



Redis DB & Redis Commander

May 2024

Table of Contents

Login with SSH	2
Configuring Redis	5
Redis Commander	9
Redis Commander Web Dashboard	10
Add a New Connection	11

Login with SSH

Login with SSH access using the ip of your VM, using the following command:

```
ssh admin@your_vm_ip
```

[illegible]

Now that you've successfully logged in, first make sure you have redis installed before proceeding further.

Step 1: Install Redis on Ubuntu using the package manager:

```
sudo apt update
```

```
sudo apt install redis-server
```

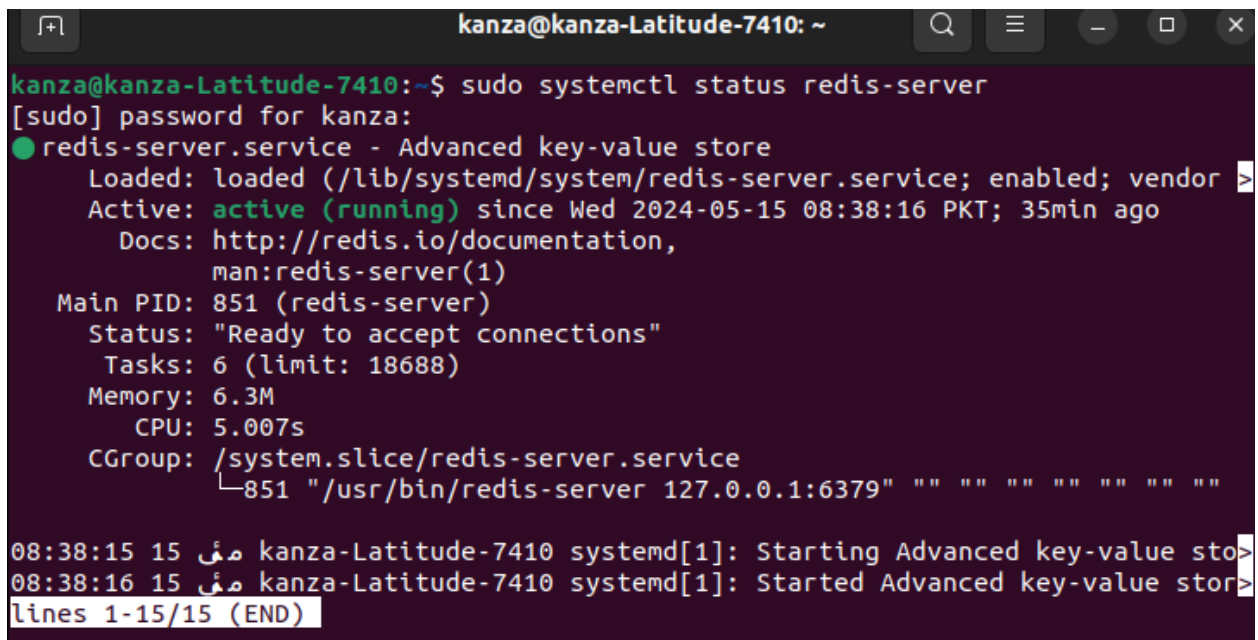
Step 2: Start the Redis server:

```
sudo systemctl start redis-server
```

Step 3: Verification

Check the status of Redis:

```
sudo systemctl status redis-server
```



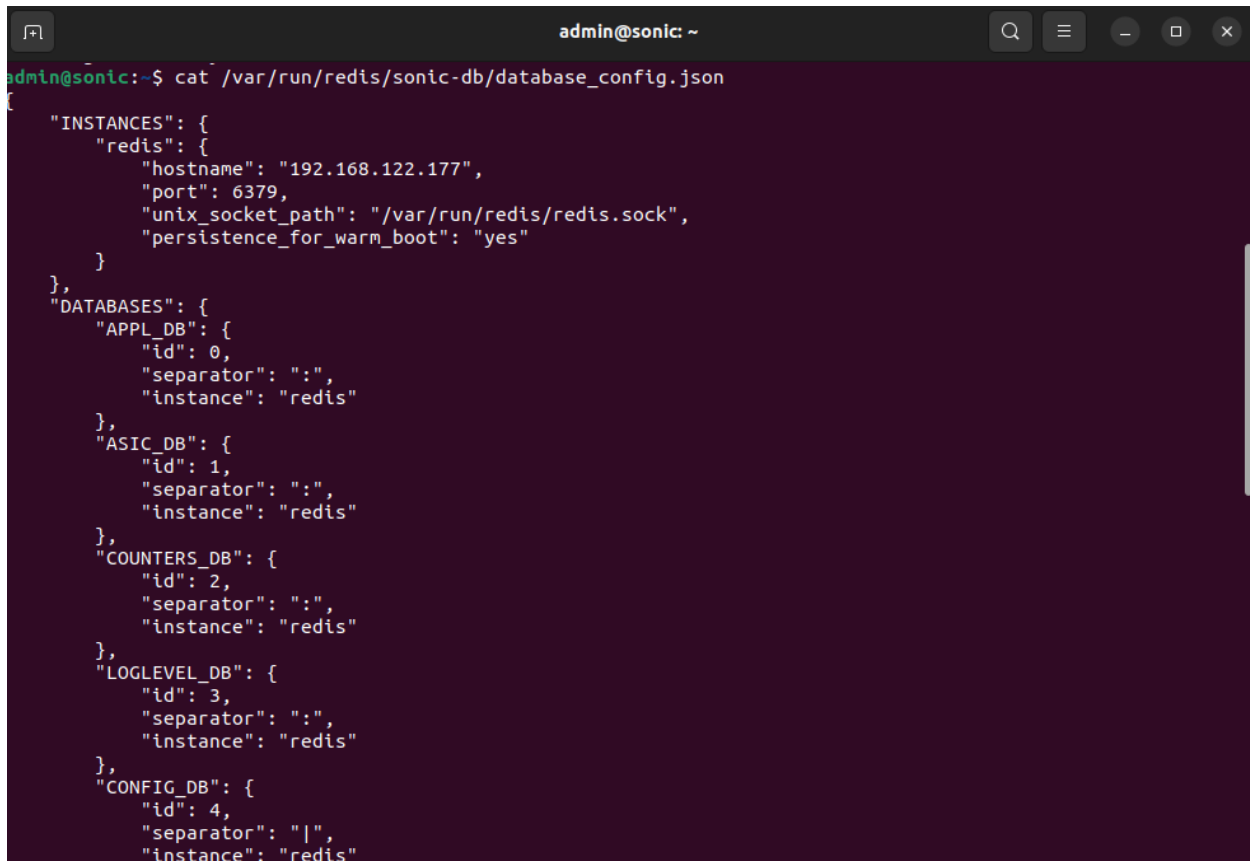
```
kanza@kanza-Latitude-7410: ~  
kanza@kanza-Latitude-7410:~$ sudo systemctl status redis-server  
[sudo] password for kanza:  
● redis-server.service - Advanced key-value store  
   Loaded: loaded (/lib/systemd/system/redis-server.service; enabled; vendor  
   Active: active (running) since Wed 2024-05-15 08:38:16 PKT; 35min ago  
     Docs: http://redis.io/documentation,  
           man:redis-server(1)  
  Main PID: 851 (redis-server)  
    Status: "Ready to accept connections"  
   Tasks: 6 (limit: 18688)  
  Memory: 6.3M  
     CPU: 5.007s  
   CGroup: /system.slice/redis-server.service  
           └─851 "/usr/bin/redis-server 127.0.0.1:6379" "" "" "" "" "" "" ""  
  
08:38:15 15 مئی kanza-Latitude-7410 systemd[1]: Starting Advanced key-value sto>  
08:38:16 15 مئی kanza-Latitude-7410 systemd[1]: Started Advanced key-value stor>  
lines 1-15/15 (END)
```

Configuring Redis

Configuration File

Step 1: Locate the Redis configuration file and open it:

```
cat /var/run/redis/sonic-db/database_config.json
```



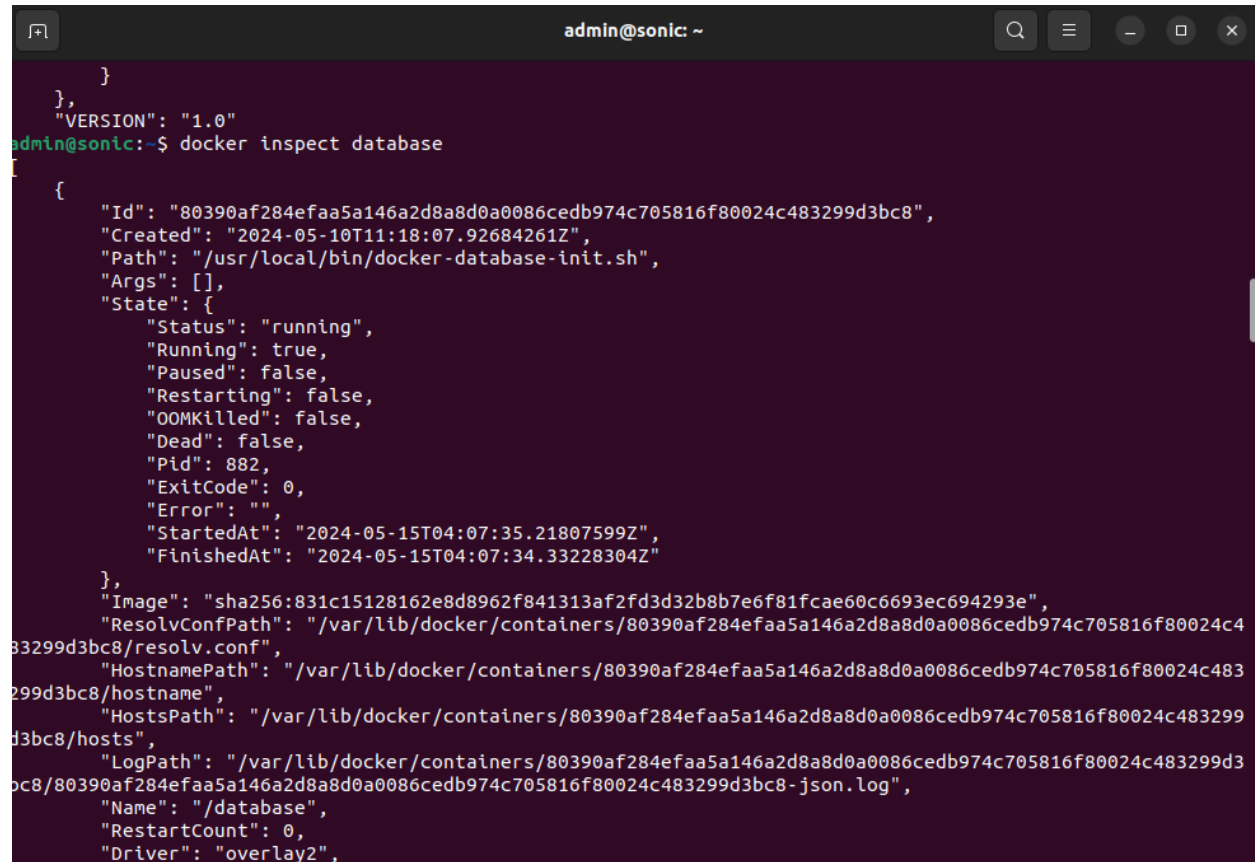
A terminal window titled 'admin@sonic: ~' showing the command 'cat /var/run/redis/sonic-db/database_config.json' and its output. The output is a JSON configuration file for Redis, defining instances and databases.

```
admin@sonic:~$ cat /var/run/redis/sonic-db/database_config.json
{
  "INSTANCES": {
    "redis": {
      "hostname": "192.168.122.177",
      "port": 6379,
      "unix_socket_path": "/var/run/redis/redis.sock",
      "persistence_for_warm_boot": "yes"
    }
  },
  "DATABASES": {
    "APPL_DB": {
      "id": 0,
      "separator": ":",
      "instance": "redis"
    },
    "ASIC_DB": {
      "id": 1,
      "separator": ":",
      "instance": "redis"
    },
    "COUNTERS_DB": {
      "id": 2,
      "separator": ":",
      "instance": "redis"
    },
    "LOGLEVEL_DB": {
      "id": 3,
      "separator": ":",
      "instance": "redis"
    },
    "CONFIG_DB": {
      "id": 4,
      "separator": "|",
      "instance": "redis"
    }
  }
}
```

Step 2: Inspect the Docker Container

Did the inspection of the Docker Container using the following command:

```
docker inspect database
```



```
admin@sonic: ~  
}   
},  
  "VERSION": "1.0"  
admin@sonic:~$ docker inspect database  
[  
  {  
    "Id": "80390af284efaa5a146a2d8a8d0a0086cedb974c705816f80024c483299d3bc8",  
    "Created": "2024-05-10T11:18:07.92684261Z",  
    "Path": "/usr/local/bin/docker-database-init.sh",  
    "Args": [],  
    "State": {  
      "Status": "running",  
      "Running": true,  
      "Paused": false,  
      "Restarting": false,  
      "OOMKilled": false,  
      "Dead": false,  
      "Pid": 882,  
      "ExitCode": 0,  
      "Error": "",  
      "StartedAt": "2024-05-15T04:07:35.21807599Z",  
      "FinishedAt": "2024-05-15T04:07:34.33228304Z"  
    },  
    "Image": "sha256:831c15128162e8d8962f841313af2fd3d32b8b7e6f81fcae60c6693ec694293e",  
    "ResolvConfPath": "/var/lib/docker/containers/80390af284efaa5a146a2d8a8d0a0086cedb974c705816f80024c483299d3bc8/resolv.conf",  
    "HostnamePath": "/var/lib/docker/containers/80390af284efaa5a146a2d8a8d0a0086cedb974c705816f80024c483299d3bc8/hostname",  
    "HostsPath": "/var/lib/docker/containers/80390af284efaa5a146a2d8a8d0a0086cedb974c705816f80024c483299d3bc8/hosts",  
    "LogPath": "/var/lib/docker/containers/80390af284efaa5a146a2d8a8d0a0086cedb974c705816f80024c483299d3bc8/80390af284efaa5a146a2d8a8d0a0086cedb974c705816f80024c483299d3bc8-json.log",  
    "Name": "/database",  
    "RestartCount": 0,  
    "Driver": "overlay2",  
  }  
]
```

Step 3: Open the configuration file with a text editor.

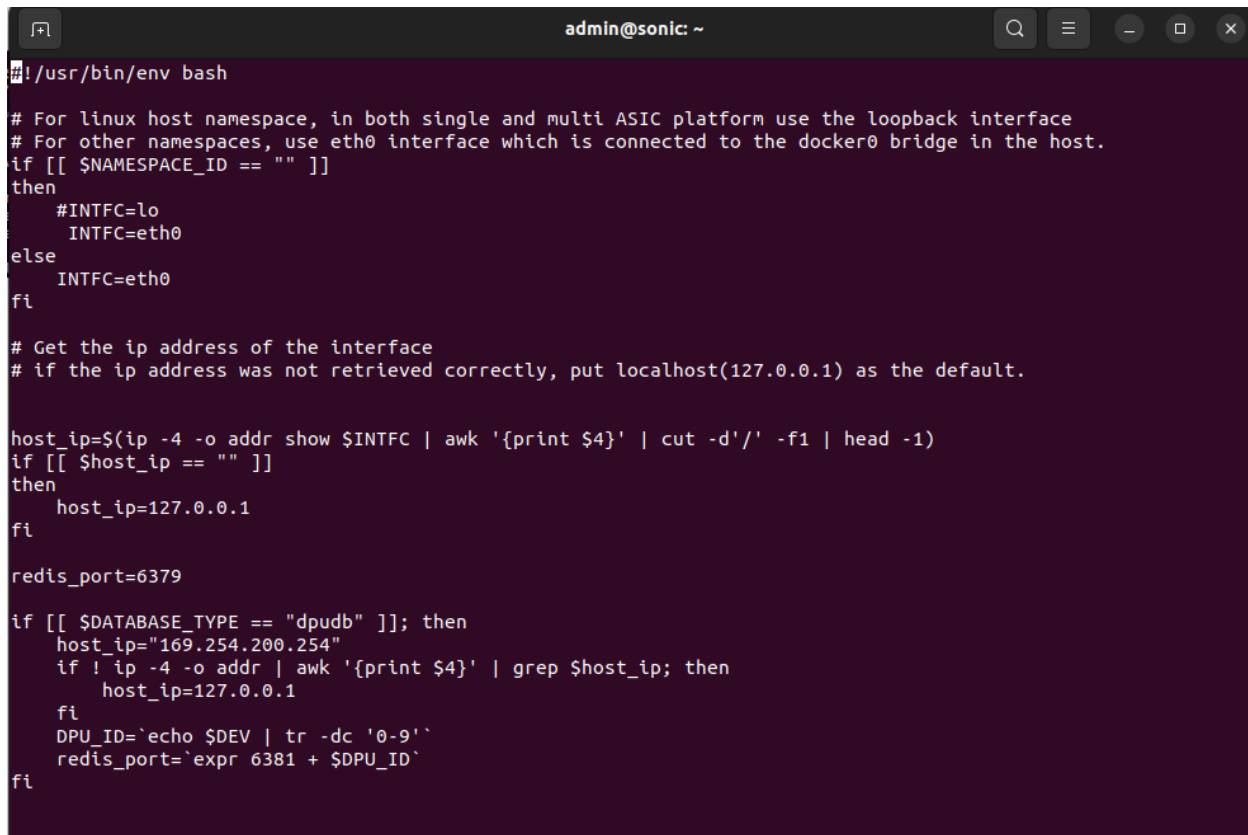
I'm using the vi editor

```
docker exec -it database bash
```

```
vi /usr/local/bin/docker-database-init.sh
```

Step 4: Modify it as needed.

```
admin@sonic:~$ docker exec -it database bash
root@sonic:/# vi /usr/local/bin/docker-database-init.sh
```



```
admin@sonic: ~
#!/usr/bin/env bash

# For linux host namespace, in both single and multi ASIC platform use the loopback interface
# For other namespaces, use eth0 interface which is connected to the docker0 bridge in the host.
if [[ $NAMESPACE_ID == "" ]]
then
    #INTFC=lo
    INTFC=eth0
else
    INTFC=eth0
fi

# Get the ip address of the interface
# if the ip address was not retrieved correctly, put localhost(127.0.0.1) as the default.

host_ip=$(ip -4 -o addr show $INTFC | awk '{print $4}' | cut -d '/' -f1 | head -1)
if [[ $host_ip == "" ]]
then
    host_ip=127.0.0.1
fi

redis_port=6379

if [[ $DATABASE_TYPE == "dpudb" ]]; then
    host_ip="169.254.200.254"
    if ! ip -4 -o addr | awk '{print $4}' | grep $host_ip; then
        host_ip=127.0.0.1
    fi
    DPU_ID=`echo $DEV | tr -dc '0-9'`
    redis_port=`expr 6381 + $DPU_ID`
fi
```

Step 5: Restarting Redis

Restart Redis to apply the changes:

```
Docker restart database
```

```
admin@sonic:~$ docker restart database  
database
```

Step 6: See the configuration file again using the following command:

```
cat /var/run/redis/sonic-db/database_config.json
```


Redis Commander

Make sure that docker is started , then run the following command to start the Redis Commander:

```
docker run --name my-redis-commander -p 8081:8081 --restart always -d rediscommander/redis-commander:latest
```

```
kanza@kanza-Latitude-7410:~$ docker run --name my-redis-commander -p 8081:8081 --restart always -d rediscommander/redis-commander:latest
```

This will download and pull the latest redis commander image.

Now that the image is downloaded, a new container with the name my-redis-commander is created

You can view all the running container using the following command:

```
docker ps
```

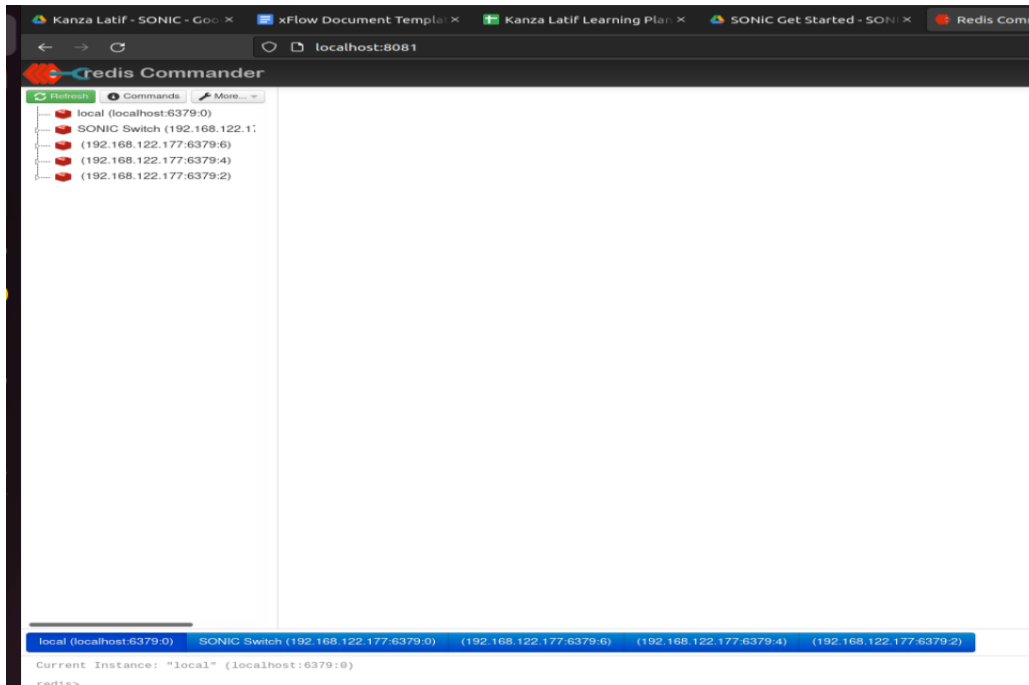
```
kanza@kanza-Latitude-7410:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND
CREATED	STATUS	PORTS
NAMES		
eba1c41b73a7	rediscommander/redis-commander:latest	"/usr/bin/dumb-init ..."
4 days ago	Up About an hour (healthy)	0.0.0.0:8081->8081/tcp, :::8081->8081/tcp
	my-redis-commander	

Redis Commander Web Dashboard

You can open Redis Commander on web using the following URL, depending on the port you assigned to it:

<http://localhost:8081>



Add a New Connection

To add a new connection, click on 'more'. You can see this option on the upper left side of the screen. Fill in the credentials. Give it a display name and port of your choice and press connect.

I have already added one connection here, so it looks like this:

The screenshot shows the Redis Commander web interface in a browser window. The address bar shows 'localhost:8081'. The interface has a sidebar on the left with a 'More...' button. The main area displays a list of Redis connections:

- local (localhost:6379:0)
- SONIC Switch (192.168.122.177:6379:0)
- (192.168.122.177:6379:6)
- (192.168.122.177:6379:4)
- (192.168.122.177:6379:2)

The 'SONIC Switch' connection is selected. Below the list, there are buttons for 'Add New Key...' and 'Disconnect'. A table shows the details of the selected instance:

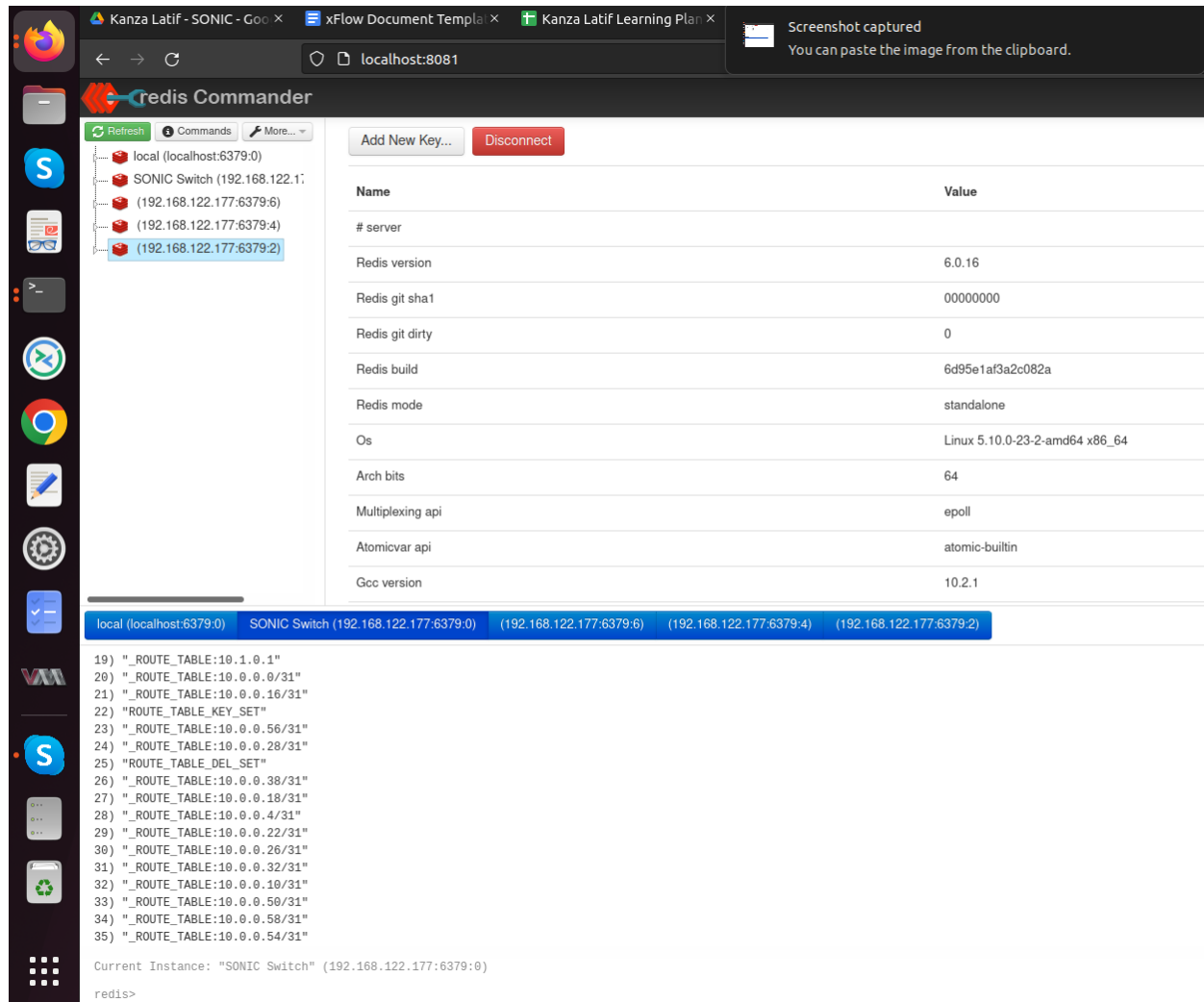
Name	Value
# server	
Redis version	6.0.16
Redis git sha1	00000000
Redis git dirty	0
Redis build	6d95e1af3a2c082a
Redis mode	standalone
Os	Linux 5.10.0-23-2-amd64 x86_64
Arch bits	64
Multiplexing api	epoll
Atomicvar api	atomic-builtin
Gcc version	10.2.1

At the bottom, the current instance is identified as 'SONIC Switch' (192.168.122.177:6379:0). The terminal shows the command 'redis> keys *|'.

You can view all the databases here on the left side. Just click on the one you want to view.

You can also run commands to view whatever you want to retrieve from the database. Click on the DB you want to run commands on. You can see in the image above.

I ran the command `'keys *'` to retrieve all the keys stored in this DB



The screenshot shows the Redis Commander web interface. The left sidebar lists several Redis instances, with the local instance (localhost:6379:0) selected. The main panel displays the Redis configuration page for the selected instance. The configuration table lists various Redis settings, including version, git sha1, build, mode, OS, architecture, and API versions. The bottom panel shows the output of the `'keys *'` command, listing 17 keys related to the `'_ROUTE_TABLE'`.

Name	Value
# server	
Redis version	6.0.16
Redis git sha1	00000000
Redis git dirty	0
Redis build	6d95e1af3a2c082a
Redis mode	standalone
Os	Linux 5.10.0-23-2-amd64 x86_64
Arch bits	64
Multiplexing api	epoll
Atomicvar api	atomic-builtin
Gcc version	10.2.1

Current Instance: "SONIC Switch" (192.168.122.177:6379:0)

```
redis> keys *
```

```
19) "_ROUTE_TABLE:10.1.0.1"
20) "_ROUTE_TABLE:10.0.0.0/31"
21) "_ROUTE_TABLE:10.0.0.16/31"
22) "ROUTE_TABLE_KEY_SET"
23) "_ROUTE_TABLE:10.0.0.56/31"
24) "_ROUTE_TABLE:10.0.0.28/31"
25) "ROUTE_TABLE_DEL_SET"
26) "_ROUTE_TABLE:10.0.0.38/31"
27) "_ROUTE_TABLE:10.0.0.18/31"
28) "_ROUTE_TABLE:10.0.0.4/31"
29) "_ROUTE_TABLE:10.0.0.22/31"
30) "_ROUTE_TABLE:10.0.0.26/31"
31) "_ROUTE_TABLE:10.0.0.32/31"
32) "_ROUTE_TABLE:10.0.0.10/31"
33) "_ROUTE_TABLE:10.0.0.50/31"
34) "_ROUTE_TABLE:10.0.0.58/31"
35) "_ROUTE_TABLE:10.0.0.54/31"
```