**Hadoop Queues, Queue Policies and Queue Issues in Apache Ranger**

Apache Ranger delivers a comprehensive approach to security for a Hadoop cluster.

It provides central security policy administration across the core enterprise security

Requirements of authorization, auditing and data protection.

**Prerequisites**

The following components are required to process apache ranger installation.

1. JDK – For Running RangerAdmin, RangerKMS

2. RDBMS – Storing policies, Ranger Users, and Groups (MySQL/Oracle/Postgres/SQL Server)

3. Solr – Store Audit Logs (Apache Solr – 5.2.1+)

4. HDFS – Store Audit Logs

5. Kerberos – Enabling Kerberos will ensure that the requests get authenticated.

**Kerberos Authentication (krb5) :**

Keytab file creation is done for new Hadoop cluster

Keytab file contains assigned user to particular cluster or environment

Keytab file used to authenticate to various remote systems using Kerberos without entering a password

We can restrict access permissions on keytab file using unix permissions commands

Keytab file provides authentication to the users to the Hadoop clusters

Keytab file contains – principal name and password

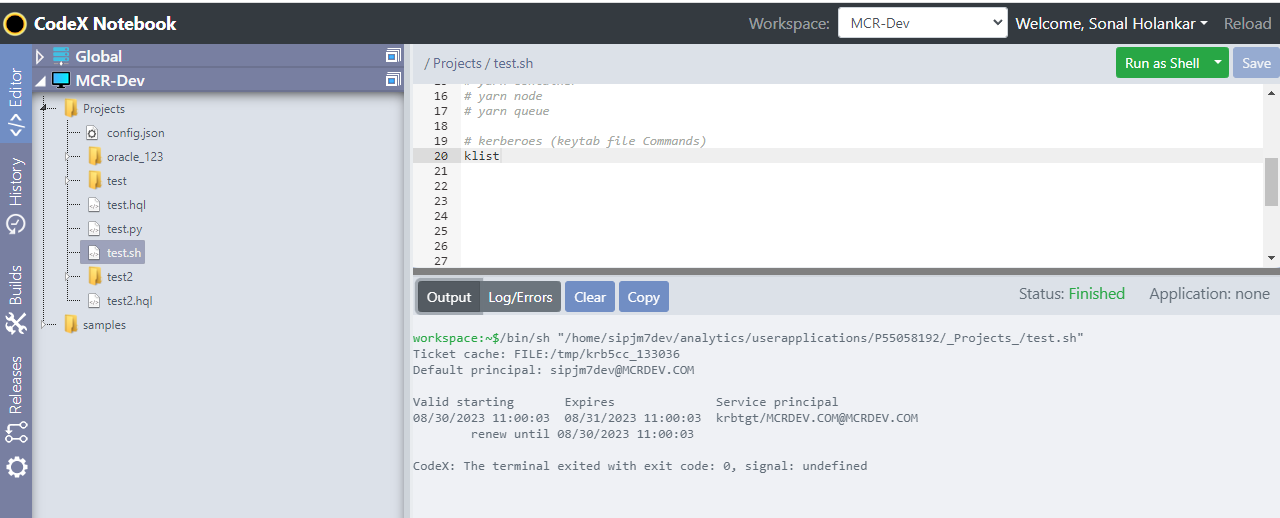
Keytab file can be same for same Hadoop cluster

Keytab file can be different for different Hadoop cluster which are Kerberos authenticated

User ---- Hadoop cluster ---- Kerberos authentication ----- keytab file creation

Klist command is used to see Keytab file details

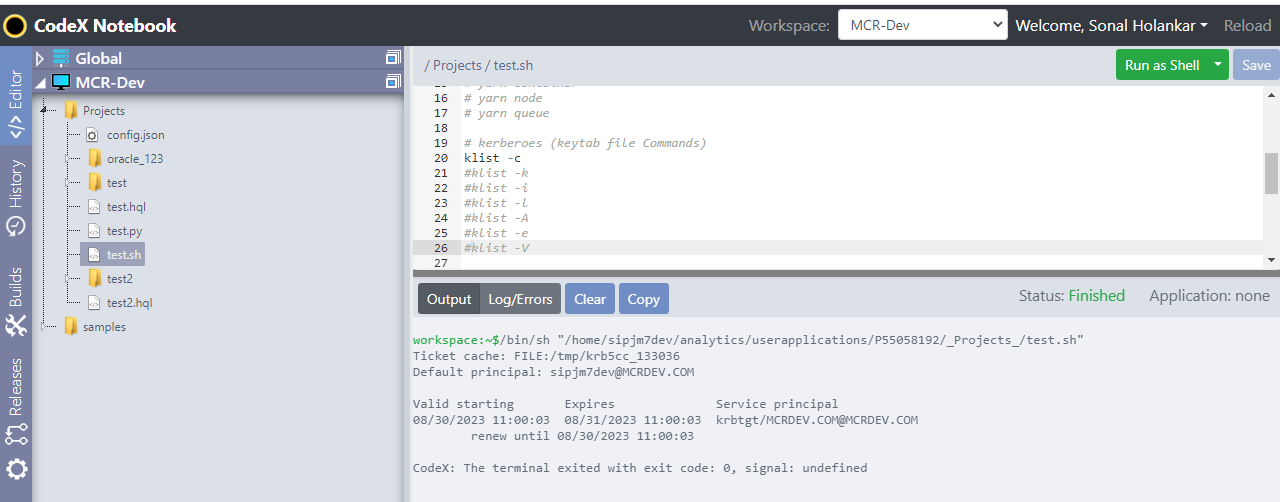
Klist command:



klist: invalid option -- 'h'

Usage: klist [-e] [-V] [[-c] [-l] [-A] [-d] [-f] [-s] [-a [-n]]] [-k [-t] [-K]] [name]

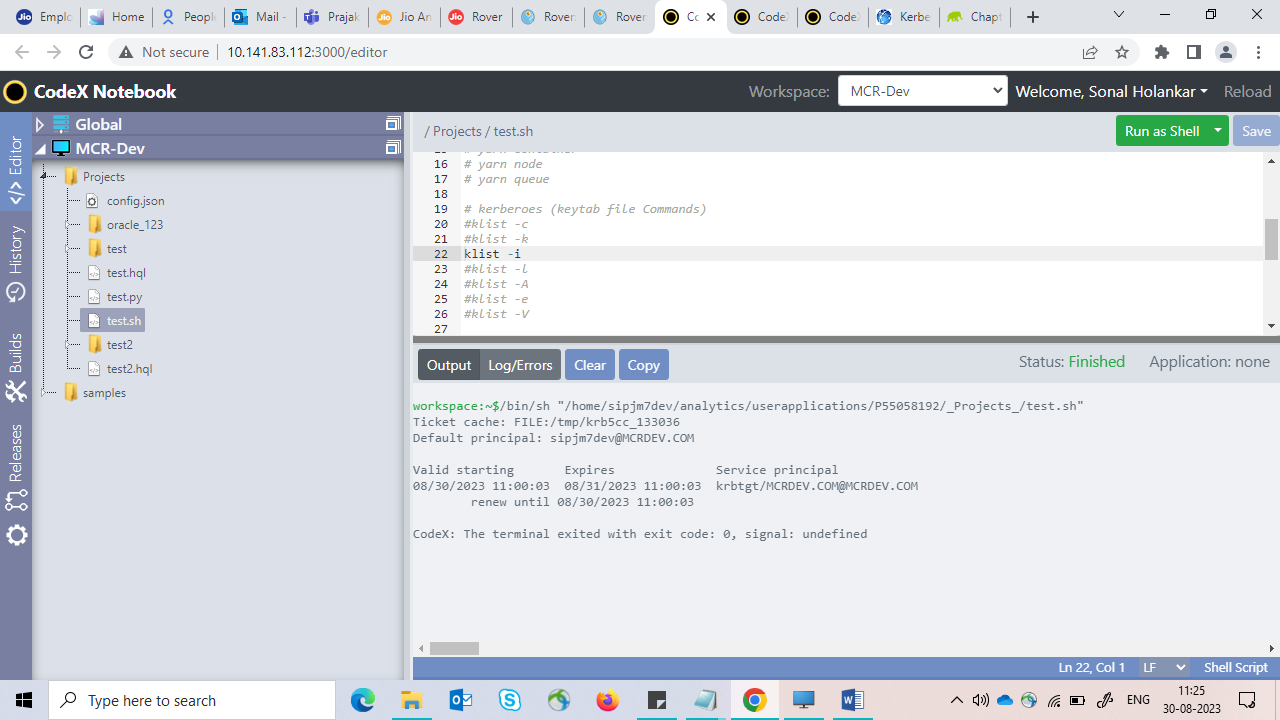
-c specifies credentials cache



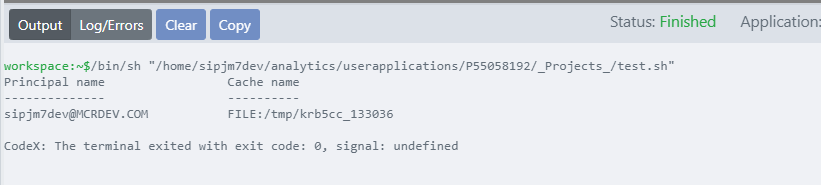
-k specifies keytab

(Default is credentials cache)

-i uses default client keytab if no name given

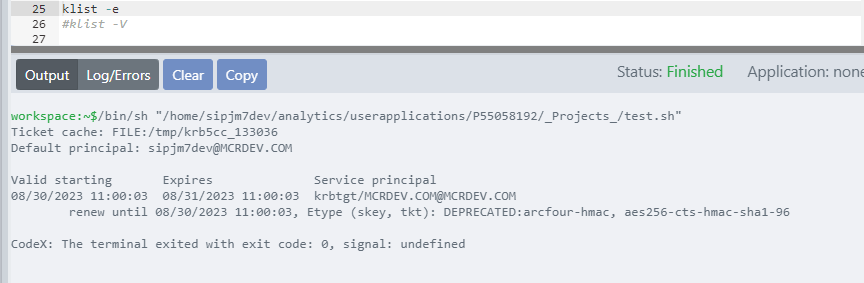


-l lists credential caches in collection



-A shows content of all credential caches

-e shows the encryption type

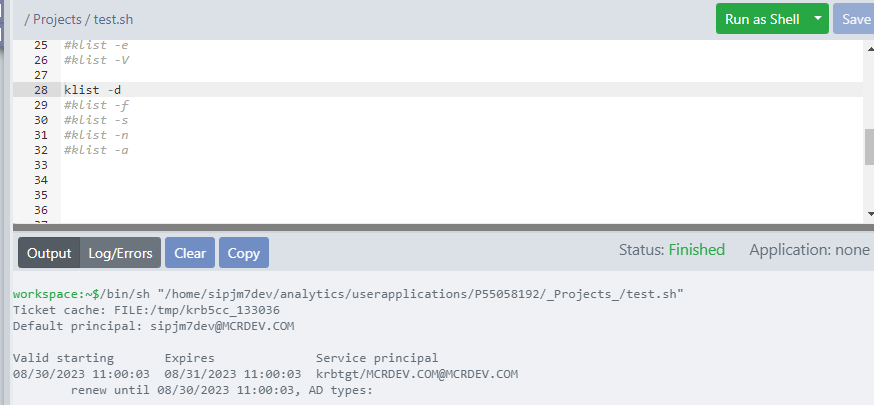


-V shows the Kerberos version and exits

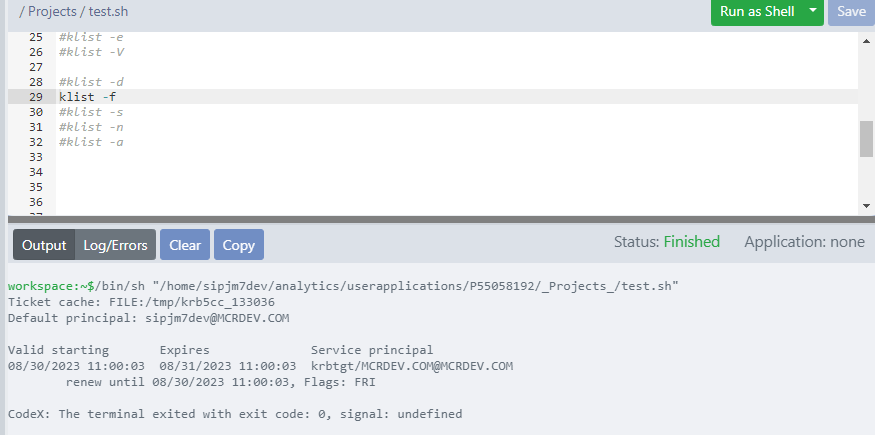


Options for credential caches:

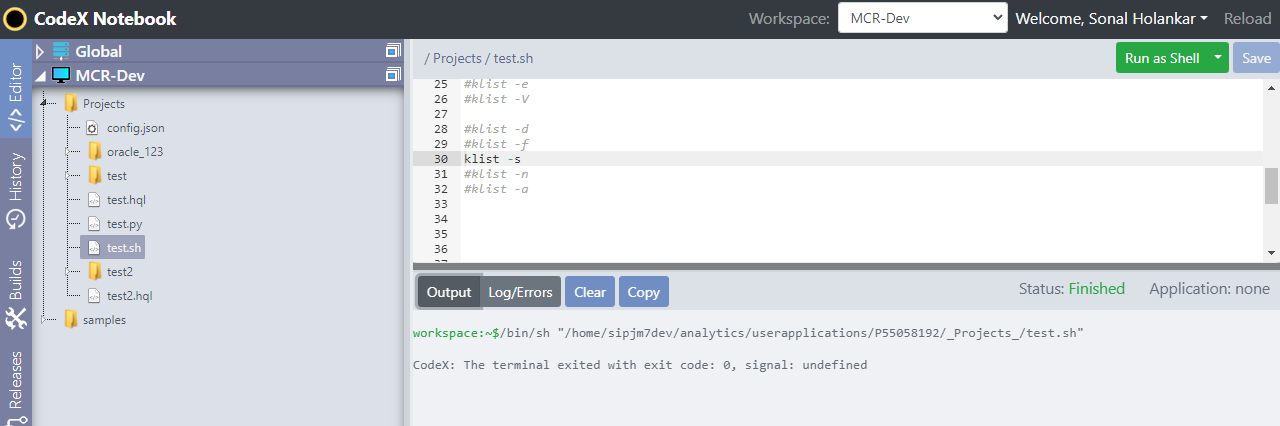
-d shows the submitted authorization data types



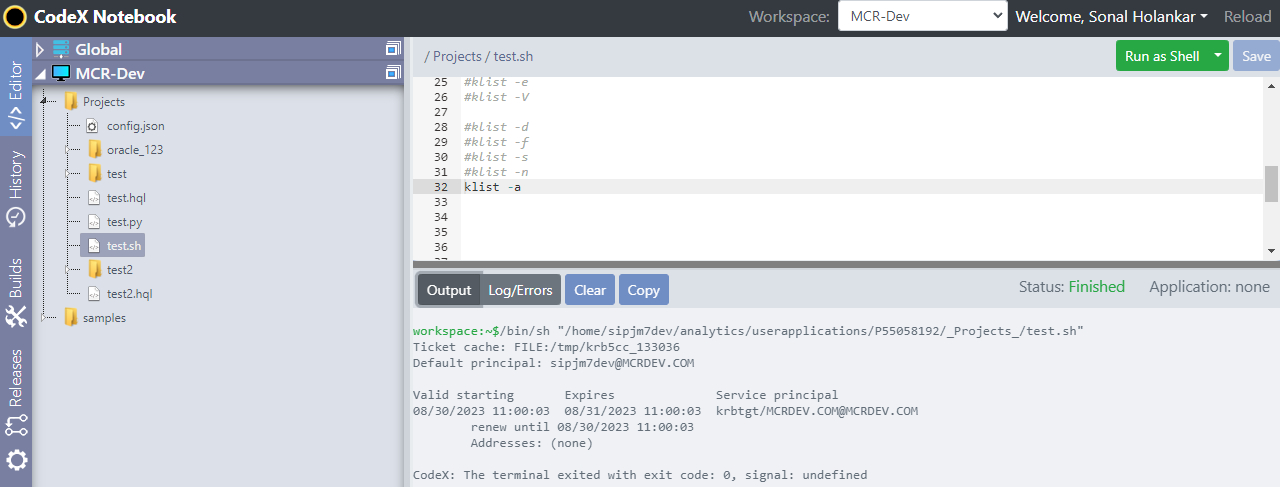
-f shows credentials flags



-s sets exit status based on valid tgt existence



-a displays the address list



-n do not reverse-resolve

Options for keytabs:

-t shows keytab entry timestamps

-K shows keytab entry keys

ktab –help ---- Kerberos key table

-l list the keytab name and entries

-a <principal\_name> [password] add an entry to the keytab

-d <principal\_name> delete an entry from the keytab

-k <keytab\_name> specify keytab name and path with FILE: prefix

-m <source\_keytab\_name> <destination\_keytab\_name>

Specify merging source keytab file name and destination keytab file name

A queue hierarchy contains three types of queues

**Root** – represents the cluster itself

**Parent** – represents organization / group or sub organization / sub-group

**Leaf**- accepts application submission

In our Hadoop clusters, only root queues are created in apache ranger. It shows the list of all queues

**All queues list:**

jm7\_qcampaign, jm7\_qkafka, jm7\_qreserve, jm7\_qaggregate, jm7\_qanalytics jm7\_qdatasource, jm7\_qingest, jm7\_qkudu, jm7\_qrfm, jm7\_qalv, default queue

**Hadoop schedulers**

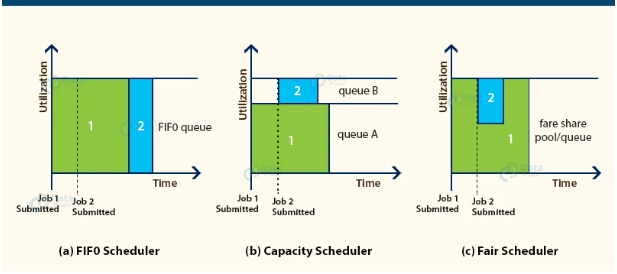
**Types of queues**

**FIFO –** First in First out service scheduling policy used. The application come first will assign to queue first.

**Fair –** Fair allows YARN applications to fairly share resources in large Hadoop clusters. It assigns resources to applications in equal amount.

**Capacity -**The Capacity Scheduler allows the sharing of the large cluster while giving capacity guarantees to each organization by allocating a fraction of cluster resources to each queue.

The CapacityScheduler provides a comprehensive set of limits to ensure that a single application/user/queue cannot use a disproportionate amount of resources in the cluster.



**Fig.no.1 Hadoop queues**

**Queue Testing for Job Scheduling in Codex Application**

The policies are created for codex users in apache ranger under yarn section. All codex users are assigned with specific queue name in MCR-Dev Ranger URL.

Following are users with queue name:

|  |  |
| --- | --- |
| **Codex User** | **Queue Name** |
| sipjm7presto2 | jm7\_qalv,jm7\_qingest |
| sipjm7rvrapp1 | jm7\_qanalytics |
| sipjm7cmp | jm7\_qcampaign |
| sipjm7dev | jm7\_qanalytics,jm7\_qreserve |
| sipjm7devcmp | jm7\_qcampaign |
| sipjm7pau1 | jm7\_qeuc |
| sipjm7pau5 | jm7\_qavr |

**MCR DEV RANGER TESTING:**

Hadoop Cluster: Mercury (MCR) Dev

The required changes made in YARN Services in the backend. Policies created in cm\_yarn in ranger, all os users are assigned to specific queue.

**Example:**

**Policy Name: MCR\_SUBMIT\_QANALYTICS**

**Queue Name: root.jm7\_qanalytics**

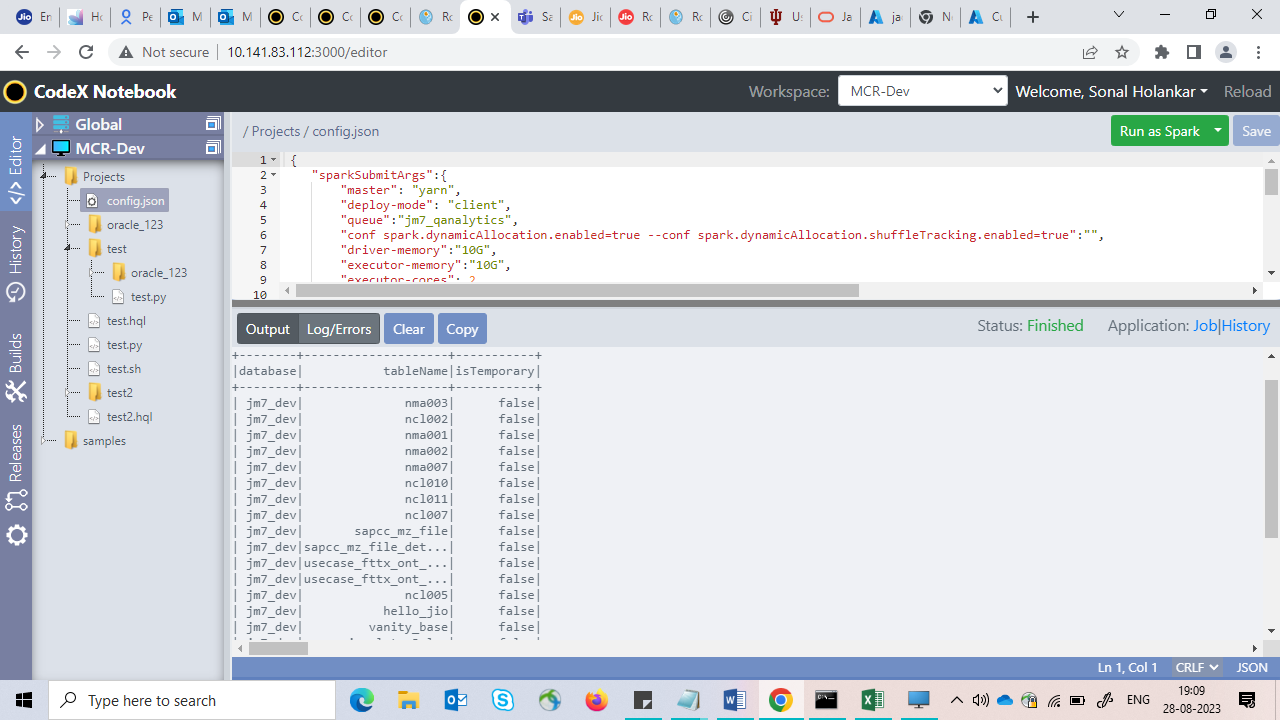
**Permissions Given: submit-app**

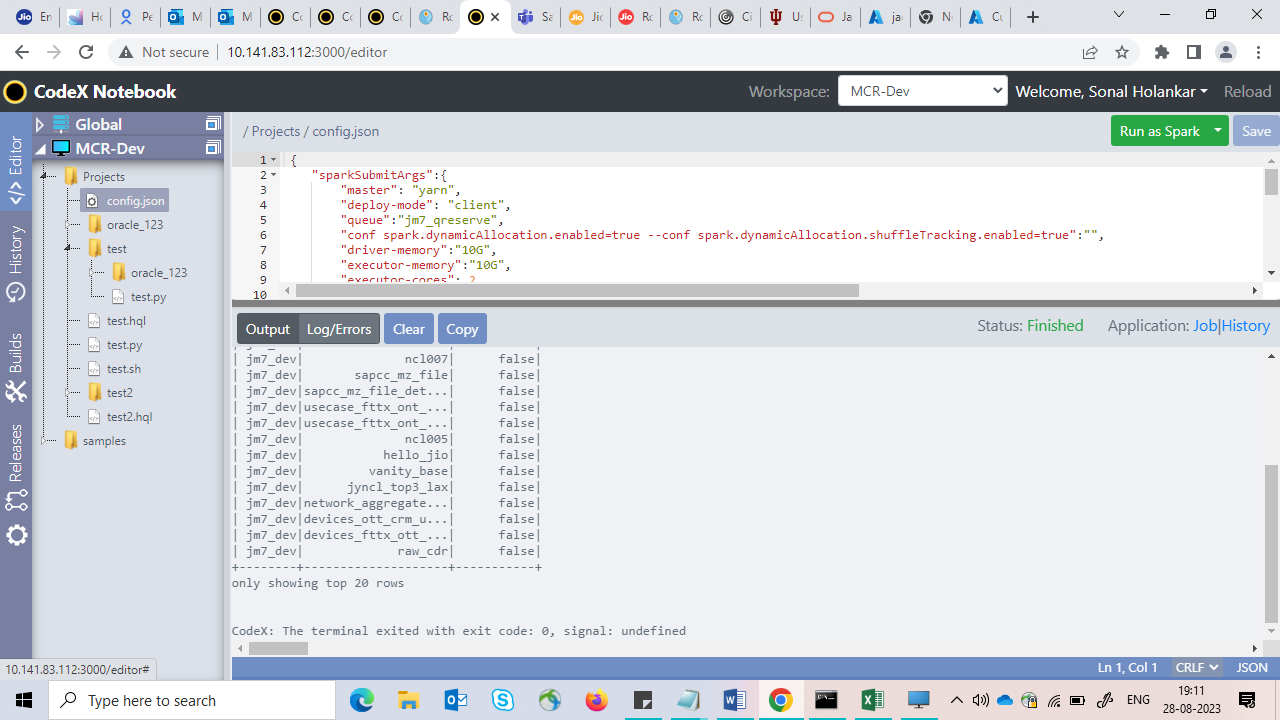
**Codex User: sipjm7dev**

**Queue Name: jm7\_qanalytics**

**Test Case 1:** jm7\_qanalytics and jm7\_qreserve is assigned to sipjm7dev user as per policy.

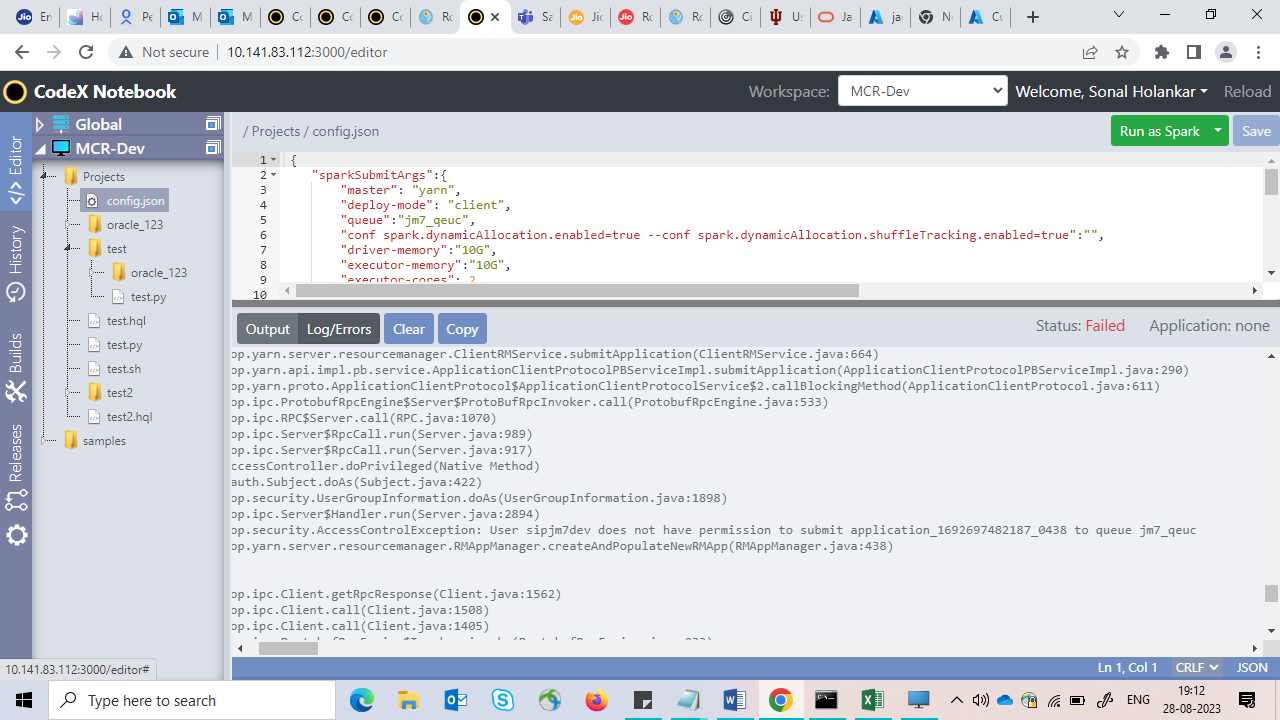
Therefore application is submitted to YARN and shows the output in below screenshots.

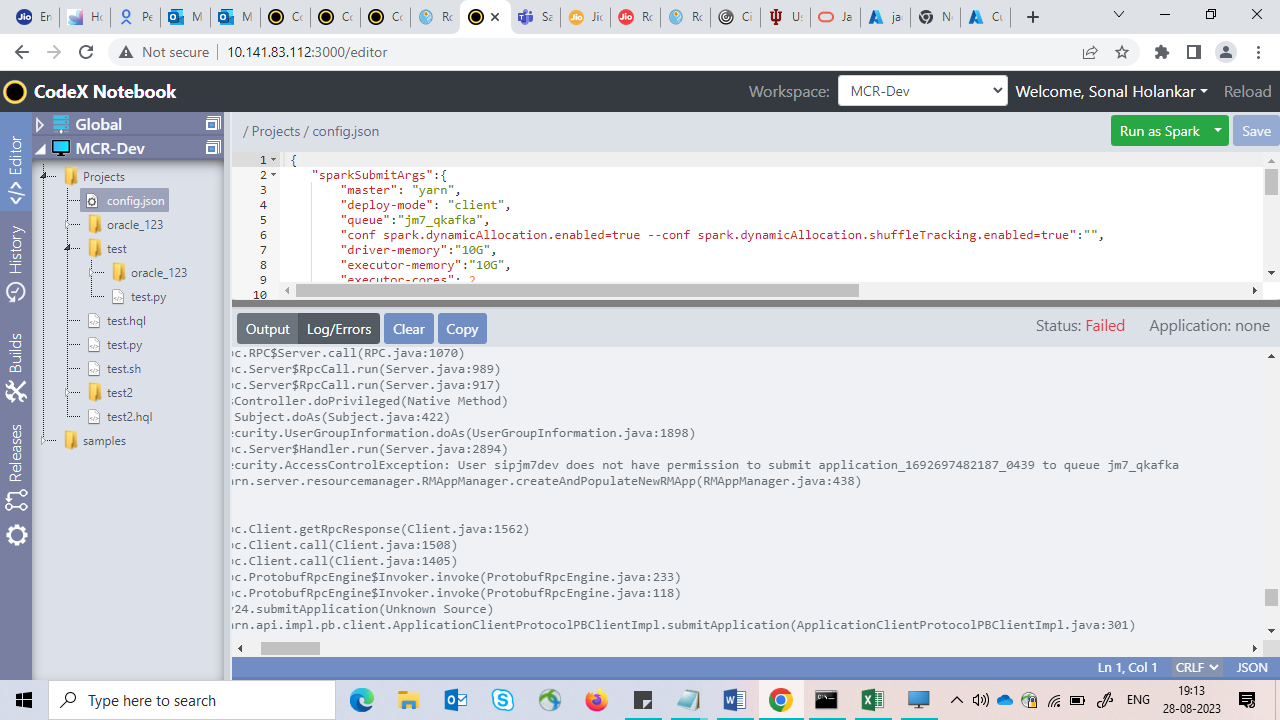


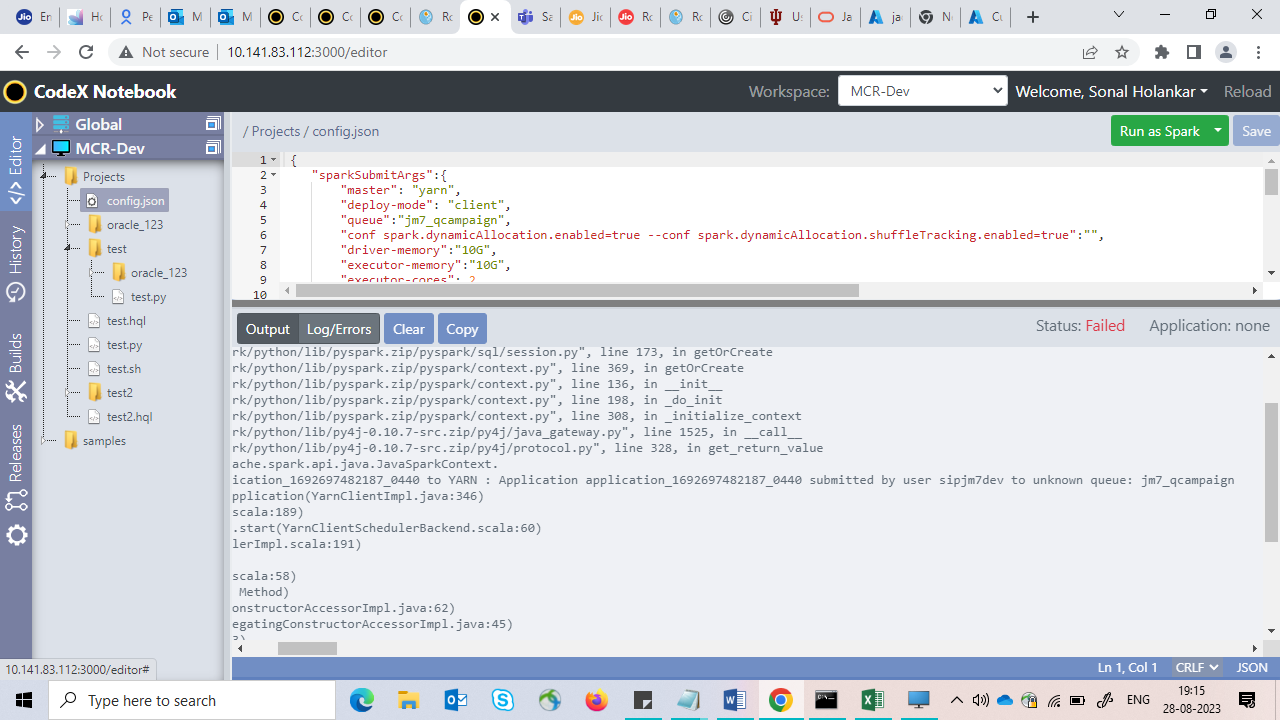


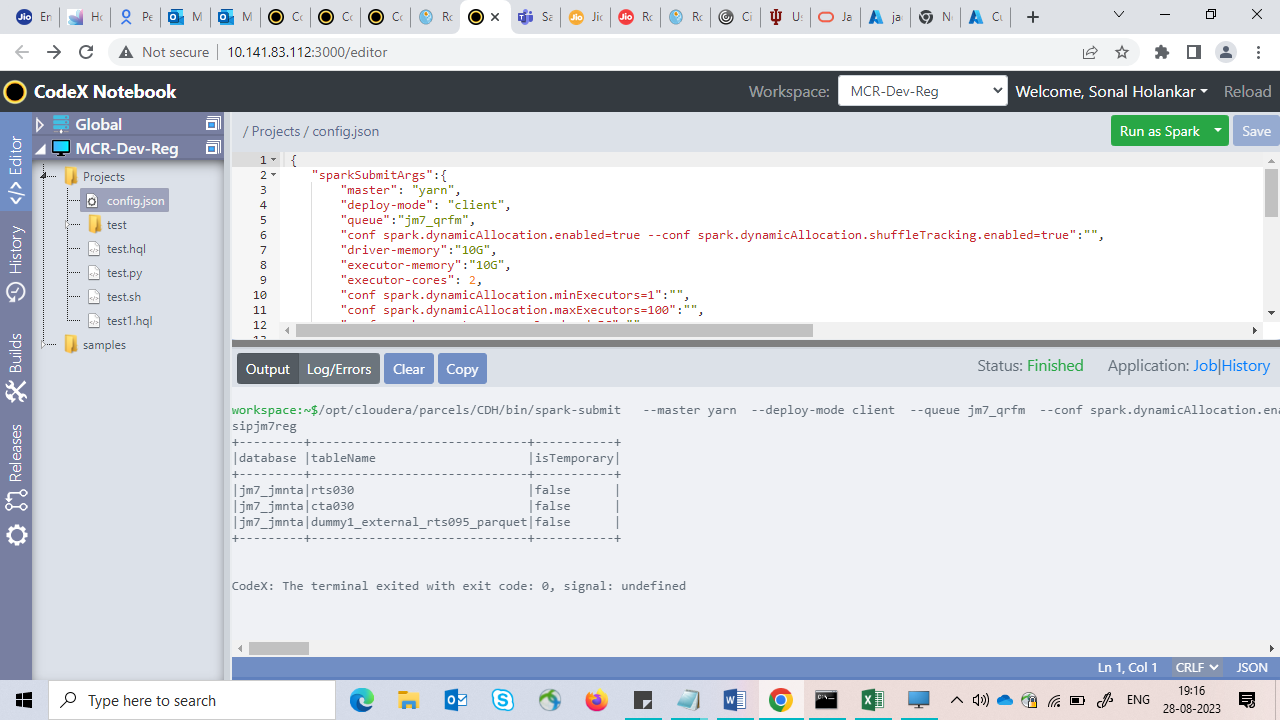
**Test Case 2:** The queues like jm7\_qeuc, jm7\_qkafka and jm7\_qcampaign are not assigned to sipjm7dev user as per policy.

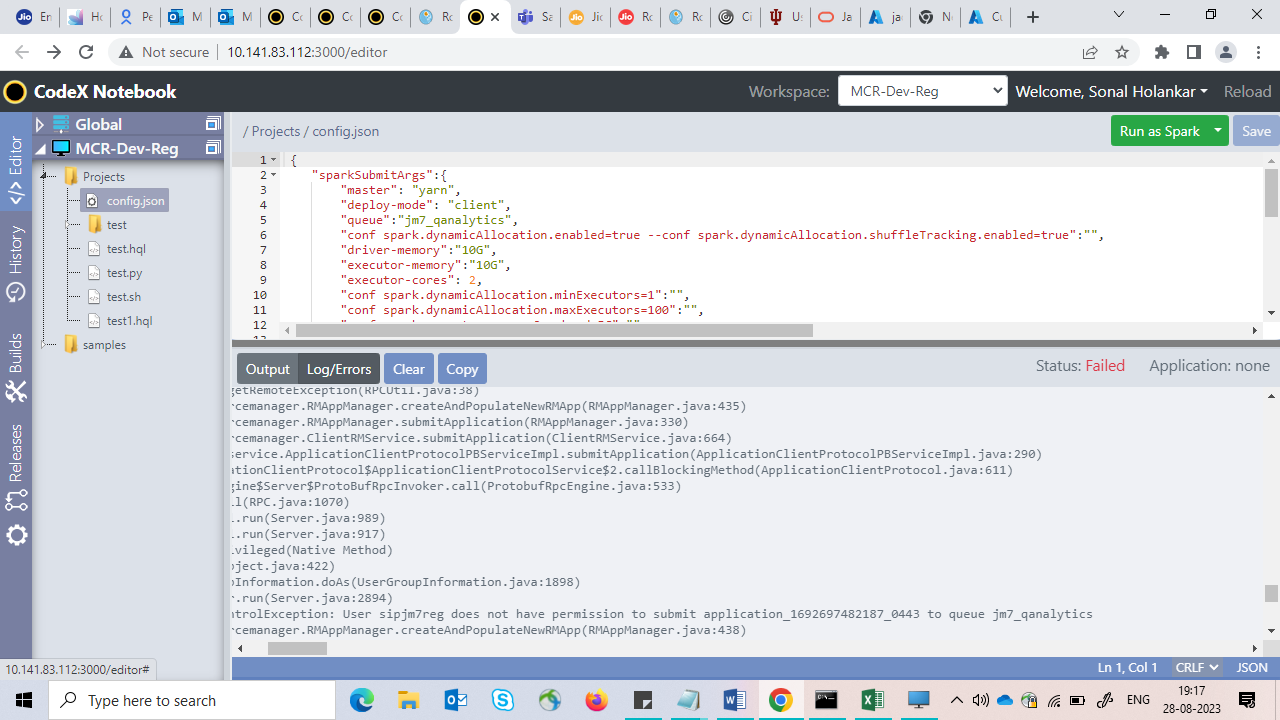
Therefore application is not submitted to YARN and it throws error as user is unknown queue to the in below screenshots.

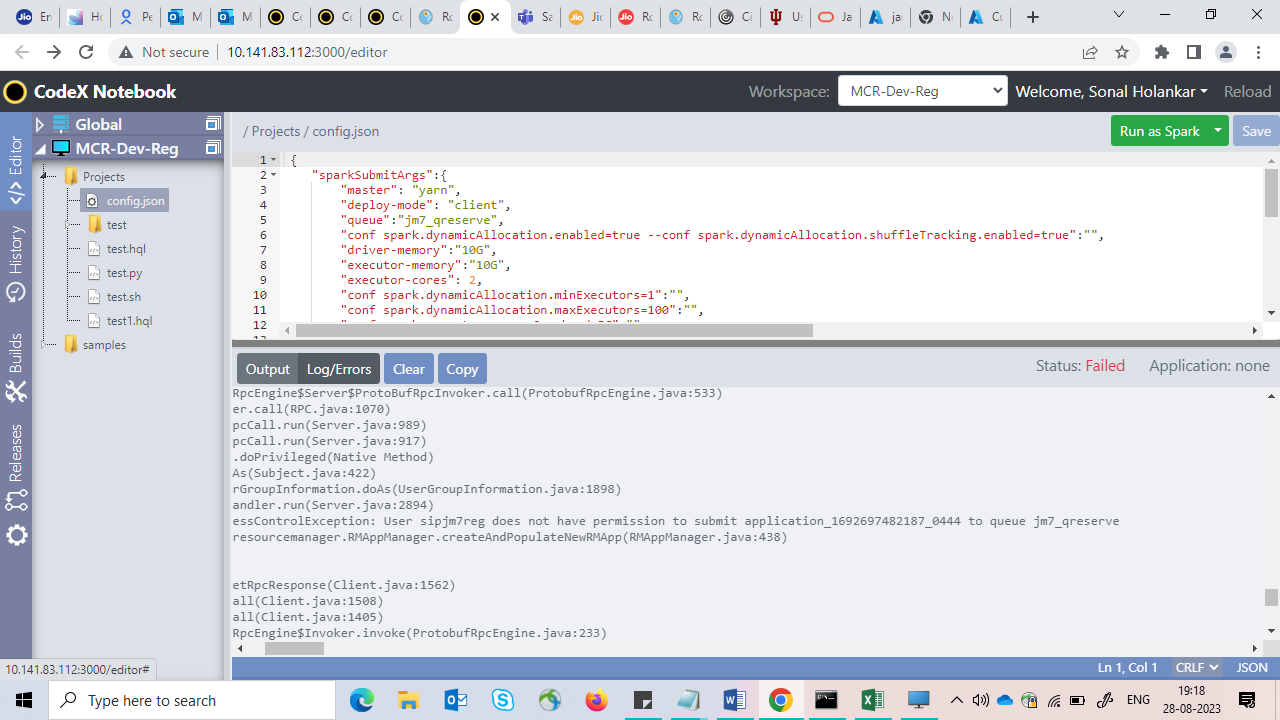


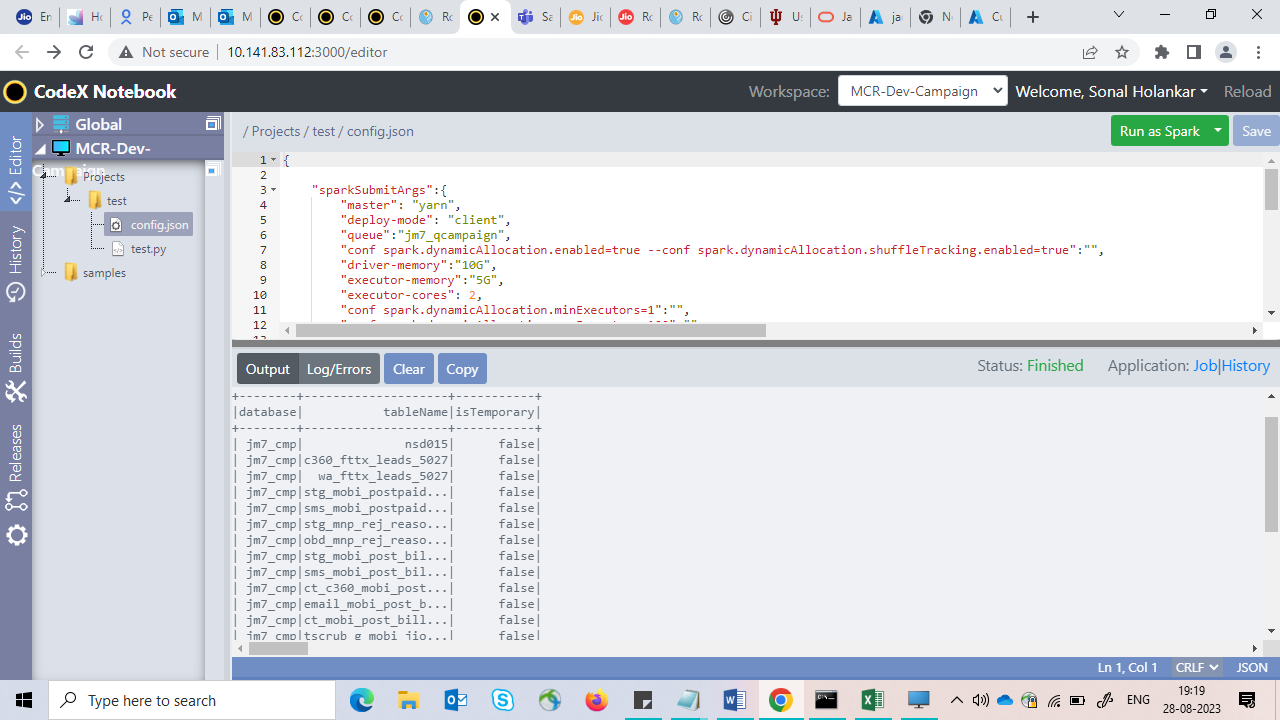








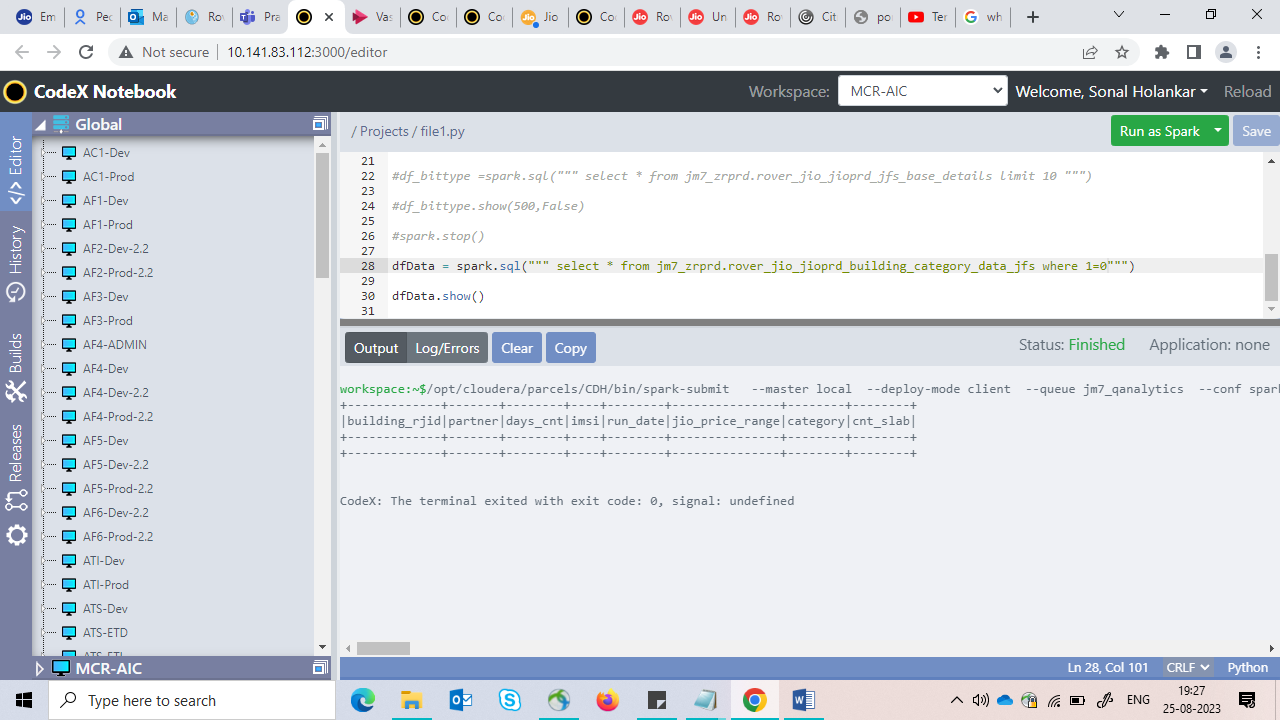




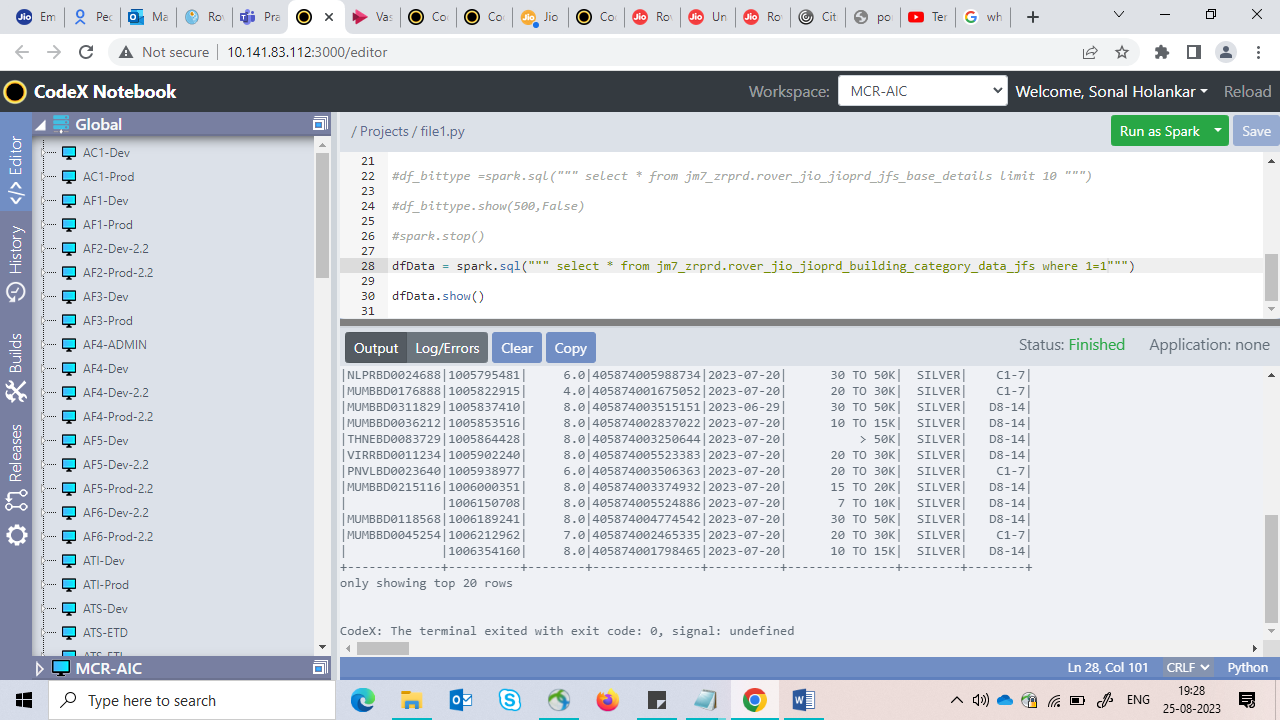
**Spark SQL queries in Codex Application:**

**Where clauses in Spark SQL queries**

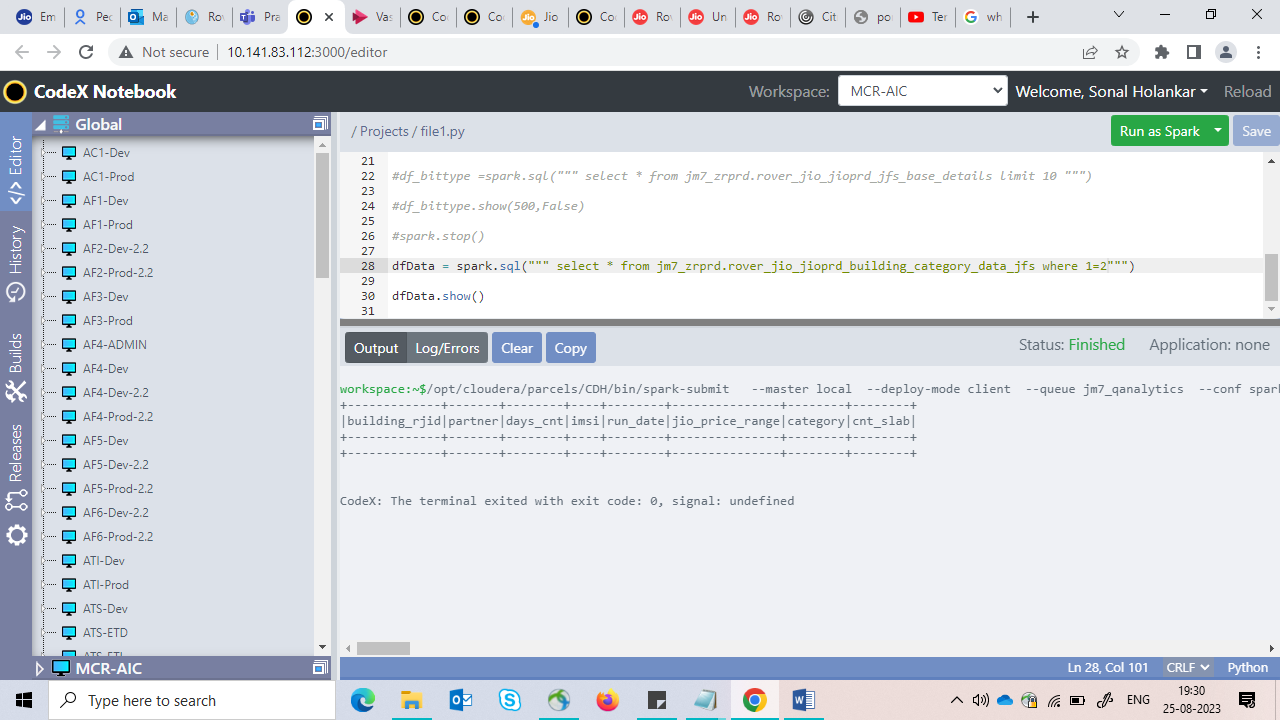
1. where 1=0 can be used to stop the query from returning any rows. It returns empty set

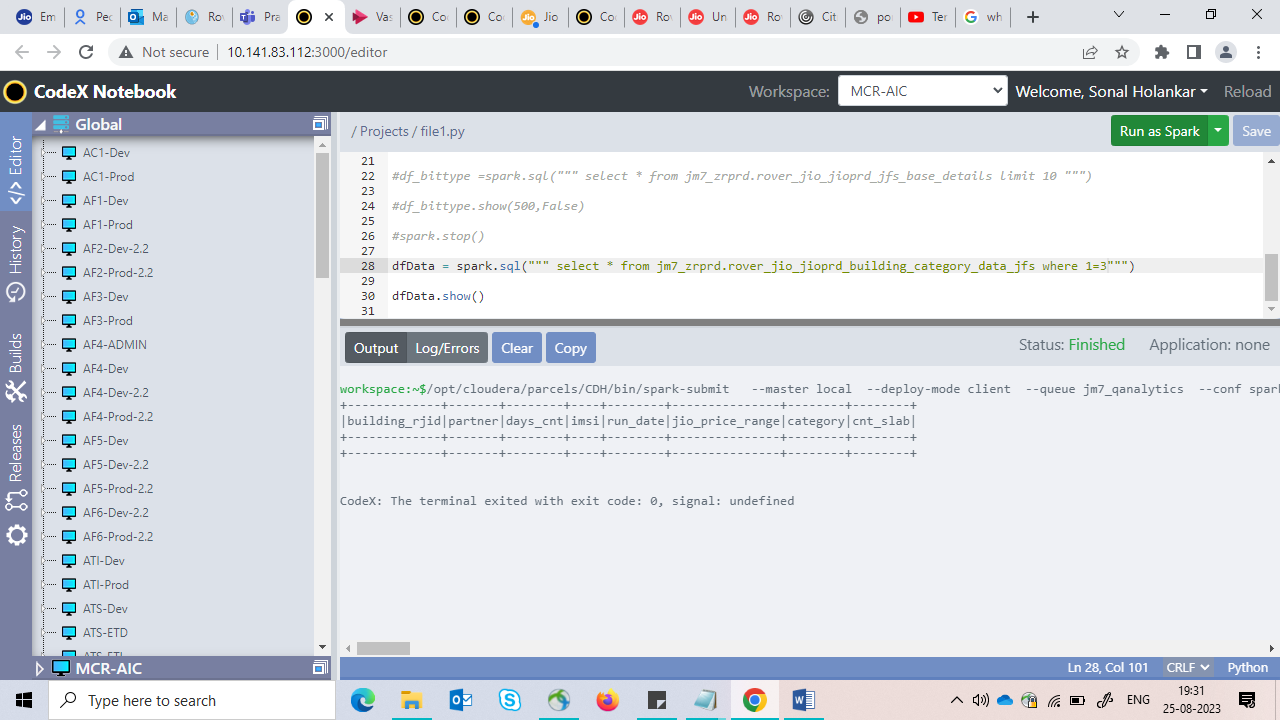


1. where 1=1 can be used to show true condition on table

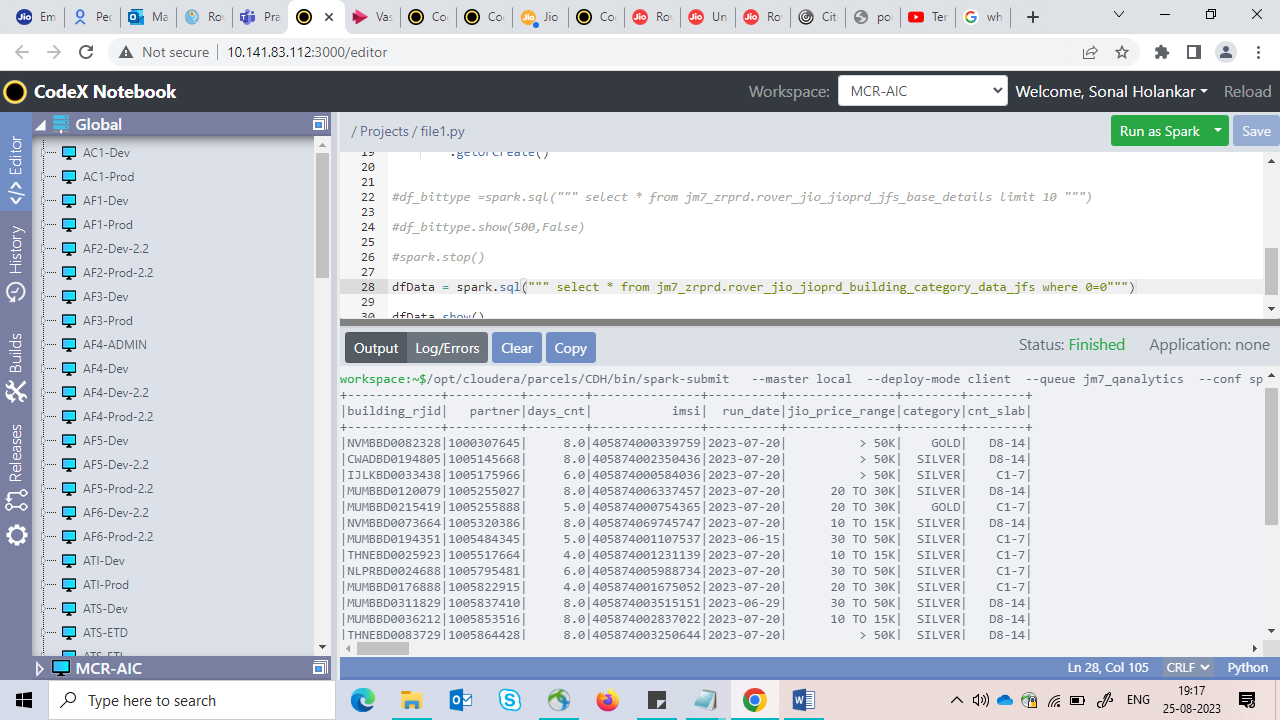


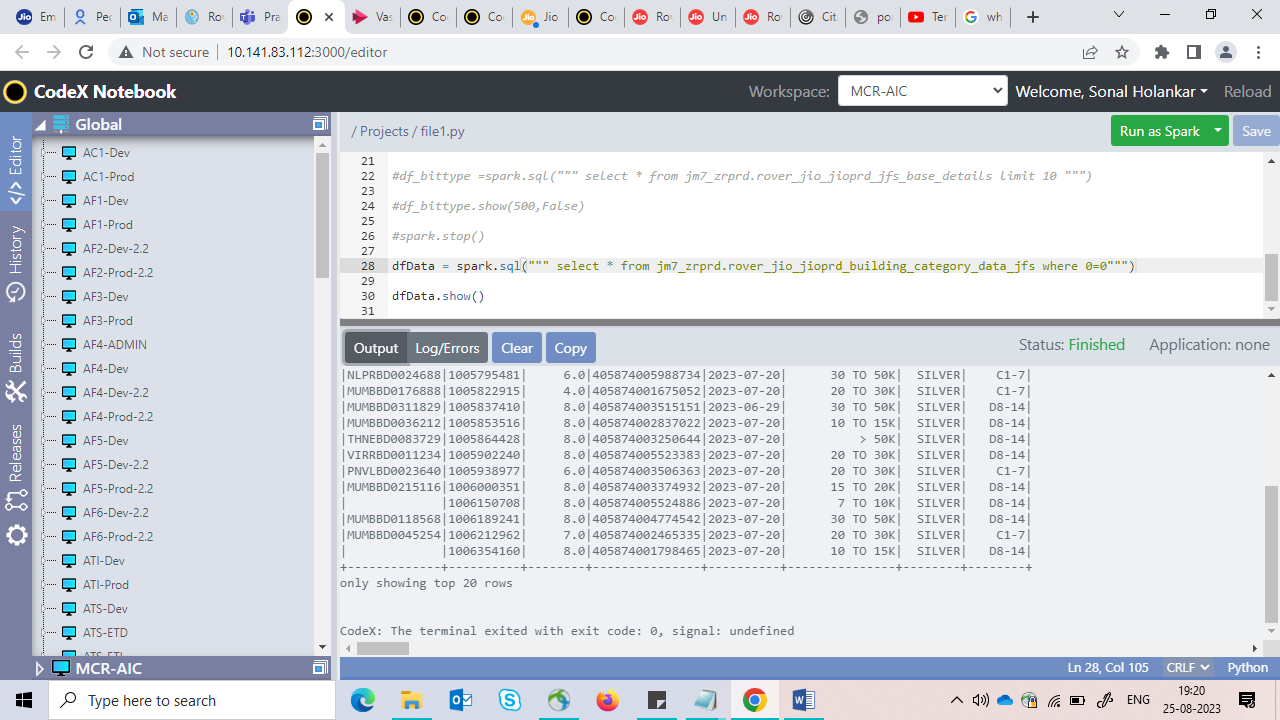
1. where 1=2 create a field-copy of the existing table with no data



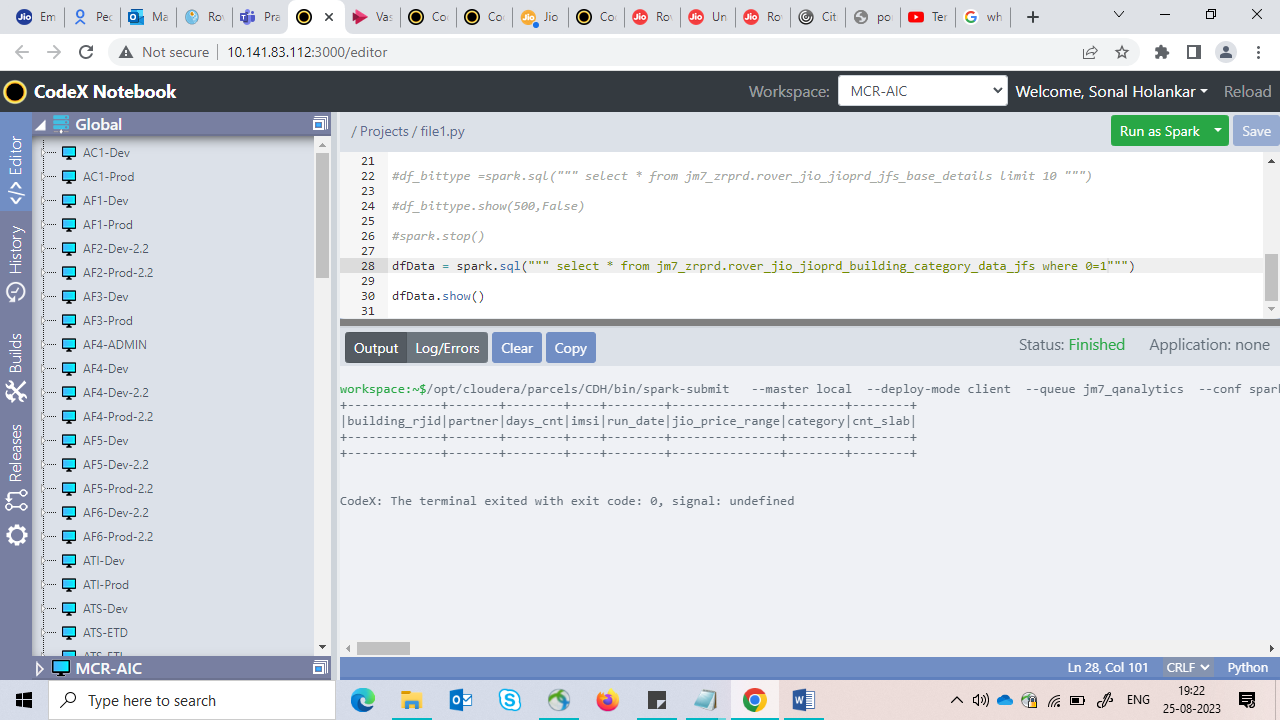


1. where 0=0 always true and your condition will always be executed

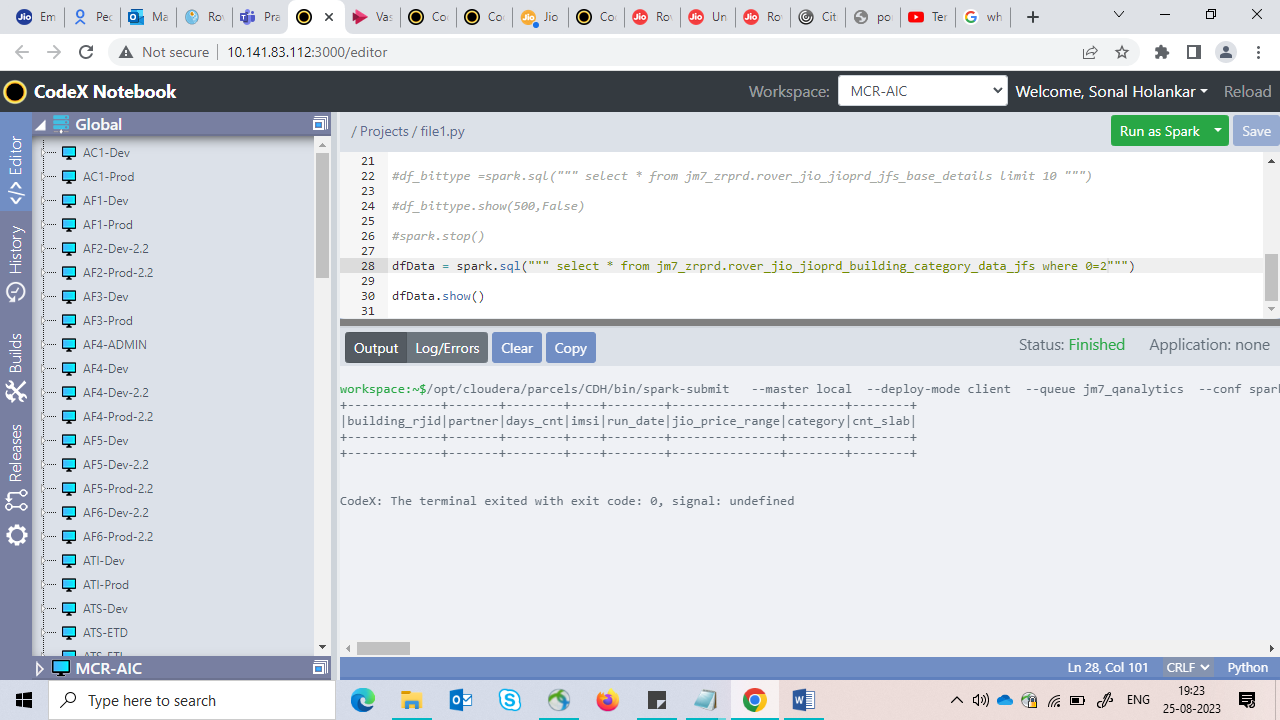




1. where 0=1 can be used to show true condition on table



1. where 0=2 create a field-copy of the existing table with no data



**References:**

[**https://www.agiratech.com/apache-ranger-0-7-1-installation**](https://www.agiratech.com/apache-ranger-0-7-1-installation)

<https://cwiki.apache.org/confluence/display/RANGER/Ranger+authorization+and+auditing+for+Yarn>