Self-Hosted Git Service Setup on AWS EC2 (Ubuntu 22.04)

This detailed guide provides the Standard Operating Procedure (SOP) for setting up Gitea, a lightweight, self-hosted Git service, on a minimal AWS EC2 instance. This procedure covers setup, security hardening, Nginx reverse proxy, SSH configuration, and a basic backup strategy.

Prerequisites

- AWS Free-Tier account access.
- Local machine running Linux/macOS/Windows with SSH client.
- SSH Key file: ~/keys/my-ec2.pem (or your chosen path).
- Target OS: **Ubuntu 22.04 LTS**.

1. Launch EC2 Instance (Minimal, Free-Tier Friendly)

Follow these steps in the AWS Management Console:

1. Navigate to **EC2 Launch instance**.
2. **Application and OS Images (AMI):** Select **Ubuntu Server 22.04 LTS**.
3. **Instance type:** Select **t3.micro** (Free Tier eligible).
4. **Key pair (login):** Select or **create a new key pair** (e.g., my-ec2.pem) and ensure the file is secured locally.
5. **Storage:** Default root volume (8-20 GB) is sufficient.
6. **Network settings Security Group (Inbound rules):** Configure the following inbound rules:
   - **SSH TCP 22:** Source: **Your IP only** (recommended).
   - **HTTP TCP 80:** Source: **0.0.0.0/0**.
   - **HTTPS TCP 443:** Source: **0.0.0.0/0**.
7. Click **Launch instance**.

Record the Public IPv4 address of the launched instance. This will be referred to as **<EC2_IP>** in all subsequent commands.

2. SSH into the Instance

From your local machine, execute the following to secure your key and connect:

```
chmod 400 ~/keys/my-ec2.pem
ssh -i ~/keys/my-ec2.pem ubuntu@<EC2_IP>
```

(The default username for Ubuntu AMIs is ubuntu.)

3. System Update and Required Package Installation

Run system updates and install necessary packages (Git, Nginx, Certbot, UFW) on the EC2 instance:

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y git nginx wget ufw certbot python3-certbot-nginx
```

Enable and Configure Firewall (UFW)

Enable the Uncomplicated Firewall (UFW) to only allow required traffic:

```
sudo ufw allow OpenSSH
sudo ufw allow 'Nginx Full'
sudo ufw --force enable
sudo ufw status verbose
```

4. Create Git System User and Gitea Directories

Gitea requires a dedicated system user (git) to run the service and manage repository data.

```
# Create the dedicated system user 'git'
sudo adduser --system --shell /bin/bash --gecos 'Git Version Control' --group
--disabled-password --home /home/git git

# Create data directories
sudo mkdir -p /var/lib/gitea/{custom,data,log,repositories}

# Assign ownership to the 'git' user
sudo chown -R git:git /var/lib/gitea
sudo chmod -R 750 /var/lib/gitea

# Create configuration directory
sudo mkdir /etc/gitea
sudo chown root:git /etc/gitea
sudo chmod 770 /etc/gitea
```

5. Download and Install Gitea Binary

Download the executable binary and move it to a location in the system's PATH.

```
VERSION="1.22.0"     # Check Gitea website for the latest stable version
wget -O gitea
[https://dl.gitea.io/gitea/$](https://dl.gitea.io/gitea/$){VERSION}/gitea-${VERSION}-linux-amd64
sudo mv gitea /usr/local/bin/gitea
sudo chmod +x /usr/local/bin/gitea
```

6. Create Systemd Service and Start Gitea

Create the service file at /etc/systemd/system/gitea.service to allow Gitea to run as a background service managed by the system.

```
# /etc/systemd/system/gitea.service content:
[Unit]
Description=Gitea (Git with a cup of tea)
After=network.target

[Service]
Type=simple
User=git
Group=git
WorkingDirectory=/var/lib/gitea
Environment=USER=git HOME=/home/git GITEA_WORK_DIR=/var/lib/gitea
ExecStart=/usr/local/bin/gitea web --config /etc/gitea/app.ini
Restart=always
RestartSec=2s

[Install]
WantedBy=multi-user.target
```

Execute the following commands to initialize and start the service:

```
sudo systemctl daemon-reload
sudo systemctl enable --now gitea
sudo systemctl status gitea --no-pager
# Check logs for potential errors
sudo journalctl -u gitea -n 200 --no-pager
```

*Gitea is now running on **port 3000**.*

7. Set up Nginx as a Reverse Proxy (for Port 80/443 Access)

This step directs standard web traffic (ports 80/443) to Gitea's internal port (3000). Create the configuration file /etc/nginx/sites-available/gitea.

Note: If testing only with IP, remove the server_name line. If using a domain, replace yourdomain.com.

```
server {
    listen 80;
    server_name yourdomain.com;    # Replace with your domain OR remove for IP testing

    location / {
        proxy_pass [http://127.0.0.1:3000](http://127.0.0.1:3000);
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Host $server_name;
    }
}
```

Enable the configuration and reload Nginx:

```
sudo ln -s /etc/nginx/sites-available/gitea /etc/nginx/sites-enabled/gitea
sudo nginx -t
sudo systemctl reload nginx
```

Install SSL/TLS Certificate with Certbot

If you are using a domain with an A record pointing to <EC2_IP>, run Certbot to configure HTTPS automatically:

```
sudo certbot --nginx -d yourdomain.com
sudo certbot renew --dry-run
```

8. Complete Gitea Web Installer

Navigate to the Gitea installation page in your browser:

- **IP only:** http://<EC2_IP>:3000
- **With Nginx/Domain:** http://yourdomain.com (or https://yourdomain.com if SSL is set up)

Fill in the installer fields exactly:

Section	**Field**	**Value**
Database Settings	Database Type	**SQLite3**
	Path	/var/lib/gitea/data/gitea.db
General Settings	Repository Root Path	/var/lib/gitea/data/gitea-repositories
	Run As Username	git
	SSH Server Port	22
	Gitea HTTP Listen Port	3000
	Gitea Base URL	http://<EC2_IP>:3000/ (or https://yourdomain.com/)
Server and Third-Party	Enable Local Mode	**Checked**
	Disable Self-Registration	**Checked** (Recommended for private hosting)
Administrator Account	Admin Username	*Choose one (e.g., admin)*

Click **Install Gitea** and then log in with your administrator credentials.

9. Create a Repository in Gitea

1. Log in as the Administrator.
2. Click **New Repository**.
3. **Repository Name:** My-test-repo
4. **Initialize Repository: Check this box** (adds README.md and initial commit).
5. Click **Create Repository**.

10. Configure Local Repository Remote URL

On your local machine, navigate to your project directory. Update the remote URL to the correct format, replacing <GITEA_USERNAME> and <EC2_IP>:

```
# If a remote already exists with a bad URL, remove it first
git remote remove origin
git remote add origin http://<EC2_IP>:3000/<GITEA_USERNAME>/My-test-repo.git
```

Always ensure the URL starts with exactly one http:// or https://.

11. Initial Push and Sync

Since the remote repository was initialized via the web UI (Step 9), you must pull the remote content before pushing your local changes.

```
# Commit local changes (if not done)
git add .
git commit -m "initial commit"

# Pull remote changes (README) and rebase local commits on top
git pull origin main --rebase

# Push the merged/rebased branch
git push -u origin main
```

If conflicts occur, resolve them, use git add <files resolved>, and run git rebase --continue.

12. Set up SSH Authentication

This configuration allows password-less interaction with the repository.

1. **Generate SSH key (if needed):**
   ```
   ssh-keygen -t ed25519 -C "your_email@example.com"
   ```
2. **Copy Public Key:**
   ```
   cat ~/.ssh/id_ed25519.pub
   # Copy the entire output line
   ```
3. **Add Key to Gitea:**
   - In the Gitea web UI, go to **Profile SSH / GPG Keys Add Key**.
   - Paste the public key and save.

4. **Update Local Remote URL to SSH:**
   ```
   git remote set-url origin git@<EC2_IP>:<GITEA_USERNAME>/My-test-repo.git
   ```

5. **Test SSH Connection:**
   ```
   ssh -T git@<EC2_IP>
   # Expected reply: "Hi <user>! You've successfully authenticated, but Gitea does not provide shell access."
   ```

13. Backup Procedure

This is a simple procedure for creating an archive of the entire Gitea working directory and configuration.

1. **Stop the Gitea service:**
   ```
   sudo systemctl stop gitea
   ```

2. **Create a compressed archive:**
   ```
   sudo tar czf /root/gitea-backup-$(date +%F).tar.gz /var/lib/gitea /etc/gitea
   ```

3. **Restart the Gitea service:**
   ```
   sudo systemctl start gitea
   ```

4. **Download the backup locally (from your local machine):**
   ```
   scp -i ~/keys/my-ec2.pem ubuntu@<EC2_IP>:/root/gitea-backup-2024-07-25.tar.gz ~/Downloads/
   ```

   Replace the date with the actual date in the backup file name.

14. Troubleshooting Command Reference

Issue	Command	Purpose
Check Gitea service	sudo systemctl status gitea	Confirms the service is active and running.
Review Gitea logs	sudo journalctl -u gitea -n 200 --no-pager	Review recent output for errors.
Check port 3000 status	`ss -tulpn	grep 3000`
Check Nginx config	sudo nginx -t	Checks for syntax errors in the Nginx configuration.

Check SSH authorized keys	sudo cat /home/git/.ssh/authorized_keys	Verifies Gitea successfully wrote the public key.