



Architecture diagrams for travel

Architecture of Crew Management System



Architecture of Crew Management System: Architecture diagrams for travel

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Crew management architecture i

 Crew management architecture diagram 1

 Further reading 2

 Diagram history 2

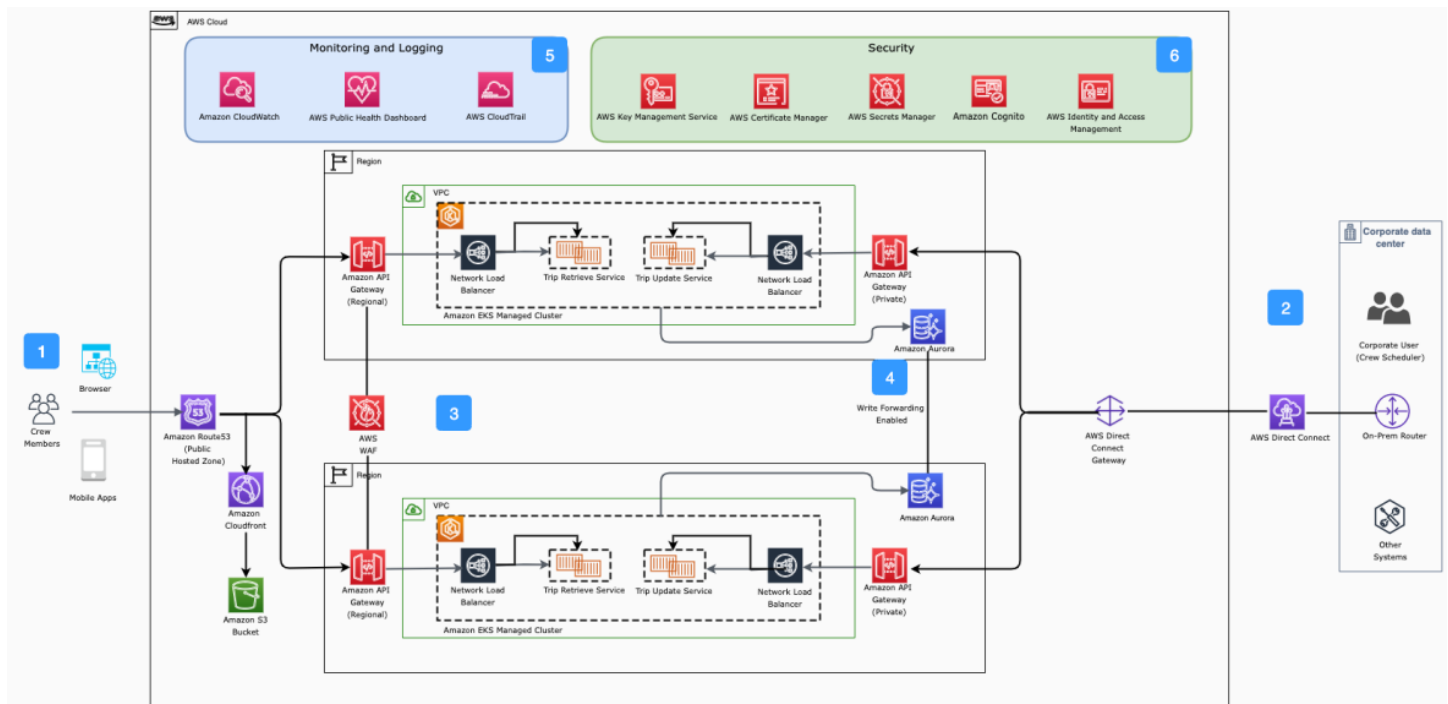
Architecture of Crew Management System

Publication date: **November 29, 2021** ([Diagram history](#))

This architecture focuses on Trip Update and Retrieve services. You can use AWS to create a highly available, secure, flexible, and cost-effective architecture for crew management systems.

This architecture extends the [Architecture for Airline Crew Management Systems](#). It deploys across multiple Regions for high availability and reduced latency.

Crew management architecture diagram



The following steps describe the architecture:

1. Crew apps resolve domain names through [Route 53](#) to the nearest region and [CloudFront](#) distribution. A regional [API Gateway](#) provides access to read crew trip data. CloudFront serves static pages from [Amazon S3](#).
2. An [Amazon Aurora](#) Global Database provides high availability and low latency with Aurora Replicas. With read replica write forwarding, applications send writes to any Aurora Global Database replica.

3. Deploy the application in multiple Regions for high availability and reduced latency. Use [CloudFormation](#) for single-click deployment. Use a fully managed [Amazon EKS](#) cluster for Trip Update and Retrieve services. Network Load Balancer routes traffic to Amazon EKS services.
4. Connect corporate users and systems by using [AWS Direct Connect](#). A private API Gateway provides access to crew schedulers for update and read operations.
5. Use [CloudWatch](#) and monitoring tools to optimize operational health.
6. Use AWS cloud security services for at-rest, end-to-end encryption.

Further reading

For additional information, see the following resources:

- [AWS Architecture Icons](#)
- [AWS Architecture Center](#)
- [AWS Well-Architected](#)

Diagram history

To receive updates about this reference architecture diagram, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Reference architecture diagram first published.	November 29, 2021

RSS subscription requirement

To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.