Aufbau & Installation

Konzept und Aufbau

Konzept

Wird das Script newNIG.sh ausgeführt kopiert dies die einzelnen Dateien an die benötigen Stellen und ändert vorher platzierte Platzhalter in die nötigen Werte. Dies ermöglicht die einzelnen Environments im Nachhinein noch anzupassen, zu starten etc.. Jedes Environment hat seine eigene docker-compose Datei. Alle sind in eigenen Ordnern im Ordner “nig-environments” geordnet. Lediglich die nginx.conf befindet sich im Ordner “/etc/nginx/nigs”.

Aufbau

Es gibt einen NIG-Root-Ordner. In diesem liegen alle Dateien die zur Einrichtung benötigt werden. In diesem werden auch alle NIG-Environments gespeichert. Alle NIGs verwenden persistenten Speicher. In dem NIG-Ordner befinden sich mehrere Ordner und das Installationsscript. Die Ordnerstruktur:

- base-nginx
- base-nig
- newNIG.sh
- nig-environments

In base-nginx befindet sich eine Datei zur Configuration von Nginx. In base-nig befindet sich die docker-compose Datei für ein NIG-Environment, ebenso eine custom Configurationsdatei für grafana (grafana.ini) und nodered (settings.js).

Installation

Was installiert sein muss

1. Docker
2. Docker-Compose
3. Node
4. Nginx Reverse Proxy

Installation

1. Ordered List Item

nig.sh

```
# Define all variables
DOMAIN="nig.eolab.de"
```

```
ROOT="groupX"
NODE_RED_USERNAME="groupX"
NODE_RED_PASSWORD="password"
NODERED_PORT=17609
GRAFANA_PORT=17709
GRAFANA_USERNAME="groupX"
GRAFANA_PASSWORD="password"
# Generate the hash for the password
NODE_RED_PASSWORD=$(echo $(node -e
"console.log(require('bcryptjs').hashSync(process.argv[1], 8));"
$NODE_RED_PASSWORD) | sed s/[$]$
# Copy all the base-nig files to a new directory
cp -r ./base-nig/ ./nig-environments/$ROOT/
# Edit the docker-compose file for the new environment
sed -i "s/{{ROOT}}/$ROOT/g" ./nig-environments/$ROOT/docker-compose.yml
sed -i "s/{{NODE_RED_USERNAME}}/$NODE_RED_USERNAME/g" ./nig-
environments/$ROOT/docker-compose.yml
sed -i "s|{{NODE_RED_PASSWORD}}|$NODE_RED_PASSWORD|g" ./nig-
environments/$ROOT/docker-compose.yml
sed -i "s/{{GRAFANA_USERNAME}}/$GRAFANA_USERNAME/g" ./nig-
environments/$ROOT/docker-compose.yml
sed -i "s/{{GRAFANA_PASSWORD}}/$GRAFANA_PASSWORD/g" ./nig-
environments/$ROOT/docker-compose.yml
sed -i "s/{{GRAFANA_PORT}}/$GRAFANA_PORT/g" ./nig-
environments/$ROOT/docker-compose.yml
sed -i "s/{{NODERED_PORT}}/$NODERED_PORT/g" ./nig-
environments/$ROOT/docker-compose.yml
# Edit grafana.ini for the new environment
sed -i "s/{{ROOT}}/$ROOT/g" ./nig-environments/$ROOT/grafana.ini
sed -i "s/{{DOMAIN}}/$DOMAIN/g" ./nig-environments/$ROOT/grafana.ini
# Copy base-nginx config to nginx-nig folder
cp -r ./base-nginx/base-nig.conf "/etc/nginx/nigs/$ROOT.conf"
# Edit the base-nginx config
sed -i "s/{{ROOT}}/$ROOT/g" "/etc/nginx/nigs/$ROOT.conf"
sed -i "s/{{GRAFANA_PORT}}/$GRAFANA_PORT/g"
"/etc/nginx/nigs/$ROOT.conf"
sed -i "s/{{NODERED_PORT}}/$NODERED_PORT/g"
"/etc/nginx/nigs/$ROOT.conf"
# No need to edit the node_red settings.js -> it works with environment
variables (is already modified tho)
# Make the data for all ubuntu-users available
sudo chmod -R 777 ./nig-environments/$ROOT
# Start the NIG-Environment
cd ./nig-environments/$ROOT
sudo docker-compose up -d
# The persitent volumes are now created but not accessible for docker
sudo docker-compose stop
# Otherwise the persistent storage is not available for docker
sudo chmod -R 777 ./
# Now it has access and can be started
```

```
sudo docker-compose up -d
# Restart nginx
sudo systemctl restart nginx
# Otherwise the persistent storage is not available for docker
sudo chmod -R 777 ./
```

From:
https://wiki.eolab.de/ - **HSRW EOLab Wiki**

Permanent link:
https://wiki.eolab.de/doku.php?id=user:jan001:nigdocu:aufbau&rev=1596450525

Last update: **2021/08/24 17:34**