

Infra (docker, k8s, helm etc..)

- [Linux \(incl wsl2\)](#)
 - [Oh My ZSH](#)
 - [wsl2 setup](#)
- [k8s](#)
 - [k3s](#)
 - [Helm](#)
- [Docker](#)
 - [Install \(ubuntu\)](#)

Linux (incl wsl2)

Linux (incl wsl2)

Oh My ZSH

Install ZSH

```
sudo apt get install zsh -y
```

Install oh my zsh

```
sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

Install powerlevel10k

Manual font installation

1. Download these four ttf files:

- [MesloLGS NF Regular.ttf](#)
- [MesloLGS NF Bold.ttf](#)
- [MesloLGS NF Italic.ttf](#)
- [MesloLGS NF Bold Italic.ttf](#)

Select all the four font files, right mouse -> more options -> install for all users

- **Windows Terminal** by Microsoft (the new thing): Open *Settings* (Ctrl+I), click either on the selected profile under *Profiles* or on *Defaults*, click *Appearance* and set *Font face* to

MesloLGS NF.

Oh My Zsh

1. Clone the repository:

```
git clone --depth=1 https://github.com/romkatv/powerlevel10k.git ${ZSH_CUSTOM:-$HOME/.oh-my
```

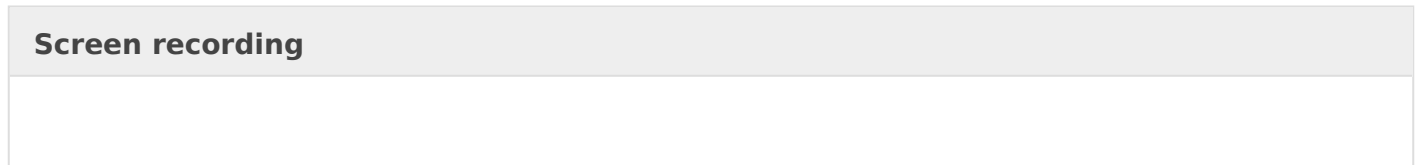
Users in China can use the official mirror on gitee.com for faster download.

```
git clone --depth=1 https://gitee.com/romkatv/powerlevel10k.git ${ZSH_CUSTOM:-$HOME/.oh-my-
```

2. Set `ZSH_THEME="powerlevel10k/powerlevel10k"` in `~/.zshrc`.

Configuration wizard

Type `p10k configure` to access the builtin configuration wizard right from your terminal.



All styles except [Pure](#) are functionally equivalent. They display the same information and differ only in presentation.

Configuration wizard creates `~/.p10k.zsh` based on your preferences. Additional prompt customization can be done by editing this file. It has plenty of comments to help you navigate through configuration options.

Linux (incl wsl2)

wsl2 setup

/etc/wsl.conf

[boot]

systemd=true

sudoers:

Allow members of group sudo to execute any command

##%sudo ALL=(ALL:ALL) ALL

%sudo ALL=(ALL) NOPASSWD: ALL

k8s

k8s

k3s

Install Script

K3s provides an installation script that is a convenient way to install it as a service on systemd or openrc based systems. This script is available at <https://get.k3s.io>. To install K3s using this method, just run:

```
curl -sfL https://get.k3s.io | sh -
```

After running this installation:

- The K3s service will be configured to automatically restart after node reboots or if the process crashes or is killed
- Additional utilities will be installed, including `kubectl`, `crictl`, `ctr`, `k3s-killall.sh`, and `k3s-uninstall.sh`
- A `kubeconfig` file will be written to `/etc/rancher/k3s/k3s.yaml` and the `kubectl` installed by K3s will automatically use it

A single-node server installation is a fully-functional Kubernetes cluster, including all the datastore, control-plane, kubelet, and container runtime components necessary to host workload pods. It is not necessary to add additional server or agent nodes, but you may want to do so to add additional capacity or redundancy to your cluster.

To install additional agent nodes and add them to the cluster, run the installation script with the `K3S_URL` and `K3S_TOKEN` environment variables. Here is an example showing how to join an agent:

```
curl -sfL https://get.k3s.io | K3S_URL=https://myserver:6443 K3S_TOKEN=mynodetoken sh -
```

Setting the `K3S_URL` parameter causes the installer to configure K3s as an agent, instead of a server. The K3s agent will register with the K3s server listening at the supplied URL. The value to use for `K3S_TOKEN` is stored at `/var/lib/rancher/k3s/server/node-token` on your server node.

Kubectl config

First set up your an environmental variable for `KUBECONFIG=~/.kube/config`.

```
export KUBECONFIG=~/.kube/config
```

Then let's generate the file at that location. [Your `k3s.yaml` file should **NOT** be world readable.](#) This is by-design. It should be owned by root and set to `0600`. Instead copy the config locally as [described here](#),

```
mkdir ~/.kube 2> /dev/null
sudo k3s kubectl config view --raw > "$KUBECONFIG"
chmod 600 "$KUBECONFIG"
```

You can add `KUBECONFIG=~/.kube/config` to your `~/.profile` or `~/.bashrc` to make it persist on reboot.

k8s

Helm

```
$ curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
$ chmod 700 get_helm.sh
$ ./get_helm.sh
```

Docker

Install (ubuntu)

Run the following command to uninstall all conflicting packages:

```
for pkg in docker.io docker-doc docker-compose podman-docker containerd runc; do sudo apt-get remove $pkg; done
```

`apt-get` might report that you have none of these packages installed.

Images, containers, volumes, and networks stored in `/var/lib/docker/` aren't automatically removed when you uninstall Docker. If you want to start with a clean installation, and prefer to clean up any existing data, read the [uninstall Docker Engine](#) section.

Install using the apt repository

Before you install Docker Engine for the first time on a new host machine, you need to set up the Docker repository. Afterward, you can install and update Docker from the repository.

Set up the repository

1. Update the `apt` package index and install packages to allow `apt` to use a repository over HTTPS:

```
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
```

2. Add Docker's official GPG key:

```
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

3. Use the following command to set up the repository:

```
echo \  
  "deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]  
https://download.docker.com/linux/ubuntu \  
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

“ **Note**

If you use an Ubuntu derivative distro, such as Linux Mint, you may need to use `UBUNTU_CODENAME` instead of `VERSION_CODENAME`.

Install Docker Engine

1. Update the `apt` package index:

```
sudo apt-get update
```

2. Install Docker Engine, containerd, and Docker Compose.

- [Latest](#)
- [Specific version](#)

To install the latest version, run:

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin  
docker-compose-plugin
```

3. Verify that the Docker Engine installation is successful by running the `hello-world` image.

```
sudo docker run hello-world
```

This command downloads a test image and runs it in a container. When the container runs, it prints a confirmation message and exits.

```
sudo groupadd docker  
sudo usermod -aG docker $USER  
newgrp docker
```