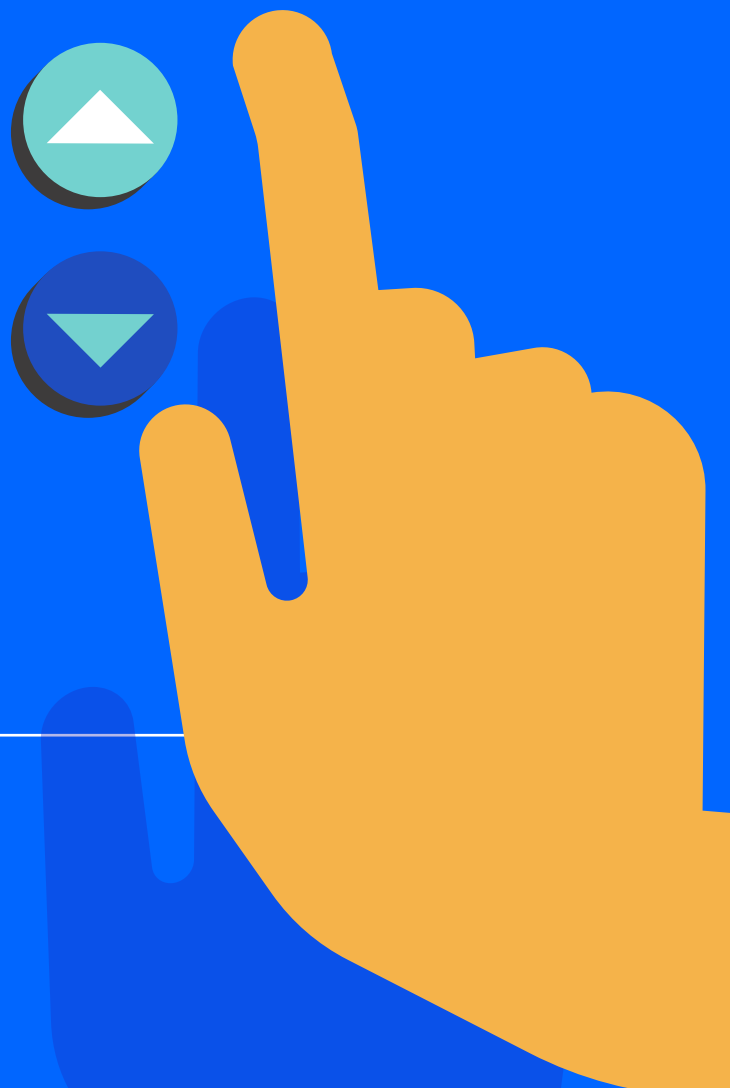
A Nuxeo Whitepaper

Architecting Applications to Scale in the Cloud



NUXEO

Architecting Applications to Scale in the Cloud

Table of Contents

Executive Summary 1
- Between IaaS and SaaS 1
- Nuxeo and AWS 1

AWS for Content Management 2

Architecting Applications for the Cloud 3

Introduction to the Nuxeo Platform 5

Tailoring Apps to Individual IaaS Capabilities 6

Scale Out and Distributed Architecture 9
- CPU Scale Out 9
- Dedicated Resources and Specific Processing Requirements 10
- Storage Scale Out 12
- Query Scale Out 12
- Scaling Out with Multiple Data Stores 13
- Scaling Out with NoSQL 14

About Nuxeo 16

Executive Summary

Amazon Web Services (AWS) and similar cloud-service offerings have revolutionized the ways in which organizations approach application development and deployment. Through cloud services, organizations can obtain on-demand, fast and modular infrastructures for deploying virtual instances of networking components, storage repositories, compute platforms, and management frameworks. As cloud services continue to mature, the Platform-as-a-Service (PaaS) model has been steadily gaining recognition and favor.

The abstraction layers provided when using a PaaS model offer a particularly effective means for developers to focus on coding and to implement a variety of business applications efficiently on public clouds.

These abstractions also relieve administrators of many of the configuration and deployment challenges, as well as maintaining traditional hardware servers, and easily configuring the PaaS environment through administrative tools and control panels to address changing requirements.

Between IaaS and SaaS

In a typical implementation, PaaS serves as a partial operating system and middleware, residing in the technology stack between an underlying Infrastructure-as-a-Service (IaaS) layer and the Software-as-a- Service (SaaS) layer that provides a user interface. As discussed throughout this paper, **the combination of AWS and PaaS opens up many advantages and provides numerous mechanisms for developers and architects to design, plan, and implement new technologies and prototype new solutions without the cost and commitment of traditional computing.** Software developers following agile development processes have the flexibility to move from ideas and concepts through rapid development[1] to a marketable product— swiftly and cost effectively—within a responsive PaaS environment.

Nuxeo and AWS

Using a PaaS model optimized for the capabilities and features of AWS, Nuxeo has designed an extensible modular framework for handling high-volume document management and complex digital asset management tasks—with high scalability and reliability. This design framework makes it possible to **efficiently leverage AWS cloud services and customize the platform for a diverse variety of requirements**. This paper provides an architectural overview of the fundamental mechanisms for building and deploying content management applications on AWS using the Nuxeo Platform.

AWS for Content Management

Since the introduction of Amazon Simple Storage Services (S3) in 2006, AWS has evolved into a powerful, responsive infrastructure that has helped shape the way in which applications are developed and how enterprises handle IT requirements—providing commodity-level access to a full range of cloud services. Several AWS characteristics make it well suited for content management tasks, including:

- **Fast, low-latency content distribution:** Delivery of content over the web to end users using Amazon CloudFront is a fast, cost-effective way for handling everything from streaming video to entire dynamic websites. Through integration with other AWS offerings, such as S3 and EC2, content delivery can be optimized to take advantage of Amazon's global network of edge locations, minimizing latency and boosting performance. As a usage-based service with no commitment, the only costs accrued derive from the actual volume of content delivered. Content collaboration can be handled efficiently with full access for workgroups spread out across multiple locations. Even extremely large files, such as those encountered in engineering projects and digital asset management environments, can be delivered securely worldwide—at high speed—to support collaborative efforts.

- **Inexpensive, reliable storage:** S3, Amazon's pioneering cloud storage service, continues to provide value to enterprises that require a reliable, scalable method for storing many kinds of digital assets in the cloud. Developers gain secure access to object containers, referred to as buckets, that are addressable by URL, located in a specified geographical region, and scalable to accommodate escalating storage demands.

- **Elastic compute resources:** Content-centric applications often place varying demands on compute resources for media processing, handling different levels of user traffic, data migration, sorting and indexing, and other tasks. AWS provides a number of elastic capabilities that can scale to meet demands, including the capability to scale out virtual servers (EC2), media transcoding (Elastic Transcoder), perform Auto Scaling, and adjust load balancing (ELB).

- **Proven work environment:** AWS has refined and enhanced its cloud service offerings over many years to ensure a secure and reliable work environment for the mission-critical applications encountered in content management operations. Important features are available and accessible, including database snapshots, automated load balancing, key performance metrics, high-volume bandwidth availability, monitoring, and so on.

Other AWS capabilities support content-centric applications very effectively. For example, disaster recovery can be accomplished rapidly using failover techniques, minimizing downtime and potential enterprise losses from business interruptions. In digital asset management scenarios, back-end processing nodes can be separated from front-end servers so that the scaling of images and video processing can be handled more efficiently.

AWS provides a capable IaaS framework for new-generation digital asset management projects, distributed content collaboration, and global content distribution.

Architecting Applications for the Cloud

Simply porting traditional applications over to a cloud platform is usually not the best option: not leveraging the PaaS offering, developers would have to build and manage their own services.To take full ad-vantage of a cloud-services environment, whether IaaS, PaaS, or SaaS, the key to success entails understanding the limitations of the environment and then effectively leveraging the built-in advantages. **Designing an application from the ground up—the best way to build apps for the cloud—requires that developers adopt a different mindset and master a new paradigm.**

For example, architecting an application that can rapidly adapt to high volumes of transactions and a varying number of users typically requires that developers create a large amount of code to handle the demands of scaling, which might include caching, database scaling, asynchronous messaging, and so on. A well-designed PaaS platform will already be optimized with built-in capabilities for these functions. Instead of writing code, the developer can simply tap into the appropriate functions as needed.

General guidelines for architecting applications for availability in the cloud include:

- **Anticipate failure:** Be aware of the possibility that parts of the cloud will sometimes fail. Design and test applications for resiliency and the ability to respond to failures. Componentize applications so that multiple modules that communicate with each other through an API can recover independently, if necessary. Data replication and multiple deployments of critical components are useful in this regard.

Useful references on this topic include:

- Open Data Center Alliance: Architecting Cloud-Aware Applications
- Architecting the Cloud: Design Decisions for Cloud Computing Service Modes (SaaS, PaaS, and IaaS)
- Architecting Applications for the Cloud: Best Practices

- **Employ stateless computing techniques:** Using stateless protocols eliminates the possibility that a state stored in memory will be lost due to an outage or service interrup-tion. Because the internal state of an application won't be available as conditions change or failures take place, store these states in an object store, database, or message queue.

- **Scale up and out:** When it comes to pure processing, scaling out horizontally is much more effective, offering essentially unlimited scalability that takes advantage of elasticity of cloud resources. However, for the persistence layer (i.e. databases), there are a lot of cases where scaling up can make sense and would actually be the right solution. If you're using a SQL PaaS like RDS, scale up is the option you should look at. If you're using NoSQL and natively distributed storage, you gain the possibility to scale out by adding nodes, but it may involves data migration (even if automated).

- **Keep data consistency in mind:** Because there can be multiple instances of an application residing in different geographic regions, some changes in the database or the application may not be reflect-ed for a number of milliseconds. To maintain a model where data is replicated and highly available within this environment, developers must devise an approach that handles potential inconsistencies when different application instances draw from the same database.

Investigate the specifications of the PaaS that you select to fully understand the capabilities and feature set as you begin development. In the case of the Nuxeo Platform, even if your application requirements are modest, such as basic document management or digital asset management that includes workflow processes, built-in cloud-aware capabilities that have been architected into the product can save considerable time and effort, as well as improving the reliability of your application. Nuxeo also frequently updates the platform to include new features and capabilities, so that as cloud service technologies evolve, architects and developers can take advantage of the latest enhancements.

Introducing the Nuxeo Platform

The Nuxeo Platform, when coupled with AWS and the available toolset, provides a comprehensive, extensible platform, readily adaptable to business application development. Through the efficiencies of operating tightly with AWS, the Nuxeo Platform enables architects and developers to easily build and run content-focused business applications that can handle extremely large document sets (even at volumes ranging into the billions).

Several pre-built applications are included, allowing development teams to quickly deploy and launch a number of fully featured content-management tools or customize these applications to meet specific requirements. These modern technologies, powerful plug-in model, integrated development environment, and flexible packaging capabilities make the Nuxeo Platform an ideal environment to rapidly design, develop, and deploy applications, on premises or within a cloud environment.

Nuxeo Platform supports the creation of end-to-end workflows—through a graphical interface— for performing content-management processes. Alternatively, applications can be built within the integrated development environment, accessing the exposed functionality in an API that supports the representational state transfer (REST) model.

Tailoring Apps to Individual PaaS Capabilities

Capabilities and functionality of individual IaaS offerings vary. For maximum efficiency, interoperability, and performance, when building applications to run on top of an IaaS framework, developers gain many advantages by exploiting the built-in features, components, and capabilities available.

The infrastructure offered by the cloud provider typically:

- Includes components that are fully integrated and tested for interoperability
- Features mechanisms for manual or automated scaling of virtual servers, storage, network resources, and other system resources
- Provides an easy way to monitor ongoing costs by usage, with billing limited to those computeresources that are used
- Costs substantially less to use than comparable on-premises systems

As a developer, when architecting a solution to deploy in an IaaS environment, you should:

- Rely as much as possible on open standards, as well as industry standards that are widely accepted.
- Build solutions based on a pluggable architecture.

The Nuxeo Platform supports both a standards-based architecture and pluggable component model (Extension Point). Nuxeo is well suited for leveraging the AWS infrastructure, featuring:

- Meta-Data Store: Oracle RDS or PostgreSQL
- Binary Store: S3 Binary Store
- ElasticCache / Redis
- AWS Elasticsearch
- AWS MSK (for Kafka)
- AWS Lambda
- Conversions
- AI services

These same basic concepts apply to a number of cloud-specific services, including provisioning and monitoring.

As much as possible, the solution should fit in the model as defined by the IaaS provider and exploit those capabilities that make it easy to perform operations without the need for extensive coding. Ideally, this includes the capability to configure automated processes that would otherwise need to be manually managed or configured.

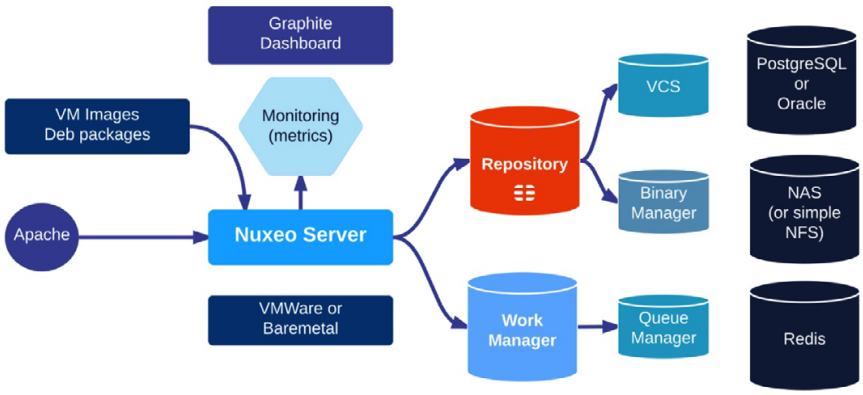


Figure 1: Nuxeo Platform standard installation in an on-premises environment

By default, Nuxeo is packaged as Debian packages and also has available installers for a number of other environments.

Nuxeo Platform can also be set up using Amazon Machine Instance.

For monitoring and managing resource use, Nuxeo exposes its metrics by means of Java Management Extensions (JMX).

These metrics can be accessed by AWS CloudWatch and used to engage AutoScaling to respond rapidly to changing application demands.

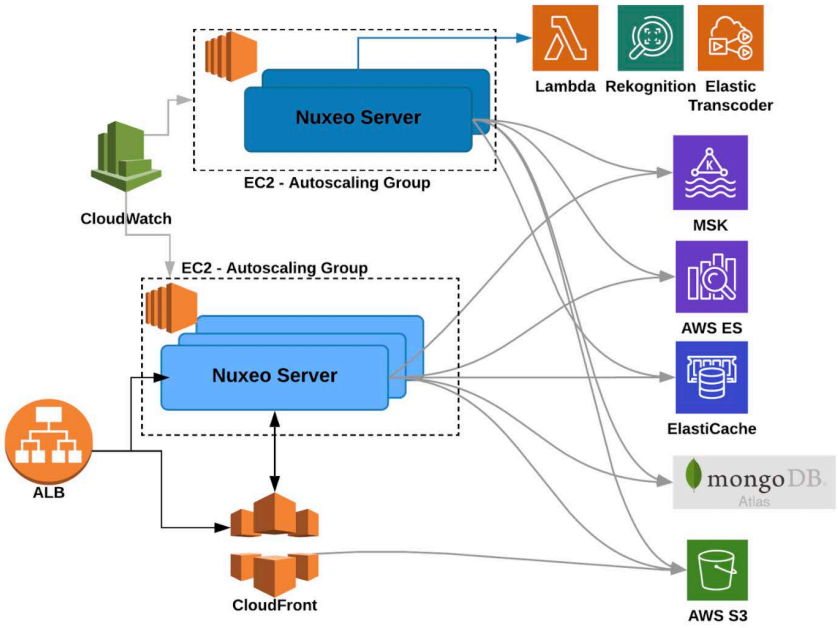


Figure 2. Nuxeo Platform on AWS

Scale Out and Distributed Architecture

Scaling fluidly to meet the demands of business is a key advantage of cloud services. The AWS IaaS includes a number of features that make it possible to react as fast as possible and adapt to the demand, removing the need for administrative monitoring and manual actions to address resource issues.

The underlying architecture of cloud services makes it easy to quickly provision new servers (scaling out), but much more difficult to quickly provision the basic resources of a virtual server. However, you can increase available processing power using clustering of virtual servers to scale out, effectively achieving the same processing gains as would be achieved by scaling up. Alternatively, to meet specialized application requirements, you can select virtual machines (VMs) that are pre-provisioned with certain characteristics, such as VMs optimized for handling heavy I/O or providing substantial amounts of memory.

The bottom line is: your application will be able to scale in the cloud if it supports scale out. If you are limited to scale up, you won't be able to gain much benefit from cloud services.

Nuxeo Platform architecture can scale out along different axes, adapting to whatever type of demands are to be absorbed, as described in the following sections.

CPU Scale Out

Nuxeo processing demands can easily be scaled out using the built-in clustering model. In support, AWS includes specific features to accommodate high-performance computing in the cloud, using Cluster Compute. This lets you scale out applications across thousands of cores to more effectively handle massive throughput demands with tightly-coupled I/O across a high-bandwidth network.

Using several mid-size VMs, you can build a high-performance Nuxeo Platform application and then use AutoScaling to automatically add one or more VMs when the load increases.

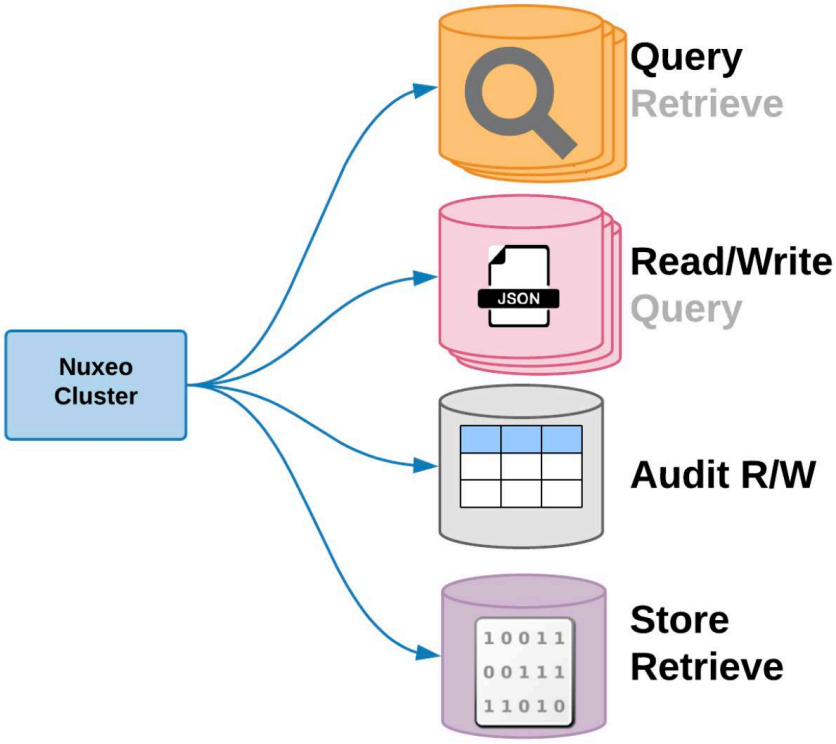


Figure 3. Nuxeo Clustering with Scale Out and AutoScaling

Dedicated Resources and Specific Processing Requirements

Certain types of processing operations require specific resources and hardware. AWS offers several types of VM configurations to meet a range of requirements, including:

- I/O provisioned to handle I/O-intensive operations
- Graphic processing unit (GPU) or High CPU to take care of processor-intensive tasks, such as videotranscoding or artificial intelligence
- High-memory VM to accommodate applications that require large blocks of contiguous memory space to operate efficiently

The Nuxeo Platform architecture provides the flexibility to dedicate nodes for specific types of processing operations so that you can:

- Use a general-purpose VM to accomplish basic tasks
  - ¤ Ensure good response time for the interactive users
  - ¤ Ensure cost-effective processing
- Leverage specific VMs for particular demanding tasks

Dedicated processing nodes can be useful for:

- Performing video transcoding
- Processing high-resolution digital images
- Scanning large files for viruses
- Performing optical character recognition on scanned images
- Running cryptographic algorithms to encrypt and decrypt files
- Indexing large collections of documents

Nuxeo recommends two approaches:

1. Deploy exactly the same Nuxeo image on each node. The one exception would be for hardware and the work-queue configurations. This ensures that if the dedicated nodes become unavailable, the standard nodes can continue with the processing.
2. Isolate the different types of processing inside the application: it will be easier to monitor and scale out the part that is needed and will avoid the "noisy neighbors" issues.

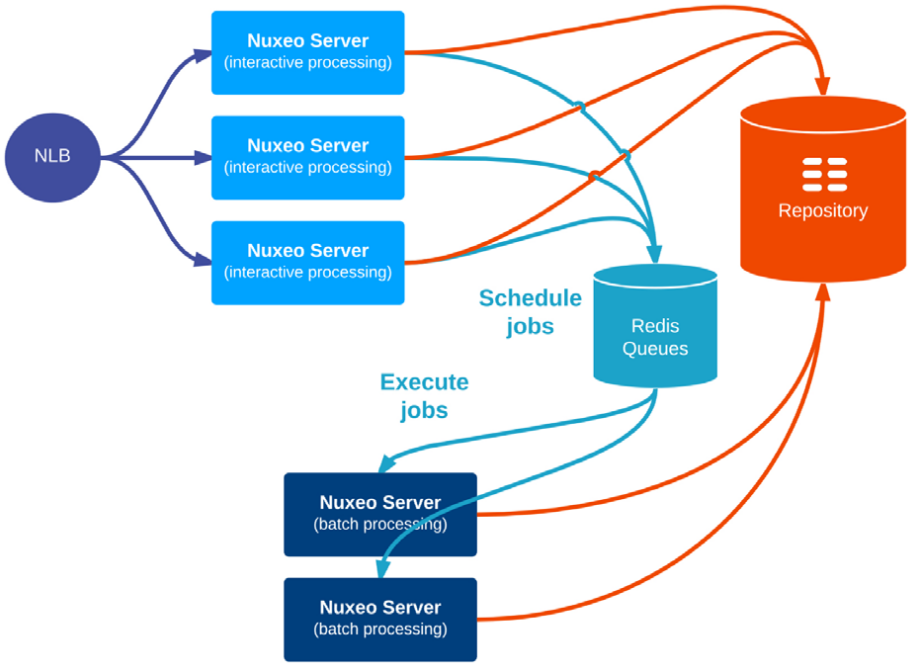


Figure 4. Nuxeo cluster with dedicated nodes and redis WorkManager

Storage Scale Out

Typically, solutions rely on scaling up the data tier. Even if this is somehow possible—with solutions like AWS RDS—it is usually not the optimal approach.

- Scaling up is not cost effective.
- Scaling up cannot be progressive and transparent.
- Scaling up cannot be continued indefinitely.

To address this, the Nuxeo Platform includes a number of options that support scaling out, to improve processing at the data level.

Query Scale Out

When using the default storage back end, Nuxeo makes extensive use of the database. Queries in particular create a great deal of database activity with a potential for bottlenecks. In certain configurations, the Nuxeo Visible Content Store (VCS) generates complex SQL queries that keep the database server very busy.

When the volume of data queries increases, the number of concurrent accesses increases as well. Queries typically present the primary bottleneck that diminishes the database server performance, slowing query response times.

At the SQL level, you essentially have two options for reducing the bottleneck:

- Use a database server with greater capacity.
- Denormalize the data to make the query run faster.

The use of Elasticsearch as the primary query engine lets you transparently direct the query to scale on multiple nodes and maintain high-end performance—even on massive volumes—using mid-range hardware. By nature, Elasticsearch does not try to enforce the same kind of integrity that ACID properties do. Because of that, it can be easily distributed across several nodes, providing a very simple and efficient scale-out solution for queries.

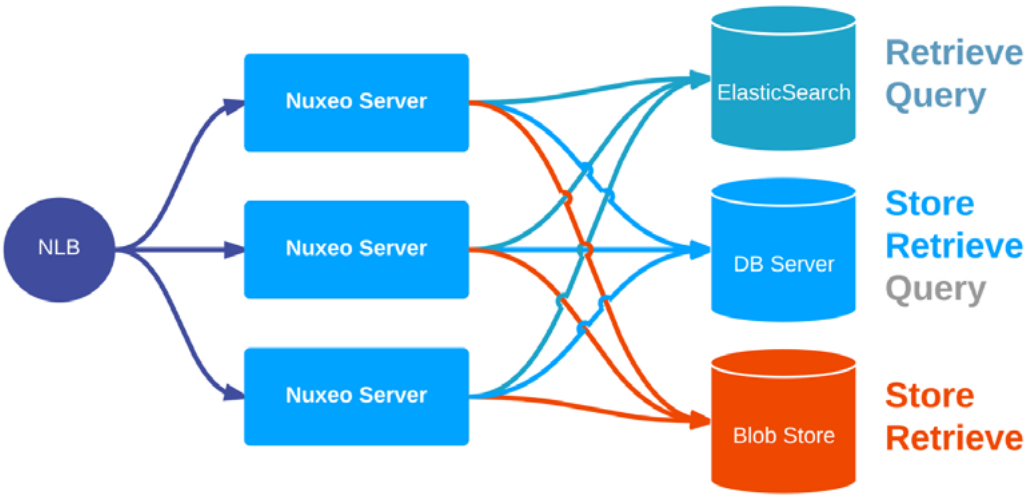


Figure 5. Nuxeo and Elasticsearch architecture

As concurrent users increase, the requests handled per second scales very effectively. As you can see on benchmarks.nuxeo.com, platform testing showed an impressive degree of scaling to handle concurrent requests even at volume levels of one billion documents.

Scaling Out with Multiple Data Stores

In terms of storage, if a database server reaches it's upper limits for store-and-retrieve operations, the Nuxeo Platform supports data sharding across several repositories. With this capability, you can exceed the scale-up limitations of one database server, because the application can distribute data across several repositories, each of them associated with a single database. This effectively boosts both database performance and scalability.

Elasticsearch makes it possible to maintain a unique index for perform-ing federated search operations across multiple repositories. From a document and asset management perspective, this mechanism can be extremely useful, providing a single interface to a diverse range of information resources and returning query results in a single list. It also makes it possible to link directly to each resource to further expand the search.

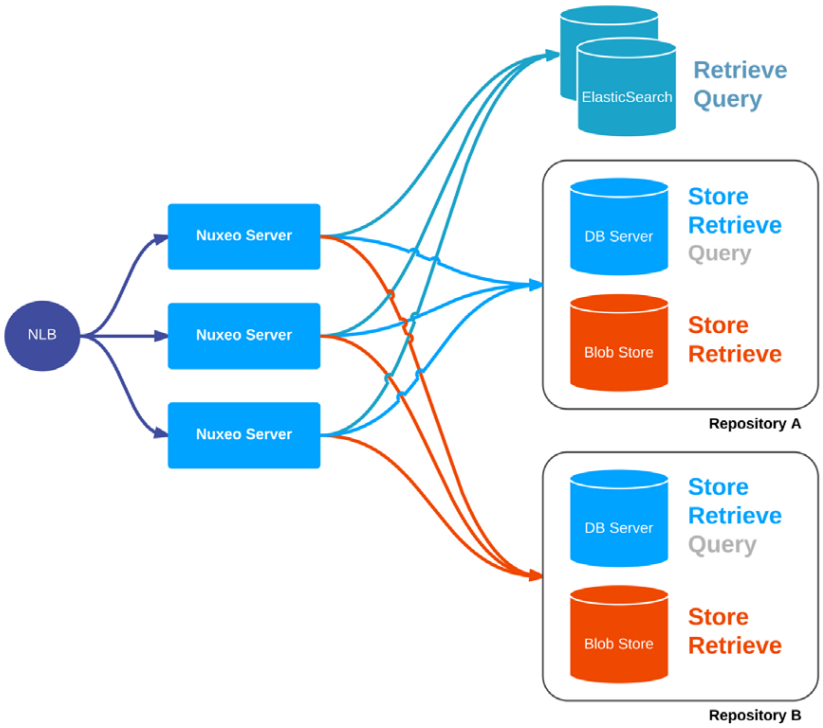


Figure 6. Nuxeo and Elasticsearch and MultiRepo

Scaling Out with NoSQL

If your database requirements approach the level of billions of documents, traditional SQL databases may not be the right choice, even with the help of Elasticsearch. **Nuxeo Platform support for MongoDB, a NoSQL database, extends high-volume database capabilities with powerful scaling.** The NoSQL database movement originated in response to the well-recognized limitations of traditional relational da-tabases and the difficulties in storing and analyzing massive volumes of data encountered in many of today's web applications.

MongoDB offers these benefits in a content management environment:

- Relies on a native distributed architecture
- Scales out very easily

With the pluggable Nuxeo architecture, switching from a SQL backend (VCS) to a MongoDB backend (DBS) becomes a basic configuration task that can be accomplished during deployment.

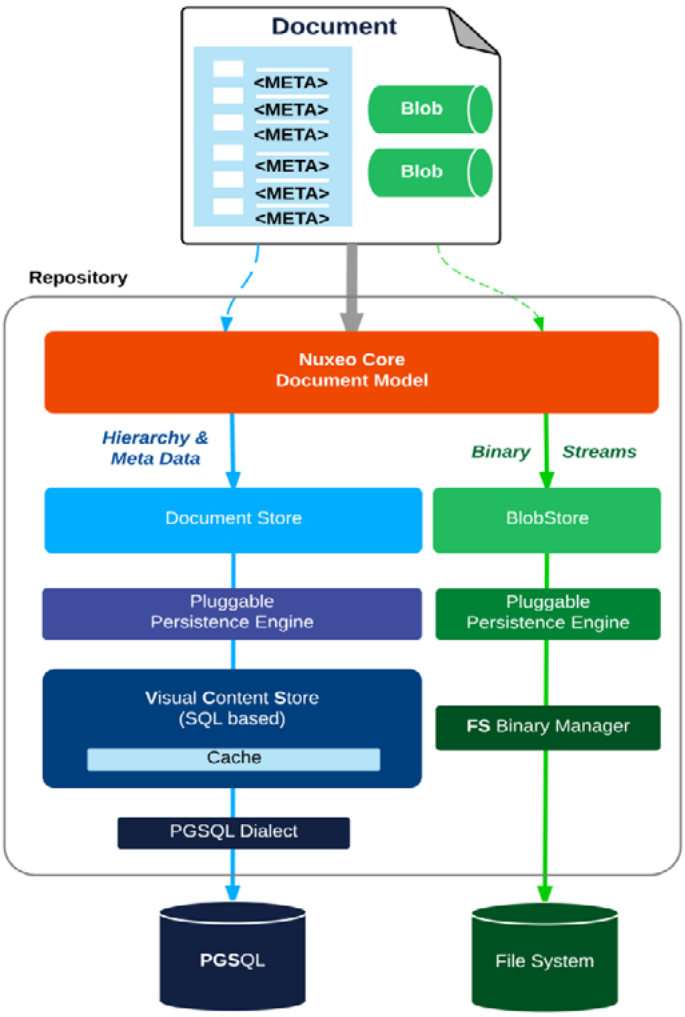


Figure 7. Nuxeo VCS/DBS with PGSQL

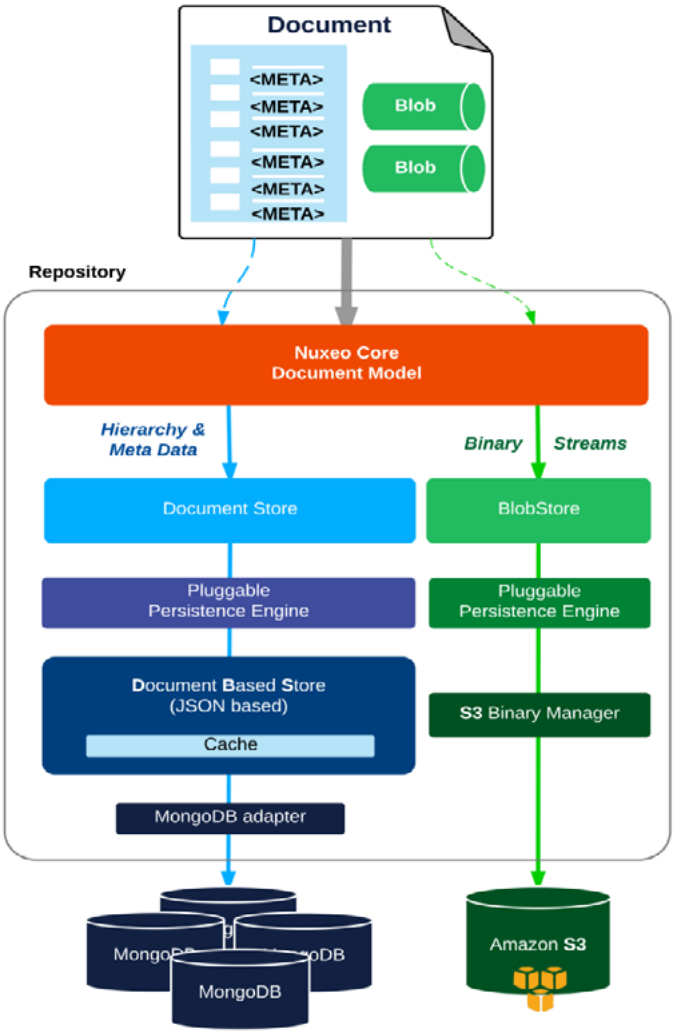


Figure 8. Nuxeo VCS/DBS with MongoDB

About Nuxeo

Nuxeo incorporates all the elements of modern-day architecture. A services-based platform exposing hundreds of content, data, and workflow API's, all delivered on a highly scalable component-based cloud-native architecture.

Let Nuxeo show you how to deliver tomorrows applications today, faster than you thought imaginable. At Nuxeo, we are revolutionizing the way organizations look at content and data together!

Visit **nuxeo.com**