eDrilling Docs

On-Premise Installation

This describes the hardware and software installation and configuration when customer does not support Kubernetes.

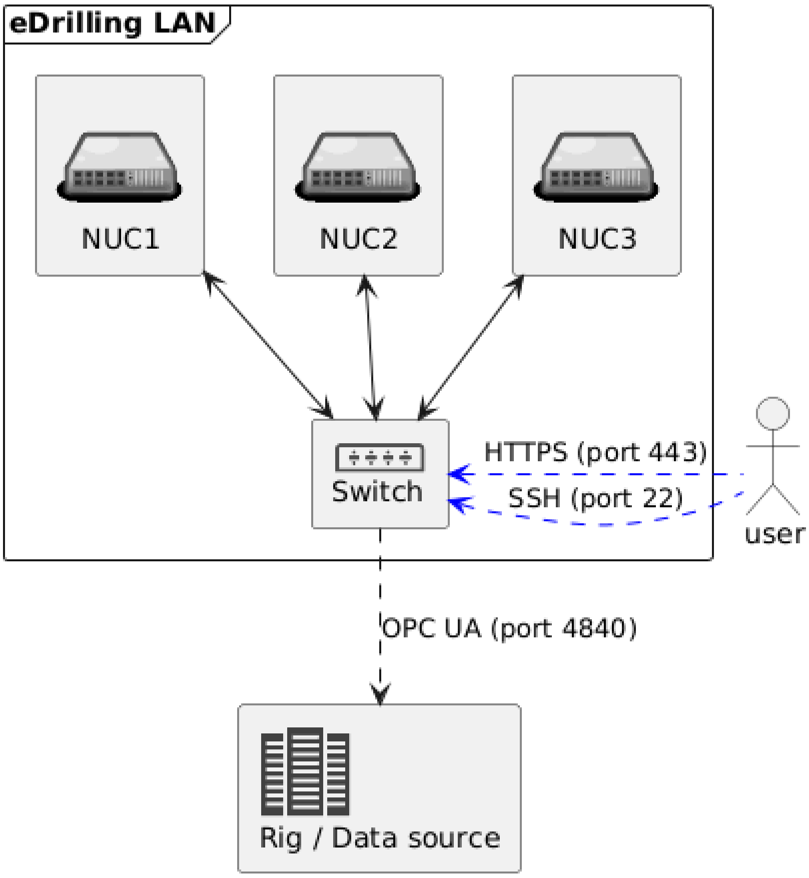

Figure 1. Hardware architecture

1. Installation

NOTE

The work described in this chapter can be done in the eDrilling HQ.

The system cluster will run on:

- 3 x NUC13ANKi7. Buy it here, and include a power cable.
- 3 x Samsung 990 PRO 2TB. Buy it here.
- 6 x Kingston FURY Impact 16GB DDR4 3200MHz SODIMM 260-pin. Buy it here.
- 1 x Netgear GS108E switch

These will have Ubuntu 24.04 installed, and from there an Ansible script will install Microk8s from the Snap Store. The Ansible script will continue to link them together in a cluster.

From there we will run a new script which installs:

1. Longhorn
2. KeyCloak
3. Create wellAhead cluster, by pulling the images from the eDrilling docker repository
4. Precreated HTTPS certificate either from us, or from customer

NOTE

After this process the hardware will be sent to the customer.

2. Concept

One of the NUCs is assigned as the control plane, and the other two are workers. All aliases are set up to point to the control plane, and it’s the only machine handling incoming traffic. Traefik is used as a reverse proxy for accessing services in the kubernetes cluster, and is exposed on port 80/443 on the control plane. In order for traefik to use HTTPS, each externally exposed service needs a certificate. This is currently done by generating a self-signed certificate that is used by traefik.

NOTE

It may be possible to generate a CA certificate that traefik can use to generate certificates for each service, but this has not been tested.

3. Deployment

NOTE

For this part the hardware will be received at the customer site.

The cluster needs to be plugged in to a network where users can access it, and have the following ports open:

1. 22: SSH
2. 443: HTTPS
3. 4840: OPC UA

eDrilling personnel needs to be aware of the following caveats:

1. If the NUC cluster does not have internet access (outbound traffic), a local docker registry should be created.
2. The cluster has been preconfigured with IP addresses and host names, so might need to be re-configured on arrival to a new network.

3.1. Domain

NOTE

NOTE

> It *might* be possible to avoid this section, but then we need to use the IP address to access the cluster, which can be an issue. **We have never tried deploying without a domain.**

A domain must be registered on the company DNS, which is linked to the IP of the main NUC server. This is an example of our own setup:

```
cluster1.edrilling -> <CONTROL_PLANE_IP>
auth.cluster1.edrilling (alias to cluster1.edrilling) -> <CONTROL_PLANE_IP>
minio.cluster1.edrilling (alias to cluster1.edrilling) -> <CONTROL_PLANE_IP>
```

3.2. Certificate

To allow web browsers to accept this domain, we currently use self-signed certificates, and add these manually to the client computers.

As an example, we've used the following to generate self-signed certificates:

```
#!/bin/bash
git clone https://github.com/jsha/minica.git
go run minica/main.go --domains cluster1.edrilling,*.cluster1.edrilling
```

This generates a root certificate, and a leaf certificate for the supplied domains. The leaf certificate is then used by traefik to serve HTTPS.

3.3. Using a preconfigured router

By using a preconfigured router (instead of a switch) at eDrilling HQ, we might solve configuration issues on arrival on-premise:

1. NUCs can keep the same IP address and don't loose its the cluster connection.
2. Router can be pre-configured with domains and certificates, but then only computers connected to this domain will get access with HTTPS.