# Not All Cloud Services Are Built Alike

## Okta's High Availability Architecture

**Okta Inc.**
301 Brannan Street
San Francisco, CA 94107

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

# Contents

# Not All Cloud Services are Built Alike

So, you've found the cloud application of your dreams. It does everything you ever thought you could want and ten things you didn't know you wanted but now can't imagine living without. It took less than 13 seconds to fully configure, and after rolling it out you found that several users had placed pictures of your IT team on their desk next to pictures of their kids and spouses. You feel pretty good about your purchase.

Then it happens...

At 3:30 ET your helpdesk starts lighting up. Your cloud app is serving error pages intermittently and 20 minutes later, the app goes down completely. Users are asking you for answers, the vendor's support page is silent and after sitting on hold for 20 minutes, a tech tells you that they hope to have more information soon. Eight hours later, you get a form email telling you that the service is partially up, but your users' data won't be available for another 8 hours. Twenty-four hours after that, the data returns, you've lost two days of user productivity, and your IT team's picture moves from desks to dart boards.

In the cloud world, you as the IT person hand over control of critical application infrastructure to your partner, expecting that the cloud vendor can manage that infrastructure better and more cheaply that you can.

But not every cloud vendor delivers on this promise, and it's equally important to evaluate each application based on its architecture as well as its feature set.

Now imagine you are evaluating not just one application, but a core cloud infrastructure service that will be the central point through which you secure and manage access to ALL your applications.

An on demand identity & access management service is one of those core cloud infrastructure services and should be:

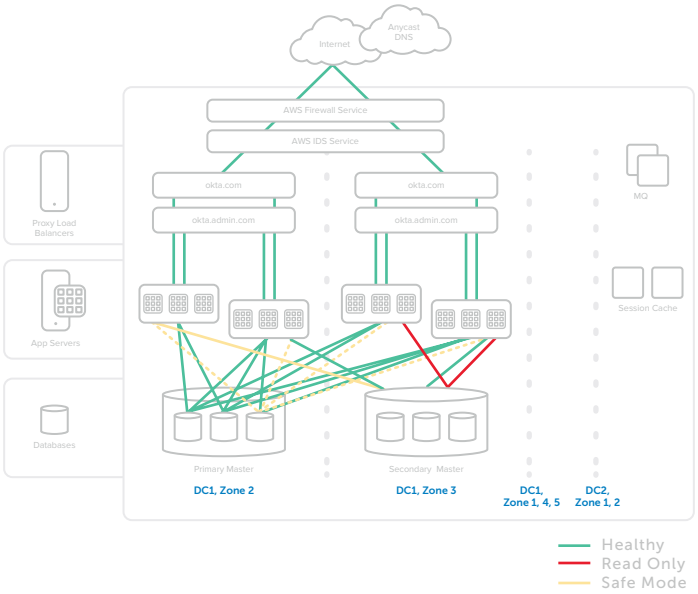
- Built for Web Scale — the service must scale up and down seamlessly with your needs.
- Always Available — the service must be architected for zero downtime. No maintenance windows required.
- Secure — the service must be more secure than anything you could build and operate on your own.
- Constantly Evolving — the service must deliver rapid innovation that enables new capabilities and insulates you from the constantly changing IT landscape.

At Okta we take this to heart and have built the software, operationalized the processes and hired the people it takes to deliver on all of these fronts. This technical whitepaper will provide an overview of the software and operational architecture that enables Okta to run a scalable, highly available, on demand identity and access management service.

# A Cloud Native, Multitenant Architecture

One of the most critical aspects of Okta's architecture is that it is completely multitenant. With multitenancy, all of our customers share the same underlying environment. Because it is shared, Okta can make the infrastructure extremely robust in terms of scale, redundancy, monitoring and processes.

The entire company is focused on making this one environment perfect. The overall picture of the service looks like this:




The system consists of a front end tier containing proxy load balancers (global load balancing is achieved leveraging DNS) and firewall services, an app tier where our software mostly runs, a set of functionally optimized databases. Everything is hosted running in Amazon Web Services (AWS) across multiple availability zones and geographically separated datacenters. The service isdesigned for high scale, high throughput, and 100% availability.

The core design elements of the system are:

## Stateless

All components other than the databases are completely stateless. As a result, above the database tier, any server in the stack can handle any request.

That means that all of the components of our system can be scaled up at will simply by spinning up new VMs in AWS and any individual component can fail at any time and will simply be routed around to one of several other active systems.

## Functionally Optimized Databases

Throughout the system we generally like to use the right tool for the job. When it comes to databases we follow that rule and have segmented the databases based on access patterns and types of data being stored ensuring that we can meet stringent requirements in each dimension without compromising the others.

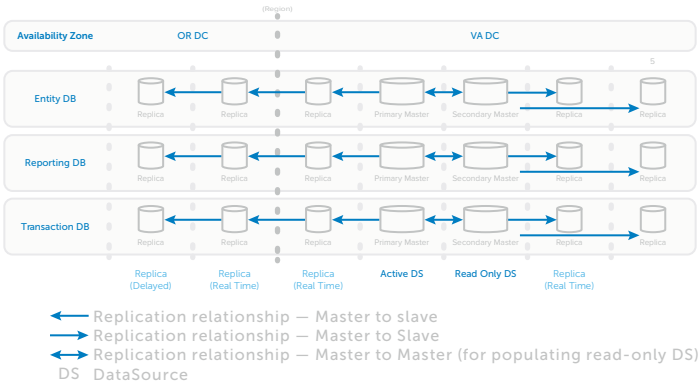
In our case that breaks out as follows:

- Entity Database — It holds extensible configuration and application metadata. Data size and throughput requirements are medium so a flexible schema is needed.
- Session Database — Data tends to be small but it has high throughput requirements so needs to support schema changes without going down.
- Transaction/Logging Database — It has high data size and write throughput requirements with medium read throughput needs.
- Reporting — It has large data size and medium write and read throughput requirements and must be denormalized to support querying on many dimensions.

It's complicated, but because Okta is multitenant, we can do this once and do it right.

## Data Replication and Backup

While it is possible to add and remove stateless components above the data tier at any time that is not true for the database layer. Having built this tier on AWS EC2 this is especially challenging as instances can disappear at any time so we are using extreme replication and incremental backups to compensate for that.




- We run a master-master configuration with read replicas so there is no single point of failure. If one master goes down the other is promoted.
- Replicas are live across six availability zones and we have a time delayed replica in a seventh.
- For further redundancy a full replica of the entire system is running in two availability zones in a geographically separate datacenter.
- For backups we do incremental EBS snapshotting to S3 and take full portable backups in case we need to restore outside of AWS.

## Zero Planned Downtime

Okta must be available for any other application to be accessed, by any user, from any part of the world. There is no good time for the Okta service to be down. With this in mind, we designed Okta for zero planned downtime. Most services try to solve for zero downtime in one of two ways: they either require engineers to write code that can handle reading writing to multiple versions of the DB or they have a read-only mode. The first approach creates a lot of inefficiencies in the code and reduces agility.

We have taken the second approach, having a read-only mode, a step further by supporting a read-only mode at all layers of the stack. This comprehensive approach to read only enables us to both deliver regular updates to the system with minimal impact on administrators and no impact on end users and to prevent the system from going down as a result of an unexpected outage.

## No Maintenance Windows

Traditionally the ability to provide a highly available service is in direct conflict with delivering continuous innovation to that service. At Okta we knew we could not make that compromise. By combining our read only architecture with an automated testing and deployment process we are able to maintain service availability while also delivering continuous innovation. No maintenance windows required.

Any developer commit to our mainline code kicks of a new build and a series of unit, functional and parallelized UI test. Once that fully automated process completes and ensures that the code has passed our rigorous testing the software is ready to be deployed.

The operations team then deploys the fully tested code to our pre-production and production servers. Doing this takes careful orchestration and leverages all of the design elements we have described thus far. Throughout the rolling deployment process we ensure upgraded app servers only talk to upgraded versions of the DB.

The process is:

1. Move the system into read-only mode
   - App servers point to secondary master
   - Apps continue to write to the primary Transaction DB
   - Session DBs remain live (they don't generally change)
   - Replication on the primary master is halted
2. Disable half of the app servers from each load balancer pool, and stop the apps on that half
3. Upgrade the primary master DB to a new schema
4. Upgrade the offline app servers with new code
5. Switch traffic from old code to new by enabling the app servers running new code and disabling the apps running old code Service now live on new code in READ-WRITE mode
6. Enable replication on the primary master DB
7. Perform the trailing upgrade: repeat the process for the now-disabled half of the app servers
8. Restore the secondary master DB replication Throughout the entire process the service continues to process authentications so that access to applications is never impacted

### Ability to Handle the Unexpected

As anyone who's worked in IT knows, there will always be problems that occur that you simply can't plan for. Networks go down, storage fails, software breaks in unexpected ways. Critical cloud services like Okta need to be built and operated with the expectation that these problems will occur and be tolerant of these problems.

The first layer of defense against the unexpected is robust instrumentation and monitoring for all components of the system. We split this into two categories: external monitors and internal monitors.

For external monitoring, Okta uses two third party services with globally distributed test agents to constantly monitor the Okta application. This provides us with a constant feed with real data on how our service is operating. Because we are fully multitenant, the data is real and applicable to all of our customers.

If the monitors say we're up, we're definitely up. If they say we're down or slow, we're definitely slow. Any problem seen from multiple monitors results in a notification to our operations team and an immediate response.

We use internal monitors to show us not only when things are having problems, but more importantly, what is having problems. They are more sensitive than the external monitors, so they frequently give us warning of problems before they affect site performance or availability. Okta uses internal monitors on all subsystems, and instruments all of our software components for maximum visibility.

The last line of defense is still the read-only mode that we mentioned earlier and allows the okta service to stay up even with unplanned disasters strike. For example, even if AWS suffered a multiple availability zone outage where all of our master databases were running, the service would still function running on the read-only replicas in the unaffected availability zones.

## Conclusion

At Okta we adhere to the highest standards for security and reliability in all we do, from our hiring practices to the architecture and development of the software that powers Okta and the data center strategies and operations that enable us to deliver a world-class service. Complementary to these investments is our philosophy to be as transparent with our partners and customers about how we do this. This combination we believe establishes a solid foundation of trust that is critical to a successful long term relationship between Okta and your organization.

Want to learn more? We'd love to hear from you, please email us at: info@okta.com.

## About Okta

Okta is the foundation for secure connections between people and technology. By harnessing the power of the cloud, Okta allows people to access applications on any device at any time, while still enforcing strong security policies. It integrates directly with an organization's existing directories and identity systems, as well as 4,000+ applications. Because Okta runs on an integrated platform, organizations can implement the service quickly at large scale and low total cost. More than 2,500 customers, including Adobe, Allergan, Chiquita, LinkedIn, MGM Resorts International and Western Union, trust Okta to help their organizations work faster, boost revenue and stay secure.

For more information, visit us at www.okta.com or follow us on www.okta.com/blog.