# Software Installation

Please check the [support matrix](https://daos-stack.github.io/release/support_matrix) to select the appropriate software combination.

## Software Dependencies

DAOS requires a C99-capable compiler, a Go compiler, and the scons build tool. Moreover, the DAOS stack leverages the following open source projects:

* [*gRPC*](https://grpc.io/) provides a secured out-of-band channel for DAOS administration.
* [*PMDK*](https://github.com/pmem/pmdk.git) for persistent memory programming.
* [*SPDK*](http://spdk.io/) for userspace NVMe device access and management.
* [*ISA-L*](https://github.com/01org/isa-l) and [*ISA-L\_Crypto*](https://github.com/01org/isa-l_crypto) for checksum and erasure code
* [*Argobots*](https://github.com/pmodels/argobots) for thread management.
* [*hwloc*](https://github.com/open-mpi/hwloc) for discovering system devices, detecting their NUMA node affinity and for CPU binding
* [*libfabric*](https://github.com/ofiwg/libfabric) for detecting fabric interfaces, providers, and connection management.

The DAOS build system can be configured to download and build any missing dependencies automatically.

## Distribution Packages

DAOS RPM packaging is currently available, and DEB packaging is under development and will be available in a future DAOS release. Integration with the [Spack](https://spack.io/) package manager is also under consideration.

### Installing DAOS from RPMs

DAOS RPMs are available from the Intel© Registration Center. Clicking the [Intel© Registration Center](https://registrationcenter.intel.com/forms/?productid=3412) link will take you to the registration center, where you will create an account. After creating an account, the following files can be downloaded:

* daos\_debug.tar - *debuginfo* packages
* daos\_packages.tar - client and server main packages
* daos\_source.tar - source RPMs

Recommended steps after download:

$ sudo tar -C / -xf daos\_packages.tar  
$ sudo cp /opt/intel/daos\_rpms/packages/daos\_packages.repo /etc/yum.repos.d  
$ rm /opt/intel/daos\_rpms/packages/libabt\*  
 (cd /opt/intel/daos\_rpms/packages/ && createrepo .)  
$ sudo yum install epel-release  
$ sudo yum install daos-server  
$ sudo yum install daos-client

!!! note Only daos-client OR daos-server needs to be specified on the yum command line.

## DAOS from Scratch

The following instructions have been verified with CentOS. Installations on other Linux distributions might be similar with some variations. Developers of DAOS may want to review the additional sections below before beginning, for suggestions related specifically to development. Contact us in our [*forum*](https://daos.groups.io/g/daos) for further help with any issues.

### Build Prerequisites

To build DAOS and its dependencies, several software packages must be installed on the system. This includes scons, libuuid, cmocka, ipmctl, and several other packages usually available on all the Linux distributions. Moreover, a Go version of at least 1.10 is required.

An exhaustive list of packages for each supported Linux distribution is maintained in the Docker files (please click on the link):

* [CentOS](https://github.com/daos-stack/daos/blob/master/utils/docker/Dockerfile.centos.7#L53-L79)
* [OpenSUSE](https://github.com/daos-stack/daos/blob/master/utils/docker/Dockerfile.leap.15#L71-L100)
* [Ubuntu](https://github.com/daos-stack/daos/blob/master/utils/docker/Dockerfile.ubuntu.20.04#L21-L39)

The command lines to install the required packages can be extracted from the Docker files by removing the “RUN” command, which is specific to Docker. Check the [utils/docker](https://github.com/daos-stack/daos/tree/master/utils/docker) directory for different Linux distribution versions.

Some DAOS tests use MPI. The DAOS build process uses the environment modules package to detect the presence of MPI. If none is found, the build will skip building those tests.

### DAOS Source Code

The DAOS repository is hosted on [GitHub](https://github.com/daos-stack/daos).

To checkout the latest stable version, simply run:

$ git clone --recurse-submodules -b v1.0.1 https://github.com/daos-stack/daos.git

For the current development version, then run:

$ git clone --recurse-submodules https://github.com/daos-stack/daos.git

This command clones the DAOS git repository (path referred as ${daospath} below) and initializes all the submodules automatically.

### Building DAOS & Dependencies

If all the software dependencies listed previously are already satisfied, then type the following command in the top source directory to build the DAOS stack:

$ scons --config=force install

If you are a developer of DAOS, we recommend following the instructions in the [DAOS for Development](https://daos-stack.github.io/dev/development/#building-daos-for-development) section.

Otherwise, the missing dependencies can be built automatically by invoking scons with the following parameters:

$ scons --config=force --build-deps=yes install

By default, DAOS and its dependencies are installed under ${daospath}/install. The installation path can be modified by adding the PREFIX= option to the above command line (e.g., PREFIX=/usr/local).

!!! note Several parameters can be set (e.g., COMPILER=clang or COMPILER=icc) on the scons command line. Please see scons --help for all the possible options. Those options are also saved for future compilations.

### Environment setup

Once built, the environment must be modified to search for binaries and header files in the installation path. This step is not required if standard locations (e.g. /bin, /sbin, /usr/lib, …) are used.

CPATH=${daospath}/install/include/:$CPATH  
PATH=${daospath}/install/bin/:${daospath}/install/sbin:$PATH  
export CPATH PATH

If using bash, PATH can be set up for you after a build by sourcing the script utils/sl/utils/setup\_local.sh from the daos root. This script utilizes a file generated by the build to determine the location of daos and its dependencies.

If required, ${daospath}/install must be replaced with the alternative path specified through PREFIX.

## DAOS in Docker

This section describes how to build and run the DAOS service in a Docker container. A minimum of 5GB of DRAM and 16GB of disk space will be required. On Mac, please make sure that the Docker settings under “Preferences/{Disk, Memory}” are configured accordingly.

### Building a Docker Image

To build the Docker image directly from GitHub, run the following command:

$ docker build https://github.com/daos-stack/daos.git#master \  
 -f utils/docker/Dockerfile.centos.7 -t daos

or from a local tree:

$ docker build . -f utils/docker/Dockerfile.centos.7 -t daos

This creates a CentOS 7 image, fetches the latest DAOS version from GitHub, builds it, and installs it in the image. For Ubuntu and other Linux distributions, replace Dockerfile.centos.7 with Dockerfile.ubuntu.20.04 and the appropriate version of interest. master should be replaced by the relevant release tag (e.g. v1.0.1) to build a stable version.

### Simple Docker Setup

Once the image created, one can start a container that will eventually run the DAOS service:

$ docker run -it -d --privileged --cap-add=ALL --name server -v /dev:/dev daos

!!! note If you want to be more selective with the devices that are exported to the container, individual devices should be listed and exported as volume via the -v option. In this case, the hugepages devices should also be added to the command line via -v /dev/hugepages:/dev/hugepages and -v /dev/hugepages-1G:/dev/hugepages-1G

!!! warning If Docker is being run on a non-Linux system (e.g., OSX), -v /dev:/dev should be removed from the command line.

The daos\_server\_local.yml configuration file sets up a simple local DAOS system with a single server instance running in the container. By default, it uses 4GB of DRAM to emulate persistent memory and 16GB of bulk storage under /tmp. The storage size can be changed in the yaml file if necessary.

The DAOS service can be started in the docker container as follows:

$ docker exec server daos\_server start \  
 -o /home/daos/daos/utils/config/examples/daos\_server\_local.yml

!!! note Please make sure that the uio\_pci\_generic module is loaded on the host.

Once started, the DAOS server waits for the administrator to format the system. This can be triggered in a different shell, using the following command:

$ docker exec server dmg -i storage format

Upon successful completion of the format, the storage engine is started, and pools can be created using the daos admin tool (see next section).

For more advanced configurations involving SCM, SSD or a real fabric, please refer to the next section.