



Projectverslag Project Hosting

Groep 1

Brent Beliën, Jesse Van Doninck, Manuel Hayen, Pei Yang, Philippe Boets, Robin Bruynseels en Wouter Bosmans

Klasgroep CCS A&B

Project Hosting 2020 (academiejaar 2019-2020)

Inhoudsopgave

Inhoud

Inhoudsopgave	3
Inleiding	4
1 Onderzoek	5
1.1 Software	5
1.2 Verantwoording keuzes	6
2 Doelstellingen.....	12
Bronvermelding	21
Bijlage/tutorials.....	22
1. Netwerk	22
1.1. Debian 10 buster netwerkconfiguratie	22
1.2. Linux mint 19.3 GUI netwerkconfiguratie	24
1.3. DNS.....	27
2. Puppet.....	30
2.2 puppet client	33
2.3 configuratie puppet	34
2.4 Troubleshooting	38
2. Backup.....	39
3. ClamAV en chkrootkit	40
4. Nfs installatie	42
5. Docker (Swarm & Compose) – Portainer & Swarmprom	44
5.1 Docker (Swarm)	44
5.2 Portainer	44
5.3 Docker Compose.....	49
5.4 Swarmprom	51

Inleiding

Project Hosting is de grootste uitdaging voor de 2^{de} jaarsstudenten. In dit project hebben we verschillende mogelijkheden overwogen om een hosting omgeving op te zetten voor de studenten APP/BI.

Deze studenten moeten momenteel al hun projecten lokaal of op zelfstandige basis hosten. Dit was geen makkelijke opdracht. In het deel "onderzoek" overlopen we welke softwareproducten wij onderzocht hebben en uiteindelijk gaan aanbieden aan onze toekomstige "klanten". Natuurlijk hebben we vele keuzes moeten maken doorheen dit project en ons steeds gebaseerd aan de noden van de klant.

Alle grote opdrachten die we hebben uitgewerkt doorheen dit project hebben we verdeeld. Dit zijn vooral de belangrijke aspecten van ons project, natuurlijk zijn dit niet de enige zaken en bevat ons project meer, al dan niet, kleinere taken/opdrachten. De volgorde die u hier vindt is gebaseerd op basis van prioriteit.

In het deel "onderzoek" vindt u een overzicht van hoe wij onze opstelling gerealiseerd hebben. Hier kan u zien waar welke machine draait.

Onze ops report cards zijn zeker niet te vergeten. Ook hier zal u een deel vinden waar we onze gekozen ops report cards gaan bespreken en gedetailleerder uitleggen. Wij hebben zo goed mogelijk onze report cards uitgewerkt.

Ten slotte vindt u een evaluatie van ons project. In deze evaluatie kijken wij terug naar ons project van start tot einde. We kijken terug op de belangrijkste keuzes en besluiten of we nu eenmaal blij zijn met het resultaat of niet.

1 Onderzoek

De onderzoekfase heeft ons veel tijd gekost. In deze fase gaan we bij onze keuzes zoveel mogelijk informatie zoeken naar de betreffende producten/software. Verder kan u al onze keuzes terugvinden met vergaarde informatie waarom we deze keuze gekozen hebben.

1.1 Software

Onze eerste zorg was de name resolution. Doorheen het tweede jaar van onze richting hebben we een groot stuk van DNS geleerd. Dit leek ons de perfecte kans om dit zo goed mogelijk te integreren. We hebben dus gebruik gemaakt van BIND9. De DNS-server werd snel en correct opgezet en al snel konden we onze applicaties contacteren zonder steeds het IP-adress te moeten ingeven.

De volgende stap was het opzoek gaan naar oplossingen voor onze ops report cards. Om alle verschillende/interessante ops cards te bestuderen hebben we de eerste les al deze zaken samen overlopen. Daarna hebben we de cards op Monday.com gezet om zo individueel keuzes te maken wie welke interessant vind. Hierna hebben we een overzicht wie aan welke zaken gaat werken en kunnen we aan de slag.

Voor de back up waren we eerst van plan om bacula te gebruiken. Dit leek ons een goed idee omdat we deze konden gebruiken als GUI. Echter liepen we later op het einde van het project toch op problemen. Dit gaat u later verder kunnen lezen. Zo hebben we snel een alternatief moeten zoeken. Zo zijn we op 'rsync' gekomen. Een linux based applicatie die makkelijk te installeren en beheersen valt. Verder zult u de implementatie en documentatie van rsync vinden.

Het automatische gedeelte hadden we eerst een beetje uit het oog verloren. Daarna zijn we met puppet aan de slag gegaan. Puppet was ons bekend door het aanleren in de OPO Linux.

Voor het ticketsysteem hebben wij gekozen voor OSTicket. Het was een ticketsysteem dat niet super simple te integreren viel maar had wel al de nodige functies die we goed konden gebruiken voor verschillende ops report cards. We hebben ook verschillende andere overwogen maar in onze ogen was OSTicket system compatibeler dan de andere.

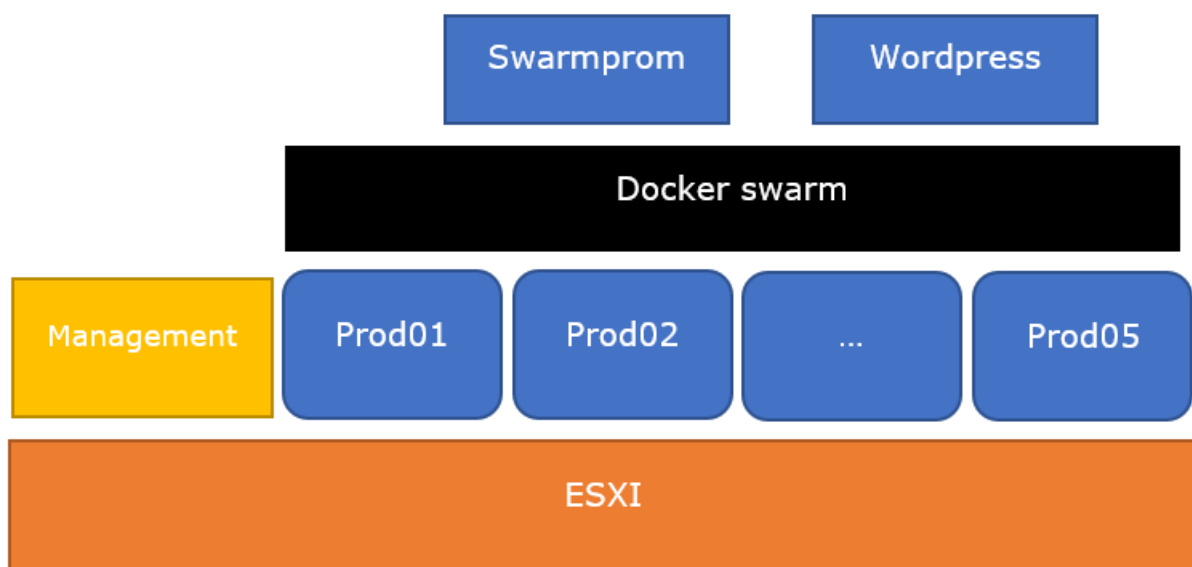
Als central beheer van codes en andere zaken gebruiken wij onze meest vertrouwde GIT. Vanaf het eerste jaar hebben we met GitHub leren werken en daarom hebben we dus gebruik gemaakt om code zo bij te houden. We hebben kort naar een eventueel alternatief gezocht. Maar de kennis die we al hadden heeft de keuze snel gemaakt.

Tenslotte bespreken we nog kort het centraliseren van de wachtwoorden. Hier zijn ook vele mogelijkheden voor. Wij zijn uitgekomen bij LastPass. Dit was ook kosteloos. Echter had er hier ook een andere password safe mogelijk geweest.

1.2 Verantwoording keuzes

1.2.1 Besturingssysteem

Hier ziet u een schema dat onze opstelling voorstelt:



We kregen een ESXI omgeving ter beschikking om onze applicatie op te realiseren. Zoals in het schema te zien is hebben we 6 virtuele machines op gezet. Deze stellen de fysieke machines voor die we in een echte omgeving ter beschikking zouden hebben. We hebben er voor gekozen om een combinatie van Mint en Debian systemen te implementeren. Op de Management vm draait Linux mint 19.3 en op de productie vm's draait Debian 10. Eerst was het plan om elke klant een eigen omgeving te geven via docker swarm. Productie 1 tot en met 5 zen telkens een swamr node, met 5 als swarm master. Dit zou er voor zorgen dat de workload automatisch over de 5 productie vm's gebalanceerd wordt. Na enig onderzoek bleek het niet zo evident te zijn om voor elke klant een eigen LAMP stack op te zetten in een Docker swarm.

Uiteindelijk hebben we gekozen om de websites die klanten willen hosten op Prod02 te zetten, met maar 1 LAMP stack waar verschillende partities per klant op staan. We hebben er voor gekozen om Docker swarm niet te verwijderen, maar we hebben er Wordpress en Swarmprom op gezet. Swarmprom dient als monitoring voor ons systeem:



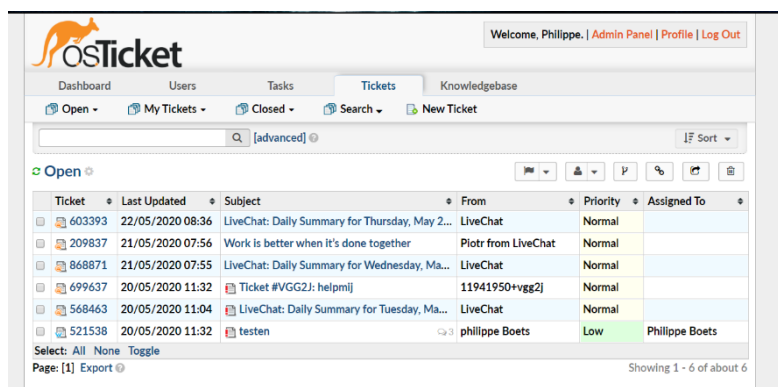
1.2.2 Overzicht software per machine

Hier vindt u een overzicht van alle geïnstalleerde software per vm, verder in het document wordt alle software nog besproken. De services die op de Docker Swarm draaien worden automatisch verdeeld over de verschillende nodes.

1. Management:
 - a. LAMP stack rechtstreeks op OS waar osTicket op draait.
2. Prod01:
 - a. LAMP stack rechtstreeks op OS waar Zabbix op draait
 - b. Deze vm is een Swarm Node voor de Docker Swarm
3. Prod02:
 - a. LAMP stack rechtstreeks op OS met verschillende partities per klant, nu draaien er 3 websites op namelijk:
 - i. www.easy-hosting.team01.local (onze eigen website)
 - ii. www.naaimachinesbriersjos.team01.local (klant met statische Apache)
 - iii. www.phpppuzzle.team01.local (klant met PHP + apache)
 - b. Deze vm is een Swarm Node voor de Docker Swarm
4. Prod03 en Prod 04 zij beide Swarm Nodes voor de Docker Swarm
5. Prod05:
 - a. Dit is onze backup server waar bacula en rsync op staan.
 - b. BIND9 DNS server
 - c. Storage server werkend met NFS4 voor de content van alle websites van de klanten.
 - d. Deze vm is de Swarm Master maar is zelf ook een Swarm Node
 - e. Portainer
6. Docker Swarm:
 - a. Swarmprom:
 - i. Hier draait ons monitoring systeem op werkend met grafana
 - b. Wordpress:
 - i. Hiermee tonen we aan dat we Wordpress Websites kunnen hosten.

1.2.3 Ticketing

Tijdens ons onderzoek naar de geschikte ticketing software zijn we tot de conclusie gekomen dat er veel verschillende alternatieven zijn voor dit aspect van ons project. Software zoals freshdesk, osticket, zendesk en hubspot zijn allemaal aan bod gekomen. Uiteindelijk hebben we gekozen voor osticket omdat dit volgens ons de beste features had en het eenvoudigste te implementeren was.



Voor onze applicatie is de gratis versie van oSTicket voldoende om onze noden in te vullen. Een groot voordeel van oSTicket is dat het werkt via een gui, dit vergemakkelijkt het gebruik niet alleen voor de gebruiker maar ook voor ons als developer. Via deze gui ,in de vorm van een

webapplicatie gehost op ons platform, kan een admin alle instellingen aanpassen zoals het automatisch toekennen van prioriteiten aan tickets, automatische respons, Een agent, de persoon die de tickets afhandelt, krijgt hier ook een overzicht van alle tickets die binnengekomen zijn. Dit overzicht kan gerangschikt worden op prioriteit, zo hebben bijvoorbeeld tickets over bugs of connectieproblemen een hogere prioriteit dan tickets met algemene problemen of vragen.

oSTicket biedt een online platform aan waar klanten hun tickets (problemen met connecties, vragen, bugs, ...) kunnen indienen. Dit platform is gemakkelijk te bereiken via onze website. Hier kan de klant een nieuw ticket openen of de status van eerder ingediende tickets raadplegen. Elke klant krijgt ook een automatisch een mail aan ter bevestiging dat hun ticket in behandeling is.

SUPPORT CENTER

Support Ticket System

Guest User | [Sign In](#)

[Support Center Home](#) [Open a New Ticket](#) [Check Ticket Status](#)

Welcome to the Support Center

In order to streamline support requests and better serve you, we utilize a support ticket system. Every support request is assigned a unique ticket number which you can use to track the progress and responses online. For your reference we provide complete archives and history of all your support requests. A valid email address is required to submit a ticket.

[Open a New Ticket](#)

[Check Ticket Status](#)

1.2.4 Backup

Het back up aspect van ons project was zeker een moeilijk punt. Het leek ons in het begin een 'clear case' dat we voor Bacula zouden kiezen. Zoals eerder vermeld heeft Bacula een GUI instantie 'Baculum' die volgens verschillende artikels makkelijk te gebruiken valt. De installatie was vroegtijdig gedaan. Wij zelf dachten dat de installatie goed gelukt was, echter kwamen we op het einde van ons project tot de conclusie dat we deze niet goed hadden geïmplementeerd in ons system. Dit betekent niet dat we Bacula geen goed system vinden, we zijn er van overtuigd dat als je de configuratie tot een goed einde brengt dat dit een zéér goed systeem kan zijn. Zoals u hieronder kan zien hebben we de GUI en bacula kunnen installeren. Echter was de configuratie van files een probleem en draaide het op het einde niet succesvol uit.



```
admin05@T01PROD05:~$ bconsole
Connecting to Director T01PROD05:9101
1000 OK: 103 T01PROD05-dir Version: 9.6.3 (09 March 2020)
Enter a period to cancel a command.
*status dir
T01PROD05-dir Version: 9.6.3 (09 March 2020) x86_64-pc-linux-gnu debian 10.3
Daemon started 20-May-20 03:17, conf reloaded 20-May-20 03:17:47
Jobs: run=0, running=0 mode=0,0
Heap: heap=131,072 smbytes=80,309 max bytes=85,298 bufs=353 max bufs=363
Res: njobs=3 nclients=5 nstores=2 npools=3 ncats=1 nfsets=2 nscheds=2

Scheduled Jobs:
=====
Level      Type      Pri  Scheduled      Job Name      Volume
=====
Incremental Backup    10  20-May-20 23:05 BackupClient1 Vol-0001
Full       Backup    11  20-May-20 23:10 BackupCatalog  Vol-0001
=====

Running Jobs:
Console connected at 20-May-20 15:45
No Jobs running.
=====

Terminated Jobs:
=====
JobId Level  Files  Bytes  Status  Finished      Name
=====
1 Full      21    9.700 M OK      18-May-20 23:05 BackupClient1
2 Full      1    53.46 K OK      18-May-20 23:10 BackupCatalog
3 Incr       0      0 OK      19-May-20 23:05 BackupClient1
4 Full      1    59.00 K OK      19-May-20 23:10 BackupCatalog
=====
```

ClientId	Name	Uname	AutoPrune	Job retention	File retention	Actions
2	T01PROD01-fd		Yes	180 days	60 days	Details
3	T01PROD02-fd		Yes	180 days	60 days	Details
4	T01PROD03-fd		Yes	180 days	60 days	Details
5	T01PROD04-fd		Yes	180 days	60 days	Details
1	T01PROD05-fd	9.6.3 (09Mar20) x86_64-pc-linux-gnu,debian,10.3	Yes	180 days	60 days	Details

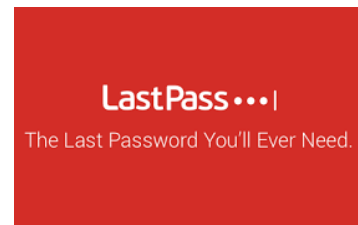
Zo zijn wij op het einde moeten opzoek gaan naar een alternatief. We zijn zo terecht gekomen bij rsync. Zoals reeds vermeld is dit een linux based applicatie die snel en handig te installeren viel. We zijn er meteen op gesprongen. En opzoek gegaan naar gepaste tutorials voor ons nieuw back up systeem. Deze vonden we al snel en kwamen zo tot een succesvolle installatie. Hiervoor vonden we vele tutorials, van installatie, tot verschillende manieren om back ups te runnen.



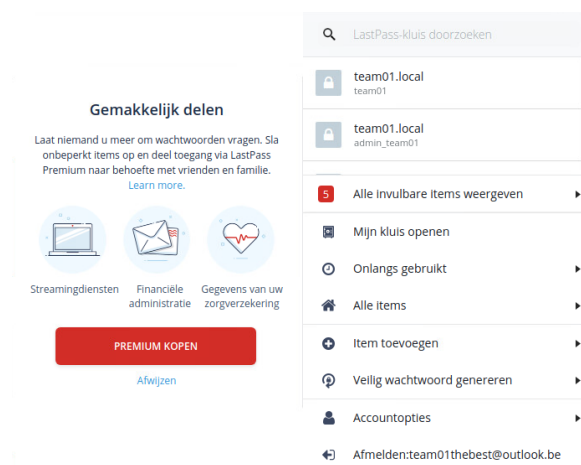
Rsync is een applicatie die een full copie (back up) van het systeem kan en zal nemen. Zoals u in het deel van nfs kunt zien maakten we verschillende mountpoints. We hebben besloten om een mountpoint te maken voor de backups, zodat we alle backups kunnen opslagen op één plaats. Bij ons is dit het /backupdisk/prod01-5 path op de prod05 VM. Op de andere machines konden we deze files raadplegen via het /mnt/team01_backups/prod01-5 path. We hebben ook de back up kunnen automatiseren via de crontab functie. De correcte installatie en commando's die wij hebben uitgevoerd kan u vinden in de tutorials (bijlage) en de bronnen.

1.2.5 Wachtwoorden beheren

Omdat we zo veel mogelijk met GUI's wouden werken hebben we gekozen voor LastPass. We werkten ook met een dockerswarm en portainer. Hier hadden we dus ook GUI waar we met verschillende wachtwoorden gingen werken. Dit is maar één van de vele voorbeelden die we kunnen geven waar we gebruik maakten van verschillende logins en passwords. Omdat verschillende personen van ons team al werkten met LastPass was voor ons de keuze snel gemaakt om voor LastPass te kiezen.



Het is een zéér gemakkelijke installatie. We volgden een korte tutorial die u ook in de bronvermelding kan vinden.



1.2.6 Automatische configuraties

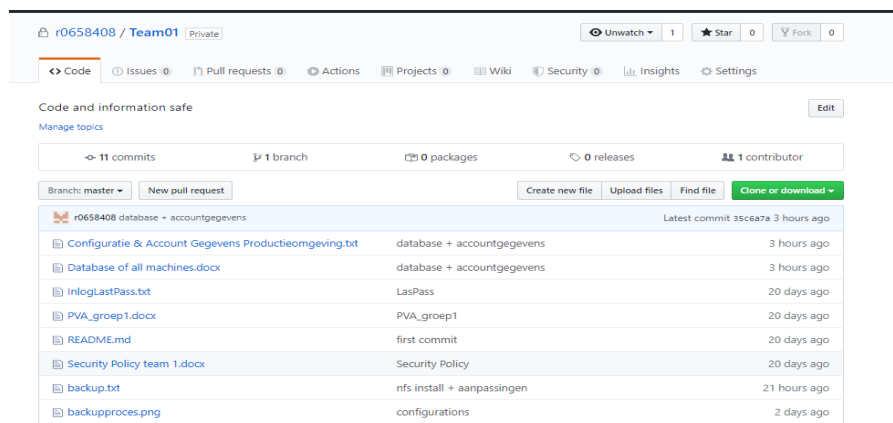
Docker Swarm (& Portainer) heeft het zoals in het projectplan gezegd, gehaald in onze opstelling. Misschien niet helemaal op de manier dat we verwachtten, maar nog steeds speelt het zijn rol naast nog een andere keuze die we gemaakt hebben. Naast docker swarm maken we gebruik van Puppet voor automatische updates voor onze productie VMs en het opkuisen van Docker Swarm containers

die niet langer gebruikt worden. Details hierover kunnen teruggevonden worden in de handleiding.

1.2.7 Monitoring

1.2.8 Versiecontrole

Om alle belangrijke zaken bij te houden maken wij gebruik van Git. Git hebben we al snel leren kennen in het eerste jaar in de IT Factory. Het is een zeer handige tool om zaken bij te houden. Als team heeft het ook enorm veel voordelen. Een logische takstructuur, flexibel,... Het is makkelijk om teamgenoten uit te nodigen zodat zij ook kunnen bijdragen aan de team01 repository.



1.2.9 Antivirus & malware

We gingen op zoek naar compatibele antivirussen voor onze Debian VM's. Al snel kwamen we uit op ClamAV. ClamAc is een opensource standard voor gateway scanning. Het detecteert trojans, virussen,... dus dit is perfect voor toekomstige klanten die met mail,... gaan werken. De installatie van ClamAV was vanzelfsprekend. We volgen de commandline install van ClamAV. Natuurlijk is het belangrijk dat de scans automatisch gebeuren. Nog een pluspunt van ClamAV is dat dit default ingesteld staat om het uur te scannen. Wat wij perfect vonden voor onze opdracht. Ook is er de mogelijkheid om dit aan te passen naar mindere keren per dag. Dit is dus instelbaar naar wens van de klant (wij zelf). De installatie en tutorials die wij gevolgd hebben kan u vinden in de bronvermelding.





Naast ClamAV waren we nog opzoek naar een scanner die ook de VM root scant op eventuele bedreigingen. Zo werd er in ons team vermeld dat er een linux based applicatie bestaat die chkrootkit heet. Deze scant de hele root directory van een VM. Op deze manier kunnen we voorkomen dat onze VM

vanbinnenuit kan bedreigt worden. De installatie hiervan was puur commandline based. Maar chkrootkit had ook zijn minpunten. De default instellingen van chkrootkit staan zo dat het enkel handmatig kan gebeuren. Hierdoor moesten we in de config files van chkrootkit `"/etc/chkrootkit"`. Hier kunnen we de daily run time veranderen zodat deze ook automatisch de directory scant.

Zo komen we tot de conclusie dat chkrootkit en ClamAV elkaar zeer goed aanvullen. Het ene doet wat het andere mist. Het grootste pluspunt is dat deze allebei automatisch kunnen lopen. Voor specifieke installatie stappen kan u deze vinden in de bronvermelding en tutorials (bijlage).

2 Doelstellingen

2.1 Ops Report Cards

Naast de verschillende servers en opstellingen die we nodig hebben om PHP-projecten te hosten hebben we ook andere zaken gerealiseerd. Deze zaken moesten zeker aanwezig zijn om een goed project op te leveren.

Verder kan u een opsomming vinden van de verschillende ops report cars die we hebben gerealiseerd. We hadden verschillende cards gekozen om af te werken, we hebben niet al onze keuzes kunnen afwerken maar hebben er zeker voldoende afgewerkt..

2.2 Uitgewerkte cards

Hier vindt u de lijst met al de ops report cards die wij hebben uitgewerkt in ons team. Ook de verantwoording en hoe we deze hebben uitgewerkt kan u hieronder terugvinden. Diepere uitleg vindt u op het onderdeel van de betreffende card zelf.

- Are user requests tracked via a ticket system?
- Are "the 3 empowering policies" defined and published?
- Does the team record monthly metrics?
- Do you have a "policy and procedure" wiki?
- Do you have a password safe?
- Is your team's code kept in a source code control system?
- In your bugs/tickets, does stability have a higher priority than new features?
- Does your team write "design docs"?
- Do you have a "post-mortem" process?
- Does each service have appropriate monitoring?
- Do you have a pager rotation schedule?
- Do you use configuration management tools like cfengine/puppet/chef?
- Is there a database of all machines?
- Can you automatically patch software across your entire fleet?
- Do you have a PC refresh policy?
- Can your servers keep operating even if 1 disk dies?
- Are your backups automated?
- Do machines in your data center have remote power / console access?
- Do desktops/laptops/servers run self-updating, silent, anti-malware software?
- Do you have a written security policy?
- Can a user's account be disabled on all systems in 1 hour?
- Can you changed all privileged (root) passwords in 1 hour?

Ticket system

Het gebruik van een ticketsysteem in je project biedt enorm veel voordelen en kan zéér polyvalent gebruikt worden. Het zorgt ervoor dat verschillende teamgenoten niet aan hetzelfde probleem werken. Maar het kan ook gebruikt worden om klanten aan te werven. We kunnen ook taken verdelen over de teamgenoten.

Tijdsbeheer is ook van belang bij een ticketsysteem. Een ticketsysteem voorkomt ookonderbrekingen van gebruikers die aanvragen willen doen of de status van aanvragen willen opvragen.

Voor managers is een ticketsysteem ook zeer interessant. Hij of zij kan zien wie welke tickets afhandelt en waar sommige collega's vastlopen. Later kunnen we ook trends evalueren en zoeken waar de meeste fouten verschijnen. Systeemproblemen kunnen veel beter geïdentificeerd worden.

Voor ons project maken we gebruik van OSTicket system. Onder dit deel kan je meer uitleg vinden of hoe het werkt en wat het doet.

Is there a database of all machines?

Elke site welke machines daar draaien. Het is zéér belangrijk om te weten welke systemen er draaien om zo toekomstige problemen effectief te kunnen oplossen.

Ook voor installaties is het zéér belangrijk. Zo moeten we weten welke commando's uit te voeren op welke systemen. Ook al lijken vele systemen principieel hetzelfde. Toch zijn er kleine verschillen om commando's succesvol uit te voeren.

Met zo'n database kan je beslissingen gaan nemen en helpt het dus problemen voorkomen.

Ons team heeft een document opgesteld om het op papier vast te leggen. Maar dit deel is ook mogelijk gemaakt door Zabbix. In het puntje van Zabbix kan u meer informatie vinden over hoe wij hiermee te werk gaan.

Machine	Specs	Installed Features
EXSI (Shared datacenter)	CPU: 52Ghz RAM: 216GB HDD: 8TB	Note: Shared with all teams
T01MGMT01	OS: Linux mint 19.3 IP: 172.27.66.110 MAC: 00:50:56:9f:92:e9 CPU: 6 CPU's RAM: 10GB HDD: 256GB	VM-ware tools SSH manager for other machines Portainer client (GUI)
T01PROD01	OS: Debian 10 IP: 172.27.66.111 MAC: 00:50:56:9f:39:fc CPU: 4 CPU's RAM: 8GB HDD: 64GB	VM-ware tools Docker Swarm worker Zabbix OS-Ticket Puppet agent
T01PROD02	OS: Debian 10 IP: 172.27.66.112 MAC: 00:50:56:9f:56:8f CPU: 4 CPU's RAM: 8GB HDD: 64GB	VM-ware tools Docker Swarm worker Puppet agent
T01PROD03	OS: Debian 10 IP: 172.27.66.113 MAC: 00:50:56:9f:8d:66 CPU: 4 CPU's RAM: 8GB HDD: 64GB	VM-ware tools Docker Swarm worker Puppet agent
T01PROD04	OS: Debian 10	VM-ware tools

	IP: 172.27.66.114 MAC: 00:50:56:9f:3e:36 CPU: 4 CPU's RAM: 8GB HDD: 64GB	Docker Swarm worker Puppet agent
T01PROD05	OS: Debian 10 IP: 172.27.66.115 MAC: 00:50:56:9f:b3:64 CPU: 4 CPU's RAM: 8GB HDD: 64GB	VM-ware tools Docker Swarm manager MySQL Bacula Backup manager Portainer Puppet manager

Metric system

Een metric system is heel belangrijk als de data moet bijgehouden worden voor beslissingen of hoger management stuurt.

Het is zéér belangrijk om deze metrics te bepalen op korte maar krachtige vragen. Metrics worden vaak bijgehouden in grote hoeveelheden die dan zelf handmatig moeten worden gesorteerd om zo specifieke data weer te geven. Ook voor dit deel gerbuiken we Zabbix. Zabbix geeft de mogelijkheid om specifiek data of metrics weer te geven. U vindt meer info over hoe we dit gebruiken in het deeltje over Zabbix.

Policy and procedure

In een policy and procedure documenteert u al uw beleid en procedures. Dit kan in een team zeer nuttig zijn.

Alle procedures moeten in dit document staan voor het toevoegen/wijzigen/verwijderen die iedereen in het team zou moeten kunnen doen.

Enkel voorbeelden zijn:

- Wanneer een nieuwe medewerker begint.
- Wanneer een medewerker het bedrijf verlaat.
- Nieuwe machine geïnstalleerd word.
- ...

Eenmaal het gedocumenteerd is zou iedereen in het team deze stappen moeten kunnen doen. Heel occasioneel kan het ook gebruikt worden voor nieuwe medewerkers.

Onze policy and procedure kan u terugvinden op onze slimwiki pagina onder de rubriek policy and procedure.

Password safe

Door het gebruik van een password safe laat u zien dat u op een volwassen manier wachtwoorden kan beheren.

De password safe die wij gebruiken is LastPass. Al onze verantwoording kan u vinden onder het deel van LastPass.

Ook hebben wij een beschreven password policy die je kan raadplegen in onze slimwiki pagina onder de rubriek security policy.

Source code control system

Er bestaan maar weinige programmeurs of IT concullegas die geen source code control gebruiken. Source code control kan op verschillende manieren gebruikt worden. Van scripts tot configuratiebestanden.

Zoals eerder vermeld maken wij gebruik van Git. Doorheen onze eerste 2 jaar in de IT Factory hebben we al snel kennis gemaakt met Git. Zo waren al onze teamleden hier al mee gekent. Verdere informatie vindt u onder het deel van Git.

In your bugs/tickets, does stability have a higher priority than new features.

Het ticketsysteem dat wij gebruiken is OSTicketsystem. In de GUI van OSTicketsystem zijn er mogelijkheid om prioriteiten in te stellen naar onze wens. Dit hebben we natuurlijk ook gedaan.

Onder het deel van OSTicketsytem kan u zien hoe wij deze instellingen gebruikt hebben en welke prioriteiten wij hebben ingesteld in ons systeem.

Pager rotation schedule

Voor het pager rotation schedule hebben we één pager op papier waar u duidelijk kan zien wie wanneer aanwezig zal zijn. Ook hebben we een google kalender waar u dezelfde informatie zal vinden maar dan in een andere omgeving.

We hebben dus besloten om tussen 08:00 en 20:00 steeds iemand beschikbaar te stellen. Bij grote problemen probeert de persoon met de permanentie andere teamleden te contacteren zodat het probleem makkelijker opgelost geraakt. Een persoon onderneemt nooit handelingen waarvan hij geen kennis heeft, dit om het systeem niet verder kapot te maken. De verdeling ziet er als volgt uit. Er is ook een google calender voorzien waarop wijzigingen kunnen worden aangebracht. Link google calender: [BIT.LY/2A9G3XT](https://bit.ly/2A9G3XT)

DAG VAN DE WEEK	VERANTWOORDELIJKE PERSOON
MAANDAG	WOUTER BOSMANS
DINSDAG	ROBIN BRUYNSEELS
WOENSDAG	PHILIPPE BOETS
DONDERDAG	PEI YANG
VRIJDAG	MANUEL HAYEN
ZATERDAG	JESSE VAN DONINCK
ZONDAG	BRENT BELIEN

Wanneer iemand onverwacht ziek is wordt er verwacht dat iemand deze dag tijdelijk overneemt. Deze persoon moet dit zo snel mogelijk melden zodat er tijdig een vervanger kan worden gevonden. Onderling wisselen is steeds mogelijk zolang er maar elke dag een permanentie is. Eens per maand wordt de verdeling aangepast zodat niet iedereen steeds dezelfde dag heeft.

Security policy and other policies

Voor onze written policies hebben wij gebruik gemaakt van het wiki platform, slimwiki. Het is zéér handig te beheren en zaken aan te passen en/of toe te voegen. Ook biedt slimwiki de mogelijkheid om accounts toe te voegen en in bijvoorbeeld teams op te delen. Zo kunnen we ook toegang geven aan klanten die betrekking hebben tot bepaalde policies en andere niet.



Op dit platform hebben wij een teamaccount gemaakt zodat iedereen hieraan kan meewerken. Ook hier vindt u alle policies die wij hebben gemaakt. Hieronder een overzicht van welke policies en andere zaken wij hebben uitgewerkt:

- Database of all machines
- Design doc
- Pager rotation
- Policy and procedure
- Post mortem
- Refresh policy
- Security policy
- The 3 empowering policies

Voor de volledig inhoud kan u met volgende link de policies bekijken.

<https://slimwiki.com/team-01>

Post mortem

Voor het post mortem process gebruiken we een template van PagerDuty. Hiermee kunnen we als er iets is mis gegaan of iets niet succesvol was analyseren in een meeting. Dit is bedoeld om het project in de toekomst te optimaliseren en risico's te beperken. Wij vinden dat dit een belangrijk onderdeel is van ons project omdat we graag op een efficiënte manier te werk gaan en onze risico's willen beperken. We bepalen hierin wat er is mis gegaan en hoe we dit kunnen verbeteren in de toekomst. We hebben volgend template twee keer gebruikt.

Postmortem Backup

Guidelines

This page is intended to be reviewed during a postmortem meeting that should be scheduled within 5 business days of any event. Your first step should be to schedule the postmortem meeting for within 5 business days after the incident. Don't wait until you've filled in the info to schedule the meeting, however, make sure the page is completed by the meeting.

Postmortem Owner:

Manuel Hayen

Meeting Scheduled for:

22/05/2020

Overview:

Problemen met het opzetten van een backup via Bacula.

What Happened:

We gingen onze backup doen via Bacula, dit is uiteindelijk misgelopen.

Root Causes:

We vermoeden dat we tijdens de installatie een fout hebben gemaakt en daardoor vast kwamen te zitten met de uiteindelijke implementatie ervan.

We hadden op voorhand wat meer research kunnen doen dan hadden we dit probleem niet gehad.

Resolution:

We zijn overgestapt op een andere backup software -> rsync.

Via rsync was dit allemaal gemakkelijker, het enige verschil was dat we geen user interface hebben maar dit is uiteindelijk geen probleem omdat rsync heel gemakkelijk te managen is.

Are backups automated

Zoals eerder vermeld hebben wij redelijk wat problemen gehad met het zoeken naar een sterk backupsysteem. Zo zijn we uiteindelijk terecht gekomen bij Rsync. Een sterk linux based applicatie die exacte copies neemt van de zaken die wij willen backupperen.

In het deel van Rsync automated backup vindt u meer uitleg over hoe wij hiermee te werk zijn gegaan.

Bronvermelding

<https://support.rackspace.com/how-to/manually-enable-auto-updates-on-debian-ubuntu-14.04-and-16.04-public-cloud-servers/>

<https://libre-software.net/ubuntu-automatic-updates/>

<https://www.freecodecamp.org/news/how-to-write-a-good-software-design-document-66fcf019569c/>

https://linuxhint.com/install_chkrootkit/

<https://linuxconfig.org/how-to-set-up-a-nfs-server-on-debian-10-buster/>

<https://www.howtogeek.com/135533/how-to-use-rsync-to-backup-your-data-on-linux/>

<https://www.itzgeek.com/how-tos/linux/debian/configure-dns-server-on-debian-9-ubuntu-16-04.html>

https://wiki.debian.org/NetworkConfiguration#The_resolv.conf_configuration_file

https://puppet.com/docs/puppet/latest/puppet_index.html

<https://www.cloudreach.com/en/resources/blog/ct-apache-docker-containers/>

<https://forums.docker.com/t/shared-web-hosting-with-docker-best-practices/7893>

<https://www.centlinux.com/2020/03/configure-lamp-stack-in-docker-containers.html>

https://linuxhint.com/lamp_server_docker/

https://postmortems.pagerduty.com/resources/post_mortem_template/

Bijlage/tutorials

1. Netwerk

1.1. Debian 10 buster netwerkconfiguratie

Eerst en vooral moeten we de versie waarmee we werken controleren.

```
admin01@T01PROD01:~$ hostnamectl
Static hostname: T01PROD01
Icon name: computer-vm
Chassis: vm
Machine ID: 9cd7dc590ffa4961937ad3a20f0e1dbf
Boot ID: f5130198d1b64693ab5d65379284c85b
Virtualization: vmware
Operating System: Debian GNU/Linux 10 (buster)
Kernel: Linux 4.19.0-8-amd64
Architecture: x86-64
```

Dus we werken wel degelijk met debian 10 buster.

Het volgende dat we doen is

```
admin01@T01PROD01:~$ ls /sys/class/net
ens192  lo
admin01@T01PROD01:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:9f:39:fc brd ff:ff:ff:ff:ff:ff
    inet 172.27.66.70/24 brd 172.27.66.255 scope global dynamic ens192
        valid_lft 4515sec preferred_lft 4515sec
    inet6 fe80::250:56ff:fe9f:39fc/64 scope link
        valid_lft forever preferred_lft forever
```

Met dat eerste commando `ls /sys/class/net` kunnen we onze netwerkinterfaces in een lijst krijgen.

Of met `"ip link show"` gaat dit ook...

En met `ip a` kunnen we zien waarvoor deze gebruikt worden op het moment.

Ens192 zal een statisch IP address krijgen.

Om dit te doen gaan we met een editor zoals nano naar `/etc/network/interfaces`.

`Sudo nano /etc/network/interfaces`

En dan passen we de configuratie bij de juiste interface (ens192) aan.

```

GNU nano 3.2 /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens192
iface ens192 inet dhcp

```

Pas aan naar wat relevant is.

```

GNU nano 3.2 /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens192
iface ens192 inet static
    address 172.27.66.111
    netmask 255.255.255.0
    gateway 172.27.66.254
    dns-nameservers 208.67.222.222 208.68.220.220

```

En nu herstarten we de service.

```
admin01@T01PROD01:~$ sudo systemctl restart networking
```

Nu heb ik persoonlijk nog een reboot gedaan maar dat zou niet persé nodig moeten zijn.

Maar als je nu ip a uitvoert zou je je statische ip instelling moeten zien!

```

admin01@T01PROD01:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:9f:39:fc brd ff:ff:ff:ff:ff:ff
    inet 172.27.66.111/24 brd 172.27.66.255 scope global ens192
        valid_lft forever preferred_lft forever
    inet6 fe80::250:56ff:fe9f:39fc/64 scope link
        valid_lft forever preferred_lft forever

```

Nu kan je google eens pingen om te zien of het echt werkt.

```
admin01@T01PROD01:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=55 time=7.22 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=55 time=6.63 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=55 time=6.48 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 6ms
rtt min/avg/max/mdev = 6.480/6.775/7.219/0.319 ms
```

Als je je eigen dns instelt pas dit dan aan in de interfaces file EN in de /etc/resolv.conf file, hier is een voorbeeld.

```
admin01@T01PROD01: ~
File Edit View Search Terminal Help
GNU nano 3.2 /etc/resolv.conf

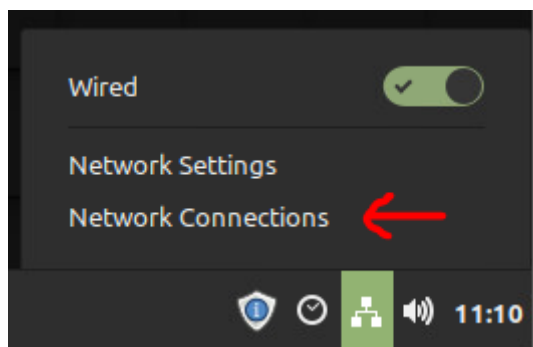
domain team01.local
search team01.local
nameserver 172.27.66.115
```

1.2. Linux mint 19.3 GUI netwerkconfiguratie

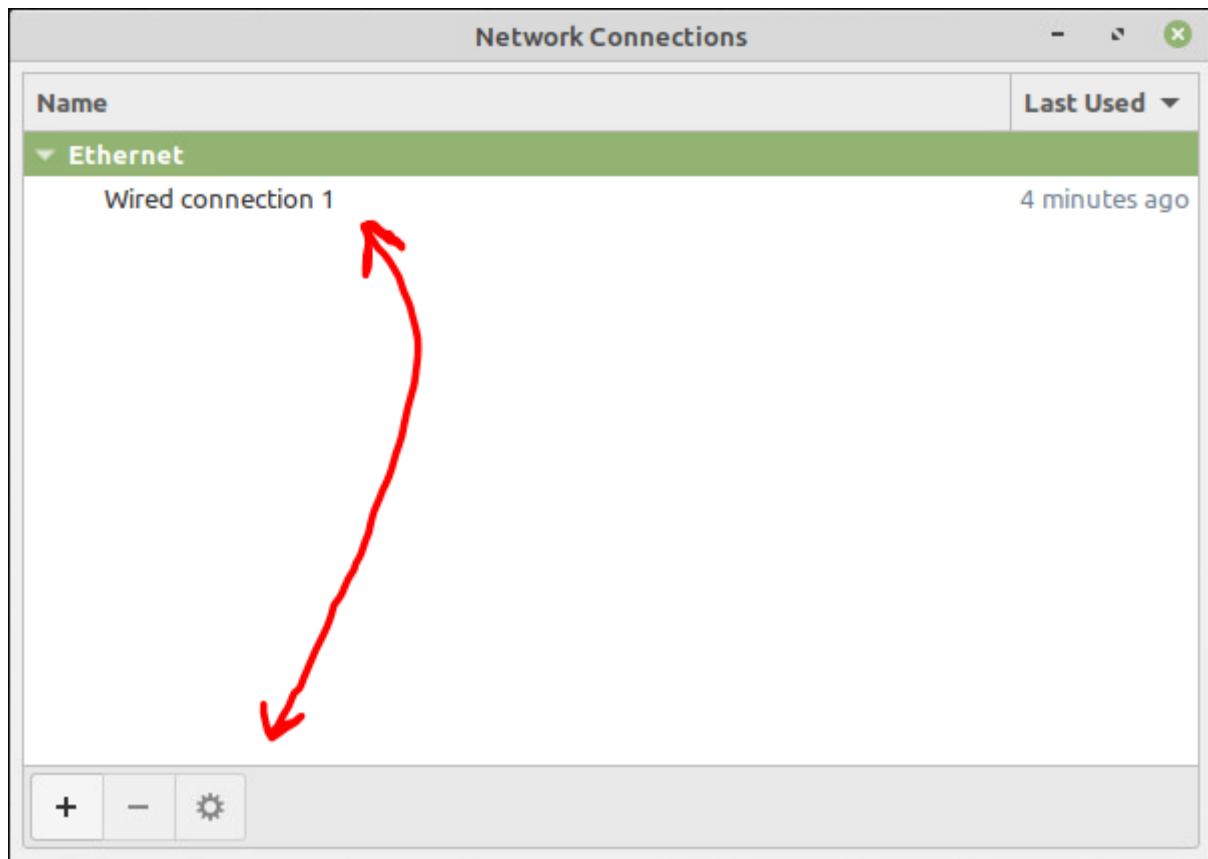
Om de versie waarmee we werken te checken kunnen we het commando herhalen van de vorige tutorial in de terminal.

Om te beginnen moeten we naar "network connections" gaan in de GUI.

Je kan dit ofwel typen in de zoek functie textbox of je kan rechts vanonder in de taakbalk net zoals bij Windows op het volgende icoontje drukken en het daar openen.



Selecteer in het volgende scherm de wired connection en druk op edit.



In het scherm dat daarop volgt gaan we naar de IPv4 Settings tab en passen we eerst en vooral de method aan naar Manual, daarna drukken we op de Add knop om een ip te adden en dergelijke.

Vervolgens kunnen we de DNS servers aanpassen.

Save wanneer je klaar bent.

Editing Wired connection 1

Connection name:

General Ethernet 802.1X Security DCB Proxy **IPv4 Settings** IPv6 Settings

Method:

Addresses

Address	Netmask	Gateway
172.27.66.110	255.255.255.0	172.27.66.254

DNS servers:

Search domains:

DHCP client ID:

☐ Require IPv4 addressing for this connection to complete

Ten slotte kunnen we weer testen na een reboot.

```

admin01@T01MGMT01: ~
File Edit View Search Terminal Help
admin01@T01MGMT01:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:9f:92:e9 brd ff:ff:ff:ff:ff:ff
    inet 172.27.66.110/24 brd 172.27.66.255 scope global noprefixroute ens160
        valid lft forever preferred_lft forever
    inet6 fe80::c101:1c64:290:63dc/64 scope link noprefixroute
        valid lft forever preferred_lft forever
admin01@T01MGMT01:~$ ping google.com
PING google.com (172.217.17.110) 56(84) bytes of data.
64 bytes from ams15s29-in-f14.1e100.net (172.217.17.110): icmp_seq=1 ttl=55 time=7.17 ms
64 bytes from ams15s29-in-f14.1e100.net (172.217.17.110): icmp_seq=2 ttl=55 time=6.68 ms
64 bytes from ams15s29-in-f14.1e100.net (172.217.17.110): icmp_seq=3 ttl=55 time=7.29 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 6.680/7.049/7.292/0.282 ms

```

1.3. DNS

Aangezien DNS een vrij uitgebreid onderwerp is gaan wij hier enkel over de essentiële informatie.

T01PROD05 dient bij ons als de DNS server, en we gebruiken bind9.

```
admin05@T01PROD05:~$ sudo named -v
[sudo] password for admin05:
BIND 9.11.5-P4-5.1+deb10u1-Debian (Extended Support Version) <id:998753c>
```

Vergeet nooit de systemd-resolved, en bind9.service services te restarten na veranderingen!

Onze named.conf.local instellingen zijn als volgt.

```
admin05@T01PROD05: /etc
File Edit View Search Terminal Help
GNU nano 3.2 /etc/bind/named.conf.local

//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

// (C) 2020 Wouter Bosmans 06.05.2020

zone "team01.local" IN {
type master; //primary zone
file "/etc/bind/forward.team01.local.db"; //Forward lookup database file
allow-update {none;}; //geen externe DNS updates toestaan!
};

zone "0.66.27.172.in-addr.arpa" IN {
type master; //primary zone
```

De forward zonefile ziet er als volgt uit om maar een idee te geven van hoe hierop kan uitgebreid worden. Deze is constant voor verandering vatbaar. Vergeet nooit het serial nummer te verhogen na aanpassingen!

```
admin05@T01PROD05: /etc/bind
File Edit View Search Terminal Help
GNU nano 3.2 forward.team01.local.db

;
; BIND data file for local loopback interface
;
$TTL 604800
@ IN SOA ns1.team01.local. root.team01.local. (
    11 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL

@ IN NS ns1.team01.local.
ns1 IN A 172.27.66.115
t01mgmt01 IN A 172.27.66.110
t01prod01 IN A 172.27.66.111
t01prod02 IN A 172.27.66.112
t01prod03 IN A 172.27.66.113
t01prod04 IN A 172.27.66.114
t01prod05 IN A 172.27.66.115
ftp IN CNAME t01prod05.team01.local.
portainer IN CNAME t01prod05.team01.local.
puppet IN CNAME t01mgmt01.team01.local.
osticket IN CNAME t01mgmt01.team01.local.
zabbix IN CNAME t01prod01.team01.local.
bacula IN CNAME t01prod05.team01.local.
www.josbriersnaaimachines IN CNAME t01prod02.team01.local.
josbriersnaaimachines IN CNAME t01prod02.team01.local.
www.easy-hosting IN CNAME t01prod02.team01.local.
easy-hosting IN CNAME t01prod02.team01.local.
wordpress IN CNAME t01prod05.team01.local.

; (C) 2020 Wouter Bosmans r0737827. All rights reserved!
```

En dan de inhoud van de reverse zonefile.

```
admin05@T01PROD05: /etc/bind
File Edit View Search Terminal Help
GNU nano 3.2 reverse.team01.local.db

;
; BIND reverse data file for local loopback interface
;
$TTL 604800
@      IN      SOA      team01.local. root.team01.local. (
; Serial
                        3          ; Refresh
                        604800     ; Retry
                        86400      ; Expire
                        2419200    ; Negative Cache TTL
                        604800 )
@      IN      NS       ns1.team01.local.
110    IN      PTR      t01mgmt01.team01.local.
111    IN      PTR      t01prod01.team01.local.
112    IN      PTR      t01prod02.team01.local.
113    IN      PTR      t01prod03.team01.local.
114    IN      PTR      t01prod04.team01.local.
115    IN      PTR      t01prod05.team01.local.
```

Vergeet niet de rechten correct te zetten (chmod)

```
admin05@T01PROD05:/etc/bind$ ls -l
total 56
-rw-r--r-- 1 root root 2761 Jun 21 2019 bind.keys
-rw-r--r-- 1 root root 237 Jun 21 2019 db.0
-rw-r--r-- 1 root root 271 Jun 21 2019 db.127
-rw-r--r-- 1 root root 237 Jun 21 2019 db.255
-rw-r--r-- 1 root root 353 Jun 21 2019 db.empty
-rw-r--r-- 1 root root 270 Jun 21 2019 db.local
-rw-r--r-- 1 root bind 1021 May 21 19:36 forward.team01.local.db
-rw-r--r-- 1 root bind 463 Jun 21 2019 named.conf
-rw-r--r-- 1 root bind 498 Jun 21 2019 named.conf.default-zones
-rw-r--r-- 1 root bind 596 May 17 13:42 named.conf.local
-rw-r--r-- 1 root bind 846 Jun 21 2019 named.conf.options
-rw-r--r-- 1 root bind 472 May 6 22:16 reverse.team01.local.db
-rw-r----- 1 bind bind 77 May 6 21:42 rndc.key
-rw-r--r-- 1 root root 1317 Jun 21 2019 zones.rfc1918
```

Restart de service en test eventueel met nslookup.

Als er problemen zijn zoals DNS caching voer dan een flush uit en restart de services.

Sudo systemctl restart bind9.service

Sudo system-resolve --flush-caches

Sudo service system-resolved restart

2. Puppet

2.1. Puppet master

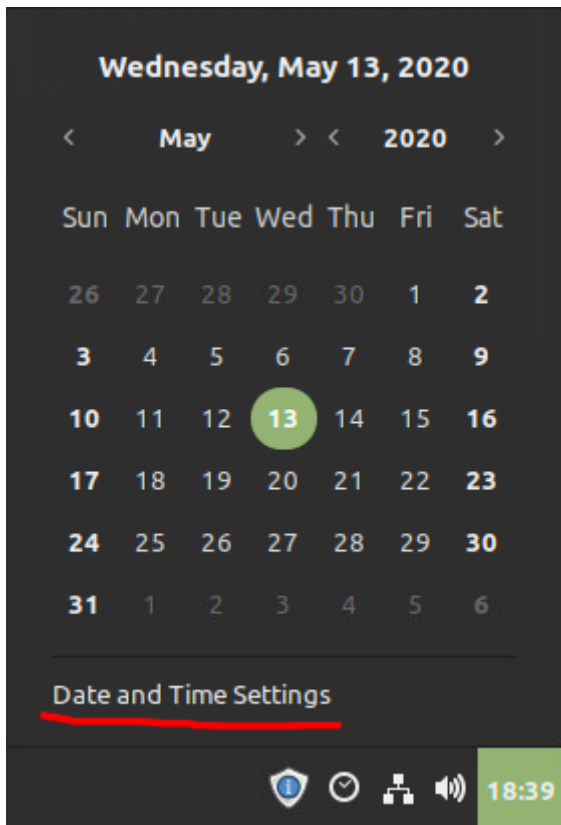
We gaan onze management server klaarzetten als puppet master.

Om dit te doen gaan we als volgt tewerk.

We kunnen de date & time settings juist zetten aan de hand van commando's in de terminal, maar dit is niet nodig op deze VM met GUI dus zetten we deze juist via de tijd instelling die je rechts vanonder terugvind.



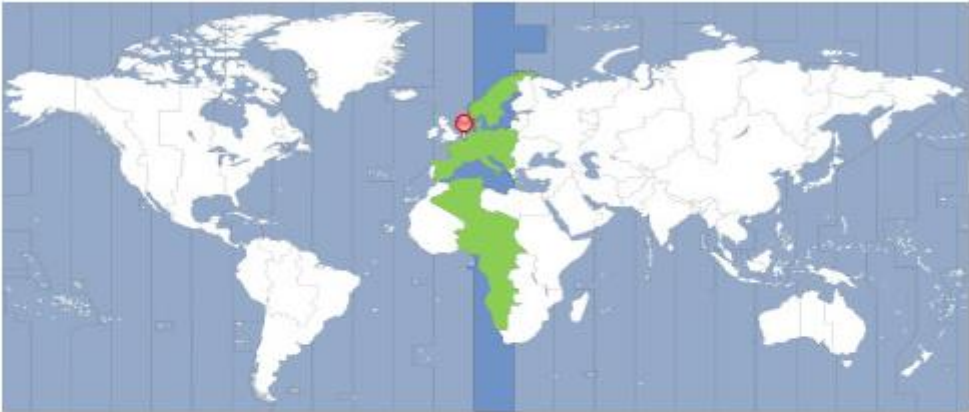
Druk vervolgens op Date and Time Settings.



Pas de instellingen aan als dit nodig is want deze moeten overeen komen op de VMs.

Date & Time

Date and Time



Region Europe ▼ City Brussels ▼

Network time ✓

Format

Use 24h clock ✓

Display the date ×

Display seconds ×

First day of week Use locale default ▼

Pas de /etc/hosts file aan naar horen (sudo nano).

```
admin01@T01MGMT01: ~  
File Edit View Search Terminal Help  
GNU nano 2.9.3 /etc/hosts Modified  
127.0.0.1 localhost  
172.27.66.110 T01MGMT01  
172.27.66.111 T01PROD01  
172.27.66.112 T01PROD02  
172.27.66.113 T01PROD03  
172.27.66.114 T01PROD04  
172.27.66.115 T01PROD05  
  
# The following lines are desirable for IPv6 capable hosts  
::1 ip6-localhost ip6-loopback  
fe00::0 ip6-localnet  
ff00::0 ip6-mcastprefix  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Reboot hierna en begin dan de puppet master installatie.

```
admin01@T01MGMT01:~$ sudo apt-get install puppetmaster
```

We passen vervolgens de /etc/puppet/puppet.conf file aan naar horen.

```
admin01@T01MGMT01: ~  
File Edit View Search Terminal Help  
GNU nano 2.9.3 /etc/puppet/puppet.conf  
[main]  
ssldir = /var/lib/puppet/ssl  
certname = t01mgmt01  
server = T01MGMT01  
environment = production  
  
[master]  
vardir = /var/lib/puppet  
cadir = /var/lib/puppet/ssl/ca  
dns_alt_names = T01MGMT01
```

Nu kunnen we de service starten via

```
admin01@T01MGMT01:~$ sudo systemctl start puppet-master  
admin01@T01MGMT01:~$ sudo systemctl enable puppet-master  
Synchronizing state of puppet-master.service with SysV service script  
with /lib/systemd/systemd-sysv-install.  
Executing: /lib/systemd/systemd-sysv-install enable puppet-master
```


2.2 puppet client

Zoals voordien moeten we eerst en vooral de datum en tijd weer correct zetten (deze moeten overeenkomen tussen de VMs) en de /etc/hosts file aanpassen. Maar dit keer via commands, en installatie van ntpdate.

```
admin02@T01PROD02:~$ sudo apt-get install ntp ntpdate
```

```
admin02@T01PROD02:~$ sudo ntpdate -u 0.ubuntu.pool.ntp.org
```

```
admin02@T01PROD02:~$ sudo timedatectl set-timezone Europe/Brussels
```

```
admin02@T01PROD02: ~
File Edit View Search Terminal Help
GNU nano 3.2 /etc/hosts
127.0.0.1    localhost
172.27.66.110 T01MGMT01
172.27.66.111 T01PROD01
172.27.66.112 T01PROD02.prod.team01.local T01PROD02
172.27.66.113 T01PROD03
172.27.66.114 T01PROD04
172.27.66.115 T01PROD05

# The following lines are desirable for IPv6 capable hosts
::1    localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Nu kunnen we puppet op de client installeren.

```
admin02@T01PROD02:~$ sudo apt-get install puppet
```

En passen we weer de configuratie file aan naar horen

```
admin02@T01PROD02: ~
File Edit View Search Terminal Help
GNU nano 3.2 /etc/puppet/puppet.conf

[main]
ssldir = /var/lib/puppet/ssl
certname = t01prod02
server = T01MGMT01
environment = production

[agent]
certname = t01prod02
```

Hierna rebooten we de client en pingen we naar de puppet master voor een test.

```
admin02@T01PROD02:~$ ping t01mgmt01
PING T01MGMT01 (172.27.66.110) 56(84) bytes of data.
64 bytes from T01MGMT01 (172.27.66.110): icmp_seq=1 ttl=64 time=0.274 ms
64 bytes from T01MGMT01 (172.27.66.110): icmp_seq=2 ttl=64 time=0.663 ms
^C
--- T01MGMT01 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 14ms
rtt min/avg/max/mdev = 0.274/0.468/0.663/0.195 ms
```

Om de puppet service vervolgens te starten op de client doen we

```
admin02@T01PROD02:~$ sudo /usr/bin/puppet resource service puppet ensure=running
enable=true
[sudo] password for admin02:
2020-05-13 21:23:42.022002 WARN puppetlabs.facter - locale environment variable
s were bad; continuing with LANG=C LC_ALL=C
Notice: /Service[puppet]/ensure: ensure changed 'stopped' to 'running'
service { 'puppet':
  ensure => 'running',
  enable => 'true',
}
```

Nu wordt er een certificate naar de puppetmaster gestuurd dat we daar moeten signen (ik heb dit al voor meerdere gedaan dus krijg ik hier al een relatieve lijst).

```
admin01@T01MGMT01:~$ sudo /usr/bin/puppet cert list --all
"t01prod01" (SHA256) 6A:66:7B:A4:0C:0D:EB:A9:2F:EA:41:42:BD:0C:9B:B5:DB:75:EC:63:07:9B:1B:15:53:AB:A1:89:27:63:94:0B
"t01prod02" (SHA256) 5D:4C:B9:5B:3B:85:78:6C:8A:87:97:96:E5:DE:90:F5:33:96:94:54:17:37:68:61:66:22:44:C8:F0:85:90:7A
"t01prod03" (SHA256) 1B:23:37:D9:F8:C4:3A:2D:AB:96:F9:FF:CB:26:6D:1C:1F:4C:74:03:A3:8D:25:BD:08:92:1E:5C:01:71:41:5B
"t01prod04" (SHA256) 85:C5:6E:F2:DF:90:41:D1:91:8A:4F:BF:66:52:00:CB:5B:D4:E6:A3:49:E5:E4:8B:E2:AB:48:AF:A1:FF:27:BE
"t01prod05" (SHA256) 7A:60:71:8B:46:FA:8D:AD:14:74:A7:DD:48:68:C0:5D:3C:96:AB:C6:3B:96:E0:5F:3A:14:FA:4E:BC:4E:26:0F
+ "t01mgmt01" (SHA256) 96:A1:17:F0:6F:C5:55:EE:02:8E:08:A5:F7:11:7A:56:93:C5:1B:A7:45:13:8D:A0:13:E3:C4:A2:AA:35:1F:7F
(alt names: "DNS:puppet", "DNS:t01mgmt01")
```

Om alle certificates te signen doe ik

```
admin01@T01MGMT01:~$ sudo /usr/bin/puppet cert sign --all
[sudo] password for admin01:
Signing Certificate Request for:
"t01prod01" (SHA256) 6A:66:7B:A4:0C:0D:EB:A9:2F:EA:41:42:BD:0C:9B:B5:DB:75:EC:63:07:9B:1B:15:53:AB:A1:89:27:63:94:0B
Notice: Signed certificate request for t01prod01
Notice: Removing file Puppet::SSL::CertificateRequest t01prod01 at '/var/lib/puppet/ssl/ca/requests/t01prod01.pem'
Signing Certificate Request for:
"t01prod02" (SHA256) 5D:4C:B9:5B:3B:85:78:6C:8A:87:97:96:E5:DE:90:F5:33:96:94:54:17:37:68:61:66:22:44:C8:F0:85:90:7A
Notice: Signed certificate request for t01prod02
Notice: Removing file Puppet::SSL::CertificateRequest t01prod02 at '/var/lib/puppet/ssl/ca/requests/t01prod02.pem'
Signing Certificate Request for:
"t01prod04" (SHA256) 85:C5:6E:F2:DF:90:41:D1:91:8A:4F:BF:66:52:00:CB:5B:D4:E6:A3:49:E5:E4:8B:E2:AB:48:AF:A1:FF:27:BE
Notice: Signed certificate request for t01prod04
Notice: Removing file Puppet::SSL::CertificateRequest t01prod04 at '/var/lib/puppet/ssl/ca/requests/t01prod04.pem'
Signing Certificate Request for:
"t01prod05" (SHA256) 7A:60:71:8B:46:FA:8D:AD:14:74:A7:DD:48:68:C0:5D:3C:96:AB:C6:3B:96:E0:5F:3A:14:FA:4E:BC:4E:26:0F
Notice: Signed certificate request for t01prod05
Notice: Removing file Puppet::SSL::CertificateRequest t01prod05 at '/var/lib/puppet/ssl/ca/requests/t01prod05.pem'
Signing Certificate Request for:
"t01prod03" (SHA256) 1B:23:37:D9:F8:C4:3A:2D:AB:96:F9:FF:CB:26:6D:1C:1F:4C:74:03:A3:8D:25:BD:08:92:1E:5C:01:71:41:5B
Notice: Signed certificate request for t01prod03
Notice: Removing file Puppet::SSL::CertificateRequest t01prod03 at '/var/lib/puppet/ssl/ca/requests/t01prod03.pem'
```

2.3 configuratie puppet

Voor onze configuratie hebben we gebruikgemaakt van modules op puppet forge met namen "cesnet/autoupdate" en "puppet/cron".

Om deze te installeren en te gebruiken kunnen we gewoon de instructies op de website volgen.



cesnet/autoupdate by: CESNET

Enable autoupdates

[Project URL](#) [RSS Feed](#)

Latest version is compatible with:

- [RedHat](#), [Ubuntu](#), [Debian](#), [Scientific](#), [CentOS](#), [Fedora](#)

Start using this module:

- [new! Bolt](#)
- [r10k or Code Manager](#)
- [Manual installation](#)

Tags: [apt](#), [yum](#), [update](#), [unattended](#)

[Report issues](#)

Version **1.0.3** released **Feb 25th 2020**

[README](#) [Dependencies](#) [Compatibility](#) [License](#) [Scores](#) [Issues \(1\)](#)

autoupdate

build [passing](#)



puppet/cron by: Vox Pupuli

Module to manage cron jobs via /etc/cron.d/

[Project URL](#) [RSS Feed](#)

Latest version is compatible with:

- Puppet Enterprise 2019.7.x, 2019.5.x, 2019.4.x, 2019.3.x, 2019.2.x, 2019.1.x, 2019.0.x, 2018.1.x
- Puppet >= 5.5.8 < 7.0.0
- [RedHat](#), [Debian](#), [SLES](#), [Scientific](#), [CentOS](#), [Gentoo](#)

Start using this module:

- [new! Bolt](#)
- [r10k or Code Manager](#)
- [Manual installation](#)

Tags: [cron](#)

[Report issues](#)

Version **2.0.0** released **Jan 13th 2020**

[README](#) [Changelog](#) [Compatibility](#) [License](#) [Scores](#) [Issues \(10\)](#)

Puppet Cron Module

license: [Apache-2.0](#) build: [passing](#) puppetforge: [v2.0.0](#) downloads: [401k](#) score: [unknown](#)

Deze eerste module zal net zoals het in de naam zegt automatische updates uitvoeren voor ons op een dagelijkse basis. Op een uur wanneer het niemand zou storen, we gaan dit volledig naar onze wensen configureren.

Van de tweede module gaan we gebruikmaken om onze docker omgeving schoon te houden, zodat ongebruikte docker containers opgeruimd worden.

We gaan voordat we met de implementatie beginnen, op elk systeem cron-apt installeren met

Sudo apt-get install cron-apt

Eens dit gedaan is kunnen we beginnen met op de puppet master de modules te installeren.

Manual installation

Manually install this module with Puppet module tool:

```
puppet module install cesnet-autoupdate --version 1.0.3
```

Manual installation

Manually install this module with Puppet module tool:

```
puppet module install puppet-cron --version 2.0.0
```

En dan gaan we deze modules implementeren, we kunnen de modules hier terugvinden.

```
admin01@T01MGMT01:/etc/puppet/code/environments/production/modules$ ls
autoupdate  cron  ntp  stdlib  yum_autoupdate
```

In de /production/manifests folder (maak deze aan als je deze nog niet had met mkdir) maken we een site.pp file aan waarmee we die module oproepen en voor gebruik toepassen voor onze productiemachines.

```
admin01@T01MGMT01: /etc/puppet/code/environments/production
File Edit View Search Terminal Help
GNU nano 2.9.3 manifests/site.pp

node 't01prod01.team01.local','t01prod02.team01.local','t01prod03.team01.local','t01prod04.team01.local','t01prod05.team01.local' {
  class { 'autoupdate':
    email    => 'team01thebest@outlook.be',
    hour     => 1,
    minute   => 0,
    randomwait => 3000,
  }
  include cron
  cron::job { 'cleanup_docker':
    minute   => '4',
    hour     => '*',
    date     => '*',
    command  => 'docker system prune -a -f',
    user     => 'root'
  }
}
```

In het kort staat hier dat we de module toepassen op de 5 productiemachines, er zal een mail gestuurd worden naar ons dummy mail account wanneer er effectief dingen updates krijgen. Dit zal ergens gebeuren om 1u 's nachts (hour, minute parameters). En dit zal op een willekeurig tijdstip in de volgende 40 minuten vanaf die tijd gebeuren.

Daarnaast zal de cleanup_docker job elk uur alle docker containers en images die niet meer gebruikt worden opruimen.

We kunnen nu gewoon de puppet service zijn werk laten doen of we kunnen de veranderingen al binnen halen op een agent door daar een test uit te voeren.

```
admin01@T01PROD01: ~
File Edit View Search Terminal Help
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/T01PROD01--vg-var 3.9G  2.4G  1.3G   67% /var

admin01@T01PROD01:~$ sudo puppet agent -t
2020-05-24 02:06:23.922959 WARN  puppetlabs.facter - locale environment variables were bad; continuing with
LANG=C LC_ALL=C
Info: Using configured environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Retrieving locales
Info: Loading facts
Info: Caching catalog for t01prod01.team01.local
Info: Applying configuration version '1590278788'
Notice: /Stage[main]/Main/Node[t01prod01.team01.local]/Cron::Job[cleanup_docker]/File[job_cleanup_docker]/c
ontent:
--- /etc/cron.d/cleanup_docker  2020-05-24 02:03:22.221825459 +0200
+++ /tmp/puppet-file20200524-31980-199gut7      2020-05-24 02:06:28.509993243 +0200
@@ -7,4 +7,4 @@
# Environment Settings

# Job Definition
-4 * * * * root  docker system prune -a -f > "/mnt/team01_backups/docker_cleanup_($hostname -s).txt"
+4 * * * * root  docker system prune -a -f

Info: Computing checksum on file /etc/cron.d/cleanup_docker
Info: /Stage[main]/Main/Node[t01prod01.team01.local]/Cron::Job[cleanup_docker]/File[job_cleanup_docker]: Fi
lebucketed /etc/cron.d/cleanup_docker to puppet with sum 1bf9ba99fa660f811a7a2cfd1daecdff
Notice: /Stage[main]/Main/Node[t01prod01.team01.local]/Cron::Job[cleanup_docker]/File[job_cleanup_docker]/c
ontent: content changed '{md5}1bf9ba99fa660f811a7a2cfd1daecdff' to '{md5}fc572838e95e756066c7f259c1d8857f'
Notice: Applied catalog in 0.24 seconds
```


2.4 Troubleshooting

Als je puppetmaster installeert op een machine maar later beslist van deze toch op andere machine te willen draaien, dan kan het zijn dat er conflicten voorkomen tijdens het versturen van certificaten en de machine zal missen in de certificate lijst, om dit op te lossen zijn we als volgt tewerk gegaan...

```
admin01@T01PROD01:~$ sudo nano /etc/puppet/puppet.conf
admin01@T01PROD01:~$ sudo puppet agent --test
2020-05-14 21:17:36.808945 WARN puppetlabs.facter - locale environment variables were bad; continuing with LANG=C LC_ALL=C
Warning: Unable to fetch my node definition, but the agent run will continue:
Warning: SSL_connect returned=1 errno=0 state=error: certificate verify failed (self signed certificate in certificate chain): [self signed certificate in certificate chain for /CN=Puppet CA: t01mgmt01]
Info: Retrieving pluginfacts
Error: /File[/var/cache/puppet/facts.d]: Failed to generate additional resources using 'eval_generate': SSL_connect returned=1 errno=0 state=error: certificate verify failed (self signed certificate in certificate chain): [self signed certificate in certificate chain for /CN=Puppet CA: t01mgmt01]
Error: /File[/var/cache/puppet/facts.d]: Could not evaluate: Could not retrieve file metadata for puppet:///pluginfacts: SSL_connect returned=1 errno=0 state=error: certificate verify failed (self signed certificate in certificate chain): [self signed certificate in certificate chain for /CN=Puppet CA: t01mgmt01]
Info: Retrieving plugin
Error: /File[/var/cache/puppet/lib]: Failed to generate additional resources using 'eval_generate': SSL_connect returned=1 errno=0 state=error: certificate verify failed (self signed certificate in certificate chain): [self signed certificate in certificate chain for /CN=Puppet CA: t01mgmt01]
Error: /File[/var/cache/puppet/lib]: Could not evaluate: Could not retrieve file metadata for puppet:///plugins: SSL_connect returned=1 errno=0 state=error: certificate verify failed (self signed certificate in certificate chain): [self signed certificate in certificate chain for /CN=Puppet CA: t01mgmt01]
Error: Could not retrieve catalog from remote server: SSL_connect returned=1 errno=0 state=error: certificate verify failed (self signed certificate in certificate chain): [self signed certificate in certificate chain for /CN=Puppet CA: t01mgmt01]
Warning: Not using cache on failed catalog
Error: Could not retrieve catalog; skipping run
Error: Could not send report: SSL_connect returned=1 errno=0 state=error: certificate verify failed (self signed certificate in certificate chain): [self signed certificate in certificate chain for /CN=Puppet CA: t01mgmt01]
admin01@T01PROD01:~$ sudo nano /etc/puppet/puppet.conf
admin01@T01PROD01:~$ sudo find /var/lib/puppet/ssl -name puppet.pem -delete
admin01@T01PROD01:~$ sudo find /var/lib/puppet/ssl -name t01prod01.pem -delete
admin01@T01PROD01:~$ sudo find /var/lib/puppet/ssl -name t01mgmt01.pem -delete
admin01@T01PROD01:~$ sudo puppet agent --test
2020-05-14 21:19:40.346032 WARN puppetlabs.facter - locale environment variables were bad; continuing with LANG=C LC_ALL=C
Info: Creating a new SSL key for t01prod01
Info: csr attributes file loading from /etc/puppet/csr_attributes.yaml
Info: Creating a new SSL certificate request for t01prod01
Info: Certificate Request fingerprint (SHA256): 6A:66:7B:A4:0C:0D:E8:A9:2F:EA:41:42:BD:0C:9B:85:DB:75:EC:63:07:9B:1B:15:53:AB:A1:89:27:63:94:0B
Exiting; no certificate found and waitforcert is disabled
admin01@T01PROD01:~$
```

We moesten oude certificate files dankzij de eerste puppetmaster installatie eerst verwijderen die de conflicten veroorzaakten anders kon er nooit een nieuw certificate van deze machine aankomen bij de nieuwe master.

Als er foute certificates zijn opgeslagen dan mag je de /var/lib/puppet/ssl directory volledig opkuisen op de agent, en doe je even een "sudo puppet cert clean <certnaam van de puppet agent in kwestie>". Vervolgens doe je opnieuw "sudo puppet agent -test" op de agent zodat deze een nieuwe certificate kan genereren, en dan sign je deze opnieuw op de master.

Let ook op het gebruik van dns, geef altijd de volledige dns naam op in de puppet.conf file op elke vm wanneer je verwijst naar dingen zoals "server" en "certname".

2. Backup

1. We installeren op alle VM's de rsync applicatie:

```
sudo apt-get install rsync
```

2. De tweede stap was het maken van een mountpoint. Dit deden we in de fstab file. We hebben een mount point gemaakt op een harddisk van 1TB die op de prod05 draait als backup locatie. Deze delen we dan

in de prods als volgt: /mnt/team01_backups/pord01-5. Zo zien we een duidelijk onderscheid tussen de verschillende back ups.

3. Verder gaan we onze backup al één maal handmatig laten lopen. Rsync biedt de mogelijkheid een volledige kopie te maken (backup). Ook hier zijn de mogelijkheden om het extern buiten onze Virtual omgeving op te slagen.

Rsync biedt vanzelf de incrementele optie aan. Het gaat kijken naar eventuele veranderingen die plaats vonden in het resource path.

We doen dit met het commando:

```
sudo rsync -aAXv --  
exclude={"/dev/*","/proc/*","/sys/*","/tmp/*","/run/*","/mnt/*","/media/*","/lost+found"} /path/to/backup/folder
```

Mogelijke problemen: Problemen die wij op ons pad tegenkwamen waren vooral de permissies die niet goed stonden op de directory en files zelf (/backupdisk/prod01-5)

Quick reminder: Het laatste stuk '/path/to/backup/folder' moeten we dus als volgt aanpassen adhv de productie VM's /mnt/team01_backups/pord01-5.

4. Ten slotte was de opdracht om het backup automatisch te laten lopen. Hier gaan we de crontab optie voor gebruiken. Hier kunnen we kiezen wanneer we dit laten uitvoeren. We kiezen om het om 22h te laten backup'en.

Hier ziet u welke code dit gaat doen:

```
- crontab -e
```

```
- 0 22 * * * sudo rsync -aAXv --  
exclude={"/dev/*","/proc/*","/sys/*","/tmp/*","/run/*","/mnt/*","/media/*","/lost+found"} /path/to/backup/folder
```

5. Later bij uitbreiding brengen we in de pager rotation dat er na elke maand (30dagen), zips gemaakt worden. Om capaciteit onder controle te houden geven we de klant dan de keuze om dit al dan niet bij te houden.

3. ClamAV en chkrootkit

1. We beginnen met een update van het systeem. Dit is belangrijk om te kijken of alles up to date is.

```
sudo apt-get update
```

```
sudo apt upgrade
```

2. Nu alles up to date is gaan we over naar de installatie. ClamAV is een commandline based applicatie die makkelijk te beheersen is. Verder ziet u de stappen om ClamAV te installeren.

```
sudo apt-get install clamav clamav-freshclam
```

We moeten ook zeker zijn dat de applicatie running is. Dit kan u op volgende manieren doen.

```
sudo service clamav-freshclam start
```

```
sudo service clamav-freshclam status
```

3. Het handige aan clamav is dat deze al door de default instellingen het systeem scant om het uur. Dit gebeurt natuurlijk op de achtergrond zonder dat we er last van hebben.

DE /etc/clamav/freshclam.conf staat op default op 24. Dit betekent dat het om het uur een scan uitvoert.

Maar dit is ook aan te passen dmv volgende werkwijze:

```
sudo nano /etc/clamav/freshclam.conf
```


--> we veranderen de '24', Naar bv. 1, zodaat het 1 maal per dag scant.

4. Het is ook mogelijk om een handmatige scan uit te voeren. Dit kan zowel voor heel het systeem als voor een bepaalde directory.

```
freshclam -v
```

Chrootkit

=====

We hebben gekozen om naast clamAv ook chkrootkit te installeren. Deze twee vullen elkaar heel mooi aan. Zo gebruiken we chkrootkit om malware lokaal te scannen. Terwijl we

ClamAV gebruiken als antivirus om gevaren als trojans, virussen,...

Ook hier hebben we een stappenplan gevolgt dat u hier zult kunnen volgen/nakijken.

1. Ook hier beginnen we met een update van het systeem.

```
Sudo apt-get update
```

```
sudo apt upgrade
```

2. We gaan nu over naar de installatie van de applicatie.

```
sudo apt install chkrootkit -y
```

3. Eens we de install gedaan hebben gaan we chkrootkit runnen. Dit doen we door het volgende commando in te geven:

```
sudo chkrootkit
```

4. In tegenstelling tot ClamAV runt chkrootkit enkel manueel door de default instellingen. Maar dit is aanpasbaar in de /etc/chkrootkit.conf file.

```
sudo nano /etc/chkrootkit
```

```
run_daily="false" --> run_daily="true"
```

Quick reminder: We runnen alle commando's met users. Als je de installaties runt als root moet je het 'sudo' deel niet invoegen.

4. Nfs installatie

Team01 Robin Bruynseels & Wouter Bosmans 19.05.2020

```
t01prod05:/srv/webserverroot /mnt/webserverroot nfs4 auto 0 0
```

```
t01prod05:/backupdisk /mnt/team01_backups nfs4 auto 0 0
```

```
=====
```

Wij hebben gewerkt met nfs4. Wouter Bosmans had hier veel ervaring mee en heeft dit mooi en succesvol geïmplementeerd.

Quick reminder: We hebben dezelfde methode gebruikt om de backups op te slaan in de /mnt/team01_backups.

T01PROD05:

1. mkdir /srv/webserverroot
2. install nfs server components apt-get
3. nfs share /srv/webserverroot 172.27.66.0/24 rw ...

T01MGMT01:

- 1 install nfs client components apt-get (nfs-common)

2. mount temporarily /mnt/webserverroot ==>
t01prod05.team01.local:/srv/webserverroot
3. automount in /etc/fstab => persistent mount by every reboot
4. make shortcut to bookmarks, desktop, ... for easy use

T01PROD01-04:

Oorspronkelijk plan: mount vanuit Portainer in docker swarm naar NFS share

=> werkt nog steeds, maar we hebben het anders gedaan

Nieuw plan: mount on every production server /mnt/webserverroot ==>
t01prod05.team01.local:/srv/webserverroot via /etc/fstab

=> gevolg: in iedere host (onder docker/swarm) is pad /mnt/webserverroot
beschikbaar, afgeleid naar server t01prod05

=> gevolg in portainer/docker/swarm: je kan in containers een pad van de host
"mounten" op een pad in de container

=> eindgevolg:

- a) in container: /var/www/webserverroot => /mnt/webserverroot (op host)
- b) in docker host: /mnt/webserverroot => t01prod05:/srv/webserverroot
(op t01prod05)
- c) in t01prod05: nfs share export EXT4 op /srv/webserverroot voor toegang
tot directory via netwerk

=====

T01PROD05 backupdisk:

EXT4 partition op /dev/sdb

1 TB - thin provisioned

mounted in /etc/fstab as /backupdisk => /dev/sdb

permissions chmod 777 /dev/sdb

T01MGMT01 + T01PROD01 t.e.m. T01PROD04

sudo nano /etc/fstab

```
t01prod05:/backupdisk          /mnt/team01_backups   nfs4   auto   0     0
```

sudo reboot

5. Docker (Swarm & Compose) – Portainer & Swarmprom

5.1 Docker (Swarm)

We maken gebruik van Docker containers om de effectieve hosting omgeving in op te zetten voor de klant, sommige van deze containers werken in Docker Swarm, aldus betekend dit dat we ook Docker Swarm zullen installeren. Daarnaast maken we ook gebruik van Docker Compose voor een lamp-stack.

De standaard Docker installatie gebruiken we onder andere voor lamp-stack containers als hosting omgevingen

We beginnen met de normale Docker installatie op alle productie VMs die wij op de officiële Docker website gevolgd hebben:

<https://docs.docker.com/engine/install/debian/>

Vervolgens gaan we verder met Docker Swarm installeren op productie VM 5, we gebruiken dit onder andere voor een Wordpress template die we ophalen vanuit Portainer. Portainer zal tevens ook onze Docker Swarm managen, en draait zelf ook in een Docker container op productie VM 5.

1. <https://docs.docker.com/engine/swarm/swarm-tutorial/>
2. <https://docs.docker.com/engine/swarm/swarm-tutorial/create-swarm/>
3. Productie VMs 1 t.e.m. 4 zullen bij de Docker Swarm komen als worker nodes: <https://docs.docker.com/engine/swarm/swarm-tutorial/add-nodes/>

5.2 Portainer

Nu kunnen we beginnen met de Portainer installatie

<https://www.portainer.io/installation/>

En dan voeren we uit op Productie VM 5 wat we in de volgende sectie vinden (negeer de rest).

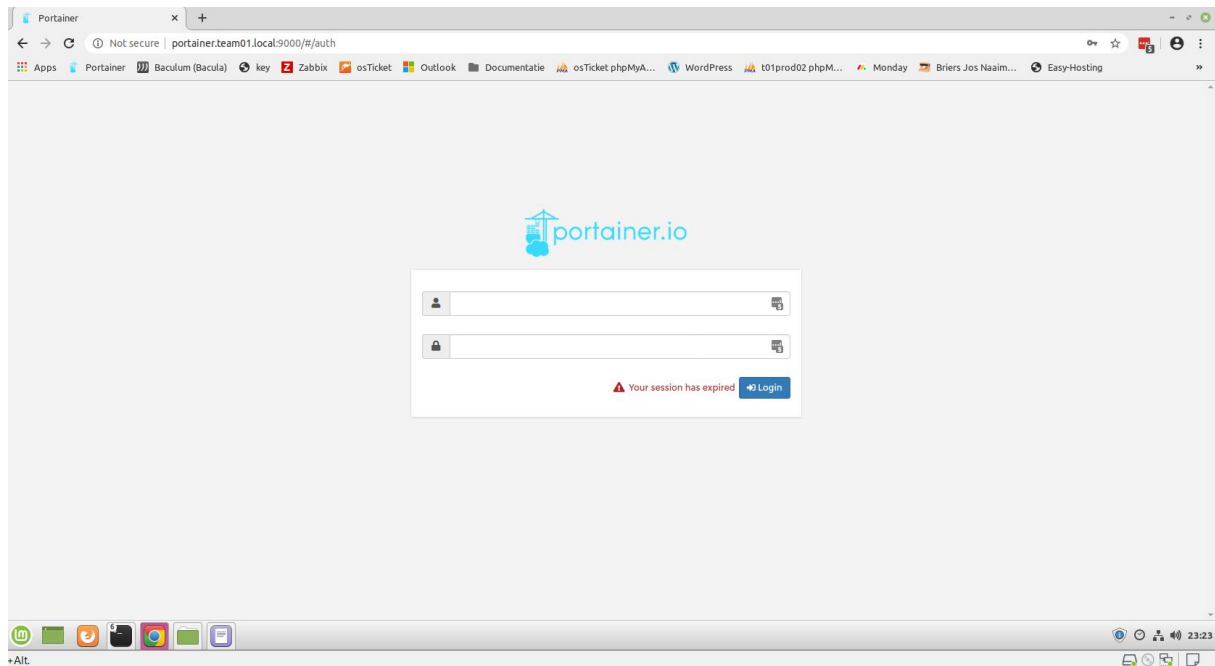
Manage a **LINUX Swarm cluster** with Portainer Server and the Portainer Agent

Deploying Portainer and the Portainer Agent to manage a Swarm cluster is easy ! You can directly deploy Portainer as a service in your Docker cluster. Note that this method will automatically deploy a single instance of the Portainer Server, and deploy the Portainer Agent as a global service on every node in your cluster.

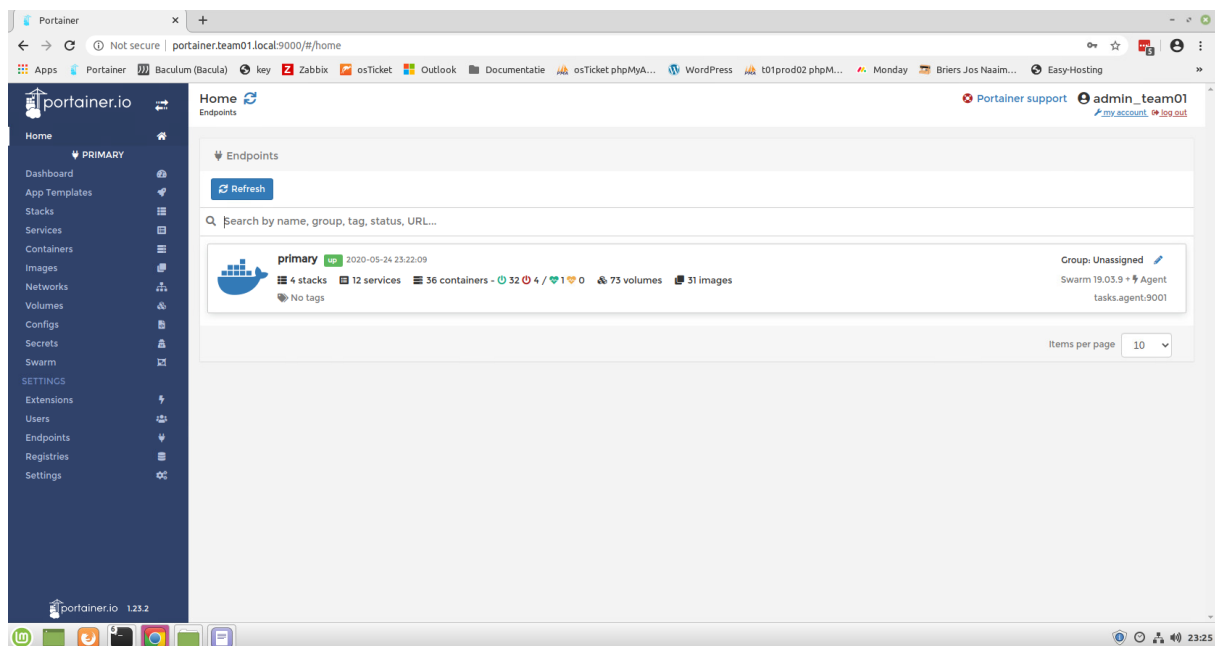
```
$ curl -L https://downloads.portainer.io/portainer-agent-stack.yml -o portainer-agent-stack.yml
$ docker stack deploy --compose-file=portainer-agent-stack.yml portainer
```

You'll just need to access the port 9000 of the Docker engine where portainer is running using your browser.

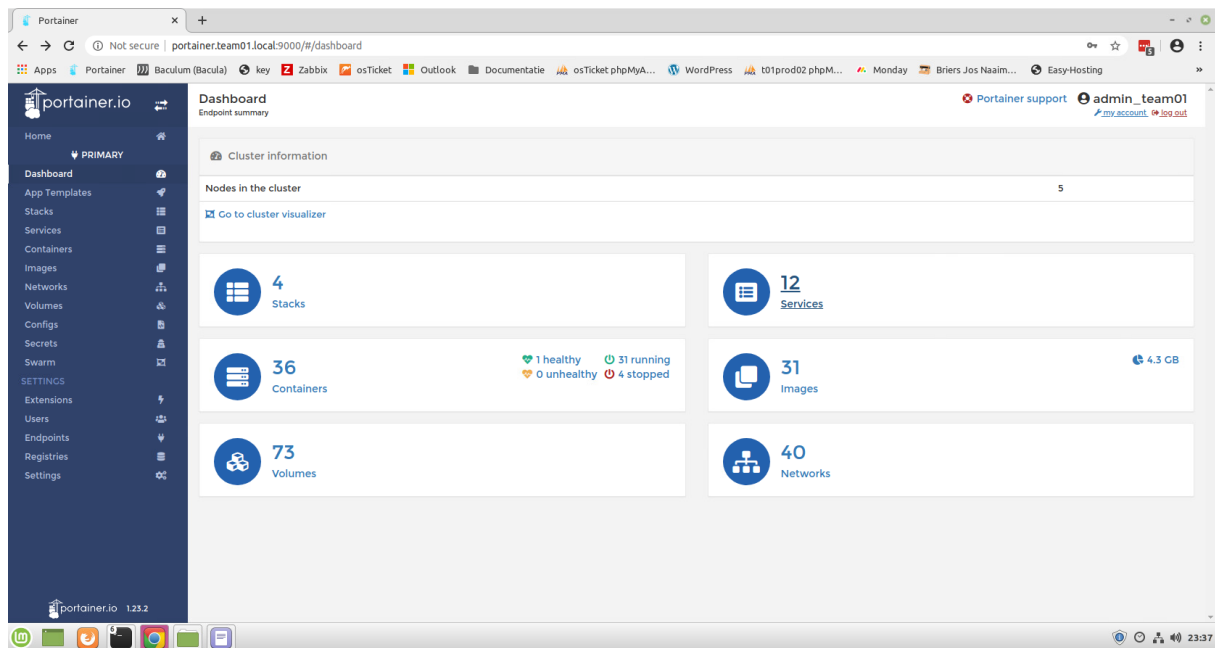
En net zoals er staat kunnen we dan naar de website gaan op de management VM



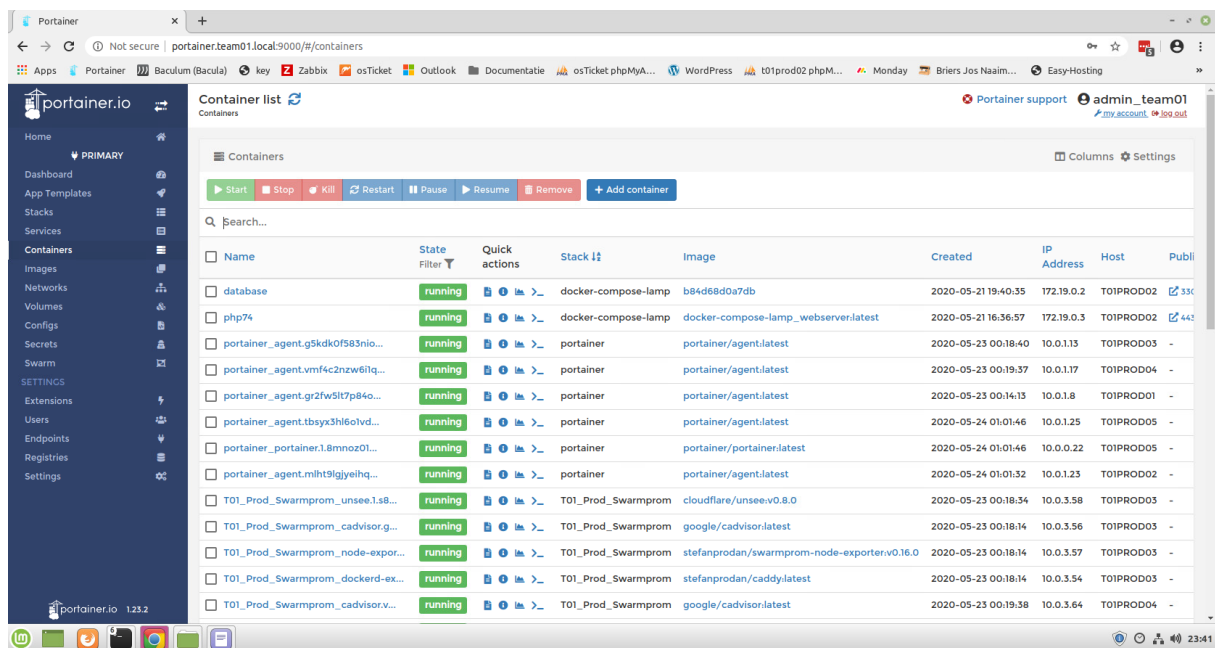
Als we inloggen (na het aanmaken van onze user) komen we hier terecht



Er is te veel informatie om geheel Portainer te overlopen maar wat wij hier in essentie al zeker kunnen doen is een mooi overzicht krijgen van onze Docker containers en de status ervan. Om dit te doen, kunnen we klikken op de cluster in het midden van het scherm met de naam "primary".



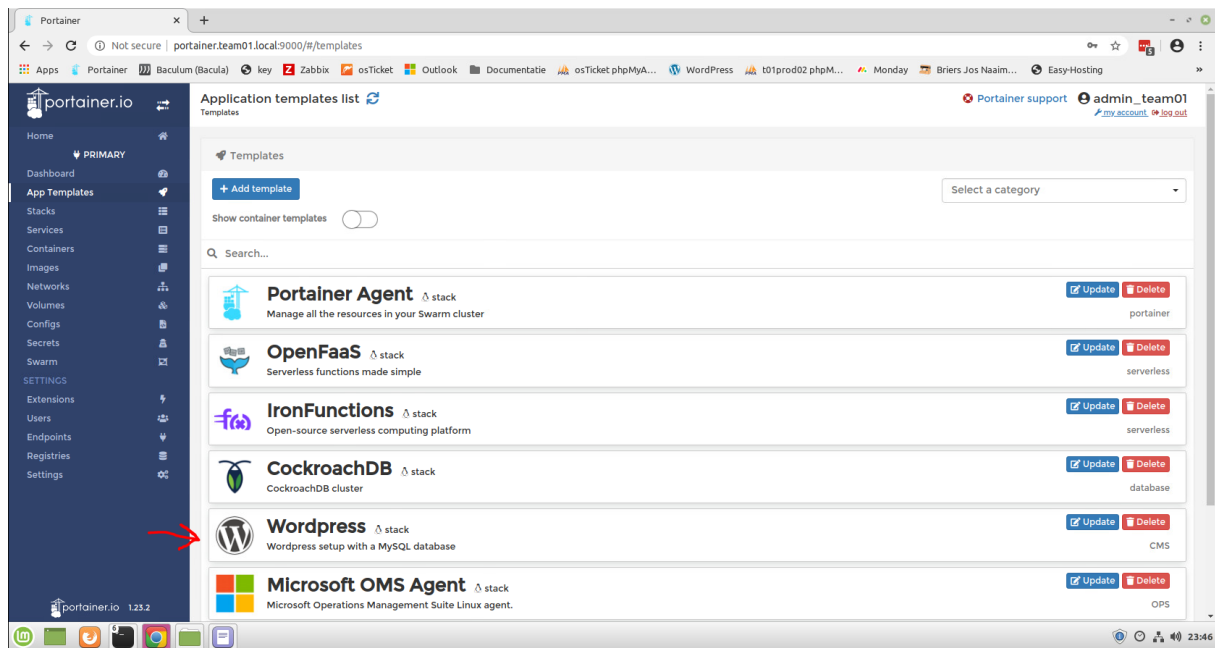
Vanaf hier kunnen we naar “containers” gaan om dus onze Docker containers te monitoren en te beheren.



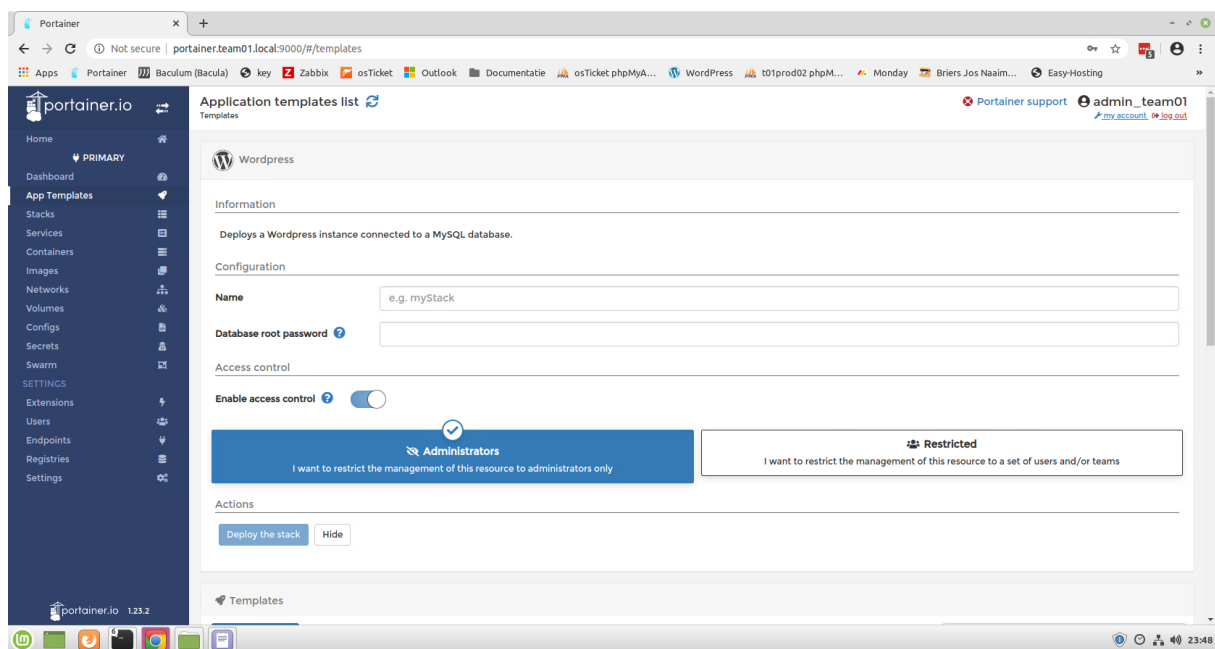
Zoals we kunnen zien kunnen we hier de status zien van containers, acties uitvoeren zoals starten, stoppen, etc.



Als we naar app templates gaan in het dashboard aan de linkerkant kunnen we het Wordpress template implementeren voor gebruik.

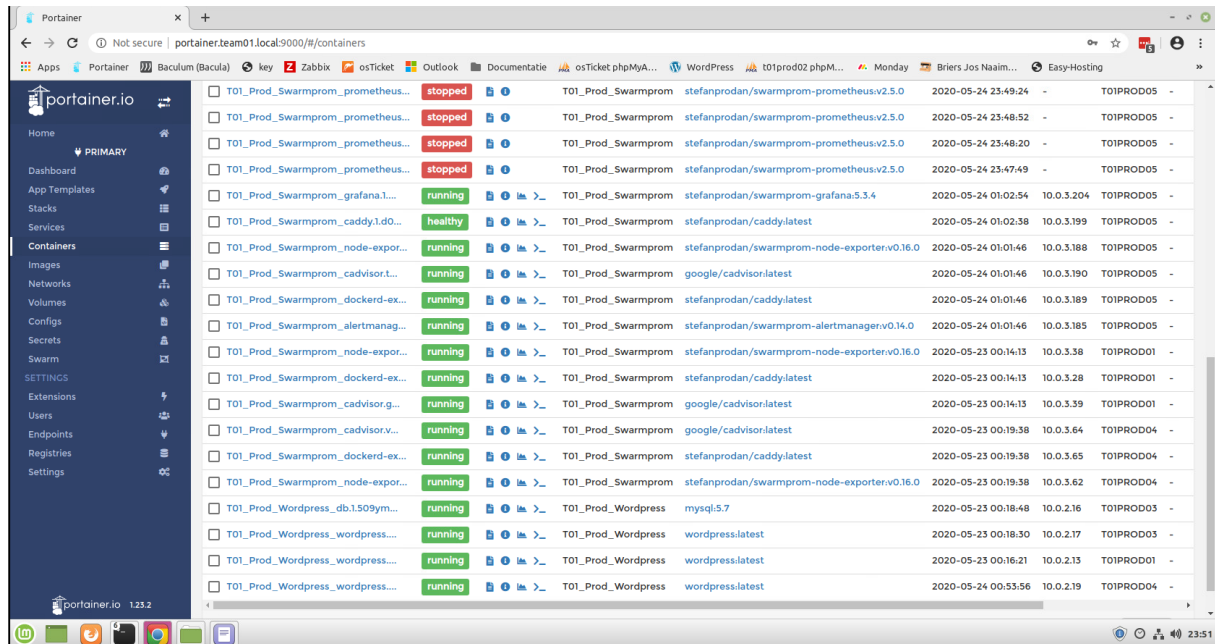


Dan moeten we deze nog een naam geven en vervolgens deployen



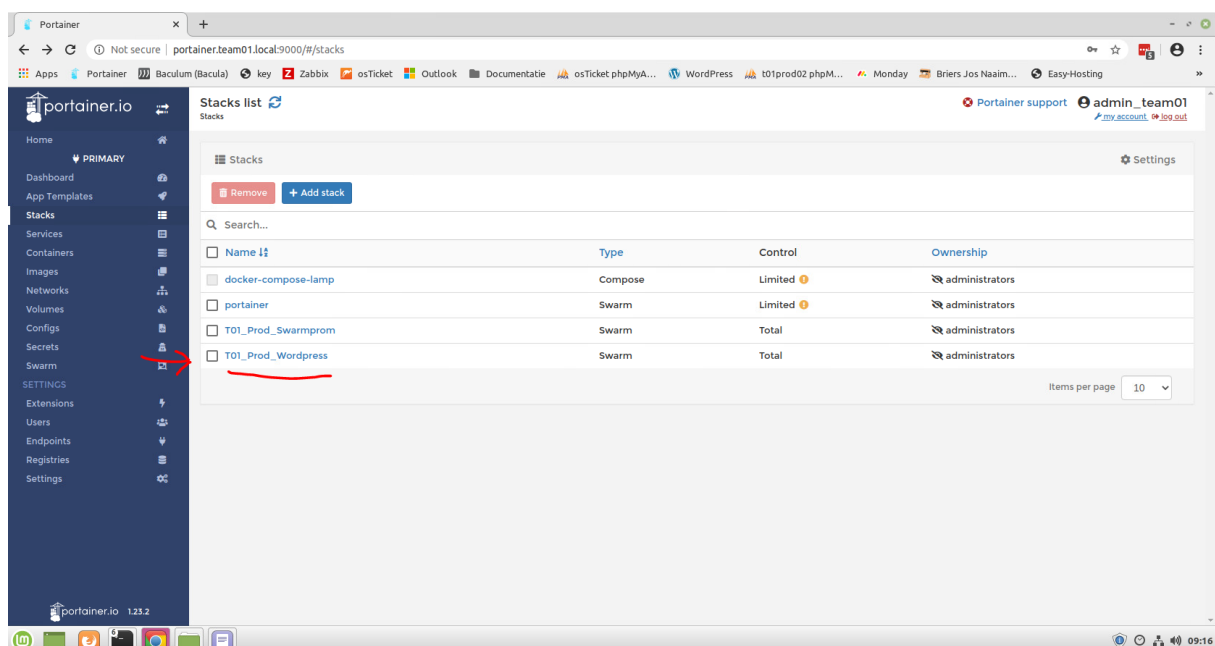
Als we dan terug gaan naar de containers tab dan kunnen we hier (vanonder) de status van de Wordpress containers terugvinden, dit zal zo zijn met al onze

containers



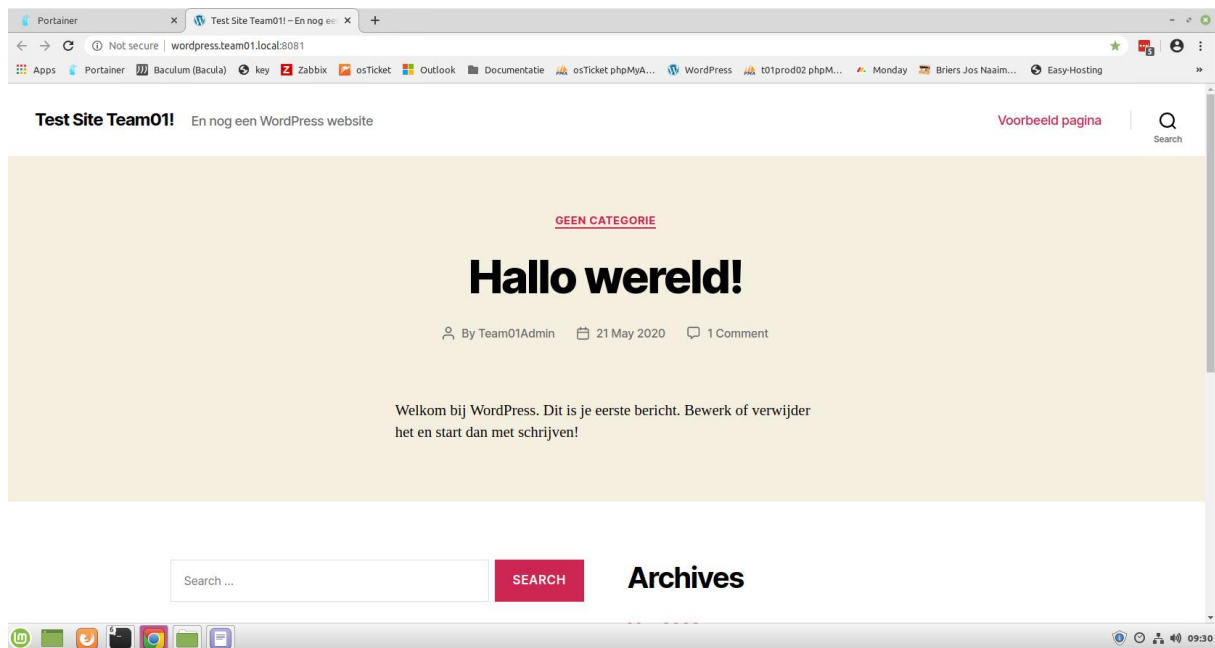
Container Name	Status	Image	Created	Labels
T01_Prod_Swarmprom_prometheus...	stopped	stefanprodan/swarprom-prometheus:v2.5.0	2020-05-24 23:49:24	TO1PROD05
T01_Prod_Swarmprom_prometheus...	stopped	stefanprodan/swarprom-prometheus:v2.5.0	2020-05-24 23:48:52	TO1PROD05
T01_Prod_Swarmprom_prometheus...	stopped	stefanprodan/swarprom-prometheus:v2.5.0	2020-05-24 23:48:20	TO1PROD05
T01_Prod_Swarmprom_prometheus...	stopped	stefanprodan/swarprom-prometheus:v2.5.0	2020-05-24 23:47:49	TO1PROD05
T01_Prod_Swarmprom_grafana.1...	running	stefanprodan/swarprom-grafana:5.3.4	2020-05-24 01:02:54	TO1PROD05
T01_Prod_Swarmprom_caddy.1.d0...	healthy	stefanprodan/caddy:latest	2020-05-24 01:02:38	TO1PROD05
T01_Prod_Swarmprom_node-expor...	running	stefanprodan/swarprom-node-exporter:v0.16.0	2020-05-24 01:01:46	TO1PROD05
T01_Prod_Swarmprom_cadvisor.t...	running	google/cadvisor:latest	2020-05-24 01:01:46	TO1PROD05
T01_Prod_Swarmprom_dockerd-ex...	running	stefanprodan/caddy:latest	2020-05-24 01:01:46	TO1PROD05
T01_Prod_Swarmprom_alertmanag...	running	stefanprodan/swarprom-alertmanager:v0.14.0	2020-05-24 01:01:46	TO1PROD05
T01_Prod_Swarmprom_node-expor...	running	stefanprodan/swarprom-node-exporter:v0.16.0	2020-05-23 00:14:13	TO1PROD01
T01_Prod_Swarmprom_dockerd-ex...	running	stefanprodan/caddy:latest	2020-05-23 00:14:13	TO1PROD01
T01_Prod_Swarmprom_cadvisor.g...	running	google/cadvisor:latest	2020-05-23 00:14:13	TO1PROD01
T01_Prod_Swarmprom_cadvisor.v...	running	google/cadvisor:latest	2020-05-23 00:19:38	TO1PROD04
T01_Prod_Swarmprom_dockerd-ex...	running	stefanprodan/caddy:latest	2020-05-23 00:19:38	TO1PROD04
T01_Prod_Swarmprom_node-expor...	running	stefanprodan/swarprom-node-exporter:v0.16.0	2020-05-23 00:18:48	TO1PROD04
T01_Prod_Wordpress_db.1.509ym...	running	mysql:5.7	2020-05-23 00:18:30	TO1PROD03
T01_Prod_Wordpress_wordpress...	running	wordpress:latest	2020-05-23 00:18:30	TO1PROD03
T01_Prod_Wordpress_wordpress...	running	wordpress:latest	2020-05-23 00:16:21	TO1PROD01
T01_Prod_Wordpress_wordpress...	running	wordpress:latest	2020-05-24 00:53:56	TO1PROD04

Nu is het ook van belang om te weten op welke poort Wordpress draait, anders kunnen we niet naar de website. Daarvoor moeten we naar de "Stacks" gaan in het dashboard aan de linkerkant. Vervolgens klikken we op de Wordpress stack.



Name	Type	Control	Ownership
docker-compose-lamp	Compose	Limited	administrators
portainer	Swarm	Limited	administrators
T01_Prod_Swarmprom	Swarm	Total	administrators
<u>T01_Prod_Wordpress</u>	Swarm	Total	administrators

Onder "Published Ports" kunnen we zien dat Wordpress draait op poort 8081. Als we dit opzoeken komen we op onze testwebsite van Wordpress terecht.



5.3 Docker Compose

We hebben ook een Docker Compose lamp-stack die we gebruiken als hosting omgeving.

Daarvoor hebben we de volgende tutorials geraadpleegd:

<https://docs.docker.com/compose/install/>

Zonder Docker Compose te installeren kunnen we uiteraard niet beginnen.

<https://github.com/sprintcube/docker-compose-lamp>

Zoals gewoonlijk met Github staat in de "readme" beschreven hoe we deze repository implementeren.

Vervolgens kunnen we via portainer aan deze container werken

Container list

Containers

Portainer support admin_team01

Columns Settings

Start Stop Kill Restart Pause Resume Remove Add container

Search...

Name	State	Quick actions	Stack	Image	Created	IP Address	Host	Public
database	running		docker-compose-lamp	b84d68d0a7db	2020-05-21 19:40:35	172.19.0.2	T01PROD02	330
php74	running		docker-compose-lamp	docker-compose-lamp_webserver:latest	2020-05-21 16:36:57	172.19.0.3	T01PROD02	443
portainer_agent.gr2fw5lt7p84o...	running		portainer	portainer/agent:latest	2020-05-23 00:14:13	10.0.1.8	T01PROD01	-
portainer_agent.g5kdK0f583nio...	running		portainer	portainer/agent:latest	2020-05-23 00:18:40	10.0.1.13	T01PROD03	-
portainer_agent.vmf4c2nzW6ilq...	running		portainer	portainer/agent:latest	2020-05-23 00:19:37	10.0.1.17	T01PROD04	-
portainer_agent.mhlt9lgjyehq...	running		portainer	portainer/agent:latest	2020-05-24 01:01:32	10.0.1.23	T01PROD02	-
portainer_agent.tbsyx3hl6o1vd...	running		portainer	portainer/agent:latest	2020-05-24 01:01:46	10.0.1.25	T01PROD05	-
portainer_portainer.18mnoz01...	running		portainer	portainer/portainer:latest	2020-05-24 01:01:46	10.0.0.22	T01PROD05	-
T01_Prod_Swarmprom_node-expor...	running		T01_Prod_Swarmprom	stefanprodan/swarmprom-node-exporter:v0.16.0	2020-05-23 00:14:13	10.0.3.38	T01PROD01	-
T01_Prod_Swarmprom_dockerd-ex...	running		T01_Prod_Swarmprom	stefanprodan/caddy:latest	2020-05-23 00:14:13	10.0.3.28	T01PROD01	-
T01_Prod_Swarmprom_cadvisor.g...	running		T01_Prod_Swarmprom	google/cadvisor:latest	2020-05-23 00:14:13	10.0.3.39	T01PROD01	-
T01_Prod_Swarmprom_unsee.1s8...	running		T01_Prod_Swarmprom	cloudflare/unsee:v0.8.0	2020-05-23 00:18:34	10.0.3.58	T01PROD03	-
T01_Prod_Swarmprom_cadvisor.g...	running		T01_Prod_Swarmprom	google/cadvisor:latest	2020-05-23 00:18:14	10.0.3.56	T01PROD03	-

En met de console optie kunnen we aan de slag

Container details

Containers > php74

Portainer support admin_team01

Actions

Start Stop Kill Restart Pause Resume Remove Recreate Duplicate/Edit

Container status

ID 5bc9a7a1db6ea4f0281372add9a4203077e3b8a970815e73b38741b28aa3b2f

Name php74

Status Running for a day

Created 2020-05-21 16:36:57

Start time 2020-05-24 00:55:21

Logs Inspect Stats Console Attach

Access control

Ownership administrators

Change ownership

Create image

In de volgende screenshot kan u zien dat we (via NFS share) een opslagplaats hebben voor de websites.

Container console

Containers > php74 > Console

Portainer support admin_team01

Execute

Exec into container as default user using command bash Disconnect

```

root@2ebcd0333d4:/var/www/html# cd ..
root@2ebcd0333d4:/var/www# ls
html webserverroot
root@2ebcd0333d4:/var/www# cd webserverroot
root@2ebcd0333d4:/var/www/webserverroot# ls
testfile.txt website_easyhosting website_naaimachines
root@2ebcd0333d4:/var/www/webserverroot#

```

Onze Apache2.conf file hebben wij aangepast met behulp van deze tutorial:

<https://httpd.apache.org/docs/2.4/vhosts/examples.html>

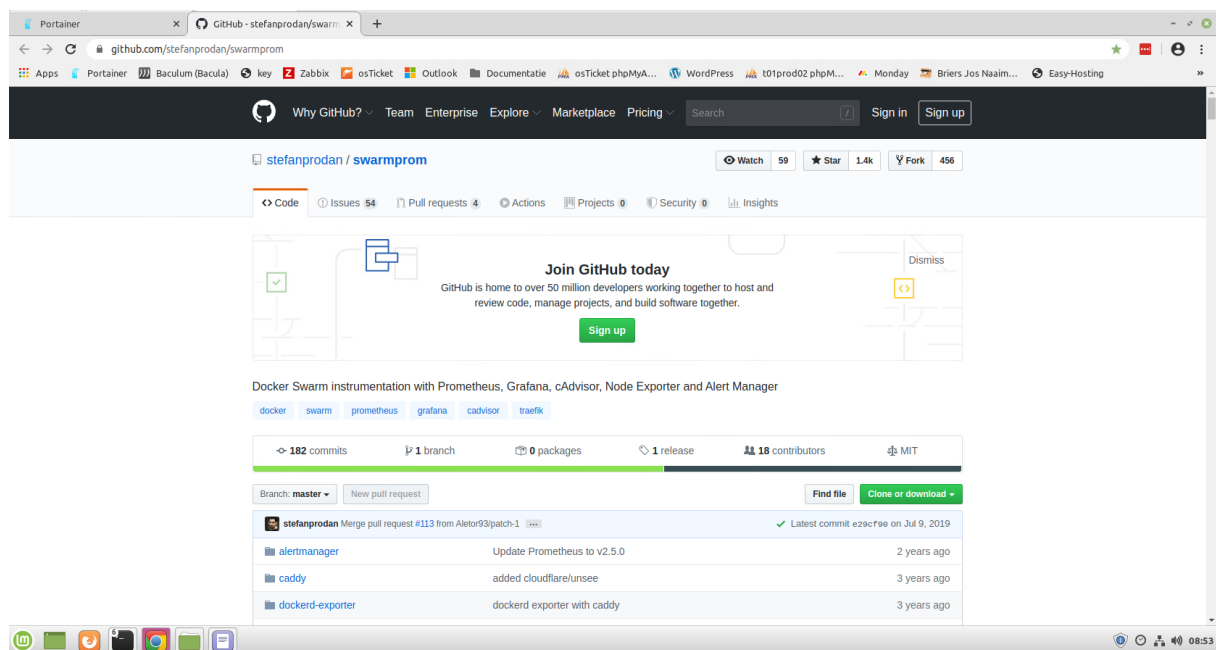
5.4 Swarmprom

Portainer is niet het enige dat we gebruiken om onze Docker omgeving te managen en te monitoren, we maken ook gebruik van "swarmprom" zoals we terug kunnen vinden op Github.

<https://github.com/stefanprodan/swarprom>

een overview van wat swarmprom allemaal doet en de installatie kan je hier terugvinden:

<https://dockerswarm.rocks/swarprom/>



Portainer

GitHub - stefanprodan/swarprom

Why GitHub? Team Enterprise Explore Marketplace Pricing Search Sign in Sign up

stefanprodan / swarprom

Watch 59 Star 1.4k Fork 456

Code Issues 54 Pull requests 4 Actions Projects 0 Security 0 Insights

Join GitHub today

GitHub is home to over 50 million developers working together to host and review code, manage projects, and build software together.

Sign up

Docker Swarm instrumentation with Prometheus, Grafana, cAdvisor, Node Exporter and Alert Manager

docker swarm prometheus grafana cAdvisor traefik

182 commits 1 branch 0 packages 1 release 18 contributors MIT

Branch: master New pull request Find file Clone or download

stefanprodan Merge pull request #113 from Alelor93/patch-1 Latest commit e2ac7ee on Jul 9, 2019

alelor93	Update Prometheus to v2.5.0	2 years ago
alelor93	added cloudflare/unsee	3 years ago
alelor93	dockerd exporter with caddy	3 years ago

<https://www.cloudreach.com/en/resources/blog/ct-apache-docker-containers/>

<https://forums.docker.com/t/shared-web-hosting-with-docker-best-practices/7893>

<https://www.centlinux.com/2020/03/configure-lamp-stack-in-docker-containers.html>

https://linuxhint.com/lamp_server_docker/