

Project: Building a Distributed Key-Value Store with etcd (3 Weeks)

This project guides you through building a simple distributed key-value store using etcd, a popular key-value store for distributed systems. You'll explore basic functionalities and gain experience working with etcd during the 3-week period.

Project Goal: Develop a key-value store application that allows users to set and get key-value pairs using etcd for storage.

Week 1: Setting Up and Understanding etcd

● Deliverables:

- Install and configure a single-node etcd cluster locally.
- Write a program (Python/Go/Java) to connect to the etcd cluster.
- Implement functions to list all keys, get the value for a specific key, and put a key-value pair into etcd.

● Tasks:

- Download and install etcd according to your OS instructions (<https://github.com/etcd-io/etcd/releases>).
- Follow tutorials or documentation to set up a single-node cluster (<https://etcd.io/docs/v3.5/quickstart/>).
- Choose a programming language (Python, Go, Java are good options) and explore client libraries for interacting with etcd (<https://etcd.io/docs/v3.4/integrations/>).
- Write functions to:
 - List all keys using etcd client library.
 - Get the value for a specific key provided by the user.
 - Put a key-value pair into etcd, allowing users to specify both key and value.

Week 2: Adding Features and Error Handling

● Deliverables:

- Implement functionality to delete a key-value pair.

- Incorporate error handling for various operations (e.g., key not found, connection issues).
- Design a simple user interface (command-line or basic web interface) for interacting with the key-value store.

- **Tasks:**

- Extend your program to include a delete function that removes a key-value pair based on the provided key.
- Implement error handling mechanisms to catch potential issues like:
 - Key not found errors when getting or deleting non-existent keys.
 - Connection errors when the etcd cluster is unavailable.
- Design a user interface (text-based command-line or a simple web interface using a framework like Flask/Django) to allow users to:
 - See a list of available options (put, get, delete, list).
 - Provide key and value inputs for put operation.
 - Enter a key for get and delete operations.
 - Display appropriate messages based on the operation's success or failure.

Week 3: Scaling and Testing

- **Deliverables:**

- Explore running a multi-node etcd cluster (optional).
- Write unit tests for your program's functionalities.
- Document your project with explanations and comments in the code.

- **Tasks:**

- (Optional) Experiment with setting up a multi-node etcd cluster to understand how data is distributed across nodes. You can find instructions in the etcd documentation (<https://etcd.io/docs/>).
- Write unit tests for the core functionalities of your program (put, get,

delete, list) to ensure they behave as expected under different scenarios.

o Document your project with comments in the code explaining each function's purpose and overall program logic. You can also create a separate README file outlining the project setup, functionalities, and instructions to run the program.