Linux Server Administration

Information on Linux Server administration functions

- Using tmux for Safe Remote Linux Server Upgrades

Using tmux for Safe Remote Linux Server Upgrades

> **Purpose:** This guide explains how to use `tmux` to safely perform remote Linux server upgrades over SSH. This is especially useful for Proxmox, XCP-ng, Debian, Ubuntu, and other Linux-based systems where a network drop could interrupt your terminal session.

Why Use tmux?

When performing upgrades remotely over SSH, there is always a risk that your VPN, Wi-Fi, internet connection, or SSH session may disconnect before the upgrade finishes. If that happens during a normal SSH session, the command may be interrupted or you may lose visibility into the upgrade process.

`tmux` solves this by creating a persistent terminal session on the server itself. If your SSH session disconnects, the upgrade keeps running inside `tmux`. You can reconnect later and attach back to the same session.

> **Important:** tmux does not protect you from a failed upgrade, bad repository configuration, broken networking, or a failed reboot. It only protects your running terminal session from SSH/VPN disconnections.

When to Use tmux

Use `tmux` when performing remote maintenance such as:

- Proxmox VE upgrades
- XCP-ng host updates
- Debian or Ubuntu distribution upgrades
- Kernel updates

- Large package upgrades
- Long-running scripts
- Remote firewall/router/server maintenance

Basic tmux Workflow

1. Install tmux

On Debian, Ubuntu, or Proxmox:

```
apt update
apt install tmux -y
```

On RHEL, Rocky Linux, AlmaLinux, or CentOS-based systems:

```
dnf install tmux -y
```

On older CentOS systems:

```
yum install tmux -y
```

On XCP-ng, if available from enabled repositories:

```
yum install tmux -y
```

> **Note:** Some appliance-style systems may not include tmux by default. If tmux is not available, you may need to enable the appropriate repository or use `screen` as an alternative.

---

2. Start a Named tmux Session

Create a named session before starting the upgrade:

```
tmux new -s server-upgrade
```

You are now inside a tmux session named `server-upgrade`.

> **Best Practice:** Always use a descriptive tmux session name, such as `pve9upgrade`, `xcp-update`, or `ubuntu-upgrade`.

---

3. Run the Upgrade Inside tmux

Once inside tmux, run your upgrade commands.

Example: Debian / Ubuntu / Proxmox

```
apt update
apt dist-upgrade
```

For normal updates:

```
apt update
apt upgrade
```

Example: XCP-ng

```
yum update
```

Or, depending on the update procedure being followed:

```
yum update xcp-ng-release
yum update
```

> **Do not blindly use -y during major upgrades.** For major upgrades, avoid commands like `apt dist-upgrade -y` or `yum update -y` unless you are fully confident. It is safer to review prompts manually, especially when configuration files are involved.

Detaching and Reattaching

Detach from tmux Without Stopping the Upgrade

To leave the tmux session running in the background:

```
CTRL + B, then press D
```

This detaches your terminal from tmux. The upgrade continues running on the server.

List Existing tmux Sessions

```
tmux ls
```

Example output:

```
server-upgrade: 1 windows (created Sun May 17 10:15:22 2026)
```

Reconnect to an Existing tmux Session

```
tmux attach -t server-upgrade
```

If your VPN or SSH session drops, reconnect to the server and run the attach command above.

Recommended Upgrade Safety Checklist

Before Starting

- Confirm you have a working backup.
- Confirm you have SSH access.
- Confirm you have console/IPMI/KVM access if possible.
- Check available disk space.
- Check current OS and kernel version.
- Start a tmux session before running upgrade commands.

```
df -h
uname -a
cat /etc/os-release
```

For Proxmox VE Upgrades

Before a major Proxmox upgrade, run the upgrade checklist tool first.

```
pve8to9 --full
```

Only proceed when the report shows:

```
WARNINGS: 0
FAILURES: 0
```

Then start tmux:

```
tmux new -s pve9upgrade
```

Run the upgrade from inside tmux:

```
apt update
apt dist-upgrade
```

> **Critical:** If you are upgrading Proxmox remotely over VPN/SSH only, understand that tmux protects the upgrade process from SSH disconnects, but it cannot help if the server fails to boot or networking does not come back after reboot.

For XCP-ng Updates

Start tmux before running updates:

```
tmux new -s xcp-update
```

Then run:

```
yum update
```

After the update completes, reboot only if required:

```
reboot
```

> **Important for Hypervisors:** Shut down or migrate running VMs before host updates when required by your maintenance plan.

Common tmux Commands

Task	Command / Shortcut
Start a new named session	`tmux new -s server-upgrade`
Detach from session	`CTRL + B`, then `D`
List sessions	`tmux ls`
Attach to session	`tmux attach -t server-upgrade`
Kill a session	`tmux kill-session -t server-upgrade`
Create a new window	`CTRL + B`, then `C`
Move to next window	`CTRL + B`, then `N`
Move to previous window	`CTRL + B`, then `P`

Example: Full Remote Upgrade Workflow

```
# 1. Connect to the server over SSH
ssh root@server-ip-address

# 2. Install tmux if needed
apt update
apt install tmux -y

# 3. Start a named tmux session
tmux new -s server-upgrade
```

```
# 4. Run pre-checks
df -h
uname -a
cat /etc/os-release

# 5. Run the upgrade
apt update
apt dist-upgrade

# 6. Reboot only after the upgrade fully completes
reboot
```

What to Do If SSH Disconnects

If your SSH or VPN session disconnects during the upgrade:

1. Reconnect to the VPN.
2. SSH back into the server.
3. List tmux sessions.
4. Attach to the upgrade session.

```
ssh root@server-ip-address
tmux ls
tmux attach -t server-upgrade
```

You should see the upgrade still running or completed inside the tmux session.

After the Upgrade Completes

Debian / Ubuntu / Proxmox

```
pveversion
uname -r
systemctl status pveproxy pvedaemon pvescheduler pvestatd --no-pager
```

For general Debian/Ubuntu systems:

```
uname -r
cat /etc/os-release
systemctl --failed
```

XCP-ng

```
xe host-list
uname -r
systemctl --failed
```

Best Practices

- Use tmux for every remote upgrade.
- Do not run major upgrades from a normal SSH session.
- Do not reboot until package upgrades fully complete.
- Always verify backups before hypervisor upgrades.
- Use a maintenance window.
- Have console, IPMI, iLO, iDRAC, or onsite access available when possible.
- For major upgrades, avoid automatic yes flags like `-y`.
- Document the commands you ran and the final version after reboot.

Quick Reference

```
# Install tmux
apt install tmux -y

# Start session
tmux new -s upgrade

# Detach
CTRL + B, then D

# List sessions
tmux ls

# Reattach
tmux attach -t upgrade

# Kill session after done
tmux kill-session -t upgrade
```

Summary: tmux is one of the safest tools to use when performing remote upgrades. It allows the upgrade to continue even if SSH or VPN disconnects, making it extremely useful for Proxmox, XCP-ng, Debian, Ubuntu, and other Linux servers.