

Introduction to Containerization

Docker Backup

npNOG 10

November 25 - 28, 2024



This material is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>)

Introduction to Docker Backup

What is Docker Backup?

Docker backup refers to the process of saving container data and configurations to protect against data loss or system failure.

Components to Backup

- Container Data (Volumes)
- Container Images
- Container Configurations
- Docker Compose files

Why Docker Backup is Important

- **Data Protection**
Prevents loss of critical data (e.g., databases, files) stored in Docker volumes.
- **Disaster Recovery**
Enables quick recovery from system crashes, hardware failures, or data corruption.
- **Service Continuity**
Ensures that services can be restored quickly without significant downtime.
- **Portability**
Backups can be used to migrate Docker applications to new servers or environments.
- **Testing and Development**
Backups allow developers to easily replicate production environments for testing, troubleshooting, or creating development environments without affecting live services.
- **Replication**
Docker backups can be used to replicate containers across multiple environments, helping in load balancing, scaling, or setting up identical systems on new infrastructure.

Key Components to Backup

- **Volumes (Data Storage)**
 - Containers often use volumes to store persistent data.
 - Backing up volumes ensures data stored outside the container's lifecycle is safe.
- **Container Configurations**
 - Save Docker images and settings, including environment variables, port mappings, and linked services.
- **Docker Compose Files**
 - These files define multi-container setups and should be backed up to recreate the environment easily.

Backup/Restore Docker Volumes

1. Identify the Volume
2. Create a Backup
3. Store the Backup
4. Restore the Backup

Step 1: Identify the Volume

The first step in taking a backup of a volume is to identify the volume that you want to back up. You can do this by running the following command:

```
docker volume ls
```

This will show you a list of all the volumes that are currently available on your Docker host. Note down the name of the volume that you want to back up.

Step 2: Create a Backup

To create a backup of the volume, you can use the docker run command to start a container that mounts the volume you want to back up and a separate container that writes the backup data to a file.

Here's an example of how to do this:

```
docker run --rm \  
-v "<volume-name>" :<target> -v "$(pwd) " :/backup \  
busybox tar -czvf /backup/<backup-filename>.tar.gz <target>
```

In this command, replace `<volume-name>` with the name of the volume you want to back up, `<target>` with the mount point inside the docker container, and `<backup-filename>` with a name for the backup file.

Step 3: Move the Backup File to an External Server

- After you have created a backup file, it's a good idea to move it to an external server or storage device to ensure that it's safe and secure.
- Storing the backup file on a separate server or storage device can help to protect it in the event of a disaster, such as a server failure or a security breach.
- To move the backup file to an external server, you can use SCP.

```
scp /path/to/backupfile user@external-server:/path/to/destination
```

SCP: Secure Copy (SCP) is a secure file transfer protocol that allows you to transfer files between servers using SSH. To use SCP, you will need to have SSH access to both the source and destination servers.

Step 4: Restore the Volume

If you need to restore the volume from the backup, you can use the docker run command to start a container that mounts the backup file and a separate container that writes the backup data to the volume.

Here's an example of how to do this:

```
docker run --rm \  
-v "<volume-name>":<target> -v "$(pwd)":/backup \  
busybox tar -xzf /backup/<backup-filename>.tar.gz -C <target>
```

In this command, replace `<volume-name>` with the name of the volume you want to restore, `<target>` with the mount point inside the docker container, and `<backup-filename>` with the name of the backup file.

Backup/Restore Container Configurations (Images)

- **Backup Container Configurations (Images):**
 - Save Docker images to a file.

```
docker save -o image_name_backup.tar image_name
```

- **Restore Container Configurations (Images):**
 - Load the image from the backup file.

```
docker load -i image_name_backup.tar
```

Backup/Restore Docker Compose Files

- **Backing Up Docker Compose Files:**

- Copy the `docker-compose.yml` to your backup directory.

```
cp /path/to/docker-compose.yml /backup/
```

- **Restore Docker Compose Environment:**

- Place the `docker-compose.yml` in the correct directory and recreate the environment.

```
cd /path/to  
cp /backup/docker-compose.yml .  
docker-compose up -d
```

