

KUBERNETES

Irma Nurliza Lumbantoruan

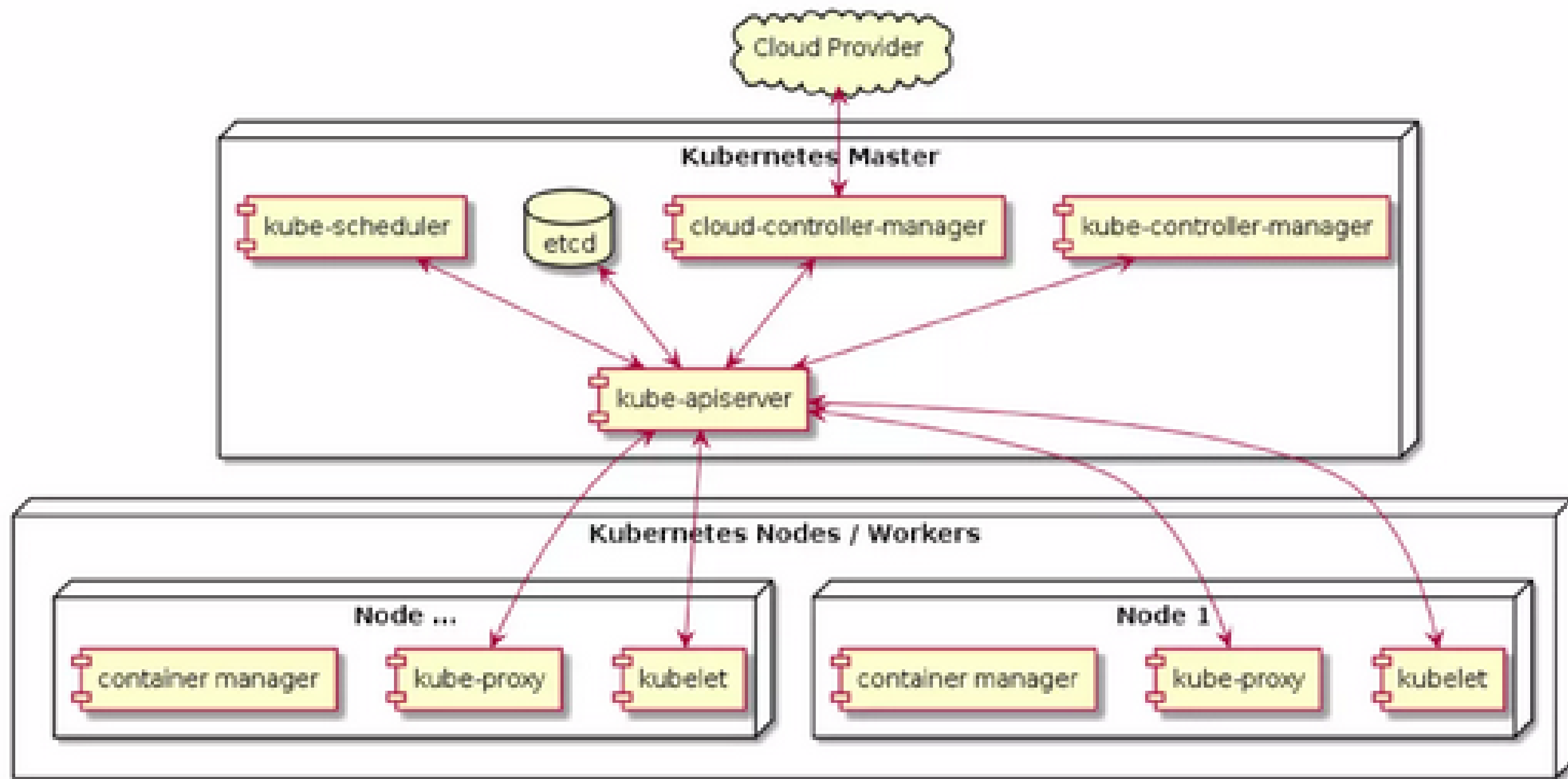


Pengertian Kubernetes

- Kubernetes adalah aplikasi untuk automation deployment, scaling, dan manajemen aplikasi berbasis container
- Kubernetes ini juga merupakan aplikasi open source yang paling populer
- Kubernetes ini banyak digunakan oleh perusahaan-perusahaan



Arsitektur Kubernetes



4

Kubernetes Master

- master - bertindak sebagai pengendali utama untuk kubernetes dan bertanggung jawab untuk menjalankan Api-server, scheduler, dan cluster.
- Kube-apiserver bertugas sebagai API yang digunakan untuk berinteraksi dengan kubernetes cluster
- etcd bertugas sebagai data base untuk menyimpan data Kubernetes Cluster
- Kube-scheduler bertugas untuk memperhatikan aplikasi yang kita jalankan dan meminta node untuk menjalankan aplikasi
- Kube-control-manager bertugas untuk mengontrol kubernetes cluster
- cloud-control-manager bertugas untuk melakukan kontrol terhadap interaksi dengan cloud provider



Kubernetes Nodes

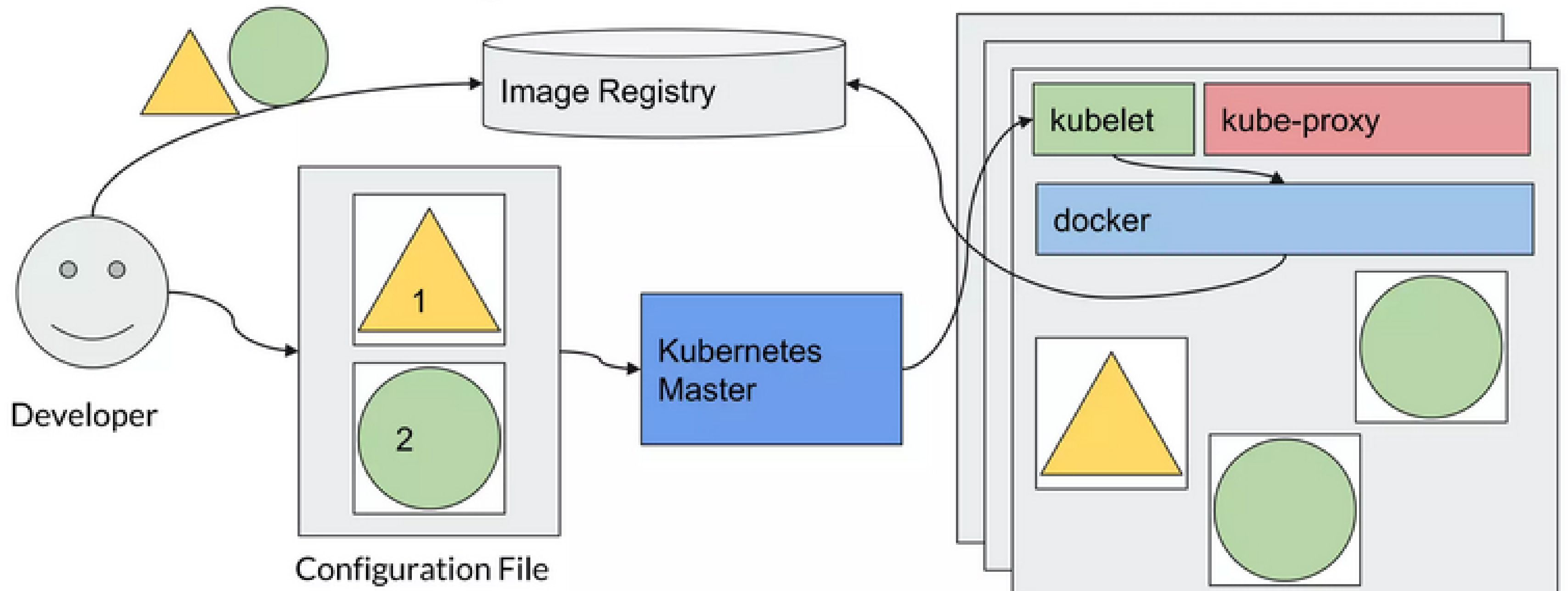
- Kubelet berjalan di setiap node dan bertugas untuk memastikan bahwa aplikasi kita berjalan di atas node
- Kube-proxy berjalan di setiap node dan bertugas sebagai proxy terhadap arus network yang masuk ke aplikasi kita dan juga berperan sebagai load balancer
- Container-manager berjalan di setiap Node dan bertugas sebagai container manager.



Konsep Inti Kubernetes

- Cluster - kumpulan host yang menggabungkan sumber daya yang tersedia termasuk cpu, ram, disk, dan perangkatnya ke dalam kumpulan yang dapat digunakan
- Node - sebuah host tunggal, baik fisik maupun virtual yang mampu menjalankan pod.
- Namespace - cluster atau lingkungan logis. Metode utama untuk membagi cluster atau akses pelingkupan

Detail Alur Kerja Kubernetes



Dokumentasi

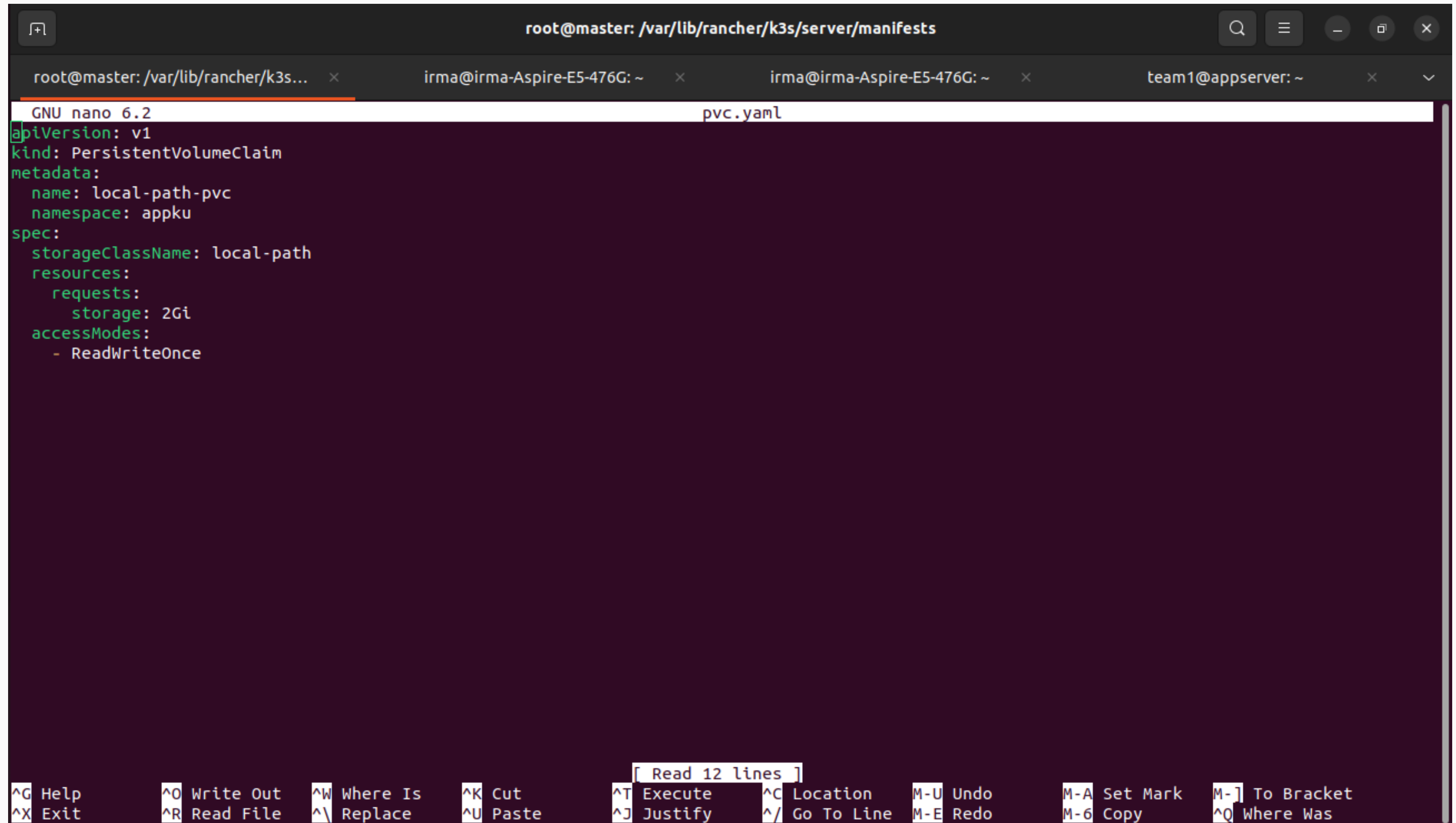
```
root@master: /var/lib/rancher/k3s/server/manifests
```

```
sihombing@master:~$ sudo su  
root@master:/home/sihombing# curl -sfL https://get.k3s.io | sh -  
[INFO] Finding release for channel stable  
[INFO] Using v1.28.5+k3s1 as release  
[INFO] Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.28.5+k3s1/sha256sum-amd64.txt  
[INFO] Downloading binary https://github.com/k3s-io/k3s/releases/download/v1.28.5+k3s1/k3s  
[INFO] Verifying binary download  
[INFO] Installing k3s to /usr/local/bin/k3s  
[INFO] Skipping installation of SELinux RPM  
[INFO] Creating /usr/local/bin/kubectrl symlink to k3s  
[INFO] Creating /usr/local/bin/crictl symlink to k3s  
[INFO] Creating /usr/local/bin/ctr symlink to k3s  
[INFO] Creating killall script /usr/local/bin/k3s-killall.sh  
[INFO] Creating uninstall script /usr/local/bin/k3s-uninstall.sh  
[INFO] env: Creating environment file /etc/systemd/system/k3s.service.env  
[INFO] systemd: Creating service file /etc/systemd/system/k3s.service  
[INFO] systemd: Enabling k3s unit  
Created symlink /etc/systemd/system/multi-user.target.wants/k3s.service → /etc/systemd/system/k3s.service.  
[INFO] systemd: Starting k3s  
root@master:/home/sihombing# systemctl status k3s  
● k3s.service - Lightweight Kubernetes  
   Loaded: loaded (/etc/systemd/system/k3s.service; enabled; vendor preset: enabled)  
   Active: active (running) since Fri 2024-01-19 20:41:47 WIB; 1min 5s ago  
     Docs: https://k3s.io  
 Process: 30920 ExecStartPre=/bin/sh -xc ! /usr/bin/systemctl is-enabled --quiet nm-cloud-setup.service 2>/dev/null (code=exited, status=0)  
 Process: 30922 ExecStartPre=/sbin/modprobe br_netfilter (code=exited, status=0/SUCCESS)  
 Process: 30928 ExecStartPre=/sbin/modprobe overlay (code=exited, status=0/SUCCESS)  
 Main PID: 30929 (k3s-server)  
    Tasks: 84  
   Memory: 974.5M  
      CPU: 47.716s  
   CGroup: /system.slice/k3s.service  
           └─30929 "/usr/local/bin/k3s server"  
             └─31339 "containerd " "  
               └─31910 /var/lib/rancher/k3s/data/28f7e87eba734b7f7731dc900e2c84e0e98ce869f3dcf57f65dc7bbb80e12e56/bin/containerd-shim-runc-v2 ->  
                 └─32015 /var/lib/rancher/k3s/data/28f7e87eba734b7f7731dc900e2c84e0e98ce869f3dcf57f65dc7bbb80e12e56/bin/containerd-shim-runc-v2 ->
```


Dokumentasi

```
root@master: /var/lib/rancher/k3s/server/manifests
root@master: /var/lib/rancher/k3s... x  irma@irma-Aspire-E5-476G: ~ x  irma@irma-Aspire-E5-476G: ~
GNU nano 6.2 ingress-nginx.yaml
apiVersion: v1
kind: Namespace
metadata:
  name: ingress-nginx
---
apiVersion: helm.cattle.io/v1
kind: HelmChart
metadata:
  name: ingress-nginx
  namespace: ingress-nginx
spec:
  repo: https://kubernetes.github.io/ingress-nginx
  chart: ingress-nginx
  targetNamespace: ingress-nginx
  valuesContent: |-
    controller:
      image:
        tag: "v1.8.1"
      service:
        type: LoadBalancer
```

Dokumentasi

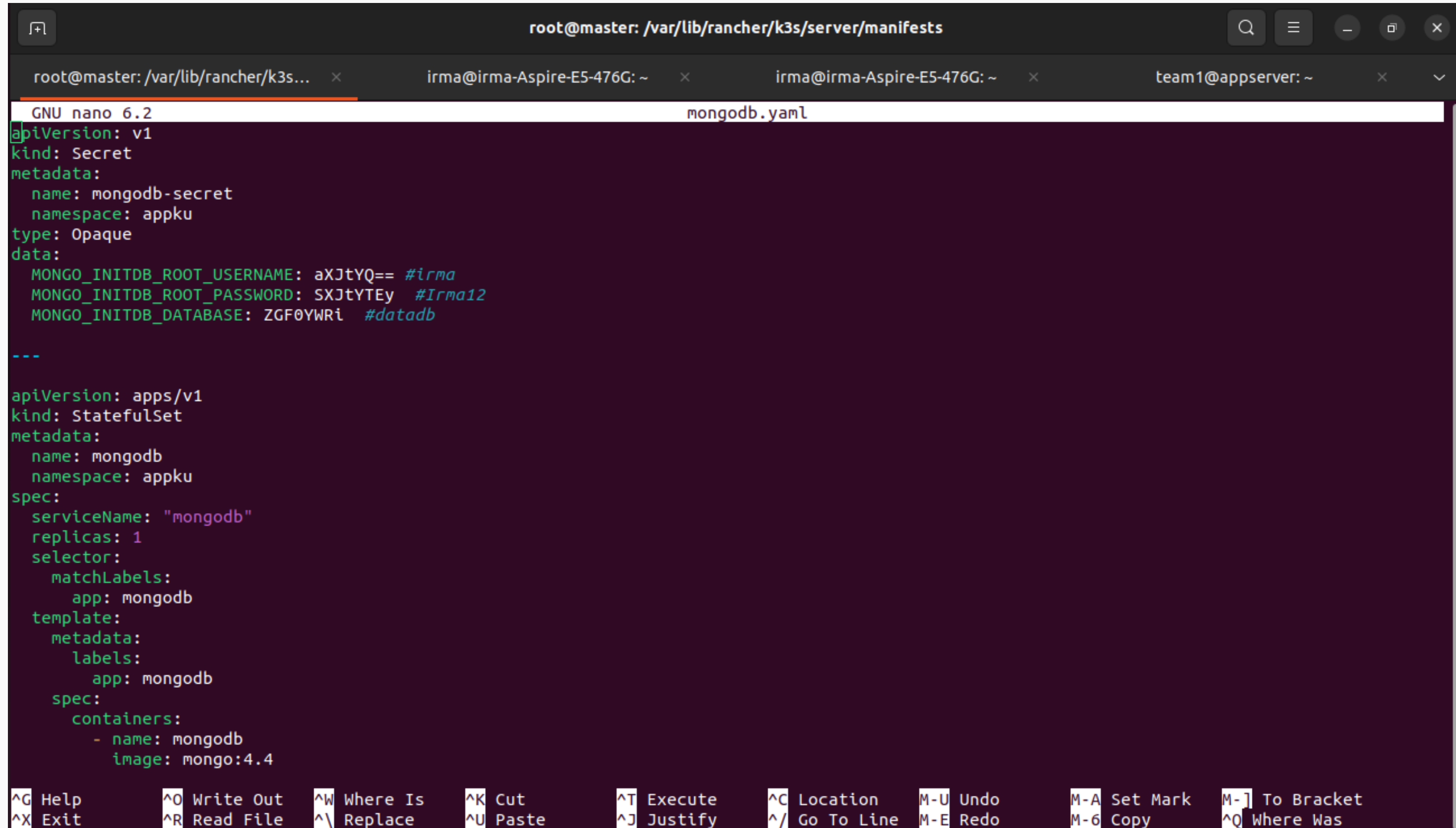


```
root@master: /var/lib/rancher/k3s/server/manifests
GNU nano 6.2 pvc.yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: local-path-pvc
  namespace: appku
spec:
  storageClassName: local-path
  resources:
    requests:
      storage: 2Gi
  accessModes:
    - ReadWriteOnce
```

Read 12 lines

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo M-6 Copy ^Q Where Was

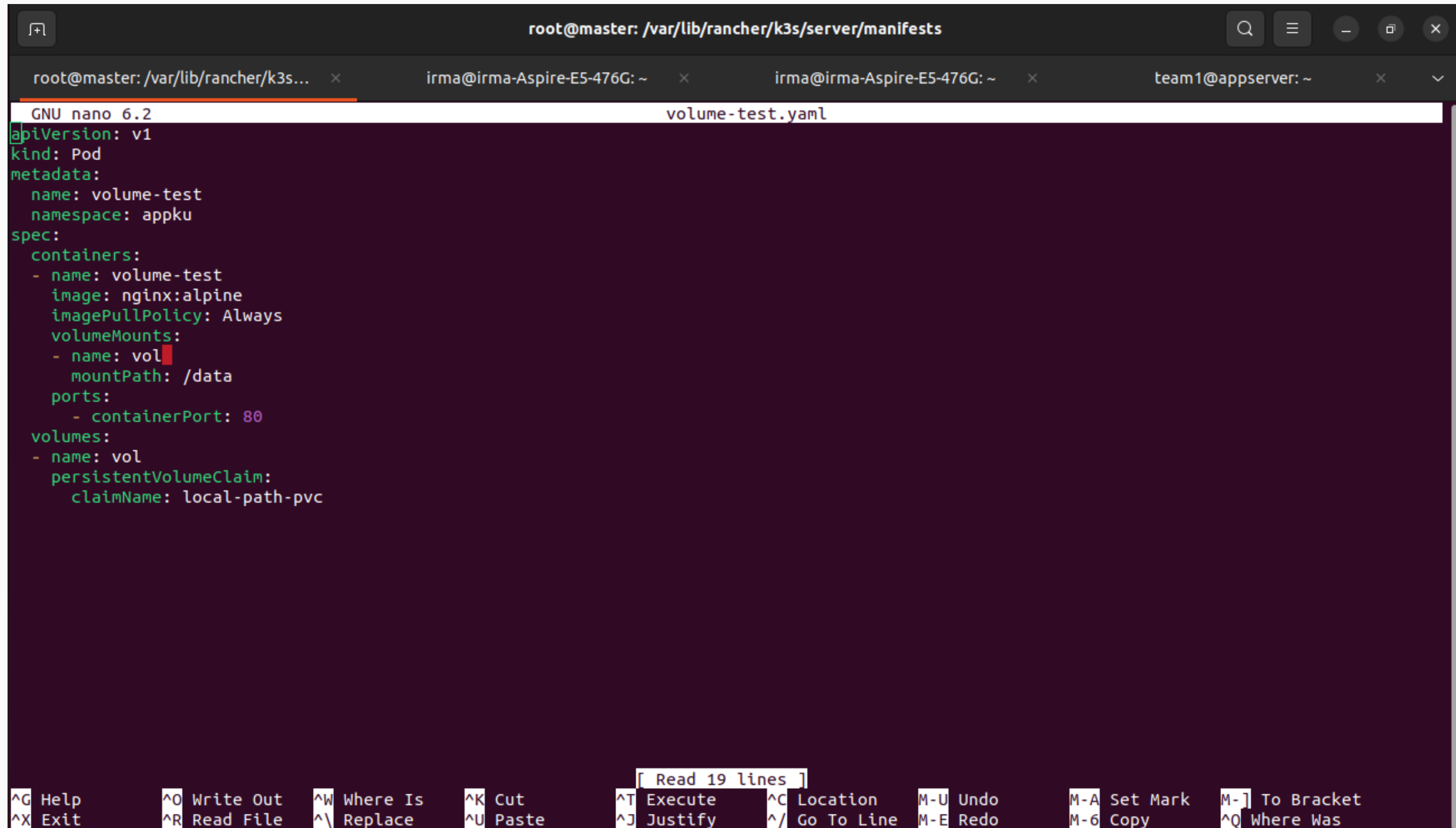
Dokumentasi



```
GNU nano 6.2 mongodb.yaml
apiVersion: v1
kind: Secret
metadata:
  name: mongodb-secret
  namespace: appku
type: Opaque
data:
  MONGO_INITDB_ROOT_USERNAME: aXJtYQ== #irma
  MONGO_INITDB_ROOT_PASSWORD: SXJtYTEy #Irma12
  MONGO_INITDB_DATABASE: ZGF0YWRI #datadb
---
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mongodb
  namespace: appku
spec:
  serviceName: "mongodb"
  replicas: 1
  selector:
    matchLabels:
      app: mongodb
  template:
    metadata:
      labels:
        app: mongodb
    spec:
      containers:
        - name: mongodb
          image: mongo:4.4

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo     M-A Set Mark M-] To Bracket
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo     M-6 Copy     ^Q Where Was
```

Dokumentasi

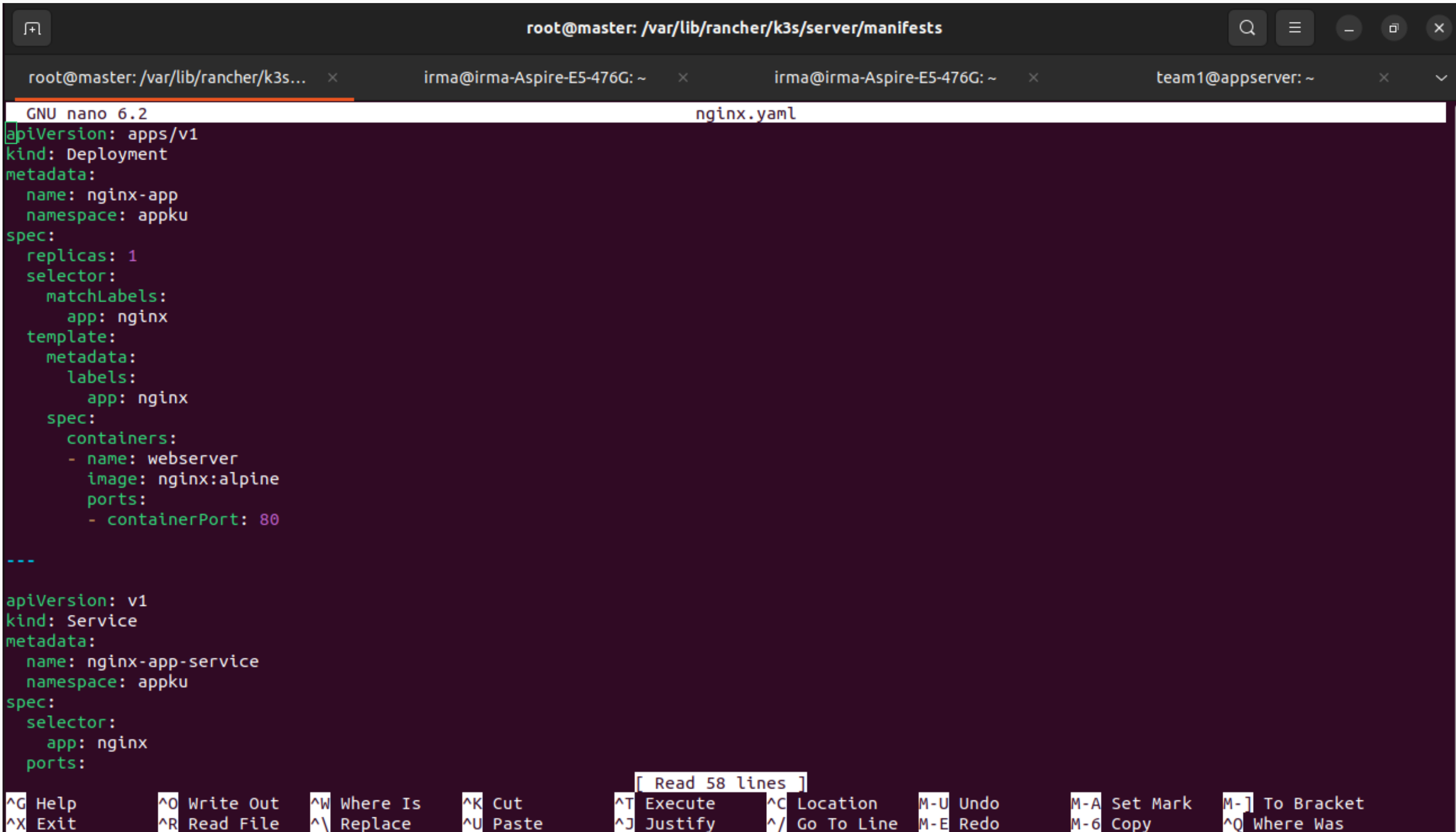


```
root@master: /var/lib/rancher/k3s/server/manifests
GNU nano 6.2 volume-test.yaml
apiVersion: v1
kind: Pod
metadata:
  name: volume-test
  namespace: appku
spec:
  containers:
  - name: volume-test
    image: nginx:alpine
    imagePullPolicy: Always
    volumeMounts:
    - name: vol
      mountPath: /data
    ports:
    - containerPort: 80
  volumes:
  - name: vol
    persistentVolumeClaim:
      claimName: local-path-pvc
```

Read 19 lines

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark M-] To Bracket
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo M-6 Copy ^Q Where Was

Dokumentasi



```
root@master: /var/lib/rancher/k3s/...  irma@irma-Aspire-E5-476G: ~  irma@irma-Aspire-E5-476G: ~  team1@appserver: ~
GNU nano 6.2 nginx.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-app
  namespace: appku
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: webserver
        image: nginx:alpine
        ports:
        - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: nginx-app-service
  namespace: appku
spec:
  selector:
    app: nginx
  ports:
```

Read 58 lines

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location	M-U Undo	M-A Set Mark	M-] To Bracket
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^_ Go To Line	M-E Redo	M-6 Copy	^Q Where Was

Terima Kasih

