

# Setting up n8n workflow

## Docker Grafana module **WHMCS**

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

## Overview

The **Docker Grafana 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

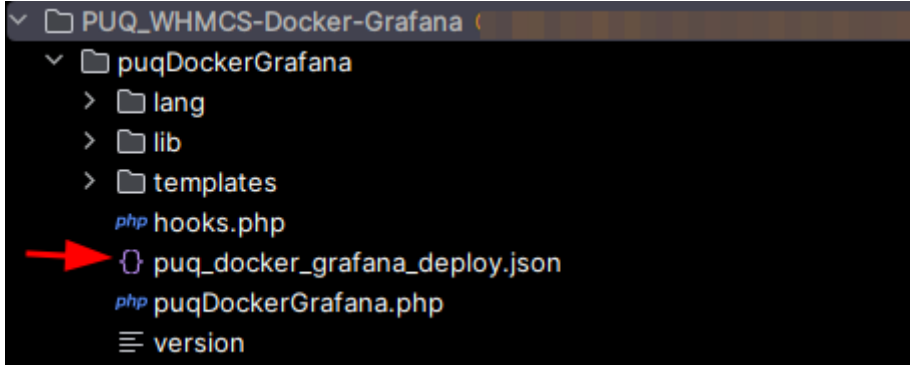
You have two options:

#### **Option 1: Use the Latest Version from the n8n Marketplace**

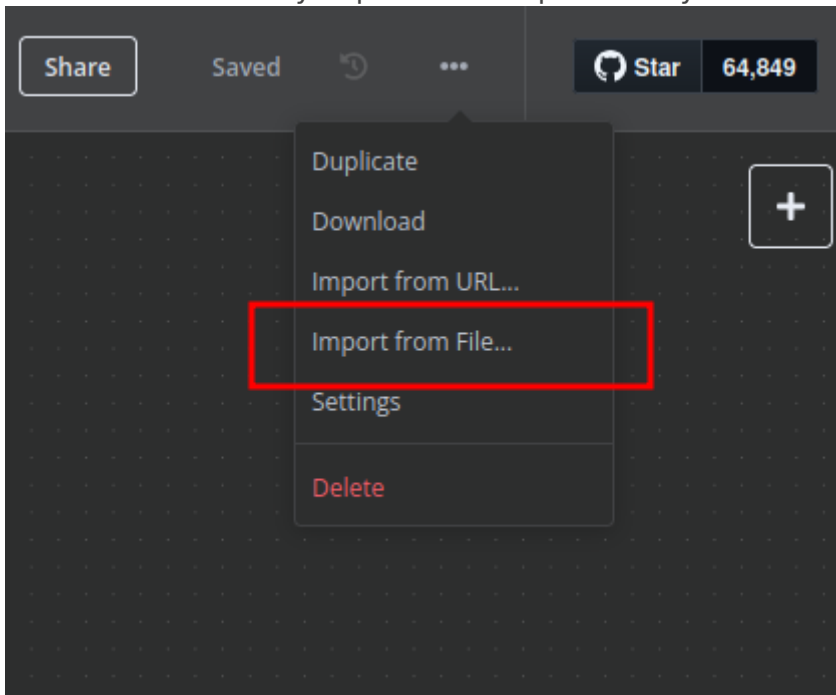
- The latest workflow templates for our modules are available on the official n8n marketplace.
- Visit our profile to access all available templates: [PUQcloud on n8n](#)

## Option 2: Manual Installation

- Each module version comes with a workflow template file.



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

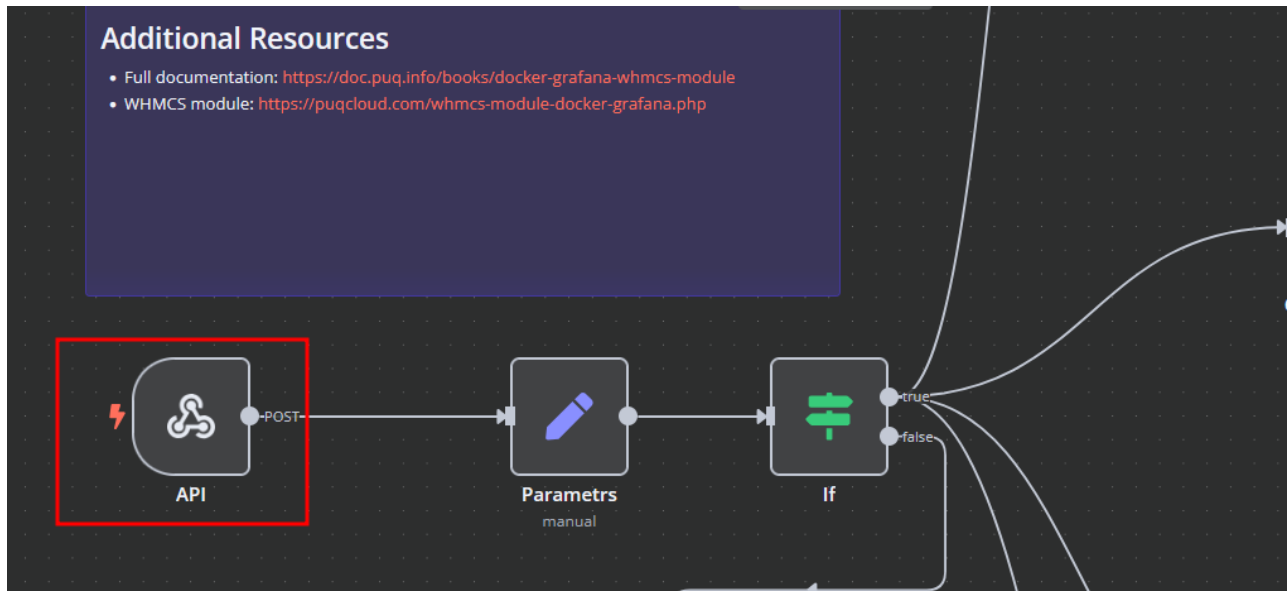


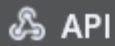
# n8n Workflow API Backend Setup for WHMCS/WISECP

## Configure API Webhook and SSH Access

- Create a **Basic Auth Credential**

for the Webhook API Block in n8n.





API

Listen for test event

Parameters

Settings

Docs

Webhook URLs

Test URL

Production URL

POST

https://n8n.puqcloud.com/webhook-test/docker-grafana

HTTP Methods

POST



Path

docker-grafana

Authentication

Basic Auth

Credential for Basic Auth

Grafana



Basic Auth

Immich

Basic Auth

InfluxDB

Basic Auth

MinIO

Basic Auth

n8n

Basic Auth

Vaultwarden

Basic Auth

+ Create new credential

Grafana Basic Auth

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

Connection

Sharing

Details

User

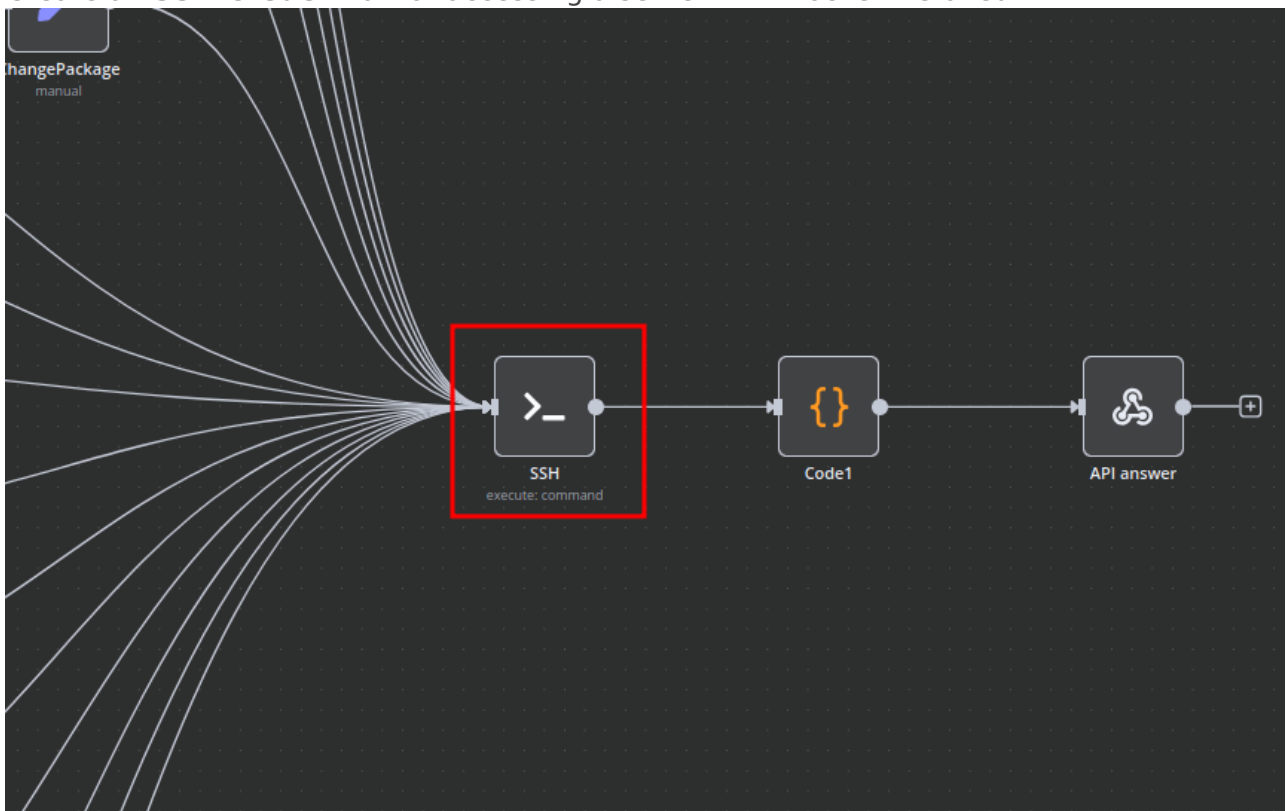
Grafana

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 /

>

d01-test.uuq.pl-puq

SSH Password

×

Connection

Sharing

Details

✔ Connection tested successfully

Retry

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

Host \*

d01-test.uuq.pl

Port \*

22

Username

puq

Password

.....

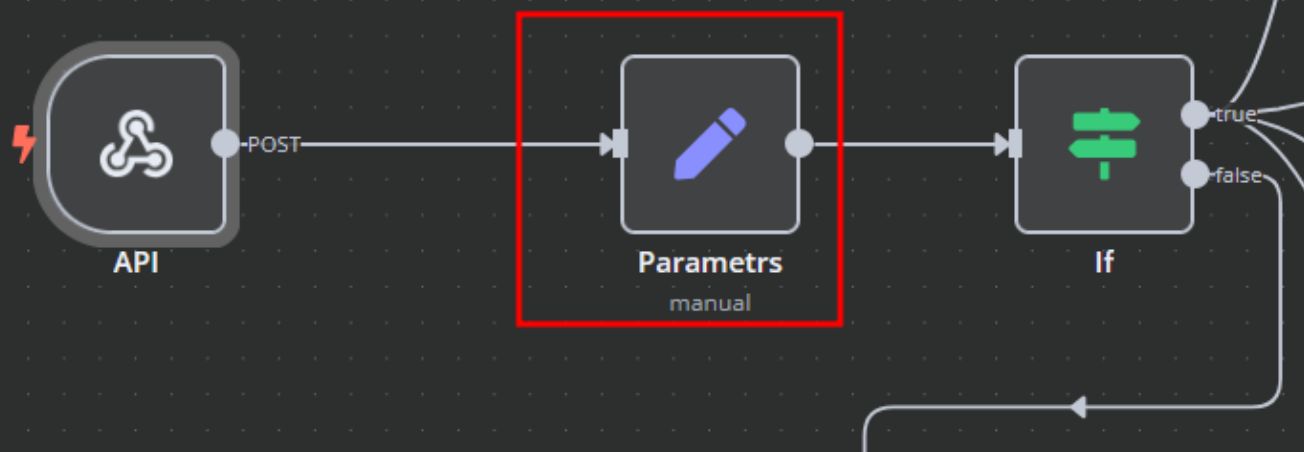
ⓘ Enterprise plan users can pull in credentials from external vaults. [More info](#)

# Modify Template Parameters


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

## Additional Resources

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







 **Parameters**

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).

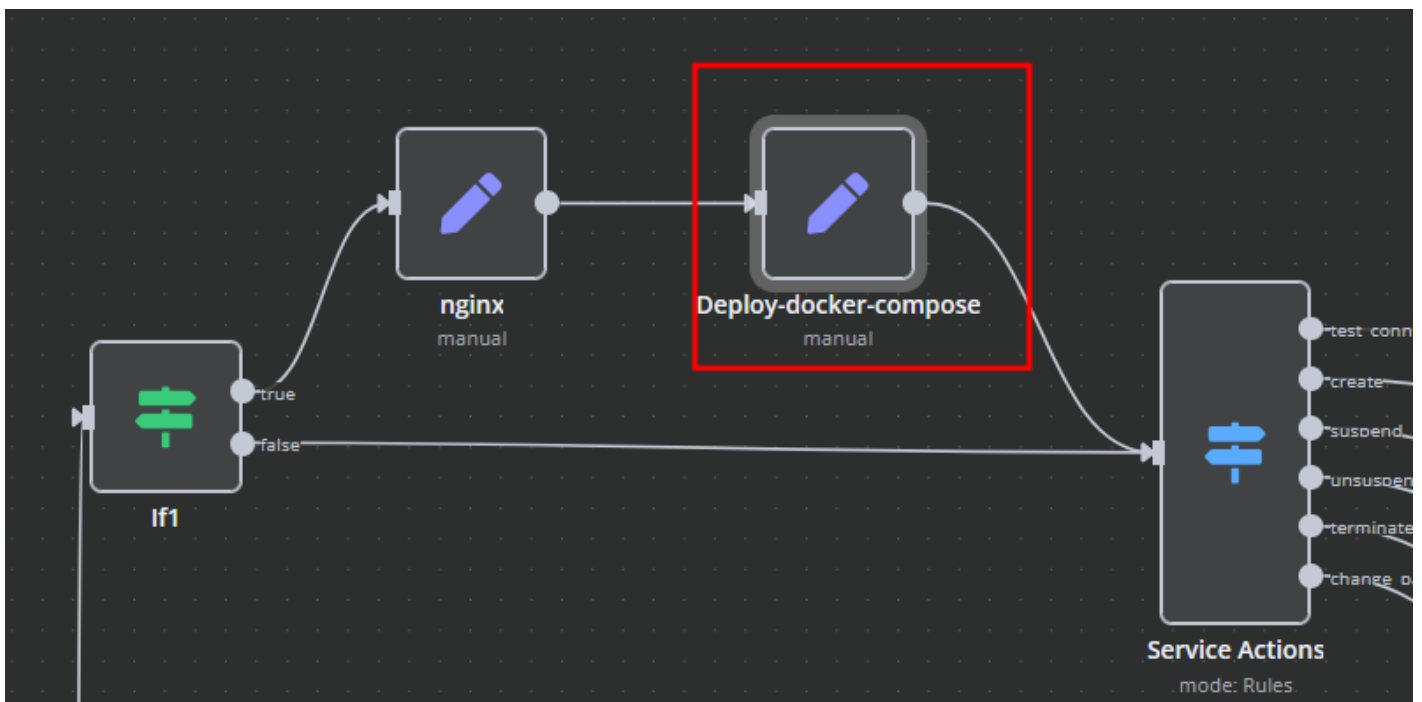
**Do not modify** the following technical parameters:

- `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](#)

```
name: "{{ $('API').item.json.body.domain }}"

services:
  {{ $('API').item.json.body.domain }}:
    container_name: {{ $('API').item.json.body.domain }}
    image: grafana/grafana:latest
    restart: unless-stopped
    volumes:
      - {{ $('Params').item.json.mount_dir }}/{{ $('API').item.json.body.domain }}/data:/var/
lib/grafana
      - {{ $('Params').item.json.mount_dir }}/{{ $('API').item.json.body.domain }}/logs:/var/
log/grafana
      - {{ $('Params').item.json.mount_dir }}/{{ $('API').item.json.body.domain }}/
provisioning:/etc/grafana/provisioning
    environment:
      - LETSENCRYPT_HOST={{ $('API').item.json.body.domain }}
      - VIRTUAL_HOST={{ $('API').item.json.body.domain }}
      - GF_SECURITY_ADMIN_USER={{ $('API').item.json.body.username }}
      - GF_SECURITY_ADMIN_PASSWORD={{ $('API').item.json.body.password }}
      - GF_PATHS_CONFIG=/etc/grafana/grafana.ini
    healthcheck:
      disable: false
    networks:
      - nginx-proxy_web
    mem_limit: "{{ $('API').item.json.body.ram }}"G
    cpus: "{{ $('API').item.json.body.cpu }}"

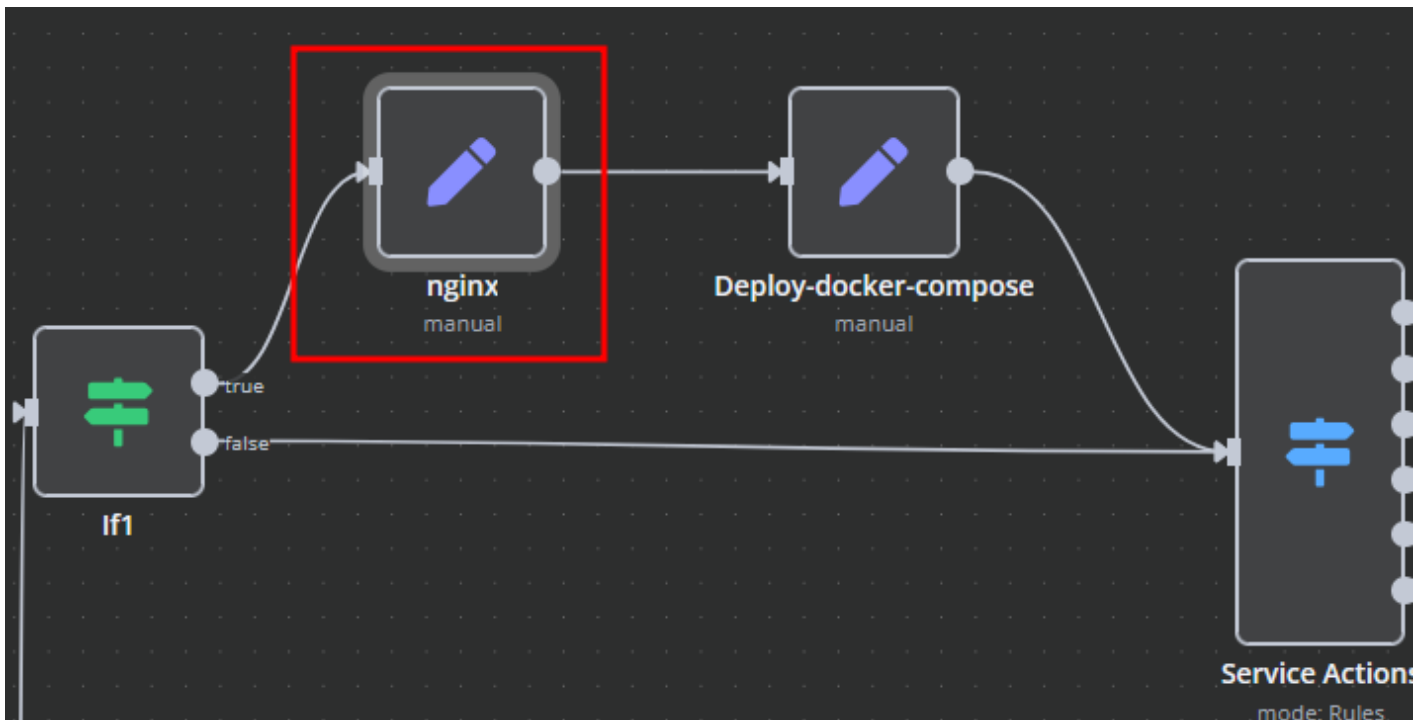
networks:
  nginx-proxy_web:
    external: true
```

# 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.



**nginx** Test step

Parameters Settings Docs

Mode

Manual Mapping

Fields to Set

main	A String	=	[empty]
main_location	A String	=	proxy_pass_header Server; proxy_set_header X-Real-IP \$remote_addr; proxy_set_header X-Forwarded-For \$proxy_add_x_forwarded_for; proxy_set_header X-Scheme \$scheme; proxy_set_header Host \$http_host;

Drag input fields here or [Add Field](#)

Include Other Input Fields

Options

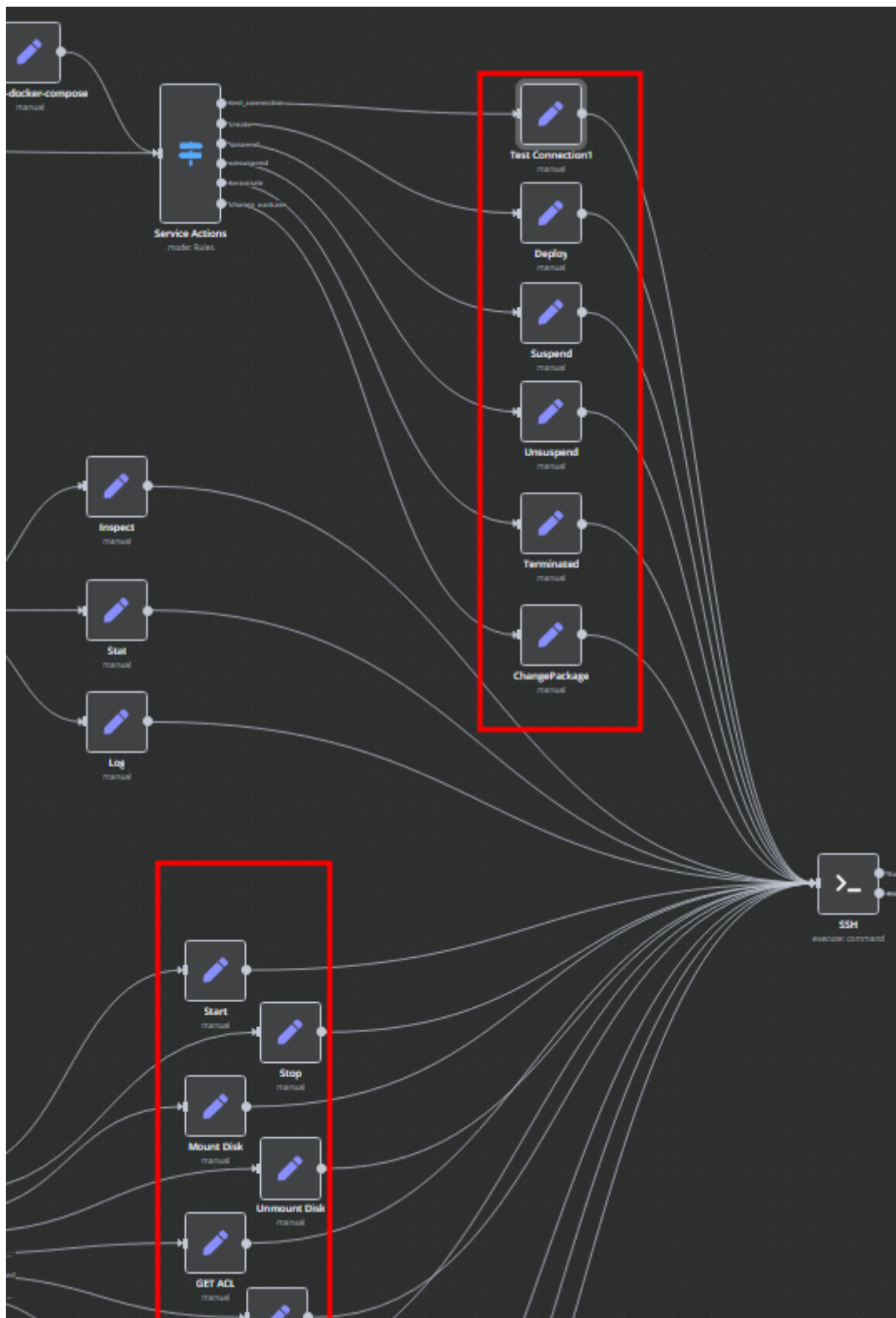
No properties

Add option

# 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 #3

Created 23 March 2025 20:30:51 by Ruslan

Updated 26 March 2025 12:01:43 by Yuliia Noha