Technical Report

Remote Site Replication and Collaboration with JFrog Artifactory and NetApp StorageGRID Webscale

Software distribution with ObjectStore for instant replication and collaboration of builds with improved performance and low cost

Bikash Roy Choudhury, NetApp
Vishnu Vardhan, NetApp
Jainish Shah, JFrog

May 2017 | TR-4596

In partnership with

Abstract

NetApp, together with JFrog Artifactory, one of the leading binary management tools, has developed an integrated solution that allows large enterprises with globally distributed development teams to instantly share new and updated code artifacts with development teams around the globe.

With this solution, development managers can be confident that their development teams are always in sync regardless of where and when code updates are posted. Synchronization helps to avoid teams working on wrong code versions, enhances collaboration, lowers risk for errors and speeds time to market (TTM).

This solution offers greater synchronization speed at a lower cost, without the need to rely on cloud-based object repositories, allowing your company to maintain full control over valuable binary assets.

If your organization develops applications in more than one location, and keeping your development teams constantly updated with the latest builds is important for both code deployment and support, this document is a must read.

TABLE OF CONTENTS

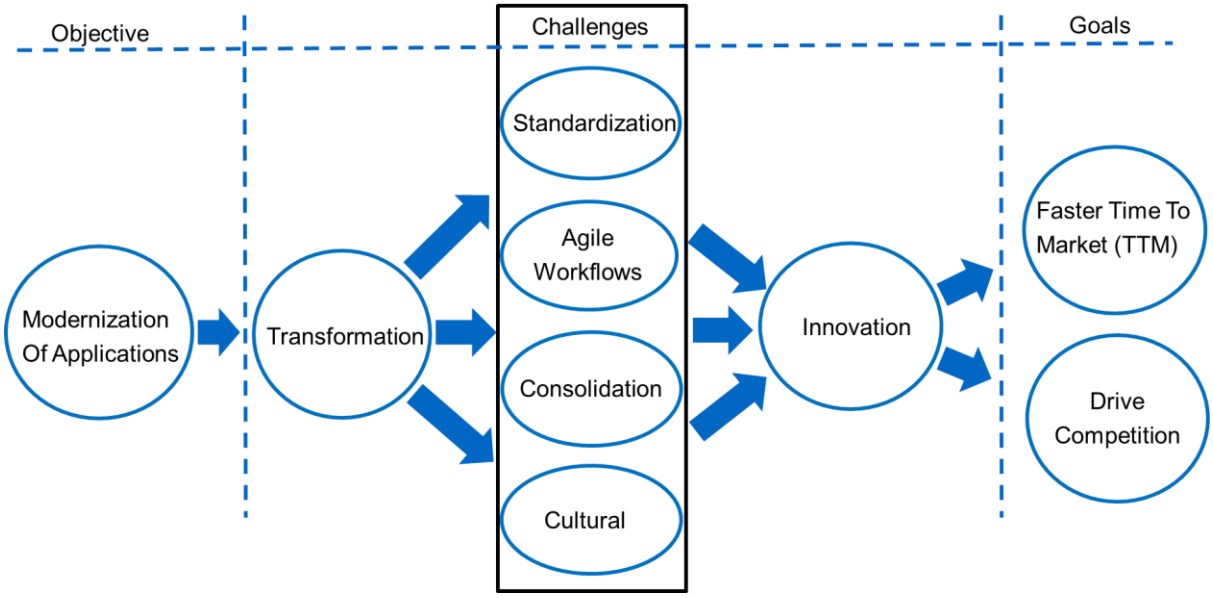
LIST OF FIGURES

1 Introduction

DevOps is the natural evolution of the development culture for developers, operations teams, and business stakeholders. The adoption of DevOps as an approach to accelerating application time to market has become a strategic initiative for most organizations, with adoption now at 78%.[1] In this digital world, developing 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 delivery or deployment (CD) process in the organization's environment.

While driving the DevOps process in the environments, every organization has different development teams who use their own set of tools for developing and deploying applications. Most of these artifacts are nonstandard and are scattered all over the organization without adequate maintenance and support. Tools could be obsolete and might not be the right choice for developing new generation applications.

Figure 1) Where challenges can occur in the path to DevOps adoption.



As illustrated in Figure 1, there is an operational challenge with many organizations to standardize and consolidate the tools, libraries, RPMs, and other dependencies that are required to build and package applications in the overall transformation and modernization of the applications. Adequate application packaging leads to better product quality that can run on any platform. This helps businesses to get to market more quickly with their products, content, and services and drive competition.

2 Binary Artifact Manager and Data Management

A binary artifact repository such as JFrog Artifactory is the leader in universal binary management tools that businesses use today. Artifactory works as a central hub for all the dependencies that are used to compile code and package applications that can be shared and accessed

[1] "2017 State of Cloud Report," RightScale.

by different development teams who use common tools and repositories. This provides a consistent runtime for applications developed and deployed on a standard platform.

With the different versions of the source code, the new versions of application builds are generated, stored, and managed by JFrog Artifactory. Although the storage footprint for the JFrog Artifactory database that contains the metadata (version, license, package name, and so on) is smaller, the sizes of the prebuild artifact files and binaries are larger. The number and size of binaries keep growing with the different versions of source code and the type of tools required to compile and package the application.

Figure 4) Continuous integration and continuous delivery workflow.

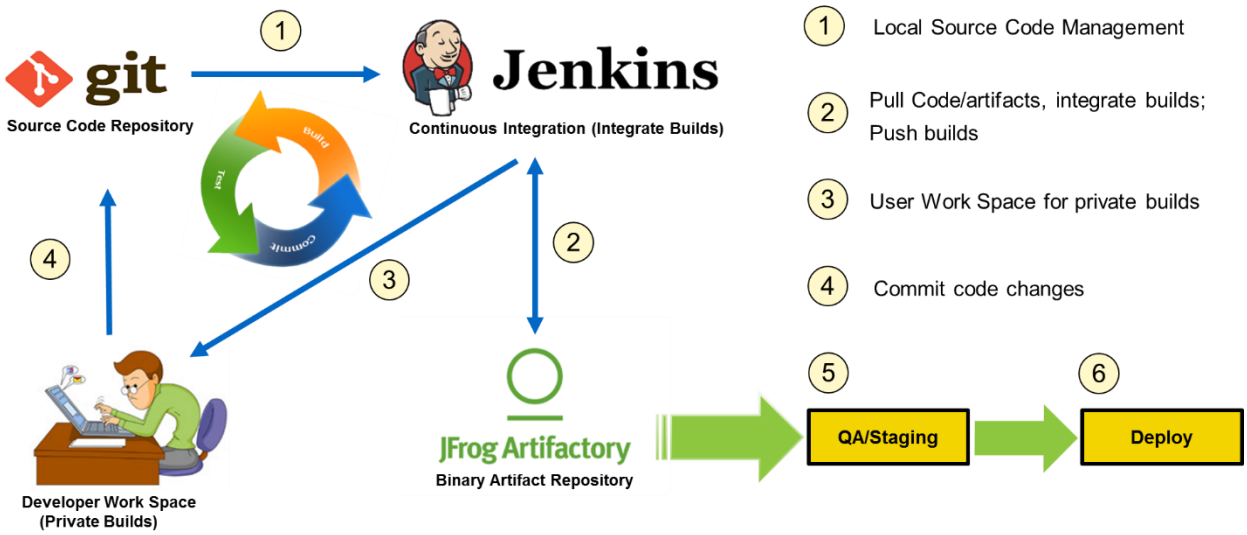


Figure 2 shows a typical CI/CD workflow from the source code management (SCM) until the application is packaged, deployed, and released to production. JFrog Artifactory sits in the center of all the development and deployment activities for continuous delivery of applications into production. The builds performed by Jenkins pull all the required dependencies from JFrog Artifactory, and the final builds are pushed to the respective repositories into Artifactory, which are eventually promoted for staging and release.

Although metadata and binaries are managed by JFrog Artifactory, the actual data generated from the artifacts and builds needs to be managed, protected, scaled, replicated, and collaborated between different development teams locally and globally. NetApp provides a stable, standard data management platform for application development. As developers increasingly move workloads to the cloud and collaborate between remote sites, data management matters more than ever.

ONTAP® 9 and StorageGRID Webscale provide better performance, scalability, replication, and collaboration for on-premises cloud environments with low costs. In contrast, public cloud vendors such as Amazon Web Services (AWS) allow more control over data and no vendor lock-in like in hybrid and multicloud environments. StorageGRID Webscale not only allows better performance for instant replication of the files globally but also provides inexpensive and deep storage to archive the high number of builds as they scale over time.

3 Business Challenges

Many midsize to large organizations have data centers scattered around the globe. Developers with specific skillsets working on certain designs, applications, and services are spread out in different remote

locations. Development using disparate and nonstandard tools and dependencies does not provide a productive environment for the developers. Consolidating the build artifacts in a central location and sharing them in a distributed manner provide a more consistent runtime environment for the developers. There are, however, some business challenges around sharing and distributing the artifacts and builds across remote sites:

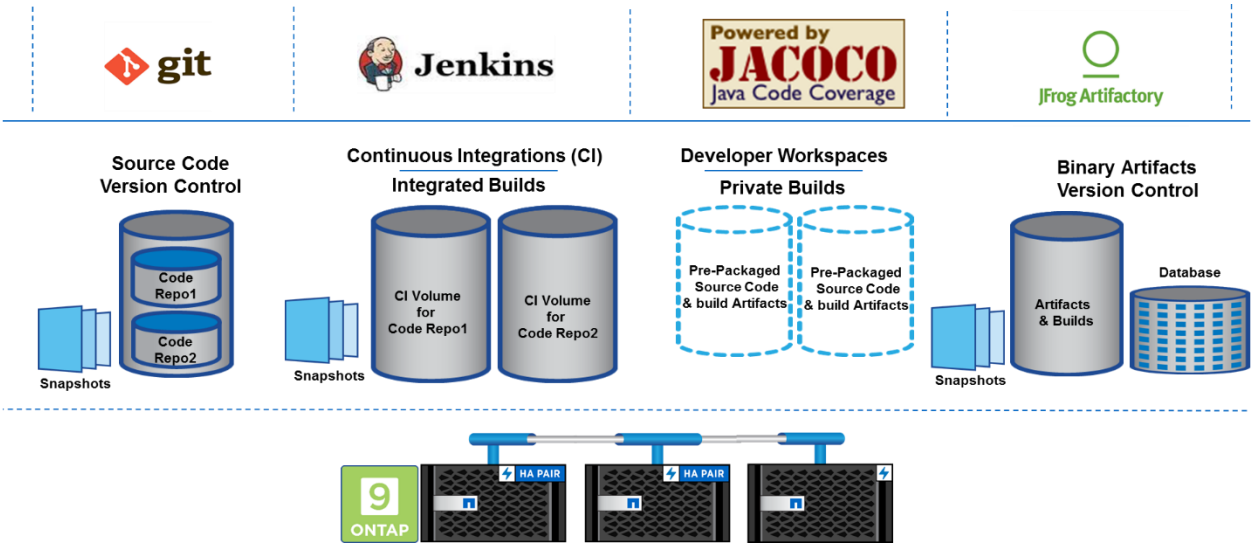
- How can you instantly sync the data changes with different data formats that happen at one site with other sites that could be located globally?
- What are the performance and communication costs to replicate the files/artifacts over WAN?
- After the files are synchronized with the all the remote sites, how can respective teams collaborate and make changes to files at the same time?
- How can the users have read/write access to the same file at the same time?
- How do you archive as the number of builds starts to increase in number and size over time and still control cost and meet performance SLOs?
- How can you achieve modular growth when you add a new site to your existing infrastructure?

4 JFrog and NetApp Integration

4.1 JFrog Artifactory in the Continuous Integration Pipeline with ONTAP 9

Integrations with JFrog Artifactory, NetApp ONTAP 9, and StorageGRID Webscale enhance performance and data management capabilities. Figure 3 illustrates the continuous integration pipeline where JFrog Artifactory functions as the binary artifact manager with private repositories for all the artifacts and builds.

Figure 7) Continuous integration pipeline.



The entire CI pipeline is set up and configured on ONTAP 9 using RESTful APIs. ONTAP 9 provides persistent data storage and data management services such as application-consistent data backups, along with disaster recovery (DR) and performance levels for all stages of the pipeline. For this technical report, the CI process consists of version control for source code using GIT, integrating builds with Jenkins, and developer workspaces for private builds. The incremental, full, or nightly builds generated after successful CI and testing are pushed to JFrog Artifactory using the Jenkins Artifactory plug-in.

The builds such as the .bin (binary for C) and .war (web archive for Java) go into the respective development (dev) repositories in Artifactory, which are later promoted to production (prod). All the persistent data storage uses Network File System (NFS) v3 to mount physical nodes, virtual machines (VMs), or Docker containers for scalability, resilience, manageability, and ease of use.

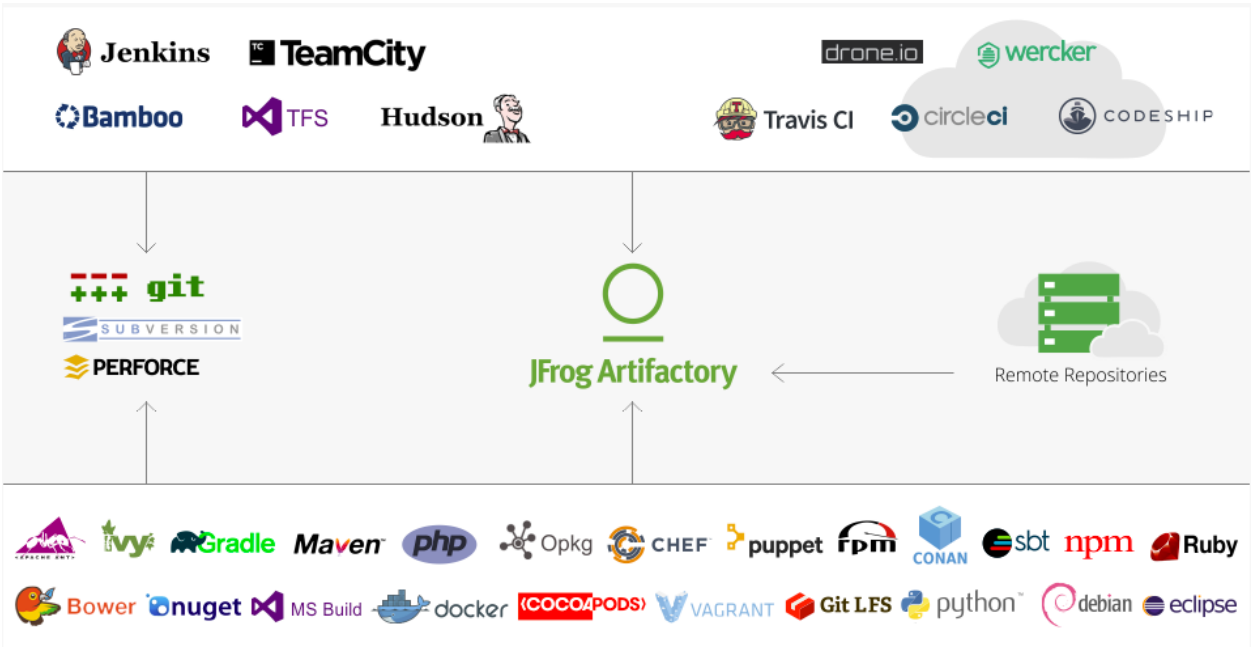
4.2 JFrog Artifactory

JFrog Artifactory is a universal artifact repository manager that fully supports software packages created with any language or technology. It provides a central hub for distributed repository management for all artifacts and builds. It works as an intermediate layer between the different development teams across an organization and external repositories.

JFrog Artifactory consists of a database that stores all the metadata (versions, dependencies, license, build date and time, documentation, approval information, and metrics for code coverage) and rules compliance.

The repository contains different packages that contain these files which are identified with a unique checksum. There are mainly three types of repositories: local, remote, and virtual. The local and remote are physical repositories. Local repositories can be deleted, created, modified, and aggregated. A remote repository works as a proxy and is not a mirror of the local repository. Artifacts are not prefetched; instead, they are fetched and stored in the remote repository cache on demand. JFrog Artifactory provides various plug-ins to connect with the different tools used in the build ecosystem. It also allows access rules to be set for users and groups that consume artifacts. Figure 4 shows the different repositories that are supported by JFrog Artifactory in the build ecosystem.

Figure 10) JFrog Artifactory repositories in the build ecosystem.



4.3 NetApp StorageGRID Webscale

StorageGRID Webscale is a hyperscale global object store and a software-defined platform for content repositories for mission-critical applications. StorageGRID Webscale nodes can run on engineered hardware appliances or can be deployed on software-only nodes using virtual machines running in VMware or KVM on OpenStack. With virtual nodes, you can deploy NetApp or third-party disk arrays

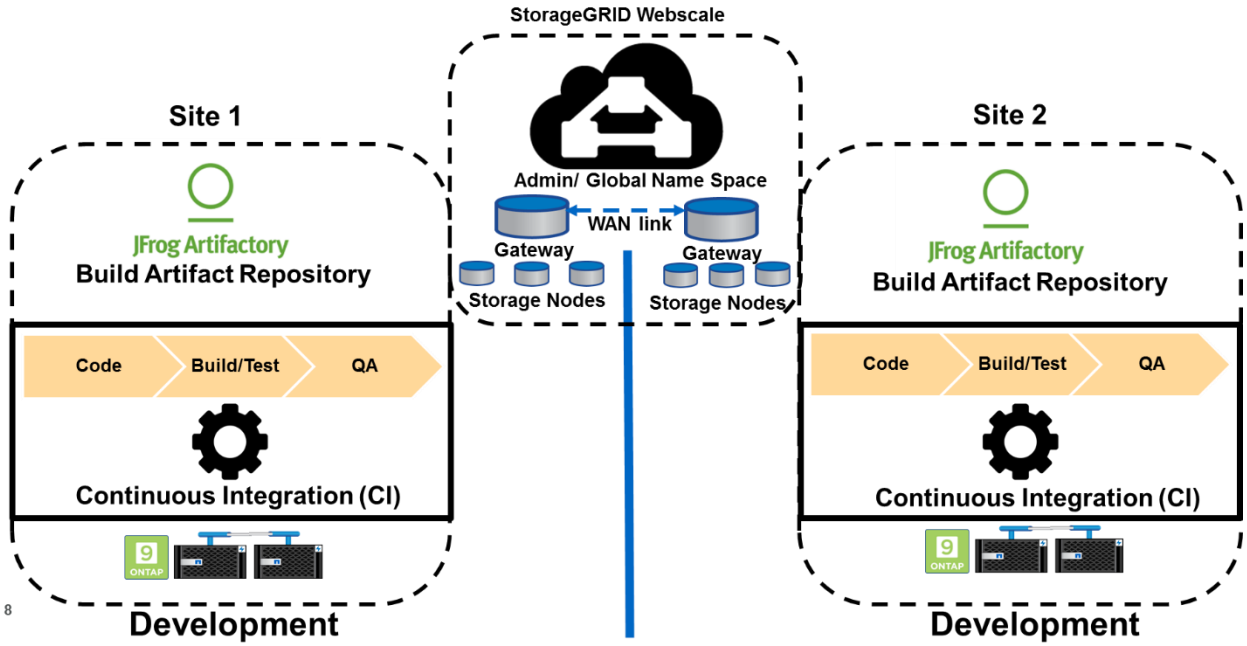
underneath. StorageGRID can also use tape and S3-compatible public cloud as an integrated tier. StorageGRID provides dynamic data lifecycle management policy. Admins can create rules that manage data lifecycle at the object level and customize data locality, durability, performance, efficiency, and retention time.

StorageGRID consists of a single grid with a set of gateway, storage, and archive nodes at different sites in different geographic locations: for example, an organization that has three major data centers across the world in the United States, Europe, and Asia. Each data center has a gateway, storage, and archive nodes depending on the capacity at each location:

- **Admin node.** Provides management services such as system configuration, monitoring, and logging. There is going to be one admin node for the entire grid.
- **Gateway node**. Provides a load-balancing interface to the StorageGRID Webscale system through which applications can connect to the system (optional).
- **Storage node**. Manages object data and metadata storage, including loss protection.
- **Archive node**. Provides an interface through which object data can be archived to an external archival storage system (tape or the cloud) for long-term storage.

Figure 5 shows a multisite StorageGRID environment that has gateway and storage nodes at each site with an admin node that overlays the entire grid.

Figure 13) Multisite setup with StorageGRID Webscale.



4.4 JFrog Artifactory and StorageGRID Webscale

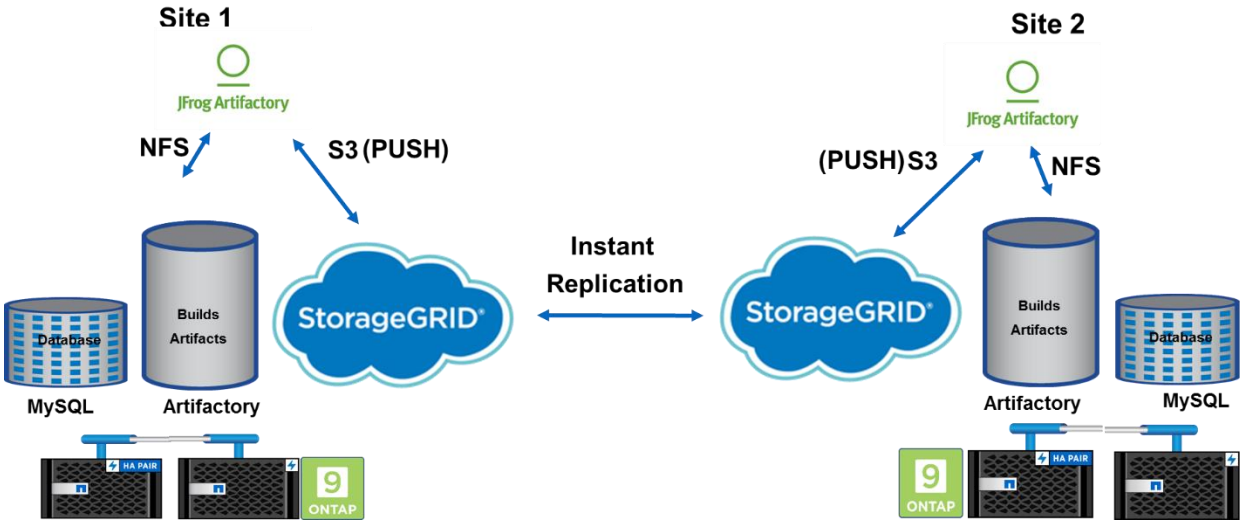
The StorageGRID Webscale integration with JFrog Artifactory was designed to address the business challenges listed in section 3, mainly:

- The requirement to be able to sync data artifacts between multiple distributed locations instantly
- The ability to provide a distributed solution that meets SLOs for performance, access, and security
- The ability to provide an affordable solution that can scale

As mentioned in section 4.1, the final build artifact is pushed into the JFrog Artifactory repository. The Artifactory metadata (database) gets updated as soon as the build artifact completes writing to the repository into the persistent data storage in ONTAP 9.

The developer can take an application-consistent checkpoint (Snapshot® copy) of the database and the repository from the JFrog user interface (UI), without the involvement of an infrastructure developer or DevOps engineer, whenever that developer pushes a new build in the Artifactory repository. The checkpoint is useful for data backup and recovering from any data corruption or disaster. This paper does not go into the details about how checkpoints are used for protecting data. Contact NetApp or JFrog if you are interested in understanding more.

Figure 16) Replicate and collaborate build artifacts across multiple sites.



The scope of this paper is focused on the integration of StorageGRID with JFrog Artifactory for multisite development teams. The specific use case focuses on what happens after an app-consistent Snapshot copy is taken at the local site.

JFrog Artifactory has the ability to read the files or checksums from the respective repositories in ONTAP over NFSv3 and push them over S3 into StorageGRID. As soon as the files are pushed to StorageGRID, the process of instant replication is initiated. The JFrog Artifactory database immediately updates the metadata change at every site. A remote user can instantly locate artifact changes and updates to the repository. Files are visible at all locations. At this point WAN latency between remote sites is the only factor that influences the download speed of files.

Developers at remote sites can collaborate on the same file or build and push it to a local JFrog Artifactory, where it is saved as a new checksum in the repository. The built-in “diff tools” can compare and identify the changes that were introduced from one version to the other. This allows for small changes to be updated and replicated almost instantly.

A secondary failsafe process is to use the shared and distributed source code management in GIT at every location. Every developer (local or remote) could perform a “git rebase” to sync the changes from the master code branch into their local workspaces, merge the code, and perform integrate builds at every site.

4.5 JFrog Artifactory and StorageGRID Webscale Validations

Test Environment

JFrog Artifactory v5.1 Enterprise Edition was used for this validation. Artifactory uses Derby as the default database, but customers can choose to use third-party databases such as Oracle, MySQL, DB2, and so on for better manageability and usability. MySQL was used as the database for this validation. Both the database and the repository were configured on ONTAP 9 over NFSv3 and mounted from a Red Hat Enterprise Linux (RHEL) 7.3 VM. The local and remote repositories were configured at both the simulated sites.

StorageGRID Webscale v10.3 was used for replication and collaboration validation. There were two simulated sites with a WAN latency of 170ms. As mentioned in section 4.3, one admin node had the access for the entire grid. One gateway and three storage nodes were used for each site. All of the StorageGRID nodes were configured on multiple VMs. CloudBerry browser was used to browse the files/checksums in the local and remote repositories.

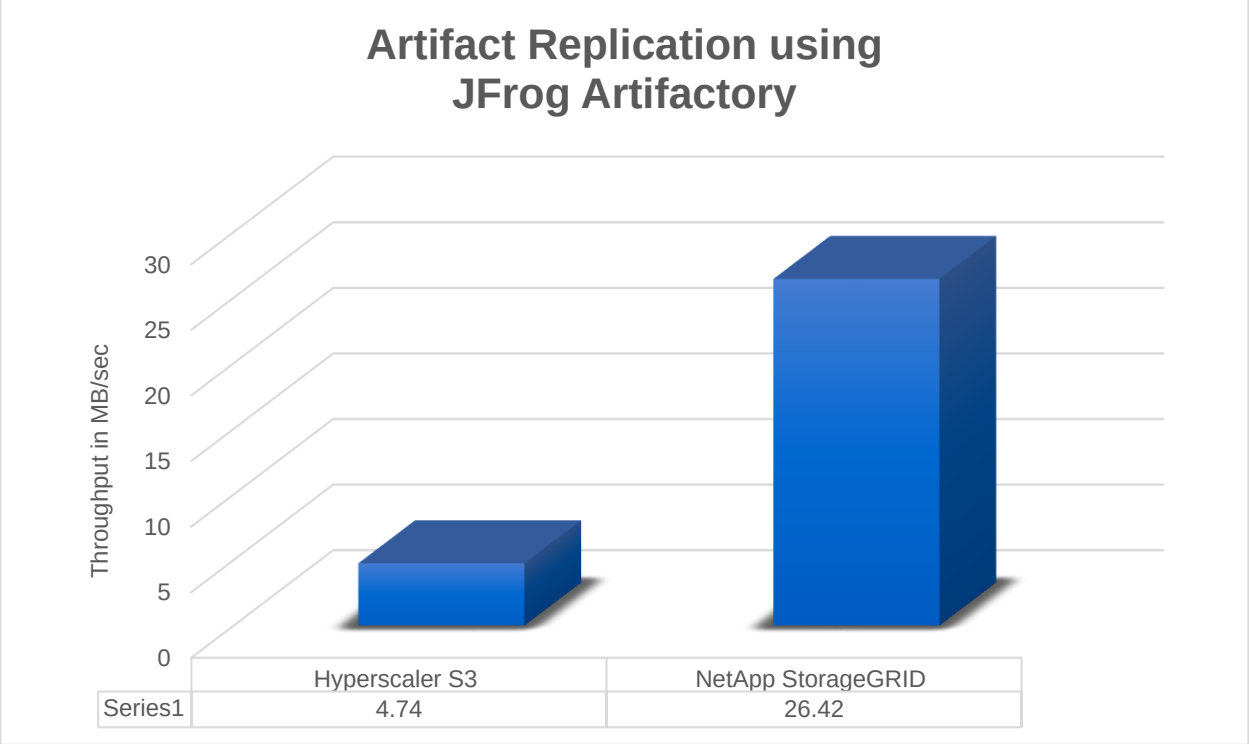
JFrog used a script to push 100 artifacts that were 95MB in size. This test was done in AWS S3 and NetApp StorageGRID to compare the download performance across sites/regions and the cost of ownership. The test process also deleted checksums in one site and was verified if the delete operation was synced with the other site. The test environment was very similar to the layout represented in Figure 5.

Results

The graph in Figure 7 shows that cross-site replication with StorageGRID was 6x faster than that provided by a well-known hyperscaler cross-region replication (CRR) alternative. The test measured the download speed of the file at the remote location while it was replicating. StorageGRID provided a global file system for the entire grid, with instant metadata replication and asynchronous file copy taking place between sites. Metadata is very small in terms of amount of storage required and hence gives the perception of instantaneous replication. This is the reason why StorageGRID is able to create a single global namespace. The metadata about any new objects is instantly available across WAN links. When the underlying data referenced by the metadata is accessed, this data needs to be fetched from its original location. If the original location is remote, then the throughput for the object depends on the WAN throughput. If a copy has been created locally, then local LAN throughput, subject to StorageGRID performance, is seen. When the object is remote, StorageGRID replication throughput for the data is directly dependent on WAN speed. An important factor to note is that StorageGRID replication throughput increases as the WAN throughput increases.

Based on comparable capacity and throughput requirements, StorageGRID not only provided faster download speeds, but also compared favorably based on cost, coming in at less than a third of the cost of a comparable cloud-based S3 alternative. The resulting return on investment (ROI) is better and can help customers achieve cost objectives more quickly using this hybrid cloud setup.

Figure 19) Cross-site replication performance results.



Observation

The JFrog Artifactory and StorageGRID Webscale validation integration provides an on-premises private cloud solution for storing and managing data that is secure. This is important for customers who consider their applications to be of strategic value or where regulatory and compliance requirements dictate that code binaries cannot leave an organization's data centers.

As the number of builds keeps growing at every location, data can be archived locally in an inexpensive and deep object store such as StorageGRID. That data can then be replicated and synchronized with all the remote locations, all without leaving the control of the development organization.

StorageGRID can protect files against any accidental deletions from JFrog Artifactory. Even though the file might not be listed in JFrog repository after the deletion, it can be recovered from StorageGRID.

Summary

JFrog artifactory allows remote users / remote sites to access a global set of artifacts, which are all stored in a central location. Artifactory is caching metadata which works/acts as a proxy for remote users, thereby allowing them to access artifacts through pointers from a central location.

Array-level data replication using NetApp products such as ONTAP and StorageGRID is not entirely a smart way either because Artifactory at the remote sites does not sync with the data changes and updates at the application layer.

JFrog Artifactory integration with NetApp ONTAP and StorageGRID complements the distributed sharing solution by providing application-aware data replication across all the remote sites. The replication is instantaneous and read/write. NetApp ONTAP and StorageGRID along with JFrog Artifactory Enterprise Edition provide the following benefits to sharing data in a distributed setup:

- There is data continuity when data is accessed in different data formats such as file shares over NFS and objects over S3 across remote sites globally. This continuity is transparent to the end users without any additional overhead.
- They also provide data access and control that automate the data flow between different platforms when data is replicated across remote sites.

Conclusion

JFrog Artifactory supports multiple protocols such as NFS and S3 to provide homogeneity of data across different platforms and cloud environments that is transparent to the developers. Artifactory, apart from being a universal binary repository manager, also can instantly sync metadata updates to all remote sites to which it is connected. All updated files can be listed immediately at the remote locations.

NetApp is a leading data management platform for application development because of unique capabilities that make the process faster, easier, and more efficient. We also provide a stable and standard storage platform that is necessary for application development.

Data should not be locked into a specific format on heterogeneous storage silos. Data should scale and move beyond the walls of local data centers into other locations to collaborate and provide local access to users.

StorageGRID Webscale provides 6x performance benefit and low cost for replicating and archiving data across regions compared to a comparable cloud-based S3 alternative. This is a huge boost for ROI as builds and artifacts start to scale. After the application is retired, all the artifacts can be archived for regulatory purposes.

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

Trademark Information