

## LevelDB and CouchDB

- The current options for the peer state database are LevelDB and CouchDB.
- LevelDB is the default key-value state database embedded in the peer process.
- CouchDB is an alternative external state database. Like the LevelDB key-value store, CouchDB can store any binary data that is modeled in chaincode (CouchDB attachments are used internally for non-JSON data).
- As a document object store, CouchDB allows you to store data in JSON format, issue JSON queries against your data, and use indexes to support your queries.

[https://hyperledger-fabric.readthedocs.io/en/release-2.2/couchdb\\_as\\_state\\_database.html](https://hyperledger-fabric.readthedocs.io/en/release-2.2/couchdb_as_state_database.html)

# CouchDB

To install

<https://docs.couchdb.org/en/latest/install/unix.html#installation-using-the-apache-couchdb-convenience-binary-packages>

to verify

<https://pouchdb.com/guides/setup-couchdb.html>

Browser

[http://localhost:5984/\\_utils/](http://localhost:5984/_utils/)

## Hyperledger Caliper

- Caliper is a blockchain benchmark framework which allows users to measure the performance of a specific blockchain implementation with a set of predefined use cases.
- Caliper will produce reports containing a number of performance indicators, such as TPS (Transactions Per Second), transaction latency, resource utilisation etc.
- The intent is for Caliper results to be used as a reference in supporting the choice of a blockchain implementation suitable for the user-specific use-cases.
- Given the variety of blockchain configurations, network setup, as well as the specific use-cases in mind, it is not intended to be an authoritative performance assessment, nor to be used for simple comparative purposes (e.g. blockchain A does 5 TPS and blockchain B does 10 TPS, therefore B is better).
- The Caliper project references the definitions, metrics, and terminology as defined by the Performance & Scalability Working Group (PSWG).

# CouchDB

## Key Characteristics

- A unified blockchain benchmark framework. We provide a common layer to integrate with major existing blockchain framework/platforms, so that the same benchmarks can be run for different blockchain systems. Some benchmark test environment will be provided to help different people run tests under the same environment, blockchain management tools like Hyperledger Cello could be integrated later to deploy and operate the environment. Also, users can use their existing environment and configure Caliper to run the test under the environment.
- A commonly accepted definition of performance indicators. You cannot compare an apple and a pear directly unless some common criteria are set. We will work closely with PSWG to provide a common definition of performance indicators that users care about, such as TPS, latency, resource utilization, etc.
- A set of commonly accepted benchmark cases. The goal of Caliper includes providing a set of easy-understandable benchmark cases so that each blockchain solution can be compared in various scenarios. This calls for much collaboration from PSWG, Requirement WG and other WG in Hyperledger community as well as blockchain practitioners to cover as many use cases that are of user's interest as possible.

## Installing and Running Caliper

<https://hyperledger.github.io/caliper/v0.4.2/installing-caliper/>

# Node js and npm

The current latest version of node and npm:

node version 10.9.0

npm version 6.4.0

But, the current [Hyperledger Fabric \(release version 1.2\) Prerequisites](#) says that there should be:

node version 8.9.x or greater (but not 9.x or 10.x)

npm version 5.6.0

<https://blog.chapagain.com.np/hyperledger-fabric-error-failed-to-load-grpc-binary-module-because-it-was-not-installed-for-the-current-system/>

# Node js and npm

## Solution

To solve this error, we need to downgrade or set the node and npm version to the one specified in the Prerequisites.

## Set NPM version to 5.6.0

```
npm install npm@5.6.0 -g
```

## Check NPM version

```
npm -v
```

## Set Nodejs version to 8.9.4

- The Prerequisites says that the node version should be 8.9.x or greater.
- But, it should be in the range of 8.9.x.
- Node version 9.x or 10.x will give the same error.
- Therefore, we set the node version to 8.9.4. This version has worked well in my case.

## Install NVM (Node Version Manager)

To set a specific node version, at first, we will install [NVM \(Node Version Manager\)](#).

You can run the nvm install script using either cURL or Wget.

using cURL:

```
curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.11/install.sh | bash
```

# Node js and npm

Now, open a new window or new tab in the terminal.

List out the commands available for nvm:

```
nvm --help
```

## **Install and use specific node version using nvm**

In our case, we will install and use node version 8.9.4.

```
nvm install v8.9.4 nvm use 8.9.4
```

After that, check the node version:

## **Check node version**

```
node -v
```



# Hyperledger fabric vs Composer

One of the significant differences between Hyperledger fabric vs Composer is that **Fabric is a framework, and Hyperledger Composer is a tool**. So, with Hyperledger Composer, you can create a new platform of framework similar to Hyperledger Fabric. But with Fabric, you can't do that.

## Hyperledger Fabric cheat sheet

<https://softwaremill.com/hyperledger-fabric-cheat-sheet/>