SOP
Secure Linux Management & Multi-Host Orchestration

v1.0.1

Author:

Eldon Gabriel

March 19, 2026

TABLE OF CONTENTS

REVISION HISTORY

Version	Date	Author	Description of Changes
1.0.0	01/27/2026	Eldon G.	Initial draft
1.0.1	03/19/2026	Eldon G.	Updated conclusion, key takeaways, and security recommendations

Disclaimer: This guide documents the author’s personal work in completing a Mossé Cyber Security Institute (MCSI) lab exercise. This reflects the author’s independent configuration and administration of the Cockpit web console for managing Linux systems (Ubuntu 24.04) in a controlled environment. No MCSI proprietary video content, instructor-led materials, or restricted lab instructions were reproduced or distributed. All technical findings and documentation follow MCSI’s academic integrity and disclosure policies.

0.0 Executive Summary

This Standard Operating Procedure (SOP) explains how to install and securely configure **Cockpit** on **Ubuntu 24.04**. The goal is to manage Linux systems from a single web dashboard. This includes controlling services, checking user accounts, and managing multiple systems.

The guide also focuses on security. It shows how to protect access using **UFW firewall rules** and how to follow basic security standards, such as the **NIST CSF and PCI-DSS,** for logging.

1.0 Cockpit Setup & Service Control

1.1 Install Cockpit

```
sudo apt update
```

```
sudo apt install cockpit -y
```

1.2 Enable and Start Cockpit

```
sudo systemctl enable --now cockpit.socket
```

1.3 Verify Cockpit Service

```
sudo systemctl status cockpit.socket
```

2.0 Access & Network Verification

2.1 Default Access URL

```
https://<VM-IP>:9090
```

Cockpit uses **port 9090** over HTTPS by default.

2.2 Confirm Cockpit Listening Port

```
sudo ss -tulpn | grep 9090
```

2.3 UFW Firewall Configuration Check

Ubuntu’s default firewall tool is an Uncomplicated Firewall (UFW). It is used to control incoming and outgoing network traffic.

```
sudo ufw status
```

Allow Cockpit if needed:

```
sudo ufw allow 9090/tcp
```

3.0 Viewing Running Services In Cockpit

3.1 Navigate to Services

In the Cockpit web interface,

System → Services

3.2 Identify Services

From the Services page, you can:

- View running, stopped, and disabled services
- Identify core services, such as
  - `ssh`
  - `cron`
  - `systemd-journald`
  - `NetworkManager`

3.3 Start / Stop Services

Select a service and toggle between

- *Start and Enable*
- *Disable*

4.0 User Account Identification

4.1 View User Accounts

Navigate to:

Accounts

Cockpit displays:

- Local system users
- UID information
- Group membership
- Login permissions

4.2 Expected Behavior

- Any user created previously on the system will appear here
- Users persist across clones unless manually deleted
- Cockpit reads system users directly from `/etc/passwd`

5.0 Command Execution Via Cockpit

5.1 Open Terminal

Navigate to:

Terminal

5.2 Execute Commands

Example:

- `uptime`
- `whoami`
- `ip a`

6.0 Multi-Host Management In Cockpit

6.1 Requirements for Secondary VM

The second Linux VM must have the following:

- SSH is installed and running

```
sudo apt install openssh-server -y
```

```
sudo systemctl enable --now ssh
```

- Network connectivity to the primary **Cockpit** host
- A valid local user account with a password

6.2 Add Remote Host

In **Cockpit**:

Dashboard → *Add new host*

Provide:

- **IP address** of secondary VM
- Username on secondary VM or leave empty to connect with the current user
- Password or SSH key

6.3 Verify Multi-Host Management

Once connected, demonstrate the following:

- Switching between hosts
- Viewing services on the second VM
- Running commands on the second VM
- Viewing user accounts on the second VM

7.0 Logs & Troubleshooting

7.1 View System Logs

Cockpit provides:

- `systemd` journal access
- Service-specific logs
- Filtering by severity and service

7.2 Common Issues

- **Connection timeout:** SSH not reachable on secondary VM
- **Auth failure:** PasswordAuthentication disabled in `sshd`
- **Host unreachable:** Incorrect network mode or **IP**

7.3 Security Notes

- **Cockpit** runs with system privileges via authenticated users
- Multi-host mode uses SSH and loads remote web components
- Only connect to **trusted systems**
- Keep **Cockpit** updated via system updates

7.4 Validation Checklist

✔ **Cockpit** installed and accessible via browser

✔ **Services** identified and managed via **Cockpit**

✔ **User accounts** identified via **Cockpit**

✔ Commands executed via the **Cockpit Terminal**

✔ Secondary Linux VM added and managed

✔ Multi-host administration demonstrated

8.0 Conclusion

The setup and testing of Cockpit on Ubuntu 24.04 showed that a web-based tool can simplify and improve the efficiency of Linux system management. It combines service control, user management, and multihost access into a single dashboard. This allows administrators to monitor systems at a high level while running detailed commands when needed.

8.1 Key Takeaways

Operational Efficiency: The Cockpit allows administrators to manage services and system logs across multiple systems from a single dashboard. This eliminates the need to open multiple SSH sessions.

Enhanced Visibility: Admins can quickly view user account details from the `/etc/passwd`. This helps identify active users and their permissions.

Scalable Administration: Multiple systems can be managed through SSH without installing additional software on each machine.

Practical Deployment: The cockpit can be installed quickly. Enabling the `cockpit.socket` service allows secure access with minimal downtime.

8.2 Security Implications and Recommendations

Cockpit improves system management; however, it also adds a web-based access point that must be secured.

Implemented Security Measures & Remediations

Unrestricted Network Access: Cockpit uses port 9090 by default, and this port can be detected by attackers.

Hardening Measure Implemented: UFW was configured to limit access to port 9090 to trusted admin IP ranges (for example, 10.x.x.x). This reduces the risk of brute-force attacks.

Privileged Access Risk: Cockpit runs with system-level access based on the logged-in user.

Remediation: Strong password policies were enforced, and SSH key authentication was established for admin accounts.

SSH Lateral Movement: Managing multiple systems depends on SSH security.

Remediation: The SSH configuration on the secondary VM was hardened by disabling root login and using secure settings.

Technical and Procedural Recommendations

Continuous Patch Management: Maintain the Cockpit and all system packages up to date through regular patching.

Encryption Protocols: Access the Cockpit using HTTPS (TLS) to protect login credentials.

Zero-Trust Management: Connect trusted and secure systems to Cockpit to reduce risk.6

Framework & Compliance Mapping

NIST CSF (PR.AC-1): Support access control by showing user accounts and permissions in one place.

ISO 27001 (A.9.2.2): Helps manage and reviews user access using the Accounts feature.

PCI-DSS Requirement 10: System logs can be used to track user activities and events.