

# Eindopdracht TCSB-V2AUT-16 v6

## Inleiding

Sportschool “Jenna Altijd Fit” (JAF) wordt groot! Logisch ook, Jenna heeft altijd hard gewerkt en door slim te investeren is zij andere sportscholen, waaronder die van Benno, mijlenver vooruitgestreefd. De volgende stap voor Jenna is een nieuwe ICT-infrastructuur: Efficiënt, schaalbaar en kosteneffectief, want beheerders zijn er weinig en zijn bovendien duur. Jullie snappen het, dat wordt OSD, CM, Containerisatie en Scripting natuurlijk. Jenna is ook niet gek, domweg investeren in die technologie is niet handig: Eerst een Proof of Concept (PoC), daarna groots uitrollen. Het netwerk? Dat zal wel werken, niet belangrijk. De systemen, daar gaat het om:

- Servers: Windows en Linux hè. Deel on-premise, ander deel in je (private) cloud. Ze verzorgen de storage van je data, zijn veilig en bieden de website en monitoren de omgeving.
- En gaat er dan toch wat mis? Dan zien de beheerders dat en grijpen onmiddellijk in.
- Hieronder de details. Maak de opdracht in tweetallen. Het resultaat geldt als het eindcijfer voor de praktijktoets van dit vak.

Ook namens Jenna: Sportieve Groet en Veel Succes!

**Deadlines: Inleveren scripting: zondag 31 oktober 2021 via Canvas.**

**Assessments (onder voorbehoud van roostering): 1 november t/m 4 november. Tijdstip wordt in overleg met je docent geregeld.**

## De Opdracht, onderdeel “Tooling”

Gebruik slimme tooling voor **OSD/Provisioning** gebruikmakend van **LTI** of beter. Houd er rekening mee, dat de omgeving **hybride** is (Linux/Windows). Windows OSD hebben we niet in de labs behandeld, kortom dat zijn bonuspunten. Linux distributie is bij voorkeur RHEL8 of CentOS8. **CentOS7 mag niet**. Wil je liever een andere linuxdistributie gebruiken en is deze release modern (uitgebracht na 1-jan-2021) dan mag dat ook.

Configuratie Management gaat voornamelijk **declaratief** gebruikmakend van **Saltstack**. Gebruik gerust **DSC** waar dit bruikbaar is.

Voor de website van Jenna zijn **containers** een logische keuze. Hoe geavanceerd die website is? Geen idee. Je hoeft de containers niet via Saltstack uit te rollen, mag wel natuurlijk (maakt het resultaat nog beter). Het gebruik van docker compose is een optie.

Tooling wordt ook gebruikt om je scripts te “**deployen**” en te starten.



## De Opdracht, onderdeel "Scripting"

Het management wil graag informatie over de in het netwerk gebruikte systemen. Dit moet heel flexibel zijn en bovendien platformonafhankelijk. Daarom is gekozen voor een eigen oplossing. Dat geeft immers de meeste flexibiliteit.

Het management wil dat de opgevraagde informatie op een begrijpelijke en visueel aantrekkelijke manier wordt gepresenteerd. Jullie moeten hiervoor Python (en PowerShell) scripts aanmaken waarmee dit mogelijk wordt. Dit gaat via een agent:

- **De Agent:** Op de systemen draait een agent script dat informatie kan opvragen van het systeem waarop het is geïnstalleerd. Schrijf hiervoor agent-code die op regelmatige basis een aantal zaken monitort en deze informatie stuurt naar de centrale management computer via HTTP. Maar ja, hoe stuur je nu informatie naar de server? Om je een beetje te helpen, zie hieronder een voorbeeld:

Als voorbeeld. Neem een webbrowser, hoe stuurt deze eigenlijk informatie naar de server:



..."onder water", kan je ditzelfde ook doen met python code. Je moet deze dan wel regelmatig starten (via crontab/systemd/at/task scheduler etc).

```
1 # v7 eindopdracht Client
2 # https://docs.python.org/3/howto/urllib2.html
3 # Python 3.2.5
4
5 import urllib.parse
6 import urllib.request
7
8 # Base URL being accessed
9 url = 'http://192.168.37.151/cgi/python CGI3.py'
10
11 # Make a PUT request and read the response
12 payload = {
13     'first_name' : 'value1',
14     'last_name' : 'value2'
15 }
16
17 data = urllib.parse.urlencode(payload)
18 data = data.encode('ascii') # data should be bytes
19 req = urllib.request.Request(url, data)
20 with urllib.request.urlopen(req) as response:
21     the_page = response.read()
22
23 print(the_page)
24
```

```
Python Interpreter
*** Python 3.2.5 (default, May 15 2013, 23:06:03) [MSC v
.1500 32 bit (Intel)] on win32. ***
>>>
b'\n\n<html>\n\n<head>\n\n<title>Hello - Second CGI
Program</title>\n\n</head>\n\n<body>\n\n<h2>Hello value1
value2</h2>\n\n</body>\n\n</html>\n\n'
```

De server code die bij dit voorbeeld hoort (zie de lesstof over CGI) is:

```
python CGI3.py - Notepad
File Edit Format View Help
#!/usr/bin/python

# Import modules for CGI handling
import cgi, cgitb
cgitb.enable()

# Create instance of FieldStorage
form = cgi.FieldStorage()

# Get data from fields
first_name = form.getvalue('first_name')
last_name = form.getvalue('last_name')

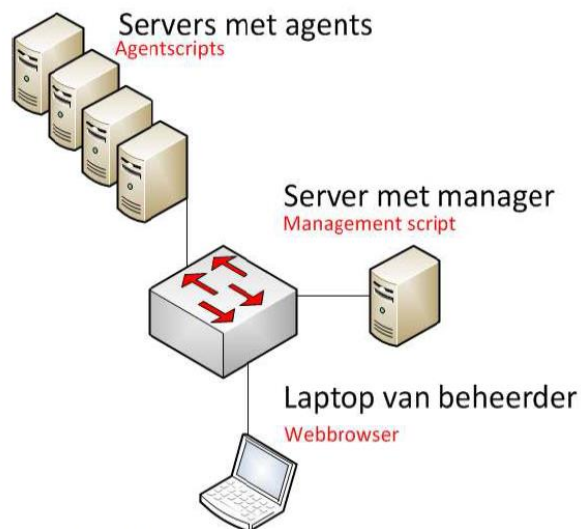
print "Content-type:text/html\r\n\r\n"
print "<html>"
print "<head>"
print "<title>Hello - Second CGI Program</title>"
print "</head>"
print "<body>"
print "<h2>Hello %s %s</h2>" % (first_name, last_name)
print "</body>"
print "</html>"
```

Kortom, data verzamelen en in de database stoppen. Als beheerders aanloggen dan kan je de resultaten tonen door de database te raadplegen.

- **Op een centrale management computer draait het management script** dat de informatie van de clients verzamelt en opslaat in een database. Daarnaast heeft deze managementwebsite ook een webpagina die je met een webbrowser kan benaderen: Deze webpagina haalt informatie uit de database en toont deze overzichtelijk op het scherm waarbij je bij voorkeur gebruik maakt van matplotlib om de informatie grafisch weer te geven. De grafiek wordt real time gegenereerd als deze in de website wordt opgevraagd.

Het zou heel interessant zijn als je binnen jouw website in staat bent om informatie van jouw systemen te combineren met informatie afkomstig van een openbare bron op het Internet, bijvoorbeeld: buienradar (om temperatuur buiten af te zetten tegen temperatuur van jouw systemen),. Dit is maar een voorbeeld, verzin liever zelf iets "nuttigs".

- Een overzicht van de hele scriptingoplossing:

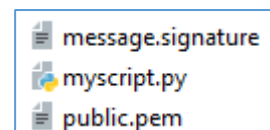


Figuur 1: Grafische situatieschets

### Waar de scripts aan moeten voldoen:

- Er moeten ten minste twee scripts opgeleverd worden die op aparte machines gerund worden. Het agent script draait op de servers en het management script op een ander systeem. Het mag op (verschillende) virtual machines die op de (eigen) laptop draaien. Het geheel op meerdere machines demonstreren mag natuurlijk ook.
- Maak gebruik van minimaal twee agents, bij voorkeur (niet verplicht) een linux en een windows agent.
- Code moet goed leesbaar zijn en voorzien zijn van commentaar in de code.
- Zorg ook voor een INSTALL.txt waarin je beschrijft wat de beheerafdeling moet doen om je code te gebruiken (uitgaande van een fresh install van het OS).
- Je code mag niet lijken op de code van anderen (plagiaat controle). Wel mag je regels code uit deze opdracht gebruiken.
- Je moet de bestanden (code/handleiding/configuratie/screendump) uploaden naar Canvas met daarbij de naam van jezelf en eventueel degene met wie je hebt samengewerkt (niet meer dan 2 studenten per script).
- Als 1 van de 2 studenten de scripts uploaden is het voldoende, je hoeft het niet allebei te doen.
- Voor het digitaal ondertekenen: Gebruik het onderstaande key pair en gebruik bestandsnamen zoals hieronder afgebeeld. Kijk dit onderdeel zelf na! Dat kan je doen via deze url: [http://mqtt-iot-csc-hu.westeurope.cloudapp.azure.com:8884/cgi-bin/cgi\\_verify.cgi](http://mqtt-iot-csc-hu.westeurope.cloudapp.azure.com:8884/cgi-bin/cgi_verify.cgi)

```
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCgUpB8X/Yr0eZldQ8nBfDm/10F
v0m4yNYbM1IHNYsEaEZUVFPD8DUttXs1bipurVMPP6IGxJZHer7UX4mVi0y4b9Fh
AkGhrn3mTZrH9sspZVXUmxhfAzxGxrPsi+9Y3/GjE30v8S3cVDI79YXSP2K0NJUG
GRR4N8haCI/zAR9c1QIDAQAB
-----END PUBLIC KEY-----
-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQCgUpB8X/Yr0eZldQ8nBfDm/10Fv0m4yNYbM1IHNYsEaEZUVFPD
8DUttXs1bipurVMPP6IGxJZHer7UX4mVi0y4b9FhAkGhrn3mTZrH9sspZVXUmxhf
AzxGxrPsi+9Y3/GjE30v8S3cVDI79YXSP2K0NJUGGRR4N8haCI/zAR9c1QIDAQAB
AoGABSZp1Ej73i/9s/Qk8ra25KC+4vyM0H4JHLITGD6w0skruzFDSJfKiYebJ1po
3An00X5UjN2bSzwG7ckM4S0qqHQMJsB/xxjtgVlQ2PFvVrSLaTAnDPL0bfN5+T4
eeDSyF7enWqsIBVoMmtwEvYapa1k94zRpcsdyPr9YQAWq0ECQQC/5eFLpFM/jGZ2
rTz6j0viekSFxvCXQMVoQ9BkrV1QQX8oJKgFFfLQ70VS9BMs/rLmaV1N11aKpiNz
4xxb5qJVAkEA1eCD66+/YCAN1FXWOnOCA5nzIrGxt4kjbb6cIHu4uG91DFsBE+HU
Wa3/N9XDjKwJHuCpx03IFd8rUXo5LSaRQQJBAJogK0SSt47i0xWkIP9ZaiAJVsA
/qms9yCYHspr1Z+pC1Uvvo/F5wG13Z3Sp3mDsLIONJEG1sEppx93DEjrp0YUCQQCg
RYG2IVpBF2Q1pTlHTf204y305tRxhgQVnkV0QoDhvSA09RFgq+1sVrY1S1wi/pDR
d7viYgN09HrWbL1AbzBAkEAtSA1N5pNJ/ZcJU0K1sYbR30jPXcUjyzmKp6HTRVn
R00nvaGSyi3Gy94R7s36v2g32uKkqhpSOy564FfKoyq4pg==
-----END RSA PRIVATE KEY-----
```



## Beoordeling:

In het beoordelingsformulier vind je de criteria voor de beoordeling. Opmerkingen hierbij:

- **Student(en) leveren het onderdeel “scripting” aan middels upload via Canvas.** Vermeld daarbij de naam van jezelf en eventueel degene met wie je hebt samengewerkt (niet meer dan 2 studenten per script). Doe dit voorafgaand aan het assessment (zie deadlines). Er moet ten minste opgeleverd worden:
  - o De scripts,
  - o De configuratiefile, in ieder geval bij het management script,
  - o De installatiehandleiding voor de docent (INSTALL.txt).
  - o Een screendump(s) van de webpagina’s met het overzicht van de verzamelde data.
- Tijdens **het assessment** krijgen jullie de kans om deze software werkend te demonstreren aan je docent. Als je dan niet zelf aanwezig bent, of de software functioneert niet, dan krijg je “NA” als beoordeling en kan je het tijdens een herkansing opnieuw proberen.
- Het assessment wordt gestart met het onderdeel **tooling**: OS en software wordt gedeployd en als alles is geïnstalleerd en geconfigureerd kijken we naar de werking van je **scripts**.
- Tijdens het assessment kan de docent vragen stellen over de oplossing en/of de code. Beide studenten moeten vragen kunnen beantwoorden. Dit kan dus betekenen dat één student een voldoende krijgt en de ander niet (**individueel cijfer**).
- De deadline voor deze opdracht staat genoemd in de opdracht (tenzij de docent anders beslist).
- Je code mag niet lijken op de code van anderen (**plagiaatcontrole**).
- Als je alleen moest werken krijg je per onderdeel 10 extra punten.
- De punten die je kan halen zie je hieronder per onderdeel gespecificeerd. Let op als er staat: 0-20 betekent dit dat je maximaal 20 punten kan krijgen, dus alleen 20 punten als het perfect is uitgevoerd.
- Mocht je onverhoopt mee moeten doen aan de **herkansing**? In dat geval moet je andere optionele scripting elementen implementeren, of dezelfde opties, maar dan totaal anders (bv andere grafiek of beveiliging). Bij twijfel, overleg *vooraf* met je docent scripting.

## Excellentie:

We willen de beste opleveringen graag helpen om een ster te bemachtigen. Om je daarbij te helpen een aanwijzing: Een oplevering is “excellent” als je niet alleen deze opdracht goed hebt uitgevoerd, maar vooral als je een stap verder bent gegaan dan is gevraagd. Wij denken dan aan opties zoals hieronder zijn genoemd, maar misschien denk jij wel aan hele andere zaken?! Bespreek de mogelijkheden eventueel van tevoren met je docent.

- Logisch gebruik maken van <http://maas.io/> (metal as a service).
- Gebruik maken van een totaaloplossing zoals mogelijk is met <https://theforeman.org/>.
- Een heel uitgebreid script.



## Beoordeling eindopdracht Automation (TCSB-V2AUT-16)

	<b>Student 1</b>	<b>Student 2</b>
<b>Student(-en)</b>		
<b>Cijfer ***</b>		
<b>Studentnummer(s)</b>		
<b>Klas</b>		
<b>Jaar</b>	2021/2022	

\*\*\* Het individuele cijfer = eindcijfer, maar de verdeling tussen beide studenten is afhankelijk van het assessment.

	<b>Aantal punten</b>
<b>Kwaliteitscriteria Tooling (maximaal 100 punten)</b>	
- Alleen gewerkt? (10)	
- OSD wordt correct uitgevoerd, d.w.z.: (0-40) <ul style="list-style-type: none"> <li>o OS-Deployment werkt met LTI (minimaal 1 RHEL8/CentOS8 linuxsysteem = max 20 punten).</li> <li>o Meer dan die 20 punten krijg je door ook OSD van Windows te laten zien en/of een Windows ZTI te realiseren en/of "captured thick image".</li> </ul>	
- Configuratie Management: (0-60) <ul style="list-style-type: none"> <li>o Zinvol gebruik van pillars binnen je Saltstack configuratie.</li> <li>o Saltstack voor uitrol van de gemaakte scripts (incl. starten en test of het ook werkt).</li> <li>o Gebruik containers voor de uitrol van de website (geen punten voor de website zelf, wel voor containers die afhankelijk zijn van database en/of storage). Gebruik van containers i.c.m. Saltstack hoeft niet, mag wel.</li> <li>o Gebruik van docker compose voor multicontainer uitrol.</li> <li>o DSC is een optie, kan bruikbaar zijn bij configuratie van Windows.</li> </ul>	
<b>Totaal tooling</b> (aantal punten / 10 *) * Moet >= 5,5 zijn, anders is dit het eindcijfer van de eindopdracht.	<b>0,0</b>
<b>Kwaliteitscriteria Scripting (maximaal 100 punten)</b>	<b>Aantal punten</b>
- Alleen gewerkt? (10)	
- Deliverables zijn door de student tijdig opgeleverd, d.w.z. ten minste: (0-8) <ul style="list-style-type: none"> <li>o Scripts (minimaal twee),</li> <li>o Configuratiefile (minimaal één),</li> <li>o Handleiding: Installatiebeschrijving en aanwijzingen voor de beheerder,</li> <li>o Screenshot van een overzicht met data zoals te zien via je webbrowser.</li> </ul>	



<ul style="list-style-type: none"> <li>- Code is goed geschreven, dat wil zeggen: (0-12) <ul style="list-style-type: none"> <li>o Niet onnodig complex.</li> <li>o Op de juiste plaatsen voorzien van voldoende white spaces en newlines.</li> <li>o Voorzien van zinvol commentaar in het script.</li> <li>o Goede foutafhandeling waar nodig.</li> <li>o Creativiteit/elegantie van de oplossing.</li> <li>o Gebruik van functies/modules waar dit zinvol is.</li> </ul> </li> </ul>	
<ul style="list-style-type: none"> <li>- Het script werkt correct, tijdens het assessment dwz: (0-30) <ul style="list-style-type: none"> <li>o De scripts functioneren zonder fouten.</li> <li>o De webpagina(s) zijn aanwezig en functioneel.</li> <li>o Agents draaien regelmatig en communiceren met de server.</li> <li>o Relatieve database (SQLite of eigen keuze) wordt gebruikt.</li> <li>o Informatie uit de database wordt goed gepresenteerd via een webpagina.</li> <li>o De configuratie komt uit een configuratie-file. Moet zinvol worden gebruikt.</li> <li>o Er is logging.</li> <li>o Digital Signing: Sign 1 van je scripts, zie info in de tekst van deze opdracht. Zorg dat verify ook werkt (test het met de website).</li> <li>o Studenten kunnen eventuele vragen over de scripts beantwoorden.</li> <li>o Gevonden resultaten worden getoond in een grafiek (matplotlib) in de webpagina. Real time gegenereerd als je de pagina opvraagt.</li> </ul> </li> </ul>	
<ul style="list-style-type: none"> <li>- Optionele onderdelen zijn aanwezig: (0-50) <ul style="list-style-type: none"> <li>o De code is (deels) object georiënteerd.</li> <li>o Communicatie over het netwerk is volledig beveiligd en/of eventuele wachtwoorden staan niet leesbaar in de configuratie.</li> <li>o Er wordt gebruik gemaakt van een XML-configuratie file (weinig punten).</li> <li>o Er wordt een externe API gebruikt om de informatie in de database te verrijken met andere informatie (kan veel punten opleveren).</li> <li>o De managementwebsite wordt automatisch getest (kan veel punten opleveren).</li> <li>o Student heeft een of meer <i>eigen</i> toevoegingen bedacht.</li> </ul> </li> </ul>	
<p><b>Totaal scripting *</b> (aantal punten / 10)  * Moet &gt;= 5,5 zijn, anders is dit het eindcijfer van de eindopdracht.</p>	<b>0,0</b>
<p><b>Eindcijfer **</b> (totaal aantal punten van Tooling+Scripting / 2):</p>	

\*\* Als zowel scripting als tooling <55 punten dan is cijfer in Osiris het laagste van beide onderdelen.

<p><b>Toelichting</b></p>

