

How to setup ED Recon

Bot for Discord

This guide explains how to setup the [ED Recon Discord bot](#).

If you get stuck or have any questions, feel free to join [EDR central](#), the community server for EDR with access to the bot, real-time alerts and troubleshooting support.

Pre-requisites

- [Discord](#) installed
- A discord server for which you have the `Manage Server` permission. This is required by Discord in order to invite any bot, EDR's included.
- Read and understood EDR's [privacy policy](#) and [terms of services](#).
 - Proceeding any further implies that you, as well as the members of your server, understand and agree to the privacy policy and the terms of services.
 - Feel free to drop by [EDR central](#) if you have any questions or concerns.

Inviting the bot

Follow these steps:

1. [Invite](#) the bot to your server.
2. **Tag the bot with a role**, e.g. `bot` and use that role to customize which channels the bot gets access to. See the privacy considerations section for more details.

Note: you might not get access to the full functionality right out of the box. Your application will be reviewed within a few days, and you might be contacted by LeKeno#8484.

Privacy considerations

All discord bots that can respond to certain commands or keywords, work the same. For any channel that the bot has `Read Messages` permission enabled, whenever someone sends a message, the bot will receive an event corresponding to that message with information such as the name of the sender, the message itself, etc. This is inherent to how the Discord API works. It is NOT possible to only receive messages that are designated bot commands, and is up to the bot to filter out messages that are not among the recognized commands.

If you are curious, you can check [EDR's source code](#) to see how it filters out messages that aren't recognized commands (in particular, see the `process` function in [edrbot.js](#)). The source code of the [EDR plugin for EDMC](#) is [also available here](#), and since it's written in Python you can also scrutinize the running code and confirm that it's identical to the source code published on github. Finally, EDR has been [reviewed and praised by Frontier](#),

confirming that its use of the player journal API is within Frontier's rules and terms of services.

Ultimately, you don't have to trust me, nor other bot authors. As long as you keep a bot in a dedicated channel that's only meant for issuing bot commands, you have absolutely nothing to worry about. See the next sections to learn more.

That said, if this sounds too complicated, consider [joining EDR central](#) for further help, or to simply get access to a hassle-free option with:

- access to the bot
- a real-time feed of outlaws sightings
- discussion channels about Elite and EDR
- and more...

Restricting the bot to a dedicated channel

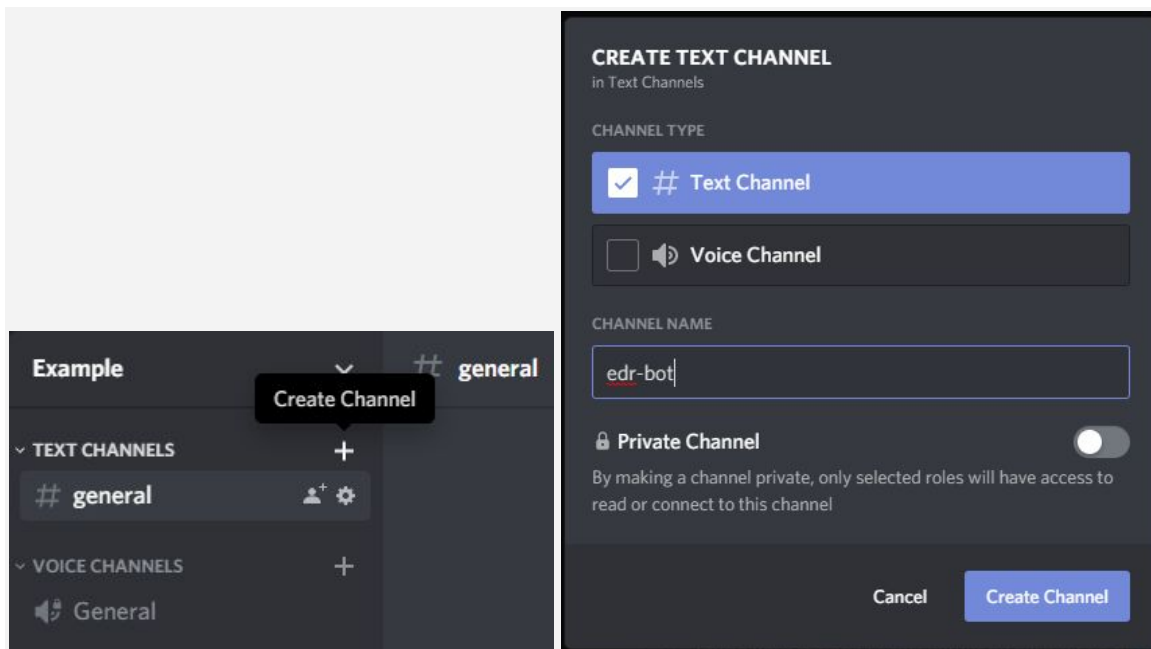
Restrictive-by-default server setups

In general, to avoid being trolled by randoms or the usual culprits from the Elite Dangerous community, you should setup your whole server to be restrictive by default:

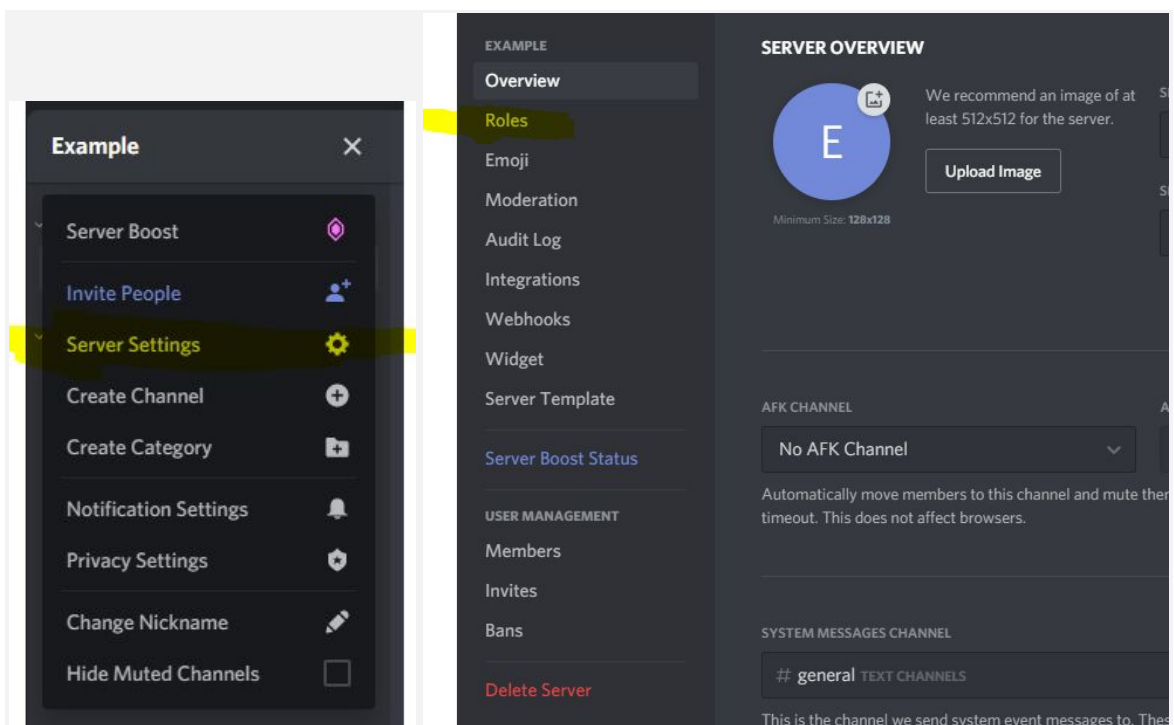
- @everyone role should NOT have any permission at all.
- In your "welcome" channel, override @everyone's defaults to only allow the basics, i.e. read messages, send messages and read message history.
- Create and assign other roles (e.g. member, leadership, guest) to control who gets access to what, and who can do what. Most roles should not have any default permissions.
- Add the relevant roles to specific channels, and tweak the permissions of each role in each channel or sections.

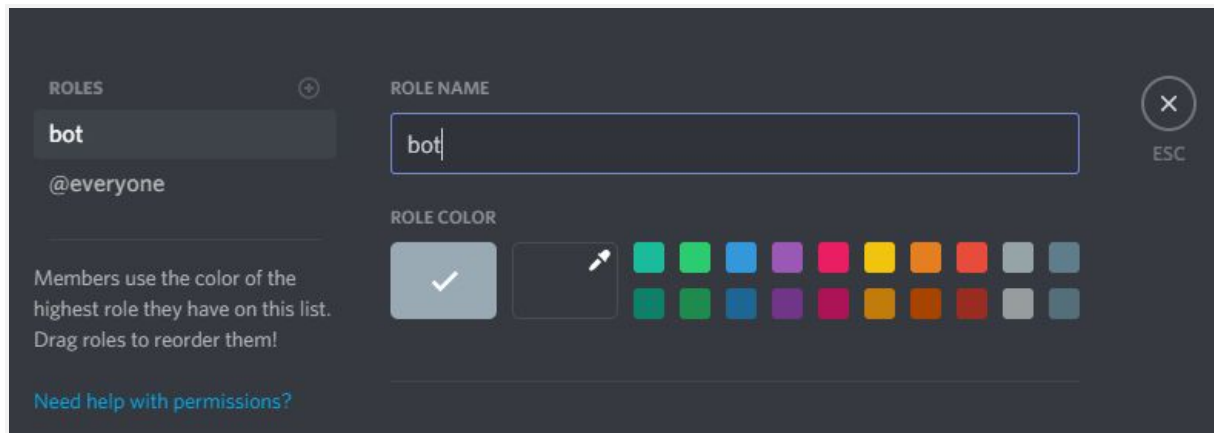
With such a server setup, here is how you can restrict the bot to a dedicated channel:

1. Create a text channel whose sole purpose is to issue commands to the bot, e.g.
`edr-bot`

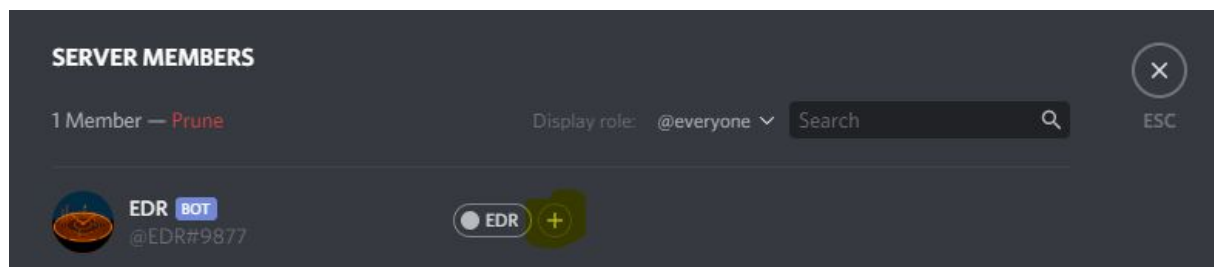
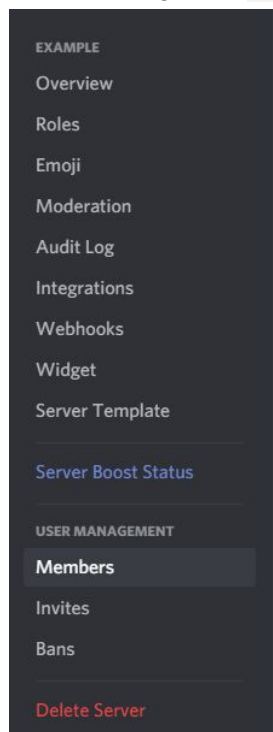


2. Create a role for the bot without any default permissions, e.g. bot

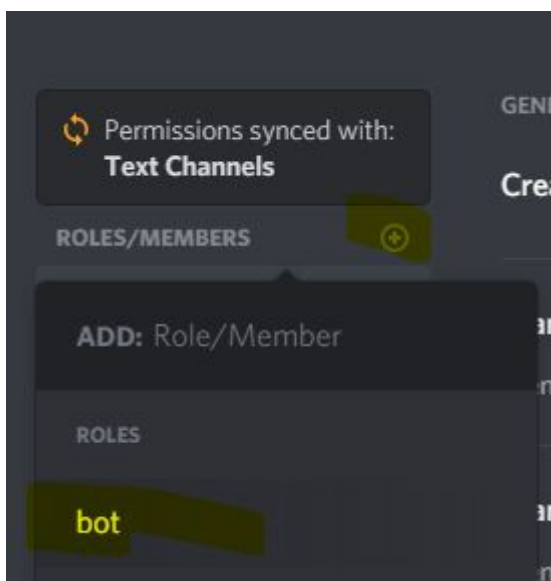
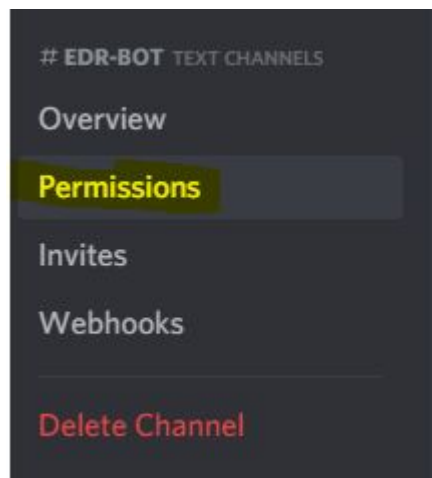
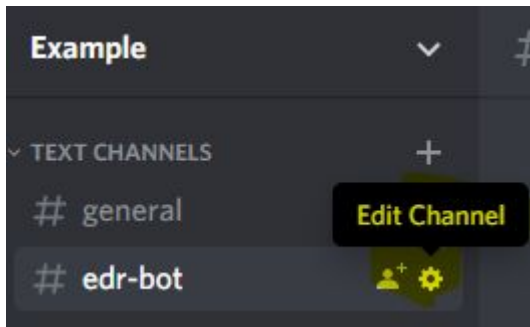




3. Assign the bot role to EDR



4. Add the bot role to the edr-bot channel.



5. Change the permissions for the bot role in that channel as follows:

TEXT PERMISSIONS

Read Messages	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Send Messages	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Send TTS Messages	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Members with this permission can send text-to-speech messages by starting a message with /tts. These messages can be heard by everyone focused on the channel.			
Manage Messages	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Members with this permission can delete messages by other members or pin any message.			
Embed Links	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Attach Files	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Read Message History	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Other server setups

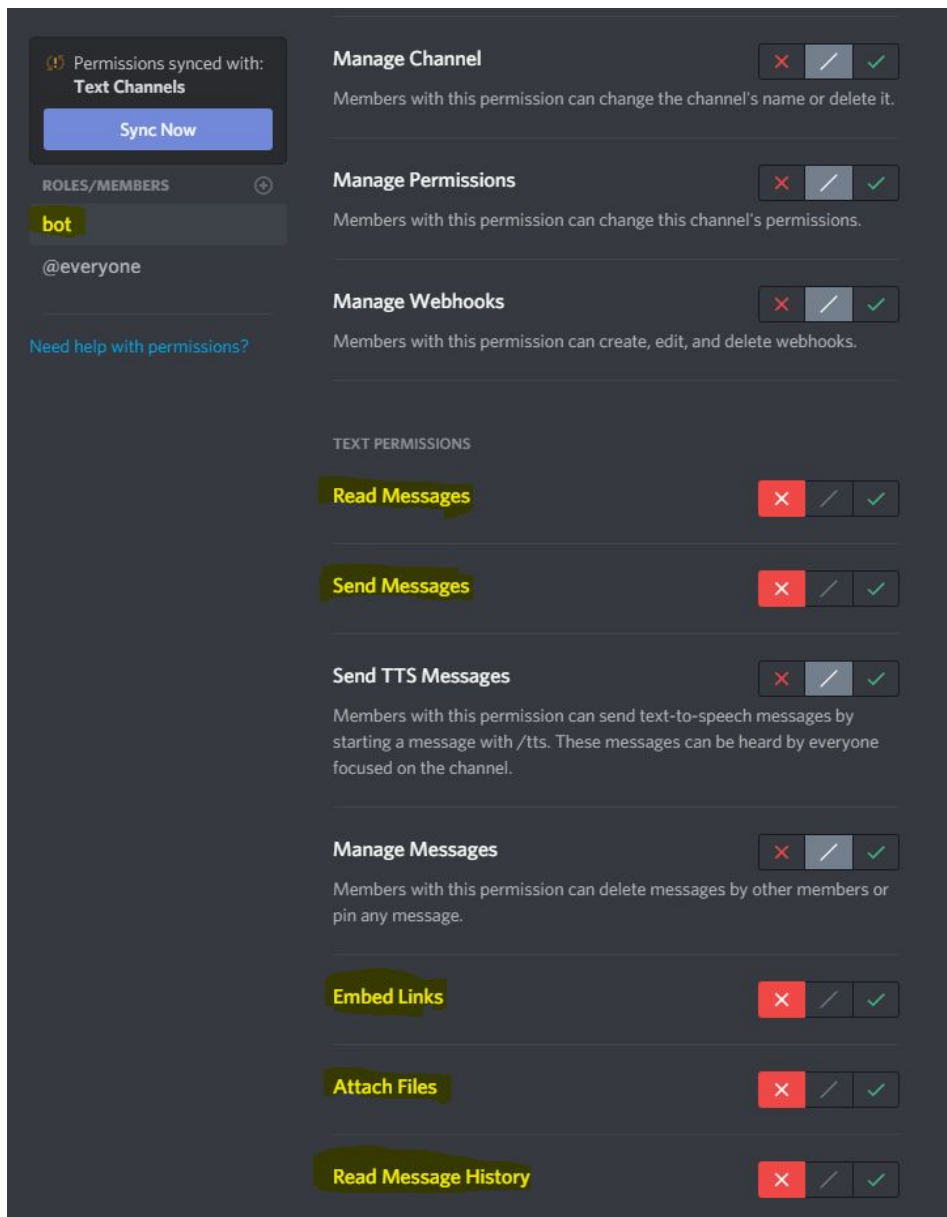
If you can't use the setup detailed above, here is how you can still restrict the bot to a dedicated channel:

1. Create a text channel whose sole purpose is to issue commands to the bot, e.g. `edr-bot`
2. Create a role for the bot, e.g. `bot`
3. Assign the `bot` role to EDR
4. Add the `bot` role to the `edr-bot` channel
5. Change the permissions for the bot role in that channel as follows:

TEXT PERMISSIONS

Read Messages	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Send Messages	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Send TTS Messages	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Members with this permission can send text-to-speech messages by starting a message with /tts. These messages can be heard by everyone focused on the channel.			
Manage Messages	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Members with this permission can delete messages by other members or pin any message.			
Embed Links	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Attach Files	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Read Message History	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

6. For all the other channels, edit each channel's permissions in the following way:
 - a. add the `bot` role
 - b. override the permissions of that role as follows:



Verification

To find out if you've successfully setup the EDR bot to only have access to a dedicated channel, do the following:

1. Send `!ping` in any channel that the EDR bot should NOT have access to. Expected outcome: no response from the EDR bot.
2. Send `!ping` in a channel the EDR bot should have access to. Expected outcome: EDR bot responds with `pong!`

About the permissions required by EDR

Here is a short explanation of why the EDR Discord bot requires each permission:

- `Read Messages`, `Read Message History`: the bot needs this to find out when a user requires its services in a given channel, e.g. to find out that someone is issuing a command such as `!w lekeno`
- `Send Messages`: the bot needs this to respond to a command.
- `Embed Links`: the bot uses fancy embeds for visual presentation of the information (e.g. commander profile) and links to various additional resources (e.g. Inara, EDSM)
- `Attach Files`: the bot uses this to attach images to the embeds (e.g. history of scans, bounties, sightings, etc.)