I created this document to keep track of my experimental installation of ECS 2.2.0.1 I made in March 2016. It is based on the official installation instructions from github.com but contains more details and comments for specific commands' output and description of mistakes I made and errors I got.

All info here is provided as is.

Content

1. References ............................................................................................................. 4
2. CentOS preparation ............................................................................................... 4
   2.1. Virtual machines ........................................................................................... 4
   2.2. Install additional packages ............................................................................ 5
   2.3. Install SSL certificate ................................................................................... 7
   2.4. Check disk and eth interface ........................................................................ 7
3. Install ECS .............................................................................................................. 8
   3.1. Installation script ........................................................................................... 8
   3.2. Check reboot persistence ............................................................................... 10
   3.3. Problems I met during the installation ........................................................... 12
       3.3.1. Problem with DockerCommandLineFlags .............................................. 12
       3.3.2. Problem with argparse ........................................................................ 13
       3.3.3. Problem DNS ....................................................................................... 13
       3.3.4. Problem with SSL certificate ............................................................... 14
       3.3.5. Problem with authentication server ..................................................... 14
4. Configure ECS ......................................................................................................... 15
   4.1. Configure by script ........................................................................................ 15
   4.2. Login .............................................................................................................. 15
   4.3. License .......................................................................................................... 16
   4.4. Storage Group, VDC and Replication Group ............................................... 17
   4.5. Add management user .................................................................................. 20
   4.6. Create namespace ...................................................................................... 21
   4.7. Add object user ............................................................................................ 22
   4.8. Create bucket ............................................................................................... 23
5. Check data access ................................................................................................. 24
   5.1. S3 protocol ..................................................................................................... 24
   5.2. Atmos protocol .............................................................................................. 26
   5.3. Swift protocol ............................................................................................... 31
6. Appendix ............................................................................................................... 34
   6.1. Configure network in CentOS ..................................................................... 34
   6.2. Create VMware template in CentOS ............................................................ 35
   6.3. Add new disk to CentOS .............................................................................. 37
   6.4. Extend a disk in CentOS .............................................................................. 38
1. References

Main instruction. Very brief and doesn't reveal any pitfalls.

Great single-node installation article from Keith Anderson. I found it when I already installed multi-node ECS and was upset that didn’t’ find it earlier. Some advices save a lot of time.

EMC SSL instructions:

Single-node installation instruction with very useful comments.
https://community.emc.com/docs/DOC-45387

ECS 2.2+ Swift access via Cyberduck problem is resolved.
https://asdjira.isus.emc.com:8443/browse/STORAGE-11753

2. CentOS preparation

2.1. Virtual machines

- Four VMs are created
- Check details in the screenshot below
Actions below have to be done on all **4 CentOS nodes**

**2.2. Install additional packages**

- Check OS version

  ```
  [root@ecsnode1 ~]# hostnamectl
  Static hostname: ecsnode1.vipr.local
    Icon name: computer-vm
    Chassis: vm
    Machine ID: 00b827eeef474cf5960ca71a30e0b320
    Boot ID: 385b2f3220af403fa9e87443904042eb
  Virtualization: vmware
  Operating System: CentOS Linux 7 (Core)
    CPE OS Name: cpe:/o:centos:centos:7
    Kernel: Linux 3.10.0-327.el7.x86_64
    Architecture: x86-64
  ```
- Update **yum**

  ```
  [root@ecsnode1 ECS]# yum clean all
  [root@ecsnode1 ECS]# yum check (be patient, it takes some time)
  [root@ecsnode1 ECS]# yum erase apf
  [root@ecsnode1 ECS]# yum upgrade
  ```

- **Install git**

  ```
  [root@ecsnode1 ECS]# yum install git
  ```

- **Clone ECS git repository**

  ```
  [root@ecsnode1 ~]# mkdir /ECS
  [root@ecsnode1 ~]# cd /ECS/
  [root@ecsnode1 ECS]# git clone https://github.com/EMCECS/ECS-CommunityEdition
  ```

- **Install EPEL**

  **Note:** Extra Packages for Enterprise Linux (EPEL) - is a Fedora Special Interest Group that creates, maintains, and manages a set of additional packages for Enterprise Linux, including, but not limited to RHEL, CentOS and Scientific Linux (SL), Oracle Linux (OL).

  ```
  [root@ecsnode1 ecs-multi-node]# yum install epel-release
  ```

- **Install pip**

  ```
  [root@ecsnode1 ~]# yum install -y python-pip
  ```

- **In case of error - update certificates**

  ```
  [root@ecsnode1 ~]# yum install -y python-pip
  Failed to set locale, defaulting to C
  Loaded plugins: fastestmirror
  Loading mirror speeds from cached hostfile
  Error: Cannot retrieve metalink for repository: epel. Please verify its path and try again
  ```

  ```
  [root@ecsnode1 ~]# yum upgrade ca-certificates --disablerepo=epel
  ...
  Dependencies Resolved
  ==============================================================
  Package    Arch  Version                Repository  Size
  ==============================================================
  Updating:
  ca-certificates  noarch  2015.2.4-65.0.1.el6_6  base    1.3 M
  ...
  Complete!
  ```
• Install `argparse` with pip

```bash
[root@ecsnode1 ~]# pip install argparse
Collecting argparse
/usr/lib/python2.6/site-packages/pip/_vendor/requests/packages/urllib3/util/ssl_.py:90:
InsecurePlatformWarning: A true SSLContext object is not available. This prevents urllib3 from configuring SSL appropriately and may cause certain SSL connections to fail. For more information, see https://urllib3.readthedocs.org/en/latest/security.html#insecureplatformwarning.
InsecurePlatformWarning
Downloading argparse-1.4.0-py2.py3-none-any.whl
Installing collected packages: argparse
Successfully installed argparse-1.4.0
```

2.3. **Install SSL certificate**

• If you are in EMC network - add SSL certificate

```bash
[root@ecsnode1 ECS]# cp /ECS/ECS-CommunityEdition/emc-ssl-cert/emc_ssl.pem /etc/pki/ca-trust/source/anchors/
[root@ecsnode1 ecs-multi-node]# update-ca-trust extract
```

2.4. **Check disk and eth interface**

• Check unpartitioned disk name

```bash
[root@ecsnode1 ~]# fdisk -l
```

```
Disk /dev/sda: 17.2 GB, 17179869184 bytes, 33554432 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x00092f81

Device  Boot  Start  End   Blocks  Id  System
/dev/sda1  *  2048  1026047  512000  83  Linux
/dev/sda2  1026048  33554431 16264192 8e  Linux LVM

Disk /dev/mapper/centos-root: 14.9 GB, 14889779200 bytes, 29081600 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/centos-swap: 1719 MB, 1719664640 bytes, 3358720 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```
Disk /dev/sdb: 108.4 GB, 108447924224 bytes, 211812352 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

- Check network adapter name

[root@ecsnode3 ~]# ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.76.246.145 netmask 255.255.255.0 broadcast 10.76.246.255
    inet6 fe80::250:56ff:feb8:c995 prefixlen 64 scopeid 0x20<link>
    ether 00:50:56:b8:c9:95 txqueuelen 1000 (Ethernet)
    RX packets 1571962 bytes 1298801446 (1.2 GiB)
    RX errors 0 dropped 819784 overruns 0 frame 0
    TX packets 318 bytes 27355 (26.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

3. Install ECS

3.1. Installation script

- Lets try to install multi-node ECS
- Execute the command below on all 4 nodes
- I ran the 1st node a little bit before cause it has to provide authentication service for three of others
- Important – be sure that you get an AuthToken when script is completed
- Authentication retries are normal. Just wait till the script is successfully ended. It can take up 20 minutes.

[root@ecsnode1 ecs-multi-node]# cd /ECS/ECS-CommunityEdition/ecs-multi-node/

[root@ecsnode1 ecs-multi-node]# python step1_ecs_multinode_install.py --ips 10.76.246.143 10.76.246.144 10.76.246.145 10.76.246.146 --hostnames ecsnode1 ecsnode2 ecsnode3 ecsnode4 --ethadapter ens160 --disks sdb

...* Are you trying to connect to a TLS-enabled daemon without TLS?
* Is your docker daemon up and running?
[01/Apr/2016 12:50:41] INFO [root:215] Updating the /etc/hosts file with the IP-Hostname of each one of the DataNodes in the cluster
[01/Apr/2016 12:50:41] INFO [root:248] Partitioning the disk '/dev/sdb'

meta-data=/dev/sdb1  isize=256  agcount=4, agsize=6619072 blks
  =  sectsz=512  attr=2, projid32bit=1
  =  crc=0  finobt=0
data  =  bsize=4096  blocks=26476288, imaxpct=25
  =  sunit=0  swidth=0 blks

name   =version 2  bsize=4096  ascii-ci=0  ftype=0
log    =internal log  bsize=4096  blocks=12927, version=2
  =  sectsz=512  sunit=0  blks, lazy-crc=

realtime =none  extsz=4096  blocks=0, rtextents=0

[01/Apr/2016 12:50:43] INFO [root:349] Executing the additional preparation script in '/dev/sdb1'

create file 0000
create file 0001
create file 0002
create file 0003
create file 0004
create file 0005
create file 0006
create file 0007
create file 0008

Phase 1 - find and verify superblock...
Phase 2 - using internal log
  - scan filesystem freespace and inode maps...
  - found root inode chunk
Phase 3 - for each AG...
  - scan (but don't clear) agi unlinked lists...
    - process known inodes and perform inode discovery...
      - agno = 0
      - agno = 1
      - agno = 2
      - agno = 3
    - process newly discovered inodes...
Phase 4 - check for duplicate blocks...
  - setting up duplicate extent list...
  - check for inodes claiming duplicate blocks...
    - agno = 0
    - agno = 1
    - agno = 2
    - agno = 3

No modify flag set, skipping phase 5
Phase 6 - check inode connectivity...
  - traversing filesystem...
- traversal finished ...
- moving disconnected inodes to lost+found ...

Phase 7 - verify link counts...
No modify flag set, skipping filesystem flush and exiting.
/dev/sdb1 uuid-1 xfs rw,noatime,seclabel,attr2(inode64,quota 0 0

... Trying to pull repository docker.io/emccorp/ecs-software-2.2 ... latest: Pulling from emccorp/ecs-software-2.2
57f711ac00c0: Pull complete
7e753bf71d17: Pull complete
7cf143d0c7ec: Pull complete
Digest: sha256:076ab6cf5cb3ee61be62fbdcd145d209e474c36fe70f76cbb85e169f1acaa3d
Status: Downloaded newer image for docker.io/emccorp/ecs-software-2.2:latest

66dbc6b654a1f8b719a8433a525c67f4bbb729ed5bdc612a70a653c5828a5d1e
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
66dbc6b654a1 emccorp/ecs-software-2.2:latest "/opt/vipr/boot/boot." 1 seconds ago Up Less
than a second
ecsmultinode

Executing getAuthToken: curl -i -k https://10.76.246.143:4443/login -u root:ChangeMe
% Total % Received % Xferd Average Speed Time Time Time Current
 0 0 0 0 0 0 0 --:-- --:-- --:-- --:--
curl: (7) Failed connect to 10.76.246.143:4443;
Connection refused

Executing getAuthToken: curl -i -k https://10.76.246.143:4443/login -u root:ChangeMe
% Total % Received % Xferd Average Speed Time Time Time Current
 0 0 0 0 0 0 0 --:-- --:-- --:-- --:--
100 93 100 93 0 0 87 0:00:00 0:00:01 87
Auth Token
BAAccD2DZUtqakrbrTQTWhom1Q5VG1IRHFSM20wpQMAjAQASHYybjpzsdG9yYWldb3M6VmlvdHVVbErHdgFDZW50XZjEYXRh0MhMjk1ZDZjLTQ0NzktdCxCiZW3ZLTZjMjkzNDY0MWY5YQIADE0NTk1MDcyMzAtQDAC51cm46VG9rZW46OWM5NDU0Y2QtNjkyZS00Njc0LWljNmEtZDk0NzAzYzQ0YzQxAgAc0A

[01/Apr/2016 13:46:57] INFO [root:702] Step 1 Completed. Navigate to the administrator website that is
available from any of the ECS data nodes. The ECS administrative portal can be accessed from port 443. For example: https://ecs-node-external-ip-address. The website may take a few minutes to
become available.

3.2. Check reboot persistence

Do that on all ECS nodes

- Check if data disk will be mounted after reboot

[root@ecsnode1 ~]# cat /etc/fstab
• Check and enable the docker service

[root@ecsnode1 ~]# systemctl docker.service status
Unknown operation 'docker.service'.
[root@ecsnode1 ~]# systemctl enable docker.service
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.

[root@ecsnode1 ~]# systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2016-04-01 12:51:11 MSK; 7h ago
   Docs: http://docs.docker.com
   Main PID: 10450 (docker)
   Memory: 18.4M
   CGroup: /system.slice/docker.service
          └─10450 /usr/bin/docker daemon

Apr 01 13:24:12 ecsnode1.vipr.local docker[10450]: time="2016-04-01T13:24:12.415524...0"
Apr 01 13:24:12 ecsnode1.vipr.local docker[10450]: 2016/04/01 13:24:12 http: respon...
Hint: Some lines were ellipsized, use -l to show in full.

[root@ecsnode1 ~]# systemctl enable docker.service
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to
/usr/lib/systemd/system/docker.service.

[root@ecsnode1 ~]# systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2016-04-01 12:51:11 MSK; 7h ago
   Docs: http://docs.docker.com
   Main PID: 10450 (docker)
   CGroup: /system.slice/docker.service
          └─10450 /usr/bin/docker daemon

• Enable systemd service

[root@ecsnode1 ~]# cp /ECS/ECS-CommunityEdition/ecs-multinode/systemd/docker.ecsmultinode.service /usr/lib/systemd/system/

[root@ecsnode1 ~]# systemctl enable docker.ecsmultinode.service
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.ecsmultinode.service to /usr/lib/systemd/system/docker.ecsmultinode.service.

[root@ecsnode1 ~]# systemctl daemon-reload

Notes:
• After the reboot wait at least 5 minutes for things to start
• Check ecsmultinode container is running

[root@ecsnode1 ~]# docker ps
CONTAINER ID          IMAGE                       COMMAND                  CREATED       STATUS    PORTS                 NAMES
66cdbe654a1          emccorp/ecs-software-2.2:latest  "/opt/vipr/boot/boot."  7 hours ago   Up 7 hours 0.0.0.0:*  ecsmultinode

• Check if port 443 is listening.

[root@ecsnode1 ~]# netstat -an | grep :443
tcp        0      0 10.76.246.143:443    0.0.0.0:*            LISTEN
tcp6       0      0 10.76.246.143:9091  10.76.246.146:44385    ESTABLISHED
tcp6       0      0 10.76.246.143:9201  10.76.246.145:44395    ESTABLISHED

3.3. Problems I met during the installation

3.3.1. Problem with DockerCommandLineFlags
ecs global name 'DockerCommandLineFlags' is not defined
• To fix the problem you need to enter DockerCommandLineFlags=[] into the script

[root@ecsnode1 ~]# vi /ECS/ECS-CommunityEdition/ecs-multi-node/step1_ecs_multinode_install.py
#!/usr/bin/env python
# An installation program for ECS SW 2.1 Multiple Data nodes
import argparse
import getopt
import subprocess
from subprocess import PIPE, Popen
import logging
import logging.config
import sys
import socket
import os
import time
import settings
import re

# Logging Initialization
logging.config.dictConfig(settings.ECS_SINGLENODE_LOGGING)
logger = logging.getLogger("root")

DockerCommandLineFlags=[]

3.3.2. Problem with argparse

Traceback (most recent call last):
  File "step1_ecs_multinode_install.py", line 3, in <module>
    import argparse
ImportError: No module named argparse
  • To fix - install argparse as described in the previous section

3.3.3. Problem DNS

Traceback (most recent call last):
  File "step1_ecs_multinode_install.py", line 143, in network_file_func
    ip_address = subprocess.check_output(['hostname', '-i']).rstrip(' \n')
  File "/usr/lib64/python2.7/subprocess.py", line 575, in check_output
    raise CalledProcessError(retcode, cmd, output=output)
CalledProcessError: Command ['hostname', '-i'] returned non-zero exit status 1
  • It failed cause I forgot to configure DNS

  • I configured DNS records

[root@ecsnode3 ~]# hostname -i
10.76.246.145
3.3.4. Problem with SSL certificate

...  
e86f1e73f912: Pulling fs layer  
dba2f1894f05: Layer already being pulled by another client. Waiting.  
  • To fix - install EMC SSL certificate as described in the previous section

3.3.5. Problem with authentication server

Executing getAuthToken: curl -i -k https://10.76.246.143:4443/login -u root:ChangeMe  
% Total  % Received % Xferd Average Speed Time Time Time Current  
  Dload Upload Total Spent Left Speed  
  0 0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 0curl: (7) Failed connect to 10.76.246.143:4443; Connection refused
[01/Apr/2016 10:40:31] INFO [root:702] Step 1 Completed. Navigate to the administrator website that is available from any of the ECS data nodes. The ECS administrative portal can be accessed from port 443. For example: https://ecs-node-external-ip-address. The website may take a few minutes to become available.  
  • To fix the problem:  
    o Delete /data directory

[root@ecsnode1 ~]# rm -rf /data  
  o Delete disk partition (unmounts it first if needed)

[root@ecsnode3 ~]# fdisk /dev/sdb  
Welcome to fdisk (util-linux 2.23.2).  
Changes will remain in memory only, until you decide to write them.  
Be careful before using the write command.

Command (m for help): d  
Selected partition 1
Partition 1 is deleted

Command (m for help): w  
The partition table has been altered!  
  o Restart installation script

[root@ecsnode1 ecs-multi-node]# python step1_ecs_multinode_install.py --ips 10.76.246.143  
10.76.246.144 10.76.246.145 10.76.246.146 --hostnames ecsnode1 ecsnode2 ecsnode3 ecsnode4 --ethadapter ens160 --disks sdb
4. Configure ECS

4.1. Configure by script


- Basic configuration can be done by script
- Be patient, it takes about 15-20 minutes

[root@ecsnode3 ~]# python step2_object_provisioning.py --ECSNodes=10.76.246.143,10.76.246.144,10.76.246.145,10.76.246.146 --Namespace=ns1 --ObjectVArray=VA1 --ObjectVPool=VP1 --UserName=user1 --DataStoreName=DS1 --VDCName=VDC1 --MethodName=

**Note:** I prefer to make basic configuration manually and follow the procedure described in the next sections

4.2. Login

- Open in browser https://<node1 IP>

EMC Elastic Cloud Storage

- Check Dashboard
4.3. License

- Apply license

[root@ecsnode1 ecs-multi-node]# /ECS/ECS-CommunityEdition/ecs-multi-node/ license.xml
4.4. Storage Group, VDC and Replication Group

- Configure Storage Pools
• Configure VDC
• Add Replication Group
4.5. Add management user

- Create admin user
4.6. Create namespace

- Configure namespace and add admin user
4.7. **Add object user**

- Create user

- Generate S3 passphrase and Swift password

  - **admin** group has to be specified for Swift user to be able list directories
4.8. Create bucket

- Create bucket
- Specify obj user created on the previous step
5. Check data access

5.1. S3 protocol

- Configure account in S3 browser
- Disable SSL connection
• Now we get access to the bucket we created

• We can store new objects
5.2. Atmos protocol

- Modify and execute subtenant creation script

```
[root@ecsnode1 ECS]# vi create-subtenant.sh
echo "++++++++++++++++++++++++++++++++++++++++++++++++++++++CREATE
SUBTENANT+++++++++++++++++++++++++++++++++++++++++++++++++++++
"
user1="bucket1user"
key1="3hI1vxPKeUuMI5OFEJqxJDGu7wJls1oMR+Hrj6/k"
endpoint="http://10.76.246.143:9022"

method="PUT"
atmos_path="/rest/subtenant"
contentType="text/plain"
date=`date -u +"%a, %d %b %Y %H:%M:%S GMT"`

signstr="$method\n$contentType\n$atmos_path\nx-emc-date:$date\nx-emc-uid:$user1"
sig=$(python -c "import base64, hmac, sha; print base64.b64encode(hmac.new(base64.b64decode( "$key1" ), "$signstr", sha).digest())")
subtid=$(curl -v -i -X $method -H "Content-Type:$contentType" -H "x-emc-date:$date" -H "x-emc-uid:$user1" -H "x-emc-signature:$sig" ${endpoint}${atmos_path} -v | grep 'subtenantID' | cut -d " " -f2 | tr -d ' ')

echo "Sub tenant id is..."
echo $subtid
uid1="$subtid"/$user1"
echo "UID string is:"
echo $uid1
```
[root@ecsnode1 ECS]# chmod +x create-subtenant.sh
[root@ecsnode1 ECS]# ./create-subtenant.sh

+++++++++++++++++++++++++++++++++++++++++++++++CREATE SUBTENANT+++++++++++++++++++++++++++++++++++ *
About to connect() to 10.76.246.143 port 9022 (#0)
* Trying 10.76.246.143...
% Total  % Received % Xferd Average Speed Time Time Current
 Dload Upload Total Spent Left Speed
0 0 0 0 0 0 0 0 --:-- --:-- --:-- --:-- 0* Connected to 10.76.246.143 (10.76.246.143) port 9022 (#0)
> PUT /rest/subtenant HTTP/1.1
> User-Agent: curl/7.29.0
> Host: 10.76.246.143:9022
> Accept: */*
> Content-Type:text/plain
> x-emc-date:Fri, 01 Apr 2016 14:14:17 GMT
> x-emc-uid:bucket1user
> x-emc-signature:pBW2mg8rstqap9SP8e3aIDix8uM=
>
> 0 0 0 0 0 0 0 0 --:-- --:-- 0:00:01 --:--:--:--:--:--:-- 0< HTTP/1.1 204 No Content
< subtenantID: 80f4b7fc83944eca8a5822f0348a1d7b
< x-emc-request-id: 0a4cf68f:153d16724f2:12b:3
< Server: Jetty(7.6.4.v20120524)
<
> 0 0 0 0 0 0 0 0 --:-- --:-- 0:00:01 --:--:-- 0
* Connection #0 to host 10.76.246.143 left intact
Sub tenant id is...
80f4b7fc83944eca8a5822f0348a1d7b
UID string is:
80f4b7fc83944eca8a5822f0348a1d7b/bucket1user

- Check if the bucket is created
- For some reason the owner of the bucket is root
- Modify subtenant’s owner

- Add account into AtmosChrome plugin
• Upload object in Namespace view

• Upload object in Object view

• Add account in AtmosFox
The uploaded object is visible in Namespace view

Nothing visible in Object view
• Type Object ID in the Find field

![Find Object by Object Id](image)

- Now object is visible in the Object view as well

![Manage Accounts](image)

5.3. **Swift protocol**

• Create Cyberduck profile
• Login using the profile

• Create new Folder
• Upload file

![Upload progress bar showing 42.5 MB (42,467,328 bytes) of 45.7 MB (92.4%, 55.6 KB/s, 56 seconds remaining) uploading Cyberduck-Installer-4.8.4.19355.exe]

• File is successfully uploaded

![File list showing Cyberduck-Installer-4.8.4.19355.exe with size 45.7 MB and modified date Today 09:29]
6. Appendix

6.1. Configure network in CentOS

- In CentOS Console
- Don’t forget to specify netmask in address (/24)

[root@localhost ~] # nmtui-edit

- Check the network config

[root@localhost ~] # cat /etc/sysconfig/network-scripts/ifcfg-ens160
TYPE=Ethernet
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=ens160
UUID=d3138700-7057-425d-b0cb-57628a60823e
DEVICE=ens160
ONBOOT=yes
IPADDR=10.76.246.143
PREFIX=32
GATEWAY=10.76.246.1
DNS1=10.76.246.101
DOMAIN=vipr.local
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_PRIVACY=no
### Network Configuration

- **Restart network**

  ```bash
  [root@localhost ~]# systemctl restart network
  ```

- **Configure hostname**

  ```bash
  [root@localhost ~]# hostnamectl set-hostname ecsnode1.vipr.local
  ```

  ```bash
  [root@localhost ~]# hostname
  ecsnode1.vipr.local
  ```

- There is no `ifconfig` in minimal CentOS 7.1 configuration

  ```bash
  [root@localhost ~]# ifconfig
  ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 10.76.246.143  netmask 255.255.255.0  broadcast 10.76.246.255
      inet6 fe80::250:56ff:feb8:d453  prefixlen 64  scopeid 0x20<link>
      ether 00:50:56:b8:d4:53  txqueuelen 1000  (Ethernet)
      RX packets 13173  bytes 14751981 (14.0 MiB)
      RX errors 0  dropped 1972  overruns 0  frame 0
      TX packets 4420  bytes 439972 (429.6 KiB)
      TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
  ```

### 6.2. Create VMware template in CentOS

- Install open vm tools

**Notes:**

KB article which describes Linux customization package installation is here:

• Install VMware tools

[root@ecsnode1 keys]# yum install open-vm-tools

... Installed:
  open-vm-tools.x86_64 0:9.10.2-4.el7
Complete!

[root@ecsnode1 keys]# vmware-toolbox-cmd -v
9.10.2.48224 (build-2822639)

Notes
Guest customization plug-in is available from VMware called the deployPkg Tools Plug-in
open-vm-tools 9.10.x includes the functionality provided by open-vm-tools-deploypkg and there is no need to install this additional package.

• Install additional packages

[root@ecsnode1 ~]# yum install net-tools
[root@ecsnode1 ~]# yum install perl gcc make kernel-headers kernel-devel -y

• Clone VM to the template

• Select DataStore and Thin Provisioning option
6.3. Add new disk to CentOS

[root@ecsnode1 ECS]# echo "- - -" > /sys/class/scsi_host/host0/scan
[root@ecsnode1 ECS]# echo "- - -" > /sys/class/scsi_host/host1/scan
[root@ecsnode1 ECS]# echo "- - -" > /sys/class/scsi_host/host2/scan
[root@ecsnode1 ECS]# fdisk -l 2>/dev/null | egrep '^Disk' | egrep -v 'dm-' | wc -l 10

[root@ecsnode1 ECS]# fdisk -l

Disk /dev/sda: 2147 MB, 2147483648 bytes
255 heads, 63 sectors/track, 261 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0003ea29

Device Boot Start   End     Blocks  Id  System
/dev/sda1 *   1    64   512000  83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2   64   262  1584128  8e  Linux LVM
Partition 2 does not end on cylinder boundary.

Disk /dev/sdb: 214.7 GB, 214748364800 bytes
255 heads, 63 sectors/track, 26108 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/mapper/VolGroup-lv_root: 1405 MB, 1405091840 bytes
255 heads, 63 sectors/track, 170 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/mapper/VolGroup-lv_swap: 213 MB, 213909504 bytes
255 heads, 63 sectors/track, 26 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Disk /dev/sdc: 17.2 GB, 17179869184 bytes
255 heads, 63 sectors/track, 2088 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

- Alternative method to rescan scsi bus

[root@ecsnode1 ECS]# yum install sg3_utils
[root@ecsnode1 ECS]# ./rescan-scsi-bus.sh

6.4. Extend a disk in CentOS


- Create new partition

[root@ecsnode1 ECS]# fdisk /dev/sdc
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0xd20a5af4.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
    switch off the mode (command 'c') and change display units to
    sectors (command 'u').

Command (m for help): p
Disk /dev/sdc: 17.2 GB, 17179869184 bytes
255 heads, 63 sectors/track, 2088 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xd20a5af4

Command (m for help): n
Command action
  e  extended
  p  primary partition (1-4)

Command (m for help): p
Partition number (1-4): 1
First cylinder (1-2088, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-2088, default 2088):
Using default value 2088

Command (m for help): t
Selected partition 1
Hex code (type L to list codes): 8e
Changed system type of partition 1 to 8e (Linux LVM)

Command (m for help): p

Device Boot Start End Blocks Id System
/dev/sdc1 1 2088 16771828+ 8e Linux LVM

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.

- Convert the new partition to a physical volume

[root@ecsnode1 ECS]# pvcreate /dev/sdc1
Physical volume "/dev/sdc1" successfully created

[root@ecsnode1 ECS]# pvdisplay
"/dev/sdc1" is a new physical volume of "15.99 GiB"
--- NEW Physical volume ---
PV Name    /dev/sdc1
VG Name
PV Size    15.99 GiB
### Allocatable
NO

### PE Size
0

### Total PE
0

### Free PE
0

### Allocated PE
0

### PV UUID
c4Nizb-1y31-ToVB-lDru-Chtx-ZJwJ-NaYj7N

- Find out how your Volume Group is called

```
[root@ecsnode1 ECS]# vgdisplay
--- Volume group ---
VG Name          VolGroup
System ID        lvm2
Format           lvm2
Metadata Areas   1
Metadata Sequence No 3
VG Status        resizable
MAX LV           0
Cur LV           2
Open LV          2
Max PV           0
Cur PV           1
Act PV           1
VG Size          1.51 GiB
PE Size          4.00 MiB
Total PE         386
Alloc PE / Size  386 / 1.51 GiB
Free PE / Size   0 / 0
VG UUID          Qu7zi0-sSpz-PoLS-OBdG-n4q7-XIXc-lihtGm
```

- Extend that Volume Group by adding the newly created physical volume to it.

```
[root@ecsnode1 ECS]# vgextend VolGroup /dev/sdc1
Volume group "VolGroup" successfully extended
```

```
[root@ecsnode1 ECS]# vgdisplay
--- Volume group ---
VG Name          VolGroup
System ID        lvm2
Format           lvm2
Metadata Areas   2
Metadata Sequence No 4
VG Status        resizable
MAX LV           0
Cur LV           2
Open LV          2
Max PV           0
Cur PV           2
Act PV           2
VG Size          17.50 GiB
PE Size          4.00 MiB
```
Total PE 4480
Alloc PE / Size 386 / 1.51 GiB
Free PE / Size 4094 / 15.99 GiB
VG UUID Qu7zi0-sSpz-PoLS-OBdG-n4q7-XIXc-lihtGm

- Check the newly added physical volume and the usable space.

[root@ecsnode1 ECS]# pvscan
/dev/sdb: read failed after 0 of 4096 at 214748299264: Input/output error
/dev/sdb: read failed after 0 of 4096 at 214748356608: Input/output error
PV /dev/sda2 VG VolGroup lvm2 [1.51 GiB / 0 free]
PV /dev/sdc1 VG VolGroup lvm2 [15.99 GiB / 15.99 GiB free]
Total: 2 [17.50 GiB] / in use: 2 [17.50 GiB] / in no VG: 0 [ ]

- Extend Logical Volume.

[root@ecsnode1 ECS]# lvextend -L+15.99G /dev/VolGroup/lv_root
Extending logical volume lv_root to 17.30 GiB
Logical volume lv_root successfully resized

[root@ecsnode1 ECS]# lvdisplay
--- Logical volume ---
LV Path /dev/VolGroup/lv_root
LV Name lv_root
VG Name VolGroup
LV UUID PEPZl0-qAVh-S2j6-dxOx-2sWk-QHfc-f2JKK4
LV Write Access read/write
LV Creation host, time localhost.localdomain, 2014-12-25 21:09:29 +0400
LV Status available
# open 1
LV Size 17.30 GiB
Current LE 4429
Segments 2
Allocation inherit
Read ahead sectors auto
- currently set to 256
Block device 253:0

--- Logical volume ---
LV Path /dev/VolGroup/lv_swap
LV Name lv_swap
VG Name VolGroup
LV UUID 1BRC2j-Vr84-2zT7-vmSB-u2YC-mZmG-zmxnpD
LV Write Access read/write
LV Creation host, time localhost.localdomain, 2014-12-25 21:09:30 +0400
LV Status available
# open 1
LV Size 204.00 MiB
Current LE 51
Segments 1
Allocation inherit
Read ahead sectors auto
- currently set to 256
  Block device 253:1

- Resize the file system to the volume group, so we can use the space.

[root@ecsnode1 ECS]# df -h
Filesystem Size Used Avail Use% Mounted on
/dev/mapper/VolGroup-lv_root 1.3G 1.3G 0 100% /
tmpfs 7.8G 0 7.8G 0% /dev/shm
/dev/sda1 485M 33M 427M 8% /boot

[root@ecsnode1 ECS]# resize2fs /dev/VolGroup/lv_root
resize2fs 1.41.12 (17-May-2010)
Filesystem at /dev/VolGroup/lv_root is mounted on /; on-line resizing required
old desc_blocks = 1, new_desc_blocks = 2
Performing an on-line resize of /dev/VolGroup/lv_root to 4535296 (4k) blocks.
The filesystem on /dev/VolGroup/lv_root is now 4535296 blocks long.

[root@ecsnode1 ECS]# df -hT
Filesystem Type Size Used Avail Use% Mounted on
/dev/mapper/VolGroup-lv_root ext4 18G 1.3G 15G 8% /
tmpfs tmpfs 7.8G 0 7.8G 0% /dev/shm
/dev/sda1 ext4 485M 33M 427M 8% /boot