

BACHELOR OPPGAVE

Automatisering av linuxtjenere i Nasjonalbiblioteket Installasjonsmanual

Utarbeidet av:

Henning Fjellheim

Studium:

Bachelorstudium i informasjonssystemer

Innlevert:

26.05.2015



Innholdsfortegnelse

Innholdsfortegnelse

| | |
|--|-----------|
| INNOLDSFORTEGNELSE | I |
| FIGURLIGSTE | II |
| 1 AUTOMATISERING MED SCRIPT | 1 |
| 1.1 OPPSETT FOR PYTHON SCRIPT | 1 |
| 1.2 PYTHON SCRIPT | 2 |
| 1.3 KICKSTART CENTOS 6..... | 4 |
| 1.4 BASHSCRIPT CENTOS 6 | 6 |
| 1.5 KICKSTARTFIL CENTOS 7 | 7 |
| 1.6 BASHSCRIPT CENTOS 7 | 9 |
| 2 INSTALLASJONSVEILEDNING OPPSETT KATELLO | 10 |
| 2.1 INSTALLASJON AV KATELLO | 10 |
| 2.2 TILKOBLING MOT OVRT | 11 |
| 2.3 OPPSETT AV SYNKRONISERING AV PAKKEBRØNNER..... | 12 |
| 2.4 AUTOMATISERT UTRULLING AV HOSTER..... | 18 |

Figurligste

| | |
|--|----|
| Figur 1 – compute resources | 11 |
| Figur 2 - compute resource tekst | 11 |
| Figur 3 - products | 12 |
| Figur 4 - products bildetekst..... | 12 |
| Figur 5 - synkroniseringsplan..... | 13 |
| Figur 6 - create repository | 13 |
| Figur 7 - add repository | 13 |
| Figur 8 - synkroniser pakkebrønn..... | 14 |
| Figur 9 - Lifecycle environment | 14 |
| Figur 10 - add new environment..... | 14 |
| Figur 11 - Yum content | 15 |
| Figur 12 - Publish repository..... | 15 |
| Figur 13 - Promote environment..... | 15 |
| Figur 14 – Choose witch environment to promote to..... | 16 |
| Figur 15 - Activation keys | 16 |
| Figur 16 - Promote activation key to environment..... | 17 |
| Figur 17 - Installation media..... | 18 |
| Figur 18 - Installation media, choose path | 19 |
| Figur 19 - Installation media, choose organization | 19 |
| Figur 20 – Installation media, choose operation system | 20 |
| Figur 21 - Installation media choose mirror | 21 |
| Figur 22 - Provisioning templates..... | 21 |
| Figur 23 - Provisioning template choose os | 22 |
| Figur 24 - Provisioning template partition table | 23 |
| Figur 25 - Provisining template installation media..... | 23 |
| Figur 26 - Provisning template choose template | 24 |

| | |
|---|----|
| Figur 27 - Infrastructure subnet | 25 |
| Figur 28 - infrastructure domains..... | 26 |
| Figur 29 - Infrastructure proxies..... | 26 |
| Figur 30 - Create host group..... | 27 |
| Figur 31 - Host group puppet classes | 27 |
| Figur 32 - Host group network | 28 |
| Figur 33 - Hostgroup operating system..... | 29 |
| Figur 34 - Create new host | 29 |
| Figur 35 - Create new host hostgroup..... | 30 |
| Figur 36 - Create new host insert root password..... | 30 |
| Figur 37 - Create new host provisining template | 31 |
| Figur 38 - Create new host network and hdd..... | 31 |
| Figur 39 - create new host installing | 32 |

1 Automatisering med script

Her vil det komme informasjon om hvordan man starter med automatisering ved hjelp av script

1.1 Oppsett for Python script

Det må nevnes, dette scriptet er ikke tunet for å kjøre i produksjon, for å gjøre dette produksjonsklart, så trengs det en del jobb med det. Det vil heller anbefales å bruke Katello for utrulling av VM'er.

Det første man må gjøre, siden dette scriptet tar i bruk ovirtsdk'en til Python er å installere denne. Kjør disse kommandoene for å sette opp ovirtsdk'en:

```
sudo yum install python-setuptools
sudo easy_install ovirt-engine-sdk-python
```

Det første man må sette opp for å få dette scriptet til å fungere er vmlinuz og initrd.img filene. Disse kjøres fra ISO domenet på oVirt hosten. Slik som dette scriptet er satt opp, så laster den vmlinuz og initrd.img for Centos 7.x og vmlinuz6 og initrd6.img for Centos 6.x

Under Centos 6, så ligger disse filene på installasjonscd'en under isolinuxområdet. Disse må gis nye navn fra vmlinuz og initrd.img til vmlinuz6 og initrd.6.img etter at dette er gjort, kjør denne kommandoen på ovirt-hosted hosten:

```
engine-iso-uploader --iso-domain=ISO_DOMAIN upload initrd6.img vmlinuz6
```

Filene ligger på same lokasjon for Centos 7, så der er det bare å utføre den same operasjonen uten å døpe om filene:

```
engine-iso-uploader --iso-domain=ISO_DOMAIN upload initrd.img vmlinuz
```

Det neste man må gjøre er å laste ned ca.pem, denne ligger på <https://ovirt-servernavn/ca.crt> last ned denne, døp den om til ca.pem og legg den i /etc/pki/ovirt-engine/ca.pem på hosten som skal kjøre scriptet.

Så over til parametre i scriptet: URL = URL til ovirt-hosted hosten
USERNAME, PASSWORD og CA_FILE disse er relativt selvforklarende.

Så må man velge Centos versjon, dette scriptet har valgene 6 og 7 (kommer til disse valgene senere)

VM_NAME = Navnet på VM'et

VM_IP = Ipadresse til VM'et

SOCKETS = Antall prosessorer

CORES = Antall Cores

MEMORY = Hvor mye ram skal VM'et ha.

DSK_SIZE = Hvor mange GB skal harddisken være

NIC_NAME = Hvilket navn skal nettverkskortet ha

NET_NAME = Hvilket nettverk inne i ovirt skal dette nettverkskortet koble seg til

KICKSTART = Hvor liggerkickstartfilen (Se under kickstartfiler for å se kickstartfilen som har blitt brukt til Centos 6 og Centos 7, disse ble utlevert fra en webserver)

dns, netmask og gateway er selvforklarende, det som må gjøres er å endre linjen: "hostname":

"{0}.henningf.domain".format(VM_NAME)} bytt ut henningf.domain til det domenet som er ønskelig at VM'et skal ha.

Ks (kickstart) scriptene for Centos 6 og Centos 7 kjører lokal installasjon før de laster ned og kjører ett lokal script som installerer puppet på klienten og setter det opp.

1.2 Python script

```

from ovirtsdk.api import API
from ovirtsdk.xml import params
import time

URL = "https://ovirt-host.domain/api"
USERNAME = "admin@internal"
PASSWORD = "secretpassword"
CA_FILE = "/etc/pki/ovirt-engine/ca.pem"

api = API(url=URL, username=USERNAME, password=PASSWORD, ca_file=CA_FILE)

# Choose centos main version, 6 or 7 NB! Defaults to Centos 7
CENTOS_OS_VERSION = 6
VM_NAME = "katello"
VM_IP = "10.0.1.85"
CLUSTER_NAME = "Default"
SOCKETS = 2
CORES = 1
MEMORY = 6
GB = 1024**3

cpu_params = params.CPU(topology=params.CpuTopology(sockets=SOCKETS,
                                                    cores=CORES))
api.vms.add(params.VM(name=VM_NAME,
                    cluster=api.clusters.get(CLUSTER_NAME),
                    template=api.templates.get("Blank"),
                    cpu=cpu_params,
                    memory=MEMORY*GB,
                    display=params.Display(type_="SPICE")))

def wait_vm_state(vm_name, state):
    while api.vms.get(vm_name).status.state != state:
        time.sleep(1)

wait_vm_state(VM_NAME, "down")

STG_DOMAIN = "VirtualMachines"
DSK_NAME = (VM_NAME+"disk1")
DSK_SIZE = 200
NIC_NAME = "eth0"
NET_NAME = "ovirtmgmt"

def wait_disk_state(disk_name, state):
    while api.disks.get(disk_name).status.state != state:

```

```

time.sleep(1)

vm = api.vms.get(VM_NAME)
stg_domain = api.storagedomains.get(STG_DOMAIN)
stg_parms = params.StorageDomains(storage_domain=[stg_domain])
# Boot disk
vm.disks.add(params.Disk(name=DSK_NAME,
                        storage_domains=stg_parms,
                        size=DSK_SIZE*GB,
                        type_="data",
                        interface='VirtIO',
                        format='cow',
                        bootable=True))
wait_disk_state(DSK_NAME, "ok")

# Boot NIC
vm.nics.add(params.NIC(name=NIC_NAME,
                      network=params.Network(name=NET_NAME),
                      interface='virtio'))
boot_if = vm.nics.get(NIC_NAME).mac.address

KICKSTART = "http://10.0.1.35/centos/7/ks.cfg"
KERNEL = "iso://vmlinuz"
INITRD = "iso://initrd.img"

if CENTOS_OS_VERSION == 7:
    KICKSTART = "http://10.0.1.35/centos/7/ks.cfg"

elif CENTOS_OS_VERSION == 6:
    KICKSTART = "http://10.0.1.35/centos/6/ks.cfg"
    KERNEL = "iso://vmlinuz6"
    INITRD = "iso://initrd6.img"

boot_params = {"ks": KICKSTART,
              "ksdevice": boot_if,
              "dns": "10.0.1.200",
              "ip": VM_IP,
              "netmask": "255.255.255.0",
              "gateway": "10.0.1.1",
              "hostname": "{0}.henningf.domain".format(VM_NAME)}
cmdline = " ".join(map("{0[0]}={0[1]}".format, boot_params.iteritems()))
vm.set_os(params.OperatingSystem(kernel=KERNEL,
                                initrd=INITRD,
                                cmdline=cmdline))

vm.update()
vm.start()

# Wait for machine to power off

```

```
wait_vm_state(VM_NAME, "down")
# Remove boot parameters
vm.set_os(params.OperatingSystem(kernel="", initrd="", cmdline=""))
vm.update()
# Start the VM after the installation
vm.start()
api.disconnect()
```

1.3 Kickstart Centos 6

```
#platform=x86, AMD64, or Intel EM64T
#version=DEVEL
# Install OS instead of upgrade
install
# Keyboard layouts
keyboard 'no'
# Shutdown after installation
poweroff
# Root password
rootpw --iscrypted kryptertpasswd
# System timezone
timezone Europe/Oslo
# Use network installation
url --url="http://10.0.1.35/centos/6/x64/"
# System language
lang en_US
# Firewall configuration
firewall --disabled

# System authorization information
auth --useshadow --passalgo=sha512

# SELinux configuration
selinux --disabled
# Do not configure the X Window System
skipx
# This runs the network configuration configured later.
%include /tmp/network.ks

# System bootloader configuration
bootloader --location=mbr
# Partition clearing information
clearpart --all --initlabel
```



```

# Trying to setup hard drive
# Added zerombr because of destroy disk question
zerombr
autopart

# Installing base packages
%packages
@core
%end

# Fix for servername in Centos 6
%pre
#!/bin/sh
for x in `cat /proc/cmdline`; do
    case $x in hostname*)
        eval $x
        echo "network --device eth0 --bootproto static --hostname ${hostname}" >
/tmp/network.ks
        ;;
    esac;
    done
%end

%post
--nochroot --log=/mnt/sysimage/root/ks-post.log
cd /tmp
/usr/bin/curl -O http://10.0.1.35/centos/6/install.sh

#/usr/bin/chmod +x /tmp/install.sh
/bin/bash /tmp/install.sh
%end

```

1.4 Bashscript centos 6

```
#!/bin/bash

# Installing epel, ovirt-guest-agent and vim.
yum install epel-release -y
yum install ovirt-guest-agent -y
yum install vim -y

# Setting up puppet repo and installing puppet.
sudo rpm -ivh http://yum.puppetlabs.com/puppetlabs-release-el-6.noarch.rpm
yum install puppet -y

# Setting up puppet config.

function puppet_config {
  cat <<EOF> /etc/puppet/puppet.conf
[main]
  # The Puppet log directory.
  # The default value is '$vardir/log'.
  logdir = /var/log/puppet

  # Where Puppet PID files are kept.
  # The default value is '$vardir/run'.
  rundir = /var/run/puppet

  # Where SSL certificates are kept.
  # The default value is '$confdir/ssl'.
  ssl_dir = $vardir/ssl

[agent]
  # The file in which puppetd stores a list of the classes
  # associated with the retrieved configuration. Can be loaded in
  # the separate ``puppet`` executable using the ``--loadclasses``
  # option.
  # The default value is '$confdir/classes.txt'.
  classfile = $vardir/classes.txt

  # Where puppetd caches the local configuration. An
  # extension indicating the cache format is added automatically.
  # The default value is '$confdir/localconfig'.
  localconfig = $vardir/localconfig
  server = puppetmaster.henningf.domain
EOF
}
# Setting up puppet config.
puppet_config
# Onetime puppet run
puppet agent --onetime --no-daemonize
# Starting puppet service
chkconfig puppet on
```

1.5 Kickstartfil Centos 7

```
#platform=x86, AMD64, or Intel EM64T
#version=DEVEL
# Install OS instead of upgrade
install
# Keyboard layouts
keyboard 'no'
# Shutdown after installation
shutdown
# Root password
rootpw --iscrypted secretpassword
# System timezone
timezone Europe/Oslo
# Use network installation
url --url="http://10.0.1.35/centos/7/x64/"
# System language
lang en_US
# Firewall configuration
firewall --disabled
# Including network from temp location
%include /tmp/network.ks
# System authorization information
auth --useshadow --passalgo=sha512
# Use graphical install
graphical
# SELinux configuration
selinux --disabled
# Do not configure the X Window System
skipx

# System bootloader configuration
bootloader --location=mbr
# Partition clearing information
clearpart --all --initlabel
# Trying to setup hard drive
autopart

# Installing base packages
%packages
@core
%end
# Setting static ip
%pre
#!/bin/sh
ip=`cat /proc/cmdline | awk {'print $2'}`
gateway=`cat /proc/cmdline | awk {'print $7'}`
hostname=`cat /proc/cmdline | awk {'print $3'}`
echo "network --device eth0 --bootproto=static --$ip --$gateway --netmask=255.255.255.0 --"
```

```
nameserver=10.0.1.200 --$hostname" > /tmp/network.ks
%end

%post
cd /tmp
/usr/bin/curl -O http://10.0.1.35/centos/7/install.sh
/usr/bin/chmod +x /tmp/install.sh
/tmp/install.sh
%end
```

1.6 Bashscript centos 7

```
#!/bin/bash

yum install epel-release -y
yum install ovirt-guest-agent -y
yum install vim -y
# Puppet stuff.
sudo rpm -ivh http://yum.puppetlabs.com/puppetlabs-release-el-7.noarch.rpm
yum install puppet -y

function puppet_config {
cat <<EOF> /etc/puppet/puppet.conf
[main]
# The Puppet log directory.
# The default value is '$vardir/log'.
logdir = /var/log/puppet

# Where Puppet PID files are kept.
# The default value is '$vardir/run'.
rundir = /var/run/puppet

# Where SSL certificates are kept.
# The default value is '$confdir/ssl'.
ssldir = $vardir/ssl

[agent]
# The file in which puppetd stores a list of the classes
# associated with the retrieved configuration. Can be loaded in
# the separate ``puppet`` executable using the ``--loadclasses``
# option.
# The default value is '$confdir/classes.txt'.
classfile = $vardir/classes.txt

# Where puppetd caches the local configuration. An
# extension indicating the cache format is added automatically.
# The default value is '$confdir/localconfig'.
localconfig = $vardir/localconfig
server = puppetmaster.henningf.domain
EOF
}
puppet_config
# Starting and enabling puppet
/usr/bin/systemctl enable puppet
/usr/bin/systemctl start puppet
```

2 Installasjonsveiledning oppsett katello

Det ble i dette prosjektet satt opp katello for å teste ut sentral styring av repositories. Dette kapittelet vil gjennomgå installasjon og oppsett av katello

2.1 Installasjon av katello

Katello ble installert på en virtuell Centos 6.6 host med 6gb med ram og 2cpu'er. Hosten ble rullet ut via utrullingscriptet. Deretter ble det kjørt en standard installasjon av katello:

Det første som ble satt opp var pakkebrønnene som trengs til Centos 6 for katello:

```
yum -y localinstall
http://fedorapeople.org/groups/katello/releases/yum/nightly/katello/RHEL/6Server/x86_64/katello-repos-latest.rpm
yum -y localinstall http://yum.theforeman.org/nightly/el6/x86_64/foreman-release.rpm
yum -y localinstall http://yum.puppetlabs.com/puppetlabs-release-el-6.noarch.rpm
yum -y localinstall http://dl.fedoraproject.org/pub/epel/epel-release-latest-6.noarch.rpm yum
-y install foreman-release-scl
```

Deretter ble katello installeren installert:

```
yum -y install katello
```

Så ble katello installert her er det ikke tatt hensyn til at katello installeren har mange valg man burde se på, som f.eks --foreman-initial-organization som standard står til Default Organization.

NB: Hosten burde være satt opp i DNS før man starter katello-installer, alternativt. Lag pekere til hosten i /etc/hosts

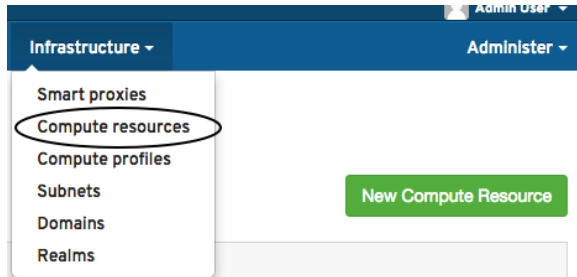
Denne kommandoen ble brukt for å installere katello:

```
katello-installer
```

Katello vil så installere seg selv og gi en output når den er ferdig, den er nå klar til å logges på.

2.2 Tilkobling mot oVirt

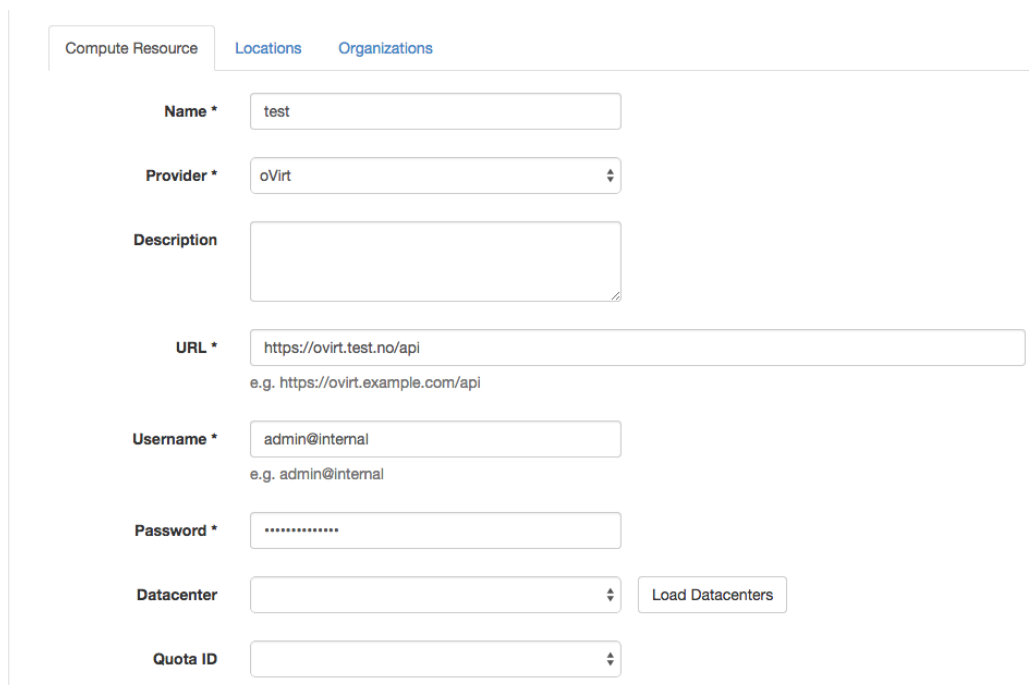
Under infrastructure velg compute resources



Figur 1 – compute resources

Deretter velger man New Compute Resource.

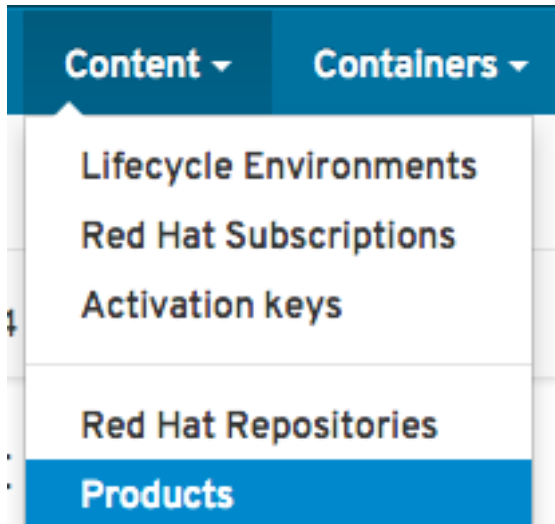
I Det nye vinduet som kommer opp, skriv inn ett navn under Name, i dropdownmenyen provider velg oVirt, i url, fyller man ut linken til ovirt serveren /api. Username må være en bruker som har tilgang til å administrere oVirt. Skriv inn passordet og trykk på load datacenters, trykk submit hvis det er riktig datasenter som er koblet til. NB: Man kan her styre hvilken lokasjon og Organisasjon som ovirt skal være koblet til. Det ble ikke lagt vekt på dette under dette prosjektet. Så disse er satt til standard.

A screenshot of a web form titled 'Compute Resource'. The form has several fields: 'Name' with the value 'test', 'Provider' with a dropdown menu showing 'oVirt', 'Description' which is empty, 'URL' with the value 'https://ovirt.test.no/api' and a hint 'e.g. https://ovirt.example.com/api', 'Username' with the value 'admin@internal' and a hint 'e.g. admin@internal', 'Password' which is masked with dots, 'Datacenter' which is empty and has a 'Load Datacenters' button next to it, and 'Quota ID' which is empty. The form is part of a larger interface with tabs for 'Compute Resource', 'Locations', and 'Organizations'.

Figur 2 - compute resource tekst

2.3 Oppsett av synkronisering av pakkebrønner

Det første man gjør er gjerne å lage ett produkt, ett produkt er en pakkebrønn. Velg Products under Content fanen.



Figur 3 - products

Deretter klikker man på new product opp til høyre, skriv inn navnet på produktet som man ønsker å lage f.eks Centos 6 x64, deretter må man lage en sync plan. Klikk på new sync plan, skriv inn ett navn på denne f.eks daglig. Velg ett intervall og når den skal starte. Starttidspunktet kan også være greit å justere. Trykk save.

| | |
|-------------|---|
| Name* | <input type="text" value="Daglig"/> |
| Description | <input type="text"/> |
| Interval* | <input type="text" value="daily"/> |
| Start Date* | <input type="text" value="2015-04-22"/> |
| Start Time | <input type="text" value="00"/> : <input type="text" value="00"/> |
| | <input type="button" value="Cancel"/> <input type="button" value="Save"/> |

Figur 4 - products bildetekst

Velg synkroniseringsplanen og trykk save.

Name*
Label*
GPG Key
Sync Plan
[+ New Sync Plan](#)
Description

Figur 5 - synkroniseringsplan

Man får så beskjed om at man ikke har noen pakkebrønner koblet til dette produktet, klikk på create repository

Repositories

You currently don't have any Repositories included in this Product, you can add Repositories using the button on the right.

Figur 6 - create repository

Skriv inn ett passende navn på repositoret f.es Centos 6 x64, sett type til yum, hvis dette er ett eldre repository til f.eks centos 5 eller redhat 5, så burde man sette checksum til sha1, hvis ikke så er det bare å la den stå til default. Skriv så inn url'en til pakkebrønnen. Trykk deretter save.

« Add New Repository

Name*
Label*
Type*
Checksum
For older operating systems such as Red Hat Enterprise Linux 5 or CentOS 5 it is recommended to use sha1.
URL
Publish via HTTP
GPG Key

Figur 7 - add repository

Når dette er utført, så får du opp repositories. Hvis du ønsker å synkronisere det hak av ved navnet på pakkebrønnen og klikk sync now

| Name | Type | Sync Status | Content |
|---|------|-------------|------------------------|
| <input checked="" type="checkbox"/> Cenos 6 x64 | yum | Not Synced | 0 Packages 0 Errata |

Figur 8 - synkroniser pakkebrønn

Synkroniseringen tar litt tid.

Deretter går man inn på content, velg lifecycle environment



Figur 9 - Lifecycle environment

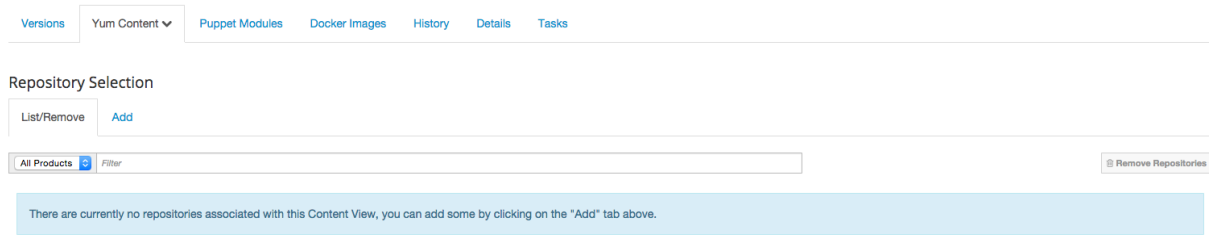
Det første man må gjøre er å lage en ny environment path, klikk på new environment path oppe i høyre hjørne, gi den ett navn. F.eks webserver, man kan nå legge til miljø som man kan koble videre opp mot denne. Man velger da add new environment til høyre over webserver:

| | webserver | webserver-test | webserver-prod |
|---------------|-----------|----------------|----------------|
| Content Views | 0 | 0 | 0 |
| Content Hosts | 0 | 0 | 0 |

Figur 10 - add new environment

Man har ingen content views koblet til denne, det neste man må gjøre er å lage en content view og koble den til ett miljø. Velg content views under content fanen, deretter velger man create new view. Gi viewet ett passende navn. F.eks webserver

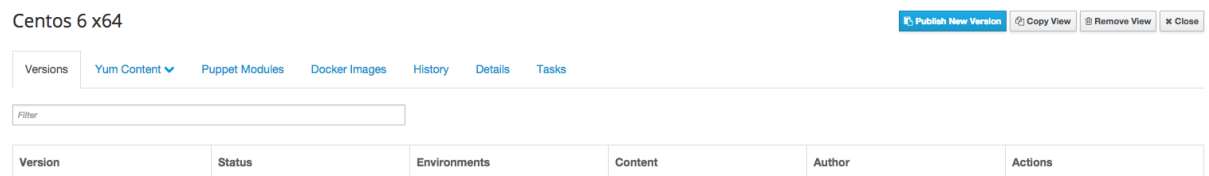
Velg deretter det nye content viewet man har laget fra venstresiden. Klikk deretter på fanen Yum content



Figur 11 - Yum content

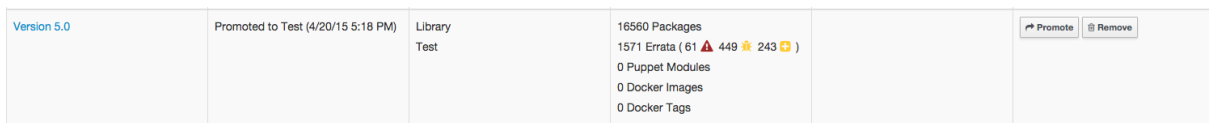
Gå videre til underfanen der det står Add, hak av ved siden av repositoret som ble opprettet for litt siden. Klikk deretter på add repositories.

Man kan nå dele ut repositoret, klikk på publish new version, klikk deretter på save.



Figur 12 - Publish repository

Når dette er ferdig, så kommer det opp versjon 1. av pakkebrønneren. Det man nå må gjøre er å sette denne til det environmentet som ble opprettet tidligere. Trykk på promote:



Figur 13 - Promote environment

Må man promote miljøet ett steg av gangen.

[« Back to Version List](#)

Promote Version 5.0

Choose a lifecycle environment from the available promotion paths.

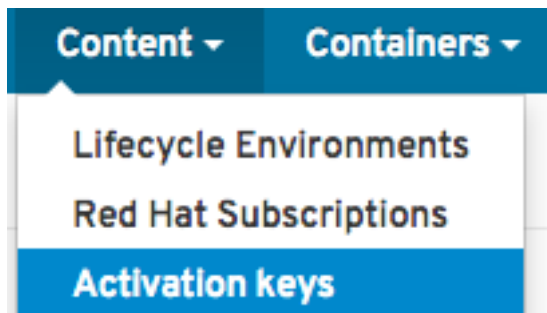


Figur 14 – Choose witch environment to promote to

Har har det blitt oppgradert til versjon 5 og versjonen er promotet til test, man kan videre promote den til dev før den blir promotet til prod til slutt.

Man lager en ny versjon av en pakkebrønn hver gang man ønsker å ha inn nye pakker, gjerne etter en synkronisering. Så, da har man dyttet en versjon til ett environment. Det man da må gjøre er å sørge for at hoster kan koble seg til dette miljøet. For å gjøre dette må man lage activation keys.

Velg content, deretter activation keys:



Figur 15 - Activation keys

Klikk på new activation key,

Gi nøkkelen ett passende navn, man ønsker gjerne å ha unlimited content hosts.

Velg environment (helst ett som man allerede har lagt pakker til som beskrevet lengre opp)

Figur 16 - Promote activation key to environment

Først må man installere subscription manager (Dette er kun testet på Centos 6 og Centos 7

```
yum install subscription-manager -y
```

Når man så har satt opp ett miljø, med en nøkkel. Så kan man registrere seg mot dette:

```
rpm -Uvh http://$KATELLO_HOSTNAME/pub/katello-ca-consumer-latest.noarch.rpm
```

Man må så registrere seg mot serveren, går man inn på activation key'en så står den eksakte infoen om hvordan man skal koble seg på, ett eksempel fra testen:

```
subscription-manager register --org="henningf" --activationkey="testkey C7"
```

Man har nå satt opp en host mot katello, det som ikke er tatt med her er at lokale pakkebrønner burde slås av, slik at man bare synkroniserer fra katello hvis det er ønskelig. Det må også huskes på at det går å koble flere pakkebrønner mot en aktiveringsnøkkel. Dette kan man gjøre når man går inn på activation keys, velg aktiveringsnøkkelen som man ønsker å legge til flere pakkebrønner mot. Gå til fanen subscriptions, velg add og hak av de pakkebrønnene som man ønsker å knytte til aktiveringsnøkkelen.

Det ble dessverre ikke tid til å gå dypere ned i materien for å teste ut akkurat dette. Men testen viser at dette er ett produkt som fungerer fint, har forsøkt å legge til og fjerne hoster fra aktiveringsnøkler, oppdatert content etc. Uten problemer, det har ikke vært tid til å teste videre rundt utvidet funksjonalitet slik som å dytte nye pakker ut fra Foreman. Men dette er

mulig hvis man legger inn katello-agent (https://access.redhat.com/documentation/en-US/Red_Hat_Satellite/6.0/html/User_Guide/Installing_the_Katello_Agent.html)

2.4 Automatisert utrulling av hoster

Det har i dette prosjektet blitt testet ut automatisk utrulling av hoster fra katello til oVirt, her blir det tatt en gjennomgang av dette.

Det første man må gjøre er å sette 2 parametre i DHCP serveren, dette er opsjon 66 boot server host name og opsjon 67 bootfilename.

I det oppsettet som har blitt brukt under testen, så ble det brukt en metode som heter DNSMasq, da setter man opsjonene slik:

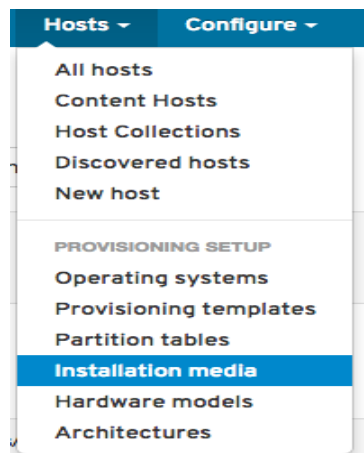
```
dhcp-boot=pxelinux.0,,10.0.1.85
```

Option 66 er her satt til 10.0.1.85 (serveren som leverer ut filen)

Option 67 er her satt til pxelinux.0 filen som skal leveres fra serveren.

Når disse er satt, er det klart til å sette opp katello for å starte opp virtuelle hoster.

Det første man må gjøre er å sette opp installasjon media, velg fanen hosts, deretter installasjons media



Figur 17 - Installation media

Velg new medium. Skriv inn ett passende navn, f.eks Centos mirror, skriv inn adressen til mediet. Dette kan være f.eks *http://mirror.centos.org/centos/\$version/os/\$arch* der \$arch vil bli byttet ut med hosten sin arkitektur f.eks x64 \$version vil bli byttet ut med versjonsnummeret til OS'et man trenger bare denne for å provisjonere ut Centoshoster. Det går også å sette opp disse lokalt.

Medium Locations Organizations

Name *

Path *

The path to the medium, can be a URL or a valid NFS server (exclusive of the architecture). for example *http://mirror.centos.org/centos/\$version/os/\$arch* where **\$arch** will be substituted for the host's actual OS architecture and **\$version**, **\$major** and **\$minor** will be substituted for the version of the operating system. Solaris and Debian media may also use **\$release**.

Operating system family

Figur 18 - Installation media, choose path

Sjekk at den står til riktig Organisasjon:

Medium Locations Organizations

Organizations

| All Items | Filter | + |
|----------------------|--------|---|
| Default Organization | | |

⇔

| Selected Items | - |
|--------------------|---|
| hennings hjemmeorg | |

Figur 19 - Installation media, choose organization

Gå inn på hosts, velg operating system, klikk på new.

Fyll ut name, major version, minor versjon OS Family celg root password hash og arhitectures

Operating System
Partition table
Installation media
Templates
Parameters
✕

Name * OS name from facter; e.g. RedHat

Major version * OS major version from facter; e.g. 6

Minor version OS minor version from facter; e.g. 5

Description e.g. RHEL 6.5

OS Family The Operating System type

Root password hash Hash function to use. Change takes effect for new or updated hosts.

Architectures

Select All

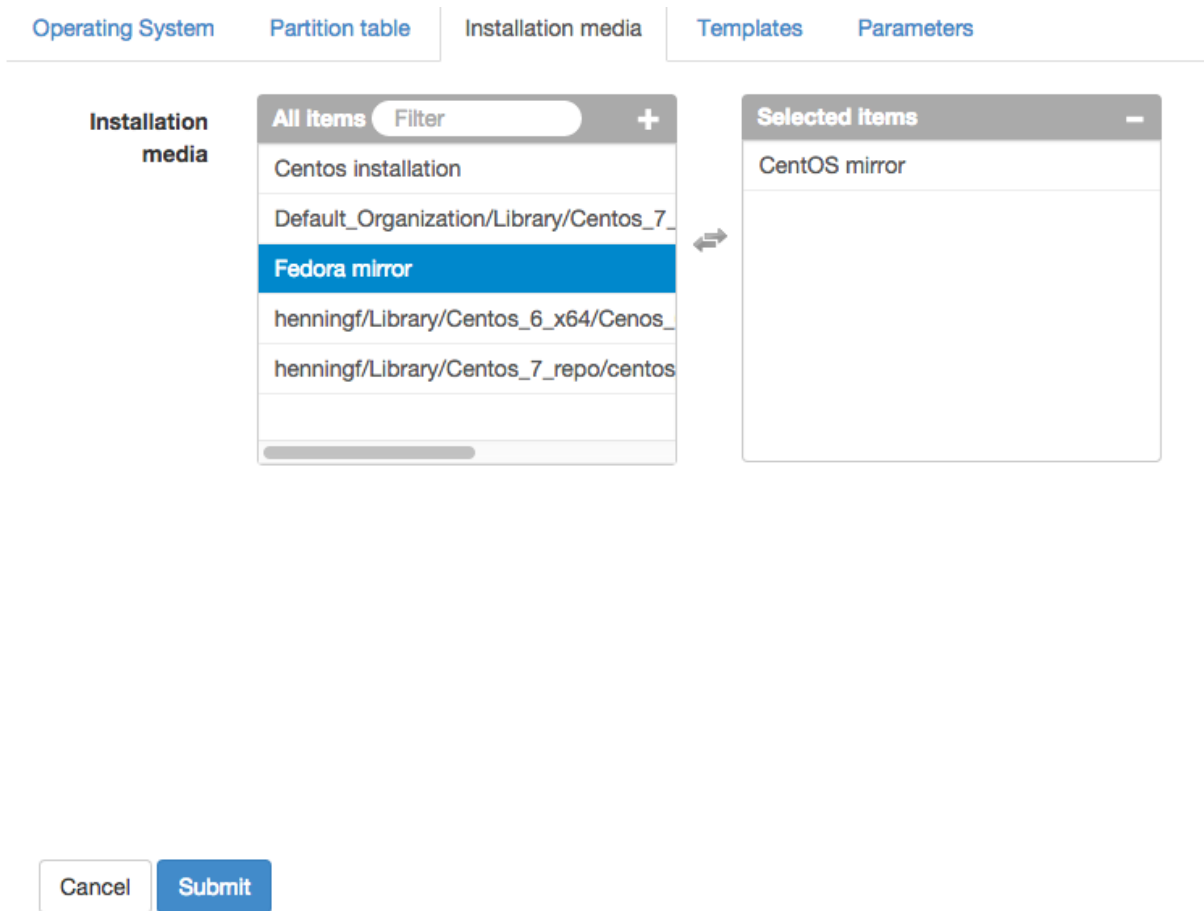
i386

x86_64

Cancel
Submit

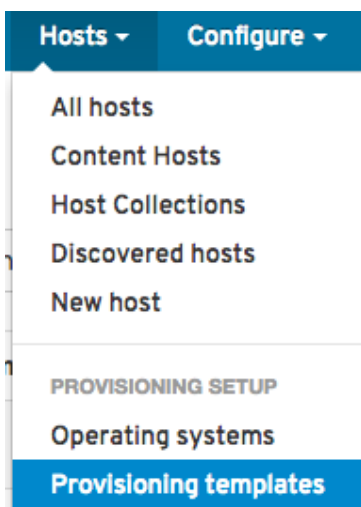
Figur 20 – Installation media, choose operation system

Gå videre til Installation media og velg mirroret som akkurat ble laget.



Figur 21 - Installation media choose mirror

Nå er det på tide å koble templates til dette oppsette, gå til hosts, provision templates.



Figur 22 - Provisioning templates

Velg deretter Katello Kickstart Default, gå på fanen til høyre og klikk clone. Gi den ett nytt navn f.eks Katello Kickstart Default NO. Inne i code, så leter man opp plassen der det står:

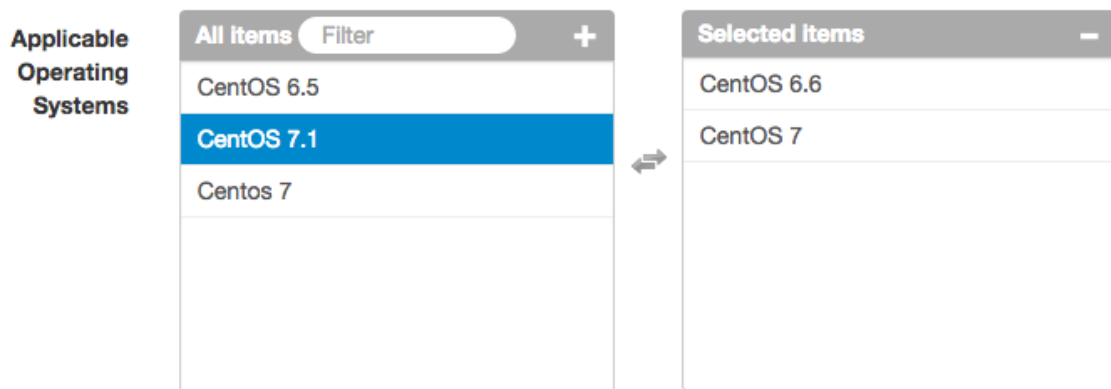
keybord us

Endre denne til

keybord no

Det er flere parametre man kan endre hvis ønskelig, slå av selinux, brannmur etc. Det er ikke brukt lang tid på å fintune denne ut ifra nb sine ønsker.

Gå deretter til fanen association, legg til ostypen som ble opprettet tidligere. Trykk deretter submit.



Valid host group and environment combinations

+ Add combination

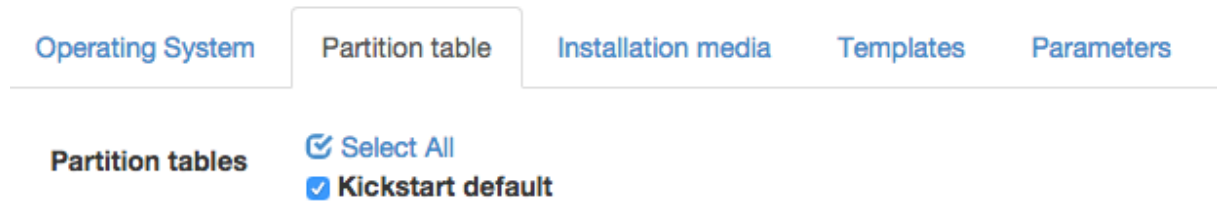
Cancel

Submit

Figur 23 - Provisioning template choose os

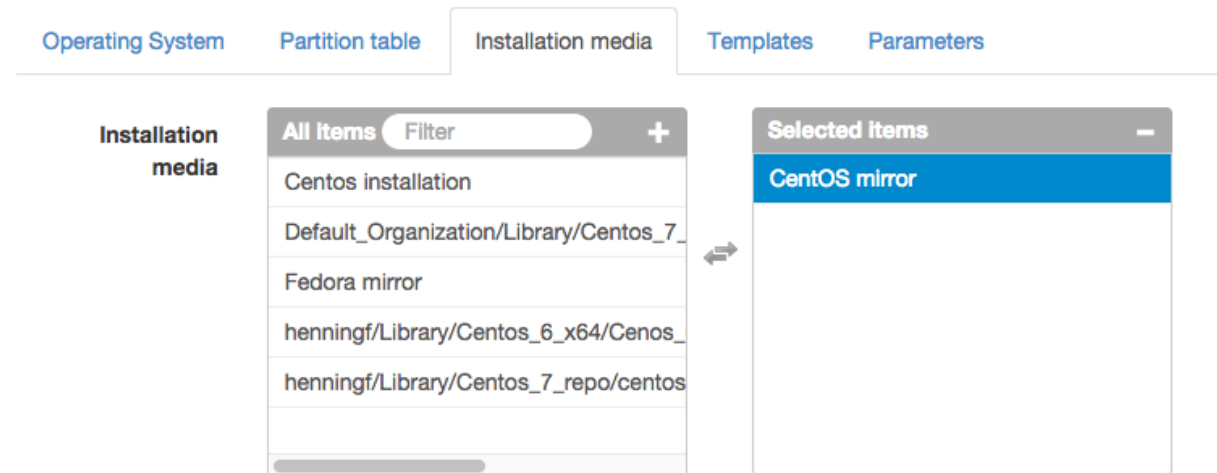
Etter at den er lagt til, trykk på submit, gå inn på på kickstart default PXELinux, kickstart default iPXE, Katello kickstart default finish og katello kickstart default user data og legg til operativsystemet i disse også.

Gå deretter til hosts, velg operation systems. Velg opartivsystemet du holder på med. Gå under partition tables, hak av for kickstart default.



Figur 24 - Provisioning template partition table

Gå deretter under installation media, velg det mirroret som ble opprettet tidligere.



Figur 25 - Provisining template installation media

Deretter går man under templates og setter inn de templatene som vi har sagt at denne hosten får lov til å bruke.

| Operating System | Partition table | Installation media | Templates | Parameters |
|--------------------|-----------------|--------------------|-------------------------------------|------------|
| finish * | | | Katello Kickstart Default Finish | |
| provision * | | | Katello Kickstart Default_NO | |
| user_data * | | | Katello Kickstart Default User Data | |
| iPXE * | | | Kickstart default iPXE | |
| PXELinux * | | | Kickstart default PXELinux | |

Figur 26 - Provisning template choose template

Trykk submit.

Det kan være greit å sette opp domen(er) for hostene som skal settes opp av foreman.

Gå inn på infrastructure, velg domains, klikk på new domain. Skriv inn dns domenenavnet, sjekk at den er koblet til organisasjonen som skal brukes under fanen organizations, det har ikke vært tid til å se på DNS proxy. Så dette er ikke dekt under denne bruksanvisningen.

Trykk submit.

Gå deretter under infrastructure og velg subnets. Trykk på new subnet, fyll informasjonen for subnettet, sett boot mode til DHCP. Gi nettverket ett fornuftig navn.

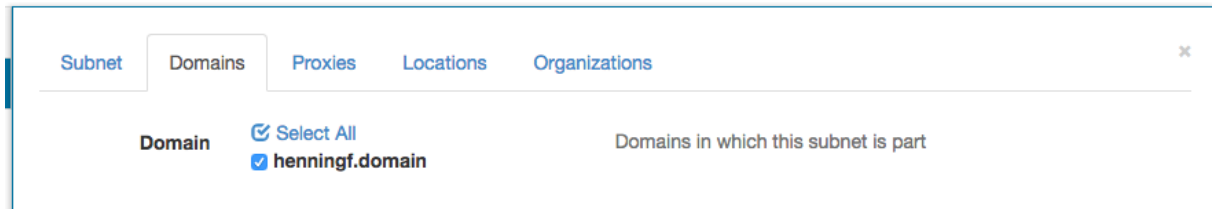
The screenshot shows a web interface for configuring a subnet. At the top, there are tabs for 'Subnet', 'Domains', 'Proxies', 'Locations', and 'Organizations'. The 'Subnet' tab is active. The form contains the following fields and options:

- Name ***: Input field containing 'lokalt nettverk'.
- Network address ***: Input field containing '10.0.1.0'.
- Network mask ***: Input field containing '255.255.255.0'. To the right, it says 'Netmask for this subnet'.
- Gateway address**: Input field containing '10.0.1.1'. To the right, it says 'Optional: Gateway for this subnet'.
- Primary DNS server**: Input field containing '10.0.1.200'. To the right, it says 'Optional: Primary DNS for this subnet'.
- Secondary DNS server**: Empty input field. To the right, it says 'Optional: Secondary DNS for this subnet'.
- IPAM**: A dropdown menu set to 'DHCP'. Below it, a note reads: 'IP Address auto suggestion mode for this subnet, DHCP works with DHCP proxy only'.
- Start of IP range**: Input field containing '10.0.1.150'. To the right, it says 'Optional: Starting IP Address for IP auto suggestion'.
- End of IP range**: Input field containing '10.0.1.160'. To the right, it says 'Optional: Ending IP Address for IP auto suggestion'.
- VLAN ID**: Empty input field. To the right, it says 'Optional: VLAN ID for this subnet'.
- Boot mode**: A dropdown menu set to 'DHCP'. To the right, it says 'Default boot mode for interfaces assigned to this subnet'.

At the bottom left, there are two buttons: 'Cancel' and 'Submit'.

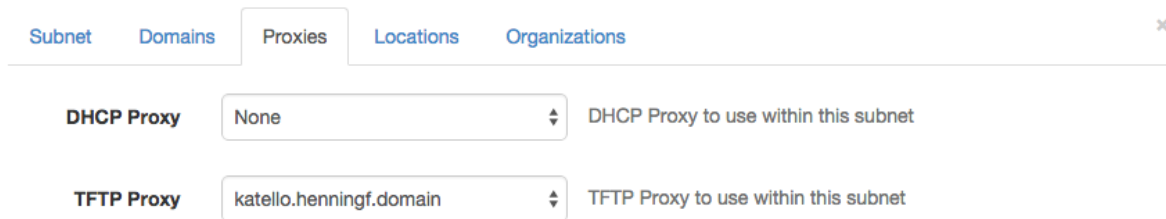
Figur 27 - Infrastructure subnet

Velg domenet:



Figur 28 - infrastructure domains

Gå deretter under fanen proxies og sett tftp proxy til katello:



Figur 29 - Infrastructure proxies

trykk submit.

Lag deretter en host gruppe.

Gå på configure og velg hosts group, klikk på new host group, gi den ett fornuft navn.

Vær obs på en ting, når man kobler til ett lifecycle environment og ett content view. Så må man ha inne ett content view som inneholder puppet pakkene. f.eks epel. Hvis ikke, så kommer ikke hostene til å få installert puppet.

| | | |
|------------------------------|--|-----|
| Parent | <input type="text" value=""/> | ⌵ |
| Name * | <input type="text" value="Centos6hosts"/> | |
| Lifecycle Environment | <input type="text" value="Test"/> | ⌵ |
| Content View | <input type="text" value="Centos 6 x64"/> | ⌵ |
| Puppet Environment | <input type="text" value="KT_henningf_Homepage_Centos_6_x64_5"/> | ⌵ f |
| Content Source | <input type="text" value="katello.henningf.domain"/> | ⌵ t |
| Puppet CA | <input type="text" value="katello.henningf.domain"/> | ⌵ t |
| Puppet Master | <input type="text" value="katello.henningf.domain"/> | ⌵ t |

Figur 30 - Create host group

Man kan så gå videre og legge til eventuell puppetkonfigurasjon

Included Config Groups

▣ IPAGroup
ipacient

⊖ Remove

Available Config Groups

Included Classes

ipacient

Available Classes

▣ ipacient

▣ stdlib

Figur 31 - Host group puppet classes

Gå videre til network og sett inn informasjonen om nettverket

| | |
|---------------|--|
| Domain | <input type="text" value="henningf.domain"/> |
| Subnet | <input type="text" value="lokalt nettverk (10.0.1.0/24)"/> |
| Realm | <input type="text"/> |

Figur 32 - Host group network

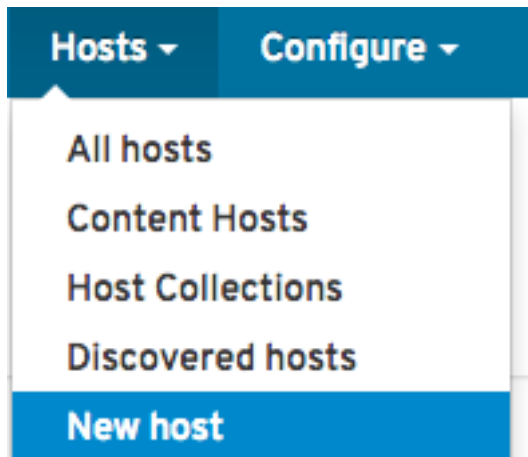
Deretter går man til operating system og setter inn informasjonen som ble opprettet tidligere:

| | | |
|---------------------------|--|----|
| Architecture | <input type="text" value="x86_64"/> | |
| Operating system * | <input type="text" value="CentOS 6.6"/> | |
| Media * | <input type="text" value="CentOS mirror"/> | |
| Partition table * | <input type="text" value="Kickstart default"/> | |
| Root password | <input type="text"/> | Pa |

Figur 33 - Hostgroup operating system

Nå skal alt være klart for å sette opp en ny host,

Gå til hosts, new hosts



Figur 34 - Create new host

Når man nå velger hostgroup, så vil de fleste parametrene bli fylt ut automatisk, husk å sette Deploy on til oVirt

| | | |
|------------------------------|--|-------------------------|
| Name * | <input type="text" value="Centos 6.6 test"/> | |
| Organization * | <input type="text" value="hennings hjemmeorg"/> | ⌵ |
| Location * | <input type="text" value="Default Location"/> | ⌵ |
| Host Group | <input type="text" value="Centos 6.6test"/> | ⌵ |
| Deploy on | <input type="text" value="HenningfOvirt (oVirt)"/> | ⌵ |
| Lifecycle Environment | <input type="text" value="Test"/> | ⌵ |
| Content View | <input type="text" value="Centos 6 x64"/> | ⌵ |
| Puppet Environment * | <input type="text" value="KT_henningf_Homepage_Centos_6_x64_5"/> | ⌵ Reset |
| Content Source | <input type="text" value="katello.henningf.domain"/> | ⌵ Use t |
| Puppet CA | <input type="text" value="katello.henningf.domain"/> | ⌵ Use t |
| Puppet Master | <input type="text" value="katello.henningf.domain"/> | ⌵ Use t |

Figur 35 - Create new host hostgroup

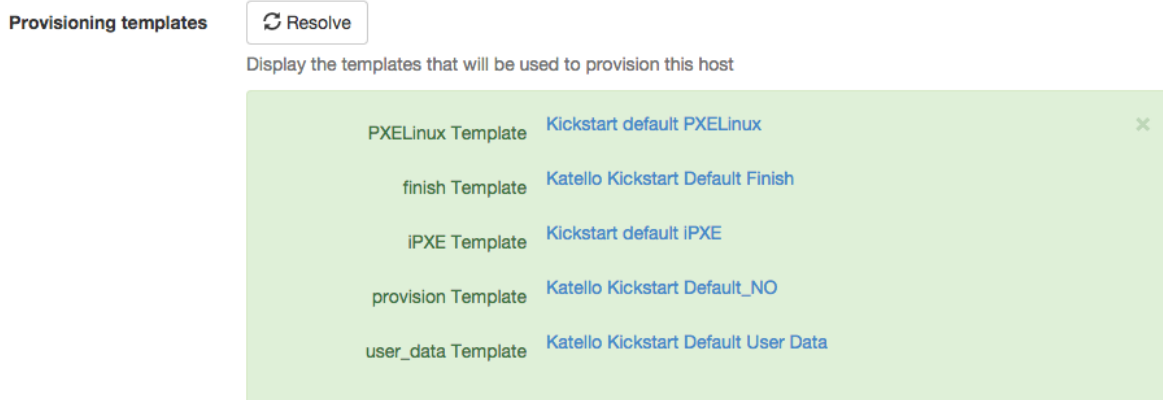
Gå videre til fanen Operating system og skriv inn root password.

Root password *

Password must be 8 characters or more

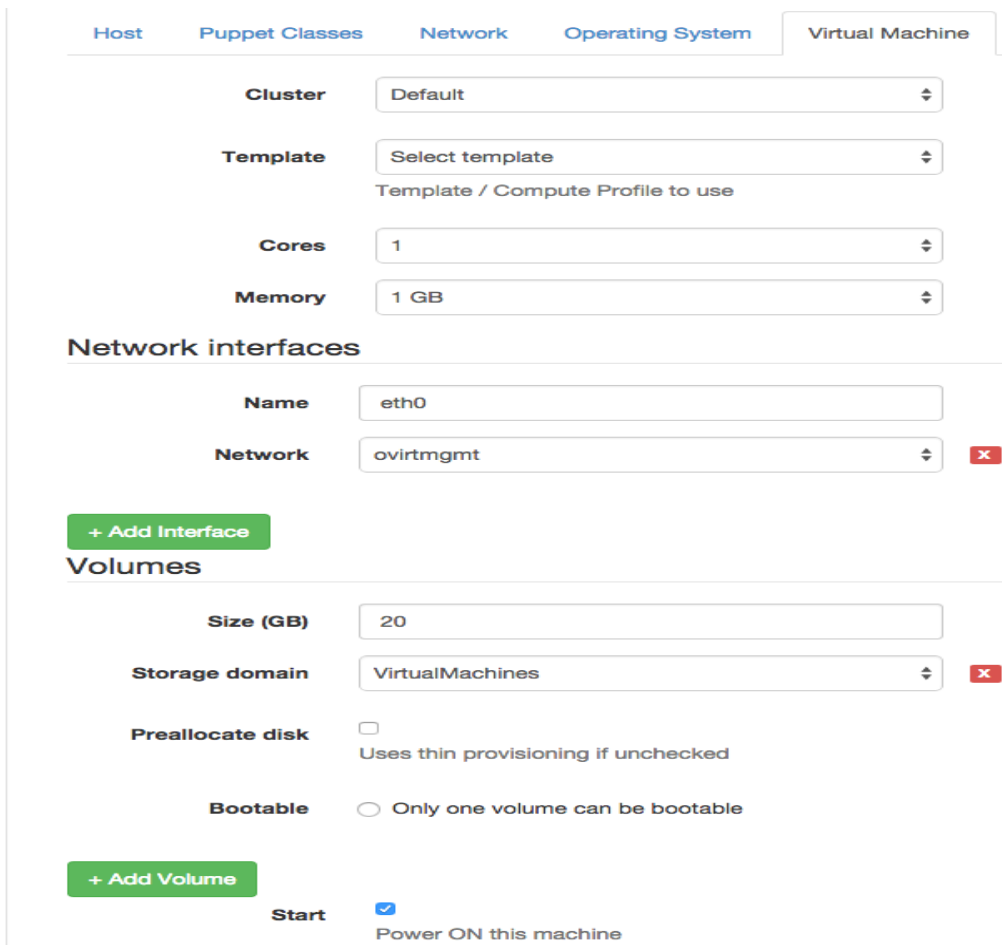
Figur 36 - Create new host insert root password

Man kan også trykke på resolve ved siden av provisioning templates for å sjekke templatene som blir lastet når man skal installere denne hosten.



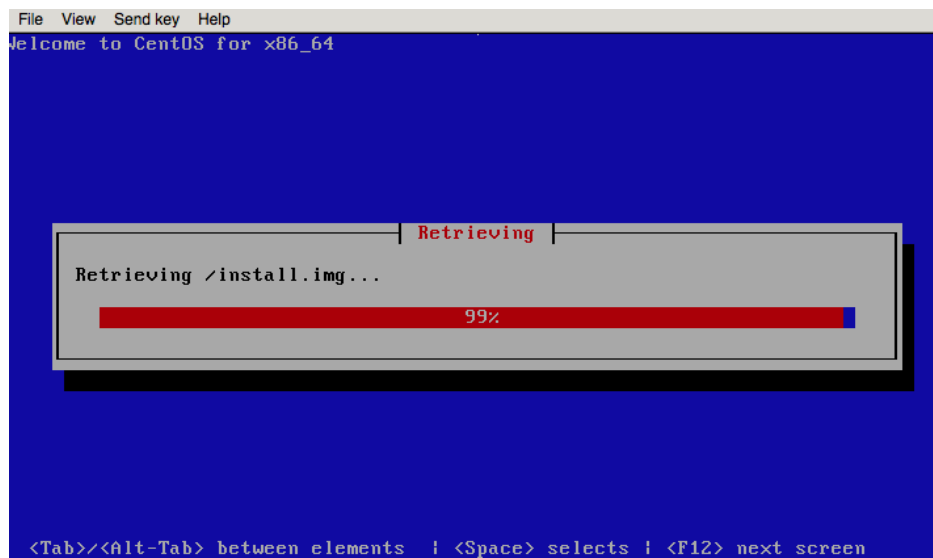
Figur 37 - Create new host provisioning template

Gå videre til virtual machine, legg til ett nettverksinterface, gi det ett navn og legg til en harddisk av adekvat størrelse



Figur 38 - Create new host network and hdd

trykk submit, hvis alt er satt opp riktig. Så kan man nå gå inn i oVirt og se at hosten starter installasjonen.



Figur 39 - create new host installing