**Self Hosting on Personal Server:**

1. **Set Up Your Personal Server:**
   * Choose a machine to act as your server (can be a physical server or a dedicated machine).
   * Install a Linux distribution (e.g., Ubuntu Server).
2. **Get a Static IP Address:**
   * Assign a static IP address to your server for consistent access.
3. **Configure Router Port Forwarding:**
   * If your server is behind a router, set up port forwarding to forward external traffic to your server's internal IP address.
4. **Purchase a Domain Name:**
   * Buy a domain name from a registrar.
5. **Configure Domain and DNS:**
   * Update DNS settings on your domain registrar's website to point to your static IP address.
6. **SSL Certificate:**
   * Obtain an SSL certificate (e.g., Let's Encrypt).
   * Configure your web server (e.g., NGINX) to use SSL.
7. **Install Required Software:**
   * Install Python and Docker on your server.
8. **Deploy Django App:**
   * Transfer your Django app code to your server.
   * Set up environment variables and configure Django settings.
   * Build and run Docker containers for your app.
9. **Configure Reverse Proxy:**
   * Configure your web server (e.g., NGINX) as a reverse proxy to forward requests to your Django app.
10. **Start and Monitor the App:**
    * Set up process management (e.g., systemd) to ensure your app runs continuously.
    * Monitor logs and error reports to ensure smooth operation.

**Note:** Here, everything has to be done by the internal maintenance guys.

There is a risk of downtime during server restart.

The server should be up and running 24/7

In house SMTP server from Jürgenhake

**Traditional Hosting Provider:**

1. **Purchase Hosting Plan:**
   * Choose a hosting provider and purchase a suitable hosting plan.
2. **Get a Domain Name:**
   * Buy a domain name from the hosting provider or a separate registrar.
3. **Configure Domain and DNS:**
   * Follow the hosting provider's instructions to configure DNS settings for your domain.
4. **SSL Certificate:**
   * Some hosting providers offer integrated SSL solutions. Follow their instructions to enable SSL for your domain.
5. **Install Required Software:**
   * Check if Python and Docker are supported by your hosting provider.
   * Follow their documentation to install necessary software.
6. **Deploy Django App:**
   * Follow the hosting provider's documentation for deploying Django apps.
   * Some providers may have specific guidelines for containerized apps.
7. **Start and Monitor the App:**
   * Follow the provider's guidelines to start and manage your Django app.
   * Monitor logs and performance metrics provided by the hosting platform.

**Note:** Here, the most difficult steps like 2, 3 and 4 are done by the providers.

There is no risk of downtime and the server will be up and running 24/7

Server maintenance is completely carried out by the providers

Many hosting providers include email services (SMTP) as part of their plans