# HolOS v0.0.10 Guide

Welcome to the official manual for **HolOS v0.0.10**.



One thing you'll notice immediately: HolOS is a **"Root-First"** environment. Unlike most Linux systems where you are a limited user, here you are the "Captain" (Root) by default. This makes things much simpler. No passwords to remember and no sudo commands required!

---

## 1. Understanding Your Environment

When you turn on your HolOS node, you will see a prompt that looks like this: holos#.

- **The # Symbol:** This means you are logged in as **Root**. You have full permission to change anything on the system.
- **No sudo:** If you try to type sudo, the system will say "command not found." This is normal! You don't need to "ask permission" because you are already the administrator.
- **Login:** If you are ever asked to log in, the username is root. Just press the **Enter** key when it asks for a password (there isn't one).

---

## 2. Remote Access: Using GitHub SSH Keys

While you can type commands directly into the node with a keyboard and monitor, it is much easier to manage your Holoport from your primary laptop or desktop using **SSH (Secure Shell)**.

### Why use SSH?

- **Copy/Paste:** It is much easier to copy long scripts or commands from this guide and paste them into a terminal on your main computer.
- **Headless Setup:** You can tuck your Holoport away in a closet—no monitor required.

- **Security:** Even though the local login has no password, SSH ensures that *only* people with your private key can access the node over the network.

---

# 3. The SSH Setup Guide (Mac, Windows, & Linux)

To use the `github_usernames` feature, you need a GitHub account and a "Key Pair." Think of the **Public Key** as a padlock you put on the Holoport, and the **Private Key** as the only physical key on your laptop that can open it.

## Step 1: Get a GitHub Account

If you don't have one, sign up at GitHub.com. It's free. Take note of your **Username**.

## Step 2: Generate your Keys

Open the **Terminal** (Mac/Linux) or **PowerShell** (Windows) on your personal computer and type:

Bash

```
None
ssh-keygen -t ed25519 -C "your_email@example.com"
```

1. When it says "Enter file in which to save," just press **Enter**.
2. When it asks for a "passphrase," you can press **Enter** for no password, or type one for extra security.
3. Your keys are now created!

## Step 3: Add Your Public Key to GitHub

You need to give GitHub your "padlock" (Public Key) so HolOS can go and grab it.

1. **Find the key:**
   - **Mac/Linux:** Type `cat ~/.ssh/id_ed25519.pub`
   - **Windows:** Type `cat $HOME\.ssh\id_ed25519.pub`
2. **Copy the text** that appears (it starts with `ssh-ed25519`).
3. Go to GitHub: **Settings** -> **SSH and GPG keys** -> **New SSH Key**.
4. Paste your key and save it.

## Step 4: Find your Holoport's IP Address

On the Holoport (using the attached keyboard/monitor), type:

Bash

```
None
hostname -I
```

**The result:** You will see a string of numbers like `192.168.1.42`. Note it down.

### Step 5: Connecting to your Holoport

Once you have added your GitHub username to the boot options (see Section 8), type this from your **laptop's** terminal:

Bash

```
None
ssh root@192.168.1.42
```

*(Replace `192.168.1.42` with the actual number you found in Step 4.)*

---

## 4. Ensure your EdgeNode survives a Reboot

By default, Docker containers stop when the power goes out.

To make sure your `edgenode` starts up automatically every time the computer turns on, simply run this command:

Bash

```
None
docker update --restart unless-stopped edgenode
```

**What this does:** This tells Docker, "If this container stops because of a crash or a power cycle, turn it back on immediately."

---

## 5. Manual LED Control

The holos-config tool is your "remote control" for the system LEDs.

```shell
holos-config set-led --color <COLOR_NAME>
```

**Available Options:**

- **`--color`**: Choose from `aurora` (default), `red`, `blue`, `green`, `orange`, `purple`, `yellow`, or `off`.
- **`--flash`**: (Optional) Add this flag to make the color blink.

| Goal | Command |
| --- | --- |
| **Default Rainbow** | holos-config set-led --color aurora |
| **Solid Green** | holos-config set-led --color green |
| **Blinking Red** | holos-config set-led --color red --flash |
| **Lights Off** | holos-config set-led --color off |

---

## 6. Creating Your First Automation Scripts

Because `holos-config` is a CLI tool, you can wrap it in scripts to make your hardware reactive to system states.

## Use Case: Docker "Heartbeat" Monitor

Set your LED to act as a visual status indicator for a specific service (e.g., a Holochain Happ).

Let's make a script that turns the LED **Aurora** if your `edgenode` container is running, and **Flashes Orange** if it isn't.

1. **Create the file:** `nano /usr/local/bin/led-check.sh`
2. **Write or Paste this code:**

Bash

```
None
1. #!/bin/bash
2. CONTAINER="edgenode"
3.
4. if [ "$(docker inspect -f '{{.State.Running}}' $CONTAINER
   2>/dev/null)" == "true" ]; then
5.     holos-config set-led --color aurora
6. else
7.     holos-config set-led --color orange --flash
8. fi
```

3. **Save and Exit:** Ctrl+O, Enter, Ctrl+X.
4. **Make it a program:** `chmod +x /usr/local/bin/led-check.sh`

## Use Case: Visualizing System Load

You can set the LED to flash **Purple** if the CPU load exceeds a specific threshold.

Bash

```
None
1. LOAD=$(uptime | awk -F'[a-z, ]+' '{print $6}')
2. THRESHOLD=2.0
3.
4. if (( $(echo "$LOAD > $THRESHOLD" | bc -l) )); then
5.     holos-config set-led --color purple --flash
6. else
7.     holos-config set-led --color aurora
```

```
8. fi
```

---

## 7. Running at Boot (Systemd Services)

To keep your LED monitor running automatically, create a service:

1. nano /etc/systemd/system/holos-led.service
2. Write or Paste this:

Ini, TOML

```
None
1. [Unit]
2. Description=HolOS LED Status Monitor
3. After=docker.service
4.
5. [Service]
6. ExecStart=/usr/local/bin/led-check.sh
7. Restart=always
8. RestartSec=10
9.
10. [Install]
11. WantedBy=multi-user.target
```

3. **Activate:**
    - systemctl daemon-reload
    - systemctl enable --now holos-led.service

---

## 8. Advanced Boot-Time Provisioning

To make your settings permanent (Wi-Fi and GitHub username):

1. Open config: nano /etc/default/grub

2. Add your info to `GRUB_CMDLINE_LINUX_DEFAULT`: `"quiet splash wifi_ssid=MyNetwork wifi_psk=MyPassword github_usernames=YourGitHubName"`
3. **Apply changes:** `update-grub`

---

# 9. Troubleshooting

- **"Command not found: sudo"**: You are root! Just leave `sudo` off.
- **Case Sensitivity**: Remember that `red` works, but `Red` does not.
- **Check the Logs**: If the LED isn't acting right, see what the system is thinking:
- Bash

```
None
journalctl -t holos-config
```