Cassandra Security:

By default, Authentication and Authorization features are disabled as Cassandra is configured to easily find and be found by other members of a cluster.

Authentication:

Authentication is pluggable in Cassandra and is configured using the **authenticator** setting in **cassandra.yaml**.

The location of cassandra.yaml file is *etc*/cassandra/ for the docker installation.

By default, Cassandra is configured with **AllowAllAuthenticator** which performs no authentication checks and therefore requires no credentials. It is used to disable authentication completely.

Enabling Password Authentication:

Just open the cassandra.yaml file in your editor and change the a**utheticator: AllowAllAuthenticator** value with **autheticator:PasswordAuthenticator.** After changing yaml file restart node to take effect changes.

Login with default superuser credentials:

```
cqlsh -u cassandra -p cassandra
```

Create A New Superuser:

As the everyone knows the default superuser credentials it is highly recommended to create your own super user with your custom password.

CREATE ROLE dba WITH SUPERUSER = true AND LOGIN = true AND PASSWORD = 'super';

Now a new **dba** user/role has been created with **super** password, so next time you can login for this node as **cqlsh -u dba -p super**

All the authentication settings are stored inside system_auth keyspace, you can further explore that keyspace for better understanding.

Turning Off the Default Super User:

Once you have created the now it’s the time to turnoff superuser to login.

ALTER ROLE cassandra WITH SUPERUSER = false AND LOGIN = false;

Authorization:

Inside cassandra.yaml change the **authorizer: CassandraAuthorizer** to limit the newly created/creating roles to not access your keyspaces/tables by default. Once you set this Authorization then you can grant/revoke permissions to specific roles/users.

Permissions:

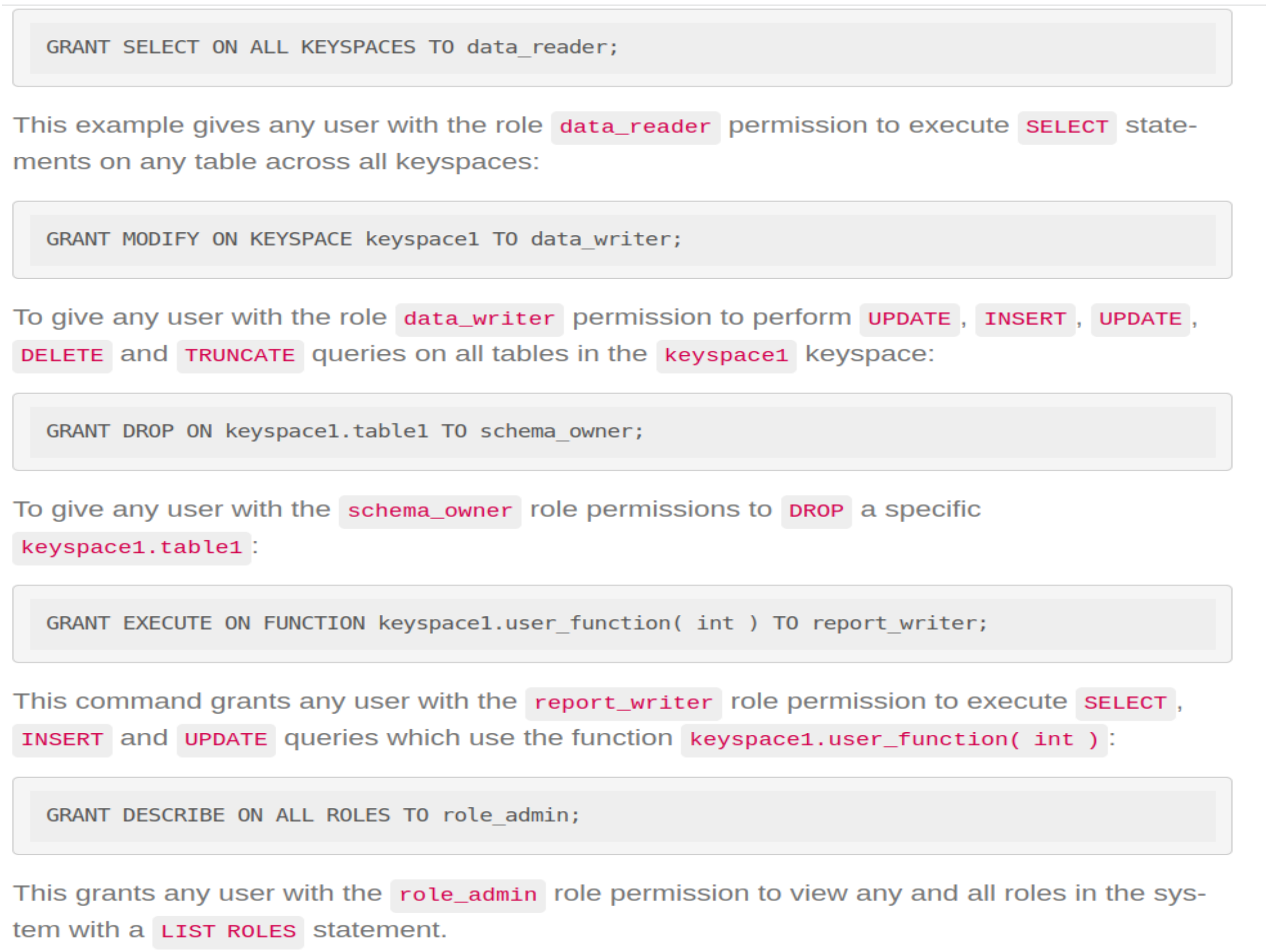
The full set of available permissions is:

- **CREATE**
- **ALTER**
- **DROP**
- **SELECT**
- **MODIFY**
- **AUTHORIZE**
- **DESCRIBE**
- **EXECUTE**

Syntax:

GRANT permissions ON resource TO role_name;

Examples:



```
GRANT SELECT ON ALL KEYSPACES TO data_reader;
```

This example gives any user with the role `data_reader` permission to execute `SELECT` statements on any table across all keyspaces:

```
GRANT MODIFY ON KEYSPACE keyspace1 TO data_writer;
```

To give any user with the role `data_writer` permission to perform `UPDATE`, `INSERT`, `UPDATE`, `DELETE` and `TRUNCATE` queries on all tables in the `keyspace1` keyspace:

```
GRANT DROP ON keyspace1.table1 TO schema_owner;
```

To give any user with the `schema_owner` role permissions to `DROP` a specific `keyspace1.table1`:

```
GRANT EXECUTE ON FUNCTION keyspace1.user_function( int ) TO report_writer;
```

This command grants any user with the `report_writer` role permission to execute `SELECT`, `INSERT` and `UPDATE` queries which use the function `keyspace1.user_function( int )`:

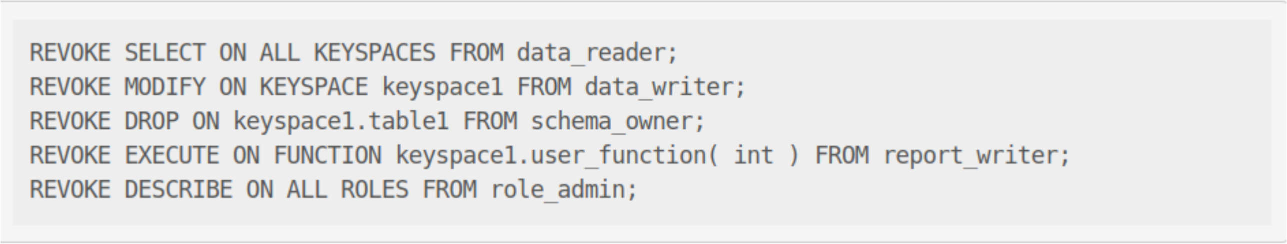
```
GRANT DESCRIBE ON ALL ROLES TO role_admin;
```

This grants any user with the `role_admin` role permission to view any and all roles in the system with a `LIST ROLES` statement.

REVOKE PERMISSION:

REVOKE permissions ON resource FROM role_name;

Examples:



```
REVOKE SELECT ON ALL KEYSPACES FROM data_reader;
REVOKE MODIFY ON KEYSPACE keyspace1 FROM data_writer;
REVOKE DROP ON keyspace1.table1 FROM schema_owner;
REVOKE EXECUTE ON FUNCTION keyspace1.user_function( int ) FROM report_writer;
REVOKE DESCRIBE ON ALL ROLES FROM role_admin;
```

Note: You can find more on Data control here: https://cassandra.apache.org/doc/latest/cassandra/cql/security.html#data-control

References:

Authentication/Authorization: https://cassandra.apache.org/doc/latest/cassandra/operating/security.html#authentication

Database Roles: https://cassandra.apache.org/doc/latest/cassandra/cql/security.html#cql-roles