**TeamLocus Chat Project Documentation**

**Document Control**

**Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version Detail** | **Date** | **Prepared By** | **Reviewed By** |
| V1 | 27/07/2023 | Ritesh Nagdive | Usama Mashayak |
| - | - | Asha Munde | - |

Contents

[1. Introduction 4](#_Toc141371160)

[2. AWS Services Involved In TL Chat Project Architecture 5](#_Toc141371161)

[3. TL Chat Project Description 7](#_Toc141371162)

[4. Other Integrated Applications with TL Chat 8](#_Toc141371163)

# Introduction

The TeamLocus (TL) Chat Project includes a multifunctional internal communication system designed to meet the specific operational requirements of an organization. It enhances communication with features such as Instant Messaging, Secure File Transfer, and Audio and Video Calling.

Simultaneously, it also provides insightful notifications about the CI/CD pipeline stages, updates on Livekit call status, and notifications related to equipment maintenance activities in the plant, amongst other essential functions.

TeamLocus serves as a comprehensive communication and efficiency tool, perfectly suited for institutions or individual’s intent on monitoring personal productivity or fostering collaborative dynamics within team-oriented settings.

Features included in TeamLocus encompass:

* Email Capability (Communicate on Tasks)
* Team Board (enabling team members to efficiently structure their tasks whilst simultaneously permitting managers to oversee the progress of said tasks).

# AWS Services Involved In TL Chat Project Architecture

1. Elastic Compute Cloud (EC2)
2. Simple Storage Service (S3)
3. Virtual Private Cloud (VPC)
4. Amazon Lambda
5. Simple Notification Service (SNS)
6. Amazon OpenSearch Service (Elasticsearch)
7. CloudWatch
8. CloudTrail
9. AWS Backup
10. AWS DataSync
11. API Gateway

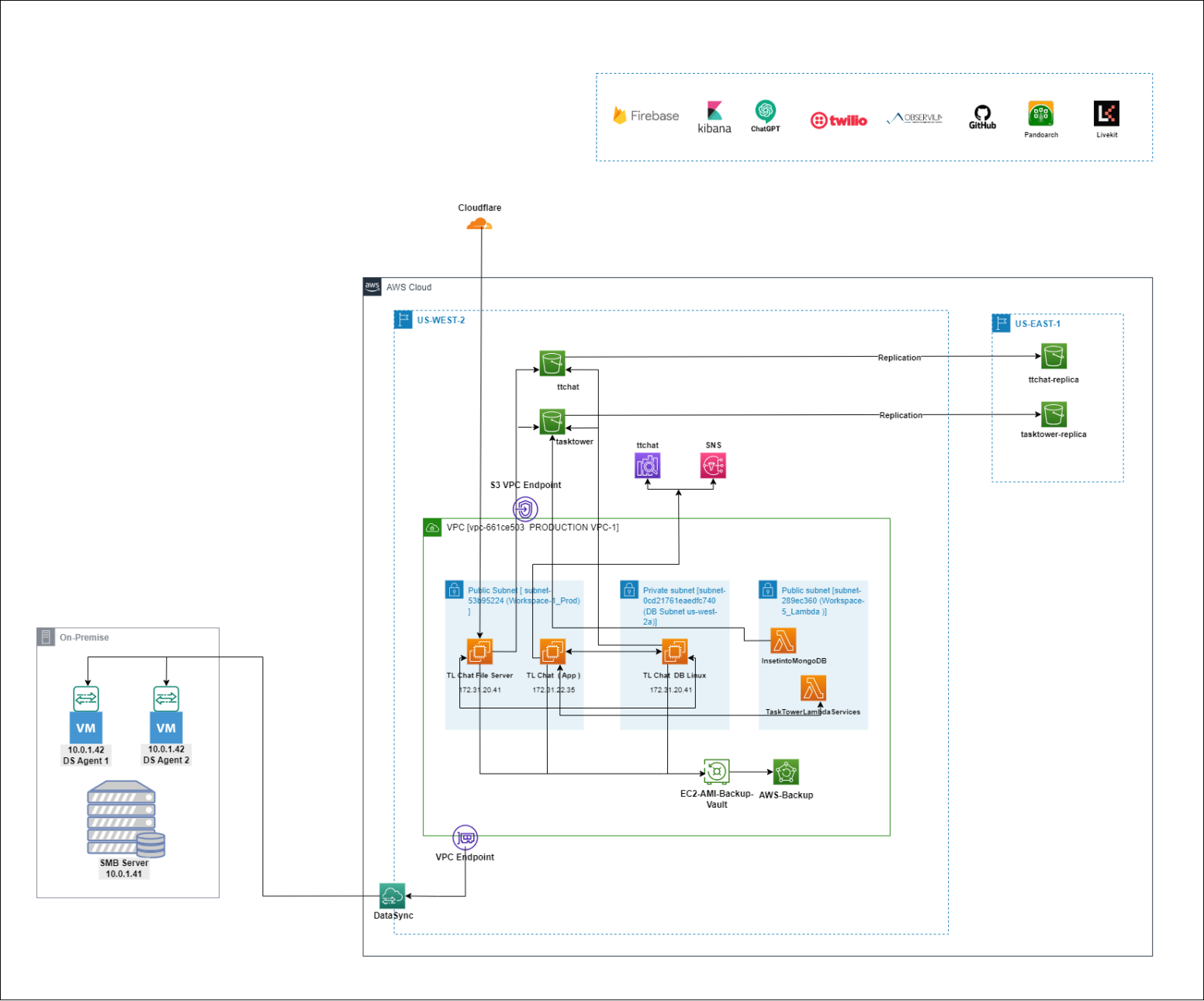


Figure 1: TL Chat Project Architecture.

# TL Chat Project Description

Our project is a robust, scalable, and secure AWS cloud-based architecture designed to support the TL Chat application. The architecture is designed with a focus on high availability, data redundancy, and low latency to ensure optimal performance for end-users.

The architecture is built on AWS cloud Environment within a Virtual Private Cloud VPC [vpc-661ce503 PRODUCTION VPC-1] that includes three subnets two public Subnet and one private Subnet.

Subnets are as follow

1. Public Subnet [ subnet-53b95224 (Workspace-1\_Prod) ]
2. Private subnet [subnet-0cd21761eaedfc740 (DB Subnet us-west-2a)]
3. Public subnet [subnet-289ec360 (Workspace-5\_Lambda )]

* In first public subnet we have host two instances. TL Chat File Server and the TL Chat App Server. The private subnet have TL Chat DB Linux which is a database server that connects to both the TL Chat File Server and the TL Chat App Server And In third Public subnet we have two Lambda functions: 'InsertIntoMongoDB' and 'Task Tower Lambda Service'.
* The 'Task Tower Lambda Service' is triggered whenever there is an update or upload of any data on the TL Chat file Server. This function moves the files from the instance to an S3 bucket name as 'Task Tower S3 Bucket'. Subsequently, the 'InsertIntoMongoDB' Lambda function is triggered, generating thumbnails that are stored in another bucket.
* The TL Chat App is integrated with Amazon OpenSearch Service having ttchat domain and have two data nodes (8peyefg9Qaavdzgc1dAQQw) &( WT2Jfw8QSU2djGzuh5jVFg) in us-west-2c, this service providing indexing and fast search results.
* Both S3 buckets are in the Oregon region, and take a backup of those buckets with cross region replication in North Virginia to ensure data redundancy and high availability.
* To provide low latency and better performance to chat app users, we are using open-source Cloudflare CDN service. The incoming requests are coming from first cloudflare and forward the request to TL Chat File Server and IP of TL Chat App Server's is pointed in Cloudflare under the name as chat.teamlocus.com.
* To ensure data safety and business continuity, we have set up the AWS Backup service, which takes daily backups [AMI] in the Oregon region having 15 days Retention period. Additionally, we have set up Data Sync service that syncs all data from S3 to an on-premises server which is located in New Jersey.
* This architecture provides a secure, scalable, and efficient environment for the TL Chat application, ensuring a seamless and high-performing user experience.

# Other Integrated Applications with TL Chat

1. Firebase
2. Kibana
3. Chat-GPT
4. Twillio
5. Obsrvium
6. Github
7. Pandoarch
8. Livekit