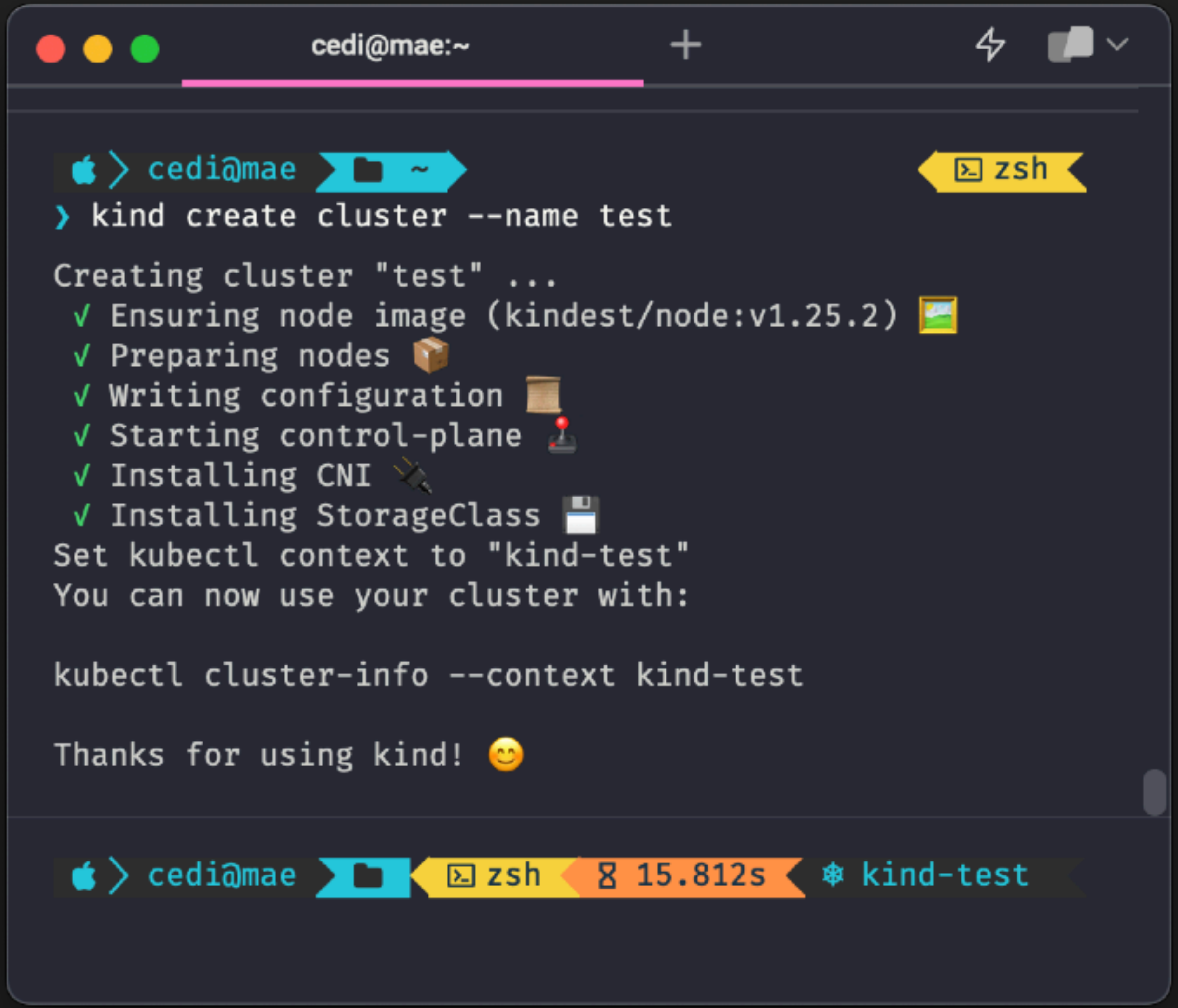
GET STARTED WITH A DEV ENVIRONMENT

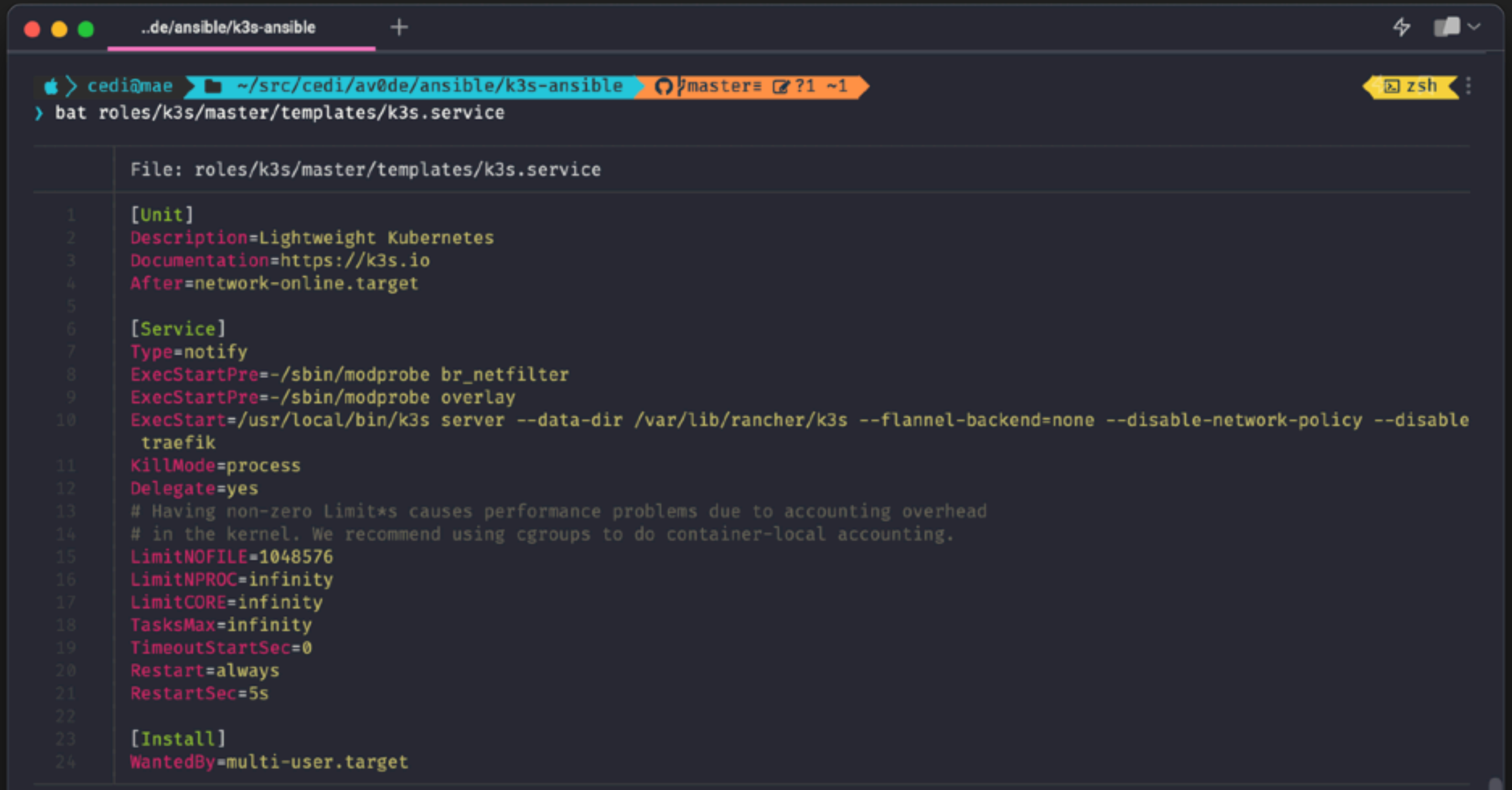
- Make your first steps with “kind” on your local machine



```
 cedi@mae  ~                                         zsh
❯ kind create cluster --name test

Creating cluster "test" ...
 ✓ Ensuring node image (kindest/node:v1.25.2) 🖼
 ✓ Preparing nodes 📦
 ✓ Writing configuration 📜
 ✓ Starting control-plane 🕹️
 ✓ Installing CNI 🔌
 ✓ Installing StorageClass 💾
Set kubectl context to "kind-test"
You can now use your cluster with:

kubectl cluster-info --context kind-test

Thanks for using kind! 😊

 cedi@mae   zsh  ⌛ 15.812s  ⎈ kind-test
```

SINGLE NODE-, EDGE-, (AND HOMELAB) DEPLOYMENT

- K3s on a single node is good enough for your home-lab
- Multi-node K3s is probably enough for most use-cases!
- Use a systemd service unit to keep k3s running
- Use the K3s Ansible Playbook



```
cedi@mae  ~/src/cedi/av0de/ansible/k3s-ansible  master≡ ?1 ~1
❯ bat roles/k3s/master/templates/k3s.service

       File: roles/k3s/master/templates/k3s.service

       [Unit]
       Description=Lightweight Kubernetes
       Documentation=https://k3s.io
       After=network-online.target

       [Service]
       Type=notify
       ExecStartPre=-/sbin/modprobe br_netfilter
       ExecStartPre=-/sbin/modprobe overlay
       ExecStart=/usr/local/bin/k3s server --data-dir /var/lib/rancher/k3s --flannel-backend=none --disable-network-policy --disable
        traefik
       KillMode=process
       Delegate=yes
       # Having non-zero Limit*s causes performance problems due to accounting overhead
       # in the kernel. We recommend using cgroups to do container-local accounting.
       LimitNOFILE=1048576
       LimitNPROC=infinity
       LimitCORE=infinity
       TasksMax=infinity
       TimeoutStartSec=0
       Restart=always
       RestartSec=5s

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