

howest
hogeschool

Research-Project

Installatiehandleiding Lagast Sean

Contents

| | |
|--|---|
| Inleiding | 3 |
| Benodigdheden | 3 |
| Start uitwerking | 4 |
| Stap 1: Voorbereiding directory | 4 |
| Stap 2: Update packages | 4 |
| Stap 3: Git installeren | 4 |
| Stap 4: Repo clonen | 5 |
| Stap 5: Powershell installeren | 5 |
| Stap 6: Azure cli installeren | 6 |
| Stap 7: Inloggen op azure | 6 |
| Stap 8: Terraform installeren | 7 |
| Stap 9: Crontab instellen | 8 |
| Stap 10: Systeem logging starten | 9 |
| Stap 11: Permissies op scripts instellen | 9 |
| Einde Installatie | 9 |

Inleiding

Dit project is bedoeld voor gebruikers die op een makkelijke manier terraform omgevingen willen opzetten voor educatieve redenen aan de hand van een simpel keuze script. Dit document legt alle stappen uit die je moet doorlopen om de begin omgeving op te zetten. Daarna wordt in de gebruikershandleiding ook uitgelegd hoe je deze begin omgeving kan uitbreiden om extra omgevingen in voor te programmeren.

Dit onderzoek kwam oorspronkelijk van een extern bedrijf dat soms cursussen geeft a.d.h.v. labo opstellingen die de 'docent (een intern persoon in het bedrijf)' telkens handmatig opstelt voor de cursisten. Daarom heb ik gekozen om via terraform te gaan werken. Je kunt makkelijk de opstelling die je wilt 1x gaan maken en uittesten, waarna je deze steeds opnieuw kan gebruiken bij verdere labo's. De uitbreiding die ik hier dan ook op gemaakt heb is dat alles dynamisch gebeurt voor meerdere gebruikers zonder zelf al te veel te hoeven doen.

Benodigheden

Om van start te kunnen gaan met dit project heb je een klein aantal zaken nodig. Aangezien dit project gebouwd en getest is op Azure, is het aangeraden deze omgeving ook te gaan gebruiken voor azure, maar je bent ook perfect in staat om terraform scripts te gaan gebruiken in bijvoorbeeld AWS. Dit project is gebouwd uit een mix van bash en powershell scripts die ervoor gaan zorgen dat je script dynamisch gaat gaan werken, en op meerdere users toepasbaar is. Persoonlijk heb ik op een windows omgeving gewerkt waar ik via wsl mijn script uitvoer, ook kun je op eender elke linux machine de komende configuratie gaan toepassen, natuurlijk zijn sommige commando's OS specifiek en zul je hier rekening mee moeten houden, oorspronkelijk is mijn omgeving ook getest op een ubuntu 20.04.3 VM-machine.

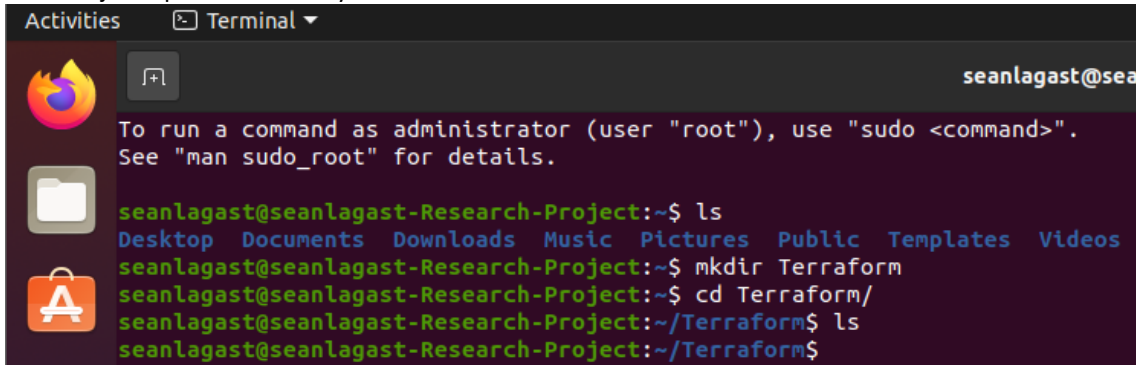
Hier ook nog eens de link naar de repo: <https://github.com/LagastSean/Research-Project>

Start uitwerking

Nu de start van de uitwerking. Zoals reeds vermeld is een linux OS zeker een must have (een ubuntu wsl omgeving werkt ook perfect), omdat de stappen die nu zullen volgen voorbereid zijn op een ubuntu 20.04.3 VM.

Stap 1: Voorbereiding directory

De 'applicatie' bestaat uit een git repository die later zal gecloned worden naar een bepaalde directory, hier kun je simpel een directory voor maken.

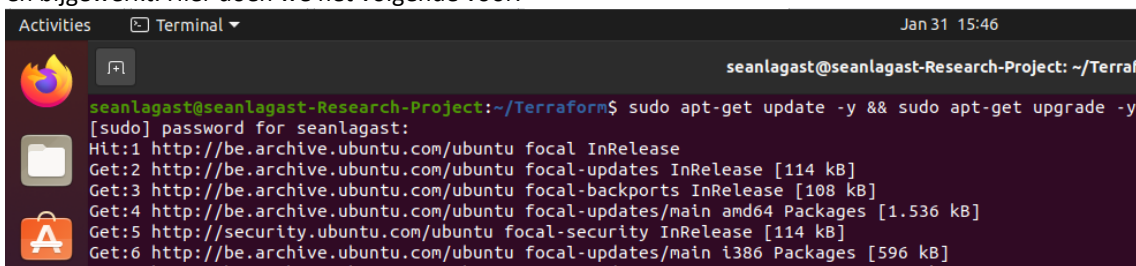
A terminal window titled 'Terminal' with a dark background. The user 'seanlagast@seanlagast-Research-Project' is at the prompt. The terminal shows the following commands and output: 'ls' lists the current directory contents: Desktop, Documents, Downloads, Music, Pictures, Public, Templates, Videos. Then 'mkdir Terraform' creates a new directory. Finally 'cd Terraform/' changes the current directory to Terraform. The prompt then shows '~ / Terraform\$' after the 'ls' command.

```
seanlagast@seanlagast-Research-Project:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
seanlagast@seanlagast-Research-Project:~$ mkdir Terraform
seanlagast@seanlagast-Research-Project:~$ cd Terraform/
seanlagast@seanlagast-Research-Project:~/Terraform$ ls
seanlagast@seanlagast-Research-Project:~/Terraform$
```

In dit voorbeeld wordt een Terraform directory aangemaakt.

Stap 2: Update packages

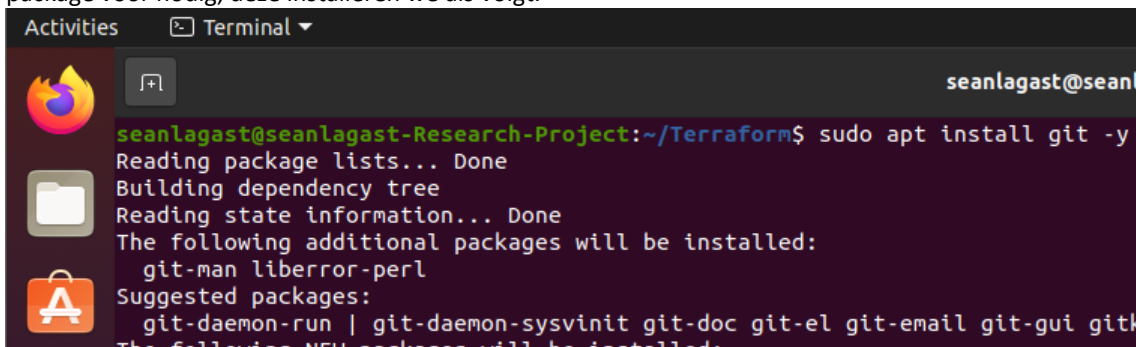
Om de simpelste stappen te doen laten werken moeten eerst de huidige packages vernieuwd worden en bijgewerkt. Hier doen we het volgende voor.

A terminal window titled 'Terminal' with a dark background. The user 'seanlagast@seanlagast-Research-Project' is at the prompt in the directory ~/Terraform. The terminal shows the command 'sudo apt-get update -y && sudo apt-get upgrade -y'. The output shows the system is updating from focal to focal-updates. It lists several packages being updated: focal-updates InRelease [114 kB], focal-backports InRelease [108 kB], focal-updates/main amd64 Packages [1.536 kB], focal-security InRelease [114 kB], and focal-updates/main i386 Packages [596 kB].

```
seanlagast@seanlagast-Research-Project:~/Terraform$ sudo apt-get update -y && sudo apt-get upgrade -y
[sudo] password for seanlagast:
Hit:1 http://be.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://be.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://be.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://be.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1.536 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:6 http://be.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [596 kB]
```

Stap 3: Git installeren

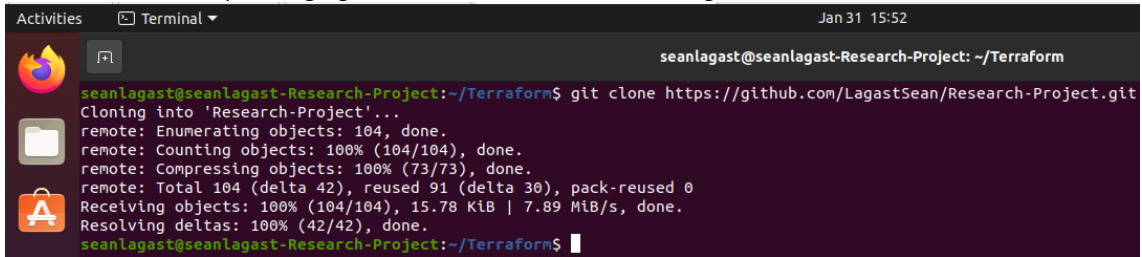
Aangezien mijn 'applicatie' op git staat, moeten we deze kunnen afhalen, daar hebben we eerst de git package voor nodig, deze installeren we als volgt.

A terminal window titled 'Terminal' with a dark background. The user 'seanlagast@seanlagast-Research-Project' is at the prompt in the directory ~/Terraform. The terminal shows the command 'sudo apt install git -y'. The output shows the package lists being read, the dependency tree being built, and state information being read. It then lists additional packages to be installed: git-man and liberror-perl. Suggested packages include git-daemon-run, git-daemon-sysvinit, git-doc, git-el, git-email, git-gui, gitk, and gitweb.

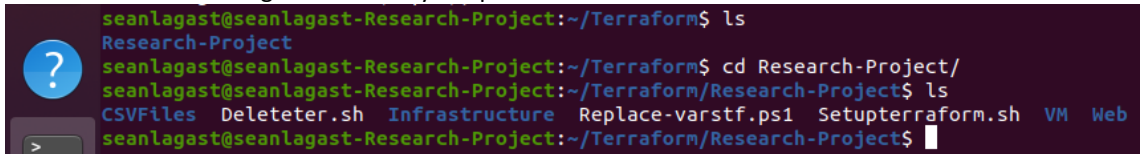
```
seanlagast@seanlagast-Research-Project:~/Terraform$ sudo apt install git -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
The following NEW packages will be installed:
```

Stap 4: Repo clonen

Nu kunnen we de repo van git gaan clonen. Doe hiervoor het volgende.

A terminal window titled 'Terminal' with a date of 'Jan 31 15:52'. The prompt is 'seanlagast@seanlagast-Research-Project: ~/Terraform'. The command 'git clone https://github.com/LagastSean/Research-Project.git' is entered. The output shows the cloning process: 'Cloning into 'Research-Project'...', 'remote: Enumerating objects: 104, done.', 'remote: Counting objects: 100% (104/104), done.', 'remote: Compressing objects: 100% (73/73), done.', 'remote: Total 104 (delta 42), reused 91 (delta 30), pack-reused 0', 'Receiving objects: 100% (104/104), 15.78 KiB | 7.89 MiB/s, done.', 'Resolving deltas: 100% (42/42), done.', and the prompt returns to 'seanlagast@seanlagast-Research-Project: ~/Terraform\$'.

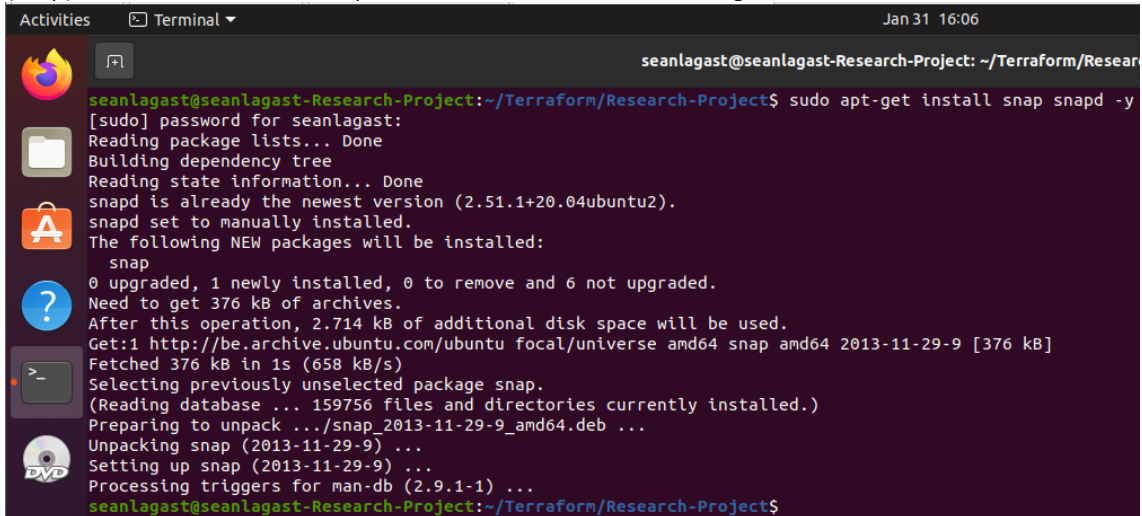
Nu kunnen we in de volgende directory kruipen.

A terminal window titled 'Terminal' with a date of 'Jan 31 15:52'. The prompt is 'seanlagast@seanlagast-Research-Project: ~/Terraform'. The command 'ls' is entered, showing 'Research-Project'. Then 'cd Research-Project/' is entered. Then 'ls' is entered again, showing 'CSVFiles', 'Deleteter.sh', 'Infrastructure', 'Replace-varstf.ps1', 'Setupterraform.sh', 'VM', and 'Web'. The prompt returns to 'seanlagast@seanlagast-Research-Project: ~/Terraform/Research-Project\$'.

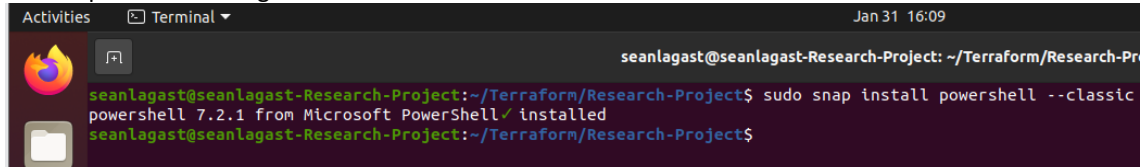
Stap 5: Powershell installeren

Normaal gesproken als je in een linux omgeving werkt kun je alles makkelijk uitschrijven in bash script, maar ik koos ervoor om een mix te maken en ook powershell te gaan integreren in dit project. Dit zal dienen om makkelijk csv files te gaan uitlezen en de benodigde info in de csv files te gaan toepassen op de correcte variabelen die terraform nodig heeft. Je kunt natuurlijk dit concept op veel verschillende manieren gaan toepassen.

Om powershell te kunnen gebruiken op deze ubuntu moet er eerst een tussenstap gebeuren, namelijk (snap) installeren, anders zal de powershell module standaard niet gevonden worden.

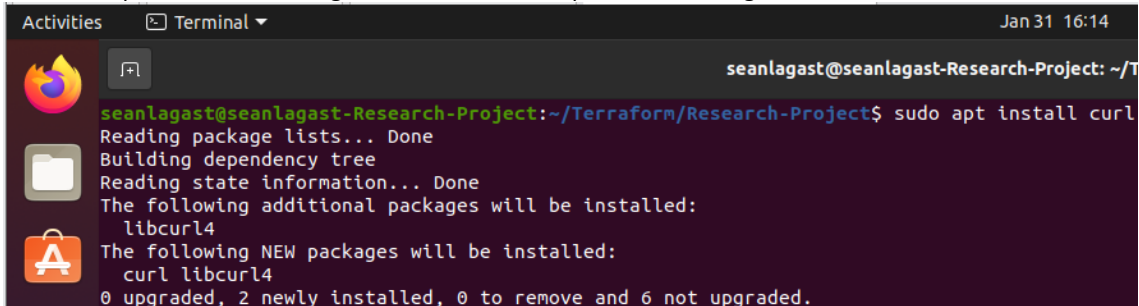
A terminal window titled 'Terminal' with a date of 'Jan 31 16:06'. The prompt is 'seanlagast@seanlagast-Research-Project: ~/Terraform/Research-Project'. The command 'sudo apt-get install snap snapd -y' is entered. The output shows the installation process: '[sudo] password for seanlagast:', 'Reading package lists... Done', 'Building dependency tree', 'Reading state information... Done', 'snapd is already the newest version (2.51.1+20.04ubuntu2).', 'snapd set to manually installed.', 'The following NEW packages will be installed:', 'snap', '0 upgraded, 1 newly installed, 0 to remove and 6 not upgraded.', 'Need to get 376 kB of archives.', 'After this operation, 2.714 kB of additional disk space will be used.', 'Get:1 http://be.archive.ubuntu.com/ubuntu focal/universe amd64 snap amd64 2013-11-29-9 [376 kB]', 'Fetched 376 kB in 1s (658 kB/s)', 'Selecting previously unselected package snap.', '(Reading database ... 159756 files and directories currently installed.)', 'Preparing to unpack .../snap_2013-11-29-9_amd64.deb ...', 'Unpacking snap (2013-11-29-9) ...', 'Setting up snap (2013-11-29-9) ...', 'Processing triggers for man-db (2.9.1-1) ...', and the prompt returns to 'seanlagast@seanlagast-Research-Project: ~/Terraform/Research-Project\$'.

Nu kan powershell zelf geïnstalleerd worden.

A terminal window titled 'Terminal' with a date of 'Jan 31 16:09'. The prompt is 'seanlagast@seanlagast-Research-Project: ~/Terraform/Research-Project'. The command 'sudo snap install powershell --classic' is entered. The output shows 'powershell 7.2.1 from Microsoft PowerShell✓ installed' and the prompt returns to 'seanlagast@seanlagast-Research-Project: ~/Terraform/Research-Project\$'.

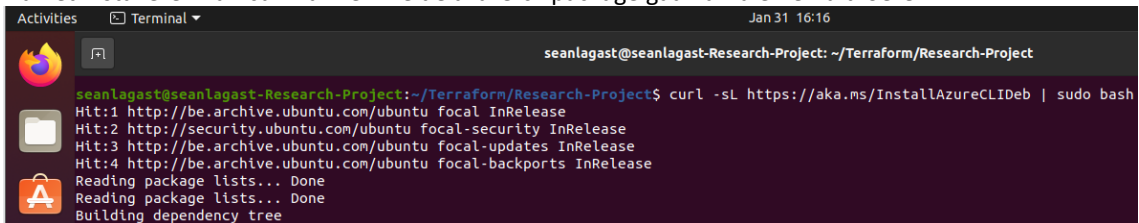
Stap 6: Azure cli installeren

Om te kunnen verbinden met azure via de ubuntu machine hebben we eerst de azure-cli nodig. Om deze te kunnen downloaden moeten we eerst de curl module installeren. Curl zorgt ervoor dat je data van een bepaalde server kunt gaan downloaden of uploaden, in ons geval downloaden.



```
seanlagast@seanlagast-Research-Project: ~/Terraform/Research-Project$ sudo apt install curl
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl libcurl4
0 upgraded, 2 newly installed, 0 to remove and 6 not upgraded.
```

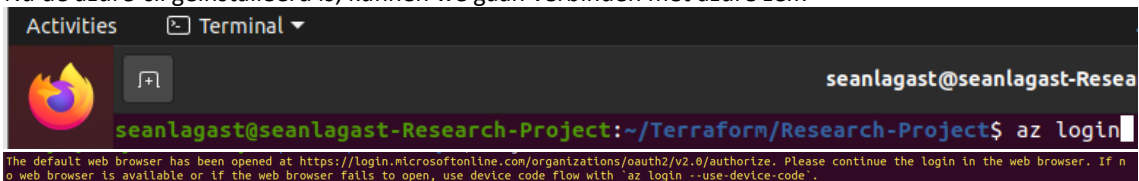
Na het installeren van curl kunnen we de azure-cli package gaan afhalen en uitvoeren.



```
seanlagast@seanlagast-Research-Project: ~/Terraform/Research-Project$ curl -sL https://aka.ms/InstallAzureCLIDeb | sudo bash
Hit:1 http://be.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:3 http://be.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://be.archive.ubuntu.com/ubuntu focal-backports InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree
```

Stap 7: Inloggen op azure

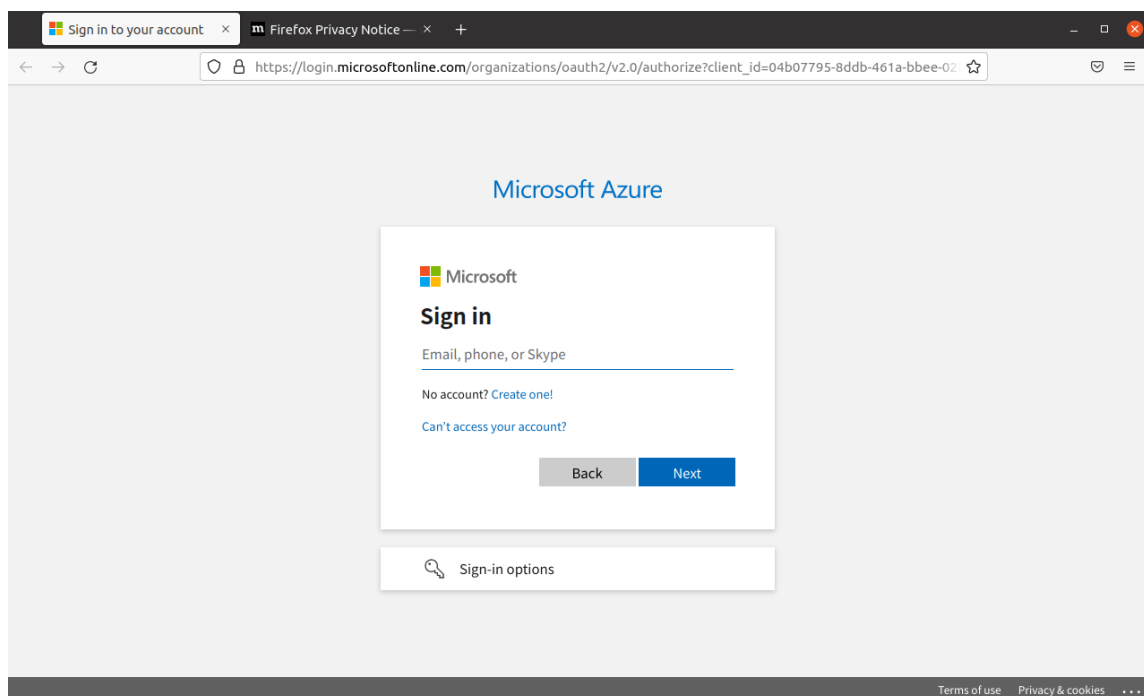
Nu de azure-cli geïnstalleerd is, kunnen we gaan verbinden met azure zelf.



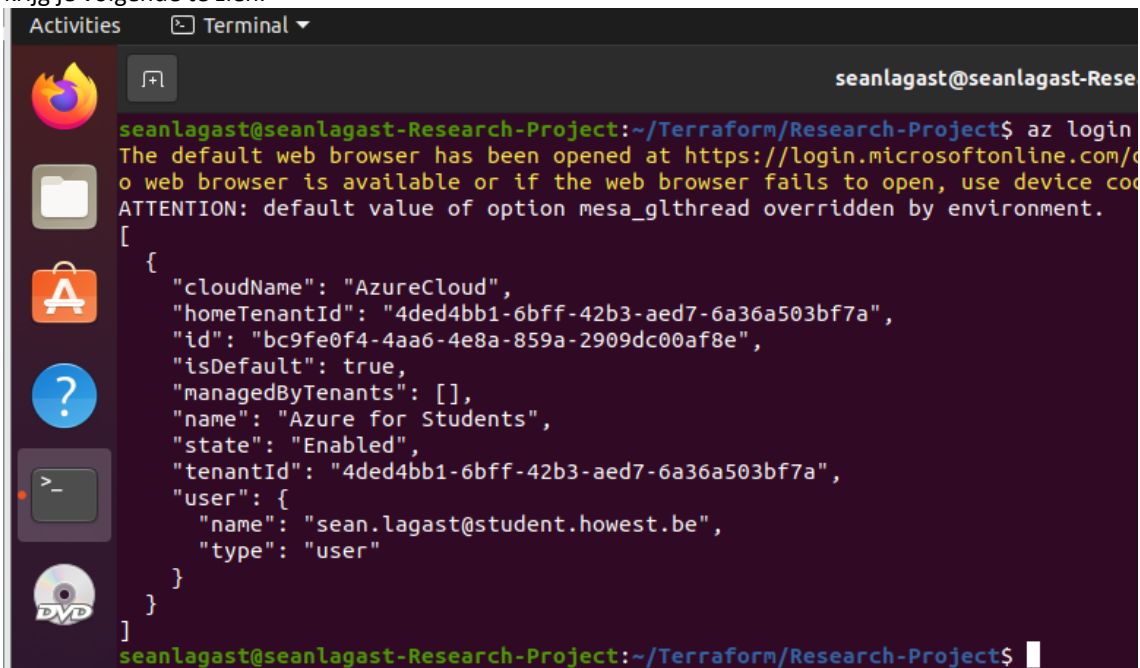
```
seanlagast@seanlagast-Research-Project: ~/Terraform/Research-Project$ az login
```

The default web browser has been opened at https://login.microsoftonline.com/organizations/oauth2/v2.0/authorize. Please continue the login in the web browser. If no web browser is available or if the web browser fails to open, use device code flow with 'az login --use-device-code'.

Normaal gezien indien je dezelfde setup hebt als mij wordt automatisch een login prompt weergegeven. Indien niet kun je naar de vermelde link surfen en de code die je gekregen hebt gebruiken.



Hier log je in met een admin account(account met rechten om resources te maken). Na het inloggen krijg je volgende te zien.

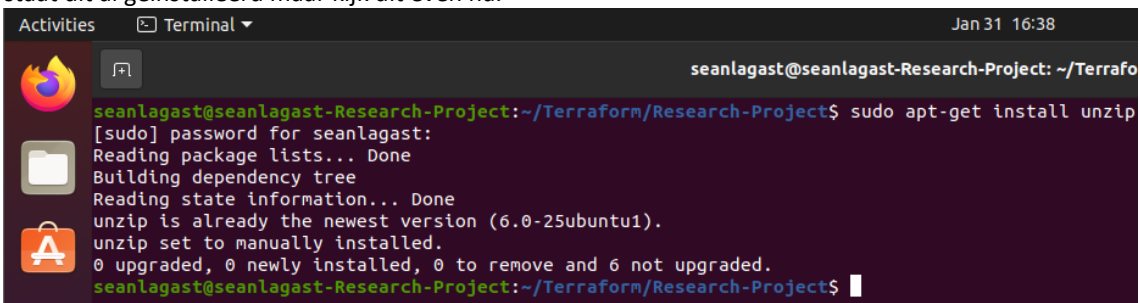


```
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$ az login
The default web browser has been opened at https://login.microsoftonline.com/...
o web browser is available or if the web browser fails to open, use device code
ATTENTION: default value of option mesa_glxthread overridden by environment.
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "4ded4bb1-6bff-42b3-aed7-6a36a503bf7a",
    "id": "bc9fe0f4-4aa6-4e8a-859a-2909dc00af8e",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Azure for Students",
    "state": "Enabled",
    "tenantId": "4ded4bb1-6bff-42b3-aed7-6a36a503bf7a",
    "user": {
      "name": "sean.lagast@student.howest.be",
      "type": "user"
    }
  }
]
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$
```

Nu ben je ingelogd op de azure-cli, later zal dit ervoor zorgen dat de terraform scripts uitgevoerd worden op de door jouw geselecteerde omgeving. Om terug uit te loggen op deze omgeving typ je i.p.v. az login -> az logout. Wees wel zeker dat er geen aanpassingen meer moeten gebeuren aan je omgeving nadat je uitlogt.

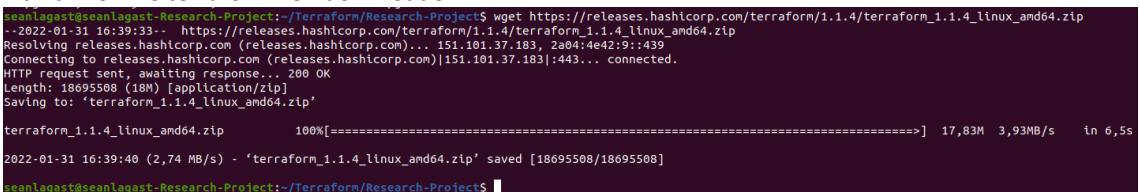
Stap 8: Terraform installeren

Terraform installeren we via een zip die we downloaden, daarom installeren we unzip. Normaal gezien staat dit al geïnstalleerd maar kijk dit even na.



```
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$ sudo apt-get install unzip
[sudo] password for seanlagast:
Reading package lists... Done
Building dependency tree
Reading state information... Done
unzip is already the newest version (6.0-25ubuntu1).
unzip set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 6 not upgraded.
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$
```

Nu kunnen we terraform zelf downloaden.



```
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$ wget https://releases.hashicorp.com/terraform/1.1.4/terraform_1.1.4_linux_amd64.zip
--2022-01-31 16:39:33-- https://releases.hashicorp.com/terraform/1.1.4/terraform_1.1.4_linux_amd64.zip
Resolving releases.hashicorp.com (releases.hashicorp.com)... 151.101.37.183, 2a04:4e42:9::439
Connecting to releases.hashicorp.com (releases.hashicorp.com)[151.101.37.183]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18695508 (18M) [application/zip]
Saving to: 'terraform_1.1.4_linux_amd64.zip'

terraform_1.1.4_linux_amd64.zip 100%[=====] 17,83M 3,93MB/s in 6,5s

2022-01-31 16:39:40 (2,74 MB/s) - 'terraform_1.1.4_linux_amd64.zip' saved [18695508/18695508]
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$
```

Let op dat je de gewenste versie van terraform installeert. Dit kun je aanpassen door de versie in de link te veranderen, bijvoorbeeld:

https://releases.hashicorp.com/terraform/1.1.4/terraform_1.1.4_linux_amd64.zip ->

https://releases.hashicorp.com/terraform/1.1.5/terraform_1.1.5_linux_amd64.zip

Indien je de 1.1.5 versie wil gebruiken, (versie bestaat nog niet op datum van aanmaak document).

Nu we de zip hebben kunnen we deze gaan uitpakken en op de juiste plaats zetten.

```
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$ unzip terraform_1.1.4_linux_amd64.zip
Archive:  terraform_1.1.4_linux_amd64.zip
  inflating: terraform
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$
```

De zip kan terug verwijderd worden.

```
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$ sudo rm terraform_1.1.4_linux_amd64.zip
```

Ten slotte moet de map op de juiste locatie gezet worden.

```
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$ sudo mv terraform /usr/local/bin/
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$
```

Stap 9: Crontab instellen

Dit project zorgt dus voor de eenvoudige aanmaak van omgevingen, maar zorgt ook dat deze omgevingen terug afgebroken worden. Hier komt crontab aan te pas. Terraform heeft nog geen ingebouwde functie die een omgeving een x-aantal tijd laat bestaan en dan terug verwijderd. Crontab zal ervoor zorgen dat het Deleteter.sh script uitgevoerd wordt, wat op zijn beurt de correcte omgeving terug afbreekt.

Eerst moeten we zeker zijn dat deze service aanstaat.

```
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$ sudo service cron status
cron.service - Regular background program processing daemon
Loaded: loaded (/lib/systemd/system/cron.service; enabled; vendor preset: enabled)
Active: active (running) since Mon 2022-01-31 15:39:16 CET; 1h 14min ago
Docs: man:cron(8)
Main PID: 739 (cron)
Tasks: 1 (limit: 4593)
Memory: 500.0K
CGroup: /system.slice/cron.service
└─739 /usr/sbin/cron -f

Jan 31 15:39:16 seanlagast-Research-Project cron[739]: (CRON) INFO (pidfile fd = 3)
Jan 31 15:39:16 seanlagast-Research-Project cron[739]: (CRON) INFO (Running @reboot jobs)
Jan 31 16:17:01 seanlagast-Research-Project CRON[37832]: pam_unix(cron:session): session opened for user root by (uid=0)
Jan 31 16:17:01 seanlagast-Research-Project CRON[37833]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)
Jan 31 16:17:01 seanlagast-Research-Project CRON[37832]: pam_unix(cron:session): session closed for user root
Jan 31 16:30:01 seanlagast-Research-Project CRON[38357]: pam_unix(cron:session): session opened for user root by (uid=0)
Jan 31 16:30:01 seanlagast-Research-Project CRON[38357]: (root) CMD ( cd / && run-parts --report /etc/cron.daily)
Jan 31 16:30:01 seanlagast-Research-Project CRON[38357]: pam_unix(cron:session): session closed for user root
Jan 31 16:43:01 seanlagast-Research-Project CRON[38501]: pam_unix(cron:session): session opened for user root by (uid=0)
Jan 31 16:43:01 seanlagast-Research-Project CRON[38502]: (root) CMD ( test -x /etc/cron.daily/popularity-contest && /etc/cron.daily/popularity-contest --cron)
Jan 31 16:43:01 seanlagast-Research-Project CRON[38501]: pam_unix(cron:session): session closed for user root
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$
```

Indien dit niet zo is, zet deze dan zeker eerst aan.

```
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$ sudo service cron restart
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$
```

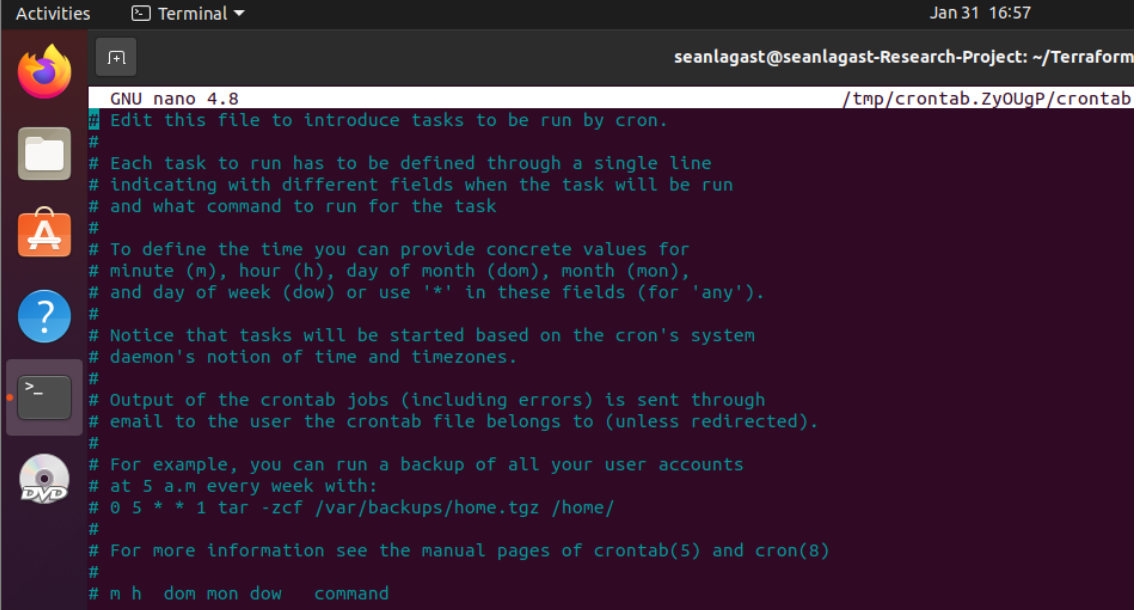
Ook moeten we eerst eens een initiatie doen van het crontab menu.

```
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$ crontab -e
no crontab for seanlagast - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.tiny
 3. /bin/ed

Choose 1-3 [1]: 1
```


Na deze stap krijg je volgende te zien.

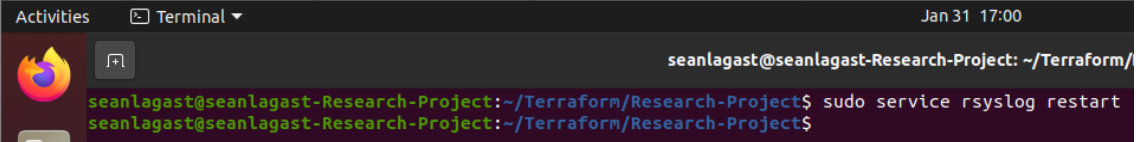


```
seanlagast@seanlagast-Research-Project: ~/Terraform
GNU nano 4.8 /tmp/crontab.Zy0UgP/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
```

Hier mag je terug uit gaan met de shortcut ctrl+x.

Stap 10: Systeem logging starten

Om later te kunnen zien of cron zijn job goed uitvoert moet logging aanstaan.



```
seanlagast@seanlagast-Research-Project: ~/Terraform/Research-Project$ sudo service rsyslog restart
seanlagast@seanlagast-Research-Project: ~/Terraform/Research-Project$
```

Maak nu ook een aparte log file aan, je kunt deze plaats en naam zelf kiezen, maar dan zul je een kleine aanpassing moeten in het Setup script, voor deze uitleg gebruik ik het volgende.

```
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$ sudo touch /var/log/backup.log
```

Stap 11: Permissies op scripts instellen

Omdat sommige scripts en logging niet op user niveau gebeurt moet je een paar permissies voor volgende zaken aanpassen. Cron moet kunnen schrijven naar de log file daarom dat 6 gebruikt, dit zorgt dat cron ook kan schrijven naar deze file.

```
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$ sudo chmod 756 /var/log/backup.log
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$ sudo chmod +x Replace-varstf.ps1
seanlagast@seanlagast-Research-Project:~/Terraform/Research-Project$ sudo chmod +x Deleteter.sh
```

Einde Installatie

De installatie van het project is nu compleet. De gebruikershandleiding kan nu geraadpleegd worden.

howest
hogeschool