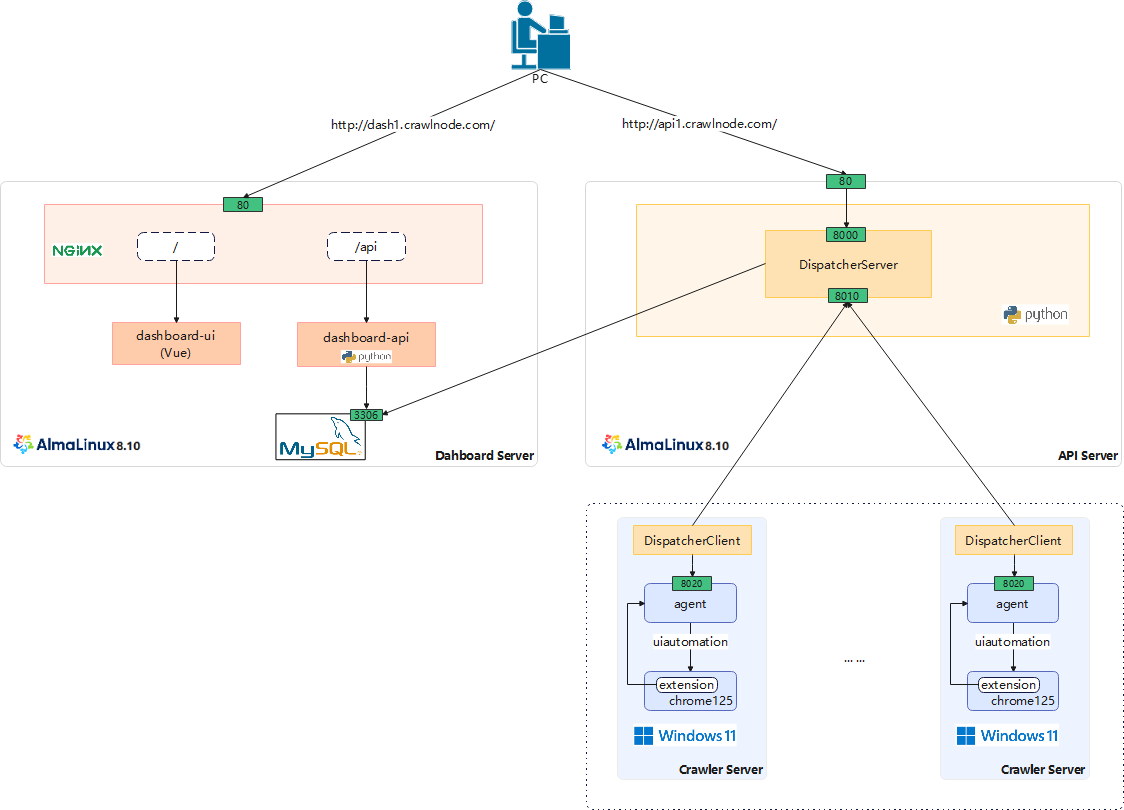
# Deployment architecture diagram



# Component Description

|  |  |  |
| --- | --- | --- |
| **Components** | **description** | **Introduction** |
| Dashboard-ui | Node.js + Vue.js frontend with Ant Design, calling /api endpoints | Web-based Dashboard: Monitor & Manage - Crawler | Session | Log |
| Dashboard-api | FastAPI/Uvicorn web API (Python) with MySQL storage, reverse-proxied through Nginx at /api |
| DispatcherServer | Distributes and load-balances requests across DispatcherClient instances.  Periodically stores data to MySQL database. | Load balancer and session distribution system |
| DispatcherClient | Connects to DispatcherServer, accepts dispatched requests, and forwards them to agent. |
| Agent | Core automation engine with HTTP API and WebSocket communication | |
| uia\_extension | Monitors browser events and network traffic | |

# Standalone crawler

## Agent deployment

* Environmental requirements:
* Operating system: Windows 11 x64
* Python: Python 3.8
* Browser: Chrome 125
* Python Installation:

1. Install Python to {pythonDir}
2. Add path to environment variables (option)
3. Install dependencies

> pip install fastapi uvicorn[standard] uiautomation psutil

* Chrome Installation

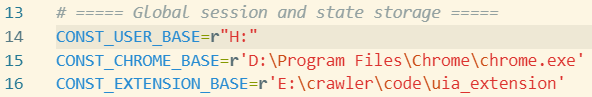
1. Install chrome to {chromeDir}
2. Create a chrome data directory {chromeDataDir}

* Extension Installation:

1. Deploy uia\_extension contents to {uiaextensionDir}

* Agent Installation:

1. Copy agent.py to {agentDir}
2. Modify user\_base、chrome\_path and ext\_path in the agent.py file



1. Start the service:

> cd {agentDir}

> {pythonDir}/python.exe agent.py

1. Verify that the startup is successful (you should see the following information) :

> INFO: Uvicorn running on http://127.0.0.1:8020 (Press CTRL+C to quit)

# Cluster Management

## Mysql preparation

1. Create database `dashboard`
2. Create a User
3. Authorize user
4. Execute the initialization script `create\_table.sql`

## Dashboard deployment

* Environmental requirements:
* Operating system: AlmaLinux 8.10
* Nginx: Nginx 1.20

### Dashboard-UI deployment

1. Deploy the dashboard-ui dist files to Nginx's root directory
2. Configure HTTP Basic Authentication

location / {

auth\_basic "secret";

auth\_basic\_user\_file /usr/share/nginx/html/.htpasswd;

}

1. Use the htpasswd tool to create a file containing usernames and encrypted passwords

### Dashboard-api deployment

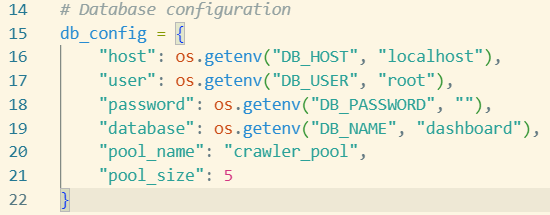
* Python Installation:

1. Install Python
2. Install dependencies

# pip install fastapi uvicorn[standard] pydantic mysql-connector-python

* Dashboard-api installation

1. Copy dashboard.py to {dashboardDir}
2. Modify db\_config in the dashboard.py file



1. Configure Nginx to proxy /api requests to the dashboard service： http://localhost:8050
2. Start the service

# cd {dashboardDir}

# nohup python dashboard.py > output.log 2>&1 &

1. Verify that the startup is successful (you should see the following information)

# INFO: Uvicorn running on http://127.0.0.1:8050 (Press CTRL+C to quit)

## DispatcherServer deployment

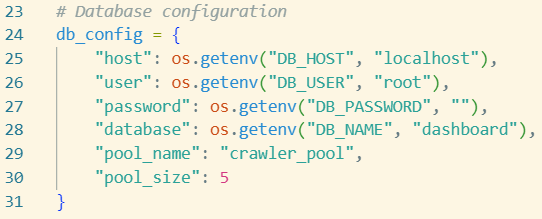
* Environmental requirements:
* Operating system: AlmaLinux 8.10
* Python: Python 3.8
* Python Installation:

1. Install Python
2. Install dependencies

# pip install mysql-connector-python

* DispatcherServer installation:

1. Copy DispatcherServer.py to {dashboardServerDir}
2. Modify db\_config in the DispatcherServer.py file



1. Start the service

# cd {dashboardServerDir}

# nohup python DispatcherServer.py > output.log 2>&1 &

1. Verify that the startup is successful (you should see the following information)

# [SERVER][MAIN] Server started. HTTP(0.0.0.0:8000), Client(0.0.0.0:8010)"

## DispatcherClient deployment

* Environmental requirements:
* Operating system: Windows 11 x64
* Python: Python 3.8
* Python Installation:

1. Install Python to {pythonDir}
2. Add path to environment variables (option)
3. Install dependencies

> pip install psutil

* DispatcherClient installation:

1. Copy DispatcherClient.py to {dispatcherClientDir}
2. Modify DISPATCHER\_HOST in the DispatcherClient.py file



1. Start the service

> cd {dispatcherClientDir}

> {pythonDir}/python.exe DispatcherClient.py

1. Verify that the startup is successful (you should see the following information)

> [CLIENT][CONN] Connected to dispatcher({X.X.X.X}:8010), forward to http(127.0.0.1:8020)