🌟 Recommended and must-install
- 🔥 **Tssh & trzsz** - ssh Login and terminal file transfer
- 🔥 **Autojump** - Quick directory jump
- 🔥 **Saferm** - Safe delete file/folder
- 🔥 **Code** - Terminal editor

📋 User Guide
1. First time use
2. The Logic of GCR
3. Overview of GCR functions

📚 Appendix (All Functions)

📖 User Manual

🌟 Recommended and must-install

🔥 Tssh & trzsz - ssh Login and terminal file transfer

💡 **Function description**：

Trzsz tssh is a tool that reshapes the terminal ssh login method and the local-server file transfer method. The ultimate goal is to cooperate with the native ssh tool to achieve a more elegant and convenient server login, and at the same time give the terminal file visual transfer function.

- **server login**： Tssh can list all available server details, and you can interactively search and select the server you need to connect to.
- **file transfer**： Support file/folder transfer in the terminal: 1. Local to server - drag and drop (what you see is what you get); 2. Server to local - download files/folders

🔗 **Installation command**：

```
# Local computer (supports Mac OS, Ubuntu, Win, for more details please refer to
https://github.com/trzsz/trzsz-ssh self-installation)
install_tssh

# remote server
install_trzsz
```

🔗 **Usage**：

```
# Connect to server
tssh

# Upload files/folders
upload # Or drag and drop directly

# Download files/folders
download File1 Folder2
```

⚠️ Description:

- Tssh supports mainstream operating systems (Mac OS, Ubuntu, Win), but the interactive interface may not be called in virtual machines or win subsystems without visual interfaces.

- Tssh is compatible with native ssh and supports rich custom configuration solutions, greatly improving efficiency and experience. Please refer to the official website to read detailed configuration rules. Here is a template for reference (compatible with native ssh) `~/.ssh/config`:

```
# All servers forward port 59863 to the local machine (for use by the code command,
another highly recommended feature)
# All servers tssh supports drag and drop file upload
Host *
    RemoteForward 59863 127.0.0.1:52698
    #!! EnableDragFile Yes

# All servers starting with nscc are classified under the nus label and nscc label
Host nscc*
    #!! GroupLabels NUS
    #!! GroupLabels NSCC

# Log in node information and hide the server
Host nscc
    HostName aspire2a.nus.edu.sg
    User username
    ServerAliveInterval 60
    PreferredAuthentications publickey
    IdentityFile ~/.ssh/nus/nscc
    #!! HideHost yes

# Real login node (specify login to nus01 node)
Host nscc-nus01
    HostName asp2a-login-nus01
    User username
    IdentityFile ~/.ssh/nus/nscc
    ProxyJump nscc
```

🔥 Autojump - Quick directory jump

💡 **Function description**:

Autojump is an intelligent directory jumping tool that can quickly locate directories based on the frequency of directories you have visited. By learning your operating habits, it allows you to jump to commonly used paths by entering only part of the directory, greatly improving work efficiency.

- **Automatic learning path**： Directories with more visits will jump faster.

- **Minimalist commands**： Just j + the directory name you want to go to. The directory name does not need to be accurate. It supports fuzzy matching and does not need to find the path step by step.

🔗 **Installation command**：

```
install_autojump
```

🔗 **Usage**：

```
j Any fuzzy keywords for where you want to go
```

🔥 Saferm - Safe delete file/folder

💡 **Function description**：

Saferm is a safe deletion tool that replaces the native rm. You don't need to pay attention to whether you delete files or folders. del can delete them all and automatically put them into the designated recycle bin for easy retrieval, making it easy to delete files. More convenient and safer.

- **Easy to operate**： Whether it is a file or a folder, you can delete them all with just one command.

- **Safe and easy to recover**： del will be put into the designated recycle bin by default, and you can easily retrieve it in the recycle bin. When there are multiple files with the same name, the deletion time suffix will be automatically added, so all information is safe!

- **Smart space management**： In order to prevent the recycle bin from taking up too much disk space, del will check the file size when deleting. When the deleted file exceeds 10 G, you will be prompted whether to put it in the recycle bin. You can also use delf to manually delete permanently (delf is equivalent to native rm, but it also doesn't matter whether it's a file or a folder)

- **Easily empty the Recycle Bin**： empty_trash clears the recycle bin with one click and supports two modes: clear all contents of the recycle bin with one click and delete files older than 30 days.

🔗 **Installation command**：

```
install_safe_rm

# Optional (recommended): Set the path to the recycle bin (recommended to store it in .zshrc)
export SAFE_RM_TRASH=$HOME/.Trash
```

🔗 **Usage**：

```
# Delete files/folders (put in recycle bin)
del 文件名/文件夹名
# Delete files/folders (without putting them in the recycle bin)
delf 文件名/文件夹名
# Empty Recycle Bin
empty_trash
```

🔥 Code - Terminal editor

💡 **Function description**：

Code is a rmate-based tool that allows you to edit code/files in the terminal more intuitively, thus eliminating VI/VIM. Suitable for situations when you want to lightly edit files but don't want to use vim.

- **Automatically identify local and server**： The code will automatically identify whether the current terminal is local or remote and enable the corresponding logic to let you edit the file.

- **Local file editing**： Code will directly use sublime text vs code to open the current file/folder, save and close it after editing. There is no need to search for and open the corresponding file in the editor, or use vim to edit in the terminal.

- **Server file editing**： Code will automatically synchronize the remote file to local temporary storage, and use local sublime text vs code to open the file. After editing, just save and close, and the code will automatically be synchronized to the server. The entire synchronization process does not require any user operation, and is the same process as editing local files.

🔗 **Installation command**：

```
# server
install_discard_vim

# Modify the local configuration according to the server-side prompts, mainly including:
# 1. Vscode/sublime plug-in (used to support rmate)
# 2. tssh config port forwarding
```

🔗 **Usage**：

```
code FileName
```

📋 User Guide

1. First time use

- After installing GCR, it is recommended to install tssh locally and improve the ssh config configuration for a better experience

- Read the recommended must-installs and install the features you are interested in (strongly recommended to install them all)
- Organize the environment configuration of the original shell, organize the environment variables that need to be retained into a new file, and source the file in ~/.zshrc

2. The Logic of GCR

GCR is implemented based on oh-my-zsh, and customizes some functions of ohmyzsh, including terminal display themes, styles, activated plug-ins, etc., and enables automatic prompts and completion by default (fixes the bugs in the original version). In addition, the core functions mainly include GCR version management, ohmyshell, ohmytool.

The overall functional function follows the first, second and third grading standards. (As the version is updated, more first-level functions may be supported, but the logic of level-by-level calls will continue to be inherited) (⚠️Second-level functions are not yet implemented under ohmytool)

Second-level functions include various functional module functions subordinate to each first-level function, such as personalized functions for anaconda, pip and other modules; second-level functions can access all third-level functions;

Third-level functions include specific functional functions under each module to perform a specific function;

Users only need to memorize one-level functions to access all gcr functions!

3. Overview of GCR functions

L1 Function	Description
ohmyshell	gcr version management, an enhancement tool for common shell native operations
ohmytool	Install various tools/software with one click

L2 Function	Description	Affiliated Function
myupdate	Update gcr version	ohmyshell
mytool	Some common shell tools	ohmyshell
mypip	Extensions for python pip	ohmyshell
myenv	Extensions for the anaconda management environment	ohmyshell
mygpu	Common commands for gpu users	ohmyshell
mypermission	Manage user passwords, file permissions, etc.	ohmyshell
mygit	Extensions for git operations	ohmyshell

L2 Function	Description	Affiliated Function
install_public	One-click installation of commonly used tools, universal type	ohmytool
install_private	[Unavailable] One-click installation of commonly used tools. The project is not intended to be open source at the beginning of development, so some functions involve privacy. This part is not available. It will be moved out of the gcr body and made into plug-ins later. Users can also import custom plug-in libraries.	ohmytool

📚 Appendix (All Functions)

L1	L2	L3	Description	Need Params	Others
ohmyshell					
	myupdate	myupdate	Update gcr version	No	
	mytool	jp	Open jupyter notebook	No	need to install jupyter lab/notebook
		listen	Ssh port forwarding	No	Support multiple forwarding modes
		pathadd	Add an environment variable to the shell	No	
		condainit	Add conda initialization to the shell	No	Equivalent to the part added to the shell by default when installing anaconda
		sz	View file/folder size	No	Supports viewing specific file/folder size, number of files or server hard disk space
		listen_vpn	Controlling Server Side Clash VPN	No	Clash needs to be enabled on the server side
		download_googledrive	Download files from Google drive	No	Requires gdown installation
		vpstest	Test server parameters	No	Common tools for testing vps
	mypip	pipset	Set pip installation source	No	Optional options: Tsinghua/Alibaba/Tencent
		pipbase	pip installs commonly used packages	No	Optional (recommended to install all)：'pygments' 'gpustat' 'jupyterlab' 'tensorboard' 'woaigpu' 'gdown' 'shell-gpt'
		pipl	pip to view installed packages	No	
		pipi	pip install		If no parameters are given, pip install -r req*.txt will be run by default.
		pipu	pip uninstall	Yes	
			activate conda		Without parameters, the activateable environment will

	myenv	env	environment	Option	be displayed and waiting for selection. With parameters, the target environment will be activated directly.
		lenv	Show that the conda environment is installed	No	
		cenv	Create a new conda environment	No	
		denv	Delete an existing conda environment	No	
	mygpu	gpu	View user usage of GPU in real time	No	need to pip install gpustat
		nvidia	View gpu usage in real time	No	nvidia-smi
		pid	View the pid of all gpu running tasks	No	kill gpu tasks
		gpus	Display the command for the specified graphics card number	No	
	mypermission	mychmod	like chmod	No	
		mychown	like chown	No	
		passwd	Change account password	No	
	mygit	gita	git add	Yes	
		gitaa	git add .	No	
		gitb	git branch	No	
		gitc	like git commit	No	
		gitl	git log	No	
		gc	git clone	No	
		gitst	git status	No	
	(Others)	sc	source ${shell_file}	No	refresh shell env
		l	joshuto	No	If joshuto installed
		cl	clear	No	
		pcat	pygmentize -g	Yes	
		tb	tensorboard	No	
		download/upload	Download/upload files/folders	Option	rely on trzsz
		startvpn	VPN for terminal	No	rely on clash
		stopvpn	close VPN for terminal	No	
		dockeradd	Add user to docker user group	No	need root permission
ohmytool					
	(None yet)	new_vps_help	-		
		install_vpn	-		

		install_xui	-		
		install_docker	install docker		
		install_trzsz	Support terminal file upload/download		Recommend!!
		install_tssh	replace ssh		Recommend!!
		install_spacevim	make vim powerful		Not recommend，recommended to use code instead of vim
		install_server-administration	Common tools for server administrators		
		install_nginxproxy	install npm		
		install_homeweb	-		
		install_newtab	-		
		install_deepl	-		
		install_cloudreve	-		
		install_overleaf	overleaf		
		install_serverstatus	Server status monitoring		
		install_frp	frp		
		install_chatgpt	chatgpt api to web		
		install_webmonitor	Website monitoring		
		install_vncdocker	vpc		
		install_qbittorrent	-		
		install_jellyfin	jellyfin movie		
		install_image			
		install_samtools	#TODO		
		install_PBShelper	PBS Assistant		Recommend!! nscc/hpc good helper
		install_htop_wo_sudo	Install htop on a server without root permissions		
		install_autojump	go to anywhere		Recommend!!
		install_fzf	terminal command fuzzy search		
		install_copypaste	-		
		install_safe_rm	safe delete		Recommend!!
		install_discard_vim	discard vim and embrace code		Recommend!!
		install_joshuto			