Introduction How ETCD Works

Overview

- ❖ ETCD Definition
- ❖ Why ETCD
- ❖ ETCD with Kubernetes
- ❖ ETCD Installation
- ❖ Examples
- ❖ Summary

- **What is ETCD:**
  - Etcd is an open source distributed key-value store used to hold and manage the critical information that distributed systems need to keep running. Most notably, it manages the configuration data, state data, and metadata for Kubernetes, the popular container orchestration platform.

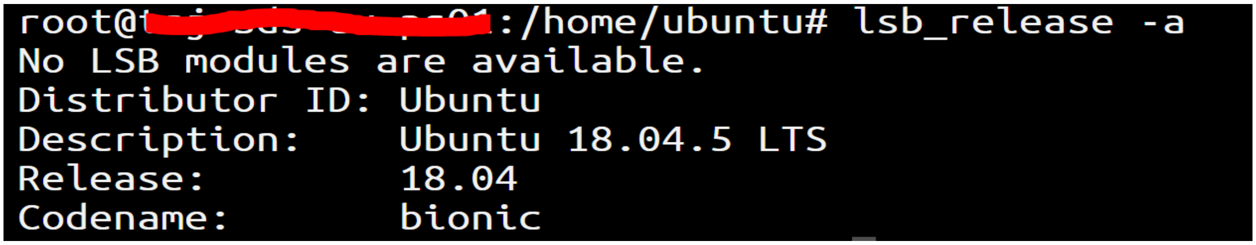
ETCD is a distributed reliable key-value store that is Simple, Secure & Fast

- **Why ETCD?**
  - It's no small task to serve as the data backbone that keeps a distributed workload running. But etcd is built for the task, designed from the ground up for the following qualities:
    - **Fully replicated:** Every node in an etcd cluster has access to the full data store.
    - **Highly available:** etcd is designed to have no single point of failure and gracefully tolerate hardware failures and network partitions.
    - **Reliably consistent:** Every data 'read' returns the latest data 'write' across all clusters.
    - **Fast:** etcd has been benchmarked at 10,000 writes per second.
    - **Secure:** etcd supports automatic Transport Layer Security (TLS) and optional secure socket layer (SSL) client certificate authentication. Because etcd stores vital and highly sensitive configuration data, administrators should implement role-based access controls within the deployment and ensure that team members interacting with etcd are limited to the least-privileged level of access necessary to perform their jobs.

- **ETCD and Kubernetes**
    - etcd is included among the core Kubernetes components and serves as the primary key-value store for creating a functioning, fault-tolerant Kubernetes cluster. The Kubernetes API server stores each cluster's state data in etcd. Kubernetes uses etcd's "watch" function to monitor this data and to reconfigure itself when changes occur. The "watch" function stores values representing the actual and ideal state of the cluster and can initiate a response when they diverge.

Now let's get started with ETCD..

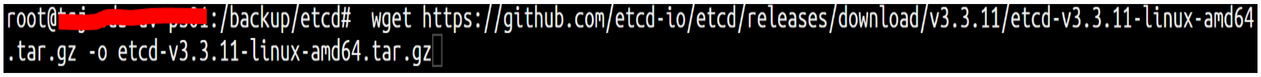
1. **Install ETCD on Ubuntu with below version:**



```
root@t[redacted]:/home/ubuntu# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 18.04.5 LTS
Release:        18.04
Codename:       bionic
```

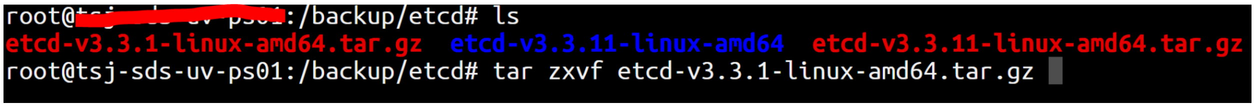
2. **Download Binaries:**

   wget
   https://github.com/etcd-io/etcd/releases/download/v3.3.11/etcd-v3.3.11-linux-amd64.tar.gz -o etcd-v3.3.11-linux-amd64.tar.gz



```
root@[redacted]:/backup/etcd#  wget https://github.com/etcd-io/etcd/releases/download/v3.3.11/etcd-v3.3.11-linux-amd64
.tar.gz -o etcd-v3.3.11-linux-amd64.tar.gz
```

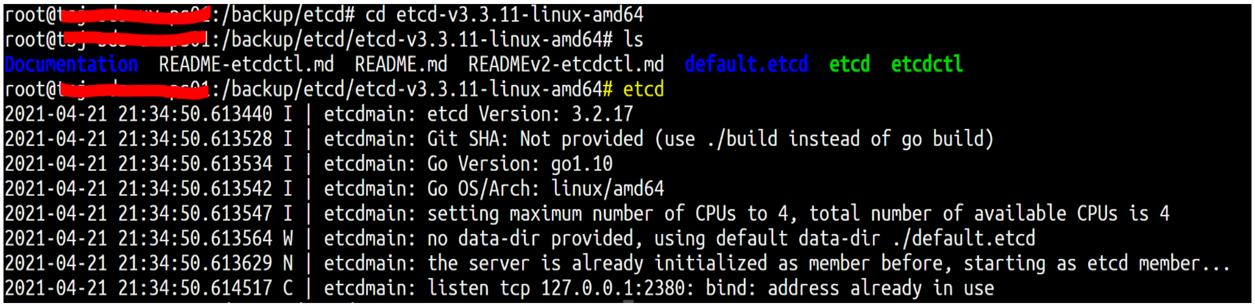
3. **Extract the .tar file:**



```
root@[redacted]:/backup/etcd# ls
etcd-v3.3.1-linux-amd64.tar.gz  etcd-v3.3.11-linux-amd64  etcd-v3.3.11-linux-amd64.tar.gz
root@tsj-sds-uv-ps01:/backup/etcd# tar zxvf etcd-v3.3.1-linux-amd64.tar.gz
```

   As can see I already untar the file *etcd-v3.3.11-linux-amd64*

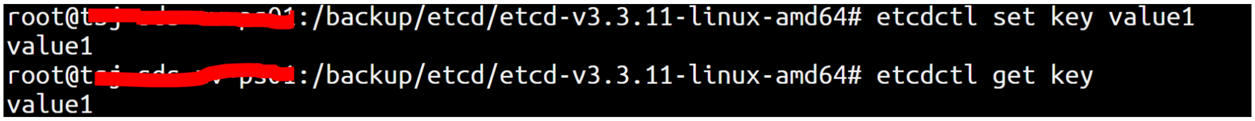
4. **Run ETCD:**



```
root@t[redacted]:/backup/etcd# cd etcd-v3.3.11-linux-amd64
root@t[redacted]:/backup/etcd/etcd-v3.3.11-linux-amd64# ls
Documentation  README-etcdctl.md  README.md  READMEv2-etcdctl.md  default.etcd  etcd  etcdctl
root@t[redacted]:/backup/etcd/etcd-v3.3.11-linux-amd64# etcd
2021-04-21 21:34:50.613440 I | etcdmain: etcd Version: 3.2.17
2021-04-21 21:34:50.613528 I | etcdmain: Git SHA: Not provided (use ./build instead of go build)
2021-04-21 21:34:50.613534 I | etcdmain: Go Version: go1.10
2021-04-21 21:34:50.613542 I | etcdmain: Go OS/Arch: linux/amd64
2021-04-21 21:34:50.613547 I | etcdmain: setting maximum number of CPUs to 4, total number of available CPUs is 4
2021-04-21 21:34:50.613564 W | etcdmain: no data-dir provided, using default data-dir ./default.etcd
2021-04-21 21:34:50.613629 N | etcdmain: the server is already initialized as member before, starting as etcd member...
2021-04-21 21:34:50.614517 C | etcdmain: listen tcp 127.0.0.1:2380: bind: address already in use
```

As you can see ETCD is already running in localhost:2380

5. **Now let's create a key, store data and retrieve it:**



```
root@[redacted]:/backup/etcd/etcd-v3.3.11-linux-amd64# etcdctl set key value1
value1
root@[redacted]:/backup/etcd/etcd-v3.3.11-linux-amd64# etcdctl get key
value1
```

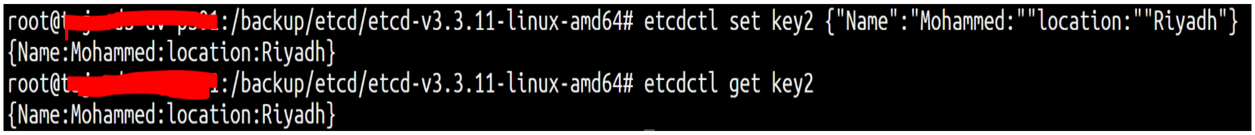
To set a value to **key:**

# etcdctl set key value1

To retrieve it:

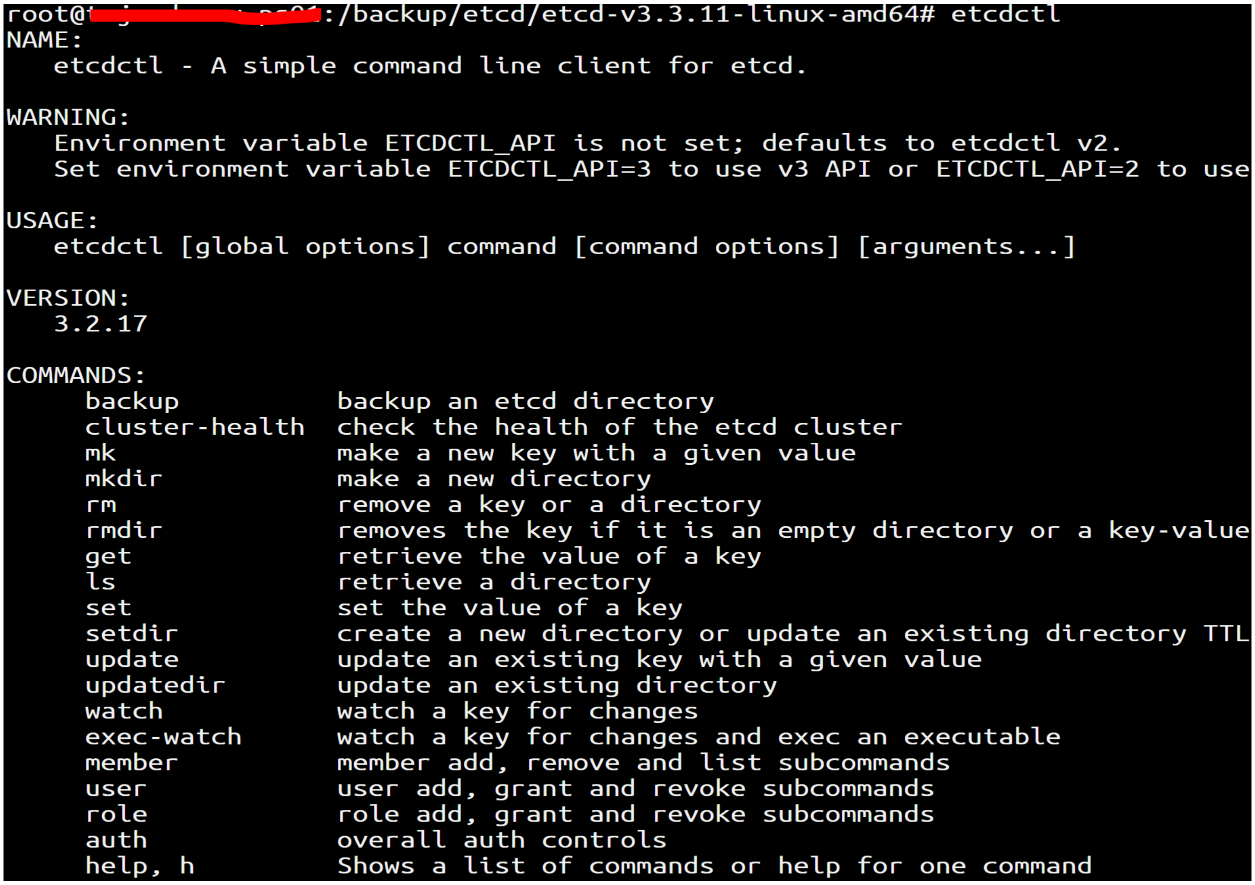
# **etcdctl get key**

- **Another example:**



```
root@[redacted]:/backup/etcd/etcd-v3.3.11-linux-amd64# etcdctl set key2 {"Name":"Mohammed:""location:""Riyadh"}
{Name:Mohammed:location:Riyadh}
root@[redacted]:/backup/etcd/etcd-v3.3.11-linux-amd64# etcdctl get key2
{Name:Mohammed:location:Riyadh}
```

- **To explore more commands:**



```
root@[redacted]:/backup/etcd/etcd-v3.3.11-linux-amd64# etcdctl
NAME:
   etcdctl - A simple command line client for etcd.

WARNING:
   Environment variable ETCDCTL_API is not set; defaults to etcdctl v2.
   Set environment variable ETCDCTL_API=3 to use v3 API or ETCDCTL_API=2 to use

USAGE:
   etcdctl [global options] command [command options] [arguments...]

VERSION:
   3.2.17

COMMANDS:
     backup          backup an etcd directory
     cluster-health  check the health of the etcd cluster
     mk              make a new key with a given value
     mkdir           make a new directory
     rm              remove a key or a directory
     rmdir           removes the key if it is an empty directory or a key-value
     get             retrieve the value of a key
     ls              retrieve a directory
     set             set the value of a key
     setdir          create a new directory or update an existing directory TTL
     update          update an existing key with a given value
     updatedir       update an existing directory
     watch           watch a key for changes
     exec-watch      watch a key for changes and exec an executable
     member          member add, remove and list subcommands
     user            user add, grant and revoke subcommands
     role            role add, grant and revoke subcommands
     auth            overall auth controls
     help, h         Shows a list of commands or help for one command
```

Summary:

This is a brief introduction on how ETCD works and how it is useful. Surly, there're plenty of ways to make use of it for instance you could use it store configurations, commands etc. Hope this helps you get started.

- **References:**

1. https://www.ibm.com/cloud/learn/etcd#:~:text=etcd%20is%20an%20open%20source,the%20popular%20container%20orchestration%20platform.
2. https://www.udemy.com/course/certified-kubernetes-administrator