



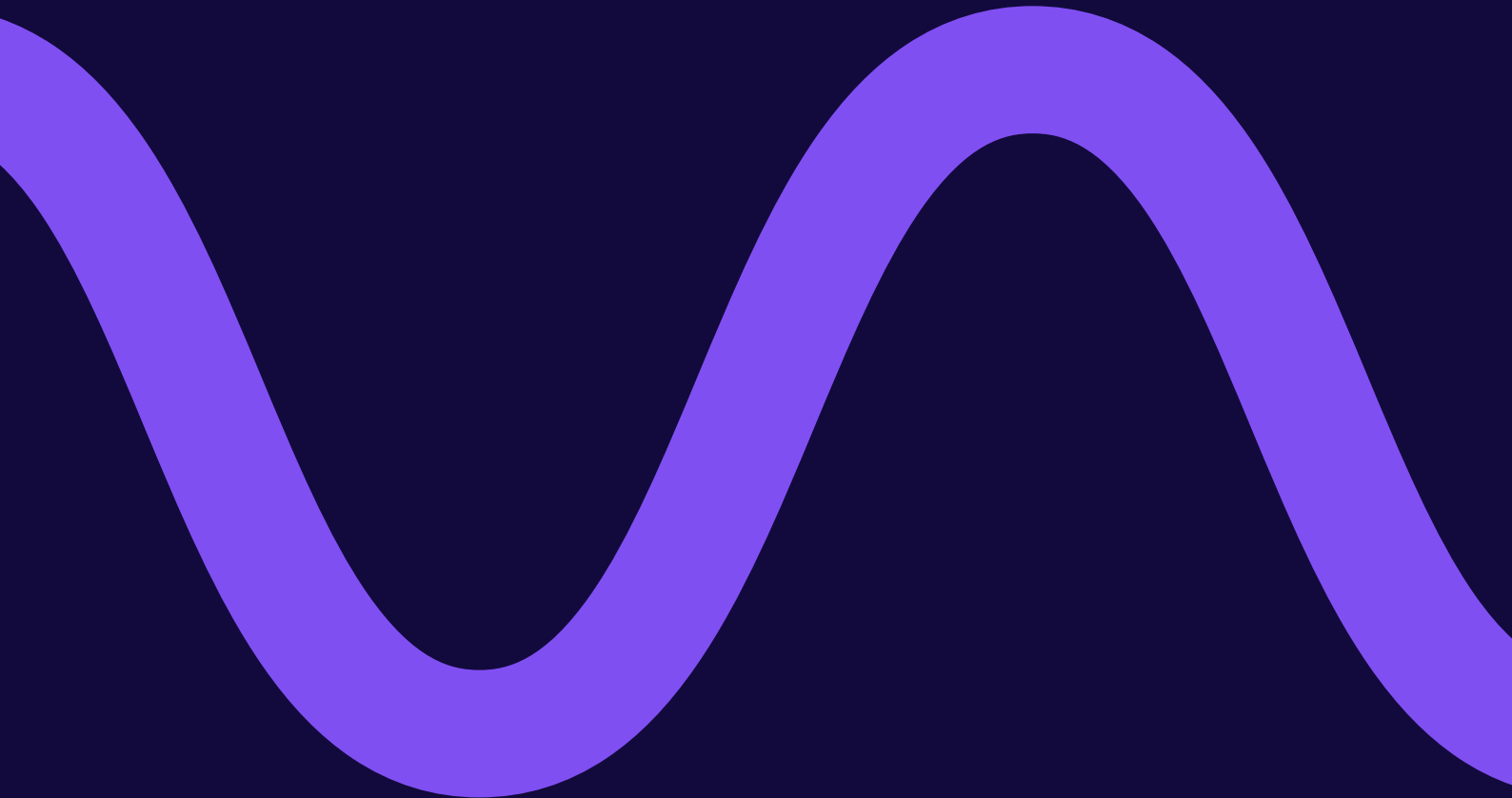
Case Study

Booking.com + Ververica Platform

Building a platform to provide Security-as-a-Service.
Delivering streaming-in security related data for
processing

Contents

About booking.com	03
The Challenge	04
The Solution	05
The Results	07



About Booking.com

Part of Booking Holdings Inc. (NASDAQ: BKNG), Booking.com's mission is to make it easier for everyone to experience the world. By investing in the technology that helps take the friction out of travel, Booking.com's marketplace seamlessly connects millions of travelers with memorable experiences every day. For more information, follow @bookingcom on social media or visit globalnews.booking.com.

```
35 self.debug = debug
36 self.logger = logging.getLogger(__name__)
37 if path:
38     self.file = open(os.path.join(path, 'request_fingerprints.txt'), 'a')
39     self.file.seek(0)
40     self.fingerprints.update(request_fingerprint(request))
41
42 @classmethod
43 def from_settings(cls, settings):
44     debug = settings.getbool('DEBUG', False)
45     return cls(job_dir(settings), debug)
46
47 def request_seen(self, request):
48     fp = self.request_fingerprint(request)
49     if fp in self.fingerprints:
50         return True
51     self.fingerprints.add(fp)
52     if self.file:
53         self.file.write(fp + os.linesep)
54
55 def request_fingerprint(self, request):
56     return request_fingerprint(request)
```

The Challenge

What wasn't working?

With the increasing need to process security data streams, we needed to have an orchestration tool that can manage a large number of Flink applications (as well as their dependencies) and

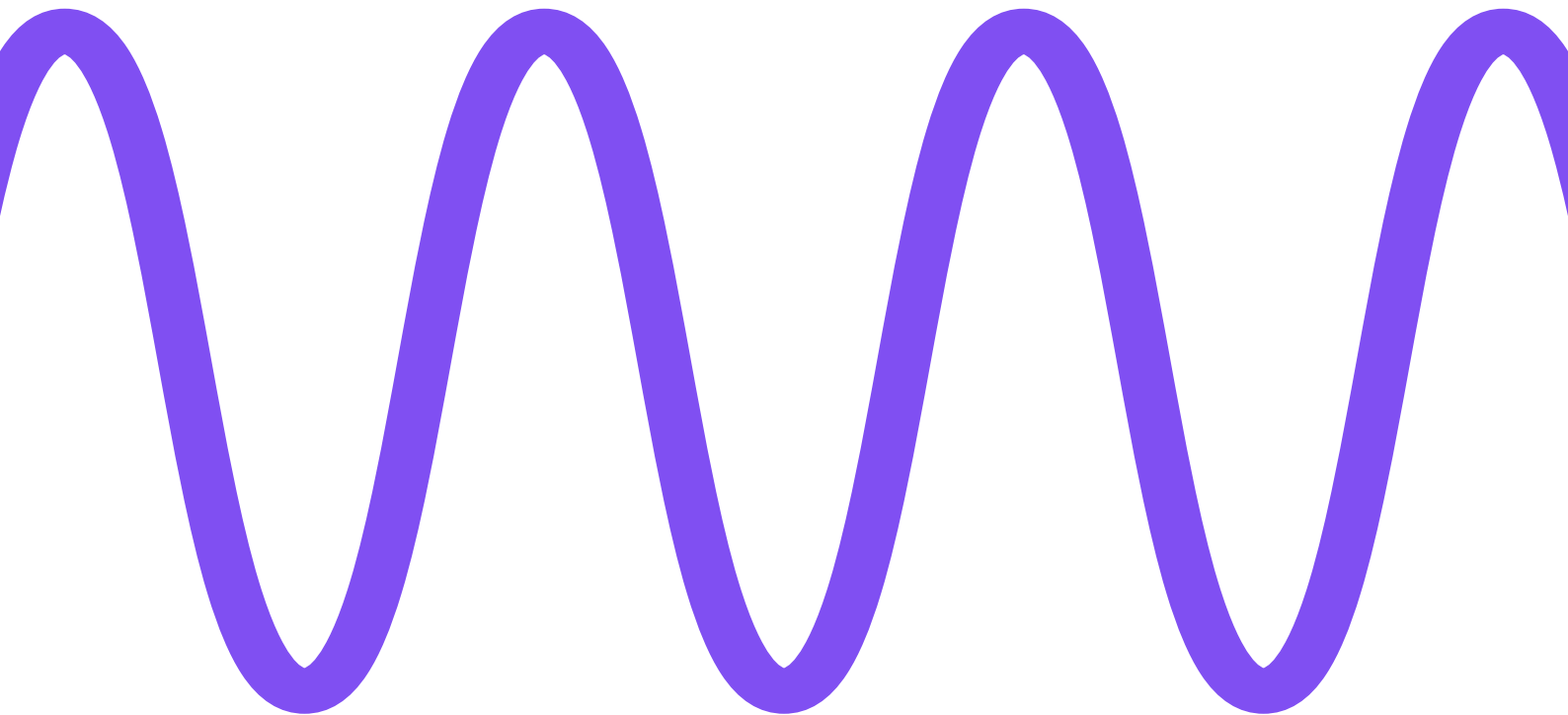
manage their stateful upgrade easily. Furthermore, the tool should enable multi-tenancy and separation of duties. Orchestrating Flink applications themselves was the problem to solve.

How were you managing your jobs previously?

Before exploring Ververica (or other solutions), we were using helm charts and shell scripts to manage our Flink jobs. It involved a lot of operational toil and manual management. In particular, it did not work well for the stateful upgrades of Flink applications.

Can you elaborate on the need for a Security-as-a-Service and if there are any industry need factors at play?

Booking.com provides an ecosystem used by both partners (accommodations, flights, rides and attractions) and travelers. A lot of data needs to be processed to provide a secure end to end experience for our users and to report cases of unsafe hotels and properties. Since we have a lot of security teams focusing on different use-cases (within security), we need to enable segregation of duties and access control. The former means the people in charge of maintaining and supporting our infrastructure should not be able to see the contents of the data. The latter; only the people specifically authorized for it should have access to the data.



The Solution

What were the key deciding factors in selecting Ververica?

For us, it is crucial to have the ability to switch to a model where Booking.com's other security teams would manage and leverage the control of the lifecycle for their own Flink applications, allowing the Platform team to just take care of keeping the solution, which makes that possible, running.

What else played a role in the selection to use Ververica Platform?

Built on a well-understood and widely used Kubernetes standard, Ververica Platform provides a security-as-a-service platform for Booking.com to manage a large number of Flink applications with an intuitive Web UI (multi-tenancy) without any limitations on Flink and Kubernetes features. Ververica Platform is flexible enough to be implemented in the Booking.com ecosystem with minimal friction and simple enough to use, providing value from the very start.

What does Ververica Platform provide for the needs of Booking.com?

Ververica Platform allowed us to solve the problem of lifecycle management of Flink applications and presenting this as a service for users to leverage on. It extends Kubernetes APIs; we're building on well-understood and widely used industry standards. Its reduction of complexity shortens the lead time to bring Flink application lifecycle management to a production-grade solution. It, in turn, brings more stability and reliability. It also provides visibility for the owners to better debug and maintain their applications themselves.

Any particular features that Ververica offered over other platform providers?

No other solution that we had a chance to study in depth worked on Kubernetes properly with abstracted snapshot management and high availability using all the native features of Flink itself, which seemed like the most forward-evolving solution.

What would you say were the instant short-term wins?

A user can log in and immediately see what Flink jobs there are in a namespace. Simple things like an inventory for Flink jobs in a namespace instead of having to rely on helm, words, or shell scripts.

A web UI also always makes life easier. Anything that shortens the work time to achieve an action has a significant impact because we're talking about avoiding downtime and faster velocity for development as well as detecting and solving some issues.

Great! Anything from a long-term perspective?

The one we nailed is the scale-up of a number of applications. That is definitely something that is well achieved, and we took advantage of it very fast. It was not initially thought to be possible until we onboarded.

We are managing more jobs with fewer work hours from our team. That's a net improvement of the ratio. The hours required for the scale-up were completely detached from the number of job applications executed. It was always the ultimate desire, but at the time, it wasn't clear how that would be in reality. The reality has proven to be above expectations; I would say perfect.

It was certainly more a midterm to a long-term desired state. We achieved it in the short term, so it was a quick win.

What exceeded your expectations?

In terms of exceeding expectations, the Web UI itself was above expectations. We set out to build an orchestration of Flink applications - declarative seamless, reducing downtime and friction. The Web UI was just something we didn't expect to have as much influence and users as it ended up having - a huge win!

Another aspect is the ability to scale up without complications other than quotas and Kubernetes. The amount of control and flexibility that it provided was exceptional because it allows full flexibility while allowing you to use pretty much everything from Flink (that you would want to be able to use) from an orchestration solution.

During implementation, we had some technical requirements to fit Ververica in our corporate environment. We needed some tweaks to Ververica itself to fit it. Your engineering team responded faster and better than we expected to accommodate this technical need.

The Results

Easily manage hundreds of Flink applications without increasing the complexity and manpower inventory of Flink jobs at a glance (no relying on helm, shell scripts, etc.).

- Shorten the lead time to go to production: the deployment time and stateful upgrade time reduced from hours to a few minutes.
- Fast response time on Flink or Ververica Platform questions reduced from potentially infinity to a few minutes/hours.

How many jobs do you manage with VVP?

Right now we have about 250 Flink Applications managed by Ververica Platform. Before Ververica, we could only handle between 10-20.

“
Ververica is allowing us to build separate environments for application lifecycles.
”

What changes did you see to your deployment times?

Deployment times went from a single job taking around 20 minutes to complete to only 1-4 minutes.

For example, in the past upgrading the code of an application would have taken anywhere between ten minutes to half an hour and would require the intervention of both developers and platform engineers. With Ververica the upgrade itself takes a couple of minutes and can be fully performed by the developers. They have also been able to improve our monitoring and visibility of what's going on during and after the upgrade

Having support means we have SLAs for getting guidance from experts. In contrast, getting help from the Flink community could go from a few hours, or even days, to infinity. With Ververica's support we can have Flink experts on the line in minutes, if needed.

About Ververica

Ververica's mission is to power the core business of every company with cutting-edge real-time stream processing technology. In order to do that, the team at Ververica focuses on building the best technology available for stream processing, while at the same time creating a global and open community around this technology.

We build and develop Ververica Platform, a stream processing platform that enables every enterprise to power their real-time business and use a production-grade streaming infrastructure while at the same time we actively contribute and participate in the open source Apache Flink® community, the underlying technology framework of Ververica Platform itself.

For more information:

sales@ververica.com

[@VervericaData](https://twitter.com/VervericaData)

www.ververica.com

