STREAM-ZERO · Follow publication

Apache APISIX: Overview and Advantages

8 min read · Jul 1, 2024

balaji bal Follow

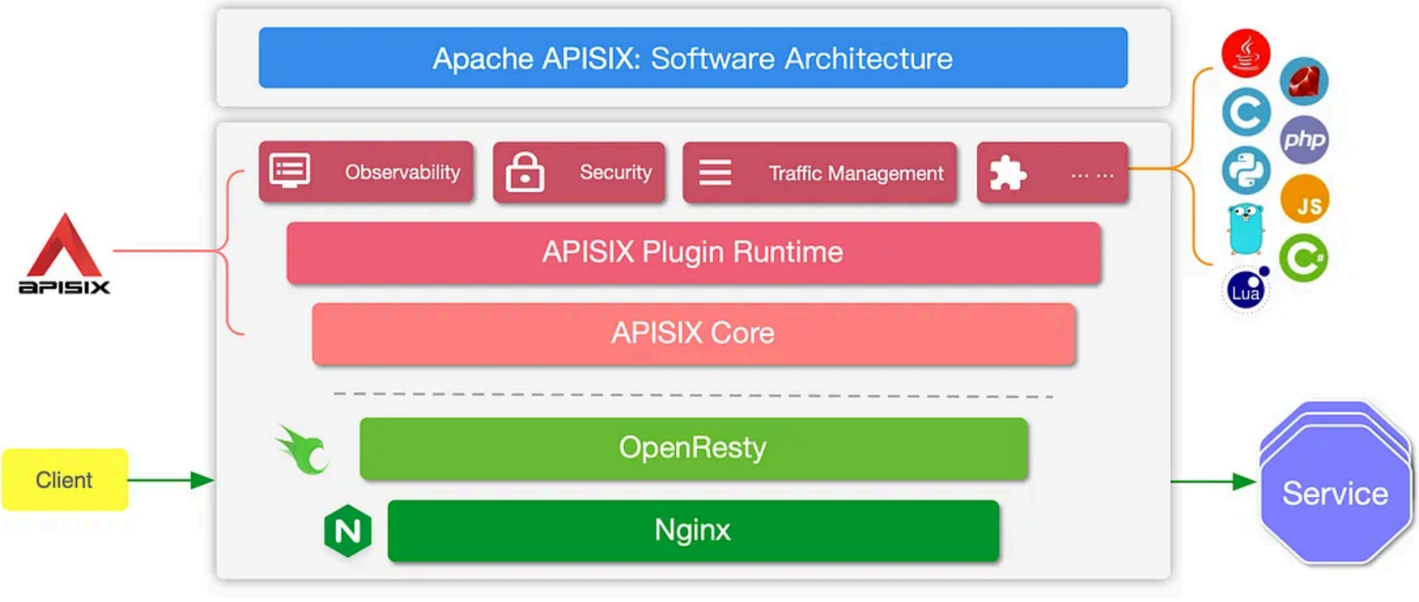
Listen Share More

Apache APISIX is a dynamic, cloud-native, distributed API gateway, offering a robust platform for managing microservices, service mesh, and API management.

This article provides an overview of Apache APISIX, its advantages, and a look at some of its key plugins.

What is Apache APISIX?

Apache APISIX is an open-source API gateway used to manage, monitor, and route API requests. It is built on top of the Nginx HTTP server with etcd as its configuration store, making it highly scalable and performance-efficient. APISIX is designed to provide dynamic, real-time, high-performance API routing and management. **It supports all the traffic management features required for API oversight, such as load balancing, dynamic upstream, canary release, circuit breaking, authentication, observability, and more.**



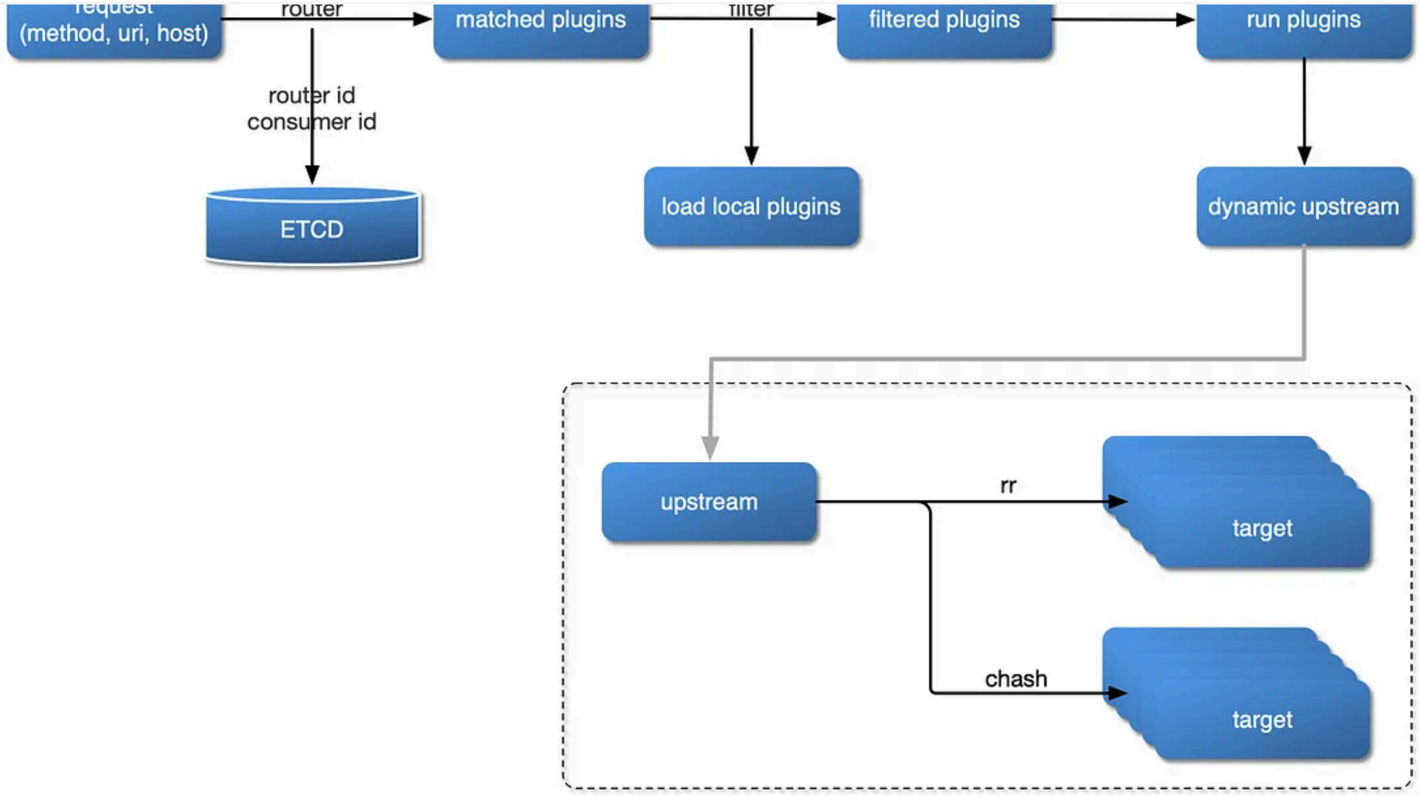
Advantages of Apache APISIX

The following are a few key advantages of APISIX

- **High Performance and Scalability:** One of the most significant advantages of APISIX is its high performance and scalability. It can handle thousands of API requests per second, with latency measured in milliseconds. Its stateless architecture allows it to scale horizontally on cloud-native environments.
- **Dynamic Configuration:** APISIX supports dynamic configuration and hot updates without the need for restarting the service. This feature is crucial for continuous integration and delivery (CI/CD) pipelines, ensuring minimal downtime and seamless updates.
- **Extensibility Through Plugins:** APISIX boasts a rich set of plugins, extending its capabilities far beyond simple API routing. These plugins can handle authentication, security, monitoring, and various other API management needs. The plugin-oriented architecture also allows developers to write their plugins, making APISIX highly customizable.
- **Multi-Protocol Support:** Beyond HTTP/HTTPS, APISIX supports a variety of communication protocols, including WebSocket, gRPC, MQTT, and more. This makes it a versatile gateway solution for different types of applications and services.

- **Security:** Security is a top priority in APISIX. It offers numerous features to safeguard your APIs, such as IP filtering, dynamic SSL certificate handling, OAuth2, JWT authentication, and more, ensuring that your API endpoints are protected against unauthorized access.

- **Cloud-Native:** Designed for the cloud-native ecosystem, APISIX integrates seamlessly with Kubernetes, providing a powerful API gateway solution for microservices architectures. It supports service discovery and dynamic configuration in a cloud-native environment, making it ideal for modern



Key Plugins Overview

The following are a few key plug-ins

Authentication Plugins

- **JWT:** Validates JSON Web Tokens to secure API access.

- **OAuth2:** Implements OAuth 2.0 authentication framework for APIs.

- **Key Auth:** Simple key-based authentication for APIs.

Security Plugins

- **IP Restriction**: Allows or blocks requests based on IP addresses.
- **URI Blocklist/Allowlist**: Blocks or allows requests based on URI patterns.
- **Bot Detection:** Protects against various types of bot traffic.

Traffic Control Plugins

- **Rate Limiting:** Limits the number of requests a user can make to an API within a given timeframe.
- **Circuit Breaker**: Prevents failures from cascading by temporarily blocking problematic services.
- **Load Balancer**: Distributes incoming API requests across multiple backend services based on configured rules.

Observability Plugins

- **Prometheus**: Exports metrics in the Prometheus format for monitoring and alerting.
- **Zipkin**: Integrates with Zipkin for distributed tracing, helping to monitor and troubleshoot latency issues.

Other Notable Plugins

- **Serverless**: Allows for the execution of serverless functions in response to API requests.
- **Transformations**: Modifies requests and responses on the fly without changing the backend service code.

Apache APISIX represents a comprehensive, efficient, and scalable solution for modern API management challenges. Its performance, dynamic configuration, extensive plugin ecosystem, and support for multiple protocols make it an attractive choice for organizations looking to optimize their API strategies. Whether you're managing microservices, legacy systems, or a mix of both, APISIX offers a flexible and powerful gateway to streamline your API operations.

Comparison APISIX vs Kong

When comparing Apache APISIX to Kong, two leading open-source API management solutions, it’s essential to delve into their architecture, performance, extensibility, and community support to understand their strengths and how they cater to different needs within the API management ecosystem.

Architecture and Performance

Apache APISIX is built on top of the Nginx HTTP server and etcd. Its design focuses on high performance and low latency, with benchmarks often highlighting its ability to handle thousands of requests per second with minimal overhead. This makes APISIX particularly well-suited for scenarios where performance and efficiency are critical.

Kong, on the other hand, also leverages the Nginx server underneath but uses a PostgreSQL or Cassandra database for configuration storage. While Kong is designed for high performance, the choice of database for configuration storage can introduce variability in performance, especially at scale. However, Kong’s architecture is robust and flexible, making it a solid choice for enterprises with complex integration needs.

Extensibility Through Plugins

Both APISIX and Kong offer extensive extensibility through plugins, which cover authentication, security, traffic control, and observability. However, the approach and availability of these plugins can differ.

APISIX boasts a rich set of plugins and emphasizes the ease of writing custom plugins. Its plugin ecosystem is rapidly growing, thanks in part to its active community. The dynamic nature of APISIX’s configuration allows for hot reloading of plugins without downtime, which is a significant advantage in continuous deployment environments.

Kong also has a strong focus on extensibility, with a wide range of plugins available. Kong’s plugin ecosystem is mature, with extensive documentation and community contributions. Writing custom plugins for Kong is straightforward, allowing businesses to tailor the gateway to their specific needs.

Community and Ecosystem

Apache APISIX has seen rapid growth in its community since its inception, with a significant increase in contributors, GitHub stars, and adoption in various industries. This vibrant community has led to a fast-paced development environment, with new features and plugins being added regularly. APISIX's status as an Apache Software Foundation project also adds a level of prestige and reliability, encouraging enterprise adoption.

Kong has a well-established community and has been around longer than APISIX. It has a broad user base, including many large enterprises, and offers both an open-source version and an enterprise version with additional features and support. Kong's community is active, with numerous resources available for users to learn from and contribute to.

APISIX Authentication Plug-Ins

Apache APISIX offers a comprehensive suite of authentication plugins, catering to a wide range of security requirements for modern web applications and services. These plugins are designed to ensure that API access is securely controlled, providing mechanisms for identifying and authorizing users and services.

Apache APISIX's authentication plugins provide a powerful and flexible solution for securing API access. From simple key-based authentication to more complex OAuth 2.0 and OIDC flows, APISIX offers a range of options to meet the security needs of any application. By leveraging these plugins, developers can ensure that their APIs are protected against unauthorized access, providing a safe and reliable experience for end-users. The choice of plugin depends on the specific requirements of the application, including the level of security needed, the user authentication workflow, and the existing infrastructure.

Below, we delve into some of the key authentication plugins available in APISIX, highlighting their features and use cases.

JWT (JSON Web Tokens)

The JWT plugin is one of the most widely used authentication mechanisms in APISIX. It allows services to verify the identity of users based on JSON Web Tokens. This stateless authentication method is highly scalable and suitable for distributed systems. The JWT plugin decodes and validates the token sent by the client, ensuring it was issued by a trusted authority and that it hasn't been tampered with. It's particularly useful in single sign-on (SSO) scenarios and when implementing cross-service authentication in microservices architectures.

OAuth2

The OAuth2 plugin in APISIX implements the OAuth 2.0 framework, providing a robust authorization mechanism. This plugin is ideal for scenarios where third-party applications need to access user data without exposing user credentials. It supports various OAuth 2.0 flows, such as the authorization code flow, implicit flow, and client credentials flow, making it versatile for different types of applications. The OAuth2 plugin is essential for building secure and flexible API ecosystems that require fine-grained access control.

Key Authentication

Key authentication (Key Auth) is a simple yet effective plugin for protecting APIs. It requires clients to include a predefined API key in their requests, either in the headers or query parameters. This method is straightforward to implement and manage, making it suitable for scenarios where complex authentication mechanisms are unnecessary. While not as secure as JWT or OAuth2, Key Auth provides a basic level of security that can be sufficient for internal or low-risk APIs.

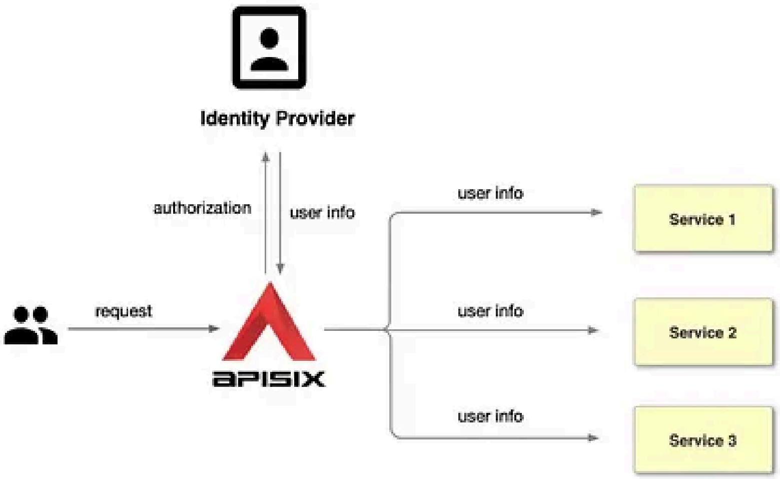
LDAP Authentication

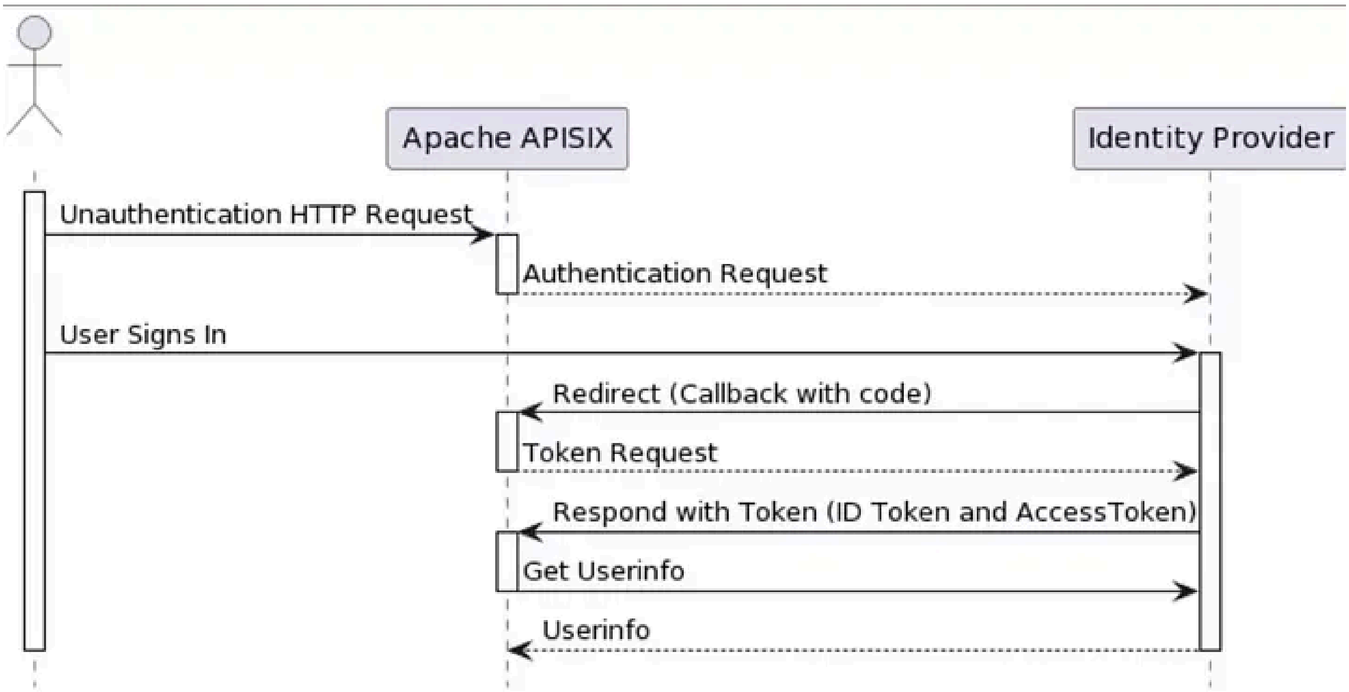
The LDAP Authentication plugin allows APISIX to integrate with LDAP (Lightweight Directory Access Protocol) servers for user authentication. This is particularly useful for organizations that already use LDAP for managing user credentials and group memberships. By leveraging this plugin, developers can easily add LDAP-based authentication to their APIs, ensuring that API access is consistent with the organization's existing security policies and user management practices.

OpenID Connect

OpenID Connect (OIDC) is a simple identity layer on top of the OAuth 2.0 protocol. The OIDC plugin in APISIX facilitates authentication and identity verification by leveraging trusted identity providers (IdPs). This plugin is particularly useful for applications needing to authenticate users across multiple domains or platforms. It simplifies the authentication process by allowing users to log in using their existing identities from well-known IdPs like Google, Facebook, or corporate identity providers supporting OIDC.

Integrating APISIX with Keycloak




Apache APISIX
Identity Provider
Unauthentication HTTP Request
Authentication Request
User Signs In
Redirect (Callback with code)
Token Request
Respond with Token (ID Token and AccessToken)
Get Userinfo
Userinfo

The following articles describes the process of integrating APISIX with Keycloak: API Gateway APISIX Integrates Keycloak for Authentication

Conclusion

Choosing between Apache APISIX and Kong often comes down to specific requirements and preferences. APISIX might be the better choice for organizations looking for high performance, dynamic configuration, and a rapidly evolving plugin ecosystem. Its architecture is particularly suited for cloud-native environments and microservices.

Kong, with its robustness, mature ecosystem, and enterprise features, is well-suited for organisations looking for a proven solution that can handle complex integrations and offers the flexibility of open-source and enterprise versions.

Both API gateways offer significant advantages, and the decision should be based on the specific needs of the project, including performance requirements, the complexity of the API ecosystem, and the desired level of community involvement and support.

At StreamZero we use APISix and have integrated it into our automation pipelines and with KeyCloak which has been a part of our infrastructure for over 5 years. Prior to switching to APISix we used OpenResty. We love it's speed and the ease of use of the APIs.

Api Gateway   Big Data   Streamzero   Kong   Enterprise Technology

Follow

Published in STREAM-ZERO

35 followers · Last published Apr 6, 2025

All things event driven.

Follow

Written by balaji bal

398 followers · 477 following

Founder @ HeadGym.com

No responses yet

Sumit Jaiswal

What are your thoughts?

More from balaji bal and STREAM-ZERO

balaji bal

Why Are Polymarket Betters Losing Money?

In this article we explore the potential reasons why only 12.7% of wallets on polymarket are showing profits.

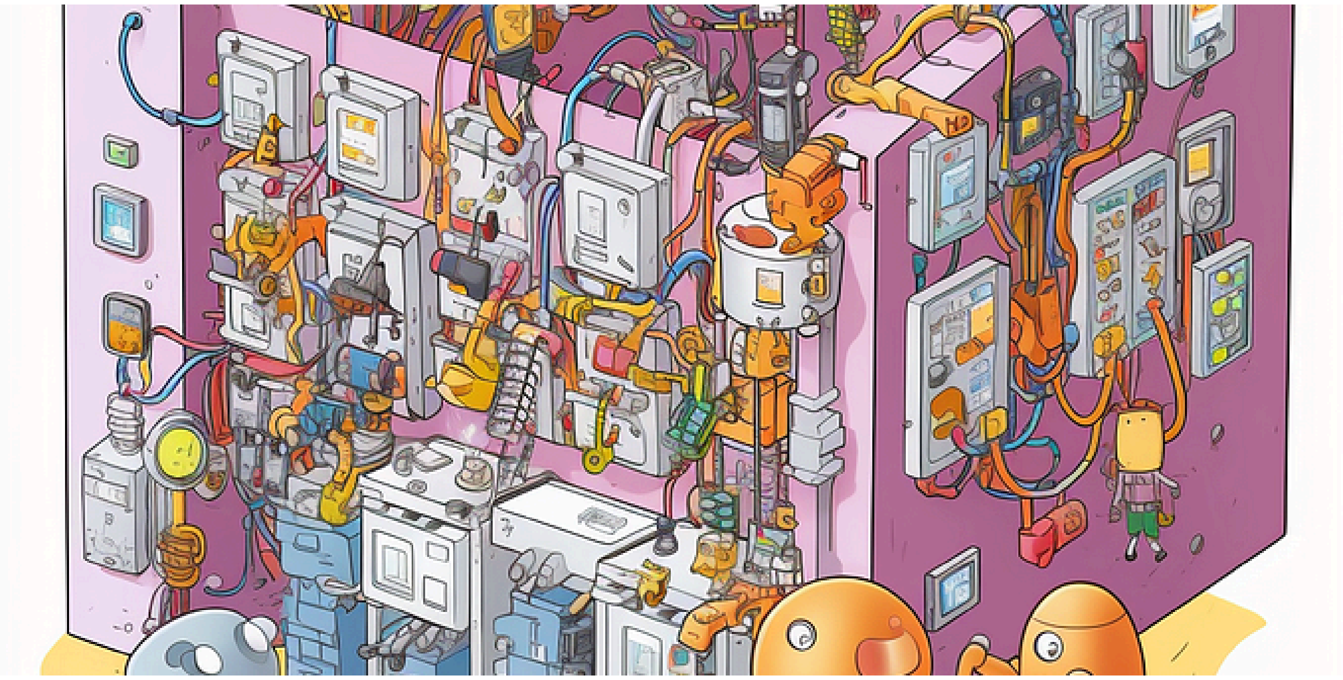
Apr 21 62 1

In STREAM-ZERO by balaji bal

Comparing Trino, ClickHouse, and Apache Doris: Architectures, Use Cases, and Performance

Datalakes and Data Platforms are going through another cycle of change and evolution. In recent years I have been implementing solutions...

Apr 29, 2024 👏 255 💬 1

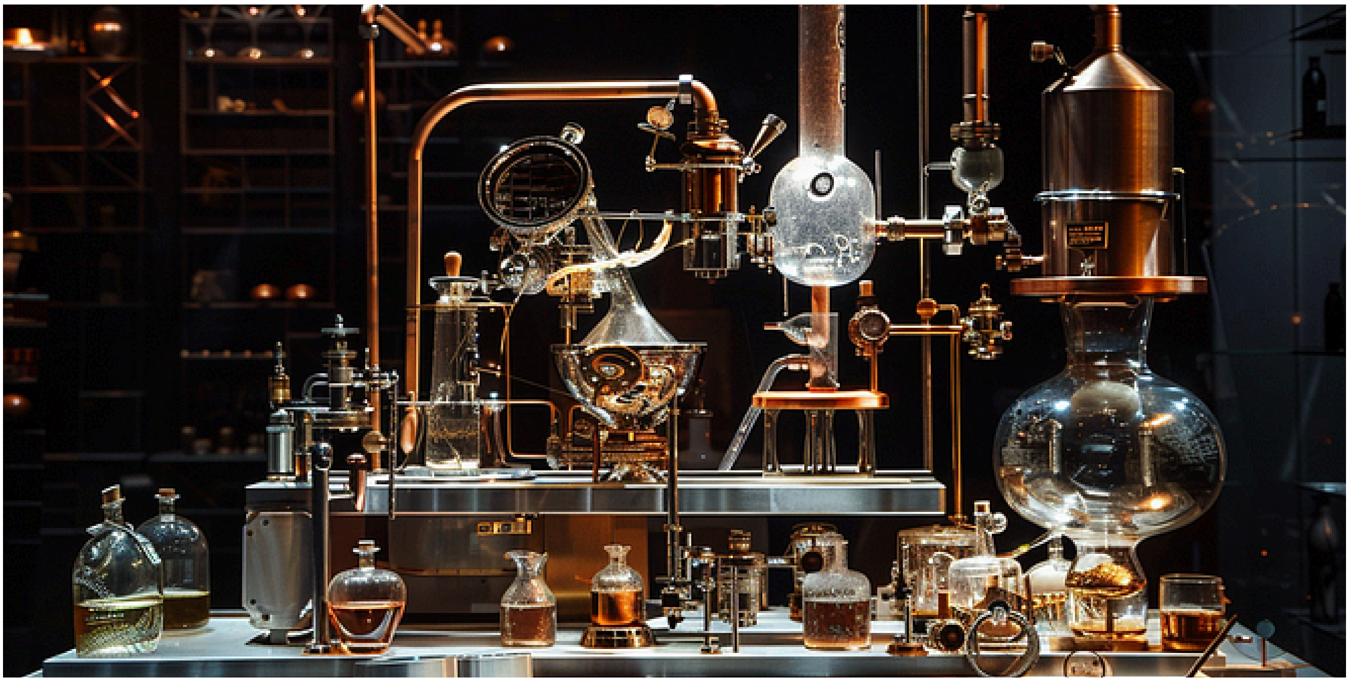


In STREAM-ZERO by balaji bal

Understanding Message Delivery in Kafka with Multiple Partitions

This is part of an ongoing series on Apache Kafka examining various aspects related to developing applications on Apache Kafka. Apache...

Nov 10, 2023 128 1



In STREAM-ZERO by balaji bal

Understanding the Essentials of Model Distillation in AI

"RAG" (Retrieval-Augmented Generation) and Model Distillation are both advanced techniques used in the field of artificial intelligence...

Jun 8, 2024 124 1

See all from balaji bal

See all from STREAM-ZERO

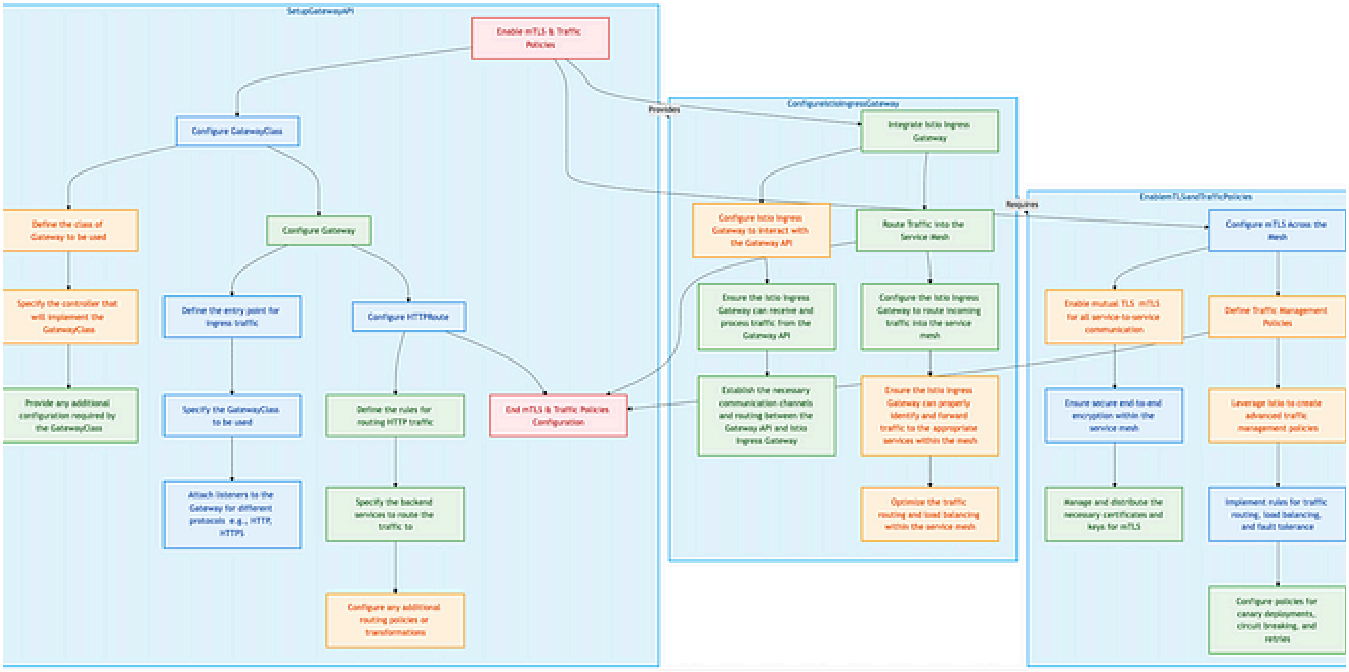
Recommended from Medium

Sohail Saifi

Kubernetes Is Dead: Why Tech Giants Are Secretly Moving to These 5 Orchestration Alternatives

I still remember that strange silence in the meeting room. Our CTO had just announced we were moving away from Kubernetes after two years...

Jun 7 3.3K 131


SetupGatewayAPI
Enable mTLS & Traffic Policies
Provides
Configure GatewayClass
Define the class of Gateway to be used
Configure Gateway
Specify the controller that will implement the GatewayClass
Define the entry point for ingress traffic
Configure HTTPRoute
Provide any additional configuration required by the GatewayClass
Specify the GatewayClass to be used
Define the rules for routing HTTP traffic
End mTLS & Traffic Policies Configuration
Attach listeners to the Gateway for different protocols e.g., HTTP, HTTPS
Specify the backend services to route the traffic to
Configure any additional routing policies or transformations
ConfigureIstioIngressGateway
Integrate Istio Ingress Gateway
Configure Istio Ingress Gateway to interact with the Gateway API
Route Traffic into the Service Mesh
Ensure the Istio Ingress Gateway can receive and process traffic from the Gateway API
Configure the Istio Ingress Gateway to route incoming traffic into the service mesh
Establish the necessary communication channels and routing between the Gateway API and Istio Ingress Gateway
Ensure the Istio Ingress Gateway can properly identify and forward traffic to the appropriate services within the mesh
Optimize the traffic routing and load balancing within the service mesh
Requires
EnablemTLSandTrafficPolicies
Configure mTLS Across the Mesh
Enable mutual TLS mTLS for all service-to-service communication
Define Traffic Management Policies
Ensure secure end-to-end encryption within the service mesh
Leverage Istio to create advanced traffic management policies
Manage and distribute the necessary certificates and keys for mTLS
Implement rules for traffic routing, load balancing, and fault tolerance
Configure policies for canary deployments, circuit breaking, and retries

In DevOps.dev by Noel Benji

Modern Kubernetes Networking With Gateway API, Istio, & Beyond

From the traditional Ingress API to the newly introduced Gateway API, and with tools like Istio and Cilium enhancing service mesh...

Feb 8 👏 1

ThreadSafe Diaries

Kafka Just Got Outclassed by This Open-Source Underdog

If you've ever built a data pipeline or worked on event-driven systems, chances are Apache Kafka was the first name that came to mind. It's...

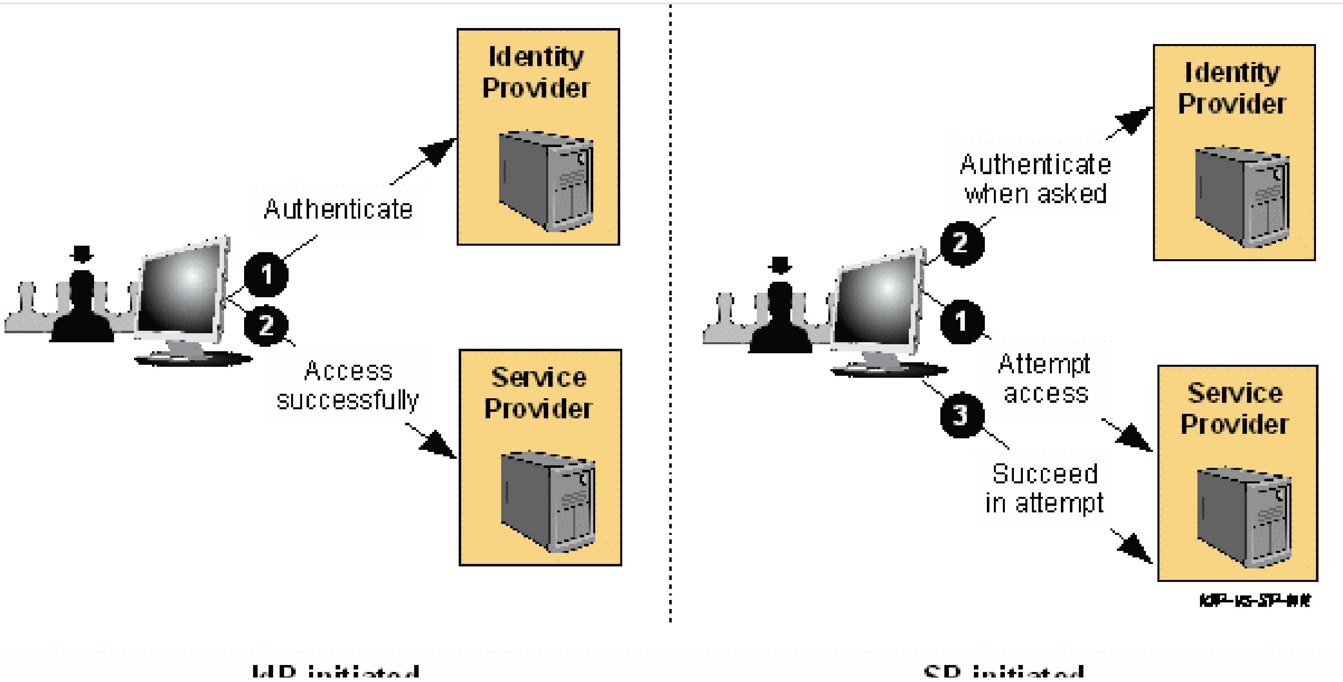
✦ Jul 7 👏 290 💬 7

Nivetha Thangaraj

Kubernetes Is Dead: Why Tech Giants Are Secretly Moving to These 5 Orchestration Alternatives

By Nivetha Thangaraj

Jul 1 185 36

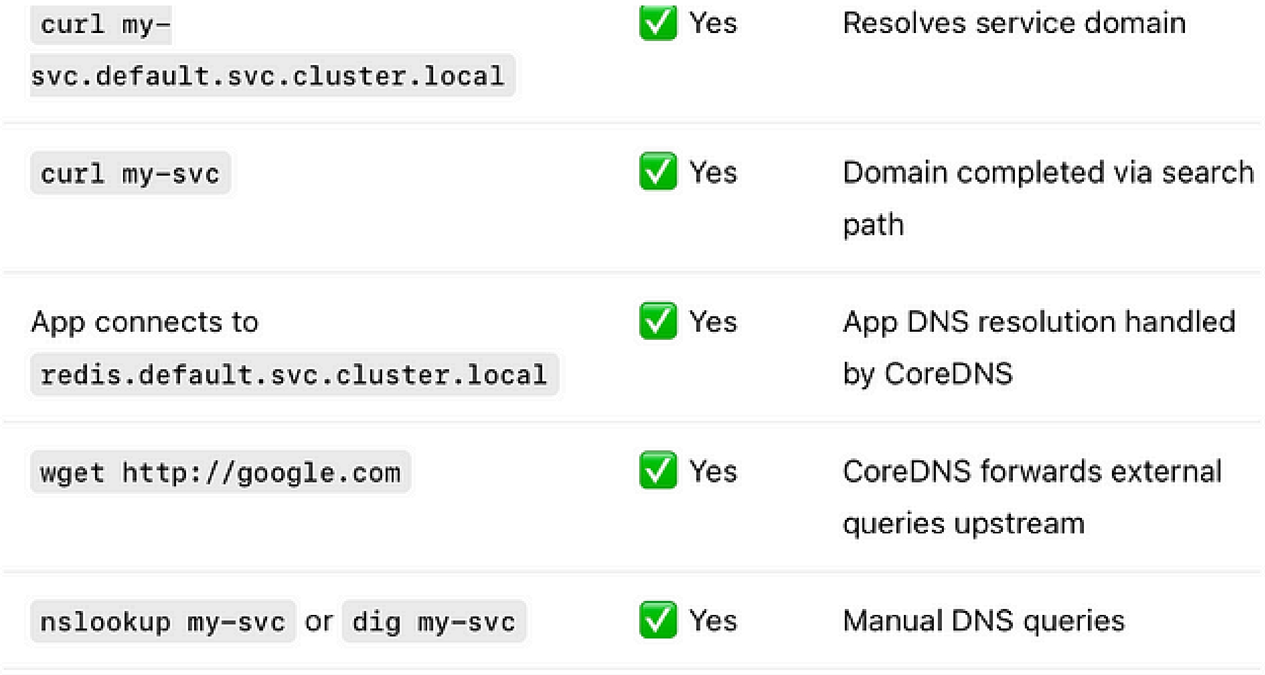


Salim Amine Bou Aram

SAML 2.0 with Keycloak and Go

Keeping user authentication in check across various applications is a major hurdle.

Jun 30



`curl my-svc.default.svc.cluster.local`	✅ Yes	Resolves service domain
`curl my-svc`	✅ Yes	Domain completed via search path
App connects to `redis.default.svc.cluster.local`	✅ Yes	App DNS resolution handled by CoreDNS
`wget http://google.com`	✅ Yes	CoreDNS forwards external queries upstream
`nslookup my-svc` or `dig my-svc`	✅ Yes	Manual DNS queries

Kevin Wan

In-Depth Understanding of CoreDNS

1. Core Concepts

May 22 13

See more recommendations