VIDERITY

STRATEGIC, CREATIVE, TECHNICAL

info@viderity.com 800.690.9170

Viderity is an IT firm that provides DevOps and cloud services. We help companies build better software and shorten the time from development to production by leaving manual processes behind. We embrace an automated, collaborative, & agile way of working to increase your organization's speed, reliability, and efficiency at scale.

We help guide your transition to the cloud by architecting & implementing custom enterprise grade cloud solutions. We engineer each solution to be flexible, fully scalable & highly available. We also enable complete management & automated deployment of your apps through custom CI/CD pipeline development & configuration management tools while providing 24/7 managed support.

SERVICES

- Infrastructure Auditing
- Infrastructure Design & Strategy
- Kubernetes (Auditing, Implementation, Support,Management)
- Cloud Migration
- Configuration Management (Chef, Puppet, Ansible, Salt)
- CI/CD Pipeline Development (Jenkins, Gitlab)
- Security Auditing (AppSec, NetSec)
- Cloud Cost Optimization
- Big Data (Hadoop, Spark, Hive, HDFS)
- Streaming (Kafka, Queuing)
- Training

SERVICES PLATFORMS

Amazon Web Services
(AWS Standard Partner)
Google Cloud Platform
Azure
On-Premise

Hybrid
Openstack
Openshift
Nutanix
VMware

DEVOPS CLIENTS

GOVERNMENT

BBG (Broadcasting Board of Governors)
DOL (Department of Labor)
Foundation for the National Archives
NASA (National Aeronautics and Space Administration)
NARA (National Archives and Records Administration)
NIST (National Institute of Standards and Technology)
NIH (National Institutes of Health)
NSF (National Science Foundation)
Presidential Libraries
Smithsonian Institution
USAID (U.S. Agency for International Development)
United States Patent and Trademark Office (USPTO)
The World Bank Group

COMMERCIAL AND NON-PROFIT

Comcast
eBay
GXS
Hitachi
Kraft Foods
NASD
National Breast Cancer Foundation
NBCUniversal
Network Solutions
Parkway Corp
PBS Foundation
Rideshare
Penn (University of Pennsylvania)
Sony
Stackston
Uncommon Goods

Case Study: Streamlining DevOps Principals through Automation

Business Challenge

A nationwide bill payment service provider that's been helping hundreds of thousands of people pay off their loans faster, simplify budgeting and pay bills more conveniently was undergoing a significant platform upgrade. The client migrated their JVM based payment application to a modular containerized service based architecture built with Spring Boot and needed to adopt modern DevOps infrastructure principals to quickly scale and deploy services.

DevOps Transformation

Scalability was critical because our client needed to be able to elastically scale its services without consuming precious engineering resources. Our client runs its entire cloud-based IT infrastructure on Amazon Web Services (AWS). As a startup, they we're very meticulous about where we invested our time. We focused on automation through continuous delivery and leveraging open-source tools such as Jenkins, Chef, Packer and Docker. We didn't want to reinvent solutions to already-solved foundational infrastructure problems so we built and maintained all environments through our custom AWS CloudFormation templates. This gave us the luxury of version-controlling their stack including but not limited to network, security groups, and IAM policies. This allowed for seamless, exact network duplication for on-demand development and staging environments. The creation of environment resources is completely automated, so developers could focus on software rather than managing servers.

Business Challenge Story

Our DevOps practice philosophy is about continuous delivery: building, testing, and releasing far more frequently, and ensuring that our client can release reliably at any time. As a result of transforming their processes, everyone became an owner. Everyone was responsible for the end goal. Developers now have a commitment to higher quality code, and infrastructure and operations felt responsible for incidents and ensuring that they close the loop. There are no more silos of information and work, and teams readily share successes and failures with one another. Streamlining the DevOps process teams are more autonomous. DevOps processes that are supported with Docker enable developers to produce self- contained applications that are delivered in a fast, repeatable way. Using Docker containers with Chef allows us to an infrastructure-as-code approach, making management and deployment transparent, seamless, and fast.

Results

Using containers and a microservice architecture together with automation enhanced their cloud capabilities. Microservices are scalable and reusable, while containers supply efficient resource management. Both microservices and containers can work independently, but it has become clear that merging them has improved runtime frequency, deployments and overall application efficiency.

> "DevOps practices enabled development efficiency"

Case Study: Using Mature DevOps and Container Orchestration to Deploy Applications Worldwide

Business Challenge

A worldwide insurance platform growing their operations wanted to make it easy for their engineers to ship applications from dev through a mature CI/CD pipeline into production. They desired to containerize all their applications for parity between environments, so that they could meet the scaling challenges that were quickly coming their way. Multi-region resiliency, role based access controls, and other international regulatory restrictions were all required.

DevOps Transformation

The engineering team is distributed worldwide so communication and collaboration between them is critical. Operations had to be automated from the time an engineer checked in code and made a pull request to the time it was deployed into production. We leveraged containerization tools such as Docker and Kubernetes to set up their infrastructure on Amazon Web Services. This allowed the team to deploy an account per environment, with a management account that oversees all operations between them all, as well as monitors, aggregates logs, and runs a Chef and CI/CD server. Each environment contains a Kubernetes cluster, with various supporting services running alongside the teams software.

Chef is utilized to provision the various resources in Amazon Web Services, from a central OpsWorks-Managed Chef Server in the management account. Basic cookbooks provision VPNs, and highly available Etcd backends, masters and nodes to ensure resiliency.

Open source tools like Jenkins, Elasticsearch, and Sensu are leveraged.

Results

Environments are all created via AWS Cloudformation and Chef Cookbooks, and control the standing up and maintenance of VPCs, VPNs, monitoring stacks, and Kubernetes clusters.

Applications are written and checked into Version Control. Once changes are detected, Jenkins on Kubernetes starts the build process, which creates a Docker image and ships it to the EC2 Container Registry. Developers can run their applications within the Regional Sandbox, which is free for all to use, with built in Guard Rails. Additionally, a User Acceptance Testing environment runs a Kubernetes install and creates a version of the application for Quality Assurance. Once tests pass and the team is satisfied, the Helm updates the application running in production. Regressions are captured in the pipeline and stopped at the door prior to getting to production.

Monitoring the applications allows engineers the ability to instantly detect issues with their deployments and alert the team. Binary monitoring is through Sensu, with metrics being handled by Prometheus. Logs are aggregated with Fluentd and shipped to Elasticsearch and S3.

The engineering team is able to Build, Test, Deploy, and Monitor their work without having to be involved in the infrastructure at all. This is because the tooling allows them to access the items they need to ensure the applications are behaving correctly and consistently performant.

Standing up new regions is a process that takes less than an hour, and delivers a new VPC, VPN if needed and Kubernetes cluster with various items already stood up inside of it, ready to be used. Teams with access can then start using it immediately, because it mirrors the existing environments.

Conclusion

The team is able to ship dozens of applications around into different environments during development & testing and efficiently deploy them to the cloud at scale, worldwide. The team is not concerned about managing different machine types anymore, instead spending their time delivering features.