This was written by chat gpt with the following prompt, the speeling mistakes have been left deliberately because they were included in the original prompt.  
  
*“Right its time for me to move the bot from Windows to a permanent linux server. Please write me a step by step deployment plan with the following considerations. It should be complete enough to be completed by a student with no prior knowledge. The bot needs to auto start with the OS The OS needs to be ubuntu server (latest LTS) The deployment plan should include cloning the repository from git (which does not include API keys) The plan should include reviewing the context and config, and committing any changes back to git.”*  
Here was its response

------------------------------------------------------------------------------------------------------------------------------------  
Deploying your Discord bot to a Linux server (Ubuntu) and setting it up for auto-start involves several steps. Below is a detailed plan, assuming you have a basic understanding of using the command line and git.

**Step 1: Setting Up the Ubuntu Server**

1. **Install Ubuntu Server**:
   * Download the latest LTS version of Ubuntu Server from the [official website](https://ubuntu.com/download/server).
   * Follow the installation instructions provided by Ubuntu.
2. **Update and Upgrade**:
   * Once Ubuntu is installed, open a terminal.
   * Run **sudo apt update** and **sudo apt upgrade** to ensure all packages are up to date.
3. **Install Git**:
   * Install Git with **sudo apt install git**.
4. **Install Python**:
   * Install Python (if not already installed) using **sudo apt install python3 python3-pip**.

**Step 2: Cloning the Repository and Setup**

1. **Clone Your Git Repository**:
   * Navigate to the directory where you want to clone the bot, e.g., **cd /home/username/**.
   * Clone your repository: **git clone https://your-repository-url.git**.
   * Navigate into the cloned directory: **cd your-repository-name**.
2. **Install Dependencies**:
   * If you have a **requirements.txt** file, install dependencies using **pip3 install -r requirements.txt**.
3. **Set Up API Keys**:
   * Since API keys are not in your repository, you'll need to add them manually.
   * Open **config.py** or wherever you store your keys: **nano config.py**.
   * Add your Discord bot token and any other necessary API keys or configuration data.
   * Save and close the file.

**Step 3: Testing the Bot**

1. **Run the Bot**:
   * Run your bot to make sure it works: **python3 bot.py** (replace **bot.py** with your main Python file name).
   * Test if the bot is working as expected in your Discord server.

**Step 4: Setting Up the Bot to Auto-Start**

1. **Create a Service File**:
   * Create a new service file for your bot in **/etc/systemd/system/**: **sudo nano /etc/systemd/system/discordbot.service**.
2. **Edit the Service File**:
   * Add the following content, adjusting paths and names as necessary:

makefileCopy code

[Unit] Description=Discord Bot Service After=network.target [Service] Type=simple User=username WorkingDirectory=/home/username/your-repository-name ExecStart=/usr/bin/python3 /home/username/your-repository-name/bot.py Restart=on-failure [Install] WantedBy=multi-user.target

* + Replace **username** with your actual username and **your-repository-name** with the name of your bot's directory.
  + Replace **bot.py** with the name of your bot's main Python script.

1. **Enable and Start the Service**:
   * Reload the systemd manager configuration: **sudo systemctl daemon-reload**.
   * Enable the service: **sudo systemctl enable discordbot.service**.
   * Start the service: **sudo systemctl start discordbot.service**.
   * Check the status: **sudo systemctl status discordbot.service**.

**Step 5: Committing Changes to Git**

1. **Commit Configurations**:
   * If you made changes that need to be tracked (excluding API keys), stage them: **git add .**
   * Commit the changes: **git commit -m "Update configurations"**.
   * Push the changes back to the repository: **git push origin main**.
2. **Security Note**:
   * Ensure that you do not commit sensitive information like API keys to the Git repository.

**Step 6: Maintaining the Bot**

* Periodically pull updates from the git repository to the server if you make changes to your bot's code.
* Use **git pull** in the bot's directory to fetch and merge changes from the repository.

This plan should guide a student with basic knowledge through setting up and running a Discord bot on an Ubuntu server. Remember to replace placeholders (like file paths and repository URLs) with actual values relevant to your setup.

There were some details missing so this was added

*“Ok a few things The OS is already installed, the student will be provided an admin login/ The install location should be /opt/DerbyGPT There is no requirements.txt file, please provide instructions to install a requriements for python discord bot that uses discord and OpenAI The openAi tokens and discord tokens are stored in tokens.py, these will be provided. Instructions shoudl include checkign the gitignore that the tokens are ignored. Can you produce the deployment plan in a file, like pdf, doc, or txt”*

Its response was as follows

-----------------------------------------------------------------------------------------------------------------------------------

Sure, I'll outline a revised deployment plan based on your specifications and provide it in a text format that you can easily copy into a document of your choice.

**Discord Bot Deployment Plan**

**Preparation:**

* Ensure you have administrative access to the Ubuntu server.
* The bot will be installed in **/opt/DerbyGPT**.

**Step 1: Install Dependencies**

1. Open a terminal on the Ubuntu server.
2. Update the package list: **sudo apt update**.
3. Install Python and Pip (if not already installed):

Copy code

sudo apt install python3 python3-pip

**Step 2: Clone the Git Repository**

1. Navigate to the **/opt** directory:

bashCopy code

cd /opt

1. Clone the repository into the **DerbyGPT** directory (replace **your-repository-url** with the actual URL):

bashCopy code

sudo git clone https://your-repository-url.git DerbyGPT

1. Change the ownership of the **DerbyGPT** directory to the current user (replace **username** with your username):

bashCopy code

sudo chown -R username:username DerbyGPT

1. Navigate to the newly cloned directory:

bashCopy code

cd DerbyGPT

**Step 3: Install Python Dependencies**

1. Install the **discord.py** library:

Copy code

pip3 install discord.py

1. Install the OpenAI Python library:

Copy code

pip3 install openai

**Step 4: Set Up API Tokens**

1. Ensure **tokens.py** containing the Discord and OpenAI tokens is present in the **DerbyGPT** directory.
2. Verify that **tokens.py** is listed in the **.gitignore** file to prevent it from being tracked by Git:
   * Open **.gitignore** with a text editor, e.g., **nano .gitignore**.
   * Confirm that there's a line containing **tokens.py**.

**Step 5: Test the Bot**

1. Run the bot to ensure it's working correctly:

bashCopy code

python3 bot.py # Replace 'bot.py' with your main bot script.

1. Verify that the bot is functioning as expected in your Discord server.

**Step 6: Configure the Bot to Auto-Start**

1. Create a systemd service file:

bashCopy code

sudo nano /etc/systemd/system/discordbot.service

1. Add the following content to the file:

makefileCopy code

[Unit] Description=Discord Bot Service After=network.target [Service] Type=simple User=username # Replace with your username WorkingDirectory=/opt/DerbyGPT ExecStart=/usr/bin/python3 /opt/DerbyGPT/bot.py # Replace 'bot.py' with your main bot script Restart=on-failure [Install] WantedBy=multi-user.target

1. Enable and start the service:

bashCopy code

sudo systemctl daemon-reload sudo systemctl enable discordbot.service sudo systemctl start discordbot.service

1. Check the service status:

luaCopy code

sudo systemctl status discordbot.service

**Step 7: Updating the Bot**

1. To update the bot, navigate to the **/opt/DerbyGPT** directory and pull changes from Git:

bashCopy code

cd /opt/DerbyGPT git pull

1. Restart the bot service to apply changes:

Copy code

sudo systemctl restart discordbot.service

**End of Deployment Plan**