1. **Docker overview**

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications. By taking advantage of Docker’s methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.

1. **Introduction To Docker and Containers**

DGX-2™, DGX-1™, and DGX Station are designed to run containers. Containers hold the application as well as any libraries or code that are needed to run the application. Containers are portable within an operating system family. For example, you can create a container using Red Hat Enterprise Linux and run it on an Ubuntu system, or vice versa. The only common thread between the two operating systems is that they each need to have the container software so they can run containers. Using containers allows you to create the software on whatever OS you are comfortable with and then run the application wherever you want. It also allows you to share the application with other users without having to rebuild the application on the OS they are using. Containers are different than a virtual machine (VM) such as vmware. A VM has a complete operating system and possibly applications and data files. Containers do not contain a complete operating system.

They only contain the software needed to run the application. The container relies on the host OS for things such as file system services, networking, and an OS kernel. The application in the container will always run the same everywhere, regardless of the OS/compute environment. DGX-2, DGX-1, and DGX Station all use Docker. Docker is one of the most popular container services available and is very commonly used by developers in the Artificial Intelligence (AI) space. There is a public Docker repository that holds pre-built Docker containers. These containers can be a simple base OS such as centos, or they may be a complete application such as tensorflow. You can use these Docker containers for running the applications that they contain. You can use them as the basis for creating other containers, for example for extending a container. To enable portability in Docker images that leverage gpus, NVIDIA developed the NVIDIA Container Runtime for Docker (also known as nvidia-docker2). We will refer to the NVIDIA Container Runtime simply as nvidia-docker2 for the remainder of this guide for brevity. Nvidia-docker2 is an open-source project that provides a command line tool to mount the user-mode components of the NVIDIA driver and the gpus into the Docker container at launch. These containers ensure the best performance for your applications and should provide the best single-GPU performance and multi-GPU scaling.

1. **Preparing Your DGX System for Use with NVIDIA Container Runtime**

**About this task**

Some initial setup is required to be able to access GPU containers from the Docker command line for use on DGX-2, DGX-1, or on a DGX Station, or NGC. As a result of differences between the releases of the DGX™ OS and DGX hardware, the initial setup workflow depends on the DGX system and DGX OS version that you are using. To determine the DGX OS software version on either the DGX-2, DGX-1, or DGX Station, enter the following command: $ grep VERSION /etc/dgx-release

DGX\_SWBUILD\_VERSION="3.1.1"

Based on the output from the command, choose from below which workflow best reflects your environment. Select the topics and perform the steps within that workflow.

**DGX-2 or DGX-1 with DGX OS Server 3.1.1 or Later Workflow**

1. Version 3.1.1 And Later: Preventing IP Address Conflicts Between Docker and DGX

2. Configuring the Use of Proxies

3. Enabling Users to Run Docker Containers

**DGX-1 with DGX OS Server 2.x or Earlier**

1. Version 2.x Or Earlier: Installing Docker and nvidia-docker2

2. Version 2.x Or Earlier: Preventing IP Address Conflicts Between Docker and DGX

3. Configuring the Use of Proxies

4. Enabling Users to Run Docker Containers

**DGX Station Workflow**

1. Version 3.1.1 And Later: Preventing IP Address Conflicts Between Docker and DGX
2. Configuring The Use of Proxies
3. Enabling Users to Run Docker Containers
4. **Preventing IP Address Conflicts with Docker**

To ensure that your DGX system can access the network interfaces for Docker containers, ensure that the containers are configured to use a subnet distinct from other network resources used by your DGX system. By default, Docker uses the 172.17.0.0/16 subnet. If addresses within this range are already used on your DGX system’s network, change the Docker network to specify the IP address of the DNS server, bridge IP address range, and container IP address range to be used by your GPU containers. Consult your network administrator to find out which IP addresses are used by your network.