GIGAOM

Image credit: Dolmatov

William McKnight, Jake Dolezal
May 19, 2021

High Performance API Management Testing vPPD

Product Evaluation: API7 and Kong Enterprise

High Performance API Management Testing

Product Evaluation: API7 and Kong Enterprise

Table of Contents

1. Summary

This report focuses on API management platforms deployed in the cloud. The cloud enables enterprises to differentiate and innovate with microservices at a rapid pace. It allows API endpoints to be cloned and scaled in a matter of minutes. And it offers elastic scalability compared with on-premises deployments, enabling faster server deployment and application development, and allowing less costly compute.

More importantly, many organizations depend on their APIs and microservices for high performance and availability. For the purposes of this paper, we define "**high performance**" as that required by companies which experience workloads of **more than 1,000 transactions per second** and need a **maximum latency of less than 30 milliseconds** across their API landscape. For these organizations, the need for performance is tantamount to their need for management, because they rely on these API transaction rates to keep up with the speed of their business.

An API management solution **cannot** be a performance bottleneck. On the contrary, many of these companies are looking for a solution to load balance across redundant API endpoints and enable high transaction volumes. If a business experiences 1,000 transactions per second, that translates to 3 billion API calls in a month. Large companies with high-end API traffic levels are commonly seeing monthly API calls exceed 10 billion. Thus, performance can be a critical factor when choosing an API management solution.

In this paper, we reveal the results of performance testing we completed with two full-lifecycle API management platforms: **API7 and Kong Enterprise (Kong EE)**.

API7 outperformed Kong EE at all attack rates for our single-node setup. API7 had almost 14 times lower latency than Kong EE at the 99.99th percentile at 10,000 requests per second. The latencies for API7 and Kong EE tended to diverge at higher percentiles. The difference in latency is pronounced at the 99.9th and 99.99th percentiles and at the maximum latency in all our runs.

Testing hardware and software in the cloud is very challenging. Configurations may favor one vendor over another in feature availability, virtual machine processor generations, memory amounts, storage configurations for optimal input/output, network latencies, software and operating system versions, and the workload itself. Even more challenging is testing fully managed as-a-service offerings for which the underlying configurations (processing power, memory, networking, and so forth) are unknown to us. Our testing demonstrates a narrow slice of potential configurations and workloads.

As the sponsor of the report, API7 opted for their default API gateway—the solution was not tuned or altered for performance. GigaOm selected the configuration for Kong Enterprise that was closest in terms of CPU and memory configuration.

We leave the issue of fairness for the reader to determine. We strongly encourage you, as the reader, to look past marketing messages and discern for yourself what is of value. We hope this report is

informative and helpful in uncovering some of the challenges and nuances of platform selection.

We have provided enough information in the report for anyone to reproduce this test. You are encouraged to compile your own representative workloads and test compatible configurations applicable to your requirements.

2. API Management in the Cloud

Application programming interfaces, or APIs, are now a ubiquitous method and *de facto* standard of communication among modern information technology applications. Large companies and complex organizations have turned to APIs for exchanging data to knit these heterogeneous systems together and turn data into a service. APIs have begun to replace older, more cumbersome methods of information sharing with reusable, loosely coupled microservices. This gives organizations the ability to share data across disparate systems and applications without creating the technical debt of proprietary, unwieldy vendor tools.

APIs and microservices also give companies an opportunity to create standards for the interoperability of applications—both new and old—creating modularity and governance. Additionally, they broaden the scope of data exchange with the outside world, from mobile technology and smart devices to the Internet of Things, because organizations can securely share data with non-fixed location consumers and producers of information.

Due to the proliferation of APIs, the need has arisen to manage the multitude of services a company relies on—both internally and externally. API endpoints themselves vary greatly in protocols, allowed methods, authorization/authentication schemes, and usage patterns. Additionally, IT departments need granular control over their hosted APIs to enforce rate limiting, quotas, policies, and user identification, and to ensure high availability, and prevent abuse and security breaches. Exposing APIs opens the door to many partners who can co-create and expand the core platform without knowing anything about the underlying technology.

The landscape of API management in the cloud varies greatly according to an organization's needs and underlying architecture. However, for the sake of this discussion, we will classify cloud API management into two deployment styles:

Hybrid Cloud – A hybrid cloud API management solution allows organizations to deploy and configure the underlying cloud resources of their choosing. For instance, they get to choose the Amazon Web Services Elastic Compute Cloud (AWS EC2) instance or Azure Virtual Machine types to install their API components (management tools and endpoints), giving them choices in the number of CPU cores, memory, storage, and networking options, and the ability to tune their environment at the operating system level. The advantages of this approach include being able to deploy the solution on-premises, in a private cloud, a public cloud, or in a hybrid of the three.

Fully Managed Cloud – A fully managed API management solution is a turn-key, as-a-service offering. These deployments are typically fast and convenient but often lack granular configuration and control capabilities. Often, the underlying architecture is obfuscated, such that users have no idea of the compute power "under the hood." A major difference from build-your-own is that fully managed solutions are usually tethered to a particular public cloud, like AWS, Azure, or Google. We did not review a fully managed cloud offering in this test.

API management software vendors offer either build-your-own or fully managed cloud deployment

styles, and a few offer both. While there are many platforms that can provide the functionality to manage APIs, we are interested in the high-performance use case. Again, for the purposes of this paper, we define "**high performance**" as that required by companies which experience workloads of **more than 1,000 transactions per second** and need a **maximum latency of less than 30 milliseconds** across their backend APIs and microservices.

The purpose of this paper is to explore vendor API management offerings for this high-end performance use case.

Solution Profile: API7

API7 is built on Apache APISIX and is maintained by the Shenzhen Zhiliu Technology Co. Apache APISIX is a dynamic, real-time, high-performance API gateway. APISIX provides rich traffic management features such as load balancing, dynamic upstream, canary release, circuit breaking, authentication, observability, and more. You can use Apache APISIX to handle traditional north-south traffic, as well as east-west traffic between services. It can also be used as a k8s ingress controller.

API7 is an enterprise deployment of APISIX with features that include multi-cluster management, multi-work partitioning, authority management, version management, auditing, statistical reporting and other products to meet the core needs of enterprise users.

Figure 1 shows the technical architecture of API7 on top of the Apache APISIX engine.

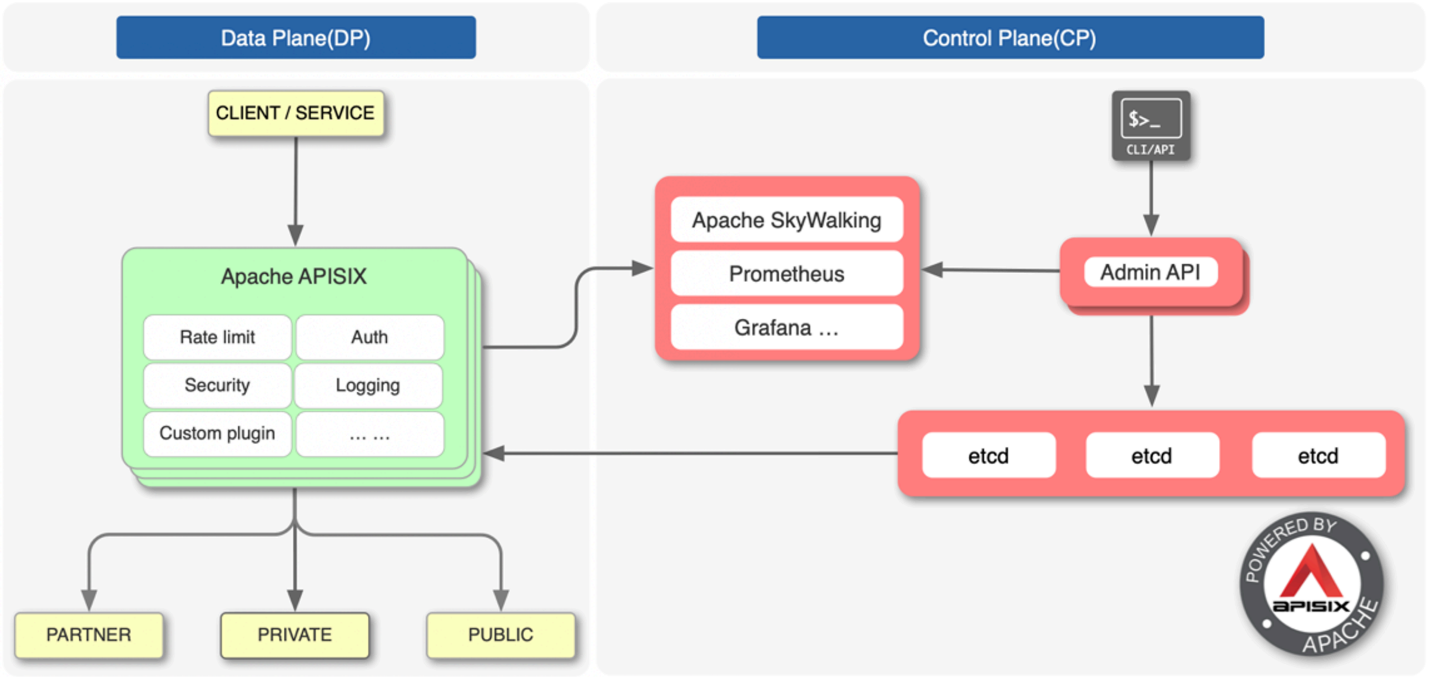

Figure 1. API7 Technical Architecture

API7 is a hybrid cloud deployment style and can be used in multi-cloud, on-premises, and hybrid environments.

Solution Profile: Kong Enterprise

Kong was originally known as Mashape until the release of its API platform. Kong became an open source project in 2015 with a wide range of functionalities. Kong leverages NGINX open source for its underlying API gateway and adds features including open source plugin support, load balancing, and service discovery. Kong Enterprise features expanded functionalities, such as a management dashboard, a customizable developer portal, security plugins, metrics, and 24×7 support. In 2019, Kong released a fully managed cloud offering, Kong Cloud. We did not test Kong Cloud.

Kong Enterprise can be deployed both in the cloud or on-premises. Debian and Red Hat-based package managers (Yum and Apt-Get) have Kong in their repositories, and Docker and CloudFormation options are also available.

Kong can operate as a single node or nodes can join each other to form a cluster. In a cluster configuration, a load balancer (such as NGINX Open Source) is used on the edge to provide a single address for clients and to distribute requests among the Kong nodes using a chosen strategy (for example, round robin or weighted). Scaling Kong horizontally is simple. Kong is stateless, so adding nodes to the cluster requires pointing a new node to its external database (PostgreSQL or Cassandra), so it can fetch all the configuration, security, services, routes, and consumers information it needs to begin processing API requests and responses, including the IP address or fully qualified domain name (FQDN) to the load balancer out front.

Kong has an ecosystem of plugins (referred to as the Kong Hub), supporting both open source and enterprise plugins such as LDAP authentication, CORS, Dynamic SSL, AWS Lambda, Syslog and others. Since it is based on NGINX, Kong allows users to create their own plugins using LuaJIT.

3. GigaOm API Workload Test Setup

API Workload Test

The GigaOm API Workload Field Test is a simple workload designed to attack an API or an API management worker node (or a load balancer in front of a cluster of worker nodes) with a barrage of identical GET requests at a constant number of requests per second.

To perform the attacks, we used the HTTP load-testing tool, WRK2[1], a free-to-use workload test kit available on GitHub. The WRK2 tool returns a latency distribution and a summary of status codes for every test. Latency was measured by the attacker as the interval between the time when an individual API request was made and the time when the API response was received. Thus, if we tested 10,000 requests per second for 60 seconds, the attack tool recorded 600,000 latency values. We used that data to compile and interpret the results of the test.

The test also requires a backend API (deployed on NGINX OSS) that can listen and respond to requests. In this case, our backend API listens for a GET request, such as:

```
http://localhost:8000/test_uri
```

The API then responds with a string of 1,024 pseudorandom Unicode characters, such as:

```
taZ3psgHkQojalo…
```

The backend API we used is further documented in the Appendix.

Kong Enterprise 2.2 was acquired and installed from the AWS Marketplace.[2] An API7 v.1.7 tarball was provided by API7.

For both API7 and Kong, we applied the rate-limiting plugin and set the value of how many HTTP requests can be made in a given period of seconds to 2,999,999.

We completed five attempts per test on each platform, configuration, and request rate. We started with an attack rate of 10,000 requests per second (rps) and scaled up to 20,000 rps. We ran each test for 60 seconds. We captured the latencies at the 50^{th}, 90^{th}, 95^{th}, 99^{th}, 99.9^{th}, and 99.99^{th} percentiles and the maximum latency seen during the test run. We recorded the test run that resulted in the median maximum. Error status codes included HTTP status codes 429 Too Many Requests or any 5xx codes, most often 500 Internal Server Error. A success rate of 100%, evident if no error notation is made on the chart, means all requests returned a 200 OK status code.

In addition, we tested the same configuration with JSON Web Token (JWT) authentication enabled,

and authenticated each request with the JWT credential. We tested using 10,000 rps attack rates.

For a fourth test, we generated 1,000 identical routes and equally distributed the requests across all 1,000 routes at 10,000 rps.

The results are shared in the Field Test Results section.

Test Environments

Single Gateway Node

NAME	NUMBER	EC2
Attack Node	1	c4.4xlarge
API Gateway data plane	1	c4.2xlarge
API Gateway control plane + Storage: Kong: PostgreSQL | API7: etcd	1	c4.2xlarge
Upstream Server	1	c5n.2xlarge

Environment Checklist

NAME	VALUE
Ping latency (between servers)	less than 0.150 ms
Operating System (API7)	CentOS 7.8
Operating System (Kong EE)	Amazon Linux 2
Rate-limiting (API7)	2,999,999 requests per second
Rate-limiting (Kong)	2,999,999 requests per second
File descriptors per process (ulimit -n)	1,024,000

Software Version information

NAME	VERSION
API Gateway	API7 1.7
API Gateway	Kong Enterprise 2.2.0.0-beta1
Upstream	NGINX 1.14.0
Test Tool	WRK2 4.0.0

Results may vary across different configurations and, again, you are encouraged to compile your own representative workloads and test compatible configurations applicable to your requirements.

---

1. https://github.com/giltene/wrk2
2. https://aws.amazon.com/marketplace/pp/B08P51PKC1

4. Test Results

This section analyzes the latencies in milliseconds from the various 60-second runs of each of the scaled GigaOm API Workload Field Tests described above. A lower latency is better—meaning API responses are coming back faster. Also, the latency reveals the response time at the 50th, 90th, 95th, 99th, 99.9th, and 99.99th percentiles and the maximum latency. These are important values for service-level agreements (SLAs) and for knowing what the slowest response times a user might experience.

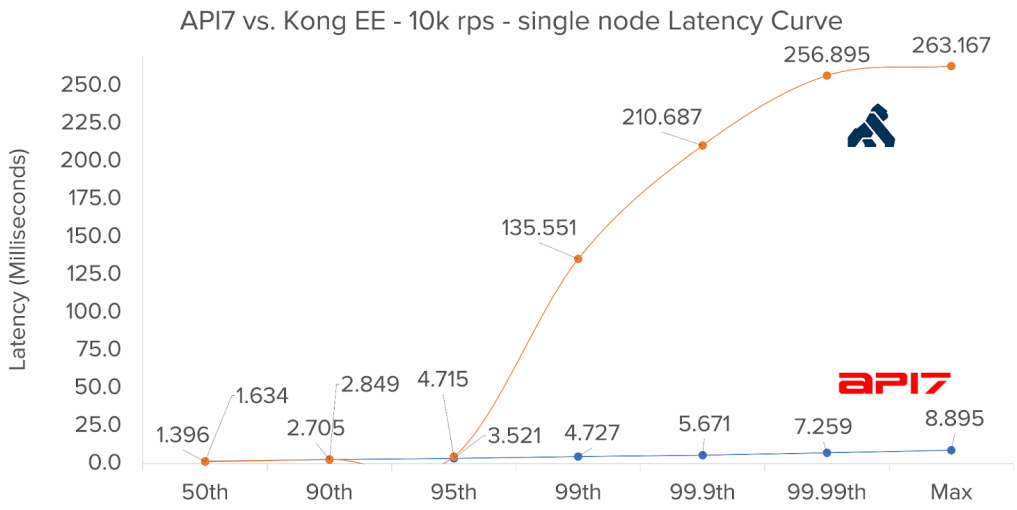

Figure 2. Baseline Latencies of Attacking Our Benchmark API Directly

API7 outperforms Kong EE at all attack rates for a single-node setup. First, we highlight the tests we performed at 10,000 rps for a single node, as shown in **Figure 2**. Although the differences are minimal until you get to the 99th percentile, the difference in latency then grows exponentially. From this chart and the one that follows (**Figure 3**) depicting latency at 20,000 rps, you can see the latencies at the 50th, 90th, 95th, 99th, 99.9th, and 99.99th percentiles and the maximum latency seen during the test runs. At the 99.99th percentile, API7 produced latency that was 97% lower than Kong EE.

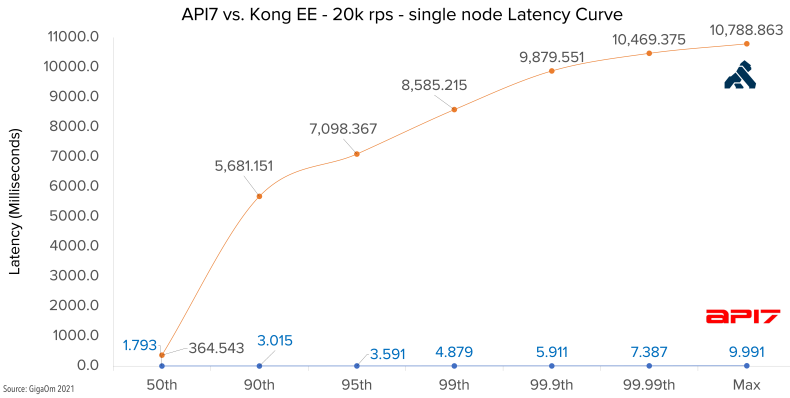

Figure 3. API7 vs. Kong EE at 20,000 rps

Note: Kong EE had 1876, or 9.4%, non-2xx or 3xx responses (failure).

Next, we tested the same configuration with JWT authentication employed, as shown in **Figure 4**.

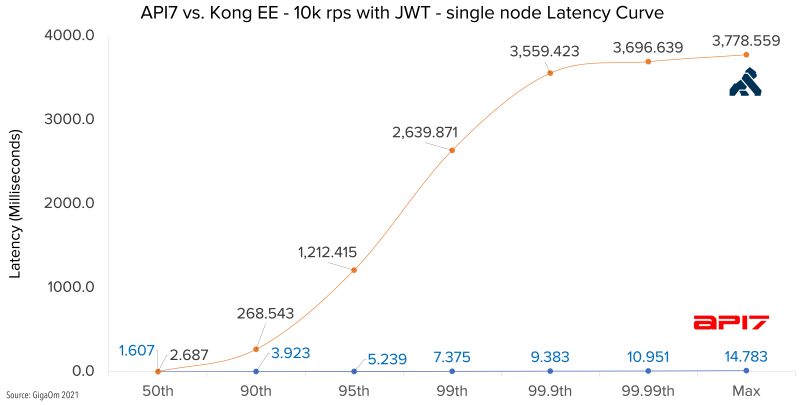

Figure 4. API7 vs. Kong EE at 10,000 rps with JWT

Note: Kong EE had 606, or 6%, non-2xx or 3xx responses (failure).

Third, we tested 1,000 identical routes (without JWT), as shown in **Figure 5**.

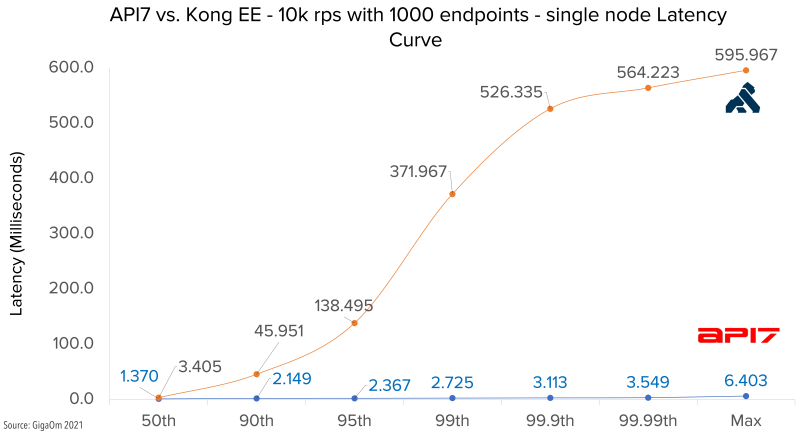

Figure 5. API7 vs. Kong EE at 10,000 rps over 1,000 Routes

5. Conclusion

This report outlines the results from a GigaOm API Workload Field Test.

API7 outperformed Kong EE at all attack rates for our single-node setup. API7 had almost 14 times lower latency than Kong EE at the 99.99th percentile at 10,000 requests per second. The latencies for API7 and Kong EE tended to diverge at higher percentiles. The difference in latency is pronounced at the 99.9th and 99.99th percentiles and at the maximum latency in all our runs.

At 20,000 requests per second, the divergence happened at the 90th percentile, with Kong EE experiencing 5,681 milliseconds of latency compared to API7's 3 milliseconds.

JWT authentication naturally increased latency but did not change the competitive profile much with Kong EE demonstrating, for example, a maximum latency of 3,778 milliseconds compared to API7's 14 milliseconds.

Using 1,000 endpoints also did not change the competitive profile. You can expect single-digit millisecond latencies for API7 across the board with 10,000 requests per second and 1,000 endpoints.

For this test using these particular workloads with these particular configurations, API requests came back with the lowest latencies and highest throughput on API7 rather than Kong EE.

Keep in mind that optimizations on all platforms would be possible as the offerings evolve or internal tests point to different configurations.

6. Disclaimer

Performance is important but it is only one criterion for selecting a high-performance API Management platform selection. This test is a point-in-time check into specific performance. There are numerous other factors to consider when selecting, including administration, features and functionality, workload management, user interface, scalability, vendor, reliability, and many others. It is also our experience that performance changes over time and is competitively different for different workloads. Moreover, a performance leader can hit a point of diminishing returns and viable contenders can quickly close the gap.

GigaOm runs all of its performance tests to strict ethical standards. The results of the report are the objective results of the application of load tests to the simulations described in the report. The report clearly defines the selected criteria and process used to establish the field test. The report also clearly states the tools and workloads used. Readers are left to determine for themselves how to qualify the information for their individual needs. The report does not make any claim regarding the third-party certification and presents the objective results received from the application of the process to the criteria as described in the report. The report strictly measures performance and does not purport to evaluate other factors that potential customers may find relevant when making a purchase decision.

This is a sponsored report. API7 chose the competitors and the test, and the API7 Plus configuration was the default provisioned by API7. GigaOm chose the most compatible configurations for Kong Enterprise as is out-of-the-box, and ran the testing workloads. Choosing compatible configurations is subject to judgment. We have attempted to describe our decisions in this paper.

7. Appendix

The backend API used in this test was deployed on NGINX using the following configuration:

The application works by NGINX listening for GET requests, such as:

```
    GET http://fqdn-or-ip-address:8080/
```

The API would respond with a string of 1,024 pseudorandom Unicode characters from /dev/urandom, such as:

```
    taZ3psgHkQnwohs…
```

The following is the NGINX configuration for the backend API, which you are free to use and modify at your own discretion. GigaOm makes no warranty or claim for its use beyond the scope of this test or report.

```
master_process on;
worker_processes auto;
worker_cpu_affinity auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;
worker_rlimit_nofile 20480;

events {
   accept_mutex off;
   worker_connections 10620;
}

http {
   access_log off;
   server_tokens off;
   keepalive_requests 10000;
   tcp_nodelay on;

   server {
       listen 1980 reuseport;
       location / {
           return 200 "
jduukjyvpeamuzmyxwoqchstlbcqdrurkvejsmsulrjezjtqnvvpaygrtfizfevcxnlrikzrbcjxwlhwm
ygdhslcauoxoljcsbnbuefblbaddnlxixoujckovkzrjijgwzkycsolwdasnmanrjiwfenhrxauhzzgll
bijpvbnvpxuqvedmniqlwflgaaioceninrlsrkngpsfhfhagudbiphxrvbwidxsjfwxkzcbdqkdgnwmvt
lkeuddebdxlrifsgccrocknvypkhepgwhsmskuymkzgbqiocnrlggdsodzqztzsbcbkuwsvdscexovpbw
gdmnkwxgfdxltbvcxmvpeyursouqfmqiuuzfzvonikxqqkqdvmpmnvnzeqencvjqpnmdclbnmsiajdwcl
ywglrrjlbzavttnafyhfspdbtecjrblankkivmqhqcxxfwyqblirxuxhtzrytoanvtqeliujysfsvjbtp
twcrbruclamdtzgpjxpdvyhxckqfsdzbuyyhjwsjpiviyxhjkllzknrywuqogrppxkamifiukuexpsiea
xzvxwcvmpxuohzpmhrjdadrphdkpfvosfhbjskdgemvvzuvgkgsxclygwrazjsqfetpsagyvnfvwkqrqv
zdexalnujfibftcraznitxnajvutmxzabzgxhjoniicafdlhgmuagbhstwfuxlhtuwkuemsyaxhkqsrxo
```

```
mnsylybxnehpeklqpawfpawflptbmscjaplnzfxifmljzufacxwnkqrfgypvjzoqapmnfleihooqcuoaz
opsiprflfybvpbzwhxqehnkuwemijosnclupjaxhjotmkinqdxaijzvvyvlnwswtyjgxqrzdxxhywoshx
saxwcxpunyamcsphyptcparakjoeakigwgtncviswvugtdmeyclpxtrnshzdynsvyamknmmmohsibafuq
nweonfzvwpjjqcmibtmuycgaygiarqqtketpovfxilhsyvkouxbe";
        }
    }
}
```

The following is the NGINX configuration for the load balancer:

```
master_process on;
worker_processes auto;
worker_cpu_affinity auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;
worker_rlimit_nofile 20480;
```

```
events {
accept_mutex off;
worker_connections 10620;
}

http {
upstream backend {
server 172.31.7.42:8000;
server 172.31.9.201:8000;
}

server {
listen 80;

location / {
proxy_pass http://backend;

}
}
}
```

8. About API7

API7 is built by the Shenzhen Zhiliu Technology Co. Shenzhen Zhiliu Technology is an open source infrastructure software company providing API processing and analytics with products and solutions for microservices and real-time traffic processing, such as an API gateway, a k8s ingress controller, and Service Mesh. We are committed to managing and visualizing business-critical traffic such as APIs and microservices for global enterprises, accelerating business decisions and driving digital transformation through big data and artificial intelligence (AI).

Underlying API7 is APISIX, which was donated by Shenzhen Zhiliu Technology to the Apache Software Foundation. It is a new generation of cloud-native API gateway that provides rich traffic management features such as load balancing, dynamic upstream, grayscale publishing, service meltdown, authentication, observability, and so forth.

API7 includes multi-cluster management, a multi-work partition, authority management, version management, auditing, statistical reporting, and other enterprise products to meet the core needs of enterprise users.

Hundreds of enterprise users worldwide are already using Apache APISIX to handle core business traffic, including financial, Internet, manufacturing, retail, carrier, and many other industries. Learn more about API7 at https://www.apiseven.com/en.

9 About William McKnight

William McKnight is a former Fortune 50 technology executive and database engineer. An Ernst & Young Entrepreneur of the Year finalist and frequent best practices judge, he helps enterprise clients with action plans, architectures, strategies, and technology tools to manage information.

Currently, William is an analyst for GigaOm Research who takes corporate information and turns it into a bottom-line-enhancing asset. He has worked with Dong Energy, France Telecom, Pfizer, Samba Bank, ScotiaBank, Teva Pharmaceuticals, and Verizon, among many others. William focuses on delivering business value and solving business problems utilizing proven approaches in information management.

10 About Jake Dolezal

Jake Dolezal is a contributing analyst at GigaOm. He has two decades of experience in the information management field, with expertise in analytics, data warehousing, master data management, data governance, business intelligence, statistics, data modeling and integration, and visualization. Jake has solved technical problems across a broad range of industries, including healthcare, education, government, manufacturing, engineering, hospitality, and restaurants. He has a doctorate in information management from Syracuse University.

11. About GigaOm

GigaOm provides technical, operational, and business advice for IT's strategic digital enterprise and business initiatives. Enterprise business leaders, CIOs, and technology organizations partner with GigaOm for practical, actionable, strategic, and visionary advice for modernizing and transforming their business. GigaOm's advice empowers enterprises to successfully compete in an increasingly complicated business atmosphere that requires a solid understanding of constantly changing customer demands.

GigaOm works directly with enterprises both inside and outside of the IT organization to apply proven research and methodologies designed to avoid pitfalls and roadblocks while balancing risk and innovation. Research methodologies include but are not limited to adoption and benchmarking surveys, use cases, interviews, ROI/TCO, market landscapes, strategic trends, and technical benchmarks. Our analysts possess 20+ years of experience advising a spectrum of clients from early adopters to mainstream enterprises.

GigaOm's perspective is that of the unbiased enterprise practitioner. Through this perspective, GigaOm connects with engaged and loyal subscribers on a deep and meaningful level.

12. Copyright