

portfolio Marc Hoogendoorn

Fontys ICT Infrastructure

FontysLogo

Who am I

Mijn naam is Marc Hoogendoorn ik ben 21 jaar oud, Ik was vorige semester al geïnteresseerd in automatisering/scripting. Dit semester is dus dan wel zeker een "blessing"

1.0 Planning

Omdat ik niet teveel vooruit wil werken heb ik er voor gekozen om deze sprint uitteplannen en de aankomende sprint voor de vakantie. Ik vind het een beetje onzin om een planning te maken voor het hele semester dan ben je toch bezig met waterval?

Hieronder zie je een van mijn planning van voor de kerstvakantie

Planning

na de kerst vakantie is dit echter helemaal veranderd en ben ik niet meer van plan om challenges te doen. Maar wil ik al mijn leerdoelen behalen binnen het project Doekoe.

Hierbij gebruik ik ook de planning die binnen ons teams staan

Planning

2.0 Leeruitkomsten

Hierbij alle leeruitkomsten voor alle onderdelen binnen dit semester en wat ik per leeruitkomst wil bereiken in dit semester.

2.1 leeruitkomsten Orchestration

[>] Deployen van een of meerdere webserver via ansible (Ubuntu Apache)

Team: Marc Hoogendoorn

Samenvatting: Een script gemaakt wat ervoor zorgt dat op verschillende server Apache word geïnstalleerd met een gerelateerde groep en user voor het beheer van de apache server

Bewijslast:

```
# - name: Reboot and going back to save point
#   reboot:
---
- name: creating a hapache webserver
  hosts: all
  become: yes
  become_user: root
  tasks:
    - name: Checking for updates/installing
      dnf:
        name: "httpd"
        state: latest
    - name: Starting up apache /enabling apache
      systemd:
        state: started
        name: "httpd"
        masked: no
        enabled: yes
    - name: Creating the group webdevelopers
      group:
        name: "web-developers"
        state: present
    - name: Creating user and adding him to web-devevelopers
      user:
        name: "web-dev-employee"
        group: "web-developers"
        create_home: no
    - name: Unlock password and set it to empty
      command: passwd -d web-dev-employee
    - name: Change httpd folders right
      file:
        path: /etc/httpd
        owner: "web-dev-employee"
        group: "web-developers"
        mode: '0665'
    - name: Changing conf folders rights and group ownership
      file:
        path: "{{ item }}"
        recurse: yes
        owner: "web-dev-employee"
        group: "web-developers"
        mode: '0665'
      with_items:
        - '/etc/httpd/conf'
        - '/etc/httpd/conf.d'
        - '/etc/httpd/conf.modules.d'
    - name: Changing the users homegroup to the HTTPD folder
      user:
        name: "web-dev-employee"
        home: '/etc/httpd'
```

Screenshots:

Deployen van DHCP scopes op een Windows enviornment (Via ansible)

Windows Docker Containers

Team: Marc Hoogendoorn

Samenvatting: Op het moment werken we met Windows server vm's voor het configureren van onze omgeving dit kost echter redelijk veel resources nu kwam ik er laatst achter dat er ook windows core docker containers bestaan die je hierbij helpen!. Dus nu gaan we kijken of we de Windows omgeving op het netlab server kunnen configureren op een Windows Core server

Relay van DHCP via TAN VPN

Team: Marc Hoogendoorn

Samenvatting: Voor het begrip van centraal beheer in onze omgeving/Effiecenty wouden wij het Relayen van DHCP request naar de centralen server in, Hiervoor bleek het dat er wel wat dingen in de infrastructuur veranderd moesten worden. Sinds we eerst gebruik maakte van een TUN VPN, die alleen gebruik maakt van Layer 3 binnen het OSI model Nu is de VPN veranderd naar een TAN vpn die gebruik maakt van Layer 2 en Layer 3 sinds DHCP zich vastbind aan Layer 2 pakketen.

Screenshots:

tapvpn

tapvpn

tapvpn

tapvpn

[>] Virtual Machine Deployment InfralablabServer (Via ansible)

team: Marc Hoogendoorn

Samenvatting: Ik wou kijken of ik een script kon maken waarin ik images gebruiken om gemakkelijk Virtual machines kon deployen op onze HyperV server zonder telkens, Opnieuw en Opnieuw door het instalatie process zou moeten

Bewijslast:

``` powershell

## !powershell

## AnsibleRequires -CSharpUtil Ansible.Basic

```
$spec = @{ options = @{ Action = @{ type = "str"; choices = 'Create','Delete'; required = $true} VMname = @{ type = "str"; required = $true} VMgeneration = @{ type = "int"; choices = 1, 2; default = 2} VMos = @{ type = "str"; choices = 'Linux','Windows'; required = $true} VMlocation = @{ type = "str"; default = 'C:\Users\Public\Documents\Hyper-V\Virtual hard disks' } VMcpu = @{ type = "int"; default = 1} startupMemory = @{ type = "str"; default = 2GB} minimalMemory = @{ type = "str"; maximumMemory = @{ type = "str"; VMswitch = @{ type = "str"; imageName = @{type = "str"; imageFolder = @{type = "str"; default = 'C:\image' } } }
```

```
$module = [Ansible.Basic.AnsibleModule]::Create($args,$spec)
```

## Define the Arugments

```
$Action = $module.params.Action $VMname = $module.params.VMname $VMgeneration = $module.params.VMgeneration
```

## \$VMos = \$module.params.VMos

```
$VMlocation = $module.params.VMlocation $VMcpu = $module.params.VMcpu
```

```
$startupMemory = $module.params.startupMemory $minimalMemory = $module.params.minimalMemory $maximumMemory = $module.params.maximumMemory
```

```
$VMswitch = $module.params.VMswitch $imageName = $module.params.imageName $imageFolder = $module.params.imageFolder
```

```
function test-hyperV{ $hyperv = Get-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V if($hyperv.State -ne "Enabled") { $module.FailJson("Hyper-V doesn't seem to be running on the Desginated Machine") } }
```

```
function new-HyperVM{
```

```

 #if(Test-Path -Path $VMLocation -eq $false){
 # $module.FailJson("The path $Vmlocation doesn't exist")
 #}

 $CheckVM = Get-VM -name $VMname -ErrorAction SilentlyContinue
 $CheckVMHardisk = Get-ChildItem -Path $VMLocation | Where-Object {$_.Name -eq "$VMname.vhdx" -or $_.Name -eq "$VMname"}
 if($CheckVMHardisk -or $CheckVM){
 $module.FailJson("There is already an Disk or VM created with the name $VMname; This Script try's to Create Unique Names for eas
 }

 try {
 New-VM -name $VMname -Generation $VMGeneration -Path $VMLocation
 }
 catch {
 $errmsg = $_.exception.message
 $module.FailJson("there was an error message: $errmsg")
 }
}

function set-VCPU{ try { Set-VMProcessor $VMname -Count $VMcpu } catch { $errmsg = $_.exception.message $module.FailJson("there was an error setting the
Virtual CPU: $errmsg")
}}

function set-VHDFromImage{ $ImagePath = Get-ChildItem -Path $ImageFolder -Filter *.vhdx | Where-Object {$_.Name -eq "$ImageName.vhdx"} | Select-Object
FullName

 if (!$ImagePath){
 $module.FailJson("Coudl not find an image in $imageFolder called: $imageName")
 } else {
 $ImagePath = $ImagePath.FullName
 }

 try {
 $VmPath = New-Item -ItemType Directory -Path "$vmlocation\$vmname\Virtual Hard Disks\"
 $VmPath = $VmPath.FullName

 Copy-Item "$imagepath" -Destination "$VmPath\$VMName.vhdx"
 $VHDpath = Get-ChildItem -Path $VmPath | Where-Object {$_.Name -eq "$VMname.vhdx"}
 $VHDpath = $VHDpath.FullName
 }
 catch {
 $errmsg = $_.exception.message
 $module.FailJson("There was an problem with copying the VMDisk to the right location message: $errmsg")
 }

 try {
 Add-VMHardDiskDrive -VMName $VMname -Path $VHDpath
 }
 catch {
 $module.FailJson("There was an error attaching the VirtualHardisk to the VM $VMname")
 }
}

function set-VRAM { try { if(!$minimalMemory -or !$maximumMemory){ Set-VMMemory $VMname -DynamicMemoryEnabled $false -StartupBytes $startupMemory
} else { Set-VMMemory $VMname -DynamicMemoryEnabled $True -StartupBytes $startupMemory -MinimumBytes $minimalMemory -MaximumBytes
$maximumMemory } } catch { $errmsg = $_.exception.message $module.FailJson("there was an error setting the VM's ram; message: $errmsg") } }

function set-VSWITCH{ try { Get-VMNetworkAdapter -VMName $VMname | Connect-VMNetworkAdapter -SwitchName $VMswitch } catch { $module.FailJson("There
was an error settings the VM's network adapter: $errmsg") } }

if($Action -eq "Create"){ test-hyperV new-HyperVM

```

```

 if($VMcpu){
 set-VCPU
 }
 if($imageName -and $imageFolder){
 set-VHDFromImage
 }
 if($startupMemory){
 set-VRAM
 }
 if($VMswitch){
 set-VSWITCH
 }

```

```

} elseif ($Action -eq "Delete") {}

```

```

$module.Exitjson();

```

```

''' yml

- name: CustomModule (win-vm-fromImage) Test Playbook
 hosts: WIN-INFRALAB.doekoe.local
 tasks:
 - name: Create Windows Coreserver from an image
 win-vm-fromImage:
 Action: "{{ Action }}"
 VMname: "{{ VMname }}"
 VMos: Windows
 imageName: "{{ ImageName }}"

```

*Screenshots:*

## 2.2 leeruitkomsten netwerk orchestration

---

### Configureren van PFsense via Ansible (VOLgende sprint)

## 2.3 leeruitkomst Automation

---

### Configureren van kerberos Protocol voor ansible

*Team:* Marc Hoogendoorn

*Samenvatting:* Ik heb gewerkt met het kerberos protocol om er zo voor te zorgen dat ansible op een secure en gemakkelijke manier op machines binnen het domain kan inloggen

*Bewijslast:*

```
To opt out of the system crypto-policies configuration of krb5, remove the
symlink at /etc/krb5.conf.d/crypto-policies which will not be recreated.
includedir /etc/krb5.conf.d/
```

```
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log
```

```
[libdefaults]
 dns_lookup_realm = false
 ticket_lifetime = 24h
 renew_lifetime = 7d
 forwardable = true
 rdns = false
 pkinit_anchors = FILE:/etc/pki/tls/certs/ca-bundle.crt
 spake_preauth_groups = edwards25519
 default_realm = DOEKOE.LOCAL
 default_ccache_name = KEYRING:persistent:%{uid}
```

```
[realms]
DOEKOE.LOCAL = {
 kdc = win-seraphine.DOEKOE.LOCAL
 admin_server = win-seraphine.DOEKOE.LOCAL
 default_domain = DOEKOE.LOCAL
 kpasswd_server = win-seraphine.DOEKOE.LOCAL
}
TESTENV.DOEKOE.LOCAL = {
 kdc = win-testenv1.TESTENV.DOEKOE.LOCAL
 admin_server = win-testenv1.TESTENV.DOEKOE.LOCAL
 default_domain = TESTENV.DOEKOE.LOCAL
 kpasswd_server = win-testenv1.TESTENV.DOEKOE.LOCAL
}
```

```
[domain_realm]
.doekoe.local = DOEKOE.LOCAL
doekoe.local = DOEKOE.LOCAL
.testenv.doekoe.local = TESTENV.DOEKOE.LOCAL
testenv.doekoe.local = TESTENV.DOEKOE.LOCAL
```

*Screenshots:*

Planning

*Testen vand Ansible Machine na het configueren van de conf file*

Planning

*Custom Credential gemaakt voor het verbinden via kerberos zodat het wachtwoord encrypt is*

Planning

\*Connectie via kerberos is geslaagd bij een simpel test script\*\*

## [>] Configureren van openSSH key For Ansible account for secure connection

*Team:* Marc Hoogendoorn

*Samenvatting:* Een script wat voor het configueren van van asnible user account op remote machines regelt

*Bewijslast:*

```
Pre work

If there is No user yet called sysop
sudo -i
useradd -m -s /bin/bash sysop
passwd Doekoe2020!
```

```

echo -e 'sysop\tALL=(ALL)\tNOPASSWD:\tALL' > /etc/sudoers.d/sysop

Creates a simple user to Show to teacher
sudo -i
useradd -m -s /bin/bash Test123
passwd Test123

- hosts: all
 vars:
 - public_key: "{{ lookup('file', '/home/ansible-op/.ssh/id_rsa.pub') }}"
 gather_facts: no
 become: yes
 tasks:

 - name: Creating the user ansible-op
 user:
 name: "ansible-op"

 - name: Add ansible-op user to the sudoers
 copy:
 dest: "/etc/sudoers.d/ansible-op"
 content: "ansible-op ALL=(ALL) NOPASSWD: ALL"

 - name: Deploy SSH Key to remote Machine
 authorized_key:
 user=ansible-op
 key="{{ public_key }}"
 state=present

 - name: Disable Password Authentication for all users
 lineinfile:
 path: /etc/ssh/sshd_config
 regexp: '^PasswordAuthentication'
 line: "PasswordAuthentication no"
 state: present
 backup: yes

 - name: Enable Password Authenticaiton for Sysop Users
 blockinfile:
 path: /etc/ssh/sshd_config
 insertafter: '^PasswordAuthentication'
 block: |
 match user sysop
 PasswordAuthentication yes
 state: present
 backup: yes

 - name: Disable Root Login
 lineinfile:
 path: /etc/ssh/sshd_config
 regexp: '^PermitRootLogin'
 line: "PermitRootLogin no"
 state: present
 backup: yes
 notify:
 - Restart ssh

 handlers:
 - name: Restart ssh
 service:
 name: sshd
 state: restarted

```

screenshots:

☐

*Aan maken van een test gebruiker*

☐

*De gebruiker heeft op het moment rechten om inteloggen*

☐

*Ansible Account met alleen een private key*

☐

*ansible kan nog niet inloggen met dit account*

☐

*Ansible script word gedraaid via het admin account*

☐

*Ansible kan nou draaien via het ansible account*

☐

*Test account kan niet meer inloggen via SSH*

☐

*ADmin account heeft nog wel de mogelijkheid om inteloggen*

## [>] Script gemaakt voor het onboarding en ofboarding van personal in het AD version 1

*Team:* Marc Hoogendoorn & lisa

*Samenvatting:* Een script wat zorgt voor het onboarding en ofboarding van gebruikers binnen windows active directory het script maakt gebruik van een SQL database voor het verkrijgen van zijn data.

*Bewijslast:*



```
#trying to invoke powershell crap in ansible so that we can do all kinds of awesome things
We enjoy doing kinda awesome things like voquein

- name: invoke SQL powershell
 hosts: all
 connection: winrm
 tasks:
 - name: Create user
 win_shell: |
 $data = Invoke-Sqlcmd -Query "SELECT * FROM [Challenge].[dbo].[user2] WHERE Creation = 1" -ServerInstance "srv-winess19"
 foreach($employee in $data)
 {
 Write-output "Deze statemennt was TRUE en ik ga nu " + $employee.username + " aanmaken"
 function Get-RandomCharacters($length, $characters)
 {
 $random = 1..$length | ForEach-Object { Get-Random -Maximum $characters.length }
 $private:ofs=""
 return [String]$characters[$random]
 }
 function Scramble-String([string]$inputString)
 {
 $characterArray = $inputString.ToCharArray()
 $scrambledStringArray = $characterArray | Get-Random -Count $characterArray.Length
 $outputString = -join $scrambledStringArray
 return $outputString
 }

 $password = Get-RandomCharacters -length 5 -characters 'abcdefghijklmnopqrstuvwxyz'
 $password += Get-RandomCharacters -length 1 -characters 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
 $password += Get-RandomCharacters -length 1 -characters '1234567890'
 $password += Get-RandomCharacters -length 1 -characters '!'

 Write-Host $password

 $password = Scramble-String $password

 Write-Host $password

 $SecurePassword = $password | ConvertTo-SecureString -AsPlainText -Force

 New-ADUser -Name $employee.username -GivenName $employee.firstname -Surname $employee.lastname -SamAccountName $employee.use
 Invoke-Sqlcmd -Query "UPDATE Challenge.dbo.user2 SET creation = 0 WHERE ID = $($employee.ID)" -ServerInstance "srv-winess19"
 Invoke-Sqlcmd -Query "INSERT INTO Challenge.dbo.passwords (ID,Username>Password) VALUES ($($employee.ID), '$($employee.usern
)"
 }
 - name: Disable Users
 win_shell: |
 $data = Invoke-Sqlcmd -Query "SELECT * FROM [Challenge].[dbo].[user2] WHERE State = 0" -ServerInstance "srv-winess19"
 foreach($employee in $data)
 {
 Write-output "Deze statemennt was False en ik ga nu " + $employee.username + " opblazen"
 Disable-ADAccount -Identity $employee.username
 }
 - name: Delete Disabled users
 win_shell: |
 $users = Get-ADUser -Filter * | where {($_.enabled -eq $false)}
 Write-Output "We are deleting $users now after 90 days"
 Write-Output "pIEP POOP BAB IK BEN ROBOT"
```

screenshots:

SQL

De sql database

\*De sql tabel

## Maken van een Custom powershell module voor Ansible

*Team:* Marc Hoogendoorn

*Samenvatting:* Ik zat dus met het probleem dat ik niet goed wist hoe ik een groot script binnen een Ansible playbook kon krijgen zonder dat het er absurd en lelijk en eigenlijk maar slecht uitziet en totaal niet leesbaar is wat nauwkeuriger het tegenovergestelde is van een Ansible Playbook. (zie Onboarding en ofboarding Version 1). Hierdoor ben ik gaan experimenteren met verschillende manieren van het maken van ansible scripts eerst was ik begonnen met het schrijven van kleinere actions binnen de playbook met kleine stukjes van powershell scripts, Maar dit leverde het probleem dat variabelen niet gemakkelijk tussen de Actions gedeeld kunnen worden en al helemaal niet als ze eigenlijk niet van ansible zijn maar van powershell!.

De oplossing was dus ik moest een custom powershell module maken. om dit een beetje te leren en te testen ben ik begonnen met een simpele module voor het verwijderen en toevoegen van een groep aan een user binnen het AD.

*Bewijslast:*

```
#!/powershell
#AnsibleRequires -CSharpUtil Ansible.Basic

$spec = @{
 options = @{
 state = @{ type = "str"; required = $true }
 member = @{ type = "str"; required = $true }
 group = @{ type = "str"; required = $true }
 }
}

$module = [Ansible.Basic.AnsibleModule]::Create($args,$spec)

$state = $module.Params.state
$member = $module.Params.member
$group = $module.Params.group

$state = $state.ToString().ToLower()
If (($state -ne "present") -and ($state -ne "absent")) {
$module.FailJson("state is '$state'; must be 'present' or 'absent'")
}
$membertype = 'user'

$memberisuser = get-aduser -filter "name -eq '$member'"
if ($null -eq $memberisuser){
 $membertype = 'group'
 $memberisgroup = get-adgroup -filter "name -eq '$member'"

 if ($null -eq $memberisgroup){
 $module.FailJson("$member is not a User Or Group Existing in Active Directory.")
 }
}

$groupisthere = get-adgroup -filter "name -eq '$group'"
if ($groupisthere){
 $groupname = $groupisthere.name
}
else{
 $module.FailJson("$group Is not a existing group in active Directory ")
}

$groupmembers = get-adgroupmember -identity $groupname | Select-Object -expandproperty name

if ($state -eq "present"){
```

```

1+($state -eq 'absent'){
 if($groupmembers -contains $groupname){
 $module.result.msg = "The $membertype : $member; is already a part of the Group: $group"
 }
 else {
 Add-adGroupMember -identity $groupname -members $member
 $module.result.changed = $true
 $module.result.msg = "The $membertype : $member; Has been Added to the Group $group"
 }
}

elseif($State -eq 'present'){
 if($groupmembers -contains $groupname){
 Remove-adGroupMember -identity $groupname -members $member
 $module.result.changed = $true
 $module.result.msg = "The $membertype : $member; Has been removed from the group $group"
 }
 else {
 $module.result.msg = "The $membertype : $member; Was not part of the Group $group"
 }
}

$module.Exitjson();

```

*screenshots:*

CustomModule

Het proberen van verschillende Actions te maken om zo powershell code leesbaarder te maken binnnen een playbook

CustomModule

Een simpel playbook na het maken van de custom module die iets veel complexers doet dan de screenshot hierboven

CustomModule De groep heeft nog geen gebruikers

CustomModule

Het script heeft gedraaid

CustomModule

De groep bezit nu wel members

## Configureren van Firewall voor Linux

*team:* Marc Hoogendoorn

*Samenvatting:* Voor ons project hadden we een klein scriptje nodig waarmee we firewall regels gemakkelijk konden toevoegen. Hiervoor heb ik een simple startup script gemaakt waarop we later verder kunne bouwen

*Bewijslast:*

```

- name: Creating Firewall Rules For ubuntuMachine With input
 hosts: all
 become: yes
 become_user: root
 tasks:

- name: check if UFW is enabled and deny acces to all ports
 community.general.ufw:
 state: enabled
 policy: deny

- name: allow acces through SSH
 community.general.ufw:
 rule: allow
 name: OpenSSH

- name: Allow access to multiple ports
 community.general.ufw:
 rule: allow
 port: "{{ item.port }}"
 proto: "{{ item.proto }}"
 loop:
 - {port: 43, proto: udp}
 - {port: 44, proto: tcp}
 - {port: 45, proto: udp}
 - {port: 46, proto: tcp}

```

## 2.4 leeruitkomsten Monitoring

## 2.5 Leeruitkomsten Security

### [?] Opzetten VPN

## 2.6 Leeruitkomsten Support services

### [!] Maken van servicedesk

### [!] Opmaken van SLA

## 2.7 Leeruitkomsten Future Oriented Organizing

## 2.8 Leeruitkomsten Personal Leadership

### [!] Project leider

*Samenvatting:* Ik ben project leider van het groepje

### [!] Buitenlandse stagen gedaan bij TCS in Budhapest

*samenvatting:* Ik heb in mijn MBO tijden gewerkt voor een halfjaar in budapest voor buitenlandse Stagen.

## 2.9 Leeruitkomsten Problem solving

### [!] Ansible Variablen

### [!] Onderzoeken

### [!] NetwerkTekeningen en itteraties van de omgeving

### [!] Windows server teamviewer start automatische op

## 2.10 Leeruitkomsten Target Oriented Interact

# 4.0 Ons Project

---

Voor mijn eind project werk ik in het groepje Doekoe, hierin werk ik aan kassa systemen

## 4.1 Projectgroep

## 4.2 Onderzoeksvragen

Welke manier heeft de voorkeur om de verschillende systemen aan elkaar te verbinden?

Op welke manier is het mogelijk om data van de kassa te verweken?