MVP Server Installation and Configuration

NOTE: NEED TO DISABLE ALL "runtime: nvidia" FOR LOCAL DEV DEPLOYMENT/INSTALL!

1. Prerequisites: Install Essential Software

Before starting the application setup, ensure you have the following software installed on your machine:

- **Operating System:** These instructions are compatible with Linux, macOS, or Windows.
- **Git:** This is required to clone the three application repositories.
    - Download and install Git from the official website:
        - https://git-scm.com/downloads
- **Docker Desktop:** This package includes **Docker Engine** (the containerization platform) and **Docker Compose** (the tool for running multi-container applications). The entire setup relies on these tools.
    - Download and install Docker Desktop for your OS. **Ensure Docker is running** before proceeding (e.g., see the green indicator in the Docker Desktop application):
        - https://www.docker.com/products/docker-desktop/
- **Terminal/Command Line Interface (CLI):** You'll need to be comfortable using your OS's command line to execute the setup steps.

---

2. Prepare the Workspace

This step ensures all necessary code is downloaded and organized.

2.1 Clone the Repositories

Use Git to download the application code. You should do this in a single, dedicated folder (e.g., `mvp-project`).

Repository Name	Command
`mvp-model-server`	`git clone https://github.com/surgivance/mvp-model-server`
`mvp-backend`	`git clone https://github.com/surgivance/mvp-backend`
`mvp-frontend`	`git clone https://github.com/surgivance/mvp-frontend`

2.2 Create the Docker Network

This isolates the application's components so they can communicate securely.

1. Open your terminal.
2. Run the network creation command:

```Shell
docker network create mvp-net
```

3. This creates the `mvp-net` bridge network.

---

3. Launch the Application Components

Follow the original `README` steps, adding conceptual clarity and explicitly instructing navigation. **You must open a new terminal window/tab for each component you start, as they run continuously.**

3.1 Start the Model Server (Terminal 1)

1. Navigate to the `mvp-model-server` directory:

```Shell
cd mvp-model-server
```

2. Start the server. This builds the container image and runs the service:

```Shell
docker compose up --build server
```

3. This terminal should be left open and running.

3.2 Configure and Start MinIO (Terminal 2)

This service acts as the **remote file storage** for the application.

1. Open a **new terminal** and navigate to the `mvp-backend` directory:

Shell

```
cd mvp-backend
```

2. **Edit `docker-compose.yml`**: Before running, you must set a secure password.
   - Open the `docker-compose.yml` file in a text editor.
   - Find the `minio` service and change the value for `MINIO_ROOT_PASSWORD` to a secure, new password (e.g., `MyMinioSecureP@ss`). **Remember this password!**
3. Start the MinIO service:

Shell

```
docker compose up minio
```

3.3 (***Domain Deploy Only!***) - Set Up MinIO Access Keys (Terminal 2 - Cont.)

The main backend application needs a secure key pair to talk to the MinIO storage.

1. Open a web browser and navigate to `http://lab.surgivance.com:8004/`.
2. **Log in** with the username `surgivance` and the secure `MINIO_ROOT_PASSWORD` you just set.
3. Click the "Access Keys" tab.
4. Click "Create access key" and then "Create".
5. **Crucial Step:** Copy the generated **"Access Key"** and **"Secret Key"**.

3.3 (***Local Deploy Only!***) Set Up MinIO Access Keys Command Line Method

The main backend application needs a secure key pair (Access Key and Secret Key) to communicate with the MinIO storage service. We will use the command line to generate these keys, bypassing the web browser.

1. Open terminal in the Docker Desktop UI for the `mvp-server` container.
2. Configure the MinIO client (`mc`) with the root credentials (`surgivance` and the password you set) to manage the local MinIO server. The server is addressed as `minio:9000` within the Docker network.

Shell

```
mc alias set myminio http://minio:9000 surgivance [MINIO_ROOT_PASSWORD]
```

3. ***Be sure and replace `[MINIO_ROOT_PASSWORD]` with the secure password you set in Section 3.2!***
4. Create a new key pair for the `surgivance` root user. This command outputs the two critical strings needed for the next step.
   Bash

Shell

```
mc admin accesskey create myminio surgivance
```

5. **Crucial Step: Copy the Keys!** The output will look similar to this:

None

```
     Access Key: 9P4D3KH8W2R7F0Z1XKCD
     Secret Key: sL0zXbA4mUf+o2yR3vN8c9Q5pI7wD+gJj0H1tKeF
     Expiration: NONE
     Name:
     Description:
```

3.4 Configure and Start the Backend App (Terminal 3)

1. Open a **new terminal** and navigate back to the.
2. **Edit `docker-compose.yml` in the** `mvp-backend` **directory (Again!)**: Find the `app` service's `environment` section.
3. You must paste the keys and set the database password.
   - Find the `app` service's `environment` section.
   - Paste the keys you copied:

None

```
environment:
  REMOTE_STORAGE_ACCESS_KEY: [PASTE YOUR ACCESS KEY HERE]
  REMOTE_STORAGE_SECRET_KEY: [PASTE YOUR SECRET KEY HERE]
```

   - Set a secure password for the PostgreSQL database (`db` service). **It must match in two places:**
      - The `db` service's `environment: POSTGRES_PASSWORD` field.
      - The `app` service's `environment: DATABASE_URL` field (just the password part of the URL).
4. Start the backend application:

Shell
```
docker compose up --build app
```

3.5 Initialize the Database (Terminal 4)

This step runs a script *inside* the newly created `app` container to prepare the database tables.

1. Open a **new terminal** and navigate to the `mvp-backend` directory.
2. First run the “holding” command:

Shell
```
docker compose up -d app
```

3. Next, run the initialization command:

Shell
```
docker compose exec app python -m src.models.db_reset
```

4. *This command will reset all data if run again later.*

3.6 Start the Job Queue Worker (Terminal 5)

This component handles background tasks and processes for the application.

1. Open a **new terminal** and navigate to the `mvp-backend` directory.
2. Run the worker:

Shell
```
docker compose up --build job-queue-worker
```

3.7 Start the Frontend (Terminal 6)

1. Open a **new terminal** and navigate to the `mvp-frontend` directory:

Shell

```shell
cd mvp-frontend
```

2. *(Optional)* Edit the `docker-compose.yml` here to set a `BASIC_AUTH_USERNAME` and `BASIC_AUTH_PASSWORD` for secure access.
3. Start the frontend:

Shell

```shell
docker compose up --build
```

---

4. Final Access

You can now access the running application:

- If deploying to the Surgivance domain, open your web browser to: `http://lab.surgivance.com:8000/`.
- If deploying to a local development installation, open your web browser to: `http://localhost:8000/`.