# Pre-deployment Checklist

This section covers the preliminary setup required on the compute and storage nodes before deploying DAOS.

## Enable IOMMU (Optional)

In order to run the DAOS server as a non-root user with NVMe devices, the hardware must support virtualized device access, and it must be enabled in the system BIOS. On Intel® systems, this capability is named Intel® Virtualization Technology for Directed I/O (VT-d). Once enabled in BIOS, IOMMU support must also be enabled in the Linux kernel. Exact details depend on the distribution, but the following example should be illustrative:

# Enable IOMMU on CentOS 7  
# All commands must be run as root/sudo!  
  
$ sudo vi /etc/default/grub # add the following line:  
GRUB\_CMDLINE\_LINUX\_DEFAULT="intel\_iommu=on"  
  
# after saving the file, run the following to reconfigure  
# the bootloader:  
$ sudo grub2-mkconfig --output=/boot/grub2/grub.cfg  
  
# if the command completed with no errors, reboot the system  
# in order to make the changes take effect  
$ sudo reboot

!!! warning VFIO support is a new feature for DAOS 1.2 and has been tested on CentOS 7.7

To force SPDK to use UIO rather than VFIO at daos\_server runtime, set ‘disable\_vfio’ in the [server config file](https://github.com/daos-stack/daos/blob/master/utils/config/daos_server.yml#L109), but note that this will require running daos\_server as root.

## Time Synchronization

The DAOS transaction model relies on timestamps and requires time to be synchronized across all the storage and client nodes. This can be done using NTP or any other equivalent protocol.

## User/Group Synchronization

DAOS ACLs store the actual user and group names (instead of numeric IDs), and therefore the servers do not need access to a synchronized user/group database. The DAOS Agent (running on the client nodes) is responsible for resolving UID/GID to user/group names which are added to a signed credential and sent to the DAOS storage nodes.

## Multi-rail/NIC Setup

Storage nodes can be configured with multiple network interfaces to run multiple engine instances.

### Subnet

Since all DAOS engines need to be able to communicate, the different network interfaces need to be on the same subnet or routing capabilities across the different subnet might be configured.

### Infiniband/RoCE

Some special configuration is required to use librdmacm with multiple interfaces.

Firstly, the accept\_local feature must be enabled on the network interfaces to be used by DAOS. This can be done using the following command ( must be replaced with the interface names):

$ sudo sysctl -w net.ipv4.conf.all.accept\_local=1

Secondly, Linux must be configured to only send ARP replies on the interface targeted in the ARP request. This is configured via the arp\_ignore parameter. This should be set to 2 if all the IPoIB interfaces on the client and storage nodes are in the same logical subnet (e.g. ib0 == 10.0.0.27, ib1 == 10.0.1.27, prefix=16).

$ sysctl -w net.ipv4.conf.all.arp\_ignore=2

If separate logical subnets are used (e.g. prefix = 24), then the value must be set to 1.

$ sysctl -w net.ipv4.conf.all.arp\_ignore=1

Finally, the rp\_filter is set to 1 by default on several distributions (e.g. on CentOS 7) and should be set to either 0 or 2, with 2 being more secure. This is true even if the configuration uses a single logical subnet.

$ sysctl -w net.ipv4.conf.<ifaces>.rp\_filter=2

All those parameters can be made persistent in /etc/sysctl.conf by adding a new sysctl file under /etc/sysctl.d (e.g. /etc/sysctl.d/95-daos-net.conf) with all the relevant settings.

For more information, please refer to the [librdmacm documentation](https://github.com/linux-rdma/rdma-core/blob/master/Documentation/librdmacm.md)

## Runtime Directory Setup

DAOS uses a series of Unix Domain Sockets to communicate between its various components. On modern Linux systems, Unix Domain Sockets are typically stored under /run or /var/run (usually a symlink to /run) and are a mounted tmpfs file system. There are several methods for ensuring the necessary directories are setup.

A sign that this step may have been missed is when starting daos\_server or daos\_agent, you may see the message:

$ mkdir /var/run/daos\_server: permission denied  
Unable to create socket directory: /var/run/daos\_server

### Non-default Directory

By default, daos\_server and daos\_agent will use the directories /var/run/daos\_server and /var/run/daos\_agent respectively. To change the default location that daos\_server uses for its runtime directory, either uncomment and set the socket\_dir configuration value in install/etc/daos\_server.yml, or pass the location to daos\_server on the command line using the -d flag. For the daos\_agent, an alternate location can be passed on the command line using the –runtime\_dir flag.

### Default Directory (non-persistent)

Files and directories created in /run and /var/run only survive until the next reboot. However, if reboots are infrequent, an easy solution while still utilizing the default locations is to create the required directories manually. To do this execute the following commands.

daos\_server:

$ mkdir /var/run/daos\_server  
$ chmod 0755 /var/run/daos\_server  
$ chown user:user /var/run/daos\_server (where user is the user you  
 will run daos\_server as)

daos\_agent:

$ mkdir /var/run/daos\_agent  
$ chmod 0755 /var/run/daos\_agent  
$ chown user:user /var/run/daos\_agent (where user is the user you  
 will run daos\_agent as)

### Default Directory (persistent)

If the server hosting daos\_server or daos\_agent will be rebooted often, systemd provides a persistent mechanism for creating the required directories called tmpfiles.d. This mechanism will be required every time the system is provisioned and requires a reboot to take effect.

To tell systemd to create the necessary directories for DAOS:

* Copy the file utils/systemd/daosfiles.conf to /etc/tmpfiles.d  
  cp utils/systemd/daosfiles.conf /etc/tmpfiles.d
* Modify the copied file to change the user and group fields (currently daos) to the user daos will be run as
* Reboot the system, and the directories will be created automatically on all subsequent reboots.

## Elevated Privileges

DAOS employs a privileged helper binary (daos\_admin) to perform tasks that require elevated privileges on behalf of daos\_server.

### Privileged Helper Configuration

When DAOS is installed from RPM, the daos\_admin helper is automatically installed to the correct location with the correct permissions. The RPM creates a “daos\_server” system group and configures permissions such that daos\_admin may only be invoked from daos\_server.

For non-RPM installations, there are two supported scenarios:

1. daos\_server is run as root, which means that daos\_admin is also invoked as root, and therefore no additional setup is necessary.
2. daos\_server is run as a non-root user, which means that daos\_admin must be manually installed and configured.

The steps to enable the second scenario are as follows (steps are assumed to be running out of a DAOS source tree which may be on a NFS share):

$ chmod -x $daospath/bin/daos\_admin # prevent this copy from being executed  
$ sudo cp $daospath/bin/daos\_admin /usr/bin/daos\_admin  
$ sudo chmod 4755 /usr/bin/daos\_admin # make this copy setuid root  
$ sudo mkdir -p /usr/share/daos/control # create symlinks to SPDK scripts  
$ sudo ln -sf $daospath/share/daos/control/setup\_spdk.sh \  
 /usr/share/daos/control  
$ sudo mkdir -p /usr/share/spdk/scripts  
$ sudo ln -sf $daospath/share/spdk/scripts/setup.sh \  
 /usr/share/spdk/scripts  
$ sudo ln -sf $daospath/share/spdk/scripts/common.sh \  
 /usr/share/spdk/scripts  
$ sudo ln -s $daospath/include \  
 /usr/share/spdk/include

!!! note The RPM installation is preferred for production scenarios. Manual installation is most appropriate for development and predeployment proof-of-concept scenarios.

## Memory Lock Limits

Low ulimit for memlock can cause SPDK to fail and emit the following error:

daos\_engine:1 EAL: cannot set up DMA remapping, error 12 (Cannot allocate  
 memory)

The memlock limit only needs to be manually adjusted when daos\_server is not running as a systemd service. Default ulimit settings vary between OSes.

For RPM installations, the service will typically be launched by systemd and the limit is pre-set to unlimited in the daos\_server.service unit file: https://github.com/daos-stack/daos/blob/master/utils/systemd/daos\_server.service#L16. Note that values set in /etc/security/limits.conf are ignored by services launched by systemd.

For non-RPM installations where daos\_server is launched directly from the commandline, limits should be adjusted in /etc/security/limits.conf as per https://software.intel.com/content/www/us/en/develop/blogs/best-known-methods-for-setting-locked-memory-size.html.