Run a Telegram Bot 24/7 on a VPS: Python and systemd Step by Step

Servers Without a Control Panel · https://www.hyyat.com/en/knowledgebase/run-telegram-bot-on-vps/

In shortA bot on your own computer stops when you switch it off. On a VPS it runs **24/7**. Put the bot in a folder with a Python virtual environment, keep the **token** in a protected file, then create a **systemd service** that starts it on boot and restarts it if it stops. A **2GB** plan is enough for most bots.



Telegram bots take orders, answer customers and send alerts. A common problem is running the bot on a personal computer, so it stops when the internet drops or the machine is turned off. A server fixes that: it is always online, and systemd restarts the bot if it crashes.

Choosing resources

Use	Suitable plan	Why
A reply or alert bot	VPS 2GB	A light Python bot usually uses under 200 MB

Use	Suitable plan	Why
Several bots or a database	VPS 4GB	Room for several services and MySQL or PostgreSQL
A bot using local AI models	VPS 16GB	Models need a lot of memory

1. Prepare the server and a user

Do not run the bot as root. Create a dedicated user:

```
apt update && apt install -y python3 python3-venv
adduser --system --group --home /opt/mybot mybot
```

2. Copy the code and install libraries

```
cd /opt/mybot
python3 -m venv venv
./venv/bin/pip install python-telegram-bot
# copy bot.py into /opt/mybot
```

3. Store the token safely

Never hard-code the token. Put it in an environment file only the bot user can read:

```
echo "BOT_TOKEN=your-token-here" > /opt/mybot/.env
chown -R mybot:mybot /opt/mybot
chmod 600 /opt/mybot/.env
```

In your code, read it with `os.environ["BOT_TOKEN"]`.

4. Create a systemd service

File `/etc/systemd/system/mybot.service`:

```
[Unit]
Description=Telegram bot
After=network-online.target
Wants=network-online.target
```

```
[Service]
User=mybot
WorkingDirectory=/opt/mybot
EnvironmentFile=/opt/mybot/.env
ExecStart=/opt/mybot/venv/bin/python bot.py
Restart=always
RestartSec=5

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

```
systemctl daemon-reload
systemctl enable --now mybot
systemctl status mybot
journalctl -u mybot -f
```

You can generate the service file with your own options using our systemd service file generator.

Polling or webhook?

	Polling	**Webhook**
How it works	The bot keeps asking Telegram	Telegram sends updates to your URL
Needs a domain and HTTPS	No	Yes
Suits	Most bots and new projects	High-traffic bots

A mistake to avoid: Pushing the token file to GitHub with your code. Whoever gets the token fully controls the bot. If that happens, create a new token in BotFather at once.

At Hyyat Host: A VPS keeps your bot running around the clock: VPS plans, with our own control panel for power, console and snapshots, and Arabic and English support.

Frequently asked questions

Can I run the bot on shared hosting?

Usually not, because shared hosting does not allow programs that run constantly in the background. A permanent bot needs a VPS.

How do I update the bot code?

Copy the new files into the bot folder, then run `systemctl restart mybot`.

Can I run more than one bot?

Yes. Create a separate folder and a systemd service with a different name for each bot.

Related guides

- systemd Service File Generator
- New VPS Setup Commands, Step by Step
- Firewall: Control the Open Ports on Your VPS
- When Should You Move from Shared Hosting to a VPS? 7 Clear Signs

All guides for servers without a panel →

Need this service from Hyyat Host?Our team handles it for you, with support replying to tickets within 24 hours at most.

Server Management