■ Docker-based n8n Setup Report

1. Objective

The goal of this setup was to run n8n (a workflow automation tool) inside a Docker container to automate workflows locally on the system.

2. Steps Performed

Step 1: Pulling n8n Docker Image

Command used:

```
docker pull n8nio/n8n
```

The image was successfully pulled from Docker Hub. All layers were downloaded and verified with the image ID: **sha256:c6f4d7dade75e6b970e3319a57454a507244d760f6d69e4f18d5d312**.

Step 2: Running n8n in a Docker Container

Command used:

```
docker run -it --rm -p 5678:5678 -e N8N_BASIC_AUTH_ACTIVE=true -e N8N_BASIC_AUTH_USER=admin -e N8N_BASIC_AUTH_PASSWORD=n8n@work n8nio/n8n
```

Environment variables were set to enable basic authentication:

- Username: admin
- Password: n8n@work

Port mapping: 5678 (container) → 5678 (host). The web interface became available at http://localhost:5678

Step 3: Initialization Logs

The container initialized successfully, generating configuration files inside `/home/node/.n8n/config`. Multiple database migrations were performed automatically to prepare the system for use.

Step 4: Accessing n8n Web Interface

Once the container was running, the user accessed the n8n setup page via browser (http://localhost:5678/setup). The setup included creating an owner account with email, name, and password.

Step 5: Workflow Dashboard

After successful setup, the user accessed the n8n dashboard where workflows, credentials, executions, and data tables can be managed. A welcome message confirmed successful setup.

Step 6: License Activation

The system displayed that the license key was sent to the registered email. After activation, the user’s Community Edition license was successfully registered.

3. Conclusion

The Docker-based deployment of n8n was successful. The application was accessed locally, configured, and the license was activated. n8n is now ready for workflow automation on the local machine.