IMESH

mindtickle™

case study

MINDTICKLE PARTNERS WITH IMESH TO ACHIEVE RESILIENT AND RELIABLE SOFTWARE NETWORK

Overview

About Mindtickle

Mindtickle is an American SaaS unicorn that provides sales enablement and buyer enablement software. More than 350 customers worldwide (including the likes of Splunk, MongoDB, Okta, and Cisco) rely on their software to grow revenue by increasing knowledge and understanding of ideal sales behaviors.

Technology background

Mindtickle has more than 300 microservices spread across multiple tech stacks that run on Kubernetes (most of their workloads are in AWS EKS). Their cloud infra is applied in regions (US & Singapore) and different zones for high availability. Mindtickle builds new features and API capabilities and releases recent changes daily. From a traffic perspective, they get around 330 million monthly requests to their SaaS application.

350
Worldwide customers

300+
Microservices in AWS EKS

330
Million online requests per month

Challenges

Resilient and reliable infra was crucial for the best customer experience.

Mindtickle has many microservices that communicate among themselves, and new services get added up frequently as a part of innovation. Based on the traffic fluctuation, the replicas or the pods of these workloads (de)scaled automatically. Being a SaaS company, Mindtickle needs to provide the best customer experience without any service degradation. Thus, DevOps and SRE wanted to ensure their complicated and dynamic network's reliability and resiliency to avoid downtime.

Allocating developers for non-core feature development was not feasible.

One of the solutions for developing resiliency and reliability features such as circuit breaking, timeouts, and rate limiting was to direct app developers to build this solution in their application. Firstly, there were services with multiple tech stacks, such as Python, Go, Java, etc, and there needed to be a standardized procedure for developing this solution in their apps. More importantly, developers will need more time to solve core business problems.

Challenges

Lack of traffic observability was a severe blow for MTTR.

Due to a lack of understanding of their traffic, it was complicated for the SREs and Ops team to understand the real-time performance of the system. Without real-time metrics such as requests per second (RPS), traffic tracing, response code, the latency of workloads, etc., it was difficult to troubleshoot and diagnose performance issues of SaaS applications, giving rise to high MTTR.

Cost of communication was high.

Mindtickle workloads are spread across regions and zones; there are 80 to 90 nodes. And communication among them involved the data transfer from one zone to another zone. Cloud vendors such as AWS charge extra money for inter-geography data transfer. Mindtickle confronted increasing cloud bills with growing online customers without optimized network communication.

Istio Implementation

The Mindtickle team evaluated that service mesh software would be the right strategy to overcome the challenges associated with a complicated cloud and container services network. Their DevOps and architect team evaluated a few service mesh software but zeroed in on Istio. They chose Istio because it is an open-source, more feature-rich for managing network operations, provides native interoperability with other network products such as Ingress and application load balancer, and observability tools such as Prometheus, Grafana, Datadog, etc.

When they implemented the Istio service mesh in EKS, all the workloads were injected with the Envoy sidecar container, and the traffic was abstracted out of the application, i.e., the communication between any services (or gateway to services) happened through sidecars.

"We selected Istio service mesh over others because it is open source, feature rich and offered native integrations with existing network and observability software."

-Mindtickle DevOps team

The abstraction of the network was vital to manage and secure the east-west and north-south traffic in the landscape. Istio is topology-aware and selects the optimal route during communication, which led Mindtickle to save the cost of inter-node data transfer in AWS.

Istio Implementation

In the initial phase of the Istio implementation, Mindtickle implemented mTLS on the gateway-to-service and service-to-service network routes, which are crucial for business. And then, they also implemented a single pane of glass observability solution using Istio. All the essential traffic and resource consumption metrics are sent from the sidecar (Envoy proxies) to Prometheus. The IT Ops team and SREs can now visualize the performance of their HTTP and gRPC calls in a single Grafana dashboard and can quickly diagnose any issues or anomalies in their network.

Wrt resiliency and reliability, Mintickle's DevOps team has leveraged Istio to implement rate limiting, timeouts, and retries on any microservice without disturbing app developers. They have also implemented Fault injection to test the performance and behavior of a new application before rolling it out to production. With Istio, the DevOps team has also implemented rate limiting in the Envoy sidecar, thus improving clients' fair usage of their APIs-based services.

Encryption of traffic in critical routes using mTLS

Deep observability into all the HTTP and gRPC calls

Implemented rate limits, timeouts, retries, and fault injection into key networks

IMESH helped Mindtickle to adopt Istio faster

Istio service mesh is heavy and feature-rich software, and implementation can take time. Istio provides many resources to manage the network and implement AuthN/Z and observability. Instead of spending months learning and experimenting with Istio, Mindtickle's DevOps team partnered with IMESH to implement Istio for their enterprise.

IMESH has helped Mindtickle to implement all resilient and reliability features (such as rate limiting, fault injection, circuit breaking, etc.) for their SaaS application using Istio service mesh. IMESH experts handle all the configurations, integrations, and custom development with Istio service mesh. To enhance the performance with less resource consumption, the team optimized and tuned Istio in production.

With frequent updates about Istio and guaranteed SLA from IMESH, Mindtickle's DevOps team can now focus on modernizing their IT further with bleeding developer platform technologies and automation in their CI/CD process.

Ganesh Kumar
Senior Manager, DevOps/SRE
Mindtickle

"IMESH provided the solution to realize our goal- increasing resiliency and reliability- using Istio service mesh in the stipulated time. With their Istio expertise, we have saved tremendous time and cost in implementing, configuring, and optimizing the mesh for our huge container workloads."

Results

With the Istio service mesh, the Mindtickle DevOps team achieved their goal of gaining control over their complicated network and making it more resilient, reliable, and secure.

>30%

Improvement in resiliency and reliability of cloud and containerized applications.

Zero

Disruption of services resulting in the superior end-customer experience of their SaaS application.

Upto 50%

Improvement in MTTR due to unified observability into real-time traffic, network and workload performance.

Low

Cost of data transfer due to topology-aware data transfer between services whose workloads are spread across geography.