This guide will walk you through creating systemd services to manage your Docker Compose deployments. This ensures that your services start automatically on boot and can be managed easily using standard systemd commands.

Prerequisites

Before you begin, ensure the following prerequisites are met:

- You have Docker and Docker Compose installed on your server.

- You have completed the deployment steps for Liberty Framework using Docker Compose.

Creating the Systemd Service for Admin Tools

1. Create a service file for `docker-admin`:

```
1   sudo nano /etc/systemd/system/docker-admin.service
```

2. Paste the following content into the file:

```
1   [Unit]
2   Description=Liberty Admin Tools Service
3   PartOf=docker.service
4   After=docker.service
5
6   [Service]
7   Type=simple
8   RemainAfterExit=true
9   WorkingDirectory=/app/liberty-admin/
10  ExecStart=/usr/local/bin/docker-compose -f /app/liberty-admin/docker-compose.yml start
11  ExecStop=/usr/local/bin/docker-compose -f /app/liberty-admin/docker-compose.yml stop
12
13  [Install]
14  WantedBy=multi-user.target
```

3. Save and close the file.

Creating the Systemd Service for Liberty Framework

1. Open a terminal.

2. Create a new directory:

```
1   mkdir -p /app/liberty-framework
2   cd /app/liberty-framework
```

3. Download the Docker Compose file from the provided URL, Using `curl`:

```
1   curl -L -o docker-compose.yml https://github.com/fblettner/liberty-
    public/blob/main/release/latest/liberty-framework.yml
```

4. Create a service file for `docker-liberty`:

```
1   sudo nano /etc/systemd/system/docker-liberty.service
```

5. Paste the following content into the file:

```
1   [Unit]
2   Description=Liberty Framework Service
3   PartOf=docker.service
4   After=docker.service
5
6   [Service]
7   Type=simple
8   RemainAfterExit=true
9   WorkingDirectory=/app/liberty/
10  ExecStart=/usr/local/bin/docker-compose -f /app/liberty-framework/docker-compose.yml start
11  ExecStop=/usr/local/bin/docker-compose -f /app/liberty-framework/liberty-compose.yaml stop
12
13  [Install]
14  WantedBy=multi-user.target
```

6. Save and close the file.

Enabling and Starting the Services

1. Enable the created services to start on boot:

```
1   sudo systemctl enable docker-liberty.service
2   sudo systemctl enable docker-admin.service
```

2. Start the services immediately:

```
1   sudo systemctl start docker-liberty.service
2   sudo systemctl start docker-admin.service
```

3. Check the status of the services to ensure they are running:

```
1   sudo systemctl status docker-liberty.service
2   sudo systemctl status docker-admin.service
```

Additional Resources

- Systemd Documentation

- Docker Documentation

- Docker Compose Documentation

By following this guide, you should be able to create and manage systemd services for your Docker Compose deployments seamlessly. If you run into any issues or have any questions, refer to the additional resources provided or reach out to the respective support communities.