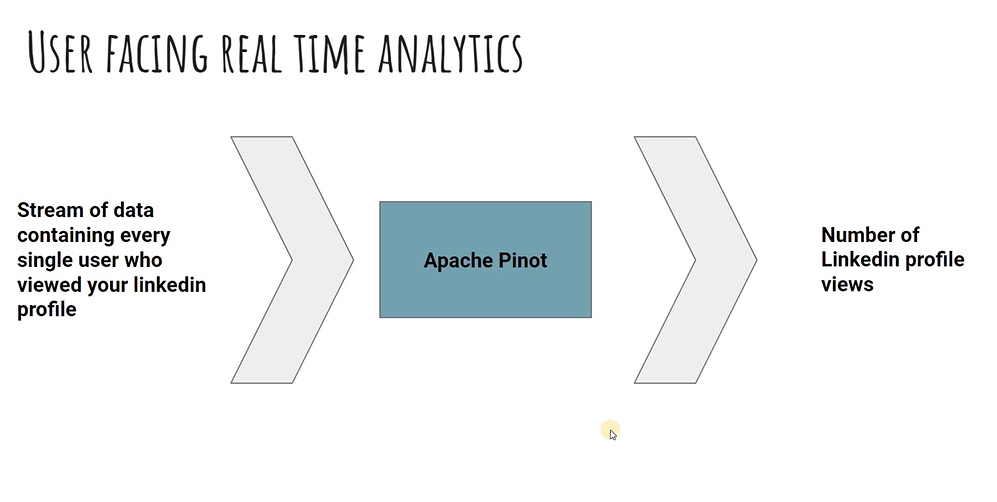
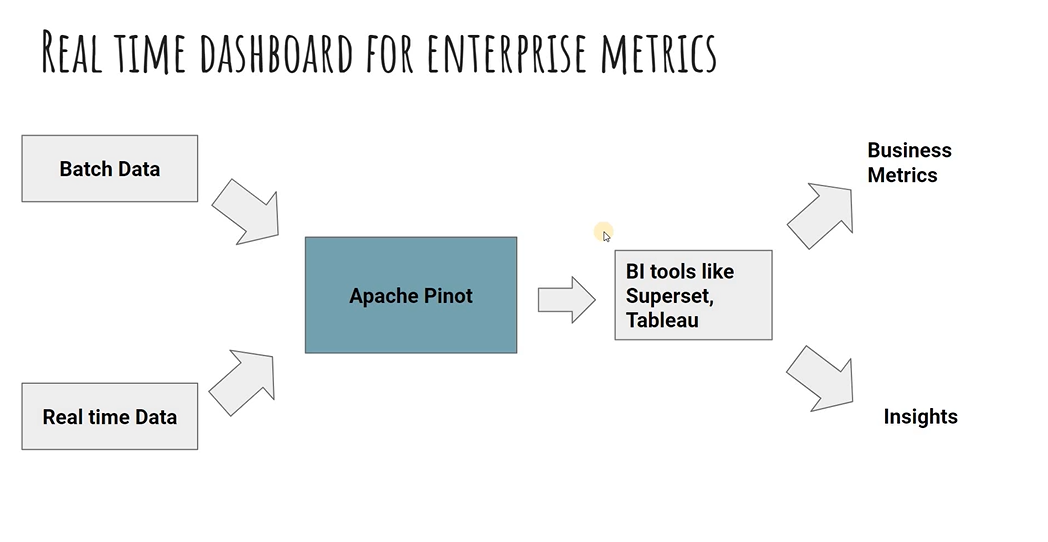
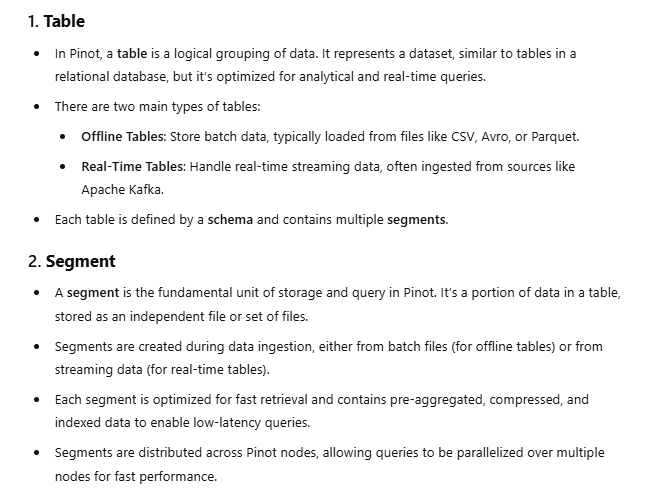
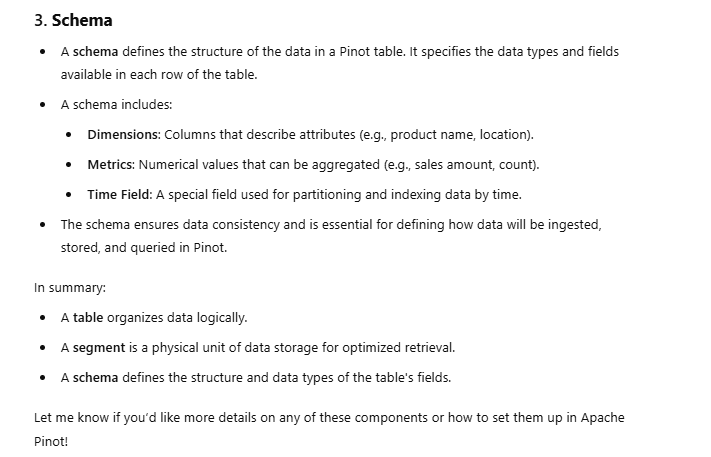
Apache Pinot

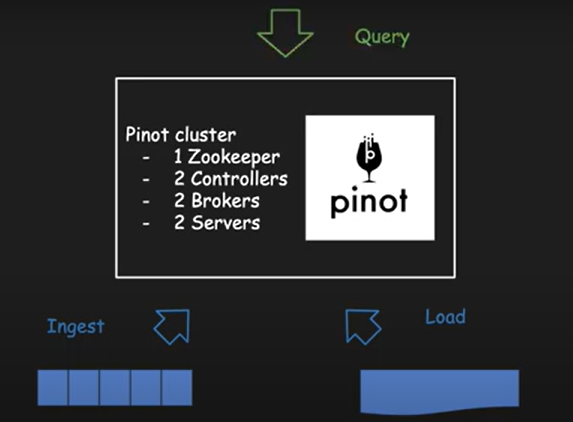
**About Apache Pinot:**

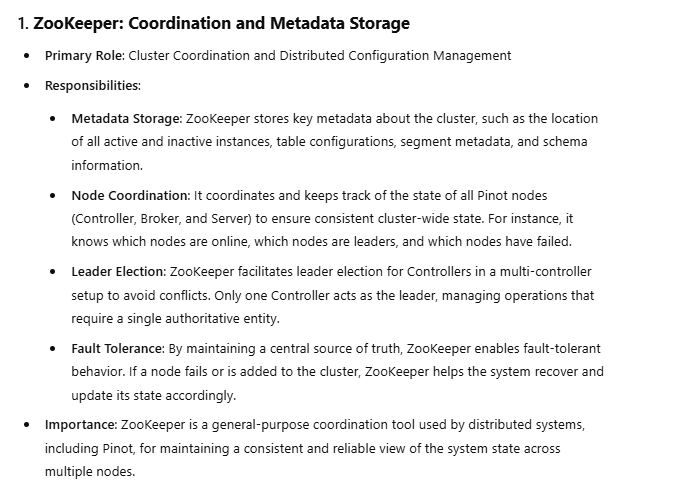


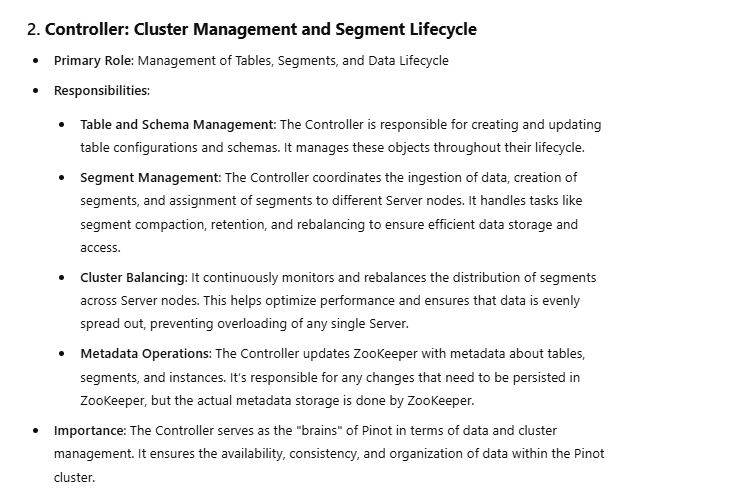


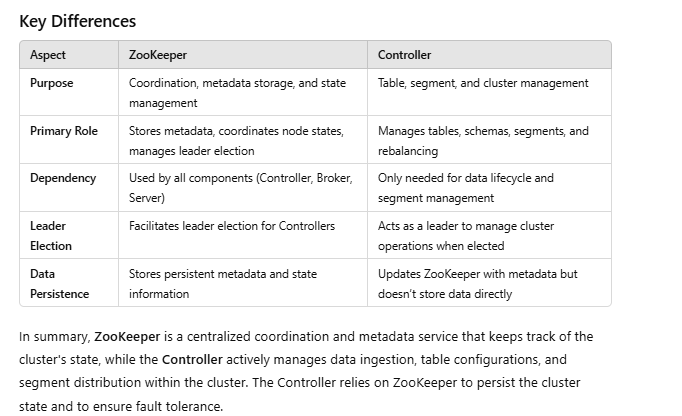


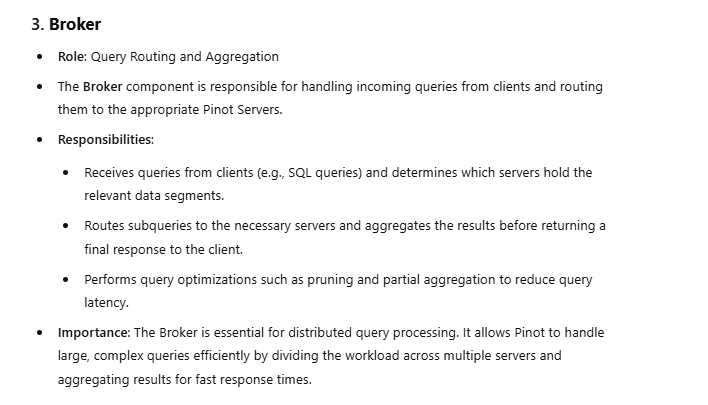


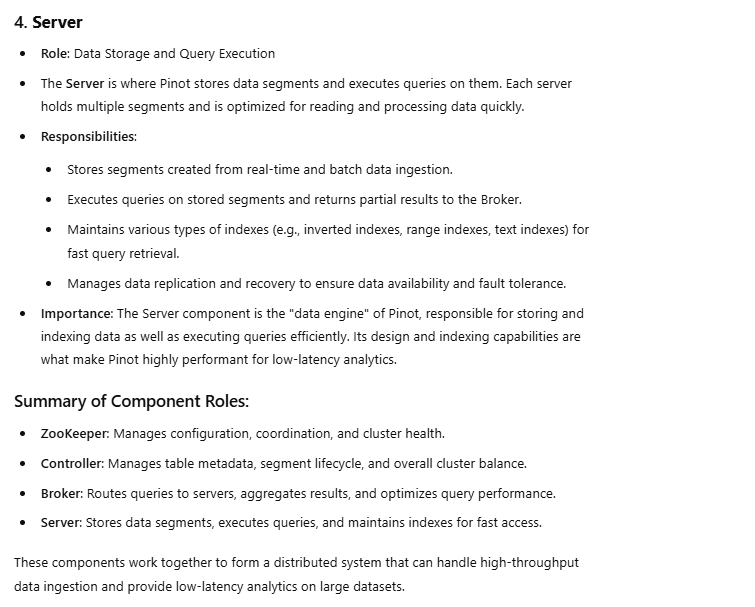
Apache Pinot Components:  


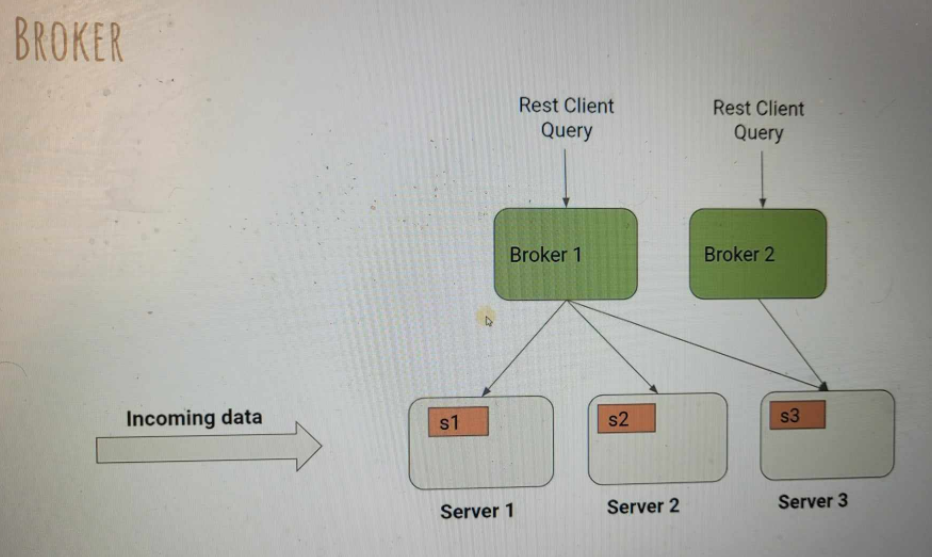




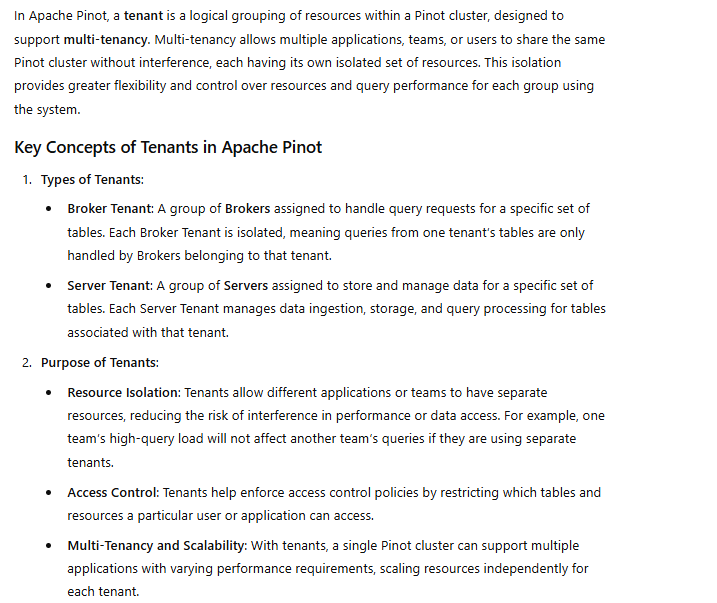


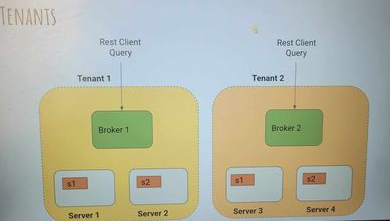




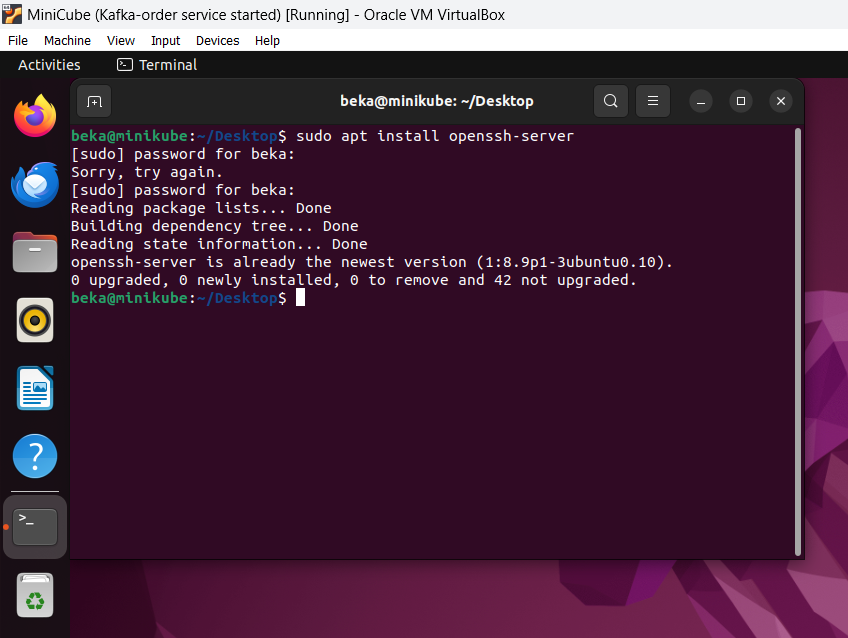


**Tenant**

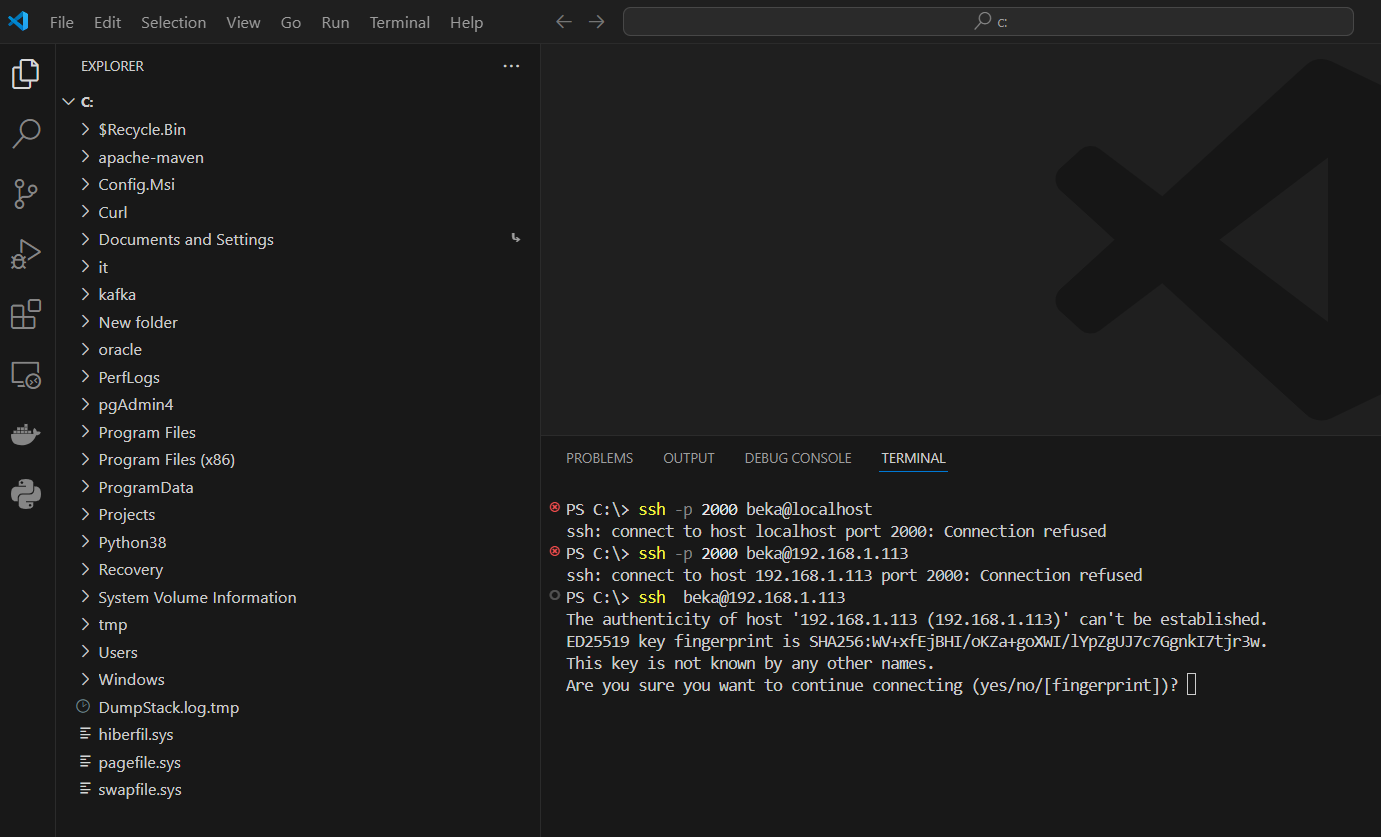




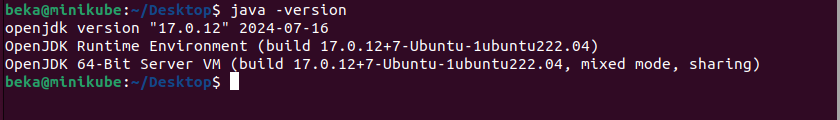
Environment Setup:



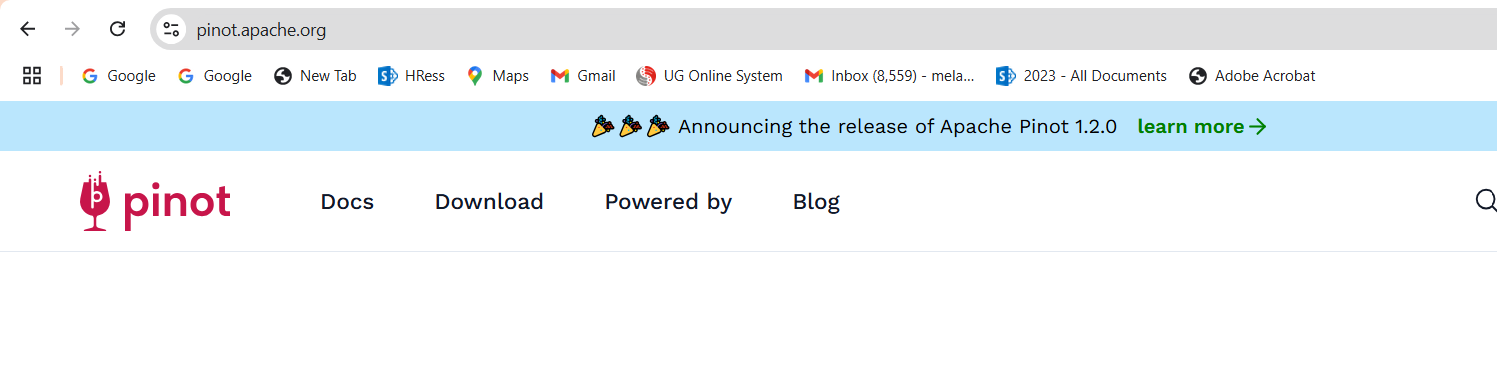
Install SSH server on your VM Host



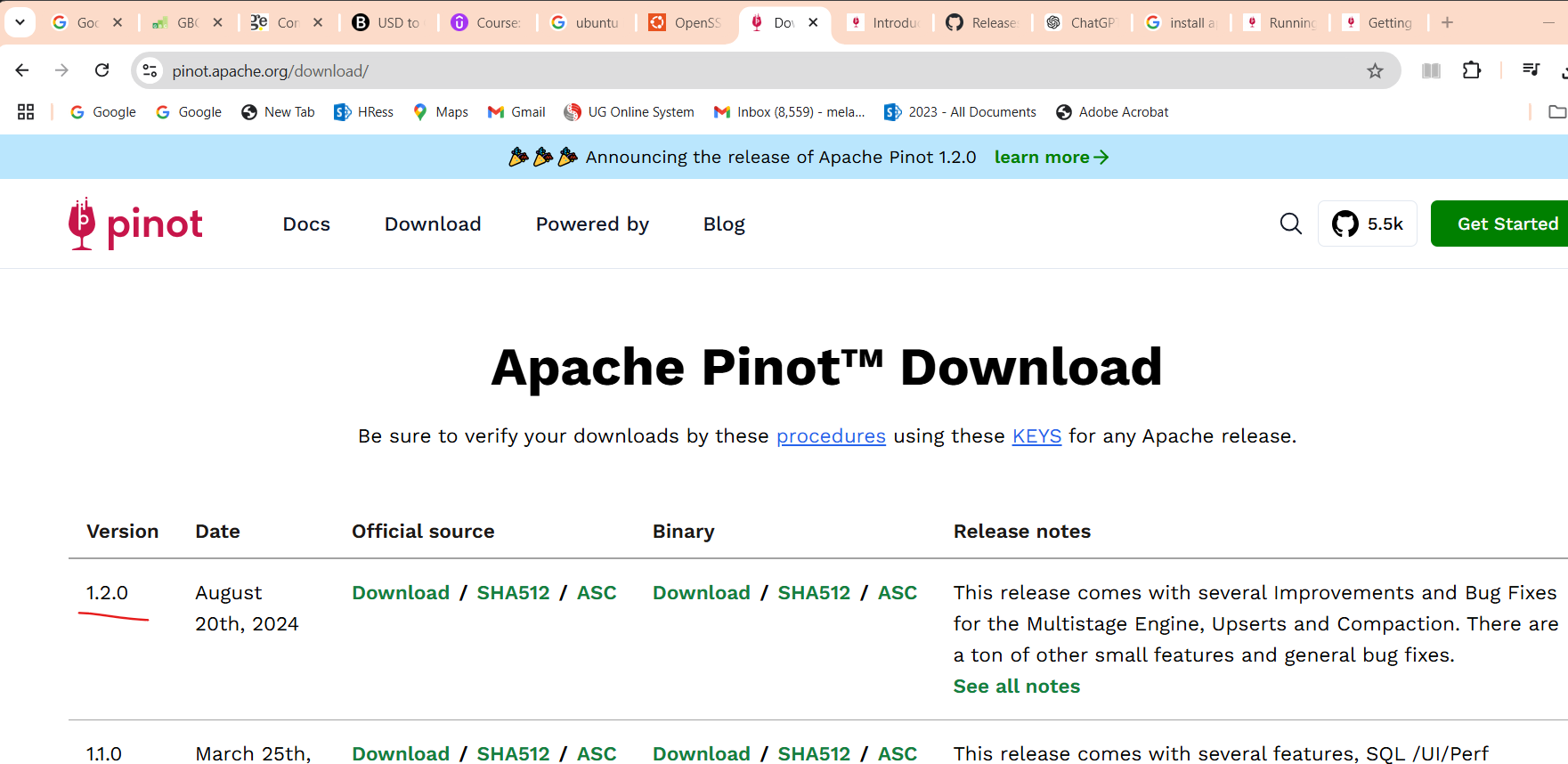
Type - y

Verify that java is installed:  


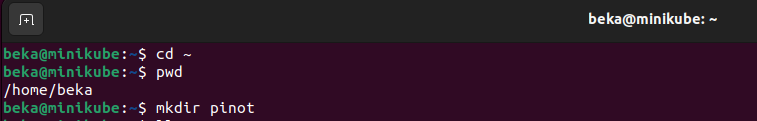
Go to the Apache Pinot official website:



Go to download section and see the latest version:



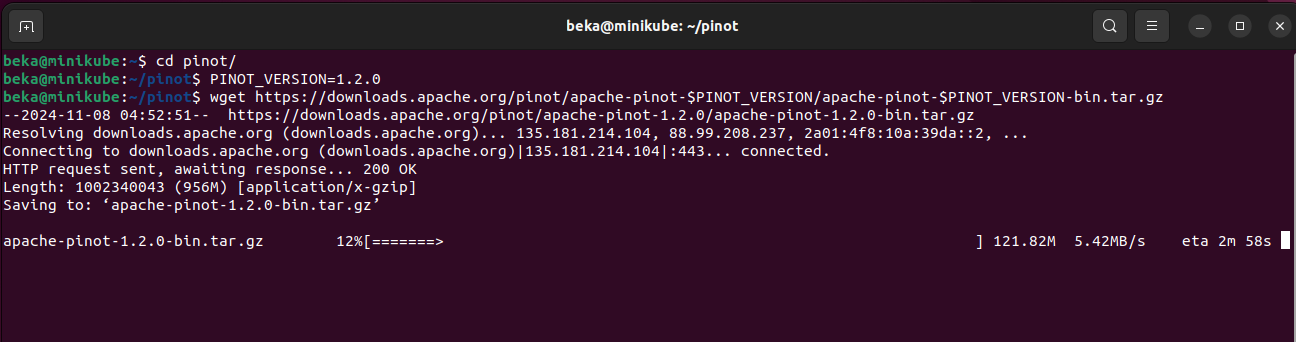
make pinot directory



Go to this directory and try to download it:

PINOT\_VERSION=1.2.0 #set to the Pinot version you decide to use

wget https://downloads.apache.org/pinot/apache-pinot-$PINOT\_VERSION/apache-pinot-$PINOT\_VERSION-bin.tar.gz



Now lets try to extract it:

tar -xzf apache-pinot-1.2.0-bin.tar.gz

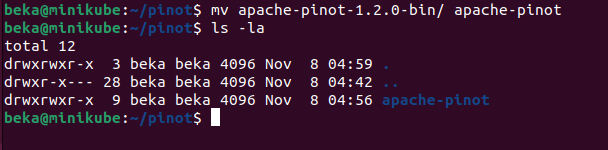
delete downloaded file:

rm -rf apache-pinot-1.2.0-bin.tar.gz

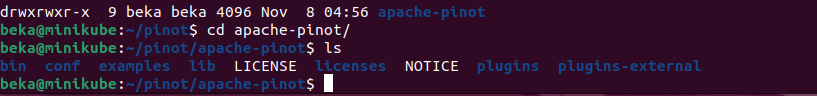


Rename extracted folder:

mv apache-pinot-1.2.0-bin/ apache-pinot



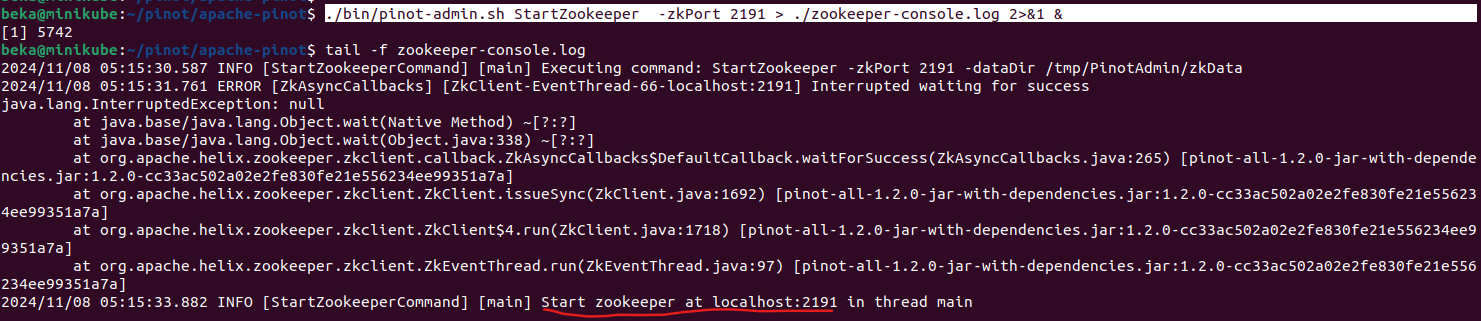
Lets go to the apache-pinot directory



Start zookeeper:

export JAVA\_OPTS="-Xms200m -Xmx200m"

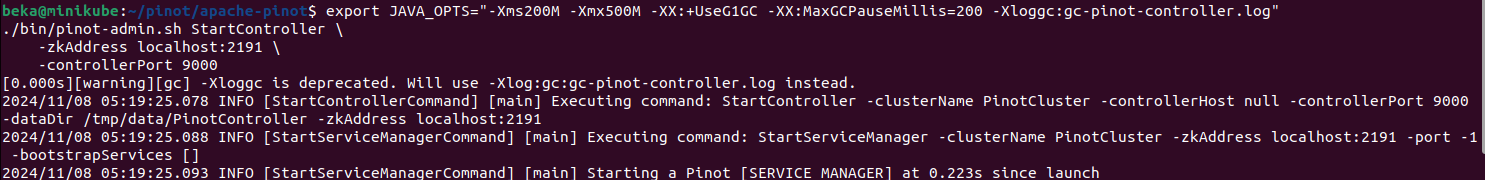
**./bin/pinot-admin.sh StartZookeeper -zkPort 2191 >** **./zookeeper-console.log 2>&1 &**



Start Controller

export JAVA\_OPTS="-Xms200m -Xmx200m"

./bin/pinot-admin.sh StartController -zkAddress localhost:2191 -controllerPort 9000 > **./controller-console.log 2>&1 &**



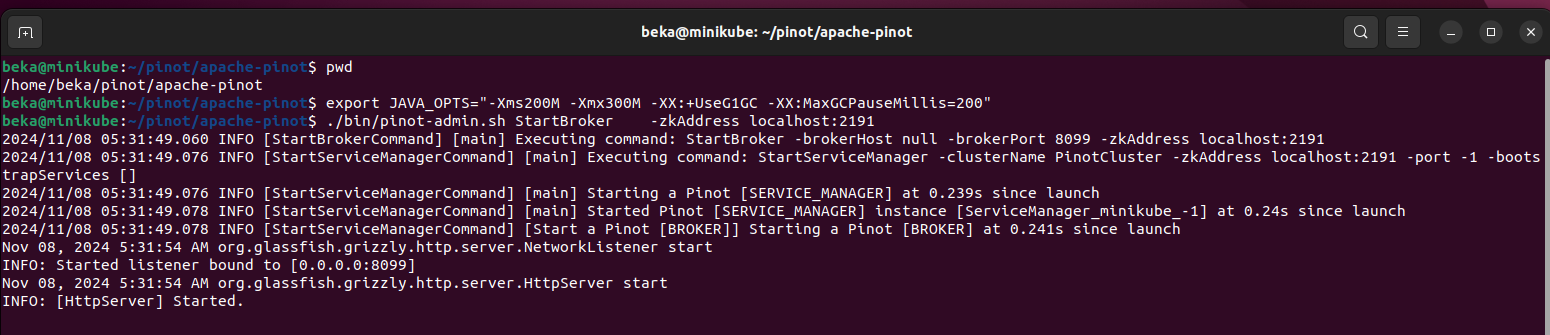
Finally:



Start Broker

export JAVA\_OPTS="-Xms200m -Xmx200m"

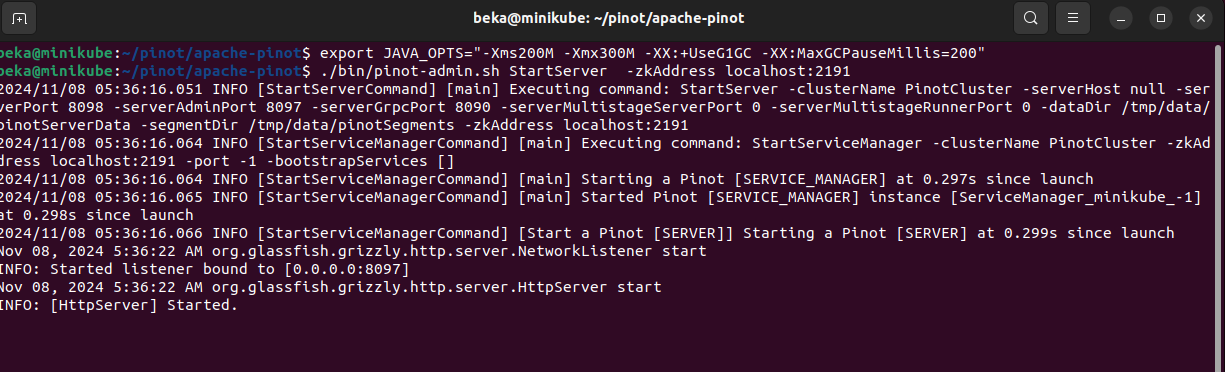
./bin/pinot-admin.sh StartBroker -zkAddress localhost:2191 > **./broker-console.log 2>&1 &**



Start Pinot Server

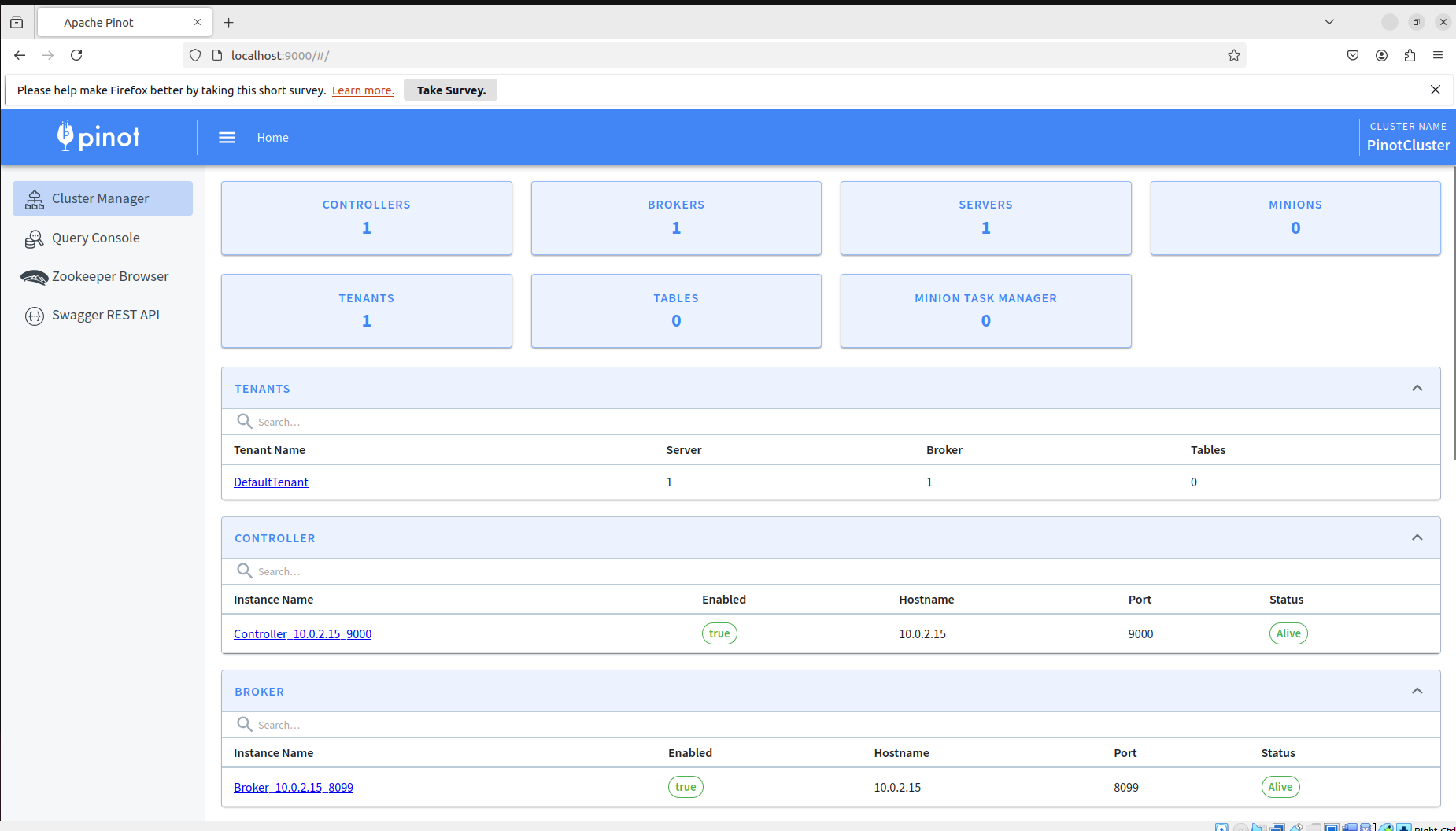
export JAVA\_OPTS="-Xms200m -Xmx200m"

./bin/pinot-admin.sh StartServer -zkAddress localhost:2191 > server**-console.log 2>&1 &**

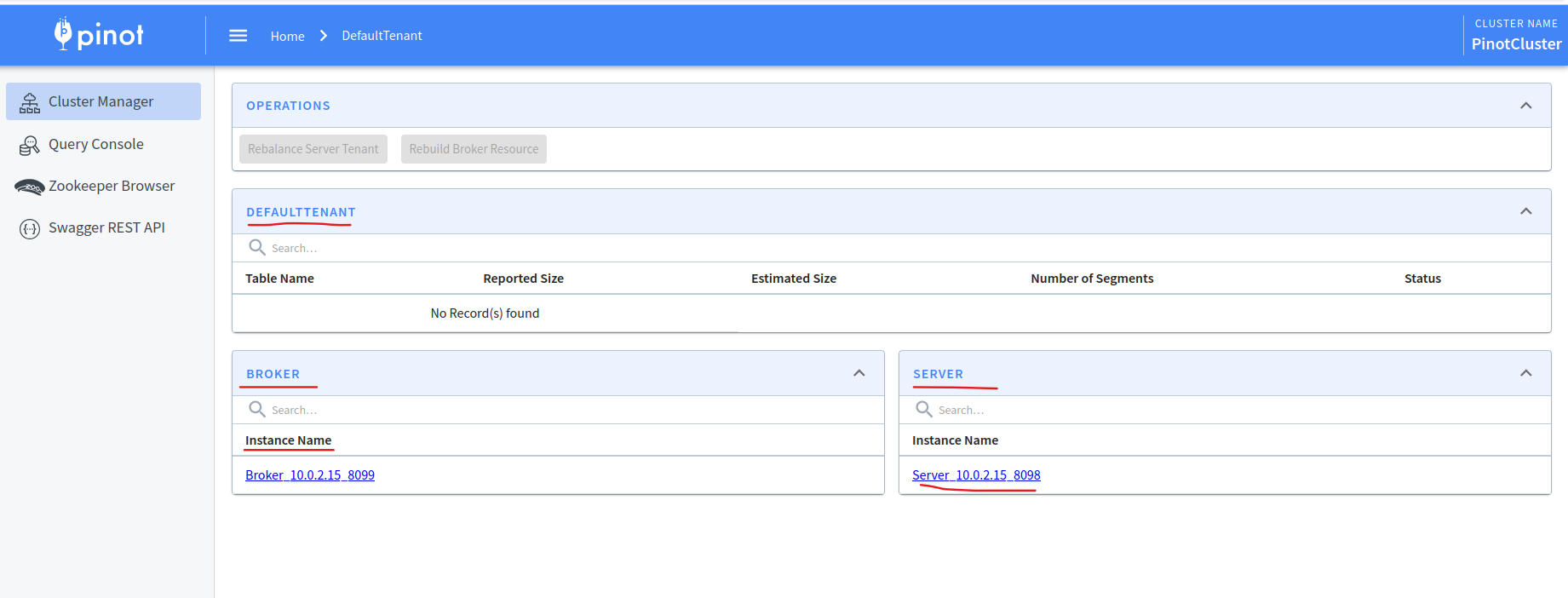


**Pinot Data Explorer:**

Lets Access http://localhost:9000



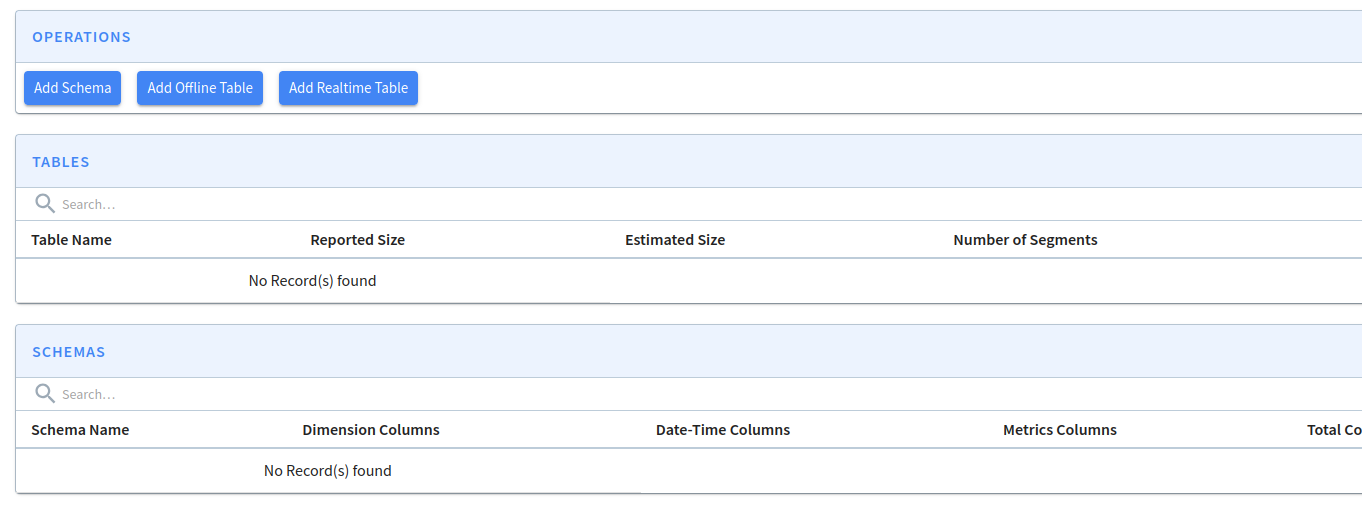
We can See that we have one default Tenant:



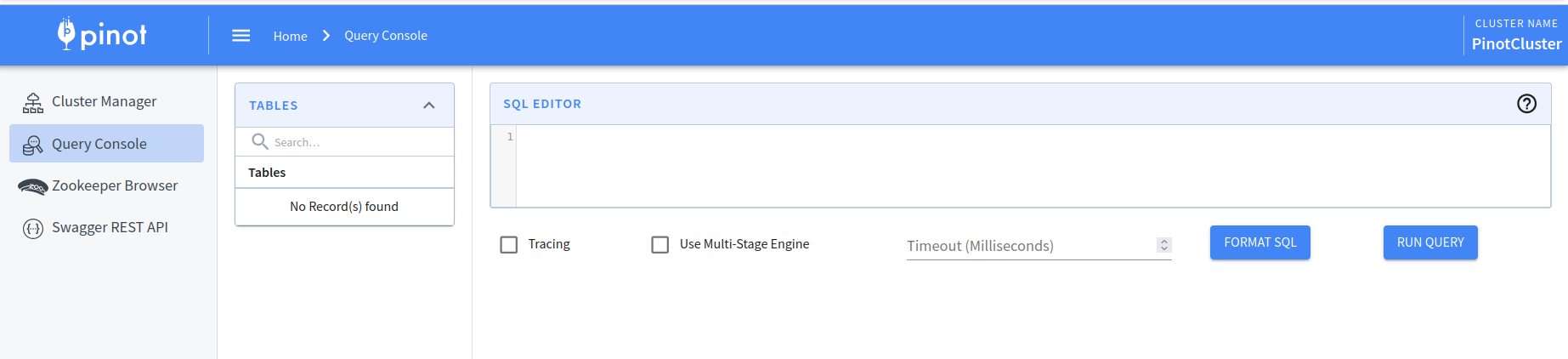
Go to the cluster manager and click once again Tables:



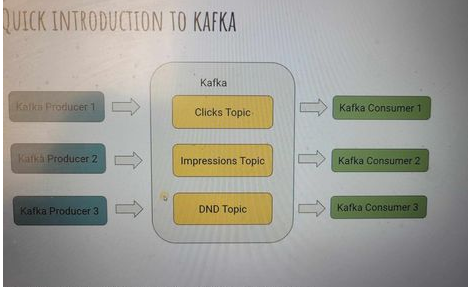
We have not created the tables yet:

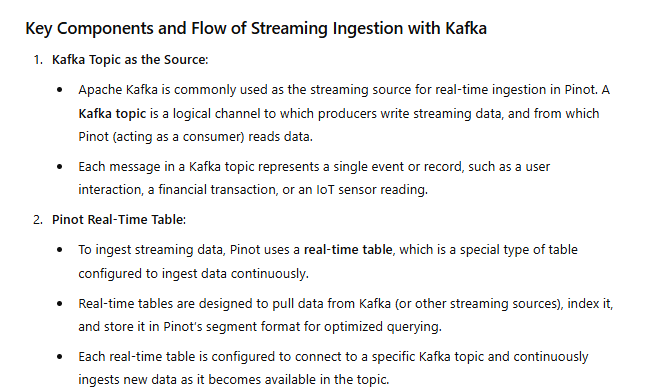


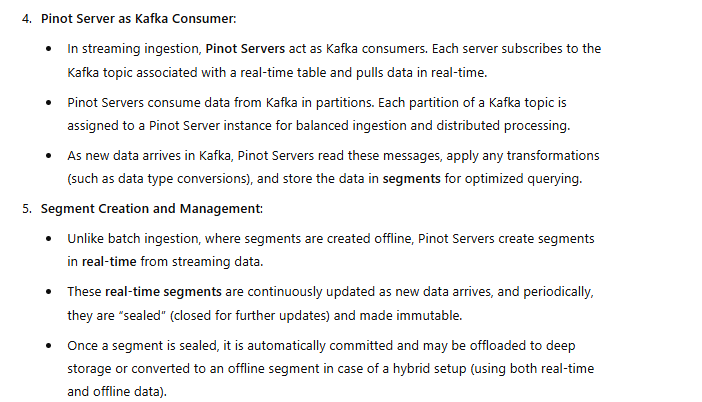
**Query Console:**

We can type several queries:  


Kafka

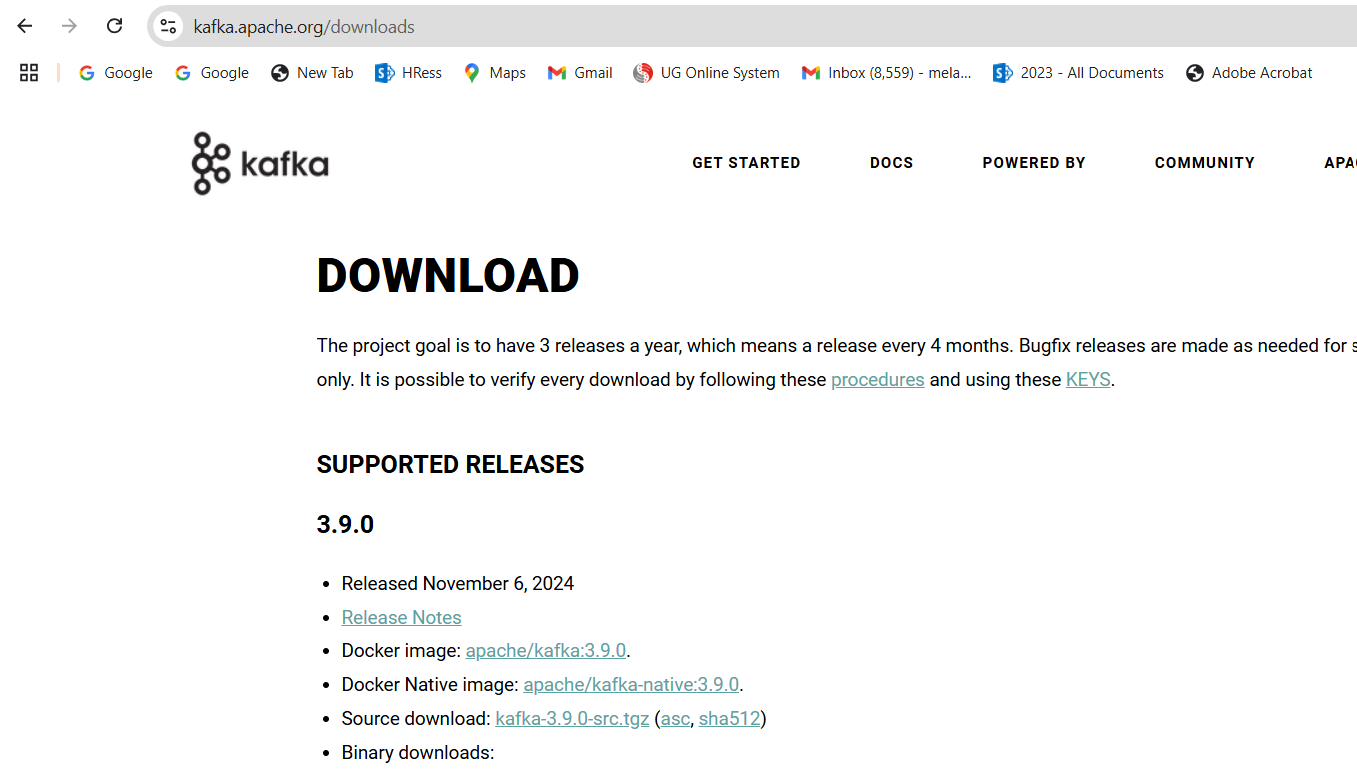






**Lets Install Kafka:**

Go to Kafka download page:



Run the following command:

wget <https://downloads.apache.org/kafka/3.9.0/kafka_2.12-3.9.0.tgz>



Lets extract Kafka from archive and lets go inside config directory:

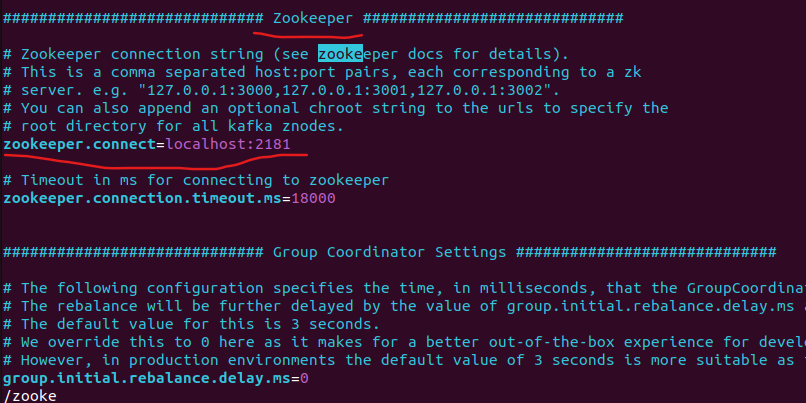




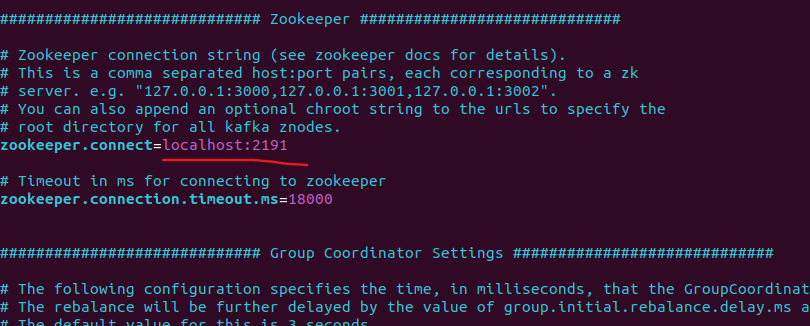
Lets modify server.properties file:

**vi server.properties**

****

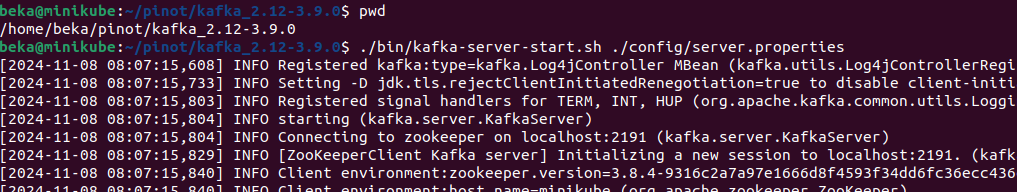


Change the port - 2181 to 2191



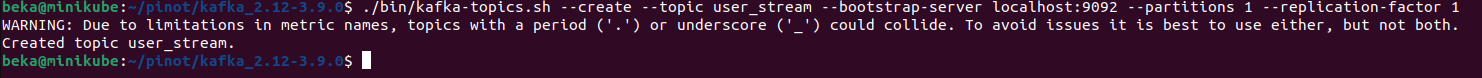
Start Kafka Server

./bin/kafka-server-start.sh ./config/server.properties



Create topic user\_stream:

./bin/kafka-topics.sh --create --topic user\_stream --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1



Now We should create the table In Pinot based on the following schema:

{

    "schemaName": "user\_stream",

    "dimensionFieldSpecs": [

      {

        "name": "userId",

        "dataType": "STRING"

      },

      {

        "name": "country",

        "dataType": "STRING"

      },

      {

        "name": "device",

        "dataType": "STRING"

      }

    ],

    "metricFieldSpecs": [

      {

        "name": "views",

        "dataType": "INT"

      }

    ],

    "dateTimeFieldSpecs": [{

      "name": "timestampInEpoch",

      "dataType": "LONG",

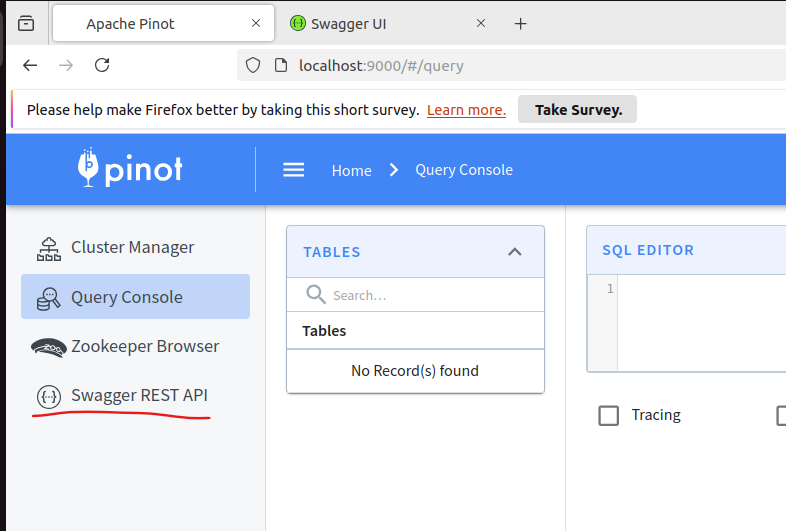
      "format" : "1:MILLISECONDS:EPOCH",

      "granularity": "1:MILLISECONDS"

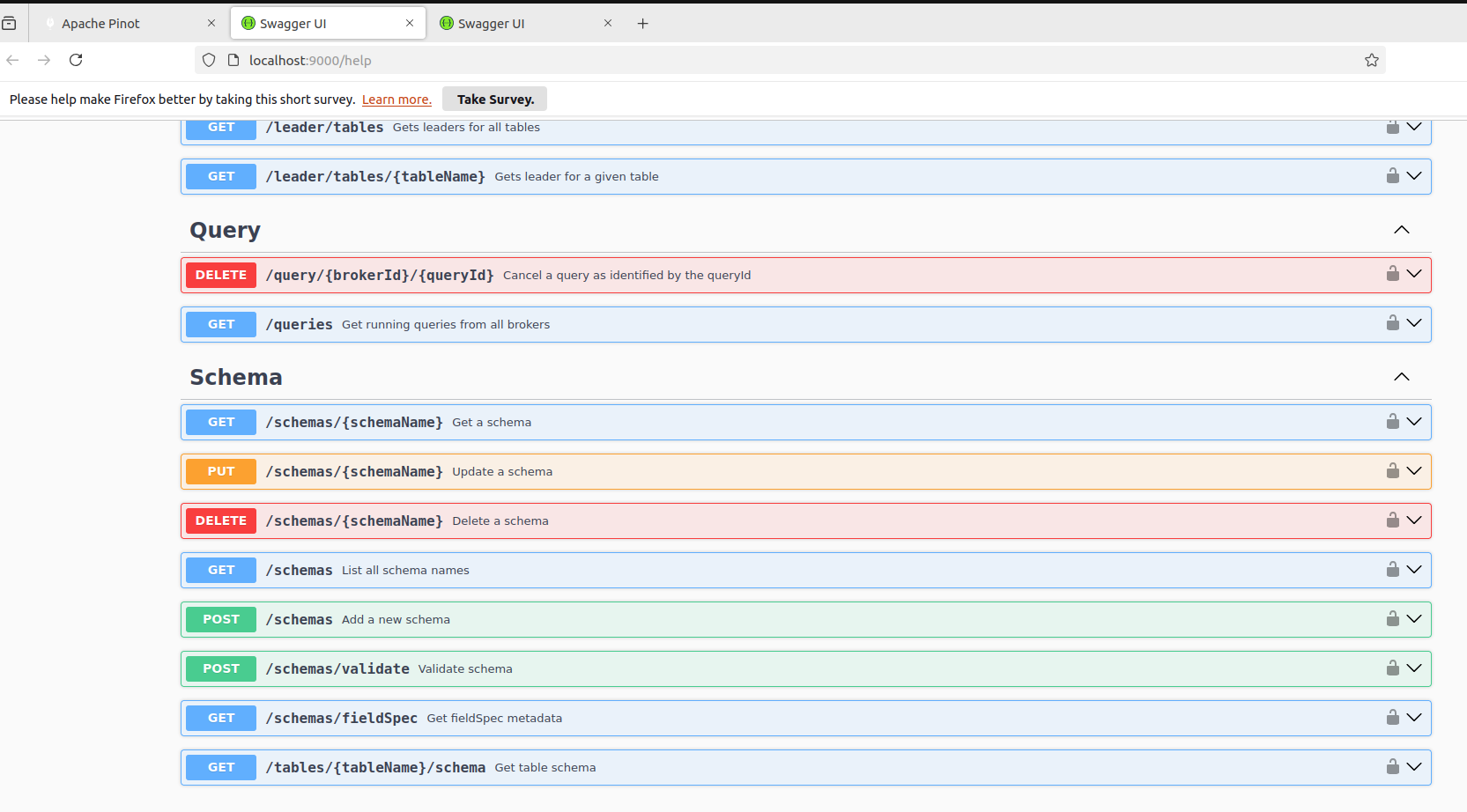
    }]

  }

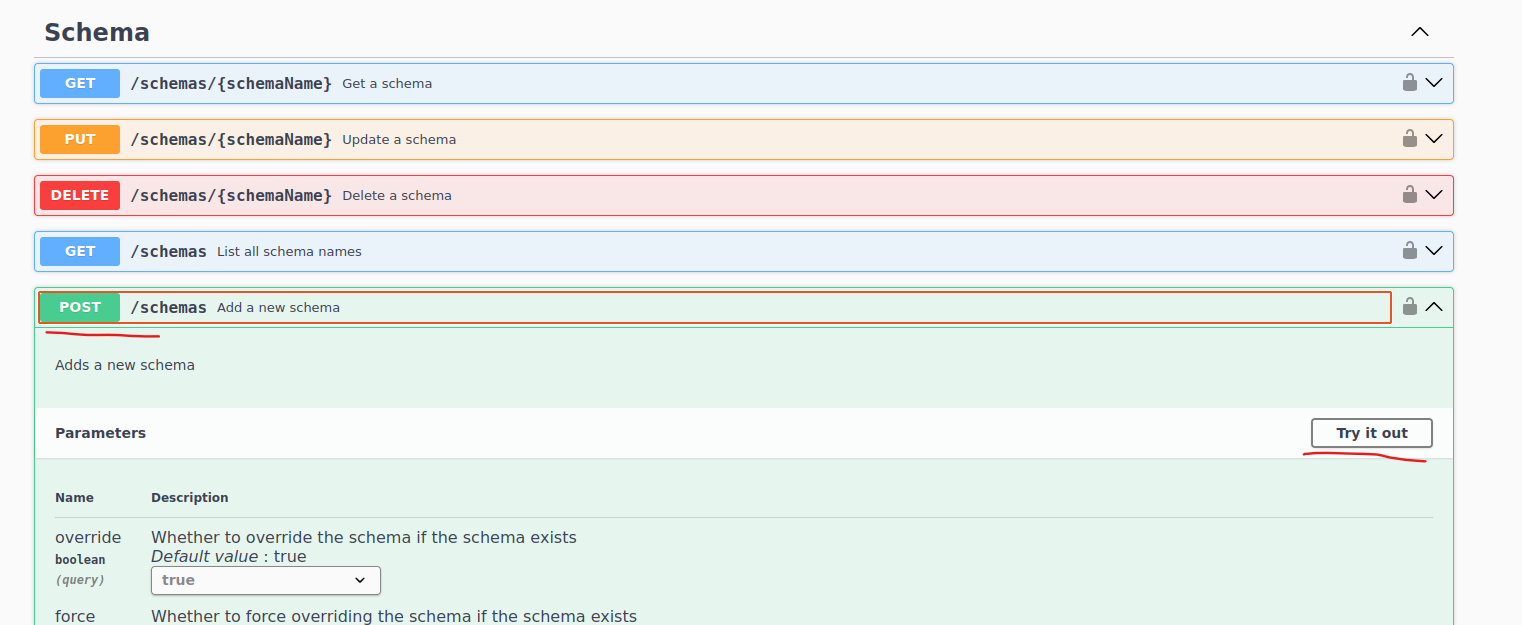
In Pinot web UI go to the swagger section:

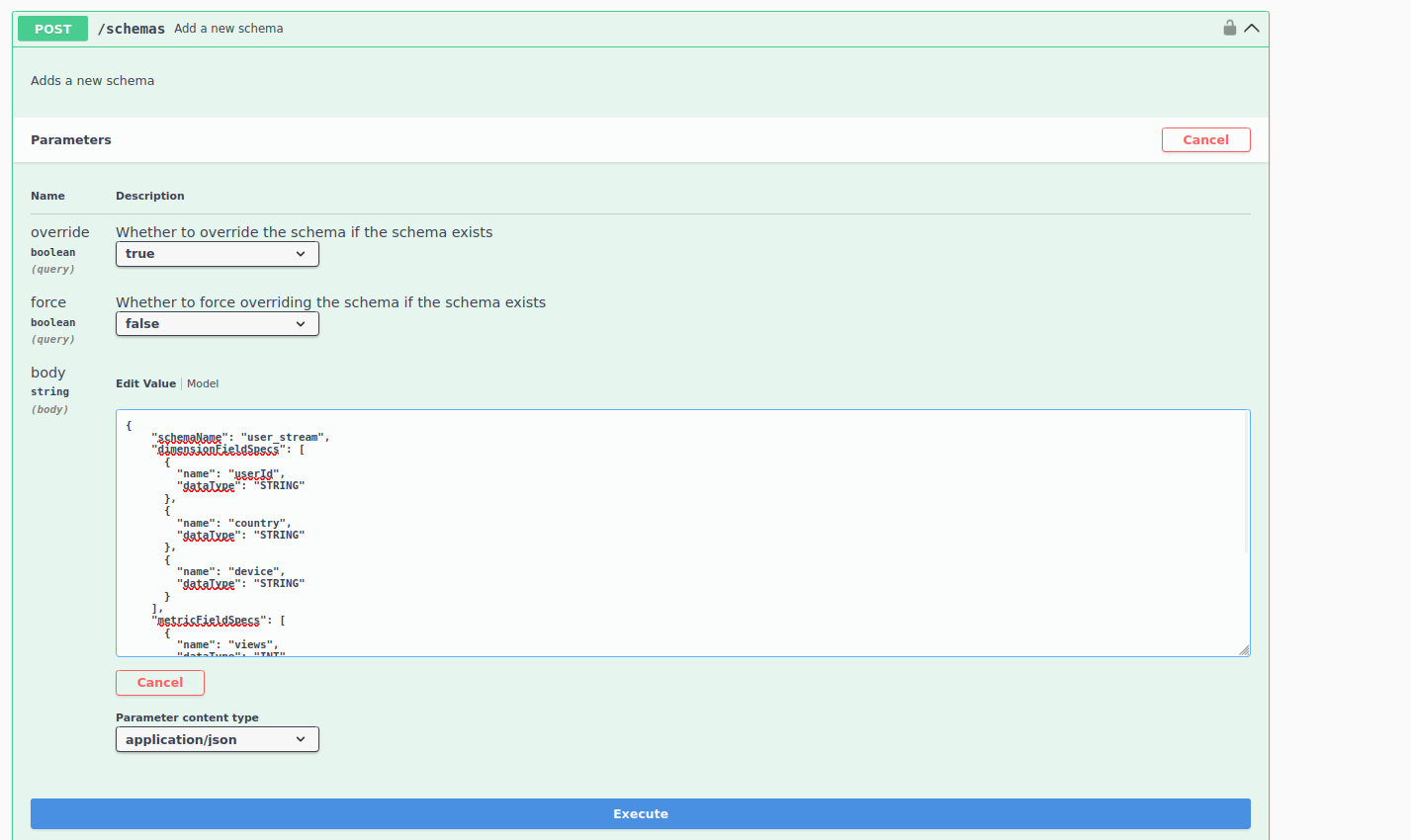


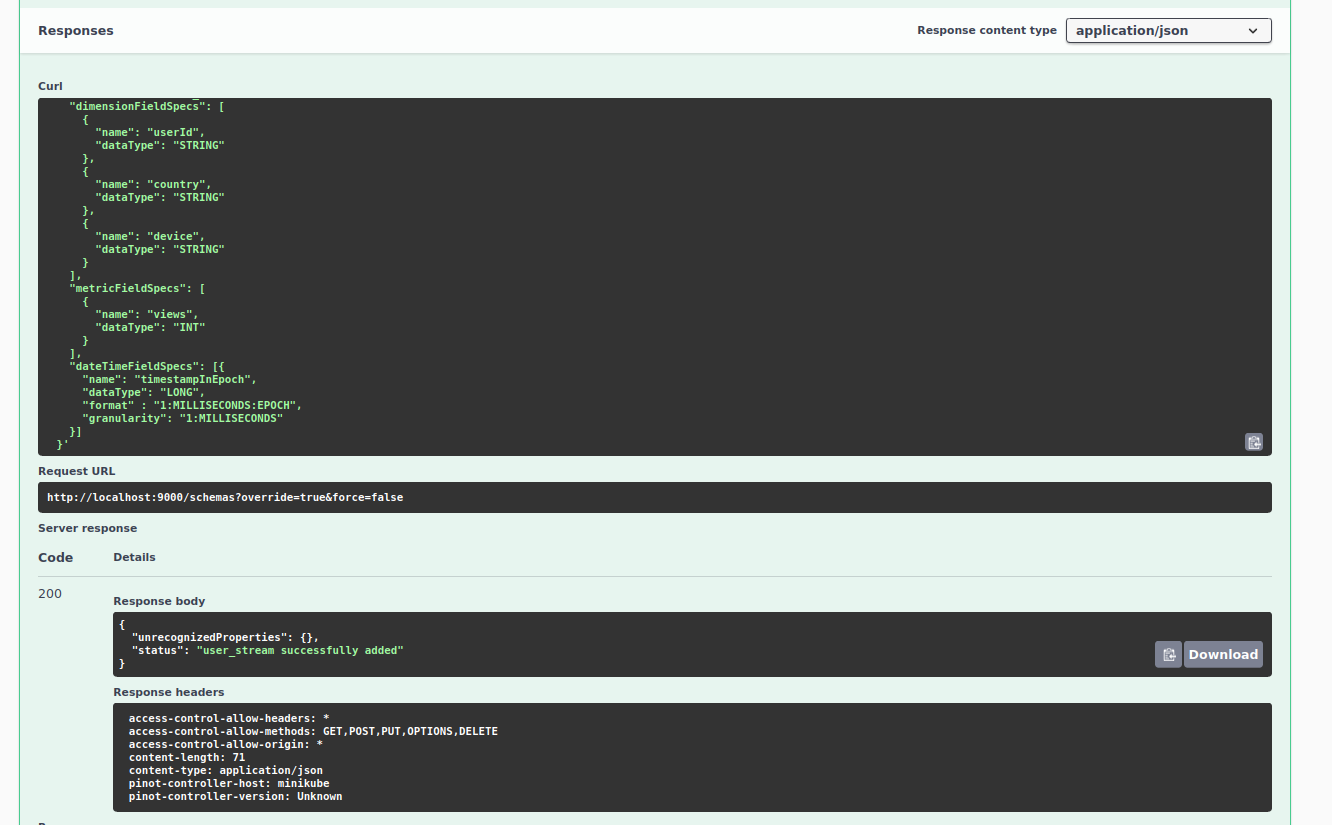
And move to “schema” section

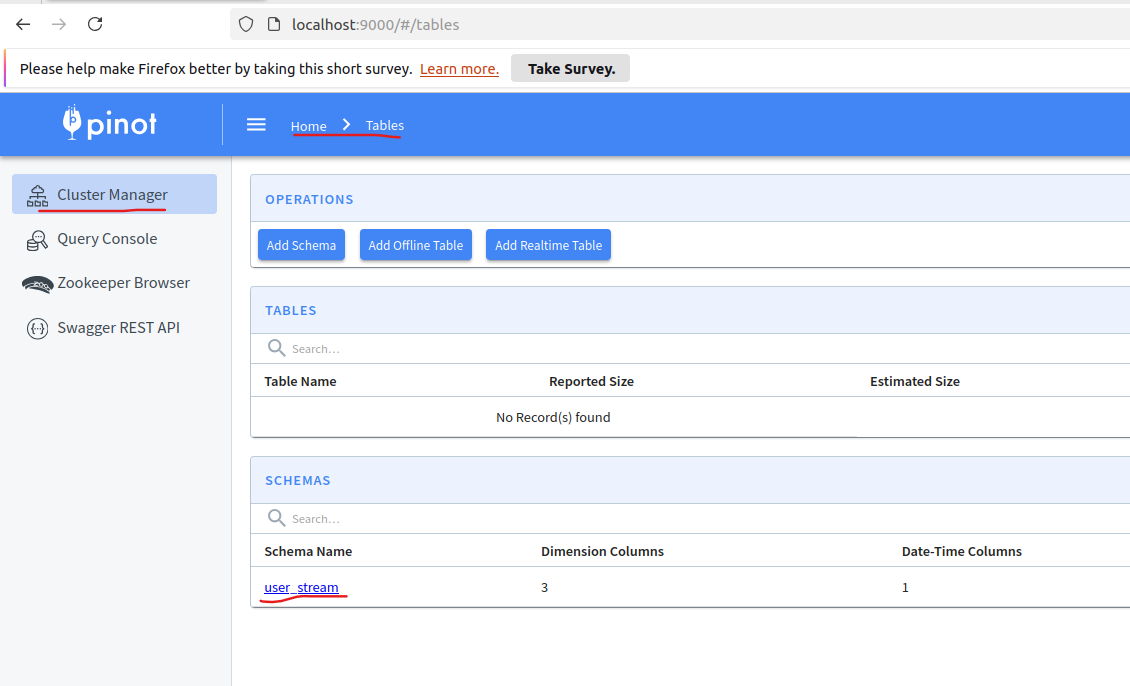


Choose Post /schemas

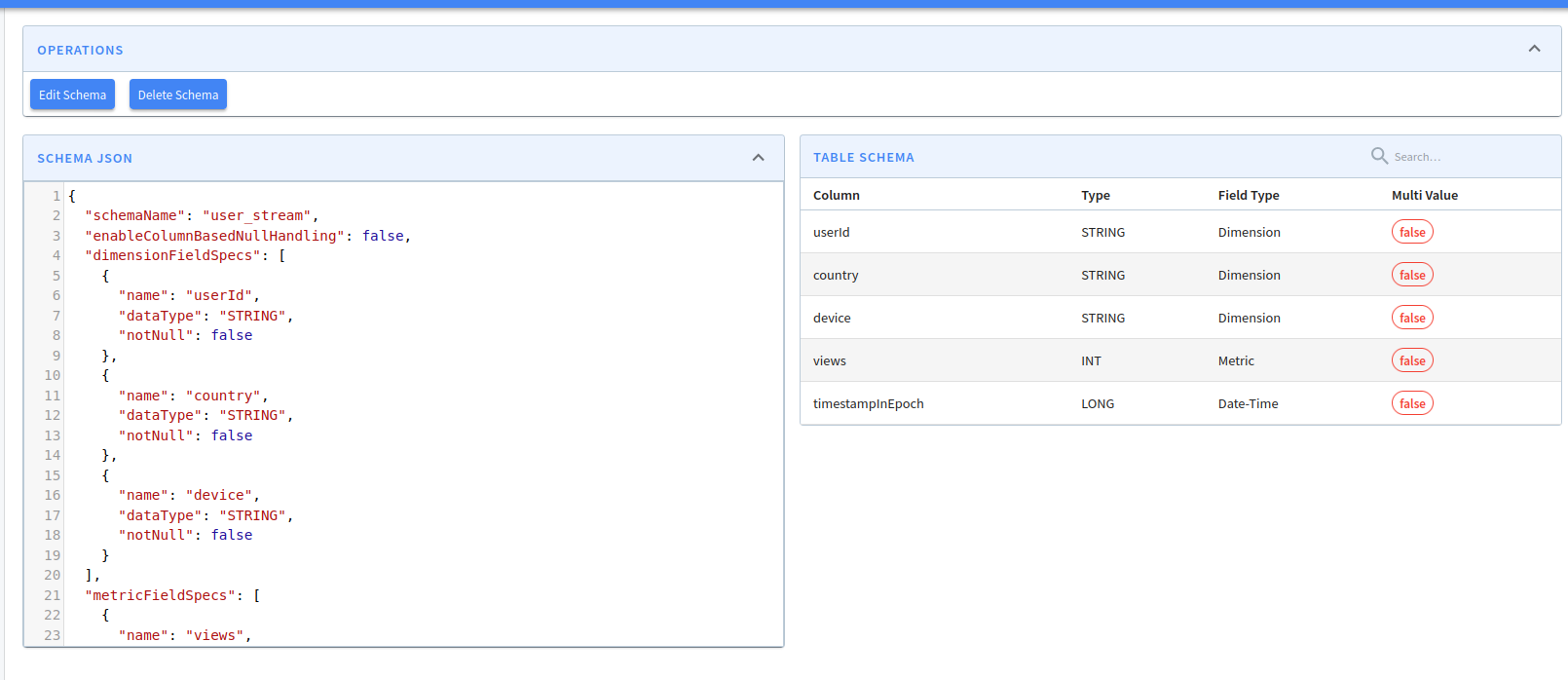


Paste the json and click execute:  






Click on user\_stream:



Now lets create the following table:  
  
{

    "tableName": "user\_stream",

    "tableType": "REALTIME",

    "segmentsConfig": {

      "timeColumnName": "timestampInEpoch",

      "timeType": "MILLISECONDS",

      "schemaName": "user\_stream",

      "replicasPerPartition": "1"

    },

    "tenants": {},

    "tableIndexConfig": {

      "loadMode": "MMAP",

      "streamConfigs": {

        "streamType": "kafka",

        "stream.kafka.consumer.type": "lowlevel",

        "stream.kafka.topic.name": "user\_stream",

        "stream.kafka.decoder.class.name": "org.apache.pinot.plugin.stream.kafka.KafkaJSONMessageDecoder",

        "stream.kafka.consumer.factory.class.name": "org.apache.pinot.plugin.stream.kafka20.KafkaConsumerFactory",

        "stream.kafka.broker.list": "localhost:9092",

        "realtime.segment.flush.threshold.time": "3600000",

        "realtime.segment.flush.threshold.size": "50000",

        "stream.kafka.consumer.prop.auto.offset.reset": "smallest"

      }

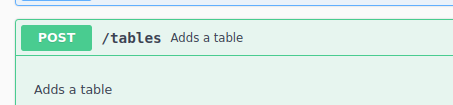
    },

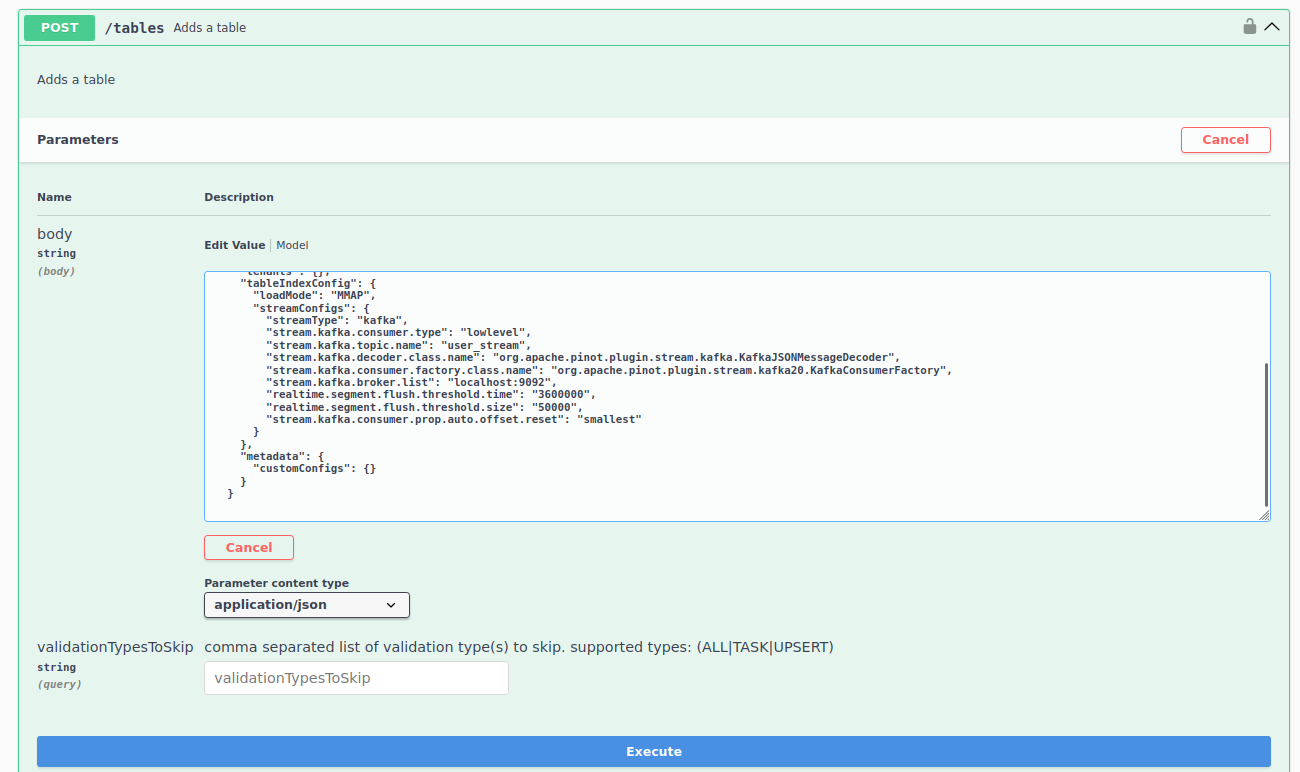
    "metadata": {

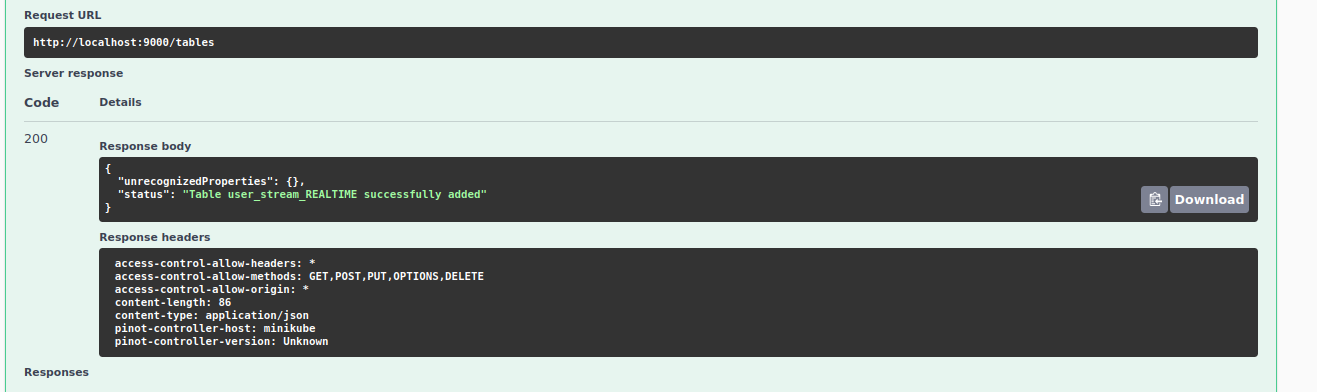
      "customConfigs": {}

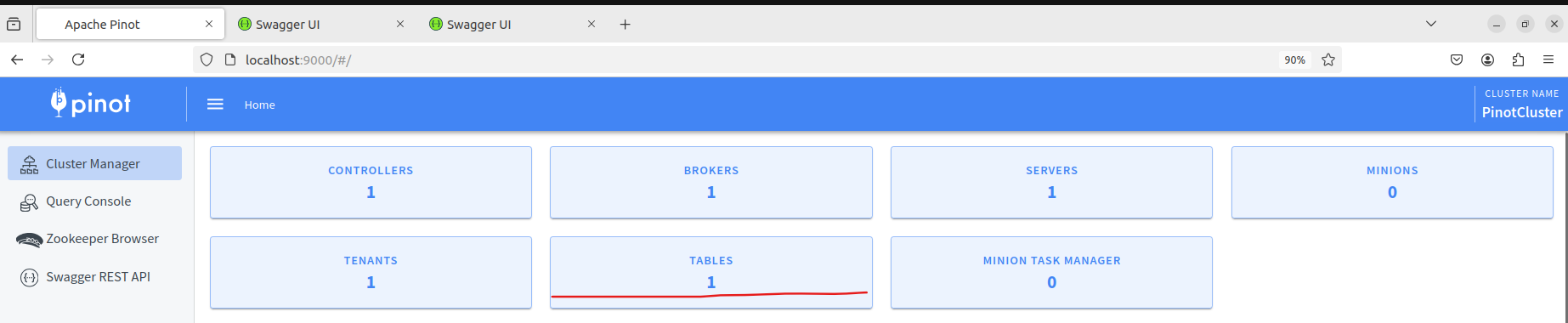
    }

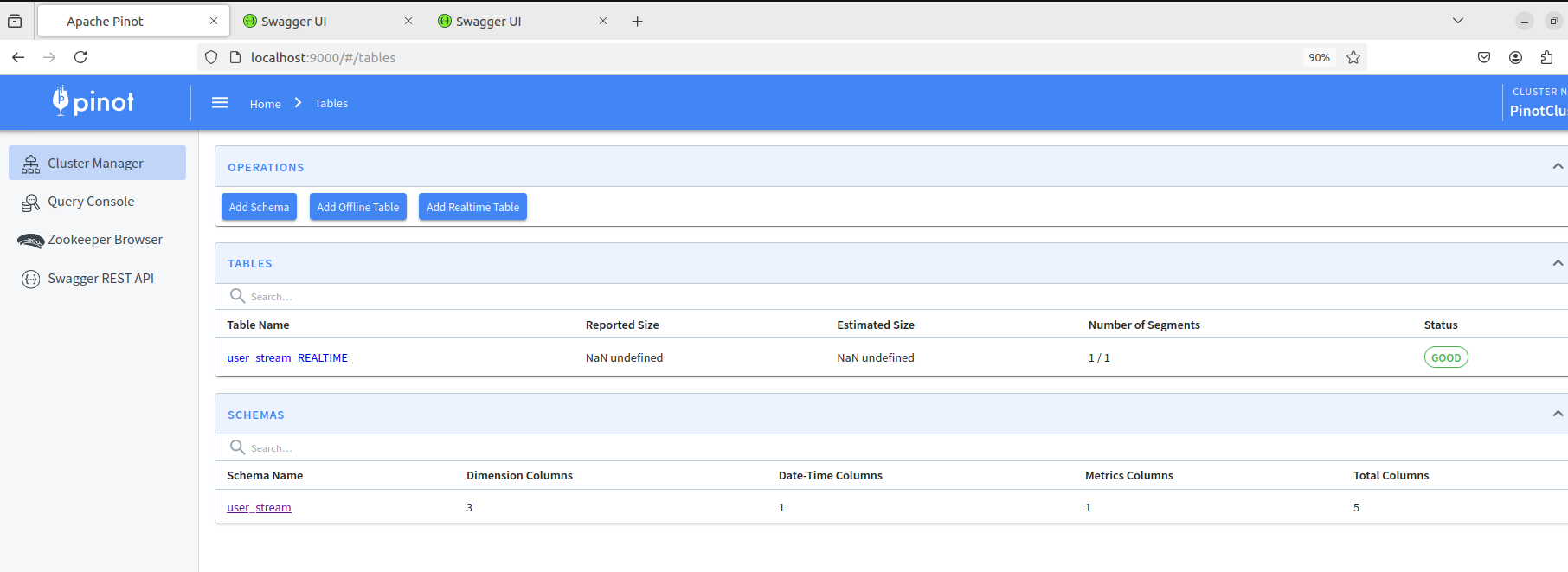
  }

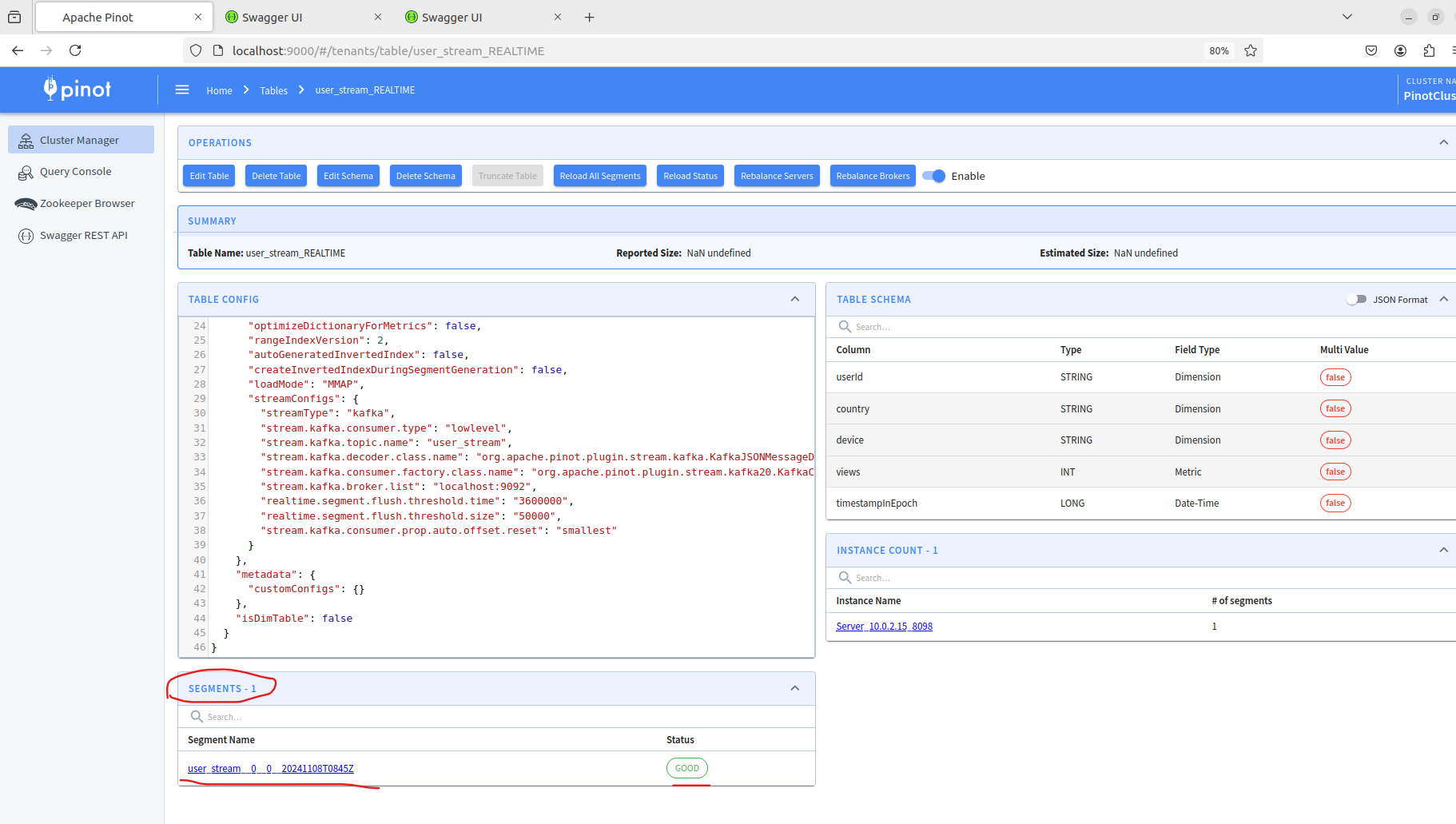
Now Lets go to the Table section and find:  












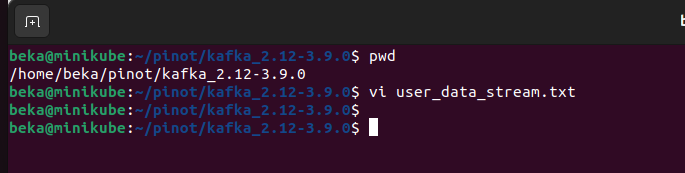
Lets see now the consuming segment:

Now lets go to the query console:



Now lets produce some data in kafka topic:

Create file : user\_stream\_data.txt:



And paste the following lines:

{ "userId": 1, "country": "USA", "device": "Android", "views": 100, "timestampInEpoch": 1620050945000}

{ "userId": 2, "country": "India", "device": "Android", "views": 300, "timestampInEpoch": 1620050945001}

{ "userId": 3, "country": "Germany", "device": "Android", "views": 200, "timestampInEpoch": 1620050945002}

{ "userId": 4, "country": "USA", "device": "IOS", "views": 500, "timestampInEpoch": 1620050945003}

{ "userId": 5, "country": "USA", "device": "Android", "views": 1200, "timestampInEpoch": 1620050945004}

{ "userId": 6, "country": "India", "device": "Android", "views": 400, "timestampInEpoch": 1620050945005}

{ "userId": 7, "country": "Germany", "device": "Android", "views": 1800, "timestampInEpoch": 1620050945006}

{ "userId": 8, "country": "USA", "device": "IOS", "views": 1900, "timestampInEpoch": 1620050945007}

{ "userId": 9, "country": "France", "device": "IOS", "views": 600, "timestampInEpoch": 1620050945008}

{ "userId": 10, "country": "Canada", "device": "Android", "views": 800, "timestampInEpoch": 1620050945009}

{ "userId": 11, "country": "USA", "device": "Android", "views": 100, "timestampInEpoch": 1620050945010}

{ "userId": 12, "country": "India", "device": "Android", "views": 300, "timestampInEpoch": 1620050945011}

{ "userId": 13, "country": "Germany", "device": "Android", "views": 200, "timestampInEpoch": 1620050945012}

{ "userId": 14, "country": "USA", "device": "IOS", "views": 500, "timestampInEpoch": 1620050945013}

{ "userId": 15, "country": "USA", "device": "Android", "views": 1200, "timestampInEpoch": 1620050945014}

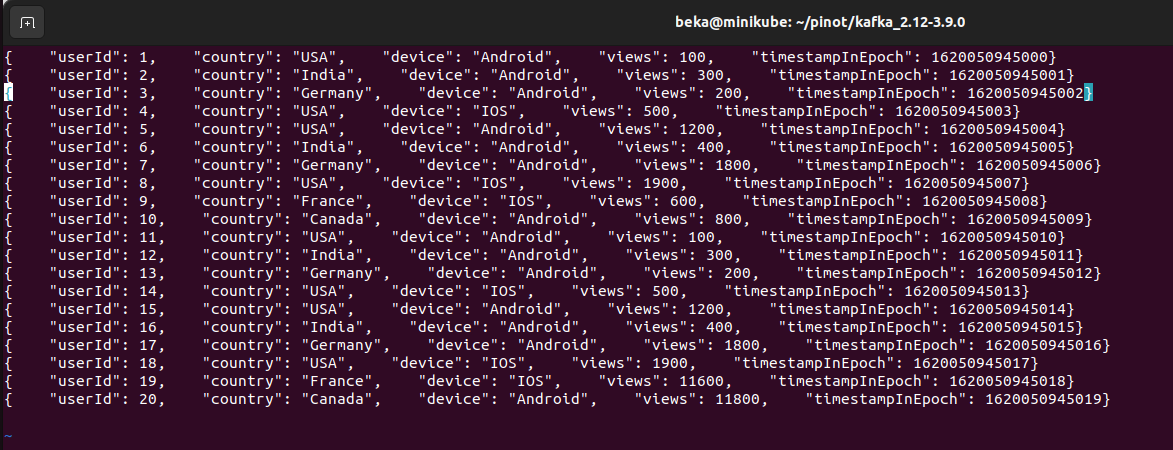
{ "userId": 16, "country": "India", "device": "Android", "views": 400, "timestampInEpoch": 1620050945015}

{ "userId": 17, "country": "Germany", "device": "Android", "views": 1800, "timestampInEpoch": 1620050945016}

{ "userId": 18, "country": "USA", "device": "IOS", "views": 1900, "timestampInEpoch": 1620050945017}

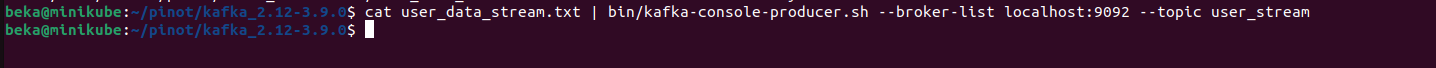
{ "userId": 19, "country": "France", "device": "IOS", "views": 11600, "timestampInEpoch": 1620050945018}

{ "userId": 20, "country": "Canada", "device": "Android", "views": 11800, "timestampInEpoch": 1620050945019}

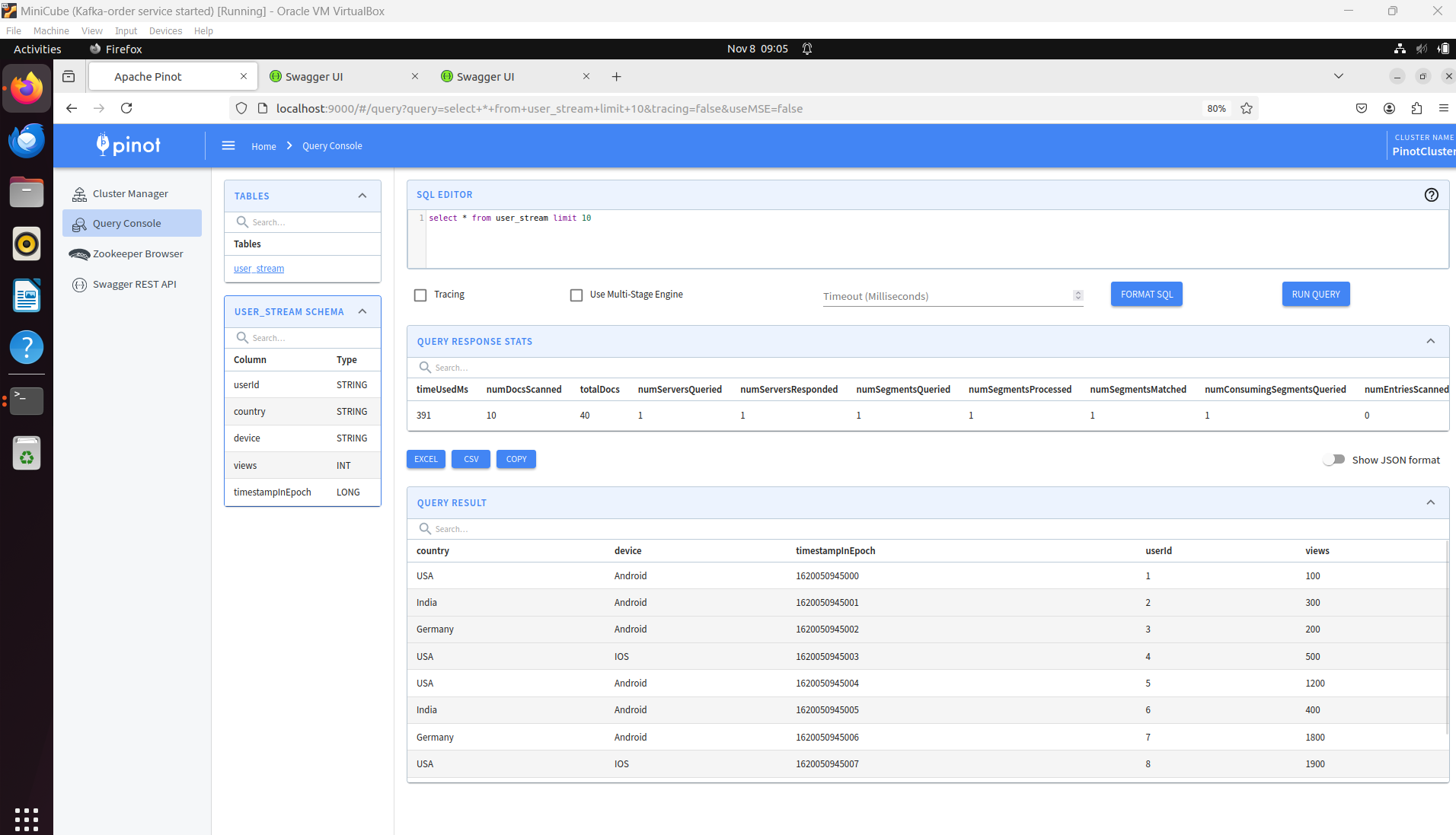


Pass this data to kafka producer:

cat user\_data\_stream.txt | bin/kafka-console-producer.sh --broker-list localhost:9092 --topic user\_stream

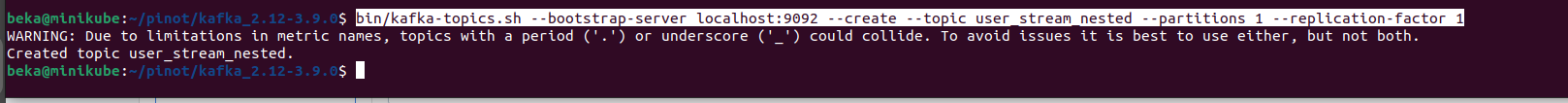


Run the same command once again:



Create another Topic

bin/kafka-topics.sh --bootstrap-server localhost:9092 --create --topic user\_stream\_nested --partitions 1 --replication-factor 1



Next Create again a schema in Pinot:

{

    "schemaName": "user\_stream\_nested1",

    "dimensionFieldSpecs": [

      {

        "name": "userstr",

        "dataType": "STRING"

      }

    ],

    "dateTimeFieldSpecs": [

      {

        "name": "last\_login",

        "dataType": "LONG",

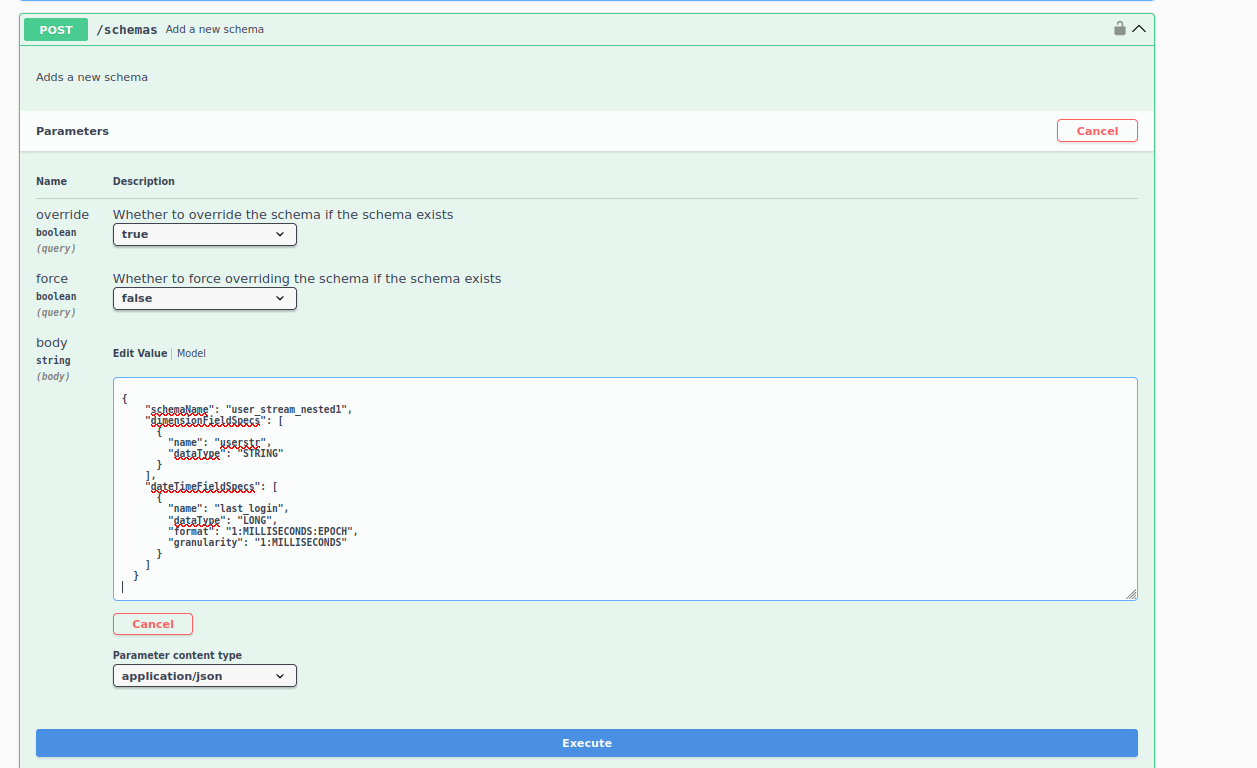
        "format": "1:MILLISECONDS:EPOCH",

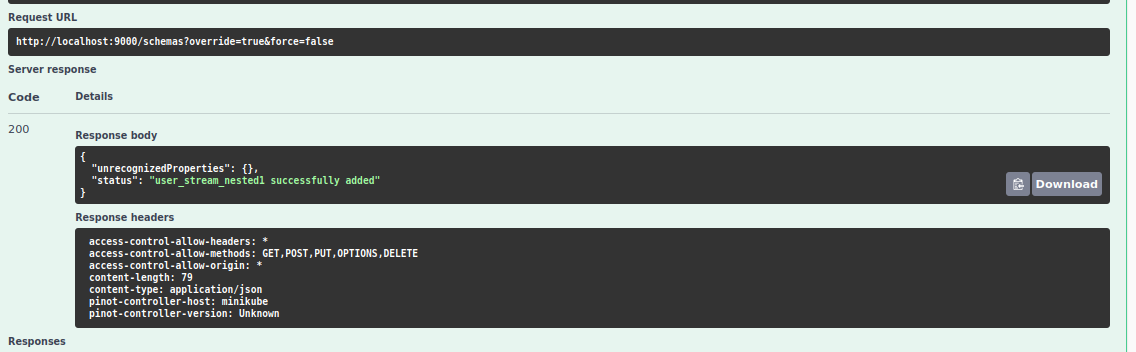
        "granularity": "1:MILLISECONDS"

      }

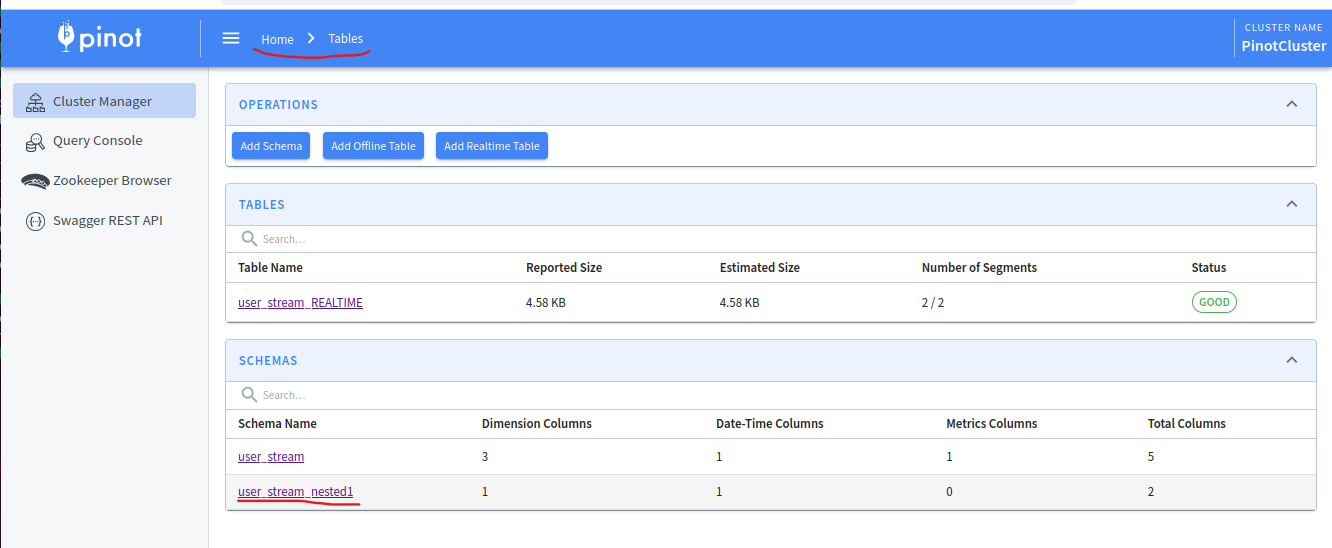
    ]

  }





<https://www.udemy.com/course/apache-pinot-a-hands-on-course/learn/lecture/26580692#learning-tools>



If you click on it

