

Installation av PUPPET master och agent för automatisk utrullning av tjänst

Steg 1: Konfiguration av puppet-master

Om ingen DNS-server används måste /etc/hosts filen redigeras där FQDN måste matcha IP-adressen.

För att installera och starta tjänsten:

```
sudo wget https://apt.puppet.com/puppet-release-noble.deb
sudo dpkg -i puppet-release-noble.deb
sudo apt update -y
sudo apt install puppetserver -y
sudo systemctl enable puppetserver
sudo systemctl restart puppetserver
```

Tjänsten kommer med en konfigurationsfil under /etc/puppet/puppet.conf och kan se ut som följande:

```
[main]
certname = // FQDN för master-servern
server = // FQDN för master-servern
environment = production // standard produktionskatalog
```

Installationen ger inte en färdig filstruktur för tjänsten, men enligt officiell dokumentation ska det se ut som följande:

```
/etc/puppet/
--- puppet.conf
--- code
----- environments
----- production
----- modules
----- [Tjänst]
----- manifests
----- init.pp
----- manifests
----- site.pp
```

Katalogen [Tjänst] ska definiera vilken tjänst som automatiskt ska rullas ut och hämtas av agenterna. Därav kan namnet på katalogen exempelvis vara nginx eller apache2. För att exemplifiera ska konfigurationen bygga på en installation av apache2 till agenten.

Filen init.pp under tjänstekatalogen ska definiera hur och vilken tjänst som ska installeras, det kallas för att man skapar upp en klass. Konfiguration kan se ut som följande:

```
class apache2 {
    package { 'apache2':
        ensure => installed,
    }
    service { 'apache2':
        ensure => running,
        enable => true,
        require => Package['apache2'],
    }
}
```

```
}
```

Filen site.pp definierar vilka agenter som tilldelas en särskild klass och kan se ut som följande:

```
node '[FQDN för agent]' {
  include [klass]
}
```

Nu är grundkonfigurationen klar, starta gärna om tjänsten för att ändringar ska appliceras.

Steg 2: Konfiguration av puppet-agent

Samma installationspaket ska hämtas ned, denna gång ska puppet-agent installeras i stället för puppet-server enligt följande:

```
sudo wget https://apt.puppet.com/puppet-release-noble.deb
sudo dpkg -i puppet-release-noble.deb
sudo apt update
sudo apt install puppet-agent -y
sudo systemctl enable puppet
sudo systemctl restart puppet
```

Här måste också /etc/hosts filen redigeras om ingen DNS-server används. Agenternas filstruktur är enklare där konfiguration endast görs i filen puppet.conf som ligger under /etc/puppet. Konfiguration kan se ut som följande:

```
[main]
certname = // FQDN för puppet-agent
server = // FQDN för puppet-master
environment = production
runinterval = 30m // Beskriver tidsintervallet när agenten hämtar uppdateringar ifrån puppet-master
```

Steg 3: Koppla ihop master- och agentserver med certifikat

För att agenten automatiskt ska fråga puppet-master om uppdateringar och att den korrekt ska vara ansluten måste certifikat skapas på agenten och signeras av puppet-master. För att generera ett certifikat skrivs följande på agenten:

```
sudo puppet ssl bootstrap
```

För att lista och signera certifikat skrivs följande på puppet-master:

```
sudo puppetserver ca list --all
```

```
sudo puppetserver ca sign --certname [FQDN för agenten som skickar certifikatsbegäran]
```

För att manuellt testa att agenten kan ta emot apache2 installationen ifrån puppet-master skrivs följande på agenten:

```
sudo puppet agent --test
```

```
! om certifikaten misslyckas, använd sudo puppet ssl clean på agenten och sudo puppetserver ca clean --certname [FQDN] på servern
```

Visualisering

```
/etc/puppet/code/environments/
└── production/
    ├── manifests/
    │   └── site.pp      ← gäller för agenter med environment = production
    └── modules/
        └── apache2/
└── web/
    ├── manifests/
    │   └── site.pp      ← gäller för agenter med environment = web
    └── modules/
        └── nginx/
└── db/
    ├── manifests/
    │   └── site.pp      ← gäller för agenter med environment = db
    └── modules/
        └── mysql/
```