

CUSTOMER STORY

Nylas improves account synchronization performance by 30x with Speedscale

“

The capability to shift-performance left was imperative for the Nylas team to meet customer expectations and do so within a short time frame.



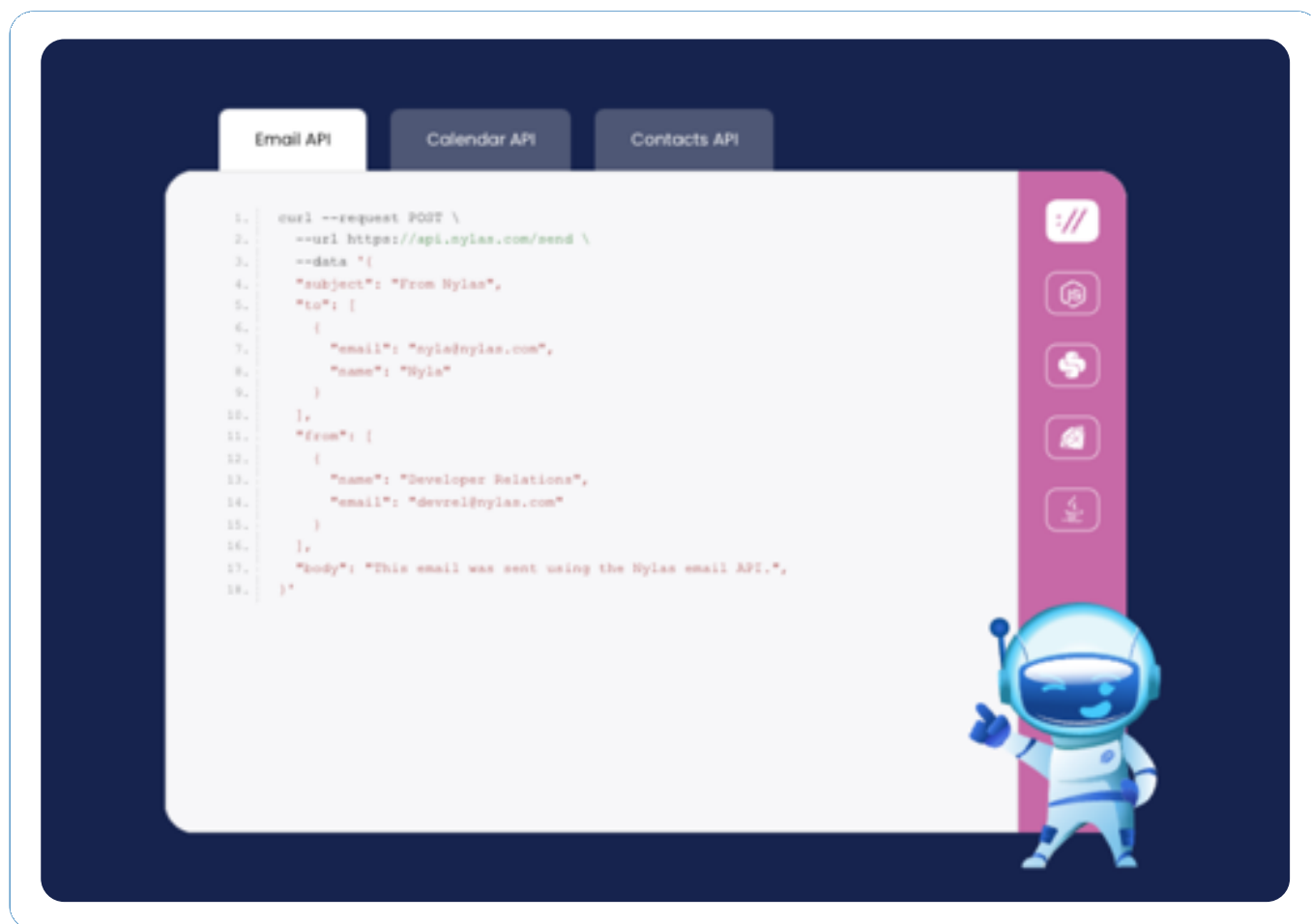
Nylas improves account synchronization performance by 30x with Speedscale



Read how Nylas made the strategic decision to implement traffic replay to ensure performance parity in their migration to Kubernetes

Company

[Nylas](#) is a fast-growing SaaS technology vendor that helps companies synchronize email, calendar, contacts and other productivity data across thousands of accounts. With more companies working from home, their product usage has spiked, and they are building a next generation platform to handle additional use cases and scale. Over 26,000 developers use Nylas to send over 1 billion API requests per day to providers such as Gmail, Microsoft Exchange, Outlook.com, Yahoo! and more.



Challenge

The company was facing a new (yet welcome) challenge they had not previously encountered—larger customers were signing on to use their platform, bringing higher integration loads with them. To prepare for the demands of these larger customers, David Ting, SVP of Engineering, was looking for a solution to help ensure uptime and stability.

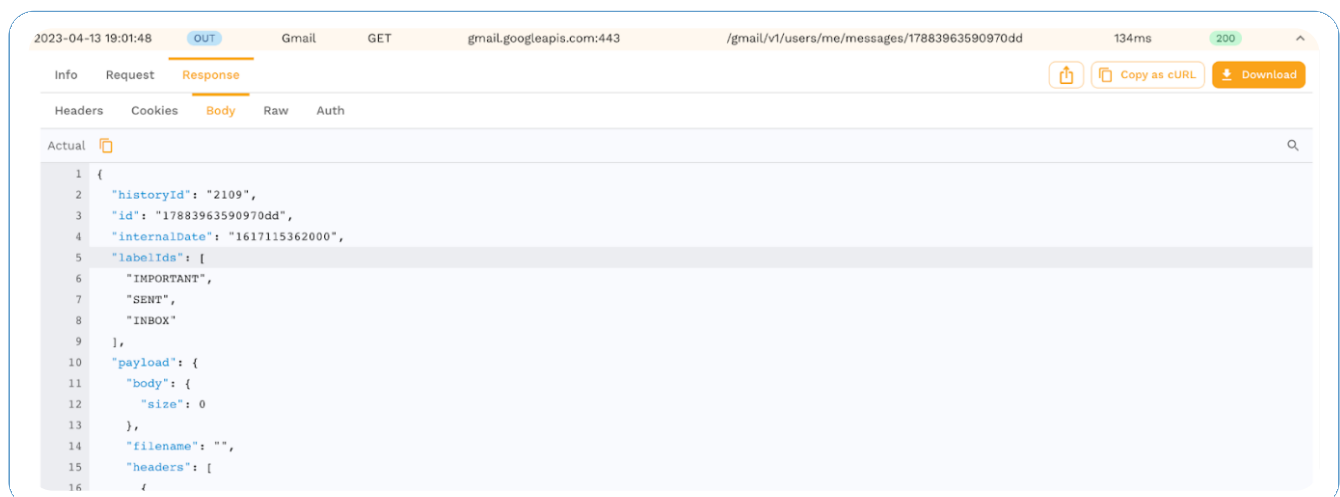
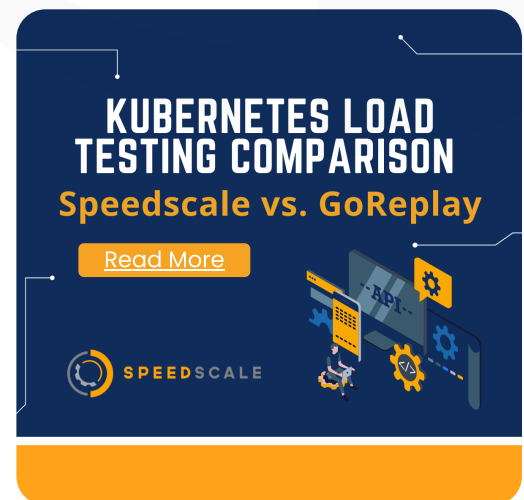
In parallel, Nylas was also undergoing a migration from EC2 (an existing monolith application) to Kubernetes. The migration was required to accelerate release velocity, as well as provide the option to let customers deploy Nylas software on-premise. With the migration to Kubernetes, Nylas gained flexibility on how code was built and deployed, but a new set of challenges was introduced:

- Each service team iterated independently, thus **dependencies and connections grew**
- Applications required **big rewrites** to maximize microservice advantages
- While Kubernetes is elastic and self-heals, **benchmarking individual APIs was difficult**
- **Lack of control over third-party services**, like Office365 and Google Apps
- **Difficulty maintaining the application** during a shift in architecture (e.g. ongoing business, release schedules, and application quality)

Evaluating Solutions

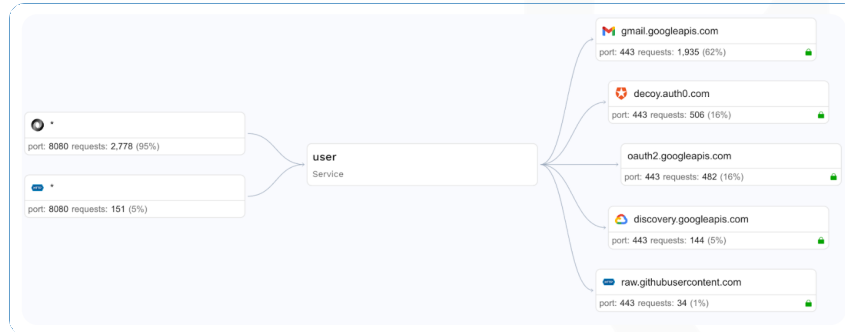
Nylas' requirements were for an easy-to-use platform that 1) supported their unique environment of Kubernetes and third party APIs with integrated TLS security and authentication, and 2) had an ability to record traffic and replay at scale. Having initially tried [GoReplay](#), Nylas found that the frequency and changing data requirements were beyond the scope of what the open source tool was capable of.

David ultimately made the strategic decision to implement Speedscale's [traffic replay](#) in order to ensure performance parity in their migration to Kubernetes, as well as allow for fast iteration and tuning on their high-throughput APIs.



Using Speedscale

While moving from EC2 to Kubernetes, Speedscale's traffic replay framework provided an easy-to-use interface for observing traffic and orchestrating and automating replays. By observing APIs, not only does Speedscale generate traffic replays, but mocks of critical dependencies as well.



Within minutes, Speedscale was able to examine and ingest all of the inbound and outbound API calls across their service architecture. An initial set of recorded transactions could be expanded to represent 10K email accounts, and made available in a single container.

The screenshot shows the Speedscale interface with tabs for Summary, Traffic, Replays, Transforms, and Token Discovery. The 'Transforms' tab is active, showing a table of traffic data. A detailed view of a specific transaction is shown on the right, including a 'Recorded Value' section with a JSON payload.

Date	Direction	Tech	Operation	Host	Endpoint	Duration	Status
2023-04-13 18:47:24	OUT	Auth0	POST	decoy.auth0.com:443	/oauth/token	2ms	404
2023-04-13 18:47:24	OUT	Google O...	POST	oauth2.googleapis.com:443	/token	1ms	404
2023-04-13 18:47:24	OUT	Auth0	POST	decoy.auth0.com:443	/oauth/token	230ms	200

Recorded Value

```
{
  "historyId": "3074",
  "id": "17a40ec0f52cd492",
  "internalDate": "16245866",
  "labelIds": [
    "UNREAD",
    "CATEGORY_UPDATES",
    "INBOX"
  ],
  "payload": {
    "body": {
      "size": 0
    }
  }
}
```

Real	Mock	Direction	Host
Real	Mock	egress	raw.githubusercontent.com
Real	Mock	egress	discovery.googleapis.com
Real	Mock	egress	oauth2.googleapis.com
Real	Mock	egress	decoy.auth0.com
Real	Mock	egress	gmail.googleapis.com



Speedscale is a game changing capability that enables large architectural upgrades with quality. Traffic replay is high coverage and fast.



David Ting

VP, Engineering, Nylas



Observing API traffic, [generating snapshots](#) and mocks, and replaying traffic was easily done through the dashboard and manifest files. Snapshot summaries and replay readouts were available during and after execution. Perhaps most importantly, orchestration of replay containers was handled by a [Kubernetes operator](#) and command-line for [integration with CI tools](#) and repeatable execution.



Nylas was able to improve account synchronization performance by 30x in just a few months.

Results & Future Outlook

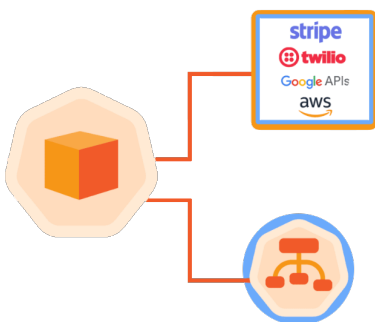
Speedscale provided many benefits to the Nylas team, but chief among them was the ability to mock 3rd party endpoints at a scale unavailable anywhere else but production. With the availability of Office365 and Google Workplace mocks, the teams could rapidly iterate and validate at production volumes, with new versions of the sync, syncback, API, and dashboard services. This removed the need to create volumes of test data, because Speedscale orchestrated the dependencies and values for repeatable scenarios. The capability to shift-performance left was imperative to meet customer expectations and do so in a short timeframe.

As a result, Nylas was able to improve account synchronization performance by 30x in just a few months by using Speedscale.

Feature Spotlight: Service Mocks

Service mocks are a critical component of development and testing. Find out the 'why' and 'how' behind our Kubernetes traffic-based mocks!

[Read the Blog](#)



The capability to shift-performance left was imperative for the Nylas team to meet customer expectations and do so in a short time frame.

The team has further plans to utilize Speedscale to test performance gains of different CPU architectures, API designs, and of course, confirm performance parity as the migration from monolith to Kubernetes continues.