HCLSoftware

Mastering HCL Sametime Administration

Trevor Tallackson
Product Manager HCL Sametime- HCL Software

Erik Schwalb
Technical Advisor – HCL Software Germany

ENGAGE

Disclaimer

HCL’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice and at HCL’s sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

Information regarding potential future products is intended to outline HCL's general product direction and should not be relied upon for any other purpose. Information relating to potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality, all products are made available for sale solely at HCL's discretion and shall be subject to relevant terms and conditions. Development, release, and timing of any future features or functionality described for our products remains at HCL's sole discretion.

Performance is based on measurements and projections using standard HCL benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

HCLSoftware

Trevor Tallackson

is the Product Manager for HCL Connections and Sametime. He has supported HCL Digital Solutions software products for over 25 years with a focus on customer solutions and education. During this time, Trevor has developed skills and relationships across the HCL Digital Solutions portfolio.

In his free time, Trevor enjoys coaching hockey, outdoor activities and spending quality time with his wife, three boys and two dogs.

Contact: trevort@pnp-hcl.com

HCLSoftware

Erik Schwalb

Working as a Technical Advisor at HCL Software in Germany, Erik is responsible for consulting and sales of the HCL Digital Solutions portfolio with a focus on Domino, Sametime and Connections.
He has 30+ years of experience in various technical sales roles at Lotus, IBM and HCL.

Erik started working with Sametime in version 1 and later co-authored the Lotus Sametime 2.0 Deployment Guide.
More recently he developed an All-in-one deployment architecture for HCL Sametime on Docker and keeps his hands-on experience current by maintaining several Sametime environments including the HCL Sametime Sandbox.

Contact: erik.schwalb@hcl-software.com

HCLSoftware

HCL Sametime Admin UI

Sametime Admin Web UI

Modern, web-based tool designed to simplify server and policy management

Available starting with 12.0.2 in all deployment variants

- Docker
- OpenShift
- Podman
- Kubernetes
- Windows

What can you do with it?

- Manage Policies
- Monitor Server Status

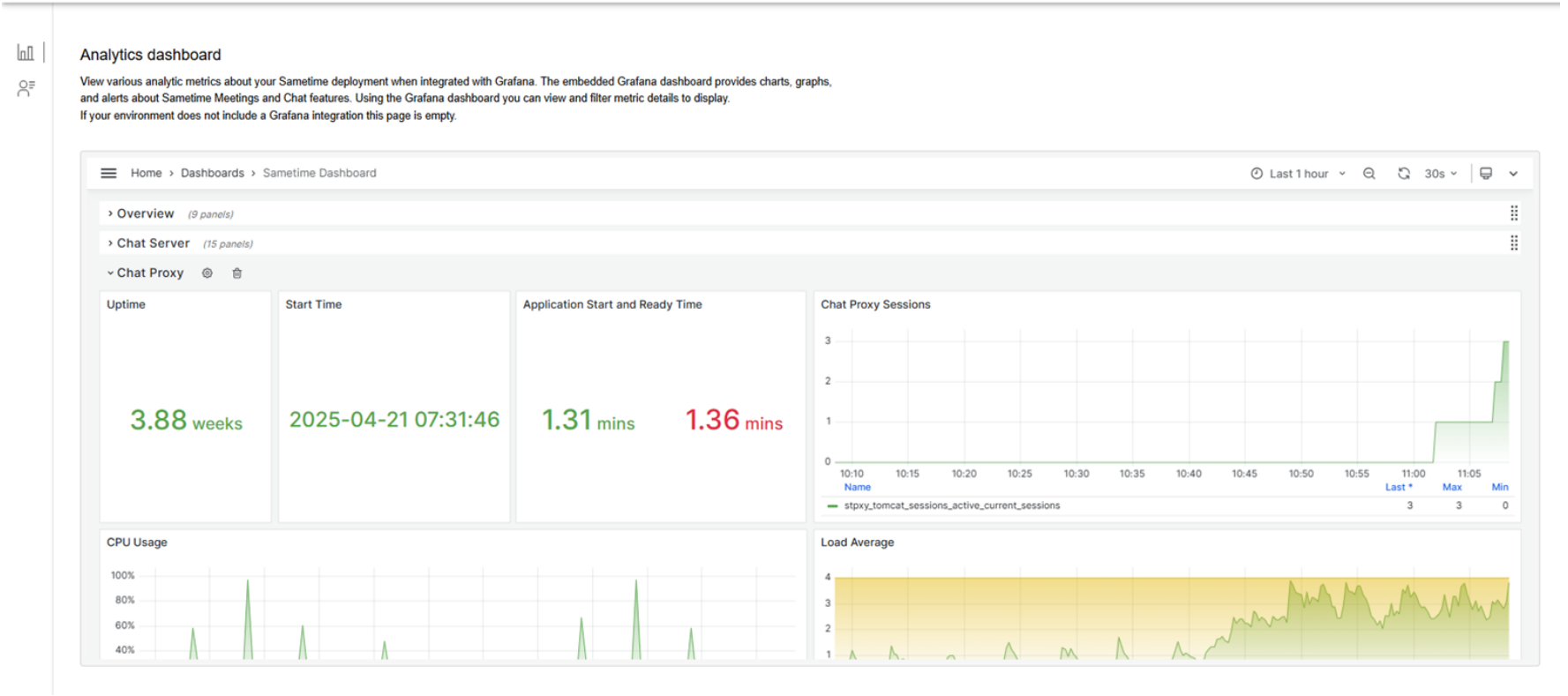
Sametime Admin Web UI

```
# Reset Grafana admin password
cd /opt/hcl/sametime
docker compose exec -it grafana bash
grafana-cli admin reset-admin-password admin
```

The Admin UI embeds the Grafana dashboard for Sametime Monitoring.

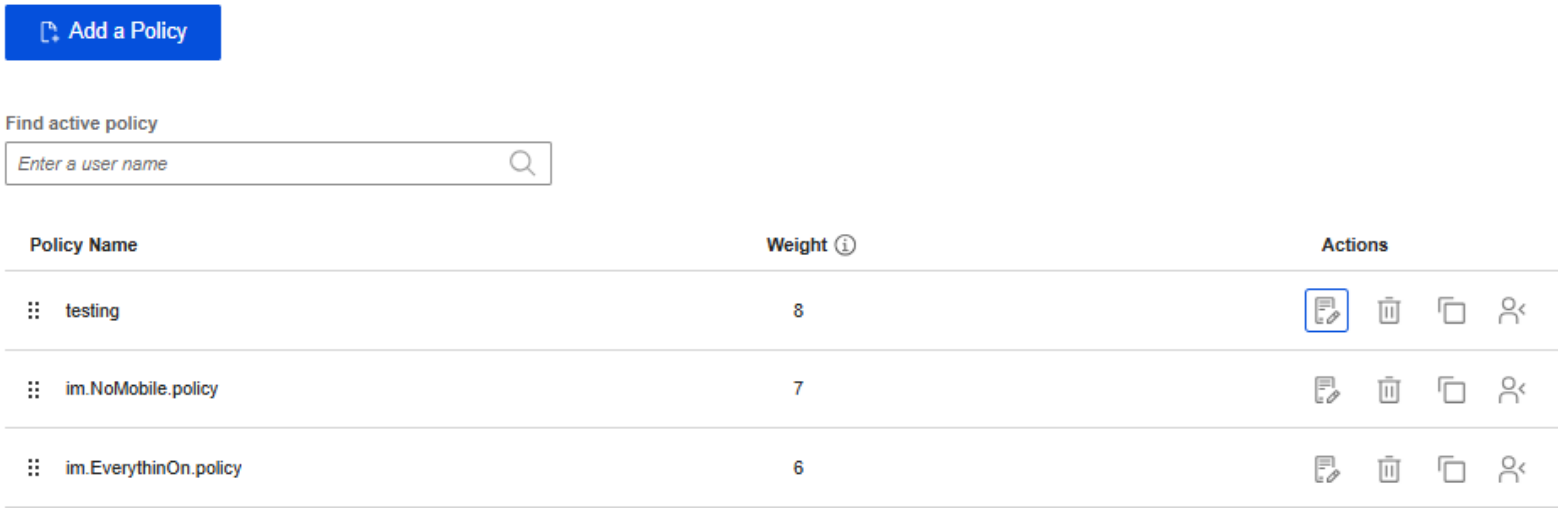
☞ **Grafana admin credentials are separate from your Sametime administrator credentials**, i.e. they don't exist in LDAP. You can also add additional Grafana users or Grafana administrators in the Grafana Web UI.

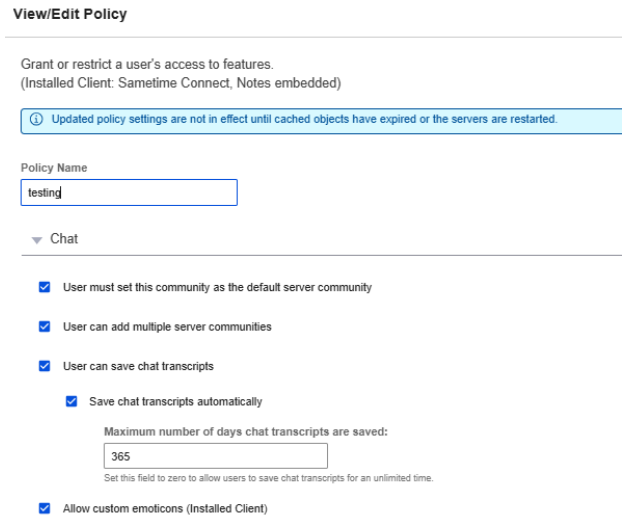
Analytics dashboard

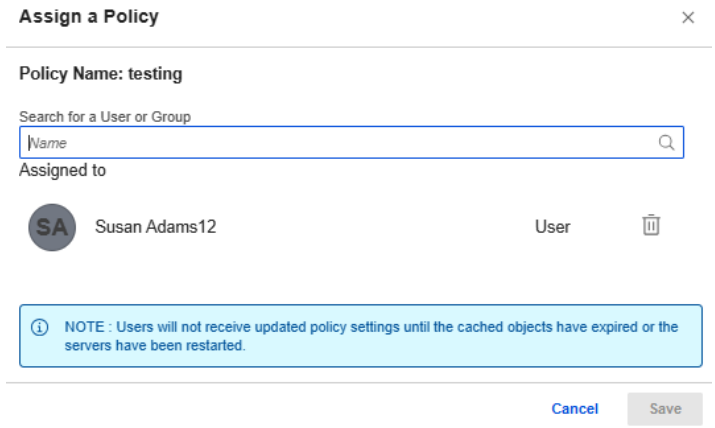


Sametime 12.0.2 Admin UI

The Sametime Admin UI (/admin) provides tools to manage Sametime policies.







How to change or add additional Sametime Administrators

When you deploy Sametime v12x you need to provide the email address of your Sametime administrator.
An account with that email address must already exist in your LDAP directory.

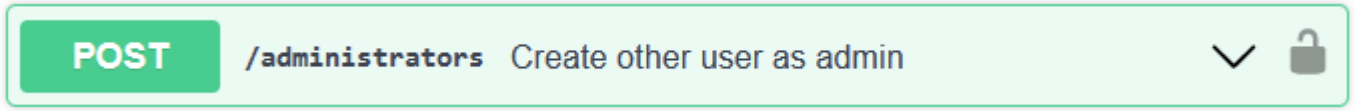
You can **add additional users to the Sametime administrator role** by using one of these methods:

- Use a Mongo shell to add a user as an administrator to the meeting database

```
use meeting
db.administrators.insertOne({"_id": (new ObjectId()).toString(), "email": "adams5@email.com"})
```

- Use the Sametime Admin API

  Retrieve CSRF token from /sametime-auth, then use /sametime-admin to set an additional admin user



Sametime API documentation
https://<your-sametime-server-hostname>/sametime-api/docs/

Sametime Admin Web UI

What would YOU like to see in the admin client?

Your feedback is needed!

- ★ Integrated User Rename
- ★ User Data Administration
- ★ Business Card Administration
- ★ Logging and Tracing
- ★ Track and Manage User/Client activity
- ★ etc.…

https://hclsw.co/sametime-ideas

HCLSoftware

New deployment option: HCL Sametime Chat on Windows

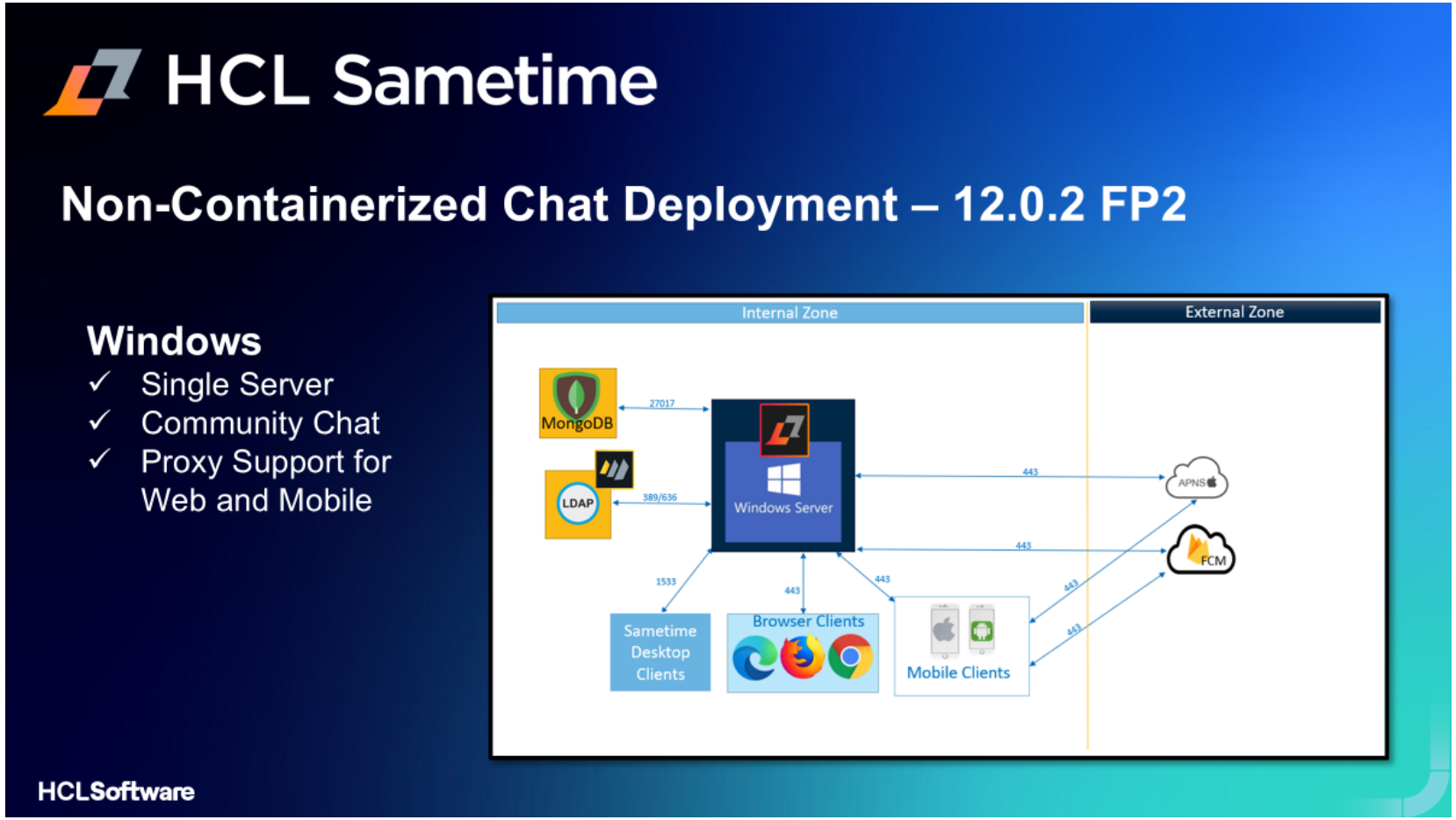
New deployment option: Sametime Chat 12.0.2 FP2 on Windows

Why we created it?

Webinar available to review!

https://youtu.be/9er-OlbvfUw?feature=shared

Unlocking the Future of HCL Sametime New Features, Deployment and Enhanced Meeting Solution



New deployment option: Sametime Chat 12.0.2 FP2 on Windows

Which config files do you need to know about and where are they located? In the install directory:

- Custom.env
- Sametime.ini
- chatlogging.ini
- Policy file(s)
- STCommunityConfig.xml
- UserInfoConfig.xml

Example: How to change LDAP filters or add additional LDAP servers?

- Edit StCommunityConfig.xml and UserInfoConfig.xml files

Same configurations previously stored in stconfig.nsf and vpuserinfo.nsf

New deployment option: Sametime Chat 12.0.2 FP2 on Windows

How to enable Monitoring for Sametime on Windows?

https://developer.ds.hcl-software.com/t/configuring-the-monitoring-dashboard-in-sametime-12-0-2-fp-on-windows/155323

Configuring the Monitoring Dashboard in Sametime 12.0.2 FP on Windows

HCL Sametime ■ Sametime Tips ■ sametime

No ratings yet

★★★★★

This is the first time Erik has posted — let's welcome them to our community!

Erik HCL Software 2 Mar 28

I have configured the Monitoring Dashboard for Sametime 12.0.2 FP2 on Windows and decided, that i wanted to run both Prometheus and Grafana on the same Windows host where Sametime is running.

Here are some tips:

1. Download the latest **Prometheus 2.53.x LTS release** https://prometheus.io/download/

2.53.4 / 2025-03-18 LTS Release notes

File name	OS	Arch	Size
prometheus-2.53.4.darwin-amd64.tar.gz	darwin	amd64	99.84 MiB
prometheus-2.53.4.darwin-arm64.tar.gz	darwin	arm64	96.00 MiB
prometheus-2.53.4.linux-amd64.tar.gz	linux	amd64	99.37 MiB
prometheus-2.53.4.windows-amd64.zip	windows	amd64	101.63 MiB

Mar 28

1 / 1

Mar 28

Mar 28

HCLSoftware

mongoDB®

MongoDB

Considerations for MongoDB

Do NOT run Sametime with an unsupported version of MongoDB

Sametime 12.0.2 FP2 supports MongoDB 7.0

https://support.hcl-software.com/csm?id=kb_article&sysparm_article=KB0117290

Release	Release Date	End of Life Date
MongoDB 8.0	October 2, 2024	TBD
MongoDB 7.0	August 15, 2023	August 31, 2026
MongoDB 6.0	July 19, 2022	July 31, 2025
MongoDB 5.0	July 13, 2021	October 31, 2024
MongoDB 4.4	July 25, 2020	February 29, 2024

https://www.mongodb.com/legal/support-policy/lifecycles

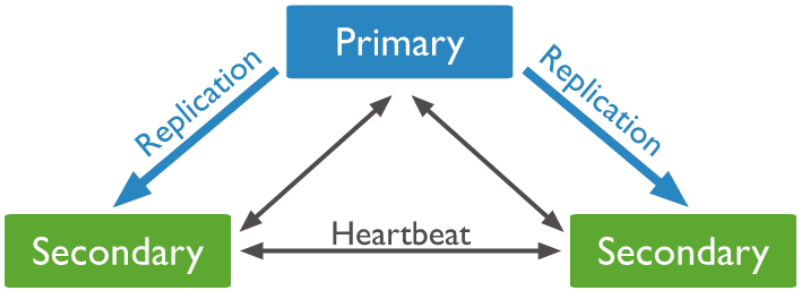
Upgrading MongoDB
In-place approach

Important:

Always follow the official MongoDB documentation for detailled instructions.

In-place upgrading a MongoDB ReplicaSet - in a nutshell:

- Upgrade the MongoDB binaries to next major release on all secondaries.
- Step down the replica set primary to force an election of a new primary.
- Upgrade the MongoDB binaries on the stepped-down primary.
- Increment the feature compatibility version.



https://www.mongodb.com/docs/manual/release-notes/7.0-upgrade-replica-set

Upgrading MongoDB
Side-by side approach

Implement a new MongoDB ReplicaSet and migrate your data with **mongodump** and **mongorestore**.

Remark: It is recommended to restore to a matching major version.

Back up your databases with mongodump

```
mongodump --host=<oldmongohost>:<port> --username=<mongouser> --password=<mongopassword> --forceTableScan
--gzip --db=meeting --out /dump

mongodump --host=<oldmongohost>:<port> --username=<mongouser> --password=<mongopassword> --forceTableScan
--gzip --db=mobileOffline --out /dump

mongodump --host=<oldmongohost>:<port> --username=<mongouser> --password=<mongopassword> --forceTableScan
--gzip --db=chatlogging --out /dump

mongodump --host=<oldmongohost>:<port> --username=<mongouser> --password=<mongopassword> --forceTableScan
--gzip --db=privacy --out /dump

mongodump --host=<oldmongohost>:<port> --username=<mongouser> --password=<mongopassword> --forceTableScan
--gzip --db=userinfo --out /dump
```

https://www.mongodb.com/docs/database-tools/mongodump/
https://www.mongodb.com/docs/database-tools/mongorestore/

Upgrading MongoDB
Side-by side approach (cont.)

Restore your databases with mongorestore

```
mongorestore --host=<newmongohost>:<port> --username=<mongouser> --password=<mongopassword> --drop
--noIndexRestore --gzip --verbose --nsInclude=meeting.* /restore

mongorestore --host=<newmongohost>:<port> --username=<mongouser> --password=<mongopassword> --drop
--noIndexRestore --gzip --verbose --nsInclude=mobileOffline.* /restore

mongorestore --host=<newmongohost>:<port> --username=<mongouser> --password=<mongopassword> --drop
--noIndexRestore --gzip --verbose --nsInclude=chatlogging.* /restore

mongorestore --host=<newmongohost>:<port> --username=<mongouser> --password=<mongopassword> --drop
--noIndexRestore --gzip --verbose --nsInclude=privacy.* /restore

mongorestore --host=<newmongohost>:<port> --username=<mongouser> --password=<mongopassword> --drop
--noIndexRestore --gzip --verbose --nsInclude=userinfo.* /restore
```

https://www.mongodb.com/docs/database-tools/mongodump/
https://www.mongodb.com/docs/database-tools/mongorestore/

Considerations for MongoDB

Preparing MongoDB for Sametime includes:

- Configure MongoDB with a ReplicaSet on either a single MongoDB server or on a full MongoDB cluster
- Create an account in MongoDB to be used by Sametime and assign that account the required roles
- Create the chatlogging database and initialize the EVENTS and SESSIONS collections with dummy values

Important:

Always follow the official MongoDB documentation for detailled instructions.

Convenience script for MongoDB

`install_mongodb7_for_sametime.sh` can help you to get started.
Send me an email if you want to get the script.

```
#############################################################################
#                                                                           #
# Automated MongoDB Installation and Configuration for HCL Sametime Premium #
#                                                                           #
# Version 1.4.2 (c) 2025 Erik Schwalb (HCL)                                 #
#                                                                           #
#############################################################################
#############################################################################
#                                                                           #
# This script will install and configure MongoDB 7 on CentOS 9 or Ubuntu 22.04. #
#                                                                           #
# If you are installing MongoDB on a different version of CentOS / Red Hat or Ubuntu #
# or on another Linux distribution, you need to modify the script before running it. #
#                                                                           #
# Examine the content of the script before running it and make sure it corresponds #
# to the instructions from the HCL Sametime product documentation and to the #
# instructions from the official MongoDB documentation.                     #
#                                                                           #
# Support Statement:                                                        #
# HCL publishes the script as a sample without any warranties.              #
#                                                                           #
#############################################################################


Your current platform: Ubuntu 24.04.2 LTS

Press Enter to continue or Ctrl-c to exit.
```

You are still responsible for properly securing your MongoDB instance!

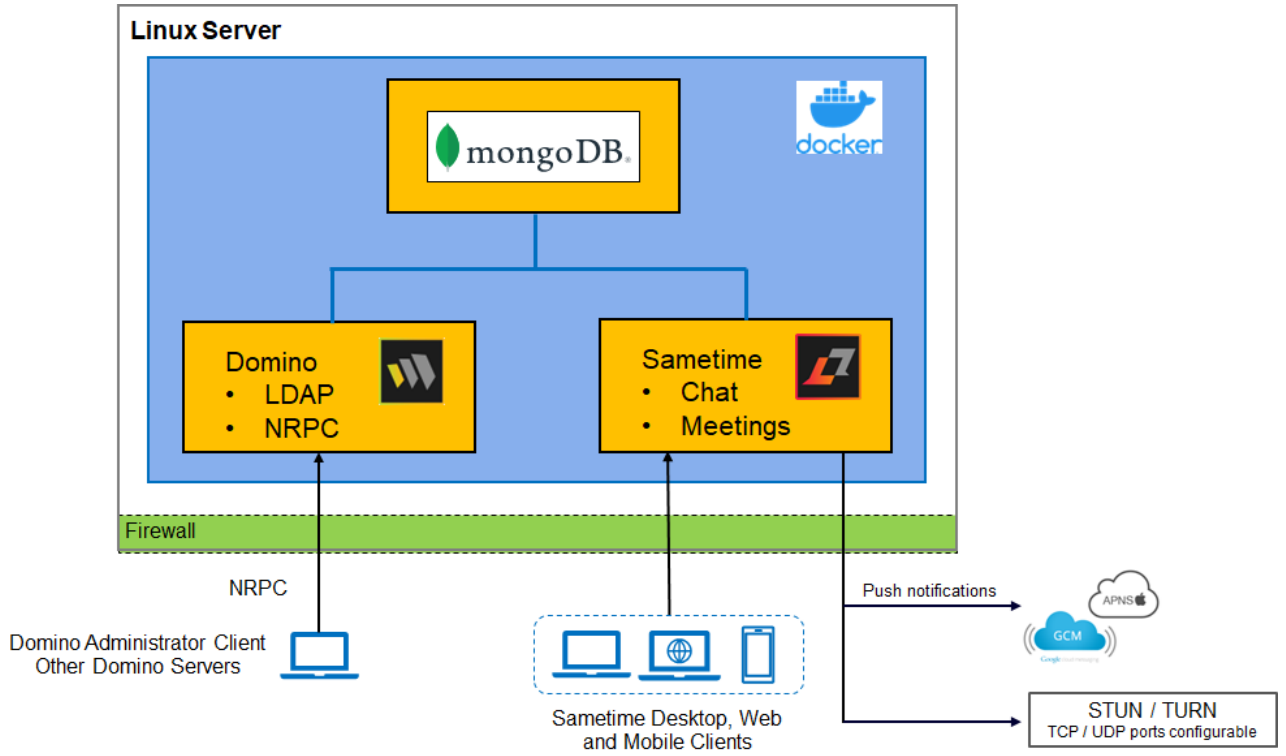
Deploying MongoDB on Docker

If you want to run MongoDB on the same Docker host as Sametime

- Deploy either a single MongoDB server or a MongoDB cluster on Docker
- Connect MongoDB container(s) to the same Docker bridge network as Sametime
- No need to expose MongoDB port(s) on localhost, no need to open Firewalls for MongoDB

Self contained Sametime software appliance:

Don’t try to run Mongo on podman unless officially supported by MongoDB!



Deploying MongoDB on Kubernetes

The "built-in" Sametime 12.0.2 FP2 Helm Charts for MongoDB can be used for a quick test.

For production use you also need to provide persistent storage (RWX PVs) in your Kubernetes cluster and implement administrative procedures for backup and restore.

For production environments you can also use…

- The official MongoDB Community Kubernetes Operator
  https://github.com/mongodb/mongodb-kubernetes-operator
- or the MongoDB Enterprise Kubernetes Operator
  https://github.com/mongodb/mongodb-enterprise-kubernetes
- or the Bitnami Helm Charts
  https://github.com/bitnami/charts/tree/main/bitnami/mongodb

HCLSoftware

Let's Encrypt

Integration with Let's Encrypt

Integrate HCL Sametime with Let’s Encrypt

Sametime can request, retrieve and apply a TLS certificate from Let’s Encrypt.

It will also automatically renew the TLS certificate before it expires.

The built-in integration is based on the ACME protocol (**A**utomatic **C**ertificate **M**anagement **E**nvironment) using an HTTP-01 challenge to verify, that you are the owner of the requesting website.

1. ACME server (= Let's Encrypt) sends a challenge to ACME client *
2. ACME server will ask via **in-bound HTTP request on port 80** for the “secret” at a well-known URL

- **The FQDN of your Sametime server must be registered in public DNS**
- **Your Sametime server must be accessible on the public Internet via http port 80**

* NGINX in Sametime on Docker / Traefik in Sametime on Windows

How to configure Sametime with Let's Encrypt

1. Backup current Sametime configuration “config 0”
2. Configure Sametime with Let’s Encrypt staging and make sure everything is fine
3. Restore previous Sametime configuration “config 0”
4. Configure Sametime with Let’s Encrypt production

The Let's Encrypt API has rate limiting (lasting up to one week, cannot be overridden).

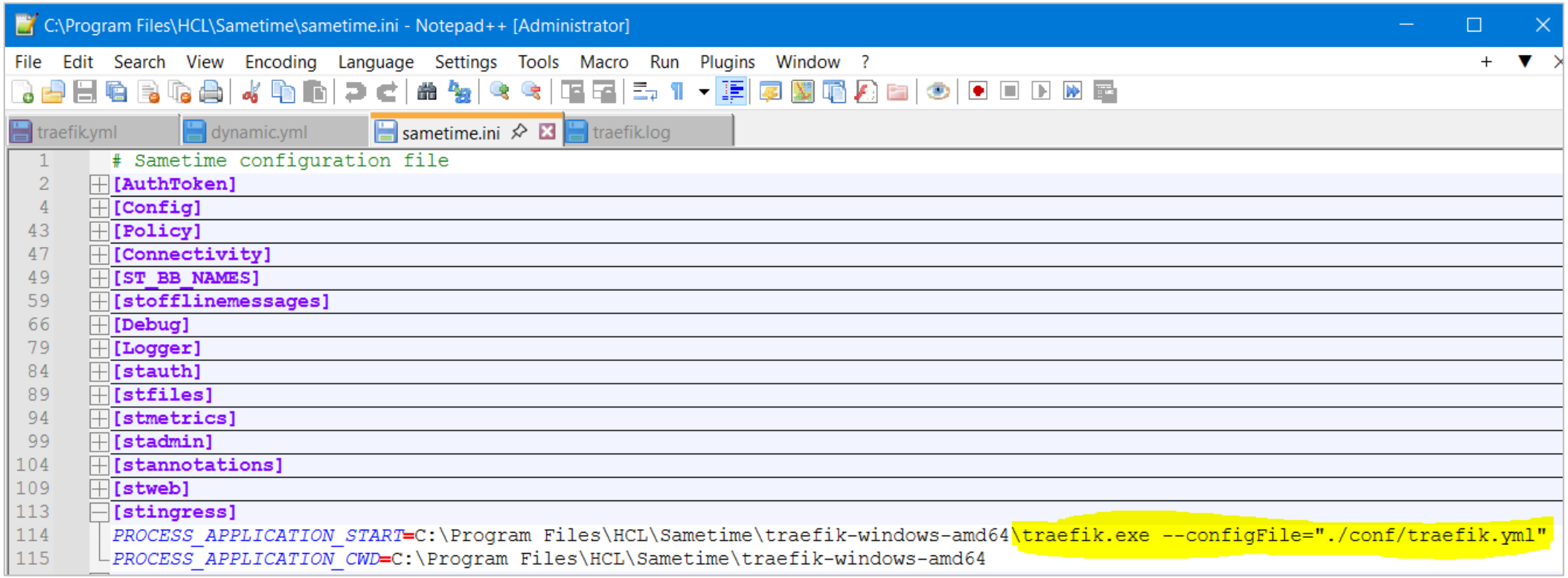
Always use the Let's Encrypt staging server when experimenting to avoid hitting this limit which would lock your Let’s Encrypt account.

Integrate Sametime on Windows with Let's Encrypt

Shutdown Sametime

Modify `traefik.yml` and `dynamic.yml` as described in the Sametime documentation

Adjust `traefik.yml` as described on the following slides, then restart Sametime



Integrate Sametime on Windows with Let's Encrypt

https://doc.traefik.io/traefik/https/acme/#caserver

Settings in **`traefik.yml`**

caServer ¶

Required, Default="https://acme-v02.api.letsencrypt.org/directory"

The CA server to use:

- Let's Encrypt production server: https://acme-v02.api.letsencrypt.org/directory
- Let's Encrypt staging server: https://acme-staging-v02.api.letsencrypt.org/directory

Using the Let's Encrypt staging server

File (YAML) | File (TOML) | CLI

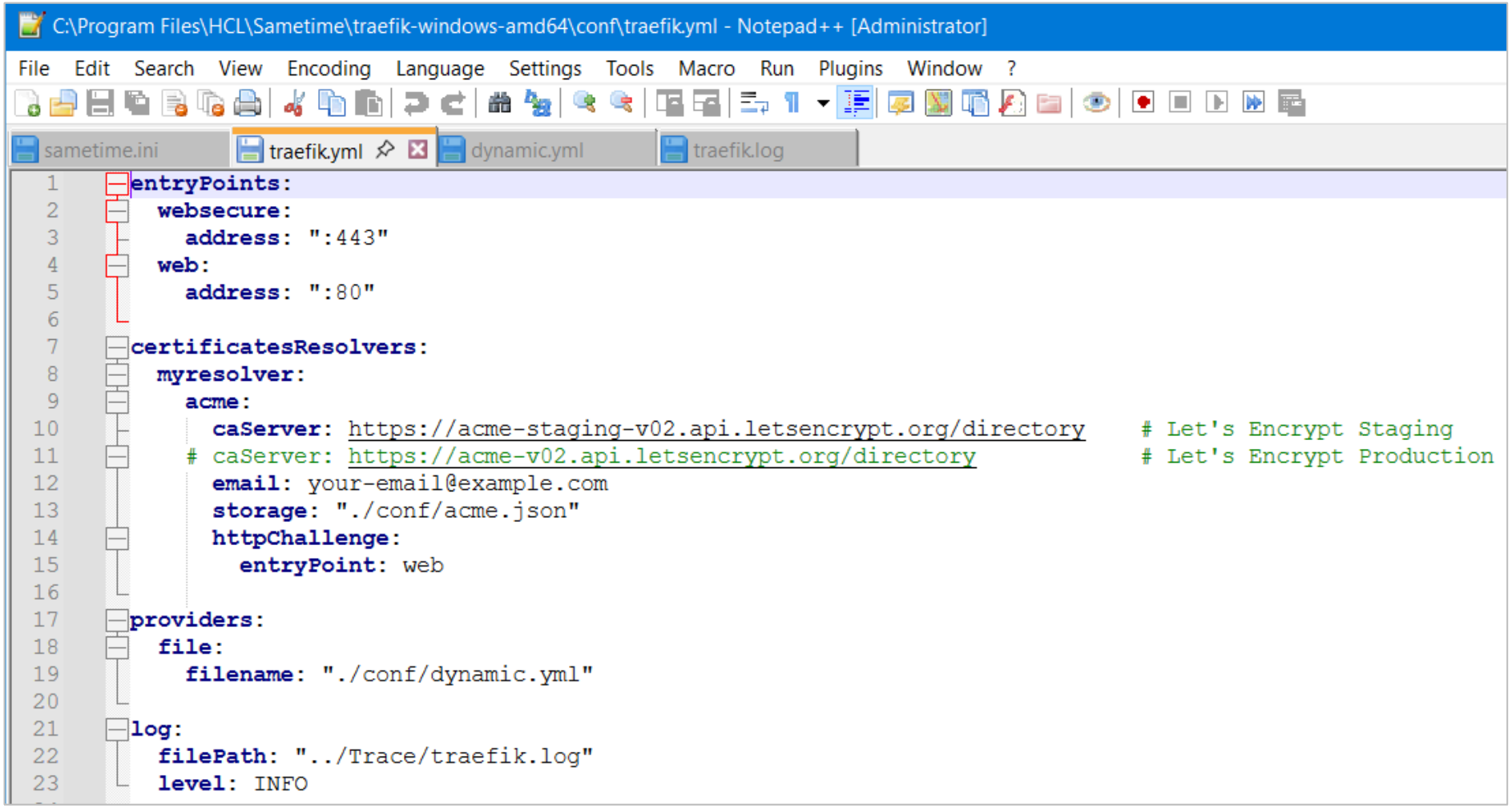
```yaml
certificatesResolvers:
  myresolver:
    acme:
      # ...
      caServer: https://acme-staging-v02.api.letsencrypt.org/directory
      # ...
```

Default = **Let's Encrypt production**

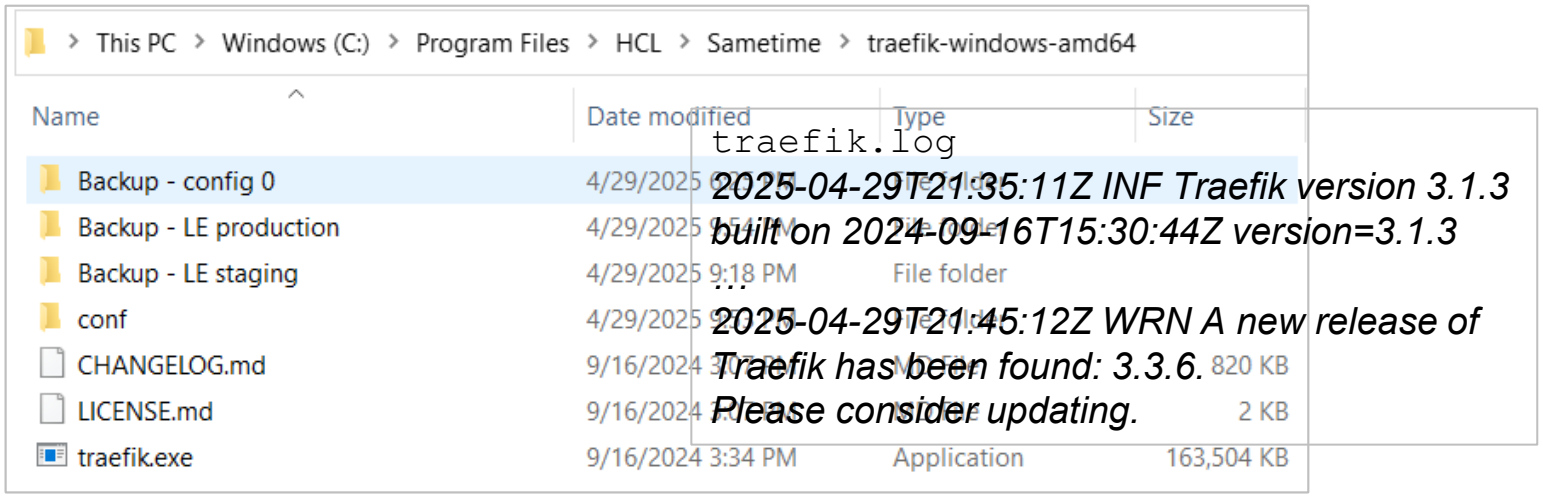
If you set up the integration for the first time, make sure to test everything first with the **Let's Encrypt staging** service enabled!

Integrate Sametime on Windows with Let's Encrypt

Settings in **`traefik.yml`**



Optional: Update Traefik to a newer 3.x version



traefik.log

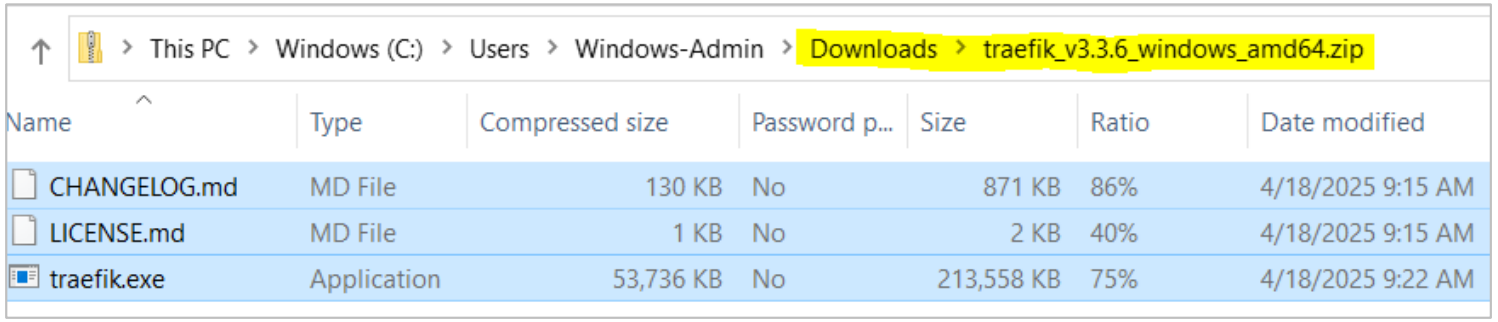
2025-04-29T21:35:11Z INF Traefik version 3.1.3 built on 2024-09-16T15:30:44Z version=3.1.3

...

2025-04-29T21:45:12Z WRN A new release of Traefik has been found: 3.3.6. Please consider updating.

Unzip and copy to Sametime

https://github.com/traefik/traefik/releases/tag/v3.3.6



Integrate Sametime on Docker with Let's Encrypt

Settings in `.env`

```
# Exposed HTTP port
HTTP_PORT=80
# Exposed HTTPS port
HTTPS_PORT=443

# Redirect HTTP traffic to HTTPS
# Necessary for Let's Encrypt, relies on standard HTTPS port (443)
ENABLE_HTTP_REDIRECT=0

# ENABLE_HTTP_REDIRECT=1

# Enable Let's Encrypt certificate generation
ENABLE_LETSENCRYPT=1
# Domain for which to generate the certificate
LETSENCRYPT_DOMAIN=<your.sametimeserver.com>
# E-Mail for receiving important account notifications (mandatory)
LETSENCRYPT_EMAIL=<YourAdminEmailAddress>
# Use the staging server (for avoiding rate limits while testing)
# LETSENCRYPT_USE_STAGING=1
```

Set HTTP_PORT to 80

For Sametime 12.0.2 incl. FP1 and FP2
ENABLE_HTTP_REDIRECT=0

For Sametime 12.0.1
ENABLE_HTTP_REDIRECT=1

If you set up the integration for the first time, make sure to test it first with the **Let's Encrypt staging** service enabled!

Integrate Sametime 12 on Docker with Let's Encrypt

Settings in **`docker-compose.yml`**

```
nginx:
    image: hclcr.io/st/meetings-web:${BUILD_LEVEL}
    restart: ${RESTART_POLICY}
    ports:
      - '${HTTP_PORT}:80'
      - '${HTTPS_PORT}:443'
    ...
    environment:
    ...
      - LETSENCRYPT_DOMAIN
      - LETSENCRYPT_EMAIL
      - LETSENCRYPT_USE_STAGING
    ...
```

Settings in **`custom.env`**

```
# Example: PUBLIC_URL=https://sametime.company.com
PUBLIC_URL=https://<your.sametimeserver.com>
```

A setting for PUBLIC_URL can be found both in `.env` and in `custom.env`.

Make sure you define a value for this setting only in `custom.env`.

If you want to use the Let's Encrypt staging service make sure `LETSENCRYPT_USE_STAGING` is included in the list of environment variables of the nginx service.

Integrate Sametime on Docker with Let's Encrypt

The nginx container will use the ACME protocol to register an account with Let's Encrypt.
If successfull it will then request and retrieve a TLS certificate.

```
| [Fri Sep  2 18:06:09 CEST 2022] Using CA: https://acme-v02.api.letsencrypt.org/directory
| [Fri Sep  2 18:06:09 CEST 2022] Run pre hook:'if [[ -d /var/run/s6/services/nginx ]]; then s6-svc -d /var/run/s6/services
| [Fri Sep  2 18:06:09 CEST 2022] Standalone mode.
| [Fri Sep  2 18:06:09 CEST 2022] Create account key ok.
| [Fri Sep  2 18:06:09 CEST 2022] Registering account: https://acme-v02.api.letsencrypt.org/directory
| [Fri Sep  2 18:06:11 CEST 2022] Registered
| [Fri Sep  2 18:06:11 CEST 2022] ACCOUNT_THUMBPRINT='IGj4UFtbU5Z4FQ2HEgo_jRTMO2RzdLET36DrDURCu_U'
| [Fri Sep  2 18:06:11 CEST 2022] Creating domain key
| [Fri Sep  2 18:06:11 CEST 2022] The domain key is here: /config/acme.sh/sametime.dnug.eu/sametime.dnug.eu.key
| [Fri Sep  2 18:06:11 CEST 2022] Single domain='sametime.dnug.eu'
| [Fri Sep  2 18:06:12 CEST 2022] Getting domain auth token for each domain
| [Fri Sep  2 18:06:14 CEST 2022] Getting webroot for domain='sametime.dnug.eu'
| [Fri Sep  2 18:06:14 CEST 2022] Verifying: sametime.dnug.eu
| [Fri Sep  2 18:06:14 CEST 2022] Standalone mode server
| [Fri Sep  2 18:06:19 CEST 2022] Success
| [Fri Sep  2 18:06:19 CEST 2022] Verify finished, start to sign.
| [Fri Sep  2 18:06:19 CEST 2022] Lets finalize the order.
| [Fri Sep  2 18:06:19 CEST 2022] Le_OrderFinalize='https://acme-v02.api.letsencrypt.org/acme/finalize/714169367/1218760562
| [Fri Sep  2 18:06:21 CEST 2022] Downloading cert.
| [Fri Sep  2 18:06:21 CEST 2022] Le_LinkCert='https://acme-v02.api.letsencrypt.org/acme/cert/049e2ec117270078520a43948fb99
| [Fri Sep  2 18:06:21 CEST 2022] Cert success.
| -----BEGIN CERTIFICATE-----
| MIIFJjCCBA6gAwIBAgISBJ4uwRcnAHhSCkOUj7mV4CzYMA0GCSqGSIb3DQEBCwUA
```

Integrate Sametime on Docker with Let's Encrypt

```
| Bk2faD7ys1QxKQfS6R4h5v93pZXYizWfroI=
| -----END CERTIFICATE-----
| [Fri Sep  2 18:06:22 CEST 2022] Your cert is in  /config/acme.sh/sametime.dnug.eu/sametime.dnug.eu.cer
| [Fri Sep  2 18:06:22 CEST 2022] Your cert key is in  /config/acme.sh/sametime.dnug.eu/sametime.dnug.eu.key
| [Fri Sep  2 18:06:22 CEST 2022] The intermediate CA cert is in  /config/acme.sh/sametime.dnug.eu/ca.cer
| [Fri Sep  2 18:06:22 CEST 2022] And the full chain certs is there:  /config/acme.sh/sametime.dnug.eu/fullchain.cer
| [Fri Sep  2 18:06:22 CEST 2022] Run post hook:'if [[ -d /var/run/s6/services/nginx ]]; then s6-svc -u /var/run/s6/services
| [Fri Sep  2 18:06:22 CEST 2022] Installing key to:/config/acme-certs/sametime.dnug.eu/key.pem
| [Fri Sep  2 18:06:22 CEST 2022] Installing full chain to:/config/acme-certs/sametime.dnug.eu/fullchain.pem
```

The TLS certificate will be placed in a subdirectory below the ./sametime-config directory, that is named after the FQDN of your server.

```
# List the TLS certificate

ls -l sametime-config/web/acme-certs/SametimeServerFQDN
```

```
-rw-r--r--. 1 root root 5597 Sep  2 18:06 fullchain.pem
-rw-------. 1 root root 1679 Sep  2 18:06 key.pem
```

Integrate Sametime on Kubernetes with Let’s Encrypt

Sametime on Kubernetes does not include built-in integration with Let's Encrypt. However, you can use cert-manager to get a TLS certificate from Let’s Encrypt and use it with your Sametime deployment.

cert-manager adds custom objects such as Certificates, CertificateRequests and Issuers as resource types in Kubernetes clusters, and simplifies the process of obtaining, renewing and using those certificates.

- Install and configure cert-manager including CRDs
- Create Let's Encrypt `ClusterIssuer` for staging and production
- Edit the file `ingress.yaml` that is included with the Sametime helm charts and insert a new line with the string `cert-manager.io/cluster-issuer: "letsencrypt-staging"` in the `annotations` section

When you then deploy Sametime, an ingress will be created as part of the deployment and cert-manager will automatically provision a TLS certificate from Let’s Encrypt.

➢ https://cert-manager.io

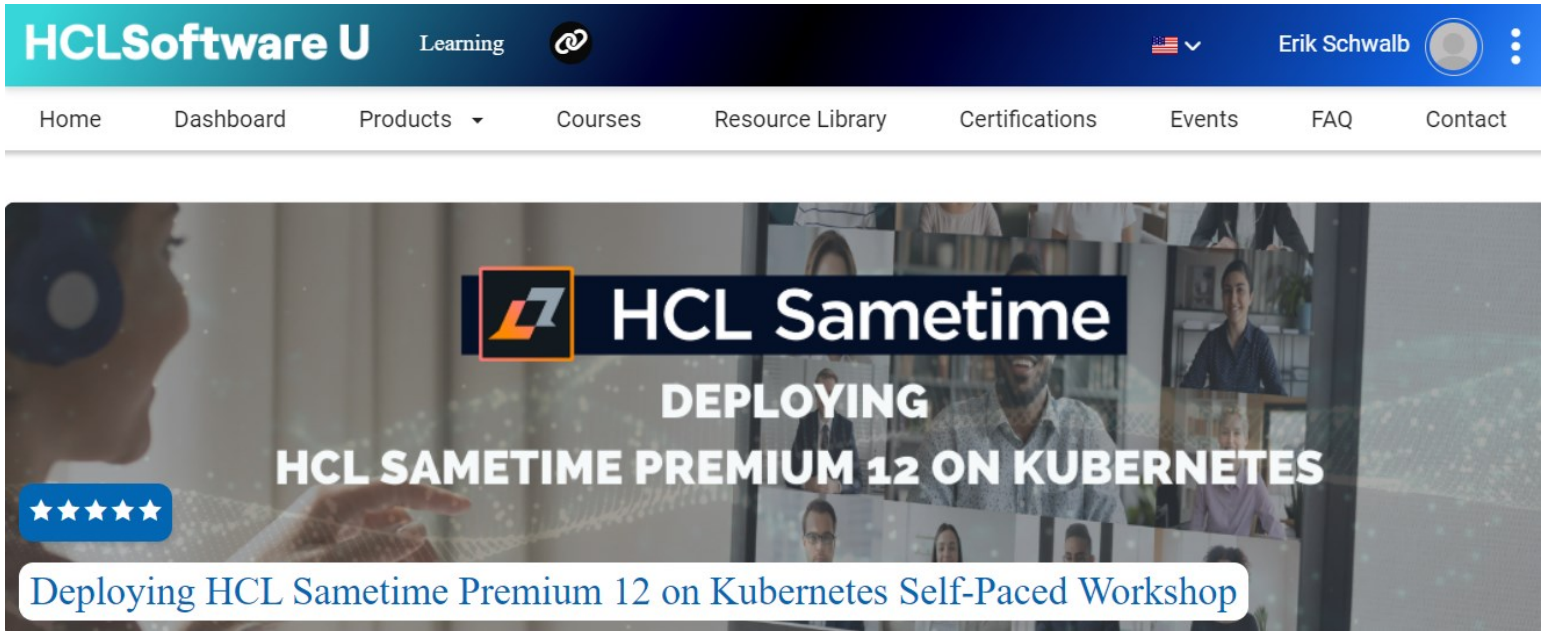
Integrate Sametime on Kubernetes with Let's Encrypt

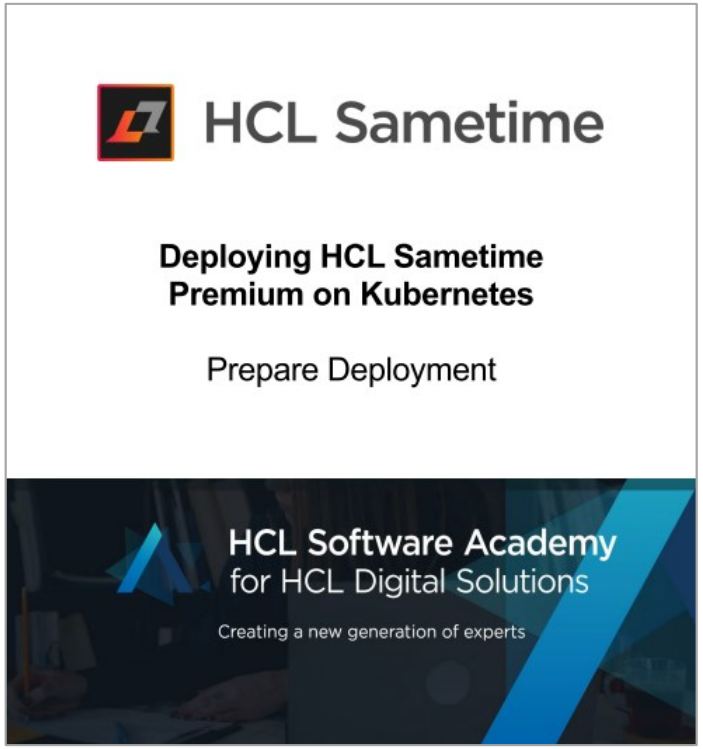

```
  GNU nano 2.3.1                File: /opt/hcl/sametime/helm/charts/web/templates/ingress.yaml

{{ if .Values.global.sofySolutionContext }}
{{ else }}
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: {{ include "web.fullname" . }}
  labels:
    {{- include "web.labels" . | nindent 4 }}
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
    networking.gke.io/v1beta1.FrontendConfig: {{ include "web.fullname" . }}-frontend-config
    nginx.ingress.kubernetes.io/proxy-body-size: "0"
    nginx.ingress.kubernetes.io/ssl-redirect: {{ not (default false .Values.global.tlsTermination
    nginx.ingress.kubernetes.io/force-ssl-redirect: {{ not (default false .Values.global.tlsTermi
    cert-manager.io/cluster-issuer: "letsencrypt-prod"
spec:
{{ if .Values.global.ingressClassName }}
  ingressClassName: {{ .Values.global.ingressClassName }}
{{ else }}
  ingressClassName: nginx
{{ end }}
```

Integrate Sametime on Kubernetes with Let's Encrypt

Register for the free of charge Deploying HCL Sametime Premium 12 on Kubernetes Self-Paced Workshop and get **step-by-step instructions** in the Prepare Deployment chapter.

- https://hclsoftwareu.hcltechsw.com/hclsoftwareu-courses/course/sametime-on-kubernetes-self-paced





HCLSoftware

Coming Soon

OIDC Authentication

Sametime support for OIDC authentication

Why did we do this?

- Modern, Standards-Based Authentication
- Seamless Single Sign-On (SSO)
- Improved Security
- Critical for Mobile and Web UX
- Essential for Digital Sovereignty and Federated Environments
- Foundation for Future Integrations

*Targeting v12.0.3 FP1 - OND 2025

HCLSoftware

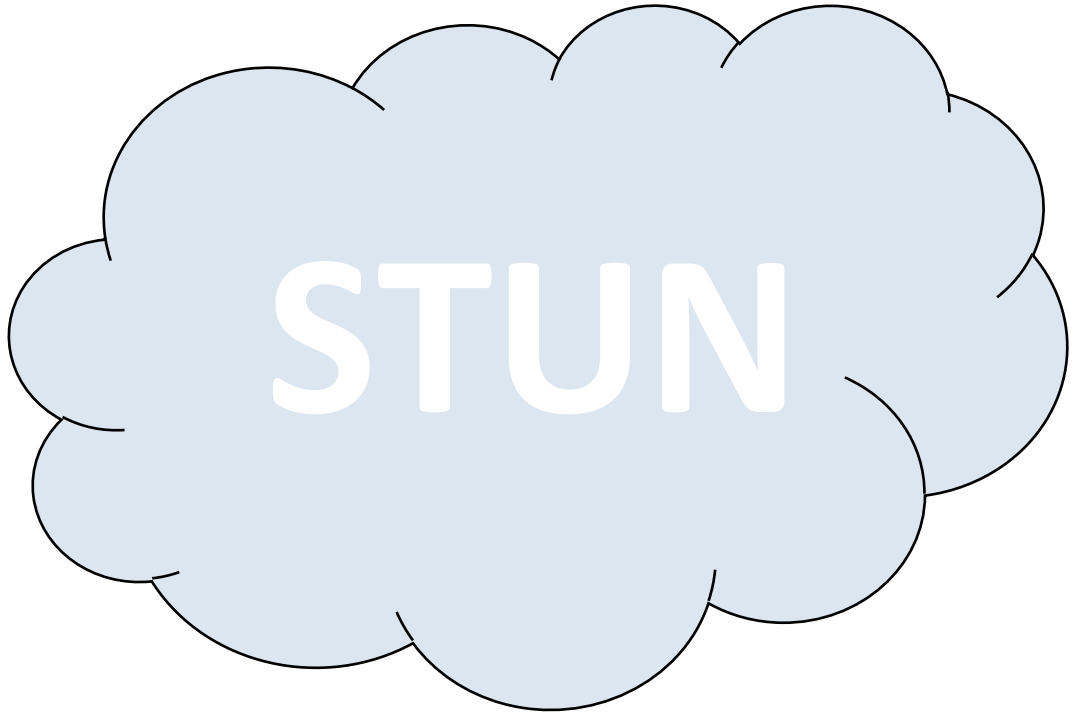
STUN and TURN

Sametime Dependency

- Helps clients determine public IP addresses to connect to each other and the Meetings Server to send and receive A/V data.
- Media A/V streams over UDP port:

  10,000 for Docker

  30,000 for Kubernetes
- Required if any user is attending from behind a firewall.
- Default configuration comes with the public Google STUN server using UDP port 19302.
- Any STUN server can be used located on the opposing public side of the NAT.

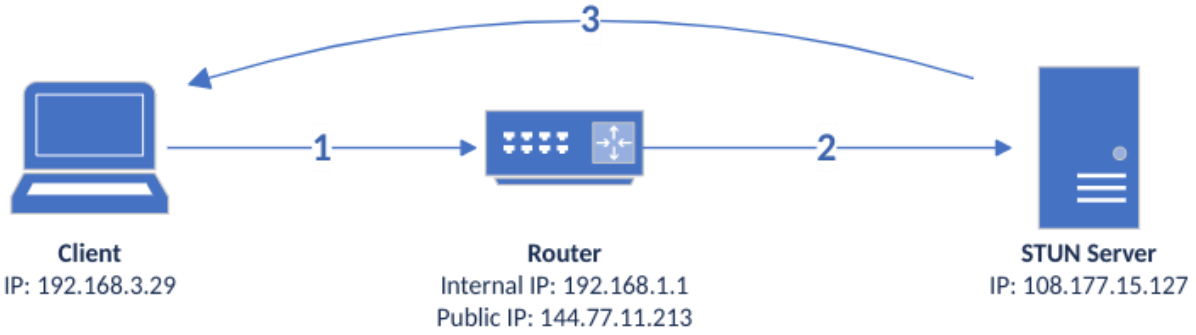


STUN (Session Traversal Utilities for NAT) is a standardized set of methods, including a network protocol, for traversal of network address translator (NAT) gateways in applications of real-time voice, video, messaging, and other interactive communications.

Implementing and configuring an internal STUN server

How does it work ?

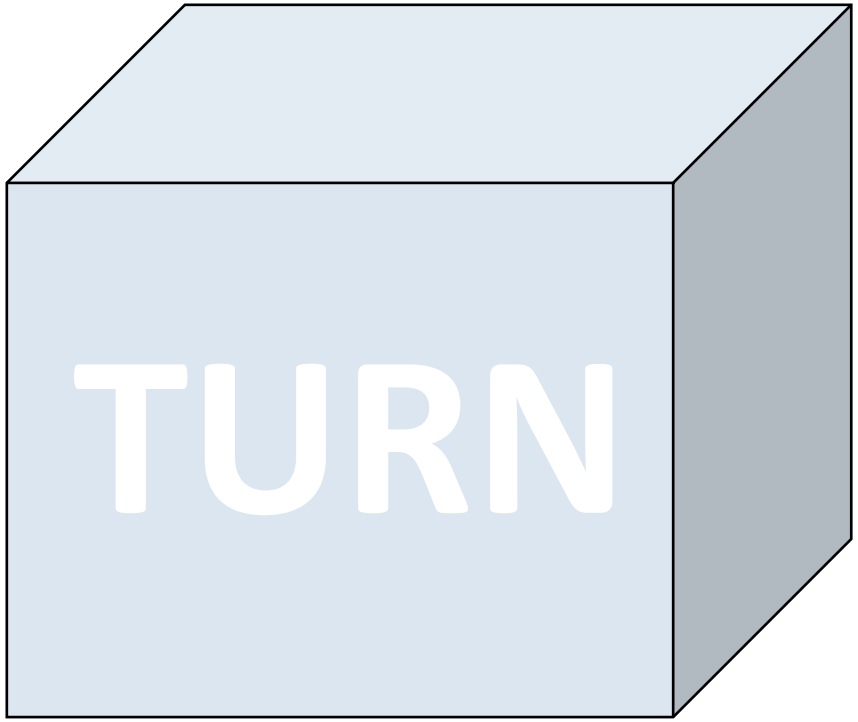
1. Client 192.168.3.29 sends a STUN request through Router 192.168.1.1 to a STUN server outside the network, listening on 108.177.15.127, using source port 5090.
2. Router 192.168.1.1 forwards the request to STUN server 108.177.15.127 and changes port 5090 to port 15090.
3. STUN Server 108.177.15.127 sends a response back to Client 192.168.3.29 through the Router with public IP 144.77.11.213 specifying that the request was received from IP 144.77.11.213 and port 15090.



https://en.wikipedia.org/wiki/STUN

Sametime Dependency

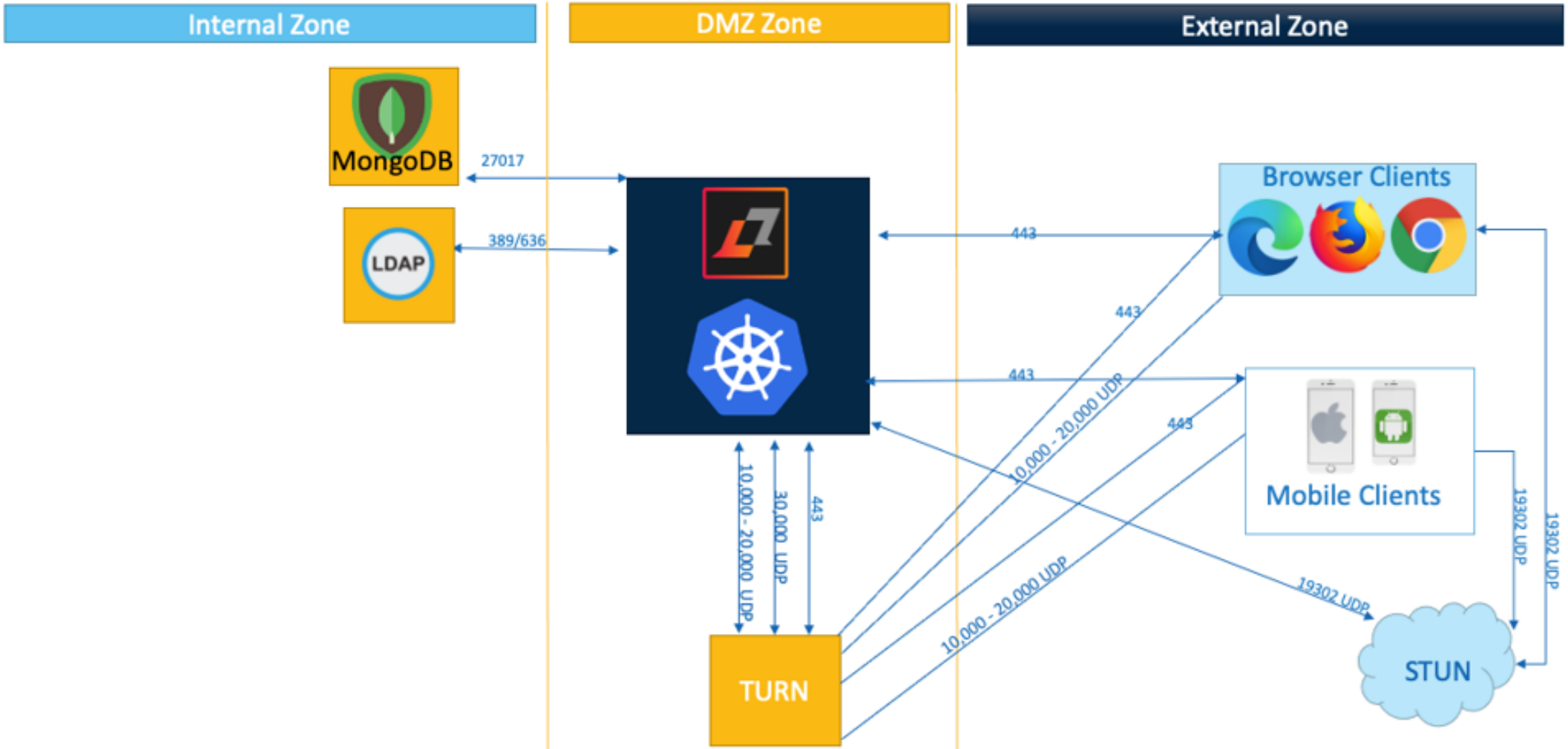
- Needed for users who may have a restrictive network environment.
- Optional service to improve user experience when there is no connectivity to the UDP port.
- Users connect to the TURN server instead of Sametime directly.
- TURN server can be used to relay A/V traffic over TCP port 443.



Traversal Using Relays around NAT (TURN) is a protocol that assists in traversal of network address translators (NAT) or firewalls for multimedia applications. It may be used with the TCP and UDP. TURN supports the connection of a user behind a NAT to only a single peer, as in telephony, for example.

Topology

Sametime with TURN



HCLSoftware

Project ZeroFail

Project ZeroFail

Mission Statement:

To eliminate all barriers to joining HCL Sametime Meetings by ensuring every user — guest or authenticated — can connect with working audio, video, and screen sharing on the first try, every time. Our goal is to restore confidence in Sametime by delivering a seamless, reliable, and frustration-free meeting experience.

Key Objectives:

- ❑ Ensure Every Meeting Join Works Right, Every Time
- ❑ Eliminate Mobile Barriers
- ❑ Optimize On-Premise Functionality
- ❑ Resolve Common Audio/Video Issues
- ❑ Improve Cross-Platform Meeting Join Experience

Project ZeroFail

Our Ask

- Share real user experience
- Be honest and engaging
- Identify, test and “ZeroFail” environments our customers (and guests) are using
- Details matter
- Use the community and support

Contact me!

trevort@pnp-hcl.com

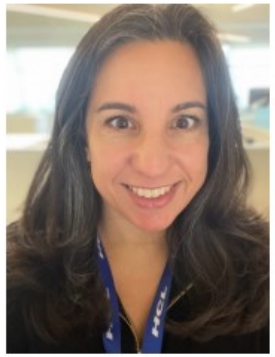
Make a Difference!

Join the new Digital Solutions Community

HCLSoftware

Deploying Sametime 12.0.2 on Kubernetes Single Server

HCLSoftware U - Whitepapers



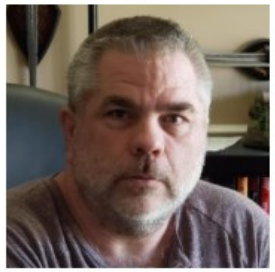
Casey Toole

HCL

Casey Toole

Casey is a Senior Software Engineer supporting the Sametime product since 1999. She handles customer escalations and has authored many whitepapers. She installed the HCL production Sametime environment and has administered it for 3 years.

Contact: casey.toole@pnp-hcl.com
linkedin.com/caseytoole



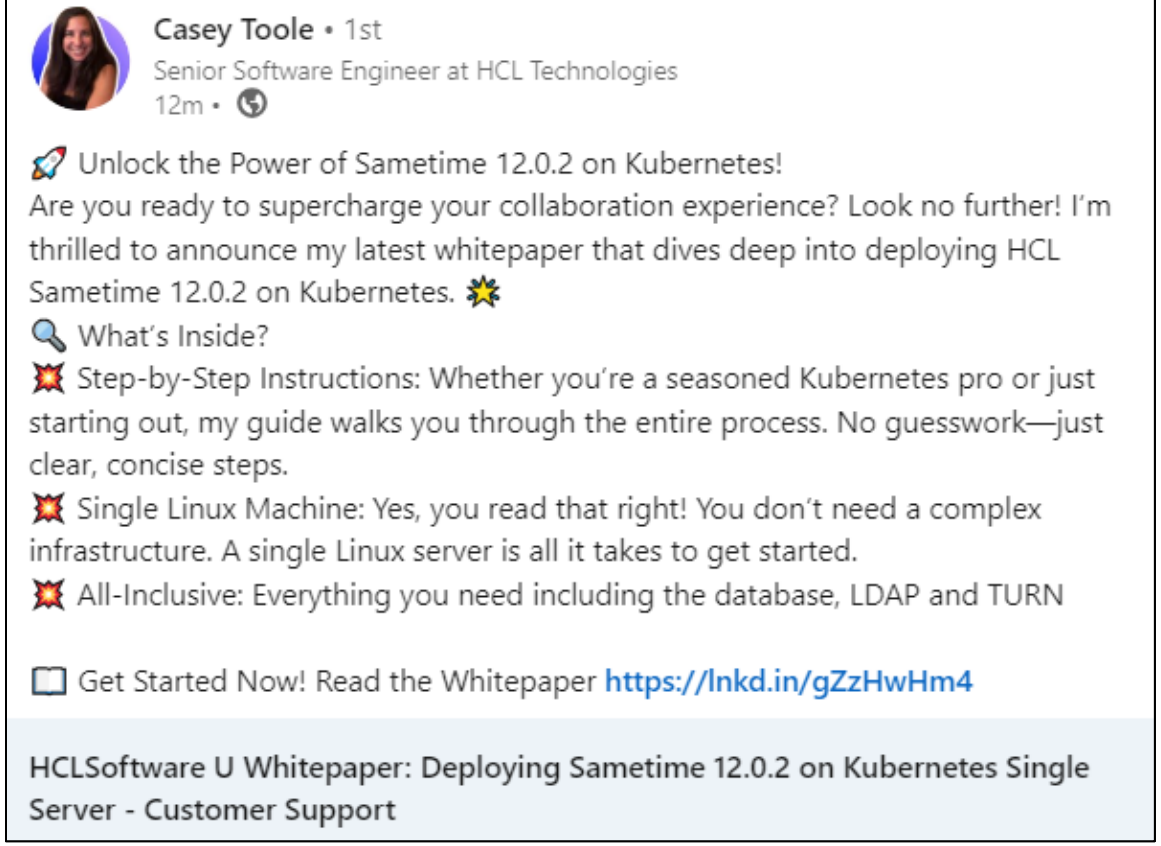
Tony Payne

HCL

Tony Payne is a senior member of our Sametime Development team with over 25 years of Sametime experience and knowledge. His current focus is on customer support as lead on our L3 support team.

Contact: tony.payne@pnp-hcl.com

https://hclsoftwareu.hcltechsw.com
https://support.hcltechsw.com/csm?id=kb_article&sysparm_article=KB0112305



Casey Toole • 1st
Senior Software Engineer at HCL Technologies
12m •

🚀 Unlock the Power of Sametime 12.0.2 on Kubernetes!
Are you ready to supercharge your collaboration experience? Look no further! I'm thrilled to announce my latest whitepaper that dives deep into deploying HCL Sametime 12.0.2 on Kubernetes. 🌟
🔍 What's Inside?
💥 Step-by-Step Instructions: Whether you're a seasoned Kubernetes pro or just starting out, my guide walks you through the entire process. No guesswork—just clear, concise steps.
💥 Single Linux Machine: Yes, you read that right! You don't need a complex infrastructure. A single Linux server is all it takes to get started.
💥 All-Inclusive: Everything you need including the database, LDAP and TURN

📖 Get Started Now! Read the Whitepaper https://lnkd.in/gZzHwHm4

HCLSoftware U Whitepaper: Deploying Sametime 12.0.2 on Kubernetes Single Server - Customer Support

HCLSoftware

Features

12.0.3

Customization

Q&A

TURN

HCLSoftware

hcltechsw.com