

# Setting up ESS Pro Helm Chart

## Getting started

This readme is primarily aimed as a simple walkthrough to setup ESS Pro. Users experienced with Helm and Kubernetes can refer directly to the chart README in [element's charts](#).

## Resource requirements

The quick setup relies on K3s. It requires at least 2 CPU cores and 2 GB of memory available.

## Prerequisites

You first need to choose what your server name is going to be. The server name makes up the latter part of a user's Matrix ID. In the following example Matrix ID, `server-name.tld` is the server name, and should point to your ESS Pro installation:

```
@alice:server-name.tld
```

**It is currently not possible to change your server name without resetting your database and having to recreate the server.**

## Quick setup

Setting up a basic environment involves only **6 steps**:

1. [Setting up DNS entries](#)
2. [Setting up K3s](#) (or use another Kubernetes distribution)
3. [Setting up TLS/certificates](#)
4. [Installing the stack](#)
5. [Creating an initial user](#)
6. [Verifying the setup](#)

The below instructions will guide you through each of the steps.

# Preparing the environment

## DNS

You need to create DNS entries to set up ESS Pro. All of these DNS entries must point to your server's IP.

- Server name: This DNS entry should point to the installation ingress. It should be the `server-name.tld` you chose above.
- Synapse: For example, you could use `matrix.<server-name.tld>`.
- Matrix Authentication Service: For example, you could use `account.<server-name.tld>`.
- Matrix RTC Backend: For example, you could use `mrtc.<server-name.tld>`.
- Element Web: This will be the address of the chat client of your server. For example, you could use `chat.<server-name.tld>`.

## Ports

For this simple setup you need to open the following ports :

- TCP 80: This port will be used for the HTTP connections of all services, which will redirect to the HTTPS connection.
- TCP 443: This port will be used for the HTTPS connections of all services.
- TCP 30881: This port will be used for the TCP WebRTC connections of Matrix RTC Backend.
- UDP 30882: This port will be used for the Muxed WebRTC connections of Matrix RTC Backend.

## K3s - Kubernetes single node setup

This guide suggests using K3s as the Kubernetes node hosting ESS Pro. Other options are possible. You can use an existing Kubernetes cluster, or use other clusters like [microk8s](#). Any Kubernetes distribution is compatible with Element Pro, so choose one according to your needs. Please raise with your support or account manager if you discover issues or incompatibilities.

The following will install K3s on the node, and configure its Traefik proxy automatically. If you want to configure K3s behind an existing reverse proxy on the same node, please see the [dedicated section](#).

If you have a firewall running on your server, please follow [k3s official recommendations](#).

1. Run the following command to install K3s:

```
curl -sfL https://get.k3s.io | sh -
```

2. Once K3s is set up, copy its kubeconfig to your home directory to get access to it:

```
mkdir ~/.kube
export KUBECONFIG=~/.kube/config
sudo k3s kubectl config view --raw > "$KUBECONFIG"
chmod 600 "$KUBECONFIG"
chown "$USER:$USER" "$KUBECONFIG"
```

3. Add `export KUBECONFIG=~/.kube/config` to `~/.bashrc` to make it persistent
4. Install Helm, the Kubernetes Package Manager. You can use your [OS repository](#) or call the following command:

```
curl -fsSL https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3 | bash
```

5. Create your Kubernetes namespace where you will deploy the Element Server Suite Pro:

```
kubectl create namespace ess
```

6. Create a directory containing your Element Server Suite configuration values:

```
mkdir ~/ess-config-values
```

## Logging in Element's registry

You can use the following command to log in to the Element's registry. Use your ESS Credentials issued in your EMS Admin Dashboard, under [On Premise Subscriptions](#).

```
helm registry login registry.element.io
```

## Downloading ESS Pro example values files

You can find the example configuration values files in helm chart archive. To download the example values files, you can use the following command:

```
helm pull oci://registry.element.io/matrix-stack --untar -d charts
```

You can find the example configuration values files in the `charts/matrix-stack/ci` directory.

# Configuring image pull authentication

ESS Pro images are hosted on the private element registry. To use these images, you need to configure your authentication tokens. Copy the file from `charts/matrix-stack/ci/fragments/ess-credentials.yaml` to `ess-credentials.yaml` in your ESS configuration values directory. Adjust the values according to your credentials.

## Certificates

We present here 3 options to set up certificates in Element Server Suite. To configure Element Server Suite behind an existing reverse proxy already serving TLS, you can [jump to the end of this section](#).

## Let's Encrypt

To use Let's Encrypt with ESS Pro, you should use [Cert Manager](#). This is a Kubernetes component which allows you to get certificates issued by an ACME provider. The installation follows the [official manual](#):

1. Add Helm Jetstack repository:

```
helm repo add jetstack https://charts.jetstack.io --force-update
```

2. Install Cert-Manager:

```
helm install \
  cert-manager jetstack/cert-manager \
  --namespace cert-manager \
  --create-namespace \
  --version v1.17.0 \
  --set crds.enabled=true
```

3. Configure Cert-Manager to allow ESS Pro to request Let's Encrypt certificates automatically. Create a "ClusterIssuer" resource in your K3s node to do so:

```
export USER_EMAIL=<your email>
```

```
kubectl apply -f - <<EOF
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: letsencrypt-prod
spec:
  acme:
    server: https://acme-v02.api.letsencrypt.org/directory
    email: $USER_EMAIL
    privateKeySecretRef:
      name: letsencrypt-prod-private-key
    solvers:
      - http01:
          ingress:
            class: traefik
EOF
```

4. In your ESS configuration values directory, copy the file `charts/matrix-stack/ci/fragments/quick-setup-letsencrypt.yaml` to `tls.yaml`.

## Certificate File

### Wildcard certificate

If your wildcard certificate covers both the server-name and the hosts of your services, you can use it directly.

1. Import your certificate file in your namespace using `kubectl`:

```
kubectl create secret tls ess-certificate --namespace ess \
--cert=path/to/cert/file --key=path/to/key/file
```

2. In your ess configuration values directory, copy the file `charts/matrix-stack/ci/fragments/quick-setup-wildcard-cert.yaml` to `tls.yaml`. Adjust the TLS Secret name accordingly if needed.

### Individual certificates

1. If you have a distinct certificate for each of your DNS names, you will need to import each certificate in your namespace using `kubectl`:

```
kubectl create secret tls ess-chat-certificate --namespace ess \
--cert=path/to/cert/file --key=path/to/key/file
```

```
kubectl create secret tls ess-matrix-certificate --namespace ess \
  --cert=path/to/cert/file --key=path/to/key/file
kubectl create secret tls ess-auth-certificate --namespace ess \
  --cert=path/to/cert/file --key=path/to/key/file
kubectl create secret tls ess-mtrc-certificate --namespace ess \
  --cert=path/to/cert/file --key=path/to/key/file
kubectl create secret tls ess-well-known-certificate --namespace ess \
  --cert=path/to/cert/file --key=path/to/key/file
```

2. In your ess configuration values directory, copy the file `charts/matrix-stack/ci/fragments/quick-setup-certificates.yaml` to `tls.yaml`. Adjust the TLS Secret name accordingly if needed.

## Using an existing reverse proxy

1. If the certificates are handled in an external load balancer, you can disable TLS in ESS. Copy the file `charts/matrix-stack/ci/fragments/quick-setup-external-cert.yaml` to `tls.yaml`.

# Configuring the database

You can either use the database provided with ESS Pro or you use a dedicated PostgreSQL Server. We recommend using a PostgreSQL server installed with your own distribution packages. For a quick set up, feel free to use the internal PostgreSQL database. The chart will configure it automatically for you by default.

# Installation

The ESS Pro installation is performed using Helm package manager, which requires configuration of a values file as specified in this documentation.

## Setting up the stack

For a quick setup using the default settings, copy the file from `charts/matrix-stack/ci/fragments/quick-setup-hostnames.yaml` to `hostnames.yaml` in your ESS configuration values directory and edit the hostnames accordingly.

Run the setup using the following helm command. This command supports combining multiple values files depending on your setup. Typically you would pass to the command line a combination of:

- If using Lets Encrypt or Certificate Files : `--values ~/ess-config-values/tls.yaml`
- If using your own PostgreSQL server : `--values ~/ess-config-values/postgresql.yaml`

Each optional additional values file used needs to be prefixed with `--values`

To install specific versions, append `:version` after `/matrix-stack`. This is required to stay on the LTS. Without specifying the version you will install the latest available. See [charts.element.io](https://charts.element.io) for a list of available versions.

For example `oci://registry.element.io/matrix-stack:25.4.1` for the April 2025 LTS.

```
helm upgrade --install --namespace "ess" ess \  
oci://registry.element.io/matrix-stack \  
--values ~/ess-config-values/ess-credentials.yaml \  
--values ~/ess-config-values/hostnames.yaml \  
--values ~/ess-config-values/tls.yaml \  
--values <optional additional values files to pass> \  
--wait
```

Wait for the helm command to finish up. ESS Pro is now installed!

## Creating an initial user

ESS Pro does not allow user registration by default. To create your initial user, use the `mas-cli manage register-user` command in the Matrix Authentication Service pod:

```
kubectl exec --namespace ess -it deploy/ess-matrix-authentication-service -- \  
mas-cli manage register-user
```

This should give you the following output:

```
Defaulted container "matrix-authentication-service" out of: matrix-authentication-service, render-config (init),  
db-wait (init), config (init)  
✓ Username · alice  
User attributes  
  ☐ Username: alice  
  ☐ Matrix ID: @alice:thisservername.tld  
No email address provided, user will be prompted to add one
```

No password or upstream provider mapping provided, user will not be able to log in

Non-interactive equivalent to create this user:

```
mas-cli manage register-user --yes alice
```

✓ What do you want to do next? (<Esc> to abort) · Set a password

✓ Password · \*\*\*\*\*

User attributes

☐ Username: alice

☐ Matrix ID: @alice:thisservername.tld

☐ Password: \*\*\*\*\*

No email address provided, user will be prompted to add one

## Allowing users registration

See [the MAS configuration page](#) for details and a configuration example.

## Verifying the setup

To verify the setup, you can:

- Log into your Element Web client website and log in with the user you created above.
- Verify that federation works fine using [Matrix Federation Tester](#).
- Login with an Element X mobile client with the user you created above.
- You can use a Kubernetes UI client such as [k9s \(TUI-Based\)](#) or [lens \(Electron Based\)](#) to see your cluster status.

## Advanced setup

For advanced setup instructions, please refer to the [Advanced setup](#) guide.

## Maintenance

For maintenance topics like upgrading, backups and restoring from backups, please refer to the [Maintenance](#) guide.



# Uninstalling

If you wish to remove ESS Pro from your cluster, you can simply run the following commands to clean up the installation. Please note deleting the `ess` namespace will remove everything within it, including any resources you may have manually created within it:

```
helm uninstall ess --namespace ess
kubectl delete namespace ess
```

If you want to also uninstall other components installed in this guide, you can do so using the following commands:

```
# Remove cert-manager from cluster
helm uninstall cert-manager --namespace cert-manager

# Uninstall helm
rm -rf /usr/local/bin/helm $HOME/.cache/helm $HOME/.config/helm $HOME/.local/share/helm

# Uninstall k3s
/usr/local/bin/k3s-uninstall.sh

# (Optional) Remove config
rm -rf ~/ess-config-values ~/.kube
```

---

Revision #19  
Created 14 May 2025 13:36:33 by Gaël Goinvic  
Updated 28 May 2025 11:49:38 by twilight