

# How to create an ECS system for test purposes

For test purposes Dell/EMC provides in the so-called *Community* an ECS software version with an included capacity license with the size of 1 TB. This software is also called ECS-CE (CE is short for *Community Edition*). You have 2 possibilities:

- You can download 'only' the ECS software and install it on a compatible Linux server.
 

Advantage	You use a Linux system of your choice and capabilities.
Caveat	The installation will be more complicated.
  
- You can download a preconfigured VMware OVA file and configure the VM via pre-installed scripts.
 

Advantage	The installation of the ECS software will obviously be more simple.
Caveat	The Linux system (Centos 7) is only installed to a minimal level. That means that some programs/tools are not available by default.

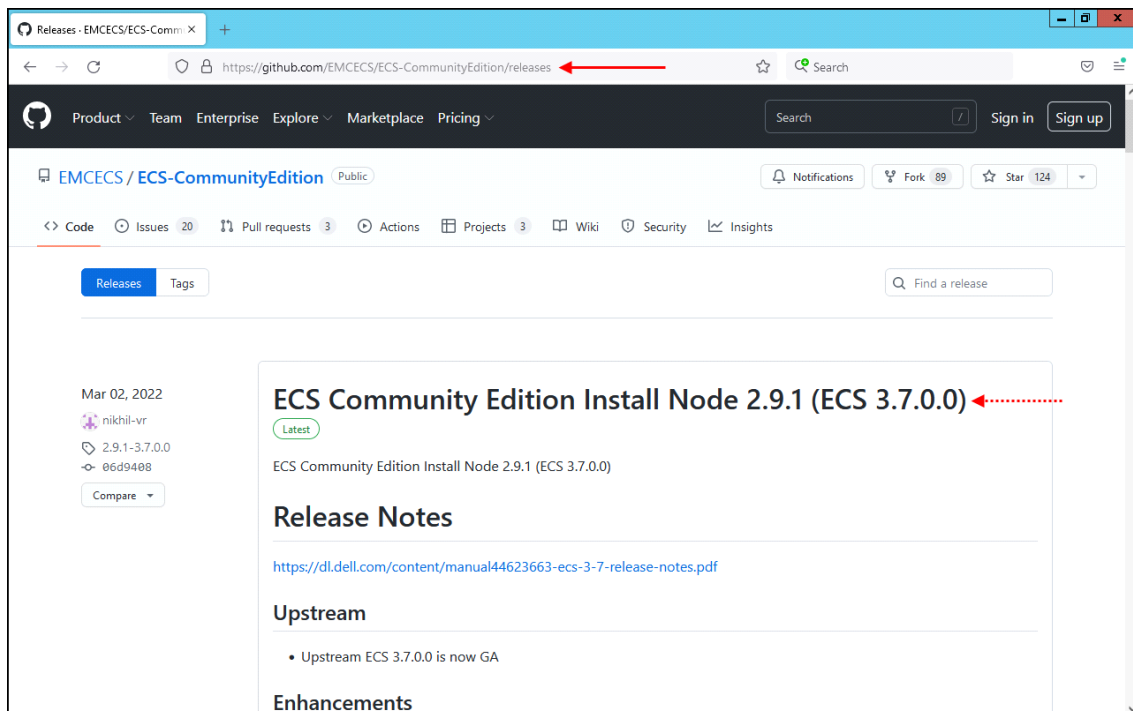
Although limited to a certain extent, I want to show you the second method - it will most likely achieve the goal pretty fast.

## I. Downloading software and documents

All necessary information (software & documents) are store on a GitHub server. In the meantime, the link from Dell's web system has been removed - consequently you must access the server directly.

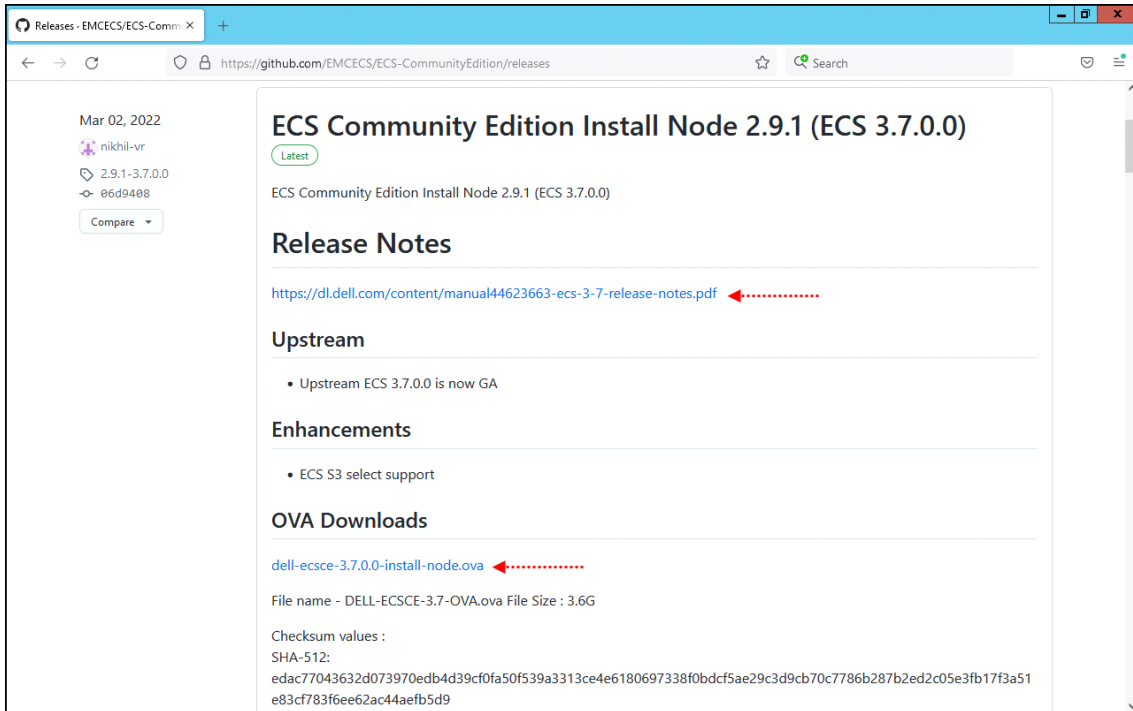
This is the download address:

<https://github.com/EMCECS/ECS-CommunityEdition/releases>

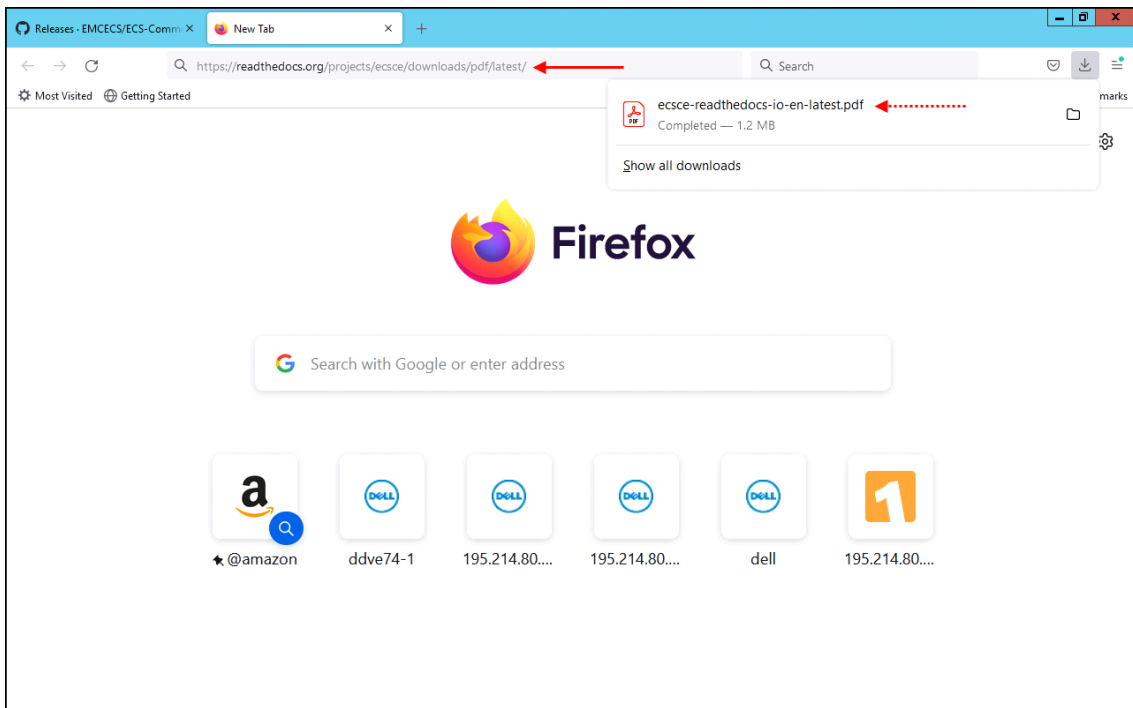


Comparing the version number with the one in Dell EMC's web system shows that this number points to the very last/current version. This is at least the case for the date 07/01/2022.

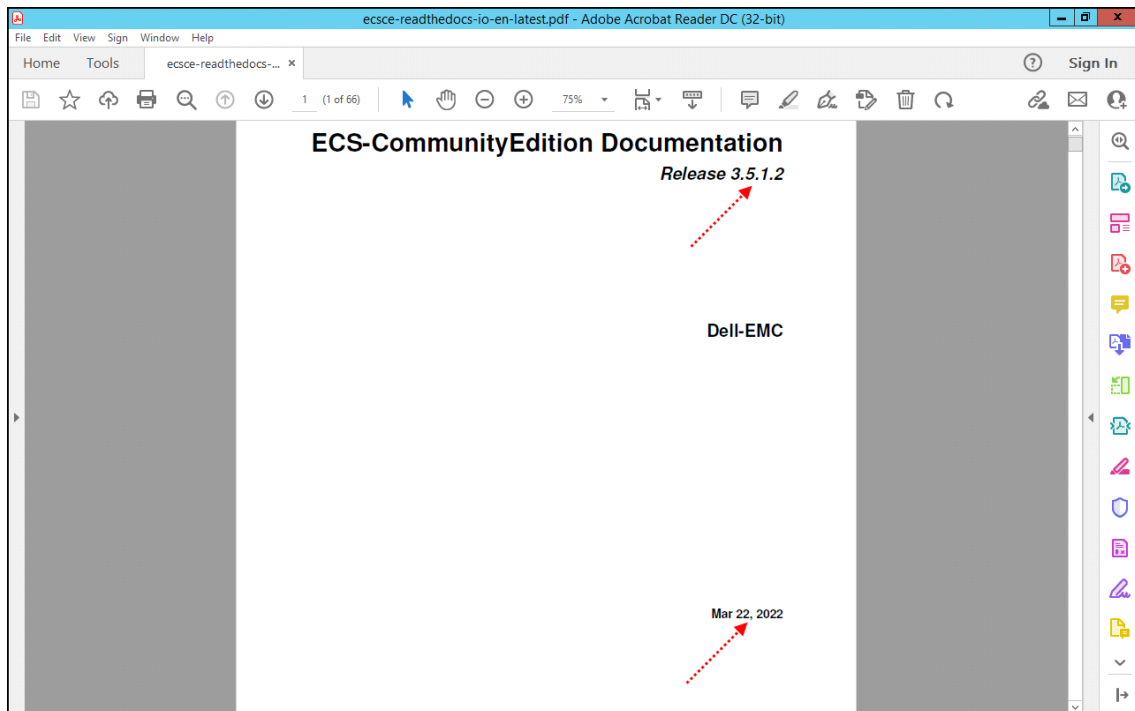
This link points to the documentation for the same version (3.7.0.0):



However, after downloading the document ...

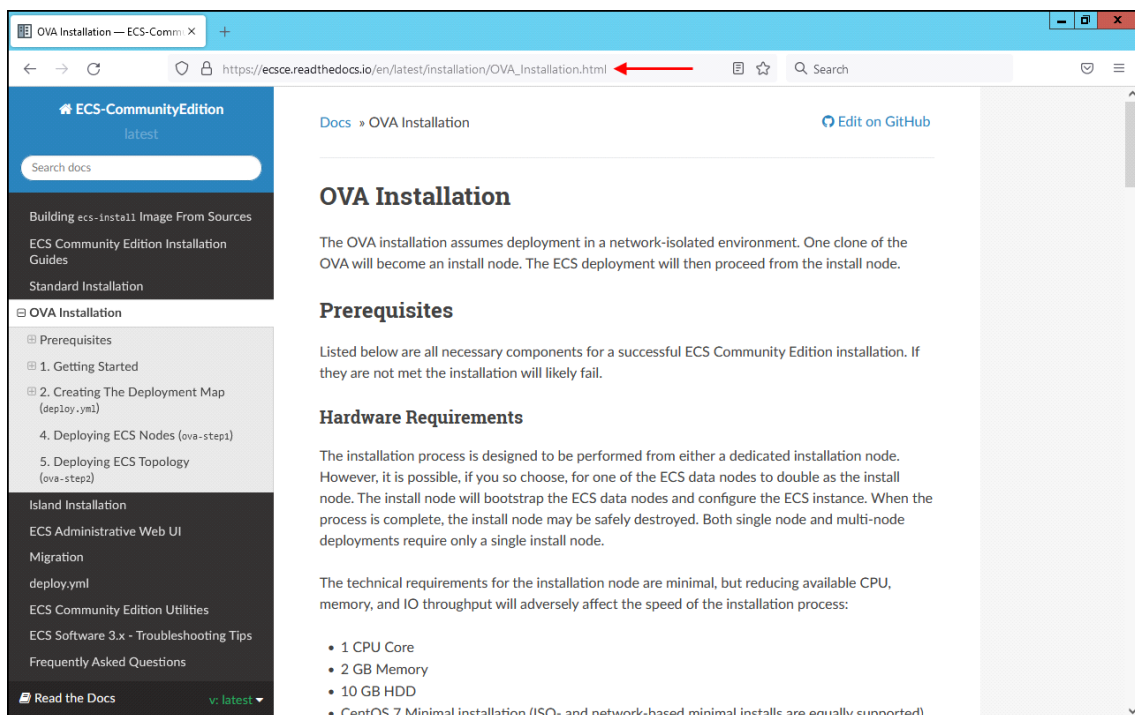


... it appears to be for the older release 3.5.1.2 - although it has a timestamp pointing more towards 3.7.0.0:



You can find even more literature if you access this URL:

[https://ecsce.readthedocs.io/en/latest/installation/OVA\\_Installation.html](https://ecsce.readthedocs.io/en/latest/installation/OVA_Installation.html)



However, at the bottom of the window you will find a copyright info pointing to the year 2017 - so this document might not be current.

## II. Installing the OVA image

These are the core steps which you must follow for a top-down installation:

- Deploy the OVA image
- Boot the Linux system
- Adapt various system parameters
- Prepare & install the ECS Software – according to your desired configuration
  - There are 3 possibilities:
    - 1-node installation with internet connection
    - 4-node installation with internet connection
    - 1-node installation without internet connection (aka *Island Installation*)

### II-1. Deploying the OVA image

For a sysadmin, this is business-as-usual - this is why I will not describe it in detail. Keep in mind that the necessary disk space will be 300 GB - a moderate value. BTW - I succeeded in deploying the image on my more powerful Windows workstation running VMware Workstation 16.



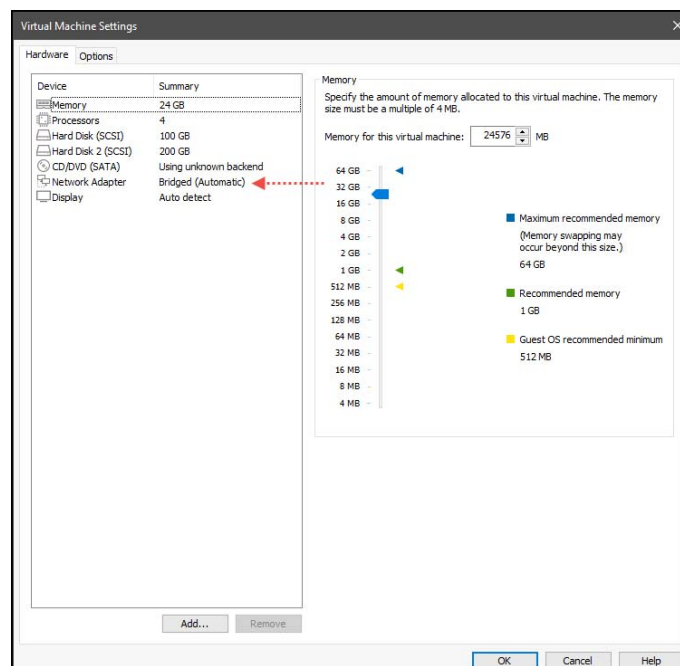
You should neither change the number of CPUs (4) nor the amount of RAM (24 GB) - the documents especially mentions this minimal requirement (at least for the install process ;-)

With ECS-CE release 3.7.0.0 the deployment will automatically create 2 disks:

- the smaller one (100 GB) for the OS and the ECS software themselves
- the larger one (200 GB) for the ECS data

The VMware Workstation software will automatically create them using the *Thin Provisioning* scheme.

These are the VM settings after the deployment:





The NIC does need an active internet connection for the installation!  
So you must connect him to an appropriate virtual network.

### 15.2.1 Deployments into Soft-Isolated and Air-Gapped Island Environments

#### Important information regarding Island deployments

Please be aware that **install node bootstrapping requires Internet access to the hardware or virtual machine** that will become the install node, but **once this step is complete, the machine can be removed from the Internet** and migrated into the Island environment.

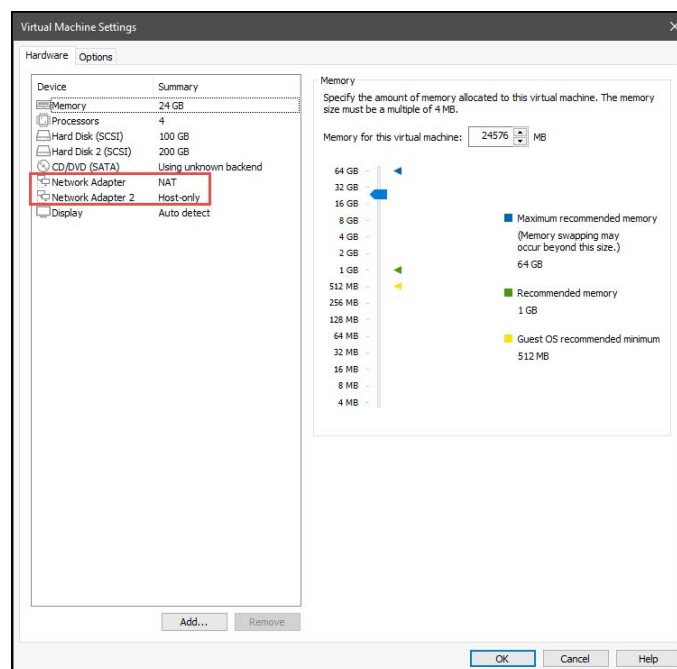
If you prefer to download a prefab install node as an OVF/OVA, follow one of the links below. Please note that OVAs are produced upon each release and do not necessarily have the most current software.

Please see the [release page](#) for OVA download links.

In my case, I use 2 networks:

- a dedicated host-only network for the domain
- a NAT network for the internet, which I only use for maintenance (i.e. software updates).

This is why I extended my NIC configuration with another adapter:



## II-2. Copy/clone the VM

Such is only necessary for a multi-node installation. In my test environment, I will setup 'only' a small single-node installation (aka *all-in-1-island-installation*).

## II-3. Booting & preparing the VM

Now it is time to boot the VM. During the process, you might notice that the OS is a CentOS 7.9 release.

Finally, you must login. With the deployment, these 2 users will be created automatically:

name	<code>admin</code>	standard password	<code>ChangeMe</code>
name	<code>root</code>	standard password	<code>ChangeMe</code>

With ECS-CE 3.7.0.0 (Centos 7.9) you can ...

- only login as `root` from the locale console
- If you want/must use the `root` account you ...
  - must either change the user (`su -`) or you
  - must use the `sudo` command prefix

### II-3.1. Changing the keyboard layout

If required, you should do this first to avoid typos. To change to the german layout, please use the following commands:

<code>localectl set-keymap de</code>	will just switch the layout for the current session
<code>localectl</code>	will permanently switch the keyboard layout

### II-3.2. Checking the OS version

Of course you must not necessarily watch the messages during the boot process. You can also verify the version later using the following command:

```
$ rpm -q centos-release
centos-release-7-9.2009.1.el7.centos.x86_64
```

See also the screenshot on the next page.

```

[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# hostname
localhost.localdomain
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# localectl set-keymap de
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# localectl
System Locale: LANG=en_US.UTF-8
    UC Keymap: de
    X11 Layout: de
    X11 Model: pc105
    X11 Options: terminate:ctrl_alt_bksp
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# rpm -q centos-release
centos-release-7-9.2009.1.el7.centos.x86_64
[root@localhost ~]#

```

### II-3.3. Generally disabling the IPv6 support

Currently, ECS-CE does not yet support IPv6. Therefore it is a good idea to disable it completely.

- For this purpose, add the following lines to your file `/etc/sysctl.conf` :

```

net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1

```

- Activate the new settings with `sysctl -p` (no re-boot necessary).

```

[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# more /etc/sysctl.conf
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
#
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# sysctl -p
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
[root@localhost ~]#
[root@localhost ~]#

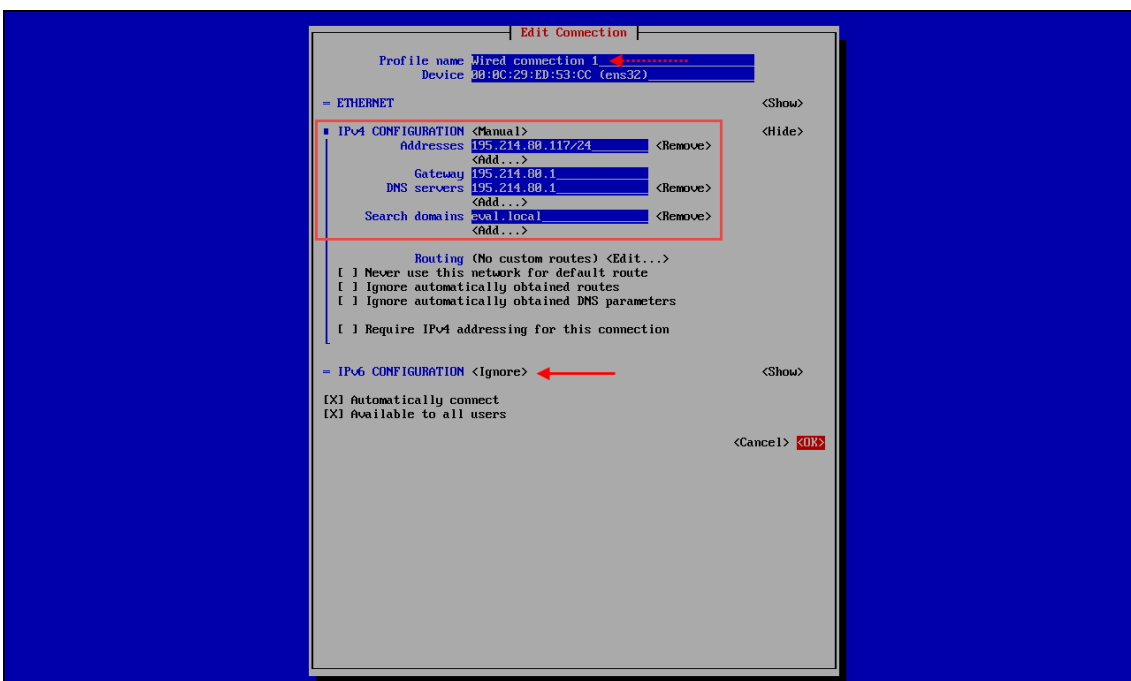
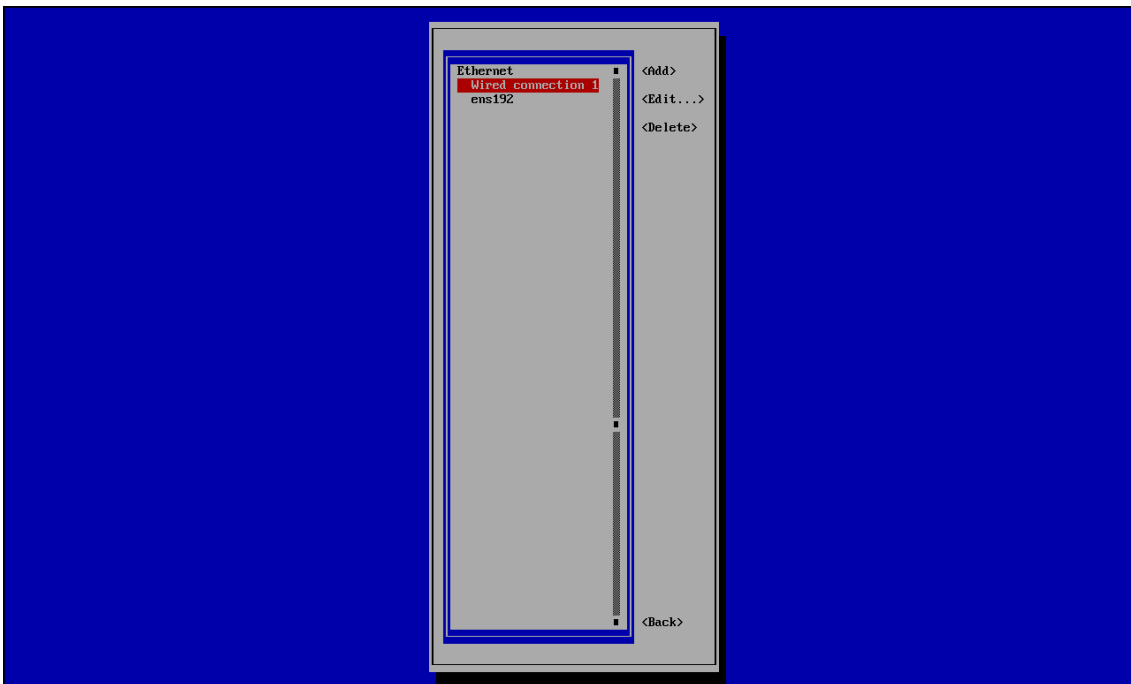
```

## II-3.4. Assigning a fixed address for the second NIC

Later, I will only access the ECS via this network adapter inside of the domain. So it makes sense, to assign him a fixed IP for this purpose. You usually do this with the command line GUI tool `nmtui` (NetworkManager TUI).

In my test environment, I will use the following settings:

IPv4 address	195.214.80.117/24	(masks the whole subnet)
Gateway	195.214.80.1	
DNS server	195.214.80.1	
Search domains	eval.local	







Alternatively, you may use the command

`ip address show`

As you see, the feedback is somehow different from what `ifconfig` returned - it will most likely confuse you:

```

[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# ip address show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:0c:29:ed:53:c2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.115.181/24 brd 192.168.115.255 scope global noprefixroute dynamic ens192
        valid_lft 4285sec preferred_lft 1785sec
3: ens32: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:0c:29:ed:53:c2 brd ff:ff:ff:ff:ff:ff
    inet 195.214.88.117/24 brd 195.214.88.255 scope global noprefixroute ens32
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feed:53cc/64 scope link
        valid_lft forever preferred_lft forever
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:12:fa:7d:1b brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# ping 195.214.88.1
PING 195.214.88.1 (195.214.88.1) 56(84) bytes of data.
64 bytes from 195.214.88.1: icmp_seq=1 ttl=128 time=0.784 ms
64 bytes from 195.214.88.1: icmp_seq=2 ttl=128 time=0.676 ms
^C
--- 195.214.88.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1814ms
rtt min/avg/max/mdev = 0.676/0.730/0.784/0.054 ms
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# ping www.avus-cr.de
PING www.avus-cr.de (85.13.137.165) 56(84) bytes of data.
64 bytes from dd16726.kasserver.com (85.13.137.165): icmp_seq=1 ttl=128 time=24.8 ms
64 bytes from dd16726.kasserver.com (85.13.137.165): icmp_seq=2 ttl=128 time=23.5 ms
^C
--- www.avus-cr.de ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1882ms
rtt min/avg/max/mdev = 23.517/24.163/24.889/0.646 ms
[root@localhost ~]#

```

Finally, as the screenshot shows, you should verify the connection to the internet as well as for the domain network.

## II-3.5. Disabling the firewall permanently

As a precaution, I have generally disabled the firewall activity on this server with these commands:

```
systemctl disable firewalld
systemctl stop firewalld
systemctl status firewalld
```

This show the process and the verification:

```
root@localhost ~#
root@localhost ~# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-07-05 08:29:28 GMT; 19s ago
     Docs: man:firewalld(1)
    Main PID: 2153 (firewalld)
       Tasks: 2
      Memory: 22.0M
    CGroup: /system.slice/firewalld.service
           └─2153 /usr/bin/python2 -Es /usr/sbin/firewalld --nofork --nopid

Jul 05 08:29:28 localhost.localdomain systemd[1]: Starting firewalld - dynamic firewall daemon...
Jul 05 08:29:28 localhost.localdomain systemd[1]: Started firewalld - dynamic firewall daemon.
Jul 05 08:29:28 localhost.localdomain firewalld[2153]: WARNING: AllowZoneDrifting is enabled. This is considered an insecure configuration option. It... it now.
Jul 05 08:29:29 localhost.localdomain firewalld[2153]: WARNING: COMMAND_FAILED: /usr/sbin/iptables -w0 -D FORWARD -i docker0 -o docker0 -j DROP' fa...chain?}.
Jul 05 08:29:29 localhost.localdomain firewalld[2153]: WARNING: COMMAND_FAILED: /usr/sbin/iptables -w0 -D FORWARD -i docker0 -o docker0 -j DROP' fa...chain?}.
Hint: Some lines were ellipsized, use -l to show in full.
root@localhost ~#
root@localhost ~#
root@localhost ~# systemctl stop firewalld
root@localhost ~#
root@localhost ~#
root@localhost ~# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Tue 2022-07-05 08:30:07 GMT; 3s ago
     Docs: man:firewalld(1)
    Process: 2153 ExecStart=/usr/sbin/firewalld --nofork --nopid $FIREWALLD_ARGS (code=exited, status=0/SUCCESS)
    Main PID: 2153 (code=exited, status=0/SUCCESS)

Jul 05 08:29:28 localhost.localdomain systemd[1]: Starting firewalld - dynamic firewall daemon...
Jul 05 08:29:28 localhost.localdomain systemd[1]: Started firewalld - dynamic firewall daemon.
Jul 05 08:29:28 localhost.localdomain firewalld[2153]: WARNING: AllowZoneDrifting is enabled. This is considered an insecure configuration option. It... it now.
Jul 05 08:29:29 localhost.localdomain firewalld[2153]: WARNING: COMMAND_FAILED: /usr/sbin/iptables -w0 -D FORWARD -i docker0 -o docker0 -j DROP' fa...chain?}.
Jul 05 08:29:29 localhost.localdomain firewalld[2153]: WARNING: COMMAND_FAILED: /usr/sbin/iptables -w0 -D FORWARD -i docker0 -o docker0 -j DROP' fa...chain?}.
Jul 05 08:30:05 localhost.localdomain systemd[1]: Stopping firewalld - dynamic firewall daemon...
Jul 05 08:30:07 localhost.localdomain systemd[1]: Stopped firewalld - dynamic firewall daemon.
Hint: Some lines were ellipsized, use -l to show in full.
root@localhost ~#
root@localhost ~#
root@localhost ~# systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
root@localhost ~#
root@localhost ~#
```



Do not forget to add an appropriate entry in your DNS configuration!

Now you should be able to successfully ping the new server from any other computer:

```
Administrator: Command Prompt
C:\>
C:\>ping 195.214.80.117
Pinging 195.214.80.117 with 32 bytes of data:
Reply from 195.214.80.117: bytes=32 time<1ms TTL=64
Reply from 195.214.80.117: bytes=32 time<1ms TTL=64
Reply from 195.214.80.117: bytes=32 time<1ms TTL=64
Reply from 195.214.80.117: bytes=32 time<1ms TTL=64

Ping statistics for 195.214.80.117:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>^n
C:\>
```

## II-3.6. Modifying the `hosts` file

This is what I added at my ECS-CE server:

```
[root@localhost ~]#  
[root@localhost ~]#  
[root@localhost ~]#  
[root@localhost ~]#  
[root@localhost ~]#  
[root@localhost ~]# more /etc/hosts  
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6  
#  
195.214.80.1 dcon dcon.eval.local  
195.214.80.111 ddve74-1 ddve74-1.eval.local  
195.214.80.222 22-nwtest 22-nwtest.eval.local  
  
[root@localhost ~]#  
[root@localhost ~]#  
[root@localhost ~]#  
[root@localhost ~]#  
[root@localhost ~]#  
[root@localhost ~]#
```

## II-3.7. Providing a NTP service

To keep the timestamps in sync later, it is mandatory to provide the correct time within the domain. So your ECS-CE server - even if isolated from the internet (remember the *island installation*) must be able to access a time normal indirectly.



**If you do not provide a time server access, the scripts will later 'hang' at such a step forever!**

For this reason, I decided to install a NTP server for my domain on the domain controller (dcon.eval.local). This seems to be also possible with Windows (you will find details in the web). However, this somehow does not seem to be compatible with the ECS software. That's why I finally installed the *Meinberg* NTP server Software. You can get it from this URL:

[https://www.meinbergglobal.com/english/sw/ntp.htm#ntp\\_stable](https://www.meinbergglobal.com/english/sw/ntp.htm#ntp_stable)

These are the benefits:

- The software is available for Windows.
- It will automatically disable the Windows specific time service.
- It will automatically be re-activated after a re-boot.
- Even if the host is disconnected from the internet an NTP server is available for the domain.
- The software seems to work very stable.



Do not forget to enable and start the `ntpd` daemon at the ECS server!

As you can see below, the NTP service will not be started by default.

You control the daemon via these commands:

```
systemctl disable | enable ntpd
systemctl start   | stop   ntpd
systemctl status  | ntpd
```

```
[root@localhost ~]#
[root@localhost ~]# systemctl status ntpd
# ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; disabled; vendor preset: disabled)
   Active: inactive (dead)

Jul 01 13:48:54 ecscce37.aval.local ntpd[2128]: Listen normally on 4 docker0 172.17.0.1 UDP 123
Jul 01 13:48:54 ecscce37.aval.local ntpd[2128]: Listening on routing socket on fd #21 for interface updates
Jul 01 13:48:54 ecscce37.aval.local systemd[1]: Started Network Time Service.
Jul 01 13:48:55 ecscce37.aval.local ntpd[2128]: 0.0.0.0 c016 06 restart
Jul 01 13:48:55 ecscce37.aval.local ntpd[2128]: 0.0.0.0 c012 02 freq_set kernel 0.000 PPM
Jul 01 13:48:55 ecscce37.aval.local ntpd[2128]: 0.0.0.0 c011 01 freq_not_set
Jul 01 13:49:01 ecscce37.aval.local ntpd[2128]: 0.0.0.0 c614 04 freq_mode
Jul 01 13:50:19 ecscce37.aval.local systemd[1]: Stopping Network Time Service...
Jul 01 13:50:19 ecscce37.aval.local ntpd[2128]: ntpd exiting on signal 15
Jul 01 13:50:19 ecscce37.aval.local systemd[1]: Stopped Network Time Service.
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# systemctl start ntpd
[root@localhost ~]#
[root@localhost ~]# systemctl status ntpd
# ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2022-07-01 13:50:47 GMT; 8s ago
   Process: 2162 ExecStart=/usr/sbin/ntpd -u ntp:ntp $OPTIONS (code=exited, status=0/SUCCESS)
   Main PID: 2163 (ntpd)
   Tasks: 1
   Memory: 612.0K
   CGroup: /system.slice/ntpd.service
           └─2163 /usr/sbin/ntpd -u ntp:ntp -g

Jul 01 13:50:47 ecscce37.aval.local ntpd[2163]: Listen and drop on 0 *wildcard 0.0.0.0 UDP 123
Jul 01 13:50:47 ecscce37.aval.local ntpd[2163]: Listen and drop on 1 *wildcard *: UDP 123
Jul 01 13:50:47 ecscce37.aval.local ntpd[2163]: Listen normally on 2 lo 127.0.0.1 UDP 123
Jul 01 13:50:47 ecscce37.aval.local ntpd[2163]: Listen normally on 3 ens192 195.214.88.117 UDP 123
Jul 01 13:50:47 ecscce37.aval.local ntpd[2163]: Listen normally on 4 docker0 172.17.0.1 UDP 123
Jul 01 13:50:47 ecscce37.aval.local ntpd[2163]: Listening on routing socket on fd #21 for interface updates
Jul 01 13:50:47 ecscce37.aval.local ntpd[2163]: 0.0.0.0 c016 06 restart
Jul 01 13:50:47 ecscce37.aval.local ntpd[2163]: 0.0.0.0 c012 02 freq_set kernel 0.000 PPM
Jul 01 13:50:47 ecscce37.aval.local ntpd[2163]: 0.0.0.0 c011 01 freq_not_set
Jul 01 13:50:54 ecscce37.aval.local ntpd[2163]: 0.0.0.0 c614 04 freq_mode
[root@localhost ~]#
```

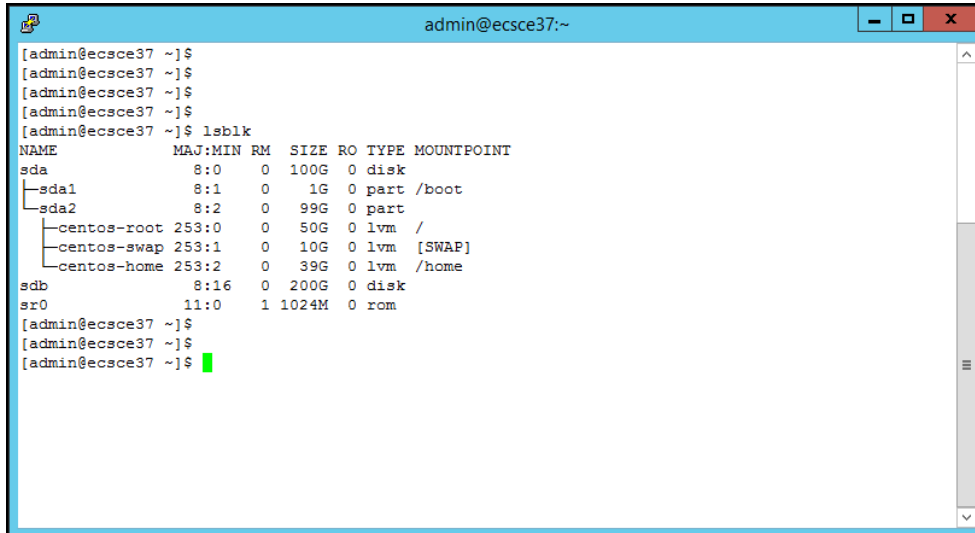
Here is the verification:

```
[root@ecscce37 ~]#
[root@ecscce37 ~]#
[root@ecscce37 ~]# systemctl status ntpd
# ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2022-07-01 13:54:30 GMT; 1min 42s ago
   Process: 832 ExecStart=/usr/sbin/ntpd -u ntp:ntp $OPTIONS (code=exited, status=0/SUCCESS)
   Main PID: 842 (ntpd)
   Tasks: 1
   Memory: 2.5M
   CGroup: /system.slice/ntpd.service
           └─842 /usr/sbin/ntpd -u ntp:ntp -g

Jul 01 13:54:32 ecscce37.aval.local ntpd_intres[846]: DNS 0.centos.pool.ntp.org -> 189.230.227.90
Jul 01 13:54:32 ecscce37.aval.local ntpd_intres[846]: DNS 1.centos.pool.ntp.org -> 164.68.116.11
Jul 01 13:54:33 ecscce37.aval.local ntpd_intres[846]: DNS 2.centos.pool.ntp.org -> 194.158.196.171
Jul 01 13:54:33 ecscce37.aval.local ntpd_intres[846]: DNS 3.centos.pool.ntp.org -> 162.159.200.123
Jul 01 13:54:35 ecscce37.aval.local ntpd[842]: Listen normally on 3 ens192 195.214.88.117 UDP 123
Jul 01 13:54:35 ecscce37.aval.local ntpd[842]: Listen normally on 4 docker0 172.17.0.1 UDP 123
Jul 01 13:54:35 ecscce37.aval.local ntpd[842]: new interface(s) found: waking up resolver
Jul 01 13:54:42 ecscce37.aval.local ntpd[842]: 0.0.0.0 c61c 0c clock_step -0.158193 s
Jul 01 13:54:42 ecscce37.aval.local ntpd[842]: 0.0.0.0 c614 04 freq_mode
Jul 01 13:54:43 ecscce37.aval.local ntpd[842]: 0.0.0.0 c618 08 no_sys_peer
[root@ecscce37 ~]#
[root@ecscce37 ~]#
[root@ecscce37 ~]# ntpq -p
remote refid st t when poll reach delay offset jitter
-----
+ntp2.am-hosting 131.188.3.222 2 u 61 64 1 15.389 1.679 1.220
+node01.meteora- 131.188.3.222 2 u - 64 1 18.434 2.736 2.793
+194.158.196.171 89.189.251.24 2 u 60 64 1 44.702 0.121 1.176
+time.cloudflare 10.40.8.105 3 u 62 64 1 19.616 1.573 1.286
[root@ecscce37 ~]#
[root@ecscce37 ~]#
[root@ecscce37 ~]# ntpq -pn -4
remote refid st t when poll reach delay offset jitter
-----
+189.230.227.90 131.188.3.222 2 u 9 64 3 16.821 2.856 2.237
+164.68.116.11 131.188.3.222 2 u 12 64 1 18.434 2.736 2.793
+194.158.196.171 89.189.251.24 2 u 8 64 3 44.098 2.440 3.326
+162.159.200.123 10.40.8.105 3 u 74 64 1 19.616 1.573 1.286
[root@ecscce37 ~]#
[root@ecscce37 ~]#
```

## II-3.8. Verifying the device name for the data disk

Use the command `lsblk` for this purpose:



```
admin@ecsce37:~$  
[admin@ecsce37 ~]$  
[admin@ecsce37 ~]$  
[admin@ecsce37 ~]$  
[admin@ecsce37 ~]$  
[admin@ecsce37 ~]$ lsblk  
NAME            MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT  
sda              8:0    0 100G  0 disk  
├─sda1           8:1    0   1G  0 part /boot  
└─sda2           8:2    0  99G  0 part  
   ├─centos-root 253:0   0   50G  0 lvm  /  
   ├─centos-swap 253:1   0  10G  0 lvm  [SWAP]  
   └─centos-home 253:2   0  39G  0 lvm  /home  
sdb              8:16   0 200G  0 disk  
sr0             11:0    1 1024M  0 rom
```

## II-3.9. Verify the 'stability' of all modifications



Make sure that all modifications/functions will remain after a reboot of the *dcon* and the ECS-CE host and that the access to the internet and the NTP server is still available. This is mandatory for the next steps.

## III. Installing and configuring the ECS software

This is the general roadmap:

- In the first step, you must prepare the system (so-called *Bootstrapping*).
  - For this purpose you must modify the settings within a configuration parameter file (`deploy.yml`).  
You only have to modify this single file.
  - Next you must start the `bootstrap` script to initiate the process.
    - This will first copy the configuration parameter file.
    - Then all the parameters will be verified - if possible.
    - Next the latest software will be downloaded from the internet.  
**This is exactly why you still need a working internet connection!**
    - Finally, the system will automatically be rebooted  
After that, it will have the new name *luna* assigned.
- For the island installation you must now cut the internet connection.
- Finally, you must run 3 installation scripts.
  - Depending on the setup you want to run, you either need 2 or 3 scripts.
  - For my intended purpose (*island = all-in-1-without-internet*) 3 scripts must be started in the proper sequence.
  - No other modification will be necessary.

### III-1. Bootstrapping the installation

#### III-1.1. Editing the parameter configuration file 'deploy.yml'

Unfortunately, the VM does provide multiple files with the same name in various directories. However, you just need this file:

```
/home/admin/ECS-CommunityEdition/deploy.yml
```

So you must only modify `/home/admin/ECS-CommunityEdition/deploy.yml`

You will see the file on the next pages. I have marked all my necessary changes in blue.

```
# deploy.yml reference implementation v2.8.0

# [Optional]
# By changing the license_accepted boolean value to "true" you are
# declaring your agreement to the terms of the license agreement
# contained in the license.txt file included with this software
# distribution.
licensing:
  license_accepted: false

#autonames:
#  custom:
#    - ecs01
#    - ecs02
#    - ecs03
#    - ecs04
#    - ecs05
#    - ecs06

# [Required]
# Deployment facts reference
facts:

# [Required]
# Node IP or resolvable hostname from which installations will be launched
# The only supported configuration is to install from the same node as the
# bootstrap.sh script is run.
# NOTE: if the install node is to be migrated into an island environment,
#       the hostname or IP address listed here should be the one in the
#       island environment.
install_node: 195.214.80.117

# [Required]
# IPs of machines that will be whitelisted in the firewall and allowed
# to access management ports of all nodes. If this is set to the
# wildcard (0.0.0.0/0) then anyone can access management ports.
management_clients:
  - 0.0.0.0/0

# [Required]
# These credentials must be the same across all nodes. Ansible uses these credentials to
# gain initial access to each node in the deployment and set up ssh public key authentication.
# If these are not correct, the deployment will fail.
ssh_defaults:
# [Required]
# Username to use when logging in to nodes
ssh_username: admin
# [Required]
# Password to use with SSH login
# *** Set to same value as ssh_username to enable SSH public key authentication ***
ssh_password: ChangeMe
# [Required when enabling SSH public key authentication]
# Password to give to sudo when gaining root access.
ansible_become_pass: ChangeMe
# [Required]
# Select the type of crypto to use when dealing with ssh public key
# authentication. Valid values here are:
# - "rsa" (Default)
# - "ed25519"
ssh_crypto: rsa

# [Required]
# Environment configuration for this deployment.
node_defaults:
  dns_domain: eval.local
  dns_servers:
    - 195.214.80.1
  ntp_servers:
    - 195.214.80.1

.....
```



.....

```
#
# [Optional]
# VFS path to source of randomness
# Defaults to /dev/urandom for speed considerations.  If you prefer /dev/random, put that here.
# If you have a /dev/srandom implementation or special entropy hardware, you may use that too
# so long as it implements a /dev/random type device.
entropy_source: /dev/urandom
#
# [Optional]
# Picklist for node names.
# Available options:
# - "moons" (ECS CE default)
# - "cities" (ECS SKU-flavored)
# - "custom" (uncomment and use the top-level autonames block to define these)
# autonaming: custom

#
# [Optional]
# If your ECS comes with differing default credentials, you can specify those here
# ecs_root_user: root
# ecs_root_pass: ChangeMe

# [Optional]
# Storage pool defaults. Configure to your liking.
# All block devices that will be consumed by ECS on ALL nodes must be listed under the
# ecs_block_devices option. This can be overridden by the storage pool configuration.
# At least ONE (1) block device is REQUIRED for a successful install. More is better.
storage_pool_defaults:
  is_cold_storage_enabled: false
  is_protected: false
  description: Default storage pool description
  ecs_block_devices:
    - /dev/sdb

# [Required]
# Storage pool layout. You MUST have at least ONE (1) storage pool for a successful install.
storage_pools:
  - name: sp1
    members:
      - 195.214.80.117
    options:
      is_protected: false
      is_cold_storage_enabled: false
      description: My First SP
      ecs_block_devices:
        - /dev/sdb

# [Optional]
# VDC defaults. Configure to your liking.
virtual_data_center_defaults:
  description: Default virtual data center description

# [Required]
# Virtual data center layout. You MUST have at least ONE (1) VDC for a successful install.
# Multi-VDC deployments are not yet implemented
virtual_data_centers:
  - name: vdc1
    members:
      - sp1
    options:
      description: My First VDC

# [Optional]
# Replication group defaults. Configure to your liking.
replication_group_defaults:
  description: Default replication group description
  enable_rebalancing: true
  allow_all_namespaces: true
  is_full_rep: false
```

.....

```
.....

# [Optional, required for namespaces]
# Replication group layout. You MUST have at least ONE (1) RG to provision namespaces.
replication_groups:
  - name: rg1
    members:
      - vdc1
    options:
      description: My First RG
      enable_rebalancing: true
      allow_all_namespaces: true
      is_full_rep: false

# [Optional]
# Management User defaults
management_user_defaults:
  is_system_admin: false
  is_system_monitor: false

# [Optional]
# Management Users
management_users:
  - username: admin1
    password: ChangeMe
    options:
      is_system_admin: true
  - username: monitor1
    password: ChangeMe
    options:
      is_system_monitor: true

# [Optional]
# Namespace defaults
namespace_defaults:
  is_stale_allowed: false
  is_compliance_enabled: false

# [Optional]
# Namespace layout
namespaces:
  - name: ns1
    replication_group: rg1
    administrators:
      - root
    options:
      is_stale_allowed: false
      is_compliance_enabled: false

# [Optional]
# Object User defaults
object_user_defaults:
  # Comma-separated list of Swift authorization groups
  swift_groups_list:
    - users
  # Lifetime of S3 secret key in minutes
  s3_expiry_time: 2592000

.....
```

.....

```
# [Optional]
# Object Users
object_users:
  - username: object_admin1
    namespace: ns1
    options:
      swift_password: ChangeMe
      swift_groups_list:
        - admin
        - users
      s3_secret_key: ChangeMeChangeMeChangeMeChangeMeChangeMe
      s3_expiry_time: 2592000
  - username: object_user1
    namespace: ns1
    options:
      swift_password: ChangeMe
      s3_secret_key: ChangeMeChangeMeChangeMeChangeMeChangeMe

# [Optional]
# Bucket defaults
bucket_defaults:
  namespace: ns1
  replication_group: rg1
  head_type: s3
  filesystem_enabled: False
  stale_allowed: False
  encryption_enabled: False
  owner: object_admin1

# [Optional]
# Bucket layout (optional)
buckets:
  - name: bucket1
    options:
      namespace: ns1
      replication_group: rg1
      owner: object_admin1
      head_type: s3
      filesystem_enabled: False
      stale_allowed: False
      encryption_enabled: False
```



As a precaution, you should now make a snapshot or a copy of your VM!

### III-1.2. Update the other files with 'update\_deploy'

Using the `admin` account, you must run the following command:

```
[admin@localhost bin]$
[admin@localhost bin]$
[admin@localhost bin]$ pwd
/home/admin/bin
[admin@localhost bin]$
[admin@localhost bin]$
[admin@localhost bin]$
[admin@localhost bin]$ update_deploy
> Updating /opt/emc/ecs-install/deploy.yml from deploy.yml
31c31
<   install_node: 195.214.80.117
---
>   install_node: 192.168.2.200
65c65
<     dns_domain: eval.local
---
>     dns_domain: local
67c67
<       - 195.214.80.1
---
>       - 192.168.2.2
69c69
<         - 195.214.80.1
---
>         - 192.168.2.2
102c102
<           - /dev/sdb
---
>           - /dev/vda
109c109
<             - 195.214.80.117
---
>             - 192.168.2.200
115c115
<               - /dev/sdb
---
>               - /dev/vda
> Recreating ecs-install data container
ecs-install> Initializing data container, one moment ..DEPRECATION: Python 2.7 will reach
the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 won't
be maintained after that date. A future version of pip will drop support for Python 2.7.
You are using pip version 19.0.3, however version 20.3.4 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
. OK
ecs-install> Applying and validating deploy.yml...

PLAY [Installer | Setup access between ecs-install and install node]
*****

TASK [Gathering Facts]
*****
ok: [195.214.80.117]

TASK [group_by]
*****
ok: [195.214.80.117]

PLAY [Installer | Create SSH keys]
*****

TASK [Gathering Facts]
*****
ok: [localhost]

.....
```

```
.....

TASK [installer_generate_ssh_keys : Installer | Check RSA Keypair]
*****
ok: [localhost]

TASK [installer_generate_ssh_keys : Installer | Generate RSA Keypair]
*****
changed: [localhost]

TASK [installer_generate_ssh_keys : Installer | Check ed25519 Keypair]
*****

TASK [installer_generate_ssh_keys : Installer | Generate ed25519 Keypair]
*****

TASK [installer_generate_ssh_keys : Installer | Fail when no crypto selected]
*****

TASK [installer_generate_ssh_keys : Installer | Ensure directory permissions on ssh
keystore] *****
ok: [localhost] => (item=/opt/ssh/id_rsa)
ok: [localhost] => (item=/opt/ssh/id_rsa)

PLAY [CentOS 7 | Setup SSH on install node]
*****

TASK [CentOS_7_configure_ssh : CentOS 7 | Distribute ed25519 ssh key]
*****

TASK [CentOS_7_configure_ssh : CentOS 7 | Distribute rsa ssh key]
*****
changed: [195.214.80.117]

TASK [CentOS_7_configure_ssh : CentOS 7 | Disable SSH UseDNS]
*****
changed: [195.214.80.117]

TASK [CentOS_7_configure_ssh : CentOS 7 | Disable requiretty]
*****
ok: [195.214.80.117]

TASK [CentOS_7_configure_ssh : CentOS 7 | Disable sudo password reverification for admin
group] *****
changed: [195.214.80.117]

TASK [CentOS_7_configure_ssh : CentOS 7 | Disable sudo password reverification for wheel
group] *****
changed: [195.214.80.117]

RUNNING HANDLER [CentOS_7_configure_ssh : CentOS 7 | Restart sshd]
*****
changed: [195.214.80.117]

PLAY RECAP
*****
195.214.80.117      : ok=8      changed=5      unreachable=0    failed=0
localhost         : ok=4      changed=1      unreachable=0    failed=0

Playbook run took 0 days, 0 hours, 0 minutes, 28 seconds

[admin@localhost bin]$
```

### III-1.3. 'Bootstrapping' the installation

Again, as user admin, just run the appropriate command. With all the dowloads which you will not see any more in the final screen, the whole process will take about 10-15 mins.

```
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$ pwd
/home/admin/ECS-CommunityEdition
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$ ./bootstrap.sh -y -g -c deploy.yml
>
> ECS Community Edition Install Node Bootstrap 2.9.1r
> ECS Software Image emccorp/ecs-software-3.7.0:3.7.0.0
> ~~~~~
>
> Escalating privileges
> You may be presented with the system sudo banner and asked
> for your password depending on the Linux flavor and default
> sudo configuration for your system.
>
>
> Environment is centos linux release 7.9.2009 (core)
> [supported: true]
>
> We collect some hardware and OS info into a log file on the
> install node in case something fails and you want help trouble-
> shooting. HOWEVER, absolutely nothing is transmitted over the
> Internet or shared with EMC, GitHub, or anyone else unless or
> until you decide to attach the log file or copy & paste its
> content into a help request on GitHub (or where ever).
>
> If you are curious to see what's collected, the log is here:
> /home/admin/ECS-CommunityEdition/install.log
>
> It is perfectly fine to remove this log file at any time.
>
> Onward to bootstrapping. This can take anywhere between five
> minutes to a few hours depending on many factors, the most
> important being the speed of your Internet connection.
> The ECS software docker image is around 1.5GiB.
> Bootstrapping requires about 10 minutes under KVM on a
> Xeon E5 with a 250Mbps Internet connection.
>
> This script installs all packages that are both required for
> the deployment and that we think will be helpful to you when
> managing and operating your environment.
>
> We're going to start working with Docker now. If you elected
> to build your own ecs-install image rather than pull one from
> the EMC Dockerhub repo, it will add some time to your initial
> bootstrap.
>
> We are now pulling the emccorp/ecs-software-3.7.0 image.
> This can take quite a long time over a slow Internet link or
> if the backing block storage system is slower than usual.
>
> All done bootstrapping your install node.
.....
```

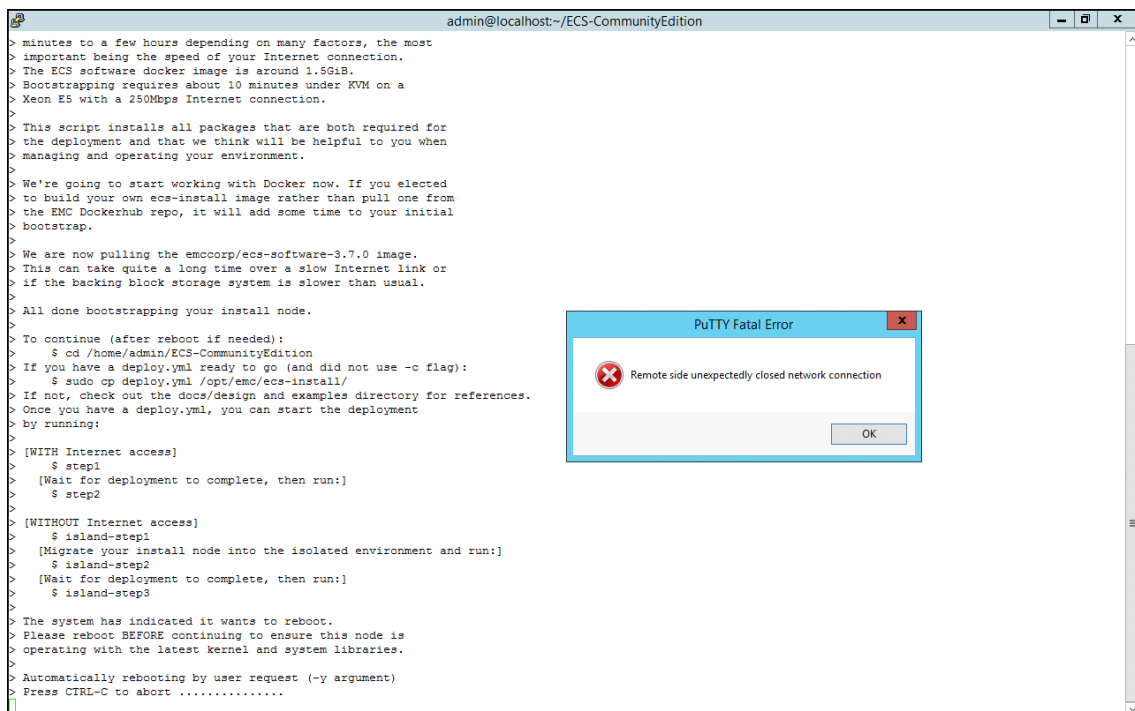
This indicates the end of the process. However, as you see on the next page, the program finally states the next steps you still have to run.

For my *island* installation this is what you must do. In the end, the server is automatically re-booted:

```

>
> .....
>
> To continue (after reboot if needed):
>   $ cd /home/admin/ECS-CommunityEdition
> If you have a deploy.yml ready to go (and did not use -c flag):
>   $ sudo cp deploy.yml /opt/emc/ecs-install/
> If not, check out the docs/design and examples directory for references.
> Once you have a deploy.yml, you can start the deployment
> by running:
>
> [WITH Internet access]
>   $ step1
>   [Wait for deployment to complete, then run:]
>   $ step2
>
> [WITHOUT Internet access]
>   $ island-step1
>   [Migrate your install node into the isolated environment and run:]
>   $ island-step2
>   [Wait for deployment to complete, then run:]
>   $ island-step3
>
> The system has indicated it wants to reboot.
> Please reboot BEFORE continuing to ensure this node is
> operating with the latest kernel and system libraries.
>
> Automatically rebooting by user request (-y argument)
> Press CTRL-C to abort .....

```



After this step, the hostname `luna` will be assigned.

Once again, you should either create a snapshot or a backup!

Finally, you must re-enable and restart the `ntpd` service and verify its function. See the next page for details.



```

[admin@luna ~]$
[admin@luna ~]$ systemctl status ntpd
? ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; disabled; vendor preset:
disabled)
   Active: inactive (dead)
[admin@luna ~]$
[admin@luna ~]$
[admin@luna ~]$
[admin@luna ~]$ systemctl start ntpd
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to manage system services or units.
Authenticating as: admin
Password: <password>
==== AUTHENTICATION COMPLETE ====
[admin@luna ~]$
[admin@luna ~]$
[admin@luna ~]$
[admin@luna ~]$ systemctl enable ntpd
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-unit-files ====
Authentication is required to manage system service or unit files.
Authenticating as: admin
Password: <password>
==== AUTHENTICATION COMPLETE ====
Created symlink from /etc/systemd/system/multi-user.target.wants/ntpd.service to /usr/lib/
systemd/system/ntpd.service.
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ====
Authentication is required to reload the systemd state.
Authenticating as: admin
Password: <password>
==== AUTHENTICATION COMPLETE ====
[admin@luna ~]$
[admin@luna ~]$
[admin@luna ~]$
[admin@luna ~]$ systemctl status ntpd
? ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2022-07-03 21:49:49 GMT; 47s ago
     Main PID: 8508 (ntpd)
    CGroup: /system.slice/ntpd.service
            +-8508 /usr/sbin/ntpd -u ntp:ntp -g

Jul 03 21:49:49 luna ntpd[8508]: Listen normally on 4 ens34 195.214.80.117 UDP 123
Jul 03 21:49:49 luna systemd[1]: Started Network Time Service.
Jul 03 21:49:49 luna ntpd[8508]: Listen normally on 5 docker0 172.17.0.1 UDP 123
Jul 03 21:49:49 luna ntpd[8508]: Listening on routing socket on fd #22 for interface
updates
Jul 03 21:49:49 luna ntpd[8508]: 0.0.0.0 c016 06 restart
Jul 03 21:49:49 luna ntpd[8508]: 0.0.0.0 c012 02 freq_set kernel 0.000 PPM
Jul 03 21:49:49 luna ntpd[8508]: 0.0.0.0 c011 01 freq_not_set
Jul 03 21:49:56 luna ntpd[8508]: 0.0.0.0 c61c 0c clock_step -0.289962 s
Jul 03 21:49:55 luna ntpd[8508]: 0.0.0.0 c614 04 freq_mode
Jul 03 21:49:56 luna ntpd[8508]: 0.0.0.0 c618 08 no_sys_peer
[admin@luna ~]$
[admin@luna ~]$
[admin@luna ~]$
[admin@luna ~]$
[admin@luna ~]$ ntpq -p
      remote           refid      st t when poll reach  delay  offset  jitter
=====
176.9.102.215 .XFAC.      16 u 35m  64    0   0.000   0.000  0.000
 5.161.44.72   .XFAC.      16 u 685   64    0   0.000   0.000  0.000
37.120.164.45 .XFAC.      16 u 1037  64    0   0.000   0.000  0.000
176.9.166.35  .XFAC.      16 u 1117  64    0   0.000   0.000  0.000
[admin@luna ~]
[admin@luna ~]

```



## IV. Installing a single-node 'all-in-1' ECS without internet connection (so-called 'Island Installation')

If all went fine so far, you may now proceed.

### IV-1. Cut the Internet Connection



Before you run the Install scripts, it is mandatory that you cut the internet connection for the first NIC!

This is how to verify the new situation. Keep in mind that the results for the command `ntpq -p` will now take longer to appear:

```

admin@luna:~$
admin@luna:~$
admin@luna:~$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    ether 02:42:ae:6b:69:83 txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 195.214.80.117 netmask 255.255.255.0 broadcast 195.214.80.255
    ether 00:0c:29:ed:53:cc txqueuelen 1000 (Ethernet)
    RX packets 633 bytes 53922 (52.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 308 bytes 41247 (40.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens192: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 00:0c:29:ed:53:c2 txqueuelen 1000 (Ethernet)
    RX packets 458 bytes 33073 (32.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 83 bytes 7948 (7.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 28 bytes 5708 (5.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 28 bytes 5708 (5.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

admin@luna:~$
admin@luna:~$
admin@luna:~$ ntpq -p
=====
      remote             refid      st t when poll reach  delay  offset jitter
-----
157.90.3.240 .XFAC.    16 u 35m  64  0  0.000  0.000  0.000
85.214.71.38  .XFAC.    16 u 382  64  0  0.000  0.000  0.000
185.21.101.59 .XFAC.    16 u 715  64  0  0.000  0.000  0.000
176.9.166.35  .XFAC.    16 u 631  64  0  0.000  0.000  0.000
admin@luna:~$
    
```

## IV-2. Executing the scripts 'island-step1'

The process will take less than a minute!

```
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$ pwd
/home/admin/ECS-CommunityEdition
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$ island-step1
ecs-install> Initializing data container, one moment ..DEPRECATION: Python 2.7 will reach
the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 won't
be maintained after that date. A future version of pip will drop support for Python 2.7.
. OK
ecs-install> Applying and validating deploy.yml...

PLAY [Installer | Setup access between ecs-install and install node]
*****

TASK [group_by]
*****
ok: [195.214.80.117]

PLAY [Installer | Create SSH keys]
*****

TASK [installer_generate_ssh_keys : Installer | Check RSA Keypair]
*****
ok: [localhost]

TASK [installer_generate_ssh_keys : Installer | Generate RSA Keypair]
*****

TASK [installer_generate_ssh_keys : Installer | Check ed25519 Keypair]
*****

TASK [installer_generate_ssh_keys : Installer | Generate ed25519 Keypair]
*****

TASK [installer_generate_ssh_keys : Installer | Fail when no crypto selected]
*****

TASK [installer_generate_ssh_keys : Installer | Ensure directory permissions on ssh
keystore] *****
ok: [localhost] => (item=/opt/ssh/id_rsa)
ok: [localhost] => (item=/opt/ssh/id_rsa)
PLAY [CentOS 7 | Setup SSH on install node]
*****

TASK [CentOS_7_configure_ssh : CentOS 7 | Distribute ed25519 ssh key]
*****

TASK [CentOS_7_configure_ssh : CentOS 7 | Distribute rsa ssh key]
*****
ok: [195.214.80.117]

TASK [CentOS_7_configure_ssh : CentOS 7 | Disable SSH UseDNS]
*****
ok: [195.214.80.117]

.....
```



```
.....

TASK [CentOS_7_configure_ssh : CentOS 7 | Disable requiretty]
*****
ok: [195.214.80.117]

TASK [CentOS_7_configure_ssh : CentOS 7 | Disable sudo password reverification for admin
group] *****
ok: [195.214.80.117]

TASK [CentOS_7_configure_ssh : CentOS 7 | Disable sudo password reverification for wheel
group] *****
ok: [195.214.80.117]

PLAY RECAP
*****
195.214.80.117 : ok=6    changed=0    unreachable=0    failed=0
localhost     : ok=2    changed=0    unreachable=0    failed=0

Playbook run took 0 days, 0 hours, 0 minutes, 3 seconds

PLAY [Installer | Build the package cache]
*****

TASK [installer_build_cache : include_vars]
*****
ok: [localhost]

TASK [installer_build_cache : Installer | Create cache directories]
*****

TASK [installer_build_cache : Installer | Create compressed cache files]
*****

TASK [installer_build_cache : Installer | Create cache distribution torrent file]
*****

PLAY [Installer | Enable torrent ffx]
*****

TASK [file]
*****

PLAY RECAP
*****
localhost     : ok=1    changed=0    unreachable=0    failed=0

Playbook run took 0 days, 0 hours, 0 minutes, 0 seconds

[admin@luna ECS-CommunityEdition]$
```

### IV-3. Verify the 'ntpd' service once again

At the very end, there should be an active `ntpd` connection with your NTP server (here: `dcon`).

This is what it should look like:

```
[admin@luna ECS-CommunityEdition]$ systemctl status ntpd
? ntpd.service - Network Time Service
   Loaded: loaded (/usr/lib/systemd/system/ntpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Sun 2022-07-03 22:05:46 GMT; 3s ago
   Process: 11479 ExecStart=/usr/sbin/ntpd -u ntp:ntp $OPTIONS (code=exited, status=0/SUCCESS)
  Main PID: 11480 (ntpd)
     Tasks: 1
    Memory: 580.0K
   CGroup: /system.slice/ntpd.service
           +-11480 /usr/sbin/ntpd -u ntp:ntp -g

Jul 03 22:05:46 luna.eval.local ntpd[11480]: ntp_io: estimated max descriptors: 1024, initial
socket boundary: 16
Jul 03 22:05:46 luna.eval.local ntpd[11480]: Listen and drop on 0 v4wildcard 0.0.0.0 UDP 123
Jul 03 22:05:46 luna.eval.local ntpd[11480]: Listen and drop on 1 v6wildcard :: UDP 123
Jul 03 22:05:46 luna.eval.local ntpd[11480]: Listen normally on 2 lo 127.0.0.1 UDP 123
Jul 03 22:05:46 luna.eval.local ntpd[11480]: Listen normally on 3 ens34 195.214.80.117 UDP 123
Jul 03 22:05:46 luna.eval.local ntpd[11480]: Listen normally on 4 docker0 172.17.0.1 UDP 123
Jul 03 22:05:46 luna.eval.local ntpd[11480]: Listening on routing socket on fd #21 for interface
updates
Jul 03 22:05:46 luna.eval.local ntpd[11480]: 0.0.0.0 c016 06 restart
Jul 03 22:05:46 luna.eval.local ntpd[11480]: 0.0.0.0 c012 02 freq_set kernel 0.000 PPM
Jul 03 22:05:46 luna.eval.local ntpd[11480]: 0.0.0.0 c011 01 freq_not_set
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$ ntpq -p
      remote           refid      st t when poll reach  delay  offset  jitter
=====
dcon          .INIT.         16 u    2   64    0   0.000   0.000   0.000
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$
```



If this is not the case, ...

- make sure the file `/etc/ntp.conf` has the correct entry for the ntp server (see next page)
- re-start `ntpd`
- re-run `ntpq -p`  
`dcon` must show up as shown above!

```

[root@luna ~]#
[root@luna ~]#
[root@luna ~]# more /etc/ntp.conf
# Ansible managed: /usr/local/src/ui/ansible/roles/CentOS_7_baseline_install/templates/
ntp.conf.j2 by 1000 on luna.eval.local
# permit the source to query or modify the service on this system.
restrict default nomodify notrap nopeer noquery

# Permit all access over the loopback interface. This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1
restrict ::1

# Hosts on local network are less restricted.
#restrict 192.168.2.0 mask 255.255.255.0 nomodify notrap

# Use NTP servers configured via ECS installer
server 195.214.80.1 iburst

#broadcast 192.168.2.255 autokey # broadcast server
#broadcastclient # broadcast client
#broadcast 224.0.1.1 autokey # multicast server
#multicastclient 224.0.1.1 # multicast client
#manycastserver 239.255.254.254 # manycast server
#manycastclient 239.255.254.254 autokey # manycast client

# Enable public key cryptography.
#crypto

includefile /etc/ntp/crypto/pw

# Key file containing the keys and key identifiers used when operating
# with symmetric key cryptography.
keys /etc/ntp/keys

# Specify the key identifiers which are trusted.
#trustedkey 4 8 42

# Specify the key identifier to use with the ntpdc utility.
#requestkey 8

# Specify the key identifier to use with the ntpq utility.
#controlkey 8

# Enable writing of statistics records.
#statistics clockstats cryptostats loopstats peerstats

# Disable the monitoring facility to prevent amplification attacks using ntpdc
# monlist command when default restrict does not include the noquery flag. See
# CVE-2013-5211 for more details.
# Note: Monitoring will not be disabled with the limited restriction flag.
disable monitor
[root@luna ~]#
[root@luna ~]#
[root@luna ~]#
[root@luna ~]# systemctl stop ntpd
[root@luna ~]#
[root@luna ~]#
[root@luna ~]#
[root@luna ~]# systemctl start ntpd
[root@luna ~]#
[root@luna ~]#
[root@luna ~]#
[root@luna ~]# ntpq -p
      remote          refid          st t when poll reach  delay  offset  jitter
=====
dcon          .INIT.          16 u   52   64    0   0.000   0.000   0.000
[root@luna ~]#
[root@luna ~]#

```

## IV-4. Running the script 'island-step2'

Prerequisites:

- Cut the internet connection. Consequently luna must now use the local NTP server!
- The local NTP server must be accessible!
- The NTP server himself (here: dcon) must have a working internet connection!



Once again, the connection to the local NTP server is mandatory! - Do not proceed without one!

The whole process will take about 15-30 minutes, depending on your system.

```
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$ pwd
/home/admin/ECS-CommunityEdition
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$ island-step2

PLAY [Common | Ping data nodes before doing anything else]
*****

TASK [ping]
*****
ok: [195.214.80.117]

PLAY [Installer | Gather facts and slice into OS groups]
*****

TASK [group_by]
*****
ok: [195.214.80.117]

PLAY [CentOS 7 | Configure access]
*****

TASK [CentOS_7_configure_ssh : CentOS 7 | Distribute ed25519 ssh key]
*****

TASK [CentOS_7_configure_ssh : CentOS 7 | Distribute rsa ssh key]
*****
ok: [195.214.80.117]

TASK [CentOS_7_configure_ssh : CentOS 7 | Disable SSH UseDNS]
*****
ok: [195.214.80.117]

TASK [CentOS_7_configure_ssh : CentOS 7 | Disable requiretty]
*****
ok: [195.214.80.117]

TASK [CentOS_7_configure_ssh : CentOS 7 | Disable sudo password reverification for admin group]
*****
ok: [195.214.80.117]

TASK [CentOS_7_configure_ssh : CentOS 7 | Disable sudo password reverification for wheel group]
*****
ok: [195.214.80.117]

TASK [firewalld_configure_access : Firewalld | Ensure service is started]
*****
ok: [195.214.80.117]

.....
```



```
.....

TASK [firewalld_configure_access : Firewallld | Add install node to firewalld trusted zone]
*****
ok: [195.214.80.117]

TASK [firewalld_configure_access : Firewallld | Add all data nodes to firewalld trusted zone]
*****
ok: [195.214.80.117] => (item=195.214.80.117)
ok: [195.214.80.117] => (item=172.17.0.1)

TASK [firewalld_configure_access : Firewallld | Whitelist management prefixes]
*****
ok: [195.214.80.117] => (item=0.0.0.0/0)

TASK [firewalld_configure_access : Firewallld | Add all public service ports to firewalld
public zone] *****
ok: [195.214.80.117] => (item=3218/tcp)
ok: [195.214.80.117] => (item=9020-9025/tcp)
ok: [195.214.80.117] => (item=9040/tcp)

TASK [firewalld_configure_access : Firewallld | Ensure service is started]
*****
changed: [195.214.80.117]

PLAY [Common | Configure hostnames]
*****

TASK [common_set_hostname : include_vars]
*****
ok: [195.214.80.117]

TASK [common_set_hostname : Common | Find node hostname]
*****
ok: [195.214.80.117] => (item=(0, u'195.214.80.117'))

TASK [common_set_hostname : Common | Set node hostname]
*****
ok: [195.214.80.117]

PLAY [Common | Configure /etc/hosts]
*****

TASK [common_etc_hosts : Common | Add install node to /etc/hosts]
*****
changed: [195.214.80.117] => (item=195.214.80.117)

TASK [common_etc_hosts : Common | Add data nodes to /etc/hosts]
*****
ok: [195.214.80.117] => (item=195.214.80.117)

PLAY [Common | Test inter-node access]
*****

TASK [common_access_test : Common | Check node connectivity by IP]
*****
ok: [195.214.80.117] => (item=195.214.80.117)
ok: [195.214.80.117] => (item=172.17.0.1)

TASK [common_access_test : Common | Check node connectivity by short name]
*****
ok: [195.214.80.117] => (item=luna)

TASK [common_access_test : Common | Check node connectivity by fqdn]
*****
ok: [195.214.80.117] => (item=luna)

PLAY RECAP
*****
195.214.80.117 : ok=21 changed=2 unreachable=0 failed=0

Playbook run took 0 days, 0 hours, 0 minutes, 12 seconds

.....
```



```
.....

PLAY [Common | Ping data nodes before doing anything else]
*****

TASK [ping]
*****
ok: [195.214.80.117]

PLAY [Installer | Slice nodes into OS groups]
*****

TASK [group_by]
*****
ok: [195.214.80.117]

PLAY [Installer | Perform preflight check]
*****

TASK [common_collect_facts : Common | Create custom facts directory]
*****
ok: [195.214.80.117]

TASK [common_collect_facts : Common | Insert data_node.fact file]
*****
ok: [195.214.80.117]

TASK [common_collect_facts : Common | Reload facts to pick up new items]
*****
ok: [195.214.80.117]

TASK [common_baseline_check : include_vars]
*****
ok: [195.214.80.117]

TASK [common_baseline_check : Common | Check RAM size]
*****

TASK [common_baseline_check : Common | Check CPU architecture]
*****

TASK [common_baseline_check : Common | Validate OS distribution]
*****

TASK [common_baseline_check : Common | (Optional) Check UTC Timezone]
*****
ok: [195.214.80.117] => {
    "msg": "Timezone is not set to UTC - It is recommended to use the UTC timezone"
}

TASK [common_baseline_check : Common | Make sure /data directory does not exist]
*****
ok: [195.214.80.117]

TASK [common_baseline_check : fail]
*****

TASK [common_baseline_check : Common | Make sure /host directory does not exist]
*****
ok: [195.214.80.117]

TASK [common_baseline_check : fail]
*****

TASK [common_baseline_check : Common | Make sure block device(s) exist on node]
*****
ok: [195.214.80.117] => (item=/dev/sdb)

TASK [common_baseline_check : fail]
*****

TASK [common_baseline_check : Common | Make sure block device(s) are at least 100GB]
*****

.....
```





```
.....

TASK [common_baseline_check : Common | Make sure block device(s) are unpartitioned]
*****
ok: [195.214.80.117] => (item=/dev/sdb)

TASK [common_baseline_check : fail]
*****

TASK [common_baseline_check : Common | Check for listening layer 4 ports]
*****
changed: [195.214.80.117]

TASK [common_baseline_check : Common | Report any conflicts with published ECS ports]
*****

TASK [common_baseline_check : Common | Report any conflicts with internal ECS ports]
*****

PLAY RECAP
*****
195.214.80.117 : ok=12 changed=1 unreachable=0 failed=0

Playbook run took 0 days, 0 hours, 0 minutes, 7 seconds

PLAY [Common | Ping data nodes before doing anything else]
*****

TASK [ping]
*****
ok: [195.214.80.117]

PLAY [Installer | Slice nodes into OS groups for package install]
*****

TASK [group_by]
*****
ok: [195.214.80.117]

PLAY [CentOS 7 | Synchronize cache]
*****

TASK [CentOS_7_sync_caches_prep : CentOS 7 | Fetch torrent package names]
*****

TASK [CentOS_7_sync_caches_prep : CentOS 7 | Make yum cache paths for torrent packages]
*****

TASK [CentOS_7_sync_caches_prep : CentOS 7 | Copy torrent packages to nodes]
*****

TASK [CentOS_7_sync_caches_prep : CentOS 7 | Install torrent packages on nodes]
*****

TASK [common_sync_caches_by_torrent : include_vars]
*****
ok: [195.214.80.117]

TASK [common_sync_caches_by_torrent : Selinux | Set selinux permissive]
*****
ok: [195.214.80.117]

TASK [common_sync_caches_by_torrent : Common | Create ecs-install host directory on nodes]
*****

TASK [common_sync_caches_by_torrent : Common | Create ecs-install cache directory on nodes]
*****

TASK [common_sync_caches_by_torrent : Common | Copy cache torrent digest to nodes]
*****

TASK [common_sync_caches_by_torrent : Common | Torrent sync caches with nodes]
*****
.....
```



```
.....

PLAY [CentOS 7 | Install cached packages]
*****

TASK [CentOS_7_baseline_install : include_vars]
*****
ok: [195.214.80.117]

TASK [CentOS_7_baseline_install : CentOS 7 | Set selinux permissive]
*****
ok: [195.214.80.117]

TASK [CentOS_7_baseline_install : CentOS 7 | Add SSL proxy cert]
*****

TASK [CentOS_7_baseline_install : CentOS 7 | Stop and disable unused services]
*****
ok: [195.214.80.117] => (item=mta)

TASK [CentOS_7_baseline_install : CentOS 7 | Configure DNS]
*****
ok: [195.214.80.117]

TASK [CentOS_7_baseline_install : CentOS 7 | Unpack package_cache.tgz]
*****

TASK [CentOS_7_baseline_install : CentOS 7 | Install all packages in the cache]
*****

TASK [CentOS_7_baseline_install : CentOS 7 | Configure ntp]
*****
ok: [195.214.80.117]

TASK [CentOS_7_baseline_install : CentOS 7 | Start and enable services]
*****
ok: [195.214.80.117] => (item=docker)
ok: [195.214.80.117] => (item=ntp)
ok: [195.214.80.117] => (item=firewall)

PLAY [Installer | Disable torrent ffx]
*****

TASK [file]
*****

PLAY RECAP
*****
195.214.80.117      : ok=10   changed=0    unreachable=0    failed=0
localhost         : ok=0    changed=0    unreachable=0    failed=0

Playbook run took 0 days, 0 hours, 0 minutes, 5 seconds

PLAY [Common | Ping data nodes before doing anything else]
*****

TASK [ping]
*****
ok: [195.214.80.117]

PLAY [Installer | Slice nodes into OS groups]
*****

TASK [group_by]
*****
ok: [195.214.80.117]

.....
```

```
.....

PLAY [CentOS 7 | Conditionally reboot nodes]
*****

TASK [CentOS_7_reboot : CentOS 7 | Check needs-restarting]
*****

TASK [CentOS_7_reboot : CentOS 7 | Set reboot action on nodes that need restarting]
*****

TASK [CentOS_7_reboot : CentOS 7 | Check if install node also needs restarting]
*****

TASK [CentOS_7_reboot : CentOS 7 | Reboot node(s)]
*****

TASK [CentOS_7_reboot : CentOS 7 | Wait for node(s) to reboot]
*****

PLAY RECAP
*****
195.214.80.117      : ok=2    changed=0    unreachable=0    failed=0

Playbook run took 0 days, 0 hours, 0 minutes, 1 seconds

Waiting for nodes to become reachable... (CTRL-C to break)
195.214.80.117 | SUCCESS => {"changed": false, "failed": false, "ping": "pong"}

PLAY [Common | Ping data nodes before doing anything else]
*****

TASK [ping]
*****
ok: [195.214.80.117]

PLAY [Common | Install ECS]
*****

TASK [common_baseline_install : include_vars]
*****
ok: [195.214.80.117]

TASK [common_baseline_install : include_vars]
*****
ok: [195.214.80.117]

TASK [common_baseline_install : Common | Create and modify paths and semaphores for docker
containers] *****
changed: [195.214.80.117] => (item=/ecs)
changed: [195.214.80.117] => (item=/ecs)
changed: [195.214.80.117] => (item=/ecs)
changed: [195.214.80.117] => (item=/ecs)
changed: [195.214.80.117] => (item=/ecs)
changed: [195.214.80.117] => (item=/ecs)
changed: [195.214.80.117] => (item=/ecs)

TASK [common_baseline_install : Common | Generate network.json]
*****
changed: [195.214.80.117]

TASK [common_baseline_install : Common | Generate object-main_network.json]
*****
changed: [195.214.80.117]

TASK [common_baseline_install : Common | Generate id.json]
*****
changed: [195.214.80.117]

TASK [common_baseline_install : Common | Generate agent.json]
*****
changed: [195.214.80.117]

.....
```





.....

```
TASK [common_deploy : Common | Configure SSM Object properties: FBL-Highwater]
*****
changed: [195.214.80.117]

TASK [common_deploy : Common | Configure SSM Object properties: FBL-Lowwater]
*****
changed: [195.214.80.117]

TASK [common_deploy : Common | Configure Common Object properties:
allowAllocationOnIgnoredPartitions] *****
changed: [195.214.80.117]

TASK [common_deploy : Common | Configure Common Object properties: System NumDtPerCoS]
*****
changed: [195.214.80.117]

TASK [common_deploy : Common | Configure Common Object properties: User NumDtPerCoS]
*****
changed: [195.214.80.117]

TASK [common_deploy : Common | Configure CM Object properties: Disable Minimum Node Count]
*****
changed: [195.214.80.117]

TASK [common_deploy : Common | Configure CM Read Page Cache: Max]
*****
changed: [195.214.80.117]

TASK [common_deploy : Common | Configure CM Read Page Cache: Core]
*****
changed: [195.214.80.117]

TASK [common_deploy : Common | Configure CM DT Write IO: OnHeap Buffers]
*****
changed: [195.214.80.117]

TASK [common_deploy : Common | Configure CM Geo Shipping IO: OnHeap Buffers]
*****
ok: [195.214.80.117]

TASK [common_deploy : Common | Configure georeceiver initialBufferNumOnHeap (1/2)]
*****
changed: [195.214.80.117]

TASK [common_deploy : Common | Configure georeceiver initialBufferNumOnHeap (2/2)]
*****
ok: [195.214.80.117]

TASK [common_deploy : Common | Increase memory for dtquery]
*****
ok: [195.214.80.117]

TASK [common_deploy : Common | Increase memory for transformsvc]
*****
ok: [195.214.80.117]

TASK [common_deploy : Common | Increase memory for objcontrolsvc]
*****
ok: [195.214.80.117]

TASK [common_deploy : Common | vnest enable UseSeparateThreadPools]
*****
changed: [195.214.80.117]

TASK [common_deploy : Common | Increase memory for sr]
*****
ok: [195.214.80.117]
```

.....



```
.....

TASK [common_deploy : Common | Merge configurations into configuration data container]
*****
changed: [195.214.80.117] => (item=/opt/storageos/conf/ssm.object.properties)
changed: [195.214.80.117] => (item=/opt/storageos/conf/ssm.object.properties)
changed: [195.214.80.117] => (item=/opt/storageos/conf/ssm.object.properties)
changed: [195.214.80.117] => (item=/opt/storageos/conf/ssm.object.properties)
changed: [195.214.80.117] => (item=/opt/storageos/conf/ssm.object.properties)
changed: [195.214.80.117] => (item=/opt/storageos/conf/ssm.object.properties)
changed: [195.214.80.117] => (item=/opt/storageos/conf/ssm.object.properties)
changed: [195.214.80.117] => (item=/opt/storageos/conf/ssm.object.properties)
changed: [195.214.80.117] => (item=/opt/storageos/conf/ssm.object.properties)

TASK [common_deploy : Common | Create the ECS StorageOS container]
*****
changed: [195.214.80.117]

PLAY [Installer | Slice nodes into OS groups for init scripts]
*****

TASK [group_by]
*****
ok: [195.214.80.117]

PLAY [CentOS 7 | Install init scripts]
*****

TASK [CentOS_7_service_policies : CentOS 7 | Insert ecs.service into systemd]
*****
changed: [195.214.80.117]

PLAY RECAP
*****
195.214.80.117 : ok=42 changed=31 unreachable=0 failed=0

Playbook run took 0 days, 0 hours, 0 minutes, 33 seconds

PLAY [Common | Ping data nodes before doing anything else]
*****

TASK [ping]
*****
ok: [195.214.80.117]

PLAY [Common | Start and enable ECS services]
*****

TASK [common_start_enable_service : Common | Enable ECS service and start ECS]
*****
changed: [195.214.80.117]

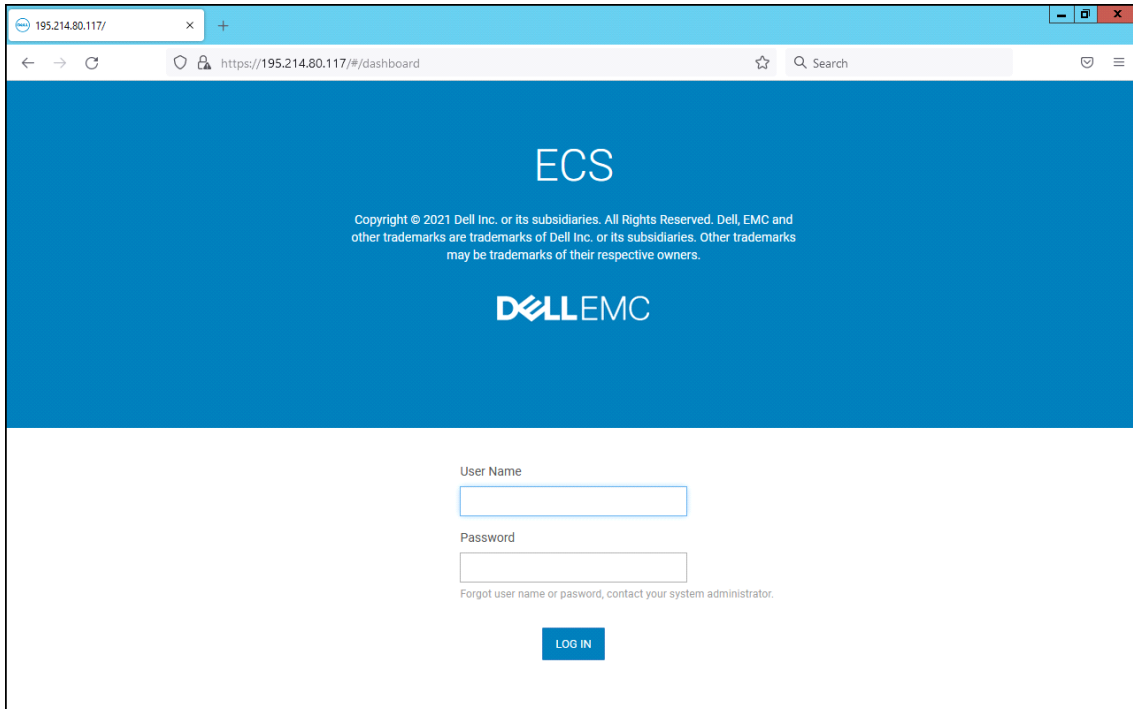
PLAY RECAP
*****
195.214.80.117 : ok=2 changed=1 unreachable=0 failed=0

Playbook run took 0 days, 0 hours, 0 minutes, 1 seconds

> Please wait for 30 minutes before running next step for services to bootstrap
[admin@luna ECS-CommunityEdition]$
```

You can proceed if you can access the admin GUI via:

[https://ecs-ce\\_ip\\_address](https://ecs-ce_ip_address)



**Do not login yet!** - the first thing you have to do is to change the password. However, if you do so, the last script will not run successfully.

## IV-5. Running the script 'island-step3'

This will take about 20 mins.

```
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$ pwd
/home/admin/ECS-CommunityEdition
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$
[admin@luna ECS-CommunityEdition]$ island-step3
> Pinging Management API Endpoint until ready
> Pinging endpoint 195.214.80.117... (CTRL-C to break)
> PONG: api_endpoint=195.214.80.117 username=root diag_endpoint=195.214.80.117 dt_total=0
dt_unready=0 dt_unknown=0

> Installing licensing in ECS VDC(s)
> Using default license
> Adding licensing to VDC: vdc1
> OK
> Added default license to ECS

> Pinging Management API Endpoint until ready
> Pinging endpoint 195.214.80.117... (CTRL-C to break)
> PONG: api_endpoint=195.214.80.117 username=root diag_endpoint=195.214.80.117 dt_total=0
dt_unready=0 dt_unknown=0

> Creating Storage Pool: vdc1/sp1
> OK
> Adding Data Stores to Storage Pool:
> vdc1/sp1/195.214.80.117

> Waiting for Storagepool to get ready , takes 15 minutes
> Pinging endpoint 195.214.80.117... (CTRL-C to break)
> PONG: api_endpoint=195.214.80.117 username=root diag_endpoint=195.214.80.117 dt_total=128
dt_unready=0 dt_unknown=0

> Waiting for Storagepool to get ready , takes 15 minutes
> Pinging endpoint 195.214.80.117... (CTRL-C to break)
> PONG: api_endpoint=195.214.80.117 username=root diag_endpoint=195.214.80.117 dt_total=384
dt_unready=0 dt_unknown=0

> Waiting for Storagepool to get ready , takes 15 minutes
> Pinging endpoint 195.214.80.117... (CTRL-C to break)
> PONG: api_endpoint=195.214.80.117 username=root diag_endpoint=195.214.80.117 dt_total=384
dt_unready=0 dt_unknown=0

> Creating all VDCs...
> vdc1
> Created all VDCs

> Waiting for all VDCs to online and become active...
> Checking vdc1:
> OK: VDC online

> Pinging Management API Endpoint until ready
> Pinging endpoint 195.214.80.117... (CTRL-C to break)
> PONG: api_endpoint=195.214.80.117 username=root diag_endpoint=195.214.80.117 dt_total=384
dt_unready=0 dt_unknown=0

> Creating replication group rg1
> Generating zone mappings for rg1/vdc1
> sp1
> Applying mappings
> OK
> Created all Replication Groups

.....
```



```

.....

> Pinging Management API Endpoint until ready
> Pinging endpoint 195.214.80.117... (CTRL-C to break)
> PONG: api_endpoint=195.214.80.117 username=root diag_endpoint=195.214.80.117 dt_total=416
dt_unready=0 dt_unknown=0

> Creating all configured management users:
> admin1
> monitor1
> Created all configured management users

> Pinging Management API Endpoint until ready
> Pinging endpoint 195.214.80.117... (CTRL-C to break)
> PONG: api_endpoint=195.214.80.117 username=root diag_endpoint=195.214.80.117 dt_total=416
dt_unready=0 dt_unknown=0

> Creating all Namespaces
> Adding namespace ns1
> OK
> Created all configured namespaces

> Pinging Management API Endpoint until ready
> Pinging endpoint 195.214.80.117... (CTRL-C to break)
> PONG: api_endpoint=195.214.80.117 username=root diag_endpoint=195.214.80.117 dt_total=416
dt_unready=0 dt_unknown=0

> Creating all configured object users:
> Creating all object users
> Creating 'object_admin1' in namespace 'ns1'
> Waiting for 'object_admin1' to become editable
> OK
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's S3 credentials
> Adding object_admin1's Swift credentials
> Creating 'object_user1' in namespace 'ns1'
> Waiting for 'object_user1' to become editable
> OK
> Adding object_user1's S3 credentials
> Adding object_user1's Swift credentials
> Created all configured object users

> Pinging Management API Endpoint until ready
> Pinging endpoint 195.214.80.117... (CTRL-C to break)
> PONG: api_endpoint=195.214.80.117 username=root diag_endpoint=195.214.80.117 dt_total=416
dt_unready=0 dt_unknown=0

> Creating all buckets
> Creating bucket: bucket1
> OK
> Created all configured buckets

[admin@luna ECS-CommunityEdition]$

```

This is the end of the installation. The whole procedure needs slightly less than 1 hour.