



EBOOK

IT Perspectives: Approach to DevOps Platform





Table of Contents

Introduction >>

Make DevOps Even More Effective: Build a DevOps Platform >>

Three-Part Architecture of our DevOps Platform >>

Application Strategy and Achieving Target State Architecture >>

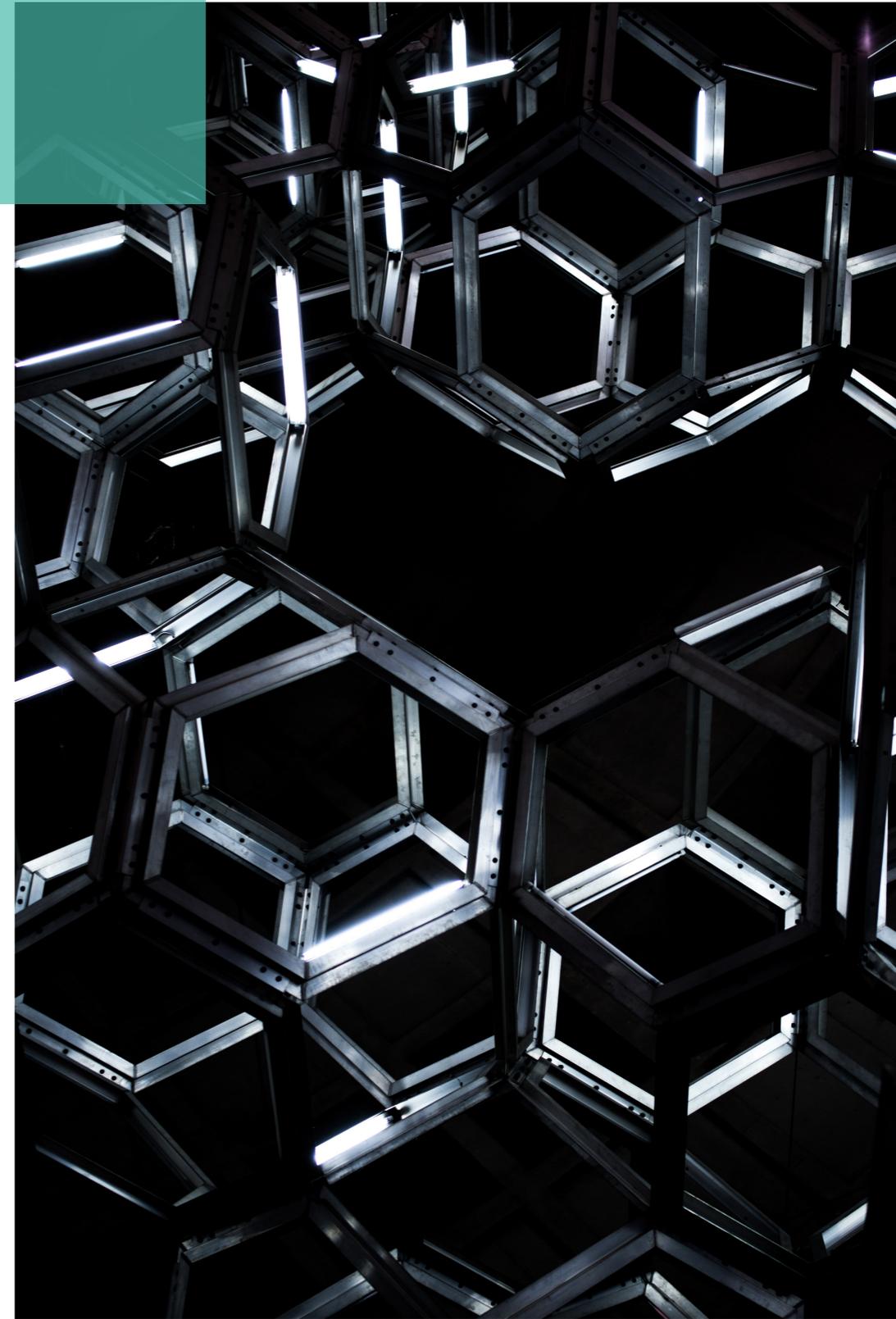
IT Operational Support >>

Building in Security by Default >>

Introduction

Traditional approaches and governance methods often impede DevOps as software developers want—and are rewarded for—speedy change delivery, new features, new technology and innovation. On the other hand, Operations wants—and is rewarded for—stability, predictability, control, and above all, non-disruptive operations. With a DevOps approach, you must break out of the traditional mode and get the developers and ops staffs to work collaboratively on projects from start to finish. To do this, we took advantage of a greenfield opportunity to build a platform with an infinite set of developer tools, platform software, and infrastructure.

This eBook covers how we approached integrating leading third-party tools along with NetApp technologies to increase Dev and Ops productivity across six essential capabilities: CI/CD, code and binary management, containers, automation, cloud and PaaS, and monitoring.



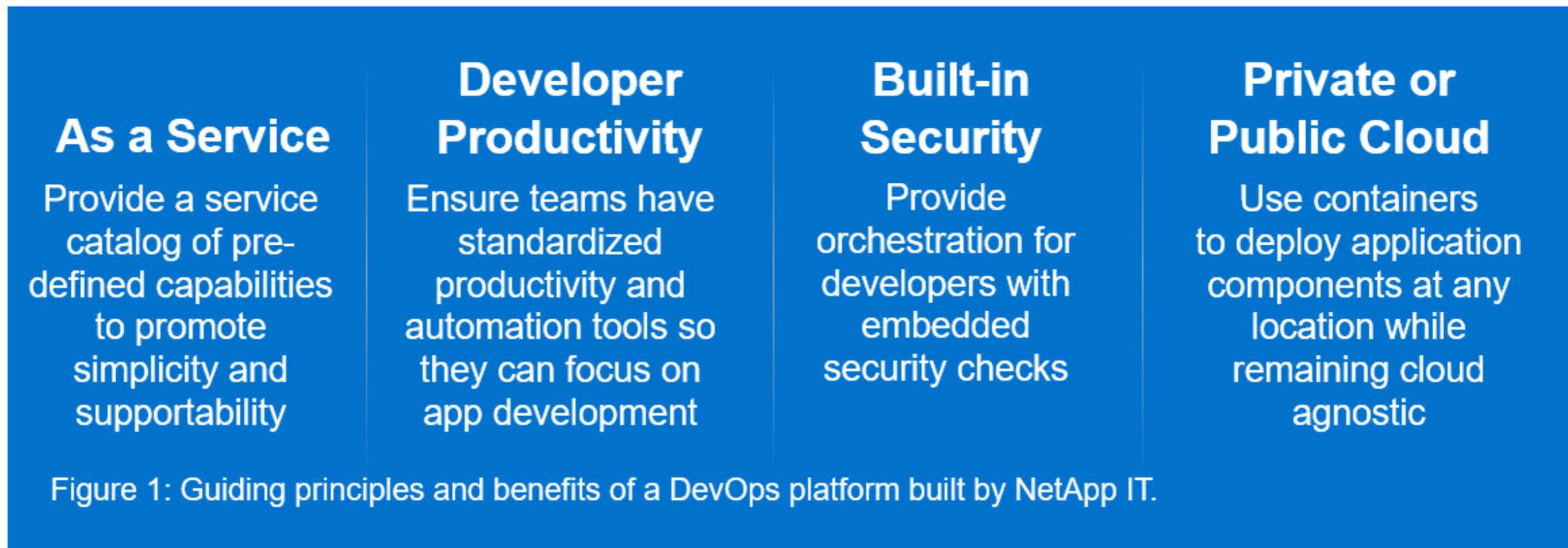


Make DevOps Even More Effective: Build a DevOps Platform

JEFF BONI
VICE PRESIDENT, IT FOUNDATIONAL SERVICES AND CUSTOMER-1

Many IT organizations are implementing DevOps practices to realize the benefits that companies like Google, Microsoft, Facebook and others have been experiencing for years. They want to deliver features faster, improve the stability of applications, and move from a reactive delivery organization to one of innovation. No doubt these are great benefits, as you cannot reach the full potential of DevOps if you only focus on getting the practice going.

Many organizations have DevOps practices split across multiple teams with each one having their own tools, applications, and processes. While each team is effective within its own microcosm, you risk losing productivity when developers move among teams or cross-team collaboration is needed. Development is interrupted as new team members adjust. It's also not uncommon for developers to build their own tools and applications.



Our approach

To address these pitfalls and reduce interruptions as developers move between projects, we built a standardized DevOps platform with a finite set of developer tools, platform software, and infrastructure. The platform automatically provides the elements that often distract developers while offering consistency across projects. Overly simple right? Well, yes and no. Since we recently started our practice, we had a greenfield opportunity to build a platform and to influence practices.

We set our own practices, process, and tools which would meet the needs of our current DevOps maturity.

The hard part is delivering a single platform that can serve different development teams, especially with the [plethora of available tools and technologies](#). There is no one single tool or technology that provides a full DevOps solution.

The first release of our DevOps platform, called CloudOne, happened in November of 2018. CloudOne provides the cloud services, automation, and CI/CD release models that our application development teams need to build cloud native applications. As shown in Figure 1, we have four guiding principles and benefits.

As a service

It is important that the platform be viewed and implemented as a service. This means it is defined by a solid value proposition for your developers, driven by their requirements, and continually enhanced and improved. As a cloud agnostic service, CloudOne can deploy environments both on premises and off prem.

- A service catalog of application stacks including Angular.js, Envoy, and OpenResty, with more to come.
- Automatic updates to the Configuration Management Database (CMDB) with what is being developed.
- Builds out the baseline structure in Azure DevOps for the application development.
- Builds and destroys development and test infrastructure as needed.
- Integrates security checks for what is being developed.

Most of this was previously done manually by the different developments teams but is now handled by the platform. We have several future releases already on the roadmap so we can continue to improve the developer experience.

Developer productivity

One of our primary goals of CloudOne is to ensure developers have the tools and automation needed to do their jobs.

We want them focused on application development, not managing tools and infrastructure. Yet, we recognize that most developers have their favorites tools like Azure DevOps, Atlassian, or Team City. The problem with “favs” is NetApp ends up with many different DevOps tools being used across multiple development teams. It results in lost efficiency and productivity as developers move from team to team, learn different tools, and spend time managing the development platform instead of developing. To promote developer productivity and achieve economies of scale, we have standardized on a set of tools which can be used across all teams. Our CloudOne platform automatically builds out an environment that is based on service catalog selections for tools, which includes programming language options.

Built-in security

Building security into the development process is a challenge as it involves an “interruption” in the developer’s workflow and is often viewed as a productivity hit. We are addressing these concerns in an upcoming release of the CloudOne platform by injecting security checks into the developer’s workflow. For example, after a developer submits code, security software runs against that code to see if there are any bad practices which may lead to a vulnerability. If so, it will

generate a bug against the code for the developer to address. In the final stage of the workflow, when the production binary is generated, additional security software is run against the binary to see if there are any vulnerabilities. Again, generating a bug if something suspicious is found.

Private or public cloud

With CloudOne we adopted a container strategy from the very beginning. As a cloud agnostic solution, we use containers to deploy application components at any location. In a future release of the CloudOne platform, we will enable portable capability to move applications to any location or deploy across multiple cloud instances simultaneously.

Summary

As a total package that provides control, standardization, and cloud agnostic capability, CloudOne is an end-to-end solution as a service. It is a single environment for managing the entire DevOps lifecycle and helps our teams deploy quality code faster. •

```
3 self.file = file
4 self.fingerprints = fingerprints
5 self.logdupes = True
6 self.debug = debug
7 self.logger = logger
8
9 if path:
10     self.file = file
11     self.file.seek(0)
12     self.fingerprints = fingerprints
13
14 @classmethod
15 def from_settings(cls, settings):
16     debug = settings.DEBUG
17     return cls(job_settings=settings)
18
19 def request_seen(self, request):
20     fp = self.request_fingerprint(request)
21     if fp in self.fingerprints:
22         return True
23     self.fingerprints.add(fp)
24     if self.file:
25         self.file.write(fp + '\n')
26
27 def request_fingerprint(self, request):
28     return request.get_full_path()
```

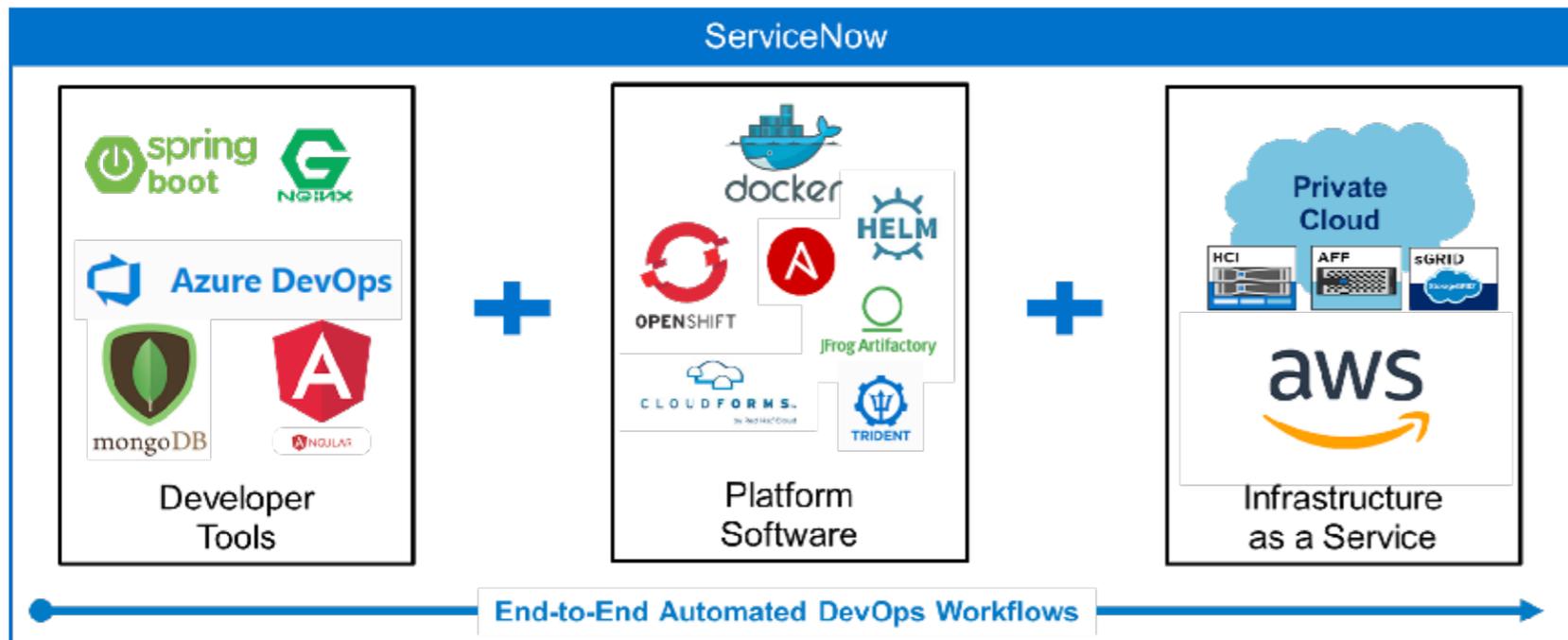
Three-Part Architecture of our DevOps Platform

MICHAEL MORRIS
SR. DIRECTOR, IT INFRASTRUCTURE

Our definition of data center is no longer four walls. It's a hybrid, multi-cloud platform where we can best get the resources and apps at the best price point and performance level. We refer to this as CloudOne. It is a software-controlled and orchestrated development platform to build and run cloud aware applications using DevOps and CI/CD delivery models.

Regardless if our resources are on premises or off premises, we need to provide IT governance oversight to ensure our investments and services support business objectives. Today all governance in NetApp IT is managed via ServiceNow, including the management of incidents, problems, and changes, our CMDB and service catalog, project and portfolio management, and more. Maintaining governance is an important part of the overarching CloudOne architecture and building a development platform for cloud-aware applications.

There are three main parts to our CloudOne architecture. The first part involves the developer tools. As a development platform, developers need tools like application stacks, e.g. OpenResty and MeanStack, and a developer ecosystem like Microsoft Azure DevOps. As a development environment, giving



developers access to open source software and tools is a given.

Another part of the architecture is platform software. Since we are building cloud-aware applications in microservices, we need containers and container management. The applications run in Docker containers, managed by OpenShift or Kubernetes, while Helm charts help us to specify the application stack. [NetApp Trident](#) is used to do the storage provisioning through OpenShift. We use RedHat Cloudforms as the cloud management platform and JFrog Artifactory to manage the binaries that get created during the build process. When our developers require resources, they

don't have to go to a PaaS platform or IaaS platform. They just do a regular code commit in their development environment because their tools are integrated with the platform seamlessly.

We then combine that with infrastructure services because the tools, the platform software, and the business applications themselves that eventually get created, all need hardware to run on. Whether it's public or private cloud, we make the assigned infrastructure transparent to the developer.

The CloudOne architecture has been built on our learnings from an Infrastructure as a Service (IaaS) offering put in place

over four years ago. This first-generation IaaS marked the beginning of our hybrid cloud strategy and allowed users to log into a central self-service portal, pick an infrastructure item from the services catalog, and then get it delivered across any of the clouds orchestrated by NetApp IT. We thought we were golden. Yet it wasn't what our developers wanted or needed.

One key principle we adhere to with our CloudOne architecture is end-to-end, automated DevOps workflows. We want our developers spending their time writing code and releasing changes, not dealing with cloud resource provisioning, OpenShift changes, or Artifactory set up. When a developer wants an environment where he or she can start building a new application, they simply go into ServiceNow to begin. Using a service catalog, a new development environment is automatically built. This allows our developers to do what they do best—write and release code. •



Application Strategy and Achieving Target State Architecture

KAMAL VYAS
IT INFRASTRUCTURE ARCHITECT

Embarking on an enterprise cloud journey without an application strategy is a recipe for disaster and will result in higher costs and disappointments for the organization. The first step in our cloud journey was to determine the right cloud solution—we call it target state architecture (TSA)—for all business applications in the enterprise portfolio. Inside NetApp, we have rationalized all our business applications, and based on business strategy and criticality, have determined the right cloud solution (IaaS, PaaS or SaaS) for each one. This included deciding if we should host applications on-premises or in a public cloud. Our approach was to:

- Rationalize our business applications and their fit in our hybrid multicloud environment
- Determine the right cloud solution for the applications using the five “R’s”
- Align applications to the right cloud solution, i.e. TSA

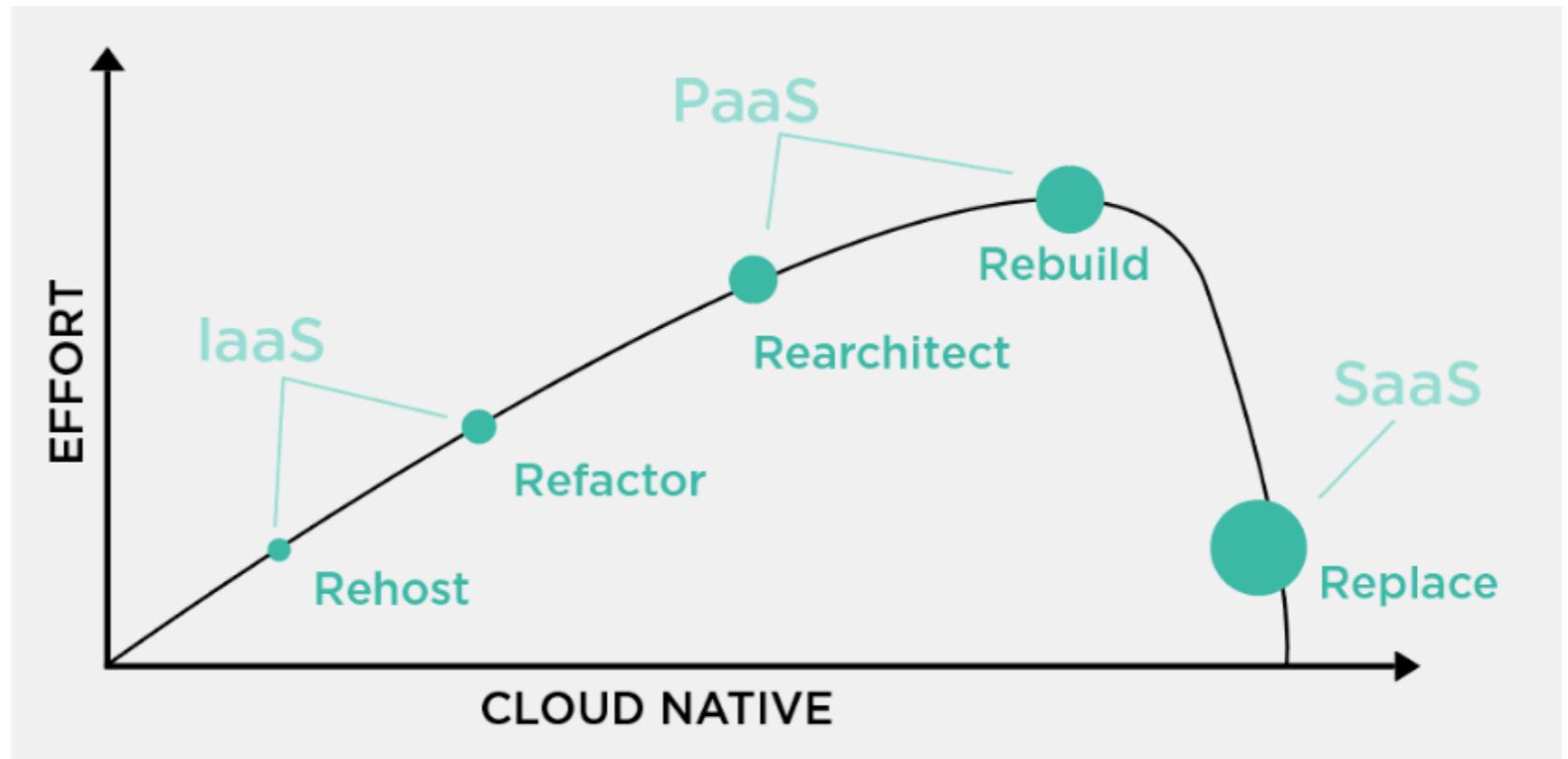
01 | Application Rationalization

Using the TIME (Tolerate, Invest, Migrate, Eliminate) model published by [Gartner Research](#), the team [completed a thorough rationalization](#) of about 350 applications in NetApp's business portfolio. Tolerate and eliminate apps will be slowly phased out. Invest apps had already been aligned to the right target state, but further functionality needs to be developed. Migrate apps are still critical to the business and will have to be altered to reach target state architecture.

Once formalized and socialized, this rationalization helped IT and the business to align on which applications were critical and strategic, and which ones should be sunseting. We have focused our cloud efforts on the invest and migrate categories.

02 | The Five "Rs"

Once we determined which apps are critical, next we determined the level of "cloudiness" required. This is where our next level of classification, called the [5Rs \(introduced by Gartner\)](#), comes into play. Each of these Rs—rehost, refactor, rearchitect, rebuild, replace—has a different level of effort associated with



it, and is aligned to the strategic value that the application provides. As the saying goes in IT, for every effort there must be a value associated with it.

For example, if an application supports a common business process not unique to NetApp's business, e.g. email, our strategy is to assign that to a Replace cloud model and use SaaS-based options like Office365. The business value here is simplification.

If an application provides strategic or innovative advantage to NetApp, we aligned it to the Rebuild cloud model,

i.e. PaaS with full DevOps. These applications are built ground up with full control of the entire stack via best in class processes and technologies. They help us deliver features and cost efficiencies at the speed our business demands.

Other applications like packaged applications bought from a vendor, such as Data Lake or Hadoop solutions, fit into one of the other three Rs. Rehost (IaaS) is the bare minimum model with infrastructure delivered as code and by applying other efficiency levels as applicable:

Commodity Services	Innovation Services	Differentiation Services	Systems of Record
<p>DEFINITION: Backoffice IT productivity applications and tools</p> <p>STRATEGY: Simplify by leveraging best in class SaaS with no customizations</p> <p>EXAMPLES: Office 365, ServiceNow, Zoom</p>	<p>DEFINITION: Applications and services that help us innovate and make products and services better</p> <p>STRATEGY: Enable business to try/fail/succeed fast, leveraging cloud-aware principles and DevOps practices</p> <p>EXAMPLES: New ideas, DevOps innovation</p>	<p>DEFINITION: Applications unique to NetApp and that differentiate NetApp from competitors</p> <p>STRATEGY: Enable business to release value faster, leveraging cloud-aware principles and DevOps practices</p> <p>EXAMPLES: Active IQ, AutoSupport Services</p>	<p>DEFINITION: Backoffice applications needed to run NetApp's business</p> <p>STRATEGY: Simplify by leveraging best in class SaaS, minimize customizations</p> <p>EXAMPLES: Workforce/HR, CRM, ERP</p>

- Refactor, i.e. IaaS + code structuring/modularization, API gateways, etc.
- Rearchitect, i.e. IaaS + containerization, cloud, etc., but not full CI/CD due to longer release cadence

Our strategy in a nutshell is to:

- Keep the business running by leveraging SaaS (Replace) for common business processes, i.e. Commodity Services and System of Record type applications
- Transform the business by leveraging PaaS and DevOps for strategic and innovative applications
- Avoid accumulating further technology debt by leveraging PaaS and DevOps for all new (non-SaaS) applications
- Leveraging IaaS as the bare minimum for

all packaged apps and gain Refactor and Rearchitect efficiency if possible

03 | Buy, Build, Rent

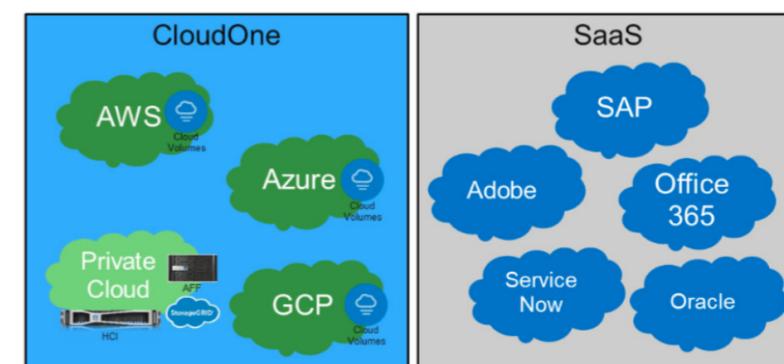
Once we completed the app categorization exercise, we were surprised to see that the best suited target state architecture for many of our applications was SaaS. This makes sense, as a lot of the applications are used to keep the business running. There are SaaS providers that can deliver common capabilities at a price and performance point which makes sense.

Our strategy is to identify the best-in-class SaaS solutions and rely on them

for our requirements. Over the next 3-5 years, we will migrate 70% of our applications to SaaS, e.g. replace where needed per the 5Rs above. For the remaining 30%—which align to the other four Rs—will move to our CloudOne platform where we will be providing IaaS, PaaS, CaaS, and DevOps services, which our developers can leverage to build modern cloud native applications.

04 | Pillars of CloudOne Platform

CloudOne is an anywhere—private or public cloud— solution for software-controlled resources at the value point we desire, and which can be orchestrated and managed via tools and automation.





Containers provide application portability within our enterprise. We are heavy on Kubernetes, using Docker as our platform.



Configuration Management allows for consistent environments. We are avid users of RedHat Ansible and RedHat CloudForms for getting infrastructure up and running.



Code and Binary Management provides fast, real-time data access. For this, NetApp IT uses Git and JFrog.



CI/CD (Continuous Integration, Continuous Deployment) provides a secure streamlined development experience. Our primary tool for this function is Jenkins.



Cloud and PaaS allow data access without infrastructure management. Our team uses OpenShift and various cloud providers.

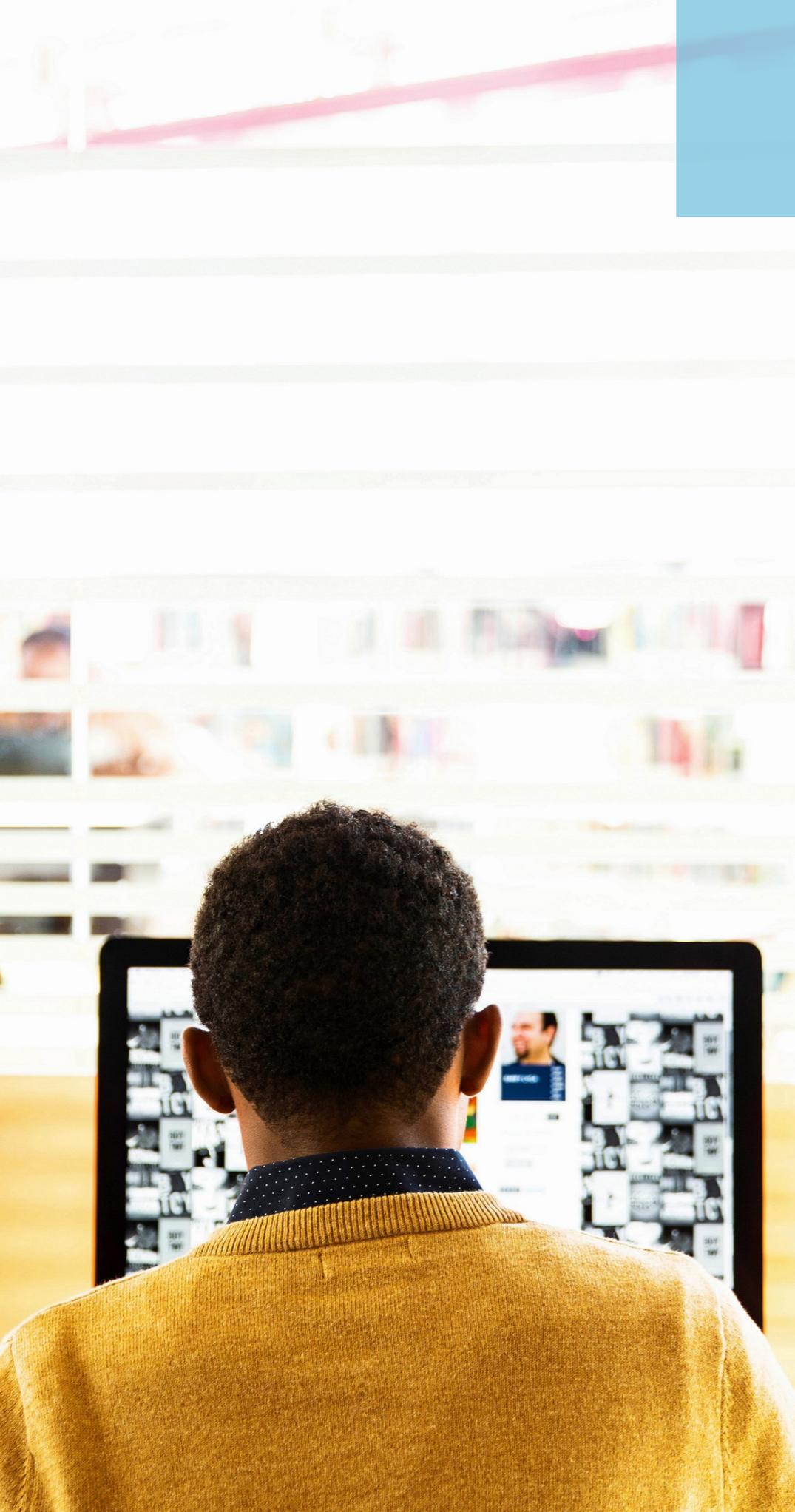


Monitoring offers actionable insights into environments. As we define and gather more metrics from our applications, this pillar will evolve to become analytics. We monitor with Zenoss and Splunk.



To the left is a glimpse of the six essential elements we used to build and maintain our CloudOne platform. The entire stack comes together to deliver application developers and application operations a complete end-to-end automated experience, where distractions are removed like building infrastructure, chasing testing/security teams, or cleaning up old environments.

We strive to remove all distractions so that our developers can focus on delivering code and value at speeds the business demands. •



IT Operational Support

MICHAEL EUBANKS
SR. DIRECTOR, IT OPERATIONS

For the on premises and private cloud environments, NetApp IT has a mature and productive operational support model that is well managed, has proven processes and established metrics, and results in little or no business disruptions. But that wasn't always the case. In late 2011, IT experienced Priority-1 (P1) outages almost every day. We found that the more change introduced, the more volatile our IT environment became, and the more our teams behaved in a reactive manner.

Fast forward to today, and we are providing a predictable, steady-state mode of operations regardless of the changes being introduced into the environment. We have seen a dramatic reduction in P1s and can go months without one! All very good stuff, yet the operational support model is shifting with the world of cloud and our adoption of a SaaS-first (Software as a Service) strategy.

Impact of a SaaS-first cloud strategy

NetApp IT plans to move 70% of our corporate business apps to Software as a Service (SaaS) providers who have already taken the time to invest and automate common business routines like email, collaboration, ERP, and CRM. For example, we have moved our Exchange environment to Microsoft Office 365. While we still have a messaging support team, the tickets that they receive are very different with more account related or entitlement related queries as opposed to infrastructure related. As a result, we are shifting the focus of our support teams to become products experts and administrators as opposed to technical gurus. The key to all of this is documentation and designing supportability from the start.

Another change that comes with our SaaS-first strategy is the need for new skillsets and processes to manage contracts, usage, documentation, configuration databases, and more. The new landscape requires strong customer service skills, the ability to listen, and liaisons to work closely with business users. The ability to ascertain what the end users are saying becomes of primary importance, as the support for cloud apps and others are UI based. It's about being able to understand and translate what the end user is saying, requesting, or complaining about, and

then working through the various support screens and settings to come to a solution. It is much like what we have with ServiceNow today.

Today we use the CMDB (configuration management data base) within ServiceNow to store application contacts, support process-related issues, or instructions, and those types of things. It is our single source of truth with a CMDB process owner focused on making sure we the right information, reports, and associated contracts are identified. In the cloud environment, we don't have this as it's behind a UI (user interface). We are having to fundamentally shift, while still ensuring the services are working, the applications are available, and the processes are happening the way they should. This requires a different focus in our monitoring strategy, but hopefully, not a different tool.

What keeps me up at night

If I had to pick one or two things that keep me up a night when thinking of IT operational support for cloud and SaaS, the first would be the CMDB process owner(s). They need to ensure we are gathering the right information, negotiating the right types of service level agreements and so forth. Traditionally, we made commitments to the business based on set service levels, but when you start negotiating with 10-15 different SaaS providers, they may all

have different ideas and criteria. It has the potential to get complicated fast.

We also need a centralized way to make sure that we have the operating level agreements in place, and that we have all of that documented in the right way. Why? Six months down the road, when everyone's moved onto another focus and a user calls to report "this doesn't work anymore," we know who to call and how to fix it quickly. Again, documentation will be key.

In the world of cloud and SaaS, there's going to be a different person, a different company, a different process for every application. We're need people whose job it is to manage it, with a central place to get help, and receive the proper attention to ensure that it gets corrected efficiently and effectively. •



Building in Security by Default

DEREK BOTTI
CLOUD & APPLICATION SECURITY ARCHITECT

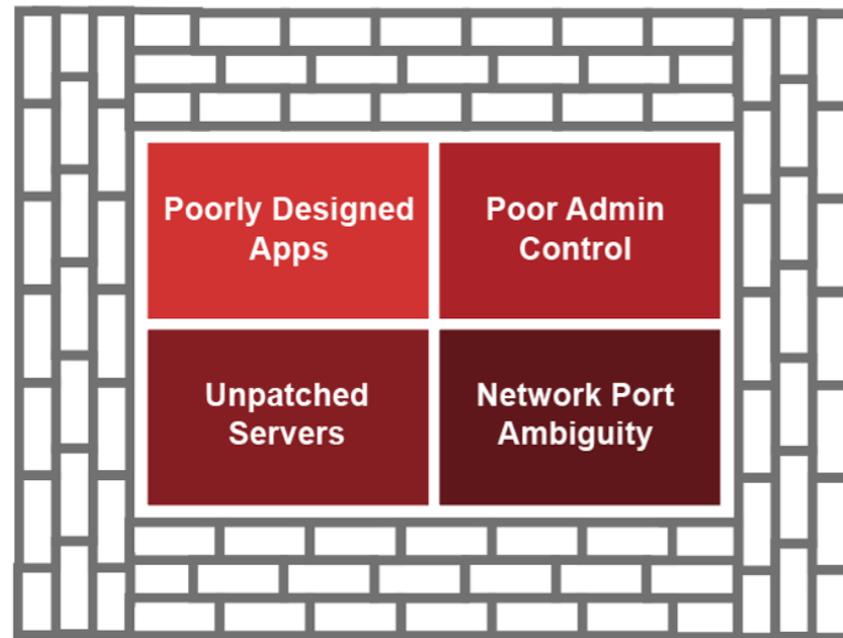
As a certified application integration and security architect, I understand the traditional data center security approach to monolithic applications. The apps are typically heavily customized, inefficiently designed, and can reside on unpatched servers with poor admin control. Network port ambiguity can also exist because the app owners don't know what ports their apps run on. As a result, the app owners often instruct IT Security to open many ports.

In response, IT Security builds a hard, complex network security perimeter. This hard shell is intended to protect the internal applications from bad actors who hack unpatched servers or hosts to get into the Active Directory, escalate privileges, and move laterally. It is a complex approach and makes automation nearly impossible.

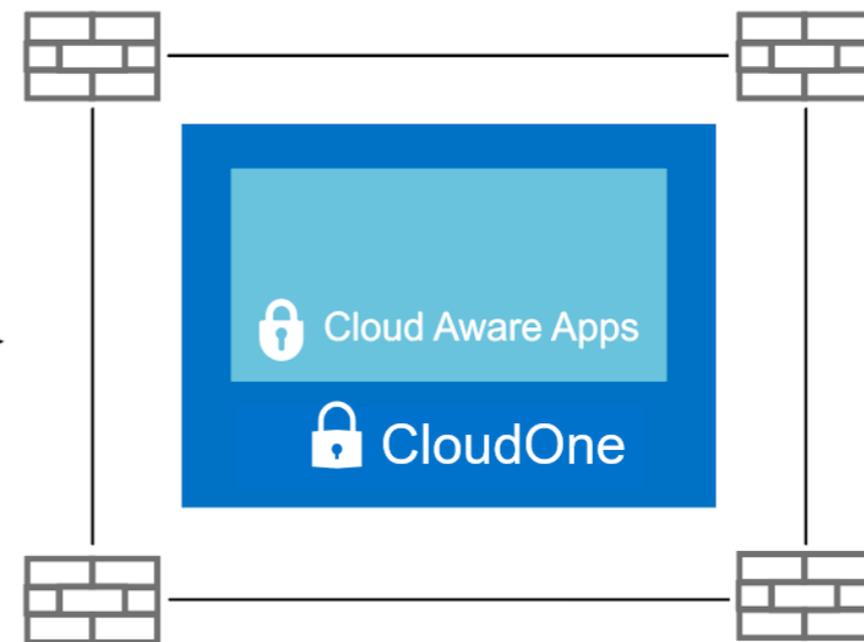
When designing our CloudOne DevOps platform inside NetApp, we wanted to build in security by default. From the beginning, we took every opportunity to build security into the platform and the applications themselves.

Current IT Environment

Future IT Environment



Hard, Complex Perimeter



Simplified and Automated Perimeter

With CloudOne we have built-in secure code practices for development teams like software-controlled development workflows that require security features and checks. Automated application testing is done that includes security vulnerability tests that are tracked as defects throughout the application lifecycle management process. We restricted administrator capabilities while infrastructure agnostic containers allow hardened virtual machines. Network segmentation is done with only known, standard protocols.

CloudOne radically simplifies, although not fully eliminates network perimeter security. Instead, it allows us the ability to automate firewall perimeter security through Continuous Integration, Continuous Deployment (CI/CD) efforts that is software controlled and orchestrated. We run a series of QA tests during the CI/CD process currently and are looking to add extra tests in the future. Automation with CloudOne is helping us to lessen the burden of our teams. •



Authors



Jeffery Boni



Derek Botti



Michael Eubanks



Michael Morris



Kamal Vyas

For information on NetApp DevOps solutions, visit [NetApp.com/DevOps](https://netapp.com/devops)

