EBOOK

DevOps Adoption in NetApp IT

NetApp®

Table of Contents

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, yet you cannot reach the full potential of DevOps if you only focus on getting the practice going.

I find that 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 to specific tools, applications, and processes. It's also not uncommon for developers to build their own tools and applications, as well as maintain them, which means they are not developing.

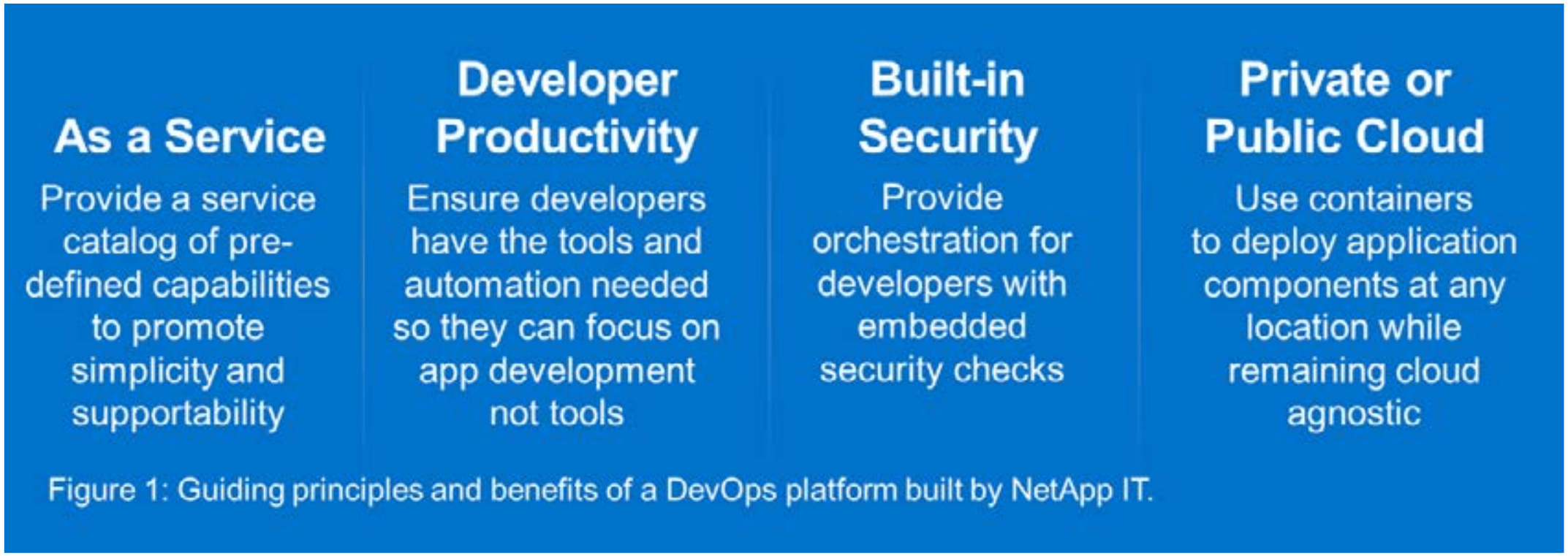


Figure 1: Guiding principles and benefits of a DevOps platform built by NetApp IT.

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 DevOps practice, we had a greenfield opportunity to build a DevOps platform and influence the DevOps 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. As you know, there is no one single tool or technology that provides a full DevOps solution; it's a variety of tools and approaches integrated together.

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. In the current release of CloudOne:

- We offer a service catalog of application stacks including Angular.js, Envoy, and OpenResty, with more to come.
- It automatically updates 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 it 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. •

RETURN TO CONTENTS

Why Adopt DevOps When Building a Cloud-Aware Enterprise

MICHAEL MORRIS
SR. DIRECTOR, IT INFRASTRUCTURE

In our ebook, Building A Cloud Aware Enterprise, we detailed NetApp IT's path to a software-defined, cloud-based data center with end-to-end DevOps workflow automation. We call this CloudOne. As the senior director for technology, strategy, and innovation within the infrastructure team, my role is to build and run the technology platform to allow our software developers to use DevOps.

CloudOne includes three major environments: developer tools, platform software, and Infrastructure-as-a-Service. The DevOps workflows are automated across all three environments. This allows software developers to push and pull source code, choose application stacks, run CI/CD workflows, and rapidly produce business results. Ultimately, developers are expected to create cloud-aware applications on CloudOne, which uses microservices architectures, running in containers, in infrastructure-agnostic designs so they are cloud-portable between public cloud providers and private cloud. Inside CloudOne, we can route application workloads to whatever cloud we

want to use.

DevOps Maturity Model

We find the DevOps Maturity model posted by Sumardi Soemartopo closely aligns to our experience. Level 1 is ad hoc—meaning no DevOps other than saying the word, to a Level 5 similar to what Netflix and Facebook do with constant updates and no noticeable downtime. With CloudOne, NetApp IT meets a Level-3 classification, i.e. we have real, automated DevOps which can create, deploy and support real applications. And yet, our capabilities, experiences, and speeds are still immature. Reaching Level 5 will require a multi-year journey to improve CloudOne features, capabilities and human skillset.

Maturity Level	People	Process	Technology
Level 1 Ad Hoc	• Silo based • Blame and finger-pointing • Dependent on experts • Lack of accountability	• Manual processes • Tribal knowledge the norm • Unpredictable and reactive	• Manual builds and deployments • Manual testing • Environment inconsistencies
Level 2 Repeatable	• Managed communications • Limited knowledge sharing	• Processes established within silos • No standards • Can repeat what is known, but can't react to unknowns	• Automated builds • Automated tests written as part of story development • Painful but repeatable releases
Level 3 Defined	• Collaboration exists • Shared decision making • Shared accountability	• Processes automated across the SDLC • Standards across organization	• Automated build and test cycle for every commit • Push button deployments • Automated user and acceptance testing
Level 4 Measured	Collaboration based on shared metrics with a focus on removing bottlenecks	• Proactive monitoring • Metrics collected and analyzed against business goals • Visibility and predictability	• Build metrics visible and acted on • Orchestrated deployments with automatic rollbacks • Nonfunctional requirements defined and measured
Level 5 Optimized	A culture of continuous improvement permeates through the organization	• Self-service automation • Risk and cost optimization • High degree of experimentation	• Zero downtime deployments • Immutable infrastructure • Actively enforce resiliency by forcing failures

DevOps Implementation Lessons

We learned three valuable lessons when addressing DevOps implementation challenges.

1) **Getting the base CloudOne environment built and ready.** This has been a year-long effort from initial vision to first release. We realized early on that it takes time because of the complexity to get DevOps workflows automated across developer tools, platform software, and IaaS.

2) **Making sure we don't build a DevOps ghost town our app development teams would not use.** Concurrently with the base CloudOne build, we educated the app development teams on cloud-aware apps and DevOps. By taking the time to provide an education focus, we were able to establish a more trusted working relationship between the infrastructure and business apps teams to create a pipeline of new and existing apps to build or migrate to CloudOne.

3) **Keeping key engineers focused.** My new favorite analogy of CloudOne is to think of it like a train which rolls down a never-ending track, taking on new cars (or features) and passengers (applications) with no end in sight.

CloudOne is a service which will go on for years, so the engineers who have participated in the initial build need to stay engaged throughout. We are creating a CloudOne operating model using agile practices and frequent releases to maintain engagement.

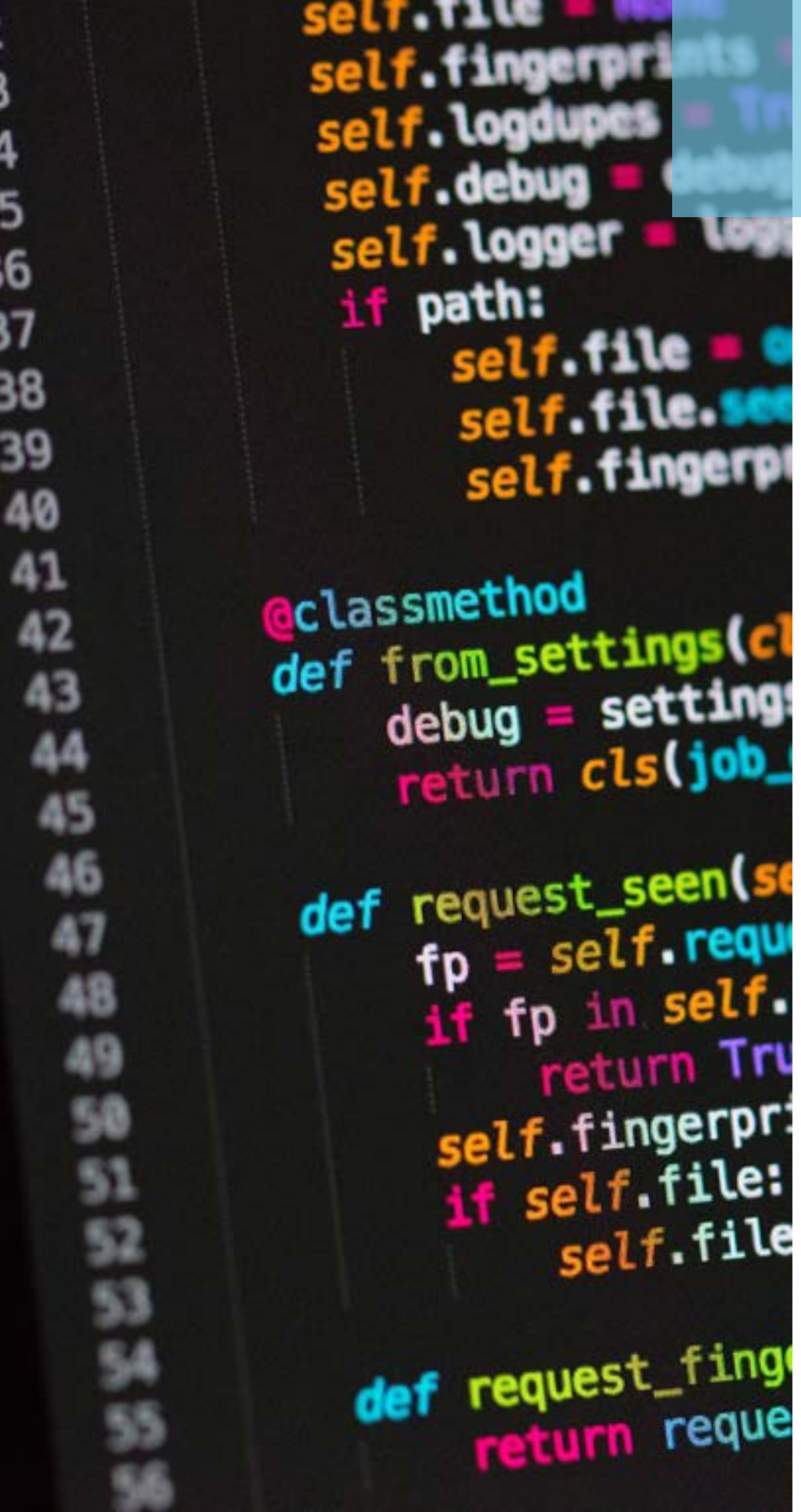
Role of NetApp Technology

As Customer-1, our goal is to use NetApp products, services, and reference designs early and as much as possible. We also spend a significant amount of time speaking to the NetApp product teams, giving our feedback on products, sharing good and bad implementation results, and influencing new features and roadmaps—ensuring we contribute to improving NetApp's customer experience and satisfaction.

CloudOne uses NetApp ONTAP Ansible playbooks for infrastructure automation and the NetApp Trident plug-in for volume creation in OpenShift. For our private cloud we are using these NetApp technologies:

- Hyperconverged Infrastructure (HCI) to provide compute, memory and (local) storage for containerized application components
- All Flash FAS (AFF) for the stateful (NFS, iSCSI) storage platform for databases and high-performance workloads
- StorageGRID as the repository for heavy object storage usage given cloud-aware apps are inherently stateless and should use a stateless storage mechanism.

These cloud-friendly technologies are tangential to our existing install-base products and are particularly important to our cloud journey. Ultimately, all DevOps tools, software, and applications created by DevOps workflows need infrastructure. Taking an infrastructure product, simplifying it, and making it software-defined has allowed us to plug into DevOps environments like CloudOne. That's the future of IT—inserting good products into existing DevOps environments through APIs and automation. •



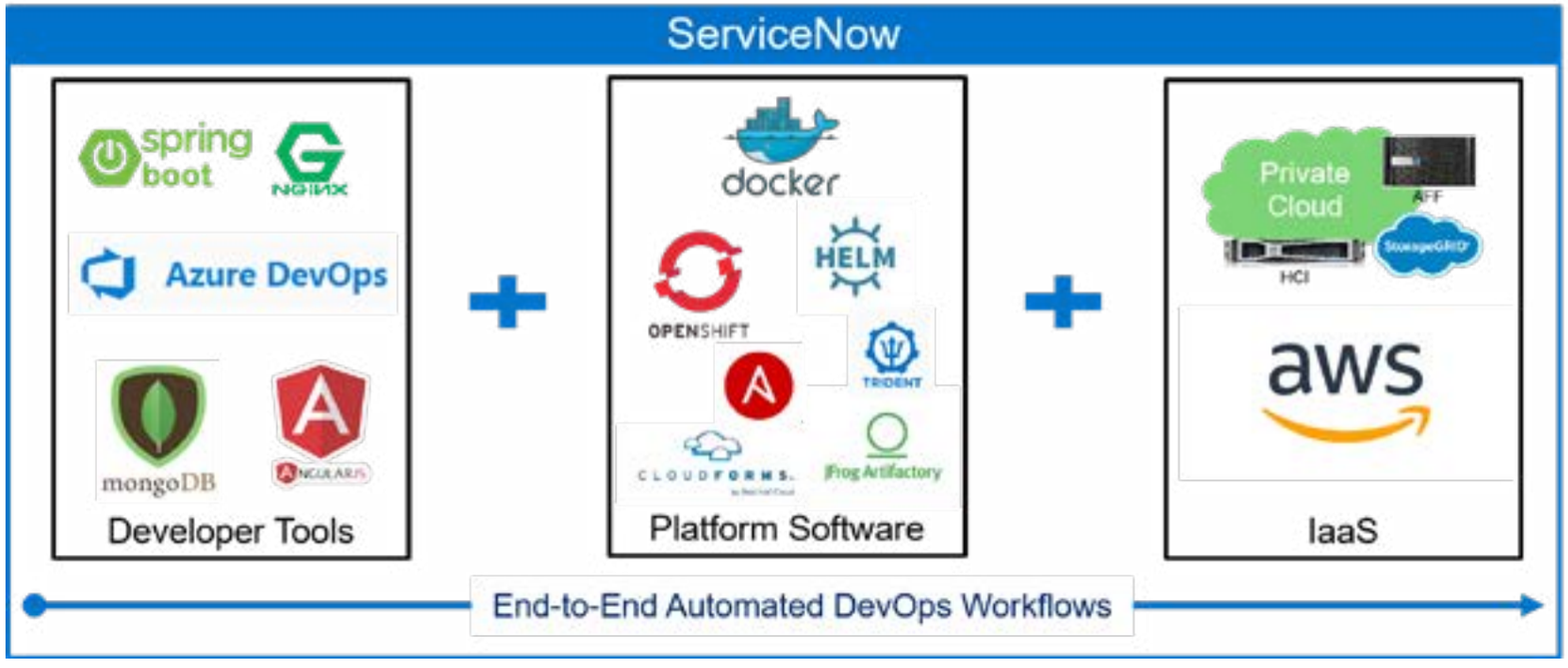
Three-Part Architecture of our DevOps Platform

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.

RETURN TO CONTENTS

Rapid, Frequent CI/CD

MOHAN RAJ
IT SR. MANAGER, CLOUDONE DEVELOPER EXPERIENCE

When introducing our CloudOne platform inside NetApp, we wanted to meet certain minimum viable product (MVP) features including the capability to allow rapid, frequent production change through Continuous Integration, Continuous Deployment (CI/CD). We wanted our developers focused on writing code and rapidly releasing application changes, and not be concerned about other IT services like servers, storage, networks, and more.

With our CloudOne platform, we provide automation for frequent small production application changes. Automation helps to reduce the risk associated with large-scale production deployments while adhering to DevOps principles like moving fast and making small changes quickly. With traditional IT there are change control processes that often involve large changes, with some requiring the application to be taken offline. All of this takes time and introduces potential problems for something going wrong. It is not fast.

My favorite feature of the CloudOne platform is how the development (DEV), stage (STG), and production (PRD)

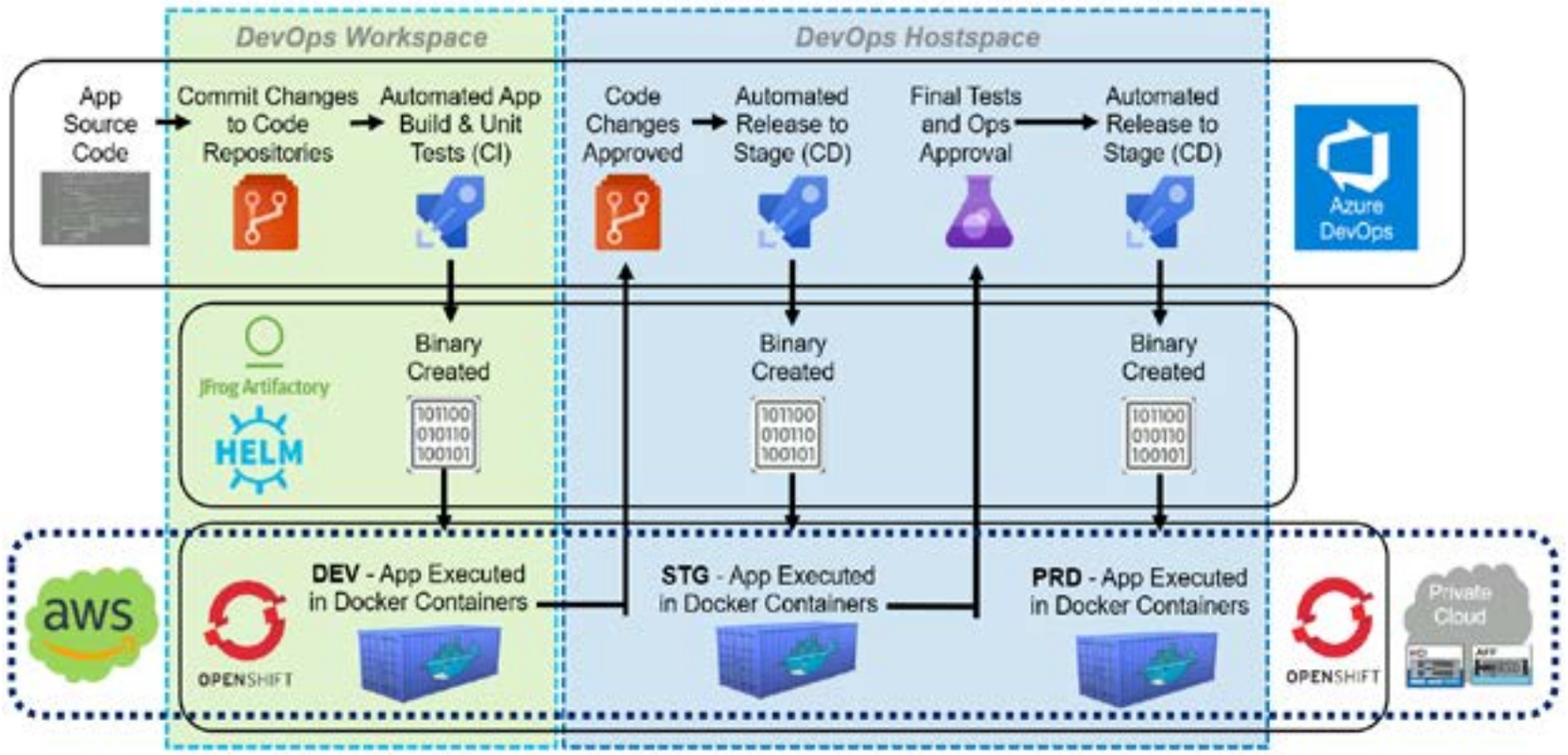
environments are handled at the application layer. In our traditional data center model, the DEV-STG-PRD environments are dedicated and sit idle until used. With CloudOne, these environments dynamically spin up and down as needed at the application layer; the infrastructure and platform reside in the operational production (PRD) environment.

CloudOne application onboarding

Our CloudOne MVP introduction happened in November 2018. The MVP goal was to onboard simple apps onto the CloudOne platform to prove out the technology, test the process, and confirm support roles. During application onboarding, we create what we call a DevOps Workspace and DevOps Host Space.

Workspace: Where developers do their work before an application goes live. For example, developers can create code branches, develop the code, create pull requests and trigger continuous integration builds. It's where they can test out new features, fix bugs, or try out new capabilities using our standards-based technology product catalog.

Host Space: Where the stage and production application run. Its where continuous delivery releases are ultimately transitioned to. Our goal is to take the work



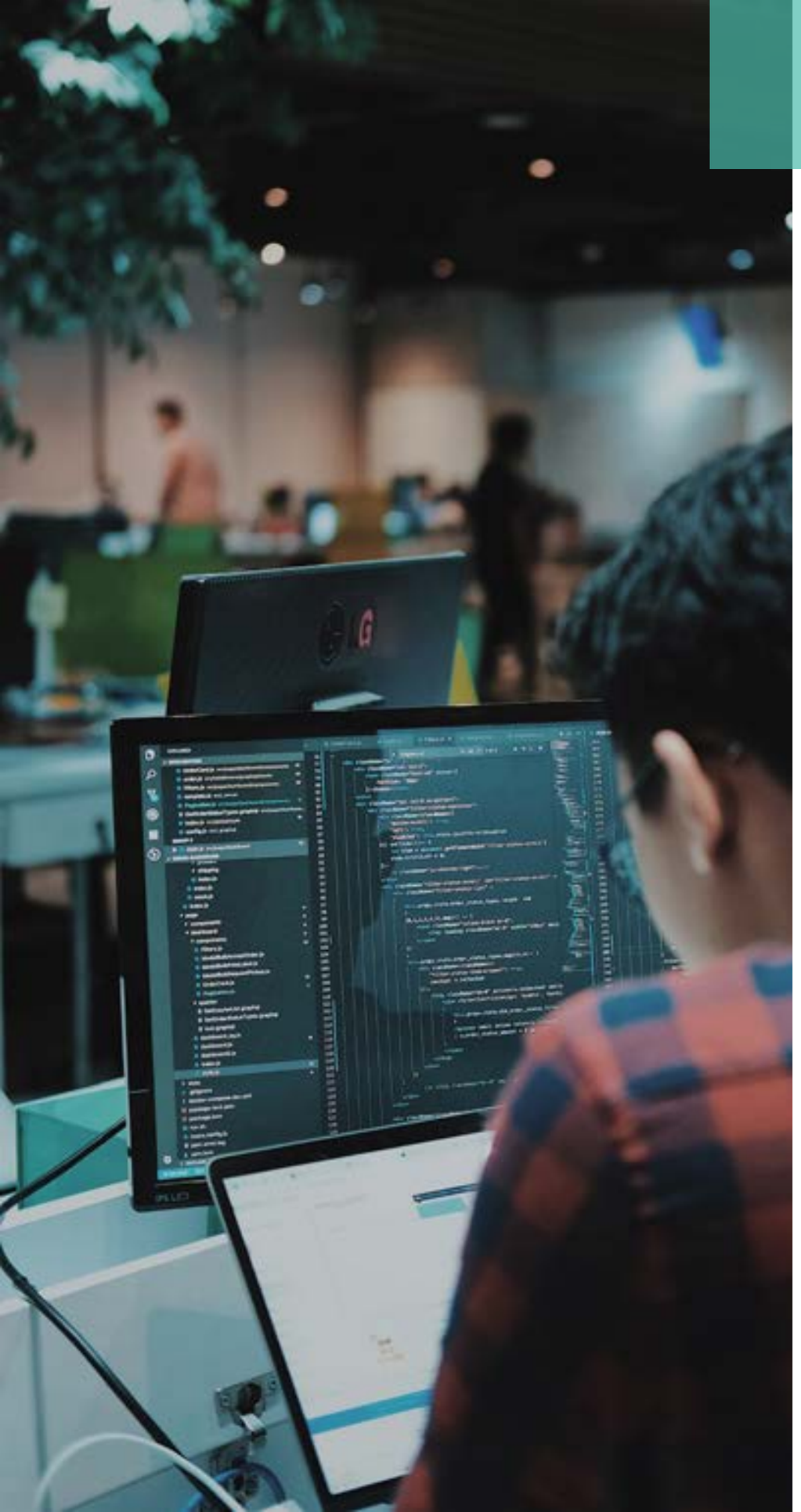
that the developers do in the workspace, and via our CI/CD process, transition the changes to production in the host space.

Putting CI/CD processes to work

Applications are onboarded once. During this onboarding, both the workspace and host space are created so that the CI/CD process can automatically work. Once it's done, then the development team can use CloudOne to rapidly move application changes to production. This is the continuous integration, continuous deployment (CI/CD) process of CloudOne.

There are three levels of CloudOne for application CI/CD. Our developers do most of their work in the first level using primarily Microsoft Azure DevOps. This service provides a full application lifecycle management ecosystem including dashboards, analytics, wiki's, and project and Kanban boards. It is where the developer's code repository resides for the applications and where they work on source code.

When an application developer submits a code change to the code repository, it kicks off an automated application



build. The build combines the application stack, along with the code changes, to create a binary that is deployed within the workspace (non-production). Within about three to four minutes—that's how long the automation takes—the developer can see their change to the application running in a container and can determine if the change or bug fix worked. If it worked, the Continuous Delivery (CD) process takes the binary that is in the workspace and copies it into the stage environment in the host space. The code change can then be approved for release to production.

Based on role-based authorization in OpenShift and Azure DevOps, the developer, development lead, UAT technician, and even operations, can do the final testing and approvals of small changes. Those involved with the change must be satisfied. Once the final approval is made, then the binary image is moved to production via automation. With OpenShift we can migrate each container one at a time to the new binary image so that the application never goes down. The whole process to implement a change—from the developer working on the code all the way to production—can be less than 15 minutes assuming no delay in approvals.

By adhering to an end-to-end, automated DevOps workflows in CloudOne, our developers can spend their time writing code and releasing changes, and not have to deal with infrastructure or platform services. The automation provides the ability for frequent, small production application changes to reduce production deployment risk.

I realize it is difficult to fully understand how a development platform works without a demonstration. Please click on this video clip to see how easy it is to fix a bug in a demo application we created called Tiny URL Tool. By automating our release process end-to-end, we can rapidly make a small change to this app without downtime, risk, or duplicated supporting infrastructure.

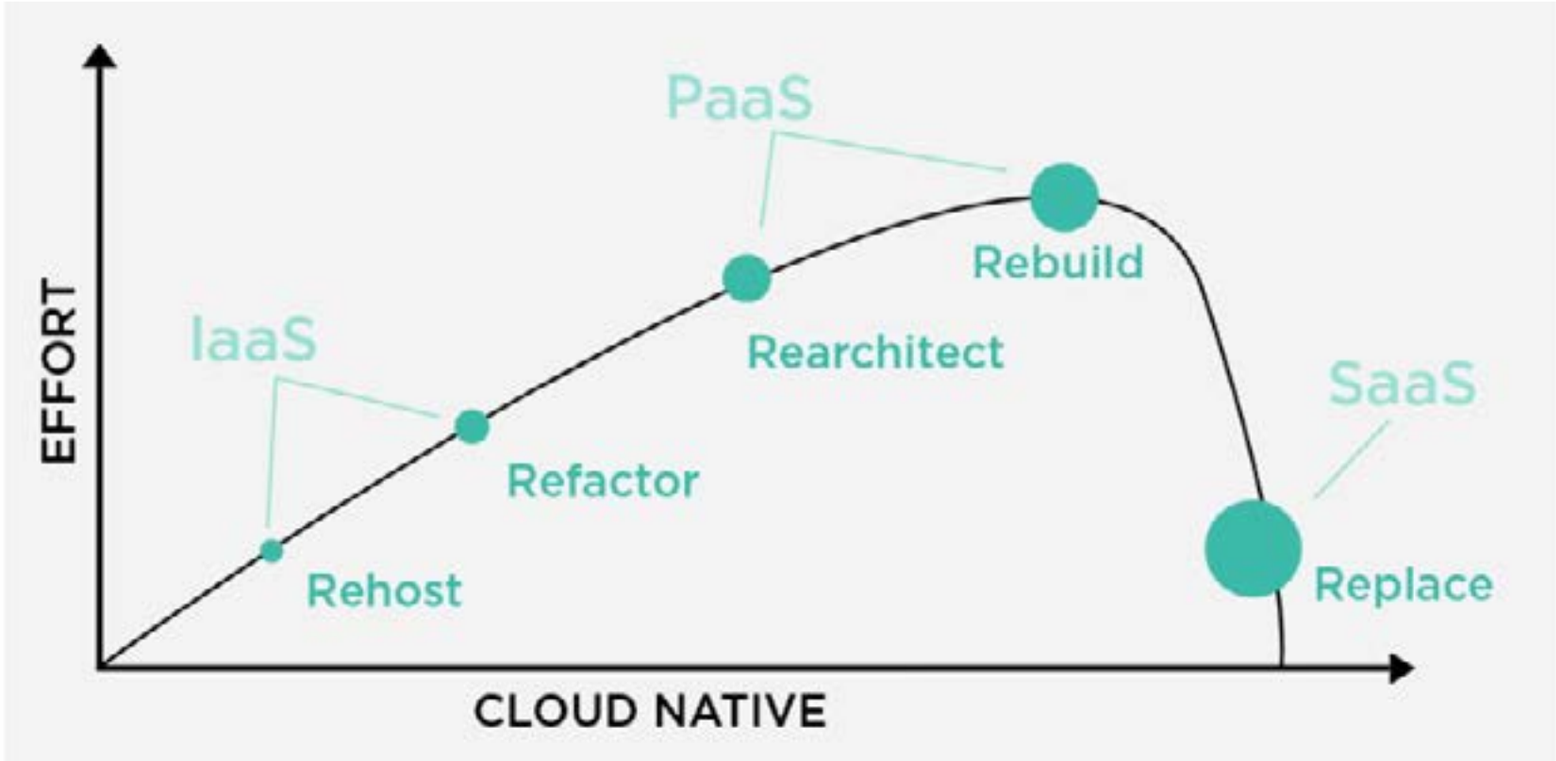
RETURN TO CONTENTS

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-prem 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 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 sunsetting. 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

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

applicable:

- 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

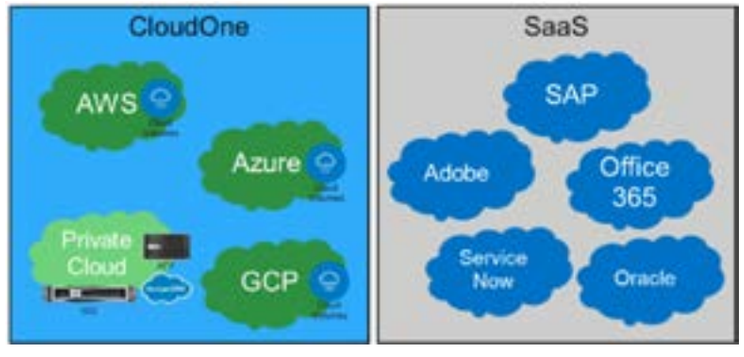
03 | Buy, Build, Rent

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, our next generation data center, is no longer the physical four-walled enterprise data center. It is anywhere—private or public cloud—where we can get 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 what we have 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. •

RETURN TO CONTENTS

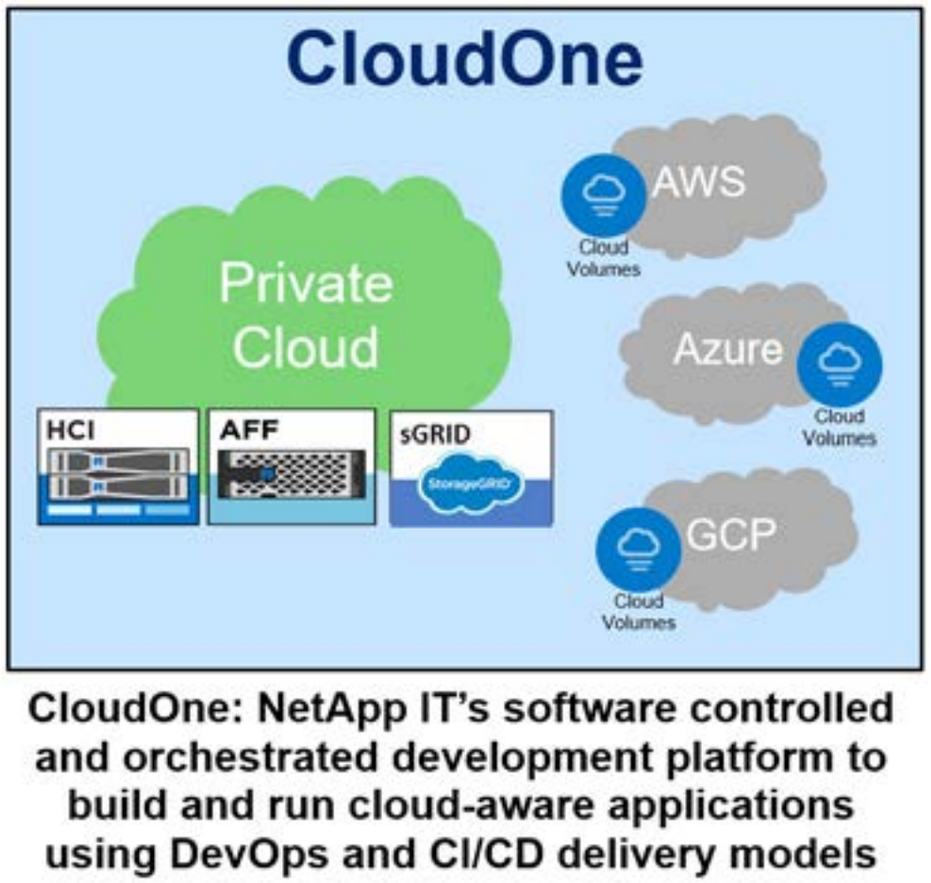
HCI, AFF and StorageGRID in our Private Cloud

GOPI SIRINENI
SR. MANAGER, CLOUD AND COMPUTE SERVICES

One of the main priorities for our IT shop in 2019 is to deliver rapid, frequent business software capabilities through automation and cloud technologies. In a NetApp IT eBook called Building A Cloud-Aware Enterprise, we shared the vision of our CloudOne platform as a software-controlled and orchestrated development platform to build and run cloud-aware applications using DevOps and CI/CD delivery models.

The goal of CloudOne is to track and manage cloud costs with visibility across our multi-cloud enterprise. We focus on taking advantage of the elastic, burstable capability of the public cloud with its variable cost model, or the fixed cost structure of our private cloud. Either option requires dial tone-like infrastructure that is scalable and supports microservices and containers.

For our private cloud, we chose three NetApp technologies that met our requirements and are controlled and orchestrated through software.



CloudOne: NetApp IT's software controlled and orchestrated development platform to build and run cloud-aware applications using DevOps and CI/CD delivery models

- NetApp Hyperconverged Infrastructure (HCI) runs container-based technologies like Kubernetes and Docker and gives the containers exactly what they need: a little compute, a little memory and a little storage. It operates at a fixed cost and provides the proper compute resources and the ability to incorporate an API-controlled technology for storage.
- NetApp StorageGRID as the repository for object storage use. It is perfect for unstructured data that sprawls, may be retained forever, is rarely updated, and is meant to be accessed by multiple application services across geographical boundaries. Cloud-aware applications are inherently stateless, so they should take advantage of a stateless storage mechanism.
- NetApp All Flash FAS (AFF) acts as a stateful (NFS, iSCSI) storage platform for our databases and high performance workloads.

NetApp also leans on tools like RedHat Ansible to automate software provisioning, configuration management, and application deployment. To automate their storage configuration tasks, the team relies on NetApp Ansible modules for ONTAP, Element OS and E-Series. For example, by using Ansible ONTAP modules they have reduced Day 0 build times from a 2-day manual process to an automated, 10-minute process (read more here).

Ultimately, what we are accomplishing with CloudOne for DevOps is to rapidly deliver application changes at the speed of business change; NetApp technologies make this possible. Our CloudOne architecture includes not only the foundation of our private cloud moving forward, but the opportunity to use the same automated orchestration to deploy resources or workloads into AWS, Azure, Google Cloud or any other public cloud provider. We want to automate services—regardless of the cloud provider—so our developers can focus on writing code and releasing application changes seamlessly using our CloudOne DevOps platform. •

RETURN TO CONTENTS

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 I see our operational support model 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 70-75% 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 CMBD (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. •

RETURN TO CONTENTS

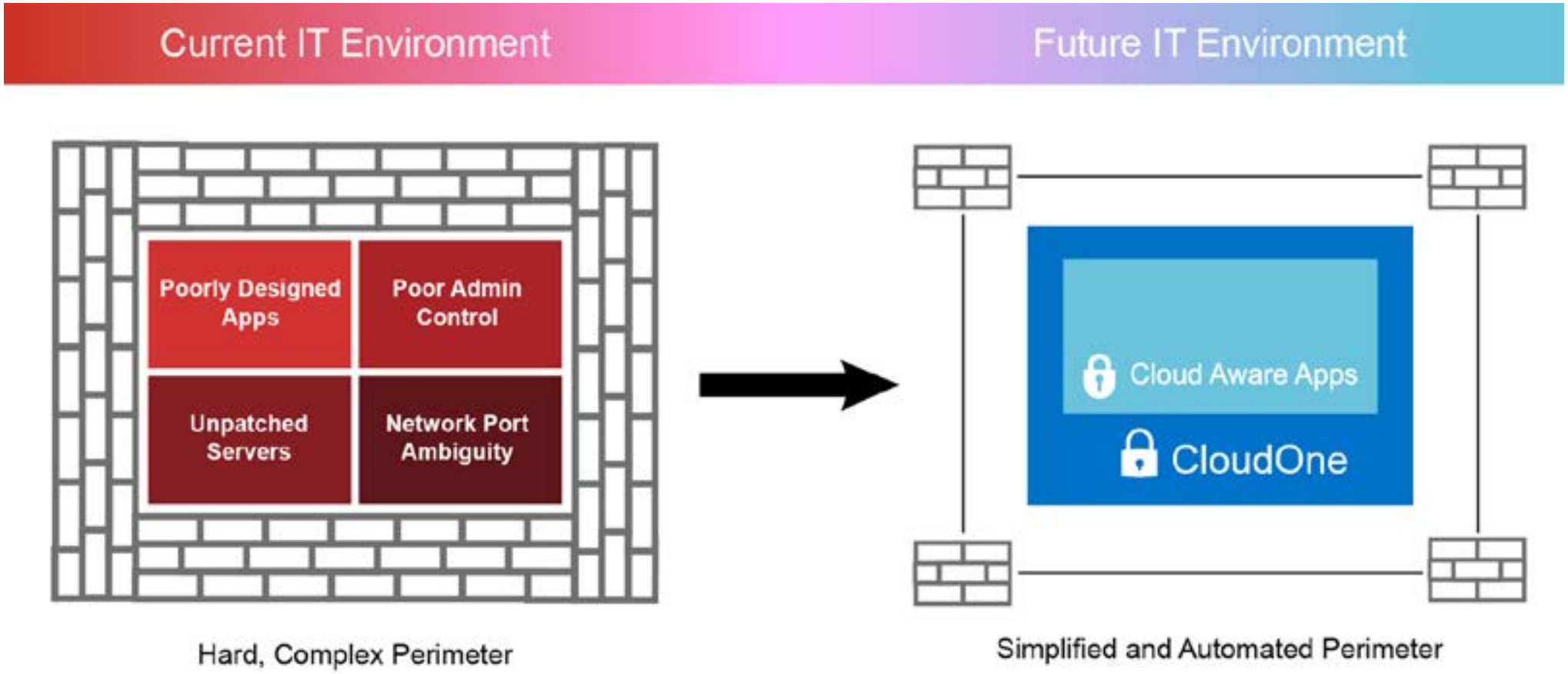
Building in Security by Default

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.



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 security team. •

RETURN TO CONTENTS

Authors

Jeffery Boni

Derek Botti

Michael Eubanks

Michael Morris

Mohan Raj

Gopi Sirineni

Kamal Vyas

RETURN TO CONTENTS