VM om dan steeds dezelfde commando's in te geven. Dit gaan we doen aan de hand van een configuration management tool. In het Project Hosting van WLNinnovations wordt er gebruik gemaakt van Puppet. Het beheer wordt dus door één machine gedaan en wordt zo verder uitgerold naar de andere machines. Zodanig dat een configuratie/update slechts één keer hoeft neergeschreven te worden en zo uitgerold kan worden op meerdere machines. Dit doen we volgens het "Canary process", dit wordt verder in het document nog uitgelegd.

Verder is hier een voorbeeld van een manifest file met de naam "welkomstbericht". Dit bestand dient als een blauwdruk voor het configureren en beheren van systeemresources. Het manifestbestand beschrijft de gewenste toestand van het systeem, en Puppet zorgt ervoor dat het systeem wordt gebracht naar die gewenste toestand.



Figuur 11: init.pp Puppet-welkomsttbericht (WLN Innovations)

De screenshot hieronder geeft nog weer dat dit met succes is gelukt.



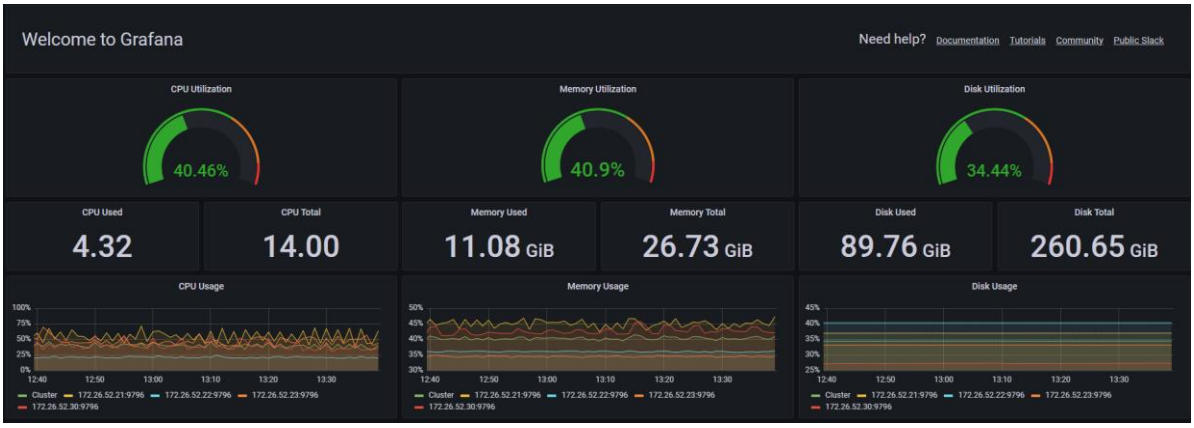
Figuur 12: succesvolle initiatie welkomstbericht (WLN Innovations)

```
# Welcome to the Project hosting VM
Server Name: test2ranwork1.itfdc.local
IP Address: 172.26.52.102
Last Updated:
Last login: Thu Jun  8 16:43:52 2023 from 192.168.213.3
wlni@test2ranwork1:~$
```

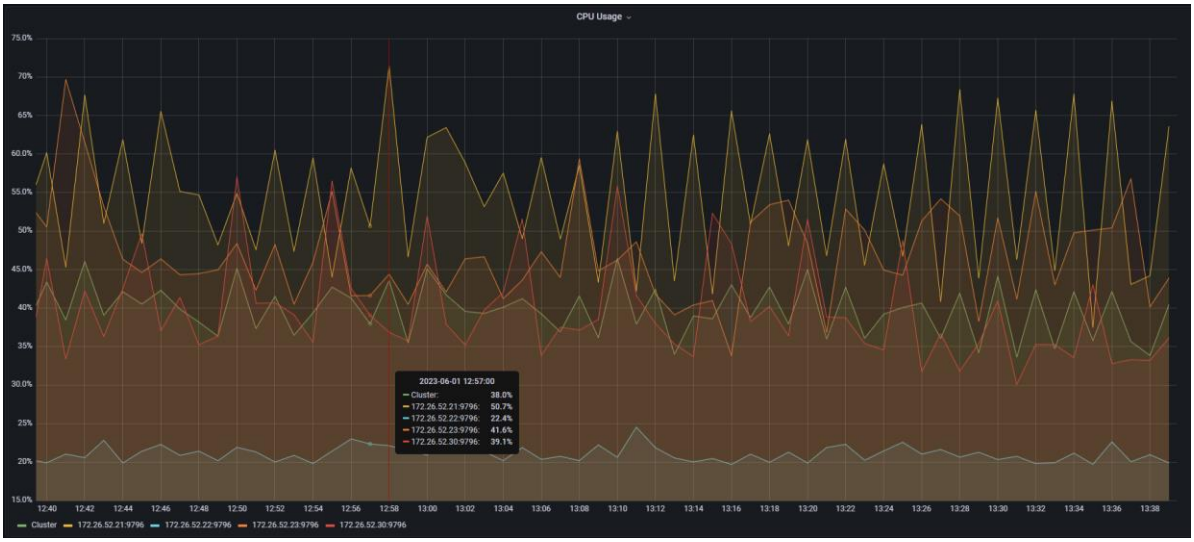
Figuur 13: screenshot welkomstbericht (WLN Innovations)

2.2.6 Monitor Tools

Monitoring is belangrijk om eventuele cruciale fouten vroegtijdig op te merken zodat er kan ingegrepen worden waar nodig. Dit is dan ook onmisbaar in het project, hiervoor is de monitoring tool Grafana gekozen. (Suse, Enable Monitoring, 2023)

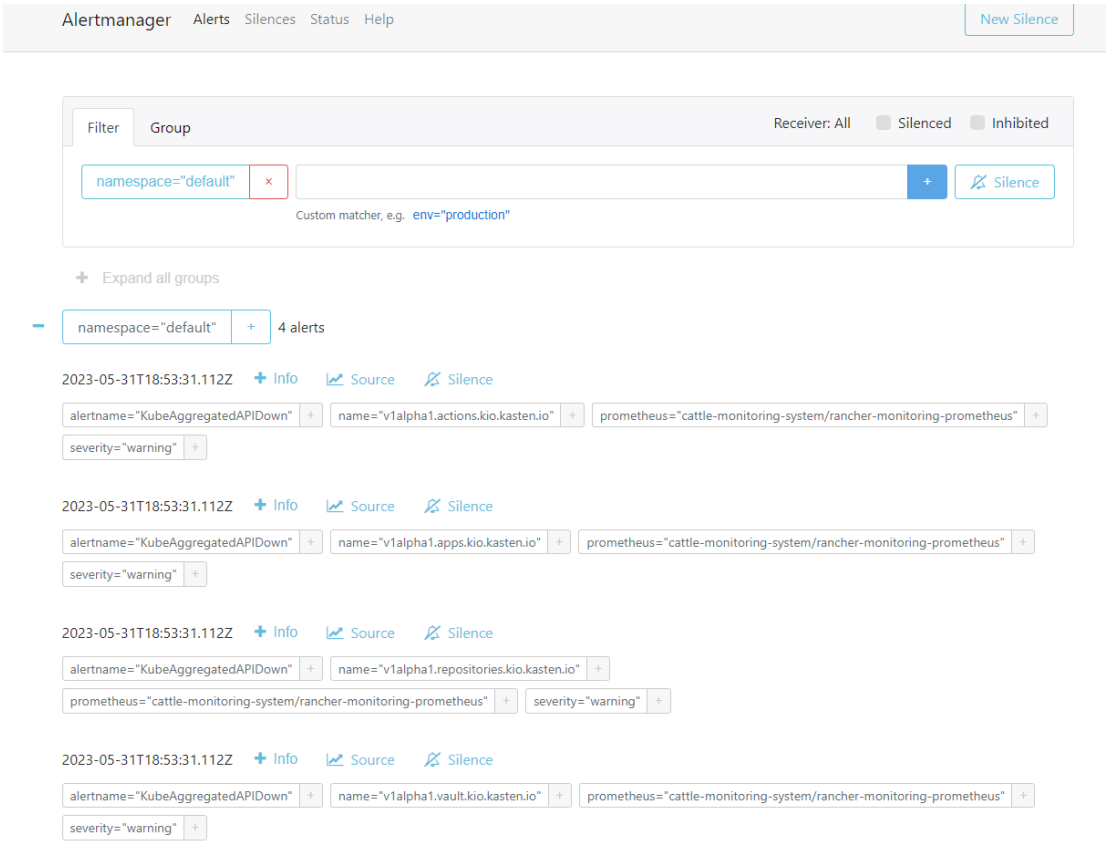


Figuur 14: Grafana monitoring CPU, memory & disk (WLN Innovations)



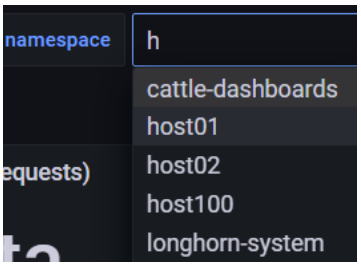
Figuur 15: Grafana monitoring CPU in detail (WLN Innovations)

Hier kan de algemene gezondheid van de machines bekeken worden. Denk maar aan CPU utilization, Memory utilization, Disk utilization,… Dit wordt weergeven in grafieken waar per specifiek IP-adres kan gekeken worden hoeveel recources er wanneer gebruikt zijn. Ook wordt er gebruik gemaakt van een Alertmanager die aangeeft waar en welke fouten zich voordoen. (Shivang, (S.a.))

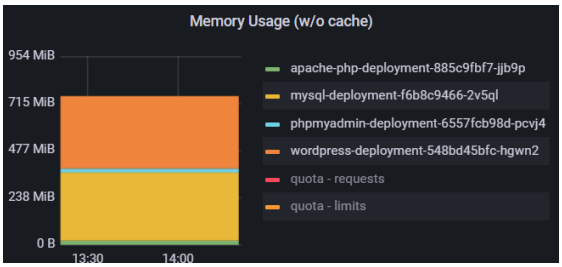


Figuur 16: alerts in alertmanager (WLN Innovations)

Op niveau van de namespaces kan je ook kijken in Grafana. Hieronder vind je een afbeelding waar het geheugengebruik van host01 wordt getoond.

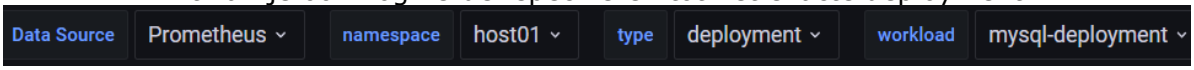


Figuur 17: monitoren namespaces (WLN Innovations)



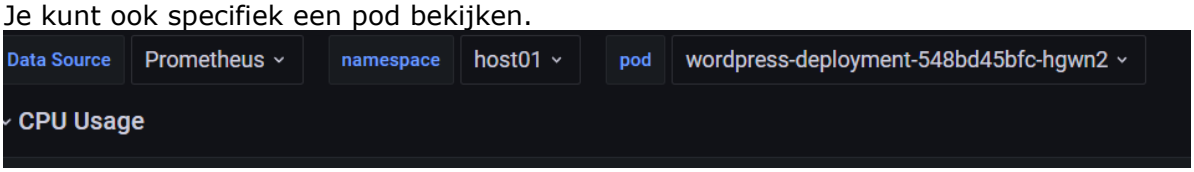
Figuur 18: monitoren deployments host01 (WLN Innovations)

Dit kan je ook nog verder specifiëren tot het exacte deployment.



Figuur 19: monitoren van een specifiek deployment (WLN Innovations)

Je kunt ook specifiek een pod bekijken.



Figuur 20: monitoren van een specifieke pod (WLN Innovations)

Overzichten kan je ook bekijken op het niveau van de nodes die binnen de Kubernetes cluster actief zijn.

Figuur 21: monitoren van een specifieke node (WLN Innovations)

Op deze manier zijn ook monitoring tools van de pvc's te raadplegen.

Figuur 22: monitoren van een specifieke pvc (WLN Innovations)

2.2.7 Back-ups

2.2.7.1 Back-ups bucket in Google Cloud

Indien er zich een fout voordoet waardoor gebruikers hun website niet meer kunnen raadplegen omdat de bestanden weg zijn, maken we gebruik van een back-up. Elke nacht om 02.00u wordt er een back up genomen die in de Google cloud wordt opgeslagen. Hierdoor kunnen gebruikers snel voorzien worden van eerder opgeslagen data. Wij hebben een account genomen op Google Cloud: het project 'wln innovations'. Van onze virtuele machines en van onze Kubernetes cluster zijn ook snapshots genomen voor of na momenten van cruciale aanpassingen. Wanneer er dan onverwachts iets misloopt, kan je ten allen tijde teruggaan naar een vorige versie, waar alles nog wel juist geconfigureerd was. (Google, 2023)

2.2.8 Beveiliging

Bij beveiliging wordt er aandacht besteed aan wachtwoorden die niet in "plain tekst" staan in de database. Ook zouden er self-signed SSL certificaten gemaakt worden zodat de gebruikers er zeker van zijn dat hun data veilig wordt verzonden over het net.

Hierbij wordt er rekening gehouden met de best best practices voor web hosting security. Denk aan Back-ups, SSL, SFTP, regelmatig veranderen van wachtwoorden. (Cloud G. , 2023)

2.2.8.1 SFTP

SFTP maakt gebruik van een vaste poort (22) om al de gewenste overdrachten en opdrachten te versleutelen. In tegenstelling tot FTP, wordt hier wel degelijk gebruik gemaakt van encryptie. SFTP staat voor Secure File Transfer Protocol en is gebouwd op een beveiligd subsysteem van SSH (de secure shell). SFTP bevat een sterke beveiliging

door het gebruik van deze SSH-sleutels voor authenticatie. Het wordt massaal door bedrijven gebruikt voor de salarisadministratie en pensioenregelingen omdat het juist zo veilig is. (Mulan, 2023)

2.2.8.2 Veiligheidsscans

Er worden regelmatig veiligheidsscans genomen in Rancher aan de hand van de app CIS Kubernetes Benchmark. Het maakt gebruik van kube-bench, een open source tool van Aqua Security. Dit is een efficiënte manier van pentesting, waarbij er op regelmatige basis een veiligheidsinspectie gebeurt van onze Kubernetes cluster op basis van best practice-beveiliging. Scans kunnen handmatig of automatisch ingepland worden. In de scanresultaten staan punten ter verbetering. In de punten staan ook specifieke suggesties om dit probleem aan te pakken. (Suse, CIS scans, 2023)

2.2.8.3 Aanmeldbanner

Wanneer een nieuwe connectie via SSH wordt gemaakt, zal je een boodschap krijgen dat er geen toegang wordt verleend aan personen die hier geen recht op hebben en dat je vervolgd kunt worden als je toch doorgaat. (Kiarie, 2021)

```
wlni@test2ranwork1:~$ ssh wlni@172.26.52.103
The authenticity of host '172.26.52.103 (172.26.52.103)' can't be established.
ED25519 key fingerprint is SHA256:Y7nSqN4dXb/I8GYfE6IO6pUSvTY0A8OtduGiAqRdFyk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.26.52.103' (ED25519) to the list of known hosts
Attention! This is a secured aria!
You are entering WLN-Innovations.
Only access for authorized users!

You will be monitored when trying to log in.
We will take legal actions if you continue without authorisation!
```

Figuur 23: banner.txt bij ssh-connectie (WLN Innovations)

2.2.8.4 Fail2ban

Fail2ban is een tool die je kunt implementeren om brute-force aanvallen aanzienlijk te verminderen. Wanneer er enkele keren achter elkaar een foutief wachtwoord is ingegeven, zal dit specifieke ip-adres geblokkeerd worden voor een bepaalde tijd (bv. 10 minuten). Factoren zoals aantal aanmeldpogingen, tijdsduur van blokkeren, het pad naar de logfiles instellen, e-mailadres opgeven voor meldingen, algemene of specifieke configuratie toepassen, etc… kan men hier allemaal instellen. In onderstaand screenshot zie je een voorbeeld hoe een ip-adres geblokkeerd wordt na teveel foutieve aanmeldpogingen: (DigitalOcean, 2022)

```
lni@test2ranwork1:~$ ssh wlni@172.26.52.103
sh: connect to host 172.26.52.103 port 22: Connection refused
lni@test2ranwork1:~$
```

Figuur 24: melding na 3 foutieve ssh-aanmeldingen (WLN Innovations)

3 PROCEDURES EN HANDLEIDINGEN

Procedures die we moeten volgen wanneer er iets misgaat, handleidingen voor de collega's en mensen met basis IT kennis zullen verder in dit hoofdstuk verduidelijkt worden.

3.1 Disaster recovery procedures

Per onderdeel van het project zijn aparte procedures/handleidingen geschreven.

3.1.1 Volledige datacenter dat down is

Hierbij zouden de gebruikers op de hoogte moeten gebracht worden van het feit dat onze services enige tijd niet beschikbaar zullen zijn. Wel kunnen ze gerust gesteld worden door het feit dat er back-ups in de cloud zijn, dus er zullen geen of weinig gegevens verloren gaan. Indien er een nieuwe server moet geïnstalleerd worden is het een kwestie van nieuwe VM's aan te maken, de rest van de configuratie kan met Puppet gedaan worden. Nadat dit succesvol gebeurd is en de nodige testen zijn uitgevoerd, kunnen de gebruikers op de hoogte gebracht worden dat onze services terug operationeel zijn.

3.1.2 Configuration management system Puppet down

Een probleem dat niet direct hinderlijk is voor de klanten maar wel belangrijk voor de systeembeheerders. Indien dit voorvalt, wordt er naar de logs gekeken en mogelijks zo een oplossing gevonden. Indien dit geen optie is zal Puppet opnieuw geïnstalleerd worden zonder dat de klanten hier hinder van ondervinden.

3.1.3 Rancher down

Indien Rancher volledig down gaat met bijbehorende worker-nodes, zal er naar de rancherbrowser moeten gekeken worden voor te verifiëren of de Docker-container nog draait. Indien dit het geval is zal er naar netwerk connectiviteit moeten gekeken worden. Als er dan nog geen oplossing gevonden wordt, kunnen er altijd snel nieuwe VM's opgestart worden met behulp van Puppet. Daarna kan de back-up naar boven gehaald worden en zal de nodige configuratie gebeuren om Rancher zo snel mogelijk werkend te krijgen.

3.1.4 Worker-node van Rancher down

Als er een worker-node van rancher down gaat, ondervinden de klanten hier geen hinder. Dit omdat de control plane van Rancher de belasting verder zal verdelen over de twee overgebleven nodes. Verder zou er naar de Grafana monitorring tool gekeken worden om te kijken of er iets uitzonderlijk gebeurd is dat ongedaan kan gemaakt worden. Indien er geen fout gevonden wordt en de VM down blijft, wordt deze VM verwijderd. Daarna zal snel een nieuwe VM aangemaakt worden met de nodige configuratie. Hierna kan de control plane de belasting terug gelijk verdelen over de 3 worker-nodes.

3.1.5 Kubernetes Cluster down

Indien de gehele cluster van Kubernetes down gaat, zal er bekeken worden of er een duidelijke oorzaak is die kan verholpen worden. Als dit niet het geval is wordt er een nieuwe cluster aangemaakt met behulp van back-ups.

3.1.6 Nginx-ingress down

Als er ergens zich ergens een error voordoet bij één van de ingresses, is het belangrijk dat dit zo snel mogelijk opgelost wordt. Klanten zullen hier rechtstreeks hinder ondervinden aangezien bepaalde services niet meer beschikbaar zullen zijn. Wanneer dit voorvalt zal er gekeken worden naar logs in de Grafana monitorring tool, indien er hier duidelijke fouten zijn die kunnen voorkomen worden, wordt dit geregeld. Kan dit niet op die manier verwezenlijkt worden, zal er een nieuwe ingress geconfigureerd worden.

3.1.7 Grafana monitorring tool down

Stel dat de Grafana monitorring tool down gaat. Zou er gekeken worden of er nog eventuele logs beschikbaar zijn die van algemeen nut zouden zijn. Is er nergens een duidelijk antwoord te vinden in de logs, wordt de container in Rancher opnieuw opgestart. Dan zal er ook opnieuw gekeken worden naar de configuratie, is er hier iets niet meer in orde, wordt dit aangepast.

3.1.8 Database op de VM die de main page host down

Hier gaan klanten rechtstreeks hinder van ondervinden. Dit omdat bij het inloggen op de hoofdpagina een connectie word gemaakt naar de SQL database. Hierbij worden ingevulde gegevens van de gebruiker vergeleken met gegevens uit de database. Echter worden ook hiervan back-ups gemaakt. Eerst zal gekeken worden naar netwerkconnectiviteit, dan wordt de configuratie opnieuw nagekeken. Blijven errors zich voordoen, kan een nieuwe database snel opgezet worden door middel van back ups en Puppet.

3.1.9 Database in de Namespace van de klant gaat stuk

Ook hier zou een specifieke klant direct hinder ondervinden. Dit doordat zijn gegevens niet meer kunnen opgehaald worden uit de database, noch gegevens in de database kan stoppen. Hier kan snel een nieuwe SQL container opgestart worden. Aangezien er gebruik wordt gemaakt van PVC (Persistent Volume Claim), kan de database opnieuw geïnstalleerd worden zonder dataverlies. Moest dit toch het geval zijn is er altijd een back-up beschikbaar.

3.1.10 Service in een Namespace gaat down

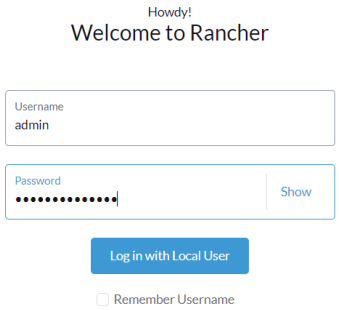
Net zoals met de database in een Namespace wordt hier een back-up van gemaakt. Indien het troubleshooten geen succes oplevert en er niet kan herstart worden. Is het dus een optie om de getroffen service te verwijderen en opnieuw te deployen. Ook weer met behulp van PVC. Als dit toch met dataverlies gepaard gaat, wordt de back-up ingeschakeld.

3.2 Handleiding voor IT-medewerkers

Een handleiding voor mensen met IT-kennis zal in dit hoofdstuk neergeschreven worden.

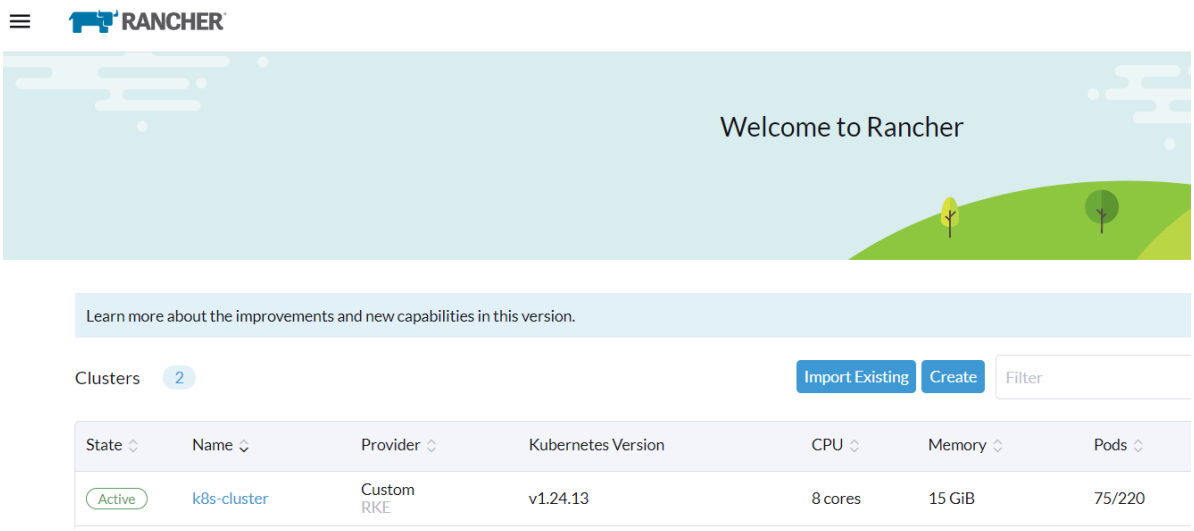
3.2.1 Gebruik van Rancher

De virtuele machine 'rancherbrowser' is gebruikt om Rancher te configureren. Om verbinding te maken met Rancher, ga je naar de browser en gebruik je de url https://172.26.52.100:443 Daarmee kom je op de inlogpagina van Rancher. Als gebruikersnaam gebruik je 'admin', als wachtwoord gebruik je 'wlninnovations'. (LinuxTechi, 2023)



Figuur 25: aanmelden in Rancher (WLN Innovations)

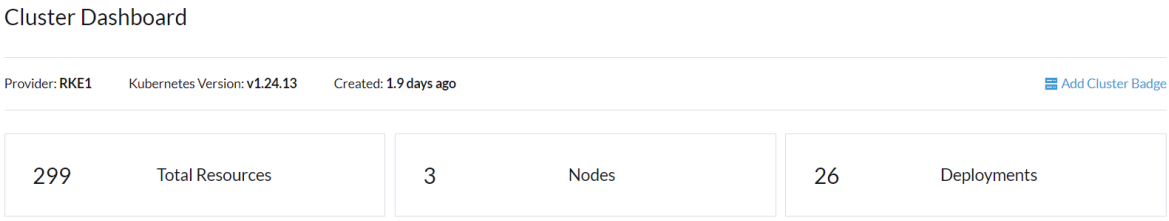
Op de welkomstpagina van Rancher vind je direct onze kubernetes-cluster 'k8s-cluster'. Klik op de clusternaam om verdere info te verkrijgen.



Figuur 26: welkomstpagina Rancher (WLN Innovations)

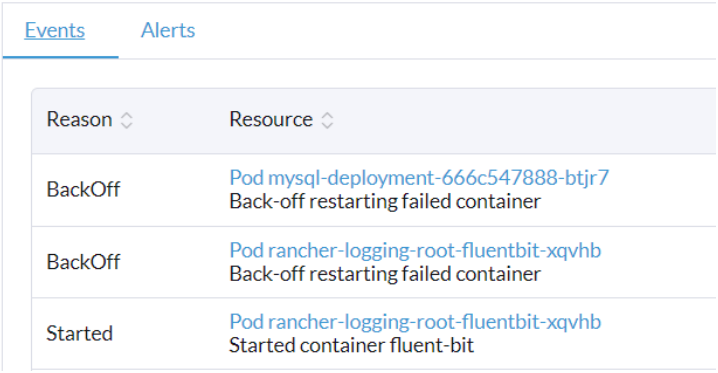
Op het dashboard van de cluster kan je veel info vinden:

- Het aantal recources
- nodes
- deployments



Figuur 27: cluster dashboard (WLN Innovations)

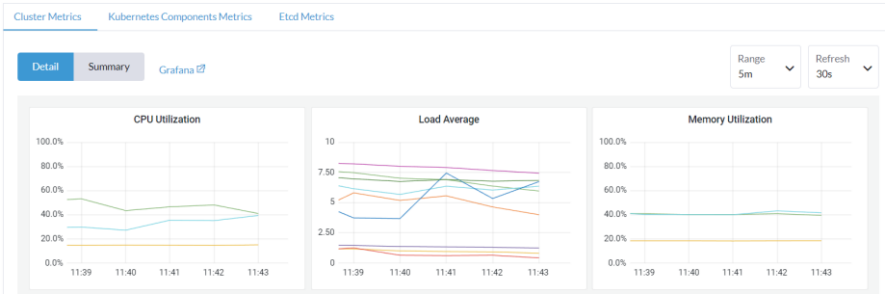
- Berichten over gebeurtenissen en alarmen



Figuur 28: gebeurtenissen en alarmen (WLN Innovations)

Verschillende metrics over de Kubernetes-cluster zoals:

- CPU-gebruik
- Load average
- Geheugen
- Netwerkverkeer
- Podinfo

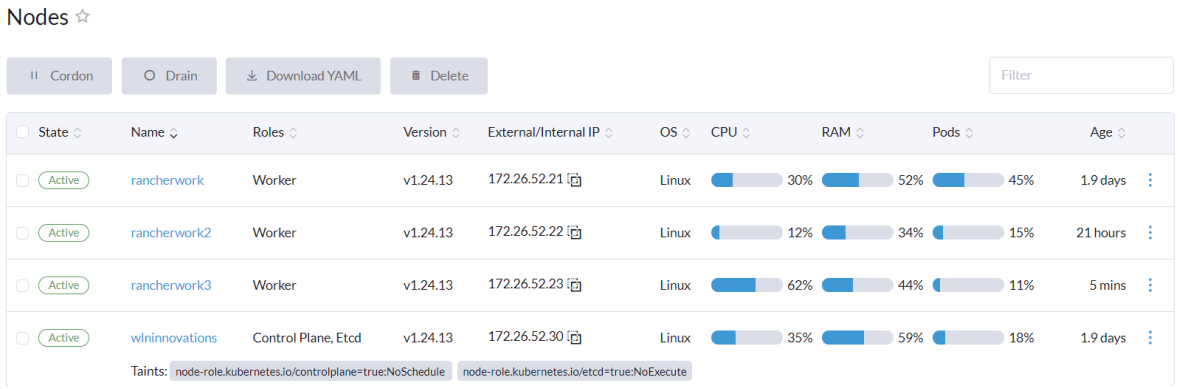


Figuur 29: monitoren van de volledige cluster (WLN Innovations)

Namespaces: elke host zal een aparte namespace krijgen waarin zijn/haar containers draaien. Ook onze aanmeldpagina voor nieuwe klanten zal in zo'n aparte namespace zitten. Op het tabblad Projects/Namespaces kunnen wij de verschillende namespaces beheren.

Figuur 30: namespaces van de cluster (WLN Innovations)

Nodes: Op het tabblad Nodes vind je een overzicht van de actieve nodes. Wij gebruiken 1 control-plane-node en 3 worker nodes. Per node kan je hier ook info vinden over CPU-gebruik, RAM geheugen en procentueel aantal pods.



Figuur 31: nodes van de cluster (WLN Innovations)

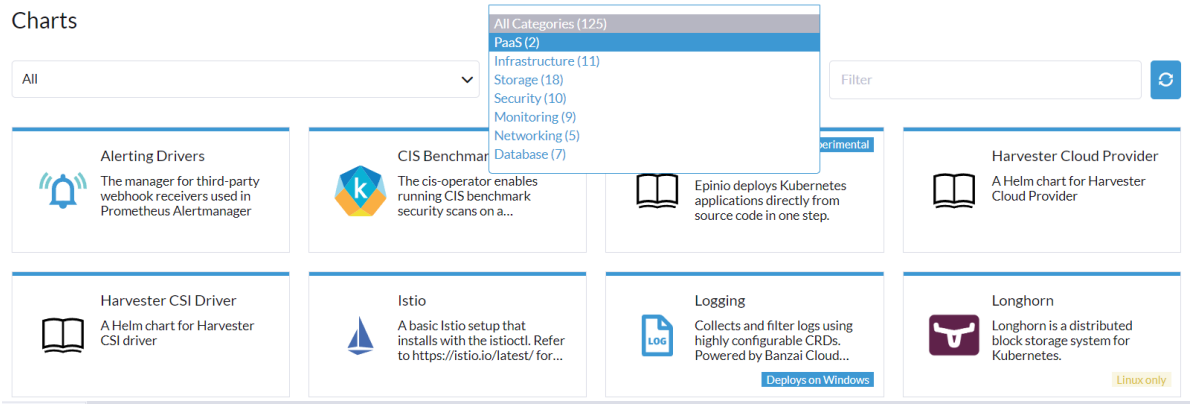
Het is mogelijk om nog nodes toe te voegen, moesten de capaciteiten van de huidige nodes onvoldoende zijn. Start hiervoor een nieuwe virtuele machine op, zorg dat deze een ip-adres krijgt in hetzelfde netwerk als de andere nodes en zorg dat de hostnamen en ip-adressen gekend zijn in al de /etc/hosts files.

Daarna ga je in Rancher naar cluster management, je selecteert de cluster waar je de node wilt toevoegen. Zoek naar het tabblad 'registration' en zorg dat enkel het vakje van worker is aangeduid. Onder deze mogelijkheden vind je nu een commando dat je dient in te geven in de worker node. Op deze manier zal deze virtuele machine zich instellen als een worker node in de cluster en zullen toekomstige pods hier ook in gedeployed worden. Met dit commando zal de agent zijn werk doen:

```
sudo docker run -d --privileged --restart=unless-stopped --net=host -v
/etc/kubernetes:/etc/kubernetes -v /var/run:/var/run  rancher/rancher-
agent:v2.6.12 --server https://172.26.52.20 --token
4jshlcln656rt2bgk8lzxpv6hkbmwlct98wfsmv9mhk7d5fp7bsp9s --ca-checksum
ef61331cdc40fada7de28ad51a7964fbbb9f85a73508033631c03195f2b758f8 --
worker
```

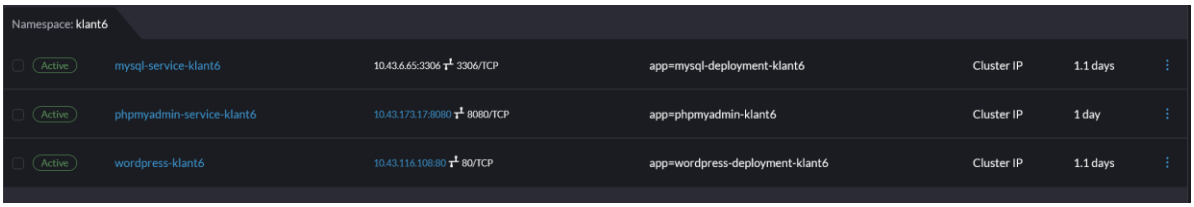
Apps/Charts: Via de Rancher interface kan je eenvoudig apps installeren die een meerwaarde kunnen bieden voor de cluster. Wij hebben Longhorn (app die de database voor de containers apache-php en mysql beheert) en Prometheus/Grafana (monitoring

tools voor de cluster) vrij snel kunnen installeren, wat deze manier van werken zeer gebruiksvriendelijk maakt. (Longhorn authors, 2023)



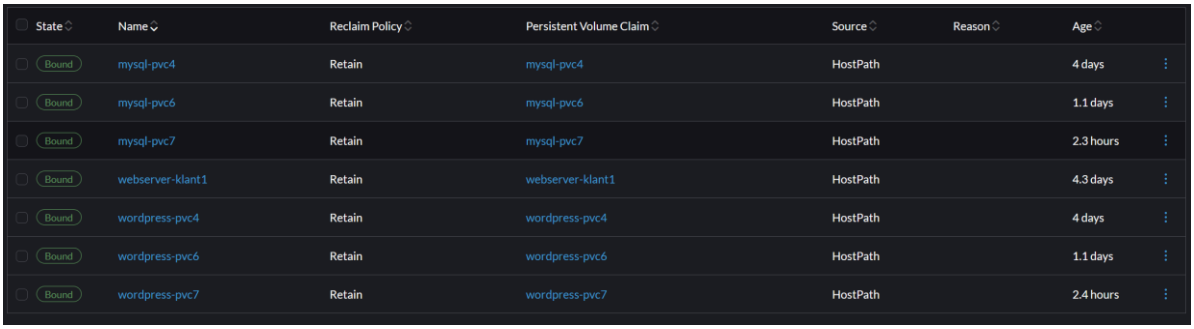
Figuur 32: apps & charts in Rancher (WLN Innovations)

Services: Services zijn essentieel voor het beheren van de recources binnen een namespace. Deze services zijn dan ook enkel zichtbaar voor al de recources die zich binnen die cluster bevinden.



Figuur 33: services van de cluster (WLN Innovations)

Persistent volumes: Bepaalde deployments, zoals mysql en apache-php, hebben een database nodig. Deze persistent volumes worden aangemaakt en toegewezen aan een specifieke pod binnen de namespace. Op deze manier gebruiken de pods de volumes om dataopslag te beheren.



Figuur 34: persistent volumes van de cluster (WLN Innovations)

3.3 Yaml-files deployments

Voor ons project hebben we besloten om het creëren van onze namespaces en bijbehorende objecten te doen via Yaml-files in plaats van de “edit config” van rancher.

Hieronder volgt een oplijsting van onze gebruikte Yaml-files die we nodig hebben bij het maken van een namespace voor een klant en bijbehorende objecten. Toch geven we graag eerst een informatieve uiteenzetting van een Yaml-file.

ApiVersion

De apiVersion specificeert de versie van de Kubernetes API die wordt gebruikt.

Kind

De kind specificeert het type object dat wordt gedefinieerd in het YAML-bestand. (Deployment, Service, VolumeClaim, PersistantVolumeClaim)

Metadata

De metadata sectie bevat metadata voor het object, zoals de naam en namespace.

Spec

De spec sectie bevat de specificatie van het object, zoals het aantal replica's, de selector en de template. Er worden 2 replicas gespecificeerd en de selector gebruikt de labels om de replica's te selecteren. De template bevat de specificaties voor de container.

In de container specificatie zijn er verschillende belangrijke onderdelen:

- image: de Docker image die wordt gebruikt voor de container.
- env: de omgevingsvariabelen die worden gebruikt door de container.
- ports: de poorten die worden blootgesteld door de container.
- volumeMounts: de volumes die worden gemount in de container.

3.3.1 Wat gebeurt er met onze deployment YAML-files

Basis beschrijving voor al de Deployment YAML-files, onder de screenshots gaan we per YAML-file dit iets uitgebreider uitlichten.

Bij het starten van de deployment-yaml zullen er pods opgezet worden, namelijk 2 (replicas: 2). Deze worden automatisch verdeeld op onze 2 verschillende worker nodes, hierdoor creeeren we al een vorm van redundantie.

We gebruiken ook de strategy: Recreate, dit wil zeggen dat wanneer er een nieuwe deployment wordt uitgerold deze onze oude pods eerst verwijderd en dan de nieuwe aanmaakt.

Verder wordt er een container aangemaakt met omgevingsvariabelen(container en env). De ports: containerport gaat de poort van de container blootstellen

Vervolgens wordt de volume gemount in het path dat gespecificeerd staat in “mountPath” /var/lib/….

3.3.2 Wat gebeurt er met onze service YAML:

De service gaat de pods selecteren aan de hand van het label app:naam als endpoints en stelt deze bloot aan poort 80. Kort gezegd: dit betekent dat verkeer dat toekomt op poort 80 door zal worden gestuurd naar de pods.

Als er meerdere pods zijn met dezelfde labels zal de service dit verkeer verdelen over die meerdere pods, dit kan helpen om de belasting van de pods te verminderen en ervoor te zorgen dat als er een pod wegvalt, we er nog altijd 1 hebben zodat de service beschikbaar blijft.

3.3.3 Wat gebeurt er met onze PersistentVolume YAML.

Er wordt een volume opgezet met een capaciteit van in ons geval 2gb. Deze volume kan gelezen en beschreven worden door meerdere pods tegerlijkertijd (ReadWriteMany). Dan wordt het volume opgeslagen op het pad op de host /mnt/klantnr. Als er dan een PersistentVolumeClaim gemaakt wordt door een pod dat overeen komt met de claimRef van de PersistentVolume yaml worden deze aan elkaar gekoppeld.

3.3.4 Wat gebeurt er met onze PersistentVolumeClaim YAML.

Qua definitie een beetje hetzelfde verhaal als de PersistentVolume YAML. Als deze YAML-definitie wordt gedeployed naar een Kubernetes-cluster, zal er onmiddellijk een PersistentVolumeClaim worden gemaakt met de opgegeven specificaties. Deze claim kan vervolgens worden gebruikt door pods om persistent storage aan te vragen.

3.3.4.1 Mysql-deployment

```yaml
---
#MYSQL Deployment deel.
apiVersion: apps/v1
kind: Deployment
metadata:
  labels:
#Hier gaan we de ".." vervangen door de nr van de klant.
    component: mysql-deployment-klant..
  name: mysql-deployment
  namespace: #HIER NAMESPACE INGEVEN
spec:
  replicas: 2
  selector:
    matchLabels:
#Hier gaan we de ".." vervangen door de nr van de klant.
      app: mysql-deployment-klant..
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
#Hier gaan we de ".." vervangen door de nr van de klant.
        app: mysql-deployment-klant..
    spec:
      containers:
        - env:
            - name: MYSQL_ROOT_PASSWORD
              value: wlninnovations
            - name: MYSQL_DATABASE
              value: wordpress
            - name: MYSQL_USER
              value: wordpress
            - name: MYSQL_PASSWORD
              value: wlninnovations
          image: mysql:latest
          imagePullPolicy: Always
          name: mysql-container
          ports:
            - containerPort: 3306
              protocol: TCP
          volumeMounts:
#Hier gaan we de ".." vervangen door de nr van de klant.
            - mountPath: /var/lib/mysql..
              name: mysql-persistent-storage
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      schedulerName: default-scheduler
      volumes:
        - name: mysql-persistent-storage
          persistentVolumeClaim:
#Hier gaan we de ".." vervangen door de nr van de klant.
            claimName: mysql-pvc..
---
```

Figuur 35: yaml-file mysql-deployment (WLN Innovations)

Hieronder beschrijven wij de belangrijkste delen code van de YAML file "mysql deployment" hierboven.

Het **apiVersion** veld geeft aan dat het Deployment-object wordt gemaakt met behulp van de apps/v1 API-versie.

De **kind** is ingesteld op "Deployment", wat aangeeft dat dit een Deployment-object is.

Onder **metadata** kunnen **labels** worden toegevoegd om het object te identificeren. Hier wordt een label genaamd "component" gebruikt met de waarde "mysql-deployment-klant..". Zoals te zien in de yaml-file screenshot van mysql deployment, vervangen we de ".." door de klantnr

De **name** geeft de naam van de Deployment aan, in dit geval "mysql-deployment".

De **namespace** moet worden ingevuld met de juiste namespace waarin het object wordt geïmplementeerd.

De **spec** bevat de specificatie van de Deployment, inclusief het aantal replica's dat moet worden uitgevoerd (hier 2).

Onder **selector** worden labels gedefinieerd om de pods te selecteren die door deze Deployment worden beheerd. Opnieuw moet ".." worden vervangen door het klantnummer.

De **strategy** bepaalt het implementatiestrategie, in dit geval "Recreate". (Kijk 3.3.1 wat dit doet)

template definieert de pod-template die wordt gebruikt om de pods te maken.

Onder **containers** wordt de configuratie van de MySQL-container gedefinieerd, inclusief omgevingsvariabelen voor wachtwoorden en database-instellingen.

image specificeert de gebruikte Docker-image, in dit geval "mysql:latest".

ports definieert de poort die moet worden blootgesteld binnen de pod.

Onder **volumeMounts** wordt een mountpad gespecificeerd voor het persistent opslaan van de MySQL-gegevens. Opnieuw moet ".." worden vervangen door het klantnummer.

volumes definieert de volumes die aan de pod worden gekoppeld, inclusief de persistent volume claim (PVC) voor het opslaan van de MySQL-gegevens. Ook hier moet ".." worden vervangen door het klantnummer.

Houd er rekening mee dat de ontbrekende waarden en placeholders moeten worden ingevuld met de juiste waarden, zoals het **klantnummer** en de **namespace**, voordat de Deployment kan worden geïmplementeerd.

3.3.4.2 Phpmyadmin deployment

```yaml
---
#PHPMYADMIN Deployment deel.
apiVersion: apps/v1
kind: Deployment
metadata:
  name: phpmyadmin-deployment
#Hier gaan we de ".." vervangen door de nr van de klant.
  namespace: klant..
spec:
  replicas: 2
  selector:
    matchLabels:
#Hier gaan we de ".." vervangen door de nr van de klant.
      app: phpmyadmin-klant..
  template:
    metadata:
      labels:
#Hier gaan we de ".." vervangen door de nr van de klant.
        app: phpmyadmin-klant..
    spec:
      containers:
#Hier gaan we de ".." vervangen door de nr van de klant.
      - name: phpmyadmin-klant..
        image: phpmyadmin/phpmyadmin
        env:
        - name: PMA_HOST
#Belangrijk, de value van de DB-HOST moet de naam zijn van de sql-service name.
          value: mysql-service-klant6
        ports:
        - containerPort: 8080
---
```

Figuur 36: yaml file PhpMyAdmin-deployment (WLN Innovations)

Het **apiVersion** veld geeft aan dat het Deployment-object wordt gemaakt met behulp van de apps/v1 API-versie.

De **kind** is ingesteld op "Deployment", wat aangeeft dat dit een Deployment-object is.

Onder **metadata** wordt de naam van de Deployment opgegeven als "phpmyadmin-deployment".

De **namespace** moet worden ingevuld met de juiste namespace waarin het object wordt geïmplementeerd. Ook hier moet ".." worden vervangen door het klantnummer.

De **spec** bevat de specificatie van de Deployment, inclusief het aantal replica's dat moet worden uitgevoerd (hier ook 2 zoals bij deployments, overal gebruiken wij er minstens 2).

Onder **selector** worden labels gedefinieerd om de pods te selecteren die door deze Deployment worden beheerd. Opnieuw moet ".." worden vervangen door het klantnummer.

template definieert de pod-template die wordt gebruikt om de pods te maken.

Onder **containers** wordt de configuratie van de PhpMyAdmin-container gedefinieerd.

name geeft de naam van de container aan, in dit geval "phpmyadmin-klant..". Zoals eerder genoemd ".." moet worden vervangen door het klantnummer.

image specificeert de gebruikte Docker-image, in dit geval "phpmyadmin/phpmyadmin".

env bevat de omgevingsvariabelen voor de container. Hier wordt de omgevingsvariabele PMA_HOST ingesteld op "**mysql-service-klant6**". Zorg ervoor dat de waarde van PMA_HOST overeenkomt met de naam van de SQL-service die wordt gebruikt. *Dit is wel belangrijk anders gaat hij de correcte database niet vinden.

ports definieert de poort die moet worden blootgesteld binnen de pod. Hier wordt de containerpoort 8080 blootgesteld.

3.3.4.3 Wordpress-deployment

```yaml
#WORDPRESS Deployment deel
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress-deployment
  #Correcte namespace invullen.
  namespace: klant6
spec:
  replicas: 2
  selector:
    matchLabels:
      #Klantnr aanpassen
      app: wordpress-deployment-klant6
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
      #Klantnr aanpassen
        app: wordpress-deployment-klant6
    spec:
      containers:
      - image: wordpress:latest
        imagePullPolicy: Always
        name: wordpress-container
        env:
        - name: WORDPRESS_DB_HOST
          #Belangrijk, de value van de DB-HOST moet de naam zijn van de sql-service name.
          value: mysql-service-klant6
        - name: WORDPRESS_DB_USER
          value: wordpress
        - name: WORDPRESS_DB_PASSWORD
          value: wlninnovations
        - name: WORDPRESS_DB_NAME
          value: wordpress
        ports:
        - containerPort: 80
          protocol: TCP
        resources: {}
        terminationMessagePath: /dev/termination-log
        terminationMessagePolicy: File
        volumeMounts:
        - mountPath: /var/www/html
          name: wordpress-persistent-storage
      dnsPolicy: ClusterFirst
      restartPolicy: Always
      schedulerName: default-scheduler
      securityContext: {}
      terminationGracePeriodSeconds: 30
      volumes:
      - name: wordpress-persistent-storage
        persistentVolumeClaim:
#Hier gaan we de ".." vervangen door de nr van de klant.
          claimName: wordpress-pvc..
```

Figuur 37: yaml-file Wordpress-deployment (WLN Innovations)

Het **apiVersion** veld geeft aan dat het Deployment-object wordt gemaakt met behulp van de apps/v1 API-versie.

De **kind** is ingesteld op "Deployment", wat aangeeft dat dit een Deployment-object is.

Onder **metadata** wordt de naam van de Deployment opgegeven als "wordpress-deployment".

De **namespace** moet worden ingevuld met de juiste namespace waarin het object wordt geïmplementeerd. Hier is de namespace ingesteld op "klant6".

De **spec** bevat de specificatie van de Deployment, inclusief het aantal replica's dat moet worden uitgevoerd (hier 2).

Onder **selector** worden labels gedefinieerd om de pods te selecteren die door deze Deployment worden beheerd. Hier wordt de label "app" gebruikt met de waarde "wordpress-deployment-klant6".

De **strategy** bepaalt het implementatiestrategie, in dit geval "Recreate.". (Kijk 3.3.1 wat dit doet)

template definieert de pod-template die wordt gebruikt om de pods te maken.

Onder **containers** wordt de configuratie van de WordPress-container gedefinieerd.

image specificeert de gebruikte Docker-image, in dit geval "wordpress:latest".

imagePullPolicy geeft aan dat de nieuwste versie van de image altijd moet worden opgehaald.

env bevat de omgevingsvariabelen voor de container, zoals het database-hostadres, gebruikersnaam, wachtwoord en naam van de database.

ports definieert de poort die moet worden blootgesteld binnen de pod. Hier wordt de containerpoort 80 blootgesteld.

volumeMounts specificeert het mountpad voor het persistent opslaan van de WordPress-gegevens.

dnsPolicy bepaalt de DNS-beleidsregel voor de pod.

restartPolicy geeft aan dat de pod altijd opnieuw moet worden gestart als deze wordt beëindigd.

schedulerName geeft de naam van de scheduler aan die wordt gebruikt om de pod te plannen.

3.3.4.4 Service voor mysql

```yaml
---
#MYSQL Service deel.
apiVersion: v1
kind: Service
metadata:
#Deze moet je gebruiken bij WORDPRESS_DB_HOST in het wordpress_deployment_deel & bij PMA_HOST in het phpmyadmin_deployment_deel.
  name: mysql-service-klant6
  namespace: #HIER NAMESPACE INGEVEN
spec:
  internalTrafficPolicy: Cluster
  ipFamilies:
    - IPv4
  ipFamilyPolicy: SingleStack
  ports:
    - port: 3306
      protocol: TCP
      targetPort: 3306
  selector:
#Hier gaan we de ".." vervangen door de nr van de klant.
    app: mysql-deployment-klant..
  sessionAffinity: None
  type: ClusterIP
---
#WORDPRESS Service deel.
apiVersion: v1
kind: Service
metadata:
#Hier gaan we de ".." vervangen door de nr van de klant.
  name: wordpress-klant..
  namespace: #HIER NAMESPACE INGEVEN
spec:
  internalTrafficPolicy: Cluster
  ipFamilies:
    - IPv4
  ipFamilyPolicy: SingleStack
  ports:
    - port: 80
      protocol: TCP
      targetPort: 80
  selector:
#Hier gaan we de ".." vervangen door de nr van de klant.
    app: wordpress-deployment-klant..
  sessionAffinity: None
  type: ClusterIP
---
```

Figuur 38: yaml-file mysql-service (WLN Innovations)

Het **apiVersion** veld geeft aan dat het Service-object wordt gemaakt met behulp van de **v1** API-versie.

- De **kind** is ingesteld op "Service", wat aangeeft dat dit een Service-object is.
- Onder **metadata** wordt de naam van de Service opgegeven als "mysql-service-klant6".
- De **namespace** moet worden ingevuld met de juiste namespace waarin het object wordt gemaakt.
- De **spec** bevat de specificatie van de Service.
- **internalTrafficPolicy** geeft aan hoe het interne verkeer wordt behandeld. Hier is het ingesteld op "Cluster", wat betekent dat verkeer binnen het cluster wordt afgehandeld.
- **ipFamilies** geeft de IP-families aan die worden ondersteund door de Service. Hier is alleen IPv4 ingesteld.

- **ipFamilyPolicy** bepaalt het IP-familiebeleid voor de Service. Hier is het ingesteld op "SingleStack", wat betekent dat er slechts één IP-familie wordt gebruikt.

- **ports** definieert de poortconfiguratie voor de Service. Hier wordt poort 3306 blootgesteld en doorgestuurd naar poort 3306 van de doelpods.

- **selector** bepaalt welke pods worden geselecteerd door deze Service. Er ontbreekt echter een specifieke labelselector in de gegeven code. De **selector** moet worden ingevuld met de juiste labels om de doelpods te selecteren

3.3.4.5 Service voor phpmyadmin

```yaml
---
#PHPMYADMIN Service deel.
apiVersion: v1
kind: Service
metadata:
  name: phpmyadmin-service-klant6
  namespace: klant6
spec:
  selector:
    app: phpmyadmin-klant6
  ports:
  - protocol: TCP
    port: 8080
    targetPort: 8080
---
```

Figuur 39: yaml-file PhpMyAdmin-service (WLN Innovations)

(screenshot van de PMA service yaml-file in virtual studio code)

- Het **apiVersion** veld geeft aan dat het Service-object wordt gemaakt met behulp van de **v1** API-versie.

- De **kind** is ingesteld op "Service", wat aangeeft dat dit een Service-object is.

- Onder **metadata** wordt de naam van de Service opgegeven als "phpmyadmin-service-klant6".

- De **namespace** moet worden ingevuld met de juiste namespace waarin het object wordt gemaakt. Dit verschilt per namespace.

- De **spec** bevat de specificatie van de Service.

- **selector** bepaalt welke pods worden geselecteerd door deze Service. Hier wordt de label "app" gebruikt met de waarde "phpmyadmin-klant6" om de pods te selecteren.

- **ports** definieert de poortconfiguratie voor de Service. Hier wordt poort 8080 blootgesteld en doorgestuurd naar poort 8080 van de doelpods.

3.3.4.6 Service voor Wordpress

```yaml
---
#WORDPRESS Service deel.
apiVersion: v1
kind: Service
metadata:
#Hier gaan we de ".." vervangen door de nr van de klant.
  name: wordpress-klant..
  namespace: #HIER NAMESPACE INGEVEN
spec:
  internalTrafficPolicy: Cluster
  ipFamilies:
    - IPv4
  ipFamilyPolicy: SingleStack
  ports:
    - port: 80
      protocol: TCP
      targetPort: 80
  selector:
#Hier gaan we de ".." vervangen door de nr van de klant.
    app: wordpress-deployment-klant..
  sessionAffinity: None
  type: ClusterIP
```

Figuur 40: yaml-file Wordpress-service (WLN Innovations)

- Het **apiVersion** veld geeft aan dat het Service-object wordt gemaakt met behulp van de **v1** API-versie.
- De **kind** is ingesteld op "Service", wat aangeeft dat dit een Service-object is.
- Onder **metadata** wordt de naam van de Service opgegeven als "wordpress-klant.." (hier moet je ".." vervangen door het klantnummer).
- De **namespace** moet worden ingevuld met de juiste namespace waarin het object wordt gemaakt.
- De **spec** bevat de specificatie van de Service.
- **internalTrafficPolicy** geeft aan hoe het interne verkeer wordt behandeld. Hier is het ingesteld op "Cluster", wat betekent dat verkeer binnen het cluster wordt afgehandeld.
- **ipFamilies** geeft de IP-families aan die worden ondersteund door de Service. Hier is alleen IPv4 ingesteld.
- **ipFamilyPolicy** bepaalt het IP-familiebeleid voor de Service. Hier is het ingesteld op "SingleStack", wat betekent dat er slechts één IP-familie wordt gebruikt.
- **ports** definieert de poortconfiguratie voor de Service. Hier wordt poort 80 blootgesteld en doorgestuurd naar poort 80 van de doelpods.
- **selector** bepaalt welke pods worden geselecteerd door deze Service. Hier wordt de label "app" gebruikt met de waarde "wordpress-deployment-klant.." (hier moet je ".." vervangen door het klantnummer) om de pods te selecteren.

- **type** geeft het Service-type aan. Hier is het ingesteld op "ClusterIP", wat betekent dat de Service alleen toegankelijk is binnen het cluster.

3.3.4.7 PersistentVolumeClaim voor mysql

```yaml
---
#MYSQL PersistenVolumeClaim deel.
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
#Hier gaan we de ".." vervangen door de nr van de klant.
  name: mysql-pvc..
  namespace: #HIER NAMESPACE INGEVEN
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 2Gi
  storageClassName: standard
  volumeMode: Filesystem
#Hier gaan we de ".." vervangen door de nr van de klant.
  volumeName: mysql-pvc..
---
```

Figuur 41: yaml-file mysql-pvc (WLN Innovations)

- Het **apiVersion** veld geeft aan dat het PersistentVolumeClaim-object wordt gemaakt met behulp van de **v1** API-versie.
- De **kind** is ingesteld op "PersistentVolumeClaim", wat aangeeft dat dit een Persistent Volume Claim-object is.
- Onder **metadata** wordt de naam van de PersistentVolumeClaim opgegeven als "mysql-pvc.." (hier moet je ".." vervangen door het klantnummer).
- De **namespace** moet worden ingevuld met de juiste namespace waarin het object wordt gemaakt.
- De **spec** bevat de specificatie van de PersistentVolumeClaim.
- **accessModes** geeft de gewenste toegangsmodi aan. Hier is het ingesteld op "ReadWriteMany", (kijk 3.3.3)
- **resources** bepaalt de gewenste bronnen voor de PersistentVolumeClaim. Hier wordt aangevraagd om 2 gigabyte aan opslagruimte.
- **storageClassName** geeft de naam van de opslagklasse aan. Hier is het ingesteld op "standard".
- **volumeMode** geeft de volumemodus aan. Hier is het ingesteld op "Filesystem".
- **volumeName** geeft de naam van het PersistentVolume aan dat aan deze claim wordt gekoppeld. Hier moet ".." worden vervangen door het klantnummer.

3.3.4.8 PersistentVolume voor Wordpress

```yaml
---
#WORDPRESS PersistentVolume deel.
apiVersion: v1
kind: PersistentVolume
metadata:
  #Hier gaan we de ".." vervangen door de nr van de klant.
  name: wordpress-pvc..
spec:
  accessModes:
    - ReadWriteMany
  capacity:
    storage: 2Gi
  claimRef:
    apiVersion: v1
    kind: PersistentVolumeClaim
    #Klantnr veranderen
    name: wordpress-pvc4
    namespace: #HIER NAMESPACE INGEVEN
  hostPath:
#Hier gaan we de ".." vervangen door de nr van de klant.
    path: /mnt/klant..
    type: ''
  persistentVolumeReclaimPolicy: Retain
  storageClassName: standard
  volumeMode: Filesystem
  #Hier gaan we de ".." vervangen door de nr van de klant.
  volumeName: wordpress-pvc..
---
```

Figuur 42: yaml-file Wordpress-pvc (WLN Innovations)

Het **apiVersion** veld geeft aan dat het PersistentVolume-object wordt gemaakt met behulp van de **v1** API-versie.

- De **kind** is ingesteld op "PersistentVolume", wat aangeeft dat dit een Persistent Volume-object is.
- Onder **metadata** wordt de naam van het PersistentVolume opgegeven als "wordpress-pvc.." (hier moet je ".." vervangen door het klantnummer).
- De **spec** bevat de specificatie van het PersistentVolume.
- **accessModes** geeft de gewenste toegangsmodi aan. Hier is het ingesteld op "ReadWriteMany". (kijk 3.3.3)
- **capacity** bepaalt de capaciteit van het PersistentVolume. Hier wordt aangegeven dat het volume 2 gigabyte aan opslagruimte heeft.

- **claimRef** verwijst naar de bijbehorende PersistentVolumeClaim. Hier wordt de API-versie, het soort object (kind), de naam van de claim ("wordpress-pvc4") en de namespace opgegeven. Zorg ervoor dat je de juiste namespace invult.

- **hostPath** definieert het hostpad op het fysieke systeem waar het volume wordt gecreëerd. Hier moet je ".." vervangen door het klantnummer en "/mnt/klant.." door het gewenste hostpad.

- **persistentVolumeReclaimPolicy** geeft het beleid voor het terughalen van het volume aan. Hier is het ingesteld op "Retain", wat betekent dat het volume behouden blijft, zelfs nadat het niet meer in gebruik is.

- **storageClassName** geeft de naam van de opslagklasse aan. Hier is het ingesteld op "standard".

- **volumeMode** geeft de volumemodus aan. Hier is het ingesteld op "Filesystem".

- **volumeName** geeft de naam van het PersistentVolume aan. Hier moet je ".." vervangen door het klantnummer.

3.3.4.9 PersistentVolume voor Mysql

```yaml
---
#MYSQL PersistentVolume deel.
apiVersion: v1
kind: PersistentVolume
metadata:
#Hier gaan we de ".." vervangen door de nr van de klant.
  name: mysql-pvc..
spec:
  accessModes:
    - ReadWriteMany
  capacity:
    storage: 2Gi
  claimRef:
    apiVersion: v1
    kind: PersistentVolumeClaim
    #Klantnr veranderen
    name: mysql-pvc4
    #Namespace veranderen naar juiste namespace
    namespace: klant4
  hostPath:
    path: /mnt/klant4
    type: ''
  persistentVolumeReclaimPolicy: Retain
  storageClassName: longhorn
  volumeMode: Filesystem
  volumeName: mysql-pvc6
---
```

Figuur 43: yaml-file mysql-pvc (WLN Innovations)

Het **apiVersion** veld geeft aan dat het PersistentVolume-object wordt gemaakt met behulp van de **v1** API-versie.

- De **kind** is ingesteld op "PersistentVolume", wat aangeeft dat dit een Persistent Volume-object is.

- Onder **metadata** wordt de naam van het PersistentVolume opgegeven als "mysql-pvc.." (hier moet je ".." vervangen door het klantnummer).

- De **spec** bevat de specificatie van het PersistentVolume.

- **accessModes** geeft de gewenste toegangsmodi aan. Hier is het ingesteld op "ReadWriteMany". (kijk 3.3.3)

- **capacity** bepaalt de capaciteit van het PersistentVolume. Hier wordt aangegeven dat het volume 2 gigabyte aan opslagruimte heeft.

- **claimRef** verwijst naar de bijbehorende PersistentVolumeClaim. Hier wordt de API-versie, het soort object (kind), de naam van de claim ("mysql-pvc..") en de namespace ("klant..") opgegeven.

- **hostPath** definieert het hostpad op het fysieke systeem waar het volume wordt gecreëerd. Hier is het pad "/mnt/klant4" en het type is leeg gelaten.

- **persistentVolumeReclaimPolicy** geeft het beleid voor het terughalen van het volume aan. Hier is het ingesteld op "Retain", wat betekent dat het volume behouden blijft, zelfs nadat het niet meer in gebruik is.

- **storageClassName** geeft de naam van de opslagklasse aan. Hier is het ingesteld op "longhorn".

- **volumeMode** geeft de volumemodus aan. Hier is het ingesteld op "Filesystem".

- **volumeName** geeft de naam van het PersistentVolume aan.

3.3.4.10 PersistentVolumeClaim voor Wordpress

```yaml
---
#WORDPRESS PersistentVolumeClaim deel.
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
#Hier gaan we de ".." vervangen door de nr van de klant.
  name: wordpress-pvc..
  namespace: #HIER NAMESPACE INGEVEN
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 2Gi
  storageClassName: standard
  volumeMode: Filesystem
  #Hier gaan we de ".." vervangen door de nr van de klant.
  volumeName: wordpress-pvc..
```

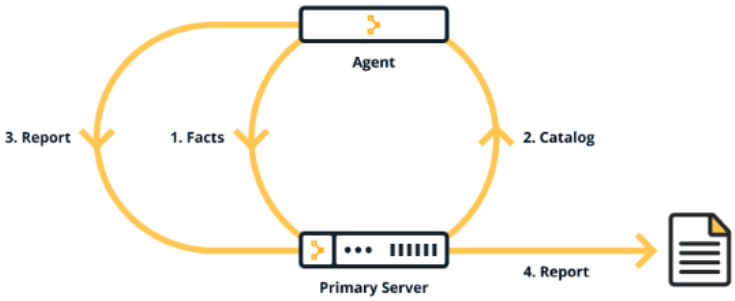
Figuur 44: yaml-file Wordpress-pvc (WLN Innovations)

- Het **apiVersion** veld geeft aan dat het PersistentVolumeClaim-object wordt gemaakt met behulp van de **v1** API-versie.
- De **kind** is ingesteld op "PersistentVolumeClaim", wat aangeeft dat dit een Persistent Volume Claim-object is.
- Onder **metadata** wordt de naam van de PersistentVolumeClaim opgegeven als "wordpress-pvc.." (hier moet je ".." vervangen door het klantnummer).
- De **spec** bevat de specificatie van de PersistentVolumeClaim.
- **accessModes** geeft de gewenste toegangsmodi aan. Hier is het ingesteld op "ReadWriteMany". (3.3.3)
- **resources** definieert de gewenste resource-eisen voor de PersistentVolumeClaim. Hier wordt aangegeven dat er 2 gigabyte aan opslagruimte wordt aangevraagd.
- **storageClassName** geeft de naam van de opslagklasse aan. Hier is het ingesteld op "standard".
- **volumeMode** geeft de volumemodus aan. Hier is het ingesteld op "Filesystem".
- **volumeName** geeft de naam van het bestaande PersistentVolume aan waaraan deze claim wordt gekoppeld. Hier moet je ".." vervangen door het klantnummer.

3.3.5 Puppet

Om de configuratie zo eenvoudig/centraal mogelijk te maken, is er, zoals eerder aangehaald, gekozen op Puppet te gebruiken. Puppet maakt gebruik van een "puppetmaster" dus de puppetserver, en trusted-nodes ook wel puppet agents genoemd. Op de puppetserver worden de manifestenbestanden geschreven en vervolgens gedeployed naar alle vertrouwde puppet-agents die aan de puppetmaster zijn

geconnecteerd via een certificaatverificatie. Dit maakt het configureren en up-to-date houden van alle nodes veel eenvoudiger. De communicatie tussen de master-node en trusted-nodes verloopt via een beveiligde SSH-verbinding, waardoor het verkeer tussen de nodes veilig is.



Figuur 45: visuele representatie Puppet (https://www.puppet.com/docs/puppet/6/puppet_overview.html)

Na we alles hebben geïnstalleerd, zowel de agents als de master, zijn we begonnen met het schrijven van configuraties voor ons hostingplatform. We hebben een nieuwe "environment" gemaakt namelijk "Projecthosting". In die environment hebben we 2 manifestbestanden gemaakt die zorgen voor de volgende configuraties; de vm's updaten, en als men een ssh verbinding maakt met 1 van onze VM's gaat er een boodschap getoond worden met een welkomsbericht, het IP-adres van de VM, en wanneer er voor het laatst een update is doorgevoerd .Puppet gaat deze 2 configuraties automatiseren met een runinterval van 1 uur. Dit wil zeggen dat elke uur, de agents gaan vragen aan de master of er nieuwe manifesten of wijzigingen zijn gebeurd. (Andrews, 2020)

```
[agent]
certname = test2ranmaster
runinterval = 1h
environment = Projecthosting
```

Figuur 46: puppet.conf file (WLN Innovations)

In het path: /etc/puppetlabs/code/environments/Projecthosting/modules gaat men 2 mappen tegenkomen, die respectievelijk de volgende namen hebben, "update" en "welkomstbericht". De manifestbestanden kan men dus in deze mappen vinden.

```
  GNU nano 6.2                                                    init.pp
class welkomsbericht {
  file { '/etc/motd':
    ensure  => file,
    owner   => 'root',
    group   => 'root',
    mode    => '0644',
    content => "# Welcome to the Project hosting VM\nServer Name: ${::fqdn}\nIP Address: ${::ipaddress}\nLast Updated: ${::timestamp}\n",
  }
}
```

Figuur 47: manifest-file Welkomstbericht (WLN Innovations)

Hierboven het manifestbestand van de configuratie voor het welkomstbericht.

Vervolgens gaan we in de main manifest file die men kan vinden langs volgend path: /etc/puppetlabs/code/environments/Projecthosting/manifests, al de puppet-agents benoemen die we willen aansturen met onze configuraties.

```
  GNU nano 6.2
node 'test2rancherbrowser', 'test2ranmaster', 'test2ranworkl' {
  include welkomsbericht
}
```

Figuur 48: welkomstbericht toegepast op agents (WLN Innovations)

Dan kunnen we de configuratie testen door middel van het volgende commando: sudo /opt/puppetlabs/bin/puppet agent –test

```
[sudo] password for wlni:
Info: Using environment 'Projecthosting'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales
Info: Caching catalog for test2ranworkl
Info: Applying configuration version '1686242299'
Notice: /Stage[main]/Welkomsbericht/File[/etc/motd]/ensure: defined content as '{md5}5fc
a959f5787b72f20031475b7a5b6ee'
```

Figuur 49: geslaagde configuratie welkomstbericht (WLN Innovations)

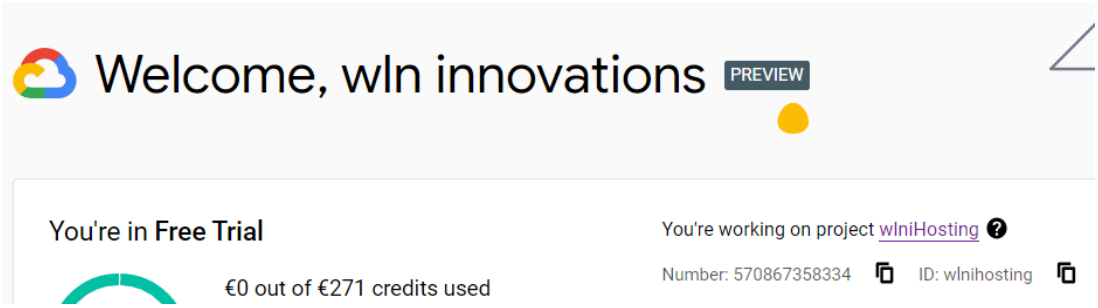
Hierboven zien we dat dit geen foutmeldingen geeft. En onderstaande foto ziet men de configuratie van “Welkomstbericht” Toegepast.

```
# Welcome to the Project hosting VM
Server Name: test2ranworkl.itfdc.local
IP Address: 172.26.52.102
Last Updated:
Last login: Thu Jun  8 19:06:39 2023 from 192.168.213.3
wlni@test2ranworkl:~$
```

Figuur 50: screenshot welkomstbericht (WLN Innovations)

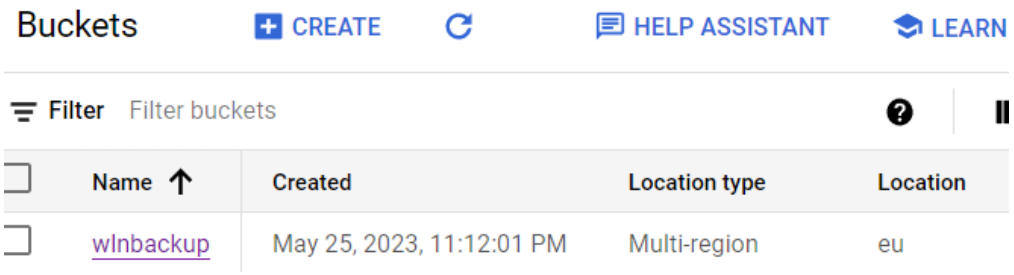
3.3.6 Back-up

Met het account innovationswln@gmail.com hebben wij een storage account genomen op Google Cloud. Het is straks de bedoeling dat we met dit service account gaan aanmelden op onze virtuele machine, om zo alles door te sturen naar onze Google Cloud back-up. Hieronder wordt uitgelegd hoe we dit gedaan hebben: (Cloud G. C., 2023)



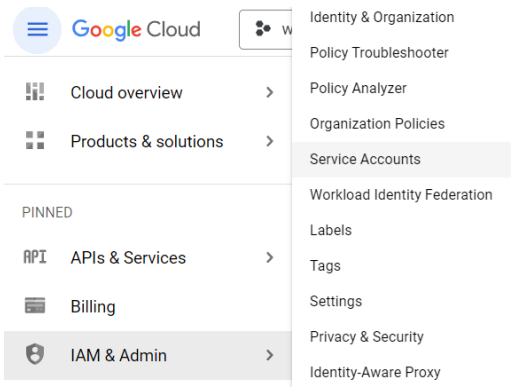
Figuur 51: dashboard Google Cloud project (WLN Innovations)

In deze cloud hebben we een bucket aangemaakt, waarin de backups worden opgeslagen. De naam van onze bucket is 'wlnbackup'. Hier willen we onze bestanden naartoe kopiëren.

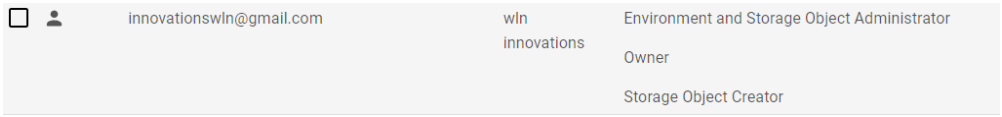


Figuur 52: Google Cloud bucket (WLN Innovations)

Cruciaal in de Google Cloud omgeving is dat je ook service accounts aanmaakt, die bepaalde rechten hebben (object_creator_storage) via de IAM & Admin. In 'Service Accounts' voeg je bepaalde gebruikers toe en geef je de juiste storage rechten. Wanneer je straks aanmeldt in de virtuele machine, zal je door deze rechten de gegevens kunnen overdragen.



Figuur 53: navigatie naar IAM & Admin (WLN Innovations)



Figuur 54: machtigingen account innovationswln@gmail.com (WLN Innovations)

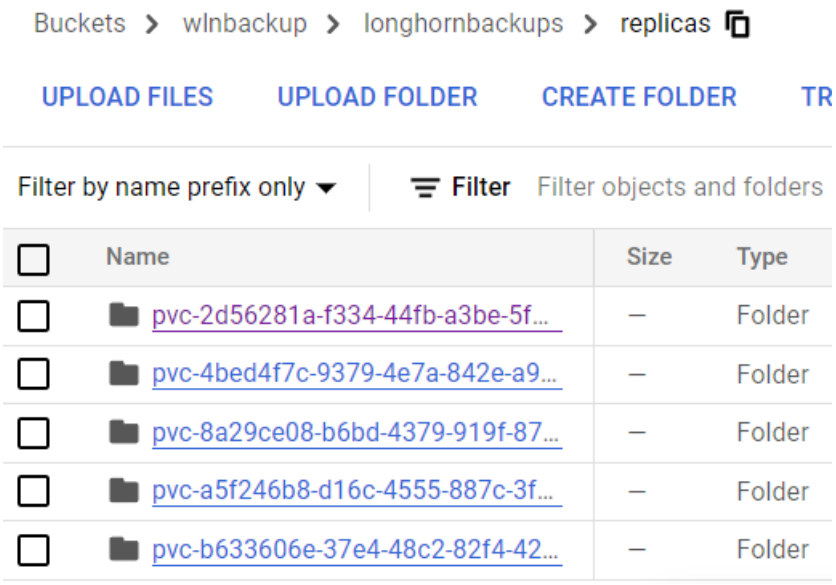
Bestanden kunnen doorgestuurd worden naar de back-up door in de virtuele machine aan te melden in het project en aan te melden als gebruiker innovationswln@gmail.com. Dit doe je door gsutil te installeren op de virtuele machine. Hiermee wordt de Google Cloud SDK (software development kit) geïnstalleerd. Eens de engine geïnstalleerd is, kan je het commando 'gcloud init' opgeven, waarmee je het project, de gebruiker en de region kunt instellen. Eens dit gebeurd is, kan je de gewenste bestanden doorsturen naar de bucket 'wlnbackup' in project 'wlninnovations'. Deze stap (en ook al de voorgaande stappen) dient wel uitgevoerd te worden als superuser (commando 'sudo -s'). Effectief doorsturen van bestanden doe je met gsutil:

gsutil -m cp -r /var/lib/longhorn/replicas gs://wlnbackup/longhornbackups/

Dit commando zal een kopie verzenden, inclusief al de submappen, van bestand /var/lib/longhorn/replicas. Hierin zitten al de persistent volumes van ons project. Dit wordt gekopieerd naar onze bucket in Google cloud en plaatst deze in de map 'longhornbackups'. Indien deze nog niet bestaat, wordt deze aangemaakt.

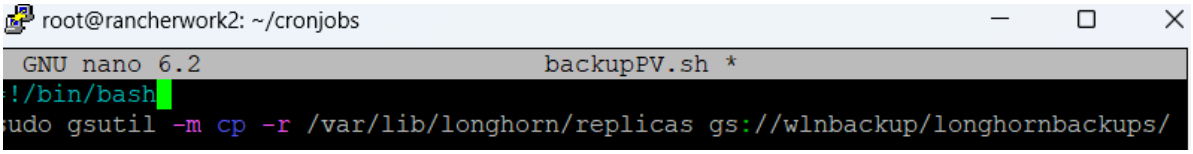
gsutil -m cp -r /home/klant1 gs://wlnbackup/backupklant1

Dit commando zorgt ervoor dat de homedirectories van onze klanten, waarmee zij verbinding maken via SFTP ook opgeslagen worden op onze back-up. Op deze manier zijn zij nooit uploads of documentatie kwijt en kunnen we die eenvoudig terug in hun home directory plaatsen.



Figuur 55: back-ups replicas (WLN Innovations)

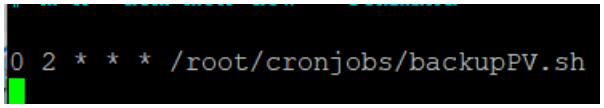
Om dit te automatiseren, gebruiken we een cronjob. Hiervoor dien je eerst een shellscript bestand aan te maken:



Figuur 56: shellscript back-up cronjob (WLN Innovations)

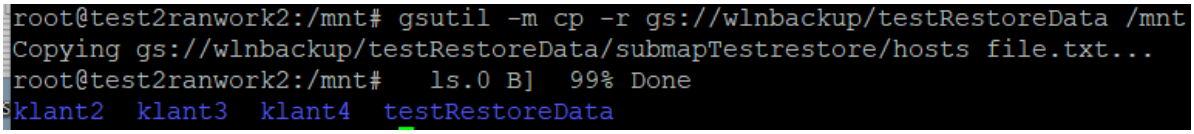
Vergeet het niet uitvoerbaar te maken: sudo chmod +x backup.sh

Daarna maak je de crontab zelf aan door het commando 'crontab -e' te geven. Dit zet je in jouw crontab bestand. Deze opdracht zal elke dag om 2 u 's nachts uitgevoerd worden.



Figuur 57: aanmaken crontab voor back-up (WLN Innovations)

De omgekeerde weg kunnen we uiteraard ook doen. De data van onze persistent volumes staat (in onze meest recente cluster) in de map /mnt en wordt elke nacht gekopieerd naar de bucket in onze Google Cloud omgeving. Stel nu dat de cluster of de virtuele machine volledig stuk gaat en je kunt ook niet meer beroepen op eerder gemaakte snapshots, dan kan je de data op Google Cloud terug opvragen, zodat deze terug in de map /mnt komen van een nieuwe virtuele machine/cluster. Dat kan je doen op deze manier:

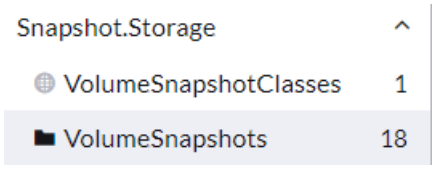


Figuur 58: restore data vanuit back-up (WLN Innovations)

Als test hadden wij een map aangemaakt 'testRestoreData' in onze Google Cloud bucket wlnbackup. In de Google Cloud SDK (Software Development Kit) zit een commando gsutil, die toelaat om gegevens tussen de bucket en de VM te transporteren. Bovenstaande commando zal meerdere bestanden tegelijk (-m) kopiëren (cp), inclusief al de submappen (-r) van de map gs://wlnbackup/testRestoreData naar de VM waar je het commando op geeft naar de map /mnt. Op deze manier kan je vrij snel de verloren data recupereren. (Google, 2023) (Cloud G. C., 2023)

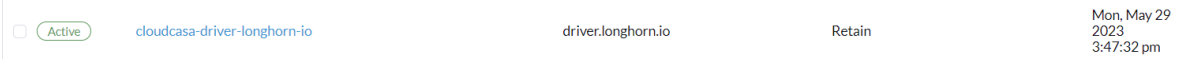
3.3.6.1 Snapshots door Cloudcasa

In Rancher kan je links in het menu bladeren door genomen snapshots in het snapshot.Storage tabblad:

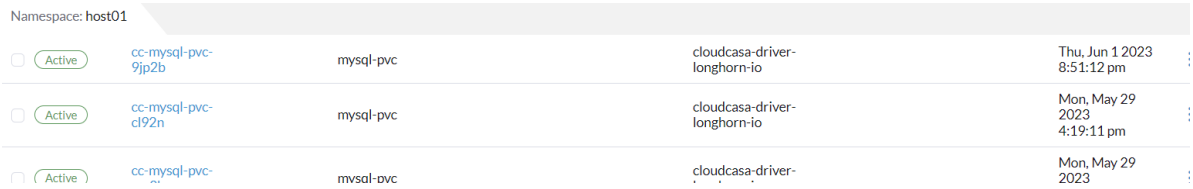


Figuur 59: tabblad snapshots (WLN Innovations)

Deze snapshots zijn gemaakt door een koppeling met Cloudcasa.io.



Figuur 60: Cloudcasa-driver (WLN Innovations)

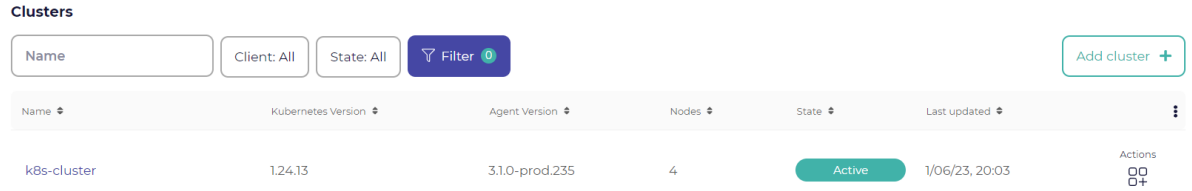


Figuur 61: Cloudcasa snapshots (WLN Innovations)

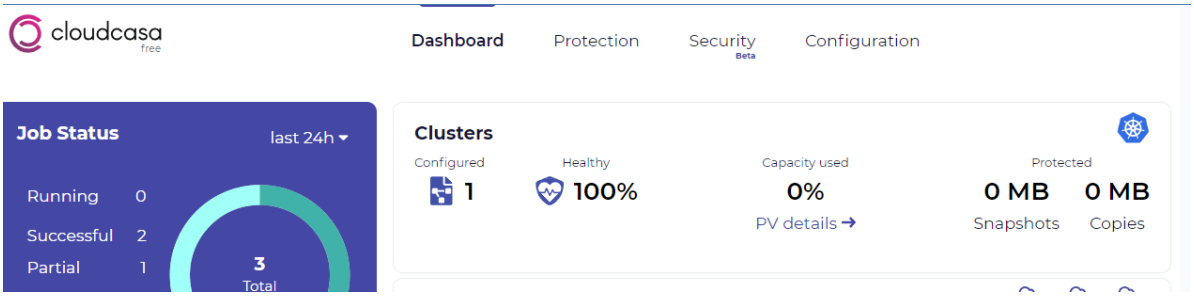
We hebben een account op Cloudcasa.io genomen, waarmee we onze cluster gekoppeld hebben. Om een bestaande cluster te verbinden met Cloudcasa, kies je eerst een geschikte clusternaam bij 'add cluster'. Daarna zal je een commando krijgen om een kubeclusteragent te installeren. Deze beheert de authenticatie en de communicatie tussen het Cloudcasa-platform en de cluster in Rancher. In de kubectl command line gebruik je dit commando om de cluster te koppelen:

kubectl apply -f https://api.cloudcasa.io/kubeclusteragents/LTHTY501BqNHV3zIHxap_8Fe9ycsBeWeomtAEfZ4MWo.yaml

Daarna zal je de cluster actief zien in jouw overzicht, samen met de beschikbare nodes in de cluster.

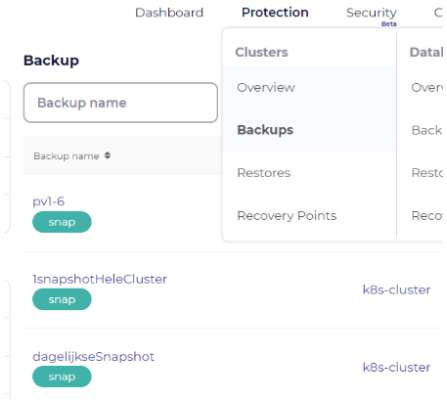


Figuur 62: gekoppelde cluster (WLN Innovations)



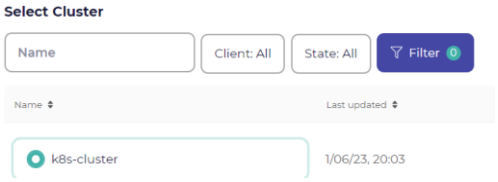
Figuur 63: Dashboard gekoppelde cluster (WLN Innovations)

Links in het dashboard zie je al enkele succesvolle snapshots. In het tabblad Protection-Backups vind je reeds afgewerkte snapshots. Hier kan je ook nieuwe snapshots plaatsen:

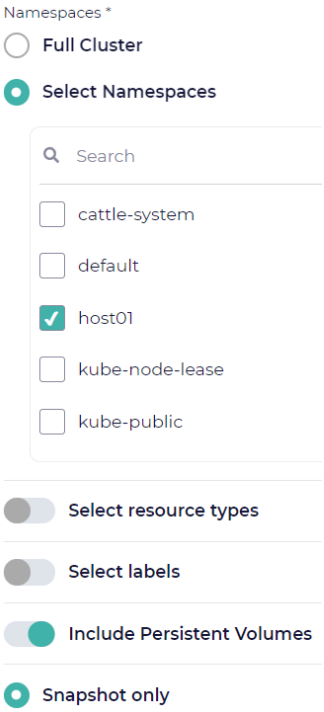


Figuur 64: back-ups cluster (WLN Innovations)

Om een nieuwe snapshot te maken, klik je rechts op 'Define backup'. Je selecteert de cluster waarvoor je een backup wilt nemen.

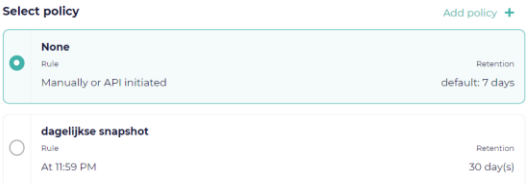


Figuur 65: clusterselectie voor backup (WLN Innovations)



Hierna kan je aanduiden of je een snapshot wenst van de volledige cluster of van bepaalde namespaces. Je kunt er ook voor kiezen om de persistent volumes wel of niet bij in de snapshot te zetten. Tenslotte kies je nog een moment: manual, waarbij je onmiddellijk een snapshot neemt of automatische snapshots, vb elke dag om 23.59u. Je kiest een geschikte naam voor de snapshot en kiest hoeveel dagen deze bewaard moet worden. Standaard staat deze op 7 dagen. (Catalogic, 2023)

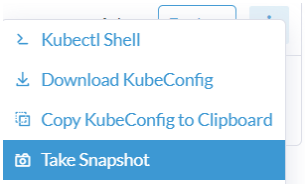
Figuur 66: selectie voor back-up (WLN Innovations)



Figuur 67: éénmalige of automatische back-up (WLN Innovations)

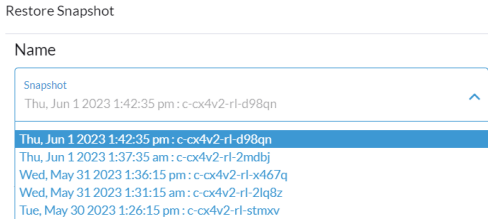
3.3.6.2 Snapshots Kubernetes cluster in Rancher

Figuur 68: cluster management (WLN Innovations)



Figuur 69: take snapshot (WLN Innovations)

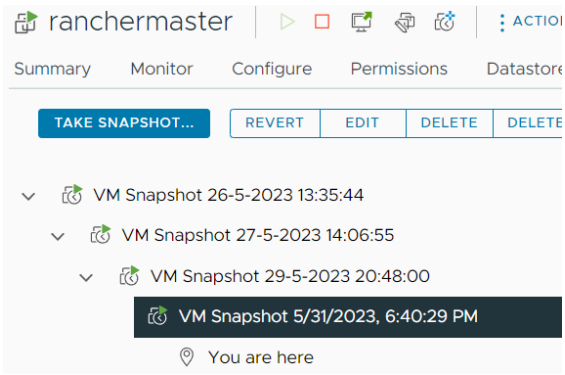
In Rancher kan je bij het Cluster management ook snapshots maken van de volledige cluster. Stel dat er iets misloopt, kan je hier ook de cluster terugzetten naar een vorige situatie. Wanneer je op restore klikt, zie je een overzicht van de verschillende snapshots.



Figuur 70: restore snapshot (WLN Innovations)

3.3.6.3 Snapshots van de VM's in Vsphere

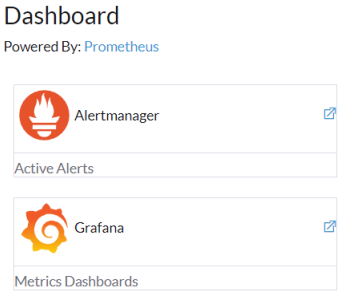
In het dashboard van Vsphere, waar al onze VM's op staan die op de server van Thomas More staan, kan je bij elke VM naar het tabblad 'snapshots' gaan. Op cruciale momenten in ons proces hebben wij telkens snapshots genomen, samen met een duidelijke omschrijving. Hiermee kunnen wij eenvoudig naar een voorgaande situatie gaan, waar alles nog werkte zoals het hoort. Op deze manier kunnen we in kleine stappen naar een werkend product gaan, waar trial & error soms tot complicaties leiden, maar ook eenvoudig teruggeschriefd kan worden. (VMware, 2023)



Figuur 71: snapshots virtuele machines Vsphere (WLN Innovations)

3.3.7 Monitoring

In Rancher kan je de volledige cluster monitoren. Hiervoor kijk je op het dasboard van Rancher aan de linkerkant en je kiest 'monitoring'. Daar krijg je het volgende dashboard te zien:

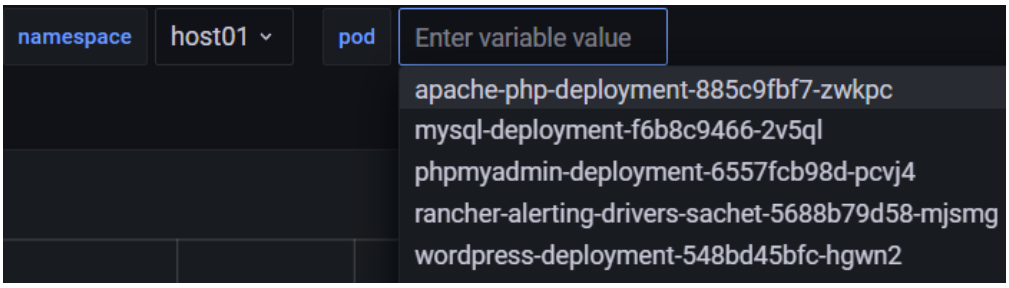


Figuur 72: monitoren met Grafana en Alertmanager (WLN Innovations)

Hier vind je de altermanager en Grafana. Met Grafana kan je inzoomen op bepaalde onderdelen van de cluster: CPU-gebruik, geheugengebruik, schijfgebruik…

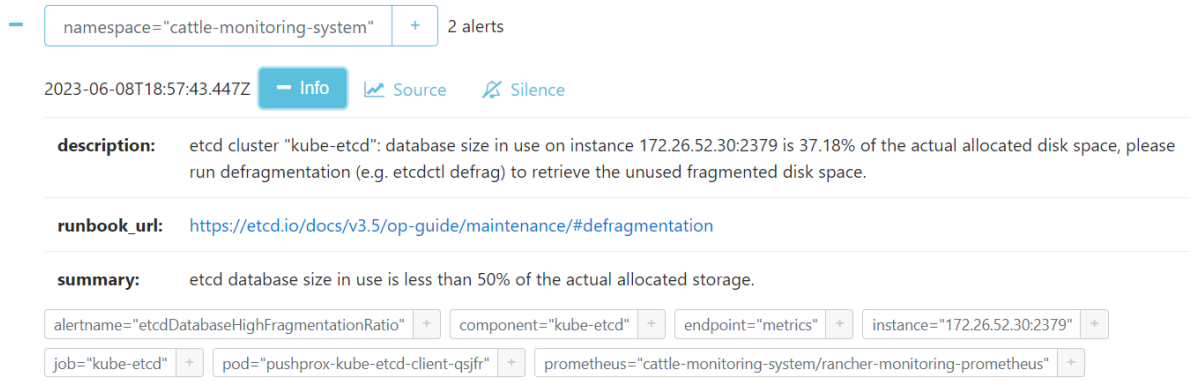
Figuur 73: monitoren van de Kubernetes cluster (WLN Innovations)

Je kan ook inzoomen op bepaalde namespaces en virtuele machines, daarbinnen kies je dan weer voor pods, persistent volumes, deployments etc…



Figuur 74: monitoren deployment binnen een namespace (WLN Innovations)

De alertmanager geeft jou signalen van problemen die zich voordoen op de cluster. Je kunt bepaalde alerts openvouwen met het plusteken. Hier vind je dan suggesties die je kunt toepassen en meer details over de verschillende componenten, zodat deze eenvoudig terug te vinden is.



Figuur 75: Alertmanager cluster (WLN Innovations)

3.3.8 Beveiliging

3.3.8.1 SFTP

Onze cliënten zullen via SFTP toegang krijgen tot een eigen /home map waar ze bestanden naar toe kunnen uploaden. Hiervoor dienen ze een programma te gebruiken zoals Filezilla. Dit programma zal moeten connecteren met de virtuele machine waar de /home mappen op staan. Uniek aan deze manier van werken, is dat elke gebruiker alleen maar in zijn eigen omgeving op de server kan werken en niet naar de hoger gelegen mappen zal kunnen in deze virtuele machine. Deze manier van werken, noemt chrooting en biedt een grote bescherming door middel van een afgeschermde omgeving wanneer een hacker toevallig weet binnen te geraken via de klant. We leggen in enkele stappen uit hoe we te werk zijn gegaan bij het creëren van deze /home-directories en het maken van beveiligde SFTP-connectie. (Accesspay, 2022)

Voor de volgende acties, kan je best aanmelden als root met het commando ‘sudo -s’. Om de gebruikers aan te maken, gebruiken we bijvoorbeeld het commando ‘adduser klant1’. Hierna zal je tweemaal het wachtwoord moeten opgeven. Je kunt eventueel een groep aanmaken voor al de sftpusers, om later eenvoudig bepaalde machtigingen te geven aan deze groep. Je voegt users aan deze groep toe met het commando ‘uermod -aG sftpusers klant1’. Met het commando ‘groups klant1’ kan je kijken of deze klant bij de groep behoort, met het commando ‘getent group sftpusers’ kan je zien welke users er allemaal bij de groep horen. Maak de home directories aan en zorg dat root eigenaar is van deze directories: ‘chown root:root /home/klant1’. Zorg dat de klanten de juiste

rechten krijgen: 'chmod 755 /home/klant2'. (nooit 777 op de rootmap, om schrijftoegang te vermijden). Wij hebben gekozen om in elke rootmap 2 mappen te zetten die de klanten dan wel kunnen gebruiken: 'uploads' en 'documentation'. Deze mappen dien je dan wel eigendom te maken van de klant, anders zal deze onvoldoende rechten hebben om hier aanpassingen in te doen. (Natarajan, 2012)

Belangrijk onderdeel bij SFTP is het feit dat we bij het installeren van onze virtuele machine al een ssh-server hebben geïnstalleerd. We moeten dus enkel nog de configuratie van sshd juist zetten, zodat elke user in een eigen home-directory terecht komt. Dit configuratiebestand vind je in /etc/ssh/sshd_config. In dit bestand moet al zeker de regel 'Subsystem sftp /usr/lib/openssh/sftp-server' uit commentaar gehaald worden. Daaronder moet je bij 'Match User klant1' enkele regels toevoegen. Hier vind je de regels voor klant1 en klant 2.

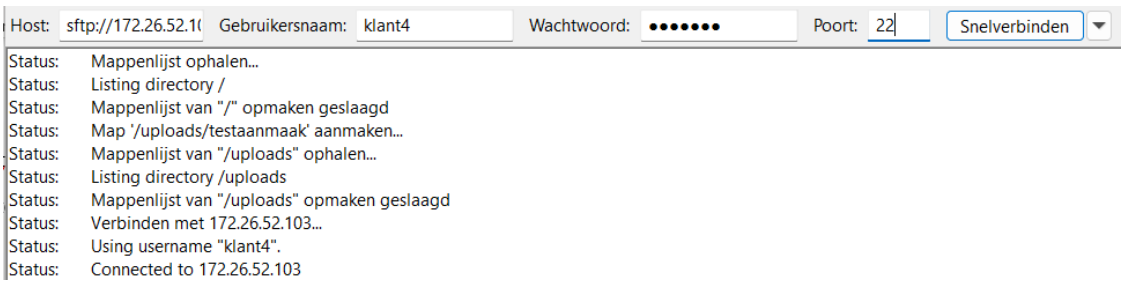
```
    # override default of no subsystems
Subsystem sftp  /usr/lib/openssh/sftp-server

Match User klant1
    ForceCommand internal-sftp
    ChrootDirectory /home/klant1
    AllowTcpForwarding no
    X11Forwarding no
Match User klant2
    ForceCommand internal-sftp
    ChrootDirectory /home/klant2
    AllowTcpForwarding no
    X11Forwarding no
```

Een woordje uitleg hierbij:

- ChrootDirectory: zorgt ervoor dat gebruikers in hun eigen chrootdirectory terecht komen en daar niet uit kunnen.
- ForceCommand internal sftp: gebruikers krijgen enkel SFTP toegang en geen shell toegang.
- AllowTcpForwarding: schakelt TCP-forwarding uit.
- X11Forwarding: schakelt X11-Forwarding uit.

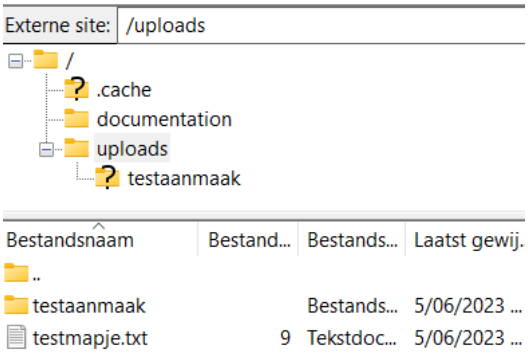
Om deze stappen actief te kunnen maken in de ssh-server, dien je deze eerst te herstarten: 'sudo service ssh restart' of 'sudo systemctl restart ssh'.

Aanmelden via Filezilla verloopt dan als volgt: Onze ssh-server draait op VM met ip-adres 172.26.52.103. Je geeft je gebruikersnaam en wachtwoord in en kiest poort 22, waar SFTP op werkt.

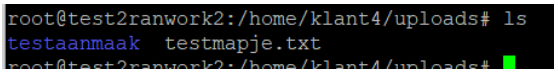


Figuur 76: aanmelden met Filezilla (WLN Innovations)

Het resultaat is nu dat elke gebruiker zijn eigen home-directory heeft. Hij/zij kan hier echter niet hoger in onze mappenstructuur duiken. Er kunnen enkel mappen aangemaakt worden of bestanden gekopieerd worden naar de mappen waar wij deze user eigenaar van hebben gemaakt, namelijk de mappen 'uploads' en 'documentation'. Als test hebben wij een map testaanmaak in de uploads aangebracht en ook een testmapje.txt gekopieerd.



Figuur 77: home-map via Filezilla (WLN Innovations)

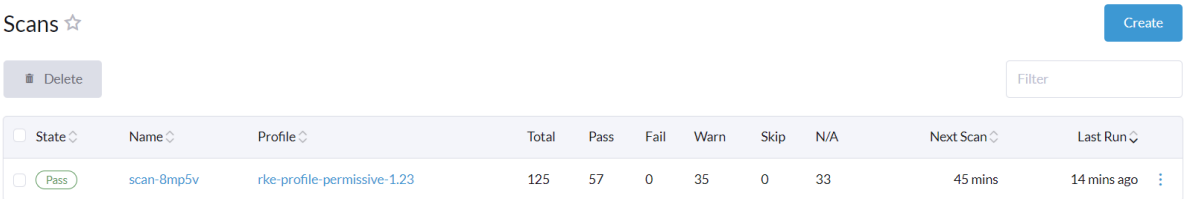


Figuur 78: home map in virtuele machine (WLN Innovations)

Eventueel kunnen wij ook nog op geregelde tijdstippen een backup nemen van de /home-directories van al onze users. Meer info over deze manier van werken vind je bij back-ups.

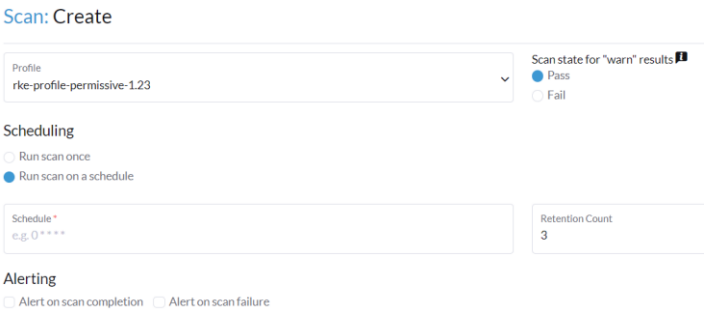
3.3.8.2 Pentesting: CIS Kubernetes Benchmark

In Rancher vind je links het onderdeel 'CIS Benchmark'. Hier kan je een nieuwe test manueel of automatisch inplannen. Klik hier rechts op 'create' om een nieuwe scan te configureren. (Suse, CIS scans, 2023)



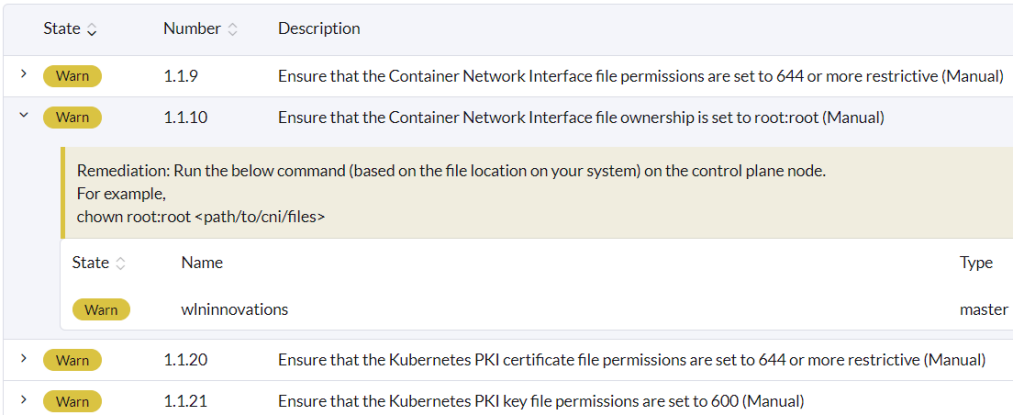
Figuur 79: scan met CIS benchmark (WLN Innovations)

In de volgende stap kan je kiezen voor 'Run scan once', dan zal deze op dat moment starten. Als je de optie 'Run scan on scedule' kiest, zal je een periode instellen zoals je een cronjob instelt. Vb. 0 0 * * 0 stelt voor: 0 minuten, 0 uur en dag van de week 0 (zondag). De scan zal dus om de week elke zondag lopen. Eventueel kan je nog een alert instellen, dat een bericht doorstuurt als de scan is afgerond. In dat geval moet je bij alertmanager ook wel bepaalde configuratie hebben ingesteld zoals, e-mail of telefoonnummer.



Figuur 80: scan instellen op regelmatige basis (WLN Innovations)

Een voorbeeld hoe een veiligheidssuggestie er uit zien, waar we eentje in detail geopend hebben:

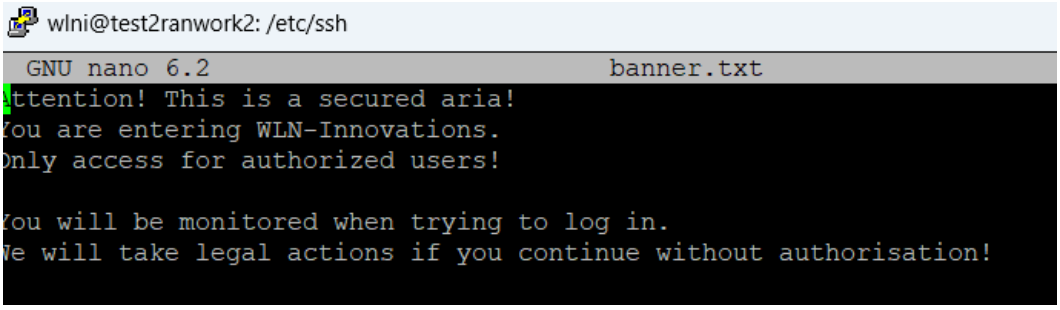


Figuur 81: veiligheidssuggestie uit scan CIS Benchmark (WLN Innovations)

3.3.8.3 Banner

Om een nieuwe banner in te stellen, neem je de volgende stappen (Kiarie, 2021):

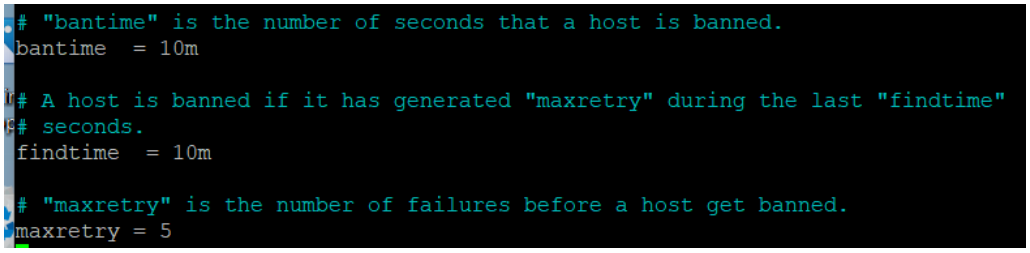
- In de file /etc/ssh/sshd_config controleer je of de volgende regel uit commentaar staat: Banner /etc/ssh/banner.txt
- In de file banner.txt typ je jouw gewenste tekst die je wilt vermelden wanneer er ssh-connectie wordt gemaakt.
- Herstart tenslotte jouw ssh-server: sudo service ssh restart



Figuur 82: ssh banner (WLN Innovations)

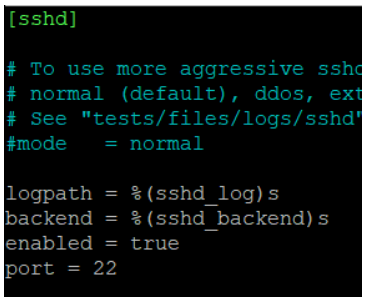
3.3.8.4 Fail2ban

Om de fail2ban te activeren, zorg je dat fail2ban geïnstalleerd is. Dit kan manueel met 'sudo apt install fail2ban' of je kunt dit instrueren via Puppet. Om een nieuw jail-bestand aan te maken, kopieer je best in de map /etc/fail2ban het bestand jail.conf naar vb jail.local. Laat het jail.conf als origineel bestand staan. 'sudo cp jail.conf jail.local'. Pas in dit bestand enkele dingen aan die je wenst in te stellen. Belangrijke opmerking hier: Wat je in default toepast, wordt aan al de vernoemde onderdelen aangewezen. Wij hebben in onderstaand voorbeeld, de bantime en de maxretry aangepast in [default] (hoe lang krijgt een ip-adres geen ssh-toegang meer na 5 onjuiste aanmeldpogingen) (DigitalOcean, 2022)



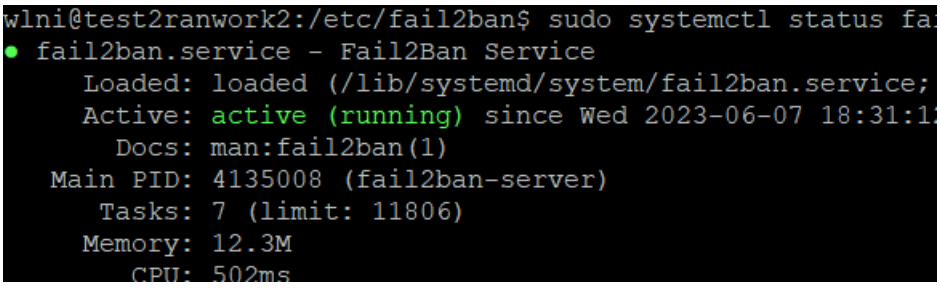
Figuur 83: configureren fail2ban (WLN Innovations)

Wil je dan een jail activeren bij een bepaald onderdeel, zet je bij dit onderdeel de enabled = true. Alles wat je hier specifiek configureert, overschrijft de default-instellingen.



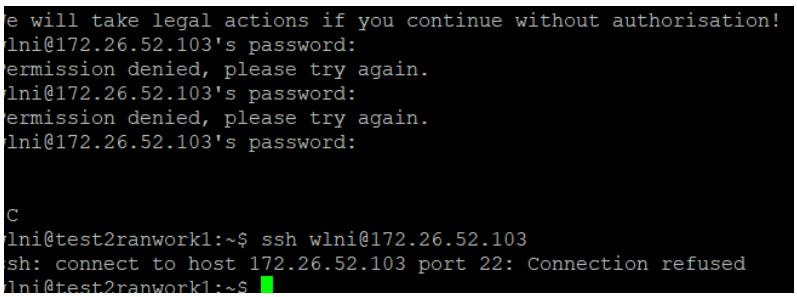
Figuur 84: activeren van een jail via fail2ban (WLN Innovations)

Enable en start de fail2ban service: sudo systemctl enable fail2ban + sudo systemctl start fail2ban. Controleer daarna of de fail2ban service actief is: sudo systemctl status fail2ban. Eventuele foutmeldingen zullen hieronder te komen staan. Test nu jouw configuratie.



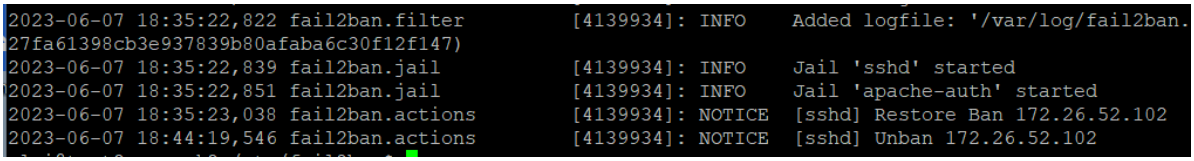
Figuur 85: controleren van activatie service (WLN Innovations)

Door enkele foutieve aanmeldpogingen te doen achter elkaar, kan je uittesten of de fail2ban service zijn werk doet. Zoals je kunt zien op de volgende screenshot, krijg ik na enkele pogingen een connection refused. Op dit moment zal ik 10 minuten geen toegang meer krijgen.



Figuur 86: testen van implementatie fail2ban (WLN Innovations)

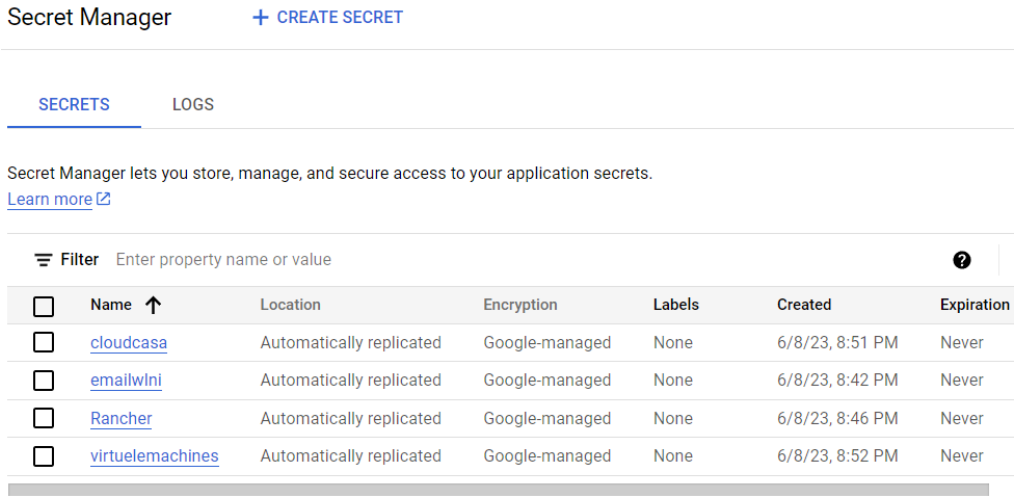
In de logfiles /var/log/fail2ban.log kan je ook info vinden over geïmplementeerde acties of aanmeldpogingen. In onderstaande screenshot zie je dat het pad naar de logfiles is ingesteld, dat de instellingen zijn toegepast op ssh(d) en op apache, je ziet ook dat ip-adres 172.26.52.102 een 10 minuten geblokkeerd geweest is.



Figuur 87: logfiles fail2ban (WLN Innovations)

3.3.8.5 Wachtwoordkluis

Wanneer je aanmeldt op console.cloud.google.com, kan je al jouw wachtwoorden opslaan in de secret manager. In het navigatiemenu zoek je naar 'beveiliging' of 'security' en daarna 'secret manager'. Hier vind je al de ingestelde wachtwoorden van het project wlniHosting. (Cloud G. , 2023)

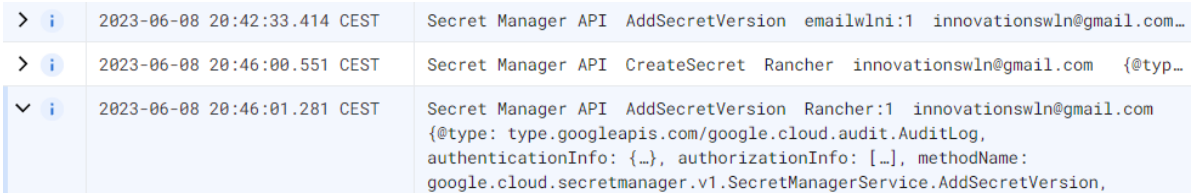


Figuur 88: wachtwoordkluis op Google Cloud (WLN Innovations)

Wanneer je hier op 'create secret' klikt, kan je een nieuw secret aanmaken. Je kiest een naam voor jouw geheim. Er zijn nu twee opties:

- Een geheim rechtstreeks geïmporteerd uit een bestand.
- Een geheim dat je hier zelf typt.

De wachtwoorden zijn geëncrypteerd door Google zelf met een CRC-32C checksum, zo wordt de data-integriteit bewaard. Toegang tot deze wachtwoordkluis en eventuele bewerkingsrechten worden beheerd door de IAM service accounts. In de logfiles van de secret manager kan je ook zien wie er aanpassingen gedaan heeft aan het wachtwoordbeheer.

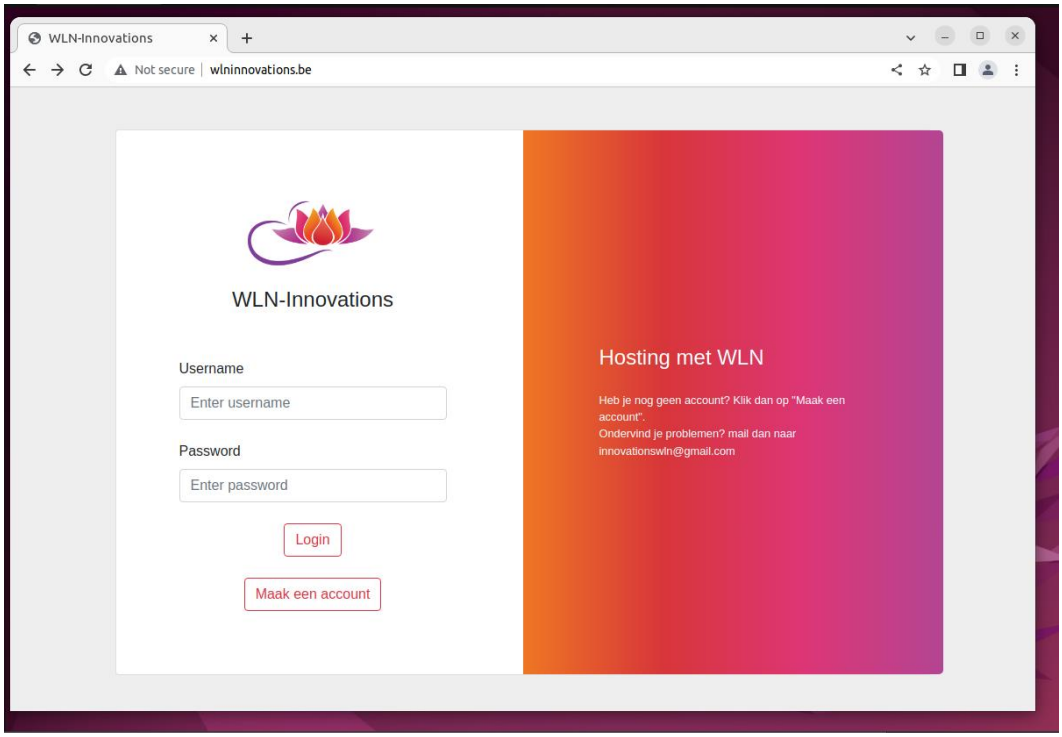


Figuur 89: logfiles wachtwoordmanager (WLN Innovations)

3.4 Handleiding voor gebruikers

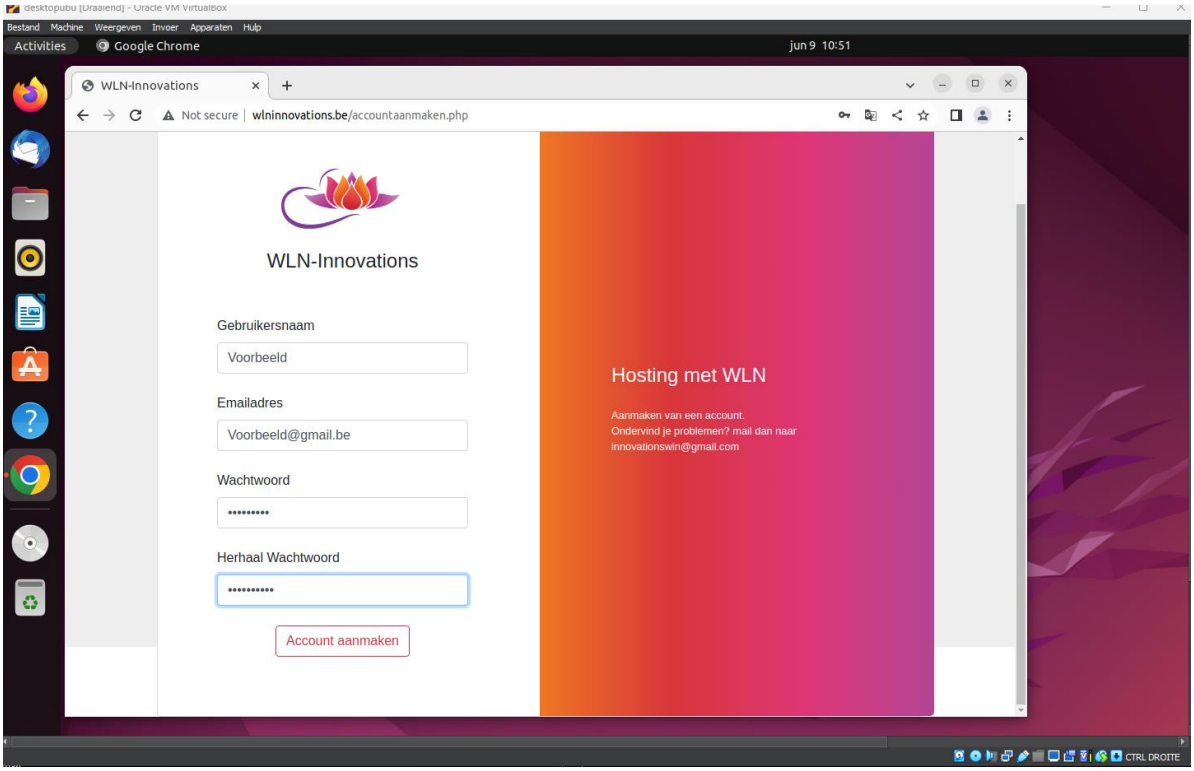
In dit hoofdstuk zal uitgelegd worden welke stappen er ondernomen moeten worden om zelf een website te maken.

Willen jullie gebruik maken van onze webhosting services? Surf dan naar onze hoofdpagina www.wlninnovations.be.



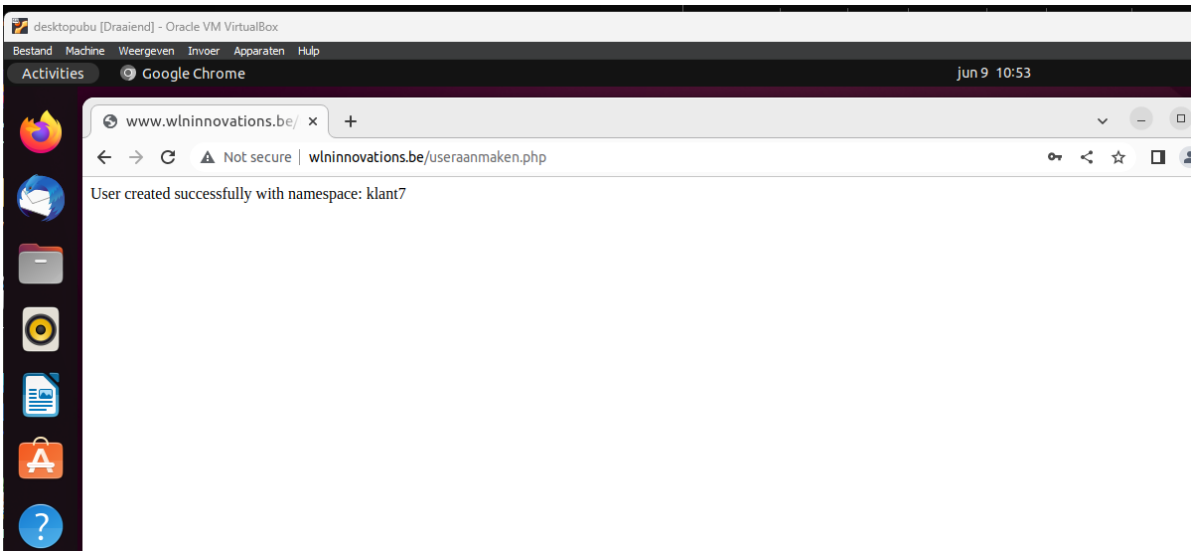
Figuur 90: aanmeldpagina www.wlninnovations.be (WLN Innovations)

Hier zal je de kans krijgen om in te loggen met je gebruikersnaam en wachtwoord. Heb je nog geen gebruikersnaam en wachtwoord? Klik dan op “maak een account” en vul je gegevens waarheidsgetrouw in.



Figuur 91: account aanmaken (WLN Innovations)

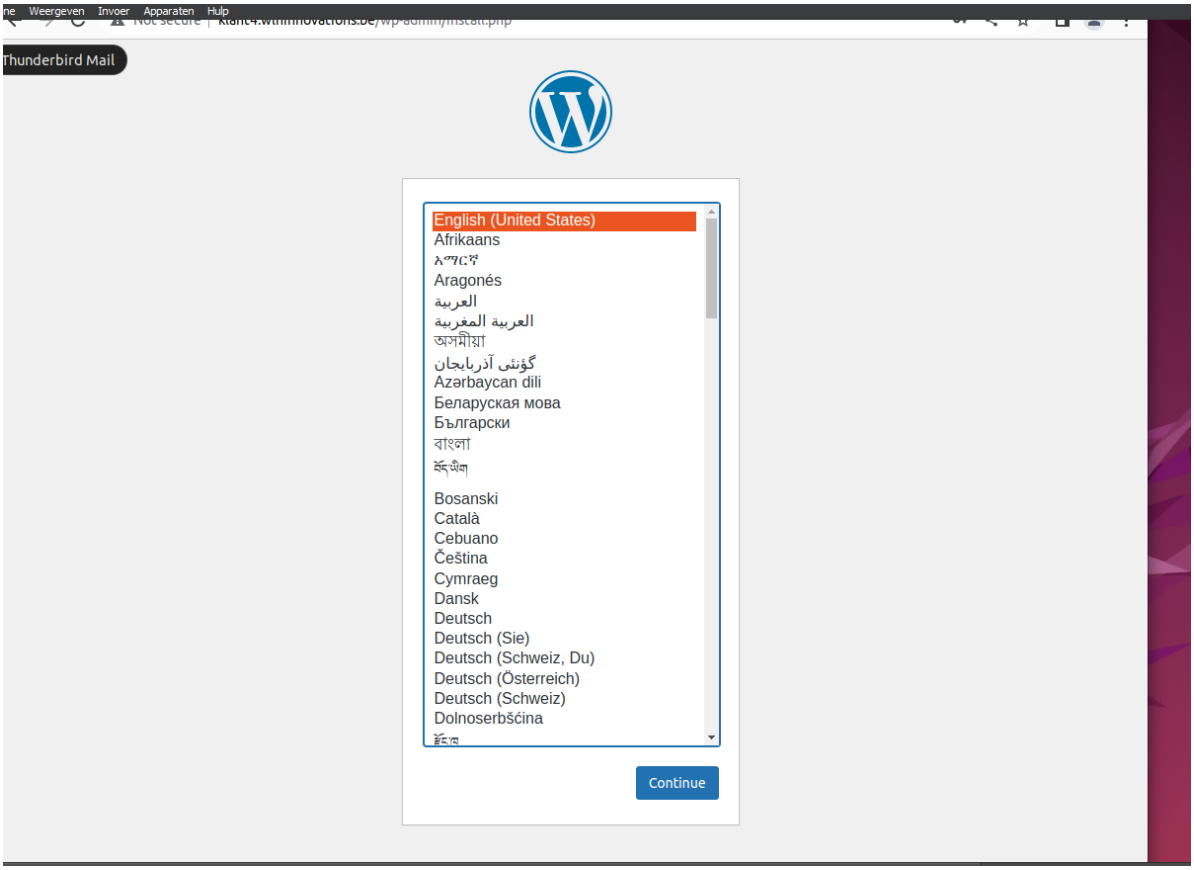
Na dat je je gegevens hebt ingevuld, dien je op “account aanmaken” te klikken. Je zal naar een externe pagina gestuurd worden waar je de melding krijgt dat dit gelukt is. Indien hier errors optreden, dient er contact opgenomen te worden met innovationswln@gmail.com. Wij helpen je zo spoedig mogelijk verder.



Figuur 92: succesvolle activatie account (WLN Innovations)

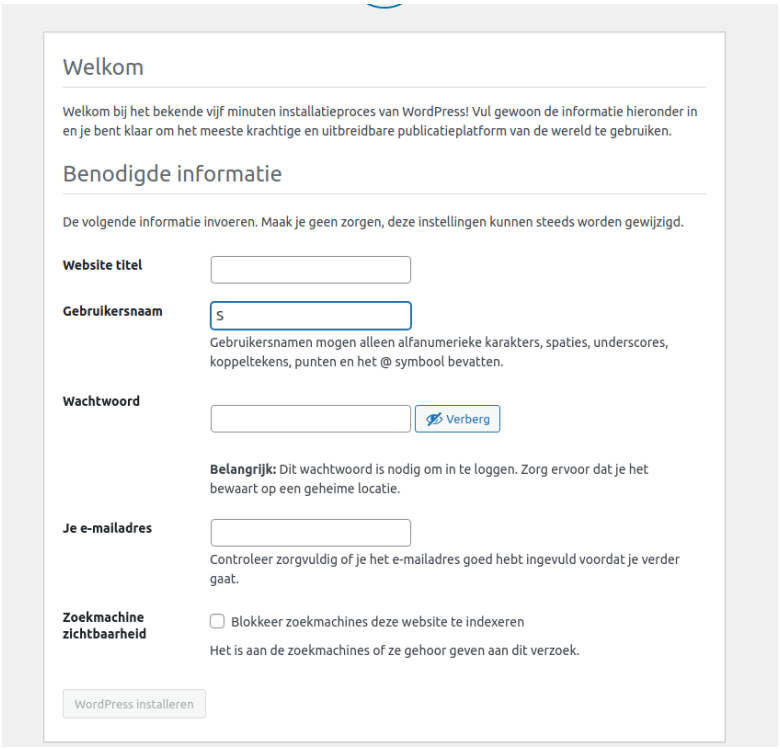
Nadat je deze melding gekregen hebt, kan je jezelf inloggen op de hoofdpagina met de gegevens van het account dat jij hebt aangemaakt. Surf terug naar de hoofdpagina en log jezelf in.

Je komt nu op je website terecht, kies de juiste taal en klik op “continue”.



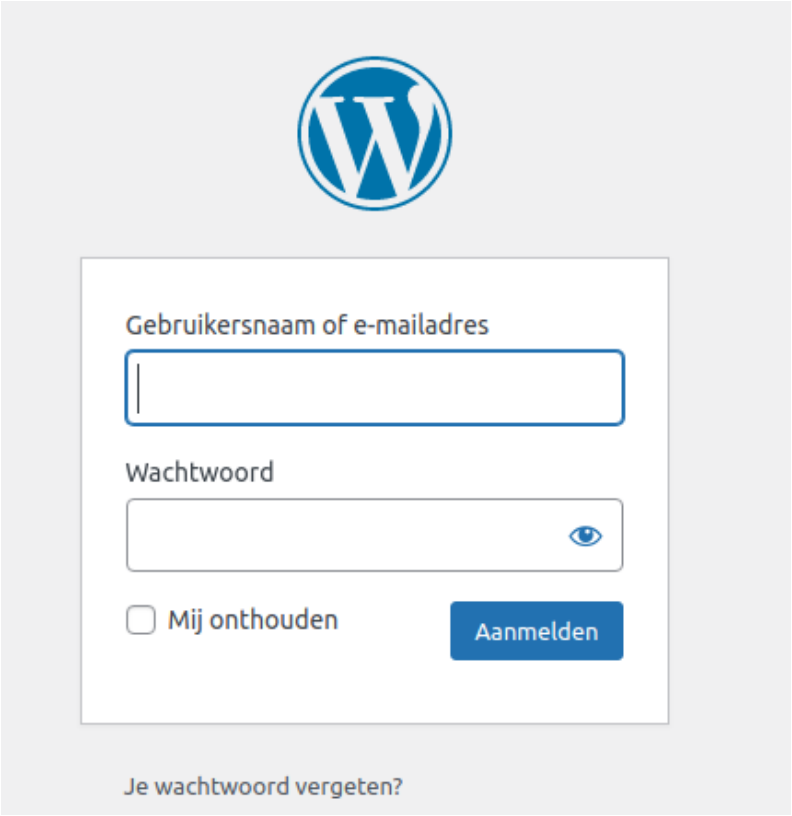
Figuur 93: index.php Wordpress (WLN Innovations)

Vul hierna onderstaand document in. Het is belangrijk dat je deze gegevens goed bewaart. Controleer ook zeker je e-mailadres.



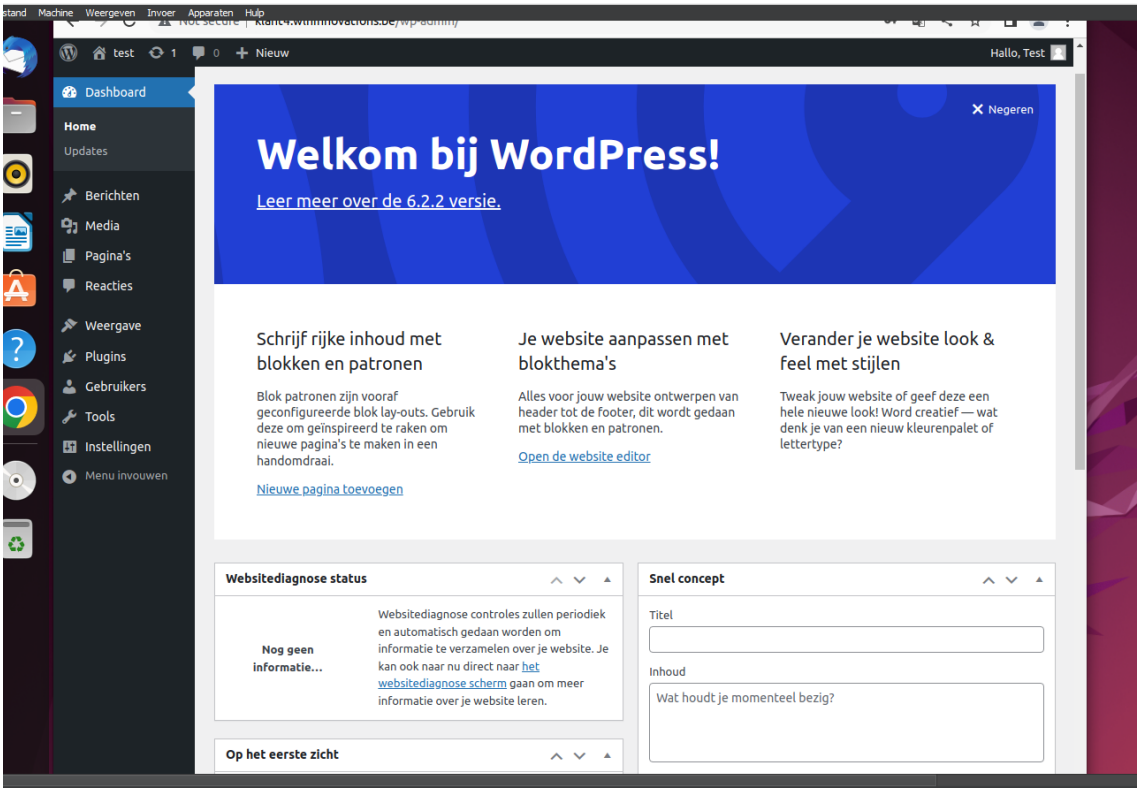
Figuur 94: installatieproces Wordpress (WLN Innovations)

Meld je aan met je gegevens:



Figuur 95: aanmelden in Wordpress (WLN Innovations)

Hierna word je doorverwezen naar de volgende pagina.



Figuur 96: welkomstpagina Wordpress (WLN Innovations)

We hechten enorm veel waard aan de klanten en willen hierdoor ook zorgen dat je de beste ervaring hebt bij het vormgeven van je website. Daarom willen we je graag doorverwijzen naar de startpagina van WordPress, waar je een schat aan informatie en documentatie zult vinden om je te helpen bij het ontwerpen en aanpassen van jouw website.

Op de startpagina van WordPress vind je gedetailleerde handleidingen, stapsgewijze instructies en waardevolle tips om het maximale uit het WordPress-platform te halen. Of je nu een geschikt thema wilt kiezen dat perfect bij jouw bedrijf of persoonlijke stijl past. De kleuren, lettertypen en lay-out wilt aanpassen, boeiende inhoud wilt toevoegen zoals teksten, afbeeldingen,… of handige functies en plugins wilt implementeren, de documentatie op de startpagina van WordPress zal je begeleiden bij elke stap van het proces.

We willen je aanmoedigen om de tijd te nemen om de documentatie zorgvuldig door te nemen. Hierdoor krijg je de volledige controle over het ontwerp en de functionaliteit van jouw website. Ontdek de uitgebreide mogelijkheden en flexibiliteit van WordPress, en laat jouw creativiteit de vrije loop om een website te creëren die helemaal bij jouw smaak, behoeften en doelen past.

Aarzel niet om contact met ons op te nemen als je vragen hebt of als je we ergens mee kunnen helpen. Wij staan altijd klaar om je te ondersteunen bij het realiseren van jouw website.

WordPress website: https://wordpress.com/nl/

(WordPress, 2023)

4 OPS REPORT CARDS

In dit hoofdstuk zullen worden al de gekozen OPS report cards neergeschreven. Ook zullen de geïmplementeerde cards plus documentatie vermeld worden.

4.1 Geselecteerde Ops report cards

A. Public Facing Practices

4.1.1 2. Are "The 3 empowering policies" defined and published?

- Hoe kunnen klanten ons contacteren wanneer er vragen zijn?
- De definitie van een noodgeval
- Hoe worden klanten geholpen, wanneer en wie

4.1.2 3. Does the team record monthly metrics?

- Maandelijks data verzamelen en hier ook nuttig mee omgaan

B. Modern Team Practices

4.1.3 5. Do you have a password safe?

- Zorgen dat onze wachtwoorden op een veilige plaats staan en niet "in plain tekst"

4.1.4 9. Does your team write "design docs?"

- "Eerst denken en dan doen" principe, noteren wat we willen doen en hoe we dit gaan uitvoeren

C. Operational Practices

4.1.5 12. Does each service have appropriate monitoring?

- Voorzien van een monitoring tool zodat we meer controle hebben over wat er juist gebeurt met onze systemen

4.1.6 15. Do roll-outs to many machines have a "canary process"?

- Eén, meerdere, allemaal "Kanarie techniek"

D. Automation Practices

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

- Onze virtuele machines gaan we beheren met behulp van Ansible

E. Fleet Management Processes

4.1.8 19. Is there a database of all machines?

- Een inventaris van al onze gebruikte machines

4.1.9 20. Is OS installation automated?

- Voorkomen van veel problemen door de installatie te automatiseren door Ansible

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

- Regelmatig zoeken naar updates en updaten via Ansible

F. Disaster Preparation Practices

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

- We gaan via Kubernetes aan load balancing en redundantie doen. Zodanig dat als bij 1 Docker container iets misloopt, we nog steeds een andere Docker container hebben lopen en de klant dus nog aan zijn/haar website kan.

4.1.12 25. Are your backups automated?

- Automatisatie van backups voorzien zodat er geen data verloren kan gaan. Dit gebeurt via cronjobs in de virtuele machines naar onze Google Cloud Storage Bucket 'wlnbackup'.

4.1.13 27. Do machines in your date canter have remote power/console access?

- We maken gebruik van een VPN tool om verbinding te maken met onze Virtual Machines

G. Security Practices

4.1.14 30. Do you submit to periodic security audits?

- Testen van security: Op regelmatige basis worden beveiligingsscans genomen van onze cluster om te controleren of er mogelijke beveiligingsrisico's aanwezig zijn in onze virtuele machines.

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

- Als we alle gebruiker opslaan in één database, kunnen er zonder problemen meerdere gebruikers gedeactiveerd worden binnen 1 uur

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

- Hiervoor moeten we noteren waar alle paswoorden staan, zodat dit bestand kan geopend worden en er snel kan gehandeld worden indien nodig

4.2 Hoe zijn de Ops report cards geïmplementeerd?

A. Public Facing Practices

4.2.1 2. Are "The 3 empowering policies" defined and published?

4.2.1.1 Hoe en wanneer bieden we ondersteuning?

In een ideaal scenario waar ons bedrijf bestaat, kunnen klanten ons contacteren vanaf 8u30 tot 17u00. Er zou ook een 24/7 wachtdienst voorzien worden voor zeer dringende zaken (zoals het vermoeden van een website die gehackt is/wordt).

Verder kunnen klanten ons altijd bereiken op onze email innovationswln@gmail.com. We zullen hier altijd ons best voor doen om elke mail binnen de twee werkdagen te beantwoorden. Een maximumtijd waarin een probleem moet opgelost worden garanderen we niet aangezien er te veel factoren zijn die voor tijdsvertraging kunnen zorgen. Verder doet team WLN hun uiterste best om de klanten optimaal te ondersteunen waar nodig.

4.2.1.2 Wat is een noodgeval?

Problemen worden bij ons onderverdeeld in kleurcodes.

- Code rood: Alle websites zijn down en compleet onbereikbaar. Dit heeft een hoge impact op de gebruiker en zijn klanten. Directe actie is vereist zodat gebruikers en hun klanten terug aan de services kunnen.

- Code oranje: Meerdere websites liggen plat, hebben problemen met verbinding of werken heel traag. Dit heeft een redelijk hoge impact op de klant. Hier moet zo snel mogelijk een oplossing voor gevonden worden.

- Code geel: Een enkele of een paar websites hebben kleine problemen zoals kleine vertragingen of bepaalde pagina's van websites die niet laden. Dit heeft wel een impact op de gebruikers en zijn klanten, maar geen enorme zoals bovenstaande scenario's. Hier is tijd voor onderzoek zodat een oplossing kan gevonden worden en er een escalatie voorkomen wordt.

4.2.1.3 Hoe kunnen gebruikers ons bereiken?

We staan altijd klaar om onze gebruikers te helpen wanneer ze met een probleem zitten. We bieden verschillende manieren waarop gebruikers contact met ons kunnen opnemen, zodat we snel en efficiënt ondersteuning kunnen bieden.

Als gebruikers hulp nodig hebben, zijn er verschillende manieren om contact met ons op te nemen. We begrijpen dat het oplossen van problemen belangrijk is, daarom bieden we verschillende kanalen voor ondersteuning. Ons doel is om snel en efficiënt te reageren op gebruikersvragen en -problemen.

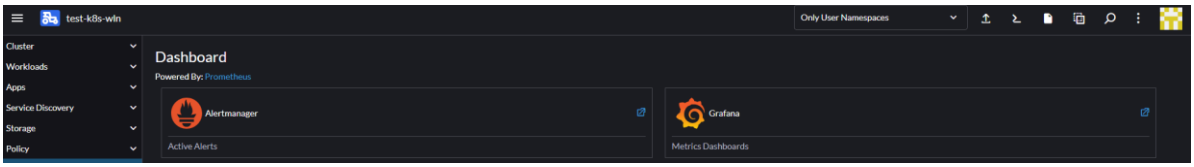
Gebruikers kunnen contact met ons opnemen via e-mail en videocall. Als er een probleem is waarvoor assistentie nodig is, kunnen gebruikers ons eenvoudig een e-mail sturen naar innovationswln@gmail.com. Ons hulpvaardige, toegewijde ondersteuningsteam zal de e-mail zo spoedig mogelijk behandelen en de nodige stappen verstrekken om het probleem op te lossen.

Voor een meer directe en persoonlijke benadering kunnen we ook videocalls regelen. Dit biedt de mogelijkheid om problemen in realtime te bespreken en directe verduidelijking te geven. Gebruikers kunnen aangeven welke videocall-service hun voorkeur heeft, zoals Zoom, Microsoft Teams of Google Meet, en we zullen een geschikte tijd en datum plannen.

We streven ernaar om problemen snel en efficiënt op te lossen, met de nadruk op een positieve gebruikerservaring. Ons ondersteuningsteam is er om gebruikers te helpen en ervoor te zorgen dat ze de beste ondersteuning krijgen. We staan klaar om vragen te beantwoorden, problemen op te lossen en begeleiding te bieden waar nodig.

4.2.2 3. Does the team record monthly metrics?

Wekelijks wordt er gekeken naar onze monitor tool Grafana en de Alertmanager. Hierbij kan er vroegtijdig actie ondernomen indien nodig. Door de monitoring kan er ook bekeken worden of alle virtuele machines nog wel in goede staat zijn, netwerkverkeer,… De alertmanager zal alerts geven en ook aanduiden waar deze alerts zich voordoen. Zodanig dat er eventuele problemen met grote gevolgen opgelost kunnen worden voor deze zich effectief voordoen.



Figuur 97: dashboard monitoring (WLN Innovations)

B. Modern Team Practices

4.2.3 5. Do you have a password safe?

We hebben toegang tot de Google Cloud console omgeving, met ons e-mailadres innovationswln@gmail.com. Hierbinnen bevindt zich een secret manager, welke wachtwoorden kan opslaan die door Google zelf geëncrypteerd worden, door een crc-32c checksum toe te passen. Hier kan ook ingesteld worden welke accounts allemaal toegang hebben tot deze secret manager en biedt logmogelijkheden om te kijken wie er acties onderneemt in deze manager.

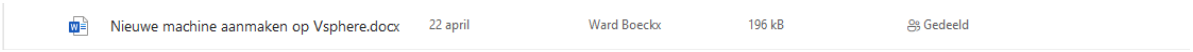
Name	Location	Encryption	Labels	Created	Expiration
cloudcasa	Automatically replicated	Google-managed	None	6/8/23, 8:51 PM	Never
emailwlni	Automatically replicated	Google-managed	None	6/8/23, 8:42 PM	Never
Rancher	Automatically replicated	Google-managed	None	6/8/23, 8:46 PM	Never
virtuelemachines	Automatically replicated	Google-managed	None	6/8/23, 8:52 PM	Never

Figuur 98: wachtwoordkluis Google Cloud (WLN Innovations)

4.2.4 9. Does your team write “design docs?”

Hierbij wordt er een gedeeld bestand bewaard, waar elke installatie van elk apart onderdeel in uitgelegd wordt.

Bijvoorbeeld:



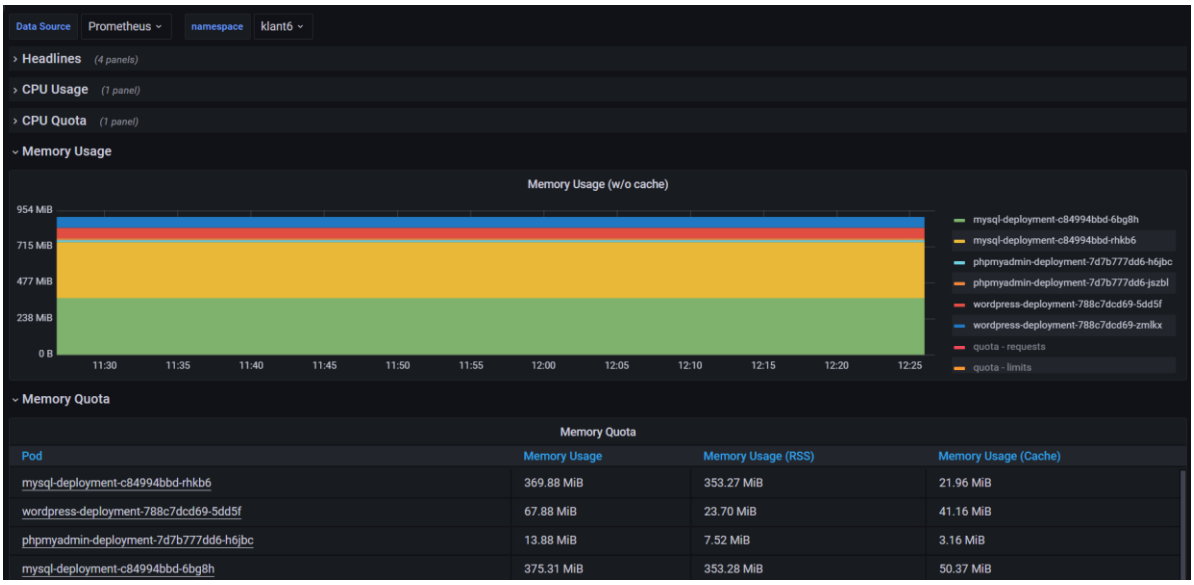
Figuur 99: documentatie virtuele machines (WLN Innovations)

Figuur 100: aanmaken virtuele machine (WLN Innovations)

C. Operational Practices

4.2.5 **12. Does each service have appropriate monitoring?**

Zoals eerder beschreven, gebruiken wij de tool Grafana om onze monitorring te doen.



Figuur 101: monitoren met Grafana (WLN Innovations)

We kunnen, indien we dit wensen, per namespace en service monitoring gaan doen. In bovenstaand voorbeeld hebben we de namespace met de naam klant6 genomen. Hier draait een Wordpress container samen met mySQL en phpmyadmin. Van elke service kan gekeken worden hoeveel CPU, geheugen, netwerk deze gebruiken.

Hierdoor kunnen we snel reageren op eventuele problemen en kunnen we de kwaliteit van onze services waarborgen.

4.2.6 **15. Do roll-outs to many machines have a "canary process"?**

Bij het uitvoeren van updates naar meerdere machines met behulp van Puppet, volgen we een gecontroleerd en geoptimaliseerd proces. We maken gebruik van een "canary-proces" waarbij één machine als "canary" fungeert en als eerste de updates ontvangt. Hierdoor kunnen we de impact en stabiliteit van de updates monitoren voordat we ze op een grotere schaal toepassen.

Tijdens de canary-fase worden de prestaties, functionaliteit en stabiliteit van de canary-machines nauwlettend in de gaten gehouden. We verzamelen gegevens en statistieken om ervoor te zorgen dat de updates succesvol zijn en dat er geen onverwachte problemen optreden. Eventuele problemen die worden geïdentificeerd, worden snel opgelost voordat de updates worden uitgerold naar alle machines.

Door het toepassen van het canary-proces kunnen we een betrouwbare en consistente infrastructuur handhaven. Het minimaliseert het risico van onvoorziene problemen en zorgt ervoor dat updates gecontroleerd en geoptimaliseerd worden uitgevoerd. Op deze manier kunnen we de stabiliteit en prestaties van ons systeem waarborgen terwijl we voortdurend verbeteringen en updates implementeren met behulp van Puppet.

D. Automation Practices

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

We gebruiken Puppet als ons configuratiemanagementtool. Met Puppet kunnen we de configuratie van onze infrastructuur automatiseren en standaardiseren. We definiëren manifests die de gewenste staat van onze systemen beschrijven en Puppet zorgt ervoor dat deze consistent wordt gehandhaafd. Het gebruik van Puppet verbetert efficiëntie, consistentie en beheerbaarheid, helpt bij naleving van beleidsregels en maakt schaalbaar beheer van onze infrastructuur mogelijk.

E. Fleet Management Processes

4.2.8 **20. Is OS installation automated?**

We maken gebruik van Puppet om het installatieproces van besturingssystemen op onze virtuele machines (VM's) te automatiseren. Met Puppet kunnen we de configuratie en instellingen van het gewenste besturingssysteem definiëren en deze automatisch toepassen tijdens het installatieproces. Dit zorgt voor een gestandaardiseerde en efficiënte aanpak, bespaart tijd en minimaliseert menselijke fouten bij het installeren van besturingssystemen op onze VM's.

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

We hebben de mogelijkheid om software automatisch te patchen. Met behulp van Puppet, ons configuratiemanagementtool, kunnen we patches en updates eenvoudig implementeren op al onze machines. We definiëren configuraties in Puppet, die

vervolgens automatisch worden toegepast op alle relevante machines. Bovendien volgen we het canary-proces, waarbij we updates eerst op een klein subset van servers uitrollen om de impact te evalueren voordat we ze breder implementeren. Dit stelt ons in staat om potentiële problemen te identificeren en op te lossen voordat de updates op alle servers worden toegepast, waardoor we een veilige en gecontroleerde patching van software over ons gehele serverpark kunnen garanderen.

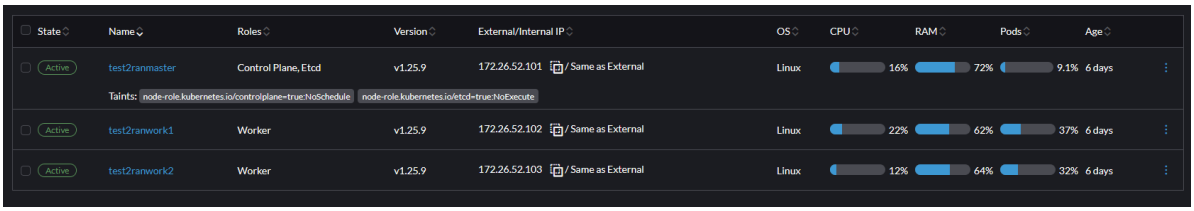
F. Disaster Preparation Practices

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

Met Kubernetes, een krachtig containerorkestratieplatform, kunnen we onze workloads en services verdelen over meerdere nodes. Dit betekent dat als er een diskstoring optreedt op een van de VM's, de workloads en services die op die specifieke node draaien automatisch worden verplaatst naar andere gezonde nodes binnen het cluster. Dit proces van automatische verplaatsing, ook wel bekend als pod rescheduling, zorgt ervoor dat de continuïteit van onze services wordt behouden, zonder merkbare onderbrekingen voor gebruikers.

Rancher, aan de andere kant, fungeert als een krachtig beheerplatform dat ons in staat stelt om het Kubernetes-cluster te orkestreren en te monitoren. Met Rancher kunnen we de gezondheid van de nodes bewaken en indien nodig actie ondernemen. In geval van een diskstoring detecteert Rancher automatisch de falende node en neemt het maatregelen om de workloads naar andere nodes te verplaatsen. Dit proces van automatische failover zorgt ervoor dat onze services blijven werken, zelfs als er een disk uitvalt.

Dankzij deze combinatie van Kubernetes en Rancher, samen met onze configuratie van drie VM's, kunnen we een robuuste en veerkrachtige serveromgeving bieden. Zelfs in het geval van een diskstoring, kunnen we de continuïteit van onze services handhaven door middel van automatische verplaatsing en failover. Hierdoor kunnen we onze gebruikers een betrouwbare en ononderbroken ervaring bieden.



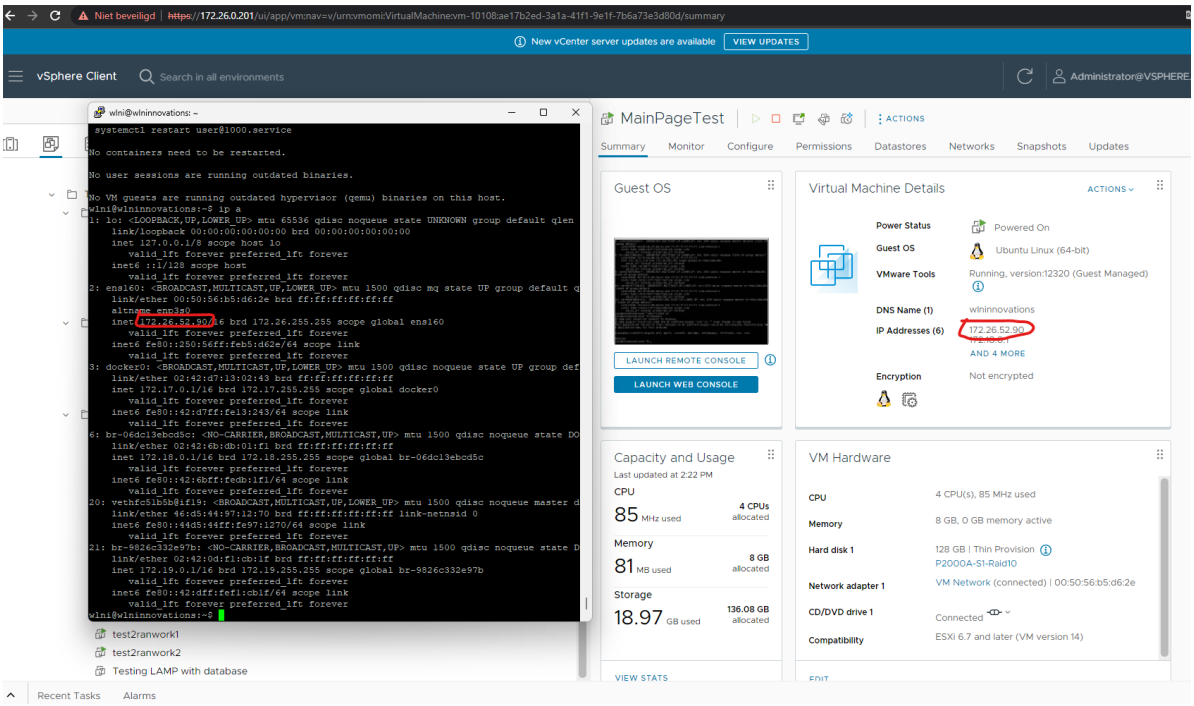
Figuur 102: automatische failover (WLN Innovations)

4.2.11 25. Are your backups automated?

We hebben geautomatiseerde backups ingesteld in onze infrastructuur met behulp van Google Cloud. Met de backupfunctionaliteit van Google Cloud kunnen we regelmatig en automatisch snapshots maken van onze gegevens en systemen. Deze backups worden opgeslagen in een veilige en betrouwbare omgeving, waardoor we in geval van gegevensverlies of systeemstoringen snel en gemakkelijk kunnen herstellen. Door gebruik te maken van de geautomatiseerde backupmogelijkheden van Google Cloud, kunnen we de integriteit en beschikbaarheid van onze gegevens en systemen waarborgen.

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

We hebben op afstand toegang tot de console van de machines in ons datacenter. Door middel van een VPN-verbinding kunnen we veilig verbinding maken met ons netwerk en vervolgens gebruiken we tools zoals PuTTY om via het vSphere-platform toegang te krijgen tot de virtuele machines (VM's). Deze configuratie stelt ons in staat om de console van de VM's te openen en directe controle en beheer over de systemen te hebben, zelfs op afstand. Dit stelt ons in staat om snel in te grijpen, configuraties te controleren en eventuele problemen op te lossen zonder fysiek aanwezig te hoeven zijn in het datacenter.



Figuur 103: remote access Vsphere (WLN Innovations)

G. Security Practices

4.2.13 30. Do you submit to periodic security audits?

We zouden ons regelmatig onderwerpen aan periodieke beveiligingsaudits om de veiligheid van ons systeem te waarborgen. Hierbij nemen we verschillende maatregelen om beveiliging te testen, waaronder penetratietesten, kwetsbaarheidsscans en code-audits. Hierdoor zouden we eventuele beveiligingslekken en zwakke punten in ons systeem identificeren en oplossen.

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

We hebben de mogelijkheid om gebruikersaccounts binnen 1 uur op alle systemen uit te schakelen. Met behulp van onze SQL-database, waarin alle gebruikersgegevens worden opgeslagen, kunnen we de status van een gebruikersaccount snel wijzigen en de toegang tot alle relevante systemen intrekken. Door het centraliseren van de gegevens in de database kunnen we efficiënt en consistent wijzigen doorvoeren op alle systemen, hierdoor kunnen we garanderen dat een snelle uitschakeling in geval van een noodgeval mogelijk is.

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

Ook hier hebben we de mogelijkheid om alle wachtwoorden binnen 1 uur te wijzigen met behulp van onze SQL-database. Door het opslaan van de wachtwoorden in de database kunnen we snel en efficiënt wijzigingen aanbrengen in de wachtwoorden van de accounts.

5 OPTIONEEL

In dit hoofdstuk zouden we graag enkele extra's willen bespreken die we aan ons project zouden toevoegen als we meer tijd hadden gehad. Hoewel we ons uiterste best hebben gedaan om een solide en functioneel project te ontwikkelen binnen de gestelde tijd, zijn er toch ideeën en functies overgebleven die we graag zouden willen verkennen en implementeren als we de gelegenheid hadden gehad. Dit zodat er toch een glimp opgevangen kan worden van de potentiële extra's die ons knap project zou kunnen bevatten.

We hadden het geweldig gevonden als we SSL-certificaten hadden kunnen implementeren in ons project. Het gebruik hiervan zou een extra laag van beveiliging hebben toegevoegd aan onze services en de vertrouwelijkheid van gegevens tussen onze gebruikers en het platform hebben verbeterd. Met SSL-certificaten zouden we een veilige versleutelde communicatie kunnen bieden, waardoor de privacy en integriteit van gebruikers worden gewaarborgd.

Automatisatie is ook een punt dat we betreuren dat het niet is klaar geraakt. Momenteel als de gebruikers een nieuw account maken, moeten we handmatig namespaces aanmaken in Rancher en de deployment van de YAML-files doen. Dit hadden we graag geautomatiseerd gehad. Dit zou niet alleen het comfort voor de gebruikers verhogen (deze moeten nu op ons wachten als ze zich aangemeld hebben), maar ook werk besparen qua configuratie van nieuwe namespaces.

Op Google Cloud hebben we momenteel ook een trial versie lopen van 90 dagen. Als we hiermee verder willen gaan, moeten we uiteraard een betalende versie aanschaffen om een back-up te blijven hebben van onze mappen en bestanden.

Als we terugblikken op onze OPS report cards, kunnen we wel zeggen dat we zeer tevreden zijn met ons eindresultaat. We hadden 16 cards geselecteerd en hebben er 15 kunnen waarmaken. Echter bleef er nog 1 card over wegens tijdgebrek: "Is there a database of all machines?". Dit hadden we nog graag als extra gehad omdat we het zelf een groot voordeel zouden vinden als er een centrale database werd gemaakt waar we echt zouden uitbreiden over elke machine. Denk maar aan het gebruik van CPU, RAM, harde schijf, geïnstalleerde programma's,… Hierdoor zouden we efficiënter kunnen werken en sneller toegang hebben tot de nodige informatie wanneer dat nodig is. Ook zou het ons nog helpen met het uitvoeren van analyses en het nemen van beslissingen met betrekking tot onze infrastructuur.

Naar beveiliging toe, hadden we oorspronkelijk zeker een firewall willen implementeren. Om dat op een correcte manier te doen, hadden we echter een overzichtelijke database nodig, zoals hierboven vermeld. Als we dan een mooi overzicht hadden van al de gebruikte poorten, konden we de firewall instellingen aanpassen op deze poorten en hadden we een degelijke bescherming van onze virtuele machines. We betreuren het dat dit niet gelukt is wegens tijdsgebrek, want we beseffen dat dit zeker een gewenste feature is, aangezien we het traject cloud & cybersecurity volgen.

Samengevat, we hebben een stevige basis gelegd, die zeker en vast voor uitbreiding en automatisatie vatbaar is.

6 PROCES

In het begin van ons project stonden wij voor een uitdagende opdracht als een team. We hebben een zeer gelijkwaardig team, in het opzicht van beperkte ervaring en kennis bij dit soort opdrachten. Wij moesten hosting voorzien en moesten op twee weken tijd een rapport samenstellen waarin wij omschreven welke programma's en onderdelen wij gingen gebruiken. Veel van deze programma's waren nog totaal onbekend voor ons. In het begin was dit voor ons dus een zeer complexe opdracht om op te starten. Er is veel hersenarbeid en overleg nodig geweest om voor onszelf toch een beetje duidelijkheid te scheppen, want wij keken echt wel tegen een hoge berg aan.

Eens er wat meer duidelijkheid was gecreëerd, konden wij echt beginnen aan onze opdracht. De taken waren verdeeld en iedereen begaf zich een beetje op zijn eigen terrein door kleine dingen op te starten en uit te testen. Die kleine dingen verliepen op zich wel redelijk goed, maar het echte werk begon pas toen we alles moesten samenvoegen.

Vele onderdelen, zoals het opzetten van de cluster, heeft veel voeten in de aarde gehad. Met het manueel opstellen van deze cluster, waren we al een volle week bezig. Toen dit eindelijk gelukt was, kwamen we toch tot de vaststelling dat het eenvoudiger is om dit met een programma zoals Rancher te doen, omdat dit een visuele interface had en veel meer gebruiksgemak bood. Nog twee weken later, hebben we zelf opnieuw een volledige cluster opgebouwd, omdat er in die vorige cluster toch nog kinderziektes zaten.

Dit project was een verhaal van 'trial & error', waarin we voornamelijk veel 'error' tegenkwamen. Het heeft ook de nodige frustratie en verloren tijd opgeleverd, omdat je zolang met iets bezig bent en het heeft precies geen zin wat je doet. Langs de andere kant, als je al zolang iets aan het proberen bent, word je ook getest in doorzettingsvermogen. Soms heb je er al zoveel tijd aan verloren, dat je zo koppig wordt om te blijven doorzetten tot het daadwerkelijk gelukt is. Als het dan effectief ook zo ver is dat het lukt, geeft dat een uitzonderlijk geluksgevoel.

Om het gewenste resultaat te verkrijgen, hebben wij verschillende soorten bronnen geraadpleegd. Uiteraard zijn websites zoals Youtube en Stackoverflow al grote bronnen van informatie geweest voor ons, maar websites die op een intelligente manier het web afschuimen, kunnen soms ook verhelderend werken. We hebben ons verdiept in handleidingen die we op de websites van de softwareleverancier vonden en zochten cursussen op waar alles stap voor stap werd uitgelegd. We hebben zelfs gemaild en gechat met de technische dienst om zo snel mogelijk geholpen te kunnen worden met correcte info.

We hebben met ons team telkens kleine stappen gezet. Onze bedoeling was ook om via het 'canary proces' te werken, kleine stappen zetten om uiteindelijk tot een groter geheel te komen. Hierdoor daagden we onszelf ook uit. Nadat er een onderdeel succesvol was afgerond, stelden we ons direct de vraag: "Wat kunnen we nu nog verder implementeren, om ons project zo ver mogelijk af te werken, zoals we het beoogd hadden?".

De samenwerking tussen de teamleden verliep eigenlijk vrij goed. Onze mindere ervaring en kennis, wat we oorspronkelijk als een zwakte zagen, werd uiteindelijk een sterkte. Waar iedereen hun eigen problemen/deelproject probeerde uit te werken, kregen zij daar een duidelijk beeld over de werking er van. Omdat we het belangrijk vonden dat ieder lid van ons project op de hoogte was van de werking, planden we regelmatig overlegmomenten in via teams om deze onderdelen dan uit te leggen aan elkaar. Zo wisten we allemaal waarmee we bezig waren en werd de kans op onvrijwillige misconfiguratie kleiner.

Het uiteindelijke project vonden wij in het begin dus echt een uitzonderlijk moeilijke opdracht. Als je kijkt naar het eindresultaat dat we nu hebben kunnen afleveren, zullen er nog wel verbeteringen mogelijk zijn, zeker naar volledige automatisatie toe. Het proces dat wij echter hebben afgelegd, vonden wij toch zeer positief en in stijgende lijn te verlopen. We hebben veel bijgeleerd door onze fouten en ook van elkaar. Dat was uiteindelijk ook de kracht van dit project, waarin we naar verloop van tijd meer en meer het nut begonnen van in te zien.

7 BESLUIT

Als we nu even stil staan bij het project en even omkijken naar wat we hebben gebouwd, kunnen we wel zeggen van onszelf dat het een mooi exemplaar is geworden.

Een hosting platform waar klanten zich kunnen registreren en vervolgens een mooie, redundante omgeving krijgen waar ze een webshop kunnen creëren via Wordpress. Dit allemaal voorzien van loadbalancing, tot back-ups, security, monitoringtools, SFTP-toegang en schaalbaarheid. Kortweg, een hosting platform met een sterke fundering.

Uiteraard is het zeer spijtig dat het gebrek aan tijd zo groot was dat we niet al de technologieën hebben kunnen toepassen. We begrijpen het belang en de voordelen van technologieën zoals SSL, firewall en ook automatisaties en hadden ze graag geïmplementeerd om ons project naar een hoger niveau te tillen. Natuurlijk leggen we de lat hoog voor onszelf en zijn we hier ook streng op, toch mogen we zeker niet vergeten te kijken naar wat wel is gelukt.

Project Hosting heeft ons vooral veel bijgeleerd, geleerd om te werken in teamverband, onze kennis enorm bijgeschaafd door te experimenteren met al de technologieën, maar bovenal ons uitgedaagd om door te gaan en vooral te geloven in wat we kunnen, collectief en individueel.

Het succes is het resultaat van toewijding, doorzettingsvermogen en een onverwoestbaar geloof in jezelf. Vandaag sluiten we dit zware project af, wetende dat we alles hebben gegeven en trots kunnen zijn op wat we hebben bereikt.

8 FIGUURLIJST

9 BRONNENLIJST

Accesspay. (2022). *SFTP eenvoudig gemaakt: Wat is het? Hoe werkt het?* Opgeroepen op juni 6, 2023, van Accesspay.com: https://accesspay.com/nl/knowledge-hub/veiligheid/wat-is-sftp-en-hoe-werkt-het/

Andrews, G. (2020, August 5). *Get Started with Puppet: A Tutorial Guide for First-Timers*. Opgeroepen op Junee 2023, van https://www.puppet.com/: https://www.puppet.com/blog/get-started-puppet-tutorial#get-started

Authors, T. K. (2023). *Kubernetes Documentation*. Opgeroepen op juni 9, 2023, van Kubernetes: https://kubernetes.io/docs/home/

Catalogic, S. (2023). *CloudCasa for Velero*. Opgeroepen op mei 28, 2023, van Cloudcasa: https://cloudcasa.io/

Cloud, G. (2023). *CRC32C and Installing crcmod*. Opgeroepen op juni 8, 2023, van cloud.google.com: https://cloud.google.com/storage/docs/gsutil/addlhelp/CRC32CandInstallingcrcmod

Cloud, G. C. (2023). *Get startted with Google Platform*. Opgeroepen op mei 27, 2023, van Google Cloud console: https://console.cloud.google.com/getting-started

DigitalOcean, L. (2022). *How To Protect SSH with Fail2Ban on Ubuntu 20.04*. Opgeroepen op juni 4, 2023, van digitalocean.com: https://www.digitalocean.com/community/tutorials/how-to-protect-ssh-with-fail2ban-on-ubuntu-20-04

Dillon, L. (2023). *What is Rancher?* Opgeroepen op juni 9, 2023, van openlogic: https://www.openlogic.com/blog/what-is-rancher

Google. (2023). *Install the gcloud CLI*. Opgeroepen op mei 27, 2023, van Google Cloud: https://cloud.google.com/sdk/docs/install

Kiarie, J. (2021). *How to Set a Custom SSH Warning Banner and MOTD in Linux*. Opgeroepen op juni 7, 2023, van tecmint.com: https://www.tecmint.com/ssh-warning-banner-linux/

LinuxTechi. (2023). *How to Install Rancher on Ubuntu 22.04 (Step by Step)*. Opgeroepen op mei 8, 2023, van Linuxtechi.com: https://www.linuxtechi.com/how-to-install-rancher-on-ubuntu/

Longhorn authors, &. L. (2023). *Cloud native distributed block storage for Kubernetes*. Opgeroepen op mei 21, 2023, van Longhorn: https://longhorn.io/

Mulan, G. (2023). *Web Hosting Security Best Practices and What to Look for in Secure Website Hosting*. Opgeroepen op mei 26, 2023, van www.hostinger.com: https://www.hostinger.com/tutorials/web-hosting-security#3_Use_SFTP_Instead_of_FTP

Natarajan, R. (2012). *How to Setup Chroot SFTP in Linux (Allow Only SFTP, not SSH)*. Opgeroepen op juni 7, 2023, van thegeekstuff.com: https://www.thegeekstuff.com/2012/03/chroot-sftp-setup/

Quinten D, C. V. (2023). *Puppet: A deep dive into puppet, config management with Puppet, Basic Concepts.* Geel: Thomas More.

Shivang. ((S.a.)). *What is Grafana? Why Use It? Everything You Should Know About It*. Opgeroepen op juni 9, 2023, van scaleyourapp.com: https://scaleyourapp.com/what-is-grafana-why-use-it-everything-you-should-know-about-it/

Suse, R. (2023). *CIS scans*. Opgeroepen op mei 18, 2023, van ranchermanager.docs.rancher.com: https://ranchermanager.docs.rancher.com/pages-for-subheaders/cis-scans

Suse, R. (2023). *Enable Monitoring*. Opgeroepen op mei 28, 2023, van ranchermanager.docs.rancher.com: https://ranchermanager.docs.rancher.com/v2.6/how-to-guides/advanced-user-guides/monitoring-alerting-guides/enable-monitoring

VMware, I. (2023). *Overview of virtual machine snapshots in vSphere (1015180)*. Opgeroepen op mei 15, 2023, van kb.vmware.com: https://kb.vmware.com/s/article/1015180

WordPress. (2023). *Welkom bij de meest populaire sitebouwer ter wereld.* Opgeroepen op juni 9, 2023, van WordPress: https://wordpress.com/nl/