**h2 Karjaa**

Prosessori: Intel i7-11700K LGA1151(tarkasta?)

Muisti: 32Gb ddr 4 MHz?

SSD: 2tb samsung 980 Pro NVMe

Näytönohjain: 3080RTX Mobile 16Gb

1 Näyttö: Clevo 17” 4k 60Hz

2 Näyttö: Acer Predator 43” 4k 120Hz

3 Näyttö: Acer Predator 27” FHD 144Hz

*Cattle, not pets.*

x) Lue ja tiivistä. (Tässä x-alakohdassa ei tarvitse tehdä testejä tietokoneella, vain lukeminen tai kuunteleminen ja tiivistelmä riittää. Tiivistämiseen riittää muutama ranskalainen viiva.)

Slater 2017: [What is the definition of "cattle not pets"?](https://devops.stackexchange.com/questions/653/what-is-the-definition-of-cattle-not-pets#654). (Vain tuo yksi vastaus DevOps Stack Exchangen kysymykseen)

* Ennen serverit olivat kuin lemmikkejä, jos serveri kaatui, piti ryhtyä hommiin, että se saadaan takaisin ylös. Nykyään servereitä on useampia ja toinen ottaa roolin, jos yksi kaatuu.
* Ennen serverit olivat uniikkeja järjestelmiä, joita ihminen ohjasi. Nykyään ne ovat automatisoituja, että ihmisten tekemisiä ei välttämättä tarvita, jos serveri kaatuu vaan automaatio hoitaa homman ja serveri palaa linjoille tai se tuhotaan. Tämä ei kyllä ole optimia, se kannattaa mieluummin ”jäädyttää” ja tutkia perimmäinen syy miksi serveri ylipäätään kaatui.

Karvinen 2017: [Vagrant Revisited – Install & Boot New Virtual Machine in 31 seconds](https://terokarvinen.com/2017/04/11/vagrant-revisited-install-boot-new-virtual-machine-in-31-seconds/) (Suosittelen koneeksi 'vagrant init debian/bullseye64')

“sudo apt-get update”

“sudo apt-get -y install vagrant”

“vagrant init bento/ubuntu-16.04”

“vagrant up”

“vagrant ssh”

“vagrant destroy”

Karvinen 2023: [Salt Vagrant - automatically provision one master and two slaves](https://terokarvinen.com/2023/salt-vagrant/)  
  
-Artikkeli käsittellee Vagrantin asennusta yhdellä masterilla ja kahdella orjalla

-Artikkeli on tehty raudalla pyörivää linux asennusta varten, ei sovellu windowsille suoraan

-Artikkelia voi soveltaa Vagrantin asennukseen Windowsille.

1. Asenna Vagrant. (Toiminee parhaiten isäntäkäyttöjärjestelmässä, siis siinä, joka pyörii raudalla)

Vagrantin asennus windowsille Vmwaren kanssa osoittautui jokseenkin haastavaksi, ohjeita ei ollut paljon. Jo pitkään ollut tämä mielessä, joten päätin asentaa Linuxin toiseksi käyttöjärjestelmäkseni pääkoneeseen Windowsin rinnalle. Valitsin Linuxiksi Debian 12.

Asensin Rufuksen Windowsille ja loin bootaavan usb tikun Debianilla

Kirjoitan seuraavan osion ulkomuistista.

Debianin asennus onnistui mutta tuli grafiikka bugeja, en aluksi päässy kirjautumaan roottina, mutta komento ”su – root” onnistui sen tekemään. Koitin päivittää näytönohjaimen ajureita, se osoittautui hankalaksi. Näytön ohjain on Nvidia 3080 RTX mobile paketinhallinnasta niitä ei löytynyt. Nvidian sivuilta sain .run tiedoston, jonka muokkasin ”chmod u+x tiedostonimi” suoritetavaksi tiedostoksi kaikki tämä kun näyttö pomppi miten sattuu ja oli hankalaa lukea. komento ./nvidia-drivers-xxx-xxx.run. Installeri varoitti, että X-server pyörii ja se pitää lopettaa. Googlasin, mutta ohjeet eivät auttaneet, tietokone oli ohjeiden jälkeen aina käynnistettävä reset napilla uudestaan. Ohjeita oli esim. CTR+ALT+F1 tai ”sudo service gdm stop” Kokeilin useita muitakin metodeja ratkaista ongelma, asensin jopa kahteen kertaan koko Debianin ja lopuksi kokeilin vielä Ubuntua asentaa, mutta se tökkäsi jo asennus vaiheessa mustaan ruuttuun. Tähän kaikkeen meni aikaa yli 6 tuntia

Päätin palata asentamaan Vagrantin Windowsille, koska Linuxin asentaminen raudalleni tuntuu olevan todella haastava homma. Tällä hetkellä Grubissa on pari debiania asennettuna. Mitenhän sen siivoaisi? 😊

Googlasin ohjeet vagrantin asennukseen Windows 11 ja Vmawarelle, löysin Github sivun: [Install vagrant with vmware for windows 11 (github.com)](https://gist.github.com/1eedaegon/6a5a2ab64f09b0800eadf3f84dad96f8)

Latasin ja asensin Gon (ohjelmointikieli), uudelleen asennus vaadittin

Skippasin kohdan 2, koska minulla oli jo Vmware asennettuna.

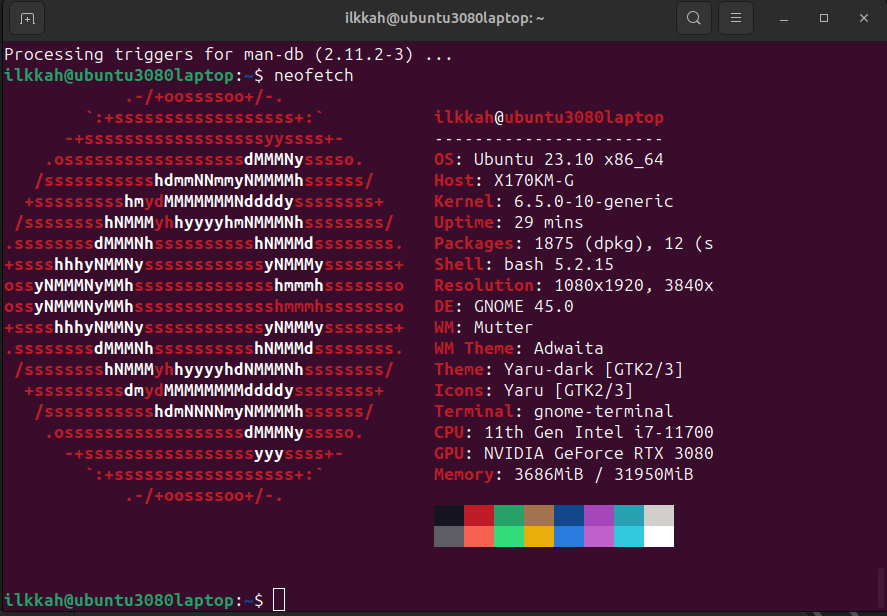
Latasin ja asensin Vagrantin VMware Utilityn.

Aloin asentamaan pluginia Vmwarelle, näiden ohjeiden mukaan: [Introducing the Vagrant VMware Desktop Plugin (hashicorp.com)](https://www.hashicorp.com/blog/introducing-the-vagrant-vmware-desktop-plugin)

Pyysin lisää aikaa tehtävien tekoon ja jatkan tästä..

Jatkan 06.11.2023

Selvitin tuo Vagrant VMware Desktop Plugin on maksullinen, koitan vielä asentaa Ubuntun tuoreimman ei LTS version koneelleni.

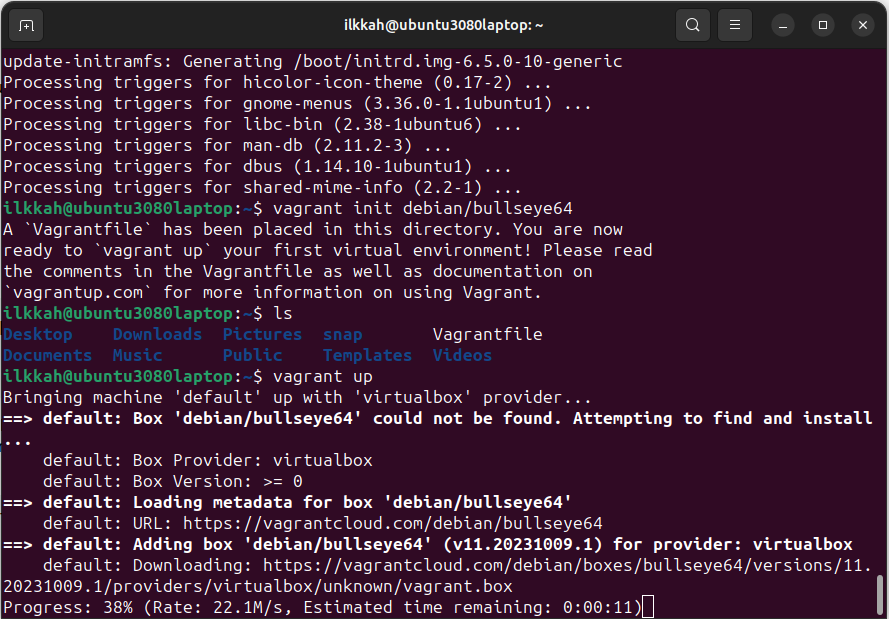


Sain Ubuntun version 23.10 ilman ongelmia pyörii nyt raudan päällä siis. Nyt tehtävien pariin.

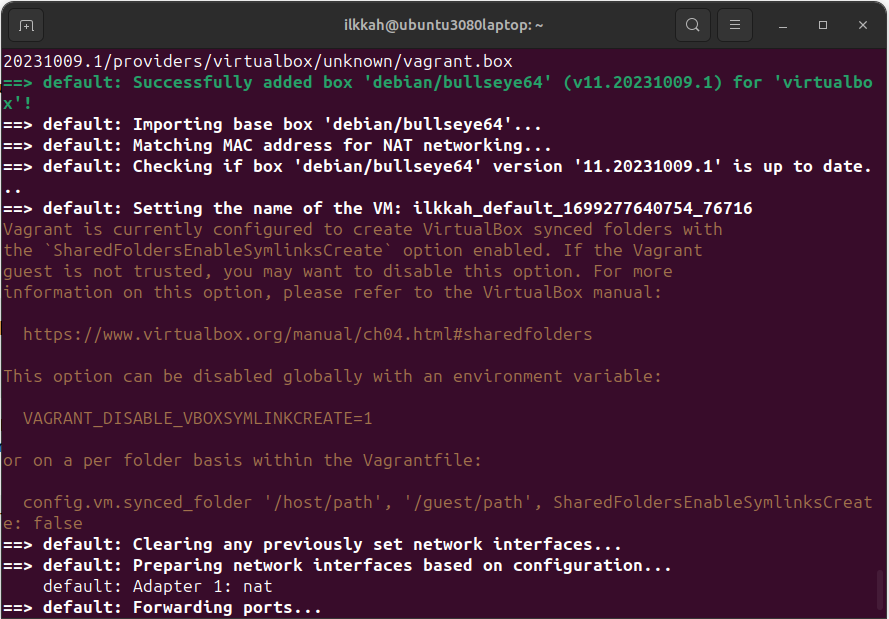
1. Yksi maankiertäjä. Asenna yksi kone Vagrantilla, ota siihen SSH-yhteys, osoita että netti toimii.

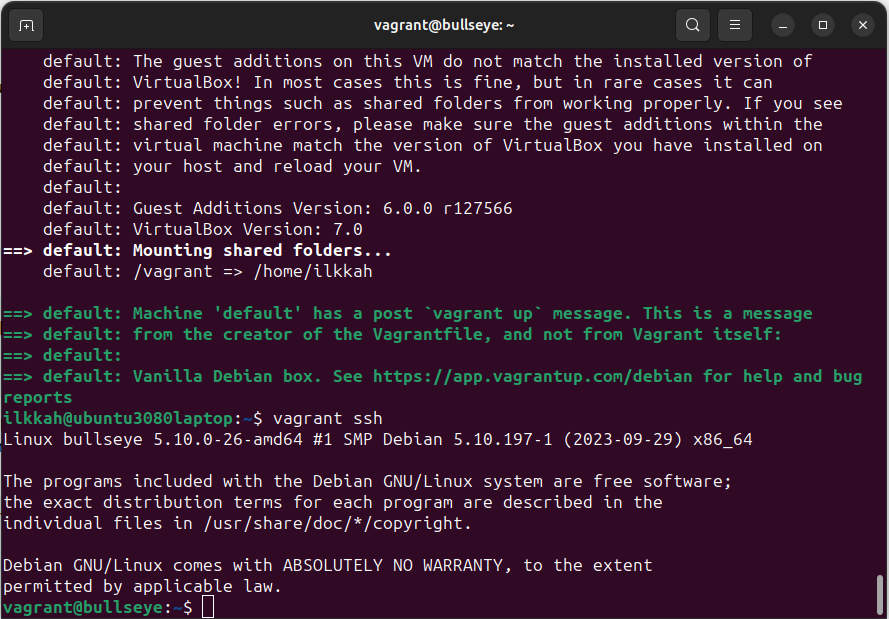
Komento ”sudo apt-get install vagrant virtualbox” komento asensi vagrantin ja virtualboxin

komento alustaa Debianin version bullseye64 toimimana vagrantissa ja luo Vagranfile tiedoston, tiedostolla voi muokata asetuksia ympäristössä ”vagrant init debian/bullseye64”



Debian/bullseye64 ei löytynyt aluksi, joten vagrant latasi sen.

Synkronoidut kansiot ovat päällä, tämä pitää kyllä opetella laittamaan pois päältä. Komentohan tuossa onkin jo annettu valmiiksi, kun muuttaa ilmeisesti vagrantfilessä olevia tietoja.

Komento ”vagrant ssh” ja siellä pyörii debian ja olen kirjautuneena sisään

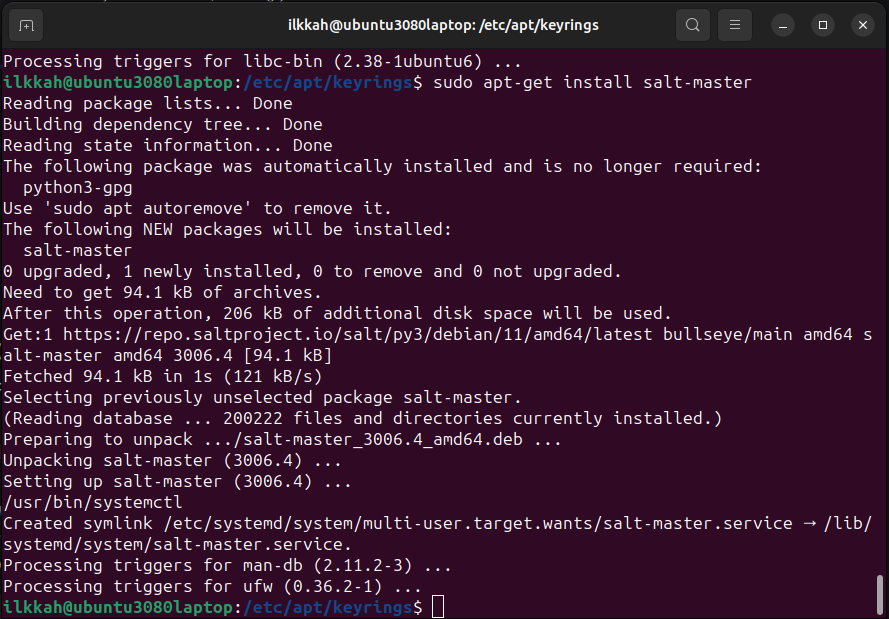
1. Oma orjansa. Asenna Salt herra ja orja samalle koneelle.

Tehtävä pitäisi olla paremmin avattu. Eli vagrant koneelle vai pääkoneelle.

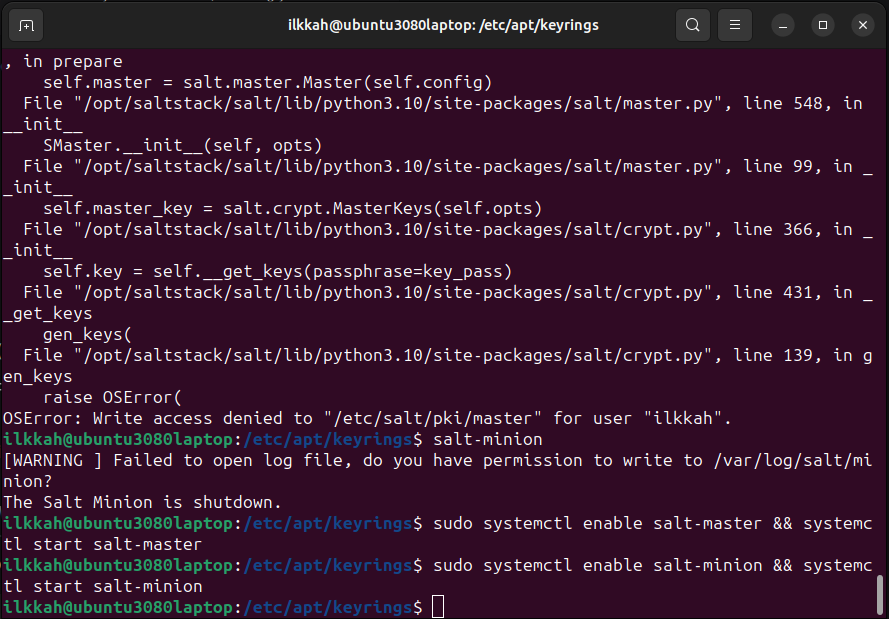
Asennan pääkoneelle. ”sudo apt-get update”, ”sudo apt-get install salt-minion”

eipä asennukkaan myöskään ubuntuun. Lisäsin paketinhallintaan

Näiden ohjeiden mukaisesti: <https://docs.saltproject.io/salt/install-guide/en/latest/topics/install-by-operating-system/debian.html#install-salt-on-debian-11-bullseye-amd64>



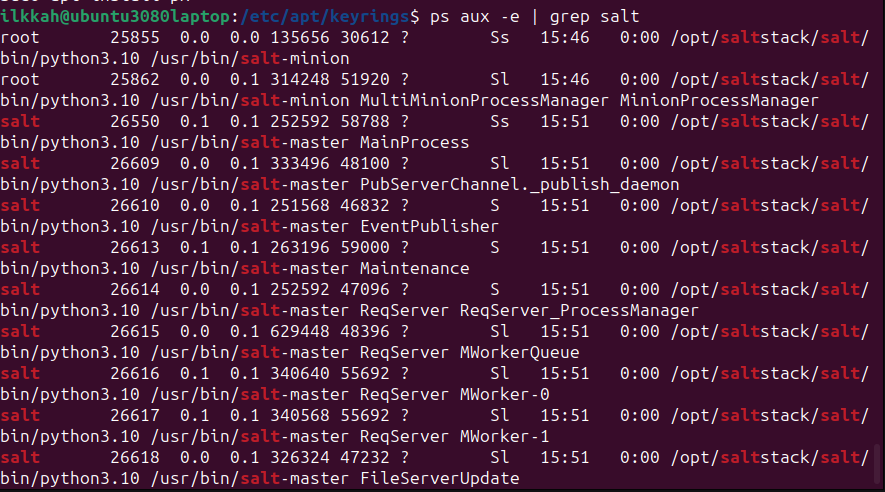
sitten komento uudestaan ”sudo apt-get update” ja sudo apt-get install salt-minion” ja vielä komento ”sudo apt-get install salt-master”, kummatkin asentuivat hienosti



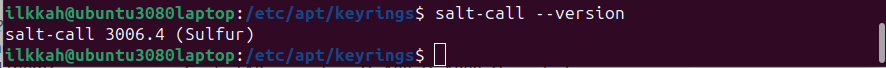
Yritin käynnistää salt masterin ja minionin, eipä lähteneetkään käyntiin.

Koitin ohjeiden mukaan, vaikka ne ovatkin debianille, mutta käsittääkseni Ubintu on debian pohjainen distro, joten ne saattavat toimia.

Laitoin komennon ”sudo systemctl enable salt-master && systemctl start salt-master” ja ukomennon ”sudo systemctl enable salt-minion && systemctl start salt-minion”



komento ”ps aux -e | grep salt” löytyi prosessit joita salt-minion ja salt-master käytttä. Se siis oli nyt asennettu koneelle.



Tarkastin vielä saltin version komennolla ” salt-call --version”

1. Asenna Saltin herra-orja arkkitehtuuri toimimaan verkon yli. (Verkko voi olla virtuaalinen verkko paikallisten virtuaalikoneiden välillä, kuten muissakin alakohdissa)

”vagrant up”

”vagrant ssh”

”sudo apt-get update”

”sudo apt-get install curl”

”sudo mkdir /etc/apt/keyrings”

”sudo curl -fsSL -o /etc/apt/keyrings/salt-archive-keyring-2023.gpg <https://repo.saltproject.io/salt/py3/debian/11/amd64/SALT-PROJECT-GPG-PUBKEY-2023.gpg>”

”echo "deb [signed-by=/etc/apt/keyrings/salt-archive-keyring-2023.gpg arch=amd64] https://repo.saltproject.io/salt/py3/debian/11/amd64/latest bullseye main" | sudo tee /etc/apt/sources.list.d/salt.list”

”sudo apt-get update”

”sudo apt-get install salt-minion”

”logout”

komento ”sudo hostname I” tällä sai masterin ip:n selville



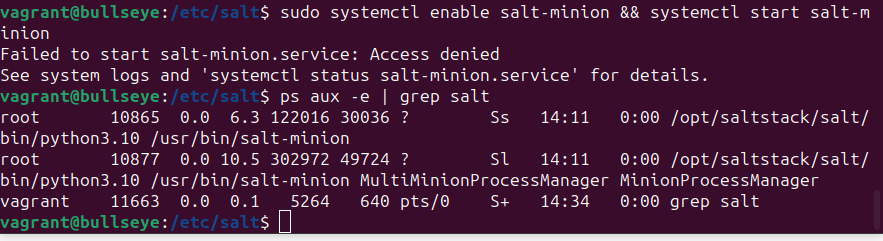
komento ”nc -vz 192.168.178.70 4505” portiin kytekeytyminen tcp yhteydellä onnistui eli portti on ylipäättään auki.

”vagrant ssh”

”sudo apt-get install micro”

”sudo micro etc/salt/minion”

lisätään tiedostoon master: 192.168.178.70

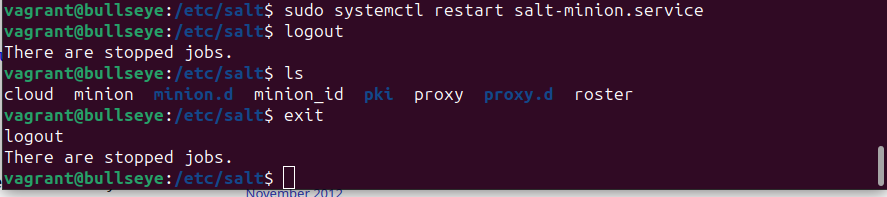


”sudo systemctl enable salt-minion && systemctl start salt-minion”

eipä antanut vagrant virtuaalikone käynnistää salt-minionia

”ps aux -e | grep salt”

kyllä se siellä pyöri kuitenkin



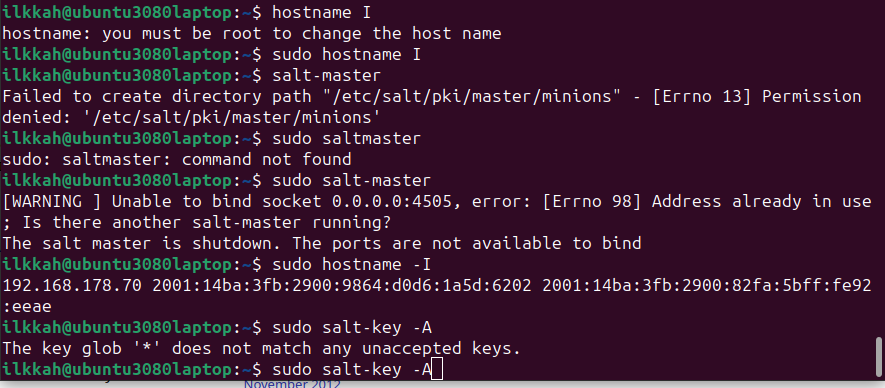
kokeilin käynnistää salt orjan uudestaan näiden ohjeiden mukaan: <https://terokarvinen.com/2018/salt-quickstart-salt-stack-master-and-slave-on-ubuntu-linux/?fromSearch=salt%20master>

”sudo systemctl restart salt-minion.service”

sitten ”logout” mutta vagrant kone sanoo, että There are stopped jobs. Mitköhän nyt sitten? :)

Googlasin ja ohjeilla <https://unix.stackexchange.com/questions/116959/there-are-stopped-jobs-on-bash-exit>

uudestaan vaan komento ”exit”



”sudo hostname I” ei antanut mitään tulosta

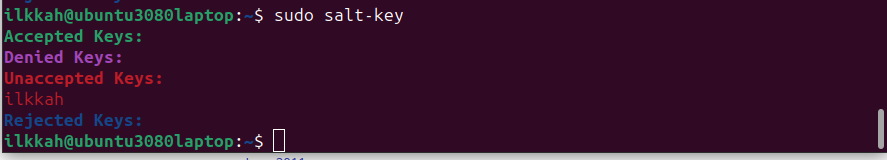
”salt-master” väärin uudelleen käynnistetty salt-master

”sudo salt-master” taas väärin uudelleen käynnistetty salt-master” virhe ilmoitus, koska se pyörii jo. Pitäisi käynnistää udeelleen systemctl avulla

”sudo hostname -I” kuitenkin antaa nyt IP osoitteen

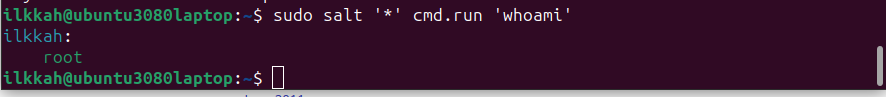
”sudo salt-key -A” The key glob ’\*’ does not match any unaccepted keys. En kyllä tajua missä vika

Kävin katsomassa vielä etc/salt/minion tiedostoa ja ip osoitteessa oli kirjoitusvirhe. Korjasin sen ja tadaa.



Komento ”sudo salt-key” paljasti, että orjaoli ilmoittautunut.

Komento ”sudo salt-ket -A” hyväksyin vaimen.



Ja testaus, että orja vastaa ja toimi.

1. Aja useita idempotentteja (state.single) komentoja verkon yli.
2. Kerää teknistä tietoa orjista verkon yli (grains.item)
3. Aja shell-komento orjalla verkon yli.
4. Hello, IaC. Tee infraa koodina kirjoittamalla /srv/salt/hello/init.sls. Aja tila jollekin orjalle. Tila voi esimerkiksi tehdä esimerkkitiedoston johonkin hakemistoon. Testaa toisella komennolla, että pyytämäsi muutos on todella tehty.

h4 Demonit

x) Lue ja tiivistä. (Tässä x-alakohdassa ei tarvitse tehdä testejä tietokoneella, vain lukeminen tai kuunteleminen ja tiivistelmä riittää. Tiivistämiseen riittää muutama ranskalainen viiva.)

Karvinen 2023: Salt Vagrant - automatically provision one master and two slaves, vain kohdat

Infra as Code - Your wishes as a text file

top.sls - What Slave Runs What States

Salt contributors: Salt overview, kohdat

Rules of YAML

YAML simple structure

Lists and dictionaries - YAML block structures

Salt contributors: Salt states, kohdat

State modules

The state SLS data structure

Organizing states

The top.sls file

Create the SSH state, Create the Apache state

Katso näitä alakohtia kriittisin silmin. Teron huomioita ja suosituksia:

Namevar: itse asia tilan id:ksi, name arvoksi tulee automaattisesti sama

Puuttuko esimerkeistä service-watch? Jos asetustiedosto päivittyy herralla, tuleeko se koskaan käyttöön orjilla?

Karvinen 2018: Pkg-File-Service – Control Daemons with Salt – Change SSH Server Port

Käytä omaa sshd\_config:ia pohjana. Tässä on jonkun toisen Linux-version tiedosto.

a) Hello SLS! Tee Hei maailma -tila kirjoittamalla se tekstitiedostoon, esim /srv/salt/hello/init.sls.

b) Top. Tee top.sls niin, että tilat ajetaan automaattisesti, esim komennolla "sudo salt '\*' state.apply".

c) Apache. Asenna Apache, korvaa sen testisivu ja varmista, että demoni käynnistyy.

Ensin käsin, vasta sitten automaattisesti.

Kirjoita tila sls-tiedostoon.

pkg-file-service

Tässä ei tarvita service:ssä watch, koska index.html ei ole asetustiedosto

d) SSHouto. Lisää uusi portti, jossa SSHd kuuntelee.

Tämä tehtävä on helpointa tehdä tavallisella virtuaalikoneella, jota Vagrant ei ohjaa.

Löydät oikean asetuksen katsomalla SSH:n asetustiedostoa

Nyt tarvitaan service-watch, jotta demoni käynnistetään uudelleen, jos asetustiedosto muuttuu masterilla

Vinkit

Ensin käsin, sitten automaattisesti

Testaa

Alkutilanne (taikurin hihat tyhjät)

Käsin tehty ja toimii

Poistettu käsin tehty ennen automaatiota

Yksi tilafunktio (esim. file) sls-tiedostossa

Lopputilanne, osat

Lopputesti - mitä käyttäjä tekisi

/etc/ssh/sshd\_config

Port: 1234

nc -vz localhost 1234

ssh -p 1234 foo@localhost

echo "Hei"|sudo tee /var/www/html/index.html # ainoa tilanne sudotella html-sivua

Omat asetustiedostot

Kun teet käsin, saat siitä mallin asetustiedostolle (masterin /srv/salt/foo/bar.cfg)

Esimerkkidokumenteissa olevat demonien asetustiedostot tuskin toimivat juuri omassa järjestelmässä

**h4 Demonit**

x) Lue ja tiivistä. (Tässä x-alakohdassa ei tarvitse tehdä testejä tietokoneella, vain lukeminen tai kuunteleminen ja tiivistelmä riittää. Tiivistämiseen riittää muutama ranskalainen viiva.)

* Karvinen 2023: [Salt Vagrant - automatically provision one master and two slaves](https://terokarvinen.com/2023/salt-vagrant/#infra-as-code---your-wishes-as-a-text-file), vain kohdat
  + Infra as Code - Your wishes as a text file
  + top.sls - What Slave Runs What States
* sudo mkdir -p /srv/salt/hello
* sudoedit /srv/salt/hello/init.sls

Sisennys on kaksi välilyöntiä YAML syntaksissa

“cat /srv/salt/hello/init.sls”

---------

/tmp/infra-as-code:

file.managed

---------

“sudo salt '\*' state.apply hello”

top.sls määrittelee mitä tiloja orjille ajetaan

”cat /srv/salt/top.sls”

-----------

base:

‘\*’:

* hello

-----------

komento: “sudo salt ‘\*’ state.apply

* Salt contributors: [Salt overview](https://docs.saltproject.io/salt/user-guide/en/latest/topics/overview.html#rules-of-yaml), kohdat
  + Rules of YAML  
      
    Koodi järjestellään avain: arvo pareina  
      
    Mappaus käyttää kaksoispistettä ja yksi välilyönti ”: ” merkitäkseen avain: arvo parit   
      
    Avainten arvot voivat olla monessa erilaisessa rakenteessa.  
      
    Isot ja pienet kirjaimet vaikuttavat avaimissa ja ominaisuuksissa.

Tabulaattoria ei käytetä, vain välilyöntiä.

Kommentointi alkaa ”#”.

* + YAML simple structure
  + 1. Skaalaajat: avain: arvo mappaukset, jossa arvo voi olla merkkijono, numero tai boolean operaattori  
    2. Listat: avain, jota seuraa lista arvoja omalla rivillää kahden välilyönnin ja viivan kanssa  
    3. Sanakirjat: kokoelma avainten arvo mappauksia ja listoja  
      
    Lists and dictionaries - YAML block structures

YAML on järjestelty lohko rakenteiksi  
Sisennys määrää kontekstin. On pakko sisentää ominaisuude ja listat yhdellä tai useammalla välilyönnillä. Kaksi välilyöntiä on standardi.

* Salt contributors: [Salt states](https://docs.saltproject.io/salt/user-guide/en/latest/topics/states.html#state-modules), kohdat
  + State modules  
    Konflikteja voi esiintyä  
      
    moduuleilla ei ole ”status check” mahdollisuutta, vaan täytyy kutsua orjaa selvittääkseen statuksen
  + The state SLS data structure  
      
    Identifier tunniste  
      
    State modulin nimi  
      
    Function jota kutsutaan, kuten installed  
      
    Name tilan nimi kutsu, yleensä tiedoston nimi  
      
    Arguments parametrit joita, funktio hyväksyy
  + Organizing states  
      
    Salt tilojen puu pitäisi kirjoittaa niin, että toinen kehittäjä helposti näkee tilan tarkoituksen ja näkee kokopuun ”workflown”  
      
    hyvä käytäntö on, että ei luo liikaa pesiä polkuun, vain pari tasoa. Näin puu on helpommin navigoitavissa ja ongelmia tulee vähemmän tulevaisuudessa.
  + The top.sls file  
      
    Koska joissain ympäristössä on satoja tiedostoja kohdistuen orjiin, ei ole käytännöllistä ajaa jokainen tilaa yksitellen. tähän auttaa top.sls tiedosto  
      
    top.sls tiedosto kartoittaa mitä orjien tulisi noutaa tietyistä ym  
    päristöistä  
      
    top.sls määrittää mitä tiloja tulisi suorittaa noista ympäristöistä
  + Create the SSH state, Create the Apache state  
      
    Eli tänne pitäisi laittaa name kohdan tilalle muttuja, tiedosto on sitten myös automatisoitu toimimaan paremmin uudelleen käytettäessä?  
      
    Apache state, apachen uudellen käynnistyksen kohdalla sanoisin, että .conf tiedosto tulee käyttöön, voisi olla hyödyllistä käynnistää demoni uudestaan service-watchin avulla, joka seuraa .conf tiedostoa. En kyllä tiedä miten tämä toteutetaan.   
    - Katso näitä alakohtia kriittisin silmin. Teron huomioita ja suosituksia:
      * Namevar: itse asia tilan id:ksi, name arvoksi tulee automaattisesti sama
      * Puuttuko esimerkeistä service-watch? Jos asetustiedosto päivittyy herralla, tuleeko se koskaan käyttöön orjilla?
  + Karvinen 2018: [Pkg-File-Service – Control Daemons with Salt – Change SSH Server Port](https://terokarvinen.com/2018/04/03/pkg-file-service-control-daemons-with-salt-change-ssh-server-port/?fromSearch=karvinen%20salt%20ssh)

Artikkeli käsittelee demonien ohjausta Saltilla ja kuinka vaihtaa SSH serveriin alternatiivinen portti  
  
  
  
a) Hello SLS! Tee Hei maailma -tila kirjoittamalla se tekstitiedostoon, esim /srv/salt/hello/init.sls.

* b) Top. Tee top.sls niin, että tilat ajetaan automaattisesti, esim komennolla "sudo salt '\*' state.apply".
* c) Apache. Asenna Apache, korvaa sen testisivu ja varmista, että demoni käynnistyy.
  + Ensin käsin, vasta sitten automaattisesti.
  + Kirjoita tila sls-tiedostoon.
  + pkg-file-service
  + Tässä ei tarvita service:ssä watch, koska index.html ei ole asetustiedosto
* d) SSHouto. Lisää uusi portti, jossa SSHd kuuntelee.
  + Tämä tehtävä on helpointa tehdä tavallisella virtuaalikoneella, jota Vagrant ei ohjaa.
  + Löydät oikean asetuksen katsomalla SSH:n asetustiedostoa
  + Nyt tarvitaan service-watch, jotta demoni käynnistetään uudelleen, jos asetustiedosto muuttuu masterilla

5 Luento

### h5 CSI Kerava

x) Lue ja tiivistä. (Tässä x-alakohdassa ei tarvitse tehdä testejä tietokoneella, vain lukeminen tai kuunteleminen ja tiivistelmä riittää. Tiivistämiseen riittää muutama ranskalainen viiva.)

* + Karvinen 2018: [Apache User Homepages Automatically – Salt Package-File-Service Example](https://terokarvinen.com/2018/04/03/apache-user-homepages-automatically-salt-package-file-service-example/)

sudo find -printf '%T+ M %p\n%A+ A %p\n%C+ C %p\n'|sort  
  
find -printf "%T+ %p\n"|sort  
  
Kaksi eri tapaa, nopea cmd ja parempi tapa, jossa ei tarvitse suorittaa shell komnentoja.

a) CSI Kerava. Näytä 'find' avulla viimeisimmäksi muokatut tiedostot /etc/-hakemistosta ja kotihakemistostasi. Selitä kaikki käyttämäsi parametrit ja format string 'man find' avulla.

A screenshot of a computer

Description automatically generated

Apachen hakemistossa komento ” find -printf '%T+ %p\n'”

Find etsi komento

-printf tulostaa komento

-%T tiedoston viimeisin muokkaus aika

-%p tiedoston nimi

-\n rivin vaihto

A screenshot of a computer

Description automatically generated

Kotihakemistossa komento ” find -printf '%T+ %p\n'”

b) Gui2fs. Muokkaa asetuksia jostain graafisen käyttöliittymän (GUI) ohjelmasta käyttäen ohjelman valikoita ja dialogeja. Etsi tämä asetus tiedostojärjestelmästä.  
  
Vaihdoin Firefoxin alku sivun ”http://www.heusala.fi/HH/PalvelintenHallinta.html”

mutta en löytänyt etc/firefoxi-esr hakemistosta mitään tiedostoa mitä olisi muokattu.

Kokeilin vielä vaihtaa fonttien asetuksia Libreofficesta, mutta en löytänyt näitäkään.

Kokeilin juuri hakemistossakin tehdä komentoa ” find -printf '%T+ %p\n' | sort” mutta tuli liikaa tiedostoja ja en alkanut sieltä etsimään, kun pari viimeistä vaihtoehtoa ei ollut sopivia

c) Komennus. Tee Salt-tila, joka asentaa järjestelmään uuden komennon.

Loin tiedoston hello tiedoston srv/salt/hello/ kansioon sisällöllä:

A screenshot of a computer program

Description automatically generated

Loin init.sls tiedoston samaan kansioon sisällöllä:

A screenshot of a computer program

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

komento ”sudo salt-call –local state.apply hello” Näkyi että uusi tiedosto oli luotu.

A screen shot of a computer

Description automatically generated

Kävin vielä katsomassa usr/local/bin hakemistosta, että tiedosto oli luotu sinne ja mitä se sisälsi. koitin vielä, mitä käy kun kirjoitaa missä tahansa hakemistossa hello. Se suoritti tuon ohjelman, joka tulostaa sanan Heloo ja päivämäärän, ajan ja aikavyöhykkeen.

d) Apassi. Tee Salt-tila, joka asentaa Apachen näyttämään kotihakemistoja.

Olen mökillä ja jäätyi putket, hitto. Pitää lähteä naapuriin saunomaan, teen tämän myöhemmin.

e) Ämpärillinen. Tee Salt-tila, joka asentaa järjestelmään kansiollisen komentoja.

tein kansion ”sudo mkdir -p kansiollinenkomentoja” srv/salt/ kansioon.

Kopion kansioon aikaisemmin luomani hello tiedoston ja loin uuden tiedoston pyyttono.py alla olevalla sisälöllä:

A screen shot of a computer program

Description automatically generated

sitten komento ”sudoedit init.sls”

A screen shot of a computer

Description automatically generated

Muuten sama sisältö tiedostossa kuin hello kansiossa, mutta lisäsin siihen uuden kohdan pyyttoni.py tiedostoa varten. ja muutin kummankin tiedoston polut vastamaan kansiorakennettani.

A screenshot of a computer

Description automatically generated

Komento ”sudo salt-call –local state.apply hello” näkyi, että yksi tiedosto oli muuttunut ja se oli pyyttoni.py, koska hello tiedosto löytyi jo usr/local/bin kansiosta.

A computer screen with white text

Description automatically generated

ajoin vielä pyyttoni.py komennon