

UAT Deployment Manual

This deployment manual provides a detailed, step-by-step guide to successfully deploy the Revolutio platform on the UAT environment. Each step has been carefully outlined to ensure a smooth and error-free deployment process. Please follow the instructions meticulously, paying close attention to details such as file paths, commands, and configurations.

By adhering to this guide, you will ensure that the platform is set up correctly and ready for use in the UAT environment.

Let's begin the deployment process.

Step 1: Pull the latest Docker Images from ACR – revolutio.azurecr.io

1. Open the VS code and open the Development branch
2. Run the command to pull the latest image from repository
docker compose -f docker-compose-onprem.yml pull (On-prem Deployment)
docker compose -f docker-compose-onprem-single-server.yml pull (Single Server Deployment)
3. Display a list of all available Docker images.

docker images

```
revolutio-uat-app@revolutio-uat-app:~/Revolutio$ docker images
```

| REPOSITORY | TAG | IMAGE ID | CREATED | SIZE |
|---|---------------|--------------|---------------|--------|
| revolutio.azurecr.io/revolutio_kubernetes/rqscheduler | latest | 1182af6af343 | 17 hours ago | 352MB |
| revolutio.azurecr.io/revolutio_kubernetes/nginx | latest | 7e03287bb4fb | 17 hours ago | 48MB |
| revolutio.azurecr.io/revolutio_kubernetes/revolutio | latest | 23de6fc73796 | 17 hours ago | 5.6GB |
| revolutio.azurecr.io/revolutio/arrowflight | latest | bcf9300ee511 | 18 hours ago | 465MB |
| revolutio.azurecr.io/revolutio_kubernetes/postgres | 15-bullseye | a14888aed3f2 | 18 hours ago | 425MB |
| revolutio.azurecr.io/revolutio_kubernetes/redis | 5.0.4-stretch | 72fdc8acal72 | 18 hours ago | 116MB |
| revolutio.azurecr.io/revolutio_kubernetes/keydb | latest | 3a37f9a9d125 | 15 months ago | 65.2MB |
| revolutiodev.azurecr.io/revolutio_kubernetes/keydb | latest | 3a37f9a9d125 | 15 months ago | 65.2MB |

Step 2: Save Docker Images

Save each Docker image individually using the following commands:

- ***docker save revolutio.azurecr.io/revolutio_kubernetes/rqscheduler:latest | pigz --fast > rqscheduler_v1beta.1001.tar.gz***
- ***docker save revolutio.azurecr.io/revolutio_kubernetes/revolutio:latest | pigz --fast > revolutio_v1beta.1001.tar.gz***
- ***docker save revolutio.azurecr.io/revolutio_kubernetes/nginx:latest | pigz --fast > nginx_v1beta.1001.tar.gz***
- ***docker save revolutio.azurecr.io/revolutio/arrowflight | pigz --fast > arrowflight_v1beta.1001.tar.gz***
- ***docker save revolutio.azurecr.io/revolutio_kubernetes/postgres:15-bullseye | pigz --fast > postgres_v1beta.1001.tar.gz***

- `docker save revolutio.azurecr.io/revolutio_kubernetes/redis:5.0.4-stretch | pigz --fast > redis_v1beta.1001.tar.gz`
- `docker save revolutio.azurecr.io/revolutio_kubernetes/keydb:latest | pigz --fast > keydb_v1beta.1001.tar.gz`
- `docker save edoburu/pgbouncer:1.15.0 | pigz --fast > pgbouncer_v1beta.1001.tar.gz`

Step 3: Transfer Docker images to the UAT App server

1. Transfer the saved Docker images to UAT app server using the External SFTP

Step 4: Place the Configuration Files and Folder

4.1 From the **Revolutio Repository in VS Code** copy the following folders and files to your local machine:

- **nginx** folder
It should have the **nginx_dev.conf** file
- **pg_bouncer** folder
It should have the **pg_hba.conf** file
- **postgres** folder
It should have the **postgresql.conf** file
- **docker-compose-onprem.yml** file
- **.env** (This is a hidden file)
- **.env_UAT** (This is a hidden file)
- **revolutio.conf** file

4.2 In the **Revolutio** folder, go to the **Platform_Configs** folder, and copy the following files to the local machine:

- **allowed_host.txt**
- **allowed_domains.json**
- **app_database_mapping.json**
- **tenant_database_mapping.json**
- **tenant_host_mapping.json**
- **user_databases.json**
- **user_databases_updatation.json**

Step 5: Set Up the Revolutio Directory on UAT App server

Create a directory named **Revolutio** on the destination on UAT App server

mkdir Revolutio

Inside the **Revolutio** folder on the destination path of UAT App server, place the above-mentioned files and folders as per **point 4.1**

Inside the **Revolutio** folder on the destination path of UAT App server, create a folder named **Platform_Configs** and place the above-mentioned files as per **point 4.2**

Please run the command to the Platform Config folder

chmod -R 777 Platform_Configs/

Step 6: Load Docker Images on UAT App Server

Load all the images one by one using the following commands:

(Each docker image will take approximately 2 minutes to load)

```
azureuser@test-vm:~/Revolutio$ docker load < revolutio_v1beta.1001.tar.gz
a483da8ab3e9: Loading layer [=====>] 77.83MB/77.83MB
c8f253aef560: Loading layer [=====>] 9.552MB/9.552MB
9559125c0303: Loading layer [=====>] 35.34MB/35.34MB
010a290fc3d5: Loading layer [=====>] 5.12kB/5.12kB
067693c93cf9: Loading layer [=====>] 13.09MB/13.09MB
165fa78c0c3f: Loading layer [=====>] 140.7MB/140.7MB
b7f64fbbb409: Loading layer [=====>] 1.621GB/1.621GB
6907d2be77b4: Loading layer [=====>] 26.54MB/26.54MB
d6e09dd16db3: Loading layer [=====>] 9.216kB/9.216kB
7a29ca8320a6: Loading layer [=====>] 11.78kB/11.78kB
cd3decc93bbe: Loading layer [=====>] 17.41kB/17.41kB
4686963dcd2b: Loading layer [=====>] 17.41kB/17.41kB
e3b3e041238c: Loading layer [=====>] 9.156MB/9.156MB
36e9f14bf9a9: Loading layer [=====>] 316.8MB/316.8MB
c40dcb5eda33: Loading layer [=====>] 3.128GB/3.128GB
4aa9d69b66a1: Loading layer [=====>] 326.9MB/326.9MB
73bc27ddd1e1: Loading layer [=====>] 15.36kB/15.36kB
43e466a25e05: Loading layer [=====>] 15.36kB/15.36kB
a80fd69f9578: Loading layer [=====>] 11.14MB/11.14MB
5f70bf18a086: Loading layer [=====>] 1.024kB/1.024kB
a580eada36c5: Loading layer [=====>] 6.656kB/6.656kB
Loaded image: revolutio.azurecr.io/revolutio_kubernetes/revolutio:latest
```

- **docker load < revolutio_v1beta.1001.tar.gz**
- **docker load < rqscheduler_v1beta.1001.tar.gz**
- **docker load < nginx_v1beta.1001.tar.gz**
- **docker load < arrowflight_v1beta.1001.tar.gz**
- **docker load < postgres_v1beta.1001.tar.gz**
- **docker load < redis_v1beta.1001.tar.gz**
- **docker load < keydb_v1beta.1001.tar.gz**
- **docker load < pgbouncer_v1beta.1001.tar.gz**

Step 7: Modify the Docker Compose File

Now in the **Revolutio** folder on the destination, which is the UAT App server, access the docker file using the following command:

vi docker-compose-onprem.yml

In the file, we need to comment out two lines from the **postgres-new** service.

*(These lines are under the **volume** section)*

- **./postgres/postgresql.conf:/var/lib/postgresql/data/postgresql.conf**
- **./pgbouncer/pg_hba.conf:/var/lib/postgresql/data/pg_hba.conf**

PRESS THE “i” KEY ON YOUR KEYBOARD, PRESS THE DOWN KEY TO NAVIGATE TO THE LINES WHICH NEED TO BE COMMENTED.

```
postgres-new:
  image: ${REGISTRY_HOST}/revolutio_kubernetes/postgres:15-bullseye
  build:
    context: "./postgres"
    dockerfile: ./Dockerfile_update
  restart: always
  environment:
    - POSTGRES_DB=Platform_DB
    - POSTGRES_USER=revolutio
    - POSTGRES_PASSWORD=supersecretpassword
    - ALLOW_EMPTY_PASSWORD=yes
  volumes:
    - postgresql-data-new:/var/lib/postgresql/data
    - postgresql-backup-new:/backups
    #- ./postgres/postgresql.conf:/var/lib/postgresql/data/postgresql.conf
    #- ./pgbouncer/pg_hba.conf:/var/lib/postgresql/data/pg_hba.conf
  # ports:
  #   - "5432:5432"
```

AFTER COMMENTING, PRESS THE “ESC” KEY AND TYPE **:wq!** TO SAVE THE FILE

Save the docker file and cross check and confirm once again if the 2 lines are commented.

Step 8: Update Configuration Files

Now in the **Revolutio** folder, go to the **Platform_Configs** folder:

We need to add the details of UATuaction domain url in the following files:

- **allowed_host.txt**
- **allowed_domains.json**
- **tenant_database_mapping.json**
- **tenant_host_mapping.json**

Use the **vi allowed_host.txt** command to access the file.

In the **allowed_host.txt** file

PRESS THE “i” KEY. REMOVE ALL THE EXISTING TEXT AND ADD THE UAT URL, MAKE SURE NOT TO MISS THE COMMA IN THE END.

```
abc.com,
~
~
```

AFTER ADDING, PRESS THE “ESC” KEY AND TYPE **:wq!** TO SAVE THE FILE

Use the **vi allowed_domains.json** command to access the file.

In the **allowed_domains.json** file

PRESS THE “i” KEY. REMOVE ALL THE EXISTING TEXT AND ADD THE UAT URL.

```
["https://abc.com"]
```

AFTER ADDING, PRESS THE “ESC” KEY AND TYPE **:wq!** TO SAVE THE FILE

Use the **vi tenant_database_mapping.json** command to access the file.

In the **tenant_database_mapping.json** file

PRESS THE “I” KEY. REMOVE ALL THE EXISTING URL AND ADD THE UAT URL.

```
{  
  "abc.com": "public",
```

AFTER ADDING, PRESS THE “ESC” KEY AND TYPE **:wq!** TO SAVE THE FILE

Use the **vi tenant_host_mapping.json** command to access the file.

In the **tenant_host_mapping.json** file

PRESS THE “I” KEY. REMOVE ALL THE EXISTING URL AND ADD THE UAT URL.

```
{  
  "public": {  
    "urlhost": [  
      "abc.com"
```

AFTER ADDING, PRESS THE “ESC” KEY AND TYPE **:wq!** TO SAVE THE FILE

Step 9: Create and Configure the Uploads Directory

Now, inside the **Platform_Configs** folder, create a new directory named “**Uploaded_Application_Backups**” and assign necessary permissions.

(Use the following commands)

mkdir Uploaded_Application_Backups

chmod -R 777 Uploaded_Application_Backups

Step 10: Update Nginx Configuration

Navigate to the nginx folder and edit the **nginx_dev.conf** file:

Use the **vi nginx_dev.conf** command to access the file.

In the **nginx_dev.conf** file

SCROLL DOWN TO THE **server** SECTION, PRESS THE “I” KEY. REMOVE ALL THE EXISTING URL AND ADD THE UAT URL.

```
server
{
    if ($blockedagent)
    {
        return 403;
    }

    # Put in all the allowed hosts separated by |
    if ( $host !~* ^(abc.com)$ ) {
        return 403;
    }
    if ( $http_host !~* ^(abc.com)$ ) {
        return 403;
    }
    listen 443 ssl;
```

AFTER ADDING, PRESS THE “ESC” KEY AND TYPE **:wq!** TO SAVE THE FILE

Now, go back to the Revolutio folder and access the docker file using the following command:

vi docker-compose-onprem.yml

Scroll down to the **nginx** service, check the last 2 lines under the **volumes** section and uncomment it.

```
#- "${DOCKER_SSL_CERT:-./nginx/ssl.crt:/etc/ssl/ssl.crt}"
```

```
#- "${DOCKER_SSL_KEY:-./nginx/ssl.key:/etc/ssl/ssl.key}"
```

Also, the name of the SSL Certificate File and Key should match with the names in the docker file.

(MAKE SURE TO NAME THE SSL CERTIFICATE AS “ssl.crt” & KEY AS “ssl.key” & PLACE THE SSL CERTIFICATE FILE AND KEY INSIDE THE nginx FOLDER)

Run the following command to assign full permissions:

chmod 777 ssl.crt ssl.key

```
nginx:
  image: ${REGISTRY_HOST}/revolutio_kubernetes/nginx:latest
  build: ./nginx
  command: nginx -g "daemon off;"
  ports:
    - 8080:8080
  restart: on-failure:5
  read_only: true
  tmpfs:
    - /run:mode=770,size=1k,uid=101,gid=101
    - /var/cache/nginx:mode=770,size=1k,uid=101,gid=101
  depends_on:
    - web
  healthcheck:
    test: ["CMD", "curl", "--fail", "--insecure", "http://localhost/"]
    interval: 20s
    timeout: 20s
    retries: 5
  volumes:
    - "${DOCKER_NGINX_VOLUME}:/nginx/nginx_dev.conf:/etc/nginx/nginx.conf"
    - static:/var/www/revolutio/static
    - media:/var/www/revolutio/media
    #- "${DOCKER_SSL_CERT}:/nginx/ssl.crt:/etc/ssl/ssl.crt"
    #- "${DOCKER_SSL_KEY}:/nginx/ssl.key:/etc/ssl/ssl.key"
```

Step 11: Start the Application Server

Start the application server on Burgan UAT using the following command: (Make sure to run this command in the **Revolutio** folder)

docker compose -f docker-compose-onprem.yml up

Wait for 2 minutes until you see the following logs:

```
revolutio tls=no
pgbouncer-new-1 | 2025-02-13 12:11:37.550 UTC [1] LOG C-0x74939f76a310: Platform_DB/revolutio@172.18.0.9:46538 closing because: pgbouncer cannot g
connect to server (age=0s)
pgbouncer-new-1 | 2025-02-13 12:11:37.550 UTC [1] WARNING C-0x74939f76a310: Platform_DB/revolutio@172.18.0.9:46538 pooler error: pgbouncer cannot
connect to server
worker-scheduler-system-1 | Waiting for PostgreSQL to become available...
pgbouncer-new-1 | 2025-02-13 12:11:37.640 UTC [1] LOG C-0x74939f76a310: Platform_DB/revolutio@172.18.0.12:38090 login attempt: db=Platform_DB user=
pgbouncer-new-1 | 2025-02-13 12:11:37.640 UTC [1] LOG C-0x74939f76a310: Platform_DB/revolutio@172.18.0.12:38090 closing because: pgbouncer cannot
connect to server (age=0s)
pgbouncer-new-1 | 2025-02-13 12:11:37.640 UTC [1] WARNING C-0x74939f76a310: Platform_DB/revolutio@172.18.0.12:38090 pooler error: pgbouncer cannot
connect to server
worker-computation-0-1 | Waiting for PostgreSQL to become available...
pgbouncer-new-1 | 2025-02-13 12:11:37.998 UTC [1] LOG C-0x74939f76a310: Platform_DB/revolutio@172.18.0.10:57244 login attempt: db=Platform_DB user=
pgbouncer-new-1 | 2025-02-13 12:11:37.998 UTC [1] LOG C-0x74939f76a310: Platform_DB/revolutio@172.18.0.10:57244 closing because: pgbouncer cannot
connect to server (age=0s)
pgbouncer-new-1 | 2025-02-13 12:11:37.998 UTC [1] WARNING C-0x74939f76a310: Platform_DB/revolutio@172.18.0.10:57244 pooler error: pgbouncer cannot
connect to server
worker-scheduler-1 | Waiting for PostgreSQL to become available...
pgbouncer-new-1 | 2025-02-13 12:11:38.476 UTC [1] LOG C-0x74939f76a310: Platform_DB/revolutio@172.18.0.13:33066 login attempt: db=Platform_DB user=
pgbouncer-new-1 | 2025-02-13 12:11:38.476 UTC [1] LOG C-0x74939f76a310: Platform_DB/revolutio@172.18.0.13:33066 closing because: pgbouncer cannot
connect to server (age=0s)
pgbouncer-new-1 | 2025-02-13 12:11:38.476 UTC [1] WARNING C-0x74939f76a310: Platform_DB/revolutio@172.18.0.13:33066 pooler error: pgbouncer cannot
connect to server
worker-data-upload-0-1 | Waiting for PostgreSQL to become available...
pgbouncer-new-1 | 2025-02-13 12:11:38.624 UTC [1] LOG C-0x74939f76a310: Platform_DB/revolutio@172.18.0.9:46544 login attempt: db=Platform_DB user=
revolutio tls=no
pgbouncer-new-1 | 2025-02-13 12:11:38.624 UTC [1] LOG C-0x74939f76a310: Platform_DB/revolutio@172.18.0.9:46544 closing because: pgbouncer cannot g
connect to server (age=0s)
pgbouncer-new-1 | 2025-02-13 12:11:38.624 UTC [1] WARNING C-0x74939f76a310: Platform_DB/revolutio@172.18.0.9:46544 pooler error: pgbouncer cannot
connect to server
worker-scheduler-system-1 | Waiting for PostgreSQL to become available...
pgbouncer-new-1 | 2025-02-13 12:11:38.709 UTC [1] LOG C-0x74939f76a310: Platform_DB/revolutio@172.18.0.12:38106 login attempt: db=Platform_DB user=
pgbouncer-new-1 | 2025-02-13 12:11:38.709 UTC [1] LOG C-0x74939f76a310: Platform_DB/revolutio@172.18.0.12:38106 closing because: pgbouncer cannot
connect to server (age=0s)
pgbouncer-new-1 | 2025-02-13 12:11:38.709 UTC [1] WARNING C-0x74939f76a310: Platform_DB/revolutio@172.18.0.12:38106 pooler error: pgbouncer cannot
connect to server
worker-computation-0-1 | Waiting for PostgreSQL to become available...
```

You should see the line **"Waiting for PostgreSQL to become available..."**

PRESS "CTRL+C" KEY TO STOP THE LOGS.

Now, access the Docker file again using the **vi docker-compose-onprem.yml** command.

We have to UNCOMMENT the same two lines from the **postgres-new** service.

(These lines are under the **volume** section)

```
- ./postgres/postgresql.conf:/var/lib/postgresql/data/postgresql.conf
- ./pgbouncer/pg_hba.conf:/var/lib/postgresql/data/pg_hba.conf
```

PRESS THE “i” KEY ON YOUR KEYBOARD, PRESS THE DOWN KEY TO NAVIGATE TO THE LINES WHICH NEED TO BE “UNCOMMENTED”.

```
postgres-new:
  image: ${REGISTRY_HOST}/revolutio_kubernetes/postgres:15-bullseye
  build:
    context: "./postgres"
    dockerfile: ./Dockerfile_update
  restart: always
  environment:
    - POSTGRES_DB=Platform_DB
    - POSTGRES_USER=revolutio
    - POSTGRES_PASSWORD=supersecretpassword
    - ALLOW_EMPTY_PASSWORD=yes
  volumes:
    - postgresql-data-new:/var/lib/postgresql/data
    - postgresql-backup-new:/backups
    - ./postgres/postgresql.conf:/var/lib/postgresql/data/postgresql.conf
    - ./pgbouncer/pg_hba.conf:/var/lib/postgresql/data/pg_hba.conf
  ports:
    - "5432:5432"
```

AFTER COMMENTING, PRESS THE “ESC” KEY AND TYPE **:wq!** TO SAVE THE FILE

Step 12: Finalize Deployment

Now start the application server in the UAT App server using the following command:

(Make sure to run this command in the **Revolutio** folder)

docker compose -f docker-compose-onprem.yml up -d

Also run the command as below

```
[+] Running 15/15
✓ Container revolutio-redis-scheduler-1      Started
✓ Container revolutio-keydb-1                Started
✓ Container revolutio-arrowflight-1          Started
✓ Container revolutio-redis-1                Started
✓ Container revolutio-pgbouncer1-new-1       Started
✓ Container revolutio-postgres-new-1         Started
✓ Container revolutio-web-1                  Started
✓ Container revolutio-worker-app-migration-1 Started
✓ Container revolutio-worker-1               Started
✓ Container revolutio-nginx-1                Started
✓ Container revolutio-worker-scheduler-1     Started
✓ Container revolutio-worker-scheduler-system-1 Started
✓ Container revolutio-rqscheduler-1          Started
✓ Container revolutio-worker-data-upload-0-1 Started
✓ Container revolutio-worker-computation-0-1 Started
azureuser@test-vm:~/Revolutio$
```


Once all the services are started, run the following command to check the logs:

`docker compose -f docker-compose-onprem.yml logs -f`

Allow the logs run for 5 minutes until you see a SUCCESS message and default migration ran as shown below:

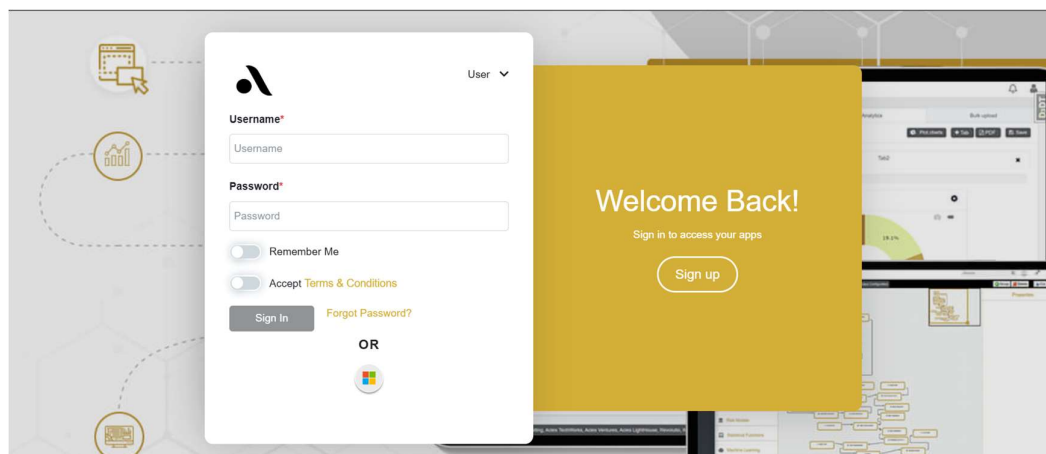
```
web-1 | CATEGORY CATEGORY_WORD
web-1 | LITERAL 46
web-1 | LITERAL 64
web-1 | LITERAL 43
web-1 | LITERAL 45
web-1 | AT AT_END
web-1 |
web-1 | 0: INFO 4 0b0 1 MAXREPEAT (to 5)
web-1 | 5: AT BEGINNING LINE
web-1 | 7: REPEAT ONE 15 1 MAXREPEAT (to 26)
web-1 | 11: IN 13 (to 25)
web-1 | 13: CHARSET [0x00000000, 0x00006800, 0x00000001, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000]
web-1 | 22: CATEGORY UNI_WORD
web-1 | 24: FAILURE
web-1 | 25: SUCCESS
web-1 | 26: AT END LINE
web-1 | 28: SUCCESS
web-1 | AT AT BEGINNING
web-1 | MAX REPEAT 1 MAXREPEAT
web-1 | IN
web-1 | CATEGORY CATEGORY_WORD
web-1 | LITERAL 46
web-1 | LITERAL 64
web-1 | LITERAL 43
web-1 | LITERAL 45
web-1 | AT AT_END
web-1 |
web-1 | 0: INFO 4 0b0 1 MAXREPEAT (to 5)
web-1 | 5: AT BEGINNING LINE
web-1 | 7: REPEAT ONE 15 1 MAXREPEAT (to 26)
web-1 | 11: IN 13 (to 25)
web-1 | 13: CHARSET [0x00000000, 0x00006800, 0x00000001, 0x00000000, 0x00000000, 0x00000000, 0x00000000, 0x00000000]
web-1 | 22: CATEGORY UNI_WORD
web-1 | 24: FAILURE
web-1 | 25: SUCCESS
web-1 | 26: AT END LINE
web-1 | 28: SUCCESS
web-1 | INFO: Started server process [1]
web-1 | INFO: Waiting for application startup.
web-1 | INFO: ASGI 'lifespan' protocol appears unsupported.
web-1 | INFO: Application startup complete.
web-1 | INFO: Uvicorn running on http://0.0.0.0:5000 (Press CTRL+C to quit)
```

Once you see this screen, you can stop the logs using “CTRL+C”

Step 14: Front \end Password Change

Now, access the URL in the browser, you will see the following screen:

`https://<domainurl>`



The user IDs are `revolutio_admin` & `revolutio_admin2`

Default password for the user IDs is `Revadmin@2023`

When you try to login with the default password, you will be prompted to change your password.

Command for restarting Application services

(Run the command inside of Revolutio folder)

```
docker compose -f docker-compose-onprem.yml down
```

```
docker compose -f docker-compose-onprem.yml up -d
```

Command for restarting Docker services

```
sudo systemctl restart docker
```

Command for checking the application logs

(Run the command inside of Revolutio folder)

```
docker compose -f docker-compose-onprem.yml logs -f
```