

Puppet installasjon på egne Ubuntu-servere

Puppet er et open-source system for “konfigurasjonsstyring”; dvs. at man kan styre så og si alt av ressurser / konfigurasjoner på sine servere og klienter. Feks kan man si at “alle mine web-servere skal ha nginx med sånn-og-sånn konfigurasjon” eller “denne brukern skal finnes på alle maskinene mine, og den skal være medlem i sudo-gruppa”. Oppgaven kan direkte knyttes opp mot læreplanmål

“Bruke verktøy for å automatisere og effektivisere driftsoppgaver”

I tillegg bør oppgave fine-tune’e linux- og terminalferdigheter.

Vi skal konfigurere en VM (e-linux-server-XX) til å være “puppet master” og så skal vi “styre” e-linux-client-XX VMen med den.

Forutsetning! Du må kunne logge inn med SSH på begge serverne VMene med SSH og ha Wireguard VPN aktivert. Husk å endre root-passordet ditt på alle VMene hvis ikke det er gjort. Bruk passwd kommandoen.

Der man skal endre på filer, så kan man bruke “sudo nano <filnavn>”, feks “sudo nano /etc/hosts”

Det anbefales å åpne to terminalviduer og ha klienten og servern i hver sin.

Denne oppgaven / guiden skal følges “top → down”

- **logg inn på e-linux-server-XX med SSH**
 - (feks ssh fredrik@e-linux-server-48.lab.bakka.party)
 - Sjekk mailen deres for server og VPN info
- **Installere puppetserver repository**
 - wget https://apt.puppet.com/puppet7-release-focal.deb
 - sudo dpkg -i puppet7-release-focal.deb
 - sudo apt update && sudo apt upgrade -y
 - sudo apt install puppetserver
- **Tweake litt innstillinger på det som skal bli vår puppetserver**
 - Endre denne linjen i “/etc/default/puppetserver”

- JAVA_ARGS="-Xms2g -Xmx2g
-Djruby.logger.class=com.puppetlabs.jruby_utils.jruby.Slf4jLogger"
 - til
 - JAVA_ARGS="-Xms1g -Xmx1g
-Djruby.logger.class=com.puppetlabs.jruby_utils.jruby.Slf4jLogger"
 - Endre en linje i filen /etc/hosts
 - "127.0.0.1 localhost"
 - til
 - "127.0.0.1 localhost puppet"
- **Starte og "enable" puppetserver-tjenesten på server**
 - sudo systemctl enable puppetserver.service
 - sudo systemctl start puppetserver.service
 - /opt/puppetlabs/bin/puppetserver -v
 - Burde gi en versjon hvis alt er OK

- **logg inn på e-linux-client-XX med SSH**

- wget https://apt.puppet.com/puppet7-release-focal.deb
- sudo dpkg -i puppet7-release-focal.deb
- sudo apt update
- sudo apt install puppet-agent

Fortsatt på klienten, så skal vi nå legge inn en DNS-record som peker navnet "puppet" til puppetservern din IP-adresse, feks:

- Finn IP-adressen på din puppetserver (e-linux-server-XX) og kopier den.
 - Legg den så inn i klienten (e-linux-client-XX) sin /etc/hosts fil etterfulgt av ordet "puppet", feks slik:
 - 172.31.24.48 puppet
 - Vis at du får pinget servern fra klienten med "ping puppet"
- Start og "enable" tjenesten for puppet-agent
 - sudo systemctl start puppet.service
 - sudo systemctl enable puppet.service
 - sudo /opt/puppetlabs/bin/puppet resource service puppet ensure=running enable=true
 - sudo /opt/puppetlabs/bin/puppet agent --test
 - (her kontakter puppet agent (klienten) server for første gang og det registreres en "certificate request" på server-siden)

- **logg inn på e-linux-server-XX med SSH**
 - `sudo /opt/puppetlabs/bin/puppetserver ca list --all`
 - Her bør du se “e-linux-client-XX” under requests
 - `sudo /opt/puppetlabs/bin/puppetserver ca sign --certname <hostname>`
 - Bytt ut <hostname> med din klient sitt navn, og ikke ta med < og >
- **logg inn på e-linux-client-XX med SSH**
 - `sudo /opt/puppetlabs/bin/puppet agent --test`
 - Du bør se at klienten nå klarer å “apply catalog”

Nå har vi kommet til det punktet at man kan “fjernstyre” e-linux-client-XX VMen sin fra via puppetserveren. Nå skal vi lage noe enkel puppet-kode som gjør slik at vår klient lager en fil hos seg.

- **logg inn på e-linux-server-XX med SSH**
 - `cd /etc/puppetlabs/code/environments/production/modules`
 - `sudo apt install pdk -y`
 - `sudo pdk new module my_module`
 - Trykk enter til den er ferdig
 - `cd my_module`
 - (her går vi inn i modulen som pdk laget)
 - `pdk new class my_class`
 - (nå lages en ny klasse i vår nylagde modul)
 - `sudo nano manifests/my_class.pp`
 - Endre filen så den inneholder dette:

```
class my_module::my_class {
  file { ['/tmp/hello']:
    ensure => 'present',
    content => "Dette kommer filen til å inneholde\n",
    path => '/tmp/hello',
  }
}
```

- Nå skal vi gå til en annen mappe og lage en fil som kaller klassen og module
 - `cd /etc/puppetlabs/code/environments/production/manifests`
 - `sudo nano site.pp`
 - Endre filen så den inneholder dette:

```
node default {  
  include my_module::my_class  
}
```

- **logg inn på e-linux-client-XX med SSH**
 - `sudo /opt/puppetlabs/bin/puppet agent --test`
 - `cat /tmp/hello`
 - (Du bør se en setning i den filen, som kommer fra puppet)

Her er noen bonus-oppgaver:

- **Endre innholdet i /tmp/hello filen på klienten til noe annet via puppet og “apply” endringen.**
- **Installer disse pakkene på klienten ved hjelp av puppet**
 - `mlocate`, `sl`, `screen`
 - (bør helst lage en klasse for pakke-installasjoner, feks `packages.pp`)
- **Last ned og installer puppetlabs-accounts modulen og bruk den til å lage en bruker på din klient via puppet.**
 - Hint: <https://forge.puppet.com/modules/puppetlabs/accounts/readme>
 - Du kan lage en egen klasse, feks `users.pp` og kalle `accounts`-modulen fra inni der, feks

```
accounts::user {  
  ( ... kode ... )  
}
```

Vis at brukeren ble laget på klienten!