set up **Gitea** using **Docker Swarm and Portainer**

First, I created a Docker volume named **gitea_data**.
This was done to store Gitea files permanently so that data is not lost.

Next, I created a **Gitea service** in Portainer using the image
gitea/gitea:latest and set **replicas = 1**. And set environmental variable

GITEA__server__HTTP_PORT=3000

At first, I used the **host network**, but the Gitea web page was not opening.

So, I checked the **logs** in Portainer.
I noticed that the container was stopping automatically due to **SIGTERM**.
This happened because of **port conflicts**.

To solve this problem, I changed the network mode from **host** to **ingress**
and mapped **port 3000**.

After restarting the service, the Gitea web interface opened successfully at
http://localhost:3000.

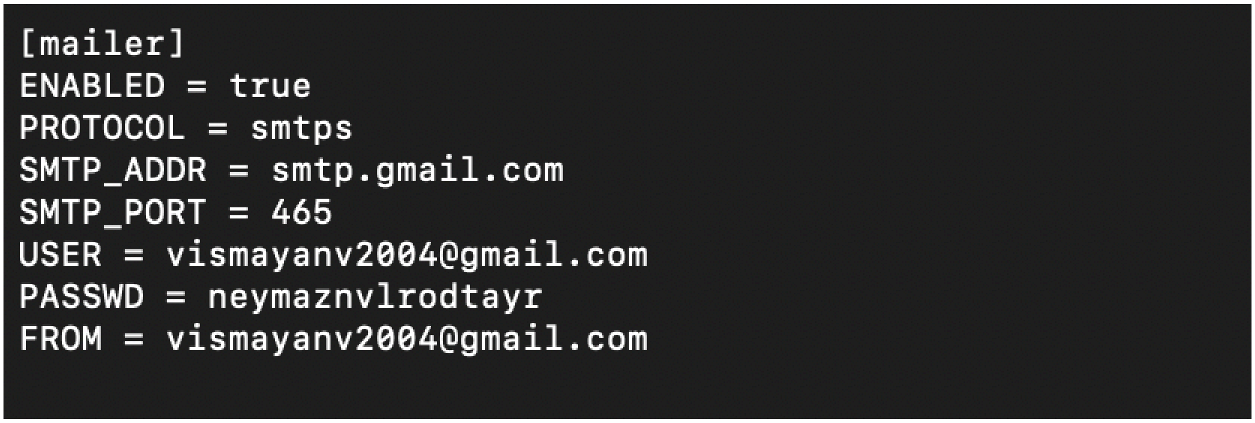
Email Sending in Gitea:

Step 1: Create Gmail App Password

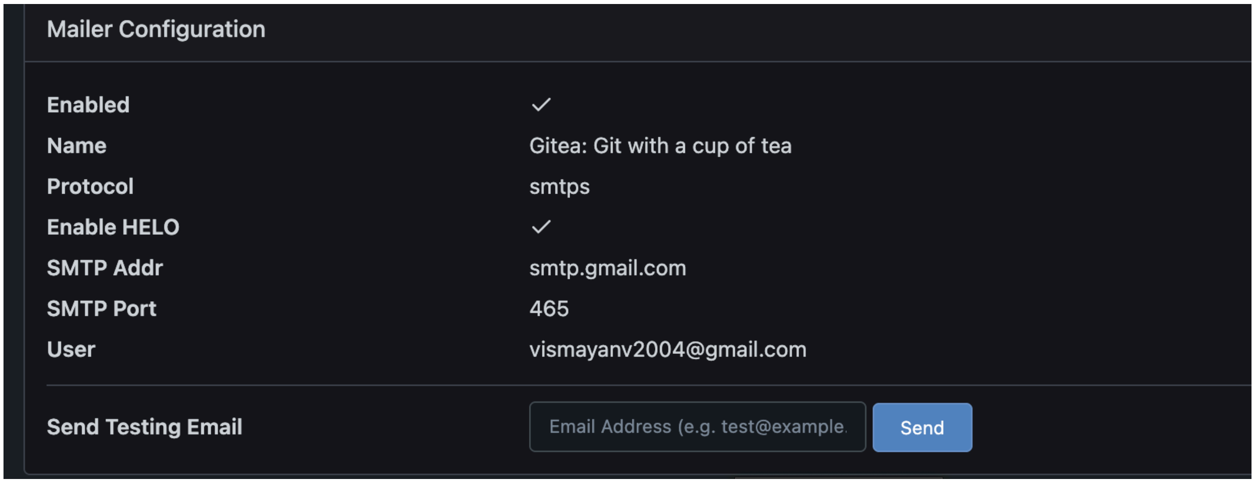
- Enabled **2-Step Verification** in Gmail
- Generated a **16-character App Password**
- Used this password only for email sending

Step 2: Configure Gitea Email Settings

Edited the Gitea configuration file:



```
[mailer]
ENABLED = true
PROTOCOL = smtps
SMTP_ADDR = smtp.gmail.com
SMTP_PORT = 465
USER = vismayanv2004@gmail.com
PASSWD = neymaznvlrodtayr
FROM = vismayanv2004@gmail.com
```



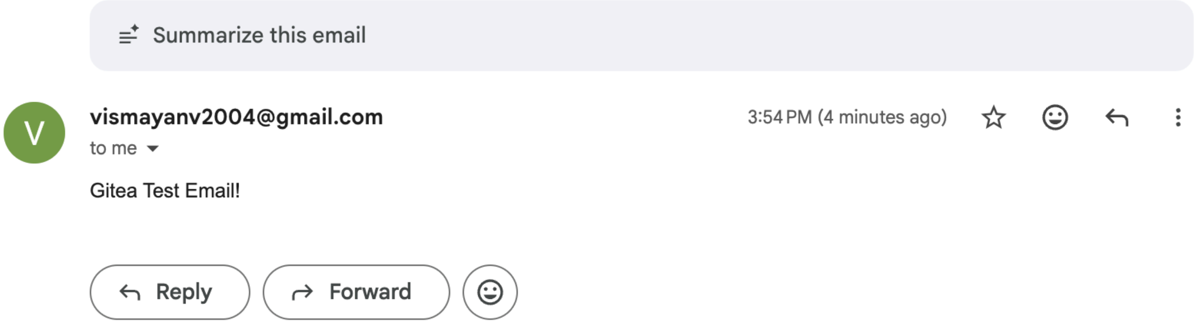
Mailer Configuration

Enabled	✓
Name	Gitea: Git with a cup of tea
Protocol	smtps
Enable HELO	✓
SMTP Addr	smtp.gmail.com
SMTP Port	465
User	vismayanv2004@gmail.com
Send Testing Email	Email Address (e.g. test@example. Send

Step 3: Restart Gitea

Step 4: Test Email

- Opened **Gitea Admin → Configuration**
- Clicked **Send Test Email**
- Email was delivered successfully



Summarize this email

vismayanv2004@gmail.com 3:54 PM (4 minutes ago)

to me

Gitea Test Email!

Difficulties Faced and Solutions

❌ Problem 1: SMTP server does not support AUTH

Reason:
Gitea was using a local SMTP server that does not support authentication.

Solution:
Configured **Gmail SMTP (smtp.gmail.com)** with proper authentication.

❌ Problem 2: EOF error while sending email

Reason:
Mismatch between SMTP protocol and port.

Solution:
Used **SMTPS protocol with port 465**, which is fully supported by Gmail.

❌ Problem 3: Gmail login failed

Reason:
Normal Gmail password was blocked by Google.

Solution:
Used a **Gmail App Password** instead of the normal password.

📘 Gitea Deployment using Docker Swarm with Self-Hosted GitHub Actions Runner

📁 Project Structure

```
gitea-setup/
│
├── .github/
│   └── workflows/
│       └── deploy.yaml         # GitHub Actions workflow
│
├── gitea-stack.yaml            # Docker Swarm stack file
├── .gitignore                 # Ignore secrets & local files
└── README.md                   # Documentation
```

🔐 Security & Secrets Management

GitHub Secrets & Variables

Configured in:

```
Repo → Settings → Secrets and variables → Actions
```

Name	Type	Description
GITEA_EMAIL	Variable	Email used by Gitea
GITEA_EMAIL_PASSWORD	Secret	Gmail App Password

Gitea-stack.yaml

```
version: "3.9"

services:
  gitea:
    image: gitea/gitea:1.22
    ports:
      - "3000:3000"
      - "2222:22"
    volumes:
      - gitea_data:/data

    environment:
      USER_UID: 1000
      USER_GID: 1000

      # Server
      GITEA__server__DOMAIN: localhost
      GITEA__server__ROOT_URL: http://localhost:3000/
      GITEA__server__SSH_PORT: 2222
      GITEA__server__SSH_DOMAIN: localhost

      # Mail
      GITEA__mailer__ENABLED: "true"
      GITEA__mailer__PROTOCOL: smtps
      GITEA__mailer__SMTP_ADDR: smtp.gmail.com
      GITEA__mailer__SMTP_PORT: "465"
      GITEA__mailer__USER: ${GITEA_EMAIL}
      GITEA__mailer__PASSWD: ${GITEA_EMAIL_PASSWORD}
      GITEA__mailer__FROM: Gitea <${GITEA_EMAIL}>

      # Security
      GITEA__security__INSTALL_LOCK: "true"
      GITEA__service__DISABLE_REGISTRATION: "true"

    deploy:
      replicas: 1
      restart_policy:
        condition: any
      placement:
        constraints:
          - node.role == manager

volumes:
  gitea_data:
```

.github/workflows/deploy.yaml

```
name: Deploy Gitea on Docker Swarm

on:
  push:
    branches:
      - main

jobs:
  deploy:
    runs-on: self-hosted

    steps:
      - name: Checkout code
        uses: actions/checkout@v4

      - name: Export secrets
        run: |
          echo "GITEA_EMAIL=${{ vars.GITEA_EMAIL }}" >> $GITHUB_ENV
          echo "GITEA_EMAIL_PASSWORD=${{
secrets.GITEA_EMAIL_PASSWORD }}" >> $GITHUB_ENV

      - name: Deploy Gitea
        run: |
          docker swarm init || true
          docker stack deploy -c gitea-stack.yaml gitea

      - name: Verify services
        run: docker service ls
```

🖥️ Self-Hosted Runner Setup

Location (IMPORTANT)

Runner must be **outside the Git repository**:

```
~/actions-runner/
```

And run commands specified in runner setup

🚀 Deployment Flow

1. Developer pushes to `main`
2. GitHub Actions triggers workflow
3. Job runs on self-hosted runner
4. Secrets injected into environment
5. Docker Swarm deploys/updates Gitea
6. Gitea available at `http://localhost:3000`