FiveM

Complete guides for setting up and managing your FiveM server.

Getting Started

Starting the Server

Start your server from the panel by clicking **Start**. The server console will display the startup logs.

> If this is the first time the server has been started, the startup process may take slightly longer while the server initializes.

Accessing txAdmin

If **txAdmin** is enabled, the server will automatically start the txAdmin management interface.

You can access it using the txAdmin address shown in the **Network** tab of your server panel.

```
http://SERVER_IP:TxAdminPort
```

> txAdmin may take a few seconds to become available after the server starts.

When accessing txAdmin for the first time you will be prompted to create an administrator account and complete the server setup.

Important Server Settings

Your server has several configurable startup settings available in the panel.

FiveM License Key

Your server's CFX license key. This is required for the server to start.
The key is automatically passed to the server during startup.

FXServer Artifact Version

Controls which FXServer artifact build will be used by the server.
The artifact version determines which version of the FiveM server runtime is installed.

Valid values include:

latest – Automatically install the newest available artifact (recommended)
A specific artifact number – Installs that exact server build

Enable txAdmin

Controls whether the server starts with **txAdmin** enabled.

- **On** – Start using txAdmin
- **Off** – Start the server directly using the configuration file

> Most users should keep txAdmin enabled as it provides the easiest way to manage your server.

txAdmin Port

The port used for the txAdmin web interface.
You normally do not need to change this unless you want to change the port, must be one of your allocated ports.

Git Repository URL

If set, the server will automatically pull files from the specified Git repository every time the server starts.

Any files in the target folder will be overwritten to match the repository.

> Local file changes will be discarded when Git deployment is enabled.

Git Branch

The branch that will be used when pulling the repository.

Default: **main**

Git Target Path

The folder where the repository will be deployed.

Default:

```
/home/container/nontxserver
```

Git Username and Git Token

If you are using a private repository, these fields allow the server to authenticate when pulling from Git.

> These values are not required when using public repositories.

Server File Locations

The server uses the following folder structure:

```
/home/container
├── .internal/        (server runtime files)
├── nontxserver/      (server files when not using txAdmin)
```

> Most users will manage their server through txAdmin and will not need to interact with these folders directly.

If the Server Fails to Start

Check the server console for error messages. Startup errors are usually caused by:

- Invalid server configuration
- Missing or incorrect license key
- Errors in resources or scripts

Understanding the FiveM Server Layout

Your FiveM server stores its files inside the `/home/container` directory. This directory contains both the server runtime files and your actual server files.

Main Server Directory

```
/home/container
├── .internal/        (FXServer runtime files)
├── nontxserver/      (server files when not using txAdmin)
```

Most users will interact with files through **txAdmin** rather than editing these folders directly.

> Files inside `.internal` are part of the server runtime and should not be modified.

The nontxserver Folder

This folder contains the server files used when the server is started **without txAdmin**.

Typical contents include:

```
/home/container/nontxserver
├── server.cfg
├── resources/
└── other server files
```

If you are deploying your server using **Git**, the repository will normally be placed in this folder.

The server.cfg File

The `server.cfg` file is the main configuration file for your server.

This file controls:

- Server name and description
- Resource loading
- Permissions and settings
- Database configuration

When running without txAdmin, this file is loaded automatically from:

```
/home/container/nontxserver/server.cfg
```

The Resources Folder

Server scripts and mods are placed inside the `resources` folder.

```
/home/container/nontxserver/resources
```

Each resource must be started inside your `server.cfg` file.

Using Git Deployment

If **Git deployment** is enabled in the server settings, the server will pull files from the configured repository every time the server starts.

The repository is deployed to the folder defined in **Git Target Path**. By default this is:

```
/home/container/nontxserver
```

> When Git deployment is enabled, local file changes inside the target folder will be overwritten on server start.

txAdmin Managed Servers

When **txAdmin** is enabled, the server configuration and resources are typically managed directly through the txAdmin interface.

txAdmin may create additional folders to store server profiles and configuration data.

>

> For most users it is recommended to manage resources and server configuration through txAdmin

Accessing and Setting Up txAdmin

txAdmin is the web-based management interface used to configure and control your FiveM server.

Accessing txAdmin

After starting your server, txAdmin will become available automatically.

You can access it from the **Network** tab in your server panel using the txAdmin port.

```
http://SERVER_IP:txAdminPort
```

> txAdmin may take a few seconds to start after the server begins running. If the page does not load immediately, wait a moment and refresh the page.

First Time Setup

The first time you open txAdmin you will be guided through a setup process.

During this process you will:

- Create a txAdmin administrator account
- Select a server configuration
- Configure your server settings

> Your FiveM license key is automatically provided to the server by the hosting platform. You do not need to enter it during setup.

Selecting a Server Configuration

txAdmin provides several **recipes** for installing common server frameworks such as ESX or QBCore.

You can either:

- Use a recipe to automatically install a framework
- Deploy an existing server configuration

> Recipes install the framework, dependencies, and recommended resources automatically.

Database Configuration

Many server recipes require a database connection. During the setup process txAdmin will ask for database information.

Some fields may already be pre-filled automatically.

The following values are typically required:

- **Database Host**
- **Database Port**
- **Database Name**
- **Database Username**
- **Database Password**

> Make sure to create a database in the game control panels database tab if your FiveM server requires one!

> The database host is usually the same as your server IP But you can find all your database information from the game control panels database page.

> The database name, username, and password must match the credentials created for your database. If these values are incorrect the server will fail to connect to the database.

Completing Setup

Once the configuration process is finished, txAdmin will automatically start your server.

You will then be able to manage your server through the txAdmin dashboard.

From the dashboard you can:

- Start and stop your server
- Install and manage resources
- View logs and console output
- Manage administrators

> Your server is now ready for further configuration and resource installation.

Installing Resources on Your FiveM Server

Resources are the scripts and mods that add features to your FiveM server.

Before You Start

Resources can be installed in different ways depending on how you manage your server.

- If you are using **txAdmin recipes**, many resources may already be installed for you
- If you are using **Git deployment**, resources should usually be added to your Git repository instead of uploaded manually
- If you are managing the server manually, resources are usually placed inside your server's `resources` folder

> If Git deployment is enabled and the resource folder is inside your Git target path, manual file changes will be overwritten the next time the server starts.

Where to Place Resources

When running without txAdmin, resources are typically stored in:

```
/home/container/nontxserver/resources
```

If you are using txAdmin, the resource location depends on how your server was deployed.

> If you are unsure where your active resource folder is, check your `server.cfg` or the resource paths configured by txAdmin.

Uploading a Resource

To install a resource manually:

1. Download the resource files

2. Upload the resource folder to your server's `resources` directory
3. Make sure the resource stays inside its own folder
4. Edit your `server.cfg` to start the resource

Example resource layout:

```
/home/container/nontxserver/resources/my_resource
├── fxmanifest.lua
├── client.lua
├── server.lua
```

> The resource folder itself must be uploaded, not just the files inside it.

Uploading Resources Using SFTP

> The web file manager does not support uploading folders directly.

You have two options when uploading resources:

- Upload the resource as a **.zip** file and extract it using the **Unarchive** option in the file manager
- Upload the resource using **SFTP**, which allows you to transfer folders directly

> SFTP is recommended when uploading larger resources or full server packs.

To connect using SFTP:

1. Open the **SFTP** tab in your server panel
2. Use the provided connection details with an SFTP client such as **FileZilla** or **WinSCP**
3. Upload your resource folder to the server's `resources` directory

Starting a Resource

After uploading the resource, add it to your `server.cfg` so the server starts it automatically.

Example:

```
ensure my_resource
```

Add one `ensure` line for each resource you want to load.

> Some older resources may state `start resource_name`, but `ensure` is recommended.

Restarting the Server

Once the resource has been uploaded and added to your configuration, restart your server.

You can also start a resource manually from the console using:

```
ensure my_resource
```

Installing Resources with Git

If your server uses Git deployment, resources should normally be added to your Git repository instead of uploaded in the file manager.

Typical workflow:

1. Add the resource to your repository
2. Commit and push the changes
3. Restart the server

On startup, the server will pull the latest version of the repository and update the files automatically.

> If Git deployment is enabled, manually uploaded resources inside the Git target folder will be removed or overwritten if they are not present in the repository.

Common Issues

If a resource does not load, check the following:

- The folder name matches the name used in `ensure`
- The resource contains a valid `fxmanifest.lua` or `__resource.lua`
- All required dependencies are installed
- The files were uploaded to the correct resources folder

>

If the server fails to start after adding a resource, check the console for errors. A broken resource, missing dependency, or invalid configuration can stop the server from loading correctly.

Deploying Your Server Using Git (Optional - Advanced Feature)

Your server supports automatic deployment from a Git repository. When configured, the server will pull the latest version of the repository every time the server starts.

This allows you to manage your server files using version control and update your server by pushing changes to your repository.

> When Git deployment is enabled, local file changes inside the Git target folder will be overwritten when the server starts.

How Git Deployment Works

If a repository URL is configured, the server will perform the following steps during startup:

1. Connect to the configured Git repository
2. Pull the selected branch
3. Overwrite the contents of the Git target folder
4. Start the server using the updated files

This ensures the server always runs the latest version of your repository.

Configuring Git Deployment

Git deployment is configured using the startup variables in the server panel.

Git Repository URL

The URL of the repository that contains your server files.

Example:

```
https://github.com/example/my-fivem-server.git
```

Git Branch

The branch that should be deployed.

Default:

```
main
```

Git Target Path

The folder where the repository will be deployed.

Default:

```
/home/container/nontxserver
```

> Your `server.cfg` and resources should normally be located inside this folder when using Git deployment.

Git Username and Git Token

These fields are used when accessing private repositories.

Enter your Git username and a personal access token with repository access.

> These values are not required for public repositories.

Typical Workflow

Once Git deployment is configured, updating your server becomes very simple:

1. Make changes to your server files locally
2. Commit and push the changes to your repository
3. Restart the server

On startup, the server will pull the latest version of the repository and update the server files automatically.

Example Repository Layout

```
repository-root
├── server.cfg
├── resources/
│   ├── resource_one/
│   ├── resource_two/
│   └── resource_three/
```

Common Issues

If Git deployment fails, check the following:

- The repository URL is correct
- The branch name exists
- Git credentials are correct when using a private repository
- The server has permission to access the repository

> If Git deployment fails, the server may not start if required files such as `server.cfg` are missing.

Troubleshooting Server Startup Issues

If your server fails to start, the most useful information will always be found in the **server console**. The console shows the full startup process and any errors that occur.

> Always check the console logs first when diagnosing startup issues.

Common Causes

Invalid or Missing License Key

Your server requires a valid FiveM license key to start.

If the key is missing or invalid, the server may stop during startup.

Make sure the **FiveM License Key** variable is correctly set in the server panel.

> License keys can be generated at https://keymaster.fivem.net

Broken or Missing Resources

If a resource fails to load, it may stop the server from starting.

Common problems include:

- Missing files inside a resource
- A missing `fxmanifest.lua` or `__resource.lua`
- Incorrect resource names in `server.cfg`
- Missing dependencies

> If the server stops during resource loading, check the last resource listed in the console logs.

Incorrect server.cfg Configuration

Errors in your `server.cfg` file can prevent the server from starting.

Common issues include:

- Invalid configuration values
- Incorrect database settings
- Resources listed that do not exist

Check your `server.cfg` file and confirm all referenced resources exist.

Database Connection Errors

If your server uses a framework such as ESX or QBCore, a database connection is usually required.

If the database credentials are incorrect, the server may fail to start.

Verify the following values:

- Database host
- Database port
- Database name
- Database username
- Database password

> If the database cannot be reached, the framework may fail to load and stop the server.

Git Deployment Issues

If Git deployment is enabled, the server will pull files from the configured repository during startup.

Startup may fail if:

- The repository URL is incorrect
- The branch does not exist
- Git credentials are invalid
- The repository does not contain required server files

> Git deployment will overwrite the contents of the configured target folder during startup.

Still Having Issues?

If you are unable to identify the issue, review the server console logs and look for the first error message shown during startup.

> If the server fails repeatedly during startup, avoid repeatedly restarting it without reviewing the console logs. This can make diagnosing the problem more difficult.