MySQL & MongoDB Cheat Sheet for Ubuntu

Section 1 — MySQL (Ubuntu) — Install, Configure & Use

Update packages:

```
sudo apt update && sudo apt upgrade -y
```

Install MySQL server: `sudo apt install mysql-server -y` **Check MySQL service status:** `sudo systemctl status mysql`

Start/Enable MySQL:

```
sudo systemctl start mysql sudo
systemctl enable mysql
```

Run security script (set root password, remove test DB): `sudo mysql_secure_installation`

Login as root (Ubuntu auth_socket vs password):

```
sudo mysql --  or if using password: mysql
-u root -p
```

Create admin user and grant privileges:

```
CREATE USER 'admin'@'%' IDENTIFIED BY 'StrongPass123';
GRANT ALL PRIVILEGES ON *.* TO 'admin'@'%' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

Create a database and user:

```
CREATE DATABASE mydb;
CREATE USER 'appuser'@'localhost' IDENTIFIED BY 'AppPass!';
GRANT ALL PRIVILEGES ON mydb.* TO 'appuser'@'localhost';
FLUSH PRIVILEGES;
```

Allow remote connections (if needed) — edit bind-address:

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
# change bind-address = 127.0.0.1 to 0.0.0.0
sudo systemctl restart mysql
```

Open firewall for MySQL (optional, UFW):

```
sudo ufw allow 3306/tcp
sudo ufw reload
```

Basic SQL commands:

```
SHOW DATABASES;
USE mydb;
CREATE TABLE users(id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100));
INSERT INTO users(name) VALUES('Karan');
SELECT * FROM users;
```

Backup & Restore:

```
Backup: mysqldump -u appuser -p mydb > mydb_backup.sql
Restore: mysql -u appuser -p mydb < mydb_backup.sql
```

Check error log: `sudo tail -n 200 /var/log/mysql/error.log`

Quick security tips: - Use strong passwords and do not expose root remotely. - Use least-privilege accounts for applications. - Keep regular backups (automate with cron). - Use TLS for remote connections if exposing MySQL externally.

Prepared By: Karan Salunkhe

Section 2 — MongoDB (Ubuntu) — Install, Configure & Use

Import the public GPG key: `wget -qO - https://www.mongodb.org/static/pgp/server-6.0.asc | sudo apt-key add` **Create list file (Ubuntu 22.04 example):**

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu jammy/mongodb-org/6.0 multiverse
```

`sudo tee /etc/apt/sources.list.d/mongodb-org-6.0.list` **Update packages:** `sudo apt update` **Install MongoDB:** `sudo apt install -y mongodb-org` **Start and enable service:**

```
sudo systemctl start mongod
sudo systemctl enable mongod
```

Check service status:

`sudo systemctl status mongod` **Verify logs (if needed):** `sudo journalctl -u mongod --no-pager --since "5 minutes ago"` **Basic mongo shell (mongosh):**

```
mongosh --host 127.0.0.1 --port 27017
# or simply: mongosh
```

Create admin user (enable auth later):

```
use admin
db.createUser({user:'admin',pwd:'StrongPass123',roles:[{role:'root',db:'admin'}]})
```

Enable authentication (edit config):

`sudo nano /etc/mongod.conf # under security: add 'authorization: "enabled"' sudo systemctl restart mongod` **Connect with auth:** `mongosh -u admin -p StrongPass123 --authenticationDatabase admin` **Create a DB and collection:**

```
use mydb db.createCollection('users')
db.users.insertOne({name:'Karan', age:23})
db.users.find().pretty()
```

Backup and restore (mongodump/mongorestore):

`Backup: mongodump --db mydb --out /tmp/mydb_dump`

`Restore: mongorestore --db mydb /tmp/mydb_dump/mydb` **Enable remote connections (bindIp):**

```
sudo nano /etc/mongod.conf # change
bindIp: 127.0.0.1 to 0.0.0.0 sudo
systemctl restart mongod # open port
sudo ufw allow 27017/tcp
```

Disable Transparent Huge Pages (recommended for prod): `echo never | sudo tee /sys/kernel/mm/transparent_hugepage/enabled` **Security tips:**

`Use role-based access; enable TLS; do not expose DB without firewall; enable IP whitelisting; enabl`

MongoDB Quick Examples:

```
Common MongoDB CRUD (mongosh) -
Insert many :
db.users.insertMany([{name:'A'},{name:'B'}])      -
Find :
db.users.find({age:{$gte:20}}).sort({name:1}).limit(5).pretty()        -
Update :
db.users.updateOne({name:'Karan'}, {$set:{age:24}})
db.users.updateMany({active:true}, {$inc:{score:1}}) -  Delete :
db.users.deleteOne({name:'A'})
db.users.deleteMany({active:false}) -
Index :
db.users.createIndex({email:1}, {unique:true})
```

Prepared By: Karan Salunkhe