Use Case: Make a digital asset in BigchainDB and a chain instance. Add the asset to the chain.

Preconditions: We must have RethinkDB and BigchainDB installed and running.

Deployment and Execution:

1. In a Python shell, create a Big Chain instance

*python*

*from bigchaindb import Bigchain*

*b = Bigchain()*

1. Create a test user. This is identified by the public and private keys.

*testuser1\_priv, testuser1\_pub = b.generate\_keys()*

1. Definite a digital asset payload, and implement a transaction.

*tx = b.create\_transaction(b.me, testuser1\_pub, None, ‘CREATE’, payload=digital\_asset\_payload)*

1. The user must sign the transaction

*tx\_signed = b.sign\_transaction(tx, b.me\_previate)*

1. Write the transaction to the bigchain. It will be stored in a backlog where it will be validated, included in a block, and written to the bigchain

*b.write\_transaction(tx\_signed)*

Postconditions: See that the digital asset (“block”) is added to the chain. This can be done by reading the transaction from the bigchain. For example, this command,

*tx\_retrieved = b.get\_transaction(tx\_signed[‘id’])*