

Jeremy Morrison

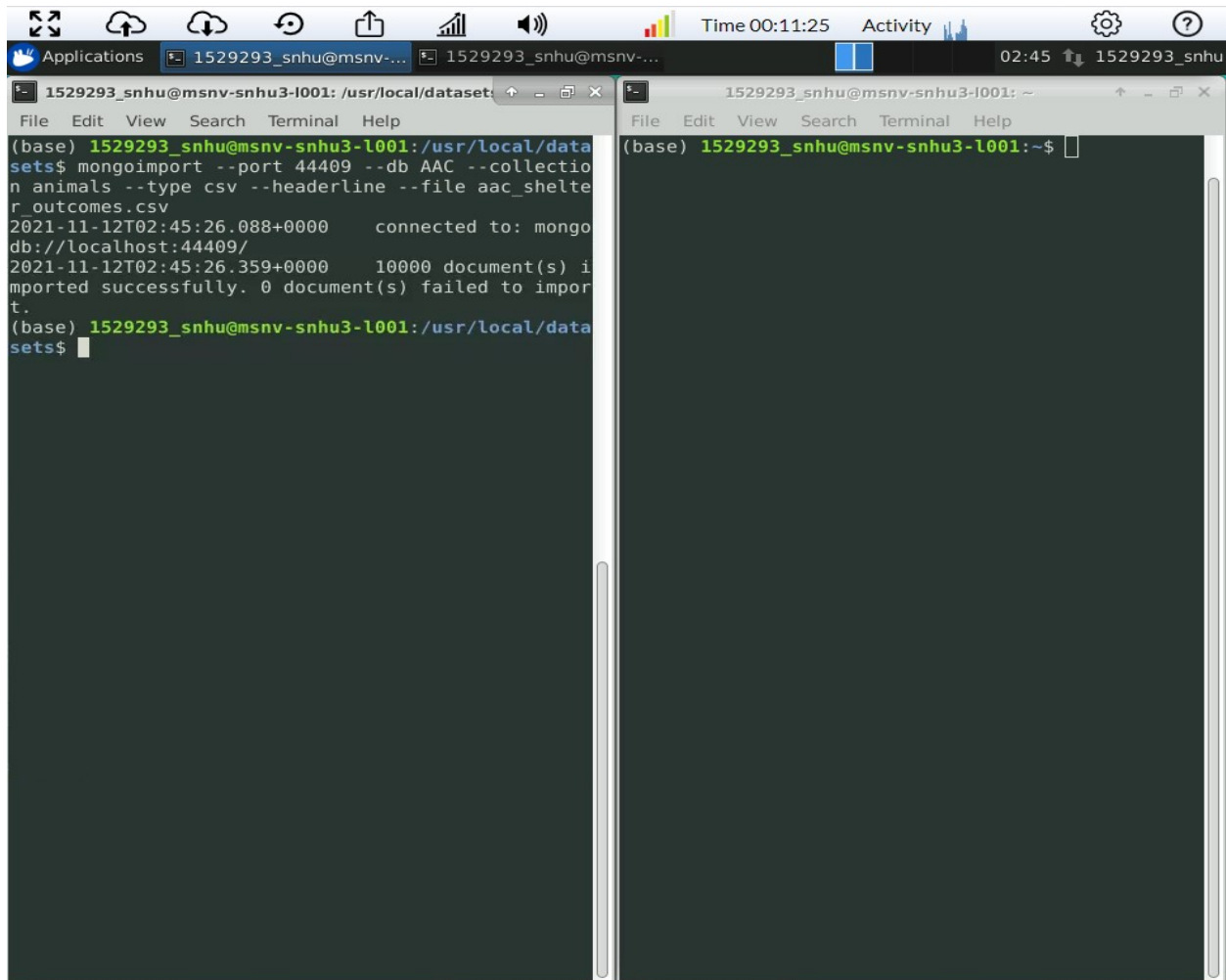
CS 340 Client/Server Development

11/13/2021

DATABASE INDEXING AND AUTHENTICATION

PART 1: IMPORTING AND INDEXING A DATA SET

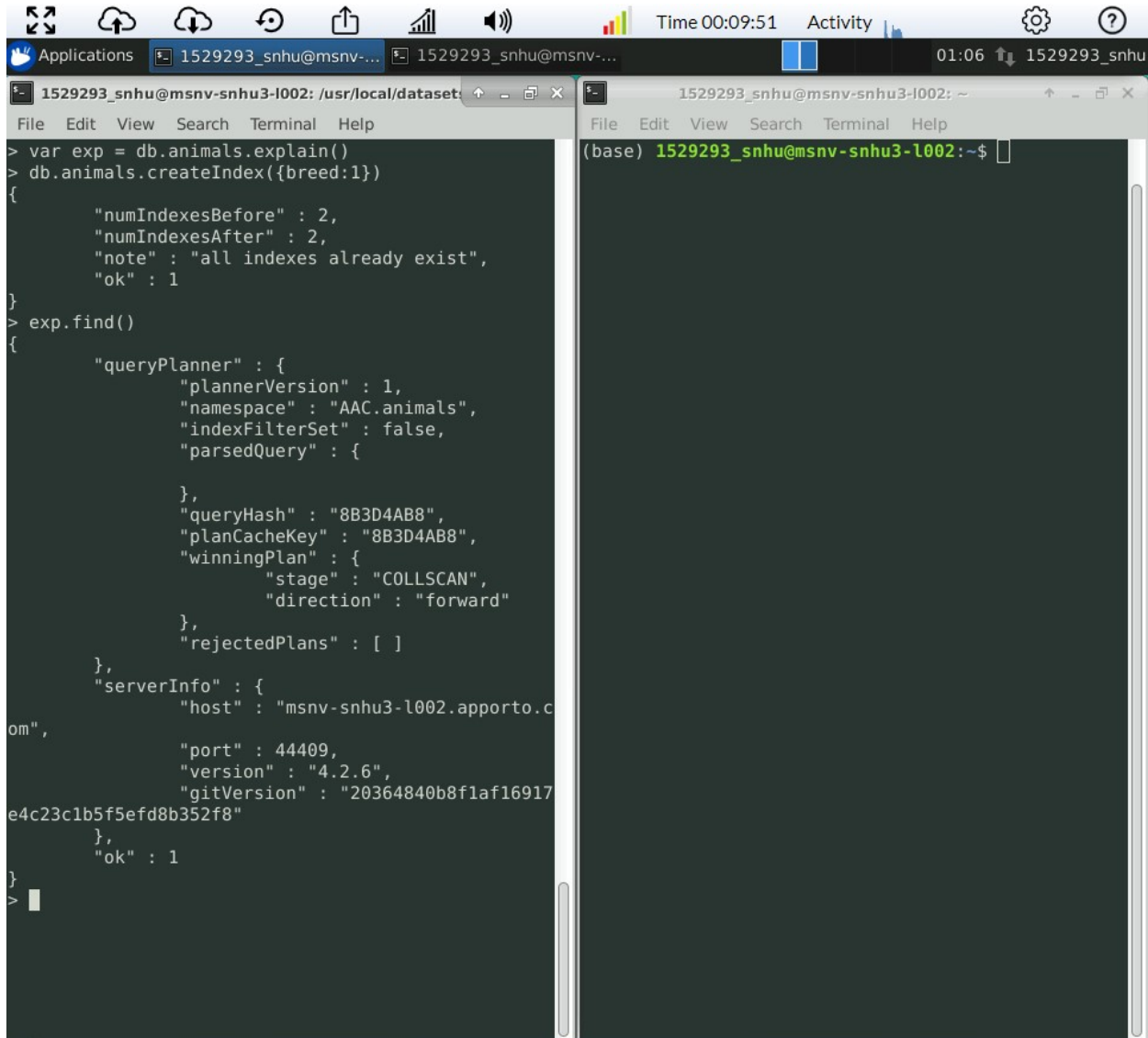
- 1.) IMPORT AAC_SHELTER_OUTCOMES.CSV FILE USING THE APPROPRIATE MONGODB IMPORT TOOL.



The screenshot shows a terminal window with the following content:

```
(base) 1529293_snhu@msnv-snhu3-l001:/usr/local/dataset$ mongoimport --port 44409 --db AAC --collection animals --type csv --headerline --file aac_shelter_outcomes.csv
2021-11-12T02:45:26.088+0000    connected to: mongo
db://localhost:44409/
2021-11-12T02:45:26.359+0000    10000 document(s) imported successfully. 0 document(s) failed to import.
(base) 1529293_snhu@msnv-snhu3-l001:/usr/local/dataset$
```

2.) CREATE A SIMPLE INDEX ON THE KEY “BREED” AND SHOW AN EXAMPLE QUERY THAT WILL USE THIS INDEX AND VERIFY WITH THE EXPLAIN FUNCTION.



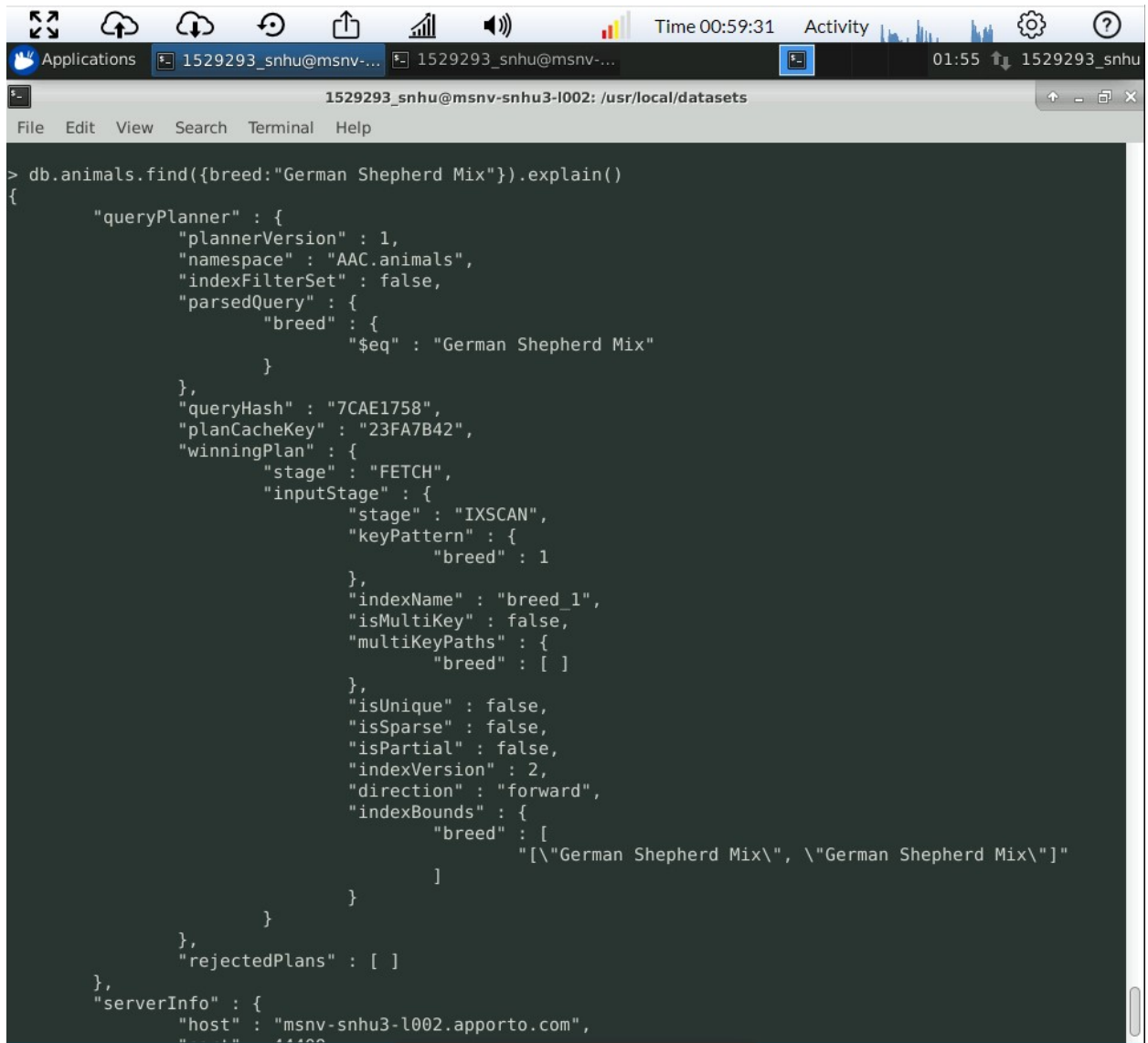
The screenshot shows a terminal window with the following commands and output:

```
> var exp = db.animals.explain()
> db.animals.createIndex({breed:1})
{
  "numIndexesBefore" : 2,
  "numIndexesAfter" : 2,
  "note" : "all indexes already exist",
  "ok" : 1
}
> exp.find()
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "AAC.animals",
    "indexFilterSet" : false,
    "parsedQuery" : {
    },
    "queryHash" : "8B3D4AB8",
    "planCacheKey" : "8B3D4AB8",
    "winningPlan" : {
      "stage" : "COLLSCAN",
      "direction" : "forward"
    },
    "rejectedPlans" : [ ]
  },
  "serverInfo" : {
    "host" : "msnv-snhu3-l002.apporto.c
om",
    "port" : 44409,
    "version" : "4.2.6",
    "gitVersion" : "20364840b8f1af16917
e4c23c1b5f5efd8b352f8"
  },
  "ok" : 1
}
```

Applications 1529293_snhu@msnv-... 1529293_snhu@msnv-... Time 00:24:10 Activity 01:20 1529293_snhu

1529293_snhu@msnv-snhu3-l002: /usr/local/dataset
File Edit View Search Terminal Help
> exp.find({breed:1})
{
 "queryPlanner" : {
 "plannerVersion" : 1,
 "namespace" : "AAC.animals",
 "indexFilterSet" : false,
 "parsedQuery" : {
 "breed" : {
 "\$eq" : 1
 }
 },
 "queryHash" : "7CAE1758",
 "planCacheKey" : "23FA7B42",
 "winningPlan" : {
 "stage" : "FETCH",
 "inputStage" : {
 "stage" : "IXSCAN",
 "keyPattern" : {
 "breed" : 1
 },
 "indexName" : "breed_1",
 "isMultiKey" : false,
 "multiKeyPaths" : {
 "breed" : []
 },
 "isUnique" : false,
 "isSparse" : false,
 "isPartial" : false,
 "indexVersion" : 2,
 "direction" : "forward",
 "indexBounds" : {
 "breed" : [[1.0, 1.0]]
 }
 }
 }
 }
}

1529293_snhu@msnv-snhu3-l002: ~
File Edit View Search Terminal Help
(base) 1529293_snhu@msnv-snhu3-l002:~\$



The screenshot shows a terminal window with a dark background. The title bar at the top indicates the user is '1529293_snhu@msnv-...' and the current directory is '/usr/local/datasets'. The terminal displays the command `db.animals.find({breed:'German Shepherd Mix'}).explain()` and its output, which is a detailed JSON query plan. The plan includes fields like 'plannerVersion', 'namespace', 'indexFilterSet', 'parsedQuery', 'queryHash', 'planCacheKey', 'winningPlan', 'rejectedPlans', and 'serverInfo'. The 'winningPlan' section details the execution stages: 'FETCH' as the main stage and 'IXSCAN' as the input stage. The 'IXSCAN' stage specifies the index 'breed_1' and its bounds, which are the exact strings 'German Shepherd Mix'.

```
> db.animals.find({breed:'German Shepherd Mix'}).explain()
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "AAC.animals",
    "indexFilterSet" : false,
    "parsedQuery" : {
      "breed" : {
        "$eq" : "German Shepherd Mix"
      }
    },
    "queryHash" : "7CAE1758",
    "planCacheKey" : "23FA7B42",
    "winningPlan" : {
      "stage" : "FETCH",
      "inputStage" : {
        "stage" : "IXSCAN",
        "keyPattern" : {
          "breed" : 1
        },
        "indexName" : "breed_1",
        "isMultiKey" : false,
        "multiKeyPaths" : {
          "breed" : [ ]
        },
        "isUnique" : false,
        "isSparse" : false,
        "isPartial" : false,
        "indexVersion" : 2,
        "direction" : "forward",
        "indexBounds" : {
          "breed" : [
            ["German Shepherd Mix", "German Shepherd Mix"]
          ]
        }
      }
    },
    "rejectedPlans" : [ ]
  },
  "serverInfo" : {
    "host" : "msnv-snhu3-l002.apporto.com",
    "port" : 27020
  }
}
```

```
Applications 1529293_snhu@msnv-... 1529293_snhu@msnv-... 02:04 1529293_snhu
1529293_snhu@msnv-snhu3-l002: /usr/local/datasets
File Edit View Search Terminal Help
> db.animals.find({breed:"German Shepherd Mix"}).hint("breed_1").explain()
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "AAC.animals",
    "indexFilterSet" : false,
    "parsedQuery" : {
      "breed" : {
        "$eq" : "German Shepherd Mix"
      }
    },
    "queryHash" : "7CAE1758",
    "planCacheKey" : "23FA7B42",
    "winningPlan" : {
      "stage" : "FETCH",
      "inputStage" : {
        "stage" : "IXSCAN",
        "keyPattern" : {
          "breed" : 1
        },
        "indexName" : "breed_1",
        "isMultiKey" : false,
        "multiKeyPaths" : {
          "breed" : [ ]
        },
        "isUnique" : false,
        "isSparse" : false,
        "isPartial" : false,
        "indexVersion" : 2,
        "direction" : "forward",
        "indexBounds" : {
          "breed" : [
            ["German Shepherd Mix\\", "German Shepherd Mix\\"]
          ]
        }
      },
      "rejectedPlans" : [ ]
    },
    "serverInfo" : {
      "host" : "msnv-snhu3-l002.apporto.com",

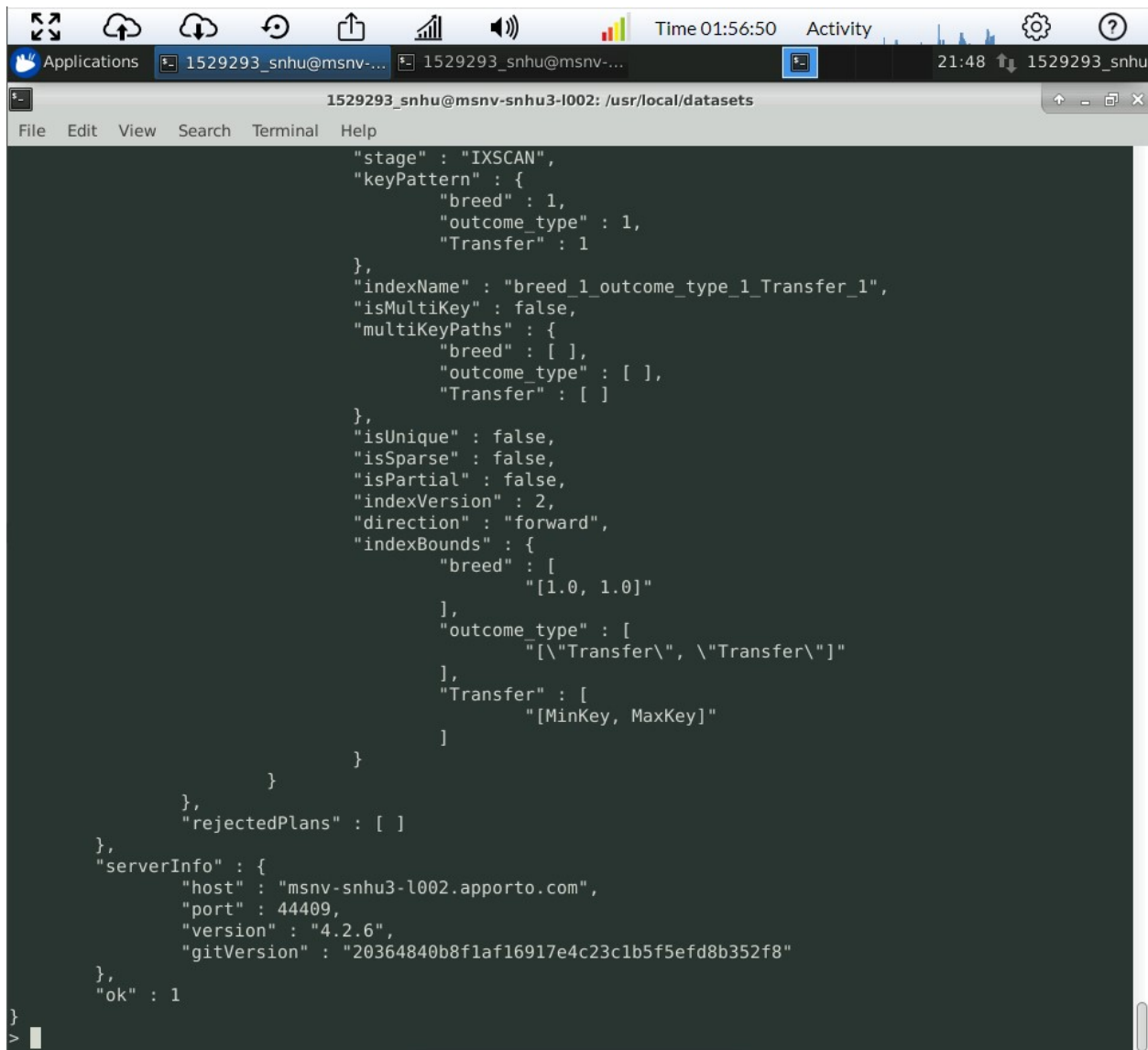
```

3.) CREATE A COMPOUND INDEX, SHOW AN EXAMPLE QUERY THAT WILL USE THIS COMPOUND INDEX AND CONFIRM WITH THE EXPLAIN FUNCTION.

```
Applications 1529293_snhu@msnv-... 1529293_snhu@msnv-... 21:41 1529293_snhu
1529293_snhu@msnv-snhu3-l002: /usr/local/dataset
File Edit View Search Terminal Help
> db.animals.createIndex({breed:1,outcome_type:1,transfer:1})
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
.379265873016
"9993","2 years","A761792","Dog","Bichon Frise/Miniature Poodle","White",2015-05-09,2017-11-10 14:36:00,"2017-11-10T14:36:00","Voler","Partner","Transfer","Intact Male",30.5819767764418,-97.3349580686668,130.944047619048
"9994","2 years","A713640","Cat","Domestic Shorthair Mix","Blue/White",2013-10-10,2015-10-21 13:44:0
```

```
Time 01:55:57 Activity
Applications 1529293_snhu@msnv-... 1529293_snhu@msnv-... 21:47 1529293_snhu
1529293_snhu@msnv-snhu3-l002: /usr/local/datasets
File Edit View Search Terminal Help

> db.animals.find({"outcome_type":"Transfer", "breed":1}).explain()
{
  "queryPlanner" : {
    "plannerVersion" : 1,
    "namespace" : "AAC.animals",
    "indexFilterSet" : false,
    "parsedQuery" : {
      "$and" : [
        {
          "breed" : {
            "$eq" : 1
          }
        },
        {
          "outcome_type" : {
            "$eq" : "Transfer"
          }
        }
      ]
    },
    "queryHash" : "A0C645BE",
    "planCacheKey" : "EE7A2759",
    "winningPlan" : {
      "stage" : "FETCH",
      "inputStage" : {
        "stage" : "IXSCAN",
        "keyPattern" : {
          "breed" : 1,
          "outcome_type" : 1,
          "Transfer" : 1
        },
        "indexName" : "breed_1_outcome_type_1_Transfer_1",
        "isMultiKey" : false,
        "multiKeyPaths" : {
          "breed" : [ ],
          "outcome_type" : [ ],
          "Transfer" : [ ]
        },
        "isUnique" : false,
        "isSparse" : false,
        "isPartial" : false,
        "indexVersion" : 2
      }
    }
  }
}
```



```

"stage" : "IXSCAN",
"keyPattern" : {
  "breed" : 1,
  "outcome_type" : 1,
  "Transfer" : 1
},
"indexName" : "breed_1_outcome_type_1_Transfer_1",
"isMultiKey" : false,
"multiKeyPaths" : {
  "breed" : [ ],
  "outcome_type" : [ ],
  "Transfer" : [ ]
},
"isUnique" : false,
"isSparse" : false,
"isPartial" : false,
"indexVersion" : 2,
"direction" : "forward",
"indexBounds" : {
  "breed" : [
    "[1.0, 1.0]"
  ],
  "outcome_type" : [
    "[\"Transfer\", \"Transfer\"]"
  ],
  "Transfer" : [
    "[MinKey, MaxKey]"
  ]
}
},
"rejectedPlans" : [ ]
},
"serverInfo" : {
  "host" : "msnv-snhu3-l002.apporto.com",
  "port" : 44409,
  "version" : "4.2.6",
  "gitVersion" : "20364840b8f1af16917e4c23c1b5f5efd8b352f8"
},
"ok" : 1
}
>

```

PART 2: USER AUTHENTICATION

1.) CREATE AN ADMINISTRATOR ACCOUNT IN THE MONGO SHELL

- a. I had previously seen the announcement and created an admin user before this week. I did so by running the mongod server without authentication by `/usr/local/bin/mongod_ctl start-noauth` . then entering the mongo shell with

mongo. Once in the mongo command prompt I used the command “use admin”, then `db.createUser({ user:”admin”, pwd:passwordPrompt(), roles: [{role”userAdminAnyDatabase”,db:”admin”},”readWriteAnyDatabase”]})`

2.) ENABLE USER AUTHENTICATION FOR THE DATABASE

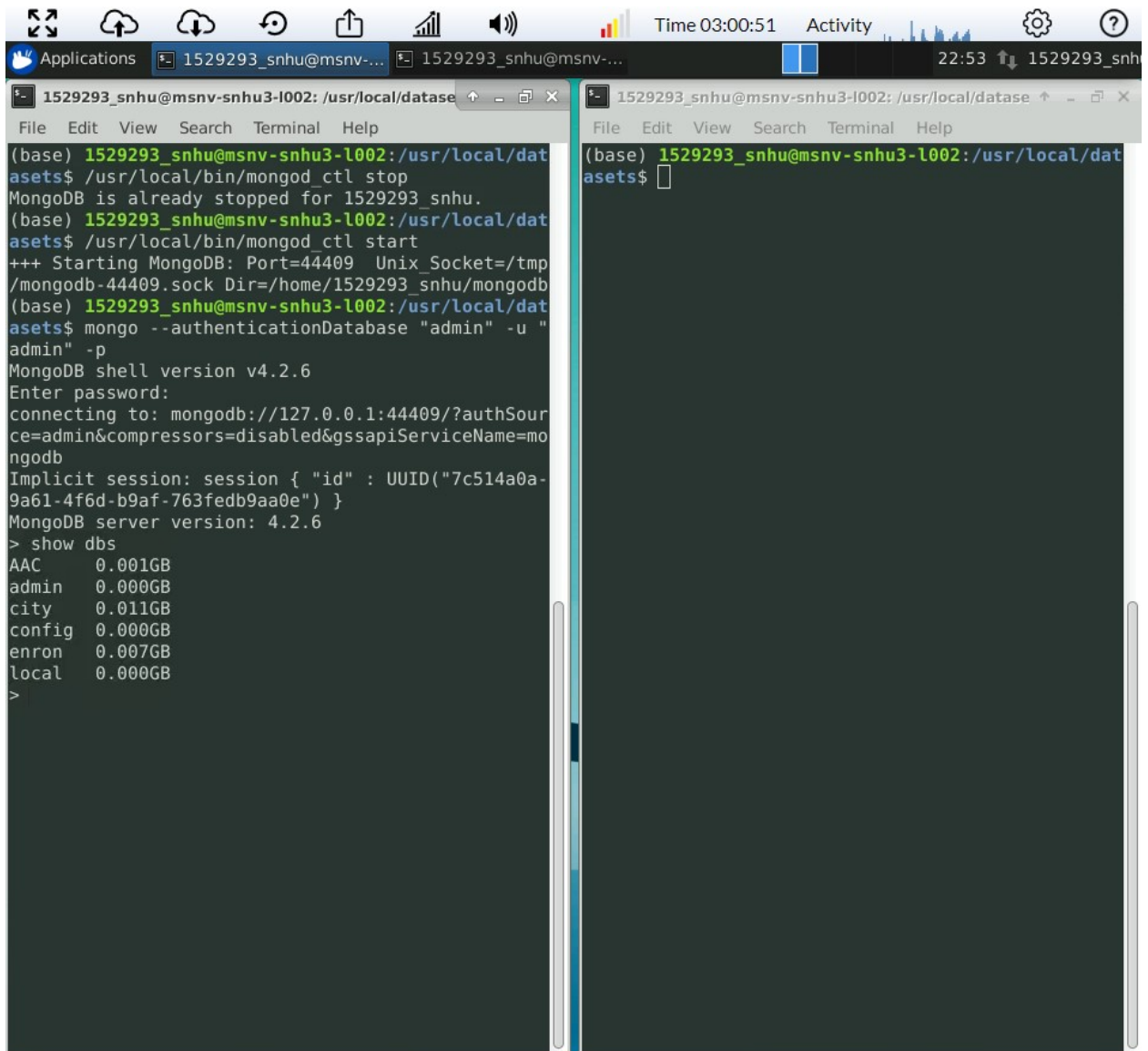
- a. Once I created an admin account I exited the shell with the “exit” command. I then stopped mongo with `/usr/local/bin/mongod_ctl stop`. I restarted the mongo shell with authentication using `/usr/local/bin/mongod_ctl start`. I then ran mongo with my “admin” account by using the command `mongo –authenticationDatabase “admin” -u “admin” -p`, entered my password and went on to create a user account.

3.) CREATE A NEW USER ACCOUNT CALLED “AACUSER”

- a. I entered the command “use AAC” then
`db.createUser({user:”aacuser”,pwd:passwordPrompt(),roles[{role:”readWrite”,db:”AAC”}]})`

4.) TAKE A SCREENSHOT OF YOUR LOGIN PROCESS

Here are the screenshots I have of the login, I had forgot the password I originally created for the aacuser so I used the documentation to figure out how to change the password for a user by logging into the admin user account and using the command `db.changeUserPassword(“aacuser”,passwordPrompt())` and entering a new password.



The image shows a dual-terminal window with two panes. The top pane contains the following text:

```
(base) 1529293_snhu@msnv-snhu3-l002:/usr/local/dat
asets$ /usr/local/bin/mongod_ctl stop
MongoDB is already stopped for 1529293_snhu.
(base) 1529293_snhu@msnv-snhu3-l002:/usr/local/dat
asets$ /usr/local/bin/mongod_ctl start
+++ Starting MongoDB: Port=44409 Unix_Socket=/tmp
/mongodb-44409.sock Dir=/home/1529293_snhu/mongodb
(base) 1529293_snhu@msnv-snhu3-l002:/usr/local/dat
asets$ mongo --authenticationDatabase "admin" -u "
admin" -p
MongoDB shell version v4.2.6
Enter password:
connecting to: mongodb://127.0.0.1:44409/?authSour
ce=admin&compressors=disabled&gssapiServiceName=mo
ngodb
Implicit session: session { "id" : UUID("7c514a0a-
9a61-4f6d-b9af-763fedb9aa0e") }
MongoDB server version: 4.2.6
> show dbs
AAC      0.001GB
admin    0.000GB
city     0.011GB
config   0.000GB
enron    0.007GB
local    0.000GB
>
```

The bottom pane is currently empty, showing only the prompt:

```
(base) 1529293_snhu@msnv-snhu3-l002:/usr/local/dat
asets$
```

