okta

An Insider Look: How Okta Builds and Runs Scalable Infrastructure

Okta Inc.
100 First Street
San Francisco, CA 94105

info@okta.com
1-888-722-7871

Building scalability, security, and reliability

Companies have always been defined by the types of products and services they provide. But now, across sectors, there's a common thread that ties them all together: the need to deliver technology-enabled solutions.

Whether they're providing a solution for employees, customers, contractors, or partners, organizations have to manage a suite of tools and applications—and the infrastructure that supports them. But they shouldn't have to build it all themselves. From the growing infrastructure-as-a-service (IaaS) and software-as-a-service (SaaS) space, companies can now select which parts of their architecture must be built in-house and which should be offloaded to a vendor so that they can better focus on their own core competencies.

At Okta, this is something we think about all the time. In this whitepaper, we'll use the lenses of scalability, security, and reliability to discuss where we have opted to rely on other providers and vendors. We'll cover how that's supported us in building our services. We'll also address how we've architected our solution and the investments we've made to remain agile and efficient to better serve our customers. We hope this look under the hood provides insights that help you make decisions about what's core to serving your customers.

The building blocks of scalable infrastructure

Over the last few years, our approach to scaling infrastructure at Okta has focused primarily on investments in horizontal scaling and a cell-based architecture. Within this framework, each cell is a combination of components and services that can communicate with each other. At Okta, every cell is an isolated, shared-nothing, identical replica of our infrastructure, spanning from the bottom layer all the way to our edge.

This approach has led to a number of benefits. It's allowed us to minimize the impact of disruptions, as an issue in one cell can be contained therein. And building this architecture across AWS regions allows us to offer customers cells in multiple geographies including North America, Europe, or APAC.

As Okta's service has grown to handle exponentially more requests, we've rolled out cells linearly. This means that over time the capacity of a single cell continues to expand to meet customer demands—a direct result of scaling horizontally with tools such as Elasticsearch, Kinesis, ProxySQL, Redis, and Storm.

How we've scaled our infrastructure

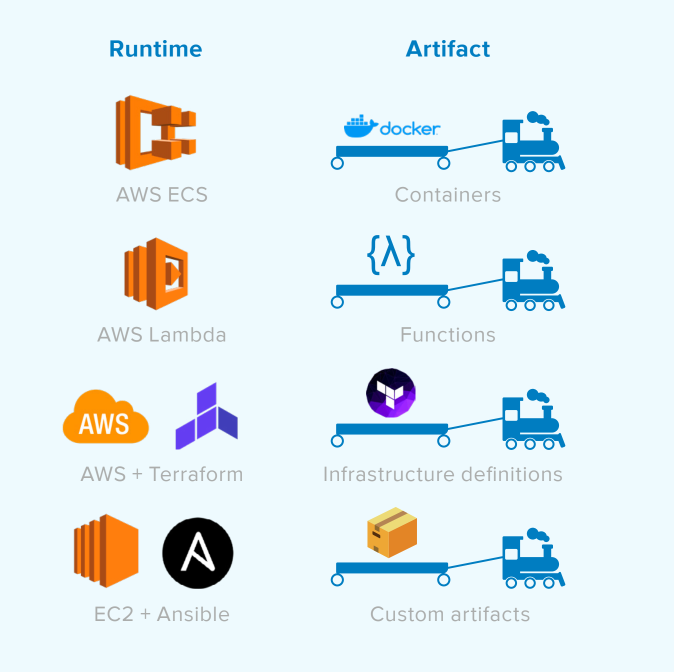
Using cell-based architecture as a foundation, we've been able to effectively enhance and scale a number of processes that span across our cells:

- **Rate limiting:** In the early days, rate limiting was a consideration for service protection and fairness. However, QA tests and approvals for rate limit increases were cumbersome and costly on a per customer basis. To address this, we've implemented continual monitoring to raise overly protective limits and added graceful bursting capabilities when we have the capacity. Though we've always provided headers to let customers know how close they are to their limits, we're currently working to offer better early-warning mechanisms to customers who are at risk of running out of capacity.

- **Capacity planning:** For systems that don't scale automatically, we've matured a capacity planning function to routinely look at key usage and capacity statistics and adjust accordingly. To ensure consistency across cells and allow for comparability when load testing, cells are shaped into one of a small number of predefined sizes. By making this a first-class function, we reduce costs by filling our unused capacity intelligently and free up engineers from potential last-minute one-offs.

Driving release trains

One of the challenges we've faced with cell-based architecture is ensuring consistency across its many environments. While our deployments were automated, each one had to be launched individually for each cell in a serialized approach, and that took time away from our engineers. To address this issue, we've developed tooling which models complete release trains. This ensures artifacts get to every environment in a consistent way, while allowing for parallelism and rollback on failure. With this system, teams can get a high-level view of their artifact's progress through production without needing to take any incremental action per environment.



We started by containerizing services in Docker and using the AWS Elastic Container Service (ECS) scheduler as our new runtime for services. We then built a tool we call Conductor around Amazon's Simple Workflow Service to model the state machine of a release train. With this in place, DevOps teams can upgrade our underlying machines or resize cells while developers release new artifacts and features across our runtime—without having to coordinate. These processes no longer have to operate in lockstep with one another. This is our model across the majority of our services, and we've also developed release trains for AWS Lambda and Terraform changes.

We recognize that some environments aren't easily containerized. With this in mind, we enabled Conductor to call out to Ansible and other existing tooling to handle visiting every environment. Developers can also use a YAML document to declaratively describe the steps of their deployment, add approval steps, and indicate the need for test automations.

The role of security in scalable architecture

Security is a vital layer of any scalable infrastructure, but it's constantly evolving. What could be considered secure today might not be tomorrow. At Okta, we're constantly thinking about how to keep every layer of our architecture—from our internal systems to the devices accessing them—up to date from a security standpoint, so our customers don't have to. We've gone to work for our customers, upgrading and refining our architecture's security measures and safely rolling out these changes:

Upgrading TLS

In 2018, we upgraded our infrastructure from Transport Layer Security (TLS) 1.1 to TLS 1.2. As most of our products and services were using TLS 1.1, this required taking inventory, confirming they were all compatible with the upgrade, and conducting rigorous QA checks.

Knowing that the update wouldn't be supported by some of our customer's clients, we collaborated with our Customer Success and Support team to provide robust documentation on how to make the necessary migrations and supported customers throughout the transition. Rather than blindly cutting over traffic on a certain date, what followed was months of reviewing Okta traffic and migrating customers from old edge machines to new ones once we saw all their clients accepted TLS 1.2.

To build stronger edge security, businesses need insight into the various layers of the network so they can determine threat vectors and protect against them. Two ways to gain this visibility include:

AWS Application Load Balances (ALB) Offers additional DDoS protection with Advanced Shield	**AWS Web Application Firewall (WAF)** Blocks traffic before it even reaches the service

To offload future security upgrades, Okta now runs AWS Application Load Balancers (ALB) in front of our product. In addition to moving this security patching concern to our vendor, we picked up additional security and DDoS protection by using AWS Web Application Firewall and AWS Advanced Shield, which require ALBs to function. Moving our edge to our vendor also provides cost savings as some forms of abuse can now be blocked before ever reaching our servers.

Evolving our approach to patching

Threats like Spectre and Meltdown show us there is always potential exposure to vulnerabilities—and we need to be able to respond fast. One high priority was a process to help us get low-level kernel updates out to our environment as quickly as possible.

In order to automate our patching processes as much as possible, we focused on getting the majority of our infrastructure into the standard Auto Scaling Group and leveraged a model that slowly updates the underlying AMI hosts. This means going to the first canary host, doing some testing, looking at our metrics, and then slowly rolling out the new version of the AMI to the rest of the environment.

The challenge with this approach, however, is that when you have a vast number of applications and microservices, you need insight into the AMI upgrade context as it's tightly coupled to the host. We've addressed this challenge with Docker and containerization: by running in containers on top of Auto Scaling Groups, multiple services can run on the same host infrastructure without the need to coordinate across app teams. Through this process, we also have a scheduler that ensures the right number of machines and containers are running for a given application.

Putting Zero Trust front and center

The theory behind Zero Trust is that we can no longer solely rely on network accessibility to gate access to applications. Instead, more granular controls manage and monitor access to each application and service. In part, this means using a principle of least privilege access, where authenticated users only have access to the resources that correspond to their role or group.

Incorporating Zero Trust into scalable infrastructure comes down to secure principles. This includes:

- Adopting the premise that all actors on a network should not be trusted by default.
- Replacing simple, static credential methods with multi-factor authentication.
- Frequently re-evaluating whether users should be allowed access based on changes in security posture.

To apply these Zero Trust principles to our infrastructure and address server access issues, Okta began dogfooding it's own Advanced Server Access (ASA) product, which got rid of long-lived static keys. This allowed Okta to scope credentials to individual requests, implement continuous authentication across users, devices, and sessions, and create seamless experiences with extendable APIs.

By combining Okta Access Policy with ASA, we can now model access policies to servers which were never possible before. For example, compliance frameworks like FedRAMP require users to be a U.S. citizen on U.S. soil when accessing a server. While this is impossible to model with long-lived keys, ASA's ephemeral keys combined with a US only access policy allow us to fully model this FedRAMP access requirement in Okta.

Learn more about how Okta can help you adopt a Zero Trust security model through its own offerings and integrations with other security providers.

Scalable infrastructure that's reliable

Providing reliable services and applications is key to our customer trust and success as a business—we keep reliability at the core of our operations and our culture:

- **Being proactive:** To understand how an incident occured, and stop it from happening again, businesses need to determine the root cause. We have implemented consistent logging; examine the resiliency of the system within the test environment; and conduct consistent root cause analyses for each and every incident.
- **Making architecture highly available:** This means ensuring that services aren't geographically dependent and are never taken offline for upgrades. Employing this strategy means that services remain untouched in the event that a third party suffers an outage.
- **Ensuring high quality:** Development teams need confidence in an artifact's ability to move between environments. At Okta, we've ensured this with synthetic transactions, which enable you to walk through a system in the same way that a user would to identify potential issues. We also use models that identify and execute tests for critical flows that run continuously across every environment. This constant monitoring is done from multiple distinct locations around the world to ensure that the functionality works on different networks and that performance is acceptable.

At Okta, these three tenets are at the core of many of our processes, allowing us to build a reliable and available infrastructure that operates at a high quality. Here's how we've put them into practice.

Cops and robbers

This approach to improving resiliency sets up Okta's test environment, and then allows "robbers" to go in and try to destroy things. At the same time, the "cops" watch our monitoring and alerting to assess how the system reacts to different kinds of attacks or outages. When the team launches new features or rolls out new infrastructure, this technique helps us find obscure bugs with complex, distributed systems.

Engineering urgent

Issues are bound to happen—but we can only be defined by how we resolve them. Our #NoMysteries mantra, coined by CTO Hector Aguilar, helps us uncover potential issues, get to their real cause, and ensure they never happen again. This includes in-depth root cause analysis (RCA) processes that help uncover how a problem occurred and how it can be prevented in the future.

How we conduct RCAs depends on the scope of the problem: sometimes we'll work with an individual team impacted by the bug, and in cases where the issue spans across products or our infrastructure, we have broader team discussions. In these meetings, we address the issue, its impact, whether it was disruptive to our customers, and then we put a plan in place to ensure it doesn't recur. Our goal is ultimately to ensure that we're being effectively reactive while also being proactive.

On the back of this, we implement an Engineering Urgent process, which triages issues from the root cause analysis and puts them at the top of team's backlogs. This ensures we close the loop on issues identified in RCAs and prioritize work to harden the system before adding new features and potential risk exposure.

Highly available architecture

At Okta, when we design for high availability, we take a layered approach that allows us to control both low-level IaaS and higher-level platform-as-a-service (PaaS) components. At the foundational layer, Okta's infrastructure is "active-active" across three data centers (AWS availability zones) to ensure resiliency. We follow this model on all low-level infrastructure running on hosts, such as caches, app servers, and databases. In addition, we replicate all data to a separate disaster recovery region.

As you move up the stack and think about PaaS offerings from vendors, you can lose a lot of control over which parts of your infrastructure are highly available. For example Amazon S3 is a single region service, and has had outages of regions in the past. Though S3 added a bucket replication feature, this is asynchronous and only highly available for reads, but not for writes. We've addressed this by augmenting the AWS SDK to do dual writes in super-critical write paths and do read repairs when an object isn't present. This ensures Okta's platform is always highly available even in the rare event of an AWS region outage.

When it comes to DNS, we first used Dyn DNS across three different Dyn regions—all of which were taken down in a massive DDoS attack. This incident led us to change tactics so that we could better manage DNS on behalf of our customers. We took on the process internally and developed a mechanism to propagate information to Dyn and Route 53 as a multi-provider.

Another good example is how we addressed an issue in our Key Management System (KMS), where master keys never left the system. We created a hierarchy whereby the keys that protect tenants' data receive a separate key protected by multiple roots of trust—a KMS in the active region and a KMS in a disaster recovery region and both are run as "active-active." If the application can't talk to one of the KMS regions, it will failover to the other one.

These are a few ways we use an "active-active" strategy in our PaaS layer to mitigate disruptions from our vendors. When KMS went down on the U.S. east coast in 2019, our encryptions and decryptions continued to function, preventing at least a 90-minute downtime for our customers.

Adding scale, security, and reliability to your infrastructure

As we continue to build a robust identity and access management platform for our customers, we always keep in mind the expression "It's turtles all the way down": as our customers rely on Okta, we leverage OSS and AWS, which in turn operates on hardware and software partially provided by vendors and open source. It's our job to ensure our infrastructure remains scalable, secure, and reliable, so our customers can continually get the best levels of service we can offer.

Your World
okta
aws
OSS
Hardware, OSS, etc.

We ensure our customers can continue to offer secure and reliable products and services to their own customers with user experiences that are both highly secure and seamless.

By positioning your technology on the Okta turtle, you can better secure and scale your own infrastructure with solutions like Adaptive Multi-factor Authentication, API Access Management, Advanced Server Access, and more. These offerings are all anchored in the tenets of Zero Trust, ensuring that only the right people have access to the right resources at the right time.

We're constantly investing in tooling and automation to innovate upon our own solutions, so you have the time, resources, and peace of mind to focus on your own core competencies, ensuring faster time to market and future-proofed applications and services that align with the latest security and identity standards.

To discover how we can help your business implement the tools required to build a scalable infrastructure, contact us today.

About Okta

Okta is the leading independent provider of identity for developers and the enterprise. The Okta Identity Cloud securely connects enterprises to their customers, partners, and employees. With deep integrations to over 6,000 applications, the Okta Identity Cloud enables simple and secure access for any user from any device.

Thousands of customers, including 20th Century Fox, Adobe, Dish Networks, Experian, Flex, LinkedIn, and News Corp, trust Okta to help them work faster, boost revenue and stay secure. Okta helps customers fulfill their missions faster by making it safe and easy to use the technologies they need to do their most significant work.

Learn more at: www.okta.com

okta