WHITE PAPER

Why Code Is Driving Infrastructure Investment

Differentiating and Adding Value with Cloud-Architected Infrastructure

NetApp®

Cloud-Architected Infrastructure: Fueling Continuous Innovation

Much has been written about the shifting sands in today's organizations. New, innovative technology companies are at the forefront of this change. These are highly successful organizations accustomed to releasing software and updates at webscale. Their ability to do so is often based on a companywide DevOps framework. DevOps opens the door for continuous innovation, rapid software deployment, and frequent updates to software-based features and products. Part of that success depends on how well DevOps is fueled by the organization's underlying data center infrastructure and processes.

Can traditional and new organizations replicate this success? What is needed to support it from the top down to the level of data center infrastructure? Most now recognize software offerings as a core differentiator, helping the business attract new customers, maintain customer loyalty, and fend off competitors. As one commentator puts it, "The business in question need not be a technology company. . . . Indeed, all companies today are technology companies whether they like it or not." [1]

Most companies also recognize that traditional application development and deployment frameworks are no longer up to the task. They see, instead, the new DevOps model of rapid microservice deployments as key. For this reason, many now seek to incorporate their own DevOps practices everywhere from HR to the data center. They are therefore adapting their own processes and data center infrastructures to drive technology and innovative new software offerings.

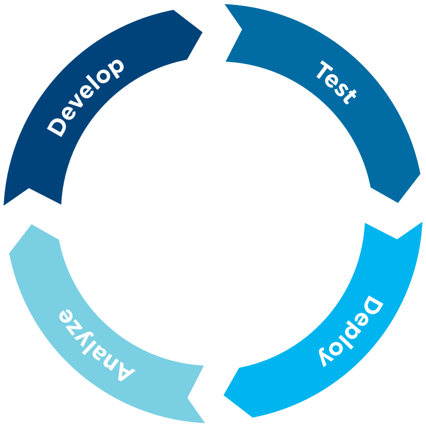
At a macro level, these facts are all well and good. But what exactly does this transformation mean on a micro level? How does this innovation-minded DevOps focus translate into practical moves and investments that organizations should make to their data center infrastructure?

This paper describes some of these practical moves and the necessary phases that an organization is likely to take on its journey to the next-generation data center. The paper also sheds light on a few trends contributing to this new kind of agile, empowered, highly innovative organization.

A Glimpse into the Data Center's Emerging Innovation Engine

What does this new, cloud-architected infrastructure look like? This is an infrastructure that helps the business compete, thanks to its ability to quickly and seamlessly create, test, deliver, and deploy new ideas and quality software to its growing base of customers. Integral to this new organization are a successful DevOps process and underlying data center infrastructure that streamline developers' needs with those of operations and testing. The result? Smooth workflows that drive continuous development, continuous testing, continuous integration, and continuous delivery of high-quality software and new features.

From an infrastructure perspective, this is an organization that can automate and quickly spin up or tear down production-quality development and test environments at will. It's about an extensible infrastructure that uses native APIs and seamless integrations to smoothly provision, change, or monitor underlying infrastructure components. It's about development and test teams that effortlessly consume such infrastructure components through code. They do so using basic definition files, API calls, or infrastructure plug-ins that are integrated and readily available from their own familiar DevOps tools, frameworks, and platforms such as Kubernetes, Ansible, Jenkins, JFrog, VMware, OpenStack, and Splunk.



Glimpse into Data Center's Emerging

It's about a world where operations teams are freed from lengthy resource provisioning and manual storage management. Using automation, they work instead to monitor the ecosystem and make sure the right infrastructure tools, integrations, and automated policies are in place to fuel their company's innovation.

How much does this new data center infrastructure sound like your organization today? What will it take to get there?

The Rise of DevOps

According to Damon Edwards's Short History of DevOps, "DevOps might just be the first technology movement that was started on and is centered around a Twitter hashtag (#devops)." [3] Depending on whom you talk to, DevOps can mean everything from a grassroots movement to a philosophy, a revolution, a cultural shift, a software framework, or a set of processes and tools.

At the Heart of New Infrastructure: Extensibility

> "Extensibility is a measurement of a piece of technology's capacity to append additional elements and features to its existing structure. A software program, for example, is considered extensible when its operations may be augmented with add-ons and plugins." [2]

Source: Techopedia.com

DevOps makes a stark contrast to traditional, often lengthy "waterfall" methods of software development and deployment typically found in the software development lifecycle (SDLC). Although it still supports the main tenets of the SDLC (develop, test, deploy, and analyze), DevOps considerably speeds this cycle while collapsing the walls and manual delays that often occur between development, test, and operations teams. DevOps practices can also streamline projects and innovation across all organizations in the business. A slate of business and technical benefits makes the move to DevOps especially appealing (see Figure 1).

Today's enterprise organizations are increasingly feeling the influence of DevOps. One case in point is Walmart, which has openly embraced DevOps practices and the use of open-source software such as OpenStack and Puppet. The retail giant uses these practices and frameworks to manage infrastructure such as helping to automate server-build processes and streamline application movement from one cloud to another.[4]

Capable of being used with either open-source or traditional enterprise software frameworks, DevOps practices have begun to have a significant impact on innovative software-as-a-service (SaaS) providers and other high-performing IT organizations. According to a survey of 30,000 technical professionals in Puppet's 2018 State of DevOps Report,[5] high-performing organizations using DevOps principles are able to deploy code 46 times more frequently. They also have 2,555 times faster lead time from commit to deploy and 7 times fewer failure rates.

How DevOps Is Driving New Needs to Consume Infrastructure Through Code

However you choose to define DevOps, its impact cannot be overlooked. This fact is now starting to drive current and future infrastructure decisions.

How seamlessly, and even invisibly, the underlying infrastructure fuels the job of DevOps will become increasingly critical to organizations. It will also be critical to a company's ability to innovate and rapidly respond to new market demands.

In a blog post, Robert Stroud, a Forrester analyst who studies trends associated with IT infrastructure and operations, alludes to the consumption of infrastructure through code. Here he describes the trend as it relates to many organizations' cloud initiatives: "The transition to cloud-first mandates a transition for infrastructure delivery, management, and maintenance to support its delivery and consumption as a reusable software component. Such infrastructure can be virtual or physical and consumed as required, without lengthy build and deployment cycles." [7]

The rise of DevOps is already ushering in a new reality: the need to consume infrastructure through code as the new normal, along with the use of code-based frameworks to automate, manage, change, or support underlying infrastructure components.

Technical Benefits

Continuous software delivery

Reduced problem complexity

More visibility

Faster resolution of issues

Less re-work and unplanned work

Business Benefits

Faster deliverty of features

More stable operating environments

More effective utilization of resources

- More time spent adding value
- Less time spent fixing and maintaining

More visibility into system outcomes

Figure 1) Benefits of DevOps.

> “Being able to iterate and adjust on the fly is invaluable. We are not trapped into decisions we made three months ago.” [6]
>
> –David Coker, Senior VP of Information Systems, Polaris Alpha

That reality could mean that developers use code to do things such as create and mount a storage volume for their own virtual machine or container environment. Again, through code, testing personnel could seamlessly and quickly provision or spin up their own test bed environments with the latest production-quality data, all without consuming much, if any, additional storage. The ability to offer real-time and on-demand resources is key to realizing the new demands on IT. Equally critical is the ability to seamlessly call and provision infrastructure components through code.

To help organizations make this transition, here are some questions for operations and infrastructure teams to ask themselves and their infrastructure vendors:

- How extensible is the underlying infrastructure?
- How can our infrastructure do more than just support applications or store data? How can it also help to drive value to DevOps and the business at large?
- How well does the infrastructure integrate with the organization’s chosen tools and frameworks?
- How easily can infrastructure components be called or accessed using software code-based methods (APIs, plug-ins, native integrations, and so on)?
- How quickly can the infrastructure adapt to changing needs and conditions?

The answers to these questions could mean the difference between minutes instead of weeks to provision resources; intuitive and automated workflows instead of arduous and manual processes; and rapid, streamlined deployment instead of trouble-filled handoffs and delays. These answers could also mean the difference between just serving application needs and adding true value to the business.

A NetApp® SolidFire® storage infrastructure offered South Africa–based provider Internet Solutions easy integration with both OpenStack and VMware. The infrastructure enabled full automation with the Internet Solutions continuous integration/continuous delivery (CI/CD) workflow. It also allowed DevOps services and virtual machines with containers to be deployed up to 15 times more quickly than with traditional architectures.

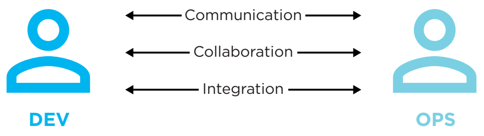
> “The integration with SolidFire and OpenStack is incredible. It is really providing significant value in that space from both integration and scalability perspectives.” [8]
>
> —Kervin Pillay, Director of Technology, Internet Solutions

DevOps: Putting a New Spin on Known Infrastructure Ideas and Features

Organizations involved in managing data center infrastructure environments are used to doing what it takes to “keep the lights on.” When it comes to managing compute, network, or storage resources supporting data center applications, this fact often translates to avoiding risk while making sure of system availability, performance, and reliability. In terms of storage systems, this focus extends to features needed to minimize data loss and downtime while reducing time to recovery. Taking rapid snapshots, or point-in-time (PiT) copies, of a dataset is one good example of this use.

Due to their ability to be rapidly recovered, snapshots are often used as a critical component of enterprise IT disaster recovery plans.

However, the emergence of DevOps and its building drive for code-based access to infrastructure is now causing infrastructure features (such as snapshots and cloning) to be viewed in a new light. In the next-generation data center, what other value will these technologies and features bring to the organization?



DevOps

What value could it bring to DevOps if snapshots and clones, for example, could be called directly by developers or testing and QA personnel? What if they could use such features to rapidly access or refresh production-level data or add new volumes on their own when they need to?

Instead of taking months and multiple teams to spin up environments, these features can be directly consumed through infrastructure integrations within existing DevOps tools and frameworks. Instead of working against stale datasets due to long lead times for provisioning, these features make it easier to move code quality to earlier phases of the process by delivering datasets in real time and on demand. The only things needed are the right type of extensible infrastructure and the know-how to enable it.

How much time and how many frustrating manual hand-offs could be saved in the development, test, and deployment phases if infrastructure were seamlessly available when needed?

The preceding description is just a small taste of the new data center.

> "We can provide a copy of the bank to anyone at any time—in 10 minutes. At the click of a button, developers can get all the data they need, all the servers, applications, configurations, testing, and everything that goes with it, completely verified, ready to rock and roll, completely off the shelf. Then they can run any test they like. It's amazing." [9]
>
> —Ben Issa, Head of IT Strategy, ING DIRECT Australia

Reaching Maturity in DevOps Methods and Infrastructure Practices

At the beginning of this paper, we described a new reality in which DevOps practices run at peak levels to accelerate innovation. In this scenario, infrastructure seamlessly moves in a rapid workflow geared toward continuous development and continuous delivery.

In the real world, however, many organizations are not there yet in their level of DevOps success, their exploration of emerging DevOps tools, or how well their infrastructure can be automatically consumed dynamically. In reality, most organizations also find themselves addressing different focus areas in their evolution at different times.

In terms of DevOps, each organization is likely to have a shifting focus that surrounds its own business priorities or its own unique strengths and weaknesses. Some might focus on improving code quality or accelerating testing, whereas others work to improve system visibility associated with what's happening in their own environment. Still others might focus on the need to more quickly identify bugs and any rework required. Some might be happy with deploying less often, at least early on, whereas others might work to deploy features multiple times per day. The priorities of the organization are also likely to change over time as constraints and needs shift. However, to support a DevOps framework, businesses must find a way to support these six core capabilities: containers, configuration management, code and binary management, continuous integration/continuous delivery, cloud and platform as a service (PaaS), and analytics.

Regardless of the focus areas and goals unique to each organization, infrastructure and operations teams can still move forward from where they are now to where they'd like to be. In doing so, many are likely to follow a few key maturity phases in their infrastructures (see Figure 2).

Working with a wide range of enterprise environments as well as cutting-edge SaaS organizations, we see four common infrastructure maturity phases that organizations are likely to traverse on the way to their own next-generation data center.

Most infrastructures have already passed through the few early phases of consolidation, virtualization, and even the use of scripted tasks to begin automating basic infrastructure tasks such as provisioning and reporting.

As organizations move forward into further stages of automation, however, what should they hope to achieve with their infrastructure? Here we outline four steps to follow.

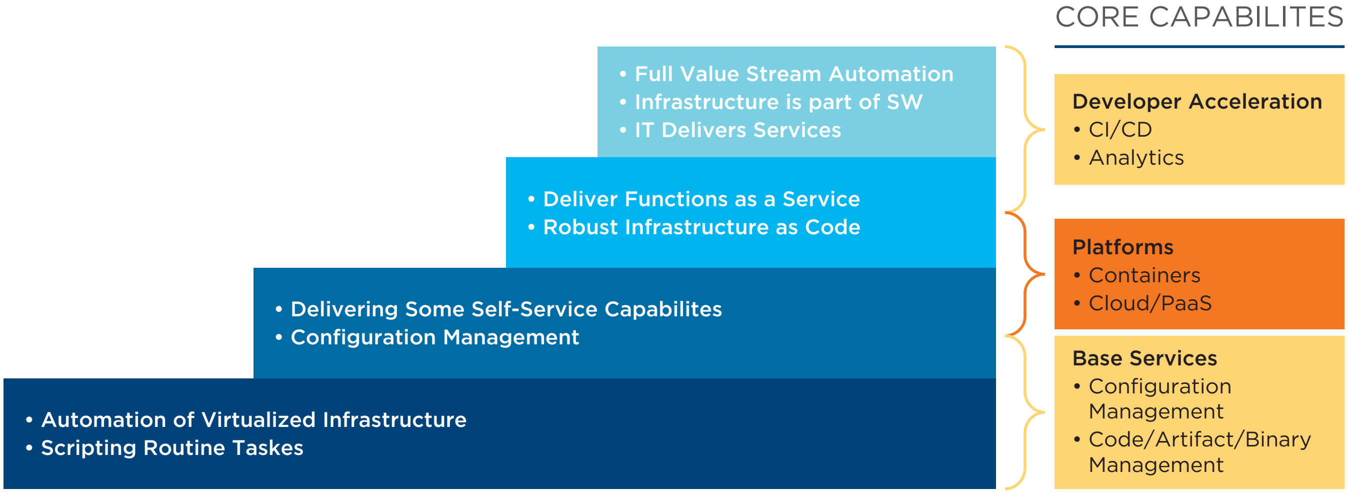

CORE CAPABILITES
• Full Value Stream Automation
• Infrastructure is part of SW
• IT Delivers Services
• Deliver Functions as a Service
• Robust Infrastructure as Code
• Delivering Some Self-Service Capabilites
• Configuration Management
• Automation of Virtualized Infrastructure
• Scripting Routine Taskes
Developer Acceleration
• CI/CD
• Analytics
Platforms
• Containers
• Cloud/PaaS
Base Services
• Configuration Management
• Code/Artifact/Binary Management

Figure 2) Path to maturity.

Step 1: Begin the Move Toward Infrastructure-Based API Integrations

This step recognizes the purpose behind infrastructure features being available through code. Here, DevOps team members who are focused on infrastructure integration should start identifying extensible infrastructure features, capabilities, and potential API integrations that could be useful for their environment and priorities. They should query infrastructure vendors to see what levels of integration they offer with the developers' preferred toolsets. They should then look to start having such components put in place as they move forward in their DevOps journey.

Examples of this work might include the ability to access infrastructure features through code instead of through an infrastructure console or GUI. Ideally, this work would include taking advantage of native integrations with common toolsets. Using such native integrations can help minimize the amount of code required. Other examples might involve using a REST protocol to access infrastructure features. Such features could then be integrated into a DevOps tool or framework of your choice if native integrations are unavailable.

Not only can Premier Eye Care now refresh its development environment in minutes, but it does so with current data that resides on its NetApp SolidFire arrays, instead of the previous night's backup. To accomplish this capability, NetApp linked Premier with services partner Wirestorm to create a Windows PowerShell script. Premier uses it to automate NetApp Snapshot™ copies of the SolidFire array for backup use in the development environment.

> "It has completely changed the way we work. When I hit the button with SolidFire, they're getting data that is live to that second. Now if I get a call at five o'clock on a Friday evening saying our internal users need the development environment refreshed, I can do it in minutes, then go home. And they're not getting data from the night before." [11]
>
> —Rob Connock, Database Administrator, Premier Eye Care

Step 2: Eliminate Hand-Offs and Promote Further Automation

Although some automation might have been done to this point, this second maturity step requires developers and infrastructure operations teams to further remove people and manual handoffs from the software develop/test/deploy cycle. Organizations should try to automate workflows that might typically involve submitting a ticket or request to another team. Menial and manual tasks should be automated—for instance, through automated policy guidelines or using infrastructure plug-ins to developer tools.

One example might involve letting an application owner or developer create a storage volume and mount it to an operating system, possibly using a Docker volume plug-in, backed by predefined operations guidelines.

> "With storage management streamlined and DevOps teams empowered by SolidFire's simplicity, storage engineers are free to pursue high-value initiatives. We're enabling our infrastructure to maintain our market dominance." [12]
>
> —Donald Talton, Senior Manager of Platform Operations and Engineering, FICO

Step 3: Build Self-Service Through Infrastructure

During this step, operations teams look more closely at how to ease self-service infrastructure delivery to consumers by further removing themselves from the equation. Development and test teams at this stage should be able to get what they need, when they need it, without human interaction with the operations or infrastructure teams. This approach might involve building self-service infrastructure systems in which technology is well integrated into developers' daily tools, whether those tools involve Ansible, Jenkins, or Kubernetes or through private cloud systems.

This approach could mean extending the use of configuration management tools (for example, Ansible) to attach or change infrastructure components such as storage directly from within the configuration management toolset.

At this stage, the role of administrators, operations, and storage teams shifts. They no longer have to carry out all infrastructure tasks themselves; instead, tasks are performed automatically through code initiated directly by the consumer. Operations staff then can focus on maintaining high-level awareness, visibility, and management of the infrastructure without having to wait for input from another team or execute manual tasks themselves. Here, security, reliability, and availability are not sacrificed, and self-service is achieved seamlessly through policy-based management and code-based integration of the infrastructure into common DevOps tools.

Examples at this phase might involve test teams using Jenkins to define a test framework. In the framework, teams might need to run multiple test bed environments against a piece of infrastructure. Using extensible infrastructure for this form of self-service could allow test environments to be provisioned or refreshed quickly, allowing tests to complete more quickly against more comparative data. The team could do all this testing without waiting for hand-offs to the infrastructure team.

Other self-service infrastructure examples might involve the use of a Kubernetes volume plug-in to give developers direct access to provision and manage persistent storage associated with a specific application. It could involve a developer directly changing the underlying infrastructure environment, without human intervention, by using other technologies such as VMware vSphere Virtual Volumes or Ansible.

> "My team was able to reduce the number of deployment systems by 10 times and drop our total deployment time from 20 hours to under a minute." [13]
>
> —Jeremy Goodrum, Vice President of Engineering, Wirestorm

Step 4: Adopt Software-Defined Infrastructure (SDI) Frameworks

When development and test teams get what they need, when they need it, without human interaction from either the operations or infrastructure team you have started to work towards a true software-defined infrastructure.

This step is about more than the ability to install software on any piece of hardware. It's about being able to set policies and frameworks that allow infrastructure to be changed on demand, when needed, to meet an application's shifting requirements, all while continuing to preserve infrastructure stability, reliability, availability, and visibility.

This phase might involve a developer who needs to change the underlying infrastructure environment. This change might involve the use of policies to promote certain infrastructure behaviors and performance characteristics, such as those associated with a specific application volume. With VMware vSphere Virtual Volumes or Ansible, teams could also use those policies to implement any necessary changes. Policy-based frameworks provide infrastructure that is adaptable and easier to manage and consume.

Helping You Get There

The journey to DevOps and the next-generation data center is not an overnight process. Nor will it be without its struggles as organizations gain the skills, knowledge, and experience necessary to try, fail, learn, and try again.

The trends, phases, and recommended milestones outlined in this paper offer a guidepost to help organizations along the way. Early in their journey, organizations should recognize the important role that extensible infrastructure plays in reaching key milestones toward DevOps.

To successfully adopt DevOps and turn IT into an innovation engine and profit center, organizations need such underlying fluidity of infrastructure to expand, contract, and adapt, as needed, to the increasing demands of developers and test/QA teams.

Infrastructure and operations teams are integral to the success of this transition. Although their role will change, their expertise is sorely needed to properly assess, select, implement, monitor, and manage the right extensible infrastructure components for the job set before them.

Such teams will gain great confidence by looking more closely now at the availability of specific extensible infrastructure features, such as plug-ins, APIs, SDKs, other integrations, and automated policy features. They should also look more carefully at partnerships between themselves, infrastructure vendors, and others that can enable them to make a more seamless transition to self-service and software-defined infrastructure that can be called or consumed through code.

Additional Resources

Learn more about the software development lifecycle and integrations with NetApp.

References

1. “Evolution Not Revolution: DevOps in the Wild,” by Boyd E. Hemphill, Feb. 4, 2015, DevOps.com, https://devops.com/evolution-not-revolution-devops-wild/. Last accessed May 3, 2017.

2. “Extensible” definition, Techopedia, https://www.techopedia.com/definition/7107/extensible. Last accessed April 19, 2017.

3. “The Short History of DevOps,” by Damon Edwards, http://itrevolution.com/the-history-of-devops/. Last accessed April 19, 2017.

4. “Walmart Boasts 213,000 Cores on OpenStack,” by Linda Hardesty, Feb. 24, 2017, SDxCentral, https://www.sdxcentral.com/articles/news/walmart-boasts-213000-cores-openstack/2017/02. Last accessed April 19, 2017.

5. “2018 State of DevOps Report,” Puppet, https://puppet.com/resources/whitepaper/state-of-devops-report. Last accessed December 7, 2018.

6. “Better Agility for DevOps Helps Polaris Alpha Accelerate Time to Market,” NetApp, http://www.netapp.com/us/media/cs-polaris-alpha.pdf. Last accessed Dec. 21, 2018.

7. “Infrastructure as Code, the Missing Element in the I&O Agenda,” by Robert Stroud, Feb. 9, 2016, Forrester, http://blogs.forrester.com/robert_stroud/16-02-09-infrastructure_as_code_the_missing_element_in_the_io_agenda_0. Last accessed April 19, 2017.

8. “We’ll Never Need to Do a Forklift Upgrade Again,” NetApp, http://www.netapp.com/us/media/cs-internet-solutions.pdf.

9. “ING DIRECT Innovates Faster with NetApp, Cisco, and Microsoft Private Cloud,” NetApp, https://www.netapp.com/us/media/ing-direct.pdf. Last accessed Dec. 21, 2018.

10. “DARZ Docker & Container-as-a-Service Drives Digital Transformation Through DevOps,” Customer Success Story: Service Provider, DARZ, NetApp, http://www.netapp.com/us/media/cs-internet-solutions.pdf.

11. “Healthcare Company Cures Storage Inefficiencies with SolidFire,” Customer Success Story: Premier Eye Care, NetApp, http://www.netapp.com/us/media/cs-premier-eye-care.pdf.

12. “Accelerated Time-to-Market Through Cloud Services Powered by Flash,” Customer Success Story: Software Services/FICO, http://www.netapp.com/us/media/cs-premier-eye-care.pdf.

13. “The New Norm for Wirestorm’s DevOps Is Fast, Ultrafast,” Customer Success Story: Service Provider/WireStorm, NetApp, http://www.netapp.com/us/media/cs-fico.pdf.

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer’s installation in accordance with published specifications.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

Data contained herein pertains to a commercial item (as defined in FAR 2.101) and is proprietary to NetApp, Inc. The U.S. Government has a non-exclusive, non-transferrable, non-sublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b).

Trademark Information
NETAPP, the NETAPP logo, and the marks listed at http://www.netapp.com/TM are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.

WP-7292-0219