🏡 How to Set Up an Always-On Holochain Node For Your Moss Group Tools with Edge Node

Welcome! This guide will walk you through setting up an **Edge Node**, an open-source tool that creates a reliable, always-on peer for your Holochain applications (hApps).

🤔 What is an Edge Node and Why Do I Need One?

Imagine your small group is using a private, peer-to-peer application. For the app to work, at least two people need to be online at the same time to share and sync data. If only one person is online and adds new information, then goes offline before anyone else comes back, that new data is temporarily unavailable. The group is out of sync.

An **Edge Node solves this problem**. It's like a dedicated member of your group that is **always online**. You can install it on any spare computer (physical or virtual), and it will run 24/7, holding a copy of the group's data and ensuring that anyone who comes online can sync up with the latest information. It provides stability and reliability for your hApp's network.

This guide will show you how to turn a dedicated machine into an "always-on" node for a private group you've created in **Moss**.

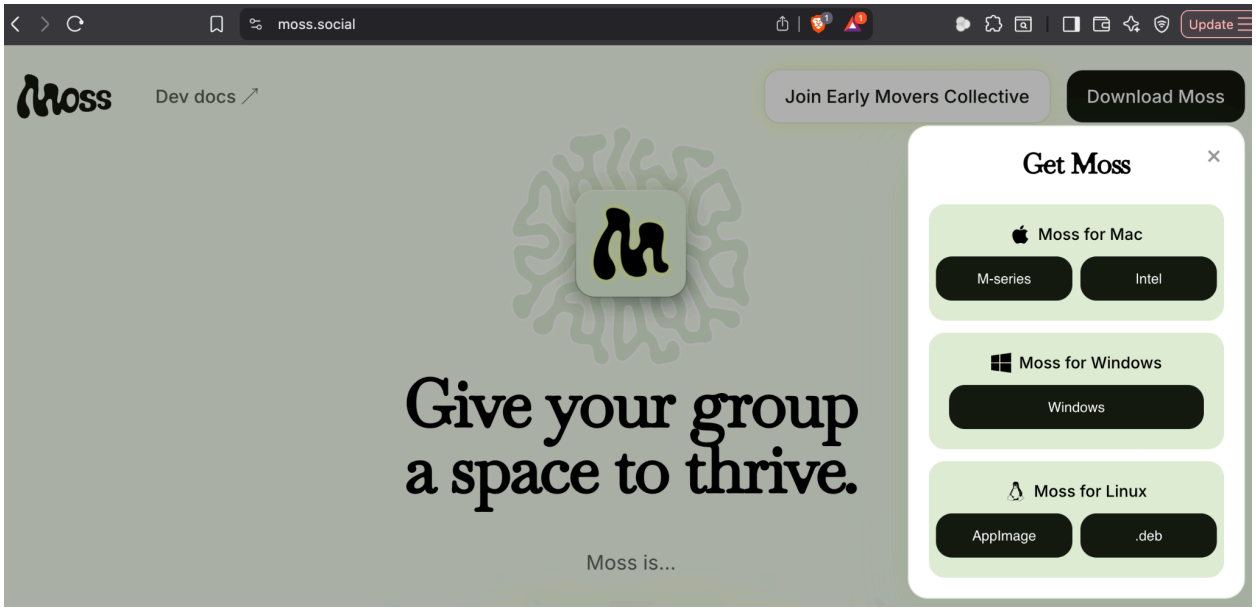
✅ What You'll Need

Before you begin, make sure you have the following:

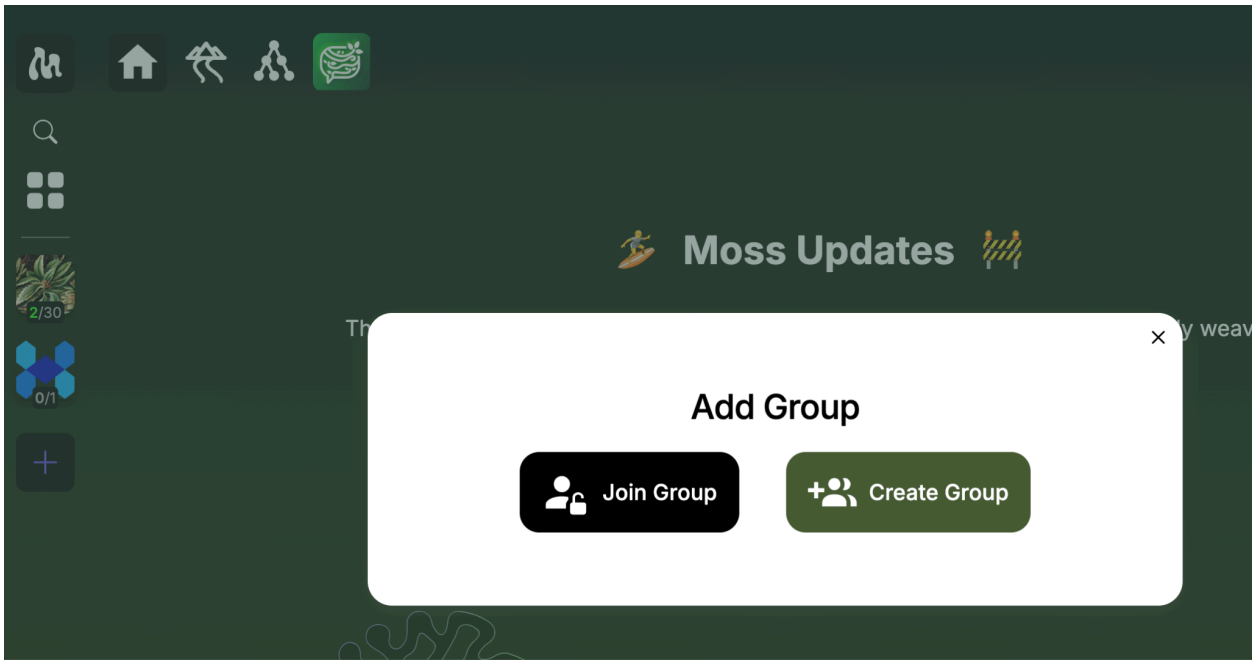
- **A dedicated machine:** This can be a HoloPort, an old laptop, or any spare computer that you can leave running and connected to the internet. **Note:** The process will erase all data on this machine.
    - Keyboard
    - Internet Connection
    - HDMI cable + Monitor
- **A USB drive:** At least 4GB in size.
- **The latest HolOS software:** You'll need to download the `.iso` file.
    - https://github.com/Holo-Host/edgenode/releases/download/v0.0.7ga.5/holos-v0.0.7ga.5.iso
        - sha256:3849cc185868f74f149f468e781447247d8e5afe988f54deba89563d0b1d1804
- **A USB flashing tool:** We recommend using BalenaEtcher (for Mac/Windows/Linux) or Rufus (for Windows).
- **The latest Moss installed on your computer:**
    - To download the latest version of Moss visit https://moss.social

⚙️ Part 1: Install Moss on Your Computer & Create a Group

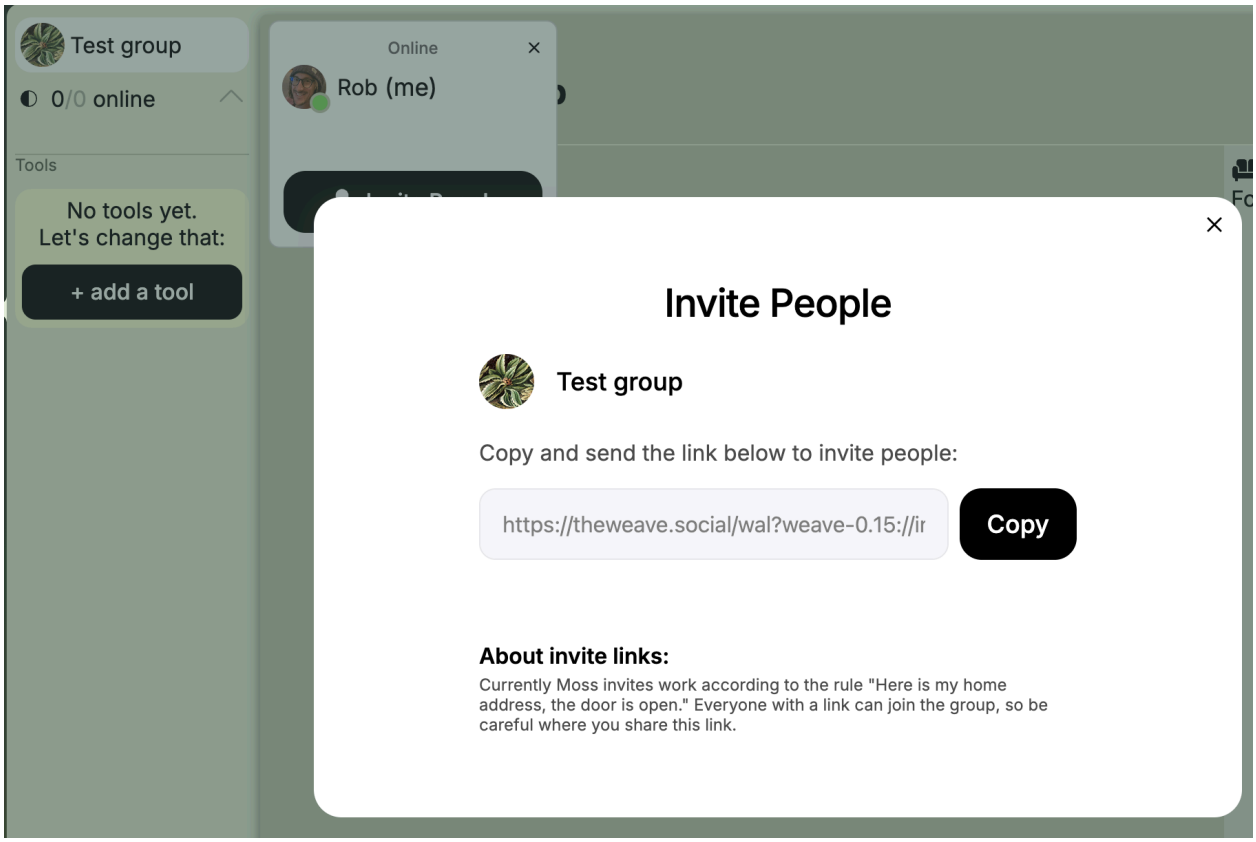
1. Download and install Moss for your computer



2. Run Moss and click the “+” icon on the lefthand navigation bar to add a group



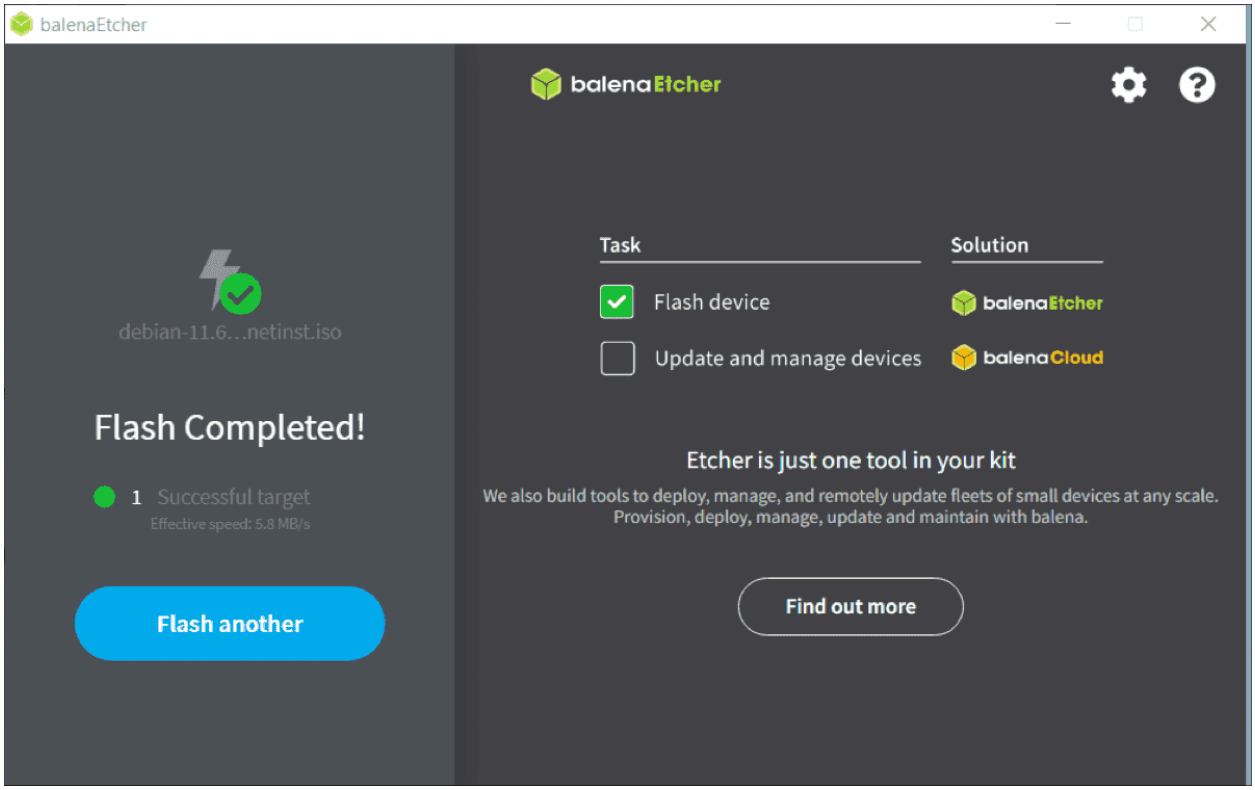
3. Click the caret (arrow) in the top left to expand online users, and click on the button “Invite People”
    a. This will reveal an invite link you can copy and later use to configure and set up your Edge Node for the Moss Group.



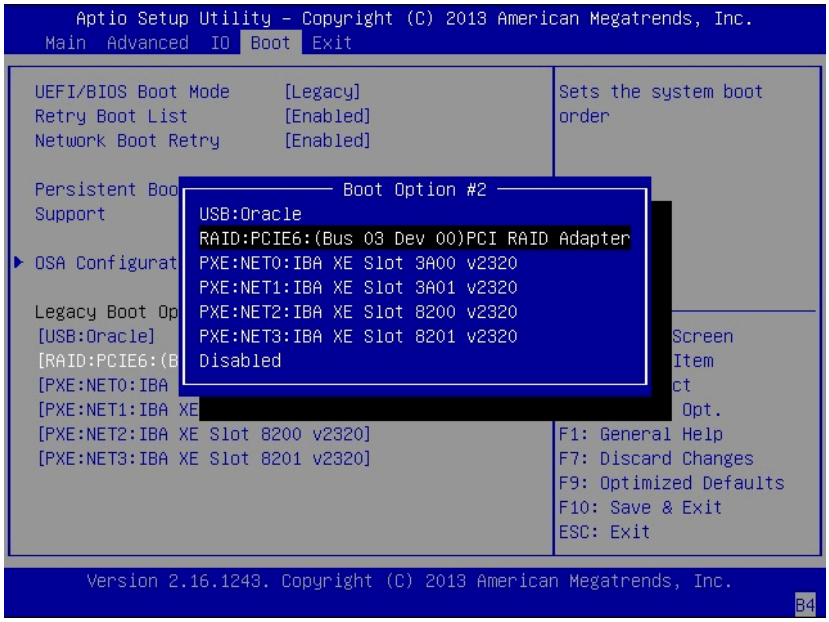
⚙️ Part 2: Install HolOS on Your Dedicated Machine

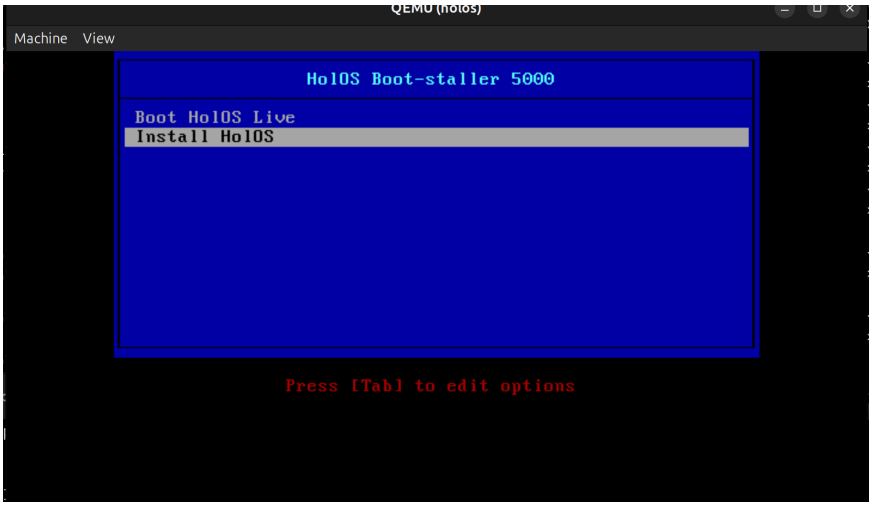
First, we need to install HolOS, the operating system that will run your Edge Node.

1. **Create a Bootable USB Drive:**
    - Open Rufus*, balenaEtcher. *Note: Use Rufus in DD mode
    - Select the HolOS `.iso` file you downloaded.
    - Select your USB drive as the target.
    - Click "Start" or "Flash!" and wait for the process to complete.



2. **Boot from the USB Drive:**
    - Plug the newly created bootable USB drive into your dedicated machine.
    - Restart the machine. As it boots up, press the key to enter the BIOS/Boot Menu. This is usually `F1`, `F2`, `F12`, `Escape` or `Delete`.
        i. Note: If you have a HoloPort tap and slowly keep tapping F11 key as your HoloPort is booting up. This loads up a boot time selection where you can select your boot disk (USB)
    - In the BIOS menu, find the "Boot Order" or "Boot Priority" settings.
    - Set the USB drive as the first boot device.
    - Save your changes and exit. The machine will now boot from the USB drive into the HolOS live environment.



3. **Install HolOS to the Hard Drive:**
    - Select Install HolOS and press return
        i. 

    - Additionally, if you boot into the HolOS live environment, you'll also have access to a command line. There is no password for the `root` user, simply type ‘root’ and press return key.
    - To install HolOS permanently from the live environment, you can run the installer command, telling it which hard drive to use. In most cases, the main hard drive is `sda`.
4. ⚠️ **Warning:** This step will completely erase all data on the specified hard drive. Double-check you have selected the correct machine and drive.
   Type the following command and press Enter:
   Bash

None

```
install-draft sda
```

    - The installer will copy the system to the hard drive and make it bootable. Once it's finished, it will automatically reboot the machine.
5. **First Boot from Hard Drive:**
    - **Remove the USB drive** as the machine reboots.
    - If the reboot is not automatically triggered, you may need to manually shut down and restart the machine.
    - The machine should now boot into your fresh installation of HolOS. Networking should be configured automatically if you're using an Ethernet cable.

🐳 Part 3: Install and Run the Edge Node Container

Now that HolOS is running, we'll use Docker (a tool for running containerized applications) to get the Edge Node software.

1. **Pull the Docker Image:** This command downloads the latest Edge Node package from the official repository.
   Bash

None

```
docker pull ghcr.io/holo-host/edgenode
```

2. **Launch the Edge Node Container:** This command starts the Edge Node and ensures any data it stores (like your hApp information) will be saved persistently on your machine's hard drive. *Note: the command below is a single command NOT two commands*
   Bash

None

```
docker run --name edgenode -dit -v $(pwd)/holo-data:/data
ghcr.io/holo-host/edgenode
```

3. **Access the Container:** Your Edge Node is now running! To configure it, you need to "enter" the container's command line environment.

Shell

```
docker exec -it edgenode bash
```

4. You are now "inside" the Edge Node container's root shell. You can leave this terminal window open for the next steps.

⚙️ Part 4: Install Always-On Nodes Using wdocker CLI (Recommended for Convenience & Simplicity)*

wdocker is a specialized CLI tool that automates the process of joining Moss groups and keeps all tools within that group updated automatically.

*Note: Moss Tool installation can also be done manually using a hApp install config file and CLI following steps in the appendix below.
** ⚠️ wdocker is not supported on Windows

1. Install Node.js

You will need to install Node.js.

First update the Package Manager cache using:

Shell

```
apk update
```

Then, install Node.js

```shell
apk add nodejs npm
```

2. Install wdocker

Then install the wdocker CLI and its dependencies globally:

```shell
npm install -g @theweave/wdocker
npm install -g @theweave/utils
npm install -g @theweave/group-client
npm install -g @theweave/moss-types
```

3. Create a Conductor

Initialize a new conductor. You will be prompted to set a password. NOTE: If you are attempting this command where you cannot open a separate terminal window for whatever reason (e.g. HoloPort connected to monitor and keyboard) append '&' to the end of the following command to automatically background the command so you can still access the prompt.

```shell
wdocker run my-moss-node
```

4. Join a Group

Open a **separate terminal window**. Copy the invite link from your Moss Group (See Part 1 above) and run:

```shell
# IMPORTANT: Use quotes around the invite link
wdocker join-group my-moss-node "your-invite-link-here"
wdocker join-group holo-community-moss-node
"https://theweave.social/wal?weave-0.15://invite/d2543bb4-b784-4a
c0-ae16-971e1b3a90c1&progenitor=uhCAkovIker1pDmpka8PiVWFWMGmSvNHj
zdi9KnTOsUJzrsGG71JI"
```

Note: If you'd like to support the Holo Community Moss Group (v0.15 Moss), the Joining Code URL is here: https://theweave.social/wal?weave-0.15://invite/d2543bb4-b784-4ac0-ae16-971e1b3a90c1&progenitor=uhCAkovIker1pDmpka8PiVWFWMGmSvNHjzdi9KnTOsUJzrsGG71JI

5. Automatic Management

That's it! The running conductor will check for new unjoined tools in the group every **5 minutes** and install them automatically.

Useful Commands:

- wdocker start [name]: Restart a stopped node.
- wdocker list-apps [name]: See what is currently running.
- wdocker status [name]: Check if your node is healthy.

Your Edge Node is now providing a stable backbone for your peer-to-peer applications.

Track EdgeNode Roadmap

Get in touch with us on Telegram, https://t.me/+8JV9ibBHBDpmOTg0

🧰 Your Moss App Toolkit: What Can You Install?

This is where it gets exciting. Once you have your Moss group, you can plug in a variety of tools. Here's a breakdown of the powerful, peer-to-peer apps available, with comparisons to the apps you already know.

📋 Project & Task Management

- **KanDo**
  - **Think: Trello**, **Asana**, or **monday.com**.
  - **What it does:** A real-time, feature-rich KanBan board for project tracking. It includes commenting, labels, categories, checklists, markdown support, and more.
- **Acorn**
  - **Think:** A hierarchical project manager like **Notion** (using toggles) or **ClickUp**.

    - **What it does:** Project management redefined for distributed, remote teams. It uses "State-of-Affairs Trees" to break down large projects into smaller, manageable parts.
- **Who's In?**
    - **Think: Doodle** or **Calendly**.
    - **What it does:** A tool for scheduling events and tracking agreements. Propose an event or task, and others can agree based on necessary roles being filled, helping ideas become reality.

💬 Communication & Collaboration

- **Vines**
    - **Think:** The core chat function of **Slack** or **Discord**.
    - **What it does:** A core capacity for group conversations and direct messages (DMs) within The Weave.
- **Rhymez**
    - **Think: Slack** (with threads) or **Discord**.
    - **What it does:** A chat app that supports channels, DMs, and threaded side-panel conversations.
- **Presence**
    - **Think: Zoom**, **Google Meet**, or **Microsoft Teams**.
    - **What it does:** Need to talk face-to-face? Presence provides peer-to-peer video calls with screen sharing.
- **Notebooks**
    - **Think:** A collaborative **Notion**, **HackMD**, or **Obsidian** doc.
    - **What it does:** A real-time, collaborative Markdown editor that includes version control. Perfect for documentation, meeting notes, or co-writing.

💡 Brainstorming & Decision Making

- **TalkingStickies**
    - **Think: Miro**, **Mural**, or **FigJam**.
    - **What it does:** The classic real-time stickies board. It's perfect for ideation, retrospectives, brain-storming, or simple note-taking.
- **Converge**
    - **Think:** A structured decision tool like **Loomio** or **1000minds**.
    - **What it does:** A group decision-making tool that helps shift the focus from *outcomes* to *criteria*, providing a clear structure for making choices together.
- **Gamez**
    - **Think:** A virtual sandbox like **Tabletop Simulator** or **Roll20**.
    - **What it does:** A freeform digital space where you can create and play board games together.

💾 Data & Utilities

- **DataTub**
  - **Think: Airtable** or **Smartsheet**.
  - **What it does:** An Airtable-like app for powerful, flexible data management that goes beyond a simple spreadsheet.
- **Spreadsheets**
  - **Think: Google Sheets** or **Excel Online**.
  - **What it does:** Sometimes you just need a spreadsheet. This app provides real-time, collaborative editing for spreadsheets.
- **Files**
  - **Think:** A simple **Dropbox**, **WeTransfer**, or **Google Drive** folder.
  - **What it does:** A utility for file sharing and sending within your group.

🧪 Experimental Tools

- **Carbon Farm Network App**
  - **Think:** A custom **supply chain management (SCM)** tool.
  - **What it does:** A highly specialized app for custom supply chain management built on the hREA framework, designed to plan, track, and facilitate transactions.

Appendix

How to Manually Install Always-On Nodes (without wDocker)

📱 Get Your hApp's Network Information

To have the Edge Node join your hApp's network, you need to find its unique "Network Seed." We'll use the **Moss** desktop application as an example. You'll do these steps on your **regular daily-use computer**, not the Edge Node machine. Note: Some of the particulars on where to find network seeds in the Moss UI may shift over time as the UI changes in different versions. As such the instructions below may not be in sync with those UI changes, however the general concept is still illustrative for the purposes of explaining how to manually set up always-on nodes with Edge Node. You may just need to dig in the settings to find a network seed. The recommended way at present to set up always-on nodes is to use docker as explained above.

1. **Install and Set Up Moss:**
    - Download and install Moss from **https://moss.social**.
        i. After downloading, check the Moss version. If there are any Moss updates you need to install them and ensure you have the most current version. This is important so your Moss app can be on the correct version of Holochain. You should be on Moss v0.15
    - Open Moss and create a new private group by clicking the + icon.
    - Add some tools (hApps) to your group, such as **Kando** or **Rhymez**.
        i. Click the “M” in the top left corner, then the four squares icon to choose which tools
        ii. Then go back to the Moss group and install the tools in the Unjoined Tools tab of the Moss Group home screen

2. **Find the Network Seed:**
    - In your Moss group, click the **Settings** link (gear icon).
    - Go to the **Group Tools** tab.
    - For the tool you want to make "always-on" (e.g., Kando), click the three-dots menu (`...`).
    - Click **Show Advanced Settings**.
    - The **Network Seed** will be a long string of letters and numbers. **Copy this value**—you'll need it in the next part.

      Note: If you’d like to support the Holo Community Moss Group, the Joining Code URL is here: https://theweave.social/wal?weave-0.15://invite/d2543bb4-b784-4ac0-ae16-971e1b3a90c1&progenitor=uhCAkovIker1pDmpka8PiVWFWMGmSvNHjzdi9KnTOsUJzrsGG71JI

🔗 Install the Always-On hApp in Your Edge Node

Let's go back to the terminal window connected to your Edge Node machine. It's time to tell it about your hApp.

1. **Create a Configuration File:** Inside the container's command line, run this command to generate a template configuration file:
   Bash

```
None

happ_config_file create
```

2. This creates a file named `example_happ_config.json`.
3. **Edit the Configuration File:** We'll use a basic text editor called `vi` to edit the file.
   Bash

```
None

vi example_happ_config.json
```

   - To start editing, press the `i` key on your keyboard (for "Insert" mode).
   - Use your arrow keys to move the cursor.
   - Update the `name`, `happUrl`, and `networkSeed` fields with the information for your hApp. Paste the Network Seed you copied from Moss. If you are using a HoloPort without SSH, you will need to type in manually.
4. Here's an example for the **Kando** hApp:
   JSON

```
None

{
  "app": {
```

```
    "name": "kando",
    "version": "0.14.0",
    "happUrl":
"https://github.com/holochain-apps/kando/releases/download/v0.14.
0/kando.happ",
    "modifiers": {
      "networkSeed": "WRITE_OR_PASTE_YOUR_NETWORK_SEED_HERE",
      "properties": ""
    }
  },
  "env": {
    "holochain": {
      "version": "",
      "flags": [""],
      "bootstrapUrl": "",
      "signalServerUrl": "",
      "stunServerUrls": [""]
    }
  }
}
```

    - Once you are done editing, press the `Esc` key to exit Insert mode.
    - Type `:wq` and press `Enter`. This will **w**rite (save) the file and **q**uit the editor.
5. **Install the hApp:** Now, use your newly created configuration file to install the hApp in your Edge Node.
   Bash

```
None

install_happ example_happ_config.json
```

6. That's it! Your Edge Node will now download the hApp and join its network using the seed you provided. It is now acting as an always-on peer for your group. You can repeat this process for every hApp in your Moss group.

⚙️ Advanced: Installing hApps from `.webhapp` Files (Rhymez Example)

After you've successfully installed HolOS, you may want to install a Holochain application (hApp). Sometimes, these are distributed as `.webhapp` files, which bundle the core `.happ` file with a web-based user interface.

The current installation process requires a direct path to the `.happ` file. The following is a **temporary workaround** that shows you how to extract the `.happ` file from a `.webhapp` file and install it locally. This process will be automated in a future release.

Here's how you can do it using the "Rhymez" hApp as an example:

1. **Download the `.webhapp` file:** Open the terminal on your HoloPort and use `wget` to download the file.

```
None

wget
https://codeberg.org/matthme/rhymez/releases/download/0.1.5/rhyme
z.webhapp
```

2. **Unpack the file:** Use the `hc` command-line tool to extract the contents into a new directory. Here, we'll create a directory named `rhymes`.

```
None

hc web-app unpack rhymez.webhapp -o rhymes
```

3. **Find the `.happ` file:** Navigate into the new directory and list its contents to find the `.happ` file.

```
None

cd rhymes
ls
```

You will see `rhymez.happ` listed among the files.

4. **Get the absolute file path:** While inside the `rhymes` directory, use the `realpath` command to get the full, absolute path to the `.happ` file. This is the exact location on your system's storage.

```
None

realpath rhymez.happ
```

   This will output something like `/home/nonroot/rhymes/rhymez.happ`. **Copy this entire path.**

5. **Edit your installation JSON file:** You'll need to modify the JSON file you use for installing hApps. Instead of providing a URL to the `.happ` file, you will provide the local file path you just copied.
   - Find the line in your JSON that specifies the app's location (e.g., a URL).
   - Replace the URL with the following format: `file://<your_absolute_path>`

6. For example, if the path from `realpath` was `/home/nonroot/rhymes/rhymez.happ`, your JSON entry would look like this:

```
None

"happUrl": "file:///home/nonroot/rhymes/rhymez.happ",
```

   (Note: Using `file://` followed by the path is what's required. The triple slash `///` is the technically correct format for an absolute path from the root directory.)

7. **Install the hApp:** Now you can proceed with your standard installation command using the modified JSON file. The system will find the `.happ` file on your local storage and install it.

This workaround allows you to install any hApp distributed as a `.webhapp` package until the functionality is built directly into the installation tools.

Setting up SSH in HolOS for your Always-On Node

This guide is divided into three phases:

1. **Preparation:** Creating your keys and adding them to GitHub.
2. **Configuration:** Telling the HoloPort to grab those keys.
3. **Connection:** Logging in.

Phase 1: Get Your "Digital Keys" Ready

(Do this on your laptop first)

Think of this like a physical lock and key. You need to create a "Lock" (Public Key) and a "Key" (Private Key). You will give the "Lock" to GitHub so your HoloPort can install it, and you will keep the "Key" on your laptop to open the door.

Step A: Generate the Keys

1. **Open your Terminal:**
    - **Windows:** Press the `Windows Key`, type `PowerShell`, and press Enter.
    - **Mac:** Press `Command + Space`, type `Terminal`, and press Enter.
    - **Linux:** Open your preferred terminal emulator.
2. **Type the following command** and press **Enter**:
   Bash

```
None

ssh-keygen -t ed25519 -C "your_email@example.com"
```

   (Replace "your_email@example.com" with your actual email).

3. **Just press Enter** for all the questions it asks (file location, passphrase). You don't need a passphrase for this setup, so you can leave it empty.
4. Your keys are now created!

Step B: Copy Your "Public Key" (The Lock)

You need to see the code you just generated so you can copy it.

- **For Windows:** Type this command and press Enter:
  PowerShell

```
None

type $env:USERPROFILE\.ssh\id_ed25519.pub
```

- **For Mac/Linux:** Type this command and press Enter:
  Bash

```
None

cat ~/.ssh/id_ed25519.pub
```

Action: You will see a long string of text starting with `ssh-ed25519`. **Highlight and Copy** that entire line of text.

Step C: Give the Key to GitHub

1. Log in to your GitHub account.
2. Click your profile photo in the top-right corner and select **Settings**.
3. On the left sidebar, look for **SSH and GPG keys**.
4. Click the green button **New SSH key**.
   - **Title:** Name it something like "My Laptop".
   - **Key:** Paste the long text you copied in Step B.
5. Click **Add SSH key**.

Phase 2: Configure the HoloPort

(Switch to your HoloPort)

Now that GitHub has your key, we need to tell the Always-On node machine running HolOS to fetch it.

1. **Plug in:** Attach a USB keyboard and a monitor to your HoloPort.
2. **Turn it on:** Power up the device.

3. **Catch the Boot Menu:** Watch the screen. As soon as you see a list of options (the GRUB menu), quickly press the **`e`** key on your keyboard.
    - *Tip: This stops the boot process and lets you change settings.*
4. **Find the Linux line:** Use the arrow keys to move the cursor down. Look for the line that starts with the word `linux`.
5. **Edit the line:** Move your cursor to the very end of that `linux` line. Add a **space**, and then type:
   Plaintext

```
None

github_users=YourGitHubUsername
```

(Replace `YourGitHubUsername` with your actual GitHub username, e.g., `github_users=holonaut`).

6. **Save and Boot:** Press **`F10`** on your keyboard.
    - The HoloPort will continue booting. Behind the scenes, it is now talking to GitHub and installing your "Lock" (Public Key).

Phase 3: Connect via SSH

(Back to your laptop)

1. **Find the IP Address:** Once the HoloPort finishes booting, look at the monitor connected to it. It should display an IP address (e.g., `192.168.1.45`). Write this down.
2. **Open Terminal:** Go back to the Terminal/PowerShell on your laptop.
3. **Connect:** Type the following command and press Enter:
   Bash

```
None

ssh root@<The_IP_Address>
```

(Example: `ssh root@192.168.1.45`)

4. **Accept Fingerprint:** If it asks "Are you sure you want to continue connecting?", type `yes` and press Enter.

You should now be logged into your HoloPort!