Scriptie ingediend tot het behalen van de graad van
PROFESSIONELE BACHELOR IN DE ELEKTRONICA-ICT

IT Automation and Monitoring

Roman Tymofiyev

academiejaar 2015-2016

AP Hogeschool Antwerpen
Wetenschap & Techniek
Elektronica-ICT

ARTESIS PLANTIJN
HOGESCHOOL ANTWERPEN

Table of Contents

Dankwoord

Graag wil ik in dit dankwoord alle docenten van Elektronica-ICT richting bedankeen voor geduld en geloof in ons, dit motiveert ons enorm. Apart wil ik meneer Maset bedanken voor studie toelating.

Ik wil stagebedrijf, Inventive Designers, bedanken voor de stageplaats en gezonde werkomgeving. Bijzondere dank aan mijn stagebegeleider Nico Brys voor de goede ondersteuning en de toegewezen tijd.

Zeker wil ik mijn promotors Patrick van Houtven en Tim Dams bedanken voor bereidheid om te helpen.

Abstract

Voor interne monitoring infrastructuur maakte Inventive Designers gebruik van Nagion monitoring systeem, waarmee ze over meer dan drie honderd servers, services en websites monitorde.

De bedoeling van mijn stage was om Nagios te vervangen door Icinga2 monitoring systeem met behulp van Puppet management tool. Puppet moest er voor zorgen dat installatie van Icinga2 en beheer van icinga2 infrastructuur op geautomatiseerde manier verliep.

Situering

Inventive Designers is een software ontwikkeling bedrijf dat richt zich op levering en ondersteuning van software, dat zorgt ervoor dat bedrijven beter communiceren met hun klanten via multichannel klantencommunicatie. Inventive Designers is bekend door Scripture Engage software.

De stage project was niet gerelateerd met de producten van het stagebedrijf, en was enkel bedoeld voor interne gebruik. Tijdens de stage is er gewerkt aan een interne project, gericht op systeem beheer en vernieuwing van een monitoring systeem.

Om over drie honderd servers, services en websites te kunnen monitoren gebruikte Inventive Designer Nagios monitiring systeem. Nagios moest op een geautomatiseerde manier vervangen worden door een nieuwe Icinga2 monitoring systeem.

Bespreking

Probleem en opdracht omschrijving:

Inventive Designers heeft beslissing genomen om externe monitoring systeem te vervangen door Icinga2 systeem. Deze opdracht heeft een externe bedrijf uitgevoerd.

Het project ontstond toen interne monitoring systeem ook vernieuwd moest worden.

Nagios, oude monitoring systeem die door Inventive Designers was gebruikt, is een open source systeem die monitoring en waarschuwing service biedt voor servers, services, applicaties en netwerk. Indien een fout melding optreedt of een host down gaat, wordt er een notificatie gestuurd naar browser, email of gsm afhankelijk van de instellingen.

Installatie van nieuwe monitoring systeem moest op een geautomatiseerde manier verlopen, dat wil zegen dat Icinga2 door middel van een management tool geïnstalleerd moest worden. Om dit te kunnen realiseren moest er een onderzoek gedaan worden naar verschillende automatisatie management tools. De keuze stond vrij.

Basis vereisten

Hieronder een opdracht samenvatting in grote lijnen:

- Onderzoek naar verschillende monitoring systemen (Nagios, Icinga2).
- Onderzoek naar verschillende automatisatie systemen (Puppet, Chef, Ansible).
- Nagios infrastructuur vervangen door Icinga2.
- Een automatisatie systemen opstelen en linken met Icinga2.
- Icinga clients moeten geïnstalleerd worden meet een automatisatie systemen.
- Linux hosts moeten toegevoegd worden aan Icinga2 met basis monitoring door middel van een script.
- Verschillende monitoring notificaties prioriteiten instellen.

Keuze technologieën

Omschrijving van technologieën die tijdens project aan bood kwamen. De keuze van technologieën waren soms van zelfsprekend of door bedrijf zelf gekozen. Keuze van automatisatie systeem moest eerst onderzocht worden.

Gebruikte technologieën en korte beschrijving er van:

- Ubuntu
- VMWare
- Apache2 server
- MySQL
- PHP
- Puppet
- PuppetDB
- Nagios
- Icinga2
- Icingaweb2

Ubuntu

Linuxdestributie, een besturing systeem die tijdens project gebruikt was om applicaties zo als apache2, php, mysql-server, icinga2 en puppet te kunnen runnen. Versie van Ubuntu server was 14.04.4, een stabiele versie van Ubunte die door stage bedrijf was gekozen.

VMWare

VMWare is een virtualisatie tool die door Inventive Designers wordt gebruikt voor aanmaak en beheer van virtuele infrastructuur. In een van de clusters werd er een aantaal Ubuntu servers aangemaakt voor verdere exploitatie.

Apache2 server

Apache2 is een HTTP server die noodzakelijk was om icingaweb2 applicatie te kunnen runnen.

MySQL

MySQL is een systeem voor beheer van databases. Voor installatie van Icinga2 en Icingaweb2 zijn er databases noodzakelijk. Als alternatieve, kon er ook een PostgreSQL database gebruikt worden.

PHP

PHP is een programeer taal voor web applicaties. Om Icingaweb2 applicatie te kunnen runnen was een PHP library noodzakelijk voor connectie en manipulatie van MySQL databases. Met behulp van PHP was er mogelijk om vanuit MySQL database dynamisch content overbrengen naar web server.

Puppet

Puppet is een open source configuratie management tool die gebruikt werd voor installatie en beheer van IT infrastructuur. Met behulp van Puppet was het mogelijk om servers en services te installeren zo als apache en Icinga2 op geautomatiseerde manier.

Puppet is agent/master applicatie, dat wil zeggen dat centrale beheer vanuit een Puppet master uitgevoerd wordt.

Omdat er al binnen de bedrijf met Puppet gewerkt was, logische beslissing was om ook gebruik van puppet te maken. Gebruikte versie 3.8.7.

PuppetDB

PuppetDB is een dataopslag plaats voor informatie die door de puppet wordt gegenereerd. PuppetDB was noodzakelijk om data op te vangen van externe puppet nodes. PuppetDB was een belangrijke onderdeel die gebruikt was om externe host te kunnen monitoren met behulp van icinga2.

Nagios

Nagios is een open source software applicatie met monitoring en waarschuwing mogelijkheden. Voert monitoring van services, servers en netwerk apparatuur uit. Nagios heeft een web based applicatie om gemonitorde componenten in kart te brengen. Nagios werd gebruikt door Inventive Designers voor monitoring van IT infrastructuur.

Icinga2

Icinga2 is een open source computer en netwerk monitoring systeem.

Icingaweb2

Icingaweb2 nieuwe moderne web gebased applicatie voor Icinga2. Icingaweb2 applicatie brengt gemonitorde services, servers netwerk componenten en sensoren in kaart, alle gemonitorde statussen worden in een web browser veergegeven.

Puppet from zero to hero

Puppet is een cross platform, open source management tool die zowel voor configuratie van een IT infrastructuur met een grote aantaal Linux en Windows hosts gebruikt kan worden, bijvoorbeeld installatie van een Apache server of MySQL server op tiental Ubuntu servers tegelijkertijd, als ook beheer van bestaande softwarepakketten, bijvoorbeeld verwijderen van PHP.

Puppet heeft een agent based architectuur (master/agent). Voor de project was de master/agent model ook het geval.

Project opstelling

- Een Puppetmaster Ubuntu server. Met puppetmaster applicatie.
  - Een development Puppet agent Ubuntu server - een test server met puppet agent applicatie, waar testinstallaties tijdens de project werden uitgevoerd.
  - Een production Puppet agent Ubunte server - een eindproduct server met puppet agent applicatie, die in theorie in productie gebruikt kon worden.

Deze drie servers werden in interne virtuele omgeving aangemaakt, met behulp van VMWaer virtualisatie software.

Installatie van Puppet master

Puppet master is een applicatie die op server wordt uitgevoerd voor centrale beheer van puppet nodes (puppet agenten).

Om een puppet master te instaleren zijn er verschillende wegen er naar toe. Specifiek voor deze project werden volgende stappen ondernomen.

- Eerst een vooral moest er een Puppet package gekozen en in het vervolg gedownload worden, de keuze van package is afhankelijk van de besturingssysteem. Tijdens de project was er gewerkt met Ubuntu 14.04.4 besturingssysteem, daarom was er voor gekozen om volgende package te gebruiken *"puppetlabs-release-trusty"*. Om gevenste packege te downloaden werd `wget` commando gebruikt.
- Dan werd de package uitgepakt met `dpkg` commando, gevolgd door de gedownload package naam.
- Om de nieuwe packages binnen te halen werd er een update commando uitgevoerd.

```
$ wget https://apt.puppetlabs.com/puppetlabs-release-trusty.deb

$ sudo dpkg -i puppetlabs-release-trusty.deb

$ sudo apt-get update

$ sudo apt-get install puppemaster
```

- In vervolg werd er Puppet master package geïnstalleerd.

- Om puppet master te starten wordt `service puppetmaster start` commando gebruikt.
- Om installatie van puppetmaster te verifiëren, kunnen volgende commando gebruikt worden.
  - `puppet -V`, om versie van geïnstallerde puppetmaster te controleren.
  - en `service puppetmaster status`, om status van puppetmaster te controleren.

```
$ puppet -V
$ service puppetmaster status
```

Na installatie van puppetmaster werd er aan installatie van puppet agent nodes begonnen. Twee Ubuntu servers, een server voor de testing en een server voor theoretische productie. Beide servers met Ubuntu 14.04.4 versie.

Installatie van Puppet agent

Puppet agent is een applicatie die op de gewenste node word uitgevoerd voor toekomstige beheer. Net zoals bij de puppet master installatie, moest er een puppet packge gedownload en uitgepack worden.

- Package werd binnen gehaald met `wget` commando, gevolgd door de package url (afhankelijk van de besturingssysteem).
- Dan werd packge uitgepakt met `dpkg` commando, gevolgd door de gedownload package naam.
- En zoals bij puppetmaster werd er update commando uitgevoerd om de nieuwe packages binnen te halen.

```
$ wget https://apt.puppetlabs.com/puppetlabs-release-trusty.deb

$ sudo dpkg -i puppetlabs-release-trusty.deb

$ sudo apt-get update

$ sudo apt-get install puppet
```

- In vervolg werd er Puppet package geïnstalleerd.
- Om puppet agent te starten wordt `service puppet start` commando gebruikt.
- Om installatie van puppet agent te verifiëren, kunnen volgende commando gebruikt worden.
  - `puppet -V`, om versie van geïnstallerde puppet te controleren.
  - en `service puppet status`, om status van puppet te controleren.

```
$ puppet -V
$ service puppet status
```

Om te kunnen connecteren met een puppet master, moest er een config file `nano /etc/puppet/puppet.conf` aangepast worden. Zodat puppet agent weet wie puppet master juist is en met wie agent connectie moet maken. In config file moest de master sectie in commentaar gezet worden en een nieuwe agent sectie toegevoegd worden. Nu weet de puppet agent vie zijn master is.

```
# [Master]

[agent]
server = puppetmaster (naam van puppet master server)
```

Standard probeert een puppet agent connectie te makken met puppet (by default). Daarom kan naam "puppet" gelinkt worden aan ip address van puppetmaster server in DNS server.

SSL connectie

Na dat de installatie van Puppet master en Puppet agent uitgevoerd was. Werd er een connectie geleegd tussen twee machines. Om dit te kunne realiseren maakt Puppet gebruik van SSL connectie.

SSL Secure sockets layer: is een encryptie-protocol die de communicatie tussen twee computers beveiligt. SSL gebruikt certificaten om de uitgewisselde gegevens te authenticeren en privacy te garanderen. wiki

Na installatie en opstarten van puppet agent service, genereert agent van zelf een certificaat die naar puppetmaster verstuurd wordt. Om te zien of een certificaat effectief gegenereerd was wordt er volgende commando uitgevoerd `puppet agent --fingerprint` op de server waar puppet agent geïnstallerd was.

Puppet master moet certificaat die door een puppet agent gegenereerd was nog ondertekenen.

Om te zien welke certificaten al onderteken zijn en welke niet, wordt er op puppet master volgende commando uitgevoerd `puppet cert list --all` . Er wordt een lijst van certificaten weergegeven die door puppet master gekkend zijn.

Om certificaat te ondertekenen werd er volgende commando op puppet master uitgevoerd `puppet cert sign + Naam Van puppet agend node` of als er meerdere certificaten ondertekend moeten worden `puppet cert sign --all` commando wordt gebruikt.

```
$ puppet cert sign puppetagent
of
$ puppet cert sig, --all
```

Certificaten die op de puppet master worden gedetecteerd kunnen automatisch ondertekend worden, door middel van etc/puppet/puppet.conf file aan te passen. In master sectie wordt er volgend lijn aangepast, door true te zetten `autosign = true`

```
[master]
autosign = true
```

Autosign kan nuttig zijn als installatie van de puppet agenten ook geautomatiseerd moet worden. Door middel van een bash script en ssh connectie, kan een installatie van de puppet automatisch uitgevoerd worden op gewenste host.

De belangrijkste commando's om dit scenario werkend te krijgen zijn:

`scp` voor secure copy van .sh file, om gewenste script file naar bestemming over te brengen.

`ssh -t` om .sh file met script op afstand uit te voeren.

Voor de project werd autosign en bash scripts niet gebruikt, omdaat er niet zo weel servers geconfigurerd moesten worden met puppet applicatie.

Installatie van de puppet master en puppet agent servers werd succesvol uitgevoerd en verbinding tussen de puppet machines door middel van ssl certificaten gegarandeerd. Meet een puppet commando op de agent node werd de connectie getest `puppet agent -t` .

Puppet werking

Werking van de puppet in master/agent opstelling.

Om puppet agent te kunnen configureren heeft puppet master twee belangrijke items nodig.

- Manifest file.
- Catalogs.

Na installatie van puppet master werden er default mappen door de puppet aangemaakt.

Een van de belangrijkste mappen is environment map `/etc/puppet/environment/` . In de environment map worden twee mappen aangemaakt. /production map en /development map.

Zo als al logisch afgeleid kon worden zijn ze gebruikt een voor eindproduct (/production) en een voor testing en ontwikkeling (/development).

In elke map werd dan en /manifests map aangemaakt om daarin manifest document(en) in te zetten. Elke manifest map bevat een of meerdere manifest files met .pp extensie.

Voor de project werd er alleen met /production map gewerkt.

Project environment map zat als volgende uit `/etc/puppet/environment/production/manifests/node.pp` .

Een manifest document is een plaats waar node definities werden gedeclareerd. Een node definitie is simpelweg declaraties van pupept agenten met code er in die op pupept agent uitgevoerd moet worden.

Gedeclareerde node naam moet overeenkomen met agent naam.

Een manifest file zit als volgende uit.

```
node 'puppetagent1' {
include '::ntp'
class { "apache": }

}

node 'puppetagent2' {
class { '::ntp':
  servers => [ 'ntp1.corp.com', 'ntp2.corp.com' ],
}

class { "apache": }

}
```

Cataloog

Zo een node definitie met code er in wordt een cataloog met statmens genoemd.

Puppet agent server voert een agent applicatie als een service op achtergrond uit, elke 300 seconden (default) word dan een rapport naar puppet master gestuurd. Een agent rapport wordt facts genoemd.

Facts zijn gegevens over een puppet agent server met huidige software en hardware specificaties en installaties en services die al reeds opgezet zijn.

Puppet master ontvangt facts over de agent en op basis er van wordt dan een cataloog samengesteld. Facts zorgen er voor dat een cataloog maar een keer uitgevoerd wordt zo lang er geen veranderingen zijn in agent facts of in een cataloog die tot agent behoort.

Een cataloog met statement die tot een bepaalde agent behoren, kan verschillende klassen bevatten. Bijvoorbeeld een linux class die verschillende packages op puppet agent installerd.

Door middel van deze 'linux' class buiten de node declaratie te zetten, kan deze klasse door verschillende nodes gebruikt worden, wat in zijn beurt dubbele code vermindert.

```
node 'puppetagent1' {
class {'linux'}
}
node 'puppetagent2' {
class {'linux'}
}

class linux {
$admintools [nano, git, screen]
package { $admintools:
  insure => 'installed',
}
}
```

Hiera

Hiera is een tool die door de puppet gebruikt kan worden om node declaraties meer gebruikvriendelijker en leesbaar te maken. Door de gebruik te makken van hiera, worden statmens in aparte files beschrijven. Hiera files kunne classes en variables bevatten. Hiera zorgt er voor dat dubbele code vermindert wordt. Daarnaast word er in hiera hiërarchie beschrijven, welke files, onder welke voorwaarden, wel of niet gebruikt mogen worden.

De grootste voordeel is splitsing van publiek en private data, zo worden paswoorden en gevoelige informatie in aparte files verborgen.

Voor de project werd er ook zowel met gewone statement in node declaraties, als met hiera documenten gewerkt.

Hiera gebruikt YAML structuur die voor de gebruiker meer leesbaar is. Ook de JSON structuur gebruikt kan worden.

Om gebruik van hiera te maken werd er een hiera.yaml file aangemaakt in `/etc/puppet/` map.

hiera.yaml file had volgende structuur:

- Backends - beschrijft welke data structuur gebruikt moet werden. (JSON, YAML)
- datadir - map waar .yaml file zich bevinden.
- hierarchy - bijvoorbeeld:
  - er kan beschrijven worden tegen welke node configuratie uitgevoerd moet worden.
  - er kan beschrijven worden welke environment (production of development) gebruikt moet worden.
  - Bovendien worden er .yaml files gedeclareerd die gebruikt moeten worden.
  - common - file waar default configuratie beschrijven wordt. Common file moet er aanwezig zijn maar hoeft geen configuratie te bevatten.

Voor de project werd er volgende hiërarchie gebruikt.

```
:backends:
  - yaml
:yaml:
  :datadir: /etc/puppet/hieradata
:hierarchy:
  - "node/%{::fqdn}"
  - "icinga2/users/users"
  - "icinga2/commands"
  ... enzovoort
  - common
:merge_behavior: deeper
```

Puppet modules

Om een agent node te configureren moesten er verschillende classes geschreven worden zo als apache, mysql en icinga2. Om die taak te vergemakkelijken werd er gebruik van puppet modules gemakt.

Een puppet module is een verzameling van code, classes en manifest files, die door de puppet community geschreven zijn om uitvoeren van installaties te versnellen en vergemakkelijken.

Standaardь ziet er elke module mappen structuur hetzelfde uit. Twee belangrijke mappen zijn.

- manifests map
    - Bevat al manifests files met .pp extensie.
    - Twee belangrijkste files in manifest map zijn init.pp en params.pp
        - init.pp file bevat mein class met parameters declaratie.
        - params.pp file bevat parameter variabelen met default value.
- templates map, bevat template files die door manifest gebruikt kunnen worden. Bijvoorbeeld in icingaweb2 module template map, bevat apache2 configuratie file die de default file op agent node moet vervangen.

Tijdens project werd er maar een module geschreven, om te zien hoe aanmaken van een module in de praktijk verloopt. Een eenvoudige module die een installatie van php voor mysql database uitvoerde. Afhankelijk van de besturingssysteem van agent node werd er een commando uitgevoerd.

```
class puppetagent {

$phpmysql =$osfamily ? {

 'debian' => 'php5-mysql',
  default => 'php-mysql',

}
package { $phpmysql:

ensure => 'present',
}
}
```

Aangezien aanmaken van eigen modules overbodig was, werd er enkel gewerkt met bestande modules die door puppet community reeds geschreven zijn.

Alternatieven

Er waren een aantaal alternatieven die gebruikt konden worden om automatisatie van IT infrastructuur te voorzien. Bijvoorbeeld Ansible, Chef en CFEngine.

Ansible was een van die tools die gebruikt kon worden als oplossing van bestaande probleem.

Net zo als Puppet, Ansible is een open source, cross platform automatisatie tool die beheer van IT infrastructuur kan bieden.

Aansable, in vergelijking met puppet, is een aget less tool die geen agent installatie nodig heeft op nodes die beheert gaan zijn. Ansible kan ad hoc gebruikt worden.

Installatie van Asible is eenvoudiger en veel sneleer, alleen installatie van package is vereist. Ansible maakt ook gebruik van modules, waar een grote aantaal by default me geïnstalleerd wordt.

Connectie met beheerde node wordt via SSH verzekert, wat voor de oplossing van onze probleem geen voordelen kon bieden. SSH keys moeten uitgewisseld worden met beheerde nodes wat zal op de zelfde scenario uitkomen als puppet installatie.

Ansible is goed voor de beheer van de virtuele nodes in cloud omgeving waar SSH keys al uitgedeeld zijn, bij installatie van cloud omgeving, en waar nodes met elkaar vertrouwd zijn.

Bovendien heeft Ansible, in vergelijking met Puppet, minder uitgebreide module van Icinga2 die gebruikt werd tijdens project.

Puppet besluit.

Puppet was een goede keuze voor geautomatiseerde systeem beheer.

Puppet beschikt over de gedetailleerd documentatie, die door de puppet community geschreven is. Bovendien een grote aantaal open source modules die beschikbaar zijn om Puppet applicatie uit te breiden naar eigen vereisten, voor snelle en effectieve oplossing van bestaande problemen.

Modules, geschreven door de community kunnen gemakkelijk aangepast en aangevuld worden met eigen klassen en variabelen, naar eigen behoeften om ingewikkelde problemen op te lossen.

Voor de andere projecten binnen het bedrijf werd er al met Puppet gewerkt, daarom was dat een logische keuze om Puppet blijven te gebruiken.

Puppet biedt management van de servers en services, daarbij kan puppet data van beheerde nodes, zo als CPU en hard disk status verzamelen en in vervolg deze data in PuppetDB database opslaan. Wat op lange termijn voordelen heeft als project uitgebreid moet worden.

Icinga2

Icinga2 is een open source, cross platfor monitoring systeem die tijdens project gebruikt werd om oude Nagios monitoring systeem te vervangen.

Nagios en Icinga2 hebben veel gelijkenissen, omdat Icinga applicatie originel als "fork" van Nagios ontwikkeld was. Door de jaren heen, werd Icinga core herschreven en is Icinga2 geworden.

In vergelijking met Nagios, ondersteunt Icinga2 PostgreSQL en Oracle databases in aanvulling op MySQL. Icinga2 heeft een grote aantal functies en moderne web-dashboard. Monitorings plugins die origineel voor Nagios geschreven waren zijn compatibel men Icinga2.

Icinga2 bidt volgende mogelijkheden:

- Monitoring van netwerk services zo als SMTP, POP3, HTTP, NNTP, ping.
- Monitoring van host resources zo als CPU load, disk usage.
- Monitoring van server componenten switches en routers, ook temperatuur en vochtigheid sensoren kunnen gemonitord worden.
- Gebruiker kan gemakkelijk eigen service checks ontwikkelen.
- Parallelle service checks.
- Makt onderscheid tussen down en unreachable hosts.
- Stuurt notificatie naar bepaalde user of een groep van users via webapplicatie, email of gsm afhankelijk van instralingen.

Installatie van Icinga2 applicatie moest op geautomatiseerde manier gebeuren, daarvoor werd puppet als oplossing gebruikt.

Om werking van Icinga2 beteer te begrijpen werd icinga2 eerst handmatig op test server geïnstalleerd, waar ook puppet agent applicatie aanwezig was.

Daarnaast werd de icingaweb2 geïnstalleerd. Icingaweb2 is een nieuwe moderne web applicatie, special ontwikkeld voor Icinga2. Applicatie heeft moderne dashboard waar statussen van gemonitorde componenten verschijnen. </br> </br>

Installatie van Icinga2

Tijdens project werd icinga2 op test server door middel van distributie's package manager geïnstalleerd. Eerst moer er een repository aan de package manager toegevoegd, afhankelijk van besturingssysteem. Voor Ubuntu systeem werd volgende commando's gebruikt om icinga2 te installeren.

```
# add-apt-repository ppa:formorer/icinga
# apt-get update
# apt-get install icinga2
```

Om installatie en status van icinga2 te verifiëren werd `service icinga2 status` gebruikt.

Installatie van Icingaweb2

Icingaweb2 applicatie had voorgeïnstalleerde componenten nodig zo als MySQL database, apache2 webserver en DB IDO plugin. DB IDO (Database Icinga Data Output). Zoals naam al zelf zegt, werd DB gebruikt om gemonitorde data van Icinga2 applicatie op te slaan, om in vervolg informatie te tonen op icingaweb2 applicatie. Commando's die tijdens project gebruikt waren voor Ubuntu systeem.

```
# apt-get install mysql-server mysql-client
# apt-get install icinga2-ido-mysql
```

Na de installatie van MySQL, werd er een database aangemaakt en een schema aan database toegevoegd.

```
# mysql -u root -p

mysql>  CREATE DATABASE icinga;
        GRANT SELECT, INSERT, UPDATE, DELETE, DROP, CREATE VIEW, INDEX, EXECUTE ON icinga.* TO 'icinga'@'localhost' IDEN
TIFIED BY 'icinga';
```

```
# icinga2 feature enable ido-mysql

# service icinga2 restart

# apt-get install apache2
```

Na de installaties van componenten werd er aan installatie van Icingaweb2 applicatie begonnen.

Icingaweb2 biedt twee mogelijkheden, installatie via setup wizard of handmatig.

Beide mogelijk heiden waren uitgetest, om werking ervan beter te begrijpen.

Om installatie via web wizard te beginne moest er naar ip address van server gebrowsed worden. `ip/icingaweb2`

```
# apt-get install icingaweb2
```

Installatie van Icinga2 applicatie via Puppet

Om icinga2 en webapplicatie via Puppet master instelleren, werd er 4 mein modules van puppet forge -gebruikt. Samen met die drie modules werd er nog een aantal modules me geïnstalleerd, die vereist zijn.

- Icinga2
- Icingaweb2
- Apache
- MySQL Installatie van een module verloopt als volgende. `puppet module install + naam van module`

Voor icinga2 werd er een module gebruikt die nog in laatste fase van ontwikkeling is. Module had genoeg componenten om gebruik te worden als oplossing van de probleem. Module was via git commando binnen gehaald en naar `/etc/puppet/modules` verplaatst.

Manifest file node.pp in `etc/puppet/environment/production/manifest` bij node declaratie, bevat drie classes. Elke klas bevat variabelen met waarde die via hiera in hieradata files gedeclareerd zijn.
`/etc/puppet/hieradata/node/icingamaster2.yaml` file waar variabelen gedefinieerd staan.

Voorbeeld. Icinga2 class installeert icinga2 applicatie op gedeclareerde node.

```
node 'icingamaster2'{
$ido_db_name = hiera{'icinga2::ido::name'}
$ido_db_pass = hiera{'icinga2::ido::pass'}
$ido_db_user = hiera{'icinga2::ido::user}


class  {'icinga2':
db_type => 'mysql'
db_host =>'localhost'
db_pass =>'$ido_db_pass'
db_name =>'$ido_db_name'
}
}
```

Besluit.

De installatie van Icinga2 applicatie en icingaweb2 web applicatie was voltooid. Er was een grote aantaal servers en services checks die van oude monitoring systeem naar Icinga2 waren overgebracht. Alle agent less cheks met basis notificatie zijn overgebracht naar nieuve monitoring systeem door middel van puppet atomatisatie tool. Tijdens project heb ik geleerd wat een automatisatie management tool is en hoe een monitoring systeem binnen een bedrijf eruitziet.