

# K3s

- [Setting Up Our K3s for our Homelab](#)
- [K3s issue when trying to access cluster from another machine](#)

# Setting Up Our K3s for our Homelab

Flow:

Flow 1: K3s Setup + Traefik + Rancher

Flow 2: Memcached, Redis Cluster

Flow 3: Setup External Storage ( ZFS, Minio ), Internal Storage should be deployment only, i don't use containers as storage.

Why i don't use longhorn as storage:

1. High load of network traffic, requirement of 10Gb ethernet
2. Faster & less complex management , less problems, less errors.

# K3s issue when trying to access cluster from another machine

## problem:

one day you don't know why the hell is going on with kubectl on your machine and you've already copied `remote_machine:./kube/config` into your local machine and that worked before but now it keep telling wrong api group list or authorized.

```
kubectl get ns
```

```
E0108 21:42:56.860495 805013 memcache.go:265] couldn't get current server API group list: the
server has asked for the client to provide credentials
E0108 21:42:56.862131 805013 memcache.go:265] couldn't get current server API group list: the
server has asked for the client to provide credentials
E0108 21:42:56.863214 805013 memcache.go:265] couldn't get current server API group list: the
server has asked for the client to provide credentials
E0108 21:42:56.864421 805013 memcache.go:265] couldn't get current server API group list: the
server has asked for the client to provide credentials
E0108 21:42:56.865299 805013 memcache.go:265] couldn't get current server API group list: the
server has asked for the client to provide credentials
error: You must be logged in to the server (the server has asked for the client to provide
credentials)
```

or when you access to the api:

```
{
  "kind": "Status",
  "apiVersion": "v1",
  "metadata": {},
  "status": "Failure",
  "message": "Unauthorized",
  "reason": "Unauthorized",
  "code": 401
}
```

## solution:

use the value `config` from the path into your `~/.kube/config`

SSH to your k3s primary server

look for this file

```
/etc/rancher/k3s/k3s.yaml
```

view it using `cat /etc/rancher/k3s/k3s.yaml`

use the value certification inside the file that includes:

```
certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tC.....  
client-certificate-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0.....  
client-key-data: MgUFJJVkfFURSBLRVktLS0t.....
```

Open your local machine `~/kube/config` and replace them

here is my complete `.kube/config` file content:

```
apiVersion: v1  
clusters:  
- cluster:  
  certificate-authority-data:  
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUJ.....ekZiSVlGcUtNcWlHVk5yMjBaeUMvQml6Y0QKWepjMzJ0RH  
F2NHlz0VFua29rVE1GULYva1A2Z3pqaW9uaXB1Q1lJRUVyMkpvMEl3U.....1DQXFRd0R3WURW.....JNdVV6KzI5QWlCM  
21n.....Ci0tLS0tRU5EIENFUlRJRklDQVRFLS0tLS0K  
  server: https://192.168.1.xxx:6443  
  name: k3s  
contexts:  
- context:  
  cluster: k3s  
  user: son  
  name: k3s  
current-context: k3s  
kind: Config  
preferences: {}  
users:  
- name: son  
  user:  
  client-certificate-data:  
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JS.....lFlQzh3Q2dZSUtVWkl6ajBFQXdJd0l6RWhNQjhHQTFVRUF3d  
1kKYXp0ekxXTnNhV1Z1ZEMxallVQXh0ekF3TURVd05UY3lNQ.....  
  client-key-data:
```

```
LS0tLS1CRUdJTiBFQyBQUk1w...0NBUUVFSU5nQzZqVEh4TUhXNGQ2SnZ5ZGs5OVVqOTBDaFQzMWU2d1lkE5QR2RUSTB
vQW9HQ0NxR1NNNDkKQXdFSG9VUURRZ0FFa0h00TRxWVBFS2pVaVUr....
```

Happy!

```
connection to 192.168.1.101 closed
son k3s-ansible main kubectl get po
NAME                                READY   STATUS    RESTARTS   AGE
helpdesk-55bd78c7cc-f66m6          1/1     Running   17 (66d ago)  205d
nfs-subdir-external-provisioner-845c8fddb-fwjgs 1/1     Running   5 (57d ago)   119d
rancher-alerting-drivers-sachet-6c954b9997-wq4m5 2/2     Running   24 (66d ago)  200d
traefik-59dfbc4666-bg69z           1/1     Running   5 (66d ago)   198d
traefik-59dfbc4666-h8w5k           1/1     Running   12 (66d ago)  200d
```

Some useful commands to remember:

```
kubectl config get-contexts
```

```
CURRENT NAME CLUSTER AUTHINFO NAMESPACE
```

```
* k3s k3s son
```

```
kubectl config use-context k3s
```