Administrator's Guide for Zimbra Chat

This document is applicable for Zimbra Daffodil version 10.1.7 onwards.

- License
- Introduction To Zimbra Chat
- Basic Chat Vs. Advanced Chat Comparisons
- Licensing Guide
  - For New Customers (Starting with 10.1.7 or later)
  - For Existing Customers Already on Zimbra 10.1.x version older than 10.1.7
  - For Customers on Older Versions (8.8.15, 9.0.x, 10.0.x)
  - For Customers with existing Zimbra Talk Entitlement (8.8.15, 9.0.x)
- Things to know before installing Zimbra Chat
- Zimbra Chat Architecture Overview
- Scaling and Infrastructure Guide
- Installation Guide
  - Prerequisite
  - Install Zimbra Chat modules
  - Install Zimbra Chat server
  - Configure Chat server
  - Configure LDAP attributes
  - Create a realm for an existing domain
  - Enable Basic Chat
  - Enable Advanced Chat
  - (Optional) Multiple domains and multiple chat servers
  - (Optional) Install and enable PGroonga for full-text search in non-English languages on Advanced Chat
- Administrator Guide
  - Admin Console
  - Zimlet Configuration Guide for one to one chat
  - Log files
  - Zimbra Chat Administrator
- Backup and Restore For Zimbra Chat
- Troubleshooting Guide
  - 1. LDAP Connection Issues
  - 2. Realm Creation Issues
  - 3. Chat Availability Issues
  - 4. Account Provisioning Issues
  - 5. Chat Server Issues
  - 6. Chat Scaling Issues
  - 7. Zimbra Server Issues
- FAQs
  - General Questions
  - Admin-Specific Questions

License

Synacor, Inc., 2025
505 Ellicott Street, Suite A39
Buffalo, NY 14203
US

https://www.synacor.com

Introduction To Zimbra Chat

Zimbra Chat is a secure, real-time messaging solution integrated into the **Zimbra Daffodil (Version 10.1.7 onwards)**, designed to enhance team communication while adhering to Zimbra's core principles of privacy, data sovereignty, and user control. This chat service is offered in two distinct variants:

- **Basic Chat**: A lightweight one to one messaging solution, ideal for users who need quick, person-to-person communication embedded within the Modern UI.
- **Advanced Chat**: A full-featured messaging platform, enabling threaded group conversations, searchable archives, and additional collaboration tools. Built using an open source platform, this variant supports advanced use cases and is suited for organizations needing rich communication experiences.

Both variants are integrated directly into Zimbra's Modern Web Client (Modern UI) and support secure on-premises deployment.

Basic Chat Vs. Advanced Chat Comparisons

	Features	Basic Chat	Advanced Chat
1	Direct Messaging (One to one chat)	✅	✅
2	Notifications	✅	✅
3	User Presence	✅	✅
4	User Preference and customisation	Limited	✅
5	Localisation	Limited	Limited
6	Emoji support	✅	✅
7	Scalability (total users)	50k+	50k+
8	Multi-domain support	✅	✅
9	OS support	Ubuntu 22/24 Only	Ubuntu 22/24 Only
10	LDAP based authentication	✅	✅
11	Auto-provisioning	✅	✅

12	Platform support	Web-client only	Web-client only
13	Comprehensive documentations for users, admin	✅	✅
14	Message History search	❌	✅
15	Ad-hoc Groups	❌	✅
16	Channels	❌	✅
17	Topic based threading	❌	✅
18	File sharing and storage	❌	✅
19	Voice	❌	In-built integration with public instance of Jitsi
20	Expressive formatting - bold, italics, numbered lists, bullet lists, code syntax	❌	✅
21	Advanced features- e.g. send later, mark unread	❌	✅
22	Role based access control	❌	✅
23	Domain level Custom Branding	❌	✅
24	Import data from other chat products	❌	❌
25	Desktop application	❌	Roadmap
26	Mobile application	❌	Roadmap

Licensing Guide

Zimbra's new Chat solution offers two distinct tiers:

- **Basic One To One Chat** – free of charge as introductory pricing, lightweight messaging.
- **Advanced Chat** – paid tier with extended features.

To begin using Zimbra Chat, ensure your system is on Zimbra 10.1.7 version or later and follow the relevant guidance below.

Prerequisite

The Zimbra license must have the following entitlements enabled:

- **BasicOneToOneChatAccountsLimit** - To enable basic one to one chat.
- **ChatAccountsLimit** - To enable advanced chat.

Both entitlements work independently of each other.

> **Note**: Zimbra Chat is an add-on module and is not included in any existing editions or bundles. It must be purchased separately, even for customers with current Zimbra licenses. Please reach out to your Zimbra representative to get these entitlements.

For New Customers (Starting with 10.1.7 or later)

1. A license is issued as part of the onboarding process.
2. Chat features (Basic or Advanced) are included based on your selected plan.
3. Once licensed:
   - Chat features will be enabled automatically.
   - You'll receive the license key via email.
4. If you have not received chat functionality or wish to upgrade, please reach out to your Zimbra representative.

For Existing Customers Already on Zimbra 10.1.x version older than 10.1.7

Basic One to One Chat (Free)

- This is available at no cost but is not enabled by default.
- To enable:
  - Contact your Zimbra representative and request Basic Chat to be enabled for your license.
  - Once updated, restart your mailbox servers to activate the chat functionality.

Advanced Chat (Paid)

- Contact your Zimbra representative and upgrade your license to enable Advanced Chat.
- Upon license update:
  - Restart all mailbox servers to ensure chat extensions are loaded.
  - Chat will be available within your Zimbra interface.

> Tip: If you're upgrading from version 10.1.6 to 10.1.7:
>
> - If your license is updated before the upgrade, no restart of mailstores is required.
> - If the license is updated after the upgrade, you must restart all mailbox servers to ensure chat extensions are loaded.

For Customers on Older Versions (8.8.15, 9.0.x, 10.0.x)

1. Your older license (v2) must be migrated to the new format (v3) before you proceed with the upgrade.
   a. Please contact your Zimbra representative and request the v3 license to get the updated entitlements.
2. Upgrade to Zimbra 10.1.7 or later.
3. Once on a latest Zimbra version:
   - Basic Chat can be enabled free of cost.
   - Advanced Chat is available via a paid upgrade.

You will now be able to access the new Chat features.

For Customers with existing Zimbra Talk Entitlement (8.8.15, 9.0.x)

1. Your older license (v2) must be migrated to the new format (v3) before you proceed with the upgrade.
   a. Please contact your Zimbra representative and request the v3 license to get the updated entitlements.
2. Upgrade to Zimbra 10.1.7 or later.
3. Once on a latest Zimbra version:
   - Basic Chat will be enabled at no additional cost.
   - Advanced Chat is available under following conditions.
     - Advanced Chat Limit will by default be set to ZTalkAccountsLimit.
     - Available till Support end date for ZTalk SKU/contract.
     - New Zimbra Chat SKUs will be provided to such customers at no additional cost.

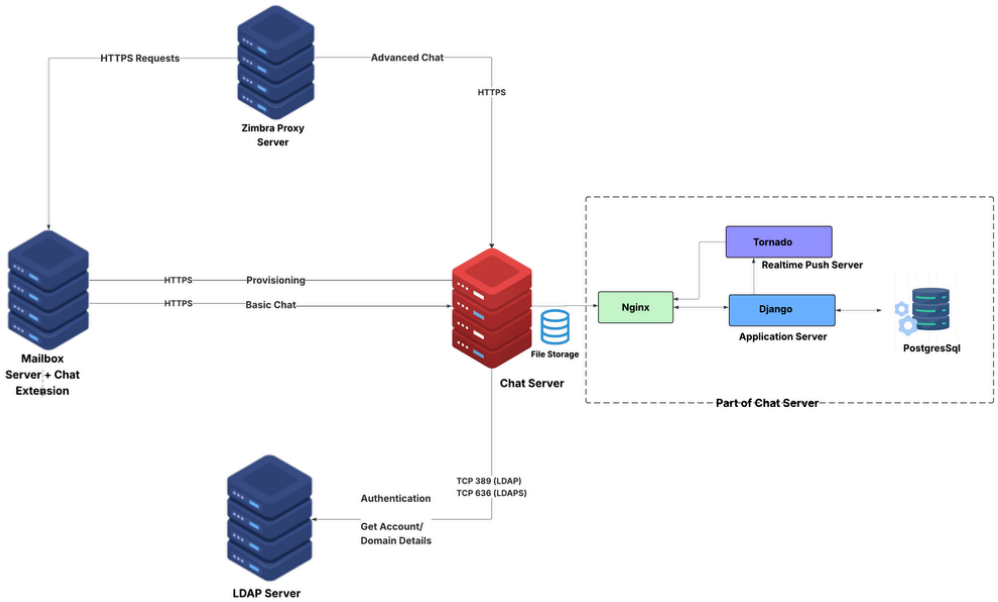
You will now be able to access the new Chat features.

Things to know before installing Zimbra Chat

Before proceeding with the installation, administrators should be aware of the following prerequisites and limitations:

1. **Separate Server Requirement**: Zimbra Chat must be installed on a dedicated server. It **cannot** be co-hosted with the Zimbra Collaboration Suite components.
2. **Version Compatibility**: Zimbra Chat is supported **only** on **Zimbra Daffodil version 10.1.7 and above**. Customers on earlier versions must upgrade their ZCS environment.
3. **Modern UI Only**: Chat is currently supported **only in the Modern UI**. Support for the Classic UI is under active development and is part of the **near-term roadmap**.
4. **LDAP Integration**: Zimbra Chat uses LDAP authentication from Zimbra Collaboration Suite(ZCS) for secure user login and provisioning.
5. **No Built-In Video**: This is a chat only solution, and **does not include video conferencing** solution. However, an integration with open source chat solution has been provided. For more details, refer to **Zimbra Chat Administrator → Video** section.
6. **No support for Migration**: There is **no support for migration** from earlier chat solutions like Zimbra Talk, or Hosted chat and video.
7. **Performance considerations**: While the solution is tested for 50k+ total users and upto 10k users on a single server, the actual performance might vary depending on usage patterns, online users, users per domain, infrastructure used for chat etc. Refer to the scaling guide for more information on best practices for scaling the solution effectively.
8. **Supported OS**: Only Ubuntu 22 and 24 are officially supported for installation of Zimbra Chat server.
9. **Add-On Model**: Zimbra Chat is not bundled with Zimbra Collaboration Suite(ZCS) by default—it is offered as an **add-on feature**, enabling customers to choose the level of functionality needed.

Zimbra Chat Architecture Overview



- **Mailbox Server + Chat Extension**: Initiates provisioning and Basic Chat requests via HTTPS.
- **Zimbra Proxy Server**: Routes HTTPS requests and forwards Advanced Chat requests to the Chat Server.
- **Chat Server**: Interacts with mailbox and LDAP servers, and forwards requests to application components.
- **LDAP Server**: Provides user authentication and domain/account information over LDAP/LDAPS.

Scaling and Infrastructure Guide

This section outlines scaling guidelines and configurations, offering recommendations for selecting the appropriate compute resources to ensure optimal performance, scalability across different deployment sizes.

Resource Sizing

- Active users include web clients performing passive background tasks (For example: `presence` for online status) against Zulip for Zimbra.
- Implement vertical scaling to fit machine resources for an expected active user count. The following sample sizes may require adjustment depending on the chat service activity levels.
- Note: Favor a dedicated PostgreSQL setup if significant growth in number of active users is anticipated.
  - **~500 Active Users Instance**
    - Provision compute instance for all-in-one - Zulip and PostgreSQL DB (4 vCPU, 16 GB RAM, 200 GB disk)
  - **~1k Active Users Instance**
    - Provision compute instance for Zulip's PostgreSQL DB (2 vCPU, 16 GB RAM,100 GB disk)
    - Provision compute instance for Zulip (4 vCPU, 32 GB RAM,200 GB disk)
  - **~3k Active Users Instance**
    - Provision compute instance for Zulip's PostgreSQL DB (2 vCPU, 16 GB RAM,200 GB disk)
    - Provision compute instance for Zulip (8 vCPU, 32 GB RAM,300 GB disk)
  - **~5k Active Users Instance**
    - Provision compute instance for Zulip's PostgreSQL DB (2 vCPU, 16 GB RAM,300 GB disk)
    - Provision compute instance for Zulip (12 vCPU, 32 GB RAM,400 GB disk)

Network Bandwidth

- When establishing network bandwidth guidelines between different systems, such as Zimbra, Zulip, and PostgreSQL, it's essential to consider several factors such as expected volume of traffic, the number of concurrent users.
- The following sample sizes may require adjustment depending on the specific size of your organisation, the volume of data exchanged, concurrent user load.

1. Network Bandwidth between Zimbra and Chat Server:

- **Recommended Bandwidth:**
- A minimum of **100 Mbps** would be suitable for small(upto ~500 Active Users) and medium-sized organizations(~500 to ~3k Active Users) where integration involves basic text messages.
- For larger deployments (more than 3k Active Users) where chat data is heavily used (for example- large message history, frequent queries), **1 Gbps** is recommended to handle high-frequency database requests and ensure low latency.

Note - Continuously monitor network performance and adjust bandwidth allocation as needed based on usage patterns and traffic spikes.

2. Network Bandwidth between Zulip and PostgreSQL Server:

PostgreSQL is often used as a backend database for chat, storing user data, messages, chat history, and configuration settings.

- **Recommended Bandwidth:**
- For smaller organizations(upto ~500 Active Users) with moderate database usage (for example: less frequent queries or fewer users), **100 Mbps** should be sufficient. This bandwidth can handle regular database queries and writes without causing performance degradation.
- For larger deployments(more than 3k Active Users) chat data is heavily used (for example: large message history, frequent queries), **1 Gbps** is recommended to handle high-frequency database requests and ensure low latency.

Note - Continuously monitor network performance and adjust bandwidth allocation as needed based on usage patterns and traffic spikes.

Scaling Process

When deploying multiple Zulip instances, consider the following approaches:

- **Balanced Distribution (Recommended):** Distribute domains evenly, by user count, across Zulip servers to optimize resource usage
- **Incremental Deployment (Acceptable):** Deploy domains gradually, adding new Zulip instances as needed

1. Local and General Zimbra Config Customisation
   - Adjust the relevant Zimbra options to support Zulip
2. For each Zulip instance
   a. Determine Zulip instance capacity and sizing
   b. Deploy and configure Zulip
   c. Provision domains and users
   d. Configure Zulip

Repeat step 2, deploying a fresh Zulip instance as more total user limit is needed.

See sections below for details.

Tornado Sharding

The Tornado service is responsible for handling real-time events in Zulip.

In general each Tornado process may account for approximately `1000` event queues. Thus in a scenario where there are more than `1000` consistently active users, multiple Tornado processes may be needed to handle the traffic.

For Example:

- 2k active users would expect about 2 to 3 tornado shards.
- 3k active users would expect about 3 to 4 tornado shards.

Shard by User Id

The Zulip config may be adjusted to shard tornado by user id. This may be useful for spreading load across a few tornado services for a Zulip instance.

Adjust and add the following to `/etc/zulip/zulip.conf` to shard user id on the local Zulip instance.

- In the following example: all users in `big-realm`, by user id, over 4 tornado services that will be accessible (by the subsequent refresh script) on ports 9800, 9801, 9802, and 9803:

```
[tornado_sharding]
9800_9801_9802_9803 = big-realm
```

- After modifying the configuration specified in above section, run the following scripts with `root` user to apply the changes, stand up the instances, update config files, refresh components, and restart nginx:

```
/home/zulip/deployments/current/scripts/zulip-puppet-apply
# accept prompt: y
/home/zulip/deployments/current/scripts/refresh-sharding-and-restart
```

You may then verify the new tornado services are accessible with `nc` on the configured ports on zulip server. For example: `nc -zv <hostname> <port_no>`

Tornado Sharding beyond 10 processes

Tornado sharding scales properly till 9 processes. Scaling tornado shards to 10 or more than 10 processes may fail with exception `TornadoQueueClient couldn't connect to RabbitMQ` in zulip's `/var/log/zulip/server.log`.

The reason being port 9810 was already in use by smokescreen.

Please follow below steps to resolve this issue:

- Stop the smokescreen with `zulip` user using `supervisorctl stop smokescreen`
- Start the chat services with `zulip` user using `/home/zulip/deployments/current/scripts/start-server`
- Check the status if all services are up with `zulip` user using `supervisorctl status`
- If you find any service in stopped state, please try bringing them up individually with `zulip` user. For example:
  - If `zulip-django` is not up, use `supervisorctl start zulip-django` to bring it up
  - If `zulip-tornado:zulip-tornado-port-9810` is not up, use `supervisorctl start zulip-tornado:zulip-tornado-port-9810` to bring it up.

Zimbra Configuration

1. Adjust the relevant **max_total_connections** options on each mailstore to support proxy traffic:
   `zmlocalconfig -e httpclient_internal_connmgr_max_host_connections=2000`
   `zmlocalconfig -e httpclient_external_connmgr_max_host_connections=2000`
   `zmlocalconfig -e httpclient_internal_connmgr_max_total_connections=2000`
   `zmlocalconfig -e httpclient_external_connmgr_max_total_connections=2000`
   `zmmailboxdctl restart`
   Note - May require further increases to accommodate increased proxy traffic to Zulip instances.
2. Adjust the `zimbraHttpNumThreads` config to support proxy traffic:
   `zmprov mcf zimbraHttpNumThreads 2000`
   Note - May require further increases to accommodate increased proxy traffic to Zulip instances.

Django

The Django service is responsible for Zulip's core APIs.

A single Django service is usually enough to handle the traffic for each of the supported sizes in a non-HA capacity. It may be necessary to increase the number of processes to speed things up in cases where the baseline incoming traffic drastically exceeds the processed traffic.

Configure Django

- Configure `/etc/zulip/uwsgi.ini` to fit Zulip instance resources
  - `processes`
    - Number of processes for web application.
    - Increasing this value increases the number of concurrent operations that may be processed by the web application, while also increasing the resource consumption of the web application.
    - Increase the number of processes to twice as the number of vCPU for zulip instance :
    - 2 * vCPU
  - `listen`
    - Size of listen queue.
    - Increasing this value increases the maximum number of requests that may be in the listen queue. Increasing this value beyond a size large enough to handle spike traffic is unlikely to be useful. If the maximum queue size is consistently hit, it indicates that the web application is processing requests too slowly to keep up with the incoming traffic's generation of requests.
    - May require adjustment to OS system configuration for `net.core.somaxcon`
- Restart the Zulip service using -
  - `su zulip -c '/home/zulip/deployments/current/scripts/restart-server'`
- Check the status of chat services using -
  - `su zulip -c 'supervisorctl status'`

Installation Guide

Prerequisite

To enable Basic Chat and/or Advanced Chat in Zimbra, ensure the following requirements are met:

License Activation

The Zimbra license must have the following entitlements enabled:

- **BasicOneToOneChatAccountsLimit** - To enable basic one to one chat.
- **ChatAccountsLimit** - To enable advanced chat.

Zimbra Collaboration Server (ZCS) Version Requirement

- Zimbra Daffodil (v10.1.7) or later

Zimbra Chat Server Hardware Requirements

Minimum Requirements:

- CPU: Dual-core processor (Intel/AMD)
- RAM: 2 GB
- Disk Space: 10 GB of free disk space

Recommended Requirements:

- CPU: Quad-core processor (Intel/AMD)
- RAM: 4 GB or more
- Disk Space: 20 GB or more (depends on expected data storage needs)

For more details refer to **Scaling and Infrastructure Guide** section.

Zimbra Chat Server Operating System Requirements

Compatible Linux Distributions:

- Ubuntu 22.04
- Ubuntu 24.04

Supported CPU Architectures:

- x86-64
- aarch64

Zimbra Chat Server Additional Requirements

- A dedicated machine or VM.
- The installer expects the Zimbra Chat server application (Zulip) and/or PostgreSQL to be the only thing running on the system; it will install system packages with `apt` (like nginx, PostgreSQL, and Redis) and configure them for its own use. We strongly recommend using either a fresh machine instance in a cloud provider, a fresh VM, or a dedicated machine.
- A connection to the Internet.

SSL Certificate Requirements

A valid SSL certificate is required for the Chat server to ensure secure HTTPS communication, allowing the Zimbra mailstore to access it correctly.

For more details refer: Administrator Guide → Install Chat server section.

Domain & URL Configuration

Suppose that Zimbra Chat server hostname and Zimbra domain have been applied as follows:

- Chat server hostname is chat1.mydomain.com
- Zimbra domain (the domain of an email address) is @domain1.example.com

The following requirements needs to be met:

- Chat server will be accessed on https (https://chat1.mydomain.com) for provisioning and some other purposes. The hostname chat1.mydomain.com must be resolvable by Zimbra proxy and mailstore servers.
- The Chat access URL for this domain will be: https://domain1examplecom.chat1.mydomain.com. The FQDN domain1examplecom.chat1.mydomain.com must be resolvable by proxy and mailstore servers.

Notes:

- DNS A record for domain1examplecom.chat1.mydomain.com should resolve to same IP as chat1.mydomain.com

Network & Firewall Configuration

- Hostname registered in DNS for the Chat server.
- Zimbra LDAP (or External LDAP or Active Directory, depending on your environment) hostname is resolvable by the Chat server.
- The Chat server hostname and FQDN for realms (domains) must be resolvable by Zimbra servers.
- Zimbra LDAP server allows 389 port access (or 636 port access if ldaps (LDAP over SSL) is used) from Chat server.
- Zimbra Proxy server allows admin proxy port (9091 by default) access from Chat server and PostgreSQL server for Chat and PostgreSQL installation.
- PostgreSQL server allows 5432 port access from Chat server **only**.
- Chat server allows 443 port access from Zimbra proxy and mailstore servers **only**.
- 🔒 **Security Requirements:**
  - **IMPORTANT:** Restrict access to the Chat web application by blocking unauthorized machines. Chat web application must always be accessed through Zimbra web client. It MUST NOT be accessible from any machine except Zimbra proxy and mailstore servers.
    - if you want to make Chat web application accessible by administrator for management purpose, you must configure network access to allow the access from a specified machine only using iptables, firewalld, network firewall or any network access control module.
  - Restrict access to the PostgreSQL database by blocking unauthorized machines.

Install Zimbra Chat modules

- Install ZCS 10.1.7 or later
  - Please proceed with the patch installation using wiki Zimbra Releases/10.1.0/patch installation - Zimbra :: Tech Center
- Install chat modules on all mailstore servers:

```
Ubuntu:
apt install zimbra-zimlet-chat

RHEL:
yum install zimbra-zimlet-chat

Common:
su - zimbra
zmmailboxdctl restart
```

Install Zimbra Chat server

SSL Certificate Requirements

A valid SSL certificate is required for the chat server to ensure secure HTTPS communication, allowing the Zimbra mailstore to access it correctly.

The installer supports three options: `Commercial`, `Let's Encrypt`, and `Self-Signed`. You must select one of these options when prompted during installation.

Commercial Certificate:

- Requires a valid SSL certificate to be pre-installed before running the Chat installer.
- DNS registration is required for the Chat server.
- Certificates must be deployed manually in the following locations:

  Private Key: `/etc/ssl/private/zulip.key`

  Certificate Chain: `/etc/ssl/certs/zulip.combined-chain.crt`
- The certificate must be able to verify the chat server's hostname and all Fully Qualified Domain Names (FQDNs), e.g., `chat1.mydomain.com`, `domain1examplecom.chat1.mydomain.com`, etc.

Let's Encrypt:

- The Chat installer uses the `--certbot` option to automatically obtain and configure an SSL certificate.
- By selecting this option, you must agree to the Let's Encrypt Subscriber Agreement when prompted by the installer. Please read the latest Let's Encrypt Subscriber Agreement at Policy and Legal Repository .
- DNS registration is mandatory.

Self-Signed Certificate:

- The Chat installer uses the `--self-signed-cert` option to generate a self-signed certificate.
- Not recommended for production use.

Prepare data for installation

The following details are required for installation.

- Zimbra Proxy Host: Hostname (FQDN) of Zimbra admin console proxy URL to send admin SOAP requests.
- Zimbra Proxy Admin Port: The port for the Zimbra admin console proxy on the proxy server (typically `9071`).
- Zulip Admin Email & Password: The email address and password of an existing administrator account in Zimbra. The email address will be used as an administrator on Zulip application as well.
- Zimbra JWT Authentication Key: The value of `zimbraChatJwtSecret` configured in Zimbra.
- Zimbra Web Client URLs: A comma-separated list of all accessible Zimbra web client URLs (e.g., `https://web.mydomain.com, https://virtual_domain1.example.com`). Do not add a trailing slash like `https://web.mydomain.com/`.

The installer will prompt you to choose an authentication source: Zimbra LDAP, External LDAP, or Active Directory (AD).

If LDAP Source is Zimbra LDAP:

- Zimbra LDAP Host: Hostname of the Zimbra LDAP server. If you have multiple LDAP servers, choose one of them to use.
- Zimbra LDAP Port: Port used by the Zimbra LDAP server.
- Zimbra LDAP Password: Password for connecting to Zimbra LDAP.
- Zimbra LDAP Protocol: Choose between `ldap` or `ldaps`.

If LDAP Source is External LDAP/AD:

- External LDAP/AD Host: Hostname of the external LDAP or Active Directory server. If you have multiple LDAP/AD servers, choose one of them to use.

- External LDAP/AD Port: Port used for LDAP/AD communication.
- External LDAP/AD Password: External LDAP or Active Directory server password for authentication.
- External LDAP/AD Protocol: Choose between `ldap` or `ldaps`.
- External LDAP/AD Auth Bind DN: Distinguished Name (DN) for authentication.
- External LDAP/AD Email Attribute: The LDAP/AD attribute storing the user's email address.
- External LDAP/AD Full Name Attribute: The LDAP/AD attribute storing the user's full name.

Deployment Modes

Single-node Setup

Install Chat server application (Zulip) and PostgreSQL on a single server .

Multi-node Setup

Install Chat server application (Zulip) and PostgreSQL on different nodes. There are two options for the order of installation.

Option 1: Install Zulip First, Then PostgreSQL

1. Install Standalone Zulip
   - When prompted:
     `Have you installed PostgreSQL on another VM, and is it ready for use? (yes/no)` → select `no`
2. Install Standalone PostgreSQL
3. Reconfigure Standalone Zulip to connect with the Standalone PostgreSQL instance
   - When prompted again:
     `Have you installed PostgreSQL on another VM, and is it ready for use?` → select `yes`

Option 2: Install PostgreSQL First, Then Zulip

1. Install Standalone PostgreSQL
2. Install Standalone Zulip
   - When prompted:
     `Have you installed PostgreSQL on another VM, and is it ready for use?` → select `yes`
   - Provide the details of the Standalone PostgreSQL server

Inputs required by installer

- **Zulip Service Type**: Choose based on your deployment mode. Options: *Standalone PostgreSQL*, *Standalone Zulip*, *Combined*.
- **Zimbra Proxy Host**: Hostname (FQDN) of Zimbra admin console proxy URL.
- **Zimbra Proxy Admin Port**: Admin console proxy port on the Zimbra proxy server (typically *9071*).
- **Zulip Admin Email & Password**: Credentials of an existing administrator account in Zimbra.
- **Zulip/Chat Server Hostname**: Fully Qualified Domain Name (FQDN) of the Zulip/Chat server.
- **Zulip Version**: Currently supported version is *9.2*.
- **Chat Server SSL Certificate Type**: Choose based on your requirement. Options: *commercial*, *letsencrypt*, *self-signed*.
- **LDAP/AD Source**: Select as per your environment. Options: *Zimbra LDAP*, *External LDAP*, *AD*.
- **Zimbra JWT Authentication Key**: The value of `zimbraChatJwtSecret` configured in Zimbra.
- **Zimbra Web Client URLs**: One or more Zimbra client URLs (comma-separated), e.g., `https://web.mydomain.com, https://virtual_domain1.example.com`.
- **Zulip Server IP**: Private IP address of the Zulip server, used by PostgreSQL to recognize the connection source.
- **PostgreSQL Host**: Hostname or IP address of the PostgreSQL server.
- **PostgreSQL Port**: Port number of the PostgreSQL service (typically *5432*).
- **PostgreSQL Password**: Password to be set for the zulip user in the PostgreSQL database.

- **PostgreSQL Version**: The version number of the PostgreSQL server. Retrieve the PostgreSQL version from the `/etc/zulip/zulip.conf` file from PostgreSQL Server.
- **PostgreSQL SSL Mode**: SSL mode for connecting to PostgreSQL.
  - For more details refer: 32.19. SSL Support

Install Zulip and PostgreSQL on a single server

1. Download the Chat Installer `.tgz` from the link below. Kindly verify the SHA256 and MD5 checksums as well.

```
https://files.zimbra.com/downloads/zfzi/1.0.0/zfzi-1.0.0.tgz
https://files.zimbra.com/downloads/zfzi/1.0.0/zfzi-1.0.0.tgz.md5
https://files.zimbra.com/downloads/zfzi/1.0.0/zfzi-1.0.0.tgz.sha256
```

2. Switch to the `root` user.
3. Extract the archive:

```
tar xf zfzi-1.0.0.tgz
cd zfzi-1.0.0
./install.pl
```

4. The installer will prompt you to enter the `Zulip Service Type` (`Standalone PostgreSQL`, `Standalone Zulip`, `Combined`), choose `Combined`.

```
[2025-04-04 04:14:53] [INFO] Logging initialized. Operations will be logged to /etc/zulip-installer/log/zulip.install.log.20250404_041453
[2025-04-04 04:14:53] [INFO] Detected OS: ubuntu:24.04, Architecture: x86_64. Proceeding.
[2025-04-04 04:14:53] [INFO] All required commands are already installed.
Enter Zulip Service Type (Available: Standalone PostgreSQL, Standalone Zulip, Combined): Combined
```

5. The installer will prompt you to provide the `Zimbra Proxy Host`, `Zimbra Proxy Admin Port`, `Zulip Admin Email`, and `Zulip Admin Email Account Password`. Please enter your inputs accordingly.

```
Enter Zimbra Proxy Host: zimbra.com
Enter Zimbra Proxy Admin Port: 9071
Enter Zulip Admin Email: admin@domain1.example.com
Enter Zulip Admin Email Account Password: PASSWORD
```

6. The installer will then verify connectivity to the Zimbra Proxy Host and validate the license using the information you provided.

```
[2025-04-04 04:15:13] [INFO] Checking connectivity to zimbra.com:9071
[2025-04-04 04:15:13] [INFO] Connected to zimbra.com:9071.
[2025-04-04 04:15:13] [INFO] Fetching Zimbra admin authentication token...
[2025-04-04 04:15:13] [INFO] Admin Auth Token fetched.
[2025-04-04 04:15:13] [INFO] Fetching Zimbra license information.
[2025-04-04 04:15:13] [INFO] License information retrieved.
[2025-04-04 04:15:13] [INFO] Chat services are supported by the existing Zimbra license.
```

7. The installer will prompt you to enter the `Chat Server Hostname`.

```
Enter Chat Server Hostname: chat1.mydomain.com
[2025-04-04 04:15:20] [INFO] Checking resolution for hostname: chat1.mydomain.com
[2025-04-04 04:15:20] [WARN] Hostname 'chat1.mydomain.com' is not resolvable via DNS. Checking /etc/hosts.
[2025-04-04 04:15:20] [INFO] Hostname 'chat1.mydomain.com' found in /etc/hosts. Proceeding.
```

8. The installer will prompt you to enter the `Zulip Version`.

```
Enter Zulip Version (Available: 9.2): 9.2
```

9. The installer will prompt you to enter the `Chat Server SSL Certificate Type`, `LDAP/AD Source`, `Zimbra JWT Auth Key`, `URLs of Zimbra web client`. Please enter your inputs accordingly.

```
Enter Chat Server SSL Certificate Type (commercial, letsencrypt, self-signed): letsencrypt
Enter LDAP/AD Source (Zimbra LDAP, External LDAP, AD): Zimbra LDAP
Enter Zimbra JWT Auth Key: SECRET_KEY
Enter URLs of Zimbra web client (comma-separated if multiple): https://zimbra.com
```

10. If you selected `letsencrypt` as the `Chat Server SSL Certificate Type`, the installer will prompt you to accept the Let's Encrypt Subscriber Agreement. You must agree to proceed with the installation.

```
Please read the latest Let's Encrypt Subscriber Agreement at:
https://letsencrypt.org/repository/#let-s-encrypt-subscriber-agreement
You must agree in order to register with the ACME server.
You must agree to the Let's Encrypt Subscriber Agreement to proceed.
Do you agree? (yes/no): yes
```

11. If you selected `letsencrypt` or `commercial` as the Chat Server SSL Certificate Type, the installer will check whether the Chat Server Hostname is publicly resolvable.

```
[2025-04-04 04:14:32] [INFO] Let’s Encrypt certificate selected. DNS registration is mandatory for Chat Server Hostname.
[2025-04-04 04:14:33] [INFO] Chat Server Hostname is publicly resolvable.
```

12. The installer will prompt for LDAP details based on the `LDAP/AD` Source you specified. Please provide the requested information accordingly. If you have multiple LDAP servers, choose one of them to use.

```
Enter Zimbra LDAP Host: zimbra.com
Enter Zimbra LDAP Port: 389
Enter Zimbra LDAP Password: LDAP_PASSWORD
Enter Zimbra LDAP Protocol (ldap/ldaps): ldap
```

13. The installer will check connectivity to the LDAP server and prompt you to confirm whether to proceed. You must enter `yes` to continue with the installation.

```
[2025-04-04 04:15:50] [INFO] Checking connectivity to ldap server at zimbra.com:389
[2025-04-04 04:15:50] [INFO] Connected to ldap server at zimbra.com:389
[2025-04-04 04:15:50] [INFO] Package already exists: /etc/zulip-installer/package/zulip-server-9.2.tar.gz
[2025-04-04 04:15:50] [INFO] The following configuration will be used:
[2025-04-04 04:15:50] [INFO] Zulip version: 9.2
[2025-04-04 04:15:50] [INFO] Zulip service type: Combined
[2025-04-04 04:15:50] [INFO] Zulip/Chat server hostname: chat1.mydomain.com
[2025-04-04 04:15:50] [INFO] Chat server SSL certificate type: letsencrypt
[2025-04-04 04:15:50] [INFO] Zulip admin email: admin@domain1.example.com
[2025-04-04 04:15:50] [INFO] LDAP/AD Source: Zimbra LDAP
[2025-04-04 04:15:50] [INFO] URL's of Zimbra web client: https://web.mydomain.com

The system will be modified - continue? (yes/no): yes
```

14. The installation process will begin and may take some time to complete. Please wait while the setup finishes.
15. The installation output will look like the following (it may vary depending on the inputs provided):

```
[2025-04-04 04:14:53] [INFO] Logging initialized. Operations will be logged to /etc/zulip-installer/log/zulip.install.log.20250404_041453
[2025-04-04 04:14:53] [INFO] Detected OS: ubuntu:24.04, Architecture: x86_64. Proceeding.
[2025-04-04 04:14:53] [INFO] All required commands are already installed.
Enter Zulip Service Type (Available: Standalone PostgreSQL, Standalone Zulip, Combined): Combined
Enter Zimbra Proxy Host: zimbra.com
Enter Zimbra Proxy Admin Port: 9071
```

```
Enter Zulip Admin Email: admin@domain1.example.com
Enter Zulip Admin Email Account Password: PASSWORD
[2025-04-04 04:15:13] [INFO] Checking connectivity to zimbra.com:9071
[2025-04-04 04:15:13] [INFO] Connected to zimbra.com:9071.
[2025-04-04 04:15:13] [INFO] Fetching Zimbra admin authentication token...
[2025-04-04 04:15:13] [INFO] Admin Auth Token fetched.
[2025-04-04 04:15:13] [INFO] Fetching Zimbra license information.
[2025-04-04 04:15:13] [INFO] License information retrieved.
[2025-04-04 04:15:13] [INFO] Chat services are supported by the existing Zimbra license.
Enter Chat Server Hostname: chat1.mydomain.com
[2025-04-04 04:15:20] [INFO] Checking resolution for hostname: chat1.mydomain.com
[2025-04-04 04:15:20] [WARN] Hostname 'chat1.mydomain.com' is not resolvable via DNS. Checking /etc/hosts.
[2025-04-04 04:15:20] [INFO] Hostname 'chat1.mydomain.com' found in /etc/hosts. Proceeding.
Enter Zulip Version (Available: 9.2): 9.2
Enter Chat Server SSL Certificate Type (commercial, letsencrypt, self-signed): letsencrypt
Enter LDAP/AD Source (Zimbra LDAP, External LDAP, AD): Zimbra LDAP
Enter Zimbra JWT Auth Key: SECRET_KEY
Enter URLs of Zimbra web client (comma-separated if multiple): https://web.mydomain.com
Please read the latest Let's Encrypt Subscriber Agreement at:
https://letsencrypt.org/repository/#let-s-encrypt-subscriber-agreement
You must agree in order to register with the ACME server.
You must agree to the Let's Encrypt Subscriber Agreement to proceed.
Do you agree? (yes/no): yes
[2025-04-04 04:14:32] [INFO] Let’s Encrypt certificate selected. DNS registration is mandatory for Chat Server Hostname.
[2025-04-04 04:14:33] [INFO] Chat Server Hostname is publicly resolvable.
Enter Zimbra LDAP Host: zimbra.com
Enter Zimbra LDAP Port: 389
Enter Zimbra LDAP Password: LDAP_PASSWORD
Enter Zimbra LDAP Protocol (ldap/ldaps): ldap
[2025-04-04 04:15:50] [INFO] Checking connectivity to ldap server at zimbra.com:389
[2025-04-04 04:15:50] [INFO] Connected to ldap server at zimbra.com:389
[2025-04-04 04:15:50] [INFO] Package already exists: /etc/zulip-installer/package/zulip-server-9.2.tar.gz
[2025-04-04 04:15:50] [INFO] The following configuration will be used:
[2025-04-04 04:15:50] [INFO] Zulip version: 9.2
[2025-04-04 04:15:50] [INFO] Zulip service type: Combined
[2025-04-04 04:15:50] [INFO] Zulip/Chat server hostname: chat1.mydomain.com
[2025-04-04 04:15:50] [INFO] Chat server SSL certificate type: letsencrypt
[2025-04-04 04:15:50] [INFO] Zulip admin email: admin@domain1.example.com
[2025-04-04 04:15:50] [INFO] LDAP/AD Source: Zimbra LDAP
[2025-04-04 04:15:50] [INFO] URL's of Zimbra web client: https://web.mydomain.com
The system will be modified - continue? (yes/no): yes
[2025-04-03 06:47:51] [INFO] Saving user inputs to config file: /etc/zulip-installer/config/config.20250403_064751
[2025-04-03 06:47:51] [INFO] User inputs saved to config file.
[2025-04-03 06:47:51] [INFO] Starting: Adding universe repository. Please wait.
[2025-04-03 06:48:00] [INFO] Completed: Adding universe repository
[2025-04-03 06:48:00] [INFO] Starting: Updating package lists. Please wait.
[2025-04-03 06:48:02] [INFO] Completed: Updating package lists
[2025-04-03 06:48:02] [INFO] Starting: Extracting Zulip package. Please wait.
[2025-04-03 06:48:03] [INFO] Completed: Extracting Zulip package
[2025-04-03 06:48:03] [INFO] Starting: Zulip and PostgreSQL installation. Please wait.
[2025-04-03 06:48:52] [INFO] Zulip and PostgreSQL installation is still in progress
[2025-04-03 06:49:22] [INFO] Zulip and PostgreSQL installation is still in progress
[2025-04-03 07:30:28] [INFO] Completed: Zulip and PostgreSQL installation
[2025-04-03 07:30:28] [INFO] Package downloaded: /etc/zulip-installer/package/chat-customizations-9.2.tar.gz
[2025-04-03 07:30:28] [INFO] Applying Zulip customizations...
[2025-04-03 07:30:29] [INFO] Zulip customizations applied successfully.
```

```
[2025-04-03 07:30:29] [INFO] Updated /etc/zulip/settings.py.
[2025-04-03 07:30:29] [INFO] Updated /etc/zulip/zulip-secrets.conf.
[2025-04-03 07:30:29] [INFO] Restarting nginx.service...
[2025-04-03 07:30:29] [INFO] Starting: Restarting nginx.service. Please wait.
[2025-04-03 07:30:29] [INFO] Completed: Restarting nginx.service
[2025-04-03 07:30:29] [INFO] Restarting zulip...
[2025-04-03 07:30:29] [INFO] Starting: Restarting zulip. Please wait.
[2025-04-03 07:31:06] [INFO] Completed: Restarting zulip
! IMPORTANT: Admins should ensure that databases are on a private network and not publicly accessible!
! IMPORTANT: Admins should ensure that zimbra mailstore servers can resolve the Chat server hostname and access Chat server on 443 port!
! IMPORTANT: Please visit admin guide to proceed to realm setup
Configuration complete
```

Install Standalone Zulip

1. Download the Chat Installer `.tgz` from the link below. Kindly verify the SHA256 and MD5 checksums as well.

```
https://files.zimbra.com/downloads/zfzi/1.0.0/zfzi-1.0.0.tgz
https://files.zimbra.com/downloads/zfzi/1.0.0/zfzi-1.0.0.tgz.md5
https://files.zimbra.com/downloads/zfzi/1.0.0/zfzi-1.0.0.tgz.sha256
```

2. Switch to the `root` user.
3. Extract the archive:

```
tar xf zfzi-1.0.0.tgz
cd zfzi-1.0.0
./install.pl
```

4. The installer will prompt you to enter the `Zulip Service Type` (`Standalone PostgreSQL`, `Standalone Zulip`, `Combined`), choose `Standalone Zulip`.

```
[2025-04-02 09:20:48] [INFO] Logging initialized. Operations will be logged to /etc/zulip-installer/log/zulip.install.log.20250402_092048
[2025-04-02 09:20:48] [INFO] Detected OS: ubuntu:24.04, Architecture: aarch64. Proceeding.
[2025-04-02 09:20:48] [INFO] All required commands are already installed.
Enter Zulip Service Type (Available: Standalone PostgreSQL, Standalone Zulip, Combined): Standalone Zulip
```

5. The installer will prompt you to provide the `Zimbra Proxy Host`, `Zimbra Proxy Admin Port`, `Zulip Admin Email`, and `Zulip Admin Email Account Password`. Please enter your inputs accordingly.

```
Enter Zimbra Proxy Host: zimbra.com
Enter Zimbra Proxy Admin Port: 9071
Enter Zulip Admin Email: admin@domain1.example.com
Enter Zulip Admin Email Account Password: PASSWORD
```

6. The installer will then verify connectivity to the `Zimbra Proxy Host` and validate the license using the information you provided.

```
[2025-04-02 09:21:09] [INFO] Checking connectivity to zimbra.com:9071
[2025-04-02 09:21:09] [INFO] Connected to zimbra.com:9071.
[2025-04-02 09:21:09] [INFO] Fetching Zimbra admin authentication token...
[2025-04-02 09:21:09] [INFO] Admin Auth Token fetched.
[2025-04-02 09:21:09] [INFO] Fetching Zimbra license information.
[2025-04-02 09:21:09] [INFO] License information retrieved.
[2025-04-02 09:21:09] [INFO] Chat services are supported by the existing Zimbra license.
```

7. The installer will prompt you to enter the `Chat Server Hostname`.

```
Enter Chat Server Hostname: chat1.mydomain.com
```

```
[2025-04-04 04:15:20] [INFO] Checking resolution for hostname: chat1.mydomain.com
[2025-04-04 04:15:20] [WARN] Hostname 'chat1.mydomain.com' is not resolvable via DNS. Checking /etc/hosts.
[2025-04-04 04:15:20] [INFO] Hostname 'chat1.mydomain.com' found in /etc/hosts. Proceeding.
```

8. The installer will prompt you to enter the `Zulip Version`.

```
Enter Zulip Version (Available: 9.2): 9.2
```

9. The installer will prompt you to enter the `Chat Server SSL Certificate Type`, `LDAP/AD Source`, `Zimbra JWT Auth Key`, `URLs of Zimbra web client`. Please enter your inputs accordingly.

```
Enter Chat Server SSL Certificate Type (commercial, letsencrypt, self-signed): letsencrypt
Enter LDAP/AD Source (Zimbra LDAP, External LDAP, AD): Zimbra LDAP
Enter Zimbra JWT Auth Key: SECRET_KEY
Enter URLs of Zimbra web client (comma-separated if multiple): https://web.mydomain.com
```

10. If you selected `letsencrypt` as the `Chat Server SSL Certificate Type`, the installer will prompt you to accept the Let's Encrypt Subscriber Agreement. You must agree to proceed with the installation.

```
Please read the latest Let's Encrypt Subscriber Agreement at:
https://letsencrypt.org/repository/#let-s-encrypt-subscriber-agreement
You must agree in order to register with the ACME server.
You must agree to the Let's Encrypt Subscriber Agreement to proceed.
Do you agree? (yes/no): yes
```

11. If you selected `letsencrypt` or `commercia` as the `Chat Server SSL Certificate Type`, the installer will check whether the Chat Server Hostname is publicly resolvable.

```
[2025-04-02 09:21:20] [INFO] Let’s Encrypt certificate selected. DNS registration is mandatory for Chat Server Hostname.
[2025-04-02 09:21:21] [INFO] Chat Server Hostname is publicly resolvable.
```

12. The installer will prompt for LDAP details based on the `LDAP/AD Source` you specified. Please provide the requested information accordingly. If you have multiple LDAP servers, choose one of them to use.

```
Enter Zimbra LDAP Host: zimbra.com
Enter Zimbra LDAP Port: 389
Enter Zimbra LDAP Password: LDAP_PASSWORD
Enter Zimbra LDAP Protocol (ldap/ldaps): ldap
```

13. The installer will check connectivity to the LDAP server.

```
[2025-04-02 09:21:51] [INFO] Checking connectivity to ldap server at zimbra.com:389
[2025-04-02 09:21:51] [INFO] Connected to ldap server at zimbra.com:389
[2025-04-02 09:21:51] [INFO] Creating package directory: /etc/zulip-installer/package
[2025-04-02 09:21:52] [INFO] Package downloaded: /etc/zulip-installer/package/zulip-server-9.2.tar.gz
```

14. The installer will ask, `Have you installed PostgreSQL on another VM, and is it ready for use? (yes/no)`. You should answer `no`, as we're first installing Standalone Zulip and the standalone PostgreSQL setup is not yet complete.

```
Have you installed PostgreSQL on another VM, and is it ready for use? (yes/no): no
! IMPORTANT: Once the current installation is complete, install and configure Remote PostgreSQL. Then, rerun the script with option `--use-remote-db` to finish setup.
```

15. The installer will prompt with "The system will be modified - continue? (yes/no):". You need to enter `yes` to proceed with the installation.

```
[2025-04-02 09:21:54] [INFO] The following configuration will be used:
[2025-04-02 09:21:54] [INFO] Zulip version: 9.2
[2025-04-02 09:21:54] [INFO] Zulip service type: Standalone Zulip
```

```
[2025-04-02 09:21:54] [INFO] Zulip/Chat server hostname: chat1.mydomain.com
[2025-04-02 09:21:54] [INFO] Chat server SSL certificate type: letsencrypt
[2025-04-02 09:21:54] [INFO] Zulip admin email: admin@domain1.example.com
[2025-04-02 09:21:54] [INFO] LDAP/AD Source: Zimbra LDAP
[2025-04-02 09:21:54] [INFO] URL's of Zimbra web client: https://web.mydomain.com

The system will be modified - continue? (yes/no): yes
```

16. The installation process will begin and may take some time to complete. Please wait while the setup finishes.

17. The installation output will look like the following (it may vary depending on the inputs provided):

```
[2025-04-02 09:20:48] [INFO] Logging initialized. Operations will be logged to /etc/zulip-installer/log/zulip.install.log.20250402_092048
[2025-04-02 09:20:48] [INFO] Detected OS: ubuntu:24.04, Architecture: aarch64. Proceeding.
[2025-04-02 09:20:48] [INFO] All required commands are already installed.
Enter Zulip Service Type (Available: Standalone PostgreSQL, Standalone Zulip, Combined): Standalone Zulip
Enter Zimbra Proxy Host: zimbra.com
Enter Zimbra Proxy Admin Port: 9071
Enter Zulip Admin Email: admin@domain1.example.com
Enter Zulip Admin Email Account Password:
[2025-04-02 09:21:09] [INFO] Checking connectivity to zimbra.com:9071
[2025-04-02 09:21:09] [INFO] Connected to zimbra.com:9071.
[2025-04-02 09:21:09] [INFO] Fetching Zimbra admin authentication token...
[2025-04-02 09:21:09] [INFO] Admin Auth Token fetched.
[2025-04-02 09:21:09] [INFO] Fetching Zimbra license information.
[2025-04-02 09:21:09] [INFO] License information retrieved.
[2025-04-02 09:21:09] [INFO] Chat services are supported by the existing Zimbra license.
Enter Chat Server Hostname: chat1.mydomain.com
[2025-04-02 09:21:19] [INFO] Checking resolution for hostname: chat1.mydomain.com
[2025-04-02 09:21:19] [WARN] Hostname 'chat1.mydomain.com' is not resolvable via DNS. Checking /etc/hosts.
[2025-04-02 09:21:19] [INFO] Hostname 'chat1.mydomain.com' found in /etc/hosts. Proceeding.
Enter Zulip Version (Available: 9.2): 9.2
Enter Chat Server SSL Certificate Type (commercial, letsencrypt, self-signed): letsencrypt
Enter LDAP/AD Source (Zimbra LDAP, External LDAP, AD): Zimbra LDAP
Enter Zimbra JWT Auth Key: SECRET_KEY
Enter URLs of Zimbra web client (comma-separated if multiple): https://web.mydomain.com
Please read the latest Let's Encrypt Subscriber Agreement at:
https://letsencrypt.org/repository/#let-s-encrypt-subscriber-agreement
You must agree in order to register with the ACME server.
You must agree to the Let's Encrypt Subscriber Agreement to proceed.
Do you agree? (yes/no): yes
[2025-04-02 09:21:20] [INFO] Let’s Encrypt certificate selected. DNS registration is mandatory for Chat Server Hostname.
[2025-04-02 09:21:21] [INFO] Chat Server Hostname is publicly resolvable.
Enter Zimbra LDAP Host: zimbra.com
Enter Zimbra LDAP Port: 389
Enter Zimbra LDAP Password:
Enter Zimbra LDAP Protocol (ldap/ldaps): ldap
[2025-04-02 09:21:51] [INFO] Checking connectivity to ldap server at zimbra.com:389
[2025-04-02 09:21:51] [INFO] Connected to ldap server at zimbra.com:389
[2025-04-02 09:21:51] [INFO] Creating package directory: /etc/zulip-installer/package
[2025-04-02 09:21:52] [INFO] Package downloaded: /etc/zulip-installer/package/zulip-server-9.2.tar.gz
Have you installed PostgreSQL on another VM, and is it ready for use? (yes/no): no
! IMPORTANT: Once the current installation is complete, install and configure Remote PostgreSQL. Then, rerun the script with option `--use-remote-db` to finish setup.
[2025-04-02 09:21:54] [INFO] The following configuration will be used:
[2025-04-02 09:21:54] [INFO] Zulip version: 9.2
[2025-04-02 09:21:54] [INFO] Zulip service type: Standalone Zulip
```

```
[2025-04-02 09:21:54] [INFO] Zulip/Chat server hostname: chat1.mydomain.com
[2025-04-02 09:21:54] [INFO] Chat server SSL certificate type: letsencrypt
[2025-04-02 09:21:54] [INFO] Zulip admin email: admin@domain1.example.com
[2025-04-02 09:21:54] [INFO] LDAP/AD Source: Zimbra LDAP
[2025-04-02 09:21:54] [INFO] URL's of Zimbra web client: https://web.mydomain.com
The system will be modified - continue? (yes/no): yes
[2025-04-02 09:21:56] [INFO] Saving user inputs to config file: /etc/zulip-installer/config/config.20250402_092156
[2025-04-02 09:21:56] [INFO] User inputs saved to config file.
[2025-04-02 09:21:56] [INFO] Starting: Adding universe repository. Please wait.
[2025-04-02 09:22:01] [INFO] Completed: Adding universe repository
[2025-04-02 09:22:01] [INFO] Starting: Updating package lists. Please wait.
[2025-04-02 09:22:03] [INFO] Completed: Updating package lists
[2025-04-02 09:22:03] [INFO] Starting: Extracting Zulip package. Please wait.
[2025-04-02 09:22:05] [INFO] Completed: Extracting Zulip package
[2025-04-02 09:22:05] [INFO] Starting: Standalone Zulip installation. Please wait.
[2025-04-02 09:28:27] [INFO] Standalone Zulip installation is still in progress
[2025-04-02 09:28:57] [INFO] Standalone Zulip installation is still in progress
[2025-04-02 09:29:27] [INFO] Standalone Zulip installation is still in progress
[2025-04-02 09:42:41] [INFO] Completed: Standalone Zulip installation
[2025-04-02 09:42:41] [INFO] Package downloaded: /etc/zulip-installer/package/chat-customizations-9.2.tar.gz
[2025-04-02 09:42:41] [INFO] Applying Zulip customizations...
[2025-04-02 09:42:42] [INFO] Zulip customizations applied successfully.
[2025-04-02 09:42:42] [INFO] Updated /etc/zulip/settings.py.
[2025-04-02 09:42:42] [INFO] Updated /etc/zulip/zulip-secrets.conf.
[2025-04-02 09:42:42] [INFO] Restarting nginx.service...
[2025-04-02 09:42:42] [INFO] Starting: Restarting nginx.service. Please wait.
[2025-04-02 09:42:43] [INFO] Completed: Restarting nginx.service
! IMPORTANT: Admins should ensure that zimbra mailstore servers can resolve the Chat server hostname and access Chat server on 443 port!
! IMPORTANT: Please visit admin guide to proceed to realm setup
```

Install Standalone PostgreSQL

1. Download the Chat Installer `.tgz` from the link below. Kindly verify the SHA256 and MD5 checksums as well.

```
https://files.zimbra.com/downloads/zfzi/1.0.0/zfzi-1.0.0.tgz
https://files.zimbra.com/downloads/zfzi/1.0.0/zfzi-1.0.0.tgz.md5
https://files.zimbra.com/downloads/zfzi/1.0.0/zfzi-1.0.0.tgz.sha256
```

2. Switch to the `root` user.
3. Extract the archive:

```
tar xf zfzi-1.0.0.tgz
cd zfzi-1.0.0
./install.pl
```

4. The installer will prompt you to enter the `Zulip Service Type` (`Standalone PostgreSQL`, `Standalone Zulip`, `Combined`), choose `Standalone PostgreSQL`.

```
[2025-03-23 17:04:57] [INFO] Logging initialized. Operations will be logged to /etc/zulip-installer/log/zulip.install.log.20250323_170457
[2025-03-23 17:04:57] [INFO] Detected OS: ubuntu:22.04, Architecture: aarch64. Proceeding...
[2025-03-23 17:04:57] [INFO] All required commands are already installed.
Enter Zulip Service Type (Available: Standalone PostgreSQL, Standalone Zulip, Combined) : Standalone PostgreSQL
```

5. The installer will prompt you to provide the `Zimbra Proxy Host`, `Zimbra Proxy Admin Port`, `Zulip Admin Email`, and `Zulip Admin Email Account Password`. Please enter your inputs accordingly.

```
Enter Zimbra Proxy Host: zimbra.com
Enter Zimbra Proxy Admin Port: 9071
Enter Zulip Admin Email: admin@domain1.example.com
Enter Zulip Admin Email Account Password: PASSWORD
```

6. The installer will then verify connectivity to the `Zimbra Proxy Host` and validate the license using the information you provided.

```
[2025-03-23 17:05:01] [INFO] Checking connectivity to zimbra.com:9071...
[2025-03-23 17:05:01] [INFO] Successfully connected to zimbra.com:9071.
[2025-03-23 17:05:01] [INFO] Fetching Zimbra admin authentication token...
[2025-03-23 17:05:01] [INFO] Admin Auth Token fetched successfully.
[2025-03-23 17:05:01] [INFO] Fetching Zimbra license information...
[2025-03-23 17:05:01] [INFO] License information retrieved successfully.
[2025-03-23 17:05:01] [INFO] Chat services are supported by the existing Zimbra license.
```

7. The installer will prompt you to enter the `Zulip Version`.

```
Enter Zulip Version (Available: 9.2) : 9.2
```

8. The installer will prompt you to enter the Zulip server IP that will be visible when connected to the PostgreSQL instance, as well as the PostgreSQL server password to set for the zulip user. Please provide these details accordingly.

```
Enter Zulip server IP visible when connected to this PostgreSQL instance : IP_ADDRESS
Enter PostgreSQL server password to set for the zulip user: ZULIP_USER_PASSWORD
```

9. The installer will display:
   `The system will be modified - continue? (yes/no):`
   Enter `yes` to proceed.

```
[2025-03-23 17:05:10] [INFO] Package downloaded : /etc/zulip-installer/package/zulip-server-9.2.tar.gz
[2025-03-23 17:05:11] [INFO] The following configuration will be used:
[2025-03-23 17:05:11] [INFO] Zulip server ip: IP_ADDRESS
[2025-03-23 17:05:11] [INFO] Zulip version: 9.2
[2025-03-23 17:05:11] [INFO] Zulip service type: Standalone PostgreSQL

The system will be modified - continue? (yes/no): yes
```

10. The installation process will begin and may take some time to complete. Please wait while the setup finishes.
11. The installation output will look like the following (it may vary depending on the inputs provided):

```
[2025-03-23 17:04:57] [INFO] Logging initialized. Operations will be logged to /etc/zulip-installer/log/zulip.install.log.20250323_170457
[2025-03-23 17:04:57] [INFO] Detected OS: ubuntu:22.04, Architecture: aarch64. Proceeding...
[2025-03-23 17:04:57] [INFO] All required commands are already installed.
Enter Zulip Service Type (Available: Standalone PostgreSQL, Standalone Zulip, Combined) : Standalone PostgreSQL
Enter Zimbra Proxy Host : zimbra.com
Enter Zimbra Proxy Admin Port : 9071
Enter Zulip Admin Email: admin@domain1.example.com
Enter Zimbra Admin Account Password: PASSWORD
[2025-03-23 17:05:01] [INFO] Checking connectivity to zimbra.com:9071...
[2025-03-23 17:05:01] [INFO] Successfully connected to zimbra.com:9071.
[2025-03-23 17:05:01] [INFO] Fetching Zimbra admin authentication token...
[2025-03-23 17:05:01] [INFO] Admin Auth Token fetched successfully.
[2025-03-23 17:05:01] [INFO] Fetching Zimbra license information...
[2025-03-23 17:05:01] [INFO] License information retrieved successfully.
[2025-03-23 17:05:01] [INFO] Chat services are supported by the existing Zimbra license.
Enter Zulip Version (Available: 9.2) : 9.2
Enter Zulip server IP visible when connected to this PostgreSQL instance : IP_ADDRESS
```

```
Enter PostgreSQL server password to set for the zulip user: ZULIP_USER_PASSWORD
[2025-03-23 17:05:10] [INFO] Package downloaded : /etc/zulip-installer/package/zulip-server-9.2.tar.gz
[2025-03-23 17:05:11] [INFO] The following configuration will be used:
[2025-03-23 17:05:11] [INFO] Zulip server ip: IP_ADDRESS
[2025-03-23 17:05:11] [INFO] Zulip version: 9.2
[2025-03-23 17:05:11] [INFO] Zulip service type: Standalone PostgreSQL
The system will be modified - continue? (yes/no): yes
[2025-03-23 17:05:14] [INFO] Saving user inputs to config file: /etc/zulip-installer/config/config.20250323_170514
[2025-03-23 17:05:14] [INFO] User inputs saved successfully to config file.
[2025-03-23 17:05:14] [INFO] Starting: Adding universe repository
[2025-03-23 17:05:21] [INFO] Completed: Adding universe repository
[2025-03-23 17:05:21] [INFO] Starting: Updating package lists
[2025-03-23 17:05:24] [INFO] Completed: Updating package lists
[2025-03-23 17:05:24] [INFO] Removing existing extracted zulip directory: /tmp/zulip-server-9.2
[2025-03-23 17:05:24] [INFO] Starting: Removing old extracted zulip
[2025-03-23 17:05:24] [INFO] Completed: Removing old extracted zulip
[2025-03-23 17:05:25] [INFO] Starting: Extracting Zulip package
[2025-03-23 17:05:26] [INFO] Completed: Extracting Zulip package
[2025-03-23 17:05:26] [INFO] Starting: Running Standalone PostgreSQL installation
[2025-03-23 17:05:35] [INFO] Completed: Running Standalone PostgreSQL installation
[2025-03-23 17:05:35] [INFO] Starting: Running PostgreSQL create database
[2025-03-23 17:05:35] [INFO] Completed: Running PostgreSQL create database
[2025-03-23 17:05:35] [INFO] PostgreSQL create database completed.
[2025-03-23 17:05:35] [INFO] Detected PostgreSQL version: 16
! IMPORTANT: Added a rule to allow the `zulip` user to connect to the `zulip` database from 10.0.0.224 using scram-sha-256 authentication.
! IMPORTANT: If changes are needed to the PostgreSQL authentication configuration, please modify the /etc/postgresql/16/main/pg_hba.conf file.
[2025-03-23 17:05:35] [INFO] Restarting postgresql.service...
[2025-03-23 17:05:38] [INFO] postgresql.service restarted successfully.
[2025-03-23 17:05:38] [INFO] Checking if PostgreSQL is accessible on host: localhost...
[2025-03-23 17:05:38] [INFO] PostgreSQL is accessible on localhost.
[2025-03-23 17:05:38] [INFO] Setting Zulip database user password...
[2025-03-23 17:05:38] [INFO] Zulip database password set successfully.
! IMPORTANT: Admins should ensure that databases are on a private network and not publicly accessible!
! IMPORTANT: Admins should ensure that Chat server can access the PostgreSQL server on port 5432!
Configuration complete
```

Configure Standalone Zulip Server with Standalone PostgreSQL

1. Return to Standalone Zulip Server
2. run `./install.pl --use-remote-db` to configure Standalone Zulip Server with Standalone PostgreSQL.
3. The installer will prompt with `Have you installed PostgreSQL on another VM, and is it ready for use? (yes/no):`. You should enter `yes`, as PostgreSQL has already been installed and is ready for use.

```
[2025-03-26 11:41:05] [INFO] Logging initialized. Operations will be logged to /etc/zulip-installer/log/zulip.configure-remote-db.log.20250326_114105
[2025-03-26 11:41:05] [INFO] Detected OS: ubuntu:24.04, Architecture: x86_64. Proceeding.
Have you installed PostgreSQL on another VM, and is it ready for use? (yes/no): yes
```

4. The installer will prompt you to enter the PostgreSQL server details. Please provide the information accordingly.

```
Enter Name or IP address of the PostgreSQL server: POSTGRESQL_SERVER
Enter PostgreSQL server port: 5432
Enter PostgreSQL server password for the zulip user: PASSWORD
Enter PostgreSQL server version: 16
```

```
Enter desired SSL mode for the PostgreSQL connection (disable, allow, prefer, require, verify-ca, verify-full):
require
```

5. The installer will check whether the PostgreSQL server is accessible and proceed accordingly based on the result.

```
[2025-03-26 11:41:35] [INFO] Checking if PostgreSQL is accessible on host: .
[2025-03-26 11:41:35] [INFO] PostgreSQL is accessible on .
[2025-03-26 11:41:35] [INFO] Configuring Zulip to use remote PostgreSQL.
[2025-03-26 11:41:35] [INFO] Updated /etc/zulip/settings.py with remote PostgreSQL details.
[2025-03-26 11:41:35] [INFO] Updated /etc/zulip/zulip-secrets.conf with PostgreSQL password.
[2025-03-26 11:41:35] [INFO] Updated /etc/zulip/zulip.conf with PostgreSQL version.
[2025-03-26 11:41:36] [INFO] Configured Zulip for remote PostgreSQL.
[2025-03-26 11:41:35] [INFO] Starting: Initializing the PostgreSQL database. Please wait.
[2025-03-26 11:41:37] [INFO] Completed: Initializing the PostgreSQL database
[2025-03-26 11:41:38] [INFO] Starting: Generating realm creation link. Please wait.
[2025-03-26 11:41:39] [INFO] Completed: Generating realm creation link
Configuration complete.
```

6. The configuration output will look like the following (it may vary depending on the inputs provided):

```
[2025-03-26 11:41:05] [INFO] Logging initialized. Operations will be logged to /etc/zulip-
installer/log/zulip.configure-remote-db.log.20250326_114105
[2025-03-26 11:41:05] [INFO] Detected OS: ubuntu:24.04, Architecture: x86_64. Proceeding.
Have you installed PostgreSQL on another VM, and is it ready for use? (yes/no): yes
Enter Name or IP address of the PostgreSQL server: POSTGRESQL_SERVER
Enter PostgreSQL server port: 5432
Enter PostgreSQL server password for the zulip user: PASSWORD
Enter PostgreSQL server version: 16
Enter desired SSL mode for the PostgreSQL connection (disable, allow, prefer, require, verify-ca, verify-
full): require
[2025-03-26 11:41:35] [INFO] Checking if PostgreSQL is accessible on host: .
[2025-03-26 11:41:35] [INFO] PostgreSQL is accessible on .
[2025-03-26 11:41:35] [INFO] Configuring Zulip to use remote PostgreSQL.
[2025-03-26 11:41:35] [INFO] Updated /etc/zulip/settings.py with remote PostgreSQL details.
[2025-03-26 11:41:35] [INFO] Updated /etc/zulip/zulip-secrets.conf with PostgreSQL password.
[2025-03-26 11:41:35] [INFO] Updated /etc/zulip/zulip.conf with PostgreSQL version.
[2025-03-26 11:41:36] [INFO] Configured Zulip for remote PostgreSQL.
[2025-03-26 11:41:35] [INFO] Starting: Initializing the PostgreSQL database. Please wait.
[2025-03-26 11:41:37] [INFO] Completed: Initializing the PostgreSQL database
[2025-03-26 11:41:38] [INFO] Starting: Generating realm creation link. Please wait.
[2025-03-26 11:41:39] [INFO] Completed: Generating realm creation link
Configuration complete.
```

Configuration & Logging

- User Inputs Storage: `/etc/zulip-installer/config/`
  - **Passwords are not stored in config**, including:
    - Zulip Admin Password
    - PostgreSQL Password
    - LDAP/AD Password
- Log Files Locations:
  - Installer Logs : `/etc/zulip-installer/log/`
  - Zulip Logs : `/var/log/zulip/`
  - PostgreSQL Logs : `/var/log/postgresql/`

Installation using config file

The installer supports non-interactive mode via a specified configuration file.
Use the following command to install using a specific configuration file in non-interactive mode:
`./install.pl --config=<file>`

All configuration keys **must be defined** in the config file according to the requirements of the selected service type.

Config file keys descriptions

ZULIP_SERVICE_TYPE: Type of Zulip service to install. Options: `Standalone PostgreSQL`, `Standalone Zulip`, or `Combined`.

ZIMBRA_PROXY_HOST: Hostname (FQDN) of Zimbra admin console proxy URL.

ZIMBRA_PROXY_ADMIN_PORT: Admin console proxy port on the Zimbra proxy server (typically `9071`).

ZIMBRA_JWT_AUTH_KEY: The `zimbraChatJwtSecret` value configured in your Zimbra environment.

CSRF_TRUSTED_ORIGINS: Comma-separated list of all accessible Zimbra web client URLs (e.g., `https://web.mydomain.com, https://virtual_domain1.example.com`).

ZULIP_ADMIN_EMAIL: Email address of an existing administrator in Zimbra.

ZULIP_ADMIN_PASSWORD: Password of the above admin account in Zimbra.

ZULIP_VERSION: Supported Zulip version. Currently: `9.2`.

ZULIP_SERVER_IP: Private IP address of the Zulip server, used by PostgreSQL to allow trusted connections.

CHAT_SERVER_HOSTNAME: Fully Qualified Domain Name (FQDN) of the Zulip/Chat server.

SSL_CERT_TYPE: SSL certificate type. Options: `commercial`, `letsencrypt`, `self-signed`.

LETS_ENCRYPT_AGREEMENT: Must be set to `yes` if `SSL_CERT_TYPE` is `letsencrypt`. Indicates agreement to the Let's Encrypt Subscriber Agreement.

LDAP_AD_SOURCE: Directory source for user authentication. Options: `Zimbra LDAP`, `External LDAP`, or `AD`.

ZIMBRA_LDAP_HOST: Hostname of the Zimbra LDAP server. If you have multiple LDAP servers, choose one of them to use.

ZIMBRA_LDAP_PORT: Port used to connect to the Zimbra LDAP server.

ZIMBRA_LDAP_PASSWORD: Password used for Zimbra LDAP authentication.

ZIMBRA_LDAP_PROTOCOL: Protocol for Zimbra LDAP connection. Options: `ldap`, `ldaps`.

EXTERNAL_LDAP_HOST: Hostname of the external LDAP or Active Directory server. If you have multiple LDAP/AD servers, choose one of them to use.

EXTERNAL_LDAP_PORT: Port used by the external LDAP/AD server.

EXTERNAL_LDAP_PASSWORD: Password for connecting to the external LDAP/AD server.

AUTH_LDAP_BIND_DN: Distinguished Name (DN) used for LDAP/AD binding.

LDAP_EMAIL_ATTRIBUTE: Attribute name that holds the user's email in LDAP/AD.

LDAP_FULLNAME_ATTRIBUTE: Attribute name that holds the user's full name in LDAP/AD.

EXTERNAL_LDAP_PROTOCOL: Protocol used for external LDAP/AD. Options: `ldap`, `ldaps`.

REMOTE_POSTGRES_READY: `yes` or `no`. Indicates whether PostgreSQL is already installed and ready on another VM.

POSTGRES_HOST: Hostname or IP address of the remote PostgreSQL server.

POSTGRES_PORT: Port on which PostgreSQL is running (typically `5432`).

POSTGRES_VERSION: Version of the PostgreSQL server. Can be retrieved from `/etc/zulip/zulip.conf` on the PostgreSQL host.

POSTGRES_PASSWORD: Password to assign for the Zulip user in the PostgreSQL database.

POSTGRES_SSL_MODE: SSL mode to use when connecting to PostgreSQL.

- For more details refer official PostgreSQL docs: SSL modes.

Config File Examples

Combined (Zulip + PostgreSQL)

```
ZULIP_SERVICE_TYPE=Combined
ZIMBRA_PROXY_HOST=
ZIMBRA_PROXY_ADMIN_PORT=
ZULIP_ADMIN_EMAIL=
ZULIP_ADMIN_PASSWORD=
CHAT_SERVER_HOSTNAME=
ZULIP_VERSION=
SSL_CERT_TYPE=commercial/letsencrypt/self-signed
ZIMBRA_JWT_AUTH_KEY=
CSRF_TRUSTED_ORIGINS=
LDAP_AD_SOURCE=Zimbra LDAP / External LDAP / AD
```

If `SSL_CERT_TYPE=letsencrypt`, include:

```
LETS_ENCRYPT_AGREEMENT=yes
```

If `LDAP_AD_SOURCE=Zimbra LDAP`, include:

```
ZIMBRA_LDAP_HOST=
ZIMBRA_LDAP_PORT=
ZIMBRA_LDAP_PASSWORD=
ZIMBRA_LDAP_PROTOCOL=
```

If `LDAP_AD_SOURCE=External LDAP / AD`, include:

```
EXTERNAL_LDAP_HOST=
EXTERNAL_LDAP_PORT=
EXTERNAL_LDAP_PASSWORD=
AUTH_LDAP_BIND_DN=
LDAP_EMAIL_ATTRIBUTE=
LDAP_FULLNAME_ATTRIBUTE=
EXTERNAL_LDAP_PROTOCOL=
```

Standalone PostgreSQL

```
ZULIP_SERVICE_TYPE=Standalone PostgreSQL
ZIMBRA_PROXY_HOST=
ZIMBRA_PROXY_ADMIN_PORT=
ZULIP_ADMIN_EMAIL=
ZULIP_ADMIN_PASSWORD=
ZULIP_VERSION=
ZULIP_SERVER_IP=
POSTGRES_PASSWORD=
```

Standalone Zulip

```
ZULIP_SERVICE_TYPE=Standalone Zulip
ZIMBRA_PROXY_HOST=
ZIMBRA_PROXY_ADMIN_PORT=
ZULIP_ADMIN_EMAIL=
```

```
ZULIP_ADMIN_PASSWORD=
CHAT_SERVER_HOSTNAME=
ZULIP_VERSION=
SSL_CERT_TYPE=commercial/letsencrypt/self-signed
ZIMBRA_JWT_AUTH_KEY=
CSRF_TRUSTED_ORIGINS=
LDAP_AD_SOURCE=Zimbra LDAP / External LDAP / AD
```

If `SSL_CERT_TYPE=letsencrypt`, include:

```
LETS_ENCRYPT_AGREEMENT=yes
```

If `LDAP_AD_SOURCE=Zimbra LDAP`, include:

```
ZIMBRA_LDAP_HOST=
ZIMBRA_LDAP_PORT=
ZIMBRA_LDAP_PASSWORD=
ZIMBRA_LDAP_PROTOCOL=
```

If `LDAP_AD_SOURCE=External LDAP / AD`, include:

```
EXTERNAL_LDAP_HOST=
EXTERNAL_LDAP_PORT=
EXTERNAL_LDAP_PASSWORD=
AUTH_LDAP_BIND_DN=
LDAP_EMAIL_ATTRIBUTE=
LDAP_FULLNAME_ATTRIBUTE=
EXTERNAL_LDAP_PROTOCOL=
```

For remote PostgreSQL support, set:

```
REMOTE_POSTGRES_READY=yes/no
```

If `REMOTE_POSTGRES_READY=yes`, include:

```
POSTGRES_HOST=
POSTGRES_PORT=
POSTGRES_PASSWORD=
POSTGRES_VERSION=
POSTGRES_SSL_MODE=
```

Configure Standalone Zulip Server with Standalone PostgreSQL

Run the installer with `--use-remote-db` and config file:

```
./install.pl --use-remote-db --config=<file>
```

including the following in the config:

```
REMOTE_POSTGRES_READY=yes
POSTGRES_HOST=
POSTGRES_PORT=
POSTGRES_PASSWORD=
POSTGRES_VERSION=
POSTGRES_SSL_MODE=
```

Configure Chat server

Major settings are defined in `/etc/zulip/settings.py` on Chat server. It is configured during the installation.

Configuration for authentication

If you face an issue on user authentication, check the following settings:

```
## Specify URI
## If port number is not 389, specify port number as well.
## If ldap over ssl is used, replace "ldap://" with "ldaps://"
AUTH_LDAP_SERVER_URI = "ldap://LDAP_HOST(:PORT)"

## Set a right BIND DN on your environment
AUTH_LDAP_BIND_DN = "uid=zimbra,cn=admins,cn=zimbra"

## Configure user search query
## Specify right objectClass(es) to search user accounts only.
AUTH_LDAP_USER_SEARCH = LDAPSearch(
    "", ldap.SCOPE_SUBTREE, "(&(mail=%(user)s)(objectClass=zimbraAccount))"
)

## Specify an attribute which stores user's email address
LDAP_EMAIL_ATTR = "mail"
AUTH_LDAP_USERNAME_ATTR = "mail"

## Specify the same objectClass(es) in search query as AUTH_LDAP_USER_SEARCH
## Note that "%(email)s" is used here. AUTH_LDAP_USER_SEARCH uses "%(user)s".
AUTH_LDAP_REVERSE_EMAIL_SEARCH = LDAPSearch("",
     ldap.SCOPE_SUBTREE, "(&(mail=%(email)s)(objectClass=zimbraAccount))")

## Specify an attribute which stores user's full name
AUTH_LDAP_USER_ATTR_MAP = {
    "full_name": "cn",
    ...
}
```

Only a single ldap server can be set in `AUTH_LDAP_SERVER_URI`. If Multi-Master Replication (MMR) is configured on your ldap servers, you can choose one of them for `AUTH_LDAP_SERVER_URI`.

Passwords are stored in `/etc/zulip/zulip-secrets.conf`. It is configured during the installation.

```
## Zimbra LDAP Password specified at installation
## It must be the same as zimbra_ldap_password in localconfig. (i.e. "zmlocalconfig -s zimbra_ldap_password")
auth_ldap_bind_password = LDAP_BIND_PASSWORD

## Zimbra JWT Auth Key specified at installation
zimbra_jwt_auth_key = RANDOM_STRING_SECRET_KEY
```

You can confirm that the Chat server can search an account on the ldap. Make sure that both `full_name` and `email` are returned.

`manage.py` is a command to manage various items on the Chat server. `query_ldap` is one of options and sends ldap search query to a ldap server specified in `AUTH_LDAP_SERVER_URI`.

```
(Run as zulip on Chat server)
$ /home/zulip/deployments/current/manage.py query_ldap user1@mydomain.com
...
full_name: user1
email: user1@mydomain.com
```

If `settings.py` or `zulip-secrets.conf` is modified, Chat server application needs to be restarted.

```
(Run as zulip on Chat server)
$ /home/zulip/deployments/current/scripts/restart-server
```

Note: If your ldap server uses a self-signed certificate, LDAP access from Chat server will fail because of an untrusted certificate.
To resolve this issue, a commercial certificate should be deployed on ldap servers.
If obtaining a commercial certificate is not feasible, you can disable certificate verification using the following setting. However, it is insecure.
It is not recommended on production.

```
$ vi /etc/zulip/settings.py
================================================================
...
AUTH_LDAP_START_TLS = True

## Add the following line
AUTH_LDAP_GLOBAL_OPTIONS = { ldap.OPT_X_TLS_REQUIRE_CERT: ldap.OPT_X_TLS_NEVER }
================================================================
$ /home/zulip/deployments/current/scripts/restart-server
```

Warning: `auth_ldap_bind_password` and `zimbra_jwt_auth_key` in `/etc/zulip/zulip-secrets.conf` are sensitive data.

Please make sure permission 640 (or 600) is set.

Please consider to plan to change `zimbra_jwt_auth_key` periodically.

If you install a Chat server with `Chat Server SSL Certificate Type: Self-signed`, Zimbra mailstore server cannot communicate with the Chat server because of an untrusted certificate.
To resolve this issue, you need to set `ssl_allow_untrusted_certs` to `false` on all mailstore servers so that each mailstore server does not check SSL certificate. However, it will affect not only on communication with Chat server but also all communication initiated from the mailstore server. It will cause security issue.
Do not apply it on production.

```
(Run as zimbra on all Zimbra mailstore servers)
$ zmlocalconfig -e ssl_allow_untrusted_certs=true
$ zmmailboxdctl restart
```

Configuration to allow access from Zimbra web client

`CSRF_TRUSTED_ORIGINS` in `/etc/zulip/settings.py` must include all Zimbra web client URLs.

```
## URLs of Zimbra web client specified at installation.
## If virtual hostname is used on Zimbra for domain(s), all access urls need to be listed.
## If access by IP address URL is allowed, it needs to be added as well.
(example 1)
CSRF_TRUSTED_ORIGINS = ["https://web.mydomain.com"]

(example 2)
CSRF_TRUSTED_ORIGINS = ["https://web.mydomain.com", "https://virtual_domain1.example.com", "https://10.0.0.2"]
```

If it is modified, Chat server application needs to be restarted.

```
(Run as zulip on Chat server)
$ /home/zulip/deployments/current/scripts/restart-server
```

Configuration for external LDAP and Active Directory

Here are examples of configuration for external LDAP and Active Directory.

External LDAP

```
## Replace values of the following keys.

## Specify URI
AUTH_LDAP_SERVER_URI = "ldap://YOUR_EXTERNAL_LDAP_HOST"

## If port number is not 389, specify port number as well.
## If ldap over ssl is used, replace "ldap://" with "ldaps://"
AUTH_LDAP_SERVER_URI = "ldap://YOUR_EXTERNAL_LDAP_HOST:12345"

## Replace BIND DN to a right one on your environment
AUTH_LDAP_BIND_DN = "uid=zimbra,cn=admins,cn=zimbra"

## Configure user search query
## Specify right objectClass(es), depending on your ldap structure
AUTH_LDAP_USER_SEARCH = LDAPSearch(
    "", ldap.SCOPE_SUBTREE, "(&(mail=%(user)s)(objectClass=zimbraAccount))"
)

## Specify an attribute which stores user's email address
LDAP_EMAIL_ATTR = "mail"
AUTH_LDAP_USERNAME_ATTR = "mail"

## Specify the same objectClass(es) in search query as AUTH_LDAP_USER_SEARCH
## Note that "%(email)s" is used here. AUTH_LDAP_USER_SEARCH specifies "%(user)s".
AUTH_LDAP_REVERSE_EMAIL_SEARCH = LDAPSearch("",
     ldap.SCOPE_SUBTREE, "(&(mail=%(email)s)(objectClass=zimbraAccount))")

## Specify an attribute which stores user's full name
AUTH_LDAP_USER_ATTR_MAP = {
    "full_name": "cn",
    ...
}
```

It is similar to external LDAP configuration on Zimbra.
For more details refer: LDAP Authentication - Zimbra :: Tech Center

Active Directory

```
## Replace values of the following keys.

## Specify URI
AUTH_LDAP_SERVER_URI = "ldap://YOUR_AD_HOST:PORT"

## Replace BIND DN to a right one on your environment
## If "Administrator" is an account to access your AD,
AUTH_LDAP_BIND_DN = "Administrator"

## Configure user search query
## Replace "mail" with an attribute which stores user's email address
## Specify right objectClass(es), depending on your directory structure
AUTH_LDAP_USER_SEARCH = LDAPSearch(
    "", ldap.SCOPE_SUBTREE, "(&(objectclass=organizationalPerson)(objectclass=person)(objectclass=user)(mail=%(user)s))"
)

## Specify an attribute which stores user's email address
LDAP_EMAIL_ATTR = "mail"
AUTH_LDAP_USERNAME_ATTR = "mail"

```

```
# Note that "%(email)s" is used here. AUTH_LDAP_USER_SEARCH specifies "%(user)s".
# To access by Active Directory email address
AUTH_LDAP_REVERSE_EMAIL_SEARCH = LDAPSearch(
    "", ldap.SCOPE_SUBTREE, "(mail=%(email)s)"
)

## Specify an attribute which stores user's full name
AUTH_LDAP_USER_ATTR_MAP = {
    "full_name": "cn",
    ...
}
```

For another example, if `userPrincipalName` has an email address of a user,

```
AUTH_LDAP_USER_SEARCH = LDAPSearch(
    "", ldap.SCOPE_SUBTREE, "(&(objectclass=organizationalPerson)(objectclass=person)(objectclass=user)(userPrincipalName=%(user)s))"
)

LDAP_EMAIL_ATTR = "userPrincipalName"
AUTH_LDAP_USERNAME_ATTR = "userPrincipalName"

AUTH_LDAP_REVERSE_EMAIL_SEARCH = LDAPSearch(
    "", ldap.SCOPE_SUBTREE, "(mail=%(email)s)"
)
```

It is similar to external Active Directory configuration on Zimbra.
For more details refer: Configure authentication with Active Directory - Zimbra :: Tech Center

Common setting

Password for external LDAP or Active Directory access needs to be added to `/etc/zulip/zulip-secrets.conf` in the same way as using Zimbra LDAP.

```
auth_ldap_bind_password = LDAP_BIND_PASSWORD
```

Configure LDAP attributes

After you can confirm that the Chat server can search an account on the ldap, some ldap attributes need to be configured.
A value of `zimbraChatJwtSecret` must be the same as the one of `zimbra_jwt_auth_key` configured on Chat server.

```
(Run as zimbra on a Zimbra mailstore server)
$ zmprov mcf zimbraChatBaseHost chat1.mydomain.com
$ zmprov mcf zimbraChatJwtSecret RANDOM_STRING_SECRET_KEY
$ zmprov fc -a all
```

If you want to configure a different Chat server for domain(s), refer: Administrator Guide → (Optional) Multiple domains and multiple chat servers section.

Create a realm for an existing domain

A "realm" is an internal term of Chat server; it is corresponds to an individual Zimbra “domain”. This “realm” has a unique URL (a realm URL) that the proxy and mailstore servers use to communicate with Chat server.
When you install the Chat server, you need to create manually realms for the domains which already exist on the Zimbra server.
A realm for a domain created after the Chat server installation can be automatically propagated to the Chat server when the domain is created with `zimbraFeatureZulipChatEnabled TRUE`.

ℹ️ The format of a realm URL is `https://[zimbraZulipChatDomainId].[zimbraChatBaseHost]`. `zimbraChatBaseHost` must be set before a realm is created. `zimbraZulipChatDomainId` is set during realm creation. `zimbraZulipChatDomainId` does not need to be set manually.
For example, when `zimbraChatBaseHost` is `chat1.mydomain.com` and `zimbraZulipChatDomainId` is `domain1examplecom` for a domain `domain1.example.com`, the realm URL for the domain is `https://domain1examplecom.chat1.mydomain.com`.
Before realm creation, a hostname set in `zimbraChatBaseHost` (`chat1.mydomain.com` in this case) must be resolvable on proxy and mailstore servers, and a valid SSL certificate for `chat1.mydomain.com` must have been deployed on the Chat server. In addition, the FQDN of a realm URL (`domain1examplecom.chat1.mydomain.com` in this case) must be resolvable by proxy and mailstore servers.

ℹ️ It may take some time to reflect a change on DNS.

⚠️ As described in Administrator Guide → Network & Firewall Configuration, a realm URL must be accessible from Zimbra proxy and mailstore servers **only**. It must not be accessed by end users.

You can create a realm for an existing domain in the following steps:

1. Confirm zimbraChatBaseHost and zimbraChatJwtSecret have been configured on a domain

```
(Run as zimbra on a Zimbra mailstore server)
zmprov gd DOMAIN zimbraChatBaseHost zimbraChatJwtSecret
```

2. Confirm a FQDN of a creating realm URL can be resolved by proxy and mailstore servers
3. Create a realm for an existing domain
   ℹ️ It is better to set a domain name removing dots (`.`) in `@id` for consistency.

```
(Run as zimbra on a Zimbra mailstore server)
$ zmsoap -vv -z -type admin CreateChatRealmRequest @id="domain1examplecom" / domain="domain1.example.com"
@by="name"
```

ℹ️ When a domain is created after Chat server is integrated, a realm can be created automatically on Chat server. `zimbraFeatureZulipChatEnabled TRUE` needs to be specified at domain creation.

```
(Run as zimbra on a Zimbra mailstore server)
## Realm is not created automatically
$ zmprov cd DOMAIN

## Realm is created automatically
$ zmprov cd DOMAIN zimbraFeatureZulipChatEnabled TRUE
```

A realm id (`zimbraZulipChatDomainId`) will be a string which removes dots from a domain name.
e.g. `another.domain.example.net` => `anotherdomainexamplenet`

Make sure that a FQDN of a realm URL for new domain can be resolved by proxy and mailstore servers before a domain is created.

If multiple Chat servers are deployed (i.e. if `zimbraChatBaseHost` and `zimbraChatJwtSecret` have not been set in globalConfig), `zimbraChatBaseHost` and `zimbraChatJwtSecret` need to be specified as well.

```
## Realm is created automatically
$ zmprov cd DOMAIN zimbraFeatureZulipChatEnabled TRUE zimbraChatBaseHost CHAT_HOSTNAME
zimbraChatJwtSecret SECRET
```

4. Confirm that `zimbraZulipChatDomainId` has been set on the domain

```
(Run as zimbra on a mailstore server)
$ zmprov gd DOMAIN zimbraZulipChatDomainId
zimbraZulipChatDomainId: domain1examplecom
```

5. Confirm that a realm has been created on Chat server

```
(Run as zulip on Chat server)
$ /home/zulip/deployments/current/manage.py list_realms
id    string_id            name                          domain
--    ---------            ----                          ------
...
N     domain1examplecom    domain1.example.com           https://domain1examplecom.chat1.mydomain.com
```

6. Run `proxyconfgen` command and reload proxy so that Zimbra proxy can forward a request to Zimbra Chat server with a right url.

```
(Run as zimbra on all Zimbra proxy servers)
$ /opt/zimbra/libexec/zmproxyconfgen
$ zmproxyctl reload
```

If you see the following error message,

```
$ zmproxyctl reload
Reloading proxy...nginx: [emerg] could not build map_hash, you should increase map_hash_bucket_size: 64
done.
```

configure the value of `proxy_web_map_hash_bucket_size` in localconfig (default: `64`). Zimbra proxy needs to be restarted.

```
(Run as zimbra on all proxy servers)
$ zmlocalconfig -e proxy_web_map_hash_bucket_size=128
$ /opt/zimbra/libexec/zmproxyconfgen
$ zmproxyctl restart
```

If you see the following error message,

```
Reloading proxy...nginx: [emerg] host not found in upstream "domain1examplecom.chat1.mydomain.com:443" in
/opt/zimbra/conf/nginx/includes/nginx.conf.chat.upstream:66
done.
```

the proxy server cannot resolve the FQDN of a realm URL `https://domain1examplecom.chat1.mydomain.com`. You need to check DNS record, or refresh DNS cache on the server.
It may take some time to reflect a change on DNS. If it needs be fixed immediately, you can add an entry to `/etc/hosts` on the server as a workaround.

7. Provision accounts on the realm
   For more details refer: Administrator Guide
   - Login to admin console
   - Go to Configure > Domain > the domain
   - Click “Provision all accounts” button
   - Click “Get accounts” button and confirm accounts have been deployed on Chat server

When an account is created after Chat server is integrated, a chat account is automatically created on Chat server.
Note: `zimbraFeatureZulipChatEnabled` must be TRUE on a domain.

```
(Run as zimbra on a Zimbra mailstore server)
## account is not created automatically on Chat server
$ zmprov ca ACCOUNT PASSWORD zimbraFeatureZulipChatEnabled FALSE

## account is created automatically on Chat server
$ zmprov ca ACCOUNT PASSWORD zimbraFeatureZulipChatEnabled TRUE
```

```
## account is created autmatically on Chat server when zimbraFeatureZulipChatEnabled is TRUE on a COS
$ zmprov ca ACCOUNT PASSWORD
```

Enable Basic Chat

1. Enable `zimbraFeatureZulipChatEnabled` and `zimbraFeatureBasicOneToOneChatEnabled` at domain, COS or account level

```
(Run as zimbra on a Zimbra mailstore server)
(Example)
$ zmprov mc default zimbraFeatureZulipChatEnabled TRUE zimbraFeatureBasicOneToOneChatEnabled TRUE
```

2. Enable `zimbra-zimlet-chat` zimlet in domain, COS or account level
3. Login to an account on Modern Web Client. Basic Chat is shown at right side of the page (in sidebar).
   a. For more details refer separate document: “User's Guide for Zimbra Chat”.

Enable Advanced Chat

1. Enable `zimbraFeatureZulipChatEnabled` and `zimbraFeatureAdvancedChatEnabled` at domain, COS or account level

```
(Run as zimbra on a Zimbra mailstore server)
(Example)
$ zmprov mc default zimbraFeatureZulipChatEnabled TRUE zimbraFeatureAdvancedChatEnabled TRUE
```

2. Enable `zimbra-zimlet-chat` zimlet in domain, COS or account level
3. Login to an account on Modern Web Client. Chat icon is shown in Apps list at the top of the page.
   a. For more details refer separate document: “User's Guide for Zimbra Chat”.

(Optional) Multiple domains and multiple chat servers

You can deploy multiple domains on a single Chat server or distribute multiple domains across different Chat servers. This can be configured using `zimbraChatBaseHost`

Note: A single domain cannot be deployed across multiple Chat servers. Each domain must be deployed on a specific Chat server.

A Chat server can connect to a single LDAP server (`AUTH_LDAP_SERVER_URI`), which can support multiple domains. However, if any domain requires a different LDAP server for authentication, a separate Chat server must be set up for it.

```
(Run as zimbra on a Zimbra mailstore server)
## set globalConfig zimbraChatBaseHost to empty to avoid misconfiguration
$ zmprov mcf zimbraChatBaseHost ""
## set zimbraChatBaseHost per domain.
## In this case, domain1 and domain2 are hosted on chat1, and others on chat2.
$ zmprov md domain1.mydomain.com zimbraChatBaseHost chat1.mydomain.com
$ zmprov md domain2.mydomain.com zimbraChatBaseHost chat1.mydomain.com
$ zmprov md domain3.mydomain.com zimbraChatBaseHost chat2.mydomain.com
$ zmprov md domain4.mydomain.com zimbraChatBaseHost chat2.mydomain.com
$ zmprov fc -a all
```

As optional, `zimbraChatJwtSecret` can be configured per Chat server. (i.e. either the same or different values can be set.)

```
(Run as zimbra on a Zimbra mailstore server)
## set globalConfig zimbraChatJwtSecret to empty to avoid misconfiguration
$ zmprov mcf zimbraChatJwtSecret ""
## set a value for chat1 server
$ zmprov md domain1.mydomain.com zimbraChatJwtSecret tx4BhsK9aGmPzyqAQ66jawN
```

```
$ zmprov md domain2.mydomain.com zimbraChatJwtSecret tx4BhsK9aGmPzyqAQ66jawN
## set a value for chat2 server
$ zmprov md domain3.mydomain.com zimbraChatJwtSecret bH94LxpaaC12jkyTgnmMcqs
$ zmprov md domain4.mydomain.com zimbraChatJwtSecret bH94LxpaaC12jkyTgnmMcqs
$ zmprov fc -a all
```

For each Chat Server deployment repeat the steps from Installation Guide: From Install Chat server to Enable Basic Chat and/or Enable Advanced Chat.

(Optional) Install and enable PGroonga for full-text search in non-English languages on Advanced Chat

PGroonga is a PostgreSQL extension to use Groonga as index. PGroonga makes PostgreSQL fast full text search platform for all languages.

All steps should be run as `root` on Chat server. It does not need to be executed on PostgreSQL server.

1. Alter the deployment setting

```
crudini --set /etc/zulip/zulip.conf machine pgroonga enabled
```

2. Update the deployment to respect that new setting

```
/home/zulip/deployments/current/scripts/zulip-puppet-apply
```

3. Add the following setting to `/etc/zulip/settings.py`

```
USING_PGROONGA = True
```

4. Apply the PGroonga migrations.
   Note that the migration may take a long time, and users will be unable to send new messages until the migration finishes.

```
su zulip -c '/home/zulip/deployments/current/manage.py migrate pgroonga'
```

5. Restart Zulip

```
su zulip -c '/home/zulip/deployments/current/scripts/restart-server'
```

If you see the error below at step 2, run `apt update` and then do the step 2 again.

```
# /home/zulip/deployments/current/scripts/zulip-puppet-apply
...
Get:1 http://iad-ad-1.clouds.archive.ubuntu.com/ubuntu jammy/universe amd64 libmsgpackc2 amd64 3.3.0-4 [15.1 kB]
Err:2 http://ppa.launchpad.net/groonga/ppa/ubuntu jammy/main amd64 libgroonga0 amd64 15.0.2-1.ubuntu22.04.1
  404  Not Found [IP: 185.125.190.80 80]
Get:3 http://iad-ad-1.clouds.archive.ubuntu.com/ubuntu jammy/universe amd64 libsimdjson9 amd64 1.0.2-2 [68.6 kB]
Get:4 https://packages.groonga.org/ubuntu jammy/universe amd64 postgresql-16-pgdg-pgroonga amd64 4.0.1-1 [789 kB]
Fetched 873 kB in 3s (265 kB/s)
E: Failed to fetch http://ppa.launchpad.net/groonga/ppa/ubuntu/pool/main/g/groonga/libgroonga0_15.0.2-1.ubuntu22.04.1_amd64.deb  404  Not Found [IP: 185.125.190.80 80]
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
Error: /Stage[main]/Zulip::Postgresql_base/Package[postgresql-16-pgdg-pgroonga]/ensure: change from 'purged' to 'latest' failed: Could not update: Execution of '/usr/bin/apt-get -q -y -o DPkg::Options::=--force-confold install postgresql-16-pgdg-pgroonga' returned 100: Reading package lists...
...
```

Administrator Guide

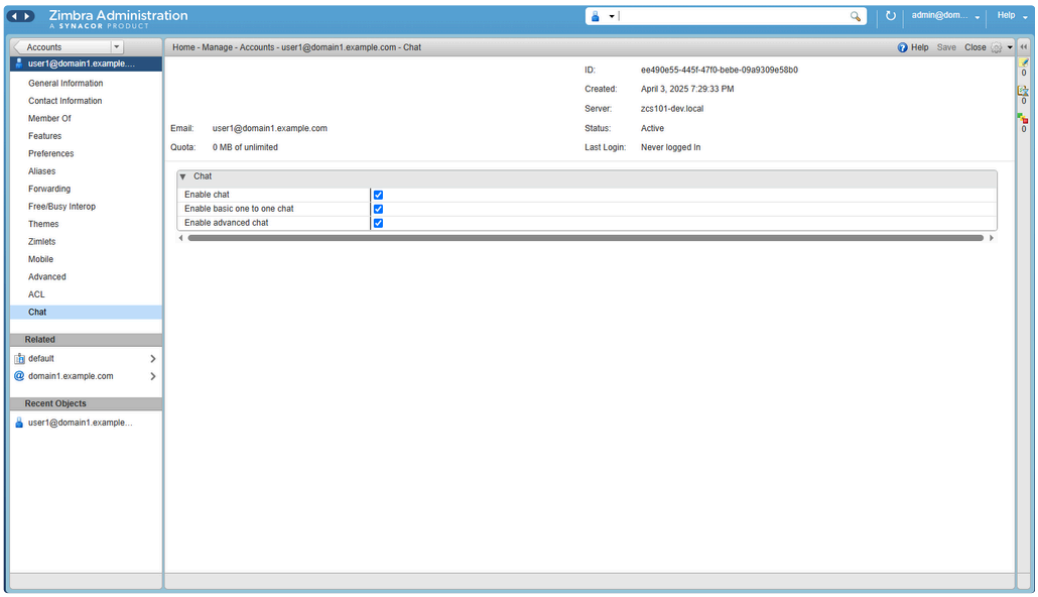
Admin Console

An administrator can manage Chat-related configuration and accounts on a Chat server.

For more details refer: FAQs section for a domain admin or delegated admin.

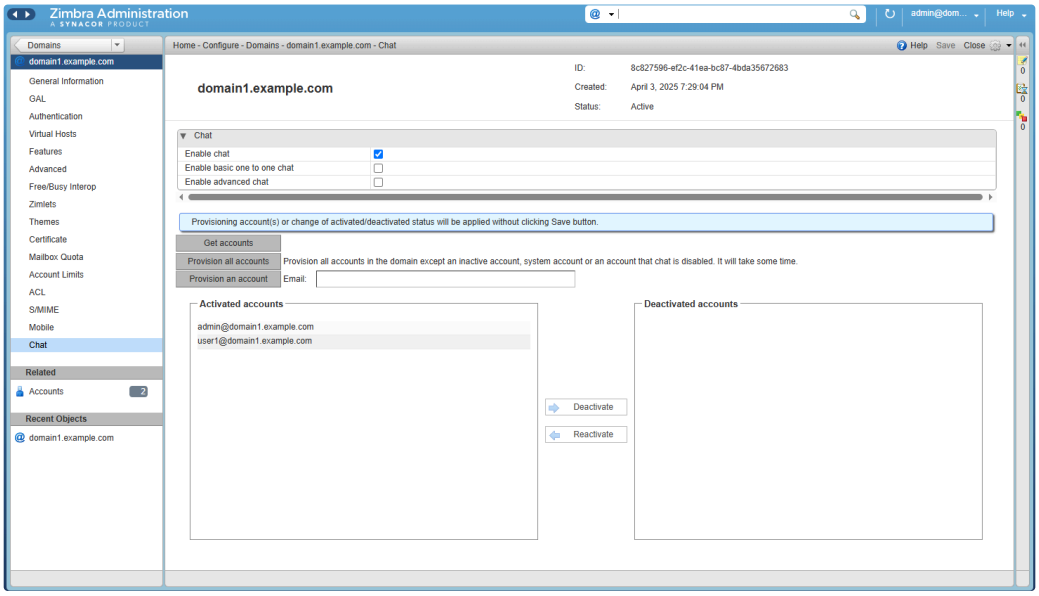
Account, COS and domain settings:

- Enable chat: whether chat feature is enabled or not. If it is FALSE, basic one to one chat and advanced chat is regarded as disabled.
- Enable basic one to one chat: whether basic chat is enabled or not.
- Enable advanced chat: whether advanced chat is enabled.



Chat accounts management in domain settings:

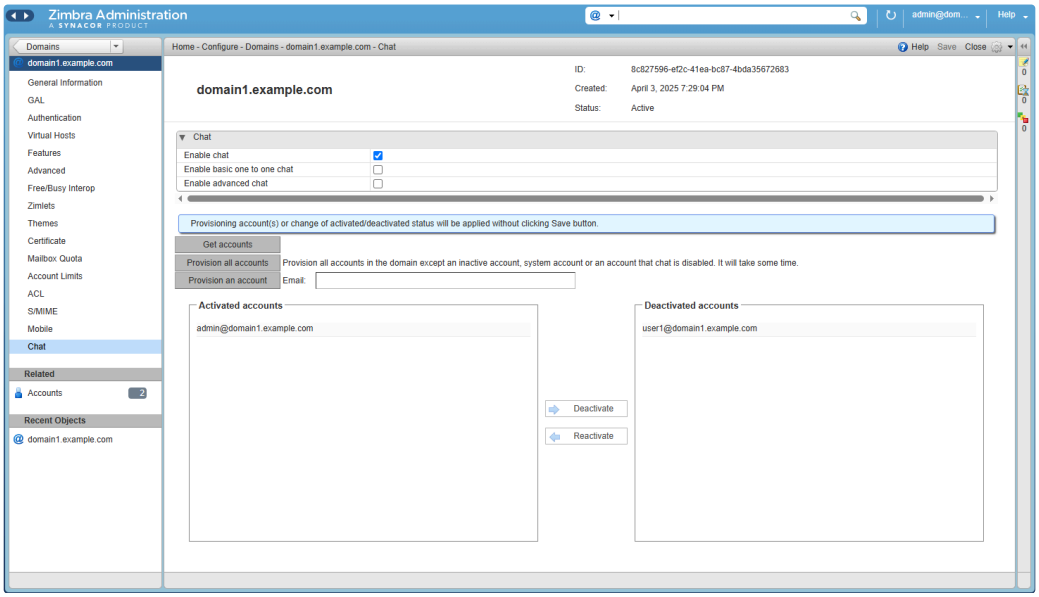
- Get accounts: accounts created on a Chat server are shown in Activated accounts and Deactivated accounts field.



- Deactivate/Reactivate: click an account in Activated accounts and click Deactivate, and then the user is deactivated. It can be reactivated by clicking an account in Deactivated accounts and click Reactivate.

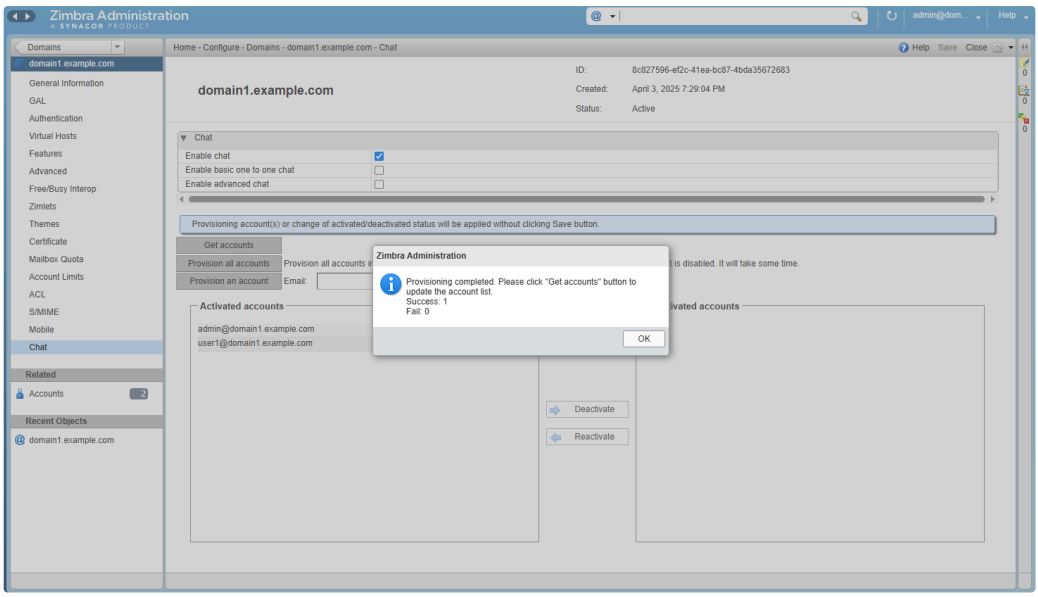
An account is deactivated on Chat server automatically when it is removed from Zimbra.

For more details refer: FAQs section on how to remove an account from Chat server.

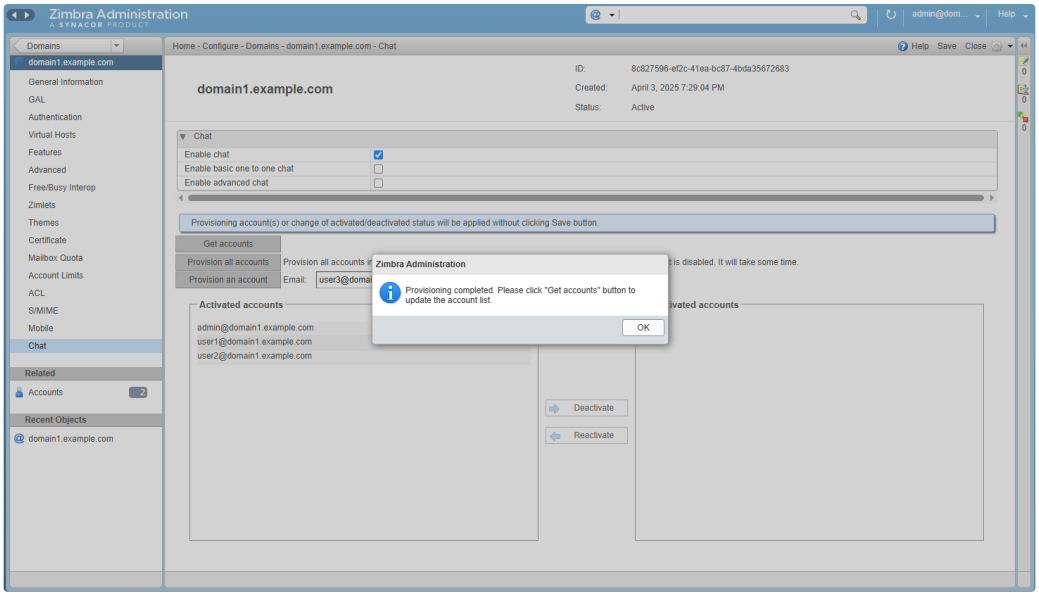


- Provision all accounts: provision all accounts on the domain to a Chat server. Activated/Deactivated accounts lists are not updated automatically. Click Get accounts to show the latest list.

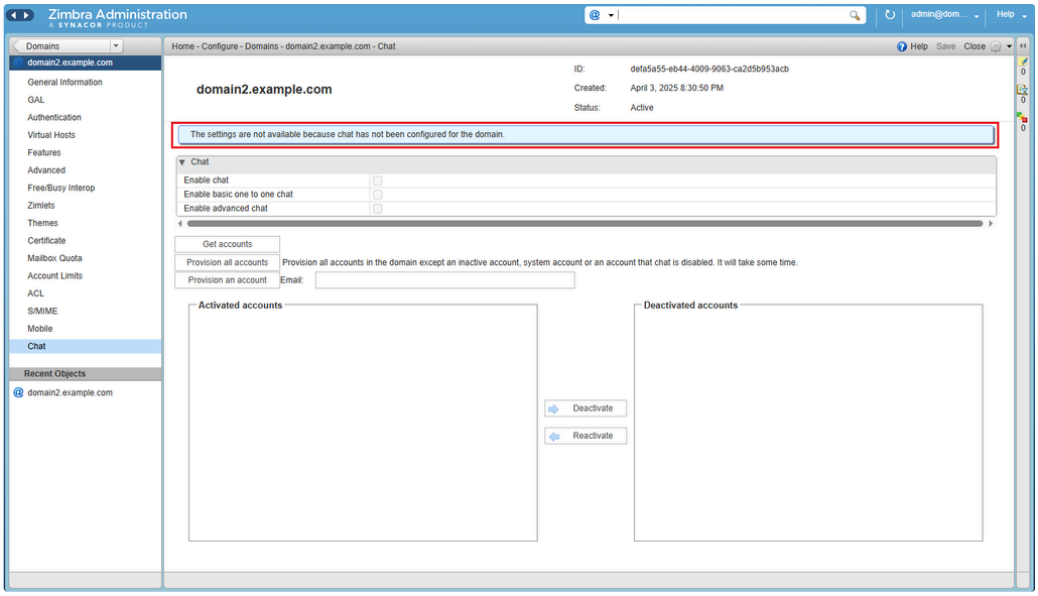
Note: The number of successes and failures of account provisioning are shown in a dialog. When you see failures, the failure event log `user creation failed: EMAIL_ADDRESS` is written in `/var/log/zulip/server.log` on the Chat server. You can find the reason for the failure in the `/var/log/zulip/server.log` and/or `/var/log/zulip/error.log`.



- Provision an account: provision a single account on the domain to a Chat server. Enter an email address in text field and click Provision an account. Activated/Deactivated accounts lists are not updated automatically. Click Get accounts to show the latest list.



If a realm has not been created on Chat server, chat settings is unavailable on the domain.



Zimlet Configuration Guide for one to one chat

Basic behavior

On one to one chat, presence data of all users in the same domain are fetched from Zimbra Chat server at initial loading process. After that, presence status (active or idle) of the user is sent to Chat server periodically (hereinafter called “presence update request”.) A response from the server includes presence data of other users whose last timestamp is updated. In addition, when other user(s) becomes active status, it is notified through a response of a `events` request (hereinafter called “presence event notification”.)

Possible performance issue on a large domain

If there are many users on a domain, it may increase some performance issues.

- response size becomes large at initial loading process
- the size of presence data included in a response of a *presence update request* becomes large, especially when many users are online. (i.e. many concurrent users)
- the number of *presence event notification* becomes large

Configuration

A zimlet zimbra-zimlet-chat has some configuration.
`config_template.xml` is included in the zimlet. You can see additional information about the settings.

1. `pingIntervalActiveInSeconds`: default `180` (seconds)

- When the user is active, the status is sent to Chat server every 3 minutes.
  i.e. it is used on a *presence update request* at active status.

2. `pingIntervalIdleInSeconds`: default `900` (seconds)

- When the user is idle, the status is sent to Chat server every 15 minutes.
  i.e. it is used on a *presence update request* at idle status.

3. `maxAllowedSecondsForNoActivity`: default `240` (seconds)

- When the user has not used a keyboard or a mouse (touch action on mobile device) for 4 minutes, the user is regarded as idle status. After that, `pingIntervalIdleInSeconds` is used to send the user's presence status.
- When a user uses a keyboard or a mouse, the user status becomes from idle to active. After that, `pingIntervalActiveInSeconds` is used.

4. `presenceUpdateMinFreqInSeconds`: default `60` (seconds)

- When many *presence update requests* are sent from a client (or a browser) in a short period of time, Chat server ignores them not to update database frequently.
- By default, minimum update period is `55` seconds. For example, when Chat server receives 5 requests in 55 seconds from the same user, it accepts the first request and update the database, but ignores other 4 requests.
- When the user becomes idle, the status is sent to Chat server. However, if it happens just after active status is sent to Chat server, the idle status notification is ignored on Chat server. The zimlet sends additional idle presence status after the specified value in consideration of the limitation.
  ⚠️ The value must not be changed until the minimum update period `55` seconds is modified on Chat server.

5. `idleEventCallDelay`: default `60` (seconds)

- An event like receiving a message and status change to online on other users are checked through *presence event notification*. When the user is active, next `events` request is sent immediately after a previous *presence event notification* fulfilled.
- When the user is idle, 1 minute wait time is added before next `events` request is sent.
- It means,
  - 1. Receive *presence event notification*
  - 2. Wait 1 minute
  - 3. Send next `events` request.
- If the user receives a message from other user or other user becomes online during the wait time, the event is notified in a response of next `events` request and reflected in UI.

6. `enablePresenceEvents`: default: `true`
   a. When it is `true`
      i. Presence data of all users in the same domain are fetched from Chat server at initial loading process.
      ii. When other user(s) becomes active status, it is notified through a response of a `events` request.
      iii. A response of *presence update request* includes presence data of other users' whose timestamp is updated.
   b. When it is `false`
      i. Presence data of all users are not loaded. Instead, presence data of only users shown in a sidebar (pinned and not pinned users) are fetched.
      ii. Presence status is NOT notified when other user(s) becomes active status.
      iii. A response of *presence update request* does NOT include presence data of other users' whose timestamp is updated.

iv. Instead of using *presence event notification* and a response of *presence update request*, another http request to get presence data of only necessary users is sent together at sending *presence update request.* (hereinafter called “presence fetch request”.)

c. **Advantage & Disadvantage**

i. When it is `true`,

ii. Advantage

1. When another user becomes active, it can be detected in shorter time

iii. Disadvantage

1. All other users' presence data are fetched at initial loading process. All other users' “active” status change are notified through *presence event notification*, even when the user communicates with some other users only. It means that response size of the initial loading may be large and many *presence event notification* may occur. It will increase memory use on a client machine, increase server load, and increase network traffic.

iv. When it is `false`,

v. Advantage

1. Size of initial loading is smaller.
2. Presence data of only necessary users are fetched from Chat server. It will achieve less memory use on a client, and less server load and network traffic especially on *presence event notification* in many case.

vi. Disadvantage

1. When a user becomes active, it is not reflected on other users immediately. It will be reflected when next presence request is triggered.
2. Additional *presence fetch request* is sent. There might be some overhead on both client and server side. However, it should not be so much in comparison with fetching presence update through a response of *presence update request* in a large domain.

Constraints

- The values for `pingIntervalActiveInSeconds`, `pingIntervalIdleInSeconds`, `maxAllowedSecondsForNoActivity` and `presenceUpdateMinFreqInSeconds` must be provided in seconds.
- The minimum values required for `pingIntervalActiveInSeconds`, `pingIntervalIdleInSeconds` and `presenceUpdateMinFreqSeconds` must be 60 seconds.
- `presenceUpdateMinFreqInSeconds` value must be less than `pingIntervalActiveInSeconds` and `pingIntervalIdleInSeconds`.
- `presenceUpdateMinFreqInSeconds` must not be changed unless a setting of minimum frequent seconds for presence update is modified on Zimbra Chat server.
- The maximum value required for `idleEventCallDelay` must be 60 seconds.

Recommended settings

On a small domain

- Default values should be good

On a large domain

- `enablePresenceEvents` is `false`
- If you see performance issue on Chat server, consider a) increasing `pingIntervalActiveInSeconds`, b) increasing `pingIntervalIdleInSeconds` and/or c) decrease `maxAllowedSecondsForNoActivity` to reduce frequency of *presence update request*. :
  They are sort of tuning parameters. Efficiency of changes depend on the number of concurrent users, Chat server specification, network bandwidth and so on.

Configuration steps

Modify `config_template.xml` and apply the change after the zimlet is deployed.
All steps should be run as `zimbra` on a Zimbra mailstore server.

1. deploy the zimlet
2. check the current configuration

```
$ zmzimletctl info zimbra-zimlet-chat
...
Config: <zimletConfig name="zimbra-zimlet-chat" version="1.0.0">...</zimletConfig>
```

3. edit `config_template.xml`. For example,

```
$ cd /path/to/your-working-directory/
$ cp /opt/zimbra/zimlets-deployed/zimbra-zimlet-chat/config_template.xml ./
$ vi config_template.xml
```

   - set upper version number
   - modify a value of setting(s)

4. apply the change

```
$ zmzimletctl configure config_template.xml
```

5. confirm the change

```
$ zmzimletctl info zimbra-zimlet-chat
```

6. run flushCache. Specify `-a` option on a multi-server environment.

```
$ zmprov fc -a zimlet
```

Log files

Main log file of Chat server application is `/var/log/zulip/server.log` on a Chat server. You can find other log files like `django.log` and `tornado.log` in `/var/log/zulip/` directory, but major log events are gathered in `server.log`.

The log format of `server.log` is as follows:

- Timestamp
- Log level
- Logger name, abbreviated as “zr” for these Zulip request logs
- IP address
- HTTP method
- HTTP status code
- Time to process
- (Optional performance data details)
- Endpoint/URL from zproject/urls.py

Nginx log files `access.log` and `error.log` exist in `/var/log/nginx/` directory on a Chat server. The format of `access.log` is defined in `/etc/nginx/nginx.conf` as follows:

```
log_format combined_with_host_and_time '$remote_addr - $remote_user [$time_local] '
                                       '"$request" $status $body_bytes_sent '
                                       '"$http_referer" "$http_user_agent" $host $request_time';
```

PostgreSQL log files exist in `/var/log/postgresql` directory. It does not contain detailed logs but just statistics by default. If you see an issue on database access, you may be able to find ERROR, FATAL or PANIC log in `postgresql-16-main.log`.

Zimbra Chat Administrator

Administration Organization basics

As a Zimbra Chat Administrator, understanding how to manage user roles and permissions is key to maintaining organized workspace. This section explains the different user roles available, how to manage them, and how to configure organization-wide permissions.

Note: These roles are chat specific and not same as the Zimbra email administrator role.

Overview of User Roles

Zimbra Chat supports multiple roles to help you assign the right permissions based on your organization's needs:

Role	Description
Organization Owner	Full control over settings, users, and other administrator. Only owners can deactivate the organization.
Organization Administrator	Can manage users, channels, settings. Cannot promote others to owner or modify owner roles.
Moderator	Can perform limited administrative functions. Permissions can be configured by Owner and Administrator.
Member	Default role for most users. Can access all public channels.
Guest	Can only access channels they are explicitly added to. Cannot see other users or channels.

Managing User Roles

You can assign and change roles either through the user profile or via organization settings.

Via User Profile

1. Hover over the user’s name in the right sidebar.
2. Click the three-dot menu → Manage this user.
3. Under User role, select a new role.
4. Click Save changes.

Via Organization Settings

1. Go to Settings → Organization Settings→ Users tab.
2. Use the dropdown next to the user’s name to assign a new role.
3. Click Save changes.

Organization Settings Management

Only Owner and Administrator can configure and view organization settings.

Common Settings:

- Organization profile and type.
- Organization logo can also be configured and customized.
- Default user permissions.

- Automated messages and email language.

To access:

- Go to Settings → Organization settings.

Managing User Access

Deactivate a User

Deactivation removes access across all devices, disables credentials, and bots.

Steps:

1. Open the user profile or go to Organization > Users tab.
2. Click Manage this user.
3. Click Deactivate user > Confirm.
4. Note: Owner role cannot be deactivated by Administrator.

Reactivate a User

1. Go to Settings → Organization → Users → Deactivated tab.
2. Click Reactivate next to the user.

The user retains all prior settings and roles upon reactivation.

Default Settings for New Users

Administrator can configure default preferences for new users, such as:

- Theme, font size, time format.
- Privacy options.
- Notifications.
- Home view (Inbox vs. Recent Conversations).

User Identity Controls

Configure these options under Organization settings → Organization permissions → User Identity:

- Prevent users from changing:
  - Name, Email, Avatar.

Deactivating an Organization

This action is irreversible from the organization side.

Only Owners can deactivate:

1. Go to Organization settings → Organization profile.
2. Click Deactivate organization → Confirm.

Note: To reverse a deactivation, contact your server administrator immediately.

Deactivation cannot be undone through the Zimbra Chat interface. Only a Zimbra system administrator has the necessary access and tools to recover or restore the organization from the server side.

Channel Management

As an Administrator, you have full control over how channels are created, managed, and used.

Control Who Can Create Channels

Go to: Settings → Organization settings → Organization permissions → Channel permissions

You can choose:

- Administrator only
- Administrator and Moderator.
- Administrator, Moderator, and Members.
- All full members.

Limit creation to Administrator and/or moderators if you want to keep things clean.

Enable/Disable Channel Types

Channel types:

- **Public** – anyone can join, all messages visible.
- **Private** – invite-only.
- **Web-public** – read-only access via public link.

To grant web-public type to Channel:

Only Administrator and Moderator can create web-public channels.

1. Go to Organization settings → Organization permissions → Channel permissions.
2. Enable: Allow web-public channels.

Set Who Can Post in Channels

You can change this anytime—even for existing channels.

For each channel:

1. Open Channel settings → click on the Channel name → Advanced configuration section.
2. Under Posting policy, choose: Everyone, Administrator & moderators, Only moderators, Administrator only.

Manage Subscribers

As an Administrator, you can:

- Add/remove users from any channel - Channel settings → click on the Channel name → Subscribers tab.
- Forcing users to subscribe to mandatory channels (via “default channels”)
  - Set Default Channels for New Users.
  - Go to: Settings → Organization settings → Default channels.
  - These channels are auto-subscribed for new users. Ideal for: #announcements, #company-news.

Rename Channel and Edit Channel Settings

Administrator can:

- Change the channel name and description.
- Make the channel public.
- Set posting policies.
- Delete channels.

Go to any channel → Channel settings → Make your changes.

Delete Channels

Administrator can:

- Delete channels - Deleting channel will immediately unsubscribe everyone. This action cannot be undone.

Require Topics in Channel Messages

By default, users can send messages without choosing a topic — these show up as “(no topic)”.

As an Administrator, you can prohibit users to use an empty topic when sending a message to a channel.

To enable this:

1. Go to Settings → Organization settings
2. Scroll to Other settings
3. Toggle Require topics in channel messages
4. Click Save changes

Control Who Can Edit Topics and Move Messages

As an Administrator, you can decide:

- Who can edit message topics
- Who can move topics to other channels
- Whether there’s a time limit for editing or moving messages

To manage these permissions:

1. Go to Settings → Organization setting → Organization permissions
2. Under Moving messages, configure each of these:
    - Who can move messages to another topic
    - Time limit for editing topics
    - Who can move messages to another channel
    - Time limit for moving messages between channels
3. Click Save changes after each update

Administrator and Moderator are able to edit and move messages no matter how long ago they were written.

Control Who Can Send Direct Messages (DMs)

As Owner or Administrator, you can control how DMs work in your organization.

There are two settings you can configure:

To set up DM permissions:

1. Go to Settings → Organization settings → Organization permissions
2. Scroll to Direct message permissions
3. Who can authorize a DM (At least one person in the conversation must be authorized to start it.)
    a. Set Who can authorize a direct message conversation
    b. Click Save changes
4. Who can start a DM (Defines who’s allowed to initiate new DM conversations.)
    a. Set Who can start a direct message conversation
    b. Click Save changes
5. To disable DMs completely:
    a. Set Who can authorize a direct message conversation to
    Direct messages disabled
    b. Click Save changes

Control Who Can Use Wildcard Mentions (@all, @channel, @topic)

As Owner or Administrator, you can manage who’s allowed to use wildcard mentions that notify large groups of people.

Wildcard mentions include:

- `@all`, `@channel`, or `@everyone` → Notifies everyone in a channel
- `@topic` → Notifies everyone active in a topic

By default, these can notify a lot of users. You can limit this to avoid spamming large groups.

This setting only applies when the channel has more than 15 members, or the topic has more than 15 participants

To restrict wildcard mentions:

1. Go to Settings → Organization settings → Organization permissions
2. Scroll to Channel permissions
3. Set Who can notify a large number of users with a wildcard mention
4. Click Save changes

Manage Message Editing, Deleting & Edit History

As an administrator, you control:

- Who can edit or delete their own messages
- Time limits for editing or deleting
- Whether edit history is visible to users

Message Editing

- Users can only edit their own messages.
- Administrator can always delete any message.

To configure editing:

1. Go to Settings → Organization settings → Organization permissions
2. Under Message editing:
   - Toggle Allow message editing.
   - Toggle Enable message edit history.
   - Set Time limit for editing messages.
3. Click Save changes.

Message Deletion

- Deleting a message permanently removes the message, users can delete their own messages.
- Administrator can delete any message.
- Users can also delete messages sent by bots they own.

To configure deletion:

1. Go to Settings → Organization settings → Organization permissions
2. Under Message deletion:
   - Set Who can delete their own messages.
   - Set Time limit for deleting messages - does not apply to administrator.
3. Click Save changes.

Message retention

As Administrator, you can also configure a global default message retention policy:

- Go to Organization settings → Message retention.
- Select message retention period.
  - By default all messages are kept indefinitely.
  - Use custom retention period and select Retention period (in days).

Media Preview Behaviour

Zimbra chat shows small previews of images, videos, and links to keep chats readable.

As an administrator you can turn off previews entirely for:

Uploaded and linked images/videos and linked websites

1. Go to Settings → Organization settings
2. Under Other settings,
3. Toggle Show previews of uploaded and linked images and videos
4. Toggle Show previews of linked websites
5. Click Save changes.

Video

Zimbra Chat is a chat only solution, and does not include any video conferencing solution. However, an integration with open source chat solution has been provided. By default, Zimbra Advanced Chat provides a built in integration with Jitsi Meet, 100% open source video conferencing solution. Users will be able to start a Jitsi Meet call and invite others using the add video call or add voice call button in the compose box.

Configure a self-hosted instance of Jitsi Meet

Administrators can use a self-hosted instance of Jitsi Meet. Zimbra Advanced Chat has cloud version of Jitsi Meet as its default video call provider. This is the community version of the Jitsi open source platform.

To configure self-hosted Jitsi Meet, as an Administrator,

1. Click on the gear icon in the upper right corner of the web or desktop app.
2. Select Organization settings.
3. On the left, click Organization settings.
4. Under Compose settings, confirm Jitsi Meet is selected in the Call provider dropdown.
5. Select Custom URL from the Jitsi server URL dropdown, and enter the URL of your self-hosted Jitsi Meet server.
6. Click Save changes.

Backup and Restore For Zimbra Chat

The Zimbra Chat server has a built-in backup tool which can be used to make a backup of chat data on the Chat server and restore it in case of disaster recovery, testing with production data, and hardware migrations.

NOTE: The backup and restore functionality here is not related to Zimbra Backup and Restore feature used for mailbox data backup.

It is fast, robust, and minimizes disruption for your users. However, it has a few limitations:

- Backups must be restored on a server running the same Zulip version.
- Backups must be restored on a server running the same PostgreSQL version.

Backup

Zimbra Chat Administrator can create a backup of all data on the server using backup tool as:

```
# As the zulip user
/home/zulip/deployments/current/manage.py backup
```

Following options can be used with above command:

`--output=/tmp/backup.tar.gz` : Filename to write the backup tarball to (default: write to a file in /tmp). On success, the console output will show the path to the output tarball.

`--skip-uploads`: If `LOCAL_UPLOADS_DIR` is set, user-uploaded files in that directory will be ignored. `LOCAL_UPLOADS_DIR` is the directory where user uploaded files are stored in Chat server. It is specified in `/etc/zulip/settings.py`

This will generate a `.tar.gz` archive containing all the data stored on your Chat server that would be needed to restore your Chat server's state on another machine perfectly.

For example:

```
# su zulip -c '/home/zulip/deployments/current/manage.py backup --output=/home/zulip/zulip-backup.tar.gz'
+ /usr/lib/postgresql/16/bin/pg_dump --format=directory --file=/tmp/zulip-backup-2025-04-04-07-29-56-533sg975/zulip-backup/database --username=zulip --dbname=zulip --no-password --host=10.0.0.231 --port=5432
+ tar --directory=/tmp/zulip-backup-2025-04-04-07-29-56-533sg975 -cPzf /home/zulip/zulip-backup.tar.gz '--transform=s|^/etc/zulip(/.*)?$|zulip-backup/settings\1|x' -- zulip-backup/zulip-version zulip-backup/os-version zulip-backup/postgres-version /etc/zulip zulip-backup/database
Backup tarball written to /home/zulip/zulip-backup.tar.gz
```

Restore backup

1. Create a new ZImbra Chat server setup using the same base OS and the same versions of Zulip and postgreSQL as the backup was taken on. Use the Zimbra Zulip installer to install and configure the setup.
   a. Any upgrade related to OS, Zulip version or postgreSQL version should be done after successful restore of backup.
   b. Ensure network security measures recommended in the Installation guide are safely in place for the new instances.
2. Copy backed-up .tar.gz file from the old Zimbra Chat server to the new Chat server.
3. Follow further restore steps based on your Zimbra Chat server setup.

Restoration on multi-node Chat server setup:

In case of multi-node Chat server setup, where PostgreSQL server is also new (i.e. PostgreSQL database also needs to be restored on the new PostgreSQL server):

- Update following details in `/etc/zulip/settings.py` on Chat server node:

```
EXTERNAL_HOST = "new-chat-server-hostname"
REMOTE_POSTGRES_HOST = "<new-postgres-host-ip>"
```

- On PostgreSQL node, add the following entry at the bottom in `/etc/postgresql/*/main/pg_hba.conf`

```
host    postgres        zulip           <zulip-server-host-ip>/32            trust
```

  Restart `postgresql` as root user.

```
# Run on postgres node as root user
systemctl restart postgresql
```

  Grant the `CREATEDB` privilege to the zulip user

```
# Run on postgres node
su - postgres
psql
ALTER ROLE zulip CREATEDB;
```

- As root, restore the backup:

```
#Run On Chat server node
/home/zulip/deployments/current/scripts/setup/restore-backup --keep-settings /path/to/backup
```

- After successful restoration of Chat server and PostgreSQL server, remove the `CREATEDB` privilege for the Zulip user:

```
# Run on postgres node
su - postgres
```

```
psql
ALTER ROLE zulip NOCREATEDB;
```

Note: `CREATEDB` privilege for the Zulip user should be revoked after restore is completed successfully.

Restoration on combined Chat setup (Database and Zimbra Chat server on single node):

In case of combined setup, restore the backup as:

- As root, restore the backup:

```
(Run as root on chat server)
$ /home/zulip/deployments/current/scripts/setup/restore-backup /path/to/backup.tar.gz
```

- After successful restoration, update the following details in `/etc/zulip/settings.py`

```
EXTERNAL_HOST = "new-zulip-server-hostname"
```

- Restart the Chat server

```
(Run as root on chat server)
$ /home/zulip/deployments/current/scripts/restart-server
```

On Zimbra mailbox server, updated the `zimbraChatBaseHost` with the new Chat server hostname. Update it for domains or at global config level.

```
(Run as zimbra on a Zimbra mailstore server)
$ zmprov mcf zimbraChatBaseHost <new-chat-server-hostname>
$ zmprov fc -a all
```

You can confirm that the restored Chat server can search an account on the LDAP. Make sure that both `full_name` and `email` are returned.

```
(Run as zulip on chat server)
$ /home/zulip/deployments/current/manage.py query_ldap <account-email>
...
full_name: user1
email: <account-email>
```

For more details, please refer to the following documentation:

Backups, export and import — Zulip 9.2 documentation

Important Points to note regarding Zimbra Chat Backup and Zimbra Collaboration Suite (ZCS) Backup and Restore:

1. There is no account level backup and restore on Zimbra Chat server.
2. There is no built-in automatic backup schedule for the Zimbra Chat server but can be configured using crontab.
3. Zimbra Chat server backup is not integrated with ZCS's backup and restore. Zimbra Chat backup and restore is managed separately.
4. Zimbra Chat Server logs are not backed up. Chat administrator should create backup of server log files manually if required.
5. Restoring a Zimbra Chat server from its backup does not create accounts on ZCS servers. If accounts are required to be restored on ZCS server, ZCS's restore should be executed first.

Troubleshooting Guide

1. LDAP Connection Issues

1.1. Chat server cannot access LDAP server or search a user

If you see `Can't contact LDAP server` error message at the following command execution,

```
$ /home/zulip/deployments/current/manage.py query_ldap EMAIL_ADDRESS
ldap.SERVER_DOWN: {'result': -1, 'desc': "Can't contact LDAP server", 'errno': 11, 'ctrls': [], 'info':
'Resource temporarily unavailable'}
```

check the following items:

- ldap hostname can be resolved on the Chat server
- ldap server allows 389 port access from the Chat server
- `AUTH_LDAP_SERVER_URI` in `/etc/zulip/settings.py` specifies a correct ldap hostname (and port). For example, `AUTH_LDAP_SERVER_URI = "ldap://ldap.your.domain"` or `AUTH_LDAP_SERVER_URI = "ldap://ldap.your.domain:389"`. Default port 389 can be omitted.

1.2. TLS Connection Issues

If you see `Connect error` at `Initiating TLS` like

```
$ /home/zulip/deployments/current/manage.py query_ldap EMAIL_ADDRESS
2025-03-28 07:29:10.948 DEBG [django_auth_ldap] Binding as uid=zimbra,cn=admins,cn=zimbra
2025-03-28 07:29:10.949 DEBG [django_auth_ldap] Initiating TLS
...
ldap.CONNECT_ERROR: {'result': -11, 'desc': 'Connect error', 'ctrls': [], 'info': '(unknown error code)'}
```

you need to verify a valid SSL certificate has been deployed on ldap server.

1.3. Invalid Credentials

If you see `ldap.INVALID_CREDENTIALS` like

```
$ /home/zulip/deployments/current/manage.py query_ldap EMAIL_ADDRESS
ldap.INVALID_CREDENTIALS: {'msgtype': 97, 'msgid': 2, 'result': 49, 'desc': 'Invalid credentials', 'ctrls': []}
```

check the following items:

- `AUTH_LDAP_BIND_DN` in `/etc/zulip/settings.py` specifies a correct value.
- `auth_ldap_bind_password` in `/etc/zulip/zulip-secrets.conf` specifies a correct password for the `AUTH_LDAP_BIND_DN`

1.4. Successful User Search

`full_name` and `email` should be returned when user search works correctly.

```
$ /home/zulip/deployments/current/manage.py query_ldap user1@mydomain.com
...
full_name: user1
email: user1@mydomain.com
```

You need to restart Chat server application to apply changes in `/etc/zulip/settings.py` and `/etc/zulip/zulip-secrets.conf`.

```
/home/zulip/deployments/current/scripts/restart-server
```

2. Realm Creation Issues

2.1. A realm is not created on Chat server

If you created a domain with `zimbraFeatureZulipChatEnabled TRUE` or ran `CreateChatRealmRequest` but a realm is not created on a Chat server, check the following items:

- Chat server hostname can be resolved on mailstore servers
- the Chat server allows 443 port access from mailstore servers
- `zimbraChatBaseHost` and `zimbraChatJwtSecret` have been configured on the domain. If each attribute has been configured in globalConfig and not been configured in domain, the value in globalConfig is applied automatically.
- `zimbraChatBaseHost` has a hostname of a Chat server
- `zimbraChatJwtSecret` has a secret key, which has been set in `/etc/zulip/zulip-secrets.conf` as `zimbra_jwt_auth_key` on the Chat server
- A valid SSL certificate has been deployed on the Chat server

3. Chat Availability Issues

3.1. A realm has been created on Chat server, but Chat is unavailable on Modern Web App

If a realm has been created on a Chat server but chat feature is unavailable in Modern Web App, check the following items:

3.1.1. Zimbra license

- a valid license is installed on Zimbra
- the license is activated
- the license is not expired
- the following feature(s) is licensed
  - `BOCAL (BasicOneToOneChatAccountsLimit)` feature enabled for Basic one to one chat
  - `CAL (ChatAccountsLimit)` feature enabled for Advanced Chat
- you can see license status by executing `zmlicense -p`

3.1.2. Other configuration issues

- `zimbra-zimlet-chat` zimlet has been deployed and enabled on an account
- a hostname of a realm url is resolvable on zimbra proxy and mailstore servers
- a valid SSL certificate for a hostname of a realm url has been deployed on the Chat server
- `zmproxyconfgen` and `zmproxyctl reload` have been executed on all proxy servers
- `zimbraFeatureZulipChatEnabled` is enabled on a domain
- Basic Chat
  - `zimbraFeatureZulipChatEnabled` and `zimbraFeatureBasicOneToOneChatEnabled` are enabled on an account
- Advanced Chat
  - `zimbraFeatureZulipChatEnabled` and `zimbraFeatureAdvancedChatEnabled` are enabled on an account
- If there are multiple mailstore servers, run `zmprov fc -a all`
- `CSRF_TRUSTED_ORIGINS` in `/etc/zulip/settings.py` on a Chat server specifies a URL of Zimbra web app

Note: Chat server application needs to be restarted when it is modified.

4. Account Provisioning Issues

4.1. I ran provision all accounts, but some accounts are not created on Chat server

Mailstore server does not ask Chat server to create an account if any of the following conditions are met:

- `zimbraAccountStatus` is not `active` on an account
- `zimbraIsSystemAccount` is `TRUE` on an account
- `zimbraFeatureZulipChatEnabled` is not `TRUE` on a domain or an account

4.2. I saw HTTP response status 504 error when I run provisioning chat accounts.

If you run "Provision all accounts" on Admin Console, you may see the following error message in a dialog.

```
Server error encountered
-------------------------
Error code: CSFE_SVC_ERROR
Method: ProvChatAccountsRequest
Details: HTTP response status 504
```

It occurs when it is taking a long time to provision accounts. A http connection between a mailstore server and a Chat server is closed due to timeout. However, the provisioning process may still be running on the Chat server. If `creating user` logs are being added to `/var/log/zulip/server.log`, the process is running. For example,

```
2025-04-04 02:08:20.833 INFO [] creating user: user925@mydomain.com
2025-04-04 02:08:20.883 INFO [] creating user: user926@mydomain.com
...
```

You don't need to (or should not) rerun "Provision all accounts". You can close the dialog on Admin Console, and just wait for the process to complete.

5. Chat Server Issues

5.1. Issue - chatlogin succeeds but register fails. Provisioning commands succeed.

Logs:

```
/var/log/zulip/errors.log

2025-04-02 05:53:42.735 WARN [zulip.queue:9800] TornadoQueueClient attempting to reconnect to RabbitMQ
2025-04-02 05:53:42.741 ERR  [pika.adapters.utils.io_services_utils:9800] Socket failed to connect: <socket.socket fd=18, family=2, type=1, proto=6, laddr=('127.0.0.1', 59468)>; error=111 (Connection refused)
2025-04-02 05:53:42.742 ERR  [pika.adapters.utils.connection_workflow:9800] TCP Connection attempt failed: ConnectionRefusedError(111, 'Connection refused'); dest=(<AddressFamily.AF_INET: 2>, <SocketKind.SOCK_STREAM: 1>, 6, '', ('127.0.0.1', 5672))
2025-04-02 05:53:42.742 ERR  [pika.adapters.utils.connection_workflow:9800] AMQPConnector - reporting failure: AMQPConnectorSocketConnectError: ConnectionRefusedError(111, 'Connection refused')
2025-04-02 05:53:42.742 ERR  [pika.adapters.utils.connection_workflow:9800] AMQP connection workflow failed: AMQPConnectionWorkflowFailed: 1 exceptions in all; last exception - AMQPConnectorSocketConnectError: ConnectionRefusedError(111, 'Connection refused'); first exception - None.
2025-04-02 05:53:42.742 ERR  [pika.adapters.utils.connection_workflow:9800] AMQPConnectionWorkflow - reporting failure: AMQPConnectionWorkflowFailed: 1 exceptions in all; last exception - AMQPConnectorSocketConnectError: ConnectionRefusedError(111, 'Connection refused'); first exception - None
2025-04-02 05:53:42.742 ERR  [pika.adapters.base_connection:9800] Full-stack connection workflow failed: AMQPConnectionWorkflowFailed: 1 exceptions in all; last exception - AMQPConnectorSocketConnectError: ConnectionRefusedError(111, 'Connection refused'); first exception - None
2025-04-02 05:53:42.742 ERR  [pika.adapters.base_connection:9800] Self-initiated stack bring-up failed: AMQPConnectionError: (AMQPConnectionWorkflowFailed: 1 exceptions in all; last exception - AMQPConnectorSocketConnectError: ConnectionRefusedError(111, 'Connection refused'); first exception - None,)
2025-04-02 05:53:42.742 WARN [zulip.queue:9800] TornadoQueueClient couldn't connect to RabbitMQ, retrying in 2 secs...
```

```
/var/log/zulip/server.log

```

```
2025-04-02 05:53:17.386 ERR  [pika.adapters.base_connection:9800] Self-initiated stack bring-up failed: AMQPConnectionWorkflowFailed: 1 exceptions in all; last exception - AMQPConnectorAMQPHandshakeError: ProbableAccessDeniedError: Client was disconnected at a connection stage indicating a probable denial of access to the specified virtual host: ('ConnectionClosedByBroker: (530) "NOT_ALLOWED - access to vhost \'/\' refused for user \'zulip\'"',); first exception - None
```

```
/var/log/rabbitmq/zulip@localhost.log

2025-04-02 05:53:15.590500+00:00 [info] <0.17977.0> accepting AMQP connection <0.17977.0> (127.0.0.1:60086 -> 127.0.0.1:5672)
2025-04-02 05:53:15.597951+00:00 [error] <0.17977.0> Error on AMQP connection <0.17977.0> (127.0.0.1:60086 -> 127.0.0.1:5672, state: starting):
2025-04-02 05:53:15.597951+00:00 [error] <0.17977.0> PLAIN login refused: user 'zulip' - invalid credentials
2025-04-02 05:53:15.603483+00:00 [info] <0.17977.0> closing AMQP connection <0.17977.0> (127.0.0.1:60086 -> 127.0.0.1:5672)
```

Root cause: Tornade cannot access rabbitmq on zulip user due to invalid credentials. Password is stored in `/etc/zulip/zulip-secrets.conf` as `rabbitmq_password`. It should match actual password stored in rabbitmq. However, if it does not match for some reason, for example, accidentally `rabbitmq_password` in `zulip-secrets.conf` is modified and saved, authentication fails.

Fix: run the following commands to recreate zulip user on rabbitmq and restart rabbitmq and Chat server application.

```
(Run as root)
# /home/zulip/deployments/current/scripts/setup/configure-rabbitmq
Deleting user "zulip" ...
Deleting user "zulip" ...
Deleting user "guest" ...
Adding user "zulip" ...
Done. Don't forget to grant the user permissions to some virtual hosts! See 'rabbitmqctl help set_permissions' to learn more.
Setting tags for user "zulip" to [administrator] ...
Setting permissions for user "zulip" in vhost "/" ...

# systemctl restart rabbitmq-server.service
# systemctl status rabbitmq-server.service
# su - zulip
$ /home/zulip/deployments/current/scripts/restart-server

```

6. Chat Scaling Issues

6.1. Tornado Sharding beyond 10 processes

Tornado sharding scales properly till 9 processes. Scaling tornado shards to 10 or more than 10 processes may fail with exception `TornadoQueueClient couldn't connect to RabbitMQ` in zulip's `/var/log/zulip/server.log`.

The reason being port 9810 was already in use by smokescreen.

Please follow below steps to resolve this issue:

- Perform all below operations with `zulip` user
- Stop the smokescreen using `supervisorctl stop smokescreen`
- Start the chat services with `zulip` user using `/home/zulip/deployments/current/scripts/start-server`
- Check the status if all services are up using `supervisorctl status`
- If you find any services in stopped state, please try bringing them up individually. For example:
  - If `zulip-django` is not up, use `supervisorctl start zulip-django` to bring it up

- If `zulip-tornado:zulip-tornado-port-9810` is not up, use `supervisorctl start zulip-tornado:zulip-tornado-port-9810` to bring it up.

7. Zimbra Server Issues

7.1. Issue - Proxy restart fails with “could not build map_hash” error

Logs:

```
zimbra@proxy:~$ zmproxyctl restart
Stopping proxy...nginx: [emerg] could not build map_hash, you should increase map_hash_bucket_size: 64 failed.
```

Root cause: The hash created based on `zimbraChatBaseHost` and `zimbraZulipChatDomainId` exceeds 64 length

Fix: Increase the localconfig value

```
zmlocalconfig -e proxy_web_map_hash_bucket_size=128
```

7.2. Issue - Proxy restart fails with “invalid number of arguments in map_hash_bucket_size directive” error

Logs:

```
zimbra@proxy:~$ zmproxyctl restart
Starting proxy...nginx: [emerg] invalid number of arguments in "map_hash_bucket_size" directive in
/opt/zimbra/conf/nginx/includes/nginx.conf.web:16
```

Root cause: `map_hash_bucket_size` is defined to refer the localconfig `proxy_web_map_hash_bucket_size`.

```
cat /opt/zimbra/conf/nginx/templates/nginx.conf.web.template
...
map_hash_bucket_size ${web.map_hash_bucket_size};
```

The value can be modified and applied by executing `zmproxyctl restart` as described in Section 6.1.
However, there is a case that it is not referred correctly and then `map_hash_bucket_size` becomes empty for some reason.

```
cat /opt/zimbra/conf/nginx/includes/nginx.conf.web
...
map_hash_bucket_size ;
```

Fix: Run `zmcontrol restart` on the proxy server instead of `zmproxyctl restart`.

7.3. Issue - Proxy restart fails with “host not found in upstream” error

Logs:

```
zimbra@proxy:~$ zmproxyctl reload
Reloading proxy...nginx: [emerg] host not found in upstream "domain1examplecom.chat1.mydomain.com:443" in
/opt/zimbra/conf/nginx/includes/nginx.conf.chat.upstream:66
```

Root cause: The hostname (FQDN) of Chat server or a realm URL for a domain cannot be resolved on the proxy server.

Fix:

- Check that the hostname has been registered in DNS. If it does not exist, register the hostname in DNS.
- Refresh DNS cache on the proxy server
- It may take some time to reflect a change on DNS. If it needs be fixed immediately, you can add an entry to `/etc/hosts` on the proxy server as a workaround. Note that it needs to be applied on all proxy servers.

7.4. Issue - Hash not created in chat upstream file for a specific domain

Issue: an upstream for a domain is not generated in `/opt/zimbra/conf/nginx/includes/nginx.conf.chat.upstream`

Root cause: either `zimbraChatBaseHost` or `zimbraZulipChatDomainId` is missing for the domain due to which server hash is not generated.

Fix: Update `zimbraChatBaseHost` and `zimbraZulipChatDomainId` followed by `/opt/zimbra/libexec/zmproxyconfgen` command

7.5. Issue - CSRF token mismatch

Logs: CSRF token was invalid, rechecking with account object from LDAP.

Root cause: CSRF token configured on Zimbra and Chat server are not matching.

Fix: Match the CSRF

7.6. Issue - Unknown host exception

Logs: `java.net.UnknownHostException: domain1examplecom.chat1.mydomain.com: Name or service not known`

Root cause: The Url in the logs is not resolvable

Fix: Register the FQDN in DNS. As a temporary solution, update the required file (`/etc/hosts`) to make the URL resolvable.

7.7. Issue - Missing JWT key

Logs: `SoapEngine - handler exception java.lang.IllegalArgumentException: Empty key`

Root cause: JWT is not configured on Zimbra server

Fix: Update `zimbraChatJwtSecret` on Zimbra server

7.8. Issue - New account cannot be searched on Basic Chat

Account search on Basic Chat is based on gal search. If Basic Chat search does not return a newly added account, GAL may not be configured correctly on the domain. You need to check the following items:

- If GAL mode is shown as “Internal” on Admin Console > domain setting > GAL, make sure zimbraGalMode is set to zimbra on CLI; `zmprov gd DOMAIN zimbraGalMode`
- If GAL mode is “Internal” or “both”, make sure a galsync account has been configured.
- If there are multiple mailstore servers on your environment, make sure a galsync account has been configured on each mailstore server

If account still cannot be searched, run the following command for all galsync accounts on the domain.

```
zmgsautil forceSync -a [GAL_SYNC_ACCOUNT] -n [DATA_SOURCE_NAME]
```

You can find `zimbraDataSourceName` for the `DATA_SOURCE_NAME` by running `zmprov gds [GAL_SYNC_ACCOUNT]` (getDataSources) command.

FAQs

General Questions

1. **Is there any tool for Migration of chats and groups from Connect to the new Chat?**
   Migration is not supported from older chat solutions, and there is no plan to provide a migration tool in future.

---

5. **Can multiple Zimbra servers connect to one Chat server?**
   No. Multiple authentication from two Zimbra servers to a single Chat server is not supported.

---

6. **When is a Chat server restart required?**
   On configuration changes or if a component is down. For more details refer: Troubleshooting Guide.

---

7. **Is uninstalling the Chat server supported?**
   Uninstallation is not supported by the underlying open source chat platform and hence not supported in Zimbra Chat.

---

8. **Is chat supported in Classic UI?**
   Yes, it is targeted for Zimbra Daffodil 10.1.8 release.

---

9. **Are there benchmarks or required resources for Chat servers?**
   Yes, For more details refer: Sizing & Scaling Guide section.

Note: Performance of any chat solution heavily depends on the usage pattern of the users, so it may not match exactly with the benchmarks provided here.

---

10. **Can chat history be exported, erased, or edited?**
    Yes, chat history can be exported.

---

11. **Is file sharing supported in Advanced Chat? Can it be restricted?**
    Yes. Administrators can configure restrictions in `settings.py`:

```
DISALLOWED_FILE_EXTENSIONS = ["exe", "bat", "sh", "cmd"]
MAX_FILE_UPLOAD_SIZE = 10
UPLOADED_FILES_EXPIRED_AFTER_DAYS = 30
```

---

13. Is there a separate mobile application or desktop application available for Zimbra Chat?
Not yet but it is under consideration for the future roadmap.

---

15. Does Chat support external authentication if Zimbra does?
Yes. The chat installer will prompt for external auth details.

---

16. Does Chat work with LDAPS (port 636)?
Yes. It can be configured using chat installer

Admin-Specific Questions

17. Can a domain admin manage Chat settings or manage chat accounts?

Yes, a domain admin can manage Chat settings and manage chat accounts on Admin Console.

A global admin can configure grants on a domain admin as follows:

- Create a domain admin account
- Set admin views to see Chat settings in account, COS and domain:
  - Account List View
  - Class Of Service List View
  - Domain List View
- Add the following grants to the domain admin:

```
adminConsoleAccountRights        target type: domain,  target name: DOMAIN_NAME
domainAdminCosRights             target type: cos,     target name: COS_NAME
adminConsoleDomainRights         target type: domain,  target name: DOMAIN_NAME
listZimlet                       target type: zimlet,  target name: zimbra-zimlet-admin-chat
getZimlet                        target type: zimlet,  target name: zimbra-zimlet-admin-chat
```

Account:

```
target type: domain, target name: DOMAIN_NAME
set.account.zimbraFeatureZulipChatEnabled
set.account.zimbraFeatureBasicOneToOneChatEnabled
set.account.zimbraFeatureAdvancedChatEnabled
```

COS:

```
target type: cos, target name: COS_NAME
set.cos.zimbraFeatureZulipChatEnabled
set.cos.zimbraFeatureBasicOneToOneChatEnabled
set.cos.zimbraFeatureAdvancedChatEnabled
```

Domain:

```
target type: domain, target name: DOMAIN_NAME
set.domain.zimbraZulipChatDomainId
set.domain.zimbraFeatureZulipChatEnabled
set.domain.zimbraFeatureBasicOneToOneChatEnabled
set.domain.zimbraFeatureAdvancedChatEnabled
```

Limitations:
Regardless of whether the admin manages multiple domains on the Zimbra non-chat environment, the domain admin's permissions allow them to manage Chat accounts (get, provision, activate, and deactivate) only within their own domain.

Example:
You belong to `domainA` (i.e. your email address is `domain_admin@domainA`), and you have admin privilege to manage `domainA` and `domainB` on the Zimbra mailbox. In such case, you can manage the Chat accounts on `domainA`, but cannot on `domainB`. If you want to manage the chat accounts on `domainB`, you need to have an account of `domainB` with a domain admin rights.

---

17. Can a domain administrator create a realm?

No, only a global administrator can create a realm on Chat server.

---

18. Can I access as Owner or Administrator of a realm on Advanced Chat?

When you log in to an admin account and go to Advanced Chat, Owner or Administrator role is applied under the following conditions:

- Admin account with `zimbraIsDomainAdminAccount = TRUE`: Owner of a realm

- Admin account with `zimbraIsAdminAccount` or `zimbraIsDelegatedAdminAccount = TRUE`: Administrator of a realm
- Non-admin account: Member of a realm

Roles:

- Owner can manage users, public channels and organization (i.e. domain or a realm) settings. Owners can do anything that an organization administrator can do.
- Administrator can manage users, public channels and organization settings. Administrator cannot make someone an owner, or change an existing owner's role.
- Member has access to all public channels.

---

19. How can I restart Chat server process?

Run as root on Chat server:

```
# systemctl restart nginx.service
# su - zulip
$ /home/zulip/deployments/current/scripts/restart-server
```

---

20. Can I see existing realms on Chat server?

Run as zulip on Chat server:

```
/home/zulip/deployments/current/manage.py list_realms
```

---

21. Can I remove an account from Chat server completely?

When an account is removed from Zimbra, its chat account is deactivated automatically. To remove it completely:

Run as zulip on Chat server:

```
# Dry run
/home/zulip/deployments/current/manage.py delete_user -u EMAIL_ADDRESS

# Delete user
/home/zulip/deployments/current/manage.py delete_user -u EMAIL_ADDRESS -f
```

Warning: All chat conversation messages sent by the account will no longer be available.

---

22. Can I provision and manage accounts on Chat server using CLI?

Admin Console provides the following functions in domain settings:

- Get all accounts in a realm on a Chat server
- Provision all accounts in a domain to a Chat server
- Provision a single account in a domain to a Chat server
- Activate an account on a Chat server
- Deactivate an account on a Chat server

You can do the same using CLI. Each command needs to run as `zimbra` user on a mailstore server. See SOAP API Reference Manual for more details.

Get all accounts in a realm on a Chat server

```
$ zmsoap -z -type admin GetAllChatAccountsRequest / domain="domain1.example.com" @by="name"
<GetAllChatAccountsResponse xmlns="urn:zimbraAdmin">
  <account>
    <a n="is_active">false</a>
    <a n="user_id">46</a>
    <a n="email">deleteduser46@domain1examplecom.chat1.mydomain.com</a>
  </account>
  <account>
    <a n="is_active">true</a>
    <a n="user_id">436</a>
    <a n="email">user1@domain1.example.com</a>
  </account>
</GetAllChatAccountsResponse>
```

If a user has been deleted (not deactivated), `localpart` *has* `deleteduser` *prefix and domain reflects realm URL.*

Provision all accounts in a domain to a Chat server

```
$ zmsoap -z -type admin ProvChatAccountsRequest / domain="domain1.example.com" @by="name"
<ProvChatAccountsResponse xmlns="urn:zimbraAdmin">
  <numSucceeded>10</numSucceeded>
  <numFailed>0</numFailed>
</ProvChatAccountsResponse>
```

Provision an account in a domain to a Chat server

```
$ zmsoap -z -type admin ProvChatAccountsRequest / domain="domain1.example.com" @by="name" ../
account="user2@domain1.example.com" @by="name"
<ProvChatAccountsResponse xmlns="urn:zimbraAdmin">
  <numSucceeded>1</numSucceeded>
  <numFailed>0</numFailed>
</ProvChatAccountsResponse>
```

Get an account on a Chat server

```
$ zmsoap -z -type admin ManageChatAccountRequest @action="get" / account="user1@domain1.example.com" @by="name"
<ManageChatAccountResponse xmlns="urn:zimbraAdmin">
  <a n="is_active">true</a>
  <a n="user_id">436</a>
  <a n="email">user1@domain1.example.com</a>
</ManageChatAccountResponse>
```

Deactivate an account on a Chat server

```
$ zmsoap -z -type admin ManageChatAccountRequest @action="deactivate" / account="user1@domain1.example.com"
@by="name"
<ManageChatAccountResponse xmlns="urn:zimbraAdmin">
  <a n="is_active">false</a>
  <a n="user_id">436</a>
  <a n="email">user1@domain1.example.com</a>
</ManageChatAccountResponse>
```

Activate an account on a Chat server

```
$ zmsoap -z -type admin ManageChatAccountRequest @action="activate" / account="user1@domain1.example.com"
@by="name"
<ManageChatAccountResponse xmlns="urn:zimbraAdmin">
  <a n="is_active">true</a>
  <a n="user_id">436</a>
```

```
  <a n="email">user1@domain1.example.com</a>
</ManageChatAccountResponse>
```

---

23. Can I deploy Chat server without a valid SSL certificate for testing?

Yes, but not recommended for production use.

If no valid SSL certificate or Let's Encrypt is available:

- Install Chat server with `self-signed` SSL type.
- Set `ssl_allow_untrusted_certs=true` on all mailstore servers:

Run as zimbra on mailstore servers:

```
$ zmlocalconfig -e ssl_allow_untrusted_certs=true
$ zmmailboxdctl restart
```

Warning: Affects all outbound communication from the mailstore server and poses a security risk. Do not use in production.