



SCALR

Case Study: PeerJ

The Challenge

Early on, PeerJ made the decision to adopt cloud in order to improve cost effectiveness, availability, and development velocity. After some research, PeerJ settled on the market leader: Amazon Web Services.

The PeerJ team understood that cloud infrastructure is in fact very different from traditional infrastructure, and that they would have to adopt new cloud-native development practices if they were to reap the benefits of their investment in cloud:

- For a given configuration, cloud infrastructure is more expensive than traditional infrastructure. However, autoscaling would let PeerJ leverage the elasticity of Amazon Web Services, and secure substantial cost savings.
- Cloud infrastructure has higher failure rates than traditional infrastructure, and hardware maintenance sometimes results in early termination of cloud instances. However, automated infrastructure provisioning and failure recovery would help PeerJ maintain high performance and availability for its service.

The PeerJ team realized that designing cloud-native applications that scale dynamically and withstand failure would represent a significant time investment and redirect engineering effort away from application development. In turn, this would decrease developer productivity and significantly reduce the company's return on its investment in cloud.

As a result, PeerJ started looking for tooling that would facilitate the deployment of cloud-native applications and let PeerJ engineers focus on building the business' applications. This effort was led by Jason Hoyt and Patrick McAndrew.

The Solution

Autoscaling

Using Scalr, PeerJ is able to dynamically scale its infrastructure to respond to traffic peaks and lulls, thus minimizing its infrastructure costs.

There were two distinct challenges that PeerJ needed to address to succeed with autoscaling and Scalr has helped the company solve both.

First, load must be measured and reported so that scaling decisions can be taken. To do so, PeerJ relies on Scalr's data collection capabilities. Using a locally installed agent, Scalr monitors operating-system-level metrics and application-level metrics.

PeerJ

About PeerJ

PeerJ is an Open Access publisher with the mission to efficiently publish the world's scholarly knowledge. PeerJ has received seed investment from Tim O'Reilly's *O'Reilly AlphaTech Ventures*, and the company's paying customer base includes prestigious institutions such as Stanford, UC Berkeley, and Cambridge.

"Scalr is an opinionated, high-level framework that encourages and helps us build applications that perform optimally on cloud infrastructure. As a result, we feel that Scalr has helped us use our cloud resources better, and secure a greater return on our investment in AWS."

— Patrick McAndrew, Senior DevOps Engineer at PeerJ

Second, infrastructure must be coordinated. For example, load balancers need to be notified when new application servers are added, so that those can start serving user traffic; likewise, they must be notified when application servers are decommissioned, so that user traffic can be safely redirected away. To coordinate its infrastructure, PeerJ leverages Scalr's messaging and automation infrastructure: the Orchestration Engine.

Lifecycle Management and Disaster Recovery

PeerJ uses Scalr to deploy its infrastructure and ensure that it is maintained in a healthy state in spite of instances becoming unstable or unavailable. As a result, PeerJ engineers are able to focus on developing the company's applications and let Scalr handle its lifecycle.

To automate the deployment of application code to its cloud infrastructure, PeerJ integrated Scalr's Orchestration Engine with the Puppet configuration management solution. In addition, PeerJ used the Orchestration Engine to integrate Scalr with its Nagios-based monitoring infrastructure, which helps PeerJ engineers keep tabs on their infrastructure.

As for disaster recovery, PeerJ relies on Scalr's infrastructure management capabilities. Out of the box, Scalr ensures that unhealthy or unstable instances are rapidly replaced and decommissioned. For critical services such as MySQL, Scalr's Orchestration Engine can perform failovers in order to minimize downtime in the event of a failure.

Conclusion

For the PeerJ team, adopting Scalr was beneficial in two ways. First, Scalr has helped maximize the return on PeerJ's investment in cloud. Second, it has helped PeerJ's engineers focus on building their business, and not on managing their cloud infrastructure.

Using Scalr, PeerJ was able to support two digit growth in transactions over the year 2013, and is gearing up to achieve similar performance in 2014.

As Hoyt puts it:

"Thanks to Scalr, we've been able to focus on what matters to our business: building PeerJ. Naturally, infrastructure management is critical, but it's definitely not our core business. Scalr's Cloud Management software has been a great help in automating that aspect of our operations."

— Jason Hoyt, Ph.D., Co-founder and CEO, PeerJ

"Using Scalr, we are able to focus on developing our product and can safely hand over its lifecycle management to Scalr. We are saving a lot of time, which lets us take our product to market faster."

— Patrick McAndrew, Senior DevOps Engineer, PeerJ

About Scalr

The Scalr cloud management platform enables today's enterprise to manage and control accelerated application development across public, private and multi-cloud environments through an enterprise grade, on-premise software solution. The Scalr platform elegantly automates the deployment, monitoring and governance of cloud computing environments. Founded in 2007, leading global organizations have selected the Scalr platform, including Samsung, Nokia, Oracle, Expedia and the Walt Disney Company.



www.scalr.com