

## Standard Operating Procedure (SOP)

### Connecting to a Robot via SSH and Accessing Docker Containers on Linux

---

#### 1. Purpose

This SOP defines the standardized procedure for connecting to a robot through SSH from a Linux system and then accessing and operating inside its Docker containers. This ensures consistent, secure, and reliable access for development, debugging, and maintenance.

---

#### 2. Scope

This procedure applies to engineers, technicians, and developers who access robot systems over the network and need to interact with ROS, robot firmware, or applications running inside Docker containers on the robot.

---

#### 3. Prerequisites

Before connecting:

- You are on a **Linux system** with SSH installed.
  - You have the robot's **IP address** or **hostname**.
  - The robot is **powered on** and connected to the same network.
  - SSH is **enabled** on the robot.
  - You have valid **SSH credentials** (username + password or key).
  - Docker is installed and running on the robot.
  - You know the container name or ID you intend to access.
- 

#### 4. Required Tools

- `ssh` client (pre-installed on Linux)
  - Docker CLI tools (pre-installed on robot)
-

## 5. Procedure

### 5.1 Identify Robot IP Address

- Use one of the following:
    - nmap network scan
    - Robot display or software
- 

### 5.2 SSH into the Robot (Password Authentication)

1. Open a terminal on your Linux machine.
2. Connect using:

```
ssh <username>@<robot_ip>
```

**Example:**

```
ssh robot@192.168.0.50
```

3. If connecting for the first time, type yes to trust the device.
  4. Enter the password when prompted.
- 

### 5.3 SSH into the Robot (Key-Based Authentication)

If the robot uses SSH key authentication:

```
ssh -i ~/.ssh/robot_key <username>@<robot_ip>
```

Example:

```
ssh -i ~/.ssh/id_rsa robot@192.168.0.50
```

---

### 5.4 Verify Successful Login

A successful login shows something like:

```
robot@robot-desktop:~$
```

Test system status:

```
uname -a
```

---

## 5.5 Listing Available Docker Containers on the Robot

Once logged in:

```
docker ps -a
```

This displays running and stopped containers.

---

## 5.6 Starting a Docker Container

If the required container is not running:

```
docker start <container_name>
```

Example:

```
docker start robot_navigation
```

---

## 5.7 Accessing a Docker Container Shell

To enter a running container:

```
docker exec -it <container_name> bash
```

Example:

```
docker exec -it robot_navigation bash
```

You will now be inside the container environment. Typical prompt:

```
root@container-id:/app#
```

---

## 5.8 Stopping a Docker Container (Optional)

If needed:

```
docker stop <container_name>
```

---

## 5.9 Exiting the Container and SSH Session

- Exit container:

```
exit
```

- Exit robot SSH session:

```
exit
```

---

## 6. Troubleshooting

### ***Cannot Connect (No Route / Timeout)***

- Confirm correct IP.
- Ensure network connectivity.
- Ping the robot:

```
ping <robot_ip>
```
- Ensure the robot's firewall allows SSH.

### ***Permission Denied***

- Verify username.
- Check SSH key permissions:

```
chmod 600 ~/.ssh/id_rsa
```

### ***Docker Command Requires Root***

If you see permission errors:

```
sudo usermod -aG docker <username>
```

Then reboot the robot.

### ***Container Not Starting***

Check logs:

```
sudo docker logs <container_name>
```

---

## 7. Safety & Security Best Practices

- Always use SSH keys when possible.
- Disable root login over SSH.
- Avoid running unnecessary containers.
- Keep robot firmware and software updated.
- Never expose SSH to public networks without protection.

---

## 8. Revision History

Version	Date	Description	Author
2.0	2025-12-03	Updated SOP for robot SSH + Docker workflow	RobotoAI

---

For more information check out the github link

<https://github.com/TeamRobotoAI/dynominion/tree/main>

