iCBD Installation Protocol DRAFT

Version 1.0.1 - Last Updated 30 Jan 2018

Luis Silva - `lmt.silva (at) campus.fct.unl.pt` | `luis.tsilva (at) reditus.pt`

In this document, we will detail all the steps needed to entirely install from scratch and start the iCBD Management Platform.

Pre-Requisites

What is needed:

- 3 x *CentOS 7* Minimum Install VM
  - 2 Hard Drives (the extra for *BTRFS*)
  - 1 or more NICs (Depending on the VM)
- iCBD install files for each VM
- Some iCBD VM images

Attention - CentOS 7 Kernel Version! The Kernel `3.10.0-693.5.2.el7.x86_64` on CentOS 7 has manifested a problem with a core component of the *coreutils* tool command, the `cp` when used with option `--reflink=always`. To circumvent the issue is advised to use an older Kernel, such as `3.10.0-514.2.2.el7.x86_64` as we confirmed is working. This until Red Hat releases a new kernel with the bug fix.

Introduction

This tutorial assumes a fresh minimal install of a CentOS 7 Operating System. The installation procedure will cover all configurations needed for the implementation of VMs that will take a role in the platform. Some of the settings are specific for one of the roles, in this case, there will be a note in the step description.

iCBD Roles

The iCBD Management Platform consists of a minimum of three VM's, but for a more complex typology, we can mix in some cache servers and some clients. So we can have the following roles:

- *iCBD-imgs* - Primary repository of VM images and facilitator of the administration process
- *iCBD-rw* - Provides read/write space to the iCBD clients
- *iCBD-home* - Hosting of Home accounts to be used by iCBD clients
- *iCBD-cache* - Hosting of VM images closest to the clients

- *iCBD-Client* - A VM shell that don't have a hard disk and will boot from network

iCBD Networks

Also, there is the need to define multiple networks. Here, as we are using the VMware platform, there is the ability to design a Distributed VSwitch with various Port Groups, each one symbolising an individual network. The networks are:

On the iCBD-DSwitch (This distributed virtual switch only works inside the cluster)

- iCBD-Net
- iCBD-Adm-Net
- iCBD-Rep
- iCBD-CacheXX-Net

On the DI-DSwitch (Outside access to DI networks and Internet)

- DMZ-PRIV-DI
- DMZ-PUB-DI
- R-ENSINO-PRIV-DI

In the next table is showed the characteristics of each VM given its role. These properties mirror what is implemented in the Cluster at *DI - FCT NOVA*. Then we present two tables: one with the sizes used for the hard drives, and the other including the networks for the NICs of each VM.

VM Hardware by Role

	iCBD-imgs	iCBD-rw	iCBD-home	iCBD-CacheXX
CPUs (cores)	8	4	4	4
RAM (GB)	32	8	8	32
Hard Drives	2	2	2	2
NICs	3	1	1	2

Hard Drives by Role

	iCBD-imgs	iCBD-rw	iCBD-home	iCBD-CacheXX
Hard Drive 1 (Root FS)	16 GB	16 GB	16 GB	16 GB
Hard Drive 2 (BTRFS)	600 GB	300 GB	100 GB	600 GB

NICs by Role

	iCBD-imgs	iCBD-rw	iCBD-home	iCBD-CacheXX
NIC 1	DMZ-PRIV-DI (Internet)	iCBD-Net	iCBD-Net	iCBD-Net
NIC 2	iCBD-Net	X	X	iCBD-CacheXX-Net
NIC 3	iCBD-Adm-Net	X	X	X

First Step

Let's start:

The first thing we need is a vanilla VM with *CentOS 7* minimal install. This VM will be our basis. Many of the procedures that we will need to implement are more conveniently executed from a terminal in your machine, so probably is a good idea to configure an *SSH* access to the VM. Anyway, you will need to *SSH* to the VM in the future, so it's better to start this way.

Setup a static IP and configure SHH

Setup a static IP address.

Depending on the machine it may be that there is more than one network card installed. In the case of the `iCBD-imgs` this is true. So, I leave here the configuration prepared in this machine.

The VM `iCBD-imgs` as 3 NICs :

- NIC1
    - Port Group: DMZ-PRIV-DI
    - DVSwitch: DSwitch1 (DI-FCT Networks)
    - Used: Outside access
    - Config File - `vi /etc/sysconfig/network-scripts/INTERFACE_NAME`

```
HWADDR=00:50:56:96:A3:52 # Interface MAC Address
TYPE=Ethernet
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=no
IPV6_FAILURE_FATAL=no
NAME=ens192
ONBOOT=yes
IPADDR=10.170.137.98    # External IP
NETMASK=255.255.255.0
NM_CONTROLLED=no        # Doesn't let the Network Manager change the
config
PREFIX=24
GATEWAY=10.170.137.254  # Gateway for the .137 network
DNS1=10.130.10.25       # FCT DNS1
DNS2=10.130.10.26       # FCT DNS1
DOMAIN=ensino.priv.di.fct.unl.pt
```

- NIC2
  - Port Group: iCBD-Net
  - DVSwitch: iCBD-DSwitch
  - Used: Main internal network. Platform clients connect were.
  - Config File - `vi /etc/sysconfig/network-scripts/INTERFACE_NAME`

  This NIC will be connected to a bridge, so this is the config for the interface, and then is shown the config for the bridge.

```
HWADDR=00:50:56:96:2E:9C
TYPE=Ethernet
#BOOTPROTO=none
#DEFROUTE=yes
#IPV4_FAILURE_FATAL=yes
#IPV6INIT=no
#IPV6_FAILURE_FATAL=no
NAME=ens224
ONBOOT=yes
#IPADDR=10.0.2.251
#PREFIX=24
BRIDGE=br0
#NETMASK=255.255.255.0
#NM_CONTROLLED=no
ZONE=internal
```

  The Bridge config:

```
DEVICE=br0
STP=yes
TYPE=Brige
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
NAME="Brige br0"
ONBOOT=yes
BRIDGIN_OPTS=priority=32768
IPADDR=10.0.2.251
PREFIX=24
ZONE=internal
```

- NIC3
  - Port Group: iCBD-Adm-Net
  - DVSwitch: Standard Switch
  - Used: Internal network for the administration machines
  - Config File - `vi /etc/sysconfig/network-scripts/INTERFACE_NAME`

```
HWADDR=00:50:56:96:74:85
TYPE=Ethernet
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=no
IPV6_FAILURE_FATAL=no
NAME=ens161
ONBOOT=yes
IPADDR=10.0.3.1
NETMASK=255.255.255.128
NM_CONTROLLED=no
PREFIX=24
```

SSH access without password

A configuration with password-less *SSH* access it's highly recommended since you will be connecting to the different servers a lot. A lot!

Still, the next step for your own machine is optional. But since in a later moment, it will be necessary to configure this between the servers and the physical machines the instructions are already here.

For some reference take a look at the next table. Each row represents a particular VM, and the columns indicate the VM keys that should be present in the `~/.ssh/authorized_keys`.

	iCBD-imgs	iCBD-rw	iCBD-home	iCBD-CacheXX	Your Machine
iCBD-imgs		√	√	√	√
iCBD-rw	√		√	√	√
iCBD-home	√	√		√	√
iCBD-CacheXX	√	√	√	√, other caches	√

To generate an *RSA key* pair to work with version 2 of the *SSH* protocol, type the following command at a shell prompt: `ssh-keygen -t rsa`

Transfer your public key to `~/.ssh/authorized_keys`

Need the command ? `cat ~/.ssh/id_rsa.pub | ssh user@server "mkdir -p ~/.ssh && cat >> ~/.ssh/authorized_keys"`

Note: If you are cloning the main VM as a template for the other services, don't forget to create a new *RSA key* and add it to the remaining servers.

Install packages

Now we need to start building the environment with all the necessary tools to run iCBD.

So first run `yum update`, to make sure that all already installed packages are up to date.

Next we need to install all of these packages:

```
yum install net-tools
yum install hdparm
yum install Xorg
yum install gdm
yum install qemu-kvm
yum install virt-manager
yum install gcc
yum install kernel-headers
yum install kernel-devel
yum install epel-release
yum install htop
yum install httpd
yum install ntp
yum install firefox
yum install open-vm-tools
yum install open-vm-tools-desktop
yum install exportfs
yum install vnc
```

```
yum install xinetd
yum install tigervnc-server-applet

yum groupinstall fonts
yum groupinstall "X window system"

yum install kde-workspace
yum install ksysguard
yum install tftp
yum install tftp-server
yum install target-cli **
yum install iscsi-initiator-utils
yum install scsi-target-utils
yum install firewall-config
yum install tcpdump
yum install libvirt
yum install qemu
yum install rsync
yum install php
yum install wget
yum install bind-utils
yum install spice-protocol
yum install spice-server
yum install iotop
yum install iftop
yum install libguestfs
yum install libguestfs-tools
yum install traceroute
yum install strace
yum install nmap
yum install whois
yum install ed
yum install sysstat
yum install rsh
yum install pure-ftpd
```

Setup a graphical environment

It's easier to perform much of the day to day operations if we have a graphical user interface. And given the today's available resources for a development environment, it helps. If you are setting up a production server, then it should be done with scripts..

To activate *KDE* just run `systemctl set-default graphical.target`

In the next restart, you will have a graphical interface instead of a console.

Update date & time

Make sure the time & date are updated

```
systemctl enable ntpd.service
ntpdate pool.ntp.org
systemctl start ntpd.service
```

and to confirm running `date` and compare with our machine.

Disable SELinux

The Security-Enhanced Linux functionality enters into conflict with many components of the iCBD platform, this way there is the need for disabling it. `vi /etc/sysconfig/selinux`

Check if the flag is set to `SELINUX=enforcing`, if so change it either to `permissive` or `disabled` [1]

Ending Step One

Do a `reboot`, just to load everything up, including KDE.

Second Step

Now we start to lay the groundwork for the *iCBD* directories and much-needed mounts. In this sense, we need to start working with the *BTRFS* File System.

Format a second hard drive with BTRFS

You can check the available disks with `ls -l /dev | grep sd`

Let's assume that you have an empty disk ready to being formatted with *BTRFS* underneath `/dev/sdb`

To format the disk with *BTRFS* do a `mkfs.btrfs /dev/sdb`

The above command makes use of the whole disk. But the `mkfs.btrfs` tool as multiple configurations and you can first create some partitions or even multiple disks in a *RAID* configuration and then format them in *BTRFS*. But for simplicity sake (and even taking into account some compartmentalisation issues) let's use the whole disk.

For some follow up on the matter of structuring the disks and multiple partitions there are numerous articles and tutorials on the web. [2]

Now you should see that there is a BTRFS file system in the OS.

Use `btrfs filesystem show` to make sure.

Third Step

Now the fun stuff. Mounts!

Caution: From this point on, it is necessary to pay close attention to the mounts, double checking them, as it is enough to fail one and the whole platform may not work.

Mounting the base for the iCBD BTRFS volume

The iCBD needs a *"couple"* of mount points, but every one of them will be under `/var/lib/`. Those will differ from server to server, given the task that it will perform. But this step is universal to every machine.

Let's create a temporary mount for the *BTRFS* disk we created earlier: Execute `mkdir /mnt/btrfs` and then `mount /dev/sdb /mnt/btrfs`.

As we are going to mount the root of the *BTRFS* file system under `/var/lib` there is the need to copy all files and directories first.

Create a sub-volume that will house the *lib* files `btrfs subv create /mnt/btrfs/Lib`, then copy everything to the new sub-volume `cp -a /var/lib/. /mnt/btrfs/Lib/`

Next mount the sub-volume `mount -o subvol=Lib /dev/sdb /var/lib` and check if the mount was sucessful `ls -lah /var/lib/`

Case it looks ok, edit the `fstab` file to make this change permanent: `vi /etc/fstab` Add the line `/dev/sdb /var/lib btrfs subvol=Lib 0 0`

(The arguments are separeted by a tab and the numbers by a space

`/dev/sdb[TAB]/var/lib[TAB]btrfs[TAB]subvol=Lib[TAB]0[SPACE]0`)

and `reboot`.

Fourth Step - iCBD-imgs

More sub-volumes!

These next steps are specific to the *iCBD-imgs VM*, that takes care of the administrations of the images, but also possesses the capability to serve them to the clients. In a future point, we will see the details for the other kind of roles.

Creating the iCBD sub-volumes

Let's create all the following sub-volumes:

```
btrfs subv create /var/lib/icbd
btrfs subv create /var/lib/icbd/.snap
btrfs subv create /var/lib/icbd/shared-vms
mkdir /var/lib/icbd/mounts
btrfs subv create /var/lib/icbd/mounts/vmware
btrfs subv create /var/lib/icbd/mounts/livirt
btrfs subv create /var/lib/icbd/mounts/tftpboot
btrfs subv create /var/lib/icbd/nfs_home
btrfs subv create /var/lib/icbd/nfs_root
btrfs subv create /var/lib/icbd/rw
btrfs subv create /var/lib/icbd/iso
btrfs subv create /var/lib/icbd/tmp
btrfs subv create /var/lib/icbd/icbd
```

The mouting of all this sub-volumes will come later.

Fifth Step - iCBD-imgs

In this installation package there should be a `iCBD-imgs_2017-11-17_bkk.tgz` file. This file is a backup of iCBD-Core and can be used to install.

Transfer the file to the VM, you can use a SSH feature for this:

`scp iCBD-imgs_2017-11-17_bkk.tgz user@host:/var/lib/icbd`

Navigate to `/var/lib/icbd/` on the VM and unzip the file directly to the folder `tar -xvzf iCBD-imgs_2017-11-17_bkk.tgz`.

After this, you can clean up the folder by removing the file: `rm iCBD-imgs_2017-11-17_bkk.tgz`.

Atention - This backup does not contain the folder `/var/lib/icbd/mounts/tftpboot`

Now the remaining mounts I promised. Edit the `fstab` and add this lines:

```
/var/lib/icbd/mounts/vmware        /var/lib/vmware    none    rbind    0 0

/var/lib/icbd/mounts/etc/iscsi    /etc/iscsi    none    rbind    0 0
/var/lib/icbd/mounts/etc/tgt        /etc/tgt    none    rbind    0 0
/var/lib/icbd/mounts/etc/httpd    /etc/httpd    none    rbind    0 0
/var/lib/icbd/mounts/etc/xinetd.d    /etc/xinetd.d    none    rbind    0 0
/var/lib/icbd/mounts/tftpboot        /var/lib/tftpboot    none    rbind
0 0

/var/lib/icbd/mounts/etc/hosts    /etc/hosts     none    bind    0 0
/var/lib/icbd/mounts/etc/exports    /etc/exports    none    bind    0 0
/var/lib/icbd/mounts/etc/dnsmasq.conf    /etc/dnsmasq.conf    none    bind
   0 0

/var/lib/icbd/icbd    /var/lib/tftpboot/icbd        none    rbind    0 0

/var/lib/icbd/bin    /var/lib/icbd/exports/bin    none    rbind    0 0
/var/lib/icbd/include    /var/lib/icbd/exports/include    none    rbind
0 0
/var/lib/icbd/client    /var/lib/icbd/exports/client    none    rbind    0
0

/var/lib/icbd/icbd    /var/lib/icbd/exports/icbd    none    rbind    0 0
/var/lib/icbd/tmp    /var/lib/icbd/exports/tmp    none    rbind    0 0
/var/lib/icbd/iso    /var/lib/icbd/exports/iso    none    rbind    0 0

/var/lib/icbd/shared-vms    /var/lib/icbd/exports/shared-vms    none
rbind    0 0
/var/lib/icbd/nfs_home    /var/lib/icbd/exports/nfs_home    none    rbind
0 0
/var/lib/icbd/nfs_root    /var/lib/icbd/exports/nfs_root    none    rbind
0 0
/var/lib/libvirt/images    /var/lib/icbd/exports/images    none    rbind
0 0
```

Save and `reboot`

Sixth Step - iCBD-imgs

Update the hosts file

Update `hosts` file. Remember, if any changes here done to this file before the last group of mounts this is now without effect. There is a sample `hosts` file in the install package. This server will serve as DHCP it's important that the IP's of the architecture are well defined.

Install the VMware Player.

Also, since we are working with virtualization, maby it's a good time to install one hypervisor. Go to the VMware site and download VMware Workstation 12.

If there is the need for some help in the installation process, check this link to the VMware KB.

Add line to sysctl

`vi /etc/sysctl.conf` and add the line `net.ipv4.ip_forward=1`

Then exe the command `sysctl net.ipv4.ip_forward=1`

Activate NAT

Add direct rules to firewalld. Add the `–permanent` option to keep these rules across restarts.

```
firewall-cmd --direct --add-rule ipv4 nat POSTROUTING 0 -o eth_ext -j
MASQUERADE
firewall-cmd --direct --add-rule ipv4 filter FORWARD 0 -i eth_int -o
eth_ext -j ACCEPT
firewall-cmd --direct --add-rule ipv4 filter FORWARD 0 -i eth_ext -o
eth_int -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Source: https://www.centos.org/forums/viewtopic.php?t=53819

Firewall configuration

Open the firewall configuration GUI.

We need to configure the firewall to let a bunch of services let through. The profile we are going to use is the one named `internal`.

Then in this profile on the tab *Services* tick the following names:

```
dhcp
dhcpv6-client
dns
ftp
http
https
iscsi-target
mdns
mountd
nfs
ntp
rpc-bind
rsyncd
samba
samba-client
squid
ssh
tftp
tftp-client
```

And in the *Masquerading* tab tick the showed box.

Lastly in the `options` dropdown select the option `Runtime to Permanent`, this way the changes are saved.

Sixth Step - iCBD-imgs

We are close to the end of the configurations on the *iCBD-imgs* server!

Launch the need services

There are some key services that need to be running in order to the platform work.

Make sure that these services are successfully running:

```
systemctl start vmware
systemctl start vmware-workstation-server
systemctl start libvirtd
systemctl start dnsmasq
systemctl start tftp
systemctl start tgtd
systemctl start nfs-server
systemctl start httpd
systemctl start ntpd
```

Check with `systemctl status -l [service_name]`

Don't forget to enable them for when a restart occur:

```
systemctl enable vmware-workstation-server
systemctl enable libvirtd
systemctl enable dnsmasq
systemctl enable tftp
systemctl enable tgtd
systemctl enable nfs-server
systemctl enable httpd
systemctl enable ntpd
```

Other Roles Services

iCBD-rw

iCBD-rw sub volumes

```
btrfs subv create /var/lib/Home
btrfs subv create /var/lib/icbd
btrfs subv create /var/lib/icbd/.snap
btrfs subv create /var/lib/icbd/nfs_home
btrfs subv create /var/lib/icbd/nfs_root
btrfs subv create /var/lib/icbd/nfs_rw
btrfs subv create /var/lib/icbd/nfs_tmp
btrfs subv create /var/lib/icbd/rw
mkdir /var/lib/icbd/mounts
btrfs subv create /var/lib/icbd/mounts/tftpboot
```

iCBD-rw Services

```
systemctl start tgtd
systemctl start nfs-server
```

iCBD-rw fstab

```
/dev/sdb        /var/lib        btrfs   subvol=Lib      0 0
/dev/sdb        /home   btrfs   subvol=Home     0 0

/var/lib/icbd/nfs_home  /var/lib/icbd/exports/nfs_home  none    rbind   0 0
/var/lib/icbd/nfs_root  /var/lib/icbd/exports/nfs_root  none    rbind   0 0
/var/lib/icbd/rw        /var/lib/icbd/exports/rw        none    rbind   0 0
/var/lib/icbd/mounts/etc/hosts  /etc/hosts      none    bind    0 0
/var/lib/icbd/mounts/etc/exports        /etc/exports    none    bind    0 0
/var/lib/icbd/mounts/tftpboot   /var/lib/tftpboot       none    rbind   0 0
/var/lib/icbd/mounts/etc/tgt    /etc/tgt        none    rbind   0 0
/var/lib/icbd/mounts/etc/httpd  /etc/httpd      none    rbind   0 0
/var/lib/icbd/mounts/etc/tgt/macs.d     /var/lib/icbd/exports/macs.d
none    rbind   0 0
```

iCBD-home

iCBD-home sub volumes

```
btrfs subv create /var/lib/icbd
btrfs subv create /var/lib/icbd/.snap
btrfs subv create /var/lib/icbd/nfs_home
btrfs subv create /var/lib/icbd/nfs_root
btrfs subv create /var/lib/icbd/exports/nfs_home
btrfs subv create /var/lib/icbd/exports/nfs_root
```

iCBD-home fstab

```
/dev/sdb        /var/lib        btrfs   subvol=Lib      0 0
/var/lib/icbd/mounts/etc/exports        /etc/exports    none    bind    0 0
```

iCBD-home Services

```
systemctl start nfs-server
```

iCBD-Cache

In the file `/etc/hosts` there is the need to change one line. Where is

`10.0.2.251 imgs.icbd.local boot.icbd.local root.icbd.local adm-s.icbd.local`

now we should have two lines:

`10.0.2.251 imgs.icbd.local`

`10.1.2.251 boot.icbd.local root.icbd.local adm-s.icbd.local`

The second IP is the subnet to be used on the second NIC of the cache server, and only to communicate with clients.

iCBD-cache sub volumes

```
btrfs subv create /var/lib/icbd
btrfs subv create /var/lib/icbd/.snap
btrfs subv create /var/lib/icbd/shared-vms
mkdir /var/lib/icbd/mounts
btrfs subv create /var/lib/icbd/mounts/vmware
btrfs subv create /var/lib/icbd/mounts/livirt
btrfs subv create /var/lib/icbd/mounts/tftpboot
btrfs subv create /var/lib/icbd/nfs_home
btrfs subv create /var/lib/icbd/nfs_root
btrfs subv create /var/lib/icbd/rw
btrfs subv create /var/lib/icbd/iso
btrfs subv create /var/lib/icbd/tmp
btrfs subv create /var/lib/icbd/icbd
```

iCBD-cache fstab

```
/dev/sdb        /var/lib        btrfs   subvol=Lib      0 0

/var/lib/icbd/mounts/vmware             /var/lib/vmware none    rbind   0 0

/var/lib/icbd/mounts/etc/iscsi  /etc/iscsi      none    rbind   0 0
/var/lib/icbd/mounts/etc/tgt            /etc/tgt        none    rbind   0 0
/var/lib/icbd/mounts/etc/httpd  /etc/httpd      none    rbind   0 0
/var/lib/icbd/mounts/etc/xinetd.d       /etc/xinetd.d   none    rbind   0 0
/var/lib/icbd/mounts/tftpboot           /var/lib/tftpboot       none
rbind   0 0

/var/lib/icbd/mounts/etc/hosts  /etc/hosts      none    bind    0 0
/var/lib/icbd/mounts/etc/exports        /etc/exports    none    bind    0 0
/var/lib/icbd/mounts/etc/dnsmasq.conf   /etc/dnsmasq.conf       none
bind    0 0


/var/lib/icbd/icbd      /var/lib/tftpboot/icbd          none    rbind   0 0

/var/lib/icbd/bin       /var/lib/icbd/exports/bin       none    rbind   0 0
/var/lib/icbd/include   /var/lib/icbd/exports/include   none    rbind   0 0
/var/lib/icbd/client    /var/lib/icbd/exports/client    none    rbind   0 0

/var/lib/icbd/icbd      /var/lib/icbd/exports/icbd      none    rbind   0 0
/var/lib/icbd/tmp       /var/lib/icbd/exports/tmp       none    rbind   0 0
/var/lib/icbd/iso       /var/lib/icbd/exports/iso       none    rbind   0 0

/var/lib/icbd/shared-vms        /var/lib/icbd/exports/shared-vms
none    rbind   0 0
/var/lib/icbd/nfs_home  /var/lib/icbd/exports/nfs_home  none    rbind   0 0
/var/lib/icbd/nfs_root  /var/lib/icbd/exports/nfs_root  none    rbind   0 0
/var/lib/libvirt/images /var/lib/icbd/exports/images    none    rbind   0 0

home.icbd.local:/nfs_home       /var/lib/icbd/nfs_home  nfs4    _netdev,rw
     0 0
home.icbd.local:/nfs_root       /var/lib/icbd/nfs_root  nfs4    _netdev,rw
     0 0
data.icbd.local:/rw     /var/lib/icbd/rw        nfs4    _netdev,rw      0 0
data.icbd.local:/rw     /var/lib/icbd/exports/rw        nfs4    _netdev,rw
     0 0
data.icbd.local:/macs.d /etc/tgt/macs.d nfs4    _netdev,rw      0 0
```

iCBD-cache Services

```
systemctl start libvirtd
systemctl start dnsmasq
systemctl start tftp
systemctl start tgtd
systemctl start nfs-server
systemctl start httpd
systemctl start ntpd
```

Change Log

2017-11-21 — Version 0.0.1 — Creation of this document.

2017-12-01 — Version 0.0.1 — Created the base structure for the description of the installation steps.

2017-12-10 — Version 0.0.1 — Added much of the content for the installation of the three main VMs. Some organisation is needed!

2017-12-12 — Version 0.0.1 — Step One formatted and updated.

2017-12-16 — Version 0.0.1 — Reference added.

2017-12-18 — Version 0.0.1 — Step Two edited.

2018-01-12 — Version 0.0.1 — Every step was edited

2018-01-14 — Version 1.0.0 — All steps tested in the installation of one physical cache server

2018-01-30 — Version 1.0.1 — Some clarifications on the introduction and on the cache server.

References

CentOS 7 Documentation - Enable or Disable SELinux

HowToForge - A Beginners Guide To btrfs