OPENNTF WEBINARS

July, 2022 OpenNTF Webinar

Sametime 12 on Docker - A Deployment Walkthrough

AGENDA

- Welcome
- Presentation – Tony Payne, HCL
- Q and A - All

THANKS TO THE OPENNTF SPONSORS

- HCL made a contribution to help our organization
  - Funds these webinars!
  - Contests like Hackathons
  - Running the organization
- Prominic donates all IT related services
  - Cloud Hosting for OpenNTF
  - Infrastructure management for HCL Domino and Atlassian Servers
  - System Administration for day-to-day operation

THIS IS OUR COMMUNITY

- Join us and get involved!
- We are all volunteers
- No effort is too small
- If your idea is bigger than you can do on your own, we can connect you to a team to work on it
- Test or help or modify an existing project
- Write guides or documentation
- Add reviews on projects / stars on Snippets

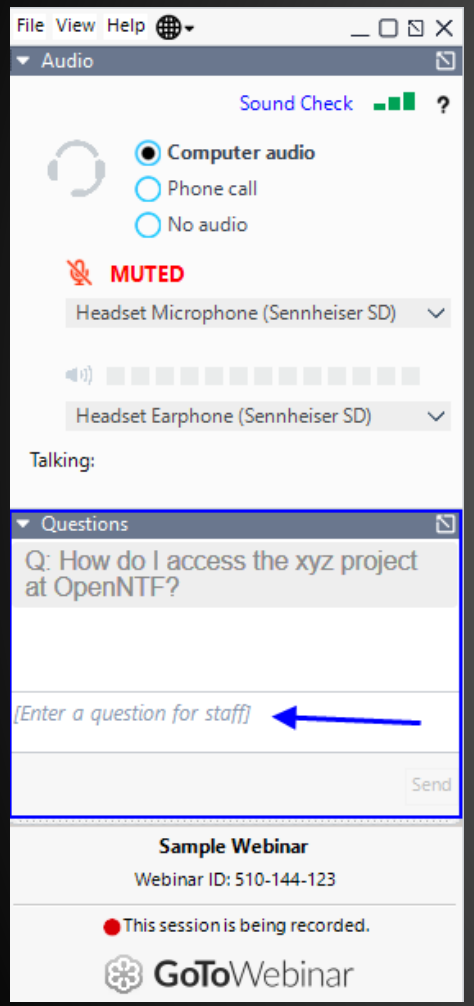
OPENNTF BOARD UPDATES

- Community Projects
  - Catalog of User Group Presentations
  - Led by Oliver Busse
  - Channel on slack.openntf.com #presentation-project
- The Future of OpenNTF
  - How to Evolve OpenNTF
  - We want your input!
  - Blog and video posted soon
  - Feedback via Discord

NEXT WEBINAR

- Watch https://www.openntf.org/webinars for more information

ASKING QUESTIONS



- First Question – Will this be recorded?
  - Yes, view on YouTube!!!
  - https://www.youtube.com/user/OpenNTF
- Use the Questions Pane in GoToWebinar
- We will get to your questions at the end of the webinar
- The speakers will respond to your questions verbally
  - (not in the Questions pane)
- Please keep all questions related to the topics that our speakers are discussing!!!
- Unrelated Question => post at:
  - http://openntf.slack.com/

SAMETIME 12 ON DOCKER - A DEPLOYMENT WALKTHROUGH

Tony Payne

HCL Software Academy
for Digital Solutions

HCL

HCL Sametime

HCL Sametime Premium Docker Deployment Walkthrough

HCL SOFTWARE

Goals of Todays Walkthrough

HCL Sametime

Goals

By the end of todays walkthrough:

- Understand Sametime 12 Topology and requirements
- Understand Migration requirements
- Overview of Configuration and Troubleshooting
  - Especially as it relates to the containerized Community and Proxy Components
- Docker Deployment – step by step!
  - Install Mongo
  - Install Docker and docker-compose
  - Deploy Sametime Premium

Along the way we'll share lessons learned and best practices tips and tricks

HCL Sametime

New Whitepaper!

- DS Academy Whitepaper: Deploying HCL Sametime v12 on Docker
  - https://support.hcltechsw.com/csm?id=kb_article&sysparm_article=KB0099251
- Sametime 12 FP1!
  - Contains many important fixes
  - Can upgrade an existing 12 environment or deploy full environment
    - Follows the same installation steps.

Deployment
Migration
Configuration

HCL Sametime

Deployment

What's it mean

- Sametime vs Sametime Premium
- "Fully Containerized"
- no Domino Dependencies?
  - What if we currently use Native Domino Directory?

Topologies

- Chat Only (Limited Use)
- Premium

Deployment

- Docker
- K8S

Configuration

Migration

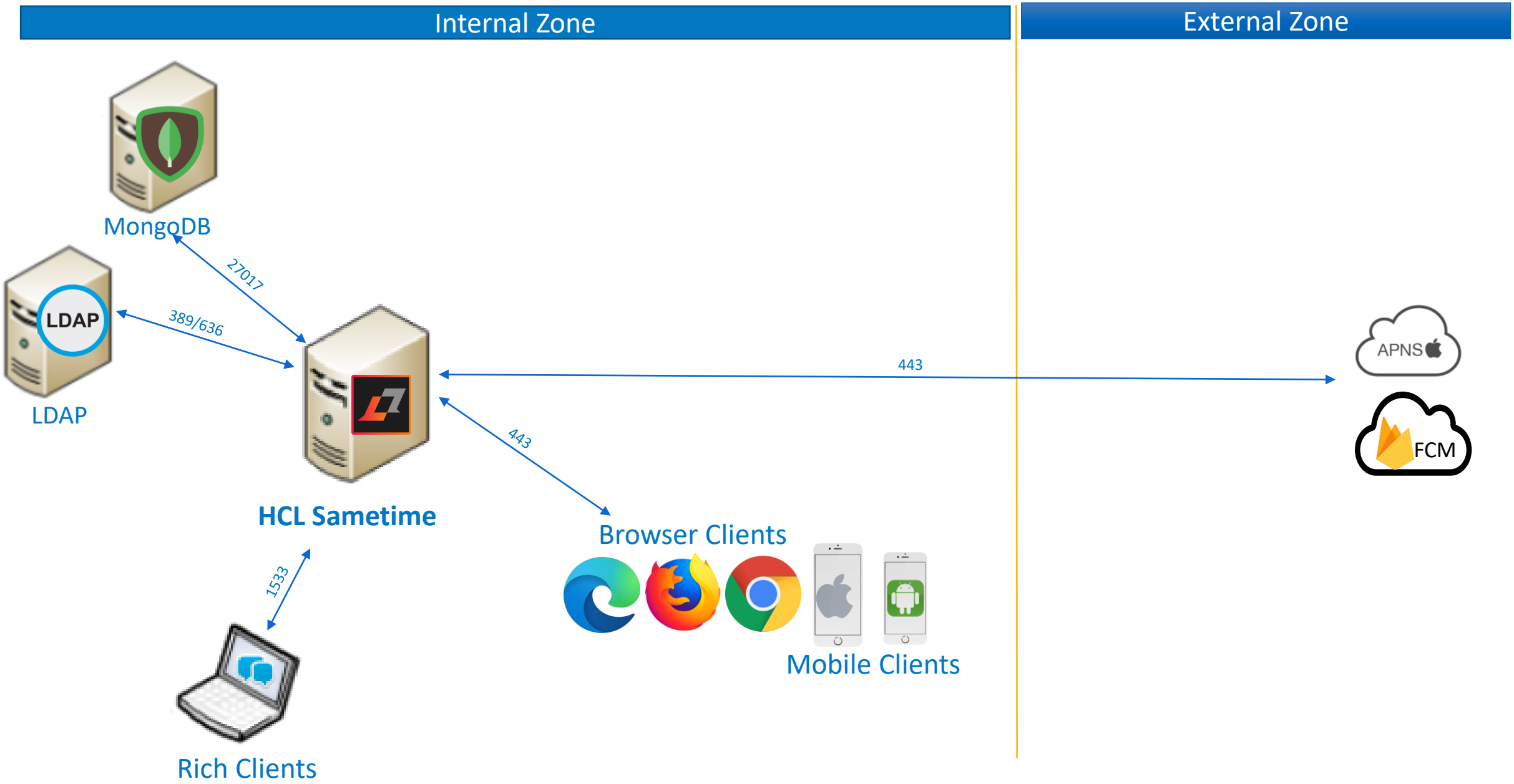
HCL Sametime

A few other thoughts as we get started

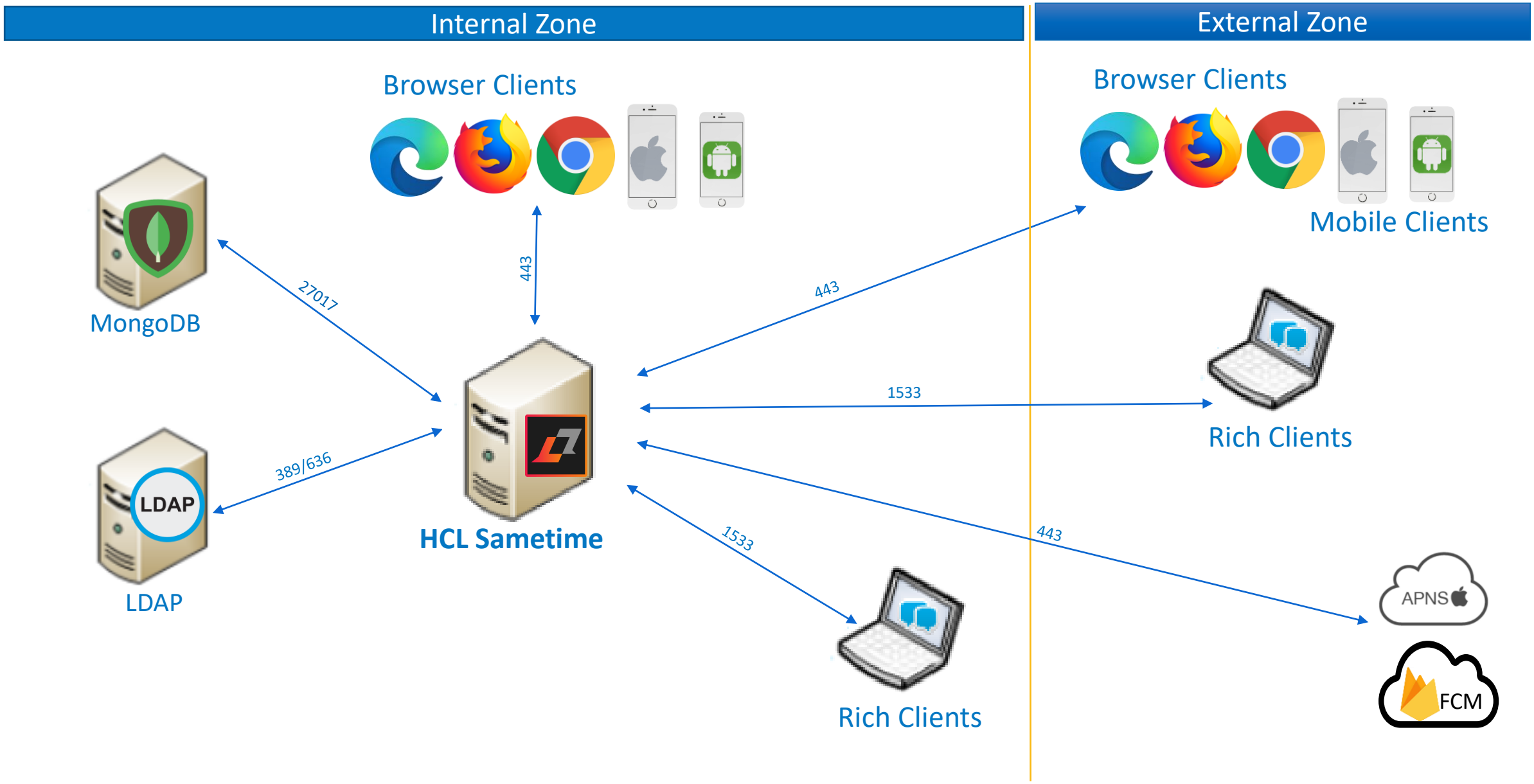
- Persistent Chat is now required. We no longer support disabling it.
  - If you do not want the ‘persistence’ of it,
    - you can set the TTL (‘time to live’) to 1 (day).
- Difference in installation between Sametime and Sametime Premium
- Upgrade Options

Topologies

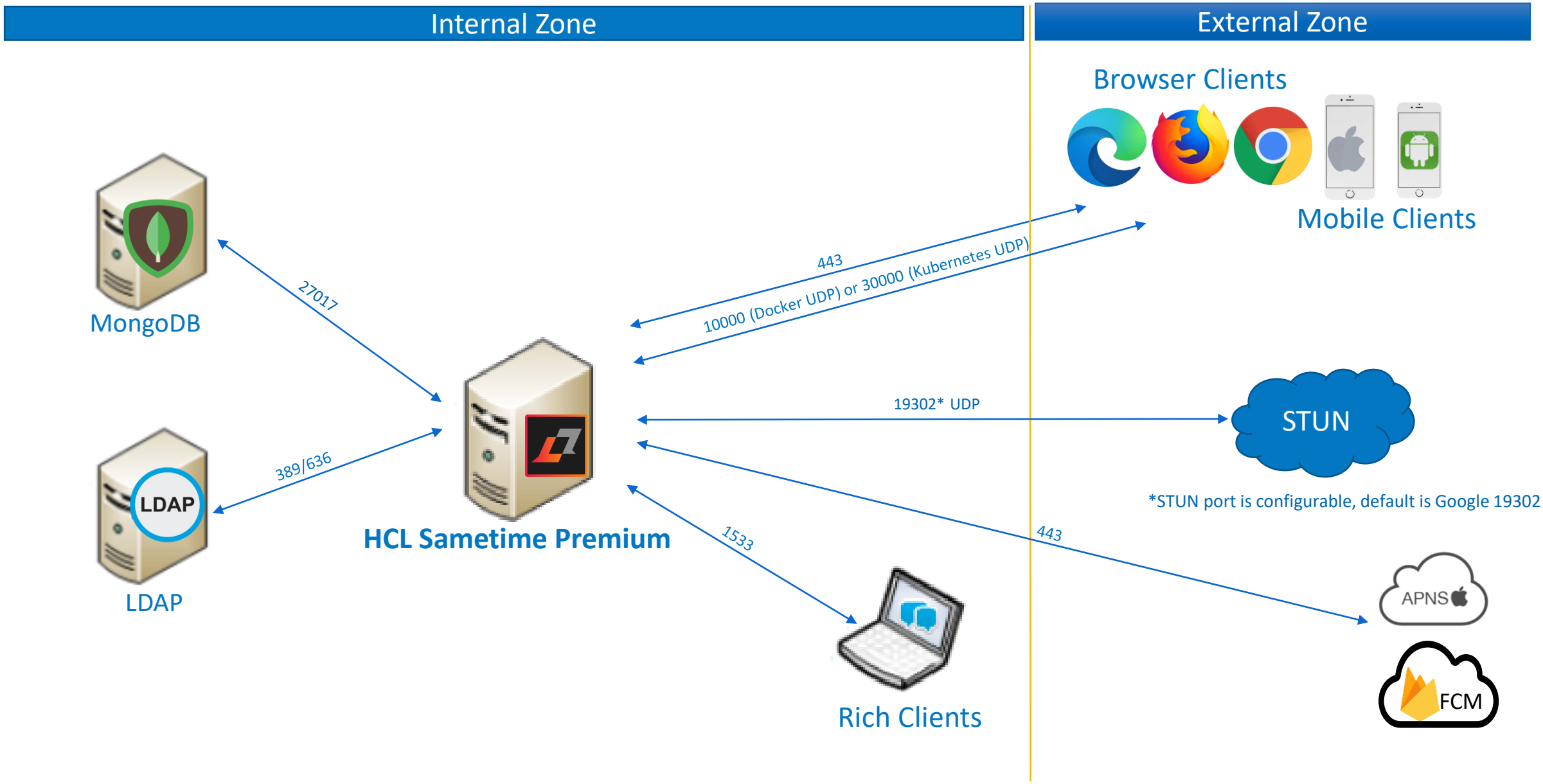
HCL Sametime - Basic Internal Deployment



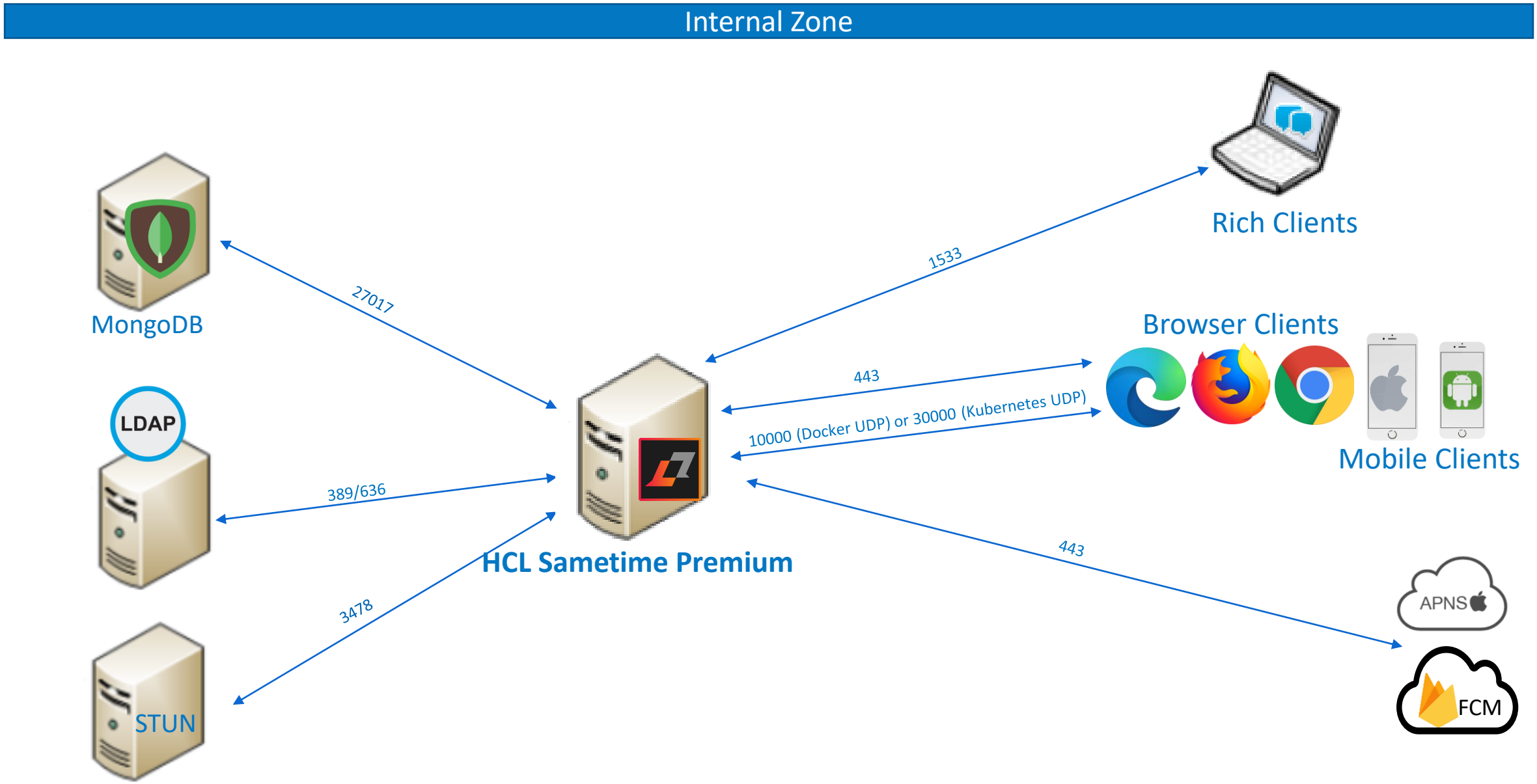
HCL Sametime with Internet



HCL Sametime Premium with Internet



HCL Sametime Premium Internal Only



Sametime Premium Network Considerations

Installed and Embedded Clients use 1533 for Community services

Web Clients use 443 to access the Chat, Mobile and Meeting Services

Clients use UDP port 30000 (Kubernetes) or UDP Port 10000 (Docker) for the Media Streams

- By default, Direct access to the Video node is required
- Server reaches out to STUN server(s) to determine IP addresses when needed
- Can use TURN as a relay if desired.

Let’s talk about STUN

Session Traversal Utilities for NAT (**STUN**) 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.

Meeting Server reaches out to STUN servers to discover IP addresses when configured

- IF everyone in the call is on the same network (internal – no NAT), may not be needed
- Given today’s needs with most working remotely, it will likely be needed
- Can use other STUN server(s) if needed

NOT a relay (TURN)

- TURN is fully supported with V12.

Let’s talk about TURN

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 Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

- It is most useful for clients on networks masqueraded by symmetric NAT devices.
- Uses STUN to gather the IP address of the server AND the clients
- It inserts itself as a relay in the SIP/SDP negotiation
- If direct connectivity is not possible, then the traffic can flow thru TURN
- CoTurn is a common open source server that is easy to install and flexible
  - https://github.com/coturn/coturn/wiki/turnserver
- The install and the prepareDeployment scripts will prompt you if you want to use TURN
- More Information: https://help.hcltechsw.com/sametime/11.6/admin/turnserver_intro.html

Let’s talk about how the client gets media streams

Most MeetingServer issues come down to Network connectivity

Lets walk thru how media streams work

- MeetingServer discovers IP address thru STUN
- When second person joins or a recording is started
  - The meetingserver provides that discovered IP address to the clients as part of a SIP/SDP ‘offer’
  - Clients then start sourcing UDP packets to that address using the appropriate port (10000/30000)
  - The Server has to be able to send the packets BACK to the clients – this is over an ephemeral port
    - This is the same port that the clients used to source data to the server.
    - In Firewall terms – this is a ‘pinhole’ – allow traffic to be sent back to the client on the port that the client used to access the server.

Let’s talk about how the client gets media streams

How do we determine the address that a given Video node is providing to the clients?

- Docker
  - docker-compose logs jvb | grep –i discovered
  - DOCKER_HOST_ADDRESS
- Kubernetes
  - kubectl get po –o wide
  - kubectl describe node <video>
  - for i in $(kubectl get po -o wide | grep video | awk ' { print $7 } '); do kubectl describe node $i | grep ExternalIP; done

Keep in mind that the IP we think the node should be using may NOT be the same IP that STUN returns

- Especially true in situations where the nodes are routing thru a firewall or NAT device

Installation

Installation Summary

Know your LDAP environment

Install MongoDB

- Configure MongoDB

Install Sametime or Sametime Premium

Hardware and Software requirements

Minimum hardware specs

- Chat Only Server - 4 core, 32gb, 200gb HD
- Premium Server – 8 core, 32gb 500gb* HD

Supported Linux versions

- RHEL/CentOS 7.9 and 8.2

PreRequisites

- MongoDB
- LDAP

Deployment - MongoDB

With V12, we support MongoDB 4.4

If upgrading , can continue to use your existing MongoDB deployment

- New installs should start with MongoDB 4.4 – many advantages

CentOs/Rhel installs

- If using Yum to install MongoDB by creating a repo file
    - Make sure the URL is correct, copy/paste tends to leave out a '-' https://docs.mongodb.com/manual/tutorial/install-mongodb-on-red-hat/

ReplicaSet – Network considerations

- Keep in mind that the replicaSet 'host' must be resolvable by the Community, Proxy and Catalog components
- In a multi-node replicaset, make sure all hosts are resolvable and reachable.
- Validate that the mongoURL is correct.

Deployment - MongoDB

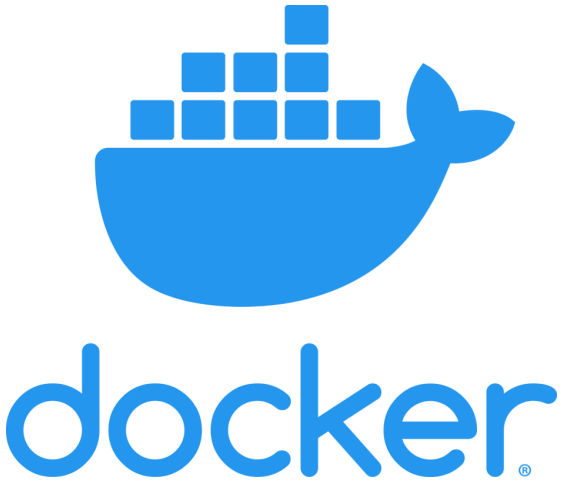
Most Common issues

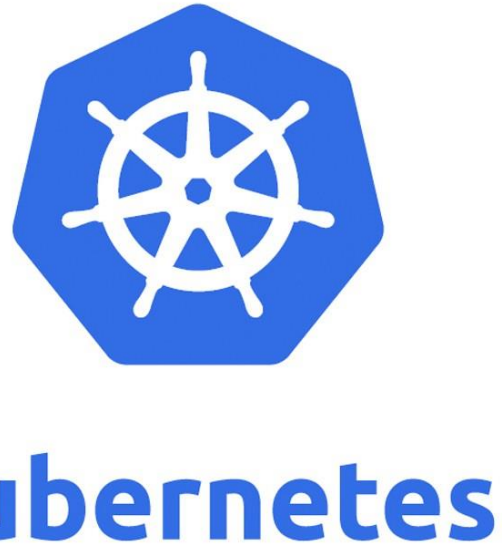
- Forgot to enable the replication set
  - Use ‘rs.status()’ command to validate
  - Hint: if you open a mongo shell and do not see rs0:Primary, this is your problem
  - This is specifically enabled in the mongod config file that runs the service:
  - mongod.cfg on Windows, /etc/mongod.conf on linux
- Hostname resolution issues for the replication set
  - Use rs.conf() output to help understand the issue

Sametime Installation

Deployment Models:

- Docker
- Kubernetes





Sametime on Docker

Intended for a small deployments and POC type environments

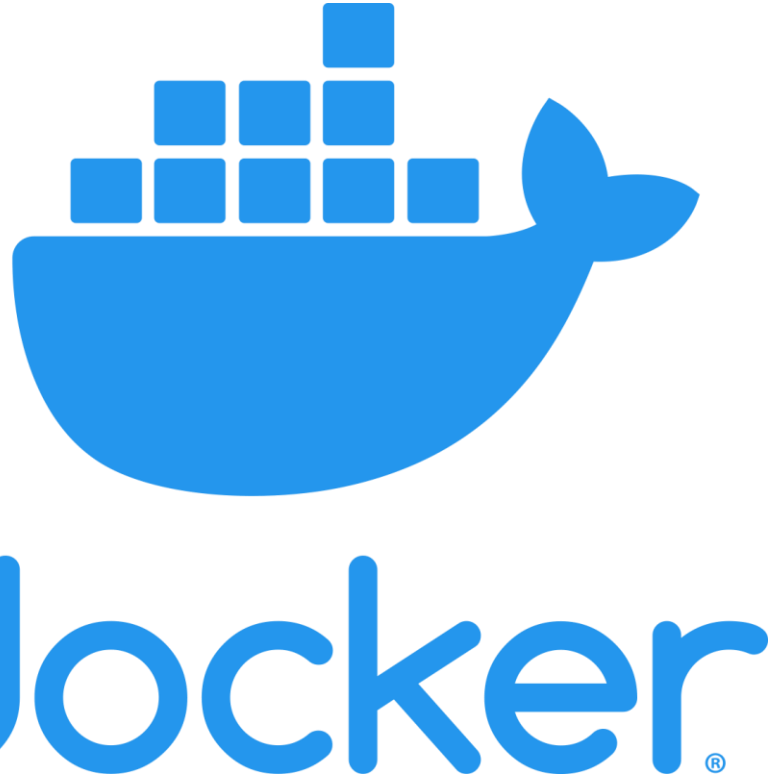
Expected for most ‘chat only’ or limited use environments

Limited number of Meetings users

Easiest to install and configure

Fully supported for production

Does not support automatic scaling, limited number of ‘recorders’

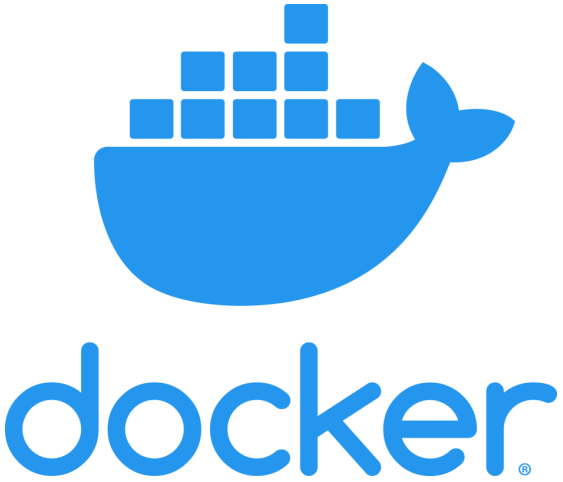


Docker is a set of platform as a service products that uses OS-level virtualization to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels.

Sametime On Docker

Installation:

- Install Docker and docker-compose
- Unzip sametime_premium.zip
- Run install.sh
    - This will also handle any upgrades
- Configure the Virtual Sound Driver



Sametime on Kubernetes

Intended for production deployments

Supports autoscaling thru Kubernetes

Requires a customer to have a full Kubernetes environment and experience

- See also “Kubernetes Quick Start”

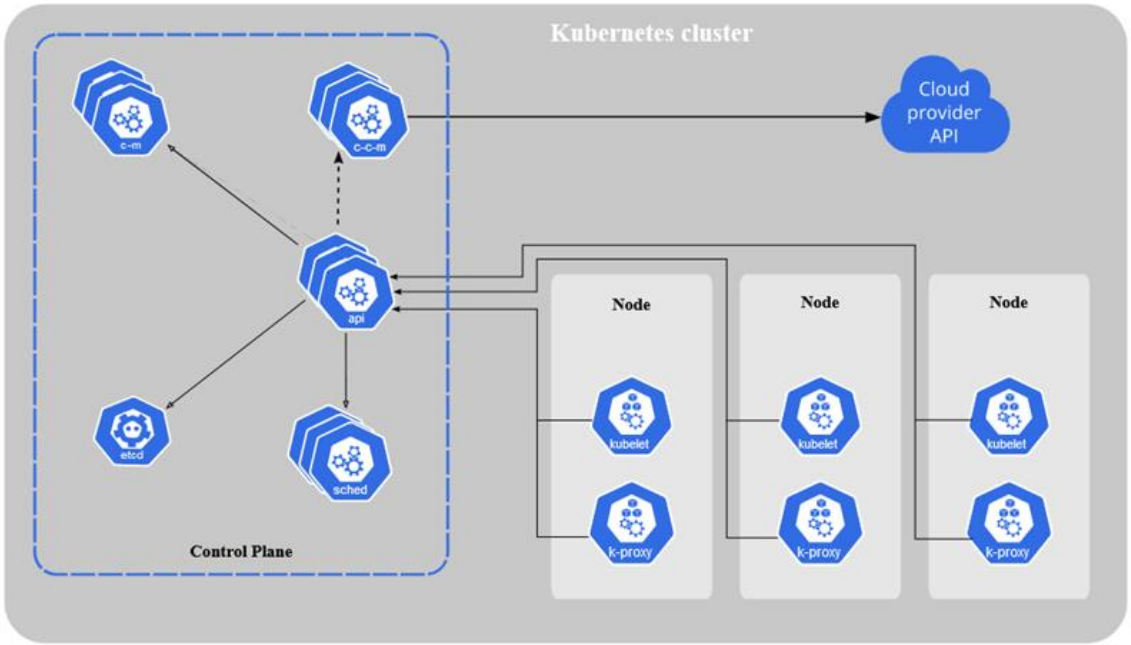
This is the suggested and recommended deployment model

kubernetes

Kubernetes is an open-source container-orchestration system for automating application deployment, scaling, and management. It was originally designed by Google and is now maintained by the Cloud Native Computing Foundation

What is a Kubernetes Cluster?

- When you deploy Kubernetes, you get a cluster.
- A Kubernetes cluster consists of a set of worker machines, called nodes, that run containerized applications. Every cluster has at least one worker node.
- The worker node(s) host the Pods that are the components of the application workload.
- The control plane manages the worker nodes and the Pods in the cluster.
- In production environments, the control plane usually runs across multiple computers and a cluster usually runs multiple nodes, providing fault-tolerance and high availability.



kubernetes

Sametime on Kubernetes

PreRequisites

- Kubernetes v1.22.0 or later* with an ingress controller
- Helm v3.1.2

Installation

- Unzip the sametime_meetings.zip
- Deploy the docker images (./load.sh)
- Prepare the Deployment (./prepareDeployment.sh)
    - This will also manage the upgrade!
- Create the recordings volume
- Deploy the helm chart (helm install/upgrade sametime-meetings .)

kubernetes

Sametime on Kubernetes

HCL Sametime 11.6 - Full stack Kubernetes Implementation

- https://support.hcltechsw.com/csm?id=kb_article&sysparm_article=KB0092467

Deploying HCL Sametime Meetings on AWS Elastic Kubernetes Service (EKS)

- https://support.hcltechsw.com/csm?id=kb_article&sysparm_article=KB0085515

Deploying HCL Sametime Meetings on Google Kubernetes Engine (GKE)

- https://support.hcltechsw.com/csm?id=kb_article&sysparm_article=KB0096200

kubernetes

Migration

Migration overview

Know your Directory

- LDAP is required.
- If using Domino Native directory
  - Must convert to LDAP

VPUserinfo database must be migrated to MongoDB

- notes-migration.zip

Like previous Migrations

- Use DNS to flip to the new environment.

Migration overview

Converting From Domino Native Directory to LDAP

- Stnamechange Utility
  - https://help.hcltechsw.com/sametime/11.6/admin/changing__names_in_contact_and_privacy_lists.html
  - You will use the “LDAP” Task to convert to LDAP
  - This is run on the current 11.x or older Community server per doc
  - Use this opportunity for any other cleanup if needed.
- There will be an updated version for v12 post v12.

Migration overview

VPUserinfo database must be migrated to MongoDB

- On the Sametime server you wish to migrate from
  - Unzip notes-migration.zip
    - Linux
      - source ./setenv.sh
        - This sets up the environment
      - ./notes-migration.sh
        - Will prompt for locations and options and then run the tasks.
    - Windows
      - notes-migration.bat
        - Will prompt for locations and options and then run the tasks

Configuration

Configuration overview

Community Configuration files

- sametime.ini
  - This is generated on the fly based on values in custom.env or values.yaml
  - Settings can be over-ridden if needed
- StCommunityConfig.xml
  - This used to be ‘stconfig.nsf’
- UserInfoConfig.xml
- Policies.user.xml
- Policies.server.xml
- Clustering
  - Only applicable to Kubernetes environments

Configuration overview

Adding sametime.ini settings to configuration

- Any sametime.ini setting can be added, the format is
  - STI__SECTION__SETTINGNAME=VALUE
  - For example, the format for

    [config]

    ST_COMMUNITY_ID

    Would be

    STI__CONFIG__ST_COMMUNITY_ID
- Keep in mind that most settings are now defaulted and many may not apply any longer, so use caution
- Docker
  - Add these variables to the custom.env file and restart to take affect
- Kubernetes
  - Add these variables to the values.yaml file, Run ‘helm upgrade’ and scale the Community pod

Configuration overview

Updating Configuration Values On Docker

- Copy the existing files out of the container:

  docker cp <container_name>:/local/notesdata/UserInfoConfig.xml .

  docker cp <container_name>:/local/notesdata/StCommunityConfig.xml .

- This will put them in the current directory
- Make needed changes, then update docker-compose.yaml by adding:

```
volumes:
 - ./StCommunityConfig.xml:/local/notesdata/StCommunityConfig.xml
 - ./UserInfoConfig.xml:/local/notesdata/UserInfoConfig.xml
networks:
 - sametime.test
```

Restart the server to take affect.

Do not have to do both files, only need to do this for the specific configuration update required.

Configuration overview

Updating Configuration Values On Kubernetes

- Copy the existing files out of the container:

  kubectl exec -it <podID> --container community -- cat StCommunityConfig.xml >StCommunityConfig.xml

  kubectl exec -it <podID> --container community -- cat UserInfoConfig.xml >UserInfoConfig.xml

  This will put them in the current directory
- Place the files in a directory called “extra-community-config”
- Make needed changes, then create a secret:
  - kubectl create secret generic extra-community-config --from-file=./
- Modify values.yaml to use this secret by adding
  - overrideCommunityConfigSecret: extra-community-config
- Run ‘helm upgrade’ and scale the Community pod
- To update the configuration:
  - kubectl delete secret extra-community-config
  - kubectl create secret generic extra-community-config --from-file=./
- And scale the Community Pod

Both files are needed on Kubernetes, even if only updating one.

Configuration overview

Updating Policy Settings On Docker

- Copy the existing files out of the container:

```
docker cp <container_name>:/local/notesdata/policies.user.xml .
docker cp <container_name>:/local/notesdata/policies.server.xml .
```

- This will put them in the current directory
- Make needed changes, then update docker-compose.yaml by adding:

```
volumes:
    - ./policies.user.xml:/local/notesdata/policies.user.xml
networks:
    - sametime.test
```

Restart the server to take affect.

Do not have to do both files, only need to do this for the specific policy update required.

Configuration overview

Updating Policy Settings On Kubernetes

- Copy the existing files out of the container:

  kubectl exec -it <podID> --container community -- cat /local/notesdata/policies.user.xml > ./policies.user.xml

  kubectl exec -it <podID> --container community -- cat /local/notesdata/policies.server.xml > ./policies.server.xml

  This will put them in the current directory
- Place the files in a directory called “extra-community-policy”
- Switch to the extra-community policy directory and make needed changes
- Then create a ConfigMap:

  kubectl create configmap extra-community-policy --from-file=/.

  - Modify values.yaml to use this secret by adding

    overrideCommunityPolicy: extra-community-policy

  - Run ‘helm upgrade’ and scale the Community pod
- To update the policy:

  kubectl delete cm extra-community-policy

  kubectl create configmap extra-community-config --from-file=./

  - And scale the Community Pod

Both files are needed on Kubernetes, even if only updating one.

Configuration overview

Community Clustering

- Only Applies to Kubernetes environments
- In values.yaml, set

  numberOfCommunityServers:

  To the number of required servers
- Run ‘helm upgrade’ and the number of servers will be started.
- Ensure the appropriate sizing of the nodes
- There is already a separate ‘mux’ pod, It will automatically detect the new nodes and route traffic.
  - Same for the Proxy Pod.

Configuration overview

Configuring TLS for LDAP Connections on Docker

- Create a tlsldap.env with the following setting and values appropriate for your configuration

  STI__Config__STLDAP_TLS_TRUST_STORE_TYPE=p12

  STI__Config__STLDAP_TLS_TRUST_STORE_FILE=/local/notesdata/ ldaptruststore.p12

  STI__Config__STLDAP_TLS_TRUST_STORE_PASSWORD=keystorepass

- Add tlsldap.env to the environment file variable in to the community section of docker-compose.yaml

  env_file:

  - tlsldap.env

- Map the keyfile into the container

  volumes:

  - ./ ldaptruststore.p12:/local/notesdata/ ldaptruststore.p12

- Restart the server
- NOTE: Ensure appropriate settings are in StCommunityConfig to enable secure LDAP connections

Configuration overview

Configuring TLS for LDAP Connections on Kubernetes

- Create a secret that contains your truststore

  kubectl create secret generic ldapconfigsecret --from-literal=KeyStorePassword=samet1me --from-file=./ldaptruststore.p12

- In values.yaml add

  ldapConfigSecret = ldapconfigsecret

- Run ‘helm upgrade’ and scale the Community Pod

Configuration overview

Configuring LTPA on Docker

- In the .env file, update the following section as appropriate

  ENABLE_LTPA=false

  LTPA_KEYS_FILE_PATH=${CONFIG}/auth/ltpa.keys

  LTPA_KEYS=

  LTPA_KEYS_PASSWORD=

- The {$CONFIG} variable point to the ‘sametime-config/auth’ folder.

Configuration overview

Configuring LTPA on Kubernetes

- This assumes you have LTPA keys from WAS or elsewhere you want to use.
- Create a secret that contains your ltpa keys

  kubectl create secret generic ltpa-keys --from-file=./ltpa.keys
- Edit the global config to add a the ltpa.keys password (base64 encoded)

  kubectl edit secret sametime-global-secrets

  Add

  LtpaKeysPassword : (base64 encoded value of the ltpa.keys password)
- In values.yaml set

  enableLtpa = true
- Run ‘helm upgrade’ and scale the Community Pod

Configuration overview

One way to generate LTPA Keys if you need them

- Use docker to run an instance of Websphere:

docker run -d -p 9080:9080 -p 9443:9443 websphere-liberty:latest

- Then copy the ltpa.keys from that instance:

docker cp cd44213f7519:/output/resources/security/ltpa.keys ./ltpa.keys

- The default password that WAS/liberty uses is "WebAS"

Configuration overview

Configuring SAML Trust Store and Fork on Docker

- Create a saml.env with the following setting and values appropriate for your configuration

  STI__Config__STSAML_TRUST_STORE_TYPE=p12

  STI__Config__STSAML_TRUST_STORE_FILE=/local/notesdata/samltruststore.p12

  STI__Config__STSAML_TRUST_STORE_PASSWORD=keystorepass

- Add saml.env to the environment file variable in to the community section of docker-compose.yaml

  env_file:

  - saml.env

- Map the keyfile into the container

  volumes:

  - ./samltruststore.p12:/local/notesdata/samltruststore.p12

- Restart the server

Configuration overview

Configuring SAML Trust Store and Fork on Kubernetes

- Create a secret that contains your truststore

  kubectl create secret generic samlconfigsecret --from-literal=KeyStorePassword=samet1me --from-file=./samltruststore.p12

- In values.yaml add

  samlConfigSecret = samlconfigsecret

- Run ‘helm upgrade’ and scale the Community Pod

Configuration overview

Configuring TLS Connection for Rich Clients On Docker

- Create a mux.env with the following setting and values appropriate for your configuration

  STI__Debug__VPMX_DISABLE_CONFIGURATION_UPDATE=1

  STI__Debug__VPMX_PORT=1533

  STI__Debug__VPMX_TLS_PORT=1533

  STI__Config__VPMX_CAPACITY=20000

  STI__Config__ST_TLS_KEY_STORE_TYPE=p12

  STI__Config__ST_TLS_KEY_STORE_FILE=/local/sametimemuxdata/keystore.p12

  STI__Config__ST_TLS_KEY_STORE_PASSWORD=keystorepass

- Add mux.env to the environment file variable in to the mux section of docker-compose.yaml

  env_file:

  - mux.env

- Map the keyfile into the container

  volumes:

  - ./keystore.p12:/local/sametimemuxdata/keystore.p12

- Restart the server

Configuration overview

Configuring TLS Connection for Rich Clients On Kubernetes

- Create a secret that contains your keystore

  kubectl create secret generic mux-secret --from-literal=KeyStorePassword=samet1me --from-file=./keystore.p12

- In values.yaml add

  muxTlsConfigSecret=mux-secret

- Run ‘helm upgrade’ and scale the Mux Pod

TroubleShooting

Debug overview

Enabling Debug for the Community Pod

- Docker
  - add the debug.env to the community section of docker-compose.yaml
- Kubernetes
  - Values.yaml - enableCommunityDebug

Accessing the logs thru kubectl/docker-compose

- docker-compose logs community
- kubectl logs deploy/community --container community

Accessing the container thru kubectl/docker-compose

- docker exec -it hcl_community_1 bash
- kubectl exec -it <podID> --container community – bash

In K8S environments

- Take advantage of the Monitoring and Statistics whitepaper!
  - https://support.hcltechsw.com/csm?id=kb_article&sysparm_article=KB0088100

HCL Sametime Docker Troubleshooting

Confirm docker and docker-compose versions

- docker version
- docker-compose version

How to gather logs and work in Docker Containers

- All about docker-compose.yml
- About that jitsi-config folder
- docker images
- docker ps
- docker-compose logs <component>
  - | logs.txt
- docker-compose exec <component> /bin/sh

Restarting the Server and applying configuration changes

- docker-compose down
- Docker-compose up -d

HCL Sametime Meetings Troubleshooting

How to gather logs and other details

- kubectl get po
- kubectl logs [po-name]
- kubectl logs deploy/video | grep -i discover
- kubectl describe [po-name]
- helm commands

Helpful helm commands

- helm install sametime .
- helm uninstall sametime
- helm upgrade sametime .
- helm rollback
- helm history
- helm list
- helm edit deploy <name>

HCL Sametime Meetings Troubleshooting

How to gather logs and other details

- kubectl get po
- kubectl logs [po-name]
- kubectl logs deploy/video | grep -i discover
- kubectl describe [po-name]
- helm commands

Helpful helm commands

- helm install sametime .
- helm uninstall sametime
- helm upgrade sametime .
- helm rollback
- helm history
- helm list
- helm edit deploy <name>

HCL Sametime Meetings Troubleshooting

OOPS !

- All authentication goes thru the Community Server via the Sametime Proxy
- First thing to check will be that Community has connectivity to LDAP and MongoDB

Must Gather:

- Community Logs
- HAR file from browser

Most Common Problems:

- LDAP and Mongo Connectivity

HCL Sametime Meetings Troubleshooting

Can’t Create a meeting room

- Validate network access to MongoDB

Must Gather:

- MeetingServer catalog, nginx and auth logs
- HAR file from browser

Most Common Problems:

- Can’t reach MongoDB server
  - Validate DNS, add extra-hosts if needed
  - Authentication issues
  - Rs.conf()
- Policy not updated on Community Server

HCL Sametime Meetings Troubleshooting

Can’t join a meeting (stuck on loading)

- docker_host_address
- STUN Connectivity
- Public IP Address of MeetingServer
- WebSockets

Must Gather:

- MeetingServer jvb logs
- HAR file from browser
- Console Screenshot from browser

Most Common Problems:

- Docker_host_address is wrong/not set
- Public IP is not reachable
- Port 10000 (docker) or 30000 (kubernetes) UDP is blocked
- Websockets are blocked

HCL Sametime Meetings Troubleshooting

Can’t record a meeting – or meeting fails with > 2 people

- Docker_host_address
- STUN Connectivity
- Public IP Address of MeetingServer
- Firewall Configuration
  - Inbound is open, but what about outbound?

Must Gather:

- MeetingServer jvb and jibri logs
- HAR file from browser

Most Common Problems:

- Docker_host_address is wrong/not set
- Public IP is not reachable
- Port 10000 (docker) or 30000 (kubernetes) UDP is blocked
- Virtual Sound Driver not installed (Documented step)

www.hcltech.com

$10 BILLION | 159,000+ IDEAPRENEURS | 50 COUNTRIES

QUESTIONS?

Use the GoToWebinar Questions Pane

Please keep all questions related to the topics that our speakers are discussing!!!

Unrelated Question => post at:

http://openntf.slack.com/