Introduction to Element Server Suite

What is Element Server Suite?

Element Server Suite provides an enterprise-grade secure communications platform that can be run either on your own premise or in our Element Cloud. Element Server Suite includes the Element Matrix Server, which provides a host of security and privacy features, including:

- Built on the Matrix open communications standard.
- Provides end to end encrypted messaging, voice, and video through a consumer style messenger with the power of a collaboration tool.
- Delivers data sovereignty.
- Affords a high degree of flexibility that can be tailored to many use cases.
- Allows secure federation within a single organisation or across a supply chain or ecosystem.

and combines them with the following Element Server Extensions:

- Group Sync: Synchronize group data from your identity provider and map these into Element spaces.
- Adminbot: Give your server administrator the ability to be admin in any rooms on your homeserver.
- Auditbot: Have an auditable record of conversations conducted on your homeserver.
- Security and feature updates: Updates are easy to deploy and handled by our installer.
- Bridges: Bridge to IRC, XMPP, Telegram, Microsoft Teams, or SIP. More coming soon.

Further, we also offer Enterprise Support, giving you access to the experts in federated, secure communications giving you confidence to deploy our platform for your most critical secure communications needs.

Given the flexibility afforded by this platform, there are a number of moving parts to configure. This documentation will step you through architecting and deploying Element Enterprise On-Premise.

Deploying to Kubernetes or to a standalone server?

Element Enterprise On-Premise can be deployed both to Kubernetes (a lightweight container orchestration platform) or a standalone server. One key benefit of going with Kubernetes is that you can add more resources and nodes to a cluster as you need them where you are capped at one node with our standalone server. In the case of our standalone server installation, we deploy microk8s (a smaller lightweight distribution of Kubernetes), which we then use for deploying our application.

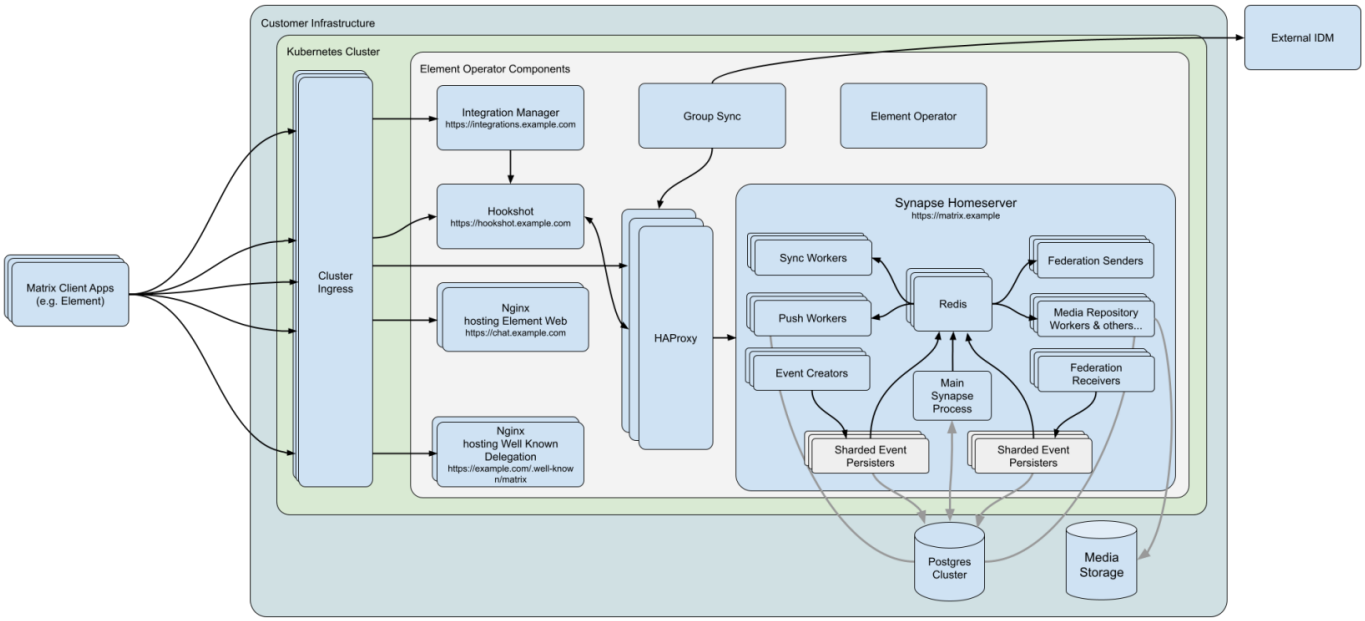
In general, regardless of if you pick the standalone server or Kubernetes deployment, you will need a base level of hardware to support the application.

For scenarios that utilise open federation, Element recommends a minimum of 8 vCPUs/CPUs and 32GB ram for the host(s) running synapse pods.

For scenarios that utilise closed federation, Element recommends a minimum of 6 vCPUs/CPUs and 16GB ram for the host(s) running synapse pods.

Architecture

This document gives an overview of our secure communications platform architecture:



(Please click on the image to view it at 100%.)

Comprising our secure communications platform are the following components:

- synapse : The homeserver itself.
- element-web : The Element Web client.
- integrator: Our integration manager.
- synapse admin ui : Our Element Enterprise Administrator Dashboard.
- postgresql (Optional) : Our database. Only optional if you already have a separate PostgreSQL database, which is required for a multiple node setup. Use an external DB if you have more than 300 users.
- groupsync (Optional) : Our group sync software
- adminbot (Optional) : Our bot for admin tasks.
- auditbot (Optional) : Our bot that provides auditability.
- hookshot (Optional) : Our integrations with gitlab, github, jira, and custom webhooks.
- hydrogen (Optional) : A light weight alternative chat client.
- jitsi (Optional) : Our VoIP platform for group conferencing.
- coturn (Optional) : TURN server. Required if deploying VoIP.
- element-call (Optional) : Our new VoIP plaftorm for group conferencing
- sfu (Optional) : Element Call LiveKit component for scalable conferencing
- prometheus (Optional) : Provides metrics about the application and platform.
- grafana (Optional) : Graphs metrics to make them consumable.
- telegram bridge (Optional) : Bridge to connect Element to Telegram.
- teams bridge (Optional) : Bridge to connect Element to MS Teams.

- xmpp bridge (Optional) : Bridge to connect Element to XMPP.
- irc bridge (Optional) : Bridge to connect Element to IRC.
- sip bridge (Optional) : Bridge to connect Element to SIP.

For each of the components in this list (excluding postgresql, groupsync, adminbot, auditbot, and prometheus), you must provide a hostname on your network that meets this criteria:

- Fully resolvable to an IP address that is accessible from your clients.
- Signed PEM encoded certificates for the hostname in a crt/key pair. Certificates should be signed by an internet recognised authority, an internal to your company authority, or LetsEncrypt.

It is possible to deploy Element Enterprise On-Premise with self-signed certificates and without proper DNS in place, but this is not ideal as the mobile clients and federation do not work with self-signed certificates.

Information on how to use self-signed certificates and hostname mappings instead of DNS can be found in How to Setup Local Host Resolution Without DNS

In addition to hostnames for the above, you will also need a hostname and PEM encoded certificate key/cert pair for your base domain. If we were deploying a domain called example.com and wanted to deploy all of the software, we would have the following hostnames in our environment that needed to meet the above criteria:

- example.com (base domain)
- matrix.example.com (the synapse homeserver)
- element.example.com (element web)
- integrator.example.com (integration manager)
- admin.example.com (admin dashboard)
- hookshot.example.com (Our integrations)
- hydrogen.example.com (Our light weight chat client)
- jitsi.example.com (Our VoIP platform)
- coturn.example.com (Our TURN server)
- grafana.example.com (Our Grafana server)
- telegrambridge.example.com (Our Telegram Bridge)
- teamsbridge.example.com (Our Teams Bridge)
- roomadmin.example.com (AdminBot)
- audit.example.com (Audit capablility)
- element-call.example.com (Our new VoIP platform)
- sfu.example.com (Scaling for Element Call)

As mentioned above, this list excludes postgresql, groupsync, adminbot, auditbot, and prometheus.

Wildcard certificates do work with our application and it would be possible to have a certificate that validated *.example.com and example.com for the above scenario. It is key to do both the base domain and the wildcard in the same certificate in order for this to work.

Further, if you want to do voice or video across network boundaries (ie: between people not on the same local network), you will need a TURN server. If you already have one, you do not have to set up coturn. If you do not already have a TURN server, you will want to set up coturn (our installer can do this for you) and if your server is behind NAT, you will need to have an external IP in order for coturn to work.

Installation

Airgapped Environments

If you are going to be installing into an airgapped environment (one without internet connectivity), you will need to also download the airgapped installer, which is ~6GB of data that will need to be transferred to your airgapped environment. More information on this can be found in our airgapped installation documentation here: https://ems-docs.element.io/books/element-on-premise-documentation/page/using-the-installer-in-an-air-gapped-environment

Software

To obtain our software, please visit our downloads page at: https://ems.element.io/on-premise/download

Kubernetes Application (Multiple Nodes)

For an installation into a kubernetes environment, make sure you have a Kubernetes platform deployed that you have access to and head over to Kubernetes Installations. You will also need a linux computer to run the installer from. That computer should be running RHEL 8 or RHEL 9 or Ubuntu.

Standalone (Single Node)

For a standalone installation, please note that we support these on the following platforms based on the ESS version

LTS ESS Version	Supported Ubuntus	Supported Enterprise Linux (RHEL, Rocky, etc)	General Python Version requirements
23.10	20.04, 22.04	8, 9	Python 3.8-3.10

LTS ESS Version	Supported Ubuntus	Supported Enterprise Linux (RHEL, Rocky, etc)	General Python Version requirements
24.04	20.04, 22.04	8, 9	Python 3.9-3.11
24.10	22.04, 24.04	8, 9	Python 3.10-3.12

Once you have a server with one of these installed, please head over to Single Node Installations

---

Revision #46
Created 2022-07-25 17:44:50 UTC by Karl Abbott
Updated 2024-11-06 17:21:57 UTC by Kieran Mitchell Lane