Yuehan Zhang

Contact me:
johnzhang514145@gmail.com

Life Update

By - Yuehan Zhang – Posted on 2025-10-12 – Posted in Cyber science

Private Deployment of Team Note-Taking Software: Tutorial for Deploying Outline on Ubuntu

Summary

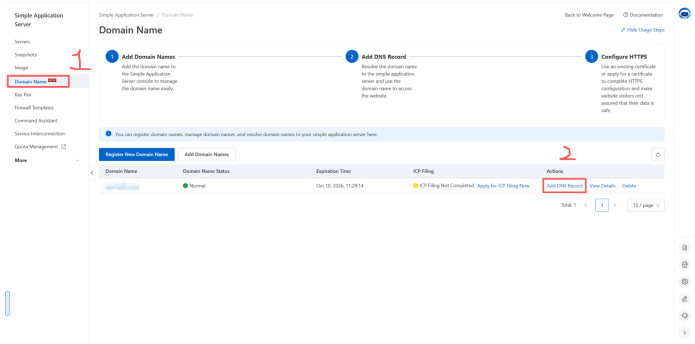
With the growing demand for team collaboration, building self-hosted code hosting and knowledge base systems has become a practical need for many individuals and teams—such systems can better ensure data privacy, support personalized configurations, and adapt to collaboration modes in different scenarios. This document provides a detailed, step-by-step guide to deploying Forgejo (a code hosting platform) and Outline (a knowledge base tool) on an Ubuntu server, covering everything from basic environment preparation (e.g., purchasing ECS and domain names, installing Docker, Nginx, and PostgreSQL) to advanced configurations (e.g., setting up multi-login options via Forgejo and Microsoft

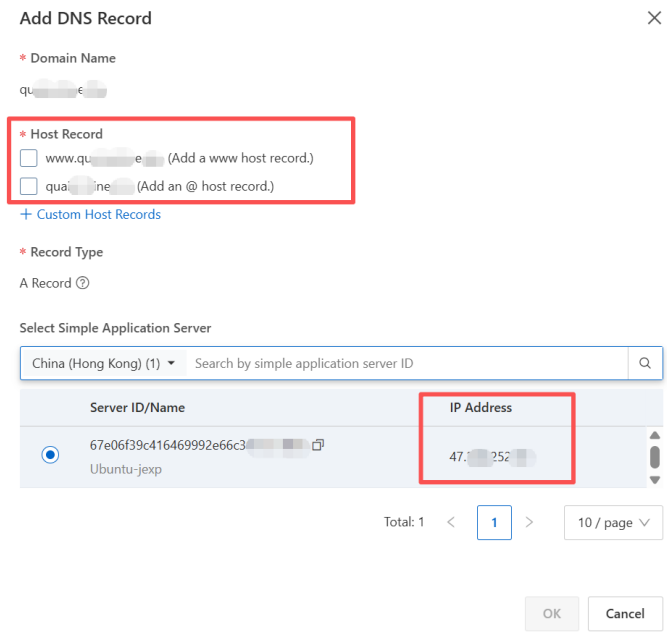
Azure). Whether you are a developer, operation and maintenance personnel, or team manager who needs to build self-hosted collaboration tools, you can complete the entire deployment smoothly by following the steps, and finally obtain a customizable code hosting and knowledge management system to support internal team collaboration.

0.Buy a Elastic Compute Service and domain name

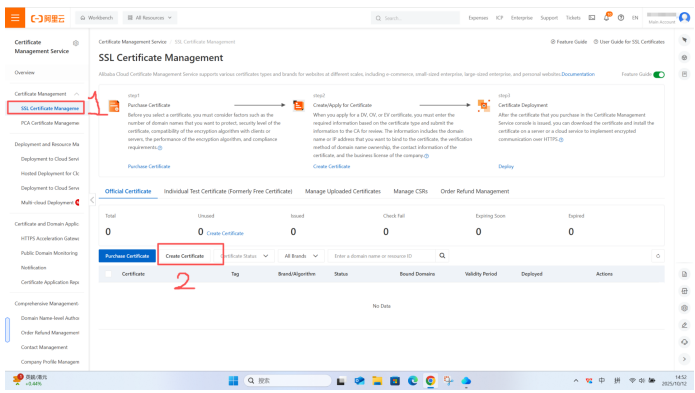
If you already have an Ubuntu server and a domain name, you can skip this section.

This article uses an Alibaba Cloud ECS (Elastic Compute Service). You can also choose other platforms, such as Amazon EC2 (Elastic Compute Cloud). First, you need to purchase a domain name on your selected platform. The specific process will not be repeated here as it is relatively simple. After purchasing the domain name, you need to add a DNS record for it, which means associating your domain name with your server's IP address. Please refer to the image for the relevant operations.





Next, we need to download the SSL certificate. For the download process on the Alibaba Cloud platform, you can refer to this article: https://help.aliyun.com/zh/ssl-certificate/user-guide/download-an-ssl-certificate?spm=a2c4g.11186623.0.i3



After these files are downloaded, you will get two files. You need to rename them to ssl.key and ssl.pem respectively.

Next, upload these two files to the following paths on the Ubuntu server:
/etc/nginx/cert/ssl.key
/etc/nginx/cert/ssl.pem
If this path does not exist, simply create a new folder named /etc/nginx/cert yourself.

1.Install Docker

```
sudo apt install -y python3 python3-pip

sudo pip3 install docker-compose
```

2.Verify if Docker is installed successfully

```
docker-compose --version
```

3.Deploy Forgejo

```
mkdir forgejo
```

Write a **docker-compose** file: In the **forgejo** folder, create a file named **forgejo.yaml** with the following content. Please copy it directly.

```
networks:
  forgejo:
    external: false
services:
  server:
    image:
codeberg.org/forgejo/forgejo:9
    container_name: forgejo
    environment:
      - USER_UID=1000
      - USER_GID=1000
    restart: always
    networks:
      - forgejo
    volumes:
      - /data/forgejo:/data
      -
/etc/timezone:/etc/timezone:ro
      -
/etc/localtime:/etc/localtime:ro
    ports:
      - '3002:3000'
      - '222:22'
```

4.Start Forgejo Service

```
docker-compose -f forgejo.yaml up -d
```

5.Use Forgejo and Microsoft azure

```
sudo ufw allow 3002/tcp

sudo ufw allow 3002/udp

sudo ufw statu
```

Access it in the browser via the internal network http://IP:3002. For example, mine is http://10.6.158.157:3002. You need to set your own password and complete a series of registration procedures. Then you'll reach this page.

Click the avatar in the upper right corner, then click Settings.

Click on Applications, then click Create Application. Complete a series of registration procedures, and record all IDs and passwords. It is recommended that you take a screenshot to save them. After clicking the “Create” button, the page will display the Client ID and Client Secret. Immediately copy these two pieces of information and save them in a notepad (because the Client Secret will disappear after refreshing the page, and it will be needed for subsequent Outline

configuration).

Change the “Redirect URL” to: http://your-ip:13090/auth/oidc.callback

Next, I have also set up a login option using Microsoft’s cloud services. So now I have two login options. If you only need the previous one, you can ignore the following deployment of the Microsoft option.

First, you need to log in to this URL: https://portal.azure.com/, and register your own account. Then click the three horizontal lines in the top-left corner.

Then, click Microsoft entra ID

Then, Manage and App registrations

Then, New registrations

Platform: web

Redirect URL: https://your domain or

ip:13090/auth/azure.callback

Create your secret and remember the value immediately.

This ID and Secret are the Microsoft-related information you need to fill in the Docker .env file later. Once the Secret is generated, you must record it immediately, as it will not be displayed again afterwards. (It's value, not Secret ID)

6.Install Nginx on the server

```
sudo apt update
```

```
sudo apt install nginx

sudo systemctl start nginx

sudo systemctl enable nginx

sudo systemctl status nginx
```

Add a configuration file in the /etc/nginx/nginx.conf

Enter the sites-enabled directory and create a new configuration file, for example, named nginx.conf.

You need to change the server_name to your own registered domain name.

```
user root;
worker_processes auto;
pid /run/nginx.pid;
error_log
/var/log/nginx/error.log warn;
include /etc/nginx/modules-
enabled/*.conf;

events {
    worker_connections 1024;
    multi_accept on;
}

http {

    sendfile on;
    tcp_nopush on;
```

```
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;

    include
/etc/nginx/mime.types;
    default_type
application/octet-stream;

    ssl_protocols TLSv1.2
TLSv1.3;
    ssl_prefer_server_ciphers
on;
    ssl_ciphers ECDHE-ECDSA-
AES128-GCM-SHA256:ECDHE-RSA-
AES128-GCM-SHA256:ECDHE-ECDSA-
AES256-GCM-SHA384:ECDHE-RSA-
AES256-GCM-SHA384:ECDHE-ECDSA-
CHACHA20-POLY1305:ECDHE-RSA-
CHACHA20-POLY1305:DHE-RSA-
AES128-GCM-SHA256:DHE-RSA-
AES256-GCM-SHA384;
    ssl_session_cache
shared:SSL:10m;
    ssl_session_timeout 1d;
    ssl_stapling on;
    ssl_stapling_verify on;

    access_log
/var/log/nginx/access.log;
    error_log
/var/log/nginx/error.log;

    gzip on;
    gzip_vary on;
    gzip_proxied any;
    gzip_comp_level 6;
    gzip_buffers 16 8k;
```

```
    gzip_http_version 1.1;
    gzip_types text/plain
text/css application/json
application/javascript text/xml
application/xml
application/xml+rss
text/javascript;

    include
/etc/nginx/conf.d/*.conf;
    include /etc/nginx/sites-
enabled/*;

    server {
        listen 13090 ssl;
        server_name ?;

        ssl_certificate
/etc/nginx/cert/ssl.pem;
        ssl_certificate_key
/etc/nginx/cert/ssl.key;

        add_header Strict-
Transport-Security "max-
age=31536000; includeSubDomains"
always;
        add_header X-Content-
Type-Options nosniff;
        add_header X-Frame-
Options SAMEORIGIN;
        add_header X-XSS-
Protection "1; mode=block";

        location / {
            proxy_pass
http://outline:3000;

            proxy_set_header
```

```
Host $host;
            proxy_set_header X-
Real-IP $remote_addr;
            proxy_set_header X-
Forwarded-For
$proxy_add_x_forwarded_for;
            proxy_set_header X-
Forwarded-Proto $scheme;
            proxy_set_header X-
Forwarded-Port $server_port;

            proxy_http_version
1.1;
            proxy_set_header
Upgrade $http_upgrade;
            proxy_set_header
Connection "upgrade";


proxy_connect_timeout 300s;
            proxy_read_timeout
300s;
            proxy_send_timeout
300s;
        }

    }

    server {
        listen 80;
        server_name ?;

        return 301
https://$host:13090$request_uri;
    }
}
```

Need to restart the Nginx service to make the new configuration take effect.

```
sudo systemctl restart nginx
```

7.Install the PostgreSQL server and client:

Execute the command to install

```
sudo apt install postgresql
postgresql-contrib
```

Create a database user

Use the `CREATE USER` statement to create a user named `outline` and set the password to `password`. The command is as follows:

```
CREATE USER outline WITH
PASSWORD 'password';
```

Create a database

Use the `CREATE DATABASE` statement to create a database named `outline`. The command is as follows:

```
CREATE DATABASE outline;
```

Grant user permissions on the database

Grant all permissions on the database `outline` to the user `outline` so that the user can connect to and operate this database. The command is as follows:

GRANT ALL PRIVILEGES ON DATABASE outline TO outline;

8.Configure the Outline file.

Create a file named `docker.env` in the `outline` folder. Please write your own IP address or domain name, as well as your own **CLIENT ID** and **SECRET** in it.

```
# ——————————————— REQUIRED ———————————————

# Generate a 32-byte random key encoded in hexadecimal. You should use `openssl rand -hex 32` in your terminal to generate a random value.

SECRET_KEY=00075933fd85083d8037964fa8e5539088251e9b8722bdffeb732f57a32e929d

# Generate a 32-byte random key encoded in hexadecimal. You should use `openssl rand -hex
```

```
32` in your terminal to generate
a random value.

UTILS_SECRET=e35305f201b6556e24c
8c2c88e4e24c1be5625273ec0d3bd4e0
aa18393d7d065

# database

DATABASE_URL=postgres://outline:
password@postgres:5432/outline
DATABASE_URL_TEST=postgres://out
line:password@postgres:5432/outl
ine_test
DATABASE_CONNECTION_POOL_MIN=0
DATABASE_CONNECTION_POOL_MAX=10

# Uncomment to disable SSL
connection to Postgres
PGSSLMODE=disable

# For Redis, you can specify an
ioredis-compatible URL like this
# Here, "redis" is the default
hostname of the database
container created above; a
separate network needs to be
established or --link used
REDIS_URL=redis://outline-redis-
1:6379
# Alternatively, if you want to
provide additional connection
options,
# use a base64-encoded JSON
connection options object. Refer
to the ioredis documentation for
a list of available options.
# Example: Using Redis Sentinel
```

```
for high availability
# {"sentinels":
[{"host":"sentinel-
0","port":26379},
{"host":"sentinel-
1","port":26379}],"name":"mymast
er"}
#REDIS_URL=ioredis://eyJzZW50aW5
lbHMiOlt7Imhvc3QiOiJzZW50aW5lbC0
wIiwicG9ydCI6MjYzNzl9LHsiaG9zdCI
6InNlbnRpbmVsLTEiLCJwb3J0IjoyNjM
3OX1dLCJuYW1lIjoibXltYXN0ZXIifQ=
=

URL=https://ip or domain
name:13090/
PORT=3000

# don't change this line
COLLABORATION_URL=

# A more detailed guide on
setting up S3 is available here:
# =>
https://wiki.generaloutline.com/
share/125de1cc-9ff6-424b-8415-
0d58c809a40f
# AWS_ACCESS_KEY_ID corresponds
to MINIO_ROOT_USER above
# AWS_SECRET_ACCESS_KEY
corresponds to
MINIO_ROOT_PASSWORD above
# AWS_REGION corresponds to
MINIO_REGION_NAME above
# AWS_S3_UPLOAD_BUCKET_URL is
the API address of MINIO; note
that this is the API address,
not the management address
```

```
AWS_ACCESS_KEY_ID=6m2lx2ffmbr9ikod
AWS_SECRET_ACCESS_KEY=2k78fpraq7rs5xlrti5p6cvb767a691h3jqi47ihbu75cx23twkzpok86sf1aw1e
AWS_REGION=cn-homelab-1
AWS_S3_ACCELERATE_URL=
AWS_S3_UPLOAD_BUCKET_URL=https://quairnote.cn:9000
AWS_S3_UPLOAD_BUCKET_NAME=outline
AWS_S3_UPLOAD_MAX_SIZE=26214400
AWS_S3_FORCE_PATH_STYLE=true
AWS_S3_ACL=private

# ———————————— Authentication ————————————

# Third-party login credentials. A working installation requires at least one of Google, Slack, or Microsoft; otherwise, you will have no login options.

# Slack
# => https://api.slack.com/apps
# When configuring the Client ID, add the redirect URL under "OAuth & Permissions":
# https://<URL>/auth/slack.callback
# SLACK_KEY=
# SLACK_SECRET=

# # To configure Google authentication, you need to
```

```
create an OAuth client ID at the
following location:
# =>
https://console.cloud.google.com
/apis/credentials
#
# When configuring the Client
ID, add the Authorized redirect
URI:
#
https://<URL>/auth/google.callba
ck
GOOGLE_CLIENT_ID=
GOOGLE_CLIENT_SECRET=

# To configure Microsoft/Azure
authentication, you need to
create an OAuth client.
# Refer to the guide for
detailed information on setting
up an Azure application:
# =>
https://wiki.generaloutline.com/
share/dfa77e56-d4d2-4b51-8ff8-
84ea6608faa4
AZURE_CLIENT_ID=your id
AZURE_CLIENT_SECRET=your secret
AZURE_RESOURCE_APP_ID=00000003-
0000-0000-c000-000000000000

# To configure generic OIDC
authentication, you need an
identity provider (IdP) of some
kind.
# The redirect URI is
https://<URL>/auth/oidc.callback

OIDC_CLIENT_ID = your id
```

```
OIDC_CLIENT_SECRET=your secret
OIDC_AUTH_URI=http://your
ip:3002/login/oauth/authorize
OIDC_TOKEN_URI=http://your
ip:3002/login/oauth/access_token
OIDC_USERINFO_URI=http://your
ip:3002/login/oauth/userinfo

#
OIDC_USERNAME_CLAIM=preferred_us
ername

# OIDC_DISPLAY_NAME=OIDC

# OIDC_SCOPES=openid profile
email

# ———————————— 可选
————————————

# This is only required if you
do not use an external reverse
proxy. See documentation:
#
https://wiki.generaloutline.com/
share/1c922644-40d8-41fe-98f9-
df2b67239d45
SSL_KEY=
SSL_CERT=

# If using a
Cloudfront/Cloudflare
distribution or similar it can
be set below.
# This will cause paths to
javascript, stylesheets, and
images to be updated to
# the hostname defined in
```

```
CDN_URL. In your CDN
configuration the origin server
# should be set to the same as
URL.
CDN_URL=

# Automatically redirect to
HTTPS in production.
# The default value is true, but
it can be set to false if you
can ensure SSL termination at an
external load balancer.

FORCE_HTTPS=true

# Allow installers to check for
updates by sending anonymous
statistics to the maintainers
ENABLE_UPDATES=false

# How many processes should be
spawned. As a reasonable rule of
thumb, divide the server's
available memory by 512 for a
rough estimate
WEB_CONCURRENCY=4

# If you have particularly large
Word documents with embedded
images, you may need to override
the maximum size for document
imports
MAXIMUM_IMPORT_SIZE=5120000

# If your reverse proxy already
logs incoming HTTP requests and
results in duplicates, this line
can be removed
```

```
DEBUG=http

# Comma-separated list of
domains allowed to log in to the
wiki. If not set, all domains
are allowed by default when
logging in with Google OAuth
ALLOWED_DOMAINS=

# For full integration with
search and posting to channels,
the following configuration is
also required, with more details
available at
# =>
https://wiki.generaloutline.com/
share/be25efd1-b3ef-4450-b8e5-
c4a4fc11e02a
#
#
SLACK_VERIFICATION_TOKEN=your_to
ken
# SLACK_APP_ID=A0XXXXXXX
# SLACK_MESSAGE_ACTIONS=true

# Google Analytics can also be
optionally enabled to track page
views in the knowledge base
GOOGLE_ANALYTICS_ID=

# Optionally enable Sentry
(Sentry.io) to track errors and
performance
SENTRY_DSN=

# To support sending outgoing
transactional emails such as
"Document Updated" or "You've
```

```
Been Invited", you need to
provide authentication for the
SMTP server
SMTP_HOST=
SMTP_PORT=
SMTP_USERNAME=
SMTP_PASSWORD=
SMTP_FROM_EMAIL=
SMTP_REPLY_EMAIL=
SMTP_TLS_CIPHERS=
SMTP_SECURE=true

# Custom logo displayed on the
authentication screen, scaled to
height: 60px
#
TEAM_LOGO=https://example.com/im
ages/logo.png

# Default interface language.
See translate.getoutline.com for
a list of available language
codes and their approximate
translation percentages.Change
it.
DEFAULT_LANGUAGE=zh_CN
```

Create a file named `docker-compose.yml` in the `outline` folder, and do not modify anything in this file.

```
services:
  outline:
    image:
outlinewiki/outline:0.82.0
    env_file: ./docker.env
```

```
    ports:
      - "3000:3000"
    expose:
      - "3000"
    volumes:
      - storage-
data:/var/lib/outline/data
    depends_on:
      - postgres
      - redis

  redis:
    image: redis
    env_file: ./docker.env
    expose:
      - "6379"
    volumes:
      - ./redis.conf:/redis.conf
    command: ["redis-server",
"/redis.conf"]
    healthcheck:
      test: ["CMD", "redis-cli",
"ping"]
      interval: 10s
      timeout: 30s
      retries: 3

  postgres:
    image: postgres
    env_file: ./docker.env
    expose:
      - "5432"
    volumes:
      - database-
data:/var/lib/postgresql/data
    healthcheck:
      test: ["CMD",
"pg_isready", "-d", "outline",
```

```
"-U", "user"]
      interval: 30s
      timeout: 20s
      retries: 3
    environment:
      POSTGRES_USER: 'outline'
      POSTGRES_PASSWORD:
'password'
      POSTGRES_DB: 'outline'

volumes:
  storage-data:
  database-data:
```

Start the dependent services (Postgres + Redis) in the outline folder: Execute the command in the deployment folder:

```
docker compose up -d postgres
redis
```

Start the Outline service in the outline folder: Execute the command:

docker compose up -d outline

9.Try it

Then you need to access https://your domain name:13090/ in your browser.

10.How to create new user through OIDC

First, enter this page: http://your-ip:3002

Yuehan Zhang
Contact me: johnzhang514145@gmail.com

Next Article

Decoded Quantum Interferometry for the max-XORSAT Problem

Leave a Reply

Your email address will not be published. Required fields are marked *.

Your N* Your E* Website

Comment

Save my name, email, and website in this browser for the next time I comment.

POST COMMENT