

# Kubernetes Installations

## Overview

Our Installer can handle the installation of Element Enterprise into your existing production kubernetes (k8s) environment.

## Server minimum requirements

The ESS deployment resource usage is described in [ESS Sizing](#).

## Prerequisites

Before beginning the installation, there are a few things that must be prepared to ensure a successful deployment and functioning installation.

## Python environment

The installer needs python3, pip3 and python3-venv installed to run.

## Kubectl environment

The installer uses your currently active `kubectl` context which can be determined with `kubectl config current-context` - make sure this is the correct context as all subsequent operations will be performed under this.

More information on configuring this can be found in the [upstream kubectl docs](#)

Be sure to `export K8S_AUTH_CONTEXT=<kube context name>` for the Installer if you need to use a context aside from your currently active one.

## PostgreSQL

Before you can begin with the installation you must have a PostgreSQL database instance available. The installer does not manage databases itself.

The database you use must be set to a locale of `C` and use `UTF8` encoding

see <https://element-hq.github.io/synapse/latest/postgres.html#set-up-database> for further details as they relate to Synapse. If the locale / encoding are incorrect, Synapse will fail to initialize the database and get stuck in a `CrashLoopBackoff` cycle.

Please make note of the database hostname, database name, user, and password as you will need these to begin the installation.

For testing and evaluation purposes, you can deploy PostgreSQL to k8s before you begin the installation process - see [Kubernetes Installations - Quick Start - Deploying PostgreSQL to Kubernetes](#) for more information.

## Kubernetes Ingress Controller

The installer does not manage cluster Ingress capabilities since this is typically a cluster-wide concern - You must have this available prior to installation. Without a working Ingress Controller you will be unable to route traffic to your services without manual configuration.

If you do not have an Ingress Controller deployed please see [Kubernetes Installations - Quick Start - Deploying ingress-nginx to Kubernetes](#) for information on how to set up a bare-bones `ingress-nginx` installation to your cluster.

## Use an existing Ingress Controller

If you have an Ingress Controller deployed already and it is set to the default class for the cluster, you shouldn't have to do anything else.

If you're unsure you can see which providers are available in your cluster with the following command:

```
$ kubectl get IngressClass
```

NAME	CONTROLLER	PARAMETERS	AGE
nginx	k8s.io/ingress-nginx	<none>	40d

And you can check to see whether an IngressClass is set to default using kubectl, for example:

```
$ kubectl describe IngressClass nginx
```

Name: nginx

Labels: app.kubernetes.io/component=controller  
app.kubernetes.io/instance=ingress-nginx  
app.kubernetes.io/managed-by=Helm  
app.kubernetes.io/name=ingress-nginx  
app.kubernetes.io/part-of=ingress-nginx  
app.kubernetes.io/version=1.1.1  
argocd.argoproj.io/instance=ingress-nginx  
helm.sh/chart=ingress-nginx-4.0.17

Annotations: ingressclass.kubernetes.io/is-default-class: true

Controller: k8s.io/ingress-nginx

Events: <none>

In this example cluster there is only an `nginx` IngressClass and it is already default, but depending on the cluster you are deploying to this may be something you must manually set.

# Beginning the Installation

Head to <https://ems.element.io/on-premise/download> and download the latest installer. The installer will be called `element-enterprise-graphical-installer-YYYY-MM.VERSION-gui.bin`. You will take this file and copy it to the machine where you will be installing the Element Server Suite from. This machine will need to be running on RHEL 8, RHEL 9, or Ubuntu and have access to your kubernetes cluster network. Once you have this file on the machine in a directory accessible to your sudo-enabled user, you will run:

```
chmod +x ./element-enterprise-graphical-installer-YYYY-MM.VERSION-gui.bin
```

replacing the `YYYY-MM.VERSION` with the appropriate tag for the installer you downloaded.

If you have multiple kubernetes clusters configured in your kubeconfig, you will have to export the `K8S_AUTH_CONTEXT` variable before running the installer :

```
export K8S_AUTH_CONTEXT=<kube context name>
```

Once you have done this, you will run:

```
./element-enterprise-graphical-installer-YYYY-MM.VERSION-gui.bin
```

replacing the `YYYY-MM.VERSION` with the appropriate tag for the installer you downloaded, and this will start a web server with the installer loaded.

You will see a message similar to:

```
[user@element-demo ~]$ ./element-enterprise-graphical-installer-2023-02.02-gui.bin
Testing network...

Using self-signed certificate with SHA-256 fingerprint:
    F3:76:B3:2E:1B:B3:D2:20:3C:CD:D0:72:A3:5E:EC:4F:BC:3E:F5:71:37:0B:D7:68:36:2E:2C:AA:7A:F2:83:94

To start configuration open:
    https://192.168.122.47:8443 or https://10.1.185.64:8443 or https://127.0.0.1:8443
```

At this point, you will need to open a web browser and browse to one of these IPs. You may need to open port 8443 in your firewall to be able to access this address from a different machine.

If you are unable to open port 8443 or you are having difficulty connecting from a different machine, you may want to try ssh port forwarding in which you would run:

```
ssh <host> -L 8443:127.0.0.1:8443
```

replacing host with the IP address or hostname of the machine that is running the installer. At this point, with ssh connected in this manner, you should be able to use the `https://127.0.0.1:8443` link as this will then forward that request to the installer box via ssh.

Upon loading this address for the first time, you may be greeted with a message informing you that your connection isn't private such as this:



## Your connection isn't private

Attackers might be trying to steal your information from **192.168.122.47** (for example, passwords, messages, or credit cards).

NET::ERR\_CERT\_AUTHORITY\_INVALID

Advanced

Go back

In this case, you'll need to click "Advanced" and then "Continue to (unsafe)" in order to view the installer. As the exact button names and links can vary between browsers, it would be hard for us to document them all, so you may have slightly different wording depending on your browser.

# The Hosts Screen

The very first page that you come to is the host screen.



# Host.

## Deployment

### Install

☐ Standalone ☒ Kubernetes Application

Kube Context Name \*

Name of a Kubernetes context already setup in your kube config to install into

☐ Skip Operator Setup

Default

☐ Skip Updater Setup

Default

### EMS Image Store

Your ems image store username / password

Username

382255a2-8d20-4a4a-aad7-2d3db34b7506

Token

.....



You will want to make sure that "Kubernetes Application" is selected. You will then need to specify the kubernetes context name for which you are deploying into. (Hint: Use `kubectl config view` to see which contexts you have access to.)

You can opt to skip the update setup or the operator setup, but unless you know why you are doing that, you should leave them alone.

The very next prompt that you come to is for an EMS Image Store Username and Token. These are provided to you by element as access tokens for our enterprise container registries. If you have lost your token, you can always generate a new token at <https://ems.element.io/on-premise/subscriptions>.

## Namespaces

☒ Create Namespaces

Create the namespaces or use an existing one

Default

Element Deployment

element-onprem

Default

The namespace where to deploy the element applications

Operator

operator-onprem

Default

The namespace where to deploy the operator controller manager

Updater

updater-onprem

Default

The namespace where to deploy the updater controller manager

## Connectivity

☒ Connected ☐ Airgapped

## Dockerhub

Optionally reduce rate limiting by providing your dockerhub credentials

Username & Password



Here, we find the ability to set the namespaces that the application will be deployed into.

The next options on the hostpage are related to connectivity. For this guide, we are assuming "Connected" and you can leave that be. If you are doing "Airgapped", you would pick airgapped at this point and then please see the section on [airgapped installations](#).

You are presented with the option to provide docker hub credentials. These are optional, but if you do not provide them, you may be rate limited by Docker and this could cause issues pulling container images.

## Host Admin

Host Admin Domain

Default

Domain name to use for this UI when running directly on the host (optional)

### Trusted HTTPS

The certificate used for accessing the host admin panel (this UI when running directly on the host)

- ☒ Self Signed   ☐ Automatic / Let's Encrypt   ☐ Certificate File
- ☐ Existing TLS Certificates in the Cluster   ☐ Externally Managed

Use an automatically generated self-signed certificate.

Finally, we come to the host admin page, which allows you to set parameters around which domain the installer and admin console should run on post deployment. This section is optional.

## The Domains Screen





# Domains.

Domain Name \*

element.demo

The domain name of this deployment. It will be used for the <localpart> of the users MXIDs, and cannot be changed afterwards. For example: @user:element.demo

Full-Qualified Domain Name \*

hs

.element.demo X

Fully qualified domain name of the ingress

Element Web Domain \*

web

.element.demo X

Fully qualified domain name of the ingress

Synapse Admin Domain \*

admin

.element.demo X

Fully qualified domain name of the ingress

Integrator Domain \*

integrator

.element.demo X

Fully qualified domain name of the ingress

[Previous](#)

[Continue](#)

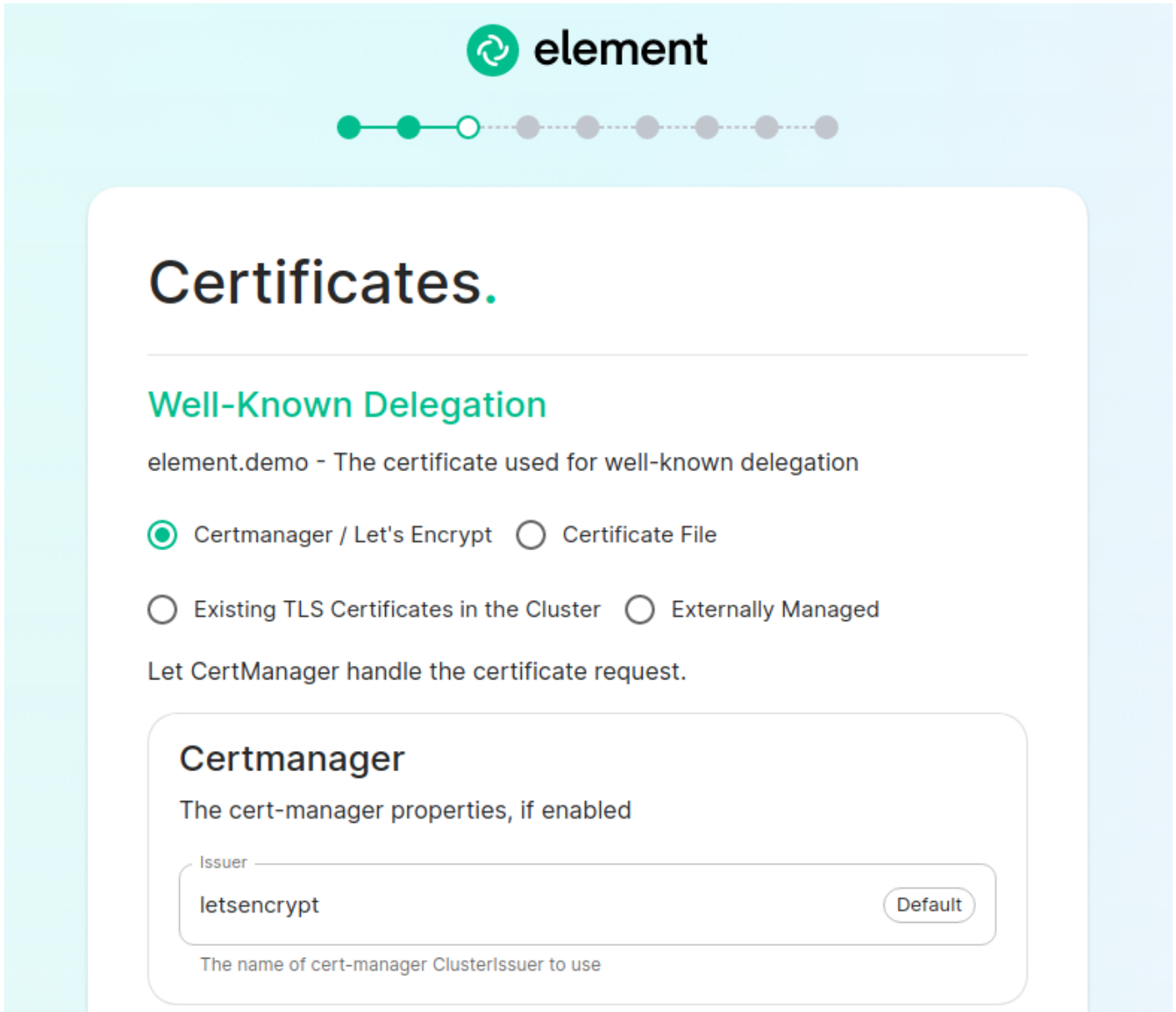
On this page, we get to specify the domains for our installation. In this example, we have a domain name of airgap.local and this would mean our MXIDs would look like @kabbott:airgap.local.

Our domain page has checking to ensure that the host names resolve. Once you get green checks across the board, you can click continue.

# The Certificates Screen

On the Certificates screen, you will provide SSL certificate information for well-known delegation, Synapse, Element Web, Synapse Admin, and Integrator.

If you are using Let's Encrypt, then each of the sections should look like:



The screenshot shows the 'Certificates' configuration screen for Element. At the top, there is a header with the 'element' logo and a progress bar with eight steps; the third step is highlighted. The main section is titled 'Certificates.' and contains a sub-section 'Well-Known Delegation'. Under this sub-section, the text 'element.demo - The certificate used for well-known delegation' is displayed. There are four radio button options: 'Certmanager / Let's Encrypt' (selected), 'Certificate File', 'Existing TLS Certificates in the Cluster', and 'Externally Managed'. Below these options, the text 'Let CertManager handle the certificate request.' is shown. A rounded rectangle contains the 'Certmanager' section, which includes the text 'The cert-manager properties, if enabled'. Inside this section, there is a text input field for 'Issuer' with the value 'letsencrypt' and a 'Default' button. A small note at the bottom of this section states 'The name of cert-manager ClusterIssuer to use'.

**element**

**Certificates.**

**Well-Known Delegation**

element.demo - The certificate used for well-known delegation

☒ Certmanager / Let's Encrypt ☐ Certificate File

☐ Existing TLS Certificates in the Cluster ☐ Externally Managed

Let CertManager handle the certificate request.

**Certmanager**

The cert-manager properties, if enabled

Issuer  Default

The name of cert-manager ClusterIssuer to use

If you are using certificate files, then you will see a screen like:

## Well-Known Delegation

element.demo - The certificate used for well-known delegation


- ☐ Certmanager / Let's Encrypt ☒ Certificate File
- ☐ Existing TLS Certificates in the Cluster ☐ Externally Managed

Upload a certificate and its private key.

### Certificate

Certificate file

Secrets / Well Known Delegation /


Well Known Delegation Certificate 

### Certificate

WellKnownDelegation Certificate

Replace Uploaded File

Secrets / Well Known Delegation /

Well Known Delegation Private Key 

### Private key

WellKnownDelegation Private Key

Replace Uploaded File

which allows you to upload a .crt and .key file for each host. These files must be in PEM encoding. Our installer does accept wildcard certificates.

Once you have completed the certificate section for each host on the page, you may click continue.

# The Database Screen

## Database.

### PostgreSQL

Configuration of Postgres database

Database \*

Please fill out this field.

PostgreSQL database name

Host \*

PostgreSQL database host

Port

5432

Default

PostgreSQL port

SSL Mode

Require

Default

TLS settings to use for the Postgres connection

User \*

PostgreSQL username

Secrets / Synapse / Postgres Password

Postgresql Password

The postgres password

As you must use an external PostgreSQL database with our kubernetes install, on this page, we provide the option to specify the database name, the database host name, the port to connect to, the SSL mode to use, and finally, the username and password to connect with. Once you have completed this section, you may click continue.

# The Media Screen

On this page, you can specify the size of your synapse media volume. Please leave "Create New Volume" checked and specify the size of the volume that you wish to allocate. You must have this space available in `/data/element-deployment` or whatever you specified back on the hosts screen. If you wish to create a 50G volume, you would need to specify `50Gi` for the Volume size.


# The Cluster Screen

Most deployments can ignore this, however, if you want to change any kubernetes cluster parameters, this is where to do it.

If you are in an environment where you have self-signed certificates, you will want to disable TLS verification, by clicking "Advanced" and then scrolling down and unchecking `Verify TLS` :

☐ Verify TLS


TLS verification

Secrets / Global / CA.pem 


Certificate authority

The CA to inject into the deployment.

Upload File

Secrets / Global / Generic Shared Secret 

Generic Shared Secret

..... 

The generic shared secret to use as a seed for all internally-generated secrets

Please bear in mind that disabling TLS verification and using self-signed certificates is not recommended for production deployments.

If your host names are not DNS resolvable, you need to use host aliases and this can be set up here. You will also click "Advanced" and scroll down to the "Host Aliases" section in "k8s". In here, you will click "Add Host Aliases" and then you will specify an IP and host names that resolve to that IP as such:

## Host Aliases

The list of hosts aliases to configure on the pod spec. It should be avoid as much as possible to use this feature. Please prefer using an DNS entry to resolve your hostnames. This can be used as a workaround when entries cannot be resolved using DNS, for example for our automated testings.

IP \*

192.168.122.121

An IP resolution to add to /etc/hosts

Hostnames

An hostname of the associated ip to add to /etc/hosts

element.demo

An hostname of the associated ip to add to /etc/hosts

hs.element.demo

An hostname of the associated ip to add to /etc/hosts

web.element.demo

An hostname of the associated ip to add to /etc/hosts

admin.element.demo

An hostname of the associated ip to add to /etc/hosts

integrator.element.demo

An hostname of the associated ip to add to /etc/hosts

metrics.element.demo

Add more Hostnames

**Important:** If you are not using OpenShift, you will need to set "Force UID GID" and "Set Sec Comp" to "Enable" under the section "Security Context" so that it looks like:

## Security Context

Force UID GID

Enable

Edited

Enable pod runAsUser and fsGroup in security context. Disable if it should not be used, in the case of openshift for example. Auto attempts to detect openshift automatically.

Set Sec Comp

Enable

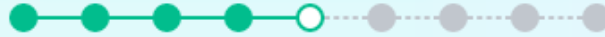
Edited

Enable RuntimeDefault pod seccomp. disable if it should not be used, in the case of openshift for example. Auto attempts to detect openshift automatically.

If you are using OpenShift, you should leave the values of "Force UID GID" and "Set Sec Comp" set to "Auto".

When you are finished with this page, you can click continue.

## The Synapse Screen



# Synapse.

This is a matrix homeserver.

## Profile

Monthly Active Users

How many users actively use your server?

Federation Type

Closed: No Federation Limited: Federation within trusted network Open: Federation with all

## Config

Accept Invites

Manual

Whether to enable auto accept invites. Defaults to manual if not set

Max MAU Users

250

Maximum number of Matrix Active Users

Registration

Closed

Synapse registration

The first setting that you will come to is our built in performance profiles. Select the appropriate answers for "Monthly Active Users" and "Federation Type" to apply our best practices based on years of running Matrix homeservers.

The next setting that you will see is whether you want to auto accept invites. The default of "Manual" will fit most use cases, but you are welcome to change this value.

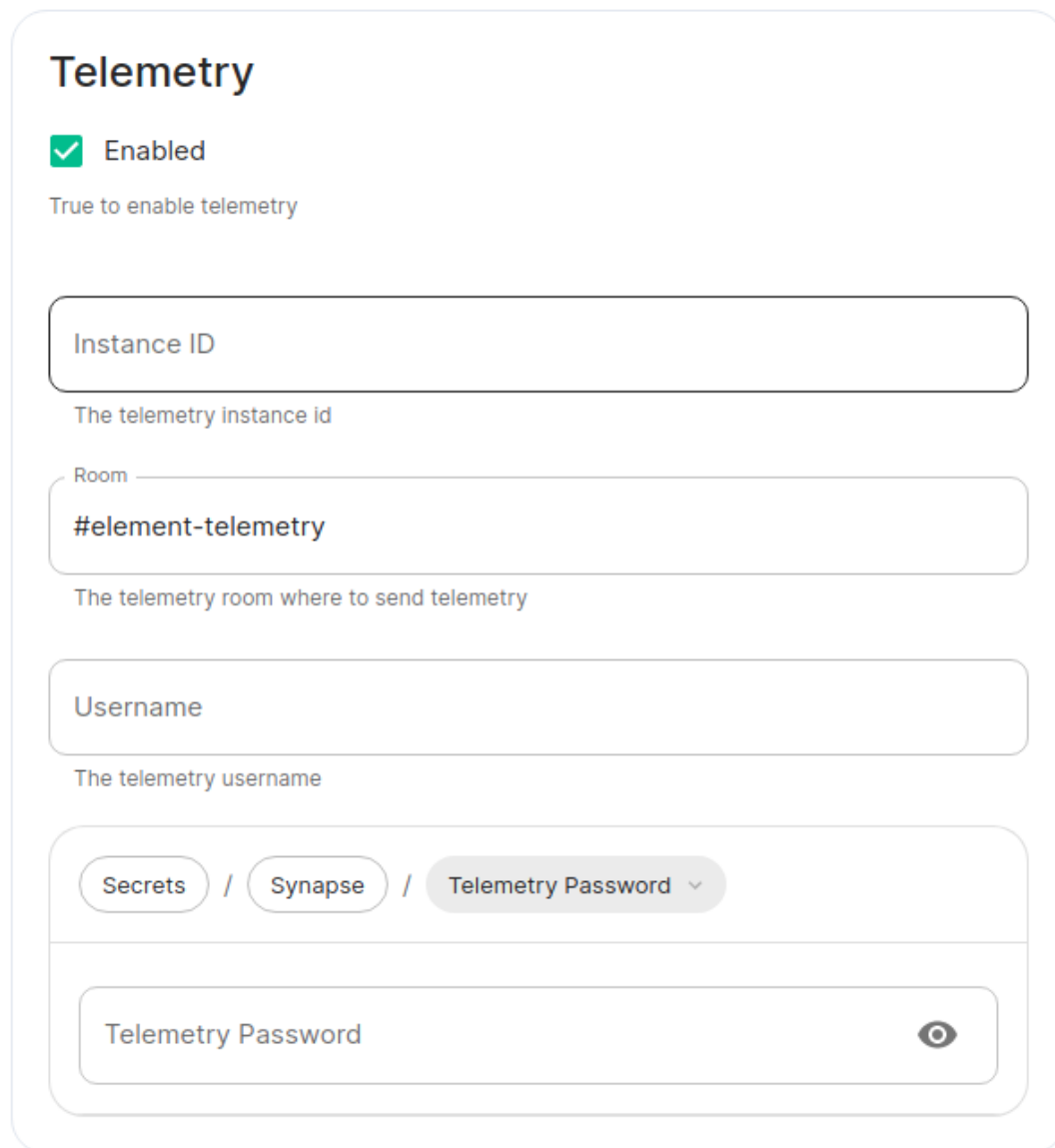
The next setting is the maximum number of monthly active users (MAU) that you have purchased for your server. Your server will not allow you to go past this value. If you set this higher than your purchased MAU and you go



over your purchased MAU, you will need to true up with Element to cover the cost of the unpaid users.

The next setting concerns registration. A server with open registration on the open internet can become a target, so we default to closed registration. You will notice that there is a setting called "Custom" and this requires explicit custom settings in the additional configuration section. Unless instructed by Element, you will not need the "Custom" option and should instead pick "Closed" or "Open" depending on your needs.

After this, you will see that the installer has picked an admin password for you. You will want to use the eye icon to view the password and copy this down as you will use this with the user `onprem-admin-donotdelete` to log into the admin panel after installation.



The screenshot shows the 'Telemetry' configuration section of the Element installer. It features a title 'Telemetry' and a toggle switch labeled 'Enabled' with a green checkmark. Below the toggle is the text 'True to enable telemetry'. There are four input fields: 'Instance ID' with the description 'The telemetry instance id', 'Room' with the value '#element-telemetry' and description 'The telemetry room where to send telemetry', 'Username' with the description 'The telemetry username', and 'Telemetry Password' with an eye icon for toggling visibility. Above the password field is a breadcrumb trail: 'Secrets / Synapse / Telemetry Password' with a dropdown arrow.

**Telemetry**


☒ Enabled  
True to enable telemetry

Instance ID  
The telemetry instance id

Room  
#element-telemetry  
The telemetry room where to send telemetry

Username  
The telemetry username

Secrets / Synapse / Telemetry Password ▾

Telemetry Password 

Continuing, we see telemetry. You should leave this enabled as you are required to report MAU to Element. In the event that you are installing into an environment without internet access, you may disable this so that it does not continue to try talking to Element. That said, you are still required to generate an MAU report at regular intervals and share that with Element.

For more information on the data that Element collects, please see: [What Telemetry Data is Collected by Element?](#)

Next, we have an advanced button, which allows you to configure further settings about synapse and the kubernetes environment in which it runs. The additional configuration text box allows you to inject additional

synapse configs ([homeserver.yaml](#)).

## Additional

Additional config to inject

1

Delegated Auth ☐

Stun ☐

Identity Server ☐

HTTP Proxy ☐

Advanced ▾

Previous Continue

You can hit continue to go to the next screen.

## The Element Web Screen

Most users will be able to simply click "Continue" here.

The Advanced section allows you to set any custom element web configurations you would like ([config.json](#)).

## Config

### Additional configuration

Element web additional configuration.

1

### K8s

Force values for components of Element Web

Show

Previous

Continue

A common custom configuration would be configuring permalinks for Element, which we have documented here: [Setting up Permalinks With the Installer](#)

Further, it provides access to the k8s section, allowing you to explicitly set any kubernetes cluster settings that you would like just for the element-web pod.

# The Homeserver Admin

Most users will be able to simply click "Continue" here. The Advanced section allows you to explicitly set any kubernetes cluster settings that you would like just for the synapse-admin-ui pod.

One word to note here is that if you are not using delegated authentication, then the initial username that an administrator will use to log into this dashboard post-installation is `onprem-admin-donotdelete`. You can find the password for this user on the Synapse page in the "Admin Password" field.

If you are using delegated authentication, you will need to assign a user admin rights as detailed in this article: [How do I give a user admin rights when I am using delegated authentication and cannot log into the admin console?](#)

# The Integrator Screen

# Integrator.

Send messages to external services

## Config

☒ Enable Custom Widgets

Enable custom widgets in Appstore

Verify TLS

Use Global Setting

Default ▼

TLS Verification

## Log

Logging settings

Level

Info

Default ▼

The maximum level of log output

☐ Structured

Output logs in logstash format. Otherwise, logs are output in a console friendly format.

Default

## Jitsi Domain



Manually configure an external jitsi domain. Will use the installer deployed one if not set.

On this page, you can set up Integrator, the integrations manager.

The first option allows you to choose whether users can add custom widgets to their rooms with the integrator or not.

The next option allows you to specify which Jitsi instance the Jitsi widget will create conferences on.

The verify TLS option allows you to set this specifically for Integrator, regardless of what you set on the cluster screen.

The logging section allows you to set the log level and whether the output should be structured or not.

The Advanced section allows you to explicitly set any kubernetes cluster settings that you would like just for the integrator pods.

Click "Continue to go to the next screen".

# The Integrations Screen

This screen is where you can install any available integrations.

Some of these integrations will have "YAML" next to them. When you see this designation, this integration requires making settings in YAML, much like the old installer. However, with this installer, these YAML files are pre-populated and often only involve a few changes.

If you do not see a "YAML" designation next to the integration then this means that will use regular GUI elements to configure this integration.

Over time, we will do the work required to move the integrations with "YAML" next to them to the new GUI format.

For specifics on configuring well known delegation, please see [Setting Up Well Known Delegation](#)

For specifics on setting up Delegated Authentication, please see [Setting up Delegated Authentication With the Installer](#)

For specifics on setting up Group Sync, please see [Setting up Group Sync with the Installer](#)

For specifics on setting up GitLab, GitHub, and JIRA integrations, please see [Setting up GitLab, GitHub, and JIRA Integrations With the Installer](#)

For specifics on setting up Adminbot and Auditbot, please see: [Setting up Adminbot and Auditbot](#)

For specifics on setting up Hydrogen, please see: [Setting Up Hydrogen](#)

For specifics on pointing your installation at an existing Jitsi instance, please see [Setting Up Jitsi and TURN With the Installer](#)

If you do not have an existing TURN server or Jitsi server, our installer can configure these for you by following the extra steps in [Setting Up Jitsi and TURN With the Installer](#)

For specifics on configuring the Teams Bridge, please see [Setting Up the Teams Bridge](#)

For specifics on configuring the Telegram Bridge, please see [Setting Up the Telegram Bridge](#)

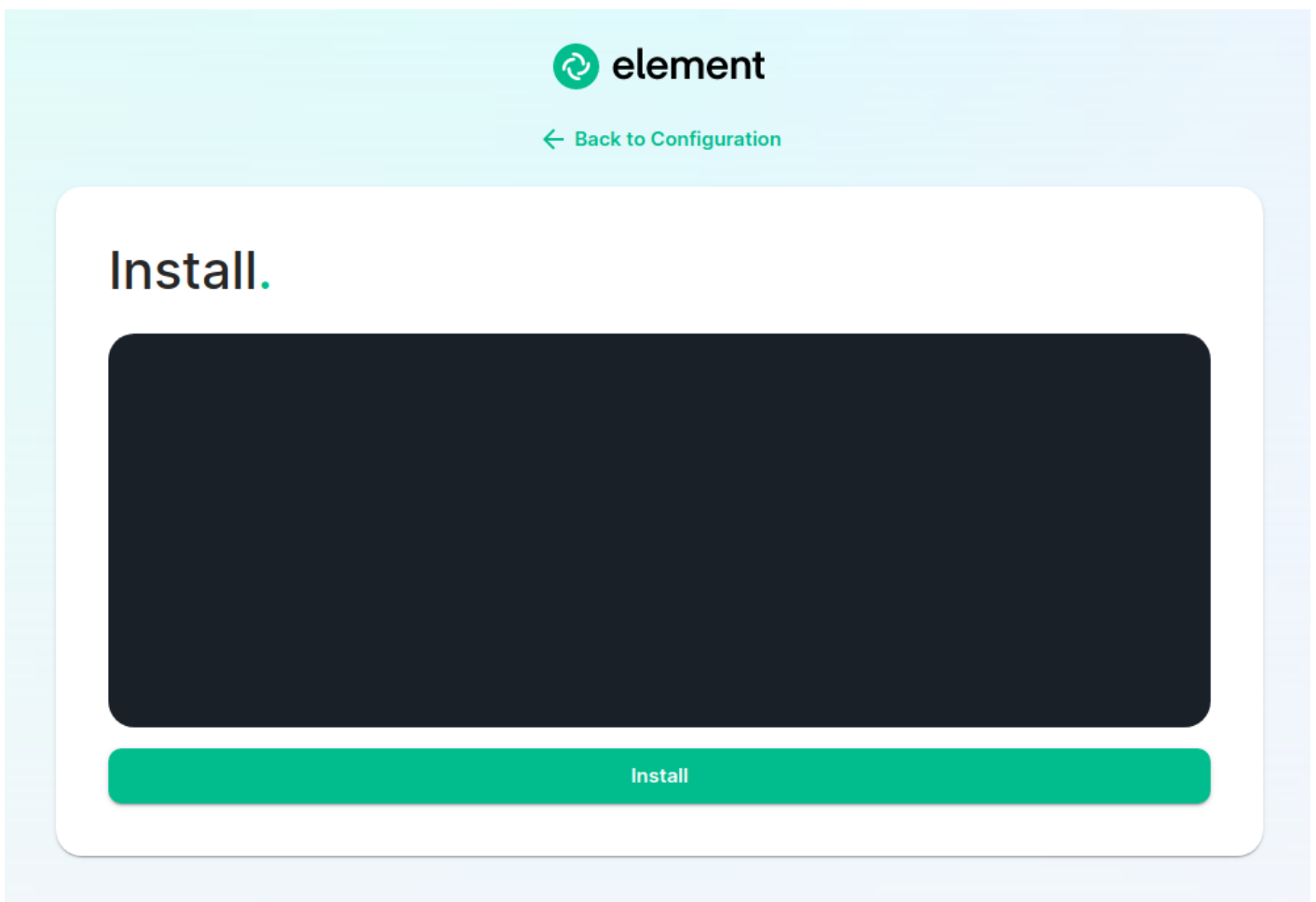
For specifics on configuring the IRC Bridge, please see [Setting Up the IRC Bridge](#)

For specifics on configuring the XMPP Bridge, please see [Setting Up the XMPP Bridge](#)

Once you have configured all of the integrations that you would like to configure, you can click "Continue" to head to the installation screen.

# The Installation Screen

On the installation screen, you should see a blank console and a start button:



Click Start.

After a moment, you will notice the installer hang. If you go back to the prompt where you are running the installer, you will see that you are being asked for the sudo password:



```

cachetools, requests-oauthlib, requests, pyasn1-modules, pyyaml, jmespath, google-auth,
canonicaljson, ansible-core, wheel, signedjson, pyopenssl, psutil, pkgutil-resolve-name, openapi-
schema-validator, netaddr, kubernetes, jmespath, importlib-resources, ansible
Attempting uninstall: setuptools
Found existing installation: setuptools 53.0.0
Uninstalling setuptools-53.0.0:
Successfully uninstalled setuptools-53.0.0
Attempting uninstall: wheel
Found existing installation: wheel 0.41.1
Uninstalling wheel-0.41.1:
Successfully uninstalled wheel-0.41.1
Successfully installed ansible-6.7.0 ansible-core-2.13.9 attrs-23.1.0 cachetools-5.3.1
canonicaljson-2.0.0 certifi-2023.7.22 cffi-1.15.1 charset-normalizer-3.2.0 cryptography-38.0.1
google-auth-2.22.0 idna-3.4 importlib-resources-6.0.0 jinja2-3.1.2 jmespath-1.0.1 jsonschema-4.17.3
kubernetes-25.3.0 markupsafe-2.1.3 netaddr-0.8.0 oauthlib-3.2.0 openapi-schema-validator-0.5.0
packaging-23.1 pkgutil-resolve-name-1.3.10 psutil-5.9.4 pyasn1-0.5.0 pyasn1-modules-0.3.0
pycparser-2.21 pynacl-1.5.0 pyopenssl-23.2.0 pyrsistent-0.19.3 python-dateutil-2.8.2 pyyaml-6.0.1
requests-2.31.0 requests-oauthlib-1.3.1 resolvelib-0.8.1 rfc3339-validator-0.1.4 rsa-4.9
setuptools-68.0.0 signedjson-1.1.4 six-1.16.0 unpaddedbase64-2.1.0 urllib3-1.26.16 websocket-
client-1.6.1 wheel-0.40.0 zipp-3.16.2
WARNING: You are using pip version 21.2.3; however, version 23.2.1 is available.
You should consider upgrading via the '/home/karl1/.element-enterprise-server/installer/.install-
env/bin/python3 -m pip install --upgrade pip' command.
Starting galaxy collection install process
Nothing to do. All requested collections are already installed. If you want to reinstall them,
consider using '--force'.
sudo: a password is required
ansible-playbook [core 2.13.9]
  config file = None
  configured module search path = ['/home/karl1/.ansible/plugins/modules',
'/usr/share/ansible/plugins/modules']
  ansible python module location = /home/karl1/.element-enterprise-server/installer/.install-
env/lib64/python3.9/site-packages/ansible
  ansible collection location = /home/karl1/.ansible/collections:/usr/share/ansible/collections
  executable location = /home/karl1/.element-enterprise-server/installer/.install-env/bin/ansible-
playbook
  python version = 3.9.14 (main, Jan  9 2023, 00:00:00) [GCC 11.3.1 20220421 (Red Hat 11.3.1-2)]
  jinja version = 3.1.2
  libyaml = True
No config file found; using defaults
Processing...

```

Install

```

[karl1@airgap ~]$ ./element-enterprise-graphical-installer-2023-02.02-gui-rc1.bin
Testing network...

Using self-signed certificate with SHA-256 fingerprint:
CB:8E:4A:75:80:32:D5:E0:A0:2C:90:A7:DF:F9:2F:9F:6D:14:F7:18:53:D0:C5:6C:20:D5:95:A8:1A:57:67:21

To start configuration open:
https://192.168.122.47:8443 or https://10.1.185.64:8443 or https://127.0.0.1:8443
API resolved without sending a response for /api/logs, this may result in stalled requests.
[sudo] password for karl1: 

```

Go ahead and enter the sudo password and the installation will continue.

On the very first time that you run the installer, you will be prompted to log out and back in again to allow Linux group membership changes to be refreshed. This means that you will need to issue a ctrl-C in the terminal running your installer and actually log all the way out of your Linux session, log back in, restart the installer, navigate back to the installer screen, click start again, and then re-enter your sudo password. You will only have

to perform this step once per server.

# Verifying Your Installation

Once the installation has finished, it can take as much as 15 minutes on a first run for everything to be configured and set up. If you use:

```
kubectl get pods -n element-onprem
```

You should see similar output to:

NAME	READY	STATUS	RESTARTS	AGE
app-element-web-c5bd87777-rqr6s	1/1	Running	1	29m
server-well-known-8c6bd8447-wddtm	1/1	Running	1	29m
postgres-0	1/1	Running	1	40m
instance-synapse-main-0	1/1	Running	2	29m
instance-synapse-haproxy-5b4b55fc9c-hnlmp	1/1	Running	0	20m

Once the admin console is up and running:

first-element-deployment-synapse-admin-ui-564cbf5665-dn8nv	1/1	Running	1 (4h4m ago)	3d1h
--	-----	---------	--------------	------

and synapse:

first-element-deployment-synapse-redis-59548698df-gqkcq	1/1	Running	1 (4h4m ago)	3d2h
first-element-deployment-synapse-haproxy-7587dfd6f7-gp6wh	1/1	Running	2 (4h3m ago)	2d23h
first-element-deployment-synapse-appservice-0	1/1	Running	3 (4h3m ago)	3d
first-element-deployment-synapse-main-0	1/1	Running	0	3h19m

then you should be able to log in at your admin panel (in our case <https://admin.airgap.local/>) with the `onprem-admin-donotdelete` user and the password that was specified on the "Synapse" screen.

## A word about Configuration Files

In the new installer, all configuration files are placed in the directory `.element-enterprise-server`. This can be found in your user's home directory. In this directory, you will find a subdirectory called `config` that contains the actual configurations.

# Running the Installer without the GUI

It is possible to run the installer without using the GUI provided that you have a valid set of configuration files in the `.element-enterprise-server/config` directory. Directions on how to do this are available at: <https://ems-docs.element.io/books/ems-knowledge-base/page/how-do-i-run-the-installer-without-using-the-gui>. Using this method, you could use the GUI as a configuration editor and then take the resulting configuration and modify it as needed for further installations.

This method also makes it possible to set things up once and then run future updates without having to use the GUI.

## End-User Documentation

After completing the installation you can share our [User Guide](#) to help orient and onboard your users to Element!

---

Revision #47

Created 1 August 2022 18:32:51 by Karl Abbott

Updated 6 November 2024 13:19:55 by Kieran Mitchell Lane