# SSH Two-Factor Authentication Setup on Ubuntu (Google Authenticator)

This guide helps you set up SSH two-factor authentication (2FA) using Google Authenticator on an Ubuntu VM.

*setup ubuntu vm server for this
link: https://releases.ubuntu.com/24.04.2/ubuntu-24.04.2-live-server-amd64.iso

*set up a live-server ubuntu vm as it comes with preinstalled sshd service

---

## 📋 Prerequisites

- Ubuntu VM (20.04 or later recommended)
- SSH access and sudo privileges
- Installed SSH server

```
puru@puru:~$ sudo systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
     Active: active (running) since Mon 2025-06-02 18:00:47 UTC; 12h ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 2484 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 2485 (sshd)
      Tasks: 1 (limit: 4552)
     Memory: 3.0M (peak: 4.4M)
        CPU: 127ms
     CGroup: /system.slice/ssh.service
             └─2485 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"
```

- *make sure that sshd is active and running*

---

## Step 1: Install Google Authenticator PAM Module

```
sudo apt update
sudo apt install libpam-google-authenticator
```

---

## Step 2: Configure Google Authenticator for Your User

*You will need to install and setup Google Authenticator in your phone

Run the following command as the user who will use SSH:

```
google-authenticator
```

Answer the prompts:

- Choose **time-based tokens**
- Scan the QR code using an app like Google Authenticator or Authy
- Save the emergency codes displayed
- Answer **yes** to the configuration questions (recommended)


---

# Step 3: Configure PAM for SSH

Edit the PAM configuration file for SSH:

```
sudo nano /etc/pam.d/sshd
```

Add this line near the top:

```
auth required pam_google_authenticator.so
```

```
# PAM configuration for the Secure Shell service
auth required pam_google_authenticator.so
# Standard Un*x authentication.
@include common-auth

# Disallow non-root logins when /etc/nologin exists.
account    required     pam_nologin.so

# Uncomment and edit /etc/security/access.conf if you need to set complex
# access limits that are hard to express in sshd_config.
# account  required     pam_access.so
```

---

## Step 4: Configure SSH Daemon

Edit the SSH daemon config:

```
sudo nano /etc/ssh/sshd_config
```

Ensure the following lines are set:

```
ChallengeResponseAuthentication yes
UsePAM yes
AuthenticationMethods password,keyboard-interactive
```

> If you use public key authentication:

```
AuthenticationMethods publickey,keyboard-interactive
```

```
# Authentication:
ChallengeResponseAuthentication yes
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile     .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none
```

---

## Step 5: Restart SSH Service

```
sudo systemctl restart ssh
```

---

## Step 6: Test the Setup

From another system, connect via SSH:

```
ssh username@your-server-ip
```

You should be prompted for:

1. Your account password
2. Your TOTP code (from Google Authenticator)

---

## Summary

| Step | Action |
|---|---|
| 1 | Install PAM module |
| 2 | Configure Google Authenticator per user |
| 3 | Update PAM and sshd_config |
| 4 | Restart SSH |
| 5 | Test login |

---