Technical Report

Continuous Integration (CI)/Continuous Deployment (CD) Pipeline with Apprenda on FlexPod with ONTAP 9
DevOps Process for Accelerated Time to Market (TTM)

Bikash Roy Choudhury, David Arnete, NetApp
Sasha Jeltuhin, Apprenda
January 2017 | TR-4559

In partnership with

APPREnda

Abstract

The software development lifecycle (SDLC) has been evolving over the years from a waterfall model to agile workflows and now continuous integration and deployment, also referred to as DevOps. DevOps is not a product or a technology, but a process or a pipeline that measures its success based on development speed and agility. This translates into the ability to improve better code quality and drive faster time to market (TTM).

Many organizations that develop custom applications use Apprenda platform as a service (PaaS) to deploy and run both cloud-native and legacy apps, implemented in Java and .NET and other technologies. Apprenda Cloud Platform provides support and operational controls for continuous integration (CI) and continuous deployment (CD) workflows, among many other services, which reduces the lead time for application deployments.

NetApp provides persistent data storage for applications running in containers. It also provides native functionalities such as NetApp® Snapshot® copies and cloning that integrate with CI/CD workflows to mitigate risks and improve developer productivity during application development and deployment cycles.

FlexPod® Datacenter from NetApp and Cisco provides a cost-effective infrastructure solution that delivers performance, agility, and reliability for private and hybrid cloud environments. The FlexPod Datacenter converged infrastructure with Apprenda running on OpenStack allows organizations to store, manage, and control the data that is created during the development and deployment process, with high availability, resiliency, and scalability.
1 Introduction

In this digital world, developing web and mobile applications is part of every business that wants to go to market much more quickly with its products, services, and solutions. Doing so requires an agile development method that involves a continuous integration (CI) and continuous deployment (CD) process in a fast and iterative manner. The shift toward agile development has forced business owners to be more exploratory and innovative, emphasizing speed in the application development workflows. The new types of applications are designed to provide one or many services that are high in value and customer satisfaction.

Being cloud native is all about application virtualization and infrastructure as code. It is structured around maintaining the same level of services for the application with additional operational practices for applications running on virtual machines (VMs) or on bare metal. A fundamental principle of cloud is that you build into the software architecture the intelligence about the transitory nature of the underlying infrastructure that consists of storage, network, and computing.

Cloud-native environments are highly distributed and should be elastic (on demand) in nature. Microservices are being chosen as the preferred architecture for distributed development, and containers are the unit of deployment. Self-healing is another property of cloud-native environments where any component in the ecosystem failure is marked as reserved, and the rest of the servers move into maintenance mode until the failure is corrected and the condition is restored to normal state.

2 Cloud Computing Stack

The cloud compute stack consists of the cloud application services layer, otherwise known as software as a service (SaaS); cloud platform services, also known as platform as a service (PaaS); and cloud infrastructure services, popularly known as infrastructure as a service (IaaS). Finally, the data layer is where the NetApp portfolio of products allows storing, managing, and protecting data during the entire application lifecycle. Figure 1 illustrates the different layers of the cloud computing stack.

Figure 1) Cloud computing stack X-as-a-service (XaaS) layers.

Software as a Service (SaaS)

- Delivery model for business application (pay as use)
- On-demand application with no installation and licenses

Platform as a Service (PaaS)

- Abstraction layer between developers and IT/DevOps
- Rapid deployment of applications
- Operational support for IT

Infrastructure as a Service (IaaS)

- Software-defined (SD) compute, network and storage
- Load balancers, DNS and security and identity

The SaaS model uses the web to deliver the application where the software, licenses, and deployment are handled by third-party vendors that host the application. The end users access the user interface (UI) directly from a web browser or a mobile device. This kind of cloud application service uses a subscription-based model where the users do not have to pay any high upfront license costs and also removes all the complexities of managing, installing, and upgrading any software and hardware resources. Common examples such as Office365, Salesforce.com, and Google apps are listed in Figure 1 in the SaaS layer.
The PaaS framework abstracts the underlying infrastructure and provides a serverless platform to the developers to rapidly develop and deploy applications. The underlying infrastructure, storage, network, and compute can be physical or virtualized. PaaS provides a consistent and unified application runtime environment, in which applications and application components are deployed and hosted in containers. It also provides access to shared databases, web servers, and a number of operational tools for developers and IT operators to simplify implementation, hosting, and SLA support of distributed applications (see Figure 1).

Applications deployed on Apprenda PaaS, for example, become highly available, scalable, and secure, just by nature of running on the platform. Apprenda PaaS offers the ability to “cloudify” legacy applications that were not designed to run in distributed environments. The platform automatically reconfigures guest applications to use its own distributed services such as caching, logging, scheduling, auditing, multitenant role-based access control (RBAC), and many others.

The Apprenda policy engine allow operators, typically IT ops and DevOps, to create policies according to which the workloads are deployed. These policies control the amount of compute resources dedicated to each application container and what services need to be lent to guest applications. Deployment policies also make sure that applications are automatically deployed in accordance with compliance security and other business requirements of the organizations.

Apprenda private PaaS can stitch together multiple sites into a single resource pool, within which containers can be moved across the locations as long as the move complies with policies put in place. An example of this hybrid installation mode can be an instance of Apprenda deployed on premises in a private DC and public IaaS such as Amazon Web Services (AWS).

IaaS provides access and monitors and manages highly scalable and automated compute resources along with persistent data and network capabilities without any direct dependencies on any proprietary storage. Infrastructure is abstracted and consumed as a platform using APIs and services. The new-generation web and mobile applications consume infrastructure as code in a cloud platform. IaaS is common choice for development and deployment of web-based applications. OpenStack among others is the is one of the IaaS model validated in private cloud environments. AWS is an example of a public IaaS.

NetApp as a data storage and management leader has native technologies that can integrate directly with development tools such as Cloudbees Enterprise Jenkins and container services such as Docker using RESTful APIs to horizontally scale in the compute as well as in the storage layers to significantly reduce build time and improve developer productivity. These developer tools can be part of the PaaS framework. NetApp also provides high data availability; storage space efficiency (which is extremely important in a cloud environment); and data security, encryption, compliance, and other data-level services.

The FlexPod Datacenter solution forms a flexible, scalable, open, integrated foundation for enterprise-grade OpenStack cloud deployments. The Apprenda platform abstracts the underlying infrastructure as code, thus providing a standard runtime for application development and deployment phases. Apprenda on OpenStack and FlexPod allows users to quickly provision cloud resources in an automated and secure fashion.

3 Objective

The premise of this paper is to integrate native ONTAP® 9 technologies, such as Snapshot copies and FlexClone® volumes, with Cloudbees Jenkins Enterprise using APIs to improve developer productivity, reduce build times, and optimize storage space on the OpenStack FlexPod solution for CI/CD workflow on the Apprenda platform using Docker containers:

- Demonstrate that integration with Apprenda with Cloudbees Jenkins Enterprise and NetApp accelerates the CI cycle by reducing the time it takes to complete test cycles. This in turn can be used to improve code quality, allowing for more frequent testing to be done.
- Demonstrate that NetApp and Apprenda integration allows a CD workflow where the applications can go through faster and more efficient QA/staging phases before being deployed to production.
- Integration of Jenkins with Apprenda and NetApp using Docker containers provides scalability, agility, and resiliency to the developers.
Apprenda platform on FlexPod Datacenter with OpenStack provides a turnkey infrastructure for cloud-native application development and deployment environments.

When a Jenkins master is run in a Docker container on the Apprenda platform, the integration of NetApp and Jenkins allows Jenkins slaves to connect to a code master. In this case the Jenkins master, running on Apprenda, functions as an abstraction of the underlying physical or virtual server, which improves portability of Jenkins server and improves the uptime and high availability of the CI/CD pipeline.

4 Types of PaaS

The PaaS market is highly fragmented, and there are various types of PaaS solutions. They can be broadly classified into structured and unstructured, based on the availability of operational controls, and then into private and public, based on the audience.

Structured PaaS platforms come with operational controls and encapsulate container-based application runtimes, HA and DR, configuration management tools, policy engines, distributed DBs, and a number of distributed services (for example, scheduling, logging, monitoring, capacity planning, health checks, and so on). Examples of structured PaaS platforms are various distributions of Cloud Foundry (HPE, GE, Pivotal, and IBM), OpenShift, and Apprenda.

Apprenda is a private structured PaaS platform that is extremely popular in highly regulated verticals, including financial and healthcare industries, due to its configurable deployment policy engine and plug-and-play architecture, which simplifies integration with third-party tools.

The public PaaS under the structured category (see Figure 2) applies to PaaS platforms that abstract the underlying infrastructure and provide a NoOps-like environment where the developers do not require to have any knowledge about the underlying infrastructure. Developers use the infrastructure in the form of different services provided by vendors such as AWS Elastic Beanstalk, Heroku from Force.com, and Google App Engine.

Private PaaS, in addition to the benefits for developers, offers operational controls for IT ops and DevOps. For example, operators of the Apprenda platform can manage capacity planning, define deployment policies, and provide support to guest applications on the platform.

Structured private PaaS solutions are more prevalent in enterprise customers, midmarket organizations, and service providers.

Unstructured PaaSes are typically homegrown platforms built with custom tools based on container services. Customers that have the right technical skills internally customize, manage, and maintain custom platform setups for the internal organizational consumption. Kubernetes from Google, Docker Swarm and Mesos are the most common forms of orchestration tools that are used by small startups, pockets of larger organizations, and web-scale companies as foundations for the in-house unstructured platforms.

Figure 2) Structured and unstructured PaaS vendors.
5 Apprenda: Value and Architecture

Apprenda is a major enterprise PaaS vendor that empowers agile application development in private and hybrid cloud setups. Apprenda provides runtime environments for Java, .NET, and Docker-based applications and removes many of the complexities associated with application development, deployment, hosting, and support. Apprenda provides deep support for hosting new-generation, cloud-native applications, but at the same time it offers the ability to onboard legacy applications that were not designed to run in distributed environments. This capability helps IT organizations to realize significant operational savings and enables them to innovate more quickly. Apprenda automatically reconfigures applications to use its own distributed components, which cuts down on the development time and improves time to market. It simplifies the usually lengthy deployment process through policy-driven automation. Finally, by employing container tech with automation, it helps organizations to trim operational costs: save on data center and virtualization licenses, streamline provisioning, and use compute and storage resources much more efficiently.

5.1 Value

Apprenda has its share of advantages over other PaaS vendors with its ability to integrate with a wide variety of third-party development tools such as Jenkins. The following are some of the benefits that Apprenda provides to developer CI/CD pipelines:

- Apprenda provides a standard platform for existing and new applications. It can run on any infrastructure: physical, virtual, or hybrid. It supports a wide array of OSs/VMs, databases, and application server images and integrates with a variety of third-party tools.

- Apprenda provides a single platform for private and public cloud deployments in a hybrid mode. It combines all the infrastructure resources into a single policy-driven resource pool for development teams to consume in a controlled, secure, compliant, and self-service fashion.

- Apprenda integrates with Cloudbees Jenkins Enterprise, forming a stable CI pipeline. Apprenda runs Jenkins master node in a managed Docker container with the following benefits:
  - The Apprenda platform runs the Jenkins master in a Docker container that abstracts from the underlying physical or virtual server in the same manner as it does with its applications.
  - In an event of any maintenance, the Jenkins master can be moved from a hosted VM or physical host to another one on an Apprenda platform without disrupting the CI pipeline. The seamless relocation of the Jenkins master node across hosts is enabled through persistent Docker volumes created with NetApp’s Docker Volume Plug-In for ONTAP. Jenkins Docker containers are mounted to a shared ONTAP volume.
  - Multiple Jenkins masters can scale on Apprenda’s platform with lower demands on the supporting infrastructure.
  - Managing multiple Jenkins masters on Apprenda is much easier and less complicated, and ONTAP provides persistent storage for the Docker volume to scale the number of Jenkins containers.

5.2 Architecture

The Apprenda platform consists of some of the key components in architecture, which makes it more resilient and enables out-of-the-box integration. Some of the components of the Apprenda architecture include:

- **Load managers.** On the Apprenda platform, the load manager (best described as a reverse proxy) serves as the initial receptor of incoming HTTP requests. IIS configuration on the load manager is managed by the Apprenda load manager service, a Windows service that creates and modifies URL rewrite rules as various guest app components are deployed to internal front-end content servers. The load manager service makes sure of the appropriate routing of inbound requests based on request URL patterns.

For high-availability purposes, we recommend a topology consisting of two load manager servers, which requires the use of a shared IIS configuration housed in a network share. In order to optimize performance, a hardware load balancer should be placed in front of these load managers.
Domain naming services (DNS) server. For the entire platform-based operation, the different servers communicate with each other using DNS-resolvable host names and therefore require an internal DNS structure to be in place that resolves IP assignments. It is recommended to have a dedicated DNS server for the Apprenda platform servers. An existing Active Directory controller or DNS server can be used for this purpose.

MS SQL nodes. The platform manages SQL Server instances on the client's behalf to provision and configure guest application databases. Any number of SQL Server instances can be managed by a single platform, and SQL Server instances can be added to the platform at any time for capacity. Our reference architecture includes a single SQL Server instance that is configured as a SQL Server failover cluster; such a configuration typically relies on shared storage on a SAN or NAS. We've included two SQL Server nodes in the cluster as a standard recommendation for simple redundancy. The platform manages this cluster as it would a normal SQL Server instance. Expansion of the database tier of our reference architecture comes in two forms:

- Adding SQL Server nodes to existing clusters increases redundancy capabilities of existing platform-managed SQL Server instances.
- Adding SQL Server nodes that are independent of existing clusters or adding entirely new platform-managed clusters increases the capacity of the database hosting layer of your Apprenda platform.

In addition to providing storage for guest applications, an Apprenda-managed SQL Server instance is necessary for hosting the Apprenda core database, which contains data necessary for platform functionality.

Platform repository. The platform repository is a network share that serves as the central storage location for all platform and guest application binaries. This location must be configured prior to installation, should be specified as a network path in the Apprenda installer, and must contain the following three folders:

- Applications
- Apprenda
- SAC

Configuring each folder as a separate network share is recommended as long as they are accessible through the same base path.

All guest application binaries, after being uploaded to the platform by developers, are stored in the Applications folder in the platform repository (in some parts of the platform, such as the repository browser in the system operations center [SOC], this folder is called the application repository).

The Applications folder also includes binaries for platform components that are themselves hosted on the platform as guest applications, such as the developer portal and the SOC. All other platform binaries are stored in the Apprenda folder (which is sometimes referred to as the system repository). Upon workload deployment, binaries that are needed for local execution are copied to the target server from their respective locations in the platform repository.

It is recommended to have the platform repository on an NFS share from the NetApp FAS/AFF storage system. ONTAP provides thin provisioning, data protection, and scalability with predictable low-latency performance on persistent storage.

Platform coordinator nodes. Coordination of guest application workload deployment is handled by these nodes. The servers run a custom implementation of Apache ZooKeeper, running as a Windows service. Per the Apache ZooKeeper website, ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. On Apprenda, the platform coordinator nodes maintain knowledge of the topology of guest application workloads across all nodes, as well as any shared configuration in use by Apprenda components.

An optimal platform installation requires an odd number of platform coordination nodes, because a majority of extant platform coordination nodes ($(n+1)/2$, where $n$=the number of nodes) must be up
and running in order for the platform to function properly. We suggest starting with three dedicated platform coordination nodes, which allows the platform to function as long as any two nodes are running. Additional nodes can be added as needed after the environment is up and running.

- **Cache nodes.** These nodes house the Apprenda caching Windows service, a distributed Redis-based in-memory caching implementation for all platform Windows and Linux servers. Each cache node can support one instance of the caching service per processor core; the number of processor cores/caching service instances should be used as the number of service ports specified for the cache in the Apprenda installer. We recommend two dedicated cache nodes with two cores/caching service instances each to help with load distribution and mitigate infrastructure failure risk. Coordinators may coexist with cache nodes.

- **Windows application servers.** All Windows application servers host the Apprenda Windows host service. The Windows host enables the hosting of Windows Communication Foundation (WCF) and Windows services, thereby allowing both key platform and guest application service components to be hosted on these servers. It is necessary that at least one Windows application server per cloud host Apprenda's storage controlling services, which interface with SQL Server and Oracle to configure databases.

  These servers are required to have SQL Server Management Objects (SMO) 2012 installed. At installation, the platform marks any Windows application servers with SMO installed as capable of hosting the storage controlling services and deploys this component to those servers. It installs the required SMO version on a single application server if no suitable host is found.

  In order to make sure that the storage controlling services are highly available, we recommend installing a supported version of SMO (version 11.0 or higher) on two servers that are designated as application servers prior to running the Apprenda installer, because this results in both servers being designated as storage controlling services hosts. As needed, after installation, additional application servers can be configured as storage controlling services hosts by installing SMO on the servers and then designating them as such in the SOC.

- **Windows web servers.** Windows web servers are front-end web servers that host .NET-based UIs through IIS. Using portals, the Apprenda platform allows developers and platform operators to create ad-hoc web farms for .NET guest application UI workloads at any time (as long as there is sufficient infrastructure). The sum of your Windows web servers represents the compute power of your platform specifically for hosting .NET guest application UI workloads.

  Note that all Windows web servers are capable of hosting WCF and Windows services. This is necessary for the nodes to host the presentation controlling services (see the Apprenda software inventory, later in this report), which allows management of .NET UIs in IIS. Therefore, all Windows web servers are also marked as Windows application servers (described in the preceding section) after installation, even if they were designated as web servers only in the Apprenda installer. By default, however, the platform deploys WCF/Windows services to web servers only if there are no dedicated application servers available.

  Note: The same hosts can be used as web and application servers. In addition, coordinators and cache nodes can be used as web and/or app servers.

- **Linux servers.** Linux servers host Java web application components, which are deployed and managed on top of individual Tomcat/JBoss instances by the Apprenda Linux container. Linux nodes are also used for the Docker runtime. We recommend at least three nodes for redundancy of hosting both devops tools (for example, Jenkins master) and guest workloads that are managed by Apprenda in Docker containers.

- **Oracle RDBMS (optional).** The platform manages Oracle RDBMS installations on the client's behalf to provision and configure storage for guest applications.

- **AD FS nodes (optional).** As an install-time option, the platform can be configured to support identity federation using Active Directory federation services (AD FS). If you choose to use AD FS for identity federation, we recommend creating an AD FS web farm consisting of no less than two AD FS nodes backed by a SQL Server failover cluster. Contact support@apprenda.com for information about setting up an AD FS web farm prior to running the Apprenda installer.

  Note that all AD FS nodes—including those that constitute an AD FS web farm—are automatically designated as Windows application servers. This is necessary for the nodes to host the federation service (see the Apprenda software inventory, later in this report), which allows the platform to interface with AD FS.
Apprenda Jenkins Plug-In and Apprenda Jenkins Plug-In for OpenStack are used to support the described CI/CD solution. The plug-ins are available as a part of the installation and can be installed and configured by authorized platform operators in Apprenda SOC.

Figure 3 shows how the different layers are stacked up in an Apprenda PaaS on ONTAP solution.

Figure 3) NetApp and Apprenda architecture.

6 NetApp

Apprenda’s integration with NetApp and Jenkins using ONTAP 9 APIs provides a stable and automated CI/CD workflow in a DevOps pipeline. Application developers and business owners are often challenged with lack of proper test and automation tools that are required to mitigate the risk of breaking the application logic or corrupting the data during software development cycles. Long lead times for application deployment due to infrastructure provisioning are another factor that slows down the time to market. Irrespective of how the development and deployment workflows are designed, data needs to be stored, protected, and managed. Developers do not need to understand the storage semantics. The platform abstracts the underlying storage infrastructure and automates storage provisioning for application development and deployment.

6.1 NetApp Value

NetApp has a strong storage footprint in many enterprises where code is developed and deployed in production. NetApp technologies are extensively used to accelerate agile workflows:

- Native technologies such as Snapshot copies and FlexClone volumes, along with NetApp FlexVol® volumes, are integrated to reduce build times and improve developer productivity with Appenda and Cloudbees Jenkins Enterprise.
- Tests have indicated that All Flash FAS (AFF) with ONTAP 9 provides up to a 65% space savings through compaction, compression, deduplication, and the use of FlexClone volumes. This improvement can be more or less depending on the type of builds and file system size. This is a significant storage space savings in the DevOps process because there is cost associated with every
data block consumed in a cloud infrastructure. These savings allow more work to be done with less storage.

- ONTAP APIs are used as a consumption model to abstract the storage layer during the CI/CD process.
- Running Jenkins master on a Docker container provisioned by Apprenda helps to horizontally scale, even if local storage on the servers is not scalable. The NetApp Docker volume Plug-In (nDVP) allows a container to be mounted as persistent storage to provide a complete scalable and resilient environment.

6.2 NetApp Technologies

The following are some of the key technologies that are used in the Apprenda and Jenkins on ONTAP 9 integration.

**NetApp Flexible Volumes A.K.A FlexVol**

NetApp FlexVol thin provisioning helps developers:

- Avoid the costly overprovisioning and time-consuming reconfiguration that are typical with other storage solutions.
- Allocate storage on demand for volumes and LUNs across all protocols according to application needs.

**NFS Protocol**

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 for blocks and/or NFS/SMB for file shares are common choice of protocols in cloud environments. The Apprenda Jenkins integration on NetApp uses NFSv3 as the choice of protocol and is more scalable, easier to implement, and thin provisioned by default. Operations teams provision the database and the repository over NFS protocols. NetApp recommends this 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.

**NetApp Snapshot Copies**

NetApp Snapshot technology enables developers to create point-in-time copies of source code repositories 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® feature, 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 code line and the build operation has completed successfully, keeping storage efficiency intact.

**OnCommand API Service**

OnCommand® API services provide an API through which partner applications can issue programmatic requests to execute specified data storage provisioning, monitoring, and information retrieval operations.
NetApp Docker Volume Plug-In (nDVP)

nDVP provides direct integration with the Docker containers from ONTAP 9. nDVP provisions and manages storage resources from any ONTAP platform to Docker hosts. It allows for volumes to be created and deleted and provides persistent storage over NFS or iSCSI protocol. nDVP uses NetApp-network-attached storage (NAS) driver to mount a container over Network File System (NFS) v3 for this NetApp Apprenda integration.

6.3 Data Services

Apart from the fundamental technologies that are used to integrate with the workflow, there are other critical requirements around data during the CI/CD process.

Figure 4 illustrates that NetApp provides all the data governance and security requirements along with integrated data protection in the storage layer along with service automation analytics and secure multitenant technologies:

- **Data compliance.** NetApp provides different forms of compliance such as Federal Information Processing Standards (FIPS) 140-2 not only for the active data but also for data at rest.

- **Data security.** NetApp not only provides data security with full encryption but also provides security at the protocol (NFS and SMB) layers. Irrespective of how data is accessed from the development workflows, data is secure in active state, or at rest.

- **Service automation analytics.** NetApp provides various levels of basic and compound APIs that can report, monitor, and provision data. Service-level objective (SLO)–based APIs for storage can also provide performance headroom on different storage controllers using APIs, which allows handling of data growth, scalability, and load balancing. These APIs can be provisioned or consumed as infrastructure as code or configuration as code in agile development environments.

- **Secure multitenancy.** NetApp storage can provide secure tenants that can run in the same cluster. Sales, marketing, and finance can have their own tenants and coexist in the same cluster. This keeps each tenant secure and also allows better data management capabilities.

- **Storage efficiencies.** Thin-provisioned volumes and Snapshot copies use space very efficiently. Snapshot copies are very critical to enabling consistent checkpoints or recovery points for data. FlexClone volumes are used to create near-instantaneous workspaces for development workflows that take up very little capacity relative to the dataset being cloned. Inline deduplication allows build QA test copies to use storage space efficiently because most of the build files are full copies and not
delta copies. This saves a lot of storage space. A lot of build files are small in size (<4k). Compaction features in All Flash FAS (AFF) further optimize the storage space. All the space savings add up to reducing storage cost, thus improving the ROI.

- **Integrated data protection.** SnapMirror® provides the ability to easily replicate data between different environments, including cloud instances, without requiring lock-in to any one provider. This also allows data to move into different availability zones in the cloud, even spanning different geographic locations for DR purposes. SnapVault® can also be used to archive the files for data in rest. Data can also be moved in object stores with StorageGRID® for cheaper and denser archival purposes.

### 7 FlexPod Datacenter with Apprenda and OpenStack

FlexPod is a predesigned, best practice data center architecture that is built on the Cisco Unified Computing System (Cisco UCS), the Cisco Nexus family of switches, and NetApp FAS and AFF storage systems. FlexPod is a suitable platform for running a variety of virtualization hypervisors as well as bare-metal operating systems (OSs) and enterprise workloads. FlexPod delivers a baseline configuration and can also be sized and optimized to accommodate many different use cases and requirements.

Figure 5 depicts the component families of the FlexPod solution.

**Figure 5) FlexPod Datacenter components.**

FlexPod provides a uniform approach to IT architecture, offering a well-characterized and documented shared pool of resources for application workloads. FlexPod delivers operational efficiency and consistency with the versatility to meet a variety of SLAs and IT initiatives, including the following:
• Application rollouts or migrations
• Business continuity and disaster recovery (DR)
• Desktop virtualization
• Cloud delivery models (public, private, and hybrid) and service models (IaaS, PaaS, and SaaS)
• Asset consolidation and virtualization
• Data center consolidation and footprint reduction

Cisco and NetApp have thoroughly validated and verified the FlexPod solution architecture and its many use cases. In addition, they have created a portfolio of detailed documentation, information, and references to assist customers in transforming their data centers to this shared infrastructure model. This portfolio includes, but is not limited to, the following items:

• Best practice architectural design
• Workload sizing and scaling guidance
• Implementation and deployment instructions
• Technical specifications (rules for what is and what is not a FlexPod configuration)
• Frequently asked questions (FAQs)
• NVAs and Cisco Validated Designs (CVDs) that focus on a variety of use cases

Cisco and NetApp have also built a robust and experienced support team focused on FlexPod solutions, from customer account and technical sales representatives to professional services and technical support engineers. This support alliance provides customers and channel services partners with direct access to technical experts who collaborate with cross vendors and have access to shared lab resources to resolve potential issues.

FlexPod supports tight integration with virtualized and cloud infrastructures, making it the logical choice for long-term investment. As a key partner in the FlexPod Cooperative Support program, Red Hat OpenStack platform is fully supported by Red Hat.

### 7.1 Red Hat OpenStack Platform on FlexPod

IT organizations are rapidly adopting a cloud services model for all IT services. OpenStack is an open-source virtualized infrastructure and management framework that offers a variety of compute, storage, and networking services with a common API layer that helps customers deliver self-service capabilities to end users and streamlines management and operations for IT staff. OpenStack removes dependencies on the underlying physical hardware, enabling highly flexible environments that better meet the needs of newer cloud-native application architectures. Red Hat OpenStack platform is an industry-leading distribution of OpenStack that delivers all of the power and flexibility of OpenStack along with enterprise-class support.

FlexPod Datacenter is the ideal physical platform for OpenStack, offering unmatched flexibility, scalability, and nondisruptive operations. NetApp FAS and AFF storage systems offer industry-leading scalability, performance, and data management. Cisco UCS servers have high-density CPU and memory options and policy-based management that enable highly efficient provisioning and management of compute resources. Cisco Nexus switches include high-bandwidth interconnect capabilities, and both Cisco UCS and Cisco Nexus have certified OpenStack Neutron drivers to enable dynamic network configuration from the OpenStack interfaces and APIs.

Both NetApp and Cisco have partnered with Red Hat to integrate the installation of all necessary drivers and services into Red Hat OpenStack platform, enabling streamlined deployment and lifecycle management using Red Hat OpenStack director. After the deployment is complete, FlexPod Datacenter with Red Hat OpenStack platform delivers highly flexible and automated IaaS.

For more information about FlexPod Datacenter with Red Hat OpenStack platform, refer to TR-4506.
7.2 Elastic Grid with Apprenda and OpenStack

Apprenda on OpenStack provides an elastic grid for applications that are developed and deployed in cloud environments. Apprenda policy engine maps with the OpenStack security groups to securely segment both the IaaS and PaaS layers from different workloads generated during the build/test and deployment phases.

Cost is an important factor in a cloud environment. Apprenda with its policy engine can grow and shrink compute resources (Nova VMs and containers) on demand for building, testing, and validating applications. This provides a high level of resource efficiency.

On an event of any maintenance activities such as troubleshooting VMs or upgrading physical nodes, Apprenda has the ability to move applications nondisruptively running on containers from one physical host or VM to another.

Figure 6 illustrates the FlexPod, Red Hat OpenStack platform, and Apprenda architecture as a standard runtime environment for developers writing code and deploying applications.

Figure 6) Apprenda and OpenStack on FlexPod architecture.

7.3 Automating Apprenda on OpenStack and FlexPod

Installation of the Apprenda platform on OpenStack and FlexPod is fully automated. The automation is based on a series of postinitialization scripts that are passed along to the compute API and are executed by the cloud-init script on each new VM orchestrated for Apprenda. The end user provides several input values using a simple user interface of the included platform orchestrator. The configuration values required by the orchestrator are:

- OpenStack parameters (API URL, tenant ID, user credentials, and desired API versions) to communicate with OpenStack API
- Information about the domain that underpins Apprenda (domain name, suffix, IP of the domain controller if it exists), desired image and variety of the VM for the new domain controller, network, and security group
- Information about the platform (platform version, license, and admin credentials)
- Defined list of Apprenda nodes:
- Load manager
- Three coordinator nodes
- Two distributed cache nodes
- One SQL Server DB node
- Desired number of Windows hosts and Linux hosts for the container runtime

- For each node, following parameters need to be provided:
  - Image ID
  - Variety ID
  - Network
  - Security group
  - Floating IP (for load manager only)

After all the mandatory parameters are filled out, the orchestrator will create all the required core nodes and requested compute nodes for Apprenda, configure storage for the platform, and install the core services and agents on the new VMs. It will also apply the software license and make the Apprenda platform available for immediate use. Out of the box, the platform portals and APIs will become available from any VM on the Apprenda domain. To enable external access to the platform, a DNS record needs to be created and pointed to the floating IP of the load manager VM of Apprenda.

### 8 NetApp and Appenda Integration

The NetApp Jenkins integration with Appenda enables developers to implement a robust CI and CD workflow on the same platform. The Docker engine managed by Appenda along with nDVP provides persistent storage to the code and the applications being developed and deployed, further reducing the lead time and allowing for faster time to market. This integration provides horizontal scaling both at the compute layer and at the storage layer. NetApp storage also provides storage space efficiency and high availability and uptime for the data generated during the agile workflows.

#### 8.1 CI Pipeline

The continuous integration (CI) pipeline consists of the CI environment, developer environment, and build artifact environment. All of these environments are hosted in the Apprenda platform.

Figure 7) Automating CI workflow with Apprenda Jenkins on NetApp using Docker.
As illustrated in Figure 7, the Jenkins master is running in a Docker container managed by the Apprenda platform mounting an NFS volume from NetApp storage. This allows Apprenda to move the Jenkins master to another VM or physical node to either retire the VM, upgrade the physical node, or perform other maintenance activities without losing data and affecting the CI pipeline.

All the other components, for example, software control management (SCM) tools such as GIT, CI or development branches, user workspaces, and the build artifact, are running in sibling Docker containers and communicating with the Jenkins master on port 50000. All the Docker containers in the CI pipeline in the Jenkins master are mounting volumes from NetApp for persistent storage using nDVP.

The Apprenda platform has to be customized to support this architecture. Some of the configurations that are important to the NetApp and Apprenda integration include:

- **Apprenda Jenkins plug-in.** Pushes the application archive with application artifacts to Apprenda for deployment.
- **Application bootstrap policy.** Enables the user to enforce deployment based on certain conditions such as type of application and the properties tied to various servers in the platform. In this architecture, the policy would allow the creation of Docker containers on only those hosts that support Docker, are specially mapped to NetApp shared volumes, and have enough resources (CPU, memory) to accommodate the application.
- **Resource policies.** Allow the user to set CPU and memory limits by a particular container type.
- **Deployment manifest.** A standard element in Apprenda deployment packages that encapsulates all configuration values used by the platform for the purposes of deployment. The file is in XML format. It is uploaded to the platform in a zip file (application archive) along with all other application artifacts (binaries, scripts, config files, and so on).

The deployment manifest file that supports the deployment of Jenkins master nodes holds the following information: name and location of the Docker image for Jenkins, port references, volume mappings, and the resource policy that must be utilized for this application. Refer to the appendix for an example of the deployment manifest. The platform creates and deploys a container with Jenkins and stores the Jenkins home directory on a NetApp NFS share. This directory is used to perform builds and keep compiled build artifacts related to various jobs.

**Source code management (SCM).** A version control system that manages various different versions of the code during the development process. SCM tools such as GitLab or Perforce may be used to manage the different code revisions. Multiple code repositories can coexist in a single NetApp volume and can be managed by a single SCM tool. For example, we can have Jenkins and Android open-source software (OSS) in the same NetApp volume, and GitLab can manage both these repositories. Having a local SCM volume on NetApp has multiple benefits:

- Avoid compute and network resource overhead for every “git clone” operation from a local or private hosted code repository.
- Taking Snapshot copies on the local recall repository provides data protection from any accidental code corruption or loss and recovers very quickly to a stable state.
- Identifying errors in the code from an unsuccessful or build failure is quick using “git bisect” without any resource overhead.
- As a best practice recommendation, no builds are performed on this SCM volume. Only different versions of code are checked out and checked in to this volume. This eliminates any resource overhead from high build traffic by different users.
- Further, NetApp provides compaction and deduplication of the source code for improved storage space efficiency, which can provide significant space and cost savings in a cloud environment.

**CI or development branches.** CI or development branches can be used interchangeably. These Docker containers mounting NetApp volumes check out the entire source code from the SCM or a particular branch. If the code base is small, the entire code can be checked into the CI containers, or else branching the code base is recommended. These CI containers can also represent different code bases that are managed by the SCM tool.
From the example stated earlier, there can be a CI container instance mounting a NetApp volume with Jenkins open source and another container instance of Android open source code. These container instances can be built independently, using specific tools depending on the applications. Java applications will, for example, require “maven” and a specific version of Java Runtime Environment (JRE).

Incremental CI and nightly builds are performed in their respective CI containers that mount persistent NetApp volumes in isolation, while the SCM tool manages all the code revisions in different directories (repositories) in the same NetApp volume. The CI instance of each code branch has all the tools, RPMs, libraries, compilers, and other dependencies along with the source code and the .jar or .war file (if it is Java source code) after a successful build completion.

A Snapshot copy of the NetApp mounted volumes on the container instances after a successful build guarantees a consistent point in time of the entire environment with the source code base and other precompiled and compiled objects.

**Developer workspaces.** Can be instantly cloned from the last successful build of the code base in the CI or the development instances. The developer automatically has the entire environment with the source code and all the dependencies in their user workspaces. The users make changes to the code and submit the changes to the code repository in the SCM volume. This has multiple advantages that improve developer productivity significantly:

- The user workspaces are instantly cloned using NetApp FlexClone volumes and mounted on Docker containers.
- The ownership of the files and directories is also changed to the respective user instantly.
- This eliminates multiple streams of “git clone” or “rsync” process to create multiple copies of the source code and the compute and network overhead in that process.
- These cloned workspaces are extremely space efficient. These take a small fraction of the actual size of the CI or the development instances.
- Because the user workspaces are clones from a parent CI volume that contains precompiled and compiled objects, the user builds limit the lines of code change and do not perform a full build all the time. The incremental builds reduce the build time considerably.

**Build artifacts,** The container instance mounts a NetApp volume to store all the build artifacts such as the tools, libraries, compilers, and RPMs that are required during a build process. The entire contents of the successful builds from the CI or development volumes are zipped and copied into the build artifact volume. The zip file can be used by the QA/staging teams at a later stage, or developers may want to replicate a scenario from an old build to recreate and fix bugs or errors reported by users. The zip file consists of the entire environment of a QA or a developer to run tests. Even if the tool versions may have changed as the applications start to evolve, recreating the environment with the right set of tools in retrospect should not be a challenge with the zip files in the build artifact volume.

Refer to [TR-4547](#) for more details on the CI workflow on ONTAP with Jenkins plug-in using APIs.

### 8.2 Continuous Deployment

Continuous deployment (CD) is an automated process of deploying applications into production. Apprenda reduces the friction between building and deploying applications. The NetApp and Apprenda integration allows organizations to define and perform user acceptance tests (UATs) and publish applications at scale in a single location and across multiple hybrid clouds. This integration allows businesses to accelerate the process from designing the user requirements to finally deploying the application to production.

The continuous deployment pipeline starts after the code is built successfully in the CI phase and an executable file in the form of a .war file (if it is a Java-based application) is created. As mentioned earlier, the content of the entire CI or the development instance is zipped and copied into the build artifact volume. However, there is a small enhancement made during this process to automate the CD process with Apprenda. A clone is created from the last successful CI build volume and is temporarily mounted on the container along with the build artifact volume.
The Apprenda platform requires an application archive file for the deployment. The archive has a specific folder structure, as shown in Figure 8. It consists of a deployment manifest that has a subfolder structure that has to be unzipped; the newly generated .war has to be copied in the “wars” folder and then zipped. A series of git commands are performed to move this Apprenda zip file into a specific location under the SCM tool directory. The Apprenda Jenkins plug-in now automatically looks for the archive file in the new location mentioned in Figure 8 and finally deploys it to the platform. The new application instance, containerized on Apprenda, is now available for testing and verification that can be performed by any appropriate tools, standalone or integrated with Jenkins. After the application instance is verified, the containers are discarded as a part of the clean-up process. This entire process is automated to improve the deployment experience with Apprenda on NetApp.

After the Apprenda archive (zip) file is moved into the appropriate SCM folder location, the entire contents of the CI volume that was cloned are zipped and copied into the build artifact volume, and the temporary clone and the container are deleted. However, the build artifact volume still has all the artifacts and the build zip files that are stored on NetApp persistent as storage. This process is repeated for any new build that needs to be moved to the build artifact volume.

Refer to the Jenkins plug-in page on GitHub at https://github.com/NetApp/Jenkins-Plugin for more information.

9 FlexPod and Apprenda Integration: Value for DevOps

Apprenda on OpenStack integrated with ONTAP using APIs in a FlexPod has many advantages from which developers and operations teams can benefit. This joint solution makes sure that organizations and businesses can deliver an agile development workflow with CI and CD for the development and QA teams and deploy applications more quickly to achieve better customer satisfaction:

- Apprenda can provide an accelerated runtime environment for applications when the data is accessed from the underlying storage more quickly. All Flash FAS (AFF) from NetApp with the next-generation SSDs provides the best performance and capacity with higher IOPS and predictable low latency that are required during parallel builds:
- Creating instantaneous sandboxes for developers and during the QA process to develop and test code improves the developer's productivity significantly.
- The CI workflow described in section 8 enables developers to reduce build times and the infrastructure resource efficiencies with Apprenda on NetApp.

- Transformation from the traditional software development to a more agile development workflow is a goal for many organizations and businesses. During this transformation several things are important for business or application owners:
  - Legacy applications can coexist with the new cloud-native applications.
  - Reducing infrastructure costs along with storage is often a requirement.
  - Horizontal scaling during development is as important as it is during deployment because organizations want to be able to handle bursts of developer demand.
  - High uptime of data during development, deployment, and postdeployment phases.

- All of these requirements can be achieved with Apprenda and FlexPod:
  - Apprenda has the ability to support both legacy and new cloud-native applications.
  - Apprenda and OpenStack on FlexPod provide an out-of-box experience that reduces the lead time of deploying applications. This also helps organizations that do not have the resources with the right skill set to support and maintain the infrastructure.
  - Apprenda and OpenStack on FlexPod provide faster provisioning of infrastructure such as compute, network, and storage with high uptime and optimized performance.
  - Apprenda, Red Hat, Cisco, and NetApp have a common support agreement for customers that are using the FlexPod solution.
  - The FlexPod and Apprenda integration proves the horizontal scalability with Docker containers on the compute and instant cloned volumes on containers in the storage layer.
  - All the primary and cloned volumes in the storage layer are thin provisioned and have significant space savings throughout the SDLC. These space savings contribute to drive overall costs down.
  - FlexPod provides six 9s as uptime for data uptime and availability. Apart from high availability, it also provides various enterprise-grade data protections and disaster recovery technologies to protect, manage, and control the data.

Apprenda has the ability to run on public hypervisors such as AWS, as well as on the premises and in hybrid cloud environments. If an application is developed on premises and needs to be deployed in AWS, Apprenda can deploy the application on AWS by moving the container in which it is built to AWS. NetApp has the ability to move the data between on-premises and various cloud environments, including ONTAP Cloud in AWS. In this way applications and data can easily and seamlessly be moved across private and public clouds based on the needs and priorities of the business and in accordance with compliance and security requirements.

10 Conclusion

Apprenda is extremely popular in the financial and healthcare sectors for Java and .NET and many other programming languages that are used for developing and deploying applications on the same platform. Apprenda has the ability to provide consistent support to both legacy and new cloud-native applications with significant savings to the business. The storage layer in FlexPod, not only stores and protects data but also manages and controls the data that is created during the SDLC. It also protects the data and provides scalability and uptime with very little storage footprint. The joint NetApp and Apprenda solution improves developer productivity, reduces build times, and provides storage space efficiency that drives costs down. The following data points summarize the benefits of the joint NetApp, Apprenda, and Jenkins integrations:

- Ease of use:
  - Using Apprenda’s PaaS capabilities, developers can easily create development workspaces that take advantage of the latest container technologies.
- Developers can continue to use existing CI/CD tools such as Jenkins and any preferred integrated testing tools without disrupting their current application workflow processes.
- Developers don't have to know infrastructure; the PaaS takes care of this for them. All developers need to specify are requirements, and the PaaS takes care of the rest.
- No special skill set is required for businesses to invest to manage infrastructure and Apprenda and OpenStack on FlexPod, thus reducing lead times for deploying applications.

- **Efficiency:**
  - Allows 2 to 3 times more developers to be supported on the same infrastructure due to lower demands for compute and network resources.
  - Through the use of Snapshot copies, FlexClone volume demand for storage can be decreased by 20 to 40 times.
  - The addition of compaction to ONTAP 9 can reduce storage requirements in the code depots and during volume creation by up to 66%.

- **Productivity:**
  - Faster workspace creation means developers can become productive 60x more quickly. For medium to large development teams, this can translate into hundreds of hours of added productivity every month.

- **Manageability and choice:**
  - Because the features of ONTAP provide a standard platform, users are not limited to where they build or deploy applications. Furthermore, they have the freedom to migrate their data and applications across public and private cloud platforms without sacrificing function or requiring changes to their applications or data.
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 © 1994–2017 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, 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.