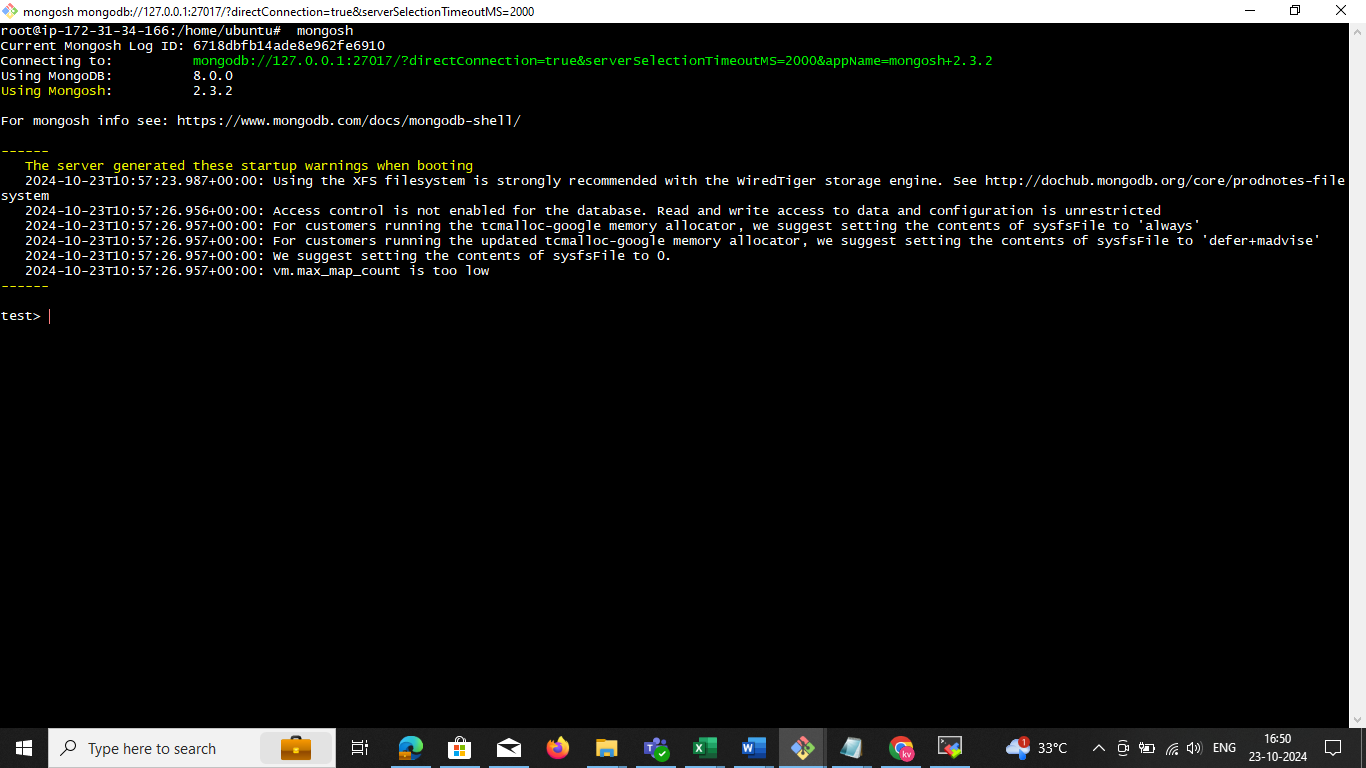
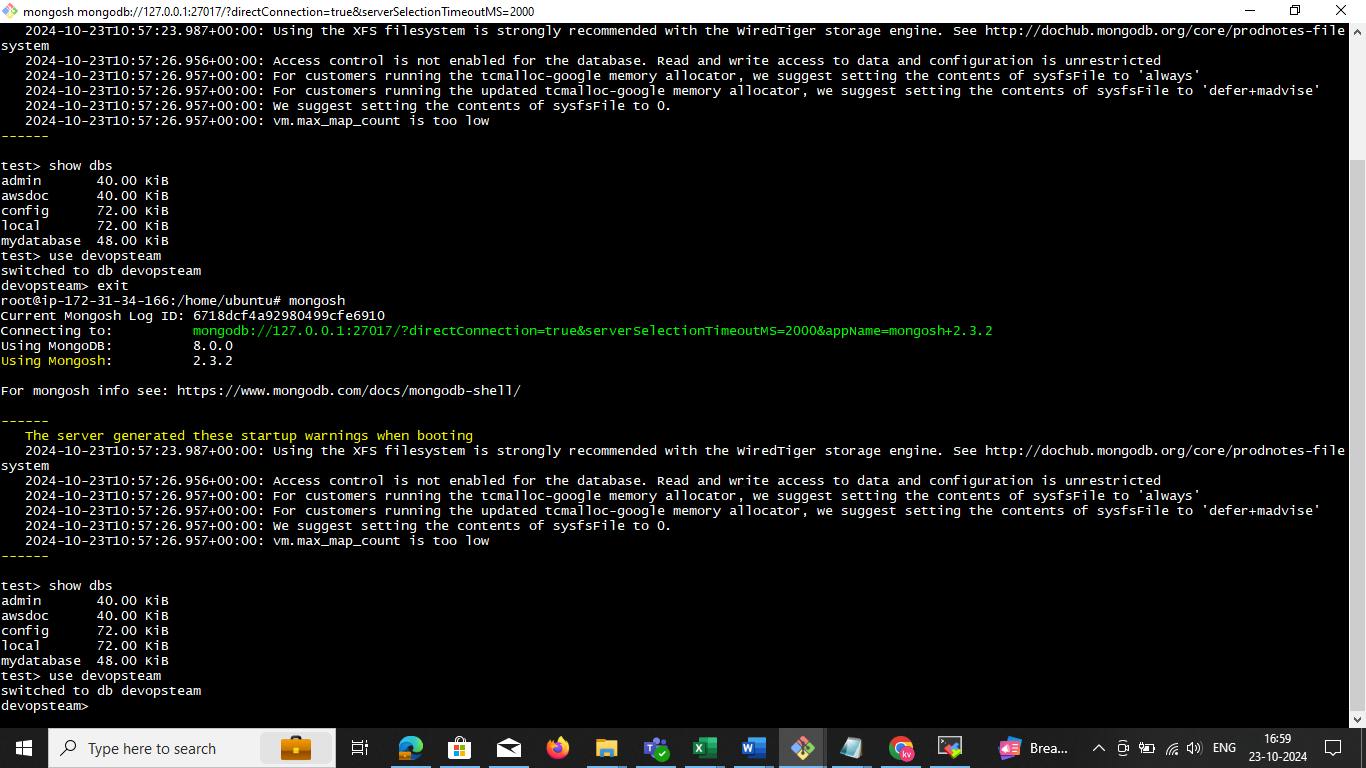
Login mongodb

If you docker continer check continer is running or not log login the

**login to the** mongosh command



**show dbs if exit db we need login use <dbname> create db**



**Crate the collection in**

**db.createCollection("azure")**

**we need to see the collection**

show collections

now you can see the collection data



 **Insert Data:**

db.mycollection.insertOne({ name: "Alice", age: 25, city: "New York" })

 **Insert Multiple Documents:**

db.mycollection.insertMany([

{ name: "Bob", age: 30, city: "Chicago" },

{ name: "Charlie", age: 28, city: "San Francisco" }

])

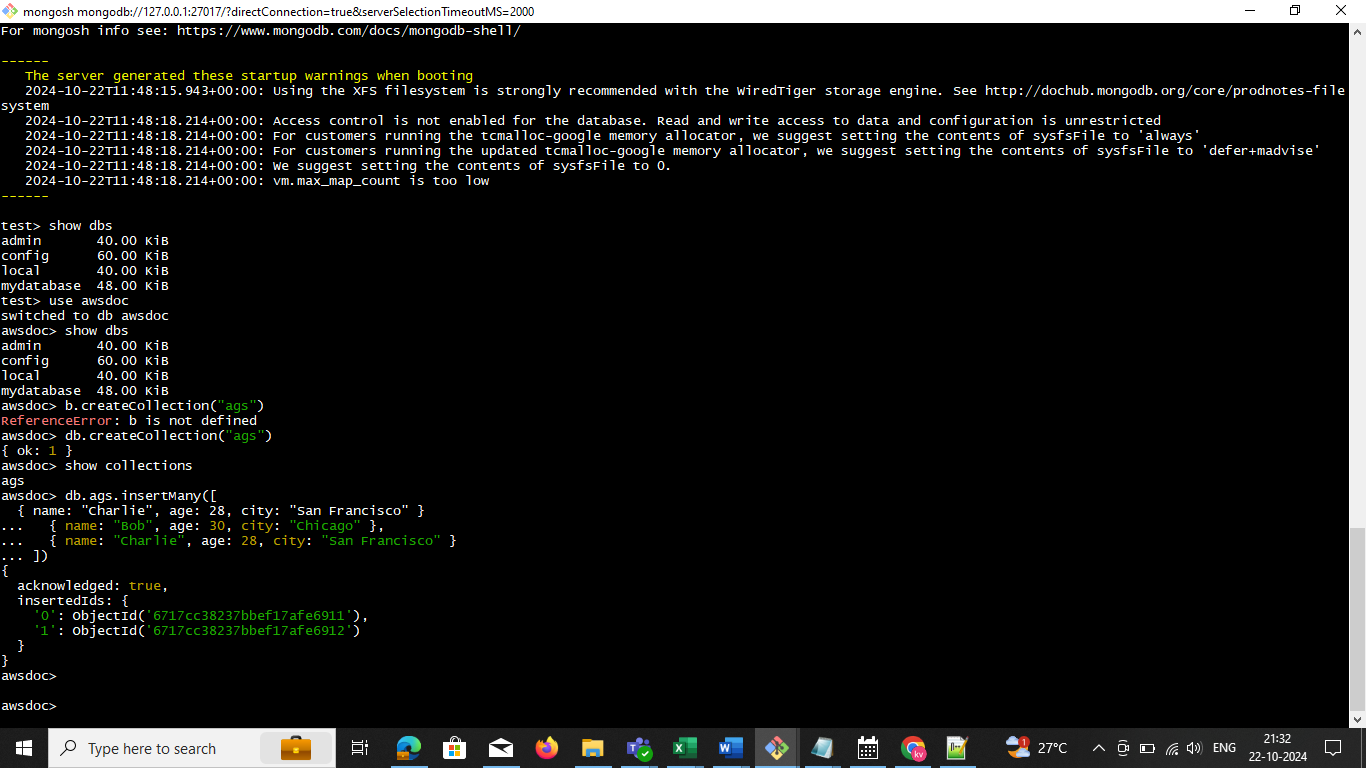
Chane the name hear

db. azure.insertMany([

{ name: "Bob", age: 30, city: "Chicago" },

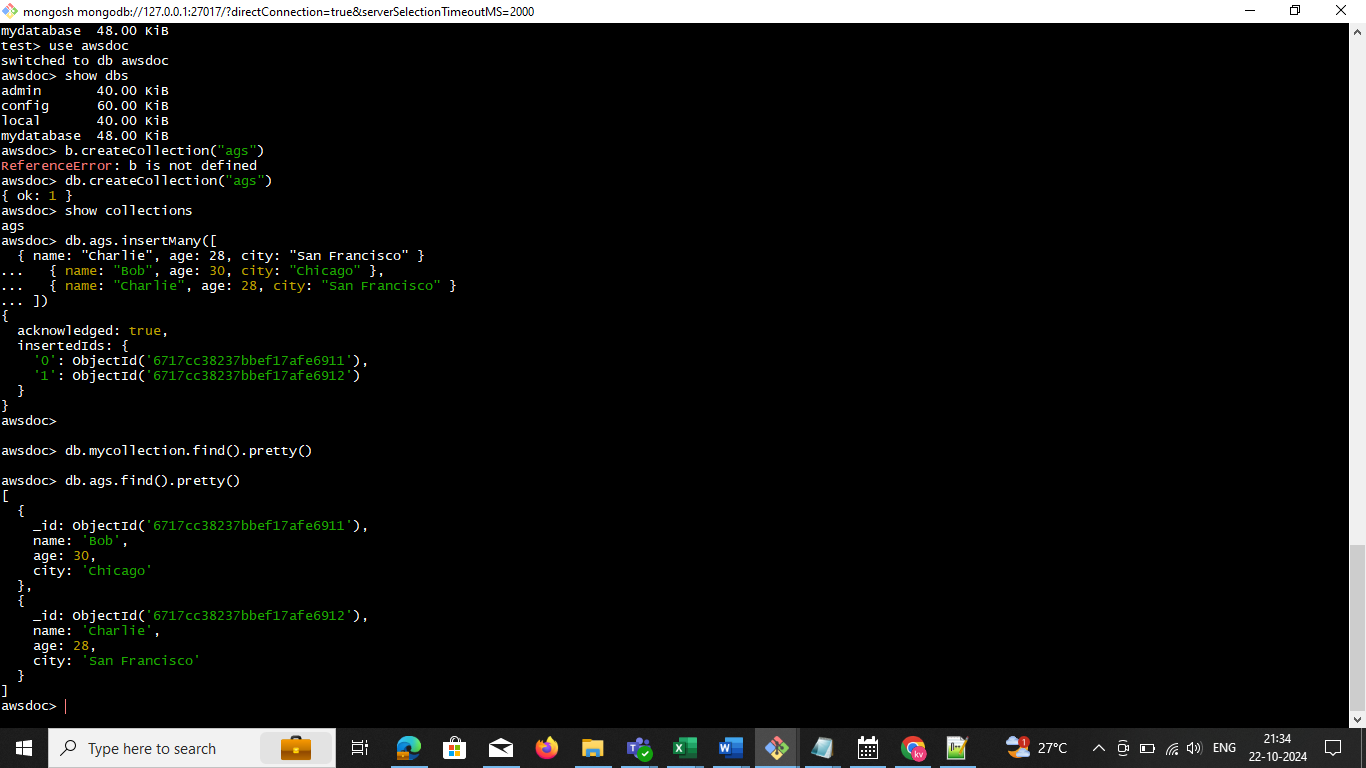
{ name: "Charlie", age: 28, city: "San Francisco" }

])



**Verify Inserted Data:**

db.mycollection.find().pretty() change name give collection name

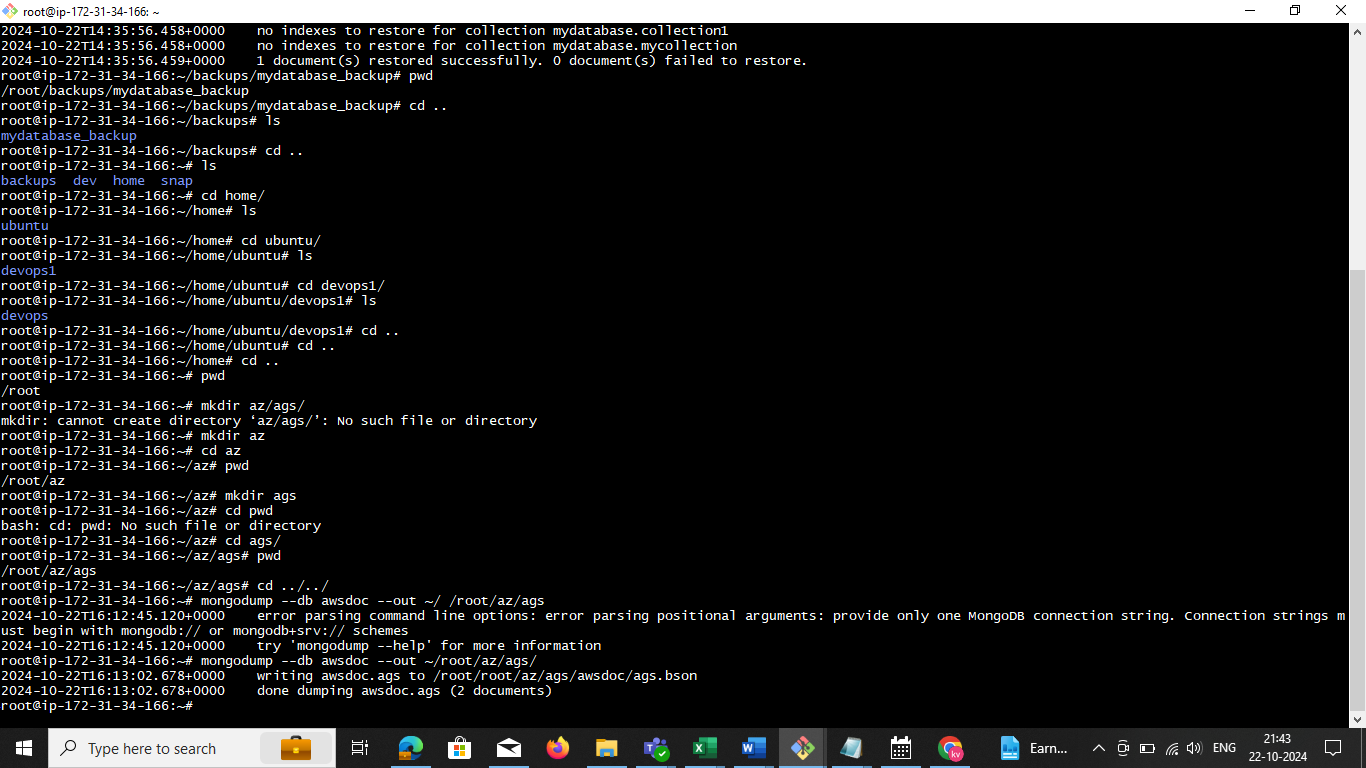


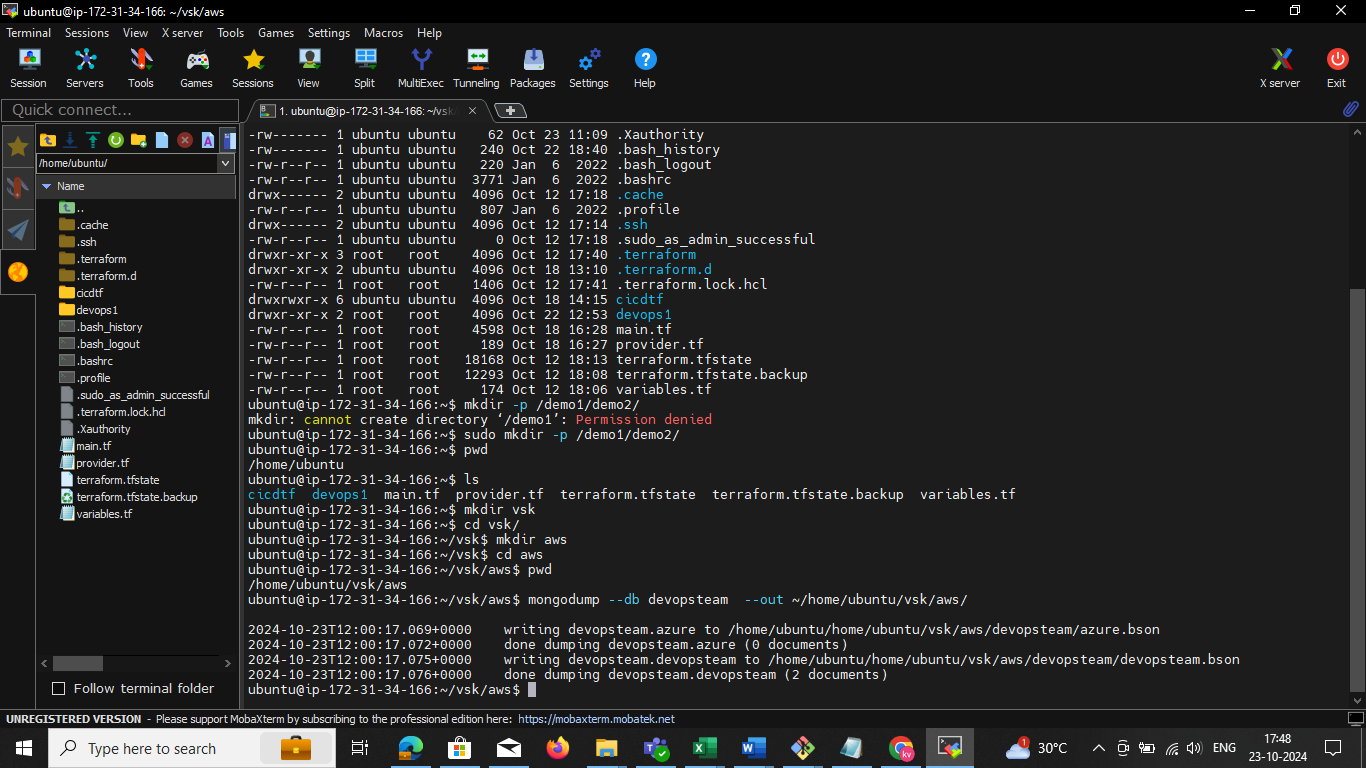
**Now, can task dump the db in to another folder**

Exit form mongosh and go to root are the

**mongodump --db mydatabase --out ~/backups/mydatabase\_backup/**

**<nameyourdb> <set your path > where you need to dump**

mongodump --db awsdoc --out ~/ /root/az/ags

 **I dumping the particular db in my custom dir-path**

kubectl exec -it mongodb-pod -- mongodump --db mgdata --out /home/ubuntu/devopd/

kubectl cp mongodb-pod:/home/ubuntu/ags/ /home/ubuntu/ags/

ls -la /home/ubuntu/uat/

kubectl exec -it mongodb-pod -- mongodump --out /home/ubuntu/allbackup/

kubectl cp mongodb-pod:/home/ubuntu/allbackup/ /home/ubuntu/allbackup/

ls -la /home/ubuntu/allbackup/

**BACKUP AND RESTORE**

step1:

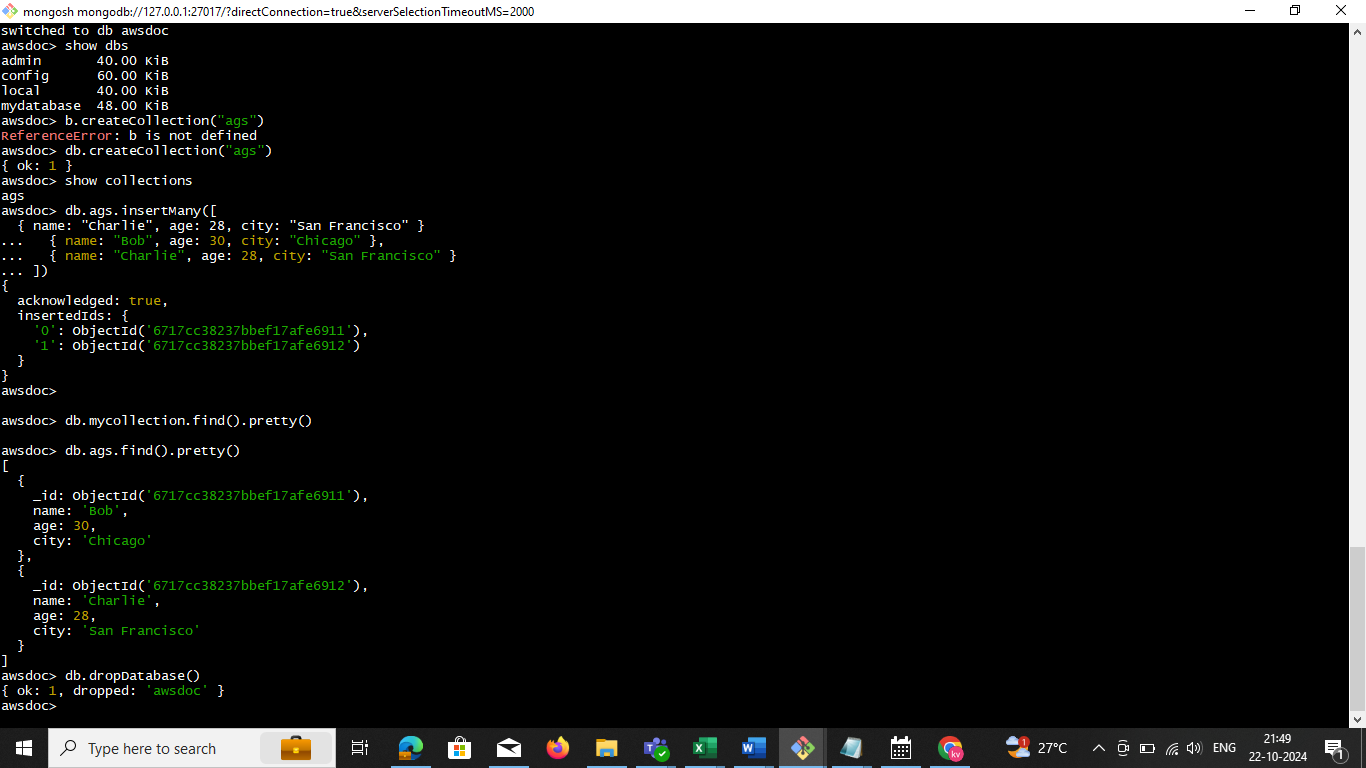
**---------------------------------------------**

Lets drop the database with the below command

**Drop the Database:** Once you've selected the database, use the dropDatabase() command to drop it:

**Javascript (your db name ) will be change**

**db.dropDatabase() now you need check db is dropped in below screen**



Check the collections

**EXP:**

mongorestore --db mydatabase ~/backups/mydatabase\_backup/mydatabase/

**mongorestore --db awsdoc ~ /root/az/ags/awsdoc /**

mongorestore --db awsdoc ~ /root/root/az/ags/awsdoc

mongorestore --db awsdoc ~ /root/az/ags/awsdoc/

2024-10-22T16:52:15.743+0000 error parsing command line options: error parsing positional arguments: provide only one polling interval in seconds and only one MongoDB connection string. Connection strings must begin with mongodb:// or mongodb+srv:// schemes 2024-10-22T16:52:15.743+0000 try 'mongorestore --help' for more information

The error in your mongorestore command is likely due to an incorrect space between ~ and the backup path. Ensure the path is written correctly without unintended spaces. Here's the corrected command

bash

Copy code

**mongorestore --db awsdoc ~/root/az/ags/awsdoc/**

Explanation:

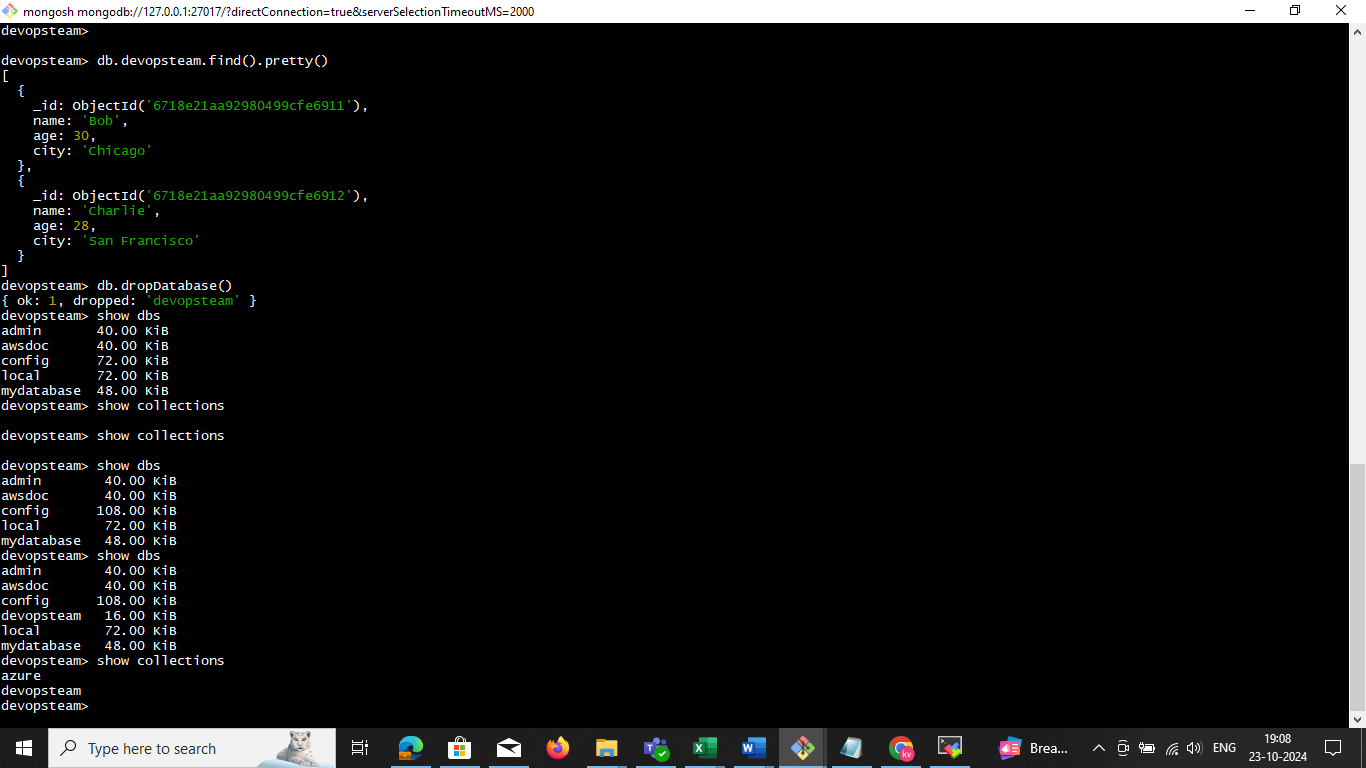
--db awsdoc: Restores the data into the awsdoc database.

~/root/az/ags/awsdoc/: Path to the backup directory.

Make sure MongoDB is running, and the backup path is accurate. If authentication is required, add --username and --password flags accordingly.

-------------------------------------------steps 2 --------------------------------------------------------------------------------

Ones you delete the db collection and DB using **showdbs ,collection ,**



**Note:** now my db and collection was deleted and

**Sep3:** now we take backup and restore in data

mongorestore --db devopsteam ~/home/ubuntu/vsk/aws/devopsteam/

**Note:** check the path ~ space I create

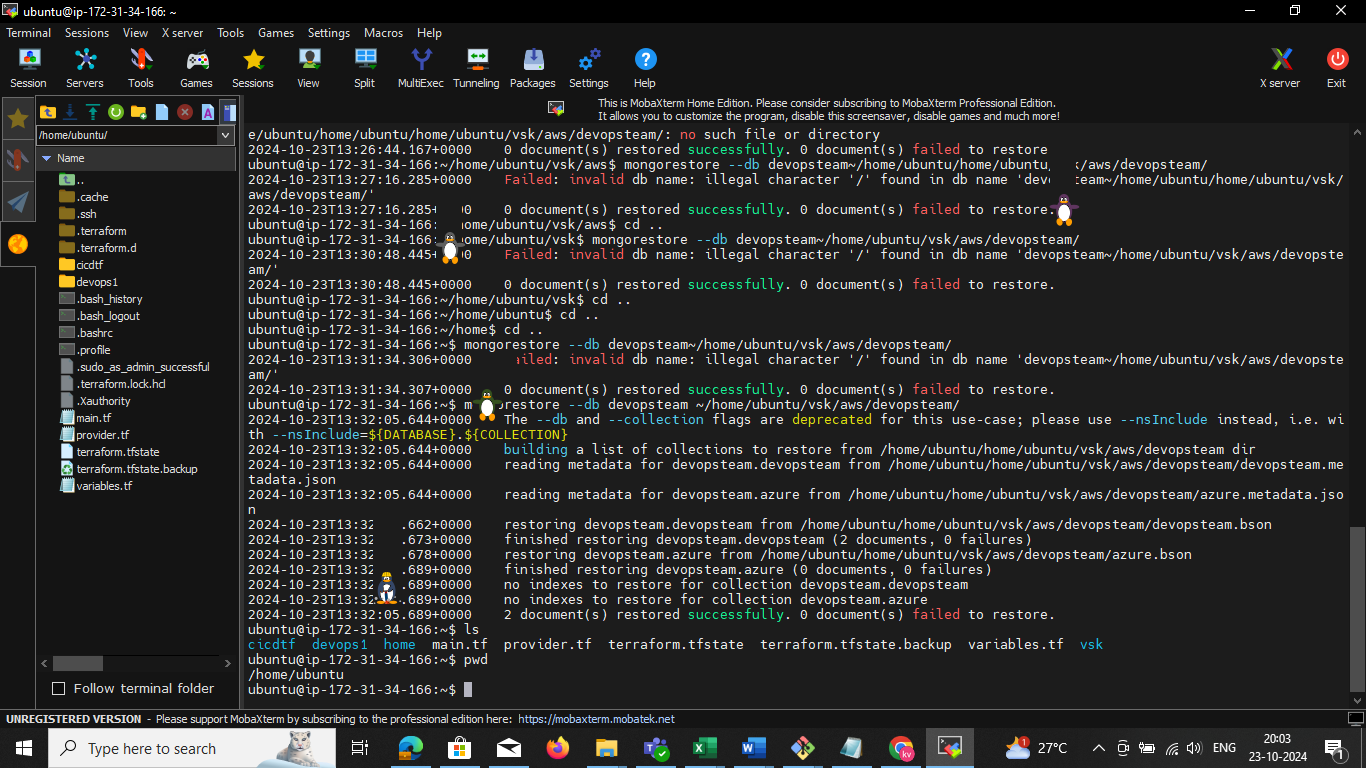
**ubuntu@ip-172-31-34-166:~$ pwd**

**/home/ubuntu**

I have created dir vsk /aws/ **go to specific path home ubuntu and task the mangodb restore**

**mongorestore --db devopsteam ~/home/ubuntu/vsk/aws/devopsteam/**

**devopsteam is called db user name in mongodb**



**--------------------------------------------------------------------EKS ------------------------------------------------------------**

To install Minikube on Ubuntu, you need the following minimum requirements:

* **CPU**: At least two CPUs

* **Memory**: 2 GB of free memory

* **Disk space**: 20 GB of free disk space

* **Internet connection**: An internet connection

* **Container or virtual machine manager**: A container or virtual machine manager, such as Docker, QEMU, Hyperkit, Hyper-V, KVM, Parallels, Podman, VirtualBox, or VMware Fusion/Workstation

* **User**: A user with sudo privileges

Minikube is a tool that runs a single-node Kubernetes cluster inside a virtual machine (VM) on your laptop. It's useful for people who want to try Kubernetes or develop with it on a day-to-day basis

**Install steps in** below

sudo apt-get update -y

**sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common -y**

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

sudo apt-get update -y

sudo apt-get install docker-ce docker-ce-cli containerd.io -y

curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl

chmod +x ./kubectl

sudo mv ./kubectl /usr/local/bin/kubectl

curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64

sudo install minikube-linux-amd64 /usr/local/bin/minikube

####

echo the script is now ready

echo manually run minikube start --vm-driver=docker --cni=calico to start minikube

sudo usermod -aG docker $USER

newgrp docker

minikube start --vm-driver=docker --cni=calico

----------------------------------------**EKS pod backup** ----------------------------------------------------------------

**Step1 :** got to pod kubectl exec -it mongodb-pod bash (or) kubectl exec -it mongodb-pod -- bash   
(or) kubectl exec -it mongodb-pod -- /bin/bash and login to shell mongodb mongosh

**Step2:** create user 🡪 **use devenv** now you can switch in to devenv

**Step3:** create the collection  **db.createCollection("azure")**

**Step4:** Insert Multiple Documents and collection

db.mycollection.insertMany([

{ name: "Bob", age: 30, city: "Chicago" },

{ name: "Charlie", age: 28, city: "San Francisco" }

])

db. **pod**.insertMany([

{ name: "Bob", age: 30, city: "Chicago" },

{ name: "pod", age: 29, city: "San Francisco" }

])

**Step5**: use this command single DB

kubectl exec -it mongodb-pod -- mongodump --db **mgdata** --out /home/ubuntu/**devopd**/

kubectl cp mongodb-pod:/home/ubuntu/ags/ /home/ubuntu/ags/

ls -la /home/ubuntu/uat/

kubectl exec -it mongodb-pod -- mongodump --out /home/ubuntu/**allbackup**/

kubectl cp mongodb-pod:/home/ubuntu/**allbackup**/ /home/ubuntu/**allbackup**/

ls -la /home/ubuntu**/allbackup/**