ECS-CommunityEdition Documentation

Release 3.0

Dell-EMC

Aug 08, 2017

Contents

CHAPTER 1

ECS Community Edition Installation

Standard Installation

ECS Community Edition now features a brand new installer. This installer aims to greatly improve user experience through automation. This document will guide the user through the new installation process.

Prerequisites

Listed below are all necessary components for a successful ECS Community Edition installation. If they are not met the installation will likely fail.

Hardware Requirements

The installation process is designed to be performed from either a dedicated installation node. However, it is possible, if you so choose, for one of the ECS data nodes to double as the install node. The install node will bootstrap the ECS data nodes and configure the ECS instance. When the process is complete, the install node may be safely destroyed. Both single node and multi-node deployments require only a single install node.

The technical requirements for the installation node are minimal, but reducing available CPU, memory, and IO throughput will adversely affect the speed of the installation process:

- 1 CPU Core
- 2 GB Memory
- 10 GB HDD
- CentOS 7 Minimal installation (ISO- and network-based minimal installs are equally supported)

The minimum technical requirements for each ECS data node are:

- 4 CPU Cores
- 16 GB Memory

- 16 GB Minimum system block storage device
- 104 GB Minimum additional block storage device in a raw, unpartitioned state.
- CentOS 7 Minimal installation (ISO- and network-based minimal installs are equally supported)

The recommended technical requirements for each ECS data node are:

- 8 CPU Cores
- 64GB RAM
- 16GB root block storage
- 1TB additional block storage
- CentOS 7.3 Minimal installation

For multi-node installations each data node must fulfill these minimum qualifications. The installer will do a pre-flight check to ensure that the minimum qualifications are met. If they are not, the installation will not continue.

Environmental Requirements

The following environmental requirements must also be met to ensure a successful installation:

- **Network:** All nodes, including install node and ECS data node(s), must exist on the same IPv4 subnet. IPv6 networking *may* work, but is neither tested nor supported for ECS Community Edition at this time.
- **Remote Access:** Installation is coordinated via Ansible and SSH. However, public key authentication during the initial authentication and access configuration is not yet supported. Therefore, password authentication must be enabled on all nodes, including install node and ECS data node(s). *This is a known issue and will be addressed in a future release*
- **OS:** CentOS 7 Minimal installation (ISO- and network-based minimal installs are equally supported)

All-in-One Single-Node Deployments

A single node *can* successfully run the installation procedure on itself. To do this simply input the node's own IP address as the installation node as well as the data node in the deploy.yml file.

1. Getting Started

Please use a non-root administrative user account with sudo privileges on the Install Node when performing the deployment. If deploying from the provided OVA, this account is username `admin` with password `ChangeMe`.

Before data store nodes can be created, the install node must be prepared. If downloading the repository from github run the following commands to get started:

0. `sudo yum install -y git`
1. `git clone https://github.com/EMCECS/ECS-CommunityEdition`.

If the repository is being added to the machine via usb drive, scp, or some other file-based means, please copy the archive into `$HOME/` and run:

- for .zip archive `unzip ECS-CommunityEdition.zip`
- for .tar.gz archive `tar -xzvf ECS-CommunityEdition.tar.gz`

Important Note

This documentation refers only to the `ECS-CommunityEdition` directory, but the directory created when unarchiving the release archive may have a different name than `ECS-CommunityEdition`. If this is so, please rename the directory created to `ECS-CommunityEdition` with the `mv` command. This will help the documentation make sense as you proceed with the deployment.

2. Creating The Deployment Map (`deploy.yml`)

Important Note

When installing using the OVA method, please run `videploy` at this time and skip to Step 2.2.

Installation requires the creation of a deployment map. This map is represented in a YAML configuration file called deploy.yml. This file *should* be written before the next step for the smoothest experience.

Create this file in the `ECS-CommunityEdition` directory that was created when the repository was cloned. A template guide for writing this file can be found here.

Below are steps for creating a basic deploy.yml. **Please note that all fields mentioned below are required for a successful installation.**

0. From the ECS-CommunityEdition directory, run the commmand: `cp docs/design/reference.deploy.yml deploy.yml`
1. Edit the file with your favorite editor on another machine, or use `vi deploy.yml` on the Install Node. Read the comments in the file and review the examples in the `examples/` directory.
2. Top-level deployment facts (`facts:`)
   (a) Enter the IP address of the Install Node into the `install_node:` field.
   (b) Enter into the `management_clients:` field the CIDR address/mask of each machine or subnet that will be whitelisted in node's firewalls and allowed to communicate with ECS management API.
   - `10.1.100.50/32` is *exactly* the IP address.
   - `192.168.2.0/24` is the entire /24 subnet.
   - `0.0.0.0/0` represents the entire Internet.

0. SSH login details (`ssh_defaults:`)
   (a) If the SSH server is bound to a non-standard port, enter that port number in the `ssh_port:` field, or leave it set at the default (22).
   (b) Enter the username of a user permitted to run commands as UID 0/GID 0 ("root") via the `sudo` command into the `ssh_username:` field. This must be the same across all nodes.
   (c) Enter the password for the above user in the `ssh_password:` field. This will only be used during the initial public key authentication setup and can be changed after. This must be the same across all nodes.

0. Node configuration (`node_defaults:`)
   (a) Enter the DNS domain for the ECS installation. This can simply be set to `localdomain` if you will not be using DNS with this ECS deployment.
   (b) Enter each DNS server address, one per line, into `dns_servers:`. This can be what's present in `/etc/resolv.conf`, or it can be a different DNS server entirely. This DNS server will be set to the primary DNS server for each ECS node.
   (c) Enter each NTP server address, one per line, into `ntp_servers:`.

0. Storage Pool configuration (`storage_pools:`)
   (a) Enter the storage pool `name:`.

(b) Enter each member data node address, one per line, in `members:`.

(c) Under `options:`, enter each block device reserved for ECS, one per line, in `ecs_block_devices:`.

0. Virtual Data Center configuration (`virtual_data_centers:`)

(a) Enter each VDC `name:`.

(b) Enter each member Storage Pool name, one per line, in `members:`

0. Optional directives, such as those for Replication Groups and users, may also be configured at this time.

1. When you have completed the `deploy.yml` to your liking, save the file and exit the `vi` editor.

2. Move on to Bootstrapping

These steps quickly set up a basic deploy.yml file

Please read the reference deploy.yml found here. It is designed to be self documenting and required fields are filled with either example or default values. The above values are only bare minimum values and may not yield optimal results for your environment.

3. Bootstrapping the Install Node (`bootstrap.sh`)

Important Note

When installing using the OVA method, please skip to Step 4.

The bootstrap script configures the installation node for ECS deployment and downloads the required Docker images and software packages that all other nodes in the deployment will need for successful installation.

Once the deploy.yml file has been created, the installation node must be bootstrapped. To do this `cd` into the ECS-CommunityEdition directory and run `./bootstrap.sh -c deploy.yml`. Be sure to add the `-g` flag if building the ECS deployment in a virtual environment and the `-y` flag if you're okay accepting all defaults.

The bootstrap script accepts many flags. If your environment uses proxies, including MitM SSL proxies, custom nameservers, or a local Docker registry or CentOS mirror, you may want to indicate that on the `bootstrap.sh` command line.

```
[Usage]
 -h              This help text

[General Options]
 -y / -n         Assume YES or NO to any questions (may be dangerous).

 -v / -q         Be verbose (also show all logs) / Be quiet (only show necessary␣
↪output)

 -c <deploy.yml> If you have a deploy.yml ready to go, use this.

 -o <ns1[,ns2,]> Override DHCP-configured nameserver(s); use these instead. No spaces!

 -g              Install virtual machine guest agents and utilities for QEMU and␣
↪VMWare.
                 VirtualBox is not supported at this time.

 -m <mirror>     Use the provided package <mirror> when fetching packages for the
                 base OS (but not 3rd-party sources, such as EPEL or Debian-style␣
↪PPAs).
                 The mirror is specified as '<host>:<port>'. This option overrides any
                 mirror lists the base OS would normally use AND supersedes any␣
↪proxies
```

```
                (assuming the mirror is local), so be warned that when using this
                option it's possible for bootstrapping to hang indefinitely if the
                mirror cannot be contacted.

 -b <mirror>    Build the installer image (ecs-install) locally instead of fetching
                the current release build from DockerHub (not recommended). Use the
                Alpine Linux mirror <mirror> when building the image.

[Docker Options]
 -r <registry>  Use the Docker registry at <registry> instead of DockerHub.
                The connect string is specified as '<host>:<port>[/<username>]'
                You may be prompted for your credentials if authentication is␣
↪required.
                You may need to use -d (below) to add the registry's cert to Docker.

 -l             After Docker is installed, login to the Docker registry to access␣
↪images
                which require access authentication. Login to Dockerhub by default␣
↪unless
                -r is used.

 -d <x509.crt>  NOTE: This does nothing unless -r is also given.
                If an alternate Docker registry was specified with -r and uses a cert
                that cannot be resolved from the anchors in the local system's trust
                store, then use -d to specify the x509 cert file for your registry.

[Proxies & Middlemen]
 -k <x509.crt>  Install the certificate in <file> into the local trust store. This is
                useful for environments that live behind a corporate HTTPS proxy.

 -p <proxy>     Use the <proxy> specified as '[user:pass@]<host>:<port>'
                items in [] are optional. It is assumed this proxy handles all␣
↪protocols.

 -t <connect>   Attempt to CONNECT through the proxy using the <connect> string␣
↪specified
                as '<host>:<port>'. By default 'google.com:80' is used. Unless you␣
↪block
                access to Google (or vice versa), there's no need to change the␣
↪default.

[Examples]
 Install VM guest agents and install the corporate firewall cert in certs/mitm.pem.
    $ ./bootstrap.sh -g -k certs/mitm.pem

 Quietly use nlanr.peer.local on port 80 and test the connection using EMC's␣
↪webserver.
    $ ./bootstrap.sh -q -p nlanr.peer.local:80 -t emc.com:80

 Assume YES to all questions and use the proxy cache at cache.local port 3128 for␣
↪HTTP-
 related traffic. Use the Docker registry at registry.local:5000 instead of DockerHub,
 and install the x509 certificate in certs/reg.pem into Docker's trust store so it can
 access the Docker registry.
    $ ./bootstrap.sh -y -p cache.local:3128 -r registry.local:5000 -d certs/reg.pem
```

The bootstrapping process has completed when the following message appears:

```
> All done bootstrapping your install node.
>
> 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
>
```

After the installation node has successfully bootstrapped you may be prompted to reboot the machine. If so, then the machine must be rebooted before continuing to Step 4.

4. Deploying ECS Nodes (`step1` or `island-step1`)

Once the deploy.yml file has been correctly written and the Install Node rebooted if needed, then the next step is to simply run one of the following commands:

- Internet-connected environments: `step1`
- Island environments: `island-step1`

After the installer initializes, the EMC ECS license agreement will appear on the screen. Press `q` to close the screen and type `yes` to accept the license and continue or `no` to abort the process. The install cannot continue until the license agreement has been accepted.

The first thing the installer will do is create an artifact cache of base operating system packages and the ECS software Docker image. If you are running `step1`, please skip to **Step 5**. If you are running `island-step1`, then the installer will stop after this step. The install node can then be migrated into your island environment where deployment can continue.

Important Note

> If you are deploying to Internet-connected nodes and used `step1` to begin your deployment, please skip to **Step 5**.

- Internet-connected environments: *automatic*
- Island environments: `island-step2`

If you are deploying into an island environment and have migrated the install node into your island, you can begin this process by running `island-step2`. The next tasks the installer will perform are: configuring the ECS nodes, performing a pre-flight check to ensure ECS nodes are viable deployment targets, distributing the artifact cache to ECS nodes, installing necessary packages, and finally deploying the ECS software and init scripts onto ECS nodes.

5. Deploying ECS Topology (`step2` or `island-step3`)

- Internet-connected environments: `step2`
- Island environments: `island-step3`

Once either `step1` or `island-step2` have completed, you may then direct the installer to configure the ECS topology by running either `step2` or `island-step3`. These commands are identical. Once `step2` or `island-step3` have completed, your ECS will be ready for use. If you would prefer to manually configure your ECS topology, you may skip this step entirely.

OVA Installation

ECS Community Edition can optionally be installed with the available single-node (recommended) and multi-node OVAs. To install with this method:

1. Download and deploy the OVA to a VM

2. Adjust the resources to have a minimum of:

```
* 16GB RAM
* 4 CPU cores
* (Optional) Increase vmdk from the minimum 104GB
```

3. Clone VM to number of nodes desired

4. Collect network information

Power on VM's and collect their DHCP assigned IP addresses from the vCenter client or from the VMs themselves

You may also assign static IP addresses by logging into each VM and running `nmtui` to set network the network variables (IP, mask, gateway, DNS, etc).

5. Log into the first VM and run `videploy`

Follow the directions laid out in the standard installation concerning the creation of the deploy.yml file (section 2).

After completing the deploy.yml file, exit out of `videploy`, this will update the deploy.yml file.

6. Run `step1`

7. Run `step2`

Important Note: `step1` and `step2` are not scripts and should not be run as such. `./step1` is not a valid command.

That's it!

Assuming all went well, you now have a functioning ECS Community Edition instance and you may now proceed with your test efforts.

CHAPTER 2

ECS Administrative Web UI

Login to the Web UI

The WebUI uses SSL and a self-signed certificate to help protect your session from casual eves-dropping. Take the IP of your first ECS node, fire up your browser, and point `https://` at it. For this example, the latest Google Chrome browser was used.

You cannot add, change, or remove administrative users in this build. Use the default below.

> Username: **root**Password: **ChangeMe**

Input License

Open *Settings*, then *Licensing* and upload the `license.xml` file located in the ecs-single-node / ecs-multi-node folder. **The UI will not automatically update the license view in this release.** Navigating away from page and returning will prompt it to update. You may need to try a few times before it updates. Once it does, you should see something like this:

Create Storage vPool

Open *Manage*, then *Storage Pools* and create a storage pool. Keep the name simple, and add all nodes to the pool. Click *Save*.

There's a known issue in this build that causes the Storage Pools view to appear frozen for about 1-2 minutes after provisioning begins. **Unlike with the license view case, this view will update on its own.** Once it's updated, you should see something similar to:

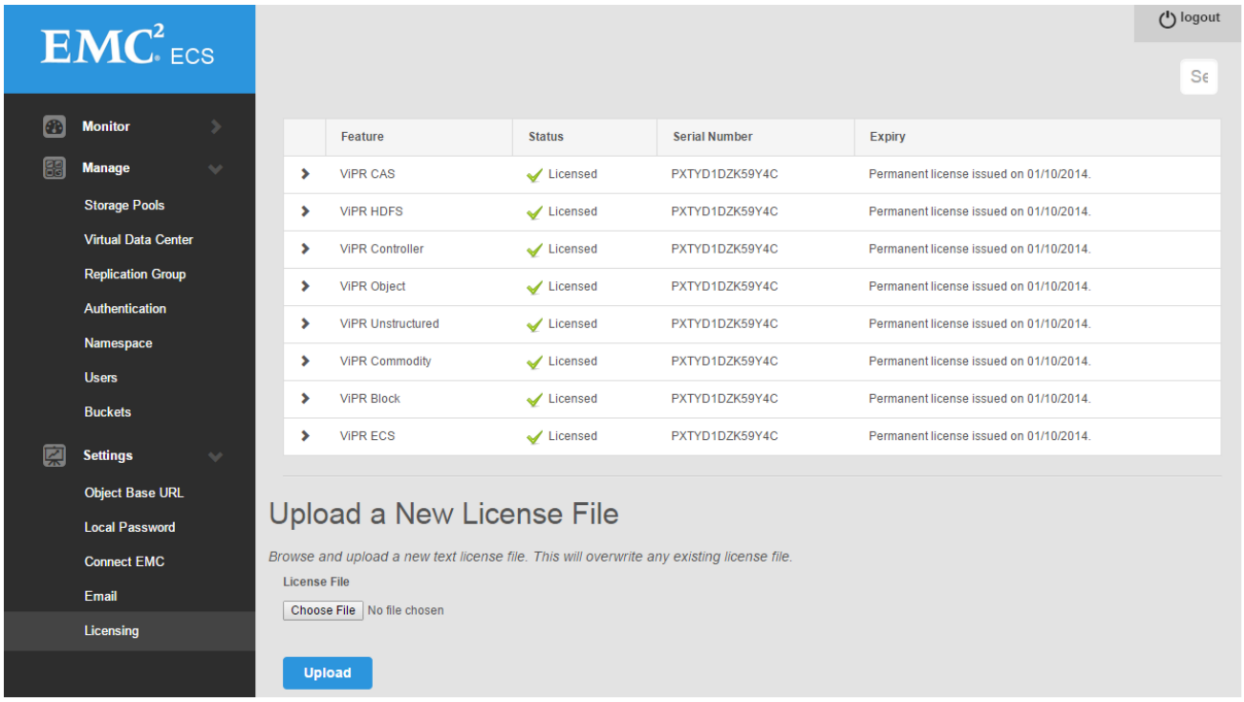

Fig. 2.1: Upload License file

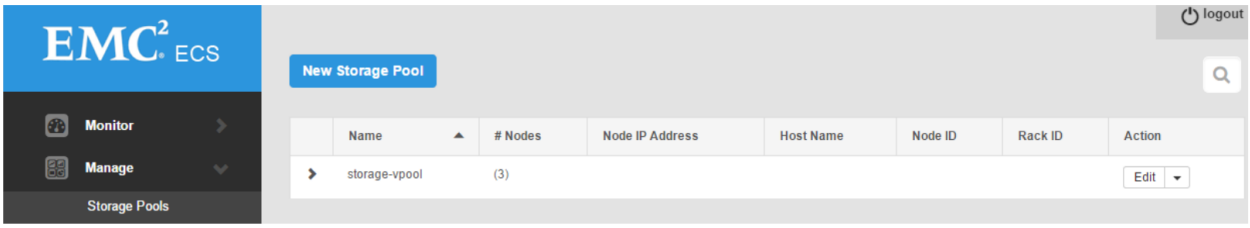

Fig. 2.2: Create Storage VPool

Create Virtual Data Center

Open *Manage*, then *Virtual Data Center* and create a Virtual Data Center using the below screenshot as a guide. **Please wait for up to 20 minutes after creating a Storage vPool before creating a Virtual Data Center.** There are several background tasks that must complete, and for object to fully initialize.

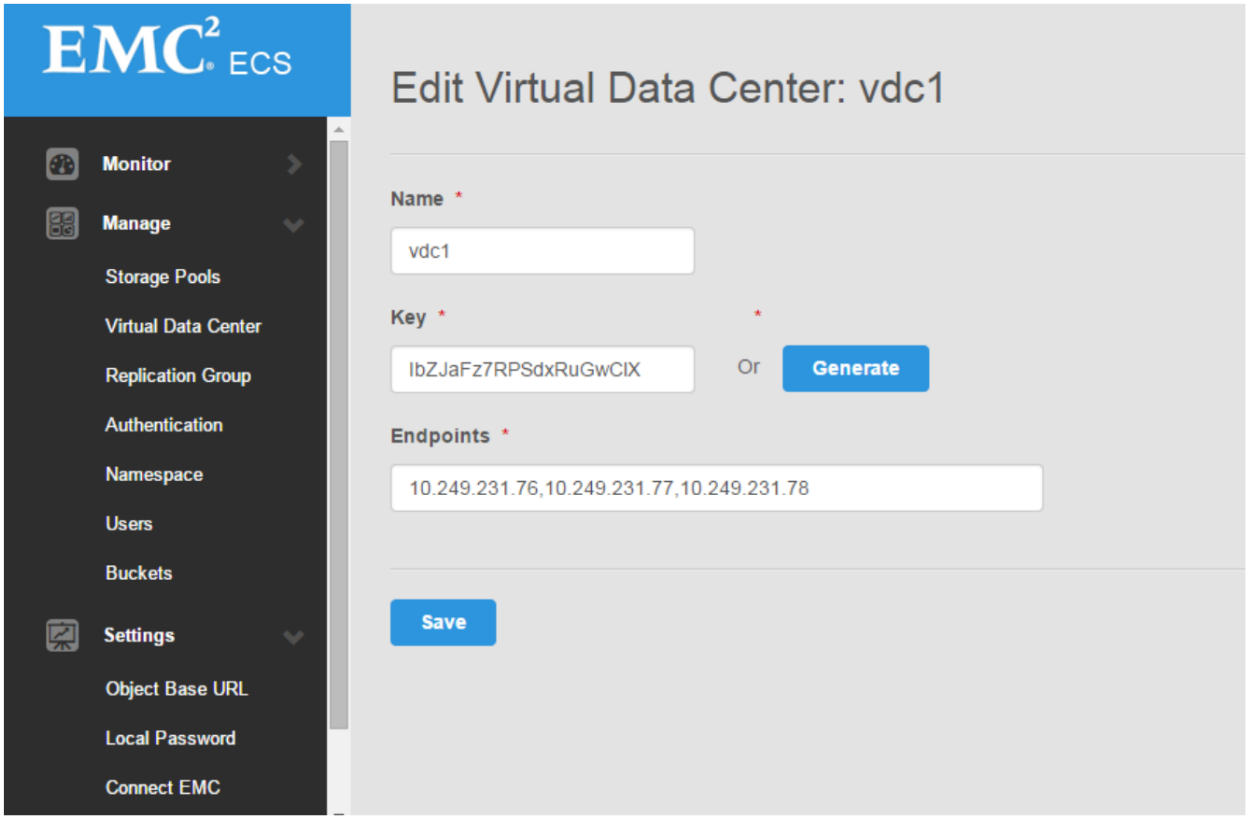

Fig. 2.3: Create Virtual Data Center

Create Replication Group

Open *Manage*, then *Replication Group* and create a Replication Group using the below as an example. Currently only one VDC in a replication group is supported.

Create Namespace

Open *Manage*, then *Namespace*. Set up a Simple Namespace with a name such as "ns". Input a namespace username to use with the namespace, such as "ecs_user". Select the replication group for the namespace, and click *Save* at the very bottom.

Namespace features available in this release

- Simple Namespace
- Retention Policies

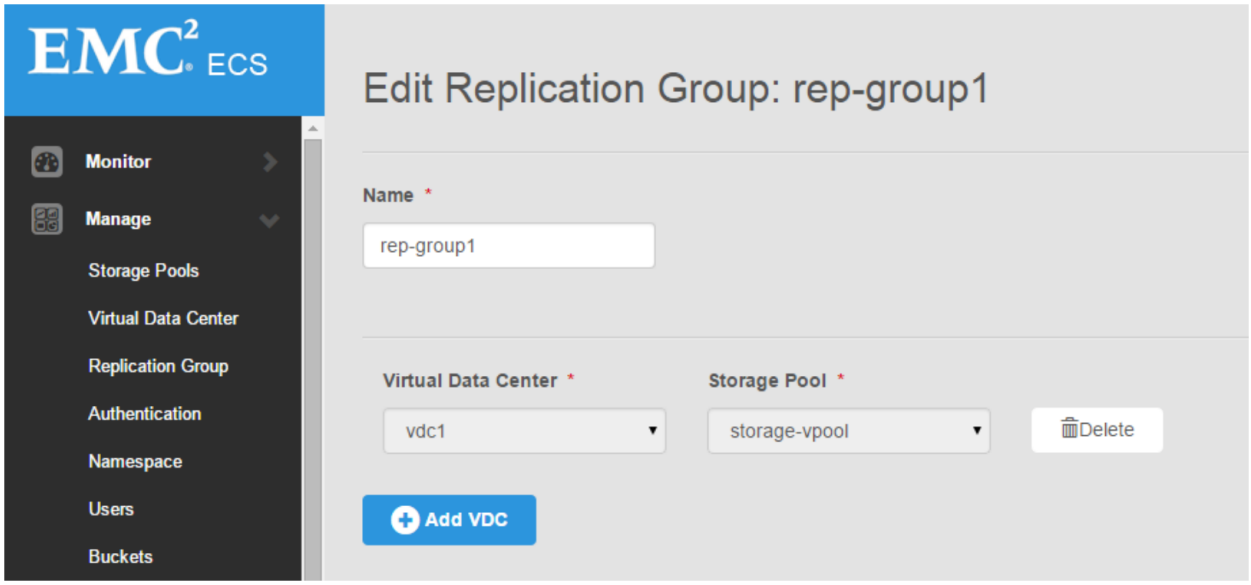

Fig. 2.4: Create Replication Group

- Quotas
- Authentication Domains

Create Object User Account

Open *Manage*, then *Users*, then click on *Object Users* and *New Object User* to set up object store credentials.

Create secrets by filling the fields and clicking the buttons.

- S3 Key: Click *Generate & Add Password* to retrieve the server-generated key.
- Swift Password: Enter your own password and click *Set Password*.

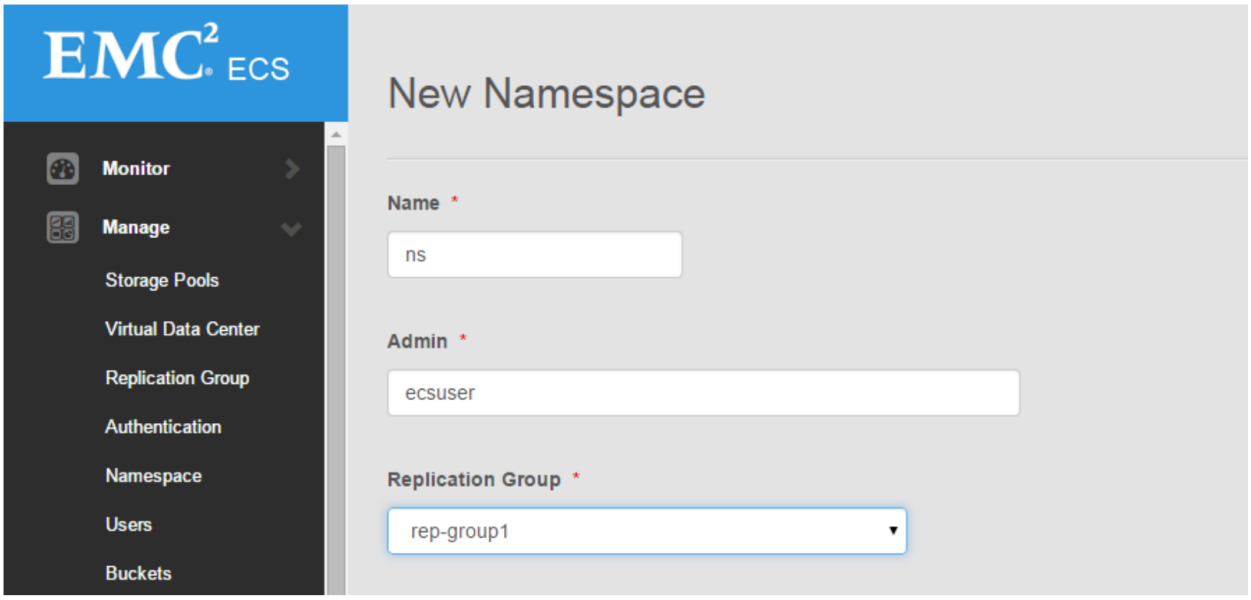

Fig. 2.5: Create Namespace

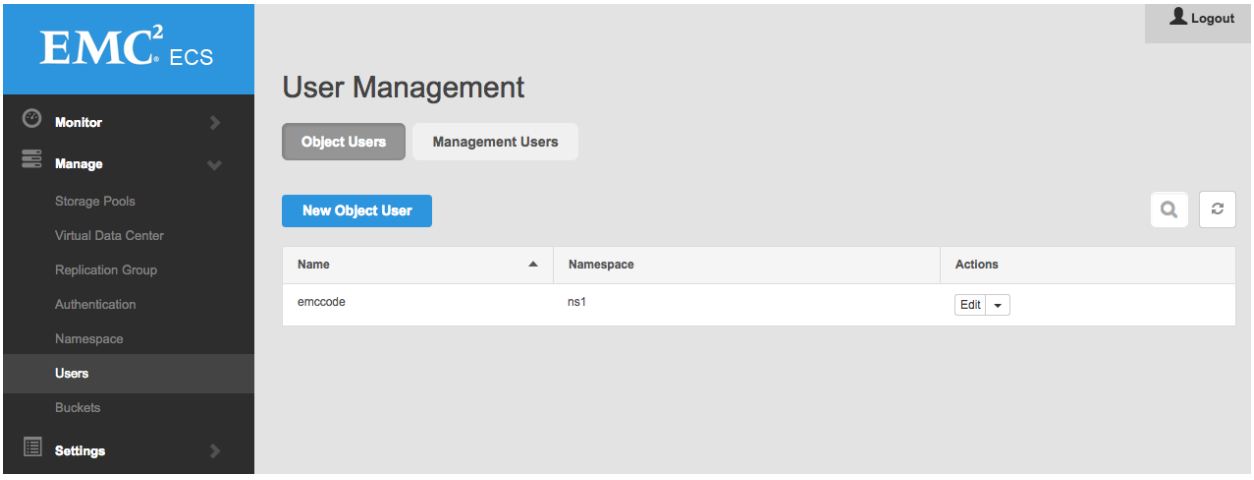

Fig. 2.6: Create Namespace

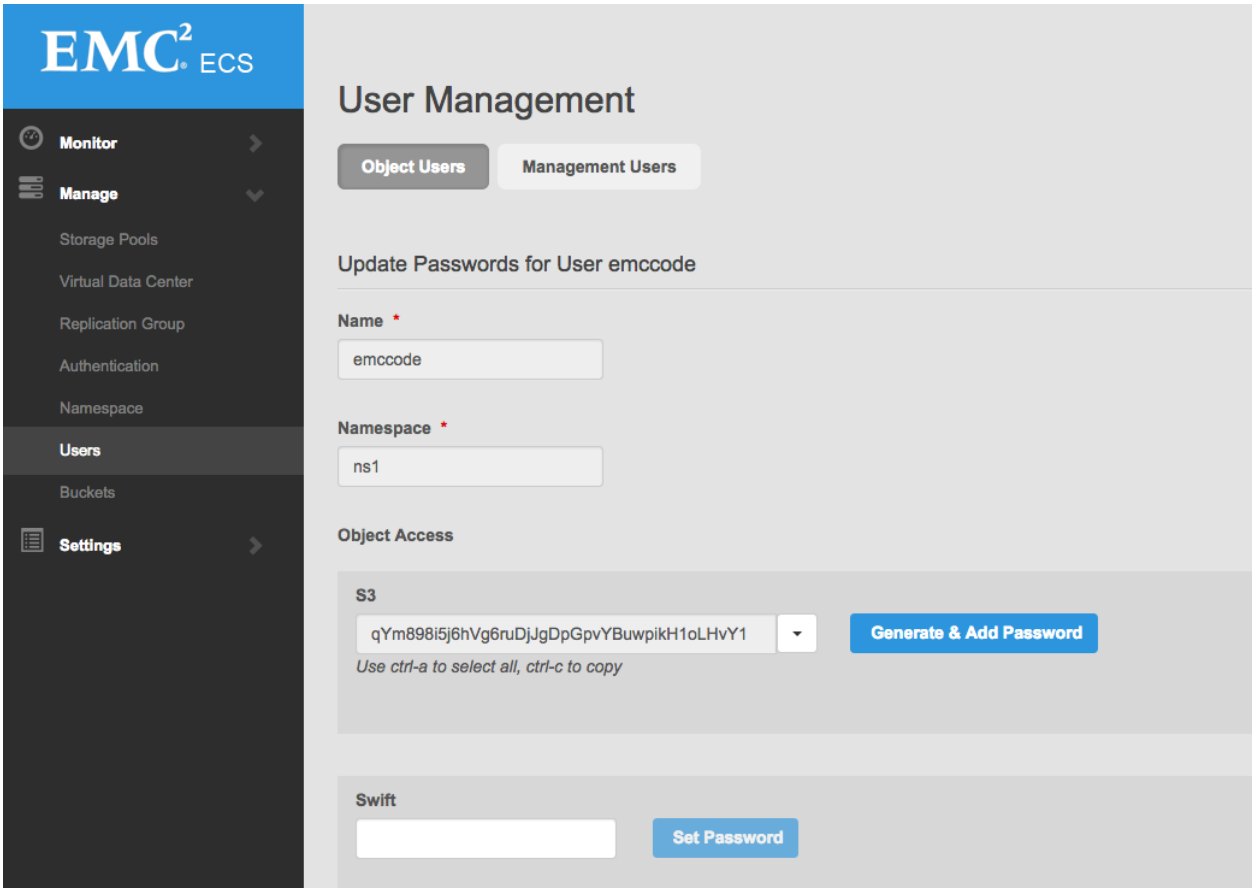

Fig. 2.7: Create User S3 and Swift Keys

CHAPTER 3

Migration

General Cases

Most migration cases can be handled by a great tool we wrote called ecs-sync, found here.

HDFS

An HDFS migration is possible with s3distcp or distcp. Please note that if using s3distcp with the s3a driver, it needs to be the latest version or you may run into issues. If using distcp, ECS's HCFS driver "viprfs" will need to be set up as a secondary FS and the distcp made from `hdfs://...` to `viprfs://...`. Instructions for installing the HCFS driver can be found here.

CHAPTER 4

ECS Software 3.x - Troubleshooting Tips

This is a list of troubleshooting tips and nuggets that will help with issues. If you still have problems, please use the support section.

Installation

If you change deploy.yml after running step1, you must run `update_deploy` before running step1 again. Otherwise you will likely get the following error:

```
{"failed": true, "msg": "An unhandled exception occurred while running the lookup␣
↪plugin 'file'.
 Error was a <class 'ansible.errors.AnsibleFileNotFound'>, original message: the file_
↪name
 '/opt/ssh/id_ed25519.pub' does not exist, or is not readable"}
```

A block device configured in deploy.yml for data nodes is already partitioned.

This error often shows up after a failed installation attempt. In order to clean up the block devices to start over run `ecsremove purge-nodes`.

Provisioning of ECS

It takes roughly 30 minutes to get the system provisioned for Step2. ECS creates Storage Pools, Replication Groups with the attached disks. If Step2 is successful, you should see something along these lines.

Checking Step 2 Object provisioning progress

If you want to see if system is making progress:

1. Log into one of ECS data nodes.
2. Navigate to the **/var/log/vipr/emcvipr-object/** directory
3. View the **/var/log/vipr/emc-viprobject/ssm.log** (tail -f /var/log/vipr/emcvipr-object/ssm.log)

Note: there are ~2k tables to be initialized for the provisioning to complete. You can check the following command to see if the tables are close to that number and if all tables are ready. Run this from the node.

```
curl -X GET "http://<YourIPAddress>:9101/stats/dt/DTInitStat"
```

ECS Services

Docker Container immediately exits on startup

If your docker instance immediately exits when started, please ensure that the entries in `/etc/hosts` on the host system and `network.json` in the install directory are correct (the latter should reflect the host's public IP and the corresponding network adapter).

ECS web portal will not start

The portal service will listen on ports 443 and 4443; check to make sure no other services (such as virtual hosts or additional instances of ECSCE) are not attempting to utilize these same ports.

For multiple-node installations, the `/etc/hosts` file on the host VM should include entries for each node and their hostname. Additionally, many services including the ECS web portal will not start until all nodes specified to the installation step 1 script have been successfully installed and concurrently running; the installation script should be run on all nodes in a cluster before attempting authentication or use of the GUI.

If attempting to authenticate results in a response of "Connection Refused", review the below section and ensure all necessary ports are open on all ECS nodes in the cluster.

NFS

Necessary NFS Ports

The following ports must be opened for NFS to function properly

Port Number
111
2049

NFS Volume Refuses to Mount

ECS does support the NFS file system. However, troubles can occur when ECS is installed on the full version, or "Everything" version, of CentOS 7. ***Note that the following solution is not necessary on CentOS 7 Minimal.***

The Problem

CentOS 7 Everything starts with NFS/RPC/Portmap components running in the root scope. This is a problem as the ECS-CE Docker container runs its own version of rpcbind. This is the instance of rpcbind that ECS is intended to communicate with. When CentOS is running rpcbind in root scope in addition to the ECS Docker container, a conflict is created and a NFS volume cannot be mounted.

This can be seen by `# rpcinfo -p` returning no NFS services.

The Solution

The conflict can be resolved by simply running `systemctl disable rpcbind`. This command will shut down the rpc service running on the host OS while leaving the Docker instance untouched.

To confirm the CentOS service is gone, run `rpcinfo -p` in the CentOS shell. This should return an error: `rpcinfo: can't contact portmapper: RPC: Remote system error - No such file or directory`

The same command, `rpcinfo-p`, can be run in the Docker container, which should return something similar to:

```
program vers proto   port  service
 100000    4   tcp    111  portmapper
 100000    3   tcp    111  portmapper
 100000    2   tcp    111  portmapper
 100000    4   udp    111  portmapper
 100000    3   udp    111  portmapper
 100000    2   udp    111  portmapper
 100005    3   tcp   2049  mountd
 100005    3   udp   2049  mountd
 100003    3   tcp   2049  nfs
 100024    1   tcp   2049  status
 100021    4   tcp  10000  nlockmgr
 100021    4   udp  10000  nlockmgr
```

NFS should now function correctly.

IBM Tivoli Monitoring

Issue

ECS Community edition will fail to completely initialize the storage pool on machines that have the IBM Tivoli Monitoring agent installed. The storage pool will forever stick in the "Initializing" state and attempts to create a VDC will result in HTTP 400 errors.

Analysis

Doing a `ps -ef` inside the container will show that dataheadsvc and metering are restarting frequently. Looking at `/opt/storageos/logs/metering.log` will show a bind exception on port 10110. This port is already bound by Tivoli's `k10agent` process.

Workaround

1. Uninstall Tivoli Monitoring or
2. Change the port on impacted nodes.

Changing the port on ECS

On *all* nodes, you will need to edit `/opt/storageos/conf/mt-var.xml` to change the bind port from 10110 to 10109. Edit the file and change the line:

```
<property name="serviceUrl" value="service:jmx:rmi://127.0.0.1:10110/jndi/rmi://127.0.
↪0.1:10111/sos" />
```

to:

```
<property name="serviceUrl" value="service:jmx:rmi://127.0.0.1:10109/jndi/rmi://127.0.
↪0.1:10111/sos" />
```

Then restart the metering service:

```
kill `pidof metering`
```

Network Troubleshooting

For those operating behind EMC firewall

To install ECS Community Edition under these conditions, please view the readme file under **/emc-ssl-cert** for further instructions in installing the necessary CA certificate.

Disabling IPv6

ECS Community Edition does not yet support IPv6. The following procedure can be used to disable IPv6 in CentOS 7.

To disable IPv6 on startup:

Add the following to /etc/sysctl.conf

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

To disable IPv6 running:

```
echo 1 > /proc/sys/net/ipv6/conf/all/disable_ipv6
echo 1 > /proc/sys/net/ipv6/conf/default/disable_ipv6
```

or

```
sysctl -w net.ipv6.conf.all.disable_ipv6=1
sysctl -w net.ipv6.conf.default.disable_ipv6=1
```

Get correct interface name

CentOS 7 does not assign network interface names as eth0, eth1, etc, but rather assigns "predictable" names to each interface that generally look like `ens32` or similar. There are many benefits to this that can be read about here.

This can be disabled as documented in the above link, however, these names can otherwise be simply found and used in the ECS-Community installer without issue. To find the names for each device enter the following command: `ip a`. This command will output a list of network devices. Simply find the corresponding device and substitute it for eth0 in the stage1 installation script.

Port Conflicts

It is possible that on multinode installations ECS may run into a port conflict. So far there exists a port conflict with the following:

- ScaleIO - Ports: 9011, 9099

In these instances the user can attempt to:

1. Enter the container
2. Change all instances of the conflicting ports to unused ports in `/opt/storageos/conf`
3. Reboot the nodes after altering the conf file.

List of open ports required on each ECS data node

Ensure the ports in the following table are open for communication. In the case of a multiple-node installation, additionally ensure that each node is trusted to itself and to other nodes in the system by using the following command on each node:

`firewall-cmd --permanent --zone=trusted --add-source=<ECS-node-IP>/32`

followed by `firewall-cmd --reload` for each host.

`fwd_settings.sh` in the main directory will invoke the `firewalld` service and permanently open necessary ports. In the case of a failure in this setup referencing `iptables`, please ensure that your docker network bridge is running and installed using `yum install bridge-utils`.

Port Name-Usage=Port Number
port.ssh=22
port.ecsportal=80
port.rcpbind=111
port.activedir=389
port.ecsportalsvc=443
port.activedirssl=636
port.ssm=1095
port.rm=1096
port.blob=1098
port.provision=1198

Continued on next page

Table 4.1 – continued from previous page

Port Name-Usage=Port Number
port.objhead=1298
port.nfs=2049
port.zookeeper=2181
port.coordinator=2889
port.cassvc=3218
port.ecsmgmtapi=4443
port.rmmvdcr=5120
port.rmm=5123
port.coordinator=7399
port.coordinatorsvc=7400
port.rmmcmd=7578
port.objcontrolUnsecure=9010
port.objcontrolSecure=9011
port.s3MinUnsecure=9020
port.s3MinSecure=9021
port.atmosMinUnsecure=9022
port.atmosMinSecure=9023
port.swiftMinUnsecure=9024
port.swiftMinSecure=9025
port.apiServerMinUnsecure=9028
port.apiServerMinSecure=9029
port.hdfssvc=9040
port.netserver=9069
port.cm=9091
port.geoCmdMinUnsecure=9094
port.geoCmdMinSecure=9095
port.geoDataMinUnsecure=9096
port.geoDataMinSecure=9097
port.geo=9098
port.ss=9099
port.dtquery=9100
port.dtqueryrecv=9101
port.georeplayer=9111
port.stat=9201
port.statWebServer=9202
port.vnest=9203
port.vnesthb=9204
port.vnestMinUnsecure=9205
port.vnestMinSecure=9206
port.hdfs=9208
port.event=9209
port.objcontrolsvc=9212
port.zkutils=9230
port.cas=9250
port.resource=9888
port.tcpIpcServer=9898

CHAPTER 5

Frequently Asked Questions

Can I add storage to ECS-CommunityEdition after initializing an installation?

No. Unfortunately because ECS Community Edition lacks the 'fabric' layer present in full featured ECS, it is not possible to add storage space after a completed installation.

I am locked out of my ECS management console, can I reset the root password?

Currently there is no procedure to reset the root password if you are locked out.

The storage capacity statistics presented on the ECS dashboard seem wrong, what's going on here?

ECS uses a data boxcarting strategy called "chunking". Chunks are pre-allocated when ECS is first initialized. When user data is written into ECS, pre-allocated chunks are filled and ECS pre-allocates however many new chunks ECS "thinks" would be best for the future based on what it knows about the past.

Capacity statistics are calculated based on allocated and pre-allocated chunks at the time statistics are requested, and don't exactly reflect the actual amount of user data stored within ECS. We do this because it is a performance-enhancing heuristic that is a "good enough" representation of capacities without having to churn through the whole system to figure out the actual user data capacity numbers. In short, the numbers you are seeing are not designed to be exact, but are close estimates.

Can I use a data store node as an NTP server for the rest of the nodes?

No, this is not a supported deployment option. An external NTP server is required.

My ECS functions but the storage pools never initialize.

If you can store objects in buckets without issue, then it's likely that your storage pools and data stores are fully initialized. ECS Community Edition is a bit weird in that there are some UI/display issues with storage pools showing "Not Ready" and data stores showing "initializing" even after they have become fully initialized. If you can create VDCs, replication groups, namespaces, and buckets, then your storage pools are certainly initialized as those higher level abstractions require a working storage pool.

ECS Community Edition

See changelog.md file for release notes.

CHAPTER 6

Description

EMC Elastic Cloud Storage (ECS) is a stateful containerized cloud storage. It provides persistence for your applications that can access data through standardized Object protocols like AWS S3 or OpenStack Swift. ECS can be set up on one or more hosts / VMs in a single-site or a multi-site geo replicated configuration. We want the wider community to use ECS and provide feedback. Usage of this software is under the following End User License Agreement.

ECS Community Edition is a free, reduced footprint, version of Dell EMC's Elastic Cloud Storage software. Of course, this means there are some limitations to the use of the software, so the question arises; how is the Community Edition of ECS different from the production version?

License difference

As noted with the included license, ECS Community cannot be used in production environments and is intended to be used for trial and proof of concept purposes only. This software is still owned and protected by Dell EMC.

Feature differences

It it important to note that ECS-Community Edition is **not** the same as ECS software and as such lacks some features that are integral to the actual ECS software.

- ECS Community Edition does NOT support encryption.
- ECS Community Edition does NOT include ECS' system management, or "fabric", layer.

Notice

Because of these differences, ECS Community Edition is absolutely **not** qualified for testing failure scenarios. Failure scenarios can only be adequately mimicked on the full version of ECS Software.

CHAPTER 7

Quick Start Guide

If you have the following:

1. A CentOS 7.3 Minimal instance with:
   (a) 16GB RAM
   (b) 16GB block device for system
   (c) 104GB block device for ECS
2. Internet access
3. No proxies, local mirrors, or special Docker registries

Then you should be able to get up and going with a Single-Node All-in-One install using these commands on your VM:

```
# git clone https://github.com/EMCECS/ECS-CommunityEdition
# cd ECS-CommunityEdition
# cp docs/design/reference.deploy.yml deploy.yml
# echo "Edit this deploy.yml to match your VM's environment"
# vi deploy.yml
# ./bootstrap.sh -y -c deploy.yml
```

And then after the node reboots (you did use a clean minimal install from ISO or netinstall right?):

```
# step1
# step2
```

And if all went well, you now have a working stand-alone ECS, mostly configured, and ready for use.

CHAPTER 8

Hardware Requirements

Hardware or virtual machine with:

- 4 CPU Cores
- 16GB RAM
- 16GB root block storage
- 104GB additional block storage
- CentOS 7.3 Minimal installation

Hardware or virtual machine with:

- 8 CPU Cores
- 64GB RAM
- 16GB root block storage
- 1TB additional block storage
- CentOS 7.3 Minimal installation

CHAPTER 9

Deployment Scenarios

ECS Multi-Node All-in-One Deployment with Install Node (recommended, full-featured)

Deploy a multi-node ECS instance to two or more hardware or virtual machines and enable all ECS features. Three nodes are required for all ECS 3.0 and above features to be activated.

ECS Single-Node All-in-One Deployment (smallest footprint)

Deploy a stand-alone instance of a limited set of ECS kit to a single hardware or virtual machine.

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.

- dellemc-ecsce-3.0.0.2-install-node-2.3.0-vm0.ova

ECS Multi-Node Deployment with Install Node (recommended, most reusable, full-featured)

Using an Install Node for isolated environments, deploy a multi-node ECS instance to two or more hardware or virtual machines and enable all ECS features. Three nodes are required for all ECS 3.0 and above features to be activated.

ECS Single-Node Deployment with Install Node

Using an Install Node for isolated environments, deploy a stand-alone instance of a limited set of ECS kit to a single hardware or virtual machine.