



White Paper

# Platform as a Service: Apprenda with NetApp for DevOps

## Agile Deployment Model for Application Development

Bikash Roy Choudhury, NetApp  
Sasha Jeltuhin, Apprenda

February 2016 | WP-7220

### Abstract

In today's digital world, in which applications are developed and deployed to address a variety of business requirements and services, software development has matured considerably and gained increased momentum. Mobile phones and other portable devices have changed the way that personal and business data is accessed and managed globally. As the amount of data everywhere explodes, software developers are writing applications that use data to provide different levels of service for businesses and end users.

Any organization or individual that writes code must manage, protect, and own its data no matter where the software or application is developed: on the premises (in a private cloud), in a hybrid cloud, or in a public cloud. Data has volume, variety, and gravity. With its unique ability to manage and protect data, NetApp enables software development to be more seamless. Integrating NetApp with Apprenda to provide a platform as a service enables enterprise application developers to rapidly and securely deploy their code, test, and production environments with minimal friction with system and platform teams.

**TABLE OF CONTENTS**

**1 Introduction to Software Development .....3**

**2 Software Development Challenges .....3**

**3 The Evolution in Software Development .....4**

**4 DevOps in the Continuous Model.....5**

**5 Platform as a Service (PaaS).....5**

    5.1 Apprenda..... 5

    5.2 NetApp ..... 6

**6 Financial Services Customer Deployment Example.....8**

    6.1 Challenges ..... 8

    6.2 Benefits ..... 8

**7 Conclusion .....9**

**LIST OF FIGURES**

Figure 1) Software development workflow ..... 3

Figure 2) Software development evolution..... 4

Figure 3) Apprenda with NetApp PaaS stack..... 6

## 1 Introduction to Software Development

Over the past several years, there has been a constant push to optimize hardware infrastructure to perform, be efficient, and be reliable with high uptime. Features such as thin provisioning, deduplication, compression, high availability, RAID, solid-state devices, and flash transformed the infrastructure in the data center. Any organization that wrote code for its product or service tried to squeeze the last ounce of resource efficiency and performance from its data center infrastructure. Now, infrastructure has reached its optimization limit, and the focus has shifted to optimizing the software development process in the application layer.

Software development ideally involves a series of steps: develop code, build, and release or deploy. The following flow chart depicts the different phases of the software development cycle. Although writing code, or source code; building an image, or binaries or executable files; and releasing the software into production is the natural workflow, other subprocesses, are introduced. These subprocesses, such as unit tests, smoke tests, QA, and prestaging, are iterative and feed into the main codeline at various stages before the software is released into production. The importance of the subprocesses is covered in section 4.

Figure 1) Software development workflow.



*Software development* refers to the traditional development process for intellectual properties, embedded software, and system software such as operating systems and databases. *Application development* refers to next-generation applications developed for web, mobile, gaming, advertising and marketing, tools and automation, and the Internet of Things. For the purposes of this paper, *software development* and *application development* are used interchangeably.

Newer-generation application development demands that the development process be more nimble and dynamic from conception to deployment. NetApp is a popular storage choice for software development environments because of its storage efficiency, reliability, performance, and data manageability tools. NetApp is used as shared storage for all software code repositories, builds, and release versions.

NetApp® FlexClone® technology creates instantaneous user workspaces (sand boxes) with no additional storage, which provides more business agility, risk mitigation, and storage efficiency. NetApp SnapMirror® technology enables distributed software development for geo-dispersed locations and remote offices. SnapMirror also provides the agility to move data from an on-premises private cloud to on-demand compute resources in a hybrid model. SnapMirror also can move data into a public cloud (hyperscaler) environment such as Amazon Web Services or Microsoft Azure.

## 2 Software Development Challenges

During the development process, developers should have very little friction with the infrastructure and operations teams. Developers write and develop code, and they should not have to worry about infrastructure and automation. However, lack of tools and proper automation to develop, test and build, and release software in time affects developer productivity. Most of the time, developers cannot forecast the amount of resources they need to meet unexpected scaling needs in the development cycle. This issue adds significant upfront costs.

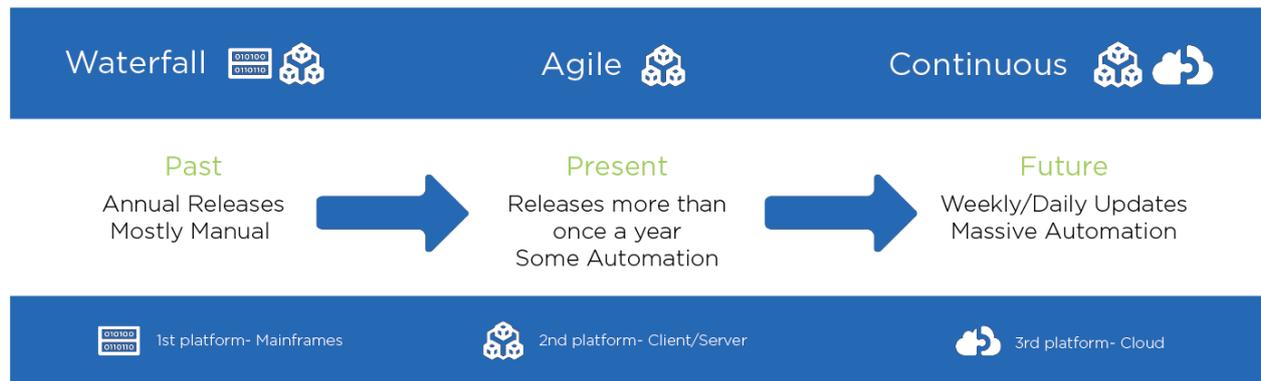
The different lines of business that develop software to drive business work with their respective system and platform teams to provide developers with the infrastructure and tools they need. Setting up, provisioning, configuring, and maintaining the entire infrastructure prolongs the development cycle for both the developer and the business. Lack of proper coordination between the developers and the IT

teams also delays projects. In addition, many resources are underutilized by developers during the software development process.

### 3 The Evolution in Software Development

In recent times, the software development process evolved from a waterfall model with manual operations and long release cycles. The process has moved to a more agile model with limited automation and multiple release cycles in a year. The agile model has been adopted by most of the midsize to large organizations that write code for products, tools, or services. The following diagram illustrates the different phases of the evolution of software development.

Figure 2) Software development evolution.



In the waterfall model, developers wrote code in a linear manner, and all the code changes were incorporated into one large codebase. Apart from the inefficiencies of provisioning hardware resources to run builds for this large codebase, the process of introducing and then identifying the errors introduced in the codebase was monotonic and less productive. There was little automation between the phases and there was less cooperation and communication with the operations team. The release cycles took more than a year and drove increased time to market.

The agile model introduced more synergy between the operations team and the developers. The systems and platform teams within IT became more responsible to stakeholders to provision infrastructure required by the developers. The developers' process of writing code linearly changed to their writing code in parallel. Each developer can now run unit tests in his or her independent workspace without affecting the integrity of the main codebase. This change eliminated the risk of polluting the contents of the original code before pushing into the main codebase.

In addition, each developer can run incremental builds from his or her workspace to validate code changes. Running unit tests in isolation also provides a healthy way to identify errors in the code changes made by every developer before merging the changes and pushing into the codeline. This process takes place ahead of the build process, reducing the chance of failure during integration tests, which mostly happen during the build process. The QA team then picks a clean version of the full build of the code and runs their set of tests. Depending on the type of software development, the software is finally released to production or goes through a prestaging phase right after the build. Limited access to prestaging is provided to customers to check the quality and stability of the software before making it generally available.

These changes still do not address the requirements for an ever-faster time to market for internally hosted and business applications. The newer-generation applications have fewer lines of code and require a shorter development cycle as well as a faster time to market. Such development is unlike the development of more traditional, monolithic software, with millions of lines of code and a more reasonable release cadence. The next phase of evolution will be the continuous model, for which time to market will be very short and happen daily or weekly.

## 4 DevOps in the Continuous Model

The next wave of software development is a continuous model, for which there is a continuous process of integration, delivery, and deployment of code. In the continuous process, changes in the code are integrated with the main repository and automated tests are run to generate build copies of the code. Continuous delivery means that, at any given time, the builds generated from the build process are ready for production, leading to shorter release cycles. In continuous deployment, every change in the code that passes the continuous integration tests is automatically deployed into production. Any software build can be promoted to a production release code.

What, then, is DevOps? DevOps is a software development process in which developers and platform and systems operations work together in continuous synergy. Platform operations include all of the infrastructure components, such as compute, network, storage, or the cloud, that need to be operational in the data center or in the cloud for system administrators to work on the physical hardware. Systems operations involve virtualizing, installing operating systems, scripting, patching, resource provisioning, and so on. Developers require a version control system for the code, builds, and application versions deployed; tools that include libraries, compilers, distribution kits, and so on; and automation and data manageability.

DevOps provides developers with much cleaner code, and less code reworking leads to faster time to market. DevOps enables more collaboration and communication between the line of business team, developers, and the operations team. Integration and automation that take place during the development workflow provide more business agility and much desired resource efficiency. Developers spend less time worrying about the infrastructure and can focus more on code development, which leads to more innovation.

Innovation, however, is not limited to developers. Automation is also a means to an end for operations teams. Automating the infrastructure layer provides more time for operations teams to innovate. Automation also enables a shorter release cycle, controlled infrastructure downtime that is more predictable, and better use of resources. All of these benefits are great for traditional software development. However, next-generation application development has more aggressive requirements. The following section provides information about why platform as a service is becoming more popular with the new breed of developers.

## 5 Platform as a Service (PaaS)

What does the shift in focus to continuous application development and deployment mean to the underlying infrastructure? If developing an application is all about code, then all the infrastructure and other dependencies should also be managed as code. Infrastructures should look and work like code, or, in other words, like a bunch of services that cater to different developer needs, such as database as a service, user workspace as a service, code pipeline as a service, build as a service, staging or deployment as a service, and so on. This policy can apply in a private, hybrid, or public cloud environment.

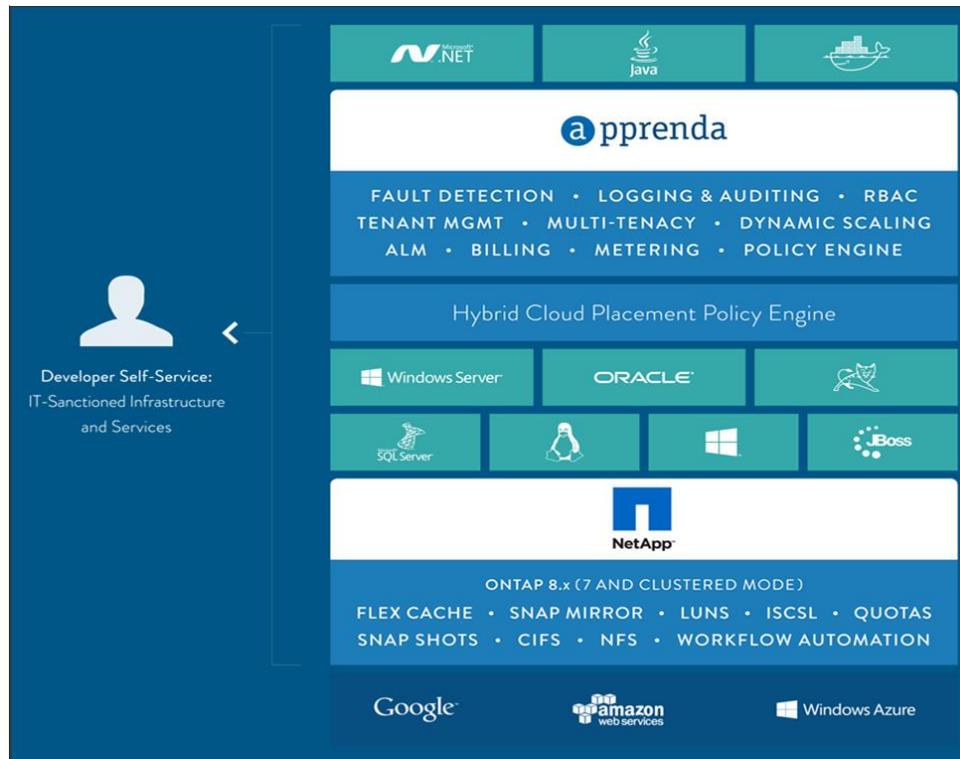
Every service layer consists of collective pieces of code that tie into the infrastructure workflow. Each piece of code performs different functions—for example, role back access control, fault detection, resource management, auditing, billing, metering, and so on. More and more of the application development process is moving to PaaS because PaaS provides options for out-of-the-box and custom automations and a standard platform for deployment. The deployment can be for writing code, building and testing, and production release of the application. Automation reduces the mundane and titive jobs performed by the operations team. Deploying the entire application lifecycle on a standard platform provides better overall application quality.

### 5.1 Appenda

Appenda gives developers and platform operators the ability to provision containerized compute and memory allocations for their applications, increasing virtual machine density. Appenda also removes the need for developers to manage infrastructure, enabling them to focus on their application's performance

and functionality. The following diagram shows how the different layers are built up in an Apprenda with NetApp PaaS solution to provide a self-service model to developers.

Figure 3) Apprenda with NetApp PaaS stack.



Many developers, however, have a growing desire to have a service-enabled process that provisions file storage. Because popular enterprise SAN and NAS systems whose extensibility feature permits such provisioning saturate the enterprise market, Apprenda created integrations to interact with enterprise storage systems and provision file storage for developers.

## 5.2 NetApp

Apprenda partnered with NetApp to provide this capability. Developers can now self-provision storage for their applications without needing to conquer the hurdles and obstacles—difficult protocols, political battles, delays, and much more—that have long hamstrung enterprise IT. And by using Apprenda’s policy-based system, platform operators and system administrators need not fear overconsumption or abuse of their enterprise storage. Configurable checks and balances are in place to prevent such misuse.

All software and application development generates data, and that data needs to be stored and managed. Even though most of the following NetApp features are available to the operations team, with adequate automation, developers can create consistent copies of their code changes and instant user workspaces for risk mitigation and business agility. Developers can also initiate consistent copies of their codebase and take back and/or replicate the data to a disaster recovery site. In addition, developers can collaborate with users in remote sites or replicate data from and to hybrid and public clouds. These key requirements are achieved by integrating the following NetApp features that are available through the Apprenda platform.

## NetApp FlexVol Volumes

NetApp FlexVol<sup>®</sup> thin provisioning helps developers:

- Avoid the costly overprovisioning and time-consuming reconfiguration that are typical with other storage solutions.
- Reduce storage provisioning time from days to minutes.
- Allocate storage on demand for LUNs and volumes across all protocols (iSCSI, NFS, CIFS, and more) according to application needs.

## Protocols

NetApp supports multiple protocols: Network File System (NFS) v3 and v4.1, Server Message Block (SMB) 2.1 and 3.0, iSCSI, and FCP. However, for application development, iSCSI LUNS and/or NFS and CIFS protocols are mainly used. Operations teams provision the database component over iSCSI LUNs, whereas the repository, journal, and logs are accessed over NFS or SMB protocols. NetApp recommends a hybrid architecture to provide the performance and simplicity needed to manage the data generated during the development process. Platform and system operators can control which protocols are made available to developers.

## Access Control Lists (ACLs)

ACLs prevent unauthorized access to particular file shares by telling NetApp which users and applications are authorized to read and write to the shares. Platform operators can control and/or harden how ACLs are set up and disable the ACL configuration for security purposes.

## NetApp Snapshot Copies

NetApp Snapshot<sup>®</sup> technology enables developers to create point-in-time copies of user workspaces and mark them with a unique change number or identifier. These identifiers are required to isolate the different consistent Snapshot copies that are taken every time the developer changes the code. Using NetApp SnapRestore<sup>®</sup> software, you can restore back to an earlier change number or identifier if the current Snapshot copy does not pass the unit or continuous integration tests. Restores can be made rapidly from any of the copies, providing developers with an exceptional recovery time objective.

## FlexClone Volumes

Developers can create instant user workspace copies to support code testing. Instead of using rsync or other tools and cloning mechanisms for a full copy of the source code, libraries, and development tools to populate user workspaces, developers can use FlexClone copies that:

- Can be created almost instantly
- Mitigate the risk of polluting the original copy of the code
- Take very little space
- Have negligible performance impact

Developers can destroy their workspaces (FlexClone volumes) after the code changes are submitted to the main codeline and the build operation has completed successfully, keeping storage efficiency intact.

## NetApp SnapMirror

NetApp SnapMirror technology gives applications developed on Apprenda the capability to protect or mirror the data in their provisioned volumes to DR sites. SnapMirror also provides the capability to replicate data into a hybrid cloud setup or into any hyperscaler that the developer chooses to use. This feature can be set up on a recurring policy to preserve up-to-date Snapshot copies of the volume.. It provides an elegant first line of defense in an enterprise disaster recovery scenario and can have data back up and running in minutes. Data mobility is the key because it provides more freedom to move data between an on-premises private cloud and any hyperscaler. There is no lock-in of data with specific service providers or a public cloud. End users have complete ownership of the data.

## 6 Financial Services Customer Deployment Example

Apprenda and NetApp provide a feature-rich PaaS solution for developers and the operations team in a DevOps environment. A large, popular financial institution successfully deployed multiple instances of Apprenda on NetApp as a PaaS solution for agile deployment and continuous development. This financial company had 430 development teams and more than 2,000 applications developed on Java and .NET environments. As the company grew its assets, it wanted to scale its ability to quickly write and develop applications that manage and provide services to its internal and external customers.

The developers were not productive using the traditional ways to develop applications. They were plagued with various inefficiencies to develop applications quickly and to control the cost of building those applications as they started to scale. Following are some of the challenges that the company had to overcome to improve application development quality and to get to market faster while lowering and controlling operating costs.

### 6.1 Challenges

The company had developed thousands of custom .NET and Java applications that run on its infrastructure. With thousands of applications and slim IT staff, and with thousands of servers, developing, running, and managing applications were not simple, inexpensive tasks. The scale of IT operations and the mission-critical nature of many of the business's core functions put the bank in a league of its own. These problems resulted in productivity loss, inefficient infrastructure spend, and lack of agility. The bank experienced:

- Long lead times for application deployment because of infrastructure provisioning and software stack build and verification
- Inflexible capacity management that required precise upfront forecasting and had difficulty meeting unexpected scaling needs
- Lack of effective cost control with large upfront cost requirements and severe underutilization of physical and virtual infrastructure
- Redundant efforts of development teams that caused developers to treat application architecture patterns, security configuration, high availability, and common services such as application caching as "one-off" engagements, rather than relying on standards
- Inability of other PaaS vendors to accommodate existing application development along with the new applications that are in the pipeline for development

### 6.2 Benefits

The PaaS solution from Apprenda and NetApp addressed most of the bank's application development challenges and provided a new approach to the bank's application development and management of its resources and services. With the new private cloud PaaS infrastructure, the bank not only significantly improved developers' productivity and resource management but kept operating costs down. The PaaS solution with Apprenda on NetApp was globally deployed with an enterprise-grade solution footprint and is now used extensively within the bank. The solution:

- Improved application time to market by 59 days. Developer productivity for application deployment was boosted 700% through standard productivity patterns and platform services.
- Increased use of infrastructure from an average of 40% to an average of 70%, resulting in a 45% drop in infrastructure costs. The bank experienced a 300% increase in infrastructure use.
- Enabled 100% uptime with no unscheduled environment outages.
- Optimized existing applications through powerful APIs and inherited cloud architecture.
- Standardized deployment and availability of standard application building blocks across development teams, resulting in massive boosts in developer productivity and agility.

## 7 Conclusion

DevOps is evolving, and it needs automation in the infrastructure layer and the ability to integrate with tools from and the application development process of different vendors. Developers who take part in the software development process want complete autonomy from infrastructure provisioning. To meet developers' needs, the infrastructure must be more agile, elastic, and scalable. The success of any development environment depends on the way operations and developers communicate as they put together a flexible infrastructure that can be quickly deployed for code, building and testing, and release/production.

Apprenda PaaS with NetApp enables IT and lines of business to align application development and operational resources in a cost-effective, agile, and continuous platform with high infrastructure utilization and high uptime. Deploying code, building and testing operations, and production on a standard platform yields better application code quality and faster time to market.

DevOps is an evolving process. The Apprenda on NetApp PaaS solution is a step toward having effective collaboration and communication between the developers and operations teams and producing smooth code integration and development to drive business.

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.

## Copyright Information

Copyright © 1994–2016 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

## Trademark Information

NetApp, the NetApp logo, Go Further, Faster, AltaVault, ASUP, AutoSupport, Campaign Express, Cloud ONTAP, Clustered Data ONTAP, Customer Fitness, Data ONTAP, DataMotion, Fitness, Flash Accel, Flash Cache, Flash Pool, FlashRay, FlexArray, FlexCache, FlexClone, FlexPod, FlexScale, FlexShare, FlexVol, FPolicy, GetSuccessful, LockVault, Manage ONTAP, Mars, MetroCluster, MultiStore, NetApp Insight, OnCommand, ONTAP, ONTAPI, RAID DP, RAID-TEC, SANtricity, SecureShare, Simplicity, Simulate ONTAP, SnapCenter, Snap Creator, SnapCopy, SnapDrive, SnapIntegrator, SnapLock, SnapManager, SnapMirror, SnapMover, SnapProtect, SnapRestore, Snapshot, SnapValidator, SnapVault, StorageGRID, Tech OnTap, Unbound Cloud, WAFL, and other names are trademarks or registered trademarks of NetApp Inc., in the United States and/or other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. A current list of NetApp trademarks is available on the web at <http://www.netapp.com/us/legal/netapptmlist.aspx>. WP-7220-0216