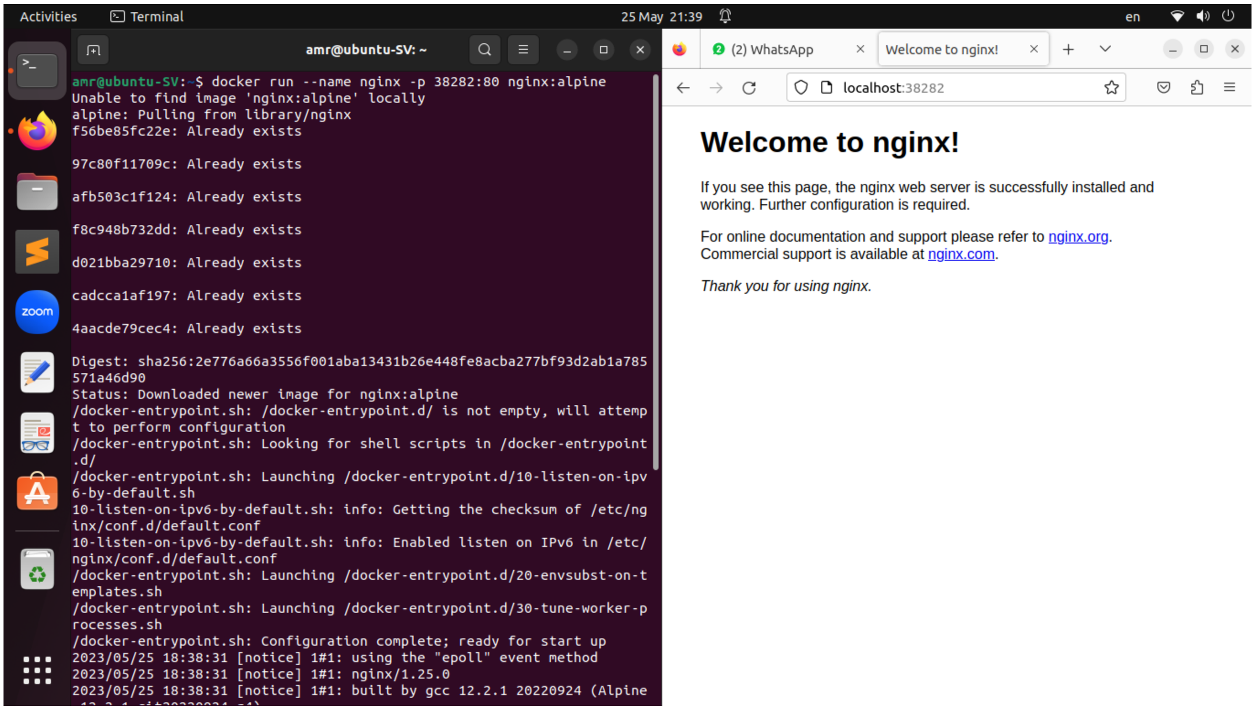
Docker Lab (2)

11- Run an instance of nginx:alpine with a name nginx and map port 80 on the container to 38282 on the host.



```
amr@ubuntu-SV:~$ docker run --name nginx -p 38282:80 nginx:alpine
Unable to find image 'nginx:alpine' locally
alpine: Pulling from library/nginx
f56be85fc22e: Already exists

97c80f11709c: Already exists

afb503c1f124: Already exists

f8c948b732dd: Already exists

d021bba29710: Already exists

cadcca1af197: Already exists

4aacde79cec4: Already exists

Digest: sha256:2e776a66a3556f001aba13431b26e448fe8acba277bf93d2ab1a785
571a46d90
Status: Downloaded newer image for nginx:alpine
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attemp
t to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint
.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv
6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/ng
inx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/
nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-t
emplates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-p
rocesses.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/05/25 18:38:31 [notice] 1#1: using the "epoll" event method
2023/05/25 18:38:31 [notice] 1#1: nginx/1.25.0
2023/05/25 18:38:31 [notice] 1#1: built by gcc 12.2.1 20220924 (Alpine
```

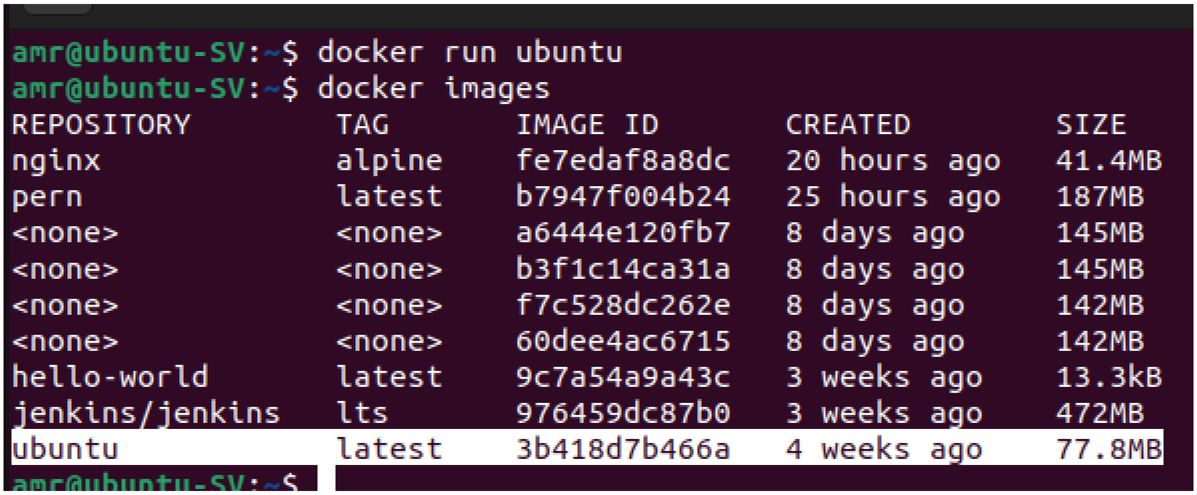
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

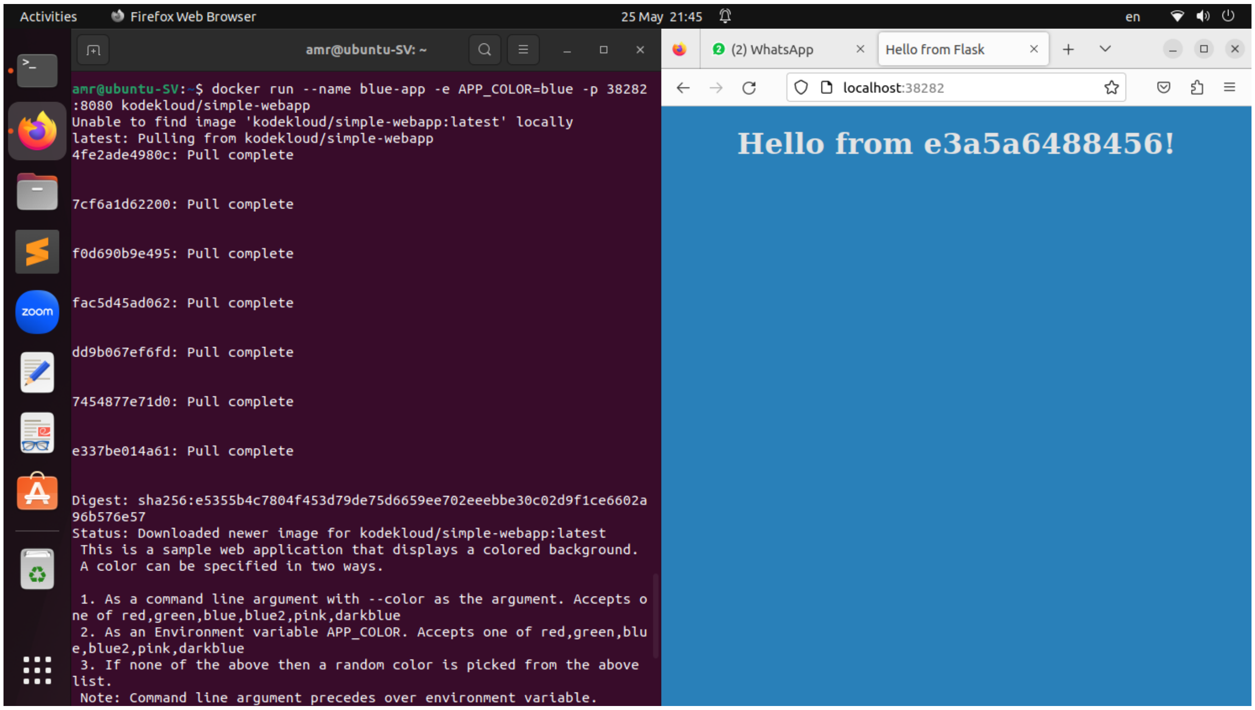
Thank you for using nginx.

12- create ubuntu image and check the size of it



```
amr@ubuntu-SV:~$ docker run ubuntu
amr@ubuntu-SV:~$ docker images
REPOSITORY        TAG       IMAGE ID       CREATED         SIZE
nginx             alpine    fe7edaf8a8dc   20 hours ago    41.4MB
pern              latest    b7947f004b24   25 hours ago    187MB
<none>            <none>    a6444e120fb7   8 days ago      145MB
<none>            <none>    b3f1c14ca31a   8 days ago      145MB
<none>            <none>    f7c528dc262e   8 days ago      142MB
<none>            <none>    60dee4ac6715   8 days ago      142MB
hello-world       latest    9c7a54a9a43c   3 weeks ago     13.3kB
jenkins/jenkins   lts       976459dc87b0   3 weeks ago     472MB
ubuntu            latest    3b418d7b466a   4 weeks ago     77.8MB
amr@ubuntu-SV:~$
```

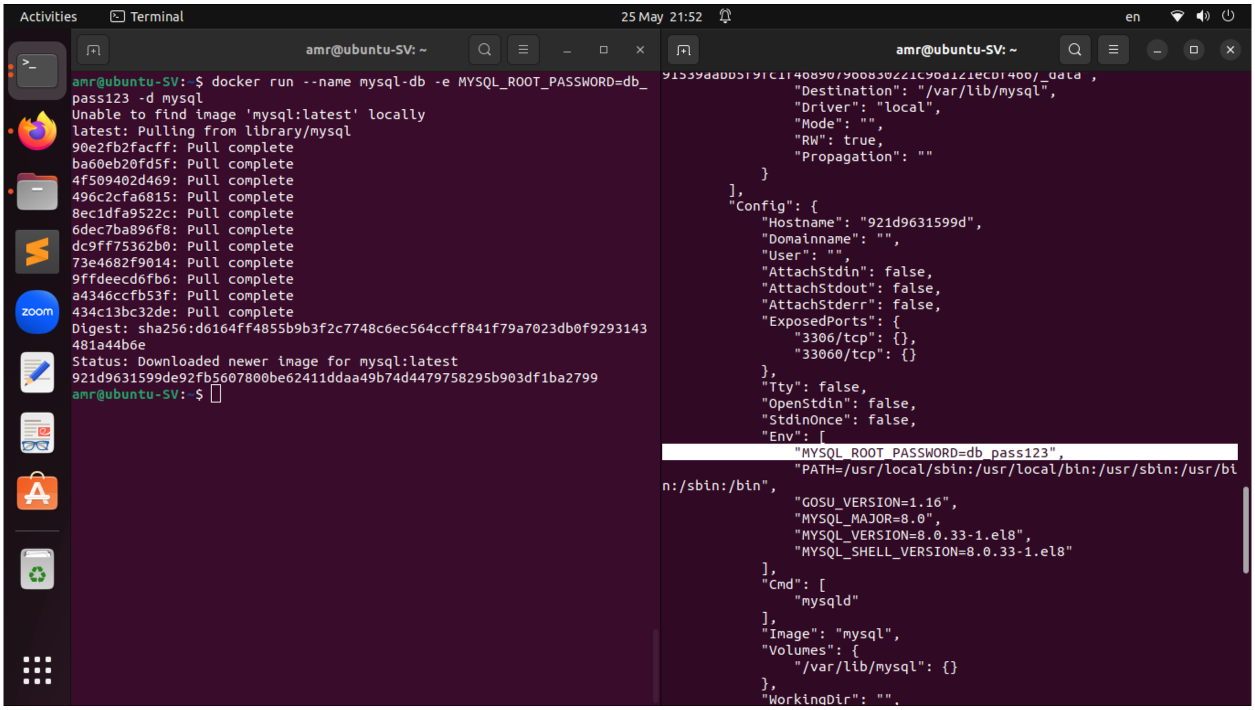
13- Run a container named blue-app using image kodekloud/simplewebapp and set the environment variable APP_COLOR to blue. Make the application available on port 38282 on the host. The application listens on port 8080.



```
amr@ubuntu-SV:~$ docker run --name blue-app -e APP_COLOR=blue -p 38282
:8080 kodekloud/simple-webapp
Unable to find image 'kodekloud/simple-webapp:latest' locally
latest: Pulling from kodekloud/simple-webapp
4fe2ade4980c: Pull complete


7cf6a1d62200: Pull complete


f0d690b9e495: Pull complete


fac5d45ad062: Pull complete


dd9b067ef6fd: Pull complete


7454877e71d0: Pull complete


e337be014a61: Pull complete


Digest: sha256:e5355b4c7804f453d79de75d6659ee702eeebbe30c02d9f1ce6602a
96b576e57
Status: Downloaded newer image for kodekloud/simple-webapp:latest
 This is a sample web application that displays a colored background.
 A color can be specified in two ways.

 1. As a command line argument with --color as the argument. Accepts o
ne of red,green,blue,blue2,pink,darkblue
 2. As an Environment variable APP_COLOR. Accepts one of red,green,blu
e,blue2,pink,darkblue
 3. If none of the above then a random color is picked from the above
list.
 Note: Command line argument precedes over environment variable.
```

Hello from e3a5a6488456!

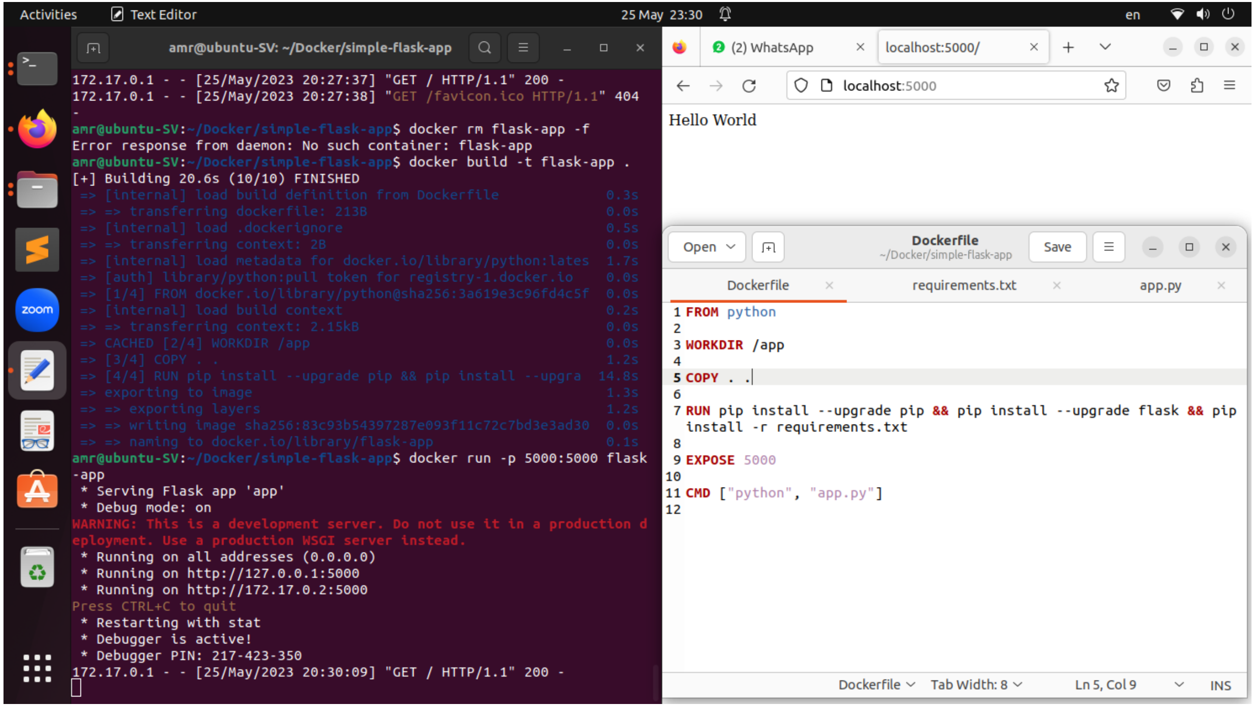
14- Deploy a mysql database using the mysql image and name it mysqldb Set the database password to use db_pass123 then inspect it to check the value



```
amr@ubuntu-SV:~$ docker run --name mysql-db -e MYSQL_ROOT_PASSWORD=db_
pass123 -d mysql
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
90e2fb2facff: Pull complete
ba60eb20fd5f: Pull complete
4f509402d469: Pull complete
496c2cfa6815: Pull complete
8ec1dfa9522c: Pull complete
6dec7ba896f8: Pull complete
dc9ff75362b0: Pull complete
73e4682f9014: Pull complete
9ffdeecd6fb6: Pull complete
a4346ccfb53f: Pull complete
434c13bc32de: Pull complete
Digest: sha256:d6164ff4855b9b3f2c7748c6ec564ccff841f79a7023db0f9293143
481a44b6e
Status: Downloaded newer image for mysql:latest
921d9631599de92fb5607800be62411ddaa49b74d4479758295b903df1ba2799
amr@ubuntu-SV:~$ 
```

```
91539aabb5f9fc1f468907968830221c96a121ecbf466/_data",
                "Destination": "/var/lib/mysql",
                "Driver": "local",
                "Mode": "",
                "RW": true,
                "Propagation": ""
            }
        ],
        "Config": {
            "Hostname": "921d9631599d",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "ExposedPorts": {
                "3306/tcp": {},
                "33060/tcp": {}
            },
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [
                "MYSQL_ROOT_PASSWORD=db_pass123",
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bi
n:/sbin:/bin",
                "GOSU_VERSION=1.16",
                "MYSQL_MAJOR=8.0",
                "MYSQL_VERSION=8.0.33-1.el8",
                "MYSQL_SHELL_VERSION=8.0.33-1.el8"
            ],
            "Cmd": [
                "mysqld"
            ],
            "Image": "mysql",
            "Volumes": {
                "/var/lib/mysql": {}
            },
            "WorkingDir": "",
```

15-pull the code from https://github.com/sabreensalama/simpleflask-app/tree/main and create a docker file for this flask app

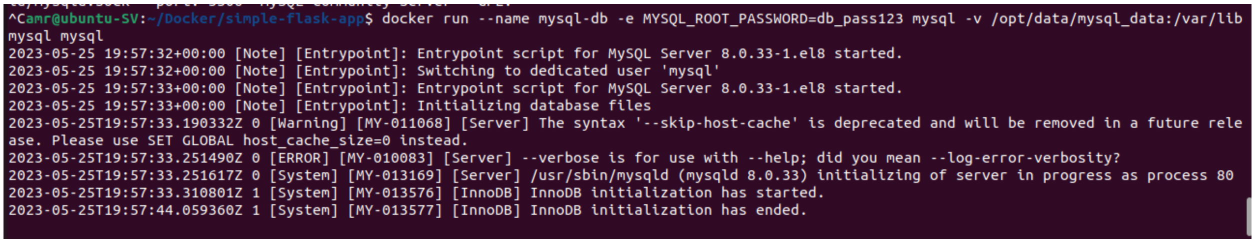


```
172.17.0.1 - - [25/May/2023 20:27:37] "GET / HTTP/1.1" 200 -
172.17.0.1 - - [25/May/2023 20:27:38] "GET /favicon.ico HTTP/1.1" 404
-
amr@ubuntu-SV:~/Docker/simple-flask-app$ docker rm flask-app -f
Error response from daemon: No such container: flask-app
amr@ubuntu-SV:~/Docker/simple-flask-app$ docker build -t flask-app .
[+] Building 20.6s (10/10) FINISHED
 => [internal] load build definition from Dockerfile                 0.3s
 => => transferring dockerfile: 213B                                 0.0s
 => [internal] load .dockerignore                                    0.5s
 => => transferring context: 2B                                      0.0s
 => [internal] load metadata for docker.io/library/python:lates      1.7s
 => [auth] library/python:pull token for registry-1.docker.io        0.0s
 => [1/4] FROM docker.io/library/python@sha256:3a619e3c96fd4c5f      0.0s
 => [internal] load build context                                    0.2s
 => => transferring context: 2.15kB                                  0.0s
 => CACHED [2/4] WORKDIR /app                                        0.0s
 => [3/4] COPY . .                                                   1.2s
 => [4/4] RUN pip install --upgrade pip && pip install --upgra      14.8s
 => exporting to image                                               1.3s
 => => exporting layers                                              1.2s
 => => writing image sha256:83c93b54397287e093f11c72c7bd3e3ad30      0.0s
 => => naming to docker.io/library/flask-app                         0.1s
amr@ubuntu-SV:~/Docker/simple-flask-app$ docker run -p 5000:5000 flask
-app
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production d
eployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.17.0.2:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 217-423-350
172.17.0.1 - - [25/May/2023 20:30:09] "GET / HTTP/1.1" 200 -
```

Hello World

```
FROM python

WORKDIR /app

COPY . .

RUN pip install --upgrade pip && pip install --upgrade flask && pip install -r requirements.txt

EXPOSE 5000

CMD ["python", "app.py"]
```

16- Create a volume called mysql_data, Run a mysql container again, but this time map a volume to the container so that the data stored by the container is stored at /opt/data on the host.
Use the same name : mysql-db and same password: db_pass123 as before. Mysql stores data at /var/lib/mysql inside the container.



```
^Camr@ubuntu-SV:~/Docker/simple-flask-app$ docker run --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 mysql -v /opt/data/mysql_data:/var/lib
mysql mysql
2023-05-25 19:57:32+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.33-1.el8 started.
2023-05-25 19:57:32+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2023-05-25 19:57:33+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.33-1.el8 started.
2023-05-25 19:57:33+00:00 [Note] [Entrypoint]: Initializing database files
2023-05-25T19:57:33.190332Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future rele
ase. Please use SET GLOBAL host_cache_size=0 instead.
2023-05-25T19:57:33.251490Z 0 [ERROR] [MY-010083] [Server] --verbose is for use with --help; did you mean --log-error-verbosity?
2023-05-25T19:57:33.251617Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.0.33) initializing of server in progress as process 80
2023-05-25T19:57:33.310801Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2023-05-25T19:57:44.059360Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
```