F13 Hobbies Community

Team :- Sanjay C Hiremath, Prajna Dutt, Mokshank Wankhede, Stuti Gupta, Akhila

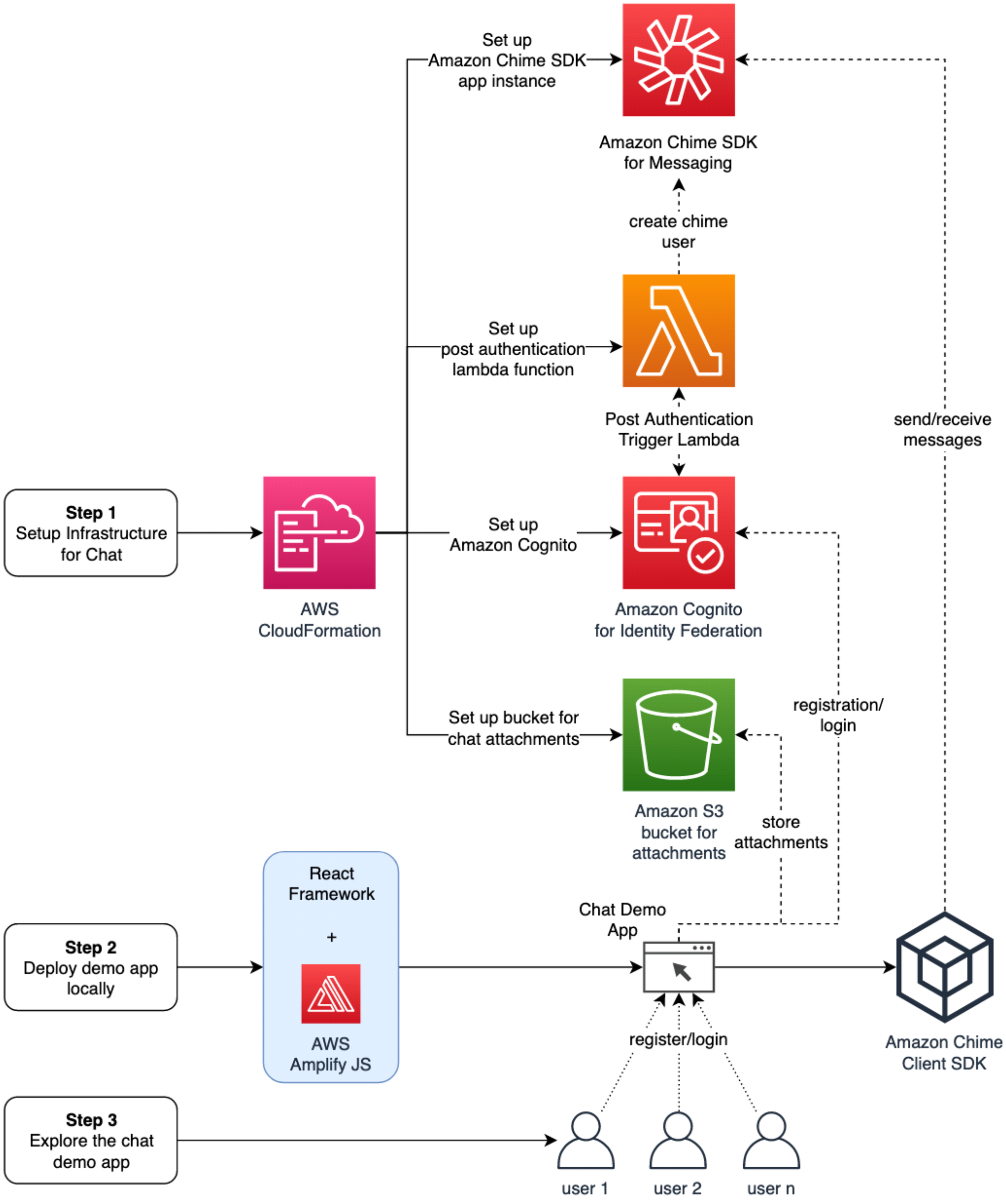
Introduction

Our project is dedicated to the development of a state-of-the-art chat application, seamlessly integrating AWS services to ensure scalability, security, and an exceptional user experience. This report outlines our architecture, highlighting key components like AWS CloudFormation, Lambda functions, Amazon Cognito, S3 buckets, React framework, and the Amazon Chime SDK.

AWS CloudFormation forms the foundation, enabling the creation and management of AWS resources in a reproducible manner. Amazon Cognito handles authentication and user identity, while Lambda functions enhance application flexibility. Amazon S3 ensures secure storage of chat attachments, and the React framework, coupled with AWS Amplify JS, powers our front-end.

Real-time messaging is achieved through the Amazon Chime SDK, offering low-latency communication features. This report delves into the architecture's intricacies, emphasizing our commitment to scalability, security, and developer efficiency for an unparalleled chat experience.

AWS architecture for the chat application:



AWS CloudFormation

- **AWS CloudFormation**: This service is used to create and manage a collection of AWS resources by provisioning and updating them in an orderly and predictable fashion. In the context of the chat application, CloudFormation likely defines the networking resources, security policies, and other AWS service configurations necessary to support the application. It allows developers to use a template file (in YAML or JSON format) to define the “infrastructure as code,” which makes the setup reproducible and version-controllable.

Lambda Function and Amazon Cognito

- **Post Authentication Lambda Function**: After a user authenticates, this AWS Lambda function is triggered. This could be used for various tasks such as logging, custom metrics, adding the user to a database, or even invoking other AWS services.

- **Amazon Cognito**: This service provides user identity and data synchronization, enabling the application to authenticate users. It supports sign-in with social identity providers like Google, Facebook, and Amazon, as well as enterprise identity providers via SAML 2.0. It can also provide temporary AWS credentials for accessing AWS services like S3 and DynamoDB, ensuring that users can only access resources they're entitled to.

S3 Bucket and Messaging

- **Amazon S3 Bucket for Attachments**: This is used to store any files or attachments that users send in the chat. Amazon S3 provides secure, durable, and highly-scalable object storage. It's common to use S3 in conjunction with features like pre-signed URLs to securely upload and download files directly from a client application.

Front-end and Local Deployment

- **React Framework + AWS Amplify JS**: The chat application's front-end is built with the React framework, likely for its efficient update and rendering capabilities. AWS Amplify facilitates the integration of the front-end with AWS services, providing libraries and tools that enable authentication, API integration, analytics, and more.

Amazon Chime SDK for Messaging

- **Amazon Chime SDK**: This SDK is designed to enable real-time communication features within applications. For messaging, it provides the capabilities for real-time, scalable chat functionality. It manages WebSocket connections to enable real-time message delivery, presence information, and more.

Application Workflow

Infrastructure Setup for Chat

Initially, the infrastructure is set up using AWS CloudFormation. This script-driven setup allows AWS to automatically establish the required infrastructure components.

Amazon Cognito Configuration

Identity federation is managed by Amazon Cognito, which has been configured to authenticate users from various identity providers and to provide access permissions.

A Lambda function, to be triggered post-authentication, is established, likely to execute backend processes or for logging purposes after users are authenticated.

Amazon Chime SDK Setup

For messaging services, the Amazon Chime SDK has been set up. An app instance within Chime is created, along with user setup configurations, enabling the application to utilize the SDK's messaging capabilities.

Chat Attachments Bucket Setup

A dedicated Amazon S3 bucket for storing chat attachments has been established. This is the designated storage location for files exchanged through the chat service.

Local Deployment of the Demo App

The chat application, developed with the React framework and integrated with AWS Amplify, is deployed locally. This step is crucial for testing and further development of the application.

Exploration of the Chat Demo App

The chat application is then explored, with multiple users (user 1, user 2, ..., user n) interacting with the application to verify its functionality.

Integration of the Chat Demo App

The chat application integrates several components previously set up. It is designed to interact with the Amazon Chime Client SDK, which facilitates the messaging features within the application.

User Registration and Login

The registration and login process is handled, with Amazon Cognito overseeing the secure storage and retrieval of user credentials.

Messaging Functionality

With registration and login successfully completed, the messaging functionality is enabled. Users are able to send and receive messages through the chat application, a process powered by the integrated Amazon Chime SDK.

Example Workflow:

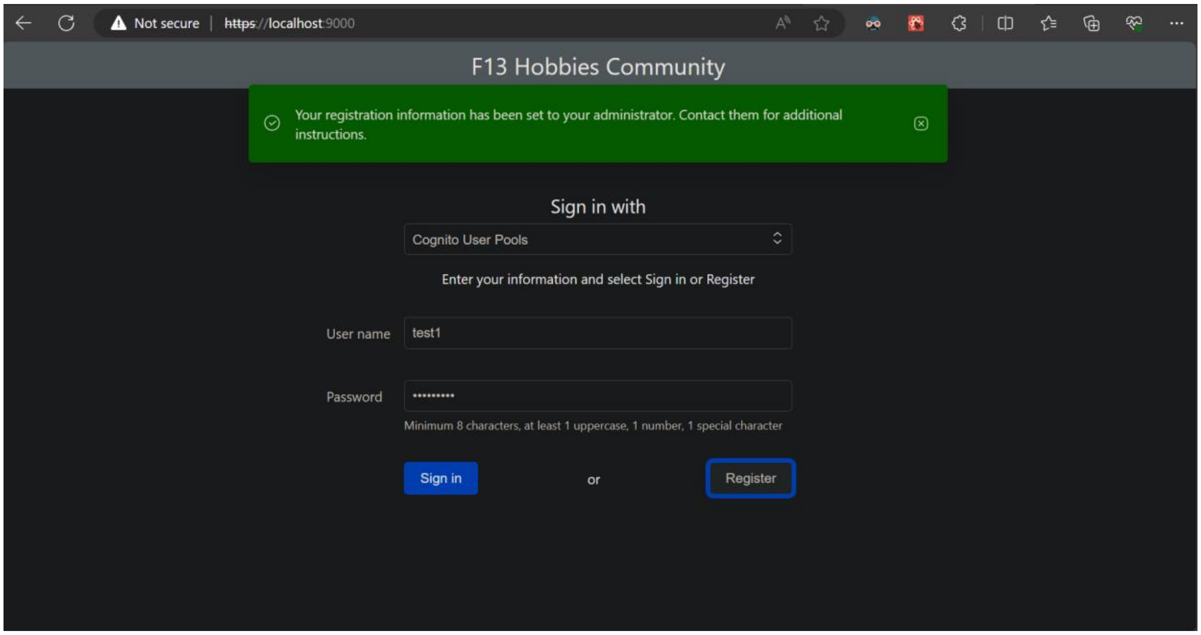
- User Interaction Initiation: A user, let's say Alice, starts a chat session via the application's frontend interface.
- Authentication Request: Alice's login attempt prompts a request to Amazon Cognito.
- Identity Verification: Amazon Cognito verifies Alice's credentials and authenticates her identity.
- Lambda Function Activation: Upon successful authentication, a post-authentication Lambda function is triggered, performing actions such as logging the event and associating Alice's session with her user profile.
- Message and Attachment Handling: Alice sends a message with an attachment to another user within the application.
- Amazon Chime SDK Messaging: The Amazon Chime SDK manages the real-time transmission of Alice's message to the intended recipient.

- S3 Bucket Storage: The attachment included by Alice is concurrently uploaded to an Amazon S3 bucket, dedicated for storing such files.
- End-to-End Integration: The entire process, from Alice's interaction to the backend operations, is streamlined by AWS services, ensuring secure and efficient communication.

Code Zip link:

https://drive.google.com/file/d/1RrXZ9pVGi79xaj35cDWKxOnDt8qBlydq/view?usp=sharing

Results:



Your registration information has been set to your administrator. Contact them for additional instructions.

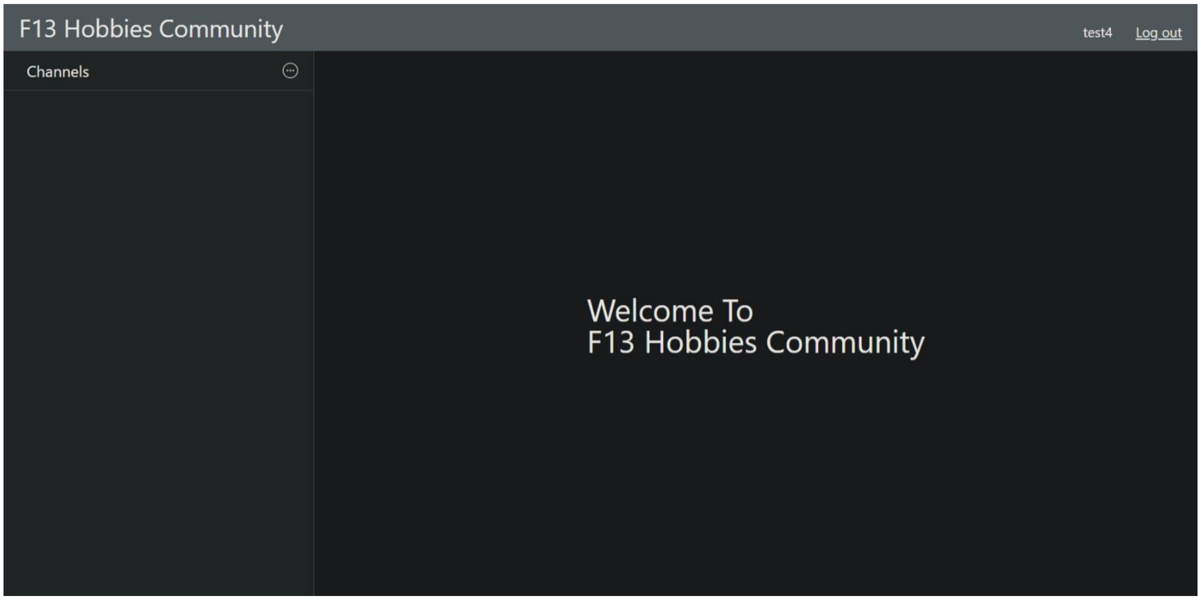
Cognito User Pools

Enter your information and select Sign in or Register

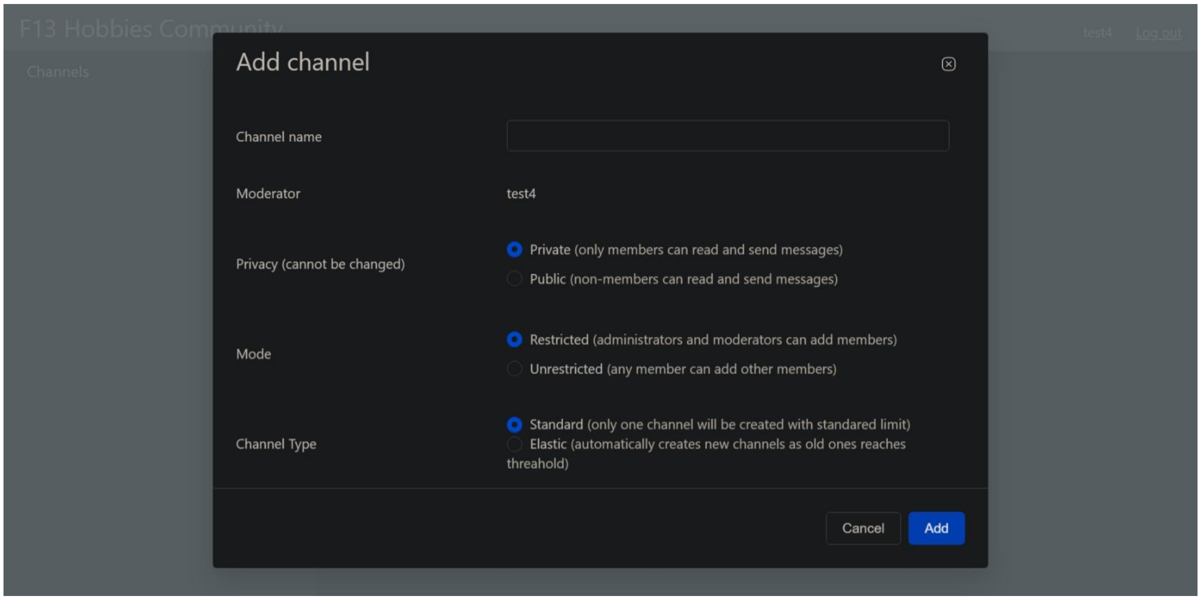
User name test1

Password •••••••••

Minimum 8 characters, at least 1 uppercase, 1 number, 1 special character



Welcome To
F13 Hobbies Community



Add channel

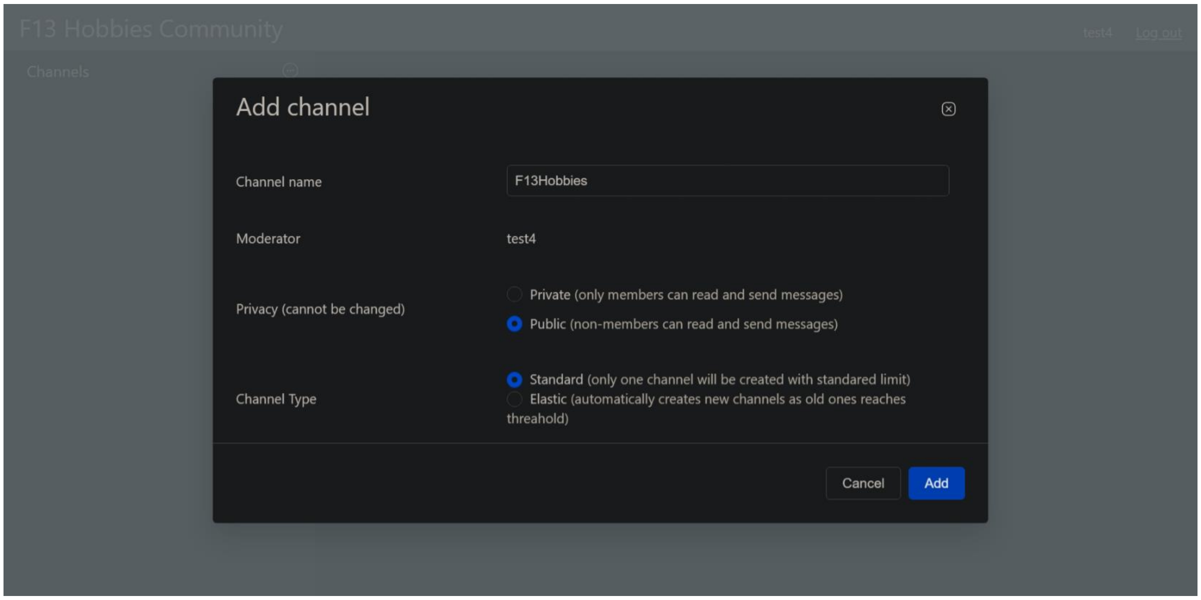
Channel name

Moderator test4

Privacy (cannot be changed)
● Private (only members can read and send messages)
○ Public (non-members can read and send messages)

Mode
● Restricted (administrators and moderators can add members)
○ Unrestricted (any member can add other members)

Channel Type
● Standard (only one channel will be created with standared limit)
○ Elastic (automatically creates new channels as old ones reaches threahold)



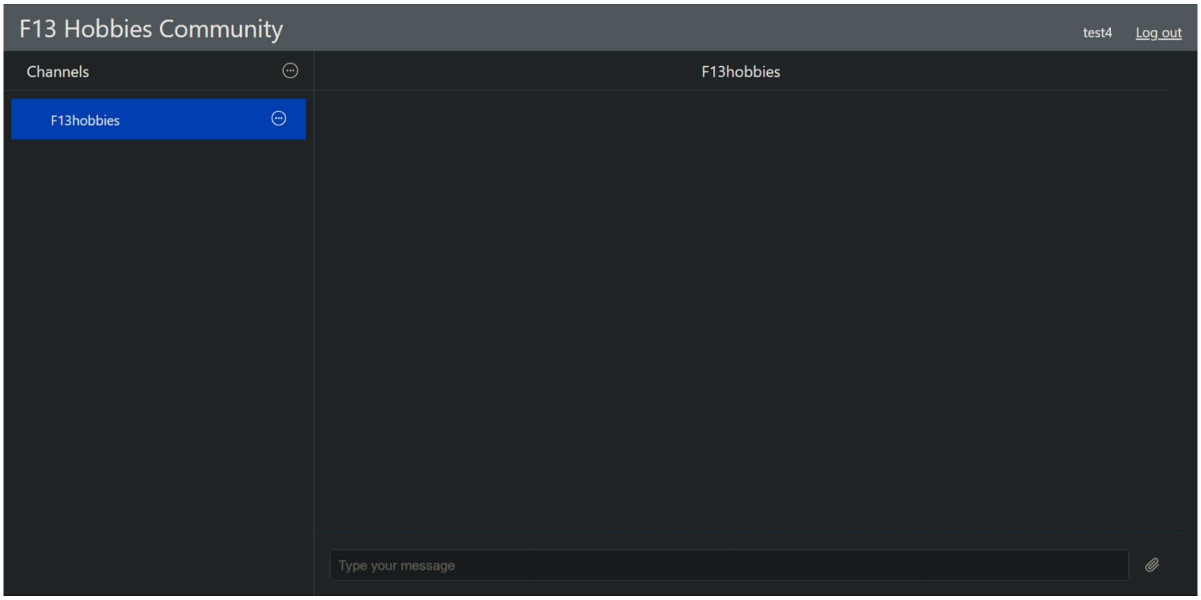
Add channel

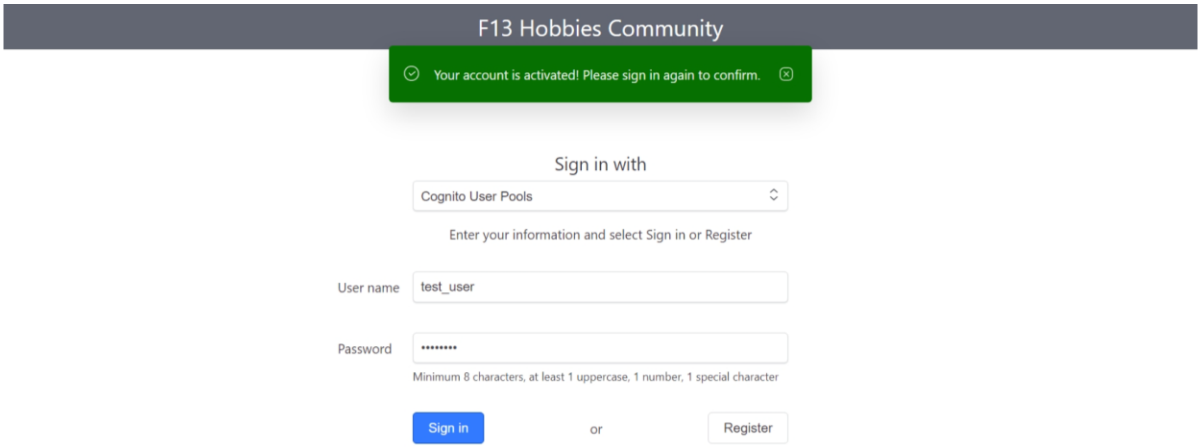
Channel name F13Hobbies

Moderator test4

Privacy (cannot be changed)
○ Private (only members can read and send messages)
● Public (non-members can read and send messages)

Channel Type
● Standard (only one channel will be created with standared limit)
○ Elastic (automatically creates new channels as old ones reaches threahold)





Your account is activated! Please sign in again to confirm.

Sign in with

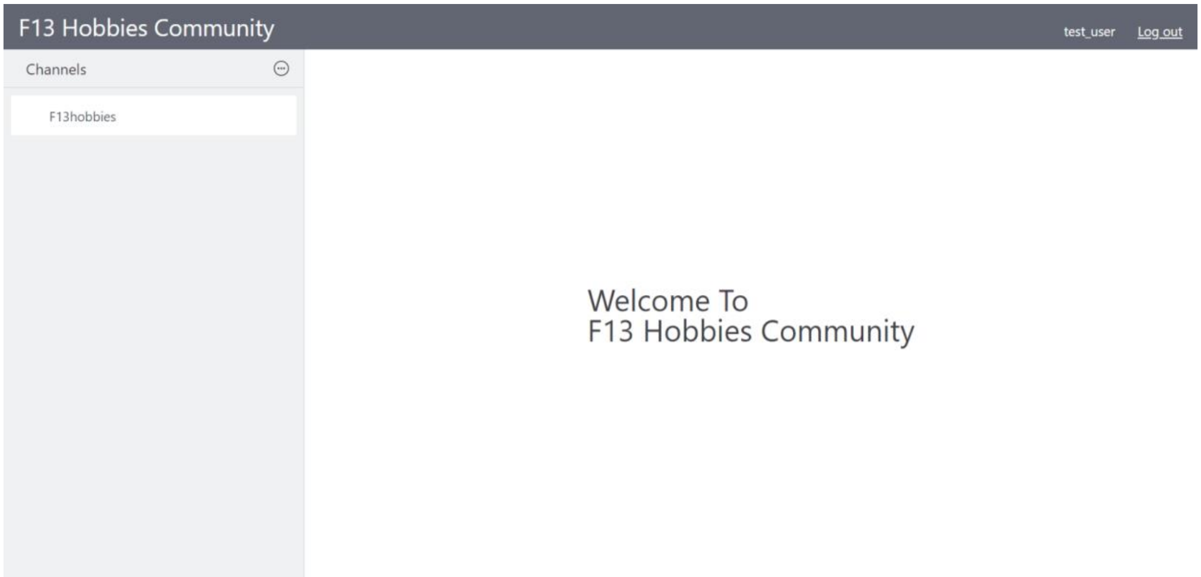
Cognito User Pools

Enter your information and select Sign in or Register

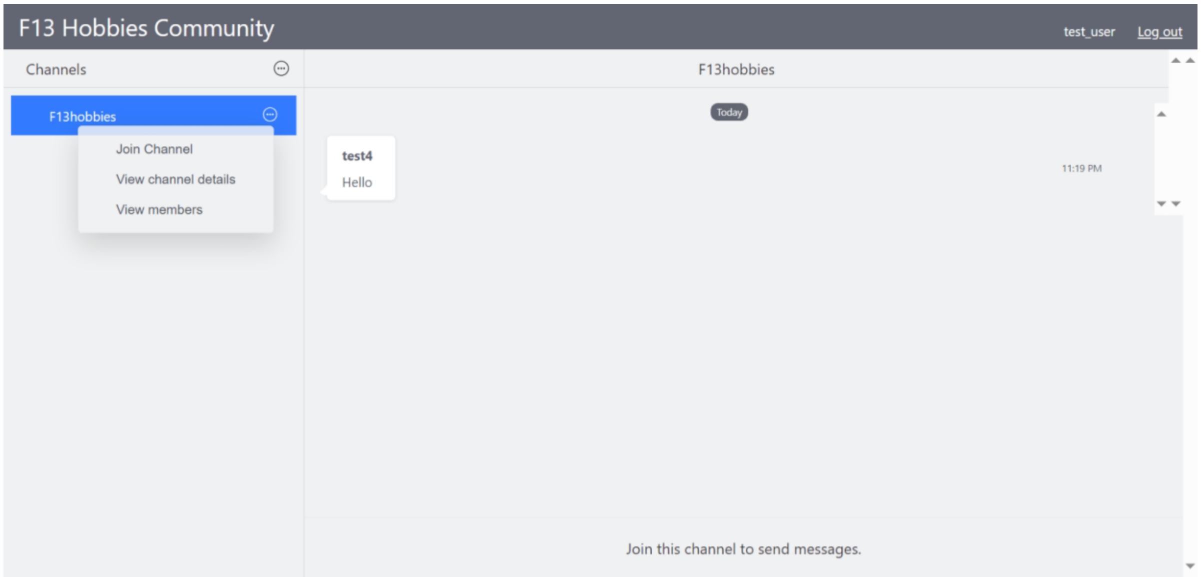
User name: test_user

Password: ••••••••

Minimum 8 characters, at least 1 uppercase, 1 number, 1 special character

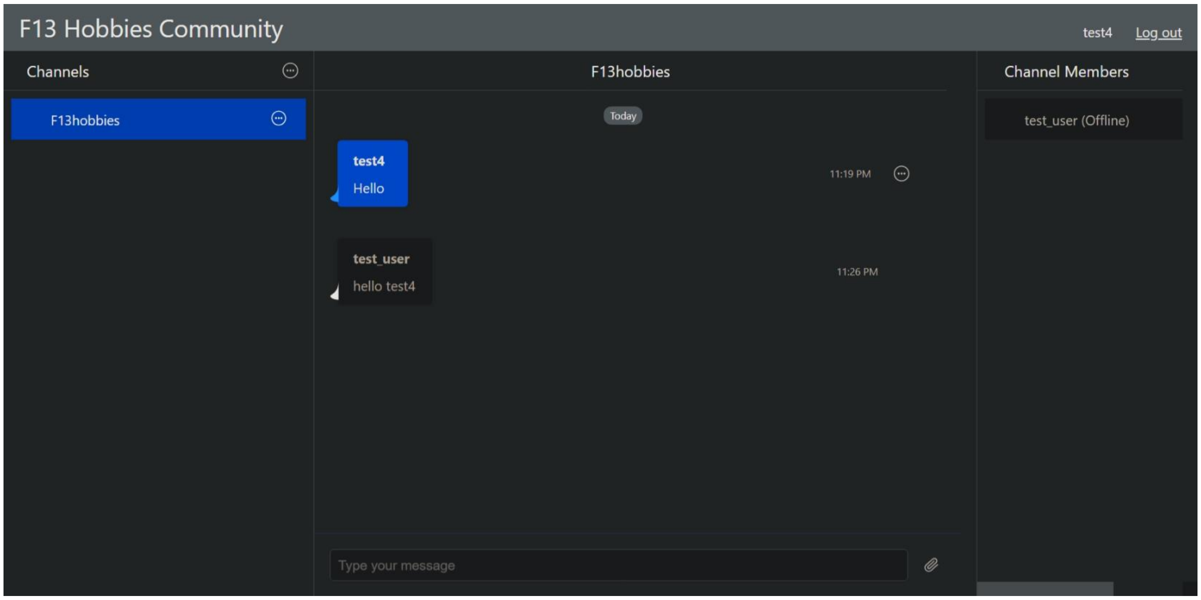


Welcome To
F13 Hobbies Community



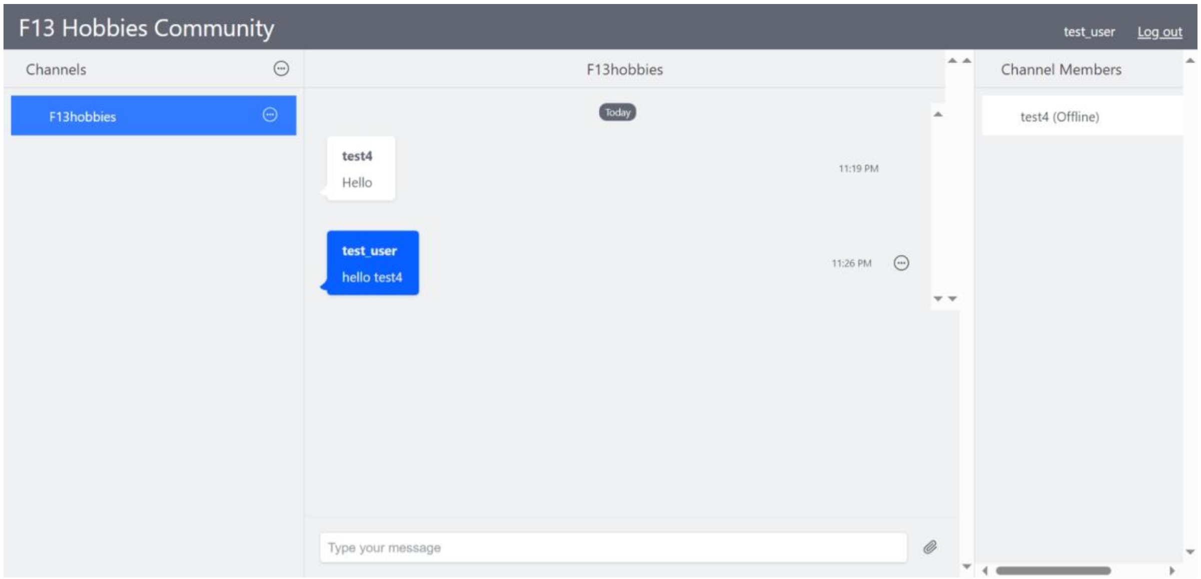
test4
Hello
11:19 PM

Join this channel to send messages.



test4
Hello
11:19 PM

test_user
hello test4
11:26 PM

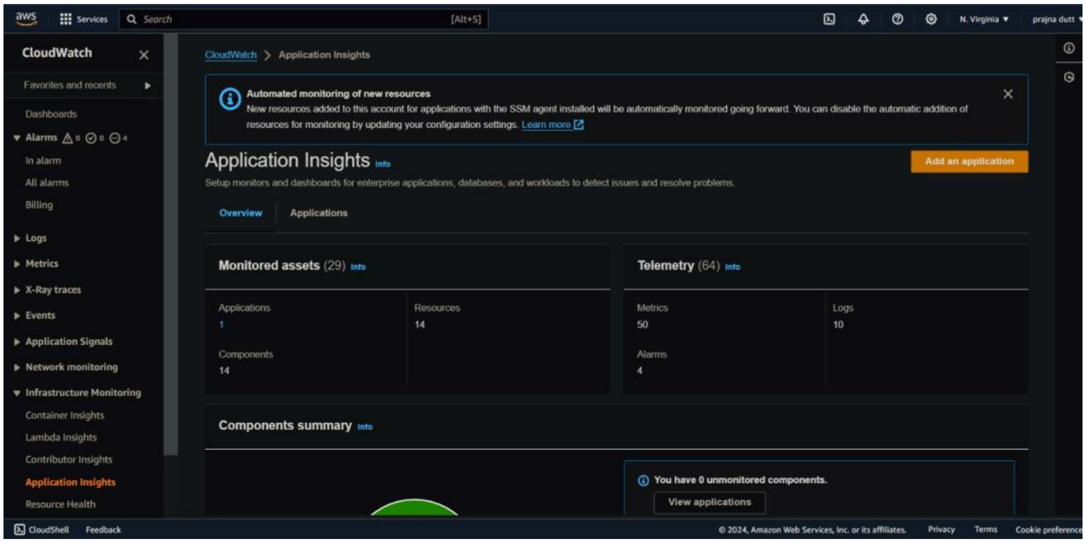


test4
Hello
11:19 PM

test_user
hello test4
11:26 PM

Amazon CloudWatch and monitoring with the Amazon Chime SDK

- Amazon CloudWatch provides metrics and logs for monitoring Amazon Chime SDK applications. It can track metrics like connection quality, bitrate, and other performance indicators.
- CloudWatch metrics are generated automatically by the Amazon Chime SDK. Developers can view them in the CloudWatch console to monitor their application.
- CloudWatch logs allow developers to send custom log events from their Amazon Chime SDK application code. These can be used for debugging issues or tracking application usage.
- The Amazon Chime SDK has built-in support for sending logs and metrics to CloudWatch. Developers can enable this in their application configuration.
- Key metrics to monitor include connection stability, audio/video quality, and meeting errors. Log events can capture application usage, errors, and customer experience.
- CloudWatch alarms can be configured to trigger notifications or auto-scaling actions if certain thresholds are crossed for critical metrics.
- CloudWatch dashboards give an at-a-glance view of multiple CloudWatch metrics for an Amazon Chime application. Developers can create customized dashboards.
- Overall, CloudWatch provides important monitoring capabilities to track the health, performance, and usage of Amazon Chime SDK applications. The metrics and logs enable developers to rapidly diagnose issues and improve the application.



Automated monitoring of new resources
New resources added to this account for applications with the SSM agent installed will be automatically monitored going forward. You can disable the automatic addition of resources for monitoring by updating your configuration settings. Learn more

Application Insights

Setup monitors and dashboards for enterprise applications, databases, and workloads to detect issues and resolve problems.

Monitored assets (29)

Applications 1
Resources 14
Components 14

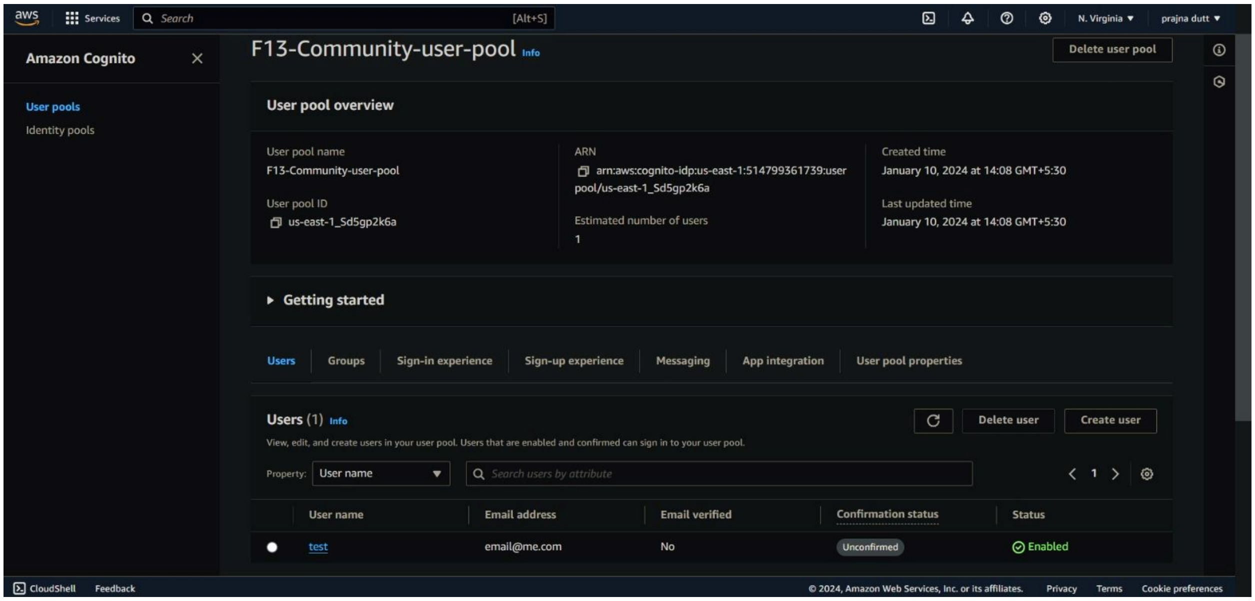
Telemetry (64)

Metrics 50
Logs 10
Alarms 4

Components summary

You have 0 unmonitored components.

© 2024, Amazon Web Services, Inc. or its affiliates.



F13-Community-user-pool Info

User pool overview

User pool name
F13-Community-user-pool

User pool ID
us-east-1_Sd5gp2k6a

ARN
arn:aws:cognito-idp:us-east-1:514799361739:userpool/us-east-1_Sd5gp2k6a

Estimated number of users
1

Created time
January 10, 2024 at 14:08 GMT+5:30

Last updated time
January 10, 2024 at 14:08 GMT+5:30

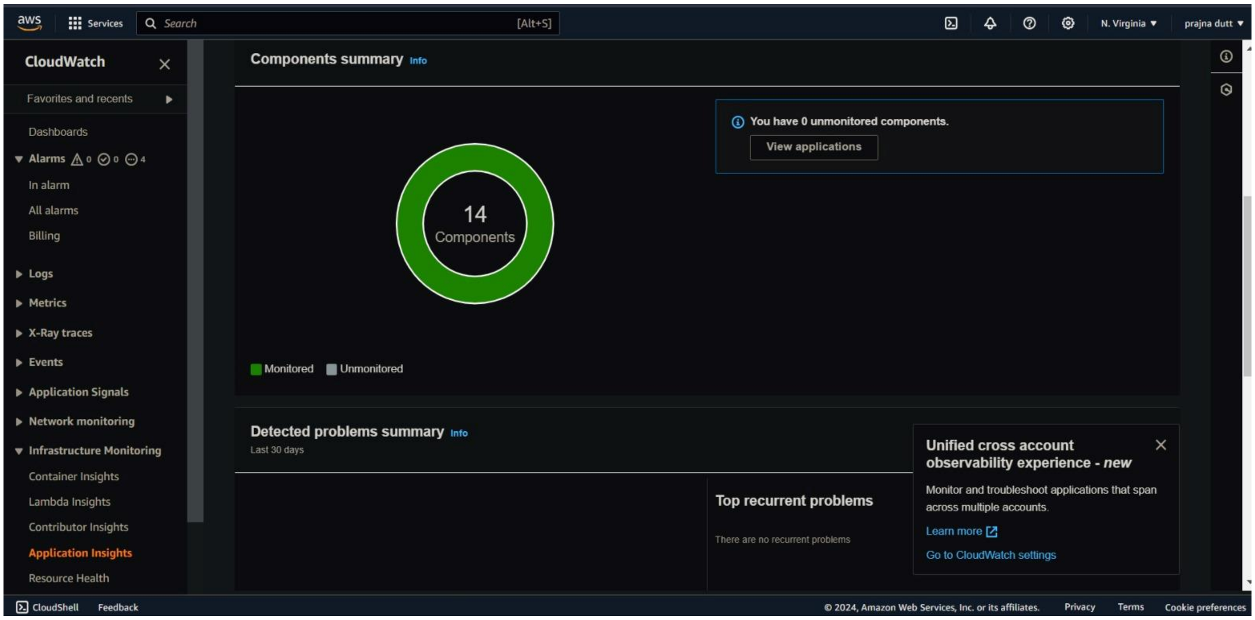
▸ Getting started

Users | Groups | Sign-in experience | Sign-up experience | Messaging | App integration | User pool properties

Users (1) Info

View, edit, and create users in your user pool. Users that are enabled and confirmed can sign in to your user pool.

	User name	Email address	Email verified	Confirmation status	Status
○	test	email@me.com	No	Unconfirmed	✓ Enabled



Components summary Info

You have 0 unmonitored components.

Detected problems summary Info

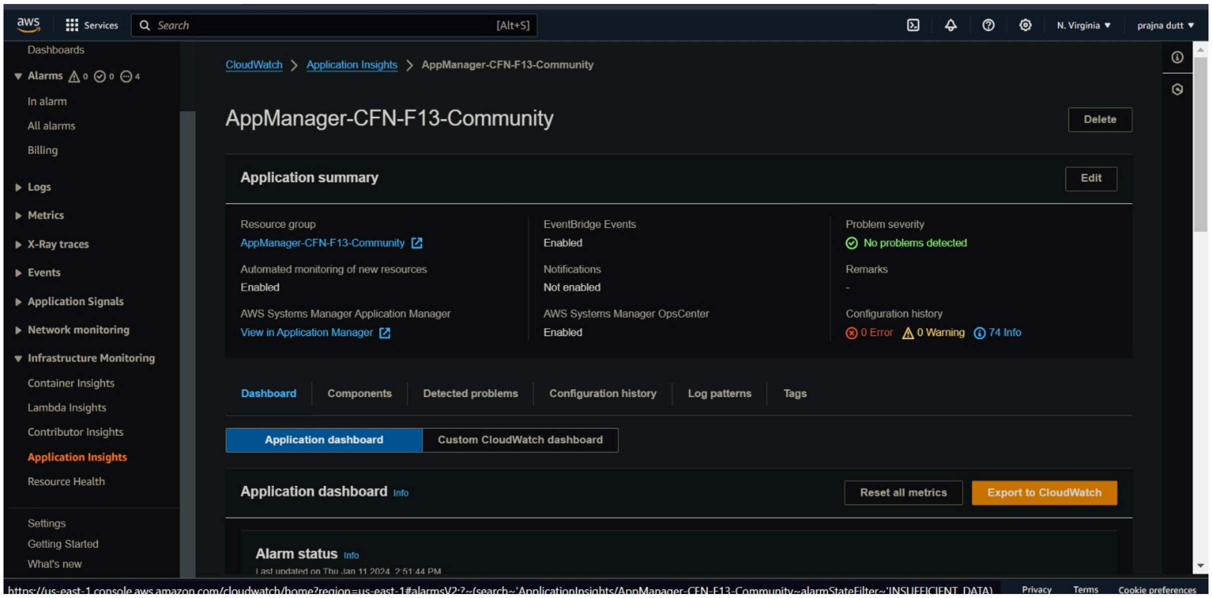
Last 30 days

Top recurrent problems

There are no recurrent problems

Unified cross account observability experience - *new*

Monitor and troubleshoot applications that span across multiple accounts.



AppManager-CFN-F13-Community

Application summary

Resource group
AppManager-CFN-F13-Community

EventBridge Events
Enabled

Problem severity
No problems detected

Automated monitoring of new resources
Enabled

Notifications
Not enabled

Remarks
-

AWS Systems Manager Application Manager
View in Application Manager

AWS Systems Manager OpsCenter
Enabled

Configuration history
0 Error 0 Warning 74 Info

Application dashboard

Alarm status
Last updated on Thu Jan 11 2024 2:51:44 PM

https://us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarmsV2:?~(search~'ApplicationInsights/AppManager-CFN-F13-Community~alarmStateFilter~'INSUFFICIENT_DATA)

Metrics

- Predefined metrics like AudioUpstreamBitrate, AudioDownstreamBitrate, VideoUpstreamBitrate, VideoDownstreamBitrate can be used to monitor real-time media quality.

- AudioPacketLossPercent and VideoPacketLossPercent track packet loss affecting call quality.

- Metrics like MeetingsCreated, MeetingsEnded, and MeetingErrors show overall usage trends.

- Custom metrics can also be defined and sent from application code to monitor business-specific data.

Logs

- Logs can be enabled through the SDK configuration to get debug information on connections, devices, and other events.

- Custom log events can be added in application code to track key user actions or application state changes.

- Log filtering and analysis features in CloudWatch help quickly search for and aggregate log data.

- Log data can be exported to S3 for longer term retention and big data analytics.

Alarms

- Alarms can trigger SNS notifications or Auto Scaling actions when a metric crosses a threshold.

- Alarms can help promptly identify and resolve issues like consistent high packet loss or meeting failures.

- An alarm could also alert when meetings created hits a scale limit, indicating time to provision more resources.

Dashboards

- Dashboards provide an intuitive visual way to monitor multiple metrics and logs.

- They can include graphs showing trends over time as well as current values of key metrics.

- Drill-down capabilities help navigate from overview to detailed metrics.

- Dashboards can be customized for different teams like developers, SREs, business analysts.

Applications

Amazon CloudWatch can be very useful for monitoring and troubleshooting Amazon Chime SDK Applications:

- Metrics for number of messages published, message delivery delays, and message errors can show overall chat activity and health

- Alarms on message publish or delivery errors can quickly notify developers of potential issues. For example, an alarm when errors cross a threshold of 5% of traffic.

- Log analysis can provide insights into user chat patterns - when are peak chat times, how many concurrent chats on average, etc.

- Metrics on message latency can help track performance of chat systems under different traffic loads. Spikes may indicate a need to scale resources.

- For community apps with many chat channels, metrics per channel help identify hot channels based on volume of messages or active users.

- Log filters for message contents can be used to analyse trends in topics, keywords, and emojis used in community chats.

- Statistics on monthly active users and messages provide insight on growth and engagement over time.

- In case of reports of offensive content, logs can be analyzed to understand spread and take mitigation steps.

- Latency and error metrics per chat user help identify bad acting accounts disrupting the system.

- For highly engaging Casts with 100,000+ live viewers, metrics help track viewer concurrency and engagement.

In conclusion, the ability to get granular real-time metrics on every aspect of a messaging system along with detailed log data enables in-depth monitoring and informed troubleshooting.