White Paper

Atlassian Bitbucket Data Center on NetApp for Scalable DevOps

Scalability, Collaboration, Staging, and Disaster Recovery for Bitbucket (Git) on ONTAP (ONTAP 9, ONTAP Select, ONTAP Cloud)

Bikash Roy Choudhury, NetApp

September 2017 | WP-7256

Abstract

The Atlassian stack consists of a wide variety of tools that perform specific functions in the software development lifecycle (SDLC) to enable product, service, and content delivery into production. The most popular tools, for example, Confluence, JIRA, and Git Bitbucket, are used in data centers and cloud environments for development and delivery workflows.

NetApp® ONTAP® provides a standard data management platform for applications and workflows that require data availability, scalability, data protection, and recovery from failures. The robustness of the Atlassian stack is complemented by ONTAP features such as thin-provisioned volumes, Snapshot™ copies, clones, storage-efficient mirrors, and a stable shared file system.

NetApp also uses these Atlassian tools in production, and this paper provides different integrated architectures for scalability, staging, remote collaboration, and disaster recovery for Git Bitbucket used in NetApp production environments. Similar architectures using ONTAP features might apply for other Atlassian tools.

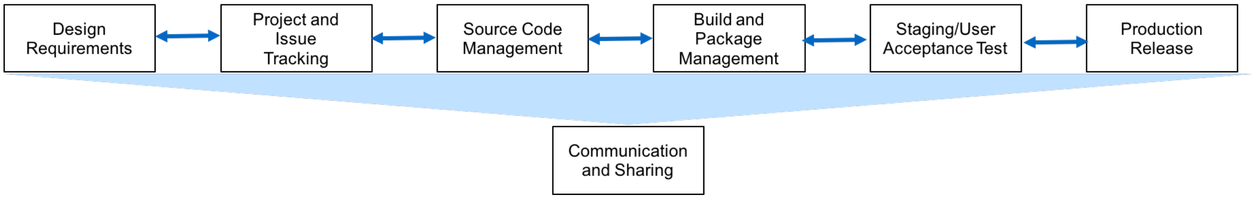
TABLE OF CONTENTS

LIST OF FIGURES

1 Introduction

In a software development lifecycle (SDLC), the development process goes through a series of steps before the application is released into production, as illustrated in Figure 1. Precise use cases are determined, and design requirements are scoped. Collaboration within the development team, as well as with the business or application owners at the beginning of a project, is essential in order to successfully deliver business value to the end users. Code development starts as soon as the design specifications are locked in and a new project is kicked off to track the progress and the issues during the development cycle.

Figure 1) Software Lifecycle Management


Design Requirements
Project and Issue Tracking
Source Code Management
Build and Package Management
Staging/User Acceptance Test
Production Release
Communication and Sharing

Different software revisions are managed and tracked by a version control system for different iterations of code changes. The developers write code. This is followed by a build and test phase before new code is committed and changes are integrated into the main code base. Upon successful compilation and packaging of the source code, the final build is deployed for staging or user acceptance tests. This is typically the last phase before code is released to production. This paper focuses on using Atlassian tools in an agile and continuous integration/continuous delivery (CI/CD) workflow.

Atlassian provides different tools for product, content, and services development that are commonly used by many small, midsize, and enterprise organizations. The Atlassian stack consists of a suite of tools that provide different benefits to the developers and project owners to improve productivity and collaboration among team members. The most common tools used from the Atlassian stack are Confluence, JIRA, and Bitbucket.

The Atlassian stack enables the agile and continuous integration of workflows by simplifying and automating the development and deployment of applications. Data that is generated during the SDLC has to be stored, managed, and protected by any business and asset owner. Project owners and development team members require scalability, collaboration, staging (preproduction testing), and data recovery from any loss or disaster during the development and deployment process.

NetApp provides a standard data management platform to provide vertical and horizontal scalability, high availability, and accessibility of data based on performance requirements and service-level objectives. ONTAP 9, ONTAP Select (software defined), and ONTAP Cloud are three different NetApp data management platforms that run on hardware appliances such as fabric-attached storage (FAS) on the premises, on virtual machines (private or hybrid clouds), and in hyperscalers such as AWS and Azure, respectively. ONTAP provides not only data management capabilities such as protection and portability, but also storage efficiencies that can provide better return on investment (ROI), along with other forms of services such as data security, governance, and compliance.

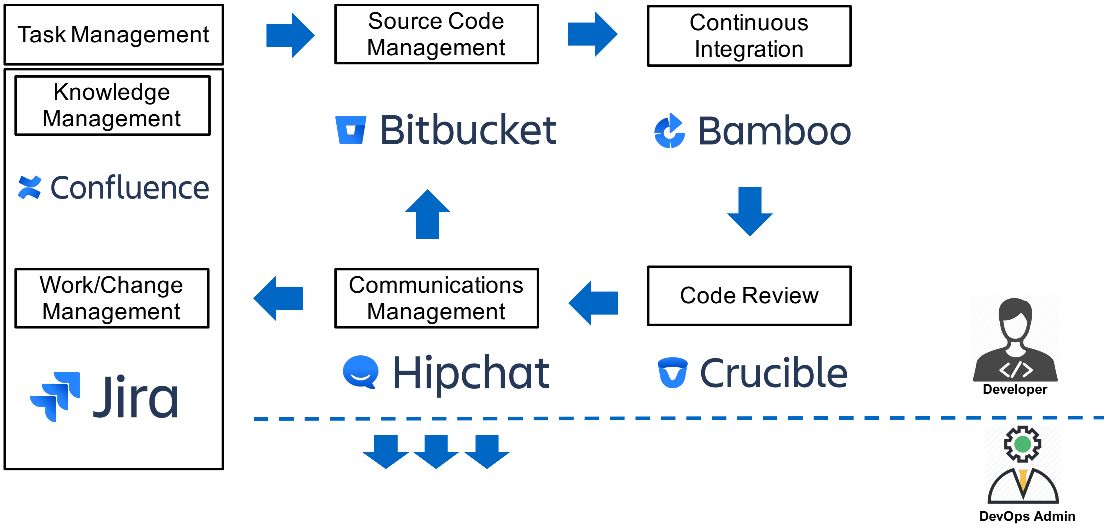
The Atlassian data center deployment option for tools (for example, JIRA, Bitbucket, Confluence, and others) is designed for high availability, performance at scale, and instant scalability when hosting these applications in your own data center. This document describes how native ONTAP features complement the Atlassian application resiliency provided by Atlassian tool stack.

2 Atlassian Tools

NetApp as an enterprise user of the Atlassian suite of products uses the following six tools in the development environment: Bitbucket (Git), Bamboo, and Crucible are for agile or continuous integration workflows; HipChat is for collaboration; and Confluence and JIRA are for task management. Figure 2 provides a workflow diagram showing where in the development cycle these tools are typically used:

- **Confluence.** This tool replaces the traditional "wiki pages" to allow the organized documentation of use cases and requirements. Confluence enables exchange and knowledge sharing between development teams and project owners.
- **JIRA.** This is one of the most popular and commonly used task management tools, along with Confluence from the Atlassian stack. JIRA tracks the progress of a project and issues reported and solved by the development team members. The Confluence tool uses plug-ins or add-ons to communicate automatically with JIRA. This combination of tools allows software incidents, changes, and new feature requests to be logged and simultaneously directed to the developers directly responsible for an application.

Figure 2) Flow diagram of phases that use some of the Atlassian tools in application development

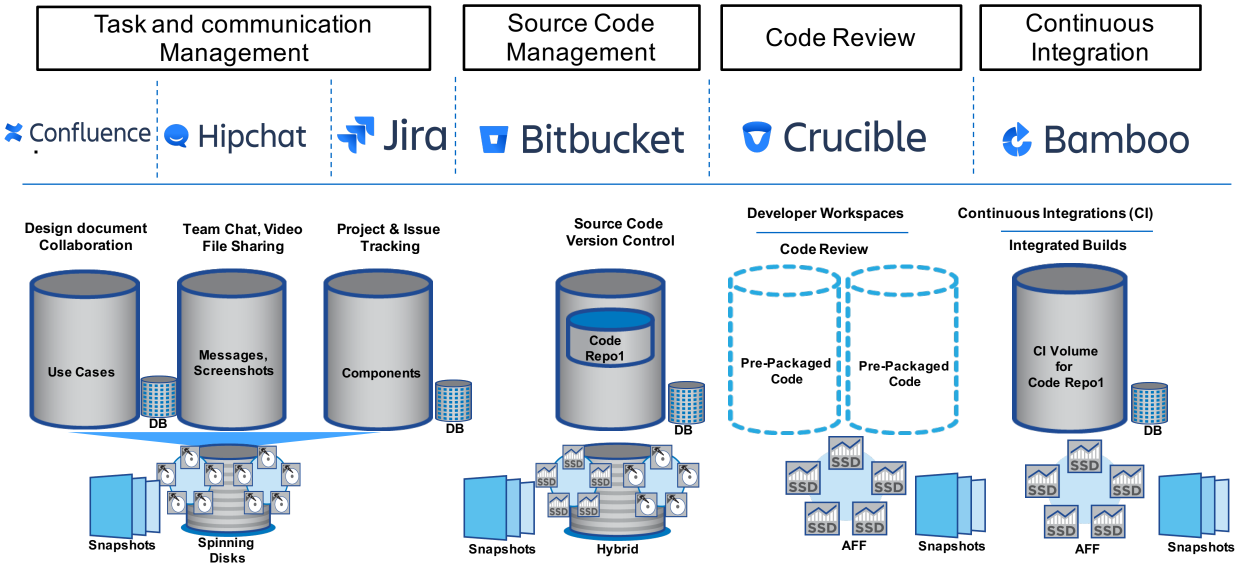


- **Bitbucket (Git).** Bitbucket is Atlassian's solution for source code management (SCM) and serves as a version control system. This tool, formerly known as Stash, manages the different source code revisions. Bitbucket has two versions: server and data center. Bitbucket server is a single instance running on a single machine with limited features and capabilities. Bitbucket data center provides scalability and high availability and shares the code in a distributed manner using smart mirroring. Bitbucket provides "pipelines" for code manageability for continuous delivery and large file storage (LFS) for storing large files such as audio samples, videos, and graphics. Bitbucket also comes with built-in support for code reviews. In this paper the focus is on Bitbucket data center.
- **Bamboo.** This is a tool for CI workflows that feed build, test, and deploy activities. Bitbucket has a plug-in to communicate with JIRA to automatically open bugs or incidents from a failed or unsuccessful build. JIRA then tracks the bug until it gets resolved.
- **Crucible.** Provides workflow-based code review and assigns code reviewers to approve the changes before the new code is checked in to the main code base managed by Bitbucket. Crucible can be used to review code for other version control systems such as Perforce and SVN.
- **HipChat.** HipChat is an instant messenger that allows developers and DevOps admins to instantly communicate among themselves internally. HipChat provides notifications to development teams and enables improved collaboration to complete projects efficiently.

3 Data Management and Storage Efficiencies with ONTAP

ONTAP provides a powerful data management platform that can be implemented in multiple cloud environments (private, public, and hybrid) to optimize infrastructure costs and provision workloads based on performance and security service-level requirements. Atlassian tools leverage native ONTAP features for scalability, staging, collaboration, and disaster recovery of data during the SDLC. NetApp uses all the Atlassian tools (discussed in section 2) in its own development environment. Figure 3 illustrates NetApp’s deployment of the Atlassian tools with ONTAP in a CI process.

Figure 3) CI workflow using Atlassian tools on ONTAP



Bitbucket (Git) administrators and users are always challenged with having the database and the repository or file system on a shared storage that can scale, provide security to the data in use and data in motion, and run more workloads with a smaller storage footprint. The following ONTAP features are used to integrate with Confluence, JIRA and Bitbucket (Git) and complement the best practices suggested by Atlassian and further improve the overall productivity, performance, and efficiency of Bitbucket and tools in the Atlassian stack during the development and deployment cycle.

- **Shared file system.** Network File System (NFS) v3 is the most common shared file system used to mount shared databases, repositories, and/or CI builds for the different tools illustrated in Figure 3. NFS in ONTAP is unique for the following reasons:
  - NFSv3 support is native to ONTAP. Exported file systems can be mounted on any UNIX client that follows RPFC 1318 without any external dependencies such as emulators and so on.
  - NFSv3 locking is external to the protocol stack. ONTAP, with its central lock manager, provides effective file-locking capabilities in multiprotocol (NFS, CIFS, FC, ISCSI) environments. ONTAP supports any advisory and/or exclusive locks set by the applications.
  - In highly secure environments, security features for authentication (Kerberos with AES and 3DES encryption) and authorization (access control lists) and more granular control for permissions with export policies on volumes and directories are also supported.

  Apart from the advantages of using NFSv3 on ONTAP, Bitbucket derives these added benefits from using NFSv3:

  - NFSv3 provides for fast and flexible scaling with performance. For Bitbucket (Git) users, this allows for faster file access for development teams regardless of their size or location.

  - Improved security. Git does not natively support security features such as Kerberos. For businesses that require more security for user authorization to access data, ONTAP provides Kerberos support in NFSv3.
  - Shared databases can be mounted separately on NFSv3 in data center tools, so if one of the databases is offline, other applications can still be operational. Allowing for separate databases also provides more control for data protection and recovery in the event of data losses or corruption.
- **Thin-provisioned volumes.** A typical development project requires a developer (or an associated ops person) to specify an amount of storage space they believe they will need to complete a project. After this request has been made, this storage space is allocated for the developer's exclusive use. Whether the reasons are uncertainty, fear of underestimating, or a lack of a connection between the request and the cost of said request, usually more storage than is needed is requested. The result is overprovisioning, in which excess capacity sits unused as a buffer due to a lack an upfront understanding of actual requirement. From the developer's standpoint, they should not have to concern themselves with this issue. From an operations standpoint, this creates major planning, management, and cost issues for what is often considered a fixed pool of storage capacity.

  Thin provisioning is designed specifically to address this issue. The application consumes only the data blocks written to the storage and not the actual space allocated by the logical volume. Thin provisioning refers to a set of technical capabilities that include inline compression, deduplication, and compaction that allow a fixed amount of physical storage to appear to be several times larger than it is in use, sometimes as much as 5 to 10 times larger. In practice, thin provisioning can allow 10TB of physical storage to be allocated to appear like 50TB or more. Further, the logical boundaries between how a storage space, or volume, is used can automatically and dynamically shift to meet the requirements of each user without affecting other users in the same physical space. The feature that relies on these thin-provisioning capabilities is called FlexVol® volumes.

  FlexVol volumes take advantage of ONTAP dynamic storage provisioning capabilities by provisioning more workloads for less storage footprint that reduces storage cost. Also, volumes can be dynamically scaled up or down (with the addition or removal of physical capacity) to meet requirements. This is referred to as "autogrow." An added benefit of thin provisioning is the reduced overhead it requires to set up and allocate workspaces. Not only does this save on costs (less storage functioning like more storage), it also translates into faster provisioning times (hours or days to minutes) and a better overall user experience, not to mention fewer headaches for DevOps engineers and admins.
- **Snapshot copies**. ONTAP developers to create point-in-time read-only 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. In developer vernacular, Snapshot copies are also called checkpoints.

  Snapshot copies can be initiated on demand to a level as granular code changes/updates. Bitbucket can also trigger a Snapshot copy tied to a change number (or SHA#). Using the ONTAP SnapRestore® feature, a file or volume can be recovered back to an earlier Snapshot copy. This ability is particularly helpful in the event of an outage or any form of corruption to the file or files being worked on. NetApp Snapshot copies are also highly efficient. By taking advantage of thin-provisioning features (see earlier) and data "pointers" instead of actual data copies, a Snapshot copy needs only take of a fraction of the size of the file or volume it represents. This translates into addition efficiencies, which developers experience as speed, and operations people see as saving on resources.
- **FlexClone® volumes.** Similar to a Snapshot copy, a FlexClone volume is a full copy of a dataset. The difference is that a FlexClone volume can be read/write. FlexClone volumes take advantage of NetApp thin provisioning and pointer-based data-mapping technologies to allow users to provision fully addressable read/write copies of any type of data (source code, builds, production data, and so on) in a fraction of the space of the master data it represents: often less than 1%. Contrast this to a

feature such as GitClone, which also allows files and volumes to be duplicated, but they also take up the same amount of space as the file being cloned.

The efficiency with which FlexClone volumes work allows developers to create instant read/write workspace copies to support code testing, as shown in Figure 4. Instead of using “git clone” for a full copy of the source code, libraries, and development tools to populate user workspaces, developers or DevOps engineers can clone large code bases, or even production data (several GB in size) in seconds instead of minutes or even hours. FlexClone volumes can significantly speed up workspace creation. Also, in dev/test phases, FlexClone volumes can allow for larger test datasets to be used, improving the veracity of the testing process and resulting in improved overall quality. This improves developer productivity and reduces infrastructure costs by adding source code, build, test, and CI artifacts.

Developers can use FlexClone copies that can scale and:

- Can be created almost instantly
- Mitigate the risk of polluting the master codebase
- 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.

Finally, teams using NetApp FlexClone volumes and Snapshot copies for continuous integration can quickly recover issues due to bad code or merge conflicts. Restores can be made rapidly from any of the copies, providing developers with an exceptional recovery time objective.

- **SnapMirror.** ONTAP SnapMirror® technology gives applications developed using Atlassian tools the capability to protect or mirror the data in their provisioned volumes to DR sites:
  - SnapMirror 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 as a recurring policy to preserve up-to-date Snapshot copies of a volume. It also provides an elegant first line of defense in an enterprise disaster recovery scenario by enabling data backup and running to be done 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 specific data lockin.

For the remainder of this white paper, the use cases and benefits of using ONTAP technologies and efficiencies with Bitbucket (Git) are discussed in the following sections. Although Bitbucket is the basis for these use cases, similar benefits are also gained using ONTAP with Atlassian Confluence and both JIRA’s data center and server versions. The benefits described are based on NetApp’s experiences as an enterprise user of the Atlassian suite of products. The purpose of this paper is to share our insights gained and best practices learned from using the Atlassian tools in conjunction with ONTAP and the key features and capabilities described in the first half of this paper. The use cases are anonymous but represent a set of consolidated experiences captured from NetApp and NetApp customers.

Bitbucket (Git) data center and servers are available in Amazon Web Services (AWS) as a self-managed instance that allows scalability, performance, and availability. However, the scalability may be limited with Elastic Block Storage (EBS) volumes and escalating costs by overprovisioning Elastic Compute Cloud (EC2) instances to obtain optimum performance.

ONTAP Cloud is available in AWS as an Amazon Machine Image (AMI) that functions as a data management layer not only in AWS but also to connect to any on-premises Bitbucket (Git) instance to provide multi-cloud portability and flexibility to tier data to Simple Storage Service (S3). ONTAP Cloud provides NFS as a standard protocol to scale seamlessly and can reduce the EC2 costs by offloading a lot of processing power to the data management layer with instant FlexClone volumes. SnapMirror can work as a transport layer to various endpoints such as different availability zones or to an on-premises data center or self-managed colocation to have a reduced mean time to recover (MTTR) for any kind of disaster.

4 Bitbucket (Git) Integration with ONTAP Use Cases

Bitbucket (Git) integrated with ONTAP provides a variety of use cases for scalability, collaboration, preproduction staging, and disaster recovery that apply to private, public, and hybrid cloud environments. The following sections elaborate on the data management and storage efficiencies of using Bitbucket on ONTAP.

4.1 Bitbucket (Git) Environment

Bitbucket in the Atlassian data center environment includes one or more Bitbucket servers that run on virtual machines (VMs) or on physical nodes. These are configured with a shared database and shared file system that consist of various Git repositories. This setup includes a pair of high-availability (HA) proxy nodes and a node for elastic search. For the purpose of this paper, VMs are used for setting up the Bitbucket environment:

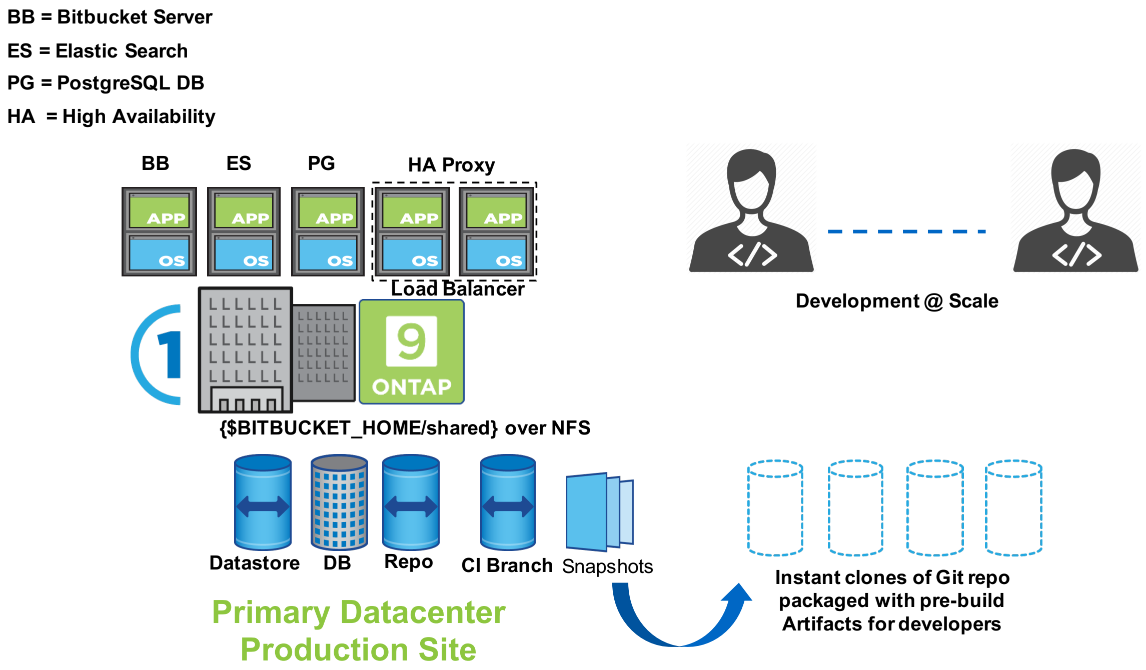
- **Shared database.** As mentioned in section 3, the shared database volume is mounted over NFSv3 on ONTAP. The database consists of metadata such as the repository name, managed accounts and permissions, settings, and so on that are related to users and projects. It also includes pull request reviews and comments and linked information from JIRA/Bamboo that Atlassian recommends for use with PostgreSQL. This paper outlines the setup with the database running on PostgreSQL for Bitbucket.
- **Shared file system or repository.** The repository is configured on a separate NFSv3 volume for data protection, scalability, and performance reasons. The repository consists of different code revisions, the branch history, and changes that are stored as unique hash or SHA#s. The shared file system also stores configuration files, attachments, and the plug-ins needed during the development process.
- **Load balancer.** Two nodes (VMs) are used to set up an HA proxy. This is used as an interface for the users from a public network to access the Bitbucket server behind the firewall inside a private network over HTTP. The end users use HTTPS to reach the HA proxy where the SSL is terminated, and plain HTTP requests are directed to the Bitbucket server. This is done for performance reasons.
- **Elastic search.** This module is part of the Bitbucket installation. It provides a smarter and faster way of searching code, file names, commits, project keys, and so on using wildcards, regular expressions, and so on. This search capability does not require any additional storage on ONTAP.

4.2 Scalability

Atlassian recommends using NFSv3 for shared databases and file systems for running data center tools such as JIRA, Confluence, and Bitbucket. Bitbucket data center is licensed by user count, not by the number of servers or CPUs, so environments can be scaled without additional licensing costs. To attain a scalable solution, manageability, performance, and usability are the key attributes along with adding hardware or infrastructure resources.

Scenario 1: An enterprise customer that generates 6M+ IOPS/day on a single Git Bitbucket server (128 cores, 512GB memory) where 4,500 developers are doing multiple "`git clone`" operations per day. Every time a "`git clone`" operation is performed, it generates a "`pack`" file that is very CPU intensive and does a memory-to-memory copy operation from the server to a user desktop. During peak hours, the Bitbucket server experiences a demand surge similar to a boot storm and becomes a bottleneck due to the high volume of "`git clone`" operations and other Git process requests, such as encryption for HTTPS and SSL.

Figure 4) Bitbucket (Git) scalability architecture on ONTAP



As a common process, additional VMs are added to the Bitbucket (Git) cluster to mitigate the single server bottleneck. However, patching and load-balancing the additional server resources to obtain an acceptable performance result in additional management and logistical overhead.

As shown in Figure 4, ONTAP technologies such as thin-provisioned FlexClone volumes can clone the CI environment that uses tools such as Jenkins from the latest Snapshot copy to instantaneously create a developer workspace. This can be done at scale (all developers) without taking any additional time, space, or performance overhead. The developer workspaces are unique to each user and contain source code, precompiled objects, and binaries:

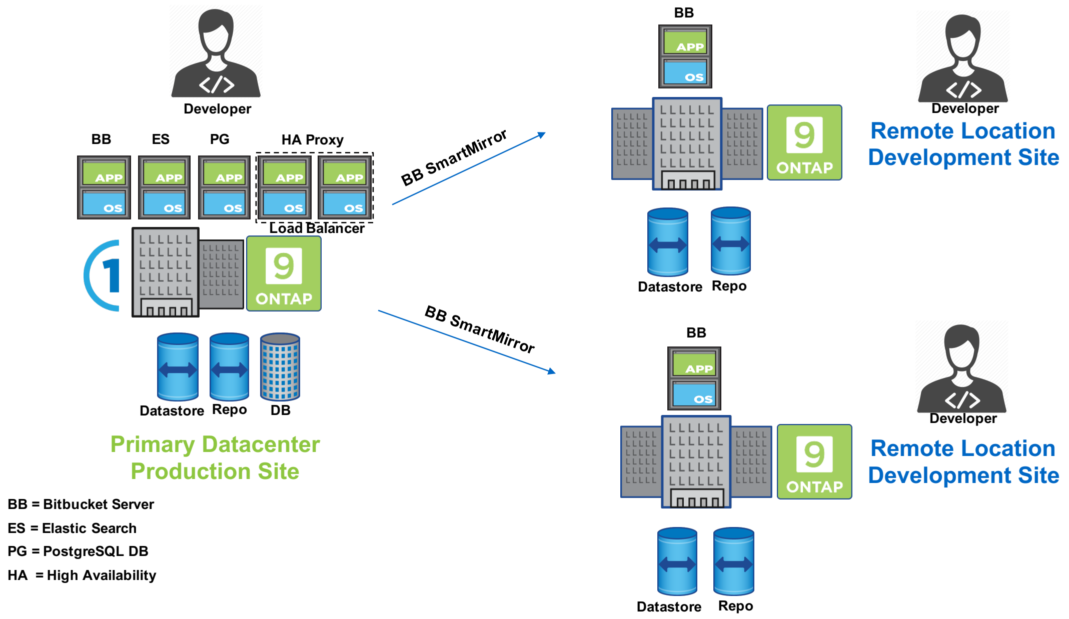
- Using ONTAP FlexClone capability not only provides efficient, seamless, nondisruptive scaling for workspace creation and code distribution, but also can help improve developer productivity by removing the provision bottlenecks and dramatically reducing developer wait times. The "`git clone`" operations can be offloaded to ONTAP to minimize the Bitbucket server footprint and optimize how the infrastructure is used to support developer demands. The user workspaces from ONTAP can be mounted on individual user home directories, and users can use SSH or an integrated development environment (IDE) such as Eclipse to connect from their desktops.

- The "`git clone`" operation does not clone to a custom webhook provided by the DevOps admin. Developers have to create their own webhook for their development and test cycles. This runs counter to the recommended best practice of standardizing different custom webhooks for developers working on the same code base. With FlexClone volumes, the custom webhooks can be made available in the workspaces during the FlexClone process, thus providing a high degree of manageability of a standard set of webhooks that can be shared by all the developers in the team working on the same code.

4.3 Remote Site Collaboration

Most midsize to enterprise organization that do software development have teams spread out in different geographical locations. Remotely located developers performing “git clone” operations often deal with WAN latency and prefetching delays at these remote sites. Atlassian Bitbucket data center provides smart mirror technology to mirror an entire repository corresponding to a project on which the users are working to remote sites to accelerate "git clone" and fetch operations locally.

Figure 5) Bitbucket (Git) remote collaboration architecture on ONTAP



In Figure 5, there is a single Bitbucket server at every remote location that is connected with an active smart mirror session to a main data center. Each of the local Bitbucket mirrored nodes is mounting a FlexVol volume over NFSv3 for storing the part of the repository mirrored from the main data center.

The local user credentials are authenticated by the central Bitbucket database on the main data center. The mirrored nodes at the remote site provide a read-only copy of the code. All the code changes (only) made by the remote users are written directly to the repository in the main data center. After the changes are committed to the main repository, the mirror nodes in the remote sites are synchronized and updated with the new changes in the code.

The new code change updates on the mirror nodes take additional space at the remote sites. ONTAP FlexVol volumes in the remote sites allow the mirrored nodes to autogrow the volume on demand. This eliminates the overhead of configuring additional disk groups for repositories set up on VMs or physical nodes for incremental storage. Thin-provisioned FlexVol volumes provide the flexibility to grow and shrink storage space on demand for repositories mirrored to each of the remote sites, thus providing automatic provisioning of storage and removing the manual and time-consuming process of data management.

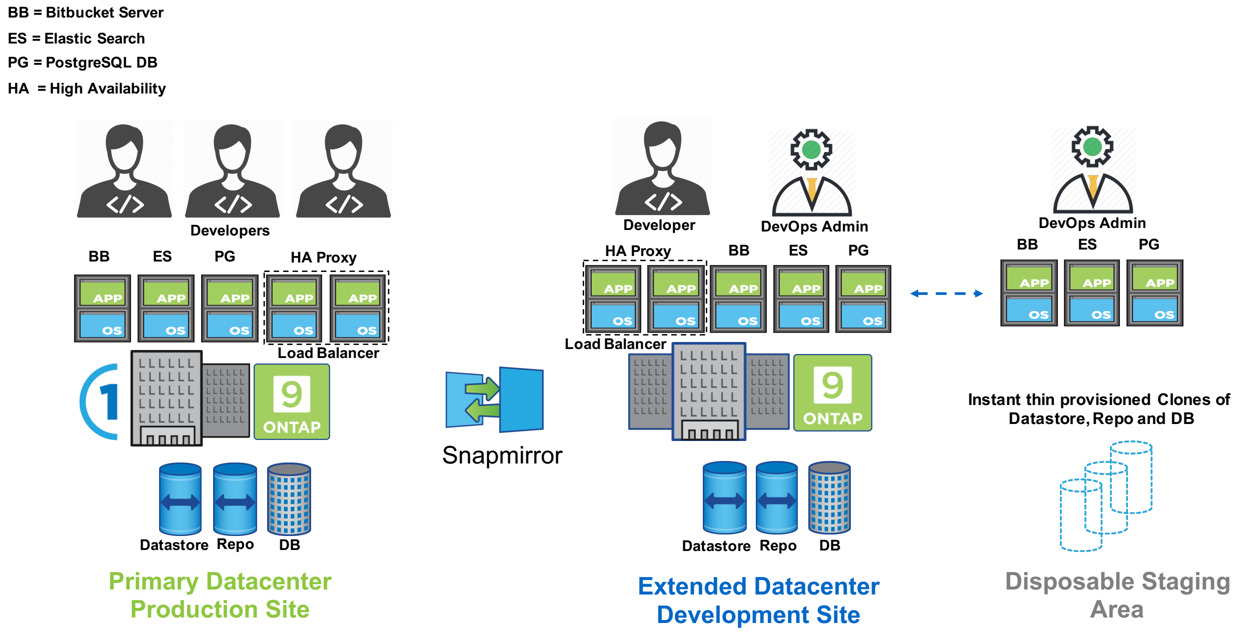
4.4 Staging (Preproduction Testing)

Staging is a process of previewing and testing any new code, feature, patch, plug-in, add-on software, or configuration change before it is deployed to production. Staging is an essential step used to reduce or migrate the risk associated with making changes of any type and scale to a production system. Atlassian recommends a three-tier architecture for setting up a Git Bitbucket environment: production, development, and staging for enterprise environments:

- Production is a live environment with very high availability and reduced downtime with stable code changes.
- Development is a testing ground for developers and DevOps admins from different business units with new scripts and third-party software integrations before requesting production rollout.

Staging is a preproduction testing environment used by DevOps admins to test different upgrades and procedural changes such as patches, packages such as RPMs, and so on before rolling them out into production.

Figure 6) Bitbucket staging architecture on ONTAP



The ideal recommendation is to take a backup of the production environment and copy the entire backup for development and staging. This process is time and space consuming and does not incrementally update the development and staging environments with new changes. This process can negatively affect productivity and increase the lead times required to deploy the changes to production.

In Figure 6, the three-tier architecture when rolled out on ONTAP provides agility, flexibility, and scalability and mitigates risk, all with a high degree of storage efficiencies. A SnapMirror copy of a production site provides a mirrored copy of the data in the development site that asynchronously updates the changes from production. The mirrored copies of the shared database and file system preserve the storage space savings (compression and deduplication) of the production (master) copy.

The developers and DevOps admins then perform a FlexClone copy of the mirrored copies in the development site. This converts the read-only copies to read-write copies, all without requiring the use of additional storage space. After testing is complete, the test and staging environment can be disposed of and recreated on demand after new mirrored updates are taken from production. During staging, if a test fails, the user can use the SnapRestore feature to quickly revert the test clone to the previous Snapshot copy to rerun the tests. In this way, both deployment and rollback risks are reduced. Development teams can deploy with the confidence that they have a stable recovery point on which to fall back if required.

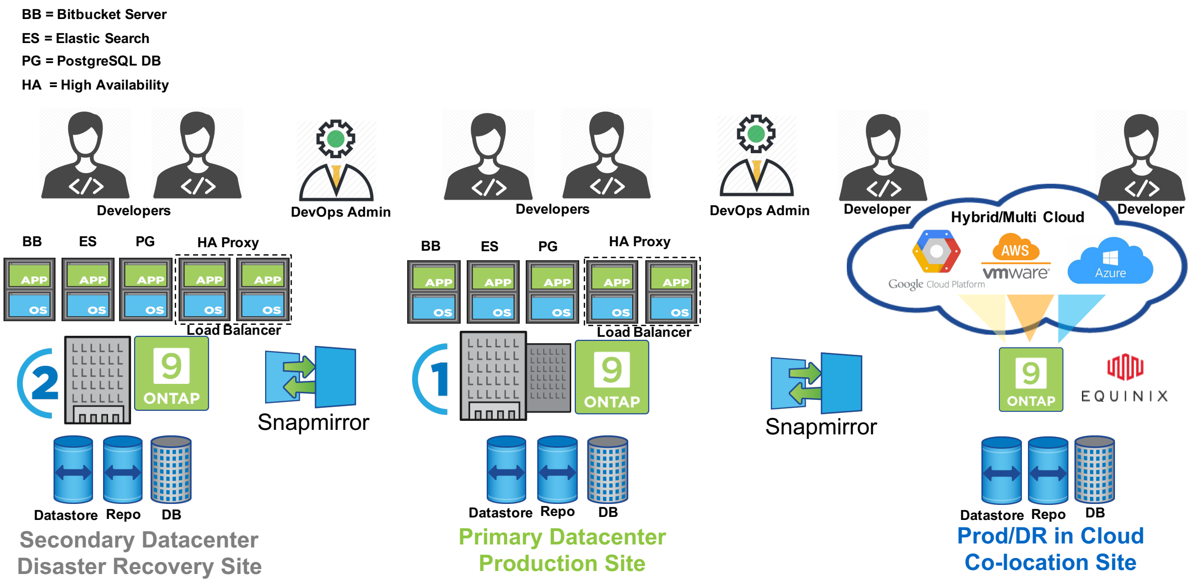
4.5 Disaster Recovery

Unlike high availability, which applies to an individual component failure, disaster recovery is an active-passive scenario where operations move to the secondary site if a disaster hits or the the production site goes completely offline and there is a loss of access to an application and/or data. A secondary site for disaster recovery is typically designed around three main measurable objectives: recovery point objective (RPO), recovery time objective (RTO), and recovery cost objective (RCO):

- RPO means how updated the DR site is from the last changes in the production site for the application to recover from a failure.
- RTO indicates how quickly the DR site can come online with respect to the application version, data updates, and network changes.
- RCO relates to the cost associated to set up a disaster recovery environment.

These objectives are set by a combination of business and IT owners. In some instances, regulatory and compliance, legal, or industry requirements may dictate what these must be. The MTTR to restore an application or a service when a disaster happens is one of the common challenges that the business owner has in the business process. The solution that NetApp describes in this section provides an MTTR of 30 minutes or less.

Figure 7) Bitbucket (Git) Disaster Recovery (DR) architecture on ONTAP



To provide for RPOs, deploying DR environments using ONTAP allows data to be asynchronously copied using SnapMirror from a production site to the secondary site and keep the changes updated through incremental mirror copies. Based on the data churn at the production site, mirror updates are typically set up on a predetermined schedule. Depending on the WAN latency to the secondary site, the volume and rate of data change, and the business RPO for data, the mirror intervals are set up. These intervals can range from days or weeks to as brief as every few minutes. The size of the largest expected change combined with time to transmit these changes (a function of transmission speed and latency) typically determines the shortest feasible update cycle.

In the data center model, a Bitbucket server node that consists of a shared database and file system is mirrored to the secondary site. The application-consistent Snapshot copy for a consistency group (PostgreSQL and repository) is taken using a NetApp tool such as SnapCreator®.[1] The public and private network and the HA proxy settings along with the required server nodes are all configured at the DR site. As soon as the DR site comes online, the users accessing the Bitbucket URL to the production site are redirected to the DR site. This setup provides a quick RTO.

[1] SnapCenter support for PostgreSQL is not currently supported, but is planned for in the SnapCenter product roadmap.

As shown in Figure 7, ONTAP can also run in a colocation facility such as Equinix in a hybrid/multicloud environment and also in a hyperscaler such as AWS and Azure. Organizations can use SnapMirror to copy their data to the colocation and can use compute and network from any cloud provider such as AWS or Azure for surge requirements, elasticity, or to optimize infrastructure costs. In this scenario, an organization retains complete ownership of its data, and there is no cloud vendor lockin. If an organization does not have infrastructure and the budget for a DR site, similar benefits can be achieved using a hybrid/public cloud that allows for the use of ONTAP and ONTAP DR and data protection features. Using ONTAP, the DR site for Git Bitbucket can be set up anywhere. The organization has the flexibility to choose a secondary data center or a hybrid/multicloud environment to provide the best RCO.

Conclusion

Atlassian tools run in production at many business organizations. The toolset has a vast range of capabilities and benefits for development teams of all sizes, as well as project owners, in different phases of the development and deployment cycles. Data manageability and availability are critical elements of any well-designed software development lifecycle. ONTAP, with its native features and functionalities described in this paper (FlexVol volumes, Snapshot copies, FlexClone volumes, SnapMirror, NFSv3 support, and so on), enables easy and seamless integration with the Atlassian development toolchain. Tools such as Bitbucket (Git) integrate with ONTAP to not only accelerate development and delivery processes, but also provide an application-aware data-driven process to improve the reliability and quality of the intellectual property and business assets of the organization.

Atlassian tools integration and hosting on ONTAP back-end storage provide superior performance, efficiency, resiliency, and data protection. Organizations benefit from improved developer productivity, improved IaaS efficiency and ROI (from data center through to the cloud), improved control over operational metrics associated with IT performance, improved control over code and development assets, and reduced business and IT risk through improved risk mitigation and the ability to quickly recover from issues should they occur.

References

The following references were used in this paper:

- https://confluence.atlassian.com/bitbucketserver/clustering-with-bitbucket-data-center-776640164.html
- https://confluence.atlassian.com/bitbucketserver/disaster-recovery-guide-for-bitbucket-data-center-833940586.html
- https://confluence.atlassian.com/bitbucketserver/scaling-bitbucket-server-776640073.html
- https://confluence.atlassian.com/bitbucketserverkb/how-to-establish-staging-server-environments-for-bitbucket-server-779171160.html
- https://confluence.atlassian.com/bitbucketserver/getting-started-with-bitbucket-server-and-aws-776640193.html

Version History

Version	Date	Document Version History	Author
Version 2.0	August 11, 2017	Atlassian data center tools on NetApp for DevOps (edits)	L. Bunka
Version 1.0	August 5,’ 2017	Atlassian data center tools on NetApp for DevOps	Bikash Roy Choudhury

Acknowledgements

Names
Daniel Holmes, Atlassian Tools Architect, NetApp
Kumarswamy Namburu, Senior Manager Development Tools and Services, NetApp
Lawrence Bunka, Product Manager, DevOps, NetApp

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

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.

Trademark Information