

# Cloud-scale analytics

Article • 12/10/2024

With larger, more sophisticated forms of cloud adoption, your journey to the cloud becomes more complex. Azure cloud-scale analytics is a scalable, repeatable framework that meets your organization's unique needs for building modern data platforms.

Cloud-scale analytics covers both technical and nontechnical considerations for analytics and governance in the cloud. This guidance strives to support hybrid and multicloud adoption by being cloud agnostic, but the included technical implementation examples focus on Azure products.

Cloud-scale analytics has the following goals:

- Serve data as a product, rather than a byproduct
- Provide an ecosystem of data products, rather than a singular data warehouse that might not best fit your data scenario
- Drive a default approach to enforce data governance and security
- Drive teams to consistently prioritize business outcomes instead of focusing just on the underlying technology.

Cloud-scale analytics builds upon Microsoft's cloud adoption framework and requires an understanding of [landing zones](#). If you don't already have an implementation of Azure landing zones, consult your cloud teams about how to meet prerequisites. For more information, see [Ensure the environment is prepared for the cloud adoption plan](#).

Reference architectures allow you to begin with a small footprint and grow over time, adapting the scenario to your use cases.

Cloud-scale analytics includes repeatable templates that accelerate five core infrastructure and resource deployments. It's also adaptable for different organization sizes. If you're a small enterprise with limited resources, a centralized operations model mixed with some business subject matter experts might fit your situation. If you're a larger enterprise with autonomous business units (each with their own data engineers and analysts) as your goal, then a distributed operating model such as data mesh or data fabric might better address your needs.

## Objectives

Cloud-scale analytics provides a framework that is built on the following principles. These principles address challenges with complex data architectures that don't scale to

the needs of organizations.

[+] Expand table

Principle	Description
Allow	<ul style="list-style-type: none"><li>Scaling without increased complexity</li><li>Separation of concerns to facilitate governance</li><li>Creation of self-serve data infrastructure</li></ul>
Follow	<ul style="list-style-type: none"><li>Best practices for well-architected cloud services</li></ul>
Support	<ul style="list-style-type: none"><li>On-premises and multicloud scenarios</li></ul>
Adopt	<ul style="list-style-type: none"><li>Product and vendor agnostic approach</li><li>Cloud Adoption Framework</li></ul>
Commit	<ul style="list-style-type: none"><li>Azure landing zones as baseline infrastructure for all workloads</li><li>Operating model</li></ul>
Enable	<ul style="list-style-type: none"><li>Common data infrastructure</li><li>Distributed architecture under centralized governance</li><li>Secure network line-of-sight</li></ul>

## Implementation guidance

Implementation guidance can be broken into two sections:

- Global guidance that applies to all workloads.
- Cloud-scale specific guidance

### Global guidance

[+] Expand table

Documentation	Description
Cloud Adoption Framework	Managing and governing data is a lifecycle process, which begins by building on your existing cloud strategy and carries all the way through to your ongoing operations. The Cloud Adoption Framework helps guide your data estate's full lifecycle.

Documentation	Description
Azure Well-Architected Framework	Workload architecture and operations have a direct effect on data. Understand how your architecture can improve your management and governance of workload data.

## Cloud-scale specific guidance

[+] Expand table

Section	Description
Build an Initial Strategy	How to build your data strategy and pivot to become a data driven organization.
Define your plan	How to develop a plan for cloud-scale analytics.
Prepare analytics estate	Overview of preparing your cloud-scale analytics estate with key design area considerations like enterprise enrollment, networking, identity and access management, policies, business continuity and disaster recovery.
Govern your analytics	Requirements to govern data, data catalog, lineage, master data management, data quality, data sharing agreements and metadata.
Secure your analytics estate	How to secure analytics estate with authentication and authorization, data privacy, and data access management.
Organize people and teams	How to organize effective operations, roles, teams, and team functions.
Manage your analytics estate	How to provision platform and observability for a scenario.

## Architectures

This section addresses the details of physical implementations of cloud-scale analytics. It maps out the physical architectures of data management landing zones and data landing zones.

Cloud-scale analytics has two key architectural concepts:

- The data landing zone
- The data management landing zone

- Integration with software-as-a-service solutions such as Microsoft Fabric and Microsoft Purview

These architectures standardize best practices and minimize deployment bottlenecks for your development teams, and can accelerate the deployment of common cloud-scale analytics solutions. You can adopt their guidance for lakehouse and data mesh architectures. That guidance highlights the capabilities you need for a well-governed analytics platform that scales to your needs.

For more information, see: [Architectures Overview](#)

## Best practices

The following advanced, level-300+ articles in the **cloud-scale analytics** table of contents can help central IT teams deploy tools and manage processes for data management and governance:

- [Data ingestion for cloud-scale analytics](#)
- [Data lake storage for cloud-scale analytics](#)
- [Use Azure Synapse Analytics for cloud-scale analytics](#)

## Featured Azure products

Expand the **Featured Azure products** section in the **cloud-scale analytics** table of contents to learn about the Azure products that support cloud-scale analytics.

## Common customer journeys

The following common customer journeys support cloud-scale analytics:

- **Prepare your environment.** Use the [Prepare your environment](#) articles as resources. Establish processes and approaches that support the entire portfolio of workloads across your data estate.
- **Influence changes to individual workloads.** As your cloud-scale analytics processes improve, your central data governance teams find requirements that depend on knowledge of the architecture behind individual workloads. Use the [Architecture](#) articles to understand how you can use the scenarios within for your use case.
- **Optimize individual workloads and workload teams.** Start with the [Azure Well-Architected Framework](#) guidance to integrate cloud-scale analytics strategies into

individual workloads. This guidance describes best practices and architectures that central IT and governance teams should use to accelerate individual workload development.

- **Use best practices to onboard individual assets.** Expand the **Best practices** section in the **cloud-scale analytics** table of contents to find articles about processes for onboarding your entire data estate into one cloud-scale analytics control plane.
- **Use specific Azure products.** Accelerate and improve your cloud-scale analytics capabilities by using the Azure products in the **Featured Azure products** section of the **cloud-scale analytics** table of contents.

## Take action

For more information about planning for implementing the cloud-scale analytics, see:

- [Develop a plan for cloud-scale analytics](#)
- [Introduction to cloud-scale analytics](#)

## Next steps

Begin your cloud-scale analytics journey:

[Introduction to cloud-scale analytics](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Introduction to cloud-scale analytics

Article • 01/08/2025

Cloud-scale analytics builds upon Azure landing zones to simplify deployment and governance. The main purpose of an Azure landing zone is to ensure that, when you deploy an application or workload on Azure, the required infrastructure is already in place. Before you deploy your cloud-scale analytics landing zone, you need to work through the [Cloud Adoption Framework for Azure](#) to deploy an [Azure landing zone architecture](#) that has platform landing zones.

For sovereign workloads, Microsoft provides the [Sovereign Landing Zone \(SLZ\)](#), which is a variant of the enterprise-scale Azure landing zone. The SLZ is intended for organizations that need advanced sovereign controls. Cloud-scale analytics can be deployed against this Azure landing zone variant.

Cloud-scale analytics involves deploying to application landing zones. These zones typically reside under the landing zone management group. Policies filter down to the sample templates that Microsoft provides.

You can use these sample templates for your data lakehouse and [data mesh](#) deployments.

## Cloud-scale analytics evaluation

Often, a business seeks clarity or prescriptive guidance before it starts to define the technical details for a specific use case or project, or for end-to-end cloud-scale analytics. As a business formulates its overall data strategy, it can be challenging to ensure that all the required and strategic principles in the scope of the current use are taken into account.

To speed up the delivery of this end-to-end insights implementation, while taking these challenges into account, Microsoft has developed a prescriptive scenario for cloud-scale analytics. It aligns with the key themes that are discussed in [Develop a plan for cloud-scale analytics](#).

Cloud-scale analytics builds on the Cloud Adoption Framework and applies the principles of the Azure Well-Architected Framework. The Cloud Adoption Framework provides prescriptive guidance and best practices for cloud operating models, reference architectures, and platform templates. This guidance is based on real-world experiences from some of our most challenging, sophisticated, and complex environments.

Cloud-scale analytics helps you prepare to build and operationalize landing zones to host and run analytics workloads. You build the landing zones on the foundations of enhanced security, governance, and compliance. Landing zones are scalable and modular, but they support autonomy and innovation.

## History of data architecture

In the late 1980s, data warehouse generation 1 was introduced. This model combines disparate data sources from across an enterprise. In the late 2000s, generation 2 emerged, with the introduction of big data ecosystems like Hadoop and data lakes. The mid-2010s brought the cloud data platform: streaming data ingestion, like Kappa or Lambda architectures, were introduced. In the early 2020s, data lakehouses, data meshes, data fabrics, and data-centric operational patterns were introduced.

In spite of these advances, many organizations still use the centralized monolithic platform: generation 1. This system works well, up to a point. However, bottlenecks can occur because of interdependent processes, tightly coupled components, and hyperspecialized teams. Extract, transform, and load (ETL) jobs can become prominent and slow down delivery timelines.

Data warehouses and data lakes are still valuable and play an important role in your overall architecture. The following documentation highlights some of the challenges that can occur when you use these traditional practices for scaling. These challenges are especially relevant in a complex organization, where data sources, requirements, teams, and outputs change.

## Moving to cloud-scale analytics

Your current analytical data architecture and operating model can include data warehouse, data lake, and data lakehouse structures, data fabric, or data mesh.

Each data model has its own merits and challenges. Cloud-scale analytics helps you shift your current approach to data management so that it can evolve with your infrastructure.

You can support any data platform and scenario to create an end-to-end cloud-scale analytics framework that serves as your foundation and allows for scaling.

## Modern data platform and desired outcomes

One of the first steps is to activate your data strategy to meet your challenges by iteratively building a scalable and agile modern data platform.

Instead of being overwhelmed with service tickets and trying to meet competing business needs, when you implement a modern data platform, you can play a more consultative role because you can free up your time to focus on more valuable work. You provide lines of business with the platform and systems to self-serve data and analytics needs.

Following are recommended areas of initial focus:

- Improve data quality, facilitate trust, and gain insights to make data-driven business decisions.
- Implement holistic data, management, and analytics at scale, across your organization.
- Establish robust data governance that enables self-service and flexibility for lines of business.
- Maintain security and legal compliance in a fully integrated environment.
- Quickly create the foundation for advanced analytics capabilities by using an out-of-the-box solution of well-architected, repeatable, modular patterns.

## Govern your analytics estate

A second consideration is to determine how your organization will implement data governance.

Data governance is the process of ensuring that the data you use in your business operations, reports, and analysis is discoverable, accurate, trusted, and that it can be protected.

For many companies, the expectation is that data and AI will drive a competitive advantage. As a result, executives are eager to sponsor AI initiatives in their determination to become data driven. However, for AI to be effective, it must use trusted data. Otherwise, decision accuracy can be compromised, decisions might be delayed, or actions might be missed, which can affect the outcome. Companies don't want the quality of their data to be poor. Until you review the effect that digital transformation has had on data, it might seem simple to fix data quality.

Organizations that have data spread across a hybrid multicloud and distributed-data landscape struggle to find where their data is and to govern it. Ungoverned data can have a considerable effect on business. Poor data quality affects business operations because data errors cause process errors and delays. Poor data quality also affects

business decision-making and the ability to remain compliant. Ensuring data quality at the source is often preferred because fixing quality issues in the analytical system can be more complex and costly than applying data quality rules early in the ingestion phase. To help you track and govern data activity, data governance must include:

- Data discovery.
- Data quality.
- Policy creation.
- Data sharing.
- Metadata.

## Secure your analytics estate

Another major driver for data governance is data protection. Data protection can help you ensure compliance with regulatory legislation and can prevent data breaches. Data privacy and the growing number of data breaches have made data protection a top priority. Data breaches highlight the risk to sensitive data, such as personally identifiable customer data. The consequences of data privacy violation or a data security breach can include:

- Serious damage to brand image.
- Loss of customer confidence and market share.
- A reduction in share price, which affects stakeholder return on investment and executive salaries.
- Significant financial penalties because of audit or compliance failures.
- Legal action.
- Secondary effects of the breach, for example, customers might fall victim to identity theft.

In most cases, publicly quoted companies must declare breaches. If breaches occur, customers are likely to blame the company rather than the hacker. Customers might boycott the company for several months or might never return.

Failure to comply with regulatory legislation on data privacy can result in significant financial penalties. Governing your data helps you avoid these risks.

## Operating model and benefits

Adopting a modern data strategy platform doesn't just change the technology that your organization uses. It also changes how the organization operates.

Cloud-scale analytics provides guidance to help you organize and train your employees, including:

- Persona, role, and responsibility definitions.
- Suggested structures for agile, vertical, and cross-domain teams.
- Training resources, including Azure data and AI certifications via Microsoft Learn.

It's also important to engage your end users throughout the modernization process and as you continue to evolve your platform and onboard new use cases.

## Architectures

Azure landing zones represent the strategic design path and target technical state for your environment. They make deployment and governance easier so that you can improve agility and compliance. They also ensure that, when a new application or workload is added to your environment, the proper infrastructure is already in place. Azure data management and data landing zones, integrated with Microsoft software as a service (SaaS) governance and analytics solutions, are designed with these foundational principles in mind and, when combined with the other elements of cloud-scale analytics, can help to enable:

- Self-service.
- Scalability.
- A fast start.
- Security.
- Privacy.
- Optimized operations.

## Data management landing zone

The data management landing zone provides the foundation for your platform's centralized data governance and management across your organization. It also facilitates communication to ingest data from your entire digital estate, including multicloud and hybrid infrastructures.

The data management landing zone supports numerous other data management and governance capabilities, such as:

- Data catalogs.
- Data quality management.
- Data classification.
- Data lineage.

- Data modeling repositories.
- API catalogs.
- Data sharing and contracts.

### 💡 Tip

If you use partner solutions for data catalog, data quality management, or data lineage capabilities, they should reside in the data management landing zone. Alternatively, you can deploy Microsoft Purview as a SaaS solution, connecting to both the data management landing zone and the data landing zones.

## Data landing zones

Data landing zones bring data closer to users and enable self-service while maintaining common management and governance via connection to the data management landing zone.

They host standard services like networking, monitoring, and data ingestion and processing, in addition to customizations like data products and visualizations.

Data landing zones are key to enabling your platform's scalability. Depending on your organization's size and needs, you can start with one or multiple landing zones.

When you decide between single and multiple landing zones, consider regional dependencies and data residency requirements. For example, are there local laws or regulations that require data to stay in a specific location?

Regardless of your initial decision, you can add or remove data landing zones as needed. If you start with a single landing zone, we recommend that you plan to extend to multiple landing zones to avoid future needs for migration.

### ⓘ Note

Where Microsoft Fabric is deployed, the data landing zone hosts non-SaaS solutions like data lakes and other Azure data services.

For more information about landing zones, see [Azure landing zones for cloud-scale analytics](#).

## Conclusion

After you read this documentation set, in particular the governance, security, operating, and best practices sections, we recommend that you set up a proof-of-concept environment by using the deployment templates. These templates, along with architecture guidance, give you hands-on experience with some of the Azure and Microsoft SaaS technologies. For more information, see the [Getting started checklist](#).

## Next step

[Integrate cloud-scale analytics into your cloud adoption strategy](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Integrate cloud-scale analytics into your cloud adoption strategy

Article • 11/27/2024

Create a single, centralized cloud adoption strategy for your organization using the [Strategy methodology in Azure's Cloud Adoption Framework](#). If you haven't already recorded your cloud adoption strategy, use the [strategy and plan template](#) to do so.

This article contains considerations for cloud-scale analytics scenarios that affect your broader strategy.

Before implementing cloud-scale analytics, have a plan in place for your data strategy. You can start small with a single use case, or you can have a larger set of use cases that require prioritization. Having a strategy helps you establish your processes and spark initial conversations about pillars you need to focus on.

## Prioritize business outcomes for your data strategy

Having a successful data strategy gives you a competitive advantage. You should always align your data strategy with your desired business outcomes. Most business outcomes can be classified into one of the following four categories:

- **Empower your employees:** Provide your workforce with real-time knowledge of customers, devices, and machines. This knowledge helps them efficiently collaborate to meet customer or business needs with agility.
- **Engage with customers:** Deliver a rich, personalized, and connected experience inspired by your brand. Harness the power of data and insights to drive customer loyalty along every step of a customer journey.
- **Optimize operations:** Increase the flow of information across your entire organization. Synchronize your business processes and use a data-driven approach to make every interaction valuable.
- **Transform your products and development life cycle:** Gather telemetry data about your services and offerings. Use the telemetry data to prioritize a release or create a new feature, and to evaluate effectiveness and adoption continuously.

After prioritizing your business outcomes, examine your current projects and long-term strategic initiatives and classify them accordingly. Consider combining the four

categories of business outcomes in a matrix format that's based on complexity and impact. Also, consider adding architectural pillars to help you dive deeper into your scenario.

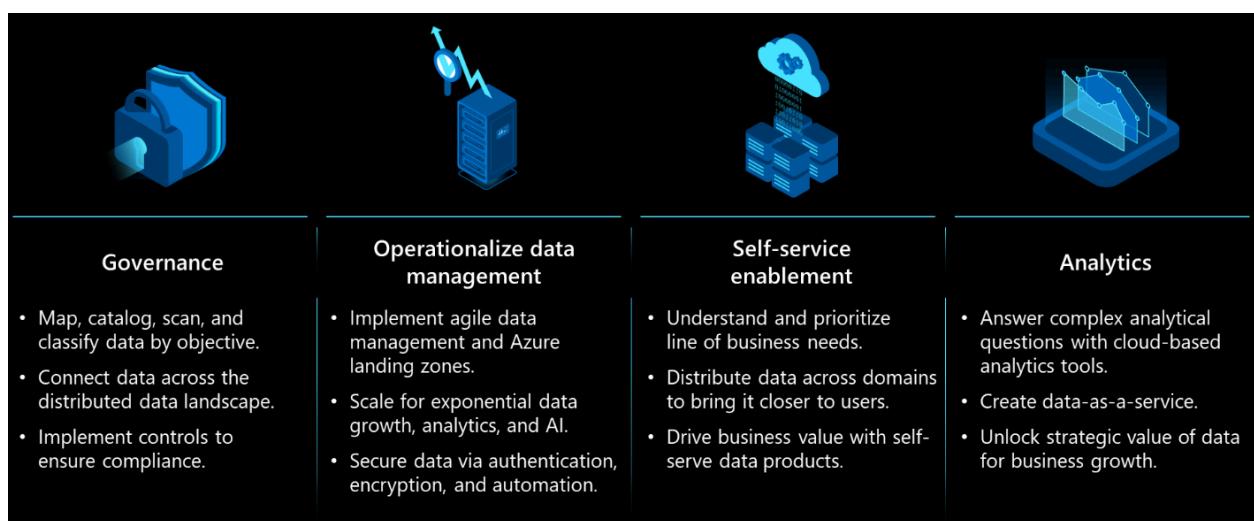
## Unlock strategic value

Building a data-driven culture that drives business forward in a consistent, forward-thinking, agile, and informed way has some inherent complexities and ground realities. Before you enter your deployment phase, focus efforts towards the formation of a coherent data strategy that can help you achieve your desired business outcomes.

Cloud-scale analytics are aligned with [innovation-focused motivations](#). The following common drivers motivate customers to integrate this scenario into their cloud adoption strategy:

- A scalable analytics framework, which lets you build an enterprise data platform
- Self-service, which empowers users in data exploration, data asset creation, and product development
- A data-led culture with reusable data assets, data communities, secure third-party exchange, and in-place sharing
- Sharing data with confidence, using policies, common identity, confidentiality, and encryption
- Improved customer experiences and engagements
- Transformation of products or services
- Market disruption with new products or services

The following diagram contains key themes that help you realize these motivations in your own strategy. Carefully analyze these themes and how they contribute to a coherent data strategy. Also, consider how they can unlock your data's strategic value and enable consistent business growth.



*"A data strategy is the foundation to using data as an asset and driving business forward. It's not a patch job for data problems. It's a long-term, guiding plan that defines the people, processes, and technology to put in place to solve data challenges."*

Creating your strategy is one step. Executing your strategy at an enterprise scale poses a great challenge to your organization's existing culture, people, processes, and technology choices. Execution requires commitment and clear ownership at all levels of your organization.

## Increasing efficiencies

The agility of the cloud requires organizations to adapt quickly and bring efficiencies to all areas of business. According to the [report on emerging risks by Gartner](#), despite organizations continuing to focus on and invest in digital initiatives, two-thirds of these organizations demonstrate enterprise weaknesses and fail to deliver upon expectations, even though they continue to focus on and invest in digital initiatives.

## Operationalize data management

Many organizations have slowly been decentralizing central IT to enable agility. Organizations want to innovate quickly, and having access to enterprise-wide unified data in a self-serve manner helps them meet challenging business requirements.

There are many reasons why businesses fail to tap into the full potential of their data. It might be because business functions work in silos, where each team is using different tools and standards for data analysis. Or it might be because of a failure to link key performance indicators to overall business goals.

Data democratization helps you deliver value back to business and achieve challenging business growth targets.

- Understand and prioritize your LOBs needs.
- Distribute your data across domains to enable ownership and bring data closer to users.
- Deploy self-service data products to drive insights and business value.

For data governance, you must strike a proper balance in the decentralized world of data democratization. If you enforce governance too strictly, you can stifle innovation. However, if you don't have at least some core principles and processes in place, you're likely to end up with data silos. These silos can damage your organization's reputation and potential revenues. A holistic data governance approach is fundamental for you to unlock the strategic value of your data in a consistent manner.

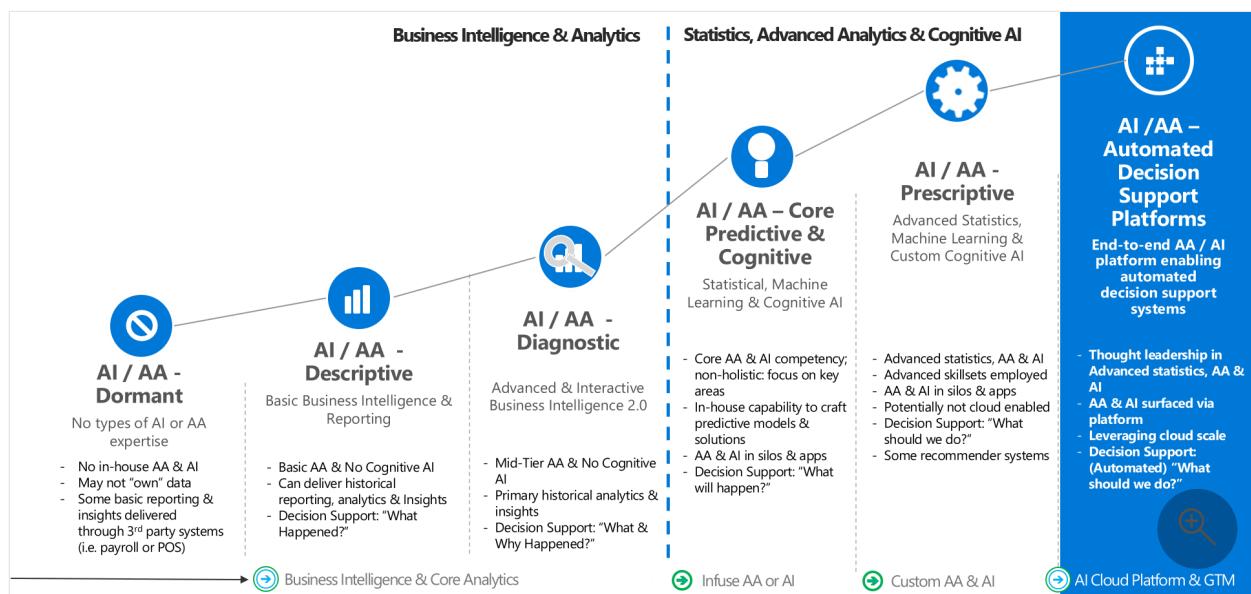
The absence of a well-thought-out data strategy leads to a need to just "get going" and quickly start providing value to your organization. Address current business problems by acting on the previously mentioned key themes or using them as strategic principles within a framework. Using these key themes can also help you create a holistic data strategy that is iterative with validation, yet still provides timely results. Business and technology leaders must develop the strategies and mindset required to generate value from data and quickly scale in a simplified, structured manner.

For more information, see [What is data governance?](#).

## Develop a data-driven culture

To build a successful data strategy, you need a data-driven culture. Develop a culture that consistently fosters open, collaborative participation. In this type of culture, your entire workforce can learn, communicate, and improve the organization's business outcomes. Developing a data-driven culture also improves each employee's ability to generate impact or influence backed by data.

Your journey's starting point depends on your organization, your industry, and your current location along the maturity curve. The following diagram shows an example maturity model outlining the maturity levels of an organization's AI usage:



## Level 0

Data isn't exploited programmatically and consistently. The organization's data focus is from an application development perspective.

At Level 0, the organization often has unplanned analytics projects. Each application is highly specialized to unique data and stakeholder needs. Each application also has

significant code bases and engineering teams, with many engineered outside IT. Use case enablement and analytics are siloed.

## **Level 1**

At Level 1, teams are being formed and strategy is being created, but analytics remains departmentalized. The organization tends to be good at traditional data capture and analytics. It might have some level of commitment to a cloud-scale approach. For example, it might already access data from the cloud.

## **Level 2**

The organization's innovation platform is almost ready. Workflows are in place to address data quality. The organization can answer a few "why" questions.

At Level 2, the organization is actively searching for an end-to-end data strategy that uses centrally governed data lake stores to control data store sprawl and improve data discoverability. The organization is ready for smart applications that bring compute to centrally governed data lakes. These smart applications reduce privacy risks, compute costs, and the need for federated copies of important data.

At this level, the organization is also ready to use multitenant, centrally hosted, shared data services for common data computing tasks. These shared data services enable rapid insights from data-science-driven intelligence services.

## **Level 3**

The organization uses a holistic data approach. Projects related to data are integrated within business outcomes. The organization uses analytics platforms to make predictions.

At Level 3, the organization unlocks digital innovation from both data estate and application development standpoints. Foundational data services are in place, including data lakes and shared data services.

Multiple teams across the organization successfully deliver on critical business workloads, key business use cases, and measurable outcomes. New shared data services are identified using telemetry. IT is a trusted advisor to teams across the company, using a trusted and connected end-to-end data strategy to help improve critical business processes.

## Level 4

At Level 4, the entire organization uses frameworks, standards, enterprise, and a data-driven culture. Automation, data-driven feedback loops, and centers of excellence around analytics or automation can be observed in action.

## Develop business-aligned objectives

Identifying priorities in line with the business vision and keeping a "think big, start small, and act fast" ideology are keys to success. Picking up the right use case doesn't always need to be a long-haul, difficult vetting process. It could be an ongoing problem in any business unit where there's enough data to validate its return on investment, more appetite, and easy buy-in. Things can move quickly, and that's where most of the organization can be struggling to get started.

## Understand data attributes

To build a strong data strategy, you need to understand how data works. Knowing data's core characteristics helps you build a principled practice for dealing with data.

Data travels fast, but its velocity can't defy the laws of physics. Data must conform to the laws of the land and the industry that created it.

Data doesn't change on its own, but it's prone to changes and accidental loss unless you put measures in place to mitigate such challenges. Put anti-corruption measures for controls, databases, and storage in place so you can deal with unforeseen changes. Also, ensure you set up monitoring, audits, alerts, and downstream processes.

On its own, data doesn't produce any insights or yield any value. To gain insights or extract value, you must put most or all of your data through four discrete steps:

1. Ingestion
2. Storage
3. Processing
4. Analytics

Each of these four steps has its own principles, processes, tools, and technologies.

Withholding your data assets and related insights can affect socioeconomic, political, research, and investment decisions. It's critical that your organization is capable of providing insights in a secure and responsible way. All data you generate or acquire

must go through a data classification exercise unless otherwise explicitly stated. Encryption is the gold standard for handling confidential data both at rest and in transit.

Data, applications, and services all have their own gravitational pulls, but data's pull is the largest. Unlike Sir Isaac Newton's legendary apple, data doesn't have any physical mass that affects surrounding objects. It instead has latency and throughput, which act as accelerators for your analytics process. Latency, throughput, and ease of access often require you to duplicate data, even when that isn't desirable. Set up your people, processes, tools, and technologies appropriately so you can balance such requirements with your organization's data policies.

Architectural constructs govern the speed at which you can process data. Constructs are facilitated through innovations in software, hardware, and networking. Some architectural considerations are:

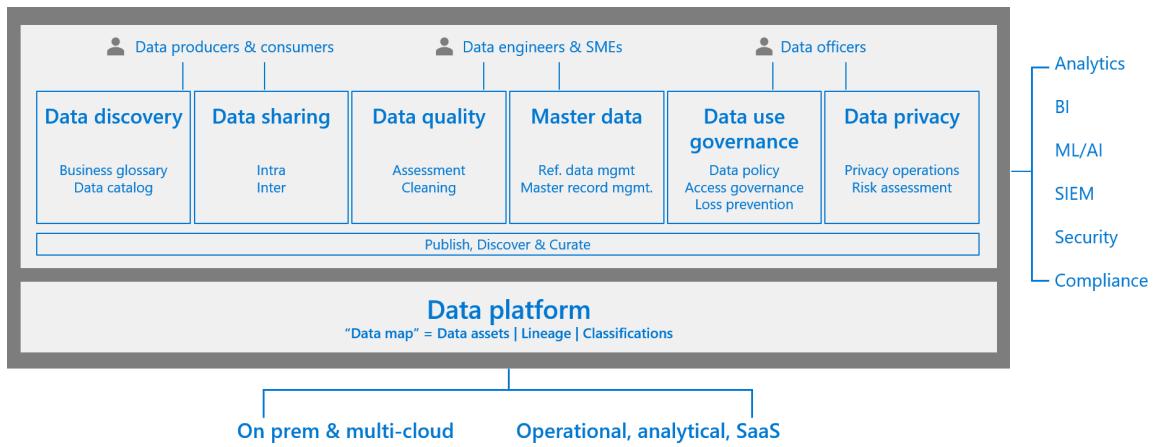
- Setting up data distribution
- Partitioning
- Cache technologies
- Batch versus stream processing
- Balancing back-end and client-side processing

## Define your data strategy

Using data as a competitive advantage for building better products and higher-value services is not a new concept. However, the volume, velocity, and variety of data enabled by cloud computing are unprecedented.

The design of a modern data analytics platform in the cloud consists of security, governance, monitoring, on-demand scaling, data operations, and self-service. Understanding the interplay between these facets is what distinguishes a great data strategy from a good one. Use tools like the Cloud Adoption Framework to ensure architectural cohesiveness, integrity, and best practices.

To be effective, your data strategy must contain provisions for data governance. The following diagram shows the main stages of a data life cycle, focusing on data governance as its focus:



The following sections describe considerations you should use while deciding on design principles for your data strategy's layers. Focus on delivering business outcomes and value from your data.

## Data ingestion

A key consideration for data ingestion is your ability to build a data pipeline quickly in a secure and compliant manner, from requirements all the way to production. Important elements include metadata-driven, self-service, and low-code technologies that hydrate your data lake.

When building pipelines, consider both design and your ability to wrangle data, distribute data, and scale compute. You must also ensure you have the right DevOps support for your pipeline's continuous integration and delivery.

Tools like Azure Data Factory support a plethora of on-premises data sources, software as a service (SaaS) data sources, and other data sources from other public clouds.

## Storage

Tag and organize your data in both physical and logical layers. Data lakes are part of all modern data analytics architectures. Your organization must apply appropriate data privacy, security, and compliance requirements that meet all data classification and industry compliance requirements you operate under. Cataloging and self-service aid organization-level data democratization, which fuels your innovation while being guided by appropriate access control.

Choose the right storage for your workload. Even if you don't get storage exactly correct the first time, the cloud allows you to fail over quickly and restart your journey. Use your application requirements to choose the best database. Be sure to consider your ability to process batch and streaming data as you're choosing your analytics platform.

## Data processing

Your data processing needs vary with each workload. Most large-scale data processing contains elements of both real-time and batch processing. Most enterprises also have elements of time series processing requirements and a need to process free-form text for enterprise search capabilities.

Online transaction processing (OLTP) provides the most popular organizational processing requirements. Some workloads need specialized processing like high-performance computing (HPC), sometimes called "big compute." These workloads solve complex mathematical tasks using many CPU or GPU-based computers.

For certain specialized workloads, customers can secure execution environments like Azure confidential computing, which helps users secure data while the data is in use within public cloud platforms. This state is required for efficient processing. Data is protected inside a trusted execution environment (TEE), also known as an enclave. A TEE protects code and data against any outside viewing and modification. TEEs allow you to train AI models without sacrificing data confidentiality, even while you use data sources from different organizations.

## Analytical processing

The extract, transform, load (ETL) construct relates to online analytical processing (OLAP) and data warehousing needs. A business-aligned data model and a semantic model allowing organizations to implement business rules and Key Performance Indicators (KPIs) are often implemented as part of the analytical process. One useful capability is automatic schema drift detection.

## Data strategy summary

Taking a principled approach to other considerations, like data governance and responsible AI, pays dividends later on.

At Microsoft, we follow four core principles: fairness, reliability and safety, privacy and security, and inclusiveness. The two foundational principles of transparency and accountability underpin all four core principles.

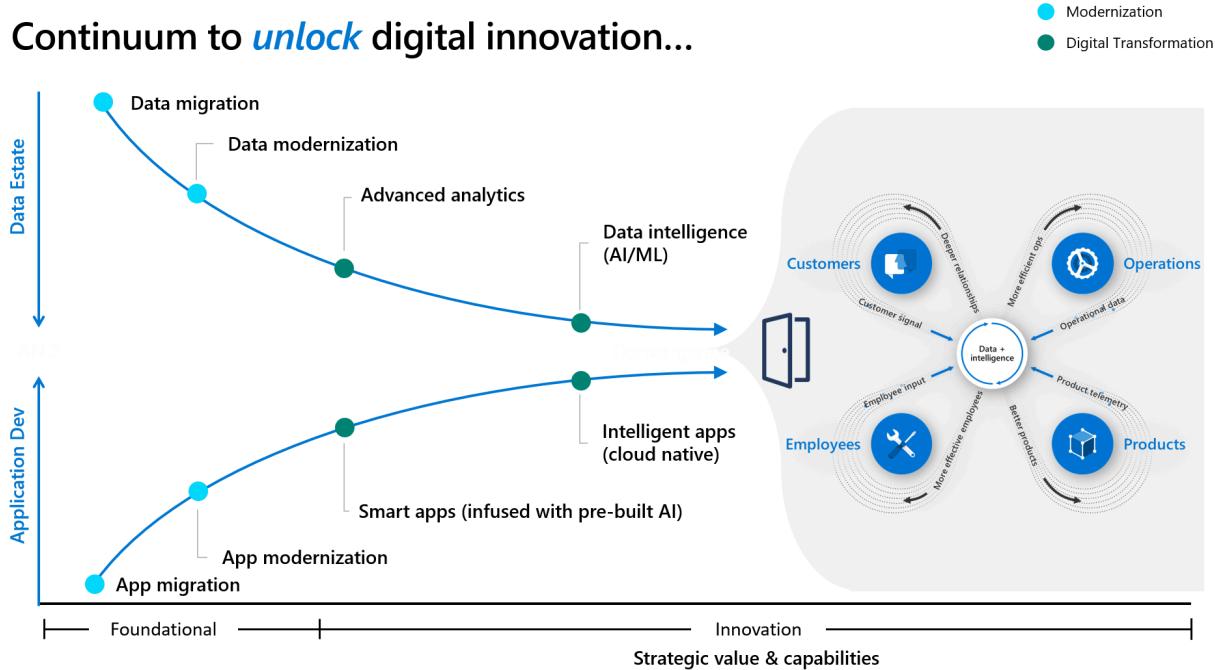
We put our principles and responsible AI into practice by developing resources and a system of governance. Some of our guidelines address human/AI interaction, conversational AI, inclusive design, an AI fairness checklist, and a data sheet for datasets.

We've also developed a set of tools to help others understand, protect, and control AI at every stage of innovation. These tools are a result of multidisciplinary collaboration efforts to strengthen and accelerate responsible AI. Collaboration has spanned software engineering and development, social sciences, user research, law, and policy.

To improve collaboration, we open-sourced many tools like InterpretML and Fairlearn. Others can contribute to and build upon these open-source tools. We also democratized tools through Azure Machine Learning.

The pivot to becoming a data-driven organization is fundamental to delivering competitive advantage in the new normal. We want to help our customers shift from an application-only approach to an application and data-led approach. An approach focusing on applications and data helps create an end-to-end data strategy that ensures repeatability and scalability across current and future use cases that affect business outcomes.

### Continuum to *unlock* digital innovation...



## Foster commitment, communication, and engagement

All key roles involved in making your data strategy a success must clearly understand your adopted approach and common business objectives. Your key roles might include a leadership team (C-level), business units, IT, operations, and delivery teams.

Communication is one of the most important parts of this framework. Your organization must devise a process for effective communication across roles. Communication helps you deliver effectively in the context of your current project. It also establishes a forum

that helps everyone involved remain in line, up to date, and focused on the overall objective of building a holistic data strategy for your future.

Engagement is essential between the following two groups:

- Team members who design and implement the data strategy
- Team members who contribute to, consume, and exploit the data (such as business units that make decisions and build outcomes based upon the data)

To put it another way, data strategies and associated data platforms that are built without user engagement risk challenges in relevance and adoption.

Two strategic processes help you deliver successfully in this framework:

- Formation of a center of excellence
- Adoption of an agile delivery method

For more information, see [Develop a plan for cloud-scale analytics](#).

## Deliver value

When you deliver data products against the success criteria in a standardized and structured way, that delivery validates your iterative framework. Additionally, using your learning to continuously innovate helps you build business confidence and widen data strategy goals. This process provides clearer and faster adoption across your organization.

The same applies to your data platform. When you have a setup where multiple teams operate fairly autonomously, you should drive towards a mesh. Getting there's an iterative process. In many cases, it requires significant changes to your organizational setup, readiness, and business alignment.

## Next steps

Read the following articles to find guidance for your cloud adoption journey and make your cloud adoption scenario successful:

- [Develop a plan for cloud-scale analytics](#)
- [Review your environment for Azure landing zones](#)
- [Govern cloud-scale analytics](#)
- [Secure cloud-scale analytics](#)

# Feedback

Was this page helpful?

 Yes

 No

# Develop a plan for cloud-scale analytics

Article • 11/27/2024

The Cloud Adoption Framework's [Plan methodology](#) helps you create an overall cloud adoption plan to guide all programs and teams involved in your cloud-based digital transformation. The Plan methodology also provides templates to help you create your backlog and plans to help your teams build necessary skills. The backlog and plans you create should be based on what you plan to do in the cloud.

This article provides further guidance for data estate rationalization and skilling plans that are specific to cloud-scale analytics.

## Data estate rationalization

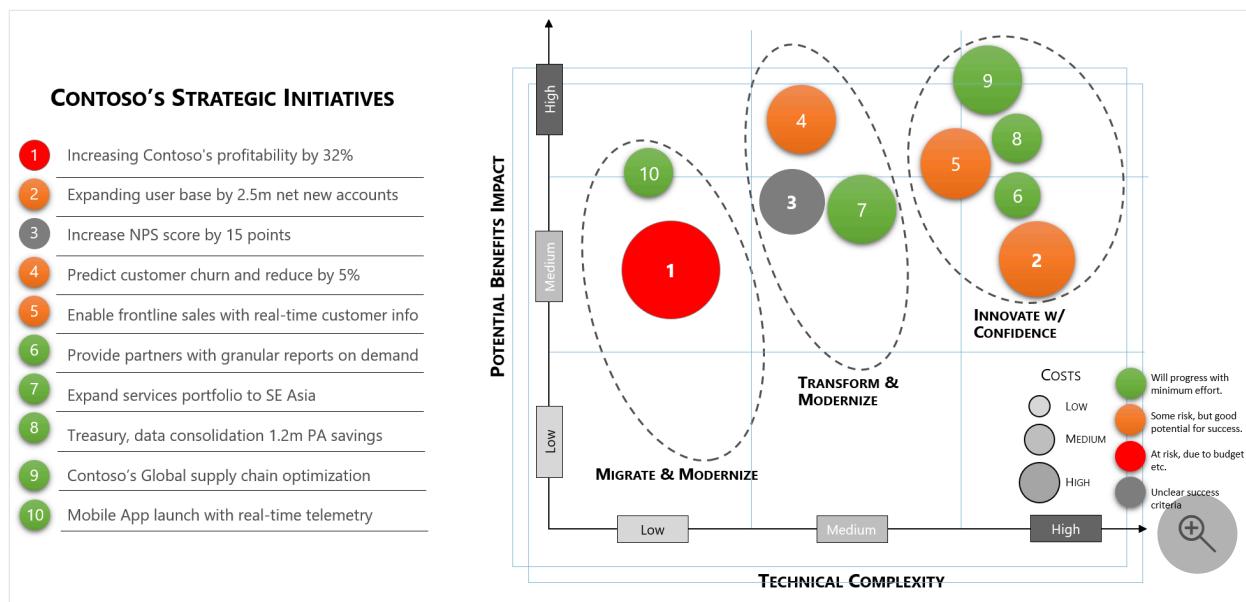
Much of the guidance in the Plan methodology focuses on the [five Rs of rationalizing your digital estate](#).

Using a cloud-scale analytics scenario shifts the primary focus of rationalization to the **data estate**, a subset of the overall digital estate. Your organization must evaluate the data estate more broadly and deeply than other scenarios require. Include plans for the overall analytics and [data governance](#) needed to support your desired maturity.

## Strategic initiatives

Begin to properly rationalize your data estate by aligning your business outcomes with each of your data initiatives. This alignment allows you to prioritize and clearly understand what value you can derive from each data initiative.

In your cloud migration plan, initiatives with small business impacts and lesser migration complexity can deliver quick efficiency gains. Initiatives with large business impacts or greater technical complexity require more detailed planning, but they can provide long-term innovation value.



## Prioritization

To begin prioritizing data projects, complete an [inventory and benchmark of your data estate](#). You can use tools like Azure Migrate to capture rich benchmarking data from the infrastructure and data assets in your estate. This benchmarking data helps you track progress and measure success. It can also help you quantify the exact investment needed for people, processes, and technology.

A mapping of business impact (from your strategic business outcomes) and technical complexity (from your data estate inventory) can guide your prioritization of data projects. The mapping achieves this prioritization by helping you identify waves of your cloud adoption effort. The waves can guide you as you prioritize data projects. The following table describes these cloud adoption waves in more detail.

[\[+\] Expand table](#)

Wave	Rationalization	Outcomes
Migrate & Modernize	Rehost and refactor	Quick, tactical wins can be included in standard migration projects alongside other applications and infrastructure. Use tools like Azure Migrate to automate this type of one-time cloud migration. This approach allows you to modernize data platforms to Azure SQL Database, Azure Cosmos DB, or other transactional data structures.
Transform & Modernize	Rehost and refactor	When business value increases, so can the complexity of data estate management. Some amount of transmission, transformation, and synchronization is likely required to keep on-premises processes running while enabling richer functions in the cloud. Use tools like Azure Data Factory to help with the

Wave	Rationalization	Outcomes
		ongoing transformation after your data asset is migrated and modernized.
Innovate with confidence	Rearchitect or rebuild	Achieving high business value requires an ability to innovate with confidence. Use cloud-native data tools to democratize data, analyze information, and predict outcomes.

## Workload identification

Strategic initiatives are delivered by the workloads that run on top of your data environment. To properly architect workloads, you must first identify the workloads running within your data estate. The identification process can be complex. Data workloads can include one or more data sources. They can also include multiple processes for preparing data, analyzing information, or predicting outcomes.

Use the previously described wave planning approach to simplify workload identification. For each wave, identify the data sources, applications, and infrastructure required to deliver your strategic initiative. Use the Azure Migrate tool to evaluate their dependencies and clearly understand workload groupings.

Transactional data assets are typically associated with an existing application, making workload identification easier.

Analysis and AI/machine learning solutions can be more complex, requiring a more granular review of the outcomes delivered by each. Associate analysis and AI solutions with the business processes that consume their outputs, often creating an application-level mapping. For cross-application BI, AI, or machine learning solutions, create new workload names to map the data assets to the business processes they impact.

Workloads identified in the digital estate assessment can be used throughout your adoption to drive business impact classification. Record the derived values using the [naming and tagging standards](#) that apply to all Azure cloud adoption efforts.

Identifying workloads will also help you gain a better understanding of the skills your teams need to be successful.

## Develop a skilling plan

Developing a skilling plan is part of building your capability to drive your data strategy. It's important to create a clear mapping of your product, services, or tools and your

organization's people skills. The following exercise helps you to develop your skilling plan by preparing early and practicing agility.

## Prepare your plan with these tips

This section provides useful tips for developing your skilling plan.

### Prepare for potential challenges and roadblocks early

Harnessing the power of data in a secure and compliant manner is a challenge. You can run into various difficulties throughout the process, including:

- Organizational silos dividing your organization
- Roadblocks in your effort to build a data-driven culture
- Multiple tools and technologies being in use across your organization

Time-to-market is one of the most critical factors for any business. Your organization can have an excellent idea and the data to enable it, but challenges and roadblocks can significantly extend your time-to-market. An unexpected challenge might prevent you from gaining insights and business value from your data for weeks or months. It's important for you to prepare for potential challenges and roadblocks early, so you minimize the impact they can have on your time.

### Adopting agile delivery method

Agile is the ability to create and respond to change. It's a way to deal with, and ultimately succeed in, any uncertain and turbulent environment.

Agility requires you to think through what's going on in your current environment, identify any uncertainties, and plan how to adapt as you go.

## Next steps

The following articles can guide your cloud adoption journey and help your cloud adoption scenario to succeed:

- [Review your environment for Azure landing zones](#)
- [Govern cloud-scale analytics](#)
- [Secure cloud-scale analytics](#)

# Feedback

Was this page helpful?

 Yes

 No

# Azure landing zones for cloud-scale analytics

Article • 12/10/2024

In response to the need for frictionless governance and a platform for actionable insights to the business, cloud-scale analytics represents a strategic design path and targets the technical state for an Azure analytics and AI environment.

The pattern relies upon the distribution of the data and its pipelines across domains. This pattern enables ownership of accessibility, usability, and development. Largely based on these patterns, cloud-scale analytics includes the following capabilities:

- Storage
- Data governance
- Data ingestion
- Data quality
- Access provisioning
- Networking
- Encryption
- Resiliency
- Observability

## ⓘ Note

Cloud-scale analytics builds on the [Start with Cloud Adoption Framework enterprise-scale landing zones](#) and should be considered a supplement to it.

Cloud-scale analytics builds on top of the Microsoft Cloud Adoption Framework while applying our Well-Architected framework lens. The Microsoft Cloud Adoption Framework provides prescriptive guidance and best practices on cloud operating models, reference architecture, and platform templates. It's based on real-world learnings from some of our most challenging, sophisticated, and complex environments.

Cloud-scale analytics paves the way for customers to build and operationalize landing zones to host and run analytics workloads. You build the landing zones on the foundations of security, governance, and compliance. They're scalable and modular while supporting autonomy and innovation.

Cloud-scale analytics considers five critical design areas that help translate organizational requirements to Azure constructs and capabilities. Lack of attention to

these design areas typically creates dissonance and friction between the enterprise-scale definition and Azure adoption. Cloud-scale analytics uses these design areas to help address the mismatch between on-premises and cloud-design infrastructure.

To learn more, see:

- [Data management landing zone](#)
- [Data landing zone](#)
- [Data products](#)
- [Data platform operational excellence](#)

## Data management landing zone

At the heart of cloud-scale analytics is its management capability. This capability is enabled through the data management landing zone.

For more information, see [Data management landing zone](#).

## Data landing zone

**Data landing zones** are subscriptions that host multiple analytics and AI solutions relevant to their respective domain or domains. These subscriptions within cloud-scale analytics represent primary business groups, integrators, and enablers. These groups own, operate, and often provide an innate understanding of the source systems.

A few important points to keep in mind about data landing zones:

- Automated ingestion capabilities can exist in each data landing zone. These capabilities allow subject matter experts to pull in external data sources into the data landing zone.
- A data landing zone is instantiated based on its core architecture. It includes key capabilities to host an analytics platform.
- A data landing zone can contain multiple [data products](#).

For more information, see [Data landing zone](#).

## Data products

A data product is anything that drives business value and is pushed to a polyglot store such as the data landing zone data lake.

Data products manage, organize, and make sense of the data within and across domains. A data product is a result of data from one or many transactional system integrations or other data products.

For more information, see [cloud-scale analytics data products in Azure](#).

**ⓘ Important**

When ingesting data from operational systems into a read data source, apart from data quality checks and other applied data, the data should avoid having other data transformations applied to it. This drives reusability of the data product and allows other domains to consume, subject to access, for their use cases as opposed to having multiple extractions from the same operational system.

## Operational excellence

Cloud-scale analytics is designed with operational excellence at its core through self-service enablement, governance, and streamlined deployments. The working model for data operations enables these core principles by using infrastructure-as-code and deployment templates. It also uses deployment processes that include a forking and branching strategy and a central repository.

For more information, see [Organize Operations](#).

## Other design considerations

To get started with the data management and data management landing zones, you need to make sure that you have the underpinning architectural components to enable a successful deployment.

- [Enterprise enrollment and Microsoft Entra tenants for cloud-scale analytics](#)
- [Identity and access management for cloud-scale analytics](#)
- [Network topology and connectivity for cloud-scale analytics landing zones](#)
- [Resource organization for cloud-scale analytics](#)
- [Security, governance, and compliance for cloud-scale analytics](#)
- [Management and monitoring for cloud-scale analytics](#)
- [Business continuity and disaster recovery considerations for AKS](#)

## Next steps

- Enterprise enrollment
- 

## Feedback

Was this page helpful?



# Enterprise enrollment

Article • 11/27/2024

Azure landing zones for cloud-scale analytics don't have any considerations or recommendations that affect enterprise enrollment or Microsoft Entra tenant decisions.

However, it's important to understand the decisions made by the cloud platform team, and to be aware of existing [enterprise enrollment or Microsoft Entra tenant decisions](#).

## Identity and access management

Review [identity and access management considerations](#). These considerations can help you understand how the Microsoft Entra tenant is applied in the design of authentication and authorization solutions. You can also evaluate the [resource organization considerations](#) to understand how the enrollment might be organized into management groups, subscriptions, and resource groups.

## Next steps

- [Identity and access management](#)
- 

## Feedback

Was this page helpful?

 Yes

 No

# Identity and access management

Article • 01/30/2025

This article outlines design considerations and recommendations for identity and access management. It focuses on the deployment of a cloud-scale analytics platform on Microsoft Azure. Because cloud-scale analytics is a mission-critical component, you should follow the guidance about Azure landing zone design areas when you design your solution.

This article builds on considerations and recommendations about Azure landing zones. For more information, see [Identity and access management design area](#).

## Data landing zone design

Cloud-scale analytics supports an access control model by using Microsoft Entra identities. The model uses Azure role-based access control (Azure RBAC) and access control lists.

Review the Azure administration and management activities that your teams perform. Evaluate your cloud-scale analytics on Azure. Determine the best possible distribution of responsibilities within your organization.

## Role assignments

To develop, deliver, and serve data products autonomously within the data platform, data application teams require several access rights within the Azure environment. It's important to note that you should use different access models for development and higher environments. Use security groups when possible to reduce the number of role assignments and to simplify the management and review process of RBAC rights. This step is crucial because of the [limited number of role assignments that you can create for each subscription](#).

The development environment should be accessible to the development team and their respective user identities. This access enables them to iterate more quickly, learn about certain capabilities within Azure services, and troubleshoot problems effectively. Access to a development environment can help you develop or enhance infrastructure as code and other code artifacts.

After you confirm that an implementation works as expected in the development environment, it can be rolled out continuously to higher environments. Higher

environments, such as testing and production, should be locked off for the data application team. Only a service principal should have access to these environments. As such, all deployments must be run through the service principal identity by using continuous integration and continuous delivery (CI/CD) pipelines. In the development environment, provide access rights to both a service principal and user identities. In higher environments, restrict access rights to the service principal identity only.

To create resources and role assignments between resources within the data application resource groups, you must provide `Contributor` and `User Access Administrator` rights. These rights allow the teams to create and control services in their environment within the [boundaries of Azure Policy](#).

To reduce the risk of data exfiltration, it's cloud-analytics best practices to use private endpoints. The Azure platform team blocks other connectivity options via policies, so data application teams need access rights to the shared virtual network of a data landing zone. This access is essential for setting up the necessary network connectivity for the services that they plan to use.

To follow the principle of least privilege, avoid conflicts between different data application teams and have a clear separation of teams. It's cloud-scale analytics best practices to create a dedicated subnet for each data application team and create a `Network Contributor` role assignment for that subnet, or child resource scope. This role assignment allows the teams to join the subnet by using private endpoints.

These first two role assignments enable the self-service deployment of data services within these environments. To address cost management concerns, organizations should add a cost center tag to the resource groups to enable cross-charging and distributed cost ownership. This approach raises awareness within the teams and helps ensure that they make informed decisions about required SKUs and service tiers.

To enable self-service use of other shared resources within the data landing zone, a few extra role assignments are required. If access to an Azure Databricks environment is required, organizations should use [SCIM sync from Microsoft Entra ID](#) to provide access. This synchronization mechanism is important because it automatically synchronizes users and groups from Microsoft Entra ID to the Azure Databricks data plane. It also automatically removes access rights when an individual leaves the organization or business. In Azure Databricks, give the data application teams `Can Restart` access rights to a predefined cluster so that they can run workloads within the workspace.

Individual teams require access to the Microsoft Purview account to discover data assets within their respective data landing zones. Teams often need to edit cataloged data assets that they own to provide extra details like the contact information of data owners

and experts. Teams also require the ability to provide more granular information about what each column in a dataset describes and includes.

## Summary of RBAC requirements

To automate the deployment of data landing zones, the following roles are required:

Role name

Description

Scope

### [Private DNS Zone Contributor](#)

Deploy all private DNS zones for all data services into a single subscription and resource group. The service principal needs to be `Private DNS Zone Contributor` on the global DNS resource group that was created during the data management landing zone deployment. This role is required to deploy A-records for the private endpoints.

(Resource group scope)

```
/subscriptions/{{dataManagement}subscriptionId}/resourceGroups/{resourceGroupName}
```

### [Network Contributor](#)

To set up virtual network peering between the data landing zone network and the data management landing zone network, the service principal needs `Network Contributor` access rights on the resource group of the remote virtual network.

(Resource group scope)

```
/subscriptions/{{dataManagement}subscriptionId}/resourceGroups/{resourceGroupName}
```

### [User Access Administrator](#)

This permission is required to share the self-hosted integration runtime that gets deployed into the `integration-rg` resource group with other data factories. It's also required to assign the Azure Data Factory and Azure Synapse Analytics managed identities access on the respective storage account file systems.

(Resource scope) `/subscriptions/{{dataLandingZone}subscriptionId}`

#### Note

In a production scenario, you can reduce the number of role assignments. The `Network Contributor` role is only required to set up the virtual network peering between the data management landing zone and the data landing zone. Without this role, DNS resolution fails. Also, inbound and outbound traffic is dropped because there's no line of sight to Azure Firewall.

The `Private DNS Zone Contributor` role isn't required if the deployment of DNS A-records for the private endpoints is automated through Azure policies with the `deployIfNotExists` effect. The same is true for the `User Access Administrator` role because you can automate the deployment by using `deployIfNotExists` policies.

## Role assignments for data products

The following role assignments are required to deploy a data product within a data landing zone:

Role name

Description

Scope

### [Private DNS Zone Contributor](#)

Deploy all private DNS zones for all data services into a single subscription and resource group. The service principal needs to be `Private DNS Zone Contributor` on the global DNS resource group that was created during the data management landing zone deployment. This role is required to deploy A-records for the respective private endpoints.

(Resource group scope)

```
/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}
```

### [Contributor](#)

Deploy all data integration streaming services into a single resource group within the data landing zone subscription. The service principal requires a `Contributor` role assignment on that resource group.

(Resource group scope)

```
/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}
```

## Network Contributor

To deploy private endpoints to the specified Azure Private Link subnet that was created during the data landing zone deployment, the service principal requires Network Contributor access on that subnet.

(Child resource scope)

```
/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}  
/providers/Microsoft.Network/virtualNetworks/{virtualNetworkName}/subnets/{subnetNa  
me}"
```

## Access to other resources

Outside of Azure, data application teams require access to a repository to store code artifacts, collaborate effectively, and roll out updates and changes consistently to higher environments via CI/CD. You should provide a project board to allow for agile development, sprint planning, task tracking, and managing user feedback and feature requests.

To automate CI/CD, establish a connection to Azure. This process is done in most services via service principals. Because of this requirement, teams must have access to a service principal to achieve automation in their project.

## Manage access to data

Manage access to data by using Microsoft Entra groups. Add user principal names or service principal names to the Microsoft Entra groups. Then add those groups to the services and grant permissions to the group. This approach allows for fine-grained access control.

For more information about how to drive security for data management landing zones and data landing zones that manage your data estate, see [Authentication for cloud-scale analytics in Azure](#).

## Next step

[Networking overview](#)

# Feedback

Was this page helpful?

 Yes

 No

# Networking overview

Article • 12/10/2024

This article has design considerations and guidelines for networking and connectivity to or from data management landing zones and data landing zones. It builds on information in the [Azure landing zone design area for network topology and connectivity](#) article.

Since data management and data landing zones are important, you should also include the guidance for the Azure landing zone design areas in your design.

This section outlines gives a high level overview of the networking pattern with further links to deploying in both single and multiple Azure regions.

Cloud-scale analytics promises the possibility to easily share and access datasets across multiple data domains and data landing Zones without critical bandwidth or latency limitations and without creating multiple copies of the same dataset. To deliver on that promise, different network designs have to be considered, evaluated, and tested to make sure that these are compatible with the existing hub and spoke and vWAN deployments of corporations.

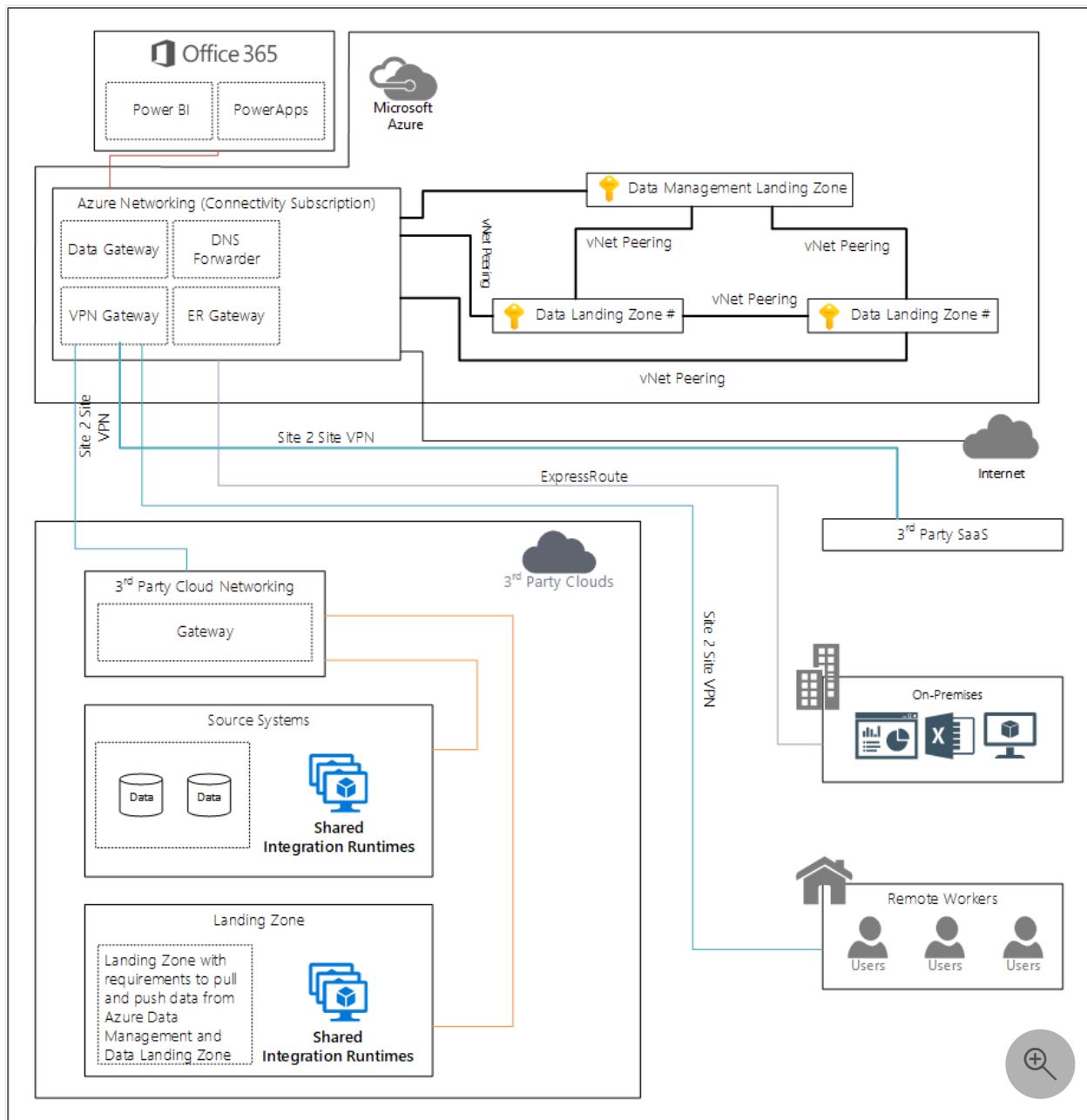


Figure 1: Networking overview for cloud-scale analytics.

### ⓘ Important

This article and other articles in the networking section outline cross-business units that share data. However, this might not be your initial strategy and that you need to start at a base level first.

Design your networking so that you can eventually implement our recommended setup between data landing zones. Ensure you have the data management landing zones directly connected to the landing zones for governance.

## Data management landing zone networking

You can connect virtual networks to each other with virtual network peering. These virtual networks can be in the same or different regions, and are also known as global virtual network peering. After peering the virtual networks, resources in both virtual networks communicate with each other. This communication has the same latency and bandwidth as if the resources were in the same virtual network.

The data management landing zone connects to the Azure networking management subscription using virtual network peering. The virtual network peering then connects to on-premises resources using ExpressRoute circuits and third-party clouds.

Data management landing zone services that support Azure Private Link are injected into the data management landing zone virtual network.

## Data management landing zone to data landing zone

For every new data landing zone, you should create a virtual network peering from the data management landing zone to the data landing zone.

 **Important**

A data management landing zone connects to an data landing zone using virtual network peering.

## Data landing zones to data landing zones

There are options on how to make this connectivity and depending on if you have a single or multiple region deployments it's recommended that you consider the guidance in:

- [Single-region data landing zone Connectivity](#)
- [Cross-region data landing zone connectivity](#)

## Data management landing zone to third-party clouds

To set up connectivity between a data management landing zone and a third-party cloud, use a [Site-to-Site VPN](#) gateway connection. This VPN can connect your on-

premises or third-party cloud landing zone to an Azure virtual network. This connection is created over an IPsec or internet key exchange v1 or v2 (IKEv1 or IKEv2) VPN tunnel.

Site-to-Site VPNs can provide better continuity for your workloads in a hybrid cloud setup with Azure.

### ⓘ Important

For connections to a third-party cloud, we recommend implementing a Site-to-Site VPN between your Azure connectivity subscription and the third-party cloud connectivity subscription.

## Private endpoints

Cloud-scale analytics uses [Private Link](#), where available, for shared platform as a service (PaaS) functionality. Private Link is available for several services and is in public preview for more services. Private Link addresses data exfiltration concerns related to service endpoints.

For the current list of supported products, see [Private Link resources](#).

If you're planning on implementing cross tenant private endpoints, it's recommended that you review [Limit cross-tenant private endpoint connections in Azure](#).

### ✖ Caution

By design, cloud-scale analytics networking uses private endpoints where available to connect to PaaS services.

## Implement Azure DNS resolver for private endpoints

Handle DNS resolution for private endpoints through central [Azure Private DNS](#) zones. Required DNS records for private endpoints can be automatically created using Azure Policy to allow access through fully qualified domain names (FQDNs). The lifecycle of the DNS records follows the lifecycle of the private endpoints. It's automatically removed when the private endpoint is deleted.

## Next steps

- [Single-region Data Landing Zone Connectivity](#)

---

# Feedback

Was this page helpful?

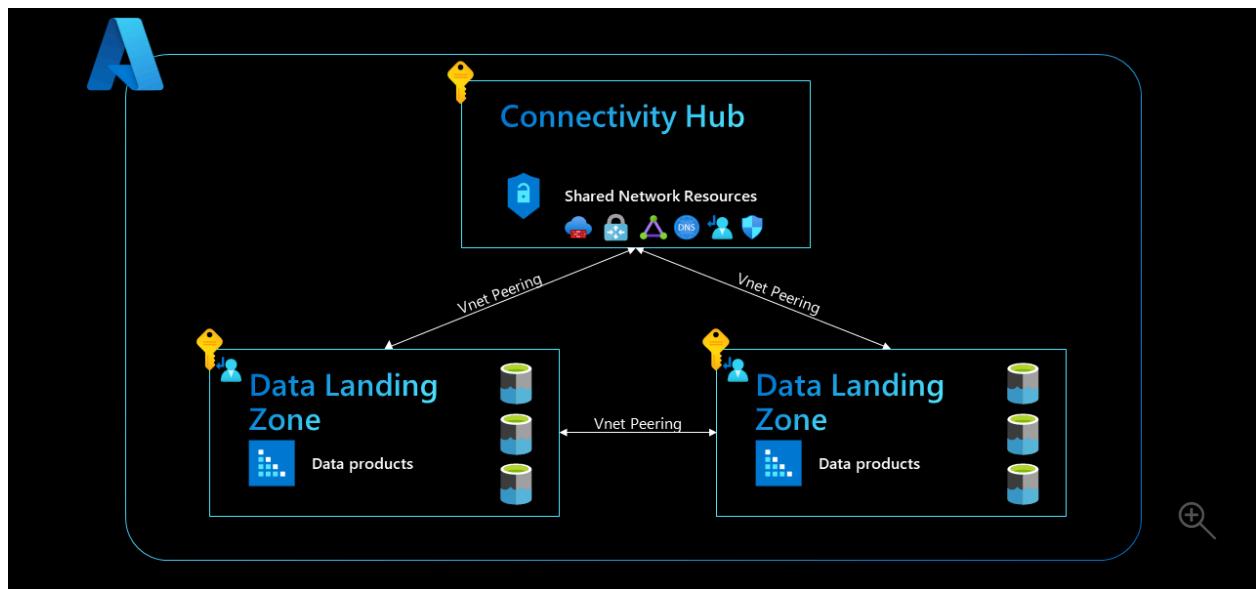
 Yes

 No

# Single-region data landing zone connectivity

Article • 02/24/2025

In a single-region setup, the data management landing zone, data landing zones, and all related services are established in the same region. Also, all landing zones reside in the same connectivity hub subscription. This subscription hosts shared network resources, such as a network virtual appliance (NVA) like Azure Firewall, an Azure ExpressRoute gateway, a virtual private network gateway, a hub virtual network, or an Azure Virtual WAN hub.



Based on the capabilities of Azure networking services, we recommend that you use a meshed network architecture. Establish virtual network peering between:

- The connectivity hub and the data management zone.
- The connectivity hub and each data landing zone.
- The data management zone and each data landing zone.
- Each data landing zone.

This article describes the pros and cons of each network architecture option for cloud-scale analytics.

The first section of this article focuses on a single-region pattern, where the data management zone and all data landing zones are hosted in the same region.

Each design pattern is evaluated by using the following criteria:

- Costs
- User access management

- Service management
- Bandwidth
- Latency

Each design option is analyzed with the following cross-data landing zone use case in mind.

### Note

Virtual machine (VM) B that's hosted in data landing zone B loads a dataset from storage account A that's hosted in data landing zone A. Then VM B processes that dataset and stores it in storage account B, which is hosted in data landing zone B.

### Important

This article and other articles in the networking section outline cross-business units that share data. However, this might not be your initial strategy, and you might need to start at a more foundational level first.

Design your networking so that you can eventually implement our recommended setup between data landing zones. Ensure that you have the data management landing zones directly connected to the landing zones for governance.

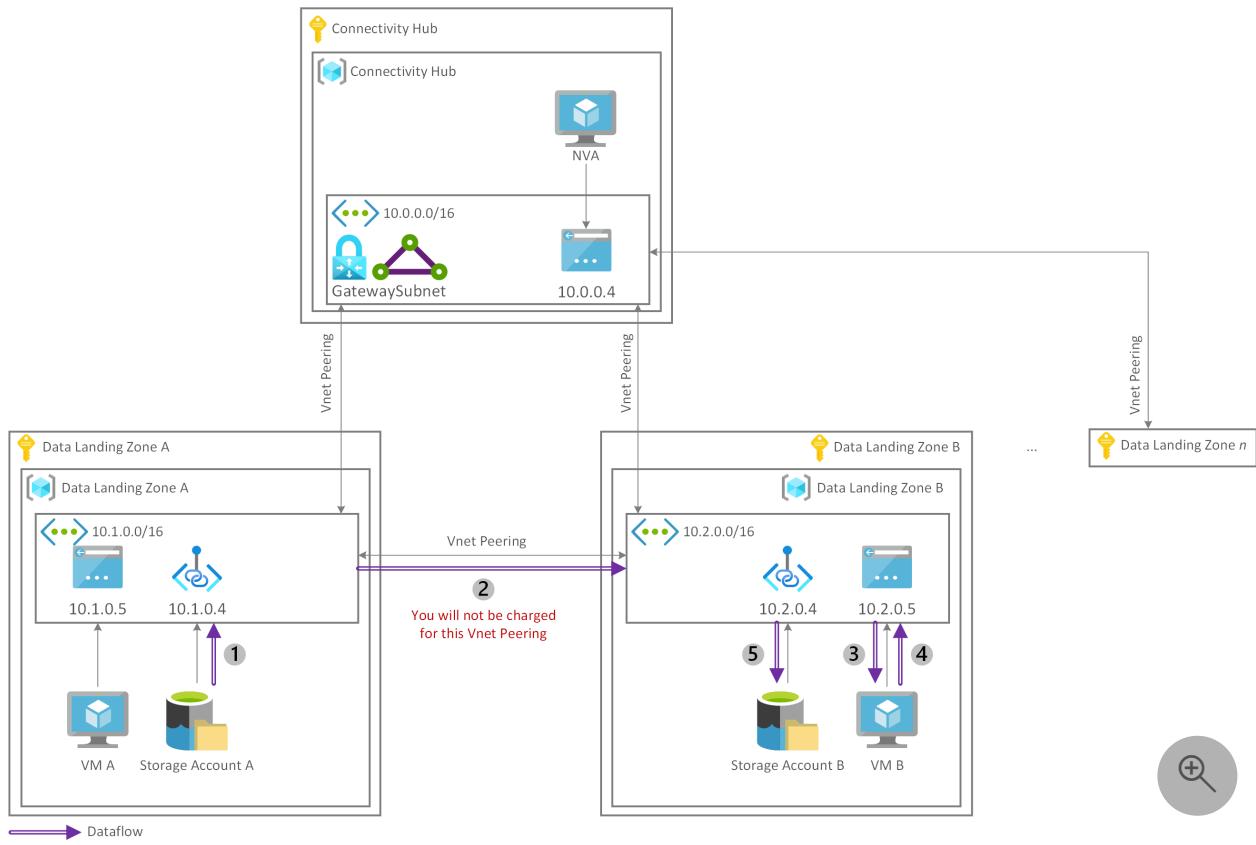
## Meshed network architecture (Recommended)

Use a network mesh architecture when you adopt cloud-scale analytics. To implement a network mesh architecture, in addition to the existing hub-and-spoke network design setup in your tenant, you need to do two things:

- Add virtual network peerings between all data landing zone virtual networks.
- Add virtual network peerings between your data management landing zone and all data landing zones.

In the following architecture, data loaded from storage account A transits a virtual network peering connection (2) set up between the two data landing zone virtual networks. VM B (3) and (4) loads and processes the data, then sends it through the local private endpoint (5) to be stored in storage account B.

In this scenario, the data doesn't pass through the connectivity hub. It stays in the data platform that consists of a data management landing zone and one or more data landing zones.



## User access management in a meshed network architecture

In a meshed network architecture design, data application teams need only two things to create new services, including private endpoints:

- Write access to their dedicated resource group in the data landing zone
- Join access to their designated subnet

In this design, data application teams can deploy private endpoints themselves. If they have the necessary access rights to connect private endpoints to a subnet in a given spoke, they can set up the necessary connectivity without extra support.

Summary: **+** **+** **+**

The plus icon indicates the pros and cons of each scenario.

## Service management in a meshed network architecture

A meshed network architecture design doesn't have a NVA that might serve as a single point of failure or limit throughput. Datasets aren't sent through the connectivity hub, so your central Azure platform team has less overhead, as long as you don't need to scale out that NVA.

This design implies that the central Azure platform team can no longer inspect and log all traffic that's sent between data landing zones. However, cloud-scale analytics is a coherent platform that spans multiple subscriptions. This capability allows for scale and overcomes platform-level limitations.

With all resources hosted in a single subscription, your central Azure platform team no longer inspects all data in the central connectivity hub, either. You can still capture network logs by using network security group (NSG) flow logs. You can consolidate and store other application-level and service-level logs by using service-specific diagnostic settings.

You can capture all these logs at scale by using [Azure policy definitions for diagnostic settings](#).

This design also allows you to create an Azure-native Domain Name System (DNS) solution that's based on private DNS zones. You can automate the DNS A-record lifecycle via [Azure policy definitions for private DNS groups](#).

Summary: 

## Meshed network architecture cost

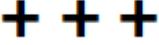
### Note

When you [access a private endpoint across a peered network](#), you're only charged for the private endpoint itself and not for the virtual network peering.

In this network design, you only pay for:

- Your private endpoints, per hour.
- The ingress and egress traffic sent through your private endpoints to load your raw dataset (1) and store your processed dataset (6).

Virtual network peering isn't charged (2), which is why this option has minimal cost.

Summary: 

## Bandwidth and latency in a meshed network architecture

This design has no known bandwidth or latency limitations because no NVAs limit throughput for its cross-data landing zone data exchange. The design is limited only by

the speed of fiber-optic cables in the datacenters.

Summary: **+++**

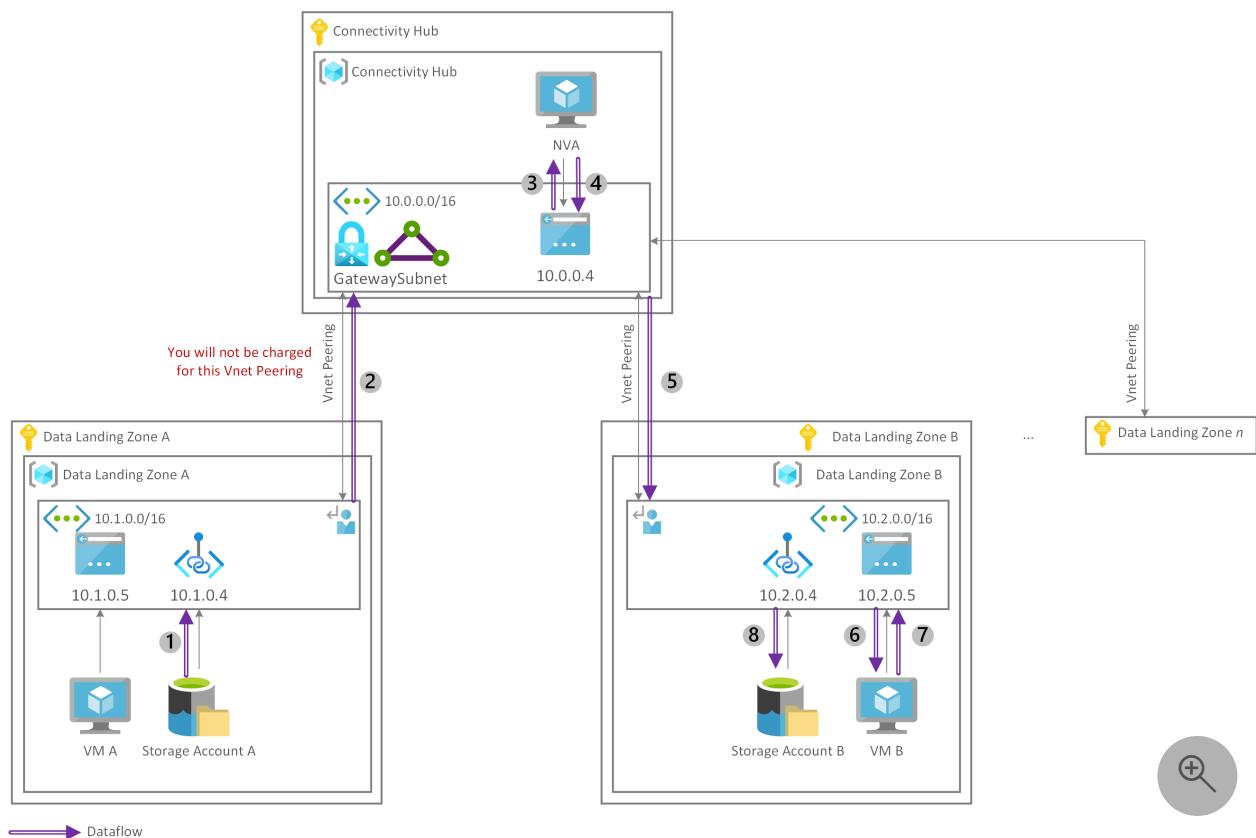
## Meshed network architecture summary

If you plan to adopt cloud-scale analytics, we recommend that you use the meshed network design. A meshed network provides maximum bandwidth and low latency at minimal cost without compromising user access management or the DNS layer.

If you need to enforce other network policies in the data platform, use NSGs instead of central NVAs.

## Traditional hub-and-spoke architecture (Not recommended)

The hub-and-spoke network architecture design is a common choice that many enterprises use. In this design, network transitivity is established in the connectivity hub to access data in storage account A from VM B. Data traverses two virtual network peerings (2) and (5) and an NVA that's hosted inside the connectivity hub (3) and (4). The VM (6) loads the data and stores it back into storage account B (8).

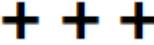


# User access management in a traditional hub-and-spoke architecture

In a traditional hub-and-spoke design, data application teams need only two things to create new services, including private endpoints:

- Write access to their resource group in the data landing zone
- Join access to their designated subnet

In this design, data application teams can deploy private endpoints themselves. If they have the necessary access rights to connect private endpoints to a subnet in a given spoke, they can set up the necessary connectivity without extra support.

Summary: 

# Service management in a traditional hub-and-spoke architecture

This network design is well-known and consistent with existing network setups for most organizations. This familiarity makes the design easy to explain and implement. You can also use a centralized Azure-native DNS solution with private DNS zones to provide FQDN resolution inside your Azure tenant. You can use private DNS zones to automate the DNS A-record lifecycle via [Azure policies](#).

Another benefit of this design is that traffic is routed through a central NVA, so network traffic sent from one spoke to another spoke can be logged and inspected.

One downside of this design is that your central Azure platform team has to manually manage route tables. This requirement helps ensure transitivity between spokes, which enables data asset sharing across multiple data landing zones. Route management can become complex and error-prone over time, so you must consider it from the start.

Another downside to this network setup is how your central NVA is set up. The NVA functions as a single point of failure and can cause serious downtime inside the data platform if a failure occurs. As dataset sizes increase in a data platform and the number of cross-data landing zone use cases increases, more traffic is sent through the central NVA.

Over time, this situation can result in gigabytes or even terabytes of data being sent through the central instance. Because the bandwidth of existing NVAs is often limited to just one-digit or two-digit gigabyte throughput, the central NVA can become a bottleneck that severely limits traffic flow between data landing zones and limits data asset shareability.

The only way to avoid this problem is to scale out your central NVA across multiple instances, which has major cost implications for this design.

Summary: -

## Traditional hub-and-spoke architecture cost

### ⓘ Note

When you [access a private endpoint across a peered network](#), you're only charged for the private endpoint itself and not for the virtual network peering.

For this network, you're charged per hour for the private endpoints of your storage accounts. You're also charged for ingress and egress traffic sent through the private endpoints to load a raw dataset (1) and store the processed dataset (8).

Your customer is charged for the ingress and egress of one virtual network peering (5). The first virtual network peering isn't charged (2).

This network design (3) and (4) results in a high central NVA cost. You have to either purchase extra licenses and scale out the central NVA based on demand or pay the charge processed per gigabyte, similar to Azure Firewall charges.

Summary: - - -

## Bandwidth and latency in a traditional hub-and-spoke architecture

This network design has serious bandwidth limitations. The central NVA becomes a critical bottleneck as your platform grows, which limits cross-data landing zone use cases and dataset sharing. It can also create multiple copies of datasets over time.

This design also heavily affects latency, which becomes especially critical in real-time analytics scenarios.

Summary: - - -

## Traditional hub-and-spoke architecture summary

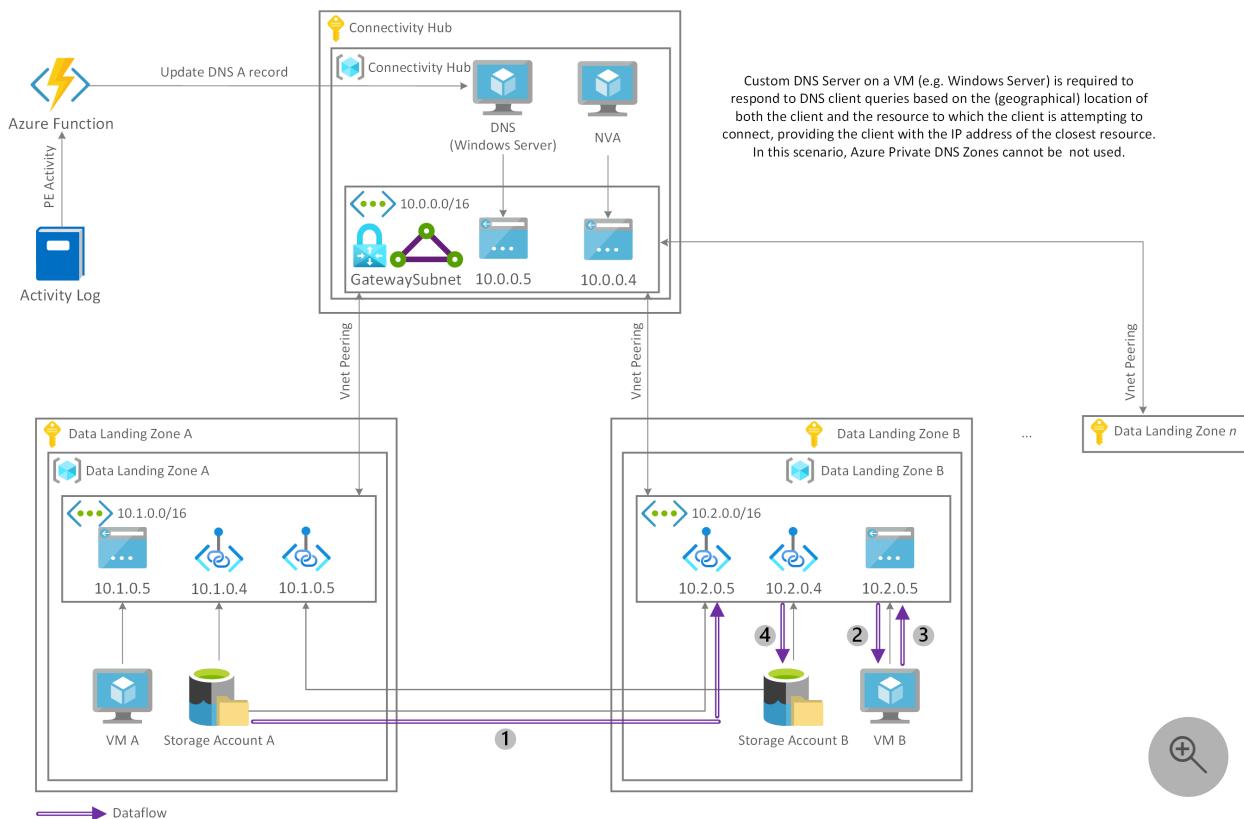
This hub-and-spoke network design has access management and some service management benefits. But because of critical limitations of service management and

bandwidth and latency, we can't recommend this network design for cross-data landing zone use cases.

## Private endpoint projection architecture (Not recommended)

Another design option is the projection of private endpoints across all landing zones. In this design, a private endpoint for storage account A is created in each landing zone. So the first private endpoint in data landing zone A connects to the virtual network in data landing zone A and the second private endpoint connects to the virtual network in data landing zone B. This configuration repeats for each landing zone so that they all have their own dedicated connection.

The same setup applies to storage account B, and potentially to other services inside the data landing zones. If you define the number of data landing zones as  $n$ , then you have  $n$  private endpoints for at least all the storage accounts and potentially other services in the data landing zones. This situation results in an exponential increase in the number of private endpoints.



Because all private endpoints of a specific service (like storage account A) have the same FQDN (like `storageaccounta.privatelink.blob.core.windows.net`), this solution creates challenges in the DNS layer that private DNS zones can't solve. You need a custom DNS solution that can resolve DNS names based on the origin or IP address of the requestor. This setup allows you to make VM A connect to the private endpoints connected to the

virtual network in data landing zone A, and to make VM B connect to the private endpoints that are connected to the virtual network in data landing zone B. You can achieve this connection by using a Windows Server-based setup. In contrast, you can automate the DNS A-records lifecycle through a combination of the Azure Monitor activity log and Azure Functions.

In this setup, you can load the raw dataset in storage account A into VM B by accessing the dataset through the local private endpoint (1). After you load and process the dataset (2) and (3), you can store it in storage account B by directly accessing the local private endpoint (4). In this scenario, data must not traverse any virtual network peerings.

## User access management in the private endpoint projection architecture

This design's approach to user access management is similar to that of the [meshed network architecture](#). In this design, you can require access rights for other data landing zones to create private endpoints, not only in a designated data landing zone and virtual network, but also in other data landing zones and their respective virtual networks. Because of this approach, your data application teams require three things to create new services themselves:

- Write access to a resource group in a designated data landing zone
- Join access to their designated subnet
- Access to a resource group and subnet inside all the other data landing zones to create their respective local private endpoints

This network design increases complexity in your access management layer because your data application teams require permissions in all data landing zones. The design can also be confusing and result in inconsistent role-based access control over time.

If data landing zone teams and data application teams aren't given necessary access rights, problems can occur, similar to the problems in the [traditional hub-and-spoke architecture](#).

Summary: ▶

## Service management in the private endpoint projection architecture

This network design is similar to the [meshed network architecture](#) design, but it has the advantage of not relying on an NVA, which can become a single point of failure or limit throughput.

It also reduces management overhead for your central Azure platform team by not sending datasets through the connectivity hub because there's no need to scale out the NVA. This change implies that the central Azure platform team can no longer inspect and log all traffic sent between data landing zones. However, cloud-scale analytics is a coherent platform that spans multiple subscriptions. This capability allows for scale and overcomes platform-level limitations.

With all resources hosted in a single subscription, traffic isn't inspected in the central connectivity hub. You can still capture network logs by using NSG flow logs, and you can consolidate and store other application and service-level logs by using service-specific diagnostic settings. You can capture all these logs at scale by using Azure policies. However, the network address space that your data platform requires increases because of the exponential increase in required private endpoints, which isn't optimal.

The major concerns about this network architecture are its DNS challenges. You can't use an Azure-native solution in the form of private DNS zones, so this architecture requires a non-Microsoft solution that can resolve FQDNs based on the origin or IP address of the requestor. You also have to develop and maintain tools and workflows to automate private DNS A-records, which significantly increase management overhead compared to the [Azure policy-driven solution](#).

You can create a distributed DNS infrastructure by using private DNS zones, but this infrastructure creates DNS islands. DNS islands can cause problems when you try to access private link services hosted in other landing zones in your tenant. Therefore, this design isn't a viable option.

Summary: - - -

## Private endpoint projection architecture cost

### Note

When you [access a private endpoint across a peered network](#), you're only charged for the private endpoint itself and not for the virtual network peering.

In this network design, you're only charged for the private endpoints per hour and the ingress and egress traffic sent through those private endpoints to load raw datasets (1) and store processed datasets (4). However, you can expect extra costs because of the

exponential increase in the number of your data platform's private endpoints. Because you're charged per hour, the amount of extra cost highly depends on how many private endpoints are created.

Summary: 

## Bandwidth and latency in the private endpoint projection architecture

This design has no known bandwidth and latency limitations because it doesn't have NVAs to limit throughput for cross-data landing zone data exchange. The design is limited only by the speed of fiber-optic cables in the datacenters.

Summary:   

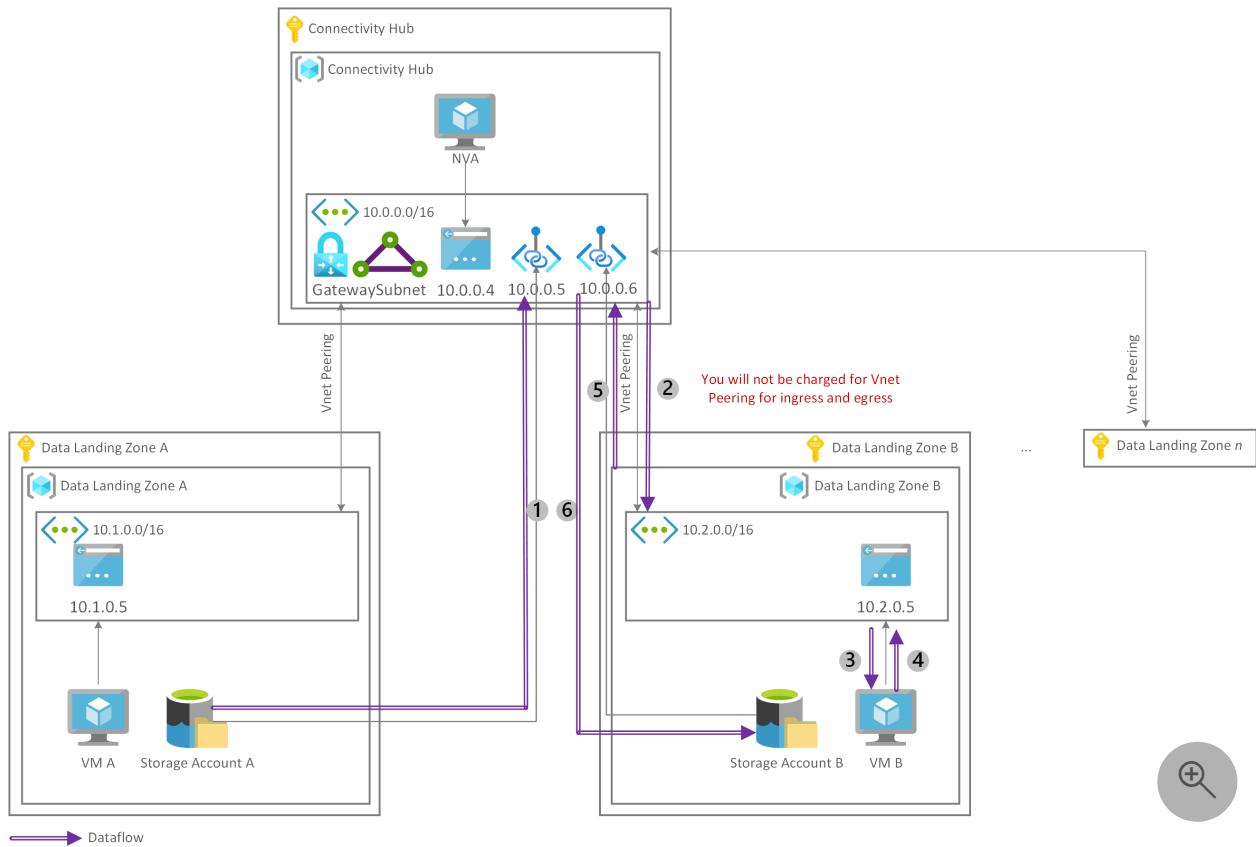
## Private endpoint projection architecture summary

The exponential growth of private endpoints in this network architecture can cause you to lose track of which private endpoints are used for what purpose and in which location. You're also limited by access management problems and DNS layer complexities. Because of these problems, we don't recommend this network design for cross-data landing zone use cases.

## Private endpoints in connectivity hub architecture (Not recommended)

Another network option is to host private endpoints in your connectivity hub and connect them to the hub virtual network. In this solution, you host a single private endpoint for each service on your corporate virtual network. Because of the existing hub-and-spoke network architecture at most corporations, and the fact that your connectivity hub hosts your private endpoints in this solution, transitivity isn't required. Virtual network peering between your connectivity hub and data landing zones enables direct access.

Data traverses a single virtual network peering between the connectivity hub and data landing zone to load a dataset that's stored in storage account A in VM B. After that dataset is loaded and processed (3) and (4), it traverses the same virtual network peering a second time (5) before finally getting stored in storage account B through the private endpoint that's connected to the hub virtual network (6).



## User access management in the connectivity hub architecture

In this network design, your data landing zone teams and data application teams need two things to be able to connect private endpoints to the hub virtual network:

- Write permissions to a resource group in your connectivity hub subscription
- Join permissions to the hub virtual network

Your connectivity hub is designated for your organization's Azure platform team and is dedicated to hosting your organization's necessary and shared network infrastructure. This infrastructure includes firewalls, gateways, and network management tools. This network option makes that design inconsistent because it doesn't follow access management principles from the enterprise-scale landing zone base principles. Therefore, most Azure platform teams are unlikely to approve this design option.

Summary: - - -

## Service management in the connectivity hub architecture

This design is similar to the [meshed network architecture](#) but has no NVA that could serve as a single point of failure or limit throughput. It also reduces management overhead for your central Azure platform team by not sending datasets through the

connectivity hub because there's no need to scale out the virtual appliance. This design implies that the central Azure platform team can no longer inspect and log all traffic sent between data landing zones. However, cloud-scale analytics is a coherent platform that spans multiple subscriptions, which allows for scale and overcomes platform-level limitations.

With all resources hosted in a single subscription, traffic isn't inspected in the central connectivity hub. You can still capture network logs by using NSG flow logs, and you can consolidate and store other application and service-level logs by using service-specific diagnostic settings. You can capture all these logs at scale by using Azure policies.

This design also allows you to create an Azure-native DNS solution that's based on private DNS zones and automate the DNS A-record lifecycle through [Azure policies](#).

Summary: 

## Connectivity hub architecture cost

### Note

When you [access a private endpoint across a peered network](#), you're only charged for the private endpoint itself and not for the virtual network peering.

In this network design, you're only charged for your private endpoints per hour and ingress and egress traffic sent through those private endpoints to load a raw dataset (1) and store the processed dataset (6).

Summary: 

## Bandwidth and latency in the connectivity hub architecture

This design has no known bandwidth and latency limitations because it doesn't have NVAs to limit throughput for cross-data landing zone data exchange. The design is limited only by the speed of fiber-optic cables in the datacenters.

Summary: 

## Private endpoints in the connectivity hub architecture summary

This network architecture design has multiple benefits, but its access management inconsistencies make it subpar. So we don't recommend this design choice.

## Single-region data landing zone connectivity conclusion

Out of all the reviewed network architecture options and their advantages and disadvantages, the [meshed network architecture](#) is the most suitable. It provides significant benefits for throughput, cost, and management. These benefits make it the preferred choice for deploying cloud-scale analytics. Peering spoke virtual networks isn't common, which results in problems with sharing datasets across domains and business units.

Cloud-scale analytics is a coherent solution that spans multiple subscriptions. In a single subscription setup, network traffic flow equals the flow in the meshed network architecture. Users are likely to reach the platform's [subscription level limits and quotas](#), which cloud-scale analytics tries to prevent.

## Next step

[Cross-region data landing zone connectivity](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Cross-region data landing zone connectivity

Article • 12/10/2024

If you have a presence in more than one Azure region and need to host your data platform and data applications across multiple geographies, connectivity becomes slightly more complicated.

Multi-region deployments generally have a connectivity hub subscription in each individual Azure location. For instance, if you have services running in both East US and West Europe, you set up a connectivity hub subscription with shared network resources in each region. Shared network resources include:

- Network virtual appliances (like Azure Firewall)
- ExpressRoute Gateways
- VPN Gateways
- Hub Virtual Networks (in a hub and spoke architecture) or vWAN Hubs (in a vWan setup)

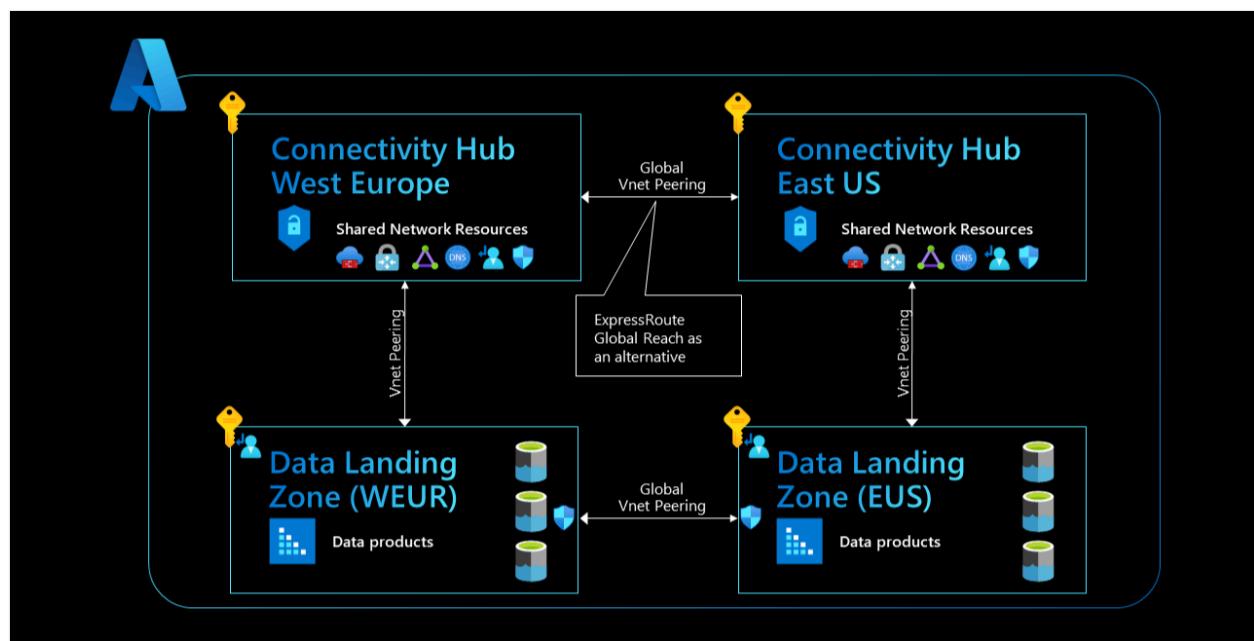


Figure 1: Cross-region Connectivity.

In hub-spoke-hub architecture, connectivity hubs' virtual networks are often connected using Global virtual network Peering. For larger environments, a common alternative is to use ExpressRoute Global Reach. Whichever connectivity option you choose, you can achieve global routing and connectivity between spoke networks across multiple geographies. This means you can move data across regions using network virtual appliances, network security groups, and Route Tables, given that your traffic doesn't get blocked in either connectivity subscription.

## **Important**

This article and other articles in the networking section outline cross-business units that share data. However, this might not be your initial strategy and that you need to start at a base level first.

Design your networking so that you can eventually implement our recommended setup between data landing zones. Ensure you have the data management landing zones directly connected to the landing zones for governance.

## **Global virtual network Peering (recommended)**

You can connect data landing zones across regions using direct Global virtual network Peering. In this setup, if we continue our previous example scenario, the virtual machine in West Europe accesses the East US storage account's private endpoint directly, without relying on any hub and spoke or vWAN network architectures. Data is directly loaded by the virtual machine over a private endpoint, processed, and then stored back on the storage account in West Europe.

### **User access management in Global virtual network Peering**

There are no particular pros or cons for either of the proposed cross-region data landing zone connectivity options.

Summary:  / 

### **Service management in Global virtual network Peering**

Global virtual network Peering has no network virtual appliance that acts as a single point of failure or throttling throughput. Data isn't sent through your connectivity hubs, so you don't need to scale the virtual appliances and gateways within the connectivity hubs. This lack of scaling reduces management overhead for your core Azure platform team. You also don't need to allowlist individual cross-region connections. Your data teams can access data from data landing zones in other regions without having to wait for route table changes.

In this network design, your central Azure platform team can no longer inspect and log all traffic using a layer 7 firewall. However, the cloud-scale analytics scenario is a coherent platform spanning multiple subscriptions, which allows for scale and

overcomes platform-level limitations, so that isn't a disadvantage. You can capture network logs by using Network Security Group Flow Logs. You can consolidate and store other application and service level logs by using service-specific Diagnostic Settings.

You can capture all of these logs at scale by using [Azure Policy definitions for diagnostic settings](#).

In some scenarios, you need to limit due to regulatory or legal implications. For instance, you might have a local regulation that requires certain datasets to stay within a particulate datacenter, so you're not allowed to transfer them across regions. You can rely on network security groups to help you comply with this kind of rule, only allowing traffic to move in one direction from East US to West Europe and not vice versa. Within your network security groups, you can ensure that traffic originating from East US is denied while traffic originating from West Europe is allowed.

This solution approach doesn't impact bandwidth and latency, and allows customers to remain compliant while still combining datasets from multiple regions. This option also has no impact on your DNS architecture and allows you to use an Azure-native solution based on Azure Private DNS Zones.

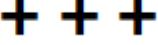
Summary: 

## Global virtual network Peering cost

### Note

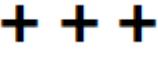
When accessing a private endpoint across a peered network, you will only ever be charged for the private endpoint itself and not for the VNet peering. You can read the official statement in [FAQ: How will billing work when accessing a private endpoint from a peered network?](#).

With this network design, you're charged for your Private Endpoints (per hour) and all ingress and egress traffic sent through them. You also have to pay a [data transfer cost](#) for traffic between regions. However, you won't be charged any Global virtual network Peering ingress and egress costs and it has noteworthy cost benefits compared to the [traditional spoke-hub-hub-spoke option](#).

Summary: 

## Bandwidth and latency in Global virtual network Peering

Impact on bandwidth and latency is lower in Global virtual network Peering than in the traditional spoke-hub-hub-spoke option. Global virtual network Peering contains a lower number of hops for cross-region data landing zone data exchange and has no network virtual appliances limiting throughput. The only things dictating the bandwidth and latency you can achieve for cross-region traffic are the physical limits of our datacenters (speed of fiber-optic cables, gateways, and routers).

Summary: 

## Global virtual network Peering summary

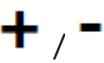
Global virtual network Peering between data landing zones in different regions offers tremendous benefits, especially as cross-region data traffic increases within your data platform. It simplifies service management for your core Azure platform team, and especially benefits use cases that require low latency and high bandwidth. It also offers significant cost benefits over the traditional spoke-hub-hub-spoke design option.

## Traditional spoke-hub-hub-spoke design (not recommended)

Your other option for cross-region data transfers is the traditional spoke-hub-hub-spoke design. In our example scenario, if a virtual machine in data landing zone A hosted in West Europe loads a dataset stored in a storage account from data landing zone B hosted in East US, data traverses two local virtual network peerings (connectivity between hub and spokes), one Global virtual network Peering (connectivity between hubs) and two Gateways or network virtual appliances before loaded by the virtual machine and then moved back into the local storage account.

## User Access Management in traditional spoke-hub-hub-spoke design

There are no particular pros or cons for either of the proposed cross-region data landing zone connectivity options.

Summary: 

## Service management in traditional spoke-hub-hub-spoke design

This solution approach is well-known and consistent with other cross-region connectivity patterns, which makes it easy to adopt and implement. It also has no impact on DNS architecture and allows you to use an Azure-native solution based on Azure Private DNS Zones.

While this connectivity option works seamlessly if you set it up correctly, it does have downsides. Cross-region traffic is often denied by default and has to be enabled on a case-by-case basis. A ticket has to be submitted to your core Azure platform team for every single required cross-region data access requirement so your team can allowlist each specific connection between a virtual machine and cross-region storage account. This process significantly increases management overhead. It also slows down your data project teams, because they can't access the data they need.

You should also note that in this option, connectivity hubs act as single points of failure. In network virtual appliance or Gateway downtime, connectivity and corresponding data platforms fail. You also have a high risk of misconfiguring routes in the connectivity hubs. This misconfiguration can cause more serious downtime in your data platform and lead to a series of dependent workflow and data product failures.

You should monitor the amount of data you need to transfer across regions while using this solution approach. Over time, this monitoring can involve gigabytes or terabytes of data moving through your central instances. Since the bandwidth of network virtual appliances is often limited to a one- or two-digit gigabyte throughput, the appliances can act as a critical bottleneck limiting the traffic flow between regions and the shareability of your data assets. Because of this bottleneck, your shared network resources can require scaling mechanisms, which are often time consuming and costly, and can impact other workloads in your tenant.

Summary: ▾

## Traditional Spoke-Hub-Hub-Spoke design cost

### ⓘ Note

When accessing a private endpoint across a peered network you will only ever be charged for the private endpoint itself and not for the VNet peering. You can read the official statement in [FAQ: How will billing work when accessing a private endpoint from a peered network? ↴](#).

In the traditional spoke-hub-hub-spoke design, you're charged for two storage accounts' Private Endpoints (per hour) and all ingress and egress traffic sent through

them. You're also charged for the ingress and egress traffic of one local virtual network peering and the global virtual network peering between your connectivity hubs. However, you aren't charged for the first virtual network peering, as we explained in the previous note.

Your central network virtual appliances also incur significant costs if you choose this network design. This cost is because you either have to purchase extra licenses to scale the appliances based on demand or pay the charge per processed gigabyte, as with Azure Firewall.

Summary: - - -

## Bandwidth and latency in traditional spoke-hub-hub-spoke design

This network design has serious bandwidth limitations. Your central network virtual appliances become critical bottlenecks as your platform grows, which limits cross-region data landing zone use cases and sharing of your datasets. It also makes it likely that multiple copies of your datasets get created over time. This design also heavily affects latency, which is especially critical for real-time analytics scenarios, since your data traverses many hops.

Summary: - - -

## Design of traditional spoke-hub-hub-spoke

The spoke-hub-hub-spoke design is well-known and established at many organizations, which makes it easy to establish in an existing environment. However, it has significant downsides for service management, cost, bandwidth, and latency. These issues are especially noticeable as your number of cross-region use cases grows.

## Conclusion

[Global virtual network Peering](#) has many advantages over the [traditional spoke-hub-hub-spoke design](#), as it's cost effective, easily managed, and offers reliable connectivity across regions. While traditional spoke-hub-hub-spoke design can be a viable option while your data volume and need for cross-region data exchange is low, we recommend you use the Global virtual network Peering approach as the amount of data you need to exchange across regions grows.

# Next steps

- Limit cross-tenant private endpoint connections in Azure
  - Resource organization for cloud-scale analytics
- 

## Feedback

Was this page helpful?

 Yes

 No

# Limit cross-tenant private endpoint connections in Azure

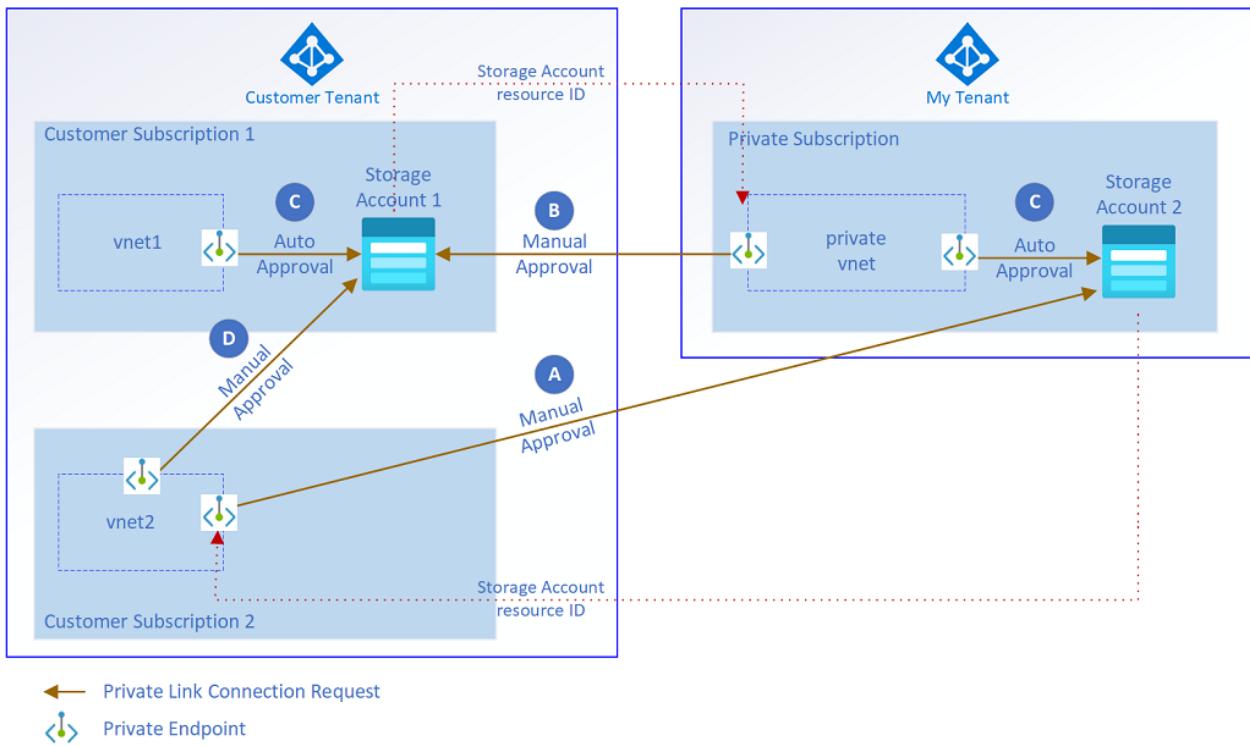
Article • 10/09/2023

Customers are increasingly using private endpoints in their tenants to connect to their Azure platform as a service (PaaS) services privately and securely. Private endpoints can connect to services across Microsoft Entra tenants. For security and compliance, you might need to block cross Microsoft Entra tenants connections on your private endpoints. This guidance shows you recommended configuration options to limit or prevent cross-tenant private endpoint connections. These options help you create data leakage prevention (DLP) controls inside your Azure environment.

## Introduction to private endpoints

Use private endpoints to control the traffic within your Azure environment using an existing network perimeter. But there are scenarios where you must keep private endpoint connections within the corporate Microsoft Entra tenant only. The following examples show connections that might create security risks.

- **Connection A:** A rogue administrator creates private endpoints on the customer virtual network. These endpoints link to services that are hosted outside the customer environment, like another Microsoft Entra tenant.
- **Connection B:** A rogue administrator creates private endpoints in other Microsoft Entra tenants that link to services hosted in the customer's Microsoft Entra tenant.



*Figure 1: Illustration of private endpoint cross-tenant scenarios.*

For both scenarios, you specify the resource ID of the service and manually approve the private endpoint connection. Users also require role-based access control (RBAC) access to run these actions.

Connections C and D in Figure 1 show scenarios that customers generally want to allow. The private endpoint connections are kept within the corporate Microsoft Entra tenant. They don't represent a security risk so these two scenarios aren't covered in this article.

The following information gives you options to prevent private endpoints provisioning across Microsoft Entra tenants.

## Deny private endpoints linked to services in other tenants

**Scenario one:** A rogue administrator requires the following rights in a subscription in the customer's Microsoft Entra tenant.

- **Microsoft.Network/virtualNetworks/join/action** rights on a subnet with **privateEndpointNetworkPolicies** set to **Disabled**.
- **Microsoft.Network/privateEndpoints/write** access to a resource group in the customer environment.

With these rights, a rogue administrator can create a private endpoint in the customer's Microsoft Entra tenant. This private endpoint links to a service in a separate subscription

and Microsoft Entra tenant. Figure 1 shows this scenario as connection A.

For this scenario, the user sets up an external Microsoft Entra tenant and Azure subscription. Next, they create a private endpoint in the customer environment by manually specifying the resource ID of the service. Finally, the rogue administrator approves the private endpoint on the linked service that's hosted in the external Microsoft Entra tenant to allow traffic over the connection.

After the rogue administrator approves the private endpoint connection, corporate data can be copied from the corporate virtual network to an Azure service on an external Microsoft Entra tenant. This security risk can only occur if access was granted using Azure RBAC.

## Mitigation for scenario one

Use the following [Azure Policy](#) to automatically block the ability to create a private endpoint in the corporate Microsoft Entra tenant that's linked to an outside Azure service.

JSON

```
"if": {
  "allof": [
    {
      "field": "type",
      "equals": "Microsoft.Network/privateEndpoints"
    },
    {
      "anyOf": [
        {
          "count": {
            "field":
"Microsoft.Network/privateEndpoints/manualprivateLinkServiceConnections[*]",
            "where": {
              "allof": [
                {
                  "field":
"Microsoft.Network/privateEndpoints/manualprivateLinkServiceConnections[*].p
rivateLinkId",
                  "notEquals": ""
                },
                {
                  "value": "
[split(concat(first(field('Microsoft.Network/privateEndpoints/manualprivateL
inkServiceConnections[*].privateLinkId')),'//'), '/')][2]]",
                  "notEquals": "
[subscription().subscriptionId]"
                }
              ]
            }
          }
        ]
      }
    }
  ]
}
```

```

        }
    },
    "greaterOrEquals": 1
},
{
    "count": {
        "field":
"Microsoft.Network/privateEndpoints/privateLinkServiceConnections[*]",
        "where": {
            "allOf": [
                {
                    "field":
"Microsoft.Network/privateEndpoints/privateLinkServiceConnections[*].private
LinkServiceId",
                    "notEquals": ""
                },
                {
                    "value": "
[split(concat(first(field('Microsoft.Network/privateEndpoints/privateLinkSer
viceConnections[*].privateLinkId')), '//'), '/')][2]]",
                    "notEquals": "
[subscription().subscriptionId]"
                }
            ]
        }
    },
    "greaterOrEquals": 1
}
]
}
],
{
    "then": {
        "effect": "Deny"
}

```

This policy denies any private endpoints created outside of the subscription of the linked service, like connections A and D. The policy also provides the flexibility to use `manualPrivateLinkServiceConnections` and `privateLinkServiceConnections`.

You can update this policy so private endpoints are only created in a certain set of subscriptions. You can make this change by adding a `list` parameter and use the `"notIn": "[parameters('allowedSubscriptions')]"` construct. But this approach isn't recommended, because it means that you'd have to constantly maintain the list of subscriptions for this policy. Whenever a new subscription is created inside your tenant, the subscription ID must be added to the parameter.

Instead, assign the policy to the top-level management group, and then use exemptions where required.

## Considerations for scenario one

This policy blocks the ability to create private endpoints that are in a different subscription than the service itself. If these endpoints are required for certain use cases, use policy exemptions. Create more policies for Data Factory and Azure Synapse to make sure that managed private endpoints hosted on the managed virtual network can only connect to services hosted within your Microsoft Entra tenant.

## Deny connections from private endpoints created in other tenants

**Scenario two:** A rogue administrator requires **write** access on the service in the customer environment for which a private endpoint should be created.

With this right, a rogue administrator can create a private endpoint in an external Microsoft Entra tenant and subscription. This endpoint links to a service in the customer's Microsoft Entra tenant. Figure 1 shows this scenario as connection B.

In this scenario, the rogue administrator needs to first configure an external private Microsoft Entra tenant and Azure subscription. Next, they create a private endpoint in their environment by manually specifying the resource ID and group ID of the service in the corporate Microsoft Entra tenant. Finally, they approve the private endpoint on the linked service to allow traffic over the connection across Microsoft Entra tenants.

After the rogue administrator or service owner approves the private endpoint, data is accessed from the external virtual network.

## Mitigation for scenario two

Use service-specific policies to prevent this scenario across the customer tenant. Private endpoint connections are subresources of the respective services and show up under their properties section. Deny noncompliant connections by using the following [policy definition](#):

JSON

```
"if": {
  "allof": [
    {
      "field": "type",
      "equals":
        "Microsoft.Storage/storageAccounts/privateEndpointConnections"
    },
  ]
}
```

```

{
    "field": "Microsoft.Storage/storageAccounts/privateEndpointConnections/privateLinkServiceConnectionState.status",
        "equals": "Approved"
},
{
    "anyOf": [
        {
            "field": "Microsoft.Storage/storageAccounts/privateEndpointConnections/privateEndpoint.id",
                "exists": false
            },
            {
                "value": "[split(concat(field('Microsoft.Storage/storageAccounts/privateEndpointConnections/privateEndpoint.id'), '//'), '/')][2]]",
                    "notEquals": "[subscription().subscriptionId]"
                }
            ]
        }
    ],
    "then": {
        "effect": "Deny"
}

```

This policy shows an example for Azure Storage. Replicate the same policy definition for other services like [Key Vault](#), [cognitive services](#), and [SQL Server](#). Note that Azure App Service doesn't support this mitigation at this time.

To further improve manageability, bundle the service-specific policies into an initiative. The policy denies the approval of private endpoint connections to private endpoints that are hosted outside of the subscription of the respective service. It doesn't deny the rejection or removal of private endpoint connections, which is the behavior customers want. Auto-approval workflows, such as connection C, aren't affected by this policy.

But the approval of compliant private endpoint connections within the portal is blocked with this method. This block occurs because the portal UI doesn't send the resource ID of the connected private endpoint in their payload. It's recommended to use [Azure Resource Manager](#), [Azure PowerShell](#), or [Azure CLI](#) to approve the private endpoint connection.

Also, assign the policy to the top-level management group and use exemptions where required.

## Considerations for scenario two

Azure Synapse Analytics and Azure Data Factory offer managed virtual networks and managed private endpoints. Because of these new capabilities, the policy blocks the secure and private usage of these services.

It's recommended that you use an **Audit** effect instead of a **Deny** effect in the policy definition you use in the [scenario two mitigation](#). This change helps you keep track of private endpoints being created in separate subscriptions and tenants. You can also use policy exemptions for the respective data platform scopes.

## Azure Data Factory

To overcome [scenario one](#) on the managed virtual network of Azure Data Factory, use the following [policy definition](#):

JSON

```
"if": {
  "allof": [
    {
      "field": "type",
      "equals":
        "Microsoft.DataFactory/factories/managedVirtualNetworks/managedPrivateEndpoints"
    },
    {
      "anyof": [
        {
          "field":
            "Microsoft.DataFactory/factories/managedVirtualNetworks/managedPrivateEndpoints/privateLinkId",
          "exists": false
        },
        {
          "value": "
[split(field('Microsoft.DataFactory/factories/managedVirtualNetworks/managedPrivateEndpoints/privateLinkId'), '/')][2]]",
          "notequals": "[subscription().subscriptionId]"
        }
      ]
    }
  ],
  "then": {
    "effect": "[parameters('effect')]"
  }
}
```

This policy denies managed private endpoints that are linked to services, which are hosted outside the subscription of the Data Factory. You can change this policy to allow

connections to services hosted in a set of subscriptions by adding a `list` parameter and by using the `"notIn": "[parameters('allowedSubscriptions')]"` construct. We recommend this change for the data platform scope inside the tenant or environments where services with managed virtual networks and managed private endpoints are extensively used.

It's recommended that you assign this policy to the top-level management group and use exemptions where required. For the data platform, make these changes and assign the policy to the set of data platform subscriptions.

## Azure Synapse

Azure Synapse also uses managed virtual networks. We recommend applying a similar policy to the Data Factory policy for [scenario one](#). Azure Synapse doesn't provide a policy alias for managed private endpoints. But there's a data exfiltration prevention feature, which can be enforced for workspaces using the following policy:

JSON

```
"if": {
  "allOf": [
    {
      "field": "type",
      "equals": "Microsoft.Synapse/workspaces"
    },
    {
      "anyOf": [
        {
          "field":
"Microsoft.Synapse/workspaces/managedVirtualNetworkSettings.preventDataExfil
tration",
          "exists": false
        },
        {
          "field":
"Microsoft.Synapse/workspaces/managedVirtualNetworkSettings.preventDataExfil
tration",
          "notEquals": true
        }
      ]
    },
    {
      "count": {
        "field":
"Microsoft.Synapse/workspaces/managedVirtualNetworkSettings.allowedAadTenant
IdsForLinking[*]",
        "where": {
          "field":
"Microsoft.Synapse/workspaces/managedVirtualNetworkSettings.allowedAadTenant
IdsForLinking[*]",
          "notEquals": "[subscription().tenantId]"
        }
      }
    }
  ]
}
```

```
        }
      ],
    }
  ],
},
"then": {
  "effect": "Deny"
}
}
```

This policy enforces the use of the data exfiltration feature of Azure Synapse. With Azure Synapse, you can deny any private endpoint that's coming from a service that's hosted outside of the customer tenant. You can also deny any private endpoint hosted outside of a specified set of tenant IDs. This policy only allows creating managed private endpoints that are linked to services, which are hosted in the customer tenant.

These policies are now available as built-in.

- Azure Synapse workspaces should allow outbound data traffic only to approved targets.

Definition ID: `/providers/Microsoft.Authorization/policyDefinitions/3484ce98-c0c5-4c83-994b-c5ac24785218`

- Azure Synapse managed private endpoints should only connect to resources in approved Microsoft Entra tenants.

Definition ID: `/providers/Microsoft.Authorization/policyDefinitions/3a003702-13d2-4679-941b-937e58c443f0`

It's recommended that you assign the policy to the top-level management group and use exemptions where required.

## Next steps

It's important to understand the recommended connectivity models for inbound and outbound connectivity to and from the public internet. The next article reviews design considerations, design recommendations, and recommended content for further reading.

[Inbound and outbound connectivity](#)

# Resource organization for cloud-scale analytics

Article • 11/27/2024

To align with the Ready methodology of the Cloud Adoption Framework, implement a naming and tagging strategy. Your strategy should include business and operational details as components of resource names and metadata tags. For more information, see [Develop your naming and tagging strategy for Azure resources](#).

Cloud-scale analytics includes a data management landing zone subscription. This subscription has the standard services of an [enterprise-scale framework](#). It's connected to the data landing zones and connectivity subscriptions by using virtual network peering. For more information on the subscriptions in cloud-scale analytics, see [data management landing zone](#) and [data landing zone](#).

You can further enforce organizational standards based on business rules by using the Azure Policy service. Assign these policies to a scope of resources, such as management groups, subscriptions, resource groups, or individual resources. Cloud-scale analytics contains custom policies that apply to the data management landing zone and data landing zone subscriptions. For more information, see [Policies](#).

## Resource naming and tagging conventions

Organize your cloud assets to support governance, operational management, and accounting requirements. Well-defined naming and metadata tagging conventions help to quickly locate and manage resources. These conventions also help associate cloud usage costs with business teams via chargeback and showback accounting mechanisms.

## Next steps

- [Security, governance, and compliance for enterprise-scale cloud-scale analytics](#)

---

## Feedback

Was this page helpful?



# Security, governance, and compliance for cloud-scale analytics

Article • 11/27/2024

When planning cloud-scale analytics architecture, pay special attention to ensure that the architecture is robust and secure. This article addresses security, compliance, and governance design criteria for enterprise-scale cloud-scale analytics. This article also discusses design recommendations and best practices for deployment of a cloud-scale analytics on Azure. Review [enterprise-scale security governance and compliance](#) to fully prepare for governance of an enterprise solution.

Cloud solutions initially hosted single, relatively isolated applications. As the benefits of cloud solutions became clear, larger-scale workloads were hosted in the cloud, such as SAP on Azure. So it became vital to address the security, reliability, performance, and cost of regional deployments throughout the lifecycle of cloud services.

The vision for cloud-scale analytics landing zone security, compliance, and governance on Azure is to provide tools and processes that help you minimize risk and make effective decisions. The Azure landing zones define security governance and compliance roles and responsibilities.

Cloud-scale analytics pattern relies on several security features that can be enabled in Azure. These features include encryption, role-based access control, access control lists, and networking restrictions.

## Security design recommendations

Both Microsoft and customers share responsibility for security. For accepted security guidance, refer to [Cybersecurity best practices](#) by the Center for Internet Security. The following sections are security design recommendations.

### Data-at-rest encryption

Data-at-rest encryption refers to the encryption of data as it persists in storage, and addresses the security risks related to direct physical access of storage media. Data-at-rest is a critical security control since the underlying data is unrecoverable and can't be changed without its decryption key. Data-at-rest is an important layer in the defense-in-depth strategy of Microsoft datacenters. Often, there are compliance and governance reasons to deploy data-at-rest encryption.

Several Azure services support data-at-rest encryption, including Azure Storage and Azure SQL databases. Although common concepts and models influence the design of Azure services, each service can apply data-at-rest encryption at different stack layers or have different encryption requirements.

**ⓘ Important**

All services that support data-at-rest encryption should have it enabled by default.

## Secure data in transit

Data is considered in transit or in flight when it moves from one location to another. This transit can occur internally, on-premises or within Azure, or externally, such as across the internet to an end user. Azure offers several mechanisms, including encryption, to keep data private during transit. These mechanisms include:

- Communication through VPNs using IPsec/IKE encryption.
- Transport Layer Security (TLS)
- Protocols available on Azure Virtual Machines, such as Windows IPsec or SMB.

Encryption using MACsec (media access control security), an IEEE standard at the data-link layer, is automatically enabled for all Azure traffic between Azure datacenters. This encryption ensures customer data confidentiality and integrity. For more information, see [Azure customer data protection](#).

## Manage keys and secrets

To control and manage disk encryption keys and secrets for cloud-scale analytics, use Azure Key Vault. Key Vault has capabilities for provisioning and managing SSL/TLS certificates. You can also protect secrets with hardware security modules (HSMs).

## Microsoft Defender for Cloud

Microsoft Defender for Cloud provides security alerts and advanced threat protection for virtual machines, SQL databases, containers, web applications, virtual networks, and more.

When you enable Defender for Cloud from the pricing and settings area, the following Microsoft Defender plans are enabled simultaneously and provide comprehensive defenses for the compute, data, and service layers of your environment:

- Microsoft Defender for servers
- Microsoft Defender for App Service
- Microsoft Defender for Storage
- Microsoft Defender for SQL
- Microsoft Defender for Kubernetes
- Microsoft Defender for container registries
- Microsoft Defender for Key Vault
- Microsoft Defender for Resource Manager
- Microsoft Defender for DNS

These plans are explained separately in the Defender for Cloud documentation.

 **Important**

Where Defender for Cloud is available for platform as a service (PaaS) offerings, you should enable this feature by default, especially for Azure Data Lake Storage accounts. For more information, see [Introduction to Microsoft Defender for Cloud](#) and [configure Microsoft Defender for Storage](#).

## Microsoft Defender for Identity

Microsoft Defender for Identity is part of the advanced data security offering, which is a unified package for advanced security capabilities. Microsoft Defender for Identity can be accessed and managed via the Azure portal.

 **Important**

Enable Microsoft Defender for Identity by default whenever it's available for the PaaS services you use.

## Enable Microsoft Sentinel

[Microsoft Sentinel](#) is a scalable, cloud-native, security information event management (SIEM), and security orchestration automated response (SOAR) solution. Microsoft Sentinel delivers intelligent security analytics and threat intelligence across the enterprise, providing a single solution for alert detection, threat visibility, proactive hunting, and threat response.

## Networking

Cloud-scale analytics prescribed view is to use Azure private endpoints for all PaaS services and not uses public IPs for all infrastructure as a service (IaaS) services. For more information, see [Cloud-scale analytics networking](#).

## Compliance and governance design recommendations

[Azure Advisor](#) helps you get a consolidated view across your Azure subscriptions. Consult Azure Advisor for reliability, resiliency, security, performance, operational excellence, and cost recommendations. The following sections are compliance and governance design recommendations.

### Use Azure Policy

[Azure Policy](#) helps enforce organizational standards and assess compliance at scale. Through its compliance dashboard, it provides an aggregated view of the overall state of the environment, with the ability to drill down into individual resources or policies.

Azure Policy helps bring your resources into compliance through bulk remediation of existing resources and automatic remediation of new resources. Several built-in policies are available, for example to restrict the location of new resources, require a tag and its value on resources, create a VM using a managed disk, or enforce naming policies.

### Automate deployments

You can save time and reduce errors by automating deployments. Reduce the deployment complexity of end-to-end data landing zones and data applications (which create data products) by creating reusable code templates. This automation minimizes the time to deploy or redeploy solutions. For more information, see [Understand DevOps automation for the cloud-scale analytics in Azure](#)

### Lock resources for production workloads

Create required core data management and data landing zone Azure resources at the start of your project. When all additions, moves, and changes are finished, and the Azure deployment is operational, lock all resources. Then, only an administrator can unlock or modify resources. For more information, see [Lock resources to prevent unexpected changes](#).

# Implement role-based access control

You can customize role-based access control (RBAC) on Azure subscriptions to manage who has access to Azure resources, what they can do with those resources, and what areas they have access to. For example, you can allow team members to deploy core assets to a data landing zone, but prevent them from altering any of the network components.

## Compliance and governance scenarios

The following recommendations apply to various compliance and governance scenarios. These scenarios represent a cost-effective and scalable solution.

  Expand table

Scenario	Recommendation
Configure a governance model with standard naming conventions, and pull reports based on cost center.	Use Azure Policy and tags to meet your requirements.
Avoid accidental deletion of Azure resources.	Use Azure resource locks to prevent accidental deletion.
Get a consolidated view of opportunity areas for cost optimization, resiliency, security, operational excellence, and performance for Azure resources.	Use Azure Advisor to get a consolidated view across SAP on Azure subscriptions.

## Next steps

[Azure policies for cloud-scale analytics](#)

## Feedback

Was this page helpful?

 Yes

 No

# Policies in cloud-scale analytics

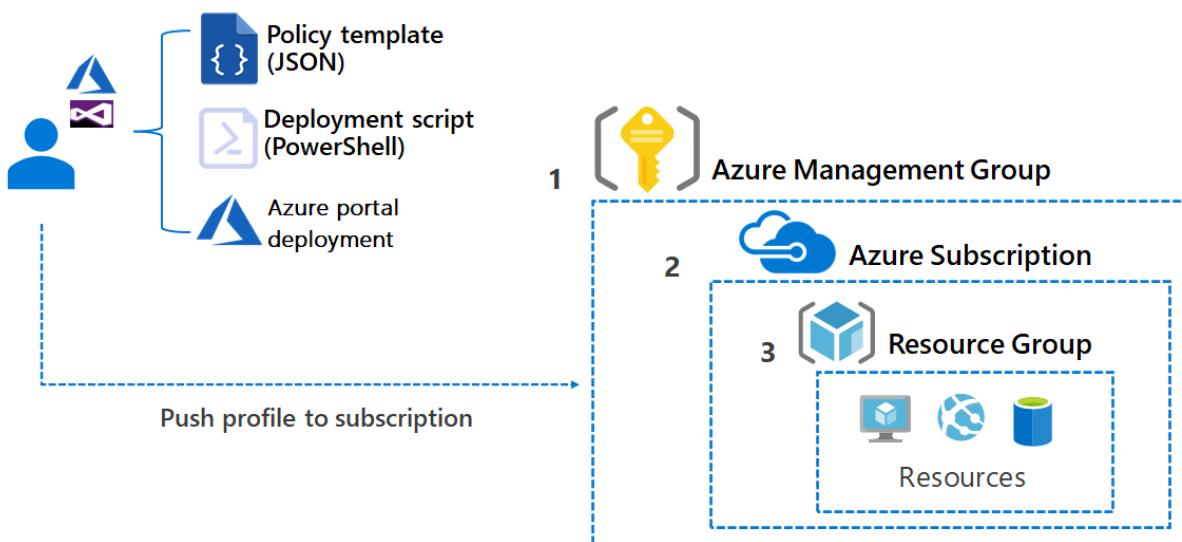
Article • 11/27/2024

Before considering a deployment, it's important for your organization to put guardrails in place. By using [Azure policies](#), you can implement governance for resource consistency, regulatory compliance, security, cost, and management.

## Background

A core principle of cloud-scale analytics is to make it easy to create, read, update, and delete resources as needed. However, while giving unrestricted resource access to developers can make them agile, it can also lead to unintended cost consequences. The solution to this problem is resource access governance. This governance is the ongoing process of managing, monitoring, and auditing the use of Azure resources to meet the goals and requirements of your organization.

The [Start with Cloud Adoption Framework enterprise-scale landing zones](#) already uses this concept. Cloud-scale analytics adds [Custom Azure policies](#) to build on these standards. The standards are then applied to our data management landing zones and data landing zones.



Azure Policy is important when ensuring security and compliance within cloud-scale analytics. It helps to enforce standards and to assess compliance at scale. Policies can be used to evaluate resources in Azure and compare them to the wanted properties. Several policies, or business rules, can be grouped into an initiative. Individual policies or initiatives can be assigned to different scopes in Azure. These scopes might be management groups, subscriptions, resource groups, or individual resources. The assignment applies to all resources within the scope, and subscopes can be excluded with exceptions if necessary.

# Design considerations

Azure policies in cloud-scale analytics were developed with the following design considerations in mind:

- Use Azure policies to implement governance and enforce rules for resource consistency, regulatory compliance, security, cost, and management.
- Use available prebuilt policies to save time.
- Assign policies to the highest level possible in the management group tree to simplify policy management.
- Limit Azure Policy assignments made at the root management group scope to avoid managing through exclusions at inherited scopes.
- Only use policy exceptions if necessary, and they require approval.

## Azure policies for cloud-scale analytics

[Implementing custom policies](#) allows you to do more with Azure Policy. Cloud-scale analytics comes with a set of precreated policies to help you implement any required guardrails in your environment.

Azure Policy should be the core instrument of the Azure (Data) Platform team to ensure compliance of resources within the Data management landing zone, data landing zones, and other landing zones within the organization's tenant. This platform feature should be used to introduce guardrails and enforce adherence to the overall approved service configuration within the respective management group scope. The platform teams can use Azure Policy to, for example, enforce private endpoints for any storage accounts that are being hosted within the data platform environment or enforce TLS 1.2 encryption in transit for any connections being made to the storage accounts. When done right, this policy prohibits any data application teams from hosting services in an incompliant state within the respective tenant scope.

The responsible IT teams should use this platform feature to address their security and compliance concerns and open up for a self-service approach within (Data) Landing Zones.

Cloud-scale analytics contains custom policies related to **resource and cost management, authentication, encryption, network isolation, logging, resilience, and more.**

- [All services](#)
- [Storage](#)
- [Key Vault](#)

- Azure Data Factory
- Azure Synapse Analytics
- Azure Databricks
- Azure IoT Hub
- Azure Event Hubs
- Azure Stream Analytics
- Azure Data Explorer
- Azure Cosmos DB
- Azure Container Registry
- Azure Cognitive Services
- Azure Machine Learning
- Azure SQL Managed Instance
- Azure SQL Database
- Azure Database for MariaDB
- Azure Database for MySQL
- Azure Database for PostgreSQL
- Azure AI Search
- Azure DNS
- Network security group
- Batch
- Azure Cache for Redis
- Container instances
- Azure Firewall
- HDInsight
- Power BI

 **Note**

The policies should be viewed as guidance-only and can be applied depending on business requirements. Policies should always be applied to the highest level possible and in most cases this will be a [management group](#).

## All services

 [Expand table](#)

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Deny-PublicIp	Network Isolation	Restrict deployment of public IPs.
Deny-PrivateEndpoint-PrivateLinkServiceConnections	Network Isolation	Deny private endpoints to resources outside of the Microsoft Entra tenant and subscription.
Deploy-DNSZoneGroup-{Service}-PrivateEndpoint	Network Isolation	Deploys the configurations of a Private DNS Zone Group by a parameter for service's private endpoint. Used to enforce the configuration to a single Private DNS Zone.
DiagnosticSettings-{Service}-LogAnalytics	Logging	Send diagnostic settings for Azure Cosmos DB to log analytics workspace.

## Storage

[\[+\] Expand table](#)

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Append-Storage-Encryption	Encryption	Enforce encryption for storage accounts.
Deny-Storage-AllowBlobPublicAccess	Network Isolation	Enforces no public access to all blobs or containers in the storage account.
Deny-Storage-ContainerDeleteRetentionPolicy	Resilience	Enforce container delete retention policies larger than seven days for storage account.
Deny-Storage-CorsRules	Network Isolation	Deny cors rules for storage account.
Deny-Storage-InfrastructureEncryption	Encryption	Enforce infrastructure (double) encryption for storage accounts.
Deny-Storage-MinimumTlsVersion	Encryption	Enforces minimum TLS version 1.2 for storage account.
Deny-Storage-NetworkAclsBypass	Network Isolation	Enforces network bypass to none for storage account.
Deny-Storage-NetworkAclsIpRules	Network Isolation	Enforces network ip rules for storage account.

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Deny-Storage-NetworkAclsVirtualNetworkRules	Network Isolation	Denies virtual network rules for storage account.
Deny-Storage-Sku	Resource Management	Enforces storage account SKUs.
Deny-Storage-SupportsHttpsTrafficOnly	Encryption	Enforces https traffic for storage account.
Deploy-Storage-BlobServices	Resource Management	Deploy blob services default settings for storage account.
Deny-Storage-RoutingPreference	Network Isolation	
Deny-Storage-Kind	Resource Management	
Deny-Storage-NetworkAclsDefaultAction	Network Isolation	

## Key Vault

[Expand table](#)

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Audit-KeyVault-PrivateEndpointId	Network Isolation	Audit public endpoints that are created in other subscriptions for key vault.
Deny-KeyVault-NetworkAclsBypass	Network Isolation	Enforces bypass network level rules for key vault.
Deny-KeyVault-NetworkAclsDefaultAction	Network Isolation	Enforces default network acl level action for key vault.
Deny-KeyVault-NetworkAclsIpRules	Network Isolation	Enforces network ip rules for key vault.
Deny-KeyVault-NetworkAclsVirtualNetworkRules	Network Isolation	Denies virtual network rules for key vault.
Deny-KeyVault-PurgeProtection	Resilience	Enforces purge protection for key vault.
Deny-KeyVault-SoftDelete	Resilience	Enforces soft delete with minimum number of retention days for key

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
		vault.
Deny-KeyVault-TenantId	Resource Management	Enforce tenant ID for key vault.

## Azure Data Factory

[\[+\] Expand table](#)

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Append-DataFactory-IdentityType	Authentication	Enforces use of system assigned identity for data factory.
Deny-DataFactory-ApiVersion	Resource Management	Denies old API version for data factory V1.
Deny-DataFactory-IntegrationRuntimeManagedVirtualNetwork	Network Isolation	Denies Integration Runtimes that aren't connected to the Managed Virtual Network.
Deny-DataFactory-LinkedServicesConnectionStringType	Authentication	Denies non Key Vault stored secrets for linked services.
Deny-DataFactory-ManagedPrivateEndpoints	Network Isolation	Denies external private endpoints for linked services.
Deny-DataFactory-PublicNetworkAccess	Network Isolation	Denies public access to data factory.
Deploy-DataFactory-ManagedVirtualNetwork	Network Isolation	Deploy managed virtual network for data factory.
Deploy-SelfHostedIntegrationRuntime-Sharing	Resilience	Share self-hosted integration runtime hosted in the Data Hub with Data Factories in the Data Nodes.

## Azure Synapse Analytics

[\[+\] Expand table](#)

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Append-Synapse-LinkedAccessCheckOnTargetResource	Network Isolation	Enforce <a href="#">LinkedAccessCheckOnTargetResource</a> in managed virtual network settings when Synapse Workspace is created.
Append-Synapse-Purview	Network Isolation	Enforce connection between central purview instance and Synapse Workspace.
Append-SynapseSpark-Computelsolation	Resource Management	When a Synapse Spark Pool is created without compute isolation then this adds it.
Append-SynapseSpark-DefaultSparkLogFolder	Logging	When a Synapse Spark Pool is created without logging then this will add it.
Append-SynapseSpark-SessionLevelPackages	Resource Management	When a Synapse Spark Pool is created without session level packages then this adds it.
Audit-Synapse-PrivateEndpointId	Network Isolation	Audit public endpoints that are created in other subscriptions for Synapse.
Deny-Synapse-AllowedAadTenantIdsForLinking	Network Isolation	
Deny-Synapse-Firewall	Network Isolation	Set up firewall of Synapse.
Deny-Synapse-ManagedVirtualNetwork	Network Isolation	When a Synapse Workspace is created without managed virtual network then this adds it.
Deny-Synapse-PreventDataExfiltration	Network Isolation	Enforced prevention of data exfiltration for Synapse managed virtual network.
Deny-SynapsePrivateLinkHub	Network Isolation	Denies Synapse Private Link Hub.
Deny-SynapseSpark-AutoPause	Resource Management	Enforces auto pause for Synapse Spark Pools.
Deny-SynapseSpark-AutoScale	Resource Management	Enforces auto scale for Synapse Spark Pools.
Deny-SynapseSql-Sku	Resource Management	Denies certain Synapse SQL Pool SKUs.
Deploy-SynapseSql-AuditingSettings	Logging	Send auditing logs for Synapse SQL pools to log analytics.

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Deploy-SynapseSql-MetadataSynch	Resource Management	Set up metadata sync for Synapse SQL pools.
Deploy-SynapseSql-SecurityAlertPolicies	Logging	Deploy Synapse SQL pool security alert policy.
Deploy-SynapseSql-TransparentDataEncryption	Encryption	Deploy Synapse SQL transparent data encryption.
Deploy-SynapseSql-VulnerabilityAssessment	Logging	Deploy Synapse SQL pool vulnerability assessments.

## Azure Databricks

[\[+\] Expand table](#)

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Append-Databricks-PublicIp	Network Isolation	Enforces no public access on Databricks workspaces.
Deny-Databricks-Sku	Resource Management	Deny nonpremium Databricks SKU.
Deny-Databricks-VirtualNetwork	Network Isolation	Deny nonvirtual network deployment for databricks.

Other policies that are applied in the Databricks workspace through cluster policies:

[\[+\] Expand table](#)

<b>Cluster policy name</b>	<b>Policy area</b>
Restrict Spark version	Resource Management
Restrict cluster size and VM types	Resource Management
Enforce Cost Tagging	Resource Management
Enforce Autoscale	Resource Management
Enforce Auto-Pause	Resource Management
Restrict DBUs per hour	Resource Management
Deny public SSH	Authentication

Cluster policy name	Policy area
Cluster credential passthrough enabled	Authentication
Enable process isolation	Network isolation
Enforce Spark monitoring	Logging
Enforce cluster logs	Logging
Allow only SQL, Python	Resource management
Deny additional setup scripts	Resource management

## Azure IoT Hub

[\[+\] Expand table](#)

Policy name	Policy area	Description
Append-IotHub-MinimalTlsVersion	Encryption	Enforces minimal TLS version for iot hub.
Audit-IotHub-PrivateEndpointId	Network Isolation	Audit public endpoints that are created in other subscriptions for iot hubs.
Deny-IotHub-PublicNetworkAccess	Network Isolation	Denies public network access for iot hub.
Deny-IotHub-Sku	Resource Management	Enforces iot hub SKUs.
Deploy-IotHub-IoTSecuritySolutions	Security	Deploy Microsoft Defender for IoT for IoT Hubs.

## Azure Event Hubs

[\[+\] Expand table](#)

Policy name	Policy area	Description
Deny-EventHub-Ipfilterrules	Network Isolation	Deny adding ip filter rules for Azure Event Hubs.
Deny-EventHub-MaximumThroughputUnits	Network Isolation	Denies public network access for my SQL servers.

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Deny-EventHub-NetworkRuleSet	Network Isolation	Enforces default virtual network rules for Azure Event Hubs.
Deny-EventHub-Sku	Resource Management	Denies certain AKUs for Azure Event Hubs.
Deny-EventHub-Virtualnetworkrules	Network Isolation	Deny adding virtual network rules for Azure Event Hubs.

## Azure Stream Analytics

[\[+\] Expand table](#)

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Append-StreamAnalytics-IdentityType	Authentication	Enforces use of system assigned identity for stream analytics.
Deny-StreamAnalytics-ClusterId	Resource Management	Enforces use of Stream Analytics cluster.
Deny-StreamAnalytics-StreamingUnits	Resource Management	Enforces number of stream analytics streaming units.

## Azure Data Explorer

[\[+\] Expand table](#)

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Deny-DataExplorer-DiskEncryption	Encryption	Enforces use of disk encryption for data explorer.
Deny-DataExplorer-DoubleEncryption	Encryption	Enforces use of double encryption for data explorer.
Deny-DataExplorer-Identity	Authentication	Enforces use of system or user assigned identity for data explorer.
Deny-DataExplorer-Sku	Resource Management	Enforces data explorer SKUs.
Deny-DataExplorer-TrustedExternalTenants	Network Isolation	Denies external tenants for data explorer.

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Deny-DataExplorer-VirtualNetworkConfiguration	Network Isolation	Enforces virtual network ingestion for data explorer.

## Azure Cosmos DB

[Expand table](#)

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Append-Cosmos-DenyCosmosKeyBasedMetadataWriteAccess	Authentication	Deny key based metadata write access for Azure Cosmos DB accounts.
Append-Cosmos-PublicNetworkAccess	Network Isolation	Enforces no public network access for Azure Cosmos DB accounts.
Audit-Cosmos-PrivateEndpointId	Network Isolation	Audit public endpoints that are created in other subscriptions for Azure Cosmos DB.
Deny-Cosmos-Cors	Network Isolation	Denies CORS rules for Azure Cosmos DB accounts."
Deny-Cosmos-PublicNetworkAccess	Network Isolation	Denies public network access for Azure Cosmos DB accounts.

## Azure Container Registry

[Expand table](#)

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Audit-ContainerRegistry-PrivateEndpointId	Network Isolation	Audit public endpoints that are created in other subscriptions for cognitive services.
Deny-ContainerRegistry-PublicNetworkAccess	Network Isolation	Denies public network access for container registry.
Deny-ContainerRegistry-Sku	Resource Management	Enforces premium Sku for container registry.

# Azure Cognitive Services

[Expand table](#)

Policy name	Policy area	Description
Append-CognitiveServices-IdentityType	Authentication	Enforces use of system assigned identity for cognitive services.
Audit-CognitiveServices-PrivateEndpointId	Network Isolation	Audit public endpoints that are created in other subscriptions for cognitive services.
Deny-CognitiveServices-Encryption	Encryption	Enforces use of encryption for cognitive services.
Deny-CognitiveServices-PublicNetworkAccess	Network Isolation	Enforces no public network access for cognitive services.
Deny-CognitiveServices-Sku	Resource Management	Deny cognitive services free SKU.
Deny-CognitiveServices-UserOwnedStorage	Network Isolation	Enforces user owned storage for cognitive services.

# Azure Machine Learning

[Expand table](#)

Policy name	Policy area	Description
Append-MachineLearning-PublicAccessWhenBehindVnet	Network Isolation	Deny public access behind virtual network for machine learning workspaces.
Audit-MachineLearning-PrivateEndpointId	Network Isolation	Audit public endpoints that are created in other subscriptions for machine learning.
Deny-MachineLearning-HbiWorkspace	Network Isolation	Enforce high business impact machine learning workspaces across the environment.
Deny-MachineLearningAks	Resource Management	Deny AKS creation (not attaching) in machine learning.
Deny-MachineLearningCompute-SubnetId	Network Isolation	Deny public IP for machine learning compute clusters and instances.

Policy name	Policy area	Description
Deny-MachineLearningCompute-VmSize	Resource Management	Limit allowed vm sizes for machine learning compute clusters and instances.
Deny-MachineLearningComputeCluster-RemoteLoginPortPublicAccess	Network Isolation	Deny public access of clusters via SSH.
Deny-MachineLearningComputeCluster-Scale	Resource Management	Enforce scale settings for machine learning compute clusters.

## Azure SQL Managed Instance

[Expand table](#)

Policy name	Policy area	Description
Append-SqlManagedInstance-MinimalTlsVersion	Encryption	Enforces minimal TLS version for SQL Managed Instance servers.
Deny-SqlManagedInstance-PublicDataEndpoint	Network Isolation	Denies public data endpoint for SQL Managed Instances.
Deny-SqlManagedInstance-Sku	Resource Management	
Deny-SqlManagedInstance-SubnetId	Network Isolation	Enforces deployments to subnets of SQL Managed Instances.
Deploy-SqlManagedInstance-AzureAdOnlyAuthentications	Authentication	Enforces Microsoft Entra-only authentication for SQL Managed Instance.
Deploy-SqlManagedInstance-SecurityAlertPolicies	Logging	Deploy SQL Managed Instance security alert policies.
Deploy-SqlManagedInstance-VulnerabilityAssessment	Logging	Deploy SQL Managed Instance vulnerability assessments.

## Azure SQL Database

[Expand table](#)

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Append-Sql-MinimalTlsVersion	Encryption	Enforces minimal TLS version for SQL servers.
Audit-Sql-PrivateEndpointId	Network Isolation	Audit public endpoints that are created in other subscriptions for Azure SQL.
Deny-Sql-PublicNetworkAccess	Network Isolation	Denies public network access for SQL servers.
Deny-Sql-StorageAccountType	Resilience	Enforces geo-redundant database backup.
Deploy-Sql-AuditingSettings	Logging	Deploy SQL auditing settings.
Deploy-Sql-AzureAdOnlyAuthentications	Authentication	Enforces Microsoft Entra-only authentication for SQL server.
Deploy-Sql-SecurityAlertPolicies	Logging	Deploy SQL security alert policies.
Deploy-Sql-TransparentDataEncryption	Encryption	Deploy SQL transparent data encryption.
Deploy-Sql-VulnerabilityAssessment	Logging	Deploy SQL vulnerability assessments.
Deploy-Sqldw-AuditingSettings	Logging	Deploy SQL DW auditing settings.

## Azure Database for MariaDB

[\[+\] Expand table](#)

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Append-MariaDb-MinimalTlsVersion	Encryption	Enforces minimal TLS version for MariaDB servers.
Audit-MariaDb-PrivateEndpointId	Network Isolation	Audit public endpoints that are created in other subscriptions for MariaDB.
Deny-MariaDb-PublicNetworkAccess	Network Isolation	Denies public network access for my MariaDB servers.
Deny-MariaDb-StorageProfile	Resilience	Enforces geo-redundant database backup with minimum retention time in days.
Deploy-MariaDb-SecurityAlertPolicies	Logging	Deploy SQL security alert policies for MariaDB

# Azure Database for MySQL

[Expand table](#)

Policy name	Policy area	Description
Append-MySQL-MinimalTlsVersion	Encryption	Enforces minimal TLS version for MySQL servers.
Audit-MySQL-PrivateEndpointId	Network Isolation	Audit public endpoints that are created in other subscriptions for MySQL.
Deny-MySQL-InfrastructureEncryption	Encryption	Enforces infrastructure encryption for MySQL servers.
Deny-MySQL-PublicNetworkAccess	Network Isolation	Denies public network access for MySQL servers.
Deny-MySQL-StorageProfile	Resilience	Enforces geo-redundant database backup with minimum retention time in days.
Deploy-MySQL-SecurityAlertPolicies	Logging	Deploy SQL security alert policies for MySQL.

# Azure Database for PostgreSQL

[Expand table](#)

Policy name	Policy area	Description
Append-PostgreSQL-MinimalTlsVersion	Encryption	Enforces minimal TLS version for PostgreSQL servers.
Audit-PostgreSQL-PrivateEndpointId	Network Isolation	Audit public endpoints that are created in other subscriptions for PostgreSQL.
Deny-PostgreSQL-InfrastructureEncryption	Encryption	Enforces infrastructure encryption for PostgreSQL servers.
Deny-PostgreSQL-PublicNetworkAccess	Network Isolation	Denies public network access for PostgreSQL servers.
Deny-PostgreSQL-StorageProfile	Resilience	Enforces geo-redundant database backup with minimum retention time in days.
Deploy-PostgreSQL-SecurityAlertPolicies	Logging	Deploy SQL security alert policies for PostgreSQL.

# Azure AI Search

[\[+\] Expand table](#)

Policy name	Policy area	Description
Append-Search-IdentityType	Authentication	Enforces use of system assigned identity for Azure AI Search.
Audit-Search-PrivateEndpointId	Network Isolation	Audit public endpoints that are created in other subscriptions for Azure AI Search.
Deny-Search-PublicNetworkAccess	Network Isolation	Denies public network access for Azure AI Search.
Deny-Search-Sku	Resource Management	Enforces Azure AI Search SKUs.

# Azure DNS

[\[+\] Expand table](#)

Policy name	Policy area	Description
Deny-PrivateDnsZones	Resource Management	Restrict deployment of private DNS zones to avoid proliferation.

# Network security group

[\[+\] Expand table](#)

Policy name	Policy area	Description
Deploy-Nsg-FlowLogs	Logging	Deploy NSG flow logs and traffic analytics.

# Batch

[\[+\] Expand table](#)

Policy name	Policy area	Description
Deny-Batch-InboundNatPools	Network Isolation	Denies inbound NAT pools for batch account VM pools.

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Deny-Batch-NetworkConfiguration	Network Isolation	Denies public IP addresses for batch account VM pools.
Deny-Batch-PublicNetworkAccess	Network Isolation	Denies public network access for batch accounts.
Deny-Batch-Scale	Resource Management	Denies certain scale configurations for batch account VM pools.
Deny-Batch-VmSize	Resource Management	Denies certain VM sizes for batch account VM pools.

## Azure Cache for Redis

[\[+\] Expand table](#)

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Deny-Cache-Enterprise	Resource Management	Denies Redis Cache Enterprise.
Deny-Cache-FirewallRules	Network Isolation	Denies firewall rules for Redis Cache.
Deny-Cache-MinimumTlsVersion	Encryption	Enforces minimum TLS version for Redis Cache.
Deny-Cache-NonSslPort	Network Isolation	Enforces turning off the non-SSL port for Redis Cache.
Deny-Cache-PublicNetworkAccess	Network Isolation	Enforces no public network access for Redis Cache.
Deny-Cache-Sku	Resource Management	Enforces certain SkUs for Redis Cache.
Deny-Cache-VnetInjection	Network Isolation	Enforces use of private endpoints and denies virtual network injection for Redis Cache.

## Container instances

[\[+\] Expand table](#)

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Deny-ContainerInstance-PublicIpAddress	Network Isolation	Denies public Container Instances created from Azure Machine Learning.

## Azure Firewall

[\[+\] Expand table](#)

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Deny-Firewall	Resource Management	Restrict deployment of Azure Firewall to avoid proliferation.

## HDInsight

[\[+\] Expand table](#)

<b>Policy name</b>	<b>Policy area</b>	<b>Description</b>
Deny-HdInsight-EncryptionAtHost	Encryption	Enforce encryption at host for HDInsight clusters.
Deny-HdInsight-EncryptionInTransit	Encryption	Enforces encryption in transit for HDInsight clusters.
Deny-HdInsight-MinimalTlsVersion	Encryption	Enforces minimal TLS version for HDInsight clusters.
Deny-HdInsight-NetworkProperties	Network Isolation	Enforces private link enablement for HDInsight clusters.
Deny-HdInsight-Sku		Enforces certain SKUs for HDInsight clusters.
Deny-HdInsight-VirtualNetworkProfile	Network Isolation	Enforces virtual network injection for HDInsight clusters.

## Power BI

[\[+\] Expand table](#)

Policy name	Policy area	Description
Deny-PrivateLinkServicesForPowerBI	Resource Management	Restrict deployment of private link services for Power BI to avoid proliferation.

## Next steps

- Requirements for governing data in a modern enterprise
  - Components needed for data governance
- 

## Feedback

Was this page helpful?



# Management and monitoring

Article • 11/27/2024

This article provides guidance about maintaining an Azure data analytics estate. It builds on considerations and recommendations of the Azure landing zones. For more information, see [Management and monitoring](#).

## Design considerations

Here are some design considerations for data analytics on Azure monitoring and management:

- Consider a centralized Azure Log Analytics workspace with Azure Monitor and Application Insights for platform and application layer monitoring. For more information, see [Create a Log Analytics workspace in the Azure portal](#).
- Consider whether individual data landing zones can query the centralized Log Analytics workspace with appropriate permissions. Each landing zone might also need its own Log Analytics workspace.
- Configure Azure Databricks to send monitoring information to Log Analytics. For more information, see [Monitoring Azure Databricks](#).
- Review sample queries. You can use some without modification or build on them for your own queries. For more information, see [Using queries in Azure Monitor Log Analytics](#).

## Design recommendations

When evaluating cloud-scale analytics, consider the following design recommendations:

- Implement threat protection. For more information, see [Microsoft Sentinel](#).
- Monitor all services deployed in the data landing zone to a Log Analytics workspace.
- Use Azure Site Recovery to recover virtual machines that support mission-critical workloads. For more information, see [About Site Recovery](#).
- In a data landing zone, all monitoring should be sent to the enterprise-scale management subscription for analysis.

# Next steps

- Management and monitoring
  - Create a Log Analytics workspace in the Azure portal
  - Business continuity and disaster recovery
- 

## Feedback

Was this page helpful?

 Yes

 No

# Business continuity and disaster recovery for cloud-scale analytics

Article • 12/10/2024

When you design architecture for a cloud service, consider your availability requirements and how to respond to potential interruptions in the service. An issue could be localized to the specific instance or region-wide. Having plans for both is important. Depending on your recovery time objective and the recovery point objective, you might choose an aggressive strategy for high availability and disaster recovery.

High availability and disaster recovery can sometimes be combined. The two areas have slightly different strategies, especially when it comes to data. To learn more, see the [Microsoft Azure Well-Architected Framework](#) and its [reliability principles](#).

Instead of trying to prevent failures, accept up front that failures can and do happen. Minimize the effects of any single failing component in the lifecycle. Your tolerance for cost, recovery point objective, and recovery time objective determine the type of solution to implement.

## Backup strategies

Many alternative strategies are available for implementing distributed compute across regions. Strategies must be tailored to business requirements and circumstances of your application. At a high level, the approaches fall into the following categories:

- **Backup and restore:** Restore the database application from the last backup copy before the disaster. This approach is commonly used following data corruption or accidental deletion.
- **Redeploy on disaster:** Redeploy the application from scratch at the time of disaster. This approach is appropriate for noncritical applications that don't require a guaranteed recovery time.
- **Warm spare (active/passive):** Create a secondary hosted service in an alternate region. Deploy roles to guarantee minimal capacity. The roles don't receive production traffic. This approach is useful for applications that aren't designed to distribute traffic across regions.
- **Hot spare (active/active):** Design the application to receive production load in multiple regions. You might configure the cloud services in each region for higher

capacity than required for disaster recovery purposes. Instead, you could scale out the cloud services as necessary at the time of a disaster and failover.

This approach requires investment in application design, but has benefits. It offers low and guaranteed recovery time. There's continuous testing of all recovery locations and efficient usage of capacity. For database applications, this approach includes a load balancer for two databases that synchronize with a single connection point.

## Disaster recovery and high availability for Azure services

Cloud Scale Analytics is made of several Azure services which are grouped into platform, core, and data. For more information on service specific reliability guides and disaster recovery, see [Azure reliability documentation](#)

## Next steps

- [Data Governance Overview](#)
- 

## Feedback

Was this page helpful?

 Yes

 No

# Data governance overview

Article • 11/27/2024

The key to successful data governance is to break down structured data into data entities and data subject areas. You can then use a data governance solution to surround your specific data entities and data subject areas with people, processes, policies, and technology. The solution helps you govern your data entities' lifecycles. Establishing a common business vocabulary in a glossary within your data catalog also helps you govern your data.

Your data catalog technology is critical. You can't govern your data if you don't know where the data is or what it means. Data catalog software provides automatic data discovery, automatic profiling that determines data quality, and automatic sensitive data detection. Data catalog technology also helps you map disparate data to the common vocabulary data names and definitions in your catalog's business glossary to understand what the data means.

Data classification categorizes data assets by assigning them unique logical labels or classes based on business context. Examples of classification labels or classes include:

- Passport number.
- Driver's license number.
- Credit card number.
- SWIFT code.
- Individual's name.

You can define data classification schemes such as a [data confidentiality classification scheme](#) in your data catalog. To define the scheme, you associate policies and rules in your catalog with different classification levels.

A [data lifecycle retention classification scheme](#) provides different retention classifications for data lifecycle management. A custom microservice lifecycle application can use this scheme to maintain the data lifecycle within your environment.

Label or tag data attributes in your business glossary with confidentiality and retention classifications that specify their governance. Labeling an attribute in your glossary automatically defines how to govern data mapped to the attribute in underlying data stores. Your data catalog maps the physical data attributes in different data stores to the business glossary attributes.

You can integrate multiple technologies with your data catalog to access these attributes and enforce policies and rules across all data stores in your distributed data

landscape. You can also apply the same classification labels to unstructured data.

Master data entities are important because their data is widely shared. Master data entities are often associated with documents. Customer and invoice, supplier and contract, and asset and operating manual are example master data entity and document pairings. By using this type of connection, you can tag related documents with master data values, such as a supplier name, and preserve relationships between structured and unstructured data.

You can create pipelines that create trusted data assets by using the common vocabulary data entities from your data catalog. You can then publish these assets in a data marketplace to share.

The key takeaway is that you can use available data governance methods to get your data under control. After your data is trusted, you can use the data to drive value. How well you organize and coordinate your data governance determines your level of success.

## Data governance maturity model

The data governance maturity model describes the maturity of your ability to cover all governance aspects across your data landscape. The following tables can help you assess your current position in the data governance maturity model.

### People

[ ] Expand table

Ungoverned	Stage 1	Stage 2	Fully governed
No stakeholder executive sponsor	Stakeholder sponsor in place	Stakeholder sponsor in place	Stakeholder sponsor in place
No roles and responsibilities	Roles and responsibilities defined	Roles and responsibilities defined	Roles and responsibilities defined
No data governance control board	Data governance control board in place but no data	Data governance control board in place with data	Data governance control board in place with data
No data governance working groups	No data governance working groups	Some data governance working groups in place	All data governance working groups in place

<b>Ungoverned</b>	<b>Stage 1</b>	<b>Stage 2</b>	<b>Fully governed</b>
No data owners accountable for data	No data owners accountable for data	Some data owners in place	All data owners in place
No data stewards appointed with responsibility for data quality	Some data stewards in place for data quality, but scope too broad, like whole department	Data stewards in place and assigned to data governance working groups for specific data	Data stewards in place assigned to data governance working groups for specific data
No one accountable for data privacy	No one accountable for data privacy	Chief privacy officer accountable for privacy, no tools	Chief privacy officer accountable for privacy with tools
No one accountable for access security	IT accountable for access security	IT security accountable for access security	IT security accountable for access security and responsible for privacy enforcement
No trusted data asset producer	Data publisher identified and accountable for producing trusted data	Data publisher identified and accountable for producing trusted data	Data publisher identified and accountable for producing trusted data
No subject-matter experts (SMEs) identified for data entities	Some SMEs identified, but not engaged	SMEs identified and in data governance working groups	SMEs identified and in data governance working groups

## Process

[\[+\] Expand table](#)

<b>Ungoverned</b>	<b>Stage 1</b>	<b>Stage 2</b>	<b>Fully governed</b>
No common business vocabulary	Common business vocabulary begun in a glossary	Common business vocabulary established	Common business vocabulary complete and maintained
No way to know data location, quality or sensitivity	Data catalog auto data discovery, profiling, and sensitive data detection on some systems	Data catalog auto data discovery, profiling, and sensitive data detection on all structured data	Data catalog auto data discovery, profiling, and sensitive data detection on structured and unstructured data in all

<b>Ungoverned</b>	<b>Stage 1</b>	<b>Stage 2</b>	<b>Fully governed</b>
			systems, with full auto-tagging
No process to govern authoring or maintenance of policies and rules	Governance of data access security, policy authoring, and maintenance on some systems	Governance of data access security, privacy, and retention policy authoring and maintenance	Governance of data access security, privacy, and retention policy authoring and maintenance
No way to enforce policies and rules	Piecemeal enforcement of data access security policies and rules across systems with no catalog integration	Enforcement of data access security and privacy policies and rules across systems with catalog integration	Enforcement of data access security, privacy, and retention policies and rules across all systems
No process to monitor data quality, data privacy, or data access security	Some ability to monitor data quality, some ability to monitor privacy, such as queries	Monitoring and stewardship of data quality and data privacy on core systems with database management system (DBMS) masking	Monitoring and stewardship of data quality and data privacy on all systems with dynamic masking
No availability of fully trusted data assets	Development started for a small set of trusted data assets using data fabric software	Several core trusted data assets created using data fabric	Continuous delivery of trusted data assets through enterprise data marketplace
No way to know whether a policy violation occurs or process if one occurs	Data access security violation detection in some systems	Data access security violation detection in all systems	Data access security violation detection in all systems
No vulnerability testing process	Limited vulnerability testing process	Vulnerability testing process for all systems	Vulnerability testing process for all systems
No common process for master data creation, maintenance, and sync	Master data management (MDM) with common master data create, read, update, and delete (CRUD) and sync processes for single entities	MDM with common master data CRUD and sync processes for some data entities	MDM with common master data CRUD and sync processes for all master data entities

# Policies

[\[+\] Expand table](#)

Ungoverned	Stage 1	Stage 2	Fully governed
No data governance classification schemes on confidentiality and retention	Data governance classification scheme for confidentiality	Data governance classification scheme for both confidentiality and retention	Data governance classification scheme for both confidentiality and retention
No policies and rules to govern data quality	Policies and rules to govern data quality begun in common vocabulary in business glossary	Policies and rules to govern data quality defined in common vocabulary in catalog business glossary	Policies and rules to govern data quality defined in common vocabulary in catalog business glossary
No policies and rules to govern data access security	Some policies and rules to govern data access security created in different technologies	Policies and rules to govern data access security consolidated in the data catalog using a classification scheme	Policies and rules to govern data access security consolidated in the data catalog using a classification scheme and enforced everywhere
No policies and rules to govern data privacy	Some policies and rules to govern data privacy	Policies and rules to govern data privacy consolidated in the data catalog using a classification scheme	Policies and rules to govern data privacy consolidated in the data catalog using a classification scheme and enforced everywhere
No policies and rules to govern data retention	Some policies and rules to govern data retention	Policies and rules to govern data retention consolidated in the data catalog using a classification scheme	Policies and rules to govern data retention consolidated in the data catalog using classification schemes and enforced everywhere
No policies and rules to govern master data maintenance	Policies and rules to govern master data maintenance for a single master data entity	Policies and rules to govern master data maintenance for some master data entities	Policies and rules to govern master data maintenance for all master data entities

# Technology

Ungoverned	Stage 1	Stage 2	Fully governed
No data catalog with auto data discovery, profiling, and sensitive data detection	Data catalog with auto data discovery, profiling, and sensitive data detection purchased	Data catalog with auto data discovery, profiling, and sensitive data detection purchased	Data catalog with auto data discovery, profiling, and sensitive data detection purchased
No data fabric software with multicloud edge and datacenter connectivity	Data fabric software with multicloud edge and datacenter connectivity and catalog integration purchased	Data fabric software with multicloud edge and datacenter connectivity and catalog integration purchased	Data fabric software with multicloud edge and datacenter connectivity and catalog integration purchased
No metadata lineage	Metadata lineage available in data catalog on trusted assets being developed by using fabric	Metadata lineage available in data catalog on trusted assets being developed by using fabric	Metadata lineage available in data catalog on trusted assets being developed by using fabric
No data stewardship tools	Data stewardship tools available as part of the data fabric software	Data stewardship tools available as part of the data fabric software	Data stewardship tools available as part of the data fabric software
No data access security tool	Data access security in multiple technologies	Data access security in multiple technologies	Data access security enforced in all systems
No data privacy enforcement software	No data privacy enforcement software	Data privacy enforcement software in some database management systems	Data privacy enforcement software in all data stores
No MDM system	Single entity MDM system	Multientity MDM system	Multientity MDM system

## Data governance maturity summary

After you determine where you currently stand in the governance maturity model, meet with your key stakeholders to map out a strategy to increase your maturity. Start by defining your requirements, technology, data quality, metadata, data sharing, and master data strategy.

## Next steps

## Feedback

Was this page helpful?

 Yes

 No

# Requirements for governing data

Article • 11/27/2024

Cloud-scale analytics recommends you consider the following requirements for governing data:

- Data entity definition to create a common business vocabulary in a business glossary. *Data entities in this context mean concepts like customer, supplier, materials, employee, and others.*
- Data entity identification and discovery.
- Data classification to govern data access security, data privacy, and data retention.
- People, such as data owners with governance accountability and data stewards responsible for data protection and quality.
- Data governance processes.
- Data lifecycle management to govern how long data should be kept.
- Policies and rules to define how specific data should be governed throughout its lifecycle.
- Policy enforcement across data stores in the distributed data landscape.
- Master data management to make the data consistent across operational and analytical systems like customer, product, and supplier.
- Metadata lineage to understand the transformation and relationship of data entities.
- Technology to make it possible to govern structured, multi-structured, and unstructured data. The governance might span across the datacenter, multiple clouds, and the edge.

One challenge is that data is being collected and stored in multiple places across the enterprise. The data might include data collected and stored in different geographies and different legal jurisdictions. As a result, different legislation might apply to governing the same data in different jurisdictions. Discover data distributed across multiple clouds and geographic locations to:

- Understand what data attributes, data entities, and data relationships exist across the distributed data landscape.
- Classify the data to know how to govern it.
- Define policies to specify how data should be governed for each type of data classification and lifecycle management.
- Enforce data quality, data access security, data privacy, and lifecycle management policies across the distributed data landscape.

# Data classification

Data classification is a way of categorizing data assets by assigning unique logical tags or classes to the data assets. Classification is based on the business context of the data.

There needs to be a way to classify data to understand its level of confidentiality and how long to keep it. The classification requires:

- A data confidentiality classification scheme.
- A data retention classification scheme.

## Data confidentiality classification scheme

[+] Expand table

Classification	Description
Public	Anyone can access the data and it can be sent to anyone. For example, open government data.
Internal use only	Only employees can access the data and it can't be sent outside the company.
Confidential	The data can be shared only if it's needed for a specific task. The data can't be sent outside the company without a non-disclosure agreement.
Sensitive (personal data)	The data contains private information, which must be masked and shared only on a need-to-know basis for a limited time. The data can't be sent to unauthorized personnel or outside the company.
Restricted	The data can be shared only with named individuals who are accountable for its protection. For example, legal documents or trade secrets.

## Data lifecycle retention classification scheme

[+] Expand table

Retention	Description
None	Data can be deleted at any time.
Temporary	Keep data for a short period of time. For example, keep Twitter data for a week.
Fixed period	Keep data for a set number of years, after which it can be deleted. For example, keep tax records for seven years to comply with government laws.

Retention	Description
Permanent	Never delete data. For example, legal correspondence.

Automating the data confidentiality and data lifecycle retention classification process using the classes defined in each scheme is needed to consistently label data across the distributed data landscape. The automation enables it to be consistently and correctly governed. Then, define rules and policies for each class in the classification scheme to specify how to govern data according to its classification.

## Data governance roles and responsibilities

Another requirement is the need for accountability. Otherwise confusion lingers as to who is accountable for governing data. If there's no accountability, how do you answer the following questions?

- Who sets success metrics and monitors how well the data governance program is working?
- Who are the data owners?
- Who defines and maintains a business glossary?
- Who creates and maintains policies on access security?
- Who is protecting personal data privacy for compliance?
- Who is looking after the quality of product data across all brochures and partner websites?
- Who ensures customer data is consistent across all systems?
- Who is policing external subscription data usage versus the license?
- Who is policing privileged users like database administrators and data scientists?
- Is it a C-level executive? Is it a department head?
- Is it the head of governance, risk, and compliance?
- What about the legal department?
- Is it IT's responsibility?

Roles and responsibilities are needed to avoid confusion and to set the foundation upon which a data culture can materialize.

## Data governance processes

Processes are needed, along with roles and responsibilities, to:

- Govern the definition and maintenance of a common business vocabulary.
- Discover and identify what data you have, what it means, and where it's stored.

- Classify data to know how to govern it.
- Govern the definition and maintenance of data access security policies.
- Govern the definition and maintenance of data privacy policies.
- Detect data quality problems and remediate them.
- Apply policies to ensure action is taken for compliance.
- Govern the maintenance of master data.

## Data governance policies and rules

Define policies and rules to govern:

- Data integrity rules
- Data ingestion policies and rules
- Data access security policies and rules
- Data privacy policies and rules
- Data quality policies and rules
- Data maintenance policies and rules
- Data retention policies and rules

Associate these policies and rules with each class in the data governance classification schemes.

## Master data management

Another requirement in governing data is master data management. Master data is the most widely shared data in any organization and includes core data entities. Core data entities include customer, supplier, materials, employee, and asset. It also includes financial chart of accounts data that is found in different financial applications. Because master data is so widely shared, it's application agnostic. It's needed by both operational transaction processing applications and analytical systems. Keeping this data synchronized can resolve many data errors and process errors. So, maintaining it centrally via a common process and synchronizing every system that needs it's the ideal situation. Also, governance is needed over who is allowed to maintain it and where that maintenance needs to happen.

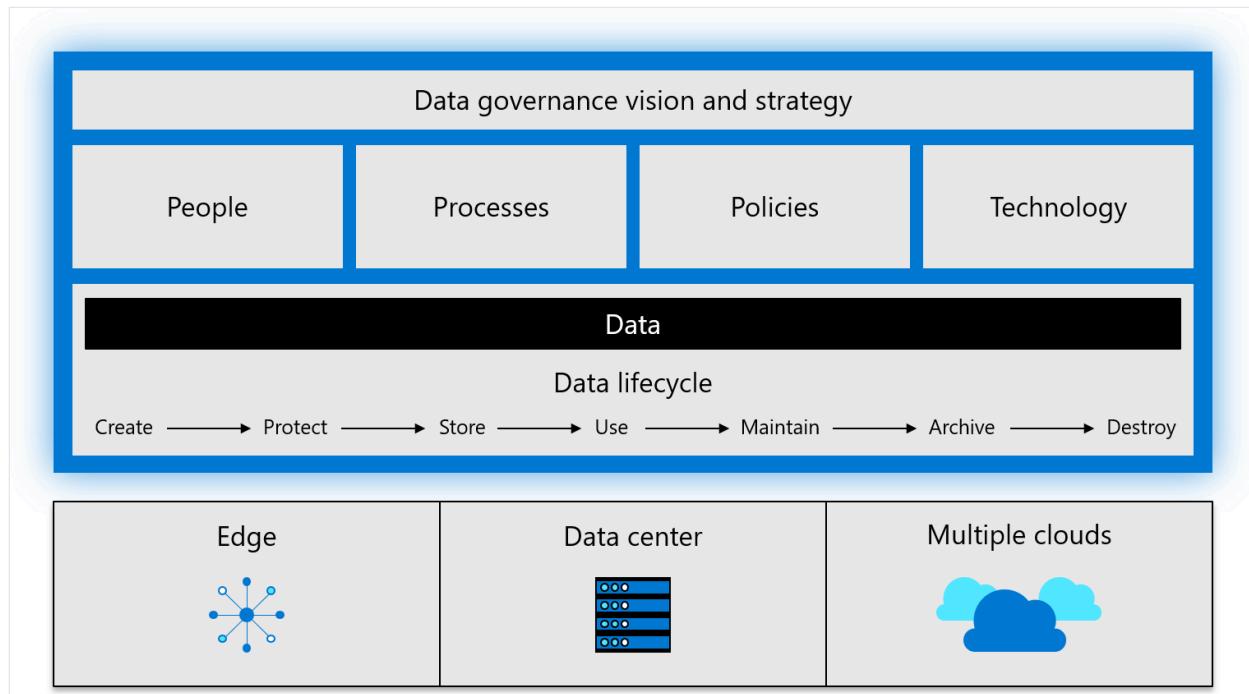
The same applies to reference data such as code sets and financial markets data. In this case, standardization and synchronization of code sets are known as reference data management, which is also a requirement.

## Metadata lineage

Finally, there's a requirement for metadata lineage. You can use an audit trail to know where data originated and how it's transformed en route to a report or a data store. You can use metadata to trace who or what is maintaining data, including when and where it occurs.

## Summary of what is needed for end-to-end data governance

You need an end-to-end solution that can govern data throughout its lifecycle across data stores in the edge, multiple clouds, and the datacenter.



Your data governance solution should have several components:

- A data governance vision and strategy
- The data itself, such as customer data, supplier data, order data, and others
- The data lifecycle from creation to destruction within which data needs to be governed
- Data governance roles and responsibilities (people)
- Data governance processes and activities and how they apply to the data lifecycle
- Policies and rules to govern data at different points in the lifecycle
- Data governance technologies to help make data governance possible

## Next steps

Data governance process

---

# Feedback

Was this page helpful?

 Yes

 No

# Data governance processes

Article • 11/27/2024

There are four categories of data governance processes.

  Expand table

Process category	Processes
<b>Data discovery processes, to understand the data landscape</b>	<p>A data and data entity discovery, mapping and cataloging process</p> <p>A data profiling discovery process to determine the quality of data</p> <p>A sensitive data discovery and governance classification process</p> <p>A data maintenance discovery process for CRUD analysis, such as from log files, to understand usage and maintenance of data such as master data across the enterprise</p>
<b>Data governance definition processes</b>	<p>Create and maintain a common business vocabulary in a business glossary define data entities, including master data, data attributes names, data integrity rules, and valid formats</p> <p>Define reference data to standardize code sets across the enterprise</p> <p>Define data governance classifications schemes to label data to determine how to govern it</p> <p>Define data governance policies and rules to govern data entity and document lifecycles</p> <p>Define success metrics and threshold</p>
<b>Data governance policy and rule enforcement processes</b>	<p>A process to automate application and enforcement of data governance policies and rules</p> <p>A process to manually apply and enforce policies and rules</p> <p>Event-driven, on-demand, and timer-driven (batch) data governance processes published as services that can be invoked to govern:</p> <p>Data ingestion - cataloging, classification, owner assignment, and storing</p> <p>Data quality</p> <p>Data access security</p> <p>Data privacy</p> <p>Data usage, for example, including sharing and to ensure licensed data is only used for approved purposes</p> <p>Data maintenance, such as master data</p> <p>Data retention</p> <p>Master data and reference data synchronization</p>
<b>Monitoring processes</b>	<p>Monitor and audit data usage activity, data quality, data access security, data privacy, data maintenance, and data retention</p> <p>Monitor policy rule violation detection and resolution</p>

The common business vocabulary should be defined in a business glossary within a data catalog.

Data governance working groups plan and develop defining data and improving specific data domains (for example, customer, or supplier); update the data governance control board on progress; and manage stewardship across the enterprise for a specific domain. Each working group should take responsibility to define a specific data entity or data subject area, such as multiple related entities. Multiple data entities in the vocabulary, along with the policies and rules, can then be worked on in parallel. For information, see [Data governance roles and responsibilities](#)

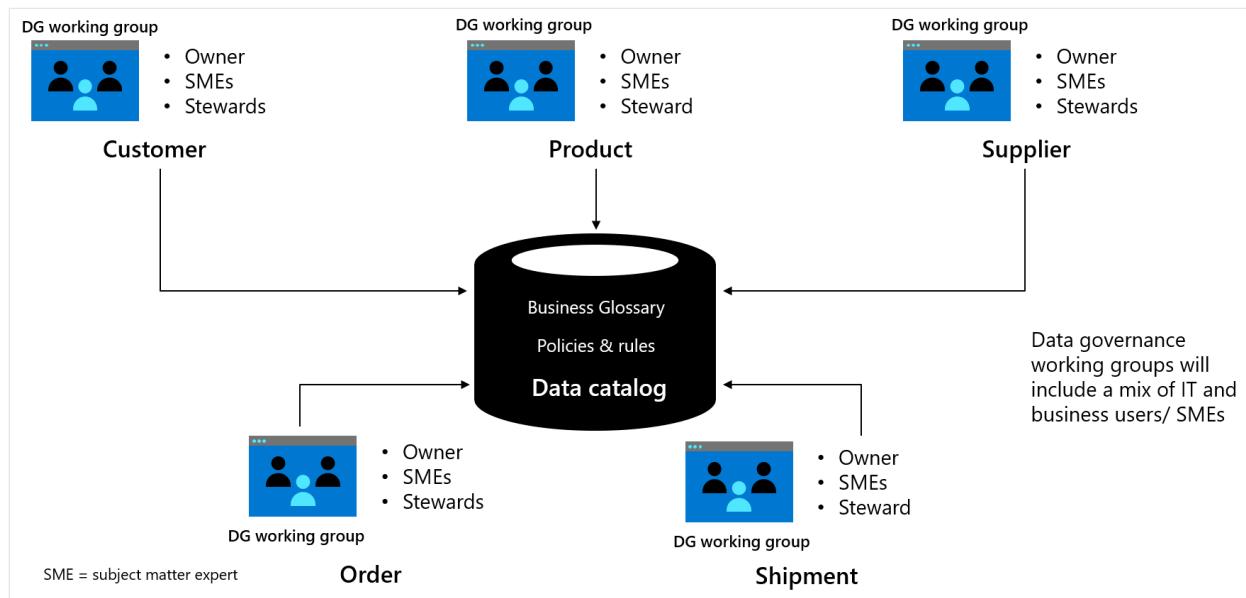


Figure 1: Example data governance working group

Integration of the catalog business glossary with other technologies is then needed to get consistent common data names into all technologies. Examples of other technologies to integrate with include:

- Extract, transform, load (ETL) tools
- Data modeling tools
- BI tools, database management systems
- Master data management
- Data virtualization tools
- Software development tools

A good practice for creating a common business vocabulary is to develop a data concept model. This model uses a top-down approach to identify data concepts that can be used as data entities in a common business vocabulary. Different data governance working groups can then be assigned to each data concept (entity) or group of related data concepts (subject area). These working groups are responsible for governing different data entities across the landscape.

When building a common business vocabulary, you can use data catalog software to automatically discover what data exists across multiple data stores. This software helps identify all the attributes associated with specific data entities, which is a bottom-up approach.

Multiple working groups can incrementally build a common business vocabulary quickly by combining the top-down approach of a data concept model with the bottom-up approach of automated data discovery.

Using a data catalog for automated data discovery enables the mapping of disparate data to a common vocabulary. The data catalog can help you understand where the data for each particular data entity in the business glossary is located across the enterprise.

## Policies and rules to govern data at different points in the lifecycle

Data governance policies describe a set of rules to control the integrity, quality, access security, privacy, and retention of data. There are different types of policy that include:

- Data integrity policies such as valid values, referential integrity.
- Data quality policies with data standardization, cleansing and matching rules.
- Data protection policies with access security and data privacy rules.
- Data retention policies to manage the lifecycle with retention, archive, and backup rules. Multiple versions of a policy might be needed to govern the same data across different legal jurisdictions.

The [data confidentiality classification scheme](#) has five classification levels:

- Public
- Internal use only
- Confidential
- Sensitive personal data
- Restricted

Govern data by combining this classification scheme with policies and rules. Use each of the five levels to label data, such as sensitive personal data. By creating rules for sensitive personal data, and attaching these rules to a policy, you create a policy for sensitive personal data. You can attach the policy to the sensitive personal data label and then attach the sensitive personal data label to the data. In this way, all data labeled as sensitive personal data is subject to the same policies and rules. This process is known as **tag-based policy management**. It's flexible because an individual rule or a

policy can be independently changed. All data labeled sensitive personal data is governed by the new rules. Equally, a sensitive personal data label can be detached from data and a confidential label used instead. In this case, the data instantly becomes governed by a new set of policies and rules associated with the confidential label.

After you define policies and rules in a data catalog for each class in a data governance classification scheme, they can be passed to other technologies from a data catalog, via APIs, for them to enforce. Instead, a common data management platform that can connect to multiple data stores could potentially enforce them.

It should then be possible to monitor data quality, privacy, access security, usage, maintenance, and retention of specific data entities throughout their lifecycle.

## Next steps

[Data Catalog](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Data catalog

Article • 12/10/2024

The data catalog registers and maintains the data information in a centralized place and makes it available for the organization. It ensures that enterprises avoid duplicate data products caused by redundant data ingestion by different project teams. We recommend that you create a data catalog service to define the metadata of the data products stored across the data landing zones.

Cloud-scale analytics depends on [Microsoft Purview](#) to register enterprise data sources, classify them, ensure data quality, and offer secure, self-service access.

Microsoft Purview is a tenant based service and can communicate with each data landing zone by creating a Managed Virtual Network deployed to the region of your data landing zones. You can deploy Azure Managed Virtual Network Integration Runtimes (IR) within Microsoft Purview Managed Virtual Networks in any available Microsoft Purview region. From there, the managed virtual network IR can use private endpoints to securely connect to and scan the supported data sources. For more information, see [Use Managed virtual network with your Microsoft Purview account](#). Creating a Managed virtual network IR within Managed Virtual Network ensures that data integration process is isolated and secure.

When using Azure Databricks, we recommend using [Azure Databricks Unity Catalog](#) in addition to Microsoft Purview. Azure Databricks Unity Catalog provides centralized access control, auditing, lineage, and data discovery capabilities across Databricks workspaces. For best practices for setting up Unity Catalog, see [Unity Catalog best practices](#).

## ⓘ Note

Although this documentation focuses primarily on using Microsoft Purview for governance, enterprises might have invested in other products, such as Alation, Okera, or Collibra. These solutions are subscription based and we would recommend deploying them to the data management landing zone. Be aware that some custom integration might be required.

# Data discovery

Data discovery reflects the state of all the data that the enterprise owns. This data is known as the data estate. During data discovery, the data estate is scanned and

classified. The data scanning process connects directly to the data source according to a set schedule.

As you add a new data landing zone to the environment, the associated data lakes and polyglot persistence sources must be registered as sources for the data catalog crawlers to scan.

With automated discovery of your data estate to populate the catalog, you can:

- Crawl metadata from Azure and on-premises data sources
- Scan your data lakes, blobs, and other supported targets
- Extract schema from your data targets for XML, TSV, CSV, PSV, SSV, JSON, Parquet, Avro, and ORC file types
- Allow automated catalog updates through configurable scheduling of scans and scan rule sets

 **Important**

When you add a new data landing zone to the environment, register the associated data lakes and polyglot storage through Azure DevOps as a source for the data catalog crawlers to scan, govern, and manage data integrity.

## Data classification

Microsoft Purview allows you to apply system or custom data classifications on file, table, or column assets.

Data classifications are like subject tags. Microsoft Purview marks and identifies the content of specific data types found within your data estate during scanning. You use sensitivity labels to identify the categories of classification types within your organizational data. You can also use sensitivity labels to group the policies you wish to apply to each category. Microsoft Purview makes use of the same sensitive information types as Microsoft 365, allowing you to extend your existing security policies and protections across your entire content and data estate.

Microsoft Purview can scan and automatically classify documents. For example, if you have a file named `multiple.docx` and it has a national ID number in its content, Microsoft Purview adds a classification such as `EU National Identification Number` in the asset detail page.

[Microsoft Defender for SQL](#) is a feature available for Azure SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytics. It includes functionality for discovering

and classifying sensitive data, surfacing and mitigating potential database vulnerabilities, and detecting anomalous activities that could indicate a threat to your database. Microsoft Defender for SQL provides a single goto location for enabling and managing these capabilities.

## Next steps

[Data lineage](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Data lineage

Article • 12/10/2024

Data lineage plays an important role in cloud-scale analytics. Lineage shows dependencies between raw data and finished products, describing the transformations and manipulations that turn raw data into final data products. Data lineage spans the lifecycle of data, from its origin to its movement across the data estate. It's used for troubleshooting, root cause analysis, data quality analysis, compliance, and impact analysis. It also adds context to datasets and products that enable data products to be discoverable and self-serviceable.

A primary feature of any data catalog is its ability to show the lineage between data products.

Microsoft Purview Data Catalog connects with various data processing, storage, and analytics systems to extract lineage information. The goal is to represent the movement, transformation, and operational metadata from each data system.

Azure Data Factory and Azure Synapse pipelines are recommended for ingestion solutions because they enable data lineage in Microsoft Purview. Alternate ingestion patterns should use the Apache Atlas API to update data lineage as part of their data processing.

Microsoft Fabric supports lineage without requiring Microsoft Purview. If you require one place to view lineage, then we recommend setting Microsoft Purview to scan a Microsoft Fabric tenant as this setting automatically brings in metadata and lineage from Fabric items, including Power BI, into Microsoft Purview Data Catalog. For more information, see [Lineage in Fabric](#) and [How to get lineage from Microsoft Fabric items into Microsoft Purview](#).

## 💡 Tip

For more information on supported systems and best practices, see [Data Lineage in Microsoft Purview](#).

## Next steps

Learn how to manage master data in Azure.

[Master data management](#)

---

# Feedback

Was this page helpful?

 Yes

 No

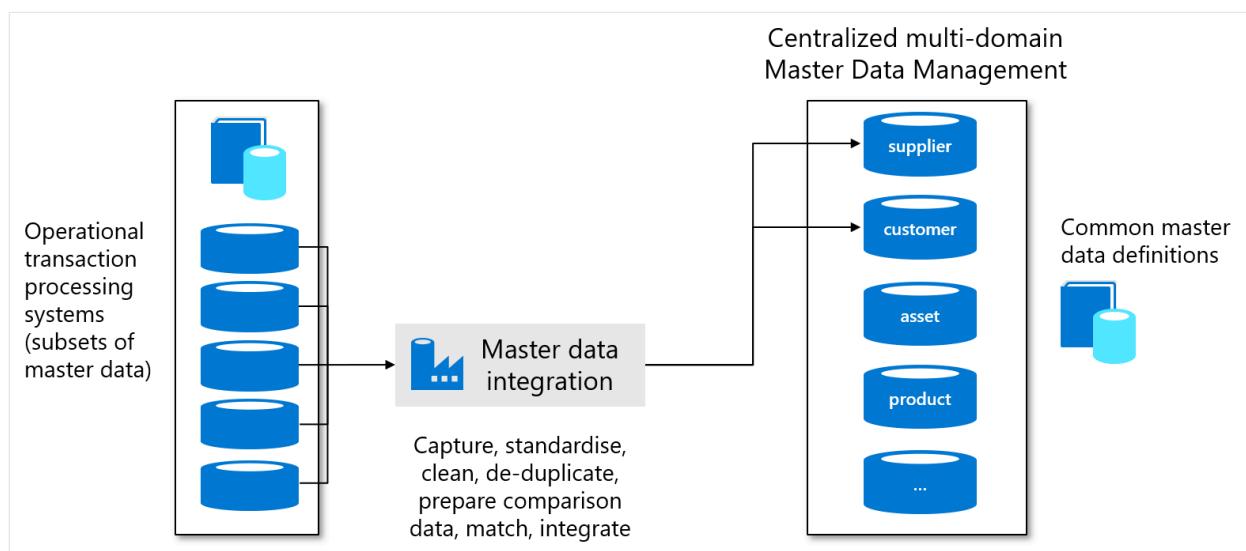
# Master data management

Article • 11/27/2024

Master data describes the objects around a business process. Customer, product, and other reference data are all master data objects. Master data isn't frequently altered, and though it's used to describe transactions, it isn't transactional in nature.

Master data management (MDM) is central to any data governance program, making the creation of trusted master data critical. Create master data by defining master data entities in your data catalog's business glossary. You can register data sources in your data catalog and search multiple data stores across the distributed data landscape to discover where various master data is located.

You can map the physical data names of discovered master data to your common business vocabulary in Microsoft Purview. You can also clean, match, and integrate the data you discovered across the distributed data landscape and use it to create golden master data records stored in a central MDM system.



After master data has been created and centrally stored, it can be synchronized with all systems that use master data to ensure their consistency.

It's important to govern your master data maintenance. Identify where maintenance takes place, noting which tasks from which business processes are involved. You can use business process identification and create, read, update, or delete (CRUD) analysis to identify these tasks. After you identify the tasks that maintain master data, you can then govern the data. Working out this data governance is often a manual task, but process mining and database log file analysis can aid you in the process.

## Master data partner solutions

Microsoft has partnered with partners to provide native integrations for Microsoft Purview. Assess these products for your organization's master data requirements.

- Microsoft Purview and Profisee integration for master data management
- Microsoft Purview and CluedIn integration for master data management
- Master Data Management with Semarchy
- Reltio Integration for Microsoft Purview

## Alternative solutions

Outside of a purpose-built MDM application, some of the technical capabilities needed to build an MDM solution can be found within the Azure ecosystem.

- **Data quality:** You can build data quality into your integration processes when loading to an analytics platform. For example, you might apply data quality transformations in an [Azure Data Factory](#) pipeline using hardcoded scripts.
- **Data standardization and enrichment:** [Azure Maps](#) is available to provide data verification and standardization for address data, which can be used in Azure Functions and/or Azure Data Factory. The standardization of other data might require the development of hardcoded scripts.
- **Duplicate data management:** You can use Azure Data Factory to [deduplicate rows](#) where sufficient identifiers are available for an exact match. Custom hardcoded scripts are likely required by the logic to merge matched data with appropriate survivorship.
- **Data stewardship:** [Power Apps](#) can be used to quickly develop simple data stewardship solutions for managing data in Azure. These solutions contain appropriate user interfaces for review, workflow, alerts, and validations.

## Next steps

[Data quality](#)

## Feedback

Was this page helpful?

 Yes

 No

# Data Quality

Article • 12/10/2024

Data quality is a management function of cloud-scale analytics. It resides in the data management landing zone and is a core part of governance.

## Data quality considerations

Data quality is the responsibility of every individual who creates and consumes data products. Creators should adhere to the global and domain rules, while consumers should report data inconsistencies to the owning data domain via a feedback loop.

Because data quality affects all the data provided to the board, it should start at the top of the organization. The board should have insights into the quality of data provided to them.

However, being proactive still requires you to have data quality experts who can clean buckets of data that require remediation. Avoid pushing this work to a central team and instead target the data domain, with specific data knowledge, for cleansing the data.

## Data quality metrics

Data quality metrics are key to assessing and increasing the quality of your data products. At a global and domain level, you need to decide upon your quality metrics. At a minimum, we recommend the following metrics:

[\[+\] Expand table](#)

Metrics	Metrics Definitions
Completeness = % total of non-nulls + nonblanks	Measures data availability, fields in the dataset that aren't empty, and default values that were changed. For example, if a record includes 01/01/1900 as a date of birth, it's highly likely that the field was never populated.
Uniqueness = % of nonduplicate values	Measures distinct values in a given column compared to the number of rows in the table. For example, given four distinct color values (red, blue, yellow, and green) in a table with five rows, that field is 80% (or 4/5) unique.
Consistency = % of data having patterns	Measures compliance within a given column to its expected data type or format. For example, an email field containing formatted email addresses, or a name field with numeric values.

Metrics	Metrics Definitions
Validity = % of reference matching	Measures successful data matching to its domain reference set. For example, given a <i>country/region</i> field (complying with taxonomy values) in a transactional records system, the value of "US of A" isn't valid.
Accuracy = % of unaltered values	Measures successful reproduction of the intended values across multiple systems. For example, if an invoice itemizes a SKU and extended price that differs from the original order, the invoice line item is inaccurate.
Linkage = % of well-integrated data	Measures successful association to its companion reference details in another system. For example, if an invoice itemizes an incorrect SKU or product description, the invoice line item isn't linkable.

## Data profiling

Data profiling examines data products that are registered in the data catalog and collects statistics and information about that data. To provide summary and trend views about the data quality over time, store this data in your metadata repository against the data product.

Data profiles help users answer questions about data products, including:

- Can it be used to solve my business problem?
- Does the data conform to particular standards or patterns?
- What are some of the anomalies of the data source?
- What are the possible challenges of integrating this data into my application?

Users can view the data product profile by using a reporting dashboard within their data marketplace.

You can report on such items as:

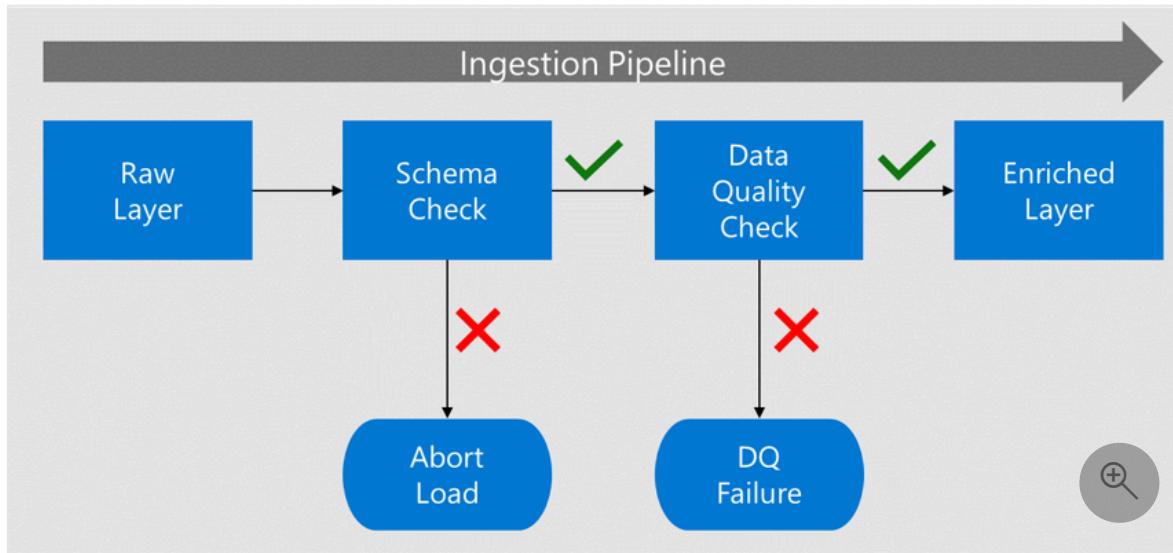
- Completeness: Indicates the percentage of data that isn't blank or null.
- Uniqueness: Indicates the percentage of data that isn't duplicated.
- Consistency: Indicates data where data integrity is maintained.

## Data quality recommendations

To implement data quality, you need to use both human and computational power as follows:

- Use solutions that include algorithms, rules, data profiling, and metrics.

- Use domain experts who can step in when there's a requirement to train an algorithm due to a high number of errors passing through the compute layer.
- Validate early. Traditional solutions apply data quality checks after extracting, transforming, and loading the data. By this time, the data product is already being consumed and errors surfaced to downstream data products. Instead, as data is ingested from the source, implement data quality checks near the sources and before downstream consumers use the data products. If there's batch ingestion from the data lake, do these checks when you move data from raw to enriched.



- Before data is moved to the enriched layer, its schema and columns are checked against the metadata registered in the data catalog.
- If the data contains errors, the load is stopped, and the data application team is notified of the failure.
- If the schema and column checks pass, the data is loaded into the enriched layers with conformed data types.
- Before you move to the enriched layer, a data quality process checks for compliance against the algorithms and rules.

### 💡 Tip

Define data quality rules at both a global and domain level. Doing so allows the business to define its standards for every created data product and allows data domains to create additional rules related to their domain.

## Data quality solutions

We recommend evaluating Microsoft Purview Data Quality as a solution for assessing and managing data quality, which is crucial for reliable AI-driven insights and decision-making. It includes:

- No-code/low-code rules: Evaluate data quality using out-of-the-box, AI-generated rules.
- AI-powered data profiling: Recommends columns for profiling and allows human intervention for refinement.
- Data quality scoring: Provides scores for data assets, data products, and governance domains.
- Data quality alerts: Notifies data owners of quality issues.

For more information, see [What is Data Quality](#).

If your organization decides to implement Azure Databricks to manipulate data, then you should assess the data quality controls, testing, monitoring, and enforcement that this solution offers. Using [expectations](#) can capture data quality issues at ingestion before they affect related child data products. For more information, see [Establish data quality standards](#) and [Data Quality Management With Databricks](#).

You can also choose from partners, open-source, and custom options for a data quality solution.

## Data quality summary

Fixing data quality can have serious consequences for a business. It can lead to business units interpreting data products in different ways. This misinterpretation can prove costly to the business if decisions are based on data products with lower data quality. Fixing data products with missing attributes can be an expensive task and could require full reloads of data from several periods.

Validate data quality early and put processes in place to proactively address poor data quality. For example, a data product can't be released to production until it achieves a certain amount of completeness.

You can use tooling as a free choice, but ensure it includes expectations (rules), data metrics, profiling, and the ability to secure the expectations so that you can implement global and domain-based expectations.

## Next steps

[Data lifecycle management?](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Data lifecycle management

Article • 12/10/2024

Data lifecycle management is the practice of using specific policies to effectively manage data for the entire time it exists within your system. These policies should consist of overarching storage and data policies that drive your data management processes. Since business goals and drivers dictate data lifecycle management policies, they generally tie into a framework of overall IT governance, management, and service level agreements (SLAs).

After you know what type of data you have and how it will be used, you already know its most likely evolution and destiny. You need to understand how your data evolves, determine how it grows, monitor changes in its usage over time, decide how long it should exist, and adhere to all rules and regulations that apply to that data.

Data lifecycle management addresses these needs using a combination of processes, policies, and software so that your teams can use appropriate technology for each phase of your data's lifecycle.

## Lifecycle of data

Data products can have different lifecycles. In a typical lifecycle pattern, newly ingested data gets used and accessed often. As its age increases, its rate of access often decreases, and older data sees a drastic drop in usage. Some data products might expire days or months after their creation, while other data products are actively used and modified across their entire lifetimes.

Data lifecycles can differ from this typical pattern, though. Some data remains unused after its initial ingestion or is rarely accessed after it's been stored. Most places have regulations that dictate how long you're required to store data, such as personal data and accounting data. A particular country/region might require you to retain primary documentation for five years for data such as incoming and outgoing invoices, cash book balances, bank vouchers, and salary slips. It also might require secondary documentation to be retained for three to five years, which includes things like letters, agreements, and notes.

## Managing data lifecycles

There are two ways to approach data lifecycle management in cloud-scale analytics:

- You can use the inbuilt data lifecycle features of each Azure service containing persisted data, such as [Azure Data Lake](#). This method is good for moving data to cold and archive tiers but fails to ensure data is deleted after a specified amount of time.

 **Important**

The archive tier is not currently supported for zone-redundant storage accounts. For more information, see [data redundancy](#).

- You can integrate data lifecycle in an onboarding process, which gives application business owners an opportunity to define their data lifecycle policy. This process involves a custom application to capture key metrics into [metadata standards](#) for each data product. Part of this method involves moving data from hot to cold to archive and ensures the deletion of data after a specified amount of time.

## Next steps

[Metadata standards](#)

## Feedback

Was this page helpful?

 Yes

 No

# Metadata standards

Article • 11/27/2024

Metadata Management plays a crucial role in data architecture. Metadata is data about other data. It describes data, providing a reference that helps you to find, secure, and control data. Metadata also binds data together. It can be used to validate data's integrity and quality, route or replicate data to a new location, transform data, and understand data meanings. Metadata is also essential in democratizing data through self-service portals.

There's a growing trend in the industry to bring data insights closer to data analysts and scientists using portals that use more metadata. This trend is known as *data observability*. Data observability uses concepts like metadata lake, knowledge graphs, or metadata graphs to describe platforms where metadata is centralized. It's a good way to build a unified view of how data is used and sourced across your organization when using a distributed data mesh.

A good metadata management strategy grows organically. It starts simple and small by first identifying the most important areas. A good metadata management strategy is also supported with services and clear processes. To get started, it is good to be aware of the different metadata categories:

- **Business metadata** describes all aspects used for governance, finding & understanding data. Some well-known examples include business terms and definitions and information on data ownership, usage, and origination.
- **Technical metadata** describes the structural aspects of data at design time. Some well-known examples include schema information, data format and protocol information, and encryption and decryption keys.
- **Operational metadata** describes processing aspects of data at run time. Some well-known examples include process information, execution time, process failure information, and job IDs.
- **Social metadata** describes the user perspective of the data from its consumers. Some well-known examples include use and user tracking information, search result data, filters and clicks, viewing time, profile hits, and comments.

In decentralized data architecture, metadata management is an organizational challenge that requires finding a balance between centrally managed metadata and federated managed metadata. It's important to understand teams and functions for cloud-scale analytics in Azure as you plan your metadata management. Using a collaborative data management practice can improve communication, integration, and data flow automation between your teams. You can address some of the metadata management

complexity by striking the right balance between central governance and domain ownership.

As you're deciding what metadata to manage centrally or federate to your data domains and begin your implementation, ask yourself:

- What business metadata is critical?
- What technical metadata is required for interoperability?
- What processes and streams capture the data?
- Where are the models or schemas created and maintained?
- What information do teams need to deliver centrally to allow the data governance department to do its work correctly?

Using your answers to these questions, map out the content life cycle for each of your metadata streams, and determine all dependencies. You then have a metadata model that can connect business domains, processes, technology, and data.

After you know which metadata you need, you must choose a place to store and process it. You can use Microsoft Purview for this.

## Use Microsoft Purview to manage your data estate at large

[Microsoft Purview](#) is a unified data governance solution that helps you manage and govern your on-premises, multicloud, and software-as-a-service (SaaS) data. It manages metadata at scale because it is a fully automated service that intelligently performs data discovery, data scanning, data quality, and access management. It also provides a holistic map with many insights about your data mesh architecture.

Microsoft Purview is a comprehensive set of solutions that can help your organization govern, protect, and manage data, wherever it lives. Microsoft Purview solutions provide integrated coverage and help address the fragmentation of data across organizations, lack of visibility that hampers data protection and governance, and the blurring of traditional IT management roles.

Microsoft Purview combines data governance and compliance solutions and services together into a unified platform to help your organization:

- Gain visibility into data across your organization
- Safeguard and manage sensitive data across its lifecycle wherever it lives
- Govern data seamlessly in new, comprehensive ways
- Manage critical data risks and regulatory requirements

When implementing Microsoft Purview, avoid introducing too much change and complexity quickly. Technical metadata forms the foundation of Microsoft Purview. You need to gather and organize your metadata before making sense of it.

After you have your metadata, start with the basics:

- Business terms
- Lists of authoritative data sources
- Lists of databases
- Governance domains
- Schema information
- Data ownership
- Data stewardship
- Security
- Data quality

Then scale by slowly involving more domain owners and data stewards and by adding more classifications and sensitivity labels. These additions improve the search experience and enable better data access management.

Microsoft Purview offers a feature called Governance domains, which establish boundaries for unified governance, ownership, and discovery of data products and business concepts within your domain-oriented architecture. For more information, see [Governance Domains in Microsoft Purview](#).

## Use Azure Cosmos DB to create a Knowledge Graph

A data insight solution must describe how data is used and the relationships between entities such as source data and data products, and between data products from one domain and dependent products from another domain. You can use a graph database or custom user interface to model these relations.

To build a unified view of your organization's data with a custom user experience, consider using Azure Cosmos DB. Azure Cosmos DB is a globally distributed, multi-model database service with NoSQL endpoints. It provides a graph database service via Azure Cosmos DB for Apache Gremlin, which can store massive graphs with billions of vertices and edges.

The end result of the Azure Cosmos DB architecture is an organization-wide graph that provides a unified view of all data in your organization with end-to-end context. The metadata lake is not only about storing information. It also actively organizes your

metadata as a graph by connecting it to other services and tools. This organized graph allows you to cross-correlate many subject areas, including:

- Domains
- Data quality
- Data usage
- Business capabilities
- Application functions
- Technical architecture information
- Operational events
- Organizational metadata
- Application ownership metadata
- Location information
- Application life cycle management information

## Next steps

[Secure cloud-scale analytics in Azure](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Secure cloud-scale analytics in Azure

Article • 10/18/2024

To limit security risk as much as possible while also providing access to do data analytics, use data governance. Data governance provides balance among operations, maintenance, and control. It follows the underlying principle of data lake solution architecture design, which uses infrastructure as code and security as code.

## Security principles

The focus of cloud-scale analytics is based on key management principles:

 Expand table

Principle	Description
<b>Single authoritative source of identity</b>	Use consistency and a single authoritative source to increase clarity, and reduce the risk from human error and configuration and automation complexity.
<b>Automated approach to data security</b>	Use automation to enable auditing, implement multiple control points, and reduce human errors. Automation also makes data governance easier and limits overhead.
<b>Grant least privilege required to complete task</b>	Grant only the amount of access to users that they need to do their jobs and limit the allowed actions for a particular scope.
<b>Simplified yet secure permissions</b>	Avoid customization. Customization leads to complexity, which inhibits human understanding, security, automation, and governance. For example, use built-in roles to assign permissions to data services and avoid permissions that specifically reference individual resources or users.
<b>Better clarity and enforceability of rules and definitions</b>	Clearly separate data to help keep the environment organized, while making it easy to enforce security rules and definitions.

### Tip

When deploying cloud-scale analytics, use automation principles to enable security instead of applying them manually. Ideally, you should only manually interact to approve or deny access requests.

# Next steps

[Authentication for cloud-scale analytics in Azure](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Authentication for cloud-scale analytics in Azure

Article • 02/10/2025

Authentication is the process of verifying the identity of a user or application. We recommend that you use a single-source identity provider to handle identity management and authentication. This provider is known as a *directory service*. It provides ways to store directory data and makes this data available to network users and administrators.

Any data lake solution should use and integrate with an existing directory service. For most organizations, the directory service for all identity-related services is Microsoft Entra ID. It's the primary and centralized database for all service and user accounts.

In the cloud, Microsoft Entra ID is a centralized identity provider and the preferred source for identity management. Delegate authentication and authorization to Microsoft Entra ID to use capabilities such as conditional access policies that require a user to be in a specific location. Microsoft Entra ID also supports multifactor authentication, which increases the level of access security. You should configure data services by integrating Microsoft Entra ID whenever possible.

If your data services don't support Microsoft Entra ID, you should perform authentication by using an access key or token. You should store the access key in a key management store, such as Azure Key Vault.

Authentication scenarios for cloud-scale analytics are:

- **User authentication.** In this scenario, Microsoft Entra ID authenticates users by using their credentials.
- **Service-to-service authentication.** In this scenario, Azure resources authenticate services by using managed identities, which Azure automatically manages.
- **Application-to-service authentication.** In this scenario, applications authenticate services by using service principals.

## Authentication scenarios

The following sections describe each of the authentication scenarios: user authentication, service-to-service authentication, and application-to-service authentication.

## User authentication

Users who connect to a data service or resource must present a credential. This credential proves that users are who they claim to be. Then they can access the service or resource. Authentication also allows the service to know the identity of the users. The service decides what a user can see and do after the identity is verified.

Azure Data Lake Storage, Azure SQL Database, Azure Synapse Analytics, and Azure Databricks support Microsoft Entra ID integration. The interactive user authentication mode requires users to provide credentials in a dialog box.

 **Important**

Don't hard-code user credentials into an application for authentication purposes.

## Service-to-service authentication

When a service accesses another service without human interaction, it must present a valid identity. This identity proves the service's authenticity and allows the service that it accesses to determine what actions are permitted.

In service-to-service authentication scenarios, we recommend that you use managed identities for authenticating Azure services. Managed identities for Azure resources allow for authentication to any service that supports Microsoft Entra authentication without any explicit credentials. For more information, see [What are managed identities for Azure resources](#).

Managed identities are service principals that can only be used with Azure resources. For example, you can create a managed identity for an Azure Data Factory instance.

Microsoft Entra ID registers this managed identity as an object that represents the Data Factory instance. You can then use this identity to authenticate to any service, such as Data Lake Storage, without any credentials in the code. Azure manages the credentials that the service instance uses. The identity can authenticate Azure service resources, such as a folder in Data Lake Storage. When you delete the Data Factory instance, Azure deletes the identity in Microsoft Entra ID.

## Benefits of using managed identities

Use managed identities to authenticate an Azure service to another Azure service or resource. Managed identities provide the following benefits:

- A managed identity represents the service for which it's created. It doesn't represent an interactive user.
- Managed identity credentials are maintained, managed, and stored in Microsoft Entra ID. There's no password for a user to keep.
- When you use managed identities, the client services don't use passwords.
- The system-assigned managed identity is deleted when the service instance is deleted.

These benefits mean that credentials are better protected and security compromise is less likely.

## Application-to-service authentication

Another access scenario is when an application, such as a mobile or web application, accesses an Azure service. The application must present its identity, which must then be verified.

An Azure service principal is the alternative option for applications and services that don't support managed identities to authenticate to Azure resources. A service principal is an identity that's created specifically for applications, hosted services, and automated tools to access Azure resources. The roles assigned to the service principal control its access. For security reasons, we recommend that you use service principals with automated tools or applications instead of allowing them to sign in with a user identity. For more information, see [Application and service principal objects in Microsoft Entra ID](#).

## Differences between managed identities and service principals

[ ] Expand table

Service principal	Managed identity
A security identity that you manually create in Microsoft Entra ID for applications, services, and tools that need to access specific Azure resources.	A special type of service principal. It's an automatic identity that's created when an Azure service is created.
Used by any application or service and isn't tied to a specific Azure service.	Represents an Azure service instance itself. It can't be used to represent other Azure services.
Has an independent lifecycle. You must delete it explicitly.	Is deleted automatically when the Azure service instance is deleted.
Password-based or certificate-based authentication.	No explicit password needs to be provided for authentication.

## Note

Both managed identities and service principals are created and maintained only in Microsoft Entra ID.

# Best practices for authentication in cloud-scale analytics

In cloud-scale analytics, implementing robust and secure authentication practices is paramount. Best practices for authentication apply to various layers of a solution, including databases, storage, and analytics services. By using Microsoft Entra ID, organizations can improve security by using features such as multifactor authentication and conditional access policies.

 Expand table

Layer	Service	Recommendation
Databases	- SQL Database - SQL Managed Instance  - Azure Synapse Analytics  - Azure Database for MySQL  - Azure Database for PostgreSQL	Use Microsoft Entra ID for authentication with databases such as <a href="#">Azure Database for PostgreSQL</a> , <a href="#">Azure SQL</a> , and <a href="#">Azure Database for MySQL</a> .
Storage	Data Lake Storage from	Use Microsoft Entra ID for authentication for security principals, such as user, group, and service principals or managed identities, with Data Lake Storage instead of a shared key or shared access signatures. This approach helps improve security because it supports multifactor authentication and conditional access policies.

Layer	Service	Recommendation
	Azure Databricks	
Analytics	Azure Databricks	Use the System for Cross-domain Identity Management to <a href="#">sync users and groups from Microsoft Entra ID</a> . To access Azure Databricks resources by using REST APIs, <a href="#">use OAuth with an Azure Databricks service principal</a> .

### Important

Giving Azure Databricks users direct storage-level access to Data Lake Storage bypasses Unity Catalog's permissions, audits, and security features, including access control and monitoring. To better secure and govern data, Unity Catalog should manage access to data that's stored in Data Lake Storage for Azure Databricks workspace users.

## Next step

[Authorization for cloud-scale analytics in Azure](#)

## Feedback

Was this page helpful?



# Authorization for cloud-scale analytics in Azure

Article • 02/05/2025

Authorization is the act of granting an authenticated party permission to perform an action. The key principle of *access control* is to give users only the amount of access that they need to do their jobs and to only allow certain actions at a particular scope. Role-based security corresponds to access control. Many organizations use role-based security to control access based on defined roles or job functions instead of individual users. Users are assigned one or more security roles, and each role is given authorized permissions to perform specific tasks.

Microsoft Entra ID is a centralized identity provider that grants authorization to access data services and storage for each user or for each application based on a Microsoft Entra identity.

## Data service authorization

Azure role-based access control (RBAC) and access-control lists (ACLs) play crucial roles in managing access and ensuring security. Azure RBAC and ACLs both require the user or application to have an identity in Microsoft Entra ID. In cloud-scale analytics, RBAC is effective for databases and Azure Data Lake Storage. ACLs are used primarily in Data Lake Storage to provide fine-grained access control at the file and directory levels. ACLs complement RBAC by providing more detailed permissions within the storage hierarchy.

Azure RBAC provides built-in roles like *Owner*, *Contributor*, and *Reader*, but you can also create custom roles for specific needs. The following built-in roles are fundamental for all Azure resource types, including Azure data services:

[+] Expand table

Role	Description
Owner	This role has full access to the resource and can manage everything about the resource, including the right to grant access to it.
Contributor	This role can manage the resource but can't grant access to it.
Reader	This role can view the resource and information, except for sensitive information like access keys or secrets, about the resource. They can't make any changes to the resource.

### Note

Some services have specific RBAC roles like *Storage Blob Data Contributor* or *Data Factory Contributor*, so you should use these roles for these services. RBAC is an additive model in which adding role assignments is an active permission. RBAC also supports *deny* assignments that take precedence over *role* assignments.

### Tip

When you plan an access control strategy, we recommend that you grant users only the amount of access that they need to perform their jobs. You should also only allow certain actions at a particular scope.

## Access control in Azure databases

RBAC in Azure databases revolves around roles, scopes, and permissions. Azure provides several built-in roles for database management. One of those roles is *SQL Server Contributor*, which enables the management of SQL servers and databases. Another role is *SQL DB Contributor*, which permits the management of SQL databases but not of the server itself. Additionally, you can create custom roles that have specific permissions to meet unique requirements.

You can assign roles at different scopes, including:

- At the subscription level, where roles apply to all resources within the subscription.
- At the resource group level, where roles apply to all resources within the specified resource group.
- At the resource level, where you can assign roles directly to individual databases or servers. This approach gives you precise control.

Permissions define the actions that a role can perform, such as read, write, delete, or security settings management. These permissions are grouped into roles to simplify management.

In **Azure SQL Database**, you can assign roles to users, groups, or applications to control access. For example, a database administrator might be assigned the *SQL Server Contributor* role to manage the server and databases. Roles like *SQL DB Contributor* allow users to create, update, and delete databases, while the *SQL Security Manager* role focuses on security configurations.

In Azure Cosmos DB, you can assign roles to manage access to Azure Cosmos DB accounts, databases, and containers. Built-in roles like *Cosmos DB Account Reader* and *Cosmos DB Account Contributor* provide varying levels of access.

In Azure Database for MySQL, Azure Database for PostgreSQL, and Azure Database for MariaDB, you can assign roles to manage database servers and individual databases. You can use roles like *Contributor* and *Reader* to control access.

For more information, see [Azure built-in roles for databases](#).

## Access control in Data Lake Storage

Azure RBAC lets you grant coarse-grained access, such as read or write access, to all storage account data. ACLs let you grant fine-grained access, such as write access to a specific directory or file.

In many scenarios, you can use RBAC and ACLs together to provide comprehensive access control in Data Lake Storage. You can use RBAC to manage high-level access to data, which helps ensure that only authorized users can access the service. Then you can apply ACLs within the storage account to control access to specific files and directories, which improves security.

Azure attribute-based access control builds on Azure RBAC by adding role assignment conditions based on attributes in the context of specific actions. It essentially allows you to refine RBAC role assignments by adding conditions. For example, you can grant read or write access to all data objects in a storage account that have a specific tag.

The following roles permit a security principal to access data in a storage account.

[+] Expand table

Role	Description
<b>Storage Blob Data Owner</b>	This role gives full access to blob storage containers and data. This access permits the security principal to set the owner of an item and to modify the ACLs of all items.
<b>Storage Blob Data Contributor</b>	This role gives read, write, and delete access to blob storage containers and blobs. This access doesn't permit the security principal to set the ownership of an item, but it can modify the ACL of items that the security principal owns.
<b>Storage Blob Data Reader</b>	This role can read and list blob storage containers and blobs.

Roles such as *Owner*, *Contributor*, *Reader*, and *Storage Account Contributor* permit a security principal to manage a storage account, but they don't provide access to the data within that account. However, these roles, excluding *Reader*, can obtain access to the storage keys, which can be used in various client tools to access the data. For more information, see [Access control model in Data Lake Storage](#).

## Access control in Azure Databricks

Azure Databricks provides access control systems for managing access within the Azure Databricks environment. These systems focus on securable objects and data governance. The three main access control systems within Azure Databricks are:

- **ACLs**, which you can use to configure permission to access workspace objects such as notebooks. For more information, see [Access control overview](#).
- **Account RBAC**, which you can use to configure permission to use account-level objects such as service principals and groups.
- **Unity Catalog**, which you can use to secure and govern data objects.

In addition to access control on objects, Azure Databricks provides built-in roles on the platform. You can assign roles to users, service principals, and groups. For more information, see [Admin roles and workspace entitlements](#).

## Best practices for authorization in cloud-scale analytics

This guide discusses the best practices for implementing RBAC in cloud-scale analytics environments. It includes general RBAC principles, database access control, and data lake access control best practices to help ensure secure and efficient resource management.

## General RBAC best practices for cloud-scale analytics

The following best practices can help you get started with RBAC:

- **Use RBAC roles for service management and operations, and use service-specific roles for data access and workload-specific tasks.** Use RBAC roles on Azure resources to grant permission to security principals that need to perform resource management and operations tasks. Security principals that need to access data within storage don't require an RBAC role on the resource because they don't need to manage it. Instead, grant permission directly to data objects. For example, grant

read access to a folder in Data Lake Storage, or grant contained database user and table permissions on a database in SQL Database.

- **Use built-in RBAC roles.** First, use the built-in RBAC Azure resource roles to manage services and assign operations roles to control access. Create and use custom roles for Azure resources only when built-in roles don't meet your specific needs.
- **Use groups to manage access.** Assign access to Microsoft Entra groups and manage group memberships for ongoing access management.
- **Consider subscription and resource group scopes.** In nonproduction environments, grant access at the resource group scope to separate service management and operations access needs instead of granting access to individual resources. This approach makes sense because, in nonproduction environments, developers and testers need to manage resources. For example, they might need to create an Azure Data Factory ingestion pipeline or a container in Data Lake Storage.

However, in production environments, you can grant access to individual resources for workload-specific tasks like data lake file system support and operations. This approach makes sense in production environments because users only need to use resources like viewing the status of a scheduled Data Factory ingestion pipeline or reading data files in Data Lake Storage.

- **Don't grant unnecessary access at the subscription scope.** The subscription scope covers all resources within the subscription.
- **Opt for least-privilege access.** Select the right and only role for the job.

## Database access control best practices

Implementing effective RBAC is crucial for maintaining security and manageability in your analytics environment. This section provides best practices for using Microsoft Entra groups and built-in roles and for avoiding direct user permissions to help ensure a streamlined and secure access management process.

Cloud-scale analytics environments typically contain multiple types of storage solutions, including PostgreSQL, MySQL, SQL Database, Azure SQL Managed Instance, and Azure Synapse Analytics.

- **Use Microsoft Entra groups instead of individual user accounts.** We recommend that you use Microsoft Entra groups to secure database objects instead of

individual Microsoft Entra user accounts. Use Microsoft Entra groups to authenticate users and protect database objects. Similar to the data lake pattern, you can use your data application onboarding to create these groups.

- **Use built-in roles to manage access.** Create custom roles only if you need to meet specific requirements or if built-in roles grant too many permissions.
- **Refrain from assigning permissions to individual users.** Use roles, like database or server roles, consistently instead. Roles help with reporting and troubleshooting permissions. Azure RBAC only supports permission assignment via roles.

 **Note**

Data applications can store sensitive data products in SQL Database, SQL Managed Instance, or Azure Synapse Analytics pools. For more information, see [Data privacy for cloud-scale analytics in Azure](#).

## Data Lake Storage access control best practices

In modern data environments, secure and efficient access control is paramount. Data Lake Storage provides robust mechanisms to manage access through ACLs. This section outlines the best practices for implementing RBAC in Data Lake Storage and applying ACLs, Microsoft Entra security groups, and the principle of least privilege to maintain a more secure and manageable data lake environment. Additionally, it highlights the importance of aligning ACLs with data partitioning schemes and using Unity Catalog for Azure Databricks users to help ensure comprehensive security and governance.

- **Use ACLs for fine-grained access control.** ACLs play an important role in defining access at a granular level. In Data Lake Storage, ACLs work with security principals to manage fine-grained access to files and directories.
- **Apply ACLs at the file and folder levels.** To control access to data in the data lake, we recommend that you use ACLs at the level of files and folders. Data Lake Storage also adopts an ACL model that's similar to the Portable Operating System Interface (POSIX). POSIX is a group of standards for operating systems. One standard defines a simple but powerful permission structure for accessing files and folders. POSIX is widely used for network file shares and Unix computers.
- **Use Microsoft Entra security groups as the assigned principal in an ACL entry.** Instead of directly assigning individual users or service principals, use this approach to add and remove users or service principals without the need to

reapply ACLs to an entire directory structure. You can just add or remove users and service principals from the appropriate Microsoft Entra security group.

- **Assign access to Microsoft Entra groups and manage membership of groups for ongoing access management.** For more information, see [Access control model in Data Lake Storage](#).
- **Apply the principle of least privilege to ACLs.** In most cases, users should only have **read** permission to the files and folders that they need in the data lake. Data users shouldn't have access to the storage account container.
- **Align ACLs with data partitioning schemes.** ACLs and data partition design must align to help ensure effective data access control. For more information, see [Data lake partitioning](#).
- **For Azure Databricks users, exclusively control access to data objects with Unity Catalog.** Granting direct storage-level access to external location storage in Data Lake Storage doesn't honor any permissions granted or audits maintained by Unity Catalog. Direct access bypasses auditing, lineage, and other security and monitoring features of Unity Catalog, including access control and permissions. Therefore, you shouldn't give Azure Databricks users direct storage-level access to Unity Catalog managed tables and volumes.

## Next step

[Secure cloud-scale analytics in Azure](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Data privacy for cloud-scale analytics in Azure

Article • 01/07/2025

Cloud-scale analytics help you determine the optimal data-access patterns that suit your requirements while safeguarding personal data at multiple levels. Personal data includes any information that can uniquely identify individuals, for example driver's license numbers, social security numbers, bank account details, passport numbers, and email addresses. Many regulations exist to protect user privacy.

To protect data privacy within a cloud environment such as Azure, you can create a data-confidentiality scheme that specifies data-access policies. These policies can define the underlying architecture that the data application resides on, define how to authorize data access, and specify what rows or columns users can access.

## Create a data-confidentiality classification scheme

Expand table

Classification	Description
Public	Anyone can access the data and it can be sent to anyone. For example, open government data.
Internal use only	Only employees can access the data and it can't be sent outside the company.
Confidential	The data can be shared only if it's needed for a specific task. The data can't be sent outside the company without a non-disclosure agreement.
Sensitive (personal data)	The data contains private information, which must be masked and shared only on a need-to-know basis for a limited time. The data can't be sent to unauthorized personnel or outside the company.
Restricted	The data can be shared only with named individuals who are accountable for its protection. For example, legal documents or trade secrets.

Before you ingest data, you must categorize the data as either *confidential* or *below or sensitive personal data*:

- Sort data into confidential or below if you don't need to restrict which columns and rows users can view.
- Sort data into sensitive personal data if you need to restrict which columns and rows users can view.

#### Important

A dataset can change from confidential-or-below to sensitive personal data when you combine data with other data products that previously had a lower classification. If you need persistent data, move it to a designated folder that aligns with its confidentiality level and the onboarding process.

## Create an Azure policy set

After you classify your data, you should align the classification with your industry policy requirements and internal company policies. You want to create an Azure policy set that governs what infrastructure can be deployed, the location where it can be deployed, and networking and encryption standards.

For regulated industries, you can use Microsoft [regulatory compliance policy initiatives](#) as a baseline for compliance frameworks.

Data classification follows the same rules for encryption, allowed infrastructure SKUs, and policy initiatives. So you can store all data within the same landing zone.

For restricted data, you should host data in a dedicated data landing zone under a management group where you can define a higher set of requirements for infrastructure. For example, you might define customer-managed keys for encryption or inbound or outbound restrictions for the landing zone.

#### Note

You can put sensitive personal data and confidential-or-below data in the same data landing zone but different storage accounts. But this practice might complicate the solution on the networking layer, for example with network security groups.

A deployed data governance solution should limit who can search for restricted data in the catalog. Consider implementing Microsoft Entra ID conditional access for all data assets and services. To enhance security, apply just-in-time access for restricted data.

# Consider encryption requirements

In addition to defining policies for locations and allowed Azure services, consider the encryption requirements for each data classification. Consider the requirements for the following areas:

- Key management
- Key storage
- Data-at-rest encryption
- Data-in-transit encryption
- Data-in-use encryption

For key management, you can use platform-managed or customer-managed encryption keys. For more information, see [Overview of key management in Azure](#) and [How to choose the right key management solution](#).

For more information about encryption options, see [Azure data encryption at rest](#) and [Data encryption models](#).

You can use the [Transport Layer Security \(TLS\)](#) protocol to protect data that travels between cloud services and customers. For more information, see [Encryption of data in transit](#).

If your scenario requires that data remains encrypted during use, the Azure Confidential Computing threat model helps minimize trust. It minimizes the possibility that cloud provider operators or other actors in the tenant's domain can access code and data during implementation.

For more information, see [Azure confidential computing products](#).

# Implement data governance

After you define the policies for the deployment of allowed Azure services, determine how to grant access to the data product.

If you have a data governance solution such as [Microsoft Purview](#) or [Azure Databricks Unity Catalog](#), you can create data assets or products for enriched and curated data lake layers. Ensure that you set the permissions within the data catalog to help secure those data objects.

Use Microsoft Purview to centrally manage, secure, and control the following areas:

- Access to data

- The data lifecycle
- Internal and external policies and regulations
- Data-sharing policies
- Identifying sensitive data
- Insights about protection and compliance
- Policies for data protection reporting

For more information about how to use Microsoft Purview to manage read or modify access, see [Concepts for Microsoft Purview data owner policies](#).

Whether you decide to implement Microsoft Purview or another data governance solution, use Microsoft Entra ID groups to apply policies to data products.

Use the data governance solution's REST API to onboard a new dataset. Your data application teams create data products and register them in the data governance solution to help identify sensitive data. The data governance solution imports the definition and denies all access to the data until your teams set up its access policies.

## Use data-protection patterns

To protect sensitive data, choose a data-protection pattern based on the data, services, and policies that you implement.

### Multiple copies

The pipeline for every data product that has a sensitive personal-data classification creates two copies. The pipeline classifies the first as confidential or below. This copy doesn't include the sensitive personal-data columns. It's created under the confidential-or-below folder for the data product. The other copy is created in the sensitive personal-data folder. This copy includes the sensitive data. Each folder is assigned a Microsoft Entra ID reader and a Microsoft Entra ID writer security group.

If you use Microsoft Purview, you can register both versions of the data product and use policies to help secure the data.

The multiple copies pattern separates sensitive personal data and confidential-or-below data. But if you grant a user access to sensitive personal data, they can query all rows. Your organization might need to consider other solutions that provide row-level security to filter rows.

### Row-level and column-level security

If you need to filter rows that users can view, you can move your data into a compute solution that uses row-level security.

To prevent re-engineering, select the appropriate Azure service or Microsoft Fabric solution for your particular use case. Different types of databases are designed for different purposes. For example, you shouldn't use an online transaction processing (OLTP) database for extensive analytics. And if you use an e-commerce application, you shouldn't use a solution that's tailored for big data analytics because it can't achieve the required millisecond response times.

If you implement solutions that support row-level security, your data application teams must create different Microsoft Entra ID groups and assign permissions based on the data's sensitivity.

In addition to row-level security, you can restrict access to certain columns. The following table shows an example of four Microsoft Entra ID groups that have read-only access:

[Expand table](#)

Group	Permission
DA-AMERICA-HRMANAGER-R	View North America HR personnel data asset <b>with</b> salary information.
DA-AMERICA-HRGGENERAL-R	View North America HR personnel data asset <b>without</b> salary information.
DA-EUROPE-HRMANAGER-R	View Europe HR personnel data asset <b>with</b> salary information.
DA-EUROPE-HRGGENERAL-R	View Europe HR personnel data asset <b>without</b> salary information.

The first level of restrictions support dynamic data masking, which hides sensitive data from users that don't have privileges. You can use a REST API to integrate this approach into a dataset's onboarding.

The second level of restrictions adds column-level security to restrict non-HR managers from viewing salaries. It also adds row-level security to restrict which rows European and North American team members can view.

## Column encryption

Dynamic data masking masks the data at the point of presentation, but some use cases require that the solution never has access to the plaintext data.

The *SQL Always Encrypted* feature enhances the security of sensitive data in SQL Server databases. SQL Always Encrypted helps ensure that sensitive data in SQL Server databases remains secure and protected from unauthorized access. This feature encrypts the data at rest and in transit, which helps maintain maximum data confidentiality and regulatory compliance. SQL Always Encrypted performs encryption and decryption operations on the client side. Integrate this feature to help safeguard your most valuable data assets.

## Next step

[Organize data operations team members](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Organize data operations team members

Article • 12/10/2024

Cloud-scale analytics architecture was designed with a set of core principles.

## Core principles

- **Self-service enablement:** Enable project teams to work on their own to allow agile development methods.
- **Governance:** Enforce guardrails across the Azure platform to ensure that project teams only see, change, and execute the functions within their permissions.
- **Streamlined deployments:** Ensure that common policies are available within the organization to help teams scale quickly and support teams with less experience in some core designs and artifacts.

## Roles and teams

Across Cloud-scale analytics, we recommend moving away from horizontally siloed teams to agile vertical cross-domain teams. Data operations teams focus on driving governance at the control plane, while data application teams focus on creating data-as-a-product. This differentiation requires organizational changes to a pattern more aligned with application development. For example, each application has a product owner who scopes out requirements and works with a cross-domain team to deliver a product. In this case, the product is data for consumption.

For more information, see [Understand the roles and teams for cloud-scale analytics in Azure](#)

## Deployment and operations

The deployment process and data operations (DataOps) model is an essential part that supports some of these core principles. The following guidelines are recommended for organizations to align with the principles:

- Use infrastructure as code.
- Deploy templates that cover core use cases within the company.

- Follow a deployment process that includes a strategy for GitHub forks and branches.
- Maintain a central repository and deploying data management landing zones.

Contributors with identifiable and individual skills should establish a platform group to centrally govern data platform infrastructure and build and deploy common data infrastructure pieces for the data management landing zone, plus various data landing zones. The platform group can also build, own, and provide agnostic technology that helps data application teams to capture, process, store, and maintain their data applications.

The platform group should present its services in a self-service manner, which can include tools for storing big data, versioning product data, organizing/implementing the data pipeline, de-identifying data, and more. These types of tools are key to minimizing bottlenecks in the workflow and reducing lead time for creating new data products.

The platform group should follow the best practices outlined in this section to achieve their objectives. Other data product teams should use the best practices in the forthcoming articles to test and automate their data.

For more information, see [DevOps automation for Cloud-scale analytics in Azure](#)

## Next steps

[Understand the teams for cloud-scale analytics in Azure](#)

---

## Feedback

Was this page helpful?

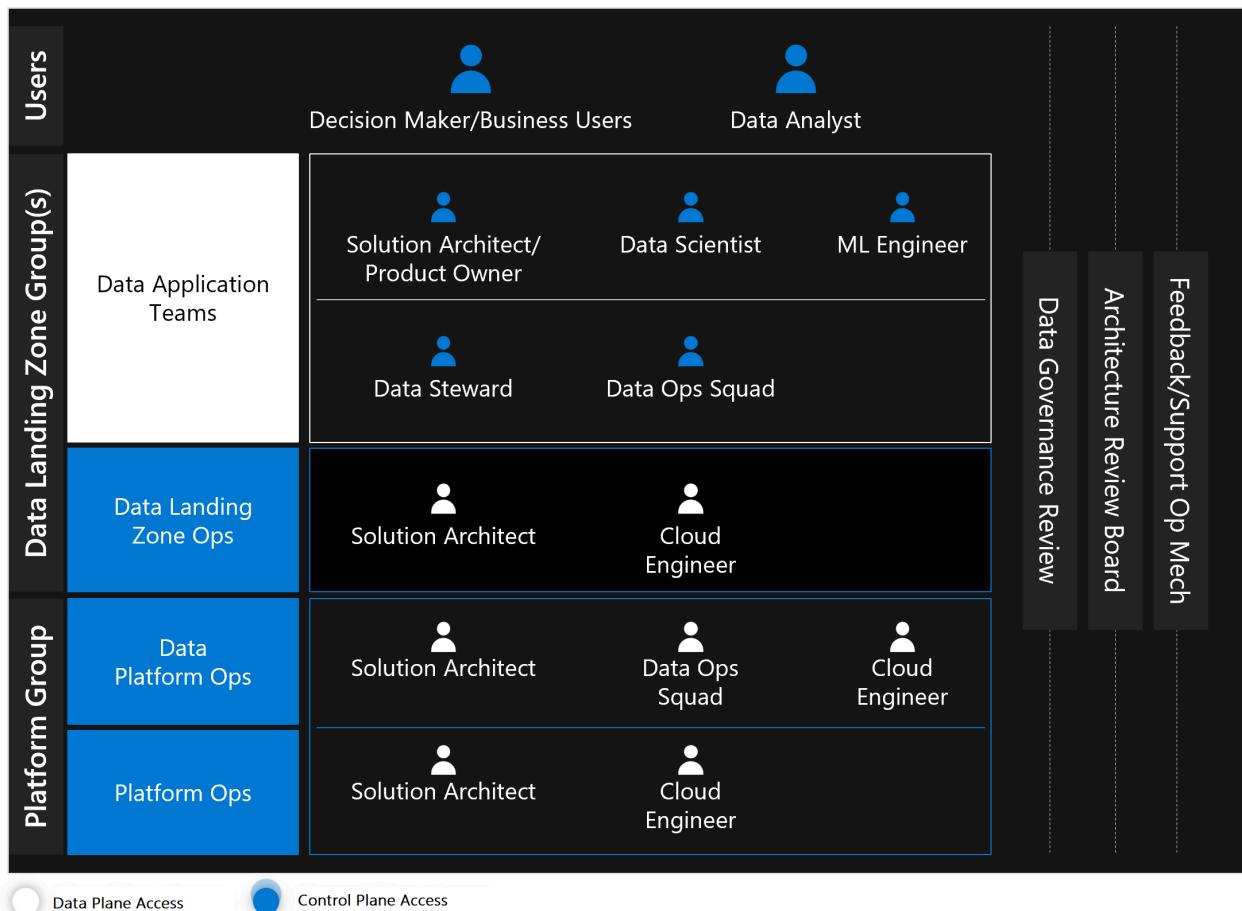
 Yes

 No

# Understand the teams for cloud-scale analytics in Azure

Article • 12/10/2024

For Cloud-scale analytics, we recommend moving teams like ingest, processing, analysis, consumption, and visualization from working in horizontally siloed teams, to agile vertical cross domain teams in each tier. Platform teams like data platform operations and platform operations are grouped together in a common platform group.



Within scale analytics, we identify the following teams:

- Platform ops
- Data platform ops
- Data landing zone ops
- Data Application teams

Each team focuses on a specific aspect of Cloud-scale analytics. For a comprehensive understanding of team functions, refer to the [Roles and Responsibilities](#) and [Understand teams and functions for cloud-scale analytics in Azure](#) guides.

# Data landing zone teams

The data landing zone group consists of three teams:

## Data Application teams (one team per application):

- Are responsible for delivering new data products such as insights, reports, notebooks, and applications.
- Partner closely with business analysts and business unit stakeholders.
- Transform data into new read data stores.
- Manage access hierarchy (groups) and approval requests.
- Furnish metadata in data catalogs.

## Data landing zone ops (one group per data landing zone):

- Operate and maintain their data landing zone instance.
- Respond to new data application service requests.

## Decide between a central or business data office

Depending on your organization's size and structure, a data landing zone group can be assembled in various ways. For instance, if you establish a data landing zone where the business already has its own data engineers, program managers, and solution architects, such as a business data office, you can provision the data landing zone. Then, you can allow the business data office to operate the data landing zone under the governance of your central platform group.

Another option is when a business doesn't have a data office to build out their data applications. In this scenario, the central data office can act as a consultancy, assigning staff to work on the data landing zone. These resources should be embedded within the business to collect and execute use cases using Scrum or agile methods. After the work is completed, they would return to the central data office.

Individuals should work within multidisciplinary teams in both scenarios, sharing goals and diverse experiences, skills, and approaches. This collaboration supports more effective outcomes than working in silos.

### Note

In the scenario where only one data landing zone is deployed, it's common for businesses to overlap in one data landing zone. This could create crossover functions where data application teams are sourced from central and business data

offices. However, data landing zone operations functions to be located in the central data office for this scenario.

## Teams within the platform group

The platform group consists of two teams:

### Data Platform Ops:

- Define common policies for data landing zone and data applications.
- Instantiate data landing zone scaffolding, including core services before passing it to data landing zone operations.
- Support stakeholders.

### Platform Operations:

- Operate and own the cloud platform.
- Instantiate data management landing zone and data landing zone scaffolding, including networking, peering, monitoring, and other core services.

## The digital security office

Digital security deals with the entire Cloud-scale analytics. It's usually a dedicated department lead by a chief information security officer. This department works closely with data platform ops, the data governance review board, and the architecture review board.

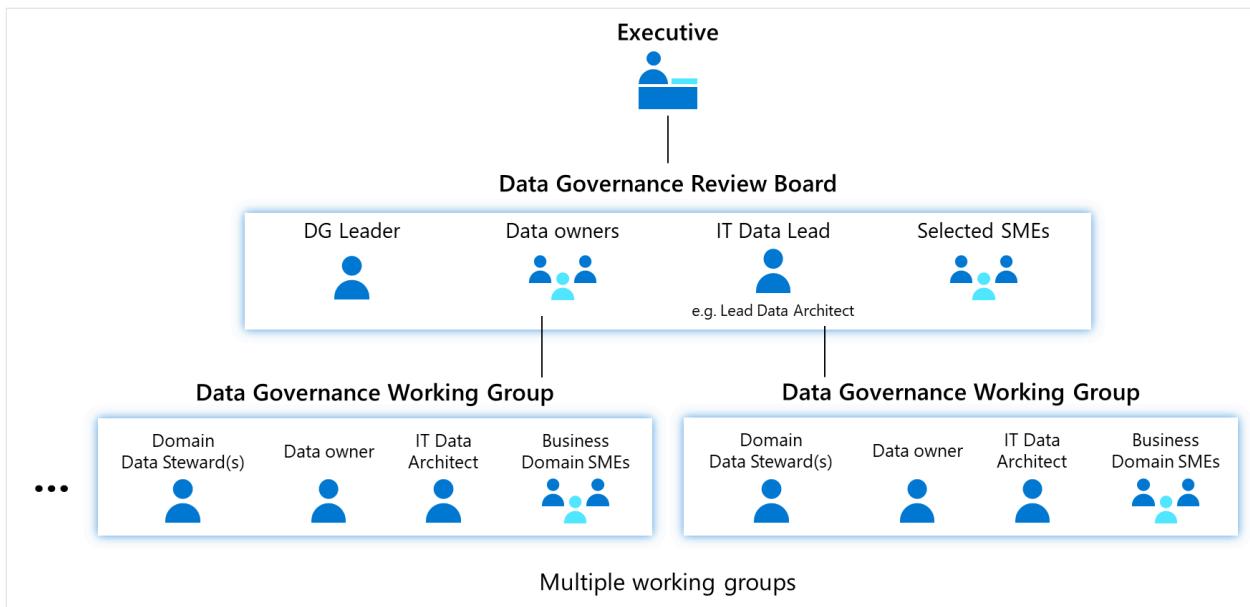
## Data governance roles and responsibilities

There are many data governance roles and responsibilities related to people. The roles and responsibilities can vary across organization. The roles and responsibilities in the table are provided as guidance only.

[ ] Expand table

Role	Responsibility
Executive sponsor, for example, the CFO or CIO.	This is a senior business stakeholder with authority and budget. They're accountable for establishing data governance.

Role	Responsibility
Data governance program leader, for example, the chief development officer, or the appointed lead.	This person is accountable and responsible for implementing the data governance program.
Data governance control board	This board includes a data governance lead and data owners. They establish metrics for success, own the data governance roadmap, select working groups, hold the budget for the data governance program, and mediate during conflicts about prioritizing and defining cross-functional data.
Data governance working group	They plan and develop defining data and improving specific data domains (for example, customer, or supplier); update the data governance control board on progress; and manage stewardship across the enterprise for a specific domain.
Data owner	This is a senior business stakeholder with authority and budget. They're accountable for quality assurance (QA) and protecting a specific data subject area or entity across the enterprise. They decide who can access and maintain that data and how it's used.
Business data steward	This business professional responsible oversees QA and protects a data subject area or entity. They're often experts in the data domain, work in a team with other data stewards across the enterprise, and monitor and decide how to maintain data quality.
Data protection officer	This is a senior business stakeholder with authority and budget, accountable for protecting personal data. Their tasks are specific to compliance legislation in all jurisdictions where company operates.
Data security team	This team is responsible and accountable for enforcing data access security and a data privacy policy.
Data publishing manager	This person is responsible and accountable for checking QA and publishing new, trusted data assets in a data marketplace for consumer use.



The goal is for businesses to organize governance in a way that enables them to effectively manage data throughout its lifecycle across a hybrid computing environment. One approach is to have multiple working groups reporting to a data governance control board, with each group responsible for a specific data domain or entity, such as customer data, or a data subject area that includes multiple data entities.

## Other groups

Companies can run several smaller teams with key stakeholders and subject matter experts across the entire operating model to maintain a centralized view of the analytics platform.

## Architecture review board

The architecture review board's main functions are to review and assess architectures, and create and maintain standards and references. The board consists of individuals who are experts in their field. Typically, the individuals are domain architects and other technical leaders invited to give opinions when needed.

## Feedback and support operating board

The feedback and support operating board receives feedback about processes and works with the other groups to create backlog items to address gaps and improve the solution.

## Next steps

## Feedback

Was this page helpful?

 Yes

 No

# Roles and Responsibilities

Article • 11/27/2024

After you've defined your cloud-scale analytics strategy, you need to organize teams to successfully deliver on it. This article describes some of the roles and responsibilities you should consider for cloud-scale analytics. You can map these roles and responsibilities to the various teams we've discussed in previous articles.

## ⓘ Important

This article highlights potential roles and responsibilities, but it isn't a complete list. Consider this article's guidance and then alter it for what works within your organization. If you're a small organization, you might not resource these roles, but that shouldn't prevent you from deploying a cloud-scale analytics platform. If you're a large organization, you might decide to streamline and consolidate roles.

## Roles

A cloud-scale analytics deployment involves multiple roles. The following table describes each role, job title, and responsibilities.

[\[+\] Expand table](#)

Role	Other Job Title	Responsibilities	Skills	Applies to:
Solution Architect	Platform Architect, Solution Architect	Design and oversight of cloud technologies to meet business/customer needs	Cloud technology subject matter experts, Architecture pattern development	Data application teams, platform group
Platform Ops	Cloud Engineer, Infrastructure Engineer, Systems Engineer	Assisting with cloud technology design, implementing and enabling cloud services and capabilities, and managing cloud resources	Cloud technology subject matter experts, programming, DevOps	Data landing zone ops, platform group
Security Architect		Security standards and policy design, security standards and policy oversight, security tool decisions, security assessments and audits, and security patterns and processes	Security technology subject matter experts, regulation, compliance, and legal controls, decision making, threat protection, and vulnerability management	Platform group
Security Engineer	Security Ops	Security policy, standards, and tooling implementation, and assist with security assessments and/or audits	Security technology subject matter expertise, regulation, compliance, and legal controls, threat detection, vulnerability management, and SOC processes	Platform group
DataOps Engineer	Data Platform Ops	Orchestrating the analytic pipeline, promoting features to production, and automating quality	Agile Development, DevOps, Statistical Process Control	Data application teams
Data Modeling Architect	Data Modeler	Data modeling, data mapping, and data patterns	Industry data standards, data tooling capabilities, lake database templates, and data governance	Data application teams
Data Solution Architect	Data Solution Architect	Data platform best practices application, data publishers/data product owner guidance, and data patterns	Industry data standards, data tooling capabilities, lake database templates, and data governance	Data application teams, platform group
Data Engineer	Data QA Engineer, ETL Engineer	Lake database templates/data lakes/warehouses/marts implementation, data movement, and data transformation	Databases, programming/scripting, simple storage, and data APIs	Data application teams
Data Owner	Source Data Owner, Technical Publisher	Access approvals, data quality, business term definition, usage rule definition, and specification of data in control file for onboarding data source	Domain subject matter expert and business relationships	Data application teams
Data Product Owner	Data Scientist Manager, Data Analyst Manager	Vision of data product, data product usage, and metric definitions	Business subject matter expert, business relationships, analytics concepts, user experience design, and product management	Data application teams

Role	Other Job Title	Responsibilities	Skills	Applies to:
Data Analyst	Data Visualization, Designer, Business Data Analyst, BI Developer, BI Engineer, Reporting Analyst	Visualization, charts, graphs, dashboards, tables, reports, and Exploratory Data Analysis	Programming/scripting/SQL, statistics, data cleaning, and data visualization	Consumers, data application teams
Data Scientist	Machine Learning Researcher, Machine Learning Engineer, Quantitative Analyst, AI Programmer	Algorithms, models, data product curation, Exploratory Data Analysis, measuring and improvement of results, and communicating findings	Business subject matter expert, advanced mathematics, machine learning, data mining tools, programming/scripting/SQL, statistics, and data visualization	Data application teams
Data Governance Manager	Data Governance Lead, Data Governance Sponsor	Data governance program oversight, data governance standards, policies and rules, and approval of tools to support governance capabilities	Governance regulations and control frameworks, business relationships, and business strategy alignment to objectives	Platform group, governance
Data Steward	Data Trustee	Data meaning, data quality, data compliance, fitness of data assets, knowledge of data products and their use, data team outreach, conceptual subject area ownership, subject area leadership with technical data owner, and data subject area stewardship	Domain subject matter expertise, data quality, governance regulations, and governance control frameworks	Data application teams, governance

## Responsibilities

A cloud-scale analytics deployment involves multiple areas of responsibility. The following tables provide overviews for each of these areas.

## Compute

[Expand table](#)

Areas of Responsibility	Data Owner	Data Product Owner	Data Analyst	Data Scientist	Data Governance Manager	Data Steward	Solution Architect	Platform Ops	Security Architect	Security Engineer	DataOps Engineer	Data Solution Architect	Data Model Architect
Manage the UI for requesting compute							A	R					
Specify what tools are used to bring compute to the data platform infrastructure as code (IaC) template							A	R					
Configure and monitor compute that accesses the data platform							A	R			R	R	R
Provide Producer support for compute that accesses the data platform							A				R		
Understand, monitor, and execute business rules & cleansing		A									R	R	R

Areas of Responsibility	Data Owner	Data Product Owner	Data Analyst	Data Scientist	Data Governance Manager	Data Steward	Solution Architect	Platform Ops	Security Architect	Security Engineer	DataOps Engineer	Data Solution Architect	Data Model Archit
for data curation													

## Data lifecycle

[Expand table](#)

Areas of Responsibility	Data Owner	Data Product Owner	Data Analyst	Data Scientist	Data Governance Manager	Data Steward	Solution Architect	Platform Ops	Security Architect	Security Engineer	DataOps Engineer	Data Solution Architect	Data Model Archit
Responsible for data model architecture for zones in the data platform								R			R	A	R
Drive architectural approval for overall continuity of the data platform											R	A	R
Own source data loaded to data platform	A					R							
Manage source data loaded to data platform		A				R							
Manage the ingestion service in the data platform												A	
Manage the handshake service in the data platform												A	
Manage archive data in raw zone (retention and archiving policies)		R			A								
Prepare data for ingestion to drop or landing to other zones	A												
Change management - changes to existing data sources		A				R							
Perform end to end testing and report results							R		R	A	R	R	R

Areas of Responsibility	Data Owner	Data Product Owner	Data Analyst	Data Scientist	Data Governance Manager	Data Steward	Solution Architect	Platform Ops	Security Architect	Security Engineer	DataOps Engineer	Data Solution Architect	Data Model Archit
Performs post testing bug fixes								R		R	A	R	R

## Data products and operations

[Expand table](#)

Areas of Responsibility	Data Owner	Data Product Owner	Data Analyst	Data Scientist	Data Governance Manager	Data Steward	Solution Architect	Platform Ops	Security Architect	Security Engineer	DataOps Engineer	Data Solution Architect	Data Model Archit
Manage backup and recovery								A	R				
Manage database performance and availability							A	R	R	R			
Manage ITSM processes for the data platform (Incident, Problem, Change, Request etc.)	R	R	R	R	R	R	R	R	R	R	R	R	
Oversee service level agreements for data platform services							A	R					
Support and management Azure subscriptions							A	R					
Understand relationships between organizations, subscriptions, licenses, user accounts, and tenants, to set a subscription model							A	R					
Review monthly Azure bill and understand usage and charges	R	R					A						
Manage costs and create showback and chargeback reports		R					A						
Cost management - make costing		R					A						

Areas of Responsibility	Data Owner	Data Product Owner	Data Analyst	Data Scientist	Data Governance Manager	Data Steward	Solution Architect	Platform Ops	Security Architect	Security Engineer	DataOps Engineer	Data Solution Architect	Data Model Archit
model able to define reporting and charges.													
Cost management - Use costs reports for trending and usage growth	R					A							
Perform code review to ensure robust integrity							R		R		A		
Build & deliver approved data solutions architecture							R		R		R		

# Drop Zone

 Expand table

Areas of Responsibility	Data Owner	Data Product Owner	Data Analyst	Data Scientist	Data Governance Manager	Data Steward	Solution Architect	Platform Ops	Security Architect	Security Engineer	DataOps Engineer	Data Solution Architect	Data Model Archit
Manage UI for Drop Zone provisioning							A	R					
Request a drop zone	A		R	R									
Approve drop zone provisioning	A			R									
Manage drop zone provisioning execution						A	R						

## General

Expand table

Areas of Responsibility	Data Owner	Data Product Owner	Data Analyst	Data Scientist	Data Governance Manager	Data Steward	Solution Architect	Platform Ops	Security Architect	Security Engineer	DataOps Engineer	Data Solution Architect	Data Model Archit
Infrastructure team support	R	R	R	R	R	R	A		R	R			
Incident management - Data platform related issues								R			A		
Define, communicate, and drive execution of data strategy and data governance strategy					A								

## Governance

[Expand table](#)

Areas of Responsibility	Data Owner	Data Product Owner	Data Analyst	Data Scientist	Data Governance Manager	Data Steward	Solution Architect	Platform Ops	Security Architect	Security Engineer	DataOps Engineer	Data Solution Architect	Data Model Archit
Owns partnership between data product owner/data owner to ensure best practices	R	R			A	R							
Main liaison with EA, cloud, and security on documented standards related to the data platform							R		R		A		R
Partner with data product owner/data owner for data integrity efforts	R	R			A	R							
Care for all data in the data platform	R	R			A	R							
Define metadata management standards related to the data platform (policies and rules about metadata creations and maintenance)	R	R			A								R
Ensure data quality and data	R	R	R	R	A	R							R

Areas of Responsibility	Data Owner	Data Product Owner	Data Analyst	Data Scientist	Data Governance Manager	Data Steward	Solution Architect	Platform Ops	Security Architect	Security Engineer	DataOps Engineer	Data Solution Architect	Data Model Architect
management standards in the data platform													
Ensure data security and compliance in the data platform	R	R			A	R			R	R			
Monitor and review data quality in the data platform	R	R	R	R	A	R							R
Measure and report data quality in the data platform to management	R	R			A	R							

# Lake database templates

 Expand table

Areas of Responsibility	Data Owner	Data Product Owner	Data Analyst	Data Scientist	Data Governance Manager	Data Steward	Solution Architect	Platform Ops	Security Architect	Security Engineer	DataOps Engineer	Data Solution Architect	Data Model Architect
Manage lake database templates mapping for ingesting data into the data platform	R	A										R	
Own lake database templates in the data platform		R									R	A	R

## Onboarding

 Expand table



Areas of Responsibility	Data Owner	Data Product Owner	Data Analyst	Data Scientist	Data Governance Manager	Data Steward	Solution Architect	Platform Ops	Security Architect	Security Engineer	DataOps Engineer	Data Solution Architect	Data Model Archit
Mapping Document													

## Purview

[Expand table](#)

Areas of Responsibility	Data Owner	Data Product Owner	Data Analyst	Data Scientist	Data Governance Manager	Data Steward	Solution Architect	Platform Ops	Security Architect	Security Engineer	DataOps Engineer	Data Solution Architect	Data Model Archit
Read access to the Data Glossary			R	R	A								
Manage Data Glossary Create, Update, Delete	R	R			A	R							
Approve Data Glossary changes	R	R			A								
Read access to the Data Catalog			R	R	A								
Manage Data Catalog Create, Update, Delete			R	R	A	R							
Approve Data Catalog changes	R	R			A								
Register a data source in Purview		R			A					R	R	R	
Manage Purview ADLS Scanning					A					R			
Manage Purview permissions					A					R			
Manage Purview policies (dataset access)					A	R							
Approve purview policies (dataset access)	R	R			A								
Monitor Purview Insight Reports	R	R			A	R							
Manage Purview					A					R			

Areas of Responsibility	Data Owner	Data Product Owner	Data Analyst	Data Scientist	Data Governance Manager	Data Steward	Solution Architect	Platform Ops	Security Architect	Security Engineer	DataOps Engineer	Data Solution Architect	Data Model Archit
Collections													

## Security

 Expand table

Areas of Responsibility	Data Owner	Data Product Owner	Data Analyst	Data Scientist	Data Governance Manager	Data Steward	Solution Architect	Platform Ops	Security Architect	Security Engineer	DataOps Engineer	Data Solution Architect	Data Model Archit
Manage encryption & decryption services for the data platform								R	A	R	R		
Approve data access requests for the Structured zone	R									R		R	R
Approve data access requests for the Gold Zone	R	R			A								
Approve data access requests for the Raw Zone	R									R		R	R
Monitor and audit data platform access	R	R							A	R			

## Next step

Learn about group alignment within data management landing zones and data landing zones:

- [Understand teams and functions for cloud-scale analytics in Azure.](#)

## Feedback

Was this page helpful?

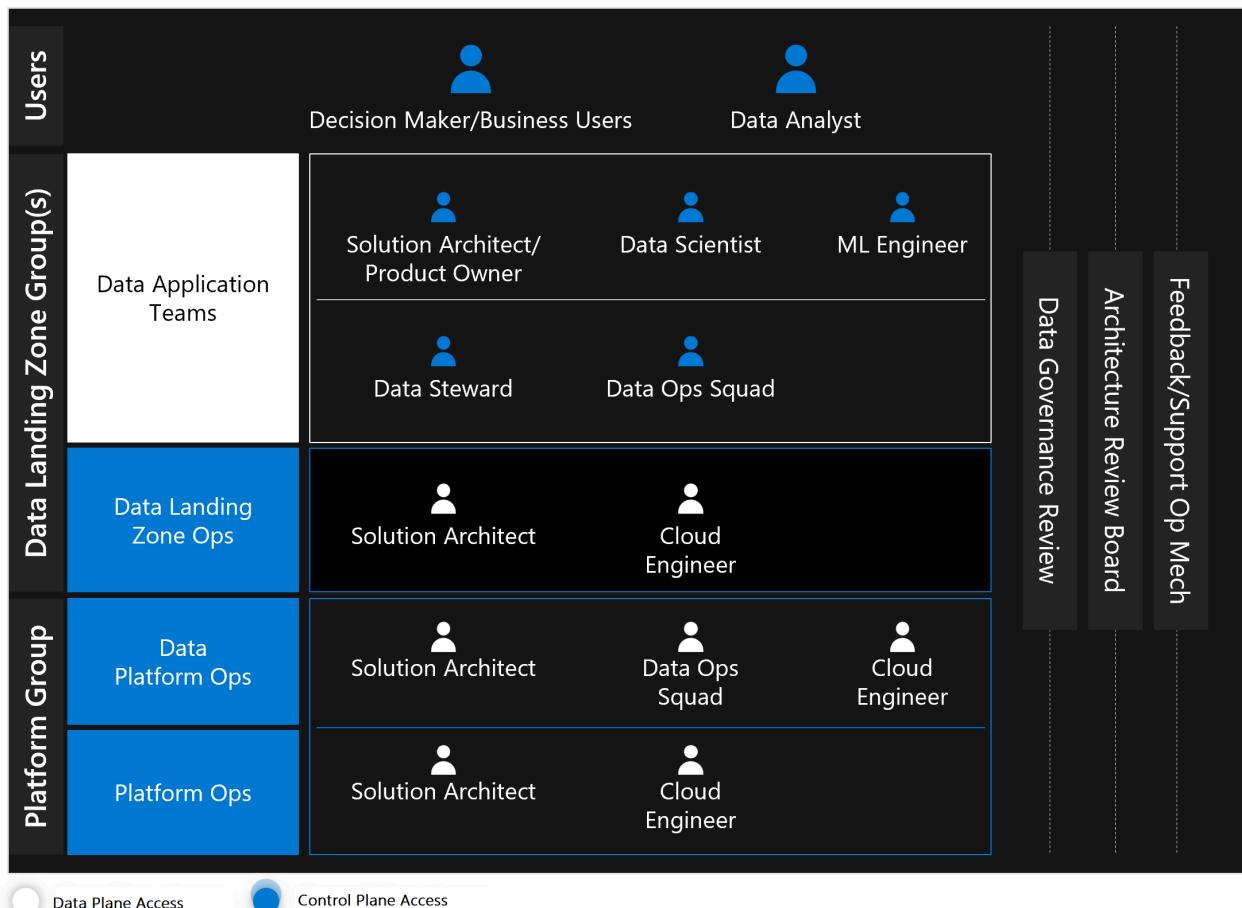
 Yes

 No

# Understand teams and functions for cloud-scale analytics in Azure

Article • 11/27/2024

For cloud-scale analytics, we recommend moving teams such as ingest, processing, analysis, consumption, and visualization from working in horizontally siloed teams to agile vertical cross-domain teams in each tier. Platform teams like data platform operations and platform operations are grouped together in a common platform group.



## Platform group

The platform group consists of two teams:

- **Platform ops:** Platform ops are part of the platform group. They operate and own the cloud platform. This team is responsible for instantiating the data management landing zone and data landing zone scaffolding, such as networking, peering, core services, and monitoring within cloud-scale analytics. They usually help data platform ops develop IT service management interfaces for personas in the data landing zone at the start of rolling out cloud-scale analytics. These interfaces tend

to be REST API calls to a service to onboard data products, set security, and add services to data landing zones.

- **Data platform ops:** The data platform ops group is housed within the platform group. Data platform ops provide services such as central monitoring, cataloging, and reusable policies for data landing zones and products. Data platform ops own the data management landing zone, and the team's other responsibilities are:

## Develop infrastructure

- Develop infrastructure-as-code templates for data landing zones; the templates must be updated and maintained over time, and they can cover multiple scenarios.
- Prioritize templates and add new functionalities based on feedback from other teams.
- Work in an agile framework with the common goal of producing standard infrastructure templates.

## Respond to new data landing zone requests

The data platform ops team must provide the tools and services to support the templates they've created. IT service management tools like ServiceNow can handle ticket requests approved by the data platform ops team for creating new data landing zones. After its approved, a new landing zone would fork from the base template to create a new DevOps project, and pipelines would deploy templates to a new environment.

## The data platform ops feedback and enhancement loop

Two options are available to enhance the templates:

- Teams in charge of infrastructure template instances can enhance their DevOps templates and deployments. If teams discover issues in the templates, data platform ops can support the teams and merge changes back from their fork into the template.
- Other data landing zone teams should be able to create improvement and backlog tickets that would enhance templates based on how the tickets are prioritized.

## Azure policies for cloud-scale analytics

Cloud-scale analytics principles emphasize self-service agility and guardrails to protect data, costs, and patterns. Data platform ops work with platform ops to define quality,

and these teams collaborate to implement specific data policies. Data platform ops should follow a review process to update and maintain new features that are added to products.

## Deploy and operate data management landing zones

Data platform ops and platform ops work together to deploy and operate data management landing zones. A data management landing zone provides shared services to data landing zones, making it a central piece of cloud-scale analytics.

## Data landing zone ops

Data landing zone ops operate and maintain their data landing zone instance while responding to data application team requests. They provide many of the same services as data platform ops but are limited to their data landing zone.

They work out of the forked repo that's created when a data landing zone is established. To request policy changes, they have to raise tickets to data platform ops to allow these exceptions.

## Support the data application team to customize data products

The data landing zone ops team supports the data application team by using pull requests to submit new product templates to their respective data product repositories.

As the owner of the landing zone, Azure DevOps routes the approval for changes to data landing zone ops:

- If approved, the template changes are moved to the main branch and deployed to production via continuous integration/continuous development, causing the data product platform/infrastructure to be updated.
- If denied, data landing zone ops work with the data application team to fix the changes.

## Respond to new data product requests

Data landing zone ops support data application teams in creating new data products. When a data application team requests assistance, an IT service management solution, such as an automation logic app, orchestrates the approval or deployment of a new data application repository. Data landing zone ops are notified of new requests and

approve or decline deployments. After approved, a new DevOps project is created, the main template and artifacts are forked, and a new data application is deployed.

## Adhere to the Azure Well-Architected Framework

Data landing zone ops are responsible for the data landing zone, and it's recommended for the team to be proficient in the [Azure Well-Architected Framework](#), which provides guidance on cost optimization, reliability, and security.

## Business as usual

Data landing zone ops are responsible for business tasks that include gathering feedback and enhancement requests. These requests are prioritized and shared with data platform ops regularly. The team monitors the data landing zone for incidents and health events. They engage other ops teams during severe incidents to mitigate, restore backups, failover, and scale services.

## Data application team

The data application team delivers new data products to the business. They source from data integrations' read data stores and transform them into business solutions. Anything that transforms data for use is classified as a **data product**. This team is often a mix of technical specialists and subject matter experts who can help the business achieve value quickly. Data products can range from simple reports and new data products to custom setups with data-driven Kubernetes web apps.

## New data products

Product owners and business representatives create requests for new data products when they're needed. The data office assesses the requirements and assembles a new data application team with a range of expertise. The team identifies the data products required and requests permission to access the data asset. If a new data product is needed, the data application team receives a ticket to ingest it. The team identifies the services required for the new data product and requests a new data product via the [data application deployment process](#). The data application team receives a forked repo from the master data application template to deploy the data application.

## Certify data products

In a self-service platform, anyone can create reports, curate data products in an Azure Data Lake developer storage account, and release data products for the business to use. Data product review requests occur when:

- Business sponsors log tickets to certify data products.
- Data platform ops nominate data products based on popularity.

A data application team can drive a certification process, defined by data platform ops and digital security, which might include:

- Tests devised to validate data transformations and business logic
- Assessments for security, compliance, or performance impact

Upon certification, artifacts are collated and uploaded to a data product repository, documentation is published, and the data application team is notified.

## Product support

Users can submit feedback with an IT service management solution or directly within the product as a ticket routed to the data product owner. This individual triages the request and determines whether to escalate it to the data application team to fix or enter feedback into a product backlog and review during product planning cycles.

## Data science applications team

While the data science products team creates data products, it's distinct because their functions lead to data products. Their work results in published models becoming data products for others to use, and the pattern follows a Machine Learning ops model associated with the data landing zone.

The data science products team starts by searching for and finding relevant data products for their use case. Data governance solutions can reveal more details like data quality, lineage, or a similar dataset or profile. They research if a sample dataset is available and if the data is relevant to the project. After data access is granted via a data catalog or a Microsoft Entra access package, the team uses the services in the data landing zone to explore and analyze the data.

Before processing all data, the team uses local or remote compute to process and analyze sample data products. They can optimize remote compute targets with larger data products to train and develop machine learning models with runs, outputs, and models tracked inside Azure Machine Learning.

When the team has developed machine learning models, they start operationalizing them. For achieving this objective, they expand the team to include DataOps and machine learning engineers who can assist with moving the models into a new data product, as outlined in a data application team role.

The data science team continues to work with the associated data product owners to capture feedback, support, and update models in production using a [machine learning ops methodology](#).

## Analyst

Analysts represent a large group that includes business analysts, power users, and generally anyone in the organization with an interest in optimizing data to create new business insights. Self-service enablement is a key principle that supports analysts in accessing analytics and data without having to secure a formal IT budget and resources.

### 💡 Tip

Enterprises should view insights created by analysts as the next set of potential data products to be certified for others to use within the business.

## Find and request data

Analysts consult data marketplaces/catalogs to discover relevant data products.

- If the data asset can't be found or doesn't exist, analysts open a support ticket with the data application team. The data application team assists with finding the dataset or adds the request to their backlog to assess it in another development cycle.
- If the dataset exists, analysts can identify Microsoft Entra group membership for assets listed in the catalog and use the Azure access package portal to request access to the Microsoft Entra group.

## Build new reports

Analysts can use tools like Microsoft Power BI to integrate data products into reports. These reports can be for their individual use or for publishing a certified data product. Before publishing the report across the organization, it would need to be certified with a data product certification process for security, compliance, and performance.

## Run as-needed queries

Cloud-scale analytics has shared workspaces where analysts can query data, subject to permissions. It's common for data products to provide dedicated compute to run queries as needed. In both cases, analysts can run queries against data products in the data landing zones, subject to permissions. The results from the queries can be stored in Azure Data Lake workspaces to be used again.

Since analysts can serve as an untapped source of information and improvements, enterprises are highly encouraged to create user feedback groups for each data landing zone.

In addition to participating in these user groups, analysts should submit data asset feedback to the data application team and data catalog issues within the data catalog or the IT service management solution. They can submit data process issues to the data application team or within an IT service management solution.

### ⓘ Note

An IT service management solution should serve as a central location for submitting feedback and escalating issues. Submitting direct feedback to individual teams might seem to be a faster solution, but this approach doesn't give the business visibility into the challenges in the platform. An IT service management solution with correct routing to the data application teams can give the business one view across the enterprise.

## Responsibility assignment matrix

- Responsible:** Who is completing the task?
- Accountable:** Who is making decisions and taking actions on the task?
- Consulted:** Who receives communications about decisions and tasks?
- Informed:** Who is updated about the decisions and actions during the project?

[\[+\] Expand table](#)

Role	Cloud environment	Data management landing zone	Data landing zone	Data integration	Data products
Service owner	Informed	Accountable	Consulted informed	Consulted informed	Consulted informed

<b>Role</b>	<b>Cloud environment</b>	<b>Data management landing zone</b>	<b>Data landing zone</b>	<b>Data integration</b>	<b>Data products</b>
Data landing zone service owner	Informed	Consulted informed	Accountable	Accountable	Accountable
Cloud platform ops	Responsible	Consulted	Consulted	Consulted	Consulted
Data platform ops	Consulted	Responsible	Responsible	Consulted	Consulted
Data landing zone ops	Informed	Responsible	Responsible	Responsible	Responsible
Data application team		Informed	Informed	Informed	Responsible

## Next steps

The [Azure Well-Architected Framework for data workloads](#)

---

## Feedback

Was this page helpful?

Yes
 No

# Manage cloud-scale analytics

Article • 04/19/2022

Today, DevOps has shifted the culture of how people think and work, accelerating the rate at which businesses realize value by helping individuals and organizations to develop and maintain sustainable work practices. DevOps combines development and operations, and is often associated with software engineering tools that support continuous integration (CI) and continuous delivery (CD) practices. These tools and practices include source code managers (such as Git, Apache Subversion, or Team Foundation Version Control) and automatic build and delivery managers (such Azure Pipelines or GitHub Actions).

DevOps combined with observability is key to providing an agile and scalable platform. DevOps give teams the ability to implement source control, CI/CD pipelines, infrastructure as code, Workflows and automation. Whilst observability enables business owners, DevOps engineers, data architects, data engineers, and site reliability engineers to detect, predict, prevent, and resolve issues in an automated fashion and avoid eliminating downtime that would otherwise break production analytics and AI.

## Source control

Source control ensures that code and configurations persist and that changes are tracked and versioned. Most source control systems also have built-in processes for review and working in different branches of a code repository. Currently, the most popular source control type currently is Git, which is a distributed version controls system that allows individuals to work offline and sync to central repositories. Git vendors typically also use branches and follow pull request guidance to support the change and review flow.

Branches isolate changes or feature developments without affecting other work that happens at the same time. The use of branches should be promoted to develop features, fix bugs, and safely experiment with new ideas. Pull requests merge the changes made from one branch into the default branch, and they support a controlled review process. For security purposes, the main branch should use pull requests to ensure code reviews.

### Important

Follow these guidelines for cloud-scale analytics repositories:

- Secure the repository's main branch by enforcing branches and pull requests to ensure a controlled review processes.
- Azure DevOps or GitHub repositories should be used for source control to track changes to the source code and allow multiple team members to develop code at the same time.
- Application code and infrastructure configurations should be checked into a repository.

## CI/CD pipelines

CI allows teams to automatically test and build source code and enables quick iterations and feedback loops to ensure high code quality in CD. Pipelines are ways to configure the CI of changes (software code or infrastructure code) and CD of the packaged/compiled changes. This is also referred to as *build and release*. CD describes the automatic deployment of applications to one or more environments. CD usually follows a CI process and uses integration tests to validate the entire application.

Pipelines can contain multiple stages with various tasks and can have simple to complex approval flows to ensure compliance and validation. Based on preference, pipelines can also be configured with various automatic triggers. For enterprise-scale and AI deployment, the production steps should always have human pre-approval, and this is built into the operation model. CI/CD pipelines should be built with GitHub Actions or Azure Pipelines, and they should be automated triggers.

## Infrastructure as code

The term *code* in IaC often raises concerns for IT staff without a developer background, but IaC doesn't refer to writing code the way in which typical software developers do it. However, it adopts many of the same tools and principles from the software development processes to deliver infrastructure in a predictable format.

IaC helps infrastructure to be provisioned, configured, and managed as part of a DevOps pipeline with full change controls, audit history, tests, validations, and approval processes, ensuring that tasks can be delegated to the appropriate roles for the project without compromising security and compliance.

The two approaches to IaC are declarative and imperative:

- Declarative refers to specifying the desired state of the infrastructure and having an orchestration engine execute the necessary actions to achieve the desired state.

In Azure, this is done with Azure Resource Manager templates. Third-party abstraction layers like Terraform are also available for this approach.

- The imperative approach refers to executing specific commands in a defined order. For Azure, this can be achieved with the command-line interface or PowerShell, but native programming language software developer kits, for example, .NET, Python, and Java, are also available if integrated solutions are required.

In Azure Resource Manager templates, the core provisioning is in the **resources** section, and the configuration of the individual resources is defined in a **properties** section. For an Azure Data Lake Storage Gen2, the configuration looks like the following:

JSON

```
{  
    "$schema": "https://schema.management.azure.com/schemas/2015-01-  
01/deploymentTemplate.json#",  
    "contentVersion": "1.0.0.0",  
    "resources": [  
        {  
            "type":  
                "Microsoft.MachineLearningServices/workspaces/datastores",  
                "name": "[concat(parameters('workspaceName'), '/',  
parameters('datastoreName'))]",  
                "apiVersion": "2020-05-01-preview",  
                "location": "[parameters('location')]",  
                "properties": {  
                    "DataStoreType": "adls-gen2",  
                    "SkipValidation": "[parameters('skipValidation')]",  
                    "ClientId": "[parameters('clientId')]",  
                    "ClientSecret": "[parameters('clientSecret')]",  
                    "FileSystem": "[parameters('fileSystem')]",  
                    "AccountName": "[parameters('accountName')]",  
                    "TenantId": "[parameters('tenantId')]",  
                    "ResourceUrl": "[parameters('resourceUrl')]",  
                    "AuthorityUrl": "[parameters('authorityUrl')]"  
                }  
        }  
    ]  
}
```

### ⓘ Important

Every layer of cloud-scale analytics such as data management landing zone, data landing zones or data applications (which create data products), should be defined with a declarative language like Azure Resource Manager or Terraform, checked into a repository, and deployed through CI/CD pipelines. This allows teams to track

and version changes to the infrastructure and configuration of Azure scope while supporting different architecture levels to be automated in an agile way. This guidance leads teams to use Git repositories to always have visibility into the state of specific Azure scopes.

## Workflows and automation

Teams should use CI/CD pipelines in multiple stages to ensure that developed code is without errors and ready for production. Some best practices are to have a development environment, a testing environment, and a production environment. These stages should also be reflected in Azure by using separate services for each environment.

The platform team is responsible for providing and maintaining deployment templates to scale quickly within an organization and simplify deployments for teams unfamiliar with IaC. These templates serve as a baseline for new artifacts within the scenario and need to be maintained over time to represent best practices and common standards within the company.

Deployments to test and production should only be managed through a CI/CD pipeline and a service connection with elevated permissions to enforce common best practices (for example, Azure Resource Manager templates).

### ⊗ Caution

Data application teams should only have read access to test and production environments, and deployments to these environments should only be executed through CI/CD pipelines and service connections with elevated permissions. To accelerate the path to production, data application teams should have write access to the development environment.

## Next steps

[Platform automation](#)

# Platform automation

Article • 12/01/2022

Cloud-scale analytics is focused on separating the runtime, automation, and user layers.

Automation to runtime interaction is done using Azure Pipelines and scripted Azure Resource Manager templates.

## Important

Cloud-scale analytics uses [Azure policies](#) to put boundaries in place and ensure that changes performed by the data landing zone operations teams are compliant.

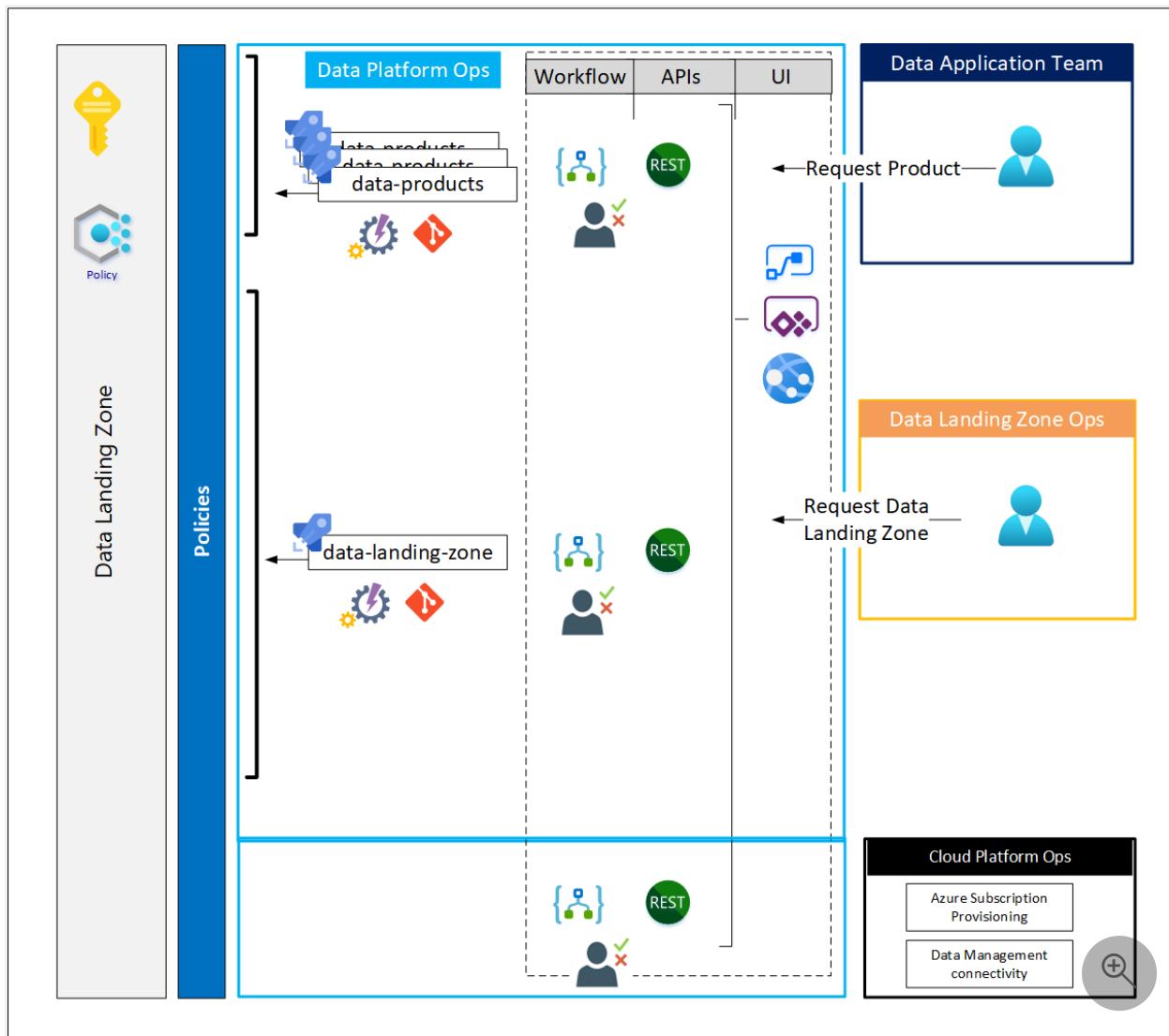
Cloud-scale analytics uses policies to enforce:

- Naming conventions.
- Network rules.
- Non-allowed services.

The data landing zone has specific requirements over the standard configuration.

- The size of subnets.
- The number of subnets.
- The number of resource groups.
- The names of resource groups.
- Key vaults.

The following diagram shows how automation principles are implemented for a data landing zone.



## Deployment models

Cloud-scale analytics consists of:

- A data management landing zone.
- One or more data landing zones.
- One or more data application which produces data products in each data landing zone.

Each application can evolve independently over time because of different requirements and lifecycles. For example, one of the data landing zones might require RA-GRS storage accounts at some point. It's important to have an infrastructure as code (IaC) representation of each of asset in a repository. This way, changes can be implemented based on requirements in the respective data landing zone and data applications.

The following table summarizes the teams involved in a cloud-scale analytics deployment.

Name	Role	Number of teams
Cloud platform team	The Azure cloud platform team in your organization.	One for the whole Azure platform.
Data platform team	In charge of creating and maintaining Azure Resource Manager template repositories for different levels of cloud-scale analytics. Also maintains the data management landing zone and supports other teams if there are deployment issues or required enhancements.	One for cloud-scale analytics.
Data landing zone team	In charge of deploying and maintaining a specific data landing zone. Also supports the deployment and enhancement of data applications which produce data products.	One team per data landing zone.
Data applications team	In charge of data products deployment and updates.	One team per data application.

## Next steps

[Provision cloud-scale analytics platform](#)

# Provision the cloud-scale analytics

Article • 04/22/2022

## Data management landing zone deployment process

The data platform operations team is responsible for deploying a data management landing zone. The data-management landing zone should have its own repository maintained by the data platform operations team.

 **Caution**

Create and deploy a data-management landing zone before any data landing zone is deployed.

## Data landing zone deployment process

Teams can use templates provided by the data platform operations team to avoid starting from scratch for each asset. We recommend a forking pattern to automate the deployment of a new landing zone.

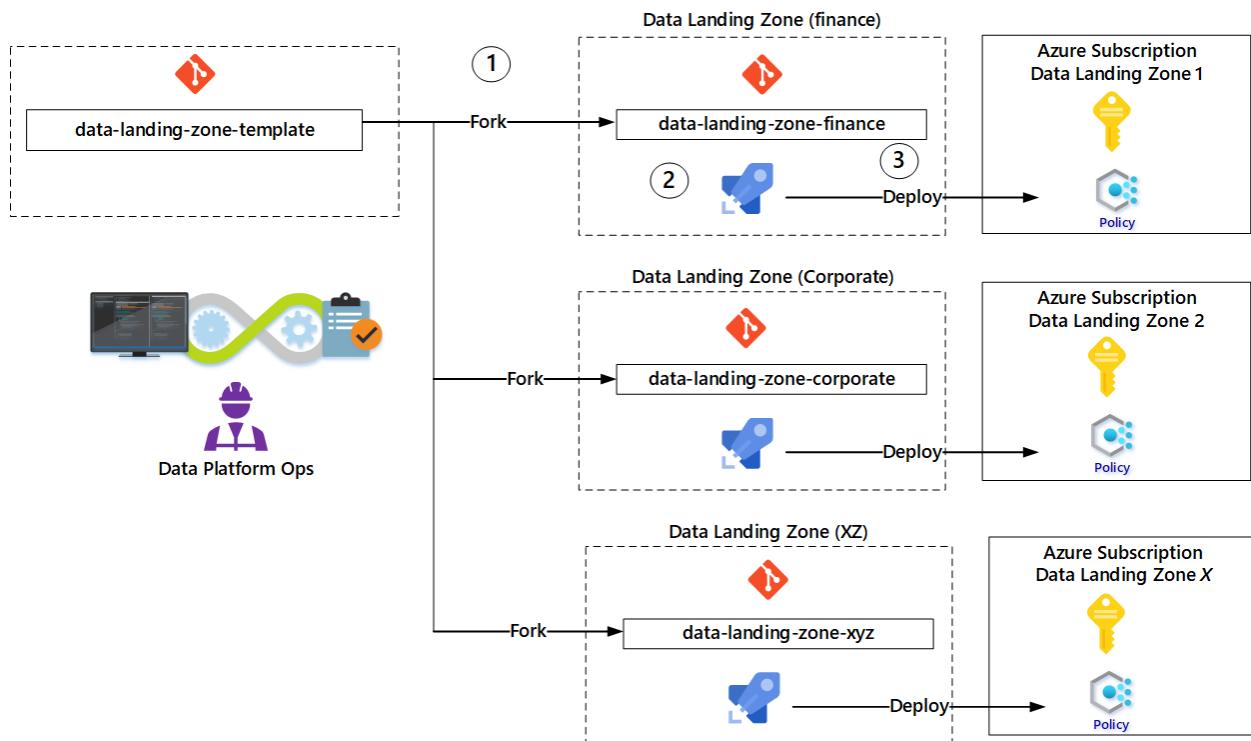
For example, a data landing zone operations team requests a new data landing zone using an IT management tool or [Power Apps](#). Upon approval of the request, start the following workflow using parameters from the request:

1. Deploy a new subscription for the new data landing zone.
2. Fork the main branch of the data landing zone template to create a new repository.
3. Create a service connection in the new repository.
4. Update parameters in the new repository based on parameters from the request.
5. Create a deployment pipeline to deploy the services, triggered by check-in of the updated parameters.
6. Notify the data landing zone operations team that the new landing zone is available.

The data landing zone operations team can now change or add Azure Resource Manager templates.

This workflow can be automated using multiple service sets on the Azure platform. Handle some of the steps, such as renaming parameters in parameter files, using CI/CD pipelines. Other steps can be executed using other workflow orchestration tools such as Logic Apps.

- ① Fork main repo with ARM Templates in each instance's Git Repo and create dedicated CI/CD Pipeline
- ② Update ARM Templates parameter files
- ③ Deploy to Azure subscription via dedicated CI/CD Pipeline



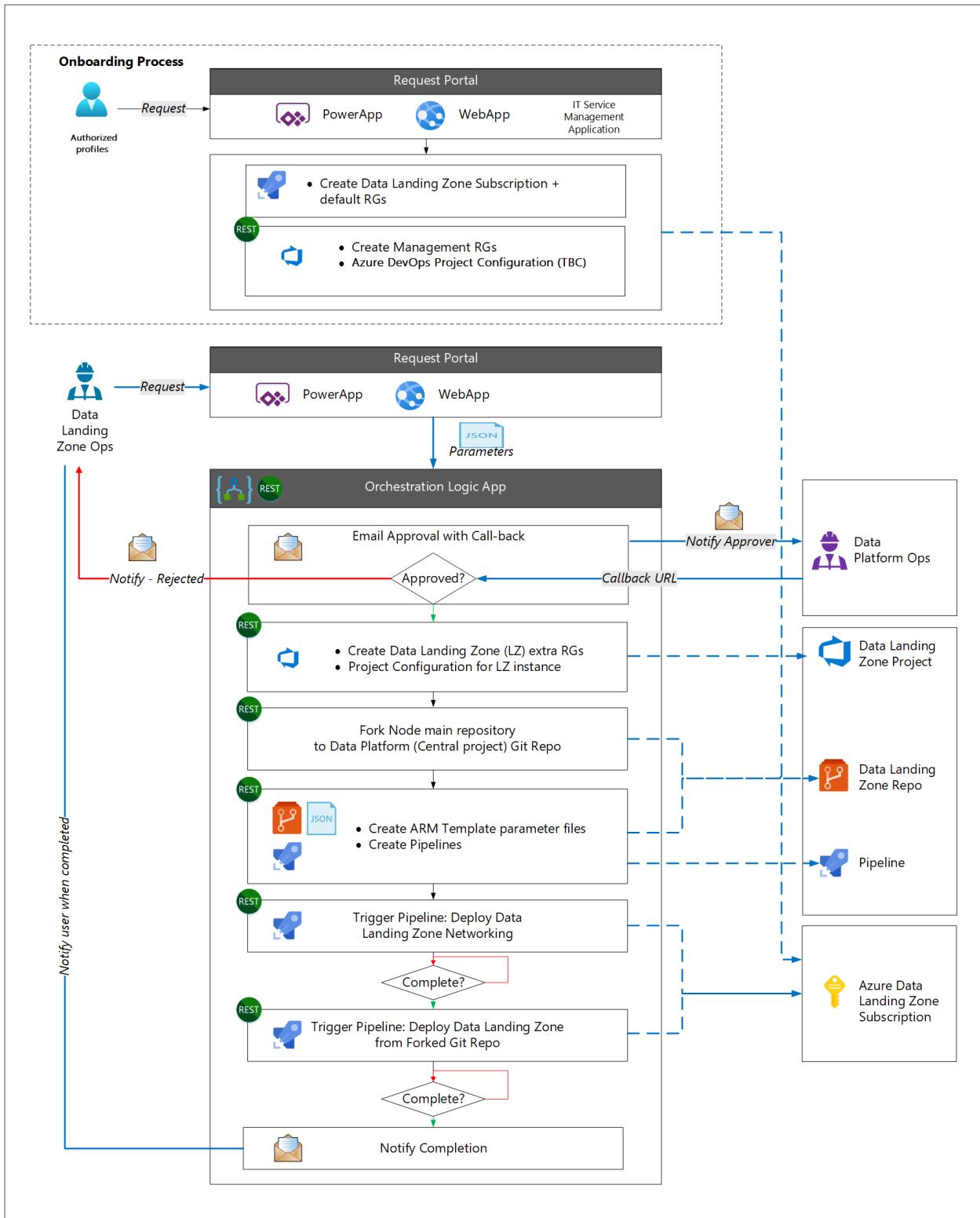
The forking pattern allows teams to update their templates from the original templates used to fork them. Also, if improvements or new features are implemented in the template repositories, operations teams can pull them into their fork.

Adopt best practices for repositories, such as:

- Secure the main branch.
- Use branches for changes, updates, and improvements.
- Define the code owners who approve pull requests before merging changes into the main branch.
- Validate branches through automated testing.
- Limit the number of actions and persons in the team, such as who can trigger build and release pipelines.

### Tip

Coordinate activities between teams to ensure that improvements or new features in the original templates are replicated in all data landing zone instances. Operations teams can pull original template changes into their fork.

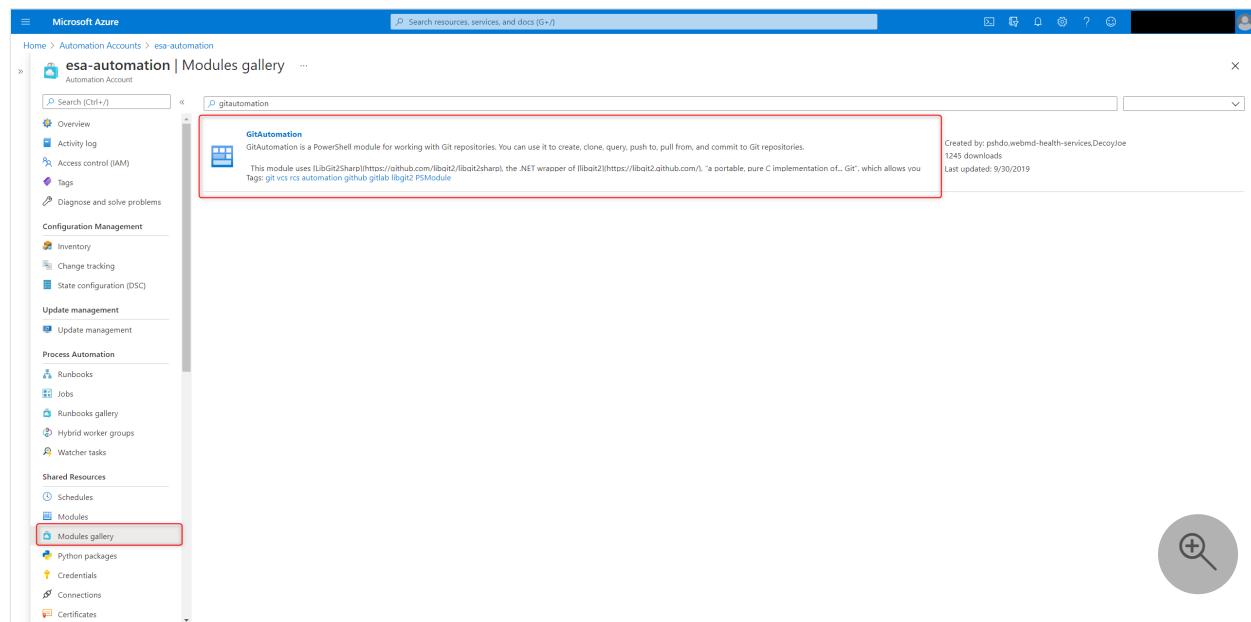


The onboarding process is separate from the data landing zone deployment process. This separation is based on the assumption that most organizations have a standard Azure subscription deployment process as part of their cloud operating model. The onboarding process deploys standard corporate components (such as a third-party IT service management tool). Data landing zone-specific components are deployed next.

There are no Git APIs available to clone/update/commit/push in the proposed automation solution. So our approach is to use an [Azure Automation account](#) containing PowerShell runbooks that:

- Set up a data landing zone
- Fork the main repository to a data platform Git repository
- Set up the subnet configurations for the data landing zone
- Set up Azure Active Directory

The runbooks use Git functions from the [GitAutomation](#) PowerShell module for working with Git repositories. By installing this module inside an Azure Automation account, users can do create, clone, query, push, pull, and commit operations in Git repositories. The following image shows the `GitAutomation` module installed inside an Azure Automation account:



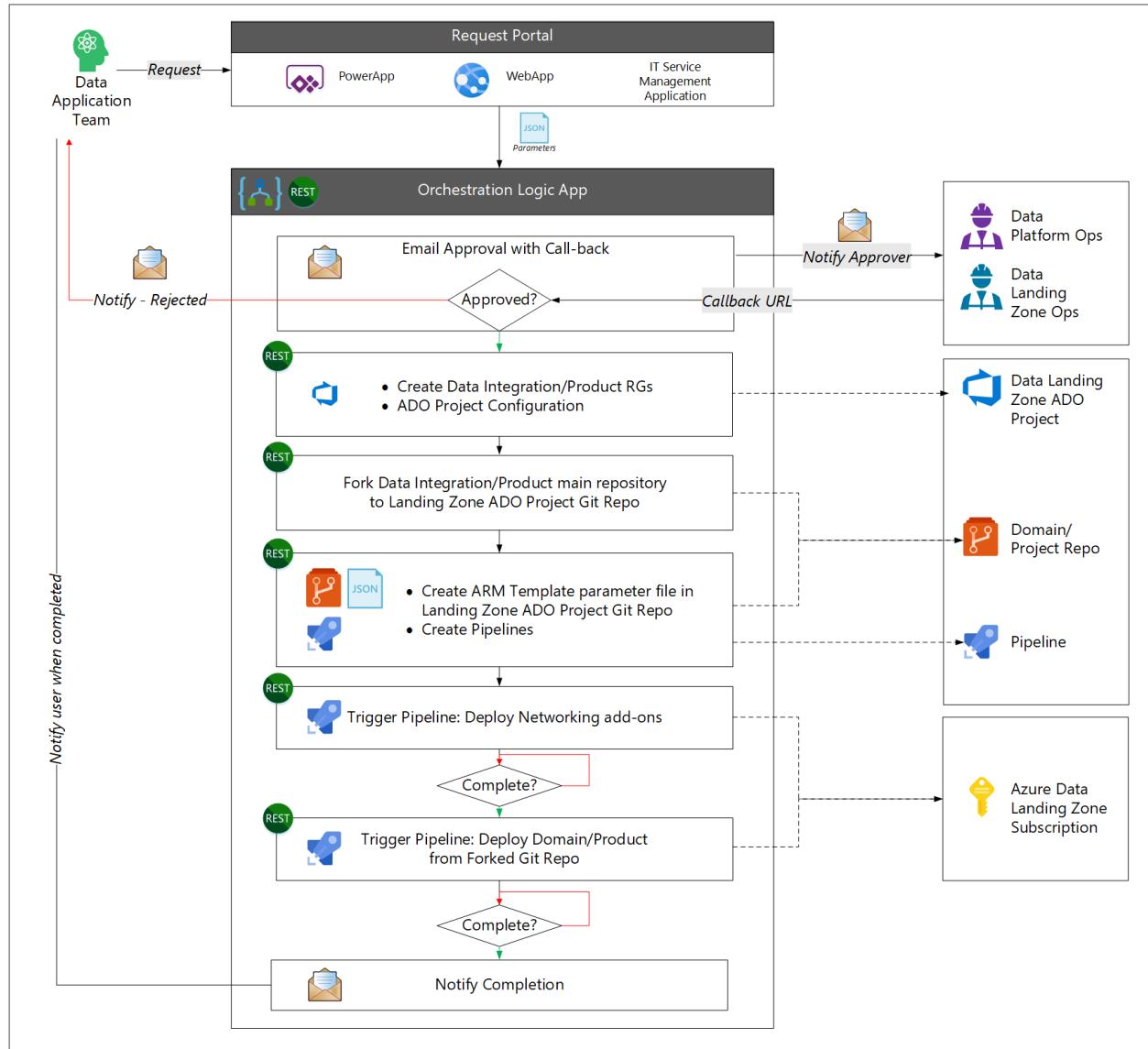
Use the `Copy-GitRepository` function from the `GitAutomation` module to clone the main Git repository from the URL specified by `URL` to the data platform Git path specified by `DestinationPath`.

This approach to data landing zone deployment is flexible, while ensuring that actions are compliant with organizational requirements. Lifecycle management is enabled by applying new features or optimizations from the original templates.

## Data application deployment process

After a data landing zone has been created, onboarding can start for the data application teams. The data platform or data landing zone operations teams grant deployment approval.

Deployment is done either directly using DevOps tooling or called via pipelines/workflows exposed as APIs. Similar to the data landing zone, deployment begins with forking the original data application repository.



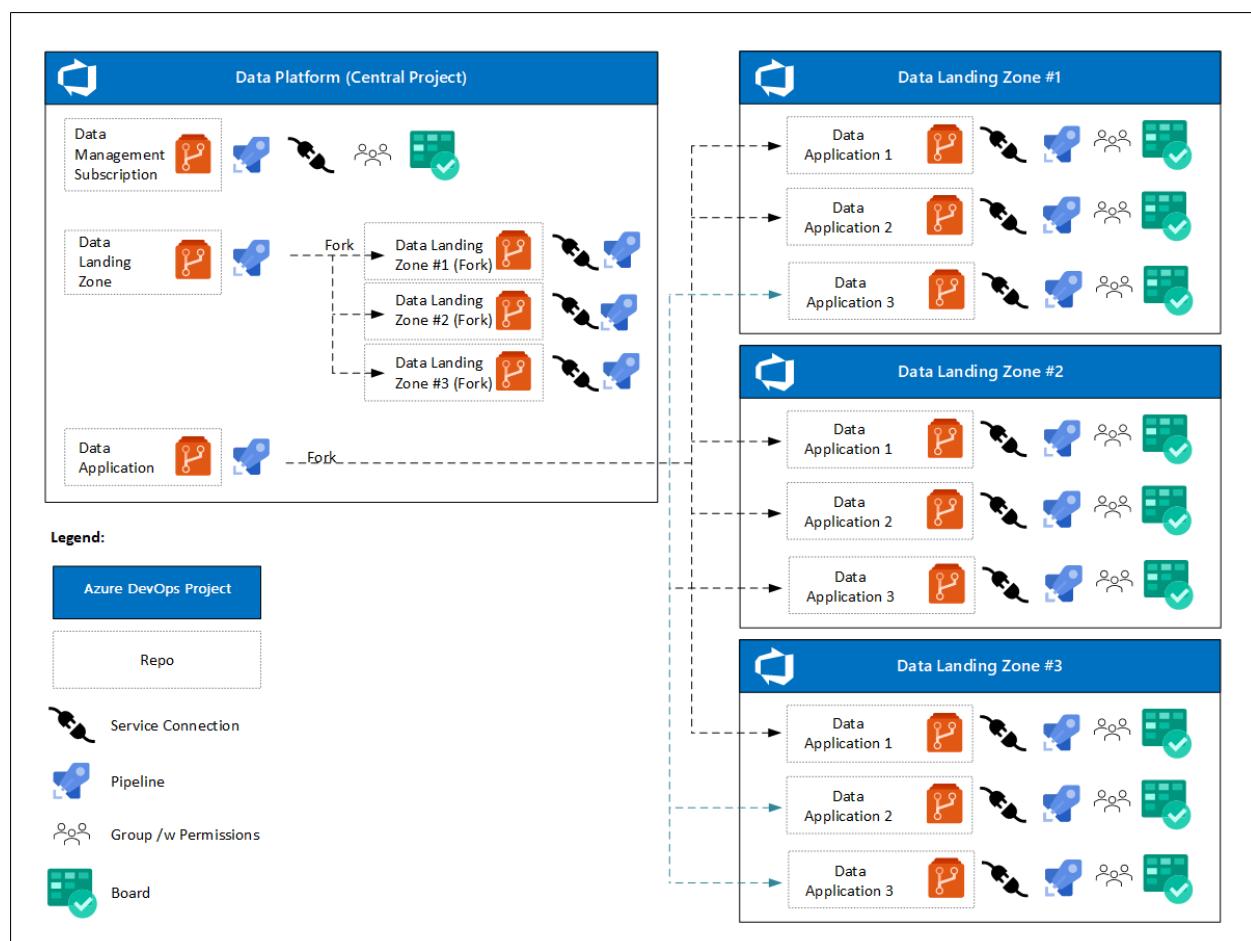
1. The user makes a request for new data application services.
2. The workflow process requests approval from the data platform or data landing zone operations team.
3. The workflow calls the IT service management API to create required resource groups, and creation of an Azure DevOps service connection. The workflow assigns a team to the Azure DevOps project.
4. The workflow forks the original data application repository to create the destination Azure DevOps project.
5. The workflow creates an Azure Resource Manager template parameter file and pipelines.
6. The workflow then starts an Azure pipeline to create the networking requirements, and another Azure pipeline to deploy the data application services.
7. The workflow notifies the user on completion.

## Tip

If you're new to DataOps, review the [DataOps for the modern data warehouse](#) hands-on lab in the Azure Architecture Center. The lab's scenario describes a fictional city planning office that can use this deployment solution. The deployment solution provides an end-to-end data pipeline that follows the modern data warehouse architectural pattern, along with corresponding DevOps and DataOps processes, to assess parking use and make informed business decisions.

## Summary

The above patterns provide control, agility, self-service, and lifecycle management of policies.



At the start of the project, the data platform has one Azure DevOps project with one or more Azure Boards. Individual DevOps teams focus on:

- One repository for the data management landing zone, pipelines, and a service connection to the cloud environment.
- One template repository for the data landing zone, pipelines to deploy a data landing zone instance, and service connections to cloud environments.

- One template repository for data product services, pipelines to deploy a data product instance, and service connections to cloud environments. These connections are forked from data landing zone Azure DevOps Projects.

Once data landing zones have been deployed, cloud-scale analytics prescribes that:

- Each data landing zone will have its own Azure DevOps project with one or more Azure Boards.
- For each data application, its data landing zone Azure DevOps project fork is created after request approval.
- Each data application includes:
  - A service connection.
  - A registered pipeline.
  - A DevOps team with access to their Azure board and repository.
  - A different set of policies for the forked repository.

To control the deployment of data applications, follow these practices:

- The data landing zone operations team owns and secures the main repository branch.
- Only the main branch is used to deploy to test and production environments.
- Feature branches can deploy to development environments.
- Feature branches are owned by the DataOps teams. They're used to test new or modified features.
- DataOps teams can merge feature branches into other feature branches without approval.
- DataOps teams create a pull request to merge feature branches into the main branch, and the data landing zone operations team provides approval.
- New features or improvements to the original templates are merged into the forked repository to keep them updated.

## Next steps

- [Deployment templates for cloud-scale analytics deployments](#)
- [An introduction to Azure Automation](#)

# Data observability

Article • 07/03/2023

Data observability is your ability to understand the health of your data and data systems by collecting and correlating events across areas like data, storage, compute and processing pipelines.

Building and operating a resilient, scalable, and performant data platform requires adopting proven DevOps-inspired processes across teams that represent functional domains. Data observability enables business owners, DevOps engineers, data architects, data engineers, and site reliability engineers to automate issue detection, prediction, and prevention, and to avoid downtime that can break production analytics and AI.

## Key areas of data observability

Most data platforms operate on these key areas of data observability:

- [Data Platform Service Monitoring](#)
- [Data Pipeline Performance Monitoring](#)
- [Data Quality Monitoring](#)
- [Data Lineage](#)
- [Data Discovery](#)

End-to-end data observability involves not just capturing events and measuring metrics across all these components but also correlating those events and metrics. This provides a comprehensive view of your enterprise data environment's health and reliability.

This article describes each component and how it contributes to achieving data observability.

## Data platform service monitoring

Foundational infrastructure for an enterprise data platform can include a mix of both provider-managed and self-managed infrastructure to enable storage and computing. DevOps engineers or infrastructure engineers need to monitor this foundational infrastructure so they can identify and resolve system outages and performance bottlenecks that affect modern data and analytics pipelines.

Monitoring data from databases and networking layers can improve your processing throughput and minimize network latency. Teams need tools that they can use to

capture metrics, notify, track, and remediate incidents and correlate with the data and analytics issues.

We recommend that your teams incorporate observability-as-code into your infrastructure-as-code layer so monitoring instrumentation is enabled out-of-box as soon as they create a resource. Most Azure services offer out-of-box instrumentation for key resource metrics like diagnostic data.

## Data pipeline performance monitoring

Increasingly complex data pipelines containing multiple stages and dependencies now generate massive amounts of monitoring data. This data includes events, metrics, and logs. You can optimize your data pipeline performance by collecting and analyzing monitoring data.

Your data teams should track the state of your data pipelines across multiple related data products and business domains. When your team is notified early about failures or runtimes that are longer than expected, they can minimize and remediate downtime. Correlation of pipeline monitoring data and platform service monitoring can provide recommendations for performance tuning, such as boosting CPU and memory for your high load pipelines.

## Data quality monitoring

Data quality is the degree to which your data is accurate, complete, timely, and consistent with your organization's requirements. You need to constantly monitor your data sets for quality to ensure that the data applications they power remain reliable and trustworthy. DataOps has been consistently improving data reliability and performance by automating data quality tests (unit, functional, and integration). These improvements make faster and more efficient fault detection and data analytics possible.

To adopt DevOps and SRE principles into data quality, teams must build repeatable, iterative processes and frameworks to catch data quality issues, track those issues in dashboards, and set up alerts for any deviations.

Time to Detect (TTD), Time to Recovery (TTR), and other data quality metrics can be tracked from your data quality monitoring. TTD refers to the length of time it takes for your data team to detect a data quality issue of any kind, from freshness anomalies to schema changes that break entire pipelines. TTR refers the length of time it takes for your team to resolve a data incident once alerted. Improving your data quality is more than a technical challenge; it involves significant organizational and cultural support.

The governance section on [data quality](#) explores how you can implement data quality within your scenario.

## Data lineage

Data lineage is broadly understood as a continuous record that follows your data's origin, transformations, and movement over time across your data estate. Data lineage is used in retrospective tasks, including troubleshooting, debugging, and tracing root causes of pipeline issues. Lineage is also used for data quality analysis, compliance, and "what if" scenarios, which are often referred to as *impact analysis*.

Lineage is represented visually to show data moving from source to destination, including how the data is transformed over time.

The governance section on [data lineage](#) explores how you can implement data lineage within your scenario.

## Data discovery

Data discovery is the first step for a data analytics or data governance workload for consumers. In an enterprise data lake platform, it's difficult for data consumers (like data scientists and analysts) to locate the data they need and evaluate its reliability. Data catalogs with accurate metadata make searches easier using data index that provides:

- locations of available data
- data quality detection
- data structure understanding
- data lineage understanding
- access to desired data

Data catalogs offering these search capabilities increase the speed of all data discovery processes.

The governance section on [data catalogs](#) explores how you can implement data discovery within your scenario.

## Set SLAs, SLIs and SLOs

Your organization's teams can adopt DevOps-style Site Reliability Engineering (SRE) practices for data monitoring. Service level agreements (SLAs), service level indicators (SLIs), and service level objectives (SLOs) can help your organization reduce downtime and ensure your data's data reliability.

# Service level agreements (SLAs)

SLAs require well-defined SLIs, which are quantitative measures of service quality, and agreed-upon SLOs, which are the ideal values or ranges each SLI should meet.

Setting a data SLA requires the active participation and collaboration of all stakeholders that will be affected by an SLA. These stakeholders can include data producers, data engineers, data analysts, data consumers, business analysts, and others.

Setting reliability SLAs usually includes three steps: defining, measuring, and tracking.

Begin setting your SLA by defining what reliability means. All key stakeholders must agree on this definition. Ensure every key stakeholder is involved and buys in, especially if your downstream consumers come from different teams or different geographical regions and time zones.

Your SLA needs to be carefully crafted. Involve your legal team if data consumers are external paid customers. For internal customers, your SLA definition should include key areas like data promise, data quality, and a process to handle data incidents if the promise isn't met.

## Example SLA

Suppose Contoso is a media company that runs an enterprise data lake, and this data lake powers multiple data products across different business domains. The Contoso's data application team is responsible for delivering the prior-day sales data that powers Contoso's sales dashboard. When they miss a data delivery or deliver incomplete data, the data engineering team faces emails from frustrated executives and has to manually triage the broken pipeline that's supposed to deliver sales data. To measure and improve on their deliverables, the data team sets an SLA with the Sales team as demonstrated in the following section.

### Service Level Agreement - Data Team to Sales Team

Agreement	Description
Business area	The data team promises to empower the sales team's ability to make data-driven decisions
Promise	The data team promises to deliver the prior-day sales data that powers the sales dashboard. This data can provide sales and conversion rates for all US regions. Data pipelines will deliver data to power the sales dashboard before 6:00 UTC

Agreement	Description
Data quality	Null check: Customer name can't be null. Missing value: Customer region can't be missing. Freshness: Sales date should include any transaction before 24:00 UTC
Data incident management	If the above promise of data delivery isn't met, the sales team can report the problem and the data team promises to resolve the problem with a TTR < 6 Hours

## Service level indicators (SLIs)

SLIs should always meet or exceed the SLOs outlined in your SLA. When setting an SLI, begin by identifying key metrics you can track and measure to achieve your agreed-upon SLA.

### SLI example

Suppose Contoso's data team identifies key metrics from different areas to meet the SLA outlined in the previous example. They also build a dashboard, set up alerts for if key metrics deviate from a set baseline, and automate actions to mitigate some issues.

Metric	Purpose
Spark cluster CPU and memory usage	To measure any performance bottle neck in the underlying infrastructure used to run data pipelines
Pipeline total run time in minutes	To measure if a pipeline takes more time than expected to run
Pipeline failure and success rates	To measure how many pipelines fail or succeed
Data quality metrics (downstream)	To ensure the data delivered by the data pipeline meets expectations
Data quality metrics (upstream)	To ensure that upstream decencies of raw data quality are met
Transformation metadata updates	To ensure that lineage from upstream to downstream contains metadata about all transformations applied to data
Downstream data indexing and updates	To ensure the sales team discovers all data sets that power their dashboard
Defined process for measuring TTD and TTR	To measure TTD and TTR and ensure TTR < 6 hours

# Service level objectives (SLOs)

An SLO consists of an SLI, the duration over which that SLI is measured, and the targeted success rate that is practically achievable. Defining your direction and targeted success can be an overwhelming task initially. Don't expect perfection, but rather steady improvement over multiple iterations.

SLOs can depend on:

- Data product
- Data category
- Data source regions
- Data observability components

## SLO example

Suppose Contoso's data team delivers sales data across seven different United States regions. 210 data sets are delivered every calendar year across all regions, and only 200 data sets are complete and meet the SLA. These successful deliveries translate to a 95.99% success rate for that year. The 10 failed (incomplete) data sets occurred at an acceptable error rate of 4%.

The data team creates a monitoring dashboard that tracks aggregated SLIs to monitor this SLO over a period of 30 days. Both the data team and sales team get notified when target success rate isn't achieved.

# Data observability maturity model

Data observability is an essential part of the DataOps framework and should be considered parallel to your efforts to improve your organization's DataOps processes. The following maturity model can help you assess the current state of your data observability and decide on the next steps for your journey.

Stage	Data platform service monitoring	Data pipeline performance monitoring	Data quality monitoring	Data lineage	Data discovery
Stage 5 (Highly advanced)	Data is collected across all the data observability components	Data pipeline performance metrics are tracked across multiple data products.	A high Level of trust in data quality is established.	Data lineage is visually represented and is used in multiple ways, such as tracing	Data consumers can easily find available data that they need.

<b>Stage</b>	<b>Data platform service monitoring</b>	<b>Data pipeline performance monitoring</b>	<b>Data quality monitoring</b>	<b>Data lineage</b>	<b>Data discovery</b>
	<p>from one or more data products in a unified view and is correlated using machine learning to find any anomalies.</p> <p>Dashboards track SLO, SLI, and SLA across all data observability components.</p>	Root cause analysis is completed and driven by the system.	consumers can verify the reliability of data.	root causes of pipeline failure, data quality analysis, and compliance.	
<b>Stage 4 (Advanced)</b>	<p>Dashboards track SLO, SLI, and SLA across the most critical data observability components.</p> <p>Platform monitoring data and pipeline performance monitoring data are correlated using automation.</p>	<p>Data incident tools monitor and measure TTD and TTR metrics for any incidents.</p>	<p>Data quality is maintained through a framework that's usable across multiple data products and tracked using dashboards.</p>	<p>Data lineage includes data quality tags and is connected to data discoverability.</p>	<p>Data lineage is now connected to data discoverability and includes data quality tags as well.</p>
<b>Stage 3 (Evolving)</b>	<p>Well defined SLO, SLI, and SLA cover most critical almost all components for Data Observability.</p> <p>Data incidents are managed with</p>	<p>Platform monitoring data is correlated with data pipeline performance monitoring using some amount of automation.</p>	<p>Data quality checks are well defined and mapped to custom metrics.</p>	<p>Data lineage has matured to contain enough metadata needed for decision making.</p>	<p>Data discoverability is achieved using specialized data catalog tools.</p>

<b>Stage</b>	<b>Data platform service monitoring</b>	<b>Data pipeline performance monitoring</b>	<b>Data quality monitoring</b>	<b>Data lineage</b>	<b>Data discovery</b>
specialized tools.					
<b>Stage 2 (Planning)</b>	An initial draft of SLO, SLI, and SLA covers the most critical components needed for data observability.  Platform monitoring data is centralized and there is a unified view of the entire data environment.  All data incident management is manual.	Data pipeline performance metrics are defined and measured.	Data quality checks exist, but no standard metric is defined, measured, and visualized.	Data lineage is limited to single data product or isn't tracked.	Data discoverability is achieved but no sophisticated tools are used.
<b>Stage 1 Learning</b>	Every critical platform service (provider-managed and self-managed) is monitored in the data landscape.	Pipeline monitoring is minimal. Failures trigger alerts, but have no insights into any possible cause.	Data quality tests can be run from the pipeline, but no metric is measured or tracked.	Data lineage doesn't exist.	Data discoverability doesn't exist.

# Azure Well-Architected Framework for data workloads

Article • 11/27/2024

The [Plan methodology of this scenario](#) outlines a process for you to rationalize your data estate, prioritize technical efforts, and identify data workloads. For many of the named workloads, it's important to adhere to a set of architectural principles. These principles help guide development and optimization of the workloads. The five architectural constructs are detailed in the [Azure Well-Architected Framework](#). This guidance provides a summary of how you can apply these principles to the management of your data workloads.

## Cost optimization

It's critical to architect with the right tool for the right solution in mind. This principal can help you analyze spend over time. It can also help you analyze your ability to scale out versus scale in when needed. For your data workloads, consider reusability, on-demand scaling, reduced data duplication, and take advantage of the Azure Advisor service.

For more information, see [Design review checklist for Cost Optimization](#).

## Performance efficiency

User delight comes from performance of your workloads. Performance can vary based on external factors. It's key to continuously gather performance telemetry and react as quickly as possible. Build on the shared environmental controls for management and monitoring to create alerts, dashboards, and notifications specific to the performance of your workload. The key considerations are:

- Storage and compute abstraction
- Dynamic scaling
- Partitioning
- Storage pruning
- Enhanced drivers
- Multilayer cache

For more information, see [Design review checklist for Performance Efficiency](#).

# Operational excellence

Operational management of your data workloads can include advanced automation that improves your ability to quickly respond to events. Build on top of centralized data operations through workload-specific process automation, automated testing, and consistency. For AI, consider using the shared MLOps framework as part of your normal release cycle.

For more information, see [Design review checklist for Operational Excellence](#).

# Security

Security and [data management](#) must be built into the architectural process at layers for every application and workload. Cloud-scale analytics focuses on establishing a foundation for security. This foundation is built when you configure your Azure landing zones and you manage them separate from the workload. However, the workload team is still responsible for validating the following minimum requirements. If necessary, workload-specific solutions might be required to augment the configuration of the environment.

- Ensure confidentiality and integrity of data, including privilege management, data privacy, and establishing appropriate controls.
- Implement appropriate [network isolation](#) and [end-to-end encryption](#), auditing, and policies at the platform level.
- Use single sign-on (SSO) integration, multifactor authentication backed conditional access, and managed service identities.
- Adhere to separation of concerns, such as control pane versus data plane, through proper application of [role-based access control \(RBAC\)](#), and where possible, attribute-based access control (ABAC).
- Ensure the workload team is involved in regular or continuous vulnerability assessment, threat protection, and compliance monitoring.
- Secure data

For more information, see [Design review checklist for Security](#).

# Reliability

Everything has the potential to break and data pipelines are no exception. Because of this, great architectures are designed with availability and resiliency in mind. The key considerations are how quickly you can detect change, and how quickly you can resume operations.

Your data environment should consider resilient architectures, cross region redundancies, service level, service-level agreements (SLAs), and critical support. The existing environment should also include auditing, monitoring, and alerting by using integrated monitoring and a notification framework.

On top of these environmental controls, the workload team should consider:

- More architecture modification to improve service level SLAs
- Redundancy of workload-specific architecture
- Processes for monitoring and notification beyond what is provided by the cloud operations teams

For more information, see [Design review checklist for Reliability](#).

## Next steps

[Introduction to architectures for cloud-scale analytics](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Architectures overview

Article • 04/19/2022

Before you start to build out the data architectures of your cloud-scale analytics framework, review the articles in the following table.

Section	Description
<a href="#">Build an Initial Strategy</a>	How to build your data strategy and pivot to become a data driven organization.
<a href="#">Define your plan</a>	How to develop a plan for cloud-scale analytics.
<a href="#">Prepare analytics estate</a>	Overview of data management and data landing zones with key design area considerations like enterprise enrollment, networking, identity and access management, policies, business continuity and disaster recovery.
<a href="#">Govern your analytics</a>	Requirements to govern data, data catalog, lineage, master data management, data quality, data sharing agreements and metadata.
<a href="#">Secure your analytics estate</a>	How to secure analytics estate with authentication and authorization, data privacy, and data access management.
<a href="#">Organize people and teams</a>	How to organize effective operations, roles, teams, and team functions.
<a href="#">Manage your analytics estate</a>	How to provision platform and observability for a scenario.

## Physical architecture

The physical implementation of cloud-scale analytics consists of two main architectures: the [data management landing zone](#) and [data landing zone](#).

## Data applications

Data applications are a core concept for delivering a data product and can be aligned to both lakehouse and data mesh patterns.

## Cloud-scale analytics

You can [scale](#) your cloud-scale analytics deployment by using multiple data landing zones.

## Data mesh

Implement [data mesh](#) by using cloud-scale analytics. Although most cloud-scale analytics guidance applies, there are some differences to be aware of for data domains, self-serve data platforms, onboarding data products, governance, data marketplace, and data sharing.

## Deployment templates for cloud-scale analytics

The following table lists reference templates that you can deploy.

Repository	Content	Required	Deployment model
<a href="#">Data management template</a> ↗	Central data management services and shared data services like data catalog and self-hosted integration runtime	Yes	One per cloud-scale analytics
<a href="#">Data landing zone template</a> ↗	Data landing zone shared services, including ingestion, management, and data storage services	Yes	One per data landing zone
<a href="#">Data integration template - batch processing</a> ↗	Additional services necessary for batch data processing	No	One or more per data landing zone
<a href="#">Data integration template - stream processing</a> ↗	Additional services necessary for data stream processing	No	One or more per data landing zone
<a href="#">Data product template - analytics and data science</a> ↗	Additional services necessary for data analytics and AI	No	One or more per data landing zone

These templates contain Azure Resource Manager templates, the templates' parameter files, and CI/CD pipeline definitions for resource deployment.

Templates can change over time due to new Azure services and requirements. Secure each repository's main branch so it remains error-free and ready for consumption and deployment. Use a development subscription to test template configuration changes before you merge feature enhancements back into your main branch.

## Connect to environments privately

The reference architecture is secure by design. It uses a multilayered security approach to overcome common data exfiltration risks.

The most simple security solution is to [host a jumpbox](#) on the virtual network of the data management landing zone or data landing zone to connect to the data services through private endpoints.

## Frequently asked questions

For a list of questions and answers about cloud-scale analytics, see [Frequently asked questions](#).

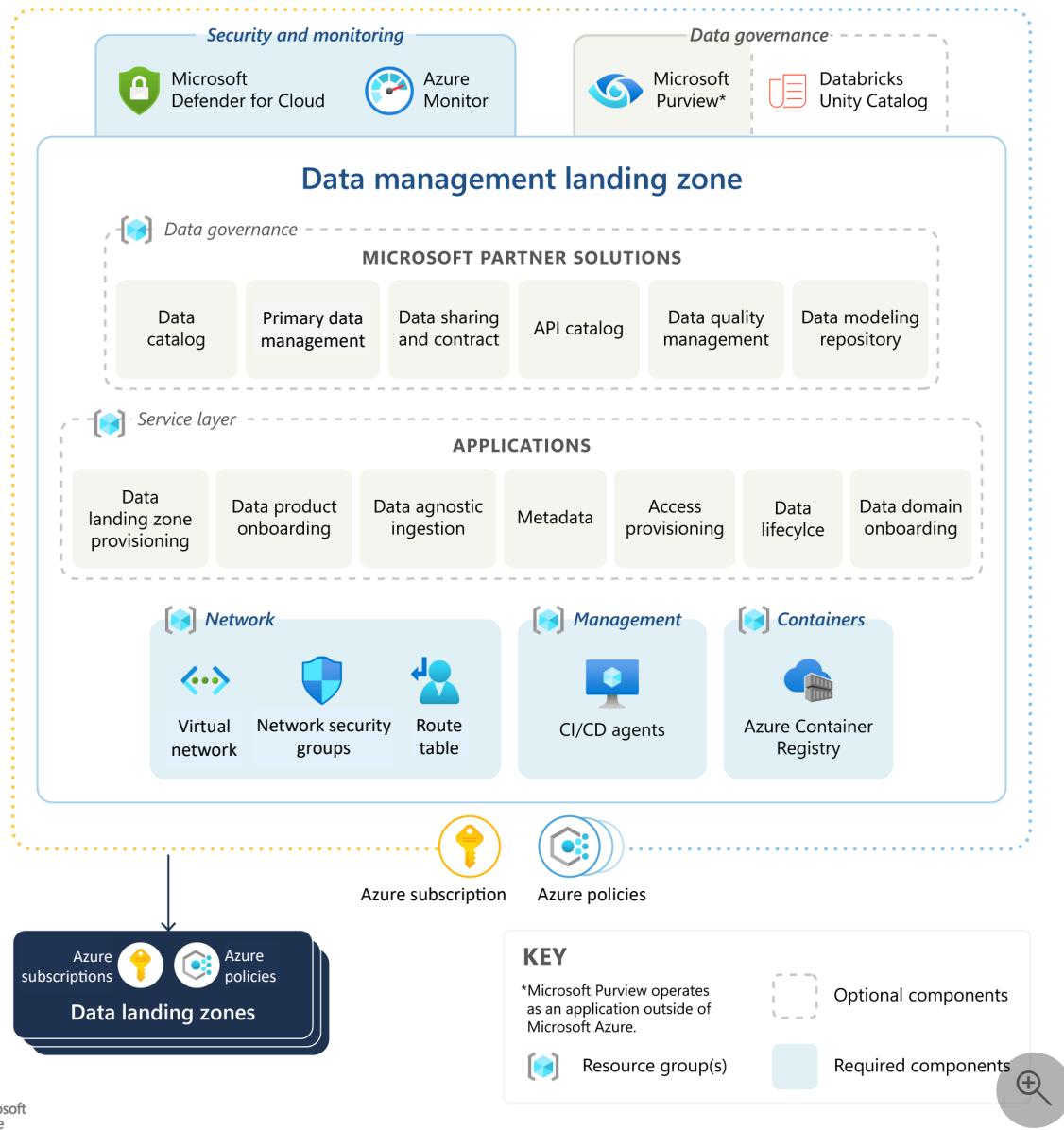
## Next steps

[Cloud-scale analytics data management landing zone overview](#)

# Data management landing zone

Article • 02/21/2025

A data management landing zone is essential for cloud-scale analytics. It oversees the governance of your entire analytics platform.



A data management landing zone is a separate subscription that has the same standard Azure landing zone services. It provides data governance through crawlers, which connect to data lakes and polyglot storage in data landing zones. Virtual network peering connects the data management landing zone to the data landing zones and connectivity subscription.

This architecture is a starting point. You can modify it to fit your specific business and technical requirements when you plan your data management landing zone.

implementation.

### Note

*Polyglot persistence* refers to the practice of using multiple data storage or data store technologies to support your data types and their storage needs. Polyglot persistence means that an application can use more than one core database or storage technology.

### Important

You must deploy your data management landing zone as a separate subscription under a management group that has the appropriate governance. Then you can control governance across your organization. The [Azure landing zone accelerator](#) describes how you should approach Azure landing zones.

## Data governance

The Azure cloud-scale analytics framework suggests that you use Microsoft Purview. Alternatively, you can deploy non-Microsoft solutions to manage specific data governance functions.

Consider the following key functions in your architecture:

- A global data catalog
- Primary data management
- Data sharing and contracts
- An API catalog
- Data quality management
- A data modeling repository

If you have partner data governance products that require deployment in a subscription, deploy them to the data governance resource group within the data management landing zone.

## Data catalog

A data catalog registers and maintains data information in a centralized place so that it's available for your organization. It minimizes the chance of different project teams ingesting redundant data, which prevents duplicate data products. We recommend that

you create a data catalog service to define the metadata of data products that you store across data landing zones.

Cloud-scale analytics relies on [Microsoft Purview](#) to register enterprise data sources, classify them, ensure data quality, and provide highly secure, self-service access.

Microsoft Purview is a tenant-based service that can communicate with each data landing zone. It creates a managed virtual network and deploys it to your data landing zone region. You can deploy Azure managed virtual network integration runtimes (IR) within these managed virtual networks in any available Microsoft Purview region. The managed virtual network IR can then use private endpoints to securely connect to and scan the supported data sources. This approach helps isolate and secure the data integration process. For more information, see [Use managed virtual networks with your Microsoft Purview account](#).

If you use Azure Databricks, we recommend using [Azure Databricks Unity Catalog](#) in addition to Microsoft Purview. Unity Catalog provides centralized access control, auditing, lineage, and data discovery capabilities across Databricks workspaces. For more information, see [Unity Catalog best practices](#).

 **Note**

This article focuses on using Microsoft Purview for governance, but your enterprise might have investments in other products, such as Alation, Okera, or Collibra. These solutions are subscription-based. We recommend that you deploy them to the data management landing zone. They might require custom integration.

## Primary data management

Primary data management control resides in the data management landing zone. For specific data mesh considerations, see [Primary data management in data mesh](#).

Many primary data management solutions fully integrate with Microsoft Entra ID, which helps secure your data and provide different views for different user groups. For more information, see [Primary data management system](#).

## Data sharing and contracts

Cloud-scale analytics uses [Microsoft Entra entitlement management](#) or [Microsoft Purview policies](#) to control access to data sharing. In addition to those features, you might require a sharing and contract repository. This repository is an organizational

function and should reside in your data management landing zone. Your contracts should provide information about data validation, models, and security policies. For more information, see [Data contracts](#).

## API catalog

Your data application teams create various APIs for their data applications, which can be hard to find across your organization. To address this problem, place an API catalog in your data management landing zone.

An API catalog standardizes your documentation, facilitates internal collaboration, and enhances consumption, publishing, and governance controls across your organization.

## Data quality management

Use your existing data quality management practices. To prevent problems from spreading across your analytics and AI systems, manage data quality at the data source.

Integrate quality metrics and validation into your data processes so that the teams most familiar with the data handle quality management. This approach helps ensure that your team has a deeper understanding and better handling of the data assets. Provide data lineage for all data products to improve data quality confidence.

For more information, see [Data quality](#).

## Data modeling repository

Store entity relationship models centrally within your data management landing zone so that data consumers can easily find conceptual diagrams. To model your data products before ingestion, use tools like [ER/Studio](#) and [OrbusInfinity](#).

## Service layer

Your organization might create several automation services to augment cloud-scale analytics capabilities. These automation services drive conformity and onboarding solutions for your analytics state.

If you build these automation services, a user interface should serve as both a data marketplace and an operation console. This interface should rely on an underlying metadata store, such as [metadata standards](#).

Your data marketplace or operations console calls a middle tier of microservices to facilitate onboarding, metadata registration, security provisioning, data lifecycle, and observability. You can provision the service layer resource group to host your metadata store.

**ⓘ Important**

The following automation services aren't actual products that you can purchase. And they don't represent future releases or updates. Use the following list to help you consider which items to automate.

[+] Expand table

Type of service	Service scope
Data landing zone provisioning	This service creates a new data landing zone. This service is infrequently used, but it ensures end-to-end onboarding solution completeness. For more information, see <a href="#">Provision cloud-scale analytics</a> .
Data product onboarding	This service creates and amends resource groups that pertain to an onboarded tenant. It also contains capabilities to upgrade and downgrade SKUs and to activate and deactivate resource groups for onboarded tenants or services. This service also creates a new data landing zone for DevOps purposes. For more information, see <a href="#">Provision cloud-scale analytics</a> .
Data agnostic ingestion	This microservice creates new data sources for ingestion into your data landing zones. To manage this process, it communicates with an Azure Data Factory and Azure SQL Database metastore that's located in each data landing zone. For more information, see <a href="#">How automated ingestion frameworks support cloud-scale analytics in Azure</a> .
Metadata	This service exposes and creates metadata for the platform. For more information, see <a href="#">Metadata standards</a> .
Access provisioning	This service uses a service principal name or user principal name to create access packages, access policies, and manual or automatic asset access approval processes. It can also expose an API to provide a list of subscription requests (or assets) that users submit in the last 90 days. For more information, see <a href="#">Data access management</a> .
Data lifecycle	This service helps maintain your data lifecycle based on metadata. This maintenance can include moving data to cold storage and deleting outdated records. For more information, see <a href="#">Data lifecycle management</a> .
Data domain onboarding	This service is only applicable to data mesh. This service captures new domain metadata and onboards the new domains as needed. It can also create, update,

Type of service	Service scope
	activate, and deactivate domain or service lines that you build into a microservice. For more information, see <a href="#">Provision cloud-scale analytics</a> .

## Azure Container Registry

Your data management landing zone hosts an Azure Container Registry instance. Data platform operations can use Container Registry to deploy standard containers for data science projects that your data application teams consume.

## Next step

[Overview of data landing zones](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Data landing zones

Article • 02/17/2025

Data landing zones are connected to your [data management landing zone](#) by virtual network peering or private endpoints. Each data landing zone is considered a [landing zone](#) related to Azure landing zone architecture.

## ⓘ Important

Before you provision a data landing zone, ensure that your DevOps and continuous integration and continuous delivery (CI/CD) operating model is in place and that a data management landing zone is deployed.

Each data landing zone has several layers that enable agility for the service data integrations and data applications it contains. You can deploy a new data landing zone with a standard set of services that allow the data landing zone to ingest and analyze data.

The following table shows the structure of a typical Azure subscription associated with a data landing zone.

[+] Expand table

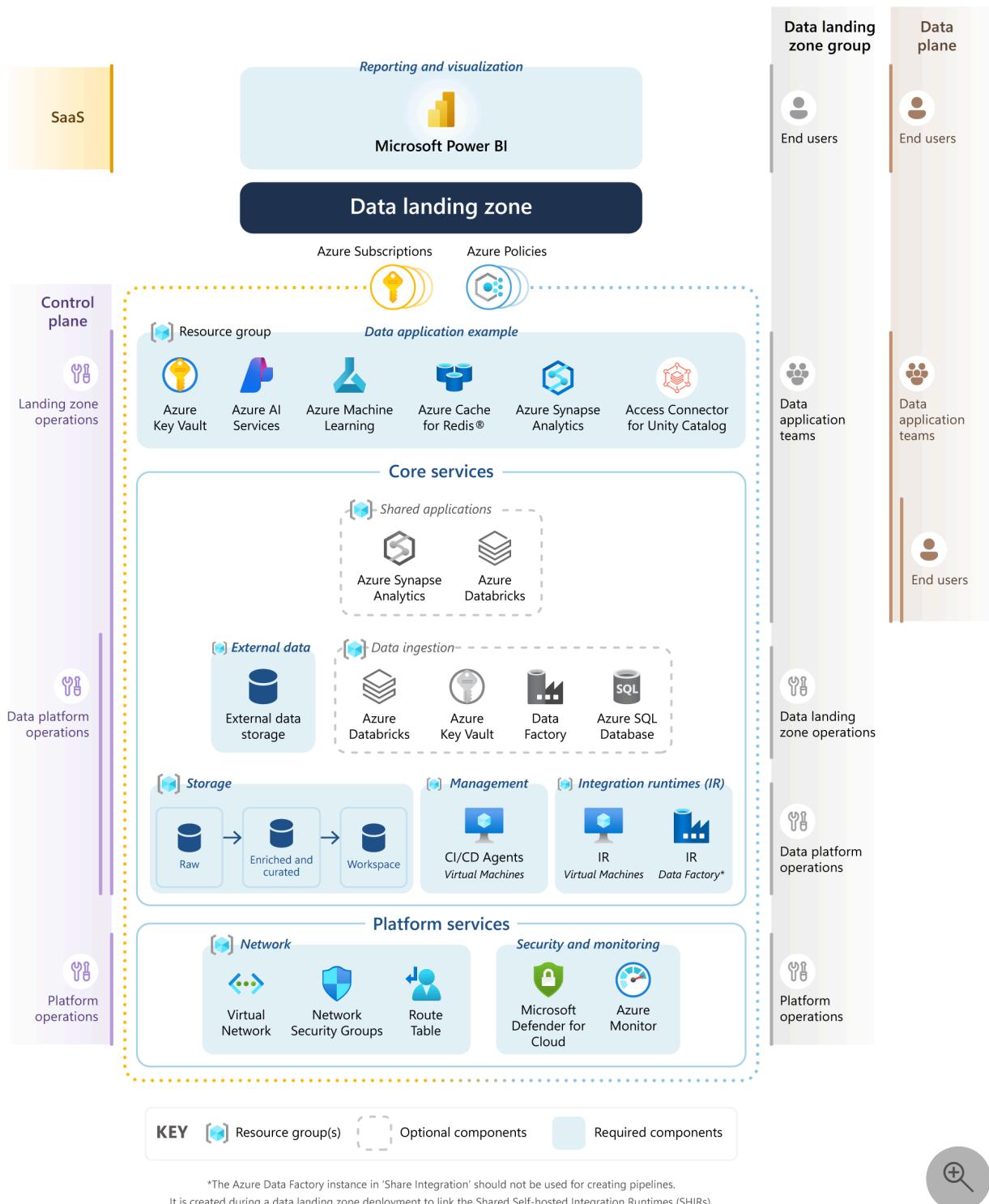
Layer	Required	Resource groups
Platform services layer	Yes	<ul style="list-style-type: none"><li>• Network</li><li>• Security</li></ul>
Core services	Yes	<ul style="list-style-type: none"><li>• Storage</li><li>• Shared integration runtimes (IRs)</li><li>• Management</li><li>• External storage</li><li>• Data ingestion</li><li>• Shared applications</li></ul>
Data application	Optional	<ul style="list-style-type: none"><li>• <a href="#">Data application</a> (one or more)</li></ul>
Reporting and visualization	Optional	<ul style="list-style-type: none"><li>• <a href="#">Reporting and visualization</a></li></ul>

## ⓘ Note

The core services layer is marked as required, but not all resource groups and services included in this article might be necessary for your data landing zone.

## Data landing zone architecture

The following data landing zone architecture illustrates the layers, their resource groups, and the services that each resource group contains. The architecture provides an overview of all groups and roles associated with your data landing zone and the extent of their access to your control and data planes. The architecture also illustrates how each layer aligns with the operating model responsibilities.



### 💡 Tip

Before you deploy a data landing zone, make sure to [consider the number of initial data landing zones that you want to deploy](#).

## Platform services

The platform services layer includes services required to enable connectivity and observability to your data landing zone within the context of cloud-scale analytics. The

following table lists the recommended resource groups.

[+] Expand table

Resource group	Required	Description
network-rg	Yes	Networking
security-rg	Yes	Security and monitoring

## Networking

The network resource group contains connectivity services, including [Azure Virtual Network](#), [network security groups](#), and [route tables](#). All these services are deployed into a single resource group.

The virtual network of your data landing zone is [automatically peered with your data management landing zone's virtual network](#) and your [connectivity subscription's virtual network](#).

## Security and monitoring

The security and monitoring resource group includes [Azure Monitor](#) and [Microsoft Defender for Cloud](#) to collect service telemetry, define monitoring criteria and alerts, and apply policies and scanning to services.

## Core services

The core services layer includes foundational services required to enable your data landing zone within the context of cloud-scale analytics. The following table lists the resource groups that provide the standard suite of available services in every data landing zone that you deploy.

[+] Expand table

Resource group	Required	Description
storage-rg	Yes	Data lake services
runtimes-rg	Yes	Shared IRs
mgmt-rg	Yes	CI/CD agents

Resource group	Required	Description
external-data-rg	Yes	External data storage
data-ingestion-rg	Optional	Shared data ingestion services
shared-applications-rg	Optional	Shared applications (Azure Synapse Analytics or Azure Databricks)

## Storage

The previous diagram shows three [Azure Data Lake Storage Gen2](#) accounts provisioned in a single data lake services resource group. Data transformed at different stages is saved in one of your data landing zone's data lakes. The data is available for consumption by your analytics, data science, and visualization teams.

Data lake layers use different terminology depending on technology and vendor. This table provides guidance on how to apply terms for cloud-scale analytics:

[+] [Expand table](#)

Cloud-scale analytics	Delta Lake	Other terms	Description
Raw	Bronze	Landing and conformance	Ingestion tables
Enriched	Silver	Standardization zone	Refined tables. Stored full entity, consumption-ready recordsets from systems of record.
Curated	Gold	Product zone	Feature or aggregated tables. Primary zone for applications, teams, and users to consume data products.
Development	--	Development zone	Location for data engineers and scientists, which consists of an analytics sandbox and a product development zone.

### ⓘ Note

In the previous diagram, each data landing zone has three data lake storage accounts. Depending on your requirements, you can choose to consolidate your raw, enriched, and curated layers into one storage account and maintain another

storage account called *workspace* for data consumers to bring in other useful data products.

For more information, see:

- [Overview of Azure Data Lake Storage for cloud-scale analytics](#)
- [Data standardization](#)
- [Data lake zones and containers](#)
- [Key considerations for Data Lake Storage](#)
- [Access control and data lake configurations in Data Lake Storage](#)

## Shared IRs

Azure Data Factory and Azure Synapse Analytics pipelines use IRs to securely access data sources in peered or isolated networks. Shared IRs should be deployed to a virtual machine (VM) or Azure Virtual Machine Scale Sets in the shared IR resource group.

To enable the shared resource group:

- Create at least one Azure Data Factory instance in your data landing zone's shared integration resource group. Use it only for linking the shared self-hosted IR, not for data pipelines.
- [Create and configure a self-hosted IR on the VM.](#)
- Associate the self-hosted IR with Azure data factories in your data landing zones.
- Use PowerShell scripts to [periodically update the self-hosted IR](#).

### Note

The deployment describes a single VM deployment that has a self-hosted IR. You can associate a self-hosted IR with multiple VMs on-premises or in Azure. These machines are called nodes. You can have up to four nodes associated with a self-hosted IR. The benefits of having multiple nodes include:

- Higher availability of the self-hosted IR so that it's no longer the single point of failure in your data application or in the orchestration of cloud data integration.
- Improved performance and throughput during data movement between on-premises and cloud data services. For more information, see [Copy activity](#)

## [performance and scalability guide](#).

You can associate multiple nodes by installing the self-hosted IR software from [Microsoft Download Center](#). Then register it by using either of the authentication keys obtained from the **New-AzDataFactoryV2IntegrationRuntimeKey** cmdlet, as described in the [tutorial](#).

For more information, see [Azure Data Factory high availability and scalability](#).

Make sure to deploy shared IRs as close to the data source as possible. You can deploy the IRs in a data landing zone, into non-Microsoft clouds, or into a private cloud if the VM has connectivity to the required data sources.

## Management

CI/CD agents run on VMs and help deploy artifacts from the source code repository, including data applications and changes to the data landing zone.

For more information, see [Azure Pipelines agents](#).

## External storage

Partner data publishers need to land data in your platform so that your data application teams can pull it into their data lakes. You can also have internal or external data sources that can't support the connectivity or authentication requirements enforced across the rest of the data landing zones. The recommended approach is to use a separate storage account to receive data. Then use a shared IR or similar ingestion process to bring it into your processing pipeline.

The data application teams request the storage blobs. These requests get approved by the data landing zone operations team. Data should be deleted from its source storage blob after it's ingested into the raw data storage.

### **ⓘ Important**

Because Azure Storage blobs are provisioned on an *as-needed* basis, you should initially deploy an empty storage services resource group in each data landing zone.

## Data ingestion

This resource group is optional and doesn't prevent you from deploying your landing zone. It applies if you have, or are developing, a data-agnostic ingestion engine that automatically ingests data based on registered metadata. This feature includes connection strings, paths for data transfer, and ingestion schedules.

The ingestion and processing resource group has key services for this kind of framework.

Deploy an Azure SQL Database instance to hold metadata that Azure Data Factory uses. Provision an Azure key vault to store secrets related to automated ingestion services. These secrets can include:

- Azure Data Factory metastore credentials.
- Service principal credentials for your automated ingestion process.

For more information, see [Data agnostic ingestion engine](#).

The following table describes services in this resource group.

[+] Expand table

Service	Required	Guidelines
Azure Data Factory	Yes	Azure Data Factory is your orchestration engine for data-agnostic ingestion.
Azure SQL Database	Yes	SQL Database is the metastore for Azure Data Factory.
Azure Event Hubs or Azure IoT Hub	Optional	Event Hubs or IoT Hub can provide real-time streaming to event hubs, plus batch and streaming processing via an Azure Databricks engineering workspace.
Azure Databricks	Optional	You can deploy Azure Databricks or Azure Synapse Spark to use with your data-agnostic ingestion engine.
Azure Synapse	Optional	You can deploy Azure Databricks or Azure Synapse Spark to use with the data-agnostic ingestion engine.

## Shared applications

Use this optional resource group when you need to have a set of shared services made available to all the teams building data applications in this data landing zone. Use cases include:

- An Azure Databricks workspace used as a shared metastore for all other Databricks workspaces created in the same data landing zone or region.
- A shared Azure Synapse Analytics instance that uses serverless SQL pools to enable users to query across isolated storage accounts.

#### Note

Azure Databricks uses Unity Catalog to govern access and visibility to metastores across Databricks workspaces. Unity Catalog is enabled at a tenant level, but metastores are aligned with Azure regions. This setup means that all Unity Catalog-enabled Databricks workspaces in a given Azure region must register with the same metastore. For more information, see [Unity Catalog best practices](#).

To integrate Azure Databricks, follow cloud-scale analytics best practices. For more information, see [Secure access to Azure Data Lake Gen2 from Azure Databricks](#) and [Azure Databricks best practices](#).

## Data application

Each data landing zone can have multiple data applications. You can create these applications by ingesting data from various sources. You can also create data applications from other data applications within the same data landing zone or from other data landing zones. The creation of data applications is subject to data steward approval.

## Data application resource group

Your data application resource group includes all the services required to make the data application. For example, an Azure Database is required for MySQL, which is used by a visualization tool. Data must be ingested and transformed before it goes into that MySQL database. In this case, you can deploy Azure Database for MySQL and Azure Data Factory into the data application resource group.

#### Tip

If you decide not to implement a data-agnostic engine for single ingestion from operational sources, or if complex connections aren't supported in your data-agnostic engine, develop a source-aligned [data application](#).

# Reporting and visualization

You can use visualization and reporting tools within Fabric workspaces, which are similar to Power BI workspaces. This feature allows you to avoid deploying unique resources within your data landing zone. You can include a resource group to deploy Fabric capacity, VMs for data gateways, or other necessary data services to deliver your data application to the user.

## Next step

[Cloud-scale analytics data products in Azure](#)

---

## Feedback

Was this page helpful?



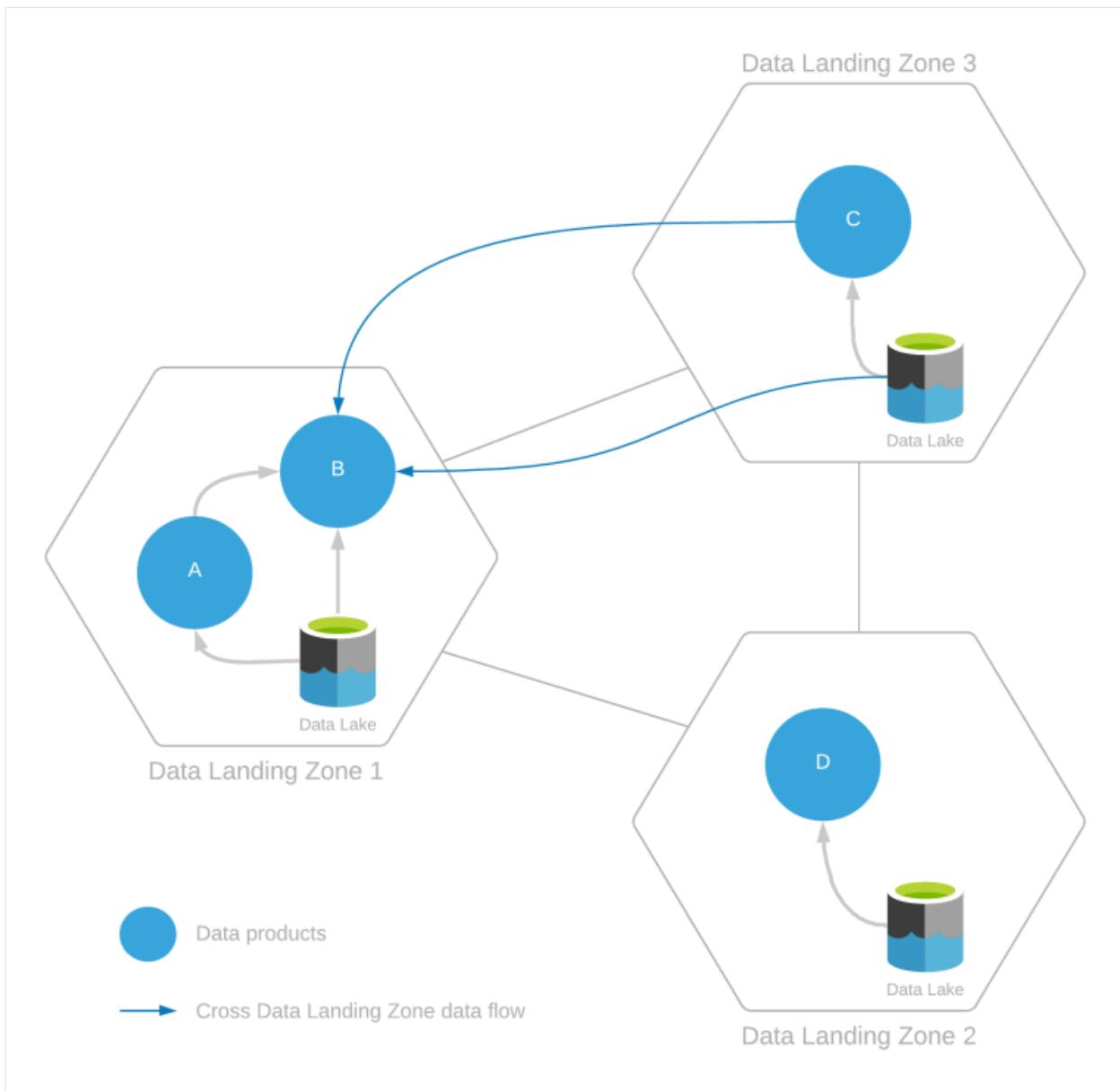
# Cloud-scale analytics data products in Azure

Article • 12/10/2024

[Data products](#) are data served as products and computed, saved, and served by polyglot persistence services, which may be required by certain use cases. The process of creating and serving a data product can require services and technologies that aren't included in the [data landing zone](#) core services. An example of this would be reporting with niche requirements, such as compliance and tax reporting.

## Design considerations

A data landing zone can serve multiple data products created by ingesting data from within the same data landing zone or from across multiple data landing zones. This is shown in the following diagram.



The example above shows:

- Intrazone data consumption:
  - Data product B consumes data from data product A and other data or data products existing in the data lake within its own landing zone.
  - Data products C and D only consume data from within their own respective data landing zones.
- Interzone data consumption:
  - Data product B also consumes data from data product C and the data in landing zone 3's data lake.

### ⓘ Important

In the case of interzone data consumption, since data product B is created by reading from data landing zone 3, this read access requires approval from the [data landing zone operations](#) and [integration operations](#) teams of data landing zone 3.

## ⓘ Important

Data product B consumes data from data products A and C. Before this can happen, data product B must register its consumption of data products via data sharing agreements. This data sharing agreement should update the lineage from data product A to data product B and from data product C to data product B.

The resource group for a data product includes all services required to create and maintain it. We can call this resource group a **data application**. Examples of services that might be part of a data application include Azure Functions, Azure App Service, Logic Apps, Azure Analysis Services, Azure Cognitive Services, Azure Machine Learning, Azure SQL Database, Azure Database for MySQL, and Azure Cosmos DB.

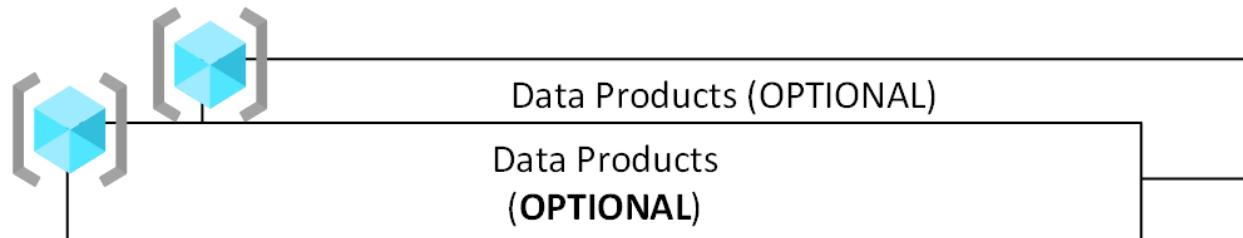
Data products have data from *READ* data sources that have had some data transformations applied. Examples might be a newly curated dataset or a BI report.

## Design recommendations

Build data products within your data landing zone by adhering to design principles that allow you to scale with data governance. The following sections provide design recommendations to help as you plan your data application ecosystem.

## Deploy multiple resource groups

Each data application is a resource group. Since data applications are compute services, polyglot persistence services, or both, they can only be required depending on certain use cases. As such, they're considered an optional data landing zone component. In cases where you do need data applications, create multiple resource groups by data application as the following diagram shows.



## Set guardrails

Azure Policy drives the default configuration of services within a data landing zone. Think of operational analytics as multiple resource groups that your data product team

can request from a standard service catalog. Using Azure Policy, you can configure the security boundary and required feature set.

**ⓘ Important**

To drive consistency, configure one Azure Policy for each data application.

## Consume data from multiple places

Data applications manage, organize, and make sense of data from multiple data assets and present any insights gained. A data product is the result of data from one or many data applications within data landing zones. Allow your data applications to access data from multiple and various sources when necessary.

## Scale as needed

Services that make up data applications are incremental deployments to the data landing zone. Scale your data applications as needed.

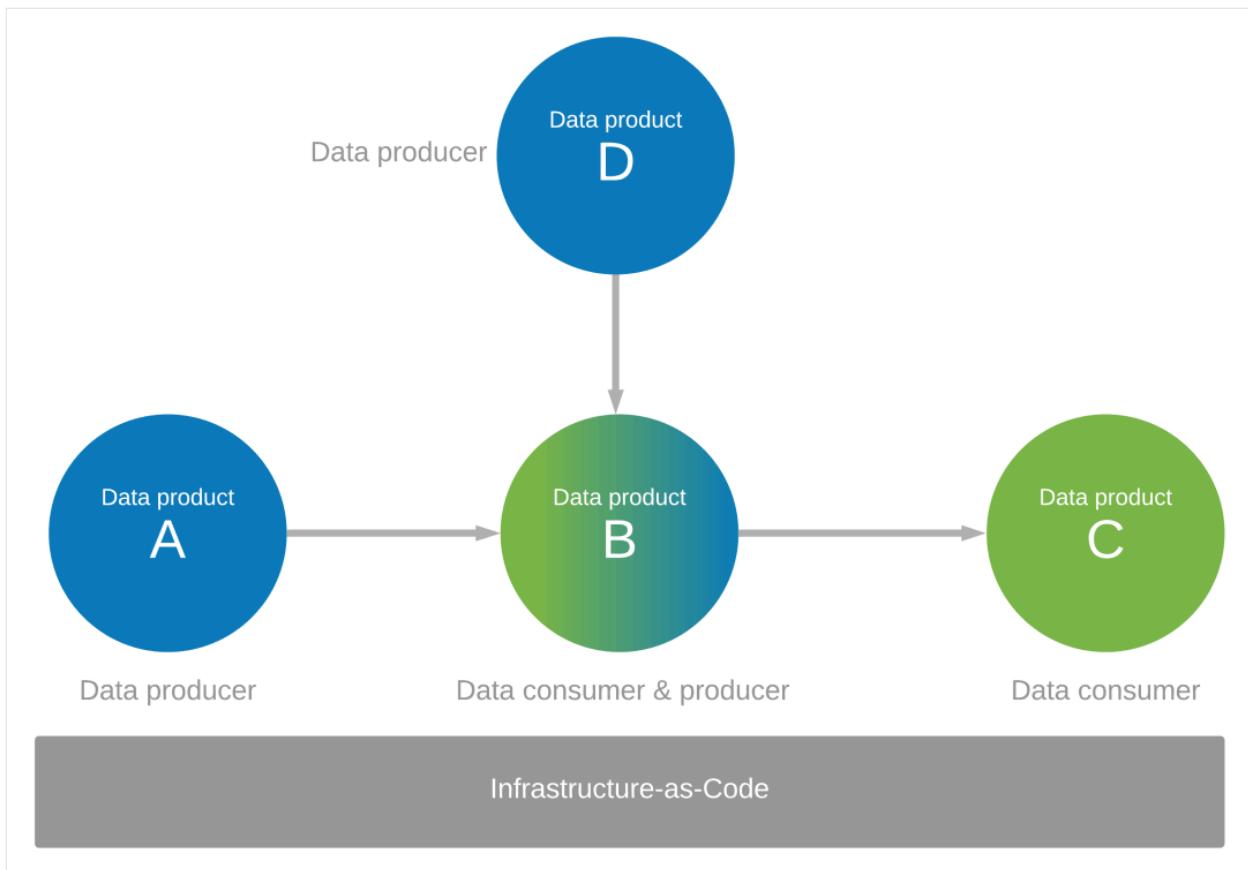
## Enable data discovery

Automatically register your data products in a data catalog such as [Microsoft Purview](#) to allow data scanning.

## Identify your data products

While starting to plan a data landing zone, identify as many data products (and the data applications that output and maintain them) as necessary to help drive your data product application architecture. Conformity to implemented platform governance should play the largest role in your decisions.

Focus on how your data applications are data producers and consumers for others. For example, assume you've identified a suite of data products (A, B, C, and D) which are produced and consumed data. You require data products A and D as sources for the data in Data Application B for data product B. Data product B is created from the data that Data Application B consumes from data products A and D. Data Application B acts as a data producer itself, and also produces data for data product C.



## Control your data application environment with infrastructure-as-code

Governance and infrastructure-as-code should control the data application environment across your data products ecosystem, as shown in the previous diagram.

## Publish data models

Your data product teams should publish their data models in a modeling repository.

## Set expectations for data product users

Update your data sharing contracts with service-level agreements and certifications for your data products so you can convey accurate expectations to potential users of the data product.

## Capture lineage

If data product B is created from data coming from data products A and D, lineage must be captured from A and D to B. Further lineage should also be captured for data product C, since it's created using data from data product B. Updated lineage should be captured in a data lineage application before every release of your data product.

### Note

Using Azure Pipelines allows you to build approval gates and invoke functions that can ensure metadata, lineage, and SLAs are registered in the correct governance service.

## Define data application architecture

You must create a detailed architecture for each data product that fully defines its relationship to other data products, its dependencies, and its access requirements.

## Next steps

[Data applications \(source-aligned\)](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Data applications (source-aligned)

Article • 12/10/2024

If you choose not to implement a data agnostic engine for ingesting data once from operational sources, or if complex connections aren't facilitated in your data agnostic engine, you should create a data application that is source-aligned. It should follow the same flow as a data agnostic engine would when ingesting data from external data sources.

## Overview

Your application resource group is responsible for data ingestion and enrichment only from external sources, such as telemetry, finance, or CRM. This layer can operate in real-time, batch, and micro-batch.

This section explains the infrastructure deployed for each data application (source-aligned) resource group inside your data landing zone.

### 💡 Tip

For data mesh, you can choose to deploy one of these per source or one per domain. The principles of data standardization, data quality, and data lineage must still be followed. Data platform ops teams can develop snippets of standard code and call upon them to achieve this.

For each data application (source-aligned) resource group in your data landing zone, you should create:

- An Azure Key Vault
- An Azure Data Factory to run developed engineering pipelines that transform data from raw to enriched
- A service principal used by the data application (source-aligned) for deploying ingest jobs to Azure Databricks (only if using Azure Databricks)

You can also create instances of other services, such as Azure Event Hubs, Azure IoT Hub, Azure Stream Analytics, and Azure Machine Learning.

## Azure Key Vault

Use Azure Key Vault functionality to store secrets within Azure whenever possible.

Each data application (source-aligned) resource group or data domain (if mesh) has an Azure Key Vault which:

- Ensures that the encryption key, secret, and certificate derivation meet the requirements of your environment
- Allows better separation of administrative duties
- Reduces the risk of mixing keys, integrations, and secrets of differing classifications

All keys related to your data application (source-aligned) should be contained in your Azure Key Vault.

 **Important**

Data application (source-aligned) key vaults should follow the least-privilege model and should avoid both transaction scale limits and secret-sharing across environments.

## Azure Data Factory

Deploy an Azure Data Factory to allow pipelines written by your data application team to take data from raw to enriched using developed pipelines. Use mapping data flows for transformations, and break out to use either Azure Databricks, Azure Synapse Spark, or Microsoft Fabric for complex transformations.

You should connect Azure Data Factory to the DevOps instance of your data application (source-aligned) repo. This connection allows CI/CD deployments.

## Event Hubs

If your data application (source-aligned) has a requirement to stream data in, you can deploy downstream Event Hubs in your data application (source-aligned) resource group.

## Next steps

[Data application reference patterns](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Data application reference patterns

Article • 12/10/2024

When onboarding a data application onto a Data Landing Zone, the team will be granted access to their dedicated resource group, subnet, and shared resources. From this point, the ownership of the environment is handed over to the data application team. These teams have to take responsibility from an end-to-end implementation and cost ownership perspective.

To simplify the process of getting started and reduce the lead time to create an environment for a specific use case, organizations can provide internal reference patterns. These reference implementations consist of the Infrastructure as Code (IaC) definitions to successfully create a set of services for a specific use case, such as batch data processing, streaming data processing, or data science, and demonstrate a path to success. Potentially, these patterns also include generic application code that can be used as a baseline when implementing data solutions. Data application reference patterns could vary between organizations and highly depend on the utilized tools and commonly and repeatedly used data implementation patterns across Data Landing Zones.

Other automation can be used to further reduce any potential friction points and automate even the initial deployment of the pattern for data application teams. For more details, please take a look at [Platform automation and DevOps for a cloud-scale analytics](#).

Ultimately, the goal should be to hand over these reference implementations to the data application teams, as they should own the overall codebase of their solution. Extra abstraction layers, such as Azure template specs, are also an option but just increase the number of friction points as required changes again need to be requested from a central team that owns and maintains these resources. The central team then needs to take action to get the changes tested and released. Additionally, a more complex release management process could be required to not impact other consumers of the Template Spec. Lastly, the templates will become more complex over time as each team could require different parameters to be exposed to apply certain changes within the template. Hence, handing over the reference patterns is the easiest and most effective solution, as this allows the data application teams to make the necessary changes if they need to. Exposing these teams to the concept of IaC is a good approach that could take some time but ultimately will result in better engineering practices across the data platform.

For more information, see [Scaling Cloud-scale analytics](#).

---

# Feedback

Was this page helpful?

 Yes

 No

# Scale cloud-scale analytics in Azure

Article • 11/12/2024

A scalable data platform is critical for accommodating the rapid growth of data. Vast amounts of data are generated every second around the world. The amount of available data is expected to continue growing exponentially over the next few years. As the rate of data generation increases, the speed of data movement also increases.

No matter how much data you have, your users demand fast query responses. They expect to wait minutes, not hours, for results. This article explains how you can scale your Azure cloud-scale analytics solution and continue to meet user demands for speed.

## Introduction

Many enterprises have large data platform monoliths. These monoliths are built around a single Azure Data Lake Gen2 account, and sometimes a single storage container. A single Azure subscription is often used for all data platform-related tasks. Subscription-level scaling is absent in most architectural platforms, which can hinder continued Azure adoption if users run into any of the [Azure subscription or service-level limitations](#). Even though some of the constraints are soft limits, hitting them can still have a significant negative effect on your data platform.

When you structure your data platform, consider the structure of your organization. Note the data ownership and functional responsibilities of your teams. If your organization gives teams large degrees of autonomy and distributed ownership, a data mesh architecture is your best option.

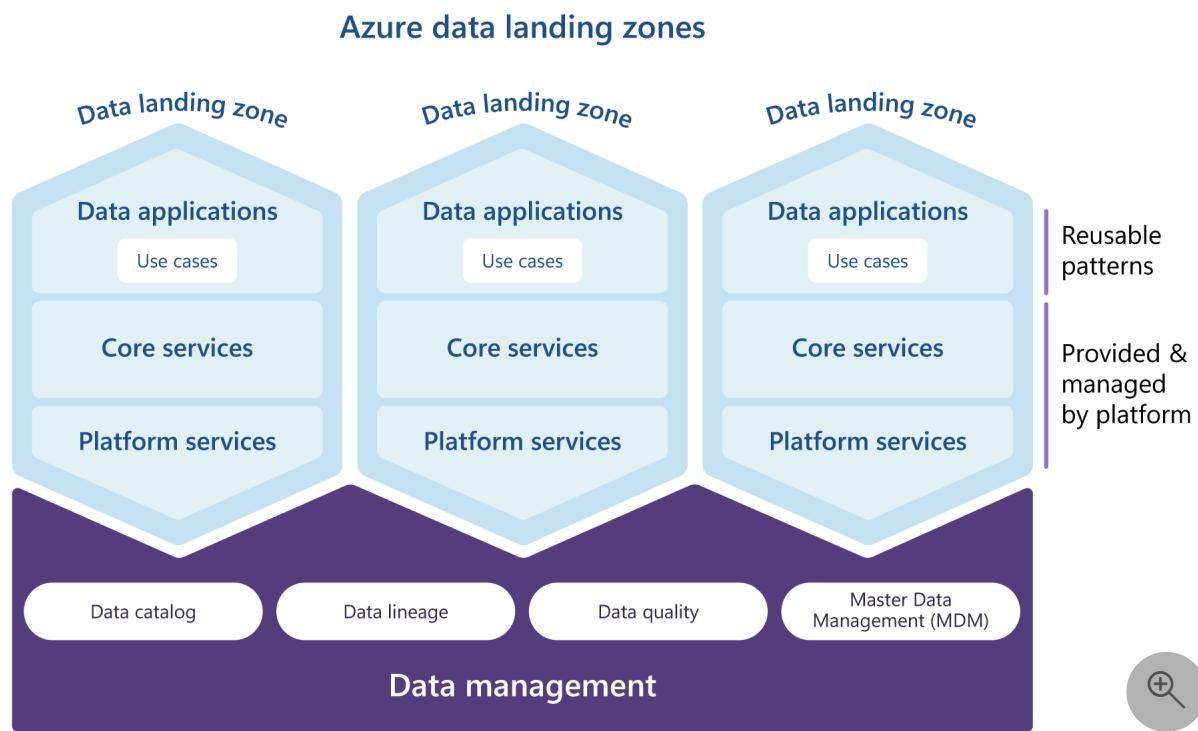
Avoid situations where different teams are responsible for various tasks of a solution—tasks such as ingestion, cleansing, aggregation, and serving. Depending on multiple teams can cause a dramatic loss of velocity. For example, if your data consumers on the serving layer need to onboard new data assets or implement functional changes for a particular data asset, they must go through a multi-step process. For this example, the steps are:

1. The data consumer submits a ticket to every team responsible for a data pipeline stage.
2. The teams must work together in sync because the layers are interconnected. The new services require changes to the data cleansing layer, which leads to changes in the data aggregation layer, which leads to changes in the serving layer. The changes can affect every pipeline stage.

3. It's difficult for the teams to see the potential effects of processing changes because they don't have an overview of the entire end-to-end lifecycle. They must work together to design a well-defined release plan that minimizes effects on existing consumers and pipelines. This dependency management increases management overhead.
4. As a rule, the teams aren't subject matter experts on the data asset that the data consumer requests. To understand new dataset features or parameter values, they have to consult an expert.
5. After all changes are implemented, the data consumer is notified that the new data asset is ready to use.

Each large organization has thousands of data consumers. A complicated process like the one described severely decreases velocity in large architectures since centralized teams become a bottleneck for business units. The result is less innovation and limited effectiveness. Potentially, business units can decide to leave the service and build their own data platform instead.

## Methods for scaling



Cloud-scale analytics addresses scaling challenges by using two core concepts:

- Data landing zones for scaling
- Data products or data integrations for scaling, to make distributed and decentralized data ownership possible

You can deploy a single data landing zone or multiple ones. Data landing zones make it possible for you to discover and manage data by connecting to a data management landing zone. Each data management landing zone is within a single Azure subscription.

Subscriptions are Azure's units of management, billing, and scale. They play a critical role in your large-scale Azure adoption plan.

## Scaling with data landing zones

The central concepts of cloud-scale analytics are Microsoft Purview, Azure Databricks Unity Catalog if you're using Azure Databricks, a data management landing zone, and the data landing zone. You should place each in its own Azure subscription. Separating them lets you clearly separate duties, follow the least privilege principle, and partially address the subscription scale issues mentioned earlier. A minimal cloud-scale analytics setup includes a single data landing zone and a single data management landing zone.

However, a minimal setup isn't sufficient for large-scale data platform deployments. Companies build large-scale platforms and make investments to consistently and efficiently scale their data and analytics efforts over time. To overcome subscription-level limitations, cloud-scale analytics uses subscriptions as the unit of scaling, as discussed in [Azure landing zones](#). This technique makes it possible to increase the data platform footprint by adding more data landing zones to the architecture. Adopting this technique also addresses the problem of one Azure Data Lake Gen2 being used for a whole organization since each data landing zone includes three data lakes. Projects and activities from multiple domains can be distributed across more than one Azure subscription, thus providing greater scalability.

Decide how many data landing zones your organization requires before you implement a cloud-scale analytics architecture. Choosing the right solution establishes the foundation for an effective and efficient data platform.

The number of data landing zones required depends on many factors, especially:

- Organizational alignment, such as how many business units need their own data landing zone
- Operational considerations, such as how your organization aligns operational resources and resources that are specific to a business unit.

Using the right data landing zone model minimizes future efforts to move data products and data assets from one landing zone to another. It also helps you effectively and consistently scale big data and analytics efforts in the future.

Consider the following factors when you decide on the number of data landing zones to deploy.

[Expand table](#)

Factor	Description
Organizational structure and data ownership	Consider how your organization is structured and how data is owned in your organization.
Region and location	If you deploy in multiple regions, decide which regions should host the data zones. Be sure to honor all data residency requirements.
Quotas	Subscription quotas aren't capacity guarantees and are applied on a per-region basis.
Data sovereignty	Due to data sovereignty regulations, data must be stored in a specific region and follow region-specific policies.
Azure policies	Data landing zones must follow the requirements of various Azure policies.
Management boundary	Subscriptions provide a management boundary for governance and isolation that clearly separates concerns.
Networking	Each landing zone has a virtual network. Because a virtual network resides in a single region, each new region requires a new landing zone. The virtual networks must be peer virtual networks to enable cross-domain communication.
Limits	A subscription has limits. By having several subscriptions, you can mitigate the dangers of hitting these limits.
Cost allocation	Consider whether shared services like storage accounts that are paid centrally need to be split by business unit or domain. Using a separate subscription creates a boundary for cost allocation. You can achieve the same functionality by using tags.
Data classifications and highly confidential data	Security mechanisms can affect data product development and the usability of a data platform. Consider data classifications and decide whether highly confidential datasets require special treatment, like just-in-time access, customer-managed keys (CMK), fine-grained network controls, or more encryption.
Other legal or security implications	Consider whether there are any other legal or security requirements that require logical or physical separation of data.

If you implement a data mesh architecture, consider the following factors as you decide how to distribute your data landing zones and data domains.

Factor	Description
Data domains	Consider the data domains that your organization uses, and decide the data domains for your data platform. Consider the size of your individual data domains. For more information, see <a href="#">What are data domains?</a>
Latency	Domains that collaborate on large amounts of data can transfer a large amount of data across landing zones. Consider allocating your domains in the same landing zone or region. Separating them increases latency and can increase costs in cross-region domains.
Security	Some service deployments or configurations require elevated privileges in a subscription. Giving these privileges to a user in one domain implicitly gives that user the same privileges in other domains within the same subscription.

You can find more considerations in the cloud adoption framework guidance for [subscriptions](#).

Many organizations want efficient scaling of their enterprise data platform. Business units should be able to build their own data solutions and applications to meet their unique requirements. Providing this ability can be a challenge because many existing data platforms aren't built around the concepts of scalability and decentralized ownership. This shortcoming is clearly seen in the architecture, team structure, and ops model of these data platforms.

Data landing zones don't create data silos within your organization. The recommended network setup for cloud-scale analytics enables secure and in-place data sharing across landing zones, which in turn enables innovation across data domains and business units. To learn more, see [Network architecture considerations](#).

The same is true for the identity layer. When you use a single Microsoft Entra tenant, you can grant identities access to data assets in multiple data landing zones. To learn more about the user and identity authorization process, see [Data access management](#).

### ⓘ Note

If you have multiple data landing zones, each zone can connect to data that's hosted in other zones. This allows groups to collaborate across your business.

Cloud-scale analytics uses a common architecture to advocate consistent governance. Your architecture defines baseline capabilities and policies. All data landing zones adhere to the same auditing and controls. Your teams can create data pipelines, ingest sources, and create data products like reports and dashboards. Teams can also do

Spark/SQL analysis as needed. You can augment data landing zone capabilities by adding services to the capability in the policy. For example, a team can add a third-party graph engine to address a business requirement.

Cloud-scale analytics places a strong emphasis on central cataloging and classification to protect data and make it possible for various groups to discover data products.

### Caution

We recommend against querying data across regions. Instead, make sure that data is close to the compute that uses it, while respecting regional boundaries.

Cloud-scale analytics architecture and the concept of data landing zones make it possible for your organization to easily increase the size of your data platform over time. You can add more data landing zones in a phased approach. Your customers don't need to have multiple landing zones at first. When you adopt this architecture, prioritize a few data landing zones and the data products that they contain. Proper prioritization helps ensure the success of your cloud-scale analytics deployment.

## Scaling with data applications

Within each landing zone, your organization can scale by using data applications. Data applications are units or components of your data architecture that encapsulate functionality that provides read-optimized data products for consumption by other data applications. In Azure, data applications are environments in the form of resource groups that make it possible for cross-functional teams to implement data solutions and workloads. An associated team takes care of the end-to-end lifecycle of the data solution, which includes ingestion, cleansing, aggregation, and serving tasks.

Cloud-scale analytics addresses the data integration and responsibility issues that were discussed earlier. Instead of monolithic functional responsibilities for table ingestion and source system integration, the reference design provides a distributed architecture driven by data domains. Cross-functional teams take over the end-to-end functional responsibility and ownership for the data scope.

Instead of having a centralized technical stack, and a team that's responsible for all tasks of your data processing workflow, you can distribute end-to-end responsibility across multiple autonomous cross-functional data integration teams. Each team owns a domain or subdomain capability and is encouraged to serve datasets as required by data consumers.

These architectural differences lead to increased velocity on your data platform. Your data consumers no longer have to rely on a set of centralized teams or fight for their requested changes to be prioritized. As smaller teams take ownership of your end-to-end integration workflow, the feedback loop between data provider and data consumer is shorter. This approach results in faster prioritization, faster development cycles, and a more agile development process. Your teams no longer need to synchronize processes and release plans among themselves because the cross-functional data integration team has full awareness of the end-to-end technical stack and the implications of changes. It can use software engineering practices to run unit and integration tests to minimize the overall effect on consumers.

Ideally, the team that owns the data integration systems also owns the source systems. This team should consist of data engineers who work on the source systems, subject matter experts (SMEs) for the datasets, cloud engineers, and data product owners. Building this kind of cross-functional team reduces the amount of communication needed with outside teams and is essential while developing your complete stack from infrastructure to actual data pipelines.

The foundation of your data platform is datasets that are integrated from source systems. These datasets make it possible for your data product teams to innovate on business fact tables and to improve decision-making and business processes. Your data integration teams and data product teams should offer SLAs to consumers and ensure that all agreements are met. The offered SLAs can be related to data quality, timeliness, error rates, uptime, and other tasks.

## Summary

Using the scaling mechanisms of your cloud-scale analytics architecture allows your organization to expand its data estate within Azure over time while avoiding common technical limitations. Both of the scaling methods described in this article help you overcome different technical complexities and can be used in a simple and efficient way.

## Next steps

[Data standardization](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Data standardization

Article • 09/16/2022

Data arrives in data lake accounts in various formats. These formats include human readable formats, such as JSON, .CSV, or XML files, and compressed binary formats, such as .tar or .gz. Arriving data also comes in many sizes, from a few edited files to an export of an entire SQL table. Data can also come as a large number of small files that are a few kbs apiece, such as real-time events from an IoT solution.

While Azure Data Lake Storage Gen2 does support storage for all kinds of data without restrictions, you should carefully consider your data formats to ensure processing pipeline efficiency and optimize costs.

Many organizations now standardize their ingest format and separate compute from storage. Because of this, the Delta Lake format has become the preferred standard for data ingestion through to the enrichment layer. From the enrichment layer, your data application team can serve data into a format that reflects their use case.

## ⓘ Note

Use Delta Lake to support both batch and streaming use cases for initial data ingestion through to enrichment layer.

This article provides an overview of Delta Lake, its performance, and how it helps you achieve compliance support, and how to standardize your data as it flows from source to enrichment layer.

## Delta Lake

Delta Lake is an open-source storage layer that brings ACID (atomicity, consistency, isolation, and durability) transactions to big data workloads and Apache Spark. Both Azure Synapse Analytics and Azure Databricks are compatible with Linux Foundation Delta Lake.

## Delta Lake key features

Feature	Description
---------	-------------

Feature	Description
ACID Transactions	Data lakes are typically populated via multiple processes and pipelines, some of which write data concurrently with reads. Data engineers used to go through a manual, error-prone process to ensure data integrity before Delta lake and transactions came into use. Delta Lake brings familiar ACID transactions to data lakes. It provides the strongest isolation level, serializability. For more information, see <a href="#">Diving into Delta Lake: Unpacking the Transaction Log ↗</a> .
Scalable Metadata Handling	In big data, even metadata can be "big data". Delta Lake treats metadata the same as other data. It uses Spark's distributed processing power to handle all metadata. Because of this, Delta Lake can easily handle petabyte-scale tables with billions of partitions and files.
Time Travel (data versioning)	The ability to "undo" a change or go back to a previous version is a key feature of transactions. Delta Lake provides snapshots of data enabling you to revert to earlier versions of data for audits, rollbacks or to reproduce experiments. Learn more in <a href="#">Introducing Delta Lake Time Travel for Large Scale Data Lakes ↗</a> .
Open Format	Apache Parquet, the baseline format for Delta Lake, lets you apply efficient compression and encoding schemes.
Unified Batch and Streaming Source and Sink	A table in Delta Lake is simultaneously a batch table and a streaming source and sink. Data ingest streaming, batch historic backfill, and interactive queries all work out of the box.
Schema Enforcement	Schema enforcement helps you ensure you have correct data types and required columns, which prevents data inconsistency from bad data. For more information, see <a href="#">Diving Into Delta Lake: Schema Enforcement &amp; Evolution ↗</a>
Schema Evolution	Delta Lake lets you make automatically applied changes to a table schema, without needing to write migration DDL. For more information, see <a href="#">Diving Into Delta Lake: Schema Enforcement &amp; Evolution ↗</a>
Audit History	The Delta Lake transaction log records details about every change made to your data. These records provide a complete audit trail of all changes.
Updates and Deletes	Delta Lake supports Scala, Java, Python, and SQL APIs for various functionalities. Merge, update, and delete operations support helps you meet compliance requirements. For more information, see <a href="#">Announcing the Delta Lake 0.6.1 Release ↗</a> , <a href="#">Announcing the Delta Lake 0.7 Release ↗</a> , and <a href="#">Simple, Reliable Upserts and Deletes on Delta Lake Tables using Python APIs ↗</a> (which includes code snippets for merge, update, and delete DML commands).
100% Compatible with Apache Spark API	Your developers can use Delta Lake with minimal change to their existing data pipelines, since it's fully compatible with existing Spark implementations.

For more information, see [Delta Lake Project](#).

For full documentation, visit the [Delta Lake Documentation Page](#)

## Performance

Using lots of small files often results in suboptimal performance and higher costs from increased read/list operations. Azure Data Lake Storage Gen2 is optimized for larger files that allow your analytics jobs to run faster and with lower cost.

Delta Lake includes many features that can help you [Optimize performance with file management](#).

Examples include:

- The transaction log minimizes expensive LIST operations.
- Z-Ordering (multi-dimensional clustering) enables optimized predicate pushdown for your query filters.
- Native caching and query optimizations reduce the amount of storage scanning you require. For more information, see [Optimize performance with caching](#).
- OPTIMIZE coalesces small files into larger ones.

Make these optimizations part of your data loading process to maintain data freshness and performance.

## Data lake partitioning

Data partitioning involves organizing data in your data store so you can manage large-scale data and control data access. Partitioning can improve scalability, reduce contention, and optimize performance.

When partitioning your data lake, ensure your setup:

- Doesn't compromise security
- Has clear isolation and aligns with your data authorization model
- Fits your data ingestion process well
- Has a well-defined path for optimal data access
- Supports management and maintenance tasks

## General practices

The general practices for data partitioning design are:

- Focus on your security implication early, and design your data partitions together with authorization.
- You might want to allow data redundancy in exchange for security.- Define a naming convention and adhere to it.
- You can nest multiple folders, but always keep them consistent.
- Include a time element in your folder structures and file names.
- Don't start your folder structure with date partitions. It's better to keep dates at the lower folder level.
- Don't combine mixed file formats or different data products in a single folder structure.

### Tip

Your folder structures should have partitioning strategies that can optimize access patterns and appropriate file sizes. In the curated zones, plan the structure based on optimal retrieval, be cautious of choosing a partition key with high cardinality, which leads to over partitioning, which in turn leads to suboptimal file sizes.

For more information on data lake zones, see [Data lake zones and containers](#)

## Compliance support

Delta Lake adds a transactional layer to provide structured data management on top of your data lake. This addition can dramatically simplify and speed up your ability to locate and remove personal information (also known as "personal data") upon consumer request. The transactional layer supports operations like DELETE, UPDATE, and MERGE. For more information, see [Best practices: GDPR compliance using Delta Lake](#).

## Summary

Apply the data standardizations listed in this article to your platform. Begin with Delta Lake format, then start adding processes for optimization and compliance. You might decide to create a service that runs some of your optimization routes on a schedule, or create a compliance service that removes personal information.

## Next steps

- [Common Data Model & Industry Data Models](#)

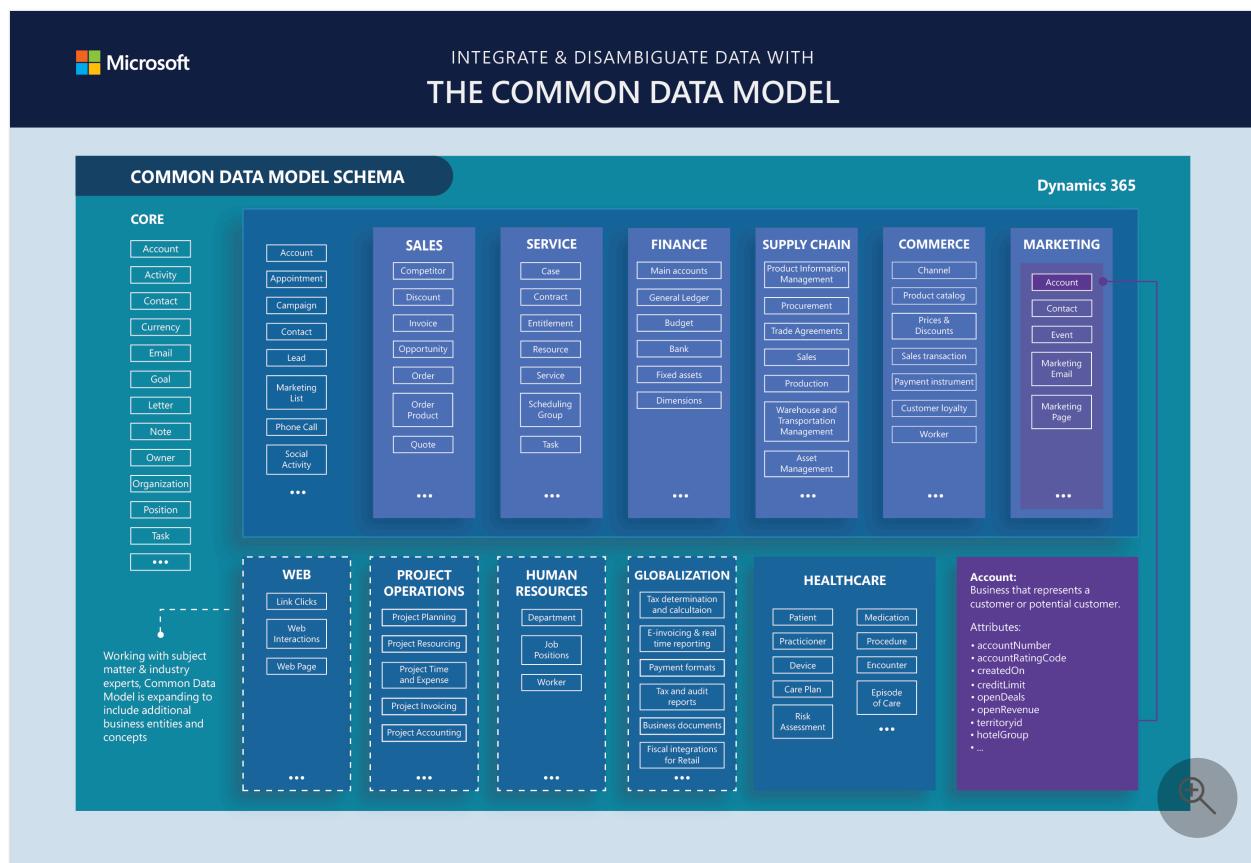
# Common Data Model

Article • 12/10/2024

The Common Data Model (CDM) enables [data product interoperability](#).

The Common Data Model defines a common language for business entities. Over time, this language covers the full range of your business processes across sales, services, marketing, operations, finance, talent, and commerce. The CDM enables data and application interoperability spanning multiple channels, service implementations, and vendors. It also provides data that self-describes (structurally and semantically), allowing applications to easily read and understand the data.

The following poster shows some elements of the [standard entities](#) available in the Common Data Model. You can also [download your own copy](#) of the CDM poster.



For more information, see:

- [Common Data Model](#)
- [Common Data Model repository on GitHub](#).
- [What is a data mesh?](#)

## Feedback

Was this page helpful?

 Yes

 No

# What is a data mesh?

Article • 12/10/2024

Data mesh is an architectural pattern for implementing enterprise data platforms in large and complex organizations. Data mesh helps scale analytics adoption beyond a single platform and a single implementation team.

## Background

The demand for analytics isn't a recent development. Organizations consistently needed to evaluate business performance and utilized computers for this purpose since their inception. Around the 1980s, organizations started to build data warehousing solutions by using databases specifically for decision support. These data warehousing solutions served organizations well for a long time.

However, as business changes and generates more diverse data, data warehousing solutions that use relational databases might not always be the best solution. In the 2000s, big data became a common term. Businesses adopted new solutions that allow analysis of large volumes of diverse data that could be generated with great velocity. These solutions include technology, like data lakes, and scale-out solutions that analyze large quantities of data.

In recent years, many organizations successfully use modern architectural and analytical patterns that combine data warehousing technologies and more recent big data technologies.

However, some organizations encounter issues when deploying analytical solutions that use analytical patterns. These solutions are commonly still implemented as monolithic solutions, where a single team is the platform provider and the team is doing data integration. Smaller organizations and organizations that have a high degree of centralization from a team setup perspective can use a single team. However, a larger organization using only a single team often creates a bottleneck. This bottleneck causes a huge backlog, which results in parts of an organization waiting for data integration services and analytical solutions.

This pattern becomes more common as organizations adopt modern data science solutions. Many modern data science solutions require more data than traditional business intelligence solutions did in the past.

The recent switch to using microservices as an application development pattern is another driver of long backlogs around data integration, because it increases the

number of data sources.

A single team handling all data ingestion on a single platform in a large organization can also be problematic. One team rarely has experts for every data source. Most organizations are decentralized and distributed from a business perspective. Different business units and departments handle different parts of the business operation, so data experts are typically spread out across various sectors.

A pattern called data mesh was introduced to solve these problems. Data mesh's goal is to let distributed teams work with and share information in a decentralized and agile manner.

Data mesh is a technical pattern that also requires organizational change. The benefits of a data mesh approach are achieved by implementing multi-disciplinary teams that publish and consume data products.

The following concepts are foundational for understanding data mesh architecture:

- Data domains
- Data products
- Self-serve platforms
- Federated governance

## Data domains

Data domains are the foundation of data mesh. The concept of data domains comes from Domain Driven Development (DDD), a paradigm often used in software development to model complex software solutions. In data mesh, a data domain is a way to define boundaries around your enterprise data. Domains can vary depending on your organization, and in some cases, you can define domains around your organization. In other cases, you might choose to model data domains based on your business processes or source systems.

There are three aspects to data domains:

- Your chosen boundaries render themselves to long term ownership. They exist over a long period of time and have identified owners.
- Your domains should match reality, not just theoretical concepts.
- Your domains need to have atomic integrity. If areas have no relationship with each other, don't combine them in a domain together.

For more information about data domains and how you should define them, see [Data domains](#).

## Data products

Data products are another important component of data mesh. Data products aim to take product thinking to the world of data. In order for your data product to be successful, it needs to provide a long term business value to the intended users. In data mesh, a data product involves data, code assets, metadata, and related policies. Data products can be delivered as an API, report, table, or dataset in a data lake.

A successful data product must be:

- **Usable:** Your product must have users outside of the immediate data domain.
- **Valuable:** Your product must maintain value over time. If it doesn't have long-term value, it can't succeed.
- **Feasible:** Your product must be feasible. If you can't actually build it, the product can't be a success. Your product must be feasible from both a data availability and a technical standpoint.

The code assets of a data product include code that generates it and code that delivers it. The code assets also include pipelines used to create the product and the product's final report.

For more information about data products, see [Cloud-scale analytics data products in Azure](#).

For specific guidance on using data mesh, see [What is a data product?](#).

## Self-serve platforms

A core of data mesh is having a platform that allows the data domains to build their data products on their own. Data domains need to define data products by using the tools and processes that are relevant for users without having a strong dependency on a central platform or a central platform team. In a data mesh, you have autonomous teams developing and managing autonomous products.

While using decentralization and alignment with business users that understand your data, remember the generalists who also work on your platform. Because you have generalists, you can't have specialized tools that require specialist knowledge to operate as the core foundation of your mesh-based platform.

You can successfully implement your self-serve platform by adopting the practices outlined in [Design considerations for self-serve data platforms](#).

## Federated governance

When you adopt a self-serve distributed data platform, you must place an increased emphasis on governance. Lack of governance leads to silos and data duplication across your data domains. Federate your governance, as people who understand the governance need exist within your domain aligned teams and among data owners.

To create your federated governance, implement automated policies around both platform and data needs. Use a high degree of automation for testing and monitoring. Adopt a code-first implementation strategy to handle standards, policies, data products, and platform deployment as code.

For more information on implementing federated governance aspects, see [Data governance overview](#).

## Summary

Data mesh can be an effective way to implement enterprise data platforms, but it isn't the best solution for all organizations. Data mesh requires autonomous teams that can work independently. Data mesh works best in large and complex organizations that have independent business units and need to scale their analytics adoption beyond a single platform and implementation team.

When using data mesh, take special care when implementing your governance so you don't create silos. Always keep product thinking for data at the core of your implementation to ensure success.

## Next steps

[Data domains](#)

## Feedback

Was this page helpful?

 Yes

 No

# Data domains

Article • 11/27/2024

Data mesh is fundamentally based on decentralization and the distribution of responsibility to domains. If you truly understand a part of the business, you're best positioned to manage the associated data and ensure its accuracy. This is the principle of domain-oriented data ownership.

To promote domain-oriented data ownership, you need to first decompose your data architecture. The data mesh founder, Zhamak Dehghani, promotes the Domain-Driven Design (DDD) approach to software development as a useful method to help you identify your data domains.

The difficulty with using DDD for data management is that DDD's original use case was modeling complex systems in a software development context. It wasn't originally created to model enterprise data, and for data management practitioners, its method can be abstract and technical. There's often a lack of understanding of DDD.

Practitioners find its conceptual notions too difficult to grasp or try to project examples from software architecture or object-oriented programming onto their data landscape. This article provides you with pragmatic guidance and clear vocabulary so you can understand and use DDD.

## Domain-driven design

Introduced by Eric Evans, Domain-Driven Design is a method for supporting software development that helps describe complex systems for larger organizations. DDD is popular because many of its high-level practices impact modern software and application development approaches, such as microservices.

DDD differentiates between bounded contexts, domains, and subdomains. Domains are problem spaces you want to address. They're areas where knowledge, behavior, laws, and activities come together. You see semantic coupling in domains, behavioral dependencies between components or services. Another aspect of domains is communication. Team members must use a language the whole team shares so everyone can work efficiently. This shared language is called the *ubiquitous language* or *domain language*.

Domains are decomposed into subdomains to better manage complexity. A common example of this is decomposing a domain into subdomains that each correspond to one specific business problem, as shown in [Operationalize data mesh for AI/ML](#).

Not all subdomains are the same. You can, for example, classify domains as core, generic, or supporting. Core subdomains are the most important. They're the secret sauce, the ingredients, that make an organization unique. Generic subdomains are nonspecific and typically easy to solve with off-the-shelf products. Supporting subdomains don't offer a competitive advantage but are necessary to keep an organization running and aren't complex.

Bounded contexts are logical (context) boundaries. They focus on the solution space: the design of systems and applications. It's an area where the alignment of focus on the solution space makes sense. Within DDD, this can include code, database designs, and so on. Between the domains and bounded contexts, there can be alignment, but there's no hard rule binding the two together. Bounded contexts are technical by nature and can span multiple domains and subdomains.

## Domain modeling recommendations

If you adopt data mesh as a concept for data democratization and implement the principle of domain-oriented data ownership to increase flexibility, how does this work in practice? What might a transition from enterprise data modeling to domain-driven design modeling look like? What lessons can you take from DDD for data management?

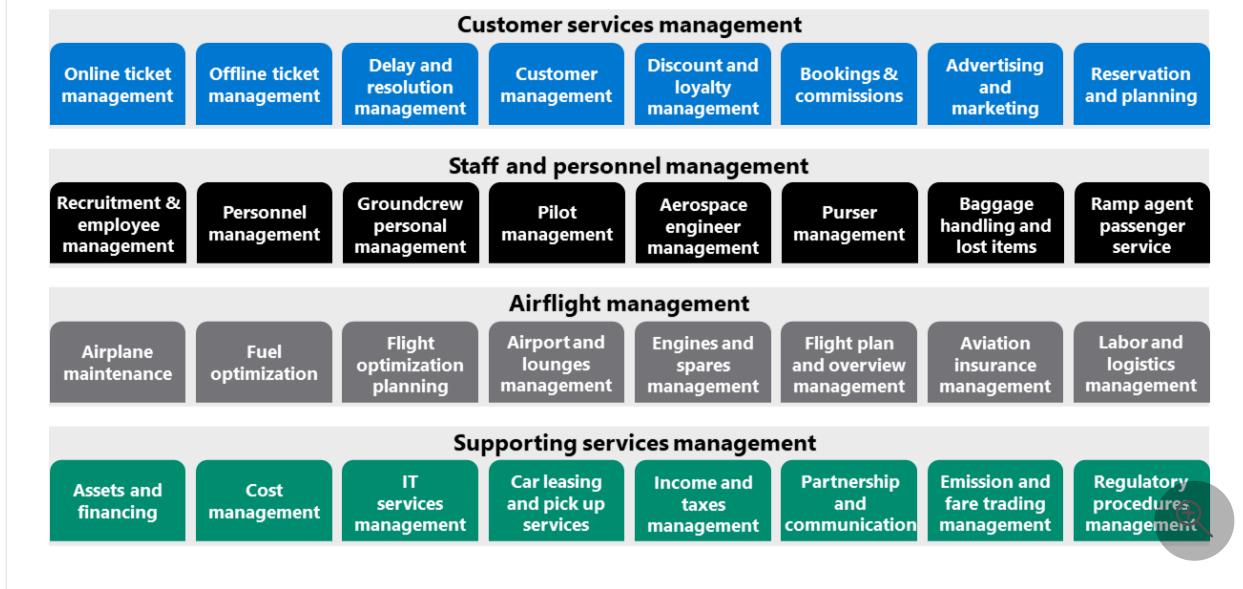
### Make a functional business decomposition of your problem spaces

Before allowing your teams to operate their data end-to-end, look at the scope and understand the problem spaces you're trying to address. It's important to do this exercise first before you jump into the details of a technical implementation. When you set logical boundaries between these problem spaces, responsibilities become clearer and can be better managed.

Look at your business architecture when grouping your problem spaces. Within business architecture, there are business capabilities: abilities or capacities that a business possesses or exchanges to achieve a specific purpose or outcome. This abstraction packs data, processes, organization, and technology into a particular context, in alignment with the strategic business goals and objectives of your organization. A business capability map shows which functional areas appear to be necessary to fulfill your mission and vision.

You can view the capability decomposition of a fictional company, Tailwind Traders, in the following model.

## Example functional domain decomposition of an Airline company



Tailwind Traders needs to master all functional areas listed in the Business Capability Map to be successful. Tailwind Traders must be able to sell tickets as part of Online or Offline Tickets Management Systems, for example, or have Pilots available to fly planes as part of a Pilot Management Program. The company can outsource some activities while keeping others as the core of its business.

What you observe in practice is that most of your people are organized around business capabilities. People working on the same business capabilities share the same vocabulary. The same is true of your applications and processes, which are typically well-aligned and tightly connected based on the cohesion of activities they support.

Business capability mapping is a great starting point, but your story doesn't end here.

## Map business capabilities to applications and data

To better manage your enterprise architecture, align your business capabilities, bounded contexts, and applications. It's important to follow some ground rules as you do so.

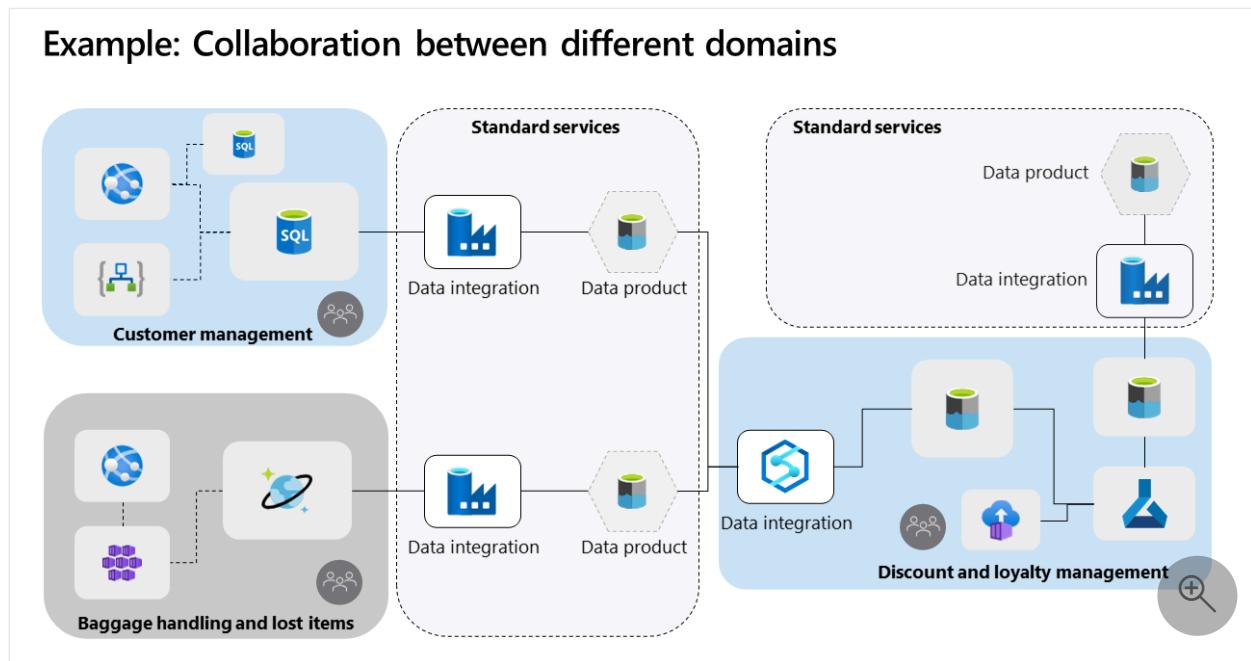
Business capabilities must stay on the business level and remain abstract. They represent what your organization does and target your problem spaces. When you implement a business capability, a realization (capability instance) for a specific context is created. Multiple applications and components work together within such boundaries in your solution space to deliver specific business value.

Applications and components aligned with a particular business capability stay decoupled from applications that are aligned with other business capabilities because they address different business concerns. Bounded contexts are derived from and

exclusively mapped to business capabilities. They represent the boundary of a business capability implementation, and they behave like a domain.

If business capabilities change, bounded contexts change. You preferably expect full alignment between domains and corresponding bounded contexts, but as you learn in later sections, reality sometimes differs from the ideal.

If we project capability mapping to Tailwind Traders, bounded context boundaries and domain implementations might look similar to the following diagram.



In this diagram, Customer Management is built upon subject matter expertise and therefore knows best what data to serve to other domains. Customer Management's inner architecture is decoupled, so all application components within these boundaries can directly communicate using application-specific interfaces and data models.

[Data products](#) and clear interoperability standards are used to formalize data distribution to other domains. In this approach, all data products also align with the domain and inherit the ubiquitous language, which is a constructed, formalized language agreed upon by stakeholders and designers from the same domain, to serve the needs of that domain.

## Extra domains from multiple capability realizations

It's important to acknowledge when working with business capability maps that some business capabilities can be instantiated multiple times.

As an example, Tailwind Traders might have multiple localized realizations (or implementations) of "baggage handling and lost items." Assume one line of their business operates only in Asia. In this context, "baggage handling and lost items" is a

capability that is fulfilled for Asia-related airplanes. A different line of business might target the European market, and in this context, another "baggage handling and lost items" capability is used. This scenario of multiple instances can lead to multiple localized implementations using different technology services and disjointed teams to operate those services.

The relationship of business capability and capability instances (realizations) is one-to-many. Because of this, you end up with extra (sub-) domains.

## Find shared capabilities and watch out for shared data

How you handle shared business capabilities is important. You typically implement shared capabilities centrally, as service models, and provide them to different lines of business. "Customer Management" can be an example of this kind of capability. In our Tailwind Traders example, both the Asian and European lines of business use the same administration for their clients.

But how can you project domain data ownership on a shared capability? Multiple business representatives likely take accountability for customers within the same shared administration.

There's an application domain and a data domain. Your domain and bounded context don't perfectly align from a data product viewpoint. You could conversely argue that there's still a single data concern from a business capability viewpoint.

For shared capabilities like complex vendor packages, SaaS solutions, and legacy systems, be consistent in your domain data ownership approach. You can segregate data ownership via data products, which might require application improvements. In our Tailwind Traders "Customer Management" example, different pipelines from the application domain might generate multiple data products: one data product for all Asia-related customers, and one for all Europe-related customers. In this situation, multiple data domains originate from the same application domain.

You can also ask your application domains to create a single data product that encapsulates metadata for distinguishing data ownership within itself. You could, for example, reserve a column name for ownership, mapping each row to a single specific data domain.

## Identify monoliths offering multiple business capabilities

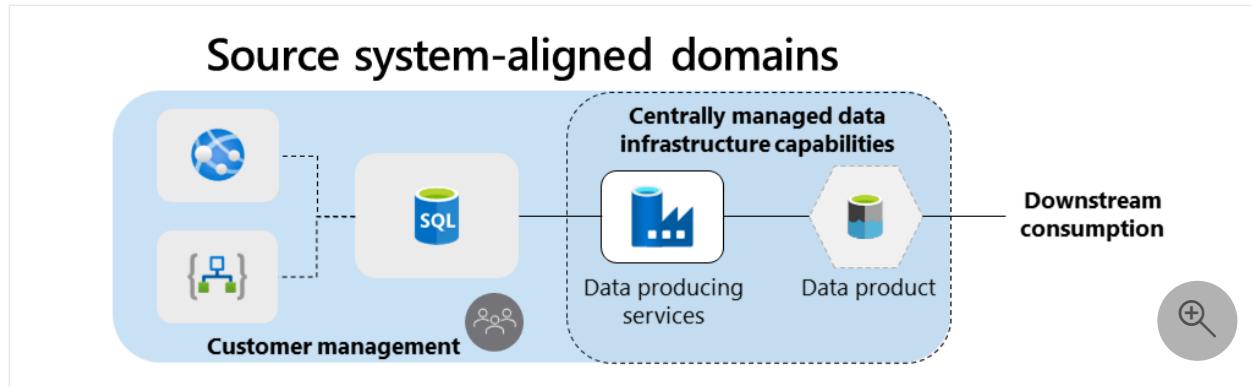
Pay attention also to applications that cater to multiple business capabilities, which are often seen in large and traditional enterprises. In our example scenario, Tailwind Traders

use a complex software package to facilitate both "cost management" and "assets and financing." These shared applications are monoliths that provide as many features as possible, making them large and complex. In such a situation, the application domain should be larger. The same thing applies to shared ownership, in which several data domains reside in an application domain.

## Design patterns for source-aligned, redelivery, and consumer-aligned domains

When you map your domains, you can choose a pattern based on the creation, consumption, or redelivery of your data. For your architecture, you can design templates that support your domains based on the domain's specific characteristics.

### Source system-aligned domains



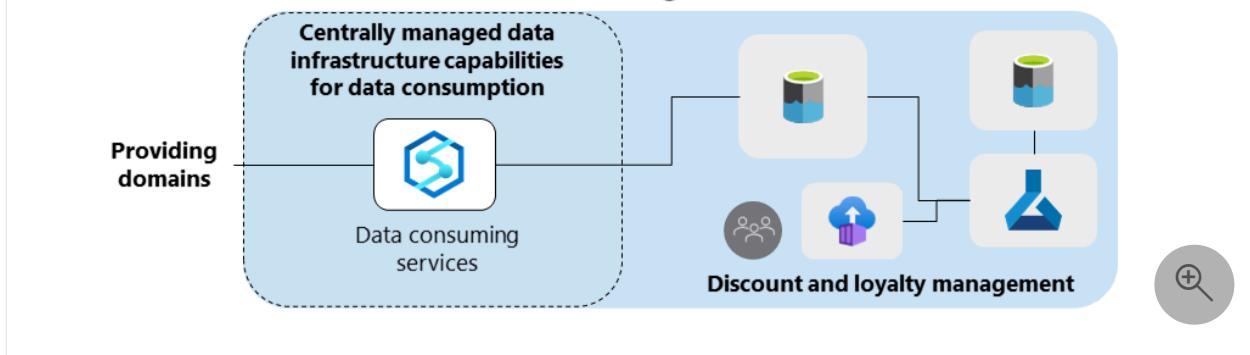
Source system-aligned domains are aligned with source systems where data originates. These systems are typically transactional or operational.

Your goal is to capture data directly from these golden source systems. Read-optimized data products from your providing domains for intensive data consumption. Facilitate these domains using standardized services for data transformation and sharing.

These services, which include preconfigured container structures, enable your source-oriented domain teams to publish data more easily. It's the path of least resistance with minimal disruption and cost.

### Consumer-aligned domains

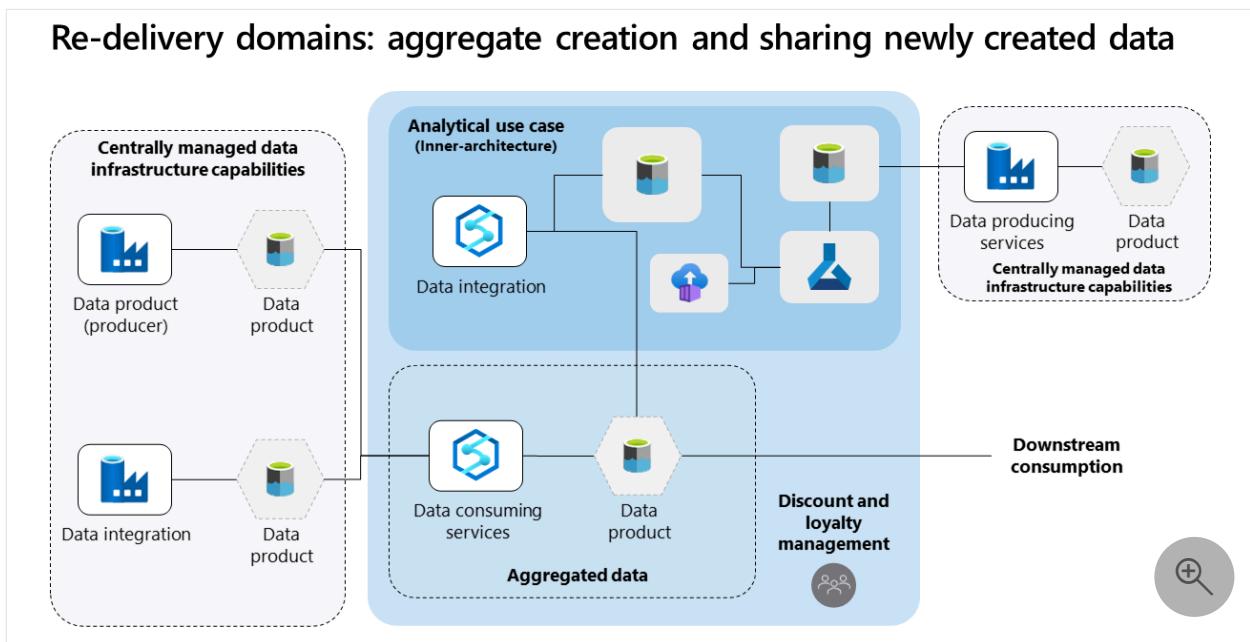
## Consumer-aligned domains



Consumer-aligned domains are the opposite of source-aligned domains. They're aligned with specific end-user use cases that require data from other domains. Customer-aligned domains consume and transform data to fit the use cases of your organization.

Consider offering shared data services for data transformation and consumption to cater to these consuming needs. You could offer domain-agnostic data infrastructure capabilities, for example, to handle data pipelines, storage infrastructure, streaming services, analytical processing, and so on.

## Redelivery domains



The reusability of data is a different and more difficult scenario. For example, if downstream consumers are interested in a combination of data from different domains, you can create data products that aggregate data or combine high-level data required by many domains. This allows you to avoid repetitive work.

Don't create strong dependencies between your data products and analytical use cases. Strive for flexibility and loose coupling instead. The following model demonstrates how you can achieve flexibility. A domain takes ownership of both data products and

analytical use cases and has designed separate processes for data product creation and data usage.

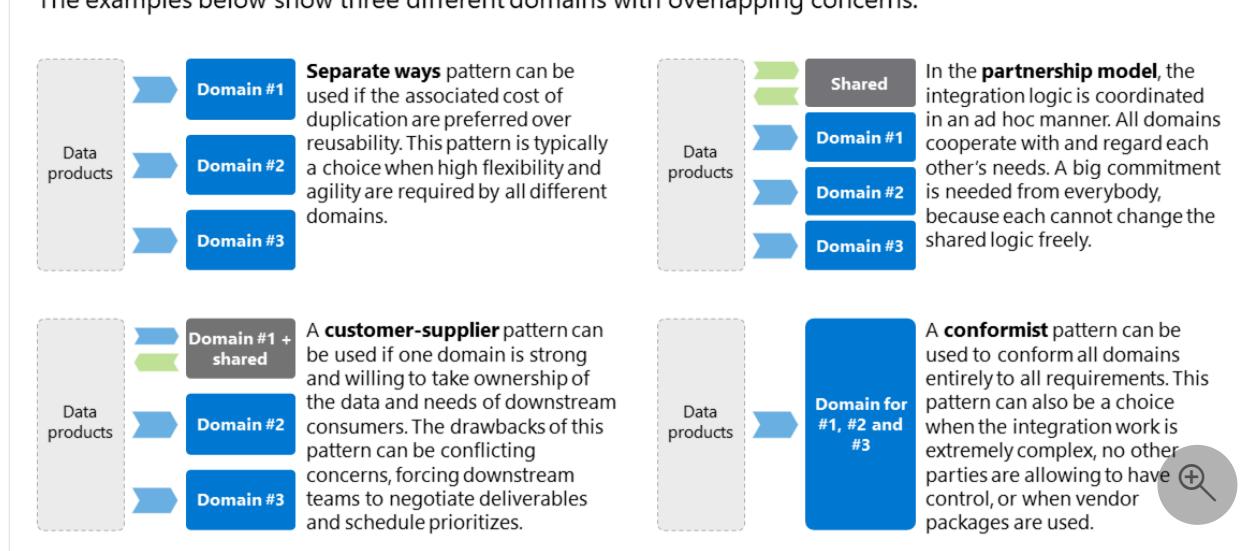
## Define overlapping domain patterns

Domain modeling often gets complicated when data or business logic is shared across domains. In large-scale organizations, domains often rely on data from other domains. It can be helpful to have generic domains that provide integration logic in a way that allows other subdomains to standardize and benefit from it. Keep your shared model between subdomains small and always aligned with the ubiquitous language.

For overlapping data requirements, you can use different patterns from domain-driven design. Here's a short summary of the patterns you could choose from:

### Best practices for overlapping contexts

Different integration patterns can be used, when multiple domain contexts and relationships exist. The examples below show three different domains with overlapping concerns.



- A **separate ways** pattern can be used if you prefer the associated cost of duplication over reusability. Reusability is sacrificed for higher flexibility and agility.
- A **customer-supplier** pattern can be used if one domain is strong and willing to take ownership of downstream consumers' data and needs. Drawbacks include conflicting concerns and forcing your downstream teams to negotiate deliverables and schedule priorities.
- A **partnership** pattern can be used when your integration logic is coordinated in an unplanned manner within a newly created domain. All teams cooperate with and regard each other's needs. Because no one can freely change the shared logic, significant commitment is needed from everyone involved.
- A **conformist** pattern can be used to conform all your domains to all requirements. Use this pattern when your integration work is complex, no other parties can have control, or you use vendor packages.

In all cases, your domains should adhere to your interoperability standards. A partnership domain that produces new data for other domains must expose their data products like any other domain, including taking ownership.

## Domain responsibilities

Data mesh decentralizes data ownership by distributing it among domain teams. For many organizations, this means a shift from a centralized model around governance to a federated model. Domain teams are assigned tasks, such as:

- Taking ownership of data pipelines, such as ingestion, cleaning, and transforming data, to serve as many data customers' needs as possible
- Improving [Data Quality](#), including adherence to SLAs and quality measures set by data consumers
- Encapsulating metadata or using reserved column names for fine-grain row- and column-level filtering
- Adhering to metadata management standards, including:
  - Application and source system schema registration
  - Metadata for improved discoverability
  - Versioning information
  - Linkage of data attributes and business terms
  - Integrity of [metadata](#) information to allow better integration between domains
- Adhering to data interoperability standards, including protocols, data formats, and data types
- Providing lineage either by linking source systems and integration services to scanners or by manually providing lineage
- Adhering to data sharing tasks, including IAM access reviews and data contract creation

## Level of granularity for decoupling

Now that you know how to recognize and facilitate data domains, you can learn to design the right level of domain granularity and rules for decoupling. Two important dimensions are in play when you decompose your architecture.

Granularity for functional domains and setting bounded contexts is one dimension. Domains conform to a particular way of working, ensuring data becomes available to all domains using shared services, taking ownership, adhering to metadata standards, and so on.

Set fine-grained boundaries where possible for data distribution. Becoming data-driven is all about making data available for intensive reuse. If you make your boundaries too loose, you force undesired couplings between many applications and lose data reusability. Strive for decoupling each time data crosses boundaries of business capabilities. Within a domain, tight coupling within the inner architecture of the domain is allowed. However, when crossing the boundaries of business capabilities, domains must stay decoupled and distribute read-optimized data products for sharing with other domains.

Granularity for technical domains and infrastructure utilization is the other important dimension. Your [data landing zones](#) enable agility for servicing [data applications](#), which create data products. How do you create this kind of landing zone with shared infrastructure and services underneath? Functional domains are logically grouped together and are good candidates for sharing platform infrastructure. Here are some factors to consider when creating these landing zones:

- Cohesion and efficiency when working with and sharing data is a strong driver of aligning functional domains to a data landing zone. This relates to data gravity, the tendency to constantly share large data products between domains.
- Regional boundaries can result in extra data landing zones being implemented.
- Ownership, security, or legal boundaries can force domain segregation. For example, some data can't be visible to any other domains.
- Flexibility and the pace of change are important drivers. Some domains can have a high velocity of innovation, while other domains strongly value stability.
- Functional boundaries can drive teams apart. An example of this could be source-oriented and consumer-oriented boundaries. Half of your domain teams might value certain services over others.
- If you want to potentially sell or separate off your capability, you should avoid integrating tightly with shared services from other domains.
- Team size, skills, and maturity can all be important factors. Highly skilled and mature teams often prefer to operate their own services and infrastructure, while less mature teams are less likely to value the extra overhead of platform maintenance.

Before you provision many data landing zones, look at your domain decomposition and determine what functional domains are candidates for sharing underlying infrastructure.

## Summary

Business capability modeling helps you to better recognize and organize your domains in a data mesh architecture. It provides a holistic view of the way data and applications

deliver value to your business while at the same time helping you prioritize and focus on your data strategy and business needs. You can also use business capability modeling for more than just data. For example, if scalability is a concern, you can use this model to identify your most critical core capabilities and develop a strategy for them.

Some practitioners raise the concern that building target-state architecture by mapping everything out upfront is an intensive exercise. They instead suggest you identify your domains organically while you onboard them into your new data mesh architecture. Instead of defining your target state from the top down, you work bottom-up, exploring, experimenting, and transitioning your current state towards a target state. While this proposed approach might be quicker, it carries significant risk. You can easily be in the middle of a complex move or remodeling operation when things start to break. Working from both directions, top-down and bottom-up, and then meeting in the middle over time is a more nuanced approach.

## Next Step

- [What is a data product?](#)
- 

## Feedback

Was this page helpful?



# What is a data product?

Article • 12/10/2024

Every application creates and stores data either temporarily or permanently. Many applications also create and save data for operational management purposes, such as error logging and health monitoring. To consume and process the data that these applications produce, centralized data teams use extract, transform, and load (ETL) processes. Application operation teams often have other data processing flows for data like application health data and KPI status monitoring data.

For data integration, a traditional waterfall approach, in which teams follow a specific order of phases, isn't ideal. It can lead to knowledge gaps, ownership problems, and communication conflicts that affect your data's quality, timeliness, and value for users. Application teams are responsible for application performance and success. When they use a waterfall approach, they make changes to downstream processes that other teams own. Sometimes these changes can affect other areas. For example, a minor upstream change might drastically alter a KPI's trend. These conflicts can affect your ability to make critical decisions.

## Data as a product

To prevent these problems, the [data mesh](#) approach adopts the concept of *data as a product*. Application owners and application teams treat data as a fully contained product that they're responsible for, rather than a byproduct of another team's process. Both applications and analytical data-serving tasks are within domain responsibility areas.

Data products are created specifically for analytical consumption. They have defined and agreed-upon shapes, consumption interfaces, and maintenance and refresh cycles, all of which are documented.

Data products are processed domain data assets or datasets that you can share with downstream processes through interfaces in a service-level objective. Unless otherwise required, you should process, shape, cleanse, aggregate, and normalize your raw data to meet agreed-upon quality standards before you make it available for use.

The following sections outline common characteristics of good data products.

## Data product characteristics

Ensure that your data products are:

- **Discoverable, understandable, and trustworthy.** To provide discoverability and clarity, share and update information about each data product, its data, its meaning, the shape format of its data, and its refresh cycle. Communicate data changes or shape changes to downstream consumers in a timely manner. To ensure trustworthiness, interfaces provide time-bounded backwards compatibility for data product shapes.
- **Addressable, natively accessible, and secure.** To provide addressability, create defined processes to locate and gain access to each data product. Implement security measures for various access requirements. Shift your data domain ownership mentality from gatekeeping data to serving data with well-defined security precautions. Well-documented access interfaces can vary across different technologies. Commonly used interfaces for natively accessible data products include APIs, database users, tables, or views, and files with necessary access rights.
- **Interoperable, truthful, and valuable.** To provide interoperability, ensure that your data follows defined common standards, such as values that have the same name and data type. For example, you might name a column that contains customer identification data *CustomerID* in every data product, and its data might always be an integer. Data products provide value to customers, and you can use them as upstream sources for new data products in the same domain or different domains. But you can't just carry and copy the same data product in multiple places. Each data product that comes from a previous data product should provide new value and information to downstream consumers. Data products must also provide truthful, accurate data.

Use well-designed, well-maintained data products and their interfaces to help avoid duplicating data and create a native single source of truth.

## Data product design recommendations

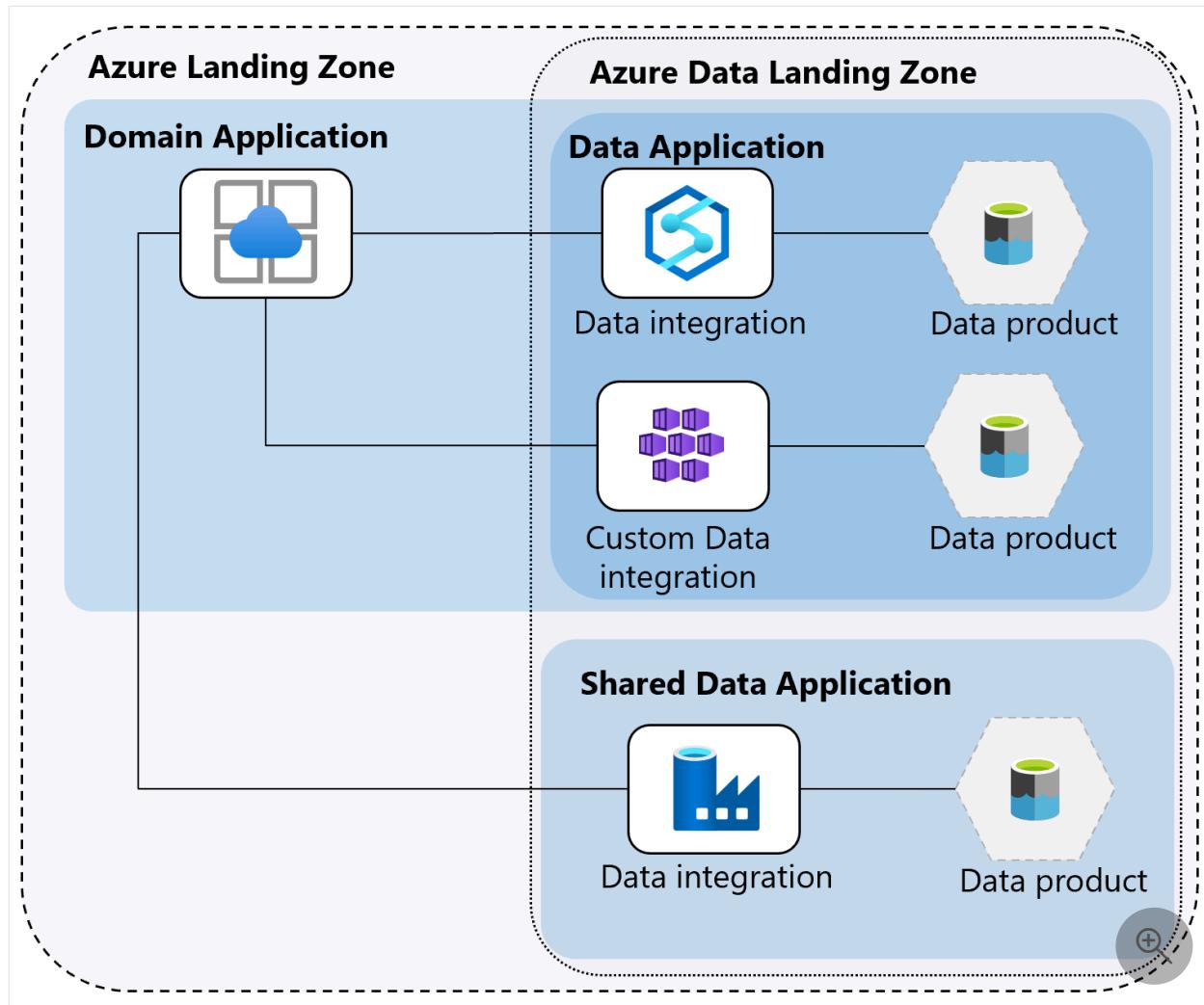
To fulfill data product serving requirements, your domain teams must acquire a new set of skills and use new tools and platforms.

To build the data applications and produce or serve data products, fully equip your domain application teams. Your teams can use a familiar technology stack to build data products. They might also prefer to have their own Spark instance or pipeline engine. For example, a large domain that serves many data products might process and serve data products from their own Azure Synapse Analytics instance. Smaller organizations and smaller domains of large organizations might develop and run their data

applications on a shared platform, such as a centrally located Azure Data Factory, Azure Synapse Analytics, or Azure Databricks instance.

Ensure that your data products have the common characteristics that are described in this article, that your lineage repository reflects your data application lineage, and that you govern your implementation and access.

The following diagram shows an example data application logical layout in a domain and landing zone.



## Next step

- Design considerations for self-serve data platforms

## Feedback

Was this page helpful?

Yes

No

# Data contracts

Article • 11/27/2024

Responsibilities are distributed between [domains](#) in a federated architecture, which can make it difficult to oversee dependencies and gain data usage insights. Data contracts can help you gain data usage insights because they provide information about who owns each [data product](#). Data contracts help you set standards and confidently manage your data pipelines. They are essential for robust data management, providing information on:

- Which data products are being consumed.
- Which users are consuming which data products.
- What purpose is leading users to consume specific data products.

Data product distribution and usage have two dimensions: technical concerns and business concerns. Technical concerns include data pipeline handling and mutual data stability expectations. Business concerns include data sharing purpose agreements, which define usage, privacy, and purpose objectives, including any limitations.

The two dimensions involve different roles. Generally, you should rely on application owners or data engineers for technical concerns and product owners or business representatives for business concerns.

## Data contracts principles

Data contracts are similar to service contracts or data delivery contracts.

In a larger or distributed architecture, it can be difficult to oversee changes. You can simplify your oversight by implementing versioning and managing compatibility whenever you have a data product that is popular and widely used.

If applications are coupled, it indicates a high degree of interdependence between the coupled applications. Applications that access or consume data from other applications always suffer when coupled. Any change to the data structure, for example, is likely to directly affect other applications that access or consume that data. In situations where you have many applications coupled together, it is common to encounter a cascading effect where a small change to a single application affects many other applications. Due to the increased likelihood of unintended effects after even minor changes, many architects and software engineers avoid building coupled architectures.

A data contract guarantees interface compatibility and includes terms of service and a service level agreement (SLA). Terms of service outline how data can be used, such as restricting its use to only development, testing, or production. SLAs describe the required quality of data delivery and interface. Quality details you might specify in an SLA include:

- Uptime
- Error rates
- Availability
- Deprecation
- A roadmap
- Version numbers

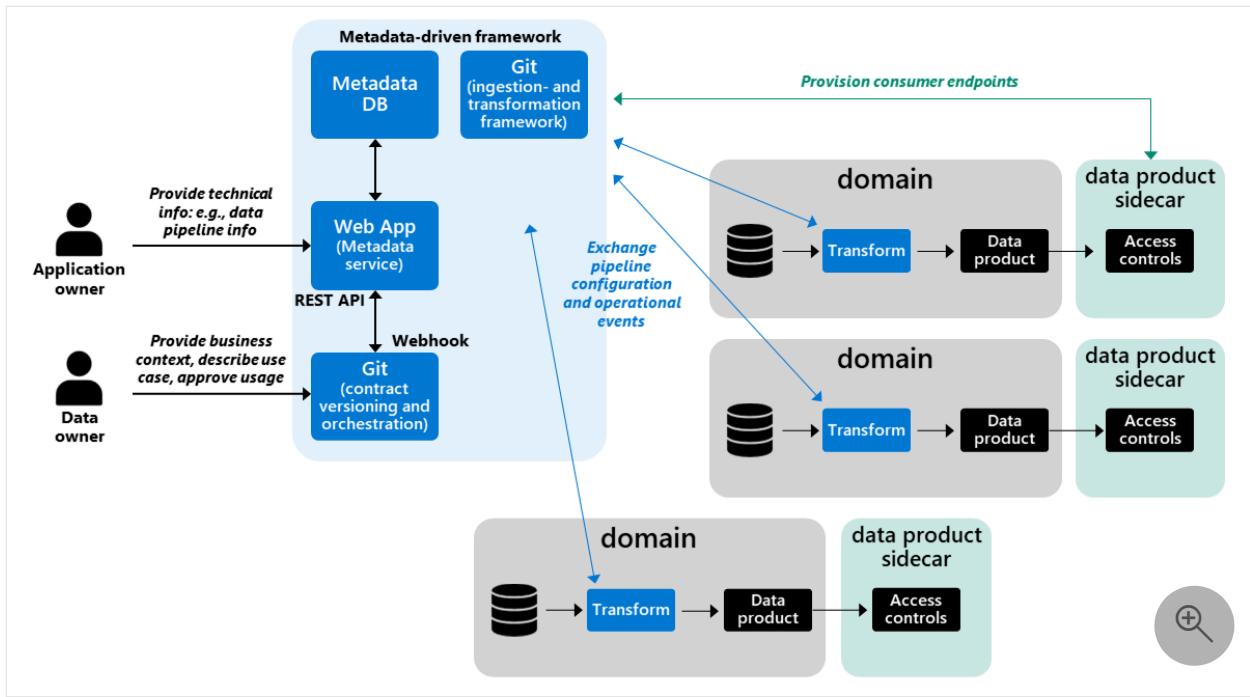
You can place the metadata that captures these details under source control, which allows for automatic triggering of validations and deployments. For more information on source control, see [Source control in Azure Data Factory](#).

Data contracts provide insight into coupling and dependencies between domains and applications. A contract also allows for [contract testing](#), which ensures that all application and interface changes are validated against your consumers' data requirements. You can tell when your data flows become vulnerable to upstream data source changes by detecting schema drift. For more information, see [Schema drift in mapping data flow](#).

Data contracts are often part of metadata-driven ingestion frameworks. You can store data contracts in metadata records within a centrally managed [metastore](#). From that central location, your data contracts play an important role in multiple areas of data ingestion, including:

- Pipeline execution
- Data product creation
- Data type validation
- Schemas
- Interoperability standards
- Protocol versions
- Defaulting rules on missing data

Data contracts involve large quantities of technical metadata. To document your data pipelines and data products, you must have a clear description of your data sources, all transformations your data has undergone, and how you ultimately deliver the data.



In a distributed architecture, you distribute a data pipeline framework across different domains, and the domains conform to a common way of working. Since the domains process data themselves, control and responsibility stay with them, while the framework and metadata remain under central governance.

When implementing a federated method, start small. Begin with basics, like metadata storage for schema validation, enterprise identifiers, and references to other datasets in your shared metadata repository. Add data [lineage](#) support to help you visualize data movement. Bootstrap your processes and implement controls for technical data quality validation.

All your controls should be part of your continuous integration procedures. Capture all runtime information, including metrics and logging, and make that information part of your metadata foundation for gaining data pipeline stability insights. This setup ensures that you have a feedback loop between your domains and your central management cockpit.

As you stabilize all data movement, capture which data attributes (like tables and columns) are used by which data consumers and use this information to continue scaling. You can include this information in your centrally managed metastore. Data usage information allows you to detect breaking changes and identify their effects on your data producers and consumers. If a data product dataset has no consumers, you can allow it to experience disruptive changes. Use source control (like Git) to allow a handshake process between providers and consumers of your data.

## Data sharing agreements

Data sharing agreements are an extension of data contracts. The agreements outline data usage, privacy, and purpose, including any limitations. Data sharing agreements are interface-independent and offer insights into what data is used for a particular purpose. They also function as input for data security controls. You can use a data sharing agreement to outline which filters or security protections must be applied to your data.

Data sharing agreements also help prevent miscommunication over data usage. Domain owners should discuss data sharing and data usage issues before any data is shared. Having a common understanding is critical for your ability to regulate data and its usage and ensure you can deliver value to your organization. After all domain owners reach a collaborative understanding, ensure that they document it in a data sharing agreement. In this agreement, you can also address areas like:

- Functional data quality
- Historization
- Data lifecycle management
- Further distribution of data

Apply classifications and conditions like sensitivity labels or filtering conditions to secure your data.

The previous section's diagram shows certain elements labeled *data product sidecar*. A data product sidecar is a component or layer for injecting policy execution, like data access controls or data consumption output methods. It's a security abstraction that uses data contracts to handle security enforcement over your domain data. You can create a data product sidecar from your data contract repository as an access control list (ACL) or serverless view, or you can create one using a duplicated dataset that you select and filter for a specific consumer. Either way, the goal is to derive security views from your data contracts in a fully automated manner.

Connect data contract attributes and your documentation. Ensure that you provide semantic context and a relationship to your glossary so that your consumers can understand how business requirements translate to an actual implementation. If a relationship with business terms is important to your organization, consider implementing policies such as only allowing data contracts to be established after all data product attributes are linked to business term entities. You might also apply this type of policy to contextual changes like relationship or definition adjustments.

## Use data contracts

Start slow when beginning to use data contracts. Don't introduce too many changes at once; data contracts require a cultural shift, and your users need time to become

familiar with them and understand the importance of data ownership. You also need to find the sweet spot between too few and too many metadata attributes in your data contracts.

The following steps outline the process of implementing data contracts for your organization:

1. Ensure your technical data pipelines are stable. Use cases can't reach production if the pipelines they travel through experience unexpected disruptions.
2. Put simple and pragmatic processes in place as you start using sharing agreements. You can begin by designing a simple form or template in Microsoft Forms. Write in clear, concise language that readers can easily understand. The focus of this first phase is a cultural shift and collecting requirements. Make sure you don't overcomplicate things; accept manual processes, limit your initial metadata requirements, and iterate until those requirements are stable.
3. After you have your first processes firmly in place, begin replacing your manual forms with a web-based application, database, and/or message queue. Your central data governance team should still be responsible for oversight during this phase. Data access granularity at this point is typically coarse-grained, focusing on folders or files. Whenever possible, use REST APIs to automatically provision your data access policies or ACLs.
4. Put data owners or data stewards in charge of a strong workflow for approval management. Your central data governance role should now oversee approvals only from a backseat role and review all data contracts regularly. At this point, you should have a data catalog like [Microsoft Purview](#) up and running that shows all your ready-for-consumption data products. Improve your data and security enforcement capability by allowing for fine-grained selections and filtering, and consider using techniques like dynamic data masking to prevent your data from being duplicated.
5. In the final stage of your data contract implementation journey, everything should be self-service and fully automated. Automated machine learning should predict data approvals. Secure views, for example, are automatically deployed after approval.

Data contracts are a relatively new yet important addition to data mesh architecture, providing transparency for data usage and dependencies. Focus on technical stability and standardization as you first begin to use data contracts, then use a lessons-learned process as you iterate. Slowly build up and automate your data governance so you don't increase your organization's overhead.

As part of your data contract documentation, you also need terms of service and service-level agreements (SLAs). Use SLAs to outline quality requirements for your data

delivery and interfaces, including uptime, error rates, and availability. SLAs can also include any deprecation, roadmap, and version number requirements you need to define.

## Next steps

- [Design considerations for self-serve data platforms](#)
- 

## Feedback

Was this page helpful?



Yes



No

# Design considerations for self-serve data platforms

Article • 05/07/2024

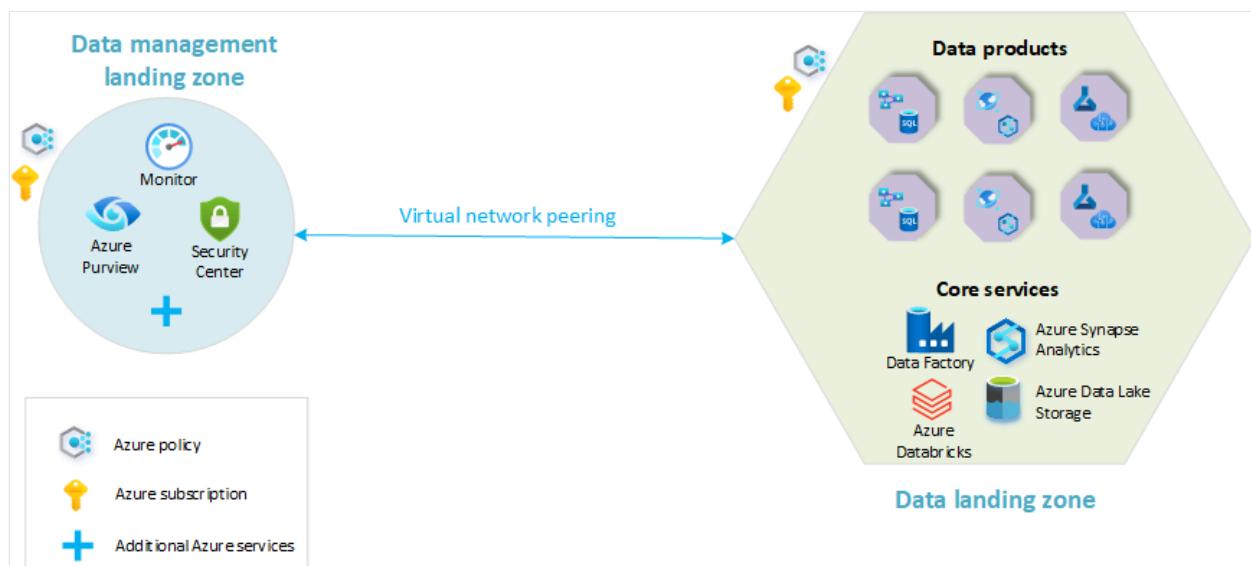
Data mesh is an exciting new approach to data architecture design and development. Unlike traditional data architecture, data mesh separates responsibility between functional [data domains](#) that focus on creating [data products](#) and a platform team that focuses on technical capabilities. This separation of responsibilities must be reflected in your platform. You must strike a balance between providing domain-agnostic capabilities and enabling your domain teams to model, process, and distribute their data across your organization.

Choosing the right level of domain granularity and rules for decoupling using platforms isn't easy. This article contains several scenarios that provide you with detailed guidance.

## Cloud-scale analytics

When you want to build a data mesh with Azure, we recommend you adopt [cloud-scale analytics](#). This framework is a deployable reference architecture and comes with open-source templates and best practices. Cloud-scale analytics architecture has two main building blocks that are fundamental for all deployment choices:

- **Data management landing zone:** The foundation of your data architecture. It contains all critical capabilities for data management, like data catalog, data lineage, API catalog, master data management, and so on.
- **Data landing zones:** Subscriptions that host your analytics and AI solutions. They include key capabilities for hosting an analytics platform.



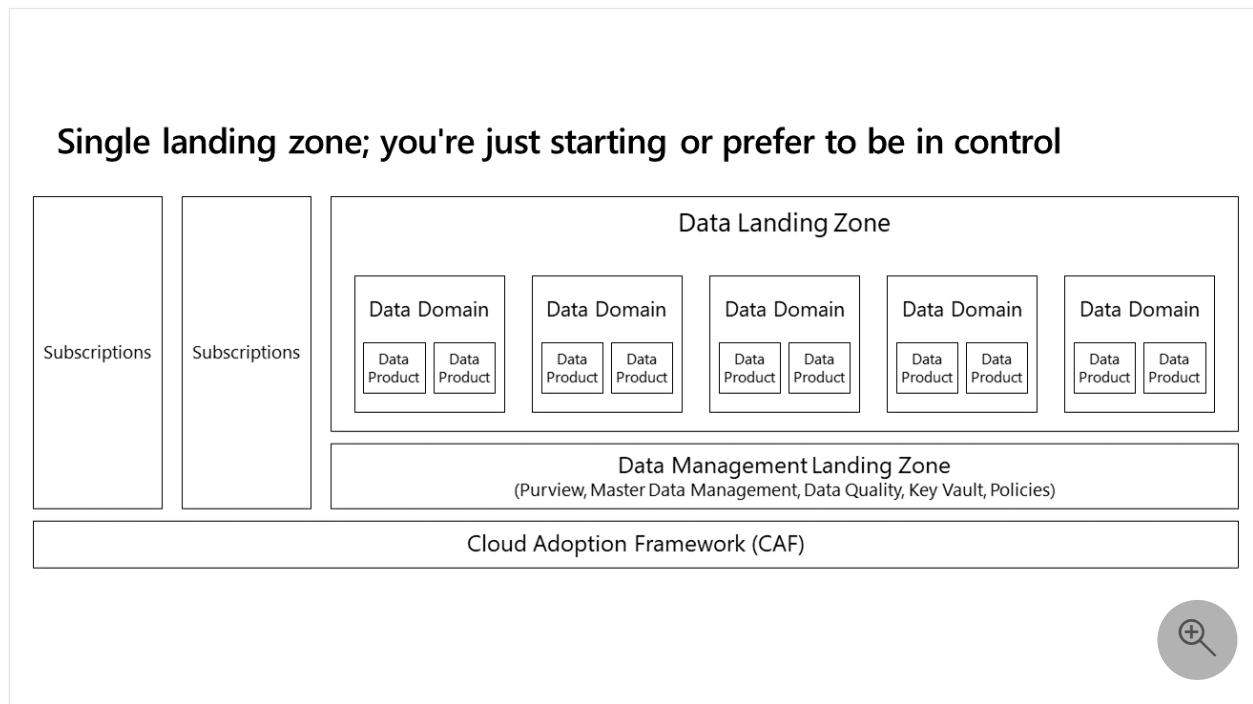
The following diagram provides an overview of a cloud-scale analytics platform with a data management landing zone and a single data landing zone. Not all Azure services are represented in the diagram. It has been simplified to highlight the core concepts resource organization within this architecture.

The cloud-based analytics framework isn't explicit on what exact type of data architecture you must provision. You can use it for many common cloud-scale analytics solutions, including (enterprise) data warehouses, data lakes, data lake houses and data meshes. All example solutions in this article use data mesh architecture.

Understand that all architectures adhere to the data mesh principles: domain ownership, data as product, self-serve data platform and federated computational governance. Different paths can all lead to a data mesh. There is no single right or wrong answer. You must make the right trade-offs for your organization's needs.

## Single data landing zone

The simplest deployment pattern for building a data mesh architecture involves one data management landing zone and one data landing zone. The data architecture in such a scenario would look like the following:



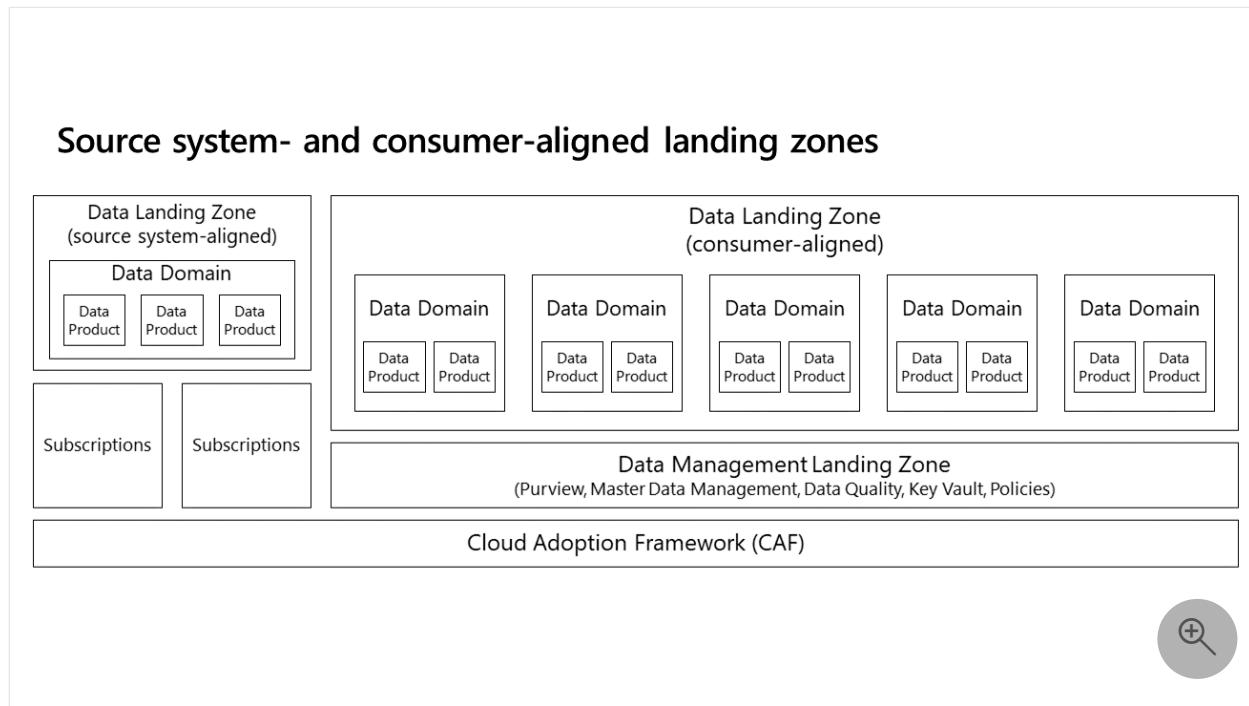
In this model, all your functional data domains reside same data landing zone. A single subscription contains a standard set of services. Resource groups segregate different data domains and data products. Standard data services, like Azure Data Lake Store, Azure Logic Apps and Azure Synapse Analytics, apply to all domains.

All data domains follow data mesh principles: data follows domain ownership, and data is treated like products. The platform is fully self-service, though there are limited variations of services. All domains should strongly adhere and conform to the same data management principles.

This deployment option can be useful for smaller companies or greenfield projects who want to embrace data mesh but not over-complicate things. This deployment can also be a starting point for an organization that plans to build something more complex. In this case, plan for expanding into multiple landing zones at a later time.

## Source system aligned and consumer aligned landing zones

In the previous model, we didn't take into account other subscriptions or on-premises applications. You can slightly alter the previous model by adding a source system-aligned landing zone to manage all incoming data. Data onboarding is a difficult process, so having two data landing zones is useful. Onboarding remains one of the most challenging parts of using data at large. Onboarding also often requires extra tools to address integration, because its challenges differ from those of integration. It helps to distinguish between providing data and consuming data.



In the architecture on the left of this diagram, services facilitate all data onboarding, like [CDC](#), services for pulling APIs, or data lake services for dynamically building datasets. Services in this platform can pull data from on-premises, cloud environments or SaaS vendors. This type of platform typically also has more overhead, because there's more

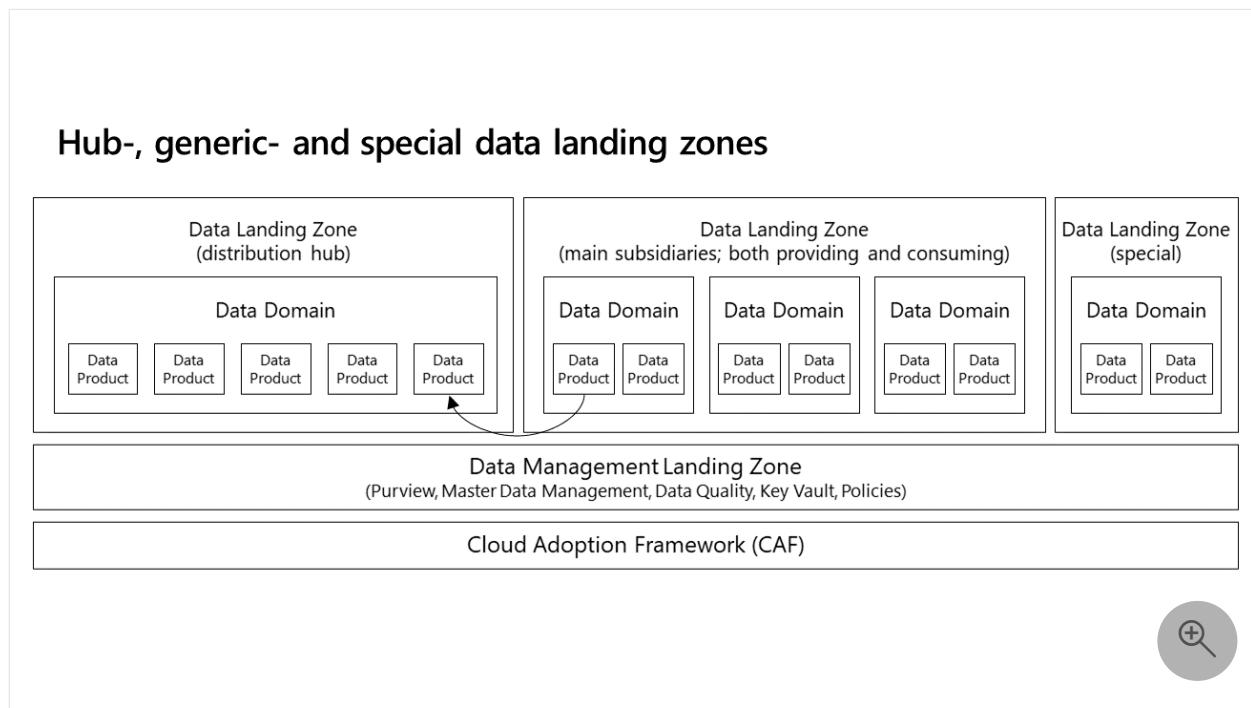
coupling with underlying operational applications. You might want to treat this differently from any data usage.

In the architecture on the right of the diagram, the organization optimizes for consumption and has services focused on turning data into value. These services can include machine learning, reporting, and so on.

These architecture domains follow all principles of data mesh. Domains take ownership of data and are allowed to directly distribute data to other domains.

## Hub, generic, and special data landing zones

The next deployment option is another iteration of the previous design. This deployment follows a governed mesh topology: data is distributed via a central hub, in which data is partitioned per domain, logically isolated, and not integrated. This model's hub uses its own (domain-agnostic) data landing zone, and can be owned by a central data governance team overseeing which data is distributed to which other domains. The hub also carries services that facilitate data onboarding.



For domains that require standard services for consuming, using, analyzing and creating new data, use generic data landing zone. This single subscription holds a standard set of services. Also apply data virtualization, as most of your data products are already persisted in the hub and you don't need more data duplication.

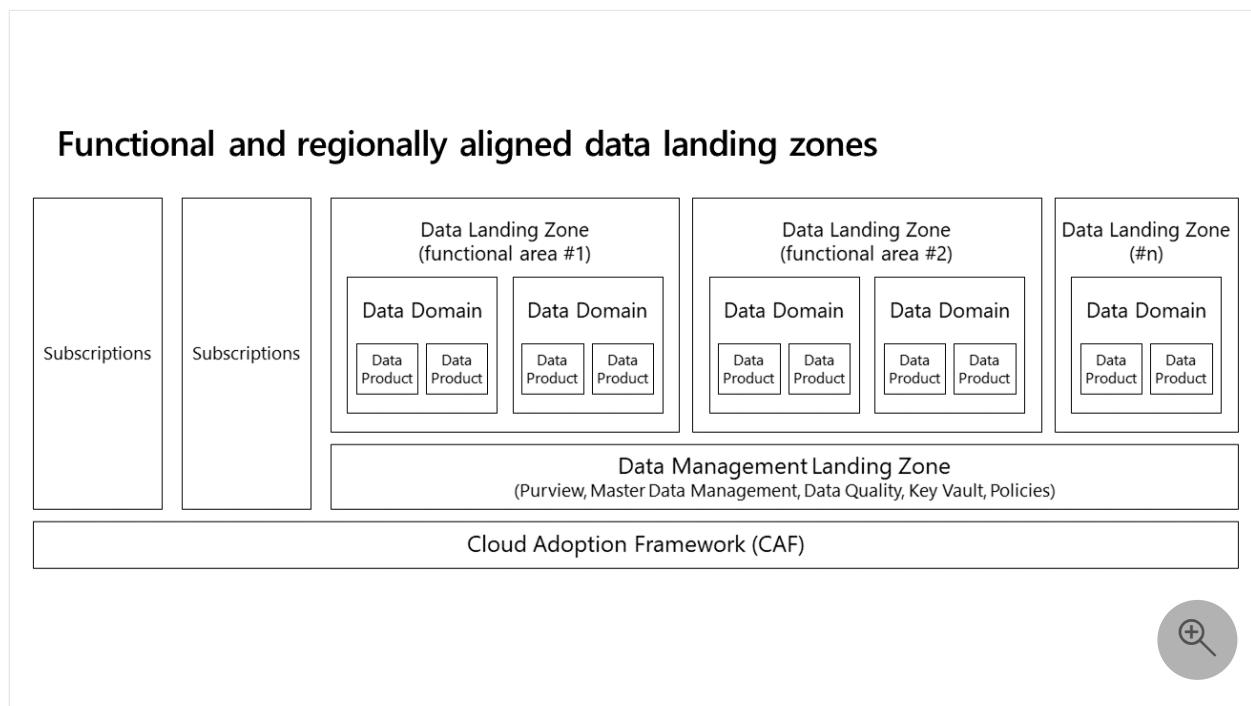
This deployment allows for 'specials': extra landing zones that you can provision when it's not possible to logically group domains. They could be needed when regional or legal boundaries apply, or when your domains have unique and contrasting

requirements. You might also need them in situations where a strong global subsidiary governance is applied with exceptions for overseas activities.

If your organization needs to control which data is distributed and consumed by which domains, hub deployment is a good option. It's also an option if you're addressing time-variant and non-volatile concerns for large data consumers. You can strongly standardize data product design, which allows your domains to time travel and perform redeliveries. This model is especially common within the financial industry.

## Functional and regionally aligned data landing zones

Provisioning multiple data landing zones can help you group functional domains based on cohesion and efficiency for working and sharing data. All your data landing zones adhere to the same auditing and controls, but you can still have flexibility and design changes between different data landing zones.



Determine the functional data domains that you want to logically group together for a shared data landing zone. For example, you might implement the same templates if you have regional boundaries. Ownership, security, or legal boundaries can force you to segregate domains. Flexibility, the pace of change, and separation or selling of your capabilities are also important factors to consider.

Further guidance and best practices can be found in [data domains](#).

Different landing zones don't stand alone. They can connect to data lakes hosted in other zones. This allows domains to collaborate across your enterprise. You can also

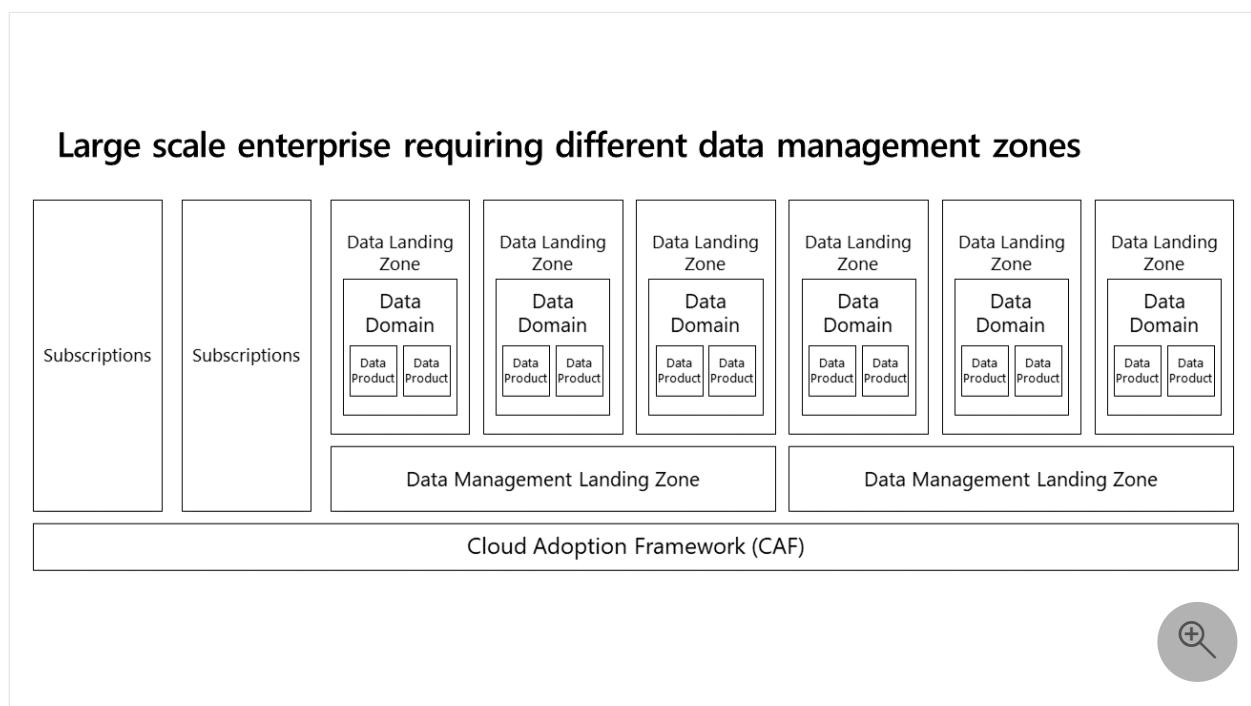
apply polyglot persistence to mix different data store technologies. Polyglot persistence allows your domains to directly read data from other domains without duplicating data.

When deploying multiple data landing zones, know that there's management overhead attached to each data landing zone. You must apply VNet peering between all data landing zones, you must manage extra private endpoints, and so on.

Deploying multiple data landing zones is good option if your data architecture is large. You can add more landing zones to your architecture to address common needs of various domains. These extra landing zones use virtual network peering to connect to both the data management landing zone and all other landing zones. Peering allows you to share datasets and resources across your landing zones. Splitting data across separate zones lets you spread workloads across your Azure subscriptions and resources. This approach helps organically implement the data mesh.

## Large scale enterprise requiring different data management zones

Large enterprises operating on a global scale can have contrasting data management requirements between different parts of their organization. You can deploy multiple data management and data landing zones together to address this issue. The following diagram shows an example of this type of architecture:



Multiple data management landing zones should justify your overhead and integration complexity. For example, another data management landing zone might make sense for

situations where your organization's (meta)data must not be seen by anyone outside your organization.

## Conclusion

The transition towards data mesh is a cultural shift involving nuances, trade offs and considerations. You can use cloud-scale analytics to obtain best practices and executable resources. This article's reference architectures offer starting points for you to kick-start your implementation.

## Next step

- [Data marketplace](#)
- 

## Feedback

Was this page helpful?

 Yes

 No

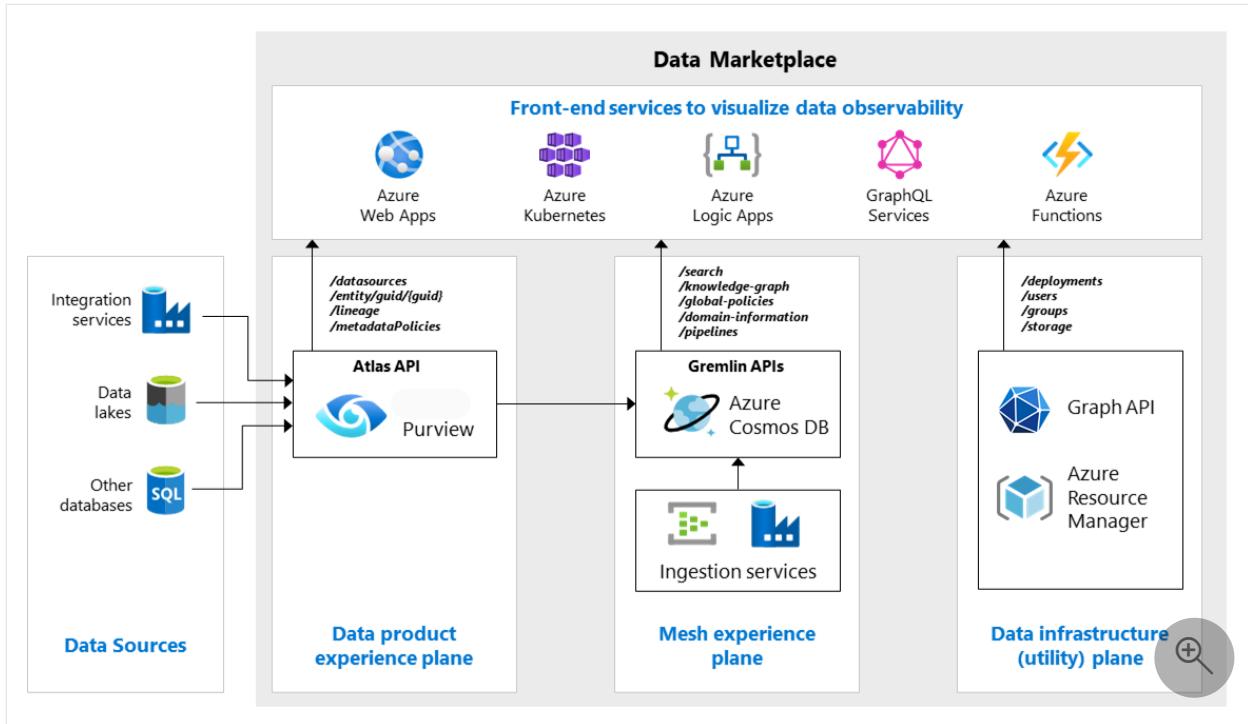
# Data marketplace

Article • 12/10/2024

Data marketplaces have a strong relationship with [metadata](#). A data marketplace offers data consumers an intuitive, secure, centralized, and standardized data shopping experience. It brings data closer to data analysts and scientists by utilizing the underlying metadata. It also tracks all your data products, which are often stored across a range of data domains.

To democratize data via your data mesh architecture, focus on several important areas:

- **Data Product Experience Plane:** Allows data providers and data consumers to work together on what data can be made available. Interfaces should provide extensive search capabilities that allow users to search using keywords, business terms, and natural language. Collaboration in data democratization is often linked to data catalogs or fully managed metadata management services that enable metadata search and discovery. [Microsoft Purview](#) is a proven approach to having a self-service collaboration portal. It supports [data discovery](#), including glossaries and [classifications](#). Data discovery enables your data consumers to easily find data. Microsoft Purview also supports [data owner access policies](#) so you can provide self-serve data access.
- **Data Infrastructure (Utility) Plane:** Helps you automate the deployment and provisioning of common and reusable consumption patterns. Consumption patterns can include storage accounts, databases, compute, identity management, and so on. Best practices for allowing your users to set up and launch their own data services can be found in [Organize data operations team members for cloud-scale analytics in Azure](#), [Deployment and management services](#), and [Development services](#).
- **Data Mesh Experience Plane:** Helps you keep sight of the health status of all interfaces, data pipelines, [data contracts](#), provisioned components, central tools, and so on. [Azure Monitor](#) helps you maximize the availability and performance of your applications and services and achieve monitoring and insight. For data observability, create an umbrella on top of your self-service collaboration portal and other metadata services. Consider designing your own [metadata lake](#) using services like [Azure Cosmos DB](#) and [Azure Event Hubs](#).



A data marketplace is typically a thin orchestration layer with an appealing look and feel, which offers unique user experiences. Data marketplaces utilize underlying metadata repositories, which can be a mixture of homegrown **metadata** stores and services like [Microsoft Purview](#). You can extend your data marketplace with extra analytical capabilities like [Cognitive Services](#) and [Machine Learning](#). For more information on how to adopt AI/ML in data mesh, see [Operationalize data mesh for AI/ML](#).

Building a data marketplace involves structure, culture, and people. It requires you to trust users, train people, and work on awareness. Don't underestimate these activities. Your users are valuable resources; they own or use specific parts of the data landscape. Making better use of your users increases the efficiency of your data knowledge and usage.

In some cases, you might need an external data marketplace. External data marketplaces enable sharing of your data products with external partners. You can use [Azure Data Share](#) as a component.

## Next Steps

[Master Data Management in Data Mesh.](#)

## Feedback

Was this page helpful?

Yes

No

# Manage master data in data mesh

Article • 11/27/2024

Enterprises using a data mesh architecture often have a large number of domains, each containing unique systems and data.

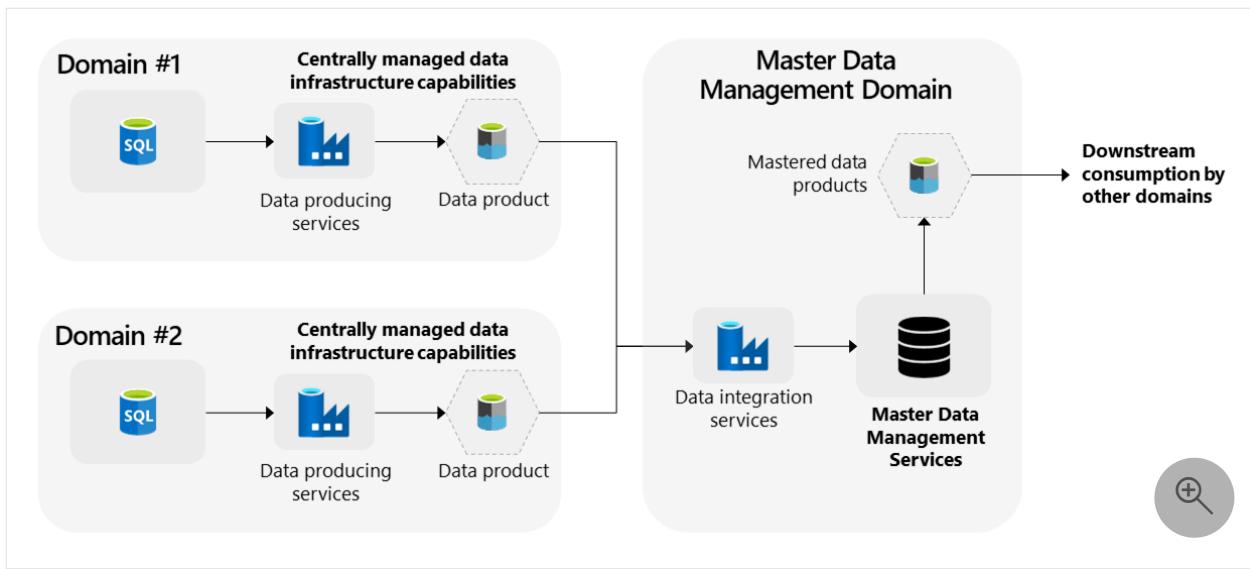
This widespread distribution of data increases complexity because multiple versions of the same data might exist in this setup. Integration requires more effort because owners have to integrate and harmonize all the different parts of the same data from multiple domains. Data can be inconsistent between these different domains, and data quality can also vary. Apply [master data management \(MDM\)](#) to address these challenges.

## Domain-oriented master data management

Master identification numbers are an important aspect of your MDM. These numbers link together mastered data and data from your domains. They're critical to your ability to track down what data has been mastered and what data belongs together. You can only identify unique data and assign master identification numbers centrally, not locally within a system. Your master data from different systems must be together within your MDM solution.

MDM works differently in domain-oriented architectures due to their distributed nature. Consistency is harder to achieve because you rely on MDM within your domains.

One way to achieve consistency is to ask your domains to conform to centrally managed master data when distributing [data products](#). You can publish a list of master data in a master data store or central repository. Your domain can classify data using the enterprise reference identifiers from your enterprise reference data when distributing data products across other domains. This lets your other domains quickly recognize any master data within those data products.



You can also create new MDM domains when grouping your MDM activities and using a master data store as a centralized repository. Each new MDM domain should contain a specific data subject that the identification and control of your master data focus on. Some well-known examples of this data include customers, products, employees, geographical locations, and finance and risk information. Mastered data from these MDM domains must find its way back to other domains. This distribution of data is similar to the distribution of your data products.

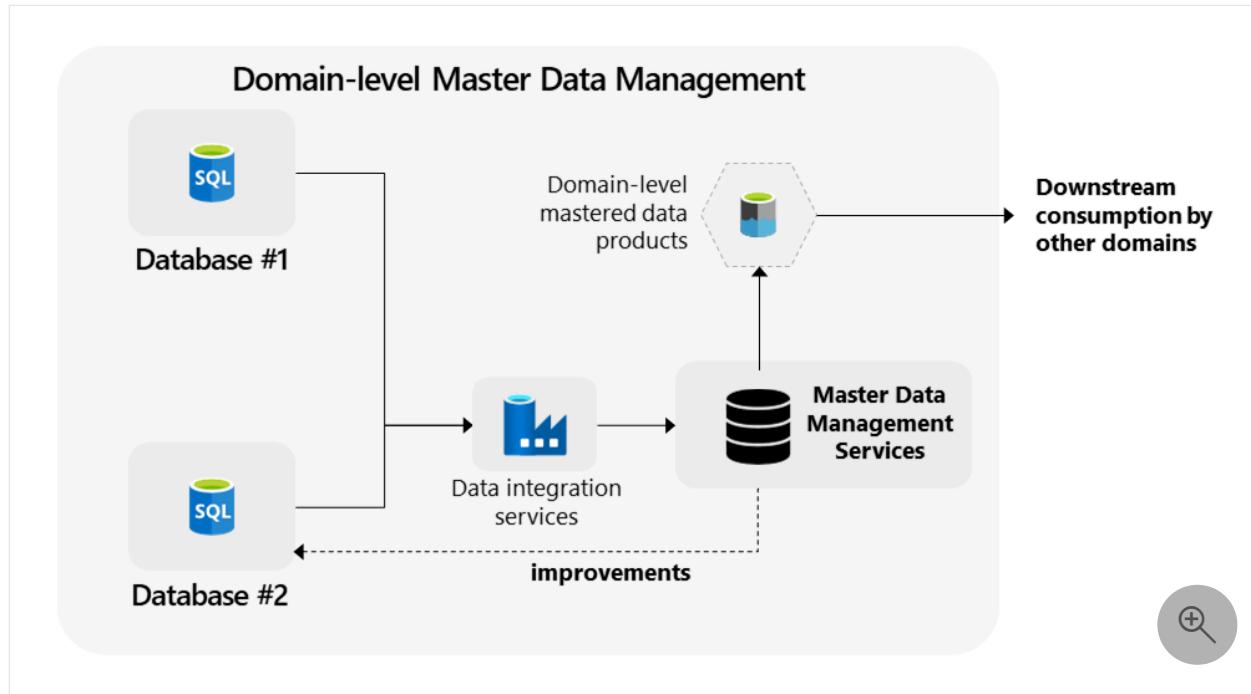
You can scope master data management and allow different approaches to data product distribution. Within the boundaries of a certain scope, data products don't have to conform to enterprise master data, but beyond the scope's boundaries, data products must conform. You can also apply this pattern in reverse, requiring adherence to master data only within a specific scope and not outside it. In these setups, your mastered data is centrally managed within your MDM solution. Your domains need to exchange master data so they know which local data to map to central master data. Identify and maintain these relationships so you know which data has been mastered and which data you can quickly link together. If a local domain keys in an operational system change, a master identifier is the only element binding everything together.

When you distribute master identifiers, don't extrapolate your MDM master identifiers to all source systems. Doing so can cause consistency issues. Only your applications or systems that are subject to MDM should obtain a master identifier from your MDM hub. Systems not subject to MDM should use their own local (domain) integrity.

## Domain-level master data management

When you look for overlapping data, you'll likely discover various degrees of overlap. Some data is generic and spans many domains. Other data has limited overlap and only spans a few domains. Distinguish the amount of data overlap and its importance by

extending MDM to domain-level MDM. You can do this by creating partial views of your master data within a specific scope. This is useful when your data is shared between some, but not all, of your domains.



It's important that overlapping domains manage data but have no central dependency. MDM solutions can help you achieve this. You can simplify usage tremendously by abstracting away infrastructure and providing MDM as a service to your domains. If you use a central solution, apply segregated views for each individual domain or scope.

## Achieve consistency with reusable components

Code sharing is another way to ensure master data collaboration and reusability. Instead of sharing master data, you share the underlying code (snippets and scripts) to generate outputs and promote effective reuse. Store this underlying code in a central and open repository with version control. Your teams can all contribute to and improve upon code that lives in this repository.

In this model, you apply business logic only within domains. Your teams can deviate, make improvements, or use slightly optimized versions of the logic as they see fit. You can regenerate your outputs as improvements from your community are added to your central code repository.

Note that allowing your teams to modify their code can make comparing results between various teams more difficult, which can impact consistency.

## Master data management summary

Users can only make correct decisions if the data they use is consistent and correct. By using MDM, you can ensure your data's consistency and quality at the enterprise level.

Your organization must find the correct balance for MDM. Having too many areas of master data or reference values leads to too much cross-domain alignment. Having no enterprise data at all makes it impossible to compare any results. A practical way to begin using MDM in a balanced way is to implement a repository. This is the simplest way to manage your organization's master data. With a repository, you don't need to adjust your domain systems to learn what data is low quality or needs to be aligned. With a repository helping you gain that information, you can deliver value more quickly.

After you implement a repository, you need to outline a clear scope. Don't fall into the trap of enterprise data unification by selecting all data. Only master data from your most important fields. Begin by selecting subjects that add the most value, like customers, contracts, products, and organizational units. Your number of attributes should be in the tens, not the hundreds or thousands.

Align your processes and governance after you've come to an agreement with your domains. Make any agreements on timelines and reviews clear to all domains. Also, make sure you work on your metadata. Catalog your master data. Ensure your domains know which data elements are candidates from which source systems, and how those elements flow through your data pipelines.

The final step, and your ultimate goal, is achieving coexistence. Your improvements should flow directly back to your domains. This is the most difficult part of the process because it requires you to make many architecture changes. Your domains need to be able to handle corrections and improvements sent from your centrally managed MDM solution.

## Next steps

- Operationalize data mesh for AI/ML domain driven feature engineering
- 

## Feedback

Was this page helpful?

 Yes

 No

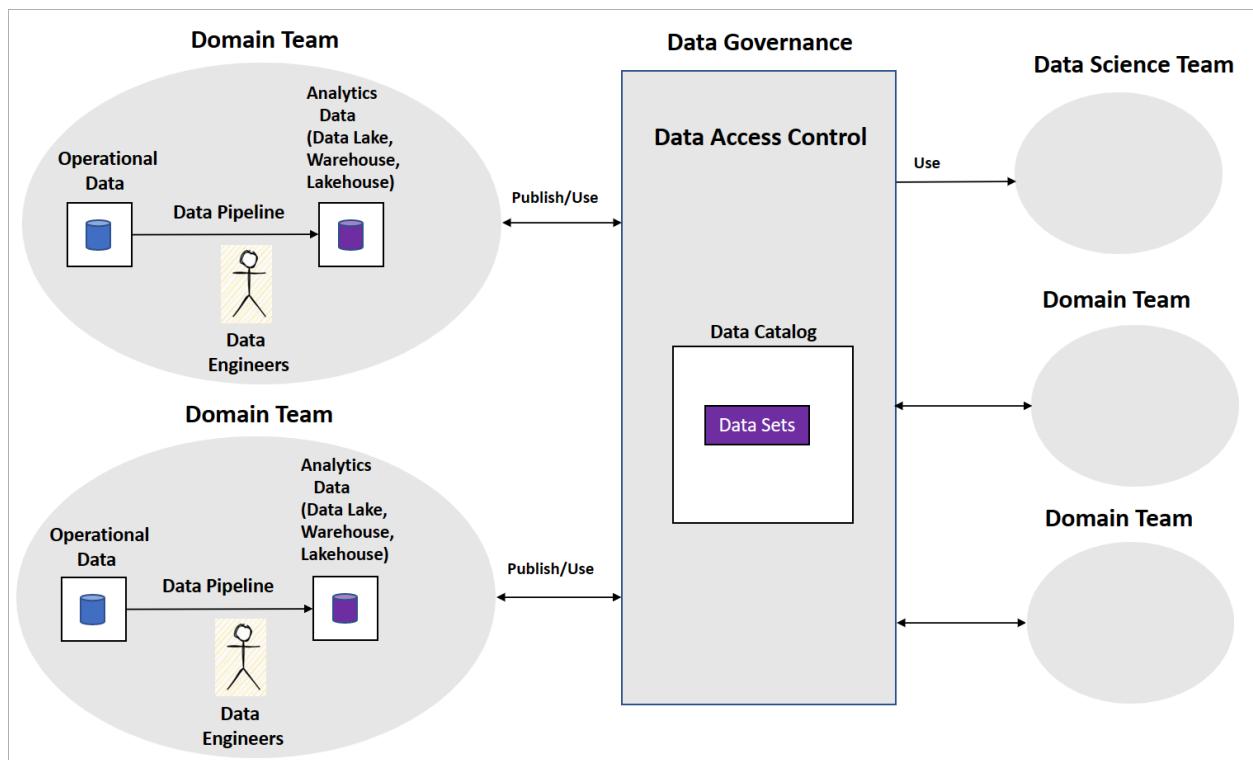
# Operationalize data mesh for AI/ML domain driven feature engineering

Article • 11/27/2024

Data mesh helps organizations move from a centralized data lake or data warehouse to a domain-driven decentralization of analytics data underlined by four principles: Domain Ownership, Data as a Product, Self-serve Data Platform, and Federated Computational Governance. [Data mesh](#) provides the benefits of distributed data ownership and improved data quality and governance that accelerates business and time to value for organizations.

## Data mesh implementation

A typical data mesh implementation includes domain teams with data engineers who build data pipelines. The team maintains operational and analytical data stores, like data lakes, data warehouses, or data lakehouses. They release the pipelines as [data products](#) for other domain teams or data science teams to consume. Other teams consume the data products using a central data governance platform as shown in the following diagram.



Data mesh is clear on how data products serve transformed and aggregated data sets for business intelligence. But it's not explicit about the approach organizations should take to build AI/ML models. Nor is there guidance on how to structure their data science

teams, the AI/ML model governance, and how to share AI/ML models or features among domain teams.

The following section outlines a couple of strategies that organizations can use to develop AI/ML capabilities within data mesh. And you see a proposal for a strategy on domain-driven feature engineering or feature mesh.

## AI/ML strategies for data mesh

One common strategy is for the organization to adopt data science teams as data consumers. These teams access various domain data products in data mesh as per the use case. They perform data exploration and feature engineering to develop and build AI/ML models. In some cases, domain teams also develop their own AI/ML models by using their data and other teams' data products to extend and derive new features.

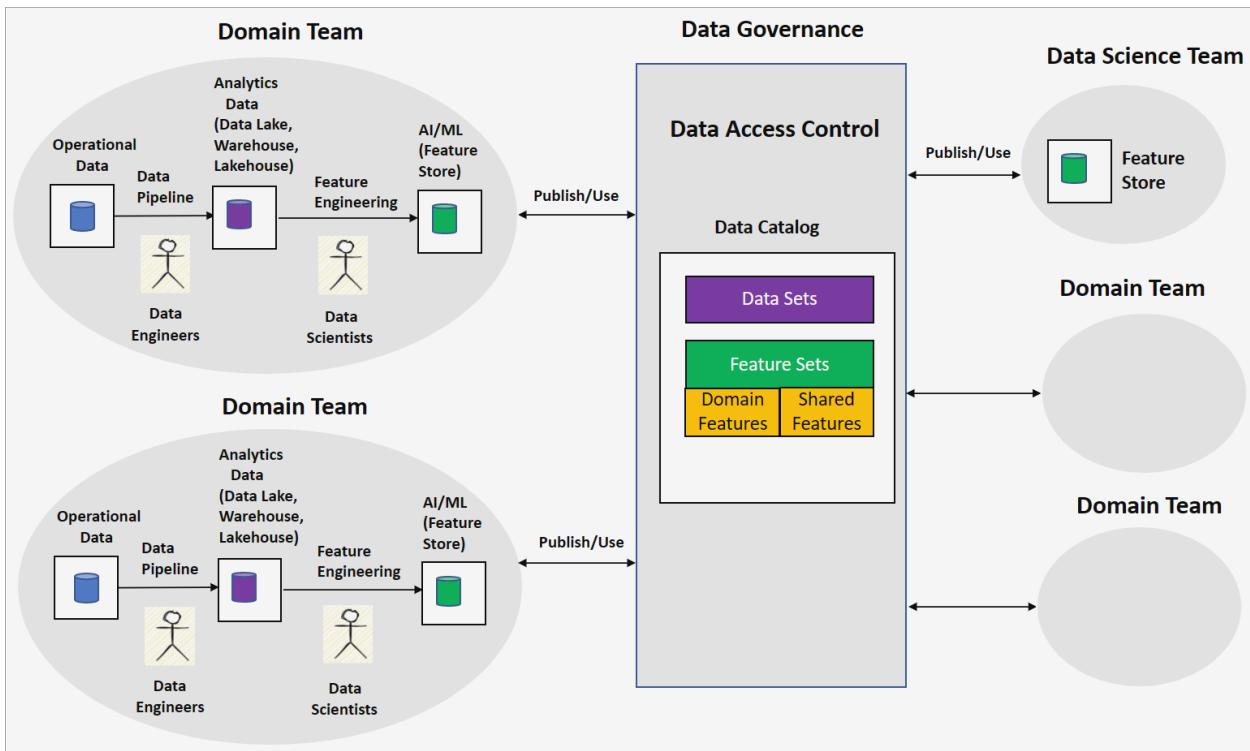
[Feature engineering ↗](#) is the core of model building and is typically complex and requires domain expertise. This strategy can be time-consuming since data science teams need to analyze various data products. They might not have complete domain knowledge to build high-quality features. Lack of domain knowledge can lead to duplicate feature engineering efforts between domain teams. Also, issues like AI/ML model reproducibility due to inconsistent feature sets across teams. Data science or domain teams need to continuously refresh features as new versions of data products are released.

Another strategy is for domain teams to release AI/ML models in a format like Open Neural Network Exchange (ONNX), but these results are black boxes and combining AI/ML models or features across domains would be difficult.

Is there a way to decentralize the AI/ML model building across domain and data science teams to address the challenges? The proposed domain-driven feature engineering or feature mesh strategy is an option.

## Domain driven feature engineering or feature mesh

The domain-driven feature engineering or feature mesh strategy offers a decentralized approach to AI/ML model building in a data mesh setting. The following diagram shows the strategy and how it addresses the four main principles of data mesh.



## Domain ownership feature engineering by domain teams

In this strategy, the organization pairs data scientists with data engineers in a domain team to run data exploration on clean and transformed data in, for example, a data lake. Engineering generates features that store in a feature store. A feature store is a data repository that serves features for training and inference and helps track feature versions, metadata, and statistics. This capability lets the data scientists in the domain team work closely with domain experts and keep the features refreshed as data changes in the domain.

## Data as a product: Feature sets

Features generated by the domain team, known as domain or local features, are published to the data catalog in the data governance platform as feature sets. These feature sets are consumed by data science teams or other domain teams for building AI/ML models. During AI/ML model development, the data science or domain teams can combine domain features to produce new features, called shared or global features. These shared features are published back to the feature sets catalog for consumption.

## Self-serve data platform and federated computation governance: Feature

# standardization and quality

This strategy can lead to adopting a different technology stack for feature engineering pipelines and inconsistent feature definitions between domain teams. Self-serve data platform principles ensure that domain teams are using common infrastructure and tools to build the feature engineering pipelines and enforce access control. The Federated Computational Governance principle ensures interoperability of feature sets through global standardization and checks on feature quality.

Using the domain-driven feature engineering or feature mesh strategy offers a decentralized AI/ML model building approach for organizations to help reduce time in developing AI/ML models. This strategy helps keep features consistent across domain teams. It avoids duplication of efforts and results in high-quality features for more accurate AI/ML models, which increase the value to the business.

## Data mesh implementation in Azure

This article describes the concepts around operationalizing AI/ML in a data mesh and doesn't cover tools or architectures to build these strategies. Azure has feature store offerings like [Azure Databricks](#) feature store and [Feathr](#) from LinkedIn. You can develop [Microsoft Purview](#) custom connectors to manage and govern feature stores.

## Next steps

- [Getting started with data mesh checklist](#)
- 

## Feedback

Was this page helpful?

 Yes

 No

# Getting started with data mesh checklist

Article • 12/10/2024

During your data journey with cloud-scale analytics, you'll find there are multiple stages in your adoption lifecycle. This section provides a quick getting started checklist to help you adopt your scenario in stages. These stages are:

- Stage 1: First landing zone
- Stage 2: Additional data domains
- Stage 3: Improve consumption readiness
- Stage 4: Critical governance components

## Stage 1 - First landing zone

- Define your first use case(s)
- Deploy your first data management landing zone
- Deploy your first data landing zone
- Define your first ingestion pattern (for example, batch parquet)
- Develop your first data product (ingested raw, abstracted to product)
- Determine 'just-enough' governance
- Define metadata requirements (application information, schema metadata)
- Register your first data consumer (manual process)

## Stage 2 - Additional data domains

- Refine your target architecture
- Deploy more data landing zones
- Extend with second, third, and fourth data products
- Realize your data product metadata repository (database or Excel)
- Implement your first set of controls (data quality, schema validation)
- Realize your consuming pipeline (taking input as output)
- Establish data ownership

## Stage 3 – Improve consumption readiness

- Implement self-service registration and metadata ingestion
- Offer other transformation patterns (transformation framework, ETL tools, etc.)
- Enrich controls on the provider side (glossary, lineage, linkage)

- Implement your consuming process: approvals, use case metadata, deploy secure views by hand
- Establish your data governance control board

## Stage 4 – Critical governance components

- Apply automation (automatic secure view provisioning)
- Deploy strong data governance, set up your dispute body
- Finalize your data product guidelines
- Define your extra interoperability standard
- Develop your self-service data consumption process
- Develop your data query, self-service, catalog, lineage capabilities, etc.
- Develop more data marketplace capabilities

## Summary

These four development stages allow you to set up a minimal viable product in stage one, learn, and iterate into stage two. Throughout your staged approach, you grow in maturity for creating a self-service, scalable, and governed platform.

## Next steps

- [Overview of reference architectures for cloud-scale analytics in Azure](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Overview of reference architectures for cloud-scale analytics in Azure

Article • 03/13/2023

Cloud-scale analytics is designed to be modular. It allows customers to start with a small footprint and grow over time. Customers should decide ahead how to organize data domains across data landing zones. The building blocks can be deployed through the Azure portal, GitHub Actions workflows, and Azure Pipelines. The template repositories for the data management landing zone, data landing zone, and data integrations/products contain sample YAML pipelines to help you get started faster with setting up your environments.

## ⓘ Note

The template repositories can be used to deploy the reference architectures listed in this article. Links to these repositories are provided in the detailed description of each architecture.

## Reference architectures examples

The following architecture examples can help you to adapt cloud-scale analytics to your use case.

Scenario	Example customer	Description
Single data landing zone	<a href="#">Adatum Corporation</a>	This reference architecture is ideal for customers that have identified a unit of their business that's ready to deploy analytics workloads to Azure. This architecture deploys a single landing zone that can be used by the business unit to manage their data estate. It provides the flexibility to add more landing zones for other business units when they're ready to move to Azure.
Multiple data landing zones	<a href="#">Relecloud</a>	This reference architecture is relevant to customers that have already implemented a basic version of cloud-scale analytics and are now ready to host a new business that modernizes its analytics operations. It demonstrates a more complex scenario with multiple landing zones, data integrations, and data products.

Scenario	Example customer	Description
Highly sensitive data landing zones	<a href="#">Lamna Healthcare</a>	This reference architecture is for customers that want to use cloud-scale analytics not only for scalability but also to secure their data. It demonstrates how access to sensitive data can be controlled and how appropriately desensitized data can be shared with analysts.
Financial institution scenario for data mesh	<a href="#">Woodgrove Bank</a>	This reference architecture is written for customers that want to use cloud-scale analytics for a data mesh analytical data architecture and operating model. It demonstrates a more complex scenario with multiple landing zones, data integrations, and data products.

## Next steps

- [Single data landing zone scenario](#)
- [Multiple data landing zones scenario](#)
- [Highly sensitive data landing zones scenario](#)
- [Financial institution scenario for data mesh](#)

# Adatum Corporation scenario for cloud-scale analytics in Azure

Article • 12/10/2024

Cloud-scale analytics is modular by design and allows organizations to start with foundational landing zones that support their data and analytics workloads, regardless of whether the projects are being migrated or are newly developed and deployed to Azure. The architecture enables organizations to start as small as needed and scale alongside their business requirements regardless of scale point.

## Customer profile

This reference architecture is ideal for customers who identified a unit of their business that's ready to deploy analytics workloads to Azure. This architecture deploys a single landing zone that can be used by the business unit to manage their data estate. It provides the flexibility to add more landing zones for other business units when they're ready to move to Azure.

Adatum Corporation is a large, international enterprise. In addition to the centralized business units at their headquarters, they also have subsidiaries around the globe that have their own business units, including accounting, marketing, sales, support, and operations.

All of these disparate groups are producing their own data. Many of the business units have embedded analytics teams. The central IT organization has provided most of the data platform that's in use, but a few business units have gone rogue and implemented their own solutions. The data platform is composed of various cloud services and on-premises solutions.

The company's vision is to have a centralized analytics platform, a single source of truth for all data. However, it has become challenging for many different stakeholders to buy into one single technology. Given the rate at which new data is being created and new options become available even early drafts of plans for centralization quickly become outdated. Meanwhile, the corporate sales team has outgrown their current solution, and the company urgently needs to use new analytics to pursue a new market segment.

Adatum has decided to implement cloud-scale analytics pattern in Azure to solve this problem. The enterprise is confident that cloud-scale analytics allows the corporate sales team to migrate their data platform today but still provide enough flexibility to accommodate other business units when they're ready to join.

## Current situation

The Adatum corporate sales group uses traditional ERP and CRM systems to process its sales transactions. Data from these systems needs to be exported to a separate analytics platform so that stakeholders across the organization can access the data and enrich it for their various projects.

## Architectural solution

In this reference architecture, we deploy a data management landing zone, which is needed for all ESA implementations, and a single data landing zone, which can be used by the corporate sales department.

### Data management landing zone

A critical concept for every cloud-scale analytics is having one data management landing zone. This subscription contains resources shared across all of the landing zones and includes shared networking components like a firewall and private DNS zones. It also includes resources for data and cloud governance. Microsoft Purview and Databricks Unity Catalog are deployed as services at tenant level.

### Data applications

The landing zone has two [data applications](#). The first integration ingests data related to customers. This step includes the customer records and their related records (like addresses, contacts, territory assignments, and contact history). This data is imported from the Adatum CRM system.

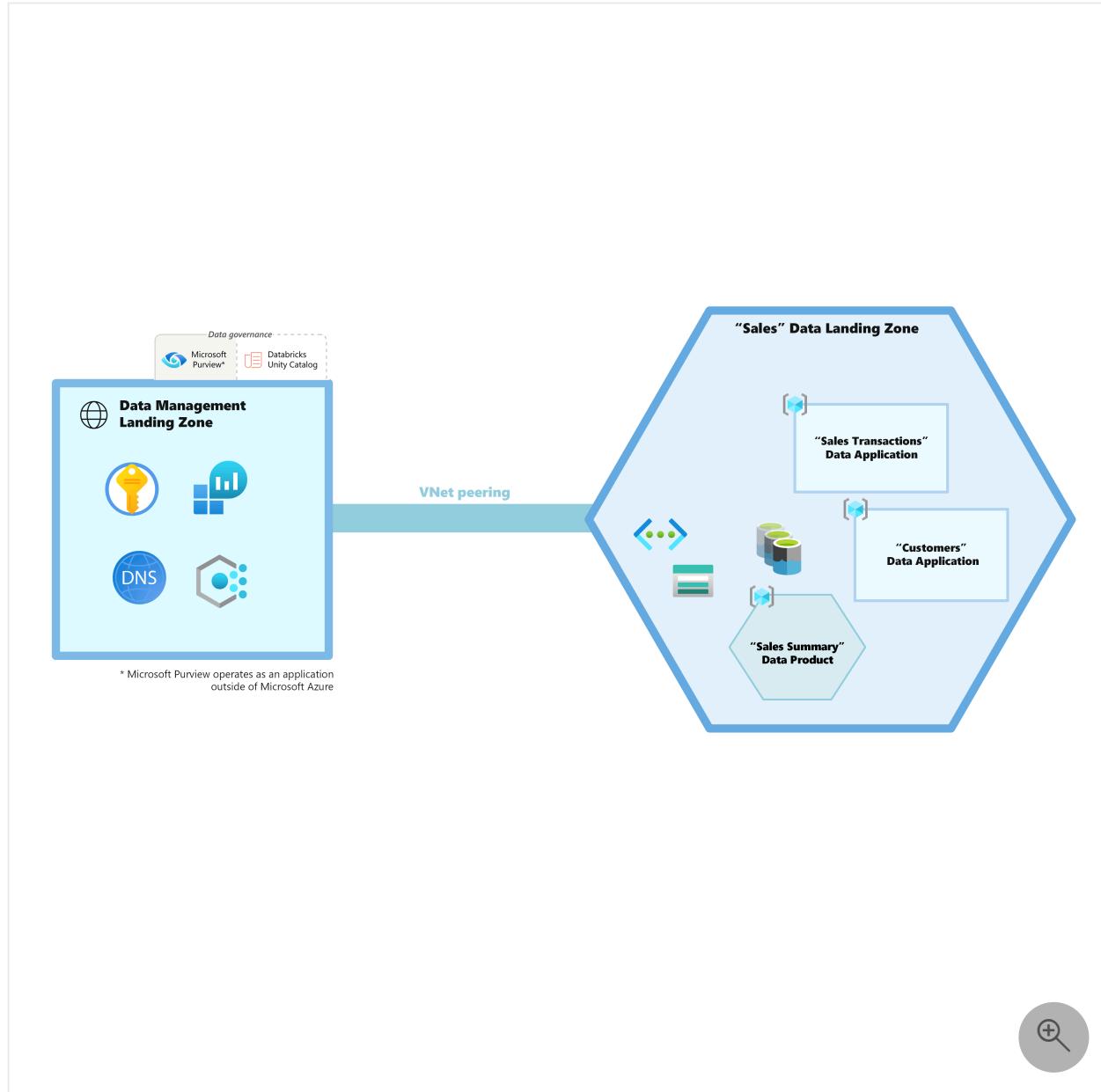
The second data application ingests sales transactions. This includes transaction headers, line item details, shipping records, and payments. All of these records are ingested from the Adatum ERP system.

These integrations won't transform or enrich the data. They only copy the data from the source systems and land it in the analytics platform. This allows many data products to consume the data in a scalable manner without putting another burden on the source system.

### Data products

In this example, Adatum has one data product. This product combines raw data from the two Data applications and transforms them into a new dataset. From there, it can be

picked up by business users for extra analysis and reporting with tools like Microsoft Power BI.



*Figure 1: Diagram of architecture. Not all Azure services are represented in the diagram. It's simplified to highlight the core concepts of how resources are organized within the architecture.*

## Rationale

### Why not put sales transactions and customers in their own data landing zones?

One of the first decisions enterprises must make about their cloud-scale analytics is how to divide the entire data estate into landing zones. Data solutions that frequently communicate with one another are strong candidates for inclusion in the same landing

zone. This decision allows enterprises to reduce the costs associated with moving data across peered VNets. In this example, sales transaction data will frequently be linked to customer data. Therefore, it makes sense to store these related Data applications in the same data landing zone.

An extra consideration for landing zones is how the teams responsible for the data are aligned within the organization. In this case, the two Data applications are owned by different teams, but those teams are both part of the sales and marketing division at Adatum.

## Why not let sales transactions and customers share one Data application?

By separating the customer data and the sales transaction data in their own Data applications, we allow the subject matter experts for those domains to make the best decisions for their particular data products. They can choose the access patterns, ingestion engines, and storage options that best meet their needs without conflicting with one another.

For example, the team that has expertise with the CRM system will be responsible for the customer Data application. Based on the team's skill set and the technologies used by the CRM system, they decide which tools best suit their needs. They won't have to worry if these decisions will also work for the sales transactions team. That team is using their own toolset and won't have to compromise to meet the requirements of the customers' team.

## Why move the sales team to the new data platform?

In this example, the corporate sales team is the first to move to the new cloud-scale analytics. The solution is designed to be scalable above all else. As other business units are ready to migrate, more landing zones can be added to accommodate their workloads.

## How to evolve in the future?

Scaling is accomplished by adding more landing zones to the architecture. These landing zones use virtual network peering to connect to the data management landing zone and all of the other landing zones. This mesh pattern allows data products and resources to be shared across zones. By splitting into different zones, the workloads are spread across Azure subscriptions and resources. This step allows enterprises to avoid reaching the limits of the Azure services and continue to grow their data estates.

# Next steps

Continue to the [Relecloud scenario for cloud-scale analytics in Azure](#).

Learn more in:

- [Overview of the data landing zone](#)
  - [Overview of the data management landing zone](#)
  - [Cloud-scale analytics data products in Azure](#)
- 

## Feedback

Was this page helpful?



# Multiple data zones for cloud-scale analytics in Azure

Article • 12/10/2024

This reference architecture is for organizations that have implemented a basic version of cloud-scale analytics and are now ready to host new business units to help modernize their analytics operations. This more complex scenario uses multiple landing zones, data applications, and data products.

*Apache Hive and the Hive logo are either registered trademarks or trademarks of the Apache Software Foundation in the United States and/or other countries. No endorsement by The Apache Software Foundation is implied by the use of these marks.*

## Problem statement

Relecloud, the fictional company in this example, is a private cloud provider that offers shared computing and storage resources to global organizations. Although Relecloud provides compute resources, they don't want to constrain their platform with their own internal operations. Therefore, they rely on Microsoft Azure for their internal computing needs.

Data analysts in the operations group use telemetry data from cloud services to understand how their customers use the platform. A separate team of analysts in the billing group studies invoicing data to gain insights about which services generate the most revenue.

Last quarter, the operations team modernized its analytics platform by migrating it to Azure. One goal in implementing cloud-scale analytics was to maximize the potential for scaling the platform and adding new organizational workloads.

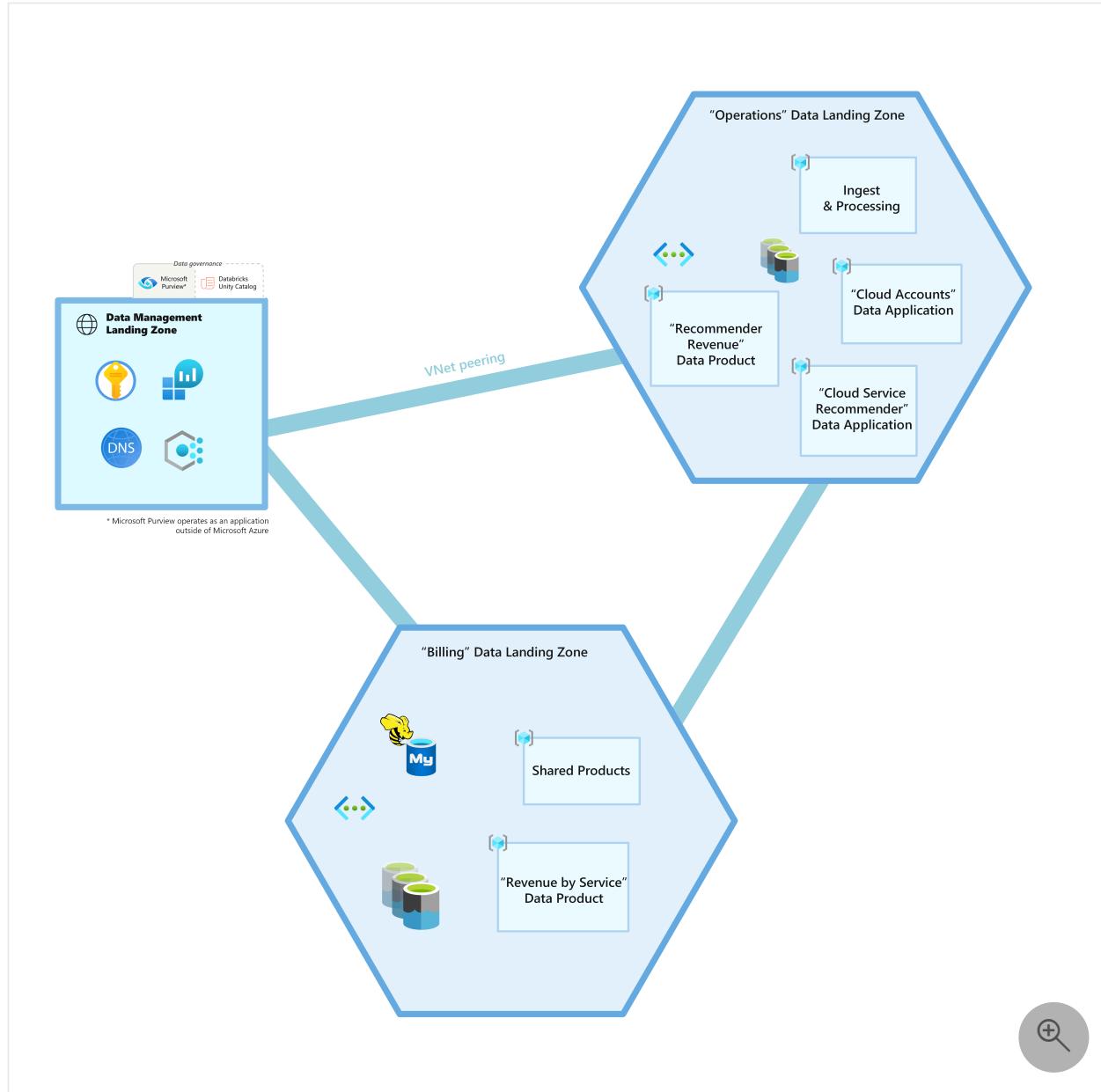
Today, the billing group has outgrown its current analytics solution. The volume of invoices to analyze is too large for their on-premises server. The team decides to follow the lead of the operations group and modernize their data analytics platform in Azure.

Analysts in the billing group have different skills than analysts in the operations group. The billing analysts don't want to be constrained to use the same tools as operations. The billing group is in a different part of the organization and wants the flexibility to implement the policies and procedures that meet their needs.

## Architectural solution

Relecloud scales their analytics platform by adding a new landing zone for the billing group. This landing zone provides a virtual workspace for the billing group to implement the analytics solutions that meet their business needs. By having a landing zone separate from the organization's other resources, the billing group can implement their own access policies and account for the costs of their services.

The following diagram doesn't represent all Azure services. The diagram is simplified to highlight the core concepts of organizing resources within the architecture.



## Data management landing zone

A key requirement for a cloud-scale analytics implementation is a data management landing zone. This subscription contains resources that are shared across all landing zones, including shared networking components like a firewall or private DNS zones. It also includes resources for data and cloud governance. Microsoft Purview and Databricks Unity Catalog have been deployed as services at the tenant level.

Relecloud created a data management landing zone when they deployed the data analytics solution for the operations group. When the billing group joins the platform, they use the same data management landing zone to share common resources with the operations group.

## Operations data landing zone

The operations group has the following solutions in its data landing zone.

### Operations data applications

The team has built a [source-aligned data application](#) that uses Apache Spark jobs in Azure Databricks to ingest service telemetry data and store it in an Azure Data Lake Storage account.

This process copies the data as-is from the source system but doesn't transform it. Analysts can work with the copied data in the analytics platform without overloading the source system. Instead of creating a dedicated deployment for this data application, the operations team uses the Databricks workspace in the shared **Ingest & Processing** resource group.

Relecloud customers can create cloud accounts to manage resources and billing in their private clouds. Each customer can have multiple accounts. The analytics team built a data application to import the cloud account data. Because the volume and frequency of data are much lower than for telemetry data, the team doesn't need to use Spark jobs. Instead, they created Azure Data Factory pipelines to copy the data.

Azure Database for MySQL acts as the Hive metastore, and Azure SQL Database is the Azure Data Factory metastore.

### Operations data products

Relecloud analysts get value from the data in the source-aligned data applications by creating new, consumer-aligned data applications. One of these consumer-aligned data applications is a **Cloud service recommender** model. Relecloud data scientists used Azure Machine Learning to build a model that looks at the services a cloud account consumes and suggests related services that could be useful. The team deploys this model to an Azure Kubernetes Service (AKS) cluster running in the landing zone and managed by Azure Machine Learning. Applications that run outside of cloud-scale analytics can call the AKS endpoint to get recommendations.

After the billing team creates their landing zone, the operations team creates a new data product that their management team requests. The management team wants to know how much revenue the **Cloud service recommender** data application generates. The new **Recommender revenue** data product uses Azure Synapse Analytics to combine data from **Cloud service recommender** and **Revenue by service** into a new data product. Business analysts can connect to Azure Synapse with Microsoft Power BI to find and report insights from this new data product.

## Billing data landing zone

The billing group was using an on-premises system to power their analytics, but as the data volume grew and the company relied more on their work, the system couldn't keep pace. The group modernizes their platform by moving to the cloud.

The billing group doesn't share a landing zone with the operations group but gets their own landing zone where they have the freedom to build the platform that best suits their needs. The new landing zone is connected to the data management landing zone and all other data landing zones with virtual network peering. This mechanism enables data to be shared securely through the Azure internal network.

## Billing data applications

To land data from existing systems into the analytics platform, the billing group builds two data applications. The first application ingests the customer data, including the full list of customers and all related data, such as customer addresses, locations, and salesperson assignments. The second application imports the company's invoice history, which includes all billing charges to customers and the related payment data.

Both of these applications are powered by pipelines in the shared Azure Synapse workspace. Each application has a dedicated compute pool to facilitate cost accounting and security boundaries. Since the applications can be fully implemented with shared resources, the billing group doesn't have to create a deployment for these data applications.

## Billing data product

The billing analysts create a new data product called **Revenue by service** that analyzes how much revenue each cloud service generates for Relecloud. This product relies on the data in the **Invoices** ingestion. The product also connects to the operations landing zone and reads the service usage data. Like the data applications, the data product also relies on the shared Azure Synapse workspace.

# Next steps

Continue to the [Lamna Healthcare scenario for secure cloud-scale analytics in Azure](#).

For more information, see the following articles:

- [Azure Machine Learning as a data product for cloud-scale analytics](#)
- [Use Azure Synapse Analytics with cloud-scale analytics](#)

---

## Feedback

Was this page helpful?



Yes



No

# Azure Synapse Analytics for landing zones

Azure Synapse Analytics

Azure Private Link

Azure Data Lake Storage

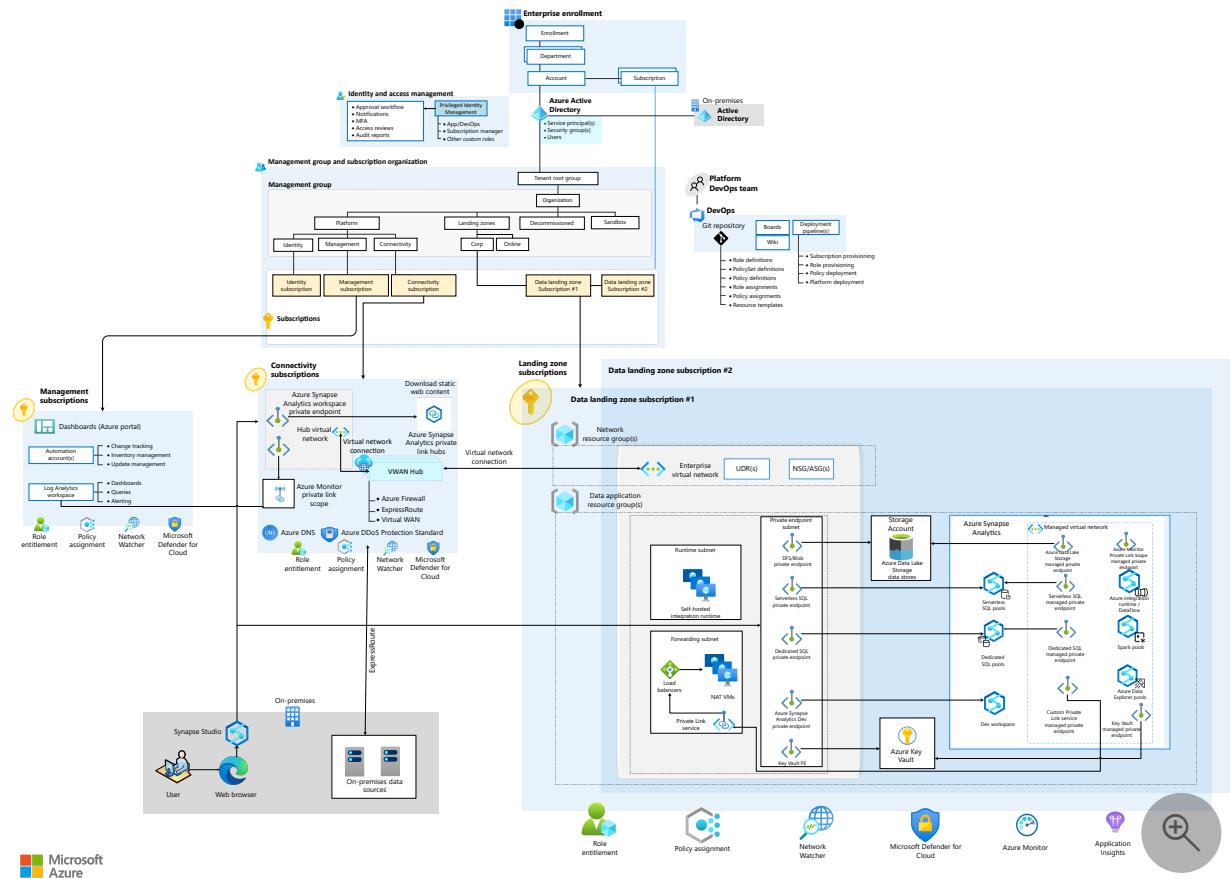
Azure Key Vault

This article provides an architectural approach for preparing Azure landing zone subscriptions for a scalable, enhanced-security deployment of Azure Synapse Analytics. Azure Synapse, an enterprise analytics service, combines data warehousing, big data processing, data integration, and management.

The article assumes that you've already implemented the platform foundation that's required to effectively construct and operationalize a [landing zone](#).

*Apache®, Spark®, and the flame logo are either registered trademarks or trademarks of the Apache Software Foundation in the United States and/or other countries. No endorsement by The Apache Software Foundation is implied by the use of these marks.*

## Architecture



Download a [Visio file](#) of this architecture.

# Dataflow

- The core component of this architecture is Azure Synapse, a unified service that provides a range of functions, from data ingestion and data processing to serving and analytics. Azure Synapse in a [Managed Virtual Network](#) provides network isolation for the workspace. By enabling [data exfiltration protection](#), you can limit outbound connectivity to only approved targets.
- Azure Synapse resources, the Azure integration runtime, and Spark pools that are located in the Managed Virtual Network can connect to Azure Data Lake Storage, Azure Key Vault, and other Azure data stores with heightened security by using [Managed private endpoints](#). Azure Synapse SQL pools that are hosted outside the Managed Virtual Network can connect to Azure services via private endpoint in the enterprise virtual network.
- Administrators can enforce private connectivity to the Azure Synapse workspace, Data Lake Storage, Key Vault, Log Analytics, and other data stores via Azure policies applied across data landing zones at the management group level. They can also enable data exfiltration protection to provide enhanced security for egress traffic.
- Users access Synapse Studio by using a web browser from a restricted on-premises network via Azure Synapse [Private Link Hubs](#). Private Link Hubs are used to load Synapse Studio over private links with enhanced security. A single Azure Synapse Private Link Hubs resource is deployed in a Connectivity subscription with a private endpoint in the hub virtual network. The hub virtual network is connected to the on-premises network via [Azure ExpressRoute](#). The Private Link Hubs resource can be used to privately connect to all Azure Synapse workspaces via Synapse Studio.
- Data engineers use the Azure Synapse pipelines Copy activity, executed in a [self-hosted integration runtime](#), to ingest data between a data store that's hosted in an on-premises environment and cloud data stores like Data Lake Storage and SQL pools. The on-premises environment is connected via ExpressRoute to the hub virtual network on Azure.
- Data engineers use the Azure Synapse Data Flow activity and Spark pools to transform data hosted on cloud data stores that are connected to the Azure Synapse Managed Virtual Network via Managed private endpoints. For data located in the on-premises environment, transformation with Spark pools requires connectivity via custom Private Link service. The custom Private Link service uses Network Address Translation (NAT) VMs to connect to the on-premises data store. For information about setting up Private Link service to access on-premises data stores from a Managed Virtual Network, see [How to access on-premises SQL Server from Data Factory Managed VNet using Private Endpoint](#).

- If data exfiltration protection is enabled in Azure Synapse, Spark application logging to the Log Analytics workspace is routed via an [Azure Monitor Private Link Scope](#) resource that's connected to the Azure Synapse Managed Virtual Network via Managed private endpoint. As shown in the diagram, a single Azure Monitor Private Link Scope resource is hosted in a Connectivity subscription with private endpoint in the hub virtual network. All Log Analytics workspaces and Application Insights resources can be reached privately via Azure Monitor Private Link Scope.

## Components

- [Azure Synapse Analytics](#) is an enterprise analytics service that accelerates time to insight across data warehouses and big data systems.
- [Azure Synapse Managed Virtual Network](#) provides network isolation to Azure Synapse workspaces from other workspaces.
- [Azure Synapse Managed private endpoints](#) are private endpoints that are created in a Managed Virtual Network that's associated with an Azure Synapse workspace. Managed private endpoints establish private link connectivity to Azure resources outside the Managed Virtual Network.
- [Azure Synapse workspace with data exfiltration protection](#) prevents exfiltration of sensitive data to locations that are outside of an organization's scope.
- [Azure Private Link Hubs](#) are Azure resources that act as connectors between your secured network and the Synapse Studio web experience.
- [Integration runtime](#) is the compute infrastructure that Azure Synapse pipelines use to provide data integration capabilities across different network environments. Run the Data Flow activity in the managed Azure compute integration runtime or the Copy activity across networks by using a self-hosted compute integration runtime.
- [Azure Private Link](#) provides private access to services that are hosted on Azure. Azure Private Link service is the reference to your own service that's powered by Private Link. You can enable your service that's running behind Azure standard load balancer for Private Link access. You can then extend Private Link service to the Azure Synapse Managed Virtual Network via Managed private endpoint.
- [Apache Spark in Azure Synapse](#) is one of several Microsoft implementations of Apache Spark in the cloud. Azure Synapse makes it easy to create and configure Spark capabilities on Azure.
- [Data Lake Storage](#) uses Azure Storage as the foundation for building enterprise data lakes on Azure.
- [Key Vault](#) allows you to store secrets, keys, and certificates with enhanced security.
- [Azure landing zones](#) are the outputs of a multi-subscription Azure environment that account for scale, security governance, networking, and identity. A landing

zone enables migration, modernization, and innovation at enterprise scale on Azure.

## Scenario details

This article provides an approach for preparing Azure landing zone subscriptions for a scalable, enhanced security deployment of Azure Synapse. The solution adheres to Cloud Adoption Framework for Azure best practices and focuses on the [design guidelines](#) for enterprise-scale landing zones.

Many large organizations with decentralized, autonomous business units want to adopt analytics and data science solutions at scale. It's critical that they build the right foundation. Azure Synapse and Data Lake Storage are the central components for implementing cloud-scale analytics and a data mesh architecture.

This article provides recommendations for deploying Azure Synapse across management groups, subscription topology, networking, identity, and security.

By using this solution, you can achieve:

- A well-governed, enhanced-security analytics platform that scales according to your needs across multiple data landing zones.
- Reduced operational overhead for data application teams. They can focus on data engineering and analytics and leave Azure Synapse platform management to the data landing zone operations team.
- Centralized enforcement of organizational compliance across data landing zones.

## Potential use cases

This architecture is useful for organizations that require:

- A fully integrated and operational control and data plane for Azure Synapse workloads, right from the start.
- An enhanced-security implementation of Azure Synapse, with a focus on data security and privacy.

This architecture can serve as a starting point for large-scale deployments of Azure Synapse workloads across data landing zone subscriptions.

## Subscription topology

Organizations building large scale data and analytics platforms look for ways to scale their efforts consistently and efficiently over time.

- By using [subscriptions as a scale unit](#) for data landing zones, organizations can overcome subscription-level limitations, ensure proper isolation and access management, and get flexible future growth for the data platform footprint. Within a data landing zone, you can group Azure Synapse and other data assets for specific analytics use cases within a resource group.
- The management group and subscription setup are the responsibility of the landing zone platform owner who provides the required access to data platform administrators to provision Azure Synapse and other services.
- All organization-wide data compliance policies are applied at the management group level to enforce compliance across the data landing zones.

## Networking topology

For recommendations for landing zones that use virtual WAN network topology (hub and spoke), see [Virtual WAN network topology](#). These recommendations are aligned with [Cloud Adoption Framework](#) best practices.

Following are some recommendations for Azure Synapse networking topology:

- Implement network isolation for Azure Synapse resources via Managed Virtual Network. Implement data exfiltration protection by restricting outbound access to approved targets only.
- Configure private connectivity to:
  - Azure services like Data Lake Storage, Key Vault, and Azure SQL, via Managed private endpoints.
  - On-premises data stores and applications over ExpressRoute, via a self-hosted integration runtime. Use custom Private Link service to connect Spark resources to on-premises data stores if you can't use a self-hosted integration runtime.
  - Synapse Studio, via private link hubs that are deployed in a Connectivity subscription.
  - The Log Analytics workspace, via Azure Monitor Private Link Scope, deployed in a Connectivity subscription.

## Identity and access management

Enterprises typically use a least-privileged approach for operational access. They use Microsoft Entra ID, [Azure role-based access control \(RBAC\)](#), and custom role definitions for access management.

- Implement fine-grained access controls in Azure Synapse by using Azure roles, Azure Synapse roles, SQL roles, and Git permissions. For more information about Azure Synapse workspace access control, see [this overview](#).
- [Azure Synapse roles](#) provide sets of permissions that you can apply at different scopes. This granularity makes it easy to grant appropriate access to administrators, developers, security personnel, and operators to compute resources and data.
- You can simplify access control by using security groups that are aligned with job roles. To manage access, you just need to add and remove users from appropriate security groups.
- You can provide security for communication between Azure Synapse and other Azure services, like Data Lake Storage and Key Vault, by using user-assigned managed identities. Doing so eliminates the need to manage credentials. Managed identities provide an identity that applications can use when they connect to resources that support Microsoft Entra authentication.

## Application automation and DevOps

- Continuous integration and delivery for an Azure Synapse workspace is achieved via Git integration and promotion of all entities from one environment (development, test, production) to another environment.
- Implement automation with Bicep / Azure Resource Manager templates to create or update workspace resources (pools and workspace). Migrate artifacts like SQL scripts and notebooks, Spark job definitions, pipelines, datasets, and other artifacts by using Synapse Workspace Deployment tools in Azure DevOps or on GitHub, as described in [Continuous integration and delivery for an Azure Synapse Analytics workspace](#).

## Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, a set of guiding tenets that you can use to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

## Reliability

Reliability ensures that your application can meet the commitments you make to your customers. For more information, see [Overview of the reliability pillar](#).

- Azure Synapse, Data Lake Storage, and Key Vault are managed platform as a service (PaaS) services that have built-in high availability and resiliency. You can use redundant nodes to make the self-hosted integration runtime and NAT VMs in the architecture highly available.
- For service-level agreement (SLA) information, see [SLA for Azure Synapse Analytics](#).
- For business continuity and disaster recovery recommendations for Azure Synapse, see [Database-restore points for Azure Synapse Analytics](#).

## Security

Security provides assurances against deliberate attacks and the abuse of your valuable data and systems. For more information, see [Overview of the security pillar](#).

- [This security baseline](#) applies guidance from Azure Security Benchmark 2.0 to Azure Synapse dedicated SQL pools.
- For information about Azure Policy security controls for Azure Synapse, see [Azure Policy Regulatory Compliance controls for Azure Synapse Analytics](#).
- For important built-in policies for Azure Synapse workspace, see [Azure Policy built-in definitions for Azure Synapse Analytics](#).

## Cost optimization

Cost optimization is about reducing unnecessary expenses and improving operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

- The analytics resources are measured in Data Warehouse Units (DWUs), which track CPU, memory, and IO. We recommend that you start with small DWUs and measure performance for resource-intensive operations, like heavy data loading or transformation. Doing so can help you determine how many units you need to optimize your workload.
- Save money with pay-as-you-go prices by using pre-purchased Azure Synapse Commit Units (SCUs).
- To explore pricing options and estimate the cost of implementing Azure Synapse, see [Azure Synapse Analytics pricing](#).
- [This pricing estimate](#) contains the costs for deploying services by using the automation steps described in the next section.

## Deploy this scenario

**Prerequisites:** You must have an Azure account. If you don't have an Azure subscription, create a [free account](#) before you start.

All code for this scenario is available in the [Synapse Enterprise Codebase repository](#) on GitHub.

The automated deployment uses Bicep templates to deploy the following components:

- A resource group
- A virtual network and subnets
- Storage tiers (Bronze, Silver, and Gold) with private endpoints
- An Azure Synapse workspace with a Managed Virtual Network
- Private Link service and endpoints
- Load balancer and NAT VMs
- A self-hosted integration runtime resource

A PowerShell script for orchestrating the deployment is available in the repository. You can run the PowerShell script or use the *pipeline.yml* file to deploy it as a pipeline in Azure DevOps.

For more information about the Bicep templates, deployment steps, and assumptions, see the [readme](#) file.

## Contributors

*This article is maintained by Microsoft. It was originally written by the following contributors.*

Principal authors:

- [Vidya Narasimhan](#) | Principal Cloud Solution Architect
- [Sabyasachi Samaddar](#) | Senior Cloud Solution Architect

Other contributor:

- [Mick Alberts](#) | Technical Writer

*To see non-public LinkedIn profiles, sign in to LinkedIn.*

## Next steps

- For information on creating an end-to-end data and analytics platform, see [Cloud-scale analytics](#) guidance.

- Explore [data mesh](#) as an architectural pattern for implementing enterprise data platforms in large, complex organizations.
- See the [Azure Synapse security white paper](#).

For more information on the services described in this article, see these resources:

- [Azure Synapse Analytics](#)
- [Azure Private Link](#)
- [Azure Data Lake Storage](#)
- [Azure Key Vault](#)

## Related resources

- [Analytics end-to-end with Azure Synapse](#)
- [Modern analytics architecture with Azure Databricks](#)

---

## Feedback

Was this page helpful?



Yes



No

# Lamna Healthcare scenario for cloud-scale analytics in Azure

Article • 12/10/2024

This reference architecture is written for customers that want to use cloud-scale analytics not only for scalability but to secure their data. It demonstrates how access to sensitive data can be controlled and how appropriately desensitized data can be shared with analysts.

## Customer profile

Lamna Healthcare (Lamna) offers patient management services to healthcare providers. They handle highly sensitive patient data throughout the course of their business. Access to the detailed data must be carefully restricted. However, Lamna would also like to safely use some version of this data to inform its business practices. They need a mechanism to share the data with analysts that doesn't violate patient trust or data protection laws.

## Current situation

Today, Lamna stores all of its data on-premises. The patient data is stored in a traditional database system. However, as their business has grown and the volume of data has increased, the company must migrate their patient applications to the cloud. As part of this transition, they would like to copy the data from the application into a cloud-based analytics platform that will allow their analysts to make better use of the data without putting extra load on the application database.

A critical concern for Lamna is the security of the patient data. As a healthcare company, they're subject to several different data protection laws.

## Architectural solution

Lamna will implement cloud-scale analytics as their solution for a cloud-based analytics platform. They rely on multiple landing zones both for increased scalability and for clear separation of sensitive data products.

## Data management landing zone

A critical concept for every cloud-scale analytics implementation is having one data management landing zone. This subscription contains resources that will be shared across all of the landing zones. This includes shared networking components, like a firewall and private DNS zones. It also includes resources for data and cloud governance. Microsoft Purview and Databricks Unity Catalog have been deployed as services at tenant level.

## Patient data landing zone

In Lamna's organizational chart, the patient management group is part of the operations group. However, given the extreme sensitivity of the data they use, they have their own data landing zone in cloud-scale analytics architecture.

This landing zone hosts a copy of the detailed patient data and health records from the company's patient management application and related data products. These data products are loaded into the landing zone by Data applications that will regularly ingest the data into the cloud and land it in Azure Data Lake Storage.

## Operations data landing zone

The operations group at Lamna is responsible for the company's core line of business, namely providing consulting services to healthcare providers. In their operations data landing zone, they store data related to these healthcare providers and the services with which they engage.

Like all business data, there's an element of sensitivity to these data products, and Lamna wants to protect its list of clients. However, since this data doesn't include health information about individuals, it's not subject to the most stringent data protection laws.

## Data applications

The operations landing zone has a [data application](#) that loads the healthcare provider data from Lamna's on-premises operations system. Like all data applications, this lands the data in the cloud as-is and doesn't apply transformations to the data products.

## Data products

Analysts throughout Lamna need access to data to build reports for the business. However, much of the data is far too sensitive for a broad audience. To safely provide access to the highly sensitive patient data, the operations team created a [Tokenized](#)

**patients dataset** product in their landing zone. Using Azure Data Factory, they copy patient data from the patients landing zone. However, the team was careful to remove or tokenize any columns containing personal data. This step allows analysts to use the data for business purposes without exposing any personal details of the patients.

## Marketing data landing zone

The marketing group is focused on obtaining new clients and managing Lamna's position within the marketplace. Their marketing landing zone is primarily used to store and analyze external data products about the markets they serve and the healthcare industry.

However, to support a new marketing push, the group wants to conduct a study of health outcomes for the patients served by Lamna's clients. They hope to produce a fact-based report supported by strong statistical evidence showing that their approach to healthcare leads to better outcomes.

To support this new effort, researchers in the marketing group need to access the highly sensitive patient data in a secure and compliant manner while still being able to obtain the information that they need.

To meet this need, the marketing team creates aggregated data products from the tokenized patients dataset created by the operations team. These data products don't contain individual health records. Instead, they group records across different axes. This helps researchers to conduct studies of the population as a whole without risking access to any individual's health information.

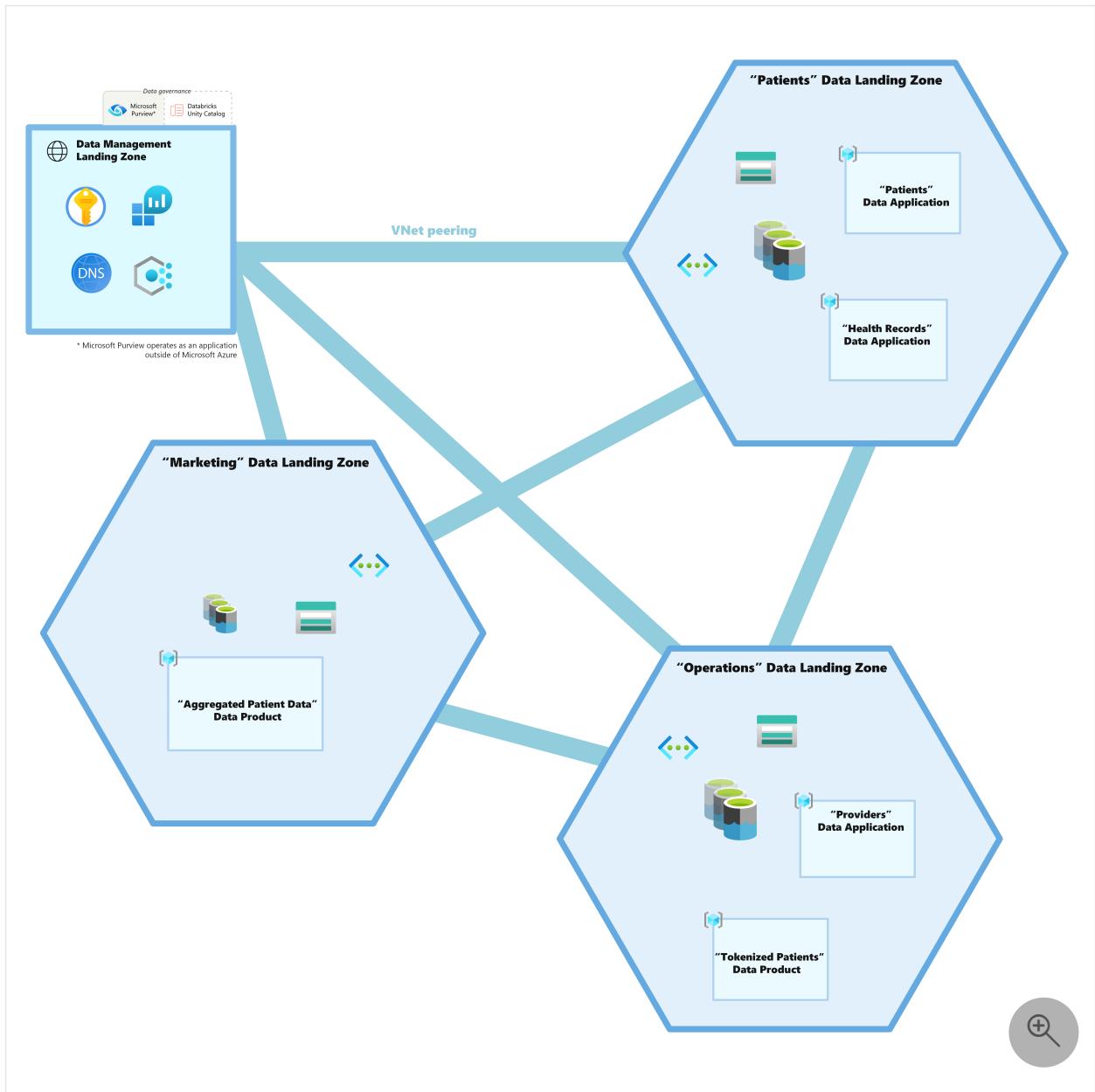


Figure 1: Diagram of Lamna architecture. Not all Azure services are represented in the diagram. It's simplified to highlight the core concepts of how resources are organized within the architecture.

## Rationale

### Should all sensitive data always be given its own data landing zone?

No. Only the most restricted data requiring specific protections, like just-in-time access or customer-managed keys, requires its own landing zone. For other scenarios, other data protection features in Azure provide a highly secure environment for your data. This includes row-level security, column-level security, and encrypted columns.

# Next steps

- Continue to [Deployment templates for cloud-scale analytics](#).
  - Learn more in [Understand data privacy for cloud-scale analytics in Azure](#).
- 

## Feedback

Was this page helpful?

 Yes

 No

# A Financial Institution Scenario for Data Mesh

Article • 11/27/2024

This scenario is for customers who want to use cloud-scale analytics for scalability and *data mesh* architectures. It demonstrates a complex scenario with landing zones, data integrations, and data products.

## Customer Profile

A fictitious enterprise, Woodgrove Bank, is a large financial services company with a worldwide footprint. Woodgrove Bank's data is housed in on-premises and cloud deployment systems. Within the Woodgrove Bank architecture, there are several data warehouse systems for consolidated marketing and integrated reporting. This architecture includes several data lakes for unplanned analytics and data discovery. Woodgrove Bank applications are interconnected via application integration patterns, which are mostly API-based or event-based.

## The Current Situation

It's challenging for Woodgrove Bank to distribute data to different locations because of the complexity of data warehousing. Integrating new data is time-consuming, and it's tempting to duplicate data. Woodgrove Bank finds it difficult to oversee the end-to-end data landscape because of point-to-point connectivity. The bank underestimated the demand for intensive data consumption. New use cases are introduced quickly, one after another. Data governance, such as data ownership and quality, and costs are hard to control. Keeping current with regulations is difficult because Woodgrove Bank doesn't know exactly where its data resides.

## Architecture Solution: Data Mesh

Over the past several years, organizations recognize that data is at the heart of everything. Data opens new efficiencies, drives innovation, unlocks new business models, and increases customer satisfaction. It's a top priority for companies to use data-driven methods, like data at scale.

Reaching a stage where the deeper value of data is accessible to all organization members is challenging. Legacy and tightly interconnected systems, centralized

monolithic platforms, and complex governance can be significant barriers to generating value out of data.

## About Data Mesh

The concept of data mesh, a term coined by Zhamak Dehghani, encompasses data, technology, processes, and organization. Conceptually, it's an accessible approach to managing data where various domains use their own data. Data mesh challenges the idea of conventional centralization of data. Rather than looking at data as one huge repository, data mesh considers the decomposition of independent data products. This shift, from centralized to federated ownership, is supported by a modern, self-service data platform typically designed using cloud-native technologies.

When you break down the data mesh concept into building blocks, here are some key points to consider:

- **Data as a Product:** Each (organizational) domain operates its data end to end. Accountability lies with the data owner within the domain. Pipelines become a first-class concern of the domains themselves.
- **Federated Computational Data Governance:** To ensure that each data owner can trust the others and share its data products, an enterprise data governance body must be established. The governance body implements data quality, central visibility of data ownership, data access management, and data privacy policies.
- **Domain-Oriented Data Ownership:** The enterprise should ideally define and model each data-domain node within the mesh by applying the principles of domain-oriented design.
- **Self-Serve Data Platform:** A data mesh requires a self-serve data platform that allows users to remove the technical complexity and focus on their individual data use cases.

## Cloud-Scale Analytics

Data-as-a-product thinking and a self-service platform model aren't new to Microsoft. Microsoft observed best practices of distributed platforms, pipelines across domains, federated ownership, and self-explanatory data for many years.

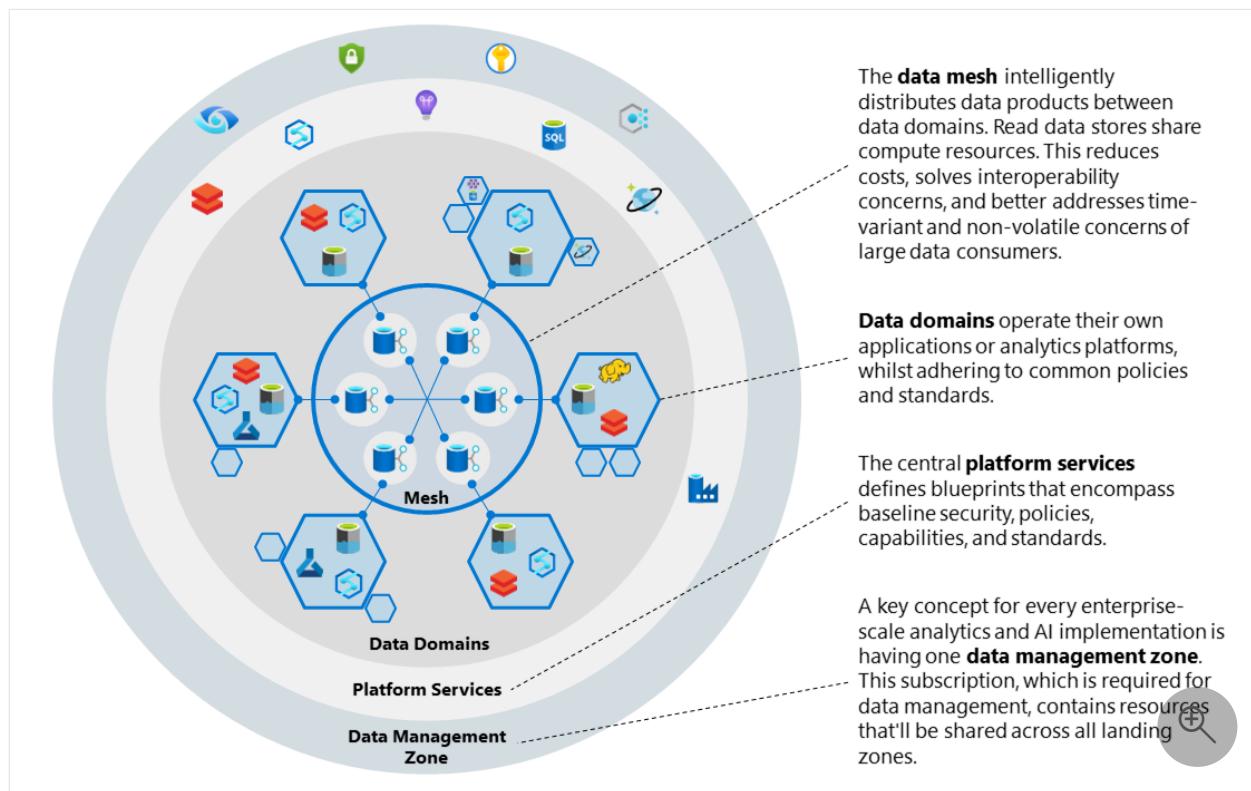
Woodgrove Bank can transition to data mesh by using cloud-scale analytics. Cloud-scale analytics is an open-source and prescriptive blueprint for designing and quickly deploying modern data platforms. It's coupled with Azure best practices and design principles and is aligned with the Azure Well-Architected Framework. Cloud-scale

analytics gives enterprises an 80 percent prescribed viewpoint, and the remaining 20 percent is customizable.

Cloud-scale analytics offers enterprises a strategic design path toward data mesh, and it can be used to quickly set up such an architecture. It offers a blueprint, including core data platform services for data management.

At the highest level, cloud-scale analytics uses a data management capability, which is enabled through the data management landing zone. This zone is responsible for the federated data governance of an organization of the (self-service) platform, and the data domains that drive business value through data products. The benefit of this approach is that it removes technical complexity while adhering to the same standards. It ensures that there's no proliferation of technology. It also allows enterprises to start modular, with a small footprint, and then grow over time.

The data management landing zone, as you can see in the following diagram, surrounds all data domains. It glues all domains together and provides the oversight that Woodgrove Bank is looking for.

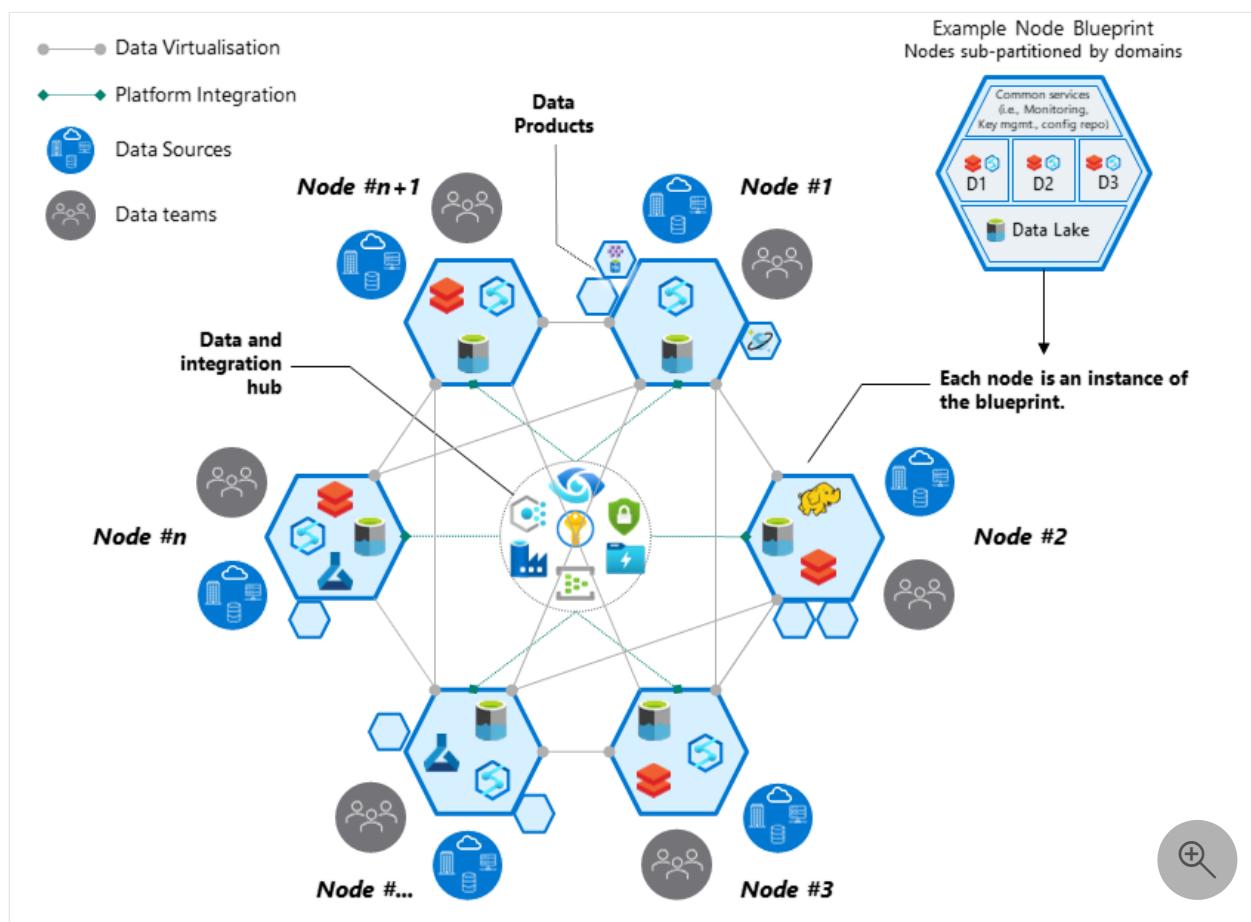


Cloud-scale analytics also advocates the application of consistent governance that uses a common architecture when data products are distributed. The framework allows direct communication between domains. It stays in control by placing an emphasis on central cataloging and classification to protect data and allow groups to discover data. It places an umbrella on top of your data estate.

# Data Domains

When you use cloud-scale analytics as a strategic path, you need to think of the decomposition of your architecture and the resulting granularity. Data mesh decomposes data by not following the borders of technologies. Instead, it applies the principles of domain-driven design (DDD), an approach to software development that involves complex systems for larger organizations. DDD is popular because of its effect on modern software and application development practices, such as microservices.

One of the patterns from domain-driven design is known as bounded context. Bounded contexts set the logical boundaries of a domain's solution space to better manage complexity. It's important that teams understand which aspects, including data, they can change and which are shared dependencies that require coordination with others. Data mesh embraces bounded context. It uses this pattern to describe how organizations can coordinate around data domains and focus on delivering data as a product. Each data domain owns and operates multiple data products with its own technology stack, which is independent of the others.



## Data Products

When you zoom in on the inner architecture of such a data domain, you expect to find data products within it.

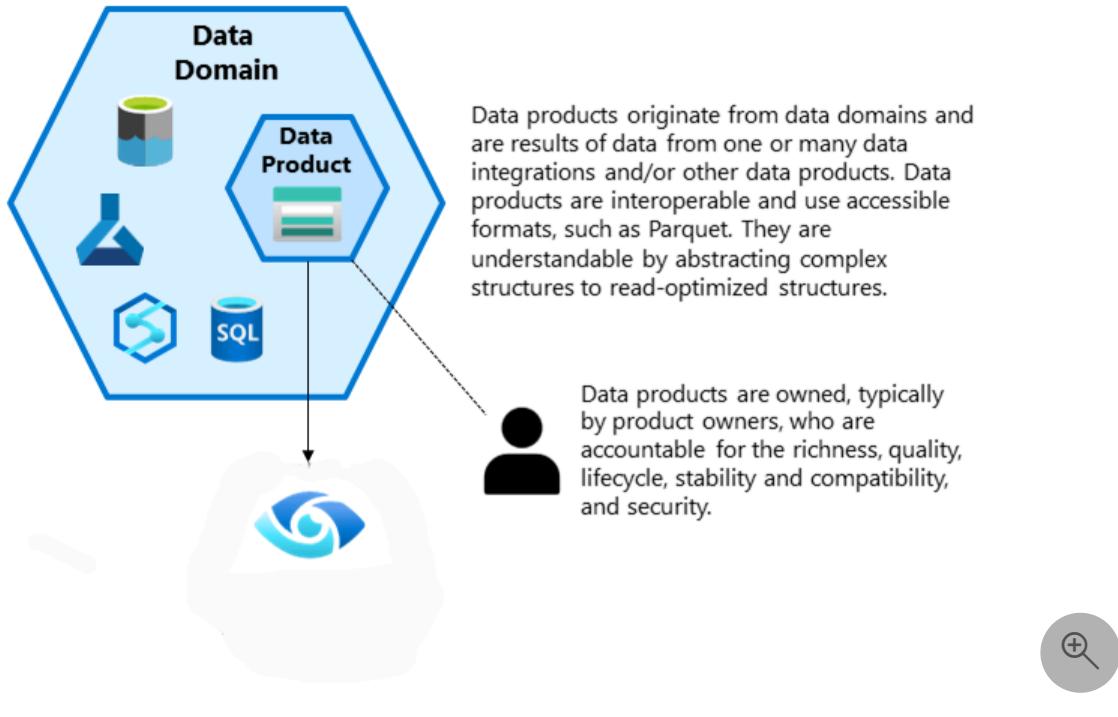
Data products fulfill a specific need within businesses that use data. Data products manage, organize, and make sense of the data across domains and then present the insights they gained. A data product results from data from one or many data integrations or other data products. Data products are closely aligned with data domains and inherit the same constructed, formalized language agreed upon by stakeholders and designers. Each domain that generates data is responsible for making these data products available to the other domains.

To help quickly deliver data products, cloud-scale analytics offers templates for data distribution and integration patterns. The framework provides data batch, streaming, and analytics to address the needs of diverse consumers.

One great thing about cloud-scale analytics is how domains and data products are organized. Each data domain aligns with one data landing zone, which is a logical construct and a unit of scale in cloud-scale analytics architecture. It enables data retention and execution of data workloads, which generates insights and value. Each data product aligns with one resource group within the data landing zone, and all data landing zones and management zones align with subscriptions. This approach eases implementation and management.

All cloud-scale analytics templates inherit the same set of policies from the data management landing zone. The templates automatically deliver necessary metadata for data discoverability, governance, security, cost management, and operational excellence. You can quickly onboard new data domains without the need for complex onboarding, integrating, and testing.

The following diagram illustrates what a data product might look like:

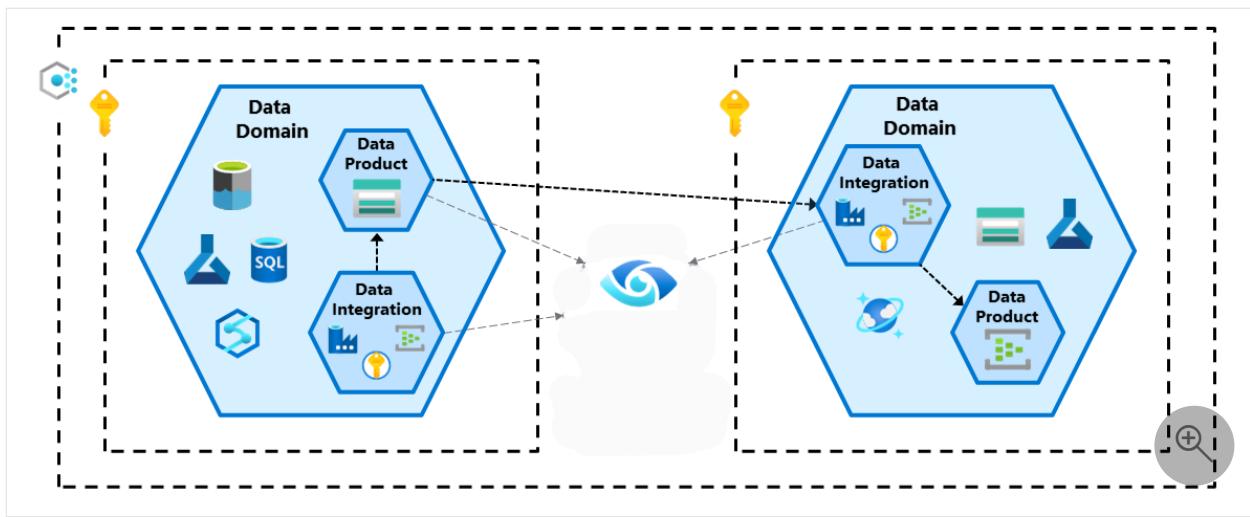


A pragmatic approach to building data products is to either align with the source, where the data originates, or with the consuming use case. In both cases, you need to provide an abstract view of the underlying (complex) application data model. You must try to hide the technical details and optimize for intensive data consumption. An Azure Synapse view or Parquet file, which logically groups data together, is an example of how a data product can be shared across various data domains.

Next, you need to work on data discoverability, provenance, usage, and lineage. A proven approach is to use a data governance service, like Microsoft Purview, to register all data. Data integration in cloud-scale analytics perfectly connects the dots because it allows building these data products as it simultaneously performs metadata registration.

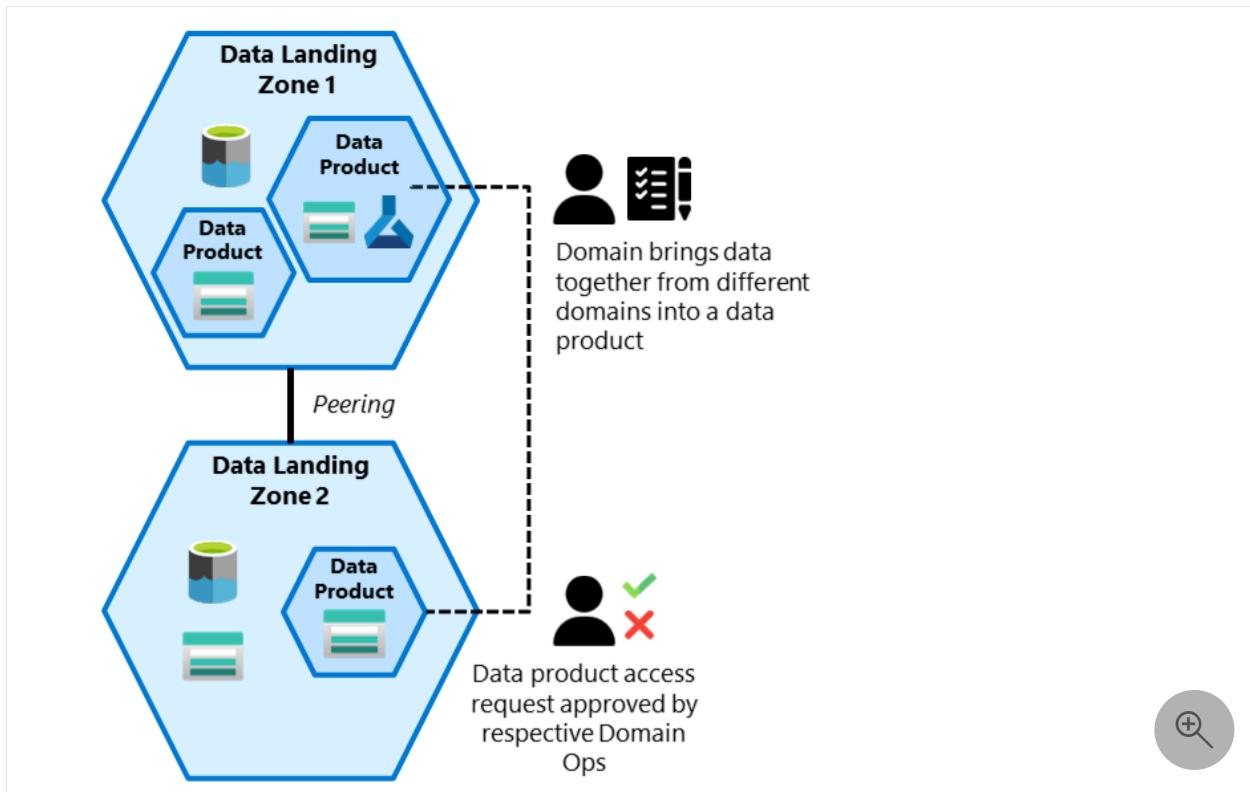
By aligning data domains and Microsoft Purview collections, you automatically capture all data origin, lineage, data quality details, and consumption information from the individual domains. With this approach, you can connect multiple data domains and products to a centralized governance solution, which stores all the metadata from each environment. The benefit is that it centrally integrates all the metadata and makes it easily accessible to various consumers. You can extend this architecture to register new data products.

The following diagram illustrates a cross-domain data mesh architecture that uses cloud-scale analytics.



The network design allows data products to be shared across domains by using minimal cost and eliminating a single point of failure and bandwidth limitations. To help ensure security, you can use the Microsoft *Zero Trust* security model. Cloud-scale analytics proposes the use of network isolation through private endpoints and private network communication, an identity-driven data access model that uses MIs, UMIs, and nested security groups, following the *principle of least privilege*.

You can use managed identities to ensure that a least privilege access model is followed. Applications and services in this model have limited access to data products. Azure policies, with the upcoming data policies, are used to enable self-service and enforce compliant resources within all data products, at scale. With this design, you can have uniform data access while staying fully in control via centralized data governance and auditing.



## Evolve Toward the Future

Cloud-scale analytics is designed with data mesh in mind. Cloud-scale analytics provides a proven approach by which organizations can share data across many data domains. This framework allows domains to have the autonomy to make choices and it governs the architecture by ring-fencing it with data management services.

When you're implementing data mesh, logically group and organize your domains. This approach requires an enterprise view and is likely a cultural shift for your organization. The shift requires you to federate data ownership among data domains and owners who are accountable for providing their data as products. It also requires teams to conform to centralized capabilities offered by the data management landing zone. This new approach might require individual teams to give up their current mandates, which are likely to generate resistance. You might have to make certain political choices and strike a balance between centralized and decentralized approaches.

You can scale a data mesh architecture by adding more landing zones to the architecture for individual domains. These landing zones use virtual network peering to connect to the data management landing zone and all other landing zones. This pattern allows you to share data products and resources across zones. When you split into separate zones, you can spread workloads across Azure subscriptions and resources. This approach allows you to implement the data mesh organically.

## Learn More

Microsoft resources:

- [Data management landing zone template ↗](#)
- [Data landing zone template ↗](#)

Article by data mesh founder Zhamak Dehghani:

- [How to move beyond a monolithic data lake to a distributed data mesh ↗](#)

---

## Feedback

Was this page helpful?



# Connect to environments privately

Article • 12/10/2024

The reference architecture is secure by design. It uses a multilayered security approach to mitigate common data exfiltration risks raised by customers. You can use certain features on a network, identity, data, and service layer to define specific access controls and expose only required data to your users. Even if some of these security mechanisms fail, the features help keep data within the enterprise-scale platform secure.

Network features such as private endpoints and disabled public network access can greatly reduce the attack surface of a data platform within an organization. Even with these features enabled, you need to take extra precautions to successfully connect to services such as Azure storage accounts, Azure Synapse workspaces, or Azure Machine Learning from the public internet.

This document describes the most common options for connecting to services inside a data management landing zone or data landing zone in a simple and secure way.

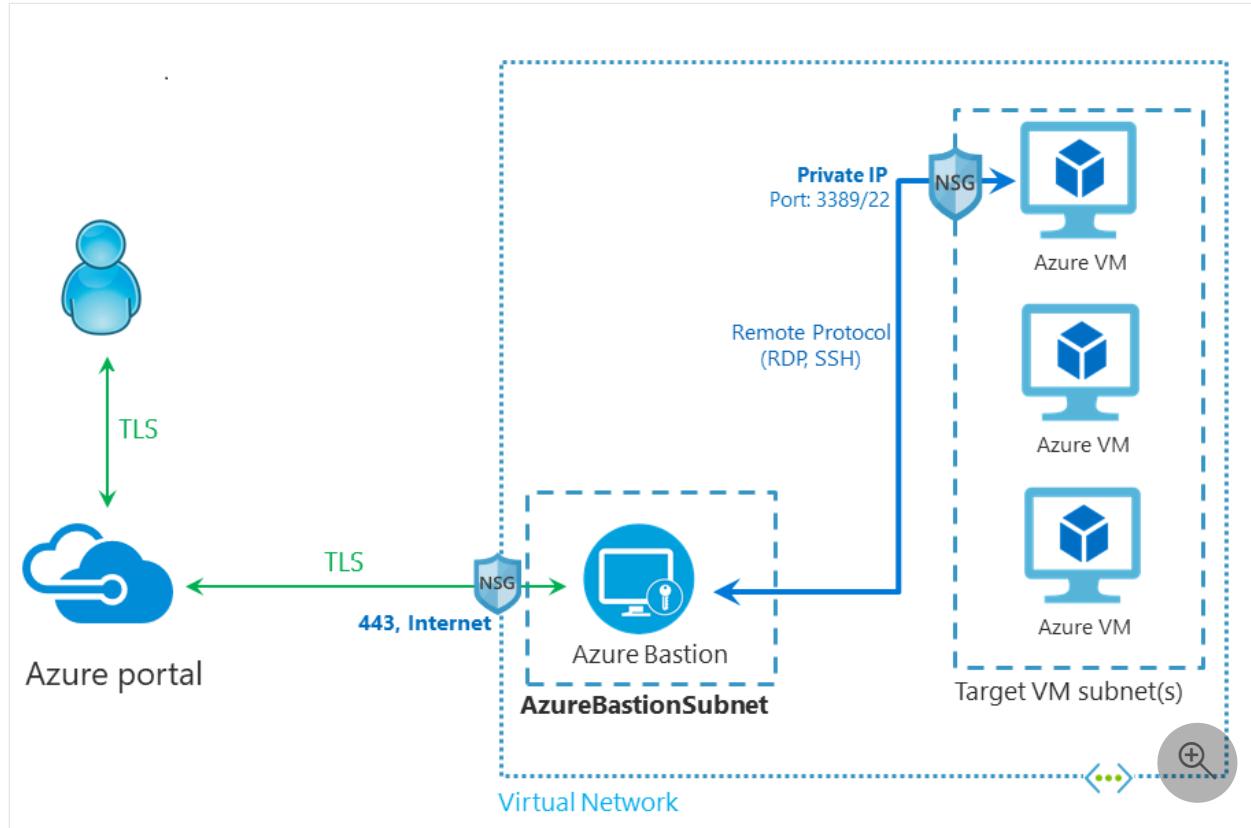
## Overview of Azure Bastion host and jumpboxes

The simplest solution is to host a jumpbox on the virtual network of the data management landing zone or data landing zone to connect to the data services through private endpoints. A jumpbox is an Azure virtual machine (VM) running Linux or Windows to which users can connect via the Remote Desktop Protocol (RDP) or Secure Shell (SSH).

Previously, jumpbox VMs had to be hosted with public IPs to enable RDP and SSH sessions from the public internet. Network security groups (NSGs) could be used to further lock down traffic to allow connections from only a limited set of public IPs. However, this approach meant that a public IP had to be exposed from the Azure environment, which increased the attack surface of an organization. Alternatively, customers could use DNAT rules in their Azure Firewall to expose the SSH or RDP port of a VM to the public internet, which leads to similar security risks.

Today, instead of exposing a VM publicly, you can rely on Azure Bastion as a more secure alternative. Azure Bastion provides a secure remote connection from the Azure portal to Azure VMs over Transport Layer Security (TLS). Azure Bastion should be set up on a dedicated subnet (subnet with the name `AzureBastionSubnet`) in the Azure data landing zone or Azure data management landing zone. You can then use it to connect to any VM on that virtual network or a peered virtual network directly from the Azure

portal. No extra clients or agents need to be installed on any VM. You can again use NSGs to allow RDP and SSH from Azure Bastion only.



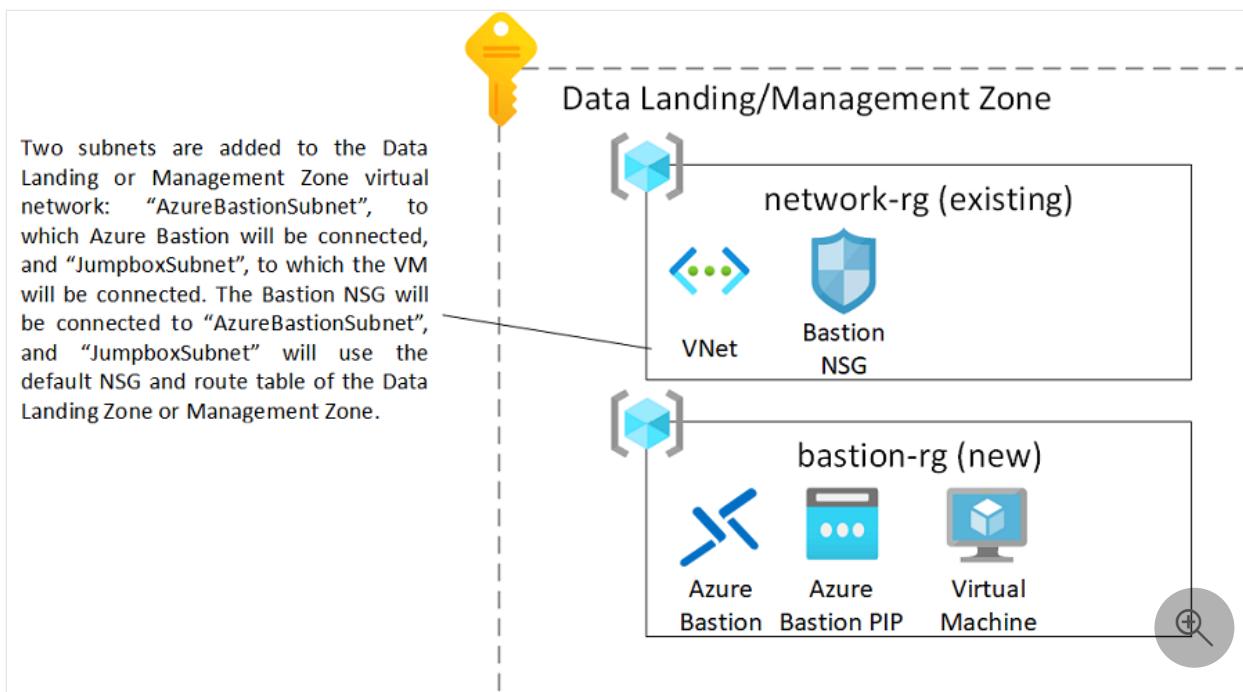
Azure Bastion provides a few other core security benefits, including:

- Traffic initiated from Azure Bastion to the target VM stays within the customer virtual network.
- You get protection against port scanning because RDP ports, SSH ports, and public IP addresses aren't publicly exposed for VMs.
- Azure Bastion helps protect against zero-day exploits. It sits at the perimeter of your virtual network. Because it's a platform as a service (PaaS), the Azure platform keeps Azure Bastion up to date.
- The service integrates with native security appliances for an Azure virtual network, such as Azure Firewall.
- Azure Bastion can be used to monitor and manage remote connections.

For more information, see [What is Azure Bastion?](#).

## Deployment

To simplify the process for users, there's a Bicep/ARM Template that can help you quickly create this setup inside your data management landing zone or data landing zone. Use the template to create the following setup inside your subscription:



To deploy the Bastion host yourself, select the **Deploy to Azure** button:



When you deploy Azure Bastion and a jumpbox through the **Deploy to Azure** button, you can provide the same prefix and environment that you use in your data landing zone or data management landing zone. This deployment has no conflicts, and it acts as an add-on to your data landing zone or data management landing zone. You can manually add other VMs to allow more users to work inside the environment.

## Connect to the VM

After the deployment, you'll notice that two extra subnets are created on the data landing zone virtual network.

Name ↑↓	IPv4 ↑↓	IPv6 ↑↓	Available IPs ↑↓	Delegated to ↑↓	Security group ↑↓	Route table ↑↓	...
AzureBastionSubnet	10.255.1.0/24	-	250	-	dlz01-dev-bastion-nsg	-	
JumpboxSubnet	10.255.2.0/24	-	250	-	dlz01-dev-nsg	dlz01-dev-routetable	...

In addition, you'll find a new resource group inside your subscription, which includes the Azure Bastion resource and a virtual machine:

The screenshot shows the Azure portal interface for the resource group 'dlz01-dev-bastion'. On the left, there's a navigation sidebar with options like Overview, Activity log, Access control (IAM), Tags, Resource visualizer, Events, Settings (Deployments, Security, Policies), and a search bar. The main area is titled 'Essentials' and lists six resources: 'dlz01-dev-bastion001' (Bastion), 'dlz01-dev-bastion001-pip' (Public IP address), 'dlz01-dev-vm001' (Virtual machine), 'dlz01-dev-vm001-disk' (Disk), and 'dlz01-dev-vm001-nic' (Network interface). There are filters at the top for 'Type == all' and 'Location == all', and a button to 'Add filter'.

To connect to the VM by using Azure Bastion, follow these steps:

1. Select the VM (for example, *dlz01-dev-bastion*), select **Connect**, and then select **Bastion**.

The screenshot shows the Azure portal interface for the virtual machine 'dlz01-dev-vm001'. The 'Connect' blade is open, displaying three options: RDP, SSH, and Bastion. The 'Bastion' option is highlighted. The main area shows the 'Overview' blade for the VM.

2. Select the blue **Use Bastion** button.
3. Enter your credentials, and then select **Connect**.

The screenshot shows the Azure portal interface for the virtual machine 'dlz01-dev-vm001' in the 'Bastion' blade. It displays a message about Azure Bastion enabling secure RDP & SSH access. It shows the provisioning state as 'Succeeded'. It includes fields for 'Username' (set to 'VnMainUser') and 'Password' (redacted). There's a checkbox for 'Open in new window' and a 'Connect' button at the bottom. A magnifying glass icon with a plus sign is visible in the bottom right corner.

The RDP session opens on a new browser tab, from which you can start connecting to your data services.

4. Sign in to the [Azure portal](#).

5. Go to the `{prefix}-{environment}-product-synapse001` Azure Synapse workspace inside the `{prefix}-{environment}-shared-product` resource group for data exploration.

The screenshot shows the Azure Synapse Analytics workspace interface. On the left, there's a navigation sidebar with various sections like Locks, Analytics pools, Security, Monitoring, Automation, and Support + troubleshooting. The main area is titled 'Analytics pools' and shows a table with columns 'Name', 'Type', and 'Size'. Three entries are listed: 'Built-in' (Serverless, Auto, DW100c), 'sqlPool001' (Dedicated), and 'bigDataPool001' (Apache Spark pool, Small). A search bar with placeholder text 'Search to filter items...' is at the top, and a magnifying glass icon is on the right.

6. In the Azure Synapse workspace, load a sample dataset from the gallery (for example, the NYC Taxi dataset), and then select New SQL Script to query TOP 100 rows.

The screenshot shows the Microsoft Azure Synapse Analytics workspace. The left sidebar shows 'Data' selected, with a tree view of resources including 'Workspace' and 'Linked'. Under 'Linked', 'Azure Blob Storage' has a child 'Sample Datasets' with 'nyc\_tlc\_yellow'. The main area shows a SQL script editor with the following code:

```

1 -- This is auto-generated code
2 SELECT
3 | TOP 100 *
4 FROM
5 OPENROWSET(
6 BULK 'https://azureopendatastorage.blob.core.windows.net/nyctlc/yellow/puYear=*/*Month=*.parquet',
7 FORMAT = 'parquet'
8 )
  
```

A context menu is open over the script, listing options: 'New SQL script...', 'New notebook...', 'Edit', 'Delete', and 'Properties'. Below the editor, the 'Results' tab is active, showing a table view of the query results. The table has columns: vendorID, tpepPickupDate, tpepDropoffDate, passengerCount, tripDistance, puLocationId, doLocationId, and startLon. The first few rows show data for different trips. A circular callout highlights the '(NULL)' value in the 'startLon' column of the last row of the results table.

If all the virtual networks are peered with each other, only a single jumpbox in one data landing zone is required to access services across all data landing zones and data management landing zones.

To learn why we recommend this network setup, see [Network architecture considerations](#). We recommend a maximum of one Azure Bastion service per data landing zone. If more users require access to the environment, you can add extra Azure VMs to the data landing zone.

## Use point-to-site connections

Alternatively, you can connect users to the virtual network by using point-to-site connections. An Azure-native solution to this approach is to set up a VPN gateway to allow VPN connections between users and the VPN gateway over an encrypted tunnel. After you establish the connection, users can start connecting privately to services that are hosted on the virtual network inside the Azure tenant.

We recommend that you set up the VPN gateway in the hub virtual network of the hub-and-spoke architecture. For detailed, step-by-step guidance on setting up a VPN gateway, see [Tutorial: Create a gateway portal](#).

## Use site-to-site connections

If users are already connected to the on-premises network environment and connectivity should be extended to Azure, you can use site-to-site connections to connect the on-premises and Azure connectivity hub. Like a VPN tunnel connection, the site-to-site connection lets you extend the connectivity to the Azure environment. Doing so allows users who are connected to the corporate network to connect privately to services that are hosted on the virtual network inside the Azure tenant.

The recommended, Azure-native approach to such connectivity is the use of ExpressRoute. We recommend that you set up an ExpressRoute gateway in the hub virtual network of the hub-and-spoke architecture. For detailed, step-by-step guidance on setting up ExpressRoute connectivity, see [Tutorial: Create and modify peering for an ExpressRoute circuit by using the Azure portal](#).

## Next steps

[Enterprise-scale FAQ](#)

---

# Feedback

Was this page helpful?

 Yes

 No

# Frequently asked questions about cloud-scale analytics

Article • 04/22/2022

The following are common questions asked about cloud-scale analytics.

## Storage accounts

### Why do I need three separate storage accounts? Can't I just have one with three containers for each layer (raw, refined, and curated)?

Most data analytics patterns today exist with the three layers of raw, refined, and curated. Although they can be kept in the same storage, when it comes to large-scale implementations it creates issues with exceeding the number of allowed role-based access control (RBAC) and access control list (ACL) permissions that are available within a single storage account. When you use separate storage accounts, most implementations can avoid this issue.

Other reasons are discussed in [Overview of Azure Data Lake Storage for cloud-scale analytics](#).

## Databricks

### Should we deploy an Azure Databricks workspace per product?

The recommendation is to use the shared product [Azure Databricks analytics and data science workspace](#) inside the landing zone.

This decision has been made to reduce the management overhead for the data platform operations team. Azure Databricks has a set of stand-alone policies that aren't integrated into the Azure policies. In a large environment, the setup of more Azure Databricks workspaces creates more management overhead. For example, maintaining policies and supported Apache Hive versions, updating ADB versions, and enforcing external Apache Hive metastore. There's no way a central platform team can enforce certain settings within any of the Databricks workspaces. We recommend having shared

workspaces for product teams in the landing zones, where the data platform ops teams can then define the necessary cluster policies and initialization scripts.

We recommend to use VNet peering between landing zones and private endpoints. For Azure Databricks, use VNet injection. As there's direct line of sight to all endpoints, there are no connectivity issues.

## Next steps

[The ingest process with cloud-scale analytics in Azure](#)

# The ingest process with cloud-scale analytics in Azure

Article • 12/10/2024

Azure provides several services to ingest and release data to native and third-party platforms. Different services can be used, depending on volume, velocity, variety, and direction. Some of these services are:

- [Azure Data Factory](#) is a service built for all data application (source-aligned) needs and skill levels. Write your own code or construct, extract, load, and transform processes within the intuitive visual environment and without code. With more than 90+ natively built and maintenance-free connectors, visually integrate data sources at no added cost. Engineers can use private endpoints and link services to securely connect to Azure platform as a service (PaaS) resources without using the PaaS resource's public endpoints. Engineers can use integration runtimes to extend pipelines to third-party environments like on-premises data sources and other clouds.

Some of these connectors support being used as a source (read) or as a sink (write). Azure native services, Oracle, SAP, and others can be used as source or sink, but not all connectors support it. In these cases, you can use generic connectors like Open Database Connectivity (ODBC), the file system, or SSH File Transfer Protocol (SFTP) connectors.

- [Azure Databricks](#) is a fast, easy, and collaborative Apache-Spark-based analytics service. For a big data pipeline, you can ingest the data (raw or structured) into Azure through Data Factory in batches or streamed in almost real time with Apache Kafka, Azure Event Hubs, or IoT Hub. This data lands in a data lake for long-term, persisted storage in Azure Data Lake Storage. Azure Databricks can read data from multiple data sources as part of the workflow.
- The Microsoft Power Platform provides [connectors to hundreds of services](#) that can be event-, schedule-, or push-driven. Microsoft Power Automate can act on events and trigger workflows optimized for single records or small data volumes.

Proprietary native and third-party tooling provides niche capabilities to integrate with specialized systems and near-real-time replication.

- [Azure Data Share](#) supports organizations to securely share data with multiple external customers and partners. After you create a data share account and add data products, customers and partners can be invited to the data share. Data

providers are always in control of the data that they've shared. Azure Data Share makes it simple to manage and monitor what data is shared, when it was shared, and who shared it.

### Important

Every data landing zone can have an [data ingestion resource group](#) that exists for businesses with an data agnostic ingestion engine. If you don't have this framework engine, the only recommended resource is deploying an Azure Databricks analytics workspace, which would be used by data integrations to run complex ingestion.

See the [data agnostic ingestion engine](#) for potential automation patterns.

## Ingest considerations for Azure Data Factory

If you have an data agnostic ingestion engine, you should deploy a single Data Factory for each data landing zone in the data ingestion resource group. The Data Factory workspace should be locked off to users, and only managed identity and service principals will have access to deploy. Data landing zone operations should have read access to allow pipeline debugging.

Data application can have there own Data Factory for data movement. Having a Data Factory in each data application resource group supports a complete continuous integration (CI) and continuous deployment (CD) experience by only allowing pipelines to be deployed from Azure DevOps or GitHub.

All Data Factory workspaces will mostly use the managed virtual network (VNet) feature in Data Factory or [self-hosted integration runtime](#) for their data landing zone within the data management landing zone. Engineers are encouraged to use the managed VNet feature to securely connect to the Azure PaaS resource.

However, it's possible to create more integration runtimes to ingest from on-premises, third-party clouds, and third-party software-as-a-service (SaaS) data sources.

## Ingest considerations for Azure Databricks

This guidance elaborates on the information within:

- [Securing access to Azure Data Lake Storage Gen2 from Azure Databricks ↗](#)
- [Azure Databricks best practices](#)

- For development, integration operations should have their own Azure Databricks environments before checking in code to be deployed to the single Azure Databricks workspace during testing and production.
- Data Factory in the data application (source-aligned) resource group should provide the framework for calling Azure Databricks jobs.
- Data applications teams can deploy short, automated jobs on Azure Databricks and expect their clusters to start quickly, execute the job, and terminate. It's recommended to set up Azure Databricks pools to reduce the time it takes for clusters to spin up for jobs.
- We recommend organizations use Azure DevOps to implement a deployment framework for new pipelines. The framework will be used to create the dataset folders, assign access control lists, and create a table with or without enforcing Databricks table access controls.

## Stream ingestion

Organizations might need to support scenarios where publishers generate high-velocity event streams. For this pattern, a message queue is recommended, for example, Event Hubs or IoT Hub, to ingest these streams.

Event Hubs and IoT Hub are scalable event processing services that can ingest and process large event volumes and data with low latency and high reliability. Event Hubs is designed as a big data streaming and event ingestion service. IoT Hub is a managed service that serves as a central message hub for bidirectional communication between an IoT application and the devices it manages. From there, data can either be exported to a data lake at regular intervals (batch) and processed with Azure Databricks in near-real-time via Apache Spark Streaming, Azure Data Explorer, Stream Analytics, or Time Series Insights.

The last Event Hubs or Apache Kafka landing zone inside the use case's specific landing zone should send its aggregated data to the data lake's raw layer in one of the data landing zones and to Event Hubs related to the data application (source-aligned) resource group in the data landing zone.

## Monitor ingestion

Out-of-the-box [Azure Data Factory pipeline monitoring](#) can be used to monitor and troubleshoot the exceptions from the Data Factory pipelines. It reduces the effort of developing a custom monitoring and reporting solution.

Built-in monitoring is one of the main reasons to use Azure Data Factory as a main orchestration tool, and Azure Policy can help to automate this setup.

## Next steps

[SAP ingestion with cloud-scale analytics in Azure](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Data agnostic ingestion engine

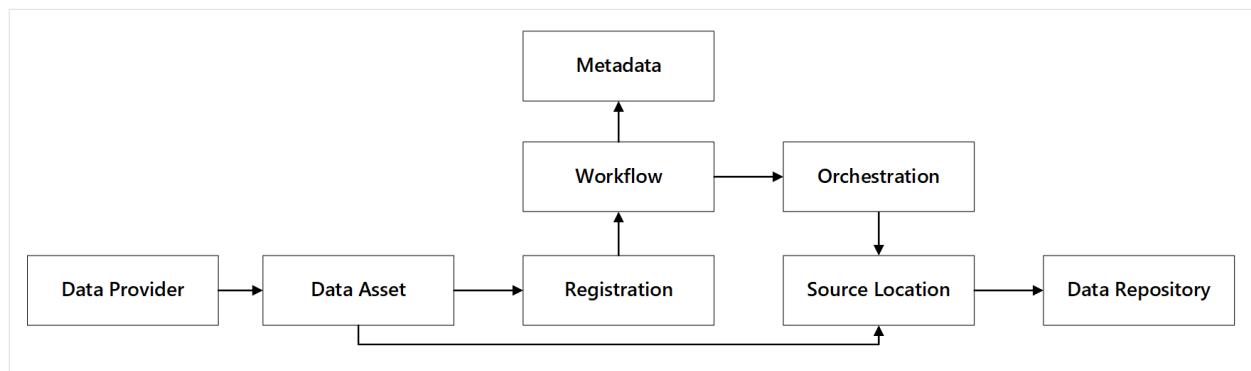
Article • 12/10/2024

This article explains how you can implement data agnostic ingestion engine scenarios using a combination of PowerApps, Azure Logic Apps, and metadata-driven copy tasks within Azure Data Factory.

Data agnostic ingestion engine scenarios are typically focused on letting non-technical (non-data-engineer) users publish data assets to a Data Lake for further processing. To implement this scenario, you must have onboarding capabilities that enable:

- Data asset registration
- Workflow provisioning and metadata capture
- Ingestion scheduling

You can see how these capabilities interact:



*Figure 1: Data registration capabilities interactions.*

The following diagram shows how to implement this process using a combination of Azure services:

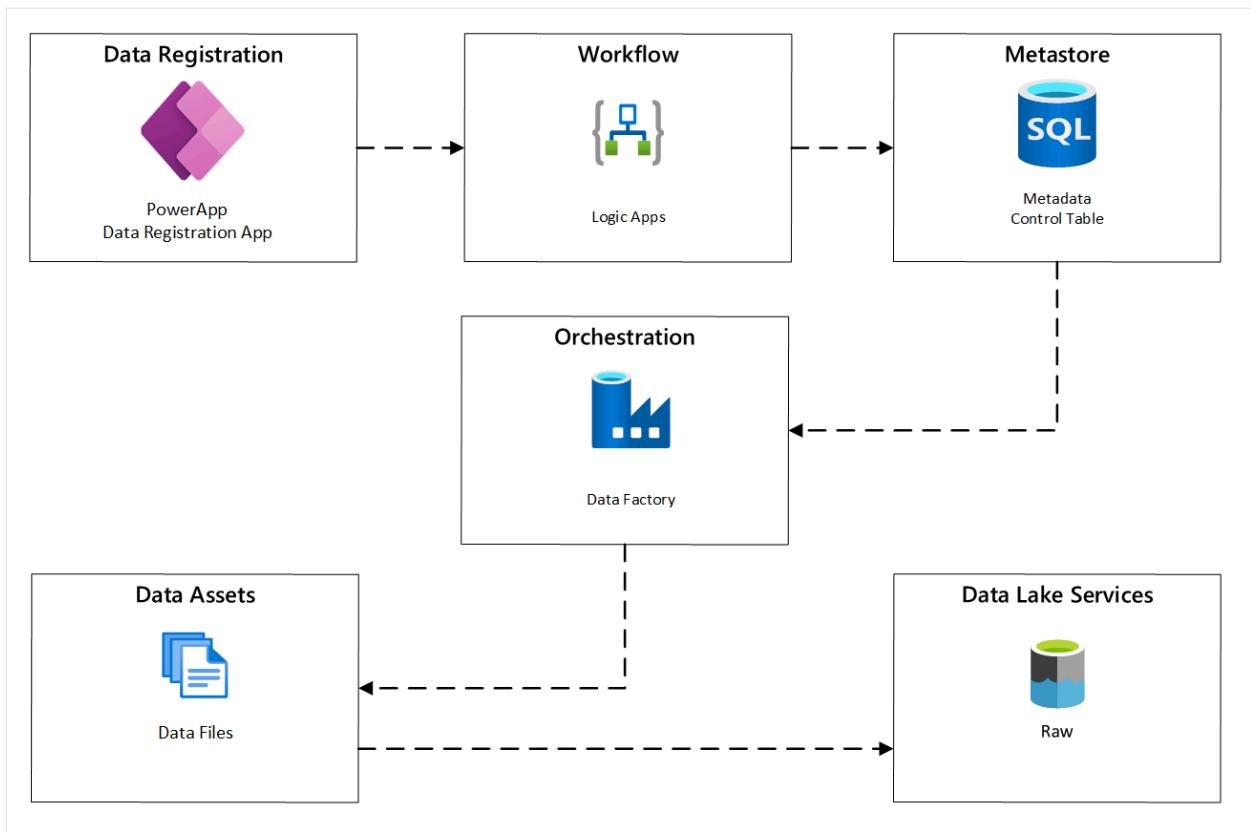


Figure 2: Automated ingestion process.

## Data asset registration

To provide the metadata used to drive automated ingestion, you need data asset registration. The information you capture contains:

- **Technical information:** Data asset name, source system, type, format and frequency.
- **Governance information:** Owner, stewards, visibility (for discovery purposes) and sensitivity.

PowerApps is used to capture metadata describing each data asset. Use a model-driven app to enter the information that gets persisted to a custom Dataverse table. When metadata is created or updated within Dataverse, it triggers an Automated Cloud flow that invokes further processing steps.

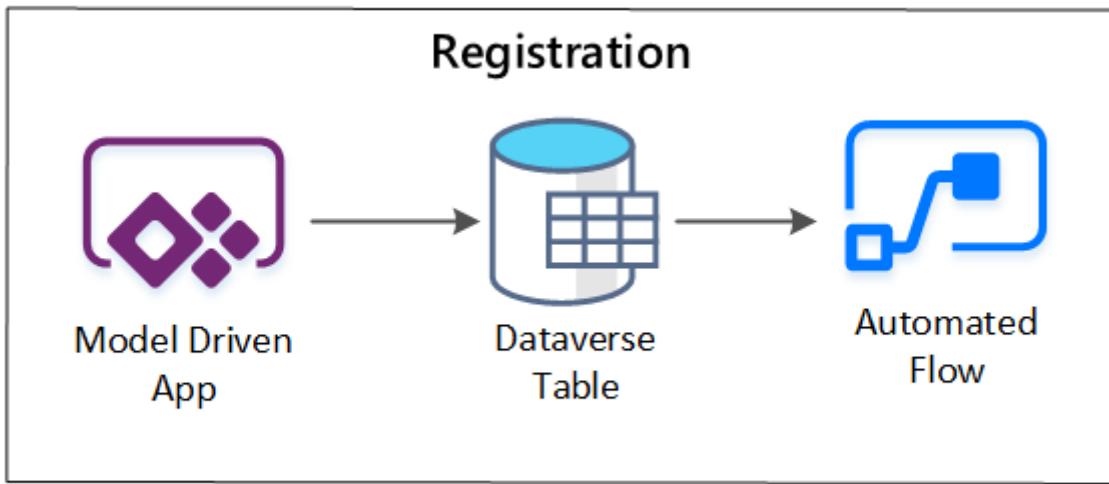


Figure 3: Data asset registration.

## Provisioning workflow / metadata capture

In the provisioning workflow stage, you validate and persist data collected in the registration stage to the metastore. Both technical and business validation steps are performed, including:

- Input data feed validation
- Approval workflow triggering
- Logic processing to trigger persistence of metadata to the metadata store
- Activity auditing

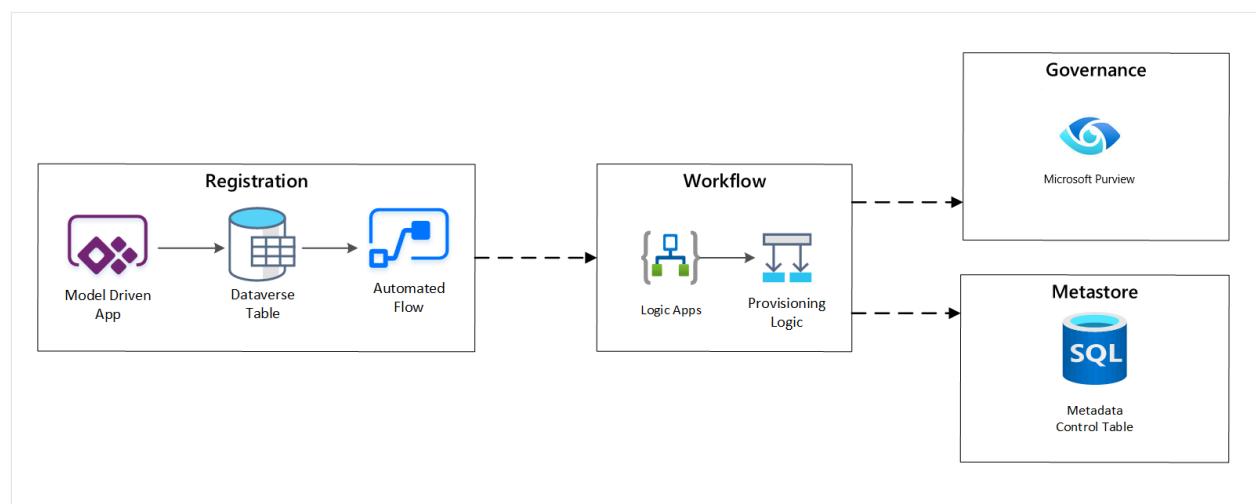


Figure 4: Registration workflow.

After ingestion requests are approved, the workflow uses the Microsoft Purview REST API to insert the sources into Microsoft Purview.

# Detailed workflow for onboarding data products

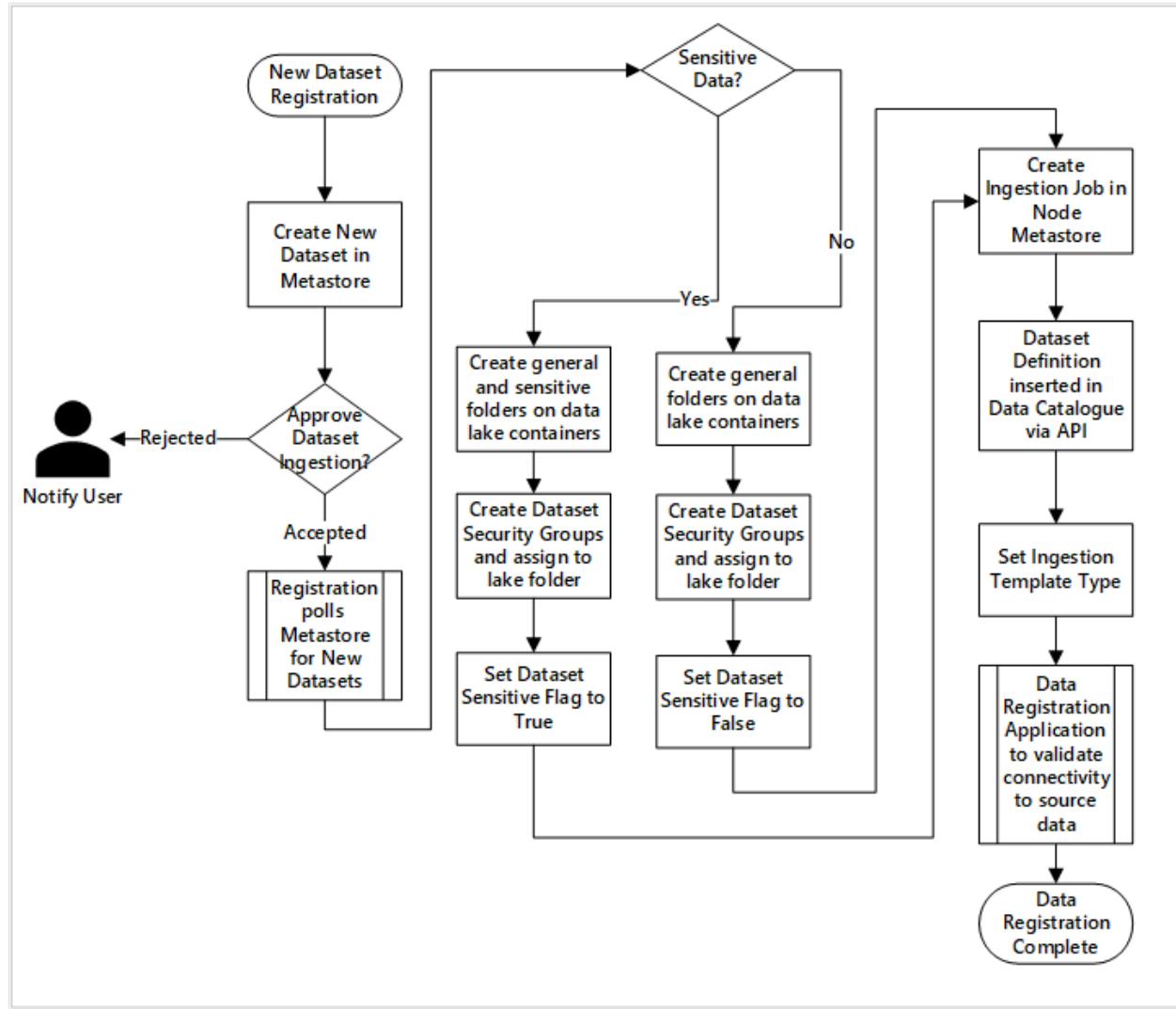


Figure 5: How new datasets are ingested (automated).

Figure 5 shows the detailed registration process for automating the ingestion of new data sources:

- Source details are registered, including production and data factory environments.
- Data shape, format, and quality constraints are captured.
- Data application teams should indicate if data is **sensitive (Personal data)**. This classification drives the process during which data lake folders are created to ingest raw, enriched and curated data. The source names raw and enriched data and the data product names curated data.
- Service principal and security groups are created for ingesting and giving access to a dataset.
- An ingestion job is created in the data landing zone Data Factory metastore.
- An API inserts the data definition into Microsoft Purview.
- Subject to the validation of the data source and approval by the ops team, details are published to a Data Factory metastore.

# Ingestion scheduling

Within Azure Data Factory, [metadata-driven copy tasks](#) provide functionality that enables orchestration pipelines to be driven by rows within a Control Table stored in Azure SQL Database. You can use the Copy Data Tool to pre-create metadata-driven pipelines.

After a pipeline has been created, your provisioning workflow adds entries to the Control Table to support ingestion from sources identified by the data asset registration metadata. The Azure Data Factory pipelines and the Azure SQL Database containing your Control Table metastore can both exist within each data landing zone to create new data sources and ingest them into data landing zones.

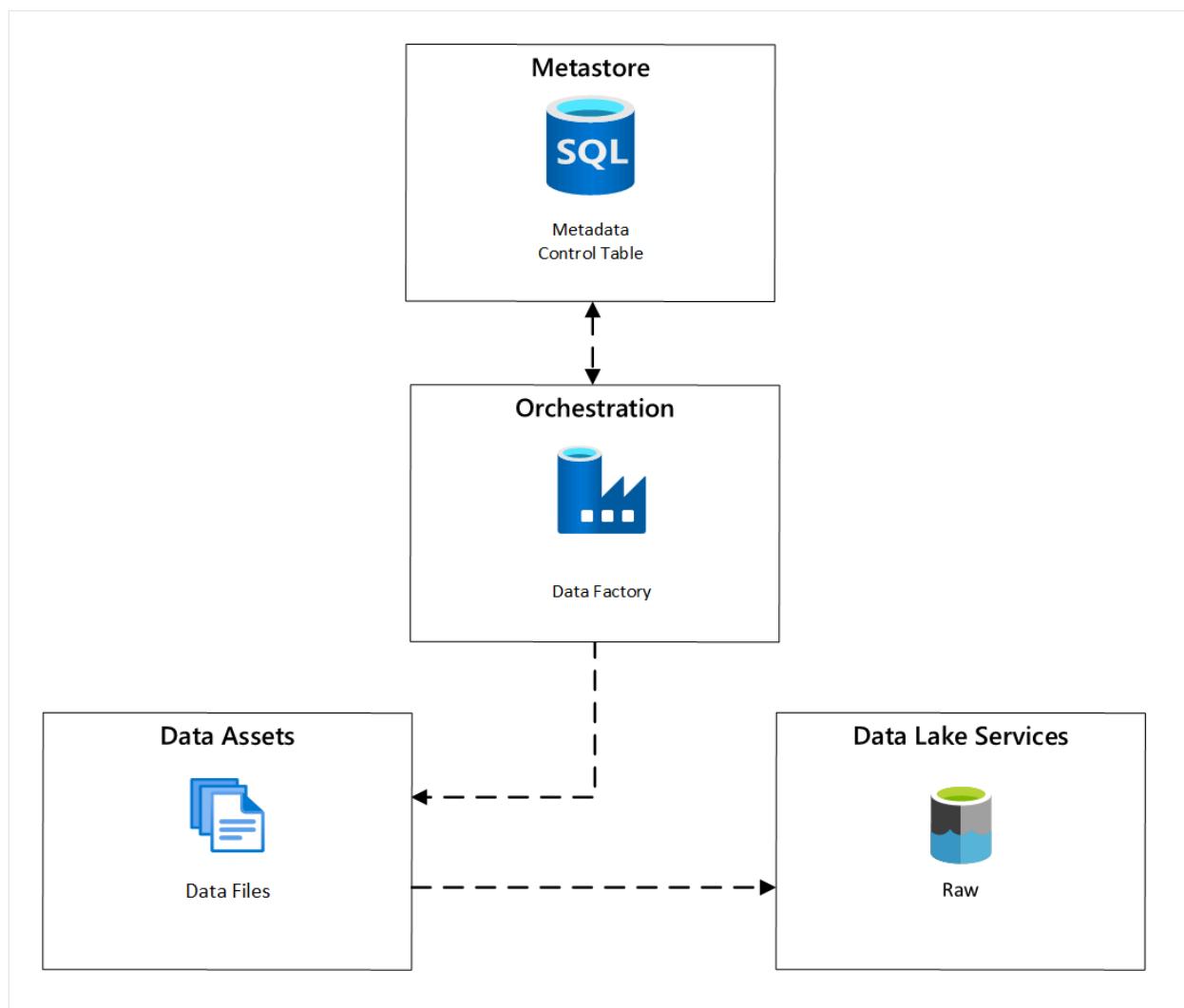
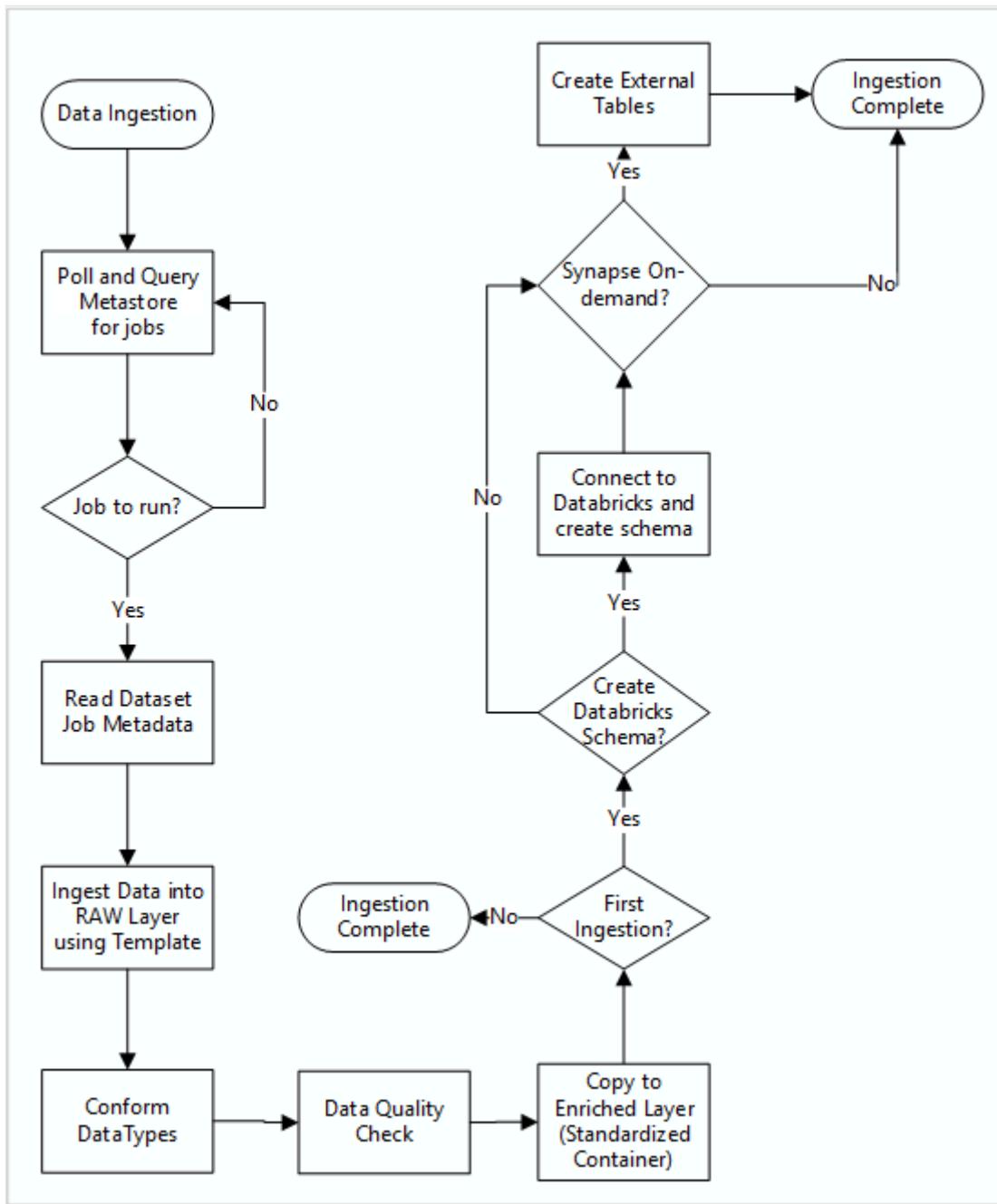


Figure 6: Scheduling of data asset ingestion.

## Detailed workflow for ingesting new data sources

The following diagram shows how to pull registered data sources in a Data Factory SQL Database metastore and how data is first ingested:



Your Data Factory ingestion master pipeline reads configurations from a Data Factory SQL Database metastore, then runs iteratively with the correct parameters. Data travels from the source to the raw layer in Azure Data Lake with little to no change. The data shape is validated based on your Data Factory metastore. File formats are converted to either Apache Parquet or Avro formats, then copied into the enriched layer.

Data being ingested connects to an Azure Databricks data science and engineering workspace, and a data definition gets created within the data landing zone Apache Hive metastore.

If you need to use an Azure Synapse serverless SQL pool to expose data, your custom solution should create views over the data in the lake.

If you require row-level or column-level encryption, your custom solution should land data in your data lake, then ingest data directly into internal tables in the SQL pools and

set-up appropriate security on the SQL pools compute.

## Captured metadata

When using automated data ingestion, you can query the associated metadata and create dashboards to:

- Track jobs and the latest data-loading timestamps for data products related to their functions.
- Track available data products.
- Grow data volumes.
- Obtain real-time updates about job failures.

Operational metadata can be used to track:

- Jobs, job steps, and their dependencies.
- Job performance and performance history.
- Data volume growth.
- Job failures.
- Source metadata changes.
- Business functions that depend on data products.

## Use the Microsoft Purview REST API to discover data

Microsoft Purview REST APIs should be used to register data during the initial ingestion. You can use the APIs to submit data to your data catalog soon after it's ingested.

For more information, see [how to use Microsoft Purview REST APIs](#).

## Register data sources

Use the following API call to register new data sources:

HTTP

PUT

`https://{{accountName}}.scan.purview.azure.com/datasources/{{dataSourceName}}`

URI parameters for the data source:

Name	Required	Type	Description
accountName	True	String	Name of the Microsoft Purview account
dataSourceName	True	String	Name of the data source

## Use the Microsoft Purview REST API for registration

The following examples show how to use the Microsoft Purview REST API to register data sources with payloads:

### Register an Azure Data Lake Storage Gen2 data source:

JSON

```
{
  "kind": "AdlsGen2",
  "name": "<source-name> (for example, My-AzureDataLakeStorage)",
  "properties": {
    "endpoint": "<endpoint> (for example, https://adls-
account.dfs.core.windows.net/)",
    "subscriptionId": "<azure-subscription-guid>",
    "resourceGroup": "<resource-group>",
    "location": "<region>",
    "parentCollection": {
      "type": "DataSourceReference",
      "referenceName": "<collection-name>"
    }
  }
}
```

### Register a SQL Database data source:

JSON

```
{
  "kind": "<source-kind> (for example, AdlsGen2)",
  "name": "<source-name> (for example, My-AzureSQLDatabase)",
  "properties": {
    "serverEndpoint": "<server-endpoint> (for example,
sqlservername.database.windows.net)",
    "subscriptionId": "<azure-subscription-guid>",
    "resourceGroup": "<resource-group>",
    "location": "<region>",
    "parentCollection": {
      "type": "DataSourceReference",
      "referenceName": "<collection-name>"
    }
  }
}
```

```
}
```

### ⓘ Note

The <collection-name> is a current collection that exists in an Microsoft Purview account.

## Create a scan

Learn how you can [create credentials](#) to authenticate sources in Microsoft Purview before setting up and running a scan.

Use the following API call to scan data sources:

HTTP

```
PUT  
https://{{accountName}}.scan.purview.azure.com/datasources/{{dataSourceName}}/scans/{{newScanName}}
```

URI parameters for a scan:

[+] [Expand table](#)

Name	Required	Type	Description
accountName	True	String	Name of the Microsoft Purview account
dataSourceName	True	String	Name of the data source
newScanName	True	String	Name of the new scan

## Use the Microsoft Purview REST API for scanning

The following examples show how you can use the Microsoft Purview REST API to scan data sources with payloads:

Scan an Azure Data Lake Storage Gen2 data source:

JSON

```
{  
  "name": "<scan-name>,"
```

```
"kind": "AdlsGen2Msi",
"properties":
{
    "scanRulesetType": "System",
    "scanRulesetName": "AdlsGen2"
}
}
```

Scan a SQL Database data source:

JSON

```
{
    "name": "<scan-name>",
    "kind": "AzureSqlDatabaseMsi",
    "properties":
    {
        "scanRulesetType": "System",
        "scanRulesetName": "AzureSqlDatabase",
        "databaseName": "<database-name>",
        "serverEndpoint": "<server-endpoint> (for example,  
sqlservername.database.windows.net)"
    }
}
```

Use the following API call to scan data sources:

HTTP

POST

<https://{{accountName}}.scan.purview.azure.com/datasources/{{dataSourceName}}/scans/{{newScanName}}/run>

## Next steps

- Overview of Azure Data Lake Storage for cloud-scale analytics

## Feedback

Was this page helpful?

 Yes

 No

# Overview of Azure Data Lake Storage for cloud-scale analytics

Article • 10/18/2024

The Azure Data Lake is a massively scalable and secure data storage for high-performance analytics workloads. You can create storage accounts within a single resource group for cloud-scale analytics. We recommend provisioning three [Azure Data Lake Storage Gen2](#) accounts within a single resource group similar to the `storage-rg` resource group described in the article [cloud-scale analytics architecture data landing zone overview](#).

Each storage account within your data landing zone stores data in one of three stages, which align to a [medallion architecture](#):

- Raw data (bronze)
- Enriched (silver) and curated data (gold)
- Development data lakes

A [data application](#) can consume enriched and curated data from a storage account which has been ingested an automated data agnostic ingestion service. You can create a [source aligned data application](#) if you don't implement data agnostics engine or facilitate complex connections for ingesting data from operational sources. This data application follows the same flow as a data agnostics engine when ingesting data from external data sources.

Data Lake Storage Gen2 supports fine-grained [access control lists](#) (ACLs) that protect data at the file and folder levels. Access control lists can help your organization implement tight security measures for authentication and authorization for data products to:

- Store data securely through encryption at rest.
- Access controls for Microsoft Entra users and security groups through Microsoft Entra integration.

## Data lake planning

When you plan a data lake, always consider appropriate consideration to structure, governance, and security. Multiple factors influence each data lake's structure and organization:

- The type of data stored

- How its data is transformed
- Who accesses its data
- What its typical access patterns are

Group consumers and producers based on their data access needs. It's a good idea to plan implementation and access control governance across your data lake.

If your data lake contains a few data assets and automated processes like extract, transform, load (ETL) offloading, your planning is likely to be fairly easy. If your data lake contains hundreds of data assets and involves automated and manual interaction, expect to spend a longer time planning, as you need a lot more collaboration from data owners.

## Data swamp analogy

A data swamp is an unmanaged data lake that is almost inaccessible to users. Data swamps occur when you don't implement data quality and data governance measures. You can sometimes see a data swamp in a data warehouse with existing hybrid models.

Proper governance and organization prevent data swamps. When you build a solid foundation for your data lake, it increases your chance of sustained data lake success and business value.

As the size, complexity, number of data assets, and number of users or departments of your data lake grows, it's increasingly critical for you to have a robust data catalog system. Your data catalog system ensures that your users can find, tag, and classify data while they process, consume, and govern your data lake.

For more information, see [data governance overview](#).

## Storage accounts in a logical data lake

Consider whether your organization needs one or many storage accounts, and consider what file systems you require to build your logical data lake. Single storage technology provides multiple data access methods and helps you standardize across your organization.

Data Lake Storage Gen2 is a fully managed platform as a service (PaaS). Multiple storage accounts or file systems can't incur a monetary cost until data is accessed or stored. Each Azure resource has administrative and operational overhead during provisioning, security, and governance, including backups and disaster recovery.

## Note

Three data lakes are illustrated in each data landing zone. However, depending on your requirements, you might be able to consolidate the raw, enriched, and curated layers into one storage account. You can create another storage account called 'development' where data consumers can bring other useful data products.

Consider the following factors when deciding between a consolidated or three storage account approach:

- Isolation of data environments and predictability
  - You might isolate activities that run in the raw and development zones to avoid potential effect on the curated zone, which holds data with great business value needed for critical decision making
- Features and functionality at the storage account level
  - You can choose if lifecycle management options or firewall rules must be applied at the data landing zone or data lake level.
  - Create multiple storage accounts, but not unwanted silos.
  - Avoid duplicate data projects from lack of visibility or knowledge-sharing across your organization.
  - Ensure that you have good data governance, project tracking tools, and a data catalog in place.
- Interaction of data processing tools and technologies with data across multiple lakes based on the configured permissions
- Regional versus global lakes
  - Globally distributed consumers or processes on the lake are sensitive to latency caused by geographic distances.
  - Storing data locally is a good practice.
  - Regulatory constraints and data sovereignty can require data to remain in a particular region.
  - For more information, see [multi-region deployments](#).

## Multi-region deployments

When dictated by data residency rules or a requirement that you keep data close to a user base, you might need to create Azure Data Lake accounts in multiple Azure regions. You need to create a data landing zone in one region, then replicate global data using AzCopy, Azure Data Factory, or partner products. Local data lives in-region, while global data gets replicated across multiple regions.

# Next steps

Data lake zones and containers

---

## Feedback

Was this page helpful?

 Yes

 No

# Data lake zones and containers

Article • 10/10/2024

It's important to plan your data structure before you land it into a data lake. When you have a plan, you can use security, partitioning, and processing effectively.

For an overview of data lakes, see [Overview of Azure Data Lake Storage for cloud-scale analytics](#).

## Overview

Your three data lake accounts should align to the typical data lake layers.

[+] Expand table

Lake number	Layers	Container number	Container name
1	Raw	1	Landing
1	Raw	2	Conformance
2	Enriched	1	Standardized
2	Curated	2	Data products
3	Development	1	Analytics sandbox
3	Development	#	Synapse primary storage number

The previous table shows the standard number of containers we recommend per data landing zone. The exception to this recommendation is if different soft delete policies are required for the data in a container. These requirements determine your need for more containers.

### Note

Three data lakes are illustrated in each data landing zone. The data lake sits across three data lake accounts, multiple containers, and folders, but it represents one logical data lake for your data landing zone.

Depending on your requirements, you might want to consolidate raw, enriched, and curated layers into one storage account. Keep another storage account named "development" for data consumers to bring other useful data products.

For more information about separating data lake accounts, see [Storage accounts in a logical data lake](#).

Enable Azure Storage with the [hierarchical name space feature](#), which allows you to efficiently manage files. The hierarchical name space feature organizes objects and files within an account into a hierarchy of directories and nested subdirectories. This hierarchy is organized the same way as the file system on your computer.

When your data agnostic ingestion engine or onboarding application registers a new system of record, it creates required folders in containers in the raw, enriched, and standardized data layers. If a source-aligned data application ingests the data, your data application team needs your data landing zone team to create the folders and security groups. Put a service principle name or managed identity into the correct group, and assign a permission level. Document this process for your data landing zone and data application teams.

For more information on teams, see [Understand roles and teams for cloud-scale analytics in Azure](#).

Each data product should have two folders in the data products container that your data product team owns.

In a standardized container's enriched layer, there are two folders per source system, divided by classification. With this structure, your team can separately store data that has different security and data classifications, and assign them different security access.

Your standardized container needs a general folder for *confidential or below* data and a *sensitive* folder for personal data. Control access to these folders by using access control lists (ACLs). You can create a dataset with all personal data removed, and store it in your general folder. You can have another dataset that includes all personal data in your *sensitive* personal data folder.

A combination of ACLs and Microsoft Entra groups restrict data access. These lists and groups control what other groups can and can't access. Data owners and data application teams can approve or reject access to their data assets.

For more information, see [Data privacy](#).

### Warning

Some software products don't support mounting the root of a data lake container. Because of this limitation, each data lake container in raw, curated, enriched, and development layers should contain a single folder that branches off to multiple

folders. Set up your folder permissions carefully. When you create a new folder from the root, the default ACL on the parent directory determines a child directory's default ACL and access ACL. A child file's ACL doesn't have a default ACL.

For more information, see [Access control lists \(ACLs\) in Azure Data Lake Storage Gen2](#).

## Raw layer (bronze) or data lake one

### ⓘ Note

The [medallion architecture](#) is a data design pattern that describes a series of incrementally refined data layers that provide a basic structure in the lakehouse. The bronze, silver, and gold layers signify increasing data quality at each level, with gold representing the highest quality.

Think of the raw layer as a reservoir that stores data in its natural and original state. It's unfiltered and unpurified. You might store the data in its original format, such as JSON or CSV. Or it might be cost effective to store the file contents as a column in a compressed file format, like Avro, Parquet, or Databricks Delta Lake.

This raw data is immutable. Keep your raw data locked down, and if you give permissions to any consumers, automated or human, ensure that they're read-only. You can organize this layer by using one folder per source system. Give each ingestion process write access to only its associated folder.

When you load data from source systems into the raw zone, you can choose to do:

- **Full loads** to extract a full data set.
- **Delta loads** to load only changed data.

Indicate your chosen loading pattern in your folder structure to simplify use for your data consumers.

Raw data from source systems for each source-aligned data application or automated ingestion engine source lands in the full folder or the delta folder. Each ingestion process should have write access to only its associated folder.

The differences between full loads and delta loads are:

- **Full load** - Complete data from the source can be onboarded if:

- The data volume at the source is small.
- The source system doesn't maintain a timestamp field that identifies if data is added, updated, or deleted.
- The source system overwrites the complete data each time.
  
- **Delta load** - Incremental data from the source can be onboarded if:
  - The data volume at the source is large.
  - The source system maintains a timestamp field that identifies if data is added, updated, or deleted.
  - The source system creates and updates files on data changes.

Your raw data lake is composed of your landing and conformance containers. Each container uses a 100% mandatory folder structure specific to its purpose.

## Landing container layout

Your landing container is reserved for raw data that's from a recognized source system. Your data agnostic ingestion engine or a source-aligned data application loads the data, which is unaltered and in its original supported format.

markdown

```
.
|-Landing
|--Log
|---{Application Name}
|--Master and Reference
|---{Source System}
|--Telemetry
|---{Source System}
|----{Application}
|--Transactional
|---{Source System}
|----{Entity}
|-----{Version}
|-----Delta
|-----{date (ex. runcdate=2019-08-22)}
|-----Full
```

## Raw layer conformance container

Your raw layer contains data quality conformed data. As data is copied to a landing container, data processing and computing is triggered to copy the data from the landing container to the conformance container. In this first stage, data gets converted

into the delta lake format and lands in an input folder. When data quality runs, records that pass are copied into the output folder. Records that fail land in an error folder.

markdown

```
.
```

- Conformance
  - Log
  - {Application Name}
  - Master and Reference
  - {Source System}
  - Telemetry
  - {Source System}
  - {Application}
  - Transactional
  - {Source System}
  - {Entity}
  - {Version}
  - Delta
  - Input
  - {date (ex. rundate=2019-08-22)}
  - Output
  - {date (ex. rundate=2019-08-22)}
  - Error
  - {date (ex. rundate=2019-08-22)}
  - Full
  - Input
  - {date (ex. rundate=2019-08-22)}
  - Output
  - {date (ex. rundate=2019-08-22)}
  - Error
  - {date (ex. rundate=2019-08-22)}

### Tip

Think about scenarios where you might need to rebuild an analytics platform from scratch. Consider the most granular data you need to rebuild downstream read data stores. Make sure you have a [business continuity and disaster recovery](#) plan in place for your key components.

## Enriched layer (silver) or data lake two

Think of the enriched layer as a filtration layer. It removes impurities and can also involve enrichment.

Your standardization container holds systems of record and masters. Folders are segmented first by subject area, then by entity. Data is available in merged, partitioned

tables that are optimized for analytics consumption.

## Standardized container

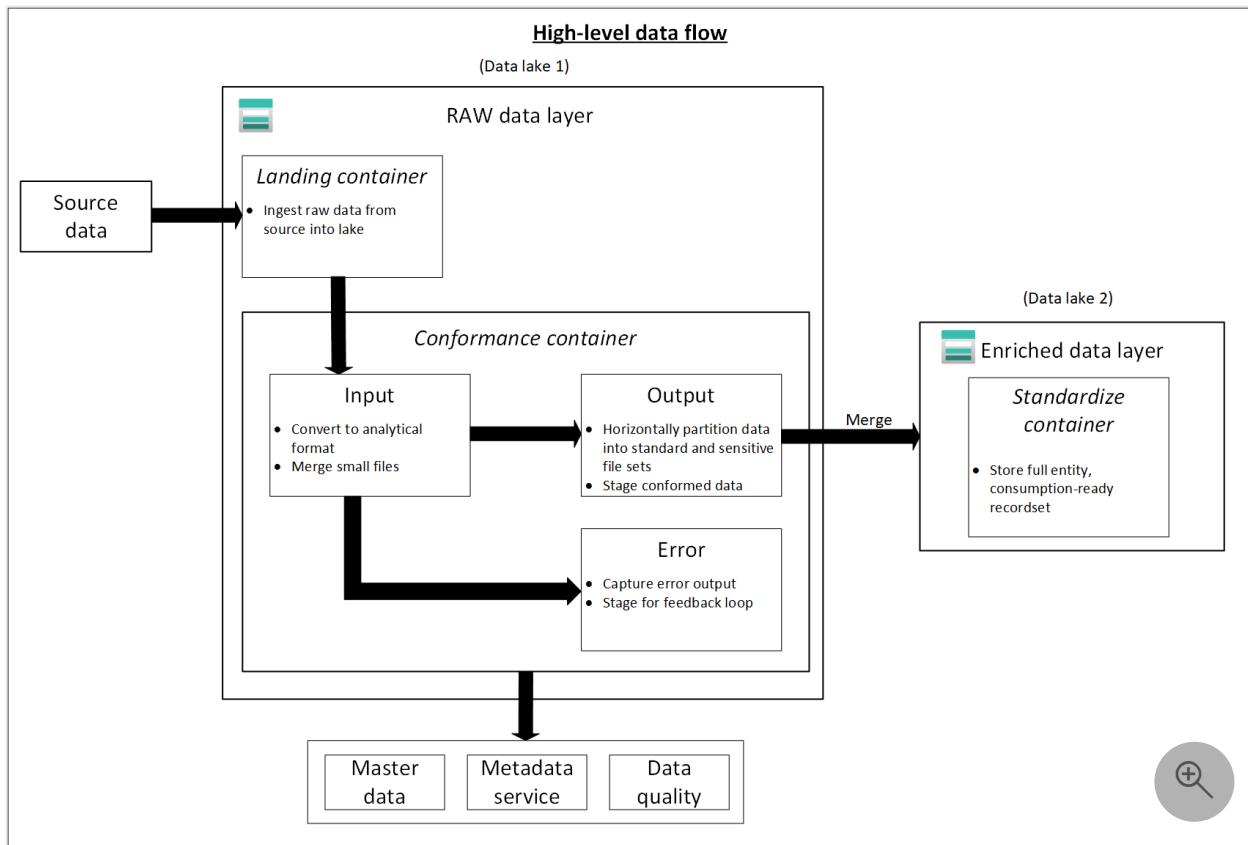
markdown

```
.  
| -Standardized  
| --Log  
| ---{Application Name}  
| --Master and Reference  
| ---{Source System}  
| --Telemetry  
| ---{Source System}  
| ---{Application}  
| --Transactional  
| ---{Source System}  
| ---{Entity}  
| ---{Version}  
| -----General  
| -----{date (ex. rundate=2019-08-22)}  
| -----Sensitive  
| -----{date (ex. rundate=2019-08-22)}
```

### ⓘ Note

This data layer is considered the silver layer or read data source. Data within this layer has had no transformations applied other than data quality, delta lake conversion, and data type alignment.

The following diagram shows the flow of data lakes and containers from source data to a standardized container.



## Curated layer (gold) or data lake two

Your curated layer is your consumption layer. It's optimized for analytics rather than data ingestion or processing. The curated layer might store data in denormalized data marts or star schemas.

Data from your standardized container is transformed into high-value data products that are served to your data consumers. This data has structure. It can be served to the consumers as-is, such as data science notebooks, or through another read data store, such as Azure SQL Database.

Use tools, like Spark or Data Factory, to do dimensional modeling instead of doing it inside your database engine. This use of tools becomes a key point if you want to make your lake the single source of truth.

If you do dimensional modeling outside of your lake, you might want to publish models back to your lake for consistency. This layer isn't a replacement for a data warehouse. Its performance typically isn't adequate for responsive dashboards or end user and consumer interactive analytics. This layer is best suited for internal analysts and data scientists who run large-scale, improvised queries or analysis, or for advanced analysts who don't have time-sensitive reporting needs. Because storage costs are lower in your data lake than your data warehouse, it can be cost effective to keep granular, low-level data in your lake. Store aggregated data in your warehouse. Generate these

aggregations by using Spark or Azure Data Factory. Persist them to your data lake before loading them into your data warehouse.

Data assets in this zone are highly governed and well documented. Assign permissions by department or by function, and organize permissions by consumer group or data mart.

## Data products container

markdown

```
.  
|-{Data Product}  
|---{Entity}  
|----{Version}  
|-----General  
|-----{date (ex. rundate=2019-08-22)}  
|-----Sensitive  
|-----{date (ex. rundate=2019-08-22)}
```

### 💡 Tip

When you land data in another read data store, like Azure SQL Database, ensure that you have a copy of that data located in your curated data. Your data product users are guided to your main read data store or Azure SQL Database instance, but they can also explore data with extra tools if you make the data available in your data lake.

## Development layer or data lake three

Your data consumers can bring other useful data products along with the data ingested into your standardized container.

In this scenario, your data platform can allocate an analytics sandbox area for these consumers. In the sandbox, they can generate valuable insights by using the curated data and data products that they bring. For example, if a data science team wants to determine the best product placement strategy for a new region, they can bring other data products, like customer demographics and usage data, from similar products in that region. The team can use the high-value sales insights from this data to analyze the product market fit and offering strategy.

### Note

The analytics sandbox area is a working area for an individual or a small group of collaborators. The sandbox area's folders have a special set of policies that prevent attempts to use this area as part of a production solution. These policies limit the total available storage and how long data can be stored.

These data products are usually of unknown quality and accuracy. They're still categorized as data products, but are temporary and only relevant to the user group that's using the data.

When these data products mature, your enterprise can promote these data products to your curated data layer. To keep your data product teams responsible for new data products, provide the teams with a dedicated folder on your curated data zone. They can store new results in the folder and share them with other teams across your organization.

### Note

For every Azure Synapse workspace you create, use data lake three to create a container to use as primary storage. This container stops Azure Synapse workspaces from interfering with your curated and enriched zones' throughput limits.

## Next steps

[Key considerations for Azure Data Lake Storage](#)

## Feedback

Was this page helpful?

 Yes

 No

# Key considerations for Azure Data Lake Storage

Article • 01/08/2025

Azure Storage offers a variety of storage options for your data. This article provides considerations to help you choose the appropriate access tier so that you can balance cost and performance. It also describes the lifecycle management of Storage, including features and best practices to help you use the access tiers effectively.

## Lifecycle management

Azure Storage offers various access tiers that you can use to store blob object data. Choose the tier that best suits your workload to optimize cost.

- Use a **hot tier** to store frequently accessed data.
- Use a **cool tier** to store infrequently accessed data. This tier stores data for at least 30 days.
- Use a **cold tier** to store infrequently accessed or modified data. This tier stores data for at least 90 days. The cold tier has lower storage costs and higher access costs compared to the cool tier.
- Use an **archive tier** to store rarely accessed data. This tier stores data for at least 180 days. Access to this data can have flexible latency requirements, which means that it can take hours to retrieve data.

### Important

The online access tiers (hot, cool, and cold) don't have reliability, security, operational excellence, or performance efficiency trade-offs. Therefore, you should base your decision on the cost for each blob. Consider your workload access data size, operational interactions, and the time before the blob is deleted. [Select the appropriate tier](#) for each blob based on these factors. For more information, see [Plan and manage costs for Azure Blob Storage](#).

Consider the following factors when you use access tiers:

- Set only the hot and cool access tiers at the account level. The account level doesn't support the archive access tier.

- Set the hot, cool, and archive tiers at the blob level during upload or after upload.
- Data in the cool and cold tiers has slightly lower availability, but these tiers offer features that are similar to those of the hot tier, such as high durability, retrieval latency, and throughput. For data in the cool or cold tiers, lower availability and higher access costs are acceptable trade-offs for reduced storage costs compared to the hot tier.
- Archive storage stores data offline and offers the lowest storage costs. But it also incurs the highest data rehydration and access costs.

For more information, see [Access tiers for blob data](#).

**ⓘ Important**

For cloud-scale analytics, use a custom microservice to implement [lifecycle management](#). Carefully consider the impact of moving user-discoverable data to cool storage. Move sections of your data lake to the cool tier only for well-understood workloads.

## Data lake connectivity

Each data lake should use private endpoints that you integrate into the virtual network of your data landing zone. To provide access across landing zones, connect your data landing zones through virtual network peering. This connection provides an optimal solution from both a cost and access-control perspective.

For more information, see [Private endpoints](#) and [Data management landing zone to data landing zone](#).

**ⓘ Important**

A data landing zone can access data in a different data landing zone via virtual network peering. Private endpoints establish the connection associated with each data lake account. We recommend that you turn off all public access to your lakes and use private endpoints. Your platform operations team should control network connectivity across your data landing zones.

## Soft delete for containers

Soft delete for containers helps protect your data from accidental or malicious deletion. If you enable container soft delete for your storage account, Storage retains deleted containers and their contents for a specified length of time. During the data-retention period, you can restore previously deleted containers. This action also restores blobs that were in that container when it was deleted.

Enable the following data-protection features to enhance end-to-end blob data protection:

- Use container soft delete to restore a deleted container. For more information, see [Enable and manage soft delete for containers](#).
- Use blob soft delete to restore a deleted blob or version. For more information, see [Enable and manage soft delete for blobs](#).

#### Warning

After you delete a storage account, you can't undo the deletion. Container soft delete doesn't protect against storage account deletion, only against the deletion of containers within an account. To protect a storage account from deletion, configure a lock on the storage account resource. For more information, see [Lock resources to prevent unexpected changes](#).

## Monitoring

In a data landing zone, send all monitoring to your [Azure landing zone management subscription](#) for analysis.

For more information, see [Monitor Azure resources with Azure Monitor](#) and [Monitor Blob Storage](#).

Log entries are created only for requests against the service endpoint. The following types of authenticated requests are logged:

- Successful requests
- Failed requests, including timeouts, throttling, network problems, authorization problems, and other errors
- Requests that use a shared access signature (SAS) or OAuth, including failed and successful requests
- Requests to analytics data, like classic log data in the `$logs` container and class metric data in the `$metric` tables

Requests made by the storage service itself, like log creation or deletion, aren't logged. The following types of anonymous requests are logged:

- Successful requests
- Server errors
- Time out errors for both client and server
- Failed HTTP GET requests that have the error code 304 (`Not Modified`)

Other failed anonymous requests aren't logged.

 **Important**

Set your default monitoring policy to audit storage and send logs to your enterprise-scale management subscription.

## Data lake zone security

We recommend the following security patterns for data lake zones:

- **Raw usage** allows access to data by using security principal names (SPNs) only. We recommend that you use managed identities.
- **Enriched usage** allows access to data by using SPNs only. We recommend that you use managed identities.
- **Curated usage** allows access to data by using SPNs and user principal names (UPNs).

For more information, see [Access control model in Data Lake Storage](#).

## Next step

- [The ingest process with cloud-scale analytics in Azure](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# Use Azure Synapse Analytics with cloud-scale analytics

Article • 12/10/2024

Azure Synapse Analytics is the provisioned, integrated analytics service that accelerates time to insight across data warehouses and big data systems. Azure Synapse Analytics brings together:

- The best SQL technologies used in enterprise data warehousing.
- Spark technologies used for big data.
- Pipelines for data application (source-aligned) and extract, transform, and load (ETL) or extract, load, and transform (ELT).

Azure Synapse studio is a tool in Azure Synapse that provides a unified experience for management, monitoring, coding, and security. Synapse studio has deep integration with other Azure services like Power BI, Azure Cosmos DB, and Azure Machine Learning.

## ⓘ Note

This section aims to describe prescribed configurations which are specific to cloud-scale analytics. It's a compliment to the official [Azure Synapse Analytics documentation](#).

## Overview

During the initial setup of a [data landing zone](#), you can deploy a single Azure Synapse Analytics workspace for use by all analysts and data scientists. You can create more workspaces for specific data integrations or data products.

You might need extra Azure Synapse Analytics workspaces if your data product needs to provide access to the [standardized data](#) with row-level and column-level security. You can provide these workspaces with Azure Synapse pools. Data products teams might require their own workspace for creating data products and a separate workspace that's only for product teams with scoped development access.

## Azure Synapse Analytics setup

The first step in the deployment Azure Synapse Analytics is to set up an Azure Synapse workspace which is [connected to an Microsoft Purview account](#).

## Azure Synapse Analytics networking

A data landing zone creates workspaces with an [Azure Synapse Analytics managed virtual network](#). Communication with Azure Synapse happens through the three endpoints it exposes: SQL pool, SQL on-demand, and the development endpoint.

At the network level, cloud-scale analytics uses [synapse managed private endpoints](#). These endpoints ensure all of the traffic between the data landing zone virtual network and Azure Synapse workspaces moves entirely over the Microsoft backbone network.

## Azure Synapse data access control

Use access control lists with [Microsoft Entra pass-through in Azure Synapse Analytics](#) to manage access to the files in the data lake.

For data where you need to restrict columns and rows returned, we recommend row-level and column-level security to restrict the data access on the tables in Azure Synapse SQL dedicated or serverless pool. Row-level security and column-level security is implemented at the database level and in addition to the database roles.

For example, row-level security ensures that users in a specific data application (source-aligned) or data product only see their own data. Even if the table contains data for the entire enterprise.

You can combine row-level security with column-level security to restrict access to columns with sensitive data. This way, both row-level security and column-level security apply the access restriction logic at the database tier rather than the application tier. The permission is evaluated every time data access is attempted from any tier.

### Note

Azure Synapse serverless SQL pool supports [Column-level security](#) for views and not for external tables. In case of external tables one can create a logical view on top of the external table and then apply Column-level security. In case of Row-level security, custom views can be used as a workaround.

For more information, see [Azure Synapse Analytics data access control](#).

## Azure Synapse data access control in Azure Data Lake

When deploying an Azure Synapse Analytics workspace, you need an Azure Data Lake Storage account from the subscription or by manually using the storage account URL.

The specified storage account is set as **primary** for the deployed Azure Synapse workspace to store its data. Azure Synapse stores data in a container that includes Apache Spark tables and Spark application logs in a folder called `/synapse/{workspaceName}`. It also has a container for managing any libraries that you choose to install.

### 💡 Tip

We recommend using a dedicated container on the [Development layer or data lake three](#) account. This container is used as primary storage to store Spark metadata.

Refer to [Azure Synapse Analytics data access control](#) for recommendations on how to set up data access.

---

## Feedback

Was this page helpful?

 Yes

 No

# Azure Machine Learning as a data product for cloud-scale analytics

Article • 04/22/2022

Azure Machine Learning is an integrated platform for managing the machine learning lifecycle from beginning to end, including help with the creation, operation, and consumption of machine learning models and workflows. A few benefits of the service include:

- Capabilities support creators to increase their productivity by helping them to manage experiments, access data, track jobs, tune hyperparameters, and automate workflows.
- The model's capacity to be explained, reproduced, audited, and integrated with DevOps, plus a rich security control model, can support operators to meet governance and compliance requirements.
- Managed inference capabilities and robust integration with Azure compute and data services can help to simplify how the service is consumed.

Azure Machine Learning covers all aspects of the data science lifecycle. It covers datastore and dataset registration to model deployment. It can be used for any kind of machine learning, from classical machine learning to deep learning. It includes supervised and unsupervised learning. Whether you prefer to write Python, R code, or use zero-code or low-code options such as the designer, you can build, train, and track accurate machine learning and deep learning models in an Azure Machine Learning workspace.

Azure Machine Learning, the Azure platform, and Azure AI services can work together to manage the machine learning lifecycle. A machine learning practitioner can use Azure Synapse Analytics, Azure SQL Database, or Microsoft Power BI to start analyzing data and transition to Azure Machine Learning for prototyping, managing experimentation, and operationalization. In Azure landing zones, Azure Machine Learning can be considered a [data product](#).

## Azure Machine Learning in cloud-scale analytics

A Cloud Adoption Framework (CAF) landing zone foundation, cloud-scale analytics data landing zones, and the configuration of Azure Machine Learning set up machine

learning professionals with a preconfigured environment to which they can repeatedly deploy new machine learning workloads or migrate existing workloads. These capabilities can help machine learning professionals to gain more agility and value for their time.

The following design principles can guide the implementation of Azure Machine Learning Azure landing zones:

- **Accelerated data access:** Preconfigure landing zone storage components as data stores in the Azure Machine Learning workspace.
- **Enabled collaboration:** Organize workspaces by project and centralize access management for landing zone resources to support data engineering, data science, and machine learning professionals to work together.
- **Secure implementation:** As a default for each deployment, follow best practices and use network isolation, identity, and access management to secure data assets.
- **Self-service:** Machine learning professionals can gain more agility and organization by exploring options to deploy new project resources.
- **Separation of concerns between data management and data consumption:** Identity passthrough is the default authentication type for Azure Machine Learning and storage.
- **Faster data application (source-aligned):** Azure Data Factory, Azure Synapse Analytics, and Databricks landing zones can be preconfigured to link to Azure Machine Learning.
- **Observability:** Central logging and reference configurations can help to monitor the environment.

## Implementation overview

### Note

This section recommends configurations specific to cloud-scale analytics. It complements Azure Machine Learning documentation and Cloud Adoption Framework best practices.

## Workspace organization and setup

You can deploy the number of machine learning workspaces that your workloads require and for every landing zone that you deploy. The following recommendations can help your setup:

- Deploy at least one machine learning workspace per project.
- Depending on your machine learning project's lifecycle, deploy one development (dev) workspace to prototype use cases and explore data early on. For work that requires continuous experimentation, testing, and deployment, deploy a staging and production workspace.
- When multiple environments are needed for dev, staging, and production workspaces in a data landing zone, we recommend avoiding data duplication by having each environment land in the same production data landing zone.
- See [Organize and set up Azure Machine Learning environments](#) to learn more about how to organize and set up Azure Machine Learning resources.

For each default resource configuration in an data landing zone, an Azure Machine Learning service is deployed in a dedicated resource group with the following configurations and dependent resources:

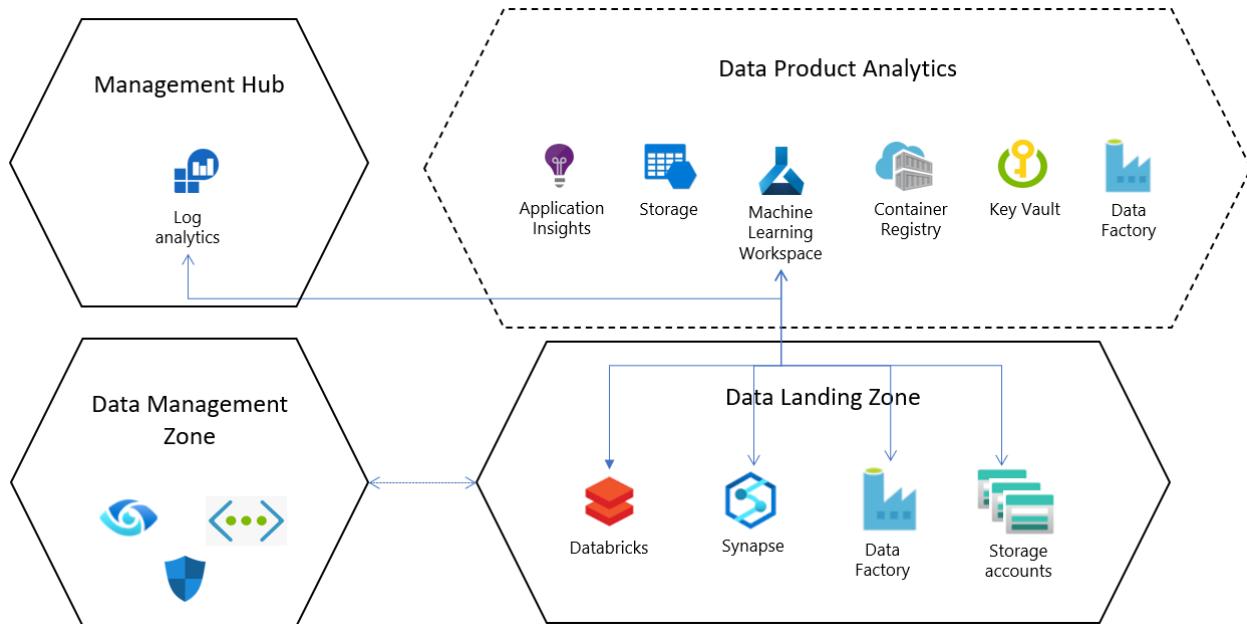
- Azure Key Vault
- Application Insights
- Azure Container Registry
- Use Azure Machine Learning to connect to an Azure Storage account and Azure Active Directory (Azure AD) identity-based authentication to help users connect to the account.
- Diagnostic logging is set up for each workspace and configured to a central Log Analytics resource in enterprise-scale; this can help Azure Machine Learning job health and resource statuses to be analyzed centrally within and across landing zones.
- See [What is an Azure Machine Learning workspace?](#) to learn more about Azure Machine Learning resources and dependencies.

## Integration with data landing zone core services

The data landing zone comes with a default set of services that are deployed in the [core services layer](#). These core services can be configured when Azure Machine Learning is deployed in data landing zone.

- Connect Azure Synapse Analytics or Databricks workspaces as linked services to integrate data and process big data.

- By default, data lake services are provisioned in the data landing zone, and Azure Machine Learning product deployments come with connections (data stores) that are preconfigured to these storage accounts.



## Network connectivity

Networking for implementing Azure Machine Learning in Azure landing zones is set up with [security best practices for Azure Machine Learning](#) and CAF [networking best practices](#). These best practices include the following configurations:

- Azure Machine Learning and dependent resources are configured to use Private Link endpoints.
- Managed compute resources are deployed only with private IP addresses.
- Network connectivity to the Azure Machine Learning public base image repository and partner services like Azure Artifacts can be configured at a network level.

## Identity and access management

Consider the following recommendations for managing user identities and access with Azure Machine Learning:

- Data stores in Azure Machine Learning can be configured to use credential- or identity-based authentication. When you use [access control and data lake configurations in Azure Data Lake Storage Gen2](#), configure data stores to use identity-based authentication; this allows Azure Machine Learning to optimize user access permissions for storage.

- Use Azure AD groups to manage user permissions for storage and machine learning resources.
- Azure Machine Learning can use [user-assigned managed identities for access control](#) and limit the range of access to Azure Container Registry, Key Vault, Azure Storage, and Application Insights.
- Create user-assigned managed identities to managed compute clusters created in Azure Machine Learning.

## Provision infrastructure through self-service

Self-service can be enabled and governed with [policies for Azure Machine Learning](#). The following table lists a set of default policies when you deploy Azure Machine Learning. For more information, see [Azure Policy built-in policy definitions for Azure Machine Learning](#).

Policy	Type	Reference
Azure Machine Learning workspaces should use Azure Private Link.	Built-in	<a href="#">View in the Azure portal</a> ↗
Azure Machine Learning workspaces should use user-assigned managed identities.	Built-in	<a href="#">View in the Azure portal</a> ↗
[Preview]: Configure allowed registries for specified Azure Machine Learning computes.	Built-in	<a href="#">View in the Azure portal</a> ↗
Configure Azure Machine Learning workspaces with private endpoints.	Built-in	<a href="#">View in the Azure portal</a> ↗
Configure machine learning computes to disable local authentication methods.	Built-in	<a href="#">View in the Azure portal</a> ↗
Append-machinelearningcompute-setupscriptscreationscript	Custom (CAF landing zones)	<a href="#">View on GitHub</a> ↗
Deny-machinelearning-hbiworkspace	Custom (CAF landing zones)	<a href="#">View on GitHub</a> ↗
Deny-machinelearning-publicaccesswhenbehindvnet	Custom (CAF landing zones)	<a href="#">View on GitHub</a> ↗
Deny-machinelearning-AKS	Custom (CAF landing zones)	<a href="#">View on GitHub</a> ↗
Deny-machinelearningcompute-subnetid	Custom (CAF landing zones)	<a href="#">View on GitHub</a> ↗

Policy	Type	Reference
Deny-machinelearningcompute-vmsize	Custom (CAF landing zones)	<a href="#">View on GitHub ↗</a>
Deny-machinelearningcomputecluster-remoteloginportpublicaccess	Custom (CAF landing zones)	<a href="#">View on GitHub ↗</a>
Deny-machinelearningcomputecluster-scale	Custom (CAF landing zones)	<a href="#">View on GitHub ↗</a>

## Recommendations for managing your environment

Cloud-scale analytics data landing zones outline reference implementation for repeatable deployments, which can help you to set up manageable and governable environments. Consider the following recommendations for using Azure Machine Learning to manage your environment:

- Use Azure AD groups to manage access to machine learning resources.
- Publish a central monitoring dashboard to monitor pipeline health, compute utilization, and quota management for machine learning.
- If you traditionally use built-in Azure policies and need to meet additional compliance requirements, build custom Azure policies to enhance governance and self-service.
- To track research and development costs, deploy one machine learning workspace in the landing zone as a shared resource during the early stages of exploring your use case.

### ⓘ Important

Use Azure Machine Learning clusters for production-grade model training, and Azure Kubernetes Service (AKS) for production-grade deployments.

### 💡 Tip

Use Azure Machine Learning for data science projects. It covers the end-to-end workflow with subservices and features, and allows the process to be fully automated.

# Next steps

Use the [Data Product Analytics](#) template and guidance to deploy Azure Machine Learning, and reference [Azure Machine Learning documentation and tutorials](#) to get started with building your solutions.

Continue to the following four Cloud Adoption Framework articles to learn more about Azure Machine Learning deployment and management best practices for enterprises:

- [Organize and set up Azure Machine Learning environments](#): When planning an Azure Machine Learning deployment, how do team structures, environments, or the geography of resources affect how workspaces are set up?
- [Azure Machine Learning best practices for enterprise security](#): Learn how to secure your environment and resources with Azure Machine Learning.
- [Manage budgets, costs, and quota for Azure Machine Learning at organizational scale](#): Organizations face many management and optimization challenges when managing workload, team, and user compute costs incurred from Azure Machine Learning.
- [Machine learning DevOps guide](#): Machine learning DevOps is an organizational change that relies on a combination of people, process, and technology to deliver machine learning solutions in a robust, scalable, reliable, and automated way. This guide summarizes best practices and information for enterprises to use Azure Machine Learning to adopt machine learning DevOps.

# Organize and set up Azure Machine Learning environments

Article • 08/30/2022

When you're planning an Azure Machine Learning deployment for an enterprise environment, there are some common decision points that affect how you create the workspace:

- **Team structure:** The way you organize your data science teams and collaborate on projects, given use case and data segregation, or cost management requirements
- **Environments:** The environments you use as part of your development and release workflow to segregate development from production
- **Region:** The location of your data and the audience to which you need to serve your machine learning solution

## Team structure and workspace setup

The workspace is the top-level resource in Azure Machine Learning. It stores the artifacts that are produced when working with machine learning and the managed compute and pointers to attached and associated resources. From a manageability standpoint, the workspace as an Azure Resource Manager resource supports Azure role-based access control (Azure RBAC), management by Policy, and you can use it as a unit for cost reporting.

Organizations typically choose one or a combination of the following solution patterns to follow manageability requirements.

**Workspace per team:** Use one workspace for each team when all members of a team require the same level of access to data and experimentation assets. For example, an organization with three machine learning teams might create three workspaces, one for each team.

The benefit of using one workspace per team is that all machine learning artifacts for the team's projects are stored in one place. You can see productivity increases because team members can easily access, explore, and reuse experimentation results. Organizing your workspaces by team reduces your Azure footprint and simplifies cost management by team. Because the number of experimentation assets can grow quickly, you can keep your artifacts organized by following naming and tagging conventions. For recommendations about how to name resources, see [Develop your naming and tagging strategy for Azure resources](#).

With this approach, each team member must have similar data access level permissions. Granular role-based access control (RBAC) and access control lists (ACL) for data sources and experimentation assets are limited within a workspace. You can't have use case data segregation requirements.

**Workspace per project:** Use one workspace for each project if you require segregation of data and experimentation assets by project, or have cost reporting and budgeting requirements at a project level. For example, you might have an organization with four machine learning teams that run three projects each for a total of 12 workspace instances.

The benefit of using one workspace per project is that you manage costs at the project level. A team typically creates a dedicated resource group for Azure Machine Learning and associated resources for similar reasons. When you work with external contributors, for example, a project-centered workspace simplifies collaboration on a project because external users only need to be granted access to the project resources, not the team resources.

Something to consider with this approach is the isolation of experimentation results and assets. The discovery and reuse of assets might be more difficult because assets are spread across multiple workspace instances.

**Single Workspace:** Use one workspace for non-team or non-project related work, or when costs can't be directly associated to a specific unit of billing, for example with R&D.

The benefit of this setup is the cost of individual, non-project related work can be decoupled from project-related costs. When you set up a single workspace for all users to do their individual work, you reduce your Azure footprint.

With this approach, the workspace might become cluttered quickly when many machine learning practitioners share the same instance. Users might require UI-based filtering of assets to effectively find their resources. You can create shared machine learning workspaces for each business division to mitigate scale concerns or to segment budgets.

## Environments and workspace setup

An environment is a collection of resources that deployments target based on their stage in the application lifecycle. Common examples of environment names are Dev, Test, QA, Staging, and Production.

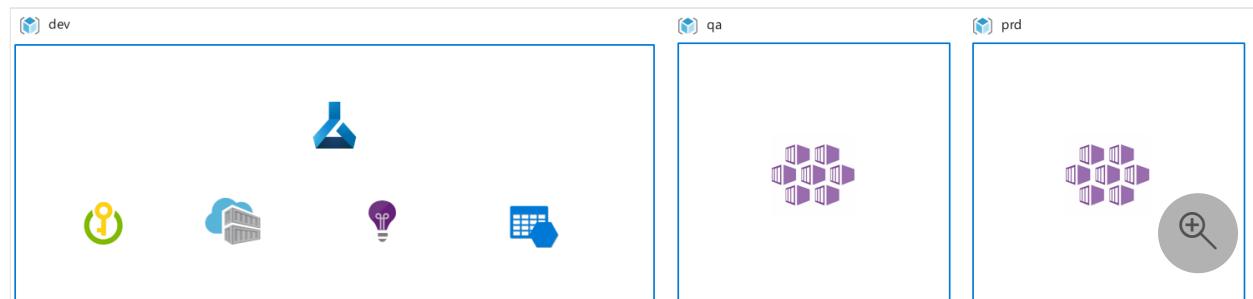
The development process in your organization affects requirements for environment usage. Your environment affects the setup of Azure Machine Learning and associated

resources, such as attached compute. For example, data availability might constrain the manageability of having a machine learning instance available for each environment. The following solution patterns are common:

**Single environment workspace deployment:** When you choose a single environment workspace deployment, Azure Machine Learning deploys to one environment. This setup is common for research-centered scenarios, where there's no need to release machine learning artifacts based on their lifecycle stage, across environments. Another scenario where this setup makes sense is when only inferencing services, and not machine learning pipelines, are deployed across environments.

The benefit of a research-centered setup is a smaller Azure footprint and minimal management overhead. This way of working implies no need to have an Azure Machine Learning workspace deployed in each environment.

With this approach, a single environment deployment is subject to data availability. So, be cautious when you set up your datastore. If you set up extensive access, for example, writer access on production data sources, you might unintentionally harm data quality. If you bring work to production in the same environment where the development happens, the same RBAC restrictions apply for both the development work and the production work. This setup might make both environments too rigid or too flexible.



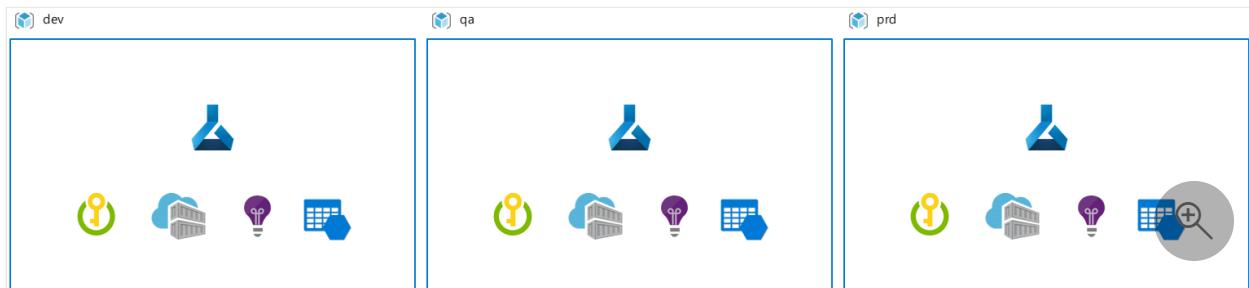
**Multiple environment workspace deployment:** When you choose a multiple environment workspace deployment, a workspace instance deploys for each environment. A common scenario for this setup is a regulated workplace with a clear separation of duties between environments, and for users who have resource access to those environments.

The benefits of this setup are:

- Staged rollout of machine learning workflows and artifacts. For example, models across environments, with the potential to enhance agility and reduce time-to-deployment.
- Enhanced security and control of resources because you can assign more access restrictions in downstream environments.

- Training scenarios on production data in non-development environments because you can give a select group of users access.

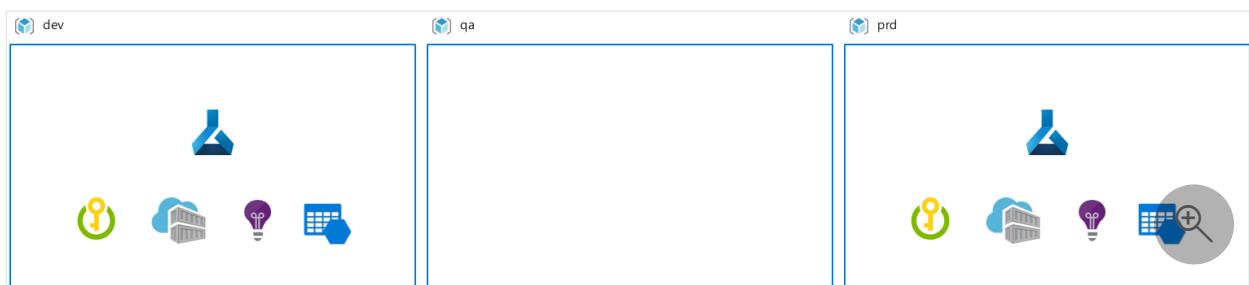
With this approach, you're at risk for more management and process overhead. This setup requires a fine-grained development and rollout process for machine learning artifacts across workspace instances. Also, data management and engineering effort might be required to make production data available for training in the development environment. Access management requires you to give a team access to resolve and investigate incidents in production. And finally, your team needs Azure DevOps and machine learning engineering expertise to implement automation workflows.



**One environment with limited data access, one with production data access:** When you choose this setup, Azure Machine Learning deploys to two environments: one with limited data access and one with production data access. This setup is common if you need to segregate development and production environments. For example, you might be working under organizational constraints to make production data available in any environment, or you might want to segregate development work from production work without duplicating data more than required due to the high cost of maintenance.

The benefit of this setup is the clear separation of duties and access between development and production environments. Another benefit is lower resource management overhead when compared to a multi-environment deployment scenario.

With this approach, you need a defined development and rollout process for machine learning artifacts across workspaces. Also, it might require data management and engineering effort to make production data available for training in a development environment. But this approach might require relatively less effort than a multi-environment workspace deployment.



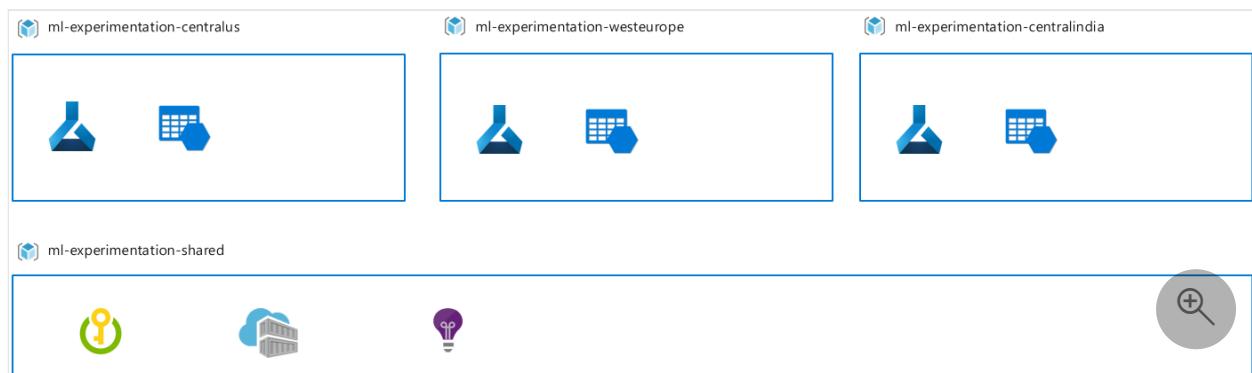
# Regions and resource setup

The location of your resources, data, or users might require you to create Azure Machine Learning workspace instances and associated resources in multiple Azure regions. For example, one project might span its resources across the West Europe and East US Azure regions for performance, cost, and compliance reasons. The following scenarios are common:

**Regional training:** The machine learning training jobs run in the same Azure region as where the data is located. In this setup, a machine learning workspace deploys to each Azure region where data is located. This scenario is common when you need to meet compliance, or when you have data movement constraints across regions.

The benefit of this setup is you can do experimentation in the data center where the data is located with the least network latency. With this approach, when a machine learning pipeline runs across multiple workspace instances, it adds more management complexity. It becomes challenging to compare experimentation results across instances and adds overhead to quota and compute management.

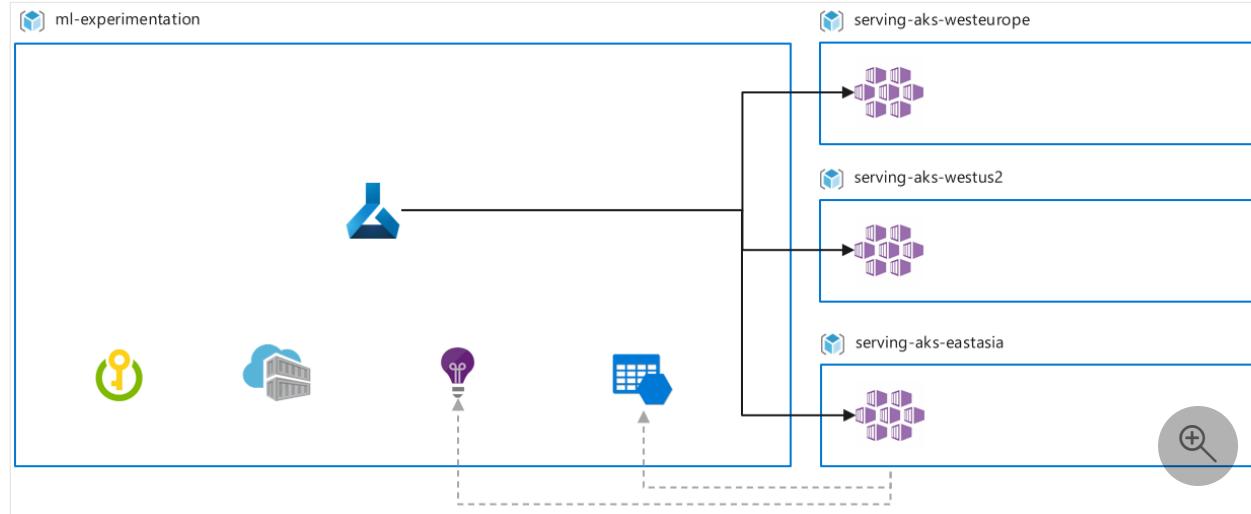
If you want to attach storage across regions, but use compute from one region, Azure Machine Learning supports the scenario of attaching storage accounts in a region rather than the workspace. Metadata, for example metrics, is stored in the workspace region.



**Regional serving:** Machine learning services deploy close to where the target audience lives. For example, if target users are in Australia and the main storage and experimentation region is West Europe, deploy the machine learning workspace for experimentation in West Europe. You then deploy an AKS cluster for inference endpoint deployment in Australia.

The benefits of this setup are the opportunity for inferencing in the data center where new data is ingested, minimizing latency and data movement, and compliance with local regulations.

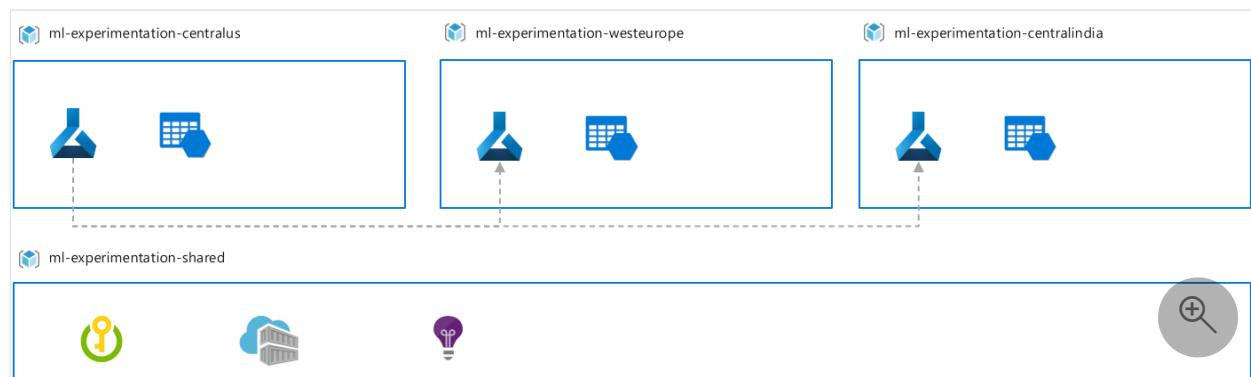
With this approach, a multi-region setup provides several advantages, but also adds more overhead on quota and compute management. When you have a requirement for batch inferencing, regional serving might require a multi-workspace deployment. Data collected through inferencing endpoints might need to be transferred across regions for retraining scenarios.



**Regional fine-tuning:** A base model trains on an initial dataset, for example, public data or data from all regions, and is later fine-tuned with a regional dataset. The regional dataset might only exist in a particular region because of compliance or data movement constraints. For example, you might need base model training to be done in a workspace in region A, while fine tuning happens in a workspace in region B.

The benefit of this setup is you can experiment compliantly in the data center where the data resides. You can also still take advantage of base model training on a larger dataset in an earlier pipeline stage.

This approach supports complex experimentation pipelines but it might create more challenges. For example, when you compare experiment results across regions, it might add more overhead to the quota and compute management.



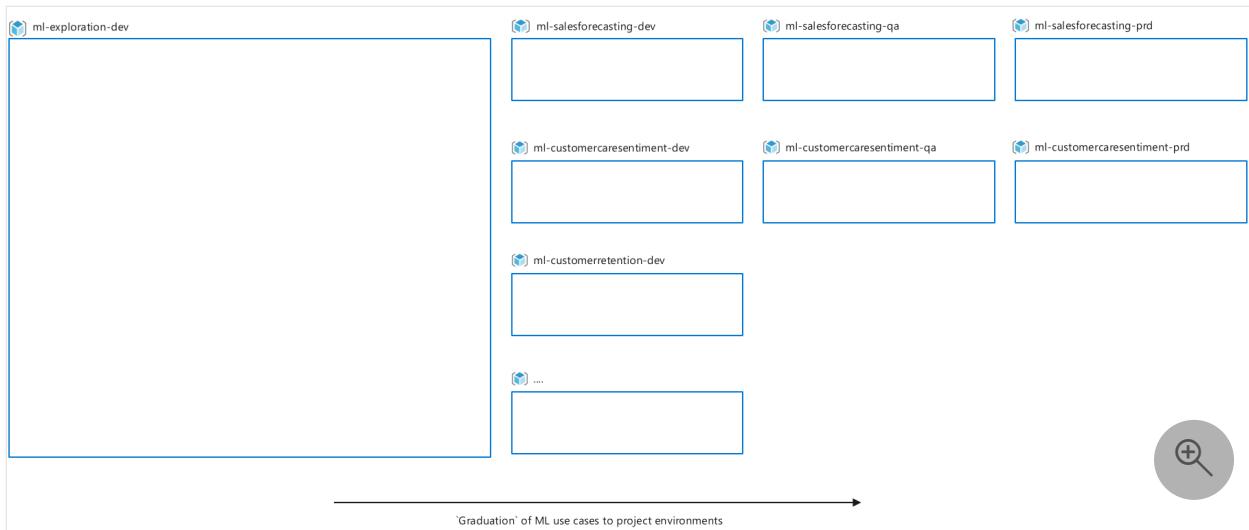
## Reference implementation

To illustrate the deployment of Azure Machine Learning in a larger setting, this section shows how the organization 'Contoso' sets up Azure Machine Learning, given their organizational constraints, reporting, and budgeting requirements:

- Contoso creates resource groups on a solution basis for cost management and reporting reasons.
- IT administrators only create resource groups and resources for funded solutions to meet budget requirements.
- Because of the exploratory and uncertain nature of Data Science, users need a place to experiment and work for use case and data exploration. Often, exploratory work can't be directly associated to a particular use case, and can be associated only to an R&D budget. Contoso wants to fund some machine learning resources centrally that anyone can use for exploration purposes.
- Once a machine learning use case proves to be successful in the exploratory environment, teams can request resource groups. For example, the company can set up Dev, QA, and Production for iterative experimentation project work, and access to production data sources.
- Data segregation and compliance requirements don't allow live production data to exist in development environments.
- Different RBAC requirements exist for various user groups by IT policy per environment, for example, access is more restrictive in production.
- All data, experimentation, and inferencing happens in a single Azure region.

To adhere to the above requirements, Contoso sets up their resources in the following way:

- Azure Machine Learning workspaces and resource groups scoped per project to follow budgeting and use case segregation requirements.
- A multiple-environment setup for Azure Machine Learning and associated resources to address cost management, RBAC, and data access requirements.
- A single resource group and machine learning workspace that's dedicated for exploration.
- Azure Active Directory groups that are different per user role and environment. For example, operations that a data scientist can do in a production environment are different than in the development environment, and access levels might differ per solution.
- All resources created in a single Azure region.



## Next steps

Learn about best practices on machine learning DevOps with Azure Machine Learning.

[Machine learning DevOps guide](#)

Learn about considerations when managing budgets, quota, and cost with Azure Machine Learning.

[Manage budgets, costs, and quota for Azure Machine Learning at organizational scale](#)

# Azure Machine Learning best practices for enterprise security

Article • 10/18/2023

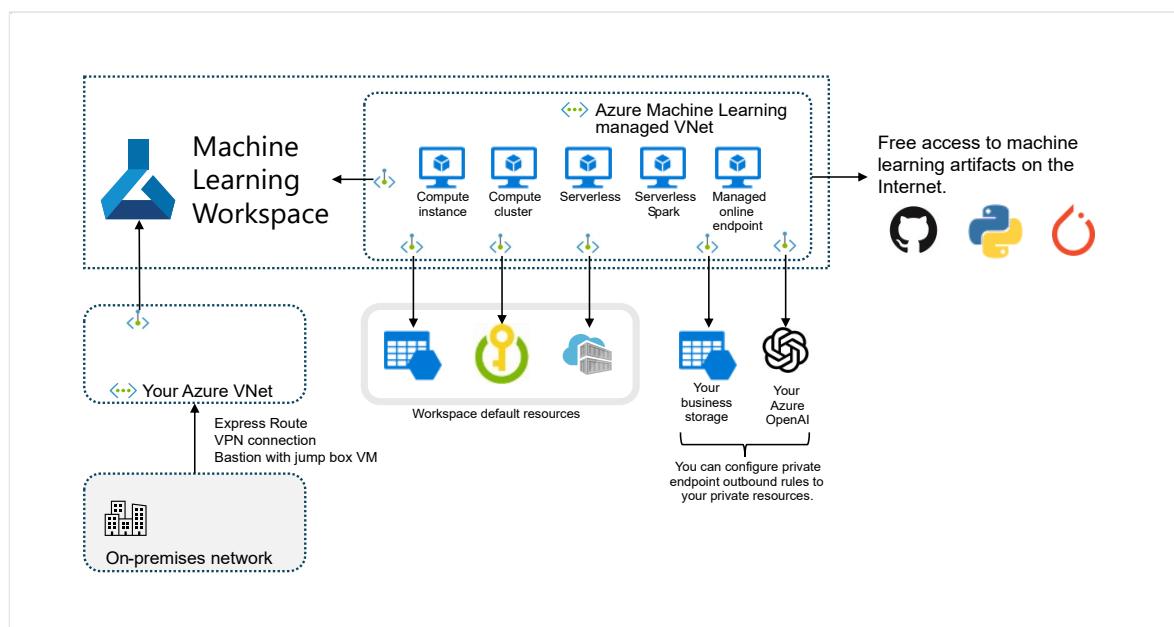
This article explains security best practices for planning or managing a secure Azure Machine Learning deployment. Best practices come from Microsoft and customer experience with Azure Machine Learning. Each guideline explains the practice and its rationale. The article also provides links to how-to and reference documentation.

## Recommended network security architecture (managed network)

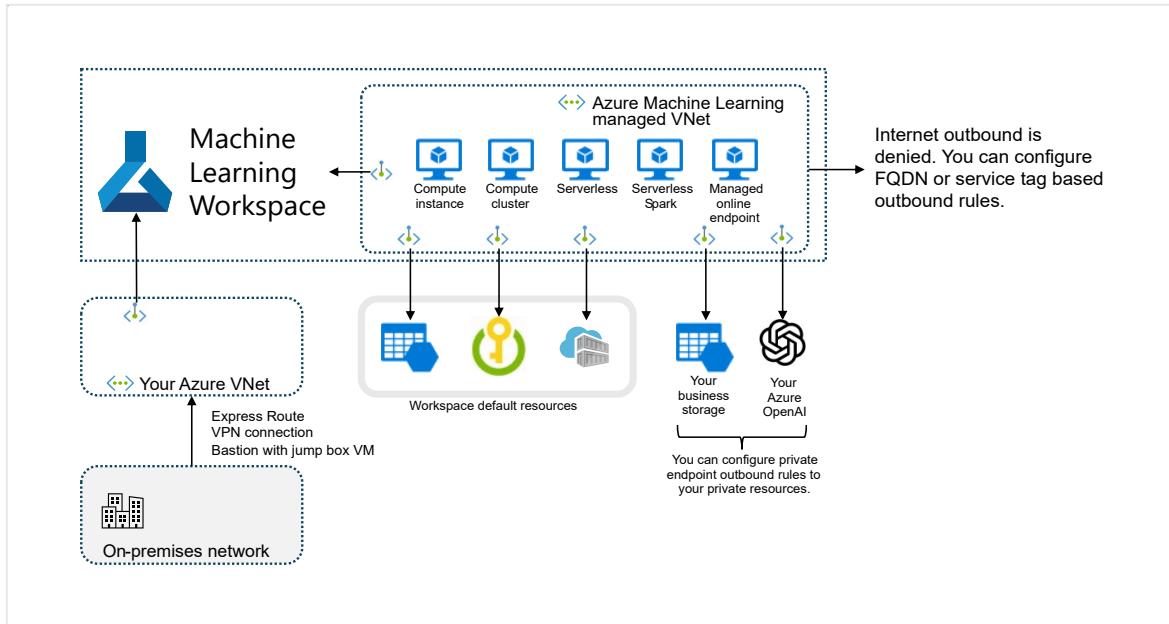
The recommended machine learning network security architecture is a *managed virtual network* (preview). An Azure Machine Learning managed virtual network secures the workspace, associated Azure resources, and all managed compute resources. It simplifies the configuration and management of network security by preconfiguring required outputs and automatically creating managed resources within the network. You can use private endpoints to allow Azure services to access the network and can optionally define outbound rules to allow the network to access the internet.

The managed virtual network has two modes that it can be configured for:

- **Allow internet outbound** - This mode allows outbound communication with resources located on the internet, such as the public PyPi or Anaconda package repositories.



- **Allow only approved outbound** - This mode allows only the minimum outbound communication required for the workspace to function. This mode is recommended for workspaces that must be isolated from the internet. Or where outbound access is only allowed to specific resources via service endpoints, service tags, or fully qualified domain names.

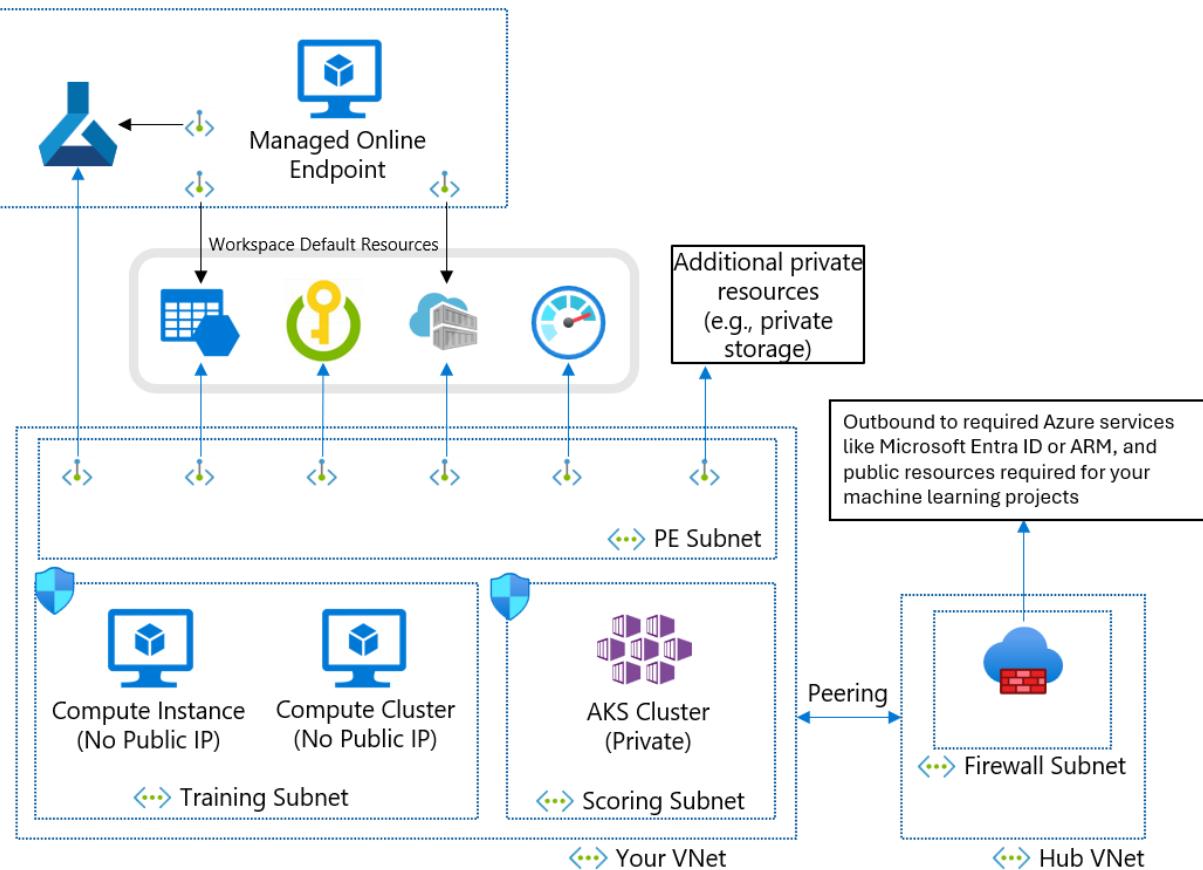


For more information, see [Managed virtual network isolation](#).

## Recommended network security architecture (Azure Virtual Network)

If you can't use a managed virtual network due to your business requirements, you can use an Azure virtual network with the following subnets:

- **Training** contains compute resources used for training, such as machine learning compute instances or compute clusters.
- **Scoring** contains compute resources used for scoring, such as Azure Kubernetes Service (AKS).
- **Firewall** contains the firewall that allows traffic to and from the public internet, such as Azure Firewall.



The virtual network also contains a *private endpoint* for your machine learning workspace and the following dependent services:

- Azure Storage account
- Azure Key Vault
- Azure Container Registry

*Outbound* communication from the virtual network must be able to reach the following Microsoft services:

- Machine learning
- Microsoft Entra ID
- Azure Container Registry, and specific registries that Microsoft maintains
- Azure Front Door
- Azure Resource Manager
- Azure Storage

Remote clients connect to the virtual network using Azure ExpressRoute or a virtual private network (VPN) connection.

## Virtual network and private endpoint design

When designing an Azure Virtual Network, subnets, and private endpoints, consider the following requirements:

- In general, create separate subnets for training and scoring and use the training subnet for all private endpoints.
- For IP addressing, compute instances need one private IP each. Compute clusters need one private IP per node. AKS clusters need many private IP addresses, as described in [Plan IP addressing for your AKS cluster](#). A separate subnet for at least AKS helps prevent IP address exhaustion.
- The compute resources in the training and scoring subnets must access the storage account, the key vault, and the container registry. Create private endpoints for the storage account, the key vault, and the container registry.
- Machine learning workspace default storage needs two private endpoints, one for Azure Blob Storage and another for Azure File Storage.
- If you use Azure Machine Learning studio, the workspace and storage private endpoints should be in the same virtual network.
- If you have multiple workspaces, use a virtual network for each workspace to create an explicit network boundary between workspaces.

## Use private IP addresses

Private IP addresses minimize your Azure resources' exposure to the internet. Machine learning uses many Azure resources, and the machine learning workspace private endpoint isn't enough for end-to-end private IP. The following table shows the major resources machine learning uses and how to enable private IP for the resources.

Compute instances and compute clusters are the only resources that don't have the private IP feature.

Resources	Private IP solution	Documentation
Workspace	Private endpoint	<a href="#">Configure a private endpoint for an Azure Machine Learning workspace</a>
Registry	Private endpoint	<a href="#">Network isolation with Azure Machine Learning registries</a>
Associated resources		
Storage	Private endpoint	<a href="#">Secure Azure Storage accounts with service</a>

Resources	Private IP solution	Documentation
		endpoints
Key Vault	Private endpoint	Secure Azure Key Vault
Container Registry	Private endpoint	Enable Azure Container Registry
<b>Training resources</b>		
Compute instance	Private IP (no public IP)	Secure training environments
Compute cluster	Private IP (no public IP)	Secure training environments
<b>Hosting resources</b>		
Managed online endpoint	Private endpoint	Network isolation with managed online endpoints
Online endpoint (Kubernetes)	Private endpoint	Secure Azure Kubernetes Service online endpoints
Batch endpoints	Private IP (inherited from compute cluster)	Network isolation in batch endpoints

## Control virtual network inbound and outbound traffic

Use a firewall or Azure network security group (NSG) to control virtual network inbound and outbound traffic. For more information on inbound and outbound requirements, see [Configure inbound and outbound network traffic](#). For more information on traffic flows between components, see [Network traffic flow in a secured workspace](#).

## Ensure access to your workspace

To ensure that your private endpoint can access your machine learning workspace, take the following steps:

1. Make sure you have access to your virtual network using a VPN connection, ExpressRoute, or jump box virtual machine (VM) with Azure Bastion access. The public user can't access the machine learning workspace with the private endpoint, because it can be accessed only from your virtual network. For more information, see [Secure your workspace with virtual networks](#).
2. Make sure you can resolve the workspace fully qualified domain names (FQDNs) with your private IP address. If you use your own Domain Name System (DNS) server or a [centralized DNS infrastructure](#), you need to configure a DNS forwarder. For more information, see [How to use your workspace with a custom DNS server](#).

# Workspace access management

When defining machine learning identity and access management controls, you can separate controls that define access to Azure resources from controls that manage access to data assets. Depending on your use case, consider whether to use *self-service*, *data-centric*, or *project-centric* identity and access management.

## Self-service pattern

In a self-service pattern, data scientists can create and manage workspaces. This pattern is best suited for proof-of-concept situations requiring flexibility to try different configurations. The disadvantage is that data scientists need the expertise to provision Azure resources. This approach is less suitable when strict control, resource use, audit traces, and data access are required.

1. Define Azure policies to set safeguards for resource provisioning and usage, such as allowed cluster sizes and VM types.
2. Create a resource group for holding the workspaces and grant data scientists a Contributor role in the resource group.
3. Data scientists can now create workspaces and associate resources in the resource group in a self-service manner.
4. To access data storage, create user-assigned managed identities and grant the identities read-access roles on the storage.
5. When data scientists create compute resources, they can assign the managed identities to the compute instances to gain data access.

For best practices, see [Authentication for cloud-scale analytics](#).

## Data-centric pattern

In a data-centric pattern, the workspace belongs to a single data scientist who might be working on multiple projects. The advantage of this approach is that the data scientist can reuse code or training pipelines across projects. As long as the workspace is limited to a single user, data access can be traced back to that user when auditing storage logs.

The disadvantage is that data access isn't compartmentalized or restricted on a per-project basis, and any user added to the workspace can access the same assets.

1. Create the workspace.

2. Create compute resources with system-assigned managed identities enabled.
3. When a data scientist needs access to the data for a given project, grant the compute managed identity read access to the data.
4. Grant the compute managed identity access to other required resources, such as a container registry with custom Docker images for training.
5. Also grant the workspace's managed identity read-access role on the data to enable data preview.
6. Grant the data scientist access to the workspace.
7. The data scientist can now create data stores to access data required for projects and submit training runs that use the data.

Optionally, create a Microsoft Entra security group and grant it read access to data, then add managed identities to the security group. This approach reduces the number of direct role assignments on resources, to avoid reaching the subscription limit on role assignments.

## Project-centric pattern

A project-centric pattern creates a machine learning workspace for a specific project, and many data scientists collaborate within the same workspace. Data access is restricted to the specific project, making the approach well suited for working with sensitive data. Also, it's straightforward to add or remove data scientists from the project.

The disadvantage of this approach is that sharing assets across projects can be difficult. It's also hard to trace data access to specific users during audits.

1. Create the workspace
2. Identify data storage instances required for the project, create a user-assigned managed identity, and grant the identity read access to the storage.

Optional, grant the workspace's managed identity access to data storage to allow data preview. You can omit this access for sensitive data not suitable for preview.
3. Create credentialless data stores for the storage resources.
4. Create compute resources within the workspace, and assign the managed identity to the compute resources.

5. Grant the compute managed identity access to other required resources, such as a container registry with custom Docker images for training.
6. Grant data scientists working on the project a role on the workspace.

By using Azure role-based access control (RBAC), you can restrict data scientists from creating new datastores or new compute resources with different managed identities. This practice prevents access to data not specific to the project.

Optionally, to simplify project membership management, you can create a Microsoft Entra security group for project members and grant the group access to the workspace.

## Azure Data Lake Storage with credential passthrough

You can use Microsoft Entra user identity for interactive storage access from machine learning studio. Data Lake Storage with hierarchical namespace enabled allows for enhanced organization of data assets for storage and collaboration. With Data Lake Storage hierarchical namespace, you can compartmentalize data access by giving different users access control list (ACL)-based access to different folders and files. For example, you can grant only a subset of users access to confidential data.

## RBAC and custom roles

Azure RBAC helps you manage who has access to machine learning resources and configure who can perform operations. For example, you might want to grant only specific users the workspace administrator role to manage compute resources.

Access scope can differ between environments. In a production environment, you might want to limit the ability of users to update inference endpoints. Instead, you might grant that permission to an authorized service principal.

Machine learning has several default roles: owner, contributor, reader, and data scientist. You can also create your own custom roles, for example to create permissions that reflect your organizational structure. For more information, see [Manage access to Azure Machine Learning workspace](#).

Over time, the composition of your team might change. If you create a Microsoft Entra group for each team role and workspace, you can assign an Azure RBAC role to the Microsoft Entra group, and manage resource access and user groups separately.

User principals and service principals can be part of the same Microsoft Entra group. For example, when you create a user-assigned managed identity that Azure Data Factory

uses to trigger a machine learning pipeline, you might include the managed identity in a **ML pipelines executor** Microsoft Entra group.

## Central Docker image management

Azure Machine Learning provides curated Docker images that you can use for training and deployment. However, your enterprise compliance requirements might mandate using images from a private repository your company manages. Machine learning has two ways to use a central repository:

- Use the images from a central repository as base images. The machine learning environment management installs packages and creates a Python environment where the training or inferencing code runs. With this approach, you can update package dependencies easily without modifying the base image.
- Use the images as-is, without using machine learning environment management. This approach gives you a higher degree of control but also requires you to carefully construct the Python environment as part of the image. You need to meet all the necessary dependencies to run the code, and any new dependencies require rebuilding the image.

For more information, see [Manage environments](#).

## Data encryption

Machine learning data at rest has two data sources:

- Your storage has all your data, including training and trained model data, except for the metadata. You're responsible for your storage encryption.
- Azure Cosmos DB contains your metadata, including run history information like experiment name and experiment submission date and time. In most workspaces, Azure Cosmos DB is in the Microsoft subscription and encrypted by a Microsoft-managed key.

If you want to encrypt your metadata using your own key, you can use a customer-managed key workspace. The downside is that you need to have Azure Cosmos DB in your subscription and pay its cost. For more information, see [Data encryption with Azure Machine Learning](#).

For information on how Azure Machine Learning encrypts data in transit, see [Encryption in transit](#).

# Monitoring

When you deploy machine learning resources, set up logging and auditing controls for observability. Motivations for observing data might vary based on who looks at the data. Scenarios include:

- Machine learning practitioners or operations teams want to **monitor machine learning pipeline health**. These observers need to understand issues in scheduled execution or problems with data quality or expected training performance. You can build Azure dashboards that [monitor Azure Machine Learning data](#) or [create event-driven workflows](#).
- Capacity managers, machine learning practitioners, or operations teams might want to [create a dashboard](#) to **observe compute and quota utilization**. To manage a deployment with multiple Azure Machine Learning workspaces, consider creating a central dashboard to understand quota utilization. Quotas are managed on a subscription level, so the environment-wide view is important to drive optimization.
- IT and operations teams can set up [diagnostic logging](#) to **audit resource access and altering events** in the workspace.
- Consider creating dashboards that **monitor overall infrastructure health** for machine learning and dependent resources such as storage. For example, combining Azure Storage metrics with pipeline execution data can help you optimize infrastructure for better performance or discover problem root causes.

Azure collects and stores platform metrics and activity logs automatically. You can route the data to other locations by using a diagnostic setting. Set up diagnostic logging to a centralized Log Analytics workspace for observability across several workspace instances. Use Azure Policy to automatically set up logging for new machine learning workspaces into this central Log Analytics workspace.

# Azure Policy

You can enforce and audit the usage of security features on workspaces through Azure Policy. Recommendations include:

- Enforce custom-managed key encryption.
- Enforce Azure Private Link and private endpoints.
- Enforce private DNS zones.
- Disable non-Azure AD authentication, such as Secure Shell (SSH).

For more information, see [Built-in policy definitions for Azure Machine Learning](#).

You can also use custom policy definitions to govern workspace security in a flexible manner.

## Compute clusters and instances

The following considerations and recommendations apply to machine learning compute clusters and instances.

### Disk encryption

The operating system (OS) disk for a compute instance or compute cluster node is stored in Azure Storage and encrypted with Microsoft-managed keys. Each node also has a local temporary disk. The temporary disk is also encrypted with Microsoft-managed keys if the workspace was created with the `hbi_workspace = True` parameter. For more information, see [Data encryption with Azure Machine Learning](#).

### Managed identity

Compute clusters support using managed identities to authenticate to Azure resources. Using a managed identity for the cluster allows authentication to resources without exposing credentials in your code. For more information, see [Create an Azure Machine Learning compute cluster](#).

### Setup script

You can use a setup script to automate the customization and configuration of compute instances at creation. As an administrator, you can write a customization script to use when creating all compute instances in a workspace. You can use Azure Policy to enforce the use of the setup script to create every compute instance. For more information, see [Create and manage an Azure Machine Learning compute instance](#).

### Create on behalf of

If you don't want data scientists to provision compute resources, you can create compute instances on their behalf and assign them to the data scientists. For more information, see [Create and manage an Azure Machine Learning compute instance](#).

### Private endpoint-enabled workspace

Use compute instances with a private endpoint-enabled workspace. The compute instance rejects all public access from outside the virtual network. This configuration also prevents packet filtering.

## Azure Policy support

When using an *Azure virtual network*, you can use Azure Policy to ensure that every compute cluster or instance is created in a virtual network and specify the default virtual network and subnet. The policy isn't needed when using a *managed virtual network*, as the compute resources are automatically created in the managed virtual network.

You can also use a policy to disable non-Azure AD authentication, such as SSH.

## Next steps

Learn more about machine learning security configurations:

- [Enterprise security and governance](#)
- [Secure workspace resources using virtual networks](#)

Get started with a machine learning template-based deployment:

- [Azure Quickstart Templates \(microsoft.com\)](#)
- [Enterprise-scale analytics and AI data landing zone](#)

Read more articles about architectural considerations for deploying machine learning:

- Learn how team structure, environment, or regional constraints affect workspace setup.

[Organize and set up Azure Machine Learning environments](#)

- See how to manage compute costs and budget across teams and users.

[Budget, cost, and quota management for Azure Machine Learning at organizational scale](#)

- Learn about machine learning DevOps (MLOps), which uses a combination of people, process, and technology to deliver robust, reliable, and automated machine learning solutions.

[Machine learning DevOps guide](#)

# Manage budgets, costs, and quota for Azure Machine Learning at organizational scale

Article • 01/26/2023

When you manage compute costs incurred from Azure Machine Learning, at an organization scale with many workloads, many teams, and users, there are numerous management and optimization challenges to work through.

In this article, we present best practices to optimize costs, manage budgets, and share quota with Azure Machine Learning. It reflects the experience and lessons learned from running machine learning teams internally at Microsoft and while partnering with our customers. You'll learn how to:

- [Optimize compute resources to meet workload requirements.](#)
- [Drive the best use of a team's budget.](#)
- [Plan, manage and share budgets, cost, and quota at enterprise-scale.](#)

## Optimize compute to meet workload requirements

When you start a new machine learning project, exploratory work might be needed to get a good picture of compute requirements. This section provides recommendations on how you can determine the right virtual machine (VM) SKU choice for training, for inferencing, or as a workstation to work from.

### Determine the compute size for training

Hardware requirements for your training workload might vary from project to project. To meet these requirements, Azure Machine Learning compute [offers various types](#) of VMs:

- **General purpose:** Balanced CPU to memory ratio.
- **Memory optimized:** High memory to CPU ratio.
- **Compute optimized:** High CPU to memory ratio.
- **High performance compute:** Deliver leadership-class performance, scalability, and cost efficiency for various real-world HPC workloads.
- **Instances with GPUs:** Specialized virtual machines targeted for heavy graphic rendering and video editing, as well as model training and inferencing (ND) with

deep learning.

You might not know yet what your compute requirements are. In this scenario, we recommend starting with either of the following cost effective default options. These options are for lightweight testing and for training workloads.

Type	Virtual machine size	Specs
CPU	Standard_DS3_v2	4 cores, 14 gigabytes (GB) RAM, 28-GB storage
GPU	Standard_NC6	6 cores, 56 gigabytes (GB) RAM, 380-GB storage, NVIDIA Tesla K80 GPU

To get the best VM size for your scenario, it might consist of trial and error. Here are several aspects to consider.

- If you need a CPU:
  - Use a [memory optimized](#) VM if you're training on large datasets.
  - Use a [compute optimized](#) VM if you're doing real-time inferencing or other latency sensitive tasks.
  - Use a VM with more cores and RAM in order to speed up training times.
- If you need a GPU, see the [GPU optimized VM sizes](#) for information on selecting a VM.
  - If you're doing distributed training, use VM sizes that have multiple GPUs.
  - If you're doing distributed training on multiple nodes, use GPUs that have NVLink connections.

While you select the VM type and SKU that best fits your workload, evaluate comparable VM SKUs as a trade-off between CPU and GPU performance and pricing. From a cost management perspective, a job might run reasonably well on several SKUs.

Certain GPUs such as the NC family, particularly NC\_Promo SKUs, have similar abilities to other GPUs such as low latency and ability to manage multiple computing workloads in parallel. They're available at discounted prices compared to some of the other GPUs. Considerately selecting VM SKUs to the workload might save cost significantly in the end.

A reminder on the importance for utilization is to sign up for a greater number of GPUs doesn't necessarily execute with faster results. Instead, make sure the GPUs are fully utilized. For example, double check the need for NVIDIA CUDA. While it might be required for high-performance GPU execution, your job might not take a dependency on it.

## Determine the compute size for inference

Compute requirements for inference scenarios differ from training scenarios. Available options differ based on whether your scenario demands offline inference in batch or requires online inference in real time.

For real-time inference scenarios consider the following suggestions:

- Use [profiling capabilities](#) on your model with Azure Machine Learning to determine how much CPU and memory you need to allocate for the model when deploying it as a web service.
- If you're doing real-time inference but don't need high availability, deploy to [Azure Container Instances](#) (no SKU selection).
- If you're doing real-time inference but need high availability, deploy to [Azure Kubernetes Service](#).
  - If you're using traditional machine learning models and receive < 10 queries/second, start with a CPU SKU. F-series SKUs often work well.
  - If you're using deep learning models and receive > 10 queries/second, try a NVIDIA GPU SKU (NCasT4\_v3 often works well) [with Triton](#).

For batch inference scenarios consider the following suggestions:

- When you use Azure Machine Learning pipelines for batch inferencing, follow the guidance in [Determine the compute size for training](#) to choose your initial VM size.
- Optimize cost and performance by scaling horizontally. One of the key methods of optimizing cost and performance is by parallelizing the workload with the help of [parallel run step](#) in Azure Machine Learning. This pipeline step allows you to use many smaller nodes to execute the task in parallel, which allows you to scale horizontally. There's an overhead for parallelization though. Depending on the workload and the degree of parallelism that can be achieved, a parallel run step may or may not be an option.

## Determine the size for compute instance

For interactive development, Azure Machine Learning's compute instance is recommended. The compute instance (CI) offering brings single node compute that's bound to a single user and can be used as a cloud workstation.

Some organizations disallow the use of production data on local workstations, have enforced restrictions to the workstation environment, or restrict the installation of packages and dependencies in the corporate IT environment. A compute instance can be used as a workstation to overcome the limitation. It offers a secure environment with

production data access, and runs on images that come with popular packages and tools for data science pre-installed.

When compute instance is running, user is billed for VM compute, Standard Load Balancer (included lb/outbound rules, and data processed), OS disk (Premium SSD managed P10 disk), temp disk (the temp disk type depends on the VM size chosen), and public IP address. To save costs, we recommend users consider:

- Start and stop the compute instance when it's not in use.
- Work with a sample of your data on a compute instance and scale out to compute clusters to work with your full set of data
- Submit experimentation jobs in *local* compute target mode on the compute instance while developing or testing, or when you switch to shared compute capacity when you submit jobs at full scale. For example, many epochs, full set of data, and hyperparameter search.

If you stop the compute instance, it stops billing for VM compute hours, temp disk, and Standard Load Balancer data processed costs. Note user still pays for OS disk and Standard Load Balancer included lb/outbound rules even when compute instance is stopped. Any data saved on OS disk is persisted through stop and restarts.

## Tune the chosen VM size by monitoring compute utilization

You can view information on your Azure Machine Learning compute usage and utilization via Azure Monitor. You can view details on model deployment and registration, quota details such as active and idle nodes, run details such as canceled and completed runs, and compute utilization for GPU and CPU utilization.

Based on the insights from the monitoring details, you can better plan or adjust your resource usage across the team. For example, if you notice many idle nodes over the past week, you can work with the corresponding workspace owners to update the compute cluster configuration to prevent this extra cost. Benefits of analyzing the utilization patterns can help with forecasting costs and budget improvements.

You can access these metrics directly from the Azure portal. Go to your Azure Machine Learning workspace, and select *Metrics* under the monitoring section on the left panel. Then, you can select details on what you would like to view, such as metrics, aggregation, and time period. For more information, see [Monitor Azure Machine Learning](#) documentation page.

The screenshot shows the Azure Machine Learning Metrics blade. On the left, there's a sidebar with sections like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Events, Settings, Properties, Locks, Monitoring, Alerts, Metrics (which is selected), Diagnostic settings, and Logs. The main area has tabs for New chart, Refresh, Share, Feedback, Line chart, Drill into Logs, New alert rule, Pin to dashboard, and Local Time: Last 24 hours (Automatic). A search bar at the top left says 'Search (Ctrl+)/'. Below the tabs, there are filters for Add metric, Add filter, and Apply splitting. The 'Scope' dropdown is set to 'brendal-test'. The 'Metric Namespace' dropdown is set to 'Machine Learning Serv...'. The 'Metric' dropdown is open, showing a list of metrics under 'MODEL': Model Deploy Failed, Model Deploy Started, Model Deploy Succeeded, Model Register Failed, Model Register Succeeded, and Active Cores. The 'Aggregation' dropdown is also open. The 'Local Time: Last 24 hours (Automatic)' button is visible at the top right.

## Switch between local, single-node, and multi-node cloud compute while you develop

There are varying compute and tooling requirements throughout the machine learning lifecycle. Azure Machine Learning can be interfaced with through an SDK and CLI interface from practically any preferred workstation configuration to meet these requirements.

To save costs and work productively, it's recommended to:

- Clone your experimentation code base locally by using Git and submit jobs to cloud compute using the Azure Machine Learning SDK or CLI.
- If your dataset is large, consider managing a sample of your data on your local workstation, while keeping the full dataset on cloud storage.
- Parameterize your experimentation code base so that you can configure your jobs to run with a varying number of epochs or on datasets of different sizes.
- Don't hard code the folder path of your dataset. You can then easily reuse the same code base with different datasets, and under local and cloud execution context.
- Bootstrap your experimentation jobs in *local* compute target mode while you develop or test, or when you switch to a shared compute cluster capacity when you submit jobs at full scale.
- If your dataset is large, work with a sample of data on your local or compute instance workstation, while scaling to cloud compute in Azure Machine Learning to work with your full set of data.
- When your jobs take a long time to execute, consider optimizing your code base for distributed training to allow for scaling out horizontally.

- Design your distributed training workloads for node elasticity, to allow flexible use of single-node and multi-node compute, and ease usage of compute that can be preempted.

## Combine compute types using Azure Machine Learning pipelines

When you orchestrate your machine learning workflows, you can define a pipeline with multiple steps. Each step in the pipeline can run on its own compute type. This allows you to optimize performance and cost to meet varying compute requirements across the machine learning lifecycle.

## Drive the best use of a team's budget

While budget allocation decisions might be out of the span of control of an individual team, a team is typically empowered to use their allocated budget to their best needs. By trading off job priority versus performance and cost wisely, a team can achieve higher cluster utilization, lower overall cost, and use a larger number of compute hours from the same budget. This can result in enhanced team productivity.

## Optimize the costs of shared compute resources

The key to optimize costs of shared compute resources is to ensure that they're being used to their full capacity. Here are some tips to optimize your shared resource costs:

- When you use compute instances, only turn them on when you have code to execute. Shut them down when they aren't being used.
- When you use compute clusters, set the minimum node count to 0 and the maximum node count to a number that is evaluated based on your budget constraints. Use the [Azure pricing calculator](#) to calculate the cost of full utilization of one VM node of your chosen VM SKU. Autoscaling will scale down all the compute nodes when there's no one using it. It will only scale up to the number of nodes you have budget for. You can configure [autoscaling](#) to scale down all the compute nodes.
- Monitor your resource utilizations such as CPU utilization and GPU utilization when training models. If the resources aren't being fully used, modify your code to better use resources or scale down to smaller or cheaper VM sizes.
- Evaluate whether you can create shared compute resources for your team to avoid computing inefficiencies caused by cluster scaling operations.
- Optimize compute cluster autoscaling timeout policies based on usage metrics.

- Use workspace quotas to control the amount of compute resources that individual workspaces have access to.

## Introduce scheduling priority by creating clusters for multiple VM SKUs

Acting under quota and budget constraints, a team must trade off timely execution of jobs versus cost, to ensure important jobs run timely and a budget is used in the best way possible.

To support best compute utilization, teams are recommended to create clusters of various sizes and with *low priority* and *dedicated* VM priorities. Low-priority computes make use of surplus capacity in Azure and hence come with discounted rates. On the downside, these machines can be preempted anytime a higher priority ask comes in.

Using the clusters of varying size and priority, a notion of scheduling priority can be introduced. For example, when experimental and production jobs compete for the same NC GPU-quota, a production job might have preference to run over the experimental job. In that case, run the production job on the dedicated compute cluster, and the experimental job on the low priority compute cluster. When quota falls short, the experimental job will be preempted in favor of the production job.

Next to VM priority, consider running jobs on various VM SKUs. It might be that a job takes longer to execute on a VM instance with a P40 GPU than on a V100 GPU. However, since V100 VM instances might be occupied or quota fully used, the time to completion on the P40 might still be faster from a job throughput perspective. You might also consider running jobs with lower priority on less performant and cheaper VM instances from a cost management perspective.

## Early-terminate a run when training doesn't converge

When you continuously experiment to improve a model against its baseline, you might be executing various experiment runs, each with slightly different configurations. For one run, you might tweak the input datasets. For another run, you might make a hyperparameter change. Not all changes might be as effective as the other. You detect early that a change didn't have the intended effect on the quality of your model training. To detect if training does not converge, monitor training progress during a run. For example, by logging performance metrics after each training epoch. Consider early terminating the job to free up resources and budget for another trial.

# Plan, manage and share budgets, cost, and quota

As an organization grows its number of machine learning use cases and teams, it requires an increased operating maturity from IT and finance as well as coordination between individual machine learning teams to ensure efficient operations. Company-scale capacity and quota management become important to address scarceness of compute resources and overcome management overhead.

This section discusses best practices for planning, managing, and sharing budgets, cost, and quota at enterprise-scale. It's based on learnings from managing many GPU training resources for machine learning internally at Microsoft.

## Understanding resource spend with Azure Machine Learning

One of the biggest challenges as an administrator for planning compute needs is starting new with no historical information as a baseline estimate. On a practical sense, most projects will start from a small budget as a first step.

To understand where the budget is going, it's critical to know where Azure Machine Learning costs come from:

- Azure Machine Learning only charges for compute infrastructure used and doesn't add a surcharge on compute costs.
- When an Azure Machine Learning workspace is created, there are also a few other resources created to enable Azure Machine Learning: Key Vault, Application Insights, Azure Storage, and Azure Container Registry. These resources are used in Azure Machine Learning and you'll pay for these resources.
- There are costs associated with managed compute such as training clusters, compute instances, and managed inferencing endpoints. With these managed compute resources, there are the following infrastructure costs to account for: virtual machines, virtual network, load balancer, bandwidth, and storage.

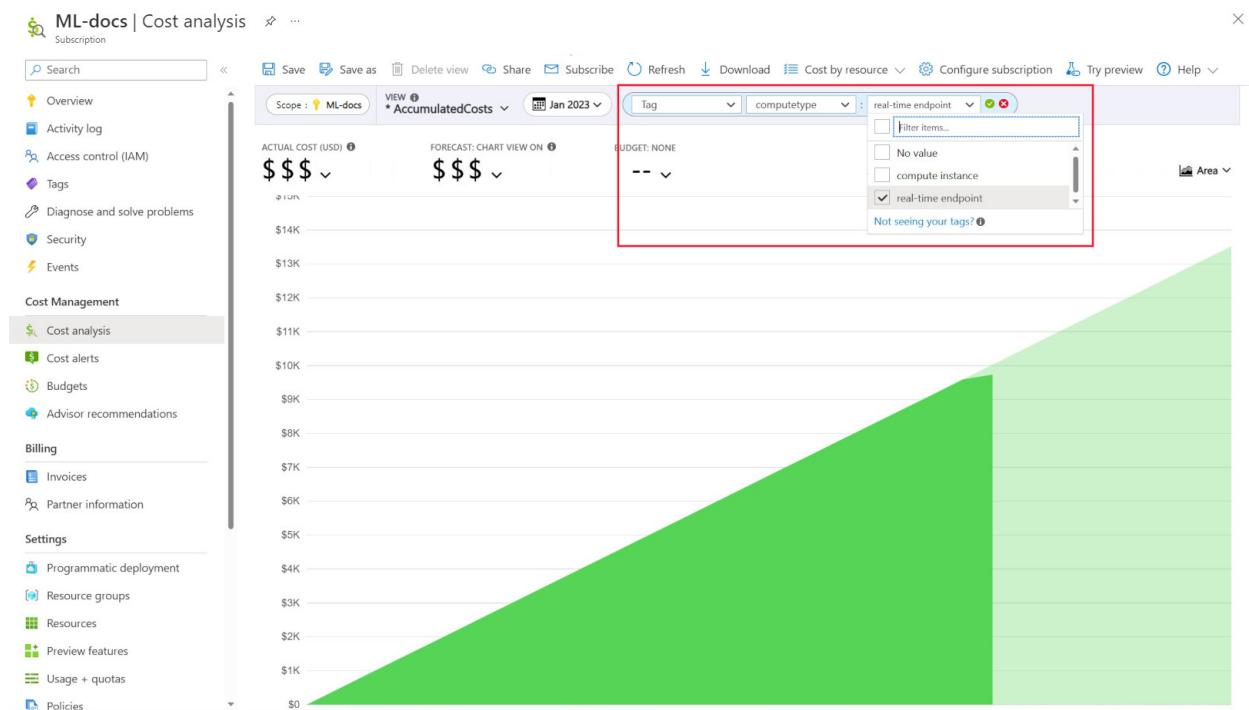
## Track spending patterns and achieve better reporting with tagging

Administrators often want to be able to track costs on different resources in Azure Machine Learning. Tagging is a natural solution to this problem and aligns with the general approach used by Azure and many other cloud service providers. With tags

support, you can now see cost breakdown at the compute level, therefore granting you access to a more granular view to assist with better cost monitoring, improved reporting and greater transparency.

Tagging enables you to place customized tags on your workspaces and computes (from Azure Resource Manager templates and Azure Machine Learning studio) to further filter on these resources in Azure Cost Management based on these tags to observe spend patterns. This functionality can be best utilized for internal charge-back scenarios. In addition, tags can be useful for capturing metadata or details associated with the compute, for e.g. a project, a team, certain billing code, etc. This makes tagging very beneficial for measuring how much money you are spending on different resources and therefore, gaining deeper insights into your cost and spend patterns across teams or projects.

There are also system injected tags placed on computes that allow you to filter in the Cost Analysis page by the “Compute type” tag to see a compute wise breakdown of your total spend and determine what category of compute resources might be attributing to the majority of your costs. This is particularly useful for gaining more visibility into your training vs inferencing cost patterns.



## Govern and restrict compute usage by policy

When you manage an Azure environment with many workloads, it can be a challenge to keep the overview on resource spend. [Azure Policy](#) can help control and govern resource spend, by restricting particular usage patterns across the Azure environment.

In specific for Azure Machine Learning, we recommend setting up policies to allow only for usage of specific VM SKUs. Policies can help prevent and control selection of expensive VMs. Policies can also be used to enforce usage of low-priority VM SKUs.

## Allocate and manage quota based on business priority

Azure allows you to set limits for quota allocation on a subscription and Azure Machine Learning workspace level. Restricting who can manage quota through [Azure role-based access control \(RBAC\)](#) can help ensure resource utilization and cost predictability.

Availability of GPU quota can be scarce across your subscriptions. To ensure high quota utilization across workloads, we recommend monitoring whether quota is best used and assigned across workloads.

At Microsoft, it's determined periodically whether GPU quotas are best used and allocated across machine learning teams by evaluating capacity needs against business priority.

## Commit capacity ahead of time

If you have a good estimate of how much compute will be used in the next year or next few years, you can purchase Azure Reserved VM Instances at a discounted cost. There are one-year or three-year purchase terms. Because Azure Reserved VM Instances are discounted, there can be significant cost savings compared to pay-as-you go prices.

Azure Machine Learning supports reserved compute instances. Discounts are automatically applied against Azure Machine Learning managed compute.

## Manage data retention

Every time a machine learning pipeline is executed, intermediate datasets can be generated at each pipeline step for data caching and reuse. The growth of data as an output of these machine learning pipelines can become a pain point for an organization that is running many machine learning experiments.

Data scientists typically don't spend their time to clean up the intermediate datasets that are generated. Over time, the amount of data that is generated will add up. Azure Storage comes with a capability to enhance the management of the data lifecycle. Using [Azure Blob Storage lifecycle management](#), you can set up general policies to move data that is unused into colder storage tiers and save costs.

# Infrastructure cost optimization considerations

## Networking

Azure networking cost is incurred from outbound bandwidth from Azure datacenter. All inbound data to an Azure datacenter is free. The key to reduce network cost is to deploy all your resources in the same datacenter region whenever possible. If you can deploy Azure Machine Learning workspace and compute in the same region that has your data, you can enjoy lower cost and higher performance.

You might want to have private connection between your on-premises network and your Azure network to have a hybrid cloud environment. ExpressRoute enables you to do that but considering the high cost of ExpressRoute, it might be more cost effective to move away from a hybrid cloud setup and move all resources to Azure cloud.

## Azure Container Registry

For Azure Container Registry, the determining factors for cost optimization include:

- Required throughput for Docker image downloads from the container registry to Azure Machine Learning
- Requirements for enterprise security features, such as Azure Private Link

For production scenarios where high throughput or enterprise security is required, the Premium SKU of Azure Container Registry is recommended.

For dev/test scenarios where throughput and security are less critical, we recommend either Standard SKU or Premium SKU.

The Basic SKU of Azure Container Registry isn't recommended for Azure Machine Learning. It's not recommended because of its low throughput and low included storage, which can be quickly exceeded by Azure Machine Learning's relatively large sized (1+ GB) Docker images.

## Consider computing type availability when choosing Azure regions

When you [pick a region for your compute](#), keep the compute quota availability in mind. Popular and larger regions such as East US, West US, and West Europe tend to have higher default quota values and greater availability of most CPUs and GPUs, compared to some other regions with stricter capacity restrictions in place.

## Learn more

Track costs across business units, environments, or projects by using the Cloud Adoption Framework

## Next steps

To learn more about how to organize and set up Azure Machine Learning environments, see [Organize and set up Azure Machine Learning environments](#).

[Organize and set up Azure Machine Learning environments](#)

To learn about best practices on Machine Learning DevOps with Azure Machine Learning, see [Machine learning DevOps guide](#).

[Machine learning DevOps guide](#)

# Machine learning operations

Article • 09/22/2022

Machine learning operations (also called *MLOps*) is the application of DevOps principles to AI-infused applications. To implement machine learning operations in an organization, specific skills, processes, and technology must be in place. The objective is to deliver machine learning solutions that are robust, scalable, reliable, and automated.

In this article, learn how to plan resources to support machine learning operations at the organization level. Review best practices and recommendations that are based on using Azure Machine Learning to adopt machine learning operations in the enterprise.

## What is machine learning operations?

Modern machine learning algorithms and frameworks make it increasingly easier to develop models that can make accurate predictions. Machine learning operations is a structured way to incorporate machine learning in application development in the enterprise.

In an example scenario, you've built a machine learning model that exceeds all your accuracy expectations and impresses your business sponsors. Now it's time to deploy the model to production, but that might not be as easy as you had expected. The organization likely will need to have people, processes, and technology in place before it can use your machine learning model in production.

Over time, you or a colleague might develop a new model that works better than the original model. Replacing a machine learning model that's used in production introduces some concerns that are important to the organization:

- You'll want to implement the new model without disrupting the business operations that rely on the deployed model.
- For regulatory purposes, you might be required to explain the model's predictions or re-create the model if unusual or biased predictions result from data in the new model.
- The data you use in your machine learning training and model might change over time. With changes in the data, you might need to periodically retrain the model to maintain its prediction accuracy. A person or role will need to be assigned responsibility to feed the data, monitor the model's performance, retrain the model, and fix the model if it fails.

Suppose you have an application that serves a model's predictions via REST API. Even a simple use case like this one might cause problems in production. Implementing a machine learning operations strategy can help you address deployment concerns and support business operations that rely on AI-infused applications.

Some machine learning operations tasks fit well in the general DevOps framework. Examples include setting up unit tests and integration tests and tracking changes by using version control. Other tasks are more unique to machine learning operations and might include:

- Enable continuous experimentation and comparison against a baseline model.
- Monitor incoming data to detect [data drift](#).
- Trigger model retraining and set up a rollback for disaster recovery.
- Create reusable data pipelines for training and scoring.

The goal of machine learning operations is to close the gap between development and production and to deliver value to customers faster. To achieve this goal, you must rethink traditional development and production processes.

Not every organization's machine learning operations requirements are the same. The machine learning operations architecture of a large, multinational enterprise probably won't be the same infrastructure that a small startup establishes. Organizations typically begin small and build up as their maturity, model catalog, and experience grows.

The [machine learning operations maturity model](#) can help you see where your organization is on the machine learning operations maturity scale and help you plan for future growth.

## Machine learning operations vs. DevOps

Machine learning operations is different from DevOps in several key areas. Machine learning operations has these characteristics:

- Exploration precedes development and operations.
- The data science lifecycle requires an adaptive way of working.
- Limits on data quality and availability limit progress.
- A greater operational effort is required than in DevOps.
- Work teams require specialists and domain experts.

For a summary, review the [seven principles of machine learning operations](#).

## Exploration precedes development and operations

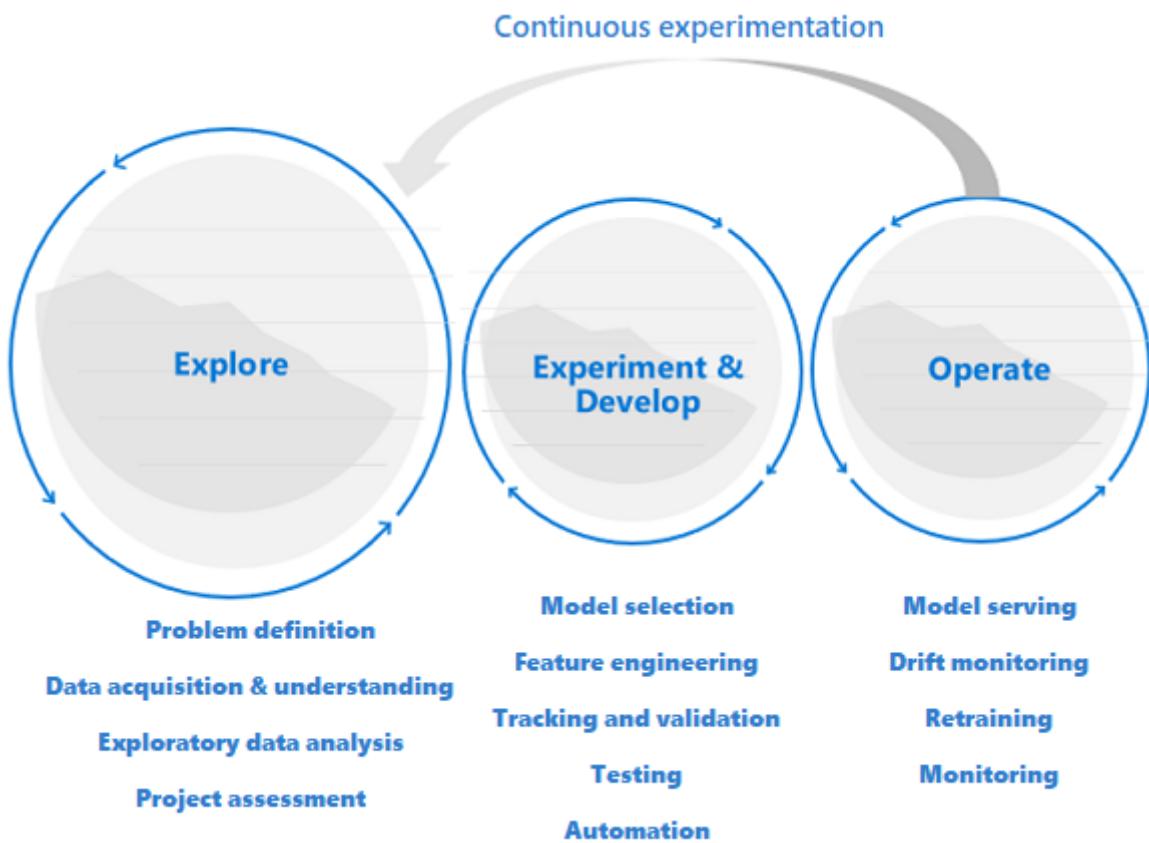
Data science projects are different from application development or data engineering projects. A data science project might make it to production, but often more steps are involved than in a traditional deployment. After an initial analysis, it might become clear that the business outcome can't be achieved with the available datasets. A more detailed exploration phase usually is the first step in a data science project.

The goal of the exploration phase is to define and refine the problem. During this phase, data scientists run exploratory data analysis. They use statistics and visualizations to confirm or falsify the problem hypotheses. Stakeholders should understand that the project might not extend beyond this phase. At the same time, it's important to make this phase as seamless as possible for a quick turnaround. Unless the problem to solve includes a security element, avoid restricting the exploratory phase with processes and procedures. Data scientists should be allowed to work with the tools and data they prefer. Real data is needed for this exploratory work.

The project can move to the experimentation and development stages when stakeholders are confident that the data science project is feasible and can provide real business value. At this stage, development practices become increasingly important. It's a good practice to capture metrics for all of the experiments that are done at this stage. It's also important to incorporate source control so that you can compare models and toggle between different versions of the code.

Development activities include refactoring, testing, and automating exploration code in repeatable experimentation pipelines. The organization must create applications and pipelines to serve the models. Refactoring code in modular components and libraries helps increase reusability, testing, and performance optimization.

Finally, the application or batch inference pipelines that serve the models are deployed to staging or production environments. In addition to monitoring infrastructure reliability and performance like for a standard application, in a machine learning model deployment, you must continuously monitor the quality of the data, the data profile, and the model for degradation or drift. Machine learning models also require retraining over time to stay relevant in a changing environment.



## Data science lifecycle requires an adaptive way of working

Because the nature and quality of data initially is uncertain, you might not accomplish your business goals if you apply a typical DevOps process to a data science project. Exploration and experimentation are recurring activities and needs throughout the machine learning process. Teams at Microsoft use a project lifecycle and a working process that reflect the nature of data science-specific activities. The [Team Data Science Process](#) and The [Data Science Lifecycle Process](#) are examples of reference implementations.

## Limits on data quality and availability limit progress

For a machine learning team to effectively develop machine learning-infused applications, access to production data is preferred for all relevant work environments. If production data access isn't possible due to compliance requirements or technical constraints, consider implementing [Azure role-based access control \(Azure RBAC\)](#) with [Azure Machine Learning](#), [just-in-time access](#), or [data movement pipelines](#) to create production data replicas and enhance user productivity.

## Machine learning requires a greater operational effort

Unlike traditional software, the performance of a machine learning solution is constantly at risk because the solution is dependent on data quality. To maintain a qualitative solution in production, it's critical that you [continuously monitor and reevaluate both data and model quality](#). It's expected that a production model requires timely retraining, redeployment, and tuning. These tasks come on top of day-to-day security, [infrastructure monitoring](#), and compliance requirements, and they require specialized expertise.

## Machine learning teams require specialists and domain experts

Although data science projects share roles with regular IT projects, the success of a machine learning effort highly depends on having essential machine learning technology specialists and domain subject matter experts. A technology specialist has the right background to do end-to-end machine learning experimentation. A domain expert can support the specialist by analyzing and synthesizing data or by qualifying data for use.

Common technical roles that are unique to data science projects are domain expert, data engineer, data scientist, AI engineer, model validator, and machine learning engineer. To learn more about roles and tasks in a typical data science team, see the [Team Data Science Process](#).

## Seven principles of machine learning operations

As you plan to adopt machine learning operations in your organization, consider applying the following core principles as the foundation:

- **Use version control for code, data, and experimentation outputs.** Unlike in traditional software development, data has a direct influence on the quality of machine learning models. You should version your experimentation code base, but also version your datasets to ensure that you can reproduce experiments or inference results. Versioning experimentation outputs like models can save effort and the computational cost of re-creating them.
- **Use multiple environments.** To separate development and testing from production work, [replicate](#) your infrastructure in at least two environments. Access control for users might be different for each environment.
- **Manage your infrastructure and configurations as code.** When you create and update infrastructure components in your work environments, use [infrastructure as code](#), so inconsistencies don't develop in your environments. Manage machine

learning experiment job specifications as code so that you can easily rerun and reuse a version of your experiment in multiple environments.

- **Track and manage machine learning experiments.** Track key performance indicators and other artifacts for your machine learning experiments. When you keep a history of job performance, you can do a quantitative analysis of experimentation success and enhance team collaboration and agility.
- **Test code, validate data integrity, and ensure model quality.** [Test](#) your experimentation code base for correct data preparation and feature extraction functions, data integrity, and model performance.
- **Machine learning continuous integration and delivery.** Use [continuous integration \(CI\)](#) to automate testing for your team. Include model training as part of continuous training pipelines. Include A/B testing as part of your [release](#) to ensure that only a qualitative model is used in production.
- **Monitor services, models, and data.** When you serve models in a machine learning operations environment, it's critical to monitor the services for their infrastructure uptime, compliance, and model quality. [Set up monitoring](#) to identify data and model drift and to understand whether retraining is required. Consider setting up triggers for automatic retraining.

## Best practices from Azure Machine Learning

Azure Machine Learning offers asset management, orchestration, and automation services to help you manage the lifecycle of your machine learning model training and deployment workflows. Review the best practices and recommendations to apply machine learning operations in the resource areas of people, process, and technology, all supported by Azure Machine Learning.

### People

- Work in project teams to best use specialist and domain knowledge in your organization. Set up [Azure Machine Learning workspaces](#) for each project to comply with use case segregation requirements.
- Define a set of responsibilities and tasks as a role so that any team member on a machine learning operations project team can be assigned to and fulfill multiple roles. Use custom roles in Azure to define a set of granular [Azure RBAC operations for Azure Machine Learning](#) that each role can perform.

- Standardize on a project lifecycle and Agile methodology. The [Team Data Science Process](#) provides a reference lifecycle implementation.
- Balanced teams can run all machine learning operations stages, including exploration, development, and operations.

## Process

- Standardize on a code template for code reuse and to accelerate ramp-up time on a new project or when a new team member joins the project. Use [Azure Machine Learning pipelines](#), [job submission scripts](#), and [CI/CD pipelines](#) as a basis for new templates.
- Use version control. Jobs that are submitted from a Git-backed folder [automatically track repo metadata](#) with the job in Azure Machine Learning for reproducibility.
- Use versioning for experiment inputs and outputs for reproducibility. Use [Azure Machine Learning datasets](#), [model management](#), and [environment management](#) capabilities to facilitate versioning.
- Build up a [run history](#) of experiment runs for comparison, planning, and collaboration. Use an experiment-tracking framework like [MLflow](#) to collect metrics.
- Continuously measure and control the quality of your team's work through [CI](#) on the full experimentation code base.
- Terminate training early in the process when a model doesn't converge. Use an experiment-tracking framework and the [run history](#) in Azure Machine Learning to monitor job runs.
- Define an experiment and model management strategy. Consider using a name like *champion* to refer to the current baseline model. A *challenger* model is a candidate model that might outperform the *champion* model in production. Apply tags in Azure Machine Learning to mark experiments and models. In a scenario like sales forecasting, it might take months to determine whether the model's predictions are accurate.
- Elevate [CI](#) for continuous training by including model training in the build. For example, begin model training on the full dataset with each pull request.
- Shorten the time it takes to get feedback on the quality of the machine learning pipeline by running an automated build on a data sample. Use [Azure Machine](#)

Learning pipeline parameters to parameterize input [datasets](#).

- Use [continuous deployment \(CD\) for machine learning models](#) to automate deployment and testing real-time scoring services in your Azure environments.
- In some regulated industries, you might be required to complete model validation steps before you can use a machine learning model in a production environment. Automating validation steps might accelerate time to delivery. When manual review or validation steps are still a bottleneck, consider whether you can certify the automated model validation pipeline. Use resource tags in Azure Machine Learning to indicate asset compliance and candidates for review or as triggers for deployment.
- Don't retrain in production, and then directly replace the production model without doing integration testing. Even though model performance and functional requirements might appear good, among other potential issues, a retrained model might have a larger environment footprint and break the server environment.
- When production data access is available only in production, use [Azure RBAC](#) and [custom roles](#) to give a select number of machine learning practitioners read access. Some roles might need to read the data for related data exploration. Alternatively, make a data copy available in nonproduction environments.
- Agree on naming conventions and tags for Azure Machine Learning [experiments](#) to differentiate retraining baseline machine learning pipelines from experimental work.

## Technology

- If you currently submit jobs via the Azure Machine Learning studio UI or CLI, instead of submitting jobs via the SDK, use the CLI or [Azure DevOps Machine Learning tasks](#) to configure automation pipeline steps. This process might reduce the code footprint by reusing the same job submissions directly from automation pipelines.
- Use event-based programming. For example, trigger an offline model testing pipeline by using Azure Functions after a new model is registered. Or, send a notification to a designated email alias when a critical pipeline fails to run. Azure Machine Learning [creates events in Azure Event Grid](#). Multiple roles can subscribe to be notified of an event.
- When you use Azure DevOps for automation, use [Azure DevOps Tasks for Machine Learning](#) to use machine learning models as pipeline triggers.

- When you develop Python packages for your machine learning application, you can host them in an Azure DevOps repository as artifacts and publish them as a feed. By using this approach, you can [integrate](#) the DevOps workflow for building packages with your Azure Machine Learning workspace.
- Consider using a staging environment to test machine learning pipeline system integration with upstream or downstream application components.
- Create unit and integration tests for your inference endpoints for enhanced debugging and to accelerate time to deployment.
- To trigger retraining, use [dataset monitors](#) and [event-driven workflows](#). Subscribe to data drift events and automate the trigger of [machine learning pipelines for retraining](#).

## AI factory for organization machine learning operations

A data science team might decide it can manage multiple machine learning use cases internally. Adopting machine learning operations helps an organization set up project teams for better quality, reliability, and maintainability of solutions. Through balanced teams, supported processes, and technology automation, a team that adopts machine learning operations can scale and focus on developing new use cases.

As the number of use cases grows in an organization, the management burden of supporting the use cases grows linearly, or even more. The challenge for the organization becomes how to accelerate time to market, support quicker assessment of use case feasibility, implement repeatability, and best use available resources and skill sets on a range of projects. For many organizations, developing an AI factory is the solution.

An AI factory is a system of repeatable business processes and standardized artifacts that facilitates developing and deploying a large set of machine learning use cases. An AI factory optimizes team setup, recommended practices, machine learning operations strategy, architectural patterns, and reusable templates that are tailored to business requirements.

A successful AI factory relies on repeatable processes and reusable assets to help the organization efficiently scale from tens of use cases to thousands of use cases.

The following figure summarizes key elements of an AI factory:

	Governance	Assets	MLOps	Operations	
Required	AI Factory Team setup Roles & Responsibilities Project Team setup	Reference Architectures Playbook / Document Project templates	Infrastructure as Code Model management Continuous Integration Continuous Delivery Code Repository	Logging & Monitoring Dashboards & Reports Data Drift & Retraining	
Recommended	Ethical Framework Security Cost Management	Shared libraries / packages Central hub Readiness assessments How to videos	ML Engineering team Testing approach Recommended tools Branching	Retrospectives Revisions to the assets Enablement plans by role	

## Standardize on repeatable architectural patterns

Repeatability is a key characteristic of an AI factory. Data science teams can accelerate project development and improve consistency across projects by developing a few repeatable architectural patterns that cover most of the machine learning use cases for their organization. When these patterns are in place, most projects can use the patterns to get the following benefits:

- Accelerated design phase
- Accelerated approvals from IT and security teams when they reuse tools across projects
- Accelerated development due to reusable infrastructure as code templates and project templates

The architectural patterns can include but aren't limited to the following topics:

- Preferred services for each stage of the project
- Data connectivity and governance
- A machine learning operations strategy tailored to the requirements of the industry, business, or data classification
- Experiment management champion and challenger models

## Facilitate cross-team collaboration and sharing

Shared code repositories and utilities can accelerate the development of machine learning solutions. Code repositories can be developed in a modular way during project development so that they're generic enough to be used in other projects. They can be made available in a central repository that all data science teams can access.

## Share and reuse intellectual property

To maximize code reuse, review the following intellectual property at the beginning of a project:

- Internal code that was designed to reuse in the organization. Examples include packages and modules.
- Datasets that were created in other machine learning projects or which are available in the Azure ecosystem.
- Existing data science projects that have a similar architecture and business problems.
- GitHub or open source repositories that can accelerate the project.

Any project retrospective should include an action item to determine whether elements of the project can be shared and generalized for broader reuse. The list of assets the organization can share and reuse expands over time.

To help with sharing and discovery, many organizations have introduced shared repositories to organize code snippets and machine learning artifacts. Artifacts in Azure Machine Learning, including [datasets](#), [models](#), [environments](#), and [pipelines](#), can be defined as code, so you can share them efficiently across projects and workspaces.

## Project templates

To accelerate the process of migrating existing solutions and to maximize code reuse, many organizations standardize on a project template to kickstart new projects.

Examples of project templates that are recommended for use with Azure Machine Learning are [Azure Machine Learning examples](#), the [Data Science Lifecycle Process](#), and the [Team Data Science Process](#).

## Central data management

The process of getting access to data for exploration or production usage can be time consuming. Many organizations centralize data management to bring together data producers and data consumers for easier access to data for machine learning experimentation.

## Shared utilities

Your organization can use enterprise-wide centralized dashboards to consolidate logging and monitoring information. The dashboards might include error logging, service availability and telemetry, and model performance monitoring.

Use Azure Monitor metrics to build a dashboard for Azure Machine Learning and associated services like Azure Storage. A dashboard helps you keep track of experimentation progress, compute infrastructure health, and GPU quota utilization.

## Specialist machine learning engineering team

Many organizations have implemented the role of machine learning engineer. A machine learning engineer specializes in creating and running robust machine learning pipelines, drift monitoring and retraining workflows, and monitoring dashboards. The engineer has overall responsibility for industrializing the machine learning solution, from development to production. The engineer works closely with data engineering, architects, security, and operations to ensure that all necessary controls are in place.

Although data science requires deep domain expertise, machine learning engineering is more technical in focus. The difference makes the machine learning engineer more flexible, so they can work on various projects and with various business departments. Large data science practices might benefit from a specialist machine learning engineering team that drives repeatability and reuse of automation workflows across various use cases and business areas.

## Enablement and documentation

It's important to provide clear guidance about the AI factory process for new and existing teams and users. Guidance helps ensure consistency and reduce the effort that's required from the machine learning engineering team when it industrializes a project. Consider designing content specifically for the various roles in your organization.

Everyone has a unique way of learning, so a mixture of the following types of guidance can help accelerate adoption of the AI factory framework:

- A central hub that has links to all artifacts. For example, this hub might be a channel on Microsoft Teams or a Microsoft SharePoint site.
- Training and an enablement plan designed for each role.
- A high-level summary presentation of the approach and a companion video.
- A detailed document or playbook.
- How-to videos.
- Readiness assessments.

## Machine learning operations in Azure video series

A video series about [machine learning operations in Azure](#) shows you how to establish machine learning operations for your machine learning solution, from initial development to production.

## Ethics

Ethics plays an instrumental role in the design of an AI solution. If ethical principles aren't implemented, trained models might exhibit the same bias that's present in the data they were trained on. The result might be that the project is discontinued. More importantly, the organization's reputation might be at risk.

To ensure that the key ethical principles that the organization stands for are implemented across projects, the organization should provide a list of these principles and ways to validate them from a technical perspective during the testing phase. Use the machine learning features in Azure Machine Learning to understand what responsible machine learning is and how to build it into your machine learning operations.

## Next steps

Learn more about how to organize and set up Azure Machine Learning environments, or watch a hands-on video series about [machine learning operations in Azure](#).

[Organize and set up Azure Machine Learning environments](#)

Learn more about how to manage budgets, quotas, and costs at the organization level by using Azure Machine Learning:

[Manage machine learning budgets, costs, and quotas with Azure Machine Learning](#)

# Best practices for data science projects with cloud-scale analytics in Azure

Article • 05/07/2024

We recommend these best practices for using cloud-scale analytics in Microsoft Azure to operationalize data science projects.

## Develop a template

Develop a template that bundles a set of services for your data science projects. Use a template that bundles a set of services to help provide consistency across various data science teams' use cases. We recommend that you develop a consistent blueprint in the form of a template repository. You can use this repository for various data science projects within your enterprise to help shorten deployment times.

## Guidelines for data science templates

Develop a data science template for your organization with the following guidelines:

- Develop a set of infrastructure as code (IaC) templates to deploy an Azure Machine Learning workspace. Include resources like a key vault, a storage account, a container registry, and Application Insights.
- Include the setup of data stores and compute targets in these templates, like compute instances, compute clusters, and Azure Databricks.

## Deployment best practices

### Real-time

- Include an Azure Data Factory or Azure Synapse deployment in templates and Azure Cognitive Services.
- The templates should provide all necessary tools to execute the data science exploration phase and the initial operationalization of the model.

### Considerations for an initial setup

In some cases, data scientists in your organization might require an environment for quick as-needed analysis. This situation is common when a data science project isn't formally set up. For example, a project manager, cost code, or cost center that might be required for cross-charging within Azure might be missing because the missing element needs approval. Users in your organization or team might need to access a data science environment to understand the data and possibly evaluate a project's feasibility. Also, some projects might not require a full data science environment because of the small number of data products.

In other cases, a full data science project might be required, complete with a dedicated environment, project management, cost code, and cost center. Full data science projects are useful for multiple team members who want to collaborate, share results, and need to operationalize models after the exploration phase succeeds.

## The setup process

Templates should be deployed on a per-project basis after they've been set up. Each project should receive at least two instances for development and production environments to be separated. In the production environment, no individual person should have access, and everything should be deployed through continuous integration or continuous development pipelines and a service principal. These production environment principles are important because Azure Machine Learning doesn't provide a granular role-based access control model within a workspace. You can't limit user access to a specific set of experiments, endpoints, or pipelines.

The same access rights typically apply to different types of artifacts. It's important to separate development from production to prevent the deletion of production pipelines or endpoints within a workspace. Along with the template, a process needs to be built to give data product teams the option to request new environments.

We recommend setting up different AI services like Azure Cognitive Services on a per-project basis. By setting up different AI services on a per-project basis, deployments occur for each data product resource group. This policy creates a clear separation from a data access standpoint and mitigates the risk of unauthorized data access by the wrong teams.

## Streaming scenario

For real-time and streaming use cases, deployments should be tested on a downsized [Azure Kubernetes Service \(AKS\)](#). The testing can be in the development environment to save on costs before you deploy to the production AKS or Azure App Service for

containers. You should perform simple input and output tests to make sure that the services respond as expected.

Next, you can deploy models to the service you want. This deployment compute target is the only one that's generally available and recommended for production workloads in an AKS cluster. This step is more necessary if graphics processing unit (GPU) or field-programmable gate array support is required. Other native deployment options that support these hardware requirements aren't currently available in Azure Machine Learning.

Azure Machine Learning requires one-to-one mapping to AKS clusters. Every new connection to an Azure Machine Learning workspace breaks the previous connection between AKS and Azure Machine Learning. Once that limitation is mitigated, we recommend deploying central AKS clusters as shared resources and attach them to their respective workspaces.

Another central test AKS instance should be hosted if stress tests should be done before you move a model to the production AKS. The test environment should provide the same compute resource as the production environment to ensure that the results are as similar as possible to the production environment.

## Batch scenario

Not all use cases need AKS cluster deployments. A use case doesn't need an AKS cluster deployment if large data amounts only need scoring regularly or are based on an event. For example, large data amounts can be based on when data drops into a specific storage account. Azure Machine Learning pipelines and Azure Machine Learning compute clusters should be used for deployment during these types of scenarios. These pipelines should be orchestrated and executed in Data Factory.

## Identify the right compute resources

Before you deploy a model in Azure Machine Learning to an AKS, the user needs to specify the resources like CPU, RAM, and GPU that should be allocated for the respective model. Defining these parameters can be a complex and tedious process. You need to do stress tests with different configurations to identify a good set of parameters. You can simplify this process with the **Model Profiling** feature in Azure Machine Learning, which is a long-running job that tests different resource allocation combinations and uses an identified latency and round trip time (RTT) to recommend an optimal combination. This information can assist the actual model deployment on AKS.

To safely update models in Azure Machine Learning, teams should use the controlled rollout feature (preview) to minimize downtime and keep the model's REST endpoint consistent.

## Best practices and the workflow for MLOps

### Include sample code in data science repositories

You can simplify and accelerate data science projects if your teams have certain artifacts and best practices. We recommend creating artifacts that all data science teams can use while working with Azure Machine Learning and the data product environment's respective tools. Data and machine learning engineers should create and provide the artifacts.

These artifacts should include:

- Sample notebooks that show how to:
  - Load, mount, and work with data products.
  - Log metrics and parameters.
  - Submit training jobs to compute clusters.
- Artifacts required for operationalization:
  - Sample Azure Machine Learning pipelines
  - Sample Azure Pipelines
  - More scripts required to execute pipelines
- Documentation

### Use well-designed artifacts to operationalize pipelines

Artifacts can speed up data science projects' exploration and operationalization phases. A DevOps forking strategy can help to scale these artifacts across all projects. Since this setup promotes the use of Git, users and the overall automation process can benefit from the provided artifacts.

#### 💡 Tip

Azure Machine Learning sample pipelines should be built with the Python software developer kit (SDK) or based on the YAML language. The new YAML experience will be more future-proof, as the Azure Machine Learning product team is currently working on a new SDK and command line interface (CLI). The Azure Machine

Learning product team is confident that YAML will serve as the definition language for all artifacts within Azure Machine Learning.

Sample pipelines don't work out of the box for each project, but they can be used as a baseline. You can adjust sample pipelines for projects. A pipeline should include the most relevant aspects of each project. For example, a pipeline can reference a compute target, reference data products, define parameters, define inputs, and define the execution steps. The same process should be done for Azure Pipelines. Azure Pipelines should also use the Azure Machine Learning SDK or CLI.

Pipelines should demonstrate how to:

- Connect to a workspace from within a DevOps pipeline.
- Check whether the required compute is available.
- Submit a job.
- Register and deploy a model.

Artifacts aren't suited for all projects all the time and might require customization, but having a foundation can speed up a project's operationalization and deployment.

## Structure the MLOps repository

You might have situations where users lose track of where they can find and store artifacts. To avoid these situations, you should request more time to communicate and construct a top-level folder structure for the standard repository. All projects should follow the folder structure.

### Note

The concepts mentioned in this section can be used across on-premises, Amazon Web Services, Palantir, and Azure environments.

The proposed top-level folder structure for a MLOps (machine learning operations) repository is illustrated in the following diagram:



# Data Domain 1 - MLOps

main	
	.cloud
	.ado/.github
	code
	docs
	pipelines
	tests
	notebooks

The following purposes apply to each folder in the repository:

[\[+\] Expand table](#)

Folder	Purpose
.cloud	Store cloud-specific code and artifacts in this folder. The artifacts include configuration files for the Azure Machine Learning workspace, including compute target definitions, jobs, registered models, and endpoints.
.ado/.github	Store Azure DevOps or GitHub artifacts like YAML pipelines or code owners in this folder.
code	Include the actual code that's developed as part of the project in this folder. This folder can contain Python packages and some scripts that are used for the respective steps of the machine learning pipeline. We recommend separating individual steps that need to be done in this folder. Common steps are <b>preprocessing</b> , <b>model training</b> , and <b>model registration</b> . Define dependencies like Conda dependencies, Docker images, or others for each folder.

Folder	Purpose
docs	Use this folder for documentation purposes. This folder stores Markdown files and images to describe the project.
pipelines	Store Azure Machine Learning pipelines definitions in YAML or Python in this folder.
tests	Write unit and integration tests that need to be executed to discover bugs and issues early during the project in this folder.
notebooks	Separate Jupyter notebooks from the actual Python project with this folder. Inside the folder, each individual should have a subfolder to check in their notebooks and prevent Git merge conflicts.

## Next step

[Cloud-scale analytics data products in Azure](#)

---

## Feedback

Was this page helpful?

 Yes

 No

# ARM template documentation

Azure Resource Manager templates are JavaScript Object Notation (JSON) files that define the infrastructure and configuration for your project.

## About ARM templates

### OVERVIEW

[What are templates?](#)

### CONCEPT

[Best practices](#)

[Frequently asked questions](#)

[Template specs](#)

[Deployment modes](#)

[Linked templates](#)

### VIDEO

[Build 2020 presentation ↗](#)

## Get started

### QUICKSTART

[Create JSON templates - VS Code](#)

[Create JSON templates - portal](#)

[Create & deploy template specs](#)

### TUTORIAL

[Beginner template tutorials](#)

### TRAINING

## Develop templates

### TUTORIAL

[Advanced templates](#)

[Template specs](#)

### HOW-TO GUIDE

[Use functions](#)

[Define parameters](#)

[Define variables](#)

[Define resources](#)

[Define outputs](#)

### REFERENCE

[Template file syntax](#)

[Azure Quickstart templates](#) ↗

## Deploy templates

### TUTORIAL

[Deployment](#)

### HOW-TO GUIDE

[PowerShell](#)

[Azure CLI](#)

[Portal](#)

[REST API](#)

[Deploy to Azure button](#)

[Cloud Shell](#)

[What-if deployment](#)

## Scoped deployments

---



[HOW-TO GUIDE](#)

[Resource group](#)

[Subscription](#)

[Management group](#)

[Tenant](#)

[Scoping extension resource](#)

[Template functions across scopes](#)

## Explore reference content

---



[Template reference](#)

[REST API](#)

[Azure PowerShell](#)

[Azure CLI](#)

[.NET](#)

[Java](#)

[Python](#)

## Manage templates

---



[Test toolkit](#)

[Export templates](#)

CI/CD

[View deployment history](#)

# Azure Blob Storage documentation

Azure Blob Storage is Microsoft's object storage solution for the cloud. Blob storage is optimized for storing massive amounts of unstructured data.

## About Blob storage

### OVERVIEW

[What is Azure Blob Storage?](#)

## Architecture

### ARCHITECTURE

[Azure Well-Architected Framework considerations](#)

## Get started

### QUICKSTART

[Upload, download, and list blobs - portal](#)

[Use Storage Explorer to manage blobs](#)

### CONCEPT

[Storage account overview](#)

[Data redundancy](#)

### HOW-TO GUIDE

[Create a storage account](#)

[Create a block blob storage account](#)

[Configure a custom domain](#)

## Search and understand blob data

### CONCEPT

[Use AI to understand blob data](#)

[Search your blob content](#)

## Analytics and insights

### OVERVIEW

[Azure Data Lake Storage](#)

### QUICKSTART

[Analyze data with Databricks](#)

### CONCEPT

[Processing big data](#)

[Multi-protocol access](#)

[Best practices](#)

### TUTORIAL

[Extract, transform, and load data with Azure Databricks](#)

[Extract, transform, load data with Apache Hive on Azure HDInsight](#)

[Insert data into Databricks tables by using Event Grid](#)

### HOW-TO GUIDE

[Use with Azure HDInsight clusters](#)

[Use with Power BI](#)

## Develop applications

## QUICKSTART

- [Azure Blob Storage for .NET](#)
- [Azure Blob Storage for Java](#)
- [Azure Blob Storage for Spring](#)
- [Azure Blob Storage for Python](#)
- [Azure Blob Storage for JavaScript using Node.js](#)
- [Azure Blob Storage for JavaScript in a browser](#)
- [Route events to a web endpoint](#)
- [Host a static website](#)

---

## CONCEPT

- [Handle events](#)
- [Static websites](#)

---

## TUTORIAL

- [Create a web app to upload image data](#)

## Manage security and identity

---

### CONCEPT

- [Authorize access to Azure Storage](#)
- [Data Lake access control](#)

---

### HOW-TO GUIDE

- [Access blob data with Microsoft Entra account](#)
- [Manage access rights with Azure RBAC](#)

## Transfer data

---

### CONCEPT

[Compare data transfer solutions](#)

[Tape migration overview](#)

[Use Azure Data Box for offline transfer](#)

---

 **HOW-TO GUIDE**

[Transfer with AzCopy](#)

[Access Azure blob data on Linux with BlobFuse2](#)

[Transfer with Azure Data Factory](#)

## Performance and scalability

---

 **CONCEPT**

[Scalability and performance](#)

[Performance and scalability checklist](#)

---

 **TUTORIAL**

[Optimize for performance and scalability](#)

## Back up and archive data

---

 **CONCEPT**

[Hot, cool, and archive storage](#)

[Manage the Blob storage lifecycle](#)

# Azure Cosmos DB documentation

Fully managed, distributed NoSQL, relational, and vector database for modern app development. High performance, high availability, and support for open-source PostgreSQL, MongoDB, and Apache Cassandra. Build cloud-native apps effortlessly.



CONCEPT

[Introduction to Azure Cosmos DB](#)



GET STARTED

[Try Azure Cosmos DB free](#)



CONCEPT

[Distributed NoSQL databases](#)



REFERENCE

[Microsoft Questions and Answers](#)



CONCEPT

[Vector Database in Azure Cosmos DB](#)



CONCEPT

[Choose an API in Azure Cosmos DB](#)



HOW-TO GUIDE

[Create an Azure Cosmos DB for NoSQL account](#)



HOW-TO GUIDE

[Create an Azure Cosmos DB for PostgreSQL account](#)

## APIs

### NoSQL

[SQL queries](#)

[Model document data](#)

[See more >](#)

### MongoDB

[Supported features and syntax](#)

[Mapping consistency levels](#)

[Extension commands](#)

[See more >](#)

### Apache Cassandra

[Wire protocol support](#)

[Partitioning](#)

[Secondary indexing](#)

[See more >](#)

## Apache Gremlin

Wire protocol support  
Graph data modeling  
Import graph data  
[See more >](#)

## Table

Build apps with API for Table  
Find request unit charge  
FAQ  
[See more >](#)

## PostgreSQL

Distributed relational databases  
Connect and run SQL commands using Python  
Distribute and modify tables  
[See more >](#)

# Concepts

## Explore core concepts

Resource model  
Distribute data globally  
Partitioning  
Throughput & request units

## Model your data

Data modeling and partitioning  
Azure Cosmos DB service quotas

## Analytics and BI

Analytics and BI overview  
Analytics and BI use cases  
Mirroring in Microsoft Fabric  
API for NoSQL mirroring tutorial in Microsoft Fabric  
Azure Synapse Link for Azure Cosmos DB

# Training & certification

## Create an API for NoSQL application

Build a .NET app  
Build a Node.js app  
Build a Java app  
[See more >](#)

## Work with API for PostgreSQL

Model data  
Ingest and query data  
Extend PostgreSQL functionality using extensions  
[See more >](#)

## Certifications

Microsoft Certified - Azure Cosmos DB Developer Specialty  
Microsoft Certified - Azure Developer Associate

# Data Migration

## Migration options

Cassandra Query Language (CQL) shell, Spark – API for Cassandra

## Migration guides

Oracle Database to Azure Cosmos DB API for NoSQL using Striim

Azure Database Migration  
Service – API for MongoDB

Oracle Database to Azure  
Cosmos DB API for Cassandra  
using Striim

Oracle Database to Azure  
Cosmos DB API for Cassandra  
using Arcion

Apache Cassandra to Azure  
Cosmos DB using Arcion

Migrate hundreds of terabytes  
of data

# Azure Database for MySQL documentation

Azure Database for MySQL is a relational database service powered by the MySQL community edition. You can use Azure Database for MySQL to host a MySQL database in Azure. It's a fully managed database as a service offering that can handle mission-critical workloads with predictable performance and dynamic scalability.

## About Azure Database for MySQL

### OVERVIEW

[What is Azure Database for MySQL?](#)

[Get started for free with an Azure free account](#)

### WHAT'S NEW

[Check out our blog ↗](#)

## Azure Database for MySQL deployment model

### OVERVIEW

[Azure Database for MySQL deployment model](#)

### WHAT'S NEW

[What's new in Azure Database for MySQL?](#)

### CONCEPT

[Server concepts](#)

[Understand compute and storage](#)

[Limitations](#)

### QUICKSTART

[Create a server using the portal](#)

## Connect and query



### HOW-TO GUIDE

[Connect and query reference guide](#)



### QUICKSTART

[Connect using MySQL Workbench](#)

[Connect using PHP](#)

[Connect using Azure Data Studio](#)

[Connect using Python](#)

## Migrations



### HOW-TO GUIDE

[Azure Database for MySQL migration guide](#)

[Dump and restore](#)

[Import and export](#)

[Migrate Amazon RDS for MySQL with MySQL Workbench](#)

[Migrate Amazon RDS for MySQL using data-in replication](#)

[Minimal-downtime migration](#)

[Common errors during or post migration](#)

## Manage and migrate data



### CONCEPT

[Manage Azure Database for MySQL using Azure portal](#)

[Migrate using dump and restore](#)

## Migrate using import and export

---

### HOW-TO GUIDE

[Migrate RDS MySQL using MySQL Workbench](#)

[Migrate using Data Migration Service](#)

[Azure Data migration guide](#)

[Troubleshoot migration errors](#)

## Application Development

---

### TUTORIAL

[Build a PHP \(Laravel\) web app with Azure Database for MySQL](#)

[Create a Web App with Azure Database for MySQL in VNET](#)

[Deploy Java Spring Boot app on AKS with Azure Database for MySQL](#)

[Deploy WordPress on App Service with Azure Database for MySQL](#)

## Troubleshoot

---

### HOW-TO GUIDE

[Troubleshoot migration errors](#)

[Troubleshoot query performance](#)

[Troubleshoot low memory issues](#)

[Troubleshoot high CPU utilization](#)

[Troubleshoot replication latency](#)

[Troubleshoot database corruption](#)

[Troubleshoot connectivity issues](#)

[Tune performance using the sys\\_schema](#)

[Troubleshooting best practices](#)

## Reference

---

### DOWNLOAD

[MySQL Workbench ↗](#)

[MySQL .NET Connector ↗](#)

---

### DEPLOY

[Azure CLI Samples - Azure Database for MySQL](#)

[Azure CLI Samples - Azure Database for MySQL - Single Server](#)

[Azure Resource Manager templates](#)

---

### REFERENCE

[Azure CLI developer reference](#)

[REST API developer reference](#)

[PowerShell](#)

# Azure Data Explorer documentation

Azure Data Explorer is a fast, fully managed data analytics service for real-time analysis on large volumes of data streaming from applications, websites, IoT devices, and more. You can use Azure Data Explorer to collect, store, and analyze diverse data to improve products, enhance customer experiences, monitor devices, and boost operations.

## About Azure Data Explorer

### OVERVIEW

[What is Azure Data Explorer?](#)

[How Azure Data Explorer works](#)

### GET STARTED

[Start for free](#)

[Kusto Detective Agency](#) ↗

[Find a partner](#)

### VIDEO

[What is Azure Data Explorer? \(19:47\)](#) ↗

## Training

### TRAINING

[Free Pluralsight training: Azure Data Explorer](#) ↗

[Training module: Introduction to Azure Data Explorer](#)

[Training module: Explore the fundamentals of data analysis using Kusto Query Language \(KQL\)](#)

[Training module: Write your first query with Kusto Query Language](#)

[Training module: Gain insights from your data by using Kusto Query Language](#)

[Training module: Multi-table queries using Kusto Query Language](#)

[Training module: Characterize an unfamiliar dataset with Azure Data Explorer](#)

[Training module: Create dashboards in Azure Data Explorer](#)

[Learning path: Data analysis in Azure Data Explorer with Kusto Query Language](#)

## Create cluster and database

### QUICKSTART

[Create a cluster and database](#)

### HOW-TO GUIDE

[Python](#)

[Azure Resource Manager template](#)

[C#](#)

[PowerShell](#)

[Azure CLI](#)

## Ingest data into Azure Data Explorer

### CONCEPT

[Data ingestion overview](#)

### QUICKSTART

[Get data from a local file](#)

### HOW-TO GUIDE

[Ingestion wizard](#)

[Ingest historical data](#)

[Streaming ingestion](#)

[Python](#)

[Event hub using Azure portal](#)

[Event Grid using Azure portal](#)

## Query data

### QUICKSTART

[Query data with the Azure Data Explorer web UI](#)

### TUTORIAL

[Write KQL queries](#)

### HOW-TO GUIDE

[Query data in Azure Data Lake](#)

[Query data using T-SQL](#)

[Query data using Python](#)

[Time series analysis](#)

### REFERENCE

[KQL quick reference](#)

[Kusto Query Language](#)

[SQL to Kusto cheat sheet](#)

## Visualize data

### CONCEPT

[Data visualization overview](#)

### HOW-TO GUIDE

[Power BI connector](#)

[Grafana](#)

[ODBC connector](#)

[Tableau](#)

[Kibana \(K2Bridge connector\)](#)

[Redash](#)

[Sisense](#)

## Integrate with other tools

### HOW-TO GUIDE

[Integration with Data Factory](#)

[Apache Spark connector](#)

[Microsoft Flow connector](#)

[Azure DevOps](#)

## Manage and monitor resources and data

### HOW-TO GUIDE

[Manage cluster horizontal scaling](#)

[Manage cluster vertical scaling](#)

[Follower databases](#)

[Manage database permissions](#)

[Use metrics to monitor cluster health](#)

[Use diagnostic logs to monitor ingestion](#)

### REFERENCE

[Management commands](#)

## Architecture

### HOW-TO GUIDE

[Big data analytics with Azure Data Explorer](#)

[Content Delivery Network analytics](#)  
[Azure Data Explorer monitoring](#)  
[Azure Data Explorer interactive analytics](#)  
[IoT analytics with Azure Data Explorer](#)  
[Geospatial data processing and analytics](#)  
[Long-term security log retention with Azure Data Explorer](#)  
[Data analytics for automotive test fleets](#)  
[Real-Time analytics with Azure Service Bus](#)

## Development

---

### HOW-TO GUIDE

[Set up your development environment](#)

---

### REFERENCE

[Kusto client libraries](#)

[REST API overview](#)

# Azure Databricks documentation

Learn Azure Databricks, a unified analytics platform for data analysts, data engineers, data scientists, and machine learning engineers.

## About Azure Databricks

### OVERVIEW

[What is Azure Databricks?](#)

### CONCEPT

[What is the lakehouse?](#)

[What is Delta?](#)

[Databricks service architecture](#)

[Data lakehouse architecture](#)

## Start here

### TUTORIAL

[Free trial & setup](#)

[Query data from a notebook](#)

[Build a basic ETL pipeline](#)

[Build an end-to-end pipeline](#)

[Build a simple lakehouse pipeline](#)

## Integrations

### HOW-TO GUIDE

[Integrate with data sources](#)

[Integrate with data partners](#)

[Integrate with BI tools](#)

## SQL warehousing



[What is data warehousing on Databricks?](#)

[Databricks SQL concepts](#)

## SQL tasks



[Use a sample dashboard](#)

[Run queries and visualize data](#)

## SQL reference



[SQL reference](#)

[API reference ↗](#)

[Release notes](#)

## Data engineering



[What is Delta Live Tables?](#)

[Work with clusters and compute](#)

[Source control with Git](#)

## Data engineering

---



HOW-TO GUIDE

[Overview](#)

[Develop code in notebooks](#)

[Storage: Where's my data?](#)

## Data engineering reference

---



REFERENCE

[API reference](#)

[Release notes](#)

## Machine learning

---



TUTORIAL

[Get started with Databricks Mosaic AI](#)

[10-minute tutorials](#)

## Machine learning tasks

---



HOW-TO GUIDE

[Prepare data & your environment](#)

[Train models](#)

[ML reference solutions](#)

## Troubleshooting

---



HOW-TO GUIDE

[Knowledge Base](#)

# Azure Data Explorer documentation

Azure Data Explorer is a fast, fully managed data analytics service for real-time analysis on large volumes of data streaming from applications, websites, IoT devices, and more. You can use Azure Data Explorer to collect, store, and analyze diverse data to improve products, enhance customer experiences, monitor devices, and boost operations.

## About Azure Data Explorer

### OVERVIEW

[What is Azure Data Explorer?](#)

[How Azure Data Explorer works](#)

### GET STARTED

[Start for free](#)

[Kusto Detective Agency](#) ↗

[Find a partner](#)

### VIDEO

[What is Azure Data Explorer? \(19:47\)](#) ↗

## Training

### TRAINING

[Free Pluralsight training: Azure Data Explorer](#) ↗

[Training module: Introduction to Azure Data Explorer](#)

[Training module: Explore the fundamentals of data analysis using Kusto Query Language \(KQL\)](#)

[Training module: Write your first query with Kusto Query Language](#)

[Training module: Gain insights from your data by using Kusto Query Language](#)

[Training module: Multi-table queries using Kusto Query Language](#)

[Training module: Characterize an unfamiliar dataset with Azure Data Explorer](#)

[Training module: Create dashboards in Azure Data Explorer](#)

[Learning path: Data analysis in Azure Data Explorer with Kusto Query Language](#)

## Create cluster and database

### QUICKSTART

[Create a cluster and database](#)

### HOW-TO GUIDE

[Python](#)

[Azure Resource Manager template](#)

[C#](#)

[PowerShell](#)

[Azure CLI](#)

## Ingest data into Azure Data Explorer

### CONCEPT

[Data ingestion overview](#)

### QUICKSTART

[Get data from a local file](#)

### HOW-TO GUIDE

[Ingestion wizard](#)

[Ingest historical data](#)

[Streaming ingestion](#)

[Python](#)

[Event hub using Azure portal](#)

[Event Grid using Azure portal](#)

## Query data

### QUICKSTART

[Query data with the Azure Data Explorer web UI](#)

### TUTORIAL

[Write KQL queries](#)

### HOW-TO GUIDE

[Query data in Azure Data Lake](#)

[Query data using T-SQL](#)

[Query data using Python](#)

[Time series analysis](#)

### REFERENCE

[KQL quick reference](#)

[Kusto Query Language](#)

[SQL to Kusto cheat sheet](#)

## Visualize data

### CONCEPT

[Data visualization overview](#)

### HOW-TO GUIDE

[Power BI connector](#)

[Grafana](#)

[ODBC connector](#)

[Tableau](#)

[Kibana \(K2Bridge connector\)](#)

[Redash](#)

[Sisense](#)

## Integrate with other tools

### HOW-TO GUIDE

[Integration with Data Factory](#)

[Apache Spark connector](#)

[Microsoft Flow connector](#)

[Azure DevOps](#)

## Manage and monitor resources and data

### HOW-TO GUIDE

[Manage cluster horizontal scaling](#)

[Manage cluster vertical scaling](#)

[Follower databases](#)

[Manage database permissions](#)

[Use metrics to monitor cluster health](#)

[Use diagnostic logs to monitor ingestion](#)

### REFERENCE

[Management commands](#)

## Architecture

### HOW-TO GUIDE

[Big data analytics with Azure Data Explorer](#)

[Content Delivery Network analytics](#)  
[Azure Data Explorer monitoring](#)  
[Azure Data Explorer interactive analytics](#)  
[IoT analytics with Azure Data Explorer](#)  
[Geospatial data processing and analytics](#)  
[Long-term security log retention with Azure Data Explorer](#)  
[Data analytics for automotive test fleets](#)  
[Real-Time analytics with Azure Service Bus](#)

## Development

---

### HOW-TO GUIDE

[Set up your development environment](#)

---

### REFERENCE

[Kusto client libraries](#)

[REST API overview](#)

# Introduction to Azure Data Lake Storage

Article • 11/15/2024

Azure Data Lake Storage is a set of capabilities dedicated to big data analytics, built on [Azure Blob Storage](#).

Azure Data Lake Storage converges the capabilities of [Azure Data Lake Storage Gen1](#) with Azure Blob Storage. For example, Data Lake Storage provides file system semantics, file-level security, and scale. Because these capabilities are built on Blob storage, you also get low-cost, tiered storage, with high availability/disaster recovery capabilities.

Data Lake Storage makes Azure Storage the foundation for building enterprise data lakes on Azure. Designed from the start to service multiple petabytes of information while sustaining hundreds of gigabits of throughput, Data Lake Storage allows you to easily manage massive amounts of data.

## What is a Data Lake?

A *data lake* is a single, centralized repository where you can store all your data, both structured and unstructured. A data lake enables your organization to quickly and more easily store, access, and analyze a wide variety of data in a single location. With a data lake, you don't need to conform your data to fit an existing structure. Instead, you can store your data in its raw or native format, usually as files or as binary large objects (blobs).

*Azure Data Lake Storage* is a cloud-based, enterprise data lake solution. It's engineered to store massive amounts of data in any format, and to facilitate big data analytical workloads. You use it to capture data of any type and ingestion speed in a single location for easy access and analysis using various frameworks.

## Data Lake Storage

Azure Data Lake Storage isn't a dedicated service or account type. Instead, it's implemented as a set of capabilities that you use with the Blob Storage service of your Azure Storage account. You can unlock these capabilities by enabling the hierarchical namespace setting.

Data Lake Storage includes the following capabilities.

- ✓ Hadoop-compatible access

- ✓ Hierarchical directory structure
- ✓ Optimized cost and performance
- ✓ Finer grain security model
- ✓ Massive scalability

## Hadoop-compatible access

Azure Data Lake Storage is primarily designed to work with Hadoop and all frameworks that use the Apache [Hadoop Distributed File System \(HDFS\)](#) as their data access layer. Hadoop distributions include the [Azure Blob File System \(ABFS\)](#) driver, which enables many applications and frameworks to access Azure Blob Storage data directly. The ABFS driver is [optimized specifically](#) for big data analytics. The corresponding REST APIs are surfaced through the endpoint `dfs.core.windows.net`.

Data analysis frameworks that use HDFS as their data access layer can directly access Azure Data Lake Storage data through ABFS. The Apache Spark analytics engine and the Presto SQL query engine are examples of such frameworks.

For more information about supported services and platforms, see [Azure services that support Azure Data Lake Storage](#) and [Open source platforms that support Azure Data Lake Storage](#).

## Hierarchical directory structure

The [hierarchical namespace](#) is a key feature that enables Azure Data Lake Storage to provide high-performance data access at object storage scale and price. You can use this feature to organize all the objects and files within your storage account into a hierarchy of directories and nested subdirectories. In other words, your Azure Data Lake Storage data is organized in much the same way that files are organized on your computer.

Operations such as renaming or deleting a directory, become single atomic metadata operations on the directory. There's no need to enumerate and process all objects that share the name prefix of the directory.

## Optimized cost and performance

Azure Data Lake Storage is priced at Azure Blob Storage levels. It builds on Azure Blob Storage capabilities such as automated lifecycle policy management and object level tiering to manage big data storage costs.

Performance is optimized because you don't need to copy or transform data as a prerequisite for analysis. The hierarchical namespace capability of Azure Data Lake Storage allows for efficient access and navigation. This architecture means that data processing requires fewer computational resources, reducing both the speed and cost of accessing data.

## Finer grain security model

The Azure Data Lake Storage access control model supports both Azure role-based access control (Azure RBAC) and Portable Operating System Interface for UNIX (POSIX) access control lists (ACLs). There are also a few extra security settings that are specific to Azure Data Lake Storage. You can set permissions either at the directory level or at the file level. All stored data is encrypted at rest by using either Microsoft-managed or customer-managed encryption keys.

## Massive scalability

Azure Data Lake Storage offers massive storage and accepts numerous data types for analytics. It doesn't impose any limits on account sizes, file sizes, or the amount of data that can be stored in the data lake. Individual files can have sizes that range from a few kilobytes (KBs) to a few petabytes (PBs). Processing is executed at near-constant per-request latencies that are measured at the service, account, and file levels.

This design means that Azure Data Lake Storage can easily and quickly scale up to meet the most demanding workloads. It can also just as easily scale back down when demand drops.

## Built on Azure Blob Storage

The data that you ingest persist as blobs in the storage account. The service that manages blobs is the Azure Blob Storage service. Data Lake Storage describes the capabilities or "enhancements" to this service that caters to the demands of big data analytic workloads.

Because these capabilities are built on Blob Storage, features such as diagnostic logging, access tiers, and lifecycle management policies are available to your account. Most Blob Storage features are fully supported, but some features might be supported only at the preview level and there are a handful of them that aren't yet supported. For a complete list of support statements, see [Blob Storage feature support in Azure Storage accounts](#). The status of each listed feature will change over time as support continues to expand.

# Documentation and terminology

The Azure Blob Storage table of contents features two sections of content. The **Data Lake Storage** section of content provides best practices and guidance for using Data Lake Storage capabilities. The **Blob Storage** section of content provides guidance for account features not specific to Data Lake Storage.

As you move between sections, you might notice some slight terminology differences. For example, content featured in the Blob Storage documentation, will use the term *blob* instead of *file*. Technically, the files that you ingest to your storage account become blobs in your account. Therefore, the term is correct. However, the term *blob* can cause confusion if you're used to the term *file*. You'll also see the term *container* used to refer to a *file system*. Consider these terms as synonymous.

## See also

- [Introduction to Azure Data Lake Storage \(Training module\)](#)
- [Best practices for using Azure Data Lake Storage](#)
- [Known issues with Azure Data Lake Storage](#)
- [Multi-protocol access on Azure Data Lake Storage](#)

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback ↗](#) | Get help at Microsoft Q&A

# Azure Key Vault

Learn how to use Key Vault to create and maintain keys that access and encrypt your cloud resources, apps, and solutions. Tutorials, API references, and more.



OVERVIEW  
[Introduction to Azure Key Vault](#)



WHAT'S NEW  
[Important updates to service](#)



OVERVIEW  
[Authentication](#)



OVERVIEW  
[Basic concepts](#)



OVERVIEW  
[Logging](#)



OVERVIEW  
[Throttling](#)

## Certificates

- [About certificates](#)
- [Set and retrieve a certificate from Azure Key Vault using the Azure portal](#)
- [Get started with Key Vault certificates](#)
- [Monitor and manage certificate creation](#)

[See more >](#)

## Keys

- [About keys](#)
- [Data Encryption](#)
- [Set and retrieve a key from Azure Key Vault using the Azure portal](#)
- [Configure key auto-rotation](#)
- [Import HSM-protected keys \(overview\)](#)

[See more >](#)

## Secrets

- [About secrets](#)
- [Set and retrieve a secret from Azure Key Vault using the Azure portal](#)
- [Rotate secrets for single-user resources](#)

[See more >](#)

## Managed HSM

- [What is managed HSM?](#)
- [Best practices](#)
- [Provision and activate a managed HSM](#)

[See more >](#)

## Client libraries

- [!\[\]\(9f4c698befec7be573d06def4d2617a0\_img.jpg\) Client libraries for Azure Key Vault](#)
- [!\[\]\(ca0e2956730de4ee2b1dbe6258ccb4db\_img.jpg\) .NET Client library](#)
- [!\[\]\(7a89c27b33d31acbe8be5f6b5df8a5a7\_img.jpg\) Python Client library](#)
- [!\[\]\(4c53662424e67857f68c595acc6dc5b1\_img.jpg\) Java Client library ↗](#)
- [!\[\]\(2daf003b16b0c43723014a69081f857c\_img.jpg\) Spring Client library](#)
- [!\[\]\(7d14ee104d9644513bb61862ae8beaad\_img.jpg\) Node.js Client library](#)

## Key Vault with Storage

- [!\[\]\(2b132f33322a2930610937e3cc33317e\_img.jpg\) Manage storage keys - CLI](#)
- [!\[\]\(6c6b2817882dbdc92b55a0cab18bc009\_img.jpg\) Manage storage keys - PowerShell](#)
- [!\[\]\(491b1449953318798115d2edfc929581\_img.jpg\) Fetch SAS tokens in code](#)

## Key Vault with Virtual Machines

- [!\[\]\(1b5859d2daa9991922cabef3d906c818\_img.jpg\) Key Vault VM extension - Windows](#)
- [!\[\]\(e43bf52f486da794c22c6959da0f8dbf\_img.jpg\) Key Vault VM extension - Linux](#)
- [!\[\]\(0e95adbe0d29afe0c26b549c96f018bd\_img.jpg\) Key Vault with Linux VMs in Ansible](#)
- [!\[\]\(79e86c231c10e3429eb25b258fac6cef\_img.jpg\) Key Vault with Azure Disk Encryption - Windows](#)
- [!\[\]\(ec1473221c265317c7bbd6a2e07b91bb\_img.jpg\) Key Vault with Azure Disk Encryption - Linux](#)
- [!\[\]\(47061ae7ac71b83446026cfb4392351f\_img.jpg\) Key Vault with Virtual Machine Scale Sets](#)

## Key Vault Integration

- [!\[\]\(715ed4b476bdf43a063c0d06f10c750d\_img.jpg\) Azure Dev Ops - Reference Key Vault secrets in Azure Pipelines](#)
- [!\[\]\(fc10b9ab96e6e5f2fef2fa7b84f90285\_img.jpg\) App Services - Reference Key Vault secrets in App Services](#)
- [!\[\]\(b4ce7d7bf34d1a49a3b0cda64d9a13e8\_img.jpg\) App Services - Maintain App Service certificates in Key Vault](#)
- [!\[\]\(a4c2d74976dea03542e84332da7602ea\_img.jpg\) Azure Kubernetes Service - Use Key Vault secrets, certificates, keys in AKS](#)
- [!\[\]\(ed1f2c6687fe6b987c63e2a6fe059d62\_img.jpg\) Azure Databricks - Secrets Management with Key Vault](#)
- [!\[\]\(e0762ceb6e87935d4764fc531810cef8\_img.jpg\) Spring Integration - Read a secret from Azure Key Vault in a Spring Boot application](#)
- [!\[\]\(19532963767650e6d22bb78d888986c3\_img.jpg\) Spring Integration - Secure Spring Boot apps using Azure Key Vault certificates](#)

# Azure Machine Learning documentation

Learn how to train and deploy models and manage the ML lifecycle (MLOps) with Azure Machine Learning. Tutorials, code examples, API references, and more.

## Overview



[What is Azure Machine Learning?](#)

[What is Responsible AI?](#)

## Setup & quickstart



[Create resources](#)

[Get started with Azure Machine Learning](#)

## Start with the basics



[Prepare and explore data](#)

[Develop on a cloud workstation](#)

[Train a model](#)

[Deploy a model](#)

[Set up a reusable pipeline](#)

## Build AI solutions



[What is Azure Machine Learning prompt flow?](#)

[Get started in prompt flow](#)

## Work with data



HOW-TO GUIDE

[Use Apache Spark in Azure Machine Learning](#)

[Create data assets](#)

## Train models



HOW-TO GUIDE

[Run training with CLI, SDK, or REST API](#)

[Build pipelines from reusable components](#)

## Deploy models



DEPLOY

[Streamline model deployment with endpoints](#)

[Real-time scoring with online endpoints](#)

## Manage the ML lifecycle (MLOps)



HOW-TO GUIDE

[Track, monitor, analyze training runs](#)

[Model management, deployment & monitoring](#)

## Security for ML projects



HOW-TO GUIDE

[Create a secure workspace](#)

[Connect to data sources](#)

[Enterprise security & governance](#)

## Reference docs

---

### REFERENCE

[Python SDK \(v2\)](#)

[CLI \(v2\)](#)

[REST API](#)

[Algorithm & component reference](#)

## Resources

---

### REFERENCE

[Upgrade to v2](#)

[Python SDK \(v2\) code examples ↗](#)

[CLI \(v2\) code examples ↗](#)

# Azure Policy documentation

Azure Policy helps you manage and prevent IT issues with policy definitions that enforce rules and effects for your resources.

## About Azure Policy

### OVERVIEW

[What is Azure Policy?](#)

[Policy effect](#)

[Definition structure](#)

[Scope](#)

[Assignment structure](#)

[Exemption structure](#)

### ARCHITECTURE

[Cloud Adoption Framework \(Govern\)](#)

[Enterprise-scale landing zones](#)

## Get started

### QUICKSTART

[Assign a policy \(Azure portal\)](#)

[Assign a policy \(Azure CLI\)](#)

[Assign a policy \(Azure PowerShell\)](#)

[Assign a policy \(REST\)](#)

[Assign a policy \(ARM template\)](#)

[Assign a policy \(Bicep\)](#)

[Assign a policy \(Terraform\)](#)

### TUTORIAL

[Create and manage policies](#)

[Use VS Code extension](#)

[Design Azure Policy as Code workflows](#)

---

 **DEPLOY**

[Index of policy samples](#)

## Author policies

---

 **TUTORIAL**

[Create a custom policy](#)

[Manage tag governance](#)

---

 **HOW-TO GUIDE**

[Create policy with SDK](#)

[Author policies for arrays](#)

[Export resources](#)

---

 **CONCEPT**

[Regulatory Compliance](#)

[Azure Policy for Kubernetes](#)

[Azure Machine Configuration](#)

## Review & Remediate resources

---

 **HOW-TO GUIDE**

[Get compliance data](#)

[Determine causes of non-compliance](#)

[Exempt resources](#)

Remediate non-compliant resources

## Get involved



### GET STARTED

[Microsoft Q&A for Azure Policy](#)

[Azure Governance YouTube channel](#)

[Request product feature](#)

[Tech Community for Azure Governance](#)

## Reference



### REFERENCE

[Azure CLI](#)

[Azure PowerShell \(Policy\)](#)

[Azure PowerShell \(Policy Insights\)](#)

[REST](#)

[Resource Manager templates](#)

## Reference (more)



### REFERENCE

[Azure SDK for .NET \(Assignments\)](#)

[Azure SDK for .NET \(Policy Definitions\)](#)

[Azure SDK for JavaScript \(Policy\)](#)

[Azure SDK for JavaScript \(Policy Insights\)](#)

[Azure SDK for Python \(Policy\)](#)

[Azure SDK for Python \(Policy client\)](#)



# Private Link documentation

Azure Private Link enables you to access Azure PaaS Services (for example, Azure Storage and SQL Database) and Azure hosted customer-owned/partner services over a Private Endpoint in your virtual network. Traffic between your virtual network and the service traverses over the Microsoft backbone network, eliminating exposure from the public Internet. You can also create your own Private Link Service in your virtual network and deliver it privately to your customers.

## What is Azure Private Link?

### OVERVIEW

[What is Private Link?](#)

[What is a private endpoint?](#)

[What is Private Link service?](#)

[Pricing ↗](#)

### TRAINING

[Introduction to Azure Private Link](#)

## Getting started

### QUICKSTART

[Create a Private Link service](#)

[Create a private endpoint](#)

[Private DNS zone values](#)

### VIDEO

[Secure Azure PaaS resources using Private Link ↗](#)

## Scenarios

### TUTORIAL

[Connect to a storage account](#)

[Inspect traffic with Azure Firewall](#)

[Private Endpoints DNS integration](#)

[Connect to a SQL server](#)

---

#### HOW-TO GUIDE

[Approve private endpoint connections across subscriptions](#)

[Manage network policies](#)

## Troubleshoot and support

---

#### HOW-TO GUIDE

[Private endpoint connectivity](#)

[Private link service connectivity](#)

## Networking foundation

---

#### GET STARTED

[Documentation](#)

[Virtual Network](#)

[DNS](#)

## Concepts and architecture

---

#### CONCEPT

[Security baseline](#)

[Multitenancy and Azure Private Link](#)

---

#### TRAINING

## ARCHITECTURE

Azure Private Link in a hub-and-spoke network

Web app private connectivity to Azure SQL Database

Multi-tier app service with private endpoint

# Microsoft Purview

Microsoft Purview is a comprehensive portfolio of products spanning data governance, data security, and risk and compliance solutions.



OVERVIEW  
[Learn about Microsoft Purview](#)



GET STARTED  
[Set up data governance solutions](#)



GET STARTED  
[Set up data security solutions](#)



GET STARTED  
[Set up risk and compliance solutions](#)



QUICKSTART  
[Try the data governance solutions trial for free](#)



QUICKSTART  
[Try the data security and risk and compliance solutions trial for free](#)



WHAT'S NEW  
[Catch up on new functionality and documentation updates](#)



TUTORIAL  
[Learn new skills for Microsoft Purview solutions](#)

## Explore Microsoft Purview solutions

Guidance to help you explore and get started with Microsoft Purview solutions in the Purview, compliance, and governance portals.



Microsoft Purview data governance



Microsoft Purview data security



Microsoft Purview risk and compliance

## solutions

[Unified Catalog](#)  
[Governance domains](#)  
[Data products](#)  
[Data quality](#)  
[Data Map](#)

## solutions

[Data Loss Prevention](#)  
[Data Security Posture Management \(preview\)](#)  
[Information Barriers](#)  
[Information Protection](#)  
[Insider Risk Management](#)

## solutions

[Audit](#)  
[Communication Compliance](#)  
[Compliance Manager](#)  
[Data Lifecycle Management](#)  
[eDiscovery](#)  
[Records Management](#)

# Azure Stream Analytics documentation

Azure Stream Analytics is a fully managed, real-time analytics service designed to help you analyze and process fast moving streams of data that can be used to get insights, build reports or trigger alerts and actions. Learn how to use Azure Stream Analytics with our quickstarts, tutorials, and samples.

## About Azure Stream Analytics

### OVERVIEW

[What is Azure Stream Analytics?](#)

[Azure Stream Analytics solution patterns](#)

### QUICKSTART

[Create a job with Azure portal](#)

### TUTORIAL

[Capture Event Hubs data in parquet format](#)

[Write to a delta table in Data Lake Storage Gen2](#)

[Build real time Power BI dashboards with no code editor](#)

[Analyze fraudulent call data and visualize results](#)

### CONCEPT

[Choose a streaming analytics technology](#)

[Choose a development tool for your jobs](#)

## Connect a job to a data source

### OVERVIEW

[Learn about input types](#)

### CONCEPT

[Stream input data](#)

[Join reference data for lookups](#)

---

#### HOW-TO GUIDE

[Use reference data from a SQL Database](#)

[Troubleshoot input connections](#)

[Process Apache Kafka for Event Hubs events](#)

[Process data from Azure Event Hubs](#)

## Send data to an output sink

---

#### OVERVIEW

[Learn about output types](#)

---

#### CONCEPT

[Output to Azure Cosmos DB](#)

[Output to Azure SQL Database](#)

[Custom blob output partitioning](#)

[Output error handling policies](#)

---

#### HOW-TO GUIDE

[Troubleshoot output connections](#)

## Monitor and troubleshoot your job

---

#### HOW-TO GUIDE

[Monitor and manage jobs using Azure portal](#)

[Monitor and manage jobs from Visual Studio](#)

[Analyze job performance using job metrics and dimensions](#)

[Debug queries locally using job diagram in Visual Studio Code](#)

[Troubleshoot using resource logs](#)

[Debug using the job diagram](#)

[Troubleshoot data errors](#)

## Review temporal concepts

### CONCEPT

[Common query patterns](#)

[Understand time handling in Stream Analytics](#)

[Internal checkpoint and replay](#)

## Transform data with Stream Analytics queries

### CONCEPT

[Introduction to windowing functions](#)

[Introduction to geospatial functions](#)

[Parse JSON and Avro data](#)

### HOW-TO GUIDE

[Troubleshoot query logic](#)

[Write JavaScript user-defined aggregates](#)

[Develop .NET user-defined functions for IoT Edge jobs](#)

### TUTORIAL

[Run a JavaScript user-defined function](#)

[Run a C# user-defined function on a Stream Analytics Edge job](#)

## Analyze with machine learning

### HOW-TO GUIDE

[Use machine learning to detect anomalies](#)

[Perform sentiment analysis with Machine Learning Studio \(classic\)](#)

[Scale jobs with Machine Learning Studio \(classic\) functions](#)

## Create and manage jobs with developer tools

### HOW-TO GUIDE

[Explore jobs with Visual Studio Code](#)

[Test queries with sample data in Visual Studio Code](#)

[Install tools for Visual Studio](#)

[Test queries with sample data in Visual Studio](#)

[Test queries with live data in Visual Studio](#)

[View jobs in Visual Studio](#)

[Develop an IoT Edge job in Visual Studio](#)

### VIDEO

[Author, manage and test with Visual Studio Code](#) ↗

## Try Stream Analytics tutorials and solutions

### TUTORIAL

[Real-time fraud detection](#)

[Trigger an Azure Function](#)

### HOW-TO GUIDE

[Build solutions with geofencing and geospatial aggregation](#)

[Run jobs on IoT Edge](#)

[Build a sentiment analysis solution](#)

[Build an IoT solution](#)

[High-frequency trading simulation](#)

Alert based on adjustable rule thresholds

## Deploy Stream Analytics jobs

### HOW-TO GUIDE

[Set up a CI/CD pipeline with Visual Studio Code](#)

[Set up a CI/CD pipeline with Visual Studio](#)

[Use REST APIs to set up CI/CD for IoT Edge jobs](#)

## Scale Stream Analytics jobs

### HOW-TO GUIDE

[Scale with streaming units](#)

[Scale with query parallelization](#)

[Scale jobs to increase throughput](#)

[Optimize processing by repartitioning](#)

# Azure SQL documentation

Find documentation about the Azure SQL family of SQL Server database engine products in the cloud: Azure SQL Database, Azure SQL Managed Instance, and SQL Server on Azure VM.

## Azure SQL Database

### OVERVIEW

[What is Azure SQL Database?](#)

### WHAT'S NEW

[What's new?](#)

[Try for free](#)

### QUICKSTART

[Create SQL Database](#)

[Configure firewall](#)

### VIDEO

[Azure SQL for beginners](#)

[Azure SQL Database essentials](#)

### CONCEPT

[Copilot in Azure SQL Database](#)

[Advanced security](#)

[Business continuity](#)

[Database watcher \(Preview\)](#)

[T-SQL differences with SQL Server](#)

## Azure SQL Managed Instance

## OVERVIEW

[What is Azure SQL Managed Instance?](#)

## WHAT'S NEW

[What's new?](#)

[Try for free](#)

## QUICKSTART

[Create SQL Managed Instance](#)

[Configure VM to connect](#)

[Restore sample database](#)

## VIDEO

[Azure SQL Managed Instance overview](#)

## CONCEPT

[Advanced security](#)

[Business continuity](#)

[Database watcher \(Preview\)](#)

[T-SQL differences with SQL Server](#)

# SQL Server on Azure VM

## OVERVIEW

[What is SQL Server on Windows VMs?](#)

[What is SQL Server on Linux VM?](#)

[SQL IaaS Agent extension](#)

## WHAT'S NEW

[What's new?](#)

## QUICKSTART

[Create SQL on Azure VM \(Windows\)](#)

[Create SQL on Azure VM \(Linux\)](#)

## VIDEO

[SQL Server on Azure VM overview](#)

## CONCEPT

[Performance guidelines](#)

[High availability & disaster recovery](#)

## Learn Azure SQL

### TRAINING

[Azure SQL for beginners](#)

[Azure SQL fundamentals](#)

[Azure SQL hands-on labs ↗](#)

[Azure SQL bootcamp](#)

[Educational SQL resources](#)

## Connect and query

### QUICKSTART

[Overview](#)

[SQL Server Management Studio \(SSMS\)](#)

[Azure Data Studio](#)

[Azure portal](#)

[Visual Studio \(.NET\)](#)

[Visual Studio Code](#)

[.NET Core](#)

Python

With Microsoft Entra ID (formerly Azure Active Directory) and SqlClient

## Reference

---

### DEPLOY

[Azure CLI samples](#)

[PowerShell samples](#)

[ARM template samples](#)

---

### DOWNLOAD

[SQL Server Management Studio \(SSMS\)](#)

[Azure Data Studio](#)

[SQL Server Data Tools](#)

[Visual Studio 2019 ↗](#)

---

### REFERENCE

[Migration guide](#)

[Transact-SQL \(T-SQL\)](#)

[Azure CLI](#)

[PowerShell](#)

[REST API](#)

## Development

---

### OVERVIEW

[Application development](#)

[Connect apps to Azure SQL](#)

[Disaster recovery app design](#)

[Managing rolling upgrades \(SQL DB\)](#)

[Development strategies \(SQL VM\)](#)

[SaaS database tenancy patterns](#)

---

#### HOW-TO GUIDE

[Design first database \(SSMS\)](#)

[Design first database \(C#\)](#)

## Migrate from SQL Server

---

#### OVERVIEW

[Migrating SQL Server Workloads FAQ](#)

---

#### DEPLOY

[Azure SQL Database](#)

[Azure SQL Managed Instance](#)

[SQL Server on Azure VMs](#)

# Azure Synapse Analytics

Azure Synapse is a limitless analytics service that brings together enterprise data warehousing and Big Data analytics. It gives you the freedom to query data on your terms, using either serverless or dedicated resources—at scale.

## About Azure Synapse Analytics

### OVERVIEW

[What is Azure Synapse](#)

[What's new?](#)

[Terminology](#)

[FAQ](#)

## Synapse SQL

### OVERVIEW

[Azure Synapse SQL architecture](#)

### QUICKSTART

[Use serverless SQL pool](#)

[Create a dedicated SQL pool using Synapse Studio](#)

### DEPLOY

[Migration guides](#)

## Get started with Azure Synapse Analytics

### GET STARTED

[Step-by-step to getting started](#)

[STEP 1 - Create and set up a Synapse workspace](#)

[STEP 2 - Analyze using a dedicated SQL pool](#)

[STEP 3 - Analyze using Apache Spark](#)

[STEP 4 - Analyze using a serverless SQL pool](#)

[STEP 5 - Analyze data in a storage account](#)

[STEP 6 - Orchestrate with pipelines](#)

[STEP 7 - Visualize data with Power BI](#)

[STEP 8 - Monitor activities](#)

[STEP 9 - Explore the Knowledge center](#)

## Data Explorer

### OVERVIEW

[Data Explorer in Azure Synapse Analytics](#)

### QUICKSTART

[Create a Data Explorer pool using the Synapse Studio](#)

## Database templates

### OVERVIEW

[Database templates in Azure Synapse Analytics](#)

### QUICKSTART

[Create a new Lake database leveraging database templates](#)

## Apache Spark

### OVERVIEW

[Apache Spark in Azure Synapse Analytics](#)

## QUICKSTART

[Create a serverless Apache Spark pool using Synapse Studio](#)

## Synapse Link

### OVERVIEW

[Azure Synapse Link for Azure Cosmos DB](#)

[Azure Synapse Link for SQL](#)

### QUICKSTART

[Connect to Azure Synapse Link for Azure Cosmos DB](#)

[Connect to Azure Synapse Link for Azure SQL DB](#)

[Connect to Azure Synapse Link for SQL Server 2022](#)

## Machine Learning

### OVERVIEW

[Machine Learning in Azure Synapse](#)

### QUICKSTART

[Create a new Azure Machine Learning linked service](#)

### TUTORIAL

[Model scoring wizard for SQL pools](#)

## Pipeline and data flow

### CONCEPT

[Pipeline and activities](#)

---

 QUICKSTART

[Transform data](#)

[Load data to SQL pool](#)

# Azure Data Factory documentation

Azure Data Factory is Azure's cloud ETL service for scale-out serverless data integration and data transformation. It offers a code-free UI for intuitive authoring and single-pane-of-glass monitoring and management. You can also lift and shift existing SSIS packages to Azure and run them with full compatibility in ADF. SSIS Integration Runtime offers a fully managed service, so you don't have to worry about infrastructure management.

## What's New in Azure Data Factory

### OVERVIEW

[What's New in Azure Data Factory](#)

[Change data capture](#)

[Workflow Orchestration Manager](#)

## About Azure Data Factory

### OVERVIEW

[Introduction to Azure Data Factory](#)

[Get started with Azure Data Factory](#)

### REFERENCE

[Pricing](#)

[Request a feature](#)

## Get started

### CONCEPT

[Monitor Azure Data Factory](#)

[Pipelines and activities in Azure Data Factory](#)

[Pipeline execution and triggers](#)

[Mapping data flows in Azure Data Factory](#)

## Using expressions and functions

---

### QUICKSTART

[Create an Azure Data Factory with UI](#)

[Use the Copy Data tool](#)

[Transform data with mapping data flows](#)

## Data movement

---

### OVERVIEW

[Data lake and data warehouse migration](#)

### CONCEPT

[Copy activity overview](#)

[Troubleshoot Copy activity performance](#)

[Troubleshoot connectors](#)

## Data transformation

---

### TUTORIAL

[Create data flow](#)

[Copy and transform data from a REST endpoint](#)

## Lift and shift SSIS packages

---

### OVERVIEW

[SSIS migration assessment rules](#)

### CONCEPT

[Integration runtime in Azure Data Factory](#)

[Create a self-hosted integration runtime](#)

---

 **TUTORIAL**

[Running SSIS packages in Azure](#)

## Workflow Orchestration Manager

---

 **CONCEPT**

[Workflow Orchestration Manager](#)

---

 **HOW-TO GUIDE**

[How does Workflow Orchestration Manager work?](#)

---

 **TUTORIAL**

[Run an existing pipeline with Workflow Orchestration Manager](#)

## Change data capture (CDC)

---

 **OVERVIEW**

[Change data capture](#)

---

 **TUTORIAL**

[Capture changed data with a CDC resource](#)

## SAP knowledge center

---

 **CONCEPT**

[Overview](#)

[SAP connectors](#)

[SAP templates](#)

## Author, Monitor and Manage

### HOW-TO GUIDE

[Visually monitor Azure data factories](#)

[Troubleshoot Azure Data Factory connectors](#)

[Continuous integration and delivery](#)

# Azure Event Hubs documentation

Learn how to use Event Hubs to ingest millions of events per second from connected devices and applications.

## About Event Hubs

### OVERVIEW

[What is Event Hubs?](#)

[Event Hubs for Apache Kafka](#)

[Schema Registry](#)

[Event Hubs Capture](#)

[Event Hubs Premium](#)

[Event Hubs Dedicated](#)

## Create an event hub

### QUICKSTART

[Azure portal](#)

[Azure CLI](#)

[Azure PowerShell](#)

[Azure Resource Manager template](#)

[Bicep](#)

[Azure Resource Manager template](#)

## Send and receive events

### QUICKSTART

[.NET Core](#)

[Java](#)

[Spring](#)

[Python](#)

[JavaScript](#)

[Apache Kafka](#)

[Apache Storm \(receive only\)](#)

[Go](#)

## Integrate with Apache Kafka

---

### OVERVIEW

[What is Event Hubs for Apache Kafka?](#)

---

### QUICKSTART

[Stream into Event Hubs for Apache Kafka](#)

[Use Spring Kafka with Azure Event Hubs](#)

---

### TUTORIAL

[Process events using Stream Analytics](#)

---

### HOW-TO GUIDE

[Migrate existing Kafka workloads to Event Hubs](#)

[Connect Apache Spark to an event hub](#)

[Connect Apache Flink to an event hub](#)

[Integrate Apache Kafka Connect with an event hub](#)

[Connect Akka streams to an event hub](#)

## Capture events

---

### OVERVIEW

[Capture events in Azure Blob Storage or Azure Data Lake Storage](#)

## QUICKSTART

[Use Azure portal to enable capturing of events](#)

[Use Resource Manager template](#)

[Enable capturing from a Python application](#)

## Process events

### TUTORIAL

[Process Event Hubs for Apache Kafka events using Stream Analytics](#)

[Stream data into Azure Databricks using Event Hubs](#)

## Monitor and manage

### HOW-TO GUIDE

[Monitor Event Hubs using Azure Monitor](#)

[Automatically scale up](#)

[Troubleshoot exceptions](#)

## Secure

### HOW-TO GUIDE

[Use firewall rules](#)

[Use virtual network service endpoints](#)

[Use private endpoints](#)

[Configure customer-managed keys for encrypting data at rest](#)

[Enforce minimum required TLS version](#)

## Create a dedicated cluster

## OVERVIEW

[What is Event Hubs Dedicated?](#)

## QUICKSTART

[Create a dedicated Event Hubs cluster](#)

## Learn modules

### TRAINING

[Explore Azure Event Hubs](#)

Enable reliable messaging for Big Data applications using Azure Event Hubs

Choose a messaging model in Azure to loosely connect your services

## Event Hubs on Azure Stack Hub preview

### OVERVIEW

[Overview](#)

[Common feature set](#)

### HOW-TO GUIDE

[Capacity planning](#)

[Installation prerequisites \(operator\)](#)

[Installation \(operator\)](#)

[Use Blob Storage as checkpoint store](#)

### QUICKSTART

[Create Event Hubs cluster](#)

# Azure HDInsight documentation

Azure HDInsight is a managed Apache Hadoop service that lets you run Apache Spark, Apache Hive, Apache Kafka, Apache HBase, and more in the cloud.

## About HDInsight

### OVERVIEW

[What is Azure HDInsight?](#)

[HDInsight OSS components and versions](#)

### GET STARTED

[Build analytical solutions with Azure HDInsight](#)

[Create HDInsight clusters](#)

[Monitoring with HDInsight](#)

[Scale cluster to save cost](#)

[HDInsight security overview](#)

## Apache Spark

### GET STARTED

[What is Apache Spark?](#)

[Create Spark clusters and run Spark in Jupyter](#)

[Load data and run Spark queries](#)

### HOW-TO GUIDE

[Manage Spark dependencies](#)

[Optimize Spark jobs](#)

## Apache Hadoop

## GET STARTED

[What is Apache Hadoop?](#)

[Create Hadoop clusters and run Hive queries](#)

[Run Map Reduce samples](#)

---

## HOW-TO GUIDE

[Monitor and manage Hadoop clusters](#)

[Use JDBC/ODBC driver](#)

## Integration

### HOW-TO GUIDE

[Spark/Hive - Connect Spark and Hive with Hive Warehouse connector](#)

[Spark/Kafka - Apache Spark structured streaming with Apache Kafka](#)

[Spark/HBase - Query Apache HBase with Apache Spark](#)

[Create on-demand clusters using ADF](#)

---

### CONCEPT

[Selecting the right VM size](#)

[HDInsight supported VM types](#)

[Compare storage options](#)

## Apache Kafka

---

### GET STARTED

[What is Apache Kafka?](#)

[Create Kafka clusters and manage Kafka topics](#)

[Use producer and consumer APIs](#)

---

### HOW-TO GUIDE

[Use Kafka REST proxy](#)

[Kafka TLS encryption & authentication](#)

## Interactive query

### GET STARTED

[What is Interactive Query?](#)

[Analyze flight data with Apache Hive](#)

### HOW-TO GUIDE

[Cluster sizing guide](#)

[Connect Hive with Power BI](#)

## Enterprise readiness

### GET STARTED

[HDInsight security overview](#)

[Plan virtual network](#)

[Hadoop on Active Directory: Enterprise Security Package](#)

[Migrate on-premises clusters to the cloud](#)

## Apache HBase

### GET STARTED

[What is Apache HBase?](#)

[Use HBase in Azure HDInsight](#)

[Query HBase with Apache Phoenix](#)

### HOW-TO GUIDE

[Enable Accelerated Writes for HBase](#)



# Network Watcher documentation

Learn how to use Azure Network Watcher. Quickstarts, tutorials, and more, show you how to gain insight into your Azure Virtual Network with tools like packet capture and flows logs, to diagnose problems with traffic filtering and routing, and to monitor connections.

## About Network Watcher

### OVERVIEW

[What is Azure Network Watcher?](#)

### TRAINING

[Introduction to Azure Network Watcher](#)

[Configure Network Watcher](#)

### DEPLOY

[Monitor the Azure network ↗](#)

## Diagnose VM traffic filter problem

### CONCEPT

[IP flow verify overview](#)

### QUICKSTART

[Diagnose VM traffic filter problem](#)

## Diagnose VM routing problem

### CONCEPT

[Next hop overview](#)



[Diagnose VM routing problem](#)

## Diagnose VM communication issues



[Connection troubleshoot overview](#)

[Connection monitor overview](#)



[Monitor communication between VMs](#)



[Troubleshoot outbound connections](#)

## Log network traffic



[Network security group flow logs overview](#)

[Virtual network flow logs overview](#)



[Log VM network traffic](#)



[Manage network security group flow logs](#)

[Manage virtual network flow logs](#)

## Traffic analytics

## CONCEPT

[Traffic analytics overview](#)

## HOW-TO GUIDE

[Manage traffic analytics using Azure Policy](#)

## Reference

### REFERENCE

[Azure PowerShell](#)

[Azure CLI](#)

[REST](#)

## Related monitoring and management services

### GET STARTED

[Documentation](#)

### OVERVIEW

[Azure Virtual Network Manager overview](#)

[Azure Monitor overview](#)

# Virtual Network documentation

Learn how to use Azure Virtual Network. Quickstarts, tutorials, samples, and more, show you how to deploy a virtual network, control traffic filtering and routing, and connect a virtual network to other virtual networks.

## Learn about Azure Virtual Network

### OVERVIEW

[What is Azure Virtual Network?](#)

[Networking Architecture](#)

### TRAINING

[Introduction to Azure Virtual Networks](#)

[Architect network infrastructure in Azure](#)

## Concepts and Architecture

### CONCEPT

[Virtual network peering](#)

[Route network traffic](#)

[Hub-spoke network topology](#)

## Plan and design

### CONCEPT

[Plan virtual networks](#)

[Connect to an on-premises network](#)

[IPv6](#)

[IP address types and allocation](#)

## Deployment

---

### QUICKSTART

[Create a virtual network](#)

---

### TUTORIAL

[Create a virtual network peer](#)

[Route network traffic](#)

---

### DEPLOY

[Provision a new application in an existing network ↗](#)

## Manage and monitor

---

### HOW-TO GUIDE

[Manage virtual networks](#)

[Manage subnets](#)

[Monitor virtual networks](#)

[Manage network interfaces](#)

## Reliability and availability

---

### HOW-TO GUIDE

[Highly available hybrid architecture](#)

[DHCP server on an Azure Virtual Machine](#)

[Azure VM TCP/IP performance tuning](#)

## Security and compliance

---

### CONCEPT

[Virtual network encryption](#)

[Network security groups](#)

[Application security groups](#)

---

#### TRAINING

[Secure and isolate access to Azure resources](#)

## Troubleshoot

---

#### HOW-TO GUIDE

[Troubleshoot peering issues](#)

[Can't delete virtual network](#)

[IP address 168.63.129.16](#)

[Outbound SMTP connectivity](#)

## How-to guidance

---

#### QUICKSTART

[Configure MTU for virtual machines](#)

[Create a subnet with multiple prefixes](#)

[Update address space for a virtual network peer](#)

[Configure subnet peering](#)

## Network foundation

---

#### GET STARTED

[Documentation ↗](#)

---

#### CONCEPT

[Azure Bastion overview](#)

[Azure DNS overview](#)

[Azure Private Link overview](#)