Focalboard Project Management Setup on Ubuntu 22.04

Date: Tuesday, May 14, 2024

In this tutorial, we will walk through the process of installing and setting up Focalboard, an open-source self-hosted project management tool, on an Ubuntu 22.04 server. Focalboard is an alternative to tools like Asana, Trello, and Notion, and can be used on Windows, Mac, or Linux systems.

Prerequisites

- **Server OS:** Ubuntu 22.04.4 amd 64
- **Hostname:** purush-vm

Installation Steps

1. **Verify Server Information:**

```
lsb_release -d
hostname -I
hostname
```

These commands check the distribution information and the current IP address and hostname of the server.

2. **Download and Extract Focalboard:**

```
VER=$(curl -s https://api.github.com/repos/mattermost/focalboard/releases/latest | grep tag_name | cut -d "" -f 4)
wget https://github.com/mattermost/focalboard/releases/download/${VER}/focalboard-server-linux-amd64.tar.gz
tar -xvzf focalboard-server-linux-amd64.tar.gz
mv focalboard /opt
```

We download the latest version of Focalboard, extract it, and move the files to the **/opt** directory.

3. **Install Dependencies:**

```
apt install nginx postgresql postgresql-contrib -y
```

This installs Nginx (a web server) and PostgreSQL (a database server) along with its contrib package.

4. **Configure Nginx:**
    - Remove default site configuration:

```
rm /etc/nginx/sites-enabled/default
```

This command removes the default Nginx site configuration to avoid conflicts.

- Create and edit the Focalboard Nginx configuration:

```
nano /etc/nginx/sites-available/focalboard
```

This opens a text editor to create a new Nginx configuration file for Focalboard.

- Add the following configuration to the file:

```
upstream focalboard {
  server localhost:8000;
  keepalive 32;
}

server {
  listen 80 default_server;
  server_name ubuntu.example.com;

  location ~ /ws/* {
     proxy_set_header Upgrade $http_upgrade;
     proxy_set_header Connection "upgrade";
     client_max_body_size 50M;
     proxy_set_header Host $http_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-Frame-Options SAMEORIGIN;
    proxy_buffers 256 16k;
    proxy_buffer_size 16k;
    client_body_timeout 60;
    send_timeout 300;
    lingering_timeout 5;
    proxy_connect_timeout 1d;
    proxy_send_timeout 1d;
    proxy_read_timeout 1d;
    proxy_pass http://focalboard;
  }

  location / {
    client_max_body_size 50M;
    proxy_set_header Connection "";
    proxy_set_header Host $http_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-Frame-Options SAMEORIGIN;
    proxy_buffers 256 16k;
    proxy_buffer_size 16k;
    proxy_read_timeout 600s;
    proxy_cache_revalidate on;
    proxy_cache_min_uses 2;
    proxy_cache_use_stale timeout;
    proxy_cache_lock on;
    proxy_http_version 1.1;
    proxy_pass http://focalboard;
  }
}
```

This configuration sets up Nginx to reverse proxy requests to the Focalboard server.

- Enable the new site:

```
ln -s /etc/nginx/sites-available/focalboard
/etc/nginx/sites-enabled/focalboard
```

This creates a symbolic link to enable the Focalboard site configuration in Nginx.

5. **Set Up PostgreSQL Database:**
    - Switch to the PostgreSQL user:

```
sudo --login --user postgres
psql
```

This logs into the PostgreSQL server as the **postgres** user and opens the PostgreSQL command line.

    - Create the database and user:

```
CREATE DATABASE boards;
CREATE USER boardsuser WITH PASSWORD 'Password';
\q
exit
```

We create a new database named **boards** and a user **boardsuser** with a password. Then we exit the PostgreSQL prompt and the **postgres** user session.

6. **Configure Focalboard:**
    - Edit the Focalboard configuration file:

```
nano /opt/focalboard/config.json
```

This opens the Focalboard configuration file for editing.

    - Update the database configuration:

```
"dbtype": "postgres",
```

```
"dbconfig":
"postgres://boardsuser:Password@localhost/boards?sslmode=disable&connect
_timeout=10",
```

We configure Focalboard to use PostgreSQL with the credentials we set up earlier.

7. **Create a Systemd Service for Focalboard:**
    - Create and edit the service file:

```
nano /lib/systemd/system/focalboard.service
```

This opens a new systemd service file for editing.

- Add the following configuration:

```
[Unit]
Description=Focalboard server

[Service]
Type=simple
Restart=always
RestartSec=5s
ExecStart=/opt/focalboard/bin/focalboard-server
WorkingDirectory=/opt/focalboard

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

This configuration sets up Focalboard to run as a service that restarts automatically on failure.

- Reload Systemd, start and enable the Focalboard service:

```
systemctl daemon-reload
```

```
systemctl start focalboard.service
systemctl enable focalboard.service
```

We reload the systemd daemon to recognize the new service, start the Focalboard service, and enable it to start on boot.

8. **Reload Nginx to Apply the New Configuration:**

```
systemctl reload nginx
```

This reloads Nginx to apply the new configuration changes.

9. **Access Focalboard:** Open your web browser and navigate to **http://purush-vm** to access the Focalboard application.

Summary

In this guide, we set up Focalboard on an Ubuntu 20.04 server. We started by downloading and installing the necessary software, including Nginx and PostgreSQL. We configured Nginx to reverse proxy to the Focalboard server, set up a PostgreSQL database and user, configured Focalboard to connect to the database, and finally created a systemd service to manage the Focalboard server process. After completing these steps, Focalboard is accessible via the configured domain.