



Déploiement d'OS

P. Grégoire

Date: 02/2023

Aperçu et Objectifs

Ce TP va vous faire découvrir les mécanismes d'installation de nœuds via le réseau.

Sommaire

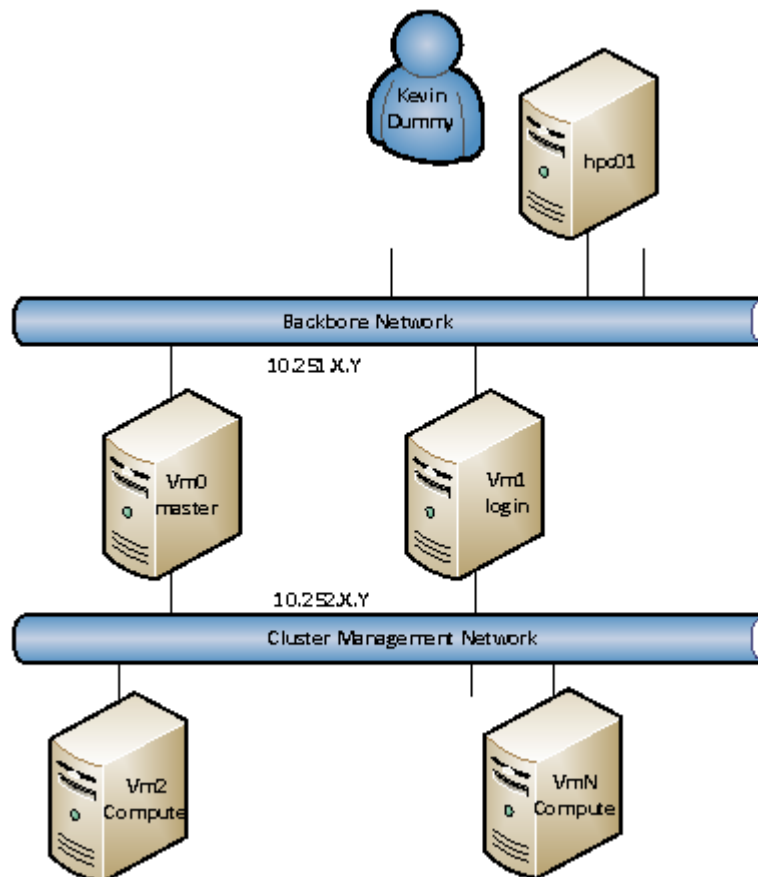
1. Introduction
2. Lancement du cluster
3. Instructions de fin de séance de TP
4. Plan d'adressage
5. Configuration du serveur DHCP
6. Mise en place du système de déploiement

Introduction

Le but de ce TP est de configurer un serveur d'installation permettant d'installer automatiquement une machine via PXE/Kickstart.

Pour le TP, vous allez créer avec pcocc un cluster de 3 nœuds vm[0-2]:

- Le nœud vm0 est la station de contrôle du cluster. Il a deux interfaces réseaux, la première sur le backbone, la seconde sur le réseau de management du cluster qui sert à contrôler les nœuds de calcul et à les installer.
- Le nœud vm1 est un nœud de login. Il a deux interfaces réseaux, la première sur le backbone, la seconde sur le réseau de management du cluster.
- Le nœud vm2 est un nœud de calcul. Il a une seule interface réseau sur le réseau de management du cluster. Son BIOS est configuré pour booter en PXE sur le réseau.



Lancement du cluster

Pour ce TP, vous avez besoin d'utiliser le X forwarding (possibilité de lancer des applications X depuis hpc01). Connectez vous toujours avec `ssh -XY hpc`. Lancer votre cluster virtuel à 3 nœuds avec `pcocc` puis connectez vous sur la première machine `vm0` et passer en root avec la commande `sudo` :

```
pc233-10$ ssh -XY hpc
[kevin.dummy@hpc01 ~]$ script -a trace-tp-install-vm0-$(id -un).txt
Le script a débuté, le fichier est trace-tp-install-vm0-kevin.dummy.txt
[kevin.dummy@hpc01 ~]$ pcocc alloc -c 3 pxe-server,dhcp-client,pxe-client
salloc: Granted job allocation 9848
Configuring hosts... (done)
(pcocc/9848) [kevin.dummy@hpc01 ~]$
```

Instructions de fin de séance de TP

A la fin de la séance du TP, vous pouvez sauvegarder l'état de votre cluster ce qui vous permettra de reprendre votre travail là où vous en étiez. Vous avez besoin de sauvegarder que la `vm0` et la `vm1` :

```
(pcocc/9848) [kevin.dummy@hpc01 ~]$ pcocc save vm0
Saving image...
vm0 frozen
vm0 thawed
vm0 disk succesfully saved to /home/kevin.dummy/.pcocc/images/test-centos74-tp-install/image-rev1
(pcocc/9848) [kevin.dummy@hpc01 ~]$ pcocc save vm1
Saving image...
vm1 frozen
vm1 thawed
vm1 disk succesfully saved to
/home/kevin.dummy/.pcocc/images/test-centos74-tp-install-dhcp-client/image-rev1
(pcocc/9848) [kevin.dummy@hpc01 ~]$ exit
exit
Terminating the cluster...
Relinquishing job allocation 9848
```

Une fois cette sauvegarde effectuée, **quitter une à une les sessions par exit** :

- sur la `vm1`, le `sudo`, puis votre login, puis la session `script`,
- sur la `vm0`, le `sudo`, puis votre login, puis la session `pcocc` puis la session `script`.

Envoyer moi immédiatement par mail à lc.ensiie@gmail.com les traces `trace-tp-install-vm*-your-login.txt`

Pour redémarrer votre cluster à partir de son état sauvegardé, relancer la même commande:

```
$ pcocc alloc -c 3 pxe-server,dhcp-client,pxe-client
```

Vous devez ensuite terminer seul ce travail et me le rendre par mail 7 jours après.

Plan d'adressage

Vous avez démarré 3 machines virtuelles : les deux premières vm0 et vm1 sur la même image centos 7.4, la troisième vm2 sur un disque vierge

Récupération des informations sur la vm0

Connectez-vous à la machine virtuelle vm0. Utilisez la commande `ip addr show` pour obtenir l'état des interfaces réseaux.

Répondre aux questions de la section correspondante dans le questionnaire.

Récupération des informations sur la vm1

A partir d'une autre fenêtre, connectez-vous au cluster hpc. Puis connectez vous à la machine virtuelle vm1 en prenant une trace :

```
[kevin.dummy@hpc01 ~]$ script -a trace-tp-install-vm1-$(id -un).txt
Le script a débuté, le fichier est trace-tp-install-vm1-kevin.dummy.txt
[kevin.dummy@hpc01 ~]$ pcocc ssh vm1
Last login: Sat Mar  9 12:29:15 2019 from hpc01.c-hpc.pedago.ensieie.fr
```

Utilisez la commande `ip addr show` pour obtenir l'état des interfaces réseaux.

Répondre aux questions de la section correspondante dans le questionnaire.

Récupération des informations sur la vm2

Connectez-vous à la console graphique de la machine virtuelle vm2 :

```
[kevin.dummy@hpc01 ~]$ pcocc display vm2 &
```

Lire avec attention les informations affichées dans cette console.

Attention: Si vous cliquez dans la console graphique, cette fenêtre peut capturer le curseur. Vous avez alors l'impression de l'avoir perdu. Si vous bougez la souris, rien ne se passe. Pour récupérer le curseur, appuyer sur les touches <Control> et <Alt> simultanément.

Répondre aux questions de la section correspondante dans le questionnaire.

Configuration du service DHCP sur la vm0

Installation du logiciel DHCP

Vous allez configurer le nœud vm0 en serveur d'installation. Commencez par le service **dhcp**.

Trouver les paquets contenant le logiciel dhcp . Plusieurs paquets sont disponibles, choisir celui qui se nomme '**Dynamic host configuration protocol software**'.

Installer le paquet. Lister tous les fichiers du paquet. Repérez les fichiers de démarrage et de configuration.

Répondre aux questions de la section correspondante dans le questionnaire.

Configuration de l'interface de management du cluster

Vous devez configurer le service **dhcp** pour qu'il n'écoute que sur l'interface de management (cf schéma). Quel est le nom de cette interface ? Dans quel état est-elle ? Un service réseau peut-il fonctionner sur une interface dans cet état ?

Pour configurer cette interface et qu'elle soit activée au démarrage (boot) du système, il faut écrire un fichier **/etc/sysconfig/network-scripts/ifcfg-<nom-de-l-interface>** . On définit dans ce fichier des variables bash qui seront utilisées par les scripts /sbin/ifup et /sbin/ifdown pour configurer les interfaces.

Consulter le fichier suivant pour connaître les différentes variables à définir, vous pouvez rechercher le pattern network-scripts pour vous amener à la section intéressante :

```
[root@vm0 13:42:01]# less /usr/share/doc/initscripts-*/sysconfig.txt
```

Écrire le fichier de configuration de l'interface de management. Vous avez 5 variables à définir :

- le nom de l'interface physique
- le démarrage automatique au boot système
- son adresse IP
- son type de configuration (ici une configuration statique - elle est décrite par avance)

- son préfixe de réseau

Utilisez le schéma de l'introduction et le fichier **/etc/hosts** pour trouver les valeurs adéquates.

Activez l'interface avec la commande **ifup**. Vérifiez son état avec la commande **ip**. Affichez le fichier de configuration :

```
[root@vm0 16:39:37]# ifup <interface>
[root@vm0 16:39:38]# ip addr show <interface>
[root@vm0 16:39:39]# echo $(grep -v '^#' /etc/sysconfig/network-scripts/ifcfg-<interface>
| sort)
```

Répondre aux questions de la section correspondante dans le questionnaire.

Configuration du serveur dhcp

Lire la page de manuel du démon et de son fichier de configuration **dhcpcd.conf**. Écrire un fichier de configuration comprenant :

- une section "global parameters" définissant dans cet ordre
 - le nom du domaine à mgnt.pcocc,
 - la valeur par défaut à 30 minutes,
 - la valeur maximum de baux à 12h.
- une section subnet pour le réseau de l'interface de management qui permettra de répondre à 10 machines en leur allouant une adresse à partir du rang 100.

Pour retrouver le netmask, utilisez la commande **ipcalc** ou **ip addr show <interface>**.

Le démon **dhcpcd** dispose d'une option **-t** pour vérifier la syntaxe. Utiliser cette option jusqu'à obtention d'un fichier de configuration correct :

```
[root@vm0 12:56:57]# dhcpcd -t
Internet Systems Consortium DHCP Server 4.2.5
Copyright 2004-2013 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Not searching LDAP since ldap-server, ldap-port and ldap-base-dn were not
specified in the config file
[root@vm0 12:57:05]# echo $?
0
[root@vm0 12:57:05]# echo $(grep -v ^# /etc/dhcp/dhcpcd.conf)
```

Avec la commande **systemctl**, autoriser le démarrage automatique du service **dhcpcd**, lancer ce service et vérifier son état. Afficher le contenu du fichier de configuration en une seule ligne via la commande : `echo $(grep -v ^# /etc/dhcp/dhpcd.conf)`

Répondre aux questions de la section correspondante dans le questionnaire.

Configuration du client dhcp vm1

Pour tester le service dhcp, nous allons configurer **sur le nœud vm1** la seconde interface pour qu'elle s'initialise en mode DHCP . Connecter vous au nœud vm1

Consulter le fichier suivant pour connaître les différentes variables à définir, vous pouvez rechercher le pattern network-scripts pour vous amener à la section intéressante :

```
[root@vm1 13:42:01]# less /usr/share/doc/initscripts-*/sysconfig.txt
```

Écrire le fichier de configuration de l'interface de management avec 3 variables à définir :

- le nom de l'interface physique
- le **non démarrage** automatique au boot système
- son type de configuration (ici une configuration dhcp)

Activez l'interface avec la commande **ifup**. Vérifiez son état avec la commande **ip**. Afficher le fichier de configuration en une ligne :

```
[root@vm1 16:39:37]# date ; ifup XXXX ; date
[root@vm1 16:39:38]# ip addr show XXXX
[root@vm1 16:39:39]# d
```

Activer l'interface avec la commande **ifup**. Vérifier avec la commande **ip a** que l'interface a récupéré une adresse IP.

La commande **ifup** lance un démon **dhclient** pour renouveler le bail de interface initialisée en **dhcp**. Rechercher le processus dhclient associé à l'interface que vous avez configuré en dhcp. Retrouvez le nom du fichier dans lequel la commande dhclient a enregistré le bail.

Lire la man page de dhclient.leases. Retrouver la définition des champs renew, rebind, expire. Vérifier que les valeurs observées dans le fichier sont cohérentes.

Répondre aux questions de la section correspondante dans le questionnaire.

Reset de la vm2

Redémarrez la vm2 avec la commande suivante (c'est comme si vous faisiez un power off/on)

```
[kevin.dummy@hpc01 ~]$ date ; pcocc reset vm2
```

Allez voir sur la console de la vm2 ce qui se passe.

Sur la vm0, récupérer les logs du démon dhcpd avec la commande journalctl.

Répondre aux questions de la section correspondante dans le questionnaire.

Analyse des logs du serveur dhcp

Utilisez la commande journalctl pour récupérer les logs du démon dhcpd sur vm0.

Après quel délai sera renouvelé le bail de la vm1 ?

En vous aidant des adresses mac des vm1 et vm2 ainsi que les dates d'action sur ces vms, repérez les messages émis

1. lors de l'allocation initiale de l'adresse IP à la vm1
2. lors de l'allocation initiale de l'adresse IP à la vm2
3. lors du renouvellement du bail de la vm1

Sur le nœud vm0, le démon dhcpd enregistre les baux alloués dans un fichier. Lequel ?

Répondre aux questions de la section correspondante dans le questionnaire.

Mise en place du système de déploiement

Configuration du serveur PXE

Il faut mettre en place le chargeur de démarrage **pxelinux** et le serveur tftp qui permettra d'envoyer ce chargeur au client pxe.

Rechercher les paquets concernant **tftp**. Installer le paquet contenant le serveur tftp. Lister les fichiers contenus dans ce paquet. Repérez le nom du service (au sens systemd).

Avec la commande **systemctl**, autoriser le démarrage automatique du service, lancer ce service et vérifier son état. Repérez les paramètres avec lesquels il est lancé - en particulier le répertoire racine utilisé par le service (nommé par la suite <tftp-root-dir>)

Reste à fournir au service **tftp** les fichiers nécessaires au bootstrap de la machine **vm2**. Ces fichiers sont dans un rpm.

Rechercher avec la commande yum les rpms avec le mot **syslinux**. Plusieurs paquets existent avec des arborescences différentes. Trouver le paquet syslinux préparé pour être utilisé directement avec tftp (yum et repoquery sont vos amis). Installer ce paquet.

Enfin, vous devez créer sous le répertoire racine du service tftp un sous-répertoire pxelinux.cfg pour y accueillir le fichier de configuration du menu pxe :

```
[root@vm0 22:07:11]# mkdir <tftp-root-dir>/pxelinux.cfg
[root@vm0 22:07:12]# cat > <tftp-root-dir>/pxelinux.cfg/default <<
default vesamenu.c32
dhcprompt 1
timeout 600
EOF
```

Le serveur **dhcpd** tel qu'il est configuré n'est pas prêt pour prendre en compte le service **bootp**. Ajoutez dans la section "global parameters" l'option suivante: **allow bootp**.

Dans la section subnet, ajouter une directive next-server avec l'adresse IP du serveur tftp et une directive **filename** avec le path du chargeur de démarrage pxelinux (entre ""')

Vérifier avec la commande `dhcpd -t` la syntaxe du fichier de configuration :

```
[root@vm0 22:39:00]# dhcpd -t
Internet Systems Consortium DHCP Server 4.2.5
Copyright 2004-2013 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Not searching LDAP since ldap-server, ldap-port and ldap-base-dn were not
specified in the config file
[root@vm0 22:39:04]#
```

Redémarrer le service **dhcpcd** et vérifier son état.

Redémarrez la vm2 avec la commande suivante (c'est comme si vous faisiez un power off/on)

```
[kevin.dummy@hpc01 ~]$ pcocc reset vm2
```

Allez voir sur la console de la vm2 ce qui se passe. Vous pouvez faire le reset plusieurs fois pour revoir la séquence sur la console de vm2.

Répondre aux questions de la section correspondante dans le questionnaire.

Mise en place du second étage de démarrage

A ce point, la machine vm2 a juste réussi à charger le premier chargeur de démarrage pxelinux.0. Celui-ci doit lui permettre de passer à la suite.

Pour ce faire, vous allez enrichir le menu '**default**' en ajoutant les directives suivantes à la fin du fichier :

```
[root@vm0 23:36:06]# cat >> <tftp-root-dir>/pxelinux.cfg/default
MENU TITLE ENSIIE Menu
    LABEL centos7_x64
    MENU LABEL CentOS 7_X64 Install
    KERNEL vmlinuz
    APPEND initrd=initrd.img
    LABEL local
    MENU LABEL Boot local hard drive
    LOCALBOOT 0
<Control-D>
[root@vm0 23:36:45]#
```

Ce menu définit deux choix :

1. Démarrer une installation Centos
2. Démarrer sur le disque local

On peut choisir l'une ou l'autre en se déplaçant avec les flèches, lancer une option avec la touche <Enter> , voir les commandes associées en sélectionnant avec la touche TAB.

L'option 1 permet de démarrer l'installation du système CentOS sur le disque.

L'option 2 permet de démarrer sur le disque système une fois qu'il est installé.

Il faut donc récupérer et installer les fichiers **vmlinuz** et **initrd.img** que le client va vouloir utiliser. **vmlinuz** contient le kernel linux compressé. **initrd.img** contient une image initiale ramdisk c'est-à-dire un fichier contenant un système de fichiers linux très basique mais suffisant pour faire une installation complète.

Vous pouvez trouver les deux fichiers sur le site :

http://vault.centos.org/7.4.1708/os/x86_64/images/pxeboot/

Vous pouvez aussi les récupérer dans le répertoire **images/pxeboot/** de l'image ISO de la centos 7.4 que vous pouvez télécharger à partir de ce site :

http://vault.centos.org/7.4.1708/isos/x86_64/

Vous aurez besoin plus tard du contenu de cette image ISO. Elle est déjà téléchargée sur hpc01. A partir de la machine **hpc01**, recopier l'image ISO à partir du répertoire /home/shared/gregoirep/tp-install/ dans le /tmp de votre vm0:

```
[kevin.dummy@hpc01 ~]$ pcooc scp /home/shared/gregoirep/tp-install/CentOS-7-x86_64-Minimal-1708.iso vm0:/tmp
CentOS-7-x86_64-Minimal-1708.iso                               100% 792MB 88.0MB/s   00:09
[kevin.dummy@hpc01 ~]$
```

Une image ISO est un fichier archive qui contient l'image d'un système de fichiers. On peut la monter en lecture comme un disque.

Monter l'image ISO récupérée sur le répertoire /mnt et recopier les 2 fichiers **vmlinuz** et **initrd.img** dans le répertoire racine du service tftp :

```
[root@vm0 10:31:55]# mount -o loop \
/tmp/CentOS-7-x86_64-Minimal-1708.iso /mnt
mount: /dev/loop0 is write-protected, mounting read-only
[root@vm0 11:15:36]# ls -l /mnt/images/pxeboot
total 53041
-rw-r--r-- 1 root root 48434768 Sep  5 2017 initrd.img
-r--r--r-- 1 root root      41 Sep  5 2017 TRANS.TBL
-rwxr-xr-x 1 root root 5877760 Aug 22 2017 vmlinuz
[root@vm0 11:16:24]# cp /mnt/images/pxeboot/{vmlinuz,initrd.img} \
<tftp-root-dir>
```

Redémarrez la vm2 avec la commande suivante (c'est comme si vous faisiez un power off/on)

```
[kevin.dummy@hpc01 ~]$ pcocc reset vm2
```

Allez voir sur la console de la vm2 ce qui se passe. Il faut attendre au moins 40 secondes après la ligne boot: pour voir le menu ENSIIE apparaître. Cliquez dans console de la vm2 et positionnez le menu sur l'entrée Boot local hard drive. Vous pouvez faire le reset plusieurs fois pour revoir la séquence sur la console de vm2. Si vous perdez le contrôle du pointeur, appuyez en même temps sur les touches Ctrl + Alt.

Sur la vm0, avec la commande journalctl, retrouvez les messages concernant in.tftpd et vérifier quels fichiers ont été téléchargés.

Répondre aux questions de la section correspondante dans le questionnaire.

Mise en place de l'environnement Kickstart

Le noyau vmlinuz que le client PXE télécharge et exécute lance le système d'installation automatique Anaconda. Anaconda pilote l'installation automatique à partir d'un fichier "kickstart". Ce fichier est donné en paramètre du noyau dans le menu pxe par la directive **ks=<protocole:pathname-du-fichier-kickstart>**

Le protocole **http** sera utilisé durant toute l'installation, d'abord pour récupérer le fichier kickstart mais aussi pour les paquets à installer. Il faut donc installer un serveur http.

Avec la commande yum, installer le paquet httpd. Activer le démarrage du service httpd au boot du système puis démarrer le service et vérifier son état.

Puis créer dans un répertoire une copie du DVD d'installation Centos 7.4 et vérifier que les deux répertoires sont identiques.

Noter la syntaxe de la commande cp qui permet de copier tous les fichiers de /mnt (même les fichiers qui commencent par un point '.') sans créer un sous-répertoire mnt sous le répertoire destinataire:

```
[root@vm0 16:01:47]# mkdir -p /var/www/html/centos/7.4.1708/
[root@vm0 16:01:53]# cp -rp /mnt/. /var/www/html/centos/7.4.1708/
[root@vm0 23:20:47]# ls -la /mnt /var/www/html/centos/7.4.1708
```

Vérifier avec une requête wget que le service httpd est fonctionnel et que vous pouvez récupérer un paquet via le service http :

```
[root@vm0 23:41:55]# ls -l /var/www/html/centos/7.4.1708/Packages/binutils-2.25.1-31.base.el7.x86_64.rpm
-rw-rw-r-- 1 root root 5665604 10 août 2017
/var/www/html/centos/7.4.1708/Packages/binutils-2.25.1-31.base.el7.x86_64.rpm
[root@vm0 23:42:06]# wget -O /dev/null
http://vm0/centos/7.4.1708/Packages/binutils-2.25.1-31.base.el7.x86_64.rpm
--2019-03-12 23:42:39-- http://vm0/centos/7.4.1708/Packages/binutils-2.25.1-31.base.el7.x86_64.rpm
Résolution de vm0 (vm0)... 10.252.0.1
Connexion vers vm0 (vm0)|10.252.0.1|:80...connecté.
requête HTTP transmise, en attente de la réponse...200 OK
Longueur: 5665604 (5,4M) [application/x-rpm]
Sauvegarde en : «/dev/null»
100%[=====>] 5 665 604 --.-K/s
ds 0,005s
2019-03-12 23:42:39 (1,06 GB/s) - «/dev/null» sauvegardé [5665604/5665604]
```

Il reste à écrire ce fichier de configuration kickstart. Recopier le modèle de fichier kickstart sur la vm0 :

```
[kevin.dummy@hpc01 ~]$ pcooc scp /home/shared/gregoirep/tp-install/ks.cfg vm0:/tmp
ks.cfg 100% 1767 1.6MB/s 00:00
[kevin.dummy@hpc01 ~]$
```

Le fichier est un modèle et donne les directives d'installation et de configuration de la machine cliente. Créer un répertoire accessible du serveur http et recopier le modèle :

```
[root@vm0 12:01:50]# mkdir /var/www/html/centos/kickstart/
[root@vm0 12:01:59]# cp /tmp/ks.cfg /var/www/html/centos/kickstart/
```

Vous avez quelques modifications à y apporter. Elles sont repérées par le pattern ICI_[A-Z]* :

```
[root@vm0 12:08:10]# grep 'ICI_[A-Z]*' /var/www/html/centos/kickstart/ks.cfg
rootpw --iscrypted ICI_MOT_DE_PASSE_DE_ROOT_CHIFFRE
logging --host ICI_ADDR_IP_LOGHOST
url --url="http://ICI_URL/centos/7.4.1708/"
timezone ICI_TIMEZONE_LOCALE
user --name=ICI_VOTRE_LOGIN --password=ICI_MOT_DE_PASSE_DE_VOUS_CHIFFRE
--iscrypted
/usr/bin/logger -n ICI_ADDR_IP_LOGHOST -t "Kickstart" "$msg_fin"
```

Pendant l'installation Kickstart, Anaconda va suivre ces directives :

rootpw --iscrypted : Anaconda initialisera le mot de passe du super-user root.

La commande **openssl** permet de chiffrer une chaîne de caractères en utilisant l'algo md5. Chiffrez le mot de passe de root (C1mdp4root) avec la commande suivante :

```
[root@vm0 12:09:11]# openssl passwd -1 C1mdp4root # passwd for root
$1$PWVBdrDS$ib5q4hsfLS3iKZRFxHvJZ.
```

logging --host : Anaconda va envoyer tous les messages d'installation sur un serveur distant. Utilisez l'adresse de l'interface eth1 de la vm0.

url --url : Anaconda utilisera le service suivant (ftp,nfs, http) pour télécharger les paquets à installer à partir d'un répertoire. Ici on utilisera le serveur **http** sur la vm0. Le path est le path relatif (pour le serveur http) qui contient les éléments d'installation.

timezone : Anaconda initialisera la zone de temps de la machine. Retrouvez la bonne timezone pour Paris (cf TP NTP).

user --name : Anaconda créera un utilisateur avec le même login que votre login sur la machine hpc.

La commande **openssl** permet de chiffrer une chaîne de caractères en utilisant l'algo md5. Chiffrez votre mot de passe (C1mdp4me) avec la commande suivante :

```
[root@vm0 12:09:25]# openssl passwd -1 C1mdp4me # passwd for me
$1$vrefMBXD$oAm1gso/X.WEOZotGjpiV/
```

La dernière ligne est juste une commande exécutée à la fin de l'installation pour envoyer un message au serveur d'installation pour lui signaler la bonne fin de l'installation.

Editer le fichier kickstart pour remplacer les patterns ICI_[A-Z_]* par les bonnes valeurs. Prenez le temps de lire ce fichier dans son intégralité. Faites un diff avec le fichier d'origine.

```
[root@vm0 12:08:10]# vim +'/ICI_[A-Z_]*' /var/www/html/centos/kickstart/ks.cfg
[root@vm0 12:08:10]# diff /var/www/html/centos/kickstart/ks.cfg /tmp/ks.cfg
```

Installer le package pykickstart. Utiliser la commande **ksvalidator** pour vérifier la syntaxe de votre fichier KickStart.

Pour répondre aux questions, vous aurez besoin d'un peu de documentation sur Kickstart

<https://pykickstart.readthedocs.io/en/latest/kickstart-docs.html#kickstart-documentation>

Reste à ajouter dans le menu le paramètre ks au kernel vmlinuz pour lui indiquer comment et où récupérer le fichier kickstart :

```
root@vm0 00:22:54]# vim /var/lib/tftpboot/pxelinux.cfg/default
[root@vm0 00:23:07]# grep APPEND /var/lib/tftpboot/pxelinux.cfg/default
APPEND initrd=initrd.img ks=http://X.Y.Z.T/centos/kickstart/ks.cfg
```

Voilà à quoi ressemble le menu pxe final :

```
[root@vm0 00:23:12]# cat /var/lib/tftpboot/pxelinux.cfg/default
default menu.c32
prompt 0
timeout 180
MENU TITLE ENSIIE Menu
    LABEL centos7_x64
    MENU LABEL CentOS 7_X64 Install
    KERNEL vmlinuz
    APPEND initrd=initrd.img ks=http://X.Y.Z.T/centos/kickstart/ks.cfg
    LABEL local
    MENU LABEL Boot local hard drive
    LOCALBOOT 0
[root@vm0 00:23:22]#
```

Modifier la configuration du démon rsyslog sur **vm0** pour le forcer à écouter les messages en tcp et udp. Il pourra ainsi recevoir les messages d'anaconda. Avec la commande `systemctl`, redémarrer le démon **rsyslogd** sur le nœud **vm0** et vérifier son état.

```
[root@vm0 12:41:39]# vim /etc/rsyslog.conf
[root@vm0 12:44:58]# systemctl restart rsyslog
[root@vm0 12:45:00]# systemctl status rsyslog
â rsyslog.service - System Logging Service
   Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled;
   vendor preset: enabled)
   Active: active (running) since Tue 2019-03-12 12:45:00 CET; 9s ago
     Docs: man:rsyslogd(8)
           http://www.rsyslog.com/doc/
  Main PID: 1882 (rsyslogd)
    CGroup: /system.slice/rsyslog.service
            ââ1882 /usr/sbin/rsyslogd -n

Mar 12 12:45:00 vm0.pcocc systemd[1]: Starting System Logging Service...
Mar 12 12:45:00 vm0.pcocc rsyslogd[1882]: [origin software="rsyslogd"
swVersion="8.24.0" x-pid="1882" x-info="http://www.rsyslog.com"] start
Mar 12 12:45:00 vm0.pcocc systemd[1]: Started System Logging Service.
[root@vm0 12:45:10]#
```


Il est recommandé de filmer avec votre smartphone la console graphique de **vm2** pour pouvoir lire plus facilement les messages pendant l'installation. Sur la machine **hpc01**, redémarrez le nœud **vm2** avec la commande : **pcocc reset vm2**.

Redémarrez la vm2 avec la commande suivante (c'est comme si vous faisiez un power off/on)

```
[kevin.dummy@hpc01 ~]$ pcocc reset vm2
```

Allez voir sur la console de la vm2 ce qui se passe. Vous pouvez aussi suivre les échanges sur vm0 avec la commande `journalctl -f`

Laisser l'installation se dérouler sans rien toucher. Sur la console graphique (**display**) du nœud **vm2**, regardez les messages qui apparaissent et notez ce qui se passe au début de l'installation pendant la phase PXE et à la fin après le reboot final.

Répondre aux questions de la section correspondante dans le questionnaire.

Mise en place du boot final sur disque pour la vm2

Recopier sur la vm0 le script `/home/shared/gregoirep/tp-install/set-boot-disk` :

```
[kevin.dummy@hpc01 ~]$ pcocc scp /home/shared/gregoirep/tp-install/set-boot-disk vm0:/tmp
set-boot-disk                                100% 698 584.7KB/s 00:00
[kevin.dummy@hpc01 ~]$
```

Installer ce script en tant que root dans `/sbin` avec les droits 755 :

```
[root@vm0 09:53:33]# cp /tmp/set-boot-disk /sbin/
[root@vm0 10:24:49]# chmod 755 /sbin/set-boot-disk
```

Modifier la configuration de rsyslog pour que script `/bin/set-boot-disk` soit exécuté à chaque fois que rsyslog reçoit un message de fin d'installation, tel qu'il est envoyé par la commande `logger` dans le fichier `kickstart`. Insérez ces 2 lignes après la ligne **### RULES ###** :

```
$template boot_on_disk,"%msg%"
:msg, contains, "Kickstart install completed on" ^/sbin/set-boot-disk;boot_on_disk
```

Redémarrez le service rsyslog sur la vm0 et vérifiez son état.

```
[root@vm0 10:25:13]# vim /etc/rsyslog.conf
[root@vm0 11:26:44]# grep -A2 RULES /etc/rsyslog.conf
#### RULES ####
$template boot_on_disk,"%msg%"
:msg, contains, "Kickstart install completed on" ^/sbin/set-boot-disk;boot_on_disk
[root@vm0 11:27:13]# systemctl restart rsyslog
[root@vm0 11:27:53]# systemctl status rsyslog
â rsyslog.service - System Logging Service
   Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled; vendor preset:
   enabled)
   Active: active (running) since Wed 2019-03-13 11:27:53 CET; 9s ago
     Docs: man:rsyslogd(8)
           http://www.rsyslog.com/doc/
  Main PID: 2748 (rsyslogd)
    CGroup: /system.slice/rsyslog.service
            ââ2748 /usr/sbin/rsyslogd -n

Mar 13 11:27:53 vm0.pcocc systemd[1]: Starting System Logging Service...
Mar 13 11:27:53 vm0.pcocc rsyslogd[2748]: [origin software="rsyslogd"
swVersion="8.24.0" x-pid="2748" x-info="http://www.rsyslog.com"] start
Mar 13 11:27:53 vm0.pcocc systemd[1]: Started System Logging Service.
```

Redémarrez la vm2 avec la commande suivante (c'est comme si vous faisiez un power off/on)

```
[kevin.dummy@hpc01 ~]$ pcocc reset vm2
```

Allez voir sur la console de la vm2 ce qui se passe. Laissez l'installation se dérouler sans rien toucher. Quand la vm2 a terminé son démarrage sur le disque, connectez vous dans le display de vm2 en tant que vous avec le mot de passe défini précédemment. Prenez un snapshot de la console.

Retrouver dans le fichier de baux dhcp sur la vm0 l'adresse IP alloué à la vm2. A partir de la vm0, connectez vous en ssh sur la vm2 avec votre login.

Étudiez le script /sbin/set-boot-disk pour comprendre son fonctionnement.

Répondre aux questions de la section correspondante dans le questionnaire.

Envoyer les traces `trace-tp-install-vm*-your-login.txt` et le snapshot de la console graphique à `lc.ensiie@gmail.com`

