

Setting up n8n workflow

Docker MinIO module **WHMCS**

[Order now](#) | [Download](#) | [FAQ](#) | [n8n](#)

Overview

The **Docker MinIO WHMCS module** uses a specially designed workflow for **n8n** to automate deployment processes. The workflow provides an API interface for the module, receives specific commands, and connects via SSH to a server with Docker installed to perform predefined actions.

Prerequisites

- You must have your own **n8n** server.
- Alternatively, you can use the official **n8n** cloud installations available at: [n8n Official Site](#)

Installation Steps

Install the Required Workflow on n8n

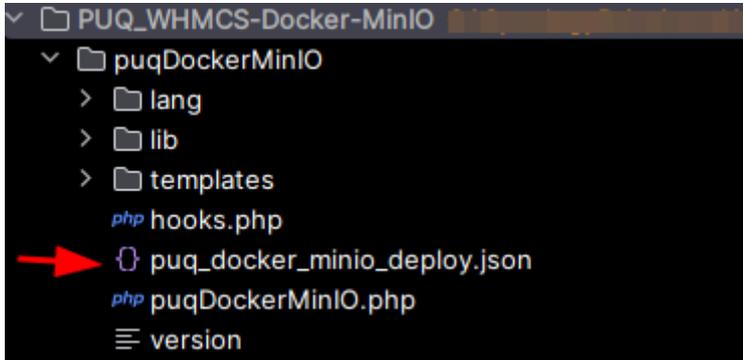
You have two options:

Option 1: Use the Latest Version from the n8n Marketplace

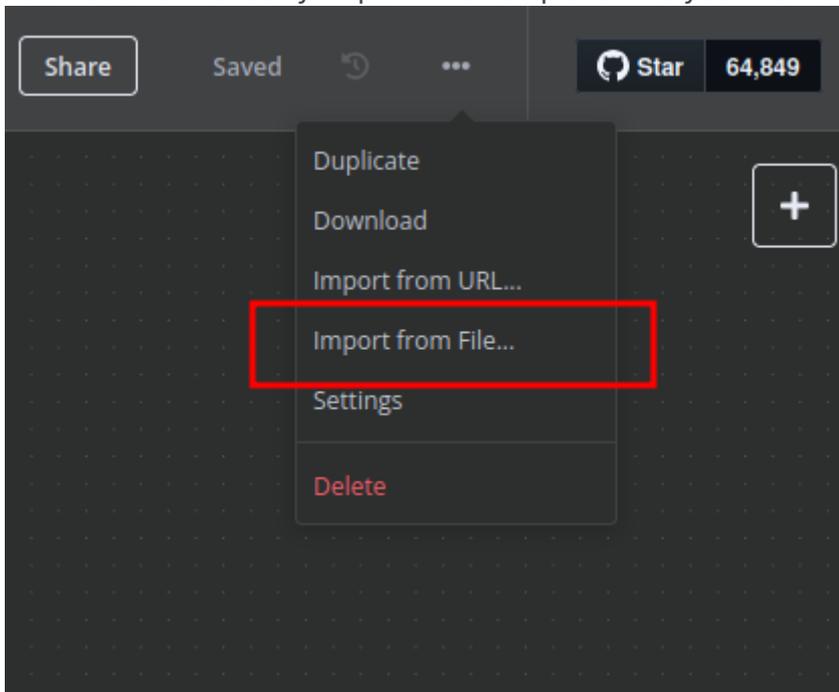
- The latest workflow templates for our modules are available on the official n8n marketplace.
- Visit our profile to access all available templates: [PUQcloud on n8n](#)

Option 2: Manual Installation

- Each module version comes with a workflow template file.



- You need to manually import this template into your n8n server.



n8n Workflow API Backend Setup for WHMCS/WISECP

Configure API Webhook and SSH Access

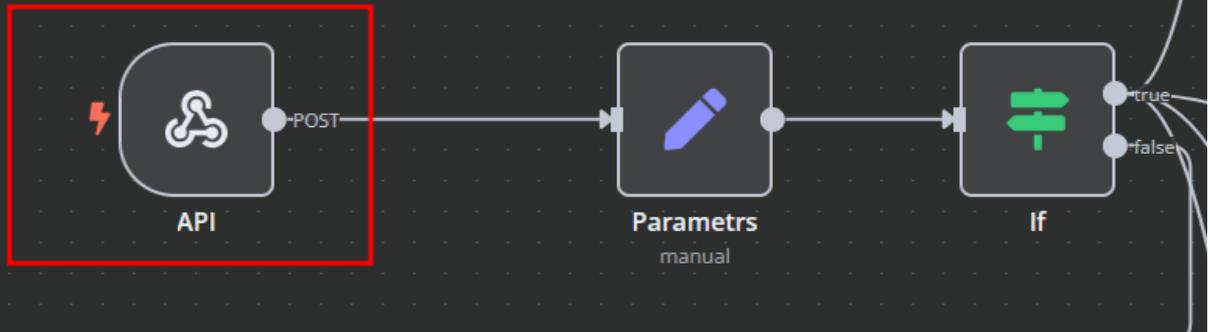
- Create a **Basic Auth Credential**

for the Webhook API Block in n8n.

- screen_right

Additional Resources

- Full documentation: <https://doc.puq.info/books/docker-minio-whmcs-module>
- WHMCS module: <https://puqcloud.com/whmcs-module-docker-minio.php>



The image shows a dark-themed user interface for configuring an API. On the left, a panel titled "Pull in events from Webhook" contains a red button labeled "Listen for test event" and a paragraph of text: "Once you've finished building your workflow, run it without having to click this button by using the production webhook URL. [More info](#)".

The right-hand panel is titled "API" and has a red button "Listen for test event" in the top right corner. Below the title are tabs for "Parameters" (selected) and "Settings", along with a "Docs" link. Under the "Parameters" tab, there is a section for "Webhook URLs" with two buttons: "Test URL" (selected) and "Production URL". Below these is a text input field containing the URL "https://n8n.puqcloud.com/webhook-test/docker-minio" with a "POST" label on the left.

Below the URL field are several configuration sections: "HTTP Methods" with a dropdown menu showing "POST"; "Path" with a text input field containing "docker-minio"; "Authentication" with a dropdown menu showing "Basic Auth"; and "Credential for Basic Auth" with a text input field containing "Incom API User".

A dropdown menu is open below the credential field, showing two options: "Incom API User" (with "Basic Auth" below it) and "Unnamed credential" (with "Basic Auth" below it). A red rectangular box highlights a button labeled "+ Create new credential" at the bottom of this menu.

At the bottom of the "Credential for Basic Auth" section, there is a "No properties" label and a dropdown menu with the text "Add option".

API credential
Basic Auth

Save

Connection

Need help filling out these fields? [Open docs](#)

Sharing

Details

User

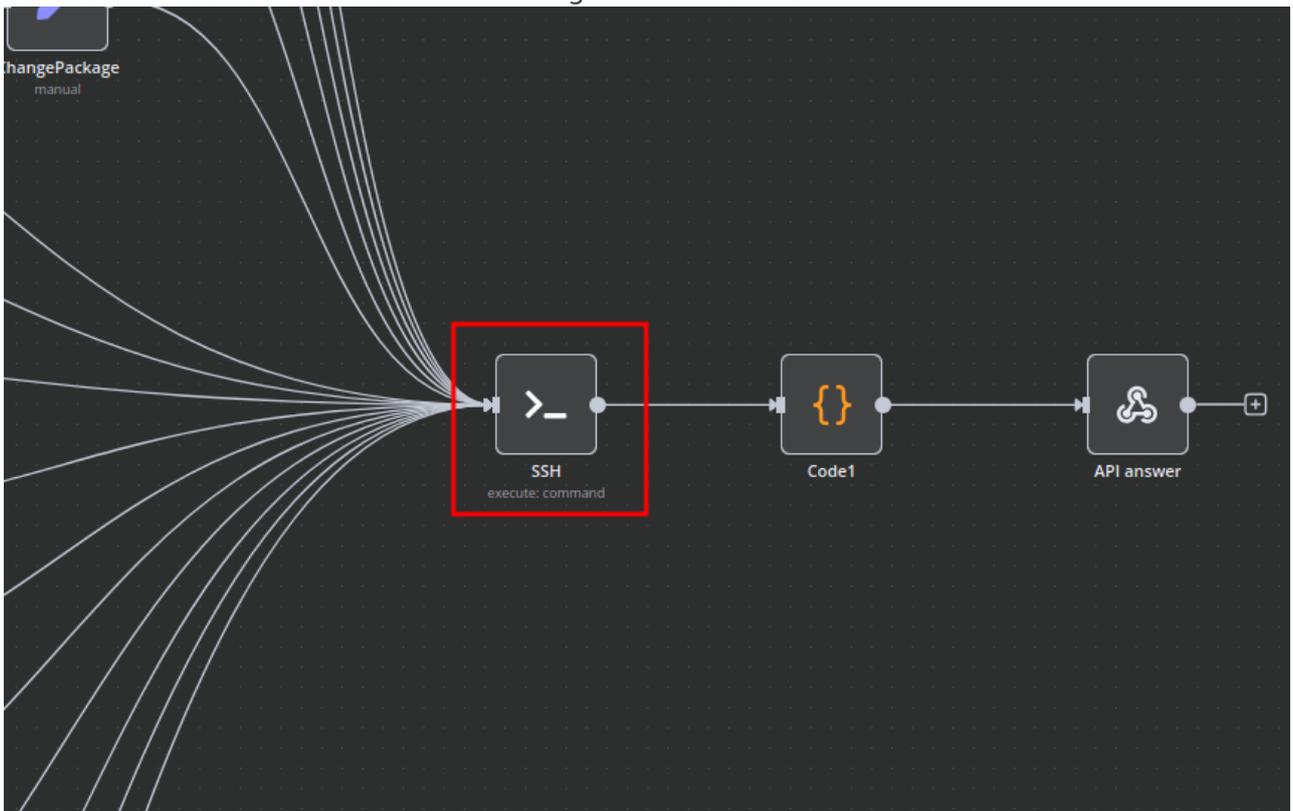
test

Password

.....

Enterprise plan users can pull in credentials from external vaults. [More info](#)

- Create an **SSH Credential** for accessing a server with Docker installed.



> SSH

Test step

Parameters

Settings

Docs

Credential to connect with

d01-test.uuq.pl-puq

d01-test.uuq.pl-puq
SSH Password

+ Create new credential

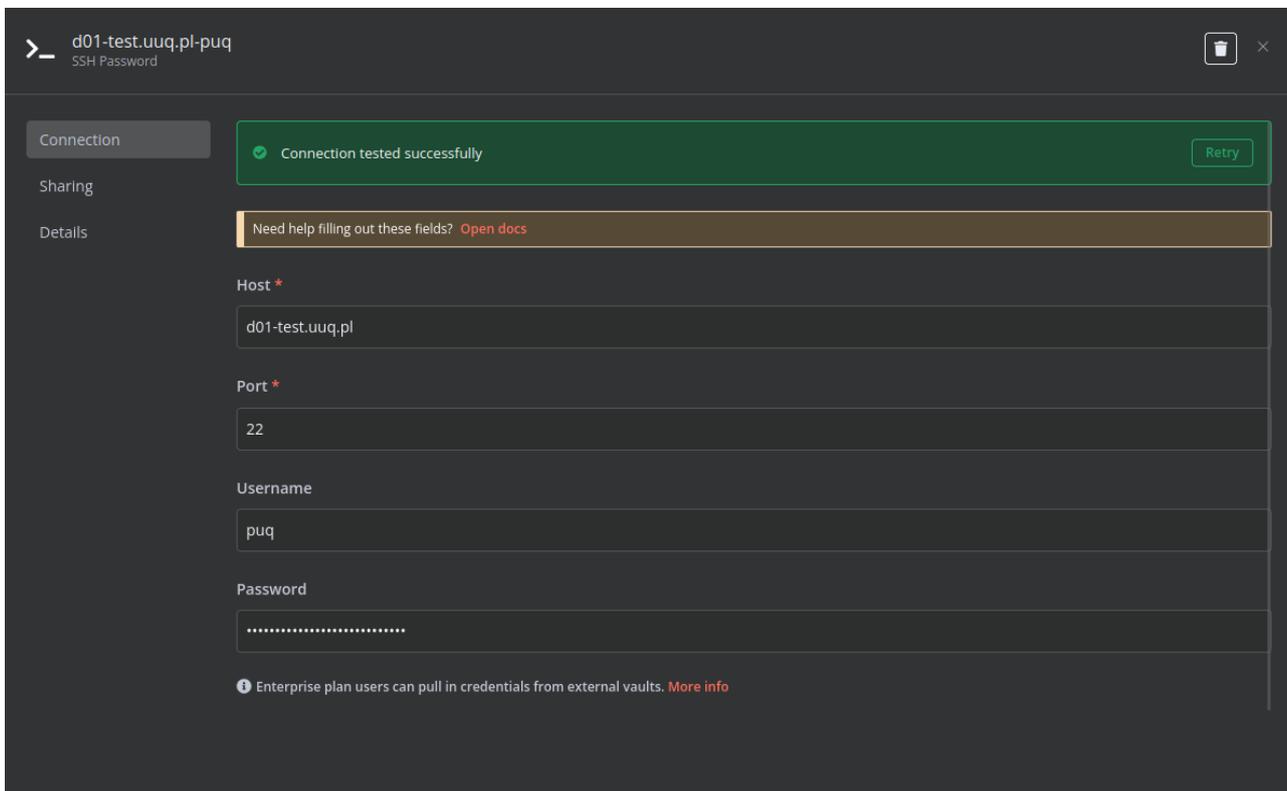
Execute

Command

fx {{ \$json.sh }}

Working Directory

fx /

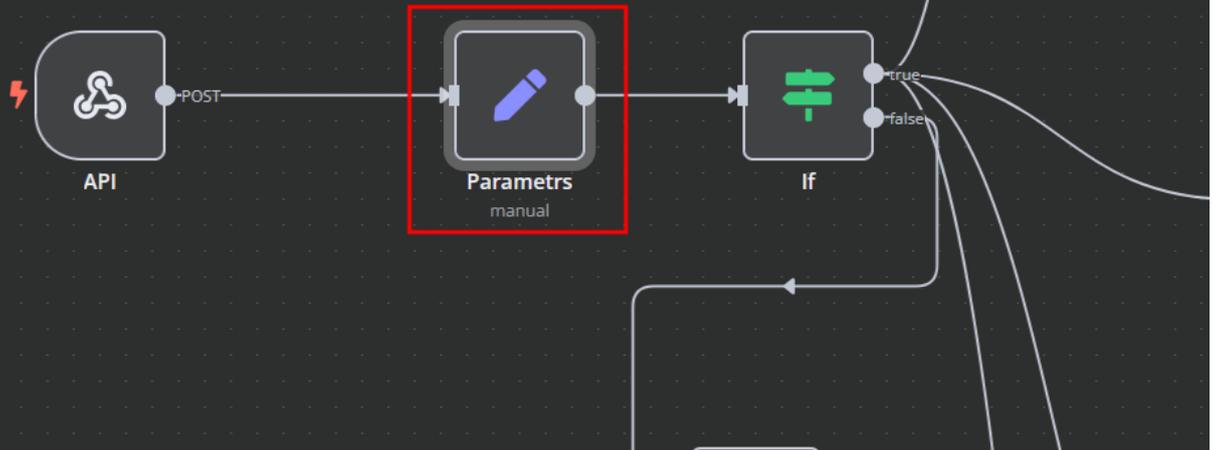


Modify Template Parameters

In the **Parameters** block of the template, update the following settings:

Additional Resources

- Full documentation: <https://doc.puq.info/books/docker-minio-whmcs-module>
- WHMCS module: <https://puqcloud.com/whmcs-module-docker-minio.php>



 **Parameters** 🚧 Test step

Parameters Settings Docs [↗](#)

Mode
Manual Mapping ▼

Fields to Set

server_domain
A String ▼
d01-test.uuq.pl
[empty]

clients_dir
A String ▼
/opt/docker/clients
[empty]

mount_dir
A String ▼
/mnt
[empty]

screen_left
A String ▼
{{
[empty]

screen_right
A String ▼
}}
[empty]

- `server_domain` – Must match the domain of the WHMCS/WISECP Docker server.
- `clients_dir` – Directory where user data related to Docker and disks will be stored.
- `mount_dir` – Default mount point for the container disk (recommended not to change).

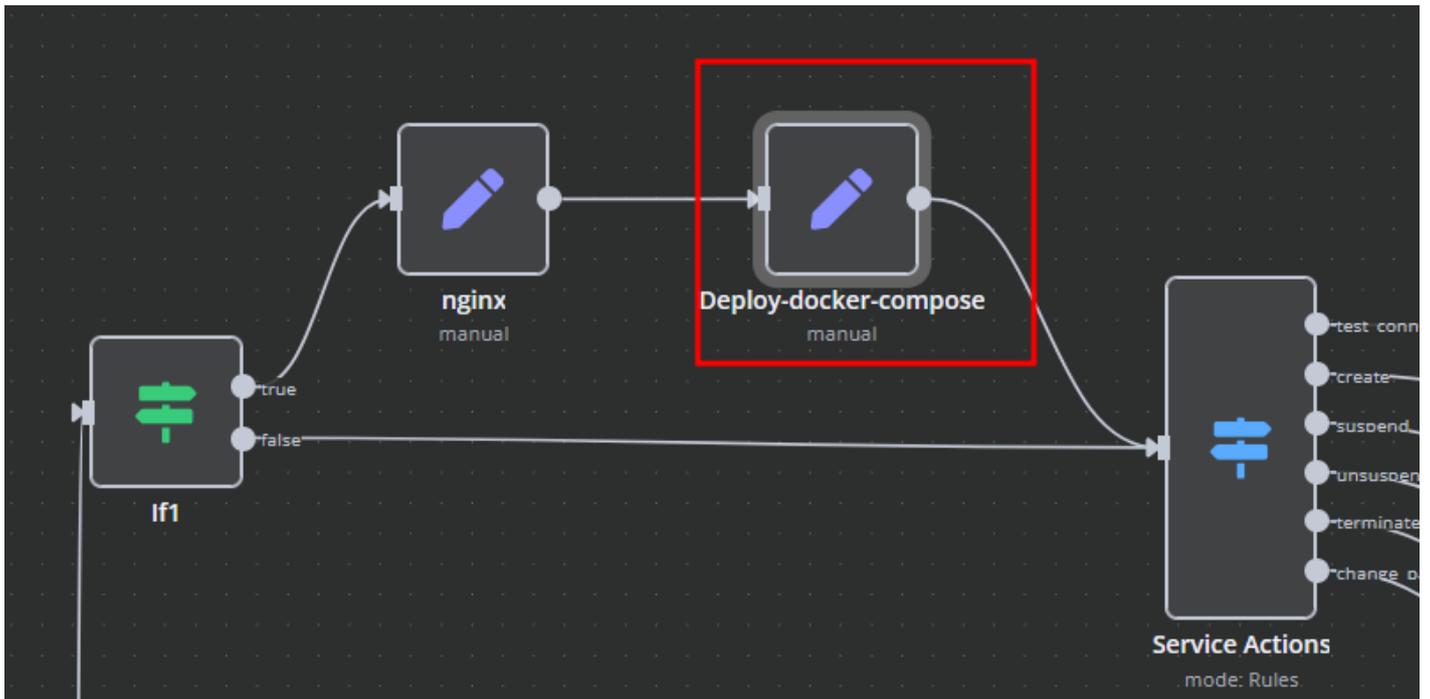
Do not modify the following technical parameters:

- `screen_left`
- `screen_right`

Deploy-docker-compose

In the **Deploy-docker-compose** element, you have the ability to modify the Docker Compose configuration, which will be generated in the following scenarios:

- When the service is created
- When the service is unlocked
- When the service is updated



Expression

Anything inside `{{ }}` is JavaScript. [Learn more](#)

```
version: "3"

services:
  {{ $('API').item.json.body.domain }}:
    image: minio/minio
    restart: unless-stopped
    container_name: {{ $('API').item.json.body.domain }}
    command: server /data --console-address ":9001"
    environment:
      MINIO_ROOT_USER: {{ $('API').item.json.body.username }}
      MINIO_ROOT_PASSWORD: {{ $('API').item.json.body.password }}
      MINIO_BROWSER_REDIRECT_URL: https://console.{{ $('API').item.json.body.domain }}
      LETSENCRYPT_HOST: {{ $('API').item.json.body.domain }},console.{{
$('API').item.json.body.domain }}
      VIRTUAL_HOST_MULTIPORTS: |-
        {{ $('API').item.json.body.domain }}:
          "/":
            port: 9000
        console.{{ $('API').item.json.body.domain }}:
          "/":
            port: 9001
    volumes:
      - "{{ $('Parameters').item.json.mount_dir }}/{{ $('API').item.json.body.domain }}/data:/
data"
    networks:
      - nginx-proxy_web
    mem_limit: "{{ $('API').item.json.body.ram }}"G"
    cpus: "{{ $('API').item.json.body.cpu }}"

networks:
  nginx-proxy_web:
    external: true
```

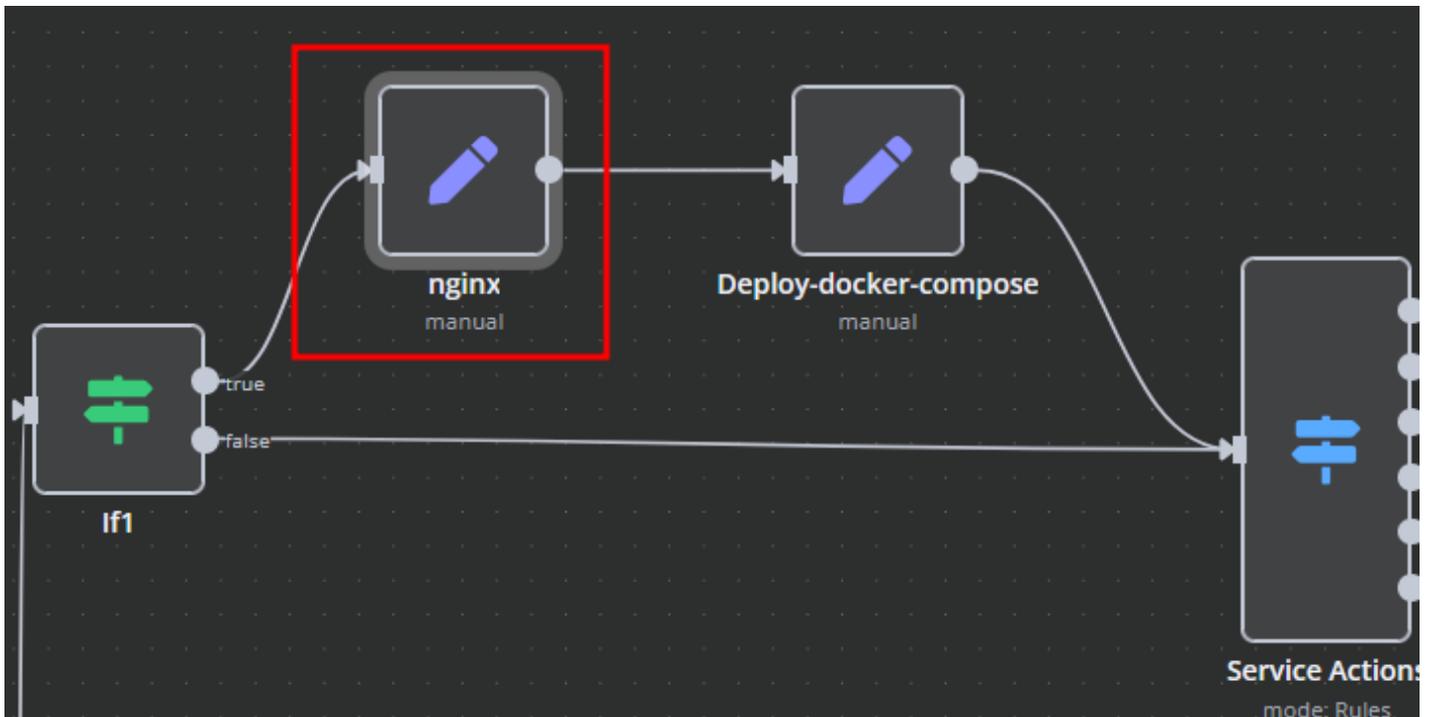
nginx

In the **nginx** element, you can modify the configuration parameters of the web interface proxy server.

- The **main** section allows you to add custom parameters to the **server** block in the proxy server configuration file.
- The **main_location** section contains settings that will be added to the **location /** block of

the proxy server configuration. Here, you can define custom headers and other parameters specific to the root location.

- The **console** section allows you to add custom parameters to the **server** block in the WEB console proxy server configuration file.
- The **console_location** section contains settings that will be added to the **location /** block of the WEB console proxy server configuration. Here, you can define custom headers and other parameters specific to the root location of WEB console.



Mode

Manual Mapping

Fields to Set

main

A String

```
= ignore_invalid_headers off;
client_max_body_size 0;
proxy_buffering off;
proxy_request_buffering off;
```

ignore_invalid_headers off; client_max_body_size 0; proxy_buffering off; proxy_request_buffering off;

main_location

A String

```
= # Custom header
proxy_set_header Host $http_host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
```

Custom header proxy_set_header Host \$http_host; proxy_set_header X-Real-IP \$remote_addr; proxy_set_head...

console

A String

```
= ignore_invalid_headers off;
client_max_body_size 0;
proxy_buffering off;
proxy_request_buffering off;
```

ignore_invalid_headers off; client_max_body_size 0; proxy_buffering off; proxy_request_buffering off;

console_location

A String

```
= # Custom header
proxy_set_header Host $http_host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Real-IP-Proxy true;
```

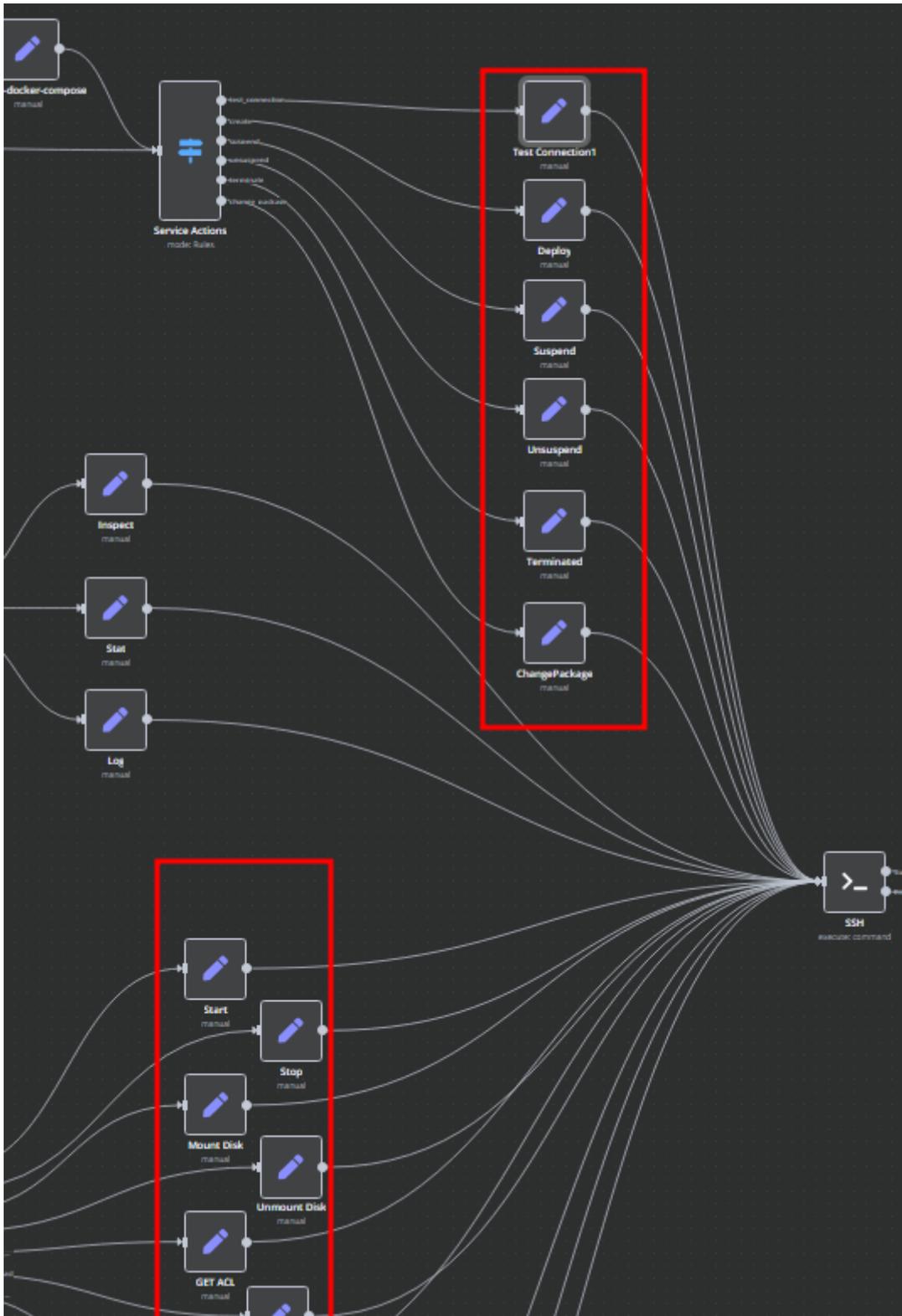
Custom header proxy_set_header Host \$http_host; proxy_set_header X-Real-IP \$remote_addr; proxy_set_head...

Bash Scripts

Management of Docker containers and all related procedures on the server is carried out by

executing Bash scripts generated in **n8n**. These scripts return either a JSON response or a string.

- All scripts are located in elements directly connected to the **SSH** element.
- You have full control over any script and can modify or execute it as needed.



Revision #2

Created 14 March 2025 17:43:07 by Ruslan

Updated 21 March 2025 14:22:06 by Yuliia Noha