USER MANUAL

GOSSHED

Your Fortress for SSH Connections

https://wahyuprimadi.com/products/gosshed.html

Gosshed User Manual

Your Fortress for SSH Connections.
Secure. Private. Uncompromising.

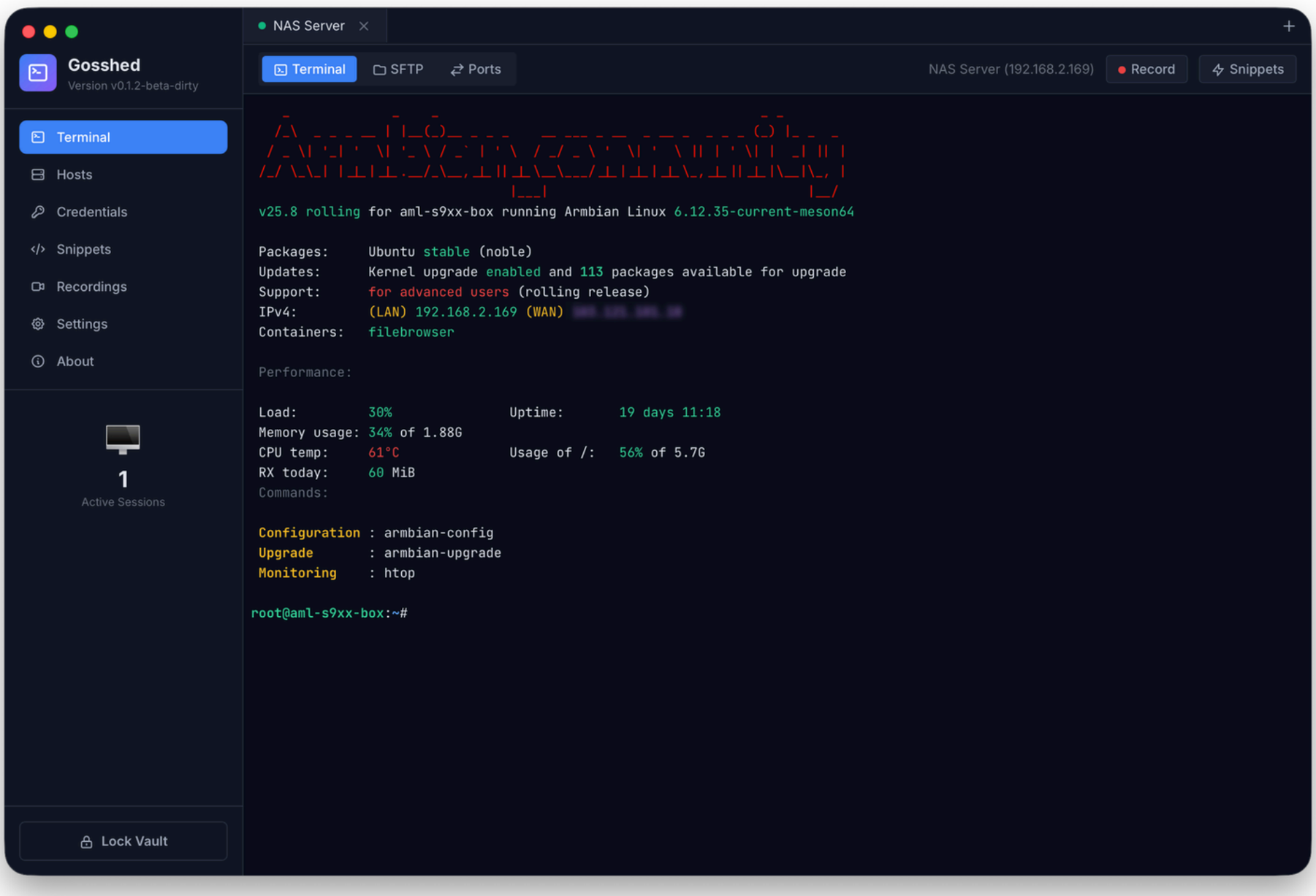
Introduction

Gosshed is a security-first SSH client designed for developers, system administrators, and security professionals who demand uncompromising protection for their server credentials and connections.

Unlike cloud-dependent alternatives, Gosshed operates entirely on your local machine. Your passwords, SSH keys, and connection details are encrypted with enterprise-grade cryptography and never leave your device. There are no accounts to create, no data synced to remote servers, and no telemetry collected.

Key Principles:

- **Zero-Knowledge Architecture**: Credentials are encrypted at rest and never exposed in plaintext on disk.
- **Local-First**: All data stays on your device. The application functions fully offline (except for SSH connections and optional update checks).
- **Enterprise-Grade Cryptography**: AES-256-GCM authenticated encryption, Argon2id key derivation, and XChaCha20-Poly1305 for vault exports.



Installation

macOS

- Download the **DMG** for your architecture (Apple Silicon, Intel, or Universal).
- Open the **DMG** and drag **Gosshed.app** into **/Applications**.

> **macOS Security Notice**
> The app is not signed with an Apple Developer certificate.
> If macOS blocks the app, run:

```
xattr -cr /Applications/Gosshed.app
```

Windows

1. Download the **.exe** that matches your system architecture (x64 or ARM64).
2. Run **gosshed.exe**.

Linux

- Download the binary/AppImage for your architecture.
- Make it executable:

```
chmod +x gosshed-linux-amd64
```

- Run:

```
./gosshed-linux-amd64
```

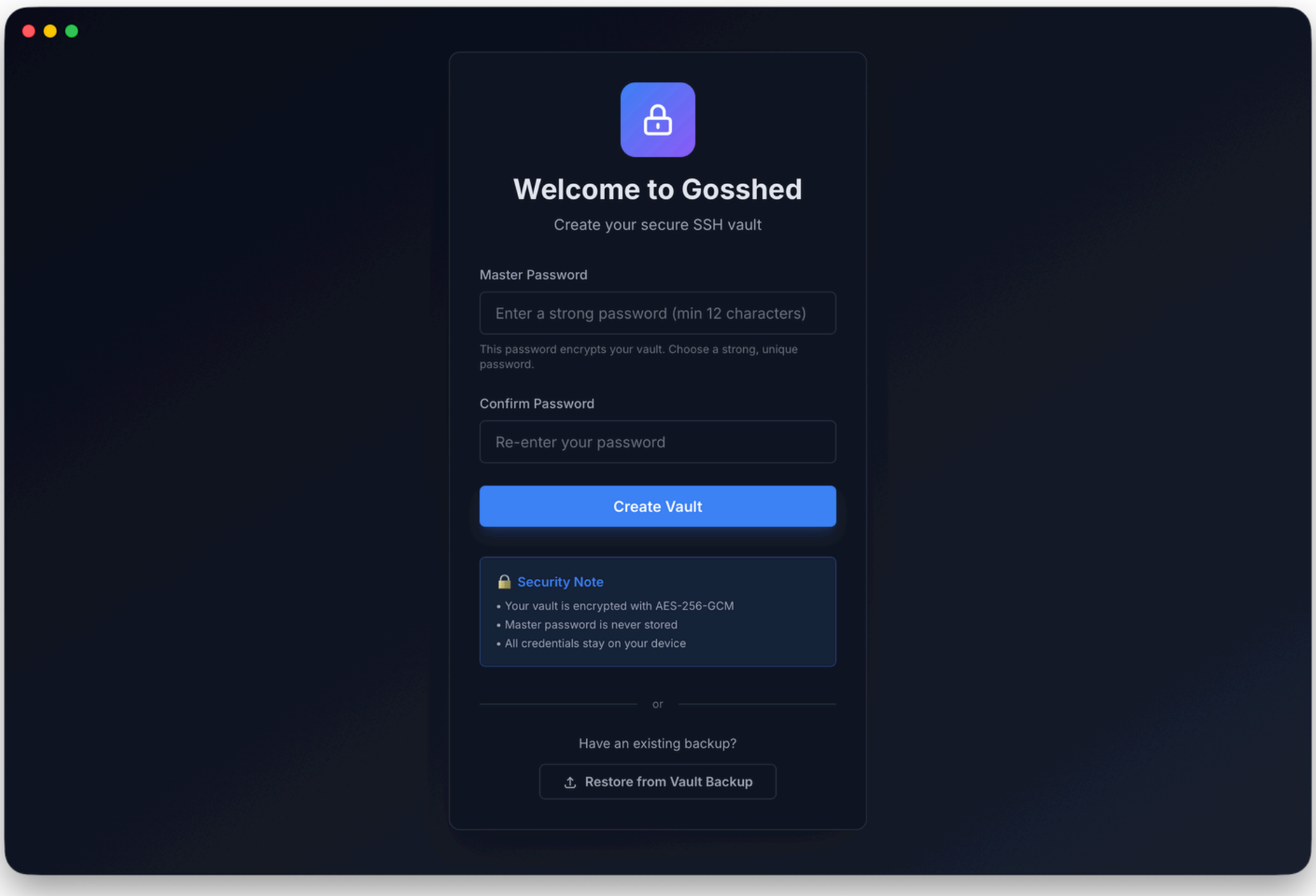
Getting Started

Creating a New Vault

When you launch Gosshed for the first time, the Vault Setup screen appears. The vault is the encrypted container where all your hosts, credentials, snippets, and recordings are stored.

- Enter a **master password** (minimum 12 characters). Choose a strong, unique password — this is the sole key to all your data.
- Confirm your password by entering it again.
- Click **Create Vault**.

The vault is created, encrypted, and automatically unlocked. You are taken directly to the main application.



Vault Setup: Create New Vault

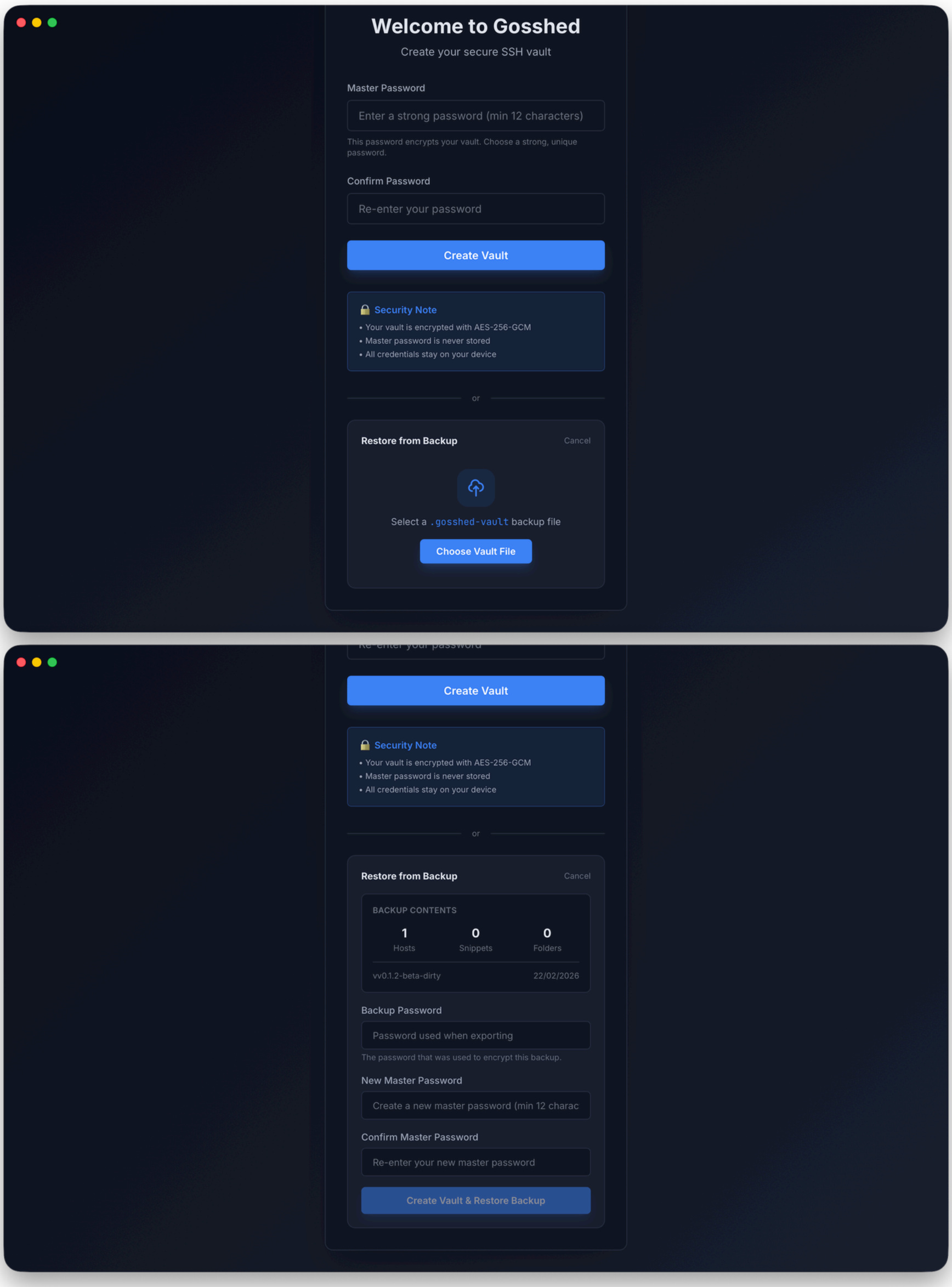
Important: Your master password is never stored anywhere, not on disk, not in memory after lock. If you forget it, there is no recovery mechanism. The only option is to purge the vault and start fresh, or restore from a backup if one exists.

Restoring from a Backup

If you have an existing **.gosshed-vault** backup file (from a previous installation or another machine), you can restore it during first launch:

- On the Vault Setup screen, click **Import Vault Backup** below the create form.
- Select the **.gosshed-vault** file using the native file dialog.
- A **preview card** appears showing the backup contents:
  - Number of hosts, snippets, and folders
  - Export date and application version
  - File size
- Enter the **export password** that was used when creating the backup.
- Enter a **new master password** for the vault on this machine.
- Confirm the new master password and click **Restore**.

All credentials from the backup are securely re-encrypted with your new master password. The original backup file remains unchanged.



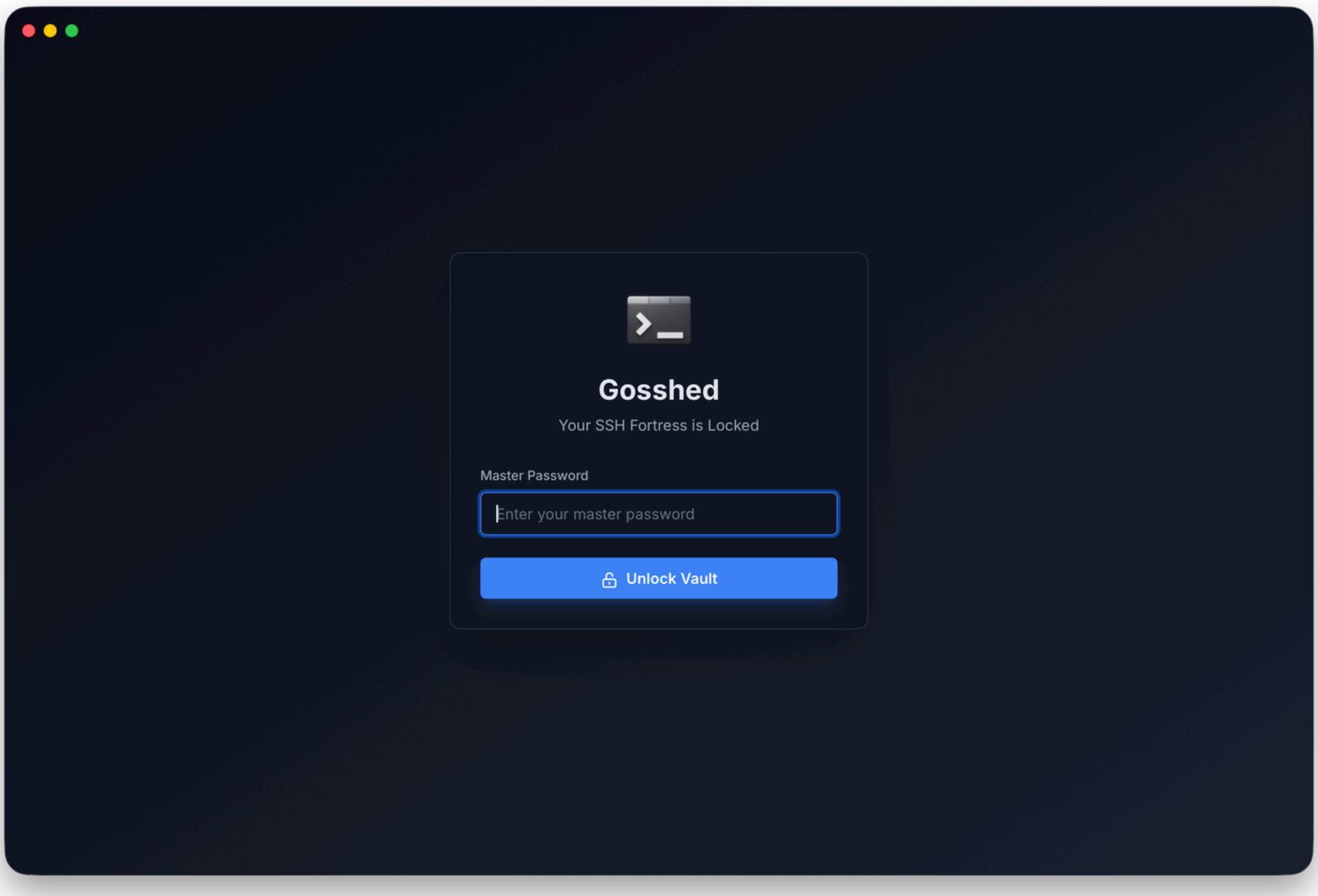
Vault Setup: Restore from Backup

Unlocking the Vault

On subsequent launches, the Vault Unlock screen appears:

- Enter your **master password**.
- Click **Unlock Vault** or press **Enter**.

After three consecutive failed attempts, a counter displays how many attempts have been made. This serves as a visual reminder — there is no lockout mechanism, as the vault is entirely local.



Vault Unlock Screen

Application Overview

The Gosshed interface is organized into a left sidebar for navigation and a main content area.

Sidebar Navigation

The sidebar contains icon-based navigation buttons for the following views:

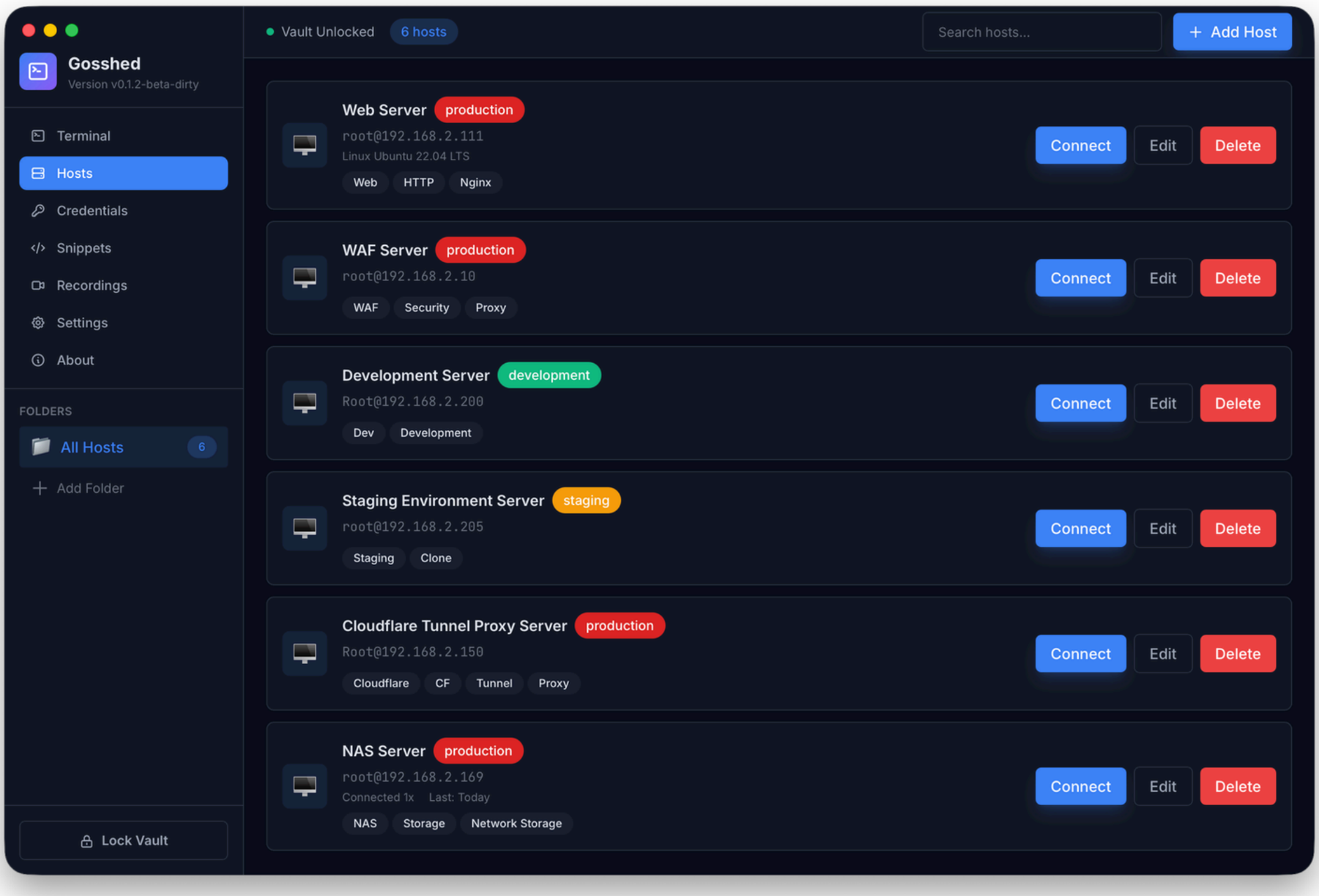
Menu	Description
Terminal	Active SSH sessions and tab management
Host	Server list, folders, and connection management
Credentials	SSH keys, passwords, and key generation
Snippets	Reusable command templates
Recordings	Session recording library and playback
Settings	Updates, vault backup/restore, and purge
About	Application info and credits

At the top of the sidebar:

- **Version number:** The currently running version of Gosshed.

At the bottom of the sidebar:

- **Lock Vault:** Immediately locks the vault, clearing all decrypted data from memory. You will need to re-enter your master password to continue.



Application Layout Overview

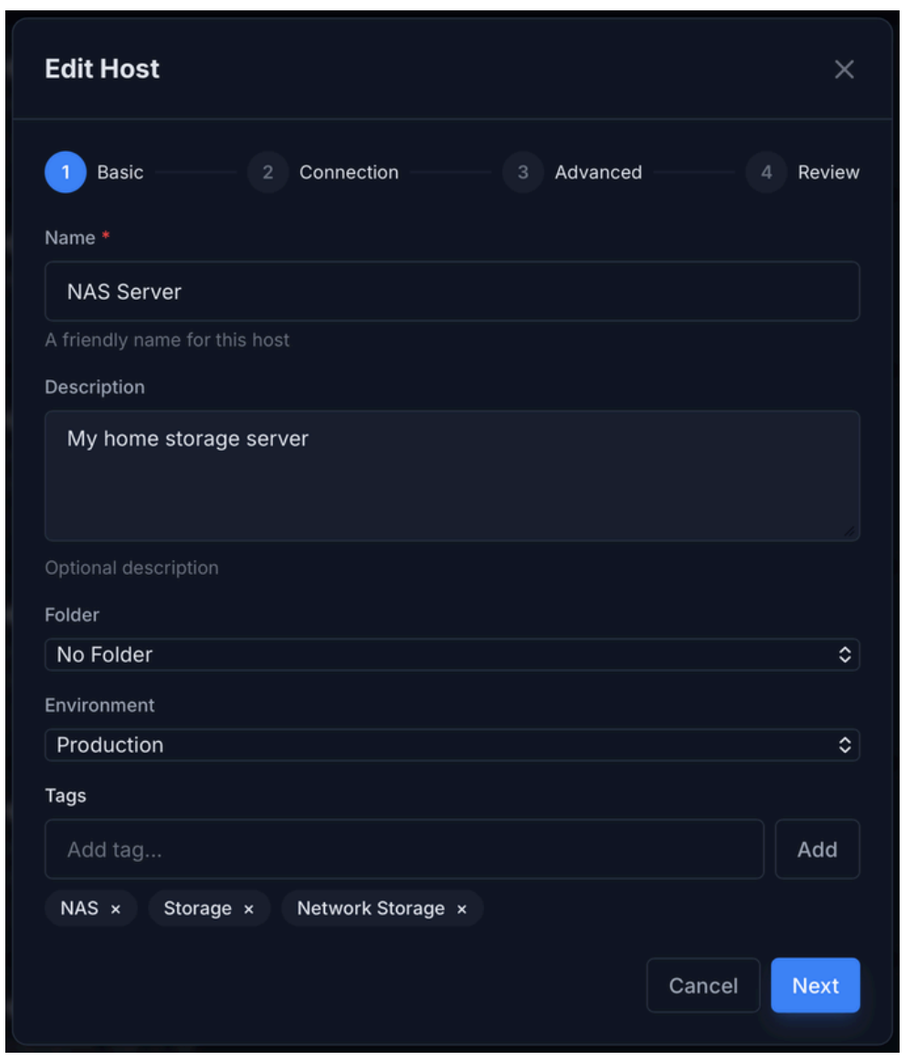
Host Management

Adding a Host

Hosts represent SSH servers you connect to. Adding a host is a guided 4-step process.

Step 1: Basic Information

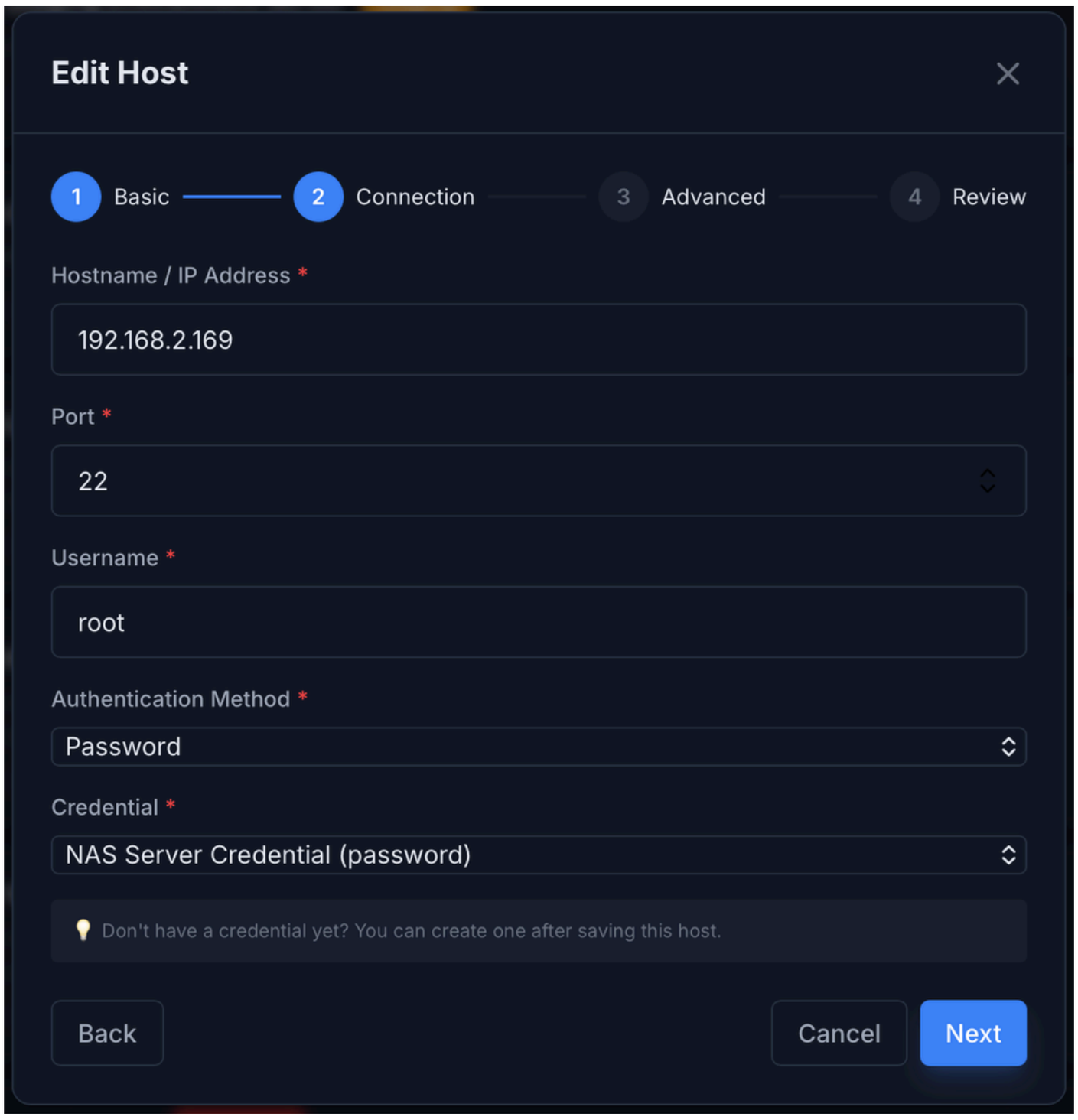
Field	Required	Description
Name	Yes	A display name for the host (e.g., "Production Web Server")
Description	No	A brief note about the host's purpose
Folder	No	Assign to a folder for organization
Environment	No	Label as Production, Staging, Development, or Test
Tags	No	Add multiple tags for filtering (e.g., "nginx", "ubuntu")



[Add Host] Step 1: Basic Information

Step 2: Connection Details

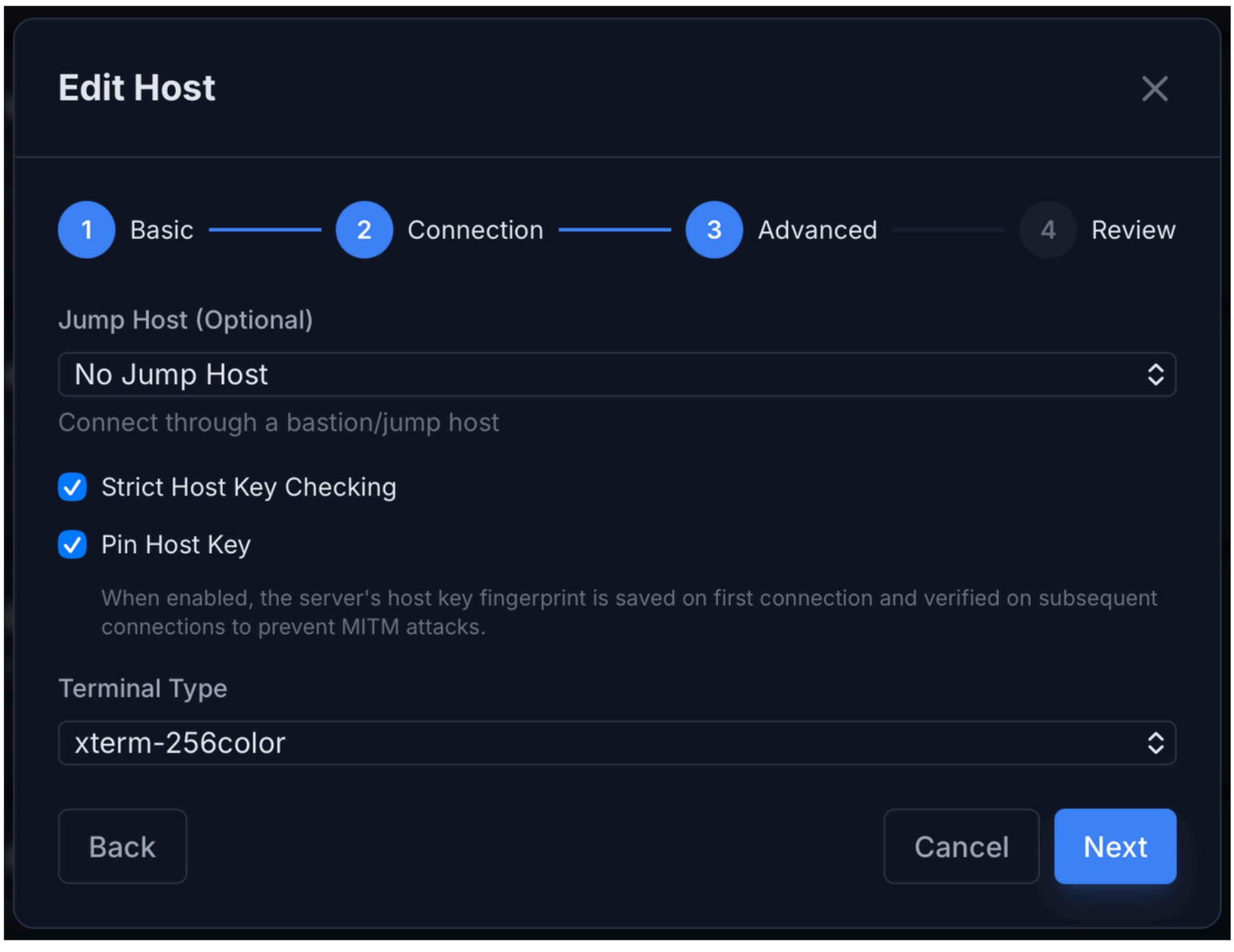
Field	Required	Default	Description
Hostname	Yes	-	IP address or domain name
Port	Yes	22	SSH port (1–65535)
Username	Yes	-	SSH login username
Authentication	Yes	Public Key	Method: Public Key, Password, Certificate, or SSH Agent
Credential	Yes	-	Select an existing credential from the vault



[Add Host] Step 2: Connection Details

Step 3: Advanced Settings

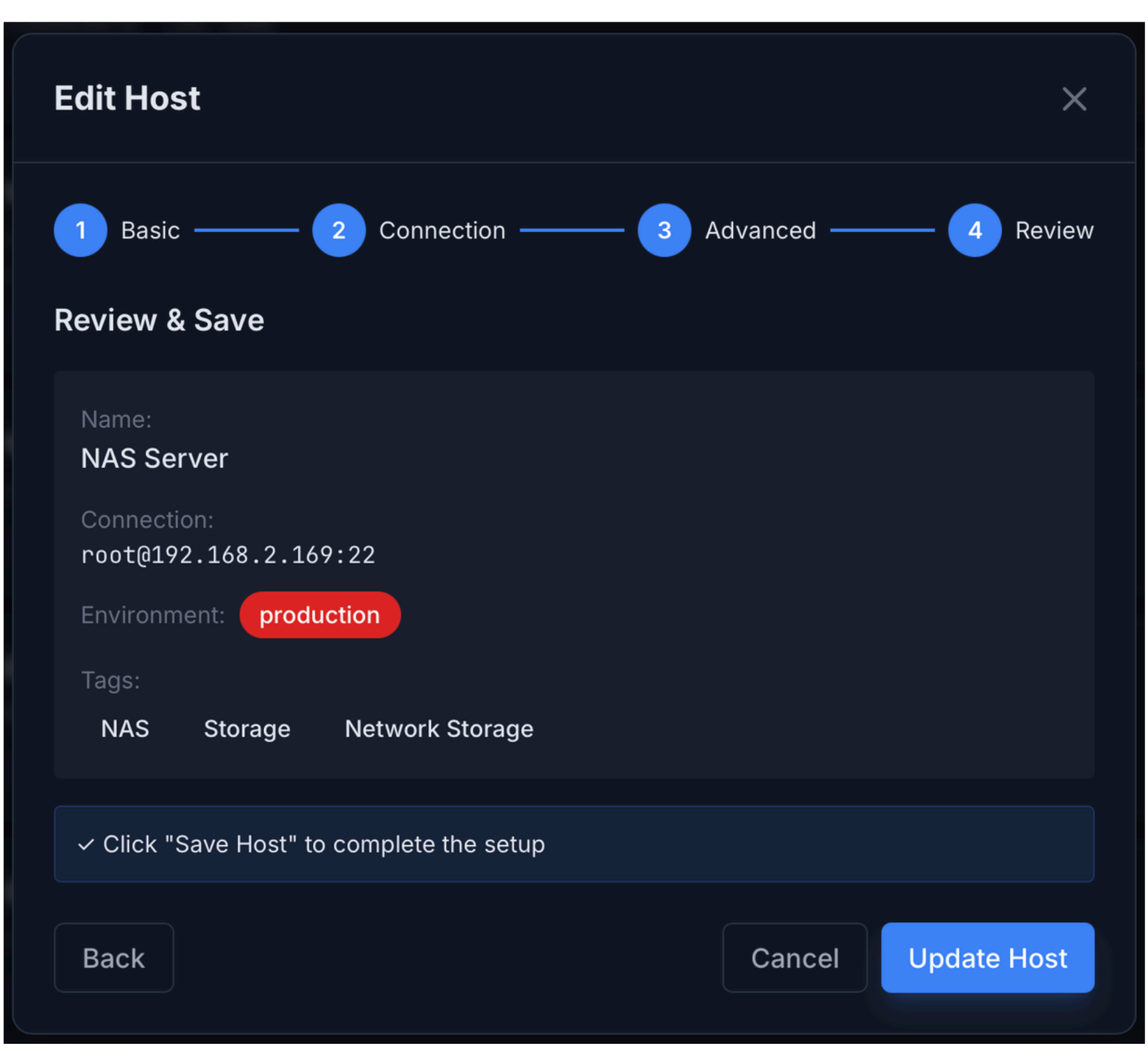
Field	Default	Description
Jump Host	None	Select a bastion/proxy host for multi-hop connections
Strict Host Key Checking	On	Verify the server's identity on every connection
Pin Host Key	On	Save and enforce the server's key fingerprint
Terminal Type	xterm-256color	Terminal identification string (xterm-256color, xterm, vt100)



[Add Host] Step 3: Advanced Settings

Step 4: Review

A summary of all configured settings. Review and click **Save** to add the host to your vault.

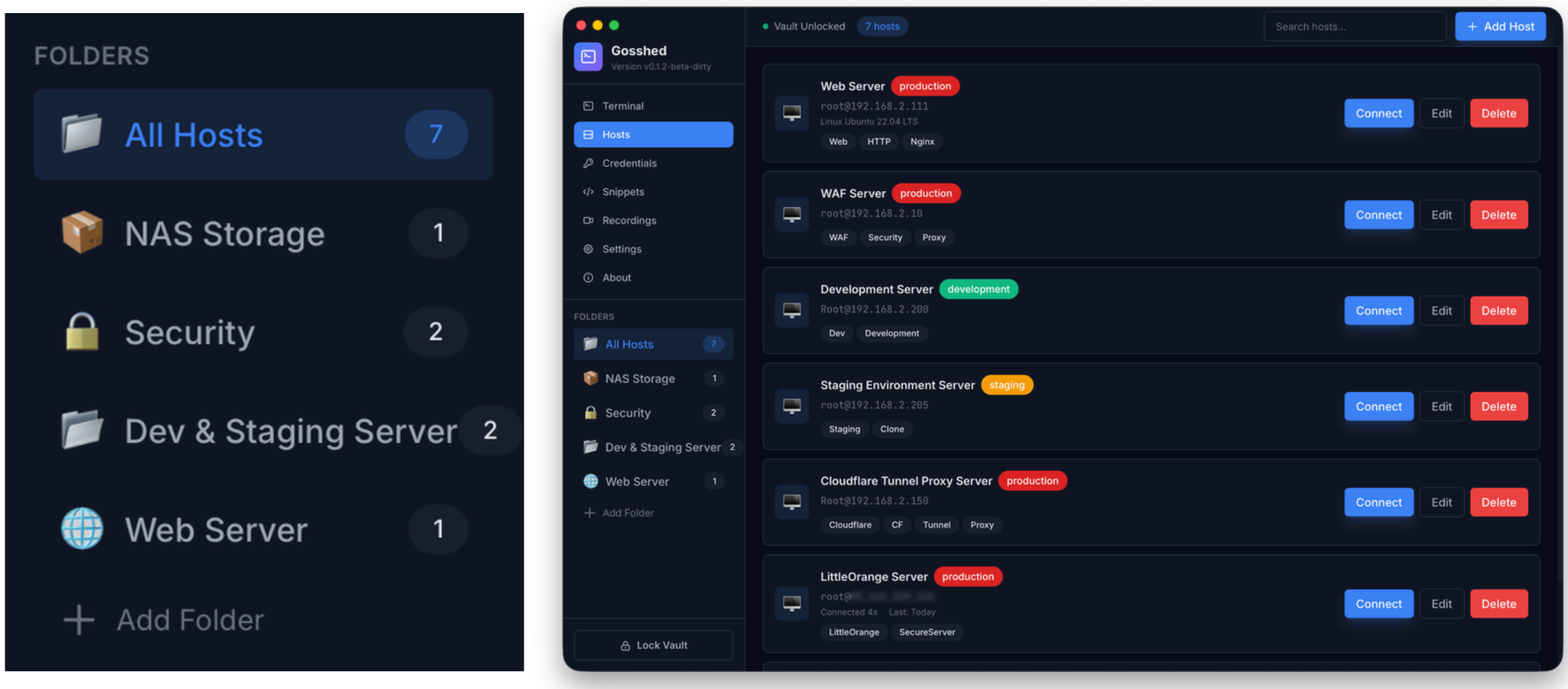


[Add Host] Step 4: Review

Folders and Organization

Hosts can be organized into folders for easier navigation.

- **Create a folder:** Click the **+** button in the sidebar when the Hosts view is active.
- **Assign hosts to folder:** Select a folder during host creation or editing.
- **Folder tree:** The sidebar displays a hierarchical folder tree with host counts per folder.
- **All Hosts:** Click this option to clear the folder filter and see all hosts.



Host Folders in Sidebar

Folders and Organization

Use the search bar at the top of the host list to filter hosts by:

- Host name
- Hostname or IP address
- Description
- Tags

Hosts can also be filtered by **environment** (Production, Staging, Development, Test) using the environment badge filters.

Credential Management

Storing Credentials

Navigate to the **Credentials** view to manage your authentication credentials. Gosshed supports the following credential types:

SSH Key

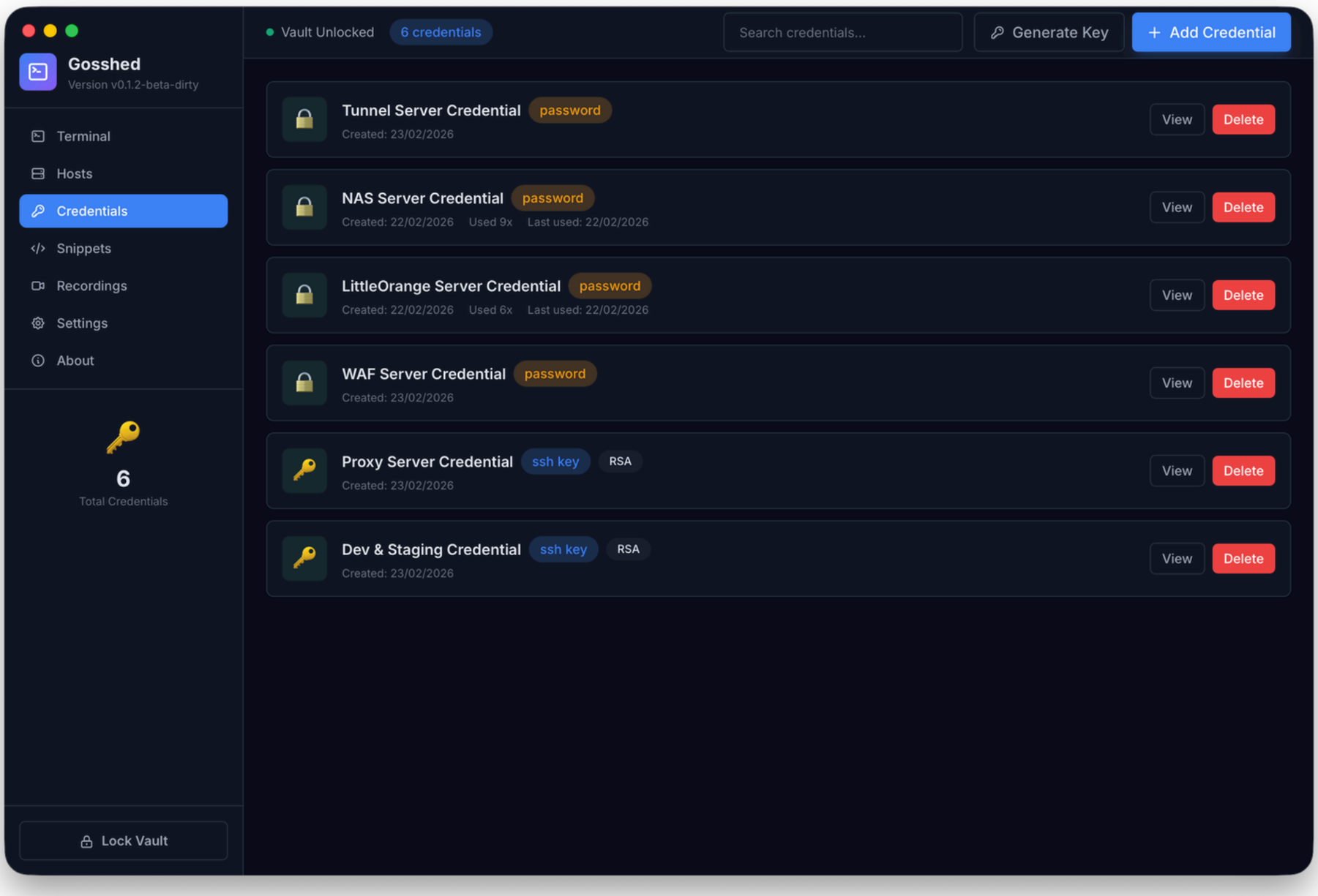
- Paste or upload a private key file (`.pem`, `.key`, or any format).
- Optionally provide a passphrase for encrypted keys.
- Optionally store the public key for reference.
- The key algorithm (ED25519, RSA, ECDSA) is auto-detected.

Password

- Store a username/password combination.
- The password is encrypted with the vault's master key.

Each credential includes:

- A descriptive **name** for identification.
- An optional **description**.
- **Usage tracking,** how many times the credential has been used and when it was last used.



Credential Management

SSH Key Generation

Gosshed can generate SSH key pairs directly within the application.

- Navigate to **Credentials** and click **Generate Key**.
- Select the key algorithm:

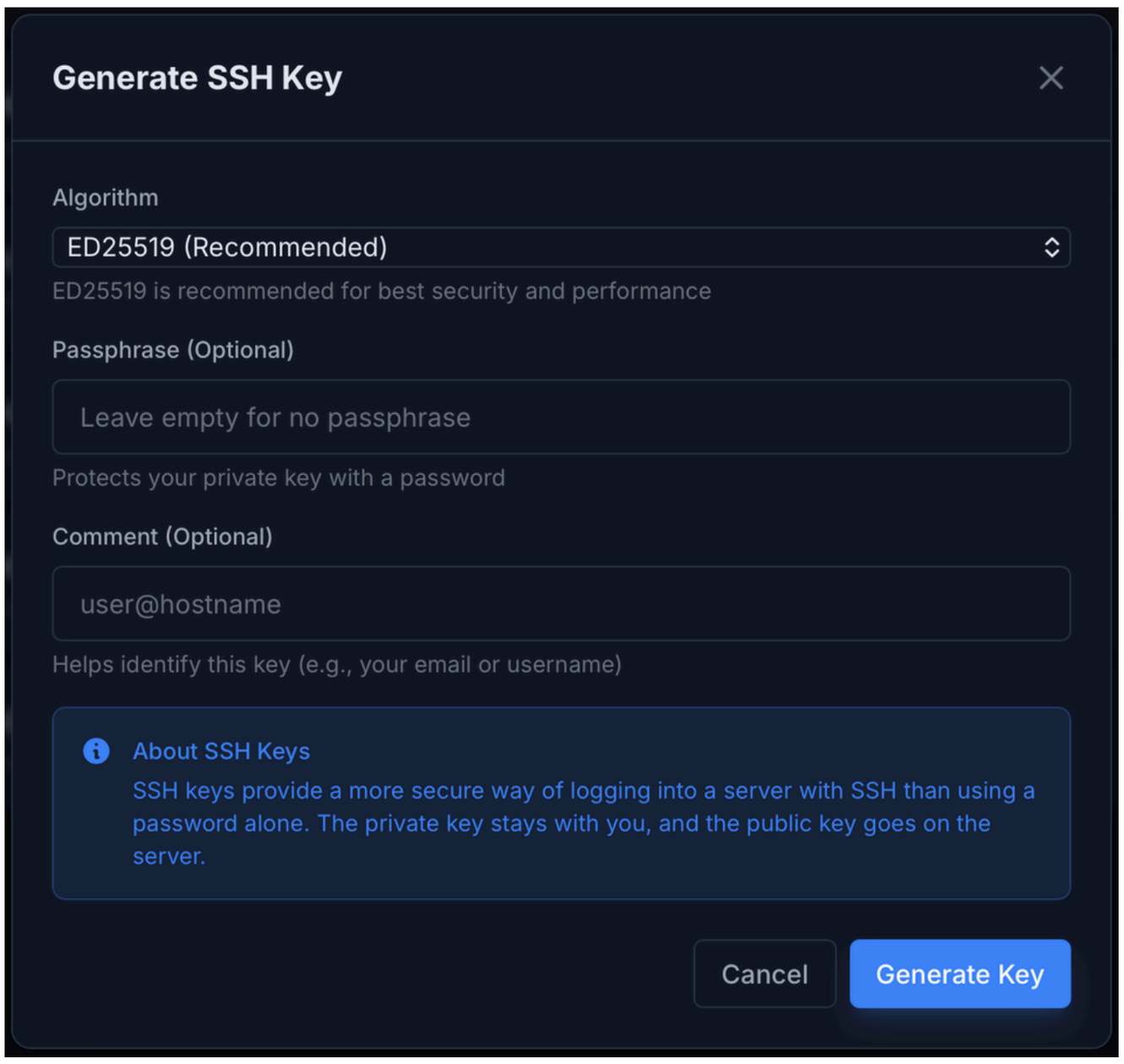
Algorithm	Description
ED25519	(Recommended) | Modern, fast, and secure. Smallest key size with strong security.
RSA 2048	Widely compatible. Suitable for legacy systems.
RSA 4096	(Recommended for RSA) | Stronger RSA variant.
ECDSA P-256	(Recommended for ECDSA) | Elliptic curve with good performance.
ECDSA P-384	Higher security ECDSA curve.
ECDSA P-521	Maximum security ECDSA curve.

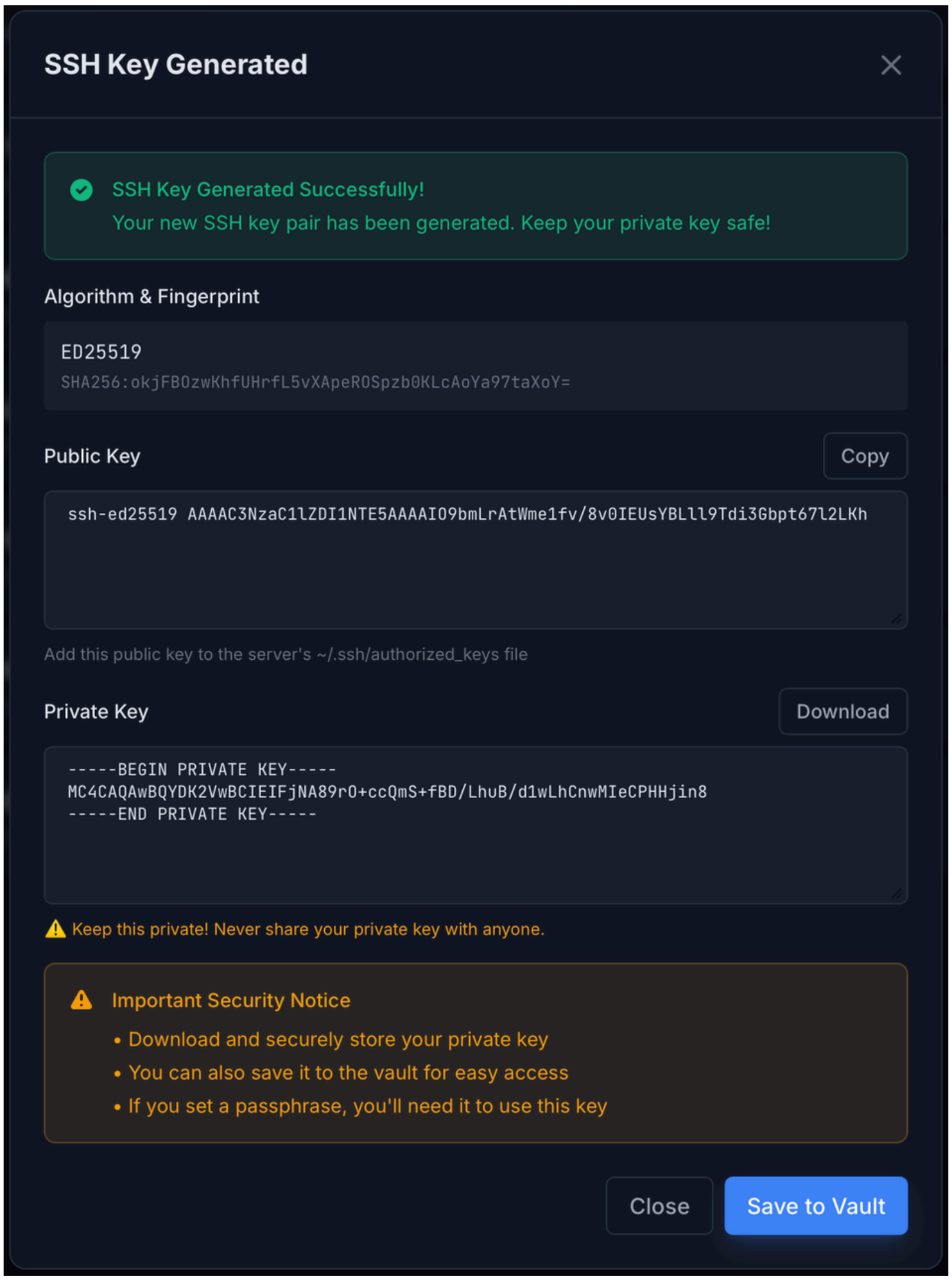
- Optionally set a **passphrase** to protect the private key.
- Optionally add a **comment** (e.g., `user@hostname`).
- Click **Generate**.

After generation, you can:

- **Copy the public key** to your clipboard (for adding to `~/.ssh/authorized_keys` on servers).
- **Download the private key** as a file.
- **Save to vault** to store the key pair encrypted in your vault.

The key **fingerprint** (SHA-256) is displayed for verification.





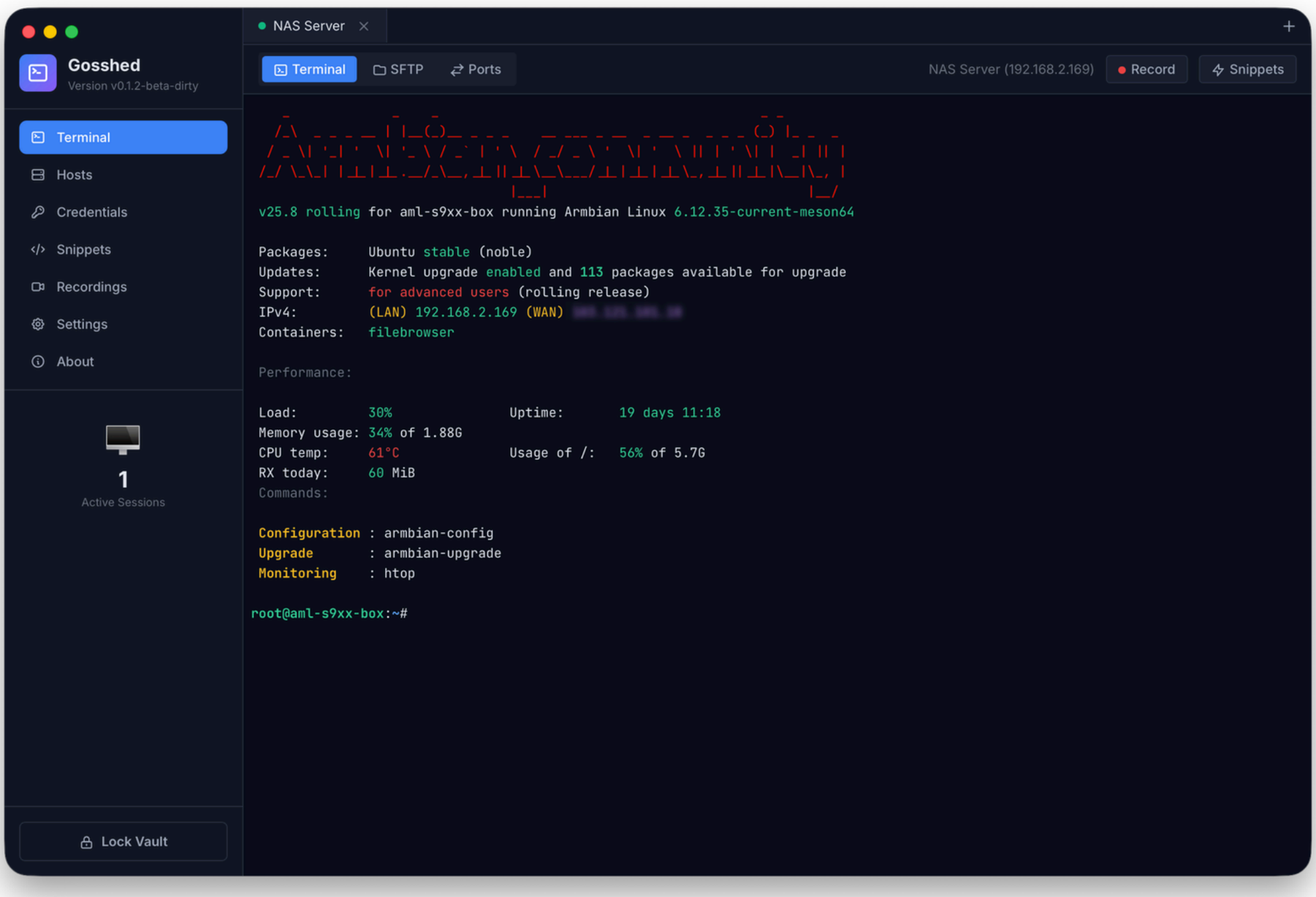
SSH Key Generation

Terminal Sessions

Connecting to a Host

- Navigate to the **Hosts** view.
- Click on a host in the list.
- If this is the first time connecting to this host, a **Host Key Verification** dialog appears (see Host Key Verification).
- A new terminal tab opens with an active SSH session.

The terminal provides full `**xterm-256color**` emulation, with a scrollback buffer of 10,000 lines. The font defaults to JetBrains Mono.

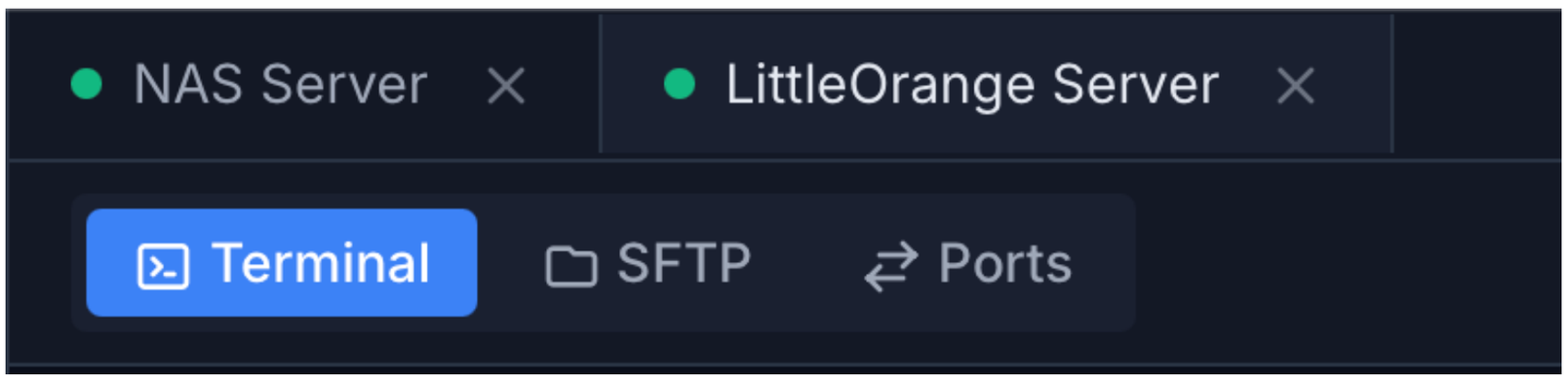


Active Terminal Session

Multi-Session Tabs

Gosshed supports multiple concurrent SSH sessions through a tab interface.

- Each connection opens in a **new tab**.
- Switch between sessions by clicking on the tab header.
- Close a session by clicking the **X** button on its tab.
- Each tab independently maintains its own terminal state, SFTP session, and port forwards.
- The tab header displays the host name and connection status.

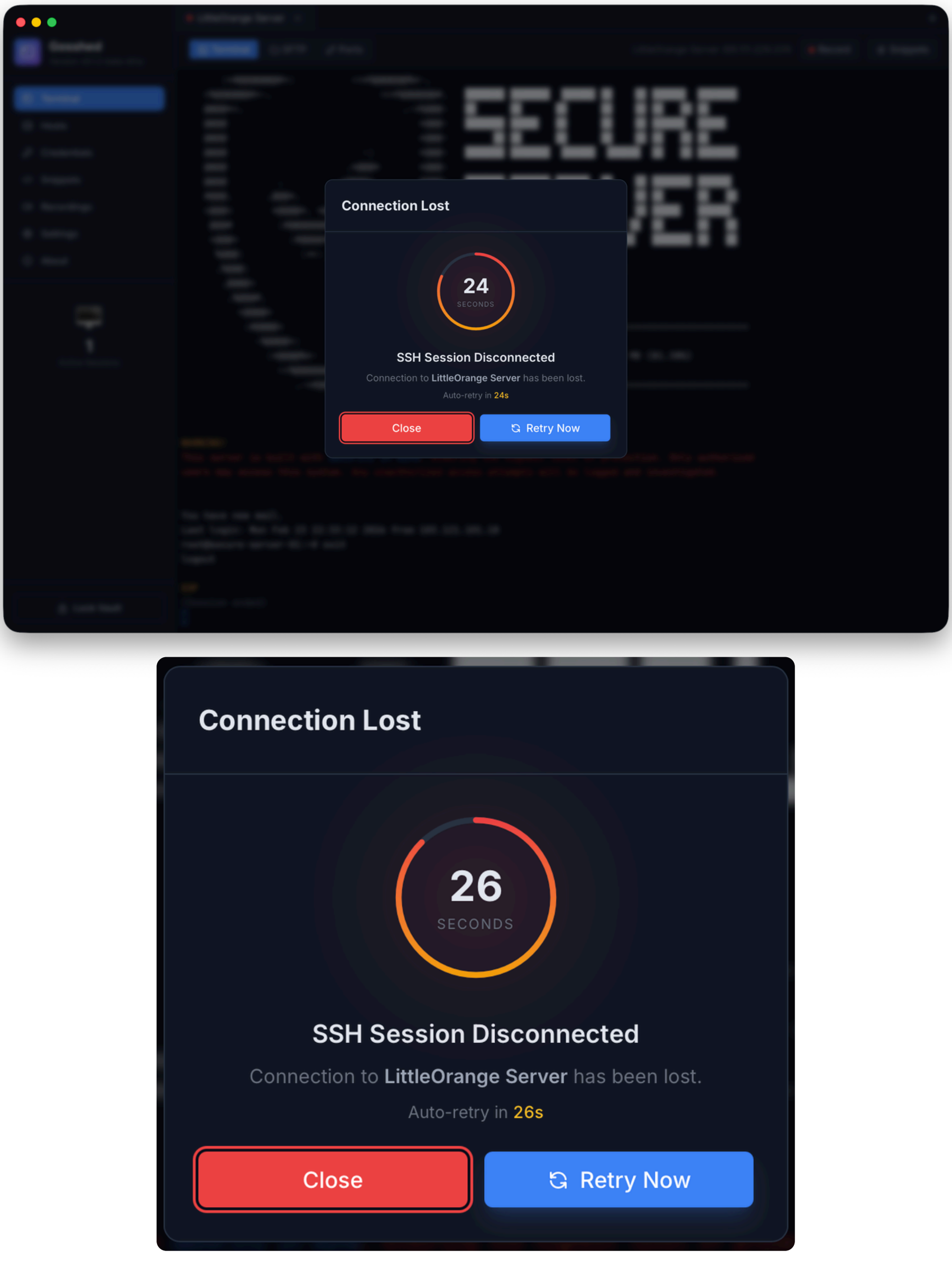


Multi-Session Tabs

Reconnection

When a connection drops unexpectedly, a **Reconnection Dialog** appears with:

- The error message explaining the disconnection.
- An automatic **countdown timer** for retry.
- A **Retry Now** button for immediate reconnection.
- A **Close** button to dismiss the session.



Reconnection Dialog

Host Key Verification

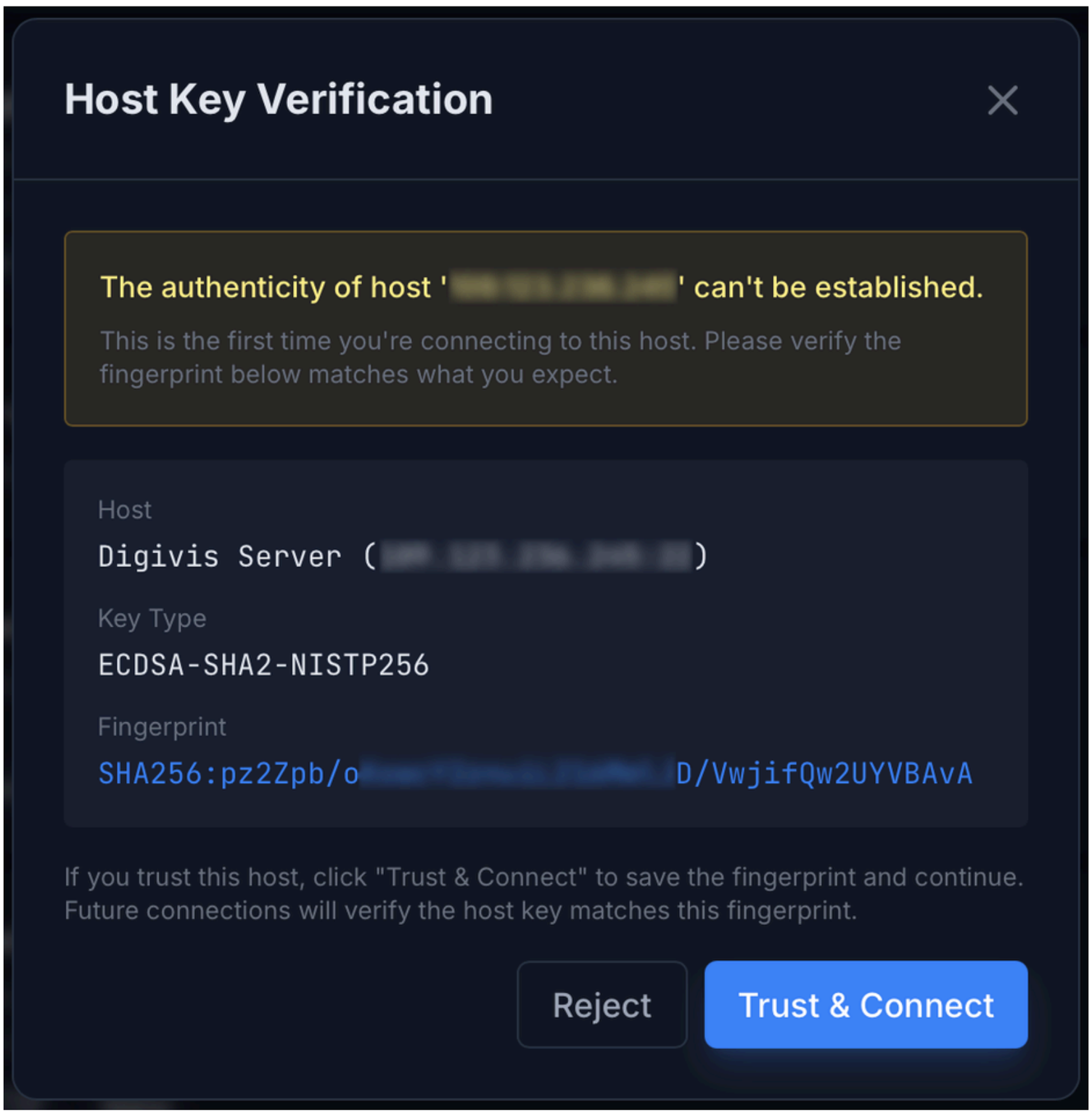
Gosshed implements **Trust On First Use (TOFU)** for host key verification to protect against man-in-the-middle attacks.

First Connection:
A dialog displays the server's key information:

- Key type (ED25519, RSA, ECDSA)
- SHA-256 fingerprint

You can **Accept** (trust and save) or **Reject** (cancel the connection). The fingerprint is stored in the vault after acceptance.

Subsequent Connections:
The saved fingerprint is compared against the server's current key. If they do not match, a **warning** is displayed indicating a possible MITM attack. The connection is blocked until you explicitly re-approve the new key.



Host Key Verification

SFTP File Manager

Dual-Panel Browser

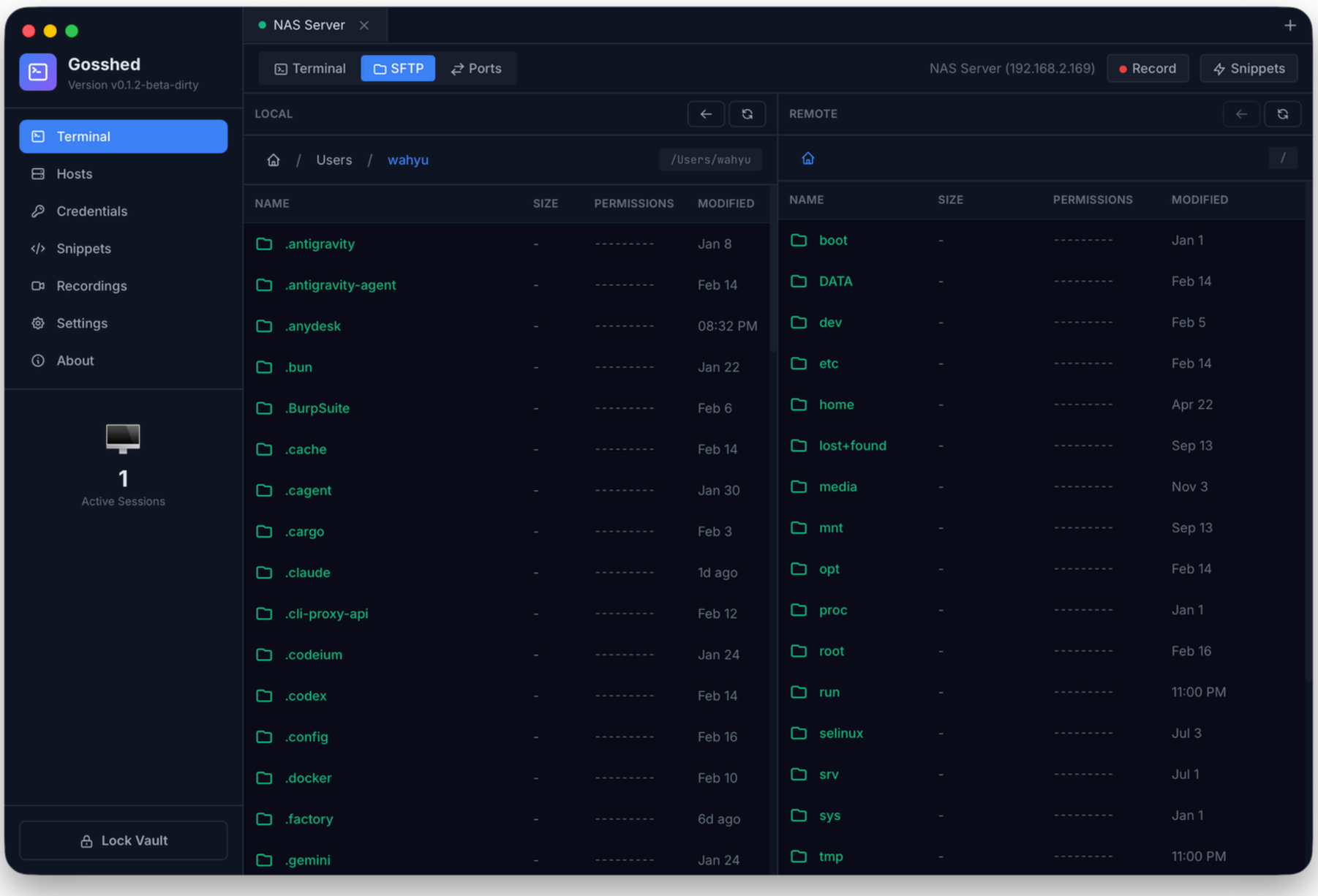
Each terminal session includes a built-in SFTP file manager. Switch to SFTP mode by clicking the **SFTP** button in the terminal toolbar.

The browser uses a dual-panel layout:

- **Left panel:** Your local filesystem.
- **Right panel:** The remote server's filesystem.

Both panels support:

- **Breadcrumb navigation:** Click any segment of the current path to jump to that directory.
- **Parent directory** (`..`)**:** Navigate up one level.
- **Directory-first sorting:** Folders appear before files.
- **File metadata:** Size (B, KB, MB), modification date, and permission string (e.g., `-rw-r--r--`).

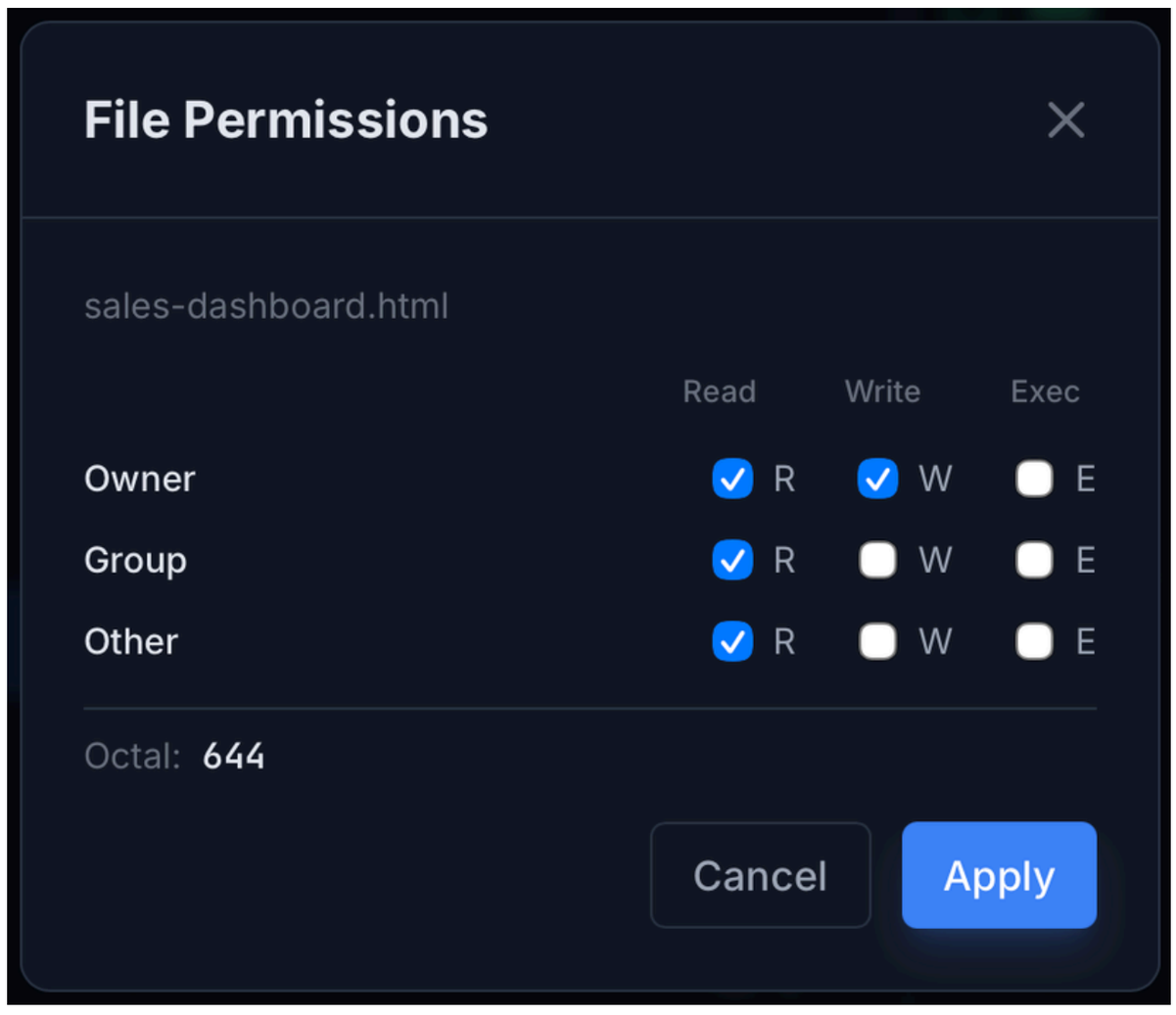


SFTP Dual-Panel Browser

File Operations

Operation	Description
Upload	Transfer files from the local panel to the remote panel
Download	Transfer files from the remote panel to the local panel
Create Directory	Create a new folder on either panel
Create New File	Create an empty file on the remote server
Rename	Rename a file or directory
Delete	Remove a file or directory (with confirmation)
File Permissions	Set chmod permissions on remote files (visual permission editor)
Copy URL to Clipboard	Copy the public or accessible URL of the selected remote file or folder to the clipboard

An **overwrite confirmation** dialog appears when a file with the same name already exists at the destination.



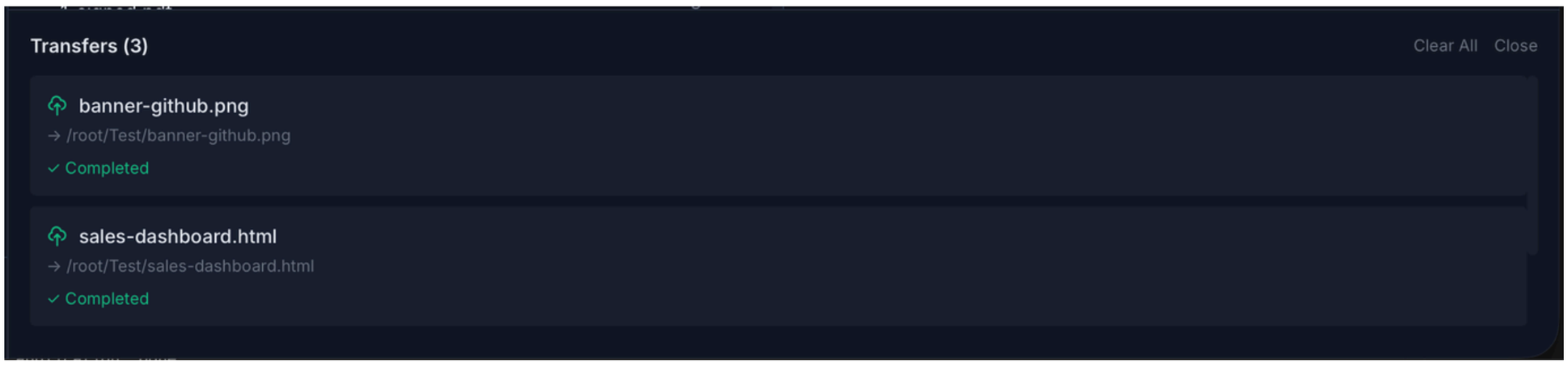
SFTP Permission Editor

Transfer Queue

Active file transfers are displayed in a transfer queue at the bottom of the SFTP panel:

- **Progress bar** with percentage completion.
- **Transfer speed** display.
- **File size** tracking.
- **Cancel** button for individual transfers.
- **Clear all** option to dismiss finished transfers.
- **Close** transfer panel.

Transfers show three states: **active** (in progress), **completed** (success), and **error** (failed with message).



SFTP Transfer Queue

Port Forwarding

Each terminal session supports SSH tunneling. Switch to port forwarding mode by clicking the **Ports** button in the terminal toolbar.

Transfer Queue

Local port forwarding (`-L`) makes a remote service accessible on your local machine.

Field	Description
Local Port	The port on your machine to listen on
Remote Host	The destination host (as seen from the SSH server)
Remote Port	The destination port

Example: Forward local port 3306 to a MySQL server accessible from the SSH host:

```
localhost:3306 → mysql.internal:3306
```

This allows you to connect to `localhost:3306` with a database client and reach the remote MySQL instance.

Remote Forward

Example: Expose a local development server on the remote machine:

```
remote:8080 → localhost:3000
```

Dynamic Forward (SOCKS5)

Dynamic port forwarding (`-D`) creates a SOCKS5 proxy on your local machine that routes all traffic through the SSH tunnel.

Field	Description
Local Port	The SOCKS5 proxy port on your machine

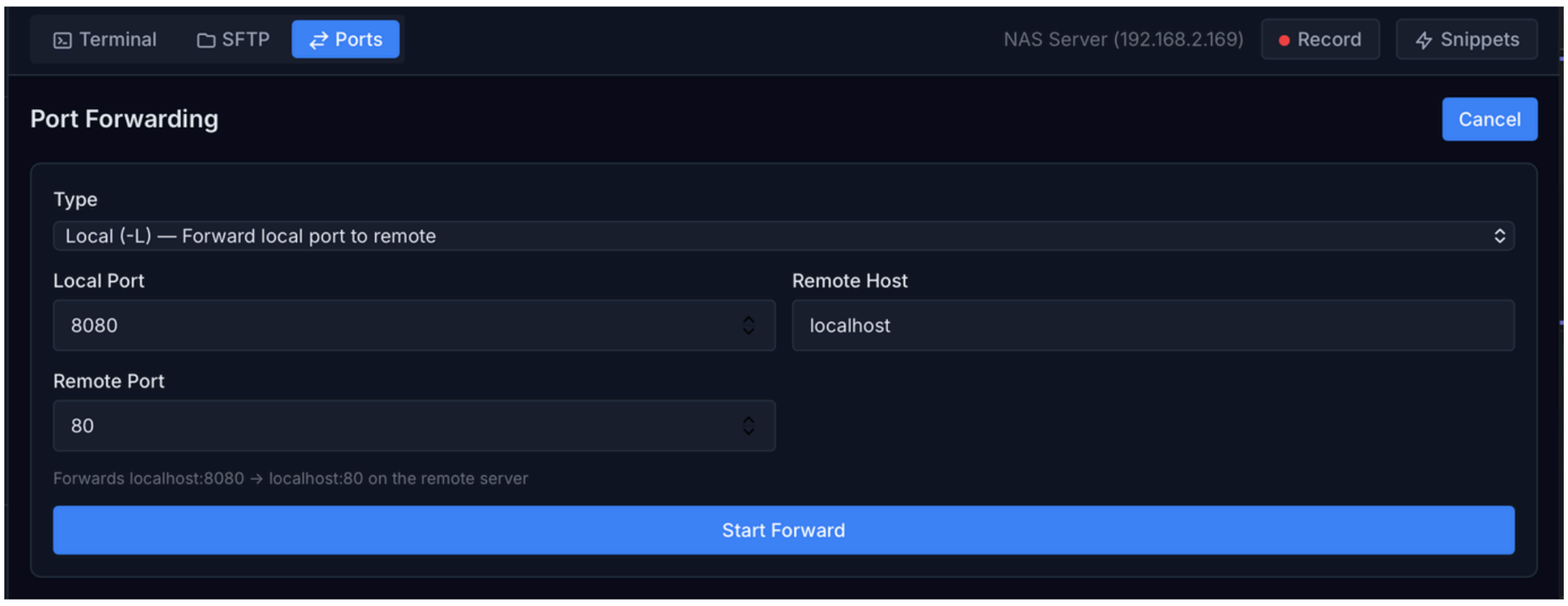
Example: Create a SOCKS5 proxy on port 1080:

```
SOCKS5 on :1080
```

Configure your browser or application to use `localhost:1080` as a SOCKS5 proxy to route traffic through the SSH server.

Managing Forwards

- Each active forward displays a **green status indicator** and a **Stop** button.
- Forward status is refreshed automatically every 3 seconds.
- Type-specific color badges distinguish Local (blue), Remote (purple), and Dynamic (amber) forwards.



Port Forwarding Panel

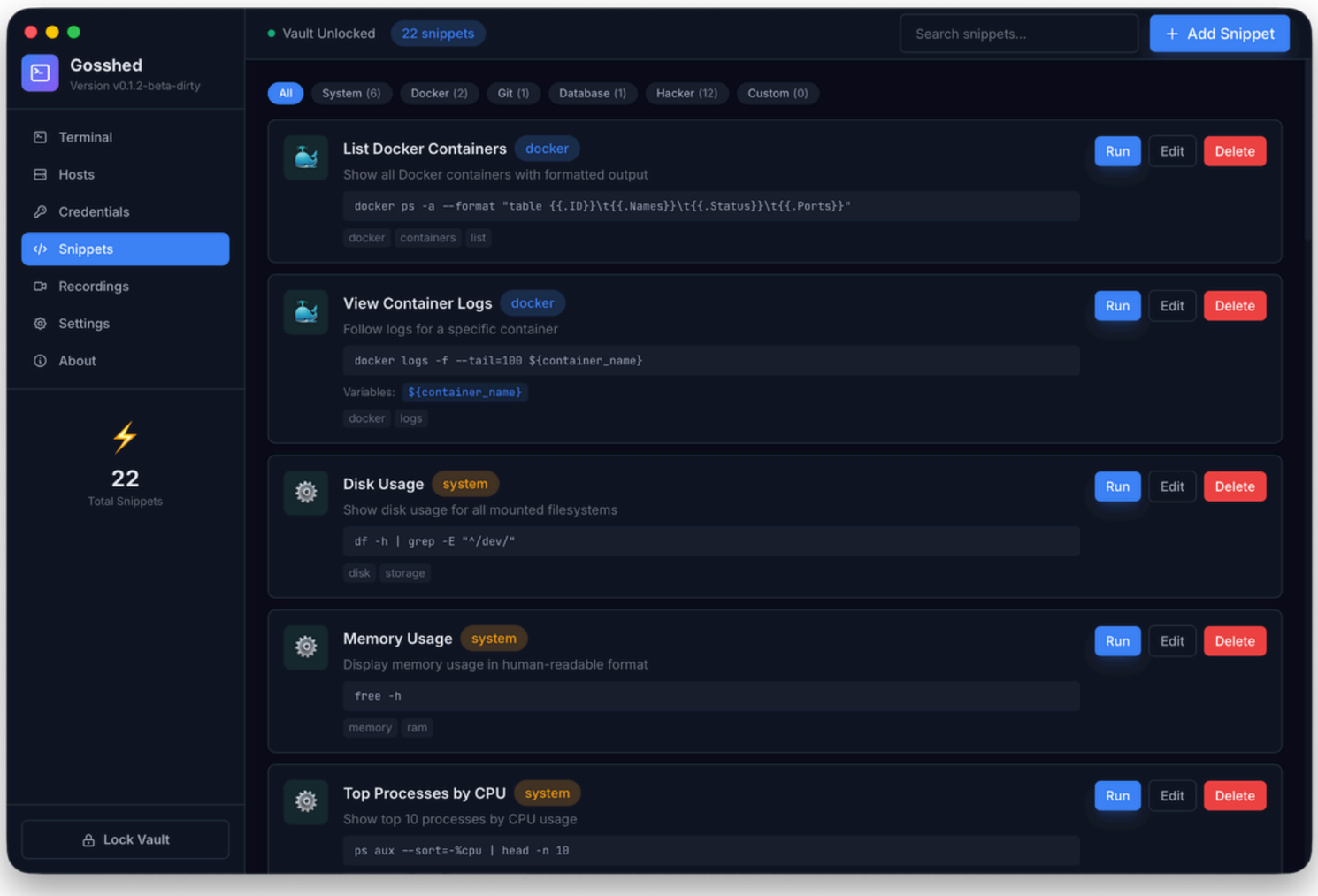
Command Snippets

Creating Snippets

Navigate to the **Snippets** view to manage reusable command templates.

Field	Required	Description
Name	**Yes**	A descriptive name (e.g., "Check Disk Usage")
Command	**Yes**	The shell command to execute
Description	**No**	A brief explanation of what the command does
Category	**No**	Categorize as System, Docker, Git, Database, Hacker/Red Team, or Custom
Tags	**No**	Add tags for filtering and search
Favorite	**No**	Mark as a favorite for quick access

Gosshed includes a library of **built-in snippets** covering common system administration tasks. These appear alongside your custom snippets.



Command Snippets

Variable Placeholders

Snippets support dynamic variable substitution using the `${variable_name}` syntax.

Example:

```
ssh-keygen -t ${algorithm} -b ${bits} -C "${comment}"
```

When executed, Gosshed prompts you to fill in each variable before running the command. A **variable count badge** on the snippet indicates how many placeholders it contains.

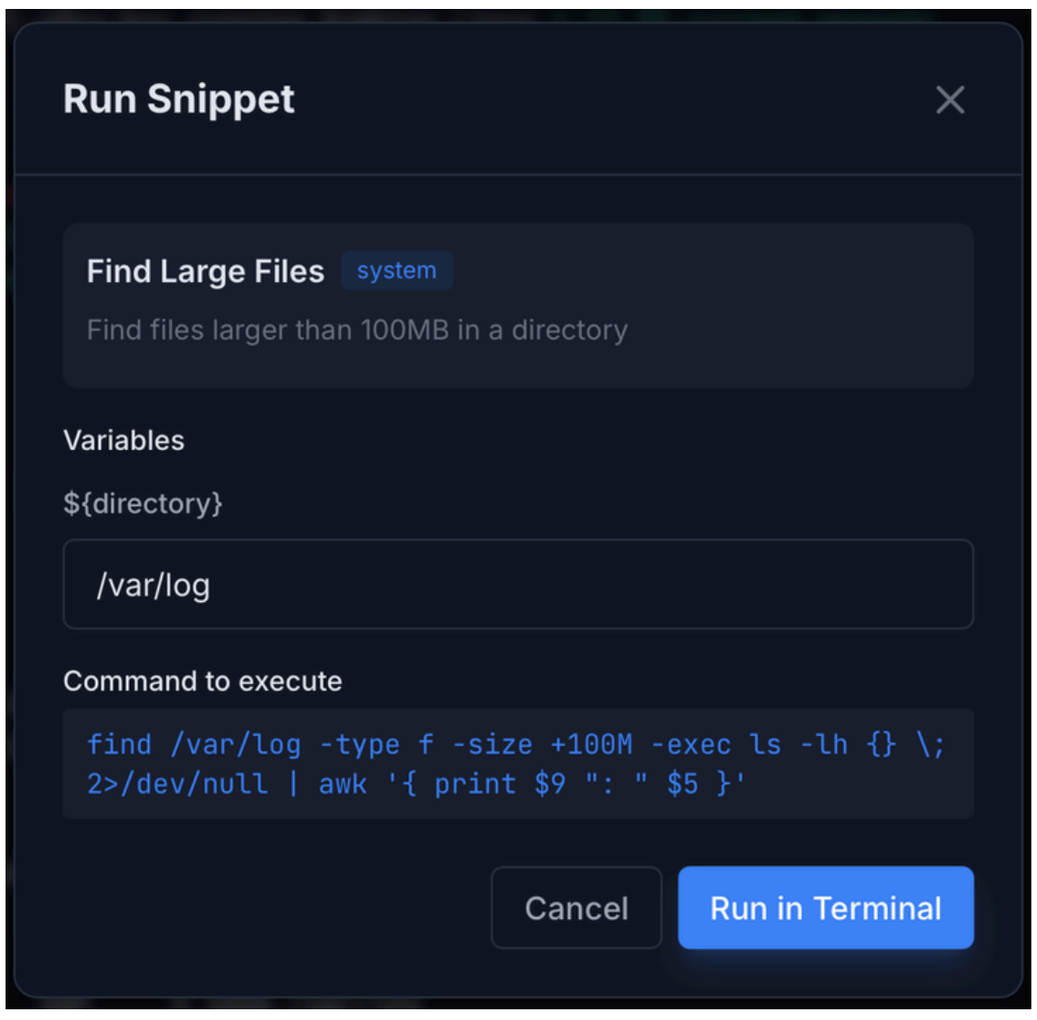
Executing Snippets

There are two ways to run a snippet:

1. **From the Snippets view:** Click a snippet and select the target session.
2. **From the Terminal view:** Click the **Snippet** button in the terminal toolbar, search for a snippet, and execute it.

- **Simple snippets** (no variables) execute immediately on the selected session.
- **Parameterized snippets** display a dialog prompting for each variable value before execution.

Usage statistics (run count and last used date) are tracked per snippet.



Snippet Execution with Variables

Session Recording and Playback

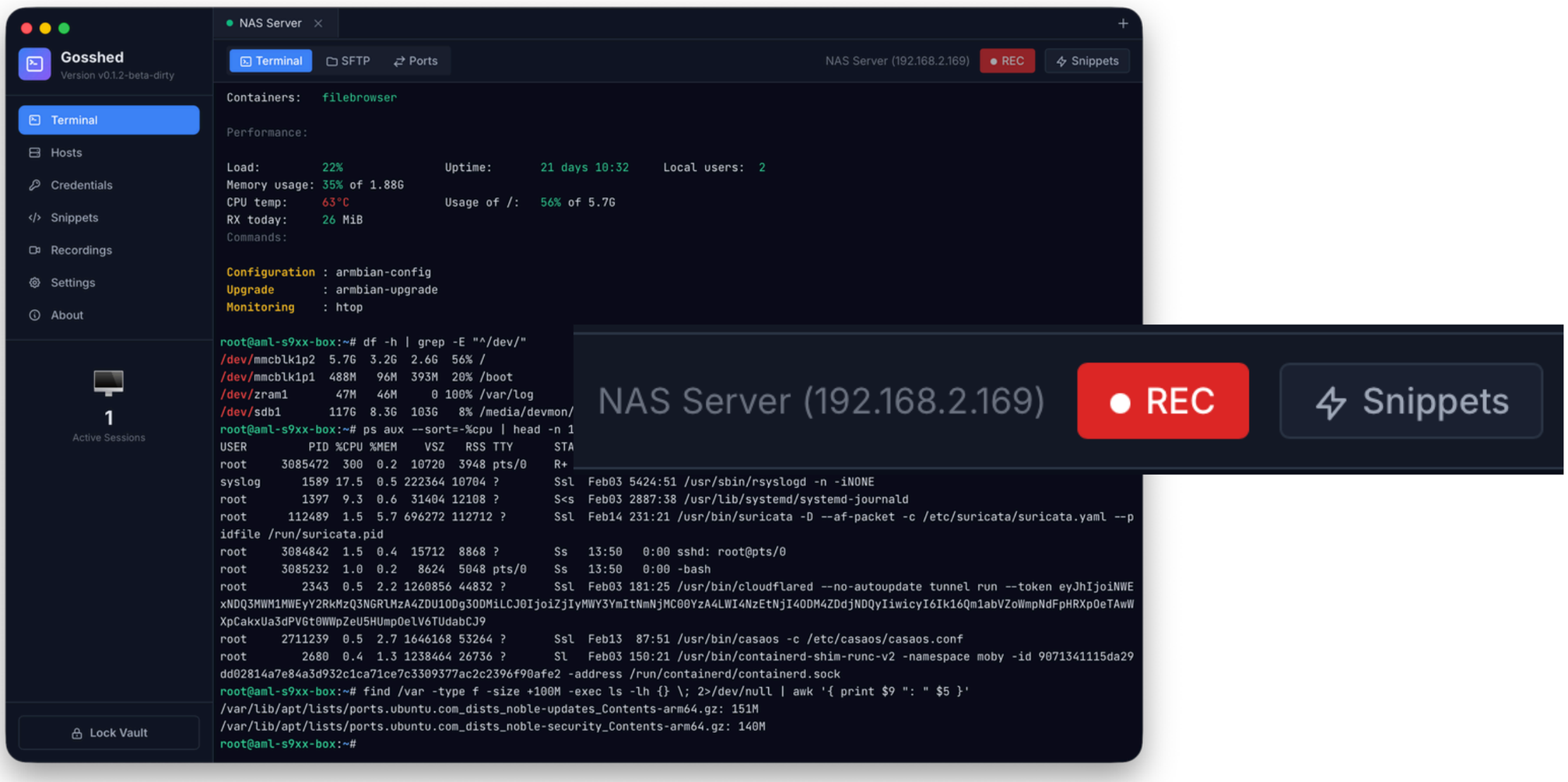
Recording a Session

1. Connect to a host and open a terminal session.
2. Click the **Record** button in the terminal toolbar.
3. The button changes to a pulsing red **REC** indicator, confirming the recording is active.
4. All terminal output is captured in real time.
5. Click the **REC** button again to stop and save the recording.

Recordings are stored in the **asciicast v2** format (`.cast` files), compatible with tools like **asciinema** (https://asciinema.org).

Each recording captures:

- Session ID and host name
- Start and end timestamps
- Duration (in seconds)
- Event count
- File size
- Terminal dimensions (columns x rows)



Active Recording Indicator

Playing Back Recordings

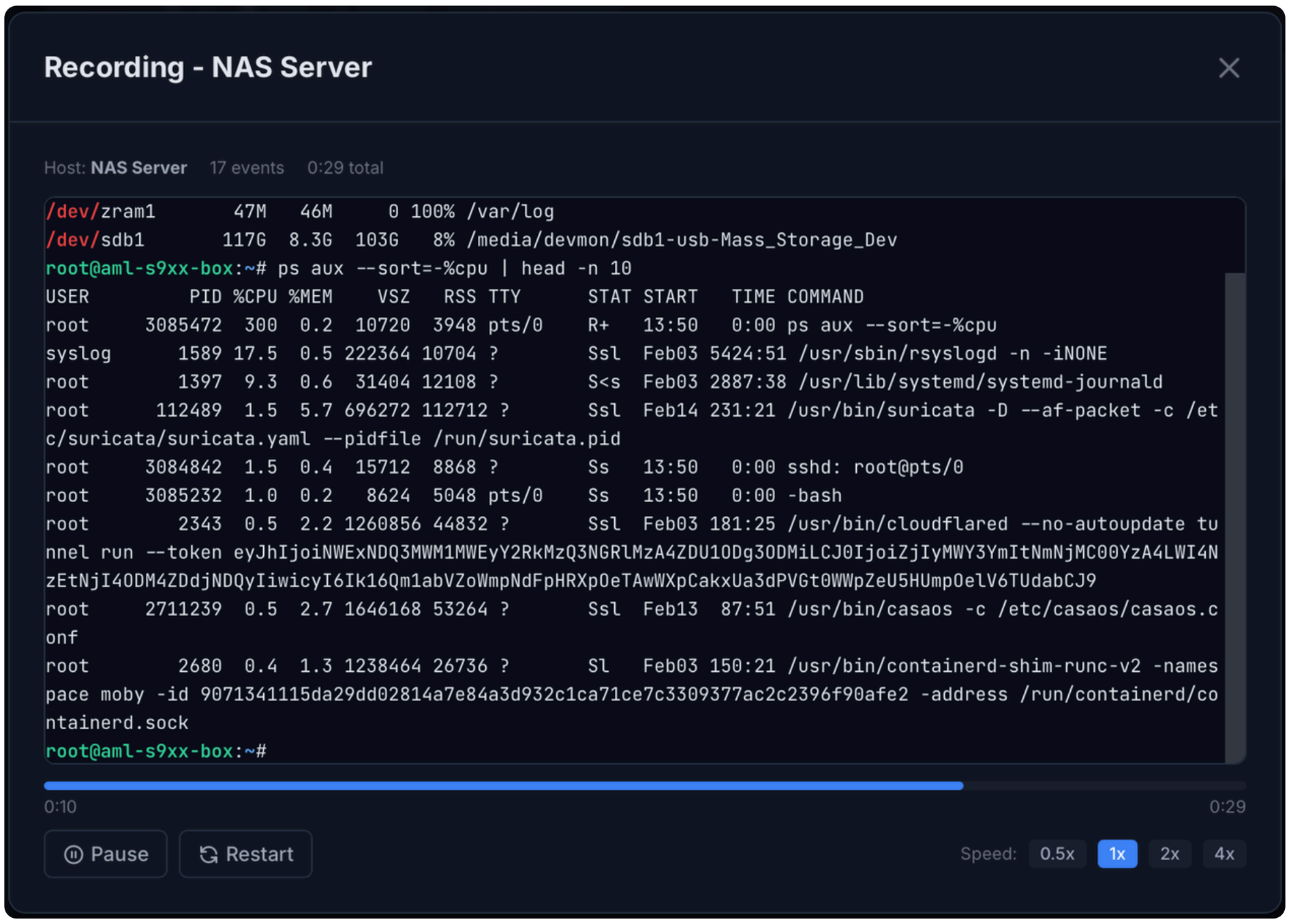
Navigate to the **Recordings** view to browse your recording library.

- Select a recording from the list.
- The **Recording Player** opens in a full-screen dialog.
- Playback uses xterm.js for accurate terminal reproduction.

Player Controls:

Control	Description
Play/Pause/Resume	Toggle playback
Restart	Return to the beginning
0.5x	Half-speed playback
1x	Normal speed
2x	Double speed
4x	Quadruple speed
Progress Bar	Scrub to any point in the recording

The player displays the current time, total duration, host name, and event count.



Recording Player

Settings

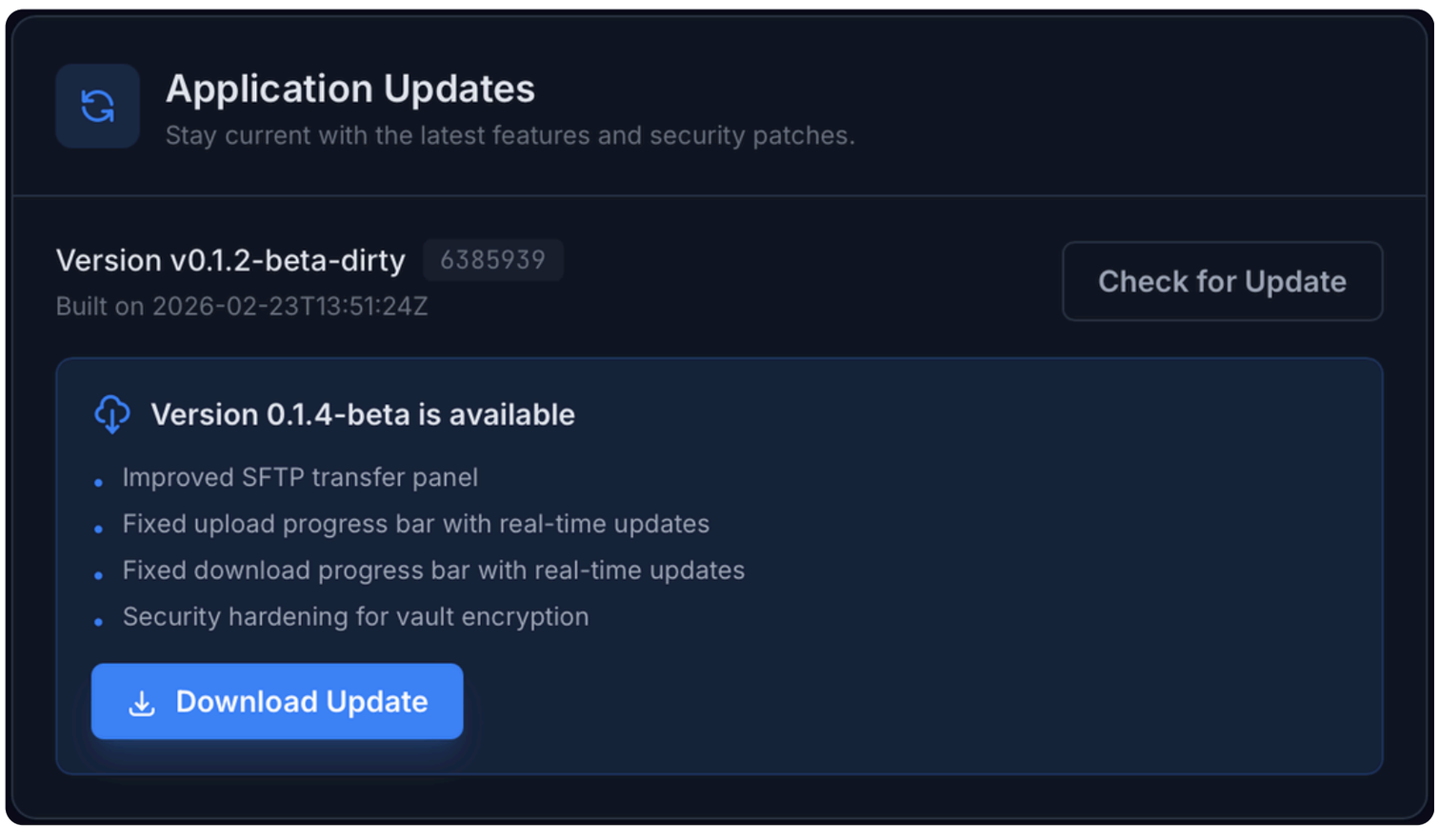
Access the Settings page from the **gear icon** in the sidebar. The page contains three sections.

Application Updates

This section displays your current version information:

- **Version number**
- **Build time**
- **Commit hash**

Click **Check for Update** to query the update manifest. The manifest is cryptographically signed with **Ed25519** and verified before display.



Settings -> Application Updates

Result	Description
Up to date	Green confirmation message
Update available	New version number, release notes, and download link
Required update	Red "Required" badge indicating a critical update

Gosshed does **not** download or install updates automatically. The update checker only verifies and displays information. You choose when and how to update.

Vault Backup and Restore

Export Vault

Create an encrypted portable backup of your entire vault.

1. Enter an **export password** (minimum 6 characters). This password is independent from your master password.
2. Confirm the export password.
3. Click **Export Vault**.
4. Choose a save location using the native file dialog.
5. The vault is exported as a `**.gosshed-vault**` file.

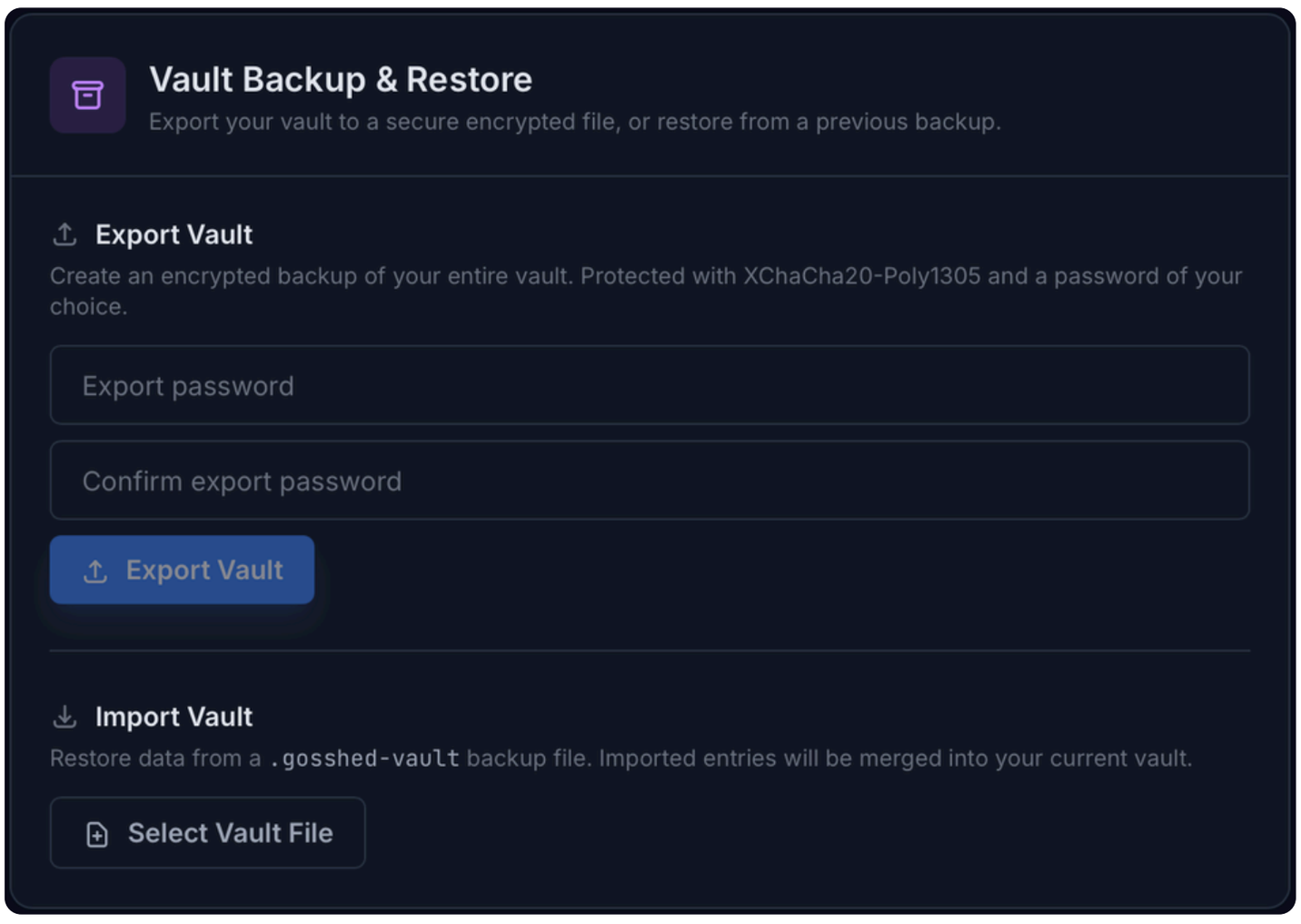
The export is encrypted with XChaCha20-Poly1305 and compressed with zstd. An HMAC-SHA256 tag ensures file integrity.

Import Vault

Restore data from a `**.gosshed-vault**` backup into your current vault.

1. Click **Select Vault File** and choose a `.gosshed-vault` file.
2. A **preview card** displays the backup contents without requiring the password:
    a. Host count
    b. Snippet count
    c. Folder count
    d. File size
    e. Export date
3. Enter the **export password** used when creating the backup.
4. Click **Import**. A confirmation dialog warns:
5. Confirm to proceed. Imported credentials are securely re-encrypted with your current vault's master key.

This will merge all imported data into your current vault. Entries with matching IDs will be permanently overwritten. Proceed with conviction.



Settings -> Vault Backup & Restore

Danger Zone

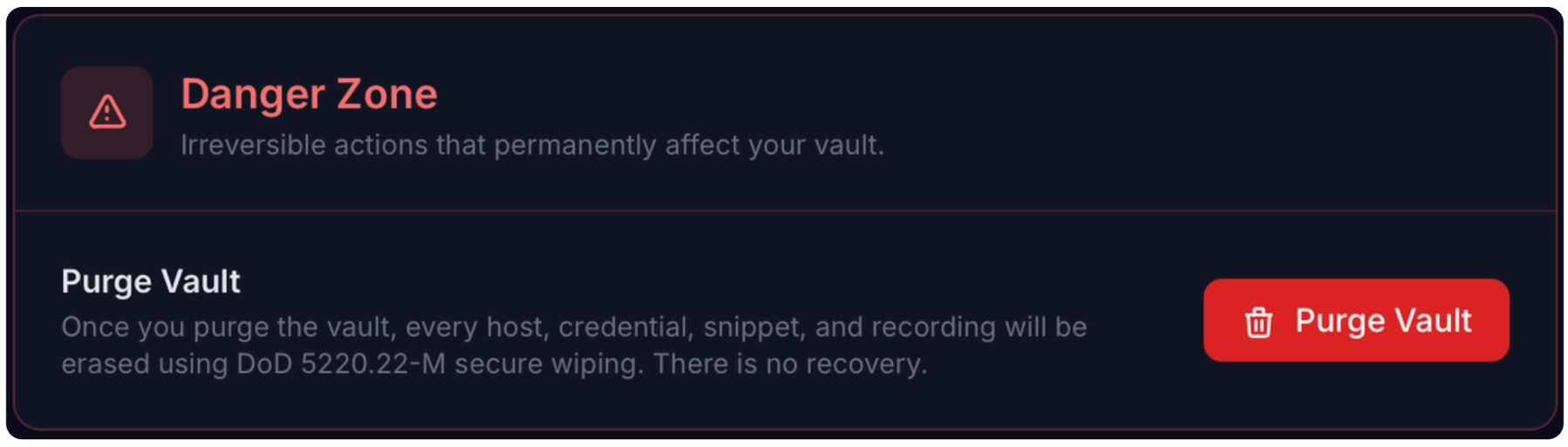
This section contains destructive operations, visually separated with a red border.

Purge Vault
Permanently and irreversibly destroys all vault data using the **DoD 5220.22-M** secure wiping standard (multi-pass overwrite with verification).

The purge process requires two confirmations:

1. **Warning dialog:** Confirms you understand that all hosts, credentials, snippets, recordings, and folders will be destroyed with no possibility of recovery.
2. **Master password verification:** Enter your master password to authorize the purge.

After purging, the application returns to the Vault Setup screen as if launched for the first time.



Settings -> Danger Zone

Jump Host and Bastion Connections

Gosshed supports multi-hop SSH connections through intermediate hosts (jump hosts or bastion servers).

Configuration

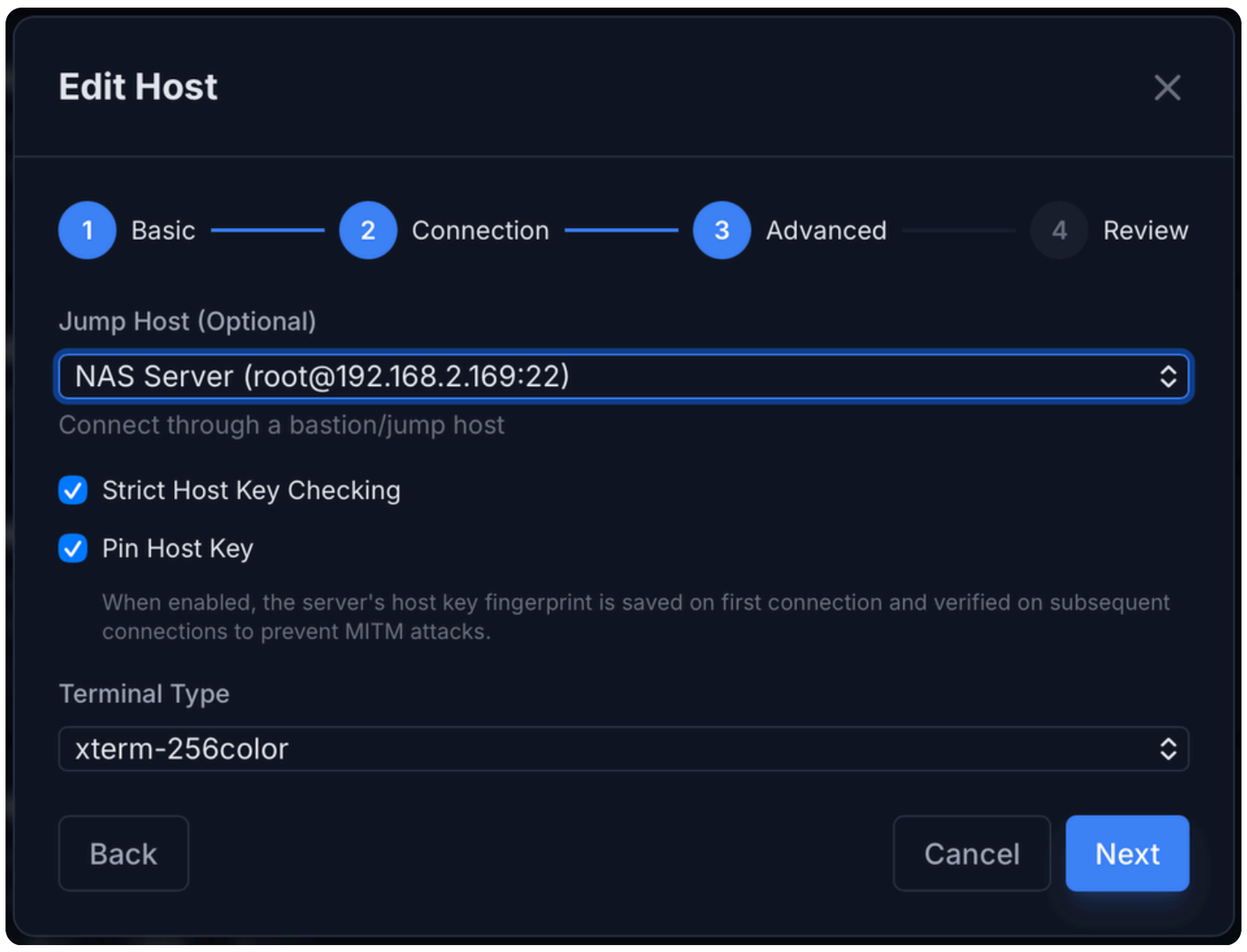
1. First, add the **jump host** as a regular host in your vault with its own credentials.
2. When adding or editing the **target host**, go to Step 3 (Advanced Settings).
3. Select the jump host from the **Jump Host** dropdown.
4. Save the host.

How It Works

When you connect to a target host that has a jump host configured:

1. Gosshed first establishes an SSH connection to the **jump host**.
2. Through that connection, it opens a tunnel to the **target host**.
3. The target host's SSH session runs through the tunnel.
4. The entire chain is transparent, you interact directly with the target host's terminal.

This is equivalent to `**ssh -J bastion target**` in **OpenSSH**.

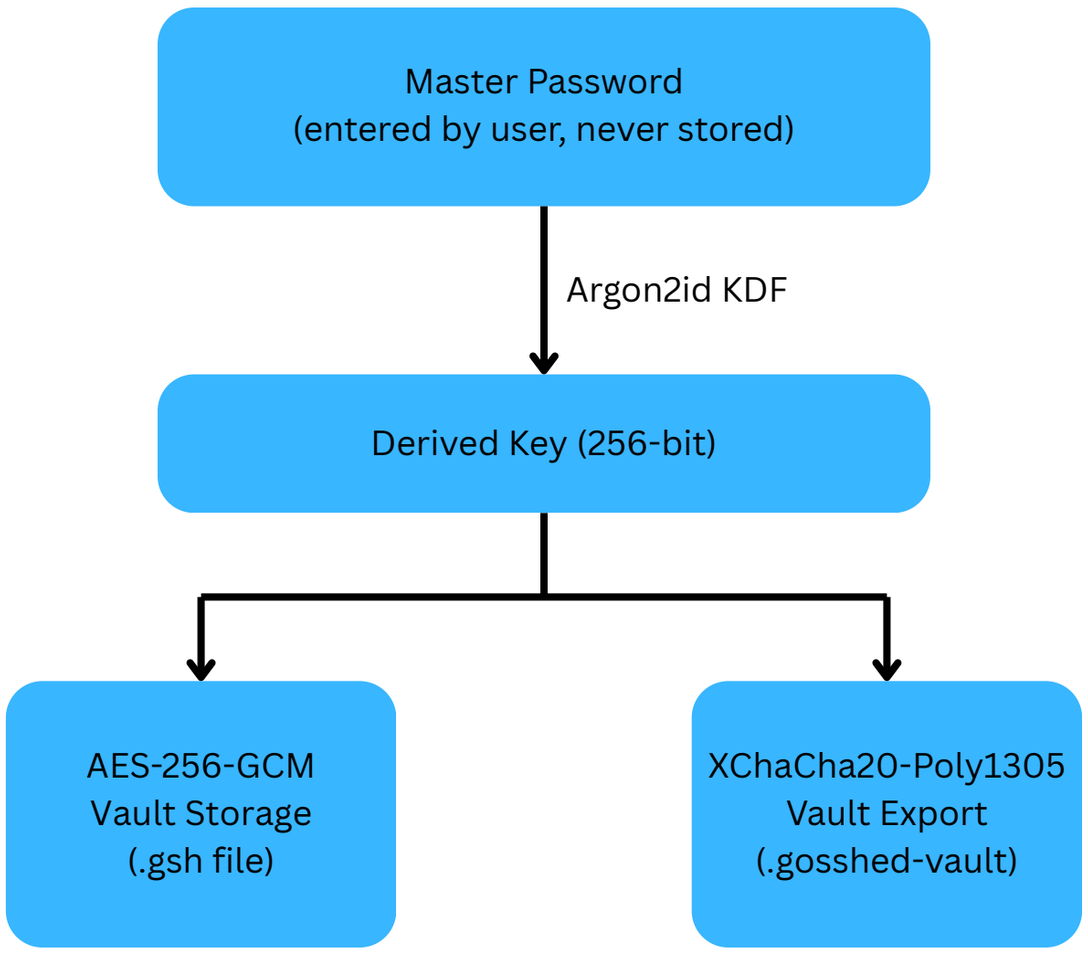


Jump Host Configuration

Security Architecture

Encryption Overview

Gosshed uses a layered encryption architecture with three distinct cryptographic systems, each purpose-built for its use case.



Key Derivation

Parameter	Value
Algorithm	Argon2id (OWASP recommended)
Memory	64 MB (65,536 KiB)
Iterations	3
Parallelism	4 threads
Salt	16 bytes (cryptographically random)
Output Key	32 bytes (256 bits) for vault, 64 bytes for export

Argon2id is a memory-hard key derivation function that resists both GPU-based and side-channel attacks. The parameters are tuned to balance security and performance on modern hardware.

Vault Encryption

Parameter	Value
Algorithm	AES-256-GCM (AEAD)
Key Size	256 bits
Nonce	12 bytes (cryptographically random)
Authentication Tag	16 bytes (128 bits)

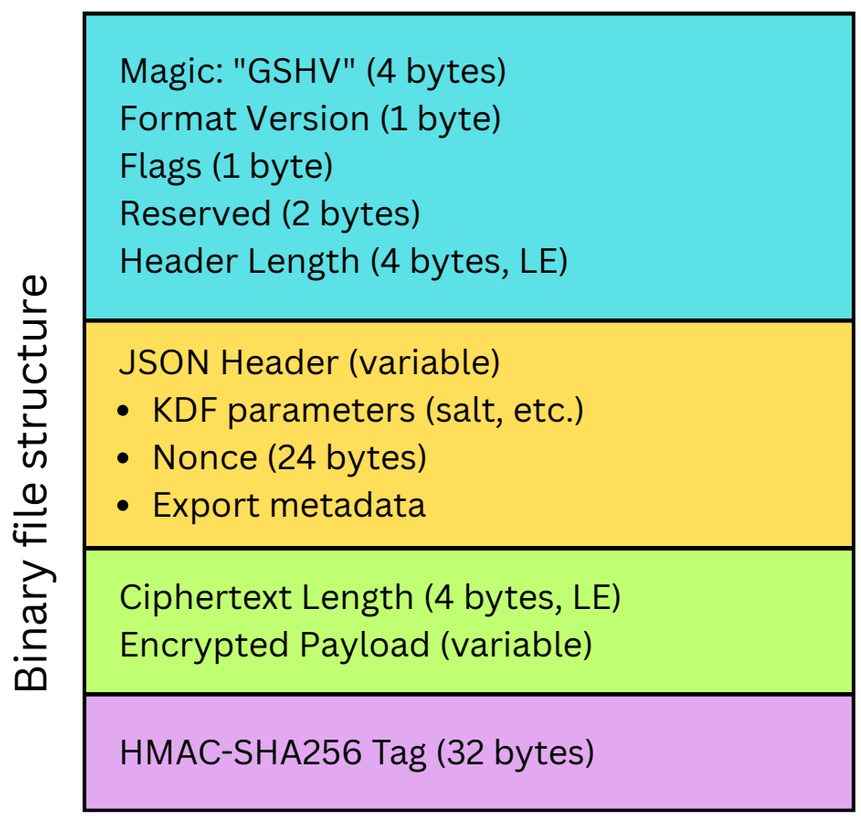
The vault file (`**vault.gsh**`) stores all hosts, credentials, snippets, recordings metadata, and folders. Each credential's sensitive data (private keys, passwords, tokens) is individually encrypted within the vault.

- **Atomic writes** prevent vault corruption during save operations.
- **Password verification** uses an encrypted verification token, no password hash is stored.
- **Password change** re-encrypts the entire vault with the new derived key and creates an automatic backup.

Export Encryption

Parameter	Value
Algorithm	XChaCha20-Poly1305 (AEAD)
Key Size	256 bits
Nonce	24 bytes (192 bits)
Authentication Tag	16 bytes (128 bits)
AAD	`gosshed-export-v1`
Integrity	HMAC-SHA256 (32 bytes) over entire file
Compression	zstd (Zstandard)

The export format uses a custom binary structure:



Key derivation for export produces 64 bytes: 32 for encryption and 32 for the HMAC key. This separation ensures the encryption and integrity keys are cryptographically independent.

Credential re-encryption: During export, each credential is decrypted from the source vault and included in plaintext within the encrypted payload. During import, each credential is re-encrypted with the destination vault's master key. At no point are credentials stored in plaintext on disk.

Secure Memory Handling

- **Memory locking (`mlock`):** Prevents sensitive data from being swapped to disk.
- **Zeroization:** Cryptographic keys and plaintext credentials are overwritten with zeros after use.
- **Core dump prevention:** Core dumps are disabled to prevent memory exposure.
- **Constant-time comparison:** Prevents timing attacks on key verification.

Update Verification

- **Ed25519 signatures:** Update manifests are signed with a private key held offline by maintainers.
- **Public key embedded:** A 32-byte Ed25519 public key is compiled into the application binary.
- **Canonical JSON:** The manifest payload is serialized with recursively sorted keys before signature verification, ensuring deterministic representation.
- **No auto-update:** The checker only verifies and displays information.
- **HTTPS only:** Manifests are fetched exclusively over HTTPS.

Vault Purge (DoD 5220.22-M)

When a vault is purged, the file is securely wiped following the **DoD 5220.22-M** standard:

1. **Pass 1:** Overwrite with zeros (`0x00`).
2. **Pass 2:** Overwrite with ones (`0xFF`).
3. **Pass 3:** Overwrite with random data.
4. **Verification:** Confirm the overwrite succeeded.

This process ensures data cannot be recovered through forensic analysis of the storage medium.

Keyboard Shortcuts

Shortcut	Action
`Enter`	Submit password / confirm dialog
`Ctrl+C` / `Cmd+C`	Copy (in terminal: sends interrupt signal)
`Ctrl+V` / `Cmd+V`	Paste into terminal
Tab click	Switch between terminal sessions

> Terminal keyboard input is forwarded directly to the remote shell. Standard terminal shortcuts (e.g., `Ctrl+C` to interrupt, `Ctrl+D` to logout, `Ctrl+Z` to suspend) behave as expected within the SSH session.

Troubleshooting

Forgot Master Password

There is no password recovery mechanism. Your options:

1. **Restore from backup:** If you previously exported a `.gosshed-vault` file, purge the vault and restore from the backup during setup.
2. **Purge and start fresh:** Go to Settings > Danger Zone > Purge Vault. All data will be permanently destroyed.

Connection Timeouts

- Verify the hostname and port are correct.
- Check that your network allows outbound SSH connections (port 22 or custom).
- If using a jump host, verify the jump host itself is accessible.
- Confirm the target host's SSH service is running.

Host Key Changed Warning

If a host key verification failure occurs on a previously connected host:

- The server may have been reinstalled or reconfigured.
- This could also indicate a **man-in-the-middle attack**.
- Verify the new key fingerprint through an independent channel before re-approving.

SFTP Operations Fail

- Ensure the SSH user has appropriate permissions on the target directories.
- Some servers restrict SFTP access; verify the server configuration allows SFTP subsystem.
- Check disk space on both local and remote systems for upload/download failures.

Export/Import Issues

- **Export fails:** Ensure you have write permission to the target directory and sufficient disk space.
- **Import preview fails:** Verify the file is a valid `.gosshed-vault` file and has not been corrupted.
- **Import decryption fails:** Double-check the export password. The export password is separate from the vault master password.
- **Maximum file size:** Import files are limited to 256 MB.