Deployment Manual for Scrapy Project and API on DigitalOcean

This manual provides instructions to deploy the Scrapy project and API with my Docker images or alternatively with custom images, database and VPN server.

Initial Setup

1. Sign up
2. Create a new project
3. Skip Moving Resources if prompted.

Database Setup

1. Create a Database Cluster:

- Go to the Databases section on DigitalOcean.
- Choose Configuration:
    - *Region: Frankfurt or the nearest location.*
    - *Database Type: PostgreSQL.*
    - *Plan: Basic - Shared CPU, with SSD storage.*
    - *Minimum Storage: 10 GB.*
    - *Cost: $13.00/month (minimum).*
- Select the project you created earlier.
- Click Create Database Cluster.

2. Secure the Database:

- Add trusted IPs under “Trusted Sources” when you create them.
    - *IP of the API’s droplet.*
    - *IP of the Scrapy project’s droplet.*
    - *IP of the VPN server’s droplet.*
    - (Optional) Add your computer’s IP for local testing.

3. Connection Details:

- Copy the database connection details:
- Add them to pipelines.py in your Scrapy project.
- Add DB URL in API/datasources/scrapy-app.db in API as a string.
- Download the CA-certificate and copy it to the API/datasources (or disable CA-authentication).

4. Initialise the Database:

- Connect locally to the database using the provided connection string.
- Run the SQL script to create the required four tables from scrapy-API/API-LB4/datasources or scrapy/db

Scrapy Project Deployment

1. Create a Droplet for the Scrapy Project:

- *Region: Frankfurt.*
- *Image: Docker (Latest) on Ubuntu 22.04 (via Marketplace).*
- *Plan: Basic - Shared CPU, SSD, $6/month.*
- *Authentication: SSH Key or Password.*
- Click Create Droplet.

2. Access the Droplet:

- SSH into the droplet:

  ssh root@<droplet-ip>

3. Deploy the Scrapy Project:

Make sure you're logged into your docker account on the droplet with *docker login.*

1- Disable any previously running containers using

docker container ps

docker container stop

docker container rm (container_id)

2- Pull my public Docker image, open port 6800 to Scrapyd, and run image:

docker pull rabiab/weather_forecasts_scraper:v1
ufw allow 6800
docker run -d --name final_v1 -p 6800:6800 rabiab/weather-forecasts-scraper:v1

4. Integrate with ScrapeOps:

- Log in to ScrapeOps.
- Add a Server:
- Server Type: Scrapyd.
- Add server details and follow installation instructions provided by ScrapeOps.
- Schedule jobs and manage spiders from the Jobs Tab.

5. Ensure Database Access:

- Add the Scrapy droplet's IP to the database's Trusted Sources.

6. Optional Custom Setup:

- Use your own Docker image and database :
- Add your personal ScrapeOps API key to an .env file in the Scrapy project.
- Add your database credentials to pipelines.py.

- Add your server IP to the scrapy/scrapy.cfg file.
- Build and push your Docker image then follow the instructions above.

API Deployment

1. Create a Droplet for the API:

- Region: Frankfurt.
- Image: Docker (Latest) on Ubuntu 22.04 (via Marketplace).
- Plan: Basic - Shared CPU, SSD, $6/month.
- Authentication: SSH Key or Password.
- Click Create Droplet.

2. Access the Droplet:

- SSH into the droplet:
  ssh root@<droplet-ip>

3. Deploy the API:

Make sure you're logged into your docker account on the droplet with *docker login.*

1- Disable any running containers using

2- Pull and run the API Docker image:
docker pull rabiab/historical_forecasts_api:v1
docker run -d --name final_v1 -p 3000:3000 rabiab/historical_forecasts_api:v1

4. Secure the API:

- Set up a firewall to whitelist VPN IPs:
- Follow this guide: [How to Whitelist IPs].
- Add the API's IP to the database's Trusted Sources.

5. Optional Custom Setup:

- Use your own database and Docker image:
- Get your database credentials and CA-certificate.
- Add them to the API's data sources folder and scrapyd-db file.
- Build and push your Docker image then follow instructions above.
- Add the API droplet's IP to the database's Trusted Sources.

VPN Server Deployment

1. Create a Droplet for the VPN Server:

- Region: Frankfurt.
- Image: VPN Server 11.4 on Ubuntu 24.04 (via Marketplace).
- Plan: Basic - Shared CPU, SSD, $6–24/month.
- Authentication: SSH Key or Password.
- Click Create Droplet.

2. Access the Droplet:

- SSH into the droplet:

ssh root@<droplet-ip>

3. Configure the VPN Server:

- Run the following command to retrieve the setup code secret:
sudo cat /vpn/setup-code.txt
- Go to http://<droplet-ip> in your browser.
- Paste the setup code and create an admin profile with your personal password password.
- Log in with the admin account.
- Create a new user with the role “user” (admin accounts cannot create VPN connections).
- Log out and log in again with the new user account.
- Create a VPN connection and download the configuration file.

4. Set Up the VPN Client:

- Start the Wireguard GUI (instructions for Wireguard and GUI installation in different file)
- Add a new profile and paste the contents of the downloaded configuration file.
- Save and connect.

5. Secure the VPN Server:

- Add the VPN’s IP to the database’s Trusted Sources.
- Secure your API by creating a firewall and whitelisting the VPN’s IP (if necessary).