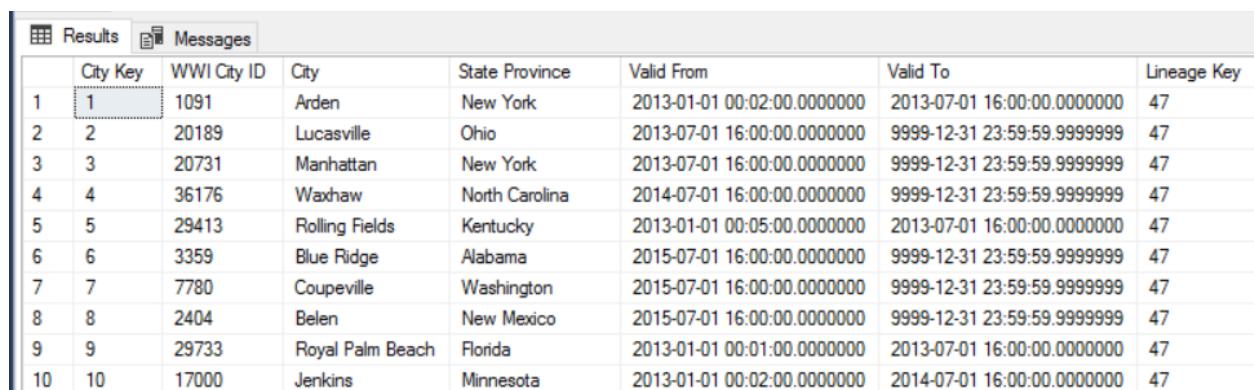


the `LastEditedWhen` column is used. For the dimension data, system-versioned temporal tables are used.

3. When the data migration is complete, update the table that stores the cutoff times.

It's also useful to record a *lineage* for each ELT run. For a given record, the lineage associates that record with the ELT run that produced the data. For each ETL run, a new lineage record is created for every table, showing the starting and ending load times. The lineage keys for each record are stored in the dimension and fact tables.



	City Key	WWI City ID	City	State Province	Valid From	Valid To	Lineage Key
1	1	1091	Arden	New York	2013-01-01 00:02:00.0000000	2013-07-01 16:00:00.0000000	47
2	2	20189	Lucasville	Ohio	2013-07-01 16:00:00.0000000	9999-12-31 23:59:59.9999999	47
3	3	20731	Manhattan	New York	2013-07-01 16:00:00.0000000	9999-12-31 23:59:59.9999999	47
4	4	36176	Waxhaw	North Carolina	2014-07-01 16:00:00.0000000	9999-12-31 23:59:59.9999999	47
5	5	29413	Rolling Fields	Kentucky	2013-01-01 00:05:00.0000000	2013-07-01 16:00:00.0000000	47
6	6	3359	Blue Ridge	Alabama	2015-07-01 16:00:00.0000000	9999-12-31 23:59:59.9999999	47
7	7	7780	Coupeville	Washington	2015-07-01 16:00:00.0000000	9999-12-31 23:59:59.9999999	47
8	8	2404	Belen	New Mexico	2015-07-01 16:00:00.0000000	9999-12-31 23:59:59.9999999	47
9	9	29733	Royal Palm Beach	Florida	2013-01-01 00:01:00.0000000	2013-07-01 16:00:00.0000000	47
10	10	17000	Jenkins	Minnesota	2013-01-01 00:02:00.0000000	2014-07-01 16:00:00.0000000	47

After a new batch of data is loaded into the warehouse, refresh the Analysis Services tabular model. See [Asynchronous refresh with the REST API](#).

Data cleansing

Data cleansing should be part of the ELT process. In this reference architecture, one source of bad data is the city population table, where some cities have zero population, perhaps because no data was available. During processing, the ELT pipeline removes those cities from the city population table. Perform data cleansing on staging tables, rather than external tables.

External data sources

Data warehouses often consolidate data from multiple sources. For example, an external data source that contains demographics data. This dataset is available in Azure blob storage as part of the [WorldWideImportersDW](#) sample.

Azure Data Factory can copy directly from blob storage, using the [blob storage connector](#). However, the connector requires a connection string or a shared access signature, so it can't be used to copy a blob with public read access. As a workaround, you can use PolyBase to create an external table over Blob storage and then copy the external tables into Azure Synapse.

Handling large binary data

For example, in the source database, a City table has a Location column that holds a [geography](#) spatial data type. Azure Synapse doesn't support the [geography](#) type natively, so this field is converted to a [varbinary](#) type during loading. (See [Workarounds for unsupported data types](#).)

However, PolyBase supports a maximum column size of `varbinary(8000)`, which means some data could be truncated. A workaround for this problem is to break the data up into chunks during export, and then reassemble the chunks, as follows:

1. Create a temporary staging table for the Location column.
2. For each city, split the location data into 8000-byte chunks, resulting in 1 – N rows for each city.
3. To reassemble the chunks, use the T-SQL [PIVOT](#) operator to convert rows into columns and then concatenate the column values for each city.

The challenge is that each city will be split into a different number of rows, depending on the size of geography data. For the PIVOT operator to work, every city must have the same number of rows. To make this work, the T-SQL query does some tricks to pad out the rows with blank values, so that every city has the same number of columns after the pivot. The resulting query turns out to be much faster than looping through the rows one at a time.

The same approach is used for image data.

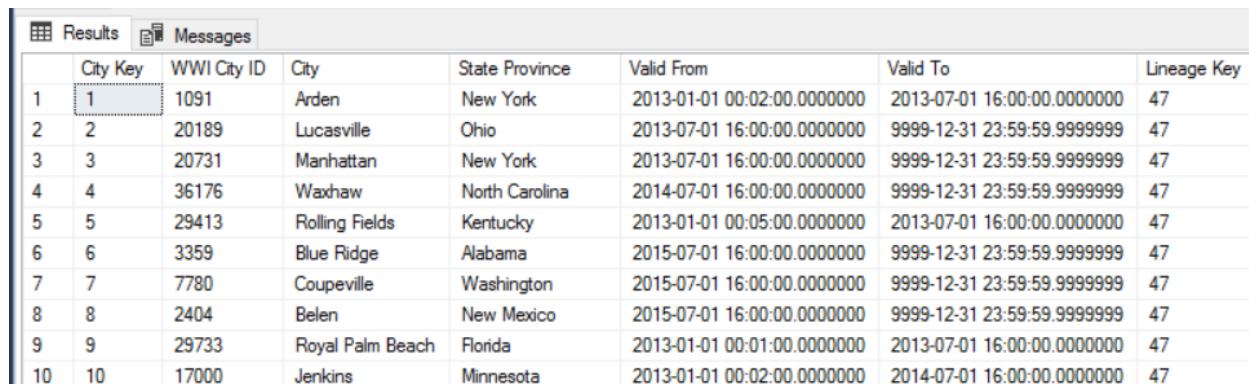
Slowly changing dimensions

Dimension data is relatively static, but it can change. For example, a product might get reassigned to a different product category. There are several approaches to handling slowly changing dimensions. A common technique, called [Type 2](#), is to add a new record whenever a dimension changes.

In order to implement the Type 2 approach, dimension tables need additional columns that specify the effective date range for a given record. Also, primary keys from the source database will be duplicated, so the dimension table must have an artificial primary key.

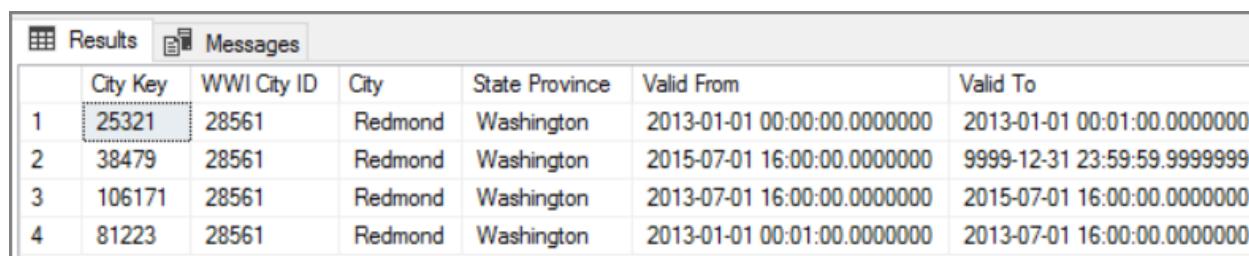
For example, the following image shows the Dimension.City table. The `WWI_City_ID` column is the primary key from the source database. The `city Key` column is an artificial key generated during the ETL pipeline. Also notice that the table has `Valid From` and

`Valid To` columns, which define the range when each row was valid. Current values have a `Valid To` equal to '9999-12-31'.



	City Key	WWI City ID	City	State Province	Valid From	Valid To	Lineage Key
1	1	1091	Arden	New York	2013-01-01 00:02:00.0000000	2013-07-01 16:00:00.0000000	47
2	2	20189	Lucasville	Ohio	2013-07-01 16:00:00.0000000	9999-12-31 23:59:59.9999999	47
3	3	20731	Manhattan	New York	2013-07-01 16:00:00.0000000	9999-12-31 23:59:59.9999999	47
4	4	36176	Waxhaw	North Carolina	2014-07-01 16:00:00.0000000	9999-12-31 23:59:59.9999999	47
5	5	29413	Rolling Fields	Kentucky	2013-01-01 00:05:00.0000000	2013-07-01 16:00:00.0000000	47
6	6	3359	Blue Ridge	Alabama	2015-07-01 16:00:00.0000000	9999-12-31 23:59:59.9999999	47
7	7	7780	Coupeville	Washington	2015-07-01 16:00:00.0000000	9999-12-31 23:59:59.9999999	47
8	8	2404	Belen	New Mexico	2015-07-01 16:00:00.0000000	9999-12-31 23:59:59.9999999	47
9	9	29733	Royal Palm Beach	Florida	2013-01-01 00:01:00.0000000	2013-07-01 16:00:00.0000000	47
10	10	17000	Jenkins	Minnesota	2013-01-01 00:02:00.0000000	2014-07-01 16:00:00.0000000	47

The advantage of this approach is that it preserves historical data, which can be valuable for analysis. However, it also means there will be multiple rows for the same entity. For example, here are the records that match `WWI City ID` = 28561:



	City Key	WWI City ID	City	State Province	Valid From	Valid To
1	25321	28561	Redmond	Washington	2013-01-01 00:00:00.0000000	2013-01-01 00:01:00.0000000
2	38479	28561	Redmond	Washington	2015-07-01 16:00:00.0000000	9999-12-31 23:59:59.9999999
3	106171	28561	Redmond	Washington	2013-07-01 16:00:00.0000000	2015-07-01 16:00:00.0000000
4	81223	28561	Redmond	Washington	2013-01-01 00:01:00.0000000	2013-07-01 16:00:00.0000000

For each Sales fact, you want to associate that fact with a single row in City dimension table, corresponding to the invoice date.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

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](#).

For additional security, you can use [Virtual Network service endpoints](#) to secure Azure service resources to only your virtual network. This fully removes public Internet access to those resources, allowing traffic only from your virtual network.

With this approach, you create a VNet in Azure and then create private service endpoints for Azure services. Those services are then restricted to traffic from that virtual

network. You can also reach them from your on-premises network through a gateway.

Be aware of the following limitations:

- If service endpoints are enabled for Azure Storage, PolyBase cannot copy data from Storage into Azure Synapse. There is a mitigation for this issue. For more information, see [Impact of using VNet Service Endpoints with Azure storage](#).
- To move data from on-premises into Azure Storage, you will need to allow public IP addresses from your on-premises or ExpressRoute. For details, see [Securing Azure services to virtual networks](#).
- To enable Analysis Services to read data from Azure Synapse, deploy a Windows VM to the virtual network that contains the Azure Synapse service endpoint. Install [Azure On-premises Data Gateway](#) on this VM. Then connect your Azure Analysis service to the data gateway.

DevOps

- Create separate resource groups for production, development, and test environments. Separate resource groups make it easier to manage deployments, delete test deployments, and assign access rights.
- Put each workload in a separate deployment template and store the resources in source control systems. You can deploy the templates together or individually as part of a CI/CD process, making the automation process easier.

In this architecture, there are three main workloads:

- The data warehouse server, Analysis Services, and related resources.
- Azure Data Factory.
- An on-premises to cloud simulated scenario.

Each workload has its own deployment template.

The data warehouse server is set up and configured by using Azure CLI commands which follows the imperative approach of the IaC practice. Consider using deployment scripts and integrate them in the automation process.

- Consider staging your workloads. Deploy to various stages and run validation checks at each stage before moving to the next stage. That way you can push updates to your production environments in a highly controlled way and minimize unanticipated deployment issues. Use [Blue-green deployment](#) and [Canary releases](#) strategies for updating live production environments.

Have a good rollback strategy for handling failed deployments. For example, you can automatically redeploy an earlier, successful deployment from your deployment history. See the `--rollback-on-error` flag parameter in Azure CLI.

- [Azure Monitor](#) is the recommended option for analyzing the performance of your data warehouse and the entire Azure analytics platform for an integrated monitoring experience. [Azure Synapse Analytics](#) provides a monitoring experience within the Azure portal to show insights to your data warehouse workload. The Azure portal is the recommended tool when monitoring your data warehouse because it provides configurable retention periods, alerts, recommendations, and customizable charts and dashboards for metrics and logs.

For more information, see the DevOps section in [Microsoft Azure Well-Architected Framework](#).

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

Use the [Azure pricing calculator](#) to estimate costs. Here are some considerations for services used in this reference architecture.

Azure Data Factory

Azure Data Factory automates the ELT pipeline. The pipeline moves the data from an on-premises SQL Server database into Azure Synapse. The data is then transformed into a tabular model for analysis. For this scenario, pricing starts from \$ 0.001 activity runs per month that includes activity, trigger, and debug runs. That price is the base charge only for orchestration. You are also charged for execution activities, such as copying data, lookups, and external activities. Each activity is individually priced. You are also charged for pipelines with no associated triggers or runs within the month. All activities are prorated by the minute and rounded up.

Example cost analysis

Consider a use case where there are two lookups activities from two different sources. One takes 1 minute and 2 seconds (rounded up to 2 minutes) and the other one takes 1 minute resulting in total time of 3 minutes. One data copy activity takes 10 minutes. One

stored procedure activity takes 2 minutes. Total activity runs for 4 minutes. Cost is calculated as follows:

Activity runs: $4 * \$0.001 = \0.004

Lookups: $3 * (\$0.005 / 60) = \0.00025

Stored procedure: $2 * (\$0.00025 / 60) = \0.000008

Data copy: $10 * (\$0.25 / 60) * 4 \text{ data integration unit (DIU)} = \0.167

- Total cost per pipeline run: \$0.17.
- Run once per day for 30 days: \$5.1 month.
- Run once per day per 100 tables for 30 days: \$ 510

Every activity has an associated cost. Understand the pricing model and use the [ADF pricing calculator](#) to get a solution optimized not only for performance but also for cost. Manage your costs by starting, stopping, pausing, and scaling your services.

Azure Synapse

Azure Synapse is ideal for intensive workloads with higher query performance and compute scalability needs. You can choose the pay-as-you-go model or use reserved plans of one year (37% savings) or 3 years (65% savings).

Data storage is charged separately. Other services such as disaster recovery and threat detection are also charged separately.

For more information, see [Azure Synapse Pricing](#).

Analysis Services

Pricing for Azure Analysis Services depends on the tier. The reference implementation of this architecture uses the **Developer** tier, which is recommended for evaluation, development, and test scenarios. Other tiers include, the **Basic** tier, which is recommended for small production environment; the **Standard** tier for mission-critical production applications. For more information, see [The right tier when you need it](#).

No charges apply when you pause your instance.

For more information, see [Azure Analysis Services pricing](#).

Blob Storage

Consider using the Azure Storage reserved capacity feature to lower cost on storage. With this model, you get a discount if you can commit to reservation for fixed storage capacity for one or three years. For more information, see [Optimize costs for Blob storage with reserved capacity](#).

For more information, see the Cost section in [Microsoft Azure Well-Architected Framework](#).

Next steps

- [Introduction to Azure Synapse Analytics](#)
- [Get Started with Azure Synapse Analytics](#)
- [Introduction to Azure Data Factory](#)
- [What is Azure Data Factory?](#)
- [Azure Data Factory tutorials](#)

Related resources

You may want to review the following [Azure example scenarios](#) that demonstrate specific solutions using some of the same technologies:

- [Data warehousing and analytics for sales and marketing](#)
- [Enterprise BI in Azure with Azure Synapse](#).

Data analytics for automotive test fleets

Azure Blob Storage

Azure Data Explorer

Azure Event Hubs

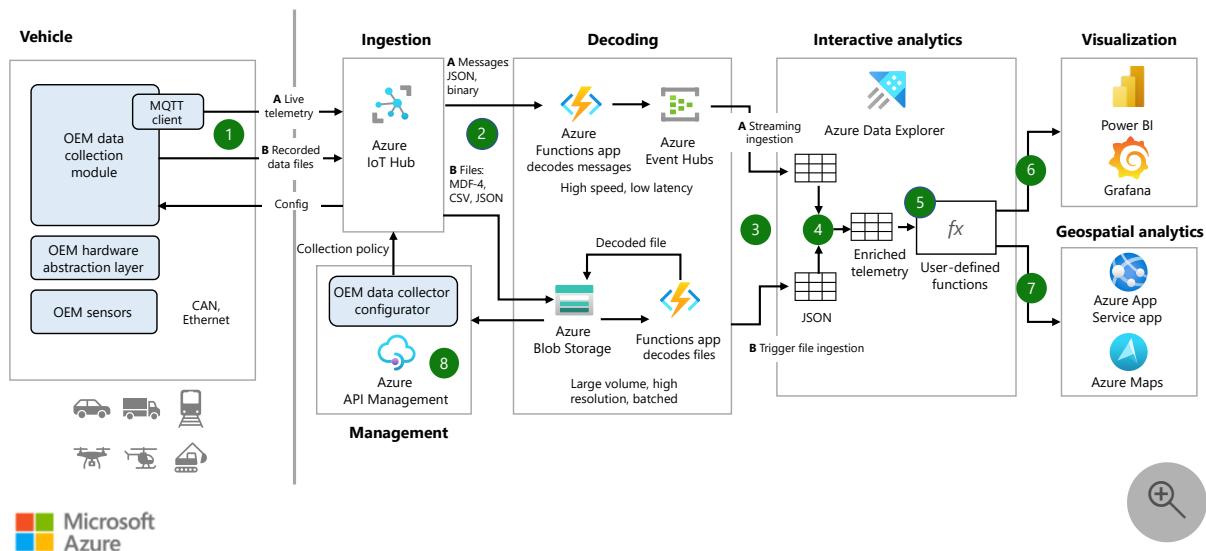
Azure Functions

Azure IoT Hub

Automotive OEMs need solutions to minimize the time between doing test drives and getting test drive diagnostic data to R&D engineers. As vehicles become more automated, software lifecycles are shorter, and digital feedback loops must become faster. New technology can democratize data access and provide R&D engineers with near real-time insights into test drive diagnostic data. Secure data sharing can enhance collaboration between OEMs and suppliers, further shortening development cycles.

This example workload relates to both telemetry and batch test drive data ingestion scenarios. The workload focuses on the data platform that processes diagnostic data, and the connectors for visualization and reporting.

Architecture



Download a [PowerPoint file](#) with all the diagrams in this article.

Dataflow

1. Azure IoT Hub ingests live, raw telemetry data (A) and uploads recorded data files (B) from the vehicle.

2. IoT Hub sends the live telemetry (A) to an Azure Functions app that decodes the telemetry to JavaScript Object Notation (JSON) and posts it to Azure Event Hubs.

IoT Hub sends the recorded data files (B) to Azure Blob Storage. A completed file upload triggers a Functions app that decodes the data and writes the decoded file into Blob Storage in a comma-separated values (CSV) format suitable for ingestion.

3. Azure Data Explorer ingests decoded JSON telemetry data from Event Hubs (A) into a raw telemetry table, and ingests the decoded CSV files (B) from Blob Storage.

4. Azure Data Explorer uses the `Update` function to expand the JSON data into a suitable row format and to enrich the data. For example, the function clusters location data to support geospatial analytics.

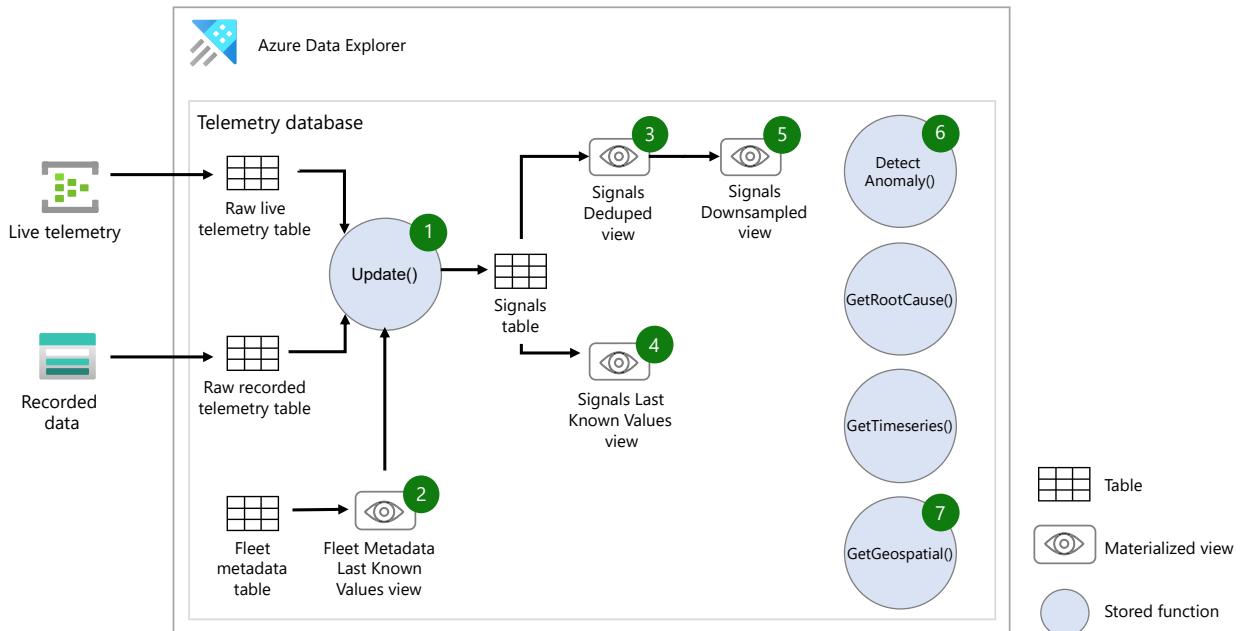
5. Data scientists and R&D engineers use Kusto Query Language (KQL) capabilities to build analytics use cases that they store as user-defined functions. KQL functions include aggregation, time series analysis, geospatial clustering, windowing, and machine learning (ML) plugins.

6. Power BI uses Dynamic Query to create visualizations with the user-defined queries. The Grafana data source plugin for Azure Data Explorer uses the user-defined queries for near real-time updates.

7. An Azure App Service app uses Azure Maps data source rendering capabilities to visualize user-defined query results that use GeoJSON format.

8. Azure API Management provides access to stored raw data files from vehicles, and a configuration API that manages third-party data collection policies.

Azure Data Explorer schema



1. The `Update()` function uses methods such as:

- `mv-expand()` to expand complex values stored in JSON structures into rows with individual signals.
- `geo_point_to_h3cell()` or `geo_point_to_geohash()` to convert latitude and longitude to geohashes for geospatial analytics.
- `toDouble()` and `toString()` to cast extracted values from dynamic JSON objects into the appropriate data types.

2. The **Fleet Metadata Last Known Values** view joins other views as part of ingestion to provide context. The historical fleet metadata is useful if new use cases require reprocessing of the raw telemetry.

3. If necessary, a **Signals Deduped** materialized view uses `take_any()` to deduplicate signals.

4. The **Signals Last Known Values** materialized view uses `arg_max()` on the timestamp for real-time reporting.

5. The **Signals Downsampled** materialized view aggregates signals by using predefined bins such as *hourly* and *daily* to simplify reporting across the fleet.

6. Stored plugin functions like `DetectAnomaly()` find anomalies in data series. ML plugins like autocluster find common patterns of discrete attributes.

7. The `GetGeospatial()` function generates GeoJSON files that contain grouped signals by geohashes.

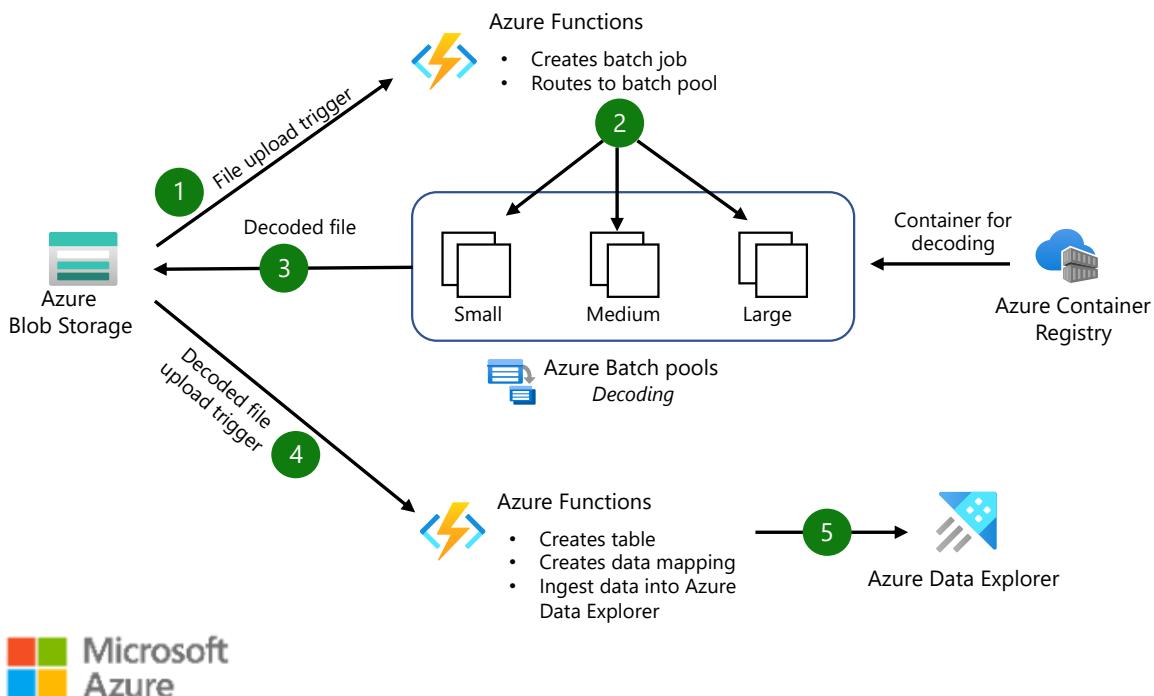
Components

The following key technologies implement this workload:

- [Azure Data Explorer](#)
- [Azure IoT Hub](#)
- [Azure Blob Storage](#)
- [Azure Event Hubs](#)
- [Azure Functions](#)
- [Azure Managed Grafana](#)
- [Azure App Service](#)
- [Azure Maps](#)
- [Azure API Management](#)
- [Power BI](#)

Alternatives

[Azure Batch](#) is a good alternative for complex file decoding. This scenario involves large numbers of files over 300 megabytes that require different decoding algorithms based on file version or type.



1. Uploading a recorded data file to Blob Storage triggers a Functions app to schedule decoding.
2. The Functions app creates a batch job, taking into consideration the file type, size, and required decoding algorithm. The app selects a suitable virtual machine (VM)

from the pool and starts the job.

3. When the job completes, Batch writes the resulting decoded file back to Blob Storage. This file must be suitable for direct ingestion in a format that Azure Data Explorer supports.
4. Uploading a decoded signal file to Blob Storage triggers a function that ingests the data into Azure Data Explorer. This function creates the table and data mapping if necessary, and starts the ingestion process.
5. Azure Data Explorer directly ingests the data files from Blob Storage.

This approach offers the following benefits:

- Azure Functions and Batch pools are able to handle scalable data processing tasks robustly and efficiently.
- Batch pools provide insight into processing statistics, task queues, and batch pool health. You can visualize status, detect problems, and rerun failed tasks.
- The combination of Azure Functions and Azure Batch supports plug-and-play processing in Docker containers.

Scenario details

Automotive OEMs use large fleets of prototype and test vehicles to test and verify all kinds of vehicle functions. Test procedures are expensive, because real drivers and vehicles need to be involved, and certain specific real-world road testing scenarios must pass multiple times. Integration testing is especially important to evaluate interactions between electrical, electronic, and mechanical components in complex systems.

To validate vehicle functions and analyze anomalies and failures, gigabytes of diagnostic data must be captured from electronic control unit (ECUs), computer nodes, vehicle communication buses like Controller Area Network (CAN) and Ethernet, and sensors. In the past, small data logger servers in the vehicles stored diagnostic data locally as master database (MDF), multimedia fusion extension (MFX), CSV, or JSON files. After test drives were complete, the servers uploaded diagnostic data to data centers, which processed it and provided it to R&D engineers for analytics. This process could take hours or sometimes days. More recent scenarios use telemetry ingestion patterns like Message Queuing Telemetry Transport (MQTT)-based synchronous data streams, or near real-time file uploads.

Potential use cases

- Vehicle management evaluates the performance and collected data per vehicle across multiple test scenarios.

- System and component validation uses collected vehicle data to verify that the behavior of vehicle components falls within operational boundaries across trips.
- Anomaly detection locates deviation patterns of a sensor value relative to its typical baseline pattern in real time.
- Root cause analysis uses ML plugins such as clustering algorithms to identify changes in the distribution of values on multiple dimensions.
- Predictive maintenance combines multiple data sources, enriched location data, and telemetry to predict component time to failure.
- Sustainability evaluation uses driver behavior and energy consumption to evaluate the environmental impact of vehicle operations.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is 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 your application can meet the commitments you make to your customers. For more information, see [Overview of the reliability pillar](#).

- [Azure availability zones](#) are unique physical locations within the same Azure region. Availability zones can protect Azure Data Explorer compute clusters and data from partial region failure.
- [Business continuity and disaster recovery \(BCDR\)](#) in Azure Data Explorer lets your business continue operating in the face of disruption.
- Consider using a [follower database](#) in Azure Data Explorer to separate compute resources between production and non-production use cases.

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](#).

It's important to understand the division of responsibility between the automotive OEM and Microsoft. In the vehicle, the OEM owns the whole stack, but as the data moves to the cloud, some responsibilities transfer to Microsoft. Azure platform-as-a-service (PaaS) provides built-in security on the physical stack, including the operating system. You can apply the following capabilities on top of the infrastructure security components.

- Private endpoints for network security. For more information, see [Private endpoints for Azure Data Explorer](#) and [Allow access to Azure Event Hubs namespaces via private endpoints](#).
- Encryption at rest and in transit.
- Identity and access management that uses Microsoft Entra identities and [Microsoft Entra Conditional Access](#) policies.
- [Row Level Security \(RLS\)](#) for Azure Data Explorer.
- Infrastructure governance that uses [Azure Policy](#).
- Data governance that uses [Microsoft Purview](#).

All these features help automotive OEMs create a safe environment for their vehicle telemetry data. For more information, see [Security in Azure Data Explorer](#).

Cost optimization

Cost optimization looks at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

This solution uses the following practices to help optimize costs:

- Correctly configure hot caches and cold storage for the Raw and Signals tables. The hot data cache is stored in RAM or SSD and provides improved performance. Cold data, however, is 45 times cheaper. Set a hot cache policy that's adequate for your use case, such as 30 days.
- Set up a retention policy on the Raw and Signals tables. Determine when the signal data is no longer relevant, for example after 365 days, and set the retention policy accordingly.
- Consider which signals are relevant for analysis.
- Use materialized views when querying the signals last known values, signals deduped, and signals downsampled. Materialized views consume fewer resources than doing source table aggregations on each query.
- Consider your real-time data analytics needs. Setting up streaming ingestion for the live telemetry table enables latency of less than one second between ingestion and query, but at a higher cost of more CPU cycles.

Performance efficiency

Performance efficiency is your workload's ability to scale efficiently to meet user demands. For more information, see [Performance efficiency pillar overview](#).

- If the number and size of recorded data files is greater than 1,000 files or 300 MB a day, consider using Azure Batch for decoding.
- Consider performing common calculations and analysis after ingest and storing them in additional tables.

Deploy this scenario

To deploy Azure Data Explorer and ingest MDF files, you can follow the [step-by-step](#) tutorial demonstrating how to deploy a free instance, parse MDF files, ingest and perform some basic queries.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Frank Kaleck](#) | Digital Architect Automotive
- [Mario Ortegon-Cabrera](#) | Principal Program Manager
- [Henning Rauch](#) | Principal Program Manager
- [Boris Scholl](#) | Partner, Chief Architect

Other contributors:

- [Hans-Peter Bareiner](#) | Cloud Solution Architect
- [Jason Bouska](#) | Sr. Software Engineer

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Ingest data from event hub into Azure Data Explorer](#)
- [Upload files with IoT Hub](#)
- [Materialized views](#)
- [Visualize data from Azure Data Explorer in Grafana](#)
- [Visualize data from Azure Data Explorer in Power BI](#)
- [Create a data source for a map in Microsoft Azure Maps](#)

Related resources

- Big data analytics with Azure Data Explorer
- Predictive insights with vehicle telematics
- Automated guided vehicles fleet control
- Building blocks for autonomous-driving simulation environments
- Process real-time vehicle data using IoT

Azure Synapse Analytics for landing zones

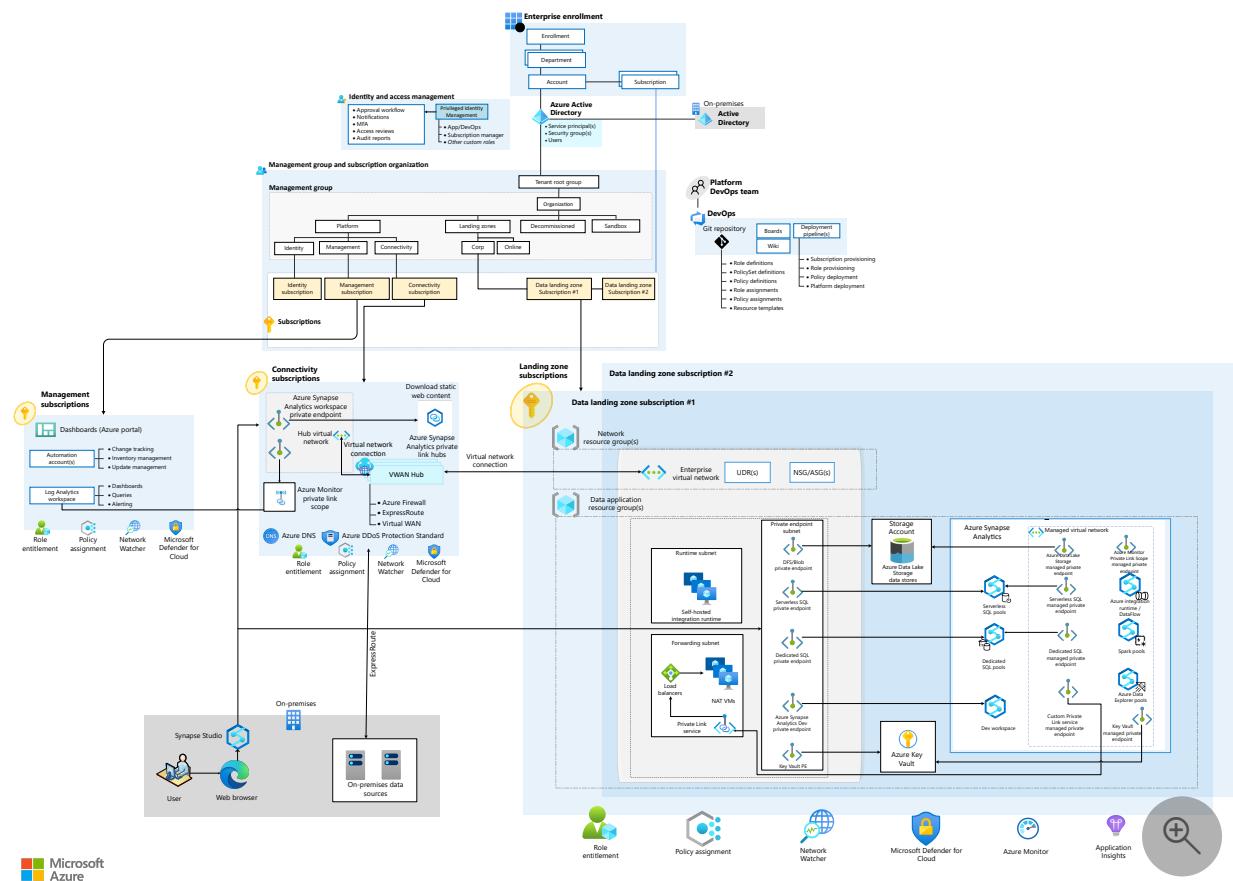
Azure Synapse Analytics Azure Private Link Azure Data Lake Storage Azure Key Vault

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 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](#)
- [Logical data warehouse with Azure Synapse serverless SQL pools](#)

Big data analytics on confidential computing with Apache Spark on Kubernetes

Azure Kubernetes Service (AKS)

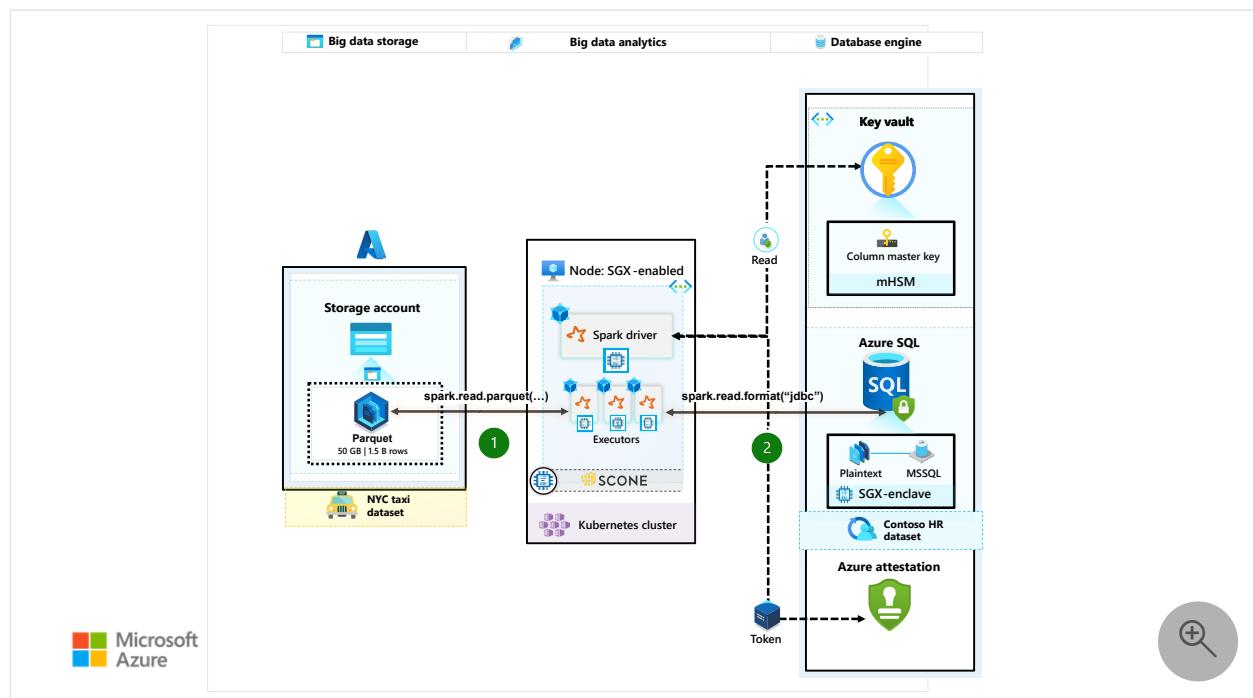
Azure SQL Database

Azure Data Lake

This solution uses confidential computing on Kubernetes to run big data analytics with Apache Spark inside confidential containers with data from Azure Data Lake and Azure SQL Database. Confidential computing is provided by Intel Software Guard Extensions and AMD EPYC™ processors with Secure Encrypted Virtualization-Secure Nested Paging. For more information on provisioning an AKS cluster with AMD SEV-SNP confidential VMs, see [Confidential VM node pool support on AKS with AMD SEV-SNP confidential VMs](#). For more information about deploying an AKS cluster with confidential computing Intel SGX agent nodes, see [Deploy an AKS cluster with confidential computing Intel SGX agent nodes by using the Azure CLI](#).

Apache®, Apache Ignite, Ignite, 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 PowerPoint file](#) of this architecture.

The preceding diagram outlines the architecture: a scalable pattern for processing larger datasets in a distributed fashion. It also showcases confidential analytics on relational database engines and storing confidential data. In particular, the containerized Spark app can process datasets from two data sources, as illustrated:

1. [Azure Data Lake Storage - Parquet/Delta Lake files](#): As shown in the [sample demonstration](#), a four-pod Spark deployment—one Driver, three Executors on the [Secure Container Environment \(SCONE\) runtime](#)—is capable of processing 1.5 billion rows of Parquet/Delta Lake files that are stored on Azure Data Lake storage within two minutes, or approximately 131 seconds.
2. [Azure SQL DB - Always Encrypted with secure enclaves](#): This example uses Spark to access Always Encrypted data as plaintext by using the [Azure SQL JDBC driver](#) inside the Spark container enclave to run analytics and machine learning pipelines.

You can easily extend this pattern to include any data sources that Spark's large ecosystem supports.

Workflow

1. Operator persona: A DevOps engineer provisions Kubernetes clusters, [Namespaces](#), [Service Accounts](#), and Confidential virtual machine (VM) node pools (for example, [DC4s_v3](#)).
2. Developer persona: A data engineer uses [PySpark](#) to write an analytics application that's designed to analyze large volumes of data.
3. Data custodian persona: The data or security engineer creates a security policy for the PySpark application from a shared repository in the organization (a one-time activity). This policy specifies the expected state of the data and app code, the minimum security requirements for the platform, and any environment variables, command-line arguments, or secrets (such as the JDBC string, input blob URI, and a SAS token for access). You can also make this configuration available to the Spark runtime by using Kubernetes [Secrets](#) or by using Azure Key Vault. (For more information, see [Use the Azure Key Vault Provider for Secrets Store CSI Driver in an AKS cluster](#)). The configuration is injected into the enclave only if the evidence that it provides is validated by an attestation provider. The [attestation provider](#) (for example, [Azure Attestation Service](#)), is also defined in the security policy.
4. With the help of the SCONE confidential computing software, the data engineer builds a confidential Docker image that contains the encrypted analytics code and

a secure version of PySpark. SCONE works within an AKS cluster that has Intel SGX enabled (see [Create an AKS cluster with a system node pool](#)), which allows the container to run inside an enclave. PySpark provides evidence that the sensitive data and app code is encrypted and isolated in a Trusted Execution Environment (TEE)—which means that no humans, no processes, and no logs have access to the plaintext data or the application code.

5. The PySpark application is deployed to the remote AKS cluster. It starts and sends its attestation evidence to the attestation provider. If the evidence is valid, an *attestation token* is returned. The remote infrastructure accepts the attestation token and verifies it with a public certificate that's found in the Azure Attestation service. If the token is verified, there's near certainty that the enclave is safe and that neither the data nor the app code have been opened outside the enclave. The configuration in the security policy (environment variables, command-line arguments, and secrets) is then injected into PySpark enclaves.
6. You can horizontally scale the PySpark execution across several Kubernetes nodes. All PySpark instances communicate over an encrypted channel, and all the files are encrypted that need to be written to their local file systems (for example, shuffle files).
7. The results of the analysis are encrypted and uploaded to an [Azure SQL Database with Always Encrypted](#) (that uses column-level encryption). Access to the output data and encryption keys can be securely granted to other confidential applications (for example, in a pipeline) by using the same sort of security policies and hardware-based attestation evidence that's described in this article.

Components

- [Azure Attestation](#) is a unified solution that remotely verifies the trustworthiness of a platform. Azure Attestation also remotely verifies the integrity of the binaries that run in the platform. Use Azure Attestation to establish trust with the confidential application.
- [Azure confidential computing](#) nodes are hosted on a specific VM series that can run sensitive workloads on AKS within a hardware-based TEE. In this environment, user-level code can allocate private regions of memory, known as [enclaves](#). Confidential computing nodes can support confidential containers or enclave-aware containers.
- [Azure Kubernetes Service](#) simplifies the process of deploying and managing a Kubernetes cluster.

- [Apache Spark](#) is an open-source, multi-language engine for executing data engineering, data science, and machine learning on both single-node machines and multi-node clusters, such as Kubernetes pods.
- [Azure SQL Database](#) now offers [Always Encrypted with secure enclaves](#), expanding the confidential computing capabilities of [SQL Server's Always Encrypted technology](#) to include in-place encryption and rich confidential queries.
- [SCONE](#) supports the execution of confidential applications in containers that run inside a Kubernetes cluster.
- [SCONE platform](#) is a solution from Scontain, an independent software vendor and Azure partner.

Alternatives

[Occlum](#) is a memory-safe, multi-process library OS (LibOS) for Intel SGX. Occlum makes it possible for legacy applications to run on Intel SGX with little to no modifications to source code. Occlum transparently protects the confidentiality of user workloads while allowing easy migration to existing Docker applications. Occlum supports Java apps.

The SCONE engineering team maintains an [Apache Spark](#) container image that runs the latest version of Spark. An alternative that isn't specific to Apache Spark is [Fortanix](#), with which you can deploy confidential containers to use with your containerized application. Fortanix provides the flexibility required to run and manage the broadest set of applications: existing applications, new enclave-native applications, and pre-packaged applications.

Scenario details

There's exponential growth of datasets, which has resulted in growing scrutiny of how data is exposed from the perspectives of both consumer data privacy and compliance. In this context, confidential computing becomes an important tool to help organizations meet their privacy and security needs for business and consumer data. Organizations can gain new insights from regulated data if the data is processed in a compliant manner. Confidential computing is especially helpful in scenarios where the scale that's provided by cloud computing is needed to process the data confidentially.

Confidential computing technology encrypts data in memory and only processes it after the cloud environment is verified, or *attested*. Confidential computing prevents data access by cloud operators, malicious admins, and privileged software, such as the

hypervisor. It also helps to keep data protected throughout its lifecycle—while the data is at rest, in transit, and also now while it's in use.

[Confidential containers](#) on Azure Kubernetes Service (AKS) provide the necessary infrastructure for customers to use popular applications, such as [Apache Spark](#), to perform data cleansing and machine learning training. This article presents a solution that Azure confidential computing offers for running an Apache Spark application on an AKS cluster by using node pools with Intel Software Guard Extensions (Intel SGX). The data from that processing is safely stored in Azure SQL Database by using [Always Encrypted with secure enclaves](#).

Note

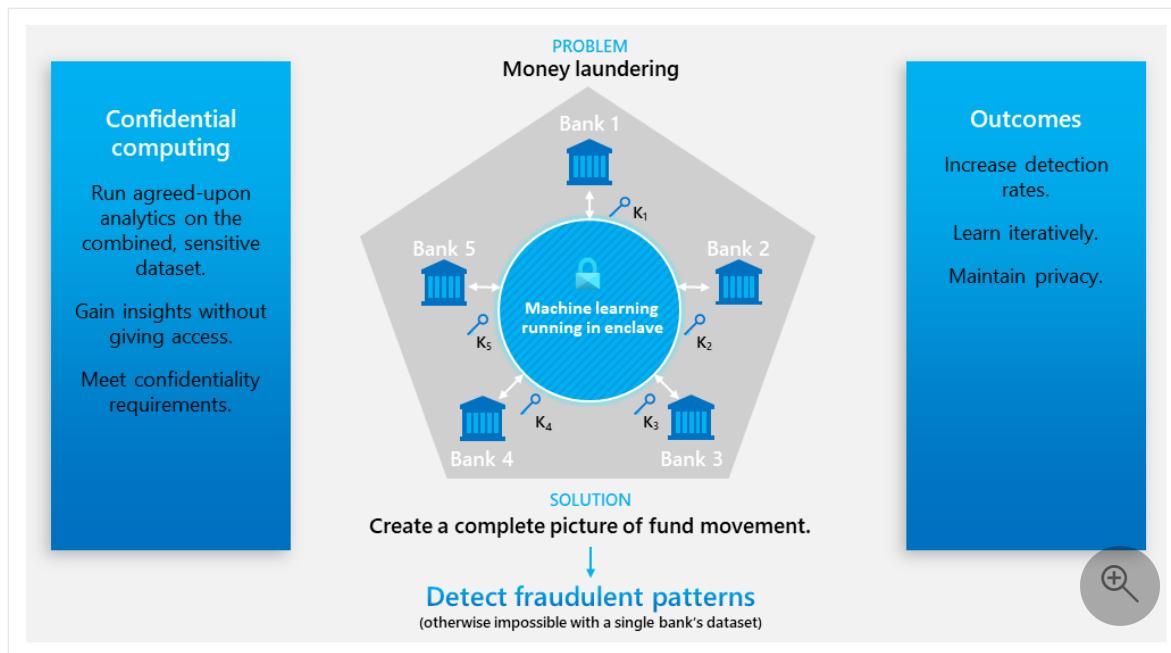
Confidential data analytics in this context is meant to imply *run analytics on sensitive data with peace of mind against data exfiltration*. This includes a potential container access breach at the root level, both internally (for example, by a rogue admin) or externally (by system compromise).

Confidential data analytics helps to meet the highest needs of security and confidentiality by removing from computation the untrusted parties, such as the cloud operator and service or guest admins. This method helps to meet data compliance needs through hardware-backed guarantees.

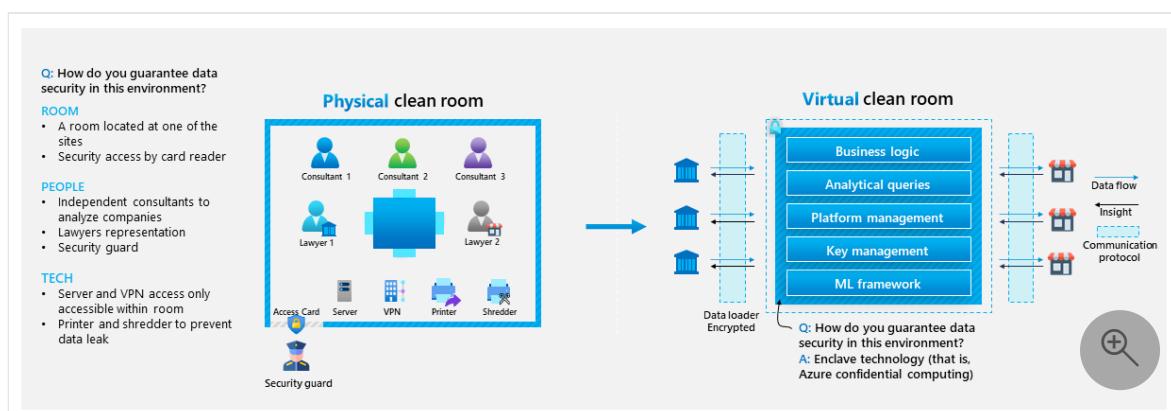
Potential use cases

Many industries, especially financial services, protect their data by using confidential computing for these purposes:

- Extending data confidentiality to cross-organization datasets (that is, multi-party computation):



- Providing collective insights across organizational boundaries by using a virtual clean room:



- Securing financial data or regulated data from a cloud operator.
- Meeting high requirements for data privacy and protection.
- Running ML model training and inferencing on sensitive data.
- Refining datasets by using familiar data preparation tools.
- Protecting container data and code integrity.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Azure confidential enclaves that use [DCsv3](#) and [DCdsv3-series](#) VMs offer large memory sizes to help run memory-intensive applications like analytics. This scenario uses Intel SGX-enabled DCsv3-series VMs. You can only deploy certain sizes in certain regions. For more information, see [Quickstart: Deploy an Azure Confidential Computing VM in the Marketplace](#) and [Products available by region](#).

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](#).

Two primary factors in security for this scenario are secure enclaves and attestation.

Enclave assurances

Kubernetes admins, or any privileged user with the highest level of access (for example, root), can't inspect the in-memory contents or source code of drivers or executors. Enclave page cache (EPC) is a specialized memory partition in Azure Confidential VMs that enclaves or confidential containers use. DCsv3 and DCdsv3-series VMs also come with regular, unencrypted memory to run apps that don't require the secure enclave. For more information about using Intel SGX for enclaves, see [Build with SGX enclaves](#).

Attestation

Attestation is a mechanism that provides to a client, or *party*, cryptographic evidence that the environment where an app is running is trustworthy, including both its hardware and software, before exchanging data. *Remote attestation* ensures that your workload hasn't been tampered with when deployed to an untrusted host, such as a VM instance or a Kubernetes node that runs in the cloud. In this process, attestation evidence provided by Intel SGX hardware is analyzed by an attestation provider.

To perform remote attestation on a SCONE application (such as Spark Driver and Executor pods), two services are required:

- **Local attestation service (LAS):** A local service that runs on the untrusted host (AKS node pool VM) and gathers the attestation evidence that's provided by Intel SGX about the application being attested. Because of SCONE's method of app deployment, this evidence is signed and forwarded to the configuration and attestation service (CAS).
- **CAS:** A central service that manages security policies (called *SCONE sessions*), configuration, and secrets. CAS compares the attestation evidence that's gathered

by LAS against the application's security policies (which are defined by the application owner) to decide whether the enclave is trustworthy. If it is, CAS allows the enclave to run, and SCONE securely injects configuration and secrets into it. To learn more about CAS and its features, such as secret generation and access control, see [SCONE Configuration and Attestation Service](#).

This scenario uses a [public CAS](#) provided by SCONE for demonstration and simplicity, and it deploys the [LAS](#) to run as a [DaemonSet](#) on each AKS node.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

To explore the cost of running this scenario, use the [Azure pricing calculator](#), which preconfigures all Azure services. Please note the additional licenses that are required by the partner to run production workloads.

Deploy this scenario

Deploying this scenario involves the following high-level steps:

- Get access to the PySpark base image that's used in this scenario from SCONE's container registry: see [registry.scontain.com:5050](#) on [SCONE curated images](#).
- Clone the demo project on GitHub, [Confidential Data Analytics with Apache Spark on Intel SGX Confidential Containers](#). This project contains all the needed resources, deployment steps, and source-code to reproduce the demo.
- Deploy [Always Encrypted with secure enclaves in Azure SQL Database - Demos](#). These demos use a confidential dataset, ContosoHR, which is included. This scenario decrypts confidential data into plaintext inside the Spark containers enclave.
- Deploy an Intel SGX-enabled AKS cluster node pool. For instructions, see [Quickstart: Deploy an AKS cluster with confidential computing nodes by using the Azure CLI](#).
- Deploy the SCONE Local Attestation Service to the cluster by using the included Kubernetes manifest.

- Build the encrypted image with SCONE confidential computing software and push it to your own Azure Container Registry. The repo has a demo application that counts the number of lines in New York City's [Yellow Taxi trip records](#), an open dataset of times, locations, fares, and other data that's related to taxi trips. You can adapt this to your specific needs.
- Deploy the Spark application by running the command `spark-submit`. This deploys a driver pod and a configurable number of executor pods (the demo uses three) that run the tasks and report the analysis results to the driver. All communication is encrypted.

Alternatively, SCONE Confidential PySpark on Kubernetes, a VM, includes the same demo that you can reproduce in a local [minikube](#) cluster. For more information, see the official documentation: [SCONE PySpark virtual machine](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Amar Gowda](#) | Principal Program Manager

Other contributor:

- [Gary Moore](#) | Programmer/Writer

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Azure confidential computing](#)
- [Confidential containers on AKS](#)

Related resources

- [Attestation, authentication, and provisioning](#)
- [Big data analytics with enterprise-grade security using Azure Synapse](#)
- [Confidential computing on a healthcare platform](#)
- [Multiparty computing with Azure services](#)

Customer 360 with Azure Synapse and Dynamics 365 Customer Insights

Customer Insights - Data

Azure Synapse Analytics

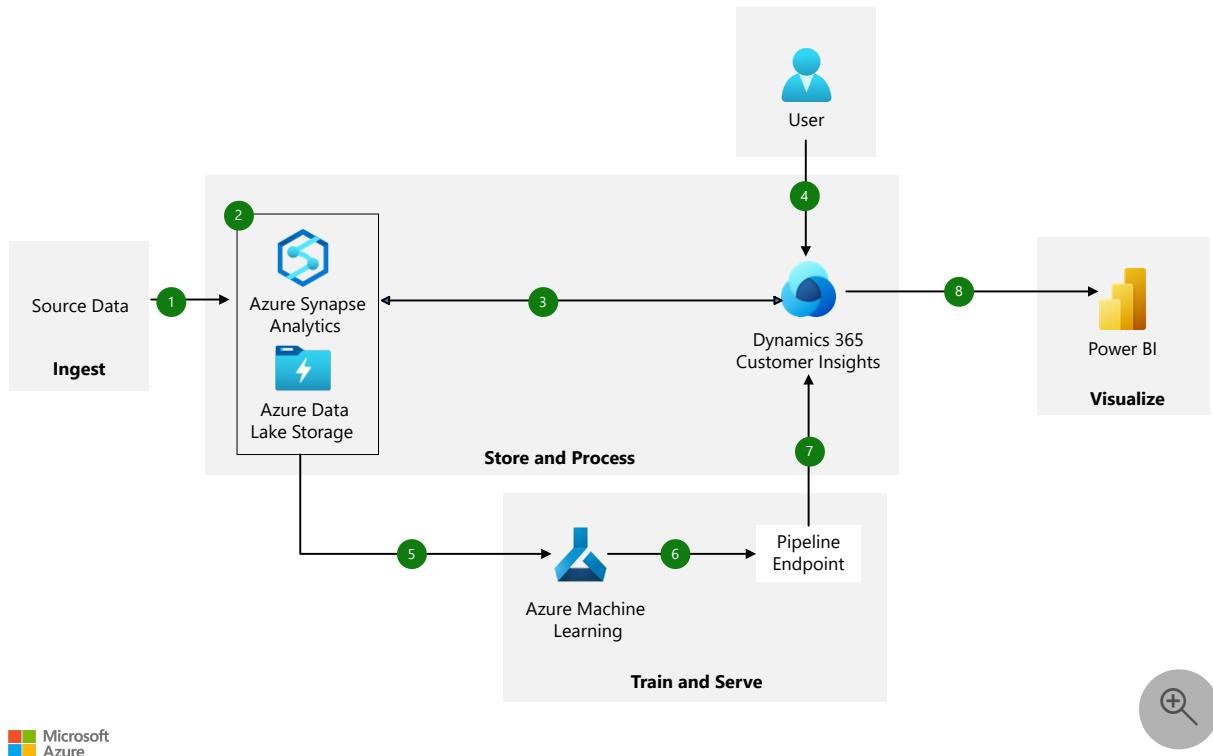
Azure Machine Learning

Power BI

This solution combines Azure Synapse Analytics with Dynamics 365 Customer Insights, to build a comprehensive view that presents your customer data and to provide the best customer experience.

Apache®, Apache Ignite, Ignite, 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

1. Azure Synapse Analytics ingests raw source data by using Azure Synapse pipelines.
2. Source data is stored in Azure Synapse Analytics and Azure Data Lake Storage Gen2.
3. Dynamics 365 Customer Insights connects to customer data from Azure Synapse.
4. Administrators configure unified customer profiles in Customer Insights, together with measures, segments, and enrichments. Unified customer profiles are ported from Customer Insights to Azure Synapse.
5. Administrators use the unified customer profile in Azure Synapse to create an Azure Machine Learning pipeline for retention prediction.
6. Administrators create a retention prediction model endpoint.
7. Administrators create a custom model workflow in Customer Insights to call the Azure Machine Learning pipeline to get the retention predictions.
8. Power BI ingests the Customer 360 data from Customer Insights to visualize the profiles and metrics.

Components

- [Dynamics 365 Customer Insights](#) can help you provide unmatched customer experiences by using world-class AI and analytics. Here, it's used to unify, segment, and enrich customer data.
- [Azure Synapse Analytics](#) is an analytics service that brings together data integration, enterprise data warehousing, and big data analytics. It's used here for data ingestion, storage, and processing.
- [Data Lake Storage](#) provides a massively scalable and secure data lake for your high-performance analytics workloads.
- [Azure Machine Learning](#) is an end-to-end machine learning service. It's used in this architecture to predict customer retention.
- [Power BI](#) can help you turn your data into coherent, visually immersive, and interactive insights. It's used here to visualize customer profiles and metrics.

Scenario details

Managing customer data from multiple sources and building a unified Customer 360 view isn't a new challenge. But it is becoming increasingly difficult with the increased number of interaction channels and touchpoints with customers. By combining Azure Synapse Analytics with Dynamics 365 Customer Insights, you can build a comprehensive view of your customers to provide the best customer experience.

Potential use cases

This solution was created for a property management organization. It can also be applied in industries like retail, financial services, manufacturing, and health care. It can be used by any organization that needs to bring data together across systems to build a Customer 360 profile and improve the customer experience.

You can use this solution to:

- Gain better insights from your customer data.
- Target sources of customer churn or dissatisfaction.
- Direct account and customer service activities.
- Run targeted promotions that are aimed at customer retention or upselling.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

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 solution uses Microsoft Entra ID to authenticate users to the Azure solutions in the architecture. You can manage permissions via Microsoft Entra authentication or role-based access control.

Follow these security guidelines when you implement this solution:

- [Security in Azure](#)
- [Access control for Azure Synapse](#)
- [User permissions for Customer Insights](#)

Scalability

This solution uses Azure Synapse Spark clusters, which can be automatically scaled up and down based on the activity needs of your workload. For more information, see [Azure Synapse Spark cluster autoscaling](#).

Azure Machine Learning training pipelines can be scaled up and down based on data size and other configuration parameters. The compute clusters support autoscaling and automatic shutdown to optimize for performance and cost.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

[Dynamics 365 Customer Insights](#) license pricing options are based on the number of customer profiles needed.

[Azure Synapse Analytics](#) has various pricing options to help you optimize costs. You can perform big data processing tasks like data engineering, data preparation, and machine learning directly in Azure Synapse by using memory-optimized or hardware-accelerated Apache Spark pools. Billing for usage of Spark pools is rounded up to the nearest minute.

[Azure Machine Learning](#) has no additional license charge. However, there are charges for compute and other Azure services that you consume, including but not limited to Azure Blob Storage, Azure Key Vault, Azure Container Registry, and Application Insights.

There are various [Power BI](#) product options to meet different requirements. [Power BI Embedded](#) provides an Azure-based option for embedding Power BI functionality in your applications.

You can deploy this solution with the following options.

- Dynamics 365 Customer Insights: 1,500 profiles
- Azure Synapse Analytics: Memory-optimized Spark cluster of medium size (8 vCores / 64 GB)
- Azure Machine Learning
 - Compute instance of type Standard_DS11_v2
 - Compute cluster of type Standard_D2_v2

Azure services like Azure Storage accounts, Key Vault, Container Registry, Application Insights, and so on, that are deployed with Azure Synapse Analytics and Azure Machine Learning incur other costs.

Deploy this scenario

To deploy this solution, follow the steps in the [Getting Started guide](#) and the step-by-step [Deployment Guide](#). You can find them in the [GitHub repository](#) for the solution.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Nalini Chandhi](#) | Sr. Technical Specialist

Next steps

- [Unlock customer intent with Dynamics 365 Customer Insights](#)
- [Product overview for Dynamics 365 Customer Insights](#)
- [What is Azure Machine Learning?](#)
- [What is Azure Synapse Analytics?](#)

Related resources

- [Enhanced customer dimension with Dynamics 365 Customer Insights](#)
- [Modern data warehouse for small and medium businesses](#)
- [Clinical insights with Microsoft Cloud for Healthcare](#)
- [Analytics architecture design](#)

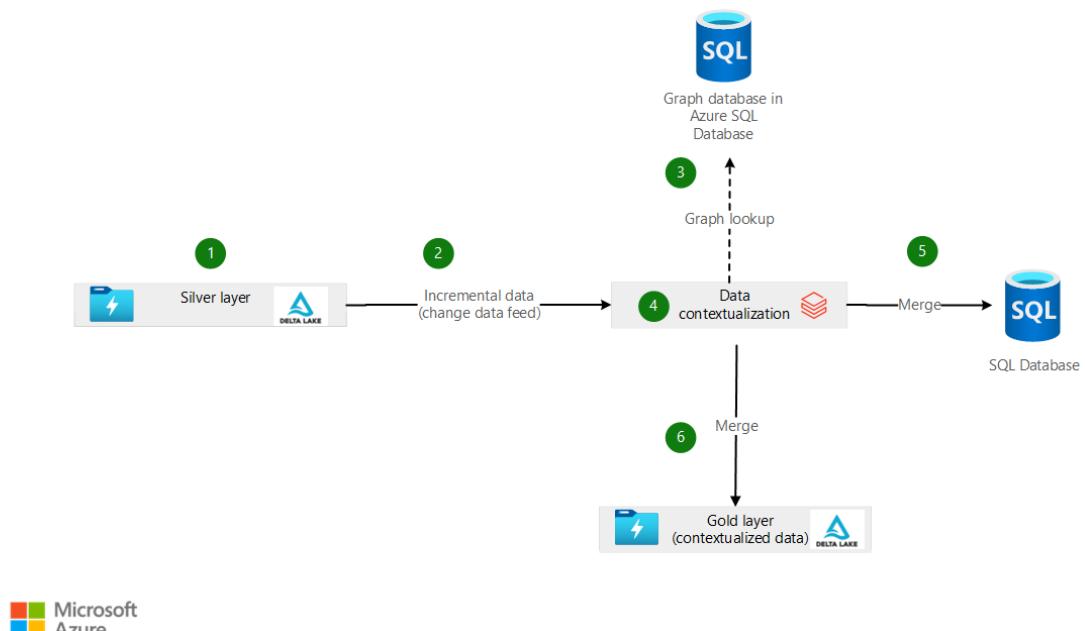
Contextualize data by using graph in SQL Database

Azure SQL Database Azure Databricks Azure Data Lake Storage

Data contextualization is the process of adding contextual information to raw data in order to enhance its meaning and relevance. It involves the use of additional information like metadata, annotations, and other relevant details to provide a better understanding of the data. Contextualization can help analysts understand the relationships between data points and the environment in which they were collected. For example, contextualization can provide information about the time, location, and other environmental factors that might have influenced the data. In data processing, contextualization is becoming increasingly important as datasets become larger and more complex. Without proper contextualization, it can be difficult to interpret data accurately and make informed decisions based on it.

This article demonstrates how to contextualize data by looking up relevant context that's stored in a graph database in Azure SQL Database.

Architecture



Microsoft
Azure



Download a [Visio file](#) of this architecture.

In this architecture, data stored in Delta Lake in the silver layer is read incrementally, contextualized based on a graph lookup, and merged into another Delta Lake instance in the gold layer. That data can be stored as a data mart in SQL Database if another system requires it.

Dataflow

The following dataflow corresponds to the preceding diagram:

1. The incoming data that needs to be contextualized is appended into the Delta table in the silver layer.
2. The incoming data is incrementally loaded into Azure Databricks.
3. Contextual information is retrieved from a graph database.
4. The incoming data is contextualized.
5. Optionally, the contextualized data is merged into the corresponding table in SQL Database.
6. The contextualized data is appended into the corresponding Delta table in the gold layer.

Components

- [Azure Data Lake Storage](#) is a scalable data lake for high-performance analytics workloads. In this solution, it stores input data and contextualized data in Delta tables.
- [Azure Databricks](#) is a unified set of tools for building, deploying, sharing, and maintaining enterprise-grade data solutions at scale. In this solution, it provides the platform on which Python notebook files are used to contextualize data.
- [SQL Database](#) is an always-up-to-date, fully managed relational database service that's built for the cloud. In this solution, it stores graph data and contextualized data.

Graph database alternatives

Many other graph databases are available. For more information, see:

- [Graph processing with SQL Database](#)
- [Azure Cosmos DB for Apache Gremlin](#)
- [Neo4J](#)
- [RedisGraph](#)
- [Apache Age for PostgreSQL](#)

There are pros and cons associated with each of these products and services. Some of them are Azure managed services, and some aren't. This architecture uses SQL Database, because:

- It's an Azure-managed relational database service that has graph capabilities.
- It's easy to get started if you're familiar with SQL Database and SQL Server.
- Solutions often benefit from the use of Transact-SQL in parallel. SQL Database graph relationships are integrated into Transact-SQL.

Scenario details

Data layers

This solution is based on the Databricks [medallion architecture](#). In this design pattern, data is logically organized in various layers. The goal is to incrementally and progressively improve the structure and quality of the data as it moves from one layer to the next.

For simplicity, this architecture has only two layers:

- The silver layer stores the input data.
- The gold layer stores the contextualized data.

The data in the silver layer is stored in [Delta Lake](#) and exposed as Delta tables.

Incremental data load

This solution implements incremental data processing, so only data that has been modified or added since the previous run is processed. Incremental data load is typical in batch processing because it helps keep data processing fast and economical.

For more information, see [incremental data load](#).

Data contextualization

Data contextualization can be applied in various ways. In this architecture, contextualization is the process of performing a graph lookup and retrieving matching values.

The solution assumes that a graph has already been created in a graph database. The internal complexity of the graph isn't a concern because the graph query is passed via a configuration and executed dynamically with passed input values.

The solution uses Azure Databricks for the data contextualization process.

Graph database

The graph database is the database that stores the graph data and models. As noted earlier, there are many graph databases available. In this solution, SQL Database and the [graph capabilities of SQL Server](#) are used to create the graph.

Optional data mart

In this architecture, [SQL Database](#) is used to store the contextualized data as a data mart, but you can use any storage option. To ensure idempotent processing, the data is merged into the system rather than appended. In addition to Delta Lake in the gold layer, you might need to separate data marts due to integration system requirements, security, performance, cost, or other reasons.

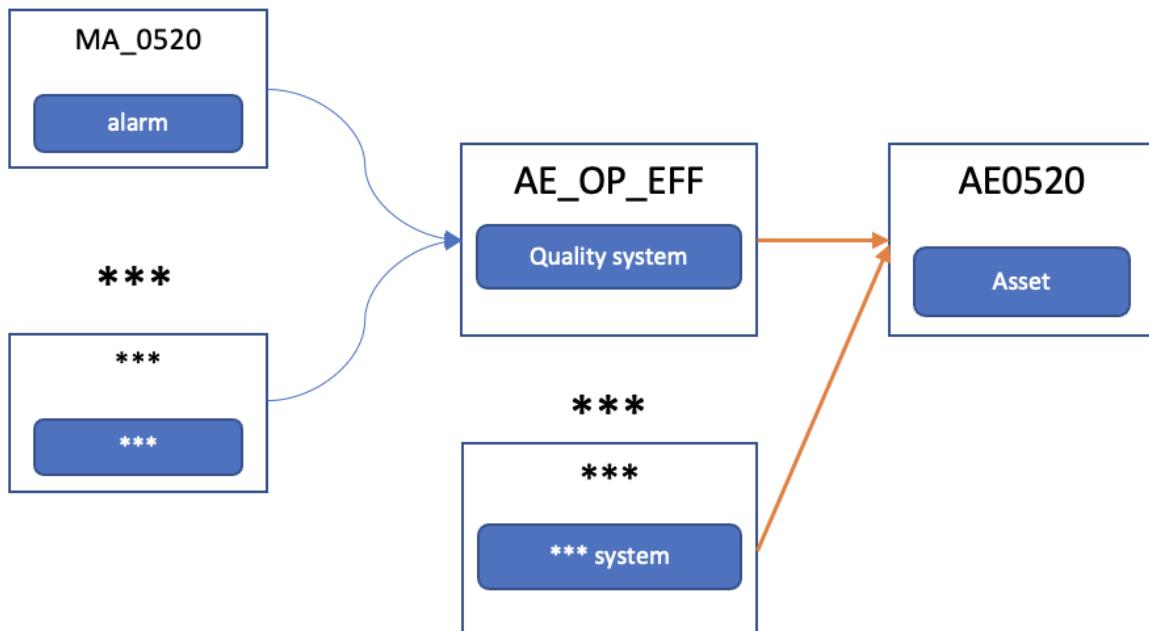
Contoso scenario

The solution in this article is based on the scenario that's described in this section.

Gary is an operations engineer at Contoso, Ltd. One of his responsibilities is to provide a weekly health report for the assets in Contoso factories within a specific city.

First, Gary needs to fetch all the asset IDs that he's interested in from the company's asset system. He then looks for all the attributes that belong to the assets to use as input for the health report. For example, the operational efficiency data of the asset with ID AE0520.

The following diagram illustrates some Contoso data relationships:



Contoso has many applications that help factory managers monitor processes and operations. Operational efficiency data is recorded in the quality system, another stand-alone application.

Gary signs in to the quality system and looks up the asset ID AE0520 in the `AE_OP_EFF` table. That table contains all the key attributes for operational efficiency data.

There are many columns in the `AE_OP_EFF` table. Gary is especially interested in the alarm status. However, the details for the most critical alarms of the asset are kept in another table called `Alarm`. Therefore, Gary needs to record that the key ID `MA_0520` of the `Alarm` table corresponds to the asset `AE0520`, because they use different naming conventions.

The relationship is actually much more complicated. Gary needs to search for more than one attribute of the asset and sign in to many tables in different systems to get all the data for a complete report. He uses queries and scripts to perform his work, but the queries are complicated and hard to maintain. Even worse, the systems are growing, and more data needs to be added to the report for different decision makers.

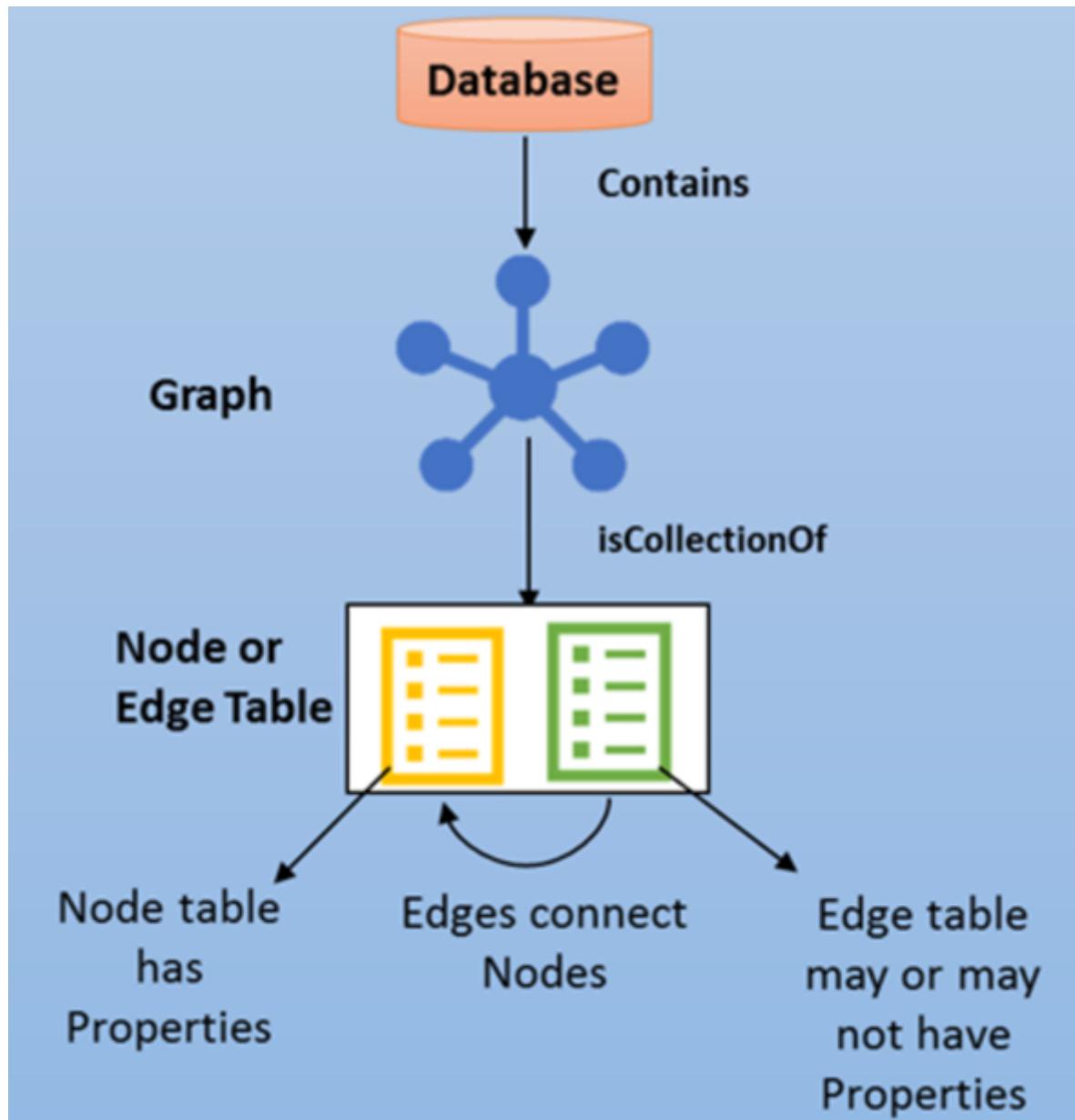
One of the main problems for Gary is that the IDs of a given asset in various systems are different. The systems were developed and are maintained separately, and they even use different protocols. Gary needs to manually query the various tables to get data for a single asset. The queries are complex and difficult to understand. As a result, Gary spends a lot of time training new operations engineers and explaining the relationships in the data.

Gary needs a mechanism to link the various names that belong to a single asset across systems. This mechanism will make report queries simpler and make Gary's job easier.

Graph design

SQL Database provides graph database capabilities for modeling many-to-many relationships. The graph relationships are integrated into Transact-SQL.

A graph database is a collection of nodes (or *vertices*) and edges (or *relationships*). A node represents an entity, like a person or an organization. An edge represents a relationship between the two nodes that it connects, for example, *likes* or *friends*.



Graph model for the scenario

This is the graph model for the Contoso scenario:

- `Alarm` is one of the metrics that belong to `Quality_System`.
- `Quality_System` is associated with an `Asset`.



This is what the data looks like:

Alarm		Asset		Quality_System	
ID	Alarm_Type	ID	Asset_ID	ID	Quality_ID
1	Fire Warning	1	AE0520	1	MA_0520_001
2	Flood Warning	2	AE0530	2	MA_0530_002
3	Carbon Monoxide Warning	3	AE0690	3	MA_0690_003

In the graph model, the nodes and edges need to be defined. Azure SQL graph uses edge tables to represent relationships. In this scenario, there are two edge tables. They record the relationships between `Alarm` and `Quality_System` and `Quality_System` and `Asset`.

The following table shows the nodes and edges:

[Expand table](#)

Nodes	Edges
Alarm	Alarm -> belongs_to -> Quality_System
Quality_System	Quality_System -> is_associated_with -> Asset
Asset	

To create these nodes and edges in SQL Database, you can use the following SQL commands:

SQL

```

...
CREATE TABLE Alarm (ID INTEGER PRIMARY KEY, Alarm_Type VARCHAR(100)) AS NODE;
CREATE TABLE Asset (ID INTEGER PRIMARY KEY, Asset_ID VARCHAR(100)) AS NODE;
CREATE TABLE Quality_System (ID INTEGER PRIMARY KEY, Quality_ID
VARCHAR(100)) AS NODE;
CREATE TABLE belongs_to AS EDGE;
CREATE TABLE is_associated_with AS EDGE;
...
  
```

These commands create the following graph tables:

- dbo.Alarm
- dbo.Asset
- dbo.belongs_to
- dbo.is_associated_with
- dbo.Quality_System

To query the graph database, you can use the [MATCH](#) clause to match patterns and traverse the graph:

SQL

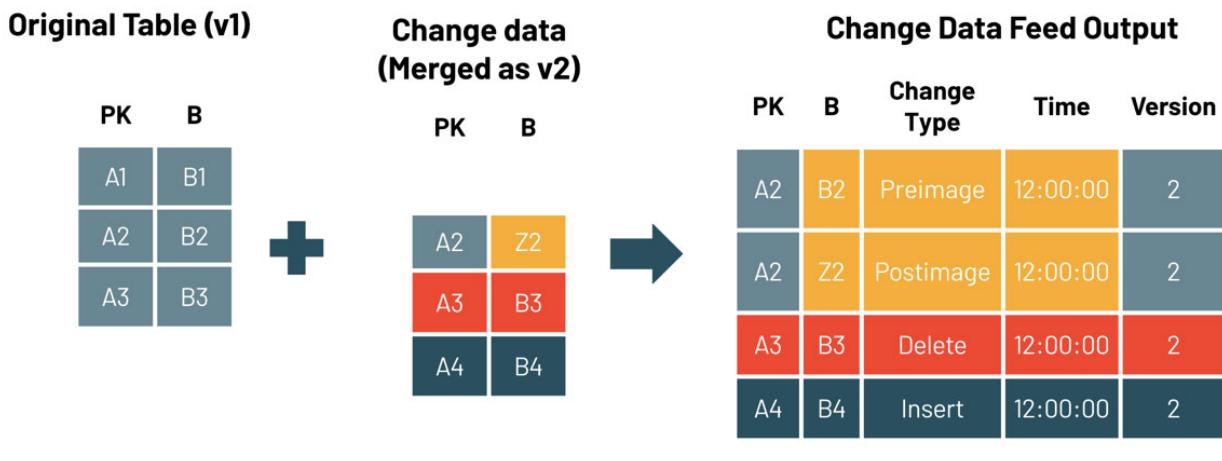
```
SELECT [dbo].[Alarm].Alarm_Type, [dbo].[Asset].Asset_ID
FROM [dbo].[Alarm], [dbo].[Asset], [dbo].[Quality_System], [dbo].
[belongs_to], [dbo].[is_associated_with]
WHERE MATCH (Alarm-(belongs_to)->Quality_System -(is_associated_with)->
Asset)
```

You can then use the query result to join the incoming raw data for contextualization.

Incremental data load

As the architecture diagram shows, the solution contextualizes only new incoming data, not the entire dataset in the Delta table. To meet this requirement, it uses an incremental data loading solution.

In Delta Lake, [change data feed](#) is a feature that simplifies the architecture for implementing change data capture. The following diagram illustrates how it works. When change data feed is enabled, the system records data changes, which, in this case, include inserted rows and two rows that represent the pre-image and post-image of an updated row. If you need to, you can use the pre-image and post-image information to evaluate the changes. There's also a delete change type that represents deleted rows. To query the change data, you can use the `table_changes` function.



In this solution, change data feed is enabled for Delta tables that store the source data. You can enable it by using this command:

SQL

```
CREATE TABLE tbl_alarm
  (alarm_id INT, alarm_type STRING, alarm_desc STRING, valid_from TIMESTAMP,
  valid_till TIMESTAMP)
  USING DELTA
  TBLPROPERTIES (delta.enableChangeDataFeed = true)
```

The following query gets the changed rows in the table. 2 is the commit version number.

SQL

```
SELECT *
FROM table_changes('tbl_alarm', 2)
```

If you only need information about newly inserted data, you can use this query:

SQL

```
SELECT *
FROM table_changes('tbl_alarm', 2)
WHERE _change_type = 'insert'
```

For more samples, see [Change data feed demo ↗](#).

You can use change data feed to load data incrementally. To do that, you need the version number of the most recent commit. You can create a Delta table to store that version number:

SQL

```
CREATE TABLE table_commit_version
  (table_name STRING, last_commit_version LONG)
  USING DELTA
```

Every time you load new data into `tbl_alarm`, you need to complete these steps:

1. Get the `last_commit_version` for the `tbl_alarm` table from `table_commit_version`.
2. Query and load the data added since the version that's stored in `last_commit_version`.
3. Get the highest commit version number of the `tbl_alarm` table.
4. Update `last_commit_version` in the `table_commit_version` table to prepare it for the next query.

Enabling change data feed doesn't have a significant effect on system performance or cost. The change data records are generated inline during the query execution process and are much smaller than the total size of the rewritten files.

Potential use cases

- A manufacturing solution provider wants to continuously contextualize the data and events that are provided by its customers. Because the context information is too complicated to represent in relational tables, the company uses graph models for data contextualization.
- A process engineer in a factory needs to troubleshoot a problem with factory equipment. The graph model stores all data, directly or indirectly related, from troubleshooting equipment to get information for root cause analysis.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is 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](#).

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](#).

For this scenario, you need to consider the security of data at rest (that is, data that's stored in Data Lake Storage, SQL Database, and Azure Databricks) and data that's in transit between the storage solutions.

For Data Lake Storage:

- Azure Storage service-side encryption (SSE) is enabled to help protect data at rest.
- Use a managed identity and role-based access control (RBAC) if data is accessed via supported Azure resources.
- If a managed identity is not supported, use shared access signature (SAS) to restrict access and permissions to data. Use HTTPS to protect data in transit.
- For more information, see [Azure security baseline for Storage](#).

For SQL Database:

- Use a managed identity and RBAC to limit access to specific operations and resources within a database.
- If you access SQL Database by using a sign-in, use a strong password. Save passwords in Azure Key Vault.
- Enable TLS to help secure in-transit data between SQL Database and Azure Databricks.
- See [Azure security baseline for Azure SQL](#) for more information.

For Azure Databricks:

- Use RBAC.
- Enable Azure Monitor to monitor your Azure Databricks workspace for unusual activity. Enable logging to track user activity and security events.
- To provide a layer of protection for data in transit, enable TLS for the JDBC connection to SQL Database.
- See [Azure security baseline for Azure Databricks](#) for more information.

In your production environment, put these resources into an [Azure virtual network](#) that isolates them from the public internet to reduce the attack surface and help protect against data exfiltration.

Cost optimization

Cost optimization is about reducing unnecessary expenses and improving operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

Cost optimization for SQL Database:

- Because solution performance isn't a goal for this architecture, it uses the lowest pricing tier that meets requirements.
- You should use the serverless compute tier, which is billed per second based on the number of compute cores that are used.

Cost optimization for Azure Databricks:

- Use the All-Purpose Compute workload and the Premium tier. Choose the instance type that meets your workload requirements while minimizing costs.
- Use autoscaling to scale the number of nodes based on workload demand.
- Turn off clusters when they aren't in use.

For more information about the cost of this scenario, see [this monthly cost estimate](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hong Bu](#) | Senior Program Manager
- [Chenshu Cai](#) | Software Engineer
- [Kenichiro Nakamura](#) | Senior Software Engineer
- [Anuj Parashar](#) | Senior Data Engineer
- [Bo Wang](#) | Software Engineer
- [Gary Wang](#) | Principal Software Engineer

Other contributor:

- [Mick Alberts](#) | Technical Writer

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [What is Azure Cosmos DB for Apache Gremlin?](#)
- [The leading graph data platform on Microsoft Azure](#)
- [Graph processing with SQL Server and SQL Database](#)
- [Use Delta Lake change data feed on Azure Databricks](#)
- [How to simplify CDC with Delta Lake's change data feed](#)
- [PostgreSQL graph search practices - 10-billion-scale graph with millisecond response](#)

- Azure security baseline for Azure Databricks

Related resources

- [Databases architecture design](#)

Query a data lake or lakehouse by using Azure Synapse serverless

Azure Data Lake

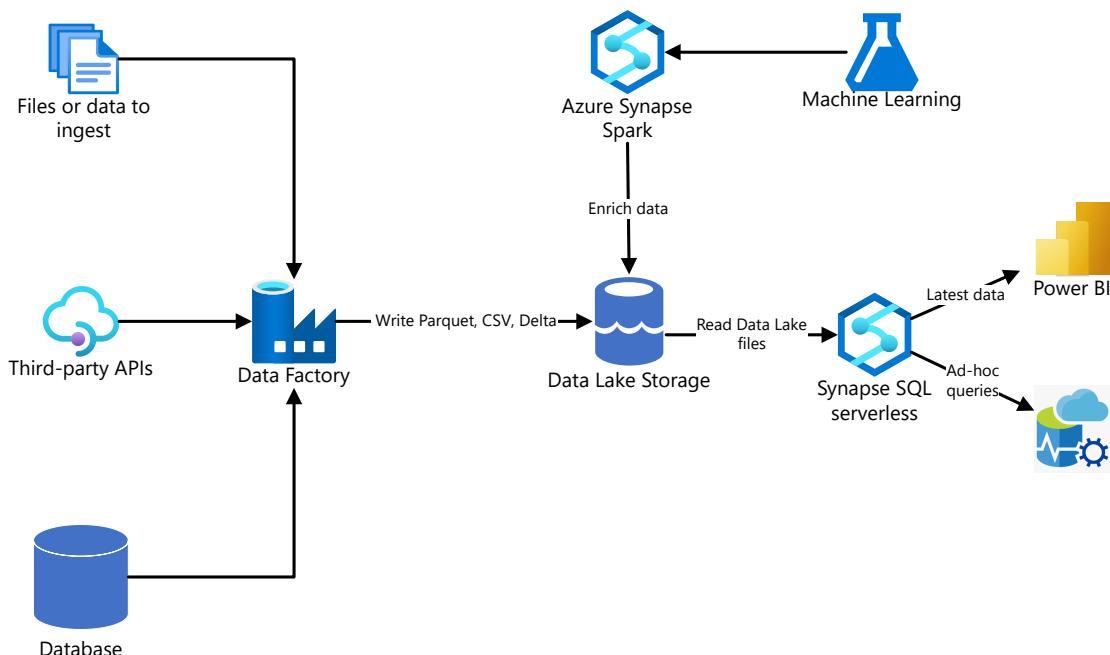
Azure Data Lake Storage

Azure Synapse Analytics

Azure Blob Storage

This article describes an alternative approach to data warehouse projects that's called *exploratory data analysis (EDA)*. This approach can reduce the challenges of extract, transform, load (ETL) operations. It focuses first on generating business insights and then turns to solving the modeling and ETL tasks.

Architecture



 Microsoft Azure



Download a [Visio file](#) of this architecture.

For EDA, you're concerned only with the right side of the diagram. Azure Synapse SQL serverless is used as the compute engine over the data lake files.

To accomplish EDA:

- T-SQL queries run directly in Azure Synapse SQL serverless or Azure Synapse Spark.
- Queries run from a graphical query tool like Power BI or Azure Data Studio.

We recommend that you persist all lakehouse data by using Parquet or Delta.

You can implement the left side of the diagram (data ingestion) by using any extract, load, transform (ELT) tool. It has no effect on EDA.

Components

- [Azure Synapse Analytics](#) combines data integration, enterprise data warehousing, and big data analytics over lakehouse data. In this solution:
 - An [Azure Synapse workspace](#) promotes collaboration among data engineers, data scientists, data analysts, and business intelligence (BI) professionals for EDA tasks.
 - [Azure Synapse serverless SQL pools](#) analyze unstructured and semi-structured data in Azure Data Lake Storage by using standard T-SQL.
 - [Azure Synapse serverless Apache Spark pools](#) do code-first explorations in Data Lake Storage by using Spark languages like Spark SQL, PySpark, and Scala.
- [Azure Data Lake Storage](#) provides storage for data that is then analyzed by Azure Synapse serverless SQL pools.
- [Azure Machine Learning](#) provides data to Azure Synapse Spark.
- [Power BI](#) is used in this solution to query data to accomplish EDA.

Alternatives

- You can replace or complement Synapse SQL serverless pools with [Azure Databricks](#).
- Instead of using a lakehouse model with Synapse SQL serverless pools, you can use [Azure Synapse dedicated SQL pools](#) to store enterprise data. Review the use cases and considerations in this article and related resources to decide which technology to use.

Scenario details

This solution shows an implementation of the EDA approach to data warehouse projects. This approach can reduce the challenges of ETL operations. It focuses first on generating business insights and then turns to solving modeling and ETL tasks.

Potential use cases

Other scenarios that can benefit from this analytics pattern:

- **Prescriptive analytics.** Ask questions of your data, like *Next Best Action*, or *what do we do next?* Use data to be more *data-driven* and less *gut-driven*. The data might be unstructured and from many external sources of varying quality. You might want to use the data as fast as possible to evaluate your business strategy without actually loading the data into a data warehouse. You might dispose of the data after you answer your questions.
- **Self-service ETL.** Do ETL/ELT when you do your data sandboxing (EDA) activities. Transform data and make it valuable. Doing so can improve the scale of your ETL developers.

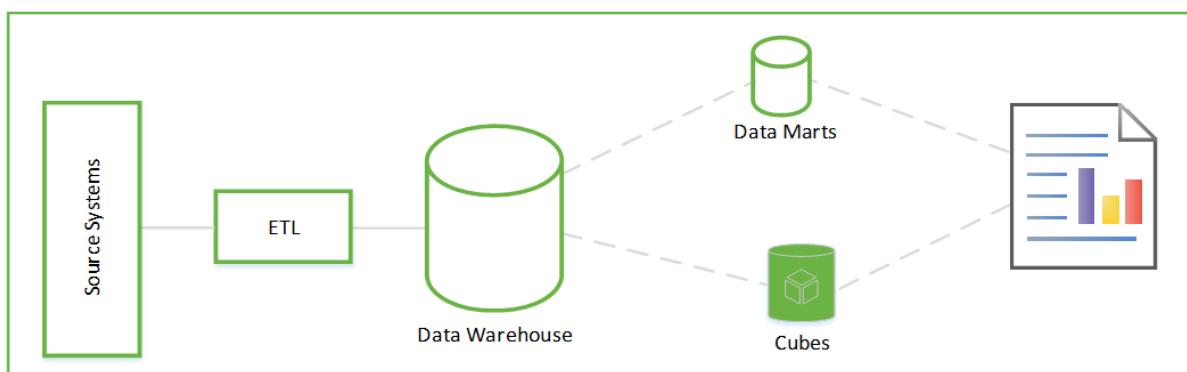
About exploratory data analysis

Before we look more closely at how EDA works, it's worth summarizing the traditional approach to data warehouse projects. The traditional approach looks like this:

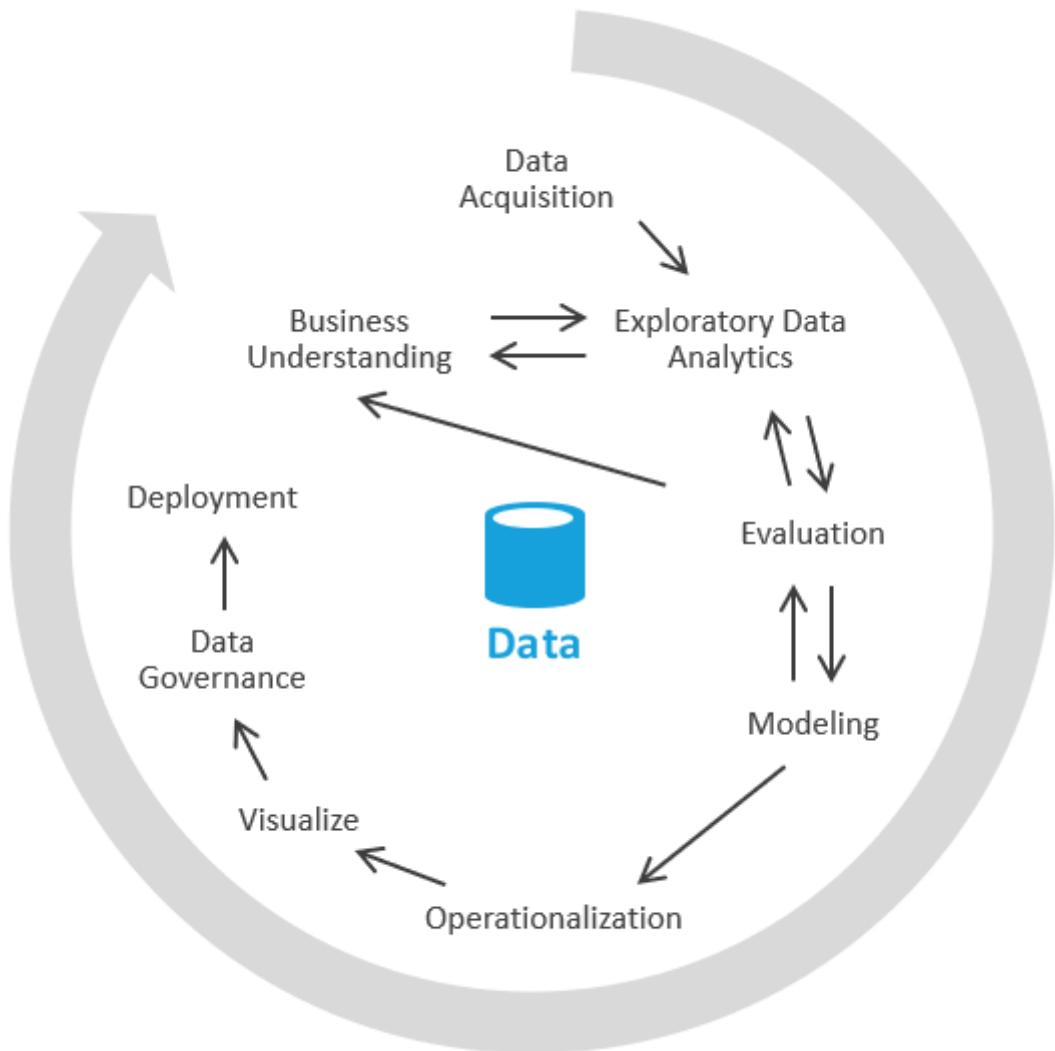
- **Requirements gathering.** Document what to do with the data.
- **Data modeling.** Determine how to model the numeric and attribute data into fact and dimension tables. Traditionally, you do this step before you acquire the new data.
- **ETL.** Acquire the data and massage it into the data warehouse's data model.

These steps can take weeks or even months. Only then can you begin to query the data and solve the business problem. The user sees value only after the reports are created. The solution architecture usually looks something like this one:

The Philosophy: Model data » Transform data » Load data » *Understand* data



You can do this in another way that focuses first on generating business insights and then turns to solving the modeling and ETL tasks. The process is similar to data science processes. It looks something like this:



In the industry, this process is called *EDA*, or *exploratory data analysis*.

Here are the steps:

- **Data acquisition.** First, you need to determine what data sources you need to ingest into your data lake / sandbox. You then need to bring that data into the landing area of your lake. Azure provides tools like Azure Data Factory and Azure Logic Apps that can ingest data quickly.
- **Data sandboxing.** Initially, a business analyst and an engineer who's skilled in exploratory data analysis via Azure Synapse Analytics serverless or basic SQL work together. During this phase, they're trying to uncover the *business insight* by using the new data. EDA is an iterative process. You might need to ingest more data, talk with SMEs, ask more questions, or generate visualizations.
- **Evaluation.** After you find the business insight, you need to evaluate what to do with the data. You might want to persist the data into the data warehouse (so you move to the modeling phase). In other cases, you might decide to keep the data in the data lake / lakehouse and use it for predictive analytics (machine learning

algorithms). In still other cases, you might decide to backfill your systems of record with the new insights. Based on these decisions, you can gain a better understanding of what you need to do next. You might not need to do ETL.

These methods are the core of true *self-service analytics*. By using the data lake and a query tool like Azure Synapse serverless that understands data lake query patterns, you can put your data assets into the hands of business people who understand a modicum of SQL. You can radically shorten the time-to-value by using this method and remove some of the risk associated with corporate data initiatives.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Availability

Azure Synapse SQL serverless pools is a platform as a service (PaaS) feature that can meet your high availability (HA) and disaster recovery (DR) requirements.

Serverless pools are available on-demand. They don't require scaling up, down, in, or out or administration of any kind. They use a pay-per-query model, so there's no unused capacity at any time. Serverless pools are ideal for:

- Ad-hoc data-science explorations in T-SQL.
- Early prototyping for data warehouse entities.
- Defining views that consumers can use, for example in Power BI, for scenarios that can tolerate performance lag.
- Exploratory data analysis.

Operations

Synapse SQL serverless uses standard T-SQL for querying and operations. You could use Synapse workspace UI, Azure Data Studio, or SQL Server Management Studio as the T-SQL tool.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost](#)

optimization pillar.

- [Data Lake Storage](#) pricing depends on the amount of data you store and how often you use the data. The sample pricing includes one TB of data stored, with further transactional assumptions. The one TB refers to the size of the data lake, not the size of the original legacy database.
- [Azure Synapse Spark pool](#) bases pricing on node size, number of instances, and uptime. The example assumes one small compute node with utilization between five hours per week and 40 hours per month.
- [Azure Synapse serverless SQL pool](#) bases pricing on TBs of data processed. The sample assumes 50 TBs processed per month. This figure refers to the size of the data lake, not the size of the original legacy database.

Contributors

This article is being updated and maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Dave Wentzel](#) | Principal MTC Technical Architect

Next steps

- [Data Engineer learning paths](#)
- [Tutorial: Get started with Azure Synapse Analytics](#)
- [Create a single database - Azure SQL Database](#)
- [Azure Synapse SQL architecture](#)
- [Create a storage account for Azure Data Lake Storage](#)
- [Azure Event Hubs Quickstart - Create an event hub by using the Azure portal](#)
- [Quickstart - Create a Stream Analytics job by using the Azure portal](#)
- [Quickstart: Get started with Azure Machine Learning](#)

Related resources

- Learn more about:
 - [Data lakes](#)
 - [Data warehousing and analytics](#)
 - [Analytics end-to-end with Azure Synapse](#)

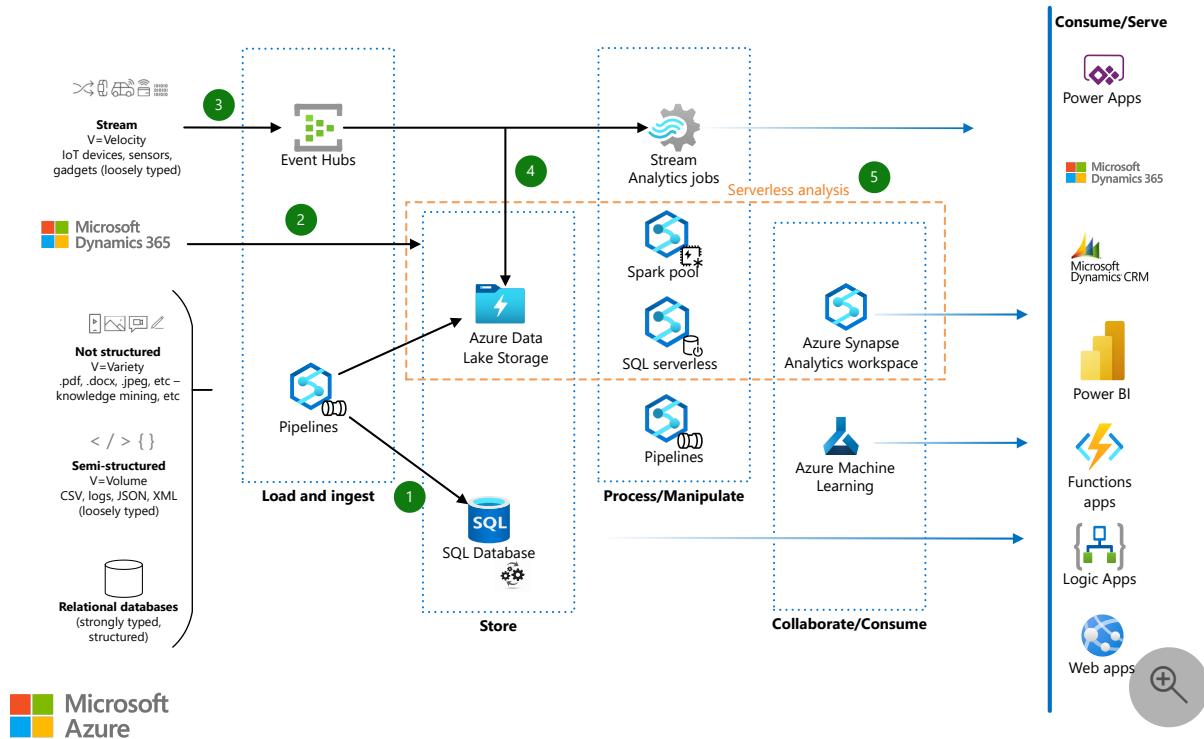
- Big data analytics with enterprise-grade security by using Azure Synapse
- Enterprise business intelligence

Modern data warehouse for small and medium business

Azure Data Lake Azure SQL Database Azure Synapse Analytics Dynamics 365 Microsoft Power Platform

This example workload shows several ways that small businesses (SMBs) can modernize legacy data stores and explore big data tools and capabilities, without overextending current budgets and skillsets. These end-to-end Azure data warehousing solutions integrate easily with tools like Azure Machine Learning, Microsoft Power Platform, Microsoft Dynamics, and other Microsoft technologies.

Architecture



Download a [Visio file](#) of this architecture.

Legacy SMB data warehouses might contain several types of data:

- Unstructured data, like documents and graphics
- Semi-structured data, such as logs, CSVs, JSON, and XML files
- Structured relational data, including databases that use stored procedures for extract-transform-load/extract-load-transform (ETL/ELT) activities

Dataflow

The following dataflow demonstrates the ingestion of your chosen data type:

1. Azure Synapse Analytics pipelines ingest the legacy data warehouses into Azure.
 - The pipelines orchestrate the flow of migrated or partially refactored legacy databases and SSIS packages into Azure SQL Database. This lift-and-shift approach is fastest to implement, and offers a smooth transition from an on-premises SQL solution to an eventual Azure platform-as-a-service (PaaS). You can modernize databases incrementally after the lift and shift.
 - The pipelines can also pass unstructured, semi-structured, and structured data into Azure Data Lake Storage for centralized storage and analysis with other sources. Use this approach when fusing data provides more business benefit than simply replatforming the data.
2. Microsoft Dynamics data sources can be used to build centralized BI dashboards on augmented datasets using Synapse Serverless analysis tools. You can bring the fused, processed data back into Dynamics and Power BI for further analysis.
3. Real-time data from streaming sources can also enter the system via Azure Event Hubs. For customers with real-time dashboard requirements, Azure Stream Analytics can analyze this data immediately.
4. The data can also enter the centralized Data Lake for further analysis, storage, and reporting.
5. Serverless analysis tools are available in the Azure Synapse Analytics workspace. These tools use serverless SQL pool or Apache Spark compute capabilities to process the data in Data Lake Storage Gen2. Serverless pools are available on demand, and don't require any provisioned resources.

Serverless pools are ideal for:

- Ad hoc data science explorations in T-SQL format.
- Early prototyping for data warehouse entities.
- Defining views that consumers can use, for example in Power BI, for scenarios that can tolerate performance lag.

Azure Synapse is tightly integrated with potential consumers of your fused datasets, like Azure Machine Learning. Other consumers can include Power Apps, Azure Logic Apps, Azure Functions apps, and Azure App Service web apps.

Components

- [Azure Synapse Analytics](#) is an analytics service that combines data integration, enterprise data warehousing, and big data analytics. In this solution:
 - An [Azure Synapse Workspace](#) promotes collaboration between data engineers, data scientists, data analysts, and business intelligence (BI) professionals.
 - [Azure Synapse pipelines](#) orchestrate and ingest data into SQL Database and Data Lake Storage Gen2.
 - [Azure Synapse serverless SQL pools](#) analyze unstructured and semi-structured data in Data Lake Storage Gen2 on demand.
 - [Azure Synapse serverless Apache Spark pools](#) do code-first explorations in Data Lake Storage Gen2 with Spark languages like Spark SQL, pySpark, and Scala.
- [Azure SQL Database](#) is an intelligent, scalable, relational database service built for the cloud. In this solution, SQL Database holds the enterprise data warehouse and performs ETL/ELT activities that use stored procedures.
- [Azure Event Hubs](#) is a real-time data streaming platform and event ingestion service. Event Hubs can ingest data from anywhere, and seamlessly integrates with Azure data services.
- [Azure Stream Analytics](#) is a real-time, serverless analytics service for streaming data. Stream Analytics offers rapid, elastic scalability, enterprise-grade reliability and recovery, and built-in machine learning capabilities.
- [Azure Machine Learning](#) is a toolset for data science model development and lifecycle management. Machine Learning is one example of the Azure and Microsoft services that can consume fused, processed data from Data Lake Storage Gen2.

Alternatives

- [Azure IoT Hub](#) could replace or complement Event Hubs. The solution you choose depends on the source of your streaming data, and whether you need cloning and bidirectional communication with the reporting devices.
- You can use [Azure Data Factory](#) for data integration instead of Azure Synapse pipelines. The choice depends on several factors:
 - Azure Synapse pipelines keep the solution design simpler, and allow collaboration inside a single Azure Synapse workspace.
 - Azure Synapse pipelines don't support SSIS packages rehosting, which is available in Azure Data Factory.

- [Synapse Monitor Hub](#) monitors Azure Synapse pipelines, while [Azure Monitor](#) can monitor Data Factory.

For more information and a feature comparison between Azure Synapse pipelines and Data Factory, see [Data integration in Azure Synapse Analytics versus Azure Data Factory](#).

- You can use [Synapse Analytics dedicated SQL pools](#) for storing enterprise data, instead of using SQL Database. Review the use cases and considerations in this article and related resources to make a decision.

Scenario details

Small and medium businesses (SMBs) face a choice when modernizing their on-premises data warehouses for the cloud. They can adopt big data tools for future extensibility, or keep traditional, SQL-based solutions for cost efficiency, ease of maintenance, and smooth transition.

However, a hybrid approach combines easy migration of the existing data estate with the opportunity to add big data tools and processes for some use cases. SQL-based data sources can keep running in the cloud and continue to modernize as appropriate.

This example workload shows several ways that SMBs can modernize legacy data stores and explore big data tools and capabilities, without overextending current budgets and skillsets. These end-to-end Azure data warehousing solutions integrate easily with Azure and Microsoft services and tools like Azure Machine Learning, Microsoft Power Platform, and Microsoft Dynamics.

Potential use cases

Several scenarios can benefit from this workload:

- Migrating a traditional, on-premises relational data warehouse that's smaller than 1 TB and extensively uses SQL Server Integration Services (SSIS) packages to orchestrate stored procedures.
- Meshing existing Dynamics or Power Platform [Dataverse](#) data with batched and real-time [Azure Data Lake](#) sources.
- Using innovative techniques to interact with centralized Data Lake Storage Gen2 data. Techniques include serverless analysis, knowledge mining, data fusion between domains, and end-user data exploration.

- Setting up eCommerce companies to adopt a data warehouse to optimize their operations.

This solution isn't recommended for:

- [Greenfield](#) deployment of data warehouses that are estimated to be > 1 TB within one year.
- Migrating on-premises data warehouses that are > 1 TB or projected to grow to that size within a year.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

The following considerations apply to this scenario.

Availability

SQL Database is a PaaS service that can meet your high availability (HA) and disaster recovery (DR) requirements. Be sure to pick the SKU that meets your requirements. For guidance, see [High availability for Azure SQL Database](#).

Operations

SQL Database uses [SQL Server Management Studio \(SSMS\)](#) to develop and maintain legacy artifacts like stored procedures.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

See a [pricing sample for a SMB data warehousing scenario](#) in the Azure pricing calculator. Adjust the values to see how your requirements affect the costs.

- [SQL Database](#) bases costs on the selected Compute and Service tiers, and the number of vCores and Database Transaction Units (DTUs). The example shows a

single database with provisioned Compute and eight vCores, based on the assumption that you need to run stored procedures in SQL Database.

- [Data Lake Storage Gen2](#) pricing depends on the amount of data you store and how often you use the data. The sample pricing includes 1 TB of data stored, with further transactional assumptions. The 1 TB refers to the size of the data lake, not the original legacy database size.
- [Azure Synapse pipelines](#) base costs on the number of data pipeline activities, integration runtime hours, data flow cluster size, and execution and operation charges. Pipeline costs increase with additional data sources and amounts of data processed. The example assumes one data source batched every hour for 15 minutes on an Azure-hosted integration runtime.
- [Azure Synapse Spark pool](#) bases pricing on node size, number of instances, and uptime. The example assumes one small compute node with five hours a week to 40 hours a month utilization.
- [Azure Synapse serverless SQL pool](#) bases pricing on TBs of data processed. The sample assumes 50 TBs processed a month. This figure refers to the size of the data lake, not the original legacy database size.
- [Event Hubs](#) bills based on tier, throughput units provisioned, and ingress traffic received. The example assumes one throughput unit in Standard tier over one million events for a month.
- [Stream Analytics](#) bases costs on the number of provisioned streaming units. The sample assumes one streaming unit used over the month.

Contributors

This article is being updated and maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- Galina Polyakova | Senior Cloud Solution Architect

Next steps

- For training content and labs, see the [Data Engineer Learning Paths](#).
- [Tutorial: Get started with Azure Synapse Analytics](#)
- [Create a single database - Azure SQL Database](#)

- [Create a storage account for Azure Data Lake Storage Gen2](#)
- [Azure Event Hubs Quickstart - Create an event hub using the Azure portal](#)
- [Quickstart - Create a Stream Analytics job by using the Azure portal](#)
- [Quickstart: Get started with Azure Machine Learning](#)

Related resources

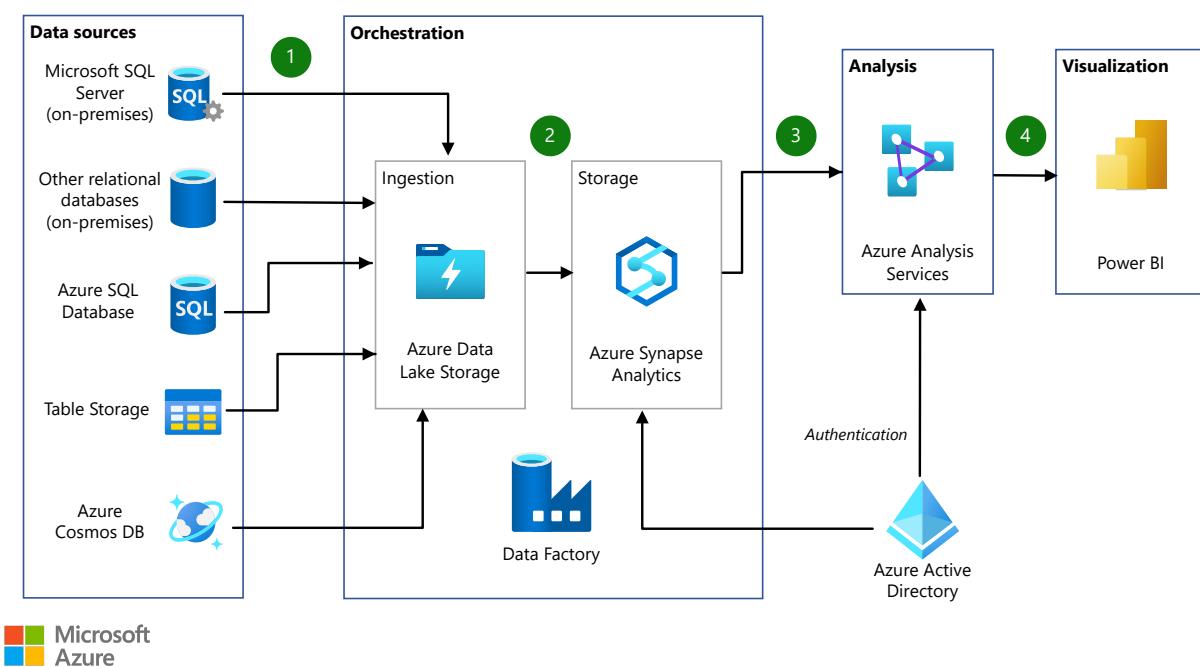
- Learn more about:
 - [Data lakes](#)
 - [Data warehousing and analytics](#)
 - [Analytics end-to-end with Azure Synapse](#)
 - [Big data analytics with enterprise-grade security using Azure Synapse](#)
 - [Enterprise business intelligence](#)

Data warehousing and analytics

Azure Data Lake Storage Azure Cosmos DB Azure Data Factory Azure SQL Database Azure Table Storage

This example scenario demonstrates a data pipeline that integrates large amounts of data from multiple sources into a unified analytics platform in Azure. This specific scenario is based on a sales and marketing solution, but the design patterns are relevant for many industries requiring advanced analytics of large datasets such as e-commerce, retail, and healthcare.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

The data flows through the solution as follows:

1. For each data source, any updates are exported periodically into a staging area in Azure Data Lake Storage.
2. Azure Data Factory incrementally loads the data from Azure Data Lake Storage into staging tables in Azure Synapse Analytics. The data is cleansed and transformed during this process. PolyBase can parallelize the process for large datasets.
3. After loading a new batch of data into the warehouse, a previously created Azure Analysis Services tabular model is refreshed. This semantic model simplifies the

analysis of business data and relationships.

4. Business analysts use Microsoft Power BI to analyze warehoused data via the Analysis Services semantic model.

Components

The company has data sources on many different platforms:

- SQL Server on-premises
- Oracle on-premises
- Azure SQL Database
- Azure table storage
- Azure Cosmos DB

Data is loaded from these different data sources using several Azure components:

- [Azure Data Lake Storage](#) is used to stage source data before it's loaded into Azure Synapse.
- [Data Factory](#) orchestrates the transformation of staged data into a common structure in Azure Synapse. Data Factory [uses PolyBase when loading data into Azure Synapse](#) to maximize throughput.
- [Azure Synapse](#) is a distributed system for storing and analyzing large datasets. Its use of massive parallel processing (MPP) makes it suitable for running high-performance analytics. Azure Synapse can use [PolyBase](#) to rapidly load data from Azure Data Lake Storage.
- [Analysis Services](#) provides a semantic model for your data. It can also increase system performance when analyzing your data.
- [Power BI](#) is a suite of business analytics tools to analyze data and share insights. Power BI can query a semantic model stored in Analysis Services, or it can query Azure Synapse directly.
- [Microsoft Entra ID](#) authenticates users who connect to the Analysis Services server through Power BI. Data Factory can also use Microsoft Entra ID to authenticate to Azure Synapse via a service principal or [Managed identity for Azure resources](#).

Alternatives

- The example pipeline includes several different kinds of data sources. This architecture can handle a wide variety of relational and non-relational data sources.

- Data Factory orchestrates the workflows for your data pipeline. If you want to load data only one time or on demand, you could use tools like SQL Server bulk copy (bcp) and AzCopy to copy data into Azure Data Lake Storage. You can then load the data directly into Azure Synapse using PolyBase.
- If you have very large datasets, consider using [Data Lake Storage](#), which provides limitless storage for analytics data.
- Azure Synapse is not a good fit for OLTP workloads or data sets smaller than 250 GB. For those cases you should use Azure SQL Database or SQL Server.
- For comparisons of other alternatives, see:
 - [Choosing a data pipeline orchestration technology in Azure](#)
 - [Choosing a batch processing technology in Azure](#)
 - [Choosing an analytical data store in Azure](#)
 - [Choosing a data analytics technology in Azure](#)

Scenario details

This example demonstrates a sales and marketing company that creates incentive programs. These programs reward customers, suppliers, salespeople, and employees. Data is fundamental to these programs, and the company wants to improve the insights gained through data analytics using Azure.

The company needs a modern approach to analysis data, so that decisions are made using the right data at the right time. The company's goals include:

- Combining different kinds of data sources into a cloud-scale platform.
- Transforming source data into a common taxonomy and structure, to make the data consistent and easily compared.
- Loading data using a highly parallelized approach that can support thousands of incentive programs, without the high costs of deploying and maintaining on-premises infrastructure.
- Greatly reducing the time needed to gather and transform data, so you can focus on analyzing the data.

Potential use cases

This approach can also be used to:

- Establish a data warehouse to be a single source of truth for your data.
- Integrate relational data sources with other unstructured datasets.

- Use semantic modeling and powerful visualization tools for simpler data analysis.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

The technologies in this architecture were chosen because they met the company's requirements for scalability and availability, while helping them control costs.

- The [massively parallel processing architecture](#) of Azure Synapse provides scalability and high performance.
- Azure Synapse has [guaranteed SLAs](#) and [recommended practices for achieving high availability](#).
- When analysis activity is low, the company can [scale Azure Synapse on demand](#), reducing or even pausing compute to lower costs.
- Azure Analysis Services can be [scaled out](#) to reduce response times during high query workloads. You can also separate processing from the query pool, so that client queries aren't slowed down by processing operations.
- Azure Analysis Services also has [guaranteed SLAs](#) and [recommended practices for achieving high availability](#).
- The [Azure Synapse security model](#) provides connection security, [authentication and authorization](#) via Microsoft Entra ID or SQL Server authentication, and encryption. [Azure Analysis Services](#) uses Microsoft Entra ID for identity management and user authentication.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

Review a [pricing sample for a data warehousing scenario](#) via the Azure pricing calculator. Adjust the values to see how your requirements affect your costs.

- [Azure Synapse](#) allows you to scale your compute and storage levels independently. Compute resources are charged per hour, and you can scale or pause these resources on demand. Storage resources are billed per terabyte, so your costs will increase as you ingest more data.
- [Data Factory](#) costs are based on the number of read/write operations, monitoring operations, and orchestration activities performed in a workload. Your

Data Factory costs will increase with each additional data stream and the amount of data processed by each one.

- [Analysis Services](#) is available in developer, basic, and standard tiers. Instances are priced based on query processing units (QPs) and available memory. To keep your costs lower, minimize the number of queries you run, how much data they process, and how often they run.
- [Power BI](#) has different product options for different requirements. [Power BI Embedded](#) provides an Azure-based option for embedding Power BI functionality inside your applications. A Power BI Embedded instance is included in the pricing sample above.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributor.

Principal author:

- [Alex Buck](#) | Senior Content Developer

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- Review the [Azure reference architecture for automated enterprise BI](#), which includes instructions for deploying an instance of this architecture in Azure.
- Learn more about the services used in this scenario:
 - [Introduction to Azure Data Lake Storage Gen2](#)
 - [Azure Data Factory documentation](#)
 - [What is dedicated SQL pool in Azure Synapse Analytics?](#)
 - [Azure Analysis Services documentation](#)
 - [Power BI documentation](#)
 - [Microsoft Entra documentation](#)

Related resources

- [Enterprise data warehouse](#)

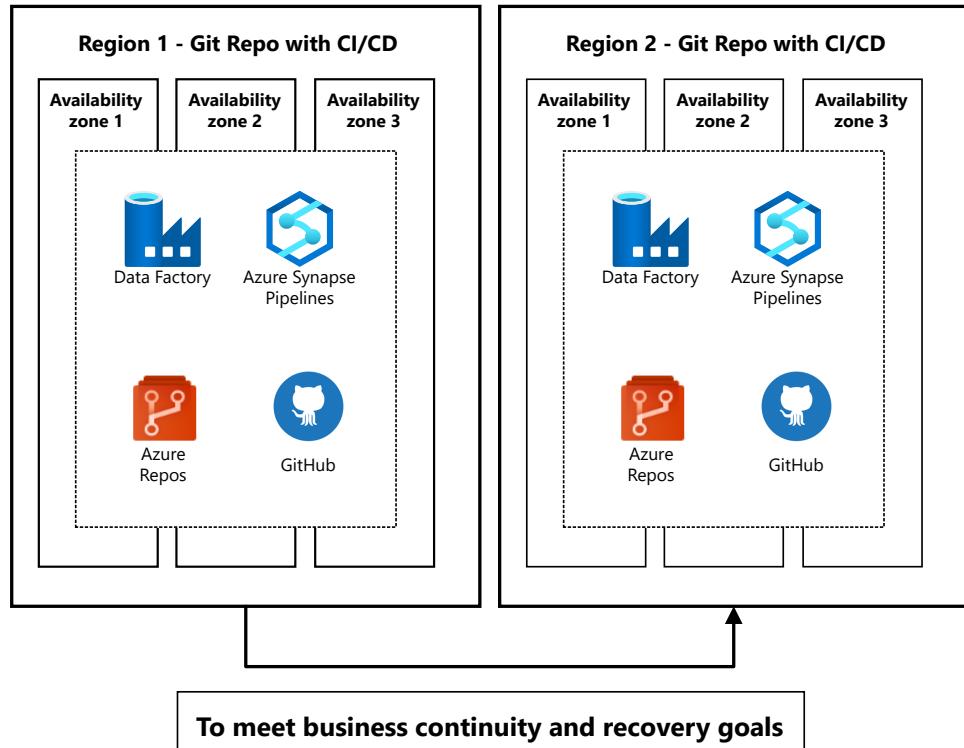
BCDR for Azure Data Factory and Azure Synapse Analytics pipelines

Azure Data Factory Azure Repos Azure Synapse Analytics GitHub

Disasters can be hardware failures, natural disasters, or software failures. The process of preparing for and recovering from a disaster is called disaster recovery (DR). This article discusses recommended practices to achieve business continuity and disaster recovery (BCDR) for Azure Data Factory and Azure Synapse Analytics pipelines.

BCDR strategies include availability zone redundancy, automated recovery provided by Azure disaster recovery, and user-managed recovery by using continuous integration/continuous delivery (CI/CD).

Architecture



 Microsoft Azure



Download a [Visio file](#) of this architecture.

Workflow

1. Data Factory and Azure Synapse pipelines achieve resiliency by using Azure regions and Azure availability zones.

- Each Azure region has a set of datacenters that are deployed within a latency-defined perimeter.
- Azure availability zones are physically separate locations within each Azure region that are tolerant to local failures.
- All Azure regions and availability zones are connected through a dedicated, regional low-latency network and by a high-performance network.
- All availability zone-enabled regions have at least three separate availability zones to ensure resiliency.

2. When a datacenter, part of a datacenter, or an availability zone in a region goes down, failover happens with zero downtime for zone-resilient Data Factory and Azure Synapse pipelines.

Components

- [Azure Data Factory](#)
- [Azure Synapse Analytics](#) and [Azure Synapse pipelines](#)
- [GitHub](#)
- [Azure Repos](#)

Scenario details

Data Factory and Azure Synapse pipelines store artifacts that include the following data:

Metadata

- Pipeline
- Datasets
- Linked services
- Integration runtime
- Triggers

Monitoring data

- Pipeline
- Triggers
- Activity runs

Disasters can strike in different ways, such as hardware failures, natural disasters, or software failures that result from human error or cyberattack. Depending on the types of

failures, their geographical impact can be regional or global. When planning a disaster recovery strategy, consider both the nature of the disaster and its geographic impact.

BCDR in Azure works on a shared responsibility model. Many Azure services require customers to explicitly set up their DR strategy, while Azure provides the baseline infrastructure and platform services as needed.

You can use the following recommended practices to achieve BCDR for Data Factory and Azure Synapse pipelines under various failure scenarios. For implementation, see [Deploy this scenario](#).

Automated recovery with Azure disaster recovery

With automated recovery provided Azure backup and disaster recovery, when there is a complete regional outage for an Azure region that has a paired region, Data Factory or Azure Synapse pipelines automatically fail over to the paired region when you [Set up automated recovery](#). The exceptions are Southeast Asia and Brazil regions, where data residency requirements require data to stay in those regions.

In DR failover, Data Factory recovers the production pipelines. If you need to validate your recovered pipelines, you can back up the Azure Resource Manager (ARM) templates for your production pipelines in secret storage, and compare the recovered pipelines to the backups.

The Azure Global team conducts regular BCDR drills, and Azure Data Factory and Azure Synapse Analytics participate in these drills. The BCDR drill simulates a region failure and fails over Azure services to a paired region without any customer involvement. For more information about the BCDR drills, see [Testing of services](#).

User-managed redundancy with CI/CD

To achieve BCDR in the event of an entire region failure, you need a data factory or an Azure Synapse workspace in the secondary region. In case of accidental Data Factory or Azure Synapse pipeline deletion, outages, or internal maintenance events, you can use Git and CI/CD to recover the pipelines manually.

Optionally, you can use an active/passive implementation. The primary region handles normal operations and remains active, while the secondary DR region requires pre-planned steps, depending on specific implementation, to be promoted to primary. In this case, all the necessary configurations for infrastructure are available in the secondary region, but they aren't provisioned.

Potential use cases

User-managed redundancy is useful in scenarios like:

- Accidental deletion of pipeline artifacts through human error.
- Extended outages or maintenance events that don't trigger BCDR because there's no disaster reported.

You can quickly move your production workloads to other regions and not be affected.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Reliability

Reliability ensures your application can meet the commitments you make to your customers. For more information, see [Overview of the reliability pillar](#).

Data Factory and Azure Synapse pipelines are mainstream Azure services that support availability zones, and they're designed to provide the right level of resiliency and flexibility along with ultra-low latency.

The user-managed recovery approach allows you to continue operating if there are any maintenance events, outages, or human errors in the primary region. By using CI/CD, the Data Factory and Azure Synapse pipelines can integrate to a Git repository and deploy to a secondary region for immediate recovery.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

User-managed recovery integrates Data Factory with Git by using CI/CD, and optionally uses a secondary DR region that has all the necessary infrastructure configurations as a backup. This scenario might incur added costs. To estimate costs, use the [Azure pricing calculator](#).

For examples of Data Factory and Azure Synapse Analytics pricing, see:

- Understanding Azure Data Factory pricing through examples
- Azure Synapse Analytics pricing ↗

Operational excellence

Operational excellence covers the operations processes that deploy an application and keep it running in production. For more information, see [Overview of the operational excellence pillar](#).

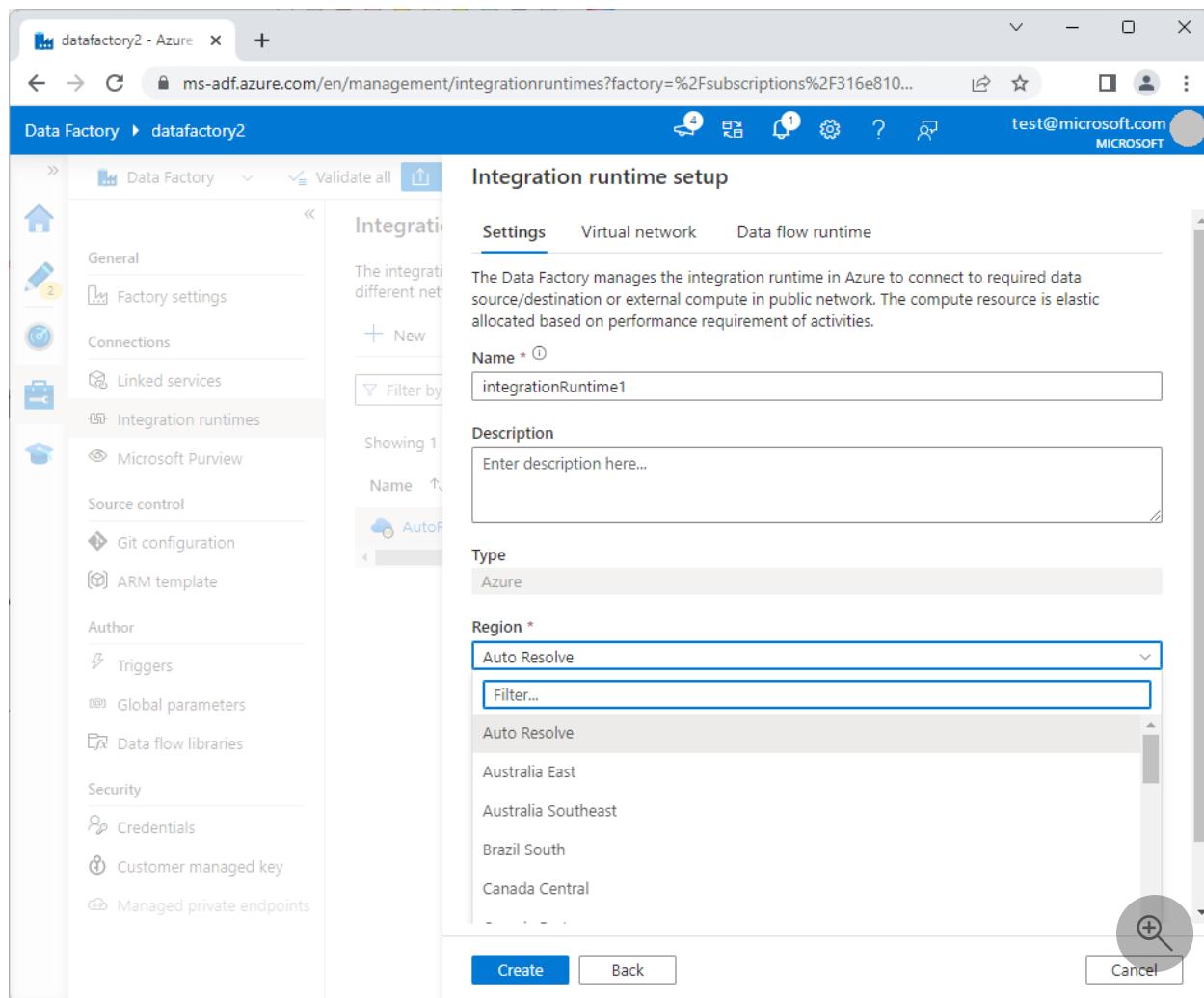
By using the user-managed CI/CD recovery approach, you can integrate to Azure Repos or GitHub. For more information about best CI/CD practices, see [Best practices for CI/CD](#).

Deploy this scenario

Take the following actions to set up automated or user-managed DR for Data Factory and Azure Synapse pipelines.

Set up automated recovery

In Data Factory, you can set the Azure integration runtime (IR) region for your activity execution or dispatch in the **Integration runtime setup**. To enable automatic failover in the event of a complete regional outage, set the **Region** to **Auto Resolve**.



In the context of the integration runtimes, IR fails over automatically to the paired region when you select **Auto Resolve** as the IR region. For other specific location regions, you can create a secondary data factory in another region, and use CI/CD to provision your data factory from the Git repository.

- For managed virtual networks, Data Factory also automatically switches over to the managed IR.
- Azure managed automatic failover doesn't apply to self-hosted integration runtime (SHIR), because the infrastructure is customer-managed. For guidance on setting up multiple nodes for higher availability with SHIR, see [Create and configure a self-hosted integration runtime](#).
- To configure BCDR for Azure-SSIS IR, see [Configure Azure-SSIS integration runtime for business continuity and disaster recovery \(BCDR\)](#).

Linked services aren't fully enabled after failover, because of pending private endpoints in the newer network of the region. You need to configure private endpoints in the recovered region. You can automate private endpoint creation by using the [approval API](#).

Set up user-managed recovery through CI/CD

You can use Git and CI/CD to recover pipelines manually in case of Data Factory or Azure Synapse pipeline deletion or outage.

- To use Data Factory pipeline CI/CD, see [Continuous integration and delivery in Azure Data Factory](#) and [Source control in Azure Data Factory](#).
- To use Azure Synapse pipeline CI/CD, see [Continuous integration and delivery for an Azure Synapse Analytics workspace](#). Make sure to initialize the Azure Synapse workspace first. For more information, see [Source control in Synapse Studio](#).

When you deploy user-managed redundancy by using CI/CD, take the following actions:

Disable triggers

Disable triggers in the original primary data factory once it comes back online. You can disable the triggers manually, or implement automation to periodically check the availability of the original primary. Disable all triggers on the original primary data factory immediately after the factory recovers.

To use Azure PowerShell to turn Data Factory triggers off or on, see [Sample pre- and post-deployment script](#) and [CI/CD improvements related to pipeline triggers deployment](#).

Handle duplicate writes

Most extract, transform, load (ETL) pipelines are designed to handle duplicate writes, because backfill and restatement require them. Data sinks that support transparent failover can handle duplicate writes with records merge or by deleting and inserting all records in the specific time range.

For data sinks that change endpoints after failover, primary and secondary storage might have duplicate or partial data. You need to merge the data manually.

Check the witness and control the pipeline flow (optional)

In general, you need to design your pipelines to include activities, like fail and lookup activities, for restarting failed pipelines from the point of interest.

1. Add a global parameter in your data factory to indicate the region, for example `region='EastUS'` in the primary and `region='CentralUS'` in the secondary data factory.

2. Create a witness in a third region. The witness can be a REST call or any type of storage. The witness returns the current primary region, for example '`EastUS`', by default.
3. When a disaster happens, manually update the witness to return the new primary region, for example '`CentralUS`'.
4. Add an activity in your pipeline to look up the witness and compare the current primary value to the global parameter.
 - If the parameters match, this pipeline is running on the primary region. Proceed with the real work.
 - If the parameters don't match, this pipeline is running on the secondary region. Just return the result.

 **Note**

This approach introduces a dependency on the witness lookup into your pipeline. Failure to read the witness halts all pipeline runs.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Krishnakumar Rukmangathan](#) | Senior Program Manager - Azure Data Factory team
- [Sunil Sabat](#) | Principal Program Manager - Azure Data Factory team

Other contributors:

- [Mario Zimmermann](#) | Principal Software Engineering Manager - Azure Data Factory team
- [Wee Hyong Tok](#) | Principal Director of PM - Azure Data Factory team

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- Business continuity management in Azure
- Resiliency in Azure
- Azure resiliency terminology
- Regions and availability zones
- Cross-region replication in Azure
- Azure regions decision guide
- Azure services that support availability zones
- Shared responsibility in the cloud
- Azure Data Factory data redundancy
- Integration runtime in Azure Data Factory
- Pipelines and activities in Azure Data Factory and Azure Synapse Analytics
- Data integration in Azure Synapse Analytics versus Azure Data Factory

Related resources

- Enterprise-scale disaster recovery
- SMB disaster recovery with Azure Site Recovery
- Build high availability into your BCDR strategy
- High availability and disaster recovery scenarios for IaaS apps
- Choose a data pipeline orchestration technology in Azure
- Business continuity and disaster recovery for Azure Logic Apps

Employee retention with Databricks and Kubernetes

Azure Databricks

Azure Kubernetes Service (AKS)

Azure Container Registry

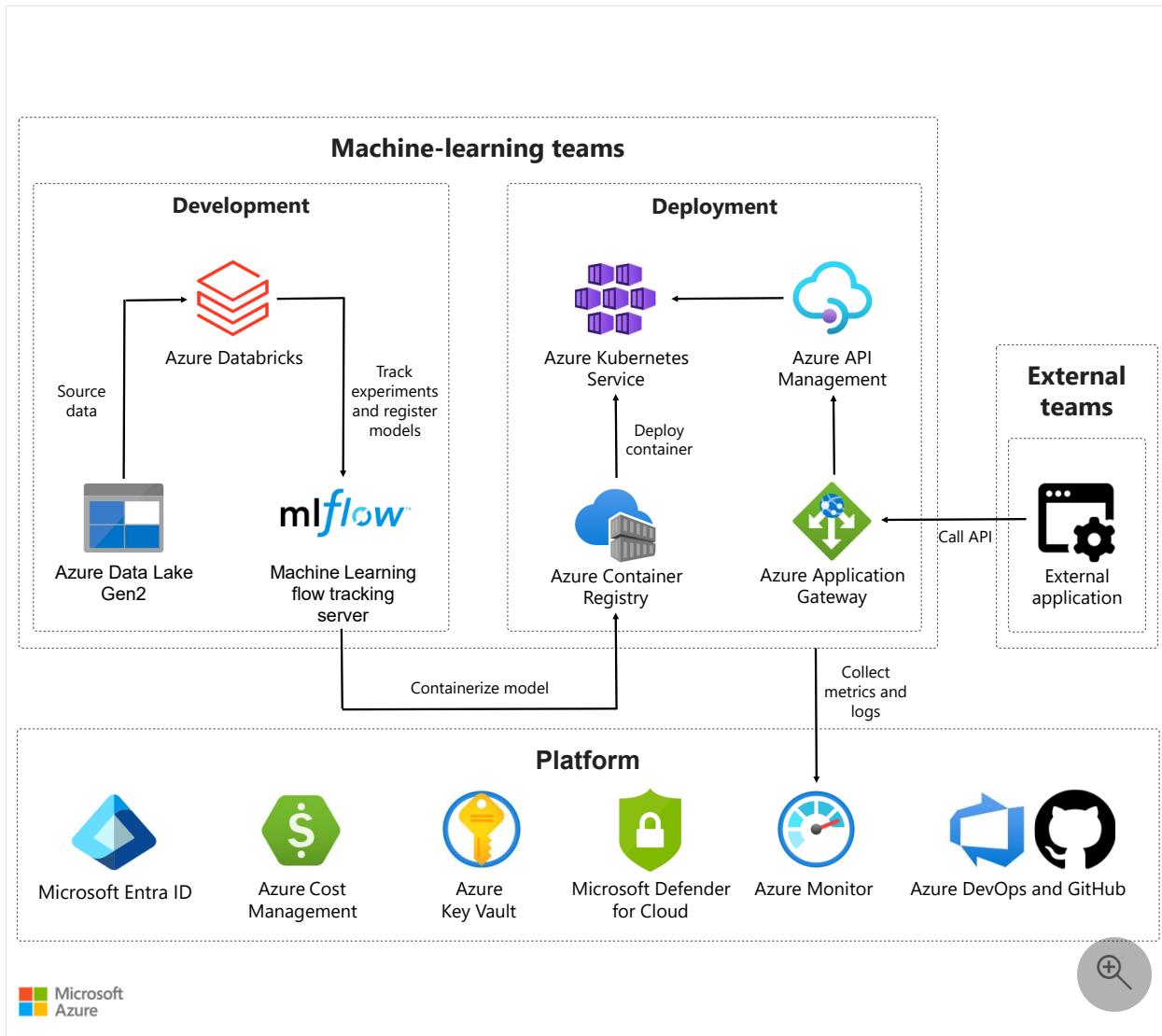
Azure Storage

Azure Monitor

This solution demonstrates how a machine-learning team can use Azure Databricks and Azure Kubernetes Service to develop and deploy machine learning, as an API, to predict the likelihood of employee attrition. The API can be integrated with external applications that are used by the Human Resources team to provide additional insights into the likelihood of attrition for a given employee within the organization. This information can be used to retain high-impact employees who are likely to leave the organization by providing Human Resources with the ability to proactively incentivize such employees to stay.

Apache®, Apache Ignite, Ignite, 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 [PowerPoint file](#) for all architectures.

Workflow

At a high level, this solution design addresses each stage of the machine-learning lifecycle:

- *Data preparation*, which includes sourcing, cleaning, and transforming the data for processing and analysis. Data can live in a data lake or data warehouse and be stored in a feature store after it's curated.
- *Model development*, which includes core components of the process of model development, such as experiment tracking and model registration by using [MLflow](#).
- *Model deployment*, which includes implementing a continuous integration/continuous delivery (CI/CD) pipeline to containerize machine-learning models as API services. These services are deployed to Azure Kubernetes clusters for end users to consume.

- *Model monitoring*, which includes monitoring the API performance and model data drift by analyzing log telemetry with Azure Monitor.

After the machine-learning team has deployed the machine-learning model as an API for real-time inference, developers can easily integrate the API with external applications that are used by external teams, such as Human Resources. Telemetry is collected when an external team uses the model service. The machine-learning team can use this telemetry to determine when the model needs to be redeployed. This approach allows teams to work independently and allows external teams to benefit from the skills of the centralized machine-learning team.

ⓘ Note

- You can use various tools, such as Azure DevOps Pipelines and GitHub Actions, when implementing a [CI/CD pipeline](#).
- The specific business requirements of your use case for analytics could require different services or features that aren't considered in this design.

Components

The following components are used as part of this design:

- [Azure Databricks](#): An analytics service for big data that's easy to use, facilitates collaboration, and is based on Apache Spark. Azure Databricks is designed for data science and data engineering.
- [Azure Kubernetes Service](#): A service that provides simplified deployment and management of Kubernetes by offloading the operational overhead to Azure.
- [Azure Container Registry](#): A private registry service for managing container images and artifacts. This service is based on the open-source Docker.
- [Azure Data Lake](#): A service that provides scalable storage that's optimized for massive amounts of unstructured data. [Data Lake Storage Gen2](#) offers file system semantics, file-level security, and scale.
- [Azure Monitor](#): A comprehensive solution for collecting, analyzing, and acting on telemetry from your workloads.
- [MLflow](#): An open-source solution that's integrated within Databricks for managing the machine-learning lifecycle from end to end.

- [Azure API Management](#) : A fully managed service that helps customers to publish, secure, transform, maintain, and monitor APIs.
- [Azure Application Gateway](#) : A load balancer for web traffic that enables you to manage traffic to your web applications.
- [Azure DevOps](#) or [GitHub](#) : Solutions for implementing DevOps practices to enforce automation and compliance with your workload development and deployment pipelines.

Scenario details

The problem of employee attrition has grown in prominence since the COVID-19 pandemic. This trend, in which employees voluntarily resign from their jobs en masse, is popularly known as *the Great Resignation*. The problem can also be magnified for certain departments in an organization that might lack dedicated teams that perform advanced analytics, such as Human Resources.

This example scenario illustrates an operating model of centralized machine learning. This comprises a central team that's responsible for building and deploying machine-learning models for external teams across departments within an organization. This approach is useful when departments are too small to maintain a team that's dedicated to machine learning while the organization aims to infuse advanced analytics into all products and processes.

Potential use cases

This scenario is focused on building a machine-learning model of employee attrition and integrating it with external applications that are used by Human Resources teams. However, the design can be generalized to many machine-learning workloads that are built by centralized and decentralized teams alike.

This generalized approach is best suited for:

- Machine-learning teams that have standardized on Databricks for data engineering or machine-learning applications.
- Machine-learning teams that have experience deploying and managing Kubernetes workloads and a preference for applying these skills for operationalizing machine-learning workloads.
- Integrating machine-learning workloads with external applications that require low latency and interactive model predictions (for example, real-time inference).

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Before you implement this solution, some factors you might want to consider include:

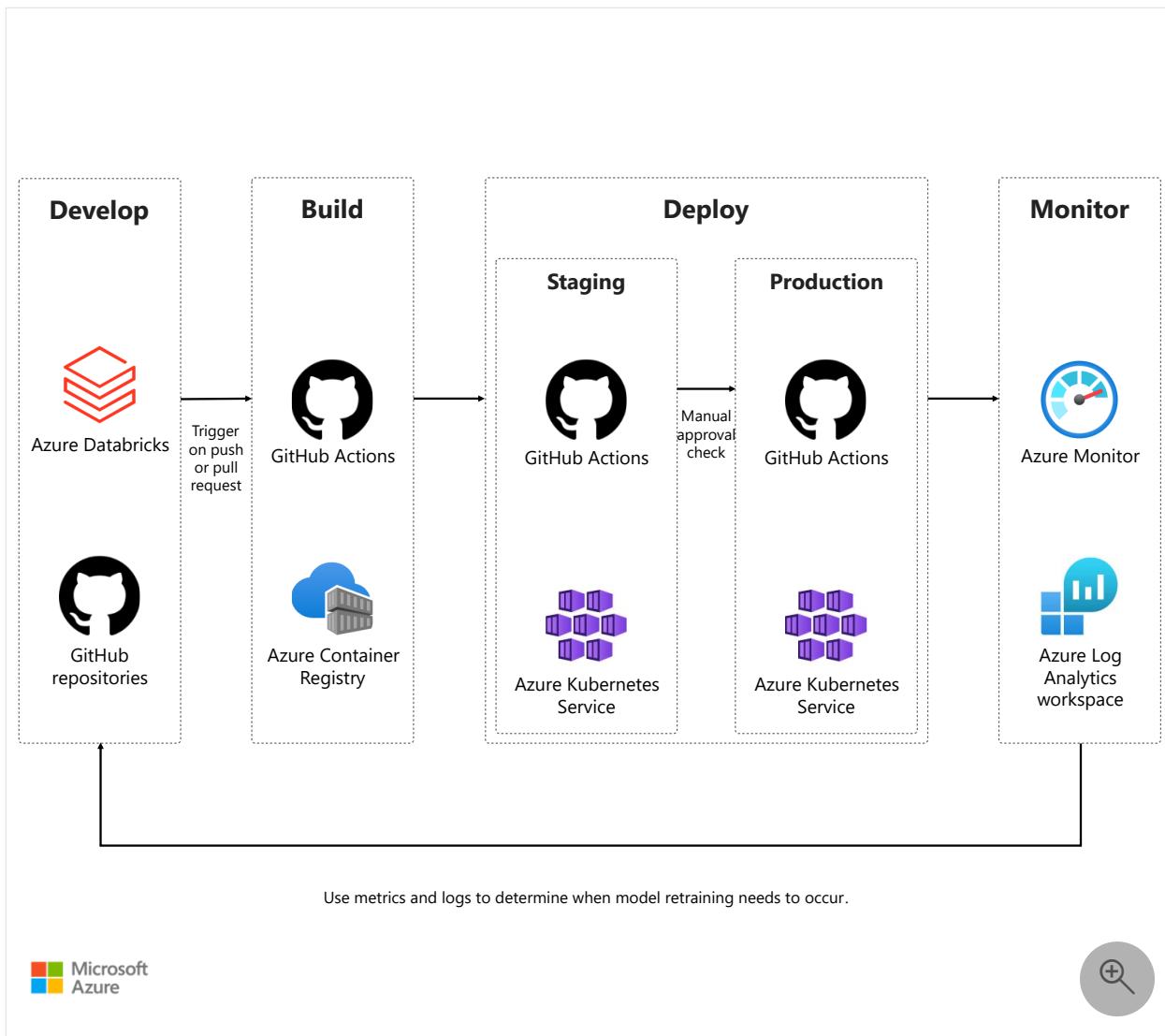
- This solution is designed for teams that require a high degree of customization and have extensive expertise in deploying and managing Kubernetes workloads. If your data science team doesn't have this expertise, consider deploying models to another service, such as [Azure Machine Learning](#).
- [Machine learning DevOps \(MLOps\) best practices with Azure Machine Learning](#) presents best practices and recommendations for adopting ML operations (MLOps) in the enterprise with machine learning.
- Follow the recommendations and guidelines defined in the [Azure Well-Architected Framework](#) to improve the quality of your Azure solutions.
- When implementing a CI/CD pipeline, you can use different tools than this example uses, such as Azure Pipelines and GitHub Actions. For more information about CI/CD, see [CI/CD for microservices architectures](#).
- Specific business requirements for your analytics use case could require the use of services or features that aren't considered in this design.

Cost optimization

All services deployed in this solution use a consumption-based pricing model. You can use the [Azure pricing calculator](#) to estimate costs for a specific scenario. For other considerations, see [Cost optimization](#) in the Well-Architected Framework.

Deploy this scenario

A proof-of-concept implementation of this scenario is available on GitHub at [Employee Retention with Databricks and Kubernetes](#).



Download a [PowerPoint file](#) for all architecture.

This proof-of-concept illustrates:

- How to train an MLflow model for employee attrition on Azure Databricks.
- How to package models as a web service by using open-source tools.
- How to deploy to Kubernetes via CI/CD by using GitHub Actions.
- How to monitor API performance and model data drift within Azure Monitor and Azure Log Analytics workspaces.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Nicholas Moore](#) | Cloud Solution Architect

Next steps

Product documentation:

- [What is Azure Databricks?](#)
- [MLflow guide](#)
- [Azure Kubernetes Service](#)
- [Introduction to private Docker container registries in Azure](#)
- [About API management](#)
- [What is Azure Application Gateway?](#)
- [Introduction to Azure Data Lake Storage Gen2](#)
- [Azure Monitor overview](#)
- [Azure DevOps documentation](#)
- [Azure and GitHub integration](#)

Microsoft Learn modules:

- [Perform data science with Azure Databricks](#)
- [Build and operate machine learning solutions with Azure Databricks](#)
- [Introduction to Kubernetes on Azure](#)
- [Develop and deploy applications on Kubernetes](#)
- [Automate your workflow with GitHub Actions](#)

Related resources

You might also find these Architecture Center articles useful:

- [Machine Learning operations maturity model](#)
- [Team Data Science Process for data scientists](#)
- [Modern analytics architecture with Azure Databricks](#)

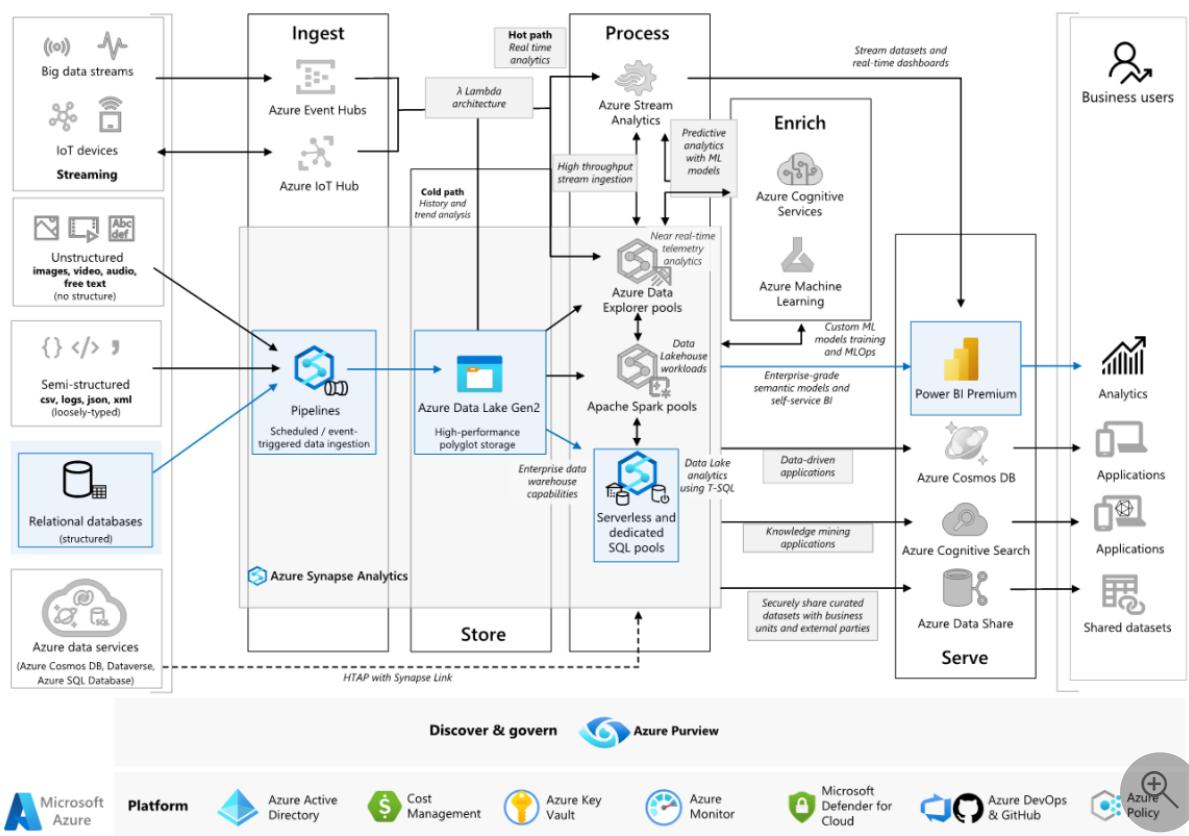
Enterprise business intelligence

Power BI Azure Synapse Analytics Azure Data Factory Microsoft Entra ID Azure Blob Storage

This example scenario shows how data can be ingested into a cloud environment from an on-premises data warehouse, then served using a business intelligence (BI) model. This approach could be an end goal or a first step toward full modernization with cloud-based components.

The following steps build on the [Azure Synapse Analytics end-to-end](#) scenario. It uses Azure Pipelines to ingest data from a SQL database into Azure Synapse SQL pools, then transforms the data for analysis.

Architecture



Download a [Visio file](#) of this architecture.

Workflow

Data source

- The source data is located in an SQL Server database in Azure. To simulate the on-premises environment, deployment scripts for this scenario provision an Azure SQL database. The [AdventureWorks sample database](#) is used as the source data schema and sample data. For information on how to copy data from an on-premises database, see [copy and transform data to and from SQL Server](#).

Ingestion and data storage

1. [Azure Data Lake Gen2](#) is used as a temporary *staging* area during data ingestion. You can then use [PolyBase](#) to [copy data into an Azure Synapse dedicated SQL pool](#).
2. [Azure Synapse Analytics](#) is a distributed system designed to perform analytics on large data. It supports massive parallel processing (MPP), which makes it suitable for running high-performance analytics. Azure Synapse dedicated SQL pool is a target for ongoing ingestion from on-premises. It can be used for further processing, as well as serving the data for [Power BI](#) through DirectQuery.
3. [Azure Pipelines](#) is used to orchestrate data ingestion and transformation within your Azure Synapse workspace.

Analysis and reporting

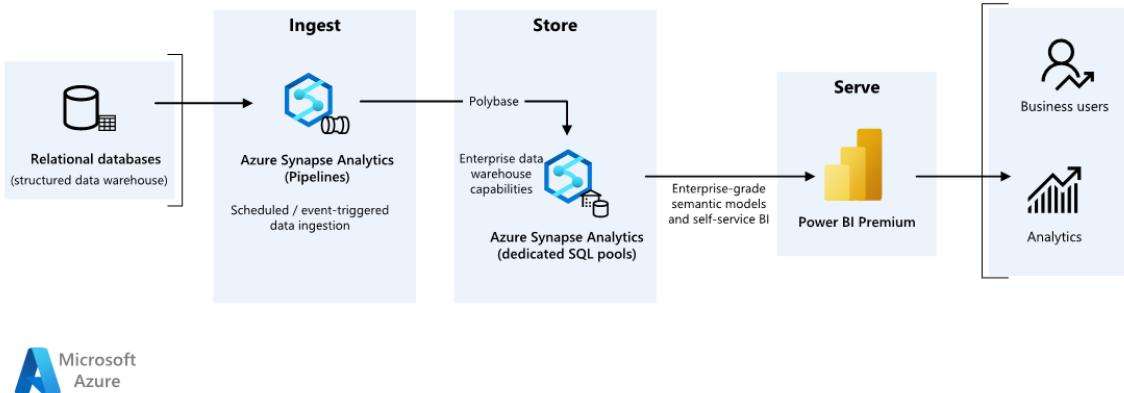
- The data-modeling approach in this scenario is presented by combining the [enterprise model](#) and [BI Semantic model](#). The enterprise model is stored in an [Azure Synapse dedicated SQL pool](#), and the BI Semantic model is stored in [Power BI Premium capacities](#). Power BI accesses the data via DirectQuery.

Components

This scenario uses the following components:

- [Azure SQL Database](#) ↗
- [Azure Data Lake](#) ↗
- [Azure Synapse Analytics](#) ↗
- [Power BI Premium](#) ↗
- [Microsoft Entra ID](#) ↗

Simplified architecture



Scenario details

An organization has a large on-premises data warehouse stored in a SQL database. The organization wants to use Azure Synapse to perform analysis, then serve these insights using Power BI.

Authentication

Microsoft Entra authenticates users who connect to Power BI dashboards and apps. Single sign-on is used to connect to the data source in Azure Synapse provisioned pool. Authorization happens on the source.

Incremental loading

When you run an automated extract-transform-load (ETL) or extract-load-transform (ELT) process, it's most efficient to load only the data that changed since the previous run. It's called an **incremental load**, as opposed to a full load that loads all the data. To perform an incremental load, you need a way to identify which data has changed. The most common approach is to use a *high water mark* value, which tracks the latest value of some column in the source table, either a datetime column or a unique integer column.

Starting with SQL Server 2016, you can use **temporal tables**, which are system-versioned tables that keep a full history of data changes. The database engine automatically records the history of every change in a separate history table. You can query the historical data by adding a `FOR SYSTEM_TIME` clause to a query. Internally, the database engine queries the history table, but it's transparent to the application.

ⓘ Note

For earlier versions of SQL Server, you can use [change data capture \(CDC\)](#). This approach is less convenient than temporal tables, because you have to query a separate change table, and changes are tracked by a log sequence number, rather than a timestamp.

Temporal tables are useful for dimension data, which can change over time. Fact tables usually represent an immutable transaction such as a sale, in which case keeping the system version history doesn't make sense. Instead, transactions usually have a column that represents the transaction date, which can be used as the watermark value. For example, in the AdventureWorks Data Warehouse, the `SalesLT.*` tables have a `LastModified` field.

Here's the general flow for the ELT pipeline:

1. For each table in the source database, track the cutoff time when the last ELT job ran. Store this information in the data warehouse. On initial setup, all times are set to `1-1-1900`.
2. During the data export step, the cutoff time is passed as a parameter to a set of stored procedures in the source database. These stored procedures query any records that were changed or created after the cutoff time. For all tables in the example, you can use the `ModifiedDate` column.
3. When the data migration is complete, update the table that stores the cutoff times.

Data pipeline

This scenario uses the [AdventureWorks sample database](#) as a data source. The incremental data load pattern is implemented to ensure we only load data that was modified or added after the most recent pipeline run.

Metadata-driven copy tool

The built-in [metadata-driven copy tool](#) within Azure Pipelines incrementally loads all tables contained within our relational database. By navigating through the wizard-based experience, you can connect the Copy Data tool to the source database, and configure either incremental or full loading for each table. The Copy Data tool then creates both the pipelines and SQL scripts to generate the control table required to store data for the incremental loading process—for example, the high watermark value/column for each table. Once these scripts are run, the pipeline is ready to load all tables in the source data warehouse into the Synapse dedicated pool.

The tool creates three pipelines to iterate over all the tables in the database, before loading the data.

The pipelines generated by this tool:

- Count the number of objects, such as tables, to be copied in the pipeline run.
- Iterate over each object to be loaded/copied and then:
 - Check whether a delta load is required; otherwise complete a normal full load.
 - Retrieve the high watermark value from the control table.
 - Copy data from the source tables into the staging account in ADLS Gen2.
 - Load data into the dedicated SQL pool via the selected copy method—for example, Polybase, Copy command.
 - Update the high watermark value in the control table.

Load data into Azure Synapse SQL pool

The [copy activity](#) copies data from the SQL database into the Azure Synapse SQL pool. In this example, because our SQL database is in Azure, we use the Azure integration runtime to read data from the SQL database and write the data into the specified staging environment.

The copy statement is then used to load data from the staging environment into the Synapse dedicated pool.

Use Azure Pipelines

Pipelines in Azure Synapse are used to define the ordered set of activities to complete the incremental load pattern. Triggers are used to start the pipeline, which can be triggered manually or at a time specified.

Transform the data

Because the sample database in our reference architecture isn't large, we created replicated tables with no partitions. For production workloads, using distributed tables is likely to improve query performance. See [Guidance for designing distributed tables in Azure Synapse](#). The example scripts run the queries using a static [resource class](#).

In a production environment, consider creating staging tables with round-robin distribution. Then transform and move the data into production tables with clustered columnstore indexes, which offer the best overall query performance. Columnstore indexes are optimized for queries that scan many records. Columnstore indexes don't perform as well for singleton lookups, that is, looking up a single row. If you need to perform frequent singleton lookups, you can add a non-clustered index to a table. Singleton lookups can run much faster using a non-clustered index. However, singleton lookups are typically less common in data warehouse scenarios than OLTP workloads. For more information, see [Indexing tables in Azure Synapse](#).

Note

Clustered columnstore tables don't support `varchar(max)`, `nvarchar(max)`, or `varbinary(max)` data types. In that case, consider a heap or clustered index. You might put those columns into a separate table.

Use Power BI Premium to access, model, and visualize data

Power BI Premium supports several options for connecting to data sources on Azure, in particular Azure Synapse provisioned pool:

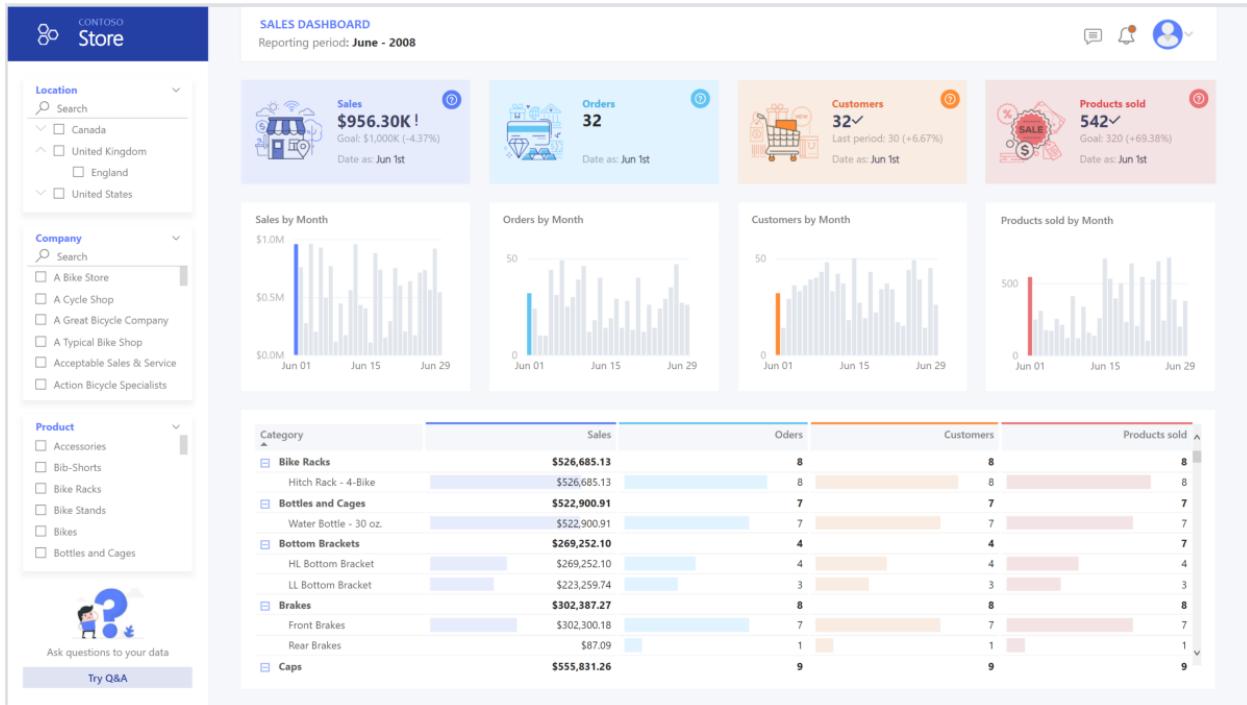
- Import: The data is imported into the Power BI model.
- [DirectQuery](#): Data is pulled directly from relational storage.
- [Composite model](#): Combine *Import* for some tables and *DirectQuery* for others.

This scenario is delivered with DirectQuery dashboard because the amount of data used and model complexity aren't high, so we can deliver a good user experience.

DirectQuery delegates the query to the powerful compute engine underneath and utilizes extensive security capabilities on the source. Also, using DirectQuery ensures that results are always consistent with the latest source data.

Import mode provides the fastest query response time, and should be considered when the model fits entirely within Power BI's memory, the data latency between refreshes can be tolerated, and there might be some complex transformations between the source system and the final model. In this case, the end users want full access to the most

recent data with no delays in Power BI refreshing, and all historical data, which is larger than what a Power BI dataset can handle—between 25-400 GB, depending on the capacity size. As the data model in the dedicated SQL pool is already in a star schema and needs no transformation, DirectQuery is an appropriate choice.



Power BI Premium Gen2 gives you the ability to handle large models, paginated reports, deployment pipelines, and built-in Analysis Services endpoint. You can also have dedicated [capacity](#) with unique value proposition.

When the BI model grows or dashboard complexity increases, you can switch to composite models and start importing parts of look-up tables, via [hybrid tables](#), and some pre-aggregated data. Enabling [query caching](#) within Power BI for imported datasets is an option, as well as utilizing [dual tables](#) for storage mode property.

Within the composite model, datasets act as a virtual pass-through layer. When the user interacts with visualizations, Power BI generates SQL queries to Synapse SQL pools dual storage: in memory or direct query depending on which one is more efficient. The engine decides when to switch from in-memory to direct query and pushes the logic to the Synapse SQL pool. Depending on the context of the query tables, they can act as either cached (imported) or not cached composite models. Pick and choose which table to cache into memory, combine data from one or more DirectQuery sources, and/or combine data from a mix of DirectQuery sources and imported data.

Recommendations: When using DirectQuery over Azure Synapse Analytics provisioned pool:

- Use Azure Synapse [result set caching](#) to cache query results in the user database for repetitive use, improve query performance down to milliseconds, and reduce

compute resource usage. Queries using cached results sets don't use any concurrency slots in Azure Synapse Analytics and thus don't count against existing concurrency limits.

- Use Azure Synapse [materialized views](#) to pre-compute, store, and maintain data just like a table. Queries that use all or a subset of the data in materialized views can get faster performance, and they don't need to make a direct reference to the defined materialized view to use it.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

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](#).

Frequent headlines of data breaches, malware infections, and malicious code injection are among an extensive list of security concerns for companies looking to cloud modernization. Enterprise customers need a cloud provider or service solution that can address their concerns as they can't afford to get it wrong.

This scenario addresses the most demanding security concerns using a combination of layered security controls: network, identity, privacy, and authorization. The bulk of the data is stored in Azure Synapse provisioned pool, with Power BI using DirectQuery through single sign-on. You can use Microsoft Entra ID for authentication. There are also extensive security controls for data authorization of provisioned pools.

Some common security questions include:

- How can I control who can see what data?
 - Organizations need to protect their data to comply with federal, local, and company guidelines to mitigate risks of data breach. Azure Synapse offers multiple [data protection capabilities](#) to achieve compliance.
- What are the options for verifying a user's identity?
 - Azure Synapse supports a wide range of capabilities to control who can access what data via [access control](#) and [authentication](#).
- What network security technology can I use to protect the integrity, confidentiality, and access of my networks and data?

- To secure Azure Synapse, there are a range of [network security](#) options available to consider.
- What are the tools that detect and notify me of threats?
 - Azure Synapse provides many [threat detection](#) capabilities like: SQL auditing, SQL threat detection, and vulnerability assessment to audit, protect, and monitor databases.
- What can I do to protect data in my storage account?
 - Azure Storage accounts are ideal for workloads that require fast and consistent response times, or that have a high number of input-output operations (IOP) per second. Storage accounts contain all your Azure Storage data objects, and have many options for [storage account security](#).

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

This section provides information on pricing for different services involved in this solution, and mentions decisions made for this scenario with a sample dataset.

Azure Synapse

Azure Synapse Analytics serverless architecture allows you to scale your compute and storage levels independently. Compute resources are charged based on usage, and you can scale or pause these resources on demand. Storage resources are billed per terabyte, so your costs will increase as you ingest more data.

Azure Pipelines

Pricing details for pipelines in Azure Synapse can be found under the *Data Integration* tab on the [Azure Synapse pricing page](#). There are three main components that influence the price of a pipeline:

1. Data pipeline activities and integration runtime hours
2. Data flows cluster size and execution
3. Operation charges

The price varies depending on the components or activities, frequency, and number of integration runtime units.

For the sample dataset, the standard Azure-hosted integration runtime, *copy data activity* for the core of the pipeline, is triggered on a daily schedule for all of the entities (tables) in the source database. The scenario contains no data flows. There are no operational costs since there are fewer than 1 million operations with pipelines a month.

Azure Synapse dedicated pool and storage

Pricing details for Azure Synapse dedicated pool can be found under the *Data Warehousing* tab on the [Azure Synapse pricing page](#). Under the Dedicated consumption model, customers are billed per DWU units provisioned, per hour of uptime. Another contributing factor is data storage costs: size of your data at rest + snapshots + geo-redundancy, if any.

For the sample dataset, you can provision 500DWU, which guarantees a good experience for analytical load. You can keep compute up and running over business hours of reporting. If taken into production, reserved data warehouse capacity is an attractive option for cost management. Different techniques should be used to maximize cost/performance metrics, which are covered in the previous sections.

Blob storage

Consider using the Azure Storage reserved capacity feature to lower storage costs. With this model, you get a discount if you reserve fixed storage capacity for one or three years. For more information, see [Optimize costs for Blob storage with reserved capacity](#).

There's no persistent storage in this scenario.

Power BI Premium

Power BI Premium pricing details can be found on the [Power BI pricing page](#).

This scenario uses [Power BI Premium workspaces](#) with a range of performance enhancements built in to accommodate demanding analytical needs.

Operational excellence

Operational excellence covers the operations processes that deploy an application and keep it running in production. For more information, see [Overview of the operational excellence pillar](#).

DevOps recommendations

- Create separate resource groups for production, development, and test environments. Separate resource groups make it easier to manage deployments, delete test deployments, and assign access rights.
- Put each workload in a separate deployment template and store the resources in source control systems. You can deploy the templates together or individually as part of a continuous integration (CI) and continuous delivery (CD) process, making the automation process easier. In this architecture, there are four main workloads:
 - The data warehouse server, and related resources
 - Azure Synapse pipelines
 - Power BI assets: dashboards, apps, datasets
 - An on-premises to cloud simulated scenario

Aim to have a separate deployment template for each of the workloads.

- Consider staging your workloads where practical. Deploy to various stages and run validation checks at each stage before moving to the next stage. That way you can push updates to your production environments in a controlled way and minimize unanticipated deployment issues. Use [blue-green deployment](#) and [canary release](#) strategies for updating live production environments.
- Have a good rollback strategy for handling failed deployments. For example, you can automatically redeploy an earlier, successful deployment from your deployment history. See the `--rollback-on-error` flag in Azure CLI.
- [Azure Monitor](#) is the recommended option for analyzing the performance of your data warehouse and the entire Azure analytics platform for an integrated monitoring experience. [Azure Synapse Analytics](#) provides a monitoring experience within the Azure portal to show insights about your data warehouse workload. The Azure portal is the recommended tool when monitoring your data warehouse because it provides configurable retention periods, alerts, recommendations, and customizable charts and dashboards for metrics and logs.

Quick start

- Portal: [Azure Synapse proof of concept](#)
- Azure CLI: [Create an Azure Synapse workspace with Azure CLI](#)
- Terraform: [Modern data warehousing with Terraform and Microsoft Azure](#)

Performance efficiency

Performance efficiency is the ability of your workload to scale to meet the demands placed on it by users in an efficient manner. For more information, see [Performance efficiency pillar overview](#).

This section provides details on sizing decisions to accommodate this dataset.

Azure Synapse provisioned pool

There's a range of [data warehouse configurations](#) to choose from.

 [Expand table](#)

Data warehouse units	# of compute nodes	# of distributions per node
DW100c	1	60
-- TO --		
DW30000c	60	1

To see the performance benefits of scaling out, especially for larger data warehouse units, use at least a 1-TB dataset. To find the best number of data warehouse units for your dedicated SQL pool, try scaling up and down. Run a few queries with different numbers of data warehouse units after loading your data. Since scaling is quick, you can try various performance levels in an hour or less.

Find the best number of data warehouse units

For a dedicated SQL pool in development, begin by selecting a smaller number of data warehouse units. A good starting point is *DW400c* or *DW200c*. Monitor your application performance, observing the number of data warehouse units selected compared to the performance you observe. Assume a linear scale, and determine how much you need to increase or decrease the data warehouse units. Continue making adjustments until you reach an optimum performance level for your business requirements.

Scaling Synapse SQL pool

- [Scale compute for Synapse SQL pool with the Azure portal](#)
- [Scale compute for dedicated SQL pool with Azure PowerShell](#)
- [Scale compute for dedicated SQL pool in Azure Synapse Analytics using T-SQL](#)

- Pausing, monitoring, and automation

Azure Pipelines

For scalability and performance optimization features of pipelines in Azure Synapse and the copy activity used, refer to the [Copy activity performance and scalability guide](#).

Power BI Premium

This article uses [Power BI Premium Gen 2](#) to demonstrate BI capabilities. [Capacity SKUs for Power BI Premium](#) range from P1 (eight v-cores) to P5 (128 v-cores) currently. The best way to select needed capacity is to undergo [capacity loading evaluation](#), install the Gen 2 [metrics app](#) for ongoing monitoring, and consider using [Autoscale with Power BI Premium](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Galina Polyakova](#) | Senior Cloud Solution Architect
- [Noah Costar](#) | Cloud Solution Architect
- [George Stevens](#) | Cloud Solution Architect

Other contributors:

- [Jim McLeod](#) | Cloud Solution Architect
- [Miguel Myers](#) | Senior Program Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [What is Power BI Premium?](#)
- [What is Microsoft Entra ID?](#)
- [Accessing Azure Data Lake Storage Gen2 and Blob Storage with Azure Databricks](#)
- [What is Azure Synapse Analytics?](#)
- [Pipelines and activities in Azure Data Factory and Azure Synapse Analytics](#)
- [What is Azure SQL?](#)

Related resources

- [Automated enterprise BI](#)
- [Analytics end-to-end with Azure Synapse](#)
- [Big data analytics with enterprise-grade security using Azure Synapse](#)

Geospatial analysis for the telecommunications industry

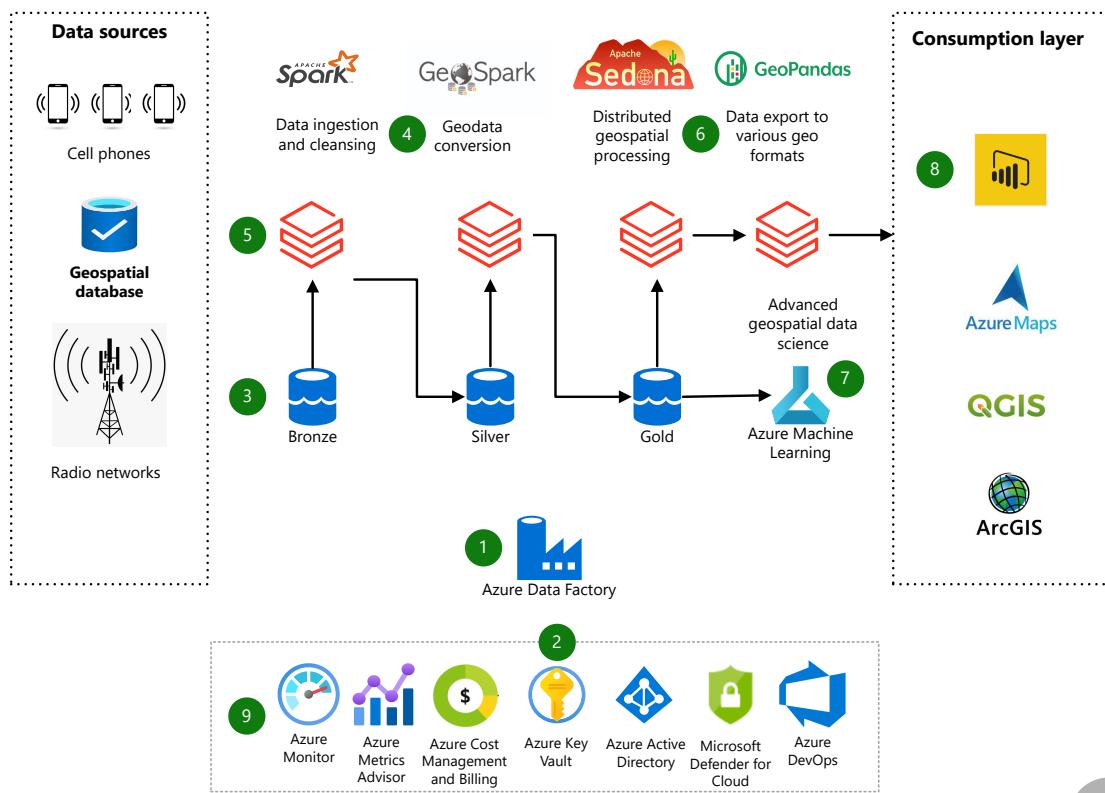
Azure Data Factory Azure Data Lake Azure Databricks Azure Machine Learning Azure Maps

The focus of this article is to showcase a practical architecture that uses Azure Cloud Services to process large volumes of geospatial data. It provides a path forward when on-premises solutions don't scale. It also allows for continued use of the current geospatial analysis tools.

Apache®, Apache Spark®, GeoSpark®, and Sedona® 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.

GeoPandas®, QGIS®, and ArcGIS® are trademarks of their respective companies. No endorsement is implied by the use of these marks.

Architecture



Download a [Visio file](#) of this architecture.

Workflow

1. Azure Data Factory ingests geospatial data into Azure Data Lake Storage. The source of this data is geospatial databases such as Teradata, Oracle Spatial, and PostgreSQL.
2. Azure Key Vault secures passwords, credentials, connection strings, and other secrets.
3. Data is placed in various folders and file systems in Data Lake Storage according to how it has been processed. The diagram shows a *multi-hop* architecture. The bronze container holds raw data, the silver container holds semi-curated data, and the gold container holds fully curated data.
4. Data is stored in formats such as [GeoJson](#), [WKT](#) and [Vector tiles](#). Azure Databricks and the GeoSpark / [Sedona](#) package can convert formats and efficiently load, process, and analyze large-scale spatial data across machines.
5. Azure Databricks and Apache Sedona do various kinds of processing at scale:
 - a. Joins, intersections, and tessellations
 - b. Spatial sampling and statistics
 - c. Spatial indexing and partitioning
6. [GeoPandas](#) exports data in various formats for use by third-party GIS applications such as QGIS and ARCGIS.
7. Azure Machine Learning extracts insights from geospatial data, determining, for example, where and when to deploy new wireless access points.
8. Power BI and Azure Maps Power BI visual (Preview) render a map canvas to visualize geospatial data. Power BI uses an Azure Databricks native connector to connect to an Azure Databricks cluster.
9. Log Analytics, a tool in the Azure portal, runs queries against data in Azure Monitor Logs to implement a robust and fine-grained logging system to analyze events and performance.

Components

- [Azure Data Lake Storage](#) is a scalable and secure data lake for high-performance analytics workloads. You can use Data Lake Storage to manage petabytes of data with high throughput. It can accommodate multiple, heterogeneous sources, and data that's in structured, semi-structured, or unstructured formats.
- [Azure Databricks](#) is a data analytics platform that uses Spark clusters. The clusters are optimized for the Azure Cloud Services platform.

- [Azure Data Factory](#) is a fully managed, scalable, and serverless data integration service. It provides a data integration and transformation layer that works with various data stores.
- [Microsoft Power BI](#) is a collection of software services, apps, and connectors that work together to turn multiple sources of data into coherent, visually immersive, and interactive insights.
- [Azure Maps](#) is a collection of geospatial services and SDKs that use fresh mapping data to provide geographic context to web and mobile applications.
- [Azure Machine Learning](#) is a fully managed cloud service that's used to train, deploy, and manage machine learning models at scale.
- [Azure Key Vault](#) is a service that can be used to securely store, manage, and tightly control access to tokens, credentials, certificates, API Keys, and other secrets.
- [Azure Monitor](#) is a comprehensive solution for collecting, analyzing, and acting on telemetry from your cloud and on-premises environments. You can use it to maximize the availability and performance of your applications and services.

Alternatives

- You can use [Synapse Spark Pools](#) for geospatial analytics instead of Azure Databricks, using the same open-source frameworks.
- Instead of using Data Factory to ingest data, you can use [Azure Event Hubs](#). It can receive massive amounts of data directly or from other event streaming services such as Kafka. Then you can use Azure Databricks to process the data. For more information, see [Stream Processing with Azure Databricks](#).
- Instead of Azure Databricks, you can use [Azure SQL Database](#) or [Azure SQL Managed Instance](#) to query and process geospatial data. These databases provide the familiar T-SQL language, which you can use for geospatial analysis. For more information, see [Spatial Data \(SQL Server\)](#).
- Like Event Hubs, [Azure IoT Hub](#) can ingest large amounts of data from sensor and telecom IoT devices. You can use the IoT Hub bi-directional capability to communicate securely with devices and potentially manage and control them from a centralized platform in the cloud.
- You can use [Azure Maps](#) to provide geographic context to your web and mobile applications. In addition to location intelligence, Azure Maps can search services to locate addresses, places, and points of interest to get real-time traffic information. [Azure Maps Power BI Visual](#) provides the same capabilities in both [Power BI Desktop](#) and the [Power BI service](#).

Scenario details

Location intelligence and geospatial analytics can uncover important regional trends and behaviors that affect telecommunications companies. The companies can use such knowledge to enhance their radio signal and wireless coverage, and thus gain competitive advantage.

Telecommunications companies have large volumes of geographically dispersed asset data, most of which is user telemetry. The data comes from radio networks, IoT sensing devices, and remote sensing devices that capture geospatial data. It's in various structured and semi-structured formats such as imagery, GPS, satellite, and textural. Making use of it requires aggregating it and joining it with other sources such as regional maps and traffic data.

After the data is aggregated and joined, the challenge is to extract insights from it. Historically, telecommunications companies relied on legacy systems such as on-premises databases with geospatial capabilities. Eventually such systems hit scalability limits due to the ever-increasing amount of data. Also, they require third-party software to perform tasks that the geospatial database systems can't.

Potential use cases

This solution is ideal for the telecommunications industry, and it applies to the following scenarios:

- Analyzing signal information across locations to assess network quality
- Analyzing real-time network infrastructure data to guide maintenance and repair
- Analyzing market segmentation and market demand
- Identifying relationships between customer locations and company marketing campaigns
- Creating capacity and coverage plans to ensure connectivity and quality of service

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Consider following the [Microsoft Azure Well-Architected Framework](#) when you implement this solution. The framework provides technical guidance across five pillars: cost optimization, security, reliability, performance efficiency, and operational excellence.

Performance

- Follow [Apache Sedona programming guides](#) on design patterns and performance tuning best practices.
- Geospatial indexing is crucial for processing large-scale geospatial data. Apache Sedona and other open-source indexing frameworks such as [H3](#) provide this capability.
- The GeoPandas framework doesn't have the distributed features of GeoSpark / Apache Sedona. Therefore, as much as possible, use Sedona framework for geospatial processing.
- Consider using Sedona's built-in functions to validate geometry formatting before processing.

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](#).

For better security, consider following this guidance:

- [Create an Azure Key Vault-backed secret scope](#)
- [Secure cluster connectivity \(No Public IP / NPIP\)](#)
- [Store credentials in Azure Key Vault](#)
- [Deploy dedicated Azure services into virtual networks](#)
- Consider using Azure Databricks Premium tier instead of Standard for more security features
- [Databricks security guide](#)

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

- To estimate the cost of implementing this solution, use the [Azure Pricing Calculator](#) for the services mentioned above.
- Power BI comes with various licensing offerings. For more information, see [Power BI pricing](#).
- Your costs increase if you have to scale your Azure Databricks cluster configurations. This depends on the amount of data and the complexity of the analysis. For best practices on cluster configuration, see Azure Databricks [Best practices: Cluster configuration](#).

- See [Overview of the cost optimization pillar](#) for ways to minimize costs.
- For the third-party components such as QGIS and ARCGIS, see the vendor websites for pricing information.
- The frameworks mentioned in this solution, such as Apache Sedona and GeoPandas, are free open-source frameworks.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Arash Mosharraf](#) | Senior Cloud Solution Architect

Next steps

- [Introduction to Azure Data Lake Storage Gen2](#)
- [What is Power BI?](#)
- [What is Azure Maps?](#)
- [What is Azure Machine Learning?](#)
- [About Azure Key Vault](#)
- [Azure Monitor overview](#)
- [Azure Maps samples](#)
- [Azure Data Factory tutorials](#)
- [Apache Sedona programming guides](#)
- [Getting Started with GeoPandas](#)
- [Getting started with GeoMesa](#)
- [Processing Geospatial Data at Scale With Databricks](#)
- [GIS file formats](#)
- [Apache Sedona reference](#)
- [Overview of the H3 Geospatial Indexing System](#)
- [Power BI and Esri ArcGIS](#)
- [QGIS](#)
- [H3: A Hexagonal Hierarchical Geospatial Indexing System](#)
- [How To Turn Visitor Cellphone Roaming Data Into Revenue?](#)
- [5G positioning: What you need to know](#)

Related resources

- Geospatial data processing and analytics
- Solutions for the telecommunications industry

Spaceborne data analysis with Azure Synapse Analytics

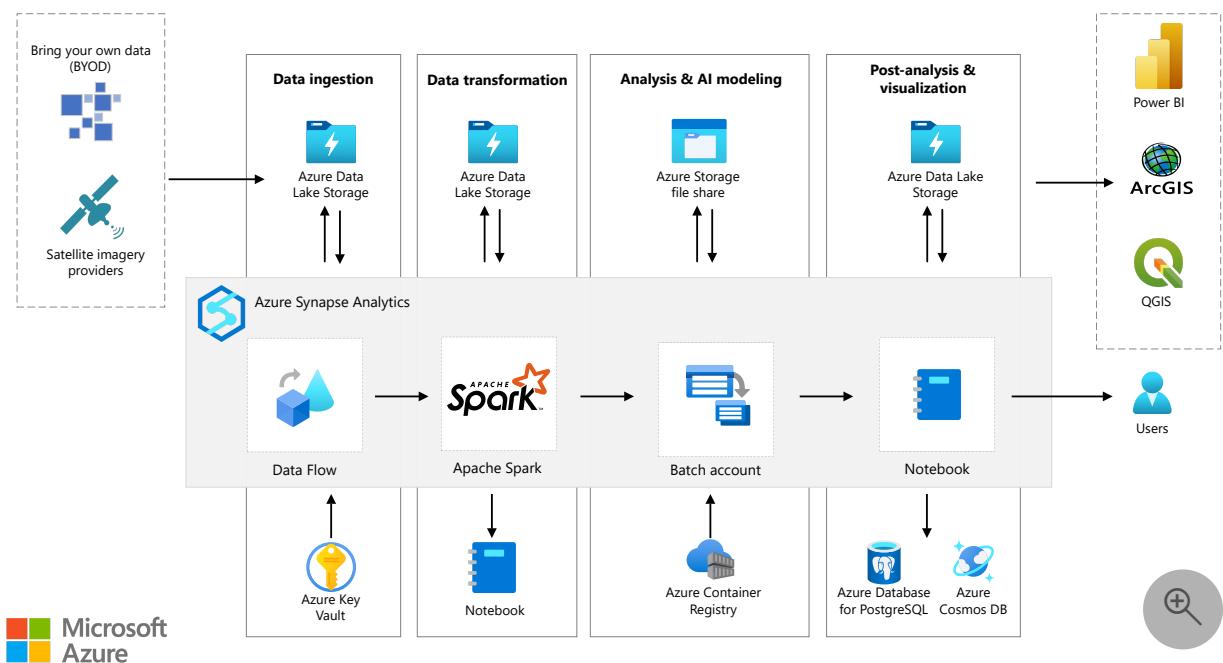
Azure Synapse Analytics Azure AI services Azure Computer Vision Azure Batch Azure Storage

This architecture is designed to show an end-to-end implementation that involves extracting, loading, transforming, and analyzing spaceborne data by using geospatial libraries and AI models with [Azure Synapse Analytics](#). This article also shows how to integrate geospatial-specific [Azure Cognitive Services](#) models, AI models from partners, bring-your-own-data, and AI models that use Azure Synapse Analytics. The intended audience for this document is users with intermediate skill levels in working with geospatial or spaceborne data.

An implementation of this architecture is available on [GitHub](#).

Apache®, Apache Spark, Spark, the Spark logo, Apache Sedona, Apache Incubator, the Apache feather logo and the Apache Incubator project 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 following sections describe the stages in the architecture.

Data ingestion

Spaceborne data is pulled from data sources like [Airbus](#), [NAIP/USDA \(via the Planetary Computer API\)](#), and [Maxar](#). Data is ingested into [Azure Data Lake Storage](#).

Azure Synapse Analytics provides various pipelines and activities, like Web activity, Data Flow activity, and Custom activities, to connect to these sources and copy the data into Data Lake Storage.

Azure Synapse Custom activities run your customized code logic on an [Azure Batch](#) pool of virtual machines or in [Docker-compatible containers](#).

Data transformation

The data is processed and transformed into a format that analysts and AI models can consume. Geospatial libraries, including GDAL, OGR, Rasterio, and GeoPandas, are available to perform the transformation.

Azure Synapse Spark pools provide the ability to configure and use these libraries to perform the data transformations. You can also use Azure Synapse Custom activities, which use Azure Batch pools.

An [Azure Synapse notebook](#) is a web interface that you can use to create files that contain live code, visualizations, and narrative text. Notebooks are a good place to validate ideas, define transformations, and do quick experiments to get insights from your data and build a pipeline. In the sample code, the GDAL library is used in a Spark pool to perform data transformations. For more information, see the [sample code](#) section of this article.

The sample solution implements this pipeline from this data transformation step. The sample is written with the assumption that data is copied in Data Lake Storage by the data ingestion methods described earlier. It demonstrates implementation of this pipeline for raster data processing.

Analysis and execution of AI models

The Azure Synapse notebook environment analyzes and runs AI models.

AI models developed with services like the Cognitive Services Custom Vision model, trained in their own environment, and packaged as Docker containers are available in the Azure Synapse environment.

In the Azure Synapse environment, you can also run AI models that are available from partners for various capabilities like object detection, change detection, and land classification. These models are trained in their own environment and packaged as Docker containers.

Azure Synapse can run such AI models via a Custom activity that runs code in Batch pools as executables or Docker containers. The sample solution demonstrates how to run a [Custom Vision AI model](#) as part of an Azure Synapse pipeline for object detection over a specific geospatial area.

Post-analysis and visualization

- For further analysis and visualization, output from analysis and execution of the AI models can be stored in Data Lake Storage, data-aware databases like Azure Database for PostgreSQL, or Azure Cosmos DB. The sample solution shows how to transform AI model output and store it as [GeoJSON](#) data in Data Lake Storage and Azure Database for PostgreSQL. You can retrieve and query the output from there.
- For visualization:
 - You can use licensed tools like ArcGIS Desktop or open-source tools like QGIS.
 - You can use Power BI to access GeoJSON from various data sources and visualize the geographic information system (GIS) data.
 - You can use client-side geospatial JavaScript-based libraries to visualize the data in web applications.

Components

Data sources

- **Imagery providers.**
 - [Airbus](#)
 - [NAIP/USDA \(via the Planetary Computer API\)](#)
 - [Maxar](#)
- **Bring your own data.** Copy your own data to Data Lake Storage.

Data ingestion

- [Azure Synapse Analytics](#) is a limitless analytics service that brings together data integration, enterprise data warehousing, and big data analytics. Azure Synapse contains the same Data Integration engine and experiences as Azure Data Factory, so you can create at-scale ETL pipelines without leaving Azure Synapse.
- [Azure Data Lake Storage](#) is dedicated to big data analytics, and is built on [Azure Blob Storage](#).
- [Azure Batch](#) enables you to run and scale a large number of batch computing jobs on Azure. Batch tasks can run directly on virtual machines (nodes) in a Batch pool, but you can also set up a Batch pool to run tasks in [Docker-compatible containers](#) on the nodes.
 - An Azure Synapse Custom activity runs customized code logic on an Azure Batch pool of virtual machines or in Docker containers.
- [Azure Key Vault](#) stores and controls access to secrets like tokens, passwords, and API keys. Key Vault also creates and controls encryption keys and manages security certificates.

Data transformation

The following geospatial libraries and packages are used together for transformations. These libraries and packages are installed in a serverless Spark pool, which is then attached to an Azure Synapse notebook. For information on installing the libraries, see [Install geospatial packages in an Azure Synapse Spark pool](#), later in this article.

- **Geospatial libraries**
 - [GDAL](#) is a library of tools for manipulating spaceborne data. GDAL works on raster and vector data types. It's a good tool to know if you're working with spaceborne data.
 - [Rasterio](#) is a module for raster processing. You can use it to read and write several different raster formats in Python. Rasterio is based on GDAL. When the module is imported, Python automatically registers all known GDAL drivers for reading supported formats.
 - [GeoPandas](#) is an open-source project that can make it easier to work with spaceborne data in Python. GeoPandas extends the data types used by Pandas to allow spatial operations on geometric types.
 - [Shapely](#) is a Python package for set-theoretic analysis and manipulation of planar features. It uses (via Python's ctypes module) functions from the widely deployed GEOS library.
 - [pyproj](#) performs cartographic transformations. It converts from longitude and latitude to native map projection x, y coordinates, and vice versa, by using [PROJ](#).

- [Azure Batch](#) enables you to run and scale a large number of batch computing jobs on Azure.
- [Azure Synapse notebooks](#) is a web interface for creating files that contain live code, visualizations, and narrative text. You can add existing Azure Synapse notebooks to an Azure Synapse pipeline by using the Notebook activity.
- [Apache Spark pool](#) provides the ability to configure and use libraries to perform data transformations. You can add existing Spark jobs to an Azure Synapse pipeline by using the Spark Job Definition activity.

Analysis and AI modeling

- [Azure Synapse](#) provides machine learning capabilities.
- [Azure Batch](#) enables you to run and scale a large number of batch computing jobs on Azure. In this solution, the Azure Synapse Custom activity is used to run Docker-based AI models on Azure Batch pools.
- [Azure Cognitive Services](#) provides the ability to embed vision into your apps. You can use [Custom Vision](#), a component of Cognitive Services, to customize and embed state-of-the-art computer vision image analysis for specific domains.
- You can also use bring-your-own AI models and Microsoft partner AI models like [blackshark.ai](#).

Post-analysis and visualization links

- [Azure Database for PostgreSQL](#) is a fully managed relational database service designed for hyperscale workloads. It supports spaceborne data via the [PostGIS](#) extension.
- [Azure Cosmos DB](#) supports indexing and querying of geospatial point data that's represented in [GeoJSON](#).
- [Power BI](#) is an interactive data visualization tool for building reports and dashboards. You can get insights on spaceborne data from Esri [ArcGIS Maps](#).
- [QGIS](#) is a free open-source GIS for creating, editing, visualizing, analyzing, and publishing geospatial information.
- [ArcGIS Desktop](#) is licensed product provided by Esri. You can use it to create, analyze, manage, and share geographic information.

Alternatives

If you want to run containerized AI models that you can call from Azure Synapse, you can use [Azure Kubernetes Service](#), [Azure Container Instances](#), or [Azure Container Apps](#).

[Azure Databricks](#) provides an alternative for hosting an analytics pipeline.

[Spark in Azure HDInsight](#) provides an alternative for using geospatial libraries in the Apache Spark environment.

Here are some alternative libraries and frameworks that you can use for spaceborne data processing:

- [Apache Sedona](#), formerly named GeoSpark, is a cluster computing system for processing large-scale spatial data. Sedona extends Spark and Spark SQL with out-of-the-box Spatial Resilient Distributed Datasets and SpatialSQL that efficiently load, process, and analyze large-scale spatial data across machines.
- [Dask for Python](#) is a parallel computing library that scales the existing Python ecosystem.

Scenario details

Spaceborne data collection is increasingly common. For the application of artificial intelligence, stored archives of data are necessary for machine learning. The need to build a cloud-based solution for spaceborne data analysis has become more important to enable enterprises and governments to drive better-informed business and tactical decisions.

Potential use cases

This solution is ideal for the aerospace and aircraft industries. It addresses these scenarios:

- Raster data ingestion and processing
- Object detection via pre-trained AI models
- Classification of land masses via AI models
- Monitoring changes in the environment via AI models
- Derived datasets from preprocessed imagery sets
- Vector visualization / small-area consumption
- Vector data filtering and cross-data joins

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Operational excellence

If you collaborate by using Git for source control, you can use Synapse Studio to associate your workspace with a Git repository, Azure DevOps, or GitHub. For more information, see [Source control in Synapse Studio](#).

- In an Azure Synapse workspace, CI/CD moves all entities from one environment (development, test, production) to another environment.
- You can use Azure DevOps release pipelines and GitHub Actions to automate the deployment of an Azure Synapse workspace to multiple environments.

Performance

- Azure Synapse supports [Apache Spark 3.1.2](#), which is more performant than its predecessors.
- For information about Spark pool scaling and node sizes, see [Spark pools in Azure Synapse Analytics](#).
- With [Azure Batch](#), you can scale out intrinsically parallel for transformations submitted in an Azure Synapse Custom activity. Azure Batch supports specialized GPU-optimized VM sizes that you can use to run AI models.

Reliability

Reliability ensures your application can meet the commitments you make to your customers. For more information, see [Overview of the reliability pillar](#).

For SLA information, see [Azure Synapse SLA](#).

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](#).

See these articles for security best practices:

- [Azure Synapse Analytics security: Introduction](#)
- [Azure Synapse Analytics security: Data protection](#)
- [Azure Synapse Analytics security: Access control](#)
- [Azure Synapse Analytics security: Authentication](#)
- [Azure Synapse Analytics: Network security](#)

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

These resources provide information about pricing and cost optimization:

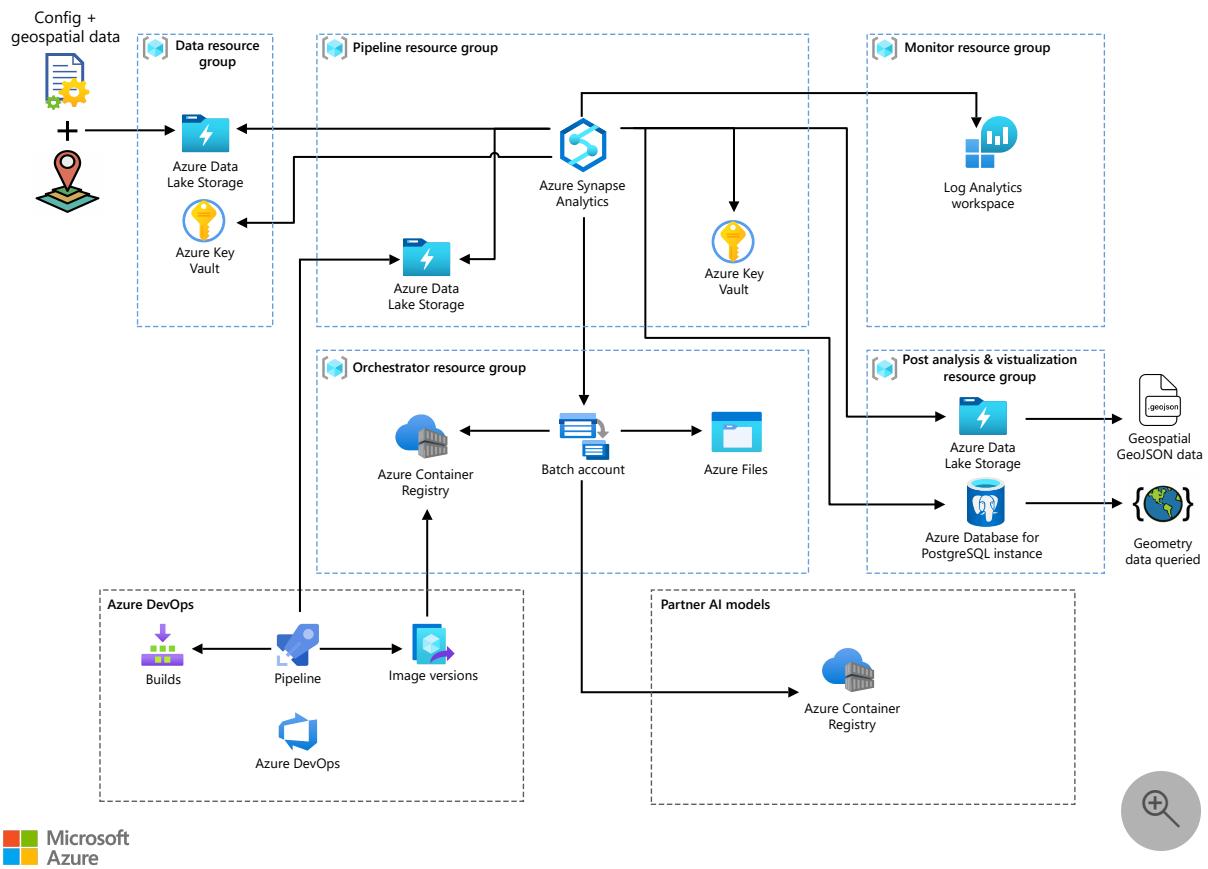
- [Plan and manage costs for Azure Synapse](#)
- [Azure Synapse in the Azure pricing calculator ↗](#)
- [Apache Spark pool in Azure Synapse](#)
- [Nodes and pools in Azure Batch](#)
- [Azure Batch in the Azure pricing calculator ↗](#)

Note

For pricing and license terms for partner AI models, see the partner's documentation.

Deploy this scenario

A [Bicep](#) deployment of the sample solution is available. To get started with this deployment, see [these instructions ↗](#).



Download a [Visio file](#) of this architecture.

Limitations

This architecture demonstrates an end-to-end geoprocessing and analytics solution that uses Azure Synapse. This sample implementation is targeted for a small to medium area of interest and limited concurrent geoprocessing of raster data.

Sample code

The following instructions describe how to read, write, and apply transformations to raster data that's stored in Azure Data Lake Storage by using a Synapse notebook. The intention is more to demonstrate the use of libraries in Synapse notebooks than to demonstrate the transformation.

Prerequisites

- [Install the geospatial libraries](#).
- [Create an Azure key vault](#) to store secrets. In this scenario, we'll store the access key of the storage account in the key vault. For instructions, see [Store credentials in Azure Key Vault](#).
- [Create a linked service](#) for Azure Key Vault by using Azure Synapse.

Instructions

- Print information from the raster data:

Python

```
from osgeo import gdal
gdal.UseExceptions()
access_key = TokenLibrary.getSecret('<key-vault-name>', '<secret-name>')
gdal.SetConfigOption('AZURE_STORAGE_ACCOUNT', '<storage_account_name>')
gdal.SetConfigOption('AZURE_STORAGE_ACCESS_KEY', access_key)
dataset_info = gdal.Info('/vsiadls/aoa/input/sample_image.tif')
#/vsiadls/<container_name>/path/to/image
print(dataset_info)
```

ⓘ Note

`/vsiadls/` is a file system handler that enables on-the-fly random reading of primarily non-public files that are available in Azure Data Lake Storage file systems. Prior download of the entire file isn't required. `/vsiadls/` is similar to `/vsiaz/`. It uses the same configuration options for authentication. Unlike `/vsiaz/`, `/vsiadls/` provides real directory management and Unix-style ACL support. For some features, hierarchical support needs to be turned on in Azure storage. For more information, see the [/vsiadls/ documentation](#).

Output

```
Driver: GTiff/GeoTIFF
Files: /vsiadls/naip/input/sample_image.tif
Size is 6634, 7565
Coordinate System is:
PROJCRS["NAD83 / UTM zone 16N",
    BASEGEOGCRS["NAD83",
        DATUM["North American Datum 1983",
            ELLIPSOID["GRS 1980",6378137,298.257222101,
                LENGTHUNIT["metre",1]]],
        PRIMEM["Greenwich",0,
            ANGLEUNIT["degree",0.0174532925199433]],
        ID["EPSG",4269]],
    CONVERSION["UTM zone 16N",
        METHOD["Transverse Mercator",
            ID["EPSG",9807]],
        PARAMETER["Latitude of natural origin",0,
            ANGLEUNIT["degree",0.0174532925199433],
            ID["EPSG",8801]],
        PARAMETER["Longitude of natural origin",-87,
            ANGLEUNIT["degree",0.0174532925199433],
            ID["EPSG",8802]]],
```

```

PARAMETER["Scale factor at natural origin",0.9996,
  SCALEUNIT["unity",1],
  ID["EPSG",8805]],
PARAMETER["False easting",500000,
  LENGTHUNIT["metre",1],
  ID["EPSG",8806]],
PARAMETER["False northing",0,
  LENGTHUNIT["metre",1],
  ID["EPSG",8807]]],
CS[Cartesian,2],
  AXIS["(E)",east,
    ORDER[1],
    LENGTHUNIT["metre",1]],
  AXIS["(N)",north,
    ORDER[2],
    LENGTHUNIT["metre",1]],
USAGE[
  SCOPE["Engineering survey, topographic mapping."],
  AREA["North America - between 90°W and 84°W - onshore and
offshore. Canada - Manitoba; Nunavut; Ontario. United States (USA) -
Alabama; Arkansas; Florida; Georgia; Indiana; Illinois; Kentucky;
Louisiana; Michigan; Minnesota; Mississippi; Missouri; North Carolina;
Ohio; Tennessee; Wisconsin."],
  BBOX[23.97,-90,84,-84]],
  ID["EPSG",26916]
Data axis to CRS axis mapping: 1,2
Origin = (427820.00000000000000,3395510.00000000000000)
Pixel Size = (1.00000000000000,-1.00000000000000)
Metadata:
  AREA_OR_POINT=Area
Image Structure Metadata:
  COMPRESSION=DEFLATE
  INTERLEAVE=PIXEL
  LAYOUT=COG
  PREDICTOR=2
Corner Coordinates:
Upper Left  ( 427820.000, 3395510.000) ( 87d45'13.12"W, 30d41'24.67"N)
Lower Left  ( 427820.000, 3387945.000) ( 87d45'11.21"W, 30d37'18.94"N)
Upper Right ( 434454.000, 3395510.000) ( 87d41' 3.77"W, 30d41'26.05"N)
Lower Right ( 434454.000, 3387945.000) ( 87d41' 2.04"W, 30d37'20.32"N)
Center      ( 431137.000, 3391727.500) ( 87d43' 7.54"W, 30d39'22.51"N)
Band 1 Block=512x512 Type=Byte, ColorInterp=Red
  Overviews: 3317x3782, 1658x1891, 829x945, 414x472
Band 2 Block=512x512 Type=Byte, ColorInterp=Green
  Overviews: 3317x3782, 1658x1891, 829x945, 414x472
Band 3 Block=512x512 Type=Byte, ColorInterp=Blue
  Overviews: 3317x3782, 1658x1891, 829x945, 414x472
Band 4 Block=512x512 Type=Byte, ColorInterp=Undefined
  Overviews: 3317x3782, 1658x1891, 829x945, 414x472

```

- Convert GeoTiff to PNG by using GDAL:

Python

```

from osgeo import gdal
gdal.UseExceptions()
access_key = TokenLibrary.getSecret('<key-vault-name>', '<secret-name>')
gdal.SetConfigOption('AZURE_STORAGE_ACCOUNT', '<storage_account_name>')
gdal.SetConfigOption('AZURE_STORAGE_ACCESS_KEY', access_key)
tiff_in = "/vsiadls/aoa/input/sample_image.tiff"
#/vsiadls/<container_name>/path/to/image
png_out = "/vsiadls/aoa/input/sample_image.png"
#/vsiadls/<container_name>/path/to/image
options = gdal.TranslateOptions(format='PNG')
gdal.Translate(png_out, tiff_in, options=options)

```

- Store GeoTiff images in Azure Data Lake Storage.

Because of how data is stored in the cloud and the fact that the file handlers `/vsiaz/` and `/vsiadls/` support only sequential writes, we use the file mount feature available in the [mssparkutils package](#). After the output is written to a mount location, copy it to Azure Data Lake Storage as shown in this sample transformation:

Python

```

import shutil
import sys
from osgeo import gdal
from notebookutils import mssparkutils

mssparkutils.fs.mount(
    "abfss://<container_name>@<storage_account_name>.dfs.core.windows.net",
    "/<mount_path>",
    {"linkedService": "<linked_service_name>"}
)

access_key = TokenLibrary.getSecret('<key-vault-name>', '<secret-name>')
gdal.SetConfigOption('AZURE_STORAGE_ACCOUNT', '<storage_account_name>')
gdal.SetConfigOption('AZURE_STORAGE_ACCESS_KEY', access_key)

options = gdal.WarpOptions(options=['tr'], xRes=1000, yRes=1000)
gdal.Warp('dst_img.tiff',
    '/vsiadls/<container_name>/path/to/src_img.tiff', options=options)

jobId = mssparkutils.env.getJobId()

shutil.copy("dst_img.tiff",
f"/synfs/{jobId}/<mount_path>/path/to/dst_img.tiff")

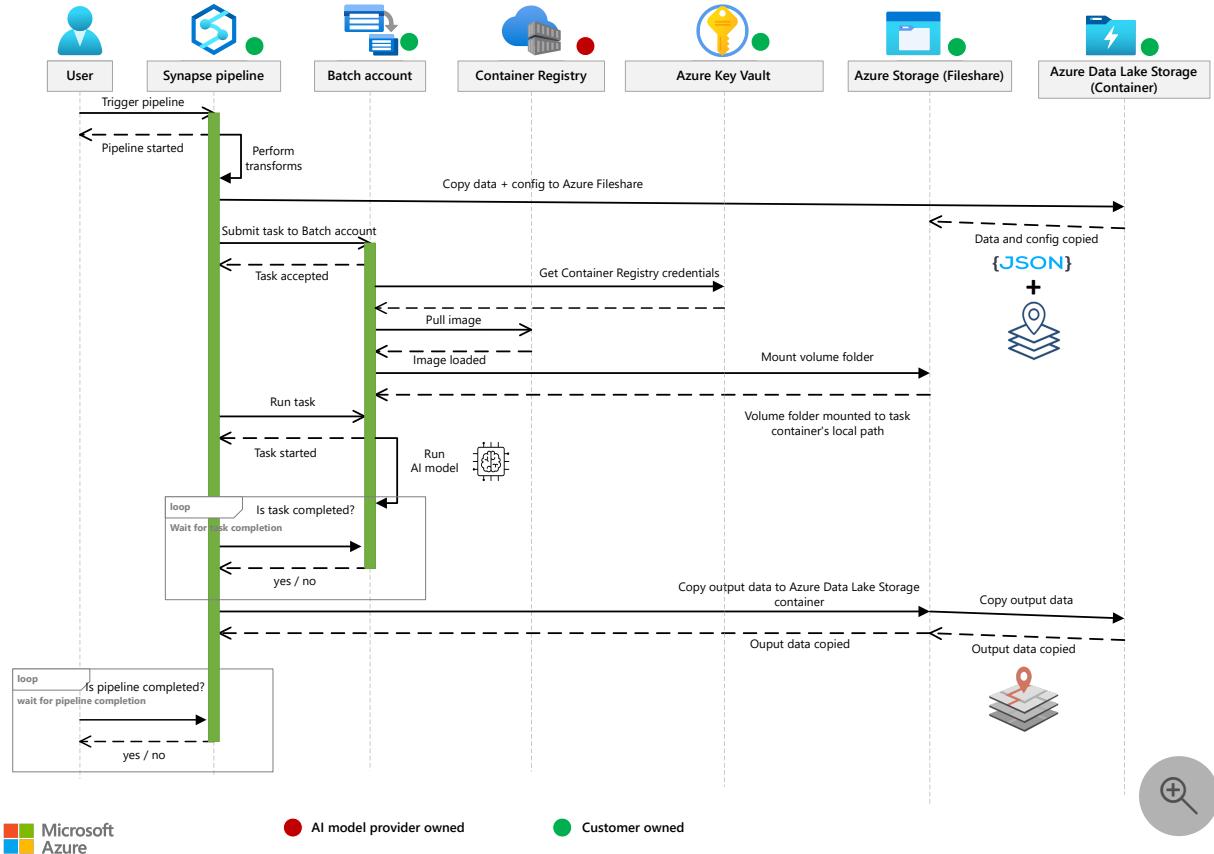
```

In Azure Synapse, you can add Azure Data Lake Storage as one of the linked services. For instructions, see [Linked services](#).

Sample solution

An implementation of this architecture is available on [GitHub](#).

This diagram shows the steps in the sample solution:



Download a [Visio file](#) of this architecture.

ⓘ Note

The data is pulled from spaceborne data sources and copied to Azure Data Lake Storage. The data ingestion isn't part of the reference implementation.

1. An Azure Synapse pipeline reads the spaceborne data from Azure Data Lake Storage.
2. The data is processed with the GDAL library in an Azure Synapse notebook.
3. The processed data is stored in Azure Data Lake Storage.
4. The processed data is read from Azure Data Lake Storage and passed to object detection Custom Vision AI models by an Azure Synapse Custom activity. The Custom activity uses Azure Batch pools to run the object detection model.
5. The object detection model outputs a list of detected objects and bounding boxes.

6. The detected objects are converted to GeoJSON and stored in Azure Data Lake Storage.
7. The GeoJSON data is read from Azure Data Lake Storage and stored in a PostgreSQL database.
8. The data is read from the PostgreSQL database. It can be visualized further in tools like ArcGIS Pro, QGIS, and Power BI.

Install geospatial packages in an Azure Synapse Spark pool

You need to install the packages in an Azure Synapse Spark pool by using the package management feature. For more information, see [Azure Synapse package management](#).

To support spaceborne data workloads on Azure Synapse, you need libraries like [GDAL](#), [Rasterio](#), and [GeoPandas](#). You can install these libraries on a serverless Apache Spark pool by using a YAML file. [Anaconda](#) libraries are pre-installed on the Spark pool.

Prerequisites

- [Create an Azure Synapse workspace](#).
- [Create the Spark pool in Azure Synapse Studio](#).

Instructions

1. The following libraries and packages are available in the [environment.yml](#) file. We recommend using this file to install the libraries in the Spark pools. If you copy the below content, make sure there are no tabs, as YAML only allows spaces as indentation.

YAML

```
name: aoi-env
channels:
- conda-forge
- defaults
dependencies:
- azure-storage-file-datalake
- gdal=3.3.0
- libgdal
- pip>=20.1.1
- pyproj
- shapely
- pip:
```

- [rasterio](#)
- [geopandas](#)

! **Note**

GDAL uses virtual file system [/vsiadls/](#) for Azure Data Lake Storage. This functionality is available starting in [GDAL v3.3.0](#). Be sure to use version 3.3.0 or later.

2. Go to <https://web.azuresynapse.net> and sign in to your workspace.
3. Select **Manage** in the navigation pane and then select **Apache Spark pools**.
4. Select **Packages** by selecting the ellipsis button (...) on the Spark pool. Upload the environment.yml file from local and apply the package settings.
5. The notification section of the portal notifies you when the installation is complete. You can also track installation progress by taking these steps:
 - a. Go to the Spark applications list on the **Monitor** tab.
 - b. Select the **SystemReservedJob-LibraryManagement** link that corresponds to your pool update.
 - c. View the driver logs.
6. Run the following code to verify that the correct versions of the libraries installed. The pre-installed libraries that Conda installs will also be listed.

Python

```
import pkg_resources
for d in pkg_resources.working_set:
    print(d)
```

For more information, see [Manage packages](#).

Contributors

This article is being updated and maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Kungumaraj Nachimuthu](#) | Senior Software Engineer
- [Karthick Narendran](#) | Senior Software Engineer

Additional contributors:

- [Mick Alberts](#) | Technical Writer
- [Taylor Corbett](#) | Senior Data Scientist
- [Tushar Dhadiwal](#) | Senior Software Engineer
- [Mandar Inamdar](#) | Principal Engineering Manager
- [Sushil Kumar](#) | Senior Software Engineer
- [Nikhil Manchanda](#) | Principal Engineering Manager
- [Safiyah Sadiq](#) | Software Engineer II
- [Xiaoyuan Yang](#) | Principal Data Science Manager
- [Tai Yee](#) | Senior Program Manager

Next steps

- [Getting geospatial insights from big data using SynapseML](#)
- [Get started with Azure Synapse Analytics](#)
- [Explore Azure Synapse Studio](#)
- [Create and consume Cognitive Services](#)

Related resources

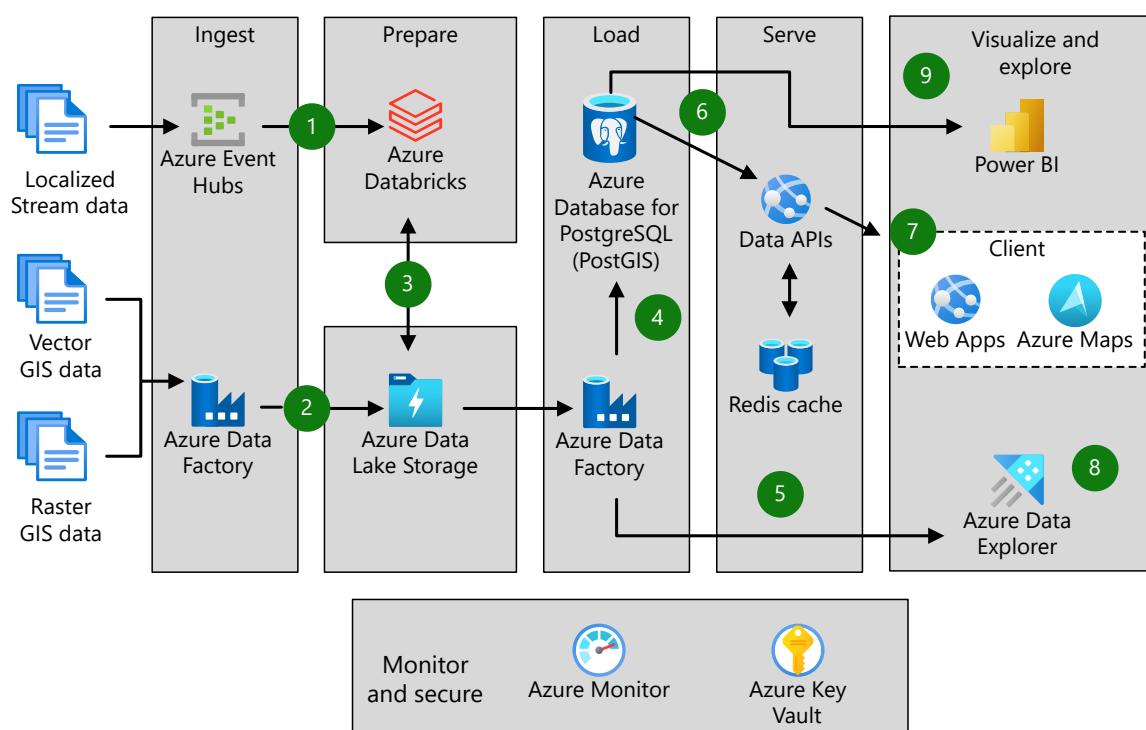
- [Geospatial data processing and analytics](#)
- [Geospatial analysis for the telecommunications industry](#)
- [Big data architectures](#)
- [End-to-end computer vision at the edge for manufacturing](#)

Geospatial data processing and analytics

Azure Data Factory Azure Data Lake Storage Azure Database for PostgreSQL Azure Databricks
Azure Event Hubs

This article outlines a manageable solution for making large volumes of geospatial data available for analytics.

Architecture



Download a [Visio file](#) of this architecture.

The diagram contains several gray boxes, each with a different label. From left to right, the labels are Ingest, Prepare, Load, Serve, and Visualize and explore. A final box underneath the others has the label Monitor and secure. Each box contains icons that represent various Azure services. Numbered arrows connect the boxes in the way that the steps describe in the diagram explanation.

Workflow

1. IoT data enters the system:
 - [Azure Event Hubs](#) ingests streams of IoT data. The data contains coordinates or other information that identifies locations of devices.
 - Event Hubs uses [Azure Databricks](#) for initial stream processing.
 - Event Hubs stores the data in [Azure Data Lake Storage](#).
2. GIS data enters the system:
 - [Azure Data Factory](#) ingests raster GIS data and vector GIS data of any format.
 - Raster data consists of grids of values. Each pixel value represents a characteristic like the temperature or elevation of a geographic area.
 - Vector data represents specific geographic features. Vertices, or discrete geometric locations, make up the vectors and define the shape of each spatial object.
 - Data Factory stores the data in Data Lake Storage.
3. Spark clusters in Azure Databricks use geospatial code libraries to transform and normalize the data.
4. Data Factory loads the prepared vector and raster data into [Azure Database for PostgreSQL](#). The solution uses the PostGIS extension with this database.
5. Data Factory loads the prepared vector and raster data into [Azure Data Explorer](#).
6. Azure Database for PostgreSQL stores the GIS data. APIs make this data available in standardized formats:
 - [GeoJSON](#) is based on JavaScript Object Notation (JSON). GeoJSON represents simple geographical features and their non-spatial properties.
 - [Well-known text \(WKT\)](#) is a text markup language that represents vector geometry objects.
 - [Vector tiles](#) are packets of geographic data. Their lightweight format improves mapping performance.
- A Redis cache improves performance by providing quick access to the data.
7. The Web Apps feature of [Azure App Service](#) works with Azure Maps to create visuals of the data.
8. Users analyze the data with Azure Data Explorer. GIS features of this tool create insightful visualizations. Examples include creating scatterplots from geospatial

data.

9. [Power BI](#) provides customized reports and business intelligence (BI). The Azure Maps visual for Power BI highlights the role of location data in business results.

Throughout the process:

- [Azure Monitor](#) collects information on events and performance.
- Log Analytics runs queries on Monitor logs and analyzes the results.
- [Azure Key Vault](#) secures passwords, connection strings, and secrets.

Components

- [Azure Event Hubs](#) is a fully managed streaming platform for big data. This platform as a service (PaaS) offers a partitioned consumer model. Multiple applications can use this model to process the data stream at the same time.
- [Azure Data Factory](#) is an integration service that works with data from disparate data stores. You can use this fully managed, serverless platform to create, schedule, and orchestrate data transformation workflows.
- [Azure Databricks](#) is a data analytics platform. Its fully managed Spark clusters process large streams of data from multiple sources. Azure Databricks can transform geospatial data at large scale for use in analytics and data visualization.
- [Data Lake Storage](#) is a scalable and secure data lake for high-performance analytics workloads. This service can manage multiple petabytes of information while sustaining hundreds of gigabits of throughput. The data typically comes from multiple, heterogeneous sources and can be structured, semi-structured, or unstructured.
- [Azure Database for PostgreSQL](#) is a fully managed relational database service that's based on the community edition of the open-source [PostgreSQL](#) database engine.
- [PostGIS](#) is an extension for the PostgreSQL database that integrates with GIS servers. PostGIS can run SQL location queries that involve geographic objects.
- [Redis](#) is an open-source, in-memory data store. Redis caches keep frequently accessed data in server memory. The caches can then quickly process large volumes of application requests that use the data.
- [Power BI](#) is a collection of software services and apps. You can use Power BI to connect unrelated sources of data and create visuals of them.

- The [Azure Maps visual for Power BI](#) provides a way to enhance maps with spatial data. You can use this visual to show how location data affects business metrics.
- [Azure App Service](#) and its [Web Apps](#) feature provide a framework for building, deploying, and scaling web apps. The App Service platform offers built-in infrastructure maintenance, security patching, and scaling.
- [GIS data APIs in Azure Maps](#) store and retrieve map data in formats like GeoJSON and vector tiles.
- [Azure Data Explorer](#) is a fast, fully managed data analytics service that can work with [large volumes of data](#). This service originally focused on time series and log analytics. It now also handles diverse data streams from applications, websites, IoT devices, and other sources. [Geospatial functionality](#) in Azure Data Explorer provides options for rendering map data.
- [Azure Monitor](#) collects data on environments and Azure resources. This diagnostic information is helpful for maintaining availability and performance. Two data platforms make up Monitor:
 - [Azure Monitor Logs](#) records and stores log and performance data.
 - [Azure Monitor Metrics](#) collects numerical values at regular intervals.
- [Log Analytics](#) is an Azure portal tool that runs queries on Monitor log data. Log Analytics also provides features for charting and statistically analyzing query results.
- [Key Vault](#) stores and controls access to secrets such as tokens, passwords, and API keys. Key Vault also creates and controls encryption keys and manages security certificates.

Alternatives

- Instead of developing your own APIs, consider using [Martin](#). This open-source tile server makes vector tiles available to web apps. Written in [Rust](#), Martin connects to PostgreSQL tables. You can deploy it as a container.
- If your goal is to provide a standardized interface for GIS data, consider using [GeoServer](#). This open framework implements industry-standard [Open Geospatial Consortium \(OGC\)](#) protocols such as [Web Feature Service \(WFS\)](#). It also integrates with common spatial data sources. You can deploy GeoServer as a container on a virtual machine. When customized web apps and exploratory queries are secondary, GeoServer provides a straightforward way to publish geospatial data.

- Various Spark libraries are available for working with geospatial data on Azure Databricks. This solution uses these libraries:
 - [Apache Sedona \(GeoSpark\)](#)
 - [GeoPandas](#)
- But [other solutions also exist for processing and scaling geospatial workloads with Azure Databricks](#).
- [Vector tiles](#) provide an efficient way to display GIS data on maps. This solution uses PostGIS to dynamically query vector tiles. This approach works well for simple queries and result sets that contain well under 1 million records. But in the following cases, a different approach may be better:
 - Your queries are computationally expensive.
 - Your data doesn't change frequently.
 - You're displaying large data sets.

In these situations, consider using [Tippecanoe](#) to generate vector tiles. You can run Tippecanoe as part of your data processing flow, either as a container or with [Azure Functions](#). You can make the resulting tiles available through APIs.

- Like Event Hubs, [Azure IoT Hub](#) can ingest large amounts of data. But IoT Hub also offers bi-directional communication capabilities with devices. If you receive data directly from devices but also send commands and policies back to devices, consider IoT Hub instead of Event Hubs.
- To streamline the solution, omit these components:
 - Azure Data Explorer
 - Power BI

Scenario details

Many possibilities exist for working with *geospatial data*, or information that includes a geographic component. For instance, geographic information system (GIS) software and standards are widely available. These technologies can store, process, and provide access to geospatial data. But it's often hard to configure and maintain systems that work with geospatial data. You also need expert knowledge to integrate those systems with other systems.

This article outlines a manageable solution for making large volumes of geospatial data available for analytics. The approach is based on [Advanced Analytics Reference Architecture](#) and uses these Azure services:

- Azure Databricks with GIS Spark libraries processes data.

- Azure Database for PostgreSQL queries data that users request through APIs.
- Azure Data Explorer runs fast exploratory queries.
- Azure Maps creates visuals of geospatial data in web applications.
- The Azure Maps Power BI visual feature of Power BI provides customized reports

Potential use cases

This solution applies to many areas:

- Processing, storing, and providing access to large amounts of raster data, such as maps or climate data.
- Identifying the geographic position of enterprise resource planning (ERP) system entities.
- Combining entity location data with GIS reference data.
- Storing Internet of Things (IoT) telemetry from moving devices.
- Running analytical geospatial queries.
- Embedding curated and contextualized geospatial data in web apps.

Considerations

The following considerations, based on the [Microsoft Azure Well-Architected Framework](#), apply to this solution.

Availability

- [Event Hubs spreads failure risk across clusters.](#)
 - Use a namespace with availability zones turned on to spread risk across three physically separated facilities.
 - Consider using the geo-disaster recovery feature of Event Hubs. This feature replicates the entire configuration of a namespace from a primary to a secondary namespace.
- See [business continuity features that Azure Database for PostgreSQL offers](#). These features cover a range of recovery objectives.
- [App Service diagnostics](#) alerts you to problems in apps, such as downtime. Use this service to identify, troubleshoot, and resolve issues like outages.
- Consider using [App Service to back up application files](#). But be careful with backed-up files, which include app settings in plain text. Those settings can contain secrets like connection strings.

Scalability

This solution's implementation meets these conditions:

- Processes up to 10 million data sets per day. The data sets include batch or streaming events.
- Stores 100 million data sets in an Azure Database for PostgreSQL database.
- Queries 1 million or fewer data sets at the same time. A maximum of 30 users run the queries.

The environment uses this configuration:

- An Azure Databricks cluster with four F8s_V2 worker nodes.
- A memory-optimized instance of Azure Database for PostgreSQL.
- An App Service plan with two Standard S2 instances.

Consider these factors to determine which adjustments to make for your implementation:

- Your data ingestion rate.
- Your volume of data.
- Your query volume.
- The number of parallel queries you need to support.

You can scale Azure components independently:

- Event Hubs automatically scales up to meet usage needs. But take steps to [manage throughput units](#) and [optimize partitions](#).
- Data Factory handles large amounts of data. Its [serverless architecture supports parallelism at different levels](#).
- [Data Lake Storage is scalable by design](#).
- Azure Database for PostgreSQL offers [high-performance horizontal scaling](#).
- [Azure Databricks clusters resize as needed](#).
- [Azure Data Explorer can elastically scale to terabytes of data in minutes](#).
- [App Service web apps scale up and out](#).

The [autoscale feature of Monitor](#) also provides scaling functionality. You can configure this feature to add resources to handle increases in load. It can also remove resources to save money.

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](#).

- Protect vector tile data. Vector tiles embed coordinates and attributes for multiple entities in one file. If you generate vector tiles, use a dedicated set of tiles for each permission level in your access control system. With this approach, only users within each permission level have access to that level's data file.
- To improve security, use Key Vault in these situations:
 - [To manage keys that Event Hubs uses to encrypt data](#).
 - [To store credentials that Data Factory uses in pipelines](#).
 - [To secure application settings and secrets that your App Service web app uses](#).
- See [Security in Azure App Service](#) for information on how App Service helps secure web apps. Also consider these points:
 - See how to [get the certificate that your app needs if it uses a custom domain name](#).
 - See how to [redirect HTTP requests for your app to the HTTPS port](#).
 - Learn about [best practices for authentication in web apps](#).

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

- To estimate the cost of implementing this solution, see a sample [cost profile](#). This profile is for a single implementation of the environment described in [Scalability considerations](#). It doesn't include the cost of Azure Data Explorer.
- To adjust the parameters and explore the cost of running this solution in your environment, use the [Azure pricing calculator](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Richard Bumann](#) | Solution Architect

Next steps

Product documentation:

- [About Azure Event Hubs](#)
- [Azure Databricks concepts](#)
- [Introduction to Azure Data Lake Storage](#)
- [What is Azure Data Factory?](#)
- [Azure App Service overview](#)

To start implementing this solution, see this information:

- [Connect a WFS to Azure Maps](#)
- [Process OpenStreetMap data ↗ with Spark.](#)
- [Explore ways to display data with Azure Maps.](#)

Information on processing geospatial data

- [Functions for querying PostGIS for vector tiles ↗](#)
- [Functions for loading PostGIS rasters ↗](#)
- [Azure Data Explorer geospatial functions](#)
- [Data sources for vector tiles in Azure Maps](#)
- [Approaches for processing geospatial data in Databricks ↗](#)

Related resources

Related architectures

- [Big data analytics with Azure Data Explorer](#)
- [Health data consortium on Azure](#)
- [\[DataOps for the modern data warehouse\]\[DataOps for the modern data warehouse\]](#)
- [Azure Data Explorer interactive analytics](#)
- [Geospatial reference architecture - Azure Orbital](#)
- [Geospatial analysis for telecom](#)
- [Spaceborne data analysis with Azure Synapse Analytics](#)

Related guides

- [Compare the machine learning products and technologies from Microsoft - Azure Databricks](#)

- Machine learning operations (MLOps) framework to scale up machine learning lifecycle with Azure Machine Learning
- [Azure Machine Learning decision guide for optimal tool selection][Azure Machine Learning decision guide for optimal tool selection]
- Monitor Azure Databricks

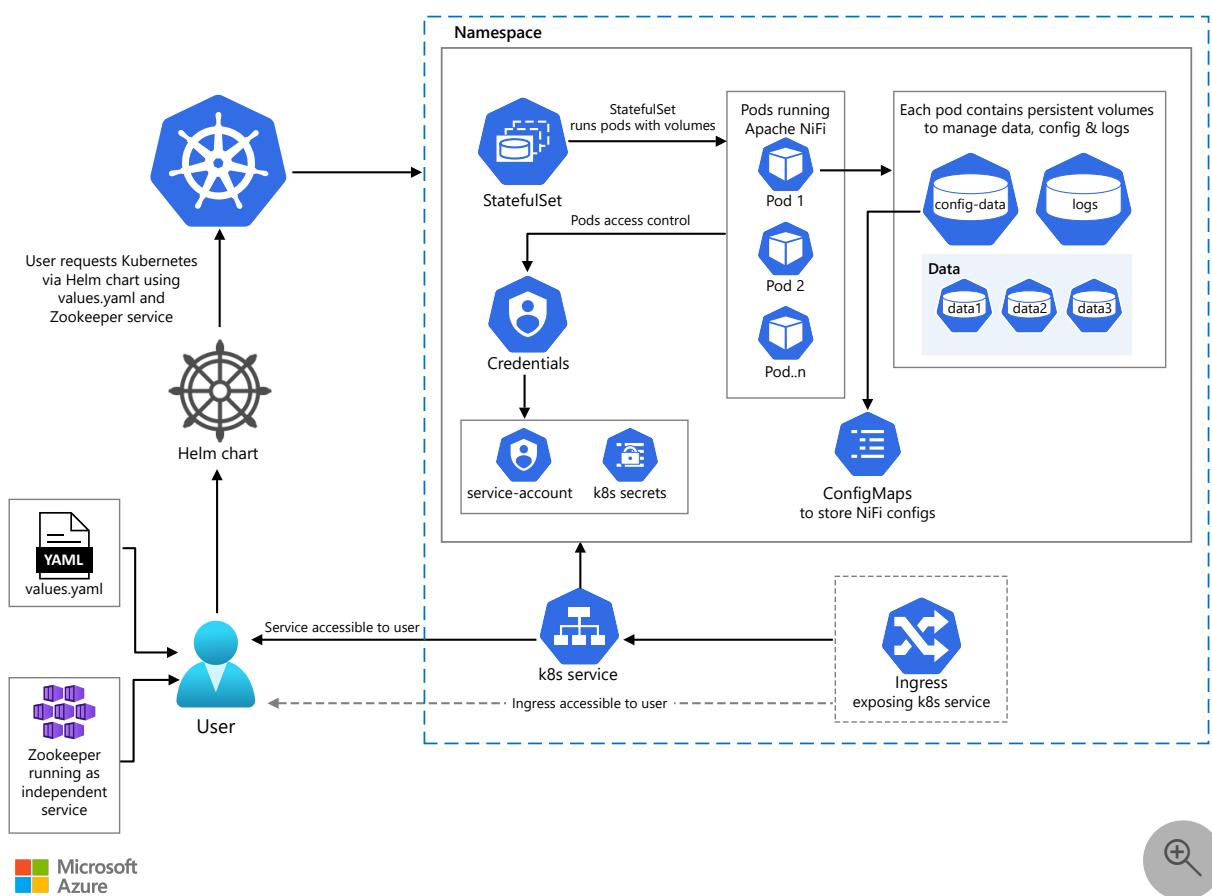
Helm-based deployments for Apache NiFi

Azure Kubernetes Service (AKS)

This solution shows you how to use Helm charts when you deploy NiFi on Azure Kubernetes Service (AKS). Helm streamlines the process of installing and managing Kubernetes applications.

Apache®, Apache NiFi®, and NiFi® 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.

Workflow

- A Helm chart contains a `values.yaml` file. That file lists input values that users can edit.
- A user adjusts settings in a chart, including values for:
 - Volume sizes.
 - The number of pods.
 - User authentication and authorization mechanisms.
- The user runs the Helm `install` command to deploy the chart.
- Helm checks whether the user input contains values for all required variables.
- Helm creates a manifest that describes the objects to deploy on Kubernetes.
- Helm sends the manifest to the Kubernetes cluster. Apache ZooKeeper provides cluster coordination.
- Kubernetes creates the specified objects. A NiFi deployment requires these objects:
 - Configuration objects.
 - Data volumes. Pod storage is temporary.
 - A log volume.
 - Pods that use an image to run NiFi in a container. Kubernetes uses a *StatefulSet* workload resource to manage the pods.
 - A Kubernetes service that makes the NiFi UI available to users.
 - Ingress routes if the cluster uses ingress to make the UI available externally.

Components

A Helm chart is a collection of files in a folder with a tree structure. These files describe Kubernetes resources. You can configure the following components in a Helm chart:

ZooKeeper

ZooKeeper uses a separate chart. You can use the standard ZooKeeper chart that Kubernetes supplies in its [incubator chart repository](#). But when your dependencies include public registry content, you introduce risk into your image development and deployment workflows. To mitigate this risk, keep local copies of public content when you can. For detailed information, see [Manage public content with Azure Container Registry](#).

As an alternative, you can deploy ZooKeeper on your own. If you choose this option, provide the ZooKeeper server and port number so that the pods that run NiFi can access the ZooKeeper service.

Kubernetes StatefulSet

To run an application on Kubernetes, you run a pod. This basic unit runs different containers that implement the application's different activities.

Kubernetes offers two solutions for managing pods that run an application like NiFi:

- A *ReplicaSet*, which maintains a stable set of the replica pods that run at any given time. You often use a ReplicaSet to guarantee the availability of a specified number of identical pods.
- A *StatefulSet*, which is the workload API object that you use to manage stateful applications. A StatefulSet manages pods that are based on an identical container specification. Kubernetes creates these pods from the same specification. But these pods aren't interchangeable. Each pod has a persistent identifier that it maintains across rescheduling.

Since you use NiFi to manage data, a StatefulSet provides the best solution for NiFi deployments.

ConfigMaps

Kubernetes offers *ConfigMaps* for storing non-confidential data. Kubernetes uses these objects to manage various configuration files like `nifi.properties`. The container that runs the application accesses the configuration information through mounted volumes and files. ConfigMaps make it easy to manage post-deployment configuration changes.

ServiceAccount

In secured instances, NiFi uses authentication and authorization. NiFi manages this information in file system files. Specifically, each cluster node needs to maintain an `authorizations.xml` file and a `users.xml` file. All members need to be able to write to these files. And each node in the cluster needs to have an identical copy of this information. Otherwise, the cluster goes out of sync and breaks down.

To meet these conditions, you can copy these files from the first member of the cluster to every member that comes into existence. Each new member then maintains its own copies. Pods generally don't have authorization to copy content from another pod. But a Kubernetes *ServiceAccount* provides a way to get authorization.

Services

Kubernetes services make the application service available to users of the Kubernetes cluster. Service objects also make it possible for member nodes of NiFi clusters to communicate with each other. For Helm chart deployments, use two service types: headless services and IP-based services.

Ingress

An ingress manages external access to cluster services. Specifically, a pre-configured ingress controller exposes HTTP and HTTPS routes from outside the cluster to services within the cluster. You can define ingress rules that determine how the controller routes the traffic. The Helm chart includes the ingress route in the configuration.

Secrets

To configure secured NiFi clusters, you need to store credentials. Kubernetes secrets provide a secure way to store and retrieve these credentials.

Scenario details

[Apache NiFi](#) users often need to deploy NiFi on Kubernetes. A Kubernetes deployment involves many objects, such as pods, volumes, and services. It's difficult to manage the *manifests*, or specification files, that Kubernetes uses for this number of objects. The difficulty increases when you deploy several NiFi clusters that use different configurations.

Helm *charts* provide a solution for managing the manifests. Helm is the package manager for Kubernetes. By using the Helm tool, you can streamline the process of installing and managing Kubernetes applications.

A chart is the packaging format that Helm uses. You enter configuration requirements into chart files. Helm keeps track of each chart's history and versions. Helm then uses charts to generate Kubernetes manifest files.

From a single chart, you can deploy applications that use different configurations. When you run [NiFi on Azure](#), you can use Helm charts to deploy different NiFi configurations on Kubernetes.

Apache®, Apache NiFi®, and NiFi® 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.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Data disks

For disk usage, consider using a striped set of disks for repositories. In test deployments that used Virtual Machine Scale Sets, this approach worked best. The following excerpt from `nifi.properties` shows a disk usage configuration:

```
config

nifi.flowfile.repository.directory=/data/partition1/flowfiles
nifi.provenance.repository.directory.stripe1=/data/partition1/provenancenifi
.niifi.provenance.repository.directory.stripe2=/data/partition2/provenancenifi.pro
venance.repository.directory.stripe3=/data/partition3/provenancenifi.content
.repository.directory.stripe2=/data/partition2/content
nifi.content.repository.directory.stripe3=/data/partition3/content
```

This configuration uses three volumes of equal size. You can adjust the values and the striping to meet your system requirements.

Deployment scenarios

You can use a public or private load balancer or an ingress controller to expose a NiFi cluster. When you use Helm charts for this implementation, two configurations are available:

- An unsecured NiFi cluster that's accessible through an HTTP URL without user authentication or authorization.
- A secured NiFi cluster that's accessible through an HTTPS URL. This kind of cluster is secured with TLS. When you configure secured clusters, you can provide your own certificates. Alternatively, the charts can generate the certificates. For this purpose, the charts use a NiFi toolkit that provides a self-signed Certificate Authority (CA).

If you configure a NiFi cluster to run as a secured cluster with TLS communication, you need to turn on user authentication. Use one of the following supported user authentication methods:

- Certificate-based user authentication. Users are authenticated by the certificate that they present to the NiFi UI. To use this kind of user authentication system, add the CA's public certificate to the NiFi deployment.
- LDAP-based user authentication. An LDAP server authenticates user credentials. When you deploy the chart, provide information about the LDAP server and the information tree.
- OpenID-based user authentication. Users provide information to the OpenID server to configure the deployment.

Resource configuration and usage

To optimize resource usage, use these Helm options to configure CPU and memory values:

- The `request` option, which specifies the initial amount of the resource that the container requests
- The `limit` option, which specifies the maximum amount of the resource that the container can use

When you configure NiFi, consider your system's memory configuration. Because NiFi is a Java application, you should adjust settings like the minimum and maximum java virtual machine (JVM) memory values. Use the following settings:

- `jvmMinMemory`
- `jvmMaxMemory`
- `g1ReservePercent`
- `conGcThreads`
- `parallelGcThreads`
- `initiatingHeapOccupancyPercent`

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](#).

Use a Kubernetes security context to improve the security of the underlying containers that run the NiFi binary. A security context manages access to those containers and their pods. Through a security context, you can grant non-root users permissions to run the containers.

Other uses of security contexts include:

- Restricting the access of OS-based users that run the containers.
- Specifying which groups can access the containers.
- Limiting access to the file system.

Container images

Kubernetes containers are the basic units that run NiFi binaries. To configure a NiFi cluster, focus on the image that you use to run these containers. You have two options for this image:

- Use the standard NiFi image to run the NiFi chart. The Apache NiFi community supplies that image. But you need to add a `kubectl` binary to the containers to configure secured clusters.
- Use a custom image. If you take this approach, consider your file system requirements. Ensure that the location of your NiFi binaries is correct. For more information on the configured file system, see [Dockerfile in the Apache NiFi source code](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Muazma Zahid](#) | Principal PM Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Helm](#)
- [Helm charts](#)
- [Kubernetes](#)
- [Kubernetes StatefulSets](#)
- [Kubernetes Volumes](#)
- [Kubernetes ConfigMaps](#)
- [Kubernetes Secrets](#)
- [Kubernetes Service](#)
- [Kubernetes Ingress](#)
- [Azure Kubernetes Service](#)

- [Apache NiFi Docker Image ↗](#)

Related resources

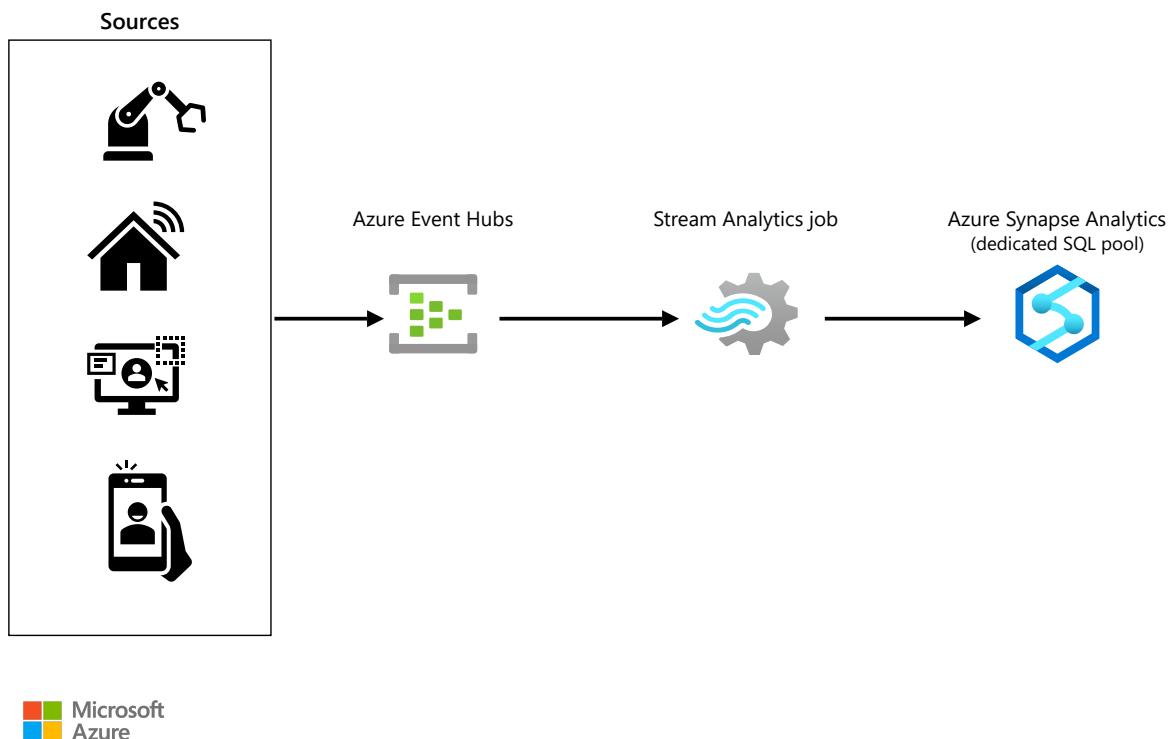
- [Apache NiFi on Azure](#)
- [Apache NiFi monitoring with MonitoFi](#)
- [Microservices architecture on Azure Kubernetes Service \(AKS\)](#)
- [Advanced Azure Kubernetes Service \(AKS\) microservices architecture](#)

High throughput stream ingestion to Azure Synapse

Azure Synapse Analytics Azure Stream Analytics

This solution uses Azure Stream Analytics to ingest data to an Azure Synapse dedicated SQL pool in high throughput scenarios.

Architecture



Download a [Visio file](#) of this architecture.

The architecture shows the components of the stream data ingestion pipeline. Data flows through the architecture as follows:

1. The source systems generate data and send it to an [Azure Event Hubs](#) instance. Event Hubs is a big data streaming platform and event ingestion service that can receive millions of events per second.
2. Next, a Stream Analytics job processes the data stream from the Event Hubs instance. Stream Analytics has first-class integration with Event Hubs to consume data streams.

3. The Stream Analytics job has an output configured to sink the data to a Synapse Analytics dedicated SQL pool. Apart from doing a simple passthrough of the data stream to target, the Stream Analytics job can perform standard stream processing tasks, such as JOINs, temporal aggregations, filtering, anomaly detection, and so on.
4. The Stream Analytics job writes the data in a Synapse Analytics dedicated SQL pool table.

Components

- [Azure Synapse Analytics](#) is an analytics service that combines data integration, enterprise data warehousing, and big data analytics. In this solution:
 - The [Dedicated SQL Pool](#) refers to the enterprise data warehousing features that are available in Azure Synapse Analytics. A dedicated SQL pool represents a collection of analytic resources that are provisioned when you use Synapse Analytics.
- [Azure Event Hubs](#) is a real-time data streaming platform and event ingestion service. Event Hubs can ingest data from anywhere, and it seamlessly integrates with Azure data services.
- [Azure Stream Analytics](#) is a real-time, serverless analytics service for streaming data. Stream Analytics offers rapid, elastic scalability, enterprise-grade reliability and recovery, and built-in machine learning capabilities.

Alternatives

- [Azure IoT Hub](#) could replace or complement Event Hubs. The solution you choose depends on the source of your streaming data, and whether you need cloning and bidirectional communication with the reporting devices.
- [Azure Synapse serverless Apache Spark pools](#) can replace Azure Stream Analytics by using Spark Structured Streaming. But the solution will be more complex to develop and maintain.

Scenario details

Traditional data warehouses work well when dealing with batch jobs to load data. But in today's world, we see many use cases where the customers can't wait for a batch job to complete for our data scientists, analysts, or dashboards to access the data. Today, more

customers require real-time representations of their business in their data warehouse. This requirement goes beyond traditional batch jobs and requires support from stream ingestion for their data warehouses.

Azure Synapse Analytics seamlessly supports both enterprise data warehousing and big data analytics workloads. Azure Stream Analytics is a serverless stream processing PaaS service that can scale with customer's needs. This example architecture will show how you can use an Azure Stream Analytics job to ingest stream to an Azure Synapse Analytics dedicated SQL pool.

Potential use cases

Several scenarios can benefit from this architecture:

- Ingest data from a stream to a data warehouse in near real-time.
- Apply different stream processing techniques (JOINS, temporal aggregations, filtering, anomaly detection, and so on) to transform the data. Then, store the result in the data warehouse.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Performance efficiency

Performance efficiency is the ability of your workload to scale to meet the demands placed on it by users in an efficient manner. For more information, see [Performance efficiency pillar overview](#).

To achieve high throughput, there are a few key points that need to be considered while implementing the solution:

- Apply partitioning in the Event Hubs instance to maximize raw I/O throughput and to parallelize the consumers' processing. Partitions don't cost money. The cost of Event Hubs depends on the throughput unit (TU). To avoid starving consumers, use at least as many partitions as consumers. Use more keys than partitions to avoid unbalanced partition loads. For detailed best practices on partitioning, see [Partitioning in Event Hubs and Kafka](#).

- An Azure Stream Analytics job can connect to Azure Synapse using both the [Azure Synapse Analytics](#) connector and the [SQL Database](#) connector. Use the Azure Synapse Analytics connector, because it can achieve a throughput of 200 MB/s. The maximum throughput for the SQL Database connector is 15 MB/s.
- Use hash or round robin distribution for the Synapse dedicated SQL pool table. Don't use a replicated table.
- Stream Analytics will parallelize the job, based on the number of partitions, and it creates separate connections to Synapse Analytics for each parallel process. The total number of connections is equal to the number of partitions multiplied by the number of jobs (connections = partitions x jobs). Synapse Analytics can have a maximum of 1024 connections. Partition wisely, to avoid more than 1024 connections. Otherwise, Synapse Analytics will start throttling the connections.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

- The [Azure Event Hubs](#) bill is based on tier, throughput units provisioned, and ingress traffic received.
- [Azure Stream Analytics](#) bases costs on the number of provisioned streaming units.
- [Azure Synapse Analytics](#) Dedicated SQL pool compute is separate from storage, which enables you to scale compute independently of the data in your system. You can purchase reserved capacity for your Dedicated SQL pool resource, to save up to 65 percent, compared to pay-as-you-go rates.

Contributors

This article is being updated and maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Raihan Alam](#) | Senior Software Engineer

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Tutorial: Get started with Azure Synapse Analytics](#)
- [Azure Event Hubs Quickstart - Create an event hub using the Azure portal](#)
- [Quickstart - Create a Stream Analytics job by using the Azure portal](#)

Related resources

- [Analytics architecture design](#)
- [Choose an analytical data store in Azure](#)
- [Analytics end-to-end with Azure Synapse](#)
- [Big data analytics with enterprise-grade security using Azure Synapse](#)
- [Stream processing with Azure Stream Analytics](#)

Ingest FAA SWIM content to analyze flight data

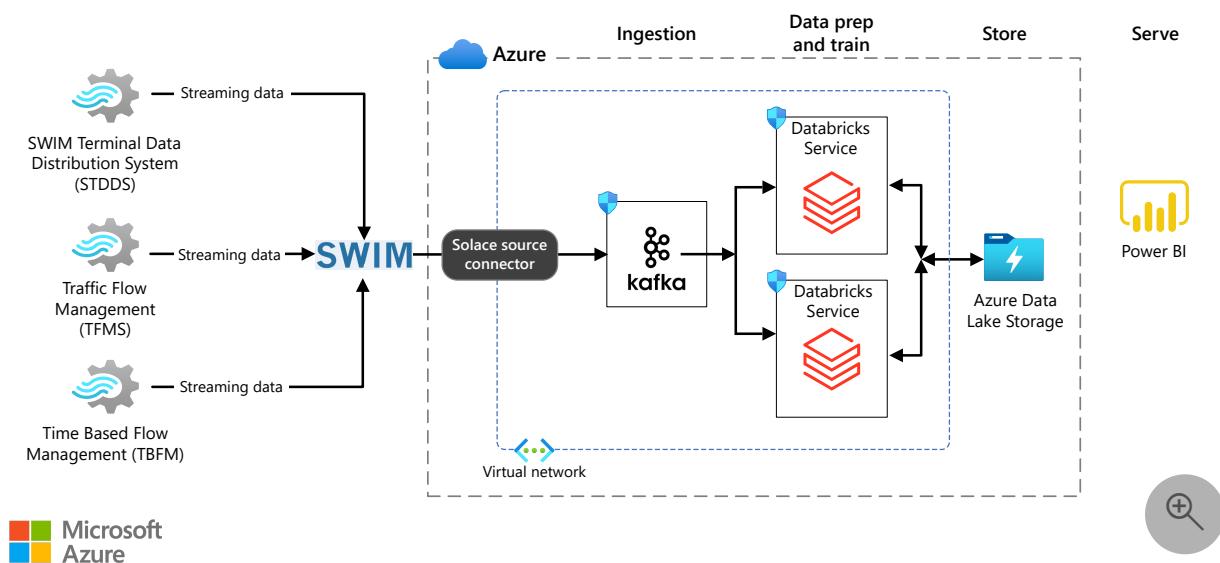
Azure Databricks Azure Data Lake Storage Power BI

This solution shows how to integrate Chef Infra, Chef InSpec, Test Kitchen, Terraform, Terraform Cloud, and GitHub Actions to automate and to create data analytics environments. It uses an Azure Databricks cluster to analyze the data.

Architecture

Apache® and Apache Kafka® 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.

Progress Chef and HashiCorp Terraform are trademarks of their respective companies. No endorsement is implied by the use of these marks.



Download a [Visio file](#) of this architecture.

The left side of the diagram shows the Federal Aviation Administration (FAA) System Wide Information Management (SWIM) system and three of its data producers. The right side shows the Azure architecture that connects, ingests, and analyzes data coming from these sources.

This solution uses Apache Kafka because of its event-driven architecture. Apache Kafka is an open-source platform that lets you collect, process, store, and integrate data from

various sources in real-time. It's based on the concept of a commit log, which is a record of events that happen in a system. You can use Kafka to publish or subscribe to streams of events, and also to transform and analyze them by using the [Streams API](#) or other external tools like Databricks.

Another important aspect of this architecture is that SWIM uses Solace, so Kafka uses a Solace source connector to connect and ingest the data. Solace provides support for open source and sink connectors that you can build and deploy in your Kafka cluster.

Workflow

1. SWIM provides data. The architecture diagram shows three of the most common data sources (TFMS, TBFM, and STDDS). The type of information that you want to analyze dictates the data source from SWIM that you need to subscribe to.
2. A Solace source connector connects and ingests the data into Kafka.
3. Messages from Kafka are cleaned, prepped, and parsed in a workspace in Azure Databricks. This Azure Databricks workspace is where data scientists do their work. They use notebooks written in Python, Scala, or R that contain the logic they need to parse the data or even train models based on it.
4. Azure Data Lake provides storage.
5. Power BI or Tableau displays the final data.

Components

- [Azure Databricks](#). A data analytics platform that's optimized for the Azure cloud.
- [Azure Data Lake Storage](#). A massively scalable and highly secure data lake for high-performance analytics workloads.
- [Power BI](#). An analytics and BI platform that can help you discover insights in your data.
- [SWIM](#). A National Airspace System (NAS) information system that provides publicly available FAA data to FAA-approved consumers via Solace Java Message Service (JMS) messaging.
- [Apache Kafka](#). An Apache event streaming platform.

Alternatives

For ingestion, the solution uses Apache Kafka in a single VM. This design configuration creates a single point of failure, but for a more robust solution, you can use one of these options:

- [Apache Kafka in Azure HDInsight](#)

- [Apache Kafka for Confluent Cloud](#)

Both are managed services and offer multiple benefits, like SLAs, simplified configuration, and scalability. They're also more expensive.

As an alternative to Power BI, you can use Tableau or another visualization option.

Scenario details

This solution relies on the Federal Aviation Administration (FAA) System Wide Information Management (SWIM) system. SWIM provides a single point of access for near real-time, relevant, and reliable aeronautical, flight, weather, and surveillance information. It delivers the infrastructure, standards, and services needed to optimize the secure exchange of relevant data across the National Airspace System (NAS) and the aviation community. As the digital data-sharing backbone of NextGen, SWIM enables both operational excellence and innovation.

This solution connects to the following FAA SWIM data sources via an Apache Kafka server:

- Traffic Flow Management Service (TFMS) distributes Traffic Flow Management (TFM) data to users via SWIM's NAS Enterprise Messaging Service (NEMS).
- Time-Based Flow Management (TBFM) enhances NAS efficiency by using the capabilities of the TBFM decision-support tool, a system already deployed at all high-altitude air traffic control centers in the contiguous United States.
- SWIM Terminal Data Distribution System (STDDS) converts legacy terminal data collected from airport towers and Terminal Radar Approach Control (TRACON) facilities into easily accessible information, which is published via NEMS.

For information about SWIM, see the [FAA SWIM page](#) .

Potential use cases

This solution consumes multiple data sources for flight data patterns. It's ideal for the aerospace, aircraft, and aviation industries. It can be used to analyze flight data patterns and to predict future flight patterns. For example, it can also be used to analyze flight data to improve flight safety.

The solution environment is flexible, so it can be extended to analyze other SWIM data sources or similar streamed data sources.

This solution also shows how to automate and create data analytics environments by using Chef Infra, Chef InSpec, Test Kitchen, Terraform, Terraform Cloud, and GitHub

Actions. This automation can be extended to other data analytics environments. For example, you can use it to automate the deployment of an Azure Databricks cluster that analyzes data from a different source.

SWIM architecture

SWIM is a NAS information system. It's an FAA cloud-based service that provides publicly available FAA SWIM content to FAA-approved consumers via Solace JMS messaging.

SWIM provides a single point of access for aviation data. Data producers publish data once, and users access the information they need through a single connection. SWIM provides multiple data producers. Depending on the type of data you need, you can subscribe to one or more of them. SWIM has a typical Publisher-Subscriber architecture.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

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](#).

A key requirement for this architecture is that all traffic must be internal and highly secure. To meet this requirement:

- Virtual network injection is used to deploy Azure Databricks. This deployment method keeps communication between the cluster and Kafka internal.
- The Azure Databricks workspace uses your Azure identity for authentication.
- Network security groups filter network traffic to and from Azure Databricks and Kafka VMs.

For more information about improving the security of your solution, see [Overview of the security pillar](#).

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

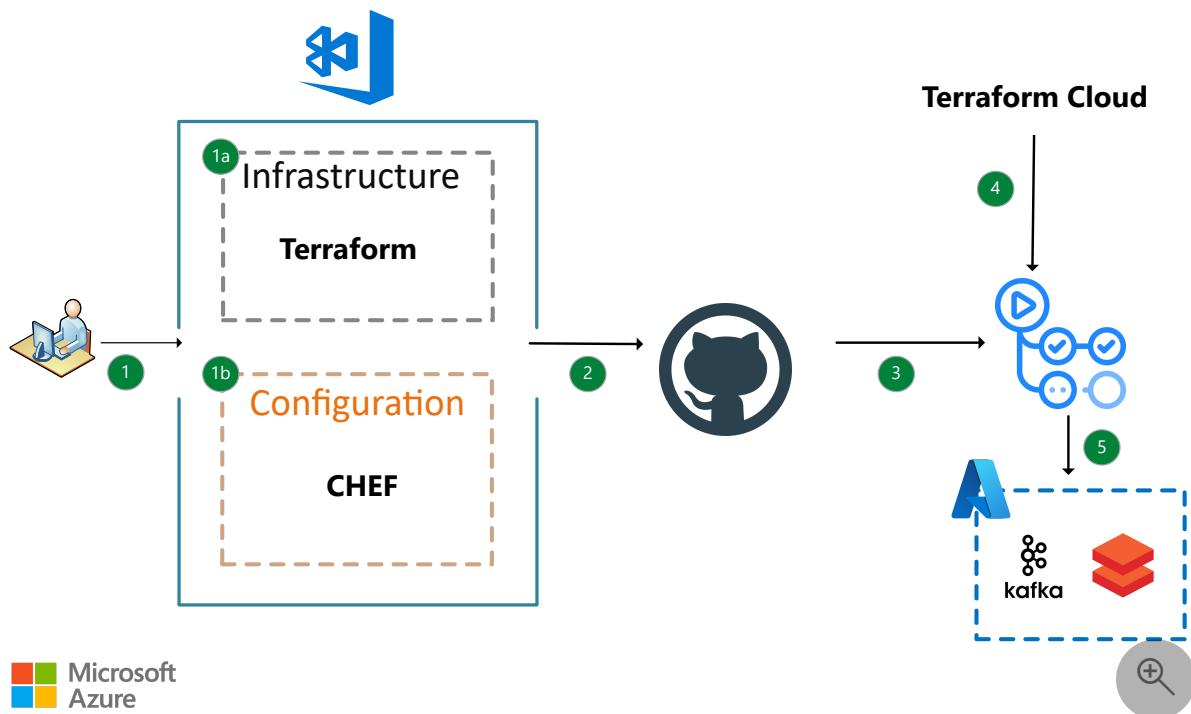
If you run this project, your account is billed. For information, see [Cost optimization documentation](#).

Operational excellence

Operational excellence covers the operations processes that deploy an application and keep it running in production. For more information, see [Overview of the operational excellence pillar](#).

CI/CD pipeline architecture

This architecture uses GitHub Actions to orchestrate the CI/CD pipeline:



[Download a Visio file](#) of this architecture.

Dataflow

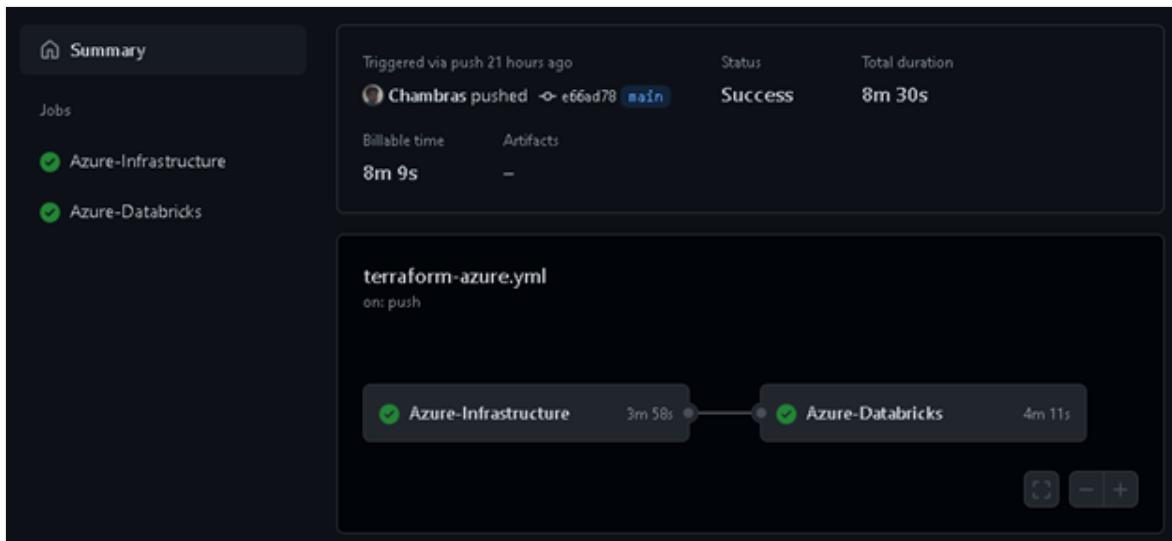
1. Developers work on either the Infrastructure or the Configuration code.
 - a. Infrastructure code uses Terraform.

- b. Configuration code uses Chef.
2. Developers push their code to the GitHub repository using a GitHub pull request.
3. The pull request triggers a GitHub Actions workflow. The workflow runs the code through a series of tests. If the changes are configuration-related, the workflow runs Chef InSpec and Test Kitchen to test the code. Once the tests are completed, it reports the results back to GitHub. If the changes are infrastructure-related, the workflow also creates a Terraform plan and posts it as a comment in the pull request. The plan shows the changes that are applied to the infrastructure if the pull request is approved.
4. If the changes are infrastructure-related, the workflow runs Terraform Cloud's remote state. Terraform Cloud is a hosted service that provides remote state management, locking, and other features.
5. If the tests pass, and the pull request is approved, the workflow merges the code into the `main` branch. After it's merged, the Actions workflow pushes changes in either the infrastructure or in the configurations for Kafka, Azure Databricks, and so on.

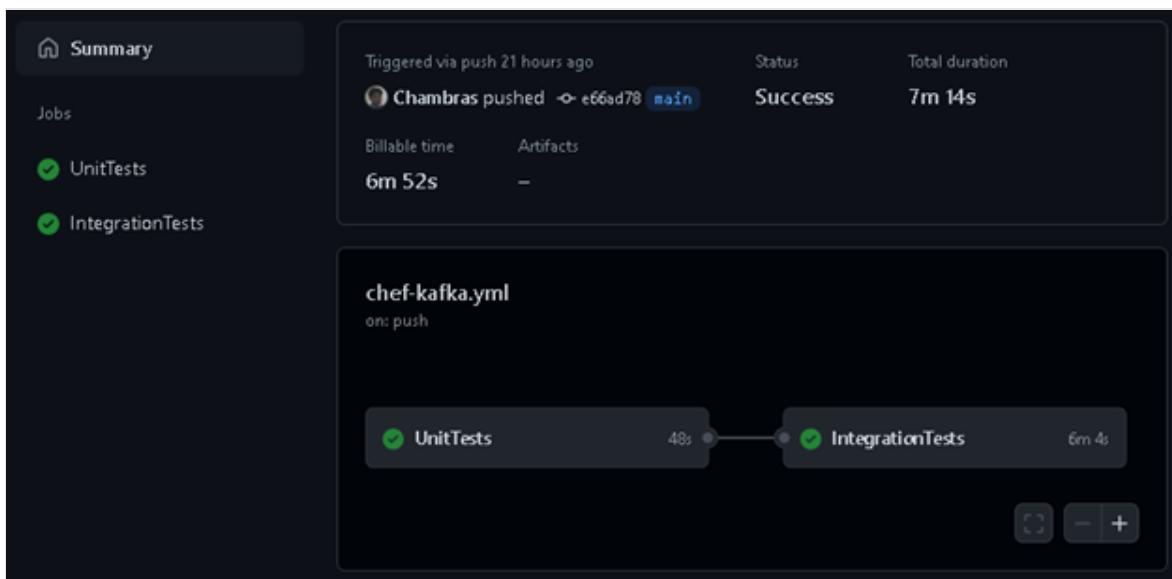
GitHub workflows

In this solution, two GitHub Actions workflows automate the infrastructure that hosts the data analytics environment. Terraform deploys the infrastructure. After provisioning is complete, Chef configures the resources that are required to connect to TFMS, TBFM, STDDS, and others if needed.

- **terraform-azure.yml** performs Terraform deployment. It uses Terraform Cloud's remote state. It also creates an Azure Databricks cluster, deploys some starter Python notebooks to test connectivity to the Kafka server, and retrieves messages. It creates all infrastructure with proper naming conventions and tagging.



- `chef-kafka.yml` performs static code analysis by using *Cookstyle*, performs unit tests by using *Chef InSpec*, and performs integration tests by using *Test Kitchen*. These tests ensure the cookbook is properly tested before it's uploaded to Chef Server.



Deploy this scenario

For information about deploying this solution, workflows, cookbooks, sample notebooks, Terraform files, and more, see the [Azure/SWIMDataIngestion](#) GitHub repository. The repository contains step-by-step instructions to deploy the solution. The deployment process is fully automated using GitHub Actions, but for it to work, it requires that some secrets are set up in GitHub.

The deployment process is divided into three parts: infrastructure, configuration, and visualization.

- The infrastructure part deploys the following Azure resources:

- A virtual network
- Subnets
- A resource group
- Kafka server
- An Azure storage account
- Azure Data Lake Storage on top of the storage account
- Network security groups
- An Azure Databricks workspace created with virtual network injection, which keeps the traffic internal
- The configuration part configures the resources that are required to connect to TFMS, TBFM, STDDS, and others if needed. To connect Kafka to SWIM, request access to SWIM and specify the data source to connect to. After it's approved, FAA sends a link to the data source endpoint, user name, password, and port for connection. Here are three of the most common data sources:
 - [STDDS](#). SWIM Terminal Data Distribution System.
 - [TFMS](#). Traffic Flow Management Service.
 - [TBFM](#). Time-Based Flow Management.
- Lastly, connect a visualization dashboard like Power BI or Tableau to Azure Databricks.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Marcelo Zambrana](#) | Senior Software Engineer

Other contributor:

- [Mick Alberts](#) | Technical Writer

Next steps

- [What is Azure Databricks?](#)
- [Introduction to Azure Data Lake Storage Gen2](#)
- [Introduction to Power BI](#)

Related resources

- All aerospace architectures
- Publisher-Subscriber pattern
- Advanced analytics architecture

Analyze news feeds with near real-time analytics using image and natural language processing

Azure Cosmos DB

Azure Functions

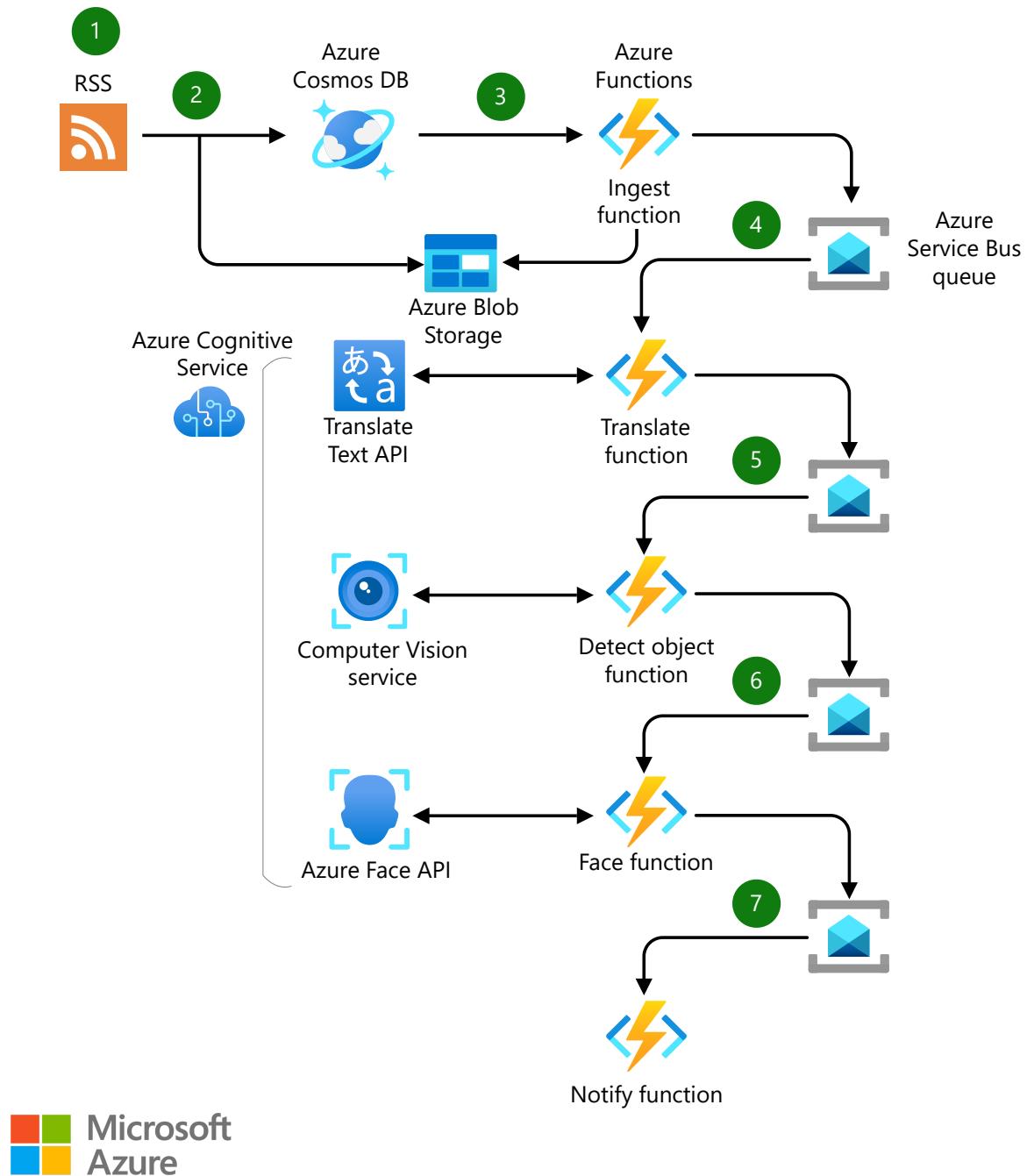
Azure Service Bus

Azure Translator Text

Azure Face API

This example scenario describes a pipeline for mass ingestion and near real-time analysis of documents coming from public RSS news feeds. It uses [Azure Cognitive Services](#) to provide useful insights based on text translation, facial recognition, and sentiment detection. Specifically, image and natural language processing steps are connected together in a messaging pipeline based on [Azure Service Bus](#). The output of the pipeline is a notification containing the insight or analysis.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

The data flows through the solution as follows:

1. An RSS news feed acts as the generator that obtains data from a document or article. For example, with an article, data typically includes a title, a summary of the original body of the news item, and sometimes images.
2. A generator or ingestion process inserts the article and any associated images into an Azure Cosmos DB [Collection](#).

3. A notification triggers an ingest function in Azure Functions that stores the article text in Azure Cosmos DB and the article images (if any) in Azure Blob Storage. The article is then passed to the next queue.
4. A translate function is triggered by the queue event. It uses the [Translate Text API](#) of Azure Cognitive Services to detect the language, translate if necessary, and collect the sentiment, key phrases, and entities from the body and the title. Then it passes the article to the next queue.
5. A detect function is triggered from the queued article. It uses the [Computer Vision](#) service to detect objects, landmarks, and written words in the associated image, then passes the article to the next queue.
6. A face function is triggered from the queued article. It uses the [Azure Face API](#) service to detect faces for gender and age in the associated image, then passes the article to the next queue.
7. When all functions are complete, the notify function is triggered. It loads the processed records for the article and scans them for any results you want. If found, the content is flagged and a notification is sent to the system of your choice.

At each processing step, the function writes the results to Azure Cosmos DB. Ultimately, the data can be used as desired. For example, you can use it to enhance business processes, locate new customers, or identify customer satisfaction issues.

Components

The following list of Azure components is used in this example.

- [Azure Storage](#) is used to hold raw image and video files associated with an article. A secondary storage account is created with Azure App Service and is used to host the Azure Function code and logs.
- [Azure Cosmos DB](#) holds article text, image, and video tracking information. The results of the Cognitive Services steps are also stored here.
- [Azure Functions](#) executes the function code used to respond to queue messages and transform the incoming content. [Azure App Service](#)  hosts the function code and processes the records serially. This scenario includes five functions: Ingest, Transform, Detect Object, Face, and Notify.
- [Azure Service Bus](#) hosts the Azure Service Bus queues used by the functions.

- [Azure Cognitive Services](#) provides the AI for the pipeline based on implementations of the [Computer Vision](#) service, [Face API](#), and [Translate Text](#) machine translation service.
- [Azure Application Insights](#) provides analytics to help you diagnose issues and to understand functionality of your application.

Alternatives

- Instead of using a pattern based on *queue notification* and Azure Functions, you could use a *topic and subscription* pattern for this data flow. [Azure Service Bus Topics](#) can be used to process the various parts of the article in parallel as opposed to the serial processing done in this example. For more information, compare [queues and topics](#).
- Use [Azure Logic Apps](#) to implement the function code and implement record-level locking such as that provided by the [Redlock algorithm](#) (which is needed for parallel processing until Azure Cosmos DB supports [partial document updates](#)). For more information, [compare Functions and Logic Apps](#).
- Implement this architecture using customized AI components rather than existing Azure services. For example, extend the pipeline using a customized model that detects certain people in an image as opposed to the generic people count, gender, and age data collected in this example. To use customized machine learning or AI models with this architecture, build the models as RESTful endpoints so they can be called from Azure Functions.
- Use a different input mechanism instead of RSS feeds. Use multiple generators or ingestion processes to feed Azure Cosmos DB and Azure Storage.
- [Azure Cognitive Search](#) is an AI feature in Azure Search that can also be used to extract text from images, blobs, and other unstructured data sources.

Scenario details

This scenario contains examples for [English](#), [Russian](#), and [German](#) news feeds, but you can easily extend it to other RSS feeds and other languages. For ease of deployment, the data collection, processing, and analysis are based entirely on Azure services.

Potential use cases

While this scenario is based on processing of RSS feeds, it's relevant to any document, website, or article where you would need to:

- Translate text to a language of choice.
- Find key phrases, entities, and user sentiment in digital content.
- Detect objects, text, and landmarks in images associated with a digital article.
- Detect people by gender and age in images associated with digital content.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

For simplicity, this example scenario uses only a few of the available APIs and services from Azure Cognitive Services. For example, text in images can be analyzed using the [Text Analytics API](#). The target language in this scenario is assumed to be English, but you can change the input to any [supported language](#).

Scalability

Azure Functions scaling depends on the [hosting plan](#) you use. This solution assumes a [Consumption plan](#), in which compute power is automatically allocated to the functions when required. You pay only when your functions are running. Another option is to use a [Dedicated plan](#), which allows you to scale between tiers to allocate a different amount of resources.

With Azure Cosmos DB, the key is to distribute your workload roughly evenly among a sufficiently large number of [partition keys](#). There's no limit to the total amount of data that a container can store or to the total amount of [throughput](#) that a container can support.

Management and logging

This solution uses [Application Insights](#) to collect performance and logging information. An instance of Application Insights is created with the deployment in the same resource group as the other services needed for this deployment.

To view the logs generated by the solution:

1. Go to [Azure portal](#) and navigate to the resource group created for the deployment.

2. Select the **Application Insights** instance.
3. From the **Application Insights** section, navigate to **Investigate\Search** and search the data.

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](#).

Azure Cosmos DB uses a secured connection and shared access signature through the C# SDK provided by Microsoft. There are no other externally facing surface areas. Learn more about security [best practices](#) for Azure Cosmos DB.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

Azure Cosmos DB is powerful but incurs the greatest [cost](#) in this deployment. You can use another storage solution by refactoring the Azure Functions code provided.

Pricing for Azure Functions varies depending on the [plan](#) it runs in.

Deploy this scenario

Note

You must have an existing Azure account. If you don't have an Azure subscription, create a [free account](#) before you begin.

All the code for this scenario is available in the [GitHub](#) repository. This repository contains the source code used to build the generator application that feeds the pipeline for this demo.

Next steps

- [Choosing an analytical data store in Azure](#)
- [Choosing a data analytics technology in Azure](#)
- [Choosing a big data storage technology in Azure](#)

- Introduction to Azure Blob storage
- Welcome to Azure Cosmos DB
- Introduction to Azure Functions

Related resources

Additional analytics architectures:

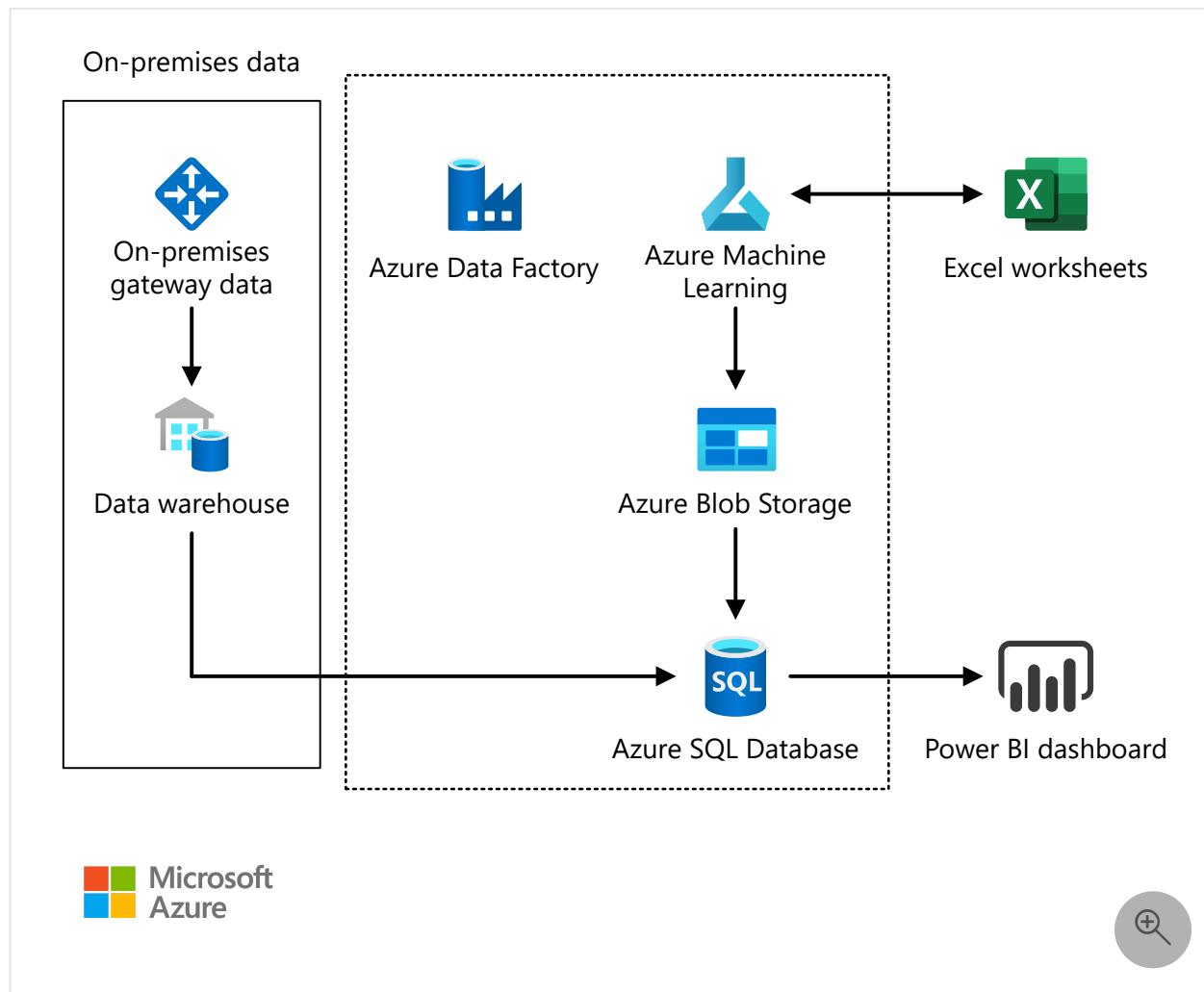
- Automated enterprise BI
- Analytics end-to-end with Azure Synapse
- Data warehousing and analytics
- Mass ingestion and analysis of news feeds on Azure
- Stream processing with Azure Databricks
- Stream processing with Azure Stream Analytics

Interactive price analytics using transaction history data

Azure Data Factory Azure Machine Learning Excel Azure Blob Storage Azure SQL Database

The Price Analytics solution utilizes your transactional history data to show you how the demand for your products responds to the prices you offer.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

1. [Azure Machine Learning](#) enables building pricing models.
2. [Azure Blob storage](#) stores model and any intermediate data that's generated.

3. [Azure SQL Database](#) stores transaction history data and any generated model predictions.
4. [Azure Data Factory](#) is used to schedule periodic (for example, weekly) model refreshes.
5. [Power BI](#) enables a visualization of the results.
6. [Excel](#) spreadsheets consume predictive web services.

Components

- [Azure Data Factory](#)
- [Azure Machine Learning](#)
- [Microsoft Excel](#) worksheets
- [Azure Blob Storage](#)
- [Azure SQL Database](#)
- [Dashboard](#) in [Power BI](#)

Solution details

The Price Analytics solution utilizes your transactional history data to show you how the demand for your products responds to the prices you offer. It recommends pricing changes and allows you to simulate how changes in price would affect your demand, at a fine granularity.

The solution provides a dashboard where you can see:

- Optimal pricing recommendations.
- Item elasticities at an item-site-channel-segment level.
- Estimates of related-product effects such as cannibalization.
- Forecasts given current process.
- Model performance metrics.

Using direct interaction with the pricing model in Excel, you can:

- Paste your sales data there and analyze your prices without the need to integrate the data into the solution database first.
- Simulate promotions and plot demand curves (showing demand response to price).
- Work with dashboard data in numerical form.

The rich functionality isn't confined to Excel. It's driven by web services that you or your implementation partner can call directly from your business applications, integrating price analysis into your business applications.

Potential use cases

This architecture is ideal for the retail industry, providing pricing recommendations, estimations, and forecasts.

Solution description

At the core of a rigorous price analysis workflow is price elasticity modeling and optimal pricing recommendations. The state-of-the-art modeling approach mitigates the two worst pitfalls of modeling price sensitivity from historical data: confounding and data sparsity.

Confounding is the presence of factors other than price that affect demand. We use a "double-ML" approach that subtracts out the predictable components of price and demand variation before estimating the elasticity. This approach immunizes the estimates to most forms of confounding. The solution can also be customized by an implementation partner to use your data capturing potential external demand drivers other than price. Our [blog post](#) gives more detail on the data science of prices.

Data sparsity occurs because the optimal price varies at a fine grain: businesses can set prices by item, site, sales channel, and even customer segment. But pricing solutions often only give estimates at product category level, because the transaction history may only contain a few sales for each specific situation. Our pricing solution uses "hierarchical regularization" to produce consistent estimates in such data-poor situations: in absence of evidence, the model borrows information from other items in the same category, same items in other sites, and so on. As the amount of historical data on a given item-site-channel combination increases, its elasticity estimate will be fine-tuned more specifically.

This pricing analytics solution idea shows you how you can develop a pricing model for products that is based on elasticity estimates from transaction history data. This solution is targeted at mid-size companies with small pricing teams who lack extensive data science support for bespoke pricing analytics models.

Interaction with the pricing model is via Excel where you can easily paste your sales data and analyze your prices without the need to integrate the data into the solution database first. In the spreadsheet, you can simulate promotions and plot demand curves (showing demand response to price), and access dashboard data in numerical form. The rich functionality of the pricing model can also be accessed from web services, integrating price analytics directly into your business applications.

Azure Machine Learning is the core logic in this solution from which elasticity models are created. Machine learning models can be set up with to avoid two common pitfalls of price modeling from historical data: confounding effects and data sparsity.

The solution provides the following advantages:

- Shows you in one glance (via the dashboard) how elastic your product demand is.
- Provides pricing recommendations for every product in your item catalog.
- Discovers related products (replacements and complements).
- Lets you simulate promotional scenarios in Excel.

Considerations

Considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

To calculate a current estimate, use the [Azure pricing calculator](#). The estimated solution should include the following service costs:

- S1 standard ML service plan
- S2 SQL Database
- App hosting plan
- Miscellaneous ADF data activities and storage costs

If you're just exploring the solution, you can delete it in a few days or hours. The costs will stop being charged when you delete the Azure components.

Deploy this scenario

The AI Gallery solution, which is an implementation of this solution architecture, has two key roles: technical resources and end users (such as pricing managers).

Technical resources deploy the solution and connect it to a business data warehouse. For more information, read the [Technical Guide](#). End users using the model via a spreadsheet (or integrated into a business application), should read the [User Guide](#).

Getting started

Deploy the solution with the button on the right. Instructions at the end of the deployment will have important configuration information. Leave them open.

The solution deploys with the same example data set of orange juice prices that you find behind the Try-It-Now button on the right.

While the solution is deploying, you can get a head start by testing and reviewing:

- The Try-It-Now dashboard.
- Read the [User Guide](#) for usage instructions from the perspective of a pricing analyst (MSFT login required).
- Review the [Technical Deployment Guide](#) for a technical implementation view (MSFT login required).
- Download the interactive Excel worksheet.

After the solution deploys, complete the [first walkthrough](#) (MSFT login required).

Solution dashboard

The solution dashboard's most actionable part is the Pricing Suggestion tab. It tells you which of your items are underpriced or overpriced. The tab suggests an optimal price for each item and the predicted impact of adopting the suggestion. The suggestions are prioritized by the largest opportunity to earn incremental gross margin.

An implementation of this pricing analytics solution idea is described in the [AI Gallery solution](#) and [GitHub repro](#). The AI Gallery solution uses your transactional history data to show how the demand for your products responds to the prices you offer, recommend pricing changes, and allow you to simulate how changes in price would affect your demand, at a fine granularity. The solution provides a dashboard, where you can see optimal pricing recommendations, item elasticities at an item-site-channel-segment level, estimates of related-product effects such "as cannibalization", forecasts given current process, and model performance metrics.

Solution architecture

The solution uses an Azure SQL Database instance to store your transactional data and the generated model predictions. There are a dozen elasticity modeling core services, which are authored in Azure ML using Python core libraries. Azure Data Factory schedules weekly model refreshes. The results display in a Power BI dashboard. The provided Excel spreadsheet consumes the predictive Web Services.

Read the [Technical Deployment Guide](#) for a more detailed discussion of the architecture, including the topic of connecting your own data and customization (GitHub login required).

Next steps

Learn more about the component technologies:

- [Introduction to Azure Data Factory](#)
- [What is Azure Machine Learning?](#)
- [Introduction to Azure Blob storage](#)
- [What is Azure SQL Database?](#)
- [What is Power BI?](#)
- [Create dashboards in Power BI](#)

Learn more about pricing solutions:

- [AI Gallery Interactive Pricing Solution](#)
- [GitHub repo for Interactive Price Analytics](#)
 - [Technical Deployment Guide](#) - for a more detailed discussion of the architecture, connecting your own data and customization.
 - [User Guide](#) - for end users of the solution such as pricing managers.
- [Blog post: A Pricing Engine for Everyone built with AzureML and Python](#)
- [Microsoft Learn Path: Build AI solutions with Azure Machine Learning](#)

Related resources

Explore related architectures:

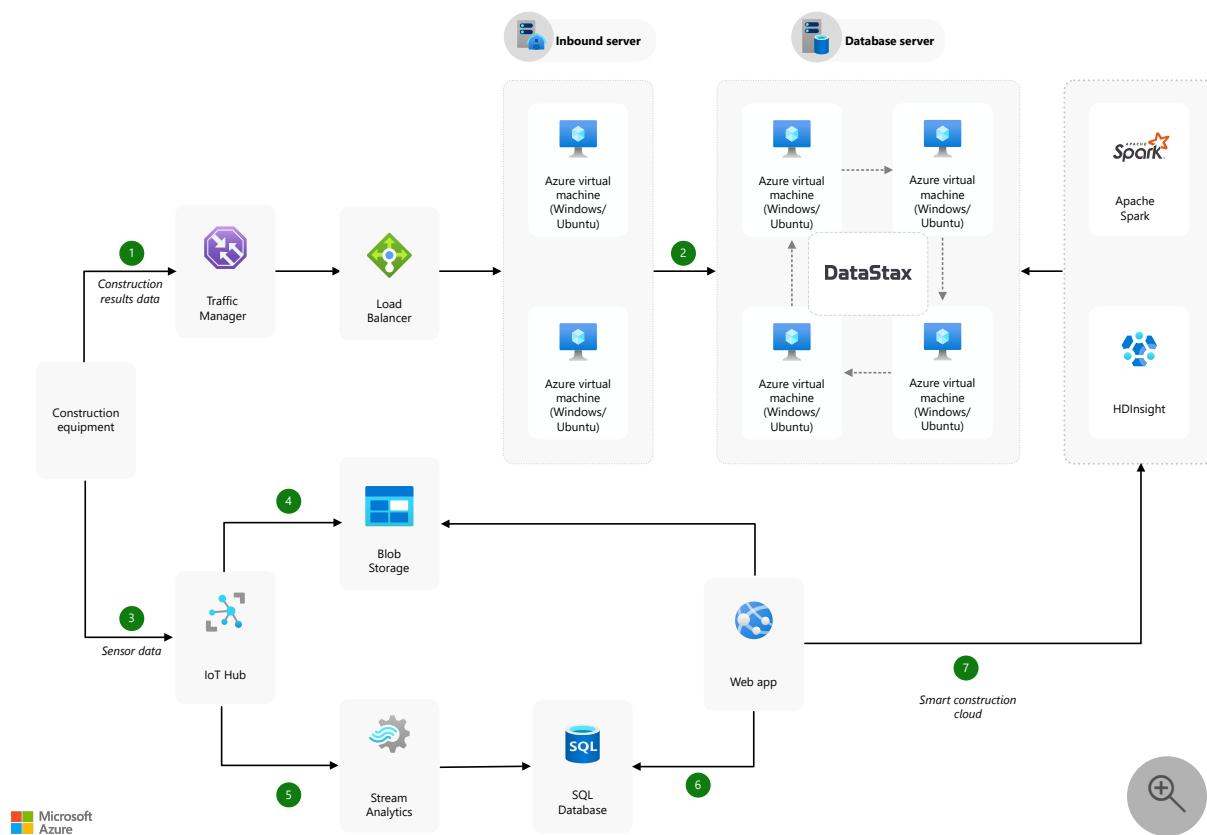
- [Demand forecasting for shipping and distribution](#)
- [Use a demand forecasting model for price optimization](#)
- [Predictive maintenance](#)
- [Predictive insights with vehicle telematics](#)
- [Predictive aircraft engine monitoring](#)

IoT and data analytics

Azure Cosmos DB Azure IoT Hub Azure SQL Database Azure Table Storage

This example scenario is relevant to organizations building solutions that integrate data from many IoT devices into a comprehensive data analysis architecture to improve and automate decision making. Potential applications include construction, mining, manufacturing, or other industry solutions involving large volumes of data from many IoT-based data inputs.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

The data flows through the solution as follows:

1. Construction equipment collects sensor data and sends the construction results data at regular intervals to load balanced web services hosted on a cluster of Azure virtual machines.

2. The custom web services ingest the construction results data and store it in an Apache Cassandra cluster also running on Azure virtual machines.
3. Another dataset is gathered by IoT sensors on various construction equipment and sent to IoT Hub.
4. Raw data collected is sent directly from IoT Hub to Azure blob storage and is immediately available for viewing and analysis.
5. Data collected via IoT Hub is processed in near real time by an Azure Stream Analytics job and stored in an Azure SQL database.
6. The Smart Construction Cloud web application is available to analysts and end users to view and analyze sensor data and imagery.
7. Batch jobs are initiated on demand by users of the web application. The batch job runs in Apache Spark on HDInsight and analyzes new data stored in the Cassandra cluster.

Components

- [IoT Hub](#) acts as a central message hub for secure bi-directional communication with per-device identity between the cloud platform and the construction equipment and other site elements. IoT Hub can rapidly collect data for each device for ingestion into the data analytics pipeline.
- [Azure Stream Analytics](#) is an event-processing engine that can analyze high volumes of data streaming from devices and other data sources. It also supports extracting information from data streams to identify patterns and relationships. In this scenario, Stream Analytics ingests and analyzes data from IoT devices and stores the results in Azure SQL Database.
- [Azure SQL Database](#) contains the results of analyzed data from IoT devices and meters, which can be viewed by analysts and users via an Azure-based Web application.
- [Blob storage](#) stores image data gathered from the IoT hub devices. The image data can be viewed via the web application.
- [Traffic Manager](#) controls the distribution of user traffic for service endpoints in different Azure regions.
- [Load Balancer](#) distributes data submissions from construction equipment devices across the VM-based web services to provide high availability.
- [Azure Virtual Machines](#) host the web services that receive and ingest the construction results data into the Apache Cassandra database.
- [Apache Cassandra](#) is a distributed NoSQL database used to store construction data for later processing by Apache Spark.
- [Web Apps](#) hosts the end-user web application, which can be used to query and view source data and images. Users can also initiate batch jobs in Apache Spark via

the application.

- [Apache Spark on HDInsight](#) supports in-memory processing to boost the performance of big-data analytic applications. In this scenario, Spark is used to run complex algorithms over the data stored in Apache Cassandra.

Alternatives

- [Azure Cosmos DB](#) is an alternative NoSQL database technology. Azure Cosmos DB provides [multi-master support at global scale](#) with [multiple well-defined consistency levels](#) to meet various customer requirements. It also supports the [Azure Cosmos DB for Apache Cassandra](#).
- [Azure Databricks](#) is an Apache Spark-based analytics platform optimized for Azure. It is integrated with Azure to provide one-click setup, streamlined workflows, and an interactive collaborative workspace.
- [Data Lake Storage](#) is an alternative to Blob storage. For this scenario, Data Lake Storage was not available in the targeted region.
- [Web Apps](#) could also be used to host the web services for ingesting construction results data.
- Many technology options are available for real-time message ingestion, data storage, stream processing, storage of analytical data, and analytics and reporting.

Scenario details

In this scenario, a construction equipment manufacturer builds vehicles, meters, and drones that use IoT and GPS technologies to emit telemetry data. The company wants to modernize their data architecture to better monitor operating conditions and equipment health. Replacing the company's legacy solution using on-premises infrastructure would be both time intensive and labor intensive, and would not be able to scale sufficiently to handle the anticipated data volume.

The company wants to build a cloud-based "smart construction" solution. It should gather a comprehensive set of data for a construction site and automate the operation and maintenance of the various elements of the site. The company's goals include:

- Integrating and analyzing all construction site equipment and data to minimize equipment downtime and reduce theft.
- Remotely and automatically controlling construction equipment to mitigate the effects of a labor shortage, ultimately requiring fewer workers and enabling lower-skilled workers to succeed.
- Minimizing the operating costs and labor requirements for the supporting infrastructure, while increasing productivity and safety.

- Easily scaling the infrastructure to support increases in telemetry data.
- Complying with all relevant legal requirements by provisioning resources in-country/region without compromising system availability.
- Using open-source software to maximize the investment in workers' current skills.

Using managed Azure services such as IoT Hub and HDInsight will allow the customer to rapidly build and deploy a comprehensive solution with a lower operating cost. If you have additional data analytics needs, you should review the list of available [fully managed data analytics services in Azure](#).

Potential use cases

Other relevant use cases include:

- Construction (facilities and real-estate), mining (energy), or equipment manufacturing scenarios
- Large-scale collection of device data for storage and analysis
- Ingestion and analysis of large datasets

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

The broad availability of Azure regions is an important factor for this scenario. Having more than one Azure region in a single country/region can provide disaster recovery while also enabling compliance with contractual obligations and law enforcement requirements. Azure's high-speed communication between regions is also an important factor in this scenario.

Azure support for open-source technologies allowed the customer to take advantage of their existing workforce skills. The customer can also accelerate the adoption of new technologies with lower costs and operating workloads compared to an on-premises solution.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

The following considerations will drive a substantial portion of the costs for this solution.

- Azure virtual machine costs will increase linearly as additional instances are provisioned. Virtual machines that are deallocated will only incur storage costs, and not compute costs. These deallocated machines can then be reallocated when demand is high.
- [IoT Hub](#) costs are driven by the number of IoT units provisioned as well as the service tier chosen, which determines the number of messages per day per unit allowed.
- [Stream Analytics](#) is priced by the number of streaming units required to process the data into the service.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Alex Buck](#) | Senior Content Developer

Related resources

Additional IoT architectures:

- [Azure IoT reference architecture](#)
- [IoT using Azure Cosmos DB](#)
- [Retail - Buy online, pickup in store \(BOPIS\)](#)
- [Predictive maintenance with the intelligent IoT Edge](#)
- [Secure your IoT SaaS app with the Microsoft identity platform](#)

IoT architecture guides:

- [IoT concepts](#)
- [Azure IoT Edge Vision](#)
- [Azure industrial IoT analytics guidance](#)
- [IoT patterns](#)

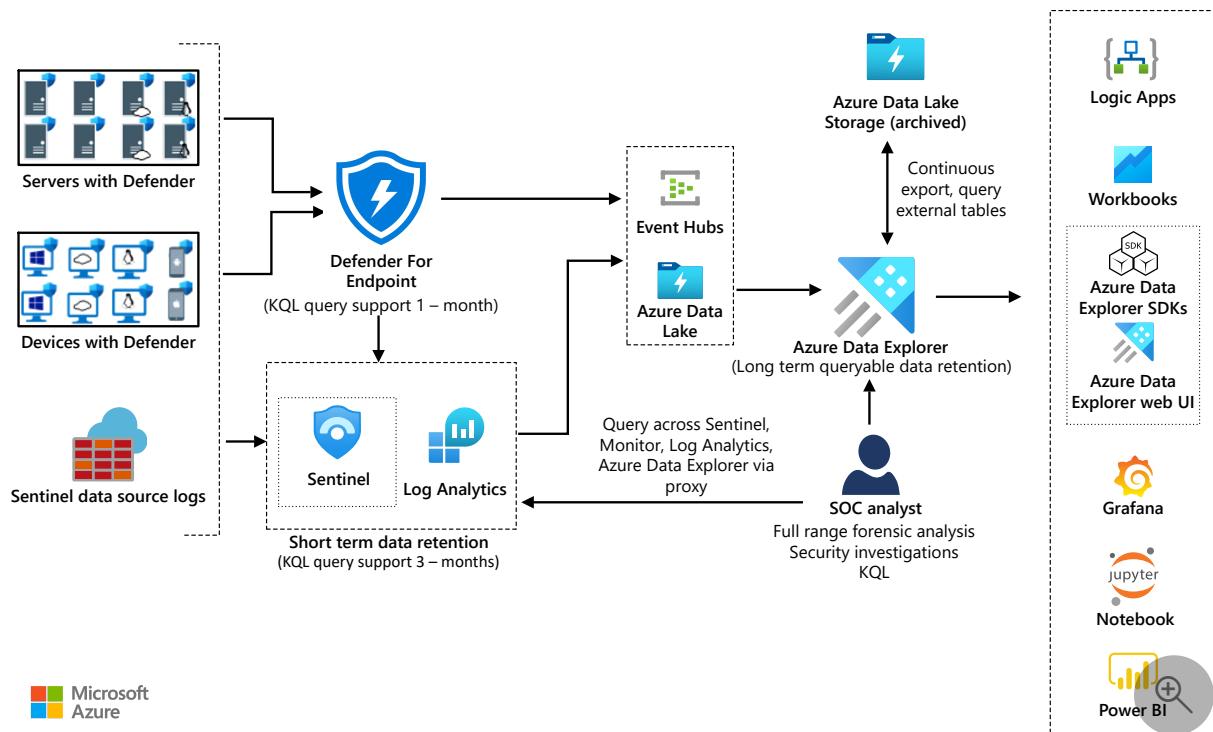
Long-term security log retention with Azure Data Explorer

Azure Data Explorer Azure Data Lake Storage Azure Event Hubs Azure Log Analytics Microsoft Sentinel

This solution stores security logs in Azure Data Explorer on a long-term basis. This solution minimizes costs and provides easy access when you need to query the data.

Grafana [↗](#) and *Jupyter Notebooks* [↗](#) are trademarks of their respective companies. No endorsement is implied by the use of these marks.

Architecture



Download a [Visio file](#) [↗](#) of this architecture.

Dataflow

1. For SIEM and SOAR, an enterprise uses Sentinel and Defender for Endpoint.
2. Defender for Endpoint uses native functionality to export data to Azure Event Hubs and Azure Data Lake. Sentinel ingests Defender for Endpoint data to monitor devices.

3. Sentinel uses Log Analytics as a data platform for exporting data to Event Hubs and Azure Data Lake.
4. Azure Data Explorer uses connectors for [Event Hubs](#), [Azure Blob Storage](#), and [Azure Data Lake Storage](#) to ingest data with low-latency and high throughput. This process uses [Azure Event Grid](#), which triggers the Azure Data Explorer ingestion pipeline.
5. If needed, Azure Data Explorer continuously exports security logs to Azure Storage. These logs are in compressed, partitioned Parquet format and are ready to be queried.
6. To follow regulatory requirements, Azure Data Explorer exports pre-aggregated data to Data Lake Storage for archiving.
7. Log Analytics and Sentinel support cross-service queries with Azure Data Explorer. SOC analysts use this capability to run full-range investigations on security data.
8. Azure Data Explorer provides native capabilities for processing, aggregating, and analyzing data.
9. Various tools provide near real-time analytics dashboards that quickly deliver insights:
 - [Azure Data Explorer dashboards](#)
 - [Power BI](#)
 - [Grafana](#)

Components

- [Defender for Endpoint](#)  protects organizations from threats across devices, identities, apps, email, data, and cloud workloads.
- [Sentinel](#) is a cloud-native SIEM and SOAR solution. It uses advanced AI and security analytics to detect, hunt, prevent, and respond to threats across enterprises.
- [Monitor](#)  is a software as a service (SaaS) solution that collects and analyzes data on environments and Azure resources. This data includes app telemetry, such as performance metrics and activity logs. Monitor also offers alerting functionality.
- [Log Analytics](#) is a Monitor service that you can use to query and inspect Monitor log data. Log Analytics also provides features for charting and statistically analyzing query results.

- [Event Hubs](#) is a fully managed, real-time data ingestion service that's straightforward and scalable.
- [Data Lake Storage](#) is a scalable storage repository that holds a large amount of data in the data's native, raw format. This data lake is built on top of [Blob Storage](#) and provides functionality for storing and processing data.
- [Azure Data Explorer](#) is a fast, fully managed, and highly scalable data analytics platform. You can use this cloud service for real-time analysis on large volumes of data. Azure Data Explorer is optimized for interactive, ad-hoc queries. It can handle diverse data streams from applications, websites, IoT devices, and other sources.
- [Azure Data Explorer dashboards](#) natively import data from Azure Data Explorer Web UI queries. These optimized dashboards provide a way to display and explore query results.

Alternatives

- Instead of using Azure Data Explorer for long-term storage of security logs, you can use Storage. This approach simplifies the architecture and can help control the cost. A disadvantage is the need to rehydrate the logs for security audits and interactive investigative queries. With Azure Data Explorer, you can move data from the cold partition to the hot partition by changing a policy. This functionality speeds up data exploration.
- Another option with this solution is to send all data, regardless of its security value, to Sentinel and Azure Data Explorer at the same time. Some duplication results, but the cost savings can be significant. Because Azure Data Explorer provides long-term storage, you can reduce your Sentinel retention costs with this approach.
- Log Analytics doesn't currently support exporting custom log tables. In this scenario, you can use Azure Logic Apps to export data from Log Analytics workspaces. For more information, see [Archive data from Log Analytics workspace to Azure Storage using Logic Apps](#).

Scenario details

Security logs are useful for identifying threats and tracing unauthorized attempts to access data. Security attacks can begin well before they're discovered. As a result, having access to long-term security logs is important. Querying long-term logs is critical for identifying the impact of threats and investigating the spread of illicit access attempts.

This article outlines a solution for long-term retention of security logs. At the core of the architecture is Azure Data Explorer. This service provides storage for security data at minimal cost but keeps that data in a format that you can query. Other main components include:

- Microsoft Defender for Endpoint and Microsoft Sentinel, for these capabilities:
 - Comprehensive endpoint security
 - Security information and event management (SIEM)
 - Security orchestration automated response (SOAR)
- Log Analytics, for short-term storage of Sentinel security logs.

Potential use cases

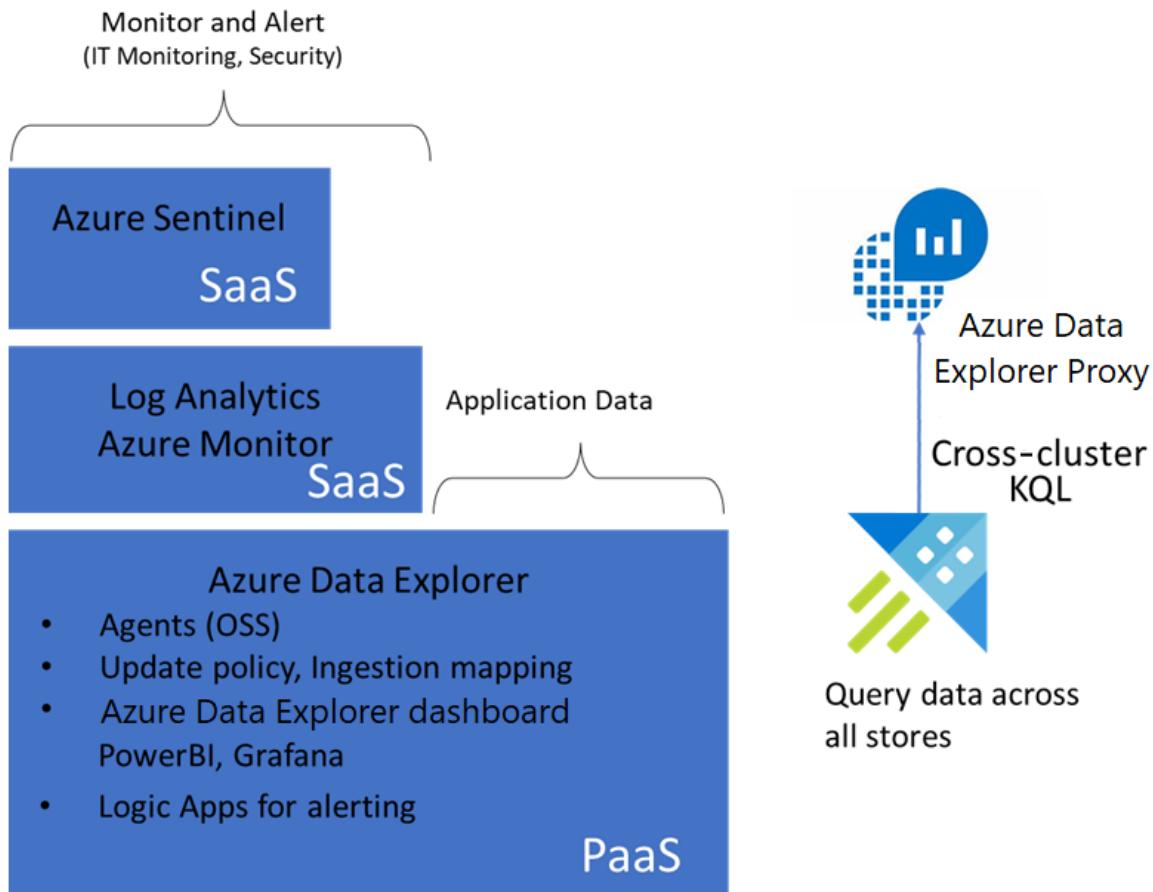
This solution applies to various scenarios. Specifically, security operations center (SOC) analysts can use this solution for:

- Full-scale investigations.
- Forensic analysis.
- Threat hunting.
- Security audits.

A customer testifies to the usefulness of the solution: "We deployed an Azure Data Explorer cluster almost a year and a half ago. In the last Solarigate data breach, we used an Azure Data Explorer cluster for forensic analysis. A Microsoft Dart team also used an Azure Data Explorer cluster to complete the investigation. Long-term security data retention is critical for full-scale data investigations."

Monitoring stack

The following diagram shows the Azure monitoring stack:



- Sentinel uses a Log Analytics workspace to store security logs and provide SIEM and SOAR solutions.
- Monitor tracks the status of IT assets and sends alerts when needed.
- Azure Data Explorer provides an underlying data platform that stores security logs for Log Analytics workspaces, Monitor, and Sentinel.

Main features

The solution's main features offer many benefits, as the following sections explain.

Long-term queryable data store

Azure Data Explorer indexes data during the storage process, making the data available for queries. When you need to focus on running audits and investigations, there's no need to process the data. Querying the data is straightforward.

Full-scale forensic analysis

Azure Data Explorer, Log Analytics, and Sentinel support cross-service queries. As a result, in a single query, you can reference data that's stored in any of these services.

SOC analysts can use the Kusto query language (KQL) to run full-range investigations. You can also use Azure Data Explorer queries in Sentinel for hunting purposes. For more information, see [What's New: Sentinel Hunting supports ADX cross-resource queries](#).

On-demand data caching

Azure Data Explorer supports [window-based hot caching](#). This functionality provides a way to move data from a selected period into the hot cache. Then you can run fast queries on the data, making investigations more efficient. You might need to add compute nodes to the hot cache for this purpose. After the investigation is complete, you can change the hot cache policy to move the data into the cold partition. You can also restore the cluster to its original size.

Continuous exporting to archive data

To follow regulatory requirements, some enterprises need to store security logs for an unlimited amount of time. Azure Data Explorer supports continuous exporting of data. You can use this capability to build an archival tier by storing security logs in Storage.

Proven query language

The Kusto query language is native to Azure Data Explorer. This language is also available in Log Analytics workspaces and Sentinel threat-hunting environments. This availability significantly reduces the learning curve for SOC analysts. Queries that you run on Sentinel also work on data that you store in Azure Data Explorer clusters.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Keep the following points in mind when you implement this solution.

Scalability

Consider these scalability issues:

Data export method

If you need to export a large amount of data from Log Analytics, you might reach Event Hubs capacity limits. To avoid this situation:

- Export data from Log Analytics into Blob Storage.
- Use Azure Data Factory workloads to periodically export the data into Azure Data Explorer.

By using this method, you can copy data from Data Factory only when the data nears its retention limit in Sentinel or Log Analytics. As a result, you avoid duplicating the data.

For more information, see [Export data from Log Analytics into Azure Data Explorer](#).

Query usage and audit preparedness

Generally, you keep data in the cold cache in your Azure Data Explorer cluster. This approach minimizes your cluster cost and is sufficient for most queries that involve data from previous months. But when you query large data ranges, you might need to scale out the cluster and load the data into the hot cache.

You can use the hot window feature of the hot cache policy for this purpose. You can also use this feature when you audit long-term data. When you use the hot window, you might need to scale your cluster up or out to make room for more data in the hot cache. After you've finished querying the large data range, change the hot cache policy to reduce your computing cost.

By turning on the optimized autoscale feature in your Azure Data Explorer cluster, you can optimize your cluster size based on the caching policy. For more information on querying cold data in Azure Data Explorer, see [Query cold data with hot windows](#).

Performance efficiency

Performance efficiency is the ability of your workload to scale to meet the demands placed on it by users in an efficient manner. For more information, see [Performance efficiency pillar overview](#).

If you need to store security data for a long time or for an unlimited period, export the logs to Storage. Azure Data Explorer supports continuous exporting of data. By using this functionality, you can export data to Storage in compressed, partitioned Parquet format. You can then seamlessly query that data. For more information, see [Continuous data export overview](#).

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

The Azure Data Explorer cluster cost is primarily based on the computing power that's used to store data in the hot cache. Queries on hot cache data offer better performance over cold cache queries. This solution stores most of the data in the cold cache, minimizing the computing cost.

To explore the cost of running this solution in your environment, use the [Azure pricing calculator](#).

Deploy this scenario

To automate deployment, use this [PowerShell script](#). This script creates these components:

- The target table
- The raw table
- The table mapping that defines how Event Hubs records land in the raw table
- Retention and update policies
- Event Hubs namespaces
- Data export rules in the Log Analytics workspace
- The data connection between Event Hubs and the Azure Data Explorer raw data table

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Deepak Agrawal](#) | Product Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Integrate Azure Data Explorer for long-term log retention](#)
- [Move Your Microsoft Sentinel Logs to Long-Term Storage with Ease](#)
- [Cross-resource query Azure Data Explorer by using Azure Monitor](#)

- HOW TO: Configure Microsoft Sentinel data export for long-term storage ↗
- Using Azure Data Explorer for long-term retention of Microsoft Sentinel logs ↗
- What's New: Microsoft Sentinel Hunting supports ADX cross-resource queries ↗
- How to stream Microsoft Defender ATP hunting logs in Azure Data Explorer ↗
- Blog Series: Limitless Advanced Hunting with Azure Data Explorer (ADX) ↗

Related resources

- [Azure Data Explorer monitoring](#)
- [Azure Data Explorer interactive analytics](#)
- [Big data analytics with Azure Data Explorer](#)

Near real-time lakehouse data processing

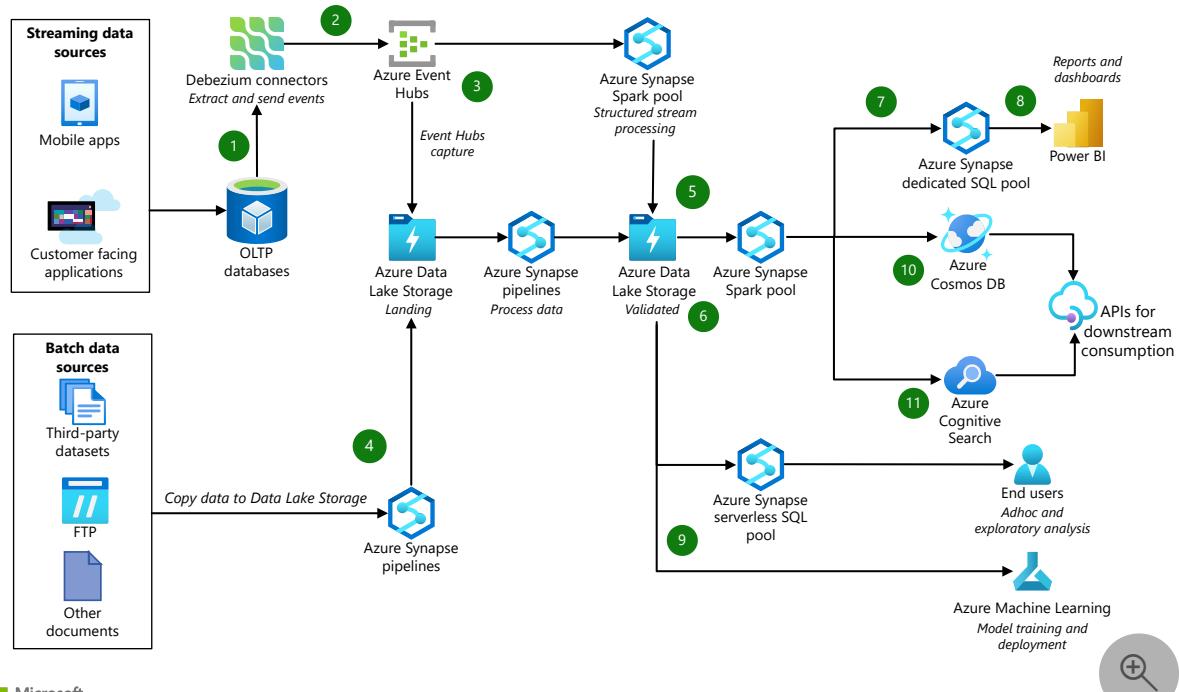
Azure AI Search Azure Cosmos DB Azure Data Lake Azure Event Hubs Azure Synapse Analytics

Data-driven enterprises need to keep their back end and analytics systems in near real-time sync with customer-facing applications. The impact of transactions, updates, and changes must reflect accurately through end-to-end processes, related applications, and online transaction processing (OLTP) systems. The tolerable latency for changes in OLTP applications to reflect in the downstream systems that use the data might be just a few minutes.

This article describes an end-to-end solution for near real-time data processing to keep lakehouse data in sync. The solution uses Azure Event Hubs, Azure Synapse Analytics, and Azure Data Lake Storage for data processing and analytics.

Apache® and [Apache Spark](#) 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

1. Change data capture is a prerequisite for source systems to listen to changes. [Debezium connectors](#) can connect to different source systems and tap into changes as they happen. The connectors can capture changes and produce events from various relational database management systems (RDBMS). Installing a Debezium connector requires a Kafka connect system.
2. The connectors extract change data and send the captured events to Azure Event Hubs. Event Hubs can receive large amounts of data from multiple sources.
3. Event Hubs directly streams the data to Azure Synapse Analytics Spark pools, or can send the data to an Azure Data Lake Storage landing zone in raw format.
4. Other batch data sources can use Azure Synapse pipelines to copy data to Data Lake Storage and make it available for processing. An end-to-end extract, transform, and load (ETL) workflow might need to chain different steps or add dependencies between steps. Azure Synapse pipelines can orchestrate workflow dependencies within the overall processing framework.
5. Azure Synapse Spark pools use fully supported Apache Spark structured streaming APIs to process data in the Spark streaming framework. The data processing step incorporates data quality checks and high-level business rule validations.
6. Data Lake Storage stores the validated data in the open [Delta Lake](#) format. Delta Lake provides atomicity, consistency, isolation, and durability (ACID) semantics and transactions, scalable metadata handling, and unified streaming and batch data processing for existing data lakes.

Using indexes for query acceleration augments Delta with further performance enhancements. Data from the Data Lake Storage validated zone can also be a source for further advanced analytics and machine learning.
7. Data from the Data Lake Storage validated zone, transformed and enriched with more rules into its final processed state, loads to a dedicated SQL pool for running large scale analytical queries.
8. Power BI uses the data exposed through the dedicated SQL pool to build enterprise-grade dashboards and reports.
9. You can also use captured raw data in the Data Lake Store landing zone and validated data in the Delta format for:

- Further ad-hoc and exploratory analysis through Azure Synapse SQL serverless pools.
 - Machine learning through Azure Machine Learning.
10. For some low-latency interfaces, data must be denormalized for single-digit server latencies. This usage scenario is mainly for API responses. This scenario queries documents in a NoSQL datastore such as Azure Cosmos DB for single-digit millisecond responses.
11. The Azure Cosmos DB partitioning strategy might not lend itself to all query patterns. If that's the case, you can augment the solution by indexing the data that the APIs need to access with Azure Cognitive Search. Azure Cosmos DB and Cognitive Search can fulfill most scenarios that require low latency query responses.

Components

This solution uses the following Azure components:

- [Event Hubs](#) is a managed, distributed ingestion service that can scale to ingest large amounts of data. With the Event Hubs subscriber-publisher mechanism, different applications can send messages to topics in Event Hubs, and downstream consumers can connect to and process messages. The Event Hubs Capture feature can write messages to Data Lake Storage in AVRO format as they arrive. This ability enables easy micro-batch processing and long-term retention scenarios. Event Hubs also offers a Kafka-compatible API and supports schema registry.
- [Data Lake Storage](#) forms the storage subsystem that stores all data in raw and validated formats. Data Lake Storage can handle transactions at scale, and supports different file formats and sizes. Hierarchical namespaces help organize data into a familiar folder structure and support Portable Operating System Interface for Unix (POSIX) permissions. The Azure Blob Filesystem (ABFS) driver offers a Hadoop-compatible API.
- [Azure Synapse Analytics](#) is a limitless analytics service that brings together data integration, enterprise data warehousing, and big data analytics. This solution uses the following features of the Azure Synapse Analytics ecosystem:
 - [Azure Synapse Spark pools](#) offer an on-demand Spark runtime that adds built-in performance enhancements to open-source Spark. Customers can configure flexible autoscale settings, submit jobs remotely through the Apache Livy endpoint, and use the Synapse Studio notebook interface for interactive experiences.

- [Azure Synapse SQL serverless pools](#) provide an interface to query lakehouse data by using a familiar T-SQL syntax. There's no infrastructure to set up, and Azure Synapse workspace deployment automatically creates the endpoint. Azure Synapse SQL serverless pools enable basic discovery and exploration of data in place, and are a good option for user ad-hoc query analysis.
- [Azure Synapse dedicated SQL pools](#) store data in relational tables with columnar storage. Dedicated SQL pools use a scale-out architecture to distribute data processing across multiple nodes. PolyBase queries bring the data into SQL pool tables. The tables can connect to Power BI for analysis and reporting.
- [Power BI](#) provides a visual interface to create and access reports and dashboards. Power BI Desktop can connect to various data sources, combine the sources into a data model, and build reports or dashboards. With Power BI, you can transform data based on business requirements, and share visuals and reports with others through the Power BI service.
- [Azure Cosmos DB](#) is a managed, multi-modal NoSQL database that supports open APIs such as MongoDB and Cassandra. This solution uses Azure Cosmos DB for applications that require single-digit millisecond response times and high availability. Azure Cosmos DB offers multi-region writes across all Azure regions. You can use [Azure Synapse Link for Azure Cosmos DB](#) to derive insights and run analytics over data in real time.
- [Azure Cognitive Search](#) is a cloud search service that can index the data your applications and APIs need. Cognitive Search has optional AI enrichment features that help with text extraction and infer text from non-text files. Cognitive Search integrates with services like Azure Data Lake Storage and Azure Cosmos DB to easily access and index data. You can query the indexed data by using a REST API or the .NET SDK. To get data from two separate indexes, you can combine them into a single index or use [complex data types](#).

Scenario details

The end-to-end workflow to process changes in near real-time requires:

- A change data capture (CDC) technology. The OLTP applications might have different back-end data stores, such as SQL Server, MySQL, and Oracle. The first step is to listen to changes as they happen, and propagate them forward.
- An ingestion buffer to publish the change events at scale. This service should have the ability to handle large amounts of data as messages arrive. Individual

subscribers can connect to this system and process the data.

- Distributed and scalable storage for data as-is in a raw format.
- A distributed, efficient stream processing system that lets users restart and manage state.
- An analytics system that runs at scale to power business decisions.
- A self-serve analytics interface.
- For low-latency API responses, a NoSQL database to store denormalized representation of the data.
- For some cases, a system to index data, refresh the index at regular intervals, and make the latest data available for downstream consumption.

All the preceding technologies should use relevant security constructs for perimeter security, authentication, authorization, and data encryption.

Potential use cases

This solution is well-suited for:

- Industries that need to propagate changes from OLTP to online analytics processing (OLAP).
- Applications that require data transformation or enrichment.

The real-time data processing scenario is especially important for financial services industries. For example, if an insurance, credit card, or bank customer makes a payment and then immediately contacts customer service, the customer support agent needs to have the latest information.

Similar scenarios apply to retail, commerce, and healthcare sectors. Enabling these scenarios streamlines operations, leading to greater organizational productivity and increased customer satisfaction.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Reliability

Reliability ensures your application can meet the commitments you make to your customers. For more information, see [Overview of the reliability pillar](#).

- Event Hubs offers 90-day data retention on the premium and dedicated tiers. For failover scenarios, you can set up a secondary namespace in the paired region and activate it during failover.
- Azure Synapse Spark pool jobs are recycled every seven days as nodes are taken down for maintenance. Consider this activity as you work through the service level agreements (SLAs) tied to the system. This limitation isn't an issue for many scenarios where recovery time objective (RTO) is around 15 minutes.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

- You can select from different Event Hubs tiers based on workload characteristics. Event Hubs bills Capture storage separately, based on the amount of data being stored on Data Lake Storage.
- Consider object lifecycle management through tiers on Azure Data Lake Storage. As data ages, you can move data from a hot tier, where you need to access recent data for analytics, to a cold storage tier that is priced much lower. The cold storage tier is a cost-effective option for long-term retention.
- You can pause the dedicated SQL pool when you're not using it in your development or test environments. You can schedule a script to pause the pool as needed, or you can pause the pool manually through the portal.
- Azure Cosmos DB offers different provisioning models, such as serverless, manual provisioned throughput, and autoscale. Consider using serverless provisioning for your development and test workloads. You can also use autoscale, where you can set maximum request units per second (RU/s) on the container. The throughput on the container scales automatically between 10% of maximum RU/s as a lower threshold and the maximum configured RU/s.

Performance efficiency

Performance efficiency is the ability of your workload to scale to meet the demands placed on it by users in an efficient manner. For more information, see [Performance efficiency pillar overview](#).

- You can scale Event Hubs through partitioning. Consider partitioning your data to preserve the order of events through a commit log. Partitioning lets you create

multiple parallel logs by maximizing the available throughput capacity.

- You can set up Azure Synapse Spark pools with small, medium, or large virtual machine (VM) SKUs, based on the workload. You can also configure autoscale on Azure Synapse Spark pools to account for spiky workloads. If you need more compute resources, the clusters automatically scale up to meet the demand, and scale down after processing is complete.
- Use best practices for designing tables in the dedicated SQL pool. Associated performance and scalability limits apply, based on the tier that the SQL pool is running on.
- Azure Cosmos DB uses partitions to scale containers, based on a partition key. All data based on a partition key forms a logical partition. Make sure to choose the correct partitioning strategy based on workload requirements. You can also use indexes for faster data retrieval.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Pratima Valavala](#) | Cloud Solution Architect

Other contributor:

- [Rajesh Mittal](#) | Cloud Solution Architect

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Azure Event Hubs connector for Apache Spark](#)
- [Scalability with Event Hubs](#)
- [Index data from Azure Cosmos DB](#)
- [What is Azure Synapse Link for Azure Cosmos DB?](#)
- [Best practices for dedicated SQL pool](#)
- [Best practices for serverless SQL pool](#)
- [Model, query, and explore data in Azure Synapse](#)
- [Build data analytics solutions using Azure Synapse serverless SQL pools](#)

Related resources

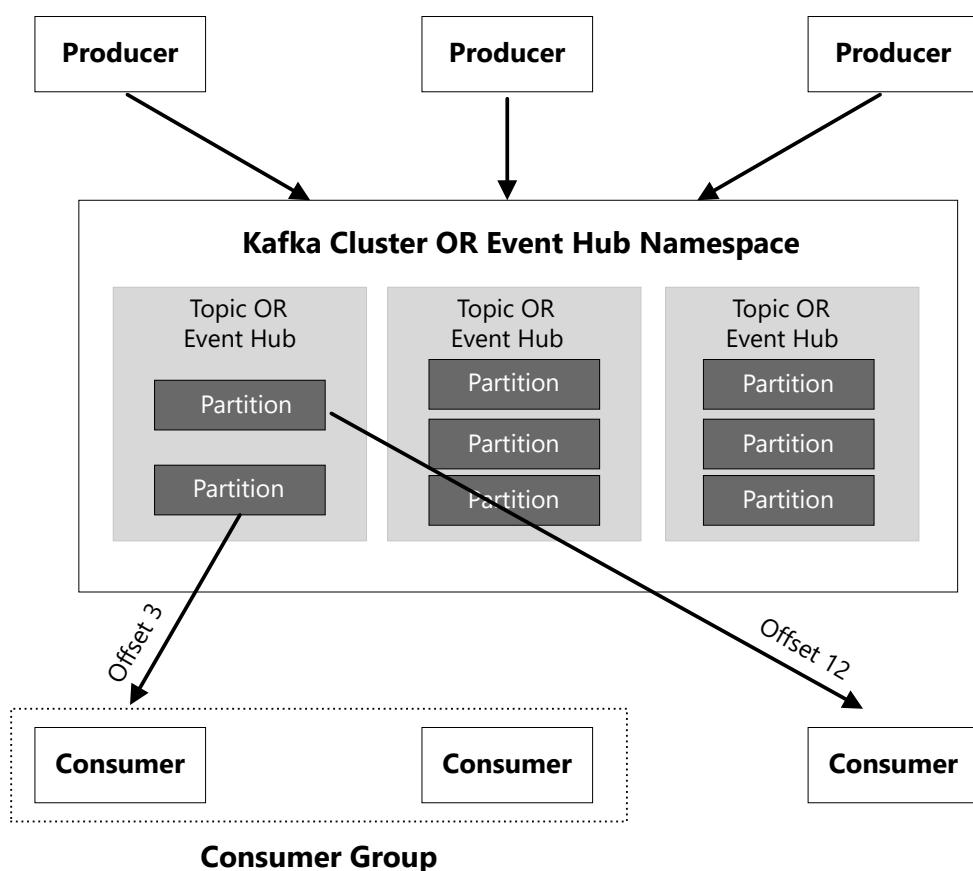
- [High throughput stream ingestion to Azure Synapse](#)
- [Secure a data lakehouse with Azure Synapse Analytics](#)
- [Query a data lake or lakehouse by using Azure Synapse serverless](#)
- [Automated enterprise BI](#)
- [Demand forecasting for shipping and distribution](#)
- [Big data analytics with enterprise grade security using Azure Synapse](#)

Partitioning in Azure Event Hubs and Kafka

Azure Blob Storage Azure Event Hubs

This reference architecture provides strategies for the [partitioning model](#) that event ingestion services use. Because event ingestion services provide solutions for high-scale event streaming, they need to process events in parallel and be able to maintain event order. They also need to balance loads and offer scalability. Partitioning models meet all of these requirements.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

- *Producers* publish data to the ingestion service, or *pipeline*. Event Hubs pipelines consist of *namespaces*. The Kafka equivalents are *clusters*.
- The pipeline distributes incoming events among *partitions*. Within each partition, events remain in production order. Events don't remain in sequence across partitions, however. The number of partitions can affect *throughput*, or the amount of data that passes through the system in a set period of time. Pipelines usually measure throughput in bits per second (bps), and sometimes in data packets per second (pps).
- Partitions reside within named streams of events. Event Hubs calls these streams *event hubs*. In Kafka, they're *topics*.
- *Consumers* are processes or applications that subscribe to topics. Each consumer reads a specific subset of the event stream. That subset can include more than one partition. However, the pipeline can assign each partition to only one consumer at a time.
- Multiple consumers can make up *consumer groups*. When a group subscribes to a topic, each consumer in the group has a separate view of the event stream. The applications work independently from each other, at their own pace. The pipeline can also use consumer groups for load sharing.
- Consumers process the feed of published events that they subscribe to. Consumers also engage in *checkpointing*. Through this process, subscribers use *offsets* to mark their position within a partition event sequence. An offset is a placeholder that works like a bookmark to identify the last event that the consumer read.

Scenario details

Specifically, this document discusses the following strategies:

- How to assign events to partitions.
- How many partitions to use.
- How to assign partitions to subscribers when rebalancing.

Many event ingestion technologies exist, including:

- [Azure Event Hubs](#): A fully managed big data streaming platform.
- [Apache Kafka ↗](#): An open-source stream-processing platform.
- [Event Hubs with Kafka](#): An alternative to running your own Kafka cluster. This Event Hubs feature provides an endpoint that is compatible with Kafka APIs.

Besides offering partitioning strategies, this document also points out differences between partitioning in Event Hubs and Kafka.

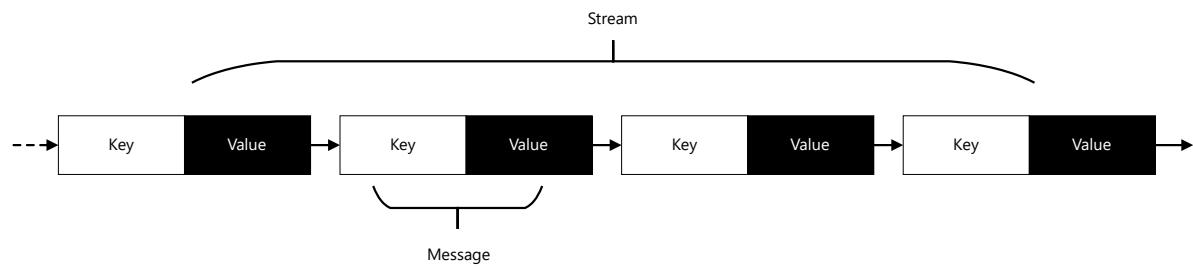
Recommendations

Keep the following recommendations in mind when developing a partitioning strategy.

Distribute events to partitions

One aspect of the partitioning strategy is the assignment policy. An event that arrives at an ingestion service goes to a partition. The assignment policy determines that partition.

Each event stores its content in its *value*. Besides the value, each event also contains a *key*, as the following diagram shows:



Download a [Visio file](#) of this architecture.

The key contains data about the event and can also play a role in the assignment policy.

Multiple approaches exist for assigning events to partitions:

- By default, services distribute events among partitions in a round-robin fashion.
- Producers can specify a partition ID with an event. The event then goes to the partition with that ID.
- Producers can provide a value for the event key. When they do, a hashing-based partitioner determines a hash value from the key. The event then goes to the partition associated with that hash value.

Keep these recommendations in mind when choosing an assignment policy:

- Use partition IDs when consumers are only interested in certain events. When those events flow to a single partition, the consumer can easily receive them by subscribing to that partition.

- Use keys when consumers need to receive events in production order. Since all events with the same key go to the same partition, events with key values can maintain their order during processing. Consumers then receive them in that order.
- With Kafka, if event grouping or ordering isn't required, avoid keys. The producer doesn't know the status of the destination partition in Kafka. If a key routes an event to a partition that's down, delays or lost events can result. In Event Hubs, events with keys first pass through a gateway before proceeding to a partition. This approach prevents events from going to unavailable partitions.
- The shape of the data can influence the partitioning approach. Consider how the downstream architecture will distribute the data when deciding on assignments.
- If consumers aggregate data on a certain attribute, you should partition on that attribute, too.
- When storage efficiency is a concern, partition on an attribute that concentrates the data to help speed up storage operations.
- Ingestion pipelines sometimes shard data to get around problems with resource bottlenecks. In these environments, align the partitioning with how the shards are split in the database.

Determine the number of partitions

Use these guidelines to decide how many partitions to use:

- Use more partitions to achieve more throughput. Each consumer reads from its assigned partition. So with more partitions, more consumers can receive events from a topic at the same time.
- Use at least as many partitions as the value of your target throughput in megabytes.
- To avoid starving consumers, use at least as many partitions as consumers. For instance, suppose eight partitions are assigned to eight consumers. Any additional consumers that subscribe will have to wait. Alternatively, you can keep one or two consumers ready to receive events when an existing consumer fails.
- Use more keys than partitions. Otherwise, some partitions won't receive any events, leading to unbalanced partition loads.
- In both Kafka and Event Hubs at the Dedicated tier level, you can change the number of partitions in an operating system. However, avoid making that change if you use keys to preserve event ordering. The reason involves the following facts:
 - Consumers rely on certain partitions and the order of the events they contain.
 - When the number of partitions changes, the mapping of events to partitions can change. For instance, when the partition count changes, this formula can produce a different assignment: `partition assignment = hash key % number of partitions`

- Kafka and Event Hubs don't attempt to redistribute events that arrived at partitions before the shuffle. As a result, the guarantee no longer holds that events arrive at a certain partition in publication order.

Besides these guidelines, you can also use this rough formula to determine the number of partitions:

$$\max(t/p, t/c)$$

It uses the following values:

- t : The target throughput.
- p : The production throughput on a single partition.
- c : The consumption throughput on a single partition.

For example, consider this situation:

- The ideal throughput is 2 MBps. For the formula, t is 2 MBps.
- A producer sends events at a rate of 1,000 events per second, making p 1 MBps.
- A consumer receives events at a rate of 500 events per second, setting c to 0.5 MBps.

With these values, the number of partitions is 4:

$$\max(t/p, t/c) = \max(2/1, 2/0.5) = \max(2, 4) = 4$$

When measuring throughput, keep these points in mind:

- The slowest consumer determines the consumption throughput. However, sometimes no information is available about downstream consumer applications. In this case, estimate the throughput by starting with one partition as a baseline. (Use this setup only in testing environments, not in production systems). Event Hubs with Standard tier pricing and one partition should produce throughput between 1 MBps and 20 MBps.
- Consumers can consume events from an ingestion pipeline at a high rate only if producers send events at a comparable rate. To determine the total required capacity of the ingestion pipeline, measure the producer's throughput, not just the consumer's.

Assign partitions to consumers when rebalancing

When consumers subscribe or unsubscribe, the pipeline rebalances the assignment of partitions to consumers. By default, Event Hubs and Kafka use a round robin approach

for rebalancing. This method distributes partitions evenly across members.

With Kafka, if you don't want the pipeline to automatically rebalance assignments, you can statically assign partitions to consumers. But you need to make sure that all partitions have subscribers and that the loads are balanced.

Besides the default round robin strategy, Kafka offers two other strategies for automatic rebalancing:

- Range assignor: Use this approach to bring together partitions from different topics. This assignment identifies topics that use the same number of partitions and the same key-partitioning logic. Then it joins partitions from those topics when making assignments to consumers.
- Sticky assignor: Use this assignment to minimize partition movement. Like round robin, this strategy ensures a uniform distribution. However, it also preserves existing assignments during rebalancing.

Considerations

Keep these points in mind when using a partitioning model.

Scalability

Using a large number of partitions can limit scalability:

- In Kafka, *brokers* store event data and offsets in files. The more partitions you use, the more open file handles you'll have. If the operating system limits the number of open files, you may need to reconfigure that setting.
- In Event Hubs, users don't face file system limitations. However, each partition manages its own Azure blob files and optimizes them in the background. A large number of partitions makes it expensive to maintain checkpoint data. The reason is that I/O operations can be time-consuming, and the storage API calls are proportional to the number of partitions.
- Each producer for Kafka and Event Hubs stores events in a buffer until a sizeable batch is available or until a specific amount of time passes. Then the producer sends the events to the ingestion pipeline. The producer maintains a buffer for each partition. When the number of partitions increases, the memory requirement of the client also expands. If consumers receive events in batches, they may also face the same issue. When consumers subscribe to a large number of partitions but have limited memory available for buffering, problems can arise.

Availability

A significant number of partitions can also adversely affect availability:

- Kafka generally positions partitions on different brokers. When a broker fails, Kafka rebalances the partitions to avoid losing events. The more partitions there are to rebalance, the longer the failover takes, increasing unavailability. Limit the number of partitions to the low thousands to avoid this issue.
- The more partitions you use, the more physical resources you put in operation. Depending on the client response, more failures can then occur.
- With more partitions, the load-balancing process has to work with more moving parts and more stress. *Transient exceptions* can result. These errors can occur when there are temporary disturbances, such as network issues or intermittent internet service. They can appear during an upgrade or load balancing, when Event Hubs sometimes moves partitions to different nodes. Handle transient behavior by incorporating retries to minimize failures. Use the [EventProcessorClient in the .NET](#) and [Java SDKs](#) or the [EventHubConsumerClient in the Python](#) and [JavaScript SDKs](#) to simplify this process.

Performance

In Kafka, events are *committed* after the pipeline has replicated them across all in-sync replicas. This approach ensures a high availability of events. Since consumers only receive committed events, the replication process adds to the *latency*. In ingestion pipelines, this term refers to the time between when a producer publishes an event and a consumer reads it. According to [experiments that Confluent ran](#), replicating 1,000 partitions from one broker to another can take about 20 milliseconds. The end-to-end latency is then at least 20 milliseconds. When the number of partitions increases further, the latency also grows. This drawback doesn't apply to Event Hubs.

Security

In Event Hubs, publishers use a [Shared Access Signature \(SAS\)](#) token to identify themselves. Consumers connect via an [AMQP 1.0 session](#). This state-aware bidirectional communication channel provides a secure way to transfer messages. Kafka also offers encryption, authorization, and authentication features, but you have to implement them yourself.

Deploy this scenario

The following code examples demonstrate how to maintain throughput, distribute to a specific partition, and preserve event order.

Maintain throughput

This example involves log aggregation. The goal isn't to process events in order, but rather, to maintain a specific throughput.

A Kafka client implements the producer and consumer methods. Since order isn't important, the code doesn't send messages to specific partitions. Instead, it uses the default partitioning assignment:

C#

```
public static void RunProducer(string broker, string connectionString,
string topic)
{
    // Set the configuration values of the producer.
    var producerConfig = new ProducerConfig
    {
        BootstrapServers = broker,
        SecurityProtocol = SecurityProtocol.SaslSsl,
        SaslMechanism = SaslMechanism.Plain,
        SaslUsername = "$ConnectionString",
        SaslPassword = connectionString,
    };

    // Set the message key to Null since the code does not use it.
    using (var p = new ProducerBuilder<Null, string>
(producerConfig).Build())
    {
        try
        {
            // Send a fixed number of messages. Use the Produce method to
            generate
            // many messages in rapid succession instead of the ProduceAsync
            method.
            for (int i=0; i < NumOfMessages; i++)
            {
                string value = "message-" + i;
                Console.WriteLine($"Sending message with key: not-
specified, " +
                    $"value: {value}, partition-id: not-specified");
                p.Produce(topic, new Message<Null, string> { Value = value
            });
        }

        // Wait up to 10 seconds for any in-flight messages to be sent.
        p.Flush(TimeSpan.FromSeconds(10));
    }
    catch (ProduceException<Null, string> e)
```

```

        {
            Console.WriteLine($"Delivery failed with error:
{e.Error.Reason}");
        }
    }
}

public static void RunConsumer(string broker, string connectionString,
string consumerGroup, string topic)
{
    var consumerConfig = new ConsumerConfig
    {
        BootstrapServers = broker,
        SecurityProtocol = SecurityProtocol.SaslSsl,
        SocketTimeoutMs = 60000,
        SessionTimeoutMs = 30000,
        SaslMechanism = SaslMechanism.Plain,
        SaslUsername = "$ConnectionString",
        SaslPassword = connectionString,
        GroupId = consumerGroup,
        AutoOffsetReset = AutoOffsetReset.Earliest
    };

    using (var c = new ConsumerBuilder<string, string>
(consumerConfig).Build())
    {
        c.Subscribe(topic);

        CancellationTokenSource cts = new CancellationTokenSource();
        Console.CancelKeyPress += (_, e) =>
        {
            e.Cancel = true;
            cts.Cancel();
        };

        try
        {
            while (true)
            {
                try
                {
                    var message = c.Consume(cts.Token);
                    Console.WriteLine($"Consumed - key:
{message.Message.Key}, "+
"$value: {message.Message.Value}, " +
"$partition-id: {message.Partition}, " +
"$offset: {message.Offset}");
                }
                catch (ConsumeException e)
                {
                    Console.WriteLine($"Error occured: {e.Error.Reason}");
                }
            }
        }
        catch(OperationCanceledException)
    }
}

```

```

    {
        // Close the consumer to ensure that it leaves the group cleanly
        // and that final offsets are committed.
        c.Close();
    }
}

```

This code example produces the following results:

```

Initializing Producer
Sending message with key: not-specified, value: message-0, partition-id: not-specified
Sending message with key: not-specified, value: message-1, partition-id: not-specified
Sending message with key: not-specified, value: message-2, partition-id: not-specified
Sending message with key: not-specified, value: message-3, partition-id: not-specified
Sending message with key: not-specified, value: message-4, partition-id: not-specified
Sending message with key: not-specified, value: message-5, partition-id: not-specified
Sending message with key: not-specified, value: message-6, partition-id: not-specified
Sending message with key: not-specified, value: message-7, partition-id: not-specified
Sending message with key: not-specified, value: message-8, partition-id: not-specified
Sending message with key: not-specified, value: message-9, partition-id: not-specified

Initializing Consumer
Consumed - key: , value: message-1, partition-id: [0], offset: 4
Consumed - key: , value: message-2, partition-id: [0], offset: 5
Consumed - key: , value: message-4, partition-id: [0], offset: 6
Consumed - key: , value: message-9, partition-id: [0], offset: 7
Consumed - key: , value: message-3, partition-id: [2], offset: 2
Consumed - key: , value: message-8, partition-id: [2], offset: 3
Consumed - key: , value: message-0, partition-id: [3], offset: 3
Consumed - key: , value: message-6, partition-id: [3], offset: 4
Consumed - key: , value: message-7, partition-id: [3], offset: 5
Consumed - key: , value: message-5, partition-id: [1], offset: 1

```

In this case, the topic has four partitions. The following events took place:

- The producer sent 10 messages, each without a partition key.
- The messages arrived at partitions in a random order.
- A single consumer listened to all four partitions and received the messages out of order.

If the code had used two instances of the consumer, each instance would have subscribed to two of the four partitions.

Distribute to a specific partition

This example involves error messages. Suppose certain applications need to process error messages, but all other messages can go to a common consumer. In this case, the producer sends error messages to a specific partition. Consumers who want to receive error messages listen to that partition. The following code shows how to implement this scenario:

C#

```

// Producer code.
var topicPartition = new TopicPartition(topic, partition);

...
p.Produce(topicPartition, new Message<Null, string> { Value = value });

// Consumer code.
// Subscribe to one partition.
c.Assign(new TopicPartition(topic, partition));

// Use this code to subscribe to a list of partitions.
c.Assign(new List<TopicPartition> {
    new TopicPartition(topic, partition1),
    new TopicPartition(topic, partition2)
});
```

As these results show, the producer sent all messages to partition 2, and the consumer only read messages from partition 2:

```

Initializing Producer
Sending message with key: not-specified, value: message-0, partition-id: 2
Sending message with key: not-specified, value: message-1, partition-id: 2
Sending message with key: not-specified, value: message-2, partition-id: 2
Sending message with key: not-specified, value: message-3, partition-id: 2
Sending message with key: not-specified, value: message-4, partition-id: 2
Sending message with key: not-specified, value: message-5, partition-id: 2
Sending message with key: not-specified, value: message-6, partition-id: 2
Sending message with key: not-specified, value: message-7, partition-id: 2
Sending message with key: not-specified, value: message-8, partition-id: 2
Sending message with key: not-specified, value: message-9, partition-id: 2

Initializing Consumer
Consumed - key: , value: message-0, partition-id: [2], offset: 4
Consumed - key: , value: message-1, partition-id: [2], offset: 5
Consumed - key: , value: message-2, partition-id: [2], offset: 6
Consumed - key: , value: message-3, partition-id: [2], offset: 7
Consumed - key: , value: message-4, partition-id: [2], offset: 8
Consumed - key: , value: message-5, partition-id: [2], offset: 9
Consumed - key: , value: message-6, partition-id: [2], offset: 10
Consumed - key: , value: message-7, partition-id: [2], offset: 11
Consumed - key: , value: message-8, partition-id: [2], offset: 12
Consumed - key: , value: message-9, partition-id: [2], offset: 13
```

In this scenario, if you add another consumer instance to listen to this topic, the pipeline won't assign any partitions to it. The new consumer will starve until the existing consumer shuts down. The pipeline will then assign a different, active consumer to read from the partition. But the pipeline will only make that assignment if the new consumer isn't dedicated to another partition.

Preserve event order

This example involves bank transactions that a consumer needs to process in order. In this scenario, you can use the customer ID of each event as the key. For the event value, use the details of the transaction. The following code shows a simplified implementation of this case:

C#

```
Producer code
// This code assigns the key an integer value. You can also assign it any
other valid key value.
using (var p = new ProducerBuilder<int, string>(producerConfig).Build())
...
p.Produce(topic, new Message<int, string> { Key = i % 2, Value = value });
```

This code produces the following results:

```
Initializing Producer
Sending message with key: 1, value: message-1, partition-id: not-specified
Sending message with key: 0, value: message-2, partition-id: not-specified
Sending message with key: 1, value: message-3, partition-id: not-specified
Sending message with key: 0, value: message-4, partition-id: not-specified
Sending message with key: 1, value: message-5, partition-id: not-specified
Sending message with key: 0, value: message-6, partition-id: not-specified
Sending message with key: 1, value: message-7, partition-id: not-specified
Sending message with key: 0, value: message-8, partition-id: not-specified
Sending message with key: 1, value: message-9, partition-id: not-specified
Sending message with key: 0, value: message-10, partition-id: not-specified

Initializing Consumer
Consumed - key: 0, value: message-2, partition-id: [0], offset: 31
Consumed - key: 0, value: message-4, partition-id: [0], offset: 32
Consumed - key: 0, value: message-6, partition-id: [0], offset: 33
Consumed - key: 0, value: message-8, partition-id: [0], offset: 34
Consumed - key: 0, value: message-10, partition-id: [0], offset: 35
Consumed - key: 1, value: message-1, partition-id: [2], offset: 38
Consumed - key: 1, value: message-3, partition-id: [2], offset: 39
Consumed - key: 1, value: message-5, partition-id: [2], offset: 40
Consumed - key: 1, value: message-7, partition-id: [2], offset: 41
Consumed - key: 1, value: message-9, partition-id: [2], offset: 42
```

As these results show, the producer only used two unique keys. The messages then went to only two partitions instead of all four. The pipeline guarantees that messages with the same key go to the same partition.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Rajasa Savant](#) | Senior Software Engineer

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- Use Azure Event Hubs from Apache Kafka applications
- Apache Kafka developer guide for Azure Event Hubs
- Quickstart: Data streaming with Event Hubs using the Kafka protocol
- Send events to and receive events from Azure Event Hubs - .NET (`Azure.Messaging.EventHubs`)
- Balance partition load across multiple instances of your application
- Dynamically add partitions to an event hub (Apache Kafka topic) in Azure Event Hubs
- Availability and consistency in Event Hubs
- Azure Event Hubs Event Processor client library for .NET
- Effective strategies for Kafka topic partitioning
- Confluent blog post: How to choose the number of topics/partitions in a Kafka cluster?

Related resources

- Integrate Event Hubs with serverless functions on Azure
- Performance and scale for Event Hubs and Azure Functions
- Event-driven architecture style
- Stream processing with fully managed open-source data engines
- Publisher-Subscriber pattern
- Apache open-source scenarios on Azure - Apache Kafka

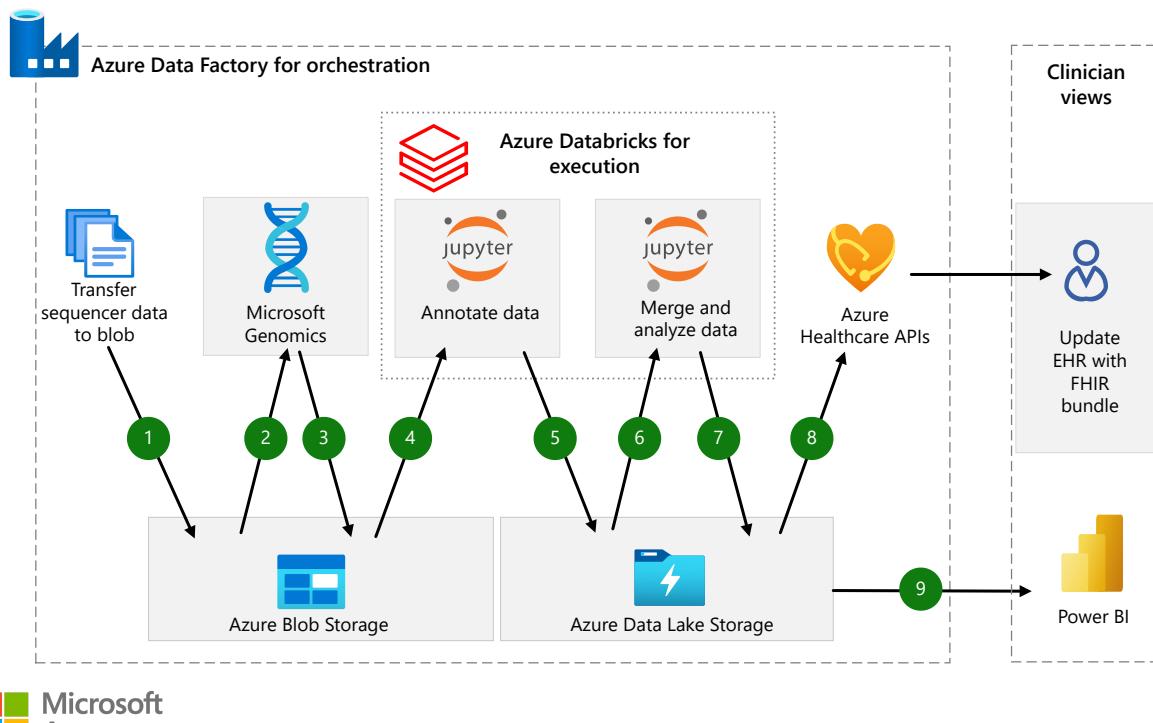
Precision medicine pipeline with genomics

Azure Blob Storage Azure Data Factory Azure Data Lake Storage Azure Databricks

Azure Microsoft Genomics

This article presents a solution for genomic analysis and reporting. The processes and results are appropriate for [precision medicine](#) scenarios, or areas of medical care that use genetic profiling.

Architecture



[Download a Visio file](#) of this architecture.

Workflow

Azure Data Factory orchestrates the workflow:

1. Data Factory transfers the initial sample file to Azure Blob Storage. The file is in FASTQ format.
2. Microsoft Genomics runs secondary analysis on the file.

3. Microsoft Genomics stores the output in Blob Storage in one of these formats:

- Variant call format (VCF)
- Genomic VCF (GVCF)

4. Jupyter Notebook annotates the output file. The notebook runs on Azure Databricks.

5. Azure Data Lake Storage stores the annotated file.

6. Jupyter Notebook merges the file with other datasets and analyzes the data. The notebook runs on Azure Databricks.

7. Data Lake Storage stores the processed data.

8. Azure Healthcare APIs packs the data into a Fast Healthcare Interoperability Resources (FHIR) bundle. The clinical data then enters the patient electronic health record (EHR).

9. Clinicians view the results in Power BI dashboards.

Components

The solution uses the following components:

Microsoft Genomics

[Microsoft Genomics](#) offers an efficient and accurate genomics pipeline that implements the industry's best practices. Its high-performance engine is optimized for these tasks:

- Reading large files of genomic data
- Processing them efficiently across many cores
- Sorting and filtering the results
- Writing the results to output files

To maximize throughput, this engine operates a Burrows-Wheeler Aligner (BWA) and a Genome Analysis Toolkit (GATK) HaplotypeCaller variant caller. The engine also uses several other components that make up standard genomics pipelines. Examples include duplicate marking, base quality score recalibration, and indexing. In a few hours, the engine can process a single genomic sample on a single multi-core server. The processing starts with raw reads. It produces aligned reads and variant calls.

Internally, the Microsoft Genomics controller manages these aspects of the process:

- Distributing batches of genomes across pools of machines in the cloud
- Maintaining a queue of incoming requests
- Distributing the requests to servers that run the genomics engine
- Monitoring the servers' performance and progress
- Evaluating the results
- Ensuring that processing runs reliably and securely at scale, behind a secure web service API

You can easily use Microsoft Genomics results in tertiary analysis and machine learning services. And because Microsoft Genomics is a cloud service, you don't need to manage or update hardware or software.

Other components

- [Data Factory](#) is an integration service that works with data from disparate data stores. You can use this fully managed, serverless platform to orchestrate and automate workflows. Specifically, [Data Factory pipelines](#) transfer data to Azure in this solution. A sequence of pipelines then triggers each step of the workflow.
- [Blob Storage](#) offers optimized cloud object storage for large amounts of unstructured data. In this scenario, Blob Storage provides the initial landing zone for the FASTQ file. This service also functions as the output target for the VCF and GVCF files that Microsoft Genomics generates. [Tiering functionality](#) in Blob Storage provides a way to archive FASTQ files in inexpensive long-term storage after processing.
- [Azure Databricks](#) is a data analytics platform. Its fully managed Spark clusters process large streams of data from various sources. In this solution, Azure Databricks provides the computational resources that Jupyter Notebook needs to annotate, merge, and analyze the data.
- [Data Lake Storage](#) is a scalable and secure data lake for high-performance analytics workloads. This service can manage multiple petabytes of information while sustaining hundreds of gigabits of throughput. The data may be structured, semi-structured, or unstructured. It typically comes from multiple, heterogeneous sources. In this architecture, Data Lake Storage provides the final landing zone for the annotated files and the merged datasets. It also gives downstream systems access to the final output.
- [Power BI](#) is a collection of software services and apps that display analytics information. You can use Power BI to connect and display unrelated sources of

data. In this solution, you can populate Power BI dashboards with the results. Clinicians can then create visuals from the final dataset.

- [Azure Healthcare APIs](#) is a managed, standards-based, compliant interface for accessing clinical health data. In this scenario, Azure Healthcare APIs passes an FHIR bundle to the EHR with the clinical data.

Scenario details

This article presents a solution for genomic analysis and reporting. The processes and results are appropriate for [precision medicine](#) scenarios, or areas of medical care that use genetic profiling. Specifically, the solution provides a clinical genomics workflow that automates these tasks:

- Taking data from a sequencer
- Moving the data through secondary analysis
- Providing results that clinicians can consume

The growing scale, complexity, and security requirements of genomics make it an ideal candidate for moving to the cloud. Consequently, the solution uses Azure services in addition to open-source tools. This approach takes advantage of the security, performance, and scalability features of the Azure cloud:

- Scientists plan on sequencing hundreds of thousands of genomes in coming years. The task of storing and analyzing this data requires significant computing power and storage capacity. With data centers around the world that provide these resources, Azure can meet these demands.
- Azure is certified for major global security and privacy standards, such as ISO 27001.
- Azure complies with the security and provenance standards that the Health Insurance Portability and Accountability Act (HIPAA) establishes for personal health information.

A key component of the solution is [Microsoft Genomics](#). This service offers an optimized secondary analysis implementation that can process a [30x genome](#) in a few hours. Standard technologies can take days.

Potential use cases

This solution is ideal for the healthcare industry. It applies to many areas:

- Risk scoring patients for cancer

- Identifying patients with genetic markers that predispose them to disease
- Generating patient cohorts for studies

Considerations

The following considerations align with the [Microsoft Azure Well-Architected Framework](#) and apply to this solution:

Availability

The service level agreements (SLAs) of most Azure components guarantee availability:

- [At least 99.9 percent of Data Factory pipelines are guaranteed to run successfully](#).
- The [Azure Databricks SLA guarantees 99.95 percent availability](#).
- [Microsoft Genomics offers a 99.99 percent availability SLA for workflow requests](#).
- Blob Storage and Data Lake Storage are part of Azure Storage, which offers [availability through redundancy](#).

Scalability

Most Azure services are scalable by design:

- [Data Factory transforms data at scale](#).
- The clusters in [Azure Databricks resize as needed](#).
- For information on optimizing scalability in Blob Storage, see [Performance and scalability checklist for Blob Storage](#).
- [Data Lake Storage can manage exabytes of data](#).
- [Microsoft Genomics runs exabyte-scale workloads](#).

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](#).

The technologies in this solution meet most companies' requirements for security.

Guidelines

Because of the sensitive nature of medical data, establish governance and security by following the guidelines in these documents:

- Security in the Microsoft Cloud Adoption Framework for Azure
- Practical guide to designing secure health solutions using Microsoft Azure ↗
- Enterprise-scale landing zones

Regulatory compliance

- See these documents for information on complying with HIPAA and the Health Information Technology for Economic and Clinical Health (HITECH) Act:
 - [HIPAA - Azure Compliance](#)
 - [Health Insurance Portability and Accountability Act \(HIPAA\) & Health Information Technology for Economic and Clinical Health \(HITECH\) Act](#)
- Components of this solution are in scope for HIPAA according to [Microsoft Azure Compliance Offerings](#) ↗. If you substitute any other components, validate them first against the list in that document's appendix.

General security features

Several components also secure data in other ways:

- [Data Factory](#) encrypts data that it transfers. It also uses Azure Key Vault or certificates to encrypt credentials.
- [Azure Databricks](#) provides many tools for securing network infrastructure and data. Examples include [access control lists](#), [secrets](#), and [no public IP \(NPIP\)](#).
- [Blob storage](#) supports [storage service encryption \(SSE\)](#), which automatically encrypts data before storing it. It also provides [many other ways to protect data and networks](#).
- [Data Lake Storage](#) provides [access control](#). Its model supports these types of controls:
 - Azure role-based access control (RBAC)
 - Portable Operating System Interface (POSIX) access control lists (ACLs)

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

With most Azure services, you can reduce costs by only paying for what you use:

- With [Data Factory](#), your activity run volume determines the cost .
- [Azure Databricks](#) offers many tiers, workloads, and pricing plans to help you minimize costs.
- [Blob Storage](#) costs depend on data redundancy options and volume .
- With [Data Lake Storage](#), pricing depends on many factors: your namespace type, storage capacity, and choice of tier .
- For [Microsoft Genomics](#), the charge depends on the number of gigabases that each workflow processes .

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Wylie Graham](#)  | Senior Program Manager
- [Matt Hansen](#)  | Senior Cloud Solution Architect

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Microsoft Genomics: Common questions](#)
- [Genomics quickstart starter kit](#) 
- [Burrows-Wheeler Aligner](#) 
- [Genome Analysis Toolkit](#) 

Related resources

Fully deployable architectures:

Data Factory solutions

- [Automated enterprise BI](#)
- [\[Hybrid ETL with Azure Data Factory\]](#)  [\[Hybrid ETL with Azure Data Factory\]](#)
- [Replicate and sync mainframe data in Azure](#)

Analytics solutions

- Data warehousing and analytics
- Geospatial data processing and analytics
- Stream processing with Azure Databricks

Healthcare solutions

- Clinical insights with Microsoft Cloud for Healthcare
- Health data consortium on Azure
- Population health management for healthcare

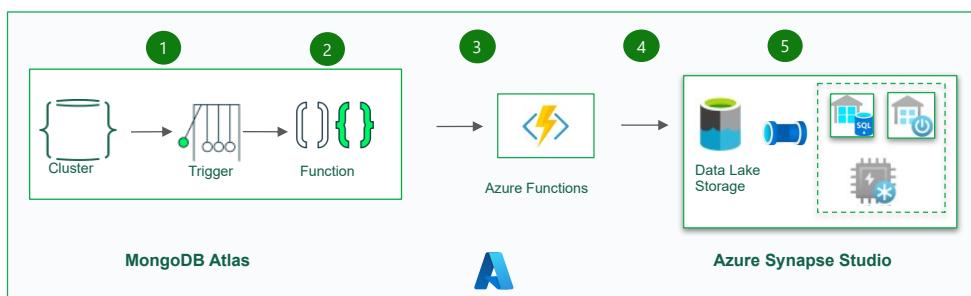
Enable real-time sync of MongoDB Atlas data changes to Azure Synapse Analytics

Azure Synapse Analytics

Real-time analytics can help you make quick decisions and perform automated actions based on current insights. It can also help you deliver enhanced customer experiences. This solution describes how to keep Azure Synapse Analytics data pools in sync with operational data changes in MongoDB.

Architecture

The following diagram shows how to implement real-time sync from Atlas to Azure Synapse Analytics. This simple flow ensures that any changes that occur in the MongoDB Atlas collection are replicated to the default Azure Data Lake Storage repository in the Azure Synapse Analytics workspace. After the data is in Data Lake Storage, you can use Azure Synapse Analytics pipelines to push the data to dedicated SQL pools, Spark pools, or other solutions, depending on your analytics requirements.



Download a [PowerPoint file](#) of this architecture.

Dataflow

Real-time changes in the MongoDB Atlas operational data store (ODS) are captured and made available to Data Lake Storage in an Azure Synapse Analytics workspace for real-time analytics use cases, live reports, and dashboards.

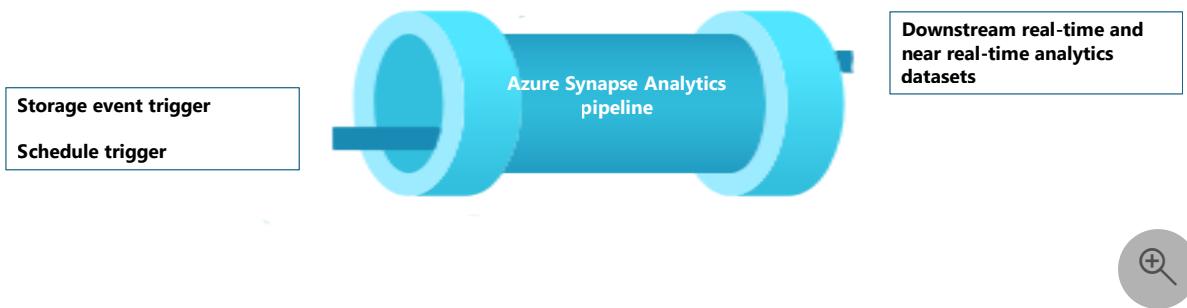
1. Data changes in the MongoDB Atlas operational/transactional datastore are captured by [Atlas triggers](#).
2. When an [Atlas database trigger](#) observes an event, it passes the change type and the document that's changed (full or delta) to an [Atlas function](#).
3. The Atlas function triggers an Azure function, passing the change event and a JSON document.
4. Azure Functions uses the Azure Storage Files Data Lake client library to write the changed document to the configured Data Lake Storage in the Azure Synapse Analytics workspace.
5. After the data is in Data Lake Storage, it can be sent to dedicated SQL pools, Spark pools, and other solutions. Alternatively, you can convert the data from JSON to Parquet or Delta formats by using Azure Synapse Analytics data flows or Copy pipelines to run additional BI reporting or AI / machine learning on the current data.

Components

- [MongoDB Atlas change streams](#) enable you to notify applications of changes to a collection, database, or deployment cluster. Change streams give applications access to real-time data changes and enable them to immediately react to changes. This functionality is critical in use cases like IoT event tracking and financial data changes, where alarms need to be raised and responsive actions need to be taken immediately. Atlas triggers use change streams to monitor collections for changes and automatically invoke the associated Atlas function in response to the trigger event.
- [Atlas triggers](#) respond to document inserts, updates, and deletes in a specific collection and can automatically invoke an Atlas function in response to the change event.
- [Atlas functions](#) are serverless, server-side JavaScript code implementations that can perform actions based on the events that invoke an Atlas trigger. Combining Atlas triggers with Atlas functions simplifies the implementation of event-driven architectures.
- [Azure Functions](#) is an event-driven, serverless compute platform that you can use to develop applications efficiently with the programming language of your

choice. You can also use it to connect seamlessly with other Azure services. In this scenario, an Azure function captures a change event and uses it to write a blob containing the changed data into Data Lake Storage by using the Azure Storage Files Data Lake client library.

- [Data Lake Storage](#) is the default storage solution in Azure Synapse Analytics. You can use serverless pools to query the data directly.
- [Pipelines](#) and [data flows](#) in [Azure Synapse Analytics](#) can be used to push the blob that contains the MongoDB changed data to dedicated SQL pools or Spark pools for further analysis. Pipelines enable you to act on changed datasets in Data Lake Storage by using both [storage event triggers](#) and [scheduled triggers](#) to build solutions for both real-time and near real-time use cases. This integration accelerates downstream consumption of change datasets.



Alternatives

This solution uses Atlas triggers to wrap the code for listening to Atlas change streams and triggering Azure Functions in response to the change event. It's therefore much easier to implement than the previously provided [alternative solution](#). For that solution, you need to write code to listen to change streams in an [Azure App Service](#) app.

Another alternative is to use the [MongoDB Spark Connector](#) to read MongoDB stream data and write it to Delta tables. The code is run continuously in a Spark Notebook that's part of a pipeline in Azure Synapse Analytics. For more information on implementing this solution, see [Sync from Atlas to Azure Synapse Analytics using Spark streaming](#).

However, using Atlas triggers with Azure Functions provides a completely serverless solution. Because it's serverless, the solution provides robust scalability and cost optimization. Pricing is based on a pay-as-you-go cost model. You can save more money by using the Atlas function to combine a few change events before invoking the Azure Functions endpoint. This strategy can be useful in heavy-traffic scenarios.

Also, Microsoft Fabric [unifies](#) your data estate and makes it easier to run analytics and AI over the data, so you get insights quickly. Azure Synapse Analytics data engineering, data science, data warehousing, and real-time analytics in Fabric can now make better use of MongoDB data that's pushed to OneLake. You can use both Dataflow Gen2 and data pipeline connectors for Atlas to load Atlas data directly to OneLake. This no-code mechanism provides a powerful way to ingest data from Atlas to OneLake.



In Fabric, you can directly reference data that's pushed to Data Lake Storage by using [OneLake shortcuts](#), without any ETL.

You can push the data to Power BI to create reports and visualizations for BI reporting.

Scenario details

MongoDB Atlas, the operational data layer of many enterprise applications, stores data from internal applications, customer-facing services, and third-party APIs from multiple channels. You can use the data pipelines in Azure Synapse Analytics to combine this data with relational data from other traditional applications and with unstructured data from sources like logs, object stores, and clickstreams.

Enterprises use MongoDB capabilities like [aggregations](#), [analytical nodes](#), [Atlas Search](#), [Vector Search](#), [Atlas Data Lake](#), [Atlas SQL Interface](#), [Data Federation](#), and [Charts](#) to enable application-driven intelligence. However, the transactional data in MongoDB is extracted, transformed, and loaded to Azure Synapse Analytics dedicated SQL pools or Spark pools for batch, AI / machine learning, and data-warehouse BI analytics and intelligence.

There are two scenarios for data movement between Atlas and Azure Synapse Analytics: batch integration and real-time sync.

Batch integration

You can use batch and micro-batch integration to move data from Atlas to Data Lake Storage in Azure Synapse Analytics. You can fetch the entire historical data at once or

fetch incremental data based on filter criteria.

MongoDB on-premises instances and MongoDB Atlas can be integrated as a source or a sink resource in Azure Synapse Analytics. For information about the connectors, see [Copy data from or to MongoDB](#) or [Copy data from or to MongoDB Atlas](#).

The source connector makes it convenient to run Azure Synapse Analytics on operational data that's stored in on-premises MongoDB and/or in Atlas. You can fetch data from Atlas by using the source connector and load the data to Data Lake Storage in Parquet, Avro, JSON, and text formats or as CSV blob storage. These files can then be transformed or joined with other files from other data sources in multi-database, multicloud, or hybrid cloud scenarios. This use case is common in enterprise data warehouse (EDW) and analytics-at-scale scenarios. You can also use the sink connector to store the results of the analytics back in Atlas. For more information about batch integration, see [Analyze operational data on MongoDB Atlas using Azure Synapse Analytics](#).

Real-time sync

The architecture described in this article can help you implement real-time sync to keep your Azure Synapse Analytics storage current with MongoDB's operational data.

This solution is composed of two primary functions:

- Capturing the changes in Atlas
- Triggering the Azure function to propagate the changes to Azure Synapse Analytics

Capture the changes in Atlas

You can capture the changes by using an Atlas trigger, which you can configure in the [Add Trigger UI](#) or by using the [Atlas App Services Admin API](#). Triggers listen for database changes caused by database events like inserts, updates, and deletes. Atlas triggers also trigger an Atlas function when a change event is detected. You can use the [Add Trigger UI](#) to add the function. You can also create an Atlas function and associate it as the trigger invocation endpoint by using the [Atlas Admin API](#).

The following screenshot shows the form that you can use to create and edit an Atlas trigger. In the **Trigger Source Details** section, you specify the collection that the trigger watches for change events and the database events it watches for (insert, update, delete, and/or replace).

The trigger can invoke an Atlas function in response to the event that it's enabled for. The following screenshot shows the simple JavaScript code, added as an Atlas function, to invoke in response to the database trigger. The Atlas function invokes an Azure function, passing it the metadata of the change event together with the document that was inserted, updated, deleted, or replaced, depending on what the trigger is enabled for.

Atlas function code

The Atlas function code triggers the Azure function that's associated with the Azure function endpoint by passing the entire `changeEvent` in the body of the request to the

Azure function.

You need to replace the `<Azure function URL endpoint>` placeholder with the actual Azure function URL endpoint.

```
exports = function(changeEvent) {  
  
    // Invoke Azure function that inserts the change stream into Data Lake Storage.  
    console.log(typeof fullDocument);  
    const response = context.http.post({  
        url: "<Azure function URL endpoint>",  
        body: changeEvent,  
        encodeBodyAsJSON: true  
    });  
    return response;  
};
```

Trigger the Azure function to propagate the changes to Azure Synapse Analytics

The Atlas function is coded to invoke an Azure function that writes the change document to Data Lake Storage in Azure Synapse Analytics. The Azure function uses the [Azure Data Lake Storage client library for Python](#) SDK to create an instance of the `DataLakeServiceClient` class that represents your storage account.

The Azure function uses a storage key for authentication. You can also use Microsoft Entra ID OAuth implementations. The `storage_account_key` and other attributes related to Data Lake Storage are fetched from the configured OS environment variables. After the request body is decoded, the `fullDocument` (the entire inserted or updated document) is parsed from the request body and then written to Data Lake Storage by the Data Lake client functions `append_data` and `flush_data`.

For a delete operation, `fullDocumentBeforeChange` is used instead of `fullDocument`. `fullDocument` doesn't have any value in a delete operation, so the code fetches the document that was deleted, which is captured in `fullDocumentBeforeChange`. Note that `fullDocumentBeforeChange` is only populated when the **Document Preimage** setting is set to on, as shown in the previous screenshot.

```
import json  
import logging
```

```

import os
import azure.functions as func
from azure.storage.filedatalake import DataLakeServiceClient

def main(req: func.HttpRequest) -> func.HttpResponse:
    logging.info('Python HTTP trigger function processed a new request.')
    logging.info(req)
    storage_account_name = os.environ["storage_account_name"]
    storage_account_key = os.environ["storage_account_key"]
    storage_container = os.environ["storage_container"]
    storage_directory = os.environ["storage_directory"]
    storage_file_name = os.environ["storage_file_name"]
    service_client = DataLakeServiceClient(account_url="{}://{}.dfs.core.windows.net".format(
        "https", storage_account_name), credential=storage_account_key)
    json_data = req.get_body()
    logging.info(json_data)
    object_id = "test"
    try:
        json_string = json_data.decode("utf-8")
        json_object = json.loads(json_string)

        if json_object["operationType"] == "delete":
            object_id = json_object["fullDocumentBeforeChange"]["_id"]["$oid"]
            data = {"operationType": json_object["operationType"],
"data":json_object["fullDocumentBeforeChange"]}
        else:
            object_id = json_object["fullDocument"]["_id"]["$oid"]
            data = {"operationType": json_object["operationType"],
"data":json_object["fullDocument"]}

        logging.info(object_id)
        encoded_data = json.dumps(data)
    except Exception as e:
        logging.info("Exception occurred : " + str(e))

    file_system_client =
    service_client.get_file_system_client(file_system=storage_container)
    directory_client =
    file_system_client.get_directory_client(storage_directory)
    file_client = directory_client.create_file(storage_file_name + " - " +
    str(object_id) + ".txt")
    file_client.append_data(data=encoded_data, offset=0,
    length=len(encoded_data))
    file_client.flush_data(len(encoded_data))
    return func.HttpResponse(f"This HTTP triggered function executed
    successfully.")

```

So far, you've seen how the Atlas trigger captures any change that occurs and passes it to an Azure function via an Atlas function, and that the Azure function writes the change document as a new file in Data Lake Storage in the Azure Synapse Analytics workspace.

After the file is added to Data Lake Storage, you can set up a [storage event trigger](#) to trigger a pipeline that can then write the change document to a dedicated SQL pool or to a Spark pool table. The pipeline can use the [Copy activity and transform the data by using a data flow](#). Alternatively, if your final target is a dedicated SQL pool, you can modify the Azure function to write directly to the dedicated SQL pool in Azure Synapse Analytics. For a SQL pool, get the ODBC [connection string](#) for the SQL pool connection. See [Use Python to query a database](#) for an example of Python code that you can use to query the SQL pool table by using the connection string. You can modify this code to use an Insert query to write to a dedicated SQL pool. There are configuration settings and roles that need to be assigned to enable the function to write to a dedicated SQL pool. Information about these settings and roles is outside the scope of this article.

If you want a near real-time solution and you don't need the data to be synchronized in real time, using scheduled pipeline runs might be a good option. You can set up scheduled triggers to trigger a pipeline with the Copy activity or a data flow, at a frequency that's at the near real-time frequency that your business can afford, to use the [MongoDB connector](#) to fetch the data from MongoDB that was inserted, updated, or deleted between the last scheduled run and the current run. The pipeline uses the MongoDB connector as source connector to fetch the delta data from MongoDB Atlas and push it to Data Lake Storage or Azure Synapse Analytics dedicated SQL pools, using these as sink connections. This solution uses a pull mechanism (as opposed to the main solution described in this article, which is a push mechanism) from MongoDB Atlas as changes occur in the MongoDB Atlas collection that the Atlas trigger is listening to.

Potential use cases

MongoDB and the Azure Synapse Analytics EDW and analytical services can serve numerous use cases:

Retail

- Building intelligence into product bundling and product promotion
- Implementing customer 360 and hyper-personalization
- Predicting stock depletion and optimizing supply-chain orders
- Implementing dynamic discount pricing and smart search in ecommerce

Banking and finance

- Customizing customer financial services
- Detecting and blocking fraudulent transactions

Telecommunications

- Optimizing next-generation networks
- Maximizing the value of edge networks

Automotive

- Optimizing the parameterization of connected vehicles
- Detecting anomalies in IoT communication in connected vehicles

Manufacturing

- Providing predictive maintenance for machinery
- Optimizing storage and inventory management

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is 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](#).

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](#).

Azure Functions is a serverless managed service, so the app resources and platform components are protected by enhanced security. However, we recommend that you use HTTPS protocol and the latest TLS versions. It's also a good practice to validate the input to ensure that it's a MongoDB change document. See [Securing Azure Functions](#) for security considerations for Azure Functions.

MongoDB Atlas is a managed database as a service, so MongoDB provides enhanced platform security. MongoDB provides multiple mechanisms to help ensure 360-degree security for stored data, including database access, network security, encryption at rest and in transit, and data sovereignty. See [MongoDB Atlas Security](#) for the MongoDB Atlas security whitepaper and other articles that can help you ensure that the data in MongoDB is secure throughout the data lifecycle.

Cost optimization

Cost optimization is about reducing unnecessary expenses and improving operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

To estimate the cost of Azure products and configurations, use the [Azure pricing calculator](#). Azure helps you avoid unnecessary costs by determining the correct number of resources to use, analyzing spending over time, and scaling to meet business needs without overspending. Azure functions incur costs only when they're invoked. However, depending on the volume of changes in MongoDB Atlas, you can evaluate using a batching mechanism in the Atlas function to store changes in another temporary collection and trigger the Azure function only if the batch exceeds a certain limit.

For information about Atlas clusters, see [5 Ways to Reduce Costs With MongoDB Atlas](#) and [Cluster Configuration Costs](#). The [MongoDB pricing page](#) can help you understand pricing options for MongoDB Atlas clusters and other offerings of the MongoDB Atlas developer data platform. [Atlas Data Federation](#) can be deployed in Azure and [supports Azure Blob Storage](#) (in preview). If you're considering using batching to optimize costs, consider writing to Blob Storage instead of a MongoDB temporary collection.

Performance efficiency

Performance efficiency is the ability of your workload to scale to meet the demands placed on it by users in an efficient manner. For more information, see [Performance efficiency pillar overview](#).

Atlas triggers and Azure functions are time-tested for performance and scalability. See [Performance and scale in Durable Functions \(Azure Functions\)](#) to understand performance and scalability considerations for Azure Functions. See [Scale On-Demand](#) for some considerations for enhancing the performance of your MongoDB Atlas instances. See [Best Practices Guide for MongoDB Performance](#) for best practices for MongoDB Atlas configuration.

Conclusion

MongoDB Atlas seamlessly integrates with Azure Synapse Analytics, enabling Atlas customers to easily use Atlas as a source or a sink for Azure Synapse Analytics. This solution enables you to use MongoDB operational data in real-time from Azure Synapse Analytics for complex analytics and AI inference.

Deploy this scenario

[Real-Time Sync from MongoDB Atlas to Azure Synapse Analytics](#)

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Diana Annie Jenosh](#) | Senior Solutions Architect - MongoDB Partners team
- [Venkatesh Shanbag](#) | Senior Solutions Architect - MongoDB Partners team

Other contributors:

- [Sunil Sabat](#) | Principal Program Manager - ADF team
- [Wee Hyong Tok](#) | Principal Director of PM - ADF team

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Creating a Synapse workspace](#)
- [MongoDB Atlas](#)

Related resources

- [Real-time analytics on big data architecture](#)
- [Real-time analytics on data with Azure Service Bus and Azure Data Explorer](#)

Build a relationship mesh dashboard on Azure

Azure Storage

Azure SQL Database

Azure Synapse Analytics

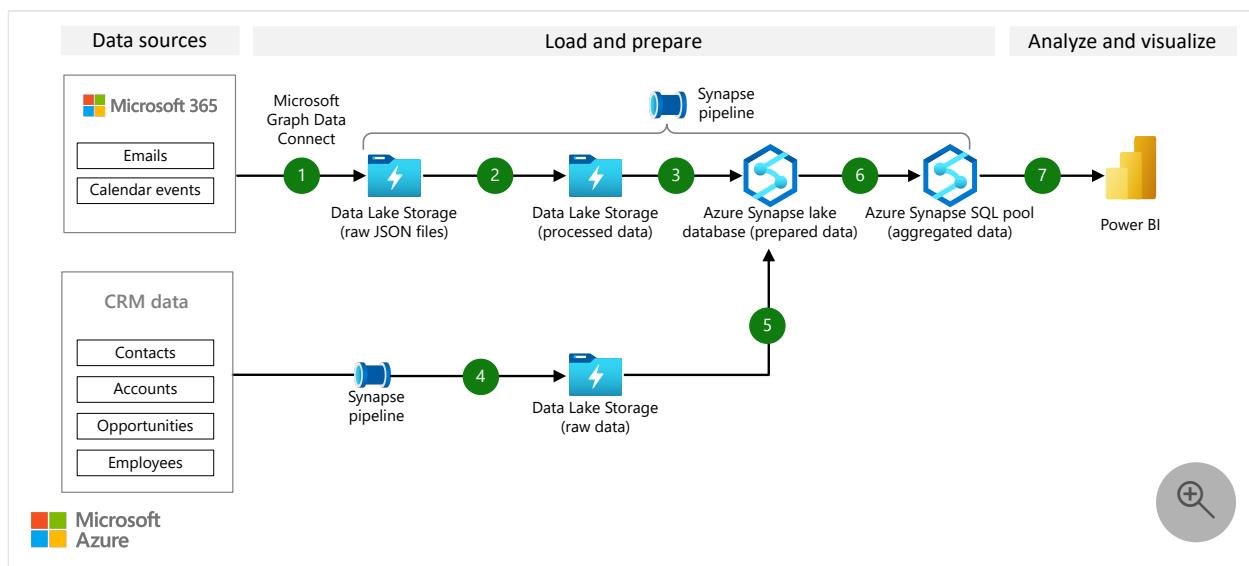
Power BI

Office 365

This solution brings data on the customer and seller relationship from various systems into a user-friendly dashboard. The dashboard provides actionable data about customer relationships at the level of both the seller account portfolio and individual account. You can use these insights to manage and improve interactions at different stages in a customer-engagement life cycle.

Apache®, 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

1. Microsoft Graph Data Connect and Azure Synapse Analytics pipelines bring in Microsoft 365 data, like emails and calendar events, to Azure Data Lake Storage.
2. The raw Microsoft 365 data is processed by using Azure Synapse pipelines and saved to Data Lake Storage.

3. Business logic for calculating relationship scores is applied on processed data and saved to a lake database in Azure Synapse.
4. The CRM data is ingested by using Azure Synapse pipelines into Data Lake Storage.
5. The raw CRM data is processed and saved to an Azure Synapse lake database.
6. The prepared Microsoft 365 data is joined with CRM data, aggregated, and saved to Azure Synapse SQL pool.
7. Power BI ingests the SQL data to visualize the relationship insights for sellers.

Components

- [Azure Synapse Analytics](#) is an analytics service that brings together data integration, enterprise data warehousing, and big data analytics. You'll use it for data ingestion, storage, and processing.
- [Azure Data Lake Storage](#) provides a scalable and secure data lake for your high-performance analytics workloads.
- [Microsoft Graph Data Connect](#) enables you to copy selected Microsoft 365 datasets into Azure data stores in a secure and scalable way.
- [Microsoft Power BI](#) can help you turn your data into coherent, visually immersive, and interactive insights. You'll use it to visualize customer profiles and metrics.

Scenario details

Building strong relationships with customers is key to improving customer retention, attracting loyalty, and increasing revenue. Companies rely heavily on platforms like customer relationship management (CRM) systems and Microsoft 365 to help them maintain customer relationships. This solution brings data on the customer and seller relationship from various systems into a user-friendly dashboard.

Potential use cases

You can use this solution to ingest data from Microsoft 365 and other systems to obtain actionable information on the relationship between sellers and their accounts. Then you can create plans to strengthen these relationships. Correlating this data provides insights on:

- The strength of multiple sellers' relationships within an account.
- The strength of a seller's relationship with individual external contacts within an account.

- The frequency and method of communication between seller and external contact for high-priority accounts.

Considerations

These considerations use the pillars of the Azure Well-Architected Framework. This framework is 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

The solution's resiliency depends on the failure modes of the individual services in the architecture. To learn more, follow the resiliency checklists in the following articles:

- [Azure Storage](#)
- [Azure Synapse Analytics](#)

Azure Blob Storage provides redundancy options that help ensure high availability. You can use either locally redundant storage (LRS) or availability zones. For more information, see [Durability and availability parameters](#).

For more information about the reliability pillar, see [Overview of the reliability pillar](#).

Security

This solution uses Microsoft Entra ID to authenticate users to the Azure solutions in the architecture. You can manage permissions via Microsoft Entra authentication or role-based access control. Follow the security guidelines in the following articles when you implement this solution:

- [Introduction to Azure security](#)
- [How to set up access control for your Azure Synapse workspace](#)
- [Microsoft Graph Data Connect granular data consent](#)
- [Microsoft Graph Data Connect data security and governance](#)

For more information about the security pillar, see [Overview of the operational excellence pillar](#).

Cost optimization

Use the [Azure pricing calculator](#) to estimate the cost of the service in this architecture.

[Microsoft Graph Data Connect](#) consumption charges are billed monthly on a pay-as-you-go basis. The price is based on the number of Microsoft Graph objects accessed.

[Azure Synapse Analytics](#) has various pricing options to help you optimize costs. You can perform big data processing tasks like data engineering, data preparation, and machine learning. These tasks are done directly in Azure Synapse by using memory-optimized or hardware-accelerated Apache Spark pools. Billing for usage of Spark pools is rounded up to the nearest minute.

There are various [Power BI](#) product options to meet different requirements. [Power BI Embedded](#) provides an Azure-based option for embedding Power BI functionality in your applications.

Azure services like Azure Storage accounts and Key Vaults that are deployed with Azure Synapse incur other costs.

For more information about the cost optimization pillar, see [Overview of the cost optimization pillar](#).

Performance efficiency

Performance efficiency, or scalability, is the ability of your workload to scale in an efficient manner to meet the demands that are placed on it by users.

Microsoft Graph Data Connect helps ingest data from Microsoft 365 tenants into Azure at scale. For more information, see [Access to data at scale](#).

This solution uses Apache Spark pool configurations in Azure Synapse, which can be scaled up and down automatically based on the activity needs of your workload. For more information, see [Autoscale](#).

For more information about the performance efficiency pillar, see [Overview of the performance efficiency pillar](#).

Deploy this scenario

Follow the steps in the [Getting Started guide](#) in GitHub to deploy this solution.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Nalini Chandhi](#) | Senior Technical Specialist
- [Malory Rose](#) | Technical Specialist (GBB)

Next steps

Review the information in this [relationship mesh solution GitHub repository](#) to determine whether you can benefit from this solution.

For more information, see these articles:

- [Azure Synapse Analytics](#)
- [Microsoft Graph Data Connect](#)
- [Microsoft Power BI](#)

Related resources

- [Analytics end-to-end with Azure Synapse](#)
- [Big data analytics with enterprise-grade security using Azure Synapse](#)
- [High throughput stream ingestion to Azure Synapse](#)

Secure a data lakehouse with Azure Synapse Analytics

Azure Synapse Analytics

Azure Data Lake Storage

Azure Virtual Network

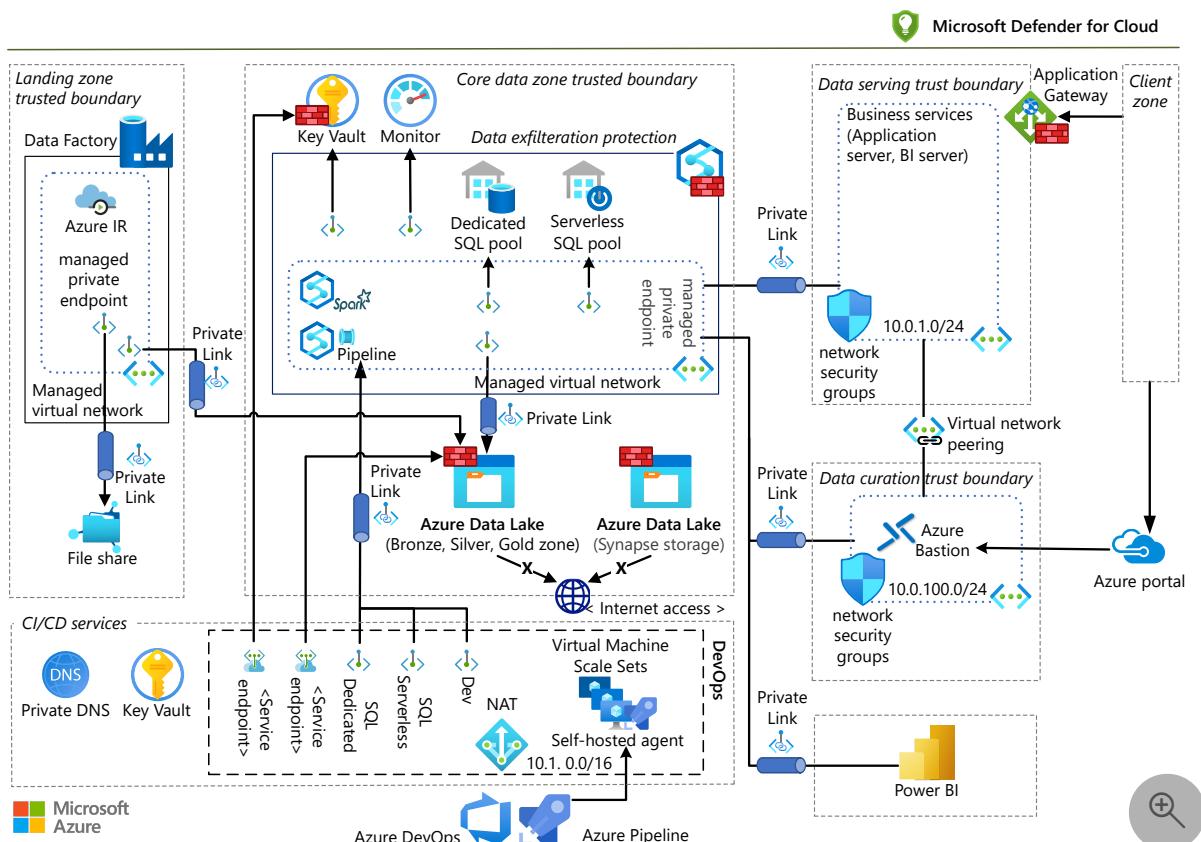
Power BI

This article describes the design process, principles, and technology choices for using Azure Synapse to build a secure data lakehouse solution. We focus on the security considerations and key technical decisions.

Apache®, 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

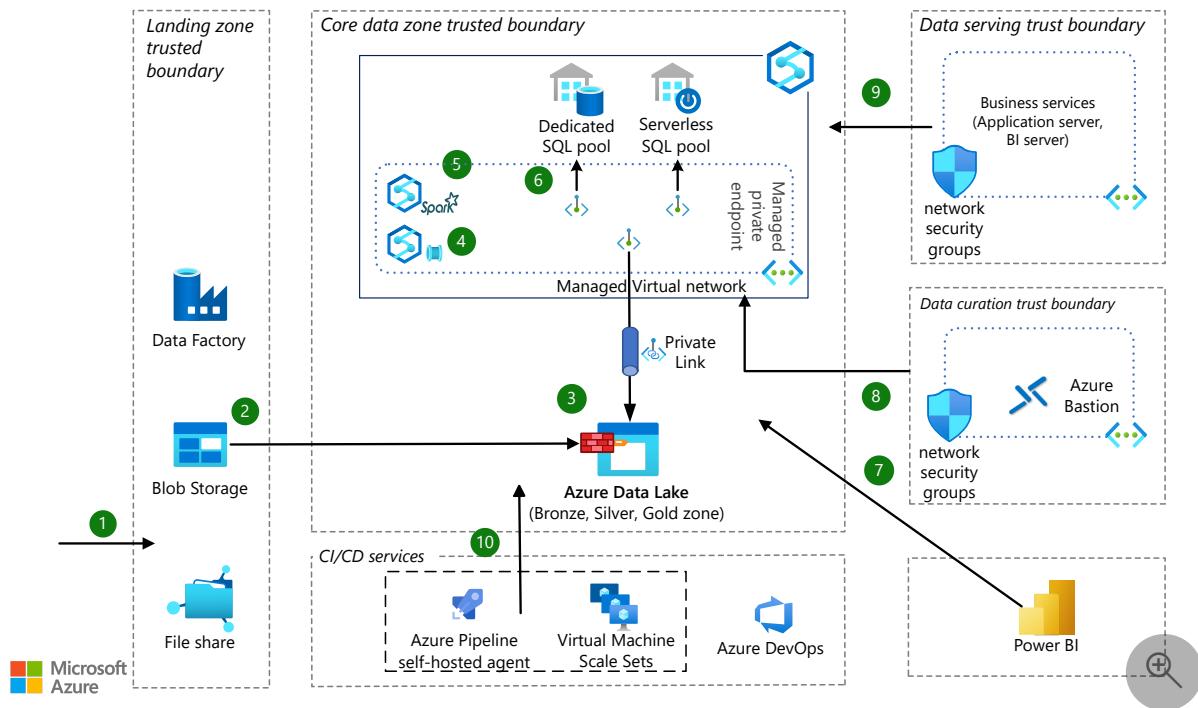
The following diagram shows the architecture of the data lakehouse solution. It's designed to control the interactions among the services in order to mitigate security threats. Solutions will vary depending on functional and security requirements.



Download a [Visio file](#) of this architecture.

Dataflow

The dataflow for the solution is shown in the following diagram:



1. Data is uploaded from the data source to the data landing zone, either to Azure Blob storage or to a file share that's provided by Azure Files. The data is uploaded by a batch uploader program or system. Streaming data is captured and stored in Blob Storage by using the Capture feature of Azure Event Hubs. There can be multiple data sources. For example, several different factories can upload their operations data. For information about securing access to Blob Storage, file shares, and other storage resources, see [Security recommendations for Blob Storage](#) and [Planning for an Azure Files deployment](#).
2. The arrival of the data file triggers Azure Data Factory to process the data and store it in the data lake in the core data zone. Uploading data to the core data zone in Azure Data Lake protects against data exfiltration.
3. Azure Data Lake stores the raw data that's obtained from different sources. It's protected by firewall rules and virtual networks. It blocks all connection attempts coming from the public internet.
4. The arrival of data in the data lake triggers the Azure Synapse pipeline, or a timed trigger runs a data processing job. Apache Spark in Azure Synapse is activated and runs a Spark job or notebook. It also orchestrates the data process flow in the data lakehouse. Azure Synapse pipelines convert data from [the Bronze zone to the Silver Zone and then to the Gold Zone](#).
5. A Spark job or notebook runs the data processing job. Data curation or a machine learning training job can also run in Spark. Structured data in the gold zone is stored in [Delta Lake](#) format.

6. A serverless SQL pool [creates external tables](#) that use the data stored in Delta Lake. The serverless SQL pool provides a powerful and efficient SQL query engine and can support traditional SQL user accounts or Microsoft Entra user accounts.
7. Power BI connects to the serverless SQL pool to visualize the data. It creates reports or dashboards using the data in the data lakehouse.
8. Data Analysts or scientists can log in to Azure Synapse Studio to:
 - Further enhance the data.
 - Analyze to gain business insight.
 - Train the machine learning model.
9. Business applications connect to a serverless SQL pool and use the data to support other business operation requirements.
10. Azure Pipelines runs the CI/CD process that automatically builds, tests, and deploys the solution. It's designed to minimize human intervention during the deployment process.

Components

The following are the key components in this data lakehouse solution:

- [Azure Synapse](#) ↗
- [Azure Files](#) ↗
- [Event Hubs](#) ↗
- [Blob Storage](#) ↗
- [Azure Data Lake Storage](#) ↗
- [Azure DevOps](#) ↗
- [Power BI](#) ↗
- [Data Factory](#) ↗
- [Azure Bastion](#) ↗
- [Azure Monitor](#) ↗
- [Microsoft Defender for Cloud](#) ↗
- [Azure Key Vault](#) ↗

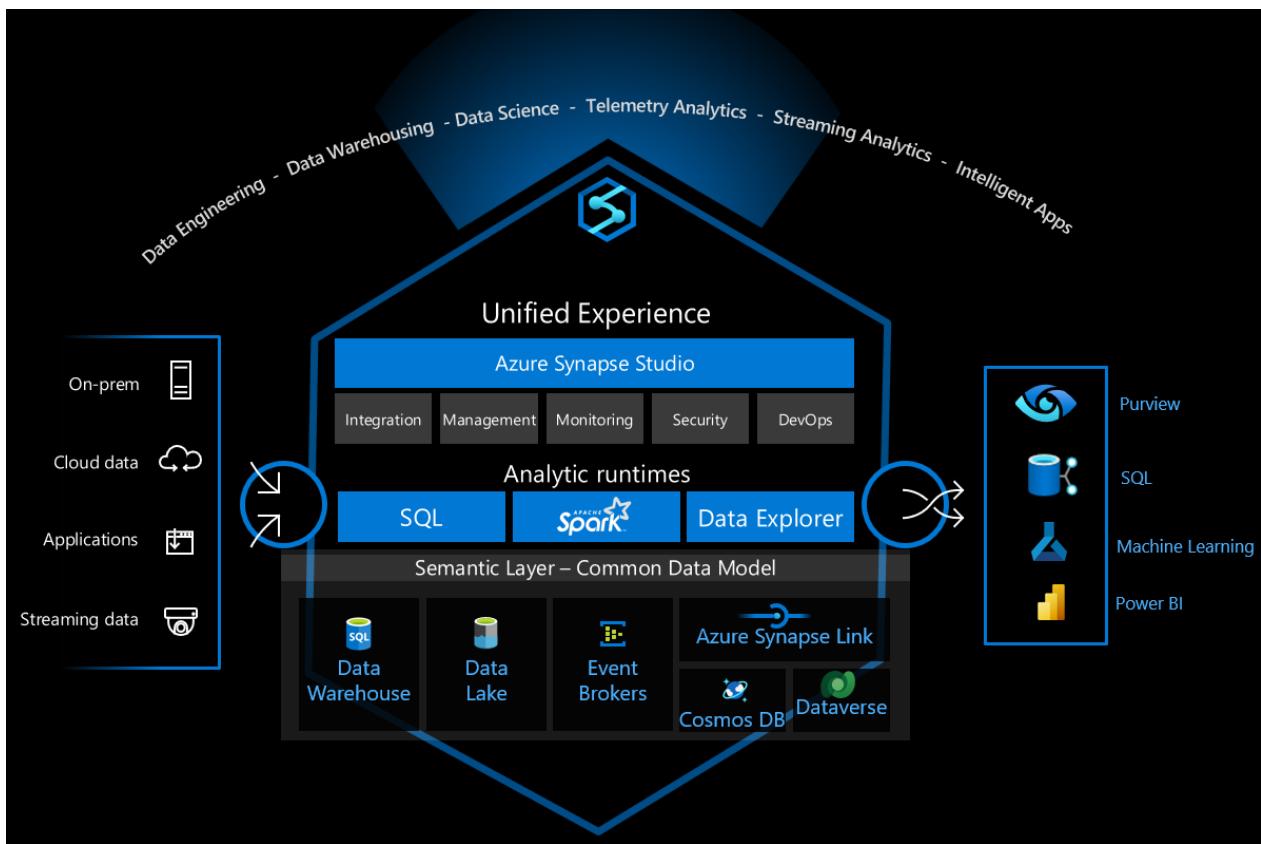
Alternatives

- If you need real-time data processing you can, instead of storing individual files on the data landing zone, use Apache Structured Streaming to receive the data stream from Event Hubs and process it.
- If the data has a complex structure and requires complex SQL queries, consider storing it in a dedicated SQL pool instead of a serverless SQL pool.
- If the data contains many hierarchical data structures—for example, it has a large JSON structure—you might want to store it in Azure Synapse Data Explorer.

Scenario details

Azure Synapse Analytics is a versatile data platform that supports enterprise data warehousing, real-time data analytics, pipelines, time-series data processing, machine learning, and data governance. To support these capabilities it integrates several different technologies, such as:

- Enterprise data warehousing
- Serverless SQL pools
- Apache Spark
- Pipelines
- Data Explorer
- Machine learning capabilities
- Purview unified data governance



These capabilities open up many possibilities, but there are many technical choices to make to securely configure the infrastructure for safe use.

This article describes the design process, principles, and technology choices for using Azure Synapse to build a secure data lakehouse solution. We focus on the security considerations and key technical decisions. The solution uses these Azure services:

- [Azure Synapse](#)
- [Azure Synapse serverless SQL pools](#)
- [Apache Spark in Azure Synapse Analytics](#)
- [Azure Synapse pipelines](#)

- Azure Data Lake
- Azure DevOps [↗](#).

The goal is to provide guidance on building a secure and cost-effective data lakehouse platform for enterprise use and on making the technologies work together seamlessly and securely.

Potential use cases

A data lakehouse is a modern data management architecture that combines the cost-efficiency, scale, and flexibility features of a data lake with the data and transaction management capabilities of a data warehouse. A data lakehouse can handle a vast amount of data and support business intelligence and machine learning scenarios. It can also process data from diverse data structures and data sources. For more information, see [What is the Databricks Lakehouse?](#).

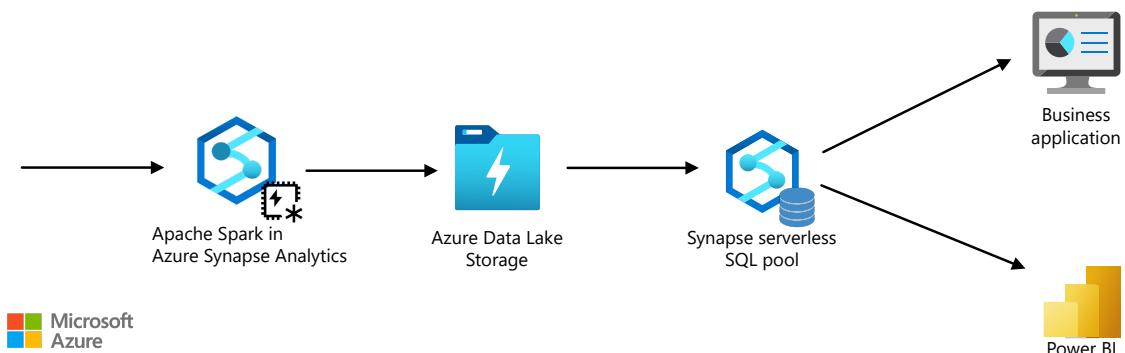
Some common use cases for the solution that's described here are:

- Analysis of Internet of Things (IoT) telemetry
- Automation of smart factories (for manufacturing)
- Tracking consumer activities and behavior (for retail)
- Managing security incidents and events
- Monitoring application logs and application behavior
- Processing and business analysis of semi-structured data

High-level design

This solution focuses on the security design and implementation practices in the architecture. Serverless SQL pool, Apache Spark in Azure Synapse, Azure Synapse pipelines, Data Lake Storage, and Power BI are the key services used to implement the [data lakehouse pattern \[↗\]\(#\)](#).

Here is the high-level solution design architecture:



Choose security focus

We started the security design by using the [Threat Modeling tool](#). The tool helped us:

- Communicate with system stakeholders about potential risks.
- Define the trust boundary in the system.

Based on the threat modeling results, we made the following security areas our top priorities:

- Identity and Access control
- Network protection
- DevOps security

We designed the security features and infrastructure changes to protect the system by mitigating the key security risks identified with these top priorities.

For details of what should be checked and considered, see:

- [Security in the Microsoft Cloud Adoption Framework for Azure](#)
- [Access control](#)
- [Asset protection](#)
- [Innovation security](#)

Network and asset protection plan

One of the key security principles in the Cloud Adoption Framework is the [Zero Trust principle](#): when designing security for any component or system, reduce the risk of attackers expanding their access by assuming that other resources in the organization are compromised.

Based on the threat modeling result, the solution adopts the [micro-segmentation deployment](#) recommendation in zero-trust and defines several [security boundaries](#). [Azure Virtual Network](#) and [Azure Synapse data exfiltration protection](#) are the key technologies that are used to implement the security boundary in order to protect data assets and critical components.

Because Azure Synapse is composed of several different technologies, we need to:

- Identify the components of Synapse and related services that are used in the project.

Azure Synapse is a versatile data platform that can handle many different data processing needs. First, we need to decide which components in Azure Synapse are used in the project so we can plan how to protect them. We also need to determine what other services communicate with these Azure Synapse components.

In the data lakehouse architecture, the key components are:

- Azure Synapse serverless SQL
- Apache Spark in Azure Synapse
- Azure Synapse pipelines
- Data Lake Storage
- Azure DevOps

- **Define the legal communication behaviors between the components.**

We need to define the allowed communication behaviors between the components.

For example, do we want the Spark engine to communicate with the dedicated SQL instance directly, or do we want it to communicate through a proxy such as Azure Synapse Data Integration pipeline or Data Lake Storage?

Based on the Zero Trust principle, we block communication if there's no business need for the interaction. For example, we block a Spark engine that's in an unknown tenant from directly communicating with Data Lake storage.

- **Choose the proper security solution to enforce the defined communication behaviors.**

In Azure, several security technologies can enforce the defined service communication behaviors. For example, in Data Lake Storage you can use an IP address allowlist to control access to a data lake, but you can also choose which virtual networks, Azure services, and resource instances are allowed. Each protection method provides different security protection. Choose based on business needs and environmental limitations. The configuration used in this solution is described in the next section.

- **Implement threat detection and advanced defenses for critical resources.**

For critical resources, it's best to implement threat detection and advanced defenses. The services help identify threats and trigger alerts, so the system can notify users about security breaches.

Consider the following techniques to better protect networks and assets:

- **Deploy perimeter networks to provide security zones for data pipelines**

When a data pipeline workload requires access to external data and the data landing zone, it's best to implement a perimeter network and separate it with an extract, transform, and load (ETL) pipeline.

- **Enable Defender for Cloud for all storage accounts**

Defender for Cloud triggers security alerts when it detects unusual and potentially harmful attempts to access or exploit storage accounts. For more information, see [Configure Microsoft Defender for Storage](#).

- Lock a storage account to prevent malicious deletion or configuration changes

For more information, see [Apply an Azure Resource Manager lock to a storage account](#).

Architecture with network and asset protection

The following table describes the defined communication behaviors and security technologies chosen for this solution. The choices were based on the methods discussed in [Network and asset protection plan](#).

[] [Expand table](#)

From (Client)	To (Service)	Behavior	Configuration	Notes
Internet	Data Lake Storage	Deny all	Firewall rule - Default deny	Default: 'Deny' Firewall rule - Default Deny
Azure Synapse Pipeline/Spark	Data Lake Storage	Allow (instance)	Virtual network - Managed private endpoint (Data Lake Storage)	
Synapse SQL	Data Lake Storage	Allow (instance)	Firewall rule - Resource instances (Synapse SQL)	Synapse SQL needs to access Data Lake Storage using managed identities
Azure Pipelines agent	Data Lake Storage	Allow (instance)	Firewall rule - Selected virtual networks Service endpoint - Storage	For integration testing bypass: 'AzureServices' (firewall rule)
Internet	Synapse workspace	Deny all	Firewall rule	

From (Client)	To (Service)	Behavior	Configuration	Notes
Azure Pipelines agent	Synapse workspace	Allow (instance)	Virtual network - private endpoint	Requires three private endpoints (Dev, serverless SQL, and dedicated SQL)
Synapse managed virtual network	Internet or unauthorized Azure tenant	Deny all	Virtual network - Synapse data exfiltration protection	
Synapse pipeline/Spark	Key Vault	Allow (instance)	Virtual network - Managed private endpoint (Key Vault)	Default: 'Deny'
Azure Pipelines agent	Key Vault	Allow (instance)	Firewall rule - Selected virtual networks * Service endpoint - Key Vault	bypass: 'AzureServices' (firewall rule)
Azure Functions	Synapse serverless SQL	Allow (instance)	Virtual network - Private endpoint (Synapse serverless SQL)	
Synapse pipeline/Spark	Azure Monitor	Allow (instance)	Virtual network - Private endpoint (Azure Monitor)	

For example, in the plan we want to:

- Create an Azure Synapse workspace with a managed virtual network.
- Secure data egress from Azure Synapse workspaces by using [Azure Synapse workspaces Data exfiltration protection](#).
- Manage the list of approved Microsoft Entra tenants for the Azure Synapse workspace.
- Configure network rules to grant traffic to the Storage account from selected virtual networks, access only, and disable public network access.
- Use [Managed Private Endpoints](#) to connect the virtual network that's managed by Azure Synapse to the data lake.
- Use [Resource Instance](#) to securely connect Azure Synapse SQL to the data lake.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is 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](#).

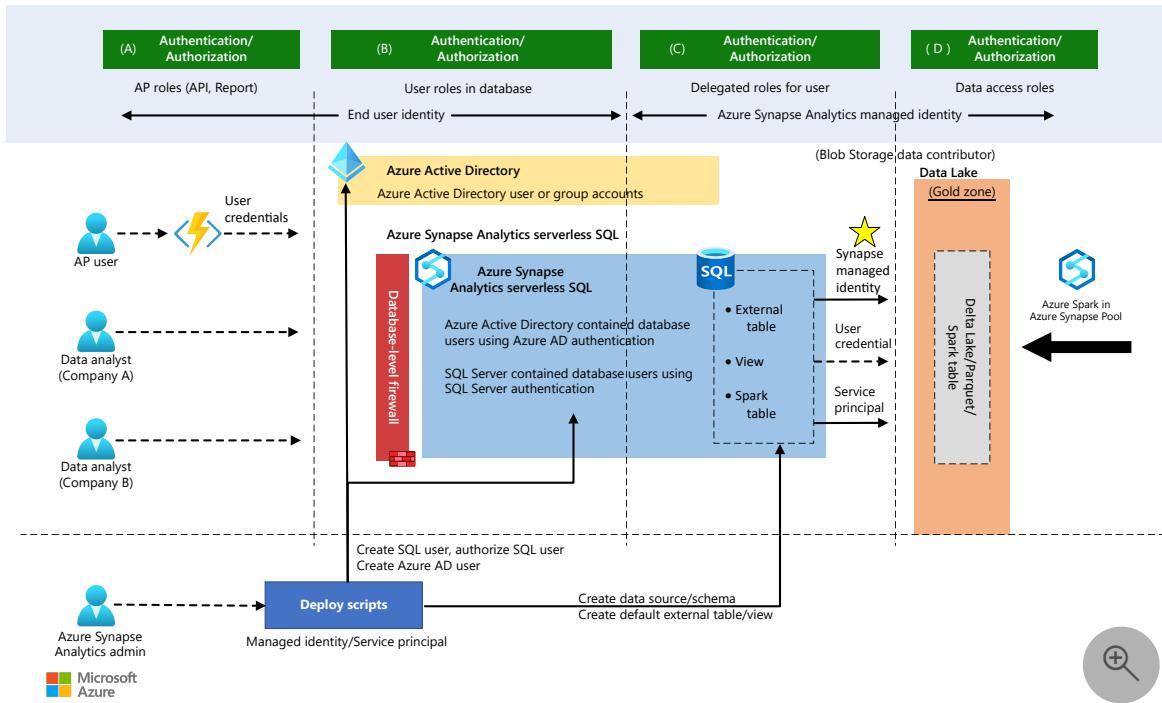
Security

For information about the security pillar of the Well-Architected Framework, see [Security](#).

Identity and access control

There are several components in the system. Each one requires a different identity and access management (IAM) configuration. These configurations need to collaborate to provide a streamlined user experience. Therefore, we use the following design guidance when we implement identity and access control.

- **Choose an identity solution for different access control layers**
 - There are four different identity solutions in the system.
 - SQL account (SQL Server)
 - Service principal (Microsoft Entra ID)
 - Managed identity (Microsoft Entra ID)
 - User Account (Microsoft Entra ID)
 - There are four different access control layers in the system.
 - The application access layer: choose the identity solution for AP Roles.
 - The Azure Synapse DB/Table access layer: choose the identity solution for roles in databases.
 - Azure Synapse access external resource layer: choose the identity solution to access external resources.
 - Data Lake Storage access layer: choose the identity solution to control file access in the storage.



A crucial part of identity and access control is choosing the right identity solution for each access control layer. The [security design principles](#) of the Azure Well-Architected Framework suggest using native controls and driving simplicity. Therefore, this solution uses the Microsoft Entra user Account of the end user in the application and Azure Synapse DB access layers. It leverages the native first-party IAM solutions and provides fine-grained access control. The Azure Synapse access external resource layer and Data Lake access layer use managed identity in Azure Synapse to simplify the authorization process.

- **Consider least-privileged access**

A Zero Trust guiding principle suggests providing just-in-time and just-enough access to critical resources. See [Microsoft Entra Privileged Identity Management \(PIM\)](#) to enhance security in the future.

- **Protect linked service**

Linked services define the connection information that's needed for a service to connect to external resources. It's important to secure linked services configurations.

- Create an [Azure Data Lake linked service with Private Link](#).
- Use [managed identity](#) as the authentication method in linked services.
- Use Azure Key Vault to secure the credentials for accessing the linked service.

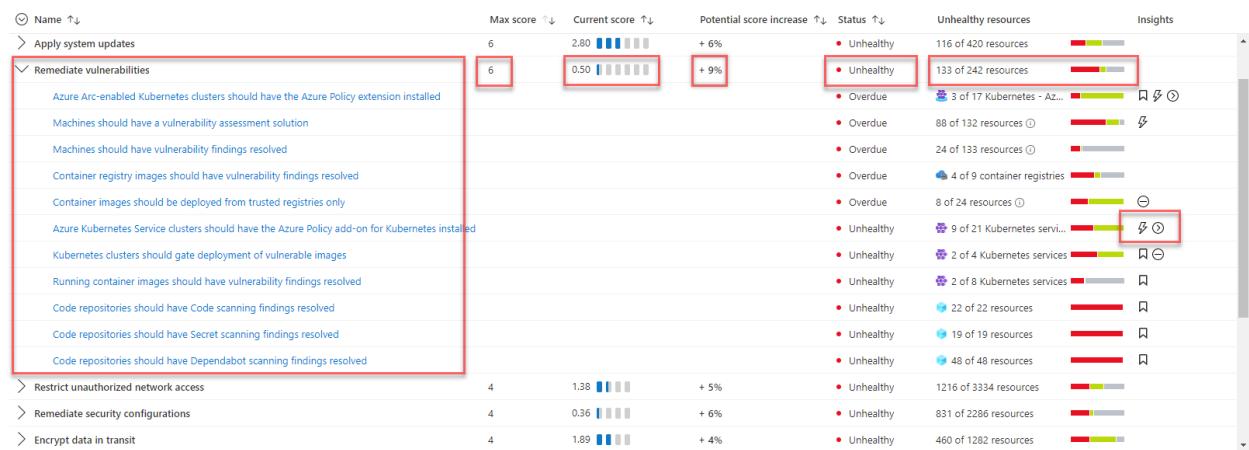
Security score assessment and threat detection

To understand the security status of the system, the solution uses Microsoft Defender for Cloud to assess the infrastructure security and detect security issues. [Microsoft Defender for](#)

Cloud is a tool for security posture management and threat protection. It can protect workloads running in Azure, hybrid, and other cloud platforms.



You automatically enable Defender for Cloud's free plan on all your Azure subscriptions when you first visit the Defender for Cloud pages in the Azure portal. We strongly recommend that you enable it to get your Cloud security posture evaluation and suggestions. Microsoft Defender for Cloud will provide your security score and some security hardening guidance for your subscriptions.



If the solution needs advanced security management and threat detection capabilities such as detection and alerting of suspicious activities, you can enable cloud workload protection individually for different resources.

Cost optimization

For information about the cost optimization pillar of the Well-Architected Framework, see [Cost optimization](#).

A key benefit of the data lakehouse solution is its cost-efficiency and scalable architecture. Most components in the solution use consumption-based billing and will autoscale. In this solution, all data is stored in Data Lake Storage. You only pay to store the data if you don't run any queries or process data.

Pricing for this solution depends on the usage of the following key resources:

- Azure Synapse Serverless SQL: use consumption-based billing, pay only for what you use.
- Apache Spark in Azure Synapse: use consumption-based billing, pay only for what you use.
- Azure Synapse Pipelines: use consumption-based billing, pay only for what you use.
- Azure Data Lakes: use consumption-based billing, pay only for what you use.
- Power BI: the cost is based on which license you purchase.
- Private Link: use consumption-based billing, pay only for what you use.

Different security protection solutions have different cost modes. You should choose the security solution based on your business needs and solution costs.

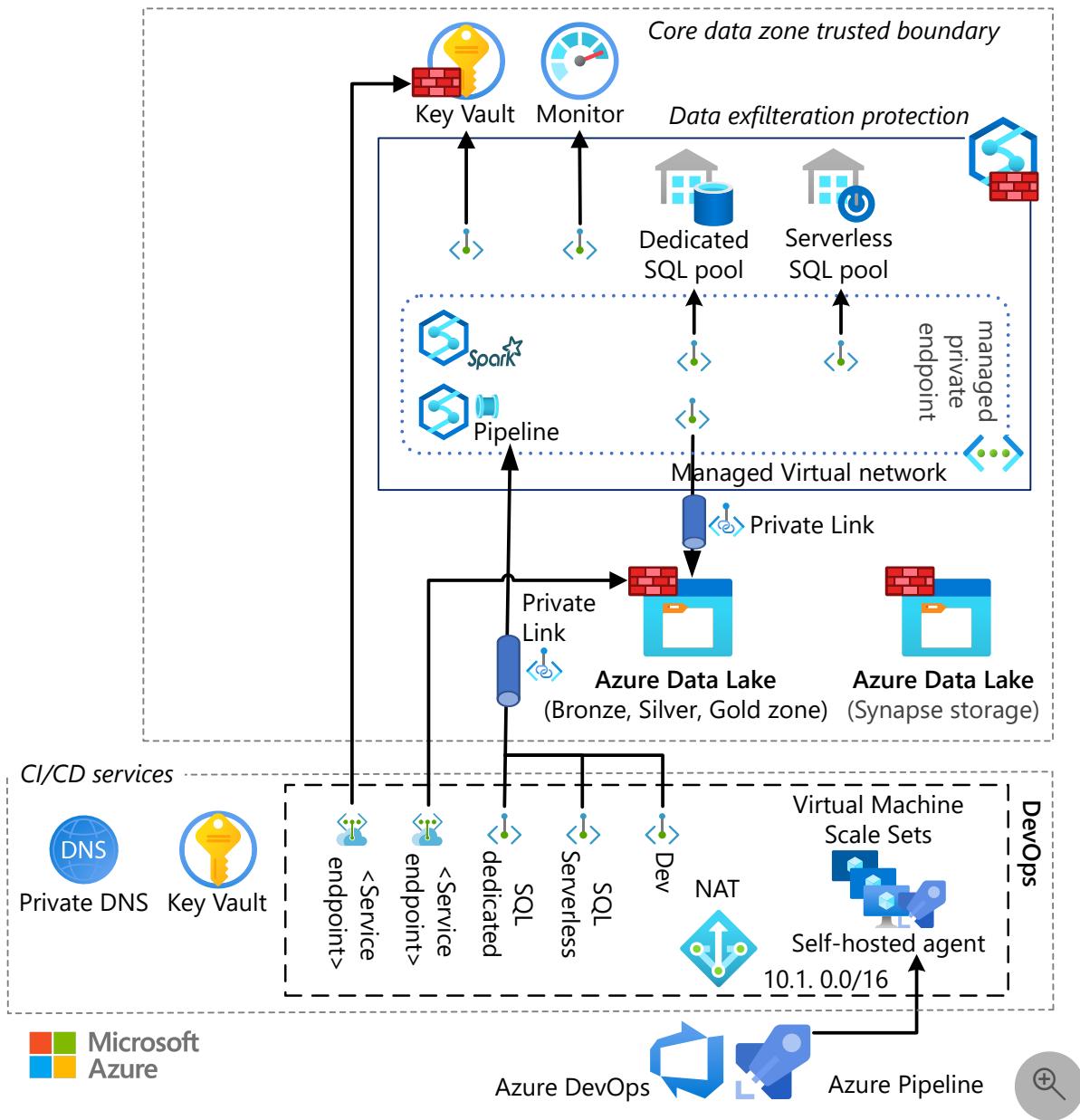
You can use the [Azure Pricing Calculator](#) to estimate the cost of the solution.

Operational excellence

For information about the operational excellence pillar of the Well-Architected Framework, see [Operational excellence](#).

Use a virtual network enabled self-hosted pipeline agent for CI/CD services

The default Azure DevOps pipeline agent doesn't support virtual network communication because it uses a very wide IP address range. This solution implements an Azure DevOps [self-hosted agent](#) in the virtual network so that the DevOps processes can smoothly communicate with the other services in the solution. The connection strings and secrets for running the CI/CD services are stored in an independent key vault. During the deployment process, the self-hosted agent accesses the key vault in the core data zone to update resource configurations and secrets. For more information, see the [Use separate key vaults](#) document. This solution also uses [VM scale sets](#) to ensure that the DevOps engine can automatically scale up and down based on the workload.



Implement infrastructure security scanning and security smoke testing in the CI/CD pipeline

A static analysis tool for scanning infrastructure as code (IaC) files can help detect and prevent misconfigurations that can lead to security or compliance problems. Security smoke testing ensures that the vital system security measures are successfully enabled, protecting against deployment failures.

- Use a static analysis tool to scan infrastructure as code (IaC) templates to detect and prevent misconfigurations that can lead to security or compliance problems. Use tools such as [Checkov](#) or [Terrascan](#) to detect and prevent security risks.
- Make sure the CD pipeline correctly handles deployment failures. Any deployment failure related to security features should be treated as a critical failure. The pipeline should retry the failed action or hold the deployment.

- Validate the security measures in the deployment pipeline by running security smoke testing. The security smoke testing, such as validating the configuration status of deployed resources or testing cases that examine critical security scenarios, can ensure that the security design is working as expected.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Herman Wu](#) | Senior Software Engineer

Other contributors:

- Ian Chen | Principal Software Engineer Lead
- [Jose Contreras](#) | Principal Software Engineering
- Roy Chan | Principal Software Engineer Manager

Next steps

- Azure product documentation
 - [Azure Synapse Analytics](#)
 - [Azure Files](#)
 - [Event Hubs](#)
 - [Blob Storage](#)
 - [Azure Data Lake Storage Gen2](#)
 - [Azure DevOps](#)
 - [Power BI](#)
 - [Data Factory](#)
 - [Azure Bastion](#)
 - [Azure Monitor](#)
 - [Microsoft Defender for Cloud](#)
 - [Azure Key Vault](#)
- Other articles
 - [What is Azure Synapse Analytics?](#)
 - [Serverless SQL pool in Azure Synapse Analytics](#)
 - [Apache Spark in Azure Synapse Analytics](#)
 - [Pipelines and activities in Azure Data Factory and Azure Synapse Analytics](#)
 - [What is Azure Synapse Data Explorer? \(Preview\)](#)
 - [Machine Learning capabilities in Azure Synapse Analytics](#)
 - [What is Microsoft Purview?](#)
 - [Azure Synapse Analytics and Azure Purview Work Better Together](#)

- [Introduction to Azure Data Lake Storage Gen2](#)
- [What is Azure Data Factory?](#)
- [Current Data Patterns Blog Series: Data Lakehouse ↗](#)
- [What is Microsoft Defender for Cloud?](#)
- [The Data Lakehouse, the Data Warehouse and a Modern Data platform architecture ↗](#)
- [The best practices for organizing Azure Synapse workspaces and lakehouse ↗](#)
- [Understanding Azure Synapse Private Endpoints ↗](#)
- [Azure Synapse Analytics – New Insights Into Data Security ↗](#)
- [Azure security baseline for Azure Synapse dedicated SQL pool \(formerly SQL DW\)](#)
- [Cloud Network Security 101: Azure Service Endpoints vs. Private Endpoints ↗](#)
- [How to set up access control for your Azure Synapse workspace](#)
- [Connect to Azure Synapse Studio using Azure Private Link Hubs](#)
- [How-To Deploy your Azure Synapse Workspace Artifacts to a Managed VIRTUAL NETWORK Azure Synapse Workspace ↗](#)
- [Continuous integration and delivery for an Azure Synapse Analytics workspace](#)
- [Secure score in Microsoft Defender for Cloud](#)
- [Best practices for using Azure Key Vault](#)
- [Adatum Corporation scenario for data management and analytics in Azure](#)

Related resources

- [Big data architectures](#)
- [Choose an analytical data store in Azure](#)
- [Enterprise data warehouse](#)
- [Modern data warehouse for small and medium business](#)

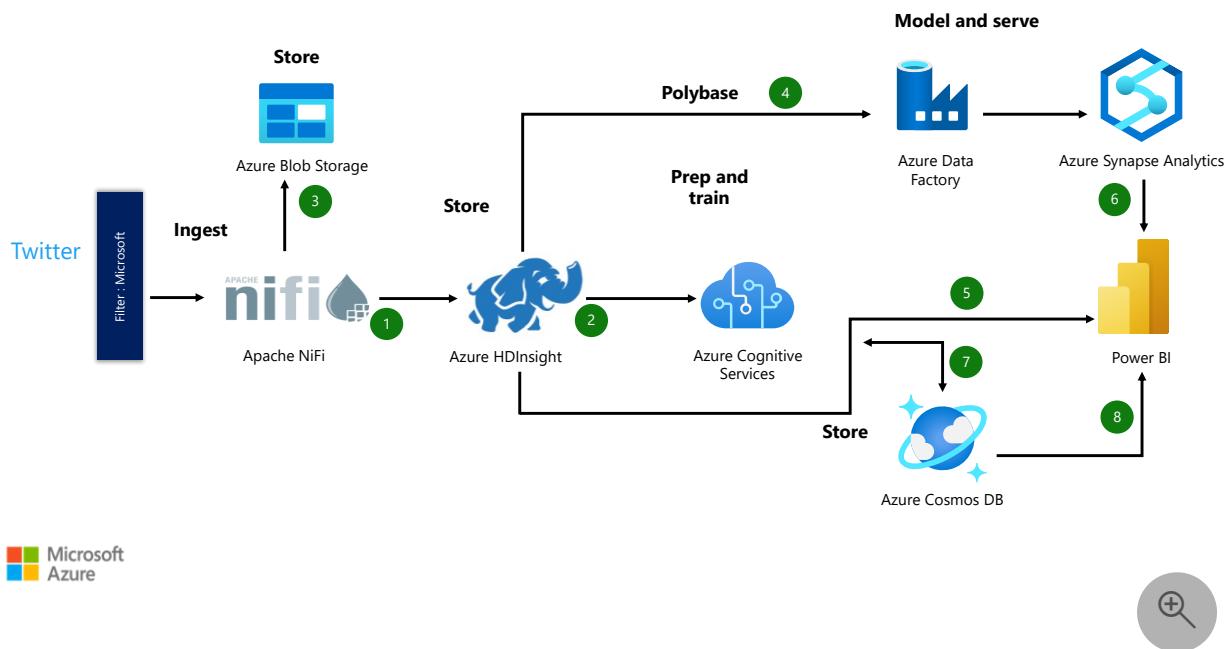
Face recognition and sentiment analysis

Azure AI services Azure Cosmos DB Azure Cosmos DB Azure HDInsight Azure Synapse Analytics

This article presents a solution for gauging public opinion in tweets. The goal is to create a transformation pipeline that outputs clusters of comments and trending subjects.

Apache®, Apache NiFi®, Apache Hadoop®, Apache Hive®, and Apache Airflow® 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 [PowerPoint file](#) of this diagram.

Ingestion pipeline

The Twitter ingestion pipeline consists of four stages.

Collect and ingest data

The following components ingest tweets:

- Hadoop Distributed File System (HDFS) (1)
- Azure Synapse Analytics via Azure Data Factory (4)
- Azure Blob Storage (4)
- Azure Cosmos DB (4)

Process data

During data processing:

- The JSON file that contains tweet data is transformed into CSV format (2).
- Apache Hive and Azure Synapse Analytics tables are created (2).
- Sentiment analysis runs on the tweets (2).
- Azure Cognitive Services processes images and identifies human faces (2).

Store data

The following components store data:

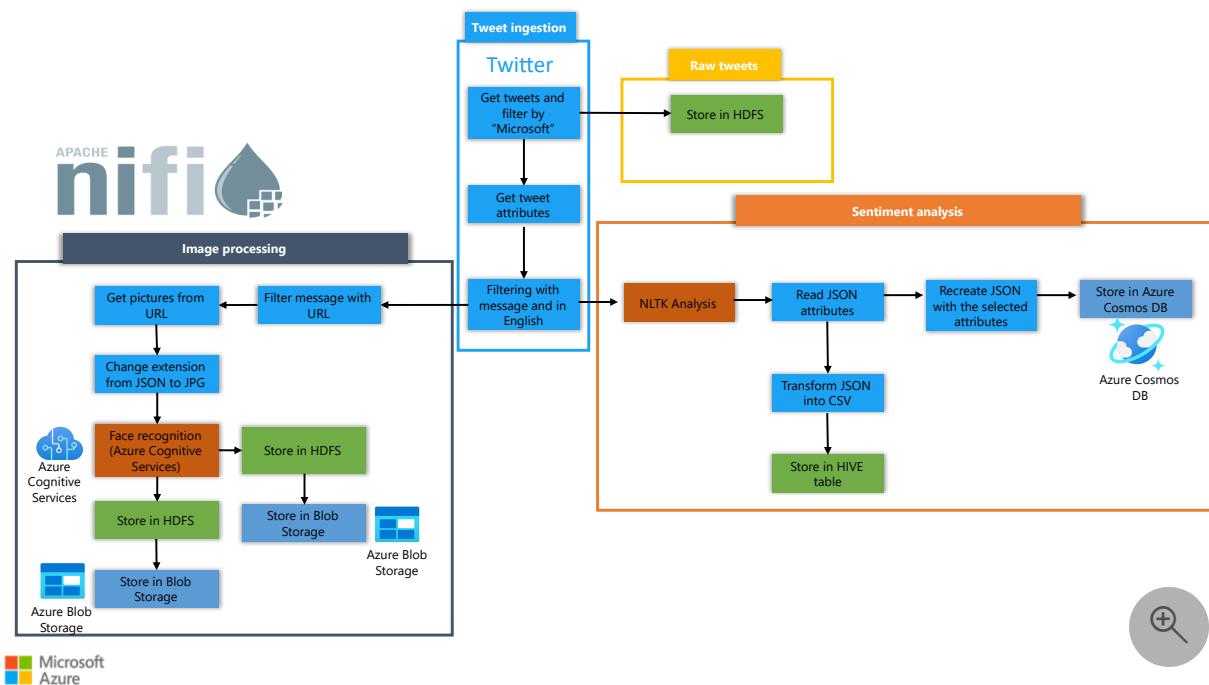
- HDFS and Hive (3)
- Azure Synapse Analytics (3)
- Blob Storage (3)
- Azure Cosmos DB (7)

Visualize data

Power BI dashboards display data from the following sources:

- Hive (5)
- Azure Synapse Analytics (6)
- Azure Cosmos DB (8)

Dataflow



[Download a *PowerPoint file*](#) of this diagram.

The solution's dataflow contains three main parts.

Ingest tweets

The file with the tweet data, which is in JSON format, is transformed into CSV format. Attributes are extracted from the JSON data to use as variables for the CSV composition.

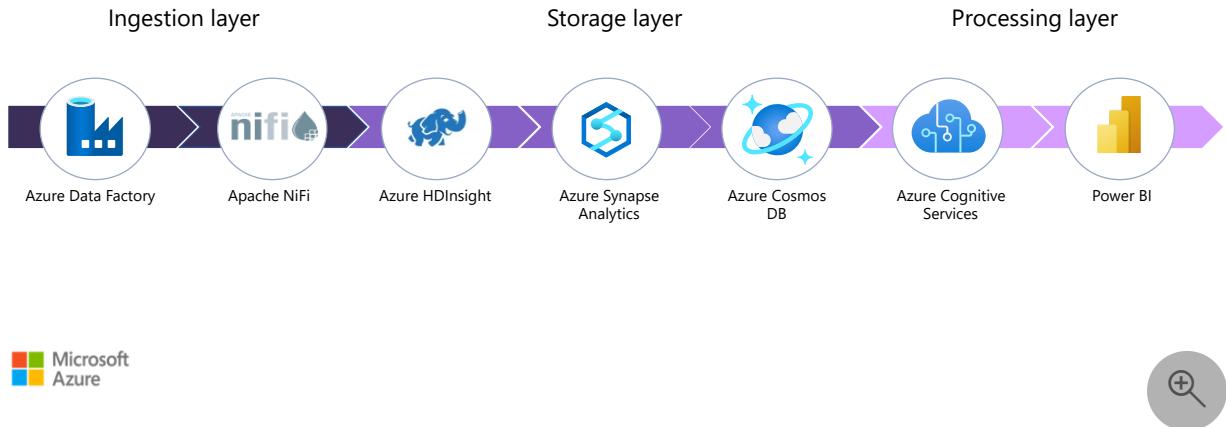
Process images

Sentiment analysis runs on tweets that contain images. After the images are collected, face detection processes run on the images. Any human faces that are recognized are stored in HDInsight.

Run sentiment analysis

A Natural Language Toolkit (NLTK) algorithm runs on the ingested messages. Sentiment analysis runs on the text in the tweets. The results are stored in CSV format in a Hive table, and the JSON data is stored in Azure Cosmos DB.

Components



Download a [PowerPoint file](#) of this diagram.

- [Data Factory](#) provides batch transformation services for various sources and sinks. As a key component of big data processing, Data Factory helps to simplify extract-transform-load (ETL) workloads. Data Factory also handles the complexities and scale challenges of big data integration.
- [NiFi](#) automates the flow of data among software systems. NiFi offers security features, an extensible architecture, and a flexible scaling model. It handles multiple sources and multiple sinks with different types of processors. NiFi functionality includes:
 - Running streaming transformations.
 - Connecting decoupled systems in the cloud.
 - Moving data in and out of Azure Storage and other data stores.
 - Integrating edge-to-cloud and hybrid-cloud applications with Azure services.
 - Providing robust data provenance capabilities.
- [HDInsight](#) is a Hadoop platform for data and analytics for on-premises environments. HDInsight can securely ingest, store, and process data in real time and in batches. HDInsight is built on Hortonworks Data Platform (HDP), an open-source framework for distributed storage and processing of large data sets that come from multiple sources.
- [Azure Synapse Analytics](#) is an analytics service for data warehouses and big data systems. It centralizes data in the cloud for easy access.
- [Azure Cosmos DB](#) is a fully managed NoSQL database for modern app development. By providing single-digit millisecond response times and automatic

and instant scalability, Azure Cosmos DB guarantees speed at any scale. Its SLA-backed availability and enterprise-grade security provide business continuity.

- [Cognitive Services](#) consists of cloud-based services that provide AI functionality. The REST APIs and client library SDKs help you build cognitive intelligence into apps even if you don't have AI or data science skills.
- [Power BI](#) is a business analytics service that's part of Microsoft Power Platform. Power BI provides interactive visualizations and business intelligence capabilities. Its easy-to-use interface makes it possible for users to create their own reports and dashboards.

Alternatives

You can substitute alternatives for most solution components. For example:

- Instead of an HDInsight cluster, you can use a Cloudera cluster.
- You can use Azure Databricks instead of Data Factory. Azure Databricks can transform and store data, but you can also use it as an orchestrator. Another alternative is to use both services. Many solutions that use Data Factory also make use of Azure Databricks.
- Instead of Nifi, you can use Apache Airflow as a workflow tool that runs ETL scripts.
- For your main file repository, you can use Elasticsearch in place of Azure Cosmos DB.
- For dashboard services, you can use Kibana instead of Power BI.

Scenario details

Branding is important to companies, because a company's value depends on the market's image of that company. As your company moves to make predictive, data-driven decisions, rather than reactive decisions, you need to monitor and understand what's happening in real time. To gain a competitive advantage, you need to use social media analysis to identify and understand public opinion. Along with identifying sentiment in tweets, you might also choose to recognize faces and images.

This solution gauges public opinion in tweets. A transformation pipeline outputs clusters of comments and trending subjects. The pipeline delivers value by seamlessly integrating open-source solutions like Apache NiFi and Azure HDInsight with Azure sentiment analysis and face recognition services. The solution applies to a broad range of industries—monitoring social networks isn't limited to one sector.

Potential use cases

This solution is ideal for any area that monitors branding on social networks, including:

- Marketing
- Communications
- Politics
- Media and entertainment
- Real estate and facilities
- Food service (travel and hospitality)
- Fashion
- Retail

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](#).

Depending on the processing tools and number of sources that you use, you might be able to streamline the solution's transformations and visualizations. If possible, consider using a basic pipeline with one sink. Instead of using multiple sources and multiple dashboards, feed that pipeline into a single dashboard.

This example uses as many services as possible. Through this approach, you can compare the performance and experiences that you have with Power BI across various sources and data types.

Reliability

Reliability ensures your application can meet the commitments you make to your customers. For more information, see [Overview of the reliability pillar](#).

In production environments, evaluate your recovery time objective (RTO) and recovery point objective (RPO). All disaster recovery decisions and scenarios depend on those evaluations.

In most cases, you need a high availability service for each tool. For effective disaster recovery, it's important to reduce your RTO. But if you have high availability, you can avoid disaster scenarios. For instance, you might create services in another region.

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](#).

Aim for a strong security posture by using an identity-based system and native Azure tools. For external components, use external authentication tools like Kerberos to ensure a robust and secure workload.

Cost optimization

For information about creating a cost-effective workload, review [Overview of the cost optimization pillar](#).

Operational excellence

Operational excellence covers the operations processes that deploy an application and keep it running in production. For more information, see [Overview of the operational excellence pillar](#).

Centralize the monitoring logs from all services. The solution uses external tools and tools that are native to Azure. To achieve a holistic view of all systems, integrate monitoring data from all tools.

Performance efficiency

Performance efficiency is the ability of your workload to scale to meet the demands placed on it by users in an efficient manner. For more information, see [Performance efficiency pillar overview](#).

Because the solution uses multiple sources, consider compression as part of the process. Also consider the file formats that you use. Configure Azure Cosmos DB to achieve a tradeoff between latency and consistency levels. But monitor and evaluate Azure Cosmos DB performance throughout the process to prevent that component from becoming a bottleneck. To decrease latency, consider partitioning data by location or moving data sources close to where you use them.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Jose Mendez](#) | Senior Cloud Solution Architect
- [Katie Novotny](#) | Senior Cloud Solution Architect

Next steps

- [What is Azure Data Factory?](#)
- [What is Azure HDInsight?](#)
- [Introduction to Azure Blob Storage](#)
- [What is dedicated SQL pool \(formerly SQL DW\) in Azure Synapse Analytics?](#)
- [Welcome to Azure Cosmos DB](#)
- [What are Azure Cognitive Services?](#)
- [What is Power BI?](#)
- [Create and consume Cognitive Services](#)
- [Analyze text with the Language service](#)
- [Detect and analyze faces with the Face service](#)

Related resources

- [Extract, transform, and load \(ETL\) using HDInsight](#)
- [Knowledge mining for customer feedback](#)
- [Apache NiFi on Azure](#)

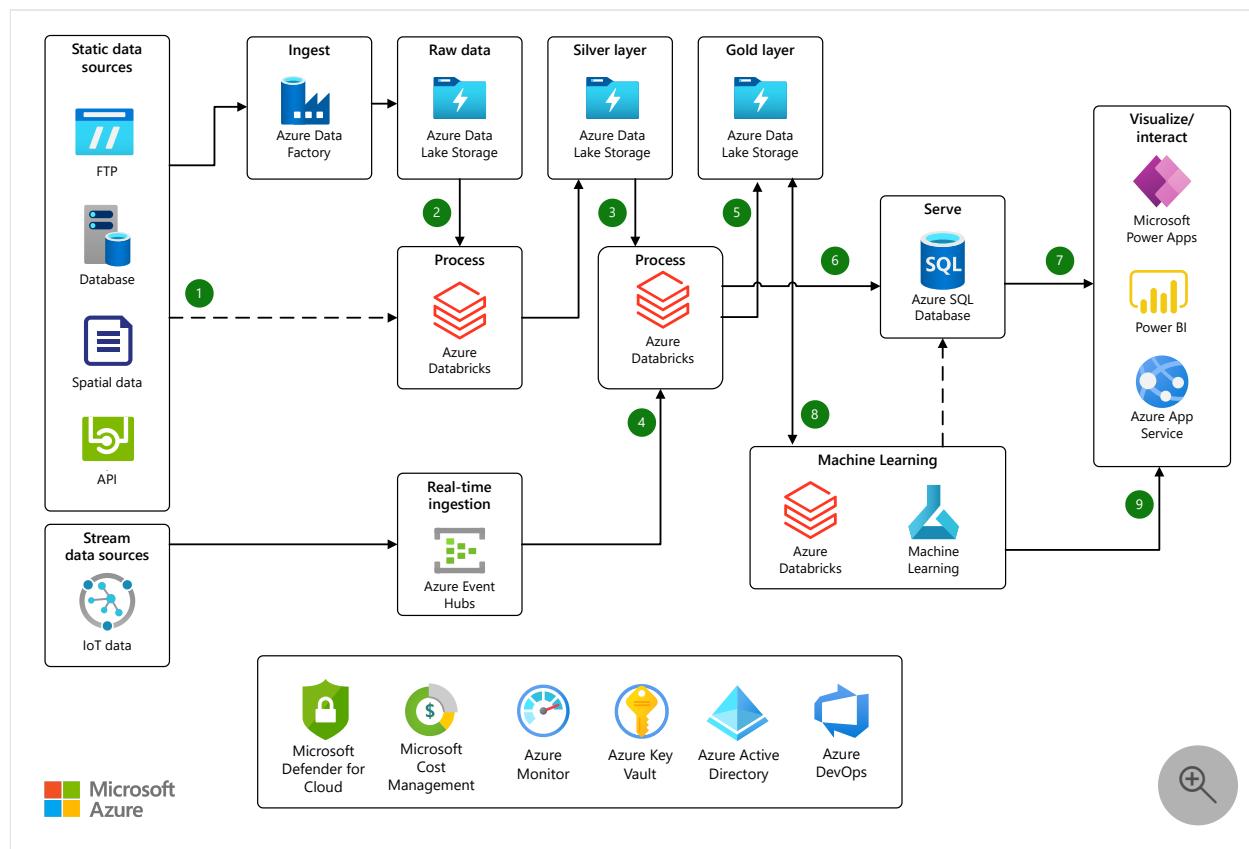
Build a sports analytics architecture on Azure

Azure Data Factory Azure Data Lake Storage Azure Databricks Azure Event Hubs Power BI

The focus of this article is to show a practical architecture that uses Azure services to process and maintain data used by sports analytics solutions. It provides a framework for sports organizations to build highly scalable solutions with, while giving them the flexibility to add more services that meet the nuanced requirements of their use cases.

Apache®, 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

1. Data is ingested from source systems by using one of the following methods:

- Azure Data Factory ingests raw data from several data sources and stores it in Azure Data Lake Storage for downstream processing.
- Some raw data sources might be large and might not need the raw data to be stored in Data Lake Storage initially, like the spatial on-court/on-field data. In these cases, you can use Azure Databricks to ingest source data and immediately transform data so that it's cleansed, normalized, and saved to Data Lake Storage in an easy-to-digest format.
- Data that's generated by sensors in real-time is ingested as messages by Azure Event Hubs.

2. Azure Databricks transforms raw data so that it's cleansed of any errors and normalized. With the cloudFiles feature of Azure Databricks Auto Loader, raw files are automatically processed as they land in Data Lake Storage. The transformed data moves back into Data Lake Storage for further curating.

3. Azure Databricks applies business logic to the transformed data. Stream data is also combined with the transformed data during this process.

4. Azure Databricks processes stream data from Azure Event Hubs and combines it with static data.

5. The final processed data is written to Data Lake Storage in Delta format.

6. Transformed data that's used in the visualization layer, like Power BI, is written to an Azure SQL Database. This database becomes the data source for any reporting needs.

7. Curated data is visualized and manipulated through Power BI, Power Apps, or a custom web application that's hosted by an Azure App Service.

8. Azure Machine Learning builds and trains machine learning models by using data imported into Azure Machine Learning Datasets and external sources. The datasets and sources are directly linked to the Azure Machine Learning Workspace. You can control access and authentication for data and the Machine Learning workspace with Microsoft Entra ID and Azure Key Vault. Models can also be retrained as necessary in Machine Learning.

9. As an alternative to storing model results in Data Lake Storage or SQL Database, you can deploy Machine Learning models to containers using Azure Kubernetes Services (AKS) as a web service and called via a REST API endpoint. The web service deploys by using an Azure App Service, and then you can send data to the REST

API endpoint and receive the prediction returned by the model within the web application.

Throughout the process:

- Azure Monitor collects information on events and performance.
- Key Vault secures passwords, connection strings, and secrets.
- Azure DevOps manages code repositories and deployment pipelines.

Components

- [Azure Data Lake Storage](#) is a scalable and secure data lake for high-performance analytics workloads. You can use Data Lake Storage to manage petabytes of data with high throughput. It can accommodate multiple, heterogeneous sources and data that's in structured, semi-structured, or unstructured formats.
- [Azure Databricks](#) is a data analytics platform that uses Spark clusters. The clusters are optimized for the Azure platform.
- [Azure Data Factory](#) is a fully managed, scalable, and serverless data integration service. It provides a data integration and transformation layer that works with various data stores.
- [Azure Machine Learning](#) is a cloud service for accelerating and managing the machine learning project lifecycle. Machine learning professionals, data scientists, and engineers can use it in their day-to-day workflows to train, deploy models, and manage MLOps. You can create a model in Machine Learning or use a model built from an open-source platform like PyTorch, TensorFlow, or scikit-learn. MLOps tools help you monitor, retrain, and redeploy models.
- [Azure Event Hubs](#) is a big-data streaming platform and event ingestion service. It can receive and process millions of events per second. Data sent to an event hub can be transformed and stored by using any real-time analytics provider or batching/storage adapters.
- [Azure SQL Database](#) is a fully managed platform as a service (PaaS) database engine that handles most of the database management functions like upgrading, patching, backups, and monitoring without user involvement. SQL Database is always running on the latest stable version of the SQL Server database engine and patched OS with high availability.
- [Power BI](#) is a collection of software services, apps, and connectors that work together to turn your unrelated sources of data into coherent, visually immersive, and interactive insights.
- [Power Apps](#) is a suite of apps, services, connectors, and data platform that provides a rapid development environment to build custom apps for your business needs. By using Power Apps, you can quickly build custom business apps that

connect to your data stored either in the underlying data platform or in various online and on-premises data sources like SharePoint, Microsoft 365, and Dynamics 365.

- [Azure App Service](#) is an HTTP-based service for hosting web applications, REST APIs, and mobile back ends. You can use it with your favorite languages, like .NET, .NET Core, and Java.
- [Microsoft Defender for Cloud](#) is a tool for security posture management and threat protection.
- [Azure Cost Management and Billing](#) helps you understand your Azure invoice, manage your billing account and subscriptions, monitor, control Azure spending, and optimize resource use.
- [Azure Monitor](#) delivers a comprehensive solution for collecting, analyzing, and acting on telemetry from your cloud and on-premises environments.
- [Azure Key Vault](#) is a cloud service for securely storing and accessing secrets.
- [Microsoft Entra ID](#) is an identity service that provides single sign-on, multifactor authentication, and conditional access to guard against most cybersecurity attacks.
- [Azure DevOps](#) provides developer services so that teams can plan work, collaborate on code development, and build and deploy applications. Azure DevOps supports a collaborative culture and set of processes that bring together developers, project managers, and contributors to develop software.

Alternatives

- You can use [Synapse Spark Pools](#) instead of Azure Databricks for sports analytics by using the same open-source frameworks.
- Instead of Azure SQL Database, you can use [Azure SQL Managed Instance](#) to store data that's served to the visualize/interact layer.
- You can use an Azure Synapse Analytics dedicated SQL pool instead of an Azure SQL Database if the reporting requirements require several terabytes of data stored in the serving layer.
- If you don't want to use a database as the serving layer for reporting, you can choose to use a *semantic lakehouse* approach. In this scenario, reporting applications connect to logical tables that are defined by a service like [Databricks SQL](#). These logical tables are used to structure data that's stored in the gold layer (data that's formatted using the Delta format) of Azure Data Lake Storage Gen2, so that the data can be easily read.
- Instead of Azure Databricks, you can use SQL Database or SQL Managed Instance to query and process data. These databases provide the familiar T-SQL language, which you can use for analysis.

- You can use [Azure Stream Analytics](#) instead of Azure Databricks to process stream data.
- You can use Azure Machine Learning instead of Azure Databricks to train your machine learning models.
- You can use GitHub instead of Azure DevOps to manage your code repositories and continuous integration and continuous delivery (CI/CD) pipelines.

Scenario details

Sports analytics is a field that applies data analytics techniques to team or individual performance data. Then you can use the data to create a competitive advantage over an opponent. In addition to analyzing traditional box score statistics, there has been an explosion of data in recent years that sports teams can use to improve the performance of an individual athlete or an entire team. Examples of such data include player data collected from sensors and spatial data that captures player movement during a game. Traditional systems struggle to process and maintain these data sources because of the large volumes of data generated. These data sources also format data in several different ways and allow users to process data at different speeds, providing more challenges for traditional data processing solutions.

Potential use cases

This solution is ideal for the sports industry, and applies to the following scenarios:

- Manage large volumes of data from several source systems in a centralized ecosystem.
- Analyze player tracking and temporal data to gain insights into individual and team performance.
- With consideration for spatial metrics, determine the best possible player positioning and strategies during gameplay.
- Process and evaluate player performance data to optimize athlete training routines.
- Analyze historical data to make well-informed personnel decisions during the draft or free agency.
- Store and analyze real-time telemetry from Internet of Things (IoT) devices that are attached to equipment like bats, shoulder pads, and volleyballs.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is 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](#).

Follow MLOps guidelines to standardize and manage end-to-end Machine Learning lifecycles that are scalable across multiple workspaces. Before going into production, ensure the implemented solution supports ongoing inference with retraining cycles and automated redeployments of models.

Use the [Azure/mlops-v2](#) GitHub repository as an MLOps resource.

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](#).

Consider using the following security resources in this architecture:

- Secure cluster connectivity (No Public IP / NPIP)
- Store credentials in Azure Key Vault
- Deploy dedicated Azure services into virtual networks
- Azure Databricks Premium
- Enterprise security for Azure Databricks

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

- To estimate the cost of implementing this solution, use the [Azure pricing calculator](#) for the services mentioned above.
- Power BI comes with different licensing offerings. For more information, see [Power BI pricing](#).
- Depending on the volume of data and complexity of your geospatial analysis, you might need to scale your Databricks cluster configurations that affect your cost. Refer to the Databricks [cluster sizing examples](#) for best practices on cluster configuration.

Performance efficiency

Performance efficiency is the ability of your workload to scale to meet the demands placed on it by users in an efficient manner. For more information, see [Performance efficiency pillar overview](#).

If you use Azure Data Factory Mapping Data Flows for extract, transform, and load (ETL), follow the performance and tuning guide for mapping data flows. Mapping data flows this way optimizes your data pipeline and ensures that your data flows meet your performance benchmarks.

Deploy this scenario

To deploy this scenario, follow the steps described in this Azure quickstart, [Deploy the Sports Analytics on Azure Architecture](#). Be sure to read the **Prerequisites** section in the quickstart before deploying the solution.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Giulia Gallo](#) | Senior Cloud Solution Architect
- [Jake Switzer](#) | Sports Analytics Solution Architect
- [Tash Tahir](#) | Principal Cloud Solution Architect

Other contributor:

- [Jason Martinez](#) | Technical Writer

Next steps

- [App Service overview](#)
- [Azure Event Hubs — A big data streaming platform and event ingestion service](#)
- [Azure security baseline for Azure Machine Learning](#)
- [Consume an Azure Machine Learning model deployed as a web service](#)
- [Data integration at scale with Azure Data Factory or Azure Synapse Pipeline](#)
- [Data engineering with Azure Databricks](#)
- [Introduction to Azure Data Lake Storage Gen2](#)
- [What is Azure Machine Learning?](#)
- [What is Azure SQL Database?](#)

- [What is Power BI?](#)
- [What is Power Apps?](#)

Related resources

- [IoT and data analytics](#)
- [Monitoring Azure Databricks](#)
- [Stream processing with Azure Databricks](#)

Stream processing with Azure Databricks

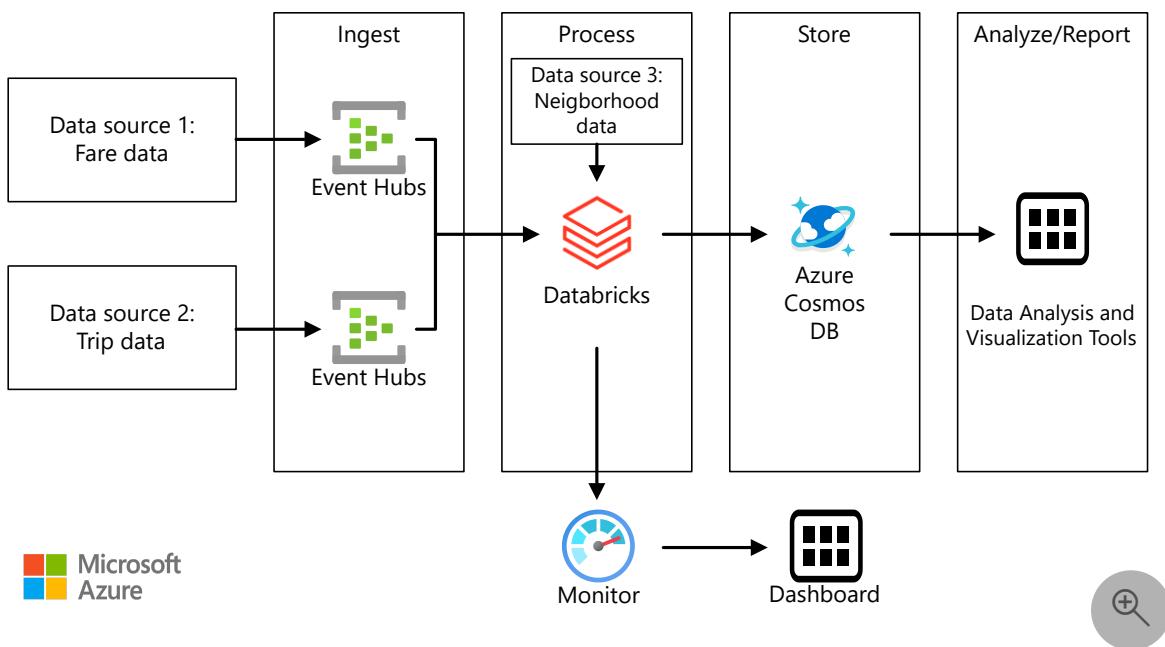
Azure Cosmos DB Azure Databricks Azure Event Hubs Azure Log Analytics Azure Monitor

This reference architecture shows an end-to-end stream processing pipeline. This type of pipeline has four stages: ingest, process, store, and analysis and reporting. For this reference architecture, the pipeline ingests data from two sources, performs a join on related records from each stream, enriches the result, and calculates an average in real time. The results are stored for further analysis.



A reference implementation for this architecture is available on [GitHub](#).

Architecture



[Download a Visio file](#) of this architecture.

Workflow

The architecture consists of the following components:

Data sources. In this architecture, there are two data sources that generate data streams in real time. The first stream contains ride information, and the second contains fare

information. The reference architecture includes a simulated data generator that reads from a set of static files and pushes the data to Event Hubs. The data sources in a real application would be devices installed in the taxi cabs.

Azure Event Hubs. [Event Hubs](#) is an event ingestion service. This architecture uses two event hub instances, one for each data source. Each data source sends a stream of data to the associated event hub.

Azure Databricks. [Databricks](#) is an Apache Spark-based analytics platform optimized for the Microsoft Azure cloud services platform. Databricks is used to correlate of the taxi ride and fare data, and also to enrich the correlated data with neighborhood data stored in the Databricks file system.

Azure Cosmos DB. The output of an Azure Databricks job is a series of records, which are written to [Azure Cosmos DB for Apache Cassandra](#). Azure Cosmos DB for Apache Cassandra is used because it supports time series data modeling.

- [Azure Synapse Link for Azure Cosmos DB](#) enables you to run near real-time analytics over operational data in Azure Cosmos DB, **without any performance or cost impact on your transactional workload**, by using the two analytics engines available from your Azure Synapse workspace: [SQL Serverless](#) and [Spark Pools](#).

Azure Log Analytics. Application log data collected by [Azure Monitor](#) is stored in a [Log Analytics workspace](#). Log Analytics queries can be used to analyze and visualize metrics and inspect log messages to identify issues within the application.

Alternatives

- [Synapse Link](#) is the Microsoft preferred solution for analytics on top of Azure Cosmos DB data.

Scenario details

Scenario: A taxi company collects data about each taxi trip. For this scenario, we assume there are two separate devices sending data. The taxi has a meter that sends information about each ride — the duration, distance, and pickup and drop-off locations. A separate device accepts payments from customers and sends data about fares. To spot ridership trends, the taxi company wants to calculate the average tip per mile driven, in real time, for each neighborhood.

Potential use cases

This solution is optimized for the retail industry.

Data ingestion

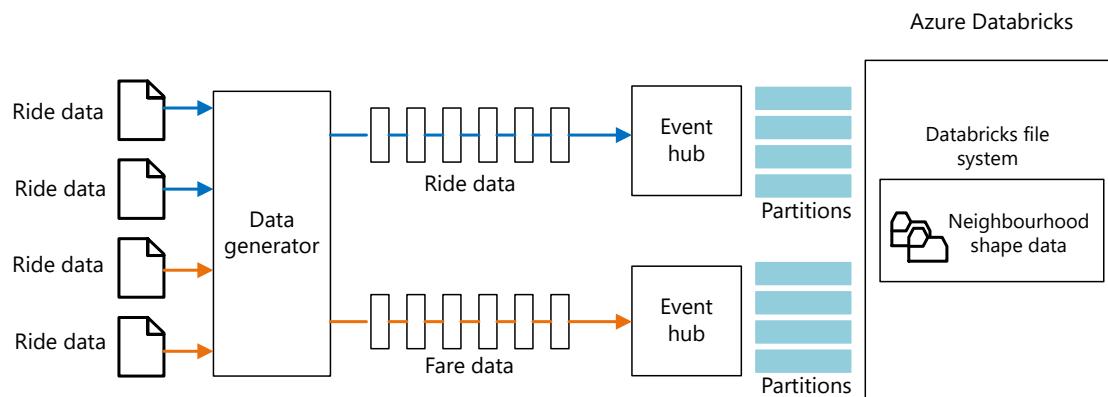
To simulate a data source, this reference architecture uses the [New York City Taxi Data](#) dataset^[1]. This dataset contains data about taxi trips in New York City over a four-year period (2010 – 2013). It contains two types of record: Ride data and fare data. Ride data includes trip duration, trip distance, and pickup and drop-off location. Fare data includes fare, tax, and tip amounts. Common fields in both record types include medallion number, hack license, and vendor ID. Together these three fields uniquely identify a taxi plus a driver. The data is stored in CSV format.

[1] Donovan, Brian; Work, Dan (2016): New York City Taxi Trip Data (2010-2013). University of Illinois at Urbana-Champaign. <https://doi.org/10.13012/J8PN93H8>

The data generator is a .NET Core application that reads the records and sends them to Azure Event Hubs. The generator sends ride data in JSON format and fare data in CSV format.

Event Hubs uses [partitions](#) to segment the data. Partitions allow a consumer to read each partition in parallel. When you send data to Event Hubs, you can specify the partition key explicitly. Otherwise, records are assigned to partitions in round-robin fashion.

In this scenario, ride data and fare data should end up with the same partition ID for a given taxi cab. This enables Databricks to apply a degree of parallelism when it correlates the two streams. A record in partition n of the ride data will match a record in partition n of the fare data.



Download a [Visio file](#) of this architecture.

In the data generator, the common data model for both record types has a `PartitionKey` property that is the concatenation of `Medallion`, `HackLicense`, and `VendorId`.

C#

```
public abstract class TaxiData
{
    public TaxiData()
    {
    }

    [JsonProperty]
    public long Medallion { get; set; }

    [JsonProperty]
    public long HackLicense { get; set; }

    [JsonProperty]
    public string VendorId { get; set; }

    [JsonIgnore]
    public string PartitionKey
    {
        get => $"{Medallion}_{HackLicense}_{VendorId}";
    }
}
```

This property is used to provide an explicit partition key when sending to Event Hubs:

C#

```
using (var client = pool.GetObject())
{
    return client.Value.SendAsync(new EventData(Encoding.UTF8.GetBytes(
        t.GetData(dataFormat))), t.PartitionKey);
}
```

Event Hubs

The throughput capacity of Event Hubs is measured in [throughput units](#). You can autoscale an event hub by enabling [auto-inflate](#), which automatically scales the throughput units based on traffic, up to a configured maximum.

Stream processing

In Azure Databricks, data processing is performed by a job. The job is assigned to and runs on a cluster. The job can either be custom code written in Java, or a Spark notebook [↗](#).

In this reference architecture, the job is a Java archive with classes written in both Java and Scala. When specifying the Java archive for a Databricks job, the class is specified for execution by the Databricks cluster. Here, the **main** method of the **com.microsoft.pnp.TaxiCabReader** class contains the data processing logic.

Reading the stream from the two event hub instances

The data processing logic uses [Spark structured streaming](#) [↗](#) to read from the two Azure event hub instances:

Scala

```
val rideEventHubOptions = EventHubsConf(rideEventHubConnectionString)
    .setConsumerGroup(conf.taxiRideConsumerGroup())
    .setStartingPosition(EventPosition.fromStartOfStream)
val rideEvents = spark.readStream
    .format("eventhubs")
    .options(rideEventHubOptions.toMap)
    .load

val fareEventHubOptions = EventHubsConf(fareEventHubConnectionString)
    .setConsumerGroup(conf.taxiFareConsumerGroup())
    .setStartingPosition(EventPosition.fromStartOfStream)
val fareEvents = spark.readStream
    .format("eventhubs")
    .options(fareEventHubOptions.toMap)
    .load
```

Enriching the data with the neighborhood information

The ride data includes the latitude and longitude coordinates of the pick up and drop off locations. While these coordinates are useful, they are not easily consumed for analysis. Therefore, this data is enriched with neighborhood data that is read from a [shapefile](#) [↗](#).

The shapefile format is binary and not easily parsed, but the [GeoTools](#) [↗](#) library provides tools for geospatial data that use the shapefile format. This library is used in the **com.microsoft.pnp.GeoFinder** class to determine the neighborhood name based on the pick up and drop off coordinates.

Scala

```
val neighborhoodFinder = (lon: Double, lat: Double) => {
    NeighborhoodFinder.getNeighborhood(lon, lat).get()
}
```

Joining the ride and fare data

First the ride and fare data is transformed:

Scala

```
val rides = transformedRides
    .filter(r => {
        if (r.isNullAt(r.fieldIndex("errorMessage"))) {
            true
        } else {
            malformedRides.add(1)
            false
        }
    })
    .select(
        $"ride.*",
        to_neighborhood($"ride.pickupLon", $"ride.pickupLat")
            .as("pickupNeighborhood"),
        to_neighborhood($"ride.dropoffLon", $"ride.dropoffLat")
            .as("dropoffNeighborhood")
    )
    .withWatermark("pickupTime", conf.taxiRideWatermarkInterval())

val fares = transformedFares
    .filter(r => {
        if (r.isNullAt(r.fieldIndex("errorMessage"))) {
            true
        } else {
            malformedFares.add(1)
            false
        }
    })
    .select(
        $"fare.*",
        $"pickupTime"
    )
    .withWatermark("pickupTime", conf.taxiFareWatermarkInterval())
```

And then the ride data is joined with the fare data:

Scala

```
val mergedTaxiTrip = rides.join(fares, Seq("medallion", "hackLicense", "vendorId", "pickupTime"))
```

Processing the data and inserting into Azure Cosmos DB

The average fare amount for each neighborhood is calculated for a given time interval:

Scala

```
val maxAvgFarePerNeighborhood = mergedTaxiTrip.selectExpr("medallion", "hackLicense", "vendorId", "pickupTime", "rateCode", "storeAndForwardFlag", "dropoffTime", "passengerCount", "tripTimeInSeconds", "tripDistanceInMiles", "pickupLon", "pickupLat", "dropoffLon", "dropoffLat", "paymentType", "fareAmount", "surcharge", "mtaTax", "tipAmount", "tollsAmount", "totalAmount", "pickupNeighborhood", "dropoffNeighborhood")  
    .groupBy(window($"pickupTime", conf.windowInterval()),  
    $"pickupNeighborhood")  
    .agg(  
        count("*").as("rideCount"),  
        sum($"fareAmount").as("totalFareAmount"),  
        sum($"tipAmount").as("totalTipAmount"),  
        (sum($"fareAmount")/count("*")).as("averageFareAmount"),  
        (sum($"tipAmount")/count("*")).as("averageTipAmount")  
    )  
    .select($"window.start", $"window.end", $"pickupNeighborhood",  
    $"rideCount", $"totalFareAmount", $"totalTipAmount", $"averageFareAmount",  
    $"averageTipAmount")
```

Which is then inserted into Azure Cosmos DB:

Scala

```
maxAvgFarePerNeighborhood  
    .writeStream  
    .queryName("maxAvgFarePerNeighborhood_cassandra_insert")  
    .outputMode(OutputMode.Append())  
    .foreach(new CassandraSinkForeach(connector))  
    .start()  
    .awaitTermination()
```

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

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](#).

Access to the Azure Databricks workspace is controlled using the [administrator console](#). The administrator console includes functionality to add users, manage user permissions, and set up single sign-on. Access control for workspaces, clusters, jobs, and tables can also be set through the administrator console.

Managing secrets

Azure Databricks includes a [secret store](#) that is used to store secrets, including connection strings, access keys, user names, and passwords. Secrets within the Azure Databricks secret store are partitioned by **scopes**:

```
Bash
```

```
databricks secrets create-scope --scope "azure-databricks-job"
```

Secrets are added at the scope level:

```
Bash
```

```
databricks secrets put --scope "azure-databricks-job" --key "taxi-ride"
```

ⓘ Note

An Azure Key Vault-backed scope can be used instead of the native Azure Databricks scope. To learn more, see [Azure Key Vault-backed scopes](#).

In code, secrets are accessed via the Azure Databricks [secrets utilities](#).

Monitoring

Azure Databricks is based on Apache Spark, and both use [log4j](#) as the standard library for logging. In addition to the default logging provided by Apache Spark, you can implement logging to Azure Log Analytics following the article [Monitoring Azure Databricks](#).

As the `com.microsoft.pnp.TaxiCabReader` class processes ride and fare messages, it's possible that either one may be malformed and therefore not valid. In a production environment, it's important to analyze these malformed messages to identify a problem with the data sources so it can be fixed quickly to prevent data loss. The `com.microsoft.pnp.TaxiCabReader` class registers an Apache Spark Accumulator that keeps track of the number of malformed fare and ride records:

Scala

```
@transient val appMetrics = new AppMetrics(spark.sparkContext)
appMetrics.registerGauge("metrics.malformedrides",
AppAccumulators.getRideInstance(spark.sparkContext))
appMetrics.registerGauge("metrics.malformedfares",
AppAccumulators.getFareInstance(spark.sparkContext))
SparkEnv.get.metricsSystem.registerSource(appMetrics)
```

Apache Spark uses the Dropwizard library to send metrics, and some of the native Dropwizard metrics fields are incompatible with Azure Log Analytics. Therefore, this reference architecture includes a custom Dropwizard sink and reporter. It formats the metrics in the format expected by Azure Log Analytics. When Apache Spark reports metrics, the custom metrics for the malformed ride and fare data are also sent.

The following are example queries that you can use in your Azure Log Analytics workspace to monitor the execution of the streaming job. The argument `ago(1d)` in each query will return all records that were generated in the last day, and can be adjusted to view a different time period.

Exceptions logged during stream query execution

Kusto

```
SparkLoggingEvent_CL
| where TimeGenerated > ago(1d)
| where Level == "ERROR"
```

Accumulation of malformed fare and ride data

Kusto

```
SparkMetric_CL
| where TimeGenerated > ago(1d)
| where name_s contains "metrics.malformedrides"
| project value_d, TimeGenerated, applicationId_s
| render timechart
```

```
SparkMetric_CL
| where TimeGenerated > ago(1d)
| where name_s contains "metrics.malformedfares"
| project value_d, TimeGenerated, applicationId_s
| render timechart
```

Job execution over time

Kusto

```
SparkMetric_CL
| where TimeGenerated > ago(1d)
| where name_s contains "driver.DAGScheduler.job.allJobs"
| project value_d, TimeGenerated, applicationId_s
| render timechart
```

For more information, see [Monitoring Azure Databricks](#).

DevOps

- Create separate resource groups for production, development, and test environments. Separate resource groups make it easier to manage deployments, delete test deployments, and assign access rights.
- Use [Azure Resource Manager template](#) to deploy the Azure resources following the infrastructure as Code (IaC) Process. With templates, automating deployments using [Azure DevOps Services](#), or other CI/CD solutions is easier.
- Put each workload in a separate deployment template and store the resources in source control systems. You can deploy the templates together or individually as part of a CI/CD process, making the automation process easier.

In this architecture, Azure Event Hubs, Log Analytics, and Azure Cosmos DB are identified as a single workload. These resources are included in a single ARM template.

- Consider staging your workloads. Deploy to various stages and run validation checks at each stage before moving to the next stage. That way you can push updates to your production environments in a highly controlled way and minimize unanticipated deployment issues.

In this architecture there are multiple deployment stages. Consider creating an Azure DevOps Pipeline and adding those stages. Here are some examples of

stages that you can automate:

- Start a Databricks Cluster
- Configure Databricks CLI
- Install Scala Tools
- Add the Databricks secrets

Also, consider writing automated integration tests to improve the quality and the reliability of the Databricks code and its life cycle.

- Consider using [Azure Monitor](#) to analyze the performance of your stream processing pipeline. For more information, see [Monitoring Azure Databricks](#).

For more information, see the DevOps section in [Microsoft Azure Well-Architected Framework](#).

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

Use the [Azure pricing calculator](#) to estimate costs. Here are some considerations for services used in this reference architecture.

Event Hubs

This reference architecture deploys Event Hubs in the **Standard** tier. The pricing model is based on throughput units, ingress events, and capture events. An ingress event is a unit of data 64 KB or less. Larger messages are billed in multiples of 64 KB. You specify throughput units either through the Azure portal or Event Hubs management APIs.

If you need more retention days, consider the **Dedicated** tier. This tier offers single-tenant deployments with most demanding requirements. This offering builds a cluster based on capacity units (CU) that is not bound by throughput units.

The **Standard** tier is also billed based on ingress events and throughput units.

For information about Event Hubs pricing, see the [Event Hubs pricing](#).

Azure Databricks

Azure Databricks offers two tiers **Standard** and **Premium** each supports three workloads. This reference architecture deploys Azure Databricks workspace in the **Premium** tier.

Data Engineering and **Data Engineering Light** workloads are for data engineers to build and execute jobs. The **Data Analytics** workload is intended for data scientists to explore, visualize, manipulate, and share data and insights interactively.

Azure Databricks offers many pricing models.

- Pay-as-you-go plan

You are billed for virtual machines (VMs) provisioned in clusters and Databricks Units (DBUs) based on the VM instance selected. A DBU is a unit of processing capability, billed on a per-second usage. The DBU consumption depends on the size and type of instance running Azure Databricks. Pricing will depend on the selected workload and tier.

- Pre-purchase plan

You commit to Azure Databricks Units (DBU) as Databricks Commit Units (DBCU) for either one or three years. When compared to the pay-as-you-go model, you can save up to 37%.

For more information, see [Azure Databricks Pricing](#).

Azure Cosmos DB

In this architecture, a series of records is written to Azure Cosmos DB by the Azure Databricks job. You are charged for the capacity that you reserve, expressed in Request Units per second (RU/s), used to perform insert operations. The unit for billing is 100 RU/sec per hour. For example, the cost of writing 100-KB items is 50 RU/s.

For write operations, provision enough capacity to support the number of writes needed per second. You can increase the provisioned throughput by using the portal or Azure CLI before performing write operations and then reduce the throughput after those operations are complete. Your throughput for the write period is the minimum throughput needed for the given data plus the throughput required for the insert operation assuming no other workload is running.

Example cost analysis

Suppose you configure a throughput value of 1,000 RU/sec on a container. It's deployed for 24 hours for 30 days, a total of 720 hours.

The container is billed at 10 units of 100 RU/sec per hour for each hour. 10 units at \$0.008 (per 100 RU/sec per hour) are charged \$0.08 per hour.

For 720 hours or 7,200 units (of 100 RUs), you are billed \$57.60 for the month.

Storage is also billed, for each GB used for your stored data and index. For more information, see [Azure Cosmos DB pricing model](#).

Use the [Azure Cosmos DB capacity calculator](#) to get a quick estimate of the workload cost.

For more information, see the cost section in [Microsoft Azure Well-Architected Framework](#).

Deploy this scenario

To the deploy and run the reference implementation, follow the steps in the [GitHub readme](#).

Next steps

- [Stream processing with Azure Stream Analytics](#)
- [Demand Forecasting](#)
- [Real Time Analytics on Big Data Architecture](#)

Stream processing with Azure Stream Analytics

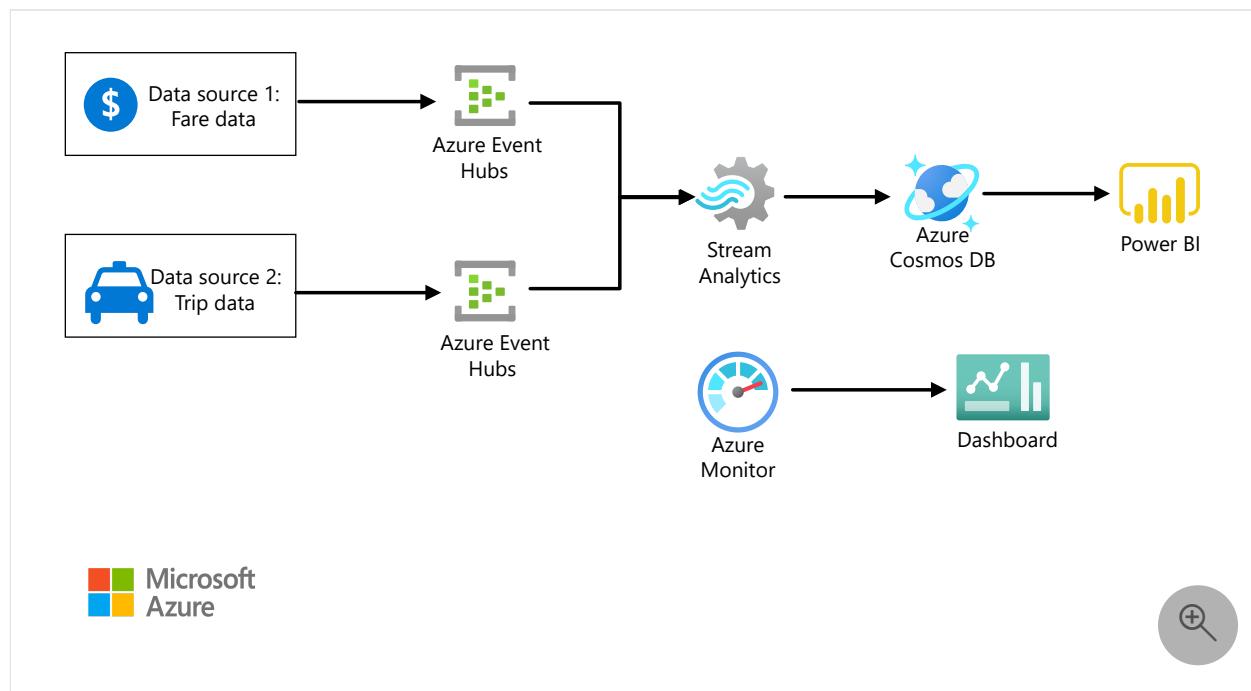
Azure Cosmos DB Azure Event Hubs Azure Monitor Azure Stream Analytics

This reference architecture shows an end-to-end stream processing pipeline. The pipeline ingests data from two sources, correlates records in the two streams, and calculates a rolling average across a time window. The results are stored for further analysis.



A reference implementation for this architecture is available on [GitHub](#).

Architecture



Download a [Visio file](#) of this architecture.

Workflow

The architecture consists of the following components:

Data sources. In this architecture, there are two data sources that generate data streams in real time. The first stream contains ride information, and the second contains fare information. The reference architecture includes a simulated data generator that reads

from a set of static files and pushes the data to Event Hubs. In a real application, the data sources would be devices installed in the taxi cabs.

Azure Event Hubs. [Event Hubs](#) is an event ingestion service. This architecture uses two event hub instances, one for each data source. Each data source sends a stream of data to the associated event hub.

Azure Stream Analytics. [Stream Analytics](#) is an event-processing engine. A Stream Analytics job reads the data streams from the two event hubs and performs stream processing.

Azure Cosmos DB. The output from the Stream Analytics job is a series of records, which are written as JSON documents to an Azure Cosmos DB document database.

Microsoft Power BI. Power BI is a suite of business analytics tools to analyze data for business insights. In this architecture, it loads the data from Azure Cosmos DB. This allows users to analyze the complete set of historical data that's been collected. You could also stream the results directly from Stream Analytics to Power BI for a real-time view of the data. For more information, see [Real-time streaming in Power BI](#).

Azure Monitor. [Azure Monitor](#) collects performance metrics about the Azure services deployed in the solution. By visualizing these in a dashboard, you can get insights into the health of the solution.

Scenario details

Scenario: A taxi company collects data about each taxi trip. For this scenario, we assume there are two separate devices sending data. The taxi has a meter that sends information about each ride — the duration, distance, and pickup and dropoff locations. A separate device accepts payments from customers and sends data about fares. The taxi company wants to calculate the average tip per mile driven, in real time, in order to spot trends.

Potential use cases

This solution is optimized for the retail scenario.

Data ingestion

To simulate a data source, this reference architecture uses the [New York City Taxi Data](#) dataset^[1]. This dataset contains data about taxi trips in New York City over a four-year period (2010–2013). It contains two types of record: ride data and fare data. Ride data

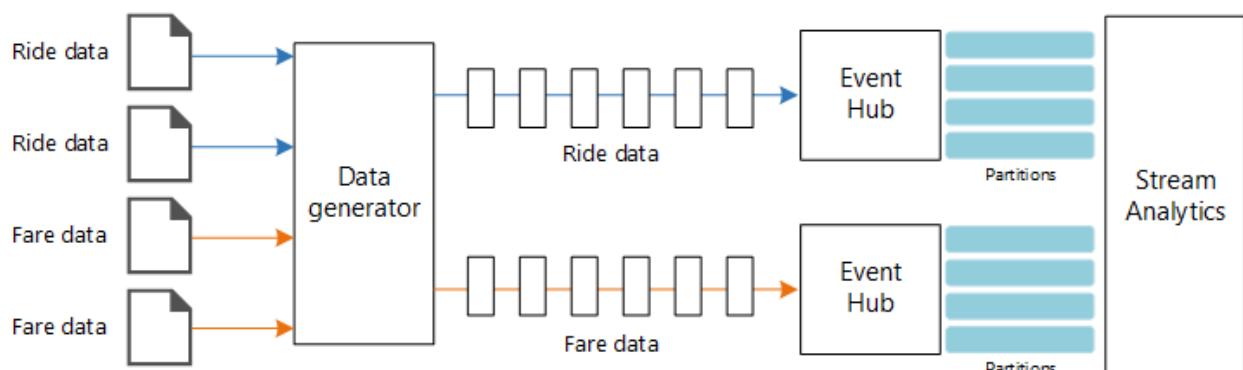
includes trip duration, trip distance, and pickup and dropoff location. Fare data includes fare, tax, and tip amounts. Common fields in both record types include medallion number, hack license, and vendor ID. Together these three fields uniquely identify a taxi plus a driver. The data is stored in CSV format.

[1] Donovan, Brian; Work, Dan (2016): New York City Taxi Trip Data (2010-2013). University of Illinois at Urbana-Champaign. <https://doi.org/10.13012/J8PN93H8>

The data generator is a .NET Core application that reads the records and sends them to Azure Event Hubs. The generator sends ride data in JSON format and fare data in CSV format.

Event Hubs uses [partitions](#) to segment the data. Partitions allow a consumer to read each partition in parallel. When you send data to Event Hubs, you can specify the partition key explicitly. Otherwise, records are assigned to partitions in round-robin fashion.

In this particular scenario, ride data and fare data should end up with the same partition ID for a given taxi cab. This enables Stream Analytics to apply a degree of parallelism when it correlates the two streams. A record in partition n of the ride data will match a record in partition n of the fare data.



In the data generator, the common data model for both record types has a `PartitionKey` property which is the concatenation of `Medallion`, `HackLicense`, and `VendorId`.

C#

```
public abstract class TaxiData
{
    public TaxiData()
    {
    }

    [JsonProperty]
    public long Medallion { get; set; }
```

```

[JsonProperty]
public long HackLicense { get; set; }

[JsonProperty]
public string VendorId { get; set; }

[JsonProperty]
public DateTimeOffset PickupTime { get; set; }

[JsonIgnore]
public string PartitionKey
{
    get => $"{Medallion}_{HackLicense}_{VendorId}";
}

```

This property is used to provide an explicit partition key when sending to Event Hubs:

C#

```

using (var client = pool.GetObject())
{
    return client.Value.SendAsync(new EventData(Encoding.UTF8.GetBytes(
        t.GetData(dataFormat))), t.PartitionKey);
}

```

Stream processing

The stream processing job is defined using a SQL query with several distinct steps. The first two steps simply select records from the two input streams.

SQL

```

WITH
Step1 AS (
    SELECT PartitionId,
        TRY_CAST(Medallion AS nvarchar(max)) AS Medallion,
        TRY_CAST(HackLicense AS nvarchar(max)) AS HackLicense,
        VendorId,
        TRY_CAST(PickupTime AS datetime) AS PickupTime,
        TripDistanceInMiles
    FROM [TaxiRide] PARTITION BY PartitionId
),
Step2 AS (
    SELECT PartitionId,
        medallion AS Medallion,
        hack_license AS HackLicense,
        vendor_id AS VendorId,
        TRY_CAST(pickup_datetime AS datetime) AS PickupTime,
        tip_amount AS TipAmount
)

```

```
    FROM [TaxiFare] PARTITION BY PartitionId
  ),
```

The next step joins the two input streams to select matching records from each stream.

SQL

```
Step3 AS (
  SELECT tr.TripDistanceInMiles,
         tf.TipAmount
    FROM [Step1] tr
  PARTITION BY PartitionId
  JOIN [Step2] tf PARTITION BY PartitionId
    ON tr.PartitionId = tf.PartitionId
   AND tr.PickupTime = tf.PickupTime
   AND DATEDIFF(minute, tr, tf) BETWEEN 0 AND 15
)
```

This query joins records on a set of fields that uniquely identify matching records (`PartitionId` and `PickupTime`).

(!) Note

We want the `TaxiRide` and `TaxiFare` streams to be joined by the unique combination of `Medallion`, `HackLicense`, `VendorId` and `PickupTime`. In this case the `PartitionId` covers the `Medallion`, `HackLicense` and `VendorId` fields, but this should not be taken as generally the case.

In Stream Analytics, joins are *temporal*, meaning records are joined within a particular window of time. Otherwise, the job might need to wait indefinitely for a match. The `DATEDIFF` function specifies how far two matching records can be separated in time for a match.

The last step in the job computes the average tip per mile, grouped by a hopping window of 5 minutes.

SQL

```
SELECT System.Timestamp AS WindowTime,
       SUM(tr.TipAmount) / SUM(tr.TripDistanceInMiles) AS AverageTipPerMile
    INTO [TaxiDrain]
   FROM [Step3] tr
  GROUP BY HoppingWindow(Duration(minute, 5), Hop(minute, 1))
```

Stream Analytics provides several [windowing functions](#). A hopping window moves forward in time by a fixed period, in this case 1 minute per hop. The result is to calculate a moving average over the past 5 minutes.

In the architecture shown here, only the results of the Stream Analytics job are saved to Azure Cosmos DB. For a big data scenario, consider also using [Event Hubs Capture](#) to save the raw event data into Azure Blob storage. Keeping the raw data will allow you to run batch queries over your historical data at later time, in order to derive new insights from the data.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Scalability

Event Hubs

The throughput capacity of Event Hubs is measured in [throughput units](#). You can autoscale an event hub by enabling [auto-inflate](#), which automatically scales the throughput units based on traffic, up to a configured maximum.

Stream Analytics

For Stream Analytics, the computing resources allocated to a job are measured in Streaming Units. Stream Analytics jobs scale best if the job can be parallelized. That way, Stream Analytics can distribute the job across multiple compute nodes.

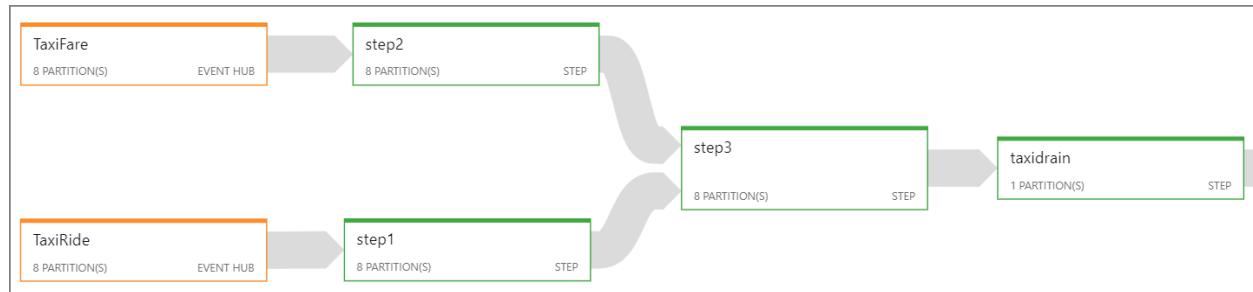
For Event Hubs input, use the `PARTITION BY` keyword to partition the Stream Analytics job. The data will be divided into subsets based on the Event Hubs partitions.

Windowing functions and temporal joins require additional SU. When possible, use `PARTITION BY` so that each partition is processed separately. For more information, see [Understand and adjust Streaming Units](#).

If it's not possible to parallelize the entire Stream Analytics job, try to break the job into multiple steps, starting with one or more parallel steps. That way, the first steps can run in parallel. For example, in this reference architecture:

- Steps 1 and 2 are simple `SELECT` statements that select records within a single partition.
- Step 3 performs a partitioned join across two input streams. This step takes advantage of the fact that matching records share the same partition key, and so are guaranteed to have the same partition ID in each input stream.
- Step 4 aggregates across all of the partitions. This step cannot be parallelized.

Use the Stream Analytics [job diagram](#) to see how many partitions are assigned to each step in the job. The following diagram shows the job diagram for this reference architecture:



Azure Cosmos DB

Throughput capacity for Azure Cosmos DB is measured in [Request Units](#) (RU). In order to scale an Azure Cosmos DB container past 10,000 RU, you must specify a [partition key](#) when you create the container, and include the partition key in every document.

In this reference architecture, new documents are created only once per minute (the hopping window interval), so the throughput requirements are quite low. For that reason, there's no need to assign a partition key in this scenario.

Monitoring

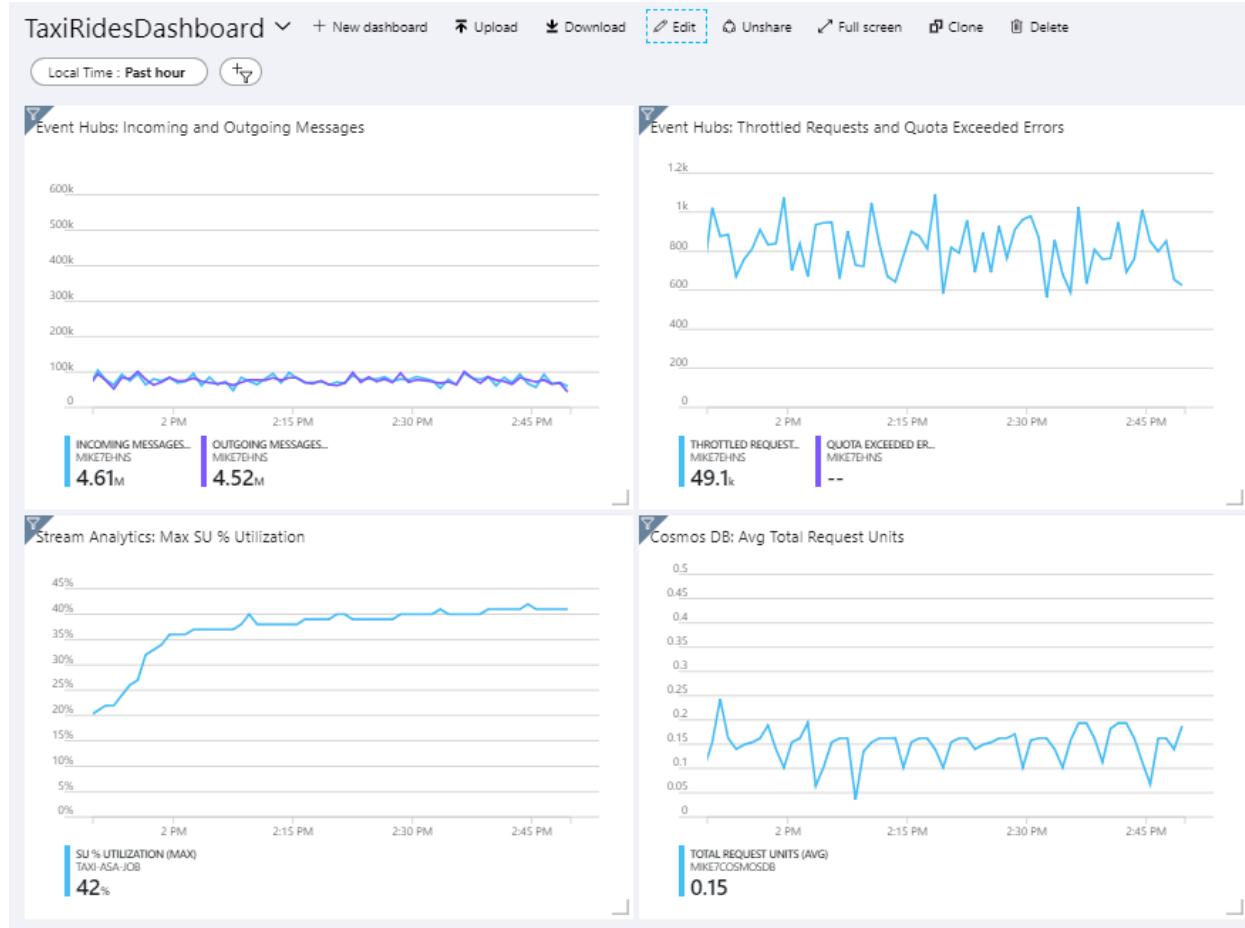
With any stream processing solution, it's important to monitor the performance and health of the system. [Azure Monitor](#) collects metrics and diagnostics logs for the Azure services used in the architecture. Azure Monitor is built into the Azure platform and does not require any additional code in your application.

Any of the following warning signals indicate that you should scale out the relevant Azure resource:

- Event Hubs throttles requests or is close to the daily message quota.
- The Stream Analytics job consistently uses more than 80% of allocated Streaming Units (SU).
- Azure Cosmos DB begins to throttle requests.

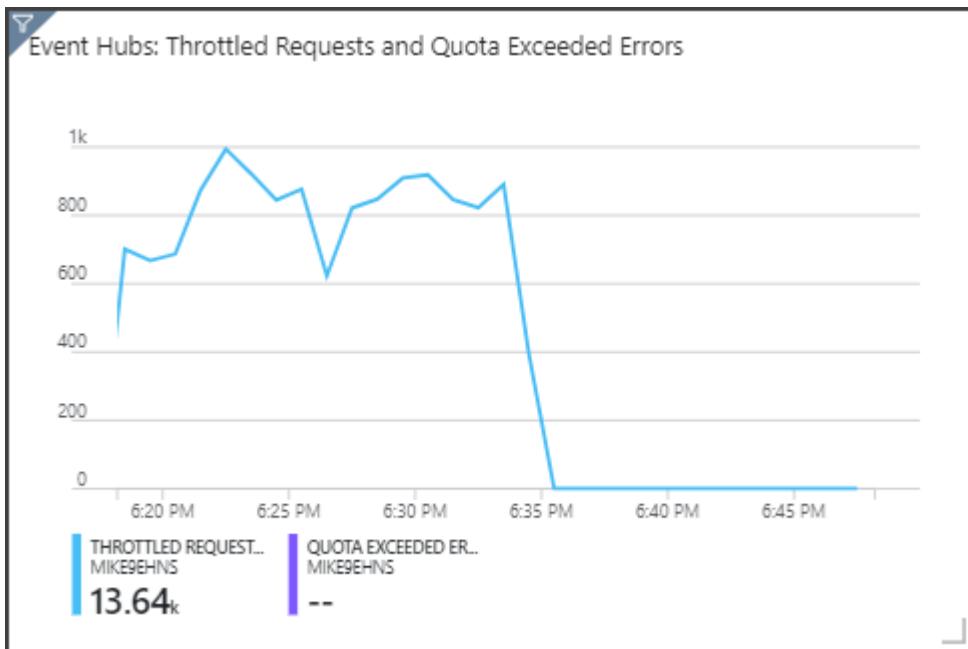
The reference architecture includes a custom dashboard, which is deployed to the Azure portal. After you deploy the architecture, you can view the dashboard by opening the [Azure portal](#) and selecting `TaxiRidesDashboard` from list of dashboards. For more information about creating and deploying custom dashboards in the Azure portal, see [Programmatically create Azure Dashboards](#).

The following image shows the dashboard after the Stream Analytics job ran for about an hour.



The panel on the lower left shows that the SU consumption for the Stream Analytics job climbs during the first 15 minutes and then levels off. This is a typical pattern as the job reaches a steady state.

Notice that Event Hubs is throttling requests, shown in the upper right panel. An occasional throttled request is not a problem, because the Event Hubs client SDK automatically retries when it receives a throttling error. However, if you see consistent throttling errors, it means the event hub needs more throughput units. The following graph shows a test run using the Event Hubs auto-inflate feature, which automatically scales out the throughput units as needed.



Auto-inflate was enabled at about the 06:35 mark. You can see the drop in throttled requests, as Event Hubs automatically scaled up to 3 throughput units.

Interestingly, this had the side effect of increasing the SU utilization in the Stream Analytics job. By throttling, Event Hubs was artificially reducing the ingestion rate for the Stream Analytics job. It's actually common that resolving one performance bottleneck reveals another. In this case, allocating additional SU for the Stream Analytics job resolved the issue.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

Use the [Azure pricing calculator](#) to estimate costs. Here are some considerations for services used in this reference architecture.

Azure Stream Analytics

Azure Stream Analytics is priced by the number of streaming units (\$0.11/hour) required to process the data into the service.

Stream Analytics can be expensive if you are not processing the data in real-time or small amounts of data. For those use cases, consider using Azure Functions or Logic Apps to move data from Azure Event Hubs to a data store.

Azure Event Hubs and Azure Cosmos DB

For cost considerations about Azure Event Hubs and Azure Cosmos DB, see Cost considerations see the [Stream processing with Azure Databricks](#) reference architecture.

DevOps

- Create separate resource groups for production, development, and test environments. Separate resource groups make it easier to manage deployments, delete test deployments, and assign access rights.
- Use [Azure Resource Manager template](#) to deploy the Azure resources following the infrastructure as Code (IaC) Process. With templates, automating deployments using [Azure DevOps Services](#), or other CI/CD solutions is easier.
- Put each workload in a separate deployment template and store the resources in source control systems. You can deploy the templates together or individually as part of a CI/CD process, making the automation process easier.

In this architecture, Azure Event Hubs, Log Analytics, and Azure Cosmos DB are identified as a single workload. These resources are included in a single ARM template.

- Consider staging your workloads. Deploy to various stages and run validation checks at each stage before moving to the next stage. That way you can push updates to your production environments in a highly controlled way and minimize unanticipated deployment issues.
- Consider using [Azure Monitor](#) to analyze the performance of your stream processing pipeline. For more information, see [Monitoring Azure Databricks](#).

For more information, see the operational excellence pillar in [Microsoft Azure Well-Architected Framework](#).

Deploy this scenario

To the deploy and run the reference implementation, follow the steps in the [GitHub readme](#).

Related resources

You may want to review the following [Azure example scenarios](#) that demonstrate specific solutions using some of the same technologies:

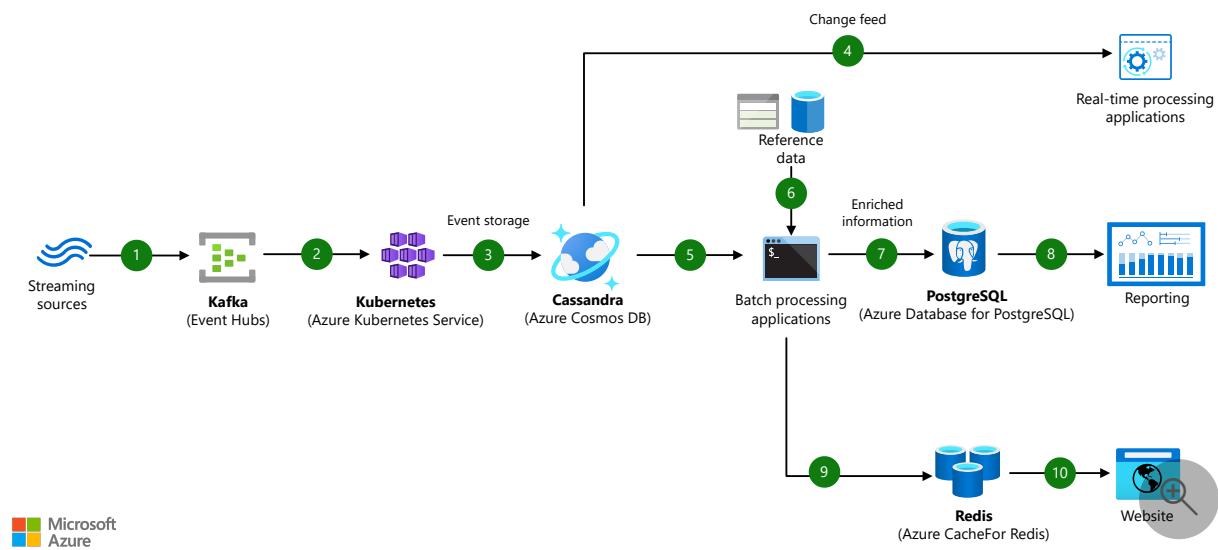
- IoT and data analytics in the construction industry
- Real-time fraud detection

Stream processing with fully managed open-source data engines

Azure Event Hubs Azure Kubernetes Service (AKS) Azure Cosmos DB Azure Database for PostgreSQL
Azure Cache for Redis

This article presents an example of a streaming solution that uses fully managed Azure data services.

Architecture



Download a [Visio file](#) of this architecture.

Workflow

1. The Event Hubs for Apache Kafka feature streams events from Kafka producers.
2. Apache Spark consumes events. AKS provides a managed environment for the Apache Spark jobs.
3. An application that uses Azure Cosmos DB for Apache Cassandra writes events to Cassandra. This database serves as a storage platform for events. AKS hosts the microservices that write to Cassandra.
4. The change feed feature of Azure Cosmos DB processes events in real time.

5. Scheduled applications run batch-oriented processing on events stored in Cassandra.
6. Stores of reference data enrich event information. Batch-oriented applications write the enriched event information to PostgreSQL. Typical reference data stores include:
 - [Azure Data Lake Storage](#), which can store data in open formats such as [Parquet](#).
 - Open-source relational data stores like [PostgreSQL](#) and [MySQL](#).
7. A batch-oriented application processes Cassandra data. That application stores the processed data in Azure Database for PostgreSQL. This relational data store provides data to downstream applications that require enriched information.
8. Reporting applications and tools analyze the PostgreSQL database data. For example, [Power BI](#) connects to the database by using the Azure Database for PostgreSQL connector. This reporting service then displays rich visuals of the data.
9. Azure Cache for Redis provides an in-memory cache. In this solution, the cache contains data on critical events. An application stores data to the cache and retrieves data from the cache.
10. Websites and other applications use the cached data to improve response times. Sometimes data isn't available in the cache. In those cases, these applications use the [cache-aside pattern](#) or a similar strategy to retrieve data from Cassandra in Azure Cosmos DB.

Components

- [Event Hubs](#) is a fully managed streaming platform that can process millions of events per second. Event Hubs provides an [endpoint](#) for [Apache Kafka](#), a widely used open-source stream-processing platform. When organizations use the endpoint feature, they don't need to build and maintain Kafka clusters for stream processing. Instead, they can benefit from the fully managed Kafka implementation that Event Hubs offers.
- [Azure Cosmos DB](#) is a fully managed NoSQL and relational database that offers multi-master replication. Azure Cosmos DB supports open-source APIs for many databases, languages, and platforms. Examples include:
 - [Apache Cassandra](#).
 - [Gremlin](#).
 - [MongoDB](#).

Through the [Azure Cosmos DB for Apache Cassandra](#), you can access Azure Cosmos DB data by using Apache Cassandra tools, languages, and drivers. Apache Cassandra is an open-source NoSQL database that's well suited for heavy write-intensive workloads.

- [AKS](#) is a highly available, secure, and fully managed Kubernetes service. [Kubernetes](#) is a rapidly evolving open-source platform for managing containerized workloads. [AKS](#) hosts open-source big data processing engines such as [Apache Spark](#). By using AKS, you can run large-scale stream processing jobs in a managed environment.
- [Azure Database for PostgreSQL](#) is a fully managed relational database service. It provides [high availability, elastic scaling, patching, and other management capabilities](#) for PostgreSQL. [PostgreSQL](#) is a widely adopted open-source relational database management system.
- [Azure Cache for Redis](#) provides an in-memory data store based on the Redis software. [Redis](#) is a popular open-source in-memory data store. Session stores, content caches, and other storage components use Redis to improve performance and scalability. [Azure Cache for Redis](#) provides open-source Redis capabilities as a fully managed offering.

Alternatives

You can replace the open-source-compatible products and services in this solution with others. For details on open-source services available in Azure, see [Open source on Azure](#).

Scenario details

Fully managed Azure data services that run open-source engines make up this streaming solution:

- Azure Event Hubs offers a [Kafka](#) implementation for stream ingestion.
- Azure Cosmos DB supports event storage in [Cassandra](#).
- Azure Kubernetes Service (AKS) hosts [Kubernetes](#) microservices for stream processing.
- Azure Database for PostgreSQL manages relational data storage in [PostgreSQL](#).
- Azure Cache for Redis manages [Redis](#) in-memory data stores.

Open-source technologies offer many benefits. For instance, organizations can use open-source technologies to:

- Migrate existing workloads.
- Tap into the broad open-source community.
- Limit vendor lock-in.

By making open-source technologies accessible, Azure tools and services help organizations take advantage of these benefits and develop the solutions of their choice.

This solution uses fully managed [platform as a service \(PaaS\)](#) services. As a result, Microsoft handles patching, service-level agreement (SLA) maintenance, and other management tasks. Another benefit is the native integration with the Azure security infrastructure.

Potential use cases

This solution applies to various scenarios:

- Using Azure PaaS services to build modern streaming solutions that use open-source technologies
- Migrating open-source stream processing solutions to Azure

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Design and implement each service with best practices in mind. For guidelines on each service, see the [Microsoft documentation site](#). Also review the information in the following sections:

Performance

- Implement [connection pooling for Azure Database for PostgreSQL](#). You can use a connection pooling library within the application. Or you can use a connection pooler such as [PgBouncer](#) or [Pgpool](#). Establishing a connection with PostgreSQL is an expensive operation. With connection pooling, you can avoid degrading application performance. PgBouncer is [built-in](#) in Azure Database for PostgreSQL Flexible Server.
- Configure Azure Cosmos DB for Apache Cassandra for best performance by using an appropriate [partitioning strategy](#). Decide whether to use a single field primary

key, a compound primary key, or a composite partition key when partitioning tables.

Scalability

- Take your streaming requirements into account when choosing an [Event Hubs tier](#):
 - For mid-range throughput requirements of less than 120 MBps, consider the [Premium tier](#). This tier scales elastically to meet streaming requirements.
 - For high-end streaming workloads with an ingress of gigabytes of data, consider the [Dedicated tier](#). This tier is a single-tenant offering with a guaranteed capacity. You can scale dedicated clusters up and down.
- Consider [autoscale-provisioned throughput](#) for Azure Cosmos DB if your workloads are unpredictable and spiky. You can configure Azure Cosmos DB to use manually provisioned throughput or autoscale-provisioned throughput. With autoscale, Azure automatically and instantly scales the request units per second according to your usage.

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](#).

- Use [Azure Private Link](#) to make Azure services part of your virtual network. When you use Private Link, traffic between the services and your network flows over the Azure backbone without traversing the public internet. The Azure services in this solution support Private Link for selected SKUs.
- Check your organization's security policies. With Azure Cosmos DB for Apache Cassandra, keys provide access to resources like key spaces and tables. The Azure Cosmos DB instance stores those keys. Your security policies might require you to [propagate those keys to a key management service](#) such as [Azure Key Vault](#). Also make sure to [rotate keys](#) according to your organization's policies.

Resiliency

Consider using [Availability zones](#) to protect business-critical applications from datacenter failures. This solution's services support availability zones for selected SKUs in [availability zone-enabled regions](#). For up-to-date information, review the [list of services that support availability zones](#).

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

To estimate the cost of this solution, use the [Azure pricing calculator](#). Also keep these points in mind:

- [Event Hubs](#) is available in Basic, Standard, Premium, and Dedicated tiers. The Premium or Dedicated tier is best for large-scale streaming workloads. You can scale throughput, so consider starting small and then scaling up as demand increases.
- [Azure Cosmos DB](#) offers two models:
 - A provisioned throughput model that's ideal for demanding workloads. This model is available in two capacity management options: standard and autoscale.
 - A serverless model that's well suited for running small, spiky workloads.
- An [AKS](#) cluster consists of a set of nodes, or virtual machines (VMs), that run in Azure. The cost of the compute, storage, and networking components make up a cluster's primary costs.
- [Azure Database for PostgreSQL](#) is available in Single Server and Flexible Server tiers. Different tiers cater to different scenarios, such as predictable, burstable, and high-performance workloads. The costs mainly depend on the choice of compute nodes and storage capacity. For new workloads, consider choosing the Flexible Server tier since it has a wider range of [supported capabilities](#) over the Single Server tier.
- [Azure Cache for Redis](#) is available in multiple tiers. These tiers accommodate caches that range from 250 megabytes to several terabytes. Besides size, other requirements also affect the choice of tier:
 - Clustering
 - Persistence
 - Active geo-replication

Deploy this scenario

Keep these points in mind when you deploy this solution:

- When you deploy Event Hubs for Kafka, refer to [Quickstart: Data streaming with Event Hubs using the Kafka protocol](#). This article provides the following information:
 - How to send and receive messages with Kafka in Event Hubs
 - Sample code for a publishing application
 - How to switch existing Kafka applications to Event Hubs for Kafka by making configuration changes
- Concerning Apache Spark:
 - For information on building a basic Spark application, see [Connect your Apache Spark application with Azure Event Hubs](#).
 - To host the Spark application on AKS, see [Running Apache Spark jobs on AKS](#).
- Consider using a Java application to write events to Cassandra. For more information, see [Quickstart: Build a Java app to manage Azure Cosmos DB for Apache Cassandra data \(v4 Driver\)](#).
- When you use the [Azure Cosmos DB change feed](#), refer to [Change feed in Azure Cosmos DB for Apache Cassandra](#) for this information:
 - How to use query predicates in [Cassandra Query Language \(CQL\)](#) to query the change feed API
 - Sample code for a Java application
- For information on processing the events that you've stored in Cassandra, refer to [Tutorial: Query data from Azure Cosmos DB for Apache Cassandra](#). This article also contains sample Java code for using CQL commands to retrieve data from tables.
- For information on writing data to Azure Database for PostgreSQL with a batch-oriented application, see [Quickstart: Use Java and JDBC with Azure Database for PostgreSQL](#). This article also contains sample Java code for storing data.
- For information on data storage and retrieval with Azure Cache for Redis, see [Quickstart: Use Azure Cache for Redis in Java](#). This article also contains sample Java code for accessing a cache.

Contributors

This article is being updated and maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Ajit Ananthram](#) | Cloud Solution Architect

Next steps

- [Apache Kafka developer guide for Azure Event Hubs](#)
- [Frequently asked questions about Azure Cosmos DB for Apache Cassandra](#)
- [Best practices for building an application with Azure Database for PostgreSQL](#)
- [Azure Cache for Redis FAQ](#)

Related resources

To learn about related solutions, see the following information:

- [Analytics architecture design](#)
- [Choose an analytical data store in Azure](#)
- [Choose a data analytics technology in Azure](#)
- [Azure Kubernetes in event stream processing](#)
- [Data streaming with AKS](#)

Advanced analytics architecture

Azure Analysis Services

Azure Blob Storage

Azure Cosmos DB

Azure Synapse Analytics

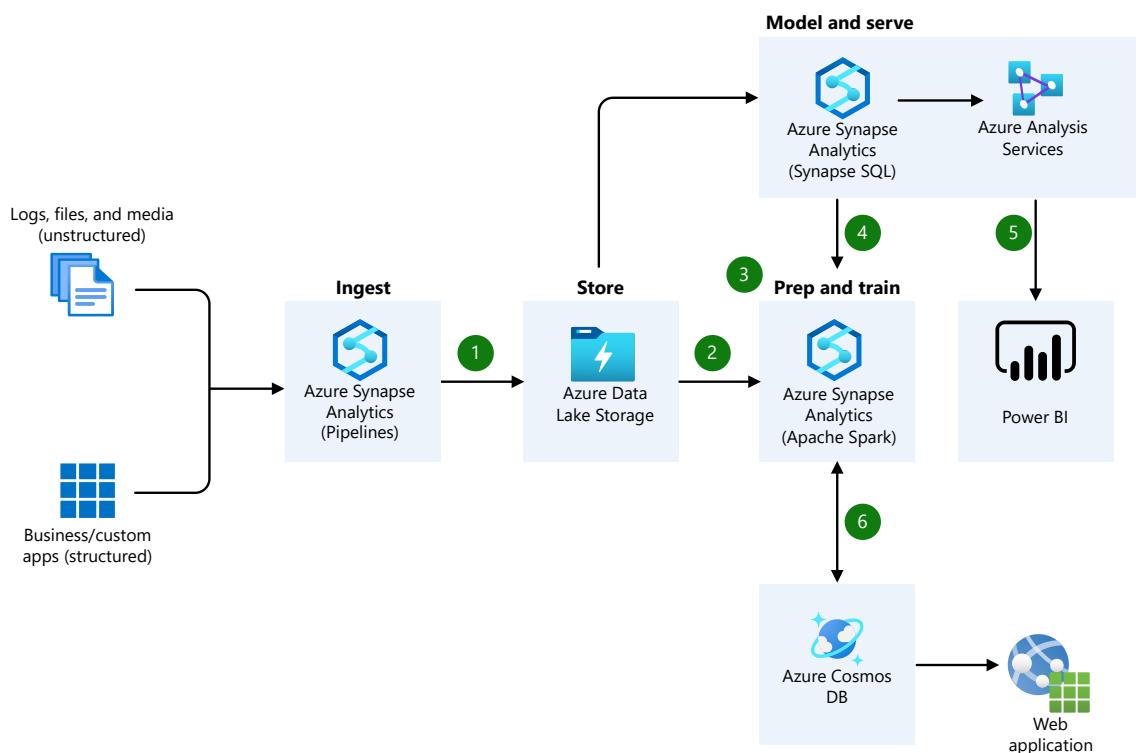
Power BI

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This architecture allows you to combine any data at any scale with custom machine learning and get near real-time data analytics on streaming services.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

1. Bring together all your structured, unstructured, and semi-structured data (logs, files, and media) using Synapse Pipelines to Azure Data Lake Storage.
2. Use Apache Spark pools to clean and transform the structureless datasets and combine them with structured data from operational databases or data warehouses.
3. Use scalable machine learning/deep learning techniques, to derive deeper insights from this data using Python, Scala, or .NET, with notebook experiences in Apache Spark pool.
4. Apply Apache Spark pool and Synapse Pipelines in Azure Synapse Analytics to access and move data at scale.
5. Query and report on data in [Power BI](#).
6. Take the insights from Apache Spark pools to Azure Cosmos DB to make them accessible through web and mobile apps.

Workflow

- [Azure Synapse Analytics](#) is the fast, flexible, and trusted cloud data warehouse that lets you scale, compute, and store elastically and independently, with a massively parallel processing architecture.
- [Synapse Pipelines Documentation](#) allows you to create, schedule, and orchestrate your ETL/ELT workflows.
- [Azure Blob storage](#) is a Massively scalable object storage for any type of unstructured data-images, videos, audio, documents, and more-easily and cost-effectively.
- [Azure Synapse Analytics Spark pools](#) is a fast, easy, and collaborative Apache Spark-based analytics platform.
- [Azure Cosmos DB](#) is a globally distributed, multi-model database service. Learn how to replicate your data across any number of Azure regions and scale your throughput independent from your storage.
- [Azure Synapse Link for Azure Cosmos DB](#) enables you to run near real-time analytics over operational data in Azure Cosmos DB, **without any performance or cost impact on your transactional workload**, by using the two analytics engines available from your Azure Synapse workspace: [SQL Serverless](#) and [Spark Pools](#).
- [Azure Analysis Services](#) is an enterprise grade analytics as a service that lets you govern, deploy, test, and deliver your BI solution with confidence.
- [Power BI](#) is a suite of business analytics tools that deliver insights throughout your organization. Connect to hundreds of data sources, simplify data prep, and drive unplanned analysis. Produce beautiful reports, then publish them for your organization to consume on the web and across mobile devices.

Alternatives

- [Synapse Link](#) is the Microsoft preferred solution for analytics on top of Azure Cosmos DB data.

Scenario details

Transform your data into actionable insights using the best-in-class machine learning tools. This solution allows you to combine any data at any scale, and to build and deploy custom machine learning models at scale. To learn how enterprise-scale data platforms are designed as part of an enterprise landing zone, refer to the [Cloud Adoption Framework Data landing zone](#) documentation.

Potential use cases

Organizations have the ability to access more data than ever before. Advanced analytics help take advantage of data insights. Areas include:

- Customer service.
- Predictive maintenance.
- Recommending products or services.
- System optimization of everything from supply chains to data center operations.
- Product and services development.

Considerations

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

- [Customize and get pricing estimates](#)

Next steps

- Learn about [enterprise-scale data platform](#) design
- Learn how to design and deploy an [end-to-end data analytics](#) platform

See the following documentation about the services featured in this architecture:

- [Synapse Analytics Documentation](#)
- [Synapse Pipelines Documentation](#)
- [Introduction to object storage in Azure](#)
- [Azure Synapse Analytics Spark pools](#)
- [Azure Cosmos DB Documentation](#)
- [Analysis Services Documentation](#)
- [Power BI Documentation](#)

Related resources

- [Secure a data lakehouse with Azure Synapse Analytics](#)
- [Analyze operational data on MongoDB Atlas using Azure Synapse Analytics](#)
- [Big data analytics with enterprise-grade security using Azure Synapse](#)
- [Spaceborne data analysis with Azure Synapse Analytics](#)
- [Analytics end-to-end with Azure Synapse](#)

Apache NiFi monitoring with MonitoFi

Azure Container Instances

Azure Container Registry

Azure Monitor

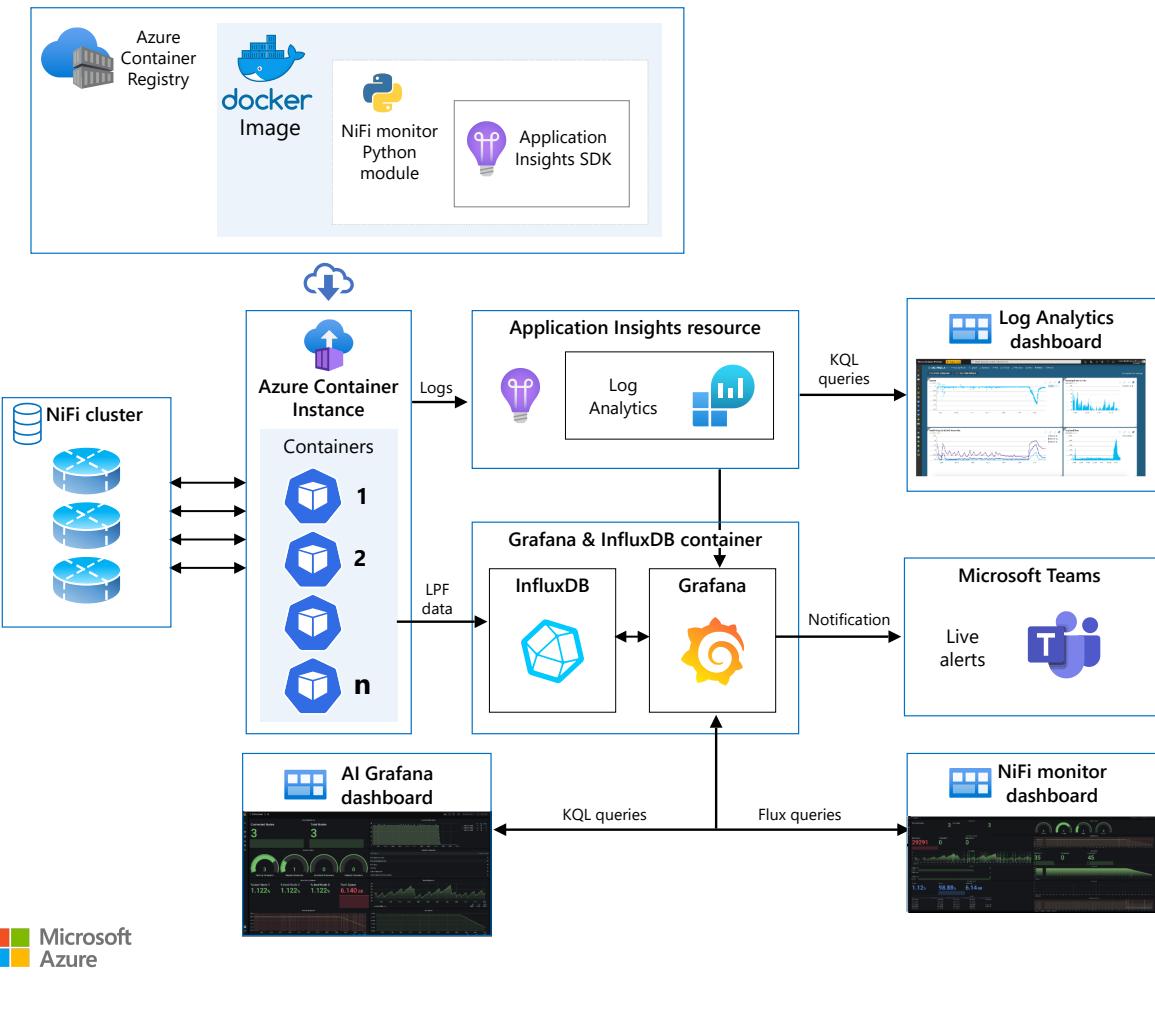
💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This solution monitors deployments of Apache NiFi on Azure by using MonitoFi. The tool sends alerts and displays health and performance information in dashboards.

Apache®, Apache NiFi®, and NiFi® 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.

Workflow

- A Docker image encapsulates a MonitoFi Python module and the [Application Insights](#) SDK. This Docker image can be retrieved from the [Docker Hub registry](#) and stored in [Container Registry](#) for use and deployment.
- If [Container Instances](#) or [Docker](#) run on a local machine, the image can be retrieved to run instances of the MonitoFi container.
- Another container that hosts an InfluxDB server and a Grafana instance is deployed locally.
- The MonitoFi container collects information on each NiFi cluster's health and performance. The container requests data:
 - From clusters at configurable time intervals.
 - From various endpoints by using the [Apache NiFi REST API](#).
- The MonitoFi container converts the cluster data into these formats:
 - A structured log format. The container sends this data to Application Insights.

- InfluxDB line protocol. In air-gapped or on-premises environments, the container stores this data in a local instance of InfluxDB.
- Grafana displays Application Insights data. This data-monitoring tool:
 - Uses Monitor as a data source.
 - Runs [Kusto Query Language](#) queries. The Application Insights dashboard includes sample queries.
- Grafana is used to display data from the local instance of InfluxDB. For querying, Grafana uses the following languages:
 - The [Flux](#) query language
 - The [Influx Query Language \(InfluxQL\)](#)
- The Grafana notification system sends real-time alerts through email and [Microsoft Teams](#) when it detects anomalies in the cluster.

Components

- MonitoFi runs in a [Docker](#) container, separately from NiFi.
- [Azure Container Registry](#) and [Azure Container Instances](#) manage and run the container images.

Other architecture components include:

- [Application Insights](#). This [Azure Monitor](#) feature monitors application usage, availability, and performance.
- [Grafana](#). This open-source analysis tool displays data and sends alerts.
- [InfluxDB](#). This platform stores data locally.

Scenario details

[MonitoFi](#) is a tool that monitors the health and performance of [Apache NiFi](#) clusters. When you run [NiFi on Azure](#) and use MonitoFi:

- MonitoFi dashboards display historic information on the state of NiFi clusters.
- Real-time notifications alert users when anomalies are detected in clusters.

Key benefits

MonitoFi has these advantages:

- Lightweight and extensible: MonitoFi is a lightweight tool that runs externally. Because MonitoFi is based on Python and is containerized, you can extend it to

add features. A MonitoFi instance that runs in one container can target multiple NiFi clusters.

- Effective and useful: MonitoFi uses local instances of InfluxDB and Grafana to provide real-time monitoring and alerts. MonitoFi can monitor clusters with latencies as low as one second.
- Flexible and robust: MonitoFi uses a REST API wrapper to retrieve JSON data from NiFi. MonitoFi converts that data into a usable format that doesn't depend on specific endpoints or field names. As a result, when NiFi REST API responses change, you don't need to change MonitoFi code.
- Easy to adopt: You don't have to reconfigure NiFi clusters to monitor them.
- Easy to use: MonitoFi offers preset configurations. It also includes templates for Grafana dashboards that you can import without modification.
- Highly configurable: MonitoFi runs in a Docker container. You configure MonitoFi by using environment variables. You can easily configure the following settings and others at runtime:
 - Endpoints
 - Settings for secure access
 - Certificates
 - Instrumentation key settings
 - The collection interval

With one container image, you can target different NiFi clusters, configurations, and different instances of Application Insights or InfluxDB. To change targets, you change the runtime command.

Deploy this scenario

To deploy this solution, see [MonitoFi: Health & Performance Monitor for Apache NiFi on GitHub](#).

Deployment examples

- In air-gapped and on-premises environments, there's no access to the public internet. As a result, these systems deploy a local instance of InfluxDB with Grafana. This approach provides a storage solution for the data. The MonitoFi container uses the NiFi REST API over a private IP address to retrieve cluster data. The

container stores this data in InfluxDB. Grafana is used to display the InfluxDB data and send email and Teams messages to alert users.

- In public environments, the MonitoFi container uses the NiFi REST API to retrieve cluster data. The container then sends this data in a structured format to Application Insights. These environments also deploy a local instance of InfluxDB and a Grafana container. MonitoFi can store data in that instance of InfluxDB. Grafana is used to display the data and send email and Teams messages to alert users.

Deployment process

MonitoFi includes a fully automated deployment script that:

- Verifies prerequisites and installs missing dependencies.
- Deploys a MonitoFi Docker container.
- Deploys containers for InfluxDB and Grafana.
- Configures databases and a retention policy for InfluxDB.
- Configures a data source in Grafana for InfluxDB.
- Optionally configures a data source in Grafana for Monitor.
- Imports the MonitoFi dashboard into Grafana. Grafana uses this dashboard to access InfluxDB data.
- Optionally imports the Application Insights dashboard into Grafana. Grafana can use this dashboard to access Application Insights data.
- Configures a notification channel that Grafana uses for real-time Teams alerts.

Deployment considerations

When you deploy this solution, keep in mind the following prerequisites and limitations:

- MonitoFi needs access to the NiFi cluster. Use one of these approaches to provide that access:
 - Place MonitoFi in the same network as the NiFi cluster. Provide access through a private IP address.
 - Make the NiFi cluster publicly accessible over the internet.
- The NiFi cluster can be secure or unsecured. For signing in, secured clusters support certificates in PKCS #12 format. Mount this type of certificate in the MonitoFi container, and make the password available.
- One MonitoFi instance can monitor multiple NiFi clusters at the same time. Another possibility is using multiple MonitoFi containers. In this case, the

containers can monitor different REST API endpoints in the same cluster or in different clusters.

- If you use more than one MonitoFi instance, it's possible to store the MonitoFi data in one InfluxDB database or send it to one common Application Insights resource. Pre-set tags mark the data and provide a way to identify its source.
- InfluxDB and Grafana run within the same Docker container. To provide a way for MonitoFi to send data to this container, use one of these options:
 - Place the Docker container in the same network as the MonitoFi container.
 - Make the Docker container publicly available.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Muazma Zahid](#) | Principal PM Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [MonitoFi: Health & Performance Monitor for Apache NiFi on GitHub](#)
- [Docker Image with InfluxDB and Grafana](#)
- [MonitoFi: Health & Performance Monitor for Apache NiFi on Docker Hub](#)
- [Docker Image with InfluxDB and Grafana on Docker Hub](#)
- [NiFi Rest API 1.14.0](#)

Related resources

- [Apache NiFi on Azure](#)
- [Helm-based deployments for Apache NiFi](#)
- [Monitoring Azure Functions and Event Hubs](#)
- [Web application monitoring on Azure](#)

Application integration using Azure Event Grid

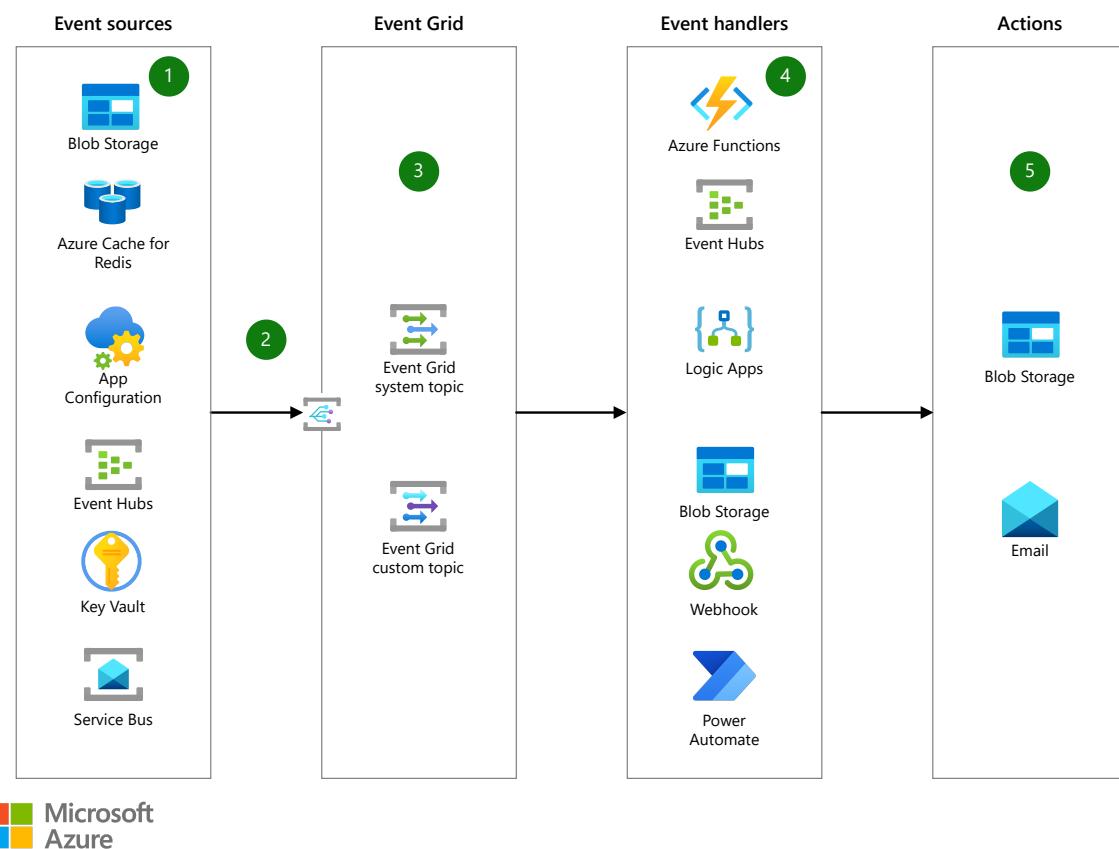
Azure Event Grid Azure Functions Azure Logic Apps

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

Event Grid connects your app with other services. This article describes how to integrate your app with Azure Event Grid to take advantage of its reliable delivery, advanced routing, and direct integration.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

1. Configure different event sources (Blob Storage, Azure Cache for Redis, App Configuration, Event Hubs, Key Vault, Service Bus) to subscribe to an Event Grid topic.
2. Triggers event sources from various scenarios, including different actions performed on the event sources.
3. Use Event Grid to support the events coming from different Azure services.
4. Leverages different event handlers (Azure Functions, Event Hubs, Logic Apps, Blob Storage, Web Hook, Power Automate) to handle different use cases.
5. Use different types of actions to handle the result of the event handlers, like Blob Storage and email for alerts.

Components

- [Azure Event Grid](#): Get reliable event delivery at massive scale.
- [Azure Blob Storage](#): A massively scalable object storage for any type of unstructured data, including images, videos, audio, documents, and more. It's easy and cost effective.
- [Azure Cache for Redis](#): A fully managed, open source-compatible, in-memory data store to power fast, scalable applications.
- [Azure App Configuration](#): Store configurations for all your Azure apps in a universal, hosted location.
- [Azure Event Hubs](#): Stream millions of events per second from any source to build dynamic data pipelines and immediately respond to business challenges.
- [Azure Key Vault](#): Safeguard cryptographic keys and other secrets that are used by cloud apps and services.
- [Azure Functions](#): An event-driven, serverless compute platform that can also solve complex orchestration problems.
- [Azure Logic Apps](#): Quickly build powerful integration solutions.
- [Web Hook](#): Event handling.
- [Power Automate](#): Easily create automated workflows.
- [Email](#): Create automated task and workflows with Azure Logic Apps and Microsoft 365 Outlook Connector to send an email.

Scenario details

Event Grid connects your app with other services. For example, create an application topic to send your app's event data to Event Grid and take advantage of its delivery,

advanced routing, and direct integration with Azure. Alternatively, you can use Event Grid with Logic Apps to process data anywhere, without writing code.

Potential use cases

Organizations can use Event Grid to assist with:

- Serverless application architectures in the cloud.
- Ops automation.
- Application integration.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

Use the [Azure pricing calculator](#) to get customized pricing estimates.

Next steps

- [Azure Event Grid documentation](#)
- [Azure Blob Storage documentation](#)
- [Azure Cache for Redis documentation](#)
- [Azure App Configuration documentation](#)
- [Azure Event Hubs documentation](#)
- [Azure Key Vault documentation](#)
- [Azure Functions documentation](#)
- [Azure Logic Apps documentation](#)
- [Power Automate documentation](#)

Related resources

- [Integration architecture design](#)

- Serverless application architectures using Event Grid
- Asynchronous messaging options

Big data analytics with Azure Data Explorer

Azure Data Explorer Azure Event Hubs Azure IoT Hub Azure Storage Azure Synapse Analytics

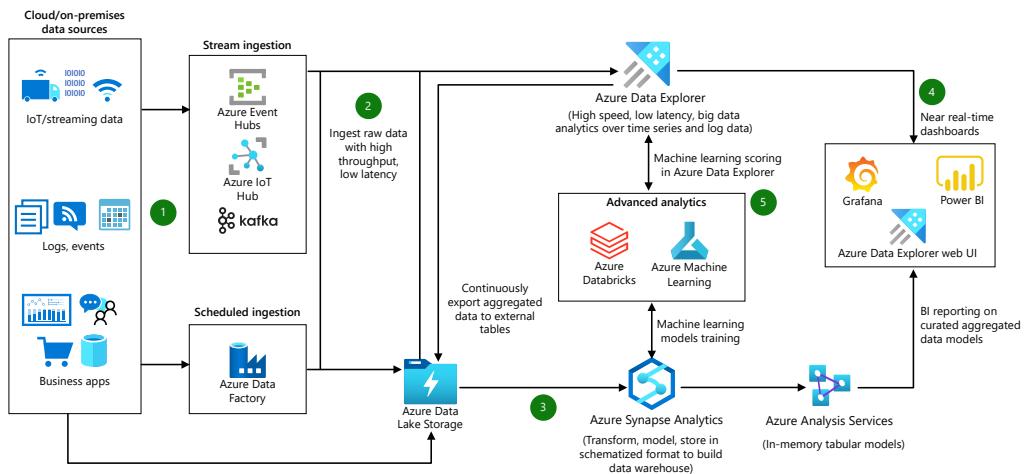
💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This solution idea demonstrates big data analytics over large volumes of high-velocity data from various sources.

Apache® and Apache Kafka® 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



Microsoft Azure



Download a [Visio file](#) of this architecture.

Dataflow

1. Raw structured, semi-structured, and unstructured (free text) data such as any type of logs, business events, and user activities can be ingested into Azure Data Explorer from various sources.
2. Ingest data into Azure Data Explorer with low-latency and high throughput using its connectors for [Azure Data Factory](#), [Azure Event Hubs](#), [Azure IoT Hub](#), [Kafka](#), and so on. Alternatively, ingest data through Azure Storage ([Blob](#) or [ADLS Gen2](#)), which uses [Azure Event Grid](#) and triggers the ingestion pipeline to Azure Data Explorer. You can also continuously export data to Azure Storage in compressed, partitioned parquet format and seamlessly query that data as detailed in the [Continuous data export overview](#).
3. Export pre-aggregated data from Azure Data Explorer to Azure Storage, and then ingest the data into Synapse Analytics to build data models and reports.
4. Use Azure Data Explorer's native capabilities to process, aggregate, and analyze data. To get insights at a lightning speed, build near real-time analytics dashboards using [Azure Data Explorer dashboards](#), [Power BI](#), [Grafana](#), or other tools. Use Azure Synapse Analytics to build a modern data warehouse and combine it with the Azure Data Explorer data to generate BI reports on curated and aggregated data models.
5. Azure Data Explorer provides native advanced analytics capabilities for [time series analysis](#), pattern recognition, [anomaly detection and forecasting](#), and [machine learning](#). Azure Data Explorer is also well integrated with ML services such as [Databricks](#) and [Azure Machine Learning](#). This integration allows you to build models using other tools and services and export ML models to Azure Data Explorer for scoring data.

Components

- [Azure Event Hubs](#) : Fully managed, real-time data ingestion service that's simple, trusted, and scalable.
- [Azure IoT Hub](#) : Managed service to enable bi-directional communication between IoT devices and Azure.
- [Kafka on HDInsight](#): Easy, cost-effective, enterprise-grade service for open source analytics with Apache Kafka.
- [Azure Data Explorer](#) : Fast, fully managed and highly scalable data analytics service for real-time analysis on large volumes of data streaming from applications, websites, IoT devices, and more.
- [Azure Data Explorer Dashboards](#): Natively export Kusto queries that were explored in the Web UI to optimized dashboards.
- [Azure Synapse Analytics](#) : Analytics service that brings together enterprise data warehousing and Big Data analytics.

Scenario details

Potential use cases

This solution illustrates how Azure Data Explorer and Azure Synapse Analytics complement each other for near real-time analytics and modern data warehousing use cases.

This solution is already being used by Microsoft customers. For example, the Singapore-based ride-hailing company, Grab, implemented real-time analytics over a huge amount of data collected from their taxi and food delivery services as well as merchant partner apps. The [team from Grab presented their solution at MS Ignite in this video \(20:30 onwards\)](#). Using this pattern, Grab processed more than a trillion events per day.

This solution is optimized for the retail industry.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Ornat Spodek](#) | Senior Content Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Azure Data Explorer documentation](#)
- [Training: Introduction to Azure Data Explorer](#)
- [Azure Synapse Analytics](#)
- [Azure Event Hubs](#)

Related resources

- [Analytics architecture design](#)
- [Analytics end-to-end with Azure Synapse](#)
- [Advanced analytics architecture](#)

Azure Data Explorer interactive analytics

Azure Data Explorer

Azure Data Factory

Azure Event Hubs

Azure IoT Hub

Azure Storage

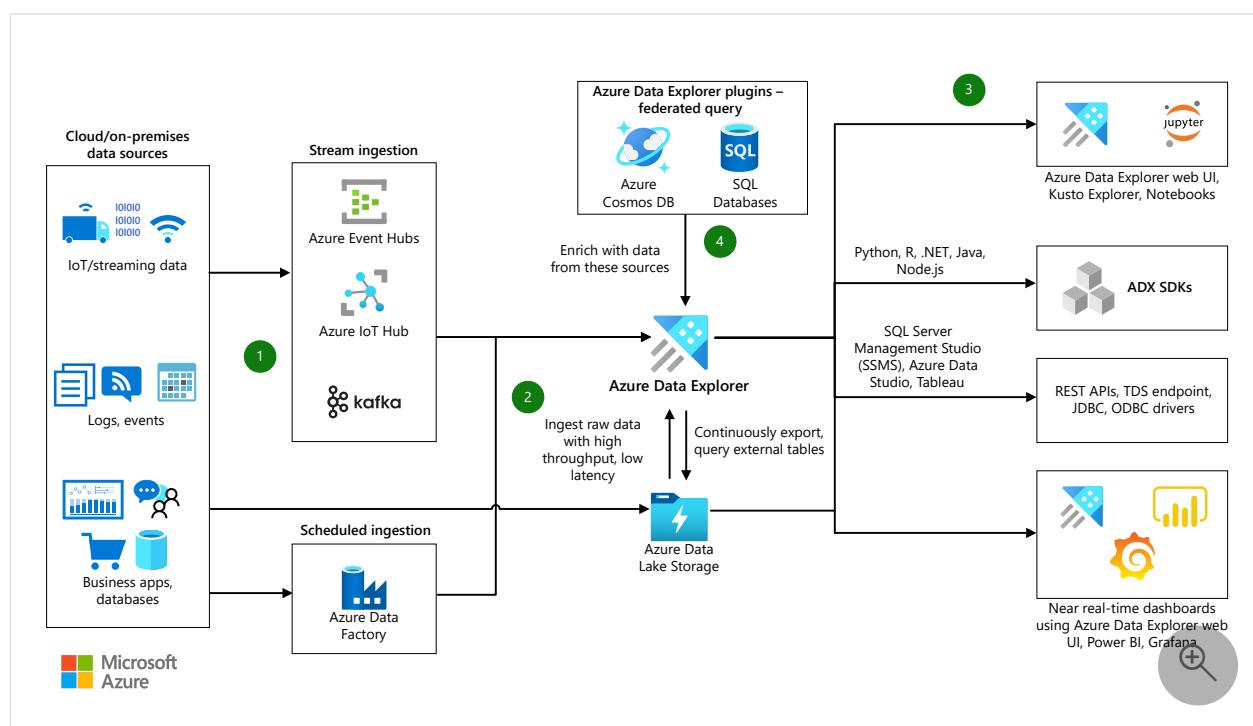
💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This solution idea demonstrates how to use interactive analytics in Azure Data Explorer. It describes how you can examine structured, semi-structured, and unstructured data with improvised, interactive, fast queries.

Jupyter is a trademark of its respective company. No endorsement is implied by the use of this mark. Apache® and Apache Kafka® 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

1. Raw structured, semi-structured, and unstructured (free text) data such as, any type of logs, business events, and user activities can be ingested into Azure Data Explorer from various sources. Ingest the data in streaming or batch mode using various methods.
2. Ingest data into Azure Data Explorer with low-latency and high-throughput using its connectors for [Azure Data Factory](#), [Azure Event Hubs](#), [Azure IoT Hub](#), [Kafka](#), and so on. Instead, ingest data through Azure Storage (Blob or ADLS Gen2), which uses [Azure Event Grid](#) and triggers the ingestion pipeline to Azure Data Explorer. You can also continuously export data to Azure Storage in compressed, partitioned parquet format and seamlessly query that data as detailed in [continuous data export overview](#).
3. Run interactive queries over small to extremely large volumes of data using native Azure Data Explorer tools or alternative tools of your choice. [Azure Data Explorer provides many plugins and integrations with the rest of the data platform ecosystem](#). Use any of the following tools and integrations:
 - For interactive analytics, use [Azure Data Explorer Web UI](#), web client for Azure Data Explorer, or [Kusto.Explorer](#), rich windows client for Azure Data Explorer.
 - To connect to your Azure Data Explorer cluster, use [Jupyter notebooks](#), [Spark connector](#), any [TDS-compliant SQL client](#), and JDBC and ODBC connections.
 - To build new apps or integrate with existing apps or frameworks, use Azure Data Explorer [REST APIs and SDKs available in different languages](#).
 - Build near real-time analytics dashboards using [Azure Data Explorer dashboards](#), [Power BI](#), or [Grafana](#).
4. Enrich data running federated queries by combining data from SQL database and Azure Cosmos DB using Azure Data Explorer plugins.

Components

- [Azure Event Hubs](#): Fully managed, real-time data ingestion service that's simple, trusted, and scalable.
- [Azure IoT Hub](#): Managed service to enable bi-directional communication between IoT devices and Azure.
- [Kafka on HDInsight](#): Easy, cost-effective, enterprise-grade service for open-source analytics with Apache Kafka.
- [Azure Data Factory](#): Hybrid data integration service that simplifies ETL at scale.

- [Azure Data Explorer](#) : Fast, fully managed and highly scalable data analytics service for real-time analysis on large volumes of data streaming from applications, websites, IoT devices, and more.
- [Azure Data Explorer Dashboards](#): Natively export Kusto queries that were explored in the Web UI to optimized dashboards.
- [Azure Cosmos DB](#) : Fully managed fast NoSQL database service for modern app development with open APIs for any scale.
- [Azure SQL DB](#) : Build apps that scale with the pace of your business with managed and intelligent SQL in the cloud.

Scenario details

This solution idea demonstrates how to use interactive analytics with Azure Data Explorer to explore data with improvised, interactive, and fast queries over small to extremely large volumes of data. This data exploration can be done using native Azure Data Explorer tools or alternative tools of your choice. This solution focuses on the integration of Azure Data Explorer with rest of the data platform ecosystem.

Potential use cases

This solution is used by Microsoft customers to track user activity, manage user profiles and user segmentation scenarios.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Ornat Spodek](#) | Senior Content Manager

Next steps

For more information, see [Azure Data Explorer documentation](#).

Related resources

- [Big data analytics with Azure Data Explorer](#)
- [IoT analytics with Azure Data Explorer](#)

- Azure Data Explorer monitoring

IoT analytics with Azure Data Explorer

Azure Cosmos DB

Azure Data Explorer

Azure Digital Twins

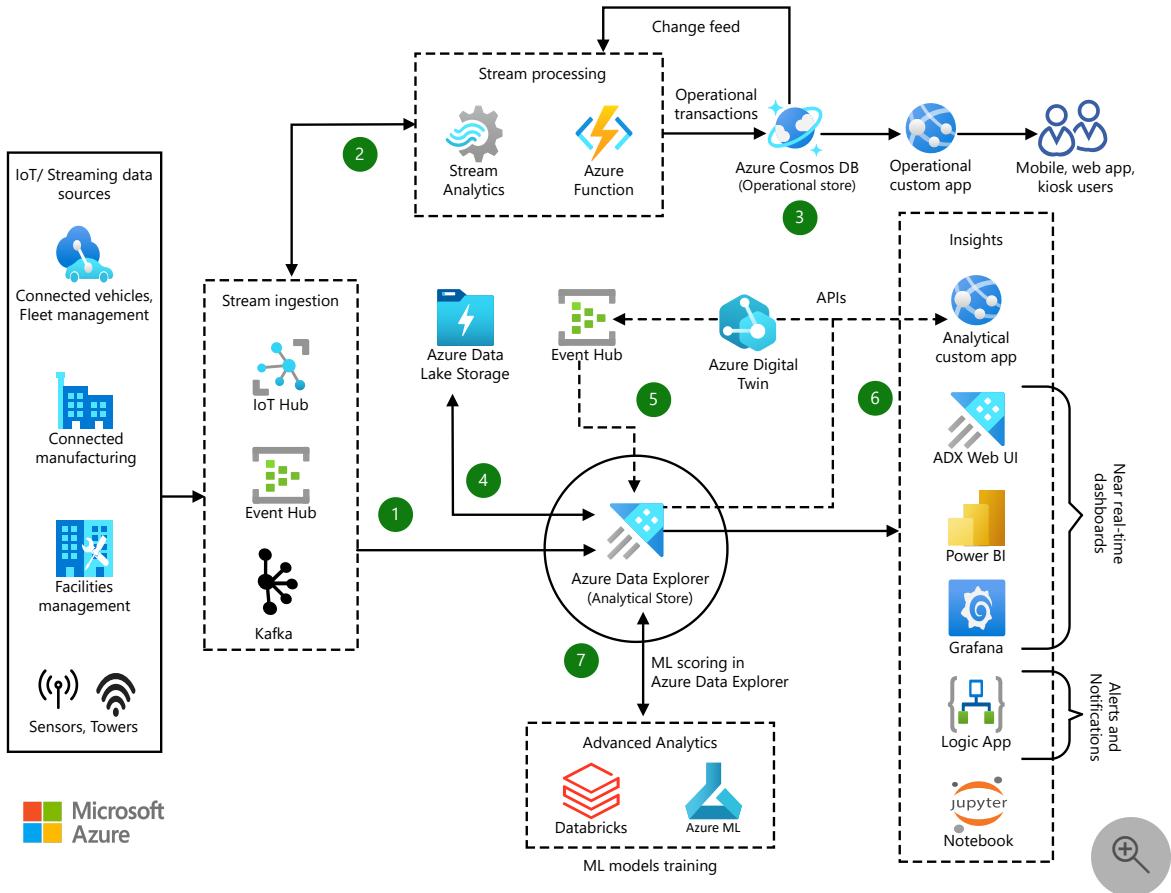
💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This solution idea describes how Azure Data Explorer provides near real-time analytics for fast flowing, high volume streaming data from internet of things (IoT) devices and sensors. This analytics workflow is part of an overall IoT solution that integrates operational and analytical workloads with Azure Cosmos DB and Azure Data Explorer.

Jupyter is a trademark of its respective company. No endorsement is implied by the use of this mark. Apache® and Apache Kafka® 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

1. Azure Event Hubs, Azure IoT Hub, or Kafka ingest a wide variety of fast-flowing streaming data such as logs, business events, and user activities.
2. Azure Functions or Azure Stream Analytics process the data in near real time.
3. Azure Cosmos DB stores streamed messages in JSON format to serve a real-time operational application.
4. Azure Data Explorer ingests data for analytics, using its connectors for [Azure Event Hubs](#), [Azure IoT Hub](#), or [Kafka](#) for low latency and high throughput.

Alternatively, you can ingest blobs from your [Azure Blob Storage](#) or [Azure Data Lake Storage](#) account into Azure Data Explorer by using an [Event Grid data connection](#).

You can also continuously export data to Azure Storage in compressed, partitioned [Apache Parquet](#) format, and seamlessly query the data with Azure Data Explorer. For details, see [Continuous data export overview](#).

5. To serve both the operational and analytical use cases, data can either route to Azure Data Explorer and Azure Cosmos DB in parallel, or from Azure Cosmos DB to Azure Data Explorer.

- Azure Cosmos DB transactions can trigger Azure Functions via change feed. Functions will stream data to Event Hubs for ingestion into Azure Data Explorer.

or

- Azure Functions can invoke Azure Digital Twins through its API, which then streams data to Event Hubs for ingestion into Azure Data Explorer.

6. The following interfaces get insights from data stored in Azure Data Explorer:

- Custom analytics apps that blend data from Azure Digital Twins and Azure Data Explorer APIs
- Near real-time analytics dashboards that use Azure Data Explorer dashboards, [Power BI](#), or [Grafana](#)
- Alerts and notifications from the [Azure Data Explorer connector for Azure Logic Apps](#)
- The Azure Data Explorer Web UI, [Kusto.Explorer](#), and [Jupyter notebooks](#)

7. Azure Data Explorer integrates with [Azure Databricks](#) and [Azure Machine Learning](#) to provide machine learning (ML) services. You can also build ML models using other tools and services, and export them to Azure Data Explorer for scoring data.

Components

This solution idea uses the following Azure components:

Azure Data Explorer

[Azure Data Explorer](#) is a fast, fully managed, and highly scalable big data analytics service. Azure Data Explorer can analyze large volumes of streaming data from applications, websites, and IoT devices in near real-time to serve analytics applications and dashboards.

Azure Data Explorer provides native advanced analytics for:

- [Time series analysis](#).
- Pattern recognition.

- [Anomaly detection and forecasting](#).
- [Machine learning \(ML\)](#).

The [Azure Data Explorer Web UI](#) connects to Azure Data Explorer clusters to help write, run, and share Kusto Query Language commands and queries. [Azure Data Explorer Dashboards](#) are a feature in the Data Explorer Web UI that natively exports Kusto queries to optimized dashboards.

Other Azure components

- [Azure Cosmos DB](#) is a fully managed, fast NoSQL database service for modern app development with open APIs for any scale.
- [Azure Digital Twins](#) stores digital models of physical environments, to help create next-generation IoT solutions that model the real world.
- [Azure Event Hubs](#) is a fully managed, real-time data ingestion service.
- [Azure IoT Hub](#) enables bi-directional communication between IoT devices and the Azure cloud.
- [Azure Synapse Link for Azure Cosmos DB](#) runs near real-time analytics over operational data in Azure Cosmos DB, without any performance or cost impact on transactional workloads. Synapse Link uses the [SQL Serverless](#) and [Spark Pools](#) analytics engines from the Azure Synapse workspace.
- [Kafka on HDInsight](#) is an easy, cost-effective, enterprise-grade service for open-source analytics with Apache Kafka.

Scenario details

This solution uses Azure Data Explorer to get near real-time IoT telemetry analytics on fast-flowing, high-volume streaming data from a wide variety of IoT devices.

Potential use cases

- Fleet management, for predictive maintenance of vehicle parts. This solution is ideal for the automotive and transportation industry.
- Facilities management, for energy and environment optimization.
- [Combining real-time road conditions with weather data for safer autonomous driving](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Ornat Spodek](#) | Senior Content Manager

Next steps

- [What is Azure Data Explorer?](#)
- [Visualize data with Azure Data Explorer dashboards](#)

Related resources

- [Azure Cosmos DB in IoT workloads](#)
- [Big data analytics with Azure Data Explorer](#)
- [IoT and data analytics](#)
- [Real-time analytics on big data architecture](#)

Augment security, observability, and analytics by using Microsoft Sentinel, Azure Monitor, and Azure Data Explorer

Azure Data Explorer

Azure Monitor

Microsoft Sentinel

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

Microsoft Sentinel, Azure Monitor, and Azure Data Explorer are based on a common technology and use Kusto Query Language (KQL) to analyze large volumes of data streamed in from multiple sources in near-real time.

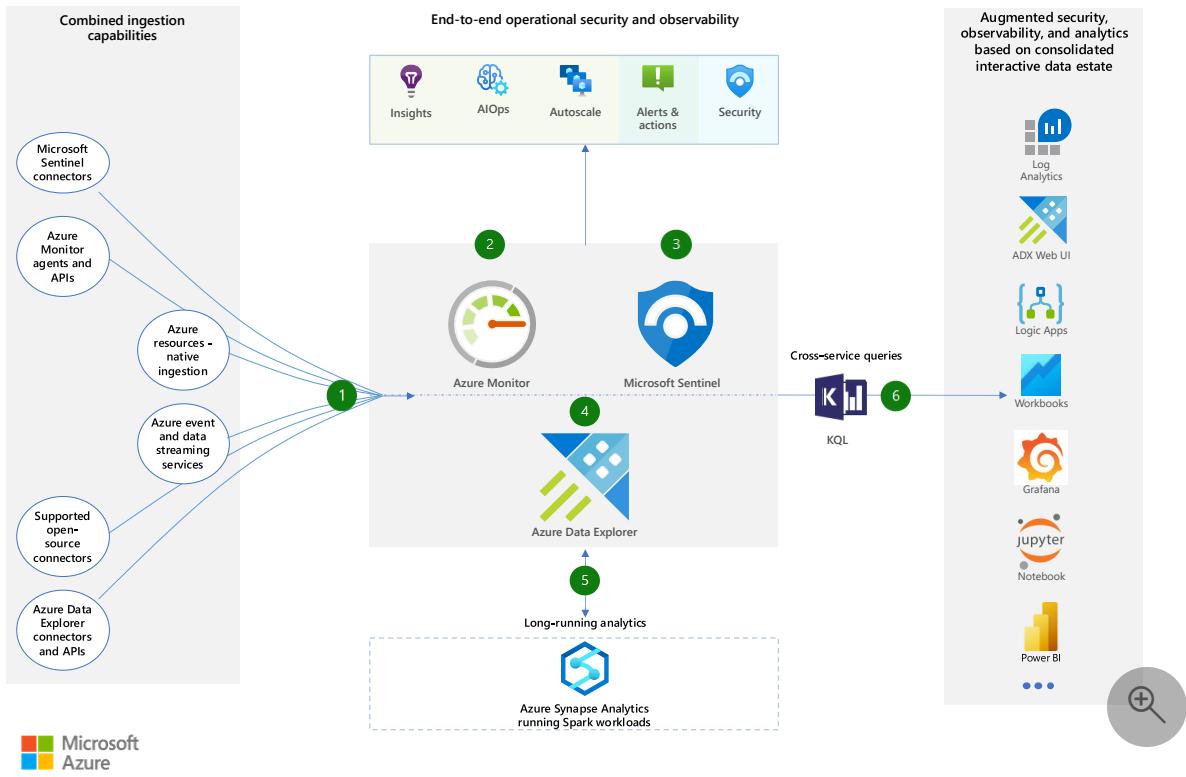
This solution demonstrates how to take advantage of the tight integration between Microsoft Sentinel, Azure Monitor, and Azure Data Explorer. You can use these services to consolidate a single interactive data estate and augment your monitoring and analytics capabilities.

ⓘ Note

This solution applies to Azure Data Explorer and also to [Real-Time Analytics KQL databases](#), which provide SaaS-grade real-time log, time-series, and advanced analytics capabilities as part of [Microsoft Fabric](#).

The Grafana and Jupyter logos and are trademarks of their respective companies. No endorsement is implied by the use of these marks.

Architecture



Download a [PowerPoint file](#) of this architecture.

Dataflow

1. Ingest data by using the combined ingestion capabilities of [Microsoft Sentinel](#), [Azure Monitor](#), and [Azure Data Explorer](#):
 - Configure diagnostic settings to ingest data from Azure services like Azure Kubernetes Service (AKS), Azure App Service, Azure SQL Database, and Azure Storage.
 - Use Azure Monitor Agent to ingest data from VMs, containers, and workloads.
 - Use a wide range of connectors, agents, and APIs supported by the three services to ingest data from on-premises resources and other clouds. Supported connectors, agents, and APIs include Logstash, Kafka, and Logstash connectors, OpenTelemetry agents, Azure Data Explorer APIs, and the Azure Monitor Log Ingestion API.
 - Stream data in by using Azure services like Azure IoT Hub, Azure Event Hubs, and Azure Stream Analytics.
2. Use Microsoft Sentinel to monitor, investigate, and alert and act on security-related data across your IT environment.
3. Use Azure Monitor to monitor, analyze, and alert and act on the performance, availability, and health of applications, services, and IT resources. Doing so enables

you to gain insights into the operational status of your cloud infrastructure, identify problems, and optimize performance.

4. Use Azure Data Explorer for any data that requires custom or more flexible handling or analytics, including full schema control, cache or retention control, deep data platform integrations, and machine learning.
5. Optionally, apply advanced machine learning on a broad set of data from your entire data estate to discover patterns, detect anomalies, get forecasts, and gain other insights.
6. Take advantage of the tight integration between services to augment monitoring and analytics capabilities:
 - Run cross-service queries from [Microsoft Sentinel](#), [Monitor](#), and [Azure Data Explorer](#) to analyze and correlate data in all three services in one query without moving the data.
 - Consolidate a single-pane-of-glass view of your data estate with customized cross-service workbooks, dashboards, and reports.

Components

Use cross-service queries to build a consolidated, interactive data estate, joining data in Microsoft Sentinel, Monitor, and Azure Data Explorer:

- [Microsoft Sentinel](#) is the Azure cloud-native solution for security information and event management (SIEM) and security orchestration, automation, and response (SOAR). Microsoft Sentinel has the following features:
 - Connectors and APIs for collecting security data from various sources, like Azure resources, Microsoft 365, and other cloud and on-premises solutions.
 - Advanced built-in analytics, machine learning, and threat intelligence capabilities for detecting and investigating threats.
 - Rules-based case management and incident response automation capabilities that use modular, reusable playbooks that are based on Azure Logic Apps.
 - KQL query capabilities that let you analyze security data and hunt for threats by correlating data from multiple sources and services.
- [Azure Monitor](#) is the Azure managed solution for IT and application monitoring. Monitor has the following features:
 - Native ingestion of monitoring data from Azure resources. Agents, connectors, and APIs for collecting monitoring data from Azure resources and any sources, applications, and workloads in Azure and hybrid environments.

- IT monitoring tools and analytics features, including AI for IT operations (AIOps) features, alerting and automated actions, and prebuilt workbooks for monitoring specific resources, like virtual machines, containers, and applications.
- End-to-end observability capabilities that help you improve IT and application efficiency and performance.
- KQL query capabilities that enable you to analyze data and troubleshoot operational issues by correlating data across resources and services.
- [Azure Data Explorer](#) is part of the Azure data platform. It provides real-time advanced analytics for any type of structured and unstructured data. It has the following features:
 - Connectors and APIs for various types of IT and non-IT data, for example, business, user, and geospatial data.
 - The full set of KQL's analytics capabilities, including hosting of machine learning algorithms in Python and federated queries to other data technologies, like SQL Server, data lakes, and Azure Cosmos DB.
 - Scalable data management capabilities, including full schema control, processing of incoming data by using KQL, materialized views, partitioning, granular retention, and caching controls.
 - Cross-service query capabilities that enable you to correlate collected data with data in Microsoft Sentinel, Monitor, and other services.

Scenario details

An architecture built on the features and flexibility provided by Microsoft Sentinel, Monitor, and Azure Data Explorer gives you:

- A broad range of data ingestion options that span various types of data and data sources.
- A powerful set of native security, observability, and data analytics features and capabilities.
- The ability to use cross-service queries to create a single-pane-of-glass view of your data by:
 - Querying IT monitoring and non-IT data.
 - Applying machine learning on a broad dataset to discover patterns, implement anomaly detection and forecasting, and get other advanced insights.
 - Creating workbooks and reports that enable you to monitor, correlate, and act on various types of data.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Guy Wild](#) | Senior Content Developer

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Azure Data Explorer documentation](#)
- [Training: Introduction to Azure Data Explorer](#)
- [Azure Monitor overview](#)
- [What is Microsoft Sentinel?](#)

Related resources

- [Big data analytics with Azure Data Explorer](#)
- [Azure Data Explorer interactive analytics](#)
- [IoT analytics with Azure Data Explorer](#)

Big data analytics with enterprise-grade security using Azure Synapse

Azure Analysis Services

Azure Data Lake Storage

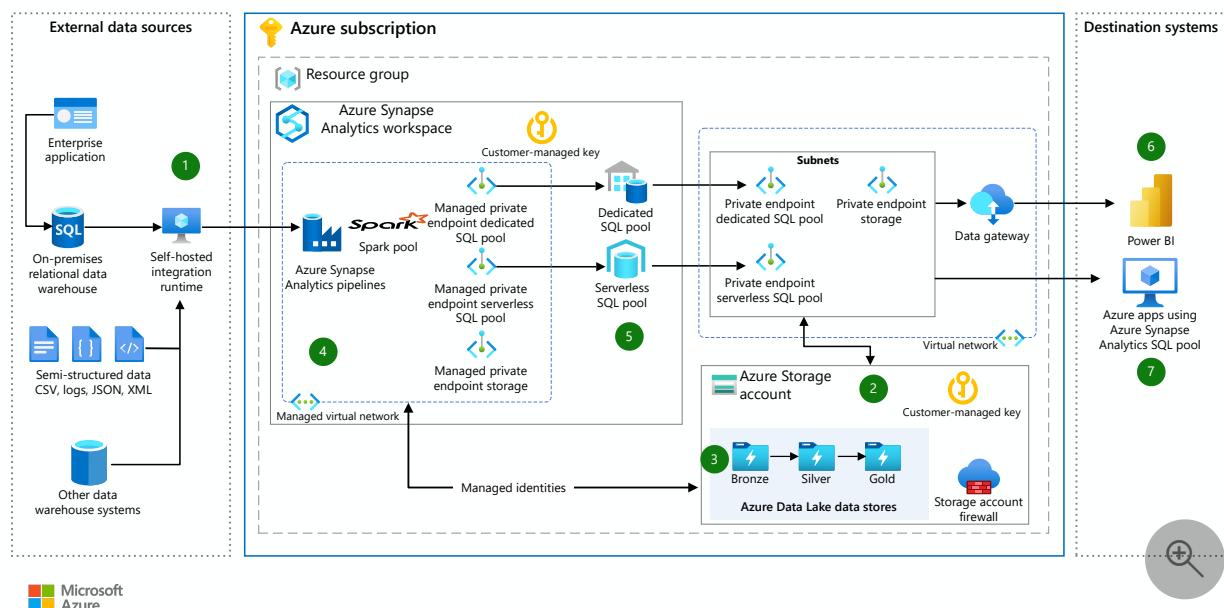
Azure Synapse Analytics

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

The solution described in this article demonstrates how to use Azure Synapse Analytics to build a modern data platform to ingest, process, store, serve, and visualize data from various sources.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

The data flows through the solution as follows:

1. The Synapse pipelines copy activities ingest raw structured data from external relational data warehouses, semistructured data such as logs, flat files, and xml, and other source systems. This ingested data is then stored in an Azure Data Lake Storage Gen2 location. Using a self-hosted integration runtime, you can also manage and run copy activities between a data store in your on-premises environment and the cloud.
2. Azure Data Lake Storage Gen2 provides secure storage.
 - Using a firewall to limit Storage Account access to trusted Azure services is recommended to limit external attack vulnerability.
 - [Private endpoints](#) for your Azure Storage accounts allow clients on the virtual network (VNet) to securely access data over a [Private Link](#). The private endpoint uses an IP address from the VNet address space for the storage account service. Network traffic between the clients on the VNet and the storage account traverses over the VNet and a private link on the Microsoft backbone network, eliminating exposure to the public internet.
3. Data is encrypted at rest once it's ingested into the data lake. Using your own customer-managed keys can further protect your encryption keys and add more flexibility when managing access controls.
4. Data is ingested using Synapse pipelines and processed in stages using the Synapse Spark pool and its Data Lake capabilities. Data is stored in the Azure Storage account using stage-specific Azure Data Lake Storage Gen 2 directories. These stages are:
 - a. The Synapse pipelines copy activities initially ingest data from the source systems. This ingested data is stored in its raw format using the data lake's *Bronze* directory.
 - b. The Synapse Spark pool then runs data quality rules to cleanse the raw data. This enriched data is then stored in the data lake's *Silver* directory.
 - c. After the cleansing process, the Spark pool applies any required normalization, data transformations, and business rules on the data in the Silver directory. This transformed data is then stored in the data lake's *Gold* directory.
5. The Synapse Apache Spark to Synapse SQL connector pushes the normalized data to the Synapse SQL pool for consumption by downstream applications and reporting services such as Power BI. This connector is designed to optimally transfer data between the serverless Apache Spark pools and the SQL pools in the Azure Synapse Analytics workspace.

6. The Power BI service uses DirectQuery mode to securely fetch data from the Synapse SQL pool. A data gateway installed in a virtual machine on the private VNet acts as a connecting platform between the Power BI service and the Synapse SQL pool, using Private Endpoint in the same VNet to securely connect.
7. External applications can access data from the Synapse serverless pools or dedicated SQL pools by accessing the appropriate private endpoints connected to the VNet.

This example solution makes use of several Azure services and features:

- [Azure Synapse Analytics](#) is the core service used in this example solution to provide data ingestion, processing, and analytics.
- [Azure Data Lake Storage \(Gen2\)](#) is built on top of [Azure Storage](#) services and provides data lake capabilities that other services in this example solution use when storing and processing data.
- [Synapse pipelines](#) copies data from original sources into the data lake storage locations.
- [Apache Spark in Azure Synapse Analytics](#) cleanses, normalizes, and performs other processing tasks on data ingested from source locations.
- [Dedicated SQL pool](#) (formerly SQL DW) provides data warehousing capabilities for data after it's been processed and normalized and is ready for use by your end users and applications.
- [Serverless SQL pool](#) allows users to quickly query and analyze processed and normalized data.
- [Azure Synapse Managed Virtual Network](#) creates an isolated managed virtual networking environment for the Azure Synapse workspace, offloading the need for you to manage networking configuration for the workspace resources.
- [Azure Synapse Managed private endpoints](#) establish private links to Azure resources and route traffic between your Azure Synapse workspaces and other Azure resources using only the Microsoft backbone network.
- [Azure Virtual Network \(VNet\)](#) provides private networking capabilities for Azure resources that aren't a part of the Azure Synapse workspace. It allows you to manage access, security, and routing between resources.
- [Azure Private Endpoint](#) provides a private IP address from the solution's VNet to Azure managed services, effectively connecting a service to the VNet. This allows

secure networking between the Azure Synapse workspace and other Azure services such as Azure Storage, Azure Cosmos DB, Azure SQL Database, or your own [Azure Private Link service](#).

- [Power BI](#) allows users to perform advanced analysis and share insights using the solution's processed data.

Components

- [Azure Synapse Analytics](#)
- [Azure Data Lake Storage](#)
- [Azure SQL Database](#)
- [Virtual Network](#)
- [Azure Private Link](#)
- [Power BI](#)

Scenario details

Azure Synapse Analytics brings together data integration, enterprise data warehousing, and big data analytics to help you build a modern data platform capable of handling the most common data challenges facing large organizations. Azure Virtual Network allows you to create your own private network in the Azure public cloud and managed network, and Azure Private Endpoint allows you to securely integrate managed cloud services into these private networks.

Potential use cases

The solution described in this article demonstrates how to combine these technologies to build a modern data platform that can ingest, process, store, serve, and visualize data from different sources, both structured and semistructured, while meeting the high security standards your organization expects. This includes supporting common requirements, such as:

- **Securing data sources.** Data sources inside the on-premises corporate network or on the virtual network are secured behind a firewall. These resources can be securely accessed by installing a self-hosted integration runtime on a resource hosted on-premises or on the virtual networks.

- **Authentication and authorization using managed identities.** Communication between Azure services can be secured using managed identities, which provide an identity for applications to use when connecting to resources that support Microsoft Entra authentication. In this example, Azure Synapse uses the managed identity to integrate pipelines.
- **Private endpoints establishing a private link to Azure resources.** Azure Synapse provides fully managed private endpoint functionality for services within the Synapse workspace (such as Azure Storage or Azure Cosmos DB). Other Azure resources such as Azure applications, Microsoft Power BI, and Azure Synapse service are secured using Private Endpoints integrated into the example solution's virtual network. Network traffic between your private network and the Synapse pools uses Private Link to move traffic over the Microsoft backbone network, eliminating exposure to the public internet.
- **Encrypting data in transit.** Data is encrypted in transit as all data transfers are via secure channel HTTPS and TLS over TCP to prevent man-in-the-middle attacks during communication with Azure services, ensuring end-to-end secure private data movement.
- **Encrypting data at rest.** Transparent data encryption in Azure Synapse Analytics helps protect against malicious activity by performing real-time encryption and decryption of your data stored within the Synapse workspace. Azure Storage also encrypts all data in a storage account at rest. By default, data is encrypted with Microsoft-managed keys, but you can manage your own keys if you need additional control over encryption.

Deploy this scenario

You must have an existing Azure account. If you don't have an Azure subscription, create a [free account](#) before you begin.

The Azure Resource Manager templates, which you'll need to deploy the components described in this architecture, are available in the [GitHub](#) repository. These templates will deploy all the services shown in the architecture diagram **except for:** the Power BI Data Gateway, self-hosted integration runtime, and Azure Key Vault for customer managed keys.

It's up to the user to create the data lake folder structure and the Azure Synapse Analytics integration pipelines that are necessary to connect to the data sources.

Deploy the ARM template directly by clicking this button:



Deploy to Azure



Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Kiran Kalyanam](#) | Senior Software Engineer

Next steps

To learn how to further develop this approach, learn the basics of Azure Synapse Analytics by completing the following tutorials:

- [Get Started with Azure Synapse Analytics](#)
- [Tutorial: Explore and Analyze data lakes with serverless SQL pool](#)
- [Analyze data in a storage account](#)
- [Analyze data with dedicated SQL pools](#)
- [Integrate with pipelines](#)

Related resources

Refer to these articles when planning and deploying solutions using Azure Synapse Analytics:

- [Data exfiltration protection for Azure Synapse Analytics workspaces](#)
- [Azure Synapse Analytics IP firewall rules](#)
- [Azure Synapse Analytics Managed Virtual Network](#)
- [Synapse Managed private endpoints](#)
- [Configure Azure Storage firewalls and virtual networks](#)
- [Connect to Azure Synapse Studio using Azure Private Link Hubs](#)
- [Connect to a secure Azure storage account from your Synapse workspace](#)

- Use Microsoft Entra authentication for authentication with Synapse SQL

Related architecture guidance

- [Analytics end-to-end with Azure Synapse](#)
- [Choosing an analytical data store in Azure](#)
- [Automated enterprise BI](#)
- [Enterprise business intelligence](#)
- [Advanced Analytics Architecture](#)
- [Real-time analytics on big data architecture](#)

Content Delivery Network analytics

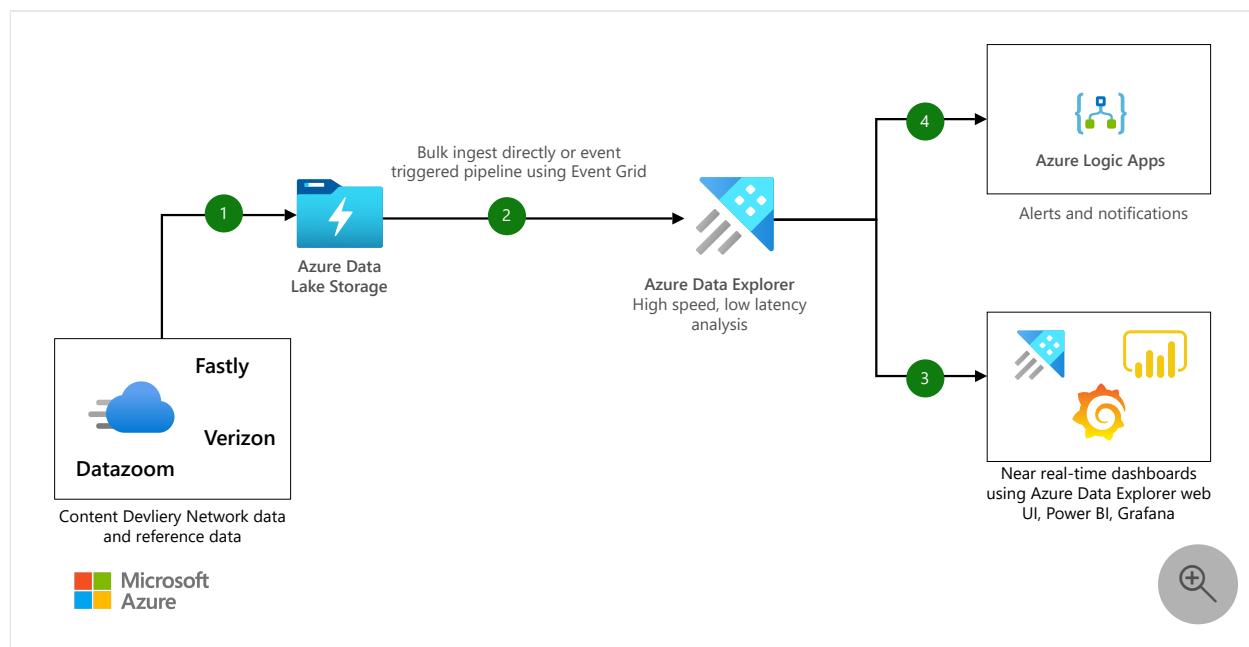
Azure Data Explorer Azure Logic Apps Azure Storage

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This solution idea demonstrates low-latency, high-throughput ingestion of large volumes of Content Delivery Network (CDN) logs. You can use this data to create near real-time analytics dashboards.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

1. Content Delivery Network providers such as Verizon and Fastly ingest huge amounts of CDN logs into Azure Data Explorer to analyze latencies, health, and performance of CDN assets.

2. Most CDN scenarios ingest data through Azure Storage ([Blob](#) or [ADLS Gen2](#)), which uses [Azure Event Grid](#) and triggers the ingestion pipeline to Azure Data Explorer. Alternatively you can bulk ingest the data using the [LightIngest tool](#). You can also continuously export data to Azure Storage in compressed, partitioned parquet format and seamlessly query that data as detailed in [Continuous data export overview](#).
3. Azure Data Explorer provides easy to use native operators and functions to process, aggregate, and analyze time series and log data, as well as supply insights at lightning speed. You can build near real-time analytics dashboards using [Azure Data Explorer dashboards](#), [Power BI](#), or [Grafana](#).
4. Create and schedule alerts and notifications using [Azure Data Explorer connector for Azure Logic Apps](#).

Components

- [Azure Storage Azure Data Explorer connector](#): Continuous ingestion from Azure Storage (Blob storage and ADLSv2) with Azure Event Grid subscription to stream these notifications to Azure Data Explorer.
- [Azure Data Explorer](#): Fast, fully managed and highly scalable data analytics service for real-time analysis on large volumes of data streaming from applications, websites, IoT devices, and more.
- [Azure Data Explorer Dashboards](#): Natively export Kusto queries that were explored in the Web UI to optimized dashboards.
- [Azure Logic Apps Azure Data Explorer connector](#): Run Kusto queries and commands automatically as part of a scheduled or triggered task.

Scenario details

You can use this solution to ingest large volumes of CDN logs for the purpose of creating near real-time analytics dashboards.

Potential use cases

- Log analytics
- Time series analytics
- IoT
- General-purpose exploratory analytics

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Ornat Spodek](#) | Senior Content Manager

Next steps

For more information, see [Azure Data Explorer documentation](#).

Related resources

- [Analytics architecture design](#)
- [Choose an analytical data store in Azure](#)
- [Big data analytics with Azure Data Explorer](#)
- [Azure Data Explorer monitoring](#)

Data management across Azure Data Lake with Microsoft Purview

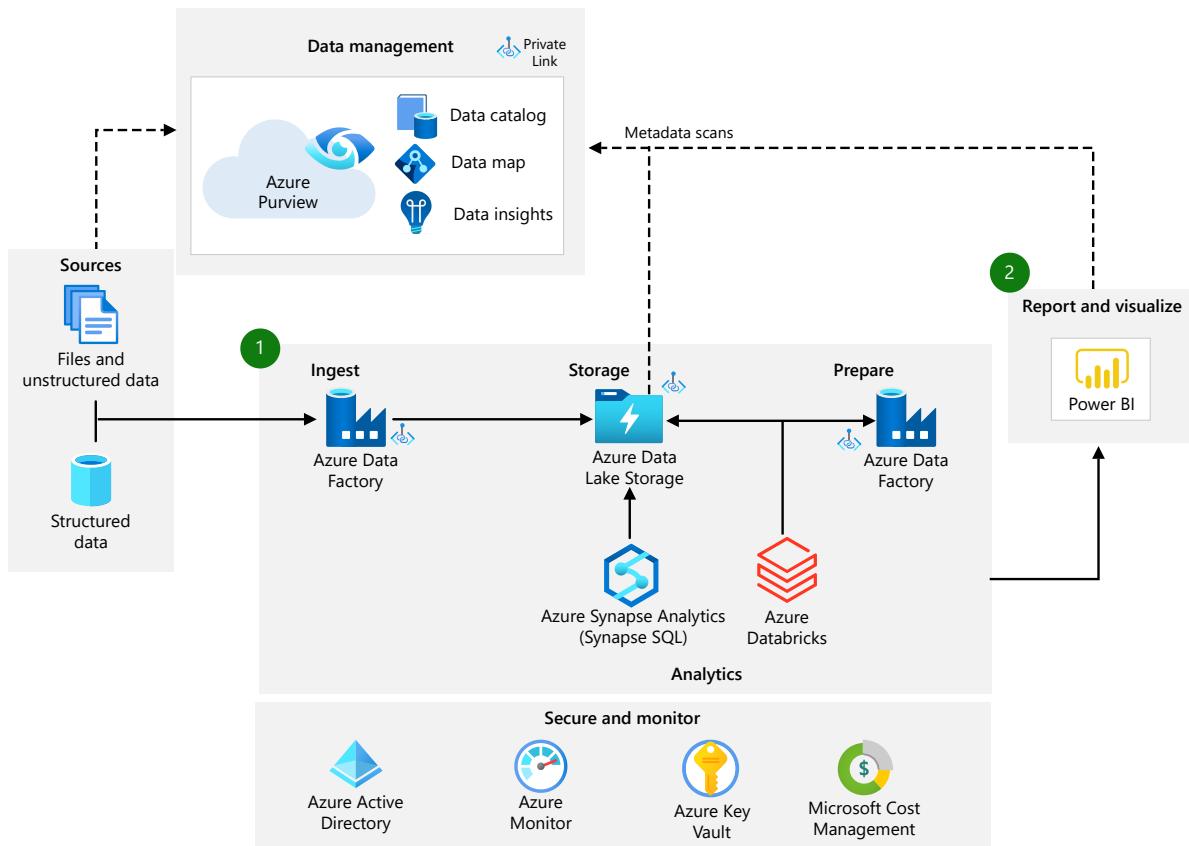
Azure Data Factory Microsoft Purview Azure Data Lake Storage Azure Synapse Analytics Power BI

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This article describes a solution that uses Azure Purview to build a foundation for data governance and management that can produce and deliver high-quality, trusted data.

Architecture



 Microsoft Azure

Download a [Visio file](#) of this architecture.

Dataflow

Azure Purview provides a single, unified data management service for the data from all sources, in the data lake, and in end reporting tools.

Scenarios for connecting Azure Purview to Data Lake services:

1. Azure Purview provides an improved-security connection to your data lake ingestion, storage, and analytics pipelines to automatically catalog data assets. It also provides lineage across these services. Specific Azure services include Data Factory, Data Lake Storage, and Azure Synapse Analytics.
2. Azure Purview connects natively with Power BI and other reporting and visualization tools. It shows the lineage of data that's used in end reports. It also shares sensitivity information from the Power BI assets to prevent incorrect data use.

Important

The information that's transferred from the sources to Azure Purview is metadata that describes the data within the scanned sources. No actual data is transferred from the sources to Azure Purview.

Capabilities

- **Catalog.** The Azure Purview Data Catalog can automatically capture and describe core characteristics of data at the source, including schema, technical properties, and location. The Azure Purview glossary allows a business-friendly definition of data to be layered on top, to improve search and discovery.
- **Classification.** Azure Purview automatically classifies datasets and data elements with 100 predefined sensitive-data classifications. It also allows you to define your own custom classification schemes that you can apply manually and automatically.
- **Lineage.** Azure Purview diagrammatically visualizes lineage across Data Factory, Azure Synapse Analytics, and Power BI pipelines. These visualizations show the end-to-end flow of data at a granular level.
- **Access control.** Azure Purview access control policy allows you to define and grant access to data assets from the catalog, directly on the underlying sources.
- **Ownership.** Azure Purview allows you to apply data ownership and stewardship to data assets and glossary items in the catalog.

- [Insight](#). Insights in Azure Purview provide multiple predefined reports to help CDOs, data professionals, and data governance professionals gain a detailed understanding of the data landscape.

Components

- [Azure Purview](#) is a unified data catalog that manages on-premises, multicloud, and software as a service (SaaS) data. This data governance service maintains data landscape maps. Features include automated data discovery, sensitive data classification, and data lineage.
- [Data Factory](#) is a fully managed, serverless data integration service that helps you construct ETL and ELT processes.
- [Data Lake Storage](#) provides massively scalable, high-security, cost-effective cloud storage for high-performance analytics workloads.
- [Azure Synapse Analytics](#) is a limitless analytics service that brings together data integration, enterprise data warehousing, and big data analytics.
- [Power BI](#) is a collection of software services and apps. These services create and share reports that connect and visualize multiple sources of data. When you use Power BI with Azure Purview, it can catalog and classify your data and provide granular lineage that's illustrated from end to end.
- [Azure Private Link](#) provides private connectivity from a virtual network to Azure platform as a service (PaaS) services, services that you own, or Microsoft partner services.
- [Azure Key Vault](#) stores and controls access to secrets like tokens, passwords, and API keys. Key Vault also creates and controls encryption keys and manages security certificates.
- [Microsoft Entra ID](#) offers cloud-based identity and access management services. These features provide a way for users to sign in and access resources.
- [Azure Monitor](#) collects and analyzes data on environments and Azure resources. This data includes app telemetry, like performance metrics and activity logs.

Scenario details

As you load more data into Azure, the need to properly govern and manage that data across all your data sources and data consumers also grows.

If you don't have high-quality data in your Azure data estate, the business value of Azure is diminished. The solution is to build a foundation for data governance and management that can produce and deliver high-quality, trusted data.

Data needs to be managed at scale across on-premises, cloud, and multicloud storage to ensure it meets compliance requirements for security, privacy, and usage. Well-managed data can also improve self-discovery, data sharing, and data quality, which improves the use of data in applications and analytics.

[Azure Purview](#) provides governance for finding, classifying, defining, and enforcing policies and standards across data. You can use it to apply definitions, classifications, and governance processes uniformly across data. It catalogs all data sources, identifies any sensitive information, and defines data lineage. It provides a central platform where you can apply definitions and ownership to data. With a single view on reports and insight, it can help you generate data standards that should be applied to your data.

Working with other Azure services, Azure Purview can automatically discover, catalog, classify, and manage data across Azure Data Lake offerings and partner services.

Potential use cases

The requirements for data management differ across industries. For all industries, the need to govern data at scale has increased as the size and complexity of data and data architectures grow. This is appropriate for organizations that would benefit from the following outcomes of well-governed data:

- Automatic discovery of data to accelerate cloud adoption.
- Improved security of data for compliance with data laws and regulations.
- Improved access, discovery, and quality of managed data to enhance analytics.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Isabel Arevalo](#) ↗ | Senior Cloud Solution Architect

Next steps

- [Azure Purview customer case studies](#) ↗

- Microsoft Purview technical documentation and best practices
- What is Microsoft Purview?
- What is Power BI? ↗
- What is Microsoft Entra ID?
- What is Data Factory?
- Introduction to Data Lake Storage
- What is Azure Databricks?
- Azure Monitor overview
- What is Azure Synapse Analytics?
- What is Azure Key Vault?

Related resources

- Data analysis workloads for regulated industries
- Query a data lake or lakehouse by using Azure Synapse serverless
- Choose an analytical data store in Azure
- Choose a data analytics technology in Azure

Ingest metadata from external catalogs to Microsoft Purview

Microsoft Purview Azure Functions Azure Event Hubs Azure Table Storage Azure Monitor

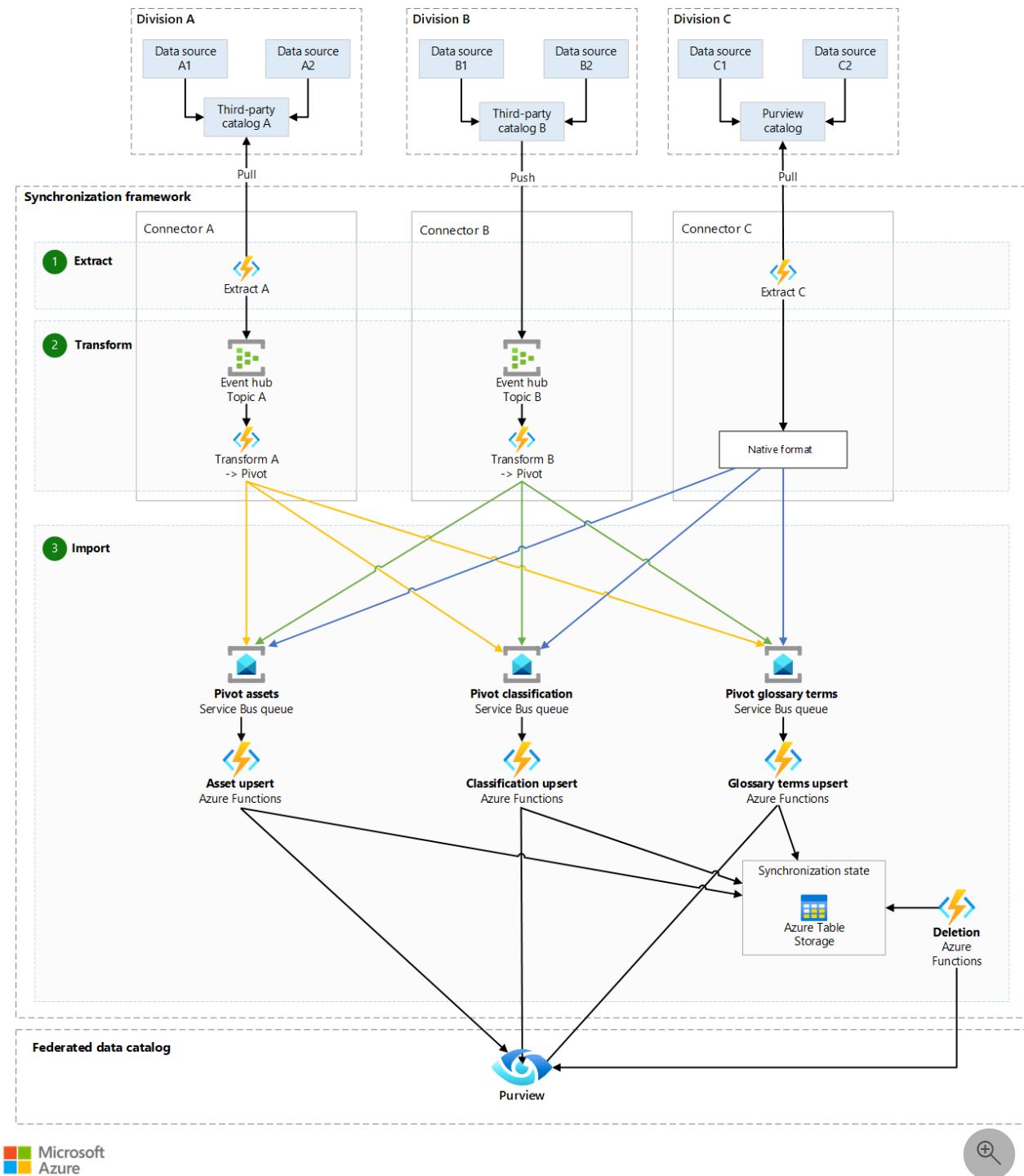
Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This article describes a highly scalable architecture for ingesting metadata from external catalogs, like [Acryl Data](#) and [data.world](#), into [Microsoft Purview](#). The company in this scenario has various subdivisions and subsidiaries that work independently.

Apache® 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.

Workflow

This solution imports three types of objects from external catalogs into Microsoft Purview: [assets](#), [glossary terms](#), and [classifications](#).

The architecture has two main components:

- Connectors (steps 1 and 2)
- An import module (step 3)

The **connectors** are specific to each external catalog. They're responsible for the first two steps: extract and transform. Because of the nature of these steps, and because the connectors depend on details that are specific to the external catalog, you need to create a connector for each catalog that the solution works with. Catalog-specific

details include how to extract the metadata, by using, for example, APIs, and how the metadata is structured. You therefore need one of each of the services components of the connectors that are shown in the diagram (Azure Functions, Azure Event Hubs) per external catalog. The output of the connector component is written in a catalog-agnostic format, referred in this article to as the *pivot* format.

The **import module** is the last step in the synchronization framework. It's also catalog-agnostic, and it works with metadata in pivot format. All metadata that was extracted and transformed into pivot format is streamed to three service buses, one for each object type: classifications, glossary terms, and assets. An Azure function listens to each service bus queue and imports the received object into Microsoft Purview as an upsert operation by using the [Microsoft Purview API](#) or [SDKs](#). The functions also populate intermediate storage, referred to as *synchronization state* in this article, with information about the imported objects. Deletion is handled on a scheduled basis, via an Azure function, based on information from synchronization state.

1. Extract

The extract step is accomplished in one of two ways: pull-based or push-based.

- **Pull-based** extraction: This approach uses an Azure function with an event hub as output binding. The Azure function pulls all the metadata from the external catalog. The function is triggered on a [scheduled basis](#).
 - The Azure function can split the extracted metadata into separate messages, as appropriate. For example, if the external catalog contains metadata about six Azure SQL tables, you can split the information into six messages, one for each table. Use the structure of the metadata from the external catalog to determine whether splitting the messages makes sense.
 - Each message is then sent to the Event Hubs topic that corresponds to that external catalog via [Event Hubs output binding](#).
- **Push-based** extraction: The subdivision owns the extraction step. The subdivision implements a mechanism that uses Kafka to directly push the metadata to an Event Hubs topic.

ⓘ Note

One Event Hubs topic is used for each external catalog.

2. Transform

During the transform step, the metadata that's received from the external catalog is converted into a common *pivot* format. This step consists of one Azure function, with an event hub as an input binding, for each external catalog. The output of this step is a set of assets, glossary terms, or classifications in the pivot format. This set is streamed to the corresponding Azure Service Bus queue via a [Service Bus output binding](#).

For example, if the message that's received in the event hub contains a glossary term, the term is transformed to its pivot format and sent to the pivot glossary terms queue.

3. Import

The last step is catalog agnostic and is responsible for loading the objects into Microsoft Purview in the intermediate pivot format. It consists of three Azure functions, each listening to its corresponding Service Bus queue. Every function implements an upsert operation (create/update) for one of the three object types. Using an upsert avoids creating duplicates. The Microsoft Purview REST API or SDKs, like the [Microsoft Purview Python SDK](#), are used for the import.

[Azure Table Storage](#) is used for the synchronization state intermediate storage. During the import, the synchronization state is updated with information about the imported object.

This synchronization framework operates based on the assumption that all metadata from the external catalog is imported during each run. Object creation and updates are handled by the upsert operation. An Azure function uses synchronization state to handle deletion. Because the synchronization framework is triggered on a schedule, a cleanup mechanism can run that identifies objects in synchronization state that weren't updated in a given number of days, even when multiple synchronizations ran during that time. In such a case, the metadata doesn't exist in the external catalog and should be removed from Microsoft Purview as well.

Components

- [Azure Functions](#) is used in all compute steps. You can easily trigger Azure functions based on a schedule or specific events, which is particularly useful in a synchronization framework.
- [Event Hubs](#) is used between the extract and transform steps. It streams the metadata that's extracted from the external catalog and consumed during the transformation step. Event Hubs provides [an interface that can consume events produced by Kafka](#).
 - An external catalog can communicate with the system by using its own protocol and language, without needing details about the internals of the system, which uses exclusively the Event Hubs protocol.
 - Event Hubs is suitable for both pull-based and push-based scenarios.
- [Service Bus](#) is used as the message broker between the transform and import steps. Separate queues are used for assets, glossary terms, and classification. Service Bus provides high scalability and built-in mechanisms that are useful in an event-driven distributed system, like Peek-Lock, a dead letter queue, and message deferral.
- [Microsoft Purview](#) is a data governance solution that provides a holistic map of metadata assets.
- [Table Storage](#) stores the state of the objects that are synchronized between the external sources and Microsoft Purview. It's cost-effective and provides highly scalable data storage, a simple API, and a strong consistency model.
- [Managed identities](#) are used for all Azure functions when they communicate with the corresponding event hubs and service bus queues. For information about using the Microsoft Purview Rest API with a service principal, see [How to use REST APIs for Microsoft Purview Data Planes](#). For more information about access control in Microsoft Purview, see [Understand access and permissions in the Microsoft Purview governance portal](#).
- [Application Insights](#), a feature of [Azure Monitor](#), is used to collect telemetry. Monitoring is important in distributed systems. For more information, see [Distributed tracing in Application Insights](#).

Scenario details

This entire process is fully idempotent, which means that retriggering the process from any step is enough to make the system eventually consistent. For more information about using Event Hubs with Azure Functions, see [Integrate Event Hubs with serverless functions on Azure](#).

This architecture is also highly scalable:

- In addition to offering plans for scalability, Event Hubs provides an [Auto-inflate feature](#) that increases the number of throughput units as needed. Similarly, Service Bus has an automatic scaling feature that adapts the number of messaging units.
- The architecture uses Event Hubs and Service Bus triggers, which enables Azure Functions to scale in or out, balance the load, and process incoming messages concurrently as needed.

Note

To avoid lock contention and to achieve the highest performance, you need to perform throughput testing.

- The Azure functions use Event Hubs triggers, which are compatible with the use of a [consumption plan](#) that bills based on per-second resource consumption and executions. Using this plan makes the architecture scalable and cost-efficient.

Potential use cases

Organizations that take advantage of the potential of data can gain significant benefits. For example, Contoso, like many large companies, wants to create a holistic view of its data assets to enable data-driven business scenarios. Contoso is made up of multiple subdivisions and subsidiaries that work independently, which results in data silos and limited collaboration.

For example, employees in Division A are starting a new project. Data assets like website logs, trend analysis, and social network analysis might be relevant to the project. However, these assets belong to other subsidiaries, and the employees aren't even aware of them, which affects the success of the project.

Contoso wants to avoid this type of situation by creating a federated metadata catalog. Examples of metadata include the name and schema of a SQL table. The metadata doesn't reveal the contents of the table.

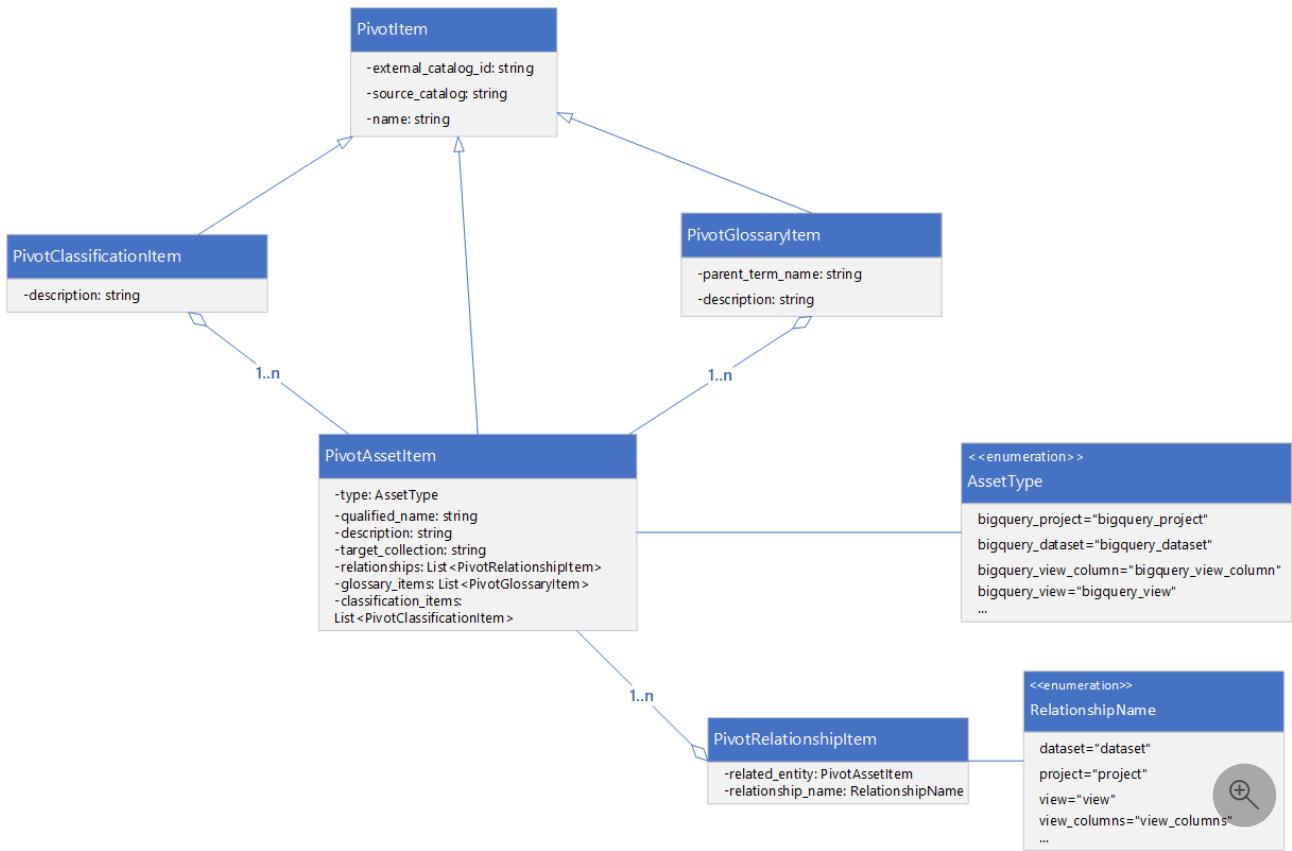
Contoso decides to use Microsoft Purview to solve this problem. Microsoft Purview enables the search and discovery of metadata about data assets. A federated catalog improves collaboration and breaks down organizational boundaries. Subdivisions and subsidiaries still own their data. However, because they share metadata about the data, collaboration improves. On a case-by-case basis, data can also be shared. Some subdivisions have already invested time and effort into implementing their own catalogs and scanning and enriching metadata, sometimes by using custom solutions. The next challenge is to determine how to import metadata from other catalogs into Microsoft Purview.

To resolve this challenge, Contoso uses the architecture described in this article to ingest metadata from external catalogs into Microsoft Purview.

Pivot classes

In this article, the word *pivot* is used to describe a set of custom classes that represents the output of the transform step and the input for the import step.

Here's the class diagram:



`PivotItem` is the base class. `PivotClassificationItem`, `PivotAssetItem`, and `PivotGlossaryItem` inherit from the base class.

By using the information that's passed in the pivot classes, the import step [can create or update an entity](#), a [glossary term](#), or a [classification](#) via, for example, the REST API.

You can use `PivotAssetItem` to create relationships between assets. You can also assign [glossary terms](#) or [classifications](#) to assets by using the `glossary_items` and `classification_items` properties.

The Microsoft Purview Data Catalog is based on the [Apache Atlas](#) format, so each metadata object that's supported in Microsoft Purview has a type.

The type is analogous to a class definition in object-oriented programming (OOP). All metadata objects managed by Microsoft Purview (out of the box or through custom types) are modeled via type definitions. For more information about the type system, see [Type definitions and how to create custom types in Microsoft Purview](#).

ⓘ Note

For even better scalability, you can [ingest metadata by using the Atlas hook](#). Microsoft Purview can have an optional underlying event hub on which Kafka Surface is enabled. However, the import step only sends Kafka messages to a topic. You have no visibility into whether the sent data is actually ingested later in the process. You can enable that visibility by adding logic, but doing so adds complexity. To provide a better user experience, this architecture uses REST APIs that provide return statuses and relies on the Azure Functions scale-out capability.

Additional details about properties

The `type` property is the type of the asset that you want to create. For example, if you want to create an asset of type SQL table, the `type` is `azure_sql_table`. For an overview of the types in Microsoft Purview, see [Types - REST API](#).

- `target_collection` is the name of the collection in which the asset should be located in Microsoft Purview.
- `external_catalog_id` is the unique identifier of the object in the external catalog.

All of these properties are set in the connector so that they can be used during the import step.

Synchronization state

The synchronization state intermediate storage maintains the state of synchronization by storing the mapping between the source metadata and Microsoft Purview metadata objects, as shown here:

[Expand table](#)

Partition Key	Row Key	Sync Start Time	Correlation ID	State	Purview Object ID	Sync End Time
<code>CatalogName_Asset</code>	123..a1c	2023-10-24T21:05:32Z	456..def	Pending		
<code>CatalogName_Glossary</code>	7c3...bdf	2023-10-25T22:12:27Z	er3..2d4	Completed	5f9...sds	2023-10-25T22:13:02Z
<code>CatalogName_Classification</code>	de3...85f	2023-11-25T22:12:27Z	ce4...13c	Failed		2023-11-25T22:13:02Z

 **Note**

Synchronization state is used to reflect the last run of the synchronization framework. It doesn't provide historical data of all runs. Therefore, only one entry per object is imported from external catalogs. The entry reflects the state of the last synchronization (`Completed`, `Pending`, or `Failed`).

Here are some details about these properties:

- **Partition Key**. A concatenation of the name of the external catalog and the type of the object (`Asset`, `Glossary`, or `Classification`). For example:
 - `CatalogA_Asset`. Assets from a catalog called Catalog A.
 - `CatalogB_Glossary`. Glossary terms from a catalog called Catalog B.
 - `CatalogC_Classification`. Classifications from a catalog called Catalog C.
- **Row Key**. The unique identifier of the metadata object in the external catalog, such as the GUID. It's used as the row key in every partition.

 **Note**

Table Storage is used for synchronization state because it provides strong consistency, which ensures that there's only one unique record per object for a given partition.

- **Correlation ID**. The unique identifier of the synchronization operation. It's created at the start of the workflow and passed from one step to another.

- `Sync Start Time` and `Sync End Time`. The start and end time of the synchronization operation (UTC).
- `State`. The state of the synchronization operation (`Pending`, `Completed`, or `Failed`).
- `Purview Object ID`. The unique identifier of the object in Microsoft Purview.

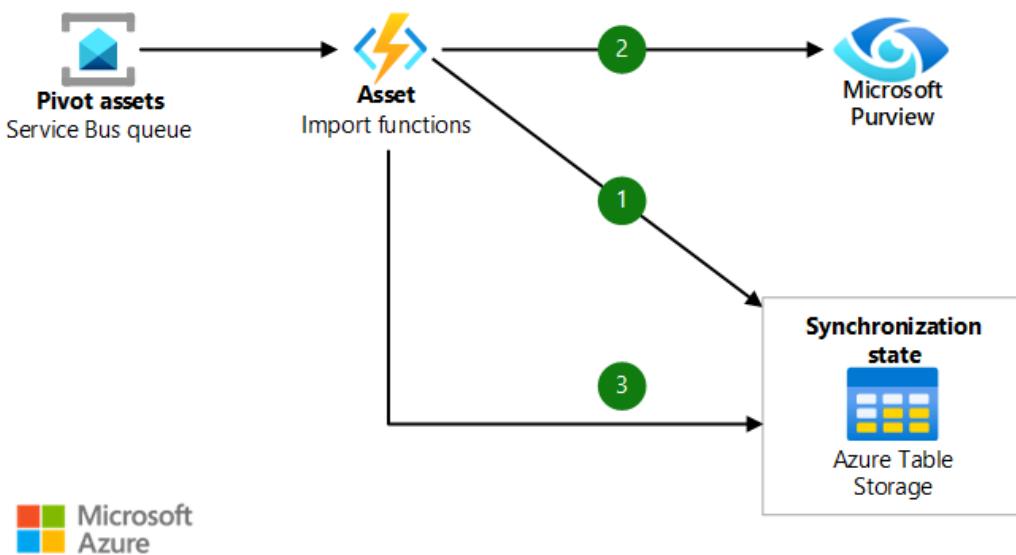
`Partition Key` and `Row Key` are used as a tuple to uniquely identify an object from the external catalog. The tuple uses the `Purview Object ID` property to link the object that's extracted from the external catalog to the object that's created in Microsoft Purview.

This configuration ensures that:

- Only one unique record exists per object for a given partition. For example, if three glossary terms are imported from Catalog A, there will be three entries in synchronization state that have `CatalogA_Glossary` as the partition key. Each entry will have a different row key, which is based on the unique identifier of that item in the external catalog.
- Synchronization state is used as a locking mechanism so that two messages can't import the same object at the same time. Table Storage is particularly useful for this purpose.

Import and synchronization state

The following diagram illustrates the details of the import flow of an asset and shows how the synchronization state storage is used:



Download a [Visio file](#) of this architecture.

Dataflow

1. The import step is triggered. Synchronization state storage is queried by `Partition Key` and `Row Key`:
 - a. If no entry exists, the import process initializes a new line with the `State` set to `Pending`. The `Row Key` is set to the unique identifier of the object in the external catalog and the `Partition Key` is set accordingly. A lock ensures that no other process can import the same metadata object at the same time. `Sync Start Time` and `Correlation ID` are also initialized.
 - b. If an entry already exists:

- i. If the `State` is `Failed`, the entry is updated with a new `Correlation ID` and a new `Sync Start Time`, and the `State` is set to `Pending`.
- ii. If the `State` is `Pending`, the message is scheduled to be re-queued later. For this purpose, the [dead-lettering queue](#) and re-queue mechanism that [Service Bus](#) offers by default is convenient.
- iii. If the `State` is `Completed`, the `Correlation ID` of the current run is compared to the one from the entry:
 - i. If they're different, the entry is updated with the current `Correlation ID` and a new `Sync Start Time`, and the `State` is set to `Pending`.
 - ii. Otherwise, the message is skipped.

 **Note**

All messages that belong to a single synchronization run have the same value for `Correlation ID`. This configuration ensures that each object is updated only one time per run. It assumes that there are no partial updates. You should consider how this behavior affects your implementation.

2. The object is created or updated in Microsoft Purview.
3. The unique identifier of the object, `Purview Object ID`, and the `Sync End Time` are written to synchronization state, and the `State` is changed to `Completed`. If the import fails, `State` is set to `Failed`.

You could use [Durable Functions](#) instead of the synchronization state intermediate storage.

Deletion

The process for deletion is based on `Sync Start Time` and `Sync End Time` in the synchronization state storage. Because the synchronization framework is triggered on a schedule, the objects in synchronization state that haven't been updated for a given time, even when multiple synchronizations have run during that time, have been removed from the external catalog. Therefore, those objects should also be removed from Microsoft Purview.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Julien Coriolan](#) | Principal Software Engineer
- [Adina Stoll](#) | Software Engineer 2

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Raouf Aliouat](#) | Software Engineer 2

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Get all type definitions - REST API \(Microsoft Purview\)](#)
- [Design a scalable partitioning strategy for Azure Table Storage \(REST API\)](#)
- [Triggers and bindings in Azure Functions](#)
- [Tutorial: How to use the Microsoft Purview Python SDK](#)
- [Compare Azure messaging services](#)

Related resources

- [Design a collection structure for a Microsoft Purview federated catalog](#)
- [Design Event Hubs and Functions for resilience](#)
- [Data governance with Profisee and Microsoft Purview](#)

Demand forecasting for shipping and distribution

Azure Blob Storage

Azure Data Factory

Power BI

Azure Stream Analytics

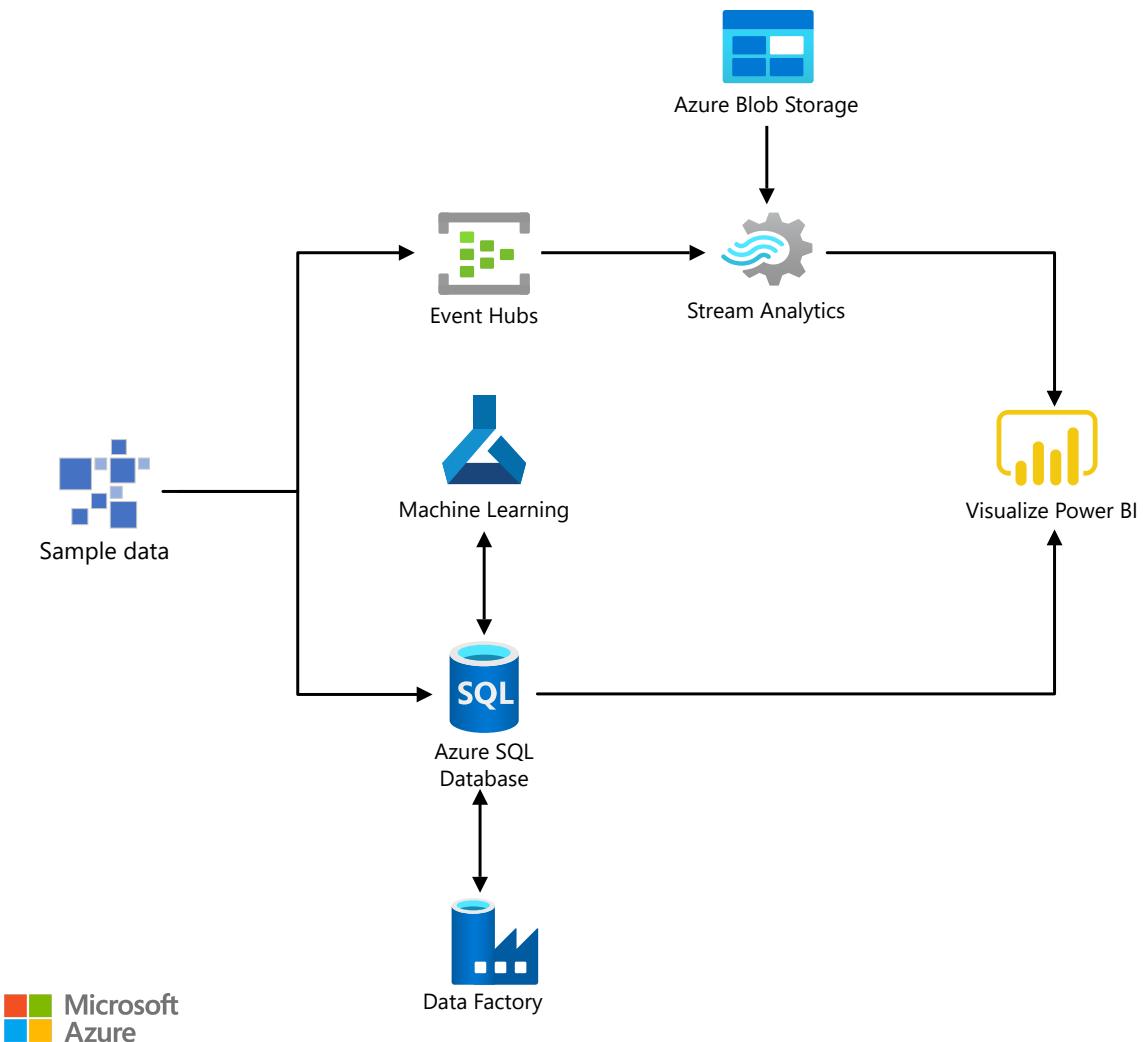
Azure Event Hubs

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This solution idea uses historical demand data to forecast demand in future periods across various customers, products, and destinations.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

For an example of a demand forecasting solution for shipping and distribution similar to the solution described in this article, see the [Azure AI Gallery](#). General characteristics of demand forecasting solutions like the one proposed here are:

- There are numerous kinds of items with differing volumes that roll up under one or more category levels.
- There's a history available for the quantity of the item at each time in the past.
- The volumes of the items differ widely, with possibly a large number that have zero volume at times.
- The history of items shows both trend and seasonality, possibly at multiple time scales.
- The quantities committed or returned aren't strongly price sensitive. In other words, the delivery company can't strongly influence quantities by short-term

changes in prices, although there may be other determinants that affect volume, such as weather.

Under these conditions, you can take advantage of the hierarchy formed among the time series of the different items. By enforcing consistency so that the quantities lower in the hierarchy (for example, individual product quantities) sum to the quantities above (customer product totals), you can improve the accuracy of the overall forecast. The same idea applies if individual items are grouped into categories, even for categories that overlap. For example, you might be interested in forecasting demand of all products in total, by location, by product category, or by customer.

The [AI Gallery solution](#) computes forecasts at all aggregation levels in the hierarchy for each time period specified. Remember that deployments of your demand forecasting solutions will incur consumption charges for the services used. Use the [Pricing Calculator](#) to predict costs. When you're no longer using a deployed solution, delete it to stop incurring charges.

Components

This demand forecasting solution idea uses the following resources hosted and managed in Azure:

- [Azure SQL Database](#) instance for persistent storage; to store forecasts and historical distribution data
- [Azure Machine Learning](#) web service to host forecasting code
- [Azure Blob Storage](#) for intermediate storage of generated forecasts
- [Azure Data Factory](#) to orchestrate regular runs of the Azure Machine Learning model
- [Power BI](#) dashboard to display and drill down on the forecasts

Scenario details

This solution uses historical demand data to forecast demand across customers, products, and destinations. One example of a use for this solution is when a shipping or delivery company wants to predict the quantities of the different products customers want delivered at different locations and at future times. The company can use demand forecasts as input to an allocation tool. The allocation tool can then optimize operations, such as delivery vehicle routing and planning capacity in the longer term. A related example is when a vendor or insurer wants to know the number of products that will be returned because of failures.

Potential use cases

The demand forecasting process described in this solution can be operationalized and deployed in [Microsoft AI platform](#). Microsoft AI platform has advanced analytics tools for data ingestion, data storage, scheduling, and advanced analytics. These tools are all the essential tools for running a demand forecasting solution that can be integrated with your current production systems.

This solution is optimized for the retail and manufacturing industries.

Next steps

See product documentation:

- [Learn more about Data Factory](#)
- [Learn more about Power BI](#)

Learn about:

- [Demand forecasting for shipping and distribution solution](#) in the Azure AI Gallery

Related resources

Read related Azure Architecture Center articles:

- [Demand forecasting and price optimization](#)
- [Demand forecasting with Azure Stream Analytics and Machine Learning](#)

Use a demand forecasting model for price optimization

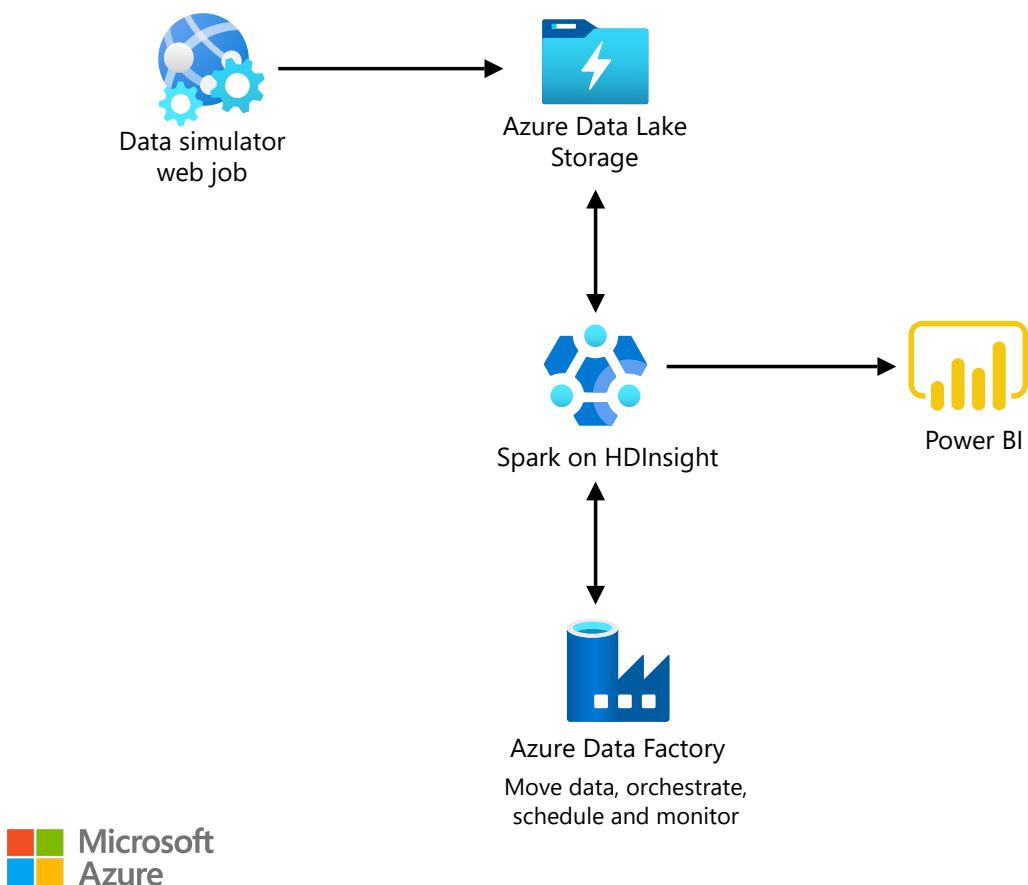
Azure Blob Storage Azure Data Factory Azure HDInsight Azure App Service Power BI

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This solution predicts future customer demand and optimizes pricing to maximize profitability using big-data and advanced-analytics services from Microsoft Azure.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

[Microsoft AI platform](#) provides advanced analytics tools such as data ingestion, storage, processing, and advanced analytics components. These tools are the essential elements for building a demand forecasting and price optimization solution.

1. Azure Data Lake (or Azure Blob Storage) stores the weekly raw sales data.
2. Apache Spark for Azure HDInsight ingests the data and executes data preprocessing, forecasting modeling, and price optimization algorithms.
3. Azure Data Factory orchestrates and schedules the entire data flow.

Components

- [Azure Data Lake Storage](#) stores the weekly raw sales data, which is read by Spark on HDInsight. As an alternative, use [Azure Blob Storage](#).
- Spark on [HDInsight](#) ingests the data and executes data preprocessing, forecasting modeling, and price-optimization algorithms.
- [Data Factory](#) handles orchestration and scheduling of the model retraining.
- [Power BI](#) enables visualization of results; monitor the results of the sales and predicted future demand and recommended optimal prices.

Scenario details

Pricing is pivotal for many industries, but it can be one of the most challenging tasks. Companies often struggle to accurately forecast the fiscal impact of potential tactics, fully consider core business constraints, and fairly validate pricing decisions once they've been made. As product offerings expand and complicate the calculations behind real-time pricing decisions, the process grows even more difficult.

This solution addresses those challenges by using historical transaction data to train a demand-forecasting model in a retail context. It also incorporates the pricing of products in a competing group to predict cannibalization and other cross-product impacts. A price-optimization algorithm then uses that model to forecast demand at various price points and factors in business constraints to maximize potential profit.

The process described above can be operationalized and deployed in [Microsoft AI platform](#).

Potential use cases

With this solution, you can ingest historical transaction data, predict future demand, and regularly optimize pricing, which saves you the time and effort you'd spend on pricing tasks.

Next steps

See product documentation:

- [Learn more about Data Lake Store](#)
- [Get started with HDInsight using a Spark cluster with R Server](#)
- [Learn more about Data Factory](#)
- [Learn more about Power BI](#)

External links about forecasting:

- [Demand forecasting and price optimization](#) in the Azure AI Gallery

Related resources

See related Azure Architecture Center articles:

- [Demand forecasting for shipping and distribution](#)
- [Demand forecasting with Azure Stream Analytics and Machine Line](#)

Demand forecasting

Azure Data Factory

Azure Event Hubs

Azure Machine Learning

Azure SQL Database

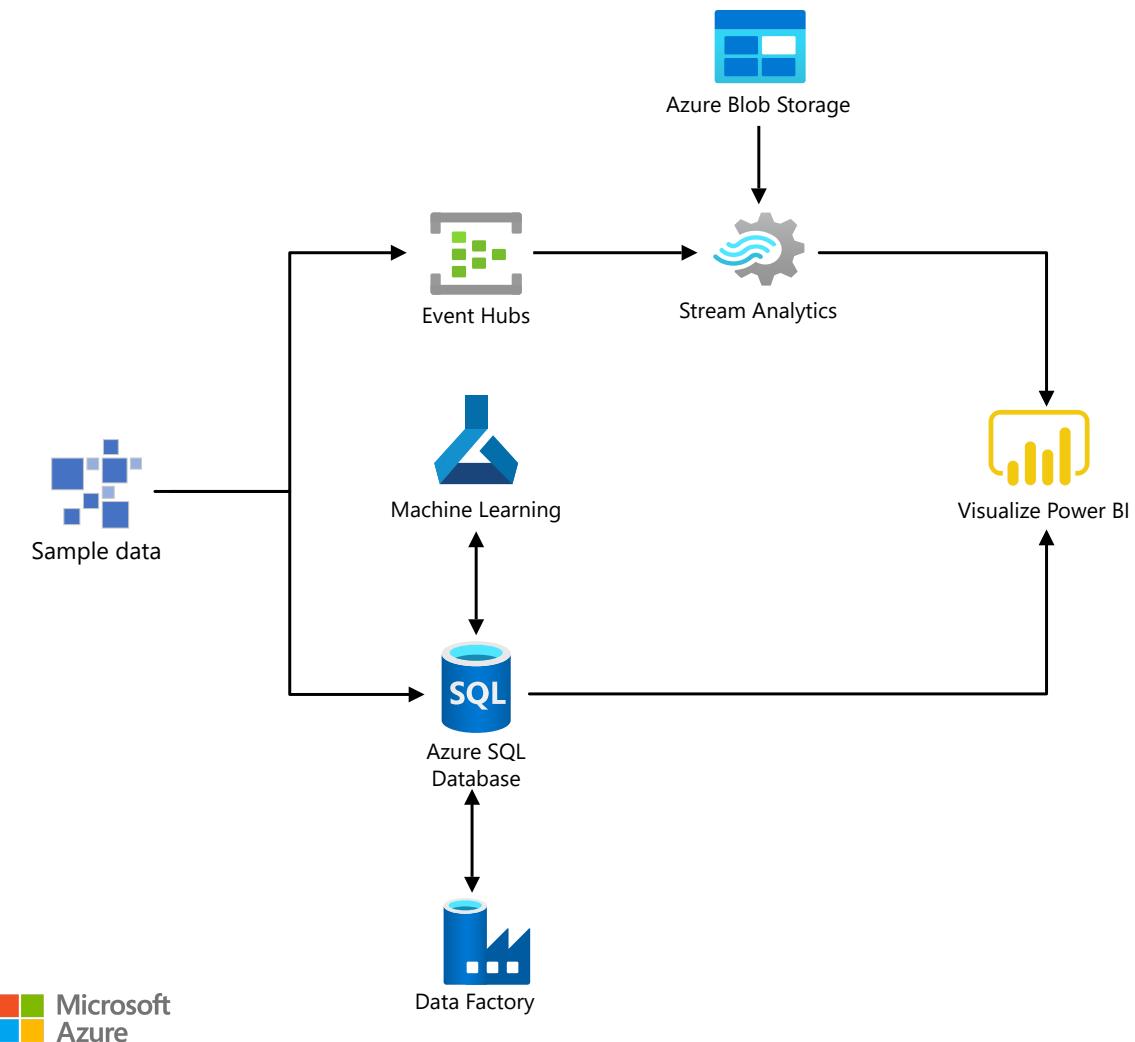
Azure Stream Analytics

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

Almost every business needs to predict the future to make better decisions and allocate resources more effectively. This article provides an architecture for an end-to-end demand-forecasting implementation on Azure.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

The Microsoft AI Platform provides advanced analytics tools through Microsoft Azure - data ingestion, data storage, data processing, and advanced analytics components. These tools include all of the essential elements for building a demand-forecasting-for-energy solution.

This solution combines several Azure services to provide actionable predictions:

1. Event Hubs collects real-time consumption data.
2. Stream Analytics aggregates the streaming data and makes it available for visualization.
3. Azure SQL Database stores and transforms the consumption data.
4. Machine Learning implements and executes the forecasting model.
5. Power BI visualizes the real-time energy consumption and the forecast results.
6. Finally, Data Factory orchestrates and schedules the entire dataflow.

Components

Key technologies used to implement this architecture:

- [Azure Event Hubs](#): Simple, secure, and scalable real-time data ingestion
- [Azure Stream Analytics](#): Provide Serverless real-time analytics, from the cloud to the edge
- [Azure SQL Database](#): Manage your intelligent SQL in the cloud
- [Azure Machine Learning](#): Build, deploy, and manage predictive analytics solutions
- [Power BI](#): Realize the value of your data and bring the insights discovered in Azure data and analytics tools to the organization.

Scenario details

This solution idea provides an architecture for forecasting demand. Accurately forecasting spikes in demand for products and services, for example, can give a company a competitive advantage. The better the forecasting, the more they can scale as demand increases, and the less they risk holding onto unneeded inventory. Use cases include predicting demand for a product in a retail/online store, forecasting hospital visits, and anticipating power consumption.

Potential use cases

The following scenarios are ways an organization can utilize demand forecasting:

- Inventory planning for retail
- Network capacity planning (telecommunications)
- Workforce planning
- Increased customer satisfaction

Next steps

- [Azure Machine Learning documentation](#)
- [Training: Get started with Azure Stream Analytics](#)
- [Welcome to Azure Stream Analytics](#)

Related resources

- [Use a demand forecasting model for price optimization](#)

- Demand forecasting for shipping and distribution
- Analytics architecture design
- Choose a real-time analytics and streaming processing technology on Azure

TimeXtender with cloud scale analytics

Azure Analysis Services

Azure Data Lake Storage

Azure Databricks

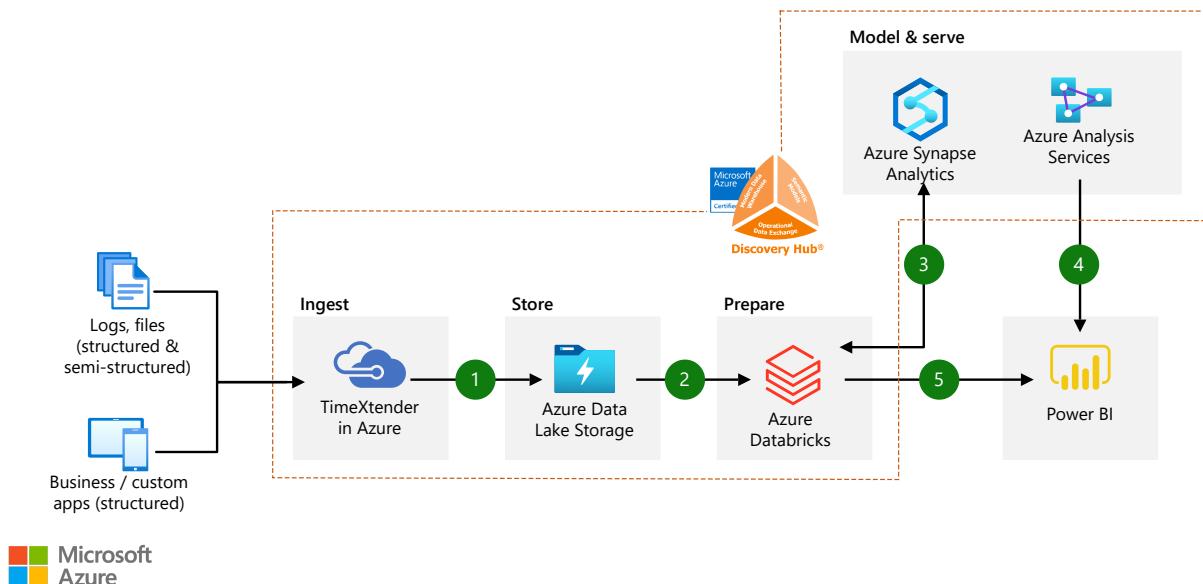
Azure Synapse Analytics

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This solution idea describes how to use the TimeXtender graphical interface to define a data estate.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

1. Combine all your structured and semi-structured data in Azure Data Lake Storage using TimeXtender's data engineering pipeline with hundreds of native data connectors.

2. Clean and transform data using the powerful analytics and computational ability of Azure Databricks.
3. Move cleansed and transformed data to Azure Synapse Analytics, creating one hub for all your data. Take advantage of native connectors between Azure Databricks (PolyBase) and Azure Synapse Analytics to access and move data at scale.
4. Build operational reports and analytical dashboards on top of SQL Database to derive insights from the data and use Azure Analysis Services to serve the data.
5. Run ad-hoc queries directly on data within Azure Databricks.

Components

- [Azure Data Lake Storage](#): Massively scalable, secure data lake functionality built on Azure Blob Storage
- [Azure Databricks](#): Fast, easy, and collaborative Apache Spark-based analytics platform
- [Azure Synapse Analytics](#): Limitless analytics service with unmatched time to insight (formerly SQL Data Warehouse)
- [Azure Analysis Services](#): Enterprise-grade analytics engine as a service
- [Power BI Embedded](#): Embed fully interactive, stunning data visualizations in your applications

Scenario details

You can use TimeXtender to define a data estate via a graphical user interface. Definitions are stored in a metadata repository. Code for building the data estate is generated automatically while remaining fully customizable. The results are a modern data warehouse that is ready to support cloud scale analytics and AI.

Potential use cases

- No infrastructure issues or maintenance
- Consistent performance
- Deploy and manage both the architecture and the data pipelines, data models and semantic models

Next steps

- [Azure Data Lake Storage documentation](#)
- [Azure Databricks documentation](#)
- [Azure Synapse Analytics documentation](#)

- [Azure Analysis Services documentation](#)
- [Power BI Embedded documentation](#)

Related resources

- [Modern data warehouse for small and medium business](#)
- [Data warehousing and analytics](#)
- [Modern analytics architecture with Azure Databricks](#)

Enhanced customer dimension with Dynamics 365 Customer Insights

Azure Data Lake Storage

Azure Synapse Analytics

Azure Data Factory

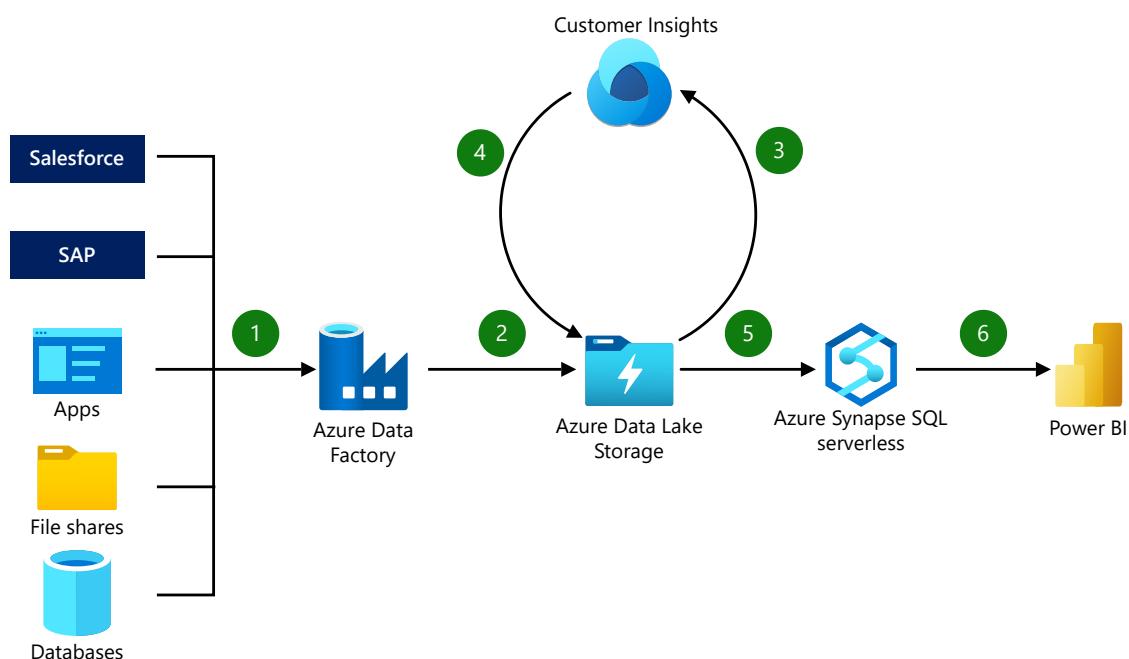
Customer Insights - Data

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This high-level architecture shows the flow of data from an organization's source systems (ERP, CRM, POS, and so on) into a data lake on Azure. This same data lake can be configured as the back end for Dynamics 365 Customer Insights. When it has a data lake back end, Customer Insights can load clean enhanced customer data into the data lake for consumption as a dimension by downstream data warehouses and apps.

Architecture



Download a [Visio file](#) of this architecture.

Azure Synapse serverless SQL consumes the enhanced Customer Insights data. Azure Synapse serverless SQL introduces a cost-effective design pattern known as Logical Data Warehouse (LDW). The LDW pattern introduces an abstraction layer on top of external data stores, like data lakes, to provide familiar relational database constructs like tables and views. Tools that support SQL Server endpoints can then consume these tables and views. In the context of this example, Power BI can source the enhanced Customer Insights data as a dimension table from a database by using Azure Synapse serverless SQL pools.

Dataflow

1. By using Data Factory or Azure Synapse pipelines, establish [linked services](#) to source systems and data stores. Data Factory and Azure Synapse pipelines support [more than 90 connectors](#), including generic protocols for data sources when a native connector isn't available.
2. Load data from the source systems into Data Lake by using the [Copy Data tool](#). You then need to transform data in the data lake to fit a Common Data Model schema. Data Factory mapping data flows support sinking data in the Common Data Model format. For more information, see [Common Data Model format in Azure Data Factory and Synapse Analytics](#).
3. To import data into Customer Insights, you need to configure a [connection to a Common Data Model folder by using a Data Lake account](#). After you import data into Customer Insights, the Customer Insights [data unification process \(map, match, and merge\)](#) can process the disparate customer data. You can then further enrich unified data in Customer Insights by using [data enrichment](#), [data segments](#), and [AI predictions](#).
4. In Customer Insights, you need to configure an export of data back to the data lake. For more information, see [Set up the connection to Azure Data Lake Storage Gen2](#).
5. [Create a Logical Data Warehouse](#) in the Azure Synapse workspace. See the [Azure Synapse serverless SQL pool best practices](#) to determine whether you need to do more transformations on the exported Customer Insights data and whether views are better suited than tables.
6. Customer Insights data in the data lake is now exposed as logical SQL Server tables and views that can easily be consumed by Power BI. See [Tutorial for using serverless SQL pools with Power BI](#) for an example.

Components

- [Azure Data Lake Storage](#). Scalable and cost-effective cloud storage that Customer Insights supports as a target for exporting data.
- [Azure Data Factory](#). Cloud-scale data integration service for orchestrating data flow.
- [Audience insights](#). The Customer Insights module that unifies customer data sources. It also provides enrichments like segmentation, customer total lifetime value (CTLV), and customer churn score.
- [Azure Synapse serverless SQL pools](#). Used to query customer data in a data lake via T-SQL and SQL Server endpoint.

Alternatives

This solution uses the Logical Data Warehouse (LDW) pattern to consume the enhanced data from Customer Insights. You can also use other data warehouse patterns.

Data Factory and Azure Synapse both provide data integration pipelines. See the [breakdown of feature parity](#) for a comparison.

Scenario details

[Dynamics 365 Customer Insights](#) can create a 360-degree customer view by unifying data from transactional, behavioral, and observational sources. You can then make this 360-degree customer view available in enterprise data lakes and/or data warehouses as an enhanced customer [dimension](#).

This article describes the dataflow, product integrations, and configurations that are available for building an enhanced customer dimension that can be consumed by analytics platforms external to Dynamics 365 and Customer Insights. [Audience insights](#) is the feature of Customer Insights that provides the ability to unify customer data sources and enhance customer profiles. For more information, see [the audience insights overview](#).

The following table shows an example of enhanced customer records that are produced by the Customer Insights data unification process. This process takes customer data from multiple source systems and cleans and merges it. Customer Insights can also enrich customer records with attributes like churn scores and brand affinities. Here are some fictional examples of this type of record:

CustomerId	Name	Address	City	State	PostalCode	Country	EmailId	DOB	RewardsPoints	ChurnScore
e7b8e460e37...	Gustie Spacie	199 Burning Wood Point	Baton Rouge	Louisiana	70820	United States	gspacie@fabrikam.com	1975-01-13	263	0.230083182454109
2b23ec54b29...	Broddie Kempston	40 Muir Terrace	Chicago	Illinois	60669	United States	bkempson@proseware.com	1943-01-28	188	0.702696681022644
0019a5d0605...	Lorrain Ruffles	33 Brown Drive	Santa Monica	California	90410	United States	lruffles@treyresearch.net	1988-10-25	178	0.802143514156342

CustomerId	Name	Industry	Brand	AffinityScore	AffinityLevel	AgeDemographicSegment
e7b8e460e37...	Gustie Spacie	Computers & Consumer Electronics	Adatum Cor	48	Medium	AGE35_49
e7b8e460e37...	Gustie Spacie	Food & Groceries	Best for you	49	Medium	AGE35_49
e7b8e460e37...	Gustie Spacie	Retailers & General Merchandise	Northwind	74	High	AGE35_49
e7b8e460e37...	Gustie Spacie	Retailers & General Merchandise	Tailspin Toy	86	VeryHigh	AGE35_49

Potential use cases

This architecture is applicable to any organization that needs to create records that draw data from multiple sources.

This solution is optimized for the retail industry.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Jon Dobrzeniecki](#) | Cloud Solution Architect

Next steps

- [Microsoft Learn: Unlock customer intent with Dynamics 365 audience insights](#)
- [Tutorial: Explore and analyze data lakes with serverless SQL pool](#)
- [Tutorial: Create a Logical Data Warehouse with serverless SQL pool](#)
- [Get started with Azure Synapse Analytics](#)
- [Customer Insights overview](#)
- [Analyze data in a storage account](#)
- [Integrate activities by using pipelines](#)

Related resources

- [Get started with analytics architecture design](#)
- [Choose an analytical data store in Azure](#)
- [Analytics end-to-end with Azure Synapse](#)

Enterprise data warehouse

Azure Blob Storage Azure Data Lake Azure Synapse Analytics

💡 Solution ideas

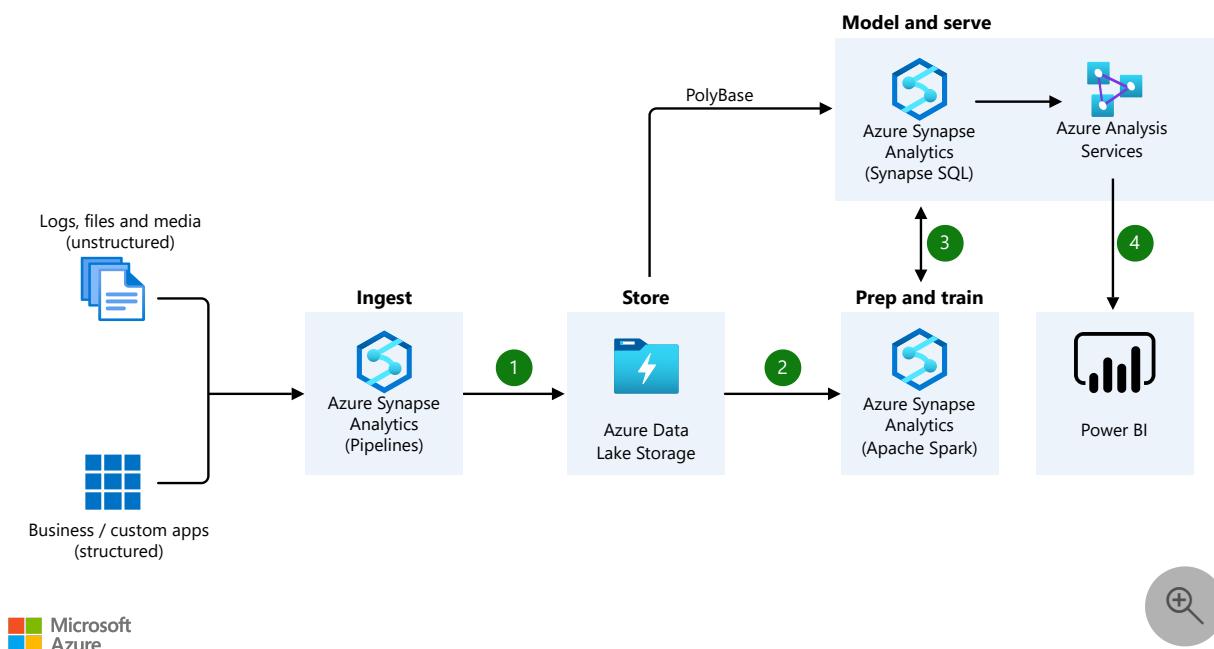
This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This article presents a solution for an enterprise data warehouse in Azure that:

- Brings together all your data, no matter the scale or format.
- Provides a way for all your users to get insights from your data through analytical dashboards, operational reports, and advanced analytics.

Apache® and [Apache Spark](#) 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

1. Azure Synapse Analytics pipelines bring together structured, unstructured, and semi-structured data, such as logs, files, and media. The pipelines store the data in Azure Data Lake Storage.
2. Apache Spark pools in Azure Synapse Analytics clean and transform the Data Lake Storage data.
3. Azure Synapse Analytics combines the processed data with existing structured data, creating one unified data hub.
4. A dedicated SQL pool makes the data available for operational reports and analytical dashboards that derive insights. Azure Analysis Services serves the reports and dashboards to thousands of end users.

Components

- [Azure Synapse Analytics](#) is an analytics service for data warehouses and big data systems. This tool uses a massively parallel processing architecture and has deep integration with Azure services.
- [Azure Synapse Analytics pipelines](#) provide a way for you to create, schedule, and orchestrate workflows, such as extract, load, transform (ELT) and extract, transform, load (ETL) workflows.
- [Azure Blob Storage](#) provides massively scalable, cost-effective object storage for any type of unstructured data—images, videos, audio, documents, and more.
- [Data Lake Storage](#) is a storage repository that holds a large amount of data in its native, raw format. Data Lake Storage is built on top of Blob Storage. As a result, Data Lake Storage offers the scalability, tiered storage, high availability, and disaster recovery capabilities of Blob Storage.
- [Azure Synapse Analytics Spark pools](#) provide a parallel processing framework that supports in-memory processing to boost the performance of big-data analytic applications.
- [Analysis Services](#) is an enterprise-grade analytics engine that provides an easy way for users to perform ad hoc data analysis. You can use Analysis Services to govern, test, and deliver business solutions at scale.
- [Power BI](#) is a suite of business analytics tools that deliver insights throughout your organization. You can use Power BI to connect to hundreds of data sources, simplify data preparation, and drive ad hoc analysis. You can also produce beautiful reports and publish them for your organization to consume on the web and across mobile devices.

Scenario details

An enterprise data warehouse brings all your data together, no matter the source, format, or scale. A data warehouse also provides a way for you to run high-performance analytics on your data, so you can gain insights through analytical dashboards, operational reports, and advanced analytics.

This solution establishes a data warehouse that:

- Is a single source of truth for your data.
- Integrates relational data sources with other unstructured datasets.
- Uses semantic modeling and powerful visualization tools for simpler data analysis.

To integrate data into a unified platform, this solution uses Azure Synapse Analytics pipelines. These pipelines offer ELT and ETL capabilities. Specifically, you can use the pipelines to move data in data-driven workflows. The pipelines work with various data formats and structures.

The pipelines store the data in Data Lake Storage, which is built on Blob Storage. This storage service can handle large volumes of unstructured data.

Azure Synapse Analytics Spark pools form a key part of the solution. These pools clean and transform data that's stored in Azure. Their parallel processing framework supports in-memory processing for speed and efficiency. The pools also support auto-scaling, so they can add or remove nodes as needed.

A dedicated SQL pool makes the processed data available for high-performance analytics. This pool stores data in relational tables with columnar storage, a format that significantly reduces the cost of data storage. It also improves query performance, so you can run analytics at massive scale.

Potential use cases

You can use this solution in scenarios like the following ones that involve large volumes of data:

- IoT device integration
- Customer data platforms
- Natural language processing
- Machine learning algorithms

Pricing

To view an estimate of the cost of this solution, see a [pricing sample in the pricing calculator](#).

Next steps

- [Azure Synapse Analytics documentation](#)
- [Azure Synapse Analytics pipelines documentation](#)
- [Introduction to object storage in Azure](#)
- [Azure Synapse Analytics Spark pools](#)
- [Analysis Services documentation](#)
- [Power BI documentation](#)

Related resources

- [Data warehousing and analytics](#)
- [Big data analytics with enterprise-grade security using Azure Synapse](#)
- [Logical data warehouse with Azure Synapse serverless SQL pools](#)
- [Modern data warehouse for small and medium business](#)

Extract, transform, and load (ETL) using HDInsight

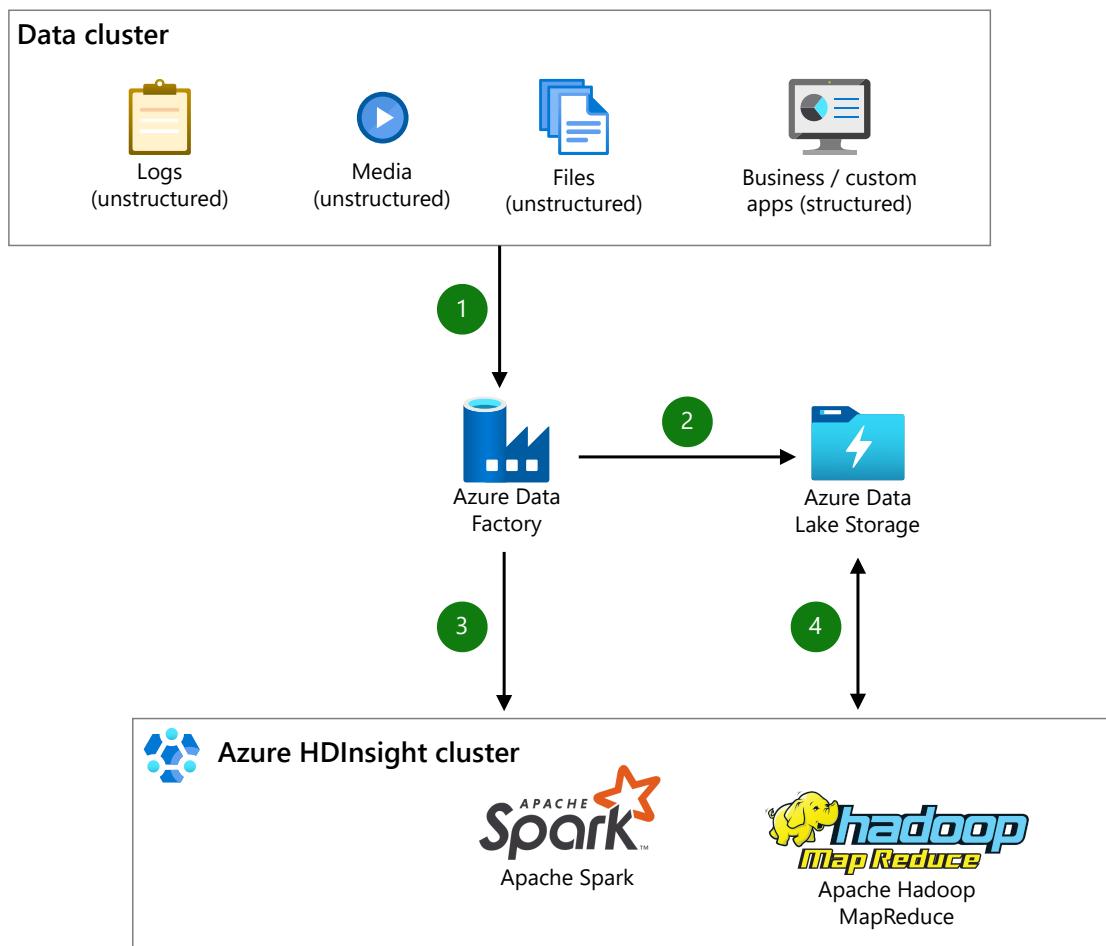
Azure Data Factory Azure Data Lake Storage Azure HDInsight

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This solution idea illustrates how to extract, transform, and load your big data clusters on demand by using Hadoop MapReduce and Apache Spark.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

The data flows through the architecture as follows:

1. Using Azure Data Factory, establish [Linked Services](#) to source systems and data stores. Azure Data Factory Pipelines support 90+ connectors that also include generic protocols for data sources where a native connector isn't available.
2. Load data from source systems into Azure data lake with the [Copy Data tool](#).
3. Azure Data Factory is able to create an on-demand HDInsight cluster. Start by creating an [On-Demand HDInsight Linked Service](#). Next, [create a pipeline](#) and use the appropriate HDInsight activity depending on the Hadoop framework being used (that is, Hive, MapReduce, Spark, etc.).
4. Trigger the pipeline in Azure Data Factory. The architecture assumes Azure Data Lake store is used as the file system in the Hadoop script executed by the HDInsight activity which was created in Step 3. The script will be executed by an on-demand HDInsight cluster that will write data to a curated area of the data lake.

Components

- [Azure Data Factory](#) - Cloud scale data integration service for orchestrating data flow.
- [Azure Data Lake Storage](#) - Scalable and cost-effective cloud storage for big data processing.
- [Apache Hadoop](#) - Big data distributed processing framework
- [Apache Spark](#) - Big data distributed processing framework that supports in-memory processing to boost performance for big data applications.
- [Azure HDInsight](#) - Cloud distribution of Hadoop components.

Scenario details

This solution idea describes the data flow for an ETL use case.

Potential use cases

You can use Azure HDInsight for various scenarios in big data processing. It can be historical data (data that's already collected and stored) or real-time data (data that's

directly streamed from the source). For more information about processing such data, see [Scenarios for using HDInsight](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Jon Dobrzeniecki](#) | Cloud Solution Architect

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

Learn more about the component technologies:

- [Tutorial: Create on-demand Apache Hadoop clusters in HDInsight using Azure Data Factory](#)
- [Introduction to Azure Data Factory](#)
- [Introduction to Azure Data Lake Storage Gen2](#)
- [Load data into Azure Data Lake Storage Gen2 with Azure Data Factory](#)
- [What is Apache Hadoop in Azure HDInsight?](#)
- [Invoke MapReduce Programs from Data Factory](#)
- [Use MapReduce in Apache Hadoop on HDInsight](#)
- [What is Apache Spark in Azure HDInsight](#)

Related resources

Explore related architectures:

- [Use a demand forecasting model for price optimization](#)
- [Predictive insights with vehicle telematics](#)

Deliver highly scalable customer service and ERP applications

Azure Cosmos DB

Azure Data Lake Storage

Azure SQL Database

Azure Synapse Analytics

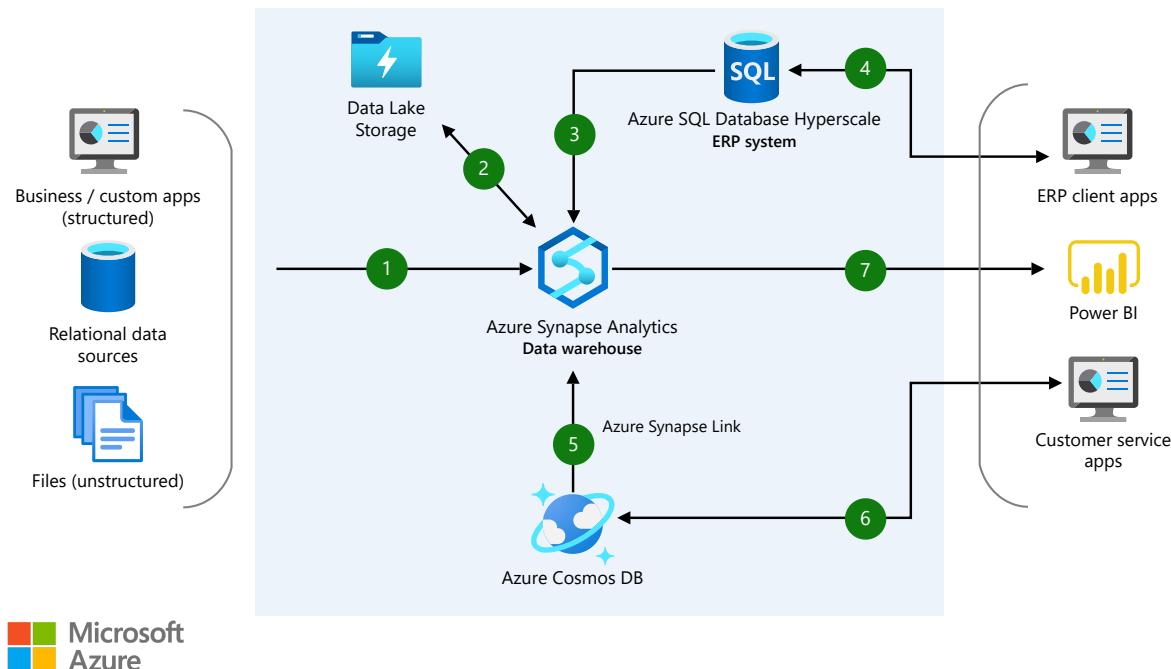
Power BI

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This solution idea describes how you can use Azure managed databases and Azure Synapse Analytics to get insights via ERP applications and Power BI.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

First, the company must ingest data from various sources.

1. Use Azure Synapse Pipelines to ingest data of all formats.

2. Land data in Azure Data Lake Storage Gen 2, a highly scalable data lake.

From there, they use Azure SQL Database Hyperscale to run a highly scalable ERP system:

3. Ingest relational data using Azure Synapse Pipelines into Azure SQL Database. The company's ERP system runs on Azure SQL Database and applies the Hyperscale service tier to scale compute or storage up to 100 TB.
4. This data is surfaced via ERP client applications to help the company manage their business processes.

To improve service to their customers, they build highly scalable customer service applications that can scale to millions of users:

5. Provide near real-time analytics and insight into user interaction with applications by applying Azure Synapse Link for Azure Cosmos DB HTAP capabilities, with no ETL needed.
 6. Power customer service applications with Azure Cosmos DB for automatic and instant scalability and SLA-backed speed, availability, throughput, and consistency.
- Finally, they surface business intelligence insights to users across the company to power data-driven decisions:
7. Power BI tightly integrates with Azure Synapse Analytics to provide powerful insights over operational, data warehouse, and data lake data.

Components

- [Azure Data Lake Storage](#) provides massively scalable and secure data lake storage for high-performance analytics workloads.
- [Azure Synapse Analytics](#) is an analytics service that brings together enterprise data warehousing and Big Data analytics within a unified experience.
- [Azure SQL Database Hyperscale](#) is a storage tier in [Azure SQL Database](#) that uses Azure architecture to scale out storage and compute resources. Hyperscale supports up to 100 TB of storage and provides nearly instantaneous backups and fast database restores in minutes – regardless of the size of data operation.
- [Azure Cosmos DB](#) is a fully managed NoSQL database service for building and modernizing scalable, high-performance applications.
- [Power BI](#) is a suite of business tools for self-service and enterprise business intelligence (BI). Here, it's used to analyze and visualize data.

Scenario details

Today's organizations are generating ever-increasing amounts of structured and unstructured data. With Azure managed databases and Azure Synapse Analytics, they can deliver insights to their employees via ERP applications and Power BI, as well as superior customer service through web and mobile applications, scaling without limits as data volumes and application users increase.

Potential use cases

Organizations utilize ERP to assist with:

- Cost savings (automate simple tasks)
- Workflow visibility (managers can see project status)
- Regulatory compliance
- Data security
- Customer management (track survey responses, support tickets, and returns)

Next steps

- [Azure Data Lake Storage](#)
- [Azure Synapse Analytics](#)
- [Azure Cosmos DB](#)
- [Azure Synapse Link](#)
- [Azure Synapse Link for Azure Cosmos DB: Near real-time analytics use cases](#)
- [Power BI](#)

Related resources

- [Azure Cosmos DB resource model](#)

Extend your on-premises big data investments with HDInsight

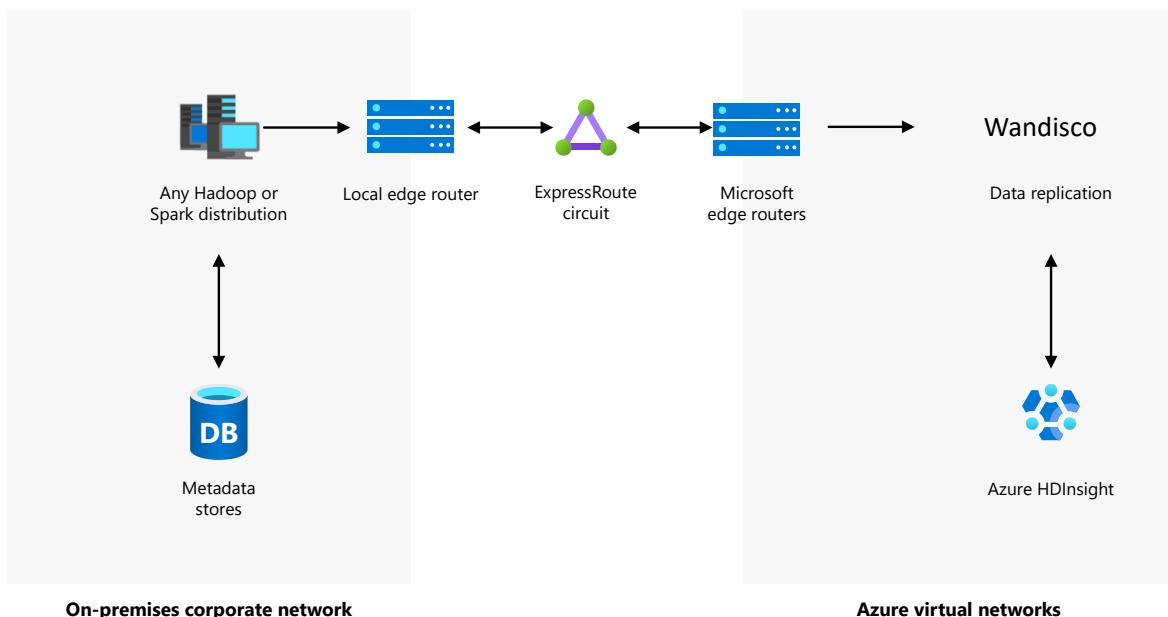
Azure HDInsight

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This solution idea describes how to extend your on-premises big data investments to the cloud and transform your business by using the advanced analytics capabilities of Azure HDInsight.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

1. Establish ExpressRoute between on-premises infrastructure and Microsoft datacenters, to allow private connection for reliable, speedy, and secure data replication from an on-premises Hadoop setup to an Azure HDInsight cluster.
2. Install the WANdisco Fusion server in the same Azure Virtual Network as the HDInsight cluster, which allows the server to access the cluster in a secure manner.
3. Install the WANdisco Fusion app on a HDInsight cluster (new or existing). In the License key field, enter the Public IP of the Fusion Server.
4. Configure the Fusion App on an HDInsight cluster to set up continuous active replication from on-premises large data/Hadoop deployments to Azure HDInsight, multi-region replication, backup and restore, and more.

Components

- [Apache Hadoop](#) or [Apache Spark](#)
- Metadata store
- Local edge router
- [Azure ExpressRoute](#) circuit
- Microsoft Edge router
- Data replication (WANdisco's [LiveData Migrator for Azure](#) and [LiveData Plane for Azure](#))
- [Azure HDInsight](#)
- [Azure Virtual Network](#)

Scenario details

This solution idea describes how to extend your on-premises big data investments to the cloud.

Potential use cases

The integration of WANdisco Fusion with Azure HDInsight presents an enterprise solution that enables organizations to meet stringent data availability and compliance requirements while seamlessly moving production data at petabyte scale from on-premises big data deployments to Microsoft Azure.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Aadi Manchanda](#) | Cloud Solution Architect

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

Learn more about the component technologies:

- [What is Azure ExpressRoute?](#)
- [Migrate your Hadoop data lakes with WANDisco LiveData Platform for Azure](#)
- [What is Azure HDInsight?](#)
- [What is Azure Virtual Network?](#)

Related resources

Explore related architectures:

- [Connect an on-premises network to Azure using ExpressRoute](#)
- [Extend an on-premises network using ExpressRoute](#)

Ingestion, ETL, and stream processing pipelines with Azure Databricks and Delta Lake

Azure Databricks

Azure Data Lake Storage

Azure IoT Hub

Azure Data Factory

Azure Event Hubs

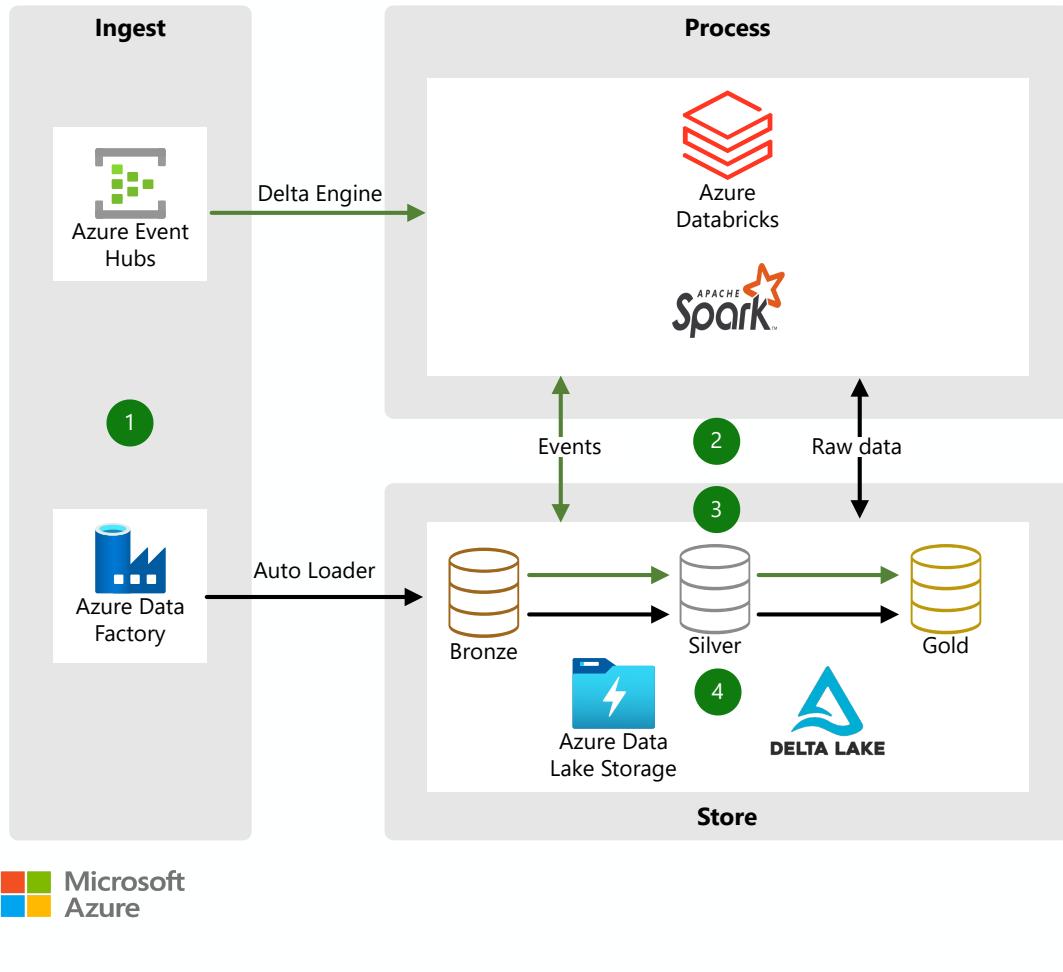
💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

Your organization needs to ingest data of any format, size, and speed into the cloud in a consistent way. The solution in this article meets that need with an architecture that implements extract, transform, and load (ETL) from your data sources to a data lake. The data lake can hold all the data, including transformed and curated versions at various scales. The data can be used for data analytics, business intelligence (BI), reporting, data science, and machine learning.

Apache® and Apache Spark™ 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

1. Data is ingested in the following ways:
 - Event queues like Event Hubs, IoT Hub, or Kafka send streaming data to Azure Databricks, which uses the optimized Delta Engine to read the data.
 - Scheduled or triggered Data Factory pipelines copy data from different data sources in raw formats. The [Auto Loader in Azure Databricks](#) processes the data as it arrives.
2. Azure Databricks loads the data into optimized, compressed Delta Lake tables or folders in the Data Lake Storage Bronze layer.
3. Streaming, scheduled, or triggered Azure Databricks jobs read new transactions from the Data Lake Storage Bronze layer. The jobs join, clean, transform, and aggregate the data before using ACID transactions to load it into curated data sets in the Data Lake Storage Silver and Gold layers.

4. The data sets are stored in Delta Lake in Data Lake Storage.

Each service ingests data into a common format to ensure consistency. The architecture uses a shared data lake based on the open Delta Lake format. Raw data is ingested from different batch and streaming sources to form a unified data platform. The platform can be used for downstream use cases such as analytics, BI reporting, data science, AI, and machine learning.

Bronze, Silver, and Gold storage layers

With the medallion pattern, consisting of Bronze, Silver, and Gold storage layers, customers have flexible access and extendable data processing.

- **Bronze** tables provide the entry point for raw data when it lands in Data Lake Storage. The data is taken in its raw source format and converted to the open, transactional Delta Lake format for processing. The solution ingests the data into the Bronze layer by using:
 - Apache Spark APIs in Azure Databricks. The APIs read streaming events from Event Hubs or IoT Hub, and then convert those events or raw files to the Delta Lake format.
 - The **COPY INTO** command. Use the command to copy data directly from a source file or directory into Delta Lake.
 - The Azure Databricks Auto Loader. The Auto Loader grabs files when they arrive in the data lake and writes them to the Delta Lake format.
 - The Data Factory Copy Activity. Customers can use this option to convert the data from any of its supported formats into the Delta Lake format.
- **Silver** tables store data while it's being optimized for BI and data science use cases. The Bronze layer ingests raw data, and then more ETL and stream processing tasks are done to filter, clean, transform, join, and aggregate the data into Silver curated datasets. Companies can use a consistent compute engine, like the open-standards [Delta Engine](#), when using Azure Databricks as the initial service for these tasks. They can then use familiar programming languages like SQL, Python, R, or Scala. Companies can also use repeatable DevOps processes and ephemeral compute clusters sized to their individual workloads.
- **Gold** tables contain enriched data, ready for analytics and reporting. Analysts can use their method of choice, such as PySpark, Koalas, SQL, Power BI, and Excel to gain new insights and formulate queries.

Components

- [Event Hubs](#) parses and scores streaming messages from various sources, including on-premises systems, and provides real-time information.
- [Data Factory](#) orchestrates data pipelines for ingestion, preparation, and transformation of all your data at any scale.
- [Data Lake Storage](#) brings together streaming and batch data, including structured, unstructured, and semi-structured data like logs, files, and media.
- [Azure Databricks](#) cleans and transforms the structureless data sets and combines them with structured data from operational databases or data warehouses.
- [IoT Hub](#) gives you highly secure and reliable communication between your IoT application and devices.
- [Delta Lake](#) on Data Lake Storage supports ACID transactions for reliability and is optimized for efficient ingestion, processing, and queries.

Scenario details

Ingestion, ETL, and stream processing with Azure Databricks is simple, open, and collaborative:

- **Simple:** An open data lake with a curated layer in an open-source format simplifies the data architecture. Delta Lake, an open-source tool, provides access to the Azure Data Lake Storage data lake. Delta Lake on Data Lake Storage supports atomicity, consistency, isolation, and durability (ACID) transactions for reliability. Delta Lake is optimized for efficient ingestion, processing, and queries.
- **Open:** The solution supports open-source code, open standards, and open frameworks. It also works with popular integrated development environments (IDEs), libraries, and programming languages. Through native connectors and APIs, the solution works with a broad range of other services, too.
- **Collaborative:** Data engineers, data scientists, and analysts work together with this solution. They can use collaborative notebooks, IDEs, dashboards, and other tools to access and analyze common underlying data.

Azure Databricks seamlessly integrates with other Azure services like Data Lake Storage, Azure Data Factory, Azure Event Hubs, and Azure IoT Hub.

Potential use cases

This solution is inspired by the system that [Providence Health Care](#) built for real-time analytics. Any industry that ingests batch or streaming data could also consider this solution. Examples include:

- Retail and e-commerce

- Finance
- Healthcare and life sciences
- Energy suppliers

Next steps

- [Providence Health Care](#) builds their data streaming solution using Azure Databricks and Azure Event Hubs to improve the National Emergency Department Overcrowding Score for each of its emergency departments.
- [Spanish Point Technologies](#) builds its Matching Engine using Azure Databricks and Azure Data Factory to ingest data at scale to help musicians get paid fairly.

Related resources

Guides and fully deployable architectures:

- [Choose an analytical data store in Azure](#)
- [Stream processing with Azure Databricks](#)
- [Automated enterprise BI](#)

Logical data warehouse with Azure Synapse serverless SQL pools

Azure Cosmos DB

Azure Data Factory

Azure Data Lake

Azure Synapse Analytics

Power BI

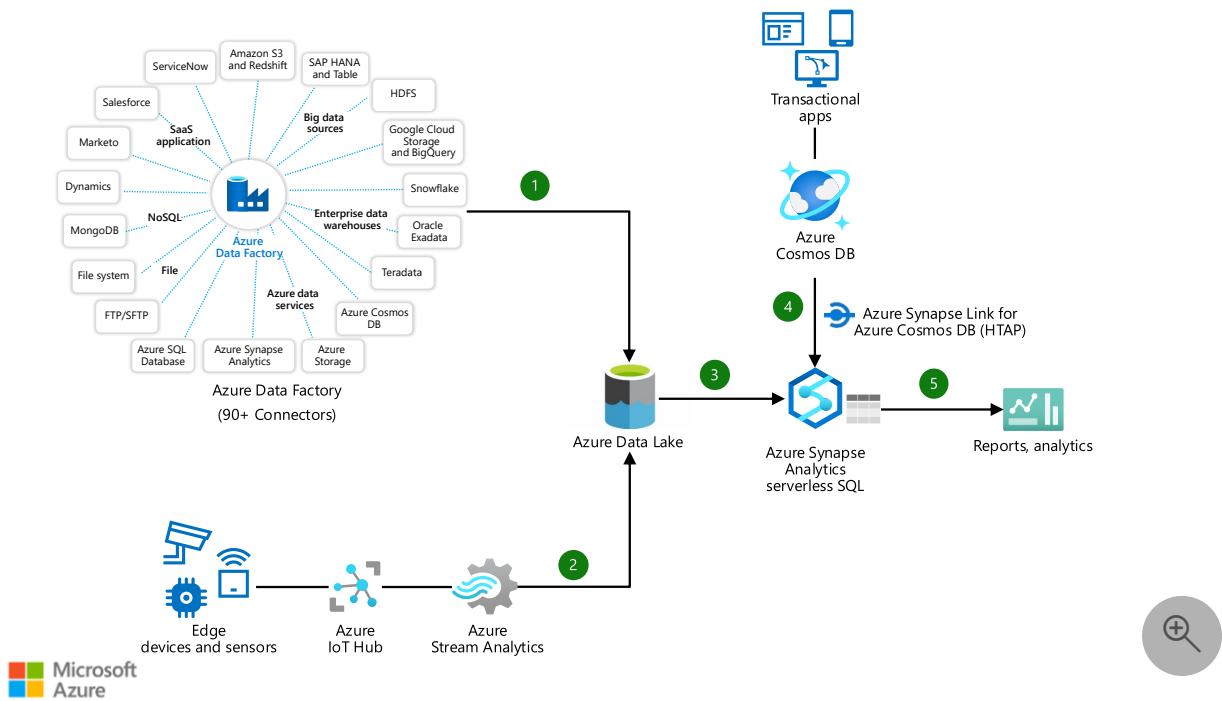
💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

The logical data warehouse (LDW) pattern lays a lightweight virtualized relational layer on top of data that's stored in a data lake or database. This virtualization layer provides data warehouse access without requiring data movement. This solution can combine online transaction processing (OLTP) data with analytical data from data lakes for a low-complexity, low-latency way to serve business intelligence (BI) and analytics workloads.

Apache Spark™ is a trademark of the Apache Software Foundation in the United States and/or other countries/regions. No endorsement by The Apache Software Foundation is implied by the use of this mark.

Architecture



Download a [PowerPoint file](#) of all the diagrams in this article.

Dataflow

1. Azure Data Factory integrates data from source systems into the enterprise data lake.
2. Device and sensor data also streams from edge devices into the cloud through Azure IoT Hub. Azure Stream Analytics processes the data and sends it to the enterprise data lake.
3. Azure Synapse serverless SQL pools define an LDW that has logical tables and views accessible through the Azure Synapse workspace [serverless SQL pool on-demand endpoint](#).
4. Azure Synapse Link for Azure Cosmos DB queries real-time transactional data through the Azure Synapse serverless SQL pools. This data joins with cold batch and hot streaming data from the enterprise data lake to create logical views.
5. Reporting, BI, and other analytics applications access LDW data and views by using the Azure Synapse workspace serverless SQL endpoint.

ⓘ Note

The Azure Synapse workspace serverless SQL endpoint is accessible from any tool or service that supports Tabular Data Stream (TDS) connections to SQL Server.

Components

- [Azure Synapse Analytics](#) is a limitless analytics service that brings together data integration, enterprise data warehousing, and big data analytics.
 - [Azure Synapse serverless SQL pools](#) query data lakes by using T-SQL and serverless SQL on-demand endpoints.
 - [Azure Synapse Link for Azure Cosmos DB](#) queries Azure Cosmos DB OLTP data by using Azure Synapse serverless SQL pools.
- [Data Factory](#) offers cloud-scale data integration and data flow orchestration.
- [IoT Hub](#) enables secure and reliable communication between internet of things (IoT) applications and devices.
- [Stream Analytics](#) provides serverless, real-time streaming analytics pipelines.
- [Azure Data Lake Storage](#) offers scalable, cost-effective cloud storage.
- [Azure Cosmos DB](#) is a fully managed NoSQL database for modern app development.

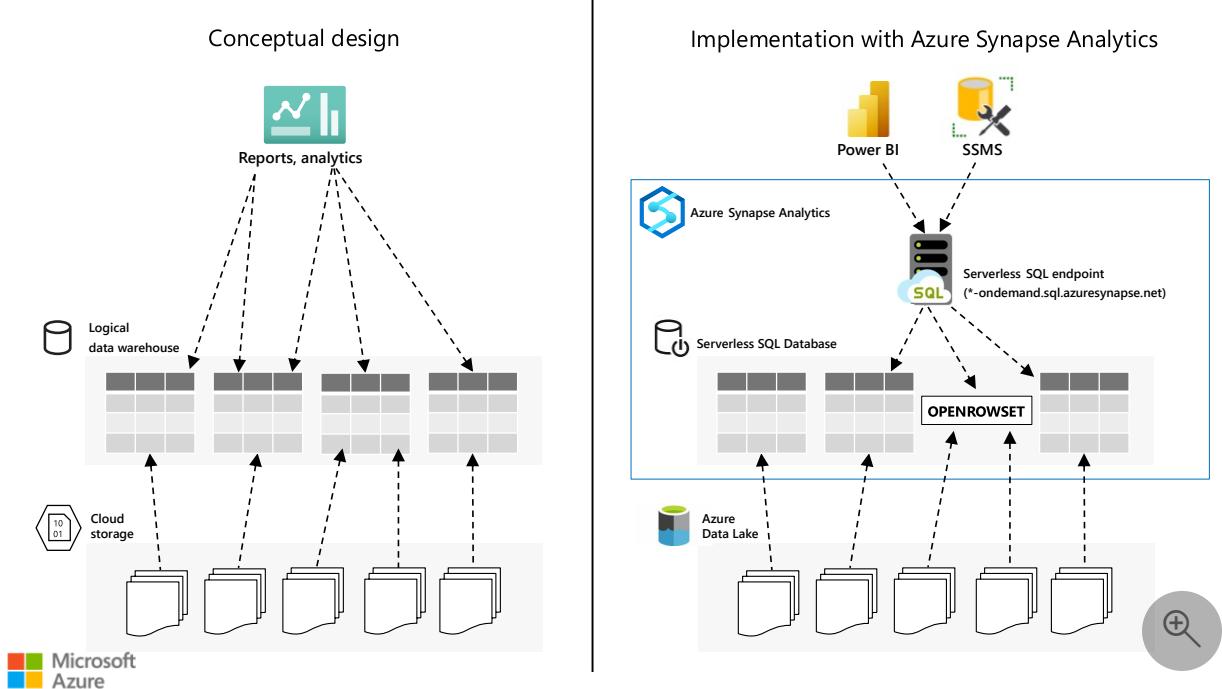
Scenario details

By using an LDW with Azure Synapse serverless SQL pools, you can join cold batch data, hot streaming data, and live transactional data in a single T-SQL query or view definition.

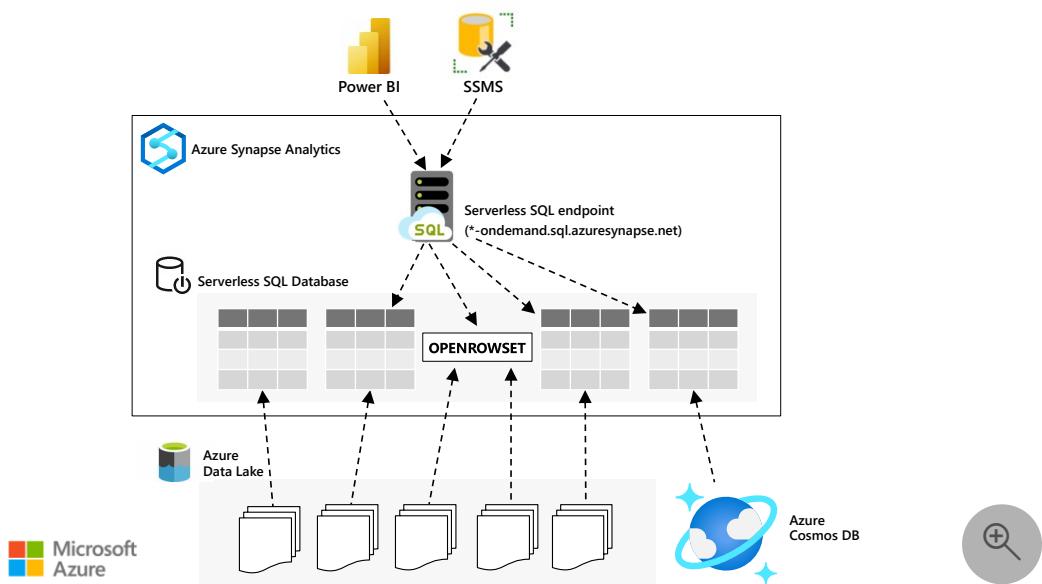
This solution avoids moving data through complex, expensive, and latency-prone extract, transform, and load (ETL) pipelines. The LDW concept is similar to a [data lakehouse](#), but LDW with Azure Synapse Analytics includes support for [hybrid transaction/analytical processing \(HTAP\)](#). HTAP uses Azure Synapse serverless SQL pools to query OLTP data that's stored in Azure Cosmos DB.

An Azure Synapse Analytics LDW is based on serverless SQL pools that are available with all Azure Synapse workspaces. An enhanced version of the [OPENROWSET](#) function enables serverless SQL pools to access data in Data Lake Storage.

This data access allows creation of relational database objects like tables and views over collections of data files that represent logical entities, like products, customers, and sales transactions. BI tools that connect by using a standard SQL Server endpoint can consume these logical entities as dimensions and fact tables.

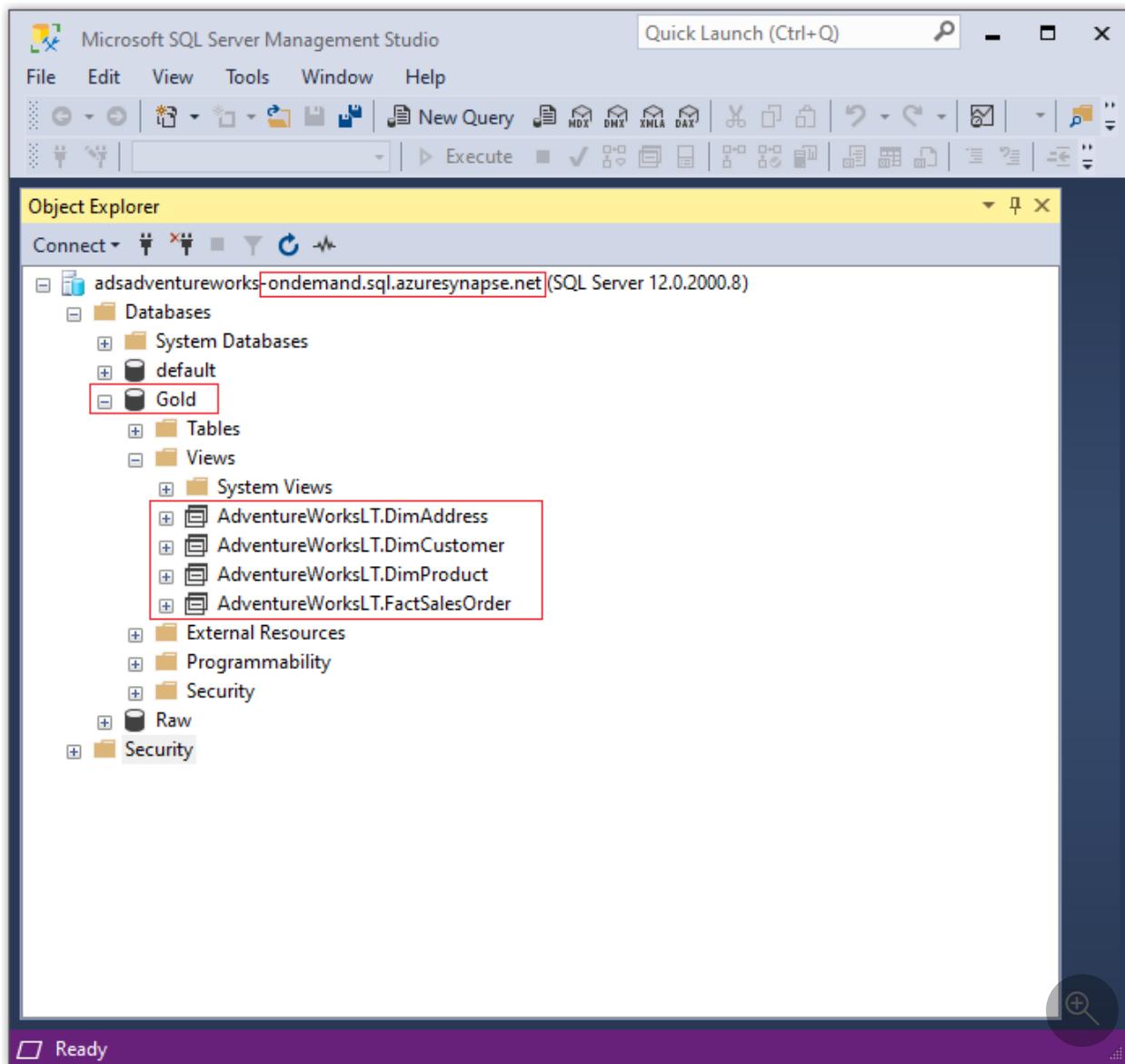


The ability to access transactional data stores like Azure Cosmos DB through the Azure Synapse Link for Azure Cosmos DB expands these capabilities. Accessing OLTP data by using HTAP architecture provides instant updates without interfering with live transactions.



Each Azure Synapse workspace includes an on-demand SQL endpoint. The endpoint lets SQL Server administrators and developers use familiar environments to work with LDWs that Azure Synapse serverless SQL pools define.

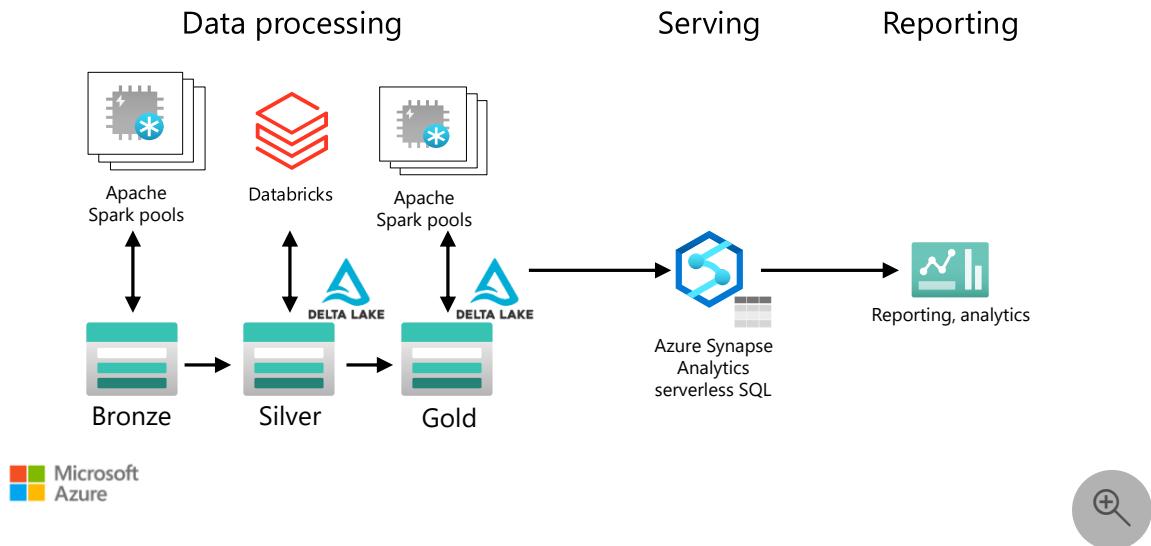
The following screenshot shows SQL Server Management Studio (SSMS) connected to an Azure Synapse serverless SQL pool.



Azure Synapse serverless SQL pools support the following file formats:

- Delimited text, such as CSV, TSV, and TXT
- JSON
- Parquet

Azure Synapse serverless SQL pools also support the [Delta Lake](#) format. This support allows patterns like *enrich in Spark, serve with SQL*, where Apache Spark™ services like [Azure Databricks](#) or [Apache Spark pools in Azure Synapse](#) engineer data to create curated datasets in the data lake. Instead of having to load these datasets into a physical data warehouse, you can define an LDW over the data lake to provide the model/serve layer for reporting.



The LDW with Azure Synapse serverless SQL pools is an implementation of the [Data Lakehouse](#) pattern. Using Databricks SQL to implement an LDW is an alternative solution. However, Databricks SQL lacks the HTAP capability of Azure Synapse Link for Cosmos DB.

Potential use cases

This pattern is useful for the following cases:

- Data warehouse serving layer for BI and other analytical use cases.
- Ad-hoc exploration of raw data in a data lake.
- Cost-effective data streaming into a data lake that doesn't require its own compute resources to write data. A logical database table, view, or ad-hoc T-SQL query can access the data instantly from the data lake.
- Instant access to Azure Cosmos DB transactional data to build real-time aggregation pipelines or join with analytical data stored in the data lake.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Jon Dobrzeniecki](#) | Sr. Cloud Solution Architect

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Query storage files with serverless SQL pool in Azure Synapse Analytics](#)
- [Tutorial: Create Logical Data Warehouse with serverless SQL pool](#)
- [What is Azure Synapse Link for Azure Cosmos DB?](#)
- [POLARIS: The distributed SQL engine in Azure Synapse](#)↗
- [What is Delta Lake?](#)

Related resources

- [Enterprise data warehouse](#)
- [Secure a data lakehouse with Azure Synapse Analytics](#)
- [Near real-time lakehouse data processing](#)
- [Query a data lake or lakehouse by using Azure Synapse serverless](#)

Manage data across Azure SQL estate with Microsoft Purview

Azure SQL Database

Microsoft Purview

Azure SQL Managed Instance

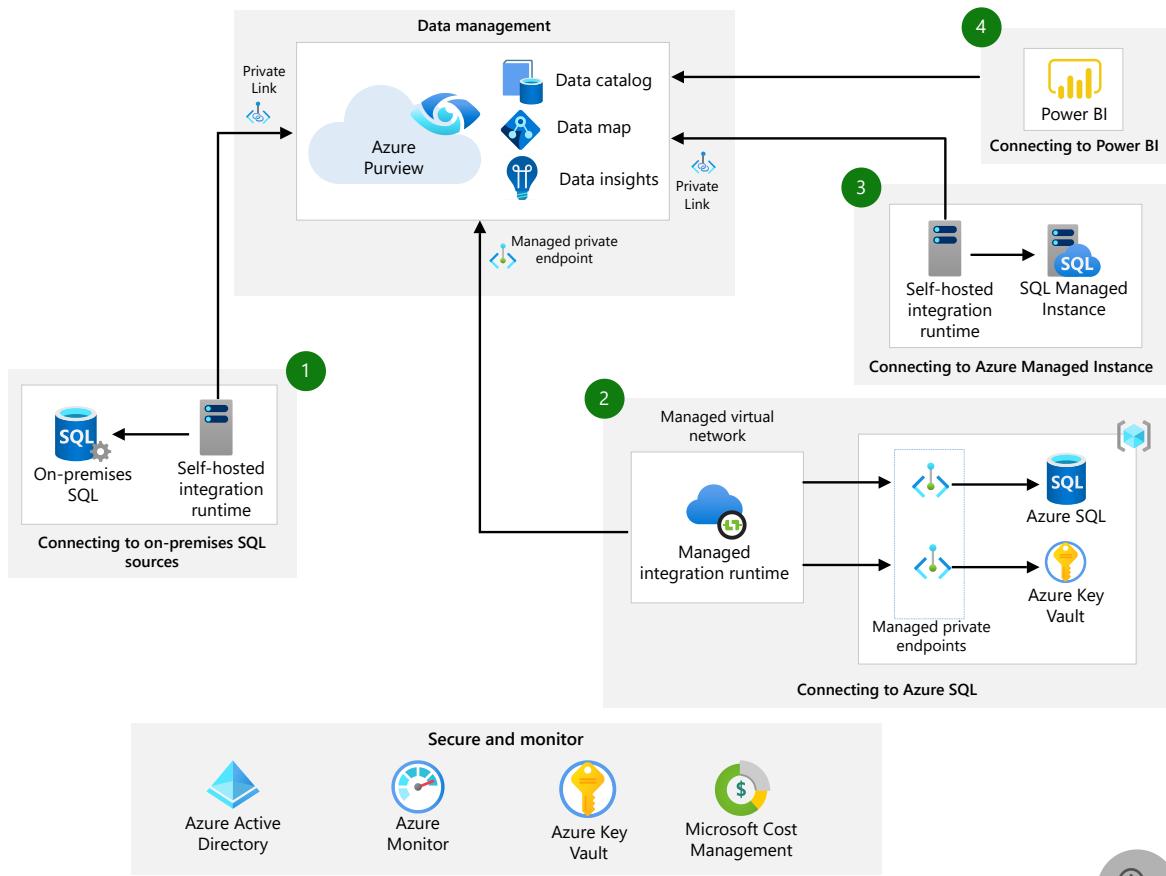
Power BI

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This article describes how you can improve your organization's governance process by using Azure Purview in your Azure SQL estate.

Architecture



Download a [Visio file](#) of this architecture.



Dataflow

The next four scenarios show the options available to you, to connect to Azure Purview securely.

1. Connect Azure Purview to **on-premises SQL** via Self-Hosted Integration Run time by Private Endpoint.
2. Connect Azure Purview to **Azure SQL** via Managed Virtual Network by Managed Private Endpoint.
3. Connect Azure Purview to **Azure SQL Managed Instance** via Self-Hosted Integration run time by Private Endpoint.
4. Connect Azure Purview natively to **Power BI**.

 **Note**

The information transferred from the sources to Azure Purview is metadata describing the data within the scanned sources. No actual data is transferred from the SQL sources to Azure Purview.

Capabilities

- **Catalog.** Azure Purview Data Catalog can automatically capture and describe core characteristics of data at the source. The characteristics include schema, technical properties, and location. Azure Purview glossary allows a business-friendly definition of data to be layered on top to improve search and discovery.
- **Classification.** Azure Purview automatically classifies datasets and data elements with over 100 predefined sensitive-data classifications. It also allows users to define their own custom classification schemes that can be applied manually and automatically.
- **Ownership.** Azure Purview allows data ownership and stewardship to be applied to data assets and glossary items within the catalog.
- **Insights.** Insights in Azure Purview provide multiple pre-defined reports to help CDOs, data professionals, and data governance professionals gain a detailed understanding of the data.

Components

The solution uses the following components:

- [Azure Purview](#) is a unified data catalog that manages on-premises, multicloud, and software as a service (SaaS) data. This data governance service maintains data landscape maps. Features include automated data discovery, sensitive-data classification, and data lineage.
- [Microsoft SQL Server](#) is a family of relational database management systems, or RDBMS. The servers are deployed and managed by your organization.
- [Azure SQL Database](#) is a fully managed SQL database built for the cloud with automatic updates, provisioning, scaling, and backups.
- [Azure SQL Managed Instance](#) is a cloud database service that provides all the features of SQL Server with added protection, connectivity, and automatic updates.
- [Power BI](#) is a collection of software services and apps. These services create and share reports that connect and visualize sources of data. When you use Power BI with Azure Purview, it can be cataloged, classified, and have granular lineage illustrated end to end.
- [Azure Private Link](#) provides private connectivity from a virtual network to Azure platform as a service (PaaS), services that you own, or Microsoft partner services.
- [Azure Key Vault](#) stores and controls access to secrets such as tokens, passwords, and API keys. Key Vault also creates and controls encryption keys and manages security certificates.
- [Microsoft Entra ID](#) offers cloud-based identity and access management services. These features provide a way for users to sign in and access resources.
- [Azure Monitor](#) collects and analyzes data on environments and Azure resources. This data includes app telemetry, such as performance metrics and activity logs.

Scenario details

As more of your organization's data is loaded into Azure, the need to properly govern and manage that data across all your data sources and data consumers grows.

Without high-quality data in your Azure SQL estate, the business value can be diminished. The solution is to build a foundation for data governance and management that can produce and deliver trustworthy high-quality data.

Data needs to be managed at scale across on-premises, cloud, and multicloud storage. This management ensures compliance requirements are met around security, privacy, and usage. Well-managed data can also improve self-discovery, data sharing, and quality—improving the use of data in applications and analytics.

With [Azure Purview](#), you can:

- Ensure that definitions, classifications, and governance processes are applied uniformly for your data.
- Provide a central platform where you can apply definitions and ownership to your data.
- With a single view on reports and insights, you can generate data standards that should be imposed on your data.
- Focus on governance to find, classify, define, and enforce policies and standards across data.

Azure Purview can automatically discover, catalog, classify, and manage data across Microsoft SQL offerings, whether on-premises or in Azure.

Potential use cases

The solution described here is appropriate for organizations that would benefit from the following outcomes of well-governed data:

- Automatic discovery of data in the organization to accelerate cloud adoption.
- Secure data for compliance with data laws and regulations.
- Improved access, discovery, and quality of managed data to enhance analytics.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Isabel Arevalo](#) | Senior Cloud Solution Architect

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Azure customer case studies](#)

- Microsoft Purview technical documentation and best practices
- What is Microsoft Purview?
- What is Power BI? ↗
- Microsoft Entra ID ↗
- Azure Cost Management and Billing ↗
- Azure Monitor ↗
- Key Vault ↗

Related resources

- Data governance with Profisee and Microsoft Purview
- Migrate master data services to Azure with CluedIn and Azure Purview

Mining equipment monitoring

Azure Analysis Services

Azure Logic Apps

Azure Data Lake Storage

Azure Databricks

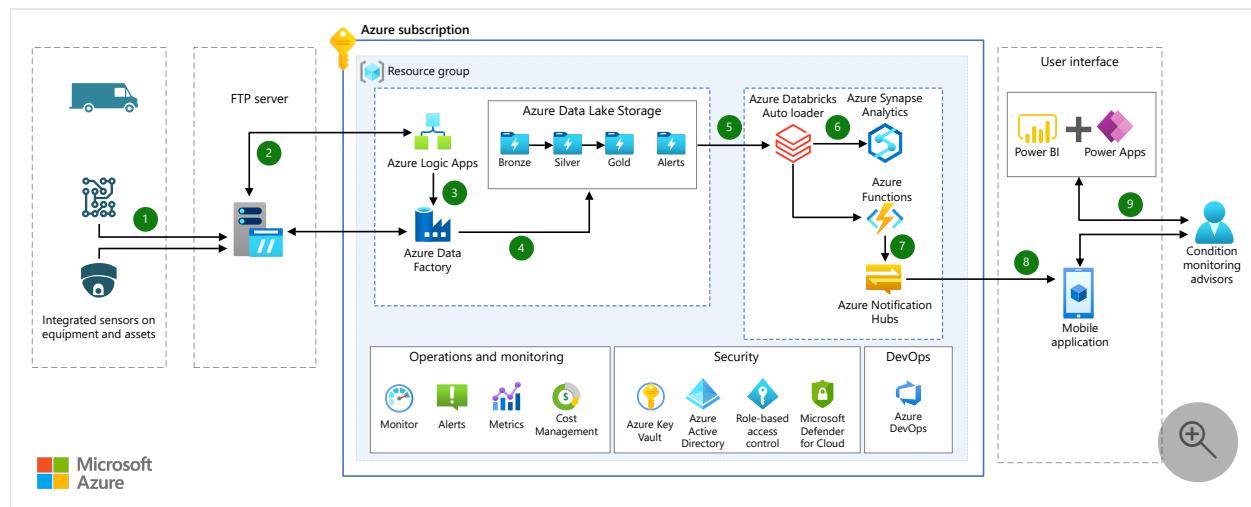
Azure Monitor

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

Mining companies can have Azure continually monitor the performance data from their equipment or from other assets. Analysis of the data identifies anomalies and results in recommendations for maintenance and repair. Such monitoring can prevent failures and reduce operating costs.

Architecture



[Download a Visio file](#) of this architecture.

Dataflow

The data flows through the solution as follows:

1. Equipment and other assets have integrated sensor systems that deliver sensor data (in CSV files) to a folder in an FTP server or to Azure Storage.
2. Azure Logic App monitors the folder for new or modified files.
3. Logic App triggers the Data Factory pipeline when a file is added to the folder, or when a previously added file is modified.

4. Azure Data Factory obtains the data from the FTP server or from Azure Storage, and stores it to a data lake that Azure Data Lake provides. The Delta Lake open-source software augments Data Lake capabilities.
5. The cloudFiles feature of Azure Databricks Auto Loader automatically processes new files as they arrive at the data lake, and can also process existing files.
 - a. cloudFiles uses structured streaming APIs to check if sensor values exceed thresholds. If so, it copies the values to a separate storage folder (Alerts).
 - b. After appropriate cleansing and transforming of the data, it moves the data to Delta Lake Bronze/Silver/Gold folders. The folders contain various transformations of the data; for example, ingested (Bronze), to refined (Silver), to aggregated (Gold).
6. An Azure Synapse connector in Azure Databricks moves the data from the data lake to an Azure Synapse Analytics dedicated SQL pool.
7. Whenever a new alert arrives in the Alerts folder, Azure Function Apps sends notifications to Azure Notification Hub.
8. Notification Hub then sends notifications to various mobile platforms to alert operators and administrators of events that require attention.
9. Monitoring advisors can create visual reports to explore the data. They can publish and share them, and collaborate with others. Power BI integrates with other tools, including Power Apps. Advisors can integrate Power BI reports into a Canvas App in Power Apps for a good user experience.

Components

Data is loaded from these different data sources using several Azure components:

- [Azure Data Lake Storage](#) makes Azure Storage the foundation for building enterprise data lakes on Azure. It can quickly process massive amounts of data (petabytes).
- [Azure Data Factory](#) is a managed service that orchestrates and automates data movement and data transformation. In this architecture, it copies the data from the source to Azure Storage.
- [Azure Logic Apps](#) are automated workflows for common enterprise orchestration tasks. Logic Apps includes [connectors](#) for many popular cloud services, on-premises products, and other applications.
- [Azure Databricks](#) is an Apache Spark-based analytics platform optimized for the Microsoft Azure cloud services platform. Databricks is integrated with Azure to provide one-click setup, streamlined workflows, and an interactive workspace that was designed in collaboration with the founders of Apache Spark.
- [Azure Databricks – Auto Loader](#) provides a structured streaming source called cloudFiles. The cloudFiles source automatically processes new files as they arrive at

a directory, and can also process other files in the directory.

- [Azure Synapse Analytics](#) is a distributed system for storing and analyzing large datasets. Its use of massive parallel processing (MPP) makes it suitable for running high-performance analytics.
- [Azure Functions](#) allows you to run small pieces of code (called "functions") without worrying about application infrastructure. Azure Functions is a great solution for processing bulk data, integrating systems, working with the internet-of-things (IoT), and building simple APIs and micro-services.
- [Power BI](#) is a suite of business analytics tools to analyze data and provide insights. Power BI can query a semantic model stored in Analysis Services, or it can query Azure Synapse directly.
- [Power Apps](#) is a suite of apps, services, and connectors for building custom business apps. It includes an underlying data platform ([Microsoft Dataverse](#)) and a rapid development environment.

Scenario details

Potential use cases

- Monitor mining equipment and other equipment that can provide the needed data. This solution is ideal for the energy industry.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Ansley Yeo](#) | Technology Leader and IoT

Next steps

- [Create, monitor, and manage FTP files by using Azure Logic Apps](#)
- [Copy data from FTP server by using Azure Data Factory](#)
- [Load files from Azure Blob storage and Azure Data Lake Storage Gen1 and Gen2 using Auto Loader](#)
- [Azure Synapse Analytics](#)
- [On GitHub: azure-notificationhubs-dotnet/Samples/AzFunctions/](#)
- [Azure SQL Data Warehouse with DirectQuery](#)

- [Power Apps visual for Power BI](#)

Information about the Delta Lake open-source project for building a Lakehouse architecture:

- [Delta Lake Key Features ↗](#)
- [What is Delta Lake](#)
- [Delta Lake and Delta Engine guide](#)

Related resources

See the following related database architectural guidance:

- [Non-relational data and NoSQL](#)
- [Big data architectures](#)
- [Choosing a batch processing technology in Azure](#)
- [Data lakes](#)
- [Choosing a big data storage technology in Azure](#)
- [Modernize mainframe & midrange data](#)
- [Master data management with Profisee and Azure Data Factory](#)
- [Master Data Management powered by CluedIn](#)
- [DataOps for the modern data warehouse](#)
- [Data warehousing and analytics](#)
- [Real Time Analytics on Big Data Architecture](#)

See the following related IoT architectural guidance:

- [IoT solutions conceptual overview](#)
- [Vision with Azure IoT Edge](#)
- [Azure Industrial IoT Analytics Guidance](#)
- [Azure IoT reference architecture](#)
- [IoT and data analytics](#)

Modern analytics architecture with Azure Databricks

Azure Data Factory

Azure Data Lake Storage

Azure Databricks

Azure Synapse Analytics

Power BI

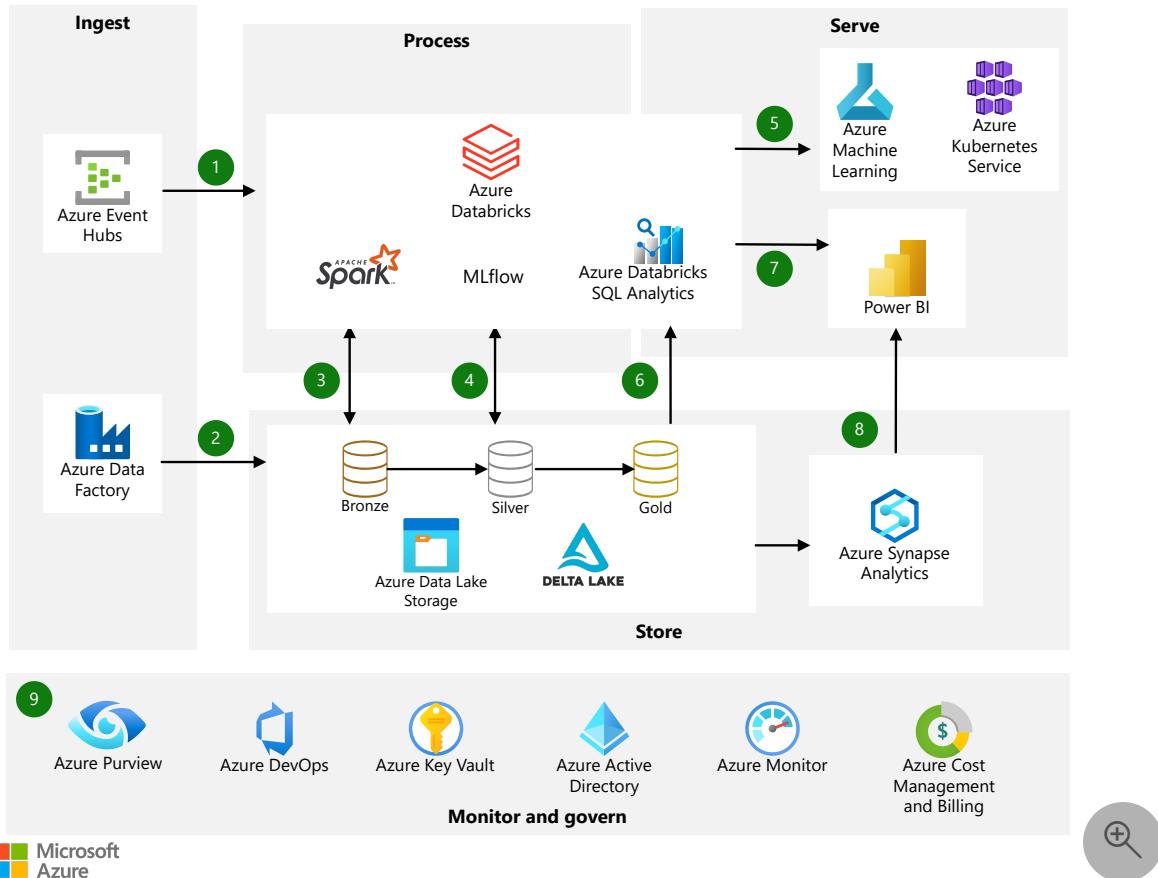
💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This solution outlines a modern data architecture. Azure Databricks forms the core of the solution. This platform works seamlessly with other services, such as Azure Data Lake Storage Gen2, Azure Data Factory, Azure Synapse Analytics, and Power BI.

Apache® and Apache Spark™ 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

1. Azure Databricks ingests raw streaming data from Azure Event Hubs.
2. Data Factory loads raw batch data into Data Lake Storage Gen2.
3. For data storage:
 - Data Lake Storage Gen2 houses data of all types, such as structured, unstructured, and semi-structured. It also stores batch and streaming data.
 - Delta Lake forms the curated layer of the data lake. It stores the refined data in an open-source format.
 - Azure Databricks works well with a [medallion architecture](#) that organizes data into layers:
 - Bronze: Holds raw data.
 - Silver: Contains cleaned, filtered data.
 - Gold: Stores aggregated data that's useful for business analytics.

4. The analytical platform ingests data from the disparate batch and streaming sources. Data scientists use this data for these tasks:

- Data preparation.
- Data exploration.
- Model preparation.
- Model training.

MLflow manages parameter, metric, and model tracking in data science code runs. The coding possibilities are flexible:

- Code can be in SQL, Python, R, and Scala.
- Code can use popular open-source libraries and frameworks such as Koalas, Pandas, and scikit-learn, which are pre-installed and optimized.
- Practitioners can optimize for performance and cost with single-node and multi-node compute options.

5. Machine learning models are available in several formats:

- Azure Databricks stores information about models in the [MLflow Model Registry](#). The registry makes models available through batch, streaming, and REST APIs.
- The solution can also deploy models to Azure Machine Learning web services or Azure Kubernetes Service (AKS).

6. Services that work with the data connect to a single underlying data source to ensure consistency. For instance, users can run SQL queries on the data lake with Azure Databricks SQL Analytics. This service:

- Provides a query editor and catalog, the query history, basic dashboarding, and alerting.
- Uses integrated security that includes row-level and column-level permissions.
- Uses a [Photon-powered Delta Engine to accelerate performance](#).

7. Power BI generates analytical and historical reports and dashboards from the unified data platform. This service uses these features when working with Azure Databricks:

- A [built-in Azure Databricks connector](#) for visualizing the underlying data.
- Optimized Java Database Connectivity (JDBC) and Open Database Connectivity (ODBC) drivers.

8. Users can export gold data sets out of the data lake into Azure Synapse via the optimized Synapse connector. SQL pools in Azure Synapse provide a data warehousing and compute environment.
9. The solution uses Azure services for collaboration, performance, reliability, governance, and security:
 - Microsoft Purview provides data discovery services, sensitive data classification, and governance insights across the data estate.
 - Azure DevOps offers continuous integration and continuous deployment (CI/CD) and other integrated version control features.
 - Azure Key Vault securely manages secrets, keys, and certificates.
 - Microsoft Entra ID provides single sign-on (SSO) for Azure Databricks users. Azure Databricks supports automated user provisioning with Microsoft Entra ID for these tasks:
 - Creating new users.
 - Assigning each user an access level.
 - Removing users and denying them access.
 - Azure Monitor collects and analyzes Azure resource telemetry. By proactively identifying problems, this service maximizes performance and reliability.
 - Azure Cost Management and Billing provide financial governance services for Azure workloads.

Components

The solution uses the following components.

Core components

- [Azure Databricks](#) is a data analytics platform. Its fully managed Spark clusters process large streams of data from multiple sources. Azure Databricks cleans and transforms structureless data sets. It combines the processed data with structured data from operational databases or data warehouses. Azure Databricks also trains and deploys scalable machine learning and deep learning models.
- [Event Hubs](#) is a big data streaming platform. As a platform as a service (PaaS), this event ingestion service is fully managed.

- [Data Factory](#) is a hybrid data integration service. You can use this fully managed, serverless solution to create, schedule, and orchestrate data transformation workflows.
- [Data Lake Storage Gen2](#) is a scalable and secure data lake for high-performance analytics workloads. This service can manage multiple petabytes of information while sustaining hundreds of gigabits of throughput. The data may be structured, semi-structured, or unstructured. It typically comes from multiple, heterogeneous sources like logs, files, and media.
- [Azure Databricks SQL Analytics](#) runs queries on data lakes. This service also visualizes data in dashboards.
- [Machine Learning](#) is a cloud-based environment that helps you build, deploy, and manage predictive analytics solutions. With these models, you can forecast behavior, outcomes, and trends.
- [AKS](#) is a highly available, secure, and fully managed Kubernetes service. AKS makes it easy to deploy and manage containerized applications.
- [Azure Synapse](#) is an analytics service for data warehouses and big data systems. This service integrates with Power BI, Machine Learning, and other Azure services.
- [Azure Synapse connectors](#) provide a way to access Azure Synapse from Azure Databricks. These connectors efficiently transfer large volumes of data between Azure Databricks clusters and Azure Synapse instances.
- [SQL pools](#) provide a data warehousing and compute environment in Azure Synapse. The pools are compatible with Azure Storage and Data Lake Storage Gen2.
- [Delta Lake](#) is a storage layer that uses an open file format. This layer runs on top of cloud storage such as Data Lake Storage Gen2. Delta Lake supports data versioning, rollback, and transactions for updating, deleting, and merging data.
- [MLflow](#) is an open-source platform for the machine learning lifecycle. Its components monitor machine learning models during training and running. MLflow also stores models and loads them in production.

Reporting and governing components

- [Power BI](#) is a collection of software services and apps. These services create and share reports that connect and visualize unrelated sources of data. Together with

Azure Databricks, Power BI can provide root cause determination and raw data analysis.

- [Microsoft Purview](#) manages on-premises, multicloud, and software as a service (SaaS) data. This governance service maintains data landscape maps. Features include automated data discovery, sensitive data classification, and data lineage.
- [Azure DevOps](#) is a DevOps orchestration platform. This SaaS provides tools and environments for building, deploying, and collaborating on applications.
- [Azure Key Vault](#) stores and controls access to secrets such as tokens, passwords, and API keys. Key Vault also creates and controls encryption keys and manages security certificates.
- [Microsoft Entra ID](#) offers cloud-based identity and access management services. These features provide a way for users to sign in and access resources.
- [Azure Monitor](#) collects and analyzes data on environments and Azure resources. This data includes app telemetry, such as performance metrics and activity logs.
- [Azure Cost Management and Billing](#) manage cloud spending. By using budgets and recommendations, this service organizes expenses and shows how to reduce costs.

Scenario details

Modern data architectures meet these criteria:

- Unify data, analytics, and AI workloads.
- Run efficiently and reliably at any scale.
- Provide insights through analytics dashboards, operational reports, or advanced analytics.

This solution outlines a modern data architecture that achieves these goals. Azure Databricks forms the core of the solution. This platform works seamlessly with other services. Together, these services provide a solution with these qualities:

- Simple: Unified analytics, data science, and machine learning simplify the data architecture.
- Open: The solution supports open-source code, open standards, and open frameworks. It also works with popular integrated development environments (IDEs), libraries, and programming languages. Through native connectors and APIs, the solution works with a broad range of other services, too.

- Collaborative: Data engineers, data scientists, and analysts work together with this solution. They can use collaborative notebooks, IDEs, dashboards, and other tools to access and analyze common underlying data.

Potential use cases

The system that Swiss Re Group built for its Property & Casualty Reinsurance division inspired this solution. Besides the insurance industry, any area that works with big data or machine learning can also benefit from this solution. Examples include:

- The energy sector
- Retail and e-commerce
- Banking and finance
- Medicine and healthcare

Next steps

- Swiss Re builds a digital payment platform by using Azure Databricks and Power BI.
- AGL achieves machine learning at scale with a standardized platform that uses Azure Databricks and Machine Learning.

Related resources

To learn about related solutions, see this information:

Related architecture guides

- [Monitor Azure Databricks with Azure Monitor](#)
- [Compare machine learning products from Microsoft](#)
- [Choose a natural language processing technology](#)
- [Choose a stream processing technology](#)

Related architectures

- [Stream processing with Azure Databricks](#)
- [Batch scoring of Spark models on Azure Databricks](#)
- [Observability patterns and metrics for performance tuning](#)
- [Build a real-time recommendation API on Azure](#)

Oil and gas tank level forecasting

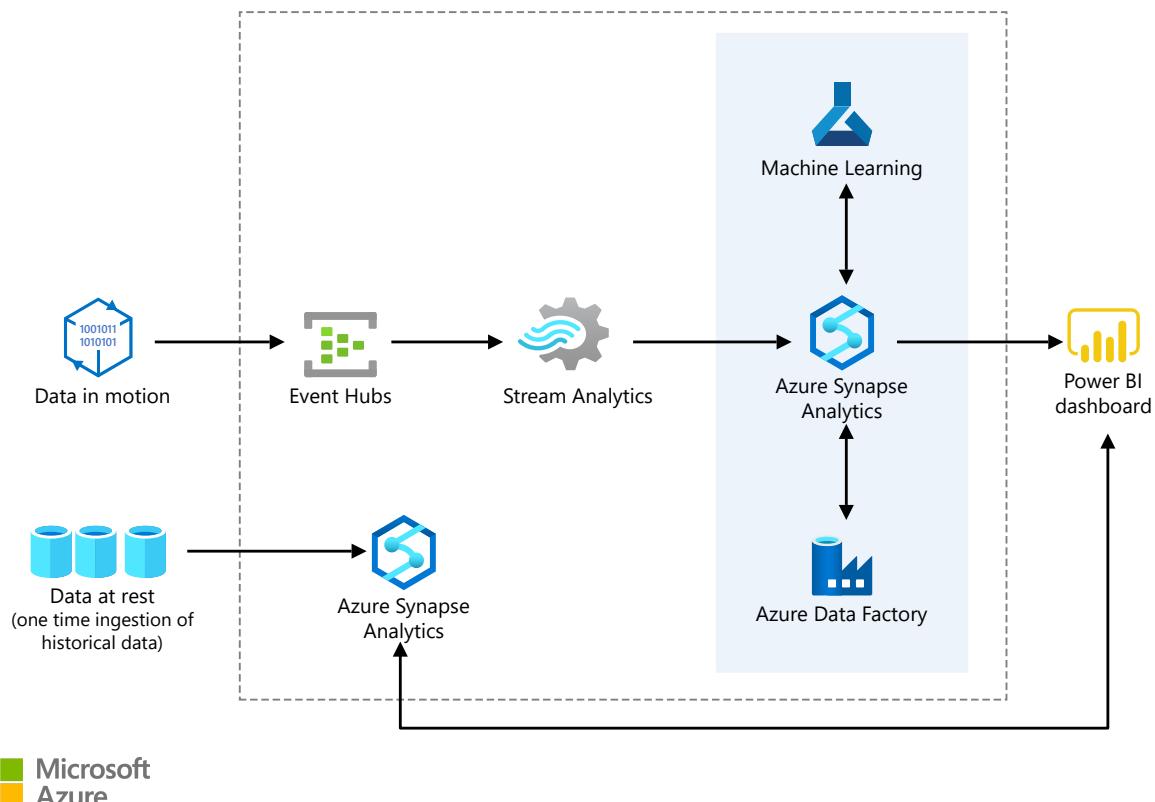
Azure Data Factory Azure Event Hubs Azure Machine Learning Azure Stream Analytics
Azure Synapse Analytics

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

Today, most facilities operate reactively to problems in tank levels. This reactivity often leads to spills, emergency shutdowns, expensive remediation costs, regulatory issues, costly repairs, and fines. Tank level forecasting helps manage and abate these and other problems.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

1. The data feeds into the [Azure Event Hubs](#) and [Azure Synapse Analytics](#) service as data points or events that will be used in the rest of the solution flow.
2. [Azure Stream Analytics](#) analyzes the data to provide near real-time analytics on the input stream from the event hub and directly publish to Power BI for visualization.
3. [Azure Machine Learning](#) is used to make forecast on the tank level of particular region given the inputs received.
4. Azure Synapse Analytics is used to store the prediction results received from Azure Machine Learning. These results are then consumed in the Power BI dashboard.
5. [Azure Data Factory](#) handles orchestration, and scheduling of the hourly model retraining.
6. Finally, [Power BI](#) is used for results visualization, so that users can monitor the tank level from a facility in real time and use the forecast level to prevent spillage.

Components

- [Azure Data Factory](#) ↗
- [Azure Event Hubs](#) ↗
- [Azure Machine Learning](#) ↗
- [Azure Stream Analytics](#) ↗
- [Azure Synapse Analytics](#) ↗
- [Power BI](#) ↗

Scenario details

The tank level forecasting process starts at the well input. Oil is measured as it comes into the facility via meters and is sent to tanks. Levels are monitored and recorded in tanks during the refining process. Oil, gas, and water output are recorded via sensors, meters, and records. Forecasts are then made using data from the facility; for example, forecasts can be made every 15 minutes.

Azure Cognitive Services is adaptable and can be customized to meet different requirements that facilities and corporations have.

Potential use cases

This solution is ideal for the energy, automotive, and aerospace industries.

Forecasts are created by harnessing the power of real-time and historical data readily available from sensors, meters, and records, which help with the following scenarios:

- Prevent tank spillage and emergency shutdowns
- Discover hardware malfunction or failure
- Schedule maintenance, shutdowns, and logistics
- Optimize operations and facility efficiency
- Detect pipeline leaks and slugging
- Reduce costs, fines, and downtime

Next steps

Product documentation:

- [What is Azure Event Hubs?](#)
- [What is Azure Synapse Analytics?](#)
- [Welcome to Azure Stream Analytics](#)
- [What is Azure Machine Learning?](#)
- [What is Azure Data Factory?](#)

Microsoft Learn modules:

- [Train a machine learning model with Azure Machine Learning](#)
- [Integrate data with Azure Data Factory or Azure Synapse Pipeline](#)

Related resources

- [Choosing a data analytics technology in Azure](#)
- [Stream processing with Azure Stream Analytics](#)
- [Anomaly Detector Process](#)
- [Demand Forecasting](#)
- [Predicting Length of Stay in Hospitals](#)
- [Predictive Aircraft Engine Monitoring](#)

Predictive aircraft engine monitoring

Azure Data Factory Azure Event Hubs Azure HDInsight Azure Machine Learning Azure Stream Analytics
Azure Monitor

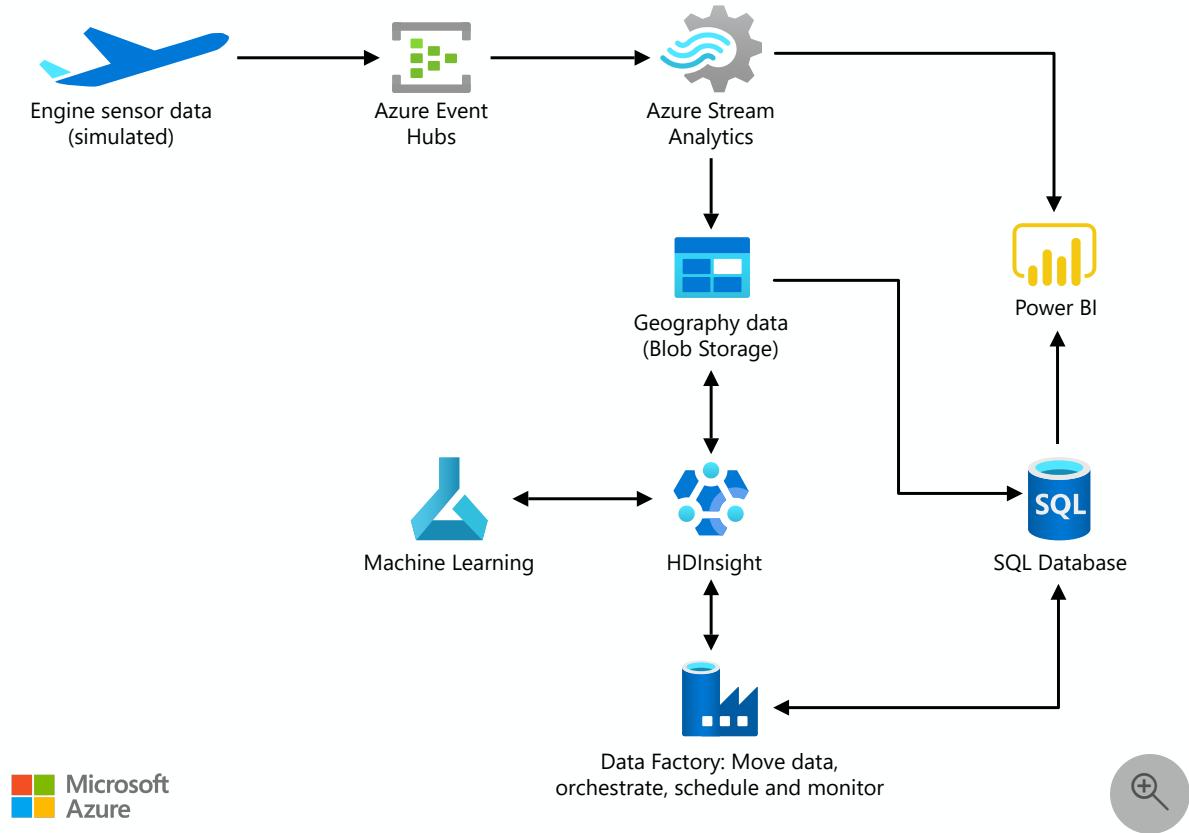
💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

Microsoft Azure's Predictive Maintenance solution demonstrates how to combine real-time aircraft data with analytics to monitor aircraft health.

This solution is built with [Azure Stream Analytics](#), [Event Hubs](#), [Azure Machine Learning](#), [HDInsight](#), [Azure SQL Database](#), [Data Factory](#), and [Power BI](#). These services run in a high-availability environment, patched and supported, allowing you to focus on your solution instead of the environment they run in.

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Stream Analytics](#) provides near real-time analytics on the input stream from Azure Event Hubs. Input data is filtered and passed to a Machine Learning endpoint, finally sending the results to the Power BI dashboard.
- [Event Hubs](#) ingests raw assembly-line data and passes it on to Stream Analytics.
- [Azure Machine Learning](#) predicts potential failures based on real-time assembly-line data from Stream Analytics.
- [HDInsight](#) runs Hive scripts to provide aggregations on the raw events that were archived by Stream Analytics.
- [Azure SQL Database](#) stores prediction results received from Machine Learning and publishes data to Power BI.
- [Data Factory](#) handles orchestration, scheduling, and monitoring of the batch processing pipeline.
- [Power BI](#) enables visualization of real-time assembly-line data from Stream Analytics and the predicted failures and alerts from Data Warehouse.

Scenario details

Potential use cases

This solution is ideal for the aircraft and aerospace industries.

With the right information, it's possible to determine the condition of equipment in order to predict when maintenance should be performed. Predictive maintenance can be used for the following items:

- Real-time diagnostics.
- Real-time flight assistance.
- Prognostics.
- Cost reduction.

Next steps

See product documentation:

- [Stream Analytics](#)
- [Event Hubs](#)
- [Azure Machine Learning](#)
- [HDInsight](#)
- [SQL Database](#)
- [Azure Data Factory](#)
- [Power BI ↗](#)

Related resources

Read other Azure Architecture Center articles about predictive maintenance and prediction with machine learning:

- [Predictive maintenance](#)
- [Predictive maintenance for industrial IoT](#)
- [Predict length of stay and patient flow](#)

Real-time analytics on big data architecture

Azure Analysis Services

Azure Event Hubs

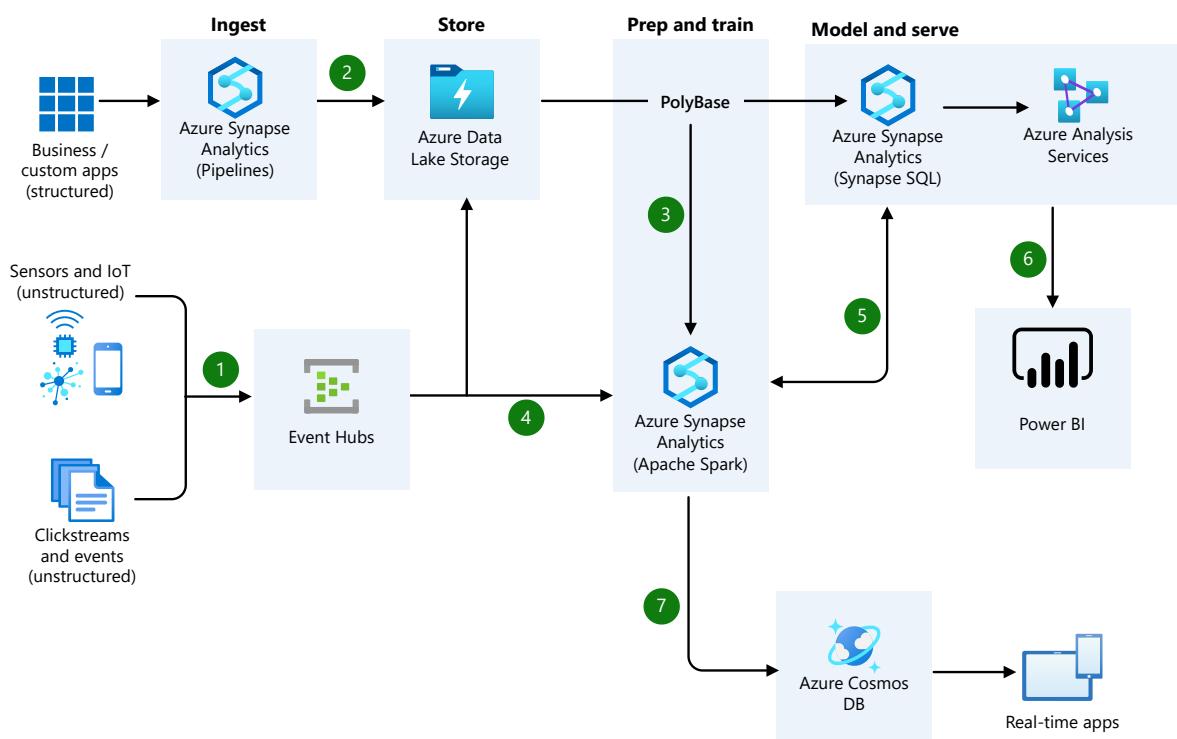
Azure Synapse Analytics

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This solution idea describes how you can get insights from live streaming data. Capture data continuously from any IoT device, or logs from website clickstreams, and process it in near-real time.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

1. Easily ingest live streaming data for an application, by using Azure Event Hubs.
2. Bring together all your structured data using Synapse Pipelines to Azure Blob Storage.
3. Take advantage of Apache Spark pools to clean, transform, and analyze the streaming data, and combine it with structured data from operational databases or data warehouses.
4. Use scalable machine learning/deep learning techniques, to derive deeper insights from this data, using Python, Scala, or .NET, with notebook experiences in Apache Spark pools.
5. Apply Apache Spark pool and Synapse Pipelines in Azure Synapse Analytics to access and move data at scale.
6. Build analytics dashboards and embedded reports in dedicated SQL pool to share insights within your organization and use Azure Analysis Services to serve this data to thousands of users.
7. Take the insights from Apache Spark pools to Azure Cosmos DB to make them accessible through real time apps.

Components

- [Azure Synapse Analytics](#) is the fast, flexible, and trusted cloud data warehouse that lets you scale, compute, and store elastically and independently, with a massively parallel processing architecture.
- [Synapse Pipelines Documentation](#) allows you to create, schedule, and orchestrate your ETL/ELT workflows.
- [Azure Data Lake Storage](#): Massively scalable, secure data lake functionality built on Azure Blob Storage
- [Azure Synapse Analytics Spark pools](#) is a fast, easy, and collaborative Apache Spark-based analytics platform.
- [Azure Event Hubs Documentation](#) is a big data streaming platform and event ingestion service.
- [Azure Cosmos DB](#) is a globally distributed, multi-model database service. Then learn how to replicate your data across any number of Azure regions and scale your throughput independent from your storage.
- [Azure Synapse Link for Azure Cosmos DB](#) enables you to run near real-time analytics over operational data in Azure Cosmos DB, **without any performance or cost impact on your transactional workload**, by using the two analytics engines available from your Azure Synapse workspace: [SQL Serverless](#) and [Spark Pools](#).

- [Azure Analysis Services](#) is an enterprise grade analytics as a service that lets you govern, deploy, test, and deliver your BI solution with confidence.
- [Power BI](#) is a suite of business analytics tools that deliver insights throughout your organization. Connect to hundreds of data sources, simplify data prep, and drive unplanned analysis. Produce beautiful reports, then publish them for your organization to consume on the web and across mobile devices.

Alternatives

- [Synapse Link](#) is the Microsoft preferred solution for analytics on top of Azure Cosmos DB data.
- [Azure IoT Hub](#) can be used instead of [Azure Event Hubs](#). IoT Hub is a managed service hosted in the cloud that acts as a central message hub for communication between an IoT application and its attached devices. You can connect millions of devices and their backend solutions reliably and securely. Almost any device can be connected to an IoT hub.

Scenario details

This scenario illustrates how you can get insights from live streaming data. You can capture data continuously from any IoT device, or logs from website clickstreams, and process it in near-real time.

Potential use cases

This solution is ideal for the media and entertainment industry. The scenario is for building analytics from live streaming data.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

You can use the [Azure pricing calculator](#) to get a customized pricing estimate.

Next steps

- [Azure Synapse Analytics Documentation](#)
- [Synapse Pipelines Documentation](#)
- [Azure Data Lake Storage documentation](#)
- [Azure Data Explorer](#)
- [Azure Synapse Analytics Spark pools](#)
- [Azure Event Hubs Documentation](#)
- [Azure Cosmos DB Documentation](#)
- [Analysis Services Documentation](#)
- [Power BI Documentation](#)

Related resources

- [Analytics end-to-end with Azure Synapse](#)
- [Geospatial analysis with Azure Synapse Analytics](#)
- [Big data analytics with enterprise-grade security using Azure Synapse](#)
- [High throughput stream ingestion to Azure Synapse](#)
- [Query a data lake or lakehouse by using Azure Synapse serverless](#)

Real-time analytics on data with Azure Service Bus and Azure Data Explorer

Azure Service Bus

Azure Data Explorer

Azure App Service

Azure SQL Database

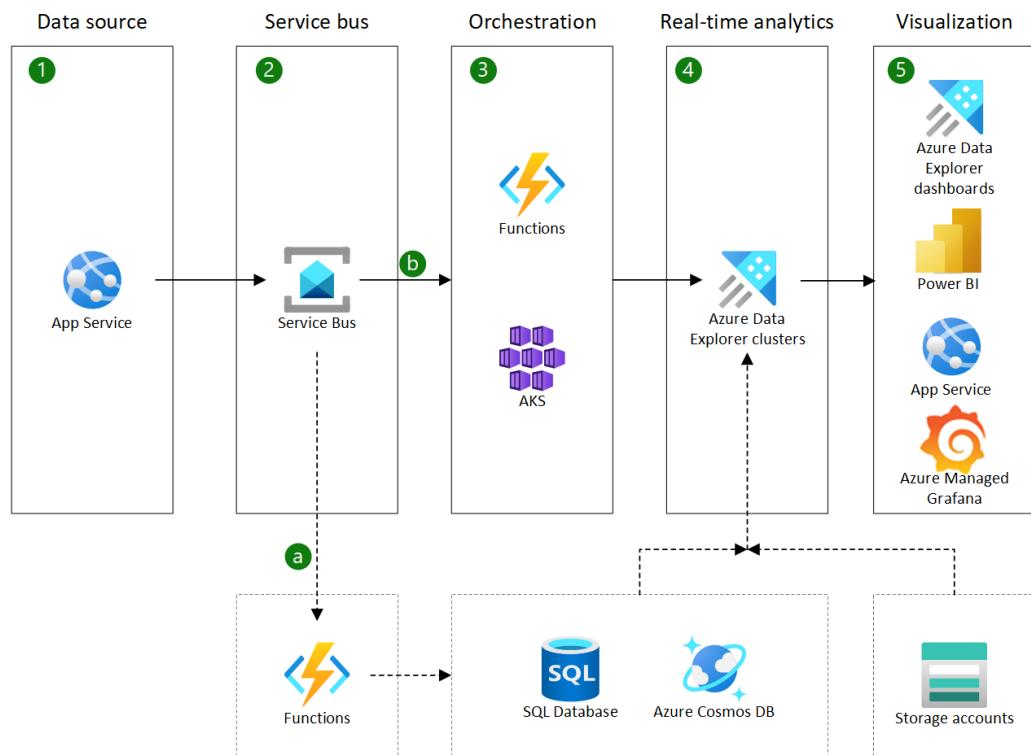
Azure Cosmos DB

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This article describes how to use Azure Data Explorer and Azure Service Bus to enhance an existing message broker architecture with near real-time analytics. It's intended for IT administrators, cloud architects, and operations and monitoring teams.

Architecture



 Microsoft Azure



Download a [Visio file](#) of this architecture.

The Grafana logo is a trademark of Raintank, Inc., dba Grafana Labs. No endorsement is implied by the use of this mark.

The diagram shows two data paths. The main path, which is represented by solid lines and boxes 1 through 5, is the ingestion of data from various sources into a service bus, where it's processed by a stream analytics job and stored in a SQL database. The second path, which is represented by dotted lines and boxes, shows the data flowing from the service bus to an Azure Data Explorer cluster, where it can be queried and analyzed via Kusto Query Language (KQL).

Service Bus is used to implement a [Queue-Based Load Leveling](#) pattern for a transactional application.

Azure Data Explorer is used to run analytics in near real-time and expose data via either APIs or direct queries to, for example, Power BI, Azure Managed Grafana, or Azure Data Explorer dashboards.

Dataflow

The data source in the architecture is an existing Online Transaction Processing (OLTP) application. Service Bus is used to asynchronously scale out the application.

1. The OLTP application (the data source), hosted in Azure App Service, sends data to Service Bus.
2. Data flows from Service Bus in two directions:
 - a. In the existing OLTP application flow, it triggers a function app to store data in Azure SQL Database, Azure Cosmos DB, or a similar operational database.
 - b. In the near real-time analytics flow, it triggers an orchestration flow.
3. The orchestration flow sends data to Azure Data Explorer for near real-time analytics. The flow can use either:
 - A function app that uses SDKs to send data in micro batches or that uses managed streaming ingestion support provided by Azure Data Explorer when it's [configured for streaming ingestion](#).
 - A polling service, like an application that's hosted on Azure Kubernetes Service (AKS) or an Azure VM, that sends data to Azure Data Explorer in micro batches. This option doesn't require configuring Azure Data Explorer streaming ingestion.

4. Azure Data Explorer processes the data, by using [schema mapping](#) and [update policies](#), and makes it available through an API, SDK, or connector for interactive analytics or reporting. Optionally, Azure Data Explorer can also ingest or reference data from other data sources, like SQL Database or Azure Data Lake Storage.
5. Applications, custom services, or reporting services like [Azure Data Explorer dashboards](#), Power BI, and Azure Managed Grafana can query the data in Azure Data Explorer in near real-time.

Components

- [App Service](#) ↗ enables you to build and host web apps, mobile back ends, and RESTful APIs in the programming language of your choice without managing infrastructure.
- [Service Bus](#) ↗ provides reliable cloud messaging as a service.
- [SQL Database](#) ↗ is a fully managed SQL database that's built for the cloud. It provides automatic updates, provisioning, scaling, and backups.
- [Azure Cosmos DB](#) ↗ is a globally distributed, multimodel database for applications of any scale.
- [Azure Functions](#) ↗ is an event-driven serverless compute platform. With Functions, you can deploy and operate at scale in the cloud and use triggers and bindings to integrate services.
- [AKS](#) ↗ is a highly available, highly secure, and fully managed Kubernetes service for application and microservices workloads.
- [Azure Data Explorer](#) ↗ is a fast, fully managed, and highly scalable data analytics service for real-time analysis of large volumes of data that streams from applications, websites, IoT devices, and more.
- [Data Lake Storage](#) ↗, built on Azure Blob Storage, provides massively scalable data lake functionality.
- [Power BI](#) ↗ can help you turn your data into coherent, visually immersive, interactive insights.
- [Azure Managed Grafana](#) ↗ is a fully managed service that enables you to deploy Grafana without spending time on configuration.

Scenario details

Real-time analytics is the process of analyzing data as soon as it's generated to get insights into the current state of the system. Organizations are increasingly adopting real-time analytics to gain a competitive edge. Near real-time analytics is a variant of real-time analytics that provides insights within seconds or minutes of data generation.

These processes enable organizations to gain insights faster, make better decisions, and respond to changing conditions more effectively. Near real-time analytics can be applied to various domains, like e-commerce, healthcare, manufacturing, and finance. For example, an e-commerce company can use near real-time analytics to monitor customer behavior, optimize pricing, and personalize recommendations.

Many organizations implement near real-time analytics in existing solutions. This solution idea demonstrates how to add near real-time analytics to an existing architecture that's based on a message broker and that's part of an operational OLTP application.

OLTP stands for Online Transaction Processing. It's a type of data processing that manages transaction-oriented applications, typically for data entry and retrieval transactions in a real-time environment. OLTP systems are designed to process small, fast transactions that are frequently financial in nature, like bank transactions or credit card purchases.

Potential use cases

Here are some use cases that illustrate the benefits of near real-time analytics:

- Healthcare providers can track patient outcomes, detect anomalies, and improve quality of care.
- Manufacturing companies can optimize production, reduce waste, and prevent downtime.
- Financial institutions can monitor transactions, detect fraud, manage risk, and ensure compliance with regulations.
- Commerce companies can monitor campaigns and gain insights to support promotion.
- Companies can monitor, optimize, analyze, and forecast supply chains.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Shlomo Sagiv](#) | Senior Content Developer

Other contributor:

- [Mick Alberts](#) | Technical Writer

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Azure Service Bus samples](#)
- [Azure Data Explorer data ingestion samples ↗](#)

Related resources

- [Near real-time lakehouse data processing](#)
- [Real-time analytics on big data architecture](#)
- [Create personalized marketing solutions in near real time](#)

Tier applications and data for analytics

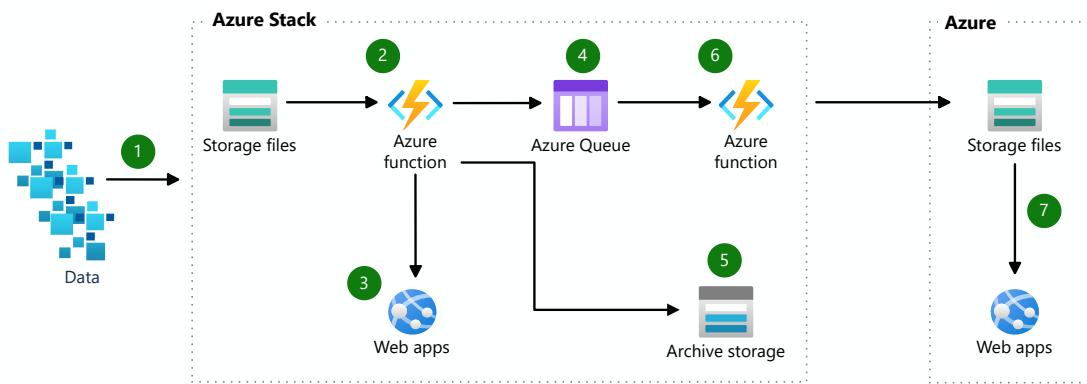
Azure Functions Azure Stack Azure Storage Azure App Service

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This solution idea describes how to tier data and applications on-premises and on Azure. As data flows into a storage account, you can use Azure Stack to analyze the data for anomalies or compliance and to display insights in apps.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

1. Data flows into a storage account.
2. Function on Azure Stack analyzes the data for anomalies or compliance.
3. Locally relevant insights are displayed on the Azure Stack app.
4. Insights and anomalies are placed into a queue.

5. The bulk of the data is placed into an archive storage account.
6. Function sends data from queue to Azure Storage.
7. Globally relevant and compliant insights are available in the global app.

Components

- [Storage](#): Durable, highly available, and massively scalable cloud storage
- [Azure Functions](#): Process events with serverless code
- [Azure Stack](#): Build and run innovative hybrid applications across cloud boundaries

Scenario details

This scenario can help you tier data and applications on-premises and on Azure. Filter unnecessary data early in the process, bring cloud applications close to the data on-premises, and analyze large-scale aggregate data from multiple locations on Azure.

Potential use cases

Tiered applications provide the following benefits:

- The ability to update the technology stack of one tier without affecting other areas of the application.
- Development teams work on their own areas of expertise.
- Able to scale the application.
- Adds reliability and more independence of the underlying servers or services.

Next steps

- [Storage documentation](#)
- [Azure Functions documentation](#)
- [Azure Stack documentation](#)

Related resources

- [Analytics architecture design](#)
- [Data analysis workloads for regulated industries](#)
- [Tiered data for analytics](#)

Multiparty computing architecture design

Azure Blob Storage

Azure Kubernetes Service (AKS)

Azure SQL Database

Multiparty computing or privacy-preserving computation allows parties in a business relationship to share data, do computations, and arrive at a mutual result without divulging their private data. Azure services can help you build a multiparty computing solution. The solution can include cloud-based and on-premises resources.

Multiparty computing has the following attributes:

- More than one company or organization is involved.
- The parties are independent.
- The parties don't trust one another with all their data.
- All parties access a common computing and data storage platform.
- Some processes must be private for some of the parties involved.

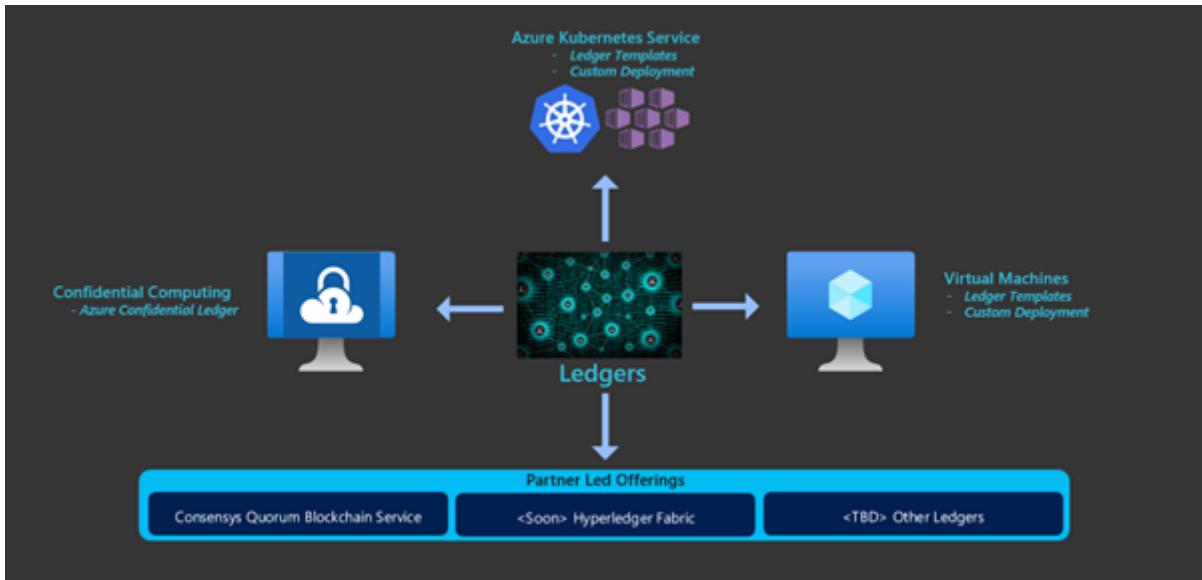
A supply chain is an example of a workflow that involves multiple parties. Raw materials flow from the point of origin to manufacturing. Goods from the manufacturer go through shipping partners to a distribution hub. From the hub, goods are sent to retail outlets.



This process has companies working together. These parties include the raw materials supplier, the manufacturer, shipping firms, warehouse operators, and the retail outlets. The product changes hands several times during the supply chain. Different parties need to track the product at all stages.

Multiparty computing technologies

Multiparty computing includes different technologies that enable parties to transact securely over a network.



One option is distributed ledgers. Blockchain is an example. Blockchain is a data ledger that can be shared between independent parties where all parties trust the data on the ledger. Transactions are collected in blocks with each block linking to the previous block. Some distributed ledgers don't use blocks. Each transaction can be linked to the previous transaction on the ledger.

Another possibility for multiparty computing uses hardware protected memory on the CPU itself. These regions, called *secure enclaves*, are cryptographically protected. This approach means that even a privileged administrator having full access to the server can't look at the process or the data inside those secure enclaves.

Since secure enclaves have the capability to remotely attest themselves to other enclaves, you can design a multiple organization network where the system runs from the enclaves. This approach is called the *trusted execution environment*.

Azure offers a managed service called Azure Confidential Ledger that lets you run a blockchain model on secure enclaves.

Finally, you could choose a centralized system, which offers immutability and trustworthiness. Azure SQL Database ledger offers the trust needed for multiparty computing in a relational database. You may not need a decentralized consensus, but just the immutability aspect of the ledger.

Blockchain network models

To decide if blockchain is a good fit for a business process, consider these questions:

- Does this business process cross trust boundaries?
- Do multiple parties share and update data?
- Are there any intermediaries that control the single source of truth?

- Does the process involve low-value manual verification steps?

If the answers to these questions are *yes*, the business process is a good candidate for a blockchain-based approach. Even if some answers are *no*, blockchain might still make sense. Look closely at the other multiparty computing options before deciding.

There are various kinds of blockchain networks to address your business needs. One characteristic is the criteria for taking part in the network. If the network is open for all, it's called a *public blockchain network*. You just download the client and join. Most cryptocurrencies work this way.

An alternative is a *permissioned blockchain network*, where you need permission from the existing members of the network to join. This model works for enterprises that deal with known organizations. For instance, a superstore may want to have a closed and permissioned blockchain network for its supply chain participants.

A business process might require only tamper-proof or tamper-evident data, which wouldn't require blockchain. If your process can run centrally, or all parties trust one another with the data, blockchain might also be unnecessary.

Azure multiparty computing

This section describes multiparty computing options available using Azure services.

Blockchain with Azure Virtual Machines

You can run ledger software using Azure Virtual Machines. Create as many virtual machines as you need and connect them in a blockchain network.

Deploying your own virtual machines allows you to customize your solution. The approach includes management overhead, such as updates, high availability, and business continuity requirements. You may have multiple organizations and multiple cloud accounts. Connecting the individual nodes can be complicated.

There are deployment templates available on Azure for most blockchain ledgers for virtual machines.

Blockchain on Kubernetes

Since most blockchain ledgers support deploying into Docker containers, you can use Kubernetes to manage the containers. Azure has a managed Kubernetes offering called

Azure Kubernetes Service (AKS) that you can use to deploy and configure your blockchain nodes.

AKS implementations come with a managed service for the virtual machines that power the AKS cluster. However, your organization must still manage your AKS clusters and any networking or storage options in your architecture.

There are deployment templates available on Azure for most blockchain ledgers for AKS.

Blockchain as a service

Azure supports third-party services that run ledger software on Azure. The service provider manages the infrastructure. They handle maintenance and updates. High availability and consortium management are included in the service.

ConsenSys offers Quorum on Azure. Quorum is an open-source protocol layer that supports Ethereum-based applications.

There may be other offerings in the future.

Azure Confidential Ledger

Azure Confidential Ledger is a managed service built on the Confidential Consortium Framework. It implements a permissioned blockchain network of nodes within Azure confidential computing. Confidential Ledger builds on existing encryption.

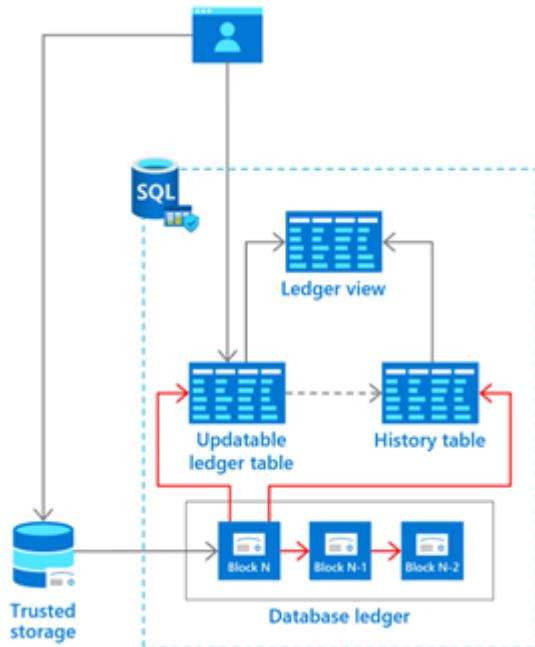
- Existing encryption
 - **Data at rest.** Encrypt inactive data when stored in blob storage or a database.
 - **Data in transit.** Encrypt data that's flowing between public or private networks.
- Confidential computing
 - **Data in use.** Encrypt data that's in use, while in memory and during computation.

Confidential computing allows encryption of data in the main memory. Confidential computing lets you process data from multiple sources without exposing the input data to other parties. This type of secure computation supports multiparty computing scenarios where data protection is mandatory in every step, such as money laundering detection, fraud detection, and secure analysis of healthcare data.

Data stored in Confidential Ledger is immutable and tamper-proof in the append-only ledger. The ledger is also independently verifiable. Confidential Ledger uses secure enclaves for a decentralized blockchain network and requires a minimal trusted computing base.

Azure SQL Database ledger

Azure SQL Database ledger allows participants to verify the data integrity of centrally-housed data without the network consensus of a blockchain network. For some centralized solutions trust is important, but decentralized infrastructure isn't necessary. This approach avoids complexity and performance implications of such an infrastructure.



ⓘ Note

Azure SQL Database ledger is currently in public preview.

Ledger provides tamper-evidence capabilities for your database. These capabilities allow you to cryptographically attest that your data hasn't been tampered with.

Ledger helps protect data from any attacker or high-privileged user, including database, system, and cloud administrators. Historical data is preserved. If a row is updated in the database, its previous value is maintained in a history table. This offers protection without any application changes.

Ledger is a feature of Azure SQL Database. It can be enabled in any existing Azure SQL Database.

Comparing options

Confidential Ledger and Azure SQL Database ledger

This table compares Confidential Ledger with Azure SQL Database ledger.

[Expand table](#)

	SQL Database ledger	Confidential Ledger
Centralized system that requires tamper evidence	Yes	No
Decentralized system that requires data to be tamper proof	No	Yes
Protects relational data from tampering	Yes	No
Protects unstructured data from tampering	No	Yes
Secure off-chain store of chain data in a blockchain	Yes	No
Secure off-chain store for files referenced to from a blockchain	No	Yes
Relational data is queryable	Yes	No
Unstructured stored data is queryable	No	Yes

Confidential Ledger and Azure Blob Storage

The immutable storage feature of Azure Blob Storage ensures that data written to it can be read but never changed. This table compares that technology with Confidential Ledger.

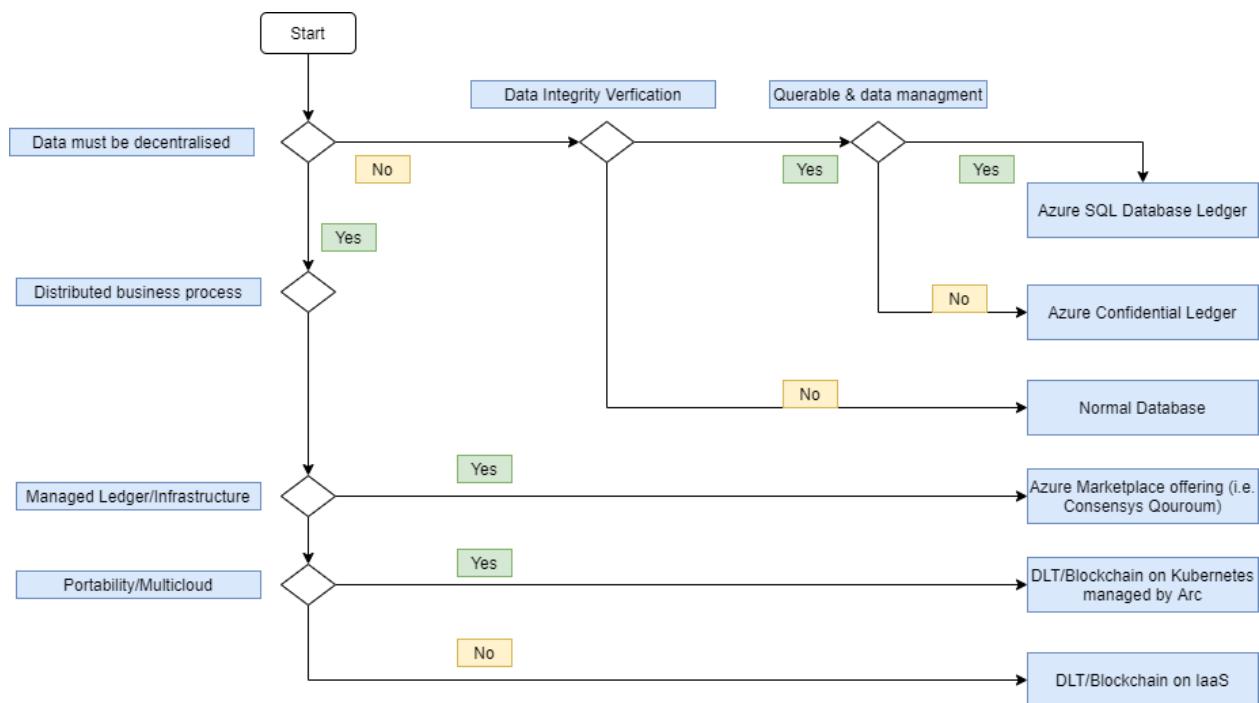
[Expand table](#)

	Confidential Ledger	Immutable storage
Confidential hardware enclaves	Yes	No
Append-only data integrity	Yes	Yes, limited to intervals

	Confidential Ledger	Immutable storage
In-use data encryption	Yes	No
Blockchain ledger proof	Yes	No

Multiparty computing decision

This diagram summarizes options for the multiparty computing with Azure services.



- Multi-cloud blockchain DLT
- Decentralized trust between banks
- Authenticating Azure confidential ledger nodes
- Azure Confidential Ledger Architecture

Microsoft Azure confidential ledger

Article • 10/12/2023

Microsoft Azure confidential ledger (ACL) is a new and highly secure service for managing sensitive data records. It runs exclusively on hardware-backed secure enclaves, a heavily monitored and isolated runtime environment which keeps potential attacks at bay. Furthermore, Azure confidential ledger runs on a minimalistic Trusted Computing Base (TCB), which ensures that no one—not even Microsoft—is "above" the ledger.

As its name suggests, Azure confidential ledger utilizes the [Azure Confidential Computing platform](#) and the [Confidential Consortium Framework](#) to provide a high integrity solution that is tamper-protected and evident. One ledger spans across three or more identical instances, each of which run in a dedicated, fully attested hardware-backed enclave. The ledger's integrity is maintained through a consensus-based blockchain.

Azure confidential ledger offers unique data integrity advantages, including immutability, tamper-proofing, and append-only operations. These features, which ensure that all records are kept intact, are ideal when critical metadata records must not be modified, such as for regulatory compliance and archival purposes.

Here are a few examples of things you can store on your ledger:

- Records relating to your business transactions (for example, money transfers or confidential document edits).
- Updates to trusted assets (for example, core applications or contracts).
- Administrative and control changes (for example, granting access permissions).
- Operational IT and security events (for example, Microsoft Defender for Cloud alerts).

For more information, you can watch the [Azure confidential ledger demo](#).

Key Features

The confidential ledger is exposed through REST APIs which can be integrated into new or existing applications. The confidential ledger can be managed by administrators utilizing Administrative APIs (Control Plane). It can also be called directly by application code through Functional APIs (Data Plane). The Administrative APIs support basic operations such as create, update, get and, delete. The Functional APIs allow direct

interaction with your instantiated ledger and include operations such as put and get data.

Ledger security

The ledger APIs support certificate-based authentication process with owner roles as well as Microsoft Entra ID based authentication and also role-based access (for example, owner, reader, and contributor).

The data to the ledger is sent through TLS 1.3 connection and the TLS 1.3 connection terminates inside the hardware backed security enclaves (Intel® SGX enclaves). This ensures that no one can intercept the connection between a customer's client and the confidential ledger server nodes.

Ledger storage

Confidential ledgers are created as blocks in blob storage containers belonging to an Azure Storage account. Transaction data can either be stored encrypted or in plaintext depending on your needs.

The confidential ledger can be managed by administrators utilizing Administrative APIs (Control Plane), and can be called directly by your application code through Functional APIs (Data Plane). The Administrative APIs support basic operations such as create, update, get and, delete.

The Functional APIs allow direct interaction with your instantiated confidential ledger and include operations such as put and get data.

Constraints

- Once a confidential ledger is created, you cannot change the ledger type (private or public).
- Azure confidential ledger deletion leads to a "hard delete", so your data will not be recoverable after deletion.
- Azure confidential ledger names must be globally unique. Ledgers with the same name, irrespective of their type, are not allowed.

Terminology

Term	Definition
ACL	Azure confidential ledger
Ledger	An immutable append-only record of transactions (also known as a Blockchain)
Commit	A confirmation that a transaction has been appended to the ledger.
Receipt	Proof that the transaction was processed by the ledger.

Next steps

- [Microsoft Azure confidential ledger architecture](#)
- [Quickstart: Azure portal](#)
- [Quickstart: Python](#)
- [Quickstart: Azure Resource Manager \(ARM\) template](#)

Architecture

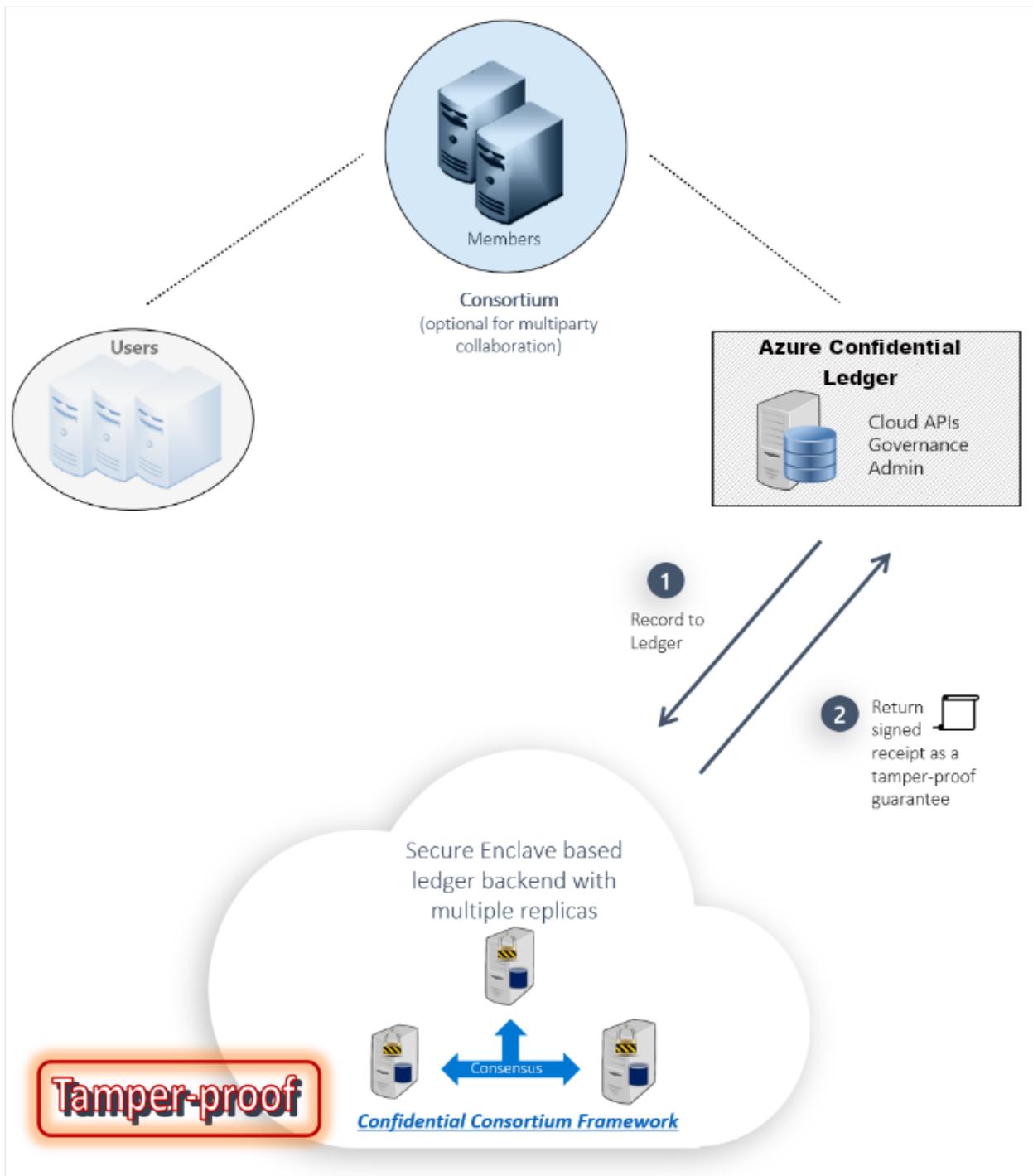
Article • 11/14/2022

The Azure confidential ledger, a REST API service, allows users to interact with the ledger through administrative and functional API calls. When data is recorded to the ledger, it is sent to the permissioned blockchain nodes that are secure enclaved backed replicas. The replicas follow a consensus concept. A user can also retrieve receipts for the data that has been committed to the ledger.

There is also an optional consortium notion that will support multi-party collaboration in the future.

Architecture diagram

This image provides an architectural overview of Azure confidential ledger, and shows Azure confidential ledger Users interacting with the Cloud APIs for a created ledger.



Next steps

- Overview of Microsoft Azure confidential ledger
- Authenticating Azure confidential ledger nodes
- Azure Confidential Ledger write transaction receipts

Authenticating Azure confidential ledger nodes

Article • 11/14/2022

Azure confidential ledger nodes can be authenticated by code samples and by users.

Code samples

When initializing, code samples get the node certificate by querying the Identity Service. After retrieving the node certificate, a code sample will query the ledger to get a quote, which is then validated using the Host Verify binaries. If the verification succeeds, the code sample proceeds to ledger operations.

Users

Users can validate the authenticity of Azure confidential ledger nodes to confirm they are indeed interfacing with their ledger's enclave. You can build trust in Azure confidential ledger nodes in a few ways, which can be stacked on one another to increase the overall level of confidence. As such, steps 1 and 2 are important confidence building mechanisms for users of Azure confidential ledger enclave as part of the initial TLS handshake and authentication within functional workflows. Beyond that, a persistent client connection is maintained between the user's client and the confidential ledger.

- 1. Validating a confidential ledger node:** This is accomplished by querying the identity service hosted by Microsoft, which provides a service certificate and thus helps verify that the ledger node is presenting a certificate endorsed/signed by the service certificate for that specific instance. Using PKI-based HTTPS, a server's certificate is signed by a well-known Certificate Authority (CA) or intermediate CA. In the case of Azure confidential ledger, the CA certificate is returned by the Identity Service in the form of the service certificate. If this node certificate isn't signed by the returned service certificate, the client connection should fail (as implemented in the sample code).
- 2. Validating a confidential ledger enclave:** A confidential ledger runs in an Intel® SGX enclave that's represented by a remote attestation report (or quote), a data blob generated inside that enclave. It can be used by any other entity to verify that the quote has been produced from an application running with Intel® SGX protections. The quote contains claims that help identify various properties of the enclave and the application that it's running. In particular, it contains the SHA-256

hash of the public key contained in the confidential ledger node's certificate. The quote of a confidential ledger node can be retrieved by calling a functional workflow API. The retrieved quote can then be validated following the steps described [here](#) ↴.

Next steps

- [Overview of Microsoft Azure confidential ledger](#)
- [Azure confidential ledger architecture](#)

What is the database ledger?

Article • 05/23/2023

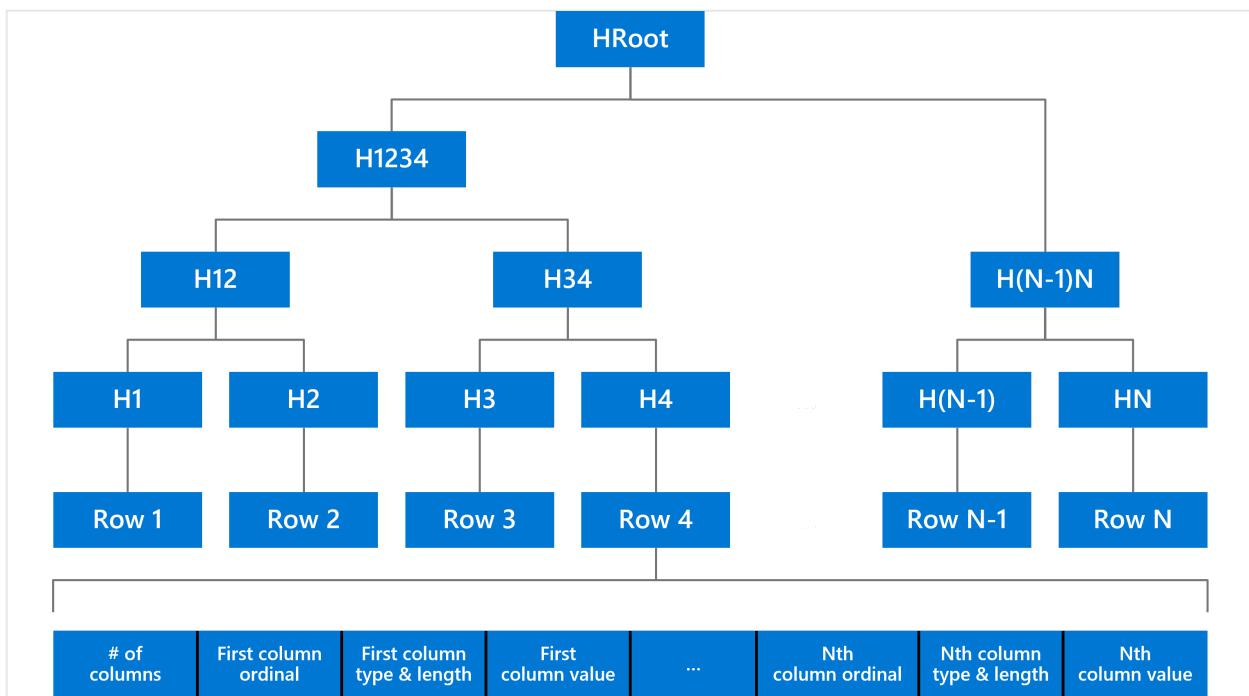
Applies to:  SQL Server 2022 (16.x)  Azure SQL Database  Azure SQL Managed Instance

The database ledger is part of the ledger feature. The database ledger incrementally captures the state of a database as the database evolves over time, while updates occur on ledger tables. It logically uses a blockchain and [Merkle tree data structures](#).

Any operations that update a ledger table need to perform some additional tasks to maintain the historical data and compute the digests captured in the database ledger. Specifically, for every row updated, we must:

- Persist the earlier version of the row in the history table.
- Assign the transaction ID and generate a new sequence number, persisting them in the appropriate system columns.
- Serialize the row content and include it when computing the hash for all rows updated by this transaction.

Ledger achieves that by extending the [Data Manipulation Language](#) (DML) query plans of all insert, update and delete operations targeting ledger tables. The transaction ID and newly generated sequence number are set for the new version of the row. Then, the query plan operator executes a special expression that serializes the row content and computes its hash, appending it to a Merkle Tree that is stored at the transaction level and contains the hashes of all row versions updated by this transaction for this ledger table. The root of the tree represents all the updates and deletes performed by this transaction in this ledger table. If the transaction updates multiple tables, a separate Merkle Tree is maintained for each table. The figure below shows an example of a Merkle Tree storing the updated row versions of a ledger table and the format used to serialize the rows. Other than the serialized value of each column, we include metadata regarding the number of columns in the row, the ordinal of individual columns, the data types, lengths and other information that affects how the values are interpreted.



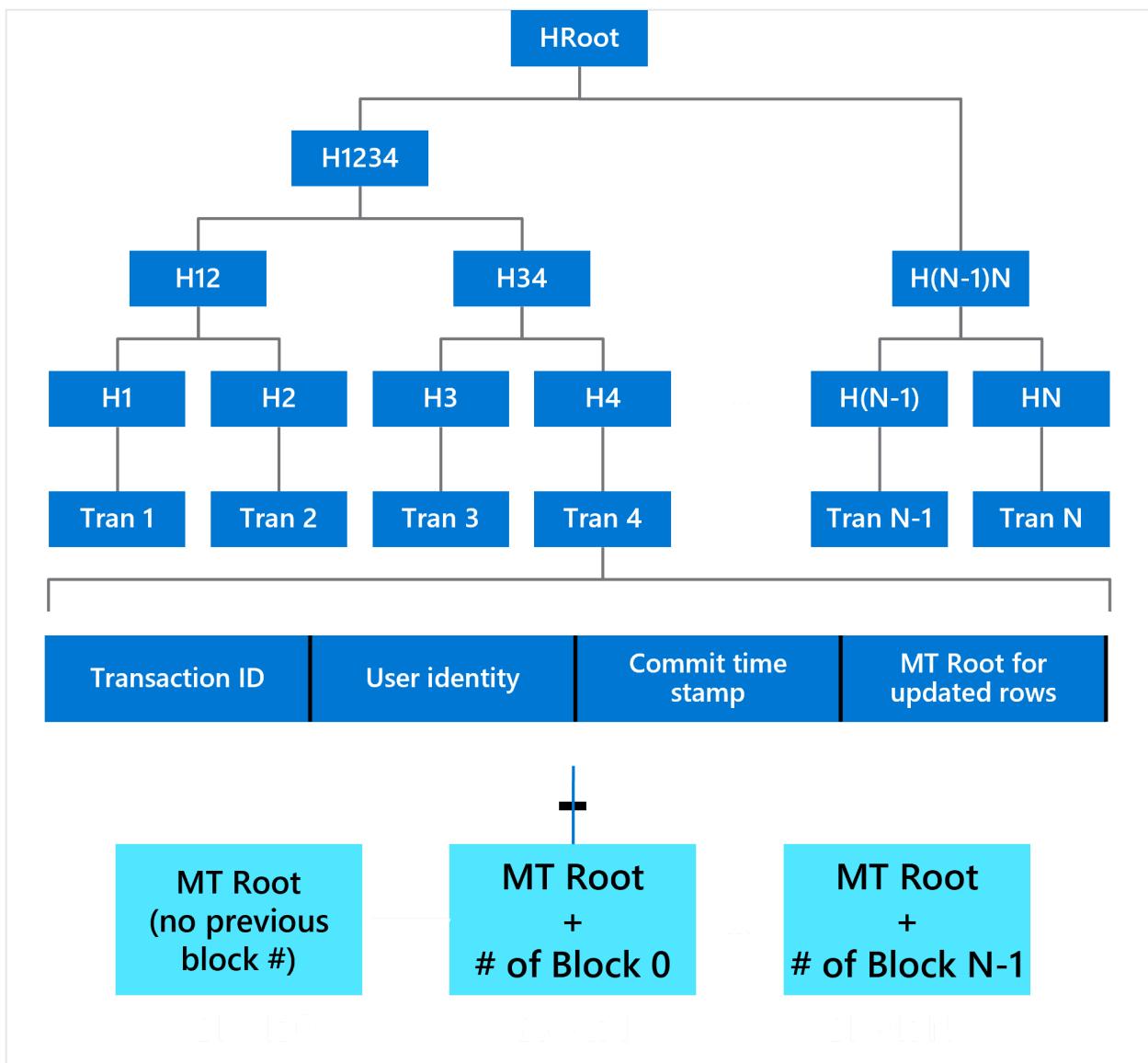
To capture the state of the database, the database ledger stores an entry for every transaction. It captures metadata about the transaction, such as its commit timestamp and the identity of the user who executed it. It also captures the Merkle tree root of the rows updated in each ledger table (see above). These entries are then appended to a tamper-evident data structure to allow verification of integrity in the future. A block is closed:

- Approximately every 30 seconds, when your database is configured for [automatic database digest storage](#)
- When the user manually generates a database digest by running the `sys.sp_generate_database_ledger_digest` stored procedure
- When it contains 100K transactions.

When a block is closed, new transactions will be inserted in a new block. The block generation process then:

1. Retrieves all transactions that belong to the *closed* block from both the in-memory queue and the `sys.database_ledger_transactions` system catalog view.
2. Computes the Merkle tree root over these transactions and the hash of the previous block.
3. Persists the closed block in the `sys.database_ledger_blocks` system catalog view.

Because this is a regular table update, the system automatically guarantees its durability. To maintain the single chain of blocks, this operation is single-threaded. But it's also efficient, because it only computes the hashes over the transaction information and happens asynchronously. It doesn't affect the transaction performance.



For more information on how ledger provides data integrity, see the articles, [Digest management](#) and [Database verification](#).

Where are database transactions and block data stored?

The data for transactions and blocks is physically stored as rows in two system catalog views:

- [sys.database_ledger_transactions](#): Maintains a row with the information of each transaction in the database ledger. The information includes the ID of the block where this transaction belongs and the ordinal of the transaction within the block.
- [sys.database_ledger_blocks](#): Maintains a row for every block in the ledger, including the root of the Merkle tree over the transactions within the block and the hash of the previous block to form a blockchain.

To view the database ledger, run the following T-SQL statements in [SQL Server Management Studio](#), [Azure Data Studio](#) or [SQL Server Developer Tools](#).

```
SQL

SELECT * FROM sys.database_ledger_transactions;
GO

SELECT * FROM sys.database_ledger_blocks;
GO
```

The following example of a ledger table consists of four transactions that made up one block in the blockchain of the database ledger:

Results		Messages				
	transaction_id	block_id	transaction_ordinal	commit_time	principal_name	table_hashes
1	999	0	0	2021-03-23 20:18:08.2700000	janders	0xB982EE5A88DFE8EF08BE7564DC2273BD17306231C8E22E052644805...
2	1002	0	1	2021-03-23 20:18:12.9300000	janders	0xB982EE5AB931133CF9B8E6FC06C9AF25C0F0C6A9A91A12C89A84AB...
3	1055	0	2	2021-03-23 20:40:08.9500000	janders	0xB982EE5A38F20FA9D8ABFC3C3523284FE65466DAA9E91166447648B...
4	1091	0	3	2021-03-23 21:36:22.2533333	janders	0x9D13BF5E345245E7456EC748BC895E0E1323379BD04EBC35638D91E...
	block_id	transactions_root_hash		block_size	previous_block_hash	
1	0	0x8F3C4C8ADF99EAE24A783CB1AC282A12E9C9ECA619DDE19B2C98D8ECCA5E4A5		4	NULL	

Permissions

Viewing the database ledger requires the `VIEW LEDGER CONTENT` permission. For details on permissions related to ledger tables, see [Permissions](#).

See also

- [Ledger overview](#)
- [Data Manipulation Language \(DML\)](#)
- [Ledger views](#)

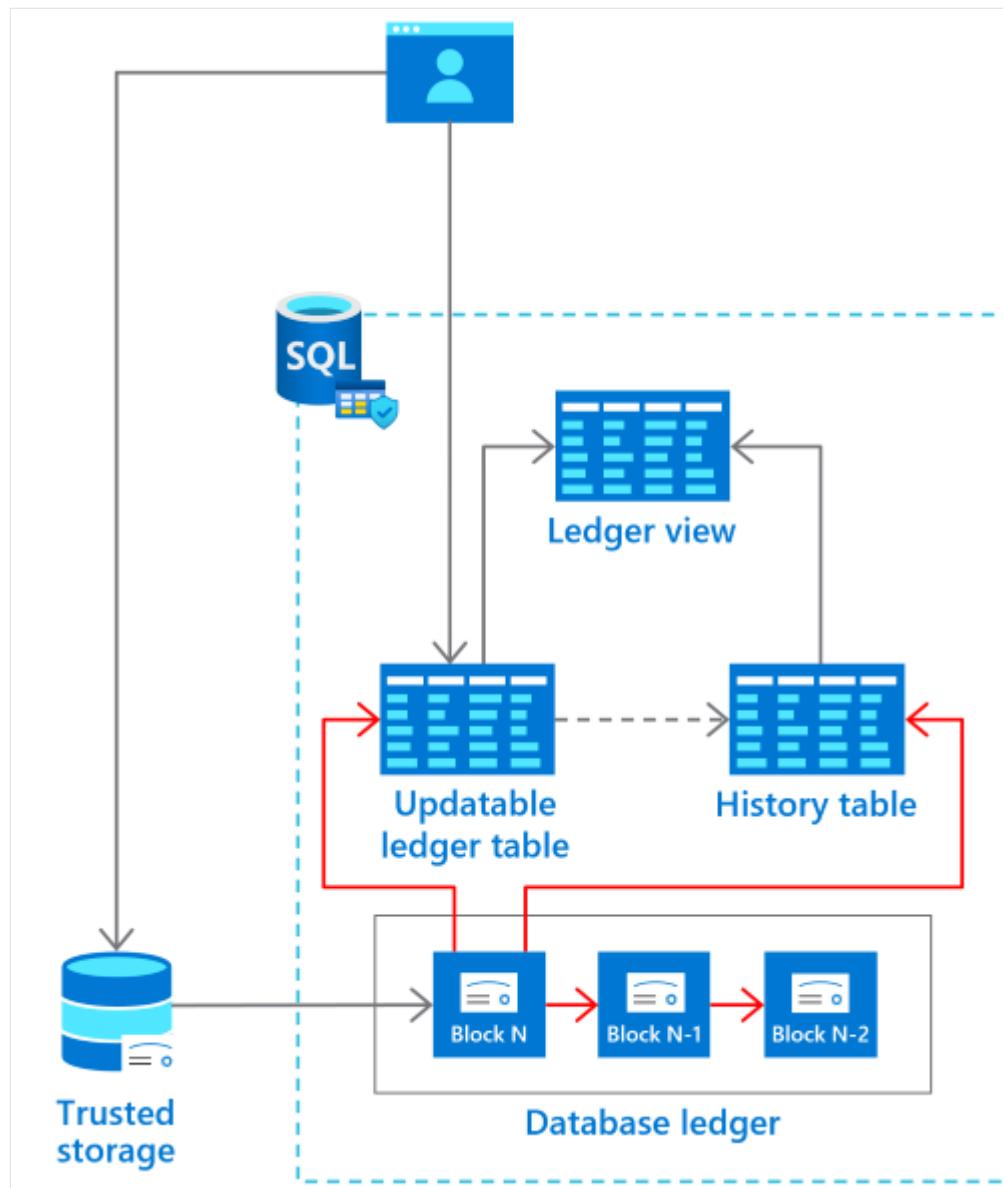
Updatable ledger tables

Article • 05/23/2023

Applies to:  SQL Server 2022 (16.x)  Azure SQL Database  Azure SQL Managed Instance

Updatable ledger tables are system-versioned tables on which users can perform updates and deletes while also providing tamper-evidence capabilities. When updates or deletes occur, all earlier versions of a row are preserved in a secondary table, known as the history table. The history table mirrors the schema of the updatable ledger table. When a row is updated, the latest version of the row remains in the ledger table, while its earlier version is inserted into the history table by the system, transparently to the application.

Both updatable ledger tables and [temporal tables](#) are system-versioned tables, for which the database engine captures historical row versions in secondary history tables. Either technology provides unique benefits. Updatable ledger tables make both the current and historical data tamper evident. Temporal tables support querying the data stored at any point in time instead of only the data that's correct at the current moment in time. You can use both technologies together by creating tables that are both updatable ledger tables and temporal tables.



You can create an updatable ledger table by specifying the `LEDGER = ON` argument in your [CREATE DATABASE \(Transact-SQL\)](#) statement.

 **Tip**

`LEDGER = ON` is optional when creating updatable ledger tables in a ledger database. By default, each table is an updatable ledger table in a ledger database.

For information on options available when you specify the `LEDGER` argument in your T-SQL statement, see [CREATE TABLE \(Transact-SQL\)](#).

 **Important**

After a ledger table is created, it can't be reverted to a table that isn't a ledger table. As a result, an attacker can't temporarily remove ledger capabilities on a ledger table, make changes, and then reenable ledger functionality.

Updatable ledger table schema

An updatable ledger table needs to have the following **GENERATED ALWAYS** columns that contain metadata noting which transactions made changes to the table and the order of operations by which rows were updated by the transaction. This data is useful for forensics purposes in understanding how data was inserted over time.

If you don't specify the required **GENERATED ALWAYS** columns of the ledger table and ledger history table in the [CREATE TABLE \(Transact-SQL\)](#) statement, the system automatically adds the columns and uses the following default names. For more information, see examples in [Creating an updatable ledger table](#).

Default column name	Data type	Description
ledger_start_transaction_id	bigint	The ID of the transaction that created a row version
ledger_end_transaction_id	bigint	The ID of the transaction that deleted a row version
ledger_start_sequence_number	bigint	The sequence number of an operation within a transaction that created a row version
ledger_end_sequence_number	bigint	The sequence number of an operation within a transaction that deleted a row version

History table

The history table is automatically created when an updatable ledger table is created. The history table captures the historical values of rows changed because of updates and deletes in the updatable ledger table. The schema of the history table mirrors that of the updatable ledger table it's associated with.

When you create an updatable ledger table, you can either specify the name of the schema to contain your history table and the name of the history table or you have the system generate the name of the history table and add it to the same schema as the ledger table. History tables with system-generated names are called anonymous history tables. The naming convention for an anonymous history table is `<schema>.<updatableledgertablename>.MSSQL_LedgerHistoryFor_<GUID>`.

`<updatableledgertablename>.MSSQL_LedgerHistoryFor_<GUID>`.

Ledger view

For every updatable ledger table, the system automatically generates a view, called the ledger view. The ledger view is a join of the updatable ledger table and its associated history table. The ledger view reports all row modifications that have occurred on the updatable ledger table by joining the historical data in the history table. This view enables users, their partners, or auditors to analyze all historical operations and detect potential tampering. Each row operation is accompanied by the ID of the acting transaction, along with whether the operation was a `DELETE` or an `INSERT`. Users can retrieve more information about the time the transaction was executed and the identity of the user who executed it and correlate it to other operations performed by this transaction.

For example, if you want to track transaction history for a banking scenario, the ledger view provides a chronicle of transactions over time. By using the ledger view, you don't have to independently view the updatable ledger table and history tables or construct your own view to do so.

For an example of using the ledger view, see [Create and use updatable ledger tables](#).

The ledger view's schema mirrors the columns defined in the updatable ledger and history table, but the `GENERATED ALWAYS` columns are different than those of the updatable ledger and history tables.

Ledger view schema

Note

The ledger view column names can be customized when you create the table by using the `<ledger_view_option>` parameter with the [CREATE TABLE \(Transact-SQL\)](#) statement. For more information, see [ledger view options](#) and the corresponding examples in [CREATE TABLE \(Transact-SQL\)](#).

Default column name	Data type	Description
ledger_transaction_id	bigint	The ID of the transaction that created or deleted a row version.
ledger_sequence_number	bigint	The sequence number of a row-level operation within the transaction on the table.
ledger_operation_type	tinyint	Contains <code>1</code> (<code>INSERT</code>) or <code>2</code> (<code>DELETE</code>). Inserting a row into the ledger table produces a new row in the

Default column name	Data type	Description
		ledger view that contains 1 in this column. Deleting a row from the ledger table produces a new row in the ledger view that contains 2 in this column. Updating a row in the ledger table produces two new rows in the ledger view. One row contains 2 (DELETE), and the other row contains 1 (INSERT) in this column.
ledger_operation_type_desc	nvarchar(128)	Contains <code>INSERT</code> or <code>DELETE</code> . For more information, see the preceding row.

Next steps

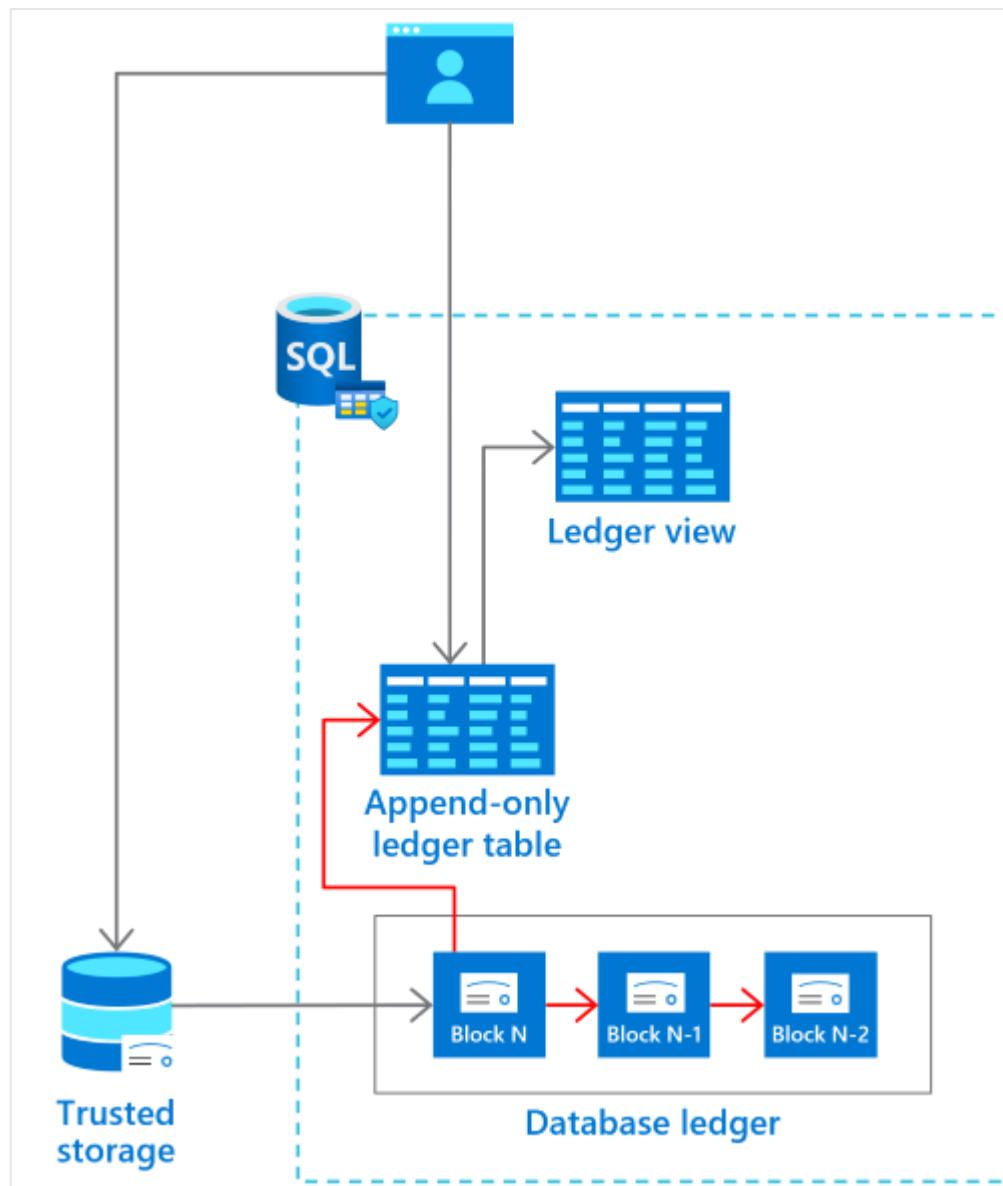
- [Create and use updatable ledger tables](#)
- [Create and use append-only ledger tables](#)
- [How to migrate data from regular tables to ledger tables](#)

Append-only ledger tables

Article • 02/28/2023

Applies to: ✓ SQL Server 2022 (16.x) ✓ Azure SQL Database ✓ Azure SQL Managed Instance

Append-only ledger tables allow only `INSERT` operations on your tables, which ensure that privileged users such as database administrators can't alter data through traditional [Data Manipulation Language](#) operations. Append-only ledger tables are ideal for systems that don't update or delete records, such as security information event and management systems or blockchain systems where data needs to be replicated from the blockchain to a database. Because there are no `UPDATE` or `DELETE` operations on an append-only table, there's no need for a corresponding history table as there is with [updatable ledger tables](#).



You can create an append-only ledger table by specifying the `LEDGER = ON` argument in your [CREATE TABLE \(Transact-SQL\)](#) statement and specifying the `APPEND_ONLY = ON` option.

Important

After a table is created as a ledger table, it can't be reverted to a table that doesn't have ledger functionality. As a result, an attacker can't temporarily remove ledger capabilities, make changes to the table, and then reenable ledger functionality.

Append-only ledger table schema

An append-only table needs to have the following [GENERATED ALWAYS](#) columns that contain metadata noting which transactions made changes to the table and the order of operations by which rows were updated by the transaction. When you create an append-only ledger table, `GENERATED ALWAYS` columns will be created in your ledger table. This data is useful for forensics purposes in understanding how data was inserted over time.

If you don't specify the definitions of the `GENERATED ALWAYS` columns in the [CREATE TABLE](#) statement, the system automatically adds them by using the following default names.

Default column name	Data type	Description
<code>ledger_start_transaction_id</code>	<code>bigint</code>	The ID of the transaction that created a row version
<code>ledger_start_sequence_number</code>	<code>bigint</code>	The sequence number of an operation within a transaction that created a row version

Ledger view

For every append-only ledger table, the system automatically generates a view, called the ledger view. The ledger view reports all row inserts that have occurred on the table. The ledger view is primarily helpful for [updatable ledger tables](#), rather than append-only ledger tables, because append-only ledger tables don't have any `UPDATE` or `DELETE` capabilities. The ledger view for append-only ledger tables is available for consistency between both updatable and append-only ledger tables.

Ledger view schema

ⓘ Note

The ledger view column names can be customized when you create the table by using the `<ledger_view_option>` parameter with the **CREATE TABLE (Transact-SQL)** statement. For more information, see **ledger view options** and the corresponding examples in **CREATE TABLE (Transact-SQL)**.

Default column name	Data type	Description
ledger_transaction_id	bigint	The ID of the transaction that created or deleted a row version.
ledger_sequence_number	bigint	The sequence number of a row-level operation within the transaction on the table.
ledger_operation_type	tinyint	Contains <code>1</code> (INSERT) or <code>2</code> (DELETE). Inserting a row into the ledger table produces a new row in the ledger view that contains <code>1</code> in this column. Deleting a row from the ledger table produces a new row in the ledger view that contains <code>2</code> in this column. Updating a row in the ledger table produces two new rows in the ledger view. One row contains <code>2</code> (DELETE), and the other row contains <code>1</code> (INSERT) in this column. A DELETE shouldn't occur on an append-only ledger table.
ledger_operation_type_desc	nvarchar(128)	Contains INSERT or DELETE . For more information, see the preceding row.

Next steps

- [Create and use append-only ledger tables](#)
- [Create and use updatable ledger tables](#)
- [How to migrate data from regular tables to ledger tables](#)

Digest management

Article • 11/15/2023

Applies to:  SQL Server 2022 (16.x)  Azure SQL Database  Azure SQL Managed Instance

Database digests

The hash of the latest block in the database ledger is called the *database digest*. It represents the state of all ledger tables in the database at the time when the block was generated. Generating a database digest is efficient, because it involves computing only the hashes of the blocks that were recently appended.

Database digests can be generated either automatically by the system or manually by the user. You can use them later to verify the integrity of the database.

Database digests are generated in the form of a JSON document that contains the hash of the latest block, together with metadata for the block ID. The metadata includes the time that the digest was generated and the commit time stamp of the last transaction in this block.

The verification process and the integrity of the database depend on the integrity of the input digests. For this purpose, database digests that are extracted from the database need to be stored in trusted storage that the high-privileged users or attackers of the database can't tamper with.

Automatic generation and storage of database digests

Note

Automatic generation and storage of database digests in SQL Server only supports Azure Storage accounts.

Ledger integrates with the [immutable storage feature of Azure Blob Storage](#) and [Azure Confidential Ledger](#). This integration provides secure storage services in Azure to help protect the database digests from potential tampering. This integration provides a simple and cost-effective way for users to automate digest management without having to worry about their availability and geographic replication. Azure Confidential Ledger has a stronger integrity guarantee for customers who might be concerned about

privileged administrators access to the digest. [This table](#) compares the immutable storage feature of Azure Blob Storage with Azure Confidential Ledger.

You can configure automatic generation and storage of database digests through the Azure portal, PowerShell, or the Azure CLI. For more information, see [Enable automatic digest storage](#). When you configure automatic generation and storage, database digests are generated on a predefined interval of 30 seconds and uploaded to the selected storage service. If no transactions occur on the system in the 30-second interval, a database digest won't be generated and uploaded. This mechanism ensures that database digests are generated only when data has been updated in your database. When the endpoint is an Azure Blob Storage, the [logical server for Azure SQL Database](#) or Azure SQL Managed Instance creates a new container, named `sqldbledgerdigests` and uses a naming pattern like: `ServerName/DatabaseName/CreationTime`. The creation time is needed because a database with the same name can be dropped and recreated or restored, allowing for different *incarnations* of the database under the same name. For more information, see [Digest Management Considerations](#).

 **Note**

For SQL Server, the container needs to be created manually by the user.

Azure Storage Account Immutability Policy

If you use an Azure Storage account for the storage of the database digests, configure an [immutability policy](#) on your container after provisioning to ensure that database digests are protected from tampering. Make sure the immutability policy allows protected append writes to append blobs and that the policy is locked.

Azure Storage account permission

If you use **Azure SQL Database** or **Azure SQL Managed Instance**, make sure that your logical server or managed instance (System Identity) has sufficient RBAC permissions to write digests by adding it to the [Storage Blob Data Contributor](#) role. In case you use Active geo-replication or auto-failover groups make sure that the secondary replica(s) have the same RBAC permission on the Azure Storage account.

If you use **SQL Server**, you have to create a shared access signature (SAS) on the digest container to allow SQL Server to connect and authenticate against the Azure Storage account.

- Create a container on the Azure Storage account, named `sqldbledgerdigests`.

- Create a policy on a container with the *Read, Add, Create, Write, and List* permissions, and generate a shared access signature key.
- For the `sqlbledgerdigests` container used for digest file storage, create a [SQL Server credential](#) whose name matches the container path.

The following example assumes that an Azure Storage container, a policy, and a SAS key have been created. This is needed by SQL Server to access the digest files in the container.

In the following code snippet, replace `<your SAS key>` with the SAS key. The SAS key looks like `'sr=c&si=<MYPOLICYNAME>&sig=<THESHAREDACCESSIONSIGNATURE>'`.

SQL

```
CREATE CREDENTIAL [https://ledgerstorage.blob.core.windows.net/sqlbledgerdigests]
WITH IDENTITY='SHARED ACCESS SIGNATURE',
SECRET = '<your SAS key>'
```

Azure Confidential Ledger Permission

If you use [Azure SQL Database](#) or [Azure SQL Managed Instance](#), make sure that your logical server or managed instance (System Identity) has sufficient permissions to write digests by adding it to the **Contributor** role. To do this, follow the steps for [Azure Confidential Ledger user management](#).

ⓘ Note

Automatic generation and storage of database digests in SQL Server only supports Azure Storage accounts.

Manual generation and storage of database digests

You can also generate a database digest on demand so that you can manually store the digest in any service or device that you consider a trusted storage destination. For example, you might choose an on-premises write once, read many (WORM) device as a destination. You manually generate a database digest by running the `sys.sp_generate_database_ledger_digest` stored procedure in either [SQL Server Management Studio](#) or [Azure Data Studio](#).

SQL

```
EXECUTE sp_generate_database_ledger_digest;
```

The returned result set is a single row of data. It should be saved to the trusted storage location as a JSON document as follows:

JSON

```
{  
    "database_name": "ledgerdb",  
    "block_id": 0,  
    "hash":  
        "0xDC160697D823C51377F97020796486A59047EBDBF77C3E8F94EEE0FFF7B38A6A",  
    "last_transaction_commit_time": "2020-11-12T18:01:56.6200000",  
    "digest_time": "2020-11-12T18:39:27.7385724"  
}
```

Permissions

Generating database digests requires the `GENERATE LEDGER DIGEST` permission. For details on permissions related to ledger tables, see [Permissions](#).

Digest management considerations

Database restore

Restoring the database back to an earlier point in time, also known as [Point in Time Restore](#), is an operation frequently used when a mistake occurs and users need to quickly revert the state of the database back to an earlier point in time. When uploading the generated digests to Azure Storage or Azure Confidential Ledger, the *create time* of the database is captured that these digests map to. Every time the database is restored, it's tagged with a new *create time* and this technique allows us to store the digests across different "incarnations" of the database. For SQL Server, the *create time* is the current UTC time when the digest upload is enabled for the first time. Ledger preserves the information regarding when a restore operation occurred, allowing the verification process to use all the relevant digests across the various incarnations of the database. Additionally, users can inspect all digests for different create times to identify when the database was restored and how far back it was restored to. Since this data is written in immutable storage, this information will be protected as well.

 **Note**

If you perform a native restore of a database backup in Azure SQL Managed Instance, you need to change the digest path manually using the Azure Portal, PowerShell or the Azure CLI.

Active geo-replication and Always On availability groups

Active geo-replication or auto-failover groups can be configured for Azure SQL Database or Azure SQL Managed Instance. Replication across geographic regions is asynchronous for performance reasons and, thus, allows the secondary database to be slightly behind compared to the primary. In the event of a geographic failover, any latest data that hasn't yet been replicated is lost. Ledger will only issue database digests for data that has been replicated to geographic secondaries to guarantee that digests will never reference data that might be lost in case of a geographic failover. This only applies for automatic generation and storage of database digests. In a failover group, both primary and secondary database will have the same digest path. Even when you perform a failover, the digest path doesn't change for both primary and secondary database.

If failover group is deleted or you drop the link, both databases will behave as primary databases. At that point the digest path of the previous secondary database will change and we will add a folder `RemovedSecondaryReplica` to the path.

When your database is part of an Always On availability group or a Managed Instance link in SQL Server, the same principle as active geo-replication is used. The upload of the digests is only done if all transactions have been replicated to the secondary replicas.

Next steps

- [Ledger overview](#)
- [Enable automatic digest storage](#)
- [sys.sp_generate_database_ledger_digest](#)

Database verification

Article • 11/14/2023

Applies to:  SQL Server 2022 (16.x)  Azure SQL Database  Azure SQL Managed Instance

Ledger provides a form of data integrity called *forward integrity*, which provides evidence of data tampering on data in your ledger tables. The database verification process takes as input one or more previously generated database digests. It then recomputes the hashes stored in the database ledger based on the current state of the ledger tables. If the computed hashes don't match the input digests, the verification fails. The failure indicates that the data has been tampered with. The verification process reports all inconsistencies that it detects.

Database verification process

The verification process scans all ledger and history tables. It recomputes the SHA-256 hashes of their rows and compares them against the database digest files passed to the verification stored procedure.

Because the ledger verification recomputes all of the hashes for transactions in the database, it can be a resource-intensive process for databases with large amounts of data. To reduce the cost of verification, the feature exposes options to verify individual ledger tables or only a subset of the ledger tables.

You accomplish database verification through two stored procedures, depending on whether you [use automatic digest storage](#) or you [manually manage digests](#).

Note

The database option **ALLOW_SNAPSHOT_ISOLATION** has to be enabled on the database before you can run the verification stored procedures.

Database verification that uses automatic digest storage

When you're using automatic digest storage for generating and storing database digests, the location of the digest storage is in the system catalog view [sys.database_ledger_digest_locations](#) as JSON objects. Running database verification consists of executing the [sp_verify_database_ledger_from_digest_storage](#) system stored

procedure. Specify the JSON objects from the [sys.database_ledger_digest_locations](#) system catalog view where database digests are configured to be stored.

When you use automatic digest storage, you can change storage locations throughout the lifecycle of the ledger tables. For example, if you start by using Azure immutable storage to store your digest files, but later you want to use Azure Confidential Ledger instead, you can do so. This change in location is stored in [sys.database_ledger_digest_locations](#).

When you run ledger verification, inspect the location of **digest_locations** to ensure digests used in verification are retrieved from the locations you expect. You want to make sure that a privileged user hasn't changed locations of the digest storage to an unprotected storage location, such as Azure Storage, without a configured and locked immutability policy.

To simplify running verification when you use multiple digest storage locations, the following script will fetch the locations of the digests and execute verification by using those locations.

SQL

```
DECLARE @digest_locations NVARCHAR(MAX) = (SELECT * FROM
sys.database_ledger_digest_locations FOR JSON AUTO, INCLUDE_NULL_VALUES);
SELECT @digest_locations as digest_locations;
BEGIN TRY
    EXEC sys.sp_verify_database_ledger_from_digest_storage
@digest_locations;
    SELECT 'Ledger verification succeeded.' AS Result;
END TRY
BEGIN CATCH
    THROW;
END CATCH
```

Database verification that uses manual digest storage

When you're using manual digest storage for generating and storing database digests, the stored procedure [sp_verify_database_ledger](#) is used to verify the ledger database. The JSON content of the digest is appended in the stored procedure. When you're running database verification, you can choose to verify all tables in the database or verify specific tables.

The following code is an example of running the [sp_verify_database_ledger](#) stored procedure by passing two digests for verification:

SQL

```

EXECUTE sp_verify_database_ledger N'
[
  {
    "database_name": "ledgerdb",
    "block_id": 0,
    "hash":
"0xDC160697D823C51377F97020796486A59047EBDBF77C3E8F94EEE0FFF7B38A6A",
    "last_transaction_commit_time": "2020-11-12T18:01:56.6200000",
    "digest_time": "2020-11-12T18:39:27.7385724"
  },
  {
    "database_name": "ledgerdb",
    "block_id": 1,
    "hash":
"0xE5BE97FDFFA4A16ADF7301C8B2BEBC4BAE5895CD76785D699B815ED2653D9EF8",
    "last_transaction_commit_time": "2020-11-12T18:39:35.6633333",
    "digest_time": "2020-11-12T18:43:30.4701575"
  }
];

```

Return codes for `sp_verify_database_ledger` and `sp_verify_database_ledger_from_digest_storage` are `0` (success) or `1` (failure).

Recommendation

Ideally, you want to minimize or even eliminate the gap between the time the attack occurred and the time it was detected. Microsoft recommends scheduling the ledger verification] regularly to avoid a restore of the database from days or months ago after [tampering was detected](#). The interval of the verification should be decided by the customer, but be aware that ledger verification can be resource consuming. We recommend running this during a maintenance window or off peak hours.

Scheduling database verification in Azure SQL Database can be done with Elastic Jobs or Azure Automation. For scheduling the database verification in Azure SQL Managed Instance and SQL Server, you can use SQL Server Agent.

Permissions

Database verification requires the `VIEW LEDGER CONTENT` permission. For details on permissions related to ledger tables, see [Permissions](#).

Next steps

- [Ledger overview](#)

- Verify a ledger table to detect tampering
- `sys.database_ledger_digest_locations`
- `sp_verify_database_ledger_from_digest_storage`
- `sp_verify_database_ledger`

Ledger considerations and limitations

Article • 11/28/2023

Applies to:  SQL Server 2022 (16.x)  Azure SQL Database  Azure SQL Managed Instance

There are some considerations and limitations to be aware of when working with ledger tables due to the nature of system-versioning and immutable data.

General considerations and limitations

Consider the following when working with ledger.

- A [ledger database](#), a database with the ledger property set to on, can't be converted to a regular database, with the ledger property set to off.
- Automatic generation and storage of database digests is currently available in Azure SQL Database, but not supported on SQL Server.
- Automated digest management with ledger tables by using [Azure Storage immutable blobs](#) doesn't offer the ability for users to use [locally redundant storage \(LRS\)](#) accounts.
- When a ledger database is created, all new tables created by default (without specifying the `APPEND_ONLY = ON` clause) in the database will be [updatable ledger tables](#). To create [append-only ledger tables](#), use the `APPEND_ONLY = ON` clause in the [CREATE TABLE \(Transact-SQL\)](#) statements.
- A transaction can update up to 200 ledger tables.

Ledger table considerations and limitations

- Existing tables in a database that aren't ledger tables can't be converted to ledger tables. For more information, see [Migrate data from regular tables to ledger tables](#).
- After a ledger table is created, it can't be reverted to a table that isn't a ledger table.
- Deleting older data in [append-only ledger tables](#) or the history table of [updatable ledger tables](#) isn't supported.
- `TRUNCATE TABLE` isn't supported.
- When an [updatable ledger table](#) is created, it adds four [GENERATED ALWAYS](#) columns to the ledger table. An [append-only ledger table](#) adds two columns to the ledger table. These new columns count against the maximum supported number of columns in Azure SQL Database (1,024).

- In-memory tables aren't supported.
- Sparse column sets aren't supported.
- SWITCH IN/OUT partition isn't supported.
- DBCC CLONEDATABASE isn't supported.
- Ledger tables can't have full-text indexes.
- Ledger tables can't be graph table.
- Ledger tables can't be FileTables.
- Ledger tables can't have a rowstore non-clustered index when they have a clustered columnstore index.
- Change tracking isn't allowed on the history table but is allowed on ledger tables.
- Change data capture isn't allowed on the history table, but is allowed on ledger tables.
- Transactional replication isn't supported for ledger tables.
- Database mirroring isn't supported.
- Azure Synapse Link is supported but only for the ledger table, not the history table.
- Change the digest path manually after a native restore of a database backup to an Azure SQL Managed Instance.
- Change the digest path manually after a Managed Instance link was created to an Azure SQL Managed Instance.
- SQL Data Sync isn't supported with ledger tables.

Unsupported data types

- XML
- SqlVariant
- User-defined data type
- FILESTREAM

Temporal table limitations

Updatable ledger tables are based on the technology of [temporal tables](#) and inherit most of the [limitations](#) but not all of them. Below is a list of limitations that is inherited from temporal tables.

- If the name of a history table is specified during history table creation, you must specify the schema and table name and also the name of the ledger view.
- By default, the history table is PAGE compressed.
- If the current table is partitioned, the history table is created on the default file group because partitioning configuration isn't replicated automatically from the current table to the history table.

- Temporal and history tables can't be a FILETABLE and can contain columns of any supported datatype other than FILESTREAM. FILETABLE and FILESTREAM allow data manipulation outside of SQL Server, and thus system versioning can't be guaranteed.
- A node or edge table can't be created as or altered to a temporal table. Graph isn't supported with ledger.
- While temporal tables support blob data types, such as `(n)varchar(max)`, `varbinary(max)`, `(n)text`, and `image`, they'll incur significant storage costs and have performance implications due to their size. As such, when designing your system, care should be taken when using these data types.
- The history table must be created in the same database as the current table. Temporal querying over Linked Server isn't supported.
- The history table can't have constraints (Primary Key, Foreign Key, table, or column constraints).
- Online option (`WITH (ONLINE = ON)`) has no effect on `ALTER TABLE ALTER COLUMN` in case of system-versioned temporal table. `ALTER COLUMN` isn't performed as online regardless of which value was specified for the `ONLINE` option.
- `INSERT` and `UPDATE` statements can't reference the `GENERATED ALWAYS` columns. Attempts to insert values directly into these columns will be blocked.
- `UPDATETEXT` and `WRITETEXT` aren't supported.
- Triggers on the history table aren't allowed.
- Usage of replication technologies is limited:
 - Always On: Fully supported
 - Snapshot, merge and transactional replication: Not supported for temporal tables
- A history table can't be configured as current table in a chain of history tables.
- The following objects or properties aren't replicated from the current table to the history table when the history table is created:
 - Period definition
 - Identity definition
 - Indexes
 - Statistics
 - Check constraints
 - Triggers
 - Partitioning configuration
 - Permissions
 - Row-level security predicates

Schema changes consideration

Adding columns

Adding nullable columns is supported. Adding non-nullable columns is not supported. Ledger is designed to ignore NULL values when computing the hash of a row version. Based on that, when a nullable column is added, ledger will modify the schema of the ledger and history tables to include the new column, however, this doesn't impact the hashes of existing rows. Adding columns in ledger tables is captured in [sys.ledger_column_history](#).

Dropping columns and tables

Normally, dropping a column or table completely erases the underlying data from the database and is fundamentally incompatible with the ledger functionality that requires data to be immutable. Instead of deleting the data, ledger simply renames the objects being dropped so that they're logically removed from the user schema, but physically remain in the database. Any dropped columns are also hidden from the ledger table schema, so that they're invisible to the user application. However, the data of such dropped objects remains available for the ledger verification process, and allows users to inspect any historical data through the corresponding ledger views. Dropping columns in ledger tables is captured in [sys.ledger_column_history](#). Dropping a ledger table is captured in [sys.ledger_table_history](#). Dropping ledger tables and its dependent objects are marked as dropped in system catalog views and renamed:

- Dropped ledger tables are marked as dropped by setting `is_dropped_ledger_table` in `sys.tables` and renamed using the following format:
`MSSQL_DroppedLedgerTable_<dropped_ledger_table_name>_<GUID>`.
- Dropped history tables for updatable ledger tables are renamed using the following format:
`MSSQL_DroppedLedgerHistory_<dropped_history_table_name>_<GUID>`.
- Dropped ledger views are marked as dropped by setting `is_dropped_ledger_view` in `sys.views` and renamed using the following format:
`MSSQL_DroppedLedgerView_<dropped_ledger_view_name>_<GUID>`.

Note

The name of dropped ledger tables, history tables and ledger views might be truncated if the length of the renamed table or view exceeds 128 characters.

Altering Columns

Any changes that don't impact the underlying data of a ledger table are supported without any special handling as they don't impact the hashes being captured in the ledger. These changes include:

- Changing nullability
- Collation for Unicode strings
- The length of variable length columns

However, any operations that might affect the format of existing data, such as changing the data type aren't supported.

Related content

- [Ledger overview](#)
- [Updatable ledger tables](#)
- [Append-only ledger tables](#)
- [Database ledger](#)

Ledger overview

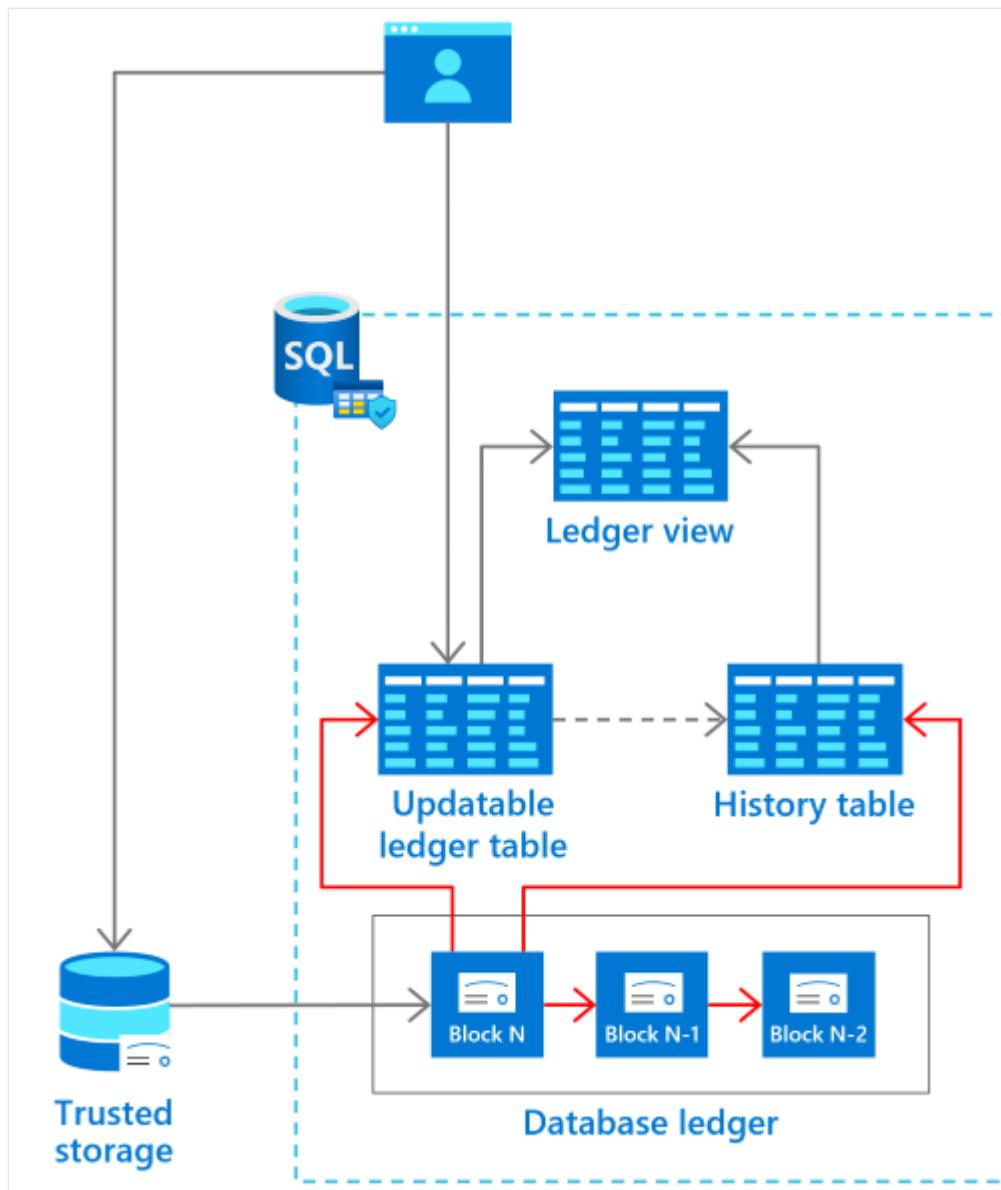
Article • 11/14/2023

Applies to:  SQL Server 2022 (16.x)  Azure SQL Database  Azure SQL Managed Instance

Establishing trust around the integrity of data stored in database systems has been a longstanding problem for all organizations that manage financial, medical, or other sensitive data. The ledger feature provides tamper-evidence capabilities in your database. You can cryptographically attest to other parties, such as auditors or other business parties, that your data hasn't been tampered with.

Ledger helps protect data from any attacker or high-privileged user, including database administrators (DBAs), system administrators, and cloud administrators. As with a traditional ledger, the feature preserves historical data. If a row is updated in the database, its previous value is maintained and protected in a history table. Ledger provides a chronicle of all changes made to the database over time.

Ledger and the historical data are managed transparently, offering protection without any application changes. The feature maintains historical data in a relational form to support SQL queries for auditing, forensics, and other purposes. It provides guarantees of cryptographic data integrity while maintaining the power, flexibility, and performance of the SQL database.



Use cases for ledger

Let's go over some advantages for using ledger.

Streamlining audits

Any production system's value is based on the ability to trust the data that the system is consuming and producing. If a malicious user has tampered with the data in your database, that can have disastrous results in the business processes relying on that data.

Maintaining trust in your data requires a combination of enabling the proper security controls to reduce potential attacks, backup and restore practices, and thorough disaster recovery procedures. Audits by external parties ensure that these practices are put in place.

Audit processes are highly time-intensive activities. Auditing requires on-site inspection of implemented practices such as reviewing audit logs, inspecting authentication, and inspecting access controls. Although these manual processes can expose potential gaps in security, they can't provide attestable proof that the data hasn't been maliciously altered.

Ledger provides the cryptographic proof of data integrity to auditors. This proof can help streamline the auditing process. It also provides nonrepudiation regarding the integrity of the system's data.

Multiple-party business processes

In some systems, such as supply-chain management systems, multiple organizations must share state from a business process with one another. These systems struggle with the challenge of how to share and trust data. Many organizations are turning to traditional blockchains, such as Ethereum or Hyperledger Fabric, to digitally transform their multiple-party business processes.

Blockchain is a great solution for multiple-party networks where trust is low between parties that participate on the network. Many of these networks are fundamentally centralized solutions where trust is important, but a fully decentralized infrastructure is a heavyweight solution.

Ledger provides a solution for these networks. Participants can verify the integrity of the centrally housed data, without the complexity and performance implications that network consensus introduces in a blockchain network.

Customer success

- Learn how [Lenovo is reinforcing customer trust using ledger in Azure SQL Database](#) by watching this [video](#).
- [RTGS.global using ledger in Azure SQL Database to establish trust with banks around the world](#).
- [Qode Health Solutions secures COVID-19 vaccination records with the ledger feature in Azure SQL Database](#)

Trusted off-chain storage for blockchain

When a blockchain network is necessary for a multiple-party business process, the ability to query the data on the blockchain without sacrificing performance is a challenge.

Typical patterns for solving this problem involve replicating data from the blockchain to an off-chain store, such as a database. But after the data is replicated to the database from the blockchain, the data integrity guarantees that a blockchain offer is lost. Ledger provides data integrity for off-chain storage of blockchain networks, which helps ensure complete data trust through the entire system.

How it works

Any rows modified by a transaction in a ledger table are cryptographically SHA-256 hashed using a Merkle tree data structure that creates a root hash representing all rows in the transaction. The transactions that the database processes are then also SHA-256 hashed together through a Merkle tree data structure. The result is a root hash that forms a block. The block is then SHA-256 hashed through the root hash of the block, along with the root hash of the previous block as input to the hash function. That hashing forms a blockchain.

The root hashes in the [database ledger](#), also called [Database digests](#), contain the cryptographically hashed transactions and represent the state of the database. They can be periodically generated and stored outside the database in tamper-proof storage, such as [Azure Blob Storage configured with immutability policies](#), [Azure Confidential Ledger](#) or on-premises [Write Once Read Many \(WORM\) storage devices](#). Database digests are later used to verify the integrity of the database by comparing the value of the hash in the digest against the calculated hashes in database.

Ledger functionality is introduced to tables in two forms:

- [Updatable ledger tables](#), which allow you to update and delete rows in your tables.
- [Append-only ledger tables](#), which only allow insertions to your tables.

Both updatable ledger tables and append-only ledger tables provide tamper-evidence and digital forensics capabilities.

Updatable ledger tables

[Updatable ledger tables](#) are ideal for application patterns that expect to issue updates and deletions to tables in your database, such as system of record (SOR) applications. Existing data patterns for your application don't need to change to enable ledger functionality.

Updatable ledger tables track the history of changes to any rows in your database when transactions that perform updates or deletions occur. An updatable ledger table is a

system-versioned table that contains a reference to another table with a mirrored schema.

The other table is called the *history table*. The system uses this table to automatically store the previous version of the row each time a row in the ledger table is updated or deleted. The history table is automatically created when you create an updatable ledger table.

The values in the updatable ledger table and its corresponding history table provide a chronicle of the values of your database over time. A system-generated ledger view joins the updatable ledger table and the history table so that you can easily query this chronicle of your database.

For more information on updatable ledger tables, see [Create and use updatable ledger tables](#).

Append-only ledger tables

[Append-only ledger tables](#) are ideal for application patterns that are insert-only, such as security information and event management (SIEM) applications. Append-only ledger tables block updates and deletions at the API level. This blocking provides more tampering protection from privileged users such as system administrators and DBAs.

Because only insertions are allowed into the system, append-only ledger tables don't have a corresponding history table because there's no history to capture. As with updatable ledger tables, a ledger view provides insights into the transaction that inserted rows into the append-only table, and the user that performed the insertion.

For more information on append-only ledger tables, see [Create and use append-only ledger tables](#).

Ledger database

Ledger databases provide an easy solution for applications that require the integrity of all data to be protected for the entire lifetime of the database. A ledger database can only contain ledger tables. Creating regular tables (that are not ledger tables) is not supported. Each table is, by default, created as an [Updatable ledger table](#) with default settings, which makes creating such tables very easy. You configure a database as a ledger database at creation. Once created, a ledger database cannot be converted to a regular database. For more information, see [Configure a ledger database](#).

Database digests

The hash of the latest block in the database ledger is called the [database digest](#). It represents the state of all ledger tables in the database at the time that the block was generated.

When a block is formed, its associated database digest is published and stored outside the database in tamper-proof storage. Because database digests represent the state of the database at the time that they were generated, protecting the digests from tampering is paramount. An attacker who has access to modify the digests would be able to:

1. Tamper with the data in the database.
2. Generate the hashes that represent the database with those changes.
3. Modify the digests to represent the updated hash of the transactions in the block.

Ledger provides the ability to automatically generate and store the database digests in [immutable storage](#) or [Azure Confidential Ledger](#), to prevent tampering. Alternatively, users can manually generate database digests and store them in the location of their choice. Database digests are used for later verifying that the data stored in ledger tables hasn't been tampered with.

Ledger verification

The ledger feature doesn't allow modifying the content of ledger system views, append-only tables and history tables. However, an attacker or system administrator who has control of the machine can bypass all system checks and directly tamper with the data. For example, an attacker or system administrator can edit the database files in storage. Ledger can't prevent such attacks but guarantees that any tampering will be detected when the ledger data is verified.

The [ledger verification](#) process takes as input one or more previously generated database digests and recomputes the hashes stored in the database ledger based on the current state of the ledger tables. If the computed hashes don't match the input digests, the verification fails, indicating that the data has been tampered with. Ledger then reports all inconsistencies that it has detected.

Next steps

- [What is the database ledger](#)
- [Create and use append-only ledger tables](#)
- [Create and use updatable ledger tables](#)
- [Enable automatic digest storage](#)
- [Configure a ledger database](#)

- Verify a ledger table to detect tampering

See also

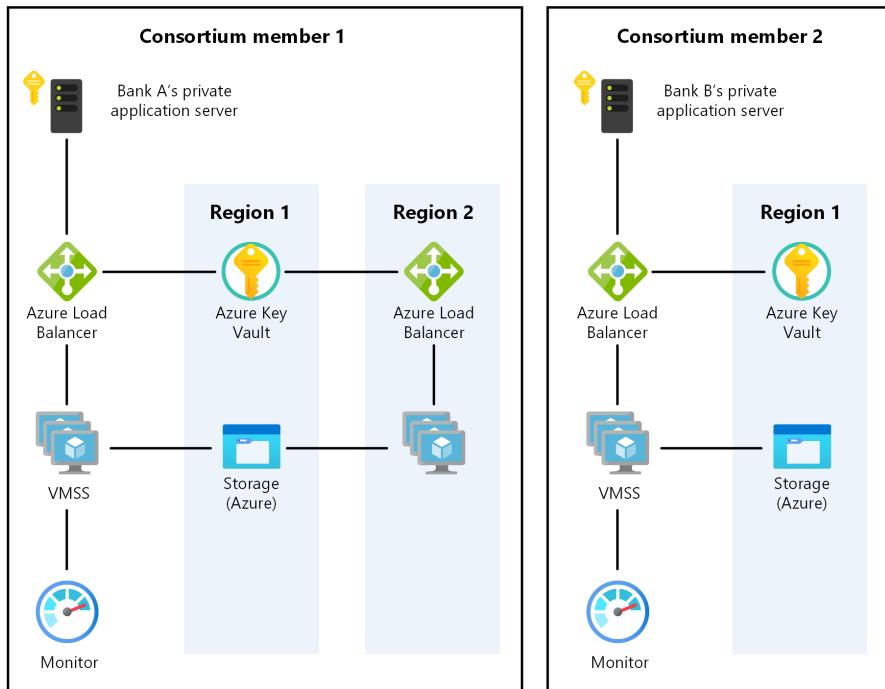
- [Bringing the power of blockchain to Azure SQL Database and SQL Server with ledger | Data Exposed](#)

Decentralized trust between banks

Azure Key Vault Azure Load Balancer Azure Monitor Azure Virtual Machines

This example will show you how Azure services such as virtual machine scale sets, Virtual Network, Key Vault, Storage, Load Balancer, and Monitor can be quickly provisioned for the deployment of an efficient private Corda network where member banks can establish their own nodes.

Architecture



Access the [Visio diagram](#) online, through Microsoft 365. Note that you must have a Visio license to access this diagram. Or, download a [Visio file](#) of this architecture.

This scenario covers the back-end components that are necessary to create a scalable, secure, and monitored private, enterprise distributed ledger technology (DLT) network within a consortium of two or more members. Details of how these components are provisioned (that is, within different subscriptions and resource groups), as well as the

connectivity requirements (that is, VPN or ExpressRoute), that are left for your consideration, are based on your organization's policy requirements.

Dataflow

1. Bank A creates/updates an individual's credit record by creating a transaction.
2. Data flows from Bank A's private application server to the [Azure Load Balancer](#), and then to a node VM on the virtual machine scale set.
3. A transaction proposal is created on the network ledger.
4. The transaction is committed to the ledger, when all the required signatures are gathered.
5. Bank B can read the credit record created by bank A by communicating with its own node.

Components

- [Virtual machines](#) , within virtual machine scale sets, provide the on-demand compute facility to host the node for the DLT network.
- [Azure Key Vault](#) is used as the secure storage facility for private keys.
- [Load Balancer](#) spreads communication requests to the VMs.
- [Azure Storage](#) hosts persistent network information and coordinates leasing.
- Application Insights (part of Azure Monitor) can be used to provide insight into available nodes, transactions per minute, and consortium members.

Alternatives

The Corda approach is chosen for this example because it is a good entry point for a consortium of organizations that want to create an environment where information can be exchanged and shared with one another easily in a trusted, decentralized, and easy to understand way. Other alternatives to Corda, such as Quorum or Hyperledger, can be considered too.

Scenario details

This example scenario is useful for banks or any other institutions that want to establish a trusted environment for information sharing without resorting to a centralized database. In this example, we will describe the scenario in the context of maintaining credit score information between banks, but the architecture can be applied to any scenario where a consortium of organizations wants to share information with one another without resorting to the use of a central system ran by one single party.

Traditionally, banks within a financial system rely on centralized sources, such as credit bureaus, for information on an individual's credit score and history. A centralized approach presents a concentration of operational risk and sometimes an unnecessary third party.

With DLTs (distributed ledger technology), a consortium of banks can establish a decentralized system that can be more efficient, less susceptible to attack, and serve as a new platform where innovative structures can be implemented to solve traditional challenges with privacy, speed, and cost.

Potential use cases

Other relevant use cases include:

- Movement of allocated budgets between different business units of a multinational corporation
- Cross-border payments
- Trade finance scenarios
- Loyalty systems involving different companies
- Supply chain ecosystems (such as manufacturing)

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

Availability

[Azure Monitor](#) is used to continuously monitor all components of the DLT network for issues to ensure availability.

Scalability

For general guidance on designing scalable solutions, see the [performance efficiency checklist](#) in the Azure Architecture Center.

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](#).

Azure Key Vault [↗](#) is used to easily store and manage the private keys.

For a production scenario, where a private network is desired, members can be connected to each other via VNet-to-VNet VPN gateway connections. The steps for configuring a VPN are included in the deployment section below.

For general guidance on designing secure solutions, see the [Azure Security Documentation](#).

Resiliency

The Corda network can itself provide some degree of resilience as the nodes can be deployed in different regions. Azure has options for deployments in over 54 regions worldwide. A DLT, such as the one in this scenario, provides unique and refreshing possibilities of cooperation to increase resilience. The resilience of the network is not just provided for by a single centralized party but all members of the consortium. DLT allows network resilience to be even more planned and deliberate.

For general guidance on designing resilient solutions, see [Designing reliable Azure applications](#).

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

To explore the cost of running this scenario, all of the services are pre-configured in the cost calculator. To see how the pricing would change for your particular use case, change the appropriate variables to match your expected performance and availability requirements.

We have provided three sample cost profiles based on the number of scale set VM instances that run your applications (the instances can reside in different regions).

- [Small ↗](#): this pricing example correlates to 2 VMs per month with monitoring turned off
- [Medium ↗](#): this pricing example correlates to 7 VMs per month with monitoring turned on
- [Large ↗](#): this pricing example correlates to 15 VMs per month with monitoring turned on

The above pricing is for one consortium member to start or join a DLT network. Typically in a consortium where there are multiple companies or organizations involved, each member will get their own Azure subscription.

Deploy this scenario

To deploy a pre-configured network of Corda nodes, review the [guide that is available in Corda's documentation](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Vito Chin](#) | Senior Cloud Solution Architect

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

Product documentation of Azure services:

- [Virtual machines](#)
- [About Azure Key Vault](#)
- [What is Azure Load Balancer?](#)
- [Introduction to Azure Storage](#)

Related resources

- [Data management in banking](#)
- [Patterns and implementations for a banking cloud transformation](#)
- [Zero-trust network for web applications with Azure Firewall and Application Gateway](#)

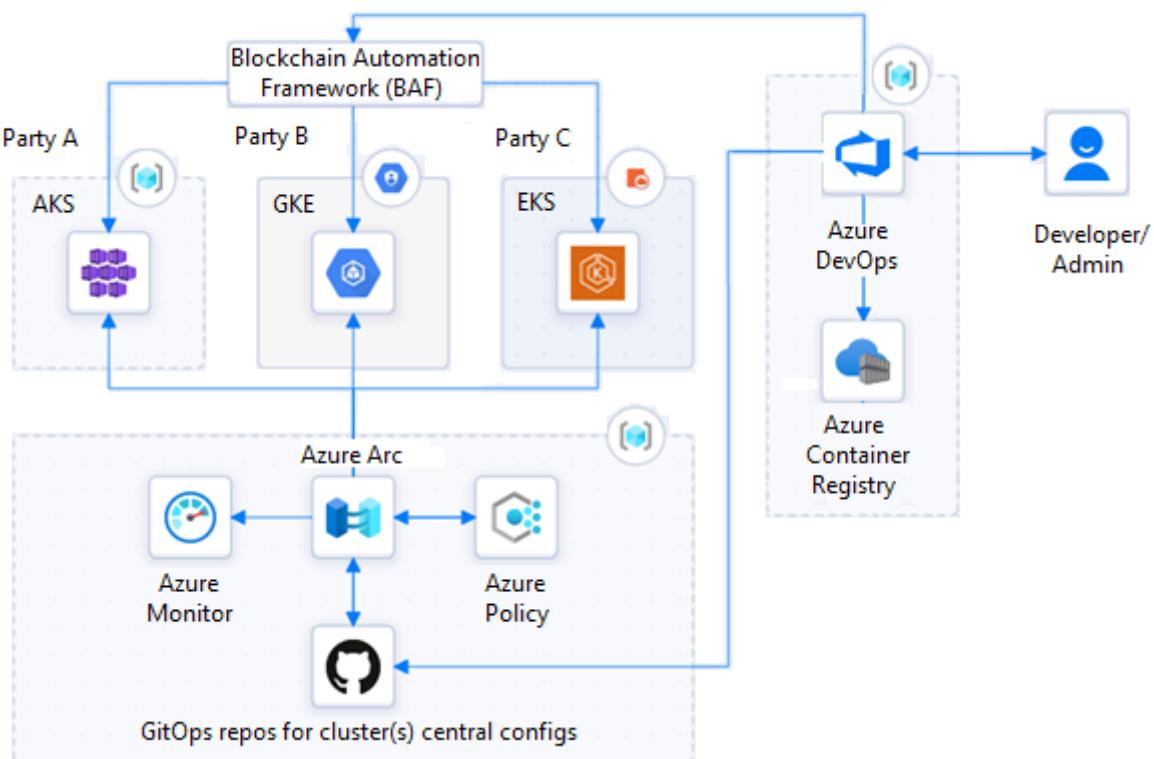
Multi-cloud blockchain DLT

Azure Arc Azure DevOps Azure Kubernetes Service (AKS)

This architecture combines the open-source Blockchain Automation Framework (BAF) and Azure Arc-enabled Kubernetes to work with multiparty DLTs and to build a cross-cloud blockchain solution.

Architecture

This solution provides a heterogeneous, multiparty, cloud-agnostic DLT network. Parties can host their nodes anywhere and still be part of the network.



Workflow

- [Kubernetes](#) is the standard infrastructure that hosts both the ledger and the application. This example assumes three managed Kubernetes clusters.
 - Party A uses [Azure Kubernetes Service \(AKS\)](#).
 - Party B uses [GCP Google Kubernetes Engine \(GKE\)](#).
 - Party C uses [Amazon Elastic Kubernetes Service \(EKS\)](#).

Each party hosts their nodes in a different location.

- BAF deploys the distributed network across the three cloud services.
- Azure Arc-enabled Kubernetes centrally manages and monitors all the Kubernetes clusters, with:
 - [GitOps-based cluster configuration deployment and management](#).
 - [Azure Monitor Container insights](#) monitoring.
 - [Azure Policy for Kubernetes](#) policy management.
- [Azure DevOps](#) provides application and infrastructure lifecycle management. An [Ansible Controller on an Azure Linux virtual machine \(VM\)](#) is the custom Azure DevOps continuous integration and continuous delivery (CI/CD) agent.
- [Azure Container Registry](#) stores and shares private, application-related container images. [Docker Registry](#) pulls ledger-specific images.

Components

- [Kubernetes](#) is the container-based infrastructure that hosts both the ledger and applications. This example assumes three managed Kubernetes clusters, one each in AKS, Amazon EKS, and GCP GKE. You can host your Kubernetes clusters in almost any public or private locations.
- The open-source [Blockchain Automation Framework \(BAF\)](#) is a way to deliver consistent, production-ready DLT networks on public and private cloud-based infrastructures. BAF supports [Quorum](#), [Corda](#), and [Hyperledger](#) DLTs.
- [Azure Arc](#) standardizes visibility, operations, and compliance across resources and locations by extending the Azure control plane.
- [Azure Arc-enabled Kubernetes](#) centrally manages Kubernetes clusters in any location. Azure Arc-enabled Kubernetes works with any Cloud Native Computing Foundation (CNCF)-certified Kubernetes cluster, including:
 - AKS engine on Azure
 - AKS engine on Azure Stack Hub
 - Amazon EKS
 - GCP GKE
 - VMware vSphere
- [Azure Monitor](#) is a comprehensive solution for collecting, analyzing, and acting on telemetry. [Azure Monitor Container insights](#) monitors the performance of container workloads deployed to Azure Arc-enabled Kubernetes clusters.

- [Azure Policy](#) helps enforce organizational standards and assess compliance at scale. [Azure Policy for Kubernetes](#) can manage and report on the compliance state of all Azure Arc-enabled Kubernetes clusters.
- [Azure Container Registry](#) can build, store, and manage container images and artifacts for all types of container deployments.
- [Azure DevOps](#) is a set of developer services providing comprehensive application and infrastructure lifecycle management. Azure DevOps includes work tracking, source control, build and CI/CD, package management, and testing solutions.

Alternatives

- [Ambassador API Gateway](#) manages cross-node communications, but you can use a cloud native API Gateway like Azure API Management over the internet. For more information, see [Deploy to Azure Kubernetes Service](#).
- You can also use [ExternalDNS](#) with [Azure DNS service](#).
- You can get Internet Protocol Security (IPSec) private connections with tools like [Submariner](#).

Scenario details

Blockchain and distributed ledger technology (DLT) networks are multiparty systems. Each party can have its own tools, methodology, and cloud provider. Some providers' public or private blockchain networks might have limited region availability, scalability, or network segregation.

The open-source [Blockchain Automation Framework \(BAF\)](#) is a consistent way to deploy production-ready DLTs across different public and private clouds. But while BAF can manage deployments, it doesn't provide central infrastructure management and monitoring. Although some cloud providers' blockchain services provide infrastructure management, they might require all parties to be in the same cloud or infrastructure.

To join forces and build a blockchain network, parties that use different cloud providers and infrastructures need a common management platform. This platform should offer standard visibility, operations, and compliance across a wide range of resources and locations, regardless of hosting infrastructure.

This article explores how the BAF and [Azure Arc-enabled Kubernetes](#) can build a cross-cloud blockchain solution that focuses on portability and control.

Potential use cases

This approach supports:

- Heterogeneous DLT deployments where separate organizations own and manage each node.
- Centralized DevOps, monitoring, and compliance management across multiparty networks.

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

For AKS best practices, see [Baseline architecture for an Azure Kubernetes Service \(AKS\) cluster](#). You can find similar guidance for other cloud providers.

Availability and scalability

Although Azure Arc can manage and monitor Kubernetes clusters, each cluster must independently implement scalability, high availability, and disaster recovery capabilities.

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](#).

BAF uses [HashiCorp Vault](#) for certificate and key storage. To use BAF, you need at least one Vault server. BAF recommends one Vault per organization for production-ready projects.

Cost optimization

Cost optimization is about looking at ways to reduce unnecessary expenses and improve operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

To estimate Azure resource costs, use the [Azure pricing calculator](#).

Deploy this scenario

1. For this example, create managed Kubernetes clusters in AKS, GKE, and EKS, and onboard the clusters to Azure Arc.
 - Connect an existing Kubernetes cluster to Azure Arc [↗](#).
 - Deploy EKS cluster and connect it to Azure Arc [↗](#).
 - Deploy GKE cluster and connect it to Azure Arc [↗](#).
2. Follow steps for installing and configuring [BAF prerequisites](#) [↗](#).
3. (Optional) [Create an Azure DevOps organization and project](#), and clone the BAF repo into the new Azure DevOps project.
4. (Optional) Create an [Ansible Controller VM](#) [↗](#) in Azure as the custom build agent to deploy BAF components.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Safi Ali](#) [↗](#) | Senior Cloud Solution Architect

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Azure Arc Jumpstart](#) [↗](#)
- [Deploy Hyperledger Fabric consortium on Azure Kubernetes Service](#)
- [CI/CD workflow using GitOps - Azure Arc-enabled Kubernetes](#)

Related resources

- [Baseline architecture for an Azure Kubernetes Service \(AKS\) cluster](#)
- [Blockchain workflow application](#) [↗](#)
- [Azure Arc hybrid management and deployment for Kubernetes clusters](#)
- [Containers and container orchestrators for AWS professionals](#)
- [Containers and container orchestrators for GCP professionals](#)

Supply chain management with Quorum Blockchain Service

Azure Virtual Machines

Azure Monitor

Azure Container Instances

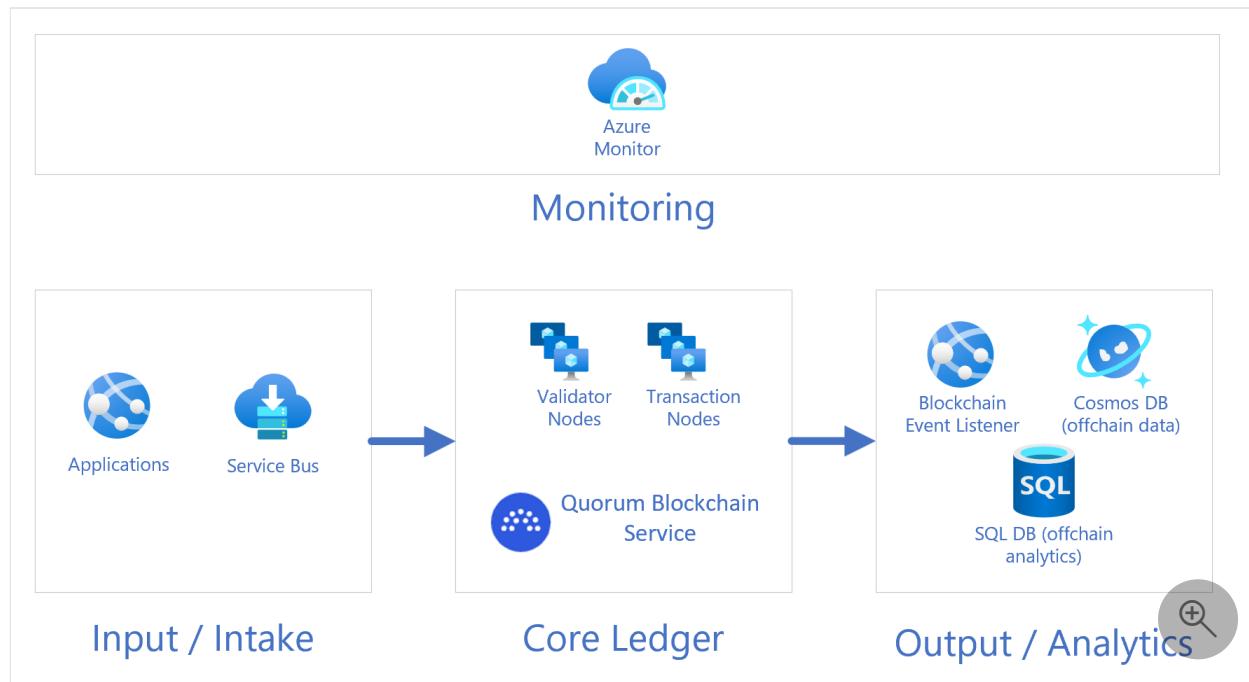
Azure Managed Applications

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

This article describes how to use Quorum Blockchain Service to track and trace physical assets, along with their metadata.

Architecture



Download a [Visio file](#) of this architecture.

Dataflow

Quorum Blockchain Service (QBS) is a managed ledger service from [ConsenSys](#). It allows organizations to run their own blockchain network without having to deal with infrastructure management.

The following are features of QBS:

- Uses standard Ethereum technology with privacy enhancements.
- Supports open-source tools such as [Truffle](#) for developing and managing lifecycle of smart contracts.
- Supports event listening models for off-chain storage and integration with open-source tools such as Ethlogger from Splunk.
- Provides extensions for Visual Studio Code (VS Code).
 - Users can create and debug smart contracts using [Solidity](#) in VS Code.
 - Users can also deploy assets to Quorum via the [TruffleSuite](#) using VS Code Extension.
- Provides management APIs and supports monitoring and logging of blockchain nodes with integration to [Azure Monitor](#).

Getting started with QBS:

QBS is currently accessible as a Private offering on Azure. You can access QBS using one of the following ways:

- Sign-up directly on the [ConsenSys Quorum Blockchain Service](#) page.
- Inquire directly with [QBS Support](#).

The elements of the architectural diagram are explained below:

- **Input/Intake**

Input to the application uses existing Azure services. For instance, a traditional web application that runs in [Azure App Services](#), serves as an input from interactive users. Additionally, the need for batch type inputs driven by business logic not directly interactive by users is handled using Azure messaging services, primarily being Service Bus.

- **Core ledger**

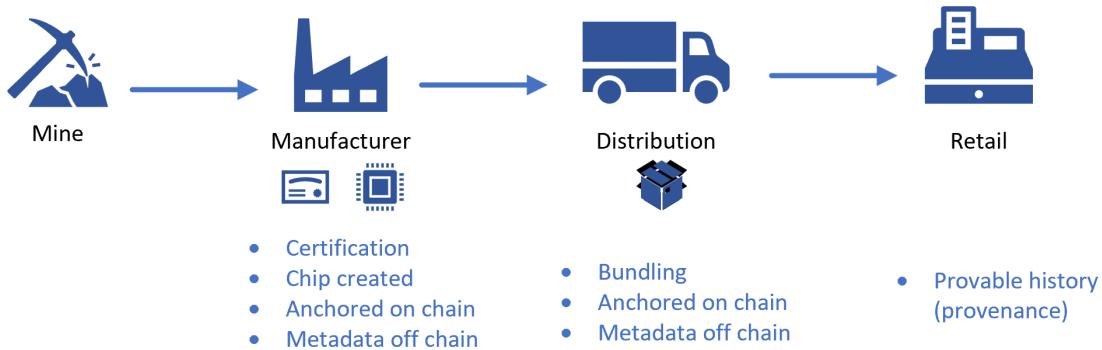
Input that is destined to be persisted in the underlying ledger for the solution is handled by blockchain, using QBS. A set of validator nodes are created as part of QBS however the interaction from the input application is with the transaction nodes. Azure Virtual Machines or Virtual Machine Scale Sets that uses agent pool in [Azure Kubernetes Service](#) provide compute and storage/persistence needed by QBS.

- **Output/Analytics**

As data continues to exist in the ledger in QBS, the need for both analytics and triggers for existing non-blockchain based systems is required in most cases. A listener supporting tooling such as Ethlogger can be configured to run in Azure and send data

to off-chain data technologies, such as Azure Cosmos DB or Azure SQL. The Ethlogger component is run on either [Azure Virtual Machines](#) or using container in [Azure Container Instances](#).

Dataflow in action



The above diagram represents a simplified version of the stages in supply chain for chip manufacturing. The stages involved are mining, manufacturing, distribution and retail, where material change hands many times in progression. For a company that values sustainability and ethical responsibility, it becomes important to track and trace the source of truth and quality details of the materials. In the past, each participant used to put the data in their own data technology stack and synchronize data to other parties.

The data flow mapping of the above diagram is described below:

- **Mine/manufacturer**

After raw material is received by the manufacturer, input is created. Then, the certification of the material is checked and ownership of the component is established with on-chain data in the ledger and off-chain data in traditional storage.

- **Distribution**

Once the components are manufactured, they are bundled for distribution. The bundle is used in the ledger with QBS to establish provenance, such as a palette. This is done using a combination of on-chain data in the ledger and off-chain data in traditional storage.

- **Retail**

The ledger system of the QBS helps organizations get the history of low-level component packed in a larger product. This system takes the ownership of the products and allows to refer to its history.

Components

- [Azure App Service](#) is an HTTP-based service to host web applications, REST APIs, and mobile back ends. The App Service is a web application/API that allows interactive use of the data by users in the supply chain. In this article, App Services are used to read or reference the data in Input/Intake stage and all other stages.
- [Azure Service Bus](#) is a fully managed enterprise message broker with message queues and publish-subscribe topics (in a namespace). In this article, Azure Service Bus is used by manufacturers to inject data into the blockchain ledger or off-chain data store.
- [Azure Managed Applications](#) enable you to offer cloud solutions that are easy for consumers to deploy and operate.
 - Publish a managed application to the Azure market place to be available for all the customers.
 - Publish a managed application to an internal catalog to be available to only your company's users.
- Event Listener: In this scenario, [Ethlogger](#) is utilized to send data to off-chain data technologies, such as Azure Cosmos DB or Azure SQL.
- [Azure Cosmos DB](#) is a fully managed NoSQL database for modern app development that provides single-digit millisecond response times, automatic and instant scalability, and guaranteed speed at any scale. In this scenario Azure Cosmos DB is a data store used to house the output of events raised on the blockchain such as state changes that are used by an analytics solution or simple reporting.
- [Azure SQL Database](#) is a fully managed platform as a service (PaaS) database engine that handles most of the database management functions such as upgrading, patching, backups, and monitoring without user involvement. In this scenario Azure SQL Database is a data store used to house the output of events raised on the blockchain such as state changes that will ultimately be used by an analytics solution or simple reporting. Additionally, by utilizing the [ledger](#) feature of Azure SQL Database, the integrity of output of events from the blockchain replicated to the SQL database can be maintained.
- [Azure Monitor](#) provides a comprehensive solution for collecting, analyzing, and acting on telemetry from cloud and on-premises environments. In this scenario Azure Monitor provides availability and performance data of the architecture components. QBS is integrated with Azure Monitor to provide additional telemetry data on blockchain nodes.

Scenario details

Blockchain technology helps organizations track and trace the product life cycle and meet regulatory, financial, ethical, and sustainability requirements.

The need for the blockchain technology started with organizations trying to understand more about the products, like:

- Where did the product originate from?
- How was a product built?
- What were the processes followed to build a product?

Building and maintaining the electronic data interface (EDI) across suppliers to track and trace parts of a product is not worthwhile for some organizations. By using ConsenSys Quorum Blockchain Service (QBS) with App Services, Azure Key Vault, Service Bus, and Azure Cosmos DB, organizations can track and trace a tokenized version of the product's physical assets and relevant metadata to meet requirements.

Potential use cases

With QBS powered by Azure services, an organization can build solutions that can track and provide an immutable history of the product parts and their metadata, such as quality certificates. Through a shared ledger these certificates cover carbon footprint and percentage of recycled component used.

Blockchain technology allows the logical data flow model to use with physical components deployed with each participant. This reduces the need for participants to trust a single partner. While this could be built using traditional centralized components, this is usually not the case with supply chain workloads. Supply chain workloads have their own systems that require isolation from others.

This solution is ideal for the sustainability, manufacturing, and energy/environment industries.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Samrat Adhikari](#) | Senior Manager, Technology Solutions Delivery
- [Cale Teeter](#) | Senior Software Engineer

Other contributor:

- [Lavanya Kasturi](#) | Technical Writer

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [Azure App Service overview](#)
- [Introduction to blockchain on Azure](#)

Related resources

- [Multiparty computing architecture design](#)
- [Multicloud blockchain DLT](#)

Supply chain track and trace

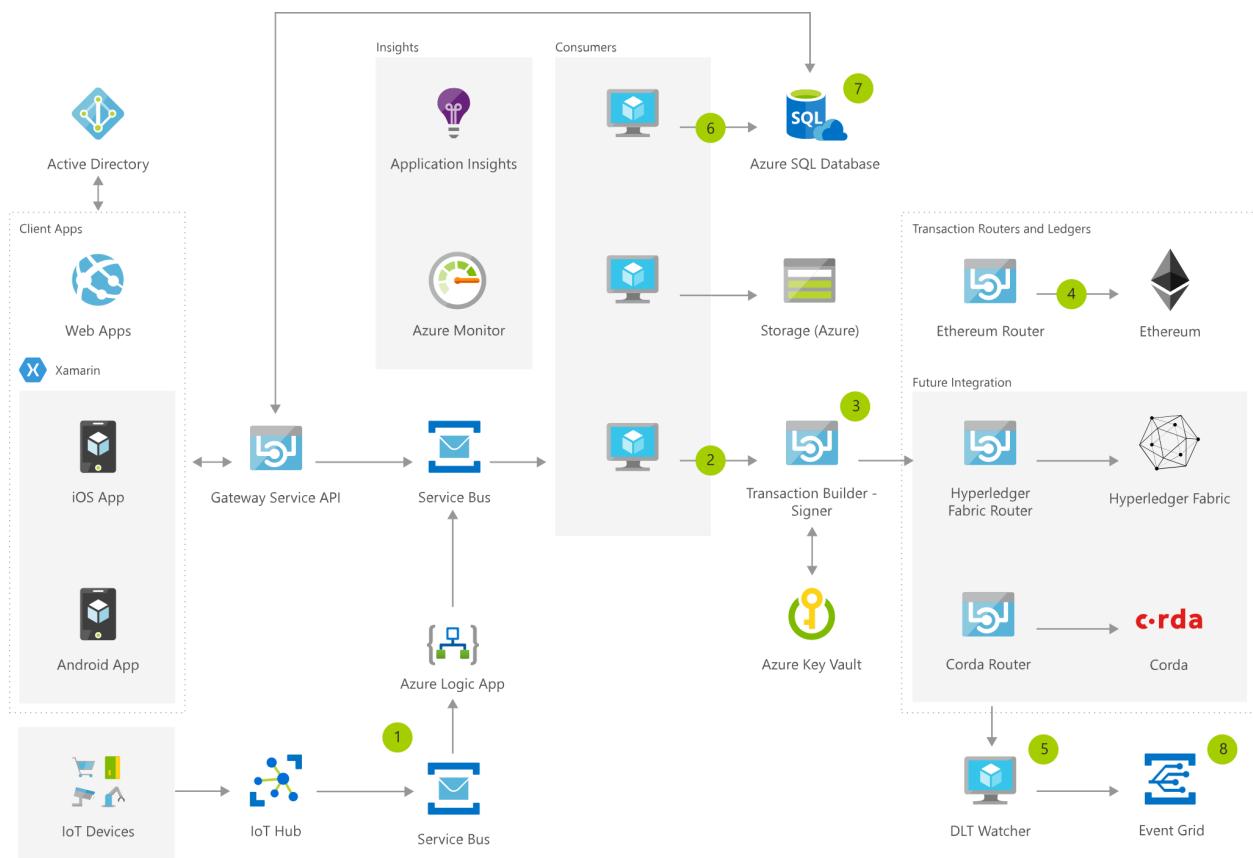
Azure IoT Hub

💡 Solution ideas

This article is a solution idea. If you'd like us to expand the content with more information, such as potential use cases, alternative services, implementation considerations, or pricing guidance, let us know by providing [GitHub feedback](#).

A common blockchain pattern is the IoT-enabled monitoring of an asset, as it moves along a multi-party supply chain.

Architecture



[Download an SVG of this architecture.](#)

IoT devices communicate with IoT Hub. IoT Hub as a route configured that will send specific messages to a Service Bus associated with that route. The message is still in the native format for the device and needs to be translated to the format used by Azure Blockchain Workbench.

An Azure Logic App performs that transformation. It's triggered when a new message is added to the Service Bus associated with the IoT hub, it then transforms the message and delivers it to the Service Bus used to deliver messages to Azure Blockchain Workbench.

The first service bus effectively serves as an "Outbox" for IoT Hub and the second one serves as an "Inbox" for Azure Blockchain Workbench.

Dataflow

1. IoT devices communicate with IoT Hub. IoT Hub as a route configured that will send specific messages to a Service Bus associated with that route. The message is still in the native format for the device and needs to be translated to the format used by Azure Blockchain Workbench. An Azure Logic App performs that transformation. It's triggered when a new message is added to the Service Bus associated with the IoT hub, it then transforms the message and delivers it to the Service Bus used to deliver messages to Azure Blockchain Workbench. The first service bus effectively serves as an "Outbox" for IoT Hub and the second one serves as an "Inbox" for Azure Blockchain Workbench.
2. DLT Consumer fetches the data from the message broker (Service Bus) and sends data to Transaction Builder - Signer.
3. Transaction Builder builds and signs the transaction.
4. The signed transaction gets routed to the Blockchain (Private Ethereum Consortium Network).
5. DLT Watcher gets confirmation of the transaction commitment to the Blockchain and sends the confirmation to the message broker (Service Bus).
6. DB consumers send confirmed blockchain transactions to off-chain databases (Azure SQL Database).
7. Information analyzed and visualized using tools such as Power BI by connecting to off-chain database (Azure SQL Database).
8. Events from the ledger are delivered to Event Grid and Service Bus for use by downstream consumers. Examples of "downstream consumers" include logic apps, functions or other code that is designed to take action on the events. For example, an Azure Function could receive an event and then place that in a datastore such as SQL Server.

Components

- Application Insights: Detect issues, diagnose crashes, and track usage in your web app with Application Insights. Make informed decisions throughout the development lifecycle.

- [Web Apps](#) : Quickly create and deploy mission critical web apps at scale
- [Storage Accounts](#) : Durable, highly available, and massively scalable cloud storage
- [Virtual Machines](#) : Provision virtual machines for Ubuntu, Red Hat, and more
- [Microsoft Entra ID](#) : Synchronize on-premises directories and enable single sign-on
- [Azure SQL Database](#) is a relational database service that lets you rapidly create, extend, and scale relational applications into the cloud.
- [Azure Monitor](#) : Highly granular and real-time monitoring data for any Azure resource.
- [Service Bus](#) : Connect across private and public cloud environments
- [Event Grid](#) : Get reliable event delivery at massive scale

Scenario details

Potential use cases

A great example of this pattern is the refrigerated transportation of perishable goods like food or pharmaceuticals where certain compliance rules must be met throughout the duration of the transportation process. In this scenario, an initiating counterparty (such as a retailer) specifies contractual conditions, such as a required humidity and temperature range, that the custodians on the supply chain must adhere to.

At any point, if the device takes a temperature or humidity measurement that is out of range, the smart contract state will be updated to indicate that it's out of compliance, by recording a transaction on the blockchain and triggering remediating events downstream.

Deploy this scenario

- [Deploy to Azure](#)

Next steps

- [Find run-time exceptions with Application Insights](#)
- [Create a blockchain app with Azure Blockchain Workbench](#)
- [Azure Storage on Blockchain Workbench](#)
- [Azure and Linux Virtual Machines](#)
- [Blockchain Workbench API app registration](#)

- [Blockchain Workbench Database](#)
- [Log Analytics Tutorial](#)
- [Service Bus on Blockchain Workbench](#)
- [Event Notifications on Blockchain Workbench](#)

Related resources

- [Real-time asset tracking and management using IoT Central](#)
- [Getting started with Azure IoT solutions](#)

High-performance computing (HPC) on Azure

Article • 09/15/2023

Introduction to HPC

<https://www.youtube-nocookie.com/embed/rKURT32faJk> ↗

High-performance computing (HPC), also called "big compute", uses a large number of CPU or GPU-based computers to solve complex mathematical tasks.

Many industries use HPC to solve some of their most difficult problems. These include workloads such as:

- Genomics
- Oil and gas simulations
- Finance
- Semiconductor design
- Engineering
- Weather modeling

How is HPC different on the cloud?

One of the primary differences between an on-premises HPC system and one in the cloud is the ability for resources to dynamically be added and removed as they're needed. Dynamic scaling removes compute capacity as a bottleneck and instead allow customers to right size their infrastructure for the requirements of their jobs.

The following articles provide more detail about this dynamic scaling capability.

- [Big Compute Architecture Style](#)
- [Autoscaling best practices](#)

Implementation checklist

As you're looking to implement your own HPC solution on Azure, ensure you're reviewed the following topics:

- ✓ Choose the appropriate [architecture](#) based on your requirements
- ✓ Know which [compute](#) options is right for your workload

- ✓ Identify the right **storage** solution that meets your needs
- ✓ Decide how you're going to **manage** all your resources
- ✓ Optimize your **application** for the cloud
- ✓ Secure your Infrastructure

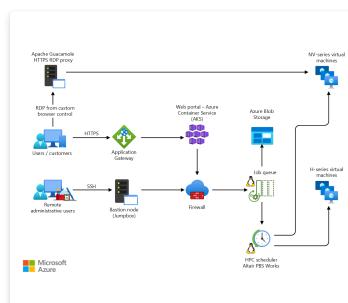
Infrastructure

There are many infrastructure components that are necessary to build an HPC system. Compute, storage, and networking provide the underlying components, no matter how you choose to manage your HPC workloads.

Example HPC architectures

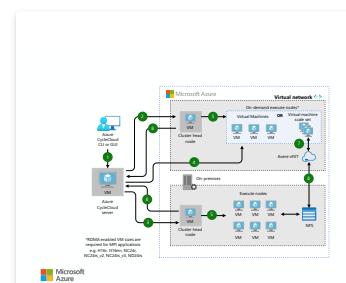
There are many different ways to design and implement your HPC architecture on Azure. HPC applications can scale to thousands of compute cores, extend on-premises clusters, or run as a 100% cloud-native solution.

The following scenarios outline a few of the common ways HPC solutions are built.



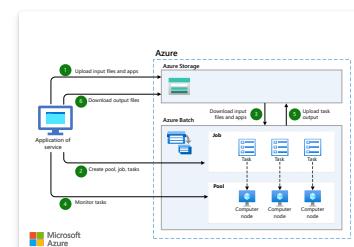
Computer-aided engineering services on Azure

Provide a software-as-a-service (SaaS) platform for computer-aided engineering (CAE) on Azure.



Computationa fluid dynamics (CFD) simulations on Azure

Execute computational fluid dynamics (CFD) simulations on Azure.



3D video rendering on Azure

Run native HPC workloads in Azure using the Azure Batch service

Compute

Azure offers a range of sizes that are optimized for both CPU & GPU intensive workloads.

CPU-based virtual machines

- [Linux VMs](#)
- [Windows VMs](#)

GPU-enabled virtual machines

N-series VMs feature NVIDIA GPUs designed for compute-intensive or graphics-intensive applications including artificial intelligence (AI) learning and visualization.

- [Linux VMs](#)
- [Windows VMs](#)

Storage

Large-scale Batch and HPC workloads have demands for data storage and access that exceed the capabilities of traditional cloud file systems. There are many solutions that manage both the speed and capacity needs of HPC applications on Azure:

- [Avere vFXT](#) ↗ for faster, more accessible data storage for high-performance computing at the edge
- [Azure NetApp Files](#)
- [Storage Optimized Virtual Machines](#)
- [Blob, table, and queue storage](#)
- [Azure SMB File storage](#)

For more information comparing Lustre, GlusterFS, and BeeGFS on Azure, review the [Parallel Files Systems on Azure](#) e-book and the [Lustre on Azure](#) ↗ blog.

Networking

H16r, H16mr, A8, and A9 VMs can connect to a high throughput back-end RDMA network. This network can improve the performance of tightly coupled parallel applications running under Microsoft Message Passing Interface better known as MPI or Intel MPI.

- [RDMA Capable Instances](#)

- Virtual Network
- ExpressRoute

Management

Do-it-yourself

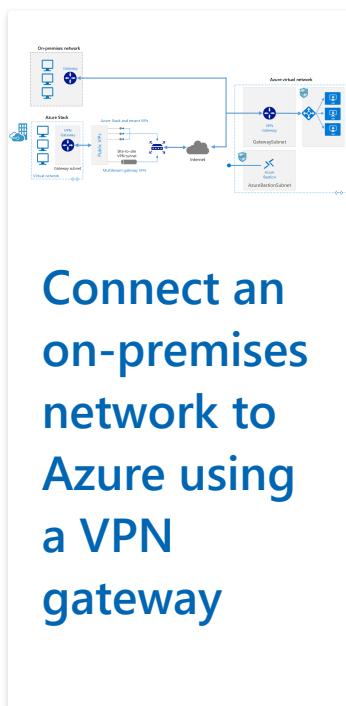
Building an HPC system from scratch on Azure offers a significant amount of flexibility, but it is often very maintenance intensive.

1. Set up your own cluster environment in Azure virtual machines or [Virtual Machine Scale Sets](#).
2. Use Azure Resource Manager templates to deploy leading [workload managers](#), infrastructure, and [applications](#).
3. Choose HPC and GPU [VM sizes](#) that include specialized hardware and network connections for MPI or GPU workloads.
4. Add [high-performance storage](#) for I/O-intensive workloads.

Hybrid and cloud Bursting

If you have an existing on-premises HPC system that you'd like to connect to Azure, there are several resources to help get you started.

First, review the [Options for connecting an on-premises network to Azure](#) article in the documentation. From there, you can find additional information on these connectivity options:



This reference architecture shows how to extend an on-premises network to Azure, using a site-to-site virtual private network (VPN).

ExpressRoute connections use a private, dedicated connection through a third-party connectivity provider. The private connection extends your on-premises network into Azure.

e with VPN failover

Implement a highly available and secure site-to-site network architecture that spans an Azure virtual network and an on-premises network connected using ExpressRoute with VPN gateway failover.

Once network connectivity is securely established, you can start using cloud compute resources on-demand with the bursting capabilities of your existing [workload manager](#).

Marketplace solutions

There are many workload managers offered in the [Azure Marketplace](#).

- [RogueWave CentOS-based HPC](#)
- [SUSE Linux Enterprise Server for HPC](#)
- [TIBCO DataSynapse GridServer](#)
- [Azure Data Science VM for Windows and Linux](#)
- [D3View](#)
- [UberCloud](#)

Azure Batch

[Azure Batch](#) is a platform service for running large-scale parallel and HPC applications efficiently in the cloud. Azure Batch schedules compute-intensive work to run on a managed pool of virtual machines, and can automatically scale compute resources to meet the needs of your jobs.

SaaS providers or developers can use the Batch SDKs and tools to integrate HPC applications or container workloads with Azure, stage data to Azure, and build job execution pipelines.

In Azure Batch all the services are running on the Cloud, the image below shows how the architecture looks with Azure Batch, having the scalability and job schedule

configurations running in the Cloud while the results and reports can be sent to your on-premises environment.



Azure CycleCloud

[Azure CycleCloud](#) Provides the simplest way to manage HPC workloads using any scheduler (like Slurm, Grid Engine, HPC Pack, HTCondor, LSF, PBS Pro, or Symphony), on Azure

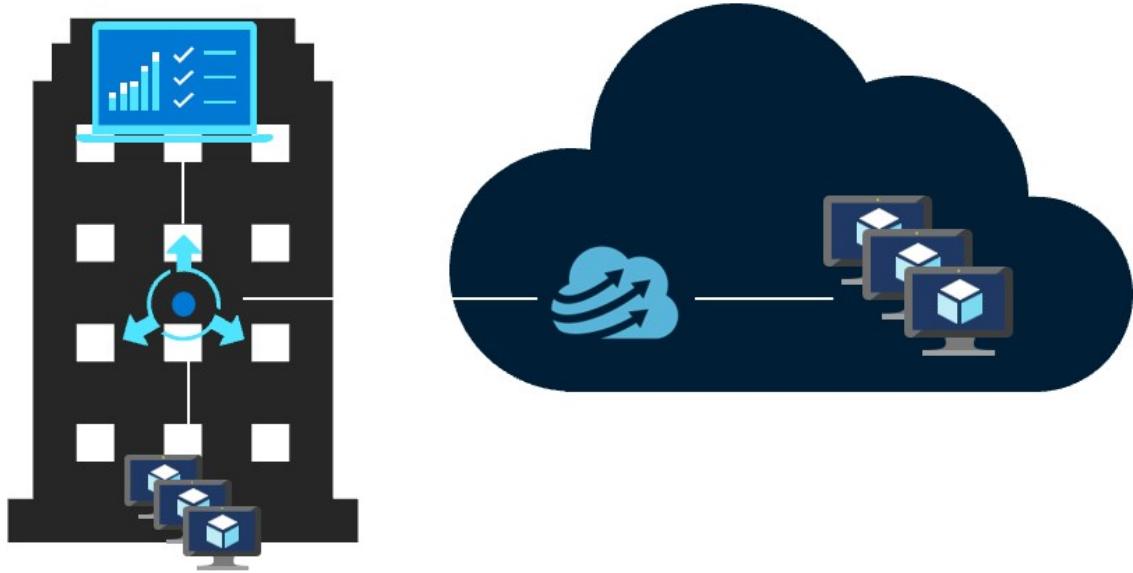
CycleCloud allows you to:

- Deploy full clusters and other resources, including scheduler, compute VMs, storage, networking, and cache
- Orchestrate job, data, and cloud workflows
- Give admins full control over which users can run jobs, as well as where and at what cost
- Customize and optimize clusters through advanced policy and governance features, including cost controls, Active Directory integration, monitoring, and reporting
- Use your current job scheduler and applications without modification
- Take advantage of built-in autoscaling and battle-tested reference architectures for a wide range of HPC workloads and industries

Hybrid / cloud bursting model

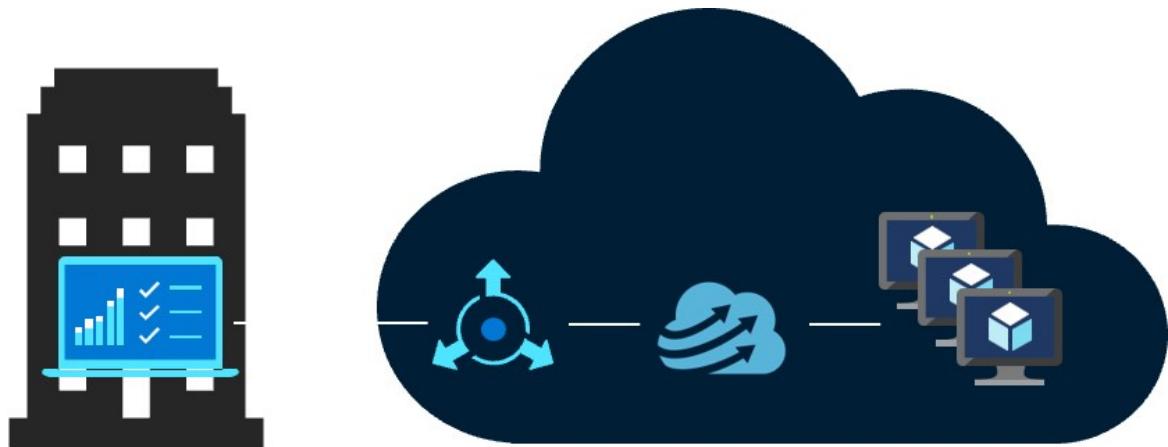
In this Hybrid example diagram, we can see clearly how these services are distributed between the cloud and the on-premises environment. Having the opportunity to run

jobs in both workloads.



Cloud native model

The cloud native model example diagram below, shows how the workload in the cloud will handle everything while still conserving the connection to the on-premises environment.



Comparison chart

[] [Expand table](#)

Feature	Azure Batch	Azure CycleCloud
Scheduler	Batch APIs and tools and command-line scripts in the Azure portal (Cloud Native).	Use standard HPC schedulers such as Slurm, PBS Pro, LSF, Grid Engine, and HTCondor, or extend CycleCloud autoscaling plugins to work with your own scheduler.

Feature	Azure Batch	Azure CycleCloud
Compute Resources	Software as a Service Nodes – Platform as a Service	Platform as a Service Software – Platform as a Service
Monitor Tools	Azure Monitor	Azure Monitor, Grafana
Customization	Custom image pools, Third Party images, Batch API access.	Use the comprehensive RESTful API to customize and extend functionality, deploy your own scheduler, and support into existing workload managers
Integration	Synapse Pipelines, Azure Data Factory, Azure CLI	Built-In CLI for Windows and Linux
User type	Developers	Classic HPC administrators and users
Work Type	Batch, Workflows	Tightly coupled (Message Passing Interface/MPI).
Windows Support	Yes	Varies, depending on scheduler choice

Workload managers

The following are examples of cluster and workload managers that can run in Azure infrastructure. Create stand-alone clusters in Azure VMs or burst to Azure VMs from an on-premises cluster.

- Alces Flight Compute
- [TIBCO DataSynapse GridServer](#) ↗
- [Bright Cluster Manager](#) ↗
- [IBM Spectrum Symphony and Symphony LSF](#) ↗
- [Altair PBS Works](#) ↗
- [Rescale](#) ↗
- [Altair Grid Engine](#) ↗
- [Microsoft HPC Pack](#)
 - [HPC Pack for Windows](#)
 - [HPC Pack for Linux](#)

Containers

Containers can also be used to manage some HPC workloads. Services like the Azure Kubernetes Service (AKS) makes it simple to deploy a managed Kubernetes cluster in Azure.

- Azure Kubernetes Service (AKS)
- Container Registry

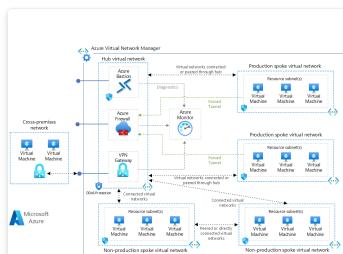
Cost management

Managing your HPC cost on Azure can be done through a few different ways. Ensure you've reviewed the [Azure purchasing options](#) to find the method that works best for your organization.

Security

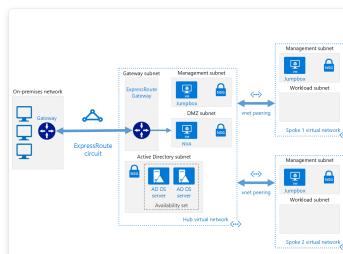
For an overview of security best practices on Azure, review the [Azure Security Documentation](#).

In addition to the network configurations available in the [Cloud Bursting](#) section, you can implement a hub/spoke configuration to isolate your compute resources:



Implement a hub-spoke network topology in Azure

The hub is a virtual network (VNet) in Azure that acts as a central point of connectivity to your on-premises network. The spokes are VNs that peer with the hub, and can be used to isolate workloads.



Implement a hub-spoke network topology with shared services in Azure

This reference architecture builds on the hub-spoke reference architecture to include shared services in the hub that can be

consumed by all spokes.

HPC applications

Run custom or commercial HPC applications in Azure. Several examples in this section are benchmarked to scale efficiently with additional VMs or compute cores. Visit the [Azure Marketplace](#) for ready-to-deploy solutions.

ⓘ Note

Check with the vendor of any commercial application for licensing or other restrictions for running in the cloud. Not all vendors offer pay-as-you-go licensing. You might need a licensing server in the cloud for your solution, or connect to an on-premises license server.

Engineering applications

- [Altair RADIOSS](#)
- [ANSYS CFD](#)
- [MATLAB Distributed Computing Server](#)
- [StarCCM+](#)

Graphics and rendering

- [Autodesk Maya, 3ds Max, and Arnold on Azure Batch](#)

AI and deep learning

- [Microsoft Cognitive Toolkit](#)

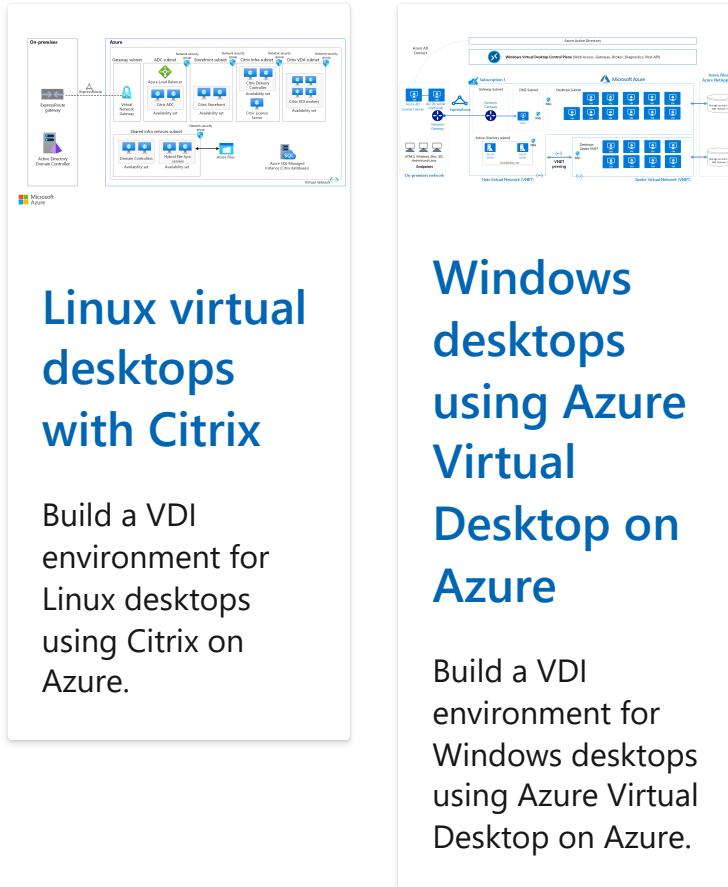
MPI providers

- [Microsoft MPI](#)

Remote visualization

Run GPU-powered virtual machines in Azure in the same region as the HPC output for the lowest latency, access, and to visualize remotely through Azure Virtual Desktop, Citrix, or VMware Horizon.

- GPU-optimized virtual machine sizes
- Configure GPU acceleration for Azure Virtual Desktop



Performance benchmarks

- Compute benchmarks

Customer stories

There are many customers who have seen great success by using Azure for their HPC workloads. You can find a few of these customer case studies below:

- [AXA Global P&C](#)
- [Axioma](#)
- [d3View](#)
- [EFS](#)
- [Hymans Robertson](#)
- [MetLife](#)

- Microsoft Research ↗
- Milliman ↗
- Mitsubishi UFJ Securities International ↗
- NeuroInitiative ↗
- Schlumberger ↗
- Towers Watson ↗

Other important information

- Ensure your [vCPU quota](#) has been increased before attempting to run large-scale workloads.

Next steps

For the latest announcements, see the following resources:

- [Microsoft HPC and Batch team blog](#)
- Visit the [Azure blog](#) ↗.

Microsoft Batch Examples

These tutorials will provide you with details on running applications on Microsoft Batch:

- [Get started developing with Batch](#)
- [Use Azure Batch code samples](#) ↗
- [Use low-priority VMs with Batch](#)
- [Use compute-intensive VMs in Batch pools](#)

Related resources

- [Big compute architecture style](#)
- [Hybrid HPC in Azure with HPC Pack](#)
- [HPC cluster deployed in the cloud](#)

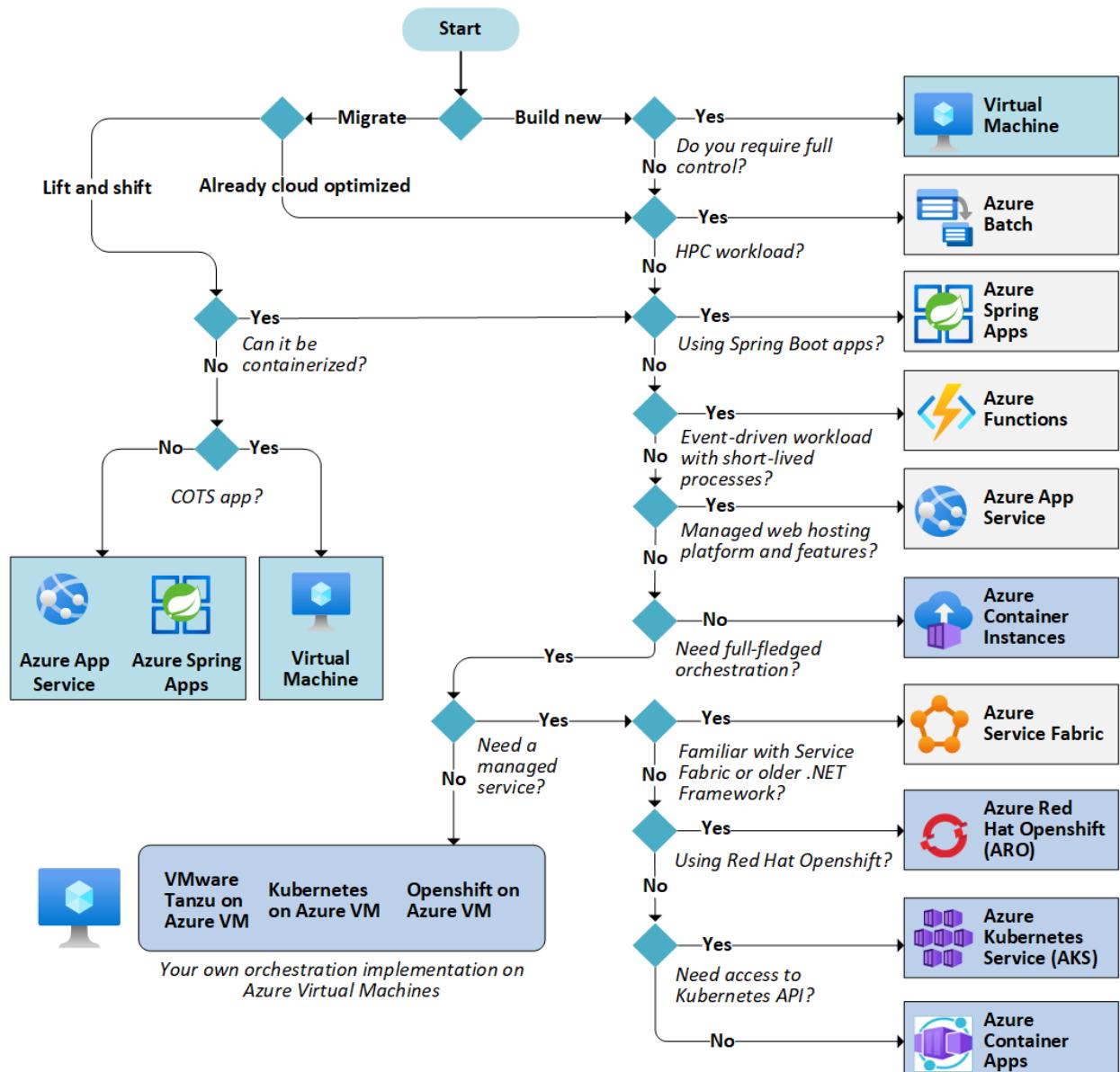
Choose an Azure compute service

Azure App Service Azure Kubernetes Service (AKS)

Azure offers many ways to host your application code. The term *compute* refers to the hosting model for the resources that your application runs on. This article helps choose a compute service for your application.

Choose a candidate service

Use the following flowchart to select a candidate compute service.



Container exclusive services

- Azure Batch
- Azure Functions
- Azure App Service
- Azure Spring Apps
- Azure Service Fabric
- Kubernetes on Azure VM
- OpenShift on Azure VM

Container compatible services

- Azure Batch
- Azure Functions
- Azure Service Fabric
- Azure Spring Apps
- Azure App Service

This diagram refers to two migration strategies:

- **Lift and shift:** A strategy for migrating a workload to the cloud without redesigning the application or making code changes. It's also called *rehosting*. For more information, see [Azure migration and modernization center](#) ↗.
- **Cloud optimized:** A strategy for migrating to the cloud by refactoring an application to take advantage of cloud-native features and capabilities.

The output from this flowchart is your starting point. Next, evaluate the service to see if it meets your needs.

This article includes several tables that can help you choose a service. The initial candidate from the flowchart might be unsuitable for your application or workload. In that case, expand your analysis to include other compute services.

If your application consists of multiple workloads, evaluate each workload separately. A complete solution can incorporate two or more compute services.

Understand the basic features

If you're not familiar with the Azure service selected in the previous section, see this overview documentation:

- [Azure Virtual Machines](#): A service where you deploy and manage virtual machines (VMs) inside an Azure virtual network.
- [Azure App Service](#): A managed service for hosting web apps, mobile app back ends, RESTful APIs, or automated business processes.
- [Azure Functions](#): A managed function as a service.
- [Azure Kubernetes Service \(AKS\)](#): A managed Kubernetes service for running containerized applications.
- [Azure Container Apps](#): A managed service built on Kubernetes, which simplifies the deployment of containerized applications in a serverless environment.
- [Azure Container Instances](#): This service is a fast and simple way to run a container in Azure. You don't have to provision any VMs or adopt a higher-level service.
- [Azure Red Hat OpenShift](#): A fully managed OpenShift cluster for running containers in production with Kubernetes.
- [Azure Spring Apps](#): A managed service designed and optimized for hosting Spring Boot apps.
- [Azure Service Fabric](#): A distributed systems platform that can run in many environments, including Azure or on-premises.
- [Azure Batch](#): A managed service for running large-scale parallel and high-performance computing (HPC) applications.

Understand the hosting models

For hosting models, cloud services fall into three categories:

- **Infrastructure as a service (IaaS):** Lets you provision VMs along with the associated networking and storage components. Then you can deploy whatever software and applications you want onto those VMs. This model is the closest to a traditional on-premises environment. Microsoft manages the infrastructure. You still manage the VMs.
- **Platform as a service (PaaS):** Provides a managed hosting environment where you can deploy your application without needing to manage VMs or networking resources. Azure App Service and Azure Container Apps are PaaS services.
- **Functions as a service (FaaS):** Lets you deploy your code to the service, which automatically runs it. Azure Functions is a FaaS service.

ⓘ Note

Azure Functions is an [Azure serverless](#) compute offering. To see how this service compares with other Azure serverless offerings, such as Logic Apps, which provides serverless workflows, see [Choose the right integration and automation services in Azure](#).

There's a spectrum from IaaS to pure PaaS. For example, Azure VMs can automatically scale by using virtual machine scale sets. This capability isn't strictly a PaaS, but it's the type of management feature found in PaaS.

There's a tradeoff between control and ease of management. IaaS gives the most control, flexibility, and portability, but you have to provision, configure, and manage the VMs and network components you create. FaaS services automatically manage nearly all aspects of running an application. PaaS falls somewhere in between.

[+] [Expand table](#)

Service	Application composition	Density	Minimum number of nodes	State management	Web hosting
Azure Virtual	Agnostic	Agnostic	1 ²	Stateless or stateful	Agnostic

Service	Application composition	Density	Minimum number of nodes	State management	Web hosting
Machines					
Azure App Service	Applications, containers	Multiple apps per instance by using App Service plan	1	Stateless	Built in
Azure Functions	Functions, containers	Serverless ¹	Serverless ¹	Stateless or stateful ⁶	Not applicable
Azure Kubernetes Service	Containers	Multiple containers per node	3 ³	Stateless or stateful	Agnostic
Azure Container Apps	Containers	Serverless	Serverless	Stateless or stateful	Agnostic
Azure Container Instances	Containers	No dedicated instances	No dedicated nodes	Stateless	Agnostic
Azure Red Hat OpenShift	Containers	Multiple containers per node	6 ⁵	Stateless or stateful	Agnostic
Azure Spring Apps	Applications, microservices	Multiple apps per service instance	2	Stateless	Built in
Azure Service Fabric	Services, guest executables, containers	Multiple services per VM	5 ³	Stateless or stateful	Agnostic

Service	Application composition	Density	Minimum number of nodes	State management	Web hosting
Azure Batch	Scheduled jobs	Multiple apps per VM	1 ⁴	Stateless	No

Notes

1. If you're using a Consumption plan. For an App Service plan, functions run on the VMs allocated for your App Service plan. See [Choose the correct service plan for Azure Functions](#).
2. Higher service-level agreement (SLA) with two or more instances.
3. Recommended for production environments.
4. Can scale down to zero after job completes.
5. Three for primary nodes and three for worker nodes.
6. When using [Durable Functions](#).

Networking

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

Service	Virtual network integration	Hybrid connectivity
Azure Virtual Machines	Supported	Supported
Azure App Service	Supported ¹	Supported ²
Azure Functions	Supported ¹	Supported ³
Azure Kubernetes Service	Supported	Supported
Azure Container Apps	Supported	Supported
Azure Container Instances	Supported	Supported
Azure Red Hat OpenShift	Supported	Supported

Service	Virtual network integration	Hybrid connectivity
Azure Spring Apps	Supported	Supported
Azure Service Fabric	Supported	Supported
Azure Batch	Supported	Supported

Notes

1. Requires App Service Environment.
2. Use [Azure App Service Hybrid Connections](#).
3. Requires App Service plan or [Azure Functions Premium plan](#).

DevOps

[] [Expand table](#)

Service	Local debugging	Programming model	Application update
Azure Virtual Machines	Agnostic	Agnostic	No built-in support
Azure App Service	IIS Express, others ¹	Web and API applications, WebJobs for background tasks	Deployment slots
Azure Functions	Visual Studio or Azure Functions CLI	Serverless, event-driven	Deployment slots
Azure Kubernetes Service	Minikube, Docker, others	Agnostic	Rolling update
Azure Container Apps	Local container runtime	Agnostic	Revision management

Service	Local debugging	Programming model	Application update
Azure Container Instances	Local container runtime	Agnostic	Not applicable
Azure Red Hat OpenShift	Minikube, Docker, others	Agnostic	Rolling update
Azure Spring Apps	Visual Studio Code, IntelliJ, Eclipse	Spring Boot, Steeltoe	Rolling upgrade, blue-green deployment
Azure Service Fabric	Local node cluster	Guest executable, Service model, Actor model, Containers	Rolling upgrade (per service)
Azure Batch	Not supported	Command-line application	Not applicable

Notes

1. Options include IIS Express for ASP.NET or node.js (iisnode), PHP web server, Azure Toolkit for IntelliJ, and Azure Toolkit for Eclipse. App Service also supports remote debugging of deployed web app.

Scalability

[\[\]](#) Expand table

Service	Autoscaling	Load balancer	Scale limit ³
Azure Virtual Machines	Virtual machine scale sets	Azure Load Balancer	Platform image: 1,000 nodes per scale set, Custom image: 600 nodes per scale set
Azure App Service	Built-in service	Integrated	30 instances, 100 with App Service Environment

Service	Autoscaling	Load balancer	Scale limit ³
Azure Functions	Built-in service	Integrated	200 instances per function app
Azure Kubernetes Service	Pod autoscaling ¹ , cluster autoscaling ²	Azure Load Balancer or Azure Application Gateway	5,000 nodes when using Uptime SLA
Azure Container Apps	Scaling rules ⁴	Integrated	5 environments per region, 20 container apps per environment, 30 replicas per container app
Azure Container Instances	Not supported	No built-in support	20 container groups per subscription (default limit)
Azure Red Hat OpenShift	Pod autoscaling, cluster autoscaling	Azure Load Balancer or Azure Application Gateway	60 nodes per cluster (default limit)
Azure Spring Apps	Built-in service	Integrated	500 app instances in Standard
Azure Service Fabric	Virtual machine scale sets	Azure Load Balancer	100 nodes per virtual machine scale set
Azure Batch	Not applicable	Azure Load Balancer	20 core limit (default limit)

Notes

1. See [Autoscale pods](#).
2. See [Automatically scale a cluster to meet application demands on Azure Kubernetes Service](#).
3. See [Azure subscription and service limits, quotas, and constraints](#).
4. See [Set scaling rules in Azure Container Apps](#).

Availability

 [Expand table](#)

Service	SLA	Multiregion failover
Azure Virtual Machines	SLA for Virtual Machines	Azure Traffic Manager, Azure Front Door, and cross-region Azure Load Balancer
Azure App Service	SLA for App Service	Azure Traffic Manager and Azure Front Door
Azure Functions	SLA for Functions	Azure Traffic Manager and Azure Front Door
Azure Kubernetes Service	SLA for AKS	Azure Traffic Manager, Azure Front Door, and Multiregion Cluster
Azure Container Apps	SLA for Container Apps	Azure Traffic Manager and Azure Front Door
Azure Container Instances	SLA for Container Instances	Azure Traffic Manager and Azure Front Door
Azure Red Hat OpenShift	SLA for Azure Red Hat OpenShift	Azure Traffic Manager and Azure Front Door
Azure Spring Apps	SLA for Azure Spring Apps	Azure Traffic Manager, Azure Front Door, and Multiregion Cluster
Azure Service Fabric	SLA for Service Fabric	Azure Traffic Manager, Azure Front Door, and cross-region Azure Load Balancer
Azure Batch	SLA for Batch	Not applicable

For guided learning on service guarantees, see [Core Cloud Services - Azure architecture and service guarantees](#).

Security

Review and understand the available security controls and visibility for each service:

- Azure Windows virtual machine
- Azure Linux virtual machine
- Azure App Service
- Azure Functions
- Azure Kubernetes Service
- Azure Container Instances
- Azure Spring Apps
- Azure Service Fabric
- Azure Batch

Other criteria

[] [Expand table](#)

Service	TLS	Cost	Suitable architecture styles
Azure Virtual Machines	Configured in VM	Windows  , Linux 	N-tier, big compute (HPC)
Azure App Service	Supported	App Service pricing 	Web-queue-worker
Azure Functions	Supported	Functions pricing 	Microservices, event-driven architecture
Azure Kubernetes Service	Ingress controller	AKS pricing 	Microservices, event-driven architecture
Azure Container Apps	Ingress controller	Container Apps pricing 	Microservices, event-driven architecture
Azure Container Instances	Use sidecar container	Container Instances pricing 	Microservices, task automation, batch jobs

Service	TLS	Cost	Suitable architecture styles
Azure Red Hat OpenShift	Supported	Azure Red Hat OpenShift pricing ↗	Microservices, event-driven architecture
Azure Spring Apps	Supported	Azure Spring Apps pricing ↗	Spring Boot, microservices
Azure Service Fabric	Supported	Service Fabric pricing ↗	Microservices, event-driven architecture
Azure Batch	Supported	Batch pricing ↗	Big compute (HPC)

Consider limits and cost

Along with the previous comparison tables, do a more detailed evaluation of the following aspects of the candidate service:

- [Service limits](#)
- [Cost ↗](#)
- [SLA ↗](#)
- [Regional availability ↗](#)

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors:

- [Ayobami Ayodeji ↗](#) | Senior Program Manager
- [Jelle Druyts ↗](#) | Principal Service Engineer
- [Martin Gjoshevski ↗](#) | Senior Service Engineer
- [Phil Huang ↗](#) | Senior Cloud Solution Architect
- [Julie Ng ↗](#) | Senior Service Engineer
- [Paolo Salvatori ↗](#) | Principal Service Engineer

To see nonpublic LinkedIn profiles, sign in to LinkedIn.

Next steps

Core Cloud Services - Azure compute options. This Learn module explores how compute services can solve common business needs.

Related resources

- [Choose an Azure compute option for microservices](#)
- [Lift and shift to containers with Azure App Service](#)
- [Technology choices for Azure solutions](#)

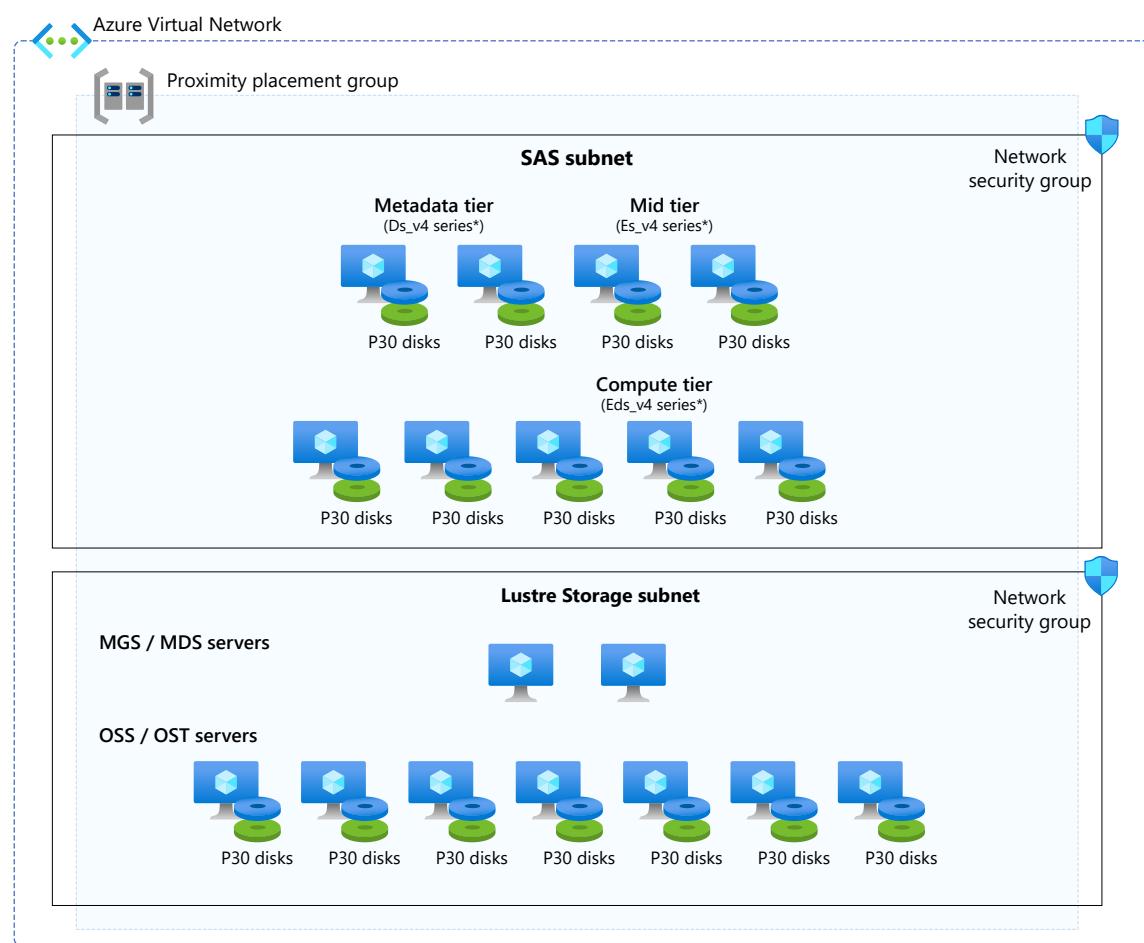
SAS on Azure architecture

Azure Virtual Machines

Azure Virtual Network

This solution runs SAS analytics workloads on Azure. The guidance covers various deployment scenarios. For instance, multiple versions of SAS are available. You can run SAS software on self-managed virtual machines (VMs). You can also deploy container-based versions by using Azure Kubernetes Service (AKS).

Architecture



Download a [Visio file](#) of this architecture.

Workflow

SAS Azure deployments typically contain three layers:

- An API or visualization tier. Within this layer:
 - The metadata tier gives client apps access to metadata on data sources, resources, servers, and users.
 - Web apps provide access to intelligence data in the mid tier.
- A compute platform, where SAS servers process data.
- A storage tier that SAS uses for permanent storage. Popular choices on Azure are:
 - Lustre
 - IBM Spectrum Scale
 - Network File System (NFS)

An Azure Virtual Network isolates the system in the cloud. Within that network:

- A proximity placement group reduces latency between VMs.
- Network security groups protect SAS resources from unwanted traffic.

Prerequisites

Before deploying a SAS workload, ensure the following components are in place:

- A sizing recommendation from a SAS sizing team
- A SAS license file
- Access to a resource group for deploying your resources
- A [virtual central processing unit \(vCPU\) subscription quota](#) that takes into account your sizing document and VM choice
- Access to a secure Lightweight Directory Access Protocol (LDAP) server

Scenario details

Along with discussing different implementations, this guide also aligns with [Microsoft Azure Well-Architected Framework](#) tenets for achieving excellence in the areas of cost, DevOps, resiliency, scalability, and security. But besides using this guide, consult with a SAS team for additional validation of your particular use case.

[As partners](#), Microsoft and SAS are working to develop a roadmap for organizations that innovate in the cloud. Both companies are committed to ensuring high-quality deployments of SAS products and solutions on Azure.

Introduction to SAS

SAS analytics software provides a suite of services and tools for drawing insights from data and making intelligent decisions. SAS platforms fully support its solutions for areas such as data management, fraud detection, risk analysis, and visualization. SAS offers these primary platforms, which Microsoft has validated:

- SAS Grid 9.4
- SAS Viya

The following architectures have been tested:

- SAS Grid 9.4 on Linux
- SAS 9 Foundation
- SAS Viya 3.5 with symmetric multiprocessing (SMP) and massively parallel processing (MPP) architectures on Linux
- SAS Viya 2020 and up with an MPP architecture on AKS

This guide provides general information for running SAS on Azure, not platform-specific information. These guidelines assume that you host your own SAS solution on Azure in your own tenant. SAS doesn't host a solution for you on Azure. For more information on the Azure hosting and management services that SAS provides, see [SAS Managed Application Services](#).

Recommendations

Consider the points in the following sections when designing your implementation.

SAS documentation provides requirements per core, meaning per physical CPU core. But Azure provides vCPU listings. On the VMs that we recommend for use with SAS, there are two vCPU for every physical core. As a result, to calculate the value of a vCPU requirement, use half the core requirement value. For instance, a physical core requirement of 150 MBps translates to 75 MBps per vCPU. For more information on Azure computing performance, see [Azure compute unit \(ACU\)](#).

Note

If you're scaling up and persisting data in a single-node SAS deployment (and not to an externalized file system), the [SAS documentation](#) recommends bandwidth of at least 150 MB/s. To achieve this bandwidth, you need to stripe multiple P30 Premium (or larger) disks.

Operating systems

Linux works best for running SAS workloads. SAS supports 64-bit versions of the following operating systems:

- Red Hat 7 or newer
- SUSE Linux Enterprise Server (SLES) 12.2
- Oracle Linux 6 or later

For more information about specific SAS releases, see the [SAS Operating System support matrix](#). In environments that use multiple machines, it's best to run the same version of Linux on all machines. Azure doesn't support Linux 32-bit deployments.

To optimize compatibility and integration with Azure, start with an operating system image from Azure Marketplace. If you use a custom image without additional configurations, it can degrade SAS performance.

Kernel issues

When choosing an operating system, be aware of a soft lockup issue that affects the entire Red Hat 7.x series. It occurs in these kernels:

- Linux 3.x kernels
- Versions earlier than 4.4

A problem with the [memory and I/O management of Linux and Hyper-V](#) causes the issue. When it comes up, the system logs contain entries like this one that mention a non-maskable interrupt (NMI):

Console

```
Message from syslogd@ronieuwe-sas-e48-2 at Sep 13 08:26:08
kernel:NMI watchdog: BUG: soft lockup - CPU#12 stuck for 22s! [swapper/12:0]
```

Another issue affects older versions of Red Hat. Specifically, it can happen in versions that meet these conditions:

- Have Linux kernels that precede 3.10.0-957.27.2
- Use non-volatile memory express (NVMe) drives

When the system experiences high memory pressure, the generic Linux NVMe driver may not allocate sufficient memory for a write operation. As a result, the system reports a soft lockup that stems from an actual deadlock.

Upgrade your kernel to avoid both issues. Alternatively, try this possible workaround:

- Set `/sys/block/nvme0n1/queue/max_sectors_kb` to `128` instead of using the default value, `512`.
- Change this setting on each NVMe device in the VM and on *each* VM boot.

Run these commands to adjust that setting:

shell

```
# cat /sys/block/nvme0n1/queue/max_sectors_kb
512
# echo 128 >/sys/block/nvme0n1/queue/max_sectors_kb
# cat /sys/block/nvme0n1/queue/max_sectors_kb
128
```

VM sizing recommendations

SAS deployments often use the following VM SKUs:

Edsv5-series

VMs in the Edsv5-series are the default SAS machines for Viya and Grid. They offer these features:

- Constrained cores. With many machines in this series, you can constrain the VM vCPU count.
- A good CPU-to-memory ratio.
- A high-throughput locally attached disk. I/O speed is important for folders like `SASWORK` and the Cloud Analytics Services (CAS) cache, `CAS_CACHE`, that SAS uses for temporary files.

If the Edsv5-series VMs are unavailable, it's recommended to use the prior generation. The [Edsv4-series VMs](#) have been tested and perform well on SAS workloads.

Ebsv5-series

In some cases, the locally attached disk doesn't have sufficient storage space for `SASWORK` or `CAS_CACHE`. To get a larger working directory, use the [Ebsv5-series of VMs](#) with premium attached disks. These VMs offer these features:

- Same specifications as the Edsv5 and Esv5 VMs
- High throughput against remote attached disk, up to 4 GB/s, giving you as large a `SASWORK` or `CAS_CACHE` as needed at the I/O needs of SAS.

If the Edsv5-series VMs offer enough storage, it's better to use them as they're more cost efficient.

M-series

Many workloads use M-series VMs, including:

- SAS Programming Runtime Environment (SPRE) implementations that use a Viya approach to software architecture.
- Certain SAS Grid workloads.

M-series VMs offer these features:

- Constrained cores
- Up to 3.8 TiB of memory, suited for workloads that use a large amount of memory
- High throughput to remote disks, which works well for the `SASWORK` folder when the locally available disk is insufficient

Ls-series

Certain I/O heavy environments should use [Lsv2-series](#) or [Lsv3-series](#) VMs. In particular, implementations that require fast, low latency I/O speed and a large amount of memory benefit from this type of machine. Examples include systems that make heavy use of the `SASWORK` folder or `CAS_CACHE`.

ⓘ Note

SAS optimizes its services for use with the Intel Math Kernel Library (MKL).

- With math-heavy workloads, avoid VMs that don't use Intel processors: the Lsv2 and Lsv3.
- When selecting an AMD CPU, validate how the MKL performs on it.

⚠ Warning

When possible, avoid using Lsv2 VMs. Please use the Lsv3 VMs with Intel chipsets instead.

With Azure, you can scale SAS Viya systems on demand to meet deadlines:

- By increasing the compute capacity of the node pool.

- By using the AKS [Cluster Autoscaler](#) to add nodes and scale horizontally.
- By temporarily scaling up infrastructure to accelerate a SAS workload.

ⓘ Note

When scaling computing components, also consider scaling up storage to avoid storage I/O bottlenecks.

With Viya 3.5 and Grid workloads, Azure doesn't support horizontal or vertical scaling at the moment. Viya 2022 supports horizontal scaling.

Network and VM placement considerations

SAS workloads are often chatty. As a result, they can transfer a significant amount of data. With all SAS platforms, follow these recommendations to reduce the effects of chatter:

- Deploy SAS and storage platforms on the same virtual network. This approach also avoids incurring peering costs.
- Place SAS machines in a [proximity placement group](#) to reduce latency between nodes.
- When possible, deploy SAS machines and VM-based data storage platforms in the same proximity placement group.
- Deploy SAS and storage appliances in the same availability zone to avoid cross-zone latency. If you can't confirm your solution components are deployed in the same zone, contact Azure support.

SAS has specific fully qualified domain name (FQDN) requirements for VMs. Set machine FQDNs correctly, and ensure that domain name system (DNS) services are working. You can set the names with Azure DNS. You can also edit the `hosts` file in the `etc` configuration folder.

ⓘ Note

Turn on accelerated networking on all nodes in the SAS deployment. When you turn this feature off, performance suffers significantly.

To turn on accelerated networking on a VM, follow these steps:

1. Run this command in the Azure CLI to deallocate the VM:

```
az vm deallocate --resource-group <resource_group_name> --name <VM_name>
```

2. Turn off the VM.

3. Run this command in the CLI:

```
az network nic update -n <network_interface_name> -g  
<resource_group_name> --accelerated-networking true
```

When you migrate data or interact with SAS in Azure, we recommend that you use one of these solutions to connect on-premises resources to Azure:

- An [Azure ExpressRoute](#) circuit
- A [virtual private network \(VPN\)](#)

For production SAS workloads in Azure, ExpressRoute provides a private, dedicated, and reliable connection that offers these advantages over a site-to-site VPN:

- Higher speed
- Lower latency
- Tighter security

Be aware of latency-sensitive interfaces between SAS and non-SAS applications.

Consider moving data sources and sinks close to SAS.

Identity management

SAS platforms can use local user accounts. They can also use a secure LDAP server to validate users. We recommend running a domain controller in Azure. Then use the domain join feature to properly manage security access. If you haven't set up domain controllers, consider deploying [Microsoft Entra Domain Services \(Microsoft Entra Domain Services\)](#). When you use the domain join feature, ensure machine names don't exceed the 15-character limit.

Note

In some environments, there's a requirement for on-premises connectivity or shared datasets between on-premises and Azure-hosted SAS environments. In these situations, we strongly recommended deploying a domain controller in Azure.

The Microsoft Entra Domain Services forest creates users that can authenticate against Microsoft Entra devices but not on-premises resources and vice versa.

Data sources

SAS solutions often access data from multiple systems. These data sources fall into two categories:

- SAS datasets, which SAS stores in the `SASDATA` folder
- Databases, which SAS often places a heavy load on

For best performance:

- Position data sources as close as possible to SAS infrastructure.
- Limit the number of network hops and appliances between data sources and SAS infrastructure.

Note

If you can't move data sources close to SAS infrastructure, avoid running analytics on them. Instead, run extract, transform, load (ETL) processes first and analytics later. Take the same approach with data sources that are under stress.

Permanent remote storage for SAS Data

SAS and Microsoft have tested a series of data platforms that you can use to host SAS datasets. The SAS blogs document the results in detail, including performance characteristics. The tests include the following platforms:

- [Sycomp Storage Fueled by IBM Spectrum Scale](#), which uses General Parallel File System (GPFS)
- [EXAScaler Cloud by DataDirect Networks \(DDN\)](#), which is based on the Lustre file system
- [Azure NetApp Files](#), which supports NFS file-storage protocols

SAS offers performance-testing scripts for the Viya and Grid architectures. The [SAS forums](#) provide documentation on tests with scripts on these platforms.

Sycomp Storage Fueled by IBM Spectrum Scale (GPFS)

For information about how Sycomp Storage Fueled by IBM Spectrum Scale meets performance expectations, see [SAS review of Sycomp for SAS Grid](#).

For sizing, Sycomp makes the following recommendations:

- Provide one GPFS scale node per eight cores with a configuration of 150 MBps per core.
- Use a minimum of five P30 drives per instance.

DDN EXAScaler Cloud (Lustre)

DDN, which acquired Intel's Lustre business, provides EXAScaler Cloud, which is based on the Lustre parallel file system. The solution is available in the Azure Marketplace as part of the DDN EXAScaler Cloud umbrella. Designed for data-intensive deployment, it provides high throughput at low cost.

Tests show that DDN EXAScaler can run SAS workloads in a parallel manner [🔗](#). DDN recommends running this command on all client nodes when deploying EXAScaler or Lustre:

shell

```
lctl set_param mdc.*.max_rpcs_in_flight=128 osc.*.max_pages_per_rpc=16M
osc.*.max_rpcs_in_flight=16 osc.*.max_dirty_mb=1024
llite.*.max_read_ahead_mb=2048 osc.*.checksums=0
llite.*.max_read_ahead_per_file_mb=256
```

Azure NetApp Files (NFS)

SAS tests have [validated NetApp performance for SAS Grid](#) [🔗](#). Specifically, testing shows that Azure NetApp Files is a viable primary storage option for SAS Grid clusters of up to 32 physical cores across multiple machines. When [NetApp provided optimizations and Linux features](#) [🔗](#) are used, Azure NetApp Files can be the primary option for clusters up to 48 physical cores across multiple machines.

Consider the following points when using this service:

- Azure NetApp Files works well with Viya deployments. Don't use Azure NetApp Files for the CAS cache in Viya, because the write throughput is inadequate. If possible, use your VM's local ephemeral disk instead.
- On SAS 9 Foundation with Grid 9.4, the performance of Azure NetApp Files with SAS for `SASDATA` files is good for clusters up to 32 physical cores. This goes up to 48 cores when [tuning](#) [🔗](#) applied.
- To ensure good performance, select at least a Premium or Ultra storage tier [service level](#) when deploying Azure NetApp Files. You can choose the Standard service level for very large volumes. Consider starting with the Premium level and

switching to Ultra or Standard later. Service level changes can be done online, without disruption or data migrations.

- Read and write [performance are different](#) for Azure NetApp Files. Write throughput for SAS hits limits at around 1600MiB/s while read throughput goes beyond that, to around 4500MiB/s. If you need continuous high write throughput, Azure NetApp Files may not be a good fit.

Other data sources

SAS platforms support various data sources:

- An [Azure Data Lake Storage account](#) that uses a [hierarchical namespace](#)
- [Azure Synapse Analytics](#)
- Apache Hadoop and Hive on [Azure HDInsight](#)
- SQL Server
- SQL Server using Open Database Connectivity (ODBC)

Considerations

These considerations implement the pillars of the Azure Well-Architected Framework, which is a set of guiding tenets that can be used to improve the quality of a workload. For more information, see [Microsoft Azure Well-Architected Framework](#).

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](#).

The output of your SAS workloads can be one of your organization's critical assets. SAS output provides insight into internal efficiencies and can play a critical role in reporting strategy. It's important, then, to secure access to your SAS architecture. To achieve this goal, use secure authentication and address network vulnerabilities. Use encryption to protect all data moving in and out of your architecture.

Azure delivers SAS by using an infrastructure as a service (IaaS) cloud model. Microsoft builds security protections into the service at the following levels:

- Physical datacenter
- Physical network
- Physical host
- Hypervisor

Carefully evaluate the services and technologies that you select for the areas above the hypervisor, such as the guest operating system for SAS. Make sure to provide the proper security controls for your architecture.

SAS currently doesn't fully support [Microsoft Entra ID](#). For authentication into the visualization layer for SAS, you can use Microsoft Entra ID. But for back-end authorization, use a strategy that's similar to on-premises authentication. When managing IaaS resources, you can use Microsoft Entra ID for authentication and authorization to the Azure portal. When using Microsoft Entra Domain Services, you can't authenticate guest accounts. Guest attempts to sign in will fail.

Use [network security groups](#) to filter network traffic to and from resources in your [virtual network](#). With these groups, you can define rules that grant or deny access to your SAS services. Examples include:

- Giving access to CAS worker ports from on-premises IP address ranges.
- Blocking access to SAS services from the internet.

You can use [Azure Disk Encryption](#) for encryption within the operating system. This solution uses the DM-Crypt feature of Linux. But we currently don't recommend using Azure Disk Encryption. It can severely degrade performance, especially when you use `SASWORK` files locally.

[Server-side encryption \(SSE\) of Azure Disk Storage](#) protects your data. It also helps you meet organizational security and compliance commitments. With Azure managed disks, SSE encrypts the data at rest when persisting it to the cloud. This behavior applies by default to both OS and data disks. You can use platform-managed keys or your own keys to encrypt your managed disk.

Protect your infrastructure

Control access to the Azure resources that you deploy. Every Azure subscription has a [trust relationship](#) with a Microsoft Entra tenant. Use [Azure role-based access control \(Azure RBAC\)](#) to grant users within your organization the correct permissions to Azure resources. Grant access by assigning Azure roles to users or groups at a certain scope. The scope can be a subscription, a resource group, or a single resource. Make sure to [audit all changes to infrastructure](#).

Manage remote access to your VMs through [Azure Bastion](#). Don't expose any of these components to the internet:

- VMs
- Secure Shell Protocol (SSH) ports

- Remote Desktop Protocol (RDP) ports

Deploy this scenario

It's best to deploy workloads using an infrastructure as code (IaC) process. SAS workloads can be sensitive to misconfigurations that often occur in manual deployments and reduce productivity.

When building your environment, see quickstart reference material at [CoreCompete SAS 9 or Viya on Azure](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Roeland Nieuwenhuis](#) | Principal Cloud Solution Architect
- [David Baumgarten](#) | Senior Cloud Solution Architect

Other contributor:

- [Drew Furgiuele](#) | Senior Cloud Solution Architect

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

For help getting started, see the following resources:

- [Implement a secure hybrid network](#)
- [Edsv5 series VMs](#)
- [Ebsv5 series VMs](#)
- [Lsv3 series VMs](#)
- [Proximity placement groups](#)
- [Azure availability zones](#)

For help with the automation process, see the following templates that SAS provides:

- [SAS Viya 4 Infrastructure as Code](#)
- [SAS Viya 3.5 Guide](#)
- [SAS 9.4 Grid](#)

Related resources

- [Azure Kubernetes in event stream processing](#)
- [GitOps for Azure Kubernetes Service](#)
- [Monitor a microservices architecture in Azure Kubernetes Service \(AKS\)](#)
- [Cost management for Kubernetes](#)
- [Oracle Database with Azure NetApp Files](#)
- [SQL Server on Azure Virtual Machines with Azure NetApp Files](#)

Deploy SAS Grid 9.4 on Azure NetApp Files

Azure NetApp Files

Azure Virtual Machines

SAS analytics software provides a suite of services and tools for drawing insights from data and making intelligent decisions. SAS solutions provide analytics, artificial intelligence, business intelligence, customer intelligence, data management, and fraud and security intelligence.

If you're deploying [SAS Grid on Azure](#), [Azure NetApp Files](#) is a [viable primary storage option](#). When you use the scalable services of Azure NetApp Files, you can scale the storage allocations up or down at any time without interruption to the services. You can also adjust the storage service level to the performance requirements dynamically.

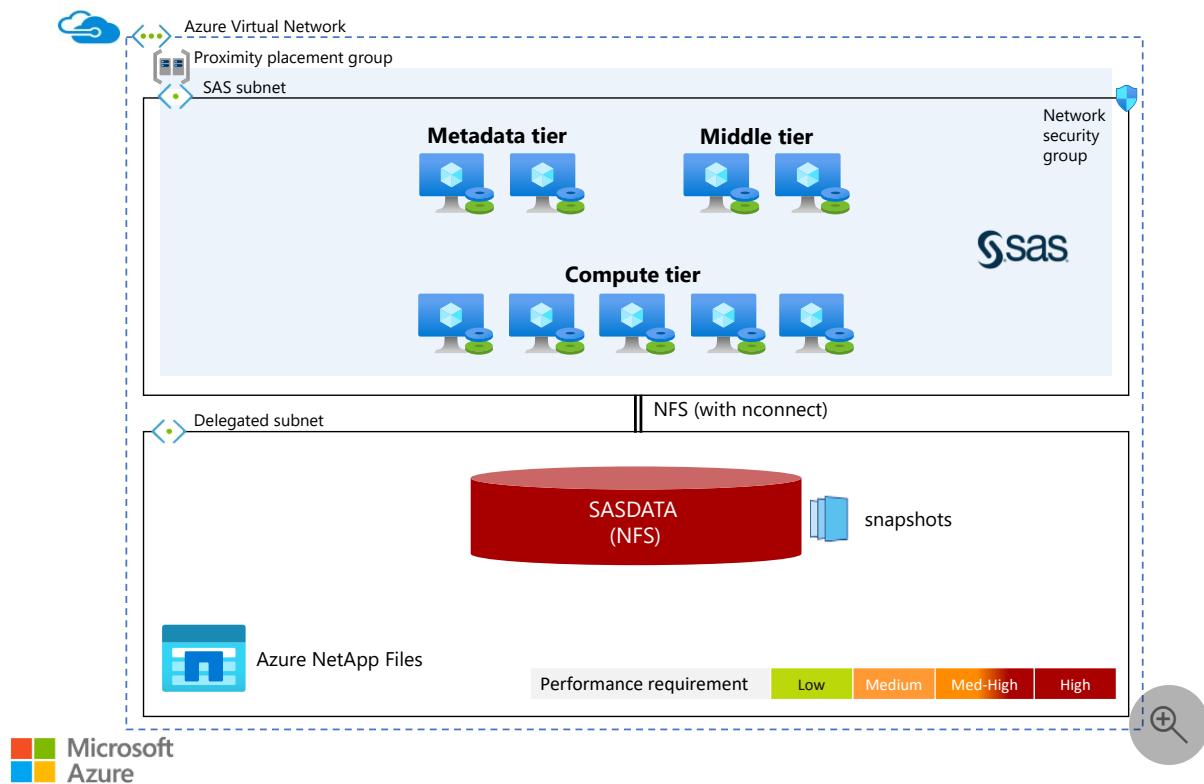
SAS offers these primary platforms, which Microsoft has validated:

- SAS Grid 9.4
- SAS Viya

SAS Grid 9.4 has been validated on Linux.

This article provides general information for running SAS Grid 9.4 on Azure, using Azure NetApp Files for SASDATA storage. It also provides guidance on storage options for SASWORK. These guidelines are based on the assumption that you host your own SAS solution on Azure, in your own tenant. SAS doesn't provide hosting for SAS Grid on Azure.

Architecture



Download a [PowerPoint file](#) of all diagrams in this article.

Dataflow

The compute tier uses SASDATA (and optionally SASWORK) volumes to share data across the grid. SASDATA is an NFS-connected volume on Azure NetApp Files.

- A compute node reads input data from SASDATA and writes results back to SASDATA.
- A subsequent part of the analytics job can be run by another node in the compute tier. It uses the same procedure to obtain and store the information that it needs to process.

Potential use cases

A scalable SAS Grid deployment that uses Azure NetApp Files is applicable to these use cases:

- Financial analytics
- Fraud detection
- Tracking and protection of endangered species
- Science and medicine
- Analytics and AI

Requirements for storage performance

For SAS 9.4 (SAS Grid or SAS Analytics Pro) deployments on Azure, Azure NetApp Files is a viable primary storage option for SAS Grid clusters of limited size. [SAS recommends 100 MiB/s throughput per physical core.](#) Given that recommendation, SAS Grid clusters that use an Azure NetApp Files volume for SASDATA (persistent SAS data files) are scalable to 32 to 48 physical cores across two or more Azure virtual machines. SAS cluster sizes are based on the architectural constraint of a single SASDATA namespace per SAS cluster and the available single [Azure NetApp Files volume bandwidth](#). The core count guidance will be revisited as Azure infrastructure (compute, network, and per file system storage bandwidth) increases over time.

Azure NetApp Files volume performance expectations

A single Azure NetApp Files volume can handle up to 4,500 MiB/s of reads and 1,500 MiB/s of writes. Given an Azure instance type with sufficient egress bandwidth, a single virtual machine can consume all the write bandwidth of a single Azure NetApp Files volume. However, only the largest single virtual machine can consume all the read bandwidth of a single volume.

SASDATA, the main shared workload of SAS 9.4, has an 80:20 read/write ratio. The important *per volume* numbers for an 80:20 workload with 64KiB read/write are:

- 2,400 MiB/s of read throughput and 600 MiB/s of write throughput running concurrently (~3,000 MiB/s combined).

For more information, see [Azure NetApp Files performance benchmarks for Linux](#).

Note

Azure NetApp Files large volumes feature is now available. This feature provides higher per-volume throughput than regular Azure NetApp Files volumes do. This capability can be considered in case more performance is required for your SASDATA (or SASWORK) volumes. See [this documentation](#) for details.

Capacity recommendations

The [Azure NetApp Files performance calculator](#) can provide guidance for sizing SASDATA volumes.

It's important to choose an appropriate service level because:

- Volume bandwidth is based on volume capacity.
- Capacity cost is based on the service level.
- Your choice of service level is based on capacity versus bandwidth needs.

In the calculator, select **advanced**, select a region, and enter the following values.

- Volume size: Desired capacity
- Throughput: Desired throughput, considering 100 MiB/s per core
- Read percentage: 80%
- IOPS: 0
- I/O size: **64KiB Sequential**

The output at the bottom of the screen provides recommended capacity requirements at each service level and the cost per month, based on the price for the selected region:

- **Throughput.** The bandwidth of the volume, based on the workload mix. For an 80% 64-KiB sequential read workload, 3,096 MiB/s is the expected maximum.
- **IOPS.** The number of IOPS the volume provides at the specified throughput.
- **Volume Size.** The amount of capacity needed by the volume at the given service levels to achieve the required throughput. Volume capacity (reported in GiBs) can be equal to or less than capacity pool size. This recommendation is based on the assumption that you're using automatic QoS capacity pool types. To further optimize capacity versus throughput distribution across volumes within a capacity pool, consider manual QoS capacity pool types.
- **Capacity Pool Size.** The pool size. A volume's capacity is carved from a capacity pool. Capacity pools are sized in 1-TiB increments.
- **Capacity Pool Cost (USD/month).** The cost per month of the capacity pool at the given size and service level.
- **Volume Show Back (USD/month).** The cost per month of the capacity for the volume at the specified capacity. Charges are based on the allocated capacity pool sizes. The volume show back indicates the volume amount.

Note

The user experience is the same regardless of the service level, as long as sufficient bandwidth is provisioned.

Control costs as needed by using volume shaping in Azure NetApp Files. Two dynamic options are available to influence performance and cost:

- [Dynamically resize a volume and capacity pool](#)
- [Dynamically change the service level of a volume](#)

Learn more about the [Azure NetApp Files cost model](#).

Data protection

Azure NetApp Files uses [snapshots](#) to help you protect your data. Snapshots provide space-efficient, crash-consistent, near-instantaneous images of your Azure NetApp Files volumes. You can create snapshots manually at any time or schedule them by using a [snapshot policy](#) on the volume.

Use a snapshot policy to add automated data protection to your volumes. You can restore snapshots in place quickly by using [snapshot revert](#). Or you can [restore a snapshot to a new volume](#) for fast data recovery. You can also use [restore to new volume functionality](#) to provide test/dev environments with current data.

For extra levels of data protection, you can use data protection solutions that use [Azure NetApp Files backup](#) or partner backup software.

Components

- [Azure Virtual Machines](#): SAS Grid requires high memory, storage, and I/O bandwidth, in an appropriate ratio with the number of cores. Azure offers predefined virtual machine (VM) sizes with lower vCPU counts that can help to balance the number of cores required with the amount of memory, storage, and I/O bandwidth.

For more information, see [Constrained vCPU capable VM sizes](#). It's important to thoroughly understand what compute resources are available with *each* instance. To run SAS Grid on Azure with Azure NetApp Files, we recommended these instance types:

- Standard_E64-16ds_v4 or Standard_E64-16ds_v5
- Standard_E64-32ds_v4 or Standard_E64-32ds_v5

Be sure to review the [best practices for using SAS on Azure](#), including the updates in the comments.

- [Azure NetApp Files](#): You can store SASDATA on an Azure NetApp Files volume, shared across the compute cluster.

You can optionally also use Azure NetApp Files NFS volumes for SASWORK.

Azure NetApp Files is available in three performance [service levels](#):

- Standard
- Premium

- Ultra

Your volume performance is mostly defined by the service level. The size of your volume is also a factor, because the obtainable throughput is [determined by the service level and the size of the volume](#).

Storage options for SASDATA

Because Azure NetApp Files can provide high throughput and low-latency access to storage, it's a viable, and faster, alternative to Premium Disk. Network-attached storage isn't throttled at the VM level as it is with managed disks, so you get higher throughput to storage.

To estimate the required tier for your SASDATA capacity, use the [Azure NetApp Files Performance Calculator](#). (Be sure to select **advanced**.)

Because Azure NetApp Files NFS volumes are shared, they're a good candidate for hosting SASDATA, when used with the properly sized VM instance types and Red Hat Enterprise Linux (RHEL) distribution, discussed later in this article.

Storage options for SASWORK

The following table shows the most common storage options for deploying SASWORK on Azure. Depending on your size (capacity) and speed (bandwidth) requirements, you have three options: temporary storage, managed disk, and Azure NetApp Files.

expand [Expand table](#)

	Temporary storage	Managed disk	Azure NetApp Files
Size	Small	Large	Extra large
Speed	Extra large	Small	Medium

Take these considerations into account when choosing an option:

- [Temporary storage](#) (or *ephemeral storage*) provides the highest bandwidth, but it's available only in smaller sizes. (Size depends on the VM SKU.) Depending on the available and required capacities, this option might be best.

- If the required SASWORK capacity exceeds the temporary storage size of the VM SKU that you've selected, consider using an Azure managed disk to host SASWORK. Keep in mind, however, that the throughput to a managed disk is limited by the VM architecture by design, and that it varies depending on the VM SKU. Therefore, this storage option is viable only for environments that have lower SASWORK performance requirements.
- For the highest SASWORK capacity requirements and an average performance requirement beyond what Azure managed disks can provide, consider Azure NetApp Files for SASWORK. It provides a large size together with fast throughput.

ⓘ Important

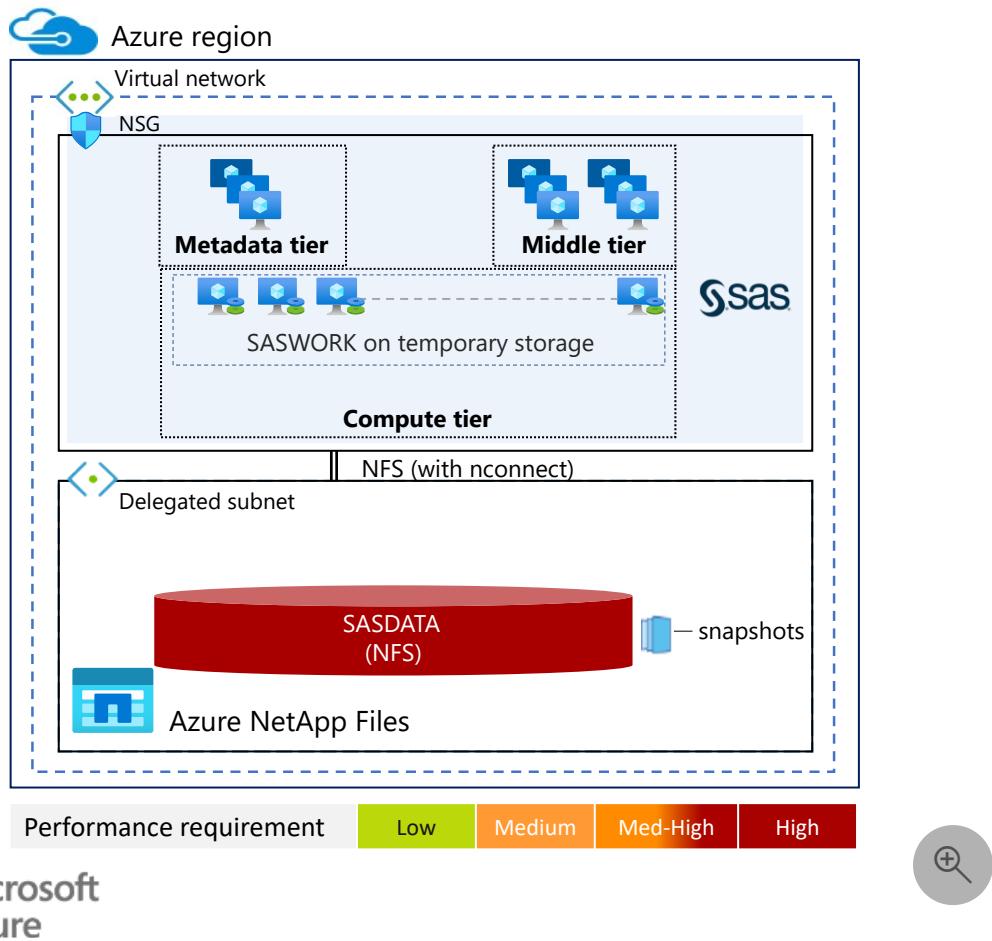
In any scenario, keep in mind that SASWORK can't be shared between VM compute nodes, so you need to create separate SASWORK volumes for each compute node. Volumes need to be NFS-mounted on only one compute node.

In using the preceding table, to decide whether your needs are small, large, medium, or extra large, take into account the scale of the deployment, the number of VMs and cores, and the associated capacity and performance requirements. You need to make these assessments for each deployment.

The options in the table correspond to deployments described in the architectures that follow. In all scenarios, SASDATA is hosted on an Azure NetApp Files NFS volume and shared across the compute nodes. For some RHEL distributions, we recommend using the NFS [nconnect](#) option to create multiple network flows to the volume. For more information, see the [NFS mount options](#) section of this article.

Temporary storage architecture

SASWORK on temporary storage

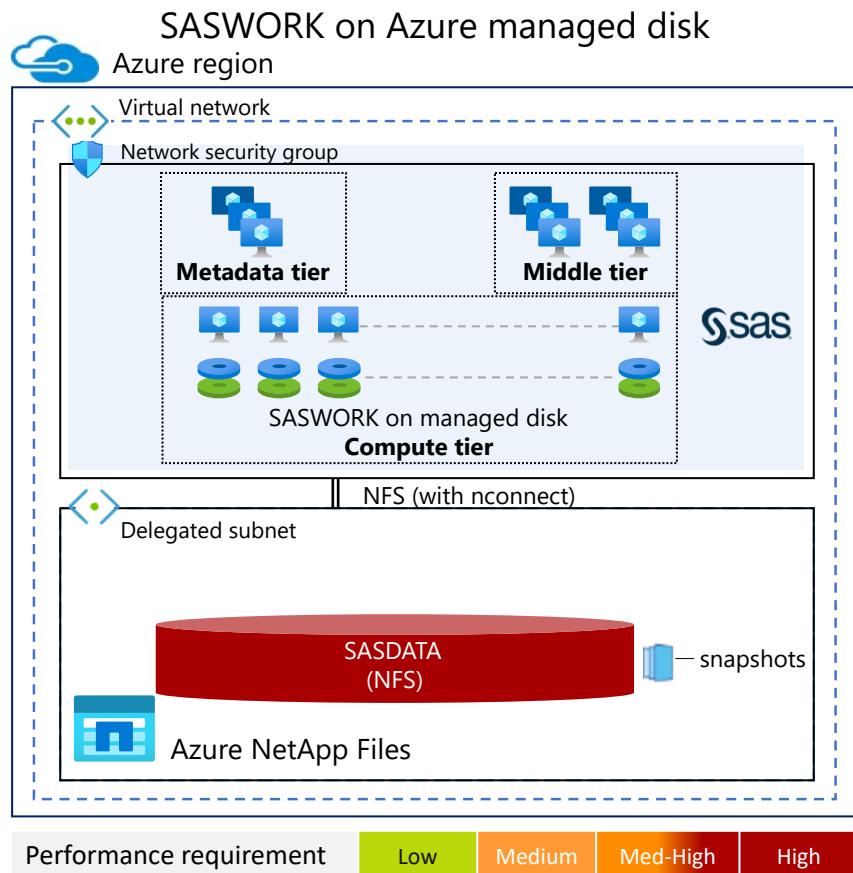


For smaller SASWORK capacity requirements, Azure VM temporary storage is a fast and cost-effective solution. In this architecture, each VM in the compute tier is equipped with some temporary storage. To determine the temporary storage sizes for the VMs you use, see the [Azure VM documentation](#).

Dataflow

- A compute node reads input data from SASDATA and writes results back to SASDATA.
- A subsequent part of the analytics job can be run by another node in the compute tier. It uses the same procedure to obtain and store the information that it needs to process.
- The temporary work directory SASWORK isn't shared. It's stored in temporary storage on each compute node.

Managed disk architecture

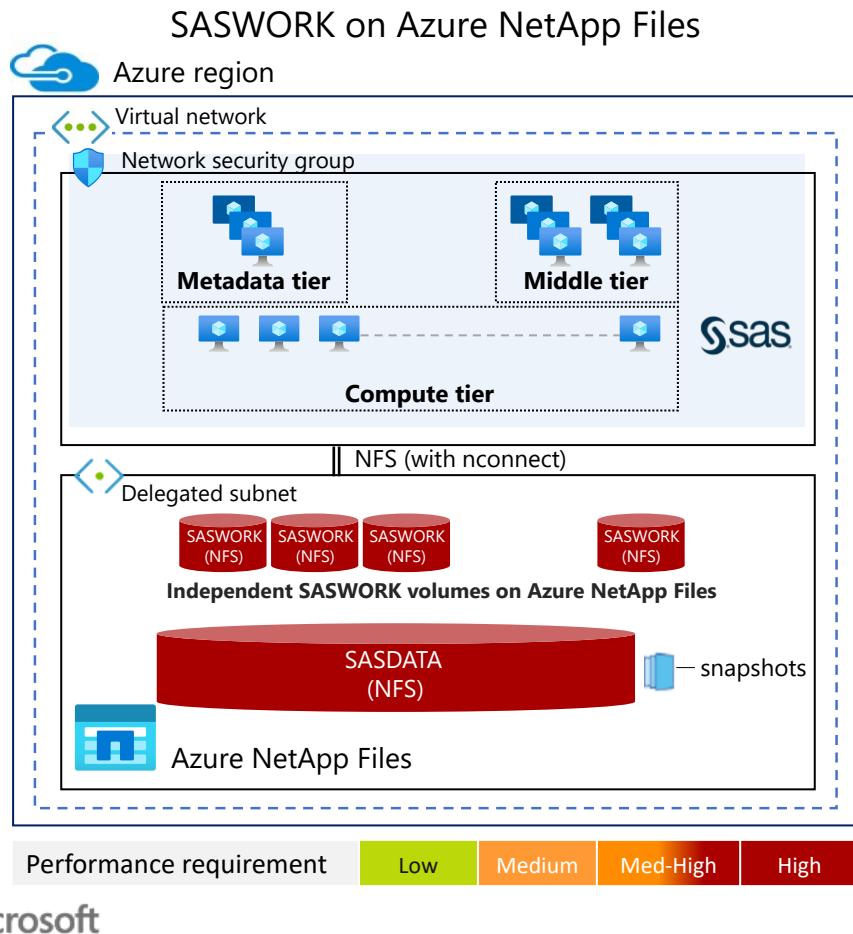


If your capacity requirements for SASWORK exceed the capacities available in temporary storage, Azure managed disks are a good alternative. Managed disks are available in various sizes and performance levels. For more information, see [Scalability and performance targets for VM disks](#).

Dataflow

- A compute node reads input data from SASDATA and writes results back to SASDATA.
- A subsequent part of the analytics job can be run by another node in the compute tier. It uses the same procedure to obtain and store the information that it needs to process.
- The temporary work directory SASWORK isn't shared. It's stored on managed disks that are attached to each compute node.

Azure NetApp Files architecture



For higher SASWORK capacity and/or medium performance requirements, consider using Azure NetApp Files. Azure NetApp Files provides volume capacities as high as 100 TiB. Each node in the compute tier should have its own SASWORK volume. The volumes shouldn't be shared.

Dataflow

- A compute node reads input data from SASDATA and writes results back to SASDATA.
- A subsequent part of the analytics job can be run by another node in the compute tier. It uses the same procedure to obtain and store the information that it needs to process.
- The temporary work directory SASWORK isn't shared. It's stored on individual Azure NetApp Files volumes that are attached to each compute node.

Scale and configuration recommendations

- For the best and most consistent latency for data traffic between the instances in the SAS cluster, make sure all VMs are created in the same [proximity placement group](#).
- Review the **General Tuning Guidance** section in [Best Practices for Using SAS on Azure](#).
- For optimal network bandwidth, enable [Accelerated Networking](#).

RHEL distributions and NFS settings

RHEL distributions

RHEL is the recommended distribution for running SAS 9 on Linux. Each kernel supported by Red Hat has its own NFS bandwidth constraints.

For specifics about running SAS on Azure, see [Best Practices for Using SAS on Azure](#).

Azure Standard_E64-16ds_v4 and Standard_E64-32ds_v4 VMs, or their v5 equivalents, are recommended for SAS. Taking these recommendations into account, this section provides some guidelines for using SAS with Azure NetApp Files.

- If you use RHEL 7, Standard_E64-16ds_v4 or Standard_E64-16ds_v5 is the best choice, based on the 100-MiB/s per physical core target for SASDATA.
 - Standard_E64-16ds_v4: 90–100 MiB/s per core
 - Standard_E64-32ds_v4: 45–50 MiB/s per core
- If you use RHEL 8.2, either Standard_E64-16ds_v4 or Standard_E64-32ds_v4, or their v5 equivalents, are possible options. Standard_E64-16ds_v4 is preferable, given the 100-MiB/s per core target for SASDATA.
 - Standard_E64-16ds_v4: 150–160 MiB/s per core
 - Standard_E64-32ds_v4: 75–80 MiB/s per core
- If you use RHEL 8.3, both Standard_E64-16ds_v4 and Standard_E64-32ds_v4, or their v5 equivalents, are fully acceptable, given the per-core throughput target:
 - Validation indicates 3,200 MiB/s of reads.
 - These results are achieved with the NFS `nconnect` mount option.

Testing shows that a single RHEL 7 instance achieves no more than roughly 750–800 MiB/s of read throughput against a single Azure NetApp Files storage endpoint (that is, against a network socket). 1,500 MiB/s of writes are achievable against the same endpoint, if you use 64-KiB `rsize` and `wszie` NFS mount options. Some evidence suggests that the previously noted read throughput ceiling is an artifact of the 3.10 kernel. For more information, see [RHEL CVE-2019-11477](#).

Testing shows that a single RHEL 8.2 instance, with its 4.18 kernel, is free of the limitations noted in the 3.10 kernel. So 1,200-1,300 MiB/s of read traffic is achievable, if you use a 64-KiB `rsize` and `wsize` NFS mount option. For large sequential writes, you can expect the same 1500 MiB/s of achievable throughput that you'd get on RHEL 7.

With a single RHEL 8.3 instance, with the [nconnect mount option ↗](#) (which is new in the RHEL 8.3 distribution), about 3,200 MiB/s read throughput is achievable from a single Azure NetApp Files volume. Don't expect more than 1,500 MiB/s of writes to an Azure NetApp Files single volume, even when you apply `nconnect`.

Kernel tunables

Slot table entries

NFSv3 doesn't have a mechanism to [negotiate concurrency](#) between the client and the server. The client and the server each define their limits without awareness of the other. For the best performance, you should line up the maximum number of client-side `sunrpc` slot table entries with that supported without pushback on the server. When a client overwhelms the server network stack's ability to process a workload, the server responds by decreasing the window size for the connection, which isn't ideal for performance.

By default, modern Linux kernels define the per-connection `sunrpc` slot table entry size `sunrpc.max_tcp_slot_table_entries` to support 65,536 outstanding operations. These slot table entries define the limits of concurrency. Values this high are unnecessary because Azure NetApp Files defaults to 128 outstanding operations.

We recommend that you tune the client to the same number:

- Kernel tunables (via `/etc/sysctl.conf`)
 - `sunrpc.tcp_max_slot_table_entries=128`

File system cache tunables

You also need to [understand the following factors](#) about file system cache tunables:

- Flushing a dirty buffer leaves the data in a clean state, usable for future reads until memory pressure leads to eviction.
- There are three triggers for an asynchronous flush operation:
 - Time based: When a buffer reaches the age defined by the `vm.dirty_expire_centisecs` or `vm.dirty_writeback_centisecs` tunable, it must be

marked for cleaning (that is, flushing or writing to storage).

- Memory pressure: For details, see [vm.dirty_ratio](#) | [vm.dirty_bytes](#).
- Close: When a file handle is closed, all dirty buffers are asynchronously flushed to storage.

These factors are controlled by four tunables. You can tune each tunable dynamically and persistently by using `tuned` or `sysctl` in the `/etc/sysctl.conf` file. Tuning these variables improves performance for SAS Grid:

- Kernel tunables (via custom tuned profile)
 - `include = throughput-performance`
 - `vm.dirty_bytes = 31457280`
 - `vm.dirty_expire_centisecs = 100`
 - `vm.dirty_writeback_centisecs = 300`

NFS mount options

We recommend the following NFS mount options for NFS shared file systems that are used for permanent **SASDATA** files:

RHEL 7 and 8.2

```
bg,rw,hard,rsize=65536,wsize=65536,vers=3,noatime,nodiratime,rdirplus,acdirm  
in=0,tcp,_netdev
```

RHEL 8.3

```
bg,rw,hard,rsize=65536,wsize=65536,vers=3,noatime,nodiratime,rdirplus,acdirm  
in=0,tcp,_netdev,nconnect=8
```

We recommend the following mount options for **SASWORK** volumes, where the respective volumes are used exclusively for SASWORK and not shared between nodes:

RHEL 7 and 8.2

```
bg,rw,hard,rsize=65536,wsize=65536,vers=3,noatime,nodiratime,rdirplus,acdirm  
in=0,tcp,_netdev,nocto
```

```
bg,rw,hard,rsize=65536,wsize=65536,vers=3,noatime,nodiratime,rdirplus,acdirm  
in=0,tcp,_netdev,nocto,nconnect=8
```

For more information on the benefits and cost of the `nocto` mount option, see [Close-to-open consistency and cache attribute timers](#).

You should also review [Azure NetApp Files: A shared file system to use with SAS Grid on MS Azure](#), including all updates in the comments.

NFS read-ahead settings

We recommend that you set the NFS read-ahead tunable for all RHEL distributions to 15,360 KiB. For more information, see [How to persistently set read-ahead for NFS mounts](#).

Alternatives

The storage solution in the preceding architectures is highly available, as specified by the [Azure NetApp Files service level agreement](#). For extra protection and availability, you can replicate the storage volumes to another Azure region by using Azure NetApp Files [cross-region replication](#).

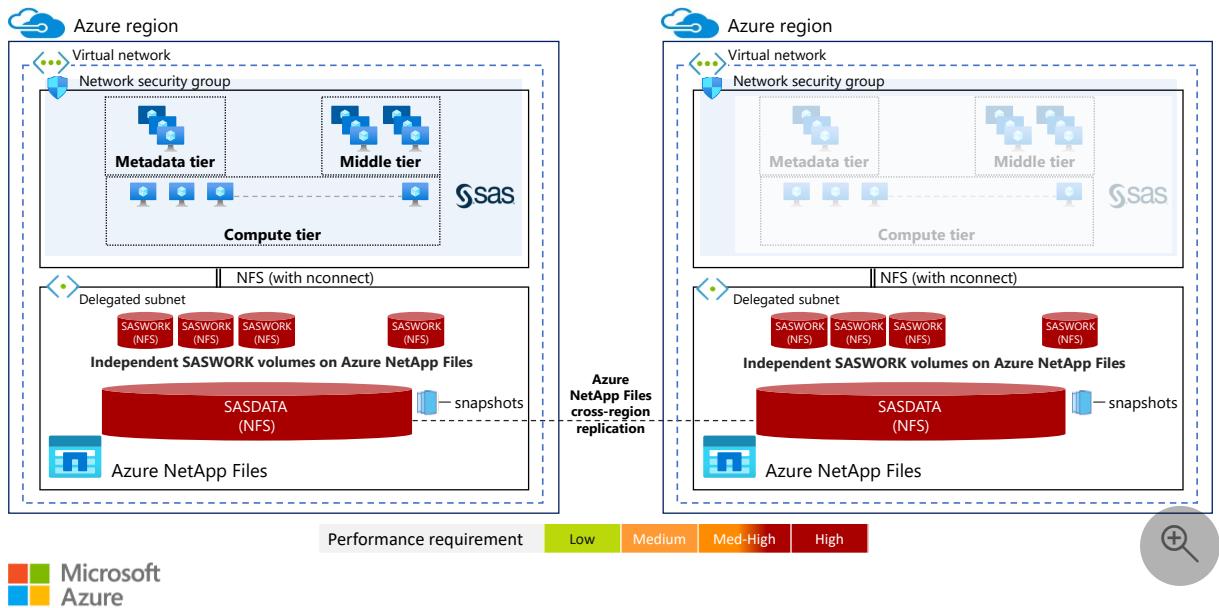
There are two key advantages to replicating the volumes via the storage solution:

- There's no additional load on the application VMs.
- This solution eliminates the need to run VMs in the destination region during normal operation.

The storage contents are replicated without the use of any compute infrastructure resources, and the destination region doesn't need to run the SAS software. The destination VMs don't need to be running to support this scenario.

The following architecture shows how the storage content on Azure NetApp Files is replicated to a second region, where the storage is populated with a replica of the production data. If there's a failover, the secondary region is brought online, and the VMs are started so production can resume in the second region. You need to reroute traffic to the second region by reconfiguring load balancers that aren't shown in the diagram.

SASDATA on Azure NetApp Files with cross-region replication



The typical RPO for this solution is less than 20 minutes when the cross-region replication update interval is set to 10 minutes.

Dataflow

- A compute node reads input data from SASDATA and writes results back to SASDATA.
- A subsequent part of the analytics job can be run by another node in the compute tier. It uses the same procedure to obtain and store the information that it needs to process.
- The temporary work directory SASWORK isn't shared. It's stored on individual Azure NetApp Files volumes that are attached to each compute node.
- Azure NetApp Files cross-region replication asynchronously replicates the SASDATA volume, including all snapshots, to a DR region to facilitate failover if there's a regional disaster.

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 your application can meet the commitments you make to your customers. For more information, see [Overview of the reliability pillar](#).

Azure NetApp Files provides a standard 99.99% availability [SLA](#) for all tiers and all supported regions. Azure NetApp Files also supports provisioning volumes in [availability zones](#) that you choose, and HA deployments across zones.

For improved RPO/RTO SLAs, integrated data protection with [snapshots and backup](#) is included with the service. [Cross-region replication](#) provides the same benefits across Azure regions.

Security

Security provides assurance against deliberate attacks and the abuse of your valuable data and systems. For more information, see [Overview of the security pillar](#).

Azure NetApp Files provides a level of [security](#) because volumes are provisioned, and data traffic stays, within your virtual networks. There's no publicly addressable endpoint. All [data is encrypted at rest](#) at all times. You can optionally encrypt data-in-transit.

[Azure Policy](#) can help you enforce organizational standards and assess compliance at scale. Azure NetApp Files supports Azure Policy via [custom and built-in policy definitions](#).

Performance efficiency

Performance efficiency is the ability of your workload to scale to meet the demands placed on it by users in an efficient manner. For more information, see [Performance efficiency pillar overview](#).

Performance

Depending on your requirements for throughput and capacity, keep the following considerations in mind:

- The [performance considerations for Azure NetApp Files](#).
- The required Azure NetApp Files capacity and service levels for SASDATA.
- The guidance in this article for choosing a storage type for SASWORK.

Note

Azure NetApp Files large volumes feature is now available. This feature provides higher per-volume throughput than regular Azure NetApp Files volumes do. This

capability can be considered in case more performance is required for your SASDATA (or SASWORK) volumes. See [this documentation](#) for details.

Scalability

You can easily scale compute performance by adding VMs to the scale sets that run the three tiers of the SAS solution.

You can dynamically scale storage of Azure NetApp Files volumes. If you use [automatic QoS](#), performance is scaled at the same time. For more granular control of each volume, you can also control the performance of each volume separately by using [manual QoS](#) for your capacity pools.

Azure NetApp Files volumes are available in three performance tiers: [Ultra](#), [Premium](#), and [Standard](#). Choose the tier that best suits your performance requirements, taking into account that available performance bandwidth [scales with the size of a volume](#). You can change the service level of a volume at any time. For more information about the Azure NetApp Files cost model, see these [pricing examples](#).

You can use the [Azure NetApp Files Performance Calculator](#) to get started.

Cost optimization

Cost optimization is about reducing unnecessary expenses and improving operational efficiencies. For more information, see [Overview of the cost optimization pillar](#).

Cost model

Understanding the [cost model for Azure NetApp Files](#) can help you manage your expenses.

Azure NetApp Files billing is based on provisioned storage capacity, which you allocate by creating capacity pools. Capacity pools are billed monthly based on a set cost per allocated GiB per hour.

If your capacity pool size requirements fluctuate (for example, because of variable capacity or performance needs), consider [dynamically resizing your volumes and capacity pools](#) to balance cost with your capacity and performance needs.

If your capacity pool size requirements remain the same but performance requirements fluctuate, consider [dynamically changing the service level of a volume](#). You can provision and deprovision capacity pools of different types throughout the month, providing just-

in-time performance and reducing costs during periods when you don't need high performance.

Pricing

Based on your capacity and performance requirements, decide which Azure NetApp Files service level you need (Standard, Premium, or Ultra). Then use the [Azure Pricing calculator](#) to evaluate the costs for these components:

- SAS on Azure components
- Azure NetApp Files
- Managed disk (optionally)
- Virtual network

Operational excellence

Operational excellence covers the operations processes that deploy an application and keep it running in production. For more information, see [Overview of the operational excellence pillar](#).

SAS Grid on Azure provides flexibility and a fast deployment. Here are some benefits:

- Meet changing business demands with dynamic workload balancing
- Create a highly available SAS computing environment
- Get faster results from your existing IT infrastructure
- Grow computing resources incrementally and cost-effectively
- Manage all your analytical workloads
- Easily transition from a siloed server or multiple-PC environment to a SAS grid environment

Deploy this scenario

It's best to deploy the workloads by using an infrastructure as code (IaC) process. SAS workloads can be sensitive to misconfigurations that often occur in manual deployments and reduce productivity.

To get a start with designing your SAS Grid on Azure solution, review [SAS on Azure Architecture](#) and [Automating SAS Deployment on Azure by using GitHub Actions](#).

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Geert van Teylingen](#) | Group Product Manager
- [Arnt de Gier](#) | Technical Marketing Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- Quickstart webinar on how to get started on Azure [↗](#)
- Azure NetApp Files: A shared file system to use with SAS Grid on Azure [↗](#)
- Azure NetApp Files Performance Calculator [↗](#)
- Azure NetApp Files documentation
- Training: Introduction to Azure NetApp Files

Related resources

- [SAS on Azure, VM sizing recommendations](#)
- [SAS on Azure, Azure NetApp Files](#)
- [SAS on Azure, network and VM placement considerations](#)
- [SAS on Azure, security considerations](#)

High-performance computing (HPC) deployments on Azure

Article • 03/07/2023

[High-performance computing \(HPC\)](#), also called *big compute*, solves complex mathematical tasks using a large number of CPU or GPU-based computers. Many industries use HPC to solve some of their most difficult problems, including genomics, semiconductor design, and weather modeling.

Run HPC applications on Azure VMs

The articles listed here describe the steps for running various HPC applications on Azure virtual machines. They also show the performance results achieved when running each application on Azure.

Application	Summary
ADS CFD Code Leo	Learn how Code Leo is a URANS-based flow solver that delivers accurate and fast flow simulations for general flow configurations.
Altair AcuSolve	See how AcuSolve provides comprehensive software and tools to solve fluid mechanics problems.
Altair EDEM	Use a discrete element method (DEM) to simulate and analyze the behavior of bulk materials, such as coal.
Altair nanoFluidX	Simulate single-phase and multiphase flows, based on a weakly compressible Lagrangian SPH formulation.
Altair Radioss	Predict crash response and dynamic, transient-loading effects on vehicles, structures, and other products. Radioss is a multidisciplinary finite-element solver for linear and nonlinear problems.
Altair ultraFluidX	Predict the aerodynamic properties of passenger and heavy-duty vehicles, and evaluate building and environmental aerodynamics.
Ansys CFX	Learn how Ansys uses an equilibrium phase change model and relies on material properties to reliably predict cavitation without the need for empirical model parameters.
Ansys Fluent	Use Ansys Fluent to model fluid flow, heat and mass transfer, chemical reactions, and more.

Application	Summary
Ansys LS-DYNA	Learn how Ansys LS-DYNA simulates the response of materials to short periods of severe loading for applications like drop tests, impact and penetration, smashes and crashes, and occupant safety.
Ansys Rocky	Simulate the flow behavior of bulk materials with complex particle shapes and size distributions. Typical applications include conveyor chutes, mills, mixers, and other material-handling equipment.
Autodesk Civil 3D	Learn how civil engineers use Civil 3D, for design automation and production, enabling multidisciplinary team coordination.
Autodesk Inventor	Learn how Autodesk Inventor provides professional-grade mechanical design, documentation, and product simulation tools.
Autodesk VRED for HPC on Azure	See how automotive designers and engineers can use Autodesk VRED to create product presentations, design reviews, and virtual prototypes by using interactive CPU and GPU ray tracing.
AVL FIRE M	Learn how the AVL FIRE M computational fluid dynamics (CFD) simulation application performs on an Azure virtual machine.
Barracuda Virtual Reactor	Simulate the 3D transient behavior in fluid-particle systems, including multiphase hydrodynamics, heat balance, and chemical reactions.
Engys ELEMENTS	Solve flow-related problems encountered in automotive design by running Engys ELEMENTS on an Azure Virtual Machine. You can also use ELEMENTS to analyze the aerodynamics of other vehicles, like high-speed trains, motorcycles, and competition bicycles.
Engys HELYX	Learn how you can run Engys HELYX on a virtual machine to simulate complex flows in your engineering analysis and design optimization. HELYX is used in the automotive, aerospace, construction, marine, turbo, and energy industries.
GROMACS	Learn how GROMACS (GROningen MACHine for Simulations) is used primarily for dynamic simulations of biomolecules and provides a rich set of calculation types and preparation and analysis tools.
Indica Labs HALO AI	Decipher and assess the complex patterns of histologically stained tissues in a way that's similar to how a pathologist thinks. HALO AI is a collection of train-by-example classification and segmentation tools underpinned by advanced deep learning neural network algorithms.
Luxion KeyShot	Use photon mapping to create 3D renderings, animations, and interactive visuals that make simulation of global illumination in complex scenes more efficient.

Application	Summary
OpenFOAM	See how OpenFOAM is a free, open-source computational fluid dynamics (CFD) application where users have permission to modify and compile the package based on the needs and the physics of the problem they're solving.
Remcom XFDTD	Learn all about XFDTD – electromagnetic simulation software that includes full-wave, static, biothermal, optimization, and circuit solvers.
Samadii DEM	Analyze and interpret large-scale particles at high speed. Samadii DEM uses a discrete element method (DEM), which is a Lagrangian method that determines the movement of particles by using the six-degrees-of-freedom equations of motion, taking into consideration all forces of individual particles.
Samadii EM	See how Samadii EM (electromagnetic) analyzes the electromagnetic field in three-dimensional space by using the Maxwell equation, using the vector finite element method (FEM) and GPU computing.
Samadii Plasma	Learn all about how Samadii Plasma is a particle-based solution for the analysis of plasma behavior. This solution is ideal for the manufacturing and electronics industries.
Samadii SCIV	Analyze fluid behavior, deposition processes, and chemical reactions on rarefied gas regions by using the direct simulation Monte Carlo (DSMC) method. SCIV also provides functions for traditional flow simulation, display deposition processes, and semiconductor device analysis in rarefied gas regions.
Sandi HiFUN	Simulate airflow over aircraft, automobiles, buildings, and ships by using Sandi HiFUN, the general-purpose computational fluid dynamics application. HiFUN is used in the aerospace, automotive, industrial, and wind/turbine industries.
Siemens NX	Use NX for design, simulation, and manufacturing solutions that enable digital twin technology. NX is used in the automotive sector and for projects ranging from supersonic cars to drones for the medical industry.
Siemens Tecnomatix	Learn how Siemens Tecnomatix is a comprehensive portfolio of digital manufacturing solutions that includes part manufacturing, assembly planning, resource planning, plant simulation, human performance, quality, production management, and manufacturing data management.
Turbostream	Enable high-fidelity methods, like unsteady full-annulus simulations, to be used as part of the routine design process.
Visiopharm	Learn all about how Visiopharm is an AI-based image analysis and tissue mining tool that supports drug development research and other research.
WRF	See how Weather Research & Forecasting (WRF) is a mesoscale numerical weather-prediction system that's designed for atmospheric research and operational forecasting applications. WRF serves a wide range of meteorological applications across scales from tens of meters to thousands of kilometers.

Related resources

- Run a Windows VM on Azure
- Run a Linux VM on Azure
- HPC system and big-compute solutions
- HPC cluster deployed in the cloud

Deploy ADS CFD Code Leo for HPC on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [ADS CFD's](#) Code Leo application on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running Code Leo on Azure.

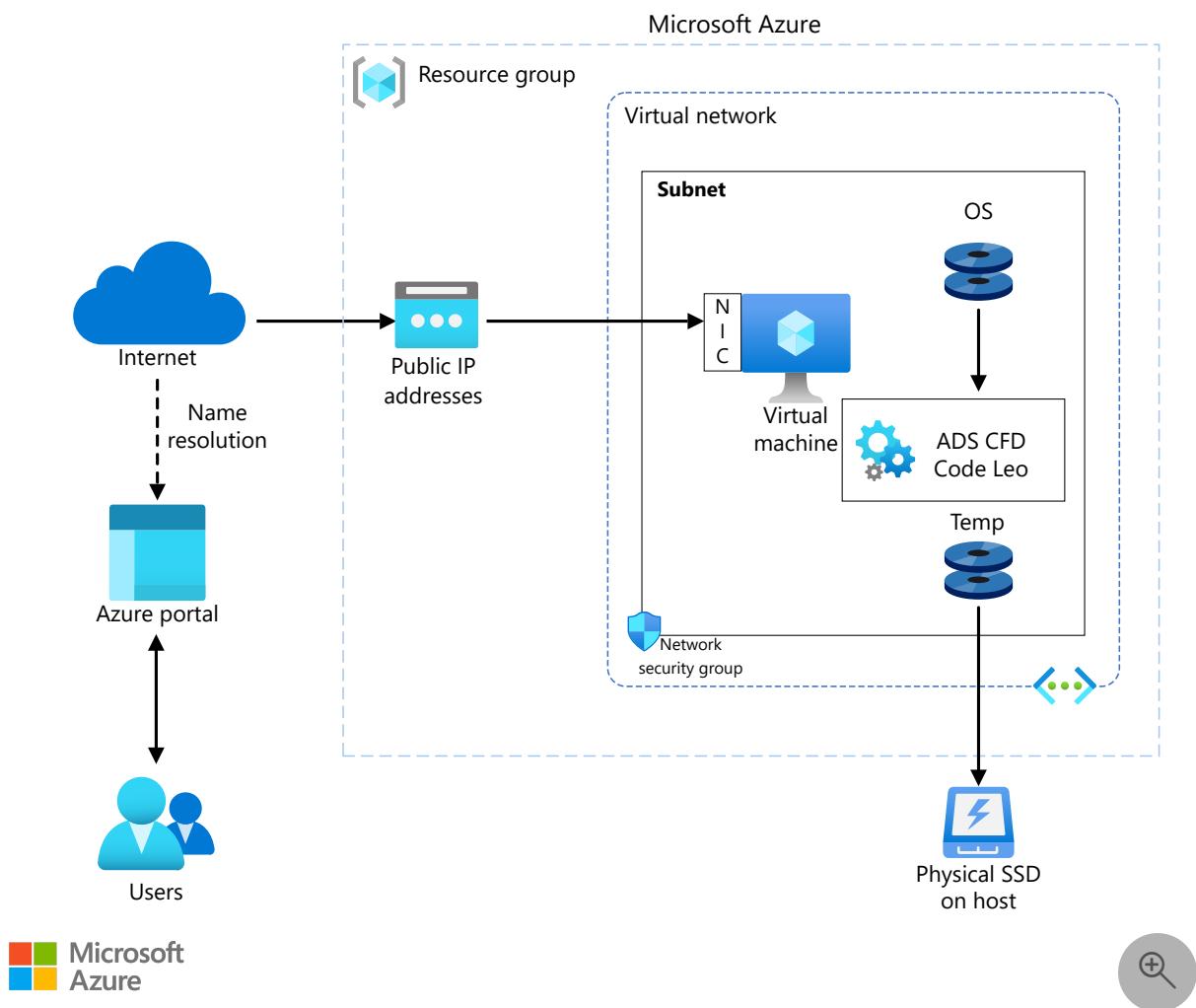
Code Leo is a URANS-based flow solver that delivers accurate and fast flow simulations for general flow configurations. Code Leo:

- Is density based and compressible and has explicit time marching and convergence acceleration.
- Is second-order accurate in time and space with low numerical smoothing.
- Runs both steady-state and unsteady simulations.
- Handles structured multi-block meshes and unstructured meshes with mixed tetrahedrons, pyramids, prisms, and hex elements.
- Allows the use of various turbomachinery rotor/stator interaction models, including sliding mesh, mixing plane, and frozen rotor models.
- Is HPC-aware and uses MPI+MP for parallel computing.

Code Leo was originally designed for CPUs, but ADS CFD extended it to take advantage of the advanced GPU architecture when GPUs became more cost-effective. It's been validated for decades by industry leaders like Air Force Research Laboratory and NASA. Code Leo users can switch easily between CPU and GPU solvers.

ADS CFD software is used primarily in the aerospace and turbomachinery (energy) industries for performance and durability assessments of jet engines and aircraft. One of the main use cases is the analysis and optimization of integrated engine/aircraft configurations so that next-generation aircraft designs can be completed on time and with confidence.

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#). Create Linux and Windows virtual machines in seconds.
- [Azure Virtual Network](#). Use Virtual Network to create your own private network infrastructure in the cloud.

Install Code Leo on a VM

Before you install Code Leo, you need to deploy and connect a VM and install the required NVIDIA and AMD drivers.

ⓘ Important

NVIDIA Fabric Manager installation is required for VMs that use NVLink or NVSwitch.

For information about deploying the VM and installing the drivers, see one of these articles:

- [Run a Windows VM on Azure](#)
- [Run a Linux VM on Azure](#)

To download Code Leo products from the ADS CFD portal:

1. Open the ADS CFD portal in a web browser and sign in.
2. Select the **Support** tab on the home page.
3. Select **Download**.
4. Select the download link for the latest version of Linux.

The package contains one run file.

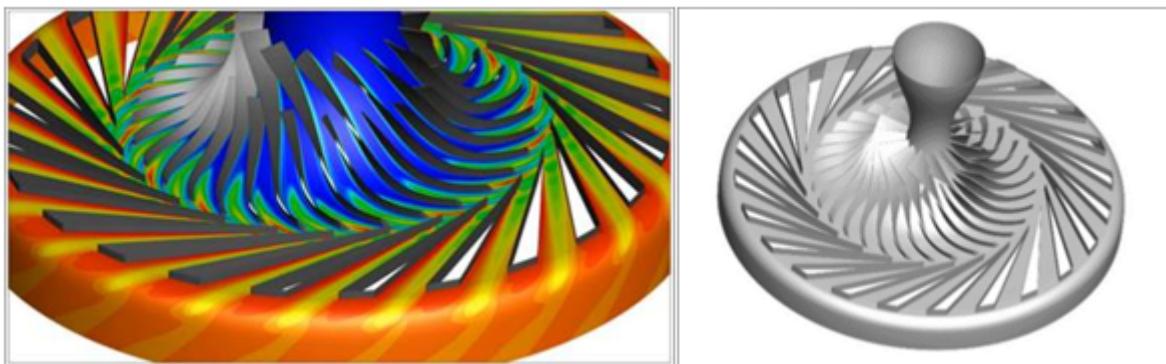
See the [ADS CFD website](#) for instructions for installing Code Leo.

Performance results of Code Leo on an Azure VM

Code Leo was used to run both the steady state and unsteady simulations.

Code Leo V8.21.08 performance results

The CC3 wheel model is used for this performance evaluation. This model has two parts, the impeller and the diffuser, as shown here:



expand [Expand table](#)

Model	Number of elements	Number of nodes
Impeller	670848	742968

Model	Number of elements	Number of nodes
Diffuser	914688	1012095

Full wheel time analyses were performed on ND96asr_v4 and NC24s_v3 Azure VMs.

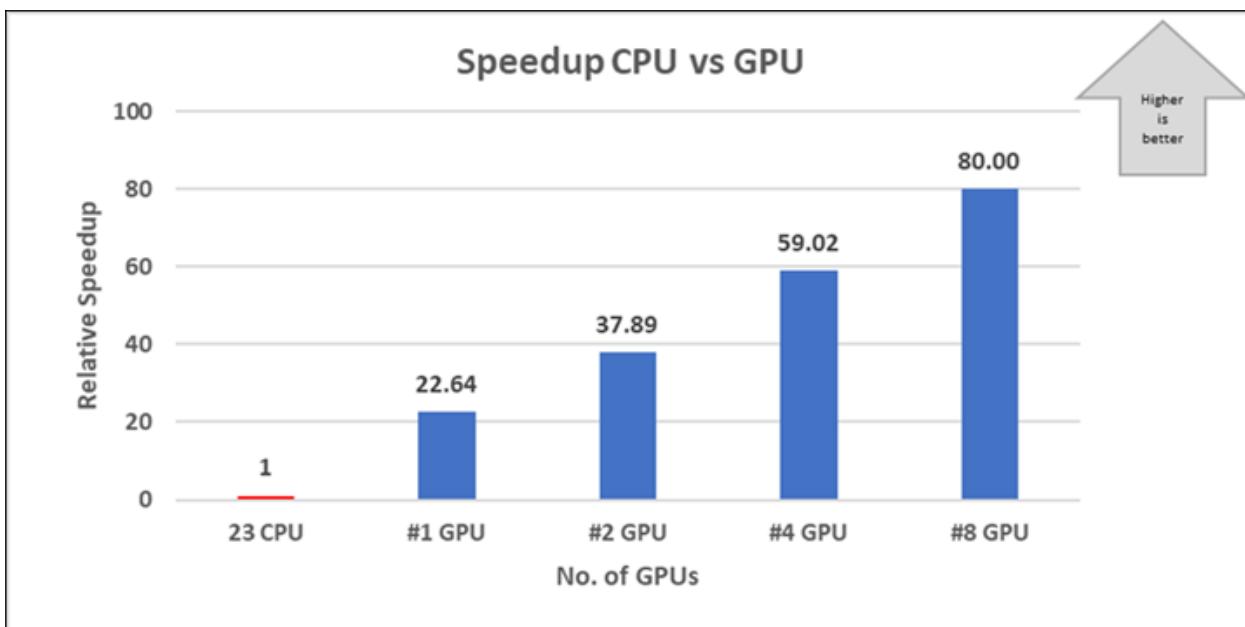
ADS CFD provided CPU results, which provide a baseline for comparing GPU runs on both VM instances.

The elapsed time for CPU simulation is 3,600 minutes. The simulation was performed on a server with Xeon 23 CPUs with a clock speed of 2.4 GHz.

Performance results for NDv4 A100, diffuser and impeller

[Expand table](#)

Number of GPUs	Elapsed time in minutes	Relative improvement
1	159	22.64
2	95	37.89
4	61	59.02
8	45	80.00



Simulations were performed on Xeon 23 CPUs with 2.4 GHz on A100, which has 8 GPUs, each with 40 GB of memory.

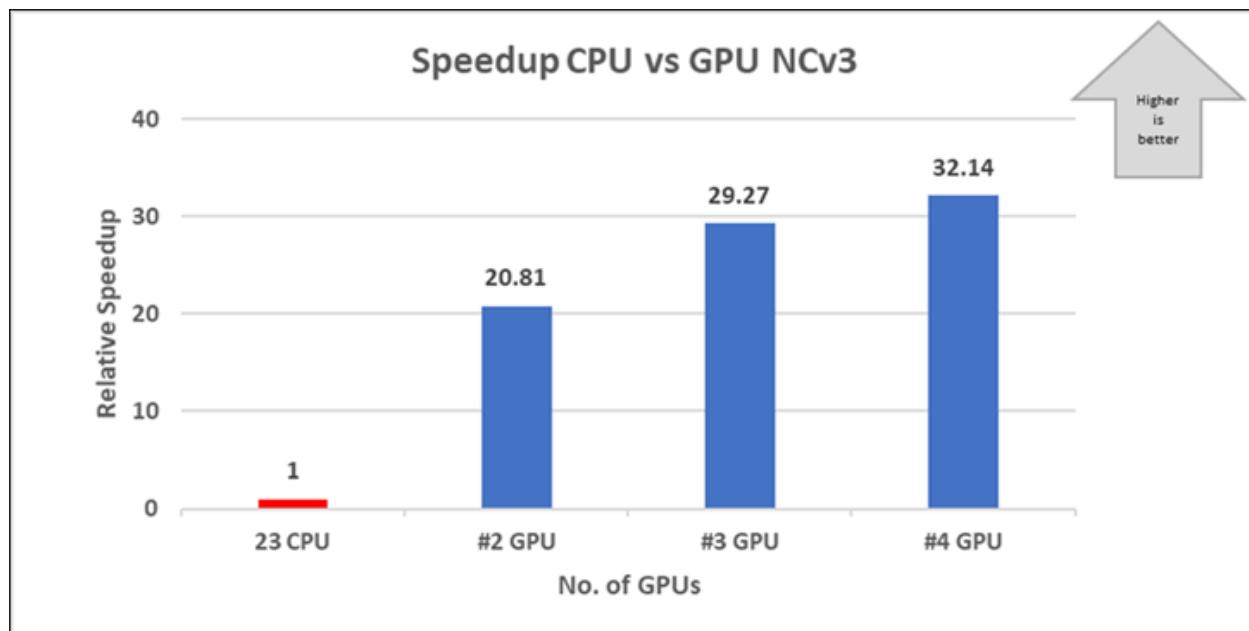
Performance results for NCv3 V100, diffuser and impeller

[Expand table](#)

Number of GPUs	Elapsed time in minutes	Relative improvement
1	-	-
2	173	20.81
4	123	29.27
8	112	32.14

ⓘ Note

The model requires high GPU memory, so the 1-GPU VM couldn't perform the run.



Simulations were performed on Xeon 23 CPUs with 2.4 GHz on V100, which has 4 GPUs, each with 16 GB of memory.

Pricing

Only model running time (wall clock time) is considered for these cost calculations. Application installation time isn't considered. The calculations are indicative. The actual numbers depend on the size of the model.

You can use the [Azure pricing calculator](#) to estimate costs for your configuration.

The following tables provide elapsed times in hours. To compute the total cost, multiply by the Azure VM hourly cost, which you can find [here for Windows](#) and [here for Linux](#).

ND96asr_v4

[[[Expand table](#)

Number of GPUs	Elapsed time in hours
1	2.65
2	1.58
4	1.02
8	0.75

NC24s_v3

[[[Expand table](#)

Number of GPUs	Elapsed time in hours
2	2.88
4	2.05
8	1.87

Additional notes about tests

- Code Leo was successfully tested on NDv4 and NCv3 VMs on Azure.
- The NDv4 A100 VM demonstrated good GPU acceleration. Every added GPU provides speed improvements. The peak performance of 80x is attained with 8 GPUs.
- The NCv3 V100 VM also demonstrated good GPU acceleration. Every added GPU provides good improvements. The peak performance of 32x is attained with 4 GPUs. For complex problems, the 1 GPU memory of 16 GB might not be sufficient, so we recommend 2 GPUs for this scenario.
- The GPU technology in Code Leo provides unprecedented processing power on Azure.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU optimized virtual machine sizes](#)
- [Windows virtual machines in Azure](#)
- [Linux virtual machines in Azure](#)
- [Virtual networks and virtual machines in Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)
- [Code Leo case studies](#)

Related resources

- [Run a Windows VM on Azure](#)
- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Altair AcuSolve on an Azure virtual machine

Azure Virtual Machines Azure Virtual Network Azure CycleCloud

This article briefly describes the steps for running [Altair AcuSolve](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running AcuSolve on Azure.

AcuSolve is a computational fluid dynamics (CFD) analyzer. It provides comprehensive software and tools to solve fluid mechanics problems like thermal analysis, aerodynamics, and noise reduction.

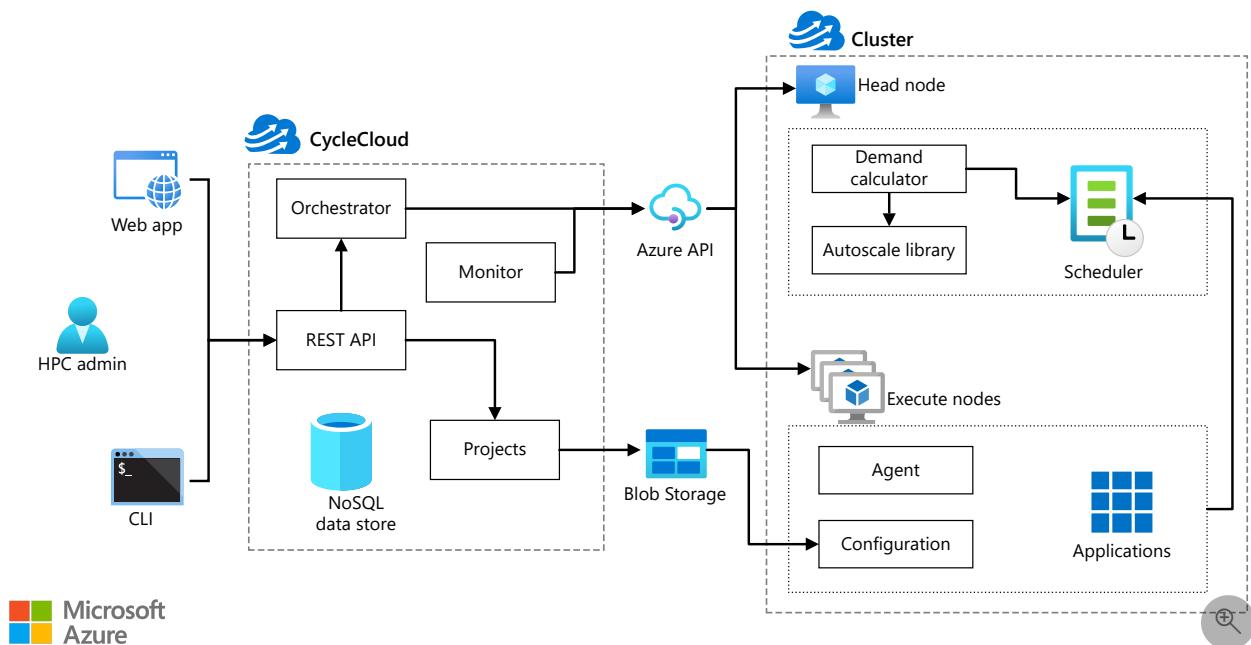
AcuSolve is finite-element based and uses distinct methods to solve all fluid problems: Navier-Stokes, smoothed-particle hydrodynamic, and Lattice Boltzmann. It enables simulations that involve flow, heat transfer, turbulence, and non-Newtonian materials. Engineers and scientists use AcuSolve for the automotive, aerospace, and energy industries, including for chemical processing and electronics cooling.

Why deploy AcuSolve on Azure?

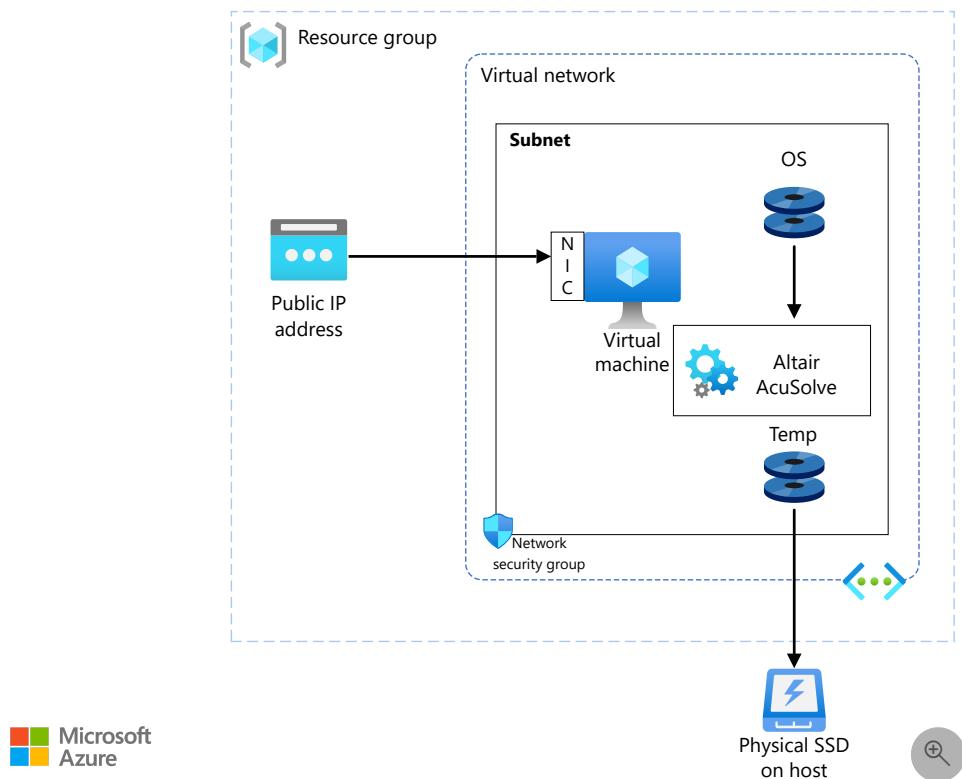
- Modern and diverse compute options to align to your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- On a single node, performance improvements of as much as 2.47 times over that of 16 CPUs

Architecture

This diagram shows a multi-node configuration:



This diagram shows a single-node configuration:



Download a [Visio file](#) of all diagrams in this article.

Components

- [Azure Virtual Machines](#) is used to create Linux VMs.
 - For information about deploying the VM and installing the drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VMs.
 - A public IP address connects the internet to the VM.
- [Azure CycleCloud](#) is used to create the cluster in the multi-node configuration.
- A physical SSD is used for storage.

Compute sizing

[HBv3-series](#) VMs were used to test the performance of AcuSolve on Azure. The following table provides configuration details:

[Expand table](#)

Size	vCPU	RAM memory (GiB)	Memory bandwidth (Gbps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120rs_v3	120	448	350	2.45	3.1	3.675	200	32
Standard_HB120-64rs_v3	64	448	350	2.45	3.1	3.675	200	32

[HBv3-series](#) VMs are optimized for high-performance computing (HPC) applications like fluid dynamics, explicit and implicit finite-element analysis, weather modeling, seismic processing, reservoir simulation, and RTL simulation.

AcuSolve installation

Before you install AcuSolve, you need to deploy and connect a Linux VM.

For information about deploying the VM, see [Run a Linux VM on Azure](#).

You can install AcuSolve from [Altair One Marketplace](#). You also need to install Altair License Manager and activate your license via Altair Units Licensing. You can find more information about installing AcuSolve and License Manager and activating your license on Altair One Marketplace. For multi-node installation, see the next section.

Multi-node configuration

You can easily deploy an HPC cluster on Azure by using [Azure CycleCloud](#).

Azure CycleCloud is a tool for orchestrating and managing HPC environments on Azure. You can use CycleCloud to provision infrastructure for HPC systems, deploy HPC schedulers, and automatically scale the infrastructure to run jobs efficiently at any scale.

Azure CycleCloud is a Linux-based web application. We recommend that you set it up by deploying an Azure VM that's based on a preconfigured Azure Marketplace image.

To set up an HPC cluster on Azure, complete these steps:

1. [Install and configure Azure CycleCloud](#)
2. [Create an HPC cluster from built-in templates](#)
3. [Connect to the head node \(the scheduler\)](#)

For multi-node configurations, the AcuSolve installation process is the same as the process described previously for a single node, except for the path to the installation directory:

- You need to select `/shared` for the Installation directory path so that the directory is accessible for all nodes.
- The shared folder path depends on your network attached storage service, like an NFS server, BeeGFS cluster, [Azure NetApp Files](#), [Azure HPC Cache](#), or [Microsoft Entra Domain Services](#).
- To authorize multi-node VMs to access License Manager, you need to include your authorization code in the job script. For more information about installing AcuSolve, see [Altair One Marketplace](#).

AcuSolve performance results

AcuSolve was tested in single-node and multi-node configurations. Computation time (wall-clock time) was measured. The Linux platform was used, with an Azure Marketplace CentOS 8.1 HPC Gen2 image. The following table provides details:

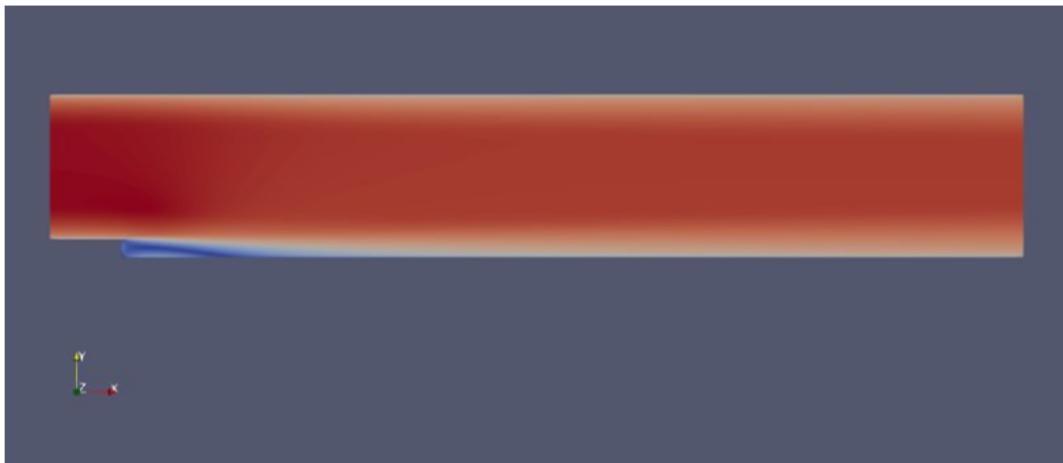
 [Expand table](#)

Operating system version	OS architecture	Processor	MPI
CentOS Linux 8.1.19 (Core)	x86-64	AMD EPYC 7003-series (Milan)	Intel MPI

Results for a single-node configuration

Two models were used to test the single-node configuration:

- Backward-facing step (0.27 million elements)



- Impinging nozzle (7.6 million elements)



The following table provides details about the backward-facing step model:

[Expand table](#)

Analysis type	Turbulence model	Number of elements	Number of nodes	Maximum time steps
Steady state	Spalart-Allmaras	276,936	279,270	300

This table provides details about the impinging nozzle model:

[Expand table](#)

Analysis type	Turbulence model	Number of elements	Number of nodes	Maximum time steps
Steady	Spalart-Allmaras	7,690,844	7,855,017	200

Results for backward-facing step, single node

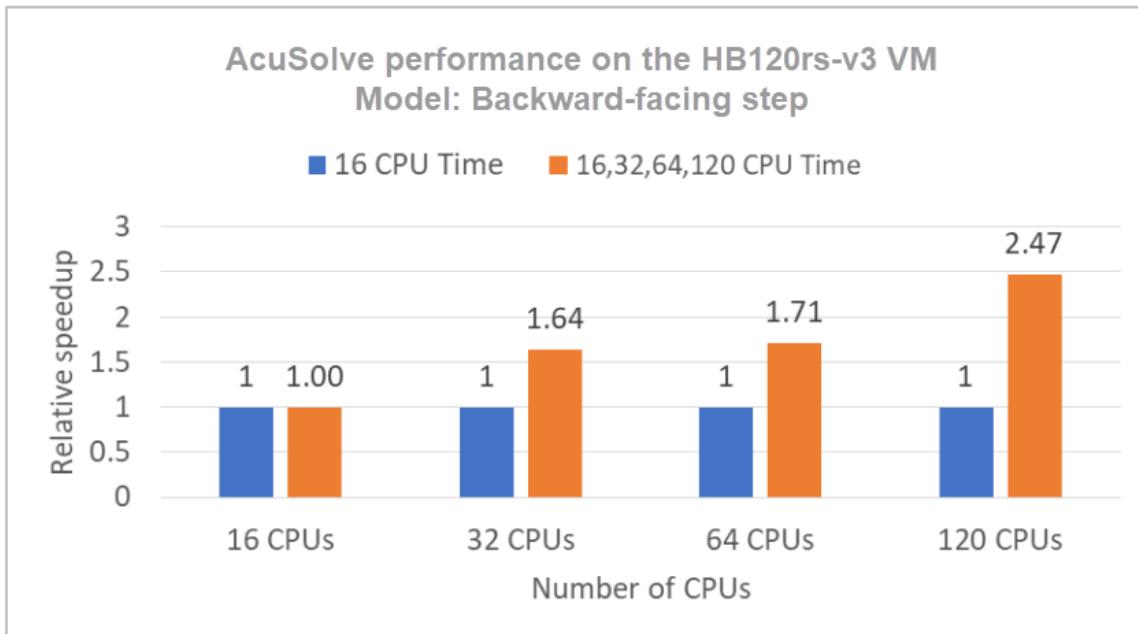
In this benchmarking exercise, AcuSolve simulates turbulent flow over a backward-facing step. The following table shows total consumption times for varying numbers of CPUs on Standard_HB120rs_v3 VMs:

[Expand table](#)

VM size	Number of processors	Wall-clock time (seconds)
Standard_HB120rs_v3	120	130.8
Standard_HB120rs_v3	64 ¹	188.7
Standard_HB120rs_v3	32 ¹	196.4
Standard_HB120rs_v3	16 ¹	322.9

1. In these cases, the number of processors was artificially limited. This VM has 120 processors.

The following graph shows the relative speed increases on the Standard_HB120rs_v3 VM:



This table shows total consumption times for varying numbers of CPUs on Standard_HB120-64rs_v3 VMs:

[Expand table](#)

VM size	Number of processors	Wall-clock time (seconds)
Standard_HB120-64rs_v3	64	127.7
Standard_HB120-64rs_v3	32 ²	199.88
Standard_HB120-64rs_v3	16 ²	266.2

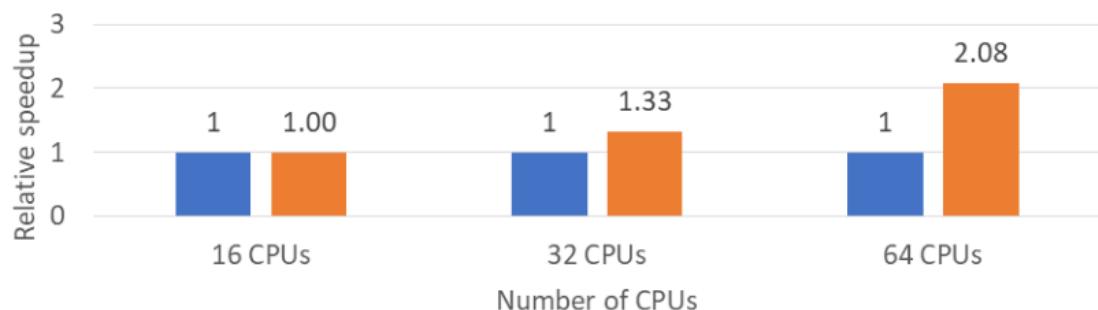
2. In these cases, the number of processors was artificially limited. This VM has 64 processors.

The following graph shows the relative speed increases on the Standard_HB120-64rs_v3 VM:

AcuSolve Speedup on HB120-64rs-v3 VM

Model: Backward Facing step

■ 16 CPU Time ■ 16,32,64 CPU Time



Results for impinging nozzle, single node

As the preceding performance results show, the Standard_HB120-64rs_v3 VM with 64 cores provides the best performance. For the impinging nozzle model, which has more elements, we used that VM for the performance evaluation.

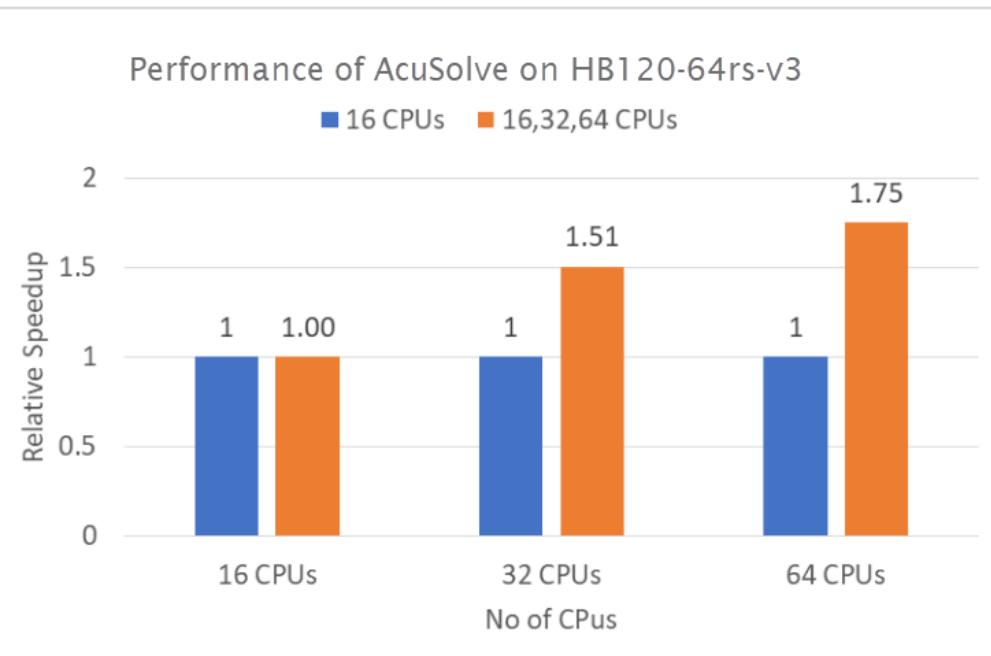
The following table shows total consumption times for varying numbers of CPUs on Standard_HB120-64rs_v3 VMs:

[Expand table](#)

VM size	Number of processors	Wall-clock time (hours)
Standard_HB120-64rs_v3	16^3	13.48
Standard_HB120-64rs_v3	32^3	8.95
Standard_HB120-64rs_v3	64	7.7

3. In these cases, the number of processors was artificially limited. This VM has 64 processors.

The following graph shows the relative speed increases on the Standard_HB120-64rs_v3 VM:



Results for a multi-node configuration

As the preceding performance results show, the Standard_HB120-64rs_v3 VM with 64 cores provides the best performance. For the multi-node configuration, we tested the impinging nozzle model, which has 7.6 million elements, on 1, 2, 4, 8, and 16 nodes of the Standard_HB120-64rs_v3 VM in an Azure HPC cluster.

[Expand table](#)

VM size	Number of nodes	Number of cores	Wall-clock time (seconds)	Relative increase
Standard_HB120-64rs_v3	1	64	24,054	1.15
Standard_HB120-64rs_v3	2	128	9,794	2.83
Standard_HB120-64rs_v3	4	256	4,241	6.53
Standard_HB120-64rs_v3	8	512	2,160	12.83
Standard_HB120-64rs_v3	16	1,024	2,250	12.31

Azure cost

Only rendering time is considered for these cost calculations. Application installation time isn't considered.

To calculate total costs, multiply the wall-clock time by the Azure hourly cost. For the current hourly costs, see [Linux Virtual Machines Pricing](#).

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

Single node on Standard_HB120rs_v3

[Expand table](#)

Number of CPUs	Wall-clock time (hours)
120	0.04

Single node Standard_HB120-64rs_v3

[Expand table](#)

Number of CPUs	Wall-clock time (hours)
64	7.74

Multi-node on Standard_HB120-64rs_v3

[Expand table](#)

Number of nodes	Number of CPUs	Wall-clock time (hours)
1	64	6.68

Number of nodes	Number of CPUS	Wall-clock time (hours)
2	128	2.72
4	256	1.18
8	512	0.60
16	1,024	0.63

Summary

- AcuSolve was successfully tested on Standard_HB120rs_v3 and Standard_HB120-64rs_v3 VMs.
- On a single node, on Standard_HB120rs_v3 VMs with 120 vCPUs, performance increases as much as 2.47 times over that of 16 CPUs, based on wall-clock time.
- On a single node, on Standard_HB120-64rs_v3 VMs with 64 vCPUs, performance increases as much as 2.08 times over that of 16 CPUs, based on wall-clock time.
- AcuSolve scales up linearly with impressive numbers up to 8 nodes on an Azure HPC cluster with Standard_HB120-64rs_v3 VM instances that have 64 cores on each node. Performance increases 12.83 times with an 8-node (512-core) configuration on AMD Milan-X processors, an excellent scale-up value for AcuSolve. Scalability increases beyond 8 nodes when a higher number of finite-element nodes are simulated.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Linux virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)
- [What is Azure CycleCloud?](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Altair EDEM on a virtual machine

Azure Virtual Machines Azure Virtual Network

This article briefly describes the steps for running [Altair EDEM](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running EDEM on Azure.

EDEM is an application that's used for bulk and granular material simulation. EDEM uses discrete element method (DEM) to simulate and analyze the behavior of coal, mined ores, soils, fibers, grains, tablets, and powders.

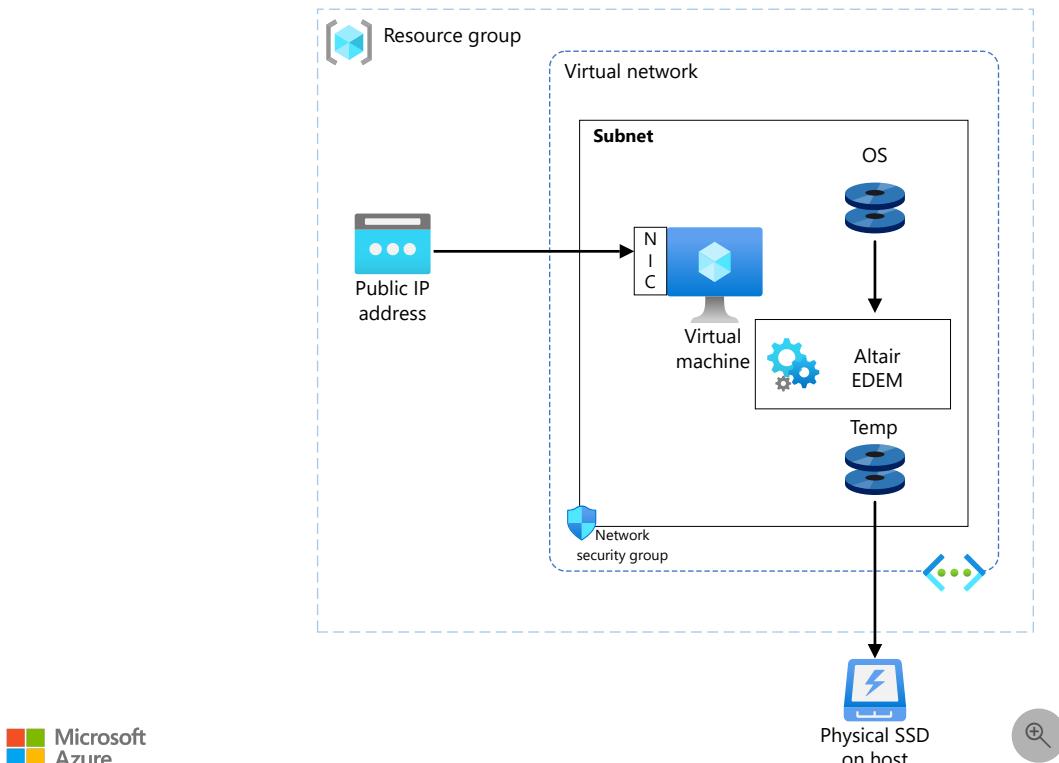
EDEM simulation provides engineers with insight into how those materials interact with equipment during a range of operation and process conditions. It can be used by itself or combined with other CAE tools.

Companies in the heavy equipment, off-road, mining, steelmaking, and process manufacturing industries use EDEM to understand and predict granular material behaviors, evaluate equipment performance, and optimize processes.

Why deploy EDEM on Azure?

- You can use EDEM to model particle shape by using the multi-sphere method.
- EDEM is highly parallelized for use on multi-core shared memory workstations, GPU hardware, and multi-GPU systems.
- The solver engine is fully double precision across all platforms.
- EDEM can simulate large and complex particle systems.
- EDEM provides advanced post-processing capabilities.

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create Windows VMs. For information about deploying VMs and installing the drivers, see [Windows VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) restrict access to the VMs.
 - A public IP address connects the internet to the VMs.

- A physical SSD provides storage.

Compute sizing and drivers

Performance tests of EDEM on Azure used [NCv3](#), [NC A100 v4](#), and [ND A100 v4](#) series VMs running on Windows. The following table provides the configuration details.

[Expand table](#)

Size	vCPUs	Memory, in GiB	Temporary storage (SSD), in GiB	GPUs	GPU memory, in GiB	Maximum data disks	Maximum uncached disk throughput: IOPS / MBps	Maximum NICs
Standard_ND96asr_v4	96	900	6,000	8	40	32	80,000 / 800	8
Standard_NC24ads_A100_v4	24	220	1,123	1	80	12	30,000 / 1,000	2
Standard_NC48ads_A100_v4	48	440	2,246	2	160	24	60,000 / 2,000	4
Standard_NC96ads_A100_v4	96	880	4,492	4	320	32	120,000 / 4,000	8
Standard_NC6s_v3	6	112	736	1	16	12	20,000 / 200	4

Required drivers

To use EDEM on the previously listed VMs as described in this article, you need to install NVIDIA and AMD drivers.

EDEM installation

Before you install EDEM, you need to deploy and connect a VM and install the required NVIDIA and AMD drivers.

For information about deploying the VM and installing the drivers, see [Run a Windows VM on Azure](#).

To download EDEM:

1. Sign in to [Altair One Marketplace](#).
2. Select EDEM in the product list.
3. Select the appropriate operating system and download.
4. Download the license manager.

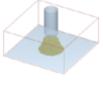
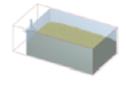
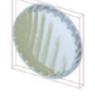
For EDEM installation instructions, see the documents on [Altair One Marketplace](#).

EDEM performance results

Seven models are used to test the performance of EDEM on Azure VMs. The following table provides details.

[Expand table](#)

Model	Angle of repose	Bed of material	Hopper discharge	Powder mixer	Screw augur	Mill	Transfer chute
Description	Cylinder angle of	Bed of material with	Hopper emptying into	Powder mixer	Screw augur	Mill operation	Transfer chute with dynamic

Model	Angle of repose	Bed of material	Hopper discharge	Powder mixer	Screw augur	Mill	Transfer chute
	repose	tillage tool	container	operation	operation		factory
							
Particle radius (m)	0.0005 - 0.001	0.002 - 0.004	0.003	0.0005	0.001	0.005	0.0045 - 0.009
Number of spheres	3	3	3	1	1	1	3
Size distribution	Random	Random	Fixed	Fixed	Fixed	Fixed	Random
Number of particles	1,000,000	1,000,000	1,000,000	1,000,000	1,000,000	1,000,000	1,000,000
Physics	Hertz-Mindlin	Hertz-Mindlin with JKR	Hertz-Mindlin	Hertz-Mindlin	Hertz-Mindlin	Hertz-Mindlin	Hertz-Mindlin with JKR
Time steps	5.73E-06	5.00E-05	4.00E-05	9.20E-06	1.40E-05	0.00016	5.97E-05
Total time	0.5	1	1	1	1	1	1
Save interval	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Grid cell size (x Rmin)	3	3	3	3	3	3	5
Factory	No	No	No	No	No	No	Yes
Periodic boundaries	No	No	No	No	No	No	No

Results for EDEM 2021.1 on NDv4 and NCv3 VMs

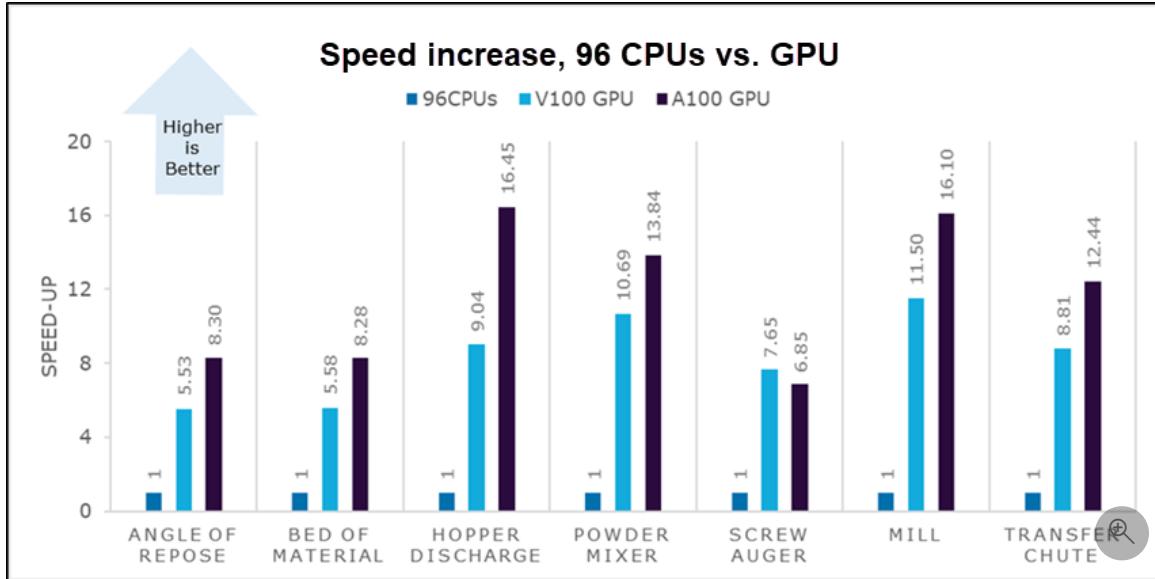
The following table shows the elapsed wall-clock times, in seconds, required to complete each simulation.

[Expand table](#)

Model	ND96asr_v4, 96 CPUs	ND96asr_v4, 1 A100 GPU	NC6s_v3, 1 V100 GPU
Angle of repose	12,819.80	1,543.66	2,319.39
Bed of material	2,650.56	320.24	475.04
Hopper discharge	9,318.89	566.59	1,030.38
Powder mixer	14,028.50	1,013.98	1,312.27
Screw auger	8,871.59	1,295.16	1,158.98

Model	ND96asr_v4, 96 CPUs	ND96asr_v4, 1 A100 GPU	NC6s_v3, 1 V100 GPU
Mill	1,339.11	83.18	116.49
Transfer chute	3,859.01	310.22	437.92

The following graph uses a Standard_ND96asr_v4, 96-vCPU VM as a baseline and shows how much the speed increases on A100-GPU and V100-GPU VMs.



Results for EDEM 2022.1 on NC A100 v4 VMs

The following table shows the elapsed wall-clock times, in seconds, required to complete each simulation.

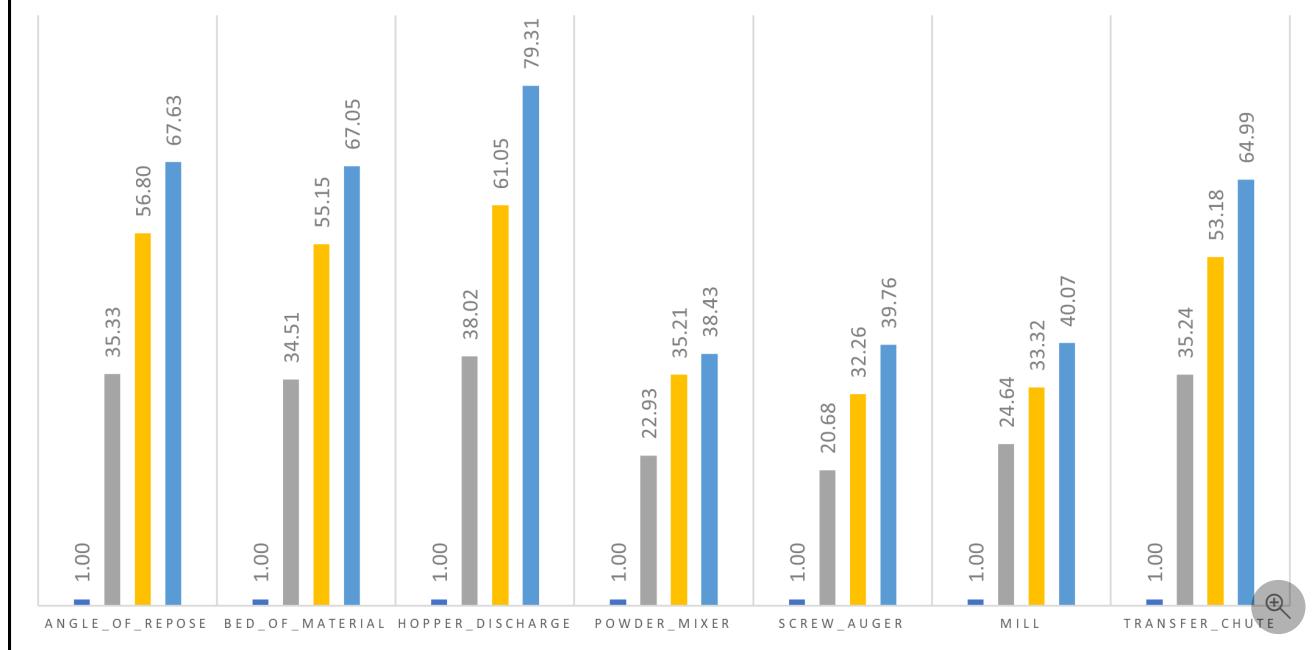
[Expand table](#)

Model	NC24ads_A100_v4, 24 vCPUs	NC24ads_A100_v4, 1 GPU	NC48ads_A100_v4, 2 GPUs	NC96ads_A100_v4, 4 GPUs
Angle of repose	22,950.80	649.59	404.05	339.38
Bed of material	4,835.23	140.10	87.67	72.11
Hopper discharge	11,457.00	301.33	187.68	144.45
Powder mixer	13,906.20	606.43	394.99	361.85
Screw auger	11,089.00	536.27	343.75	278.92
Mill	1,141.65	46.33	34.26	28.49
Transfer chute	4,146.64	117.67	77.98	63.80

The following graph uses a Standard_NC24ads_A100_v4, 24-vCPU VM as a baseline and shows how much the speed increases on VMs with varying numbers of A100 GPUs.

Speed increase, vCPU vs. GPU

■ 24 vCPU ■ 1 GPU ■ 2 GPU ■ 4 GPU



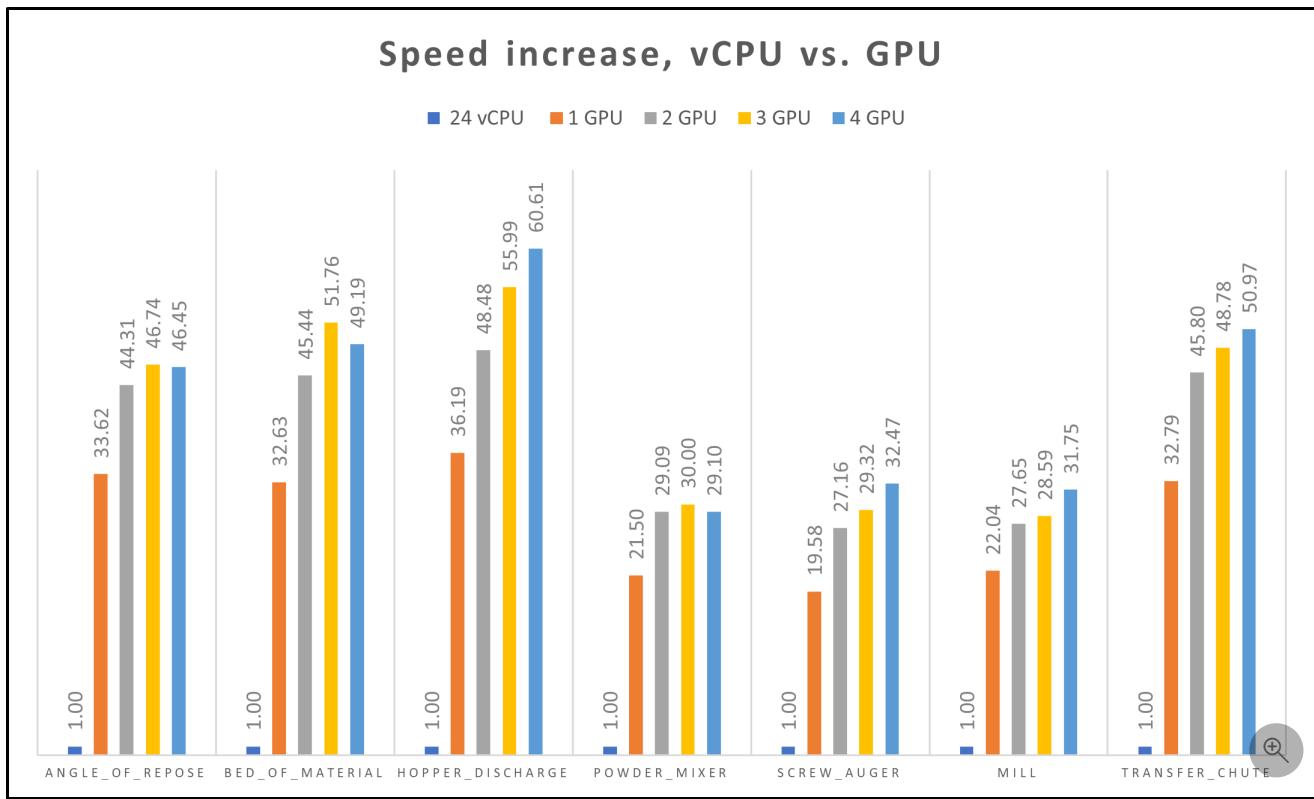
Results for EDEM 2022.1 on ND A100 v4 VMs

The following table shows the elapsed wall-clock times, in seconds, required to complete each simulation.

[Expand table](#)

Model	NC24ads_A100_v4, 24 vCPUs	ND96asr_v4, 1 GPU	ND96asr_v4, 2 GPUs	ND96asr_v4, 3 GPUs	ND96asr_v4, 4 GPUs
Angle of repose	22,950.80	682.66	517.99	491.00	494.08
Bed of material	4,835.23	148.17	106.42	93.42	98.30
Hopper discharge	11,457.00	316.62	236.32	204.62	189.02
Powder mixer	13,906.20	646.77	477.97	463.59	477.86
Screw auger	11,089.00	566.37	408.32	378.17	341.56
Mill	1,141.65	51.79	41.29	39.93	35.96
Transfer chute	4,146.64	126.46	90.54	85.01	81.35

The following graph uses a Standard_NC24ads_A100_v4, 24-vCPU VM as a baseline and shows how much the speed increases on ND96asr_v4 VMs with varying numbers of A100 GPUs.



Azure cost

Only model running time (wall-clock time) is considered for these cost calculations. Application installation time isn't considered. The calculations are indicative. The actual numbers depend on the size of the model.

You can use the [Azure pricing calculator](#) to estimate costs for your configuration.

The following tables provide elapsed times in hours. To compute the total cost, multiply the elapsed time by the Azure VM hourly cost. For current hourly costs, see [Windows Virtual Machines Pricing](#).

EDEM 2021.1 costs on ND96asr_v4 VMs

[Expand table](#)

Model	ND96asr_v4, 96 vCPUs	ND96asr_v4, 1 GPU
Angle of repose	3.56	0.43
Bed of material	0.74	0.09
Hopper discharge	2.59	0.16
Powder mixer	3.90	0.28
Screw auger	2.46	0.36
Mill	0.37	0.02
Transfer chute	1.07	0.09

EDEM 2021.1 costs on NCv3 VMs

[Expand table](#)

Model	NC6s_v3, 1 GPU
Angle of repose	0.64
Bed of material	0.13
Hopper discharge	0.29
Powder mixer	0.36
Screw auger	0.32
Mill	0.03
Transfer chute	0.12

EDEM 2022.1 costs on NC A100 v4 VMs

[Expand table](#)

Model	NC24ads_A100_v4, 24 vCPUs	NC24ads_A100_v4, 1 GPU	NC24ads_A100_v4, 2 GPUs	NC24ads_A100_v4, 4 GPUs
Angle of repose	6.38	0.18	0.11	0.09
Bed of material	1.34	0.04	0.02	0.02
Hopper discharge	3.18	0.08	0.05	0.04
Powder mixer	3.86	0.17	0.11	0.10
Screw auger	3.08	0.15	0.10	0.08
Mill	0.32	0.01	0.01	0.01
Transfer chute	1.15	0.03	0.02	0.02

EDEM 2022.1 costs on ND96asr_v4 VMs

[Expand table](#)

Model	ND96asr_v4, 1 GPU	ND96asr_v4, 2 GPUs	ND96asr_v4, 3 GPUs	ND96asr_v4, 4 GPUs
Angle of repose	0.19	0.14	0.14	0.14
Bed of material	0.04	0.03	0.03	0.03
Hopper discharge	0.09	0.07	0.06	0.05
Powder mixer	0.18	0.13	0.13	0.13
Screw auger	0.16	0.11	0.11	0.09

Model	ND96asr_v4, 1 GPU	ND96asr_v4, 2 GPUs	ND96asr_v4, 3 GPUs	ND96asr_v4, 4 GPUs
Mill	0.01	0.01	0.01	0.01
Transfer chute	0.04	0.03	0.02	0.02

Summary

- EDEM 2021.1 was deployed and tested on ND A100 v4 and NCv3 VMs with one GPU and two GPUs. EDEM 2022.1 was deployed and tested on ND A100 v4 and NC A100 v4 VMs with one GPU and multiple GPUs.
- On Azure, the GPU technology in EDEM provides faster processing than CPU configurations. Tests demonstrate speed increases of about 80x with NC A100 v4 A100 GPUs and about 60x with ND A100 v4 A100 GPUs.
- The complexity of the model affects GPU scale-up.
- The NC A100 v4 VM demonstrates better GPU acceleration than other VM configurations on Azure.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Saurabh Parave](#) | HPC Performance Engineer
- Kalai Selvan | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director of Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- GPU-optimized virtual machine sizes
- Windows virtual machines on Azure
- Virtual networks and virtual machines on Azure
- [Learning path: Run HPC applications on Azure](#)

Related resources

- [Run a Windows VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Altair nanoFluidX on an Azure virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [Altair nanoFluidX](#) (NFX) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running nanoFluidX on Azure.

Altair nanoFluidX simulates single-phase and multiphase flows. It's based on a weakly compressible Lagrangian smoothed-particle hydrodynamics (SPH) formulation. Altair nanoFluidX:

- Enables easy treatment of high-density ratio multiphase flows, like water-air.
- Provides rotating-motion options to prescribe different types of motion. These options enable the simulation of rotating gears, crankshafts, and connecting rods in powertrain applications.

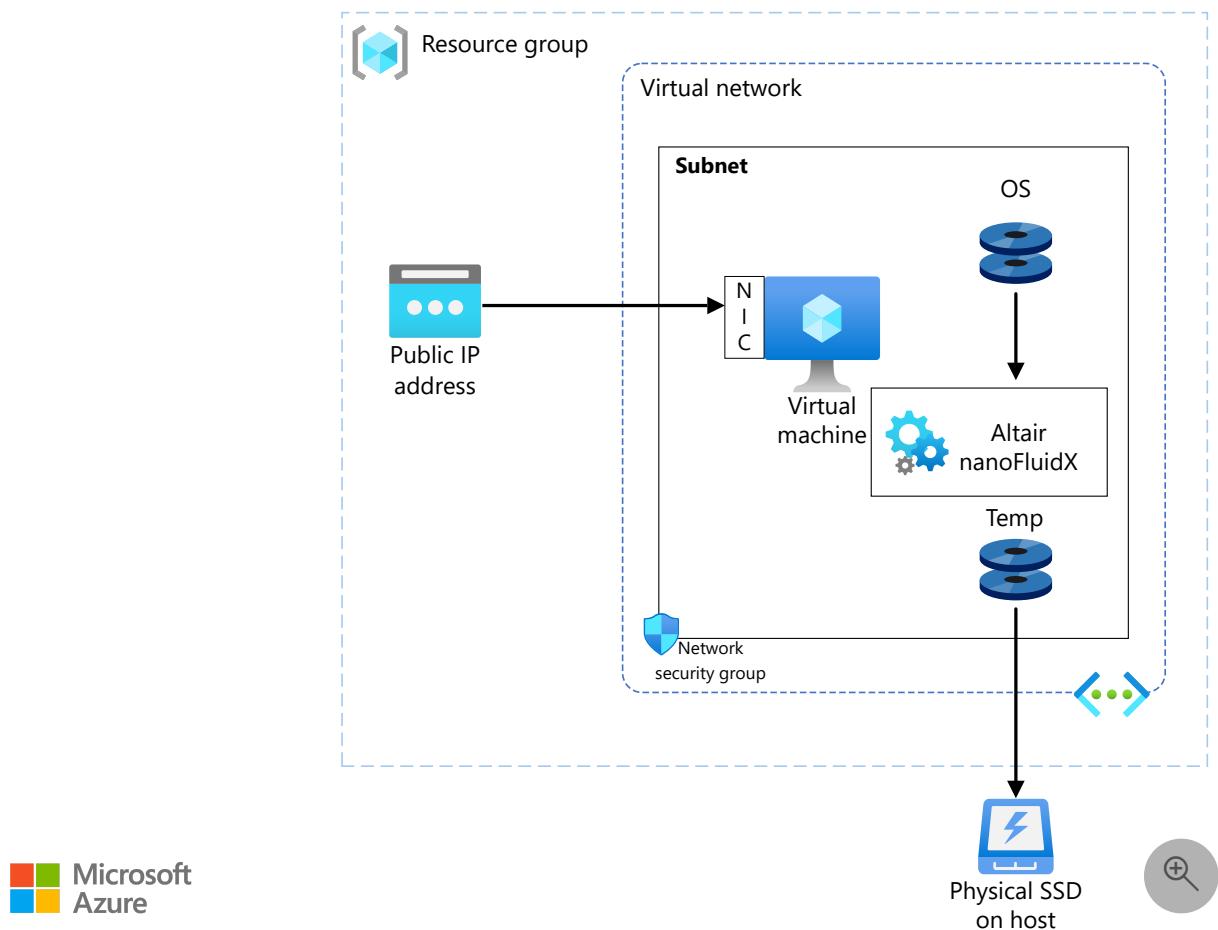
Altair nanoFluidX is designed for use on clusters of graphical processing units (GPUs), so it's fast.

It's primarily used in industries like construction, off-highway, mining (energy), agriculture, space exploration (aerospace), and process/manufacturing. It's best suited for simulation of free surface oiling, sloshing, and mixing.

Why deploy nanoFluidX on Azure?

- Modern and diverse compute options to align with your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Complex problems solved within a few hours

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create Linux VMs.
 - For information about deploying the VM and installing the drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VMs.
 - A public IP address connects the internet to the VM.
- A physical SSD is used for storage.

Compute sizing and drivers

Performance tests of nanoFluidX on Azure used [ND A100 v4](#) and [NVv3 M60](#) series VMs running Linux. The following table provides the configuration details.

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

VM size	vCPU	Memory, in GiB	SSD, in GiB	Number of GPUs	GPU memory, in GiB	Maximum data disks
Standard_ND96asr_v4	96	900	6,000	8 A100	40	32
Standard_NV12s_v3	12	112	320	1	8	12
Standard_NV24s_v3	24	224	640	2	16	24
Standard_NV48s_v3	48	448	1,280	4	32	32

Required drivers

To use nanoFluidX on Standard_ND96asr_v4 VMs, you need to install NVIDIA and AMD drivers.

To use nanoFluidX on NVv3-series VMs, you need to install NVIDIA drivers.

nanoFluidX installation

Before you install nanoFluidX, you need to deploy and connect a Linux VM and install the NVIDIA drivers. On Standard_ND96asr_v4, you need to install the AMD drivers.

 **Important**

NVIDIA Fabric Manager installation is required for Standard_ND96asr_v4 VMs.

For information about deploying the VM and installing the drivers, see [Run a Linux VM on Azure](#).

Altair nanoFluidX only runs on Linux. You can download nanoFluidX from [Altair One Marketplace](#). You also need to install License Manager. For more information, see [Altair One Marketplace](#).

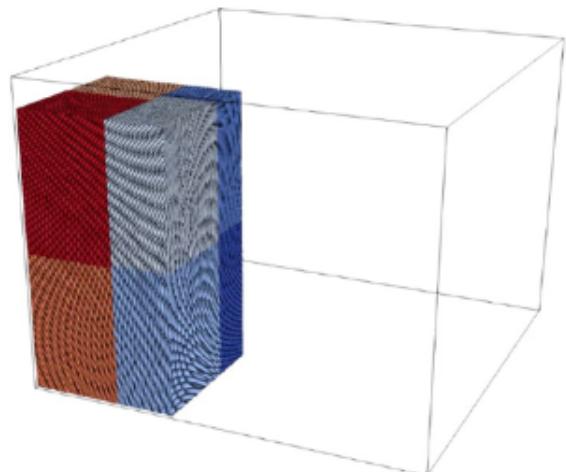
nanoFluidX performance results

Particle-based (SPH) fluid dynamics simulations were run to test nanoFluidX. Because nanoFluidX is GPU-based, the results are provided for a specific number of GPUs.

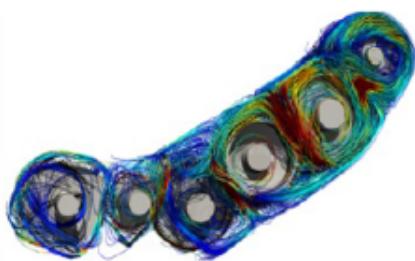
Six models of three different types were used to test nanoFluidX on Azure. This image shows some of the models:



Altair Gear Box



Dam Break



Aero gear Box

The following table shows the number of particles for each model tested.

[Expand table](#)

Model	Number of particles, in millions
Aero_gbx	21
Altair_egbx	6.5
cuboid_192^3	7
cuboid_198^3	8
dambreak_dx0001	54
dambreak_dx0002	7

Results for ND A100 v4

The following table provides the details of the ND A100 v4 VM that was used for testing.

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

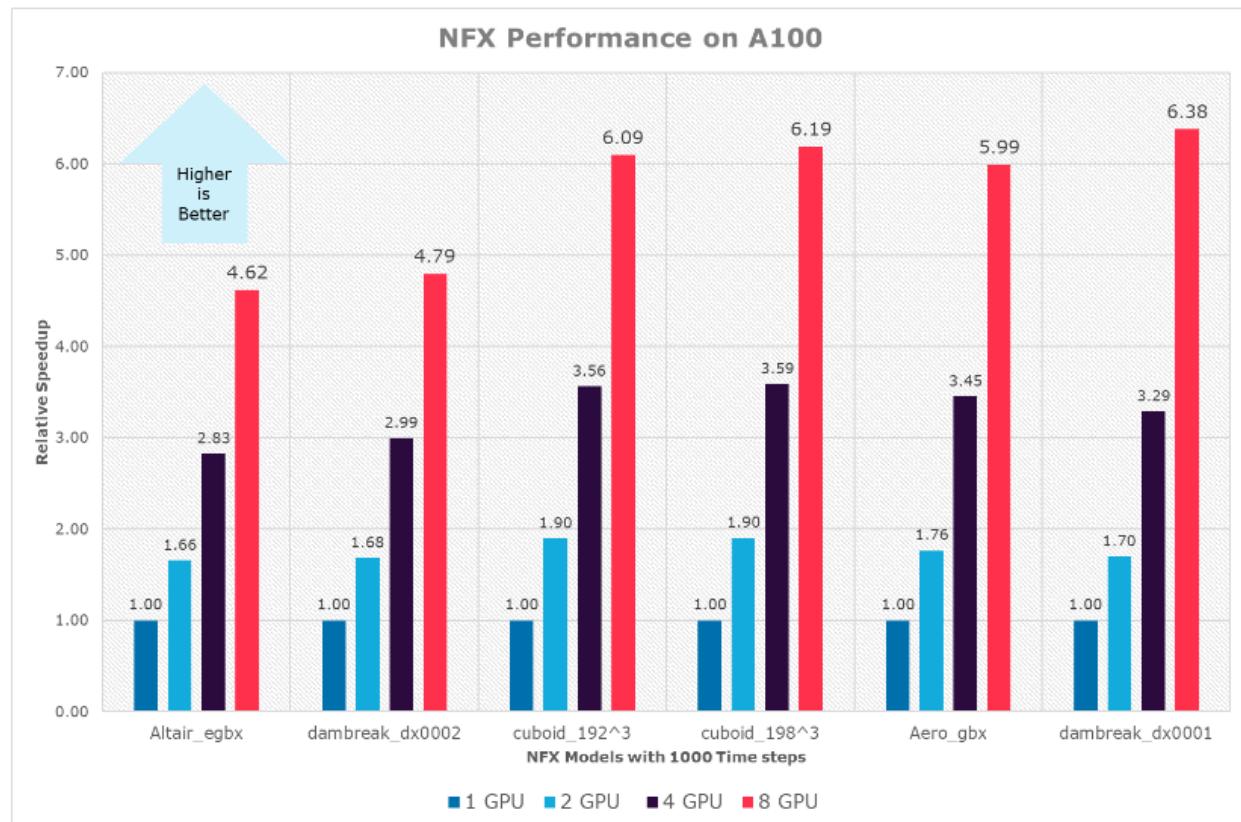
VM series	Operating system	OS architecture	GPU driver version	Cuda version
ND A100 v4	CentOS Linux release 8.1.1911 (Core)	x86-64	470.57.02	11.4

The following table presents the results in wall-clock time, in seconds, for 1,000 time steps.

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

Model	Number of particles, in millions	1 GPU	2 GPUs	4 GPUs	8 GPUs
Altair_egbx	6.5	81.62	49.26	28.88	17.67
dambreak_dx0002	7	59.58	35.39	19.90	12.43
cuboid_192^3	7	47.24	24.90	13.25	7.75
cuboid_198^3	8	51.73	27.20	14.41	8.36
Aero_gbx	21	263.76	149.49	76.47	44.04
dambreak_dx0001	54	413.15	243.03	125.56	64.75

This graph shows the relative speed increases for each increase in GPU:



Results for NVv3 M60

The following table provides the details of the NVv3 M60 VM that was used for testing.

[Expand table](#)

VM series	Operating system	OS architecture	GPU driver version	Cuda version
NVv3 M60	Red Hat Enterprise Linux release 8.2 (Ootpa)	x86-64	470.57.02	11.4

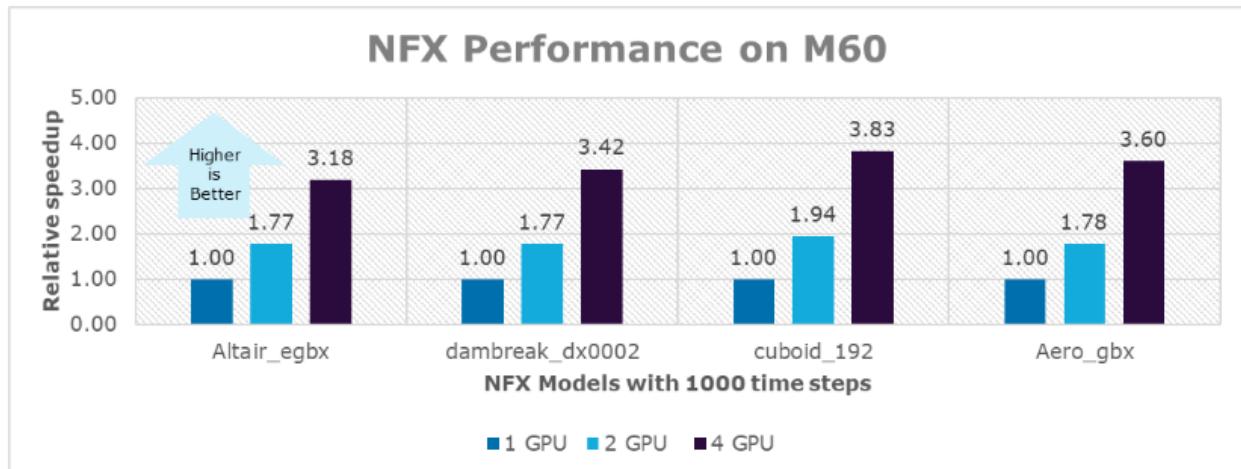
The following table presents the results in wall-clock time, in seconds, for 1,000 time steps.

[Expand table](#)

Model	Number of particles, in millions	1 GPU	2 GPUs	4 GPUs
Aero_gbx	21	1712	962	475
Altair_egbx	6.5	486	275	153

Model	Number of particles, in millions	1 GPU	2 GPUs	4 GPUs
cuboid_192^3	7	291	150	76
dambreak_dx0002	7	376	213	110

This graph shows the relative speed increases for each increase in GPU:



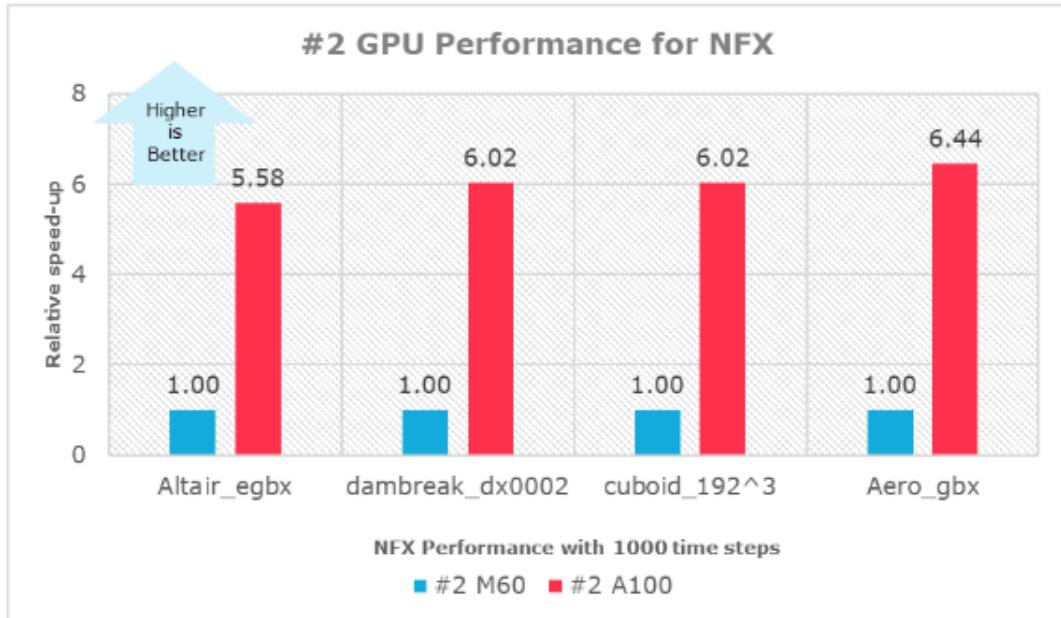
ND A100 v4 vs. NVv3 M60

This table compares the ND A100 v4 and NVv3 M60 results. Results are in wall-clock time, in seconds, with two GPUs. 1,000 time steps were used.

[Expand table](#)

Model	NVv3 M60	NDV4 A100
Altair_egbx	275	49.26
dambreak_dx0002	213	35.39
cuboid_192^3	150	24.90
Aero_gbx	962	149.49

This graph shows the relative speed increases of ND A100 v4 over NVV3 M60:

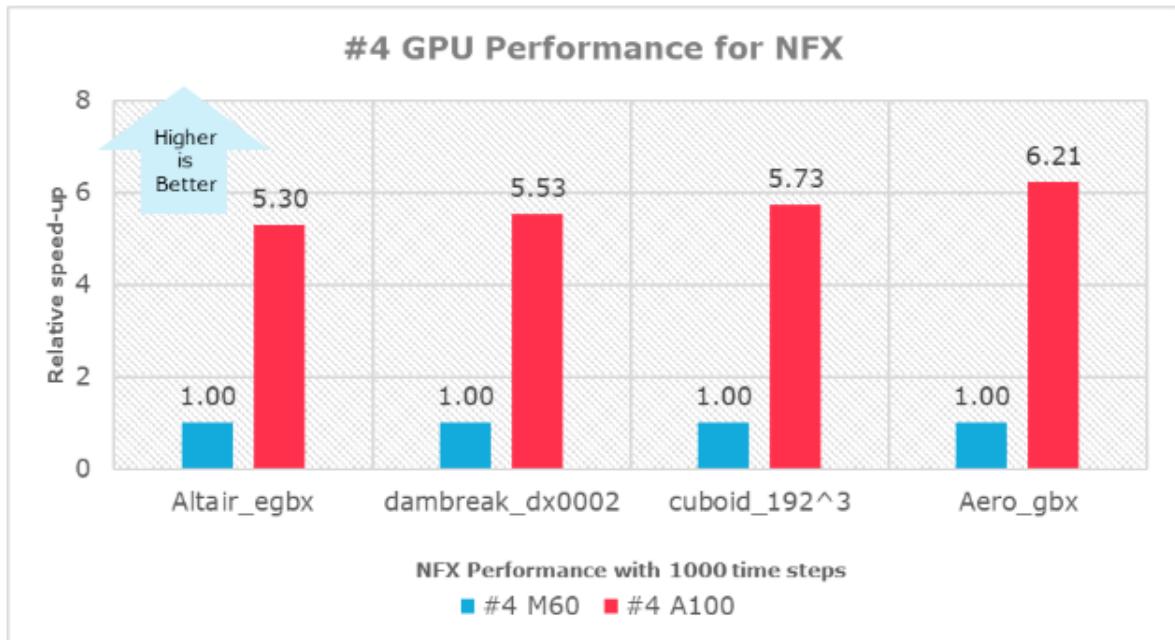


Here's the same comparison with four GPUs:

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

Model	NVv3 M60	NDV4 A100
Altair_egbx	153	28.88
dambreak_dx0002	110	19.90
cuboid_192^3	76	13.25
Aero_gbx	475	76.47

And here's a graph that shows the performance increases:



Additional notes about tests

Altair nanoFluidX 2021.2 was used in these tests. The simulations were run for 1,000 time steps and 10,000 time steps, not for the full simulation times. The performance increase in both cases is almost the same, so the tables and graphs here present only the results for 1,000 time steps.

Azure cost

Only rendering time is considered for these cost calculations. Application installation time isn't considered.

You can use the wall-clock time presented in the following tables and the Azure hourly rate to calculate costs. For the current hourly costs, see [Linux Virtual Machines Pricing](#).

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

The following tables present the wall-clock time for the simulations on Standard_ND96asr_v4 VMs, by number of GPUs.

[] [Expand table](#)

VM size	Model	Time steps	Number of GPUs	Wall-clock time, in seconds
Standard_ND96asr_v4	Aero_gbx	1,000	1	263.76
Standard_ND96asr_v4	Altair_egbx	1,000	1	81.62

VM size	Model	Time steps	Number of GPUs	Wall-clock time, in seconds
Standard_ND96asr_v4	dambreak_dx0001	1,000	1	413.15
Standard_ND96asr_v4	dambreak_dx0002	1,000	1	59.58
Standard_ND96asr_v4	cuboid_192^3	1,000	1	47.24
Standard_ND96asr_v4	cuboid_198^3	1,000	1	51.73

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

VM size	Model	Time steps	Number of GPUs	Wall-clock time, in seconds
Standard_ND96asr_v4	Aero_gbx	1,000	2	149.49
Standard_ND96asr_v4	Altair_egbx	1,000	2	49.26
Standard_ND96asr_v4	dambreak_dx0001	1,000	2	243.03
Standard_ND96asr_v4	dambreak_dx0002	1,000	2	35.39
Standard_ND96asr_v4	cuboid_192^3	1,000	2	24.90
Standard_ND96asr_v4	cuboid_198^3	1,000	2	27.20

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

VM size	Model	Time steps	Number of GPUs	Wall-clock time, in seconds
Standard_ND96asr_v4	Aero_gbx	1,000	4	76.47
Standard_ND96asr_v4	Altair_egbx	1,000	4	28.88

VM size	Model	Time steps	Number of GPUs	Wall-clock time, in seconds
Standard_ND96asr_v4	dambreak_dx0001	1,000	4	125.56
Standard_ND96asr_v4	dambreak_dx0002	1,000	4	19.90
Standard_ND96asr_v4	cuboid_192^3	1,000	4	13.25
Standard_ND96asr_v4	cuboid_198^3	1,000	4	14.41

[Expand table](#)

VM size	Model	Time steps	Number of GPUs	Wall-clock time, in seconds
Standard_ND96asr_v4	Aero_gbx	1,000	8	44.04
Standard_ND96asr_v4	Altair_egbx	1,000	8	17.67
Standard_ND96asr_v4	dambreak_dx0001	1,000	8	64.75
Standard_ND96asr_v4	dambreak_dx0002	1,000	8	12.43
Standard_ND96asr_v4	cuboid_192^3	1,000	8	7.75
Standard_ND96asr_v4	cuboid_198^3	1,000	8	8.36

These tables present the wall-clock time for the simulations on various NVv3 M60 VMs. Each VM has a different number of GPUs.

[Expand table](#)

VM size	Model	Time steps	Number of GPUs	Wall-clock time, in seconds
Standard_NV12s_v3*	Aero_gbx	1,000	1	1,712

VM size	Model	Time steps	Number of GPUs	Wall-clock time, in seconds
Standard_NV12s_v3*	Altair_egbx	1,000	1	486
Standard_NV12s_v3*	cuboid_192^3	1,000	1	291
Standard_NV12s_v3*	dambreak_dx0002	1,000	1	376

*Because of GPU memory requirements, some test cases didn't run to completion on the Standard_NV12s_v3Nvv3 VM with one GPU.

[Expand table](#)

VM size	Model	Time steps	Number of GPUs	Wall-clock time, in seconds
Standard_NV24s_v3	Aero_gbx	1,000	2	962
Standard_NV24s_v3	Altair_egbx	1,000	2	275
Standard_NV24s_v3	cuboid_192^3	1,000	2	150
Standard_NV24s_v3	dambreak_dx0002	1,000	2	213

[Expand table](#)

VM size	Model	Time steps	Number of GPUs	Wall-clock time, in seconds
Standard_NV48s_v3	Aero_gbx	1,000	4	475
Standard_NV48s_v3	Altair_egbx	1,000	4	153
Standard_NV48s_v3	cuboid_192^3	1,000	4	76
Standard_NV48s_v3	dambreak_dx0002	1,000	4	110

Finally, this table consolidates some of the preceding information. Standard_ND96asr_v4 provides better performance.

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

VM size	Number of test cases	Number of GPUs	Wall-clock time, in hours
Standard_ND96asr_v4	6	1	0.25
Standard_ND96asr_v4	6	2	0.15
Standard_ND96asr_v4	6	4	0.08
Standard_ND96asr_v4	6	8	0.04
Standard_NV12s_v3	4	1	0.80
Standard_NV24s_v3	4	2	0.44
Standard_NV48s_v3	4	4	0.23

Summary

- Altair nanoFluidX was successfully tested on ND A100 v4 and NVv3 M60 series VMs on Azure.
- Standard_ND96asr_v4 provides the best performance.
- Simulations for complex workloads are solved within a few hours on ND A100 v4 VMs.
- Increasing the number of GPUs improves performance.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- GPU-optimized virtual machine sizes
- Linux virtual machines on Azure
- Virtual networks and virtual machines on Azure
- Learning path: Run high-performance computing (HPC) applications on Azure

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Altair Radioss on an Azure virtual machine

Azure Virtual Machines Azure Virtual Network Azure CycleCloud

This article briefly describes the steps for running [Altair Radioss](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running Radioss on Azure.

Radioss is a multidisciplinary finite-element solver for linear and nonlinear problems. It's used to predict crash response and dynamic, transient-loading effects on vehicles, structures, and other products. Radioss:

- Uses battery and module macro models for crash events, road debris impacts, and shocks to simulate mechanical failures that cause electrical short circuits, thermal runaway, and risk of fire.
- Provides a composite shell element with delamination tracking and a fast parabolic tetra element.
- Implements extensive material laws and rupture criteria for crack propagation in brittle materials like windshields.
- Provides a fast solution for airbag deployment that uses finite-volume method technology.

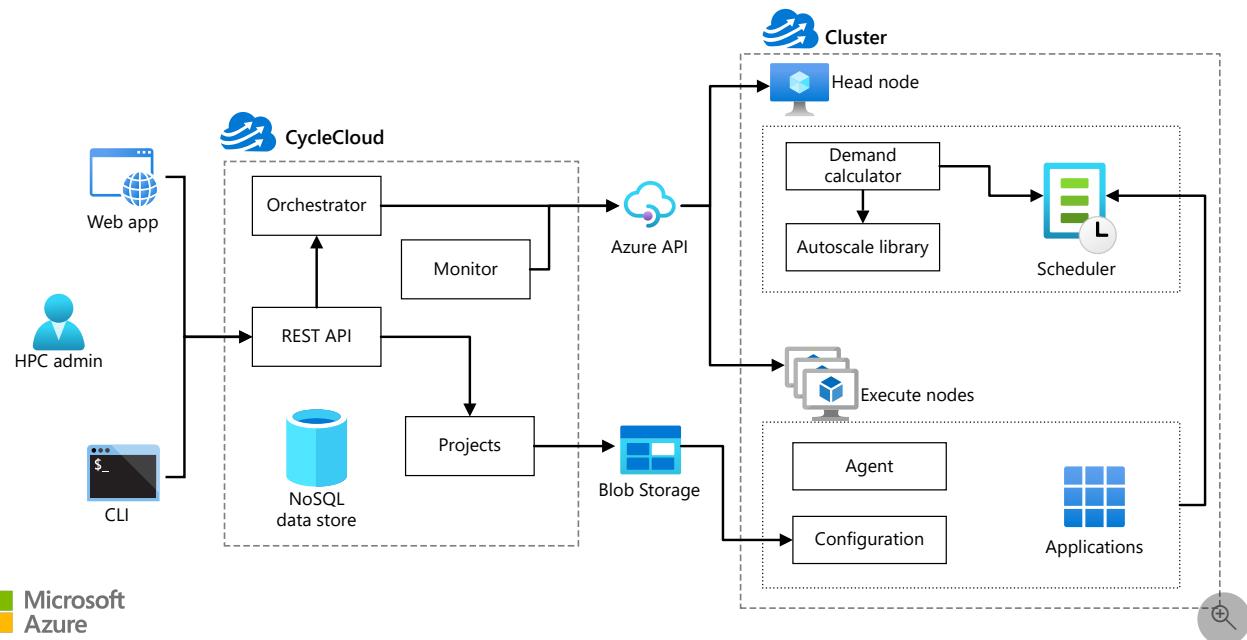
Radioss is used across industry sectors to provide multiphysics solutions to dynamic problems that combine structures, mechanisms, fluids, and thermal and electromagnetic effects. It's ideal for the automotive, aerospace, and energy industries.

Why deploy Radioss on Azure?

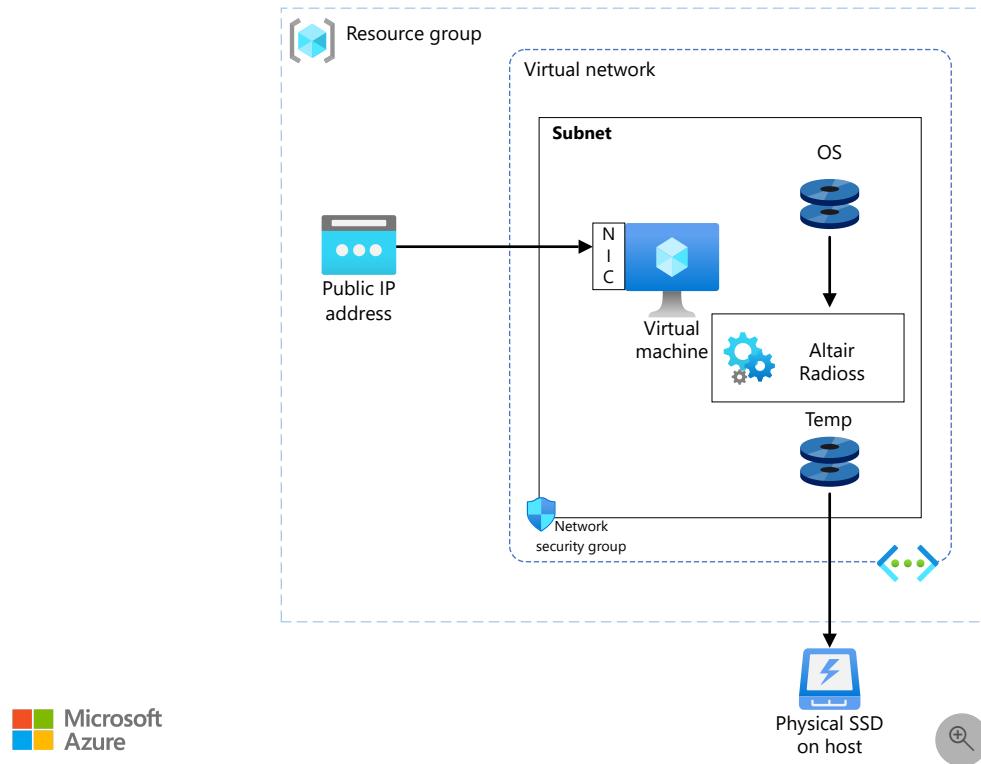
- Modern and diverse compute options to align with your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- On a single node, performance improvements of as much as 2.76 times over that of 16 CPUs

Architecture

This architecture shows a multi-node configuration, orchestrated with Azure CycleCloud:



This architecture shows a single-node configuration:



Download a [Visio file](#) of all diagrams in this article.

Components

- [Azure Virtual Machines](#) is used to create a Linux VM.
 - For information about deploying the VM and installing the drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- [Azure CycleCloud](#) is used to create the cluster in the multi-node configuration.
- A physical SSD is used for storage.

Compute sizing and drivers

Performance tests of Radioss on Azure used [HBv3-series](#) VMs running Linux. The following table provides the configuration details.

[Expand table](#)

VM size	vCPU	RAM memory (GiB)	Memory bandwidth (Gbps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120rs_v3	120	448	350	1.9	3.0	3.5	200	32
Standard_HB120-96rs_v3	96	448	350	1.9	3.0	3.5	200	32
Standard_HB120-64rs_v3	64	448	350	1.9	3.0	3.5	200	32
Standard_HB120-32rs_v3	32	448	350	1.9	3.0	3.5	200	32
Standard_HB120-16rs_v3	16	448	350	1.9	3.0	3.5	200	32

VM size	vCPU	RAM memory (GiB)	Memory bandwidth (GBps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (GBps)	Maximum data disks
16rs_v3								

HBv3-series VMs are optimized for HPC applications like fluid dynamics, explicit and implicit finite-element analysis, weather modeling, seismic processing, reservoir simulation, and RTL simulation.

HBv3 VMs with different numbers of vCPUs were deployed to determine the optimal configuration for Radioss test simulations on a single node. That optimal configuration was then tested in a multi-node cluster deployment.

Required drivers

To use the AMD CPUs on [HBv3-series](#) VMs, you need to install AMD drivers.

Radioss installation

Before you install Radioss, you need to deploy and connect a Linux VM and install the required AMD drivers.

For information about deploying the VM, see [Run a Linux VM on Azure](#).

You can install Radioss from [Altair One Marketplace](#). You also need to install Altair License Manager and activate your license via Altair Units Licensing. See the Altair Units Licensing document on [Altair One Marketplace](#). You can find more information about installing Radioss and License Manager and activating your license on Altair One Marketplace. For multi-node installation, see the next section.

Multi-node configuration

You can easily deploy an HPC cluster on Azure by using [Azure CycleCloud](#).

Azure CycleCloud is a tool for orchestrating and managing HPC environments on Azure. You can use CycleCloud to provision infrastructure for HPC systems, deploy HPC schedulers, and automatically scale the infrastructure to run jobs efficiently at any scale.

Azure CycleCloud is a Linux-based web application. We recommend that you set it up by deploying an Azure VM that's based on a preconfigured Azure Marketplace image.

To set up an HPC cluster on Azure, complete these steps:

1. [Install and configure Azure CycleCloud](#).
2. [Create an HPC cluster from built-in templates](#).
3. [Connect to the head node \(the scheduler\)](#).

For multi-node configurations, the Radioss installation process is the same as the process described previously for a single node, except for the path to the installation directory:

- You need to select `/shared` for the installation directory path so that the directory is accessible for all nodes.
- The shared folder path depends on your network attached storage service, like an NFS server, BeeGFS cluster, [Azure NetApp Files](#), [Azure HPC Cache](#), or [Microsoft Entra Domain Services](#).
- To authorize multi-node VMs to access License Manager, include your authorization code in the job script. For more information about installing Radioss, see [Altair One Marketplace](#).

Radioss performance results

Radioss was tested in single-node and multi-node configurations. Computation time (wall-clock time) was measured. The Linux platform was used, with an Azure Marketplace CentOS 8.1 HPC Gen2 image. The following table provides details.

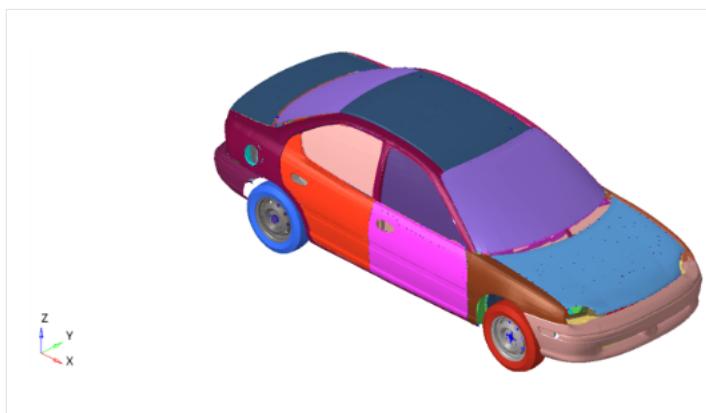
[Expand table](#)

Operating system version	OS architecture	MPI
CentOS Linux release 8.1.1911 (Core)	x86-64	Intel MPI

Results for a single-node configuration

Nonlinear finite-element analysis was performed to test Radioss on a single node with various numbers of CPUs. See the table in the [Compute sizing and drivers](#) section of this article for details about the VMs.

The Neon model was used as a test case:



The following table provides the numbers of various elements in the model.

[Expand table](#)

Nodal points	Parts	Materials	Property sets	3D solid elements	3D shell elements (four nodes)	3D beam elements	3D spring elements	3D shell elements (three nodes)	Accelerometers
1,096,865	340	21	148	2,860	1,054,611	63	4,180	176	7

The following table presents the results, in wall-clock time, in seconds.

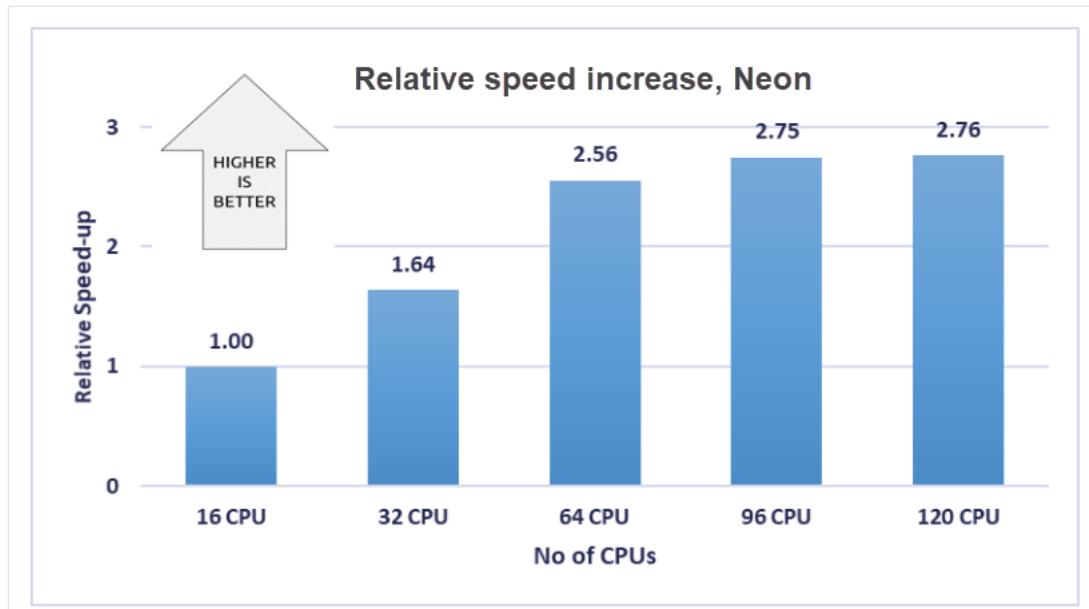
[Expand table](#)

Model	Simulation time (ms)		16 CPUs	32 CPUs	64 CPUs	96 CPUs	120 CPUs
Neon	8	Starter	20.27	22.91	25.78	29.18	31.46
Neon	8	Engine	421.99	246.65	147.31	131.74	128.74
Neon	8	Total runtime	442.26	269.56	173.09	160.92	160.2

The following table shows the relative speed increase for each increase in number of CPUs.

[Expand table](#)

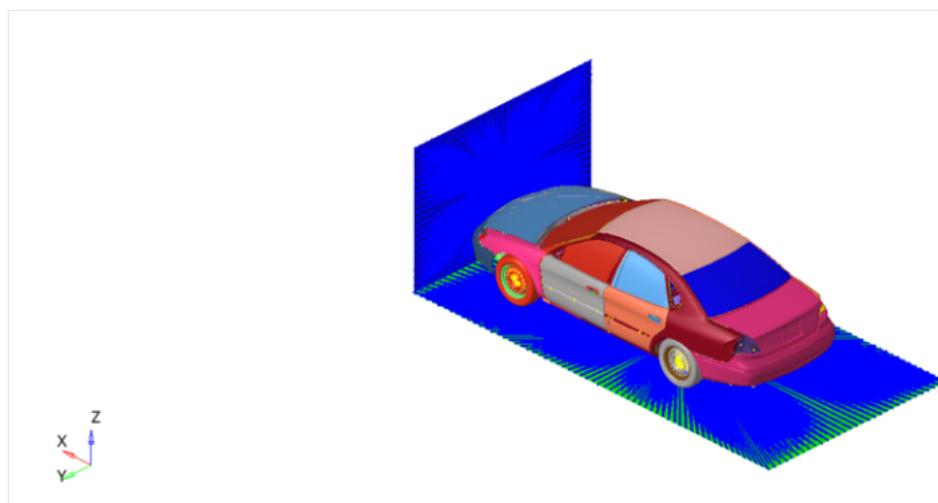
Model	16 CPUs	32 CPUs	64 CPUs	96 CPUs	120 CPUs
Neon	1.00	1.64	2.56	2.75	2.76



Results for a multi-node configuration

The initial performance testing for the multi-node configuration was performed on Radioss 2021.2. This article also includes test results for Radioss 2022.1, which is the most recent version.

The Taurus T10M model was used as a test case:



The following table provides the numbers of various elements in the model.

[Expand table](#)

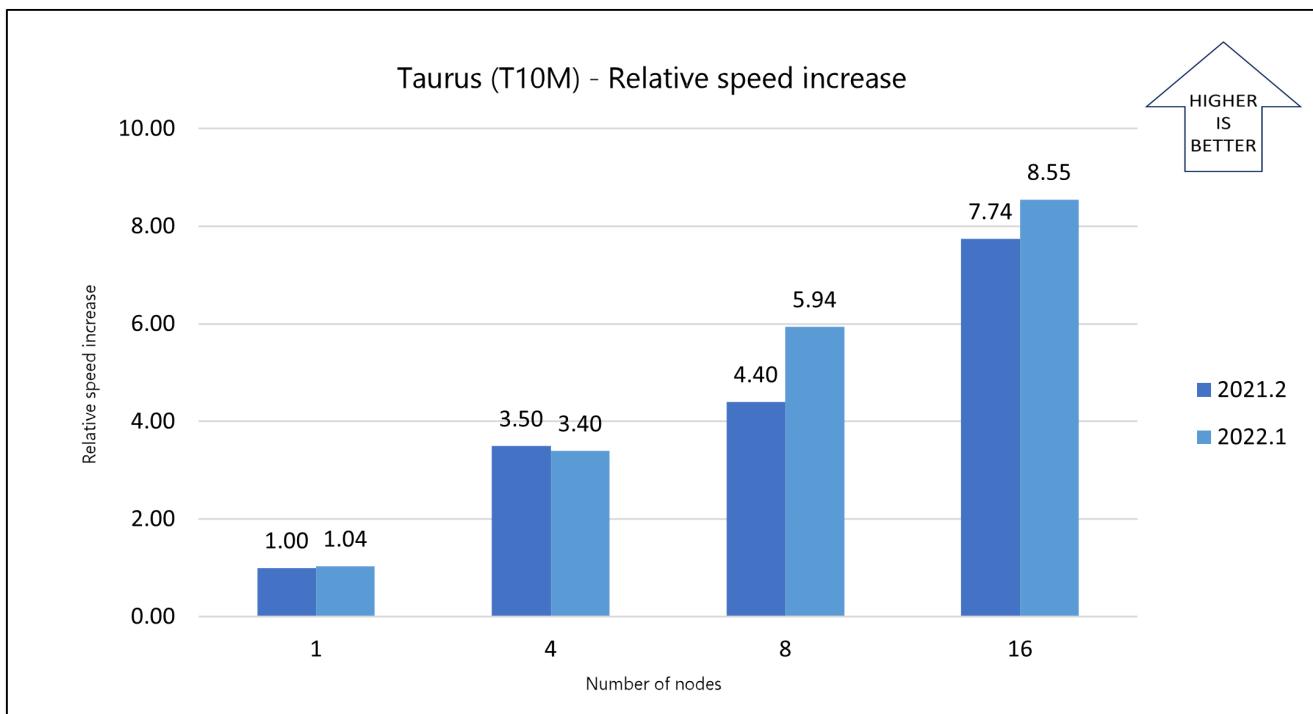
Nodal points	Parts	Materials	Property sets	Boundary conditions	3D solid elements	3D shell elements (four nodes)	3D beam elements	3D spring elements	3D shell elements (three nodes)	Grav loads
9,754,355	1,585	66	762	1	330,418	9,196,272	3,766	417	345,409	1

For the Taurus T10M model, as the preceding performance results show, a Standard_HB120-64rs_v3 VM (AMD EPYC 7V73X Milan-X processors) with 64 cores is the optimal configuration. This configuration was used in the multi-node tests. 64 cores were used on each node.

The following table shows the elapsed wall-clock times for the test runs.

[Expand table](#)

Model	Number of cycles	VM size	Number of nodes	Number of CPUs	Number of threads	Version 2021.2 runtime (seconds)	Version 2022.1 runtime (seconds)
Taurus (T10M)	603,079	Standard_HB120-64rs_v3	1	64	1	135,417.00	130,768.10
Taurus (T10M)	603,079	Standard_HB120-64rs_v3	4	256	1	38,726.00	39,871.03
Taurus (T10M)	603,079	Standard_HB120-64rs_v3	8	512	4	30,756.00	22,801.05
Taurus (T10M)	603,079	Standard_HB120-64rs_v3	16	1024	4	17,486.00	15,838.56



Azure cost

Only rendering time is considered for these cost calculations. Application installation time isn't considered.

You can use the wall-clock time and the Azure hourly cost to compute total costs. For the current hourly costs, see [Linux Virtual Machines Pricing](#).

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

The following table provides the wall-clock times for single-node configurations.

[Expand table](#)

VM size	Model	Number of CPUs	Wall clock time (seconds)
HB120rs_v3	Neon	16	442.26
HB120rs_v3	Neon	32	269.56
HB120rs_v3	Neon	64	173.09
HB120rs_v3	Neon	96	160.92

VM size	Model	Number of CPUs	Wall clock time (seconds)
HB120rs_v3	Neon	120	160.2

The following table provides the wall-clock times for multi-node configurations.

[Expand table](#)

VM size	Model	Number of CPUs	Number of nodes	Wall clock time (hours)
HB120-64rs_v3	Taurus (T10M)	64	1	36:19:28
HB120-64rs_v3	Taurus (T10M)	256	4	11:04:31
HB120-64rs_v3	Taurus (T10M)	512	8	06:20:01
HB120-64rs_v3	Taurus (T10M)	1024	16	04:23:58

Summary

- Radioss was successfully tested on HBv3-series VMs on Azure.
- Radioss on an Azure VM can solve complex workloads.
- In a single-node configuration, increasing the number of CPUs increases the relative speed. The optimal configuration is 64 CPUs.
- Radioss scales impressively up to 16 nodes (1024 CPUs) for the Taurus T10M model.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer
- [Vivi Richard](#) | HPC Performance Engineer

Other contributors:

- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see nonpublic LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Linux virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)
- [What is Azure CycleCloud?](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy ultraFluidX on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [Altair ultraFluidX](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running ultraFluidX on Azure.

Altair ultraFluidX is a simulation tool for predicting the aerodynamic properties of passenger and heavy-duty vehicles, and for the evaluation of building and environmental aerodynamics. Altair ultraFluidX:

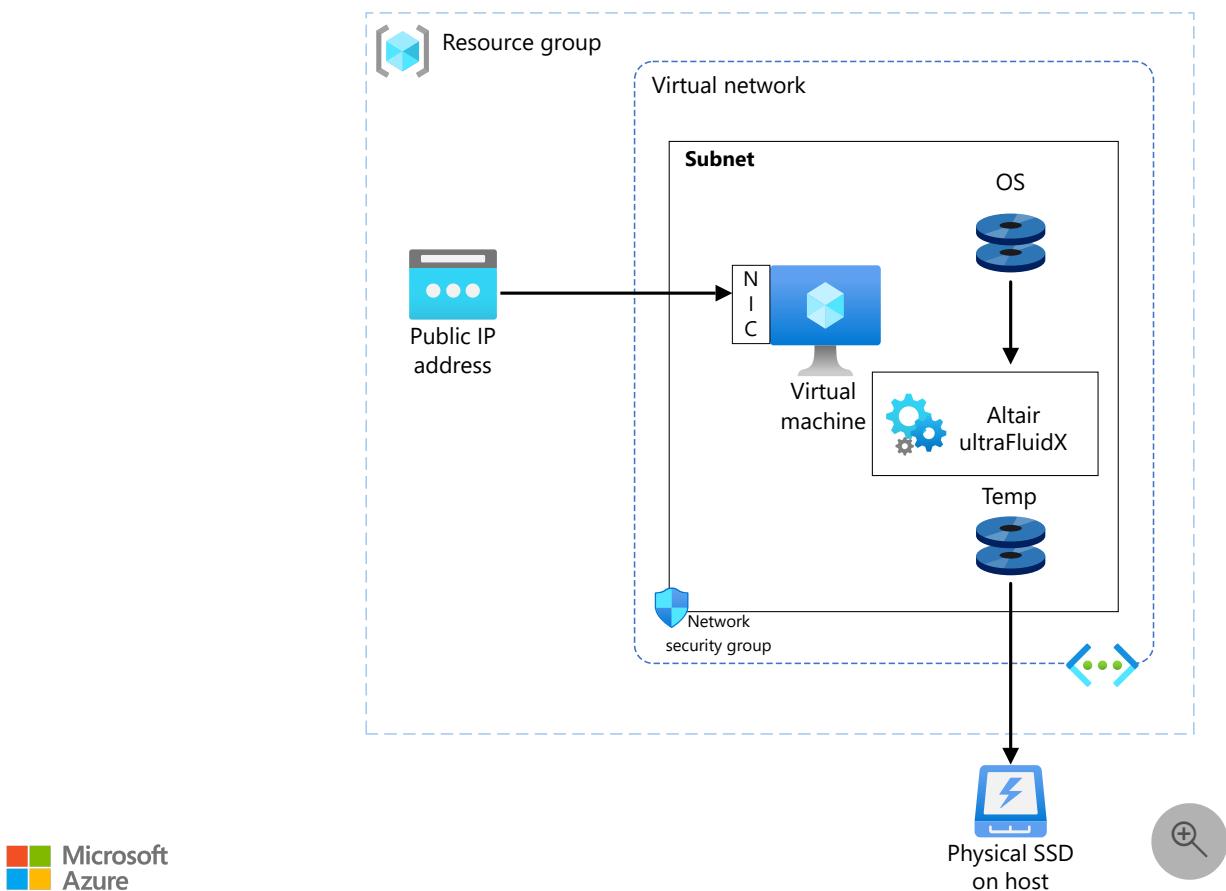
- Is based on Lattice Boltzmann methods (LBM).
- Is optimized for GPUs and supports CUDA-aware MPI for multi-GPU usage.
- Provides an LBM-consistent Smagorinsky LES turbulence model, TBLE-based wall modeling, and porous media model (pressure drop) for simulation of multiple heat exchangers.
- Handles rotating geometries via wall-velocity boundary conditions, a Moving Reference Frame (MRF) model, and truly rotating overset grids (OSM).
- Provides automated volume mesh generation with low surface mesh requirements, local grid refinement, and support for intersecting/baffle parts.

Altair ultraFluidX is used in the automotive, building, facilities, energy, and environmental industries.

Why deploy ultraFluidX on Azure?

- Modern and diverse compute options to align with your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Complex problems solved within a few hours

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Linux VM.
 - For information about deploying the VM and installing the drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

Performance tests of ultraFluidX on Azure used [ND A100 v4 series VMs](#) running Linux. The following table provides the configuration details.

[\[\] Expand table](#)

VM size	vCPU	Memory, in GiB	SSD, in GiB	GPUs	GPU memory, in GiB	Maximum data disks
Standard_ND96asr_v4	96	900	6,000	8 A100	40	32

The Standard_ND96asr_v4 VM runs NVIDIA Ampere A100 Tensor Core GPUs and is supported by 96 AMD processor cores.

Required drivers

To use ultraFluidX on Standard_ND96asr_v4 VMs as described in this article, you need to install NVIDIA and AMD drivers.

ultraFluidX installation

Before you install ultraFluidX, you need to deploy and connect a Linux VM and install the required NVIDIA and AMD drivers.

i **Important**

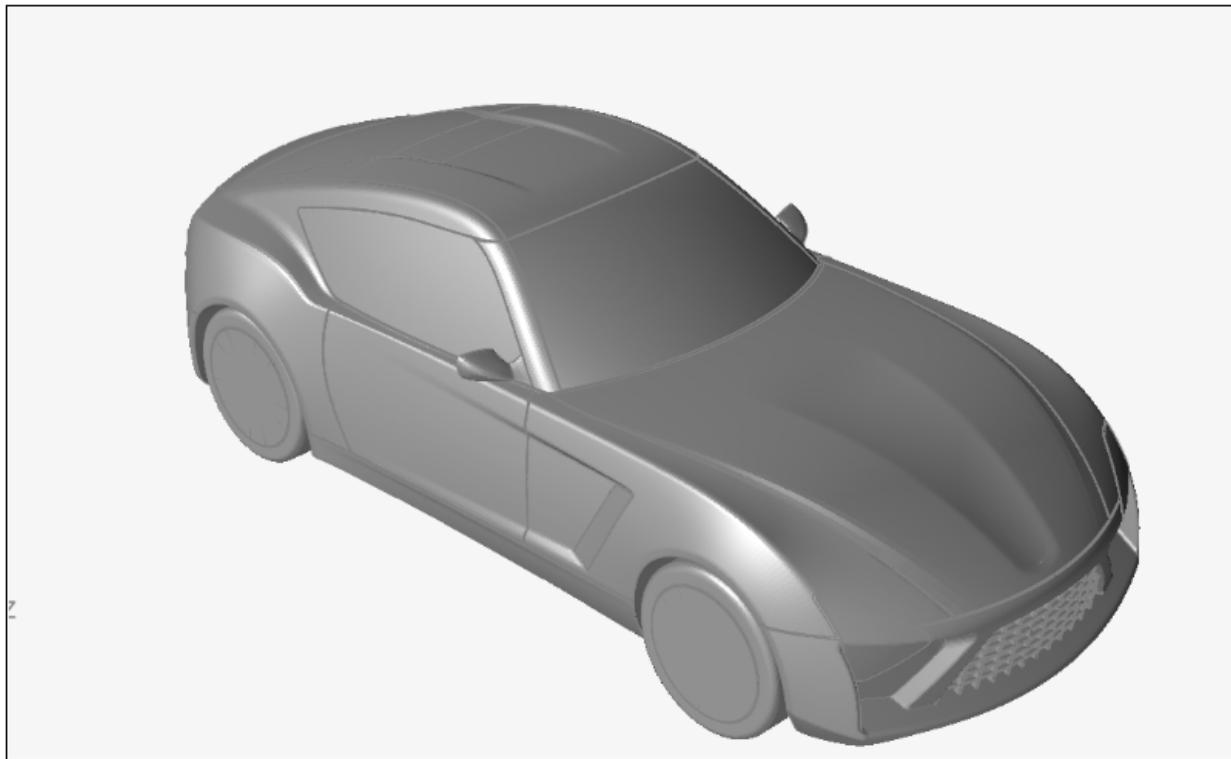
NVIDIA Fabric Manager installation is required for VMs that use NVLink or NVSwitch. Standard_ND96asr_v4 uses NVLink.

For information about deploying the VM and installing the drivers, see [Run a Linux VM on Azure](#).

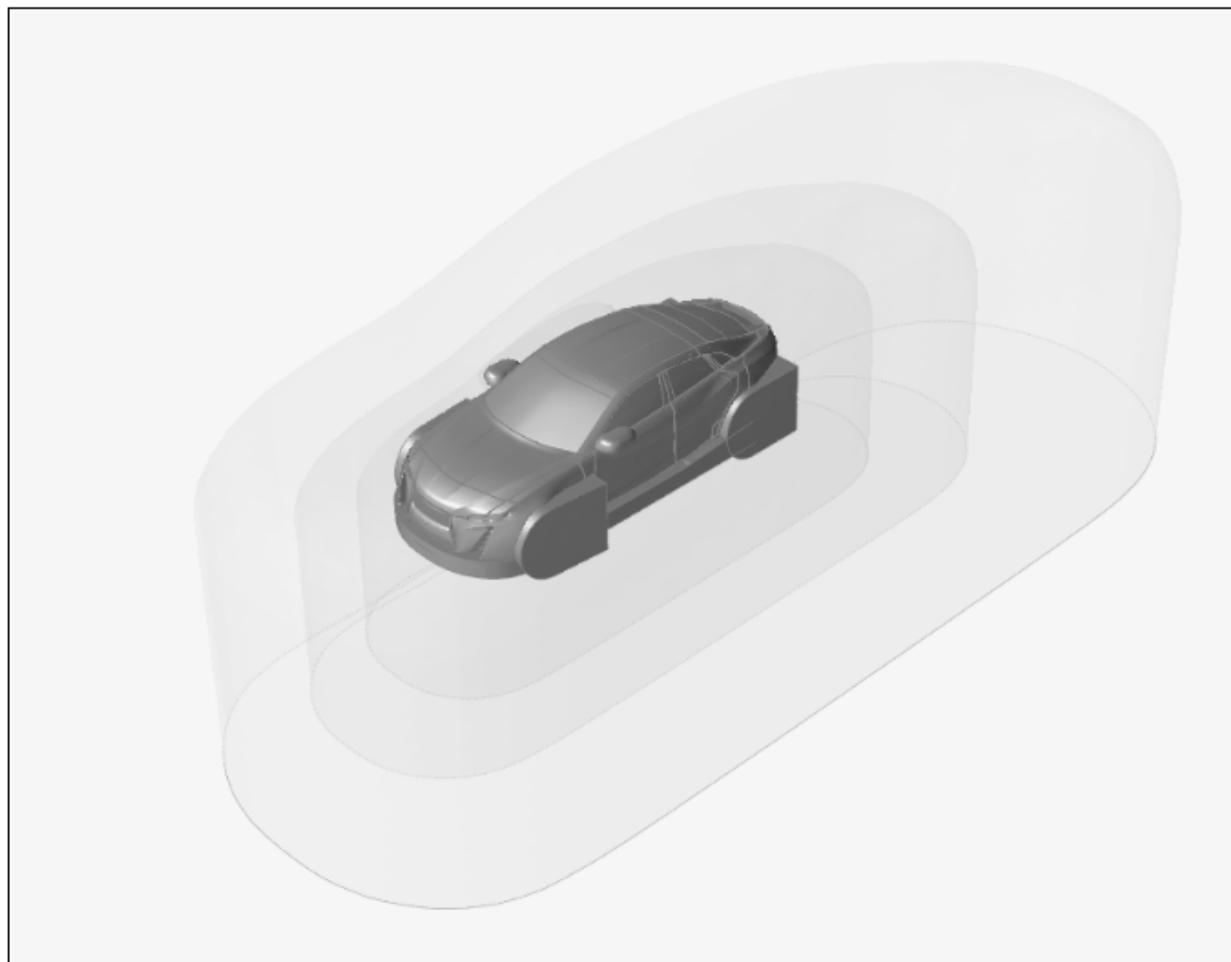
Altair ultraFluidX only runs on Linux. You can download ultraFluidX from [Altair One Marketplace](#). You also need to install Altair License Manager and activate your license via Altair Units Licensing. For more information, see the Altair Units Licensing document on [Altair One Marketplace](#).

ultraFluidX performance results

The Roadster and CX1 models were used as test cases. This image shows the roadster model:



This image shows the CX1 model:



The amount of time it takes to complete the simulation by using GPUs was measured. The Linux platform was used, with an Azure Marketplace CentOS 8.1 HPC Gen2 image. The

following table provides details about the operating system and NVIDIA drivers.

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

Operating system version	OS architecture	GPU driver version	Cuda version
CentOS Linux release 8.1.1911 (Core)	x86-64	470.57.02	11.4

GPU-based fluid dynamics simulations were run to test ultraFluidX. The simulations were run for shortened test cases, not for full production-level test cases. The projected wall-clock times and computation times for a full production run of the CX1 are provided here. Because the workload per time step is constant, these times can be computed from the computation time of the short run via linear extrapolation.

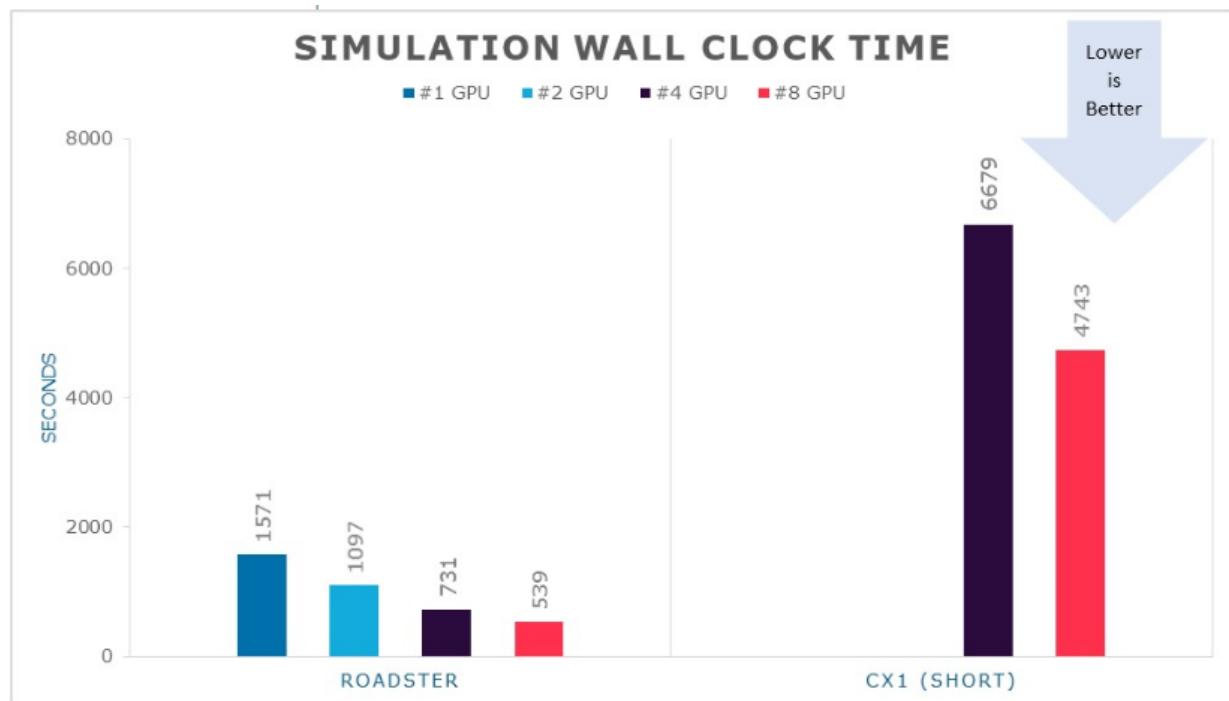
The total simulation consists of two phases: a mostly CPU-based pre-processing phase (independent of the physical simulation time) and the GPU-based computation phase. The purpose of the simulation is to test the performance of the GPU phase on the chosen VM: Standard_ND96asr_v4.

The following table shows the wall-clock times, in seconds.

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

Model	1 GPU	2 GPUs	4 GPUs	8 GPUs
Roadster	1,571	1,097	731	539
CX1 (short run)	NA*	NA*	6,679	4,743
CX1 (production run)	NA*	NA*	39,115	23,518

This graph provides the same information for the Roadster model and the short run of the CX1 model:



The following table shows the pre-processing times, in seconds.

[Expand table](#)

Model	1 GPU	2 GPUs	4 GPUs	8 GPUs
Roadster	679	607	446	350
CX1	NA*	NA*	4,926	3,728

The following table shows the computation times, in seconds.

[Expand table](#)

Model	1 GPU	2 GPUs	4 GPUs	8 GPUs
Roadster	782	433	257	174
CX1 (short run)	NA*	NA*	1,560	903
CX1 (production run)	NA*	NA*	33,996	19,678

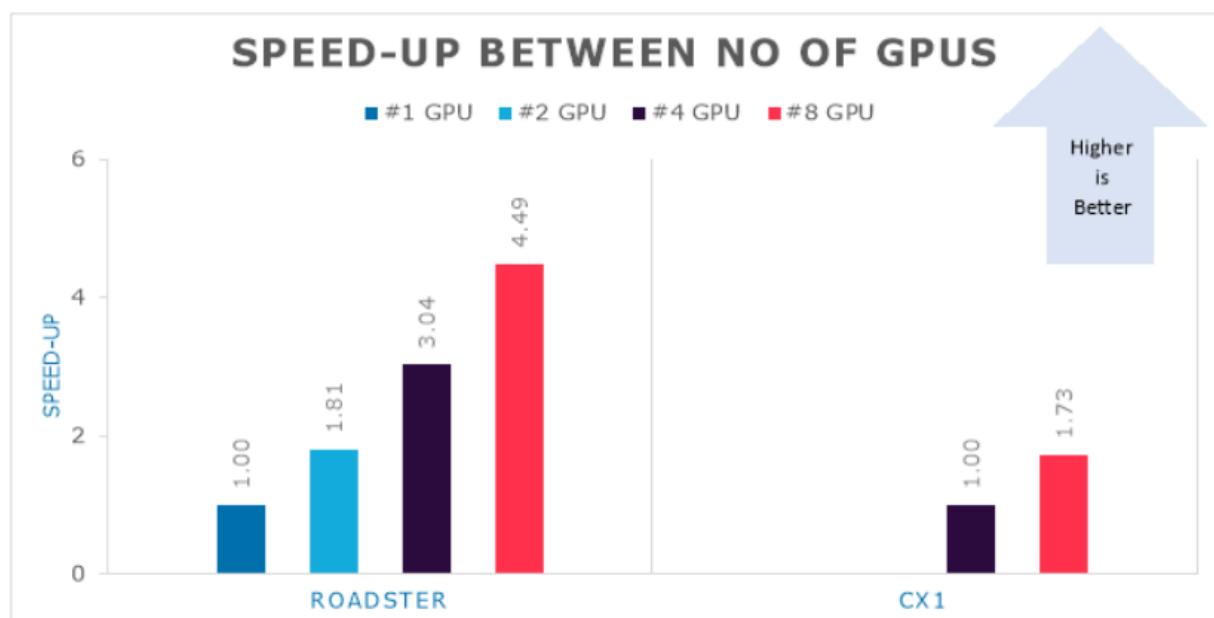
Finally, the following table shows the relative speed increases when the number of GPUs is increased. The speed increases are calculated for the computation time (the phase when GPUs are used) to provide the GPU performance.

[Expand table](#)

Model	1 GPU	2 GPUs	4 GPUs	8 GPUs
Roadster	1.00	1.81	3.04	4.49
CX1	NA*	NA*	1.00	1.73

* NA indicates that the model requires more than 100 GB of GPU memory, so the simulation can't run with only one or two GPUs.

Here's that information in graphical form:



Azure cost

The following table presents wall-clock times that you can use to calculate Azure costs. You can use the times presented here together with the Azure hourly rates for ND A100 v4-series VMs to calculate costs. For the current hourly costs, see [Linux Virtual Machines Pricing](#).

Only wall-clock time is considered for these cost calculations. Application installation time isn't considered.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

[Expand table](#)

Model	Number of GPUs*	Wall-clock time, in seconds
Roadster	1	1,571
Roadster	2	1,097
Roadster	4	731
Roadster	8	539
CX1 (short run)	4	6,679
CX1 (short run)	8	4,743
CX1 (production run)	4	39,115
CX1 (production run)	8	23,518

* The CX1 model requires more than 100 GB of GPU memory, so the simulation can't run with only one or two GPUs.

Summary

- Altair ultraFluidX was successfully tested on ND A100 v4-series VMs on Azure.
- Complex problems can be solved within a few hours on ND A100 v4 VMs.
- Increasing the number of GPUs improves performance.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy

- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- GPU-optimized virtual machine sizes
- Linux virtual machines on Azure
- Virtual networks and virtual machines on Azure
- Learning path: Run high-performance computing (HPC) applications on Azure

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Ansys CFX on a virtual machine

Azure Virtual Machines Azure Virtual Network

This article briefly describes the steps for running [Ansys CFX](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running Ansys CFX on Azure.

Ansys CFX is computational fluid dynamics (CFD) software for turbomachinery applications. It uses an equilibrium phase change model and relies on material properties to reliably predict cavitation without the need for empirical model parameters. CFX:

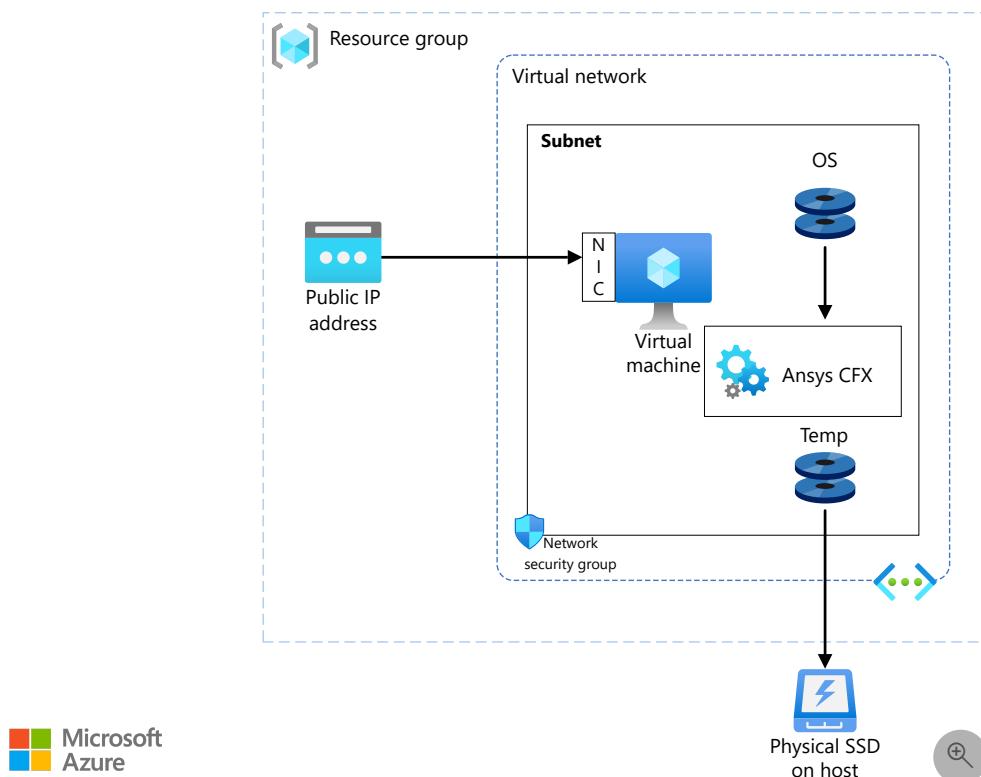
- Uses transient blade row methods to reduce geometry from a full wheel to a single passage.
- Integrates with Geolus Shape Search to rapidly find parts that are identical to a specified part, based on geometry.

CFX is used in the aerospace, defense, steam turbine, energy, automotive, construction, facilities, manufacturing, and materials/chemical processing industries.

Why deploy Ansys CFX on Azure?

- Modern and diverse compute options to align to your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Multi-node deployment as much as 17 times faster than single-node deployment

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Linux VM.
 - For information about deploying the VM and installing the drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

Performance tests of Ansys CFX on Azure used [HBv3-series](#) VMs running Linux. The following table provides the configuration details.

[Expand table](#)

VM size	vCPU	Memory (GiB)	Memory bandwidth (Gbps)	Base CPU frequency (Ghz)	All-cores frequency (Ghz, peak)	Single-core frequency (Ghz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120rs_v3	120	448	350	2.45	3.1	3.675	200	32
Standard_HB120-96rs_v3	96	448	350	2.45	3.1	3.675	200	32
Standard_HB120-64rs_v3	64	448	350	2.45	3.1	3.675	200	32
Standard_HB120-32rs_v3	32	448	350	2.45	3.1	3.675	200	32
Standard_HB120-16rs_v3	16	448	350	2.45	3.1	3.675	200	32

Required drivers

To use AMD CPUs on [HBv3 VMs](#), you need to install AMD drivers.

To use InfiniBand, you need to enable InfiniBand drivers.

CFX installation

Before you install CFX, you need to deploy and connect a Linux VM and install the required AMD and InfiniBand drivers.

For information about deploying the VM, see [Run a Linux VM on Azure](#).

For information about installing CFX, see the [Ansys website](#).

CFX performance results

CFD analysis was performed in these tests. Ansys CFX 2021 R2 was tested. The following table provides the details of the VM that was used for testing.

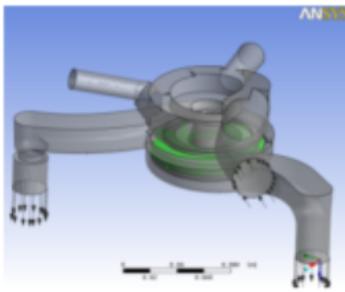
[Expand table](#)

Operating system	OS architecture	MPI
CentOS Linux release 8.1.1911 (Core)	Linux x86-64	Intel MPI

Many factors can influence HPC scalability, including the mesh size, element type, mesh topology, and physical models. To get meaningful and case-specific benchmark results, it's best to use the standard HPC benchmark cases that are available on the [Ansys Customer Portal](#).

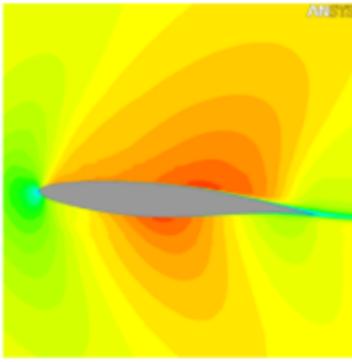
The following models were tested. For more information about the current Ansys models, see [CFX benchmarks](#).

Pump



This model represents an automotive pump with rotating and stationary components.

Airfoils



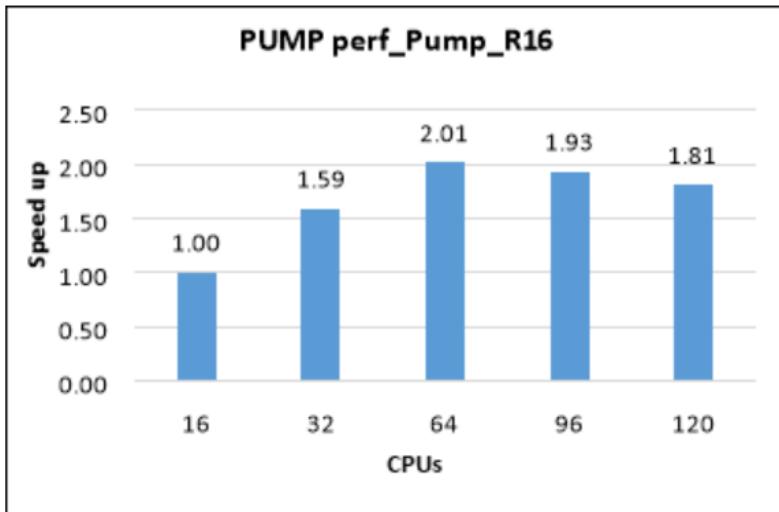
These models represent transonic flow around an airfoil. Airfoils with mesh sizes of 10 million, 50 million, and 100 million were tested.

Results, single-node configuration

The following table and graph show elapsed wall-clock times and relative speed increases for the pump model.

[Expand table](#)

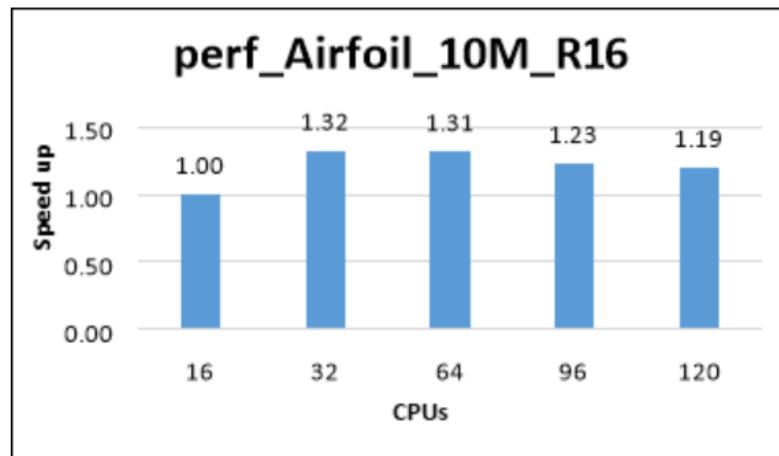
Model	Iterations	Cores	CFD solver wall-clock time (seconds)	Relative speed increase
perf_Pump_R16	10	16	32.59	1.00
perf_Pump_R16	10	32	20.48	1.59
perf_Pump_R16	10	64	16.19	2.01
perf_Pump_R16	10	96	16.85	1.93
perf_Pump_R16	10	120	18.00	1.81



The following table and graph show elapsed wall-clock times and relative speed increases for the airfoil model, with a mesh size of 10 million.

[Expand table](#)

Model	Iterations	Cores	CFD solver wall-clock time (seconds)	Relative speed increase
perf_Airfoil_10M_R16	5	16	149.40	1.00
perf_Airfoil_10M_R16	5	32	113.05	1.32
perf_Airfoil_10M_R16	5	64	113.87	1.31
perf_Airfoil_10M_R16	5	96	121.71	1.23
perf_Airfoil_10M_R16	5	120	125.10	1.19

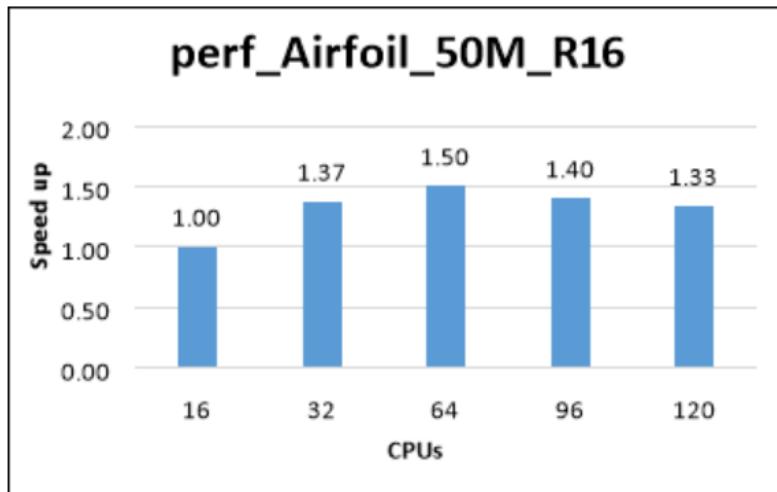


The following table and graph show elapsed wall-clock times and relative speed increases for the airfoil model, with a mesh size of 50 million.

[Expand table](#)

Model	Iterations	Cores	CFD solver wall-clock time (seconds)	Relative speed increase
perf_Airfoil_50M_R16	5	16	861.34	1.00

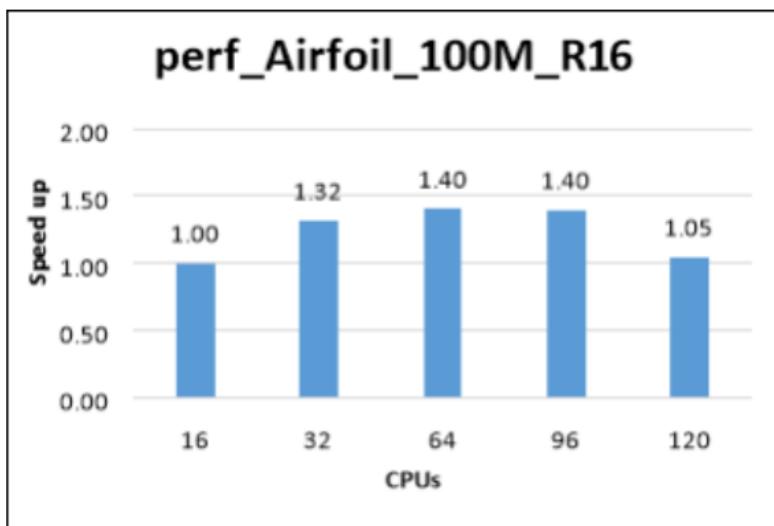
Model	Iterations	Cores	CFD solver wall-clock time (seconds)	Relative speed increase
perf_Airfoil_50M_R16	5	32	627.99	1.37
perf_Airfoil_50M_R16	5	64	573.76	1.50
perf_Airfoil_50M_R16	5	96	616.32	1.40
perf_Airfoil_50M_R16	5	120	646.07	1.33



The following table and graph show elapsed wall-clock times and relative speed increases for the airfoil model, with a mesh size of 100 million.

[Expand table](#)

Model	Iterations	Cores	CFD solver wall-clock time (seconds)	Relative speed increase
perf_Airfoil_100M_R16	5	16	2029.20	1.00
perf_Airfoil_100M_R16	5	32	1541.70	1.32
perf_Airfoil_100M_R16	5	64	1445.70	1.40
perf_Airfoil_100M_R16	5	96	1451.70	1.40
perf_Airfoil_100M_R16	5	120	1473.70	1.05



Results, multi-node configuration

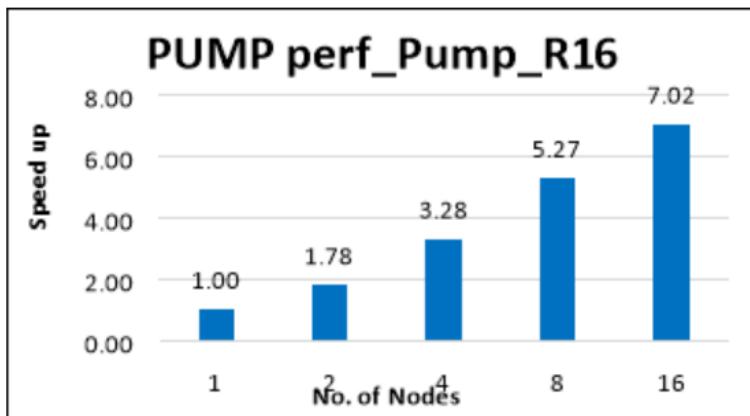
As the single-node results show, scalability improves as the number of cores increases. Because memory bandwidth is fixed on a single node, performance saturates after a certain number of cores is reached. A multi-node configuration surpasses this limitation to fully take advantage of the CFX solver capabilities.

Based on the single-node tests, the 64-CPU configuration is optimal. It's also less expensive than 96-CPU and 120-CPU configurations. The Standard_HB120-64rs_v3 VM with 64 CPUs was used for the multi-node tests.

The following table and graph show elapsed wall-clock times and relative speed increases for the pump model.

[Expand table](#)

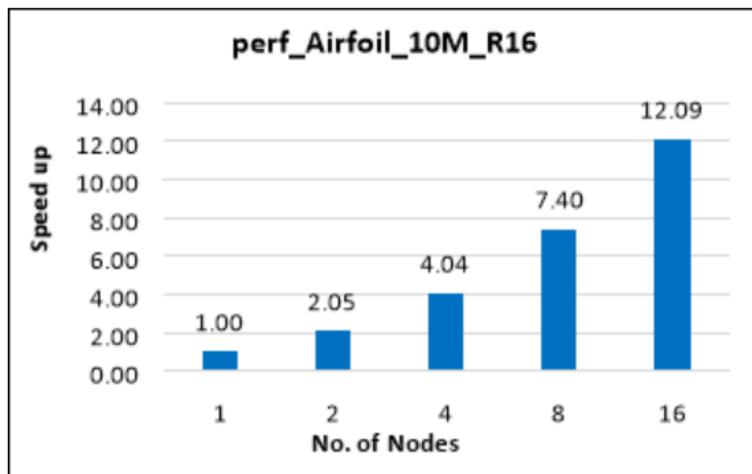
Model	Iterations	Number of nodes	Number of cores	CFD solver wall-clock time (seconds)	Relative speed increase
perf_Pump_R16	10	1	64	16.19	1.00
perf_Pump_R16	10	2	128	9.09	1.78
perf_Pump_R16	10	4	256	4.93	3.28
perf_Pump_R16	10	8	512	3.07	5.27
perf_Pump_R16	10	16	1,024	2.30	7.02



The following table and graph show elapsed wall-clock times and relative speed increases for the airfoil model, with a mesh size of 10 million.

[Expand table](#)

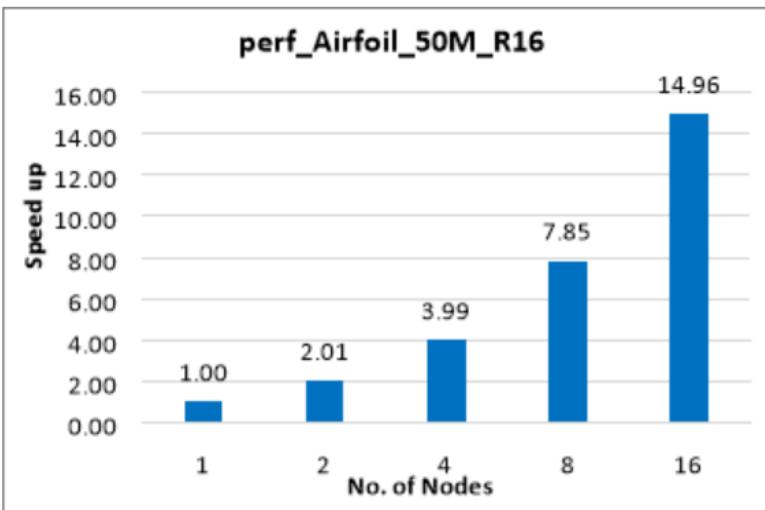
Model	Iterations	Number of nodes	Number of cores	CFD solver wall-clock time (seconds)	Relative speed increase
perf_Airfoil_10M_R16	10	1	64	113.87	1.00
perf_Airfoil_10M_R16	10	2	128	55.43	2.05
perf_Airfoil_10M_R16	10	4	256	28.21	4.04
perf_Airfoil_10M_R16	10	8	512	15.39	7.40
perf_Airfoil_10M_R16	10	16	1,024	9.42	12.09



The following table and graph show elapsed wall-clock times and relative speed increases for the airfoil model, with a mesh size of 50 million.

[Expand table](#)

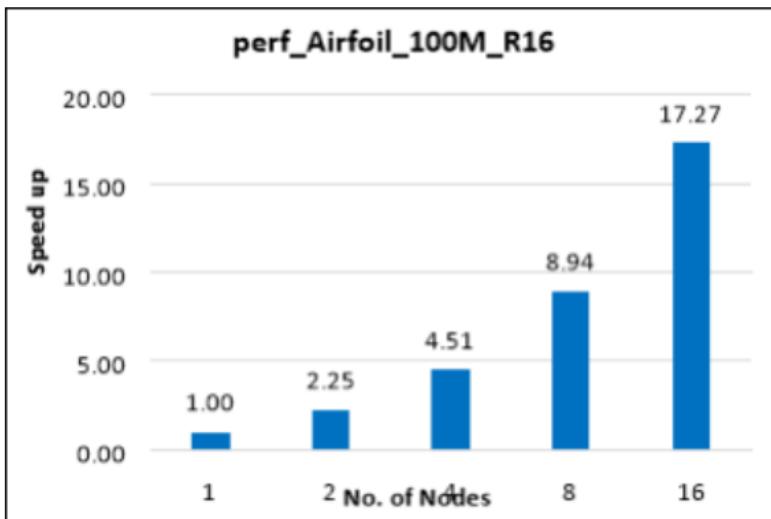
Model	Iterations	Number of nodes	Number of cores	CFD solver wall-clock time (seconds)	Relative speed increase
perf_Airfoil_50M_R16	10	1	64	573.76	1.00
perf_Airfoil_50M_R16	10	2	128	284.75	2.01
perf_Airfoil_50M_R16	10	4	256	143.73	3.99
perf_Airfoil_50M_R16	10	8	512	73.09	7.85
perf_Airfoil_50M_R16	10	16	1,024	38.35	14.96



The following table and graph show elapsed wall-clock times and relative speed increases for the airfoil model, with a mesh size of 100 million.

[Expand table](#)

Model	Iterations	Number of nodes	Number of cores	CFD solver wall-clock time (seconds)	Relative speed increase
perf_Airfoil_100M_R16	10	1	64	1445.70	1.00
perf_Airfoil_100M_R16	10	2	128	642.95	2.25
perf_Airfoil_100M_R16	10	4	256	320.27	4.51
perf_Airfoil_100M_R16	10	8	512	161.64	8.94
perf_Airfoil_100M_R16	10	16	1,024	83.73	17.27



Azure cost

The following tables present wall-clock times that you can use to calculate Azure costs. You can multiply the times presented here by the Azure hourly rates for HBv3-series VMs to calculate costs. For the current hourly costs, see [Linux Virtual Machines Pricing](#).

Only the wall-clock time per 100 iterations for running each model is considered for these cost calculations. Application installation time and license costs aren't considered.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

Costs, single-node configuration

 [Expand table](#)

VM size	Number of CPUs	CFD solver wall-clock time (hours)
Standard_HB120-16rs_v3	16	0.85
Standard_HB120-32rs_v3	32	0.64
Standard_HB120-64rs_v3	64	0.60
Standard_HB120-96rs_v3	96	0.61
Standard_HB120rs_v3	120	0.63

Costs, multi-node configuration

 [Expand table](#)

VM size	Number of nodes	Number of cores	CFD solver wall-clock time (seconds)	Hours
HB120-64rs_v3	1	64	2,175	0.60
HB120-64rs_v3	2	128	1,005	0.28
HB120-64rs_v3	4	256	504	0.14
HB120-64rs_v3	8	512	257	0.07
HB120-64rs_v3	16	1,024	134	0.04

Summary

- Ansys CFX was successfully tested on HBv3-series VMs on Azure.
- In single-node configurations, performance scales well up to 64 cores. After that point, the speed increase drops off.
- In multi-node configurations, performance scales linearly when nodes are added.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Ansys Fluent on a virtual machine

Azure Virtual Machines Azure Virtual Network

This article briefly describes the steps for running [Ansys Fluent](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running Ansys Fluent on Azure.

Fluent is a computational fluid dynamics (CFD) application that's used to model fluid flow, heat and mass transfer, chemical reactions, and more. Fluent provides:

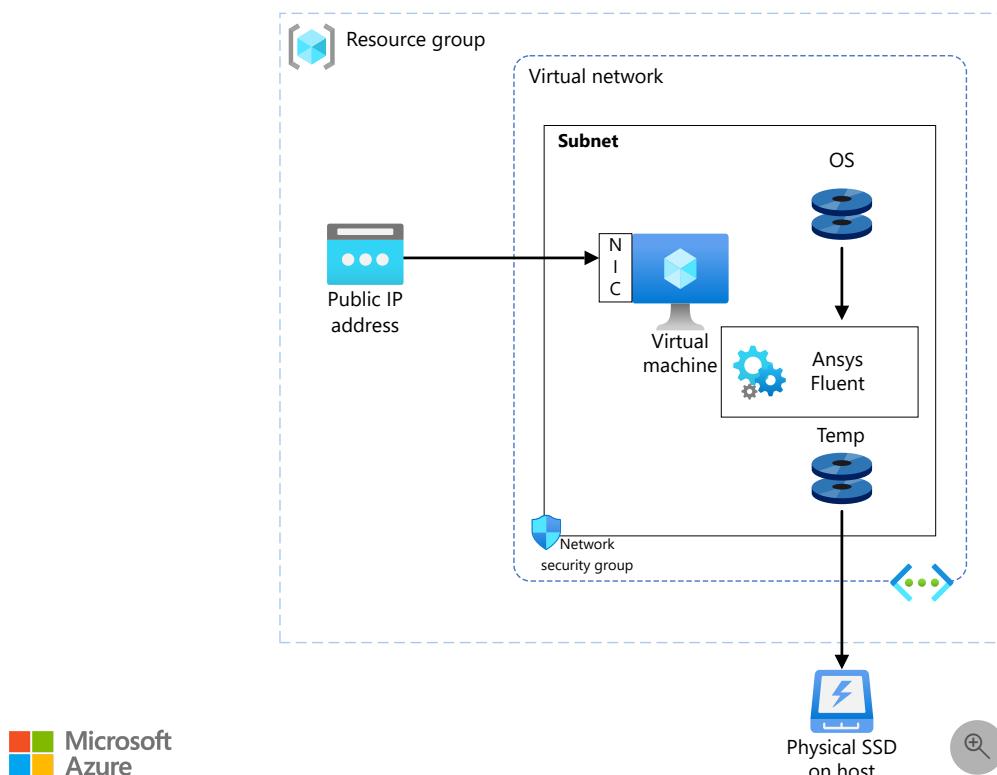
- Task-based workflows, including multiphase modeling, batteries, shape optimization, and aerodynamics, for pre-processing the generation of a CFD-ready mesh for both clean and dirty CAD.
- Accurate simulation of multiphase flows, including gas-liquid, liquid-liquid, gas-solid, particle flows, and DEM.
- A range of turbulence models, including the GEKO model.

Fluent is used in the aerospace, automotive, medical, healthcare, manufacturing, industrial equipment, communication, embedded systems, energy, retail, and consumer goods industries.

Why deploy Ansys Fluent on Azure?

- Modern and diverse compute options to align to your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Performance that scales well up to 64 or 96 CPUs on a single node and linearly on multiple nodes

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Linux VM.
 - For information about deploying the VM and installing the drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.

- A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

Performance tests of Fluent on Azure used [HBv3-series](#) VMs running the Linux CentOS operating system. The following table provides details about HBv3-series VMs.

[Expand table](#)

VM size	vCPU	Memory (GiB)	Memory bandwidth Gbps	Base CPU frequency (Ghz)	All-cores frequency (Ghz, peak)	Single-core frequency (Ghz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120rs_v3	120	448	350	1.9	3.0	3.5	200	32
Standard_HB120-96rs_v3	96	448	350	1.9	3.0	3.5	200	32
Standard_HB120-64rs_v3	64	448	350	1.9	3.0	3.5	200	32
Standard_HB120-32rs_v3	32	448	350	1.9	3.0	3.5	200	32
Standard_HB120-16rs_v3	16	448	350	1.9	3.0	3.5	200	32

Required drivers

To use AMD CPUs on [HBv3 VMs](#), you need to install AMD drivers.

To use InfiniBand, you need to enable InfiniBand drivers.

Fluent installation

Before you install Fluent, you need to deploy and connect a VM, install Linux, and install the required AMD and InfiniBand drivers.

For information about deploying the VM and installing the drivers, see [Run a Linux VM on Azure](#).

For information about installing Fluent, see the [Ansys website](#).

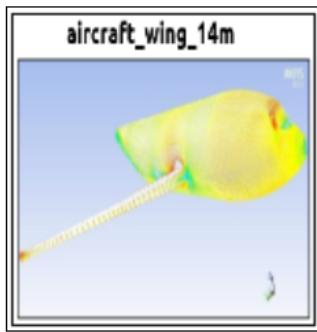
Fluent performance results

HBv3 VMs with different numbers of vCPUs were deployed to determine the optimal configuration for Fluent on a single node. That optimal configuration was then tested in a multi-node cluster deployment. Ansys Fluent 2021 R2 was tested.

Results, single-node configuration

The single-node configuration was evaluated for the following test cases.

Aircraft wing test case



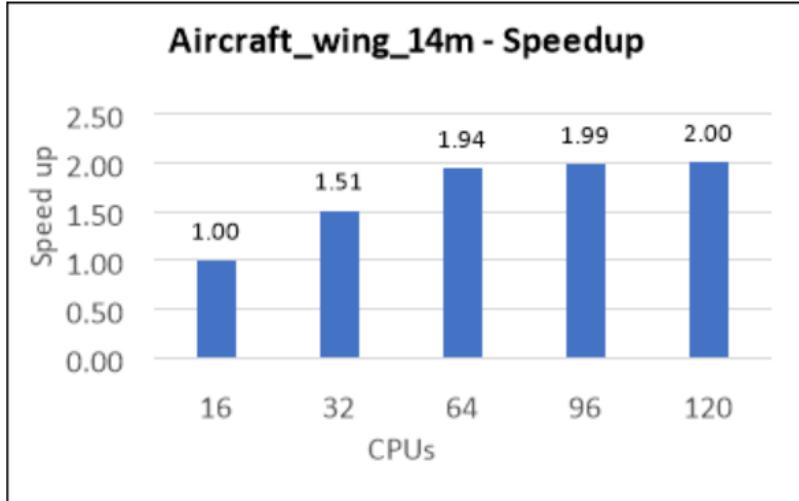
[Expand table](#)

Test case name	Number of cells	Cell type	Solver	Models
aircraft_wing_14m	14,000,000	Hexahedral	Pressure-based coupled solver, Least Squares cell based, steady	Realizable K-e Turbulence

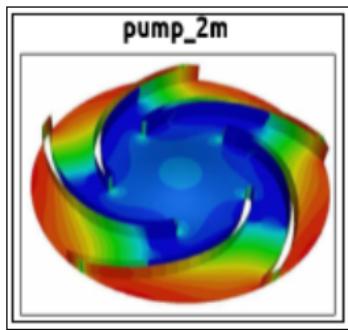
The following table and graph present the test results.

[Expand table](#)

Cores	Wall-time per 100 iterations (seconds)	Relative speed increase
16	860.67	1.00
32	569.03	1.51
64	442.69	1.94
96	433.45	1.99
120	429.54	2.00



Pump test case



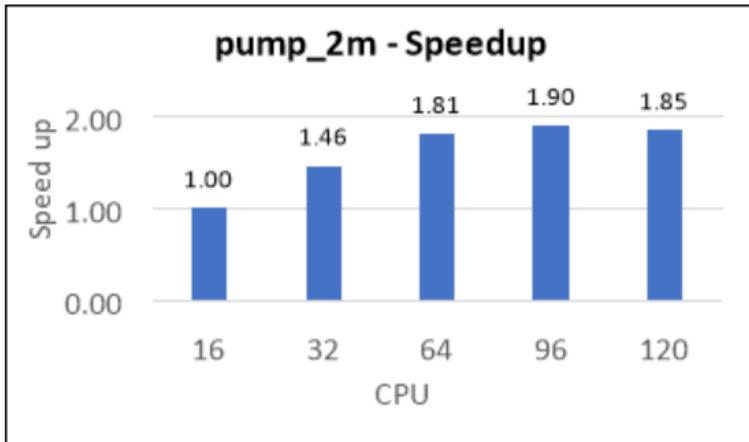
[Expand table](#)

Test case name	Number of cells	Cell type	Solver	Models
pump_2m	2,000,000	Hexahedral	Pressure-based coupled solver, Least Squares cell based, steady	Realizable K-e Turbulence, Mixture Multiphase

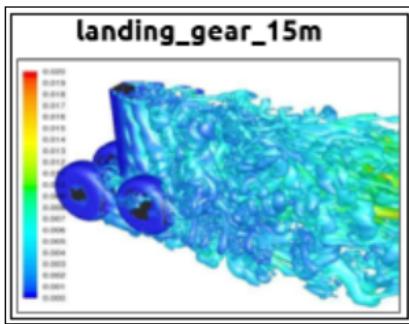
The following table and graph present the test results.

[Expand table](#)

Cores	Wall-time per 100 iterations (seconds)	Relative speed increase
16	213.83	1.00
32	146.38	1.46
64	118.26	1.81
96	112.53	1.90
120	115.47	1.85



Landing gear test case



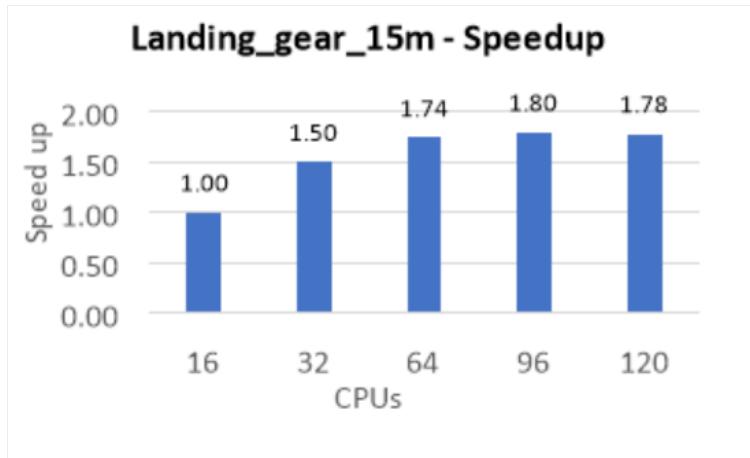
[Expand table](#)

Test case name	Number of cells	Cell type	Solver	Models
landing_gear_15m	15,000,000	Mixed	Pressure-based coupled solver, Least Squares cell based, Unsteady	LES, Acoustics

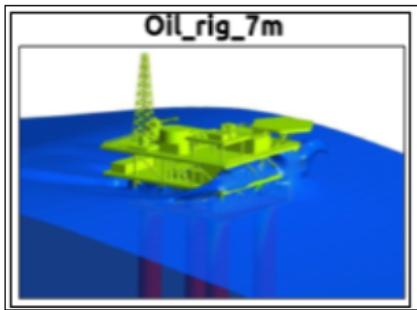
The following table and graph present the test results.

[Expand table](#)

Cores	Wall-time per 100 iterations (seconds)	Relative speed increase
16	871.37	1.00
32	580.31	1.50
64	501.02	1.74
96	484.46	1.80
120	489.96	1.78



Oil rig test case



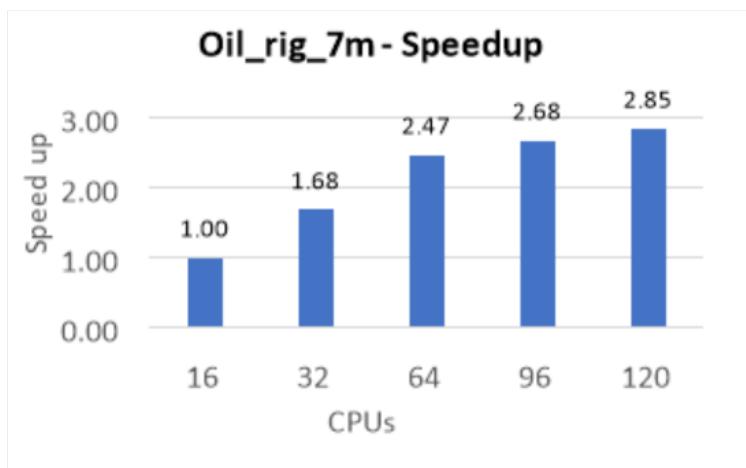
[Expand table](#)

Test case name	Number of cells	Cell type	Solver	Models
oil_rig_7m	7,000,000	Mixed	Pressure-based segregated solver, Green-Gauss cell based, unsteady	VOF, SST K-omega Turbulence

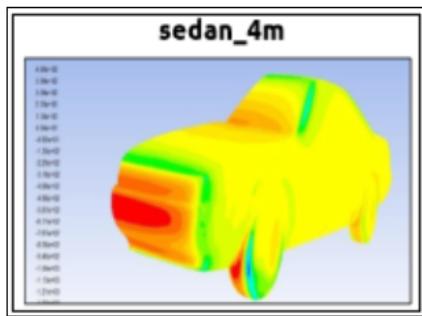
The following table and graph present the test results.

[Expand table](#)

Cores	Wall-time per 100 iterations (seconds)	Relative speed increase
16	377.11	1.00
32	224.16	1.68
64	152.42	2.47
96	140.81	2.68
120	132.34	2.85



Sedan test case



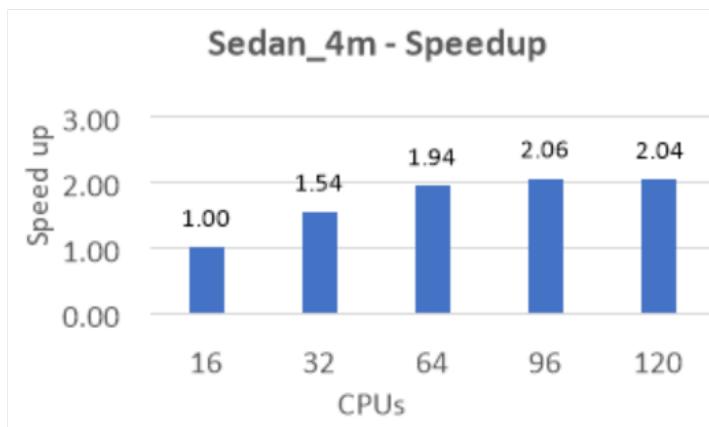
expand [Expand table](#)

Test case name	Number of cells	Cell type	Solver	Models
sedan_4m	4,000,000	Mixed	Pressure-based coupled solver, Green-Gauss cell based, steady	Standard K-e Turbulence

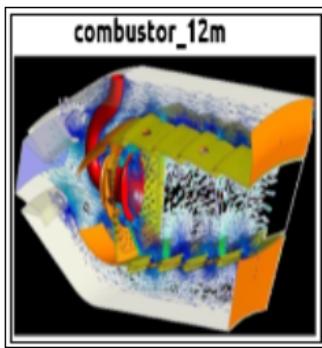
The following table and graph present the test results.

expand [Expand table](#)

Cores	Wall-time per 100 iterations (seconds)	Relative speed increase
16	154.02	1.00
32	99.88	1.54
64	79.40	1.94
96	74.88	2.06
120	75.62	2.04



Combustor test case



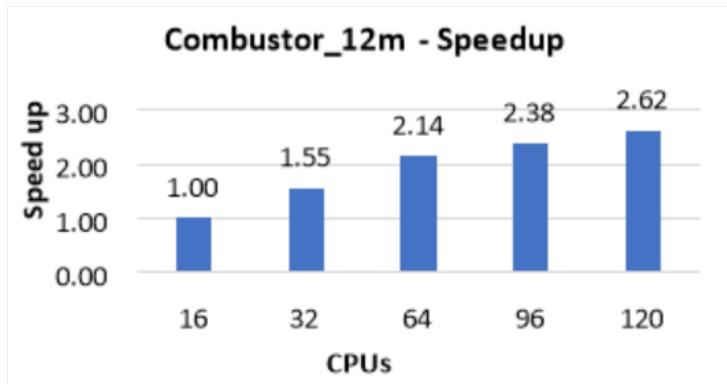
[Expand table](#)

Test case name	Number of cells	Cell type	Solver	Models
combustor_12m	12,000,000	Polyhedra	Pressure-based coupled solver, Least Squares cell based, pseudo transient	Realizable K-e Turbulence, Species Transport

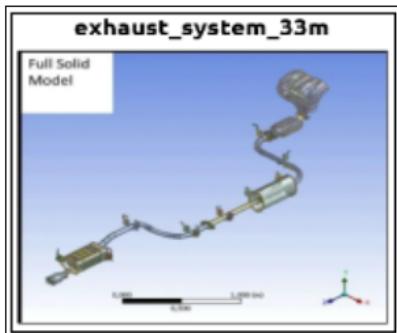
The following table and graph present the test results.

[Expand table](#)

Cores	Wall-time per 100 iterations (seconds)	Relative speed increase
16	3,238	1.00
32	2,085	1.55
64	1,513	2.14
96	1,360	2.38
120	1,236	2.62



Exhaust system test case



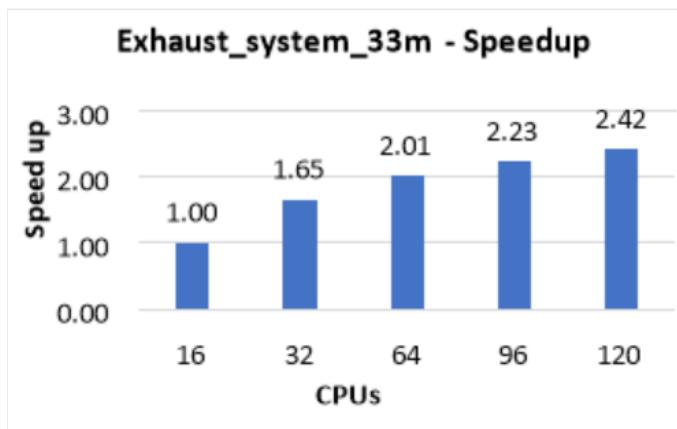
[expand] [Expand table](#)

Test case name	Number of cells	Cell type	Solver	Models
exhaust_system_33m	33,000,000	Mixed	Pressure-based coupled solver, Least Squares cell based, steady	SST K-omega Turbulence

The following table and graph present the test results.

[expand] [Expand table](#)

Cores	Wall-time per 100 iterations (seconds)	Relative speed increase
16	2,685	1.00
32	1,628	1.65
64	1,334	2.01
96	1,205	2.23
120	1,112	2.42



Results, multi-node configuration

As the preceding performance results show, HBv3-series VMs with 64 cores and 96 cores are optimal configurations. The performance improvement when you increase from 64 CPUs to 96 CPUs is between 5 and 10 percent. Taking license costs into consideration, the 64-CPU configuration is the best choice. Standard_HB120-64rs_v3 VMs, which have 64 cores, were used for the multi-node tests.

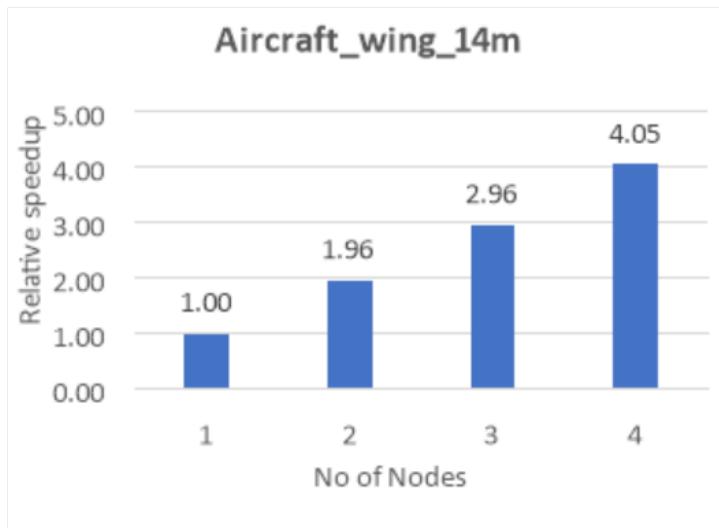
The multi-node configuration was evaluated for the same test cases.

Aircraft wing test case

[Expand table](#)

Number of nodes	Number of cores	Wall-clock time per 100 iterations (seconds)	Relative speed increase
1	64	442.69	1.00
2	128	226.06	1.96
3	192	149.31	2.96
4	256	109.23	4.05

This graph shows the relative speed increases:

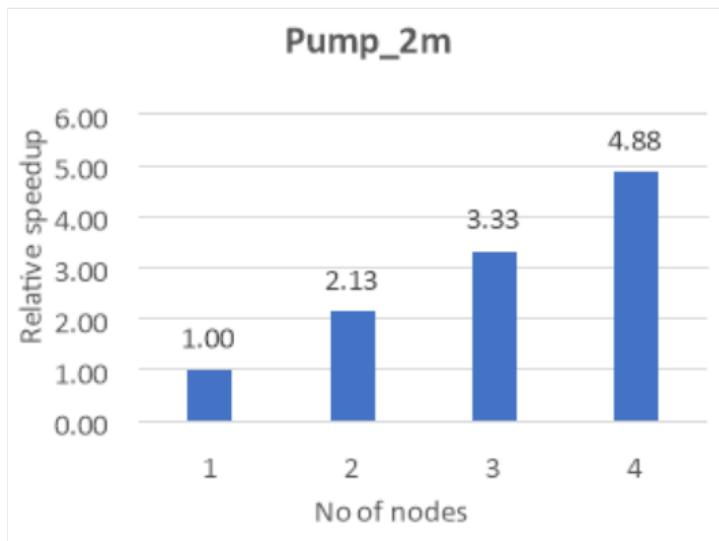


Pump test case

[Expand table](#)

Number of nodes	Number of cores	Wall-clock time per 100 iterations (seconds)	Relative speed increase
1	64	118.26	1.00
2	128	55.42	2.13
3	192	35.53	3.33
4	256	24.26	4.88

This graph shows the relative speed increases:

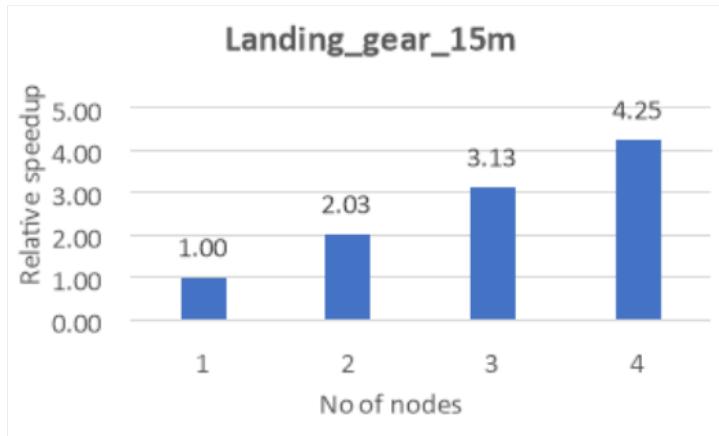


Landing gear test case

[Expand table](#)

Number of nodes	Number of cores	Wall-clock time per 100 iterations (seconds)	Relative speed increase
1	64	501.02	1.00
2	128	247.17	2.03
3	192	160.02	3.13
4	256	117.78	4.25

This graph shows the relative speed increases:



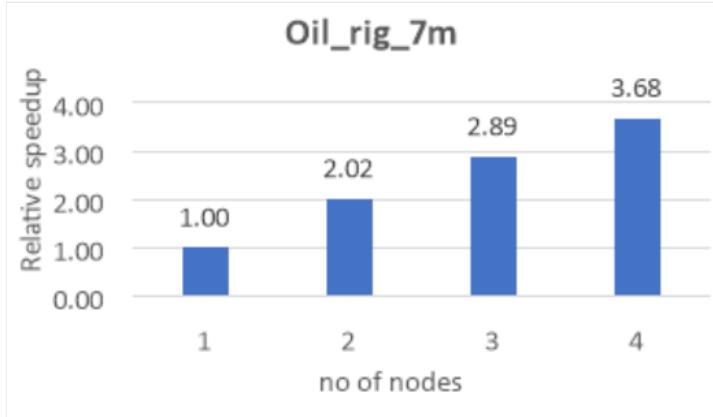
Oil rig test case

[Expand table](#)

Number of nodes	Number of cores	Wall-clock time per 100 iterations (seconds)	Relative speed increase
1	64	152.42	1.00
2	128	75.48	2.02

Number of nodes	Number of cores	Wall-clock time per 100 iterations (seconds)	Relative speed increase
3	192	52.76	2.89
4	256	41.38	3.68

This graph shows the relative speed increases:

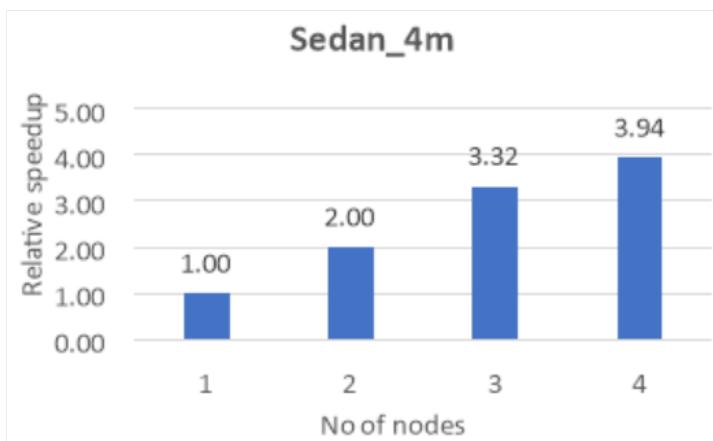


Sedan test case

[Expand table](#)

Number of nodes	Number of cores	Wall-clock time per 100 iterations (seconds)	Relative speed increase
1	64	79.40	1.00
2	128	39.66	2.00
3	192	23.90	3.32
4	256	20.15	3.94

This graph shows the relative speed increases:

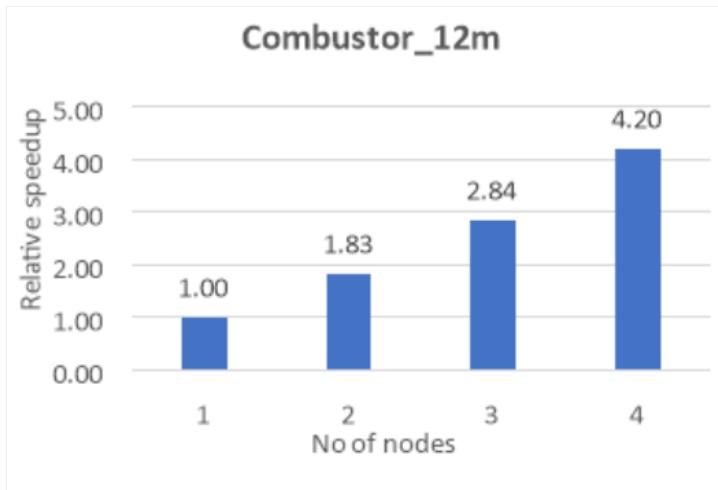


Combustor test case

[Expand table](#)

Number of nodes	Number of cores	Wall-clock time per 100 iterations (seconds)	Relative speed increase
1	64	1,512.56	1.00
2	128	828.63	1.83
3	192	531.82	2.84
4	256	359.86	4.20

This graph shows the relative speed increases:

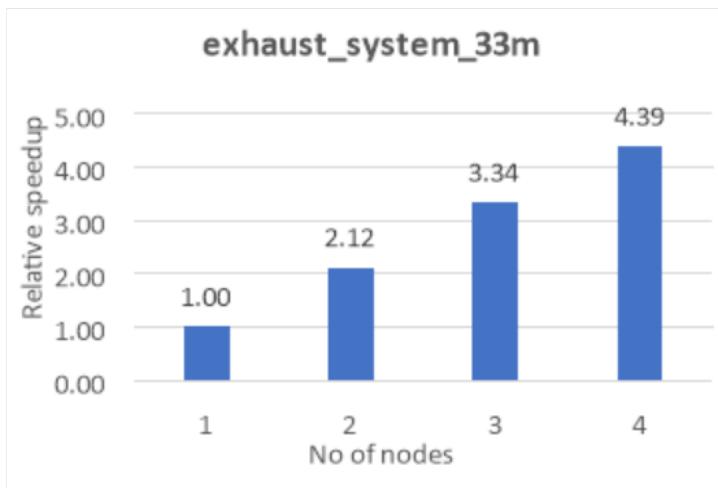


Exhaust system test case

[Expand table](#)

Number of nodes	Number of cores	Wall-clock time per 100 iterations (seconds)	Relative speed increase
1	64	1,333.72	1.00
2	128	629.02	2.12
3	192	399.66	3.34
4	256	304.05	4.39

This graph shows the relative speed increases:



Azure cost

Only wall-clock time per 100 iterations of each model is considered for these cost calculations. Application installation time and license costs aren't considered.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

For a single-node configuration, you can multiply the wall-clock times by the Azure hourly costs for HBv3-series VMs to compute total costs. For the current hourly costs, see [Linux Virtual Machines Pricing](#). Here are the times for a single-node configuration:

 [Expand table](#)

VM size	Number of CPUs	Wall-clock time (hours)
Standard_HB120-16rs_v3	16	2.33
Standard_HB120-32rs_v3	32	1.48
Standard_HB120-64rs_v3	64	1.15
Standard_HB120-96rs_v3	96	1.06
Standard_HB120rs_v3	120	1.00

For a multi-node configuration, you can multiply the wall-clock times by the number of nodes and the Azure hourly costs for HBv3-series VMs to compute total costs. For the current hourly costs, see [Linux Virtual Machines Pricing](#). Here are the times for a multi-node configuration:

 [Expand table](#)

VM size	Number of nodes	Number of cores	Wall-clock time (hours)
Standard_HB120-64rs_v3	1	64	1.15
Standard_HB120-64rs_v3	2	128	0.58
Standard_HB120-64rs_v3	3	192	0.38
Standard_HB120-64rs_v3	4	256	0.27

Summary

- Ansys Fluent 2021 R2 was successfully tested on HBv3-series Azure VMs.
- In single-node configurations, performance scaled well up to 64 or 96 CPUs. After that point, the speed increase dropped off.
- In multi-node configurations, performance scaled linearly as nodes were added.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Ansys HFSS on an Azure virtual machine

Azure Virtual Machines Azure Virtual Network Azure Bastion

This article describes the steps for installing and running [Ansys HFSS](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running HFSS.

Ansys HFSS is a 3D electromagnetic simulation application for designing and simulating high-frequency electronic products. HFSS enables engineers to address RF, microwave, IC, PCB, and EMI problems for most complex systems.

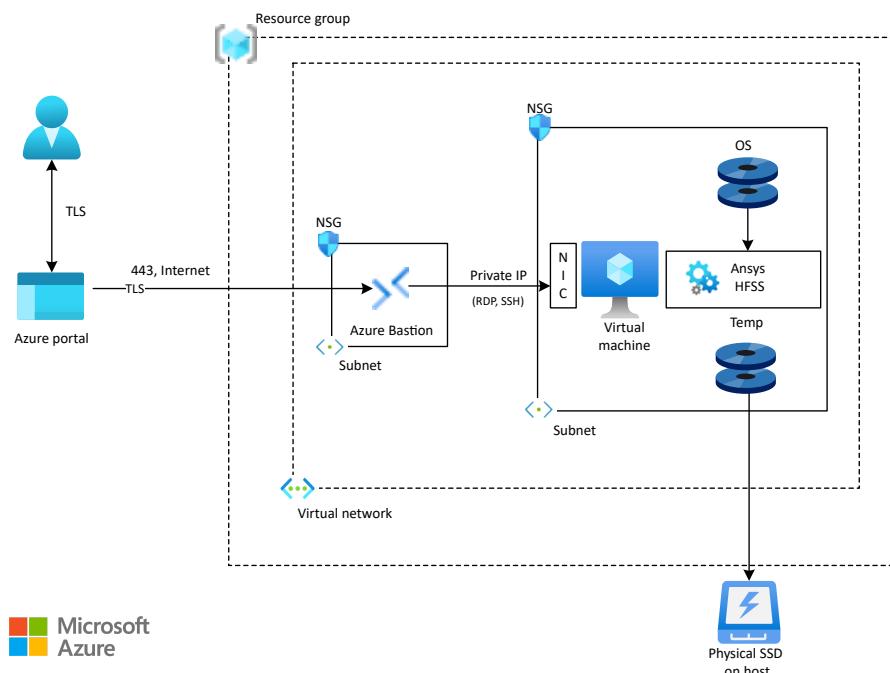
HFSS is used in simulations of high-frequency electronic products like antennas, antenna arrays, RF or microwave components, high-speed interconnects, IC packages, and printed circuit boards. Engineers use HFSS to design high-frequency, high-speed electronics found in communications systems, advanced driver assistance systems (ADAS), satellites, and IoT products.

Benefits of deploying HFSS on Azure

Azure offers:

- Modern and diverse compute options, like VM SKUs, to align to your workload requirements.
- The flexibility to create customized VMs within seconds by defining an operating system, language, and workload.
- Rapid provisioning.
- Strong GPU acceleration, with increased performance as GPUs are added.

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Windows VM.
 - For information about deploying the VM and installing the drivers, see [Windows VMs on Azure](#).
 - A physical solid-state drive (SSD) is used to store data that's related to the VM.
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud. [Network security groups](#) are used to restrict access to the VM.
- [Azure Bastion](#) provides improved-security Remote Desktop Protocol (RDP) and Secure Shell Protocol (SSH) access to VMs without any exposure through public IP addresses.

Compute sizing and drivers

NC A100 v4 series VMs were used to test the performance of HFSS on Azure.

The following table provides the configuration details:

 Expand table

Size	vCPU	Memory, in GiB	Temporary storage (with NVMe), in GiB	GPU	GPU memory, in GiB	Maximum data disks	Maximum uncached disk throughput, in IOPS / MBps	Maximum NICs / network bandwidth, in MBps
Standard_NC24ads_A100_v4	24	220	1,123	1	80	12	30,000 / 1,000	2 / 20,000
Standard_NC48ads_A100_v4	48	440	2,246	2	160	24	60,000 / 2,000	4 / 40,000
Standard_NC96ads_A100_v4	96	880	4,492	4	320	32	120,000 / 4,000	8 / 80,000

Required drivers

To take advantage of the GPU capabilities of NC A100 v4 series VMs, you need to install NVIDIA GPU drivers.

Ansys HFSS installation

Before you install HFSS, you need to deploy a VM and use the NVIDIA GPU Driver Extension provided by Azure to install NVIDIA drivers.

For information about deploying the VM and installing the drivers, see [Run a Windows VM on Azure](#).

For information about installing HFSS, see the [Ansys website](#).

Ansys HFSS performance results

The following table describes the VM that was used for testing:

 Note

These performance tests were conducted on Windows 10 Pro, 22H2. HFSS can also be deployed on newer versions of Windows.

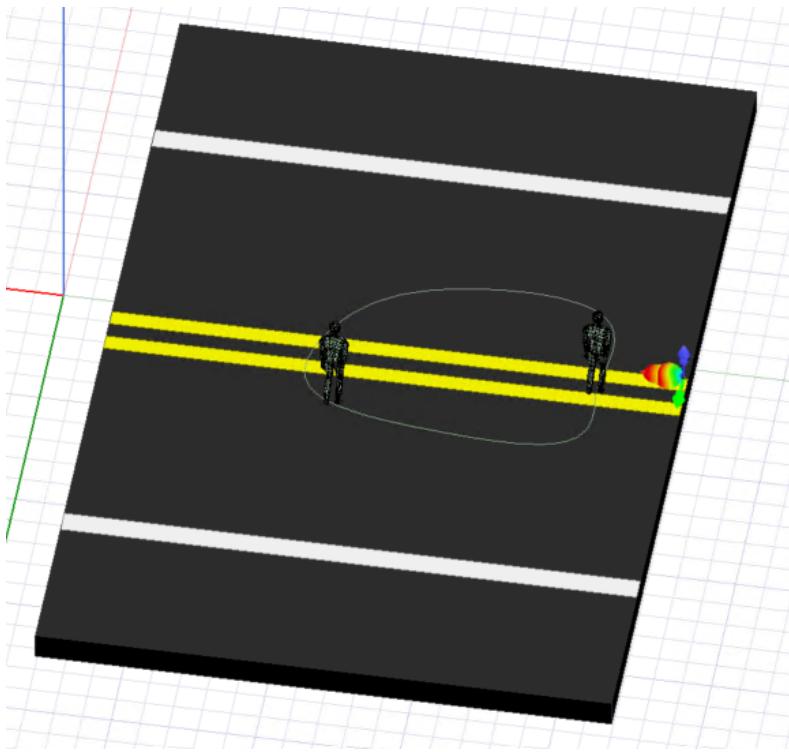
 Expand table

Operating system	OS architecture	GPU driver version	CUDA version
Windows 10 Pro, 22H2	x64	527.41	12

HFSS 2021 R2 was used to test the scalability performance of HFSS on Azure. Three models were used for the tests. The following sections describe the test models and provide the performance results for each.

Pedestrian model

The following image shows the detection and classification of vulnerable road users, such as pedestrians, via a sensor. [HFSS SBR+](#) is an asymptotic ray tracing electromagnetic solver that solves large problems.



The following table provides details about the model:

[Expand table](#)

Model name	Solver	Ray density	Maximum bounces	Distribution	Solution frequency	Far field observation points
Pedestrian	HFSS SBR+	2	3	Single point	77 GHz	519,841

This table shows the total elapsed times recorded for running the simulation with varying numbers of GPUs on the NC A100 v4 series VM:

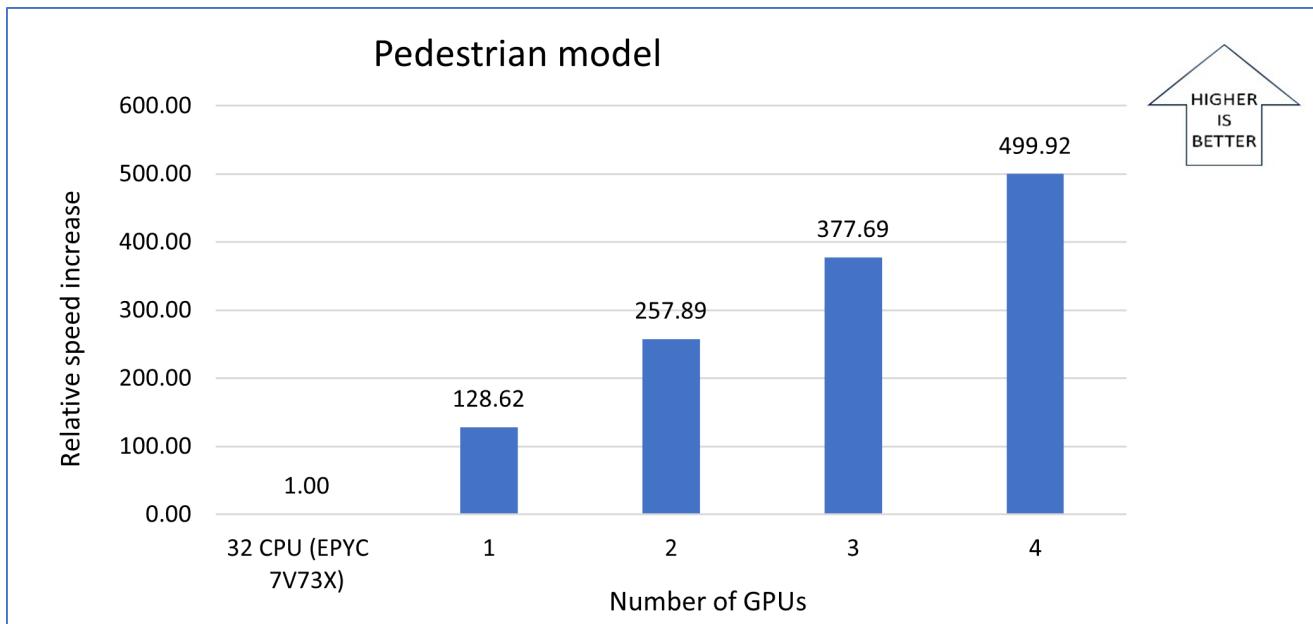
[Expand table](#)

VM/processor	Number of cores	Number of GPUs	Total elapsed time, in seconds	Relative speed increase
EPYC 7V73X	32	0	154,475	N/A
NC A100 v4	8	1	1,201	128.62
NC A100 v4	8	2	599	257.89
NC A100 v4	8	3	409	377.69
NC A100 v4	8	4	309	499.92

Note

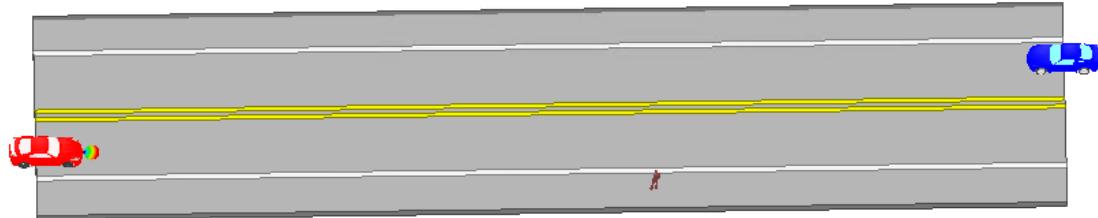
The time it takes to complete the simulation with only CPUs is used as a baseline to calculate the relative speed increases.

The following graph shows the relative speed increase as the number of GPUs increases:



Autonomous Vehicular Radar_ADP model

In the following image, one of the two vehicles uses automotive radar to detect obstacles or pedestrians before they're visible to the driver. Because radar uses electromagnetic waves to sense the environment, it can operate over long distances and in poor visibility or inclement weather conditions. Designing automotive radar that accurately captures diverse traffic situations is essential to making autonomous operations safe.



The following table provides details about the model:

[Expand table](#)

Model name	Solver	Ray density	Maximum bounces	Distribution	Solution frequency	Far field observation points
Autonomous Vehicular Radar_ADP	HFSS SBR+	4	5	Single point	77 GHz	260,281

This table shows the total elapsed times recorded for running the simulation with varying number of GPUs on the NC A100 v4 series VM:

[Expand table](#)

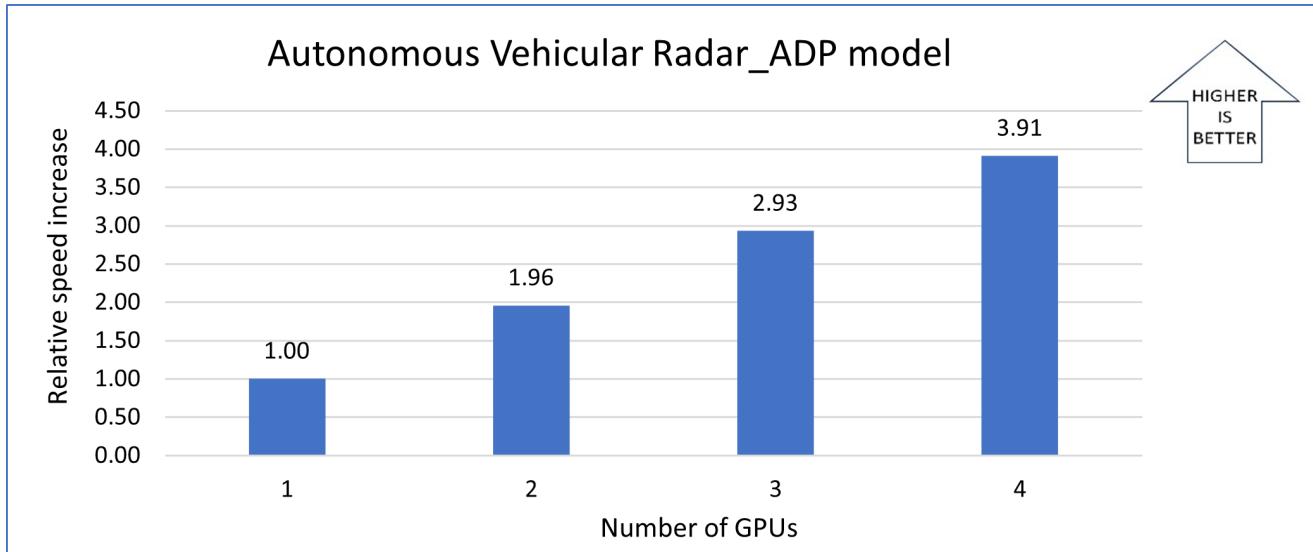
VM	Number of cores	Number of GPUs	Total elapsed time, in seconds	Relative speed increase
NC A100 v4	8	1	15,164	N/A
NC A100 v4	8	2	7,750	1.96

VM	Number of cores	Number of GPUs	Total elapsed time, in seconds	Relative speed increase
NC A100 v4	8	3	5,175	2.93
NC A100 v4	8	4	3,879	3.91

! Note

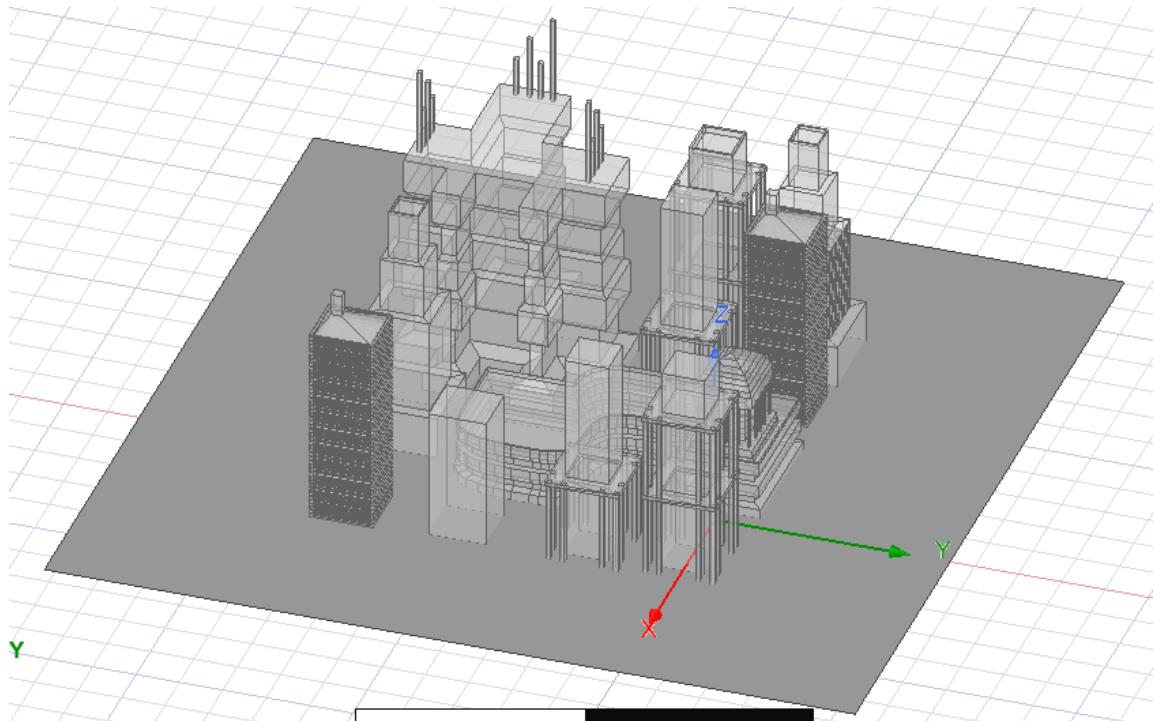
The time it takes to complete the simulation with one GPU is used as a baseline to calculate the relative speed increases.

The following graph shows the relative speed increase as the number of GPUs increases:



Urban_city model

The following image shows a 3D view of an urban city, with buildings adjacent to each other. HFSS is used to simulate the design and testing of various city features.



The following table provides details about the model:

[Expand table](#)

Model name	Solver	Ray density	Maximum bounces	Distribution	Solution frequency	Far field observation points
Urban_city	HFSS SBR+	1	5	Single point	35 GHz	519,841

This table shows the total elapsed times recorded for running the simulation with varying number of GPUs on the NC A100 v4 series VM.

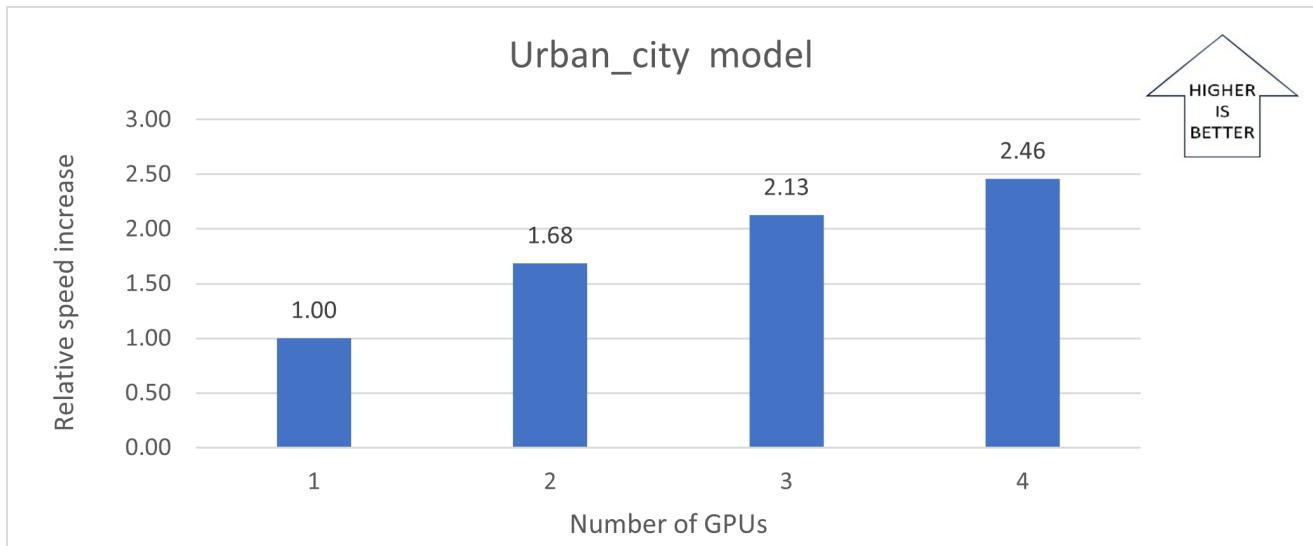
[Expand table](#)

VM	Number of cores	Number of GPUs	Total elapsed time, in seconds	Relative speed increase
NC A100 v4	8	1	4,635	N/A
NC A100 v4	8	2	2,753	1.68
NC A100 v4	8	3	2,181	2.13
NC A100 v4	8	4	1,886	2.46

Note

The time it takes to complete the simulation with one GPU is used as a baseline to calculate the relative speed increases.

The following graph shows the relative speed increase as the number of GPUs increases:



Notes about the tests

The tests for the Pedestrian model compare the use of CPUs to the use of GPUs. The 32-vCPU EPYC 7V73X processor is used as a baseline for these tests. The tests for the Autonomous Vehicular Radar_AD and Urban_city models, for which step sizes are smaller, require more computation time. Only GPU scale-up is shown for these two models. The elapsed time with one GPU is used as a baseline.

Azure cost

Only simulation running time is considered for these cost calculations. Installation time, simulation setup time, and software costs aren't considered.

You can use the [Azure pricing calculator](#) to estimate VM costs for your configurations.

The following tables provide elapsed times in hours. To compute the cost, multiply the elapsed time by the [Azure VM hourly cost for Windows](#). The Azure VM hourly rates are subject to change.

Elapsed times for the Pedestrian model

[Expand table](#)

Number of GPUs	Elapsed time, in hours
0 (CPUs are used)	42.91
1	0.33
2	0.17
3	0.11
4	0.09

Elapsed times for the Autonomous Vehicular Radar_ADP model

[Expand table](#)

Number of GPUs	Elapsed time, in hours
1	4.21
2	2.15
3	1.44
4	1.08

Elapsed times for the Urban_city model

[Expand table](#)

Number of GPUs	Elapsed time, in hours
1	1.29
2	0.76
3	0.61
4	0.52

Summary

- HFSS was successfully deployed and tested on NC A100 v4 series VMs on Azure.
- Running the Pedestrian model simulation with the SBR+ solver is 128 times faster with one GPU than running it without using GPUs. With four GPUs, it's as much as 500 times faster than it is without using GPUs.
- When four GPUs are used to run the Autonomous Vehicular Radar_ADP model, it runs 97% more efficiently than it does with one GPU.
- HFSS and the SBR+ solver use the power of GPUs to accelerate simulations. Azure provides VMs that are equipped with the latest GPUs.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Preetham Y M](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director of Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Ansys LS-DYNA on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [Ansys LS-DYNA](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running LS-DYNA on Azure.

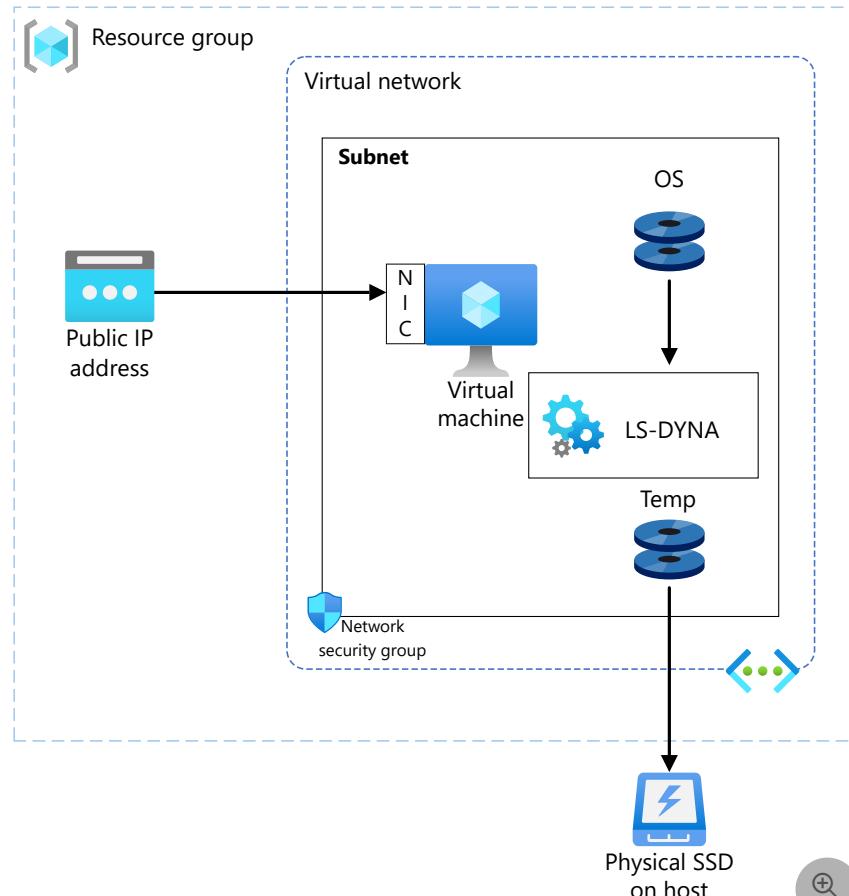
Ansys LS-DYNA is explicit simulation software for applications like drop tests, impact and penetration, smashes and crashes, and occupant safety. It simulates the response of materials to short periods of severe loading. It provides elements, contact formulations, material models, and other controls that can be used to simulate complex models with control over all the details of the problem.

LS-DYNA is used in the automotive, aerospace, construction, facilities, military, manufacturing, and bio-engineering industries.

Why deploy LS-DYNA on Azure?

- Modern and diverse compute options to meet your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- In a multi-node configuration, impressive scale-up as the number of nodes increases

Architecture



 Microsoft Azure



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Linux VM. For information about deploying the VM and installing the drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

The performance tests of LS-DYNA on Azure used an [HBv3-series](#) VM running Linux. The following table provides details about the VM.

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

Size	vCPU	Memory (GiB)	Memory bandwidth Gbps	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	Maximum data disks
Standard_HB120-64rs_v3	64	448	350	1.9	3.0	3.5	32

Required drivers

To use AMD CPUs on [HBv3-series](#) VMs, you need to install AMD drivers.

To use InfiniBand, you need to enable InfiniBand drivers.

LS-DYNA installation

Before you install LS-DYNA, you need to deploy and connect a Linux VM and install the required AMD and InfiniBand drivers.

For information about deploying the VM and installing the drivers, see [Run a Linux VM on Azure](#).

For information about installing LS-DYNA, see the [Ansys website](#).

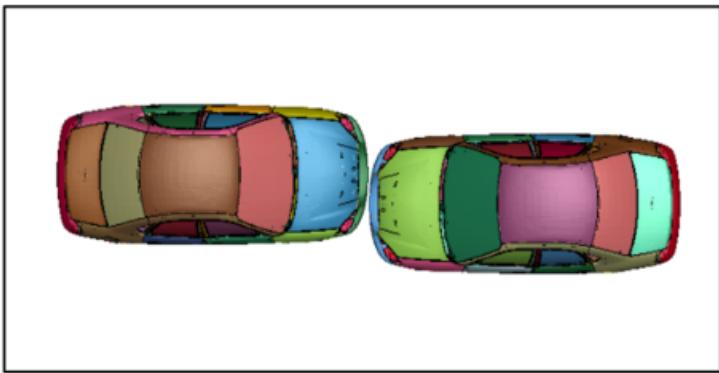
LS-DYNA performance results

The following table provides the details of the operating system that was used for testing.

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

Operating system	OS architecture	Processor
Ubuntu 20.04.3 LTS	x86-64	AMD EPYC 7V73X (Milan-X)

A crash simulation model was used for testing:



Here are the details of the model:

- **Size:** 18,299,158
- **Cell type:** Shell and solid
- **Solver:** LS-DYNA (Ansys 2022 R2)
- **Termination time:** 4 ms
- **Compiler:** Intel Fortran Compiler 19.0 SSE2

Based on the results of testing on a single node, the 64-core VM Standard_HB120-64rs_v3 is the best configuration, taking into account performance and license costs. This configuration was used in multi-node tests, with two, three, and four nodes.

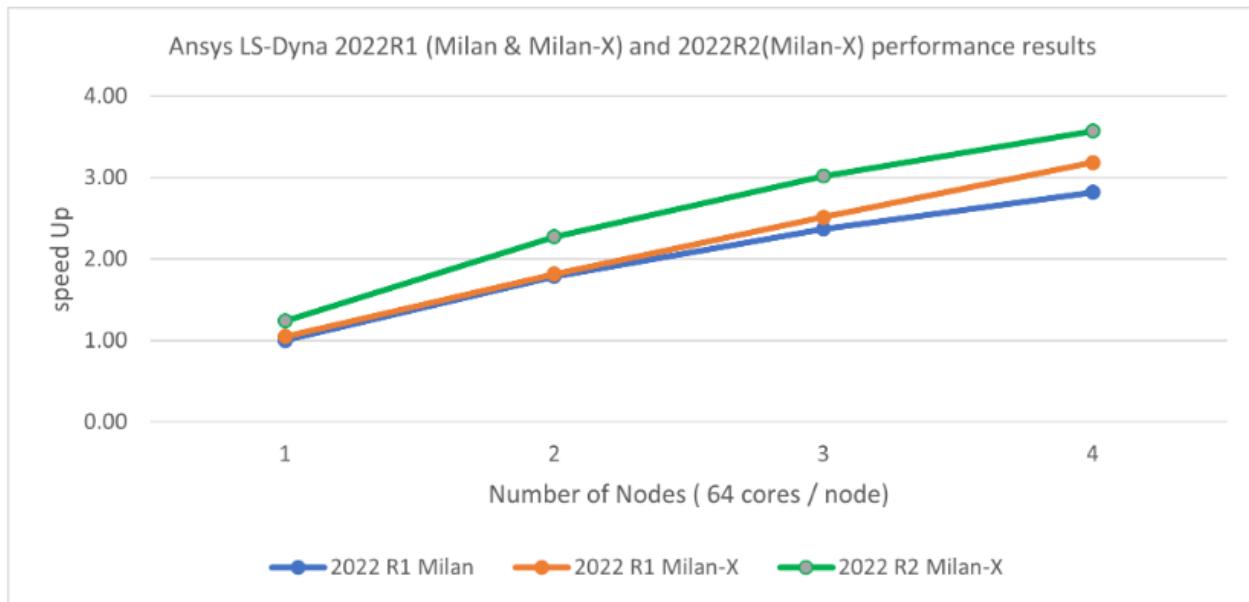
The following table shows the wall-clock times for running the simulation and the relative speed increases as the number of CPUs increases. LS-DYNA 2022 R1 on an HBv3 VM with an AMD Milan processor is used as a reference baseline to determine the speed increases. LS-DYNA 2022 R1 and LS-DYNA 2022 R2 were tested.

[Expand table](#)

Number of compute nodes	Number of CPUs	2022 R1 (seconds)	2022 R2 (seconds)	2022 R1, increase	2022 R2, increase
1	64	6,966	5,888	1.05	1.24
2	128	4,019	3,209	1.82	2.27
3	192	2,902	2,417	2.51	3.02
4	256	2,292	2,043	3.18	3.57

The performance scales impressively as the number of nodes increases.

This graph shows the relative speed increases:



Azure cost

The following table presents wall-clock times that you can use to calculate Azure costs. You can multiply the times presented here by the number of nodes and the Azure hourly rates for HBv3-series VMs to calculate costs. For the current hourly costs, see [Linux Virtual Machines Pricing](#).

Only the wall-clock times for running the model are presented in this table. Application installation time and license costs aren't included. These times are indicative. The actual times depend on the size of the model.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

[Expand table](#)

VM	Number of nodes	Number of cores	2022 R1 (hours)	2022 R2 (hours)
Standard_HB120-64rs_v3	1	64	1.94	1.64
Standard_HB120-64rs_v3	2	128	1.12	0.89
Standard_HB120-64rs_v3	3	192	0.81	0.67
Standard_HB120-64rs_v3	4	256	0.64	0.57

Summary

- Ansys LS-DYNA was successfully tested on HBv3-series Azure VMs.
- Based on earlier testing of LS-DYNA 2021 R2, simulations on a single-node configuration scale well up to 64 cores. After that point, the relative speed increase saturates.
- At each increment from one to four nodes, performance increases as the number of nodes increases.
- If we take costs into consideration, single-node and 2-node configurations are optimal.
- To optimize only for computation time, the 4-node configuration is best. Further testing is needed on 8-node and 16-node configurations.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Ansys Rocky on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [Ansys Rocky](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running Ansys Rocky on Azure.

Ansys Rocky is a 3D DEM (discrete element modeling) particle simulation application that simulates the flow behavior of bulk materials with complex particle shapes and size distributions. Typical applications include conveyor chutes, mills, mixers, and other material-handling equipment.

The solver in Rocky distributes and manages the combined memory of two or more GPU cards in a single motherboard to overcome memory limitations.

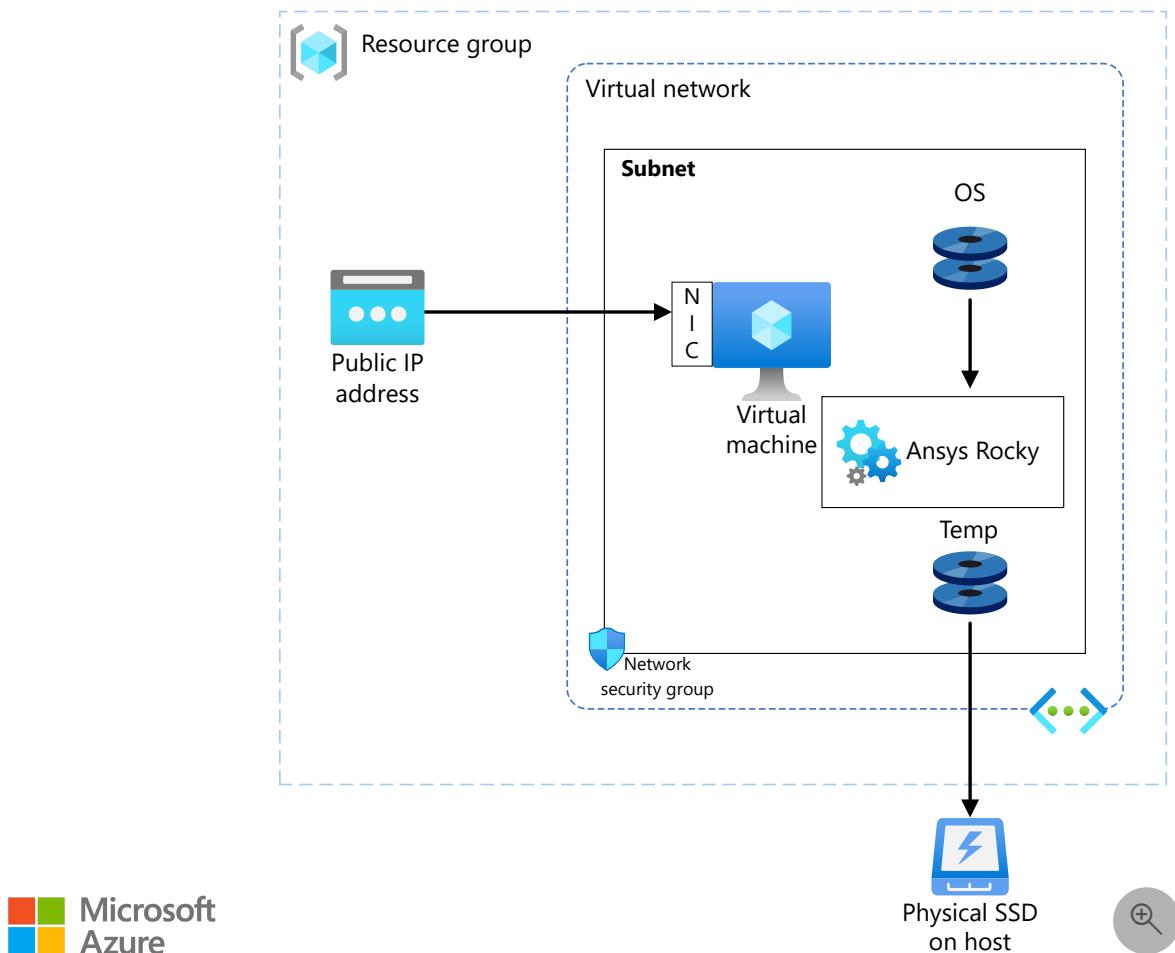
Ansys Rocky enables you to simulate a system with real particle shapes and sizes, specifying both spherical and non-spherical particle shapes, including shells and fibers.

Rocky is used in industries like food processing, manufacturing, pharmaceutical and biotech, medical and healthcare, agriculture equipment, energy, and mining/metals.

Why deploy Ansys Rocky on Azure?

- Modern and diverse compute options to meet your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Impressive performance results for simulations with varying levels of complexity

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Windows VM. For information about deploying the VM and installing the drivers, see [Windows VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

The performance tests of Ansys Rocky on Azure used [ND_A100_v4](#) and [NC_A100_v4](#) VMs running Windows. The following table provides details about the VMs.

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

VM size	vCPU	Memory, in GiB	Number of GPUs	GPU memory, in GiB	Maximum data disks	Maximum uncached disk throughput, in IOPS / MBps
Standard_ND96asr_v4	96	900	8	40	32	80,000 / 800
Standard_NC24ads_A100_v4	24	220	1	80	12	30,000 / 1,000
Standard_NC48ads_A100_v4	48	440	2	160	24	60,000 / 2,000
Standard_NC96ads_A100_v4	96	880	4	320	32	120,000 / 4,000

Required drivers

To take advantage of the GPU capabilities of [NC_A100_v4](#) and [ND_A100_v4](#) VMs, you need to install NVIDIA GPU drivers.

To use AMD processors on [NC_A100_v4](#) and [ND_A100_v4](#) VMs, you need to install AMD drivers.

Ansys Rocky installation

Before you install Ansys Rocky, you need to deploy and connect a VM, install an eligible Windows 10 or Windows 11 image, and install the required NVIDIA and AMD drivers.

For information about eligible Windows images, see [How to deploy Windows 10 on Azure](#) and [Use Windows client in Azure for dev/test scenarios](#).

i **Important**

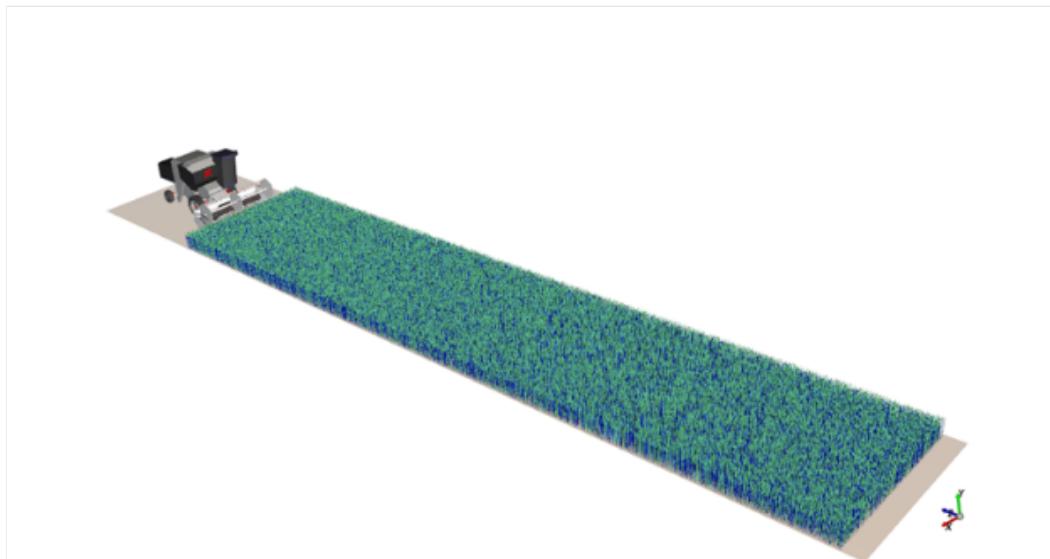
NVIDIA Fabric Manager installation is required for VMs that use NVLink or NVSwitch. NC_A100_v4 and ND_A100_v4 VMs use NVLink.

For information about deploying the VM and installing the drivers, see [Run a Windows VM on Azure](#).

For information about installing Ansys Rocky, see the [Ansys website](#).

Ansys Rocky performance results

A combine harvester simulation was used for the performance tests:



- **Total number of particles:** 110,827
- **Simulation duration:** 1 second

DEM particle simulation was used to test the application's performance. Two versions of Ansys Rocky were tested: 2021 R2.2 and 2022 R1.1.

The following table presents the wall-clock times for running the simulation, for both CPU and GPU configurations.

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

Rocky version	VM series	96 CPUs	1 GPU	2 GPUs	3 GPUs	4 GPUs
2021 R2.2	ND_A100_v4	10:57:58	00:53:24 ¹	00:44:30 ¹	00:40:42 ¹	00:46:01 ¹
2022 R1.1	ND_A100_v4	-	00:44:41 ¹	00:37:47 ¹	00:39:40 ¹	00:42:29 ¹
2022 R1.1	NC_A100_v4	-	00:40:19	00:32:11	00:30:55 ²	00:32:42

¹ In these cases, the number of GPUs was artificially limited. This VM has eight GPUs.

² In this case, the number of GPUs was artificially limited. This VM is available with one, two, or four GPUs.

The following table shows the relative speed increases as the CPUs are replaced by GPUs in increasing numbers. The 96-CPU configuration in the preceding table is used as the baseline for the comparisons.

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

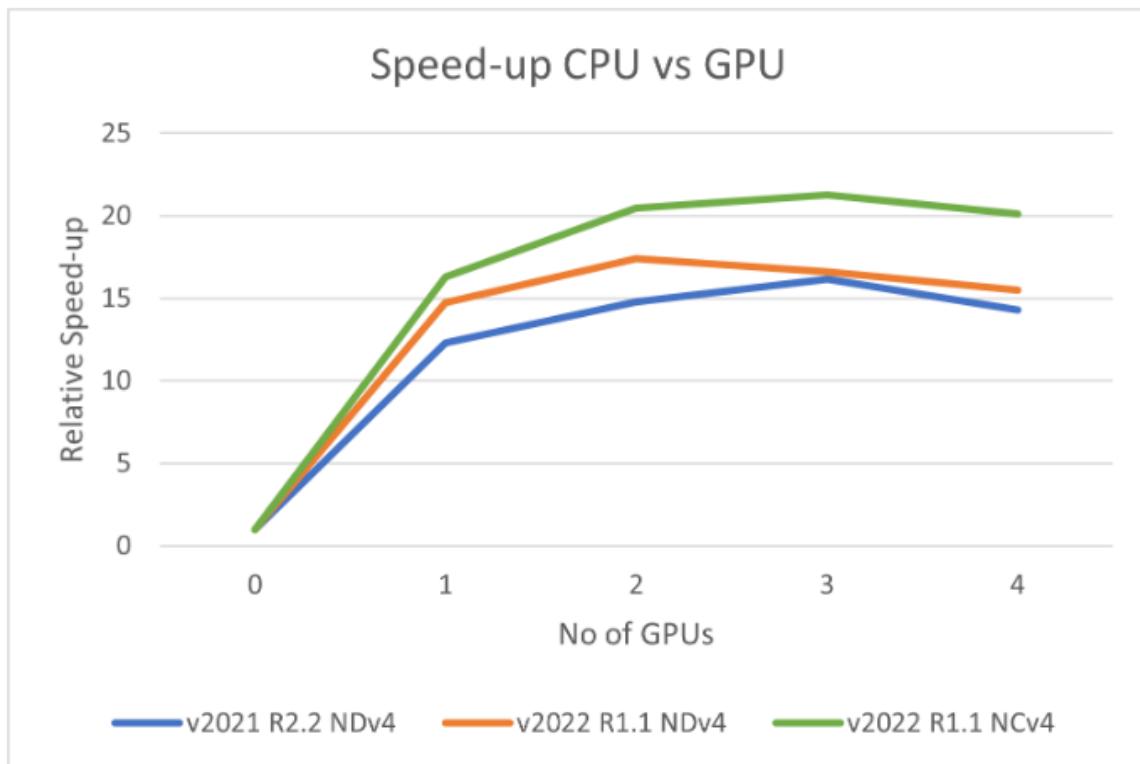
Rocky version	VM series	96 CPUs	1 GPU	2 GPUs	3 GPUs	4 GPUs
2021 R2.2	ND_A100_v4	1.00	12.32 ¹	14.79 ¹	16.17 ¹	14.30 ¹
2022 R1.1	ND_A100_v4	1.00	14.73 ¹	17.41 ¹	16.59 ¹	15.49 ¹

Rocky version	VM series	96 CPUs	1 GPU	2 GPUs	3 GPUs	4 GPUs
2022 R1.1	NC_A100_v4	1.00	16.32	20.45	21.29 ²	20.12

¹ In these cases, the number of GPUs was artificially limited. This VM has eight GPUs.

² In this case, the number of GPUs was artificially limited. This VM is available with one, two, or four GPUs.

Here's the same data, presented graphically:



In the preceding graph, 0 GPUs indicates that the simulation was run with only CPUs.

Azure cost

The following table presents wall-clock times that you can use to calculate Azure costs. You can multiply the times presented here by the Azure hourly rates for NCA100v4-series VMs to calculate costs. For the current hourly costs, see [Windows Virtual Machines Pricing](#).

Only the wall-clock times for running the model are presented in this table. Application installation time isn't included. These times are indicative. The actual times depend on the size of the model.

The wall-clock times for a full production-level simulation of the combine harvester model are longer than the times presented here, so the associated costs are higher.

The times here represent tests performed on Ansys Rocky 2022 R1.1.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

[Expand table](#)

VM size	Number of GPUs	Elapsed time, in hours
Standard_NC24ads_A100_v4	1	0.67
Standard_NC48ads_A100_v4	2	0.53
Standard_NC96ads_A100_v4	4	0.54

Summary

- Ansys Rocky was successfully tested on ND_A100_v4 and NC_A100_v4 VMs on Azure.
- The speed increases significantly when you upgrade from 96 CPUs to one GPU.
- For all models, there's an optimal configuration that achieves the best combination of price and performance. After that point, adding hardware doesn't scale the performance substantially. For this particular harvester model, optimal performance occurs with three GPUs. For more complex models, we expect the optimal number of GPUs to be higher. For models that are less complex, we expect it to be lower.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Windows VM on Azure](#)
- [HPC system and big-compute solutions](#)

- HPC cluster deployed in the cloud

Deploy Autodesk Civil 3D on an Azure virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [Autodesk Civil 3D](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running Civil 3D on Azure.

Civil 3D is a 3D model-based design solver that civil engineers can use for design automation and production, enabling multidisciplinary team coordination. Civil 3D:

- Includes purpose-built tools for critical civil engineering disciplines.
- Integrates automation and analysis with support for building information modeling throughout a project lifecycle.
- Connects project teams (when used with Collaboration for Civil 3D and BIM Collaborate Pro).
- Enables effective visualization and analysis of geospatial/geotechnical data from ArcGIS.

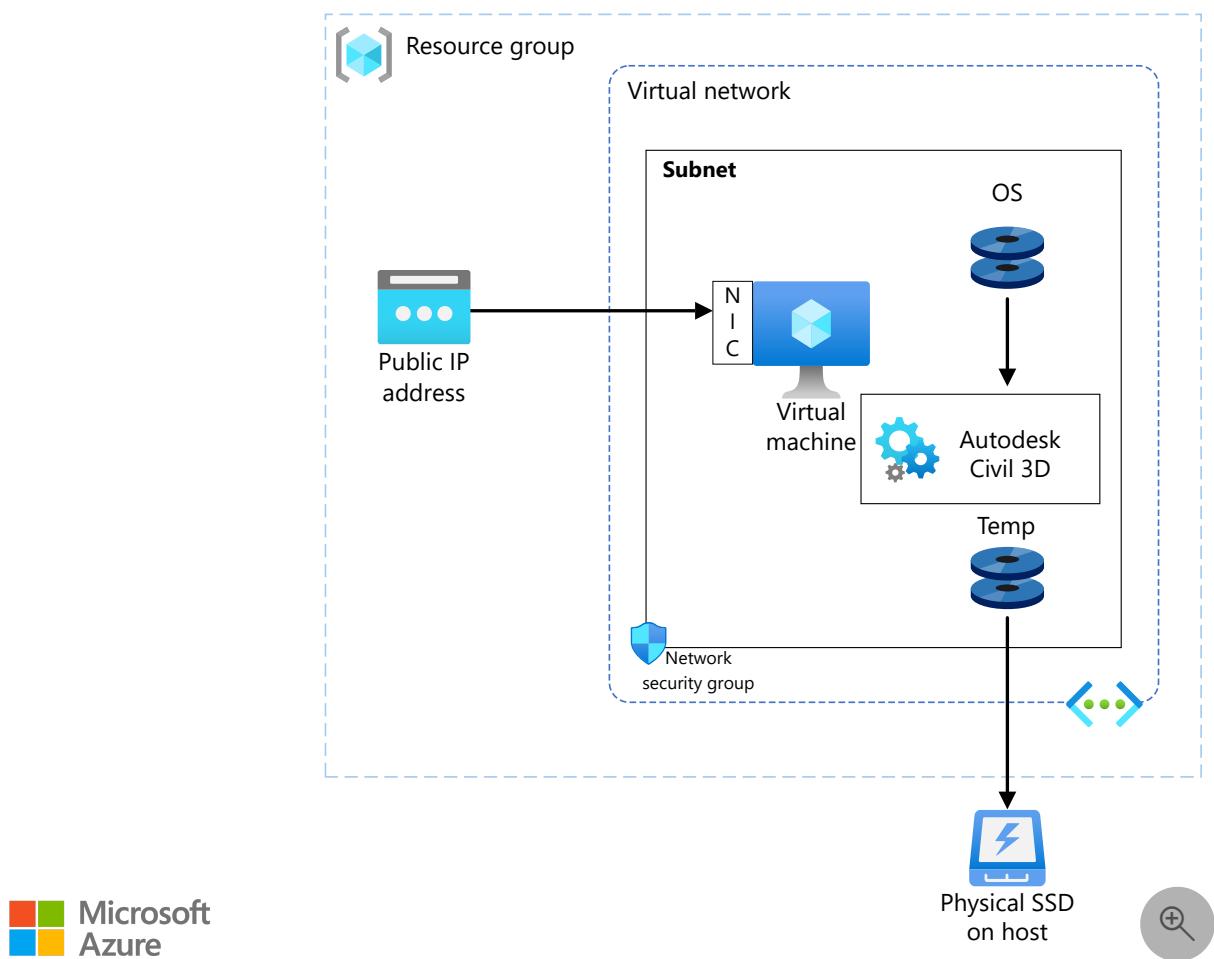
Civil 3D was originally designed as an AutoCAD add-on. Civil 3D extensions now incorporate grading optimization, rail switch, and crossing enhancements.

Civil 3D is used primarily to plan, design, and deliver land development, water, and transportation projects, including roads and highways, rail, bridges and tunnels, site development, and storm and sanitary networks. One of the main use cases is grading optimization. It's ideal for the automotive, energy, environment, construction, and facilities industries.

Why deploy Civil 3D on Azure?

- Modern and diverse compute options to align to your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Strong GPU acceleration, with increased performance as GPUs are added

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create Windows VMs.
 - For information about deploying the VM and installing the drivers, see [Windows VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VMs.
 - A public IP address connects the internet to the VM.
- A physical SSD is used for storage.

Compute sizing and drivers

[NVadsA10 v5](#) series VMs were used to test the performance of Civil 3D on Azure. The following table provides configuration details:

[Expand table](#)

VM name	vCPU	Memory (GiB)	SSD (GiB)	GPU	GPU memory (GiB)	Maximum data disks
Standard_NV6ads_A10_v5	6	55	180	1/6	4	4
Standard_NV18ads_A10_v5	18	220	720	1/2	12	8
Standard_NV36ads_A10_v5	36	440	720	1	24	16

Required drivers

To take advantage of the GPU capabilities of [NVadsA10 v5](#) series VMs, you need to install NVIDIA GPU drivers.

Civil 3D installation

Before you install Civil 3D, you need to deploy and connect a VM, install an eligible Windows 10 or Windows 11 image, and install the required NVIDIA GPU drivers.

For information about eligible Windows images, see [How to deploy Windows 10 on Azure](#) and [Use Windows client in Azure for dev/test scenarios](#).

For information about deploying the VM and installing the drivers, see [Run a Windows VM on Azure](#).

For detailed instructions, see the [Autodesk installation instructions](#).

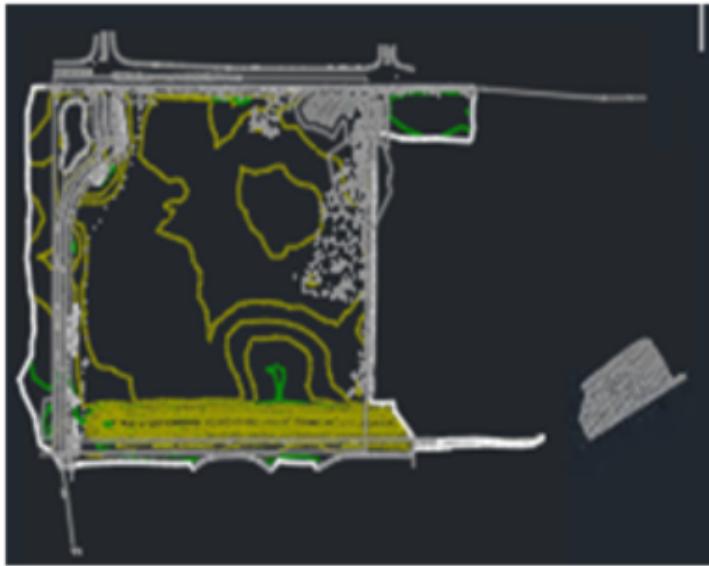
Civil 3D performance results

For this performance analysis, we used the Civil 3D 2023 trial version on Windows, on [NVadsA10 v5 series](#) VMs.

We performed the [Surfaces](#) and [Corridors](#) tutorials for these tests.

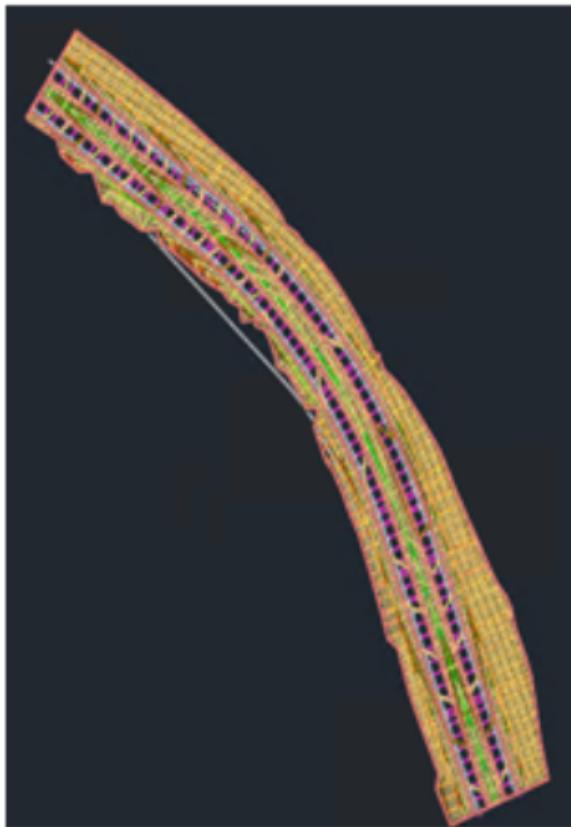
In the Surfaces Tutorial, we performed the [Rendering a Surface](#) exercise.

Here's an example of a rendered surface:



In the Corridors Tutorial, we performed the [Visualizing a Corridor](#) exercise.

Here's an example of a rendered corridor:



The following table shows the rendering sizes of the images created in the tests:

expand [Expand table](#)

Image number	Rendering size
Image 1	800 x 600 px, SVGA

Image number	Rendering size
Image 2	1920 x 1080 px, Full HDTV
Image 3	3300 x 2500 px (11 x 8.5 in @300 dpi)
Image 4	5100 x 3300 px (17 x 11 in @300 dpi)
Image 5	2480 x 3508 px (ISO A4 @300 dpi)
Image 6	3508 x 4961 px (ISO A3 @300 dpi)

Results on NVadsA10 v5

The following tables and graphs show the rendering times and relative speed increases achieved during the tests.

This table shows the rendering time, in seconds, for the six Surface-7 images:

[Expand table](#)

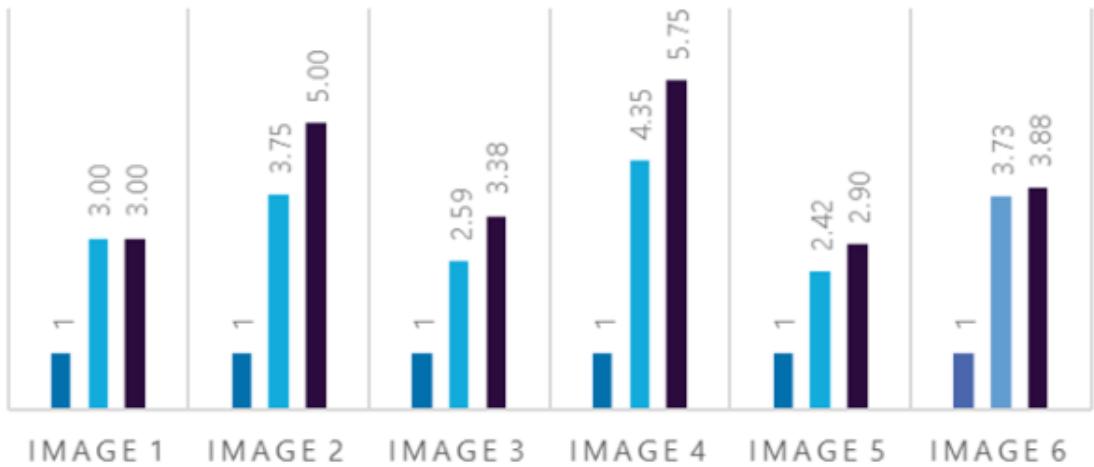
Number of GPUs	Number of CPUs	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6
1/6 GPU	6	3	15	44	161	29	97
1/2 GPU	18	1	4	17	37	12	26
1 GPU	36	1	3	13	28	10	25

This graph shows the relative speed increases for the Surface-7 drawings as the number of GPUs increases:

RELATIVE SPEEDUP

SURFACE-7

■ 1/6 GPU ■ 1/2 GPU ■ 1 GPU

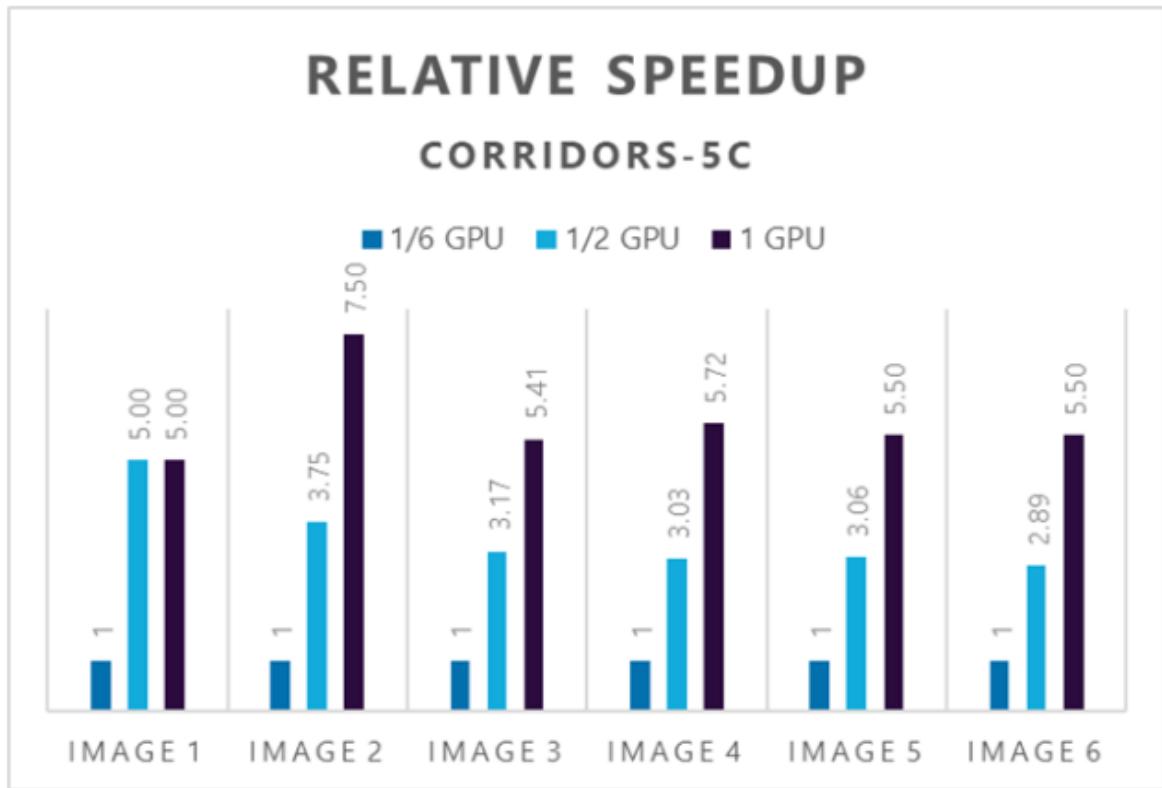


This table shows the rendering time, in seconds, for the six Corridor-5c images:

[Expand table](#)

Number of GPUs	Number of CPUs	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6
1/6 GPU	6	5	30	92	206	55	110
1/2 GPU	18	1	8	29	68	18	38
1 GPU	36	1	4	17	36	10	20

This graph shows the relative speed increases for the Corridor-5c drawings as the number of GPUs increases:



Azure cost

Only rendering time is considered for these cost calculations. Application installation time isn't considered.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

Surface-7 drawing on NVadsA10 v5

[Expand table](#)

VM size	Number of GPUs	Total time in hours
6 vCPU	1/6	0.10
18 vCPU	1/2	0.03
36 vCPU	1	0.02

Corridors-5c drawing on NVadsA10 v5

[Expand table](#)

VM size	Number of GPUs	Total time in hours
6 vCPU	1/6	0.14
18 vCPU	1/2	0.05
36 vCPU	1	0.02

To compute the cost, multiply the total time by the Azure hourly cost. For the current hourly costs, see [Windows Virtual Machines Pricing](#).

Summary

- Civil 3D was successfully tested on NVadsA10_v5 series VMs on Azure.
- NVadsA10 v5 series VMs demonstrated good GPU acceleration. Adding GPUs generally increases the speed.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU optimized virtual machine sizes](#)
- [Windows virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Windows VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Autodesk Inventor on a virtual machine

Azure Virtual Machines Azure Virtual Network

This article briefly describes the steps for running [Autodesk Inventor](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running Inventor on Azure.

Inventor is a 3D CAD application that provides professional-grade mechanical design, documentation, and product simulation tools. Inventor:

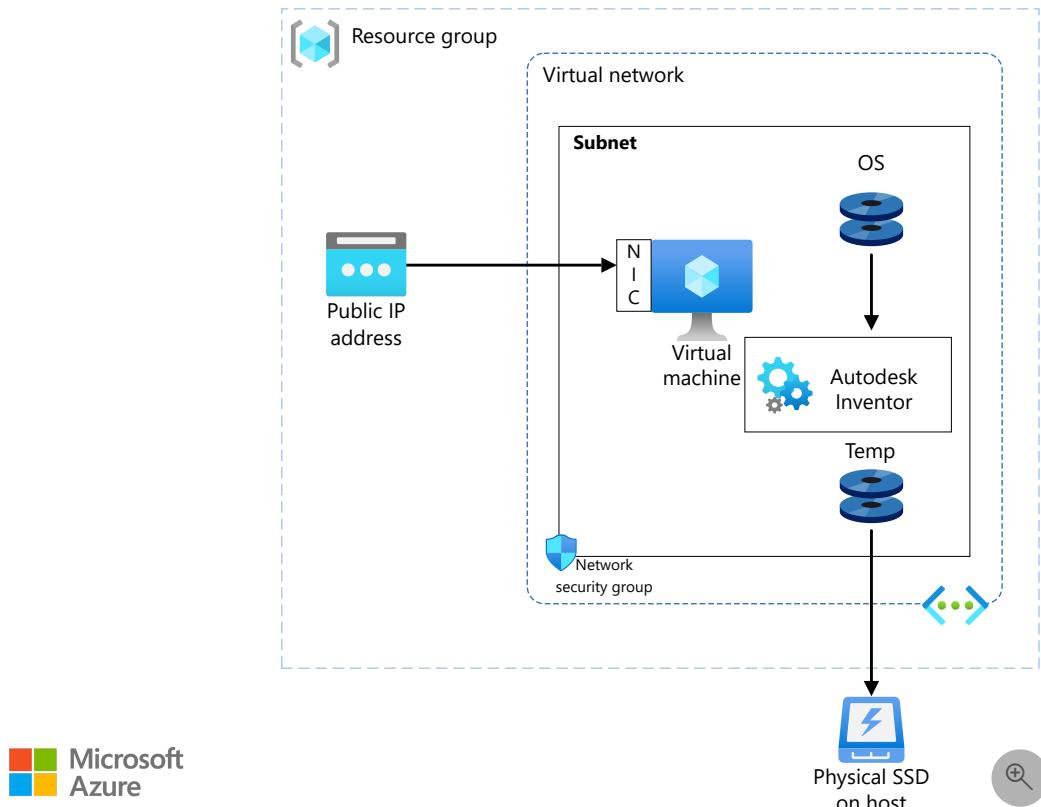
- Is rules-based and includes parametric, direct, freeform design capabilities.
- Integrates tools for sheet metal, frame design, tube and pipe, cable and harness, presentations, rendering, simulation, and machine design.
- Accesses the Forge Design Automation API for running job processes in the cloud.
- Enables Fusion 360/Revit/MCAD interoperability.

Inventor software is primarily used by mechanical engineers to quickly model, simulate, and communicate design ideas. It's well-suited to engineers who need automated and specialized tools to design components and prepare them for manufacturing. One of the main use cases is GPU ray tracing, which supports the hardware ray tracing that's used on recent graphics cards (GPUs).

Why deploy Inventor on Azure?

- Modern and diverse compute options to align to your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create Windows VMs.
 - For information about deploying the VM and installing the drivers, see [Windows VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VMs.
 - A public IP address connects the internet to the VM.
- A physical SSD is used on the host for storage.

Compute sizing and drivers

NCasT4_v3 and NVadsA10_v5 series VMs running Windows were used to test the performance of Inventor on Azure. The following table provides the configuration details:

[Expand table](#)

VM size	vCPU	Memory, in GiB	Temporary storage (SSD), in GiB	Number of GPUs	GPU memory, in GiB	Maximum data disks
Standard_NC4as_T4_v3	4	28	180	1	16	8
Standard_NC8as_T4_v3	8	56	360	1	16	16
Standard_NC16as_T4_v3	16	110	360	1	16	32
Standard_NC64as_T4_v3	64	440	2880	4	64	32
Standard_NV6ads_A10_v5	6	55	180	1/6	4	4
Standard_NV18ads_A10_v5	18	220	720	1/2	12	8
Standard_NV36ads_A10_v5	36	440	720	1	24	16

NCasT4_v3-series VMs are powered by [NVIDIA Tesla T4](#) GPUs and AMD EPYC 7V12 (Rome) CPUs. The VMs have as many as 4 NVIDIA Tesla T4 GPUs with 16 GB of memory each, and as many as 64 non-multithreaded AMD processor cores at a base frequency of 2.45 GHz, with a total memory of 440 GB.

NVadsA10 v5-series virtual machines are powered by [NVIDIA A10](#) GPUs and AMD EPYC 74F3V (Milan) CPUs with a base frequency of 3.2 GHz. NVadsA10 v5-series VMs have partial NVIDIA GPUs.

Required drivers

To use Inventor on NCasT4_v3 and NVadsA10_v5 VMs, you need to install NVIDIA and AMD drivers.

Inventor installation

Before you install Inventor, you need to deploy and connect a VM, install an eligible Windows 10 or Windows 11 image, and install the required NVIDIA and AMD drivers.

For information about eligible Windows images, see [How to deploy Windows 10 on Azure](#) and [Use Windows client in Azure for dev/test scenarios](#).

For information about deploying the VM and installing the drivers, see [Run a Windows VM on Azure](#).

For installation instructions, see the [Autodesk download and install page](#).

Inventor performance results

Inventor 2022.2.2 Professional and the free trial version of Inventor Professional 2023 were tested.

The [BenchMark HD](#) and [InvMark](#) Inventor add-ons were used to measure specific performance parameters of Inventor Professional on Azure.

Results for Inventor Professional 2022, using BenchMark HD

The following table shows elapsed times in seconds for tests on various sizes of NCast4 and NVadsA10 v5 VMs:

[Expand table](#)

VM series	VM size	Modeling	Graphics	Drawing	Ray tracing	File export	Save
NCast4	4 vCPU (1 GPU)	48.545	31.93	53.11	175.731	32.692	12.614
NCast4	8 vCPU (1 GPU)	48.615	31.106	47.056	74.841	32.794	10.143
NCast4	16 vCPU (1 GPU)	46.773	35.121	45.429	35.893	32.822	10.328
NCast4	64 vCPU (4 GPU)	53.927	37.467	51.671	10.387	33.668	11.746
NVadsA10 v5	6 vCPU (1/6 GPU)	33.981	29.889	38.807	108.275	24.203	6.646
NVadsA10 v5	18 vCPU (1/2 GPU)	33.811	23.957	33.96	36.669	23.915	6.547
NVadsA10 v5	36 vCPU (1 GPU)	33.599	24.426	33.163	19.324	23.134	6.139

The following table provides results for various BenchMark HD performance indices. A higher number indicates better performance.

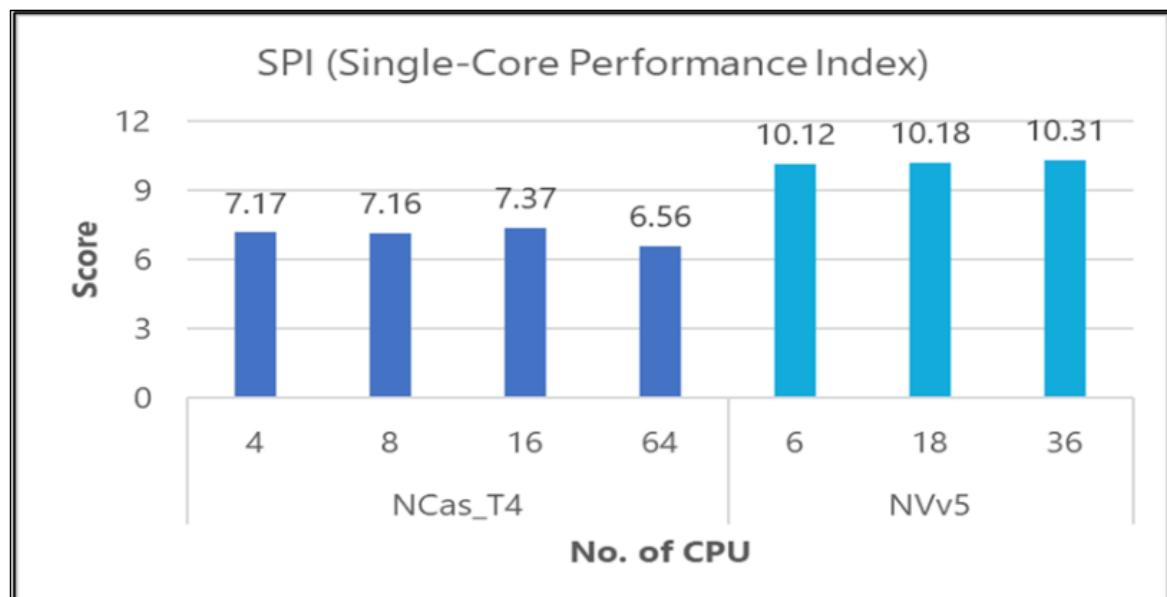
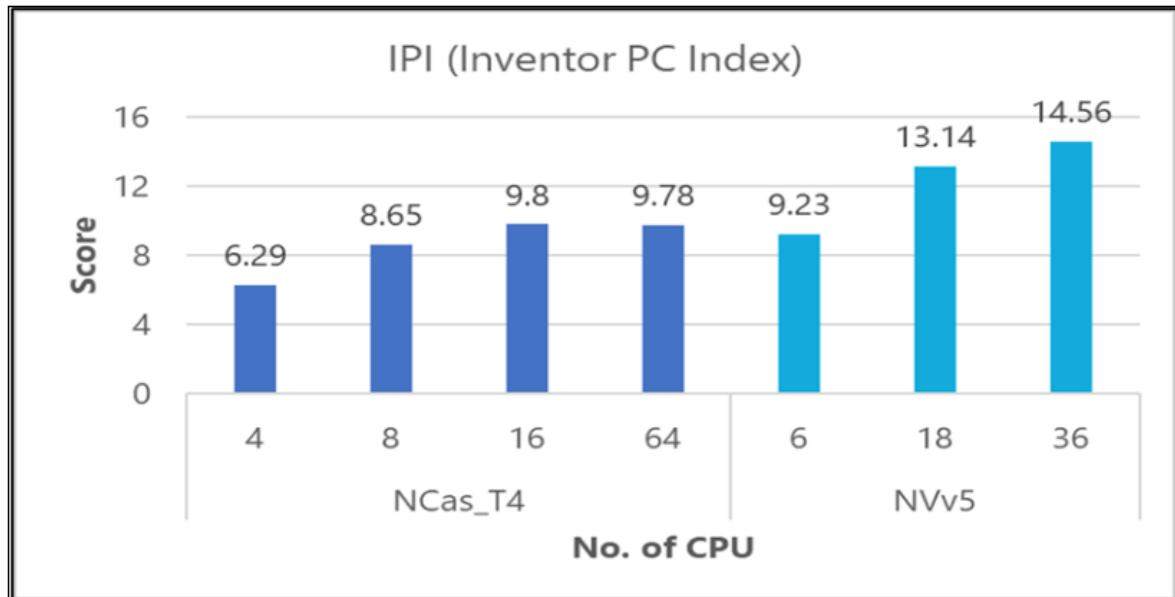
[Expand table](#)

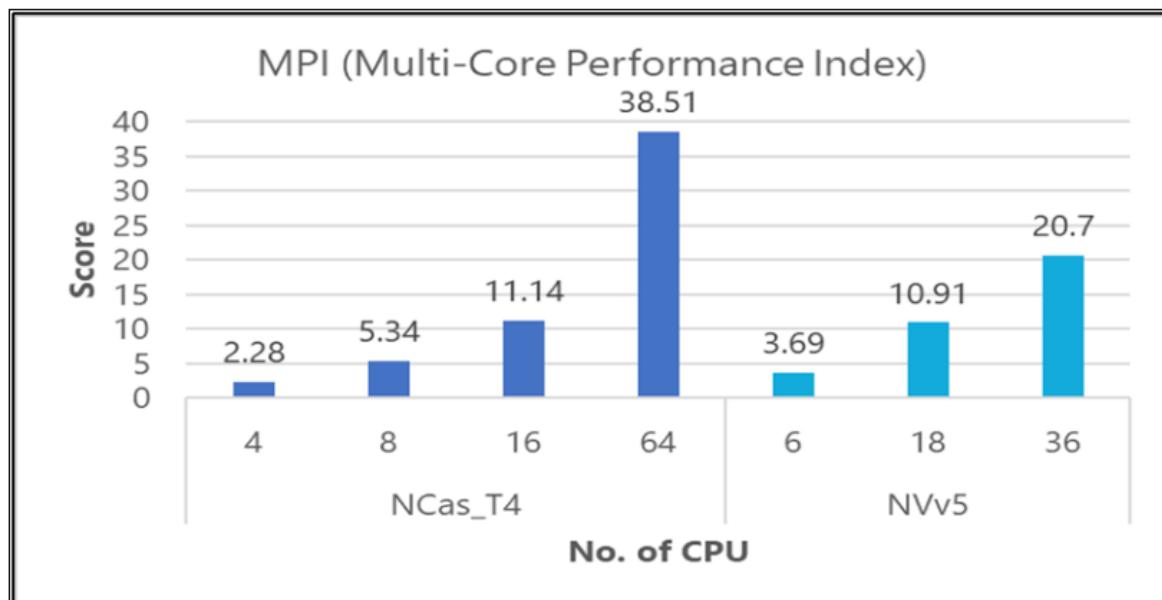
VM series	VM size	Inventor PC Index (IPI)	Single-core Performance Index (SPI)	Multi-core Performance Index (MPI)
NCast4	4 vCPU (1 GPU)	6.29	7.17	2.28
NCast4	8 vCPU (1 GPU)	8.65	7.16	5.34
NCast4	16 vCPU (1 GPU)	9.8	7.37	11.14
NCast4	64 vCPU (4 GPU)	9.78	6.56	38.51
NVadsA10 v5	6 vCPU (1/6 GPU)	9.23	10.1	3.69
NVadsA10 v5	18 vCPU (1/2 GPU)	13.1	10.2	10.91
NVadsA10	36 vCPU (1)	14.6	10.3	20.7

VM series	VM size	Inventor PC Index (IPI)	Single-core Performance Index (SPI)	Multi-core Performance Index (MPI)
v5	GPU)			

Comparisons, BenchMark HD

Overall, performance is better on NVadsA10 v5 VMs. The following graphs provide details.





Results for Inventor Professional 2022, using InvMark

The following table shows the scores of InvMark tests on various sizes of NCasT4 and NVadsA10 v5 VMs:

[Expand table](#)

VM series	VM size	Graphics	Drawing	FEA	Dynamic simulation	Assembly constraint	Ray tracing	Data translation	Assembly pattern
NCasT4	4 vCPU (1 GPU)	1412	567	901	1100	1108	458	362	1003
NCasT4	8 vCPU (1 GPU)	1440	733	928	1239	1144	897	622	1047
NCasT4	16 vCPU (1 GPU)	1358	800	915	1217	1167	1655	650	947
NCasT4	64 vCPU (4 GPU)	1060	848	810	953	1090	4016	596	905
NVadsA10 v5	6 vCPU (1/6 GPU)	1629	844	1122	1495	1439	695	891	1308
NVadsA10 v5	18 vCPU (1/2 GPU)	2934	1097	1080	1588	1468	1768	934	1145
NVadsA10 v5	36 vCPU	3180	1213	1098	1608	1602	2987	980	1252

VM series	VM size	Graphics	Drawing	FEA	Dynamic simulation	Assembly constraint	Ray tracing	Data translation	Assembly pattern
-----------	---------	----------	---------	-----	--------------------	---------------------	-------------	------------------	------------------

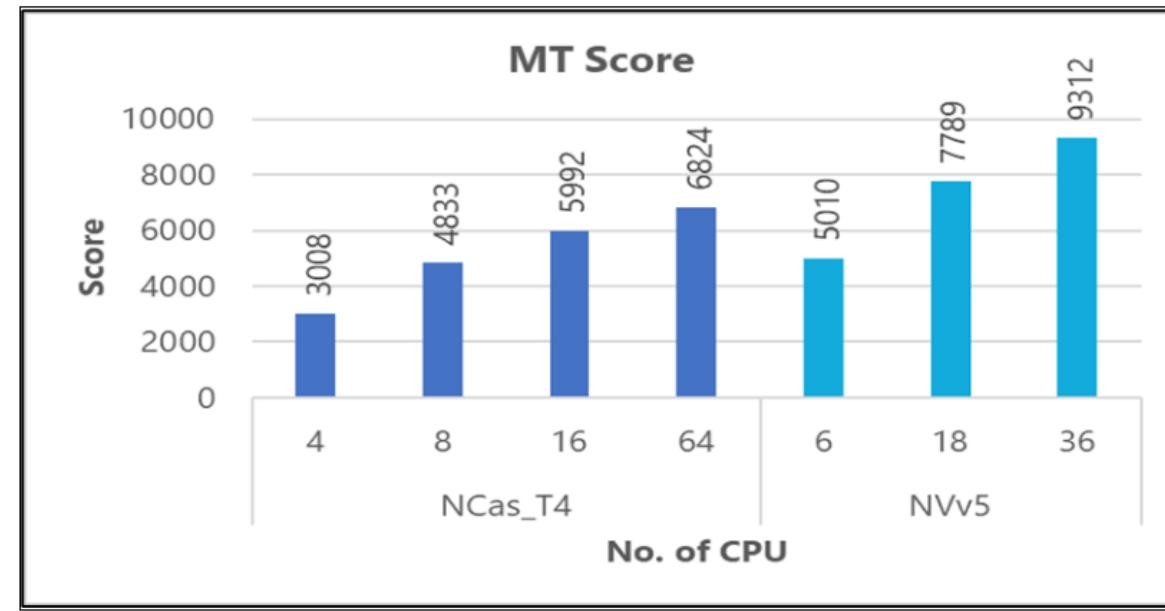
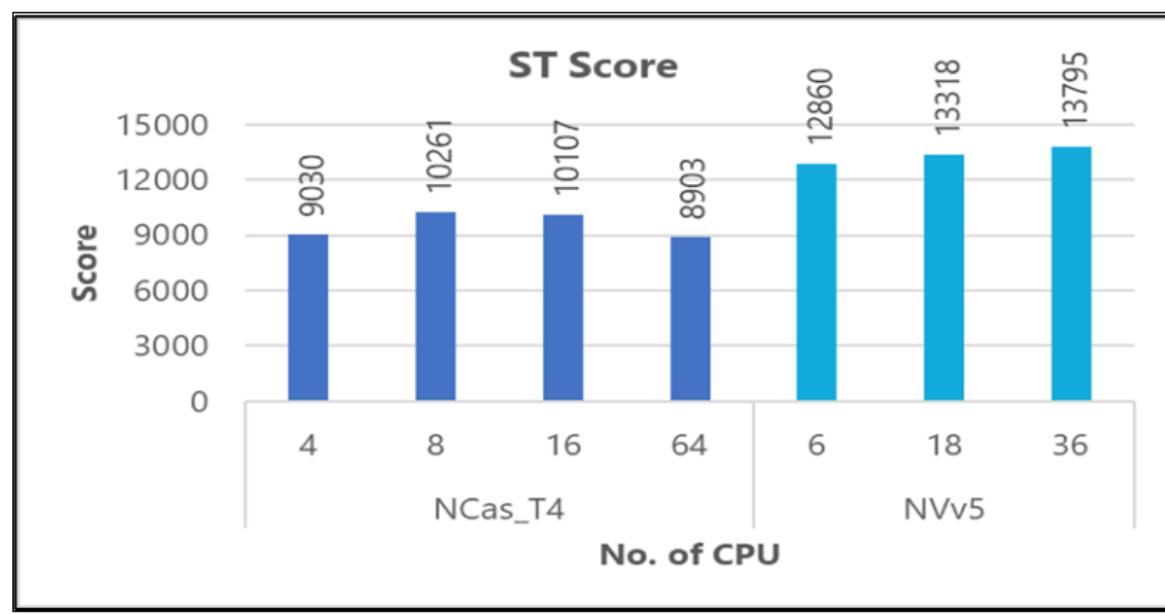
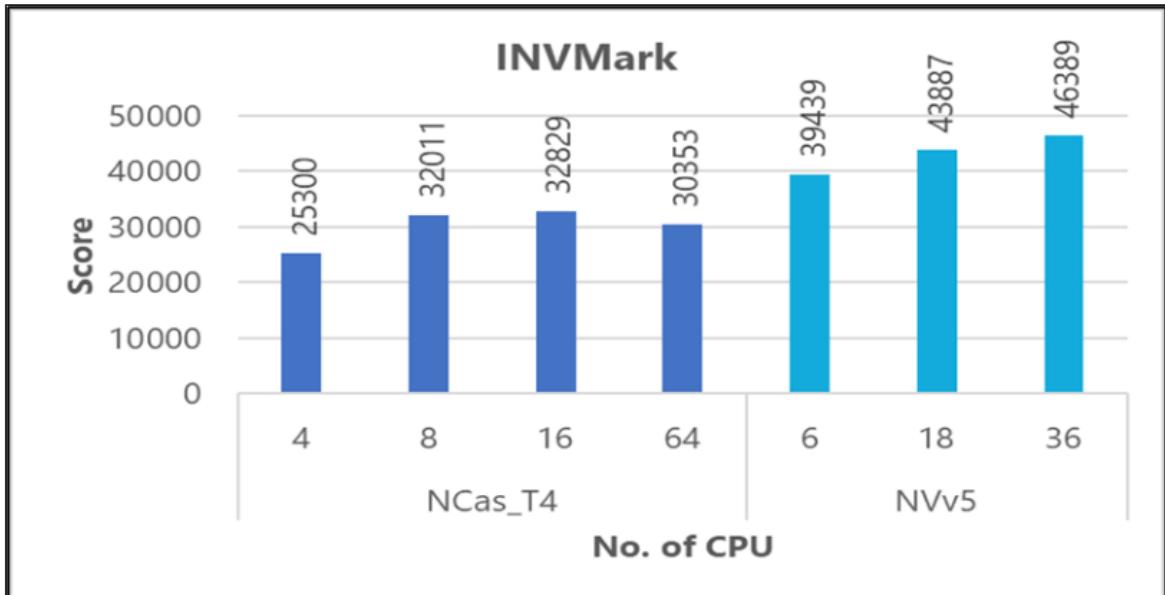
The following table provides results for various InvMark performance indices. A higher number indicates better performance.

[Expand table](#)

VM series	VM size	InvMark	ST score	MT score
NCasT4	4 vCPU (1 GPU)	25300	9030	3008
NCasT4	8 vCPU (1 GPU)	32011	10261	4833
NCasT4	16 vCPU (1 GPU)	32829	10107	5992
NCasT4	64 vCPU (4 GPU)	30353	8903	6824
NVadsA10 v5	6 vCPU (1/6 GPU)	39439	12860	5010
NVadsA10 v5	18 vCPU (1/2 GPU)	43887	13318	7789
NVadsA10 v5	36 vCPU (1 GPU)	46389	13795	9312

Comparisons, InvMark

Overall, performance is better on NVadsA10 v5 VMs. The following graphs provide details.



Azure cost

Only total elapsed times, as measured by BenchMark HD, are considered for these cost calculations. Application installation time isn't considered.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

 [Expand table](#)

VM series	VM size	Number of GPUs	Total time
NCasT4	NC4as_T4_v3	1	6 minutes
NCasT4	NC8as_T4_v3	1	4 minutes
NCasT4	NC16as_T4_v3	1	3 minutes, 44 seconds
NCasT4	NC64as_T4_v3	4	3 minutes, 31 seconds
NVadsA10 v5	6 vCPU	1/6	4 minutes, 2 seconds
NVadsA10 v5	18 vCPU	1/2	2 minutes, 38 seconds
NVadsA10 v5	36 vCPU	1	2 minutes, 19 seconds

You can use the total time and the Azure hourly cost to compute the total cost. For the current hourly costs, see [Windows Virtual Machines Pricing](#).

Summary

- Inventor Professional was successfully tested on NCasT4_v3 and NVadsA10_v5 series VMs on Azure.
- Performance is better on NVadsA10_v5 VMs than it is on NCasT4_v3 VMs. NVadsA10_v5 VMs are also less expensive.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Windows virtual machines in Azure](#)
- [Virtual networks and virtual machines in Azure](#)
- [Learning path: Run HPC applications on Azure](#)

Related resources

- [Run a Windows VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Autodesk Maya on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running the [Autodesk Maya](#) application on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running Maya on Azure.

Maya is a professional toolset for 3D animation, modeling, simulation, and rendering. You can use Maya to create realistic characters, landscapes, and action sequences. Artists, modelers, and animators rely on this award-winning toolset to add lifelike 3D assets to animated and live-action films, TV shows, and video games.

Key features of Maya include:

- Bifrost for Maya, a plug-in that you can use to create physically accurate simulations in a single visual programming environment.
- USD in Maya, an extension for the Universal Scene Description (USD) framework. You can use this extension to load and edit large datasets quickly and to use native tools to work directly with data.
- Fast playback, which provides a fast way to review animations and leads to reduced *playblasts* with the Cached Playback system in Viewport 2.0. A playblast is a real-time preview of an animation in Maya.
- Unreal Live Link for Maya, a plug-in that you can use to stream real-time animation data from Maya to Unreal.
- A nondestructive, clip-based nonlinear time editor for making high-level animation edits.
- A graph editor. You can use this graphical representation of scene animation to create, view, and modify animation curves.
- Polygon modeling, a feature for creating 3D models that uses geometry that's based on vertices, edges, and faces.
- Nonuniform rational basis spline (NURBS) modeling. By using this feature, you can construct 3D models from geometric primitives and drawn curves.
- Character setup for creating sophisticated skeletons, inverse kinematics (IK) handles, and deformers for characters that deliver lifelike performances.

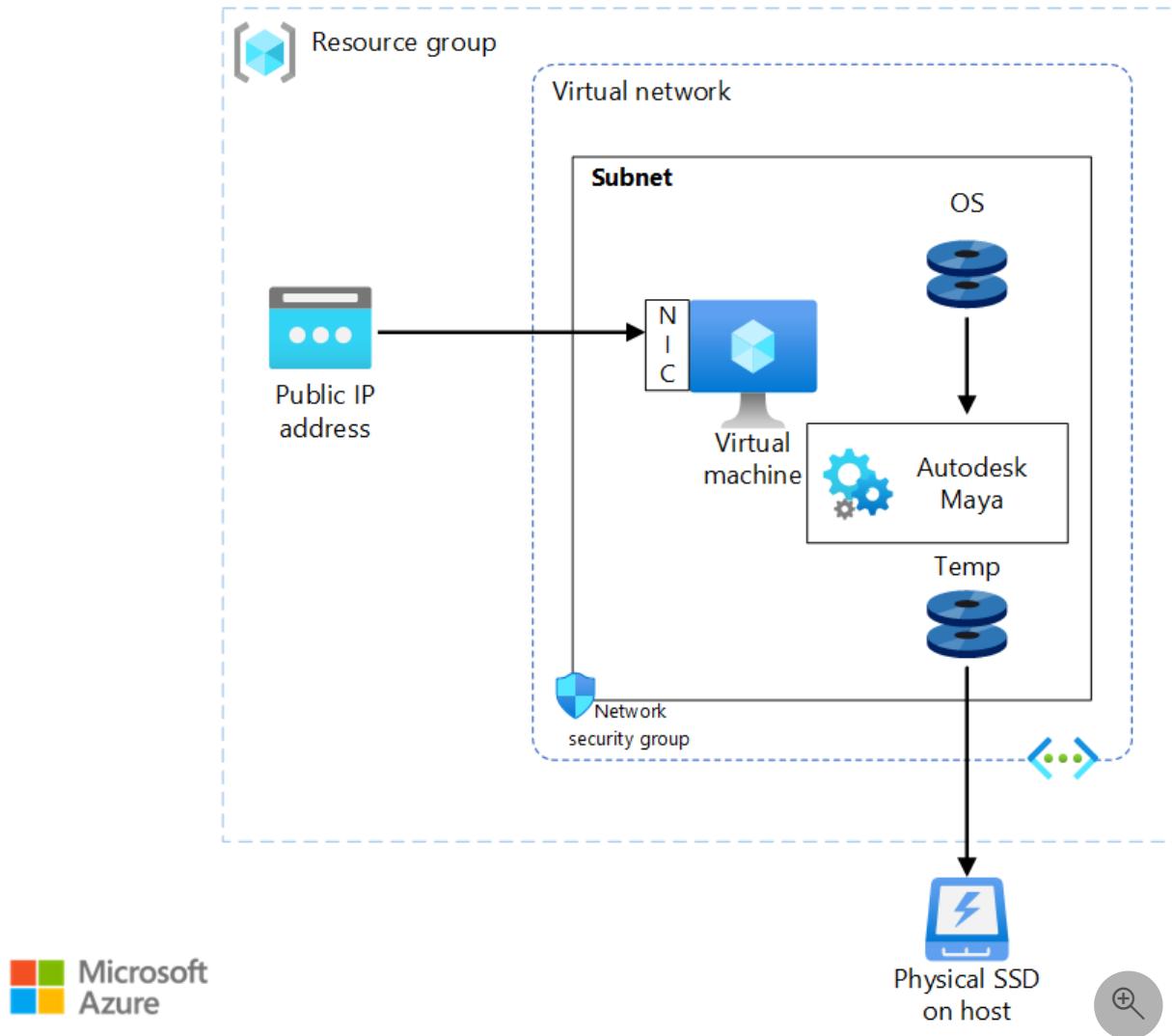
Maya was developed by Autodesk and offers a wide range of powerful tools for animation, simulation, and modeling. You can also use Maya for motion graphics, virtual reality, UV maps, low poly, and character creation. This 3D modeling software is popular in the video game industry. The Maya application can generate 3D assets for games but also for film, TV, and commercials. Besides offering a vast library of animation tools, Maya is customizable. You can use scripting languages like Maya Embedded Language (MEL) and Python to extend Maya functionality.

For information about a plug-in for using an [Arnold renderer](#) directly in Maya, see the [Arnold for Maya user guide](#).

Why deploy Maya on Azure?

- Modern and diverse compute options to align with your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Strong GPU acceleration, with increased performance as you add GPUs

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create Windows VMs. For information about deploying a VM and installing drivers, see [Run a Windows VM on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
- [Network security groups](#) restrict access to VMs at the subnet level.
- A public IP address provides users with access to Maya via the internet.

- [Azure Disk Storage](#) provides a physical solid-state drive (SSD) for storage.

Deploy infrastructure and install Maya

Deploy Azure VMs. Before you install Maya, deploy your Azure VMs. Use a [NVadsA10_v5-series](#) VM to run Maya. You should use a Premium SSD managed disk and attach it to the VM.

Create and configure the supporting infrastructure. Configure a public IP address for inbound connectivity. Use network security groups to provide security for the subnet.

Install NVIDIA drivers. To take advantage of the GPU capabilities of NVadsA10_v5-series VMs, install [NVIDIA GPU drivers](#). For information about deploying VMs and installing the drivers, see [Run a Windows VM on Azure](#).

Compute sizing

[Expand table](#)

Size	vCPUs	Memory (GiB)	SSD (GiB)	GPU partition	GPU memory (GiB)	Maximum data disks
Standard_NV6ads_A10_v5	6	55	180	1/6	4	4
Standard_NV12ads_A10_v5	12	110	360	1/3	8	4
Standard_NV18ads_A10_v5	18	220	720	1/2	12	8
Standard_NV36ads_A10_v5	36	440	720	1	24	16
Standard_NV72ads_A10_v5	72	880	1400	2	48	32

Maya installation

Before you install Maya, you need to deploy and connect to a VM via Remote Desktop Protocol (RDP) and install the required NVIDIA drivers.

Important

An NVIDIA Fabric Manager installation is required for VMs that use NV Link or NV Switch.

You can install Maya from the [Autodesk Maya portal](#). For a detailed installation procedure, see the [Autodesk Maya help documentation](#).

Maya performance results

This performance analysis uses the Autodesk Maya 2023.1 trial version on Windows [NVadsA10_v5-series](#) VMs.

The following table provides details about the testing operating system:

[Expand table](#)

Version	Operating system	Architecture	Processor
Maya 2023.1	Windows 10 Professional, 20H2	x86-64	AMD EPYC 74F3V (Milan)
Maya 2024	Windows 11 Professional, 22H2	x86-64	AMD EPYC 74F3V (Milan)

The tests use the following model:



[Expand table](#)

Model name	Image size	Image size in inches
Open scene	HD 1080	26.7 x 15.0

The following sections provide information about four metrics that measure the performance of Maya on an Azure VM.

Measurement 1: Open scene performance

The following table lists times for reading the model file with various GPU configurations. The time that it takes to load a test scene is the time that it takes to read a file.

Results on Maya 2023.1

[Expand table](#)

Number of GPUs	File read time (seconds)
1/3	15
1/2	13.3
1	12.9

Results on Maya 2024

[Expand table](#)

Number of GPUs	File read time (seconds)
1/3	13.1
1/2	15.4
1	12.1

On tests with 1/6 GPUs, Maya reports an error about not enough free memory remaining in the GPU.

The following image shows the loading view, which is the view that the independent software vendor (ISV) expects to see:



Measurement 2: Playback performance

The following table lists playback times and rates for various GPU configurations when a MEL script is used:

Results on Maya 2023.1

[Expand table](#)

Number of GPUs	Playback rate (frames/second)	Total playback time (seconds)
1/3	23.89	17.79
1/2	24.02	17.69
1	24.02	17.69

Results on Maya 2024

[Expand table](#)

Number of GPUs	Playback rate (frames/second)	Total playback time (seconds)
1/3	27.74	15.32
1/2	28.46	14.93
1	43.41	9.79

The model contains 425 frames. The playback time and rate are almost identical for all tested GPU configurations.

Measurement 3: Arnold renderer performance

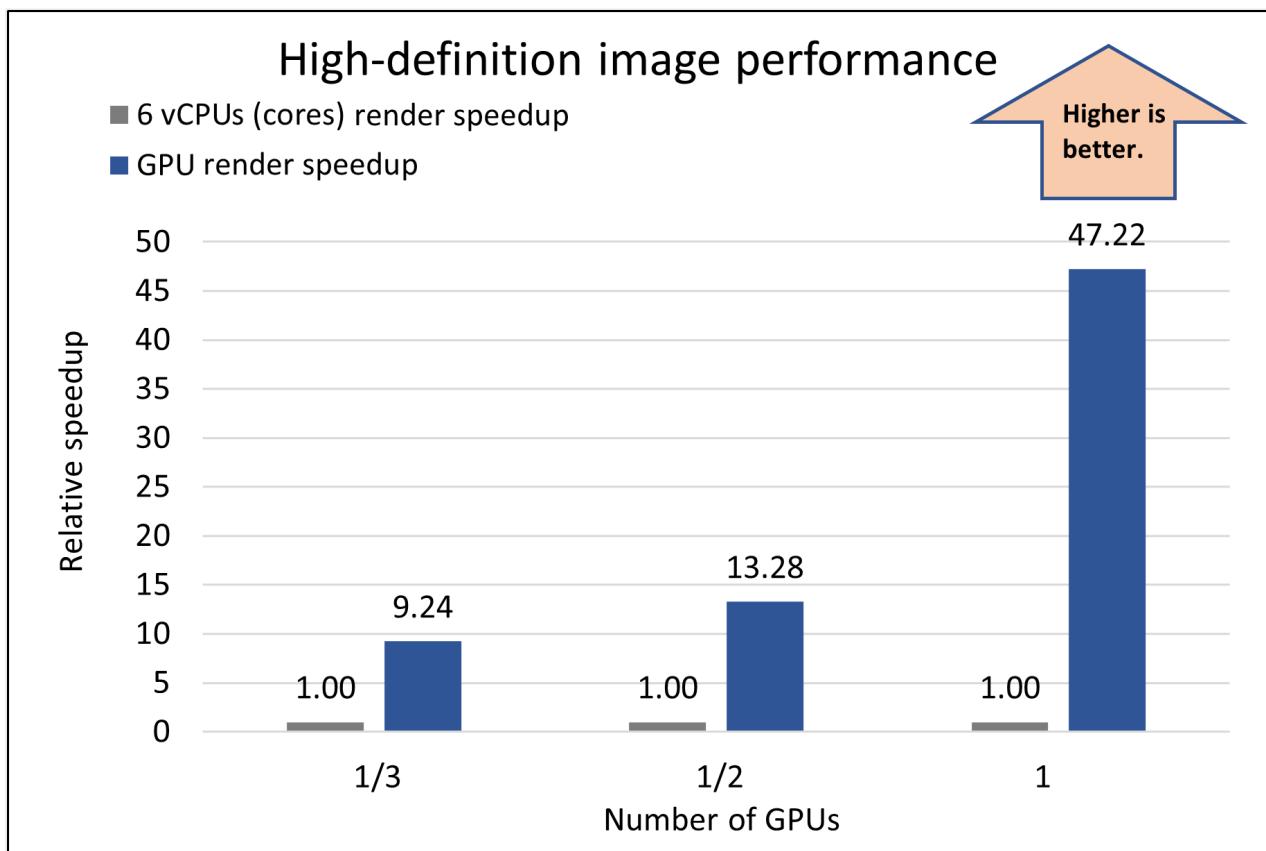
The following table shows the rendering time for various GPU configurations. Results from tests with six vCPUs (cores) are used as the baseline for determining speedup.

Results on Maya 2023.1

[Expand table](#)

Number of GPUs	Rendering time (seconds)	Relative speedup
6 vCPUs (cores)	1,275	NA
1/3	138	9.24
1/2	96	13.28
1	27	47.22

The following chart shows the relative speed increases for the model as the number of GPUs increases:



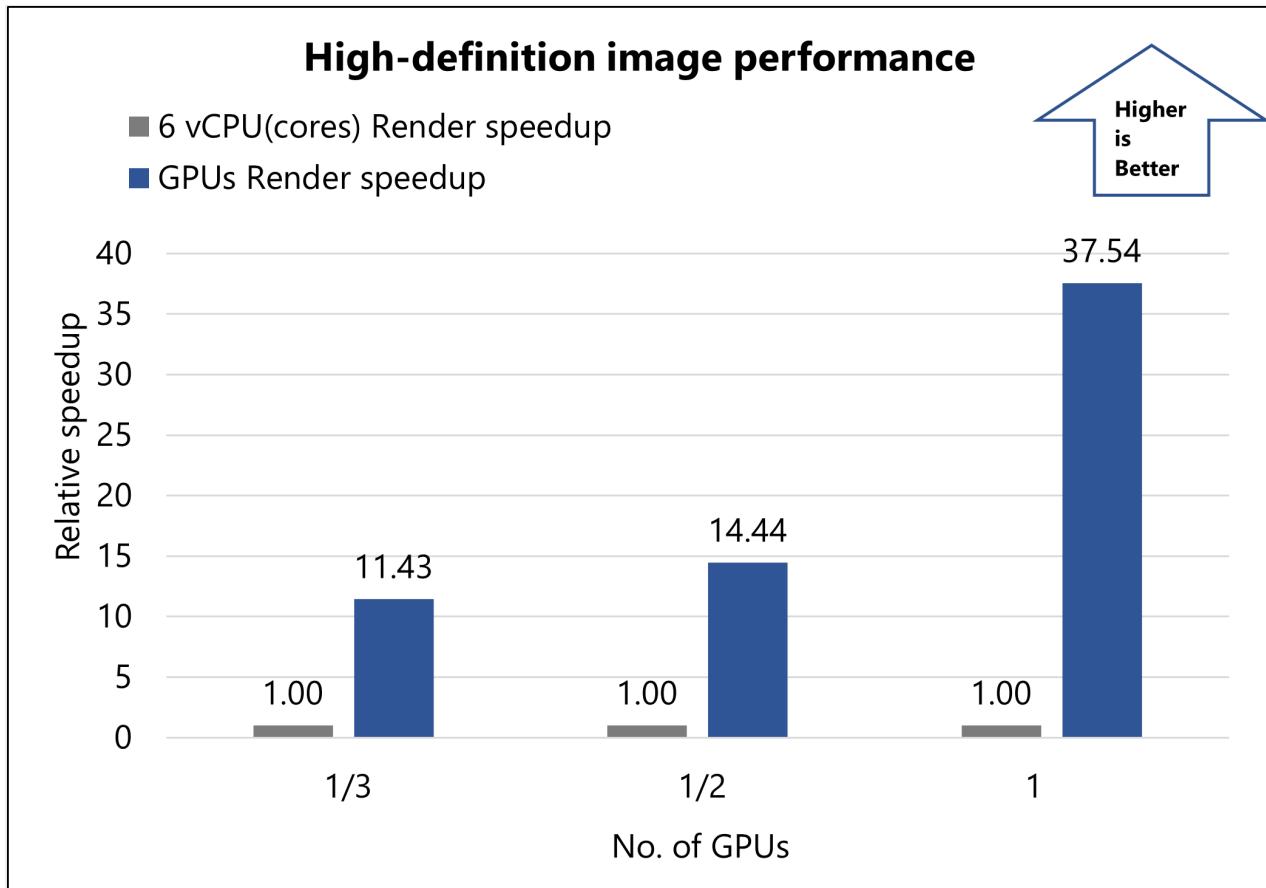
Results on Maya 2024

[Expand table](#)

Number of GPUs	Rendering time (seconds)	Relative speedup
6 vCPUs (cores)	1,314	NA
1/3	115	11.43

Number of GPUs	Rendering time (seconds)	Relative speedup
1/2	91	14.44
1	35	37.54

The following chart shows the relative speed increases for the model as the number of GPUs increases:



The following image shows the rendered output:



Measurement 4: Caching performance

The following table lists performance data for caching via a Python script. When you use the Cached Playback feature, you can view changes that you make to the animation without having to create a new playblast.

Results on Maya 2023.1

[Expand table](#)

Number of GPUs	Filling playback (frames/second)	Cached playback (frames/second)	Fill time (seconds)
1/3	6.19	21.31	0.88
1/2	17.09	24.11	0.67
1	18.86	24.11	0.60

Results on Maya 2024

[Expand table](#)

Number of GPUs	Filling playback (frames/second)	Cached playback (frames/second)	Fill time (seconds)
1/3	14.38	45.72	0.91
1/2	14.70	49.40	0.75
1	18.34	52.80	0.62

Additional notes about tests

- Tests on the Maya application run successfully on NVadsA10_v5 VMs on the Azure cloud platform.
- In tests of the rendering process on an NVadsA10_v5 VM, Maya scales well from a partial to a full GPU configuration.
- Tests of the caching performance show that the fill time decreases as the configuration improves.

Azure cost

The following table lists the elapsed times in hours for running the model with various GPU configurations. To compute the total cost, multiply these times by the Azure VM hourly cost for an NVadsA10_v5 VM. For the current hourly cost, see [Windows Virtual Machines Pricing](#) and [Linux Virtual Machines Pricing](#).

Maya 2023.1

[Expand table](#)

Number of GPUs or CPUs	Elapsed time (hours)
1/3 GPUs	0.038
1/2 GPUs	0.027
1 GPU	0.008
6 vCPUs	0.354

Number of GPUs or CPUs	Elapsed time (hours)
1/3 GPUs	0.031
1/2 GPUs	0.025
1 GPU	0.009
6 vCPUs	0.365

For these cost calculations, only the model rendering time is considered, measured in wall-clock time. The application installation time isn't considered. The calculations are indicative of your potential results, but actual values depend on the size of your model.

To estimate the cost of your configuration, use the [Azure pricing calculator](#).

Summary

- You can successfully deploy and run Maya on NVadsA10_v5-series VMs.
- In tests of the rendering process on an NVadsA10_v5 VM, Maya scales well from a partial to a full GPU configuration.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see nonpublic LinkedIn profiles, sign in to LinkedIn.

Next steps

- GPU-optimized virtual machine sizes
- Virtual machines in Azure
- Virtual networks and virtual machines in Azure
- Training path: Run high-performance computing (HPC) applications on Azure

Related resources

- Run a Windows VM on Azure
- Run a Linux VM on Azure
- HPC system and big-compute solutions
- HPC cluster deployed in the cloud

Deploy Revit on an Azure virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for installing and running Autodesk Revit on a virtual machine (VM) on Azure. It also presents the performance results of running Revit on Azure.

Revit helps architecture, engineering, and construction (AEC) teams create high-quality buildings and infrastructure.

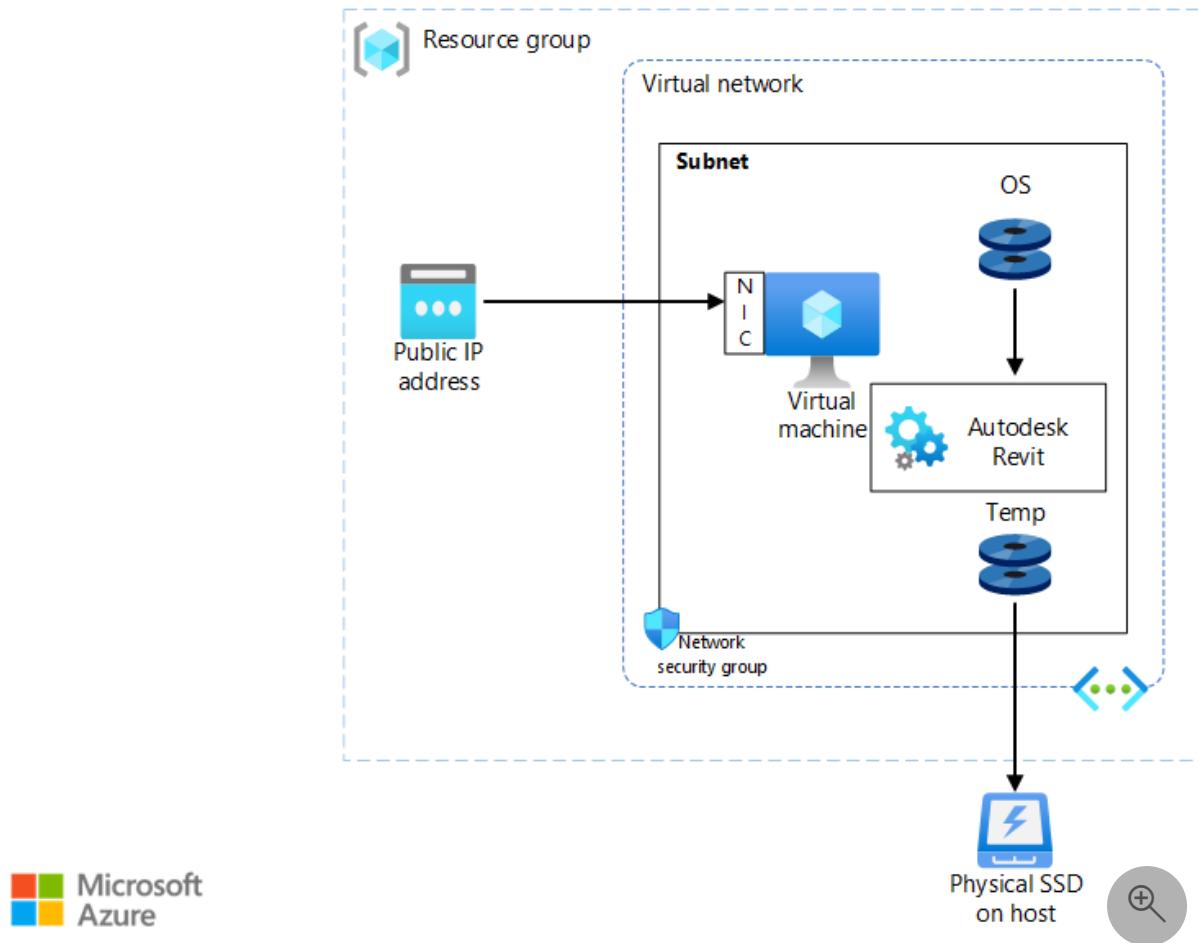
Engineers use Revit to model shapes, structures, and systems in 3D with parametric accuracy and precision and to streamline documentation work.

Revit has built-in automation for documenting design and managing deliverables. It saves, syncs, and shares model-based BIM and CAD data to connect multidisciplinary teams and workflows.

Why deploy Revit on Azure?

- Modern and diverse compute options to align to your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Strong GPU acceleration, with increased performance as GPUs are added

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create Windows VMs and run Windows. For information about deploying the VM and installing the drivers, see [Windows VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network in the cloud.
- [Network security groups](#) restrict access to VMs at the subnet level.
- A public IP address allows users to access Revit via the internet.
- A physical solid-state drive (SSD) provides storage.

Deploy infrastructure and install Revit

Deploy Azure VMs. Before you install Revit, deploy your Azure VMs. You should use a [NVadsA10_v5](#) or [NCasT4_v3](#) series VM to run Revit. You should use a Premium SSD managed disk and attach it to the VM.

Create and configure the supporting infrastructure. You need to configure a public IP address for inbound connectivity and use network security groups to provide security for the subnet.

Install NVIDIA drivers. You need to install [NVIDIA GPU drivers](#) to take advantage of the GPU capabilities of NVadsA10_v5 and NCasT4_v3 series VMs. For information about deploying VMs and installing the drivers, see [Run a Windows VM on Azure](#).

Download and install Revit. After you install the NVIDIA drivers, install Revit. To install the product, sign in to your [Autodesk](#) account. Select **Revit** under **Products**. For more information, see the Autodesk support website.

Revit performance on Azure Virtual Machines

HPC workloads require significant compute, memory, and storage resources.

Understanding the performance of different VM types with the Revit application can help you select the most appropriate VM for your workload and optimize performance and cost.

We ran six test scenarios, via scripts, for Revit. The tests were run on a trial version of Revit 2022 on [NVadsA10_v5](#) and [NCasT4_v3](#) series Azure VMs. The results of these performance tests are presented later in this document to help you determine the right hardware for your Azure deployment.

Model details

The [RFO Benchmark](#) automatic test suite is used to measure the performance of Revit on Azure Virtual Machines. Some prebuilt test scenarios are available in the Benchmark script. We used six of these scenarios to analyze performance:

- Graphics Acceleration
- Full Expanded
- Full Simplified
- Full Standard
- Graphics Comparison
- Graphics Expanded

Results on NVadsA10_v5

The following table shows the elapsed times, in seconds, for running the test sets on four NVadsA10_v5 VM configurations.

 [Expand table](#)

RFO Benchmark test name	6 vCPUs (1/6th GPU)	18 vCPUs (1/2 GPU)	36 vCPUs (1 GPU)	72 vCPUs (2 GPUs)
Graphics Acceleration	3,847.19	3,366.57	3,472.25	3,432.98
Full Expanded	13,552.75	12,539.80	11,590.06	11,567.43
Full Simplified	197.70	174.49	140.31	137.95
Full Standard	784.41	595.08	574.08	536.98
Graphics Comparison	205.31	100.58	83.22	78.82
Graphics Expanded	2,824.53	1,259.49	921.05	1,000.70

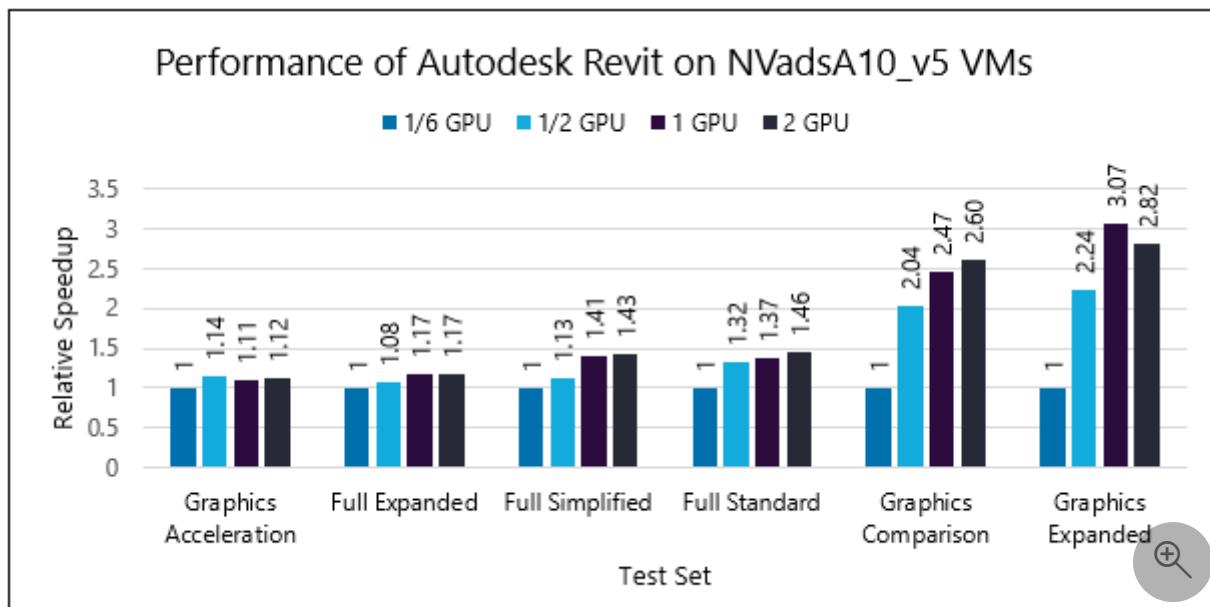
The following table shows the relative speed increases as the number of vCPUs increases. The elapsed time for 1/6th GPU is used as a baseline.

[Expand table](#)

RFO Benchmark test name	6 vCPUs (1/6th GPU)	18 vCPUs (1/2 GPU)	36 vCPUs (1 GPU)	72 vCPUs (2 GPUs)
Graphics Acceleration	1	1.14	1.11	1.12
Full Expanded	1	1.08	1.17	1.17
Full Simplified	1	1.13	1.41	1.43
Full Standard	1	1.32	1.37	1.46
Graphics Comparison	1	2.04	2.47	2.60

RFO Benchmark test name	6 vCPUs (1/6th GPU)	18 vCPUs (1/2 GPU)	36 vCPUs (1 GPU)	72 vCPUs (2 GPUs)
Graphics Expanded	1	2.24	3.07	2.82

This graph shows the relative speed increases for the six test cases. A high relative speed increase is better than a low one.



Results on NCasT4_v3

The following table shows the elapsed times, in seconds, for running the test sets on two NCasT4_v3 VM configurations.

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

RFO Benchmark test name	4 vCPUs (1 GPU)	64 vCPUs (4 GPUs)
Full Simplified	184.47	193.91
Full Expanded	864.42	741.36
Full Standard	17,353.97	16,794.71

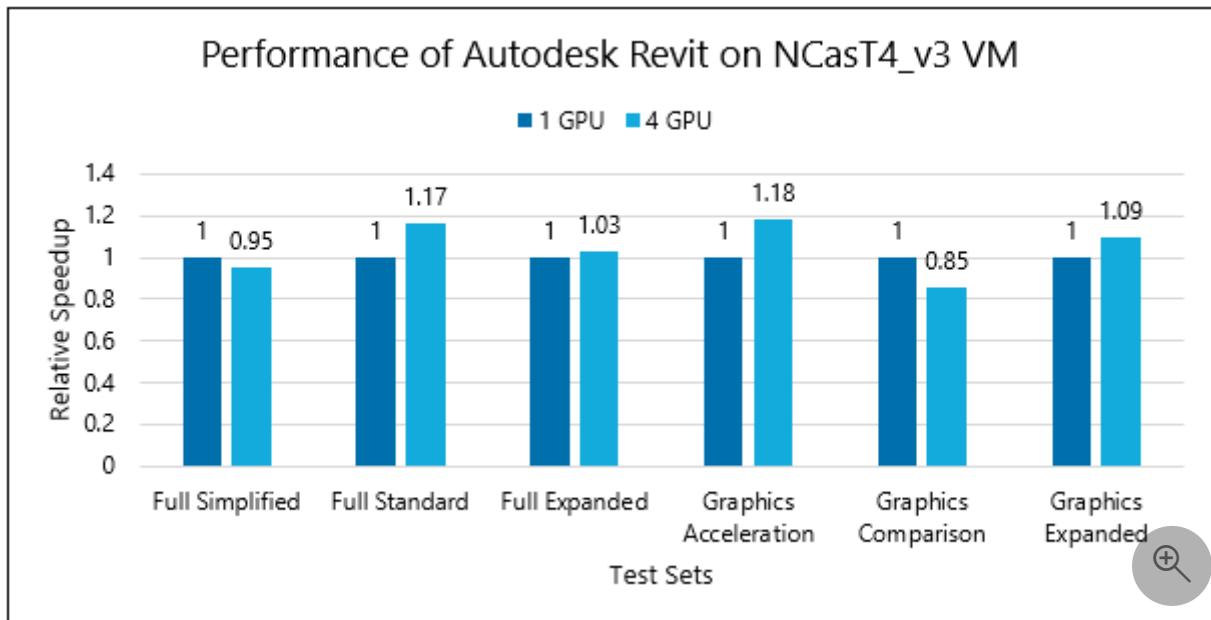
RFO Benchmark test name	4 vCPUs (1 GPU)	64 vCPUs (4 GPUs)
Graphics Acceleration	5,534.41	4691.55
Graphics Comparison	114.25	134.06
Graphics Expanded	1,783.71	1,632.31

The following table shows the relative speed increases for the six test sets, as the number of vCPUs increases.

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

RFO Benchmark test name	4 vCPUs (1 GPU)	64 vCPUs (4 GPUs)
Full Simplified	1	0.95
Full Expanded	1	1.17
Full Standard	1	1.03
Graphics Acceleration	1	1.18
Graphics Comparison	1	0.85
Graphics Expanded	1	1.09

This graph shows the relative speed increases for the six test cases. A high relative speed increase is better than a low one.



Azure cost

You can use the following data to calculate the cost of running your workload. To compute the cost, multiply the total elapsed time by the hourly cost for the VM. Because the hourly rates of VMs can change, you should use the [Windows Virtual Machines Pricing](#) calculator to compute the cost. The total elapsed time doesn't include application installation. It includes only the total time for completing the test scenarios for all models.

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

VM series	Number of vCPUs	Number of GPUs	Total elapsed time, in hours
NVadsA10_v5	6	1/6	5.95
	18	1/2	5.01
	36	1	4.66
	72	2	4.65
NCasT4_v3	4	1	7.18

VM series	Number of vCPUs	Number of GPUs	Total elapsed time, in hours
	64	4	6.72

Summary

We deployed and tested Revit on Azure NVadsA10_v5 and NCasT4_v3 series VMs.

- On NVadsA10_v5 VMs, most VM configuration upgrades result in speed increases. The relative speed increases until one GPU is reached. There's a saturation in performance with further increases in GPUs.
- On NCasT4_v3 VMs, for four of the six test scenarios, the only performance difference occurs when the number of GPUs is increased from one to four. We recommend that you use a NCasT4_v3 VM with one GPU.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Amol Rane](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director, Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Windows virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance HPC applications on Azure](#)

Related resources

- [Run a Windows VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Autodesk VRED for HPC on Azure

Article • 02/07/2023

Autodesk VRED is a 3D visualization application that helps automotive designers and engineers create product presentations, design reviews, and virtual prototypes by using interactive CPU and GPU ray tracing. VRED, which was previously limited to CPU, now uses GPU technology to support the high demands of consumers and provide interactive ray tracing and AI-powered denoising.

By using VRED, users can create digital prototypes to gain insight into how vehicles will look and perform. To be effective in guiding design decisions, the digital prototypes need to look and behave as close as possible to the real vehicles. This solution is ideal for the automotive and manufacturing industries.

This article briefly describes the steps for running VRED on a virtual machine (VM) that's deployed on Azure. It also provides performance results. For more information about VRED, see the [Autodesk website](#).

VRED 2022.1 was successfully deployed and tested on [NC64as_T4_v3](#) and [NV48s_v3](#) Azure VMs. VRED 2023.1 was deployed and tested on [NC64as_T4_v3](#) and [NVadsA10 v5](#) VMs.

Install VRED on a VM

Before you install VRED, you need to deploy and connect a VM and install the required NVIDIA and AMD drivers.

Important

NVIDIA Fabric Manager installation is required for VMs that use NVLink or NVSwitch.

For information about deploying the VM and installing the drivers, see one of these articles:

- [Run a Windows VM on Azure](#)
- [Run a Linux VM on Azure](#)

To download VRED:

1. Sign in to your Autodesk account.
2. Search for VRED in **Products and Services**.
3. Install VRED Professional.

Install License Manager

Before you install VRED on an Azure VM, you need to install Autodesk Network License Manager on the VM. You can [install Network License Manager for Windows here](#).

During installation, this folder is created: C:\Autodesk\Network License Manager.

After installation, generate a license file from your Autodesk account and save it in the Network License Manager folder. Create a text file named *debug.log* and save it in the same folder.

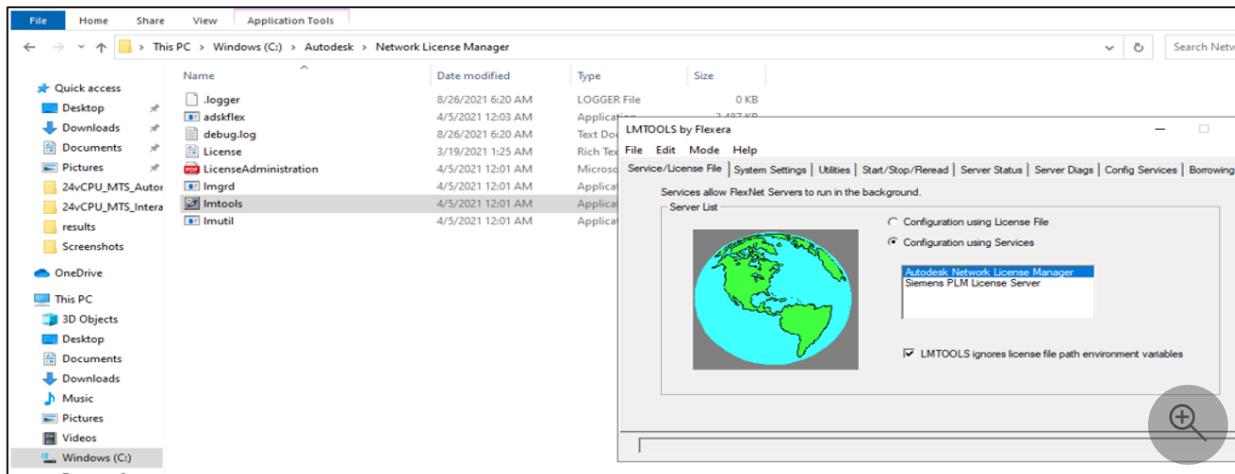
To generate the license file:

1. While signed in to your Autodesk account, select **VRED Professional downloads**.
2. Select **Generate network license file**.
3. Provide the server or VM name and the physical or MAC address.
4. Select the product.
5. Select **Get License File**. The license file is generated.

To configure the license server on Windows:

1. Open Network License Manager by typing **lmtools** in the Windows search bar and selecting it in the results. A GUI opens.
2. Select **Config Services** and provide the service name, in this case, **Autodesk Network License Manager**.
3. Provide the path of the license file and the other requested paths.
4. Select **Start Server at Power Up** at the bottom of the window.
5. Select **Save Service** and follow the prompts that appear.
6. On the **Start/Stop/Reread** tab, select **Start Server**. You should see the service name **Autodesk Network License Manager** highlighted in blue.

The Network License Manager installation is complete.



ⓘ Note

For Linux configuration instructions, see [Configure and start your license server](#).

VRED performance on Azure VMs

Rendering time is an important parameter for visualization and design software. Designers often spend a lot of time on the rendering process. By incorporating advanced capabilities like CPU and GPU ray tracing, VRED drastically reduces rendering times. To perform these complex rendering simulations on VRED, you need to use the right hardware. Microsoft partners with Nvidia to provide suitable infrastructure and hardware on Azure. Azure provides the fastest compute capabilities for both CPU-intensive and GPU-intensive workloads.

Rendering

The term *rendering* refers to the automatic process of generating digital images from three-dimensional models by using specialized software. The images simulate a 3D model's photorealistic environments, materials, lighting, and objects.

Real-time rendering is mainly used in gaming and interactive graphics, where images are calculated from 3D information at a fast pace. Dedicated graphics hardware has improved the performance of real-time rendering to ensure rapid image processing.

Offline rendering is used when less processing speed is required. Visual effects provide the highest standards of photorealism. In contrast to real-time rendering, there's no unpredictability with offline rendering.

Ray tracing is a rendering technique that can produce highly realistic lighting effects. Ray tracing generates lifelike shadows and reflections and much-improved translucence

and scattering, taking into account light phenomena like reflection and refraction. In VRED, there are two primary ray tracing options: CPU ray tracing and GPU ray tracing.

VRED application settings for rendering

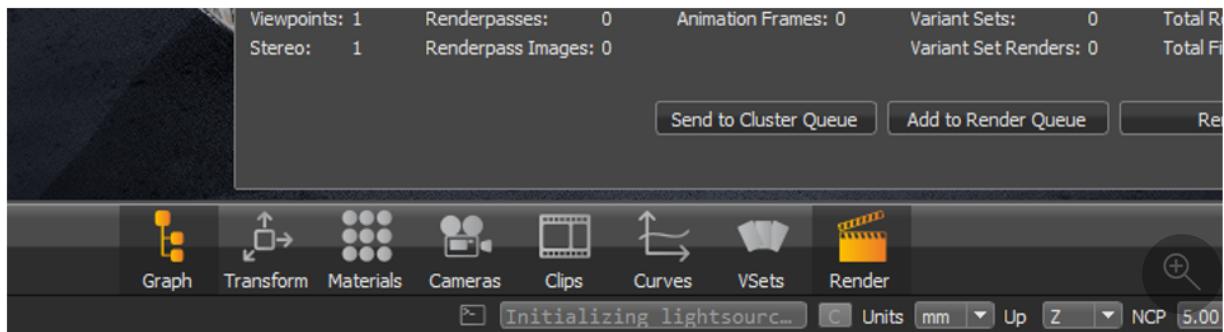
You can activate CPU and GPU ray tracing in VRED according to your requirements. To activate CPU/GPU ray tracing, select **Visualization > Raytracing > CPU/GPU Raytracing**.

Anti-aliasing settings

For CPU and GPU ray tracing rendering in these tests, we set the anti-aliasing option to high: **Visualization > Realtime Antialiasing > High**

Rendering settings

Select the rendering settings as shown here:

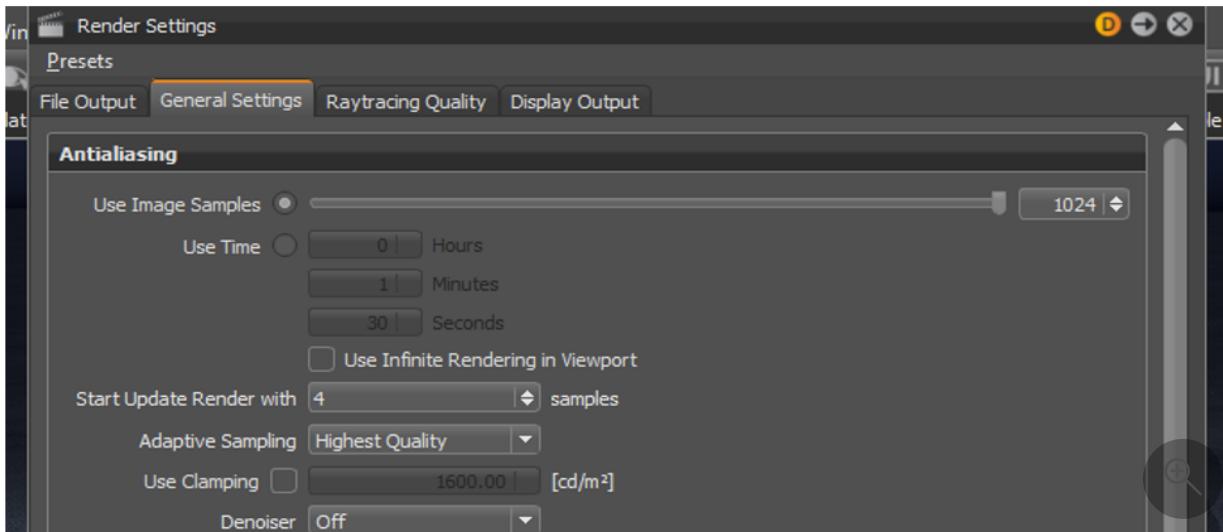


Saving images

You can save a rendered image to the desired location by selecting the path on the **File Output** tab in the **Render Settings** window. You can also choose an image size, like **HD** or **4K**, under **Image Size Presets**. We used **HD**.

General Settings

For CPU and GPU ray tracing, you need to select image samples for anti-aliasing. Your anti-aliasing output improves as you increase the number of samples. Under **General Settings**, we selected the maximum number: **1024**. For OpenGL, you can use a lower number, between 16 and 32 images, for example.



Raytracing Quality

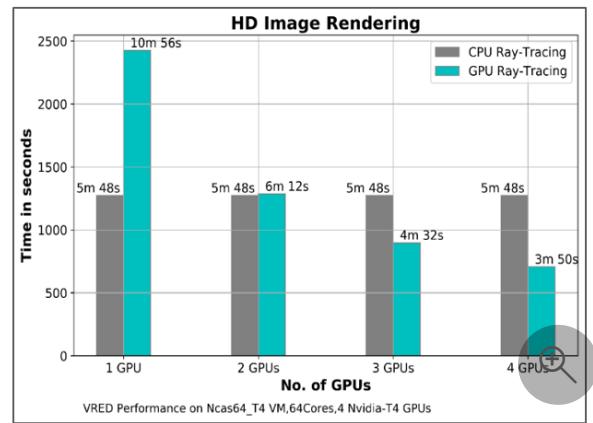
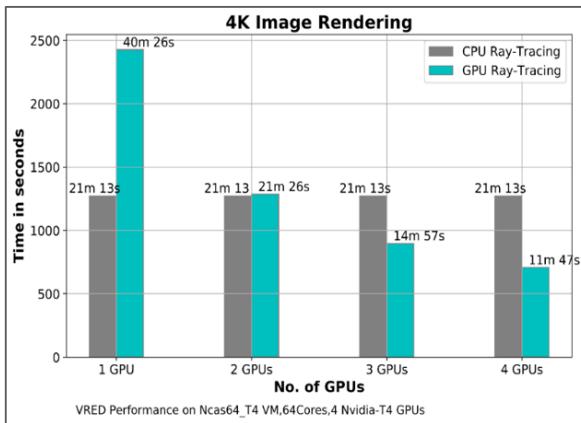
In the **Raytracing Quality** settings, for the **Illumination Mode** for both interactive and still frame, we used **Full Global Illumination**.

Benchmarking methodology for VRED performance analysis on VMs

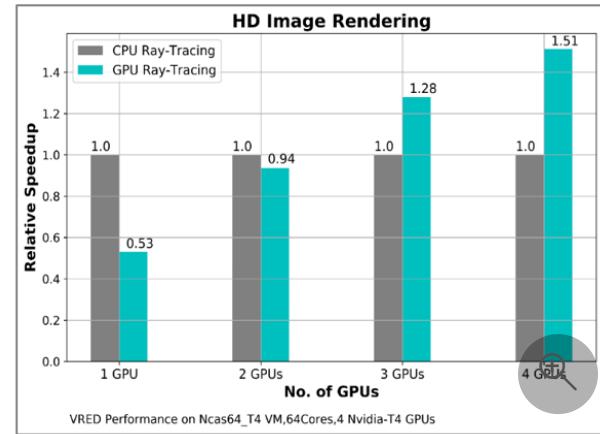
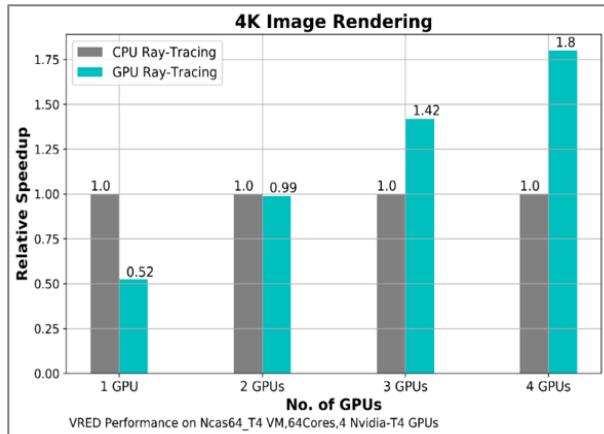
To analyze the performance of VRED on [NC64as_T4_v3](#) and [NV48s_v3](#) VMs, we tested offline image rendering and calculated the rendering times for both CPU ray tracing and GPU ray tracing. For this analysis, we rendered 4k and HD images. We tested GPU ray tracing on NC64as_T4_v3 and NV48s_v3 VMs by using 1, 2, 3, and 4 GPUs and on NVadsA10_v5 by using 1 and 2 GPUs. For CPU ray rendering, the application uses all CPU cores on the VM. We then calculated the relative speed increase of GPU rendering as compared to CPU rendering. The results are presented in the following sections.

VRED 2022.1 performance results on the NC64as_T4_v3 VM

CPU and GPU rendering times

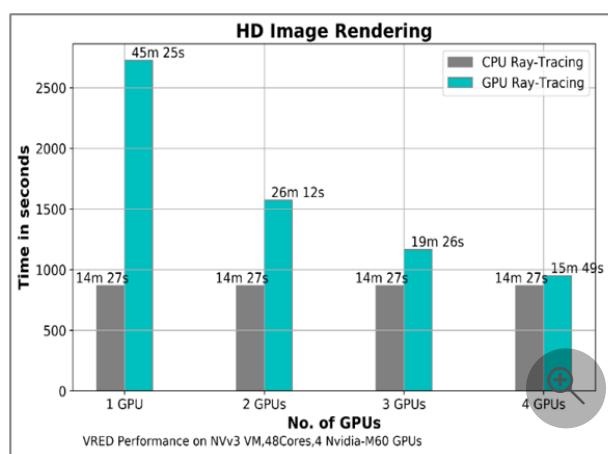
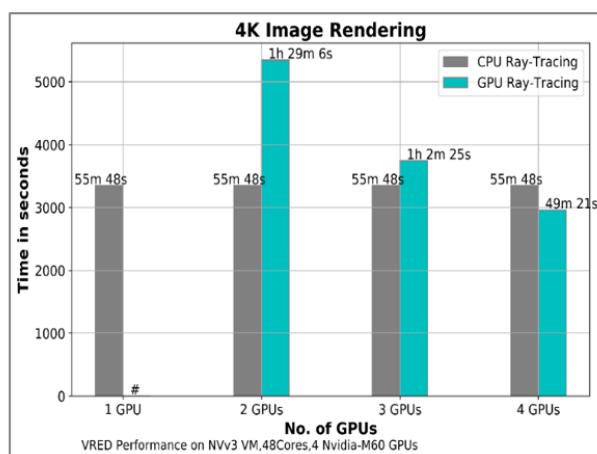


Relative speed increases between CPU and GPU ray tracing



VRED 2022.1 performance results on the NV48s_v3 VM

CPU and GPU rendering times

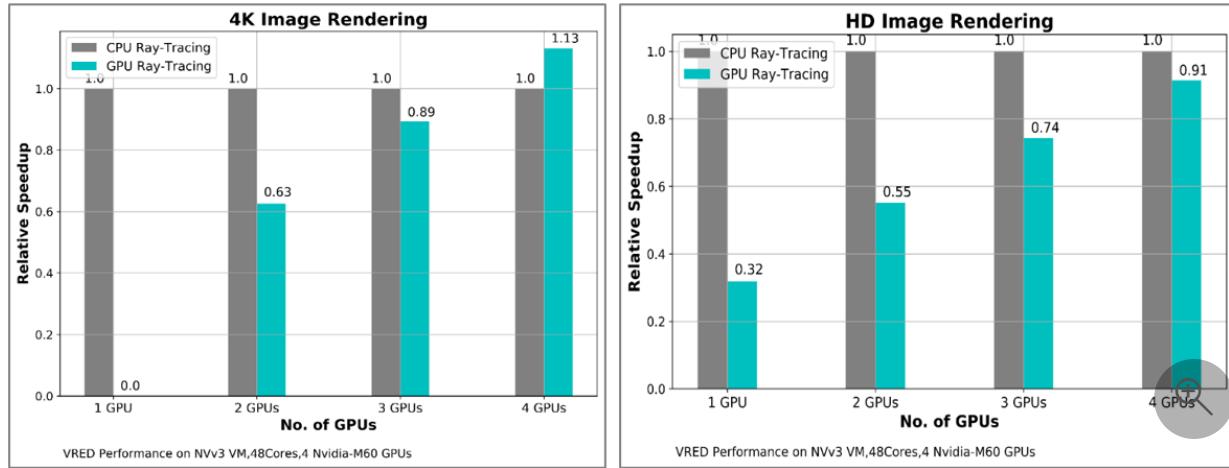


① Note

During GPU rendering on a NV48s_v3 VM with 1 GPU, the application produced an error when rendering with 4K and higher resolution. HD image rendering works

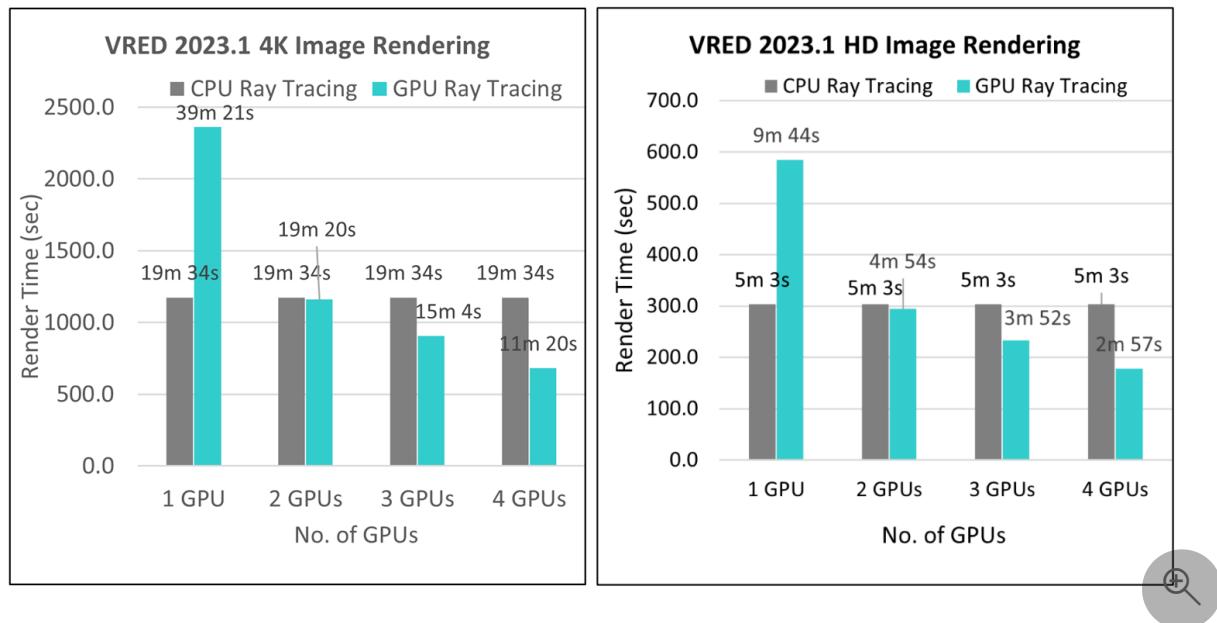
fine. The results depend on the model's complexity and environment settings. For large-scale models, we don't recommend the 1-GPU setting.

Relative speed increases between CPU and GPU ray tracing

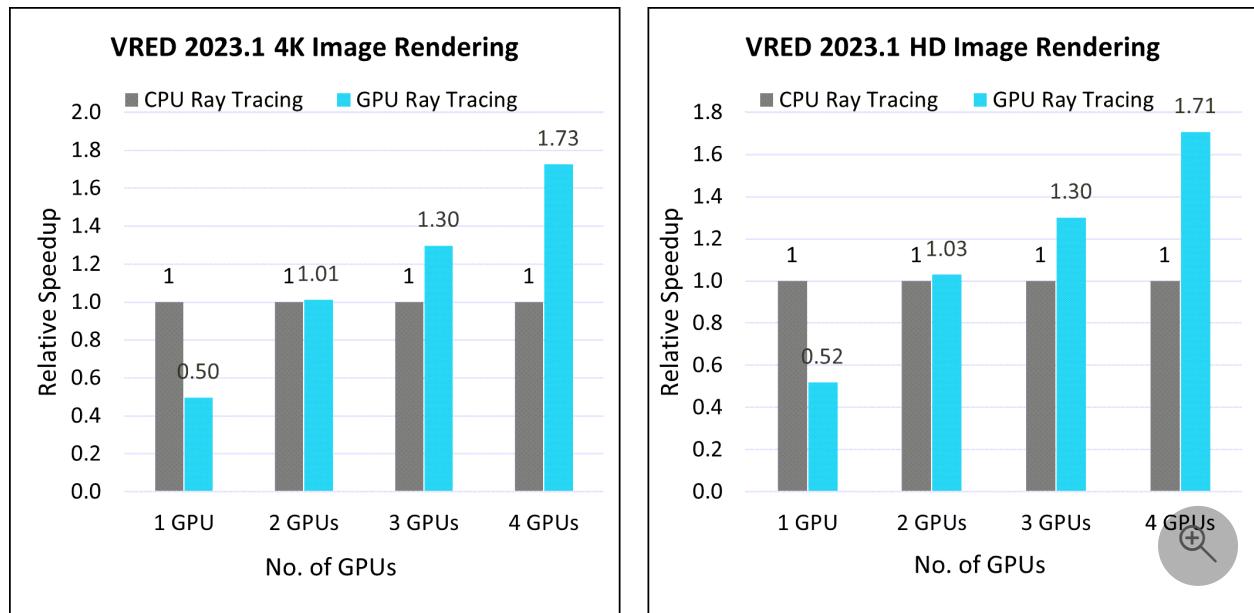


VRED 2023.1 performance results on the NC64as_T4_v3 VM

CPU and GPU rendering times

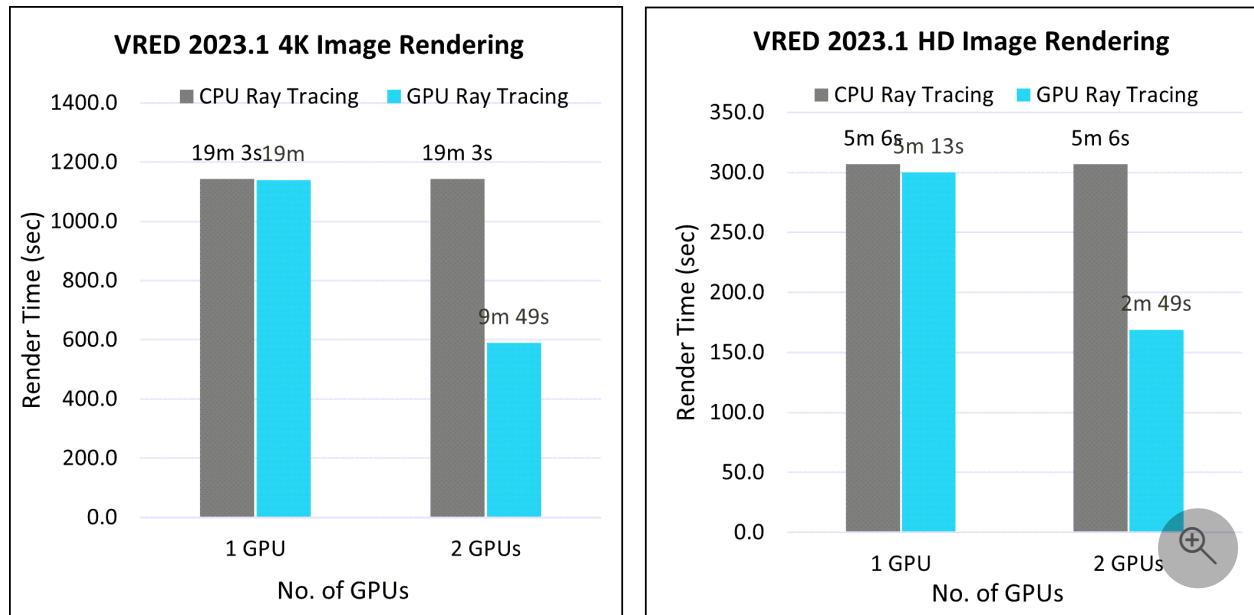


Relative speed increases between CPU and GPU ray tracing

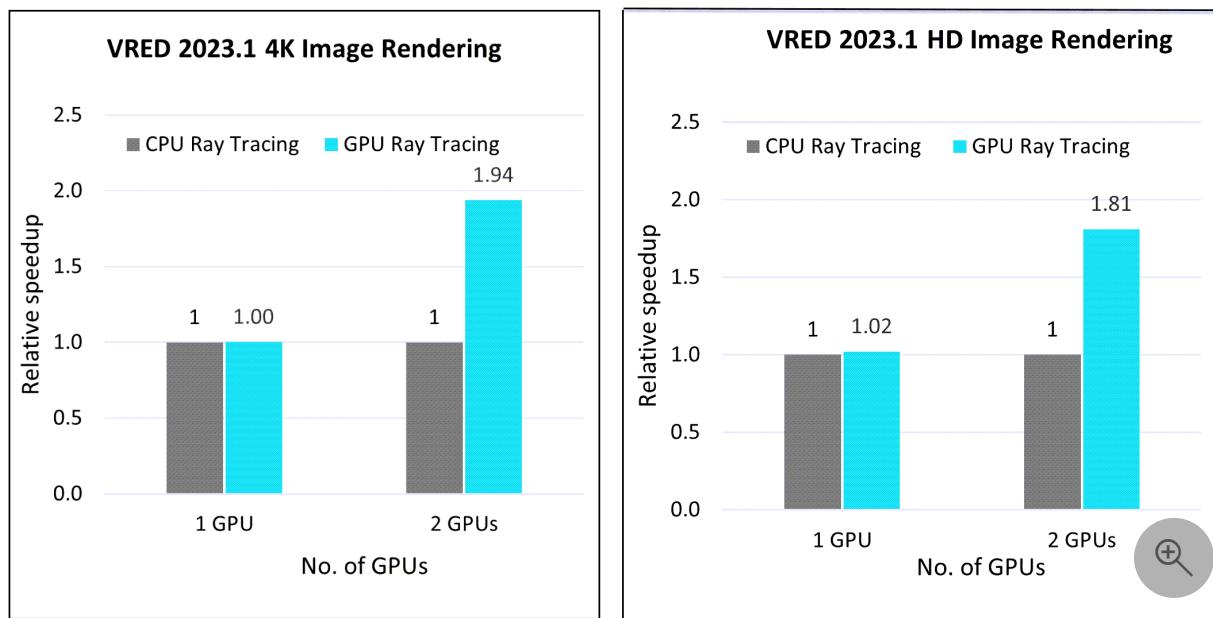


VRED 2023.1 performance results on the NVadsA10_v5 VM

CPU and GPU rendering times



Relative speed increases between CPU and GPU ray tracing



Pricing

Only model running time (wall clock time) is considered for these cost calculations. Application installation time isn't considered. The calculations are indicative. The actual numbers depend on the size of the model.

You can use the [Azure pricing calculator](#) to estimate costs for your configuration.

You can use the rendering times provided in the following tables and the Azure hourly costs to compute rendering costs. For example, if the Azure VM hourly cost is \$8.60 and the rendering time is 11 minutes and 47 seconds, the cost is \$1.69. For current Azure hourly costs, see [Windows Virtual Machines Pricing](#) or [Linux Virtual Machines Pricing](#).

Azure cost for VRED 2022.1

GPU rendering costs

VM	Number of GPUs on VM	4K image render time	HD image render time
NC64as_T4_v3	4	11 minutes and 47 seconds	3 minutes and 48 seconds
NV48s_v3	4	49 minutes and 21 seconds	15 minutes and 49 seconds

CPU rendering costs

VM	Number of CPU cores	4K image render time	HD image render time
NC64as_T4_v3	64	21 minutes and 13 seconds	5 minutes and 48 seconds
NV48s_v3	48	55 minutes and 48 seconds	14 minutes and 27 seconds

Azure cost for VRED 2023.1

GPU rendering costs

VM	Number of GPUs on VM	4K image render time	HD image render time
NC64as_T4_v3	4	11 minutes and 20 seconds	2 minutes and 57 seconds
NVadsA10_v5	2	9 minutes and 49 seconds	2 minutes and 49 seconds

CPU rendering costs

VM	Number of CPU cores	4K image render time	HD image render time
NC64as_T4_v3	64	19 minutes and 3 seconds	5 minutes and 6 seconds
NVadsA10_v5	72	19 minutes and 34 seconds	5 minutes and 3 seconds

Results and recommendations

- VRED was successfully deployed and tested on NCas_T4_v3, NVv3, and NVadsA10_v5 series VMs on Azure.
- On NC64as_T4, GPU rendering is 1.8 times faster than CPU rendering.
- On NVadsA10_v5, GPU rendering is 1.94 times faster than CPU rendering.
- On NVv3, GPU rendering doesn't significantly improve rendering time as compared to CPU rendering.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Jason Bouska](#) | Senior Software Development Engineer
- [Guy Bursell](#) | Director, Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- GPU optimized virtual machine sizes
- Windows virtual machines on Azure
- Linux virtual machines on Azure
- Learning path: Run high-performance computing (HPC) applications on Azure
- Virtual networks and virtual machines in Azure
- [VRED Render Settings and Modes](#)

Related resources

- [Run a Windows VM on Azure](#)
- [Run a Linux VM on Azure](#)
- [Deploy ADS CFD Code Leo for HPC on a virtual machine](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy AVL FIRE M on an Azure virtual machine

Azure Virtual Machines Azure Virtual Network Azure CycleCloud

This article describes the steps for running the [AVL FIRE M](#) application on a virtual machine (VM) and a high-performance computing (HPC) cluster that's deployed on Azure. The article also shows the performance results of running AVL FIRE M on Azure CycleCloud (an Azure HPC cluster).

AVL FIRE M is the leading computational fluid dynamics (CFD) simulation package for internal-combustion engines. In the new era of electromobility, it has evolved into a comprehensive software tool offering solutions for a wide spectrum of applications. It simulates:

- Fluid flow and thermal load in power train and vehicle components.
- Vehicle aerodynamics.
- Complex multi-fluid and multiphase flows.
- Efficient and reliable solutions for electrification, such as virtual development and integration of electric driveline, battery, and fuel cells.

Designed to accurately simulate relevant physics and chemistry, AVL FIRE M enables predictive simulations of fuel sprays, ignition, combustion, and engine-out emissions. It also shows how to tailor components of exhaust gas aftertreatment systems. And it provides models for electrochemistry and thermal behavior of batteries and fuel cells.

AVL FIRE M provides:

- Efficient solutions to solve demanding flow problems in various applications and industries.
- Accurate simulation of heat transfer and thermal load problems.
- Qualified and task-oriented software support along with application method development.

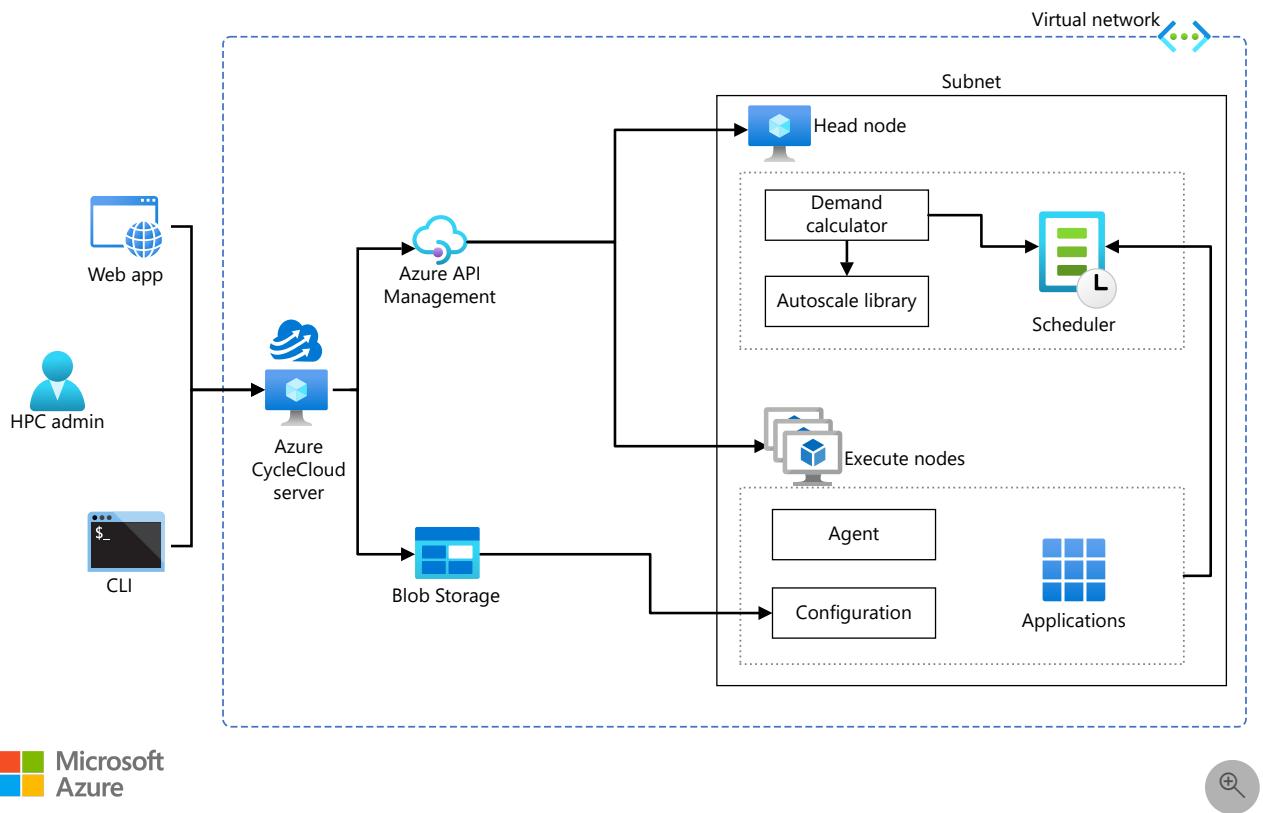
Why deploy AVL Fire M on Azure?

The following list describes the benefits of deploying AVL FIRE M on Azure:

- Modern and diverse compute options align with your workload's needs
- Flexibility for virtualization without the need to buy and maintain physical hardware
- Rapid provisioning

Architecture

This diagram shows a multi-node configuration:



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) creates Linux and Windows VMs in seconds.
- [Azure Virtual Network](#) creates a private network infrastructure in the cloud.
- [Azure CycleCloud](#) creates the cluster in the multi-node configuration.

Compute sizing and drivers

Performance tests of AVL FIRE M on Azure use [HBv3-series VMs](#) running the Linux CentOS operating system. The following table provides the configuration details:

[Expand table](#)

Size	vCPU	RAM memory (GiB)	Memory bandwidth (Gbps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120rs_v3	120	448	350	1.9	3.0	3.5	200	32
Standard_HB120-96rs_v3	96	448	350	1.9	3.0	3.5	200	32
Standard_HB120-64rs_v3	64	448	350	1.9	3.0	3.5	200	32
Standard_HB120-32rs_v3	32	448	350	1.9	3.0	3.5	200	32
Standard_HB120-32rs_v3	16	448	350	1.9	3.0	3.5	200	32

Required drivers

To use InfiniBand, you need to [enable InfiniBand drivers](#).

AVL FIRE M 2022 R1 installation

Before you install AVL FIRE M, you deploy and connect a VM or an Azure HPC cluster.

For more information about deploying the VM and installing the drivers, see either [Run a Windows VM on Azure](#) or [Run a Linux VM on Azure](#).

For more information on deploying the Azure CycleCloud and Azure HPC cluster, see these resources.

- [Install and configure Azure CycleCloud](#)
- [Create an Azure HPC cluster](#)

You can download AVL FIRE M products on the AVL self-service portal. Alternately, you can install the application without downloading the large installation image by running the installer executable (AVL SETUP.run). For more information on how to download, install, and license AVL FIRE M, see the [AVL Fire M website](#).

AVL FIRE M 2022 R1 performance results

For the performance results, AVL FIRE M ran steady state simulations. The following are details on the models used for AVL FIRE M performance validation on Virtual Machines.

DrivAer_BodyFitted: Open Cooling DrivAer Notchback is a realistic test case that's relevant to the automotive industry and accepted as the standard for automotive CFD correlation. For more information, see the [DrivAer_BodyFitted SAE technical paper](#) by Hupertz, B., Chalupa, K., Krueger, L., Howard, K. et al., *On the Aerodynamics of the Notchback Open Cooling DrivAer: A Detailed Investigation of Wind Tunnel Data for Improved Correlation and Reference*, (SAE Int. J. Adv. & Curr. Prac. in Mobility 3(4):1726-1747,- 2021).

DrivAer_EMBEDDEDBody: Embedded body approach in AVL FIRE M, requires practically no conventional meshing and is an alternative to the standard (body-fitted) for the comparative assessment. For more information, see the [DrivAer_EMBEDDEDBody SAE technical paper](#) by Basara, B., Zunic, Z., Pavlovic, Z., Sampl, P. et al., *Performance Analysis of Immersed Boundary Method for Predicting External Car Aerodynamics* (SAE Technical Paper 2022-01-0889, 2022).

Note

Analysis steady state RANS simulation is run with decomposed mesh.

The following table provides details:

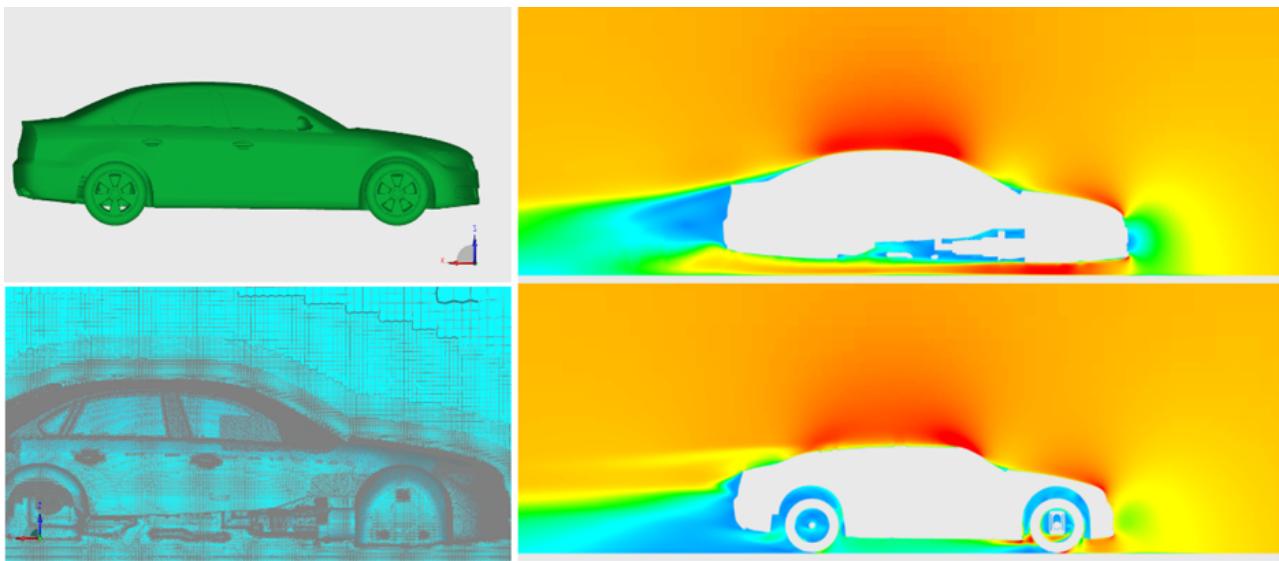
 [Expand table](#)

Model	Pressure correction equation	Pressure boundary values	Embedded bodies	Flow category
DrivAer_BodyFitted	Simple	Mirrored	No	Incompressible
DrivAer_EMBEDDEDBody	Simple	Extrapolated	Yes	Incompressible

The DrivAer_BodyFitted and DrivAer_EMBEDDEDBody models were used for this performance evaluation.

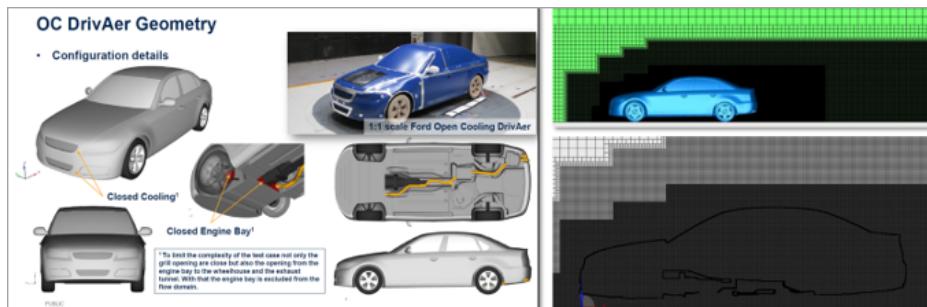
DrivAer_BodyFitted model results

The following figure shows the DrivAer_BodyFitted model results.



DrivAer_EMBEDDEDBody model results

The following figure shows the DrivAer_EMBEDDEDBody model results.



[Expand table](#)

Model	Internal cells	Run mode	Linear solver	Iterations
DrivAer_BodyFitted	128,103,525	Steady State	Pressure GSTB	300
DrivAer_EMBEDDEDBody	87,365,510	Steady State	Pressure GSTB	500

The steady state CFD analyses were performed on an Azure CycleCloud multi-node setup with [HBv3 AMD EPYC™ 7V73X](#) (Milan-X) VMs. The single node result is considered the baseline for comparing multi-node CPU runs.

Performance results for DrivAer_BodyFitted

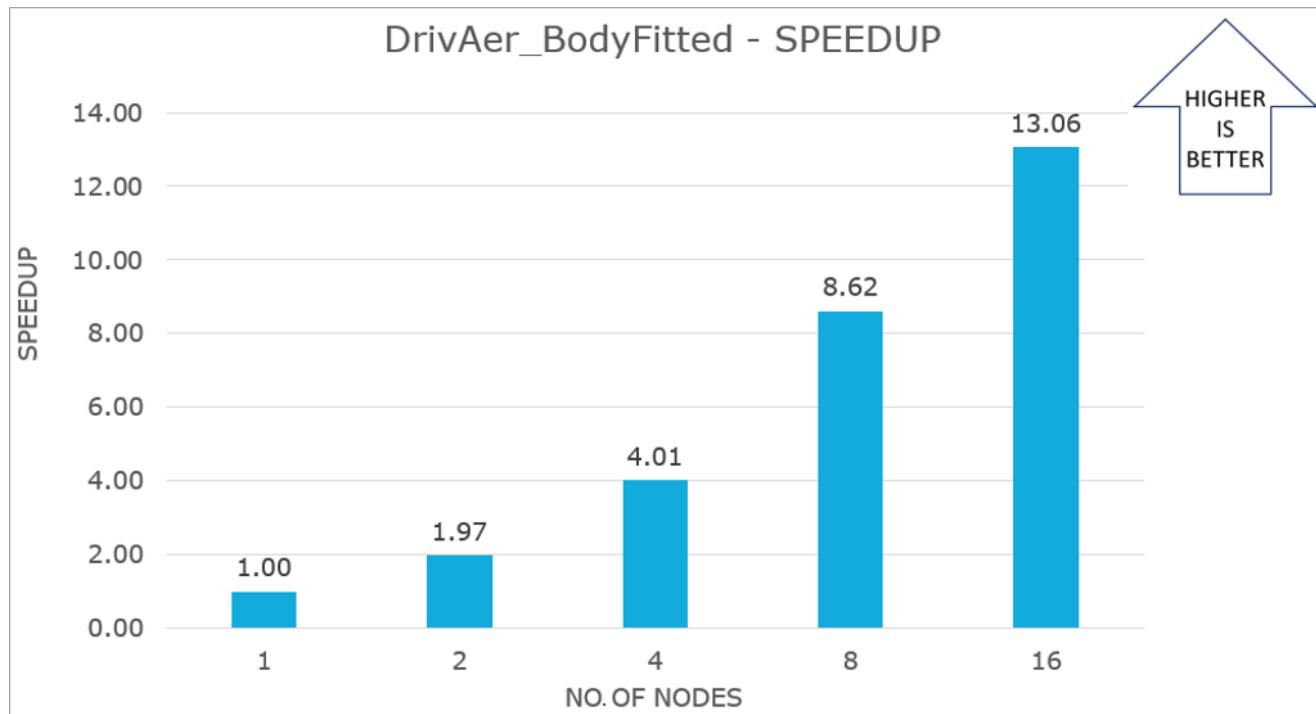
This table shows the speedup performance results for DrivAer_BodyFitted.

[Expand table](#)

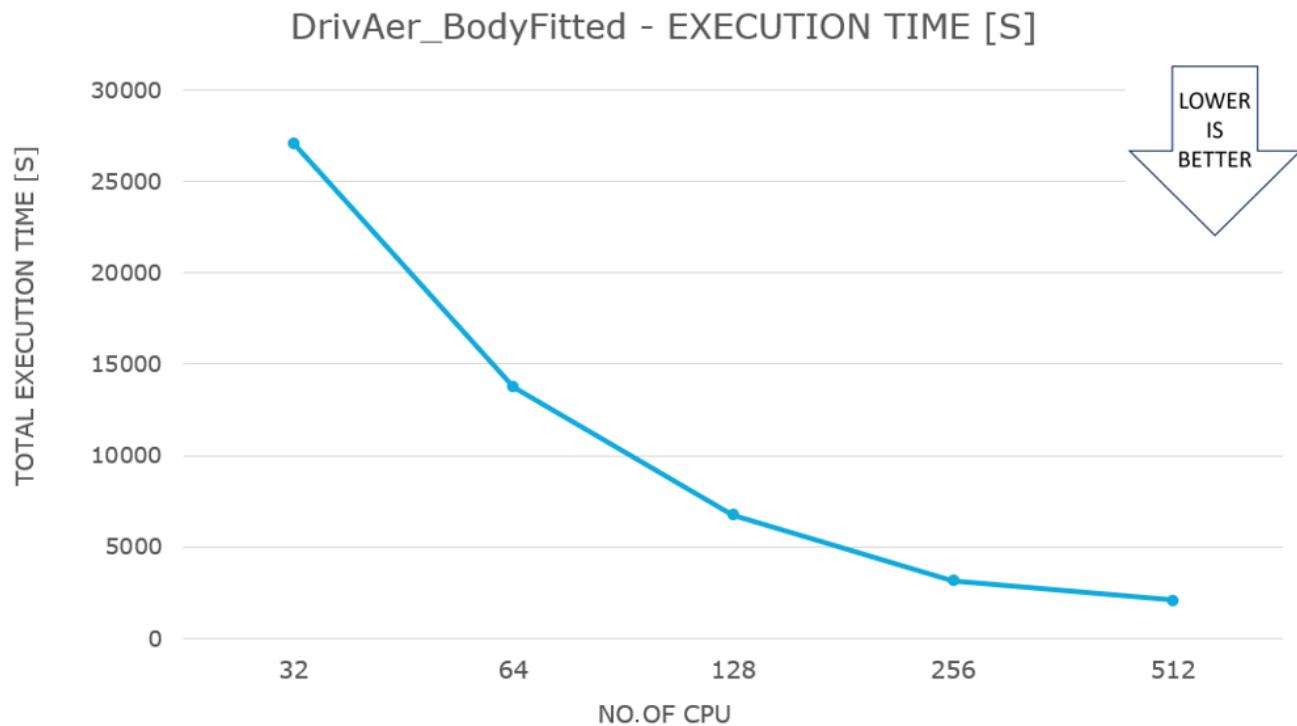
Number of nodes	Number of CPUs	Solver running time (seconds)	Relative speed increase
1	32	27,098	1.00
2	64	13,754	1.97
4	128	6,757	4.01
8	256	3,145	8.62

Number of nodes	Number of CPUs	Solver running time (seconds)	Relative speed increase
16	512	2,076	13.06

The following graph shows the DrivAer_BodyFitted speedup performance results.



The following graph shows the DrivAer_BodyFitted execution time results.



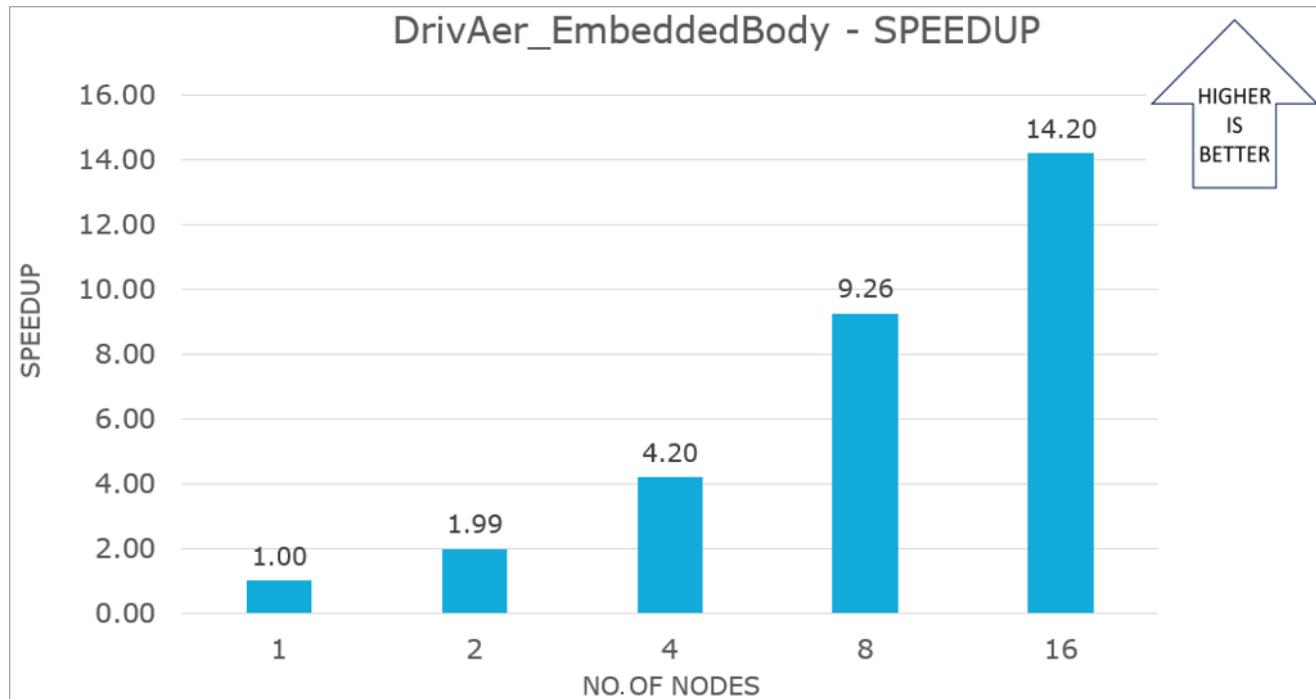
Performance results for DrivAer_EMBEDDEDBody

This table shows the speedup performance results for DrivAer_EMBEDDEDBody.

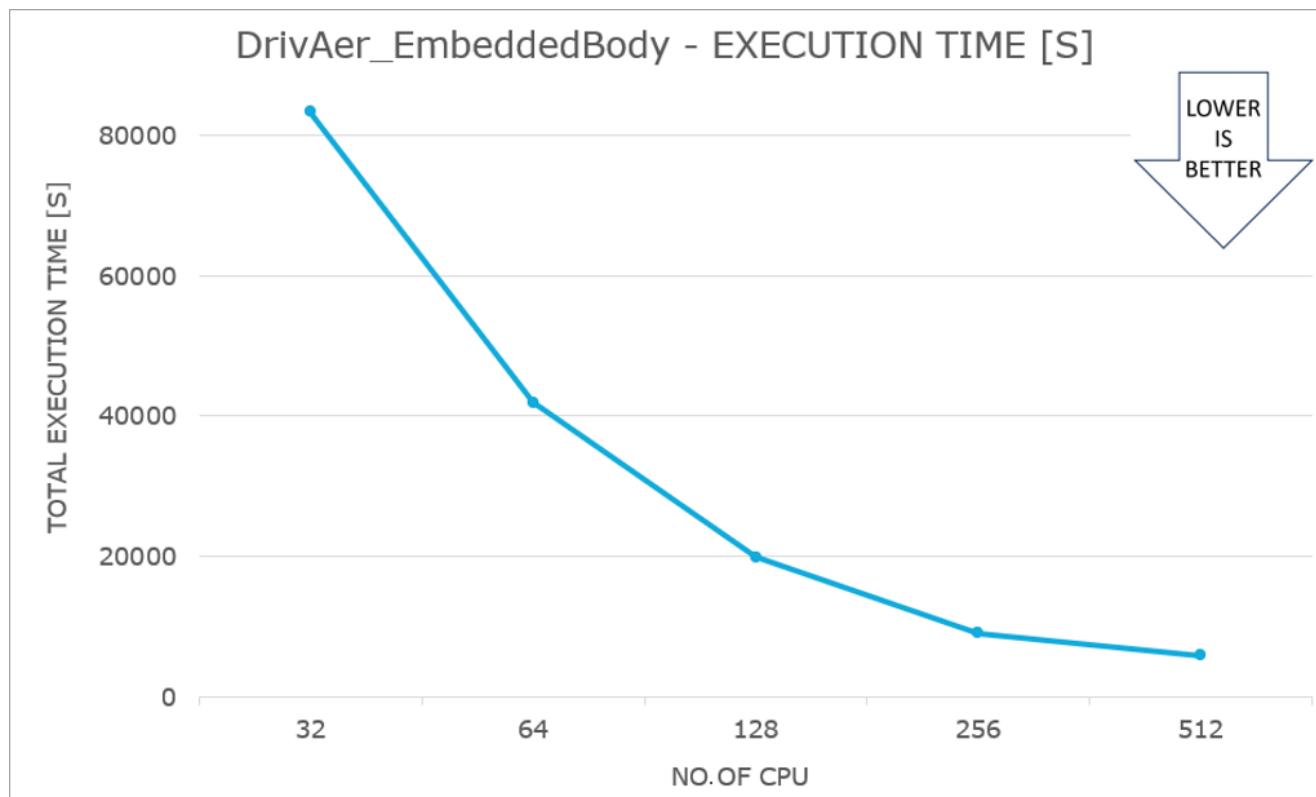
[Expand table](#)

Number of nodes	Number of CPUs	Solver running time (seconds)	Relative speed increase
1	32	83,387	1.00
2	64	41,918	1.99
4	128	19,851	4.20
8	256	9,004	9.26
16	512	5,872	14.20

The following graph shows the speedup performance results for DrivAer_EMBEDDEDBody.



The following graph shows the DrivAer_EMBEDDEDBody execution time results.



ⓘ Note

To get a better speedup, there must be a minimum of 20,000 cells per CPU for a single-phase incompressible flow simulation.

More notes about the tests:

- Both models demonstrate good CPU acceleration in all multi-node setups.
- This research is limited to a few iterations. But in real-world circumstances, iterations can be more numerous, so you can minimize decomposition time in the total run time, allowing performance to be improved even further.
- For small problems, we recommend using fewer CPUs to get better performance.

Azure cost

Only elapsed solver running time (total run time) was considered for these cost calculations. Application installation time isn't considered. The calculations are relative. The actual numbers depend on the size of the model.

You can use the [Azure pricing calculator](#) to estimate costs for your configuration. The following tables provide elapsed times in hours. To compute the total cost, multiply by the number of nodes and Virtual Machine hourly costs. For the current hourly costs, see [Linux Virtual Machines pricing](#).

DrivAer_BODYFITTED

ⓘ [Expand table](#)

Number of nodes	Solver running time (hours) *
1	7.5
2	3.8
4	1.9
8	0.9

Number of nodes	Solver running time (hours) *
16	0.6

DrivAer_EMBEDDEDBody

 [Expand table](#)

Number of nodes	Solver running time (hours) **
1	23.2
2	11.6
4	5.5
8	2.5
16	1.6

*The total run time presented here is for 300 iterations only. The analysis time for a fully converged solution can differ.

**The total run time presented here is for 500 iterations only. The analysis time for a fully converged solution can differ.

Summary

- HPC on Azure provides fully managed platform services and a robust architecture to run HPC workloads and applications.
- Azure offers robust compute services that provide unlimited scalability options for HPC applications. You can use HB-series VMs for memory-bound applications and N-series VMs for graphic-intensive applications.
- Azure CycleCloud lets you manage and orchestrate workloads, define access controls with Microsoft Entra ID, and customize cluster policies.
- AVL FIRE M was successfully tested on HBv3 AMD EPYC™ 7V73X (Milan-X) series on Azure CycleCloud multi-node setup.
- AVL FIRE M demonstrates good upscale with an increase in the number of CPUs in a multi-node setups.
- For improved performance, there must be a minimum of 20,000 cells per CPU for single-phase incompressible flow simulations.
- For small problems, we recommend that you use fewer CPUs to improve performance.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vivi Richard](#) | HPC Performance Engineer

Other contributors:

- [Liz Casey](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Linux virtual machines on Azure](#)

- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)
- [What is Azure CycleCloud?](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Barracuda Virtual Reactor on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running CPFD's [Barracuda Virtual Reactor](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running Virtual Reactor on Azure.

Virtual Reactor simulates the 3D transient behavior in fluid-particle systems, including multiphase hydrodynamics, heat balance, and chemical reactions.

Virtual Reactor has these capabilities:

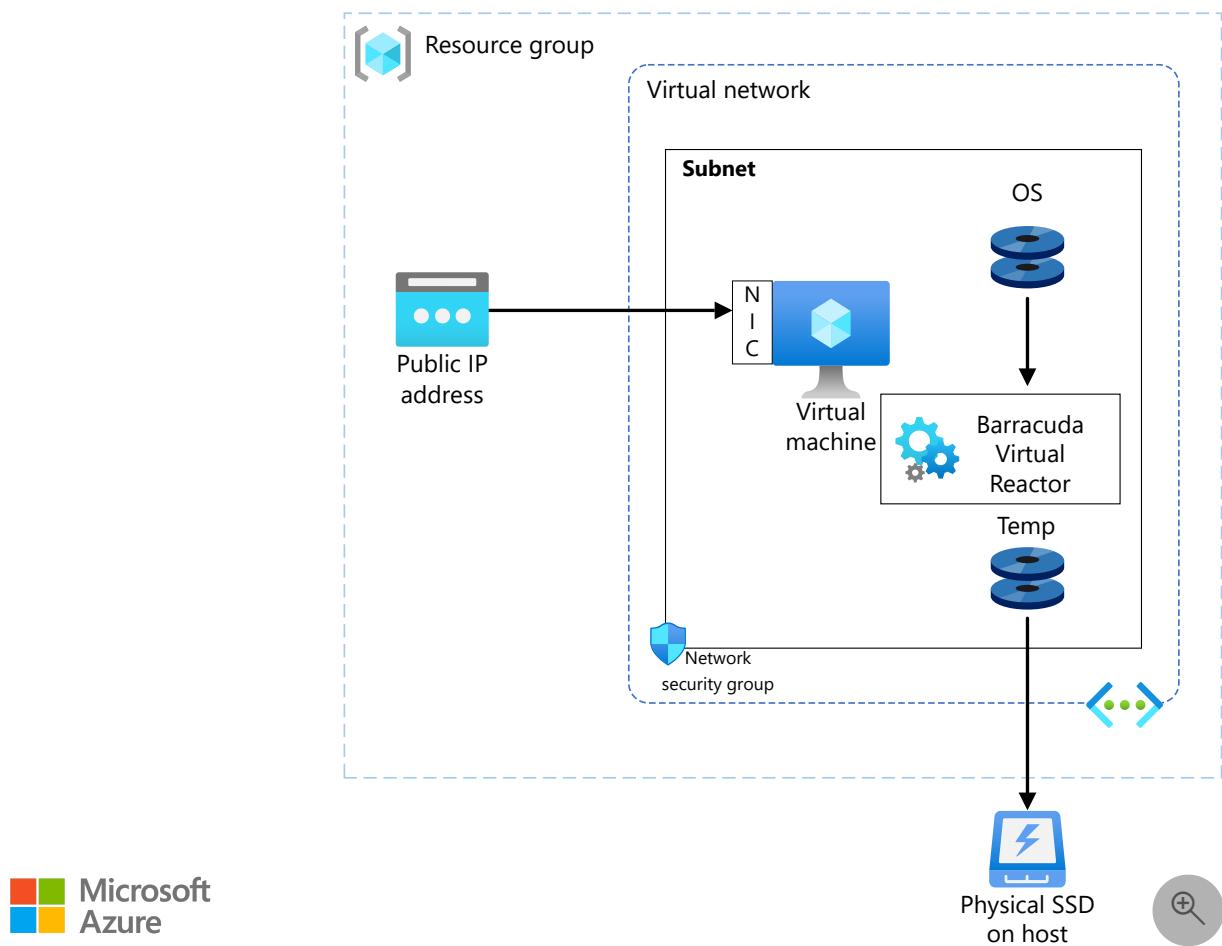
- Uses the Lagrangian formula for the particulate phase, which allows inclusion of discrete particle properties, including the particle size distribution (PSD), composition, temperature, residence time, and history.
- Provides directional particle filtering through baffles and a GUI.

Barracuda is most widely used in the oil refining, petrochemical, energy, and minerals processing industries, including the clean energy sector in the production of electricity, gas, and liquid fuels from coal and biomass.

Why deploy Barracuda Virtual Reactor on Azure?

- Modern and diverse compute options to align to your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- For simulations with high particle counts, impressive scaling as GPUs are added

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Linux VM. For information about deploying the VM and installing the drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

Performance tests of Barracuda Virtual Reactor on Azure used [ND A100_v4](#), [NCv3](#), and [NCasT4_v3](#) series VMs running Linux. The following table provides details.

[Expand table](#)

VM size	vCPU	Memory (GiB)	SSD (GiB)	GPU	GPU memory (GiB)	Maximum data disks
Standard_ND96asr_v4	96	900	6,000	8 A100	40	32
Standard_NC24s_v3	24	448	2,948	4 V100	64	32
Standard_NC64as_T4_v3	64	440	2,880	4 T4	64	32

Required drivers

To take advantage of the GPU capabilities of [ND A100_v4](#), [NCv3](#), and [NCasT4_v3](#) series VMs, you need to install NVIDIA GPU drivers.

To use AMD processors on [ND A100_v4](#) and [NCasT4_v3](#) series VMs, you need to install AMD drivers.

Barracuda Virtual Reactor installation

Before you install Virtual Reactor, you need to deploy and connect a Linux VM and install the required NVIDIA and AMD drivers.

i **Important**

NVIDIA Fabric Manager installation is required for VMs that use NVLink or NVSwitch. NDv4 series VMs use NVLink.

For information about deploying the VM and installing the drivers, see [Run a Linux VM on Azure](#).

You can install Virtual Reactor from the [CPFD Downloads page](#). For information about the installation process, see [CPFD customer support](#).

Virtual Reactor performance results

Particle-based fluid dynamics simulations were run to test Virtual Reactor. The following table provides details about the operating system and NVIDIA drivers that were used.

[] [Expand table](#)

Operating system version	OS architecture	GPU driver version	Cuda version
CentOS Linux release 8.1.1911 (Core)	x86-64	470.57.02	11.4

A number of test cases were run. The following table provides details.

[Expand table](#)

Test case number	Cell count	Number of particles	Chemistry	Thermal	P1 model
479	243,267	29,920,300	Enabled	Enabled	Disabled
499	243,267	3,581,140	Enabled	Enabled	Disabled
480	105,157	40,581,300	Disabled	Disabled	Disabled
500	105,157	9,528,670	Disabled	Disabled	Disabled
481	389,320	50,170,500	Enabled	Enabled	Disabled
501	389,320	12,994,400	Enabled	Enabled	Disabled
482	821,781	55,070,200	Enabled	Enabled	Disabled
502	821,781	22,625,700	Enabled	Enabled	Disabled

Results on NDv4

Results are presented in seconds.

[Expand table](#)

Test case number	479	480	481	482	499	500	501	502
CPU	422.55	755.8	755.8	1,258.1	92.92	222.81	594.3	601.59
1 GPU ¹	2.92	12.7	12.7	13.8	1.453	2.63	9.57	6.59

Test case number	479	480	481	482	499	500	501	502
2 GPU ¹	2.85	7.4	7.4	7.4	1.358	1.8	6.38	4.57
3 GPU ¹	2.68	5.4	5.4	6.1	1.629	1.61	5.65	4.49
4 GPU ¹	3.35	4.6	4.6	7.8	1.845	1.71	5.41	4.44
5 GPU ¹	3.35	4.5	4.5	8	2.189	1.83	5.86	5.39
6 GPU ¹	3.67	4.8	4.8	8	2.908	2.36	6.21	6.46
7 GPU ¹	4.17	4.6	4.6	9.6	3.433	2.69	8.56	6.63
8 GPU	4.62	4.6	4.6	9	3.971	3.01	8.12	7.93

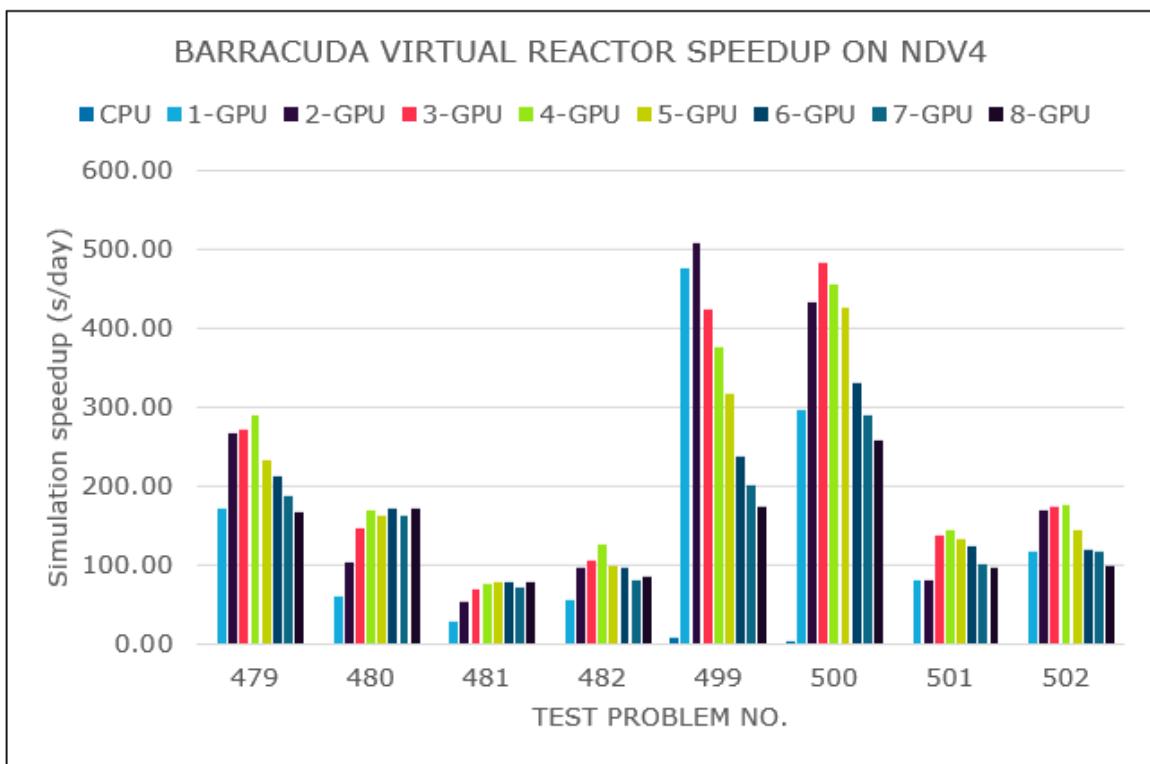
The following table and graph show speed increases, in seconds of chemical reaction completed per day, for each configuration.

[Expand table](#)

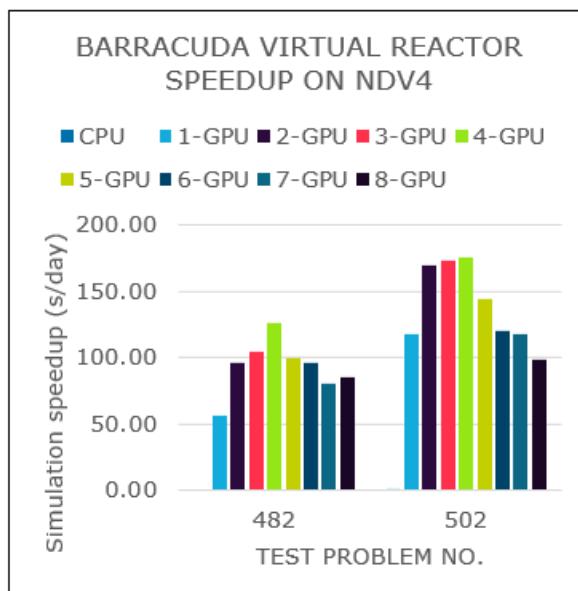
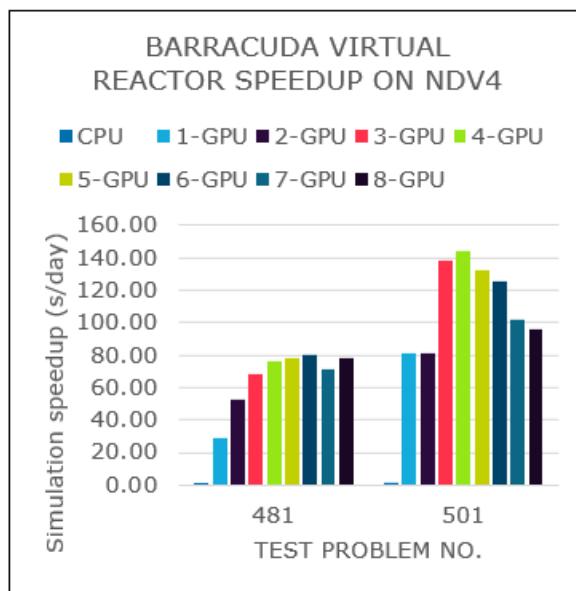
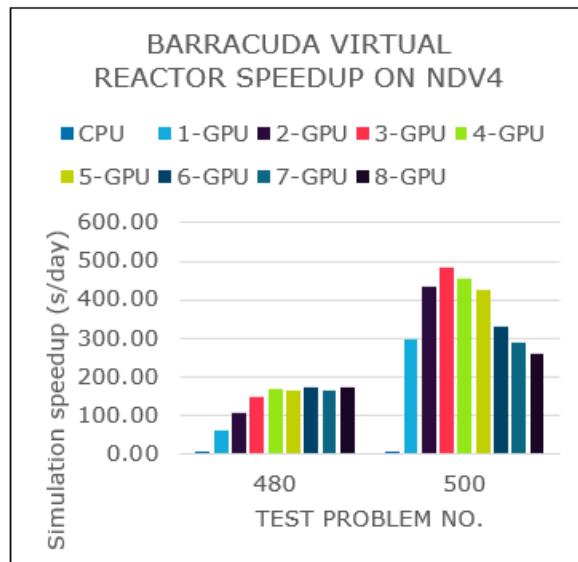
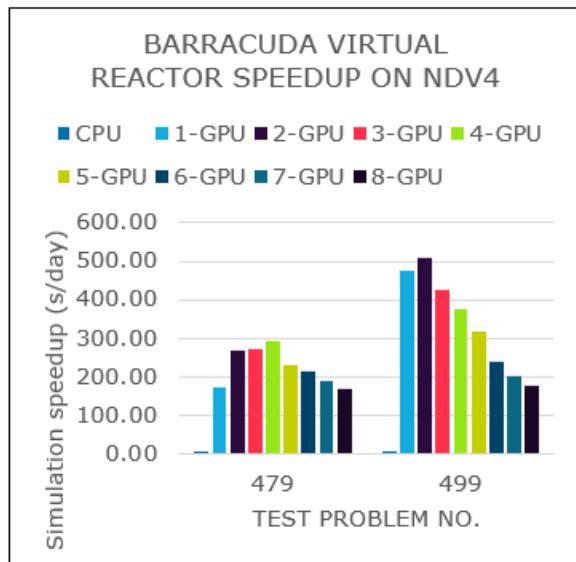
Test case number	479	480	481	482	499	500	501	502
CPU	1.84	1.03	0.49	0.62	7.44	3.49	1.31	1.29
1 GPU ¹	172.10	61.10	29.37	56.11	475.84	295.82	81.31	118.05
2 GPU ¹	266.81	104.80	52.80	96.60	509.10	433.26	81.31	170.29
3 GPU ¹	272.85	145.75	68.53	105.11	424.87	482.75	137.87	173.19
4 GPU ¹	290.68	168.87	75.92	126.11	375.23	454.95	144.20	175.39
5 GPU ¹	232.16	162.10	77.81	99.79	316.32	426.85	132.83	144.44
6 GPU ¹	211.66	172.31	79.69	96.47	238.60	329.83	125.22	120.73
7 GPU ¹	186.82	163.73	71.68	81.18	201.66	289.21	101.61	117.47

Test case number	479	480	481	482	499	500	501	502
8 GPU	168.21	171.39	78.23	85.89	174.89	258.41	96.02	98.20

1. In these cases, the number of GPUs was artificially limited. This VM has eight GPUs.



These graphs provide comparisons of models that are similar but have different particle counts:



Results on NCv3

Results are presented in seconds.

[Expand table](#)

Test case number	479	480	481	482	499	500	501	502
CPU	595.68	1,146.1	5,327.5	1,768.7	113.751	335.73	772.93	678.59
1 GPU ²	8.8	55	216.3	183.2	2.17	5.19	17.2	12.88
2 GPU ²	8.03	12.5	49.9	22.7	5.217	5.2	24.37	13.8

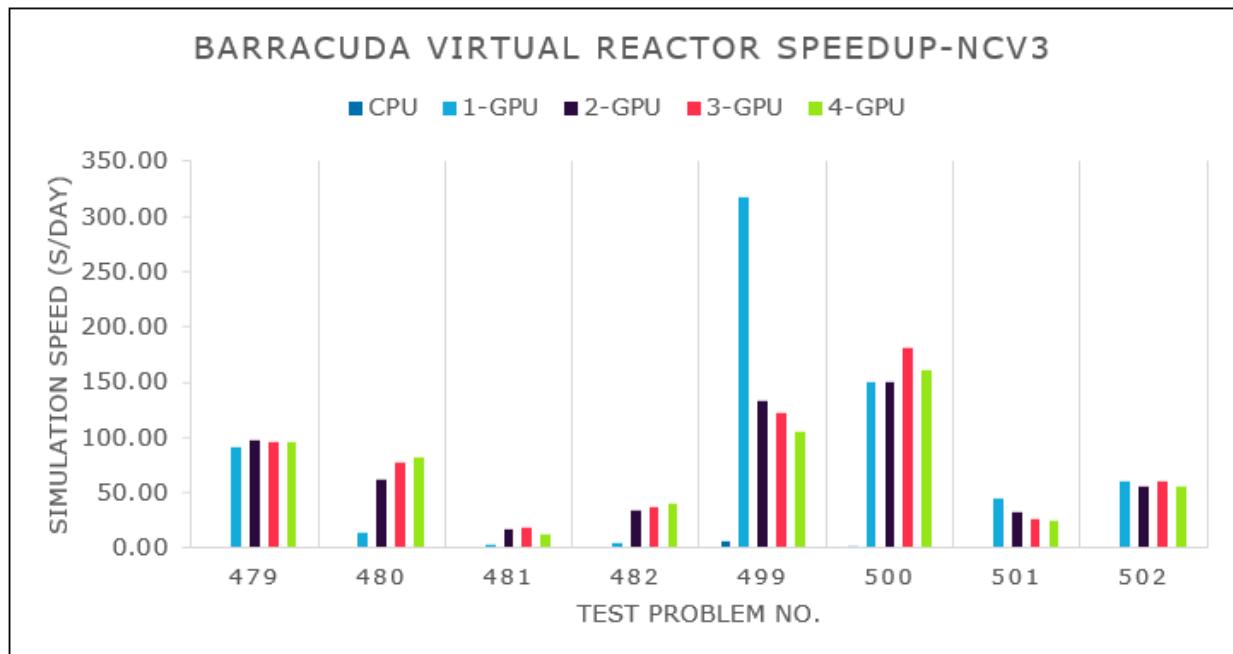
Test case number	479	480	481	482	499	500	501	502
3 GPU ²	8.06	10.1	46	20.8	5.708	4.31	29.68	12.88
4 GPU	8.07	9.5	67.1	19.1	6.61	4.84	35.98	13.93

The following table and graph show speed increases, in seconds of chemical reaction completed per day, for each configuration.

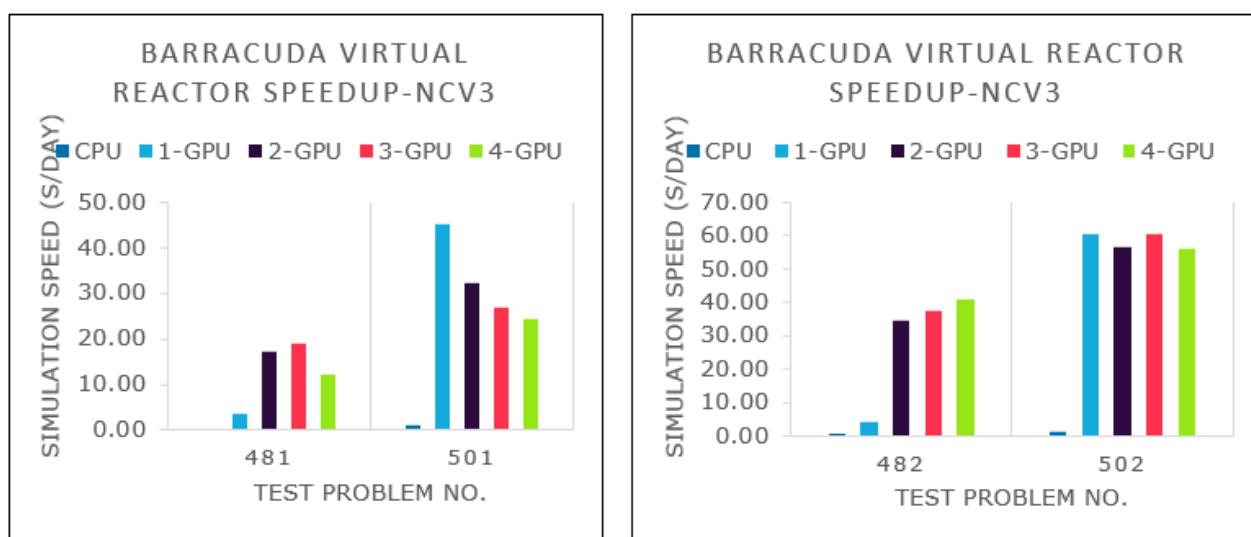
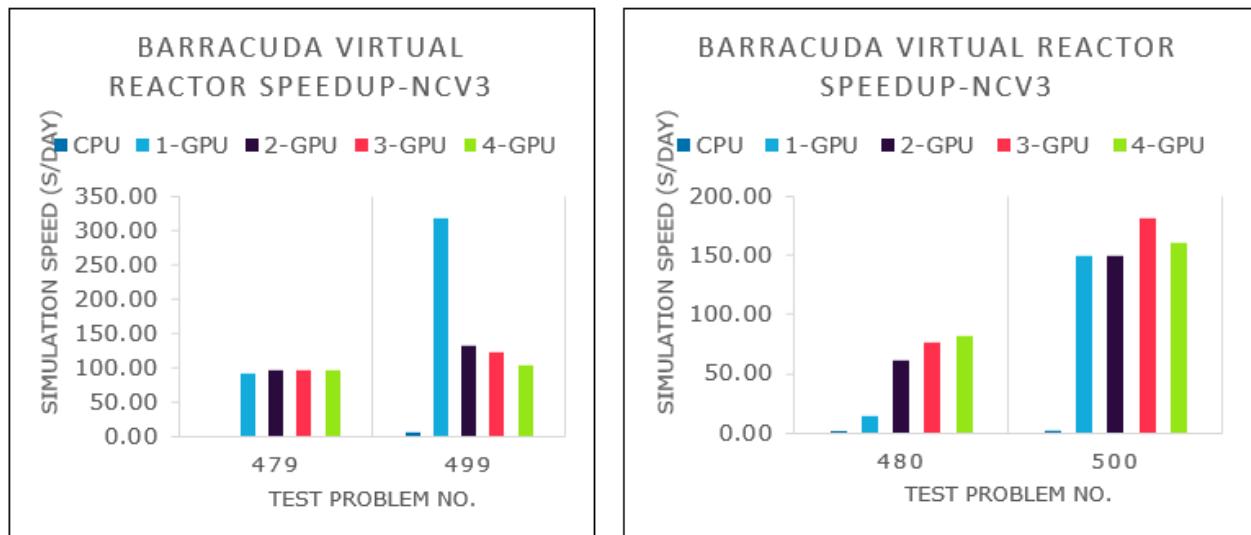
[\[+\] Expand table](#)

Test case number	479	480	481	482	499	500	501	502
CPU	1.31	0.68	0.24	0.44	6.08	2.32	1.01	1.15
1 GPU ²	91.28	14.14	3.60	4.25	317.20	149.69	45.20	60.38
2 GPU ²	96.82	61.91	17.18	34.33	132.88	149.65	32.40	56.41
3 GPU ²	96.62	76.89	18.84	37.51	122.35	180.74	27.03	60.45
4 GPU	96.45	82.22	12.13	40.68	105.15	160.74	24.38	55.87

2. In these cases, the number of GPUs was artificially limited. NCv3 VMs are available with one, two, or four GPUs.



These graphs provide comparisons of models that are similar but have different particle counts:



Results on NCasT4_v3

Results are presented in seconds.

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

Test case number	479	480	481	482	499	500	501	502
CPU	439.31	789.6	1,673.1	1,266.5	96.427	251.59	609.31	609.34
1 GPU ³	28.31	87.1	295.1	238	6.37	9.9	49.15	39.5
2 GPU ³	16.82	29.9	163.6	50.7	7.271	7.95	87.51	27.47
3 GPU ³	14.17	21.7	258.2	45.2	7.47	7.72	127.32	24.21
4 GPU	12.73	18	351.1	35.4	8.025	7.72	128.35	22.34

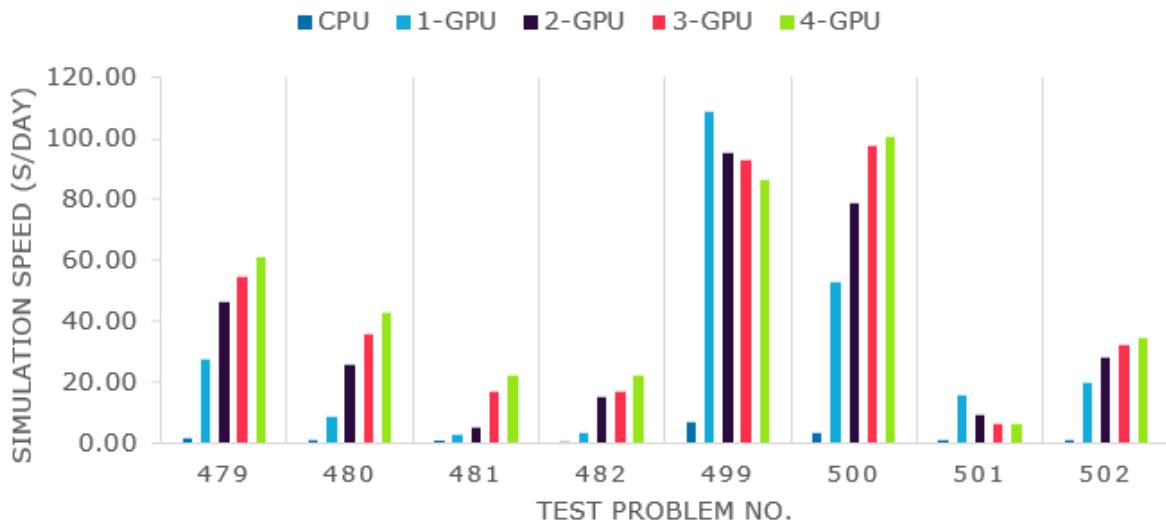
The following table and graph show speed increases, in seconds of chemical reaction completed per day, for each configuration.

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

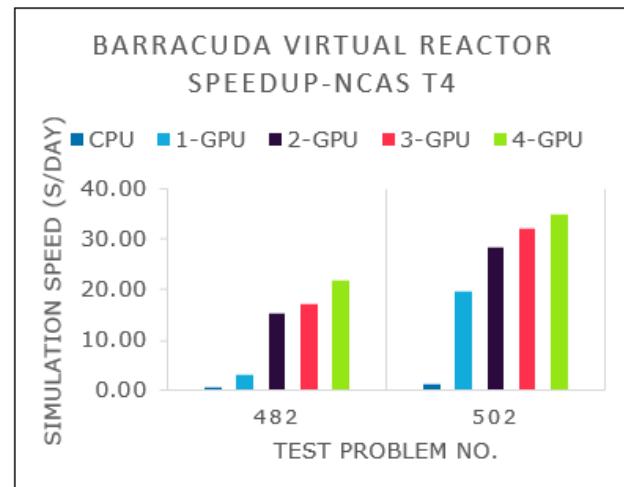
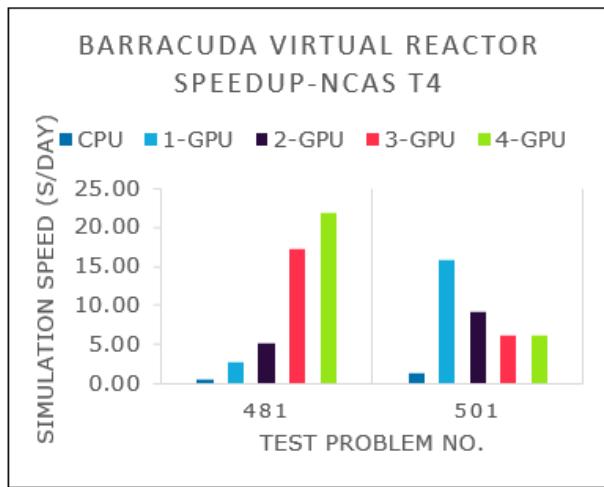
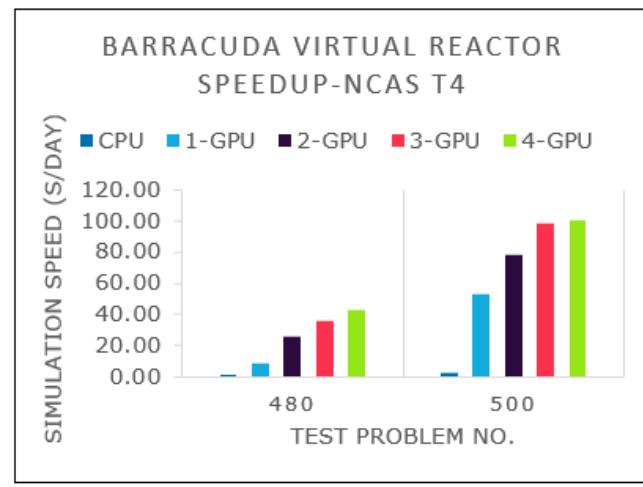
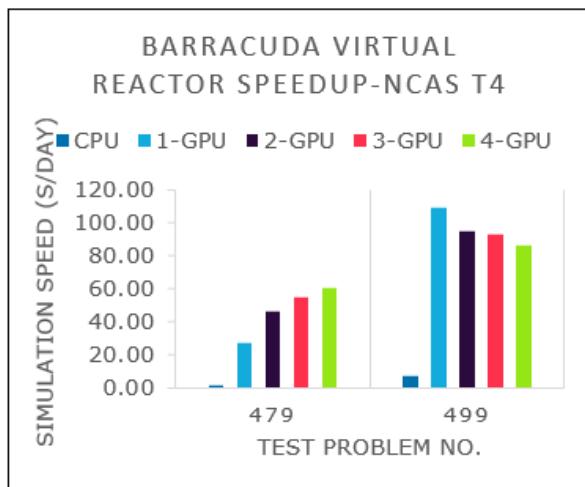
Test case number	479	480	481	482	499	500	501	502
CPU	1.77	0.98	0.47	0.61	7.17	3.09	1.28	1.28
1 GPU ³	27.50	8.93	2.64	3.27	108.69	52.73	15.82	19.69
2 GPU ³	46.23	26.00	5.08	15.32	95.26	78.55	9.19	28.30
3 GPU ³	54.90	35.83	17.21	17.21	92.92	97.88	6.12	32.16
4 GPU	61.07	43.13	21.95	21.95	86.39	100.70	6.07	34.86

3. In these cases, the number of GPUs was artificially limited. NCast4_v3 VMs are available with one or four GPUs.

BARRACUDA VIRTUAL REACTOR SPEEDUP-NCAS-T4



These graphs provide comparisons of models that are similar but have different particle counts:



Azure cost

The following tables present wall-clock times that you can use to calculate Azure costs. You can multiply the times presented here by the Azure hourly rates for NDA100v4, NCsv3, and NCas_T4_v3 series VMs to calculate costs. For the current hourly costs, see [Linux Virtual Machines Pricing](#).

The times presented in the following tables represent the total elapsed time for running all eight of the tests described earlier in this document. Only the wall-clock time for running the test cases is considered for these cost calculations. Application installation time isn't considered. The times presented are indicative. The actual times depend on the size of the simulation. The elapsed times for full production-level test cases are higher than the results presented here, so the associated costs are higher.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

Cost for ND96asr_v4

 [Expand table](#)

CPU/GPU	Elapsed time (hours)
CPU	1.31
8 GPU	0.01

Cost for NC24s_v3

 [Expand table](#)

CPU/GPU	Elapsed time (hours)
CPU	2.98
1 GPU	0.14
2 GPU	0.04
4 GPU	0.05

Cost for NC64as_T4_v3

 [Expand table](#)

CPU/GPU	Elapsed time (hours)
CPU	1.59
1 GPU	0.21
4 GPU	0.16

Summary

- Barracuda Virtual Reactor was successfully tested on NDv4, NCv3, and NCasT4_v3 VMs on Azure.
- On NDv4 VMs, the application scales well up to four GPUs for models with higher particle counts. For models with lower particle counts, it scales only up to two GPUs.
- On NCv3 VMs, performance scales up to three GPUs for models with higher particle counts. For models with lower particle counts, we recommend a one-GPU configuration.
- On NCasT4_v3 VMs, the application scales well up to four GPUs for models with higher particle counts. For models with lower particle counts, it scales well only with a one-GPU configuration.
- For simulations with high numbers of particles and cells, a single 16-GB GPU might not run well because of the required memory. In these cases, you need to run the simulation with two GPUs. You can see this issue in the results for test cases 480, 481, and 482 on the NCv3 and NCasT4_v3 VMs.
- For smaller simulations, when there are performance penalties when all GPUs are used on an instance, you can run concurrent simulations using the other GPUs. In this scenario, multiple points in the simulation parameter space can be explored more quickly.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy CP2K on a virtual machine

Azure Virtual Machines Azure Virtual Network

This article describes the steps for running a CP2K application on a virtual machine (VM) deployed on Azure. It also presents the performance results of running CP2K on single-node and multi-node VM configurations.

CP2K is a quantum chemistry and solid-state physics software package that can perform atomistic simulations of solid state, liquid, molecular, periodic, material, crystal, and biological systems. CP2K provides a general framework for different modeling methods, such as DFT, using the mixed Gaussian and plane waves approaches GPW and GAPW. Supported theory levels include DFTB, LDA, GGA, MP2, RPA, semi-empirical methods, and classical force fields. CP2K can do simulations of molecular dynamics, meta dynamics, Monte Carlo, Ehrenfest dynamics, vibrational analysis, core level spectroscopy, energy minimization, and transition state optimization using NEB or dimer method. CP2K is written in Fortran 2008 and can be run efficiently in parallel using a combination of multi-threading, MPI, and CUDA. It includes Ab-initio electronic structure theory methods using the QUICKSTEP module, Ab-initio Molecular Dynamics, and Mixed quantum-classical simulations.

Why deploy CP2K on Azure?

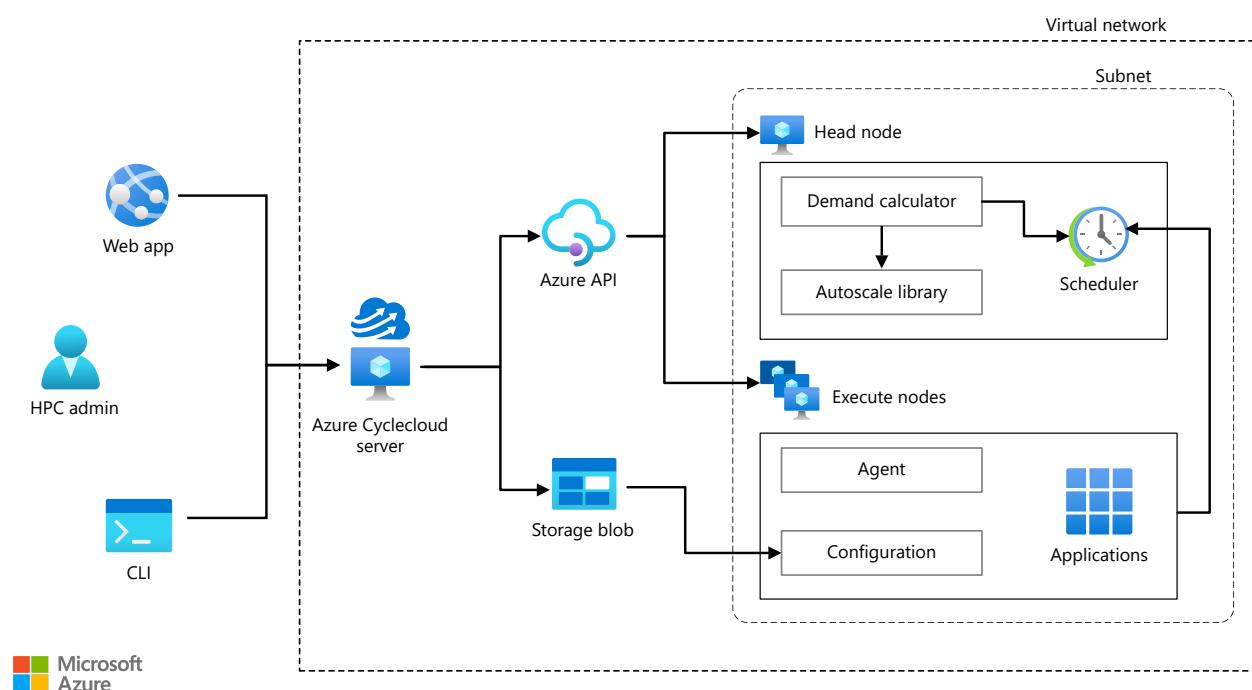
- Single-point energies, geometry optimizations, and frequency calculations
- Several nudged-elastic band (NEB) algorithms (B-NEB, IT-NEB, CI-NEB, D-NEB) for minimum energy path (MEP) calculations
- Global optimization of geometries
- Solvation via the Self-Consistent Continuum Solvation (SCCS) model
- Metadynamics including well-tempered Metadynamics for Free Energy calculations
- Classical Force-Field (MM) and Monte-Carlo (MC) KS-DFT simulations
- Static (such as spectra) and dynamical (such as diffusion) properties
- ATOM code for pseudopotential generation

Architecture

This section shows the differences between the architecture for a multi-node configuration and the architecture for a single-node configuration.

Multi-node configuration:

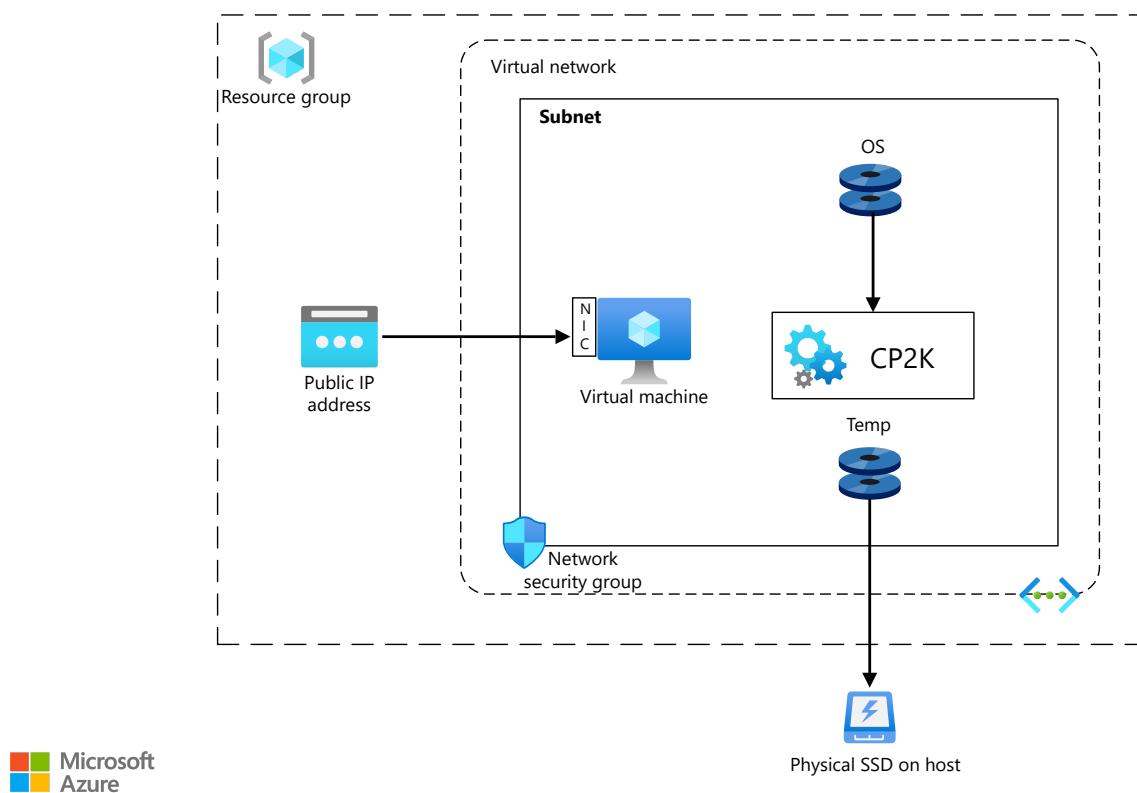
This architecture shows a multi-node configuration:



Download a [Visio file](#) of this architecture.

Single-node configuration:

This architecture shows a single-node configuration:



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Linux VM. For information about deploying the VM and installing the drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- [Azure CycleCloud](#) is used to create the cluster in the multi-node configuration.
- A physical SSD is used for storage.

Compute sizing and drivers

Performance tests of CP2K on Azure HBv3 AMD EPYC™ 7V73X VMs running Linux CentOS Operating system are presented in the following sections. This table provides details about HBv3-series VMs:

[Expand table](#)

VM size	vCPU	Memory (GiB)	Memory bandwidth (GBps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (GBps)	Maximum data disks
Standard_HB120-96rs_v3	96	448	350	1.9	3.0	3.5	200	32

VM size	vCPU	Memory (GiB)	Memory bandwidth (GBps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (GBps)	Maximum data disks
Standard_HB120-64rs_v3	64	448	350	1.9	3.0	3.5	200	32
Standard_HB120-32rs_v3	32	448	350	1.9	3.0	3.5	200	32
Standard_HB120-16rs_v3	16	448	350	1.9	3.0	3.5	200	32

Required drivers

To use InfiniBand, you must enable [InfiniBand](#) drivers.

CP2K installation

For current performance testing, CP2K 2023.1 was used. The software can be downloaded from the [CP2K Official website](#). A CP2K binary distribution need only be unzipped and can be run directly in the resulting directory. To view instructions for building from source code, see [How to compile the CP2K code](#).

Before you install CP2K, you need to deploy and connect to a VM or HPC Cluster.

For information about deploying the VM and installing the drivers, see one of these articles:

- [Run a Windows VM on Azure](#)
- [Run a Linux VM on Azure](#)

For information about deploying the Azure CycleCloud and HPC cluster, see below articles:

- [Install and configure Azure CycleCloud](#)
- [Create a HPC Cluster](#)

CP2K performance results

Three models were considered for testing the performance scalability of CP2K on Azure. The details of each test model are provided in the following sections.

[Expand table](#)

Model Details	H2O-256 (Single Node)	H2O-1024 (Multi Node)	QMMM-CLC-19 (Single Node)
Model Description	Ab-initio molecular dynamics of liquid water using the Born-Oppenheimer approach, using Quickstep DFT. Production quality settings for the basis sets (TZV2P) and the planewave cutoff (280 Ry) are chosen, and the Local Density Approximation (LDA) is used for the calculation of the Exchange-Correlation energy.	Ab-initio molecular dynamics of liquid water using the Born-Oppenheimer approach, using Quickstep DFT. Production quality settings for the basis sets (TZV2P) and the planewave cutoff (280 Ry) are chosen, and the Local Density Approximation (LDA) is used for the calculation of the Exchange-Correlation energy.	A short QM/MM MD simulation of five steps. CIC consists of a (ClC-ec1) chloride ion channel embedded in a lipid bilayer that is solvated in water.
No. of atoms	768	3072	150925

Model Details	H2O-256 (Single Node)	H2O-1024 (Multi Node)	QMMM-CLC-19 (Single Node)
Timestep	10	10	5
Run step	0.5	0.5	1.0
Temperature (K)	300	300	300

CP2K performance results on single-node

The following sections provide the performance results of running CP2K on single-node Azure HBv3 AMD EPYC™ 7V73X VMs.

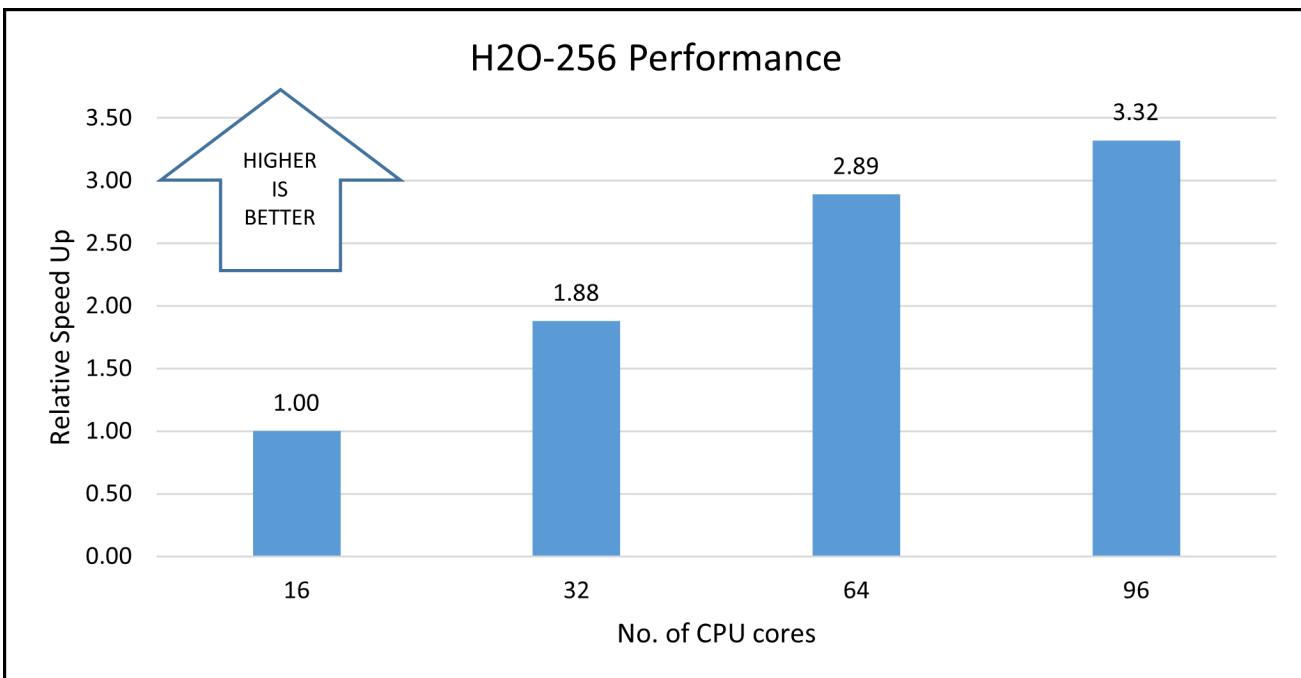
Model 1: H2O-256

This table shows total wall clock time recorded for varying number of CPUs on the standard HBv3-series VM:

[Expand table](#)

Number of cores	Wall clock time (seconds)	Relative speed up
16	1139.52	1.00
32	606.62	1.88
64	394.31	2.89
96	343.12	3.32

This graph shows the relative speedup as the number of CPUs increases:

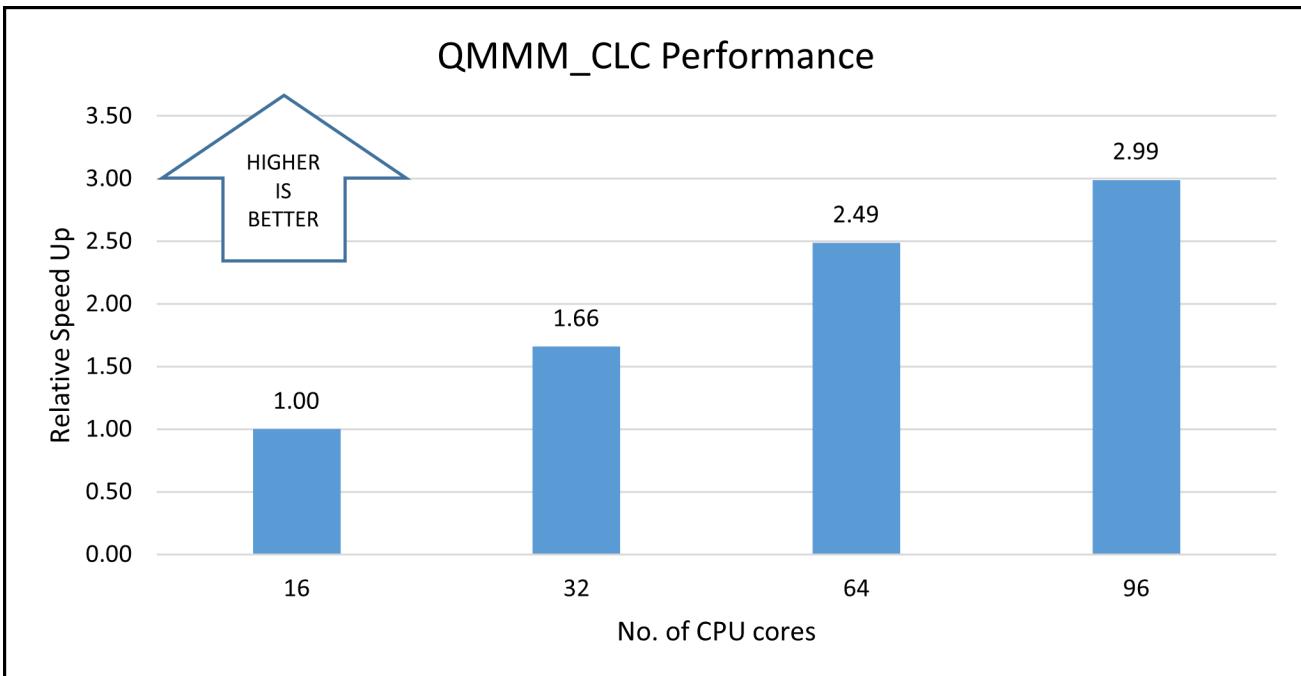


Model 2: QMMM-CLC-19

This table shows total wall clock time recorded for varying number of CPUs on the standard HBv3-series VM:

Number of cores	Wall clock time (seconds)	Relative speed up
16	569.82	1.00
32	343.29	1.66
64	229.00	2.49
96	190.47	2.99

This graph shows the relative speedup as the number of CPUs increases:



Notes about the single-node tests

For all single-node tests, we take the solver time on HB120-16rs_v3 (16 cores) as the reference to calculate the relative speedup with respect to other similar VMs with more cores. The results presented earlier show that parallel performance improves as we increase from 16 to 64 cores. Then at 96 cores, some simulations show limited improvement.

This occurrence is common with these simulations and other memory intensive applications due to the saturation of the onboard memory available on each processor. When you consider VM costs and model complexity, the 32-CPU configuration is the best choice. Standard_HB120-32rs_v3 VMs, which have 32 cores, were used for the multi-node tests.

CP2K performance results on multi-node

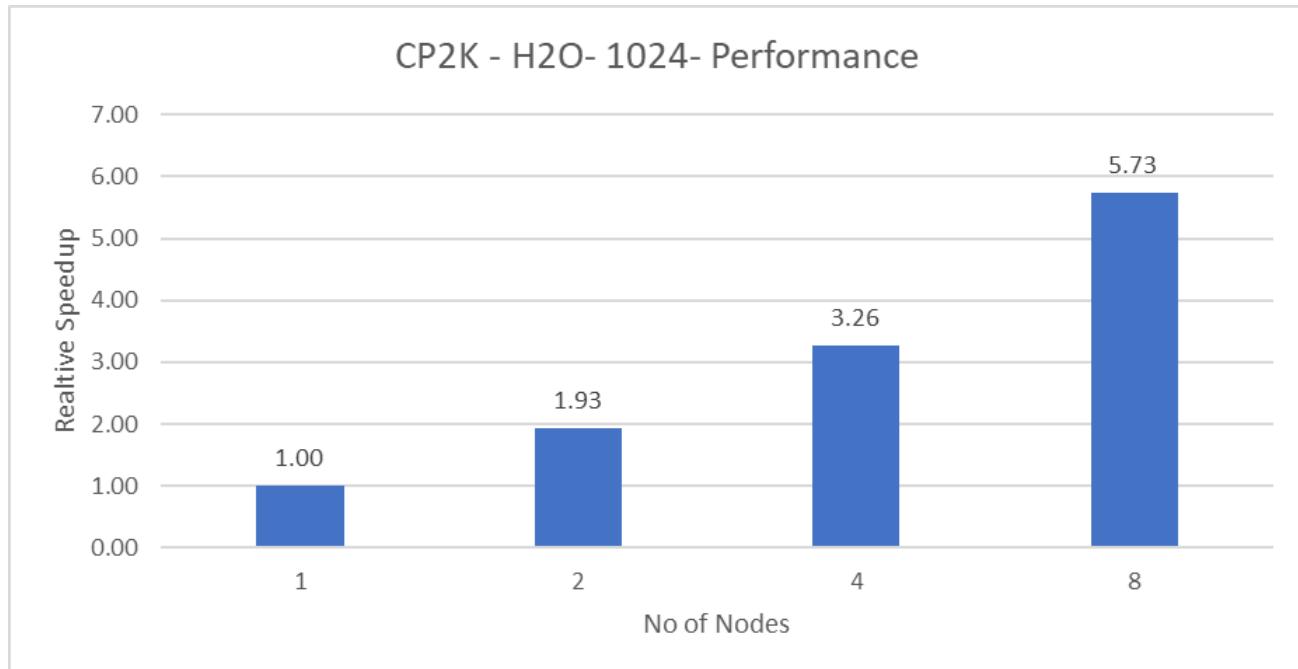
This section provides the performance results of running CP2K on multi-node VMs.

Model 3: H2O-1024

This table shows total wall clock time recorded for varying numbers of nodes with standard HBv3-series VMs:

Number of nodes	Number of cores	Wall clock time (seconds)	Relative speed up
1	32	4651.98	1.00
2	64	2413.77	1.93
3	128	1424.87	3.26
4	256	812.34	5.73

This graph shows the relative speedup as the number of nodes increases:



Azure cost

Only simulation running time is considered for the cost calculations. Installation time, simulation setup time, and software costs were ignored.

You can use the [Azure pricing calculator](#) to estimate VM costs for your configuration.

The following tables provide the solver times in hours. The Azure VM hourly rates are subject to change. To compute the cost, multiply the wall clock time by the number of nodes and the Azure VM hourly cost, which you can find [here for Windows](#) and [here for Linux](#).

Cost for model 1: H2O-256:

Expand table

Number of cores	Wall clock time (hours)
16	0.32
32	0.17
64	0.11
96	0.10

Cost for model 2: QMMM-CLC-19:

 [Expand table](#)

Number of cores	Wall clock time (hours)
16	0.16
32	0.10
64	0.06
96	0.05

Cost for model 3 : H2O-1024:

 [Expand table](#)

Number of nodes	Wall clock time (hours)
1	1.29
2	0.67
4	0.40
8	0.23

Summary

- CP2K was successfully tested on Azure using HBv3 standalone VMs and Azure Cycle Cloud multi-node (cluster) setup.
- We can see a good scaleup for all models on both single-node and multi-node setup. For single-node, we could observe a speedup of ~3.5X for H2O-256 model and a speedup of ~3X for QMMM-CLC-19 with 96 vCPUs.
- For multi-node runs, we could observe a scaleup of ~5.75X for H2O-1024 model with eight nodes.
- If necessary, we recommend that you use fewer CPUs to improve performance.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Saurabh Parave](#) | HPC Performance Engineer
- [Vivi Richard](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Windows virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Devito on an Azure virtual machine

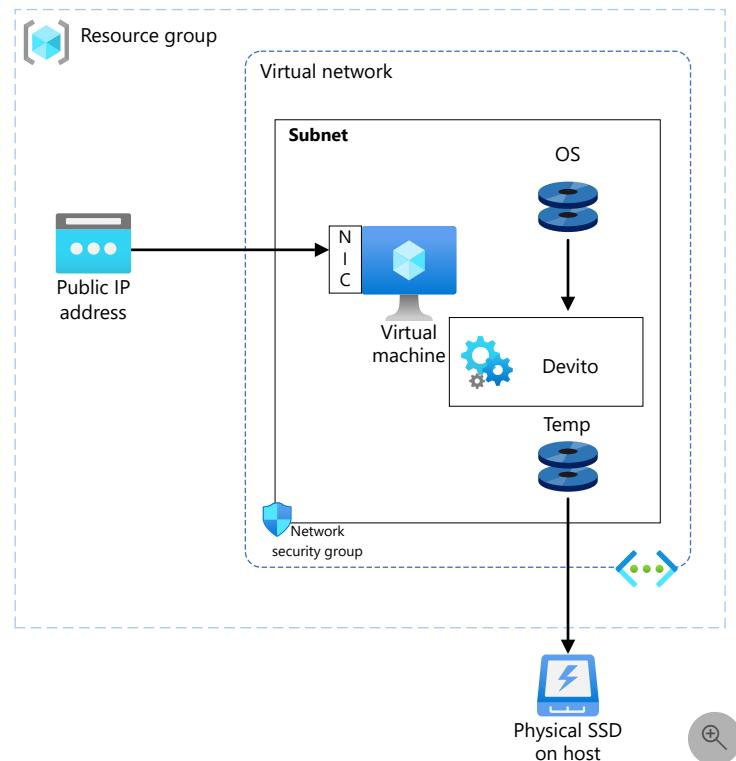
Azure Blob Storage Azure CycleCloud Azure Virtual Machines Azure Virtual Network

This article describes how to run [Devito](#) on an Azure virtual machine (VM). It also presents the performance results of running Devito on Azure.

Devito is a functional language that you can implement as a Python package. With Devito, you can use high-level symbolic problem definitions to create optimized stencil computation, such as finite differences, image processing, and machine learning. Devito is built on [SymPy](#) and uses automated code generation and just-in-time compilation to run optimized computational kernels on several compute platforms, including CPUs, GPUs, and clusters.

Architecture

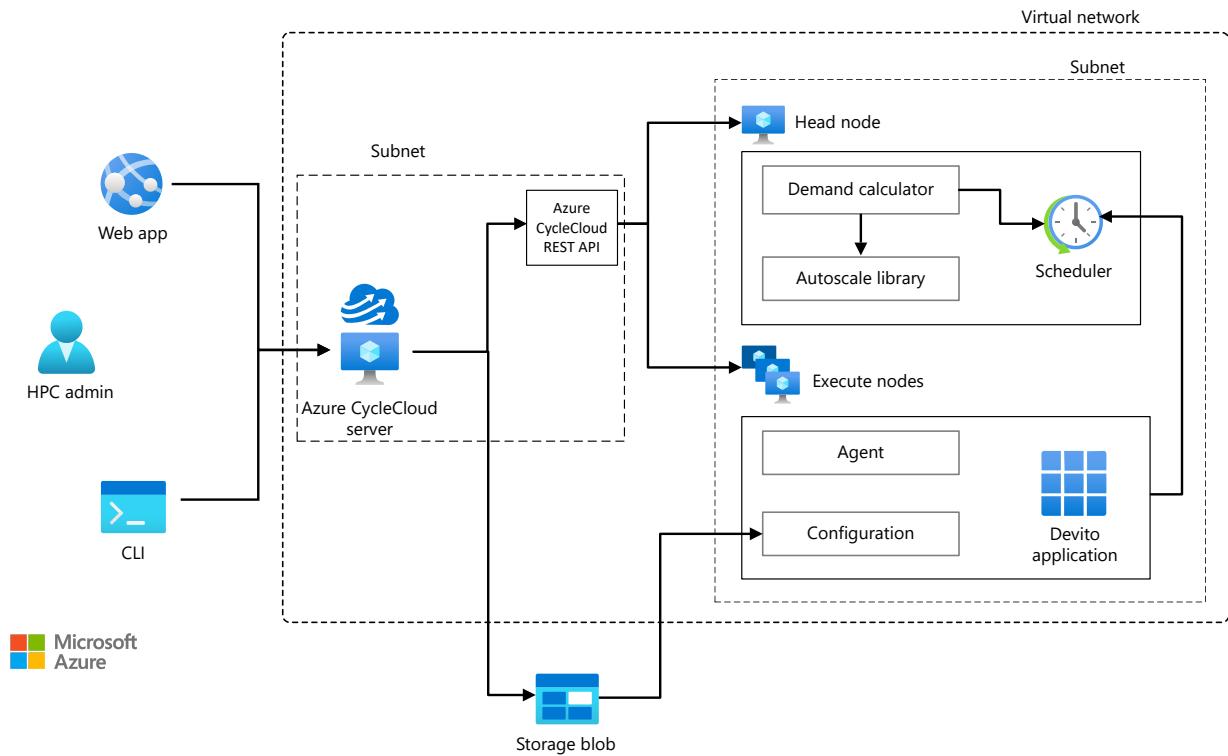
The following diagram shows a single-node architecture:



this architecture.

[Download a Visio file](#) of

The following diagram shows a multi-node architecture:



Download a [Visio file](#) of this architecture.

Components

- [Azure CycleCloud](#) is an enterprise-friendly tool that's used to orchestrate and manage HPC environments on Azure.
- [Azure Virtual Machines](#) is used to create a Linux VM. For information about how to deploy a VM and install drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
- [Network security groups](#) are used to restrict access to the VM.
- A public IP address connects the internet to the VM.
- An [Azure Blob Storage](#) physical solid-state drive (SSD) is used for storage.
- The [Azure CycleCloud REST API](#) is used to add automated and programmatic cluster management capabilities, like determining a cluster's status or creating nodes.

Scenario details

Devito provides key offerings like:

- Mechanisms to adjust finite difference discretization.
- Constructs to express various operators.
- A flexible API.
- The ability to generate highly optimized parallel code.
- Distributed NumPy arrays.
- Smooth integration with popular Python packages.

Deploy Devito on Azure to get benefits like:

- Modern and diverse compute options to meet your workload's needs.
- The flexibility of virtualization without the need to buy and maintain physical hardware.
- Rapid provisioning.

Devito provides a functional language to implement sophisticated operators that can be made up of multiple stencil computations, boundary conditions, sparse operations (for example, interpolation), and more. With Devito, you might use explicit finite difference

methods to approximate partial differential equations. For example, you can implement a 2D diffusion operator by using the following equation:

```
>>> grid = Grid(shape=(10, 10))
>>> f = TimeFunction(name='f', grid=grid, space_order=2)
>>> eqn = Eq(f.dt, 0.5 * f.laplace)
>>> op = Operator(Eq(f.forward, solve(eqn, f.forward)))
```



An operator generates low-level code from an ordered collection of `Eq`. This example is for a single equation.

There's virtually no limit to the complexity of an operator. The Devito compiler automatically analyzes the input, detects and applies optimizations (including single-node and multi-node parallelism), and generates code with suitable loops and expressions.

Install Devito

Before you install Devito, you need to deploy and connect a Linux VM, and install the required AMD and InfiniBand drivers.

For information about deploying the VM and installing the drivers, see [Run a Linux VM on Azure](#).

After you deploy the Linux VM, see the [Devito installation instructions](#) to learn about three methods for installing Devito on your VM:

- Docker installation
- Pip installation
- Conda environment installation

Compute sizing and drivers

The Devito performance tests that are presented in the next sections used [HBv3-series](#) VMs running Linux. The following table provides details about these VMs:

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

VM size	Number of vCPUs (cores)	RAM memory (GiB)	Memory bandwidth (GBps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120rs_v3	120	448	350	1.9	3.0	3.5	200	32
Standard_HB120-96rs_v3	96	448	350	1.9	3.0	3.5	200	32
Standard_HB120-64rs_v3	64	448	350	1.9	3.0	3.5	200	32
Standard_HB120-32rs_v3	32	448	350	1.9	3.0	3.5	200	32
Standard_HB120-16rs_v3	16	448	350	1.9	3.0	3.5	200	32

Devito performance results

Benchmarking Devito on Azure

To test the performance of Devito on Azure, benchmarking was performed by using the HB120rs_v3 series SKU. There are many seismic models, like acoustic, tti, elastic, and visco-elastic, available on [the tutorials page of the Devito website](#). The tests in this article use a forward operator under the acoustic model for benchmarking the performance of Devito.

The following table provides information about the operating system that was used for testing:

[Expand table](#)

Operating system and hardware details (Azure infrastructure)	
Operating system version	CentOS-based 8.1 HPC
OS architecture	x86-64
Processor	AMD EPYC 7V73X

Benchmarking a Devito operator

You can use the `benchmark.py` python file to test the performance of a Devito operator. The file is located in the `/benchmarks/user` folder that's in the Devito folder. The `benchmark.py` file implements a minimalist framework to evaluate the performance of a Devito operator, while varying:

- The problem size, for example the shape of the computational grid.
- The discretization, for example the space-order and time-order of the input/output fields.
- The simulation time (in milliseconds).
- The performance optimization level.
- The autotuning level.

Devito performance on the HB120rs_v3 series (single-node)

The Devito forward operator performance for the acoustic model was tested on Standard HBv3 series virtual machines with 16, 32, 64, 96, and 120 vCPU configurations.

The following table shows the results for the CentOS-based 8.1 HPC image:

[Expand table](#)

Number of vCPUs (cores)	Forward operator runtime (in seconds)	GFLOPS/second	Relative speed increase
16	184.39	211.24	N/A
32	126.20	308.55	1.46
64	117.61	331.22	1.57
96	132.86	293.25	1.39
120	149.99	259.78	1.23

Note that for the single-node tests, the Devito operator is run on all HBv3-series VM configurations. The Standard_HB120-16rs_v3 VM runtime is used as the baseline to calculate the relative speed increase.

Devito performance on a cluster (multi-node)

The forward operator performance in the single-node tests show the scale-up behavior for the 64 and 96 vCPU configurations. The following performance tests run the Devito operator on two cluster configurations with 64 vCPUs and 96 vCPUs, respectively. The CentOS-based 8.1 HPC image is used for these two clusters.

The following table shows the results for a cluster with 64 vCPUs per node:

[Expand table](#)

Number of nodes	Number of vCPUs (cores)	Forward operator runtime (in seconds)	GFLOPS/second	Relative speed increase
1	64	121.73	320.04	N/A
2	128	75.68	514.86	1.61
4	256	60.77	641.30	2.00
8	512	51.94	750.40	2.34

The following table shows the results for a cluster with 96 vCPUs per node:

[Expand table](#)

VM configuration	Number of nodes	Number of vCPUs (cores)	Forward operator runtime (in seconds)	GFLOPS/second	Relative speed increase
Standard_HB120-96rs_v3	1	96	137.19	284	N/A
Standard_HB120-96rs_v3	2	192	88.72	439.27	1.55
Standard_HB120-96rs_v3	4	384	75.11	518.93	1.83
Standard_HB120-96rs_v3	8	768	69.38	561	1.98

Azure cost

The following table presents the wall-clock times for running the simulations. You can multiply these times by the Azure VM hourly costs for HB120rs_v3-series VMs to calculate costs. For the current hourly costs, see [Linux virtual machines pricing](#).

The following runtimes represent only the simulation time. Application installation time isn't considered.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

The following table shows runtimes for the HB120rs_v3 series:

[Expand table](#)

Number of CPUs per node	Forward operator runtime (in hours)
Single node	0.197
64	0.086
96	0.102

Summary

- Devito was successfully deployed and tested on the HB120rs_v3 series VM on Azure.

- For the single-node configuration, the Devito scales well up to 64 and 96 cores. It has a maximum scale up of 1.57 times with 64 cores.
- For the multi-node configuration, there's a gradual scale up from one node to eight nodes in both the clusters with the Standard_HB120-64rs_v3 and the Standard_HB120-96rs_v3 virtual machines.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized VM sizes](#)
- [VMs on Azure](#)
- [Virtual networks and VMs on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Engys ELEMENTS on an Azure virtual machine

Azure Virtual Machines Azure Virtual Network Azure CycleCloud

This article describes the steps for running [Engys ELEMENTS](#) on a virtual machine (VM) and on an HPC cluster on Azure. It also presents the performance results of running ELEMENTS on single-node and multi-node VM configurations.

ELEMENTS is a computational fluid dynamics (CFD) and optimization solution for vehicle design applications. The simulation engine that's provided with ELEMENTS is powered by [HELYX](#). The resulting solution combines automotive engineering design practices with open-source CFD and optimization methods developed by Engys.

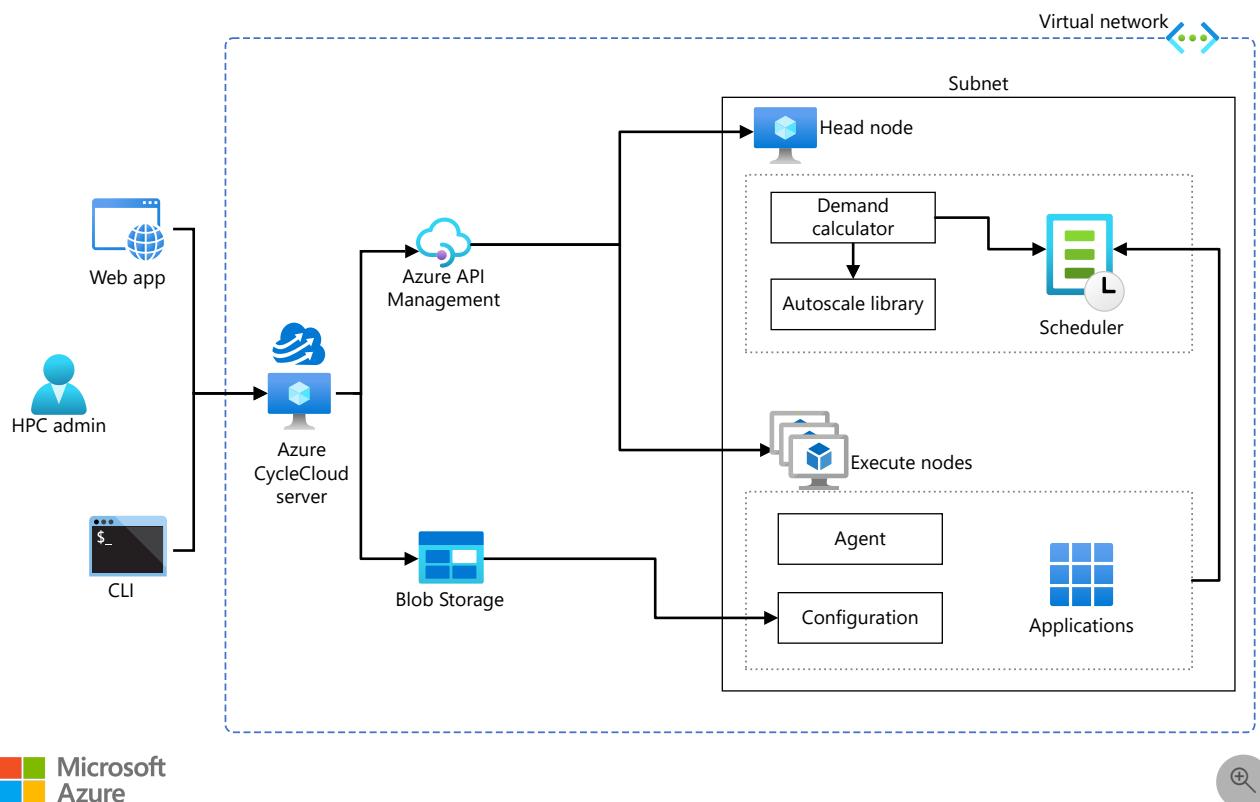
ELEMENTS is used to solve flow-related problems that are encountered in automotive design, including external vehicle aerodynamics, UHTM, HVAC and cabin comfort, aeroacoustics, powertrain, ICE, water management, and soiling. ELEMENTS is also used to analyze the aerodynamics of other vehicles, like high-speed trains, motorcycles, and competition bicycles.

Why deploy ELEMENTS on Azure?

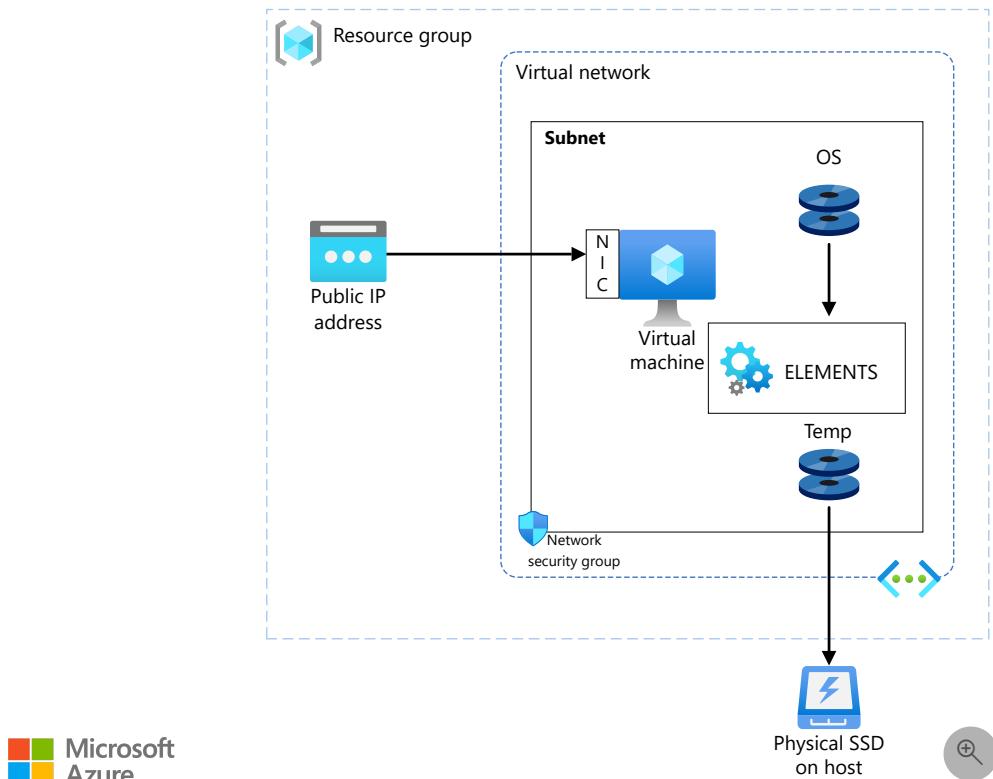
- Modern and diverse compute options to meet your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Complex problems solved within a few hours

Architecture

Multi-node configuration:



Single-node configuration:



Download a [Visio file](#) of all diagrams in this article.

Components

- [Azure Virtual Machines](#) is used to create Linux VMs.
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to control access to the VMs.
 - A public IP address connects the internet to the VM.
- [Azure CycleCloud](#) is used to create the cluster in the multi-node configuration.
- A physical SSD provides storage.

Compute sizing and drivers

Performance tests of ELEMENTS on Azure used [HBv3 AMD EPYC 7V73X \(Milan-X\)](#) VMs running Linux CentOS. The following table provides details about HBv3-series VMs.

[Expand table](#)

VM size	vCPU	Memory (GiB)	Memory bandwidth Gbps	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120rs_v3	120	448	350	1.9	3.0	3.5	200	32
Standard_HB120-96rs_v3	96	448	350	1.9	3.0	3.5	200	32
Standard_HB120-64rs_v3	64	448	350	1.9	3.0	3.5	200	32
Standard_HB120-32rs_v3	32	448	350	1.9	3.0	3.5	200	32

VM size	vCPU	Memory (GiB)	Memory bandwidth Gbps	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120-16rs_v3	16	448	350	1.9	3.0	3.5	200	32

Required drivers

To use AMD CPUs on [HBv3 VMs](#), you need to install AMD drivers.

To use InfiniBand, you need to enable [InfiniBand drivers](#).

Install ELEMENTS 3.5.0 on a VM or HPC cluster

You need to buy ELEMENTS from Engys or one of its local authorized distributors or agents to get the installation files and technical support. For information about buying [ELEMENTS](#), contact Engys.

Before you install ELEMENTS, you need to deploy and connect a VM or HPC cluster.

For information about deploying the VM and installing the drivers, see [Run a Linux VM on Azure](#).

For information about deploying the Azure CycleCloud and HPC cluster, see these articles:

- [Install and configure Azure CycleCloud](#)
- [Create an HPC cluster](#)

ELEMENTS 3.5.0 performance results

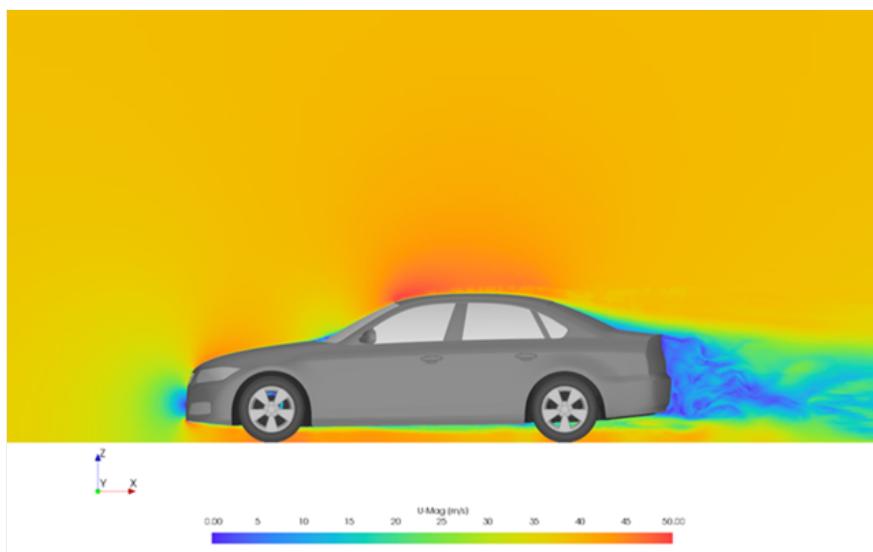
Two vehicle models were used to test the parallel scalability performance of ELEMENTS 3.5.0 on Azure:

- The [DrivAer](#) sedan model (mid-size computational grid) for external vehicle aerodynamics
- The [Generic Truck Utility \(GTU\)](#) model (large computational grid) for external vehicle aerodynamics

The hex-dominant meshing utility that's provided with ELEMENTS was used to create all computational grids. They were created in parallel as part of the execution process.

The details of each test model are provided in the following sections.

Model 1: Automotive_DESdrivAer

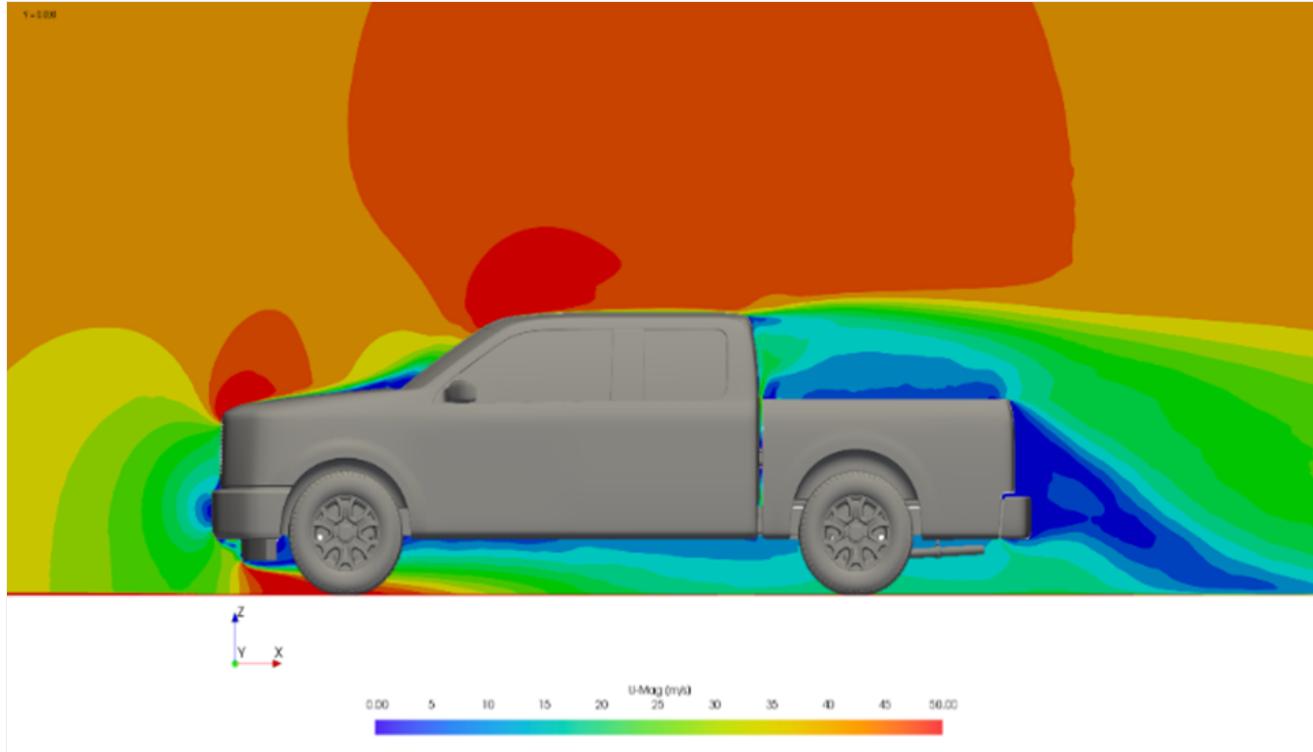


The following table provides details about the model.

[Expand table](#)

Mesh size	Solver	Transient
17,000,000 cells	DES (helyxAero)	400 time steps

Model 2: Automotive_GTU-0001



The following table provides details about the model.

[Expand table](#)

Mesh size	Solver	Transient
116,000,000 cells	DES (helyxAero)	3,583 time steps

ELEMENTS 3.5.0 performance results on single-node VMs

The following section provides the performance results of running ELEMENTS in parallel on single-node Azure [HBv3 AMD EPYC 7V73X \(Milan-X\)](#) VMs. You can use these results as a baseline for comparison with multi-node runs. Only the DrivAer model was tested in single-node configurations.

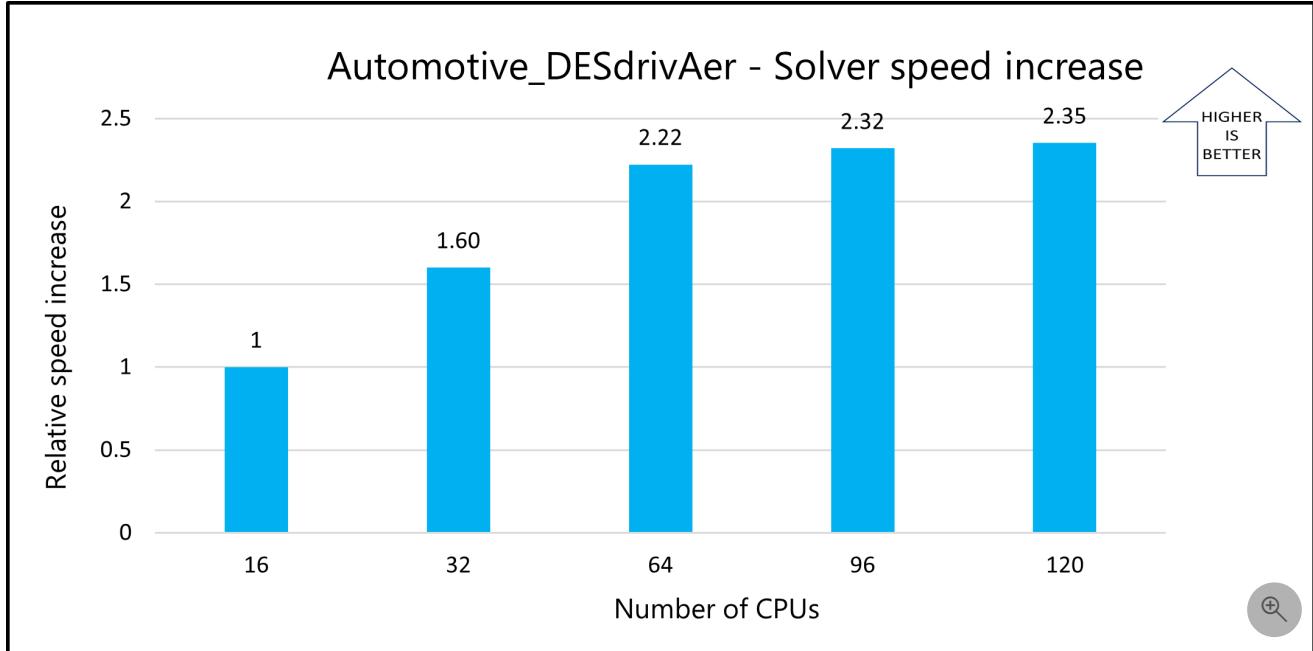
Model 1: Automotive_DESdrivAer

[Expand table](#)

Number of cores	Solver running time (seconds)	Relative speed increase
16	3,102.28	1.00
32	1,938.16	1.60

Number of cores	Solver running time (seconds)	Relative speed increase
64	1,395.36	2.22
96	1,337.25	2.32
120	1,318.55	2.35

The following graph shows the relative speed increases as the number of CPUs increases:



Notes about single-node tests

For all single-node tests, the solver running time on a Standard_HB120-16rs_v3 VM (16 cores) is used as a reference to calculate the relative speed increase with respect to similar VMs that have more cores. The previously presented results for the DrivAer model show that parallel performance in ELEMENTS improves as cores increase from 16 to 64. Above 64 cores, no further improvement occurs. This pattern is common with CFD solvers and other memory-intensive applications because of saturation of the onboard memory that's available on each processor.

The AMD EPYC 7V73-series processor (Milan-X) in the HBv3 VMs tested here is a powerful processor, with 768 MB of total L3 cache. The single-node tests confirm that this memory is sufficient to guarantee parallel scalability of the ELEMENTS solvers when you use half the cores available on each 7V73-series chip.

ELEMENTS 3.5.0 performance results on multi-node clusters

The single-node tests confirm that the solver achieves good parallel performance until you reach 64 cores with HBv3 VMs. Based on those results, only 64-core configurations on [Standard_HB120-64rs_v3](#) were used to evaluate the performance of ELEMENTS on multi-node clusters. The following sections provide the test results.

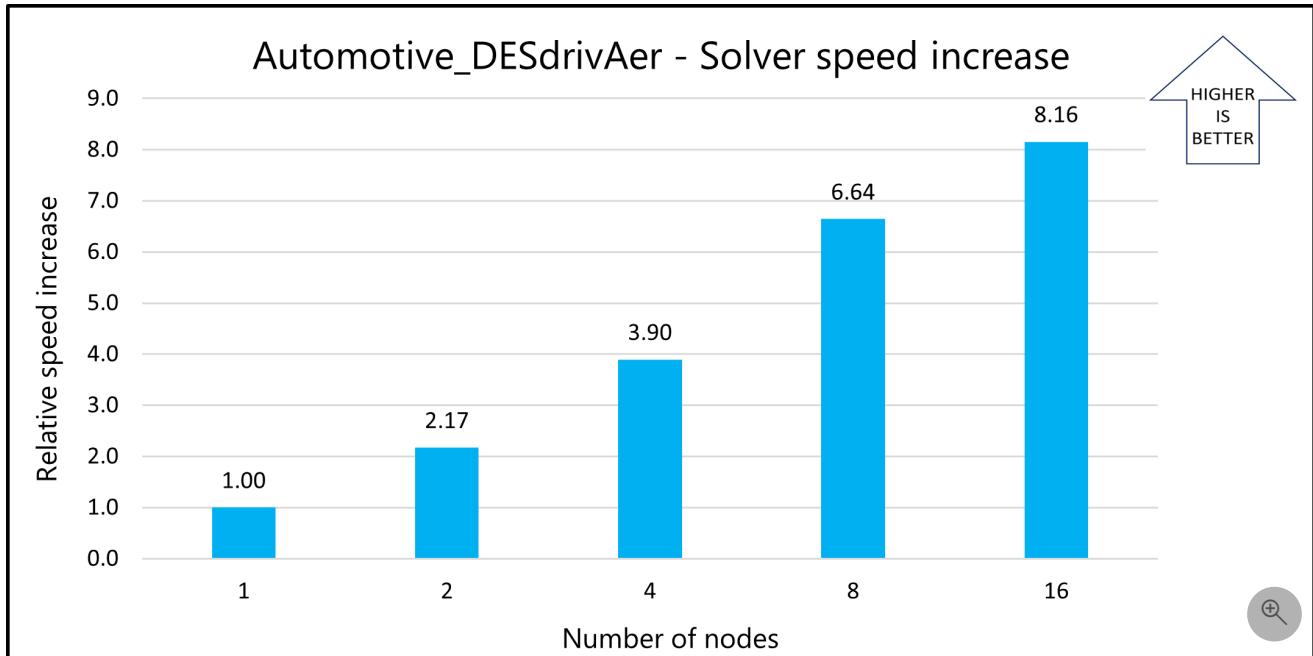
Model 1: Automotive_DESdrivAer

[Expand table](#)

Number of nodes	Number of cores	Cells per core	Solver running time (seconds)	Relative speed increase
1	64	265,625	1,370.81	1.00
2	128	132,813	630.86	2.17

Number of nodes	Number of cores	Cells per core	Solver running time (seconds)	Relative speed increase
4	256	66,406	351.83	3.90
8	512	33,203	206.36	6.64
16	1,024	16,602	168.07	8.16

The following graph shows the relative speed increases as the number of nodes increases:

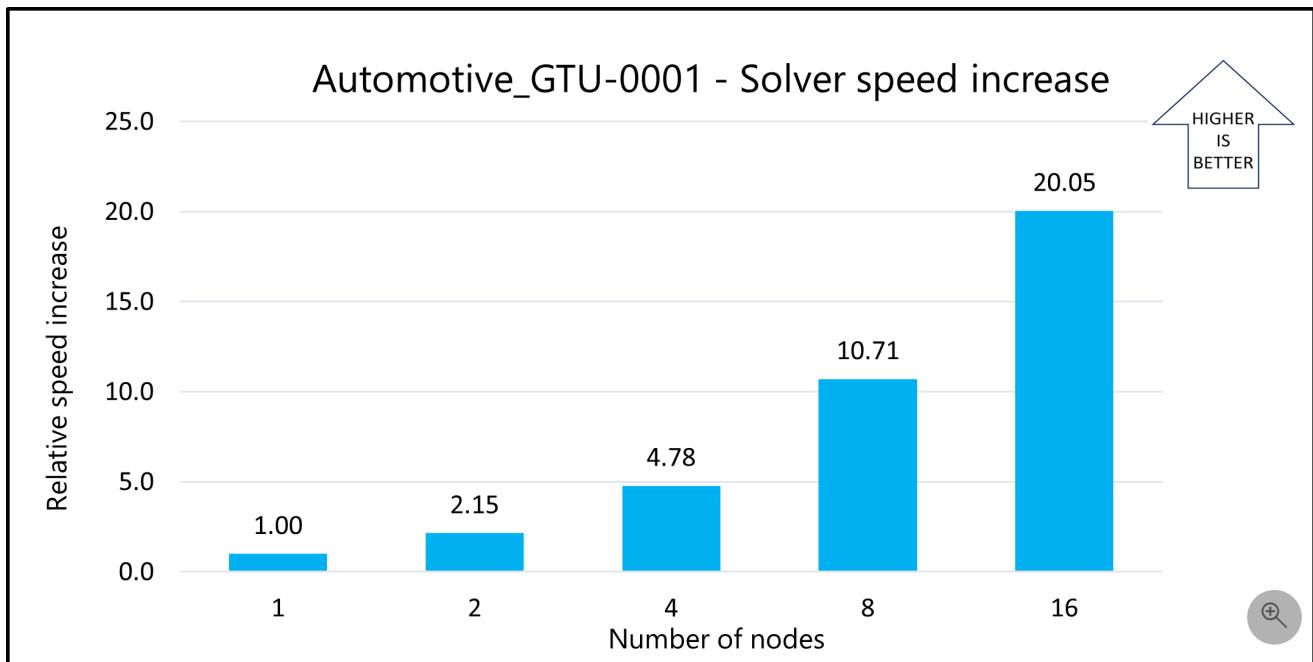


Model 2: Automotive_GTU-0001

[Expand table](#)

Number of nodes	Number of cores	Cells per core	Solver running time (seconds)	Relative speed increase
1	64	1,812,500	102,740.23	1.00
2	128	906,250	47,761.85	2.15
4	256	453,125	21,484.47	4.78
8	512	226,563	9,595.72	10.71
16	1,024	113,281	5,125.38	20.05

The following graph shows the relative speed increases as the number of nodes increases:



Notes about multi-node tests

The multi-node performance tests for the DrivAer model (mid-size mesh) show that the parallel performance improves as the number of nodes increases but is less than optimal. This suboptimal performance can be explained by the relatively low number of cells per core in 8-node and 16-node configurations. Solver performance is known to be reduced by excessive data communication between processor boundaries when the number of cells per core is low.

In contrast, the results for the GTU model (large mesh) show that solver scalability is above optimal. The number of cells per core in this case never drops below 100,000, even in 16-node configurations. These results are encouraging because most real-life CFD external vehicle aerodynamic models have 100 million cells or more.

Azure cost

Only solver running time is considered for these cost calculations. Meshing times, installation time, and software costs aren't considered.

You can use the [Azure pricing calculator](#) to estimate the VM costs for your configurations.

The following tables provide the solver running times in hours. Azure VM hourly rates are subject to change. To compute the cost, multiply the solver time by the number of nodes and the [Linux VM hourly cost](#).

Cost for model 1: Automotive_DESdrivAer

grid [Expand table](#)

Number of nodes	Solver running time (hours)
1	0.458
2	0.256
4	0.168
8	0.148
16	0.162

Cost for model 2: Automotive_GTU-0001

Number of nodes	Solver running time (hours)
1	28.888
2	13.622
4	6.249
8	2.990
16	1.761

Summary

- ELEMENTS 3.5.0 was successfully tested on HBv3 standalone VMs and on an Azure CycleCloud multi-node configuration.
- All external vehicle aerodynamics models that were tested demonstrate good CPU acceleration when running in multi-node configurations.
- The meshing, setup, and solver applications in ELEMENTS can be run in parallel, which makes it ideal for running in multi-node configurations. (There's no need for mesh decomposition and reconstruction.)
- The simulation engine that's provided with ELEMENTS is open source, so you can run as many simulations in as many processors as you need, without incurring additional license costs. This capability is particularly useful when you're performing DES-type external aerodynamic calculations.
- For better parallel performance when you run DES-type calculations by using ELEMENTS, we recommend that you use 64 cores per HBv3 node and a minimum of 50,000 cells per core.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director, Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run HPC applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Engys HELYX on an Azure virtual machine

Azure Virtual Machines Azure Virtual Network Azure CycleCloud

This article describes the steps for running [Engys HELYX](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running HELYX on single-node and multi-node VM configurations.

HELYX is a general purpose computational fluid dynamics (CFD) application for engineering analysis and design optimization. It's based on OpenFOAM libraries and uses an advanced automatic meshing utility to simulate complex flows.

HELYX provides:

- A Generalized Internal Boundaries (GIB) method to support complex boundary motions inside the finite-volume mesh.
- An advanced hex-dominant automatic mesh algorithm with polyhedra support that can run in parallel to generate large computational grids.
- A solver stack that's based on the finite-volume approach. It covers single-phase and multi-phase turbulent flows (RANS, URANS, DES, LES), thermal flows with natural/forced convection, thermal/solar radiation, and incompressible and compressible flow solutions.

HELYX supports industry-specific add-ons, including hydrodynamics analysis for marine applications, block-coupled incompressible flow solvers, and advanced two-phase volume-of-fluid (VOF) flows.

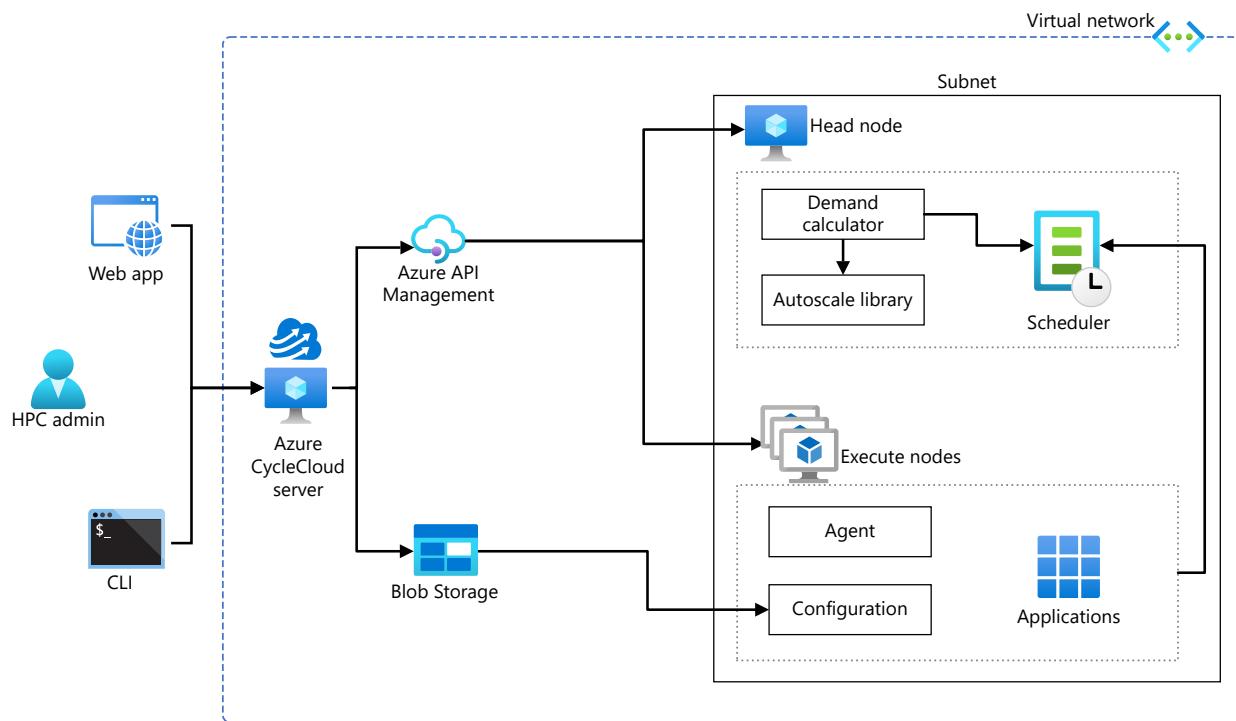
HELYX is used in the automotive, aerospace, construction, marine, turbo, and energy industries.

Why deploy HELYX on Azure?

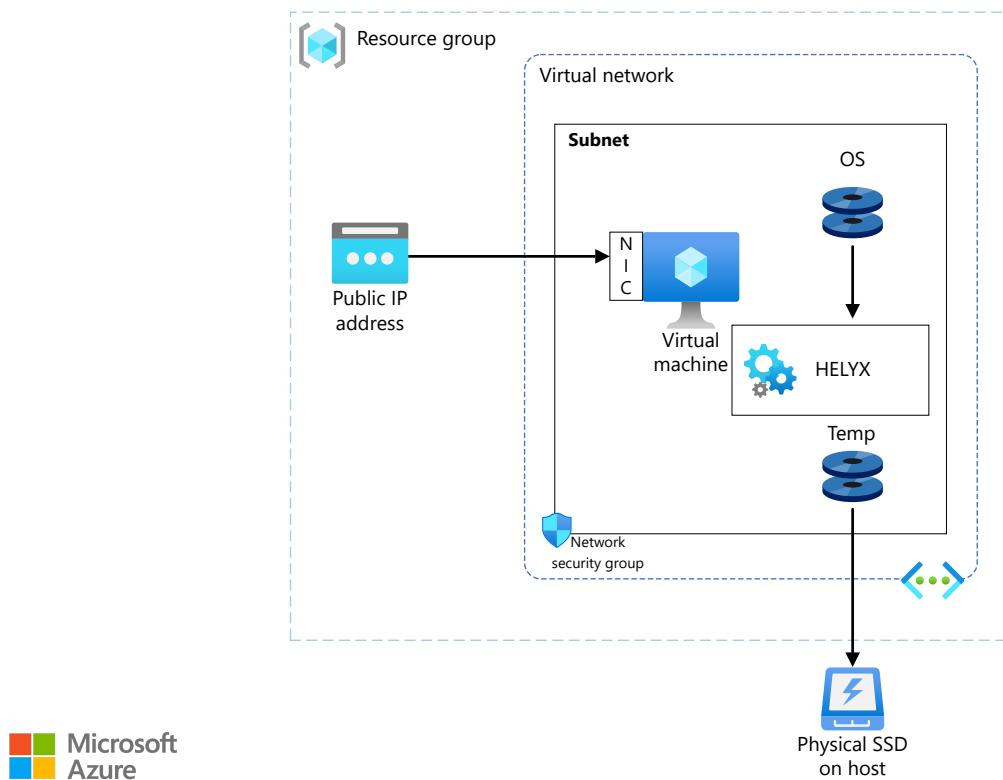
- Modern and diverse compute options to meet your workload's needs.
- The flexibility of virtualization without the need to buy and maintain physical hardware.
- Rapid provisioning.
- Complex problems solved within a few hours.

Architecture

Multi-node configuration:



Single-node configuration:



Download a [Visio file](#) of all diagrams in this article.

Components

- Azure Virtual Machines [is used to create Linux VMs. For information about deploying the VM and installing the drivers, see \[Linux VMs on Azure\]\(#\).](#)
- Azure Virtual Network [is used to create a private network infrastructure in the cloud.](#)
 - Network security groups [are used to restrict access to the VMs.](#)
 - A public IP address connects the internet to the VM.
- Azure CycleCloud [is used to create the cluster in the multi-node configuration.](#)
- A physical SSD is used for storage.

Compute sizing and drivers

Performance tests of HELYX on Azure used HBv3 AMD EPYC 7V73X (Milan-X) VMs running Linux CentOS. The following table provides details about HBv3-series VMs.

[Expand table](#)

VM size	vCPU	Memory (GiB)	Memory bandwidth (Gbps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120rs_v3	120	448	350	1.9	3.0	3.5	200	32
Standard_HB120-96rs_v3	96	448	350	1.9	3.0	3.5	200	32
Standard_HB120-64rs_v3	64	448	350	1.9	3.0	3.5	200	32

VM size	vCPU	Memory (GiB)	Memory bandwidth (Gbps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120-32rs_v3	32	448	350	1.9	3.0	3.5	200	32
Standard_HB120-16rs_v3	16	448	350	1.9	3.0	3.5	200	32

Required drivers

To use InfiniBand, you need to enable InfiniBand drivers.

Install HELYX 3.5.0 on a VM or HPC cluster

You need to buy HELYX from ENGYS or one of its local authorized distributors or agents to get access to the installation files and technical support. For information about buying HELYX, contact [ENGYS](#).

Before you install HELYX, you need to deploy and connect a VM or HPC cluster.

For information about deploying the VM and installing the drivers, see [Run a Linux VM on Azure](#).

For information about deploying Azure CycleCloud and an HPC cluster, see these articles:

- [Install and configure Azure CycleCloud](#)
- [Create an HPC cluster](#)

HELYX 3.5.0 performance results

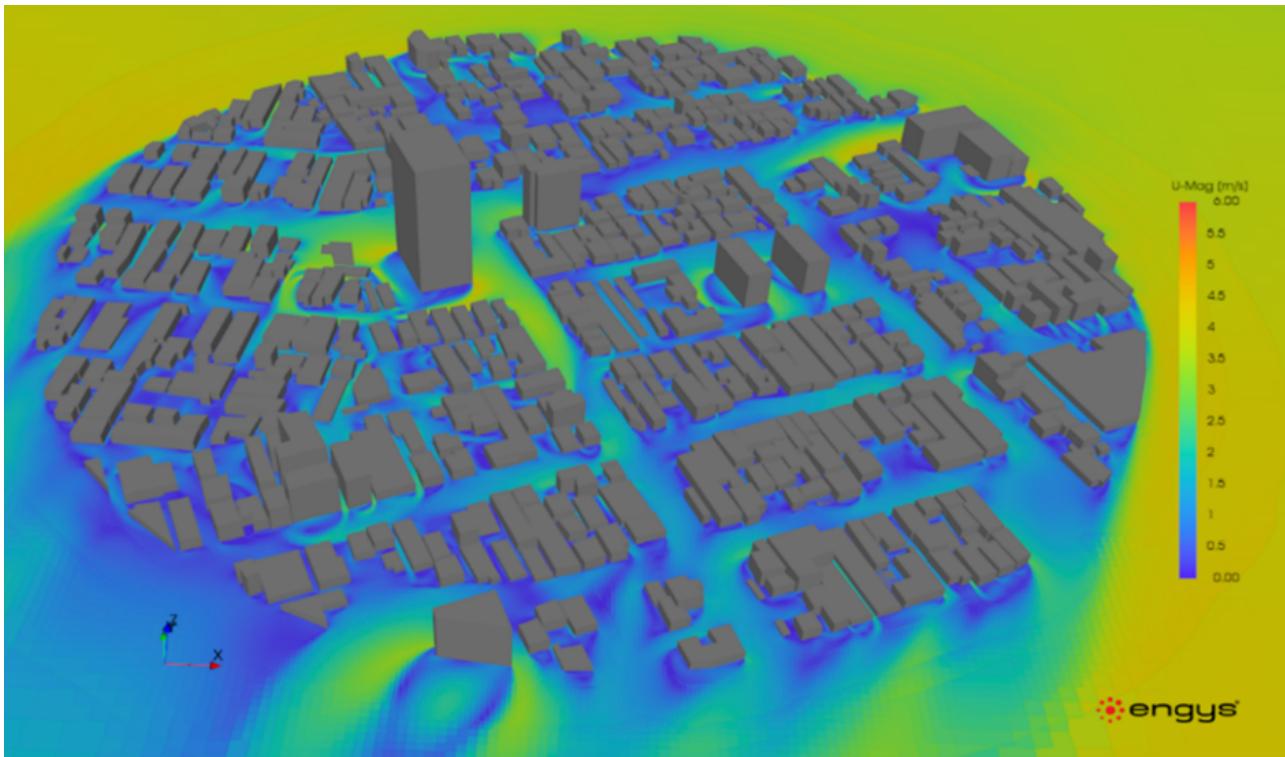
Three models are used to test the parallel scalability performance of HELYX version 3.5.0 on Azure:

- A steady-state model of a city landscape, typical of wind comfort analysis.
- A steady-state model of a ventilator fan with moving blades, approximated via an MRF approach and an arbitrary mesh interface (AMI). Two mesh densities are compared.
- A transient model of a ship moving in calm water. A two-phase VOF solver is used. Two mesh densities are compared.

All computational grids tested were created in parallel as part of the execution process. The hex-dominant meshing utility that's provided with HELYX was used for the tests.

The details of each test model are provided in the following sections.

Model 1: City_landscape_Niigata-NNE

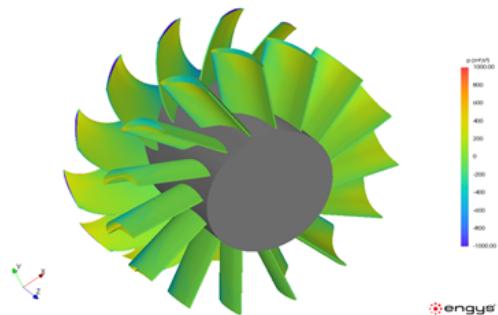


The following table provides details about the model.

[Expand table](#)

Model number	Mesh size	Solver	Steady state
1	26,500,000 cells	Single phase, turbulent flow	1,000 iterations

Model 2: Turbomachine_Ventilator-AFnq182

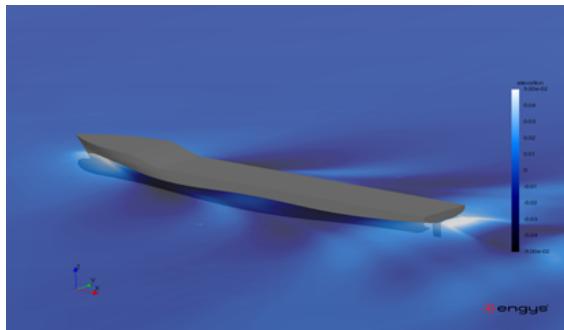


The following table provides details about the model.

[Expand table](#)

Model number	Mesh size	Solver	Steady state
2a	3,100,000 cells	Single phase, turbulent flow with MRF (AMI)	1,000 iterations
2b	11,800,000 cells	Single phase, turbulent flow with MRF (AMI)	1,000 iterations

Model 3: Marine_G2010-C2.2b-KCS-Fn026



The following table provides details about the model.

[grid icon] [Expand table](#)

Model number	Mesh size	Solver	Steady state
3a	1,350,000 cells	Two-phase VOF with automatic mesh refinement	CFL regulated for 20 seconds
3b	11,100,000 cells	Two-phase VOF with automatic mesh refinement	CFL regulated for 20 seconds

HELYX 3.5.0 performance results on single-node VMs

The following sections provide the performance results of running HELYX in parallel on single-node Azure [HBv3 AMD EPYC 7V73X \(Milan-X\)](#) VMs. You can use these results as a baseline for comparison with multi-node runs.

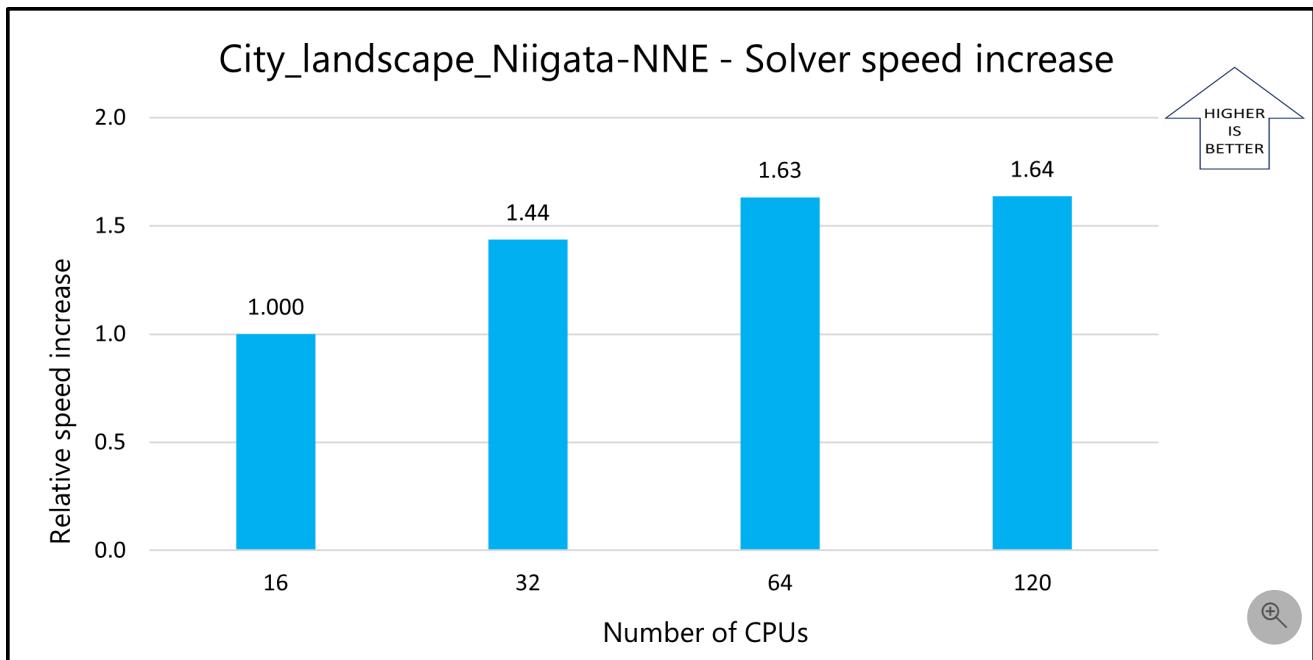
Model 1: City_landscape_Niigata-NNE

This table shows total elapsed solver running times recorded for varying numbers of CPUs on the Standard HBv3-series VM:

[grid icon] [Expand table](#)

Number of cores	Solver running time (seconds)	Relative speed increase
16	6,176.48	1.00
32	4,301.36	1.44
64	3,783.12	1.63
120	3,774.44	1.64

The following graph shows the relative speed increases as the number of CPUs increases:



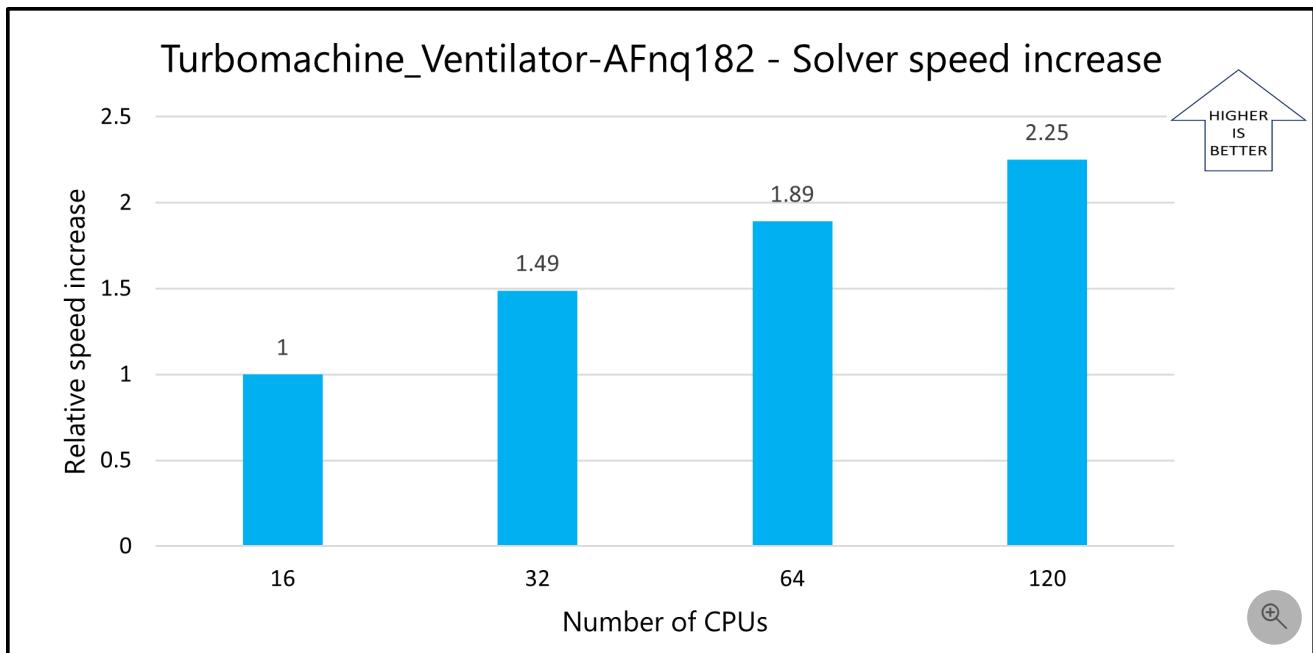
Model 2a: Turbomachine_Ventilator-AFnq182

This table shows total elapsed solver running times recorded for varying numbers of CPUs on the Standard HBv3-series VM:

[Expand table](#)

Number of cores	Solver running time (seconds)	Relative speed increase
16	1,609.85	1.00
32	1,081.2	1.49
64	850.98	1.89
120	715.07	2.25

The following graph shows the relative speed increases as the number of CPUs increases:



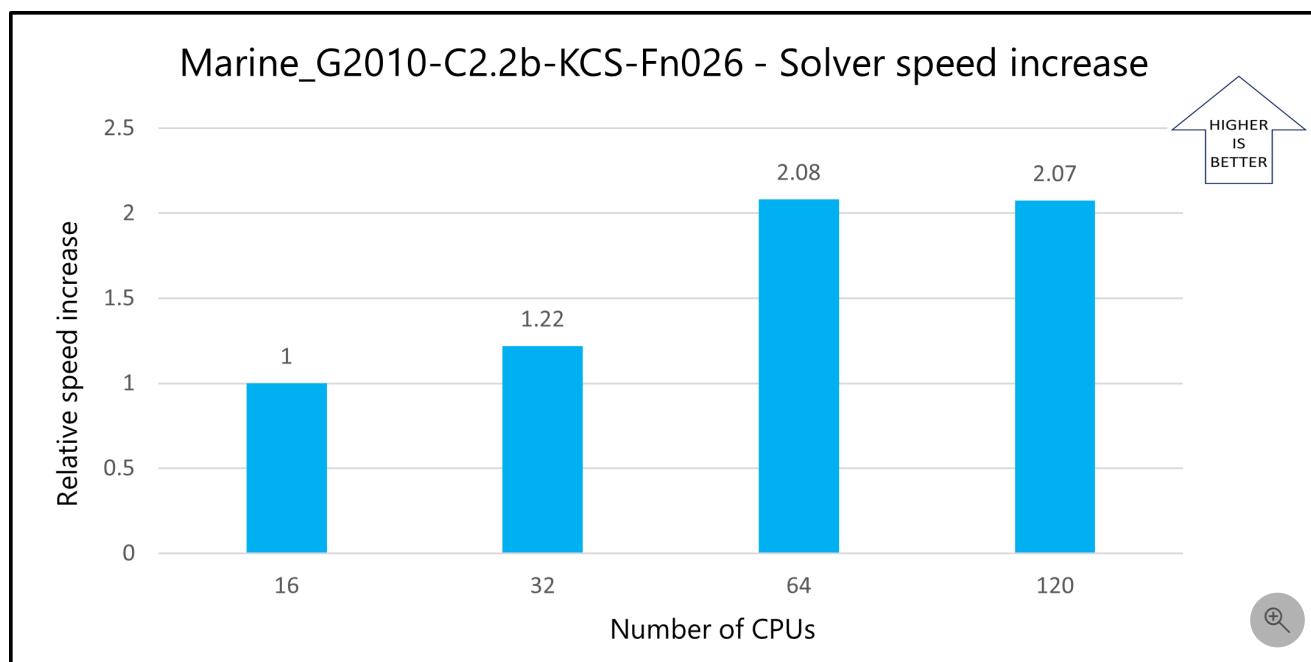
Model 3a: Marine_G2010-C2.2b-KCS-Fn026

This table shows total elapsed solver running times recorded for varying numbers of CPUs on the Standard HBv3-series VM:

[Expand table](#)

Number of cores	Solver running time (seconds)	Relative speed increase
16	16,608.29	1.00
32	13,622.88	1.22
64	7,979.05	2.08
120	8,007.08	2.07

The following graph shows the relative speed increases as the number of CPUs increases:



Notes about the single-node tests

For all the single-node tests, the solver time on a Standard_HB120-16rs_v3 VM (16 cores) is used as a reference to calculate the relative speed increase with respect to similar VMs that have more cores. The previously presented results show that parallel performance improves as cores increase from 16 to 64. At 120 cores, some simulations show limited improvement and others show a drop in performance. This pattern is common with CFD solvers and other memory-intensive applications because of saturation of the onboard memory that's available on each processor.

The AMD EPYC 7V73-series processor (Milan-X) in the HBv3 VMs tested here is a powerful processor, with 768 MB of total L3 cache. The single-node tests confirm that this memory is sufficient to guarantee parallel scalability of the HELYX solvers when you use half the cores available on each 7V73-series chip.

HELYX 3.5.0 performance results on multi-node clusters

The single-node tests confirm that the solver achieves parallel performance until you reach 64 cores on HBv3 VMs. Based on those results, only 64-core configurations on [Standard_HB120-64rs_v3](#) VMs were used to evaluate the performance of HELYX on multi-node clusters. The following sections provide the test results.

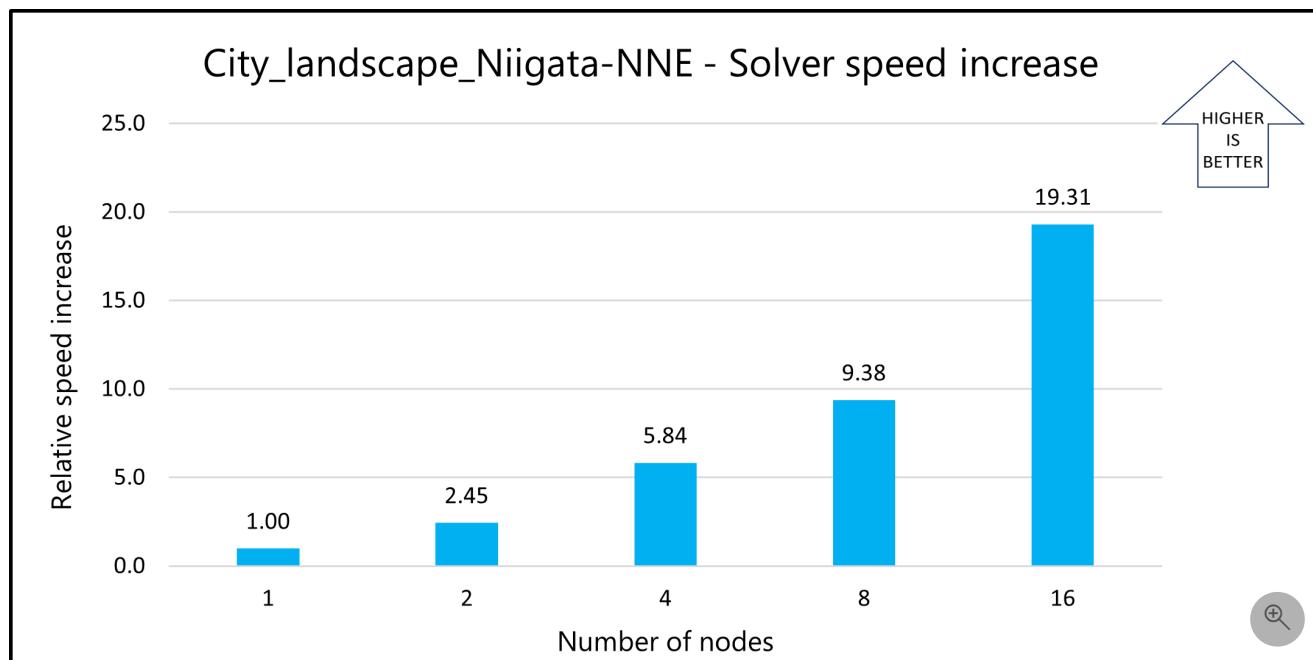
Model 1: City_landscape_Niigata-NNE

This table shows the total elapsed solver running times recorded for varying numbers of CPUs on Standard HBv3-series VMs:

[Expand table](#)

Number of nodes	Number of cores	Cells per core	Solver running time (seconds)	Relative speed increase
1	64	414,063	3,741.59	1.00
2	128	207,031	1,528.34	2.45
4	256	103,516	640.64	5.84
8	512	51,758	398.73	9.38
16	1,024	25,879	193.72	19.31

The following graph shows the relative speed increase as the number of cores increases:



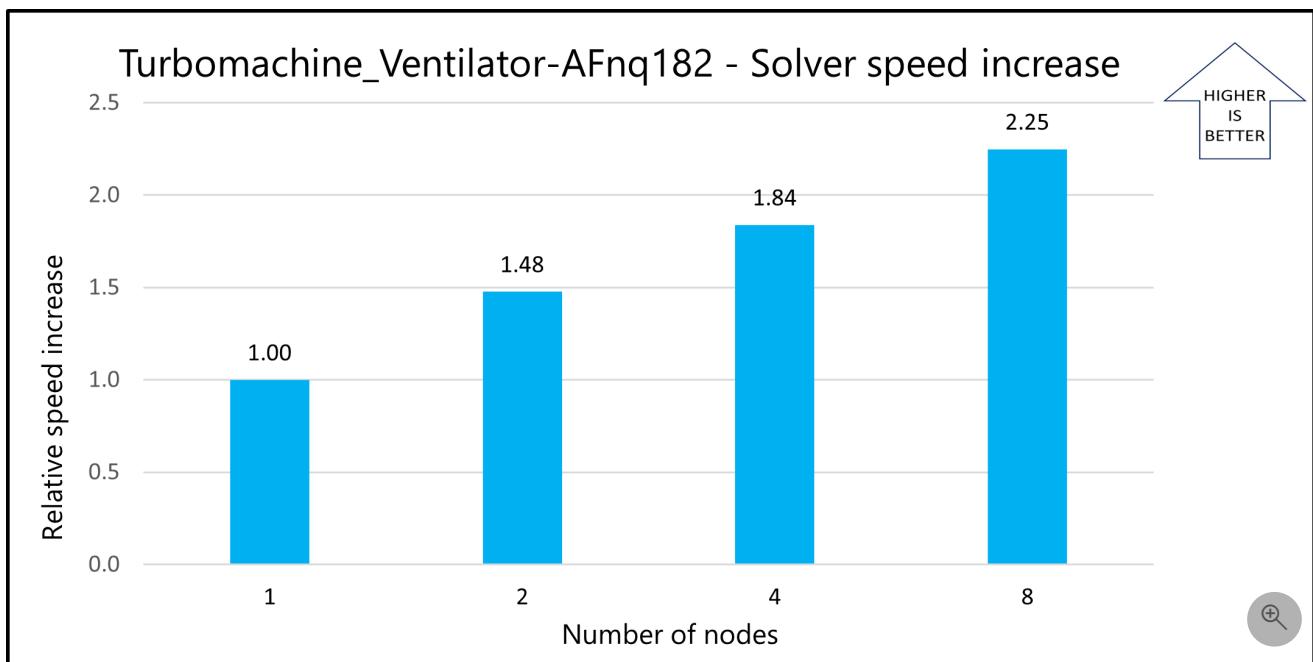
Model 2a: Turbomachine_Ventilator-AFnq182

This table shows the total elapsed solver running times recorded for varying numbers of CPUs on Standard HBv3-series VMs:

[Expand table](#)

Number of nodes	Number of cores	Cells per core	Solver running time (seconds)	Relative speed increase
1	64	48,438	838.4	1.00
2	128	24,219	567.48	1.48
4	256	12,109	455.9	1.84
8	512	6,055	372.82	2.25

The following graph shows the relative speed increase as the number of cores increases:



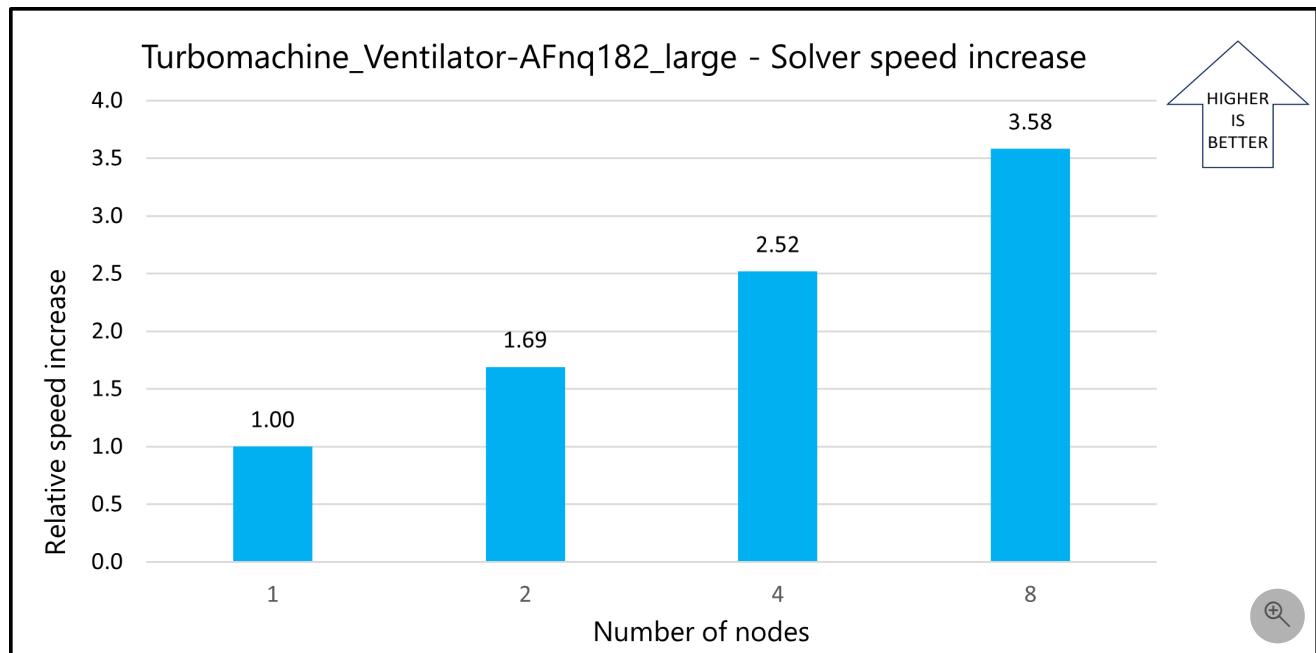
Model 2b: Turbomachine_Ventilator-AFnq182_large

This table shows the total elapsed solver running times recorded for varying numbers of CPUs on Standard HBv3-series VMs:

[Expand table](#)

Number of nodes	Number of cores	Cells per core	Solver running time (seconds)	Relative speed increase
1	64	184,375	2,710.14	1.00
2	128	92,188	1,602.64	1.69
4	256	46,094	1,076.27	2.52
8	512	23,047	756.73	3.58

The following graph shows the relative speed increase as the number of cores increases:



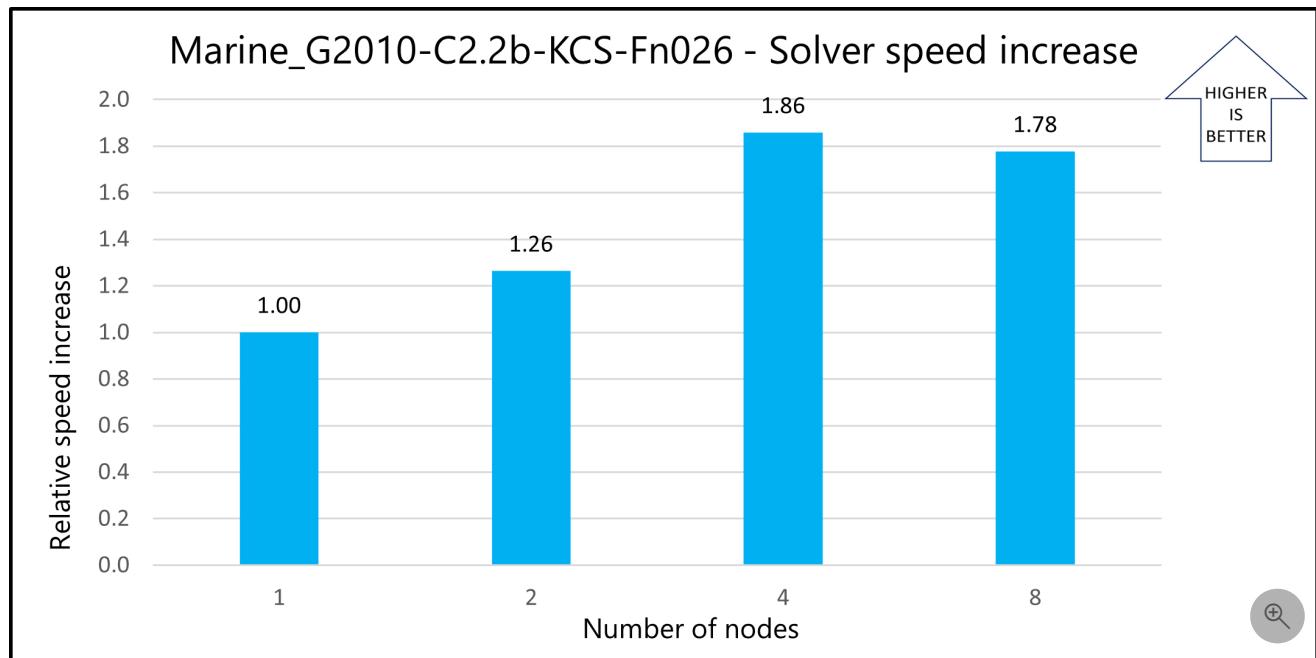
Model 3a: Marine_G2010-C2.2b-KCS-Fn026

This table shows the total elapsed solver running times recorded for varying numbers of CPUs on Standard HBv3-series VMs:

[Expand table](#)

Number of nodes	Number of cores	Cells per cores	Solver running time (seconds)	Relative speed increase
1	64	21,094	8,028.75	1.00
2	128	10,547	6,354.25	1.26
4	256	5,273	4,320.72	1.86
8	512	2,637	4,518.09	1.78

The following graph shows the relative speed increase as the number of cores increases:



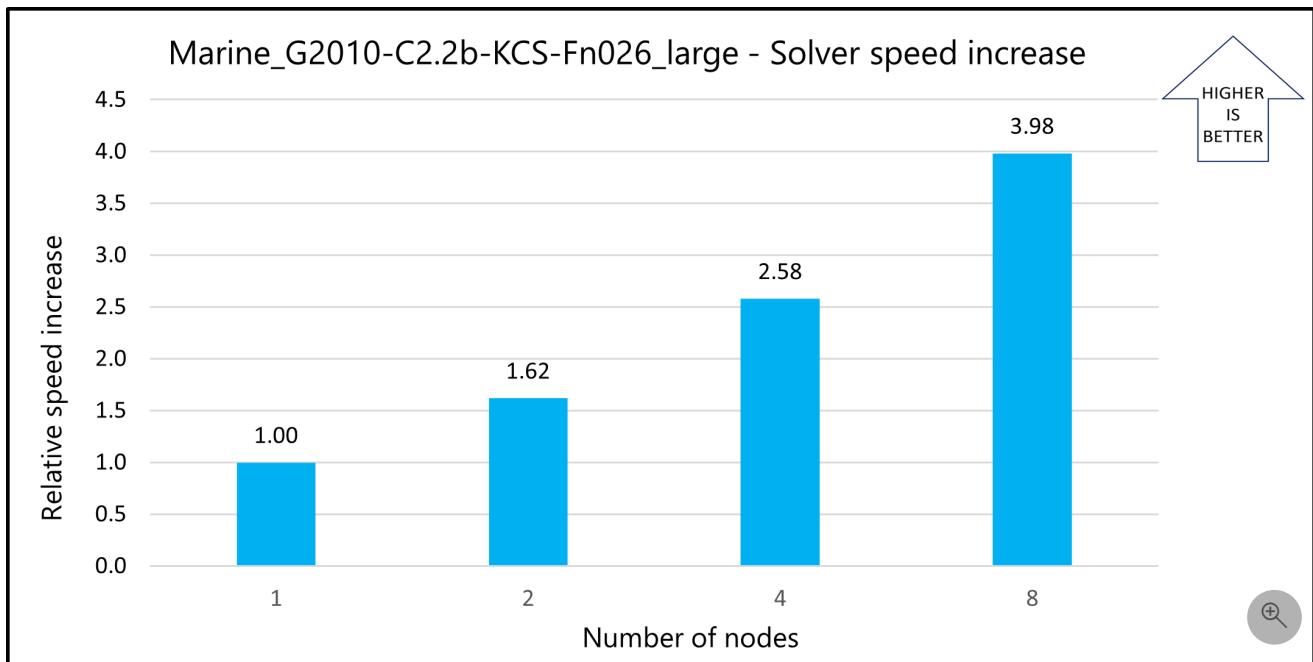
Model 3b: Marine_G2010-C2.2b-KCS-Fn026_large

This table shows the total elapsed solver running times recorded for varying numbers of CPUs on Standard HBv3-series VMs:

[Expand table](#)

Number of nodes	Number of cores	Cells per core	Solver running time (seconds)	Relative speed increase
1	64	173,438	66,860.55	1.00
2	128	86,719	41,243.12	1.62
4	256	43,359	25,901.95	2.58
8	512	21,680	16,781.86	3.98

The following graph shows the relative speed increase as the number of cores increases:



Notes about the multi-node tests

Based on the multi-node tests, parallel scalability for model 1 (steady state, incompressible, turbulent flow) is above optimal. The results of the model 2 and 3 tests show that parallel solver performance can be influenced by ancillary methods like MRF/AMI or automatic mesh refinement.

The results also show that a minimum number of cells per core is required to reach optimal scalability across multiple nodes. You can see this by comparing the model 2a results to those of 2b and the model 3a results to those of 3b. Solver performance is reduced when the number of cells per core falls below 20,000 because of excessive data communication between processor boundaries.

Azure cost

Only elapsed solver running time is considered for these cost calculations. Meshing times, installation time, and software costs aren't considered.

You can use the [Azure pricing calculator](#) to estimate the VM costs for your configurations.

The following tables provide the elapsed solver running times in hours. Azure VM hourly rates are subject to change. To compute the cost, multiply the solver running time by the number of nodes and the [Linux VM hourly cost](#).

Cost for model 1: City_landscape_Niigata-NNE

[Expand table](#)

Number of nodes	Solver running time (hours)
1	1.052
2	0.436
4	0.186
8	0.118
16	0.062

Cost for model 2a: Turbomachine_Ventilator-AFnq182

[Expand table](#)

Number of nodes	Solver time (hours)
1	0.244
2	0.168
4	0.138
8	0.118

Cost for model 2b: Turbomachine_Ventilator-AFnq182_large

[Expand table](#)

Number of nodes	Solver time (hours)
1	0.801
2	0.483
4	0.337
8	0.247

Cost for model 3a: Marine_G2010-C2.2b-KCS-Fn026

[Expand table](#)

Number of nodes	Solver time (hours)
1	2.291
2	1.823
4	1.264
8	1.336

Cost for model 3b: Marine_G2010-C2.2b-KCS-Fn026_large

[Expand table](#)

Number of nodes	Solver time (hours)
1	18.800
2	11.670
4	7.406

Number of nodes	Solver time (hours)
8	4.890

Summary

- HELYX 3.5.0 was successfully tested on HBv3 standalone VMs and on an Azure CycleCloud multi-node configuration.
- All tested models demonstrated good CPU acceleration in a multi-node configuration.
- The meshing, setup, and solver applications in HELYX can all be run in parallel, which makes it ideal for running in multi-node configurations. (There's no need for mesh decomposition and reconstruction.)
- The simulation engine delivered with HELYX is open source, so you can run as many simulations as you need, on as many processors as you need, without incurring additional license costs.
- For better parallel performance, we recommend that you use 64 cores per HBv3 node and a minimum of 20,000 cells per core.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Preetham Y M](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run HPC applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy BETA CAE EPILYSIS on an Azure virtual machine

Azure Virtual Machines

Azure Virtual Network

This article presents the performance results of running the [BETA CAE EPILYSIS](#) application on an Azure virtual machine (VM). EPILYSIS is a software program that's used to perform several types of finite element analysis on various structures and materials.

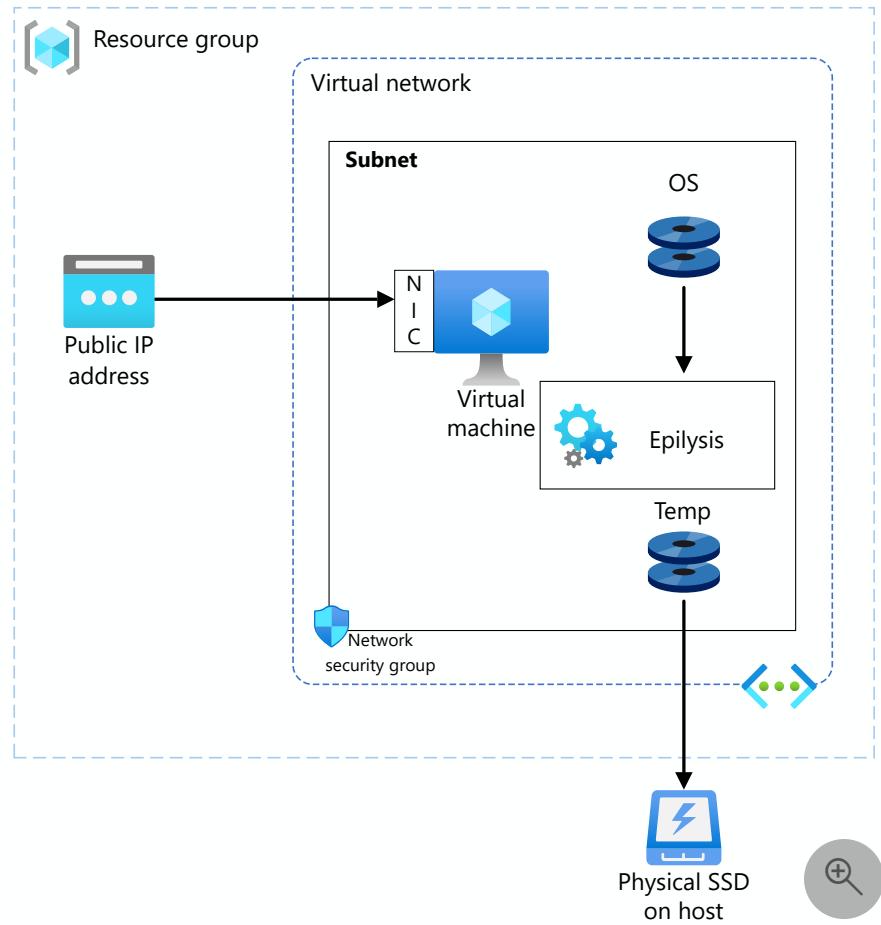
EPILYSIS is used in the aerospace, automotive, defense, high-tech, and industrial equipment industries. Engineers use EPILYSIS to help design and optimize products. EPILYSIS can be combined with other tools, such as ANSA and META, to optimize simulations. The solver covers various solution types, like structural, NVH (noise, vibration, and harshness), optimization, and more.

Why deploy EPILYSIS on Azure?

Deploy EPILYSIS on Azure to get benefits like:

- Modern and diverse compute options to meet your workload's needs.
- The flexibility of virtualization without the need to buy and maintain physical hardware.
- Rapid provisioning.
- Performance that scales as CPUs are added.

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Linux VM. For information about deploying a VM and installing drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
- [Network security groups](#) are used to restrict access to the VM.
- A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

The performance tests of EPILYSIS on Azure used an [HBv3-series](#) VM and [EadsV5-series](#) VM running on a Linux operating system.

HBv3-series VMs are optimized for HPC applications, like fluid dynamics, explicit and implicit finite element analysis, weather modeling, seismic processing, reservoir simulation, and RTL simulation.

Eadsv5-series VMs are optimized for memory-intensive enterprise applications, such as relational database servers and in-memory analytics workloads.

The following table provides the configuration details for [HBv3-series](#) and [Eadsv5-series](#) VMs:

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

VM series	VM size	vCPU	Memory (GiB)	Temp storage (GiB)	Max data disk	Max NICs
HBv3	Standard_HB120-16rs_v3	16	448	2*960	32	8
HBv3	Standard_HB120-32rs_v3	32	448	2*960	32	8
Eadsv5	Standard_E64ads_v5 ¹	64	512	2400	32	8

¹ Constrained core sizes available

EPILYSIS installation

Before you install EPILYSIS, you need to [deploy and connect a Linux VM](#). Then you can [download and install EPILYSIS](#).

EPILYSIS performance results

EPILYSIS version 23.1.1 was used for testing the *101_large_7million model*.

The following table provides details about the computing environment that was used for testing.

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

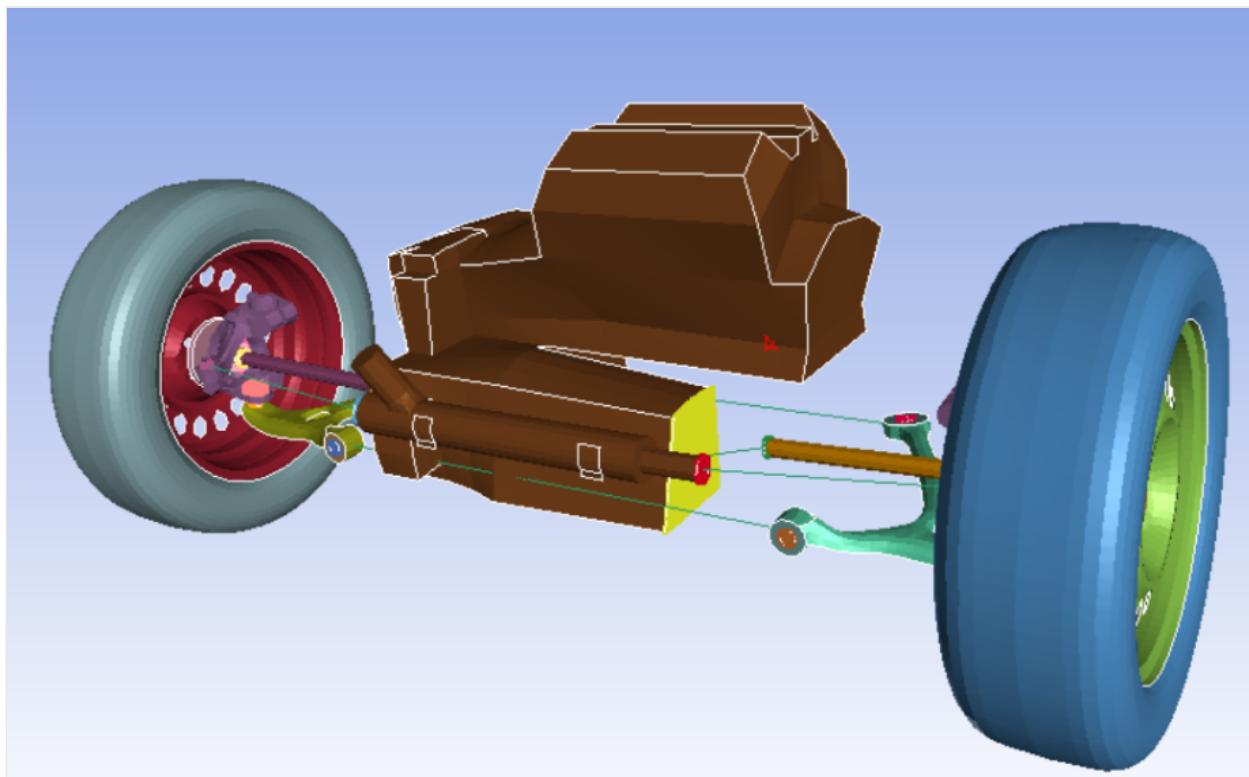
VM series	Operating system version	Operating system architecture	Processor	MPI
Eadsv5	Linux CentOS 7.9 HPC Gen2	x86-64	AMD EPYC 7763v	Open MPI 4.1.1
HBv3	Linux CentOS 7.9 HPC Gen2	x86-64	AMD EPYC 7V73X	Open MPI 4.1.1

ⓘ Important

The version of Linux discussed in this article will be discontinued in 2024. Tests that are performed on newer versions of Linux that include the same drivers are expected to produce similar results.

Model details

The 101_large_7million model is used for EPILYSIS solver validation. This model is set up for a linear static analysis (solution 101). A *linear static analysis* is an analysis where a linear relation holds between applied forces and displacements. The following image shows the 101_large_7million model.



The following table provides details about the model.

[Expand table](#)

Model name	Nodes	Shell elements	Solid elements
101_large_7million.nas	7678688	6341470	1295109

EPILYSIS performance results on a HBv3-series VM

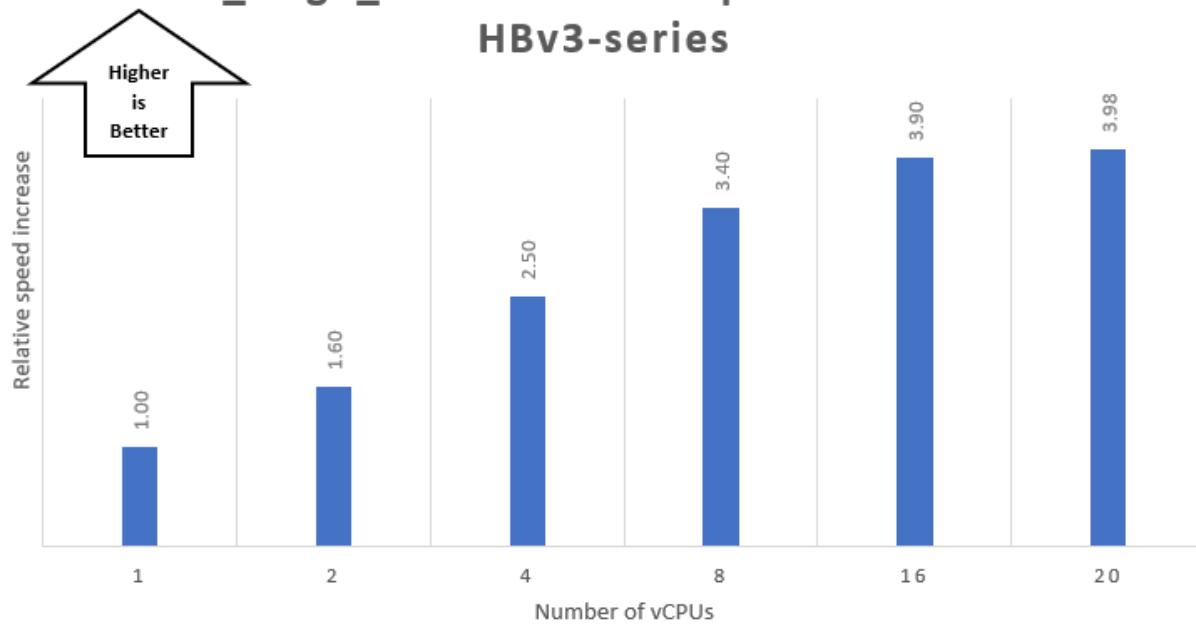
The following table shows the details of each test on a HBv3-series VM. As the number of vCPUs increases, the total elapsed time decreases, and the relative speed increase improves.

[Expand table](#)

VM size	Number of vCPUs available	Number of vCPUs used	Total elapsed time (seconds)	Relative speed increase
Standard_HB120-16rs_v3	16	1	5542	1.00
Standard_HB120-16rs_v3	16	2	3469	1.60
Standard_HB120-16rs_v3	16	4	2213	2.50
Standard_HB120-16rs_v3	16	8	1630	3.40
Standard_HB120-16rs_v3	16	16	1420	3.90
Standard_HB120-32rs_v3	32	20	1393	3.98

The following graph shows how the relative speed increase improves as you increase the vCPUs. It begins to plateau at 16 vCPUs.

101_large_7million model performance on HBv3-series



EPILYSIS performance results on EadsV5-series VMs

The following table illustrates that as the number of vCPUs used increases, the total elapsed time in seconds decreases, and the relative speed increase improves significantly. There's a strong correlation between the vCPUs used and the efficiency of the process.

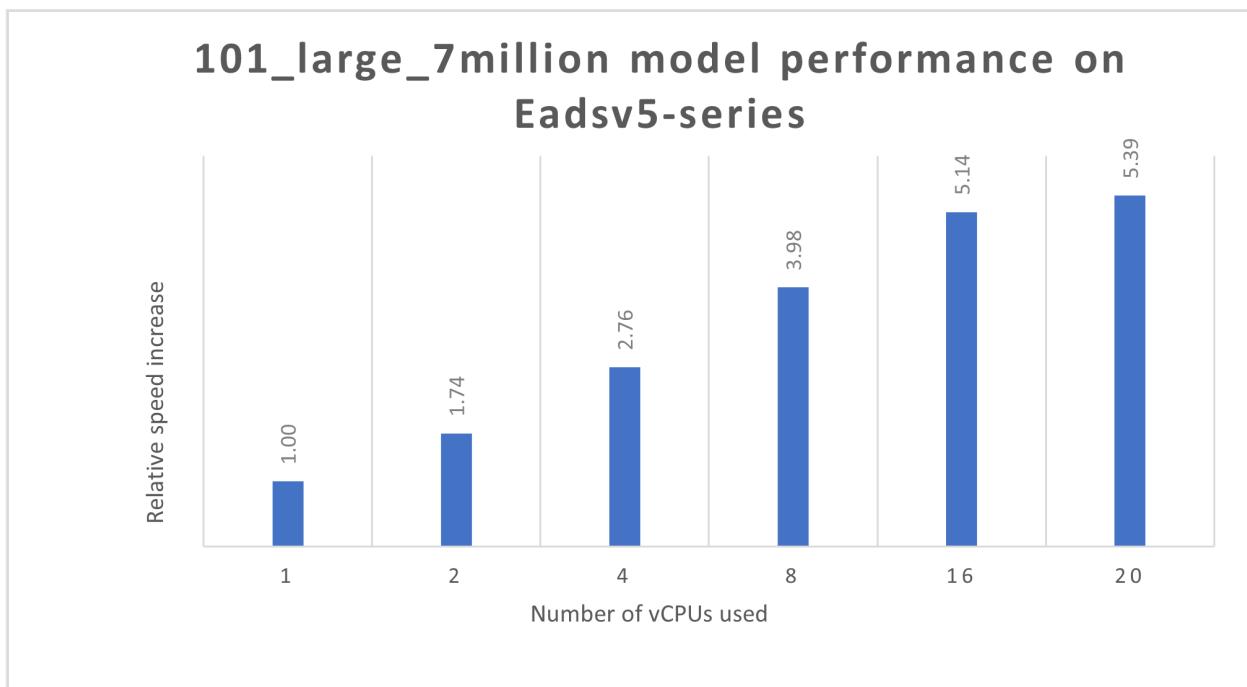
[\[+\] Expand table](#)

VM size	Number of vCPUs available	Number of vCPUs used	Total elapsed time (seconds)	Relative speed increase
Standard_E64ads_v5 ¹	64	1	6049	1.00
Standard_E64ads_v5 ¹	64	2	3480	1.74
Standard_E64ads_v5 ¹	64	4	2195	2.76
Standard_E64ads_v5 ¹	64	8	1520	3.98
Standard_E64ads_v5 ¹	64	16	1177	5.14

VM size	Number of vCPUs available	Number of vCPUs used	Total elapsed time (seconds)	Relative speed increase
Standard_E64ads_v5 ¹	64	20	1122	5.39

¹ Constrained core sizes available

The following graph shows how the relative speed increase improves as you increase the vCPUs.



Azure cost

The following table provides estimated runtimes that you can use to calculate Azure costs. To compute the cost, multiply the estimated time by the Azure VM hourly rate. For the hourly rates for Linux, see [Linux VMs pricing](#). Azure VM hourly rates are subject to change.

The cost calculations only factor in the simulation runtime. The installation time, simulation setup time, and software costs aren't included. You can use the [Azure pricing calculator](#) to estimate VM costs for your configurations.

Expand table

VM size	vCPUs used	Elapsed time (hours)
Standard_HB120-16rs_v3	1	1.54
Standard_HB120-16rs_v3	2	0.96
Standard_HB120-16rs_v3	4	0.61
Standard_HB120-16rs_v3	8	0.45
Standard_HB120-16rs_v3	16	0.39
Standard_HB120-32rs_v3	20	0.39
Standard_E64ads_v5 ¹	1	1.68
Standard_E64ads_v5 ¹	2	0.97
Standard_E64ads_v5 ¹	4	0.61
Standard_E64ads_v5 ¹	8	0.42
Standard_E64ads_v5 ¹	16	0.33
Standard_E64ads_v5 ¹	20	0.31

¹ [Constrained core sizes available](#)

Summary

- The HBv3-series and Eadsv5-series VMs on Azure were used to create a benchmarking suite, which is one of the many uses of EPILYSIS.
- EPILYSIS's performance was evaluated on two HBv3-series VMs (Standard_HB120-16rs_v3 and Standard_HB120-32rs_v3) and one Eadsv5-series VM (Standard_E64ads_v5).
- On the HBv3-series VM, there's a 400% performance improvement when the vCPU count is increased to 20 vCPUs. A single vCPU is used as a baseline.

- Similarly, on the Eadsv5-series VM, there's a 500% performance improvement when the vCPU count is increased to 20 vCPUs. A single vCPU is used as a baseline.
- According to a validation study, the performance of a Eadsv5-series VM with 20 vCPUs is 19% more efficient compared to a HBv3-series VM with the same number of vCPUs.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Rupali Konade](#) | HPC Performance Engineer
- [Shivakumar Tallolli](#) | HPC Performance Engineer

Other contributors:

- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy GROMACS on a virtual machine

Azure Virtual Machines Azure Virtual Network

This article briefly describes the steps for running [GROMACS](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running GROMACS on Azure.

GROMACS (GROningen MACHine for Simulations) is a molecular dynamics package designed for simulations of proteins, lipids, and nucleic acids. It's used primarily for dynamic simulations of biomolecules and provides a rich set of calculation types and preparation and analysis tools. GROMACS:

- Supports compressed trajectory storage format and advanced techniques for free-energy calculations.
- Runs multiple simulations as part of a single program, which enables generalized ensemble methods like replica-exchange.
- Works within an elaborate multi-level parallelism that distributes computational work across ensembles of simulations and multiple program paths.
- Describes all systems with triclinic unit cells, so complex geometries like rhombic dodecahedron, truncated octahedron, and hexagonal boxes are supported.

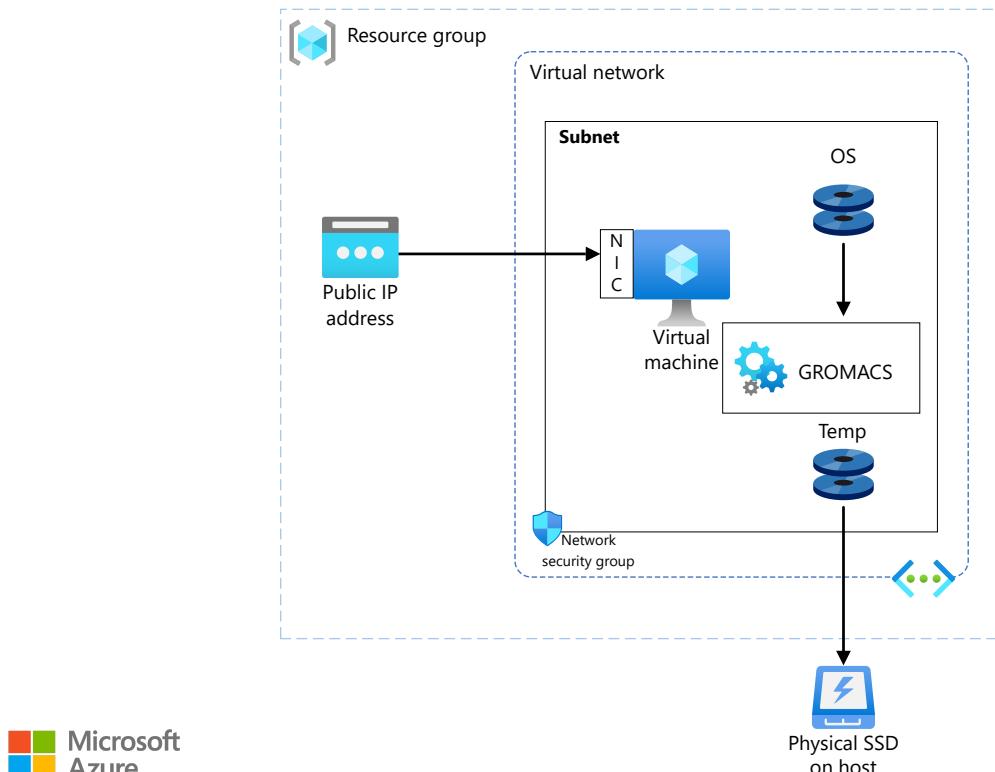
GROMACS is used across the healthcare industry by biotechnology organizations, universities and research centers, education, pharmaceutical organizations, and hospitals and clinics.

Why deploy GROMACS on Azure?

- Modern and diverse compute options to align to your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- With a 120 vCPU, a performance increase of three to four times that of 16 CPUs

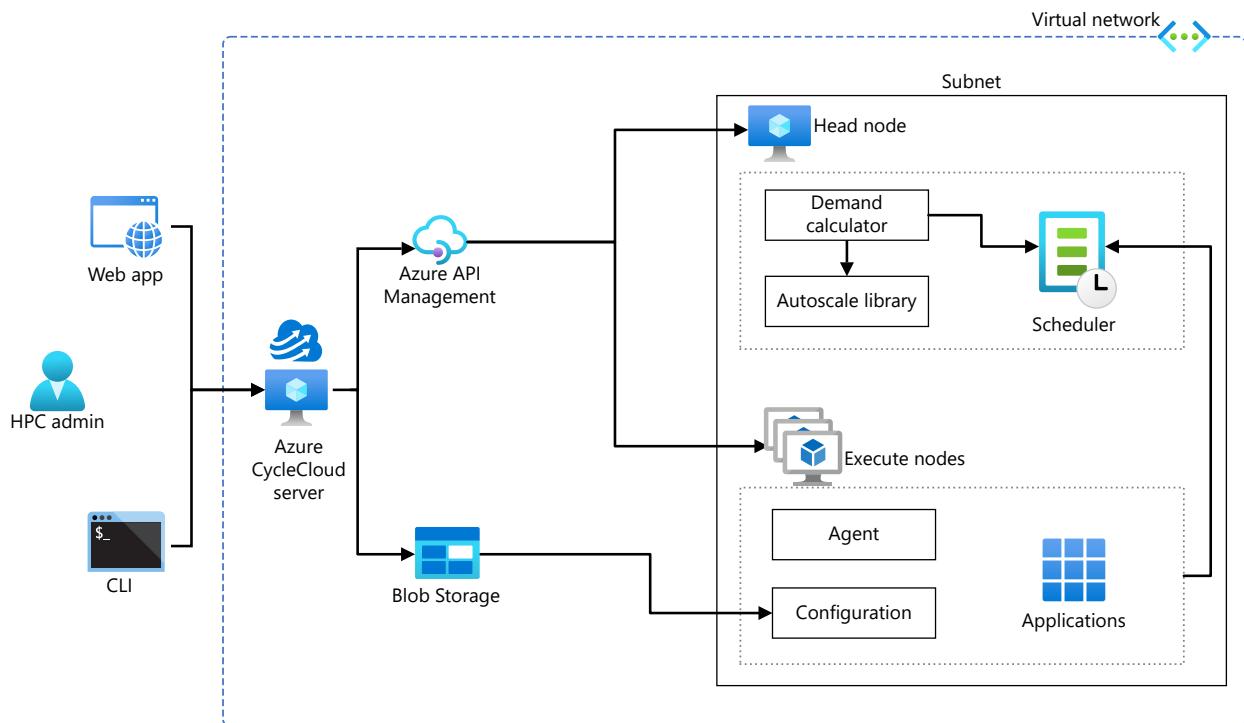
Architecture

This diagram shows GROMACS running on a single Azure VM:



Download a [Visio file](#) of this architecture.

This diagram shows a multi-node configuration, orchestrated with Azure CycleCloud:



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Linux VM. For information about deploying the VM and installing the drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- [Azure CycleCloud](#) is used to create the cluster in the multi-node configuration.
- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

Performance tests of GROMACS on Azure used [HBv3-series](#) VMs running the Linux Ubuntu operating system. The following table provides details about HBv3-series VMs.

[Expand table](#)

VM size	vCPU	Ram memory (GiB)	Memory bandwidth (Gbps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120rs_v3	120	448	350	1.9	3.0	3.5	200	32
Standard_HB120-96rs_v3	96	448	350	1.9	3.0	3.5	200	32
Standard_HB120-64rs_v3	64	448	350	1.9	3.0	3.5	200	32

VM size	vCPU	Ram memory (GiB)	Memory bandwidth (Gbps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120-32rs_v3	32	448	350	1.9	3.0	3.5	200	32
Standard_HB120-16rs_v3	16	448	350	1.9	3.0	3.5	200	32

HBv3-series VMs are optimized for HPC applications like fluid dynamics, explicit and implicit finite element analysis, weather modeling, seismic processing, reservoir simulation, and RTL simulation.

Required drivers

To use the AMD CPUs on [HBv3-series](#) VMs, you need to install AMD drivers.

To use InfiniBand, you need to enable InfiniBand drivers.

GROMACS installation

Install GROMACS on a virtual machine

Before you install GROMACS, you need to deploy and connect a Linux VM and install the required AMD and InfiniBand drivers.

For information about deploying the VM and installing the drivers, see [Run a Linux VM on Azure](#).

You can download GROMACS from the [GROMACS download](#) page. For information about installing the application, see the [Installation guide](#).

For information about using GROMACS, see the [User guide](#).

Install GROMACS on an HPC cluster

You can easily deploy an HPC cluster on Azure by using [Azure CycleCloud](#).

Azure CycleCloud is a tool for orchestrating and managing HPC environments on Azure. You can use it to provision infrastructure for HPC systems, deploy HPC schedulers, and automatically scale the infrastructure to run jobs efficiently.

Azure CycleCloud is a Linux-based web application. We recommend that you begin the implementation by deploying an Azure VM that's based on a preconfigured Azure Marketplace image.

To set up an HPC cluster on Azure, complete these steps:

- [Install and configure Azure CycleCloud](#).
- [Create an HPC cluster from built-in templates](#).
- [Connect to the head node \(the scheduler\)](#).

For multi-node configurations, the GROMACS installation process is the same as the process described previously for a single node, except for the path to the installation directory:

- You need to select `/shared` for the installation directory path so that the directory is accessible for all nodes.
- The shared folder path depends on your network attached storage service. For example, an NFS server, BeeGFS cluster, [Azure NetApp Files](#), or [Azure HPC Cache](#).

GROMACS performance results

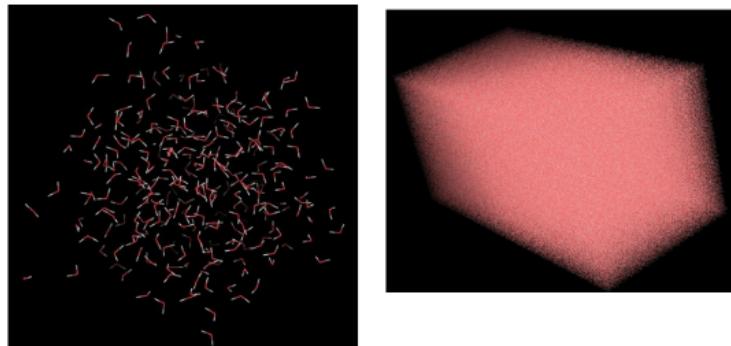
The cell and water models described later in this section were used to test GROMACS. Elapsed times of runs on VMs with varying numbers of CPUs were recorded. The following table provides the details of the operating system that was used for testing.

[Expand table](#)

Operating system version	OS architecture
Ubuntu Linux 20.04	x86-64

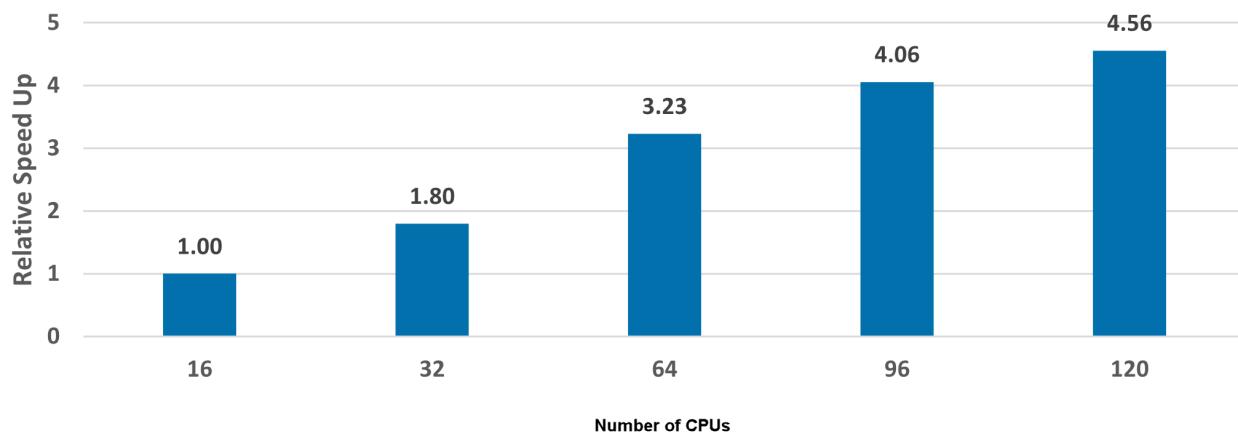
Performance results on a single node

Results for water-cut1.0_GMX50_bare

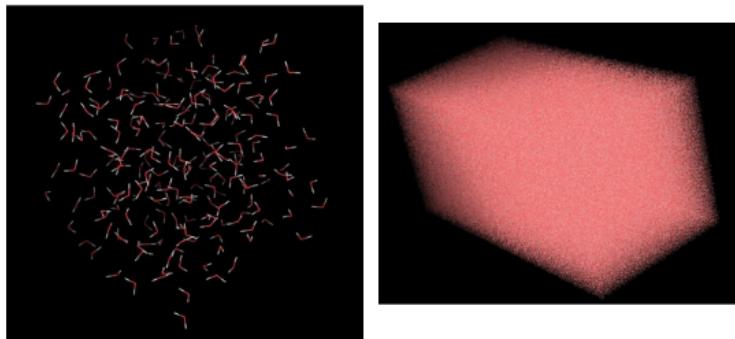
[Expand table](#)

Model	Initial element atom count	Number of cores	Elapsed time (seconds)	Relative speed increase
water-cut1.0_GMX50_bare	3,072,000	16	277.03	NA
water-cut1.0_GMX50_bare	3,072,000	32	153.93	1.80
water-cut1.0_GMX50_bare	3,072,000	64	85.69	3.23
water-cut1.0_GMX50_bare	3,072,000	96	68.26	4.06
water-cut1.0_GMX50_bare	3,072,000	120	60.77	4.56

water-cut1.0_GMX50_bare

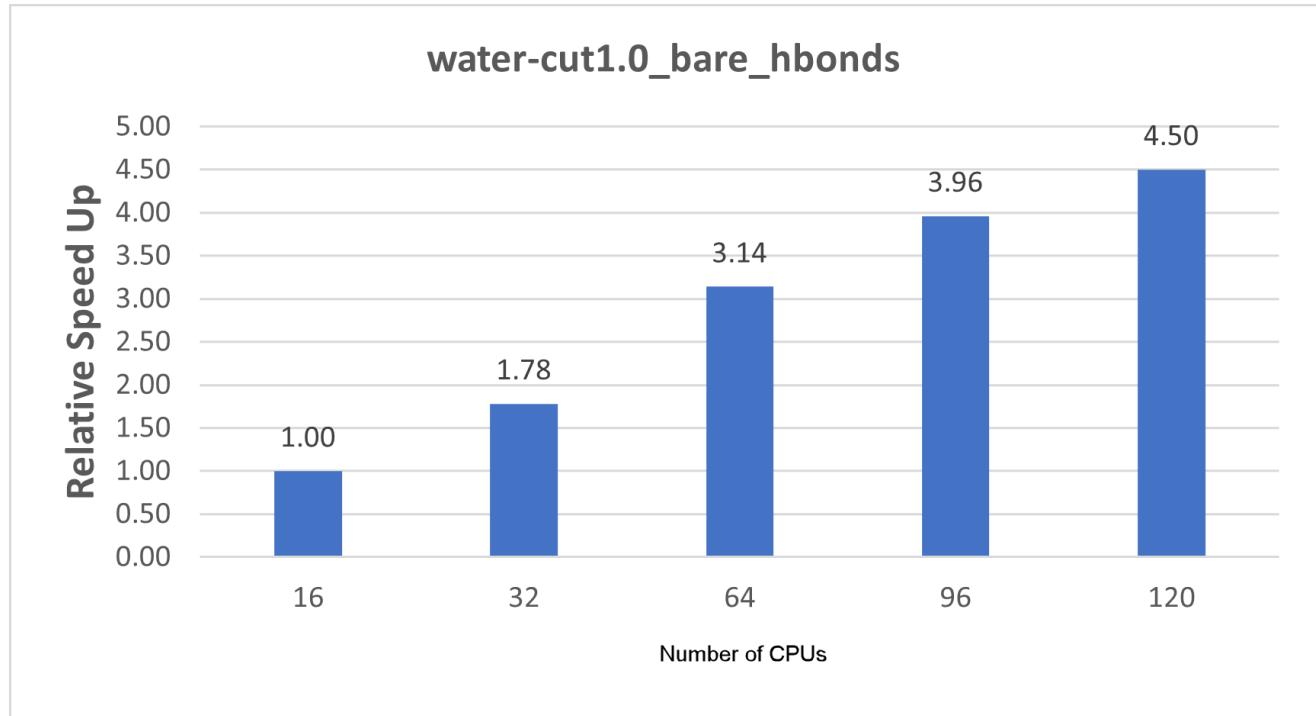


Results for water-cut1.0_bare_hbonds

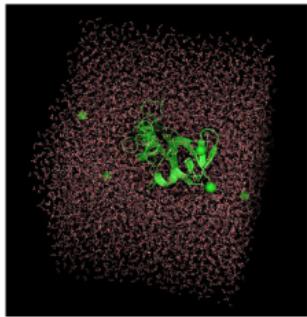


[Expand table](#)

Model	Initial element atom count	Number of cores	Elapsed time (seconds)	Relative speed increase
water-cut1.0_bare_hbonds	3,072,000	16	274.25	NA
water-cut1.0_bare_hbonds	3,072,000	32	153.92	1.78
water-cut1.0_bare_hbonds	3,072,000	64	87.32	3.14
water-cut1.0_bare_hbonds	3,072,000	96	69.22	3.96
water-cut1.0_bare_hbonds	3,072,000	120	60.91	4.50

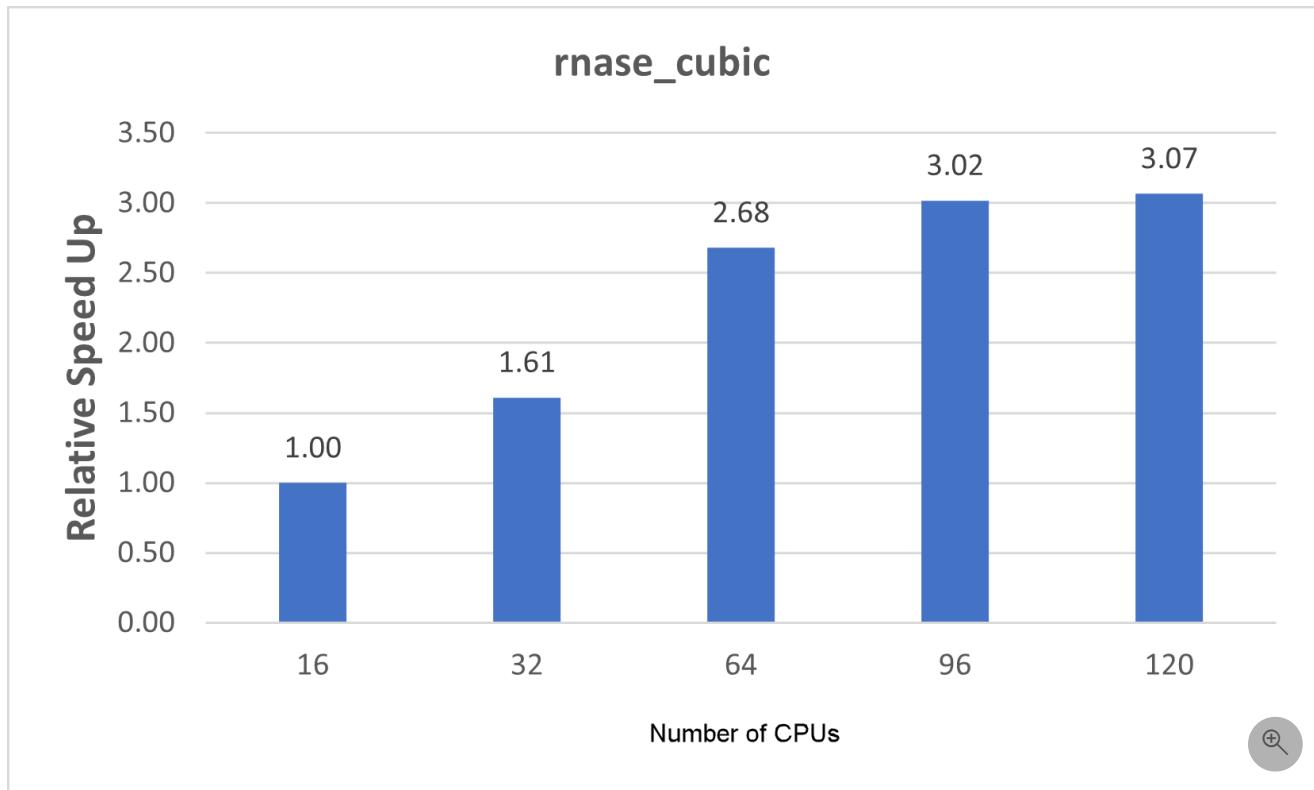


Results for rnase_bench_systems_old-allbond

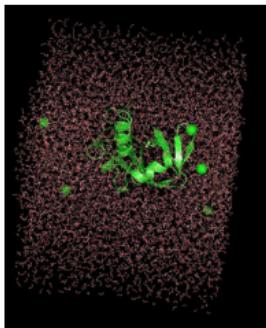


[Expand table](#)

Model	Initial element atom count	Number of cores	Elapsed time (seconds)	Relative speed increase
rnase_cubic	24,040	16	2.47	NA
rnase_cubic	24,040	32	1.53	1.61
rnase_cubic	24,040	64	0.92	2.68
rnase_cubic	24,040	96	0.82	3.02
rnase_cubic	24,040	120	0.81	3.07

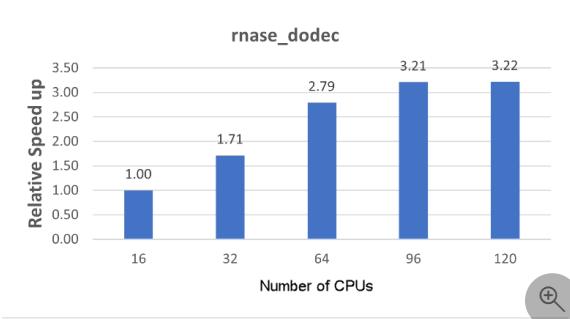
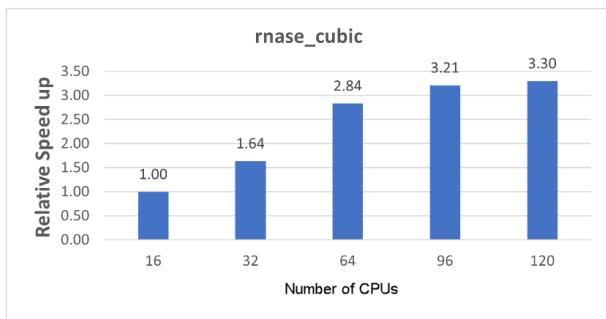


Results for rnase_bench_systems

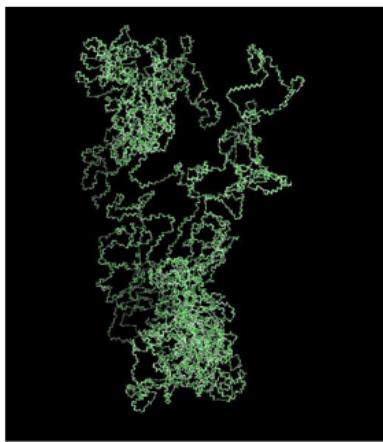


[Expand table](#)

Model	Initial element atom count	Number of cores	Elapsed time (seconds)	Relative speed increase
rnase_cubic	24,040	16	2.41	NA
rnase_cubic	24,040	32	1.47	1.64
rnase_cubic	24,040	64	0.85	2.84
rnase_cubic	24,040	96	0.75	3.21
rnase_cubic	24,040	120	0.73	3.30
rnase_dodec	16,816	16	2.17	1.00
rnase_dodec	16,816	32	1.27	1.71
rnase_dodec	16,816	64	0.78	2.79
rnase_dodec	16,816	96	0.68	3.21
rnase_dodec	16,816	120	0.67	3.22

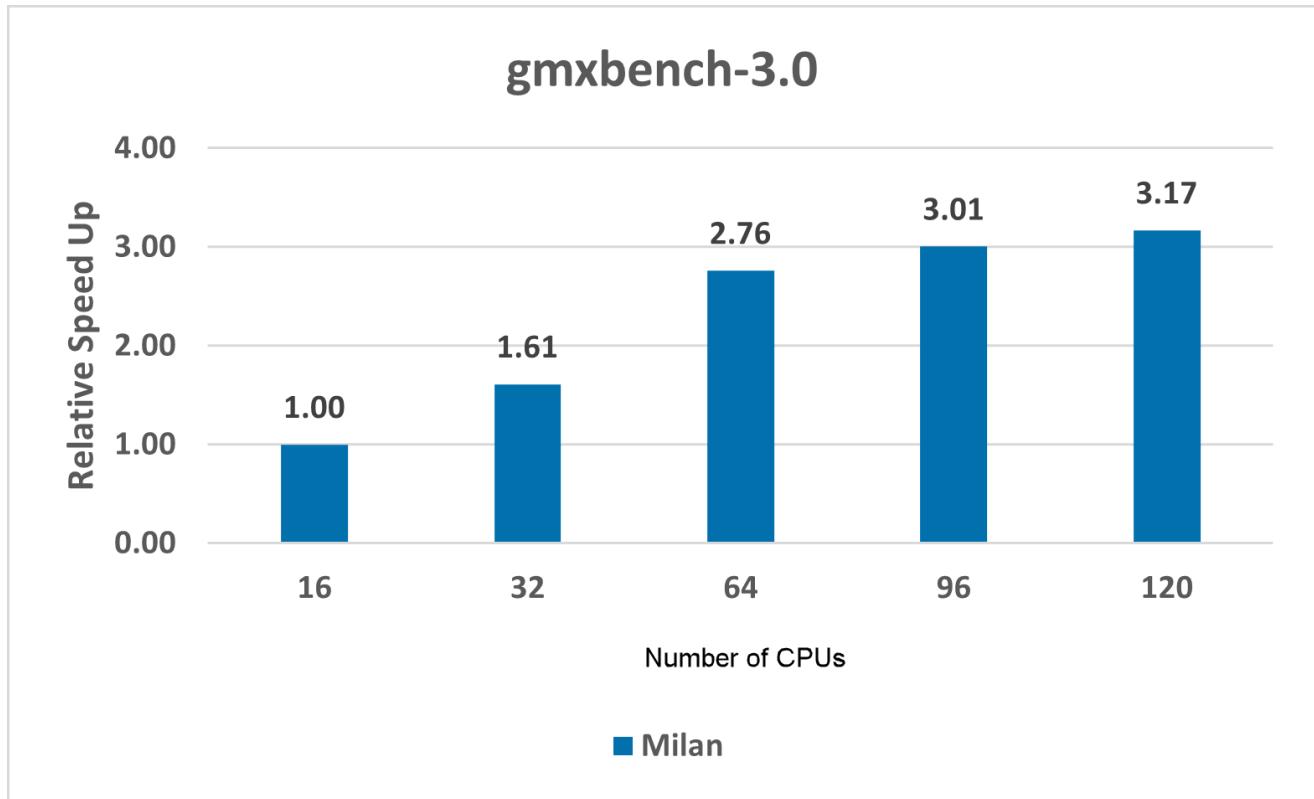


Results for gmxbench-3.0

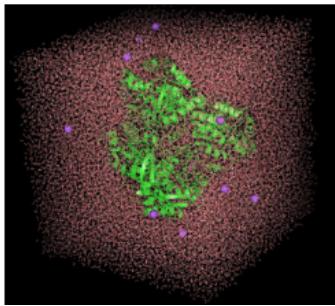


[Expand table](#)

Model	Initial element atom count	Number of cores	Elapsed time (seconds)	Relative speed increase
d.poly-ch2	6,000 atoms and 6,000 vsites	16	0.58	NA
d.poly-ch2	6,000 atoms and 6,000 vsites	32	0.36	1.61
d.poly-ch2	6,000 atoms and 6,000 vsites	64	0.21	2.76
d.poly-ch2	6,000 atoms and 6,000 vsites	96	0.19	3.01
d.poly-ch2	6,000 atoms and 6,000 vsites	120	0.18	3.17

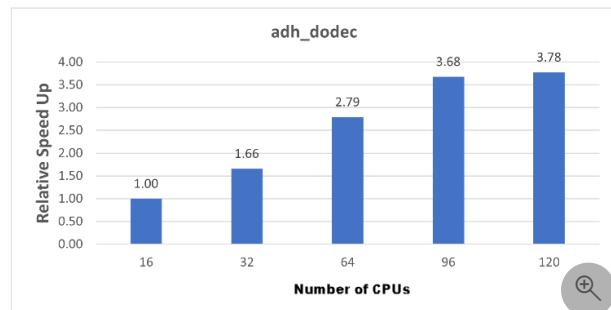
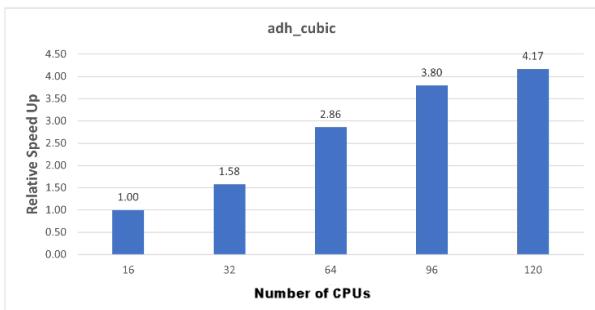


Results for ADH_bench_systems_old-allbonds

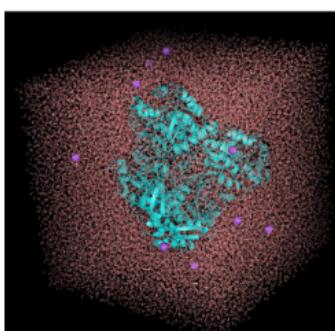


[Expand table](#)

Model	Initial element atom count	Number of cores	Elapsed time (seconds)	Relative speed increase
adh_cubic	134,177	16	12.91	NA
adh_cubic	134,177	32	8.18	1.58
adh_cubic	134,177	64	4.52	2.86
adh_cubic	134,177	96	3.40	3.80
adh_cubic	134,177	120	3.10	4.17
adh_dodec	95,561	16	10.87	1.00
adh_dodec	95,561	32	6.55	1.66
adh_dodec	95,561	64	3.89	2.79
adh_dodec	95,561	96	2.96	3.68
adh_dodec	95,561	120	2.88	3.78

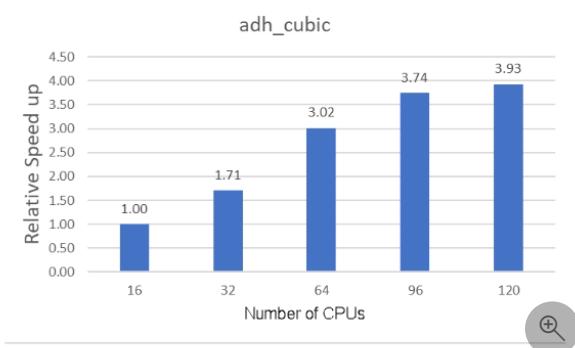
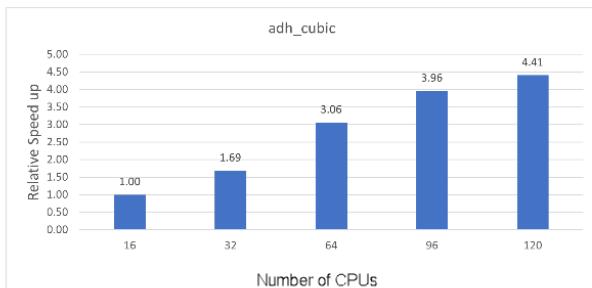


Results for ADH_bench_systems



[Expand table](#)

Model	Initial element atom count	Number of cores	Elapsed time (seconds)	Relative speed increase
adh_cubic	134,177	16	12.94	NA
adh_cubic	134,177	32	7.67	1.69
adh_cubic	134,177	64	4.23	3.06
adh_cubic	134,177	96	3.27	3.96
adh_cubic	134,177	120	2.94	4.41
adh_dodec	95,561	16	10.42	1.00
adh_dodec	95,561	32	6.1	1.71
adh_dodec	95,561	64	3.46	3.02
adh_dodec	95,561	96	2.78	3.74
adh_dodec	95,561	120	2.65	3.93



Performance results on a multi-node configuration

This VM configuration was used for the multi-node tests:

[Expand table](#)

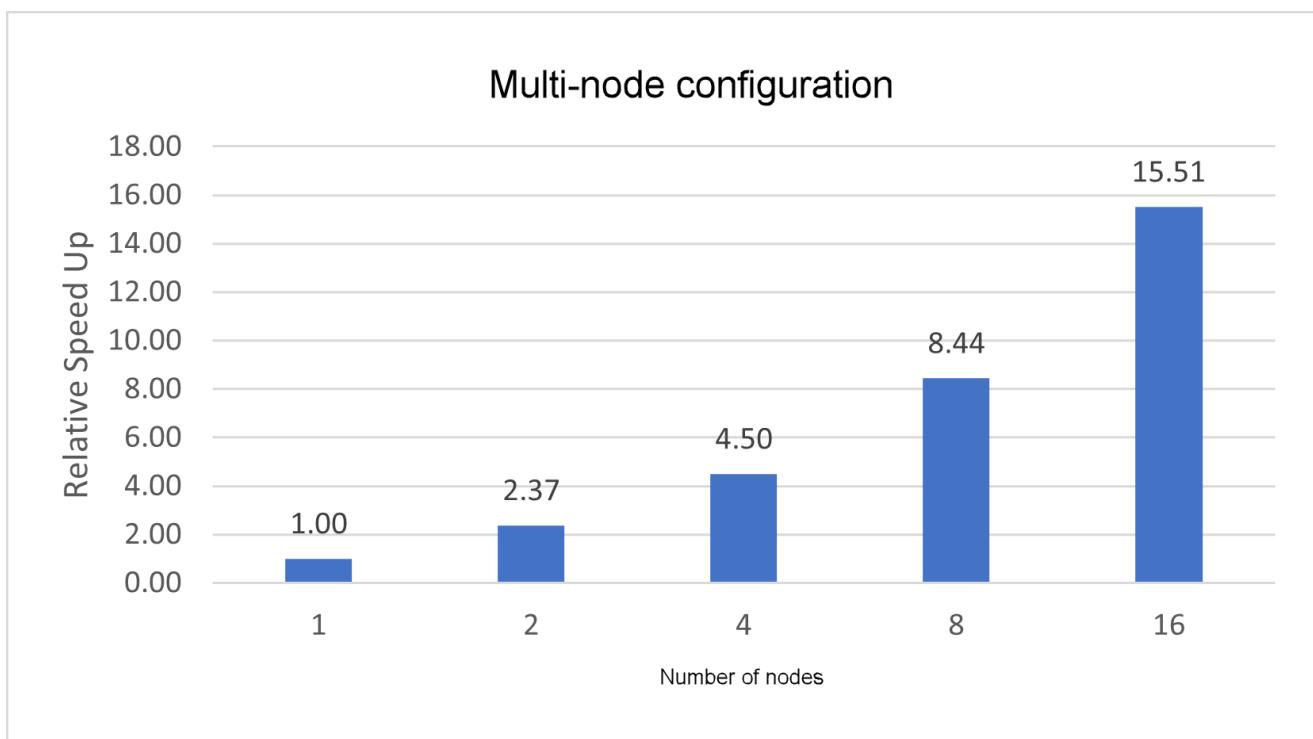
Operating system version	OS architecture	Processor
CentOS 8	x86-64	AMD EPYC 7V73X

Results for water-cut1.0_GMX50_bare

12,000 calculation steps were used in these tests.

[Expand table](#)

Model	Initial element atom count	Number of nodes	Number of cores	Elapsed time (seconds)	Relative speed increase
water-cut1.0_GMX50_bare	3,072,000	1	64	297.30	NA
water-cut1.0_GMX50_bare	3,072,000	2	128	125.37	2.37
water-cut1.0_GMX50_bare	3,072,000	4	256	66.12	4.50
water-cut1.0_GMX50_bare	3,072,000	8	512	35.22	8.44
water-cut1.0_GMX50_bare	3,072,000	16	1,024	19.17	15.51



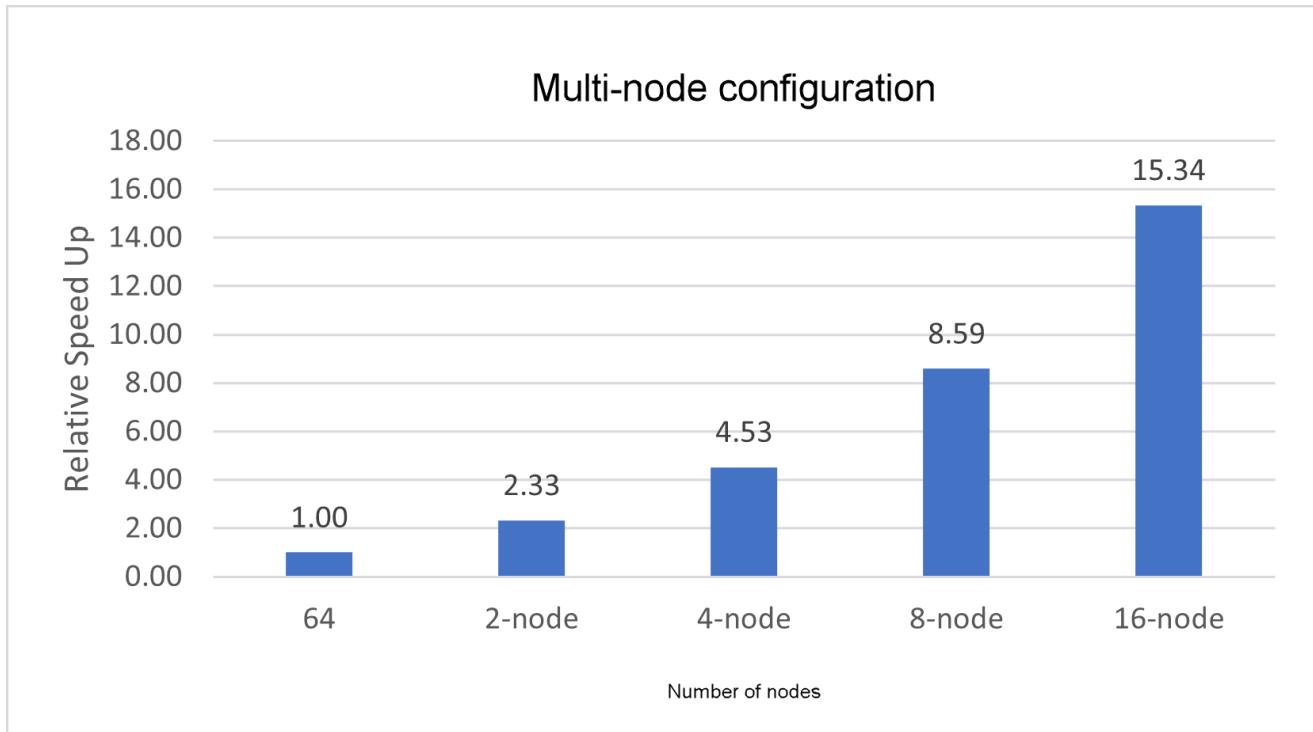
Results for water-cut1.0_bare_hbonds

12,000 calculation steps were used in these tests.

[Expand table](#)

Model	Initial element atom count	Number of nodes	Number of cores	Elapsed time (seconds)	Relative speed increase
water-cut1.0_bare_hbonds	3,072,000	1	64	298.15	NA
water-cut1.0_bare_hbonds	3,072,000	2	128	128.19	2.33
water-cut1.0_bare_hbonds	3,072,000	4	256	65.85	4.53
water-	3,072,000	8	512	34.71	8.59

Model	Initial element atom count	Number of nodes	Number of cores	Elapsed time (seconds)	Relative speed increase
cut1.0_bare_hbonds					
water-cut1.0_bare_hbonds	3,072,000	16	1,024	19.43	15.34

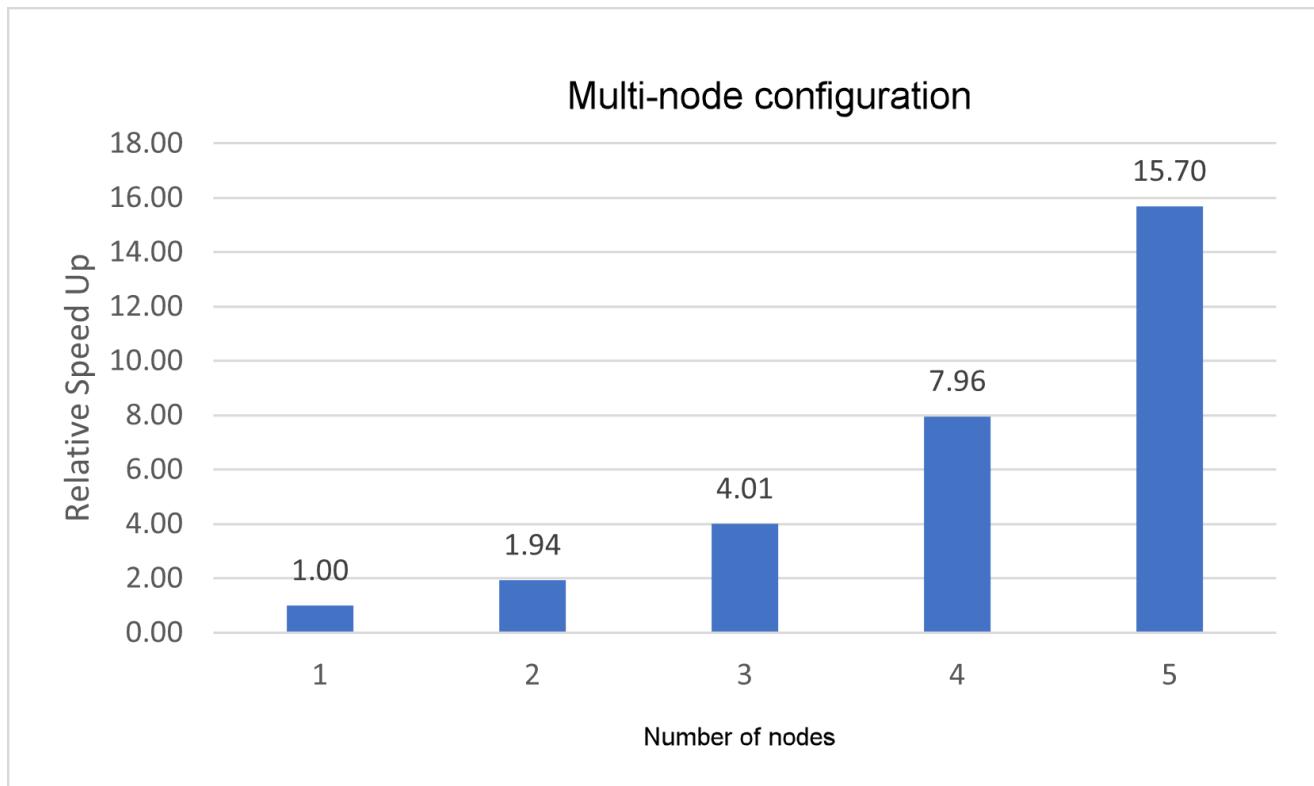


Results for benchPEP

4,000 calculation steps were used in these tests.

[Expand table](#)

Model	Initial element atom count	Number of nodes	Number of cores	Elapsed time (seconds)	Relative speed increase
benchPEP	12,000,000	1	64	499.414	NA
benchPEP	12,000,000	2	128	257.945	1.94
benchPEP	12,000,000	4	256	124.556	4.01
benchPEP	12,000,000	8	512	62.708	7.96
benchPEP	12,000,000	16	1,024	31.818	15.70



Azure cost

The following tables present wall-clock times that you can use to calculate Azure costs. You can multiply the times presented here by the Azure hourly rates for HBv3-series VMs to calculate costs. For the current hourly costs, see [Linux Virtual Machines Pricing](#).

Only the wall-clock time for running the test cases is considered for these cost calculations. Application installation time isn't considered.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

Single-node configurations

Expand table

VM size	Number of CPUs	Elapsed time (hours)
Standard_HB120-16rs_v3	16	0.169
Standard_HB120-32rs_v3	32	0.095
Standard_HB120-64rs_v3	64	0.053
Standard_HB120-96rs_v3	96	0.043
Standard_HB120rs_v3	120	0.038

Multi-node configurations

Expand table

VM size	Number of CPUs	Elapsed time (hours)
Standard_HB120-16rs_v3	64	0.304

VM size	Number of CPUs	Elapsed time (hours)
Standard_HB120-32rs_v3	128	0.142
Standard_HB120-64rs_v3	256	0.071
Standard_HB120-96rs_v3	512	0.037
Standard_HB120rs_v3	1024	0.020

Summary

- GROMACS was tested successfully on Azure HBv3-series virtual machines in single-node and multi-node configurations.
- With a 120 vCPU, the performance is three to four times the performance with 16 CPUs.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- Shivakumar Tallolli | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

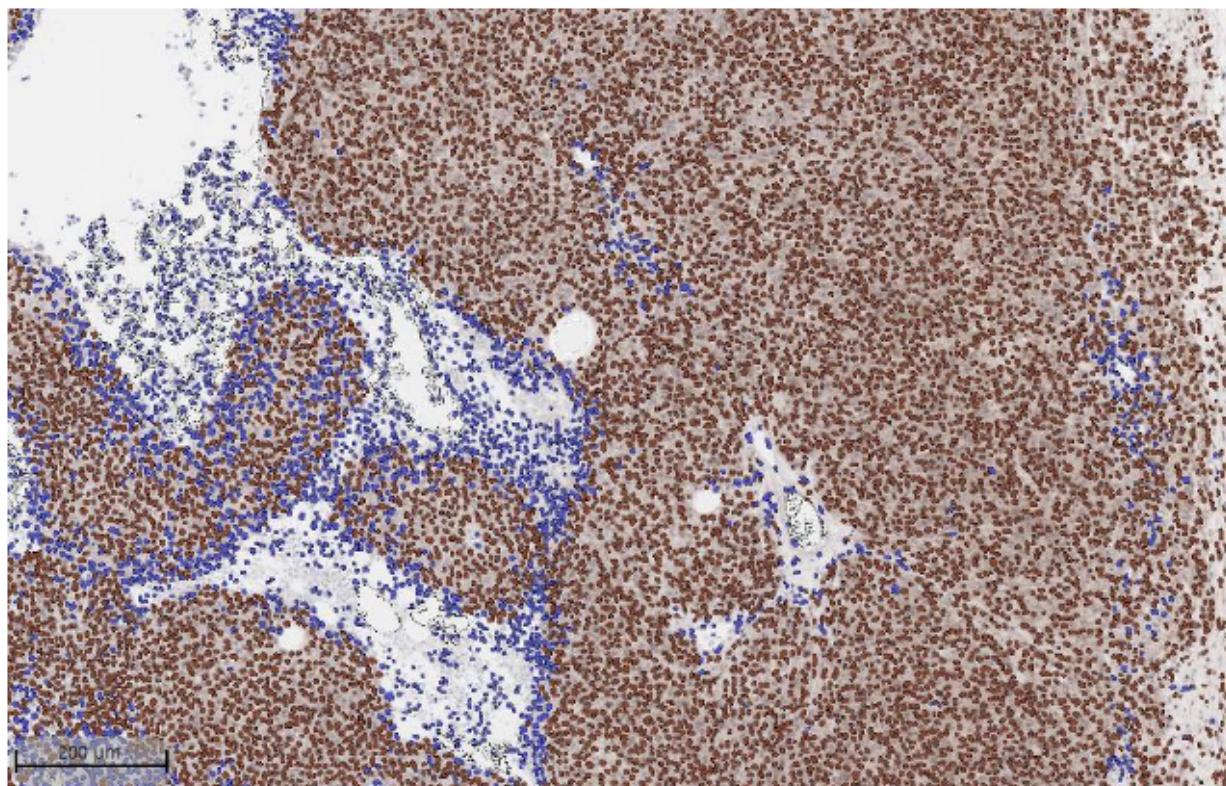
Deploy HALO AI on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [Indica Labs HALO AI](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running HALO AI on Azure.

HALO AI is a collection of train-by-example classification and segmentation tools underpinned by advanced deep learning neural network algorithms. It was originally developed as a tool that could decipher and assess the complex patterns of histologically stained tissues in a way that's similar to how a pathologist thinks.



HALO AI has the following capabilities.

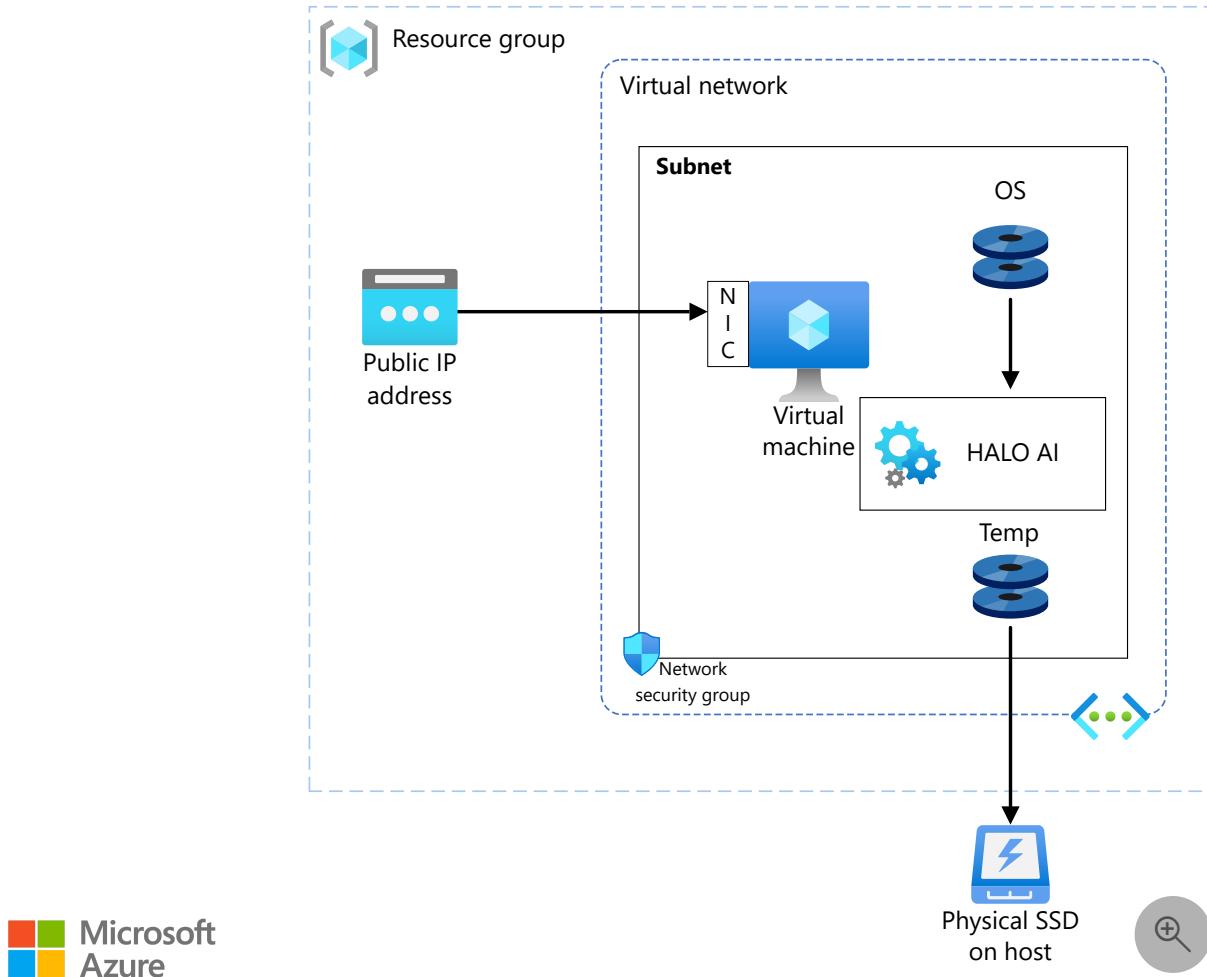
- Includes three powerful neural networks: VGG, DenseNet, and MiniNet.
- Provides pretrained networks for H&E, single IHC, or DAPI stained-images. You can also train your own nuclei segmentation network for a specific application.
- Uses a type of deep learning algorithm called a convolutional neural network (CNN), which is ideally suited for tissue classification in digital pathology.

HALO AI is used across the healthcare industry, in clinical, pharmaceutical, biotech, and central research (education) organizations.

Why deploy HALO AI on Azure?

- Modern and diverse compute options to align to your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Fast compute capabilities for GPU-intensive workloads

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Windows VM. For information about deploying the VM and installing the drivers, see [Windows VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

Performance tests of HALO AI on Azure used [Standard_NC6s_v3](#) and [Standard_NC4as_T4_v3](#) VMs running Windows. The following table provides details about the VMs.

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

VM size	vCPU	Memory (GiB)	SSD (GiB)	GPU	GPU memory (GiB)	Maximum data disks
Standard_NC6s_v3	6	112	736	V100	16	12
Standard_NC4as_T4_v3	4	28	180	T4	16	8

Required drivers

To take advantage of the GPU capabilities of [Standard_NC6s_v3](#) and [Standard_NC4as_T4_v3](#) VMs, you need to install NVIDIA GPU drivers.

To use AMD processors on Standard_NC4as_T4_v3 VMs, you need to install AMD drivers.

HALO AI installation

Before you install HALO AI, you need to deploy and connect a VM, install an eligible Windows 10 image, and install the required NVIDIA and AMD drivers.

For information about eligible Windows images, see [How to deploy Windows 10 on Azure](#) and [Use Windows client in Azure for dev/test scenarios](#).

For information about deploying the VM and installing the drivers, see [Run a Windows VM on Azure](#).

For information about installing HALO AI on an Azure VM, contact [Indica Labs](#).

HALO AI performance results

HALO AI performs best on machines that have single-GPU configurations. Testing was performed on Standard_NC6s_v3, which has one NVIDIA V100 GPU, and Standard_NC4as_T4_v3, which has one T4 GPU. Image classification was performed on 20 pathology datasets.

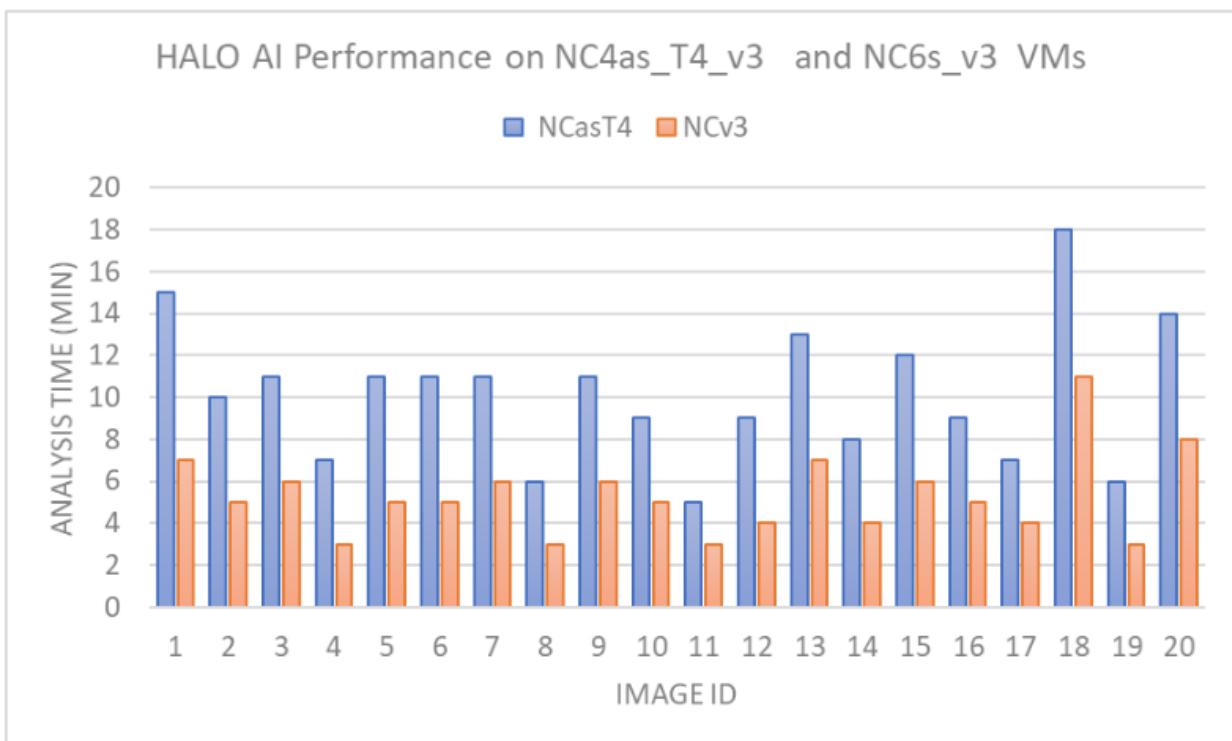
The following table shows the test results.

[Expand table](#)

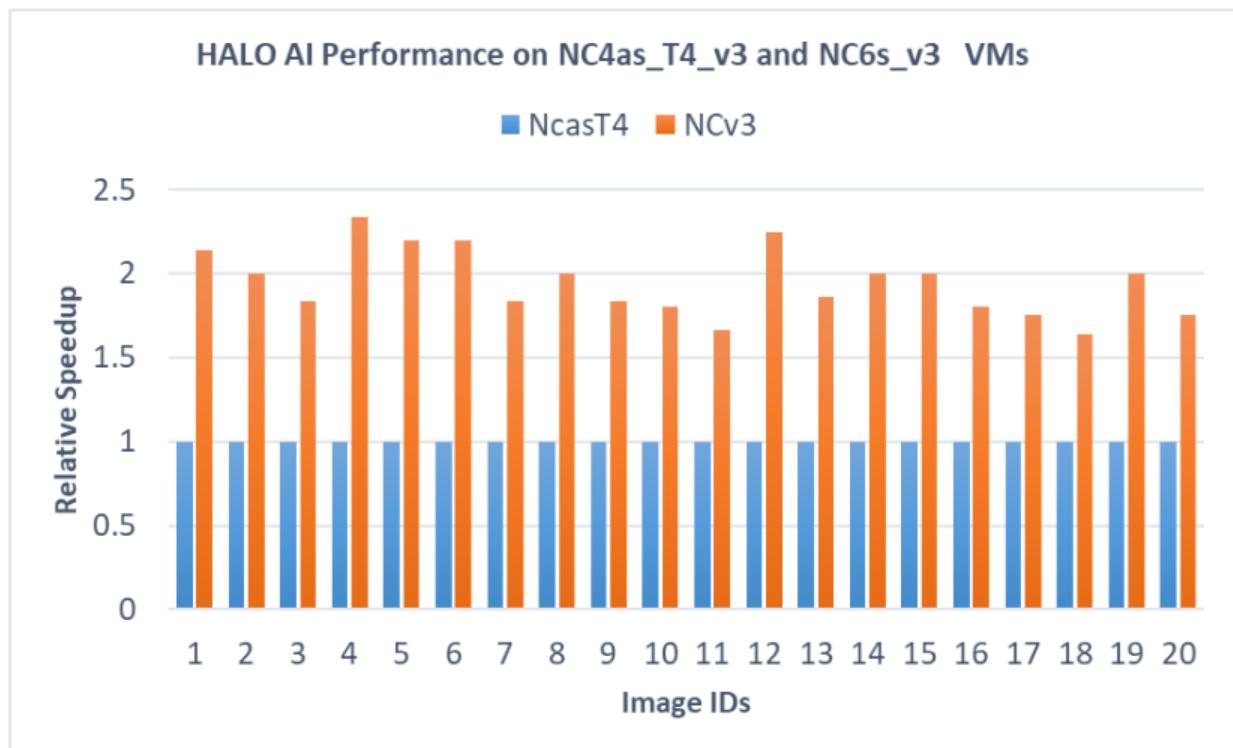
Image ID	Analysis time on NC4as_T4_v3 (minutes)	Analysis time on NC6s_v3 (minutes)
1	15	7
2	10	5
3	11	6
4	7	3
5	11	5
6	11	5
7	11	6
8	6	3
9	11	6
10	9	5
11	5	3
12	9	4
13	13	7
14	8	4
15	12	6
16	9	5

Image ID	Analysis time on NC4as_T4_v3 (minutes)	Analysis time on NC6s_v3 (minutes)
17	7	4
18	18	11
19	6	3
20	14	8

This graph shows the elapsed running times on both VMs:



NC6s_v3 is consistently faster. This graph shows the relative speed increases:



Azure cost

The following table presents elapsed times that you can use to calculate Azure costs. You can multiply the times presented here by the Azure hourly rates for NCasT4_v3 and NCsv3 series VMs to calculate costs. For the current hourly costs, see [Windows Virtual Machines Pricing](#).

Only analysis times are considered for the cost calculations. Application installation time isn't considered. These times are indicative. Actual times depend on the size of the model.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

[] [Expand table](#)

VM size	GPU	Elapsed time for all 20 images (hours)
Standard_NC4as_T4_v3	T4	3.38
Standard_NC6s_v3	V100	1.77

Summary

- HALO AI was successfully tested on NC6s_v3 and NC4as_T4 series VMs. HALO AI performs best with single-GPU configurations, so we recommend that you use a VM with that configuration.
- NC6s_v3 is almost twice as fast as NC4as_T4.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- GPU-optimized virtual machine sizes
- Virtual machines on Azure
- Virtual networks and virtual machines on Azure
- Learning path: Run high-performance computing (HPC) applications on Azure

Related resources

- [Run a Windows VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy LAMMPS on an Azure virtual machine

Azure Virtual Machines Azure Virtual Network

This article briefly describes the steps for installing and running [LAMMPS](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running LAMMPS on single-node and multi-node VM configurations.

LAMMPS is a classical molecular dynamics simulator that's used for materials modeling. It can model solid-state materials, soft matter, and coarse-grained or mesoscopic systems. It can be used to model atoms or, more generically, as a parallel particle simulator at the atomic, meso, or continuum scale.

LAMMPS is designed to run well on parallel machines, but it also runs on single-processor desktop machines. It's composed of modular code, and most of its functionality is in optional packages.

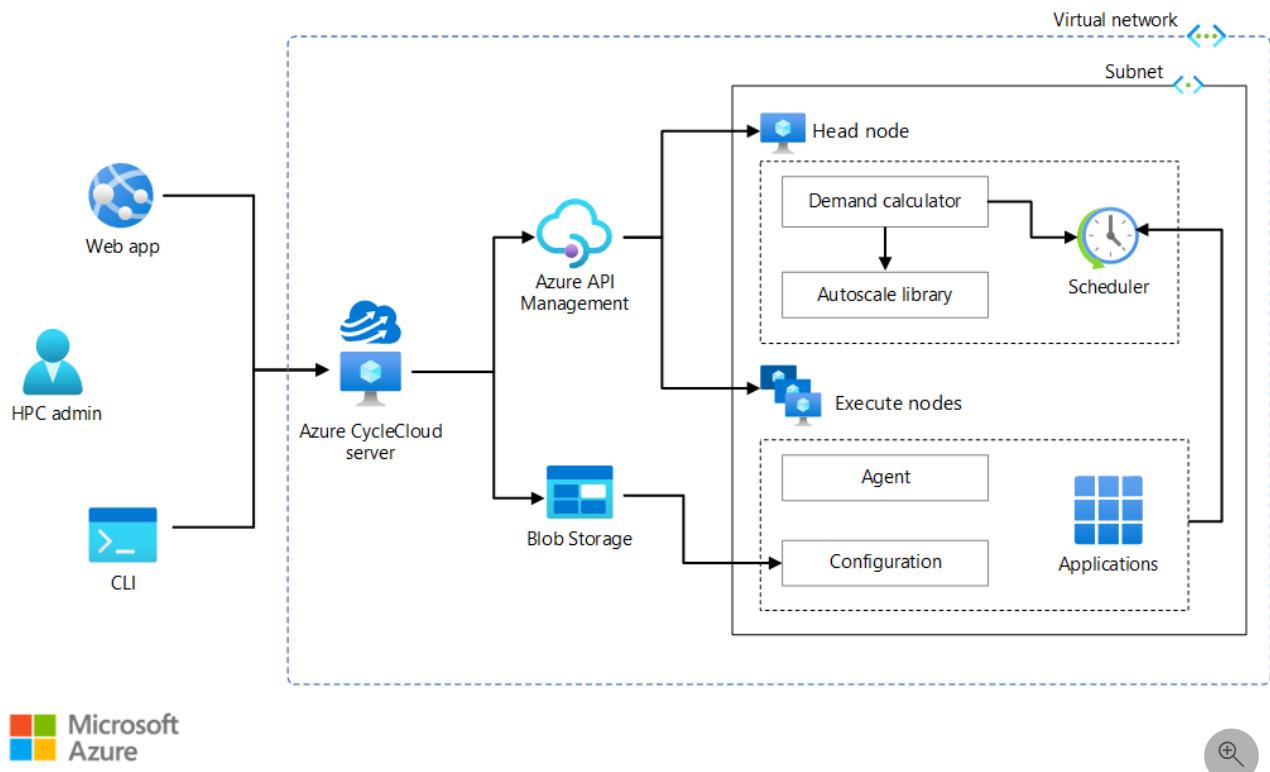
Typical LAMMPS simulations include all-atom models of liquids, solids, and explicit solvents.

Why deploy LAMMPS on Azure?

- Modern and diverse compute options to meet your workload's needs.
- The flexibility of virtualization without the need to buy and maintain physical hardware.
- Rapid provisioning.
- Support for Message Passing Interface (MPI).
- Ability to run large and time-consuming jobs.

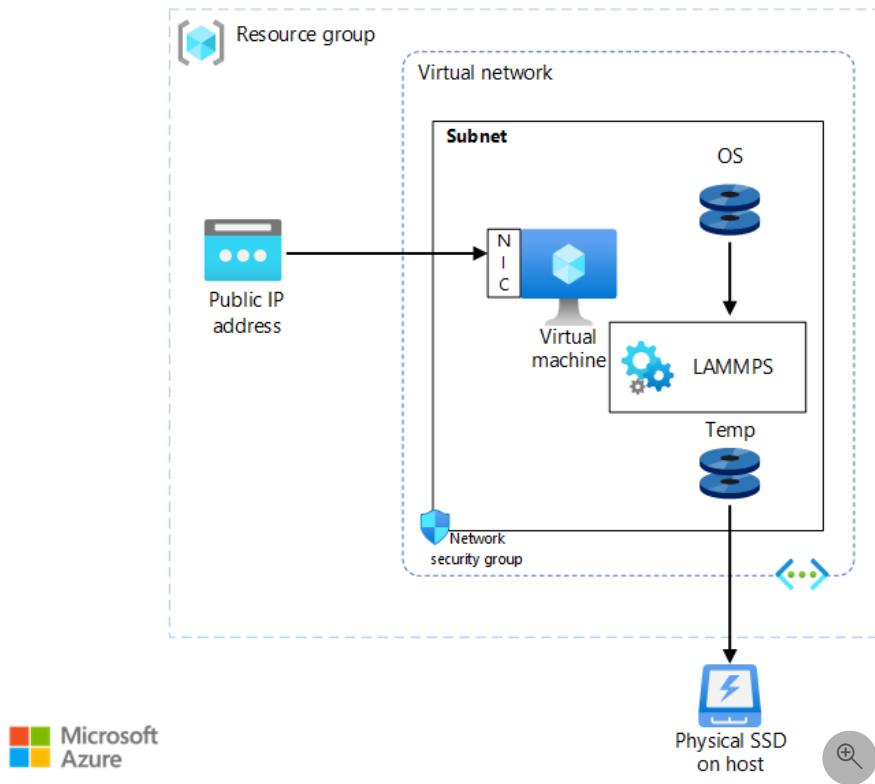
Architecture

This diagram shows a multi-node configuration:



[Download a Visio file](#) of this architecture.

This diagram shows a single-node configuration:



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create Linux VMs.
 - For information about deploying VMs, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) restrict access to VMs.
 - A public IP address connects the internet to VMs.
- [Azure CycleCloud](#) is used to create the cluster in the multi-node configuration.
- A physical SSD provides storage.

Compute sizing

Performance tests of LAMMPS on Azure used [HBv3 AMD EPYC 7V73X](#) (Milan-X) VMs running Linux CentOS. The following table provides details about HBv3-series VMs.

[Expand table](#)

VM size	vCPU	Memory (GiB)	Memory bandwidth (GBps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120rs_v3	120	448	350	1.9	3.0	3.5	200	32
Standard_HB120-96rs_v3	96	448	350	1.9	3.0	3.5	200	32
Standard_HB120-64rs_v3	64	448	350	1.9	3.0	3.5	200	32
Standard_HB120-32rs_v3	32	448	350	1.9	3.0	3.5	200	32

VM size	vCPU	Memory (GiB)	Memory bandwidth (Gbps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120-16rs_v3	16	448	350	1.9	3.0	3.5	200	32

Install LAMMPS on a VM or HPC cluster

You can download the software from the [LAMMPS](#) website. You just need to untar or unzip the LAMMPS binary distribution file, and you can run LAMMPS directly in the resulting directory. For a guide to building from source code, see [Build LAMMPS](#).

Before you install LAMMPS, you need to deploy and connect to a VM or HPC cluster.

For information about deploying a VM, see [Run a Linux VM on Azure](#).

For information about deploying the Azure CycleCloud and HPC cluster, see these resources:

- [Install and configure Azure CycleCloud](#)
- [Create an HPC cluster](#)

Install LAMMPS

Complete the following steps to install LAMMPS on single-node and cluster VMs.

1. Run the following commands:

```
export PATH=$PATH:/opt/openmpi-4.1.0/bin/
export LD_LIBRARY_PATH=/opt/openmpi-4.1.0/lib
export CC=gcc
export CXX=g++
export FC=gfortran
export FCFLAGS=-m64
export F77=gfortran
export F90=ifort
export CPPFLAGS=-DpggFortran
```

2. Download the source code from [LAMMPS](#).

3. Unzip the file:

```
tar xvf *
```

4. Locate the LAMMPS folder:

```
cd lammps-<version>
cd src
```

5. To build LAMMPS, run these commands in the *src* folder:

```
make yes-rigid  
make serial  
make mpi
```

Run LAMMPS

1. To run LAMMPS on a standalone VM, use these commands:

```
export PATH=$PATH:/opt/openmpi-4.1.0/bin/  
  
export LD_LIBRARY_PATH=/opt/openmpi-4.1.0/lib  
  
export LMP_MPI=/path/LAMMPS/lammps-<version>/src/lmp_mpi  
  
mpirun -np 16 /path/LAMMPS/lammps-<version>/src/lmp_mpi -in in.lj
```

2. To run LAMMPS on a multi-node cluster, use this script:

```
1  #!/bin/bash  
  
2  #SBATCH --job-name=LAMMPS  
  
3  #SBATCH --partition=hpc  
  
4  #SBATCH --nodes=2  
  
5  #SBATCH --ntasks-per-node=64  
  
6  #SBATCH --ntasks=128  
  
7  export PATH=$PATH:/opt/openmpi-4.1.0/bin/  
  
8  export LD_LIBRARY_PATH=/opt/openmpi-4.1.0/lib  
  
9  export LMP_MPI=/path/LAMMPS/lammps-<version>/src/lmp_mpi  
  
10 mpirun -np 64 /path/LAMMPS/lammps-<version>/src/lmp_mpi -in benchmark.in
```

ⓘ Note

In the preceding script, `ntasks` on line 6 is the number of nodes multiplied by the number of cores per VM configuration. The number of nodes is 2, as specified on line 4. The number of cores per VM configuration is 64, as specified on line 5. So `ntasks` is 128.

LAMMPS performance results on Azure VMs

Two models were used to test the performance of LAMMPS version 23 and LAMMPS version 17 on Azure.

Lennard-Jones model

Lennard-Jones (in.lj) is a simple molecular dynamics simulation of a binary fluid in the NVT ensemble. It's made of neutral dots with a Langevin thermostating.

The following table provides details about the Lennard-Jones model:

 Expand table

Number of atoms	Timestep	Thermo step	Run steps
1.0e+9	0.1	10	200

HECBioSim model

HECBioSim is a benchmark suite that consists of a set of simple benchmarks for a number of popular molecular dynamics engines, each of which is set at a different atom count.

The following table provides details about the HECBioSim model:

[Expand table](#)

Number of atoms	Timestep	Thermo step	Run steps
1,403,180	2.0	5,000	10,000

LAMMPS performance results on single-node VMs

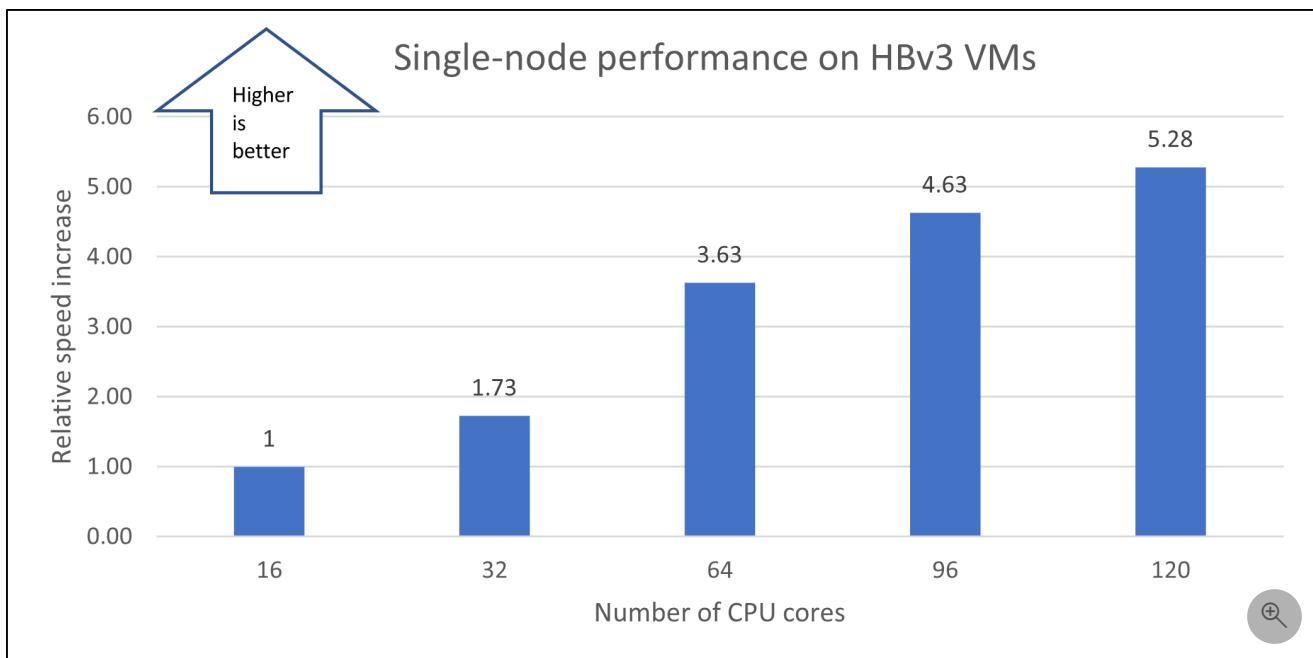
The following sections provide the performance results of running LAMMPS version 23 on single-node Azure [HBv3 AMD EPYC 7V73X](#) (Milan-X) VMs. The Lennard-Jones model is used in these tests.

This table shows the total wall clock times recorded for various numbers of CPUs on the Standard HBv3-series VM:

[Expand table](#)

Number of cores	Wall clock time (seconds)	Relative speed increase
16	7,634	1
32	4,412	1.73
64	2,102	3.63
96	1,648	4.63
120	1,445	5.28

The following graph shows the relative speed increases as the number of CPUs increases:



Notes about the single-node tests

For the single-node tests, the Standard_HB120-16rs_v3 VM (16 cores) is used as a baseline to calculate relative speed increases as the number of cores increases. The results show that parallel performance improves as the number of cores increases from 16 to 120. A speed increase of 5.3x is achieved with 120 cores.

LAMMPS performance results on multi-node clusters

The single-node tests show that optimal parallel performance is reached with 64 cores on HBv3 VMs. Based on those results, 64-core configurations on [Standard_HB120-64rs_v3](#) VMs are used to evaluate the performance of LAMMPS on multi-node clusters. The Lennard-Jones and HECBioSim models are used for the multi-node tests.

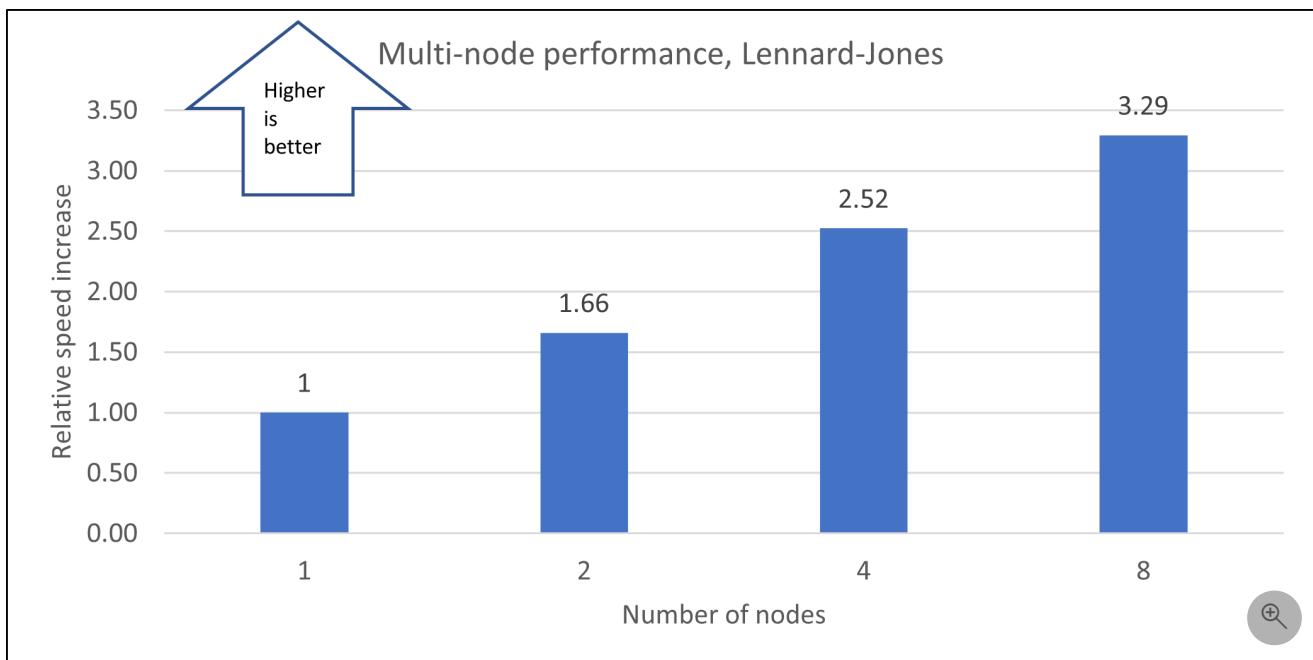
Lennard-Jones model

This table shows the total wall clock times recorded for various numbers of nodes:

[Expand table](#)

Number of nodes	Number of cores	Wall clock time (seconds)	Relative speed increase
1	64	2,612	N/A
2	128	1,573	1.66
4	256	1,035	2.52
8	512	793	3.29

The following graph shows the relative speed increases as the number of nodes increases:



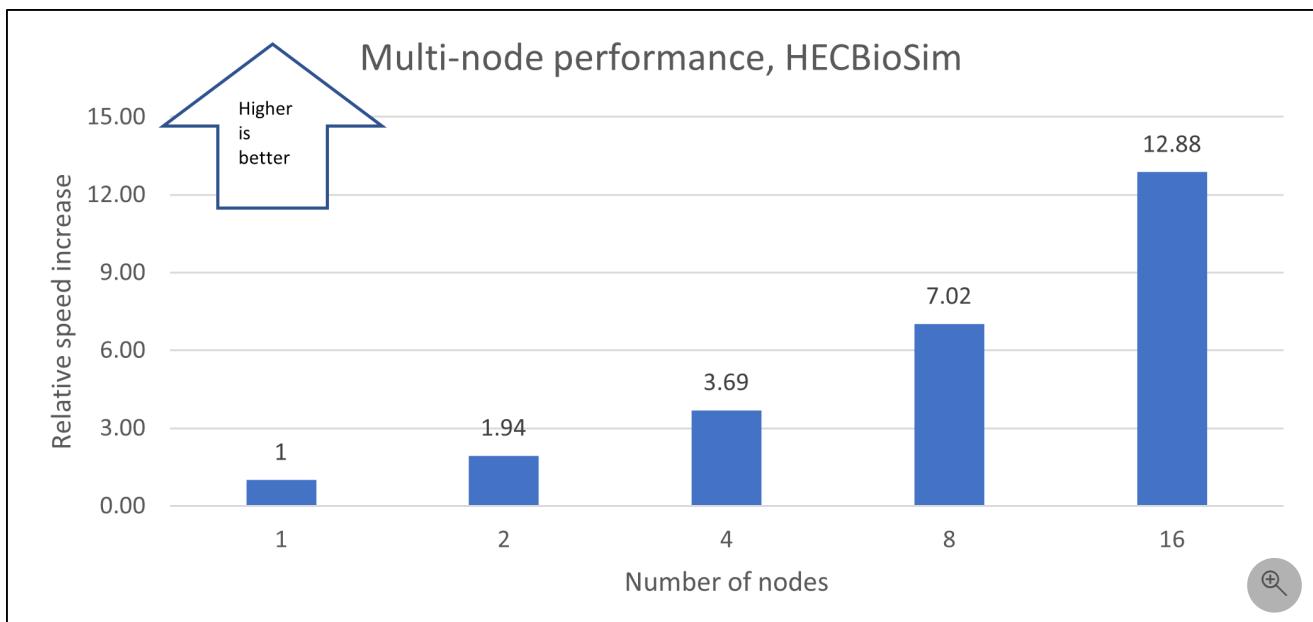
HECBioSim model

This table shows the total wall clock times recorded for various numbers of nodes:

[Expand table](#)

Number of nodes	Number of cores	Wall clock time (seconds)	Relative speed increase
1	64	3,103	N/A
2	128	1,601	1.94
4	256	840	3.69
8	512	442	7.02
16	1,024	241	12.88

The following graph shows the relative speed increases as the number of nodes increases:



Notes about the multi-node tests

- The multi-node results show that both models scale well when you increase the number of nodes.
- The Lennard-Jones model was tested with LAMMPS version 23. The HECBioSim model was tested with LAMMPS version 17.

Azure cost

The following tables provide wall clock times that you can use to calculate Azure costs. To compute the cost, multiply the wall clock time by the number of nodes and the Azure VM hourly rate. For the hourly rates for Linux, see [Linux Virtual Machines Pricing](#). Azure VM hourly rates are subject to change.

Only simulation running time is considered for the cost calculations. Installation time, simulation setup time, and software costs aren't included.

You can use the [Azure pricing calculator](#) to estimate VM costs for your configurations.

Running times for the Lennard-Jones model

 [Expand table](#)

Number of nodes	Wall clock time (hours)
1	0.73
2	0.44
4	0.29
8	0.22

Running times for the HECBioSim model

 [Expand table](#)

Number of nodes	Wall clock time (hours)
1	0.86
2	0.44
4	0.23
8	0.12
16	0.07

Summary

- LAMMPS was successfully tested on HBv3 standalone VMs and Azure CycleCloud multi-node configurations with as many as 16 nodes.
- In multi-node configurations, tests indicate speed increases of about 3.29x for the Lennard-Jones model and about 12.88x for the HECBioSim model.
- For small simulations, we recommend that you use fewer CPUs to improve performance.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Amol Rane](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director, Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign into LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run HPC applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Luxion KeyShot on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [Luxion KeyShot](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running KeyShot on Azure.

KeyShot is a standalone, real-time ray tracing and global illumination program that's used to create 3D renderings, animations, and interactive visuals. It uses photon mapping, an extension of ray tracing, which makes simulation of global illumination in complex scenes more efficient. KeyShot has the following capabilities:

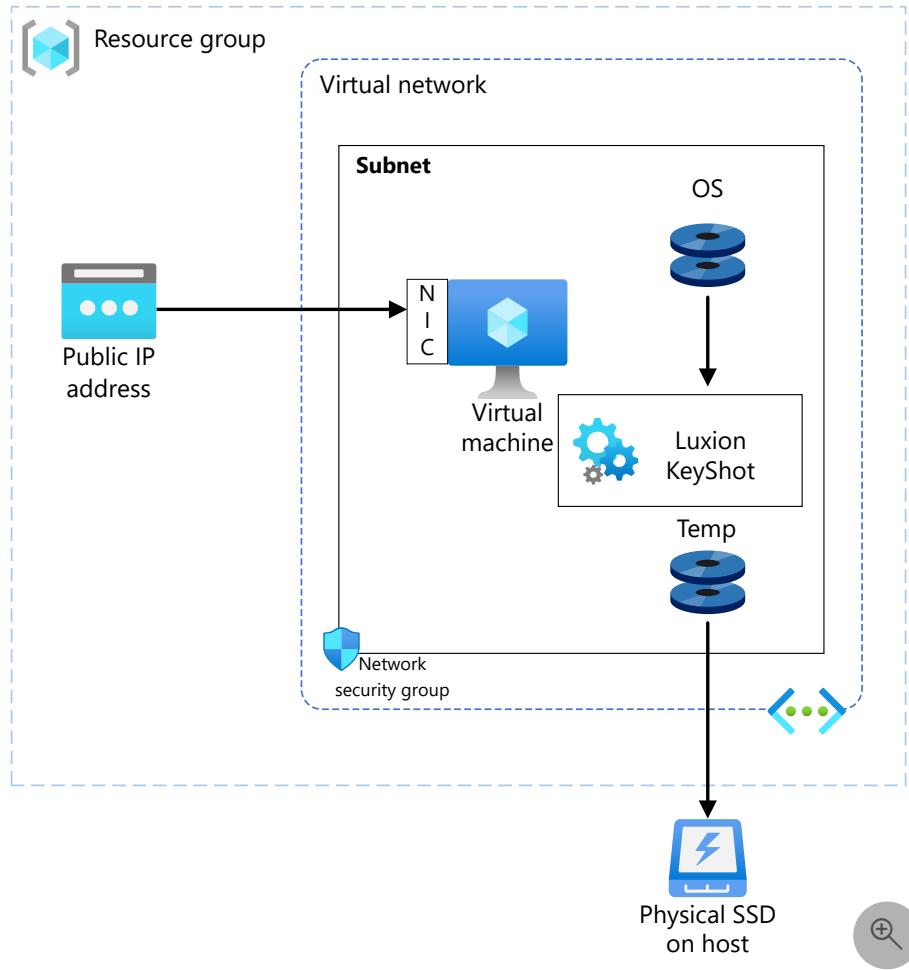
- 3D-paint enabled, so users can directly paint or stamp bump textures, colors, roughness, specularity, refractivity, and opacity.
- Provides physics simulation that allows users to record the physics of an object and apply it as a keyframe animation.
- Allows control over gravity, friction, and bounciness and the ability to adjust the time, quality, and keyframes per second.

KeyShot customers include product and industrial designers, vehicle design companies, jewelers, and architects. It's ideal for the automotive and manufacturing industries.

Why deploy KeyShot on Azure?

- Modern and diverse compute options to align to your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Fast compute capabilities for GPU-intensive workloads

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Windows VM. For information about deploying the VM and installing the drivers, see [Windows VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

Performance tests of KeyShot on Azure used [NVadsA10_v5](#) and [NC4as_T4_v3](#) series VMs running Windows. KeyShot 11 was used in these tests. The following table provides details about the VMs.

[Expand table](#)

VM size	vCPU	Memory (GiB)	Temporary storage SSD (GiB)	GPU partition	GPU memory (GiB)	Maximum data disks
Standard_NV12ads_A10_v5	12	110	360	1/3	8	4

VM size	CPU	Memory (GiB)	Temporary storage	GPU partition	GPU memory (GiB)	Maximum data disks
Standard_NV18ads_A10_v5	8	220 (GiB)	720 SSD (GiB)	1	24	16
Standard_NV36ads_A10_v5	36	440	720	1	24	32
Standard_NV36adms_A10_v5	36	880	720	2	48	32
Standard_NV72ads_A10_v5	72	880	1,400	4	64	32
Standard_NC64as_T4_v3	64	440	2,880			

Required drivers

To take advantage of the GPU capabilities of [NVadsA10_v5](#) and [NC4as_T4_v3](#) series VMs, you need to install NVIDIA GPU drivers.

To use AMD processors on [NVadsA10_v5](#) and [NC4as_T4_v3](#) series VMs, you need to install AMD drivers.

KeyShot installation

Before you install KeyShot, you need to deploy and connect a VM, install an eligible Windows 10 or Windows 11 image, and install the required NVIDIA and AMD drivers.

For information about eligible Windows images, see [How to deploy Windows 10 on Azure](#) and [Use Windows client in Azure for dev/test scenarios](#).

For information about deploying the VM and installing the drivers, see [Run a Windows VM on Azure](#).

For information about installing KeyShot, see the [KeyShot website](#).

KeyShot performance results

Three test case models were used to test the performance of KeyShot on Azure:

Watch configurator



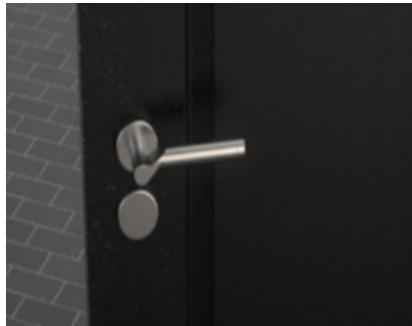
Resolution: 3,556 x 2,000 pixels

Ring configurator



Resolution: 3,556 x 2,000 pixels

Door configurator



Resolution: 3,556 x 2,000 pixels

Performance results for NVads_A10_v5 series

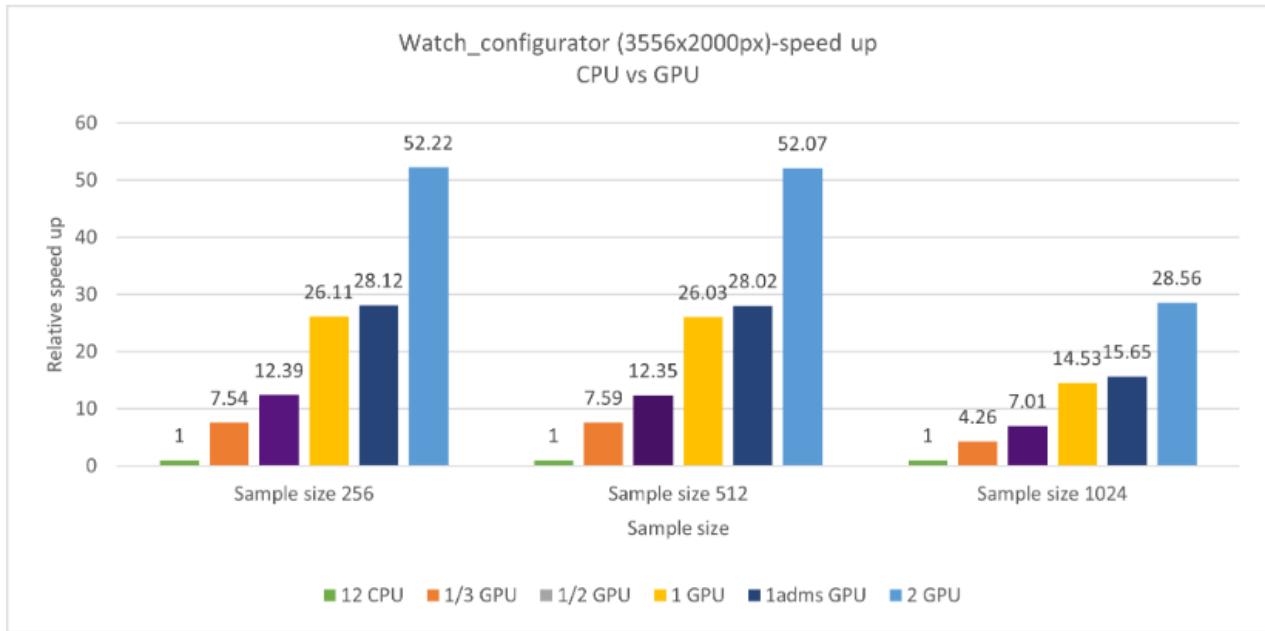
The following sections present the performance results for each model. Rendering times are shown in seconds, for three sample sizes.

Watch configurator

[Expand table](#)

VM size	CPU/GPU	Rendering time, 256	Rendering time, 512	Rendering time, 1024
Standard_NV12ads_A10_v5	12 vCPU	365	728	813
Standard_NV12ads_A10_v5	1/3 GPU	48.44	95.89	191
Standard_NV18ads_A10_v5	1/2 GPU	29.47	58.93	116
Standard_NV36ads_A10_v5	1 GPU	13.98	27.97	55.94
Standard_NV36adms_A10_v5	1 GPU	12.98	25.98	51.95
Standard_NV72ads_A10_v5	2 GPU	6.99	13.98	28.47

This graph shows the relative speed increases as the CPU/GPU increases:

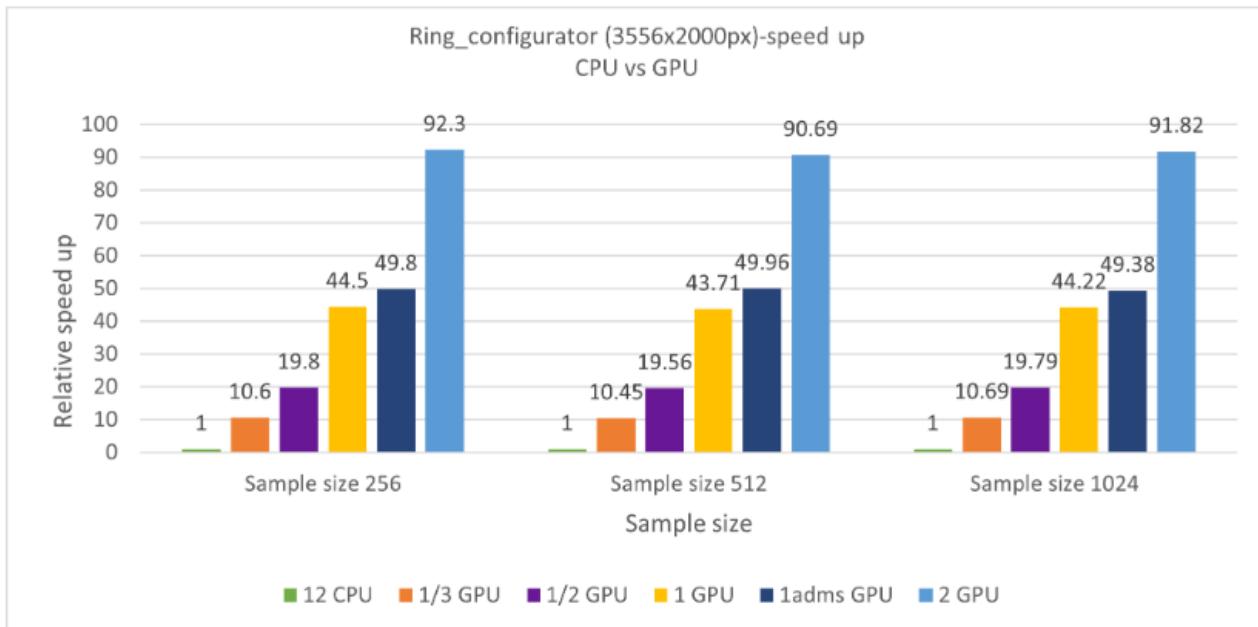


Ring configurator

[Expand table](#)

VM size	CPU/GPU	Rendering time, 256	Rendering time, 512	Rendering time, 1024
Standard_NV12ads_A10_v5	12 vCPU	1,244	2,445	4,908
Standard_NV12ads_A10_v5	1/3 GPU	117	234	459
Standard_NV18ads_A10_v5	1/2 GPU	62.95	125	248
Standard_NV36ads_A10_v5	1 GPU	27.98	55.94	111
Standard_NV36adms_A10_v5	1 GPU	24.98	48.94	99.4
Standard_NV72ads_A10_v5	2 GPU	13.48	26.96	53.45

This graph shows the relative speed increases as the CPU/GPU increases:

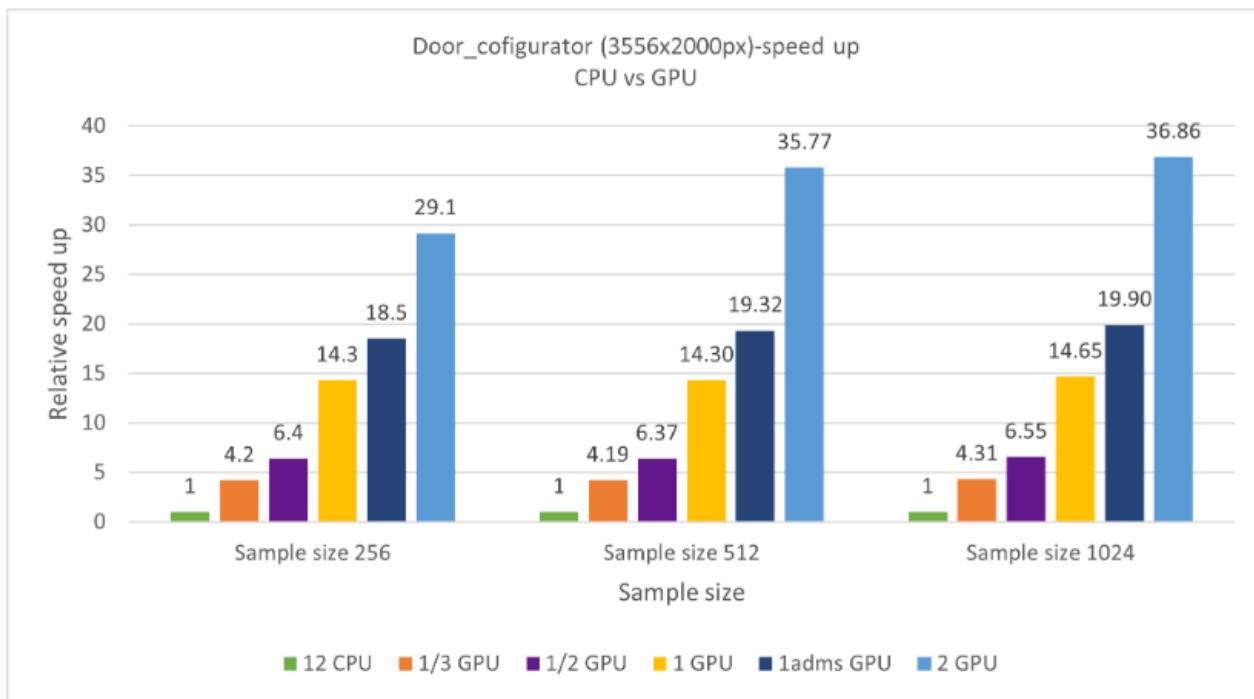


Door configurator

[\[\]](#) Expand table

VM size	CPU/GPU	Rendering time, 256	Rendering time, 512	Rendering time, 1024
Standard_NV12ads_A10_v5	12 vCPU	786	1,573	3,223
Standard_NV12ads_A10_v5	1/3 GPU	188	375	747
Standard_NV18ads_A10_v5	1/2 GPU	123	247	492
Standard_NV36ads_A10_v5	1 GPU	54.97	110	220
Standard_NV36adms_A10_v5	1 GPU	42.45	81.42	162
Standard_NV72ads_A10_v5	2 GPU	26.97	43.97	87.43

This graph shows the relative speed increases as the CPU/GPU increases:



Performance results for NC64as_T4_v3

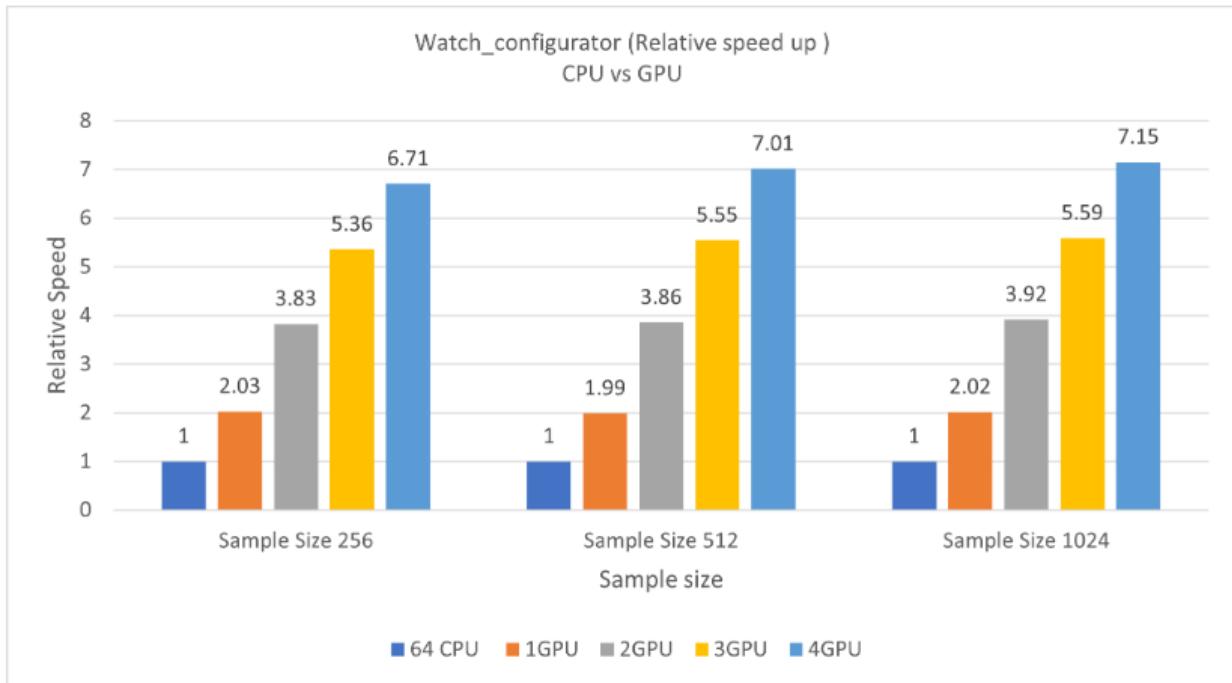
The following sections present the performance results for each model. Rendering times are in seconds, for three sample sizes.

Watch configurator

[Expand table](#)

VM size	CPU/GPU	Rendering time, 256	Rendering time, 512	Rendering time, 1024
Standard_NC64as_T4_v3	64 vCPU	66.92	133	268
Standard_NC64as_T4_v3	1 GPU ¹	32.98	66.93	133
Standard_NC64as_T4_v3	2 GPU ¹	17.48	34.48	68.43
Standard_NC64as_T4_v3	3 GPU ¹	12.49	23.98	47.95
Standard_NC64as_T4_v3	4 GPU	9.98	18.96	37.46

This graph shows the relative speed increases as the CPU/GPU increases:

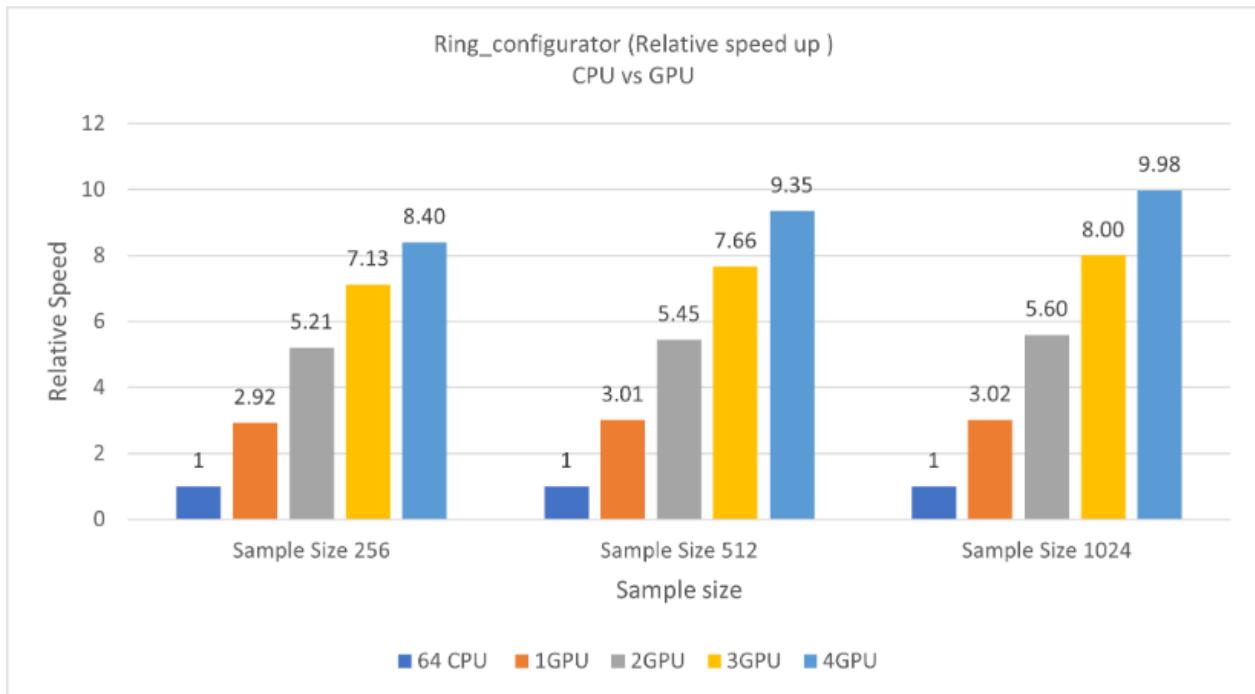


Ring configurator

[Expand table](#)

VM size	CPU/GPU	Rendering time, 256	Rendering time, 512	Rendering time, 1024
Standard_NC64as_T4_v3	64 vCPU	260	509	1,008
Standard_NC64as_T4_v3	1 GPU ¹	88.91	169	334
Standard_NC64as_T4_v3	2 GPU ¹	49.95	93.4	180
Standard_NC64as_T4_v3	3 GPU ¹	36.48	66.43	126
Standard_NC64as_T4_v3	4 GPU	30.96	54.45	101

This graph shows the relative speed increases as the CPU/GPU increases:



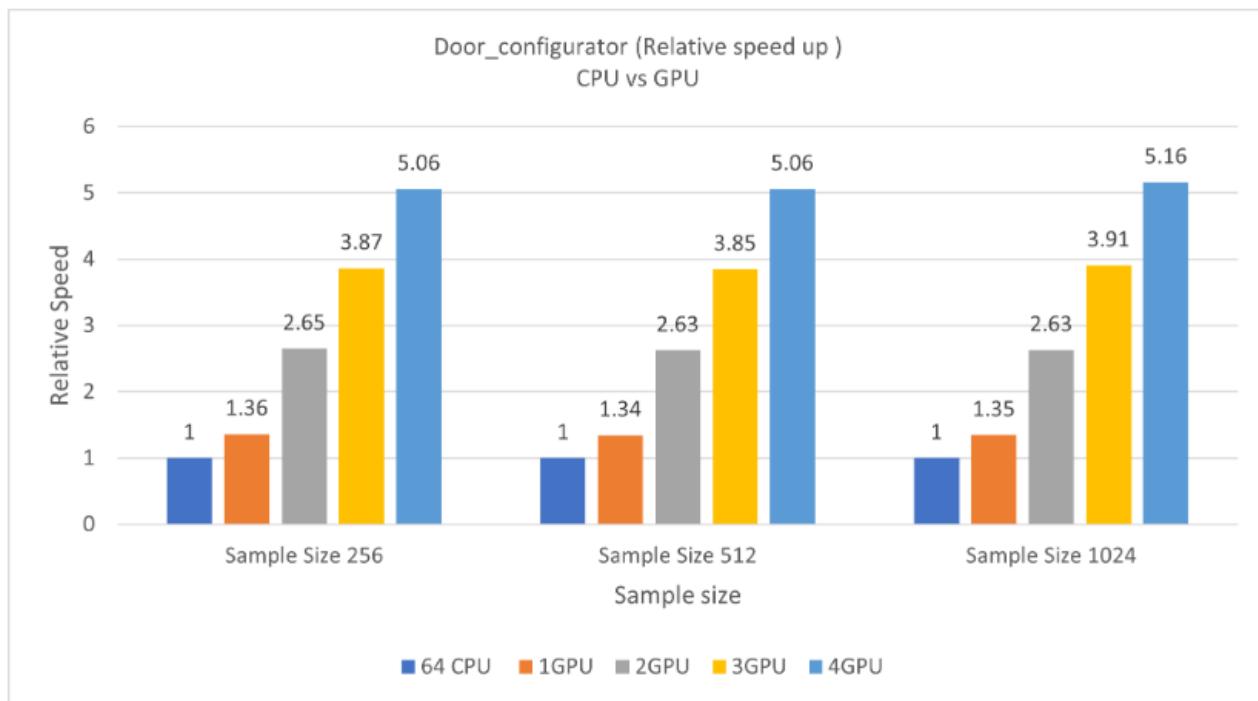
Door configurator

[Expand table](#)

VM size	CPU/GPU	Rendering time, 256	Rendering time, 512	Rendering time, 1024
Standard_NC64as_T4_v3	64 vCPU	139	273	547
Standard_NC64as_T4_v3	1 GPU ¹	102	203	406
Standard_NC64as_T4_v3	2 GPU ¹	52.44	104	208
Standard_NC64as_T4_v3	3 GPU ¹	35.96	70.93	140
Standard_NC64as_T4_v3	4 GPU	27.47	53.96	106

¹ In these cases, the number of GPUs was artificially limited. This VM has four GPUs.

This graph shows the relative speed increases as the CPU/GPU increases:



Azure cost

The following tables provide elapsed times in hours. To compute the total cost, multiply these times by the Azure VM hourly costs for NVads_A10_v5 and NCas_T4_v3 VMs. For the current hourly costs, see [Windows Virtual Machines Pricing](#).

Only model running time (wall-clock time) is considered for these cost calculations. Application installation time isn't considered. The calculations are indicative. The actual numbers depend on the size of the model.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

NVads_A10_v5 series

This table shows elapsed times, in hours, for running all three models.

[] [Expand table](#)

Sample size	12-core CPU	1/3 GPU	1/2 GPU	1 GPU	1 GPU (36adms* VM)	2 GPU
256	0.665	0.098	0.060	0.027	0.022	0.013
512	1.318	0.196	0.120	0.054	0.043	0.024
1024	2.484	0.388	0.238	0.107	0.087	0.047

* This number refers to a Standard_NV36adms_A10_v5 VM configuration.

NCast4_V3 series

This table shows elapsed times, in hours, for running all three models.

 [Expand table](#)

Sample size	64-core CPU	1 GPU	2 GPU	3 GPU	4 GPU
256	0.129	0.062	0.033	0.024	0.019
512	0.254	0.122	0.064	0.045	0.035
1024	0.506	0.243	0.127	0.087	0.068

Summary

- Luxion KeyShot 11 was successfully tested on NVads_A10_v5 and NC64as_T4_v3 VMs.
- The GPU technology in KeyShot 11 provides excellent processing power on Azure.
- Depending on the complexity of the model, the performance improvement as you increase CPU/GPU varies.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Windows VM on Azure](#)

- HPC system and big-compute solutions
- HPC cluster deployed in the cloud

Deploy M-Star on a virtual machine

Azure Virtual Machines Azure Virtual Network

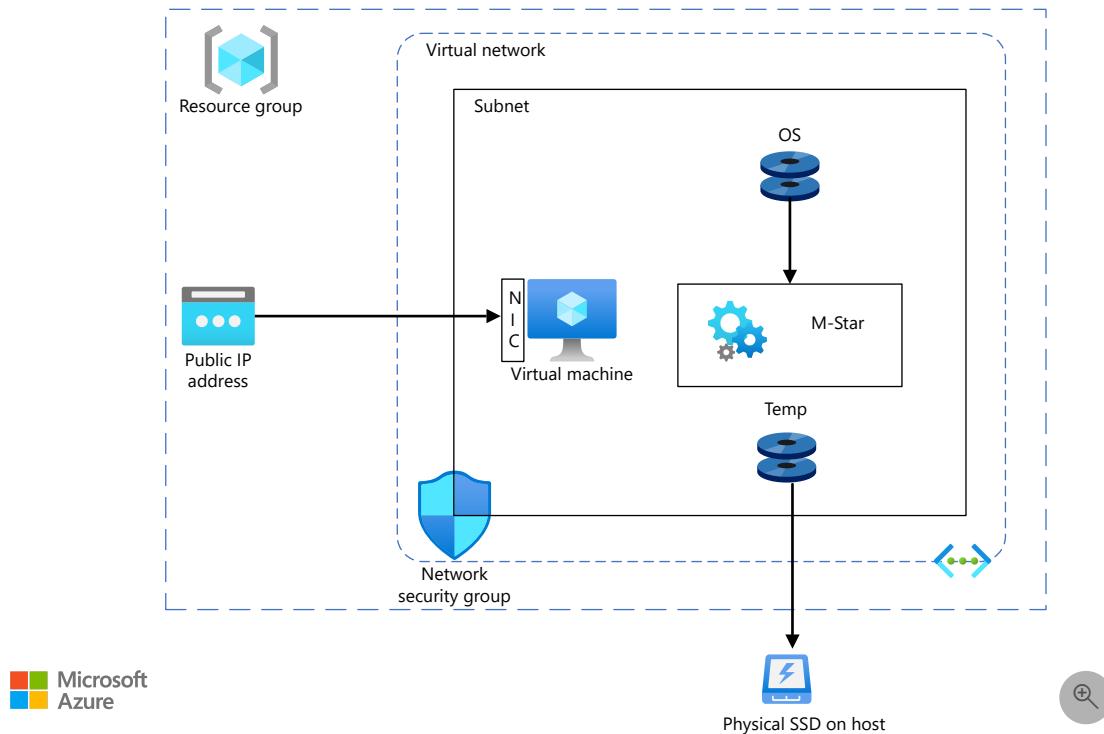
This article briefly describes the steps for running [M-Star](#) computational fluid dynamics software on an Azure virtual machine (VM). It also presents the performance results of running M-Star on Azure.

M-Star is a multiphysics modeling package that simulates fluid flow, heat transfer, species transport, chemical reactions, particle transport, and rigid-body dynamics. It uses large eddy simulation and advanced lattice-Boltzmann algorithms that run entirely on GPUs. M-Star is used in the chemical, biopharmaceutical, and energy industries.

Why deploy M-Star on Azure?

- Modern and diverse compute options to align with your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Technology that enables the creation of complex flow fields in a short amount of time
- Integrated post-processing capabilities, such as creating photorealistic renderings

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create Windows and Linux VMs. For information about deploying VMs and installing drivers, see [Windows VMs on Azure](#) and [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - **Network security groups** are used to restrict access to the VMs.
 - A public IP address connects the internet to the VMs.
- A physical SSD is used for storage.

Compute sizing and drivers

For performance tests of M-Star on [NDm A100 v4](#) and [NC A100 v4](#) series Azure VMs, the Linux operating system was used. The following table provides the configuration details of these VMs.

[Expand table](#)

Size	vCPU	Memory: GiB	Temporary storage (SSD): GiB	GPU	GPU memory: GiB	Maximum data disks	Maximum uncached disk throughput: IOPS / MBps	Maximum network bandwidth
Standard_NC48ads_A100_v4	48	440	2,246	2	160	24	60,000 / 2,000	40,000 Mbps
Standard_ND96amsr_A100_v4	96	1,900	6,400	8 A100 80-GB GPUs (NVLink 3.0)	80	32	80,000 / 800	24,000 Mbps

Required drivers

To take advantage of the GPU capabilities of [NC A100 v4](#) and [NDm A100 v4](#) series VMs, you need to install NVIDIA GPU drivers.

M-Star installation

Before you install M-Star, you need to deploy and connect to a VM and install the required NVIDIA drivers.

For information about deploying the VM and installing the drivers, see one of these articles:

- [Run a Windows VM on Azure](#)
- [Run a Linux VM on Azure](#)

Important

NVIDIA Fabric Manager is required for VMs that use NVLink or NVSwitch.

The following table provides details about the operating system and NVIDIA drivers that were used for the performance tests.

[Expand table](#)

Operating system version	OS architecture	GPU driver version	CUDA version	MPI
Linux (Ubuntu HPC 18.04 Gen 2)	x86-64	510.85.02	11.6	openmpi-4.1.1

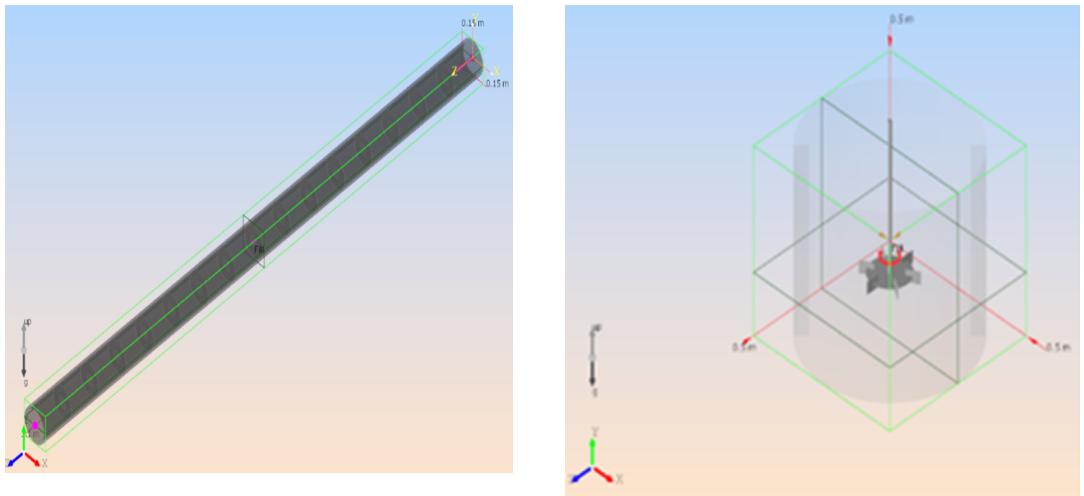
You can install M-Star from the [M-Star installation page](#). For information about the installation process, see [M-Star on Linux](#).

M-Star performance results

This performance analysis uses M-Star 3.8.27 on the Windows operating system. [NC A100 v4](#) and [NDm A100 v4](#) series VMs were used.

Two models were used to test the performance of M-Star on Azure VMs:

[Expand table](#)



Model	Pipe_500	Tank_1000
Number of grid points	500 million	1,000 million

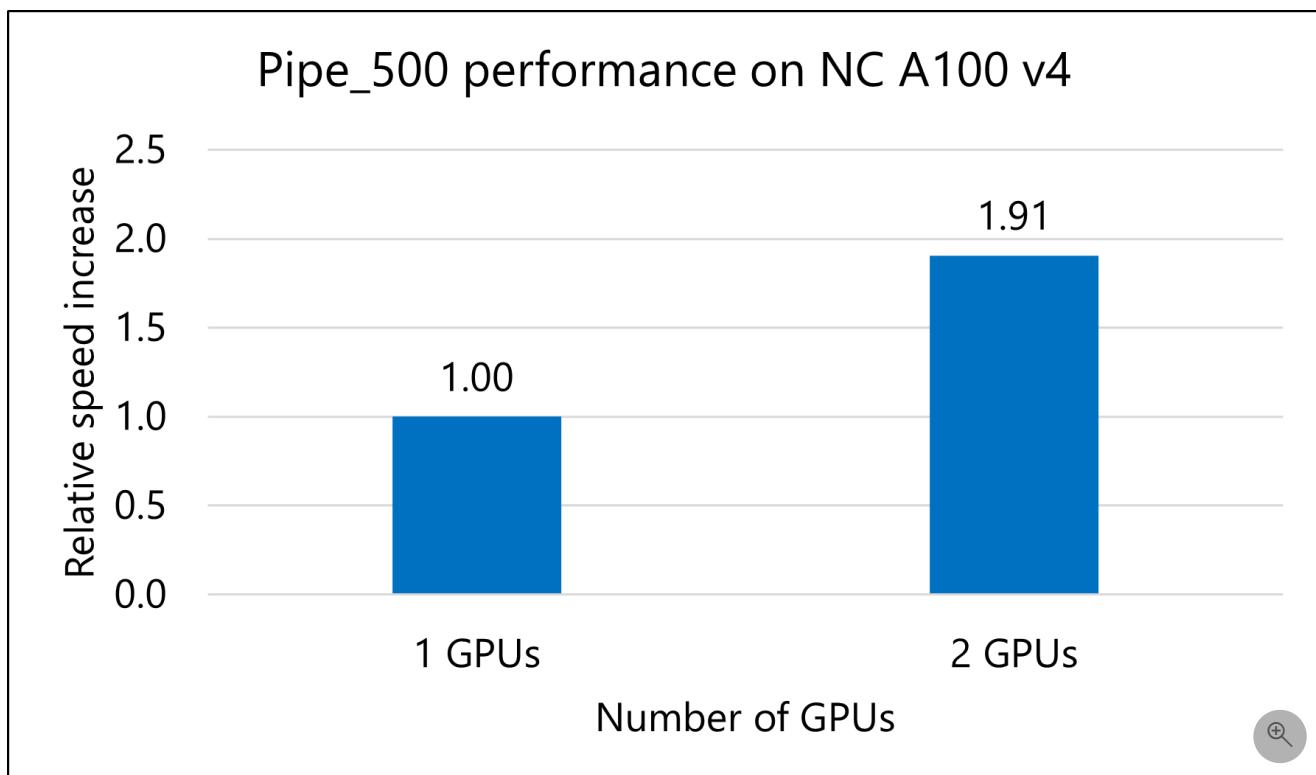
Results on NC A100 v4

Pipe_500 model

The following table shows the total runtimes and the relative speed increase as the number of GPUs is increased from one to two.

[Expand table](#)

VM size	Number of GPUs	Total runtime, in seconds	Speed increase
Standard_NC96ads_A100_v4	1	15,921.18	NA
Standard_NC96ads_A100_v4	2	8,347.98	1.91



Tank_1000 model

The following table shows the elapsed time for running the Tank_1000 model.

[Expand table](#)

VM Size	Number of GPUs	Total runtime, in seconds
Standard_NC96ads_A100_v4	1	NA
Standard_NC96ads_A100_v4	2	1,420.25

Notes about tests on NC A100 v4

- Because the Tank-1000 model is large, you can't run it on one GPU on the NCv4 VM.
- An NVLink connection is required for M-Star.
- Because the architecture of NCv4 VMs supports only dual GPU connectivity, models were run only on 1-GPU and 2-GPU configurations.

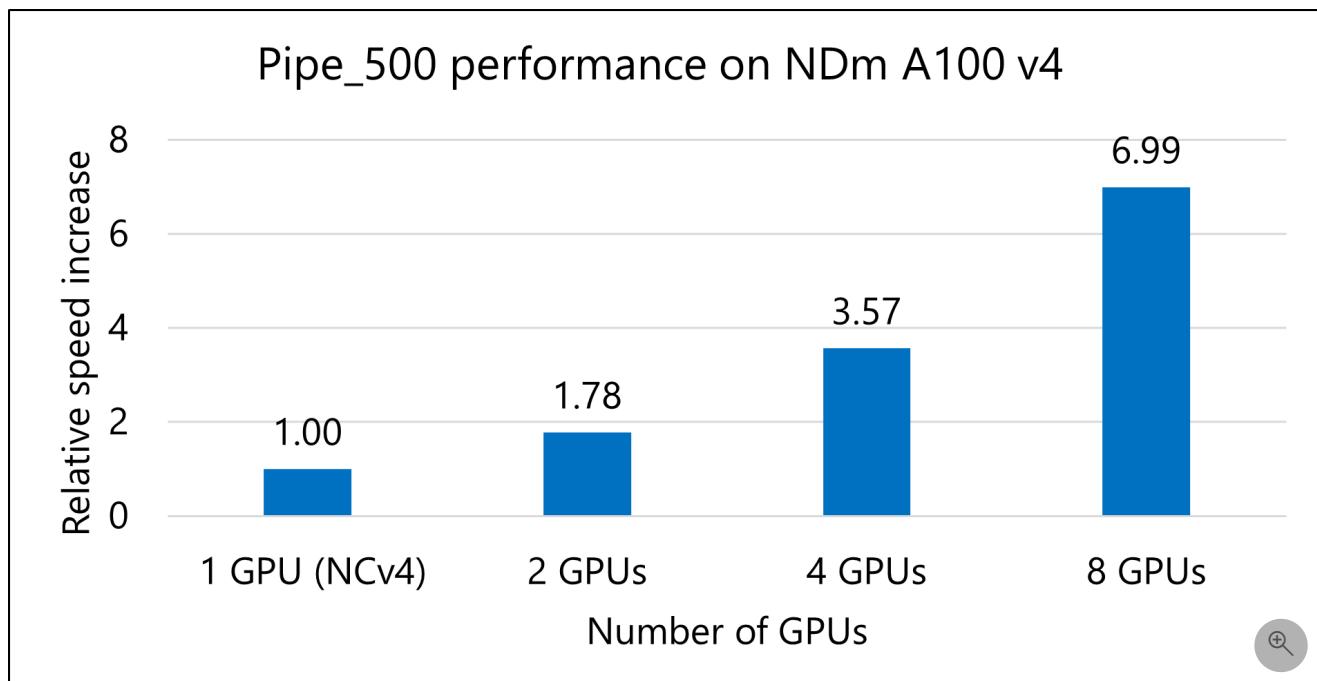
Results on NDm A100 v4

Pipe_500 model

The following table shows the total runtimes and relative speed increases for running the Pipe_500 model.

[Expand table](#)

VM size	Number of GPUs	Total runtime, in seconds	Speed increase
Standard_NC96ads_A100_v4	1	15,921.18	NA
Standard_ND96amsr_A100_v4	2	8,967.48	1.78
Standard_ND96amsr_A100_v4	4	4,463.21	3.57
Standard_ND96amsr_A100_v4	8	2,276.67	6.99

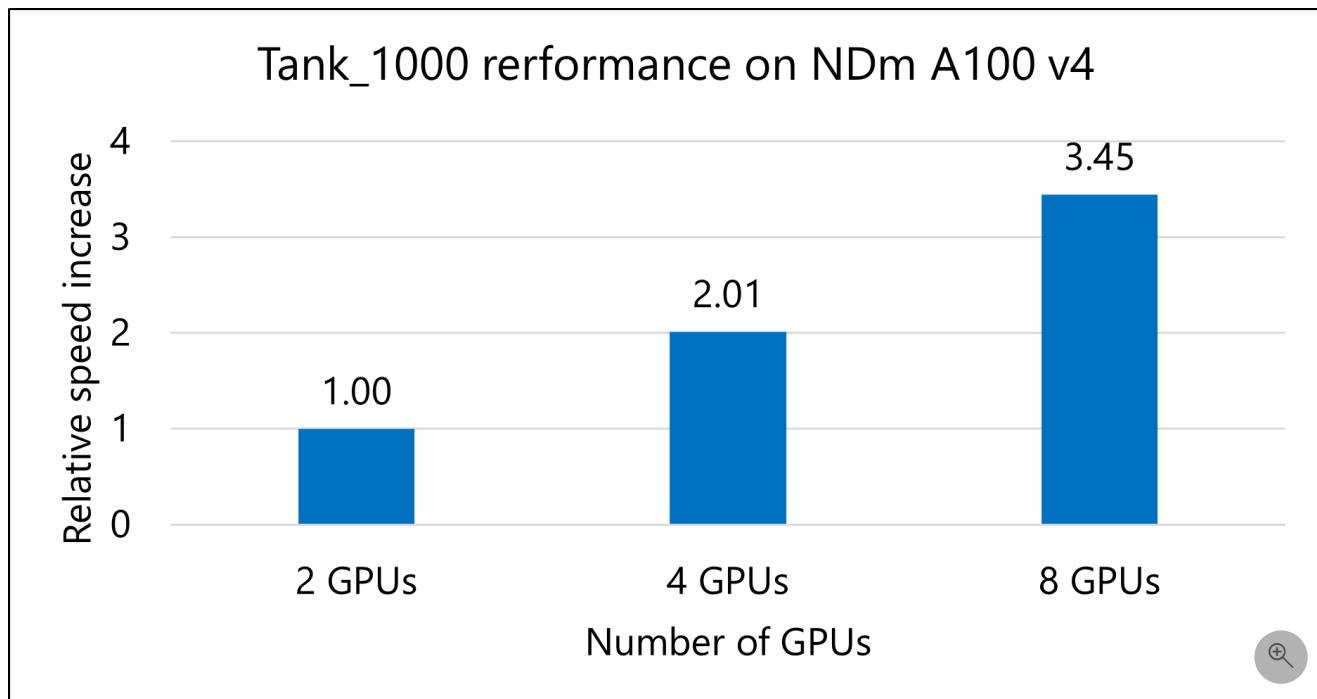


Tank_1000 model

The following table shows the total runtimes and relative speed increases for running the Tank_1000 model.

[Expand table](#)

VM size	Number of GPUs	Total runtime, in seconds	Speed increase
Standard_ND96amsr_A100_v4	2	1,481.36	NA
Standard_ND96amsr_A100_v4	4	735.31	2.01
Standard_ND96amsr_A100_v4	8	429.69	3.45



Notes about tests on NDm A100 v4

- NC A100 v4 series VMs only have individual pairs of GPUs connected peer to peer, but NDm A100 v4 series VMs have full peer-to-peer connections among all eight GPUs. You should therefore use NC A100 v4 systems for simulations that run on one or two GPUs. For anything that needs more than two GPUs, you should use NDmA100 v4 VMs.
- For the Pipe_500 model, the 1-GPU NCv4 result is used as a baseline.

Azure cost

Only model running time (wall-clock time) is considered for these cost calculations. Application installation time isn't considered. The results are indicative of your potential results. The actual numbers depend on the size of the model.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

The following tables provide elapsed times in hours. To compute the total cost, multiply by the Azure VM hourly cost, which you can find [here for Windows](#) and [here for Linux](#).

Cost for the Pipe_500 model

NDm A100 v4

[Expand table](#)

Number of GPUs	Time in, hours
8	0.63

NC A100 v4

[Expand table](#)

Number of GPUs	Time in, hours
1	4.42
2	2.31

Cost for the Tank_1000 model

NDm A100 v4

[Expand table](#)

Number of GPUs	Time in, hours
8	0.11

NC A100 v4

[Expand table](#)

Number of GPUs	Time in, hours
1	NA
2	0.39

Summary

- M-Star 3.8.27 was successfully tested on NC A100 v4 and NDM A100 v4 VMs on Azure.
- Based on the models tested, M-Star scales almost linearly as the number of GPUs increases.
- For the Pipe_500 model, tests indicate that the speed with eight GPUs is seven times faster than the speed with one GPU.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director, Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run HPC applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [Run a Windows VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy NAMD on a virtual machine

Azure Virtual Machines Azure Virtual Network

This article describes the steps for running [NAMD](#) software on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running NAMD on single-node and multi-node VM configurations.

NAMD is a computer application for molecular dynamics simulation that's based on the [Charm++](#) parallel programming model. It's often used to simulate systems that comprise millions of atoms. NAMD supports hundreds of cores for typical simulations and can support more than 500,000 cores for the largest simulations. Simulations and trajectory analysis are performed with the popular molecular graphics program VMD, but NAMD is also compatible with AMBER, CHARMM, and X-PLOR. A source code version of NAMD is available for free.

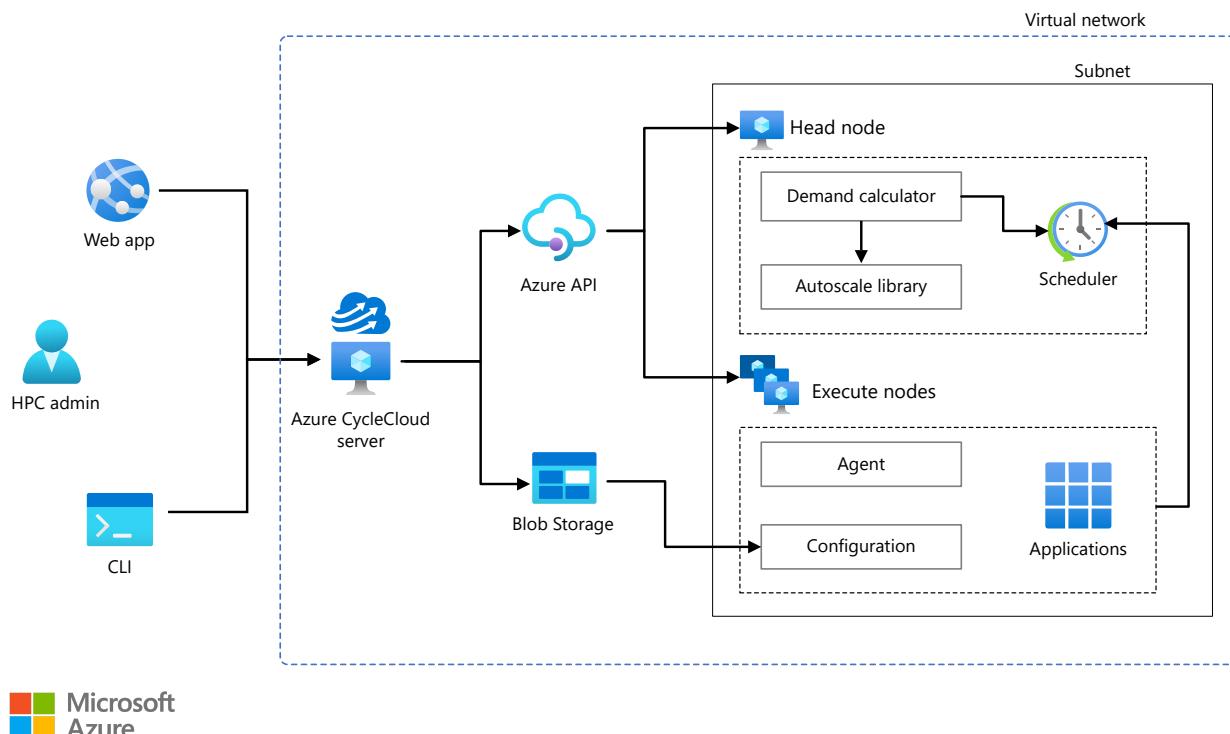
NAMD is used mainly for high-performance simulations of large biomolecular systems. Typical NAMD simulations include all-atom models of proteins, lipids, nucleic acids, and explicit solvents (water and ions).

Why deploy NAMD on Azure?

- By running NAMD on Azure HB-series VMs, you can reduce the time and cost of your simulations.
- Running molecular simulation and analysis tasks on Azure can make it easier to implement advanced simulation methods and practical solutions for many molecular modeling tasks.
- NAMD and associated tools enable popular research workflows like MDFF structure refinement and QwikMD simulation protocols to be run remotely. You don't need to invest in local computing resources, and the required expertise in high-performance computing (HPC) technologies is reduced.

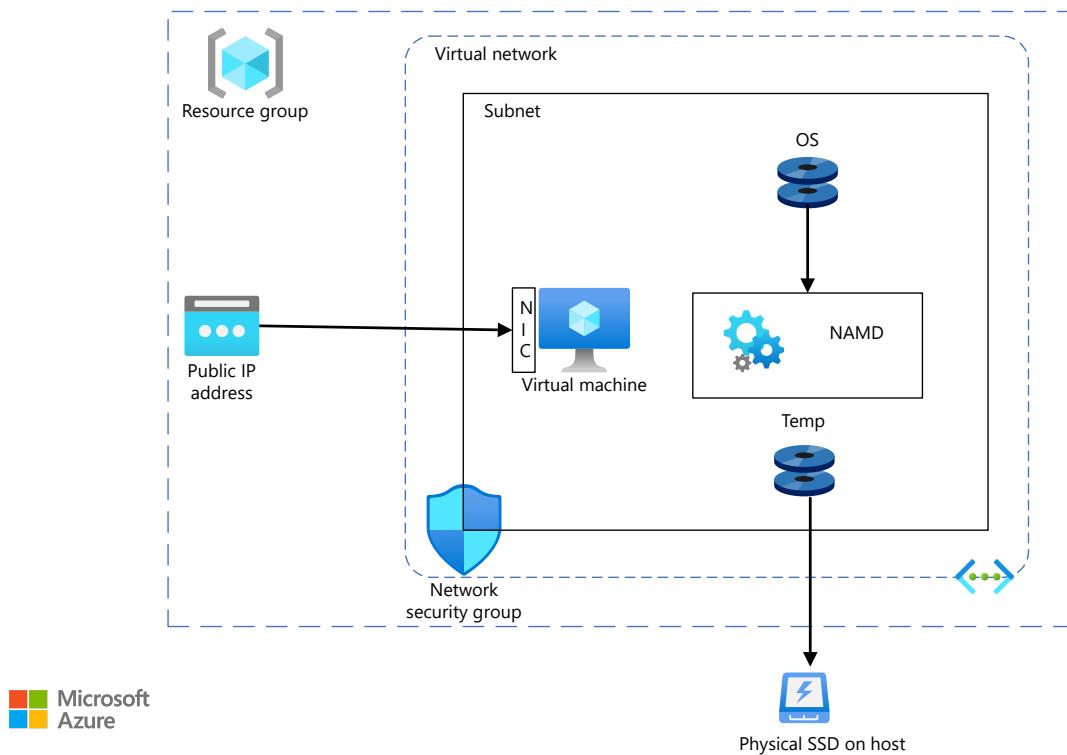
Architecture

This diagram shows a multi-node configuration:



Download a [Visio file](#) of this architecture.

This diagram shows a single-node configuration:



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create Linux VMs. For information about deploying VMs and installing the drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VMs.
 - A public IP address connects the internet to the VMs.
- [Azure CycleCloud](#) is used to create the cluster in the multi-node configuration.
- A physical SSD is used for storage.

Compute sizing and drivers

HBv3 AMD EPYC 7V73X (Milan-X) VMs running Linux CentOS were used to test the performance of NAMD on Azure. The following table provides details about HBv3-series VMs.

[Expand table](#)

Size	vCPU	Memory (GiB)	Memory bandwidth (Gbps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120rs_v3	120	448	350	1.9	3.0	3.5	200	32
Standard_HB120-96rs_v3	96	448	350	1.9	3.0	3.5	200	32
Standard_HB120-64rs_v3	64	448	350	1.9	3.0	3.5	200	32
Standard_HB120-32rs_v3	32	448	350	1.9	3.0	3.5	200	32

Size	vCPU	Memory (GiB)	Memory bandwidth (GBps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (GBps)	Maximum data disks
Standard_HB120-16rs_v3	16	448	350	1.9	3.0	3.5	200	32

Required drivers

To use InfiniBand, you need to enable [InfiniBand](#) drivers.

Install NAMD 2.14 on a VM or HPC cluster

You can download the software from the [NAMD](#) website. You just need to untar or unzip the NAMD binary distribution file and run it in the resulting directory. For information about building from source code, see [Compiling NAMD](#).

Before you install NAMD, you need to deploy and connect to a VM or an HPC cluster.

For information about deploying the VM and installing the drivers, see [Run a Linux VM on Azure](#).

For information about deploying Azure CycleCloud and the HPC cluster, see these articles:

- [Install and configure Azure CycleCloud](#)
- [Create an HPC cluster](#)

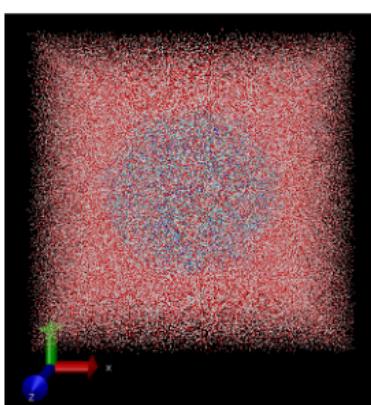
NAMD performance results

Two models were used to test the scalability performance of NAMD 2.14 on Azure:

- STMV. A small icosahedral plant virus that worsens the symptoms of infection by tobacco mosaic virus (TMV).
- ApoA1. A component of high-density lipoprotein (HDL). The ApoA1 gene provides instructions for creating a protein called apolipoprotein A-I.

The details about each test model are provided in the following sections.

Model 1: STMV



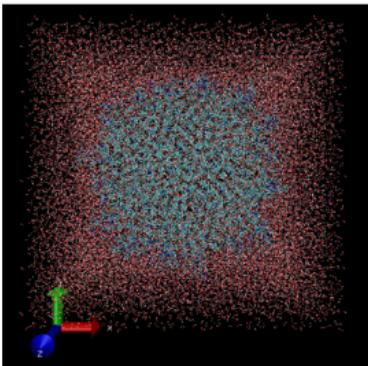
To validate NAMD 2.14 scaling on Azure HPC systems, STMV is tiled in arrays of 5x2x2 (21 million atoms). The following table provides details about two versions of the model.

[Expand table](#)

Model	Number of atoms	Time step	Number of steps	Method
1a	1,066,628	1	2,000	Particle-Mesh Ewald (PME)

Model	Number of atoms	Time step	Number of steps	Method
1b	21,000,000	2	1,200	PME

Model 2: ApoA1



The following table provides details about the model.

[Expand table](#)

Model	Number of atoms	Time step	Number of steps	Method
2	92,224	1	2,000	PME

NAMD 2.14 performance results on single-node VMs

The following sections provide the performance results of running NAMD on single-node Azure [HBv3 AMD EPYC 7V73X \(Milan-X\)](#) VMs.

Model 1a: STMV

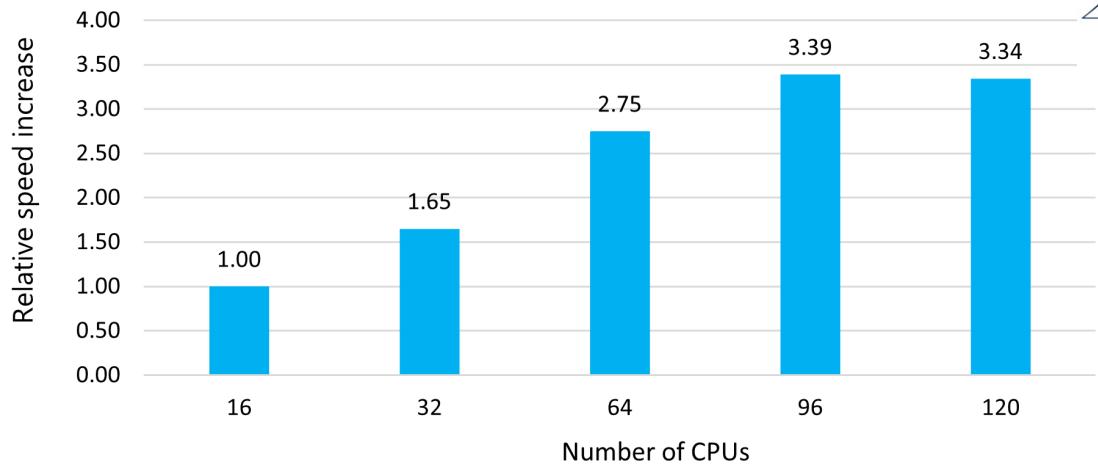
This table shows the nanoseconds per day and total wall-clock times recorded for varying numbers of CPUs on the Standard HBv3-series VM:

[Expand table](#)

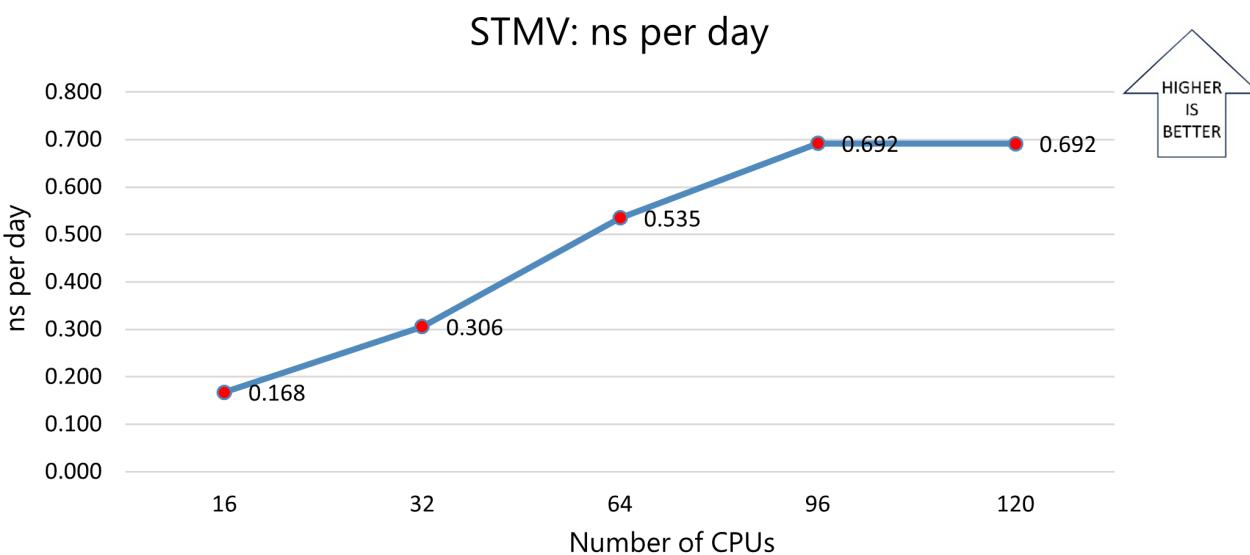
Number of cores	ns per day	Wall-clock time (seconds)	Relative speed increase
16	0.168	1,046.34	NA
32	0.306	633.58	1.65
64	0.535	380.58	2.75
96	0.692	308.60	3.39
120	0.692	312.93	3.34

The following graph shows the relative speed increases as the number of CPUs increases:

STMV: Relative speed increase



The following graph shows the nanoseconds-per-day for varying numbers of CPUs:



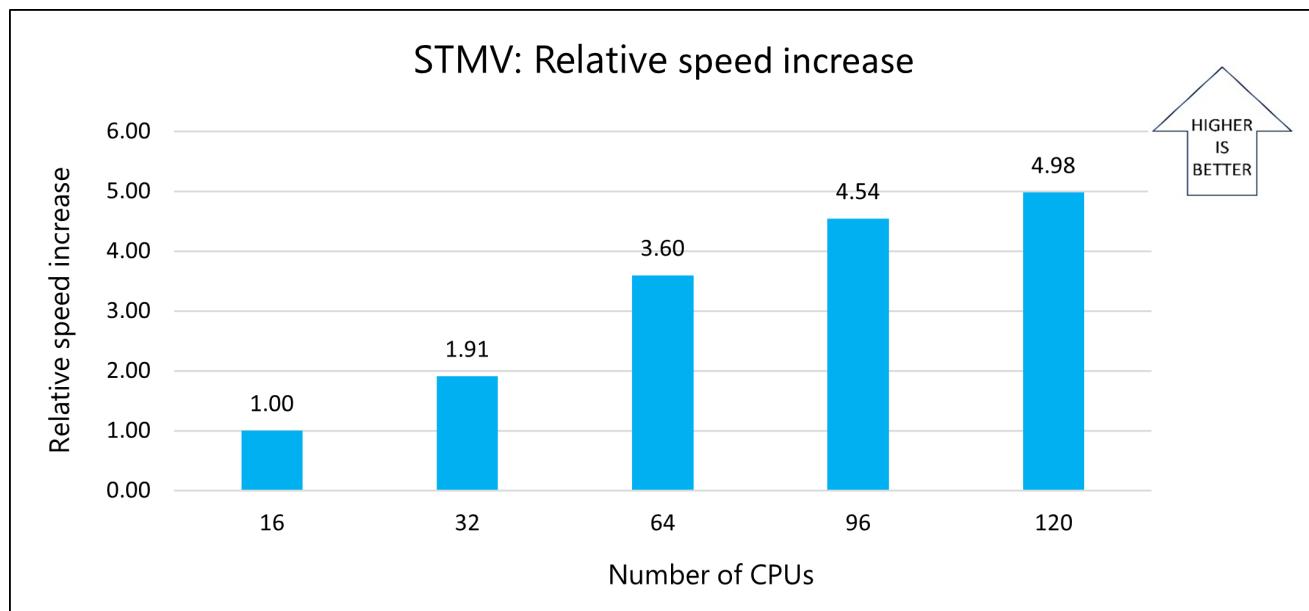
Model 1b: STMV

This table shows the nanoseconds per day and total wall-clock times recorded for varying numbers of CPUs on the Standard HBv3-series VM:

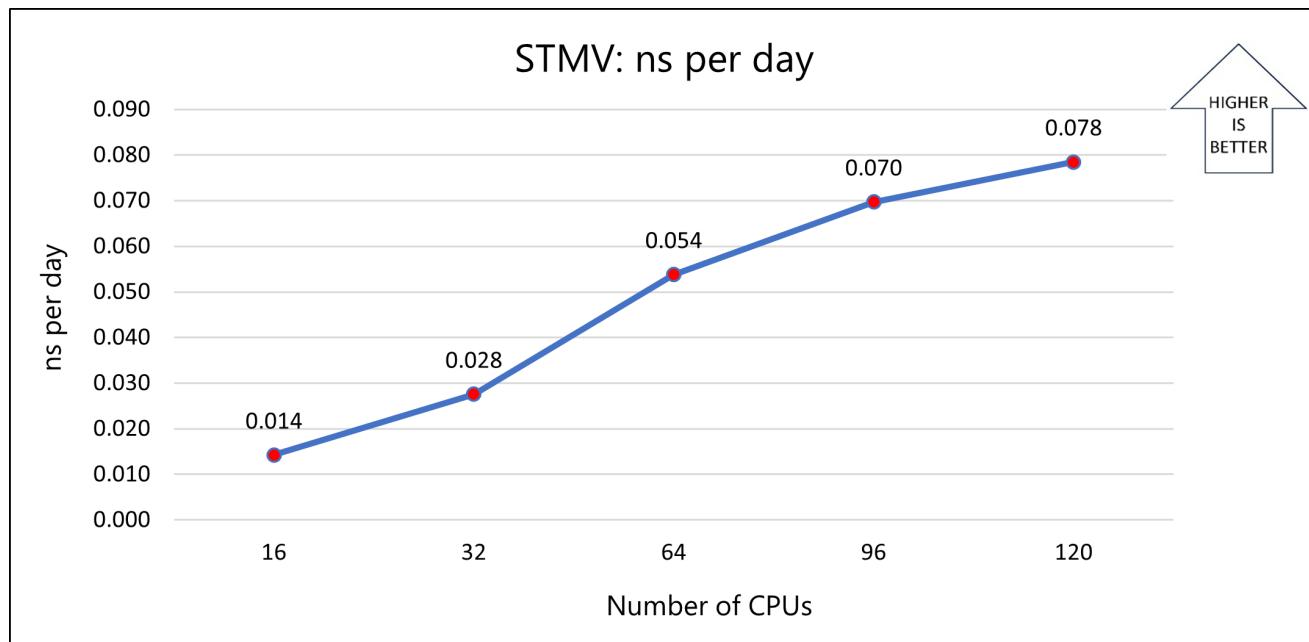
[Expand table](#)

Number of cores	ns per day	Wall-clock time (seconds)	Relative speed increase
16	0.014	14,712.03	NA
32	0.028	7,715.94	1.91
64	0.054	4,092.31	3.60
96	0.070	3,239.06	4.54
120	0.078	2,955.45	4.98

The following graph shows the relative speed increases as the number of CPUs increases:



The following graph shows the nanoseconds-per-day for varying numbers of CPUs:



Model 2: ApoA1

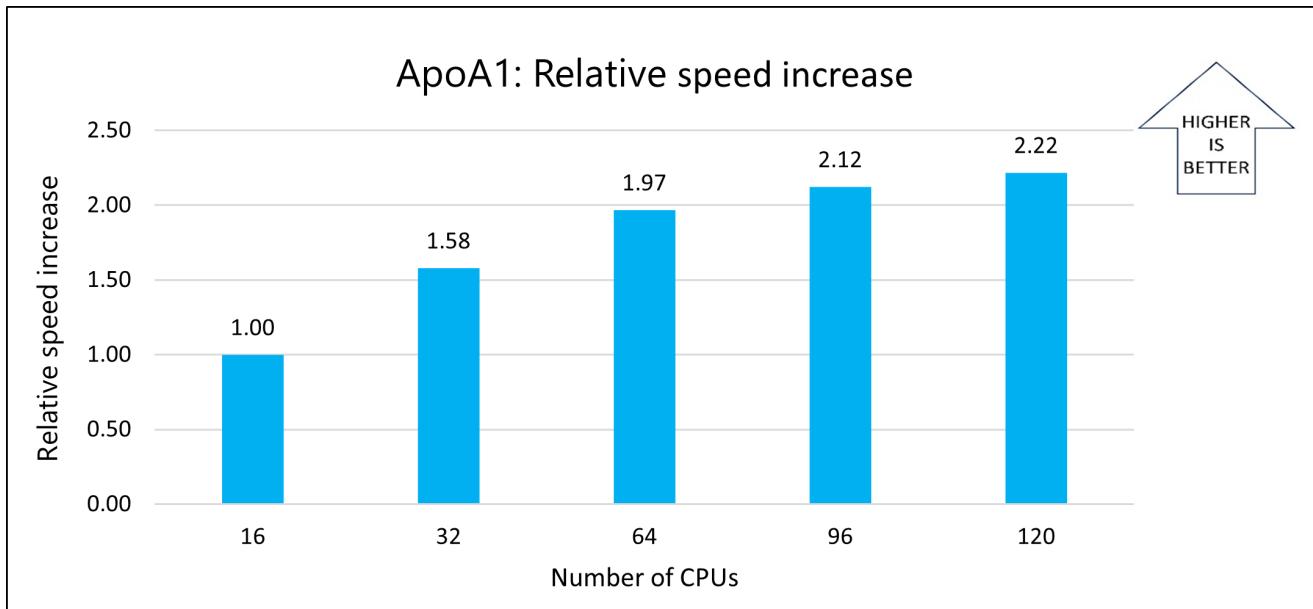
This table shows the nanoseconds per day and total wall-clock times recorded for varying numbers of CPUs on the Standard HBv3-series VM:

[Expand table](#)

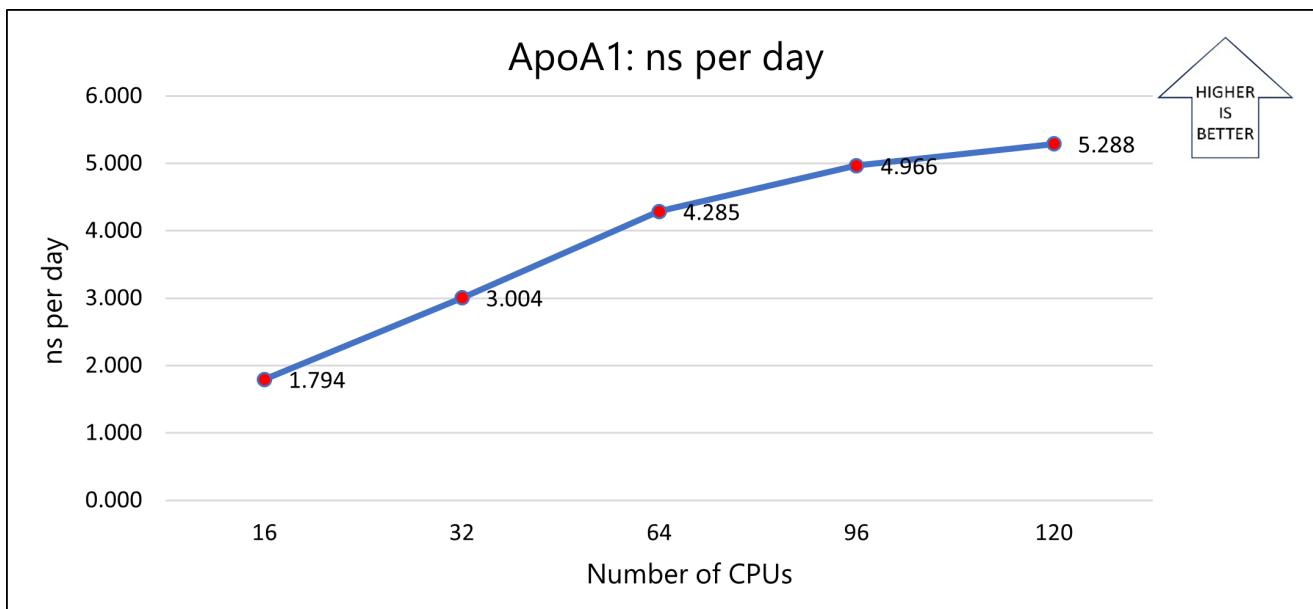
Number of cores	ns per day	Wall-clock time (seconds)	Relative speed increase
16	1.794	130.61	NA
32	3.004	82.67	1.58
64	4.285	66.36	1.97
96	4.966	61.51	2.12

Number of cores	ns per day	Wall-clock time (seconds)	Relative speed increase
120	5.288	58.89	2.22

The following graph shows the relative speed increases as the number of CPUs increases:



The following graph shows the nanoseconds per day for varying numbers of CPUs:



Notes about the single-node tests

For all single-node tests, the solver time on a `Standard_HB120-16rs_v3` VM (16 cores) is used as a reference to calculate the relative speed increase with respect to similar VMs that have more cores. The previously presented results show that parallel performance improves as cores increase from 16 to 64. At 120 cores, the improvement is limited and only occurs on some simulations. This pattern is common with these simulations and other memory-intensive applications because of the saturation of the onboard memory that's available on each processor. Taking VM costs into consideration, the 64-CPU configuration is the best choice. `Standard_HB120-64rs_v3` VMs, which have 64 cores, were used for the multi-node tests.

NAMD 2.14 performance results on a multi-node cluster

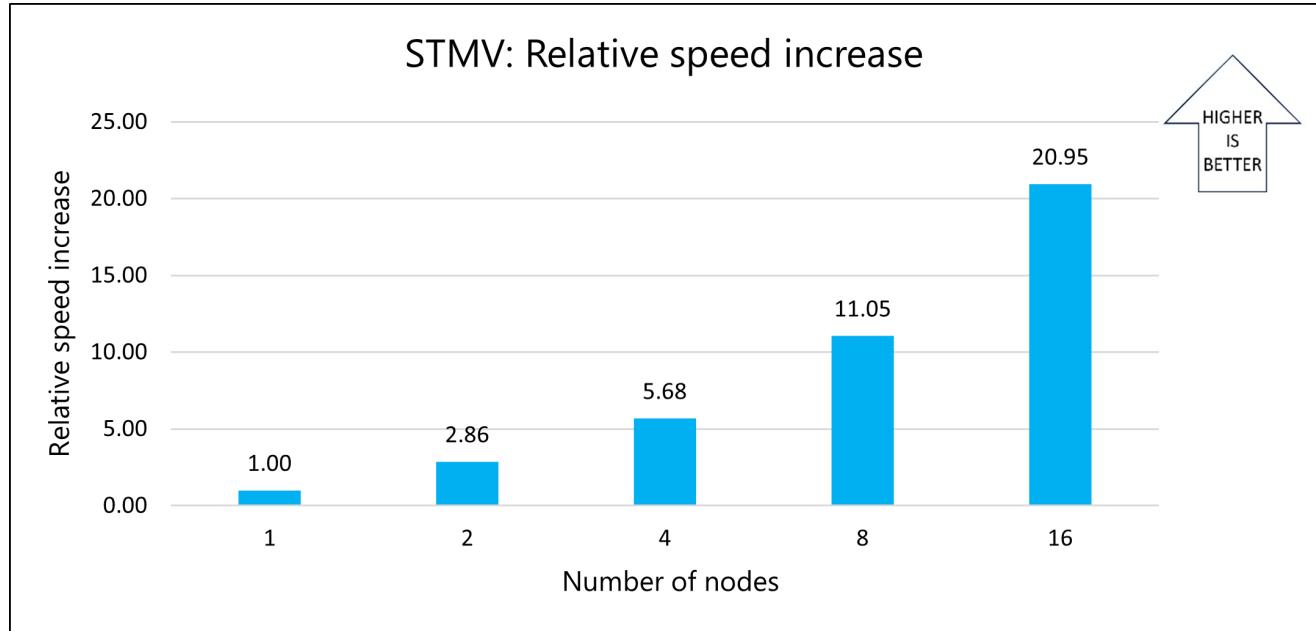
The single-node tests confirm that the solver achieves optimal parallel performance with 64 cores on HBv3-series VMs. Based on those results, 64-core configurations on `Standard_HB120-64rs_v3` VMs were used to evaluate the performance of NAMD on multi-node clusters. STMV model 1b was used for the multi-node tests.

This table shows the nanoseconds per day and total wall-clock times recorded for varying numbers of nodes on Standard HBv3-series VMs:

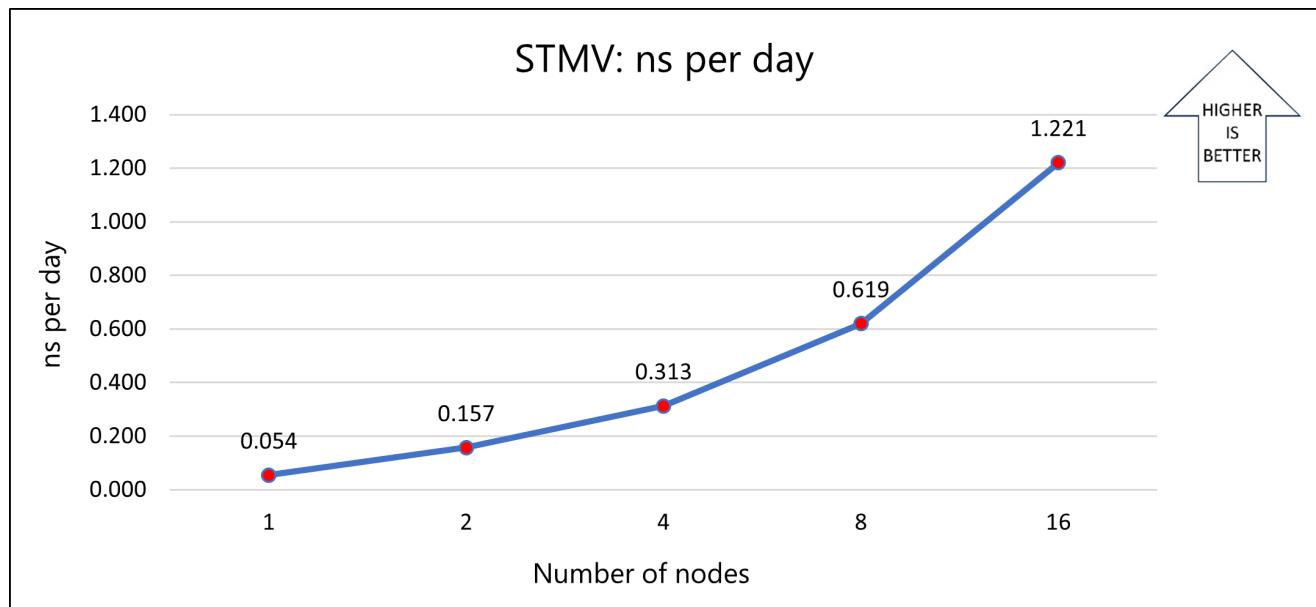
[Expand table](#)

Number of nodes	Number of cores	ns per day	Wall-clock time (seconds)	Relative speed increase
1	64	0.054	3,835.48	NA
2	128	0.157	1,340.54	2.86
4	256	0.313	675.18	5.68
8	512	0.619	346.97	11.05
16	1,024	1.221	183.09	20.95

The following graph shows the relative speed increases as the number of nodes increases:



The following graph shows the nanoseconds per day for varying numbers of nodes:



The results show that model 1b scales well as the number of nodes increases. For better performance, use a memory-optimized version of the application. You can optimize the application by compiling the source code. The simulation used for testing is limited to a few iterations. Because real-world applications can use more iterations, you can minimize the total time that's required for decomposition, which further improves performance.

Azure cost

The following table provides wall-clock times that you can use to calculate Azure costs. To compute the cost, multiply the solver running time by the number of nodes and the Azure VM hourly cost. For the current hourly costs, see [Linux Virtual Machines Pricing](#). The Azure VM hourly rates are subject to change.

Only simulation running time is considered for the cost calculations. Installation time, simulation setup time, and software costs aren't included.

You can use the [Azure pricing calculator](#) to estimate VM costs for your configurations.

STMV model 1b was used to calculate these times.

 [Expand table](#)

Number of nodes	Wall-clock time (hours)
1	1.07
2	0.37
4	0.19
8	0.10
16	0.05

Summary

- NAMD 2.14 was successfully tested on Azure HBv3 standalone VMs and on an Azure CycleCloud multi-node configuration.
- Model 1b scales well on the multi-node configuration. On a 16-node configuration, the speed is 21 times faster than it is on a single node.
- For better performance, we recommend that you use the `+p` option to run one thread per processor. We also recommend that you look for prebuilt ibverbs NAMD binaries or specify ibverbs when you build the Charm++.
- For small simulations, we recommend that you use fewer CPUs.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Preetham Y M](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director, Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run HPC applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy OpenFOAM on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [OpenFOAM](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running OpenFOAM on Azure.

OpenFOAM is a free, open-source computational fluid dynamics (CFD) application. Users have permission to modify and compile the package based on the needs and the physics of the problem they're solving.

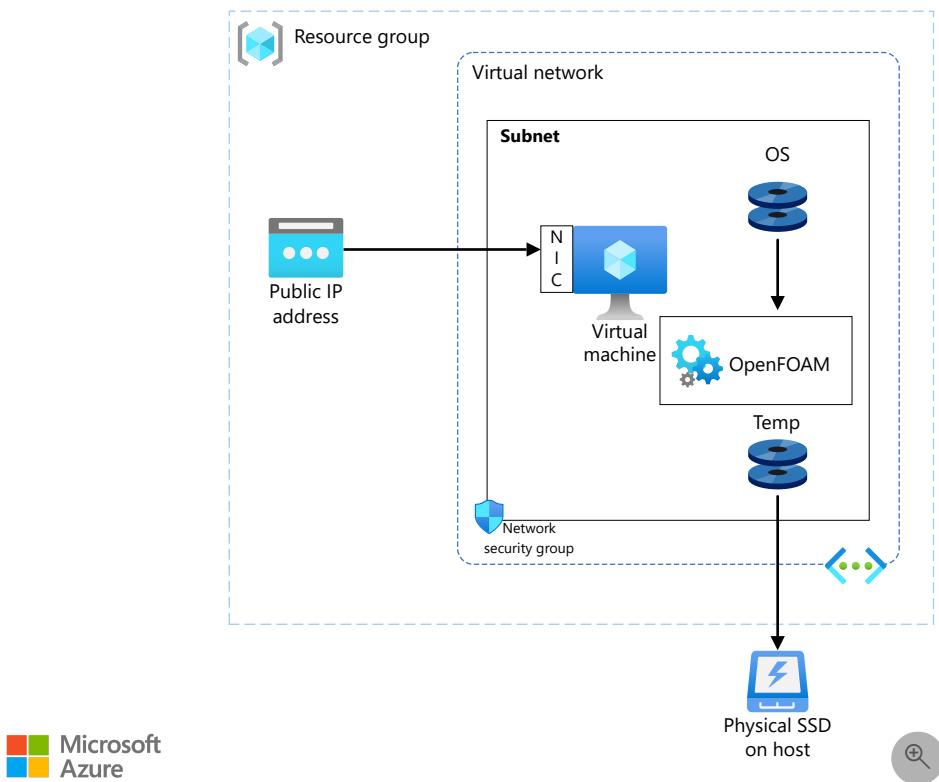
The software is a C++ toolbox for the development of customized numerical solvers. It uses explicit methods for configuring a simulation by selecting numerical schemes, solvers and their parameters, and algorithm controls.

OpenFOAM is used in academia/education and in industries like transportation, automotive, manufacturing, and healthcare.

Why deploy OpenFOAM on Azure?

- Modern and diverse compute options to align to your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Appreciable speed increase as cores increase

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Linux VM. For information about deploying the VM and installing the drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

Performance tests of OpenFOAM on Azure used [HBv3-series](#) VMs running Linux. The following table provides details about the VMs.

[Expand table](#)

VM size	vCPU	Memory (GiB)	Memory bandwidth Gbps	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)	Temp storage (GiB)	Max data
Standard_HB120rs_v3	120	448	350	1.9	3.0	3.5	200	2 * 960	32
Standard_HB120-96rs_v3	96	448	350	1.9	3.0	3.5	200	2 * 960	32
Standard_HB120-64rs_v3	64	448	350	1.9	3.0	3.5	200	2 * 960	32
Standard_HB120-32rs_v3	32	448	350	1.9	3.0	3.5	200	2 * 960	32
Standard_HB120-16rs_v3	16	448	350	1.9	3.0	3.5	200	2 * 960	32

The HBv3 (Milan-X) VMs are optimized for HPC applications like fluid dynamics, explicit and implicit finite element analysis, weather modeling, seismic processing, reservoir simulation, and RTL simulation.

Required drivers

To use AMD CPUs on [HBv3](#) VMs, you need to install AMD drivers.

To use InfiniBand, you need to enable InfiniBand drivers.

OpenFOAM installation

Before you install OpenFOAM, you need to deploy and connect a Linux VM and install the required AMD and InfiniBand drivers.

For information about deploying the VM and installing the drivers, see [Run a Linux VM on Azure](#).

For information about installation and about the various versions of OpenFOAM, see this [OpenFOAM Development page](#).

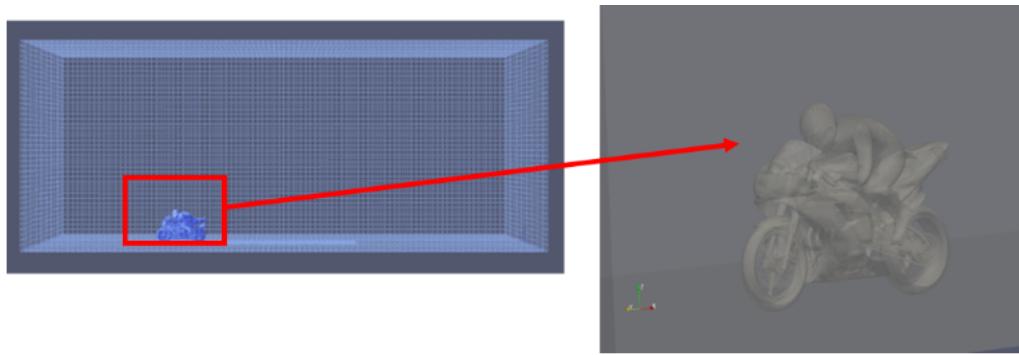
OpenFOAM performance results

The following table provides the details of the operating system that was used for testing.

[Expand table](#)

Operating system	OS architecture	Processor
CentOS 7.9.2009	x86-64	AMD EPYC 7V73X (Milan-X)

The motorbike model with 21M cells was used for testing.

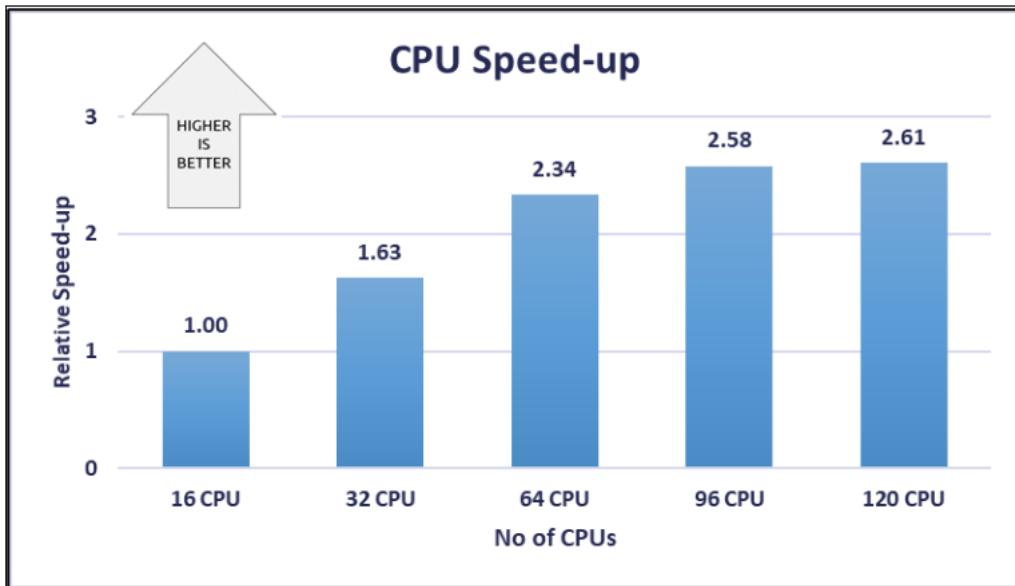


The following table shows the elapsed times, in seconds, for running the simulation with varying numbers of CPUs.

[Expand table](#)

Number of CPUs	Elapsed times (seconds)	Relative speed increase
16	412.08	1.00
32	253.04	1.63
64	176.43	2.34
96	159.62	2.58
120	157.88	2.61

This graph shows the relative speed increases as the number of CPUs increases:



Azure cost

The following table presents wall-clock times that you can use to calculate Azure costs. You can multiply the times presented here by the Azure hourly rates for HBv3-series VMs to calculate costs. For the current hourly costs, see [Linux Virtual Machines Pricing](#).

Only model running time (wall-clock time) is considered in these cost calculations. Application installation time isn't considered. The calculations are indicative. The actual costs depend on the size of the model.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

[Expand table](#)

Number of CPU cores	Wall-clock time (hours)
16	0.114
32	0.070
64	0.049
96	0.044
120	0.044

Summary

- OpenFOAM was successfully tested on HBv3-series VMs on Azure.
- An appreciable speed increase is achieved as CPU cores are increased up to 64 cores. After that point, the speed increase starts to saturate.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy OpenRadioss on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [OpenRadioss](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running OpenRadioss on Azure.

OpenRadioss is a free, open-source finite element analysis (FEA) code base that a worldwide community of researchers, software developers, and industry leaders are enhancing every day. OpenRadioss empowers users to make rapid contributions that tackle the latest challenges brought on by rapidly evolving technologies, such as:

- Battery development
- Lightweight materials and composites
- Human body models and biomaterials
- Autonomous driving and flight

OpenRadioss also provides virtual testing so that users can give passengers the safest environment possible.

- **Scalability, quality, and robustness**

Industry-proven, class leading scalability from single core to HPC (High Performance Computing), with repeatability regardless of CPU count, optimized for all modern CPU processors

- **Innovative element formulations**

Fast and accurate solutions from under integrated shell and solid elements with physical hourglass stabilization giving quality results at a fraction of fully integrated CPU cost

- **Extensive material and rupture libraries**

Advanced nonlinear material laws and failure models, for isotropic, orthotropic, composite, hyper- and viscoelastic materials

- **Comprehensive capabilities**

Wide array of contact interfaces, boundary conditions, imposed conditions, loading, joints, sensors, and output requests in animation and graph formats

- **Complex multiphysics**

Battery failure model; Airbag deployment; Smooth Particle Hydrodynamics (SPH); Arbitrary Lagrangian Eulerian (ALE); Fluid Structure Interaction (FSI); Multiphase fluids

Why deploy OpenRadioss on Azure?

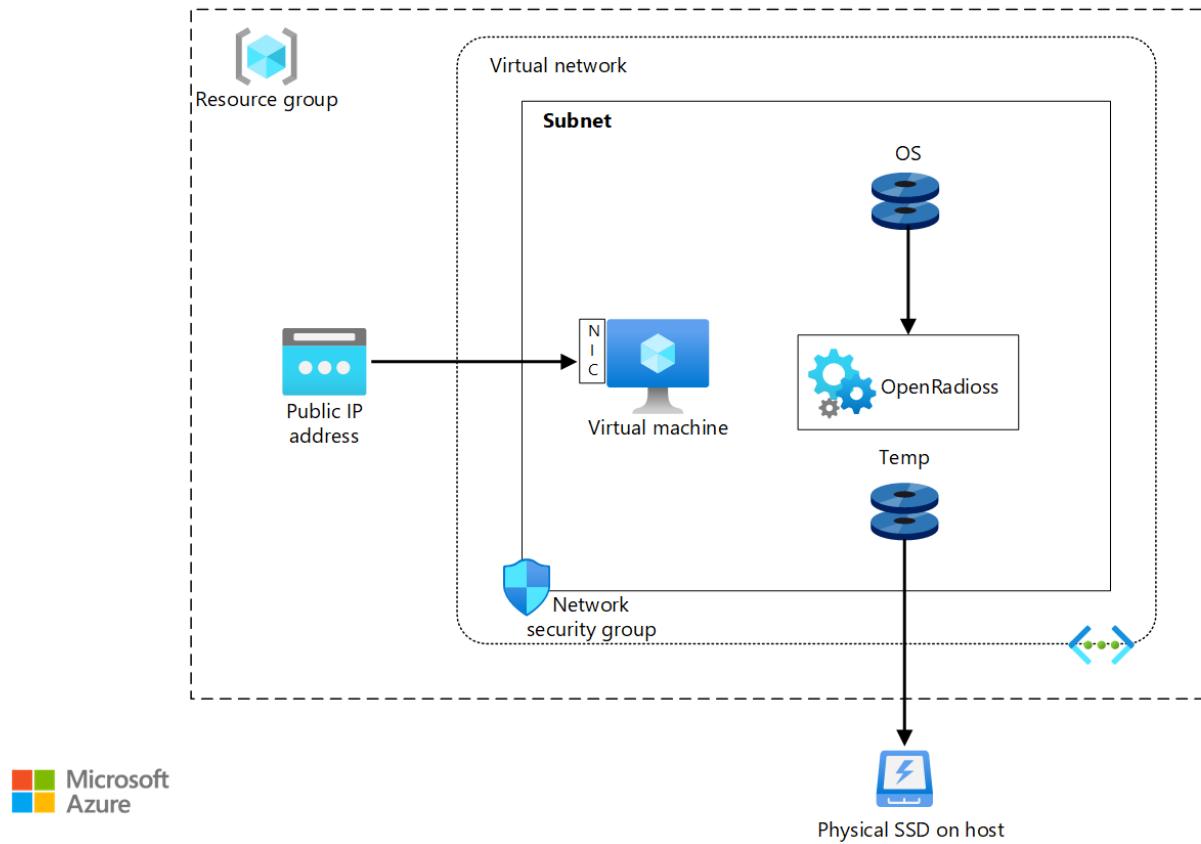
- Provides modern and diverse compute options to align with your workload's needs
- Offers the flexibility of virtualization without the need to buy and maintain physical hardware
- Provides rapid provisioning
- On a single node, improves performance as much as 2.76 times over that of 16 CPUs

Architecture

This section shows the differences between the architecture for a single-node configuration and the architecture for a multi-node configuration.

Single-node configuration:

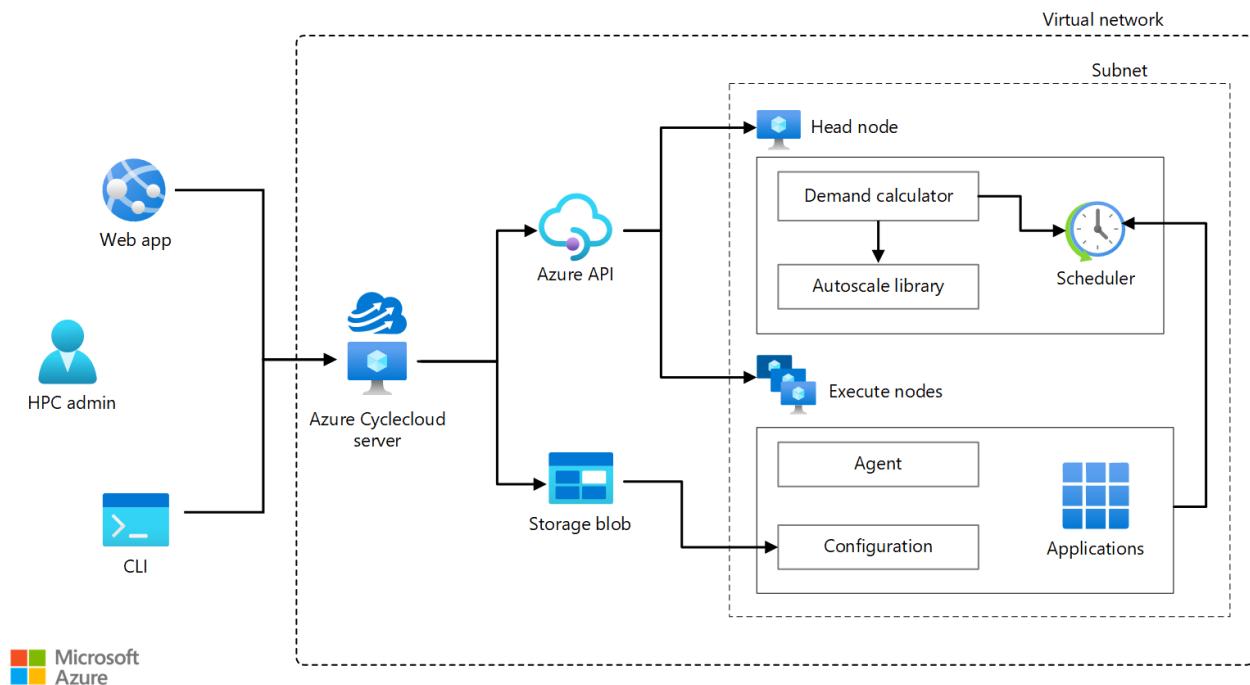
This architecture shows a single-node configuration:



[Download a Visio file](#) of this architecture.

Multi-node configuration:

This architecture shows a multi-node configuration, orchestrated with Azure CycleCloud:



[Download a Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Linux VM. For information about deploying the VM and installing the drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- [Azure CycleCloud](#) is used to create the cluster in the multi-node configuration.
- A physical SSD is used for storage.

Compute sizing and drivers

Performance tests of OpenRadioss on Azure used [HBv3-series](#) VMs running Linux. The following table provides the configuration details.

 [Expand table](#)

VM size	vCPU	RAM memory (GiB)	Memory bandwidth (Gbps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120rs_v3	120	448	350	1.9	3.0	3.5	200	32
Standard_HB120-96rs_v3	96	448	350	1.9	3.0	3.5	200	32
Standard_HB120-64rs_v3	64	448	350	1.9	3.0	3.5	200	32
Standard_HB120-32rs_v3	32	448	350	1.9	3.0	3.5	200	32
Standard_HB120-16rs_v3	16	448	350	1.9	3.0	3.5	200	32

HBv3-series VMs are optimized for HPC applications, such as:

- Fluid dynamics
- Explicit and implicit finite-element analysis
- Weather modeling
- Seismic processing
- Reservoir simulation
- RTL simulation

HBv3 VMs with different numbers of vCPUs were deployed to determine the optimal configuration for OpenRadioss test simulations on a single node. That optimal configuration was then tested in a multi-node cluster deployment.

OpenRadioss installation

Before you install OpenRadioss, deploy and connect a Linux VM and install the required AMD drivers. For information about deploying the VM, see [Run a Linux VM on Azure](#).

You can install OpenRadioss from the OpenRadioss download page. For information about the installation process, see [OpenRadioss User Documentation](#).

Multi-node configuration

You can deploy an HPC cluster on Azure by using [Azure CycleCloud](#).

Azure CycleCloud lets you orchestrate and manage HPC environments on Azure. You can use Azure CycleCloud to provision infrastructure for HPC systems, deploy HPC schedulers, and automatically scale the infrastructure to run jobs efficiently at any scale.

Azure CycleCloud is a Linux-based web application. We recommend that you set it up by deploying an Azure VM that's based on a preconfigured Azure Marketplace image.

To set up an HPC cluster on Azure, complete these steps:

1. [Install and configure Azure CycleCloud](#).
2. [Create an HPC cluster from built-in templates](#).
3. [Connect to the head node \(the scheduler\)](#).

For multi-node configurations, the OpenRadioss installation process is the same as the process described for a single node, except the path to the installation directory is different.

- Select `/shared` for the installation directory path so that the directory is accessible for all nodes.
- The shared folder path depends on your network attached storage service, like an NFS server, [BeeGFS cluster](#), [Azure NetApp Files](#), and [Azure HPC Cache](#).

OpenRadioss performance results

OpenRadioss was tested in single-node and multi-node configurations. Computation time (engine run time) was measured. The Linux platform was used, with an Azure Marketplace CentOS 8.1 HPC Gen2 image. The following table provides details.

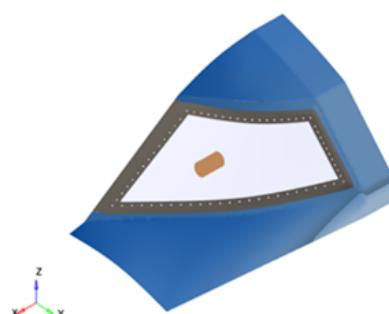
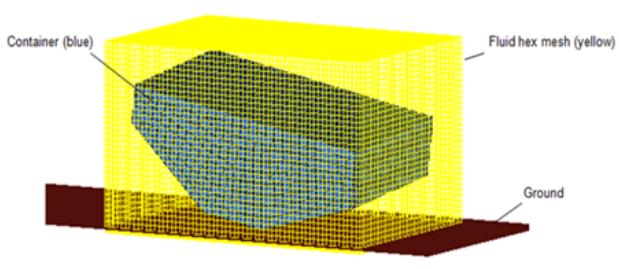
grid [Expand table](#)

Operating system version	OS architecture	MPI
CentOS Linux release 8.1.1911 (Core)	x86-64	Open MPI

Model Details

For Single-node runs:

grid [Expand table](#)

Bird Strike	INIVOL and Fluid Structure Interaction model
	

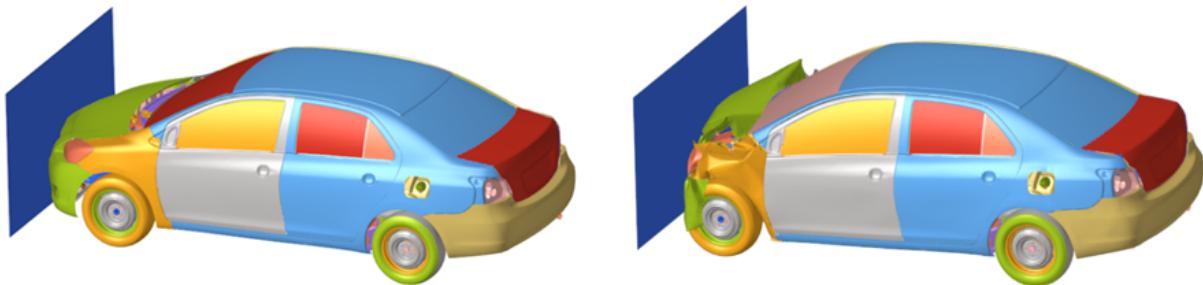
grid [Expand table](#)

Model details	Bird Strike	INIVOL and Fluid Structure Interaction model
FINAL TIME	10.00000	0.00930
TIME INTERVAL FOR TIME HISTORY PLOTS	0.10000	0.00015

Model details	Bird Strike	INIVOL and Fluid Structure Interaction model
TIME STEP SCALE FACTOR	0.90000	0.90000
MINIMUM TIME STEP	0.00000	0.00000
NUMBER OF 3D SOLID ELEMENTS	0	89670
NUMBER OF 3D SHELL ELEMENTS (4-NODES)	37020	1510
NUMBER OF 3D SHELL ELEMENTS (3-NODES)	275	44
NUMBER OF RIGID WALLS	6	1
NUMBER OF RIGID BODIES	0	1
NUMBER OF NODAL POINTS	51704	97513
NUMBER OF 3D BEAM ELEMENTS	201	0
NUMBER OF 3D SPRING ELEMENTS	144	0

For Multi-node runs:

Yaris Impact Model



[Expand table](#)

Model details	Yaris
FINAL TIME	0.20000
TIME INTERVAL FOR TIME HISTORY PLOTS	0.00010
TIME STEP SCALE FACTOR	0.90000
MINIMUM TIME STEP	0.00000
NUMBER OF 3D SOLID ELEMENTS	259803
NUMBER OF 3D SHELL ELEMENTS (4-NODES)	1189905
NUMBER OF 3D SHELL ELEMENTS (3-NODES)	65134
NUMBER OF RIGID WALLS	6

Model details	Yaris
NUMBER OF RIGID BODIES	887
NUMBER OF NODAL POINTS	1489653
NUMBER OF 3D BEAM ELEMENTS	313
NUMBER OF 3D SPRING ELEMENTS	7678

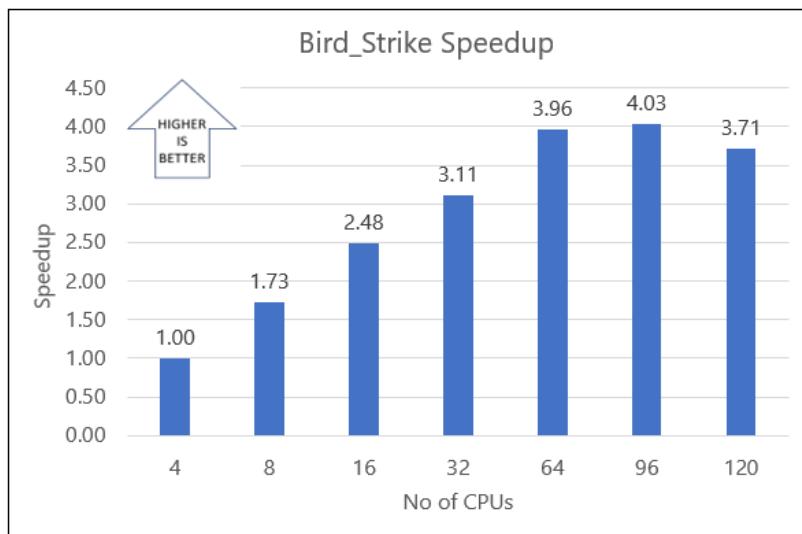
OpenRadioss Performance Results on single node

Bird Strike Model

The following table shows the elapsed wall-clock time, in seconds, for the test runs on a bird strike model of 16,600 cycles. Engine runtime was considered to calculate speedup.

[Expand table](#)

VM Size	CPU	Starter Runtime (sec)	Engine Runtime (sec)	Speedup
Standard_HB120-16rs_v3	4	0.94	551.99	1.00
Standard_HB120-16rs_v3	8	0.99	319.66	1.73
Standard_HB120-16rs_v3	16	2.17	222.45	2.48
Standard_HB120-32rs_v3	32	2.54	177.43	3.11
Standard_HB120-64rs_v3	64	2.05	139.36	3.96
Standard_HB120-96rs_v3	96	3.07	136.91	4.03
Standard_HB120rs_v3	120	4.76	148.98	3.71

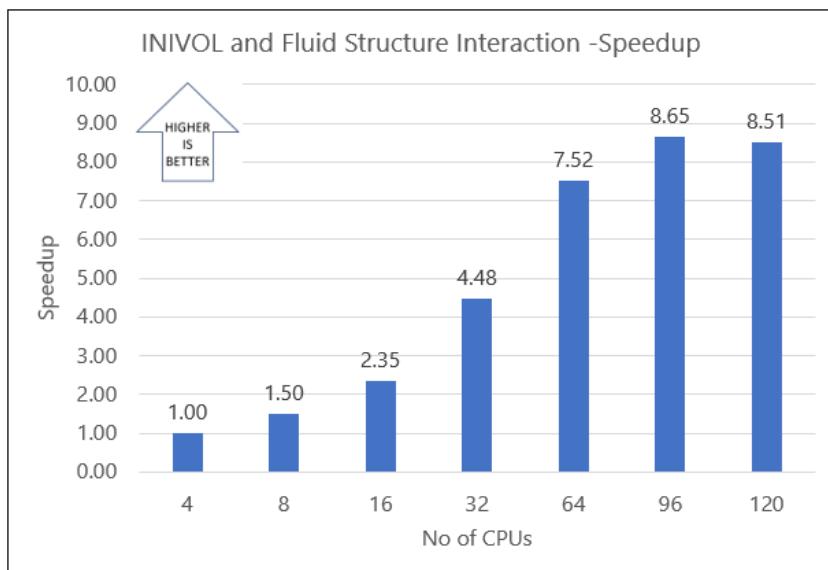


INIVOL and Fluid Structure Interaction model

The following table shows the elapsed wall-clock time, in seconds, for the test runs on an INIVOL and Fluid Structure Interaction model of 11,000 cycles. Engine runtime was considered to calculate speedup.

[Expand table](#)

VM Size	CPU	Starter Runtime (sec)	Engine Runtime (sec)	Speedup
Standard_HB120-16rs_v3	4	5.01	1112.19	1.00
Standard_HB120-16rs_v3	8	6.03	740.64	1.50
Standard_HB120-16rs_v3	16	6.12	472.55	2.35
Standard_HB120-32rs_v3	32	7.77	248.38	4.48
Standard_HB120-64rs_v3	64	9.30	147.85	7.52
Standard_HB120-96rs_v3	96	10.91	128.52	8.65
Standard_HB120rs_v3	120	11.13	130.71	8.51



OpenRadioss Performance Results on multi node

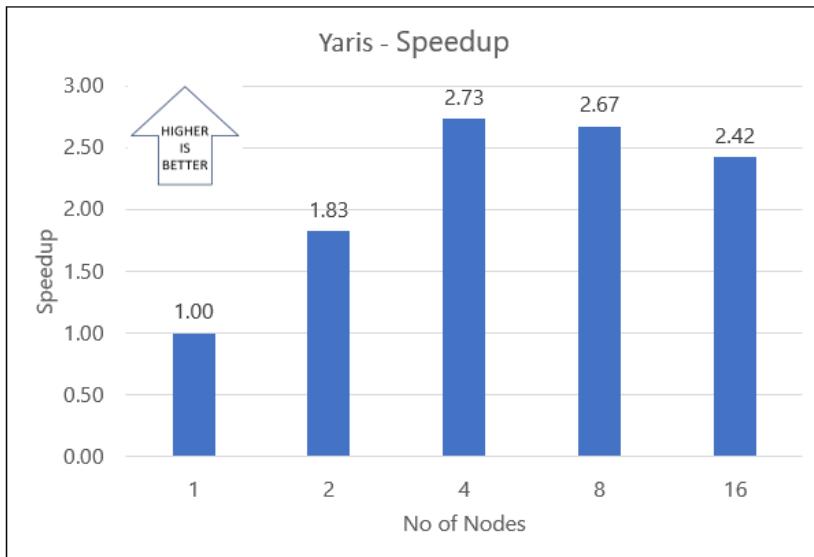
Yaris Impact Model

For Yaris Impact model, as the preceding performance results show, a [Standard_HB120-64rs_v3](#) VM (AMD EPYC™ 7V73X Processors) with 64 cores is the optimal configuration. This configuration was used in the multi-node tests. Sixty-four cores were used on each node.

The following table shows the elapsed wall-clock time, in seconds, for the test runs on a Yaris Impact model of 200,800 cycles and a VM size of [Standard_HB120-64rs_v3](#). Engine runtime was considered to calculate speedup.

[Expand table](#)

CPU	Thread	Starter Runtime (sec)	Engine Runtime (sec)	Speedup
1	64	1	58.47	8373.51
2	128	1	123.57	4578.15
4	256	1	201.13	3064.46
8	512	2	127.67	3133.47
16	1024	4	117.06	3459.74



Azure cost

Only model running time (total run time) is considered for these cost calculations. Application installation time isn't considered. The calculations are indicative. The actual numbers depend on the size of the model.

You can use the wall-clock time and the Azure hourly cost to compute total costs. For the current hourly costs, see [Linux Virtual Machines Pricing](#).

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

The following table provides the wall-clock times for single-node configurations.

Cost for running Bird Strike Model:

[Expand table](#)

VM size	Number of CPUs	Wall clock time (hours)
HB120-16rs_v3	4, 8, 16	00:18:18
HB120-32rs_v3	32	00:03:00
HB120-64rs_v3	64	00:02:21
HB120-96rs_v3	96	00:02:20
HB120-120rs_v3	120	00:02:34

Cost for running Drop Container Model:

[Expand table](#)

VM size	Number of CPUs	Wall clock time (hours)
HB120-16rs_v3	4, 8, 16	00:39:03
HB120-32rs_v3	32	00:04:16
HB120-64rs_v3	64	00:02:37

VM size	Number of CPUs	Wall clock time (hours)
HB120-96rs_v3	96	00:02:19
HB120-120rs_v3	120	00:02:22

The following table provides the wall-clock times for multi-node configurations.

 [Expand table](#)

VM size	Model	Number of CPUs	Number of nodes	Wall clock time (hours)
HB120-64rs_v3	Yaris Impact Model	64	1	02:20:32
HB120-64rs_v3	Yaris Impact Model	128	2	01:18:22
HB120-64rs_v3	Yaris Impact Model	256	4	00:54:26
HB120-64rs_v3	Yaris Impact Model	512	8	00:54:21
HB120-64rs_v3	Yaris Impact Model	1024	16	00:59:37

Summary

- OpenRadioss was successfully tested on HBv3 AMD EPYC™ 7V73X series on Azure VM and Azure CycleCloud multi-node setup.
- OpenRadioss on an Azure VM can solve complex workloads.
- OpenRadioss scales up to four nodes (256 CPUs) for a Yaris Impact model. Models that are larger than a Yaris Impact model (1.4 million nodal points) scale up better when targeting the number of nodes.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vivi Richard](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)

- HPC system and big-compute solutions
- HPC cluster deployed in the cloud

Deploy Quantum ESPRESSO on an Azure virtual machine

Azure Virtual Machines Azure Virtual Network

This article describes the steps for running the [Quantum ESPRESSO](#) application on a virtual machine (VM) deployed on Azure. It also presents the performance results of running Quantum ESPRESSO on single-node and multi-node VM configurations.

Quantum ESPRESSO is based on density-functional theory, plane waves, and pseudopotentials. Quantum ESPRESSO is an integrated suite of open-source computer codes for electronic-structure calculations and materials modeling at the nanoscale. The Quantum ESPRESSO distribution consists of a historical core set of components, and a set of plug-ins that perform more advanced tasks, plus many third-party packages designed to be inter-operable with the core components.

Quantum ESPRESSO runs on many different architectures and conditions:

- Ground state calculations utilizing local density approximation (LDA), Generalized Gradient Approximation (GGA), GGA+U, van der Waals (vdW-DF) and Hybrid Exchange-Correlation Functionals
- Support of Ultrasoft (US), Norm-Conserving (NC) pseudopotentials and Projector Augmented Wave (PAW) method
- Structural optimization and polymorphism studies
- Transition states and minimum energy paths using Nudged Elastic Bands (NEB) method
- Linear response properties within Density Functional Perturbation theory (DFPT)
- Spectroscopic properties
- Effective Screening Medium (ESM) for charged surfaces and interfaces

Quantum ESPRESSO is used mainly by researchers active in the field of electronic-structure calculations.

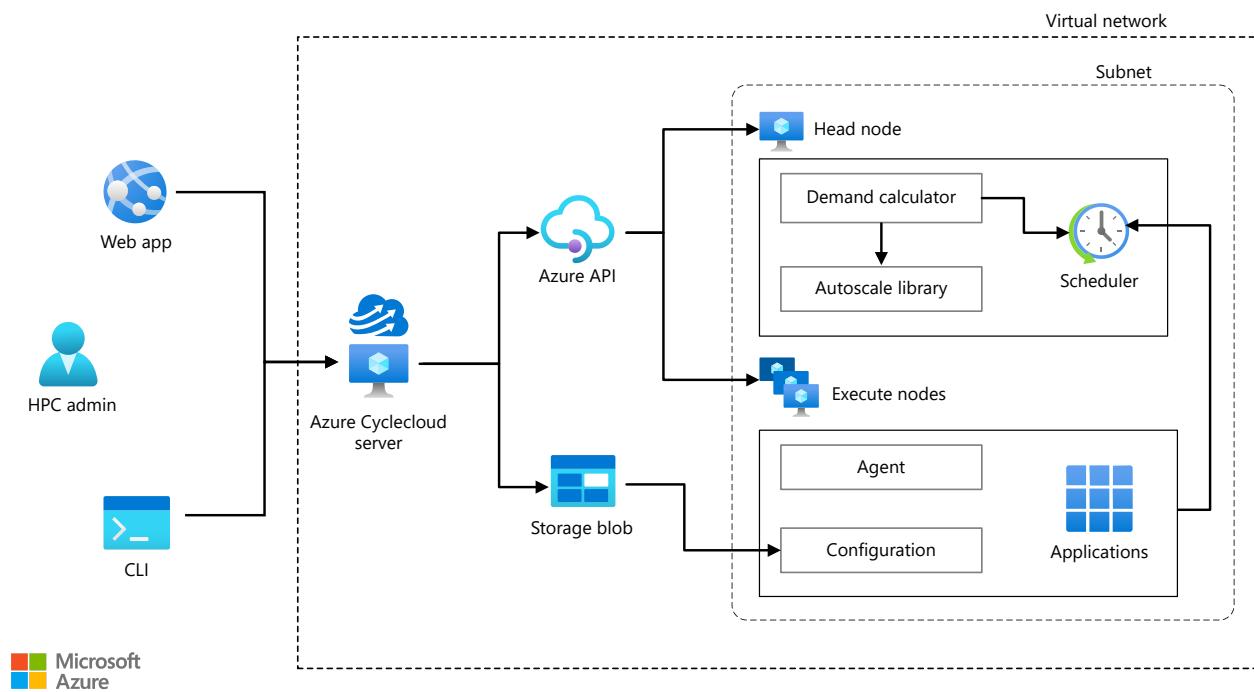
Why deploy Quantum ESPRESSO on Azure?

- Modern and diverse compute options to meet your workload's needs.
- The flexibility of virtualization without the need to buy and maintain physical hardware.
- Rapid provisioning.
- Complex problems solved within a few hours.

Architecture

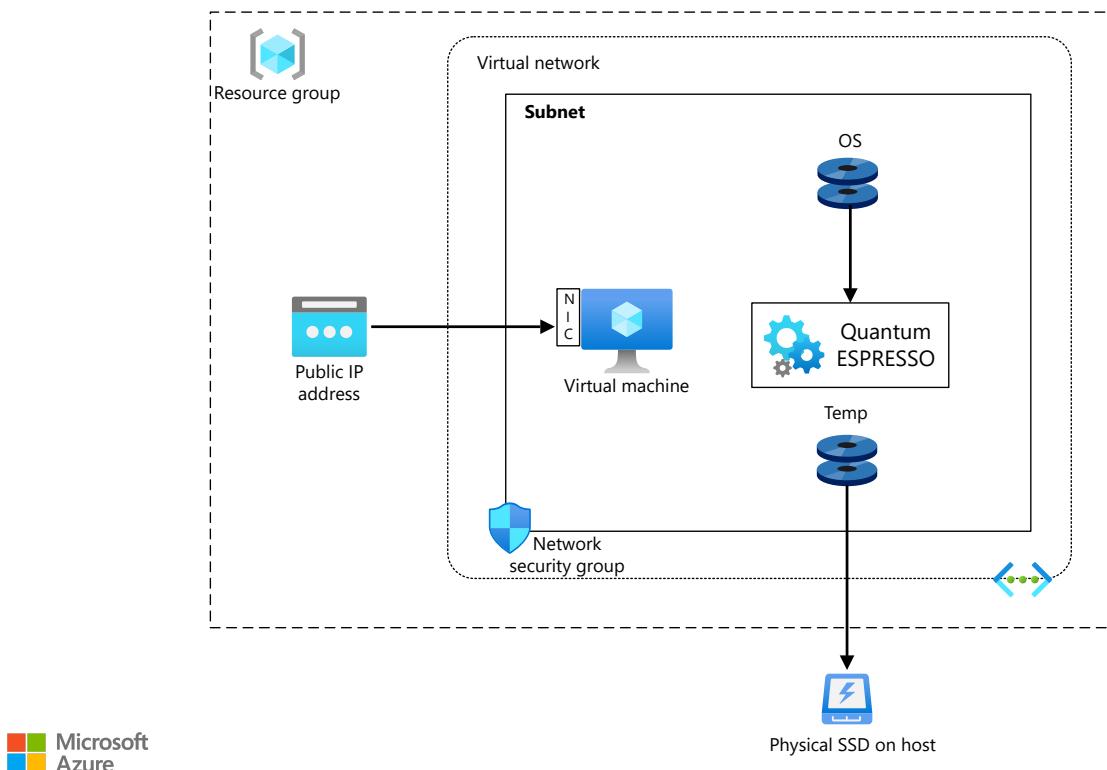
This section shows the differences between the architecture for a multi-node configuration and the architecture for a single-node configuration.

Multi-node configuration:



Download a [Visio file](#) of this architecture.

Single-node configuration:



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create Linux and Windows VMs. For information about deploying the VM and installing the drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VMs.
 - A public IP address connects the internet to the VMs.
- [Azure CycleCloud](#) is used to create the cluster in the multi-node configuration.

- A physical SSD is used for storage.

Compute sizing and drivers

Performance tests of Quantum ESPRESSO on Azure used [HBv3 AMD EPYC™ 7V73X](#) VMs running Linux CentOS Operating system. The following table provides details about HBv3-series VMs.

[Expand table](#)

VM size	vCPU	Memory (GiB)	Memory bandwidth (GBps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (GBps)	Maximum data disks
Standard_HB120rs_v3	120	448	350	1.9	3.0	3.5	200	32
Standard_HB120-96rs_v3	96	448	350	1.9	3.0	3.5	200	32
Standard_HB120-64rs_v3	64	448	350	1.9	3.0	3.5	200	32
Standard_HB120-32rs_v3	32	448	350	1.9	3.0	3.5	200	32
Standard_HB120-16rs_v3	16	448	350	1.9	3.0	3.5	200	32

Required drivers

To use InfiniBand, you must enable [InfiniBand](#) drivers.

Install Quantum ESPRESSO on a VM or HPC Cluster

Download the software from the [Quantum ESPRESSO](#) official website.

Before you install Quantum ESPRESSO, deploy and connect to a VM or HPC Cluster. For information about deploying the VM and installing the drivers, see one of these articles:

- [Run a Windows VM on Azure](#)
- [Run a Linux VM on Azure](#)

For information about deploying the Azure CycleCloud and HPC cluster, see one of these articles:

- [Install and configure Azure CycleCloud](#)
- [Create a HPC Cluster](#)

Install Quantum ESPRESSO

To install Quantum ESPRESSO on Azure Virtual Machines and run the simulations, see the following links:

- [Quantum ESPRESSO documentation](#)
- [User's Guide for Quantum ESPRESSO](#)

Quantum ESPRESSO performance results

Quantum ESPRESSO was tested in single-node and multi-node configurations. Model *ta2o5* was used to test the scalability performance of Quantum ESPRESSO version 7.1 on Azure. The details of the model used for validation are as follows:

Model Details	Number of atoms per cell	Number of atomic types	Number of electrons	Mixing beta
ta2o5	25	2	125	0.5000

Quantum ESPRESSO performance results on single-node VMs

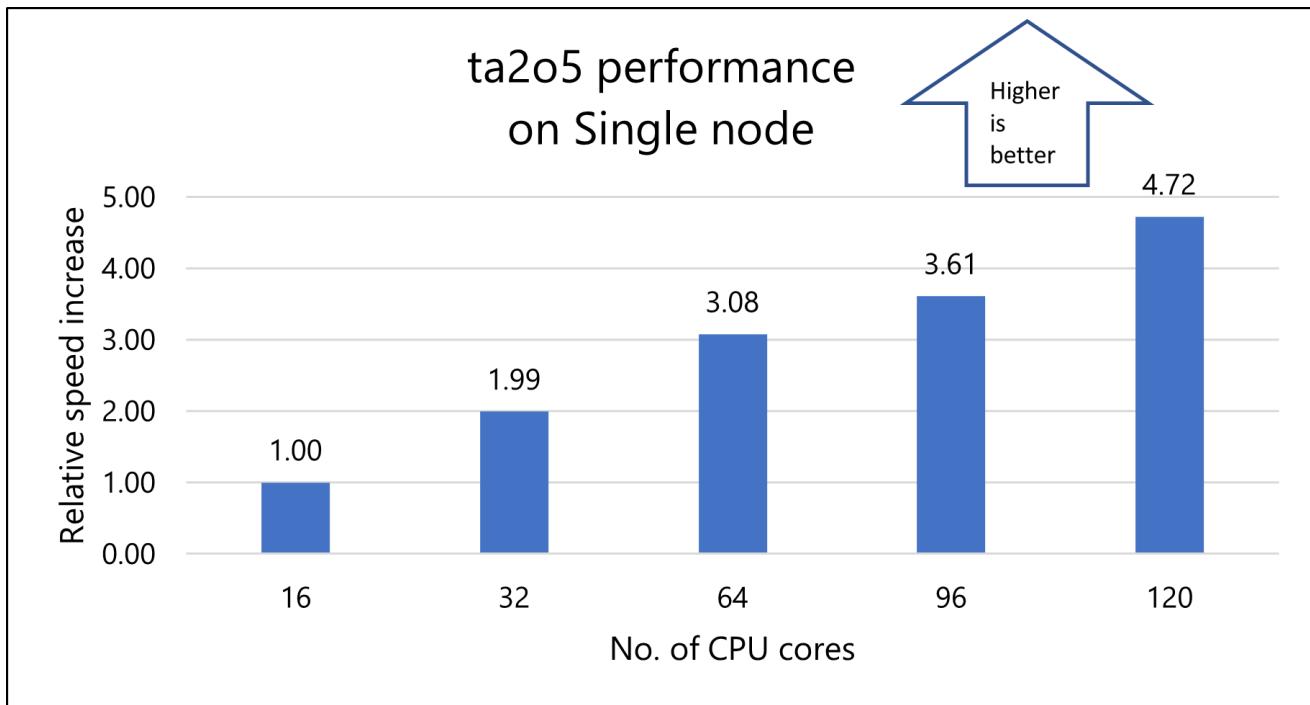
This section provides the performance results of running Quantum ESPRESSO on single-node Azure HBv3 AMD EPYC™ 7V73X VMs.

Model 1: ta2o5

This table shows total wall clock time recorded for a varying number of CPUs on the standard HBv3-series VM:

Number of cores	Wall clock time (seconds)	Relative speed up
16	7213	1.00
32	3620	1.99
64	2344	3.08
96	1998	3.61
120	1527	4.72

This graph shows the relative speedup improvement as the number of CPUs increases:



Notes about the single-node tests

For all single-node tests, a solver time of HB120-16rs_v3 (16 cores) is considered as the reference to compute the relative speed of other VMs with more cores. The speedup improves as we increase the number of cores from 16 to 120 cores. We can observe that a speedup of about 4.7x is achieved with 120 cores with the ta2o5 model.

Quantum ESPRESSO performance results on a multi-node cluster

The single-node tests confirm that the solver gives optimal parallel performance with 64 cores on HBv3 VMs. Based on those results, 64-core configurations on Standard_HB120-64rs_v3 VMs were used to evaluate the performance of Quantum ESPRESSO on multi-node clusters. The ta2o5 model is used for multi-node runs.

This section provides the performance results of running Quantum ESPRESSO on multi-node VMs.

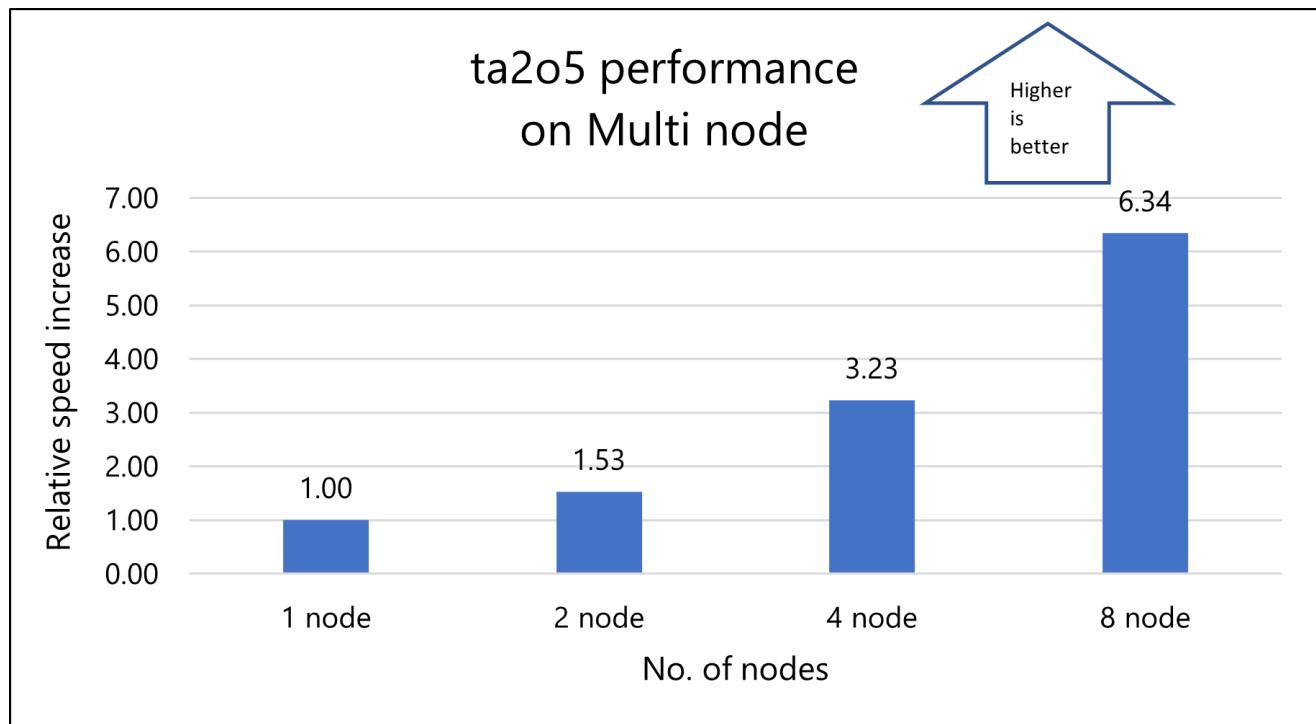
Model 1: ta2o5 for multi-node

This table shows total wall clock time recorded for varying numbers of nodes with standard HBv3-series VMs:

 Expand table

Number of nodes	Number of cores	Wall clock time (seconds)	Relative speed up
1	64	2270	1.00
2	128	1487	1.53
4	256	703	3.23
8	512	358	6.34

This graph shows the relative speed increase as the number of nodes increases:



Azure cost

Only model running time (wall clock time) is considered for these cost calculations. Application installation time isn't considered. The numbers presented here are indicative of your potential results. The actual numbers depend on the size of the model.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

The following table provides the elapsed times in hours. To compute the total cost, multiply these times by the hourly costs for [Linux VMs](#).

Cost for model 1: ta2o5:

Number of nodes	Wall clock time (hours)
1	0.63
2	0.41
4	0.20
8	0.10

Summary

- Quantum ESPRESSO was successfully tested on Azure using HBv3 standalone Virtual Machines and an Azure Cycle Cloud multi-node (cluster) setup.
- For a standalone VM, we can observe that the application is scaling up with an increase of vCPUs and achieved a speedup of 4.7x with 120 vCPUs.
- For multi-node runs, the application scales linearly with the increase of nodes. A scale up of about 6.5x is achieved for Quantum Espresso with eight nodes.
- For small problems, we recommend you use fewer CPUs to improve performance.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer
- Shivakumar Talloli | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Windows virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Remcom XFDTD on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [Remcom XFDTD](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running Remcom XFDTD on Azure.

XFDTD is electromagnetic simulation software that includes full-wave, static, biothermal, optimization, and circuit solvers.

XFDTD includes a schematic editor that's designed for antenna engineers who need to analyze matching networks and corporate feed networks.

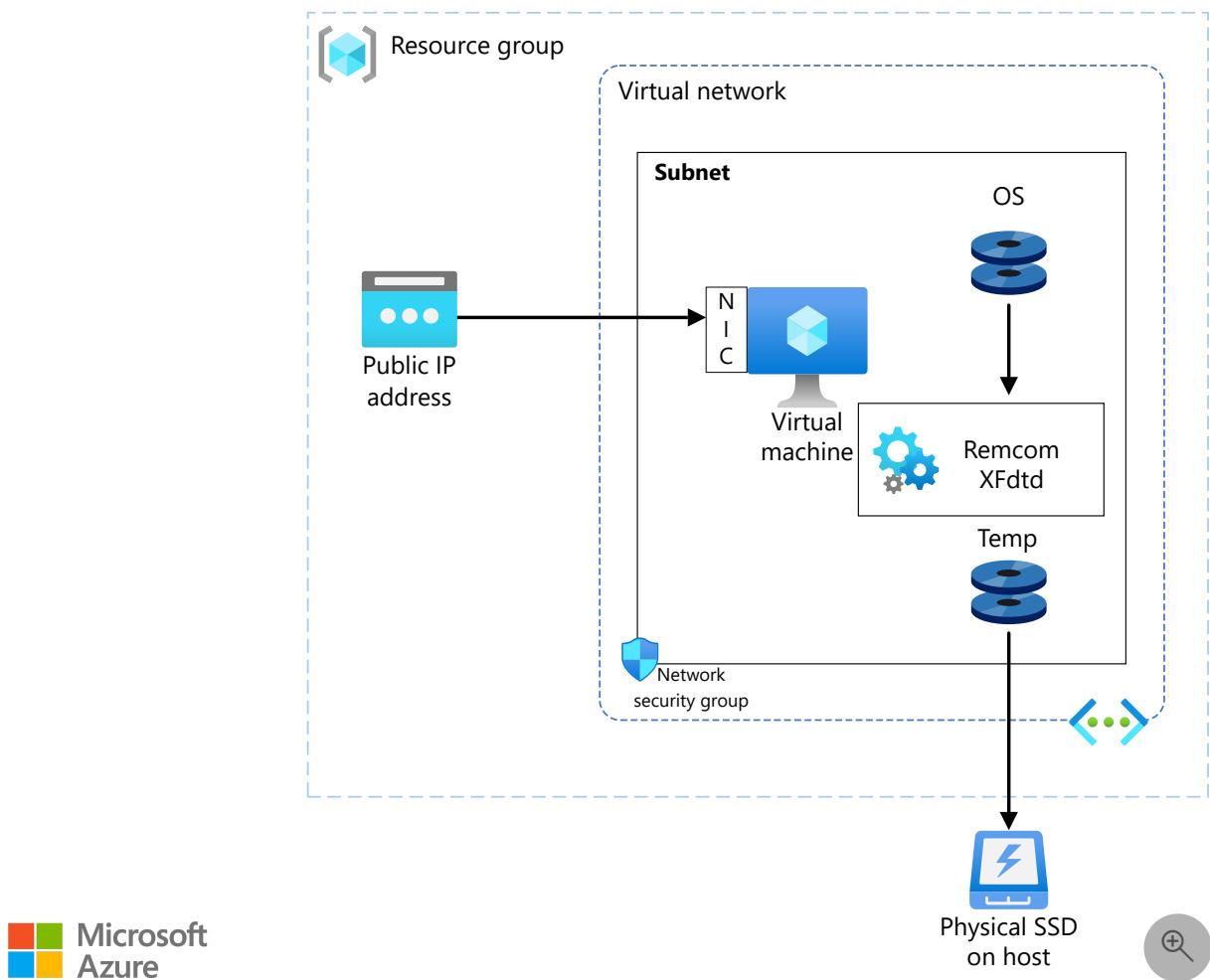
The software enables in-depth analysis of a device's stand-alone performance, with 5G-device design features that support high-frequency array antennas and complex devices that operate at millimeter wave frequencies.

XFDTD is used for antenna design and analysis, antenna modeling, 5G array analysis, biomedical effects, microwave devices and waveguides, radar/scattering, military defense, and automotive radar. It's ideal for the telecommunications, healthcare, manufacturing, and automotive industries.

Why deploy Remcom XFDTD on Azure?

- Modern and diverse compute options to meet your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Fast compute capabilities for GPU-intensive workloads

Architecture



Download a [Visio file](#) of this architecture.

Components

- Azure Virtual Machines is used to create a Windows VM. For information about deploying the VM and installing the drivers, see [Windows VMs on Azure](#).
- Azure Virtual Network is used to create a private network infrastructure in the cloud.
 - Network security groups are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

The performance tests of Remcom XFDTD used an [ND_A100_v4](#) VM running Windows 10. The following table provides details about the VM.

[Expand table](#)

VM size	vCPU	Memory, in GiB	Temporary storage (SSD), in GiB	GPUs	GPU memory, in GiB	Maximum data disks
Standard_ND96asr_v4	96	900	6,000	8 A100	40	32

Required drivers

To take advantage of the GPU capabilities of [ND_A100_v4](#) VMs, you need to install NVIDIA GPU drivers.

To use AMD processors on [ND_A100_v4](#) VMs, you need to install AMD drivers.

Remcom XFDTD installation

Before you install Remcom XFDTD, you need to deploy and connect a VM, install an eligible Windows 10 image, and install the required NVIDIA and AMD drivers.

For information about eligible Windows images, see [How to deploy Windows 10 on Azure](#) and [Use Windows client in Azure for dev/test scenarios](#).

i **Important**

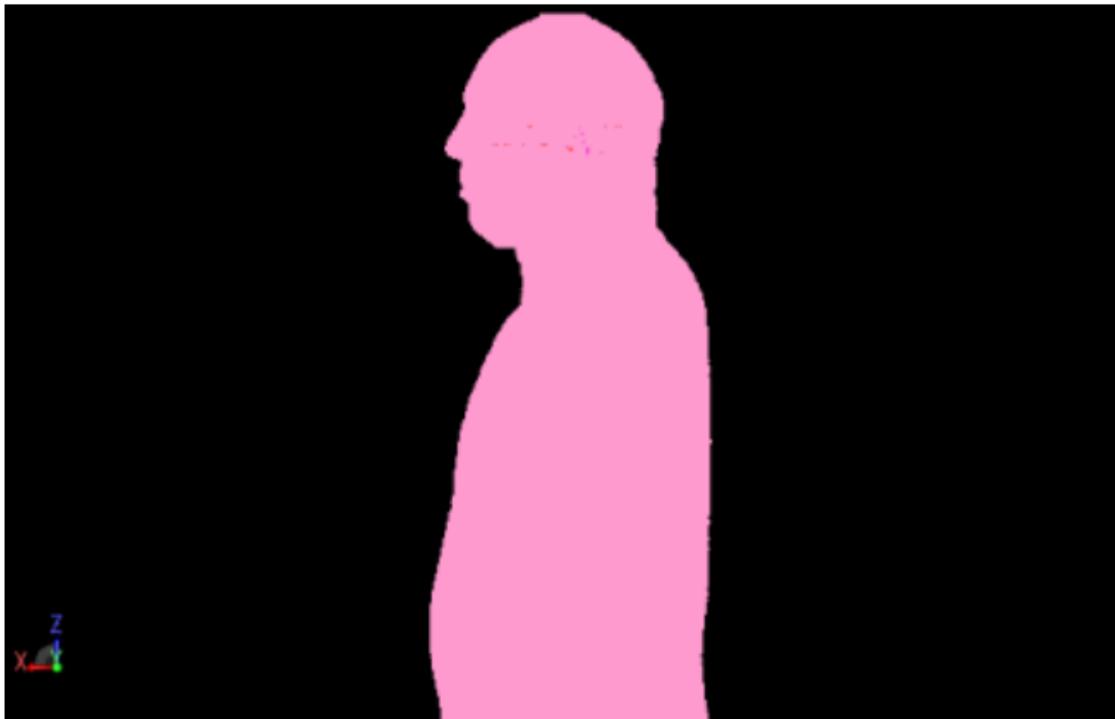
NVIDIA Fabric Manager installation is required for VMs that use NVLink or NVSwitch. ND_A100_v4 VMs use NVLink.

For information about deploying the VM and installing the drivers, see [Run a Windows VM on Azure](#).

Before you install XFDTD, you need to install a floating license server. For detailed instructions on [installing XFDTD](#), [installing the license server](#), [setting up an MPI cluster](#), and more, see the [installation manual](#).

Remcom XFDTD performance results

Remcom XFDTD version 7.10.0.1 was tested. A patch in body model was used for the tests:



The following table provides details about the model.

[Expand table](#)

Model name	Frequency range of interest	Minimum cells per wavelength	Frequency of interest	Minimum cell size
Patch in body	0 GHz to 10 GHz	15	2.45 GHz	0.735 mm

Throughput is used as a metric to test the performance of the simulation. The following table provides the test results.

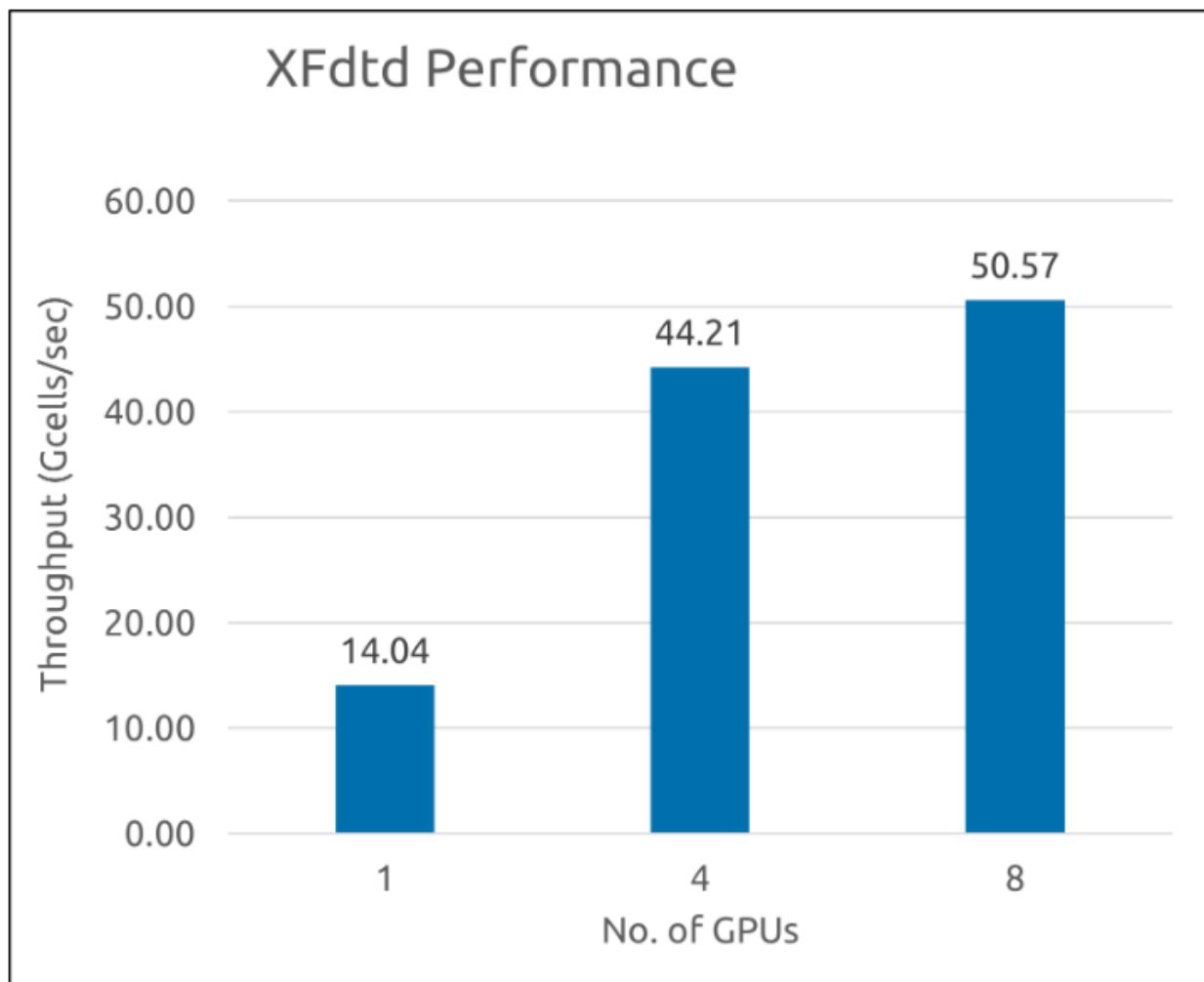
[Expand table](#)

Number of GPUs	Simulation timesteps	Total cells	Elapsed time (seconds)	Timesteps per second	Throughput (cells per second)	Throughput (Gcells ¹ per second)
1 ²	39,141	1,077,552,576	3,003	13.03	14,044,783,675	14.04
4 ²	39,141	1,077,552,576	954	41.03	44,210,152,387	44.21
8	39,141	1,077,552,576	834	46.93	50,571,325,392	50.57

¹ Gcells is the cell count divided by 1,000,000,000.

² In these cases, the number of GPUs was artificially limited. The Standard_ND96asr_v4 VM has eight GPUs.

This graph shows the throughput for the different numbers of GPUs:



Azure cost

The following table shows the wall-clock time for running the simulation, in hours. To compute the total cost, multiply this time by the Azure VM hourly cost for the NDA100v4 VM. For the current hourly cost, see [Windows Virtual Machines Pricing](#).

Only simulation runtime is included in the reported time. Application installation time isn't included.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

Expand table

VM size	Number of GPUs	Elapsed time (hours)
Standard_ND96asr_v4	8	0.23

Summary

- Remcom XFdtd was successfully tested on the Standard_ND96asr_v4 VM.
- A complex simulation ran in 0.23 hours.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- GPU-optimized virtual machine sizes
- Virtual machines on Azure
- Virtual networks and virtual machines on Azure
- Learning path: Run high-performance computing (HPC) applications on Azure

Related resources

- [Run a Windows VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Samadii DEM on a virtual machine

Azure Virtual Machines Azure Virtual Network

This article briefly describes the steps for running [Samadii DEM](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running Samadii DEM on Azure.

Samadii DEM analyzes and interprets large-scale particles at high speed. It uses discrete element method (DEM), which is a Lagrangian method that determines the movement of particles by using the six-degrees-of-freedom equations of motion, taking into consideration all forces of individual particles. It uses explicit methods for time integration to calculate the position and velocity of the particles in the next time step.

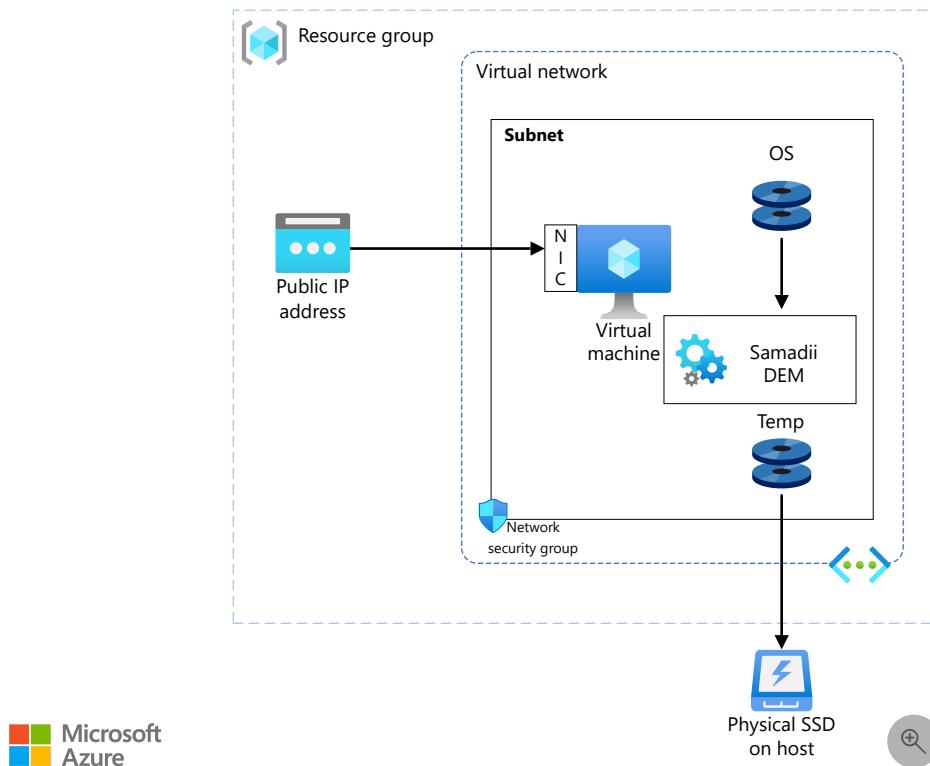
DEM requires significant memory and computing power because of its small time step and the large number of particles that it takes into account. Samadii DEM, which is designed to perform analysis by using GPU and parallel processing techniques, supplies reliable results by analyzing a variety of large-scale grain boundary issues at a high speed.

Samadii DEM is used in the mechanical, electronic, chemical, semiconductor, manufacturing, automotive, energy, and construction/facilities industries.

Why deploy Samadii DEM on Azure?

- Modern and diverse compute options to meet your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Good scale and cost efficiency on NCast4_v3-series VMs

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Windows VM. For information about deploying VMs and installing the drivers, see [Windows VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - Network security groups restrict access to the VM.

- A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) provides storage.

Compute sizing and drivers

Performance tests of Samadii DEM on Azure used [NVv3](#), [NCasT4_v3](#), [NCv3](#), [ND_A100_v4](#), and [NC A100 v4](#) series VMs running Windows 10. The following table provides details about the VMs.

[Expand table](#)

VM size	GPU name	vCPUs	Memory, in GiB	Maximum data disks	GPUs	GPU memory, in GiB	Maximum uncached disk throughput, in IOPS / MBps	Temporary storage (SSD), in GiB	Max NICs
Standard_NV12s_v3	Tesla M60	12	112	12	1	8	20,000 / 200	320	4
Standard_NC4as_T4_v3	Tesla T4	4	28	8	1	16	-	180	2
Standard_NC6s_v3	V100	6	112	12	1	16	20,000 / 200	736	4
Standard_ND96asr_v4	A100	96	900	32	8	40	80,000 / 800	6,000	8
Standard_NC24ads_A100_v4	A100	24	220	32	1	80	30,000 / 1,000	1,123	2

Required drivers

To take advantage of the GPU capabilities of [NVv3](#), [NCas_T4_v3](#), [NCv3](#), [ND_A100_v4](#), and [NC A100 v4](#) series VMs, you need to install NVIDIA GPU drivers.

Samadii DEM installation

Before you install Samadii DEM, you need to deploy and connect a VM and install the required NVIDIA drivers.

For information about deploying the VM and installing the drivers, see [Run a Windows VM on Azure](#).

Important

NVIDIA Fabric Manager installation is required for VMs that use NVLink. ND_A100_v4 VMs use this technology.

Following are some prerequisites for running Samadii applications.

- Windows 10 (x64) OS
- One or more NVIDIA CUDA-enabled GPUs: Tesla, Quadro, or GeForce series
- Visual C++ 2010 SP1 Redistributable Package
- Microsoft MPI v7.1
- .NET Framework 4.5

The product installation process involves installing a license server, installing Samadii DEM, and configuring the license server. For more information about installing Samadii DEM, contact [Metariver Technology](#).

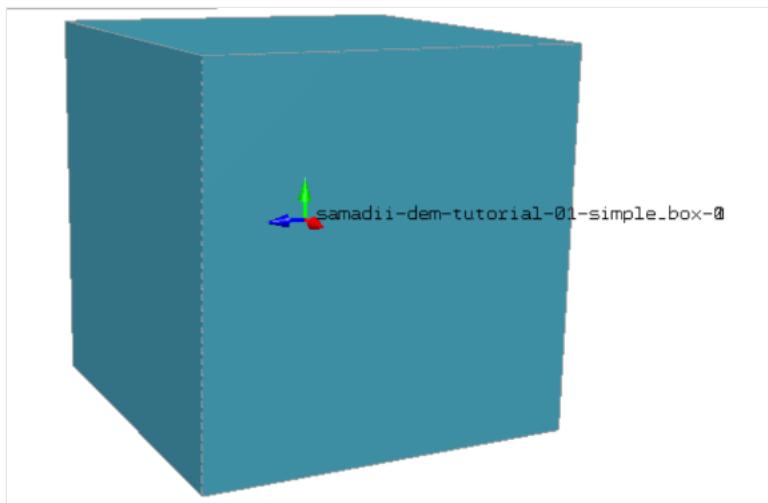
Samadii DEM performance results

The following table shows the operating system versions and processors that were used for the tests.

VM series	ND_A100_v4	NCv3	NCasT4_v3	NVv3	NC A100 v4
Operating system version	Windows 10 Professional, version 20H2	Windows 10 Professional, version 20H2	Windows 10 Professional, version 20H2	Windows 10 Professional, version 20H2	Windows 10 Professional, version 21H2
OS architecture	x86-64	x86-64	x86-64	x86-64	x86-64
Processor	AMD EPYC 7V12, 64-core processor, 2.44 GHz (2 processors)	Intel Xeon CPU E5-2690 v4	AMD EPYC 7V12, 64-core processor, 2.44 GHz	Intel Xeon CPU E5-2690 v4	AMD EPYC 7V13, 64-core processor, 2.44 GHz

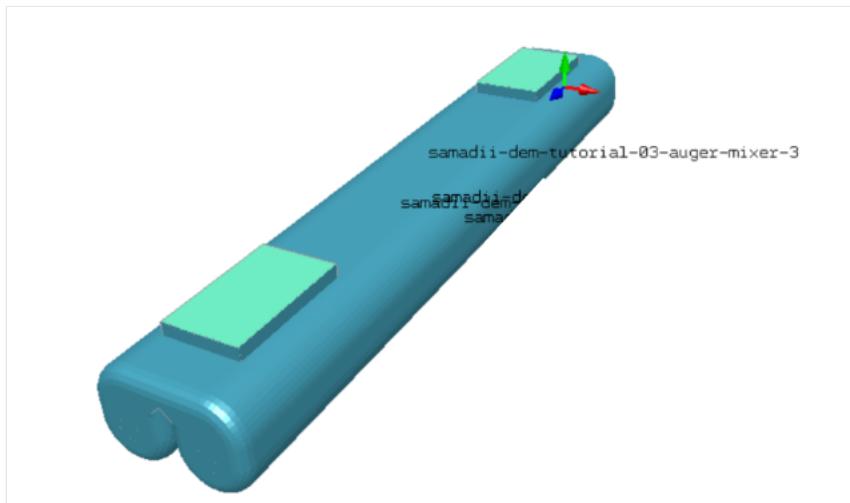
The following two models were used for testing.

Simple box



- Size: 13,560
- Cell type: Shell
- Solver: Samadii-dem-x64-v21 R2

Auger mixer



- Size: 42,446
- Cell type: Shell
- Solver: Samadii-dem-x64-v21 R2

Results for the simple box model

The following table shows the elapsed runtimes and relative speed increases for various VM configurations.

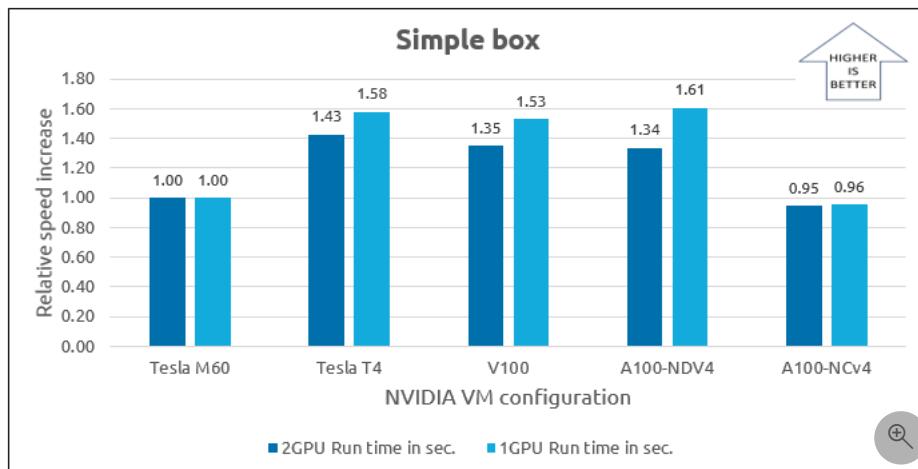
[Expand table](#)

VM series	GPU	1-GPU runtime, in seconds	Relative speed increase	2-GPU runtime, in seconds	Relative speed increase
ND_A100_v4	A100	173 ¹	1.61	252 ¹	1.34
NCv3	V100	182	1.53	249	1.35
NCasT4_v3	Tesla T4	176	1.58	236 ²	1.43
NVv3	Tesla M60	278	1.00	337	1.00
NC A100 v4	A100	290	0.96	355	0.95

¹ In these cases, the number of GPUs was artificially limited. This VM has eight GPUs.

² In these cases, the number of GPUs was artificially limited. This VM is available with one or four GPUs.

Here are the relative speed increases in graphical form:



Results for the auger mixer model

The following table shows the elapsed runtimes and relative speed increases for various VM configurations.

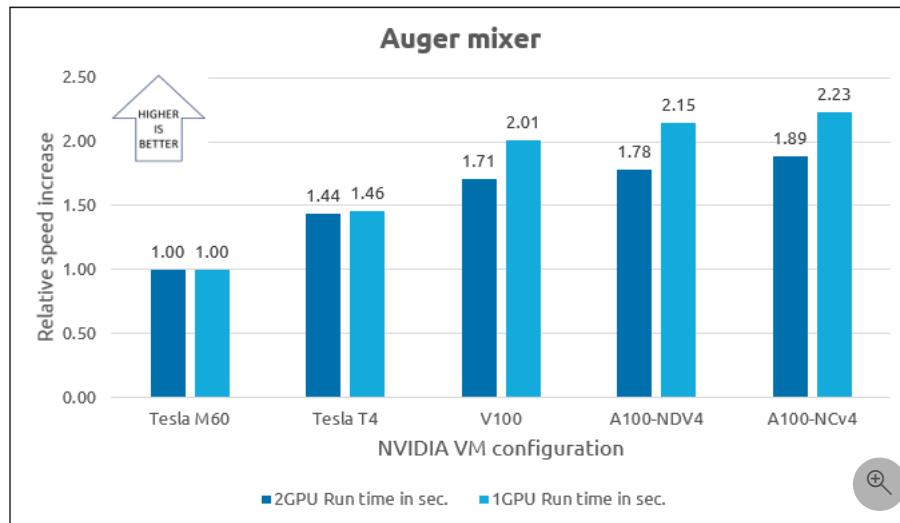
[Expand table](#)

VM series	GPU	1-GPU runtime, in seconds	Relative speed increase	2-GPU runtime, in seconds	Relative speed increase
ND_A100_v4	A100	1,766 ¹	2.15	2,054 ¹	1.78
NCv3	V100	1,885	2.01	2,140	1.71
NCasT4_v3	Tesla T4	2,601	1.46	2,543 ²	1.44
NVv3	Tesla M60	3,794	1	3,659	1
NC A100 v4	A100	1,701	2.23	1,939	1.89

¹ In these cases, the number of GPUs was artificially limited. This VM has eight GPUs.

² In these cases, the number of GPUs was artificially limited. This VM is available with one or four GPUs.

Here are the relative speed increases in graphical form:



Azure cost

The following tables present wall-clock times in hours. To compute the total cost, multiply these times by the Azure VM hourly costs for NVv3, NCsT4_v3, NCv3, ND_A100_v4, and NC_A100 v4 series VMs. For the current hourly costs, see [Windows Virtual Machines Pricing](#).

Only simulation runtime is considered for these cost calculations. Application installation time and license costs aren't included. In some cases, the number of GPUs was artificially limited for the sake of testing.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

Costs, simple box

grid icon [Expand table](#)

VM size	Number of GPUs	GPUs utilized	Wall-clock time, in hours
Standard_ND96asr_v4	8	1	0.048
Standard_ND96asr_v4	8	2	0.07
Standard_NC6s_v3	1	1	0.05
Standard_NC12s_v3	2	2	0.07
Standard_NC4as_T4_v3	1	1	0.049
Standard_NC64as_T4_v3	4	2	0.066
Standard_NV12s_v3	1	1	0.077
Standard_NV24s_v3	2	2	0.094
Standard_NC24ads_A100_v4	2	1	0.081
Standard_NC24ads_A100_v4	2	2	0.099

Costs, auger mixer

VM size	Number of GPUs	GPUs utilized	Wall-clock time, in hours
Standard_ND96asr_v4	8	1	0.49
Standard_ND96asr_v4	8	2	0.57
Standard_NC6s_v3	1	1	0.53
Standard_NC12s_v3	2	2	0.59
Standard_NC4as_T4_v3	1	1	0.72
Standard_NC64as_T4_v3	4	2	0.71
Standard_NV12s_v3	1	1	1.05
Standard_NV24s_v3	2	2	1.02
Standard_NC24ads_A100_v4	2	1	0.47
Standard_NC24ads_A100_v4	2	2	0.54

Summary

- Samadii DEM was successfully tested on ND_A100_v4, NCv3, NCasT4_v3, NVv3, and NC A100 v4 series VMs.
- Performance tests demonstrate that Samadii DEM performs well on NCasT4_v3, NCv3, NC A100 v4, and ND_A100_v4 series VMs. The speed increases are as much as 1.5 times, 2 times, 2.25 times, and 2.15 times, respectively, higher than the times recorded with Tesla M60 GPU cards.
- We recommend NCasT4_v3 VMs with a one-GPU configuration because these VMs provide good scale-up and are also cost efficient. For the simple box model, they provide relative speed increases that are comparable to those recorded on the NC A100 v4 VM.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director of Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run HPC applications on Azure](#)

Related resources

- Run a Windows VM on Azure
- HPC system and big-compute solutions
- HPC cluster deployed in the cloud

Deploy Samadii EM on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [Samadii EM](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running Samadii EM on Azure.

Samadii EM (electromagnetic) analyzes the electromagnetic field in three-dimensional space by using the Maxwell equation. It calculates the Maxwell equation by using the vector finite element method (FEM) and GPU computing.

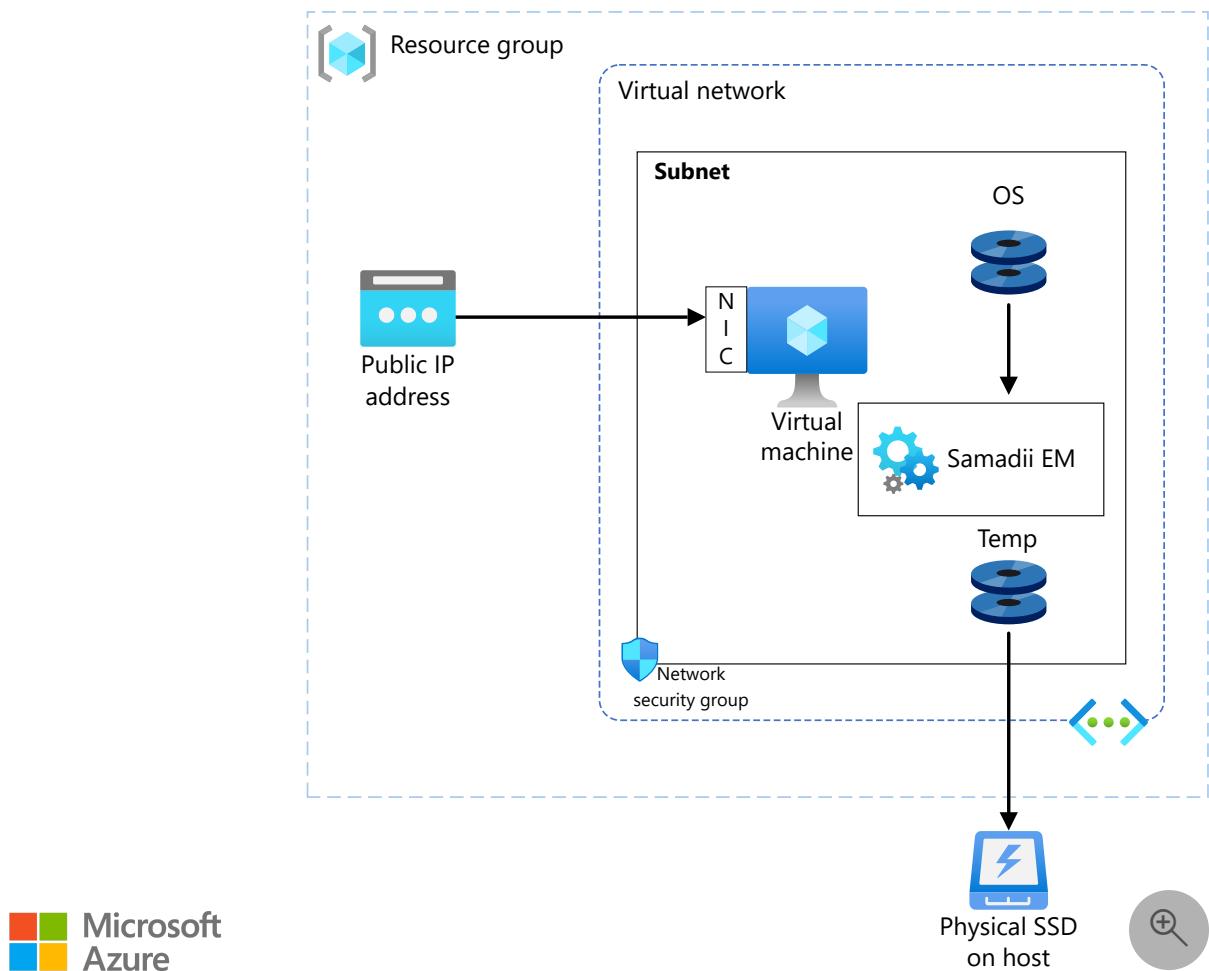
The application provides a multi-physics solution to complex electromagnetic problems. It can analyze problems in electrostatic fields, AC electromagnetic fields, and electromagnetic wave fields.

Samadii EM is used in wireless communications and by manufacturers of radar devices, motors, semiconductors, and display devices. It's ideal for the telecommunications, manufacturing, and automotive industries.

Why deploy Samadii EM on Azure?

- Modern and diverse compute options to meet your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- VM options that enable you to optimize for varying levels of simulation complexity

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Windows VM. For information about deploying the VM and installing the drivers, see [Windows VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

The performance tests of Samadii EM on Azure used [NVadsA10_v5](#), [NCas_T4_v3](#), [NCv3](#), and [NC_A100_v4](#) VMs running Windows 10.

The following table shows information about the operating systems that were used for testing:

	NCv3	NCasT4_v3	NVadsA10_v5	NC_A100_v4
Operating system version	Windows 10 Professional, version 20H2	Windows 10 Professional, version 20H2	Windows 10, version 20H2	Windows 10, version 21H2
OS architecture	x86-64	x86-64	x86-64	x86-64
Processor	Intel Xeon CPU E5-2690 v4	AMD EPYC 7V12, 64-core processor, 2.44 GHz	AMD EPYC 74F3V (Milan)	AMD EPYC 7V13, 64-core processor, 2.44 GHz

Required drivers

To take advantage of the GPU capabilities of [NVadsA10_v5](#), [NCasT4_v3](#), [NCv3](#), and [NC_A100_v4](#) VMs, you need to install NVIDIA GPU drivers.

To use AMD processors on [NVadsA10](#), [NCasT4_v3](#), and [NC_A100_v4](#) VMs, you need to install AMD drivers.

Samadii EM installation

Before you install Samadii EM, you need to deploy and connect a VM, install an eligible Windows 10 image, and install the required NVIDIA and AMD drivers.

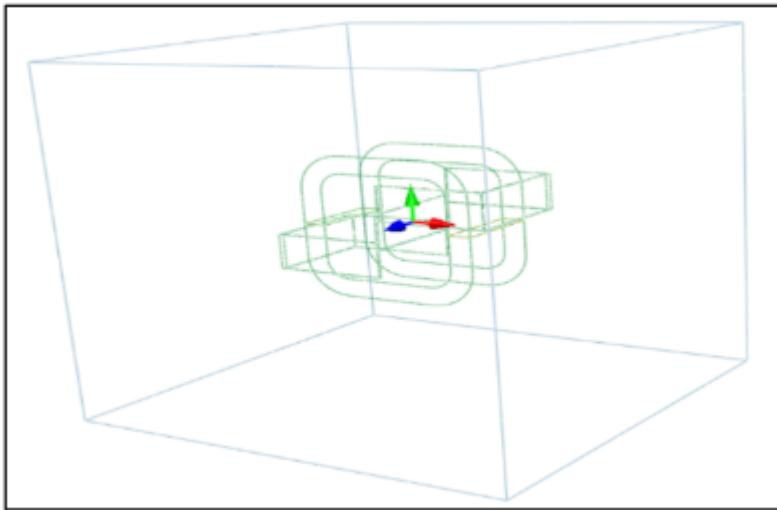
For information about eligible Windows images, see [How to deploy Windows 10 on Azure](#) and [Use Windows client in Azure for dev/test scenarios](#).

For information about deploying the VM and installing the drivers, see [Run a Windows VM on Azure](#).

The product installation process involves installing a license server, installing Samadii EM, and configuring the license server. For more information about installing Samadii EM, contact [Metariver Technology](#).

Samadii EM performance results

The nonlinear current model was used for testing:



- **Model size:** 219,398
- **Solver:** Samadii EM V21 V22 R1

To get a baseline, this model was tested on an on-premises VM with the following configuration:

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

Processor	GPU	Elapsed time (seconds)
Intel i7-3770 CPU	NVIDIA Titan X (Pascal)	4,471

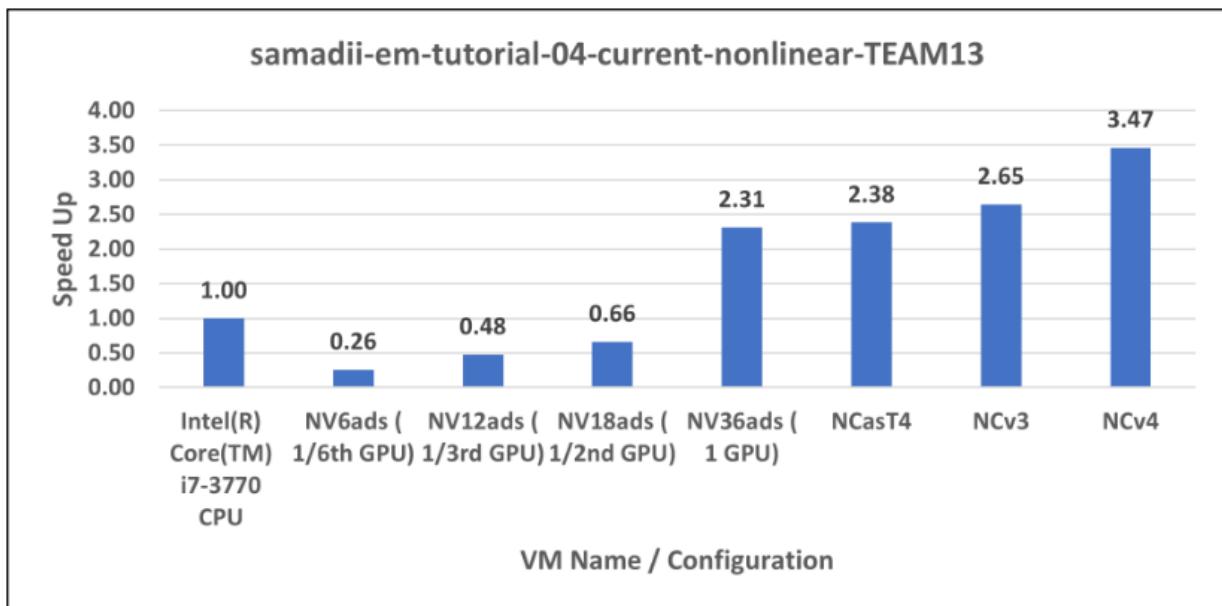
The following table shows the relative speed increases over this baseline:

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

VM	GPU	Number of GPUs used	Elapsed time (seconds)	Relative speed increase
NVadsA10_v5	NVIDIA A10	1/6	17,181	0.26
NVadsA10_v5	NVIDIA	1/3	9,350	0.48

VM	GPU	Number of GPUs used	Elapsed time (seconds)	Relative speed increase
A10				
NVadsA10_v5	NVIDIA A10	1/2	6,743	0.66
NVadsA10_v5	NVIDIA A10	1	1,933	2.31
NCasT4_v3	Tesla T4	1	1,875	2.38
NCv3	V100	1	1,689	2.65
NC_A100_v4	A100 80-GB PCIe	1	1,290	3.47

This graph shows the relative speed increases for the previous GPU configurations:



Azure cost

The following table shows wall-clock times in hours. To compute the total cost, multiply these times by the Azure VM hourly costs for NVadsA10v5, NCas_T4_v3, NCsv3, or NCA100v4 VMs. For the current hourly costs, see [Windows Virtual Machines Pricing ↗](#).

Only simulation runtime is included in the reported times. Application installation time and license costs aren't included.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

 [Expand table](#)

VM series	Number of GPUs	Wall-clock time (hours)
NVadsA10_v5	1/6	4.77
NVadsA10_v5	1/3	2.60
NVadsA10_v5	1/2	1.87
NVadsA10_v5	1	0.54
NCasT4_v3	1	0.52
NCv3	1	0.47
NC_A100_v4	1	0.36

Summary

- Samadii EM was successfully tested on NCv3, NCasT4_v3, NC_A100_v4, and NVadsA10_v5 VMs.
- For complex models, NC_A100_v4, NCv3, and NCasT4_v3 VMs, and the one-GPU configuration of the NVadsA10_v5 VM, all perform better than the NVadsA10_v5 VM with partial GPUs.
- For models that are less complex, configurations of NC_A100_v4 and NVadsA10_v5 VMs, including configurations that use partial GPUs, perform better than NCasT4_v3 and NCv3 VMs.
- If we take cost into consideration, NCasT4_v3 is the best choice.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- GPU-optimized virtual machine sizes
- Virtual machines on Azure
- Virtual networks and virtual machines on Azure
- Learning path: Run high-performance computing (HPC) applications on Azure

Related resources

- [Run a Windows VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Samadii Plasma on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [Samadii Plasma](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running Samadii Plasma on Azure.

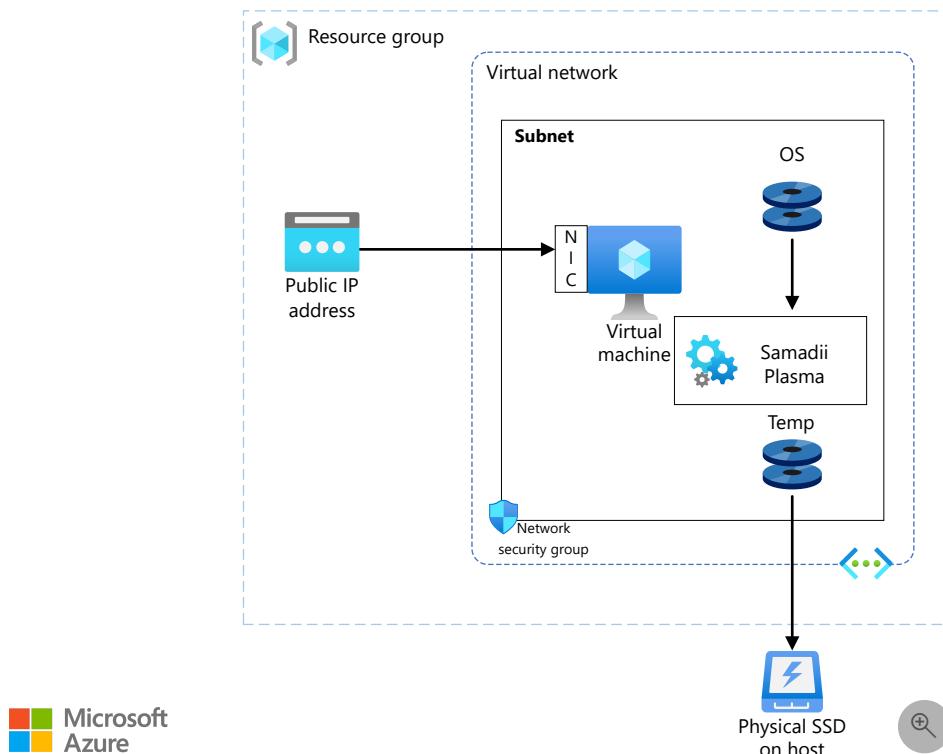
Samadii Plasma provides high-performance plasma physics simulation. It simulates plasma physics by using a method that's based on ion and electron particles. Samadii Plasma's high-speed electromagnetic field analysis capabilities and particle-based gas analysis, based on GPU technology, enable highly advanced plasma simulation.

Organizations that use Samadii Plasma include manufacturers of flat panel and OLED displays and manufacturers of semiconductors. This solution is ideal for the manufacturing and electronics industries.

Why deploy Samadii Plasma on Azure?

- Modern and diverse compute options to meet your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Strong performance scale-up, and configurations that provide either optimized scaling or optimized cost efficiency

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Windows VM. For information about deploying the VM and installing the drivers, see [Windows VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) restrict access to the VM.
 - A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) provides storage.

Compute sizing and drivers

The performance tests of Samadii Plasma on Azure used [NVv3](#), [NCasT4_v3](#), [NCv3](#), [ND_A100_v4](#), and [NC_A100_v4](#) series VMs running Windows 10. The following table provides details about the VMs.

[Expand table](#)

VM size	GPU	Number of vCPUs	Memory, in GiB	Maximum data disks	Number of GPUs	GPU memory, in GiB	Maximum uncached disk throughput, in IOPS / MBps	Temporary storage (SSD), in GiB	
Standard_NV12s_v3	Tesla M60	12	112	12	1	8	20,000 / 200	320	
Standard_NC4as_T4_v3	Tesla T4	4	28	8	1	16	-	180	
Standard_NC6s_v3	V100	6	112	12	1	16	20,000 / 200	736	
Standard_ND96asr_v4	A100	96	900	32	8	40	80,000 / 800	6,000	
Standard_NC24ads_A100_v4	A100	24	220	32	1	80	30,000 / 1,000	1,123	

Required drivers

To take advantage of the GPU capabilities of [NVv3](#), [NCasT4_v3](#), [NCv3](#), [ND_A100_v4](#), and [NC_A100_v4](#) series VMs, you need to install NVIDIA GPU drivers.

To use AMD processors on [NVv3](#), [NCasT4_v3](#), [NCv3](#), [ND_A100_v4](#), and [NC_A100_v4](#) series VMs, you need to install AMD drivers.

Samadii Plasma installation

Before you install Samadii Plasma, you need to deploy and connect a VM and install the required NVIDIA and AMD drivers.

For information about deploying the VM and installing the drivers, see [Run a Windows VM on Azure](#).

Important

NVIDIA Fabric Manager installation is required for VMs that use NVLink. ND_A100_v4 and NC_A100_v4 VMs use this technology.

Following are some prerequisites for running Samadii applications.

- Windows 10 (x64) OS
- One or more NVIDIA CUDA-enabled GPUs: Tesla, Quadro, or GeForce series
- Visual C++ 2010 SP1 Redistributable Package
- Microsoft MPI v7.1
- .NET Framework 4.5

The product installation process involves installing a license server, installing Samadii Plasma, and configuring the license server. For more information about installing Plasma, contact [Metariver Technology](#).

Samadii Plasma performance results

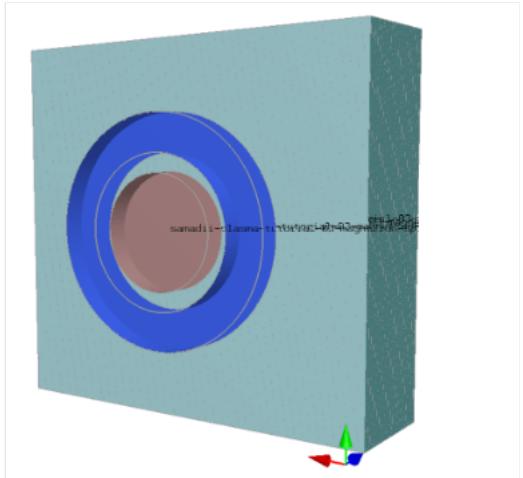
The following table shows the operating system versions and processors that were used for the tests.

[Expand table](#)

VM series	ND_A100_v4	NCv3	NCasT4_v3	NVv3	NC_A100_v4
Operating system version	Windows 10 Professional, version 20H2	Windows 10 Professional, version 20H2	Windows 10 Professional, version 20H2	Windows 10 Professional, version 20H2	Windows 10 Professional, version 21H2
OS architecture	x86-64	x86-64	x86-64	x86-64	x86-64
Processor	AMD EPYC 7V12, 64-core processor, 2.44 GHz (2 processors)	Intel Xeon CPU E5-2690 v4	AMD EPYC 7V12, 64-core processor, 2.44 GHz	Intel Xeon CPU E5-2690 v4	AMD EPYC 7V13, 64-core processor, 2.44 GHz

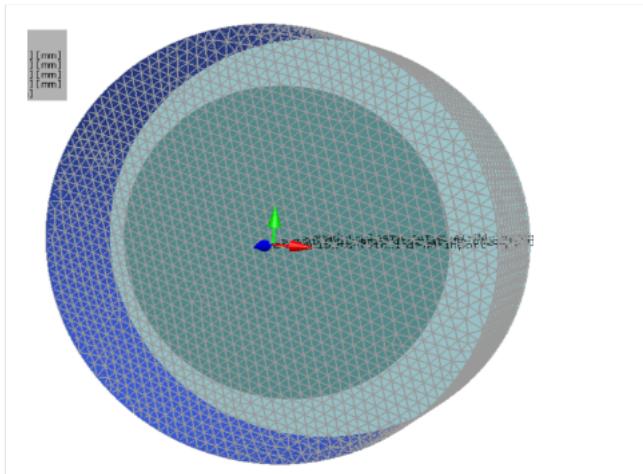
The following three models were used for testing.

Magnetron sputter



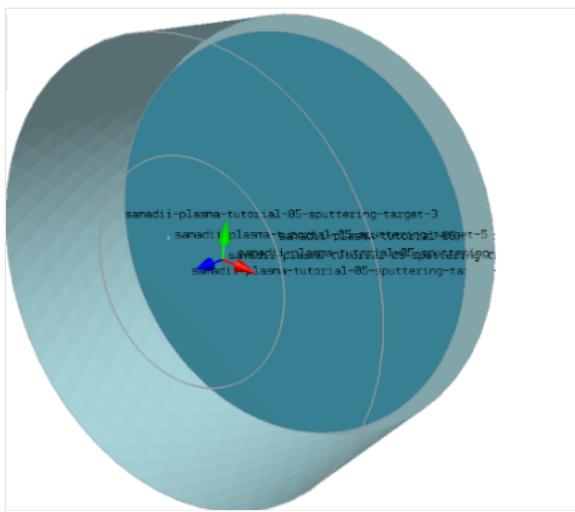
- Model size: 941,371
- Cell type: Shell and solid
- Solver: Samadii SCIV V21 R1
- Number of GPUs used for all simulations: One

Import inlet



- Model size: 141,967
- Cell type: Shell and solid
- Solver: Samadii SCIV V21 R1
- Number of GPUs used for all simulations: One

Sputtering target



- Model size: 15,991
- Cell type: Shell and solid
- Solver: Samadii SCIV V21 R1
- Number of GPUs used for all simulations: One

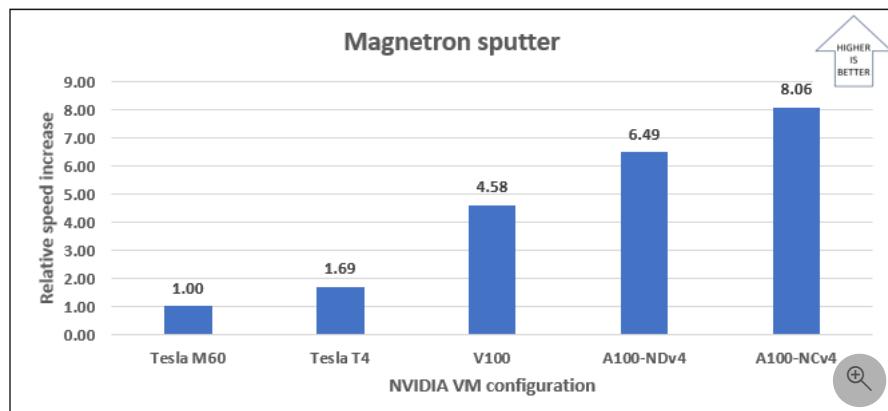
Results for the magnetron sputter model

The following table shows the elapsed runtimes and relative speed increases on each VM series. The NVv3 series VM is used as a baseline for the relative speed increases.

[Expand table](#)

VM series	GPU	Elapsed time, in seconds	Relative speed increase
NVv3	Tesla M60	12,825.36	N/A
NCasT4_v3	Tesla T4	7,606.59	1.69
NCv3	V100	2,798.55	4.58
ND_A100_v4	A100	1,977	6.49
NC_A100_v4	A100	1,590.83	8.06

This graph shows the relative speed increases.



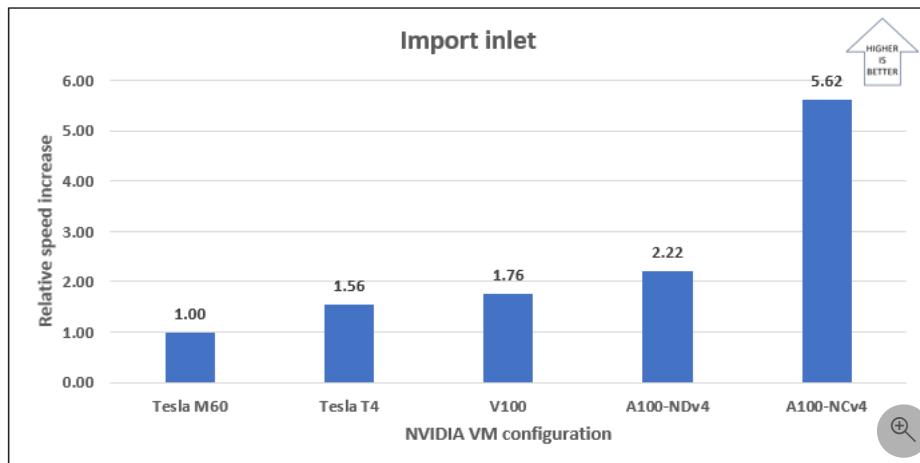
Results for the import inlet model

The following table shows the elapsed runtimes and relative speed increases on each VM series. The NVv3 series VM is used as a baseline for the relative speed increases.

[Expand table](#)

VM series	GPU	Elapsed time, in seconds	Relative speed increase
NVv3	Tesla M60	248.99	N/A
NCasT4_v3	Tesla T4	159.61	1.56
NCv3	V100	141.59	1.76
ND_A100_v4	A100	112	2.22
NC_A100_v4	A100	44.27	5.62

This graph shows the relative speed increases.



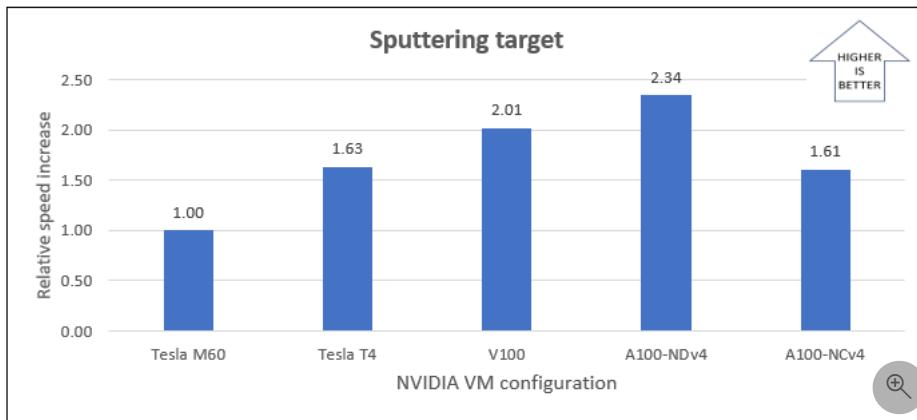
Results for the sputtering target model

The following table shows the elapsed runtimes and relative speed increases on each VM series. The NVv3 series VM is used as a baseline for the relative speed increases.

[Expand table](#)

VM series	GPU	Elapsed time, in seconds	Relative speed increase
NVv3	Tesla M60	13.82	N/A
NCasT4_v3	Tesla T4	8.46	1.63
NCv3	V100	6.86	2.01
ND_A100_v4	A100	5.9	2.34
NC_A100_v4	A100	8.61	1.61

This graph shows the relative speed increases.



Azure cost

The following tables present simulation runtimes in hours. To compute the total cost, multiply these times by the Azure VM hourly costs for NVv3, NCsT4_v3, NCv3, ND_A100_v4, and NC_A100_v4 series VMs. For the current hourly costs, see [Windows Virtual Machines Pricing](#).

Only simulation runtime is considered in these cost calculations. Application installation time and license costs aren't included.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

Cost, magnetron sputter model

[Expand table](#)

VM size	GPU	Number of GPUs	Wall-clock time, in hours
Standard_NV12s_v3	Tesla M60	1	3.56
Standard_NC4as_T4_v3	Tesla T4	1	2.11
Standard_NC6s_v3	V100	1	0.78
Standard_ND96asr_v4	A100	1	0.55
Standard_NC24ads_A100_v4	A100	1	0.44

Cost, import inlet model

[Expand table](#)

VM size	GPU	Number of GPUs	Wall-clock time, in hours
Standard_NV12s_v3	Tesla M60	1	0.07
Standard_NC4as_T4_v3	Tesla T4	1	0.04
Standard_NC6s_v3	V100	1	0.04
Standard_ND96asr_v4	A100	1	0.03
Standard_NC24ads_A100_v4	A100	1	0.01

Cost, sputtering target model

VM size	GPU	Number of GPUs	Wall-clock time, in hours
Standard_NV12s_v3	Tesla M60	1	0.0038
Standard_NC4as_T4_v3	Tesla T4	1	0.0024
Standard_NC6s_v3	V100	1	0.0019
Standard_ND96asr_v4	A100	1	0.0016
Standard_NC24ads_A100_v4	A100	1	0.0024

Summary

- Samadii Plasma was tested on Azure ND_A100_v4, NCv3, NCasT4_v3, NVv3, and NC_A100_v4 series VMs.
- For complex models, like magnetron sputter and import inlet, the Standard_NC24ads_A100_v4 VM provides the best performance.
- For models with less complexity, the NCasT4_v3 VM provides good scale-up, and the performance-to-cost ratio is better than that of the other VMs tested.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director of Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run HPC applications on Azure](#)

Related resources

- [Run a Windows VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Samadii SCIV on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [Samadii SCIV](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running Samadii SCIV on Azure.

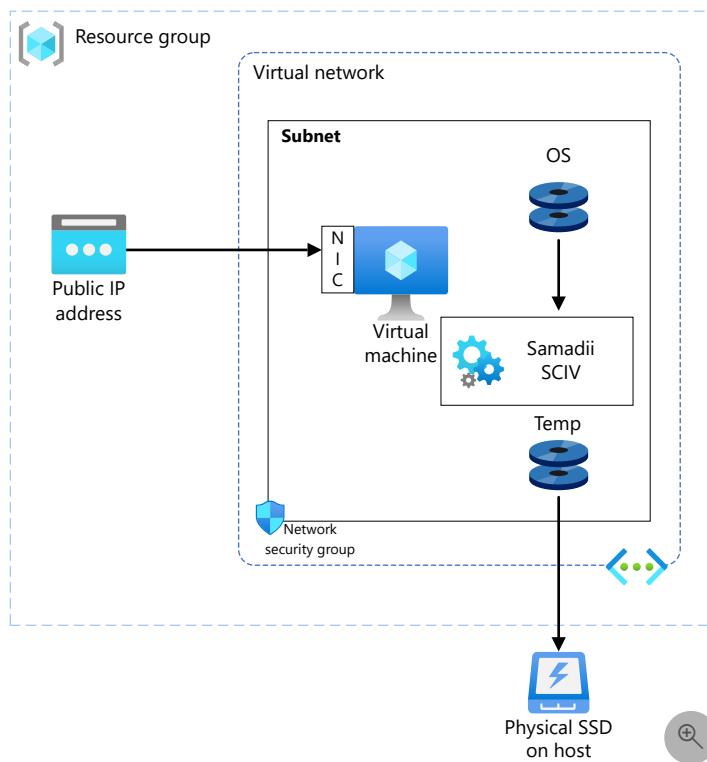
Samadii SCIV (Statistical Contact in Vacuum) analyzes fluid behavior, deposition processes, and chemical reactions on rarefied gas regions by using the direct simulation Monte Carlo (DSMC) method. To calculate the physical phenomena represented by the Boltzmann equation, the DSMC method uses representative particles, which replace the real molecules. SCIV also provides functions for traditional flow simulation, display deposition processes, and semiconductor device analysis in rarefied gas regions. Samadii SCIV is based on a GPU architecture and uses CUDA technology.

SCIV is used by manufacturers of display devices and semiconductors, in aerospace and manufacturing, and in other industries.

Why deploy Samadii SCIV on Azure?

- Modern and diverse compute options to meet your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Strong performance scale-up, with configurations that provide either optimized scaling or optimized cost efficiency

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Windows VM. For information about deploying the VM and installing the drivers, see [Windows VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) restrict access to the VM.
 - A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) provides storage.

Compute sizing and drivers

The performance tests of Samadii SCIV on Azure used [NVv3](#), [NCasT4_v3](#), [NCv3](#), [ND_A100_v4](#), and [NC_A100_v4](#) series VMs running Windows 10. The following table provides details about the VMs.

[Expand table](#)

VM size	GPU name	vCPUs	Memory, in GiB	Maximum data disks	GPUs	GPU memory, in GiB	Maximum uncached disk throughput, in IOPS / MBps	Temporary storage (SSD), in GiB	Max NICs
Standard_NV12s_v3	Tesla M60	12	112	12	1	8	20,000 / 200	320	4
Standard_NC4as_T4_v3	Tesla T4	4	28	8	1	16	-	180	2
Standard_NC6s_v3	V100	6	112	12	1	16	20,000 / 200	736	4
Standard_ND96asr_v4	A100	96	900	32	8	40	80,000 / 800	6,000	8
Standard_NC24ads_A100_v4	A100	24	220	32	1	80	30,000 / 1,000	1,123	2

Required drivers

To take advantage of the GPU capabilities of NVv3, NCasT4_v3, NCv3, ND_A100_v4, and NC_A100_v4 series VMs, you need to install NVIDIA GPU drivers.

To use AMD processors on NCasT4_v3, NCv3, ND_A100_v4, and NC_A100_v4 series VMs, you need to install AMD drivers.

Samadii SCIV installation

Before you install SCIV, you need to deploy and connect a VM and install the required NVIDIA and AMD drivers.

For information about deploying the VM and installing the drivers, see [Run a Windows VM on Azure](#).

Important

NVIDIA Fabric Manager installation is required for VMs that use NVLink. ND_A100_v4 and NC_A100_v4 series VMs use this technology.

Following are some prerequisites for running Samadii applications.

- Windows 10 (x64) OS
- One or more NVIDIA CUDA-enabled GPUs: Tesla, Quadro, or GeForce series
- Visual C++ 2010 SP1 Redistributable Package
- Microsoft MPI v7.1
- .NET Framework 4.5

The product installation process involves installing a license server, installing Samadii SCIV, and configuring the license server. For more information about installing SCIV, contact [Metariver Technology](#).

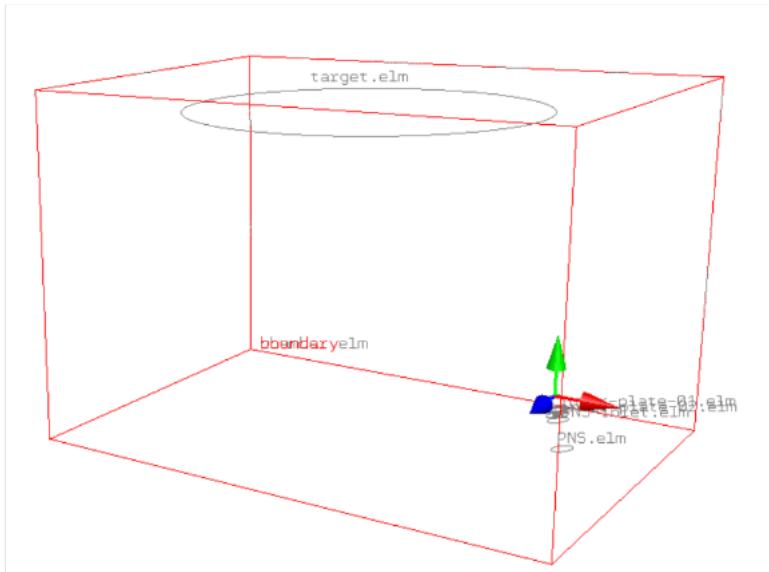
Samadii SCIV performance results

The following table shows the operating system versions and processors that were used for the tests.

[Expand table](#)

VM series	ND_A100_v4	NCv3	NCasT4_v3	NVv3	NC_A100_v4
Operating system version	Windows 10 Professional, version 20H2	Windows 10 Professional, version 20H2	Windows 10 Professional, version 20H2	Windows 10 Professional, version 20H2	Windows 10 Professional, version 21H2
OS architecture	x86-64	x86-64	x86-64	x86-64	x86-64
Processor	AMD EPYC 7V12, 64-core processor, 2.44 GHz (2 processors)	Intel Xeon CPU E5-2690 v4	AMD EPYC 7V12, 64-core processor, 2.44 GHz	Intel Xeon CPU E5-2690 v4	AMD EPYC 7V13, 64-core processor, 2.44 GHz

The PNS model was used for testing:



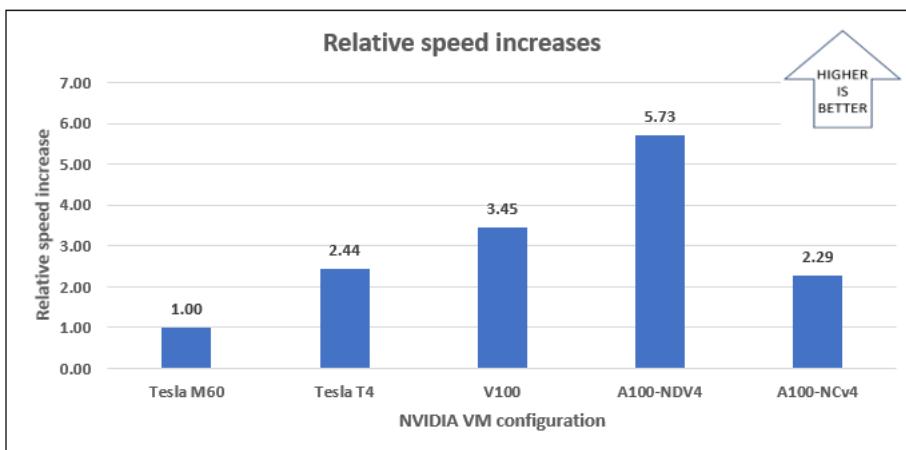
- Size: 17,652
- Cell type: Shell
- Solver: Samadii SCIV V21 R1
- Number of GPUs used for all simulations: One

The following table shows the elapsed runtimes and relative speed increases on each VM series. The NVv3 series VM is used as a baseline for the relative speed increases.

[Expand table](#)

VM series	Number of GPUs	Elapsed time, in seconds	Relative speed increase
NVv3	1	93,483.74	N/A
NCasT4_v3	1	38,311.8	2.44
NCv3	1	27,096.83	3.45
ND_A100_v4	1	16,322.98	5.73
NC_A100_v4	1	40,895.55	2.29

This graph shows the relative speed increases:



Azure cost

The following table presents simulation runtimes in hours. To compute the total cost, multiply these times by the Azure VM hourly costs for NVv3, NCasT4_v3, NCv3, ND_A100_v4, and NC_A100_v4 series VMs. For the current hourly costs, see [Windows Virtual Machines Pricing](#).

Only simulation runtime is considered in these cost calculations. Application installation time and license costs aren't included.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

[Expand table](#)

VM size	Number of GPUs	Wall-clock time, in hours
Standard_NV12s_v3	1	25.97
Standard_NC4as_T4_v3	1	10.64
Standard_NC6s_v3	1	7.53
Standard_ND96asr_v4	1	4.53
Standard_NC24ads_A100_v4	1	11.36

Summary

- Samadii SCIV was tested on ND_A100_v4, NCv3, NCasT4_v3, NVv3, and NC_A100_v4 series VMs.
- SCIV performance scales well on NCasT4_v3, NCv3, ND_A100_v4, and NC_A100_v4 series VMs.
- The NCasT4_v3 series VMs are more cost efficient than ND_A100_v4, NCv3, and NVv3 series VMs.
- Of the five VM series, ND_A100_v4 provides the best relative speed increase.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director of Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run HPC applications on Azure](#)

Related resources

- [Run a Windows VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Sandi HiFUN on an Azure virtual machine

Azure Virtual Machines Azure Virtual Network

This article briefly describes the steps for running [Sandi HiFUN](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running HiFUN on Azure.

HiFUN is a general-purpose computational fluid dynamics (CFD) application. You can use it to simulate airflow over aircraft, automobiles, and structures like buildings and ships.

HiFUN has these capabilities:

- Provides a robust, fast, and accurate solver for aerodynamic design data
- Uses an unstructured cell-centered finite volume method that can handle complex geometries and flow physics
- Handles MPI directives for parallel computing on distributed-memory HPC
- Can scale over thousands of processor cores
- Can be ported to NVIDIA GPUs for parallel computing via [OpenACC](#) constructs

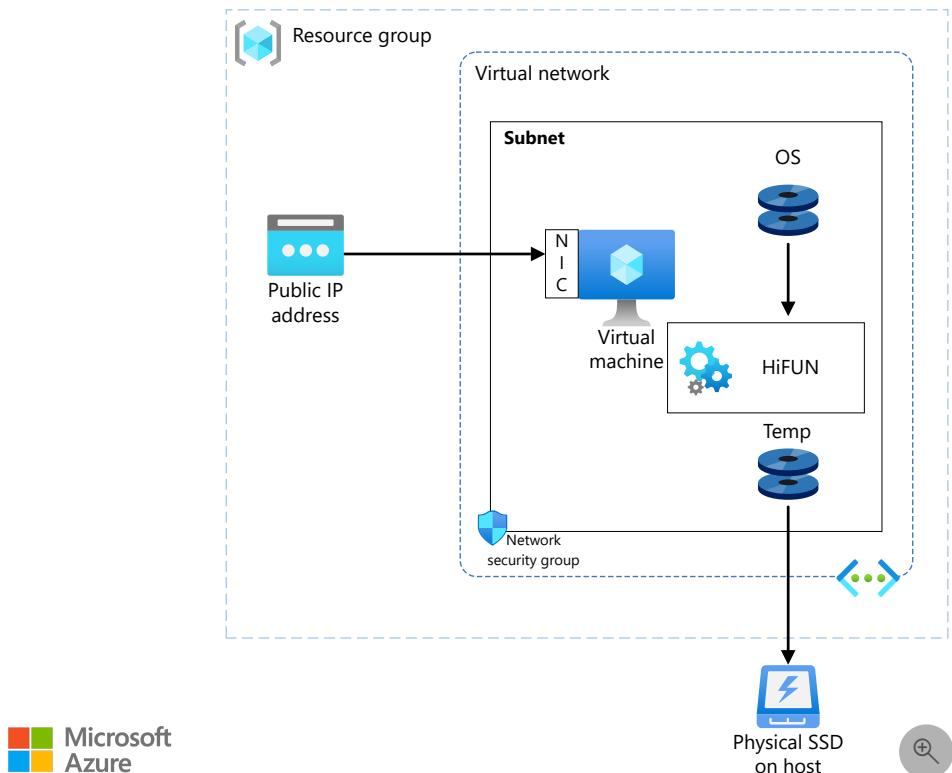
Sandi HiFUN is used in the aerospace, automotive, industrial, and wind/turbine industries.

[HBv3](#) and [NCasT4_v3](#) series VMs were used to test the performance of HiFUN on Azure.

Why deploy HiFUN on Azure?

- Modern and diverse compute options to meet your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create Linux virtual machines.

- Azure Virtual Network [↗](#) is used to create a private network infrastructure in the cloud.

Compute sizing and drivers

Performance tests of HiFUN on Azure used [HBv3](#) and [NCasT4_v3](#) VMs running the Linux CentOS operating system. The following table provides details about HBv3-series VMs.

[Expand table](#)

VM size	vCPU	Memory (GiB)	Memory bandwidth (Gbps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120rs_v3	120	448	350	1.9	3.0	3.5	200	32
Standard_HB120-96rs_v3	96	448	350	1.9	3.0	3.5	200	32
Standard_HB120-64rs_v3	64	448	350	1.9	3.0	3.5	200	32
Standard_HB120-32rs_v3	32	448	350	1.9	3.0	3.5	200	32
Standard_HB120-16rs_v3	16	448	350	1.9	3.0	3.5	200	32

The following table provides details about NCasT4_v3 VMs.

[Expand table](#)

VM size	vCPU	Memory, in GiB	Temporary storage (SSD), in GiB	GPU	GPU memory, in GiB	Maximum data disks	Maximum NICs / Expected network bandwidth, in Mbps
Standard_NC4as_T4_v3	4	28	180	1	16	8	2 / 8,000
Standard_NC8as_T4_v3	8	56	360	1	16	16	4 / 8,000
Standard_NC16as_T4_v3	16	110	360	1	16	32	8 / 8,000
Standard_NC64as_T4_v3	64	440	2,880	4	64	32	8 / 32,000

Required drivers

To use InfiniBand, you need to enable [InfiniBand drivers](#).

To enable the GPU capabilities of [NCasT4_v3](#) VMs, you need to install NVIDIA GPU drivers.

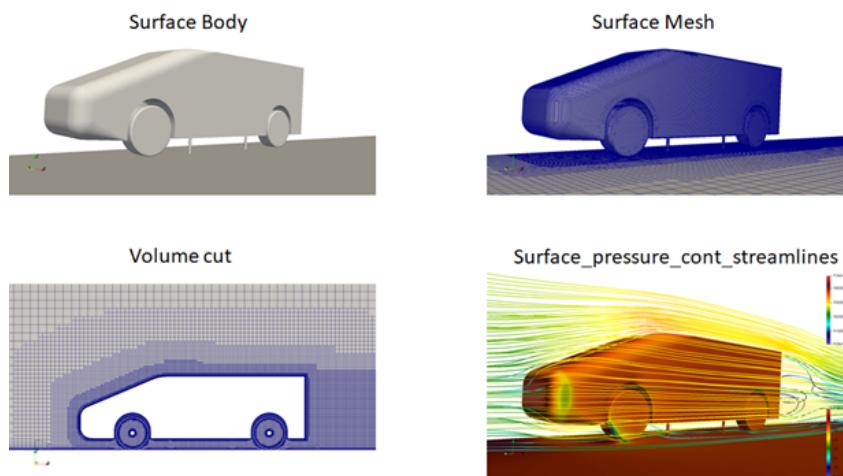
HiFUN installation

Before you install HiFUN, you need to deploy and connect a VM. For information about deploying the VM and installing the drivers, see [Run a Linux VM on Azure](#).

For more information about installing HiFUN on an Azure VM, you can contact Sandi at sales@sandi.co.in or info@sandi.co.in.

HiFUN performance results

The Windsor model is used in this performance evaluation.



The following tables provide details about the model.

[Expand table](#)

Flow conditions	
Parameter	Value
Mach number	0.1207
Velocity	40 m/s
Reynolds number	1.8 million
Flow direction	Aligned to x axis

[Expand table](#)

Workload	
Model	Windsor car body
Number of volumes	7.456 million

Performance results for HiFUN 4.1.1 on HBv3

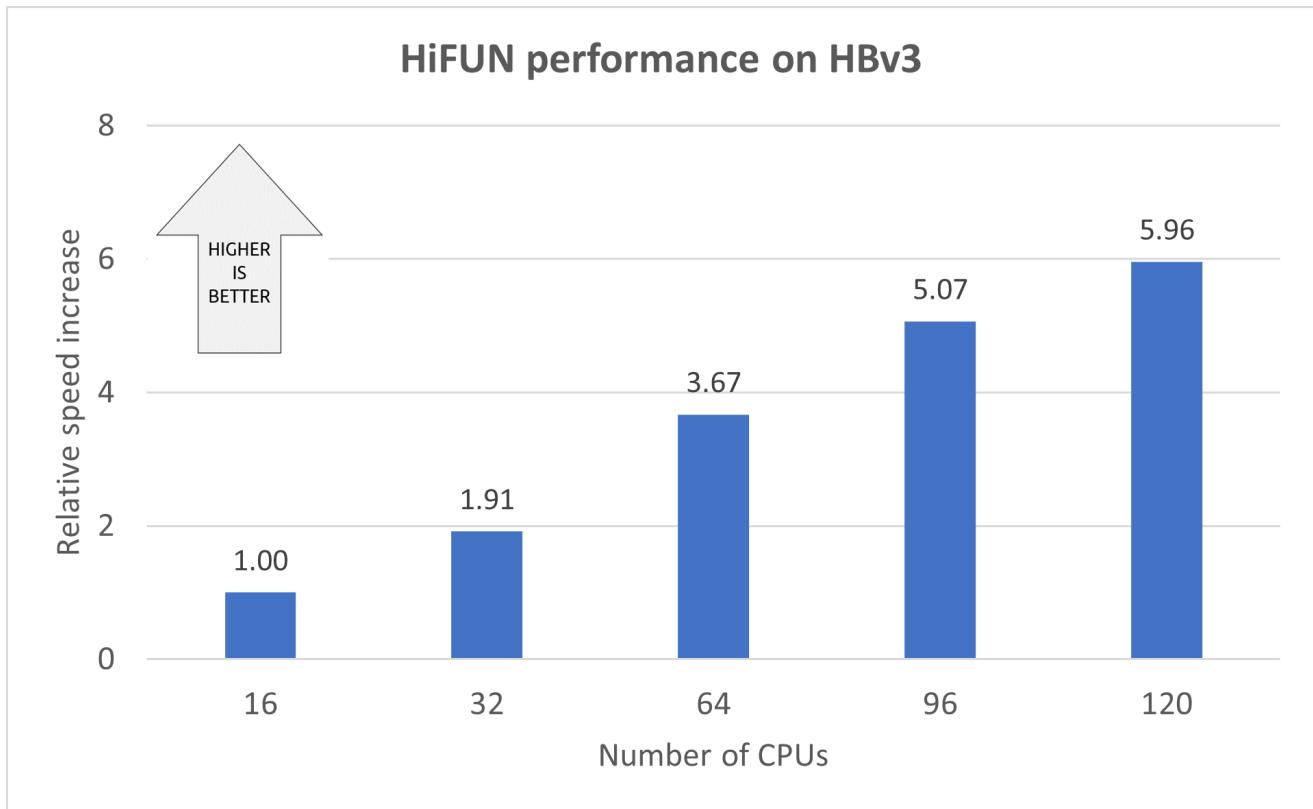
[Expand table](#)

VM size	Number of iterations	Time per iteration (seconds) ¹	Relative speed increase
Standard_HB120-16rs_v3	100	10.13	1.00
Standard_HB120-32rs_v3	100	5.29	1.91
Standard_HB120-64rs_v3	100	2.76	3.67

VM size	Number of iterations	Time per iteration (seconds) ¹	Relative speed increase
Standard_HB120-96rs_v3	100	2.00	5.07
Standard_HB120rs_v3	100	1.70	5.96

¹ To negate the effect of input/output operations per second (IOPS), the average time of 51-60 recorded iterations is presented here.

This graph shows the relative speed increase² as the number of CPUs increases:



² The 16-CPU configuration is used as a baseline for the relative-speed calculations.

Performance results for HiFUN 4.1.1 on NCasT4

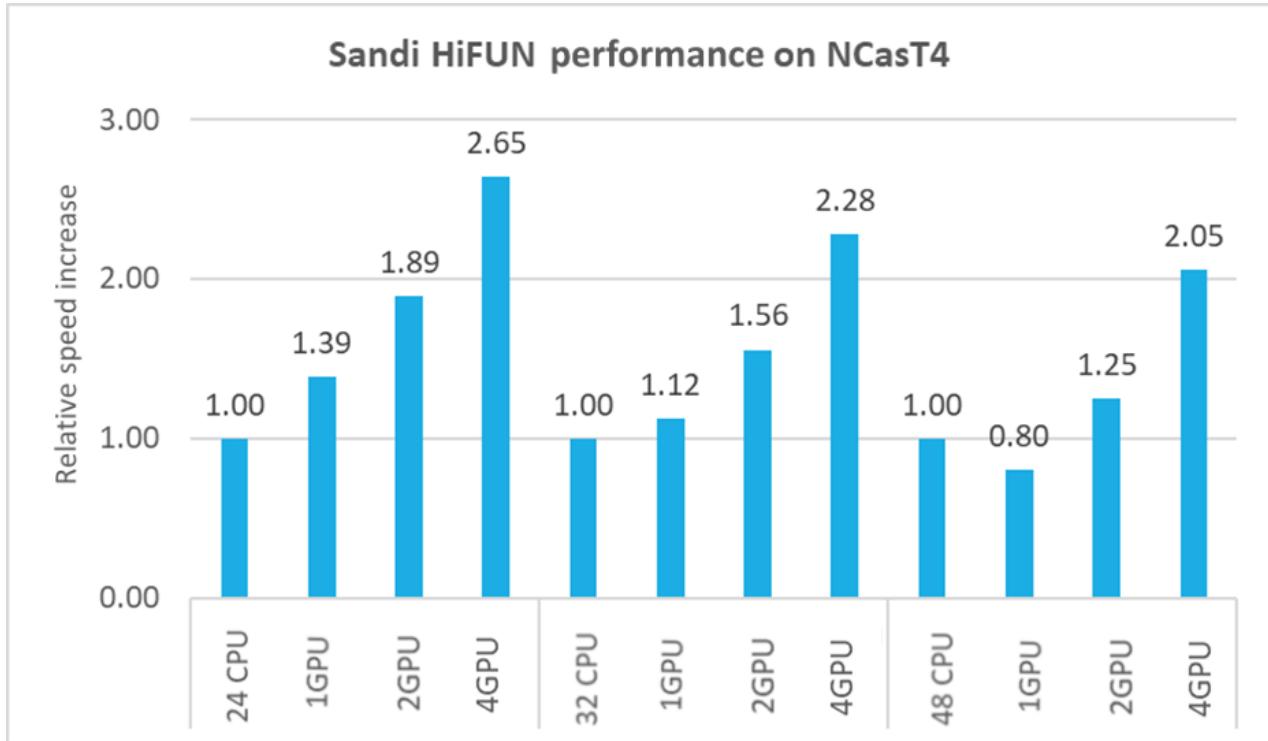
[Expand table](#)

CPU configuration	Number of CPUs/GPUs	Number of iterations	Time per iteration (seconds) ³	Relative speed increase
24 CPU	24 CPU	100	7.70	1.00
	1 GPU	100	5.55	1.39
	2 GPU	100	4.07	1.89
	4 GPU	100	2.91	2.65
32 CPU	32 CPU	100	5.59	1.00
	1 GPU	100	4.99	1.12
	2 GPU	100	3.59	1.56
	4 GPU	100	2.45	2.28

CPU configuration	Number of CPUs/GPUs	Number of iterations	Time per iteration (seconds) ³	Relative speed increase
48 CPU	48 CPU	100	4.15	1.00
	1 GPU	100	5.18	0.80
	2 GPU	100	3.32	1.25
	4 GPU	100	2.02	2.05

³ To negate the effect of IOPS, the average time of 51-60 recorded iterations is presented here.

This graph shows the relative speed increase⁴ as the number of GPUs increases:



⁴ The CPU configurations listed in the preceding table are used as the baselines for the relative-speed calculations.

Additional notes about the tests

- HiFUN was successfully tested on HBv3 and NCasT4 VMs on Azure.
- Every CPU increase provides a good speed increase on all VM sizes. The peak speed increase of 5.96x is achieved with 120 CPUs.
- Every GPU increase provides a good speed increase on all CPU configurations. The peak speed increase of 2.65x is achieved with 4 GPUs.

Azure cost

Only model running time (wall clock time) is considered for these cost calculations. Application installation time isn't considered. The numbers presented here are indicative of your potential results. The actual numbers depend on the size of the model.

You can use the [Azure pricing calculator](#) to estimate costs for your configuration.

The following tables provide elapsed times in hours. To compute the total cost, multiply these times by the hourly costs for [Linux VMs](#).

HBv3

[Expand table](#)

VM size	Number of CPUs	Elapsed time (hours)
Standard_HB120-16rs_v3	16	0.297
Standard_HB120-32rs_v3	32	0.156
Standard_HB120-64rs_v3	64	0.083
Standard_HB120-96rs_v3	96	0.061
Standard_HB120rs_v3	120	0.052

NC64as_T4_v3

 [Expand table](#)

CPU/GPU	Elapsed time (hours)
CPU	0.116
4 GPU	0.057

Summary

Azure provides robust compute services that support GPU-intensive workloads and offers unlimited scalability for HPC applications. You can use H-series virtual machines for memory-bound applications and N-series virtual machines for graphic-intensive applications.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- Kalai Selvan | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director, Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [High performance computing VM sizes](#)
- [Linux virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run HPC applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)

- HPC cluster deployed in the cloud

Deploy Siemens NX on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [Siemens NX](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running NX on Azure.

Organizations use NX for design, simulation, and manufacturing solutions that enable digital twin technology. Siemens NX:

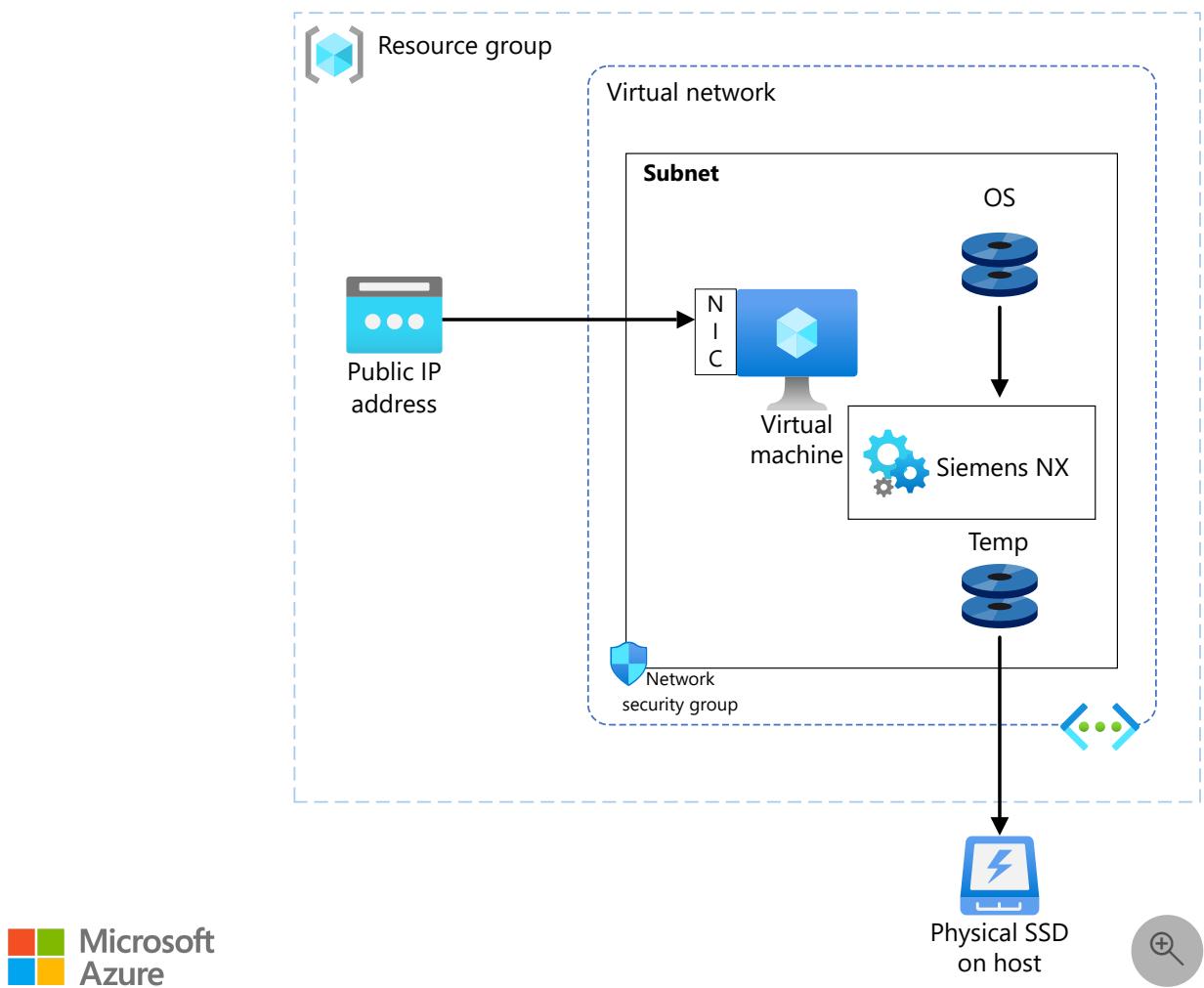
- Provides layout piping and instrumentation diagrams in two dimensions while maintaining the design tied to the 3D model.
- Eliminates re-creation of annotated 2D CAD drawings by using legacy data with model-based definition.

NX is used in the automotive sector and for projects ranging from supersonic cars to drones for the medical industry. These scenarios also relate to the aerospace and healthcare industries.

Why deploy Siemens NX on Azure?

- Modern and diverse compute options to align to your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Successful runs of all ATS and NXCP test cases

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Windows VM.
 - For information about deploying the VM and installing the drivers, see [Windows VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

Performance tests of NX on Azure used [NVv3](#) and [NCsT4_v3](#) series VMs running Windows. The following table provides the configuration details.

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

VM size	vCPU	Memory, in GiB	SSD, in GiB	Number of GPUs	GPU memory, in GiB	Maximum data disks
Standard_NV12s_v3	12	112	320	1	8	12
Standard_NV24s_v3	24	224	640	2	16	24
Standard_NC16as_T4_v3	16	110	360	1	16	32

Required drivers

To take advantage of the GPU capabilities of [NVv3](#) and [NCasT4_v3](#) series VMs, you need to install the NVIDIA GPU drivers.

To use CPUs on [NCasT4_v3](#), you need to install the AMD drivers.

NX installation

Before you install NX, you need to deploy and connect a VM, install an eligible Windows 10 or Windows 11 image, and install the required NVIDIA and AMD drivers.

For information about eligible Windows images, see [How to deploy Windows 10 on Azure](#) and [Use Windows client in Azure for dev/test scenarios](#).

For information about deploying the VM and installing the drivers, see [Run a Windows VM on Azure](#).

After you install the drivers, install Siemens PLM License Server and the NX application. You can download these applications and get download instructions from the [Siemens website](#).

 **Note**

You need to install the [Java Runtime Environment \(64-bit\) for NX](#) before you install NX.

NX performance results

Siemens Automated Testing Studio (ATS) and NXCP were used to test NX. For information about installing and using these tools, see the [Siemens website](#).

NXCP test results

The Siemens NXCP tool was used to run these tests. Test cases are grouped to demonstrate various capabilities. Each test group consists of multiple test cases.

The following table shows the test results.

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

Test group name	Number of test cases	1 GPU (NV12s_v3)	2 GPUs (NV24s_v3)	1 GPU (NCasT4, 16 CPU)
GDAT_LEGACY	15	Pass	Pass	Pass
TCVIS 2007	10	Pass	Pass	Pass
TECNOMATIX_OpenGL Display	2	Pass	Pass	Pass
TECNOMATIX_OpenGL Buffer	2	Pass	Pass	Pass
NX1847_Manual	7	Pass	Pass	Pass
NX1899_Manual	7	Pass	Pass	Pass
NX 1899 Mark	6	Pass	Pass	Pass

ATS test results

The following table shows ATS test results.

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

Case number	Description	Case ID	1 GPU (NV12s_v3)	2 GPUs (NV24s_v3)	1 GPU (NCasT4, 16 CPU)
AT-01	Verifies that Listing Window shows data correctly	Listing Window	1 minute, 11 seconds	31 seconds	18 seconds

Case number	Description	Case ID	1 GPU (NV12s_v3)	2 GPUs (NV24s_v3)	1 GPU (NCasT4, 16 CPU)
AT-02	Verifies correct mirror display in various views	Mirror Display	1 minute, 40 seconds	1 minute, 8 seconds	1 minute, 10 seconds
AT-03	Verifies de-emphasis displays	Deemphasis	1 minute, 13 seconds	1 minute	57 seconds
AT-04	Verifies grid displays	Grid Display	1 minute, 47 seconds	1 minute, 16 seconds	51 seconds
AT-05	Verifies correct display of raster images in various view ports	Raster Image	20 seconds	19 seconds	23 seconds
AT-06	Shows multiple views in one window	Multiple Views	23 seconds	26 seconds	25 seconds
AT-07	Verifies the background setting in the wireframe view	Background Setting Wireframe	45 seconds	29 seconds	34 seconds
AT-08	Verifies correct display of light and shadow in Advanced Studio	Light Direction	21 seconds	20 seconds	23 seconds
AT-09	Verifies the display section	Display Section	4 minutes, 35 seconds	2 minutes, 8 seconds	1 minute, 52 seconds
AT-10	Verifies display modes	Display Modes	17 minutes	45 seconds	43 seconds
AT-11	Activates HD3D and	HD3D	1 minute, 37 seconds	47 seconds	47 seconds

Case number	Description	Case ID	1 GPU (NV12s_v3)	2 GPUs (NV24s_v3)	1 GPU (NCasT4, 16 CPU)
	representation status with tags				
AT-12	Activates face analysis and verifies displays with reflection	Face Analysis	52 minutes	41 minutes	40 minutes, 33 seconds
AT-13	Rotates views in sync	Synchronize View	18 seconds	16 seconds	15 seconds

Azure cost

The following table presents wall-clock times that you can use to calculate Azure costs. You can use the times presented here together with the Azure hourly rates for NVv3 and NCas_T4_v3 series VMs to calculate costs. For the current hourly costs, see [Windows Virtual Machines Pricing](#).

Only elapsed running time of the tests is considered for these cost calculations. Application installation time isn't considered.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

[Expand table](#)

VM size	Number of GPUs	Wall-clock time
NV12s_v3	1	1 hour, 23 minutes, 10 seconds
NV24s_v3	2	50 minutes, 25 seconds
NC16as_T4_v3	1	49 minutes, 18 seconds

Summary

- NX was successfully tested on NVv3 and NCas_T4 series VMs on Azure.
- All test cases, for both ATS and NXCP, ran successfully on 1-GPU Standard_NV12s_v3, 2-GPU Standard_NV24s_v3, and 1-GPU Standard_NC16as_T4_v3 configurations.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Windows virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Windows VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Siemens Tecnomatix on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [Siemens Tecnomatix](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running Siemens Tecnomatix on Azure.

Siemens Tecnomatix is a comprehensive portfolio of digital manufacturing solutions. It includes part manufacturing, assembly planning, resource planning, plant simulation, human performance, quality, production management, and manufacturing data management. Tecnomatix:

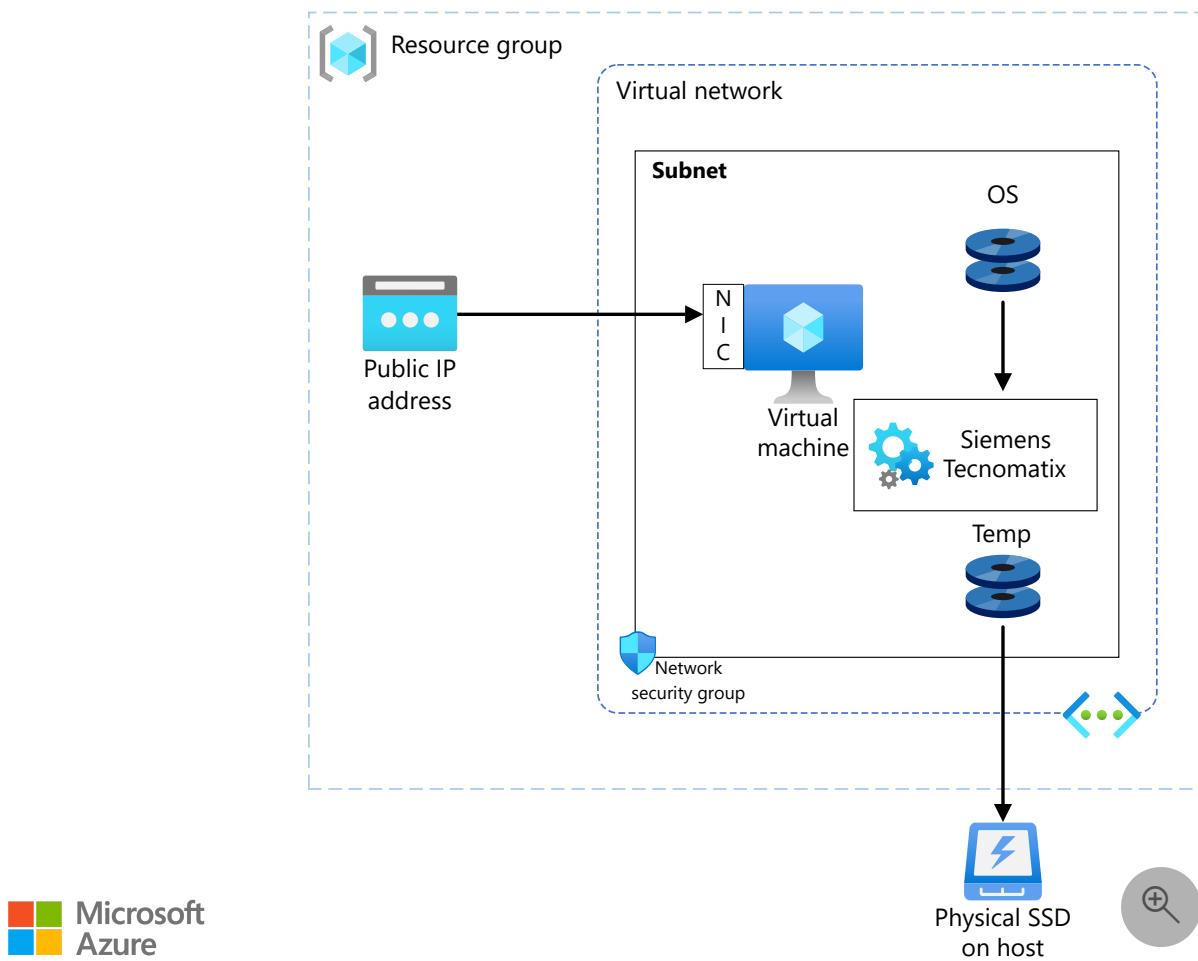
- Provides discrete event simulation and statistical analysis capabilities to optimize material handling, logistics, machine utilization, and labor requirements.
- Uses powerful graphical visualization outputs for automatic bottleneck detection, analysis of throughput, and utilization of machines, resources, and buffers.
- Includes an integrated energy analyzer that dynamically visualizes energy and that shows current, maximum, and total energy consumption.
- Provides assembly simulation for virtual verification of all process details before the start of production.

Tecnomatix is used in the automotive, electronics, manufacturing, and aerospace industries.

Why deploy Tecnomatix on Azure?

- Modern and diverse compute options to align to your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Typical workloads easily handled

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Windows VM.
 - For information about deploying the VM and installing the drivers, see [Windows VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

Performance tests of Tecnomatix on Azure used [NVadsA10_v5](#) and [NCasT4_v3](#) series VMs running Windows 10. The following table provides details about the operating system and NVIDIA drivers.

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

	NVadsA10_v5	NCasT4_v3
Operating system	Windows 10 Enterprise	Windows 10 Professional
OS architecture	x86-64	x86-64
NVIDIA driver version	462.31	511.65
Cuda version	11.2	11.6

Required drivers

To take advantage of the GPU capabilities of [NVadsA10_v5](#) and [NCasT4_v3](#) series VMs, you need to install NVIDIA GPU drivers.

To use AMD CPUs on NVadsA10_v5 and NCasT4_v3 series VMs, you need to install AMD drivers.

Tecnomatix installation

Before you install Tecnomatix, you need to deploy and connect a VM, install an eligible Windows 10 image, and install the required NVIDIA and AMD drivers.

For information about eligible Windows images, see [How to deploy Windows 10 on Azure](#) and [Use Windows client in Azure for dev/test scenarios](#).

For information about deploying the VM and installing the drivers, see [Run a Windows VM on Azure](#).

To install Tecnomatix and get information about the installation process, see the [Siemens website](#).

Tecnomatix performance results

The Factory 51 model was used as a test case for these performance evaluations.



The following scenarios were tested:

- Shadows off, with animation
- Shadows off, without animation
- Shadows on, with animation
- Shadows on, without animation

For these scenarios, frames per second (FPS) and CPU time are noted in the following tables.

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

Shadows off, with animation		
	NVadsA10_v5	NCasT4_v3
FPS	28	19
CPU time (seconds)	8.98	43.22

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

Shadows off, without animation		
	NVadsA10_v5	NCasT4_v3

Shadows off, without animation

FPS	30	21
CPU time (seconds)	2.05	2.91

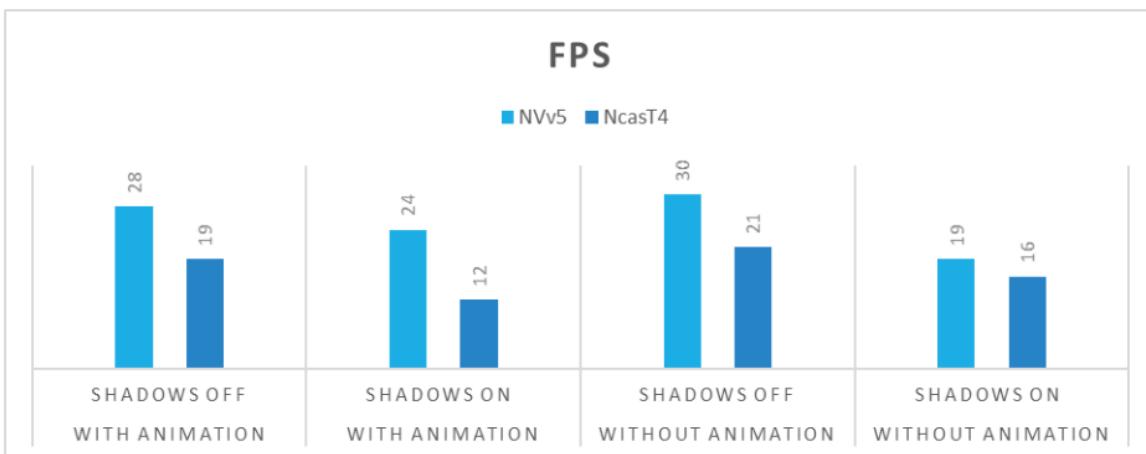
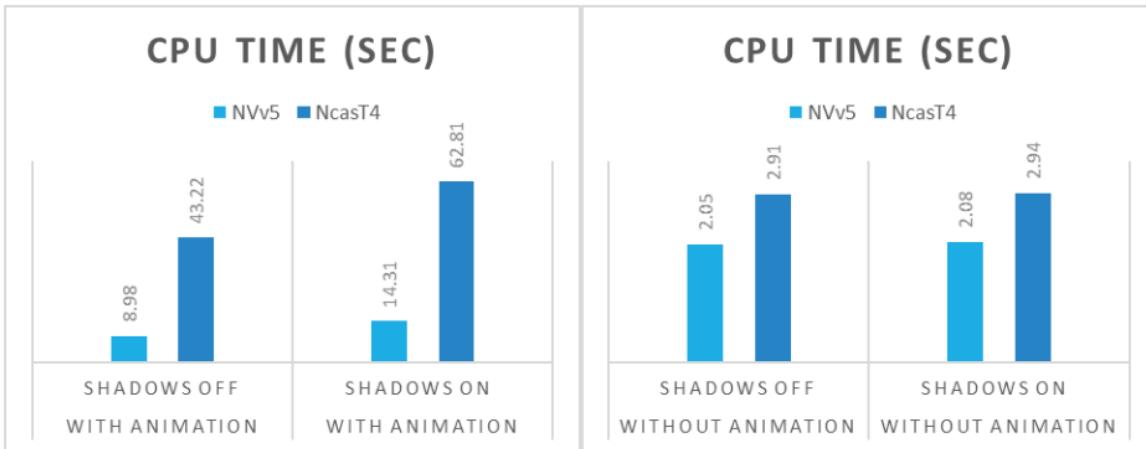
[\[\] Expand table](#)**Shadows on, with animation**

	NVadsA10_v5	NCasT4_v3
FPS	24	12
CPU time (seconds)	14.31	62.81

[\[\] Expand table](#)**Shadows on, without animation**

	NVadsA10_v5	NCasT4_v3
FPS	19	16
CPU time (seconds)	2.08	2.94

This information is presented in the following graphs:



Additional notes about tests

The default time was set for one hour for all tests. Debug was enabled in the Console. Real-time mode was disabled.

[Fast Forward Simulation](#) was used in these tests. Tests performed with this option enabled run without MU Animation or Icon Animation.

Azure cost

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

The times presented in the following table correspond to the times in the tables in the performance results section earlier in this article. You can use these times together with the Azure hourly rates for NVadsA10_v5 and NCasT4_v3 series VMs to calculate costs. For the current hourly costs, see [Windows Virtual Machine Pricing](#).

Only model running time (CPU time) is considered for these cost calculations. Application installation time isn't considered. The costs are indicative. The actual costs depend on the size of the model.

Because the simulation is set to run for one hour, the runtime is in seconds, so the cost estimations shown here are negligible.

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

VM size	GPU partition	CPU time (seconds)
NVadsA10_v5	1/6 GPU	2.05
NVadsA10_v5	1/6 GPU	14.31
NVadsA10_v5	1/6 GPU	2.08
NCasT4_v3	1 GPU	43.22
NCasT4_v3	1 GPU	2.91
NCasT4_v3	1 GPU	62.81
NCasT4_v3	1 GPU	2.94

Summary

- Tecnomatix was successfully tested on NVadsA10_v5 and NCasT4_v3 series VMs on Azure.
- Performance on NVadsA10_v5 is better, with a good FPS rate and less CPU time.
- Tecnomatix on Azure can easily handle typical workloads.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- GPU-optimized virtual machine sizes
- Virtual machines on Azure
- Virtual networks and virtual machines on Azure
- Learning path: Run high-performance computing (HPC) applications on Azure

Related resources

- [Run a Windows VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy tNavigator on a virtual machine

Azure Virtual Machines

Azure Virtual Network

Azure CycleCloud

This article briefly describes the steps for running Rock Flow Dynamics [tNavigator](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running tNavigator.

tNavigator is a reservoir modeling and simulation platform. It provides tools for geoscience, reservoir, and production engineering. It builds static and dynamic reservoir models and runs dynamic simulations. tNavigator runs on workstations and clusters. A cloud-based solution with full GUI capabilities via remote desktop is also available.

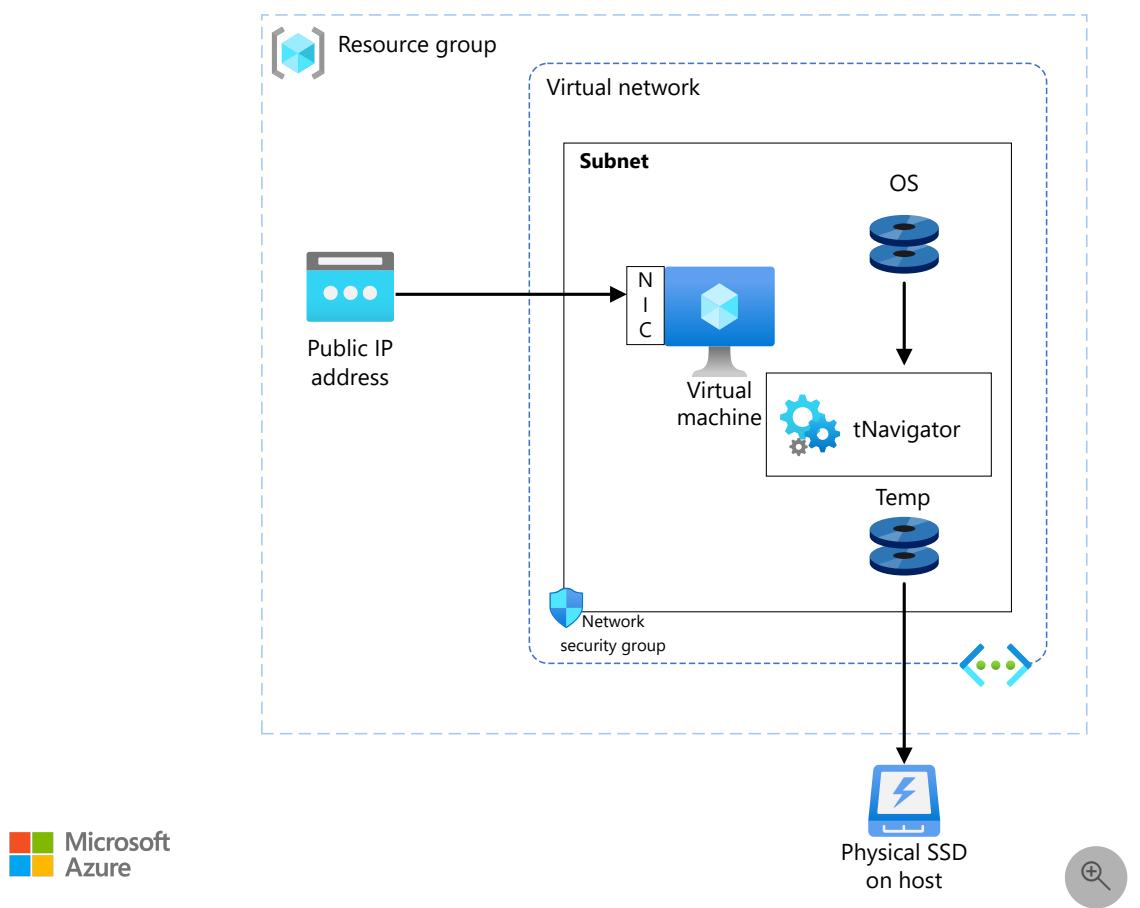
With tNavigator, you can perform extended uncertainty analysis and surface networks builds as part of one integrated workflow. All the parts of the workflow share an internal data storage system, scalable parallel numerical engine, data input and output mechanism, and graphical user interface. tNavigator supports the metric, lab, and field unit systems.

Why deploy tNavigator on Azure?

- Modern and diverse compute options to align to your workload's needs
- Flexible virtualization without the purchase of physical hardware
- Rapid provisioning
- Complex simulations solved in a few hours via an increase in the number of CPU cores

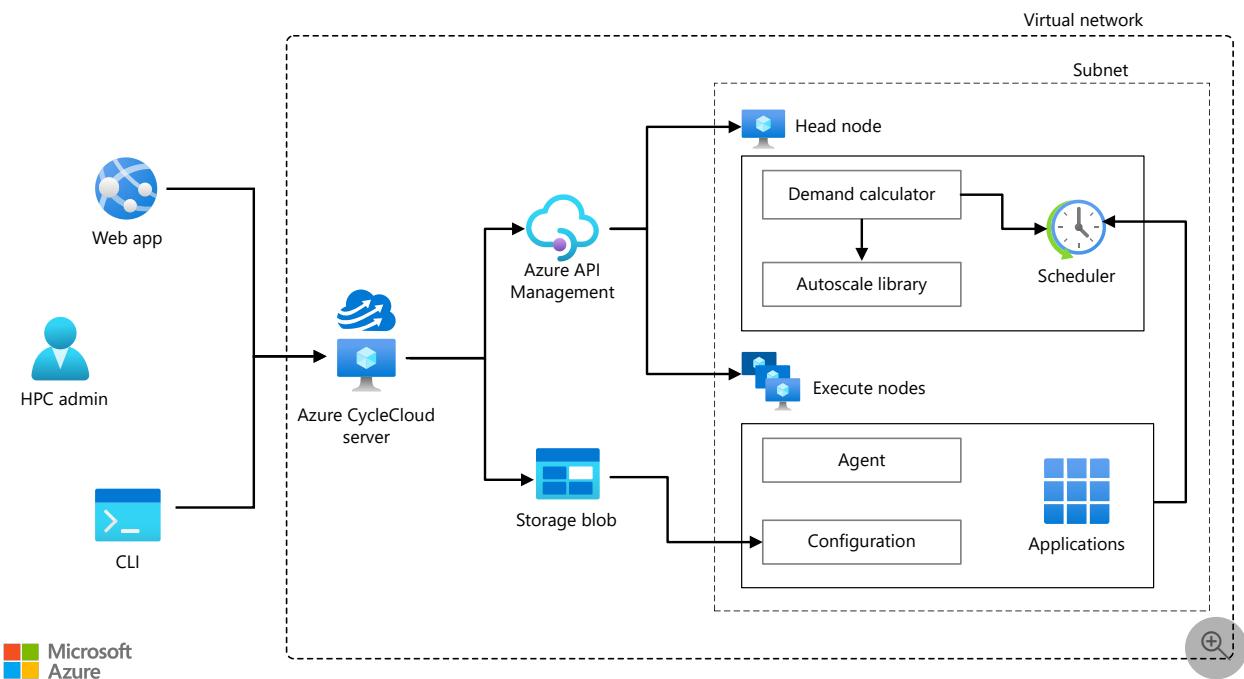
Architecture

This diagram shows a single-node configuration:



[Download a Visio file](#) of this architecture.

This diagram shows a multi-node configuration:



[Download a Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create Linux VMs. For information about deploying the VMs and installing the drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VMs.
 - A public IP address connects the internet to the VMs.
- [Azure CycleCloud](#) is used to create the cluster in the multi-node configuration.
- A physical SSD is used for storage.

Compute sizing

[HBv3-series](#) VMs running on the Linux OS were used to test the performance of tNavigator on Azure. The following table provides configuration details:

[Expand table](#)

VM size	vCPU	Memory (GiB)	Memory bandwidth (GBps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)
Standard_HB120-16rs_v3	16	448	350	1.9	3.0	3.5	200
Standard_HB120-32rs_v3	32	448	350	1.9	3.0	3.5	200
Standard_HB120-64rs_v3	64	448	350	1.9	3.0	3.5	200
Standard_HB120-96rs_v3	96	448	350	1.9	3.0	3.5	200
Standard_HB120-120rs_v3	120	448	350	1.9	3.0	3.5	200

The following table provides details about the operating system used in these tests:

[Expand table](#)

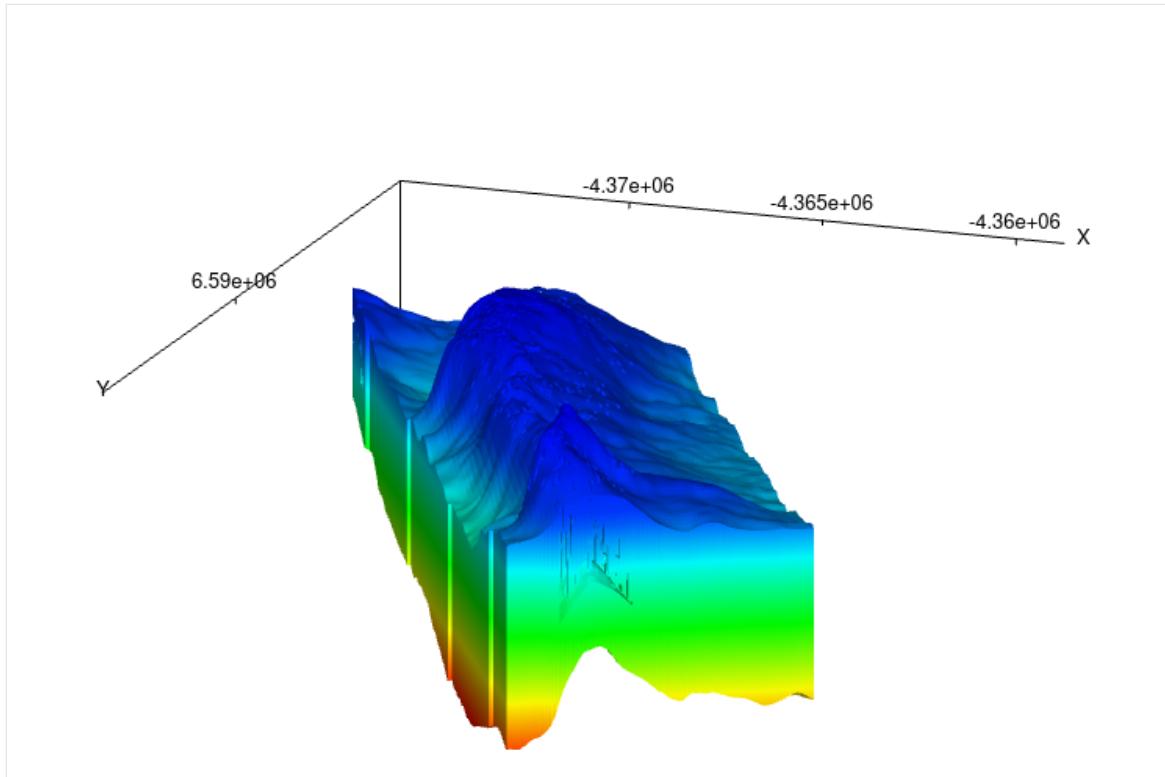
Operating system version	OS architecture
Linux CentOS 8.1 HPC	x86-64

tNavigator installation

Before you install tNavigator, you need to deploy and connect a VM. For information about deploying the VM and installing the drivers, see [Run a Linux VM on Azure](#). You can download and install tNavigator from the [Rock Flow Dynamics Resources Hub](#). You can also get information about the installation process from this hub.

tNavigator performance results

Two standard, stable models were used to test tNavigator: speed test and speed test 9. Details about the models are shown in the following tables.



These are the details for the speed test model:

[Expand table](#)

Dimensions	Total active grid blocks	Total pore volume	Mesh connection statistics
X: 88 Y: 215 Z: 177 Size: 3,348,840	2,418,989	13,563,305,518.45987 rm3	Total: 7,052,853 Geometrical: 7,052,853

These are the details for the speed test 9 model:

[Expand table](#)

Dimensions	Total active grid blocks	Total pore volume	Mesh connection statistics
X: 264 Y: 645 Z: 177 Size: 30,139,560	21,770,901	13,563,305,518.45987 rm3	Total: 64,533,441 Geometrical: 64,533,441

Results on a single node

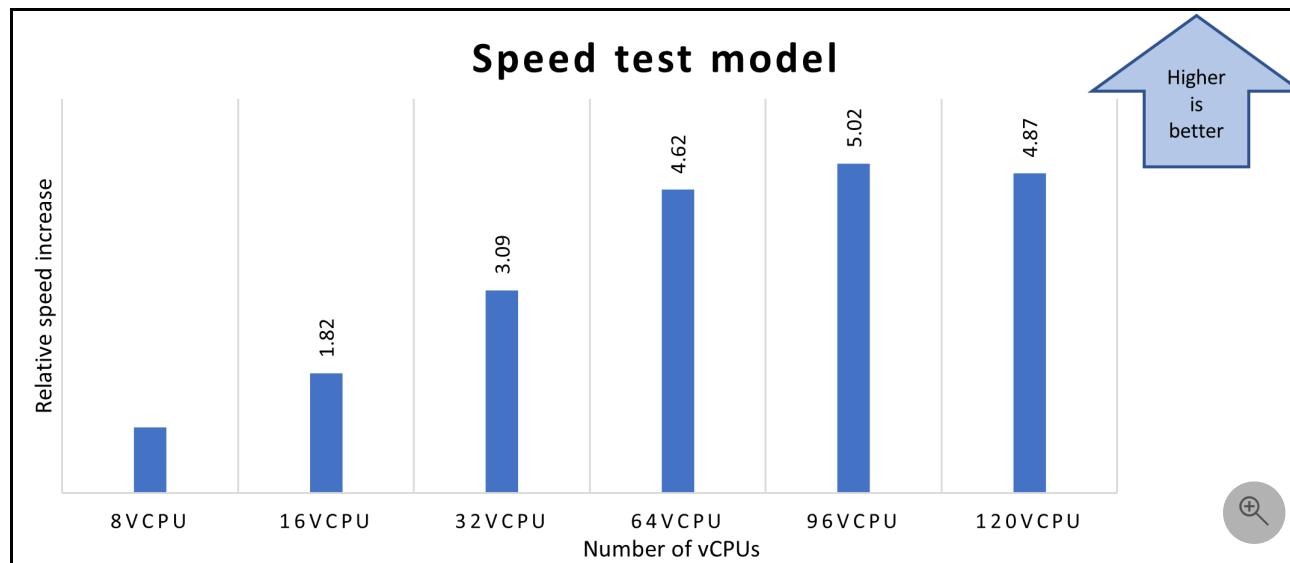
The following sections provide the performance results of running tNavigator on single-node Azure [HBv3 AMD EPYC 7V73X](#) VMs.

Speed test model

[Expand table](#)

VM size	Number of vCPUs used	Total elapsed time (seconds)	Relative speed increase
Standard_HB120-16rs_v3	8	643	N/A
Standard_HB120-16rs_v3	16	352	1.82
Standard_HB120-32rs_v3	32	208	3.09
Standard_HB120-64rs_v3	64	139	4.62
Standard_HB120-96rs_v3	96	128	5.05
Standard_HB120-120rs_v3	120	132	4.87

The following chart shows relative speed increases as the number of vCPUs increases:



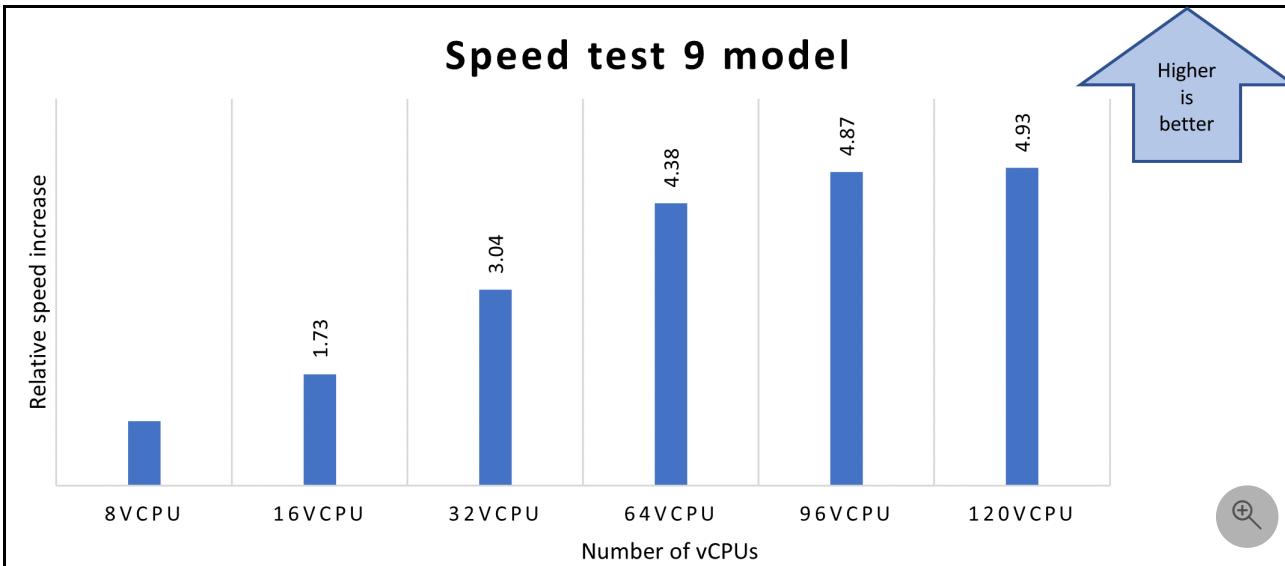
Speed test 9 model

[Expand table](#)

VM size	Number of vCPUs used	Total elapsed time (seconds)	Relative speed increase
Standard_HB120-16rs_v3	8	20,045	N/A
Standard_HB120-16rs_v3	16	11,541	1.73

VM size	Number of vCPUs used	Total elapsed time (seconds)	Relative speed increase
Standard_HB120-32rs_v3	32	6,588	3.04
Standard_HB120-64rs_v3	64	4,572	4.38
Standard_HB120-96rs_v3	96	4,113	4.87
Standard_HB120-120rs_v3	120	4,061	4.93

The following chart shows relative speed increases as the number of vCPUs increases:



Notes about the single-node tests

For all single-node tests, the slower time on a Standard_HB120-16rs_v3 VM with 8 cores is used as a reference to calculate the relative speed increases with respect to similar VMs that have more cores. The results presented previously show near linear computation performance improvements as the number of cores increases from 8 to 120, until the optimal configuration for a given model is reached.

Results in a multi-node configuration

The following sections provide the performance results of running tNavigator on multi-node Azure HBv3-series VMs. The speed test model isn't suitable for testing in a multi-node environment, so tests are restricted to the speed test 9 model.

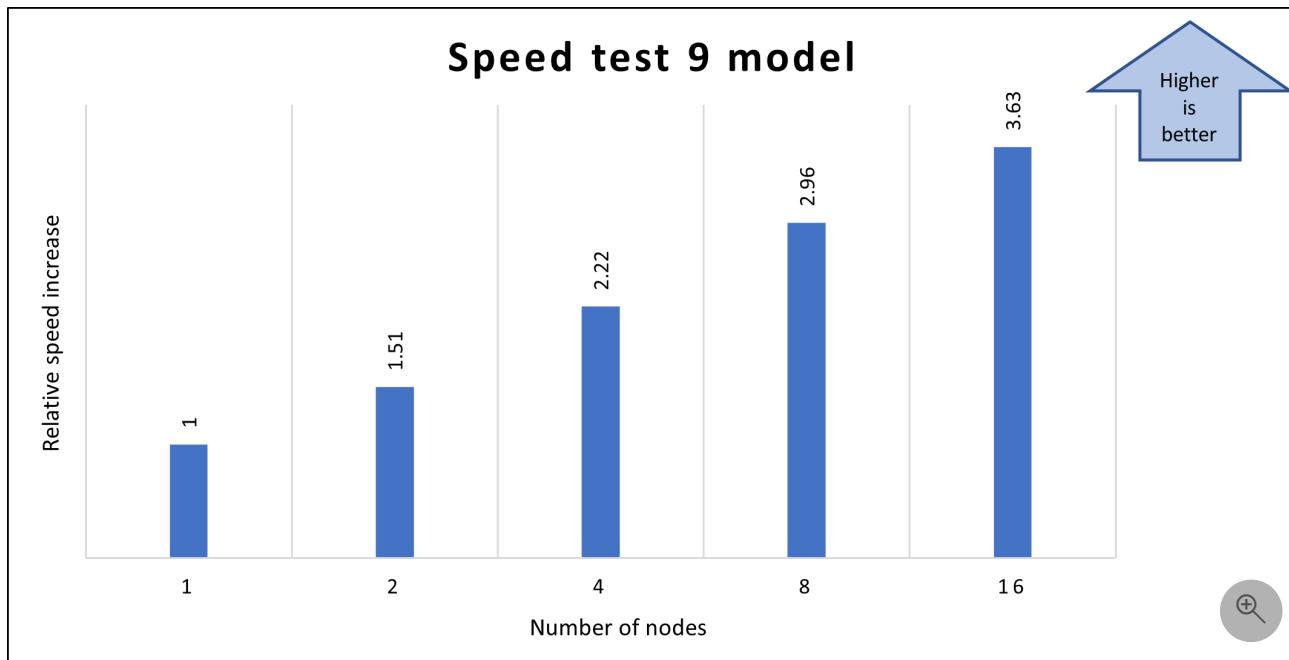
This table shows the times recorded for varying numbers of nodes of the Standard_HB120-64rs_v3 VM on Azure CycleCloud:

[Expand table](#)

VM size	Number of nodes	Number of vCPUs	Total elapsed time (seconds)	Relative speed increase
Standard_HB120-64rs_v3	1	64	5,025	N/A

VM size	Number of nodes	Number of vCPUs	Total elapsed time (seconds)	Relative speed increase
Standard_HB120-64rs_v3	2	128	3,323	1.51
Standard_HB120-64rs_v3	4	256	2,264	2.22
Standard_HB120-64rs_v3	8	512	1,697	2.96
Standard_HB120-64rs_v3	16	1024	1,383	3.63

The following graph shows the relative speed increases as the number of nodes increases:



Notes about the multi-node tests

The multi-node results show that models scale linearly until the 16-node configuration is reached. This configuration provides a maximum speed increase of 3.63 times that of the single node. Testing was limited to a few iterations.

Azure cost

Only model running time (elapsed time) is considered for these cost calculations. Application installation time isn't considered. The calculations are indicative of your potential results. The actual cost depends on the size of the model. You can use the [Azure pricing calculator](#) to estimate costs for your configuration.

The following tables provide elapsed times in hours. To compute the total cost, multiply by the [Azure VM hourly costs for Linux](#).

Elapsed times for the speed test model, single-node

[Expand table](#)

Number of vCPUs	Total elapsed time (hours)
8	0.17
16	0.09
32	0.05
64	0.03
96	0.03
120	0.03

Elapsed times for the speed test 9 model, single-node

[Expand table](#)

Number of vCPUs	Total elapsed time (hours)
8	5.56
16	3.20
32	1.83
64	1.27
96	1.14
120	1.12

Elapsed times for the speed test model, multi-node

[Expand table](#)

Number of nodes*	Total elapsed time (hours)
1	1.40
2	0.92
4	0.63

Number of nodes*	Total elapsed time (hours)
8	0.47
16	0.38

*64 cores per node.

Summary

- tNavigator exhibits high scalability when deployed on HBv3-series VMs (AMD EPYC 7V73X Milan-X CPU cores) on Azure. To determine the scalability, the ratio of the inverse of the time it takes to solve a given model is calculated.
- To evaluate the performance, the lowest VM configuration in the series, 8 vCPUs for HBv3, is used as a baseline. The results are assessed based on the performance relative to this baseline. Higher values indicate better performance.
- For the single-node configuration on HBv3, the solution finishes approximately 1.5 times faster whenever the number of cores is doubled. Scale-up decreases as the optimal configuration is reached. Increasing the number of cores beyond the optimal configuration doesn't result in improved performance.
- For the multi-mode configuration on HBv3, implemented via CycleCloud, the solution finishes approximately 1.3 to 1.5 times faster whenever the number of nodes is doubled. Scale-up decreases as the optimal configuration is reached.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Aashay Anjankar](#) | HPC Performance Engineer
- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Saurabh Parave](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director of Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- Run a Linux VM on Azure
- HPC system and big-compute solutions
- HPC cluster deployed in the cloud

Deploy Turbostream on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [Turbostream](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running TurboStream on Azure.

Turbostream is advanced simulation software that's based on a computational fluid dynamics (CFD) solver. It can run on high-speed GPUs and on conventional CPUs.

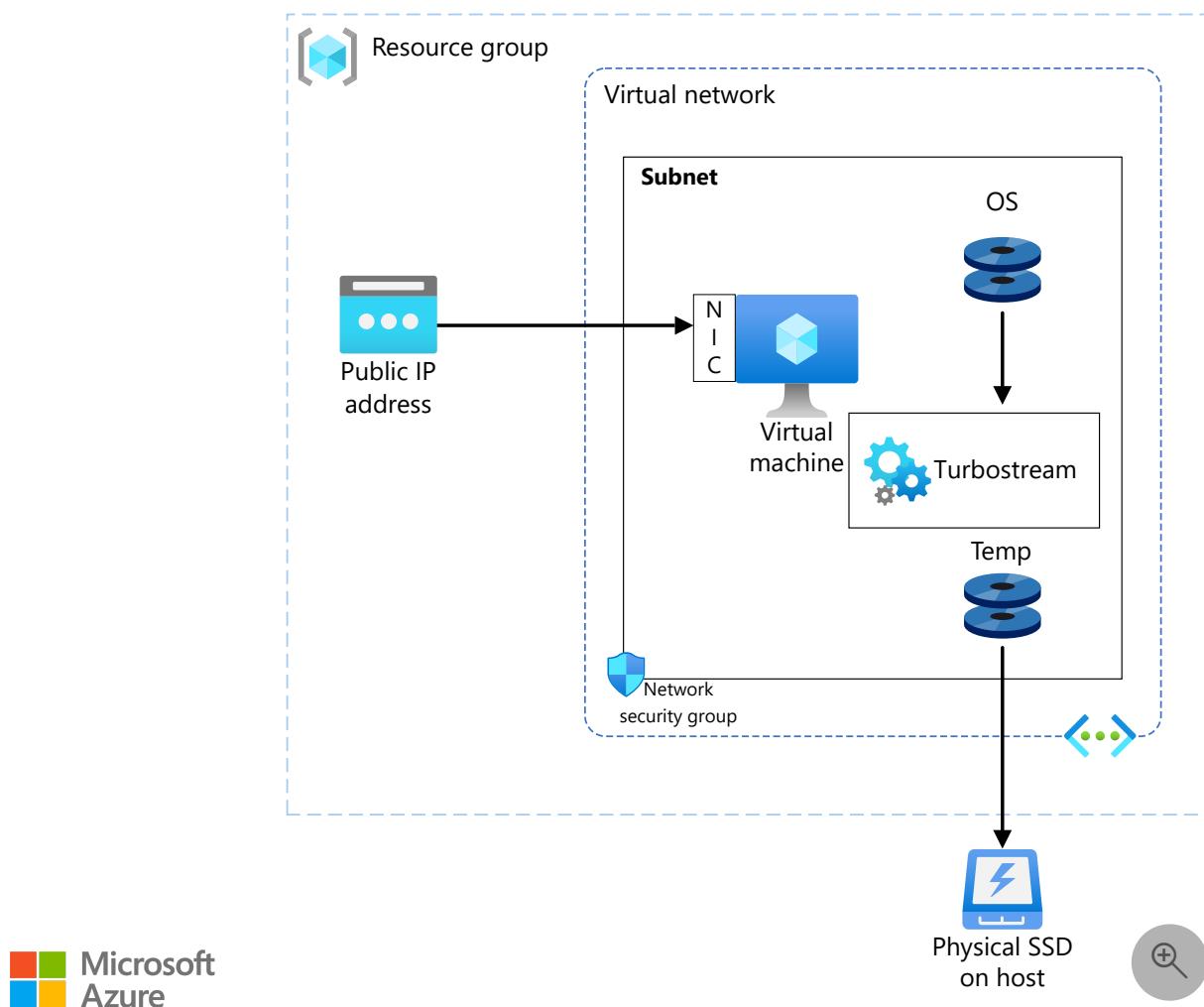
The software enables high-fidelity methods, like unsteady full-annulus simulations, to be used as part of the routine design process.

Turbostream is used by NASA and in the design of aircraft engines, turbomachinery, and gas turbines.

Why deploy Turbostream on Azure?

- Modern and diverse compute options to meet the needs of your workloads
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- With an eight-GPU configuration, a performance increase of 4.51 times that of a single GPU

Architecture



[Download a Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Linux VM. For information about deploying the VM and installing the drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

The performance tests of Turbostream used an [ND_A100_v4](#) VM running Linux. The following table provides details about the VM.

[Expand table](#)

VM size	vCPU	Memory (GiB)	SSD (GiB)	GPUs	GPU memory (GiB)	Maximum data disks
Standard_ND96asr_v4	96	900	6,000	8 A100	40	32

Required drivers

To take advantage of the GPU capabilities of [ND_A100_v4](#) VMs, you need to install NVIDIA GPU drivers.

To use AMD processors on [ND_A100_v4](#) VMs, you need to install AMD drivers.

Turbostream installation

Before you install Turbostream, you need to deploy and connect a Linux VM and install the required NVIDIA and AMD drivers.

i **Important**

NVIDIA Fabric Manager installation is required for VMs that use NVLink or NVSwitch. [ND_A100_v4](#) VMs use NVLink.

For information about deploying the VM and installing the drivers, see [Run a Linux VM on Azure](#).

You can install Turbostream by signing in to [ExaVault](#) as a customer. The downloadable files include documentation, a release package, a license file, and a test simulation package (*scaling_test.zip*). For more information, contact [Turbostream](#).

Turbostream performance results

Four simulations were tested, as described in the following table.

i [Expand table](#)

Model number	Number of grid nodes (millions)
1	6

Model number	Number of grid nodes (millions)
2	12
3	24
4	48

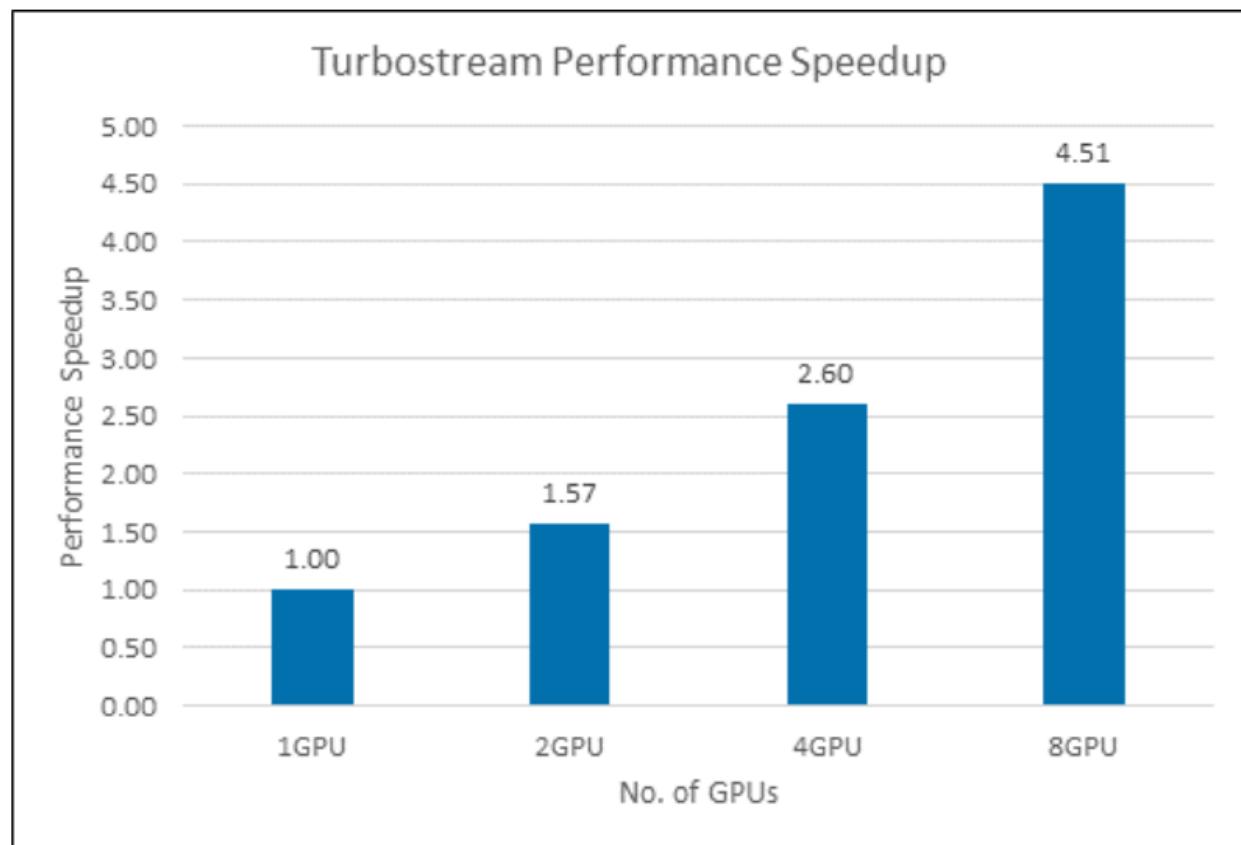
The following table describes the performance results. *Performance* is the number of grid nodes processed per second. *Relative performance* is relative to the performance described in the first line of the table.

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

Model number	Number of GPUs	Time (seconds, average of 200 iterations)	Performance	Relative performance
1	1*	0.0743	80,753,701.21	1
2	2*	0.0944	127,118,644.1	1.57
3	4*	0.1145	209,606,986.9	2.60
4	8	0.1319	363,912,054.6	4.51

* In these cases, the number of GPUs was artificially limited. The Standard_ND96asr_v4 VM has eight GPUs.

The relative performance increases are presented graphically here:



Additional notes about tests

The following table provides details about the operating system and the NVIDIA drivers that were used for testing.

[Expand table](#)

OS version	OS architecture	GPU driver version	CUDA version
CentOS Linux release 8.1.1911 (Core)	x86-64	470.57.02	11.4

Azure cost

The following table presents the elapsed time. You can use this time and the Azure VM hourly cost for the NDA100v4 VM to calculate costs. For the current hourly cost, see [Linux Virtual Machines Pricing](#).

Only simulation runtime is included in the reported time. Application installation time isn't included.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

VM size	GPUs	Elapsed time (seconds)
Standard_ND96asr_v4	8 A100	196.10

Summary

- Turbostream was successfully tested on the ND_A100_v4 VM.
- Performance with eight GPUs is 4.51 times faster than the performance with one GPU.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy Visiopharm on a virtual machine

Azure Virtual Machines

Azure Virtual Network

This article briefly describes the steps for running [Visiopharm](#) on a virtual machine (VM) that's deployed on Azure. It also presents the performance results of running Visiopharm on Azure.

Visiopharm is an AI-based image analysis and tissue mining tool that supports drug development research and other research.

Visionpharm:

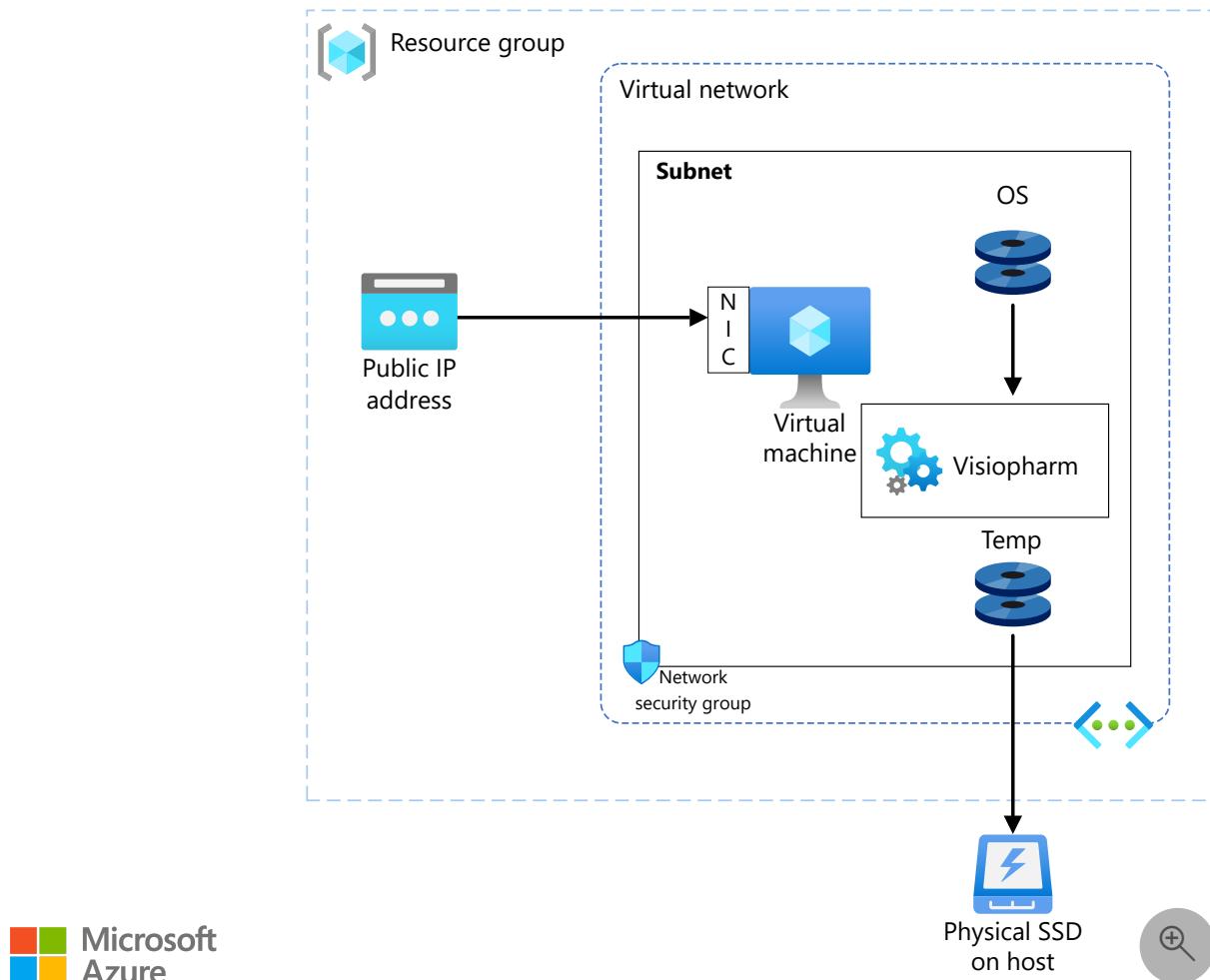
- Enables researchers to align and subsequently analyze digitized serial sections.
- Enables tissue researchers to analyze both simple and complex datasets to generate reliable quantitative results.
- Uses pre-trained nuclei segmentation APPs, suitable for bright-field and fluorescence applications.

Visiopharm is used in academic institutions, the biopharmaceutical industry, and diagnostic centers. It's ideal for the education, healthcare, and manufacturing industries.

Why deploy Visiopharm on Azure?

- Modern and diverse compute options to meet your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning

Architecture



[Download a Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Windows VM. For information about deploying the VM and installing the drivers, see [Windows VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.
- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

The performance tests of Visiopharm used [Standard_NC24s_v3](#) and [Standard_NC16as_T4_v3](#) VMs running Windows. The following table provides details about the VMs.

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

VM	vCPU	Memory (GiB)	SSD (GiB)	GPU	GPU memory (GiB)	Maximum data disks
Standard_NC24s_v3	24	448	2,948	4 V100	64	32
Standard_NC16as_T4_v3	16	110	360	1 T4	16	32

Required drivers

To take advantage of the GPU capabilities of [Standard_NC24s_v3](#) and [Standard_NC16as_T4_v3](#) VMs, you need to install NVIDIA GPU drivers.

To use AMD processors on [Standard_NC16as_T4_v3](#) VMs, you need to install AMD drivers.

Visiopharm installation

Before you install Visiopharm, you need to deploy and connect a VM, install an eligible Windows 10 or Windows 11 image, and install NVIDIA and AMD drivers, as needed.

For information about eligible Windows images, see [How to deploy Windows 10 on Azure](#) and [Use Windows client in Azure for dev/test scenarios](#).

For information about deploying the VM and installing the drivers, see [Run a Windows VM on Azure](#).

For information about installing the software, contact [Visiopharm](#).

Visiopharm performance results

To test the performance of Visiopharm, image analysis was performed on Standard_NC24s_v3 and Standard_NC16as_T4_v3 VMs, and the performance was compared. Visiopharm version 2022.03 was used for testing.

Three Visiopharm solutions (APPs) were run:

- APP1: Tissue detection
- APP2: Segmentation
- APP3: Cell detection AI

The image is called LuCa 6Plex. It's a set of pathology data that's provided by Visiopharm. The three APPs predominantly use the GPU capabilities of the VMs to run analyses.

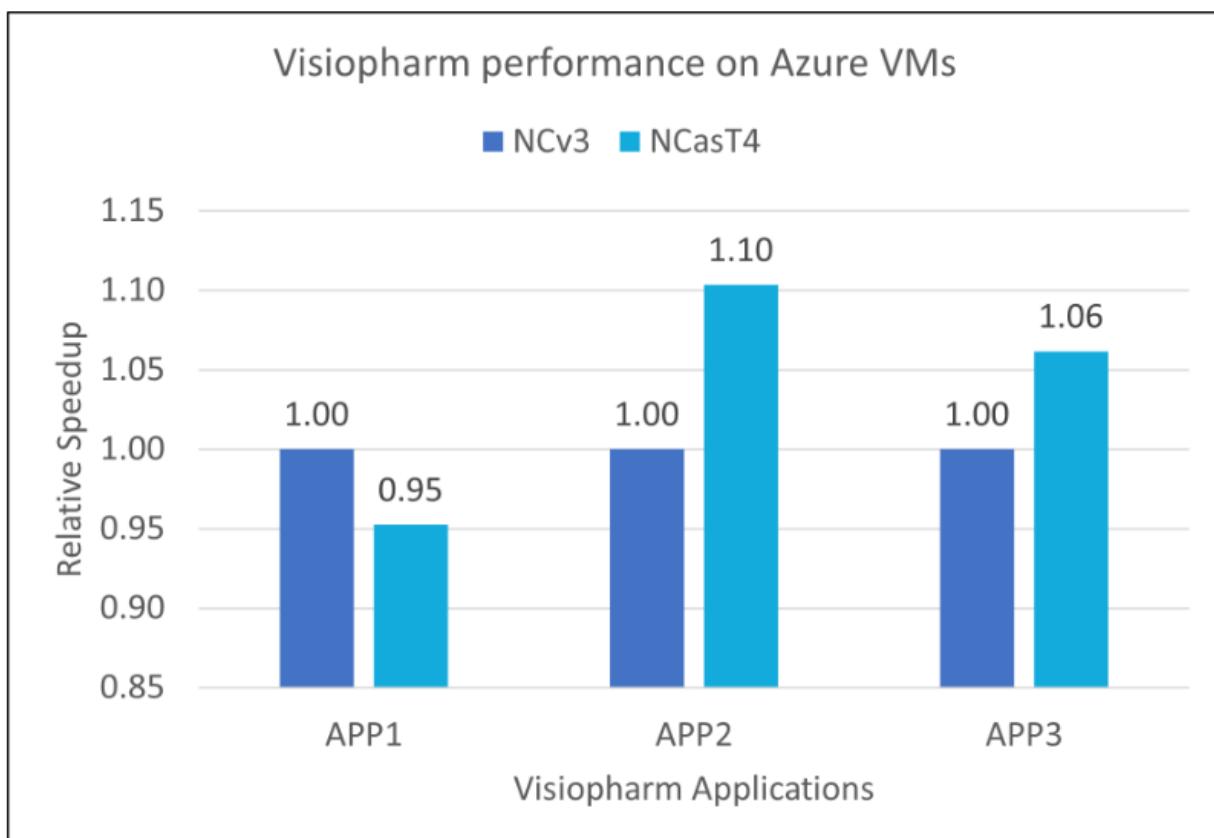
The following table shows the results.

[Expand table](#)

VM	GPU	APP1 elapsed time	APP2 elapsed time	APP3 elapsed time
Standard_NC24s_v3*	V100	00:00:20	03:55:02	00:45:21
Standard_NC16as_T4_v3	Tesla T4	00:00:21	03:33:01	00:42:43

* This test was performed with an artificially limited one-GPU configuration. This VM has four GPUs.

The following graph shows the relative performance of the two VMs. Note that this comparison uses a one-GPU configuration for both VMs. The Standard_NC24s_v3 has four GPUs, although the NCv3 series provides options with one, two, and four GPUs.



Azure cost

The following table presents the wall-clock times for running the analyses. You can use these times and the Azure VM hourly costs for NCas_T4_v3 series VMs to compute costs. For the

current hourly costs, see [Windows Virtual Machines Pricing](#).

Only analysis time is considered for these calculations. Application installation time isn't included.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

 [Expand table](#)

VM	APP1: Tissue detection	APP2: Segmentation	APP3: Cell detection AI
Standard_NC16as_T4_v3	21 seconds	3 hours, 33 minutes	42 minutes, 43 seconds

Summary

- Visiopharm was successfully tested on Standard_NC24s_v3 and Standard_NC16as_T4_v3 VMs.
- Based on a one-GPU configuration for both VMs, Standard_NC16as_T4_v3 performs better.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- [GPU-optimized virtual machine sizes](#)

- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Windows VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

Deploy WRF on a virtual machine

Azure Virtual Machines Azure Virtual Network

This article briefly describes the steps for running [Weather Research & Forecasting \(WRF\)](#) on a virtual machine (VM) deployed on Azure. It also presents the performance results of running WRF on Azure.

WRF is a mesoscale numerical weather-prediction system designed for atmospheric research and operational forecasting applications.

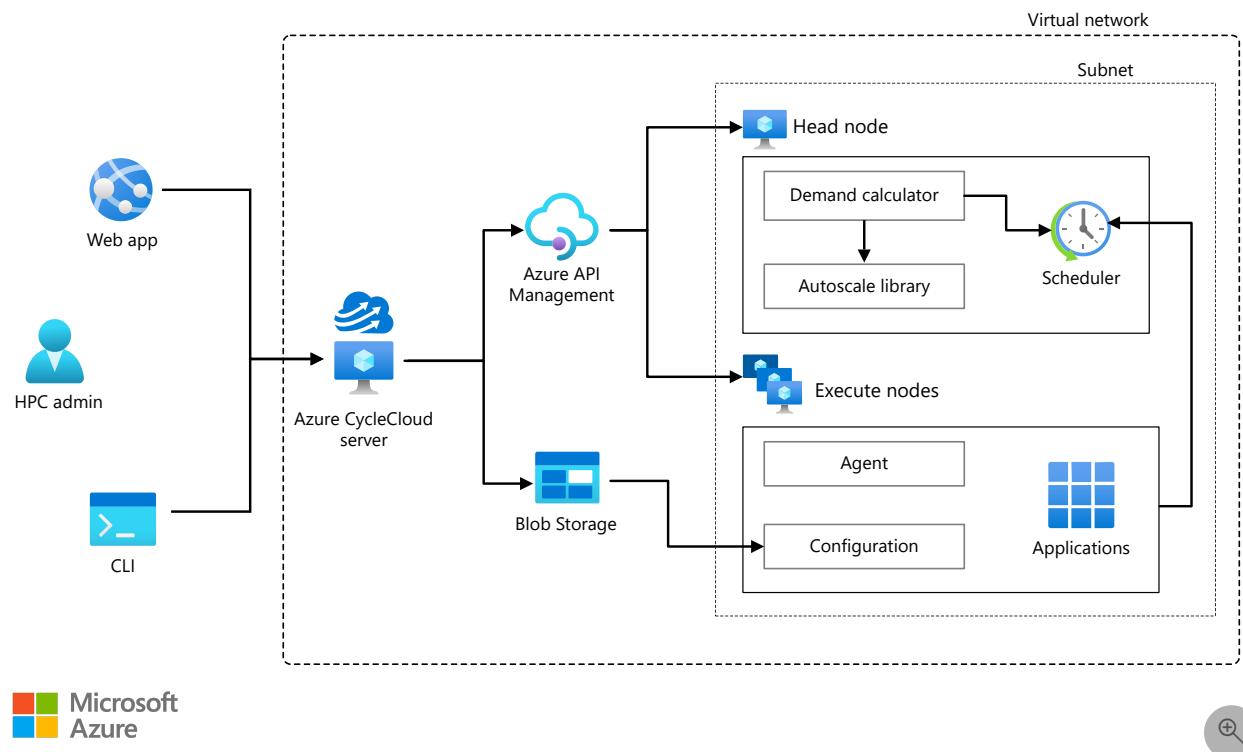
WRF serves a wide range of meteorological applications across scales from tens of meters to thousands of kilometers. It provides a flexible and computationally efficient platform while reflecting recent advances in physics, numerics, and data assimilation. The software can produce simulations based on actual atmospheric conditions or from idealized conditions.

WRF is used by academic atmospheric scientists (dynamics, physics, weather, and climate research), forecast teams at operational centers, and applications scientists (air quality, hydrology, and utilities).

Why deploy WRF on Azure?

- Modern and diverse compute options to meet your workload's needs
- The flexibility of virtualization without the need to buy and maintain physical hardware
- Rapid provisioning
- Performance that scales as CPUs are added, based on tests of a sample model

Architecture



Download a [Visio file](#) of this architecture.

Components

- [Azure Virtual Machines](#) is used to create a Linux VM. For information about deploying the VM and installing the drivers, see [Linux VMs on Azure](#).
- [Azure Virtual Network](#) is used to create a private network infrastructure in the cloud.
 - [Network security groups](#) are used to restrict access to the VM.
 - A public IP address connects the internet to the VM.

- A physical solid-state drive (SSD) is used for storage.

Compute sizing and drivers

The performance tests of WRF used [HBv3-series](#) VMs running Linux. The following table provides details about the VMs.

[Expand table](#)

Size	vCPU	RAM memory (GiB)	Memory bandwidth (Gbps)	Base CPU frequency (GHz)	All-cores frequency (GHz, peak)	Single-core frequency (GHz, peak)	RDMA performance (Gbps)	Maximum data disks
Standard_HB120rs_v3	120	448	350	1.9	3.0	3.5	200	32
Standard_HB120-96rs_v3	96	448	350	1.9	3.0	3.5	200	32
Standard_HB120-64rs_v3	64	448	350	1.9	3.0	3.5	200	32
Standard_HB120-32rs_v3	32	448	350	1.9	3.0	3.5	200	32
Standard_HB120-16rs_v3	16	448	350	1.9	3.0	3.5	200	32

HBv3-series VMs are optimized for HPC applications, such as:

- Fluid dynamics
- Explicit and implicit finite element analysis
- Weather modeling
- Seismic processing
- Reservoir simulation
- RTL simulation

WRF installation

Before you install WRF, you must deploy and connect to a VM or HPC Cluster.

For information about deploying the VM and installing the drivers, see [Run a Linux VM on Azure](#). For deploying Azure CycleCloud and HPC cluster, see these articles:

- [Install and configure Azure CycleCloud](#)
- [Create a HPC Cluster](#)

Download and compile WRF

1. Open the [WRF Portal](#) in a web browser and download the WRF source code.
2. Configure and compile WRF. For detailed compiling information, follow the steps at [How to Compile WRF: The Complete Process](#).
3. Configure options for WRF. For the current performance tests, we selected option 16 as shown here:

Please select from among the following Linux x86_64 options:

1. (serial)	2. (smpar)	3. (dmpar)	4. (dm+sm)	PGI (pgf90/gcc)
5. (serial)	6. (smpar)	7. (dmpar)	8. (dm+sm)	PGI (pgf90/pgcc): SGI MPT
9. (serial)	10. (smpar)	11. (dmpar)	12. (dm+sm)	PGI (pgf90/gcc): PGI accelerator
13. (serial)	14. (smpar)	15. (dmpar)	16. (dm+sm)	INTEL (ifort/icc)
			17. (dm+sm)	INTEL (ifort/icc): Xeon Phi (MIC architecture)
18. (serial)	19. (smpar)	20. (dmpar)	21. (dm+sm)	INTEL (ifort/icc): Xeon (SNB with AVX mods)
22. (serial)	23. (smpar)	24. (dmpar)	25. (dm+sm)	INTEL (ifort/icc): SGI MPT
26. (serial)	27. (smpar)	28. (dmpar)	29. (dm+sm)	INTEL (ifort/icc): IBM POE
30. (serial)		31. (dmpar)		PATHSCALE (pathf90/pathcc)
32. (serial)	33. (smpar)	34. (dmpar)	35. (dm+sm)	GNU (gfortran/gcc)
36. (serial)	37. (smpar)	38. (dmpar)	39. (dm+sm)	IBM (xlf90_r/cc_r)
40. (serial)	41. (smpar)	42. (dmpar)	43. (dm+sm)	PGI (ftn/gcc): Cray XC CLE
44. (serial)	45. (smpar)	46. (dmpar)	47. (dm+sm)	CRAY CCE (ftn \$(_NOOMP)/cc): Cray XE and XC
48. (serial)	49. (smpar)	50. (dmpar)	51. (dm+sm)	INTEL (ftn/icc): Cray XC
52. (serial)	53. (smpar)	54. (dmpar)	55. (dm+sm)	PGI (pgf90/pgcc)
56. (serial)	57. (smpar)	58. (dmpar)	59. (dm+sm)	PGI (pgf90/gcc): -f90=pgf90
60. (serial)	61. (smpar)	62. (dmpar)	63. (dm+sm)	PGI (pgf90/pgcc): -f90=pgf90
64. (serial)	65. (smpar)	66. (dmpar)	67. (dm+sm)	INTEL (ifort/icc): HSW/BDW
68. (serial)	69. (smpar)	70. (dmpar)	71. (dm+sm)	INTEL (ifort/icc): KNL MIC
72. (serial)	73. (smpar)	74. (dmpar)	75. (dm+sm)	FUJITSU (frtpx/fccpx): FX10/FX100 SPARC64 IXfx/XL

Enter selection [1-75] : 16

4. Download static geographic data (for *geogrid.exe*) from the [WPS V4 Geographical Static Data Downloads Page](#).
5. Download real-time data from the [NOMADS Data at NCEP](#). For real-time cases, the WRF model requires up-to-date meteorological information for both an initial condition and also for lateral boundary conditions. The following data is used for the current performance test:
 - **Date:** 12-February-2023
 - **Files:**
 - gfs.0p25.2023021200.f000.grib2
 - gfs.0p25.2023021200.f003.grib2
 - gfs.0p25.2023021200.f006.grib2
 - gfs.0p25.2023021200.f009.grib2
 - gfs.0p25.2023021200.f384.grib2
6. Run WPS, create an `met_em.*` file for more than one time period, and link or copy WPS output files to the WRF run directory. To run WPS, follow the steps at [How to Compile WRF: The Complete Process](#).
7. Configure options for WRF. For the current performance test, option 19 was selected as shown here:

Please select from among the following supported platforms.

1. Linux x86_64, gfortran (serial)
2. Linux x86_64, gfortran (serial_NO_GRIB2)
3. Linux x86_64, gfortran (dmpar)
4. Linux x86_64, gfortran (dmpar_NO_GRIB2)
5. Linux x86_64, PGI compiler (serial)
6. Linux x86_64, PGI compiler (serial_NO_GRIB2)
7. Linux x86_64, PGI compiler (dmpar)
8. Linux x86_64, PGI compiler (dmpar_NO_GRIB2)
9. Linux x86_64, PGI compiler, SGI MPT (serial)
10. Linux x86_64, PGI compiler, SGI MPT (serial_NO_GRIB2)
11. Linux x86_64, PGI compiler, SGI MPT (dmpar)
12. Linux x86_64, PGI compiler, SGI MPT (dmpar_NO_GRIB2)
13. Linux x86_64, IA64 and Opteron (serial)
14. Linux x86_64, IA64 and Opteron (serial_NO_GRIB2)
15. Linux x86_64, IA64 and Opteron (dmpar)
16. Linux x86_64, IA64 and Opteron (dmpar_NO_GRIB2)
17. Linux x86_64, Intel compiler (serial)
18. Linux x86_64, Intel compiler (serial_NO_GRIB2)
19. Linux x86_64, Intel compiler (dmpar)
20. Linux x86_64, Intel compiler (dmpar_NO_GRIB2)
21. Linux x86_64, Intel compiler, SGI MPT (serial)
22. Linux x86_64, Intel compiler, SGI MPT (serial_NO_GRIB2)
23. Linux x86_64, Intel compiler, SGI MPT (dmpar)
24. Linux x86_64, Intel compiler, SGI MPT (dmpar_NO_GRIB2)
25. Linux x86_64, Intel compiler, IBM POE (serial)
26. Linux x86_64, Intel compiler, IBM POE (serial_NO_GRIB2)
27. Linux x86_64, Intel compiler, IBM POE (dmpar)
28. Linux x86_64, Intel compiler, IBM POE (dmpar_NO_GRIB2)
29. Linux x86_64 g95 compiler (serial)
30. Linux x86_64 g95 compiler (serial_NO_GRIB2)
31. Linux x86_64 g95 compiler (dmpar)
32. Linux x86_64 g95 compiler (dmpar_NO_GRIB2)
33. Cray XE/XC CLE/Linux x86_64, Cray compiler (serial)
34. Cray XE/XC CLE/Linux x86_64, Cray compiler (serial_NO_GRIB2)
35. Cray XE/XC CLE/Linux x86_64, Cray compiler (dmpar)
36. Cray XE/XC CLE/Linux x86_64, Cray compiler (dmpar_NO_GRIB2)
37. Cray XC CLE/Linux x86_64, Intel compiler (serial)
38. Cray XC CLE/Linux x86_64, Intel compiler (serial_NO_GRIB2)
39. Cray XC CLE/Linux x86_64, Intel compiler (dmpar)
40. Cray XC CLE/Linux x86_64, Intel compiler (dmpar_NO_GRIB2)

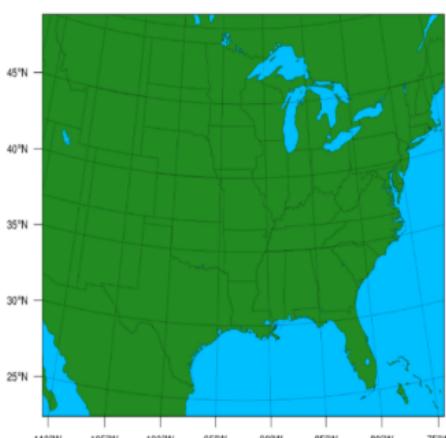
Enter selection [1-40] : 19

8. Edit the *namelist.input* file for runtime options. At minimum, you must edit `&time` control for start, end, and integration times, and `&domains` for grid dimensions.

The WRF model for this example is **CONUS 2.5 km** and is defined by the *namelist.input* file, which includes the model geometric details.

WRF performance results on Azure HPC Cluster

The New CONUS 2.5 km model is used for performance evaluation:



The following table provides details about the model:

[Expand table](#)

Model	Resolution (km)	e_we	e_sn	e_vert	Total grid points	Time step (seconds)	Simulation hours
New CONUS 2.5 km	2.5	1901	1301	35	2,473,201	15	6

The following table shows the system and operating system details:

[Expand table](#)

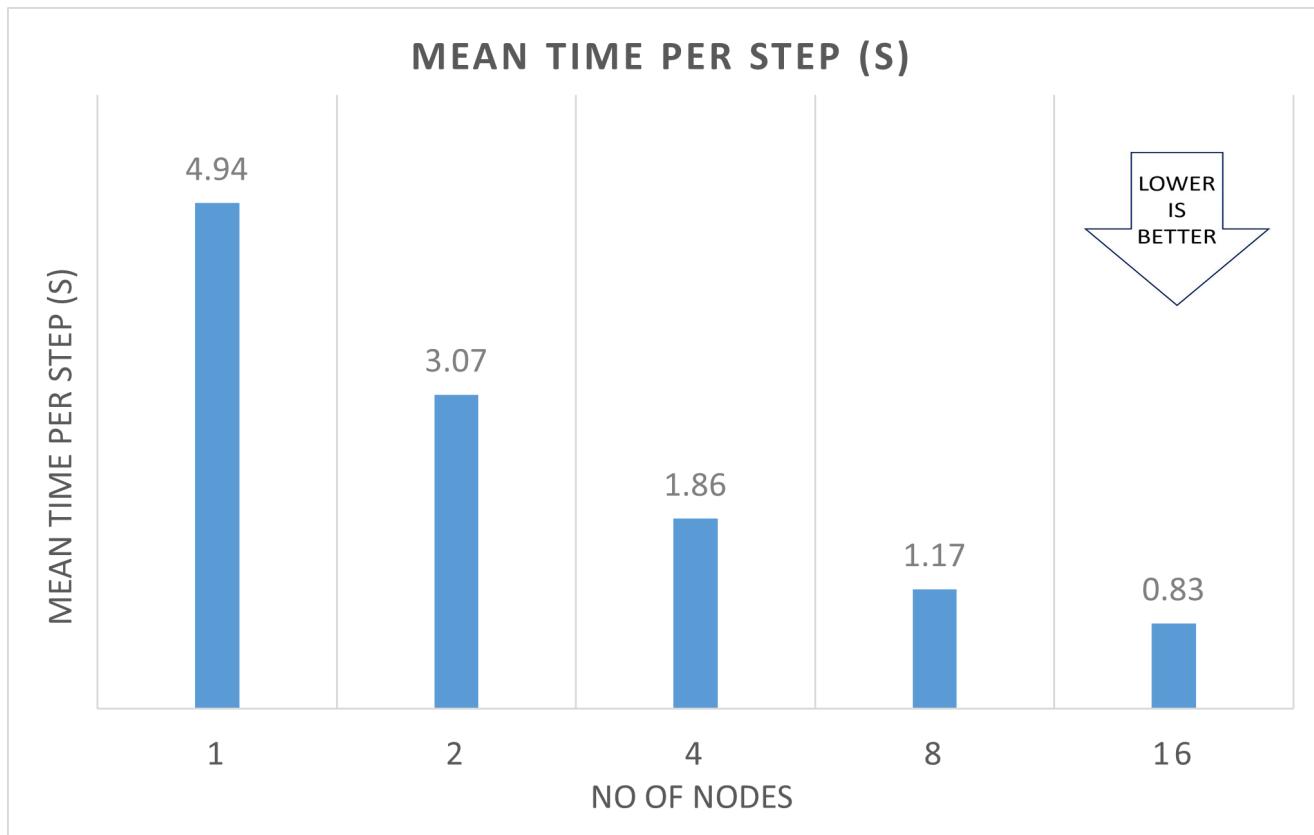
OS/Software	Details
Operating system version	CentOS Linux release 8.1.1911 (Core)
OS Architecture	X86-64
MPI	Open MPI 4.1.0
Compiler	ICC 2021.4.0

Standard_HB120-64rs_v3 VM with 64 vCPUs is considered for the cluster runs. The simulation was run on 1, 2, 4, 8 and 16 nodes and the results are shown in this table:

[Expand table](#)

VM Size	Nodes	vCPU	Tiles	Threads	Simulation time (Hrs)	Mean time per step (s)
Standard_HB120-64rs_v3	1	64	325	1	02:11:30	4.94
Standard_HB120-64rs_v3	2	128	325	1	01:27:00	3.07
Standard_HB120-64rs_v3	4	256	325	1	00:59:01	1.86
Standard_HB120-64rs_v3	8	512	325	1	00:44:36	1.17
Standard_HB120-64rs_v3	16	1024	325	1	00:34:51	0.83

The following graph shows the mean times per step, in seconds:



More notes about tests

1. WRF is successfully deployed and tested on HBv3 AMD EPYC™ 7V73X series VM on Azure Platform.
2. Expected meantime per step is achieved in all CPU cores in multi-node setup.
3. The scalability might vary depending on the dataset being used and the node count being tested. Consider these factors when you test the impact of the tile size, process, and threads per process.

Azure cost

The following table presents the wall-clock times for running the New CONUS 2.5 km simulation. You can multiply these times by the Azure VM hourly costs for HBv3-series VMs to calculate costs. For the current hourly costs, see [Linux Virtual Machines Pricing](#).

Only elapsed solver running time (simulation run time) was considered for these cost calculations. Application installation time isn't considered.

You can use the [Azure pricing calculator](#) to estimate the costs for your configuration.

expand table Expand table

Number of Nodes	Simulation time (hours)
1	02:11:30
2	01:27:00
4	00:59:01
8	00:44:36
16	00:34:51

Summary

- WRF was successfully tested on HBv3-series VMs on Azure.
- Expected mean time per step was achieved with all the nodes tested. However, scalability might vary depending on the dataset used and the node count. Be sure to test the effect of the tile size, process, and threads per process.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [Hari Bagudu](#) | Senior Manager
- [Gauhar Junnarkar](#) | Principal Program Manager
- [Vinod Pamulapati](#) | HPC Performance Engineer
- [Vivi Richard](#) | HPC Performance Engineer

Other contributors:

- [Mick Alberts](#) | Technical Writer
- [Guy Bursell](#) | Director Business Strategy
- [Sachin Rastogi](#) | Manager

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- GPU-optimized virtual machine sizes
- [Virtual machines on Azure](#)
- [Virtual networks and virtual machines on Azure](#)
- [Learning path: Run high-performance computing \(HPC\) applications on Azure](#)

Related resources

- [Run a Linux VM on Azure](#)
- [HPC system and big-compute solutions](#)
- [HPC cluster deployed in the cloud](#)

High-performance computing (HPC) for manufacturing

Azure Batch

Azure Files

Azure Machine Learning

Customers are demanding products that have these attributes: lightweight, strong, safe, sustainable, and customized. As a result, the design stage has become increasingly complex. In that stage, computers are used to visualize, analyze, simulate, and optimize. And those tasks will grow more sophisticated and computationally hungry. In addition, products are increasingly connected, and generate vast amounts of data that needs to be processed and analyzed.

It all adds up to a single need: large computing resources, on-demand.

In this article, we walk through some well-known areas in engineering and manufacturing that need large computing power. We then explore how the Microsoft Azure platform can help.

Cloud design workstations

Product designers use various software tools during the design and planning phases of the product development lifecycle. CAD tools require strong graphics capabilities on the designer's workstation, and the cost of these workstations is high. These souped-up workstations are normally inside the offices of the designers, tying them physically to a location.

As cloud solutions started to gain more popularity, and new capabilities became available, the idea of cloud workstations started to become more viable. Using a workstation that's hosted on the cloud allows the designer to access it from any location. And it allows the organization to change the cost model from capital expenses to operational expenses.

Remote desktop protocol

Microsoft's Remote Desktop Protocol (RDP) has supported TCP-only for a long time. Transmission Control Protocol (TCP) introduces more overhead than User Datagram Protocol (UDP). Starting with RDP 8.0, UDP is available to servers running Microsoft Remote Desktop Services. To be usable, a virtual machine (VM) must have enough hardware resources, namely: CPU, memory and, most critically, the graphics processing

unit (GPU). (The GPU is arguably the most critical component of a high-performance cloud workstation.) Windows Server 2016 provides several options for accessing the underlying graphics capabilities. The default Remote Desktop Services (RDS) GPU solution is also known as Windows Advanced Rasterization Platform (WARP). It's an adequate solution for knowledge-worker scenarios, but it provides inadequate resources for the cloud workstation scenarios. RemoteFX vGPU is a feature of RemoteFX that was introduced for remote connections. It provides a solution for scenarios with higher user densities per server, which allows a high-burst GPU utilization. However, when the time comes for using the power of the GPU, Discrete Device Assignment (DDA) is necessary to make the full use of the GPU power.

NV Series VMs are available with single or multiple Nvidia GPUs, as part of the Azure N Series offering. These VMs are optimized for remote visualization and virtual desktop infrastructure (VDI) scenarios. They use frameworks such as OpenGL and DirectX. Going up to 4 GPUs, you can provision workstations that take full advantage of the GPU, through Discrete Device Assignment (DDA) on Azure.

The Azure platform is programmable. It offers several options for a VM. For example, you can provision a workstation on demand. You can also keep the state of the remote machine on local files by using Azure Disks on Premium Storage or Azure Files. These options give you the ability to control costs. The Microsoft partnership with Citrix, for their XenDesktop and XenApp solutions, also provides another alternative for a desktop virtualization solution.

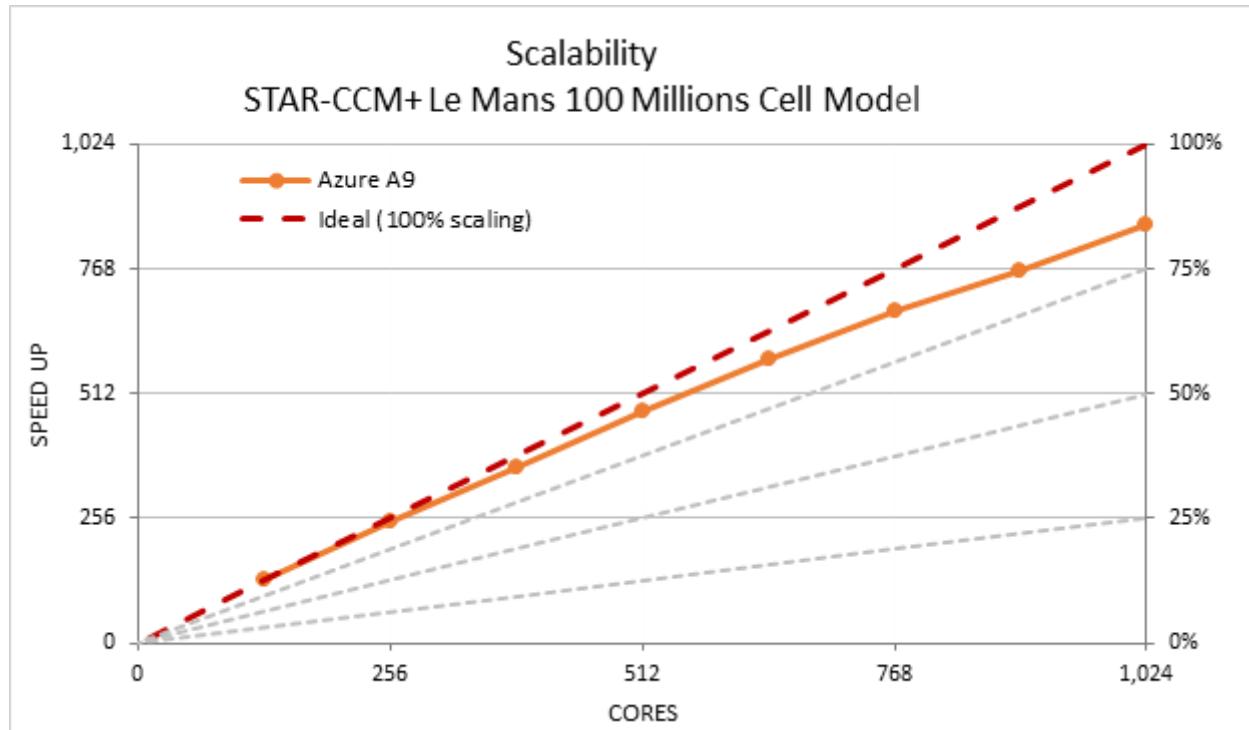
Analysis and simulation

Analysis and simulation of physical systems on computers has been around for a long time. Finite element analysis (FEA) is a numerical method that's used for solving many analysis problems. FEA requires a lot of computational power to perform large matrix calculations. The number of matrices involved in the solution of an FEA model explodes as we go from 2D to 3D and as we add granularity to the FEA mesh. This analysis requires computing power that's deployed on demand. It's important that the problem-solving code can be run in parallel to take advantage of the scalability of resources.

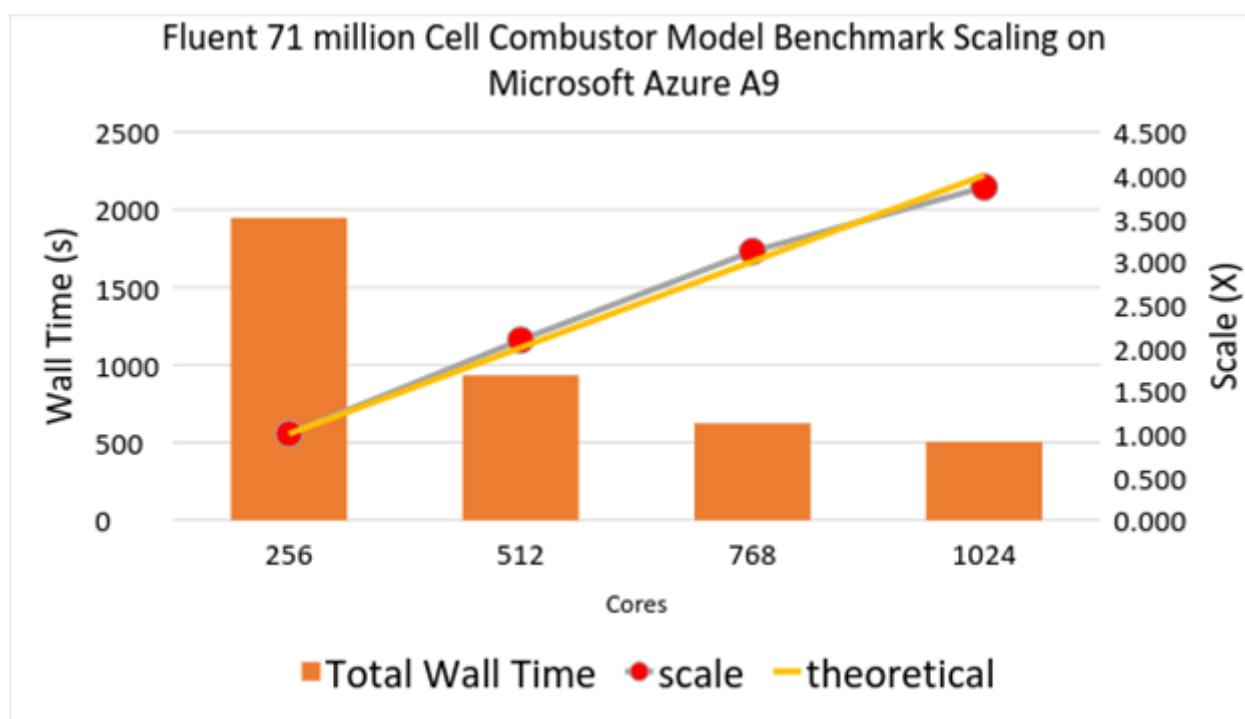
Solving simulation problems requires large-scale computing resources. High performance computing (HPC) is a class of large-scale computing. HPC requires low backend network latency, and remote direct memory access (RDMA) capabilities for fast parallel computations. The Azure platform offers VMs built for high-performance computing. They feature specialized processors that are paired with DDR4 memory, and they allow compute-intensive solutions to run effectively, both on Linux and Windows.

installations. They're available in several sizes. See [High-performance compute VM sizes](#). To see how Azure supports HPC in other ways, see big compute: [HPC & Batch](#).

The Azure platform enables solutions to scale up and out. One of the commonly known software packages for simulation is STAR-CCM+, from CD-adapco. [A published study](#) demonstrating STAR-CCM+ running "Le Mans 100 million cell" computational fluid dynamics (CFD) model provides a glimpse of the scalability of the platform. The following chart demonstrates the observed scalability as more cores are added for running the simulation:



Another popular engineering analysis software package is ANSYS CFD. It enables engineers to perform multi-physics analysis, including fluid forces, thermal effects, structural integrity, and electromagnetic radiation. [The published study](#) demonstrates the scalability of the solution on Azure and shows similar results.



Instead of investing in a local compute cluster, a software package that requires parallel execution can be deployed on Azure virtual machines or on [Virtual Machine Scale Sets](#) by using the [HPC and GPU VM](#) families for an all-cloud solution.

Burst to Azure

If a local cluster is available, another option is to extend it to Azure. This process offloads peak workloads and is known as *bursting to Azure*. To do so requires using one of the on-premises workload managers that support Azure. Recommended workload managers include [Alces Flight Compute](#), [TIBCO DataSynapse GridServer](#), [Bright Cluster Manager](#), [IBM Spectrum Symphony](#) and [Symphony LSF](#), [PBS Pro](#), and [Microsoft HPC Pack](#).

Another option is Azure Batch, which is a service for running large scale parallel and HPC batch jobs efficiently. Azure Batch allows jobs that use the Message Passing Interface (MPI) API. Batch supports both [Microsoft MPI](#) and Intel MPI with [HPC and GPU](#) optimized VM families. Microsoft also acquired [Cycle Computing](#), which provides a solution that offers a higher level of abstraction for running clusters on Azure. Another option is to run [Cray supercomputers](#) on Azure with seamless access to complementary Azure services such as [Azure Storage](#) and [Azure Data Lake](#).

Generative design

The design process is always iterative. A designer starts with a set of constraints and parameters for a target design. The designer iterates over several design alternatives, eventually settling on one that satisfies the constraints. However, when computational

power is almost infinite, one could evaluate *thousands* or even *millions* of design alternatives instead of a few. This journey started with parametric models, and their use in CAD tools. Now with vast computational resources on cloud platforms, the industry is going to its next step. Generative design is the term that describes the design process of providing parameters and constraints to the software tool. Then the tool generates design alternatives, creating several permutations of a solution. There are a few approaches to generative design: topology optimization, lattice optimization, surface optimization, and form synthesis. The details of those approaches are out of the scope of this article. However, the common pattern across those approaches is the need for access to compute-intensive environments.

The starting point of generative design is defining the design parameters over which the algorithm must iterate, along with reasonable increments and value ranges. The algorithm then creates a design alternative for each valid combination of these parameters. This process results in a huge number of design alternatives. Creating these alternatives requires a lot of computing resources. You must also run all the simulations and analysis tasks for each design alternative. The net result is that you need massive compute environments.

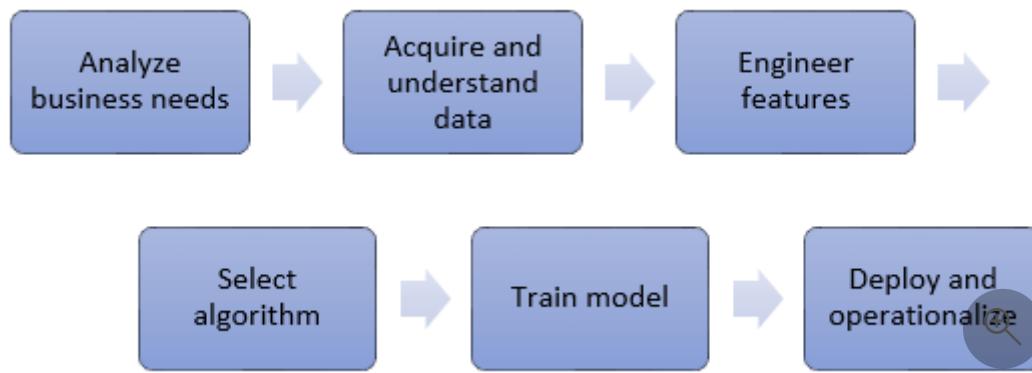
Azure's multiple options for scaling up on demand for the compute needs, through [Azure Batch](#), and [Virtual Machine Scale Sets](#), are natural destinations for those workloads.

Machine learning (ML)

At a simplistic level, we can characterize ML systems like this: when you're given a data point, or a set of data points, the system returns a correlated result. In this way, ML systems are used to solve questions, such as the following:

- Given the past house prices and properties of houses, what is the predicted price of a given house that comes to the market?
- Given the readings from various sensors and past failure cases of a machine, what is the likelihood that the machine fails in the next period?
- Given a set of images, which one is a domestic cat?
- Given a video feed of an oil pipeline, is there a damaged section of it that has substantial dents?

Adding an advanced analytic capability by using artificial intelligence (AI) and machine learning (ML) starts with developing a model, with a process that's similar to the following.

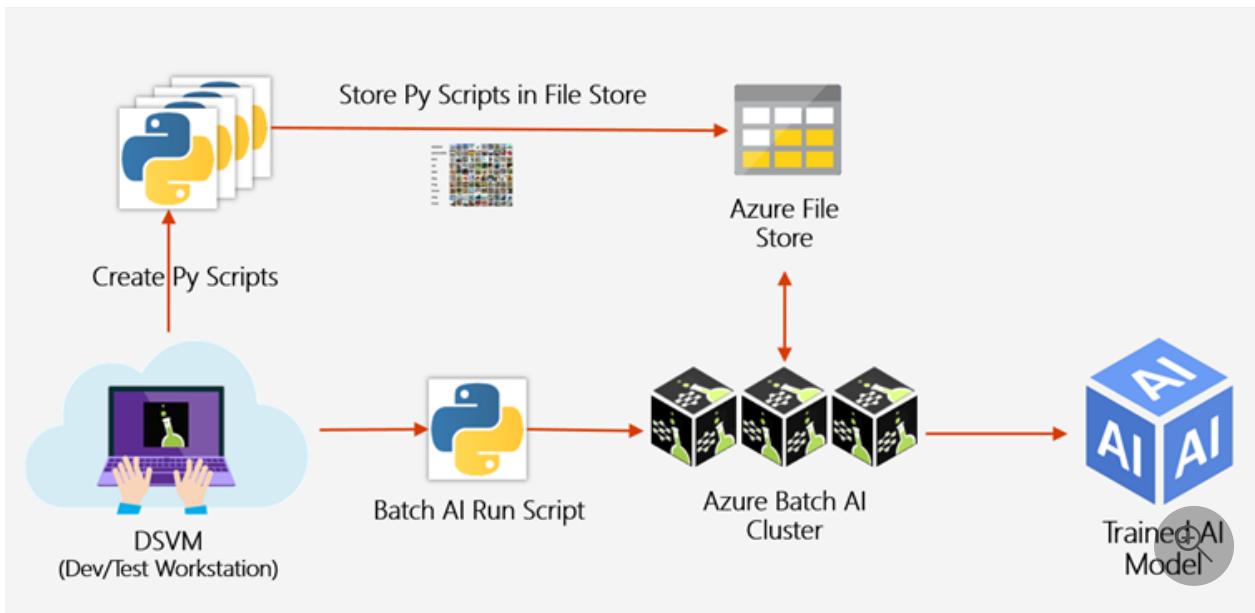


Selecting the algorithm depends on the size, the quality, the nature of the data, and the type of the answer that is expected. Based on the input size, the selected algorithm, and the computing environment, this step usually requires big compute intensive resources. It can take different times to complete. The following chart is from [a technical article](#) for benchmarking the training of ML algorithms. It shows the time to complete the training cycle given different algorithms, data set sizes, and computation targets (GPU or CPU).

Airline subsample size	Lib	CPU (50 rounds)		GPU (50 rounds)		CPU (200 rounds)		GPU (200 rounds)		CPU (500 rounds)		GPU (500 rounds)	
		training time	AUC	training time	AUC	training time	AUC	training time	AUC	training time	AUC	training time	AUC
10,000	xgb	0.139s	0.725	4.099s	0.742	0.867s	0.729	3.063s	0.746	1.287s	0.720	3.605s	0.732
	xgb_hist	0.544s	0.735	1.510s	0.742	4.791s	0.736	4.258s	0.746	5.432s	0.715	7.456s	0.722
	lgb	0.141s	0.737	0.985s	0.743	2.149s	0.724	3.924s	0.743	1.093s	0.718	6.124s	0.726
100,000	xgb	0.887s	0.775	3.773s	0.776	3.433s	0.795	8.715s	0.794	8.288s	0.801	14.523s	0.804
	xgb_hist	2.590s	0.786	3.126s	0.789	15.108s	0.795	17.139s	0.795	32.416s	0.792	33.020s	0.796
	lgb	0.720s	0.788	4.716s	0.790	4.400s	0.796	14.481s	0.794	7.575s	0.795	30.175s	0.796
1,000,000	xgb	10.509s	0.788	15.416s	0.786	51.787s	0.820	59.977s	0.821	154.981s	0.833	128.406s	0.834
	xgb_hist	6.088s	0.806	7.484s	0.805	23.949s	0.831	29.357s	0.833	50.610s	0.837	55.009s	0.839
	lgb	2.282s	0.806	8.312s	0.806	10.663s	0.831	26.191s	0.834	21.502s	0.838	42.278s	0.840
10,000,000	xgb	143.350s	0.791	153.138s	0.790	565.631s	0.828	526.385s	0.826	1579.905s	0.844	1392.987s	0.843
	xgb_hist	34.149s	0.791	64.802s	0.808	124.467s	0.840	159.932s	0.840	229.613s	0.854	238.149s	0.854
	lgb	28.167s	0.808	34.269s	0.807	111.330s	0.841	92.459s	0.841	162.910s	0.855	124.260s	0.855
100,000,000	xgb	1732.920s	0.791	-	-	6971.948s	0.828	-	-	-	-	-	-
	xgb_hist	304.908s	0.808	427.252s	0.808	1008.774s	0.841	1141.762s	0.840	1958.587s	0.856	2098.376s	0.856
	lgb	252.798s	0.809	307.032s	0.809	969.227s	0.842	506.922s	0.841	1610.885s	0.857	977.676s	0.857

The major driver for the decision is the business problem. If the problem requires a large data set+ to be processed with a suitable algorithm, the critical factor is cloud scale compute resources for training the algorithm. [Azure Batch AI](#) is a service that trains AI models in parallel and at scale.

With Azure Batch AI, a data scientist can develop a solution on the workstation using the [Azure Data Science Virtual Machine \(DSVM\)](#) or [Azure Deep Learning Virtual Machine \(DLVM\)](#) and push the training to the cluster. DSVM and DLVM are specially configured VM images with a rich set of preinstalled tools and samples.



Conclusion

The manufacturing industry requires massive numbers of mathematical calculations, using access to high-end hardware components, including graphics processing units (GPU). Scalability and elasticity of the platform that hosts the resources is crucial. To control costs, it must be available on demand while supplying the optimum speed.

The Microsoft Azure platform provides a wide array of choices for fulfilling these needs. You can start from scratch, control every resource and aspect of it to build your own solution. Or you can find a Microsoft partner to expedite the solution creation. Our partners know how to take advantage of the power of Azure.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal author:

- [Ercen Keresteci](#) | Technical Strategist

To see non-public LinkedIn profiles, sign in to LinkedIn.

Next steps

- Set up a cloud workstation by deploying an [NV series VM](#).
- Review the [options](#) for deploying a tool for your design needs to take advantage of Azure HPC capabilities.

- Learn the possibilities with [Azure Machine Learning](#).

Related resources

Related guides:

- [Introduction to predictive maintenance in manufacturing](#)
- [Predictive maintenance solution](#)
- [Use subject matter expertise in machine teaching and reinforcement learning](#)
- [Extract actionable insights from IoT data](#)
- [Azure industrial IoT analytics guidance](#)

Related architectures:

- [End-to-end computer vision at the edge for manufacturing](#)
- [Implement real-time anomaly detection for conveyor belts](#)
- [Anomaly detector process](#)
- [Quality assurance](#)
- [Predictive maintenance solution](#)
- [Facilities management powered by mixed reality and IoT](#)
- [Batch integration with Azure Data Factory for Azure Digital Twins](#)
- [Supply chain track and trace](#)

Risk grid computing in banking

Azure Batch

Azure Data Factory

Azure ExpressRoute

Azure HDInsight

In corporate finance and investment banking, one of the most important jobs is analyzing risk.

Financial risk analysts provide a comprehensive picture of the risk associated with an investment portfolio. They review research, monitor economic and social conditions, stay abreast of regulations, and create computer models of the investment climate.

Risk analysis across the many vectors that affect a portfolio is sufficiently complex that computer modeling is required. Most analysts spend quite a bit of time working with computer models to simulate and predict how financial conditions will change. When you evaluate investment risks (Market Risk, Credit Risk and Operational Risk), the computational load of processing the predicated models can be quite large due to the volume of and diversity data.

Cloud computing offers significant benefits for risk grid computing or risk modeling because it enables analysts to access massive compute resources on demand, without incurring capital costs or managing infrastructure. This article examines leveraging Microsoft Azure to augment current risk grid compute resources and optimize the cost and speed of risk grid computing workloads. Topics covered include secure and reliable connectivity, batch processing, and augmenting compute resources based on demand when on-premises servers are at capacity.

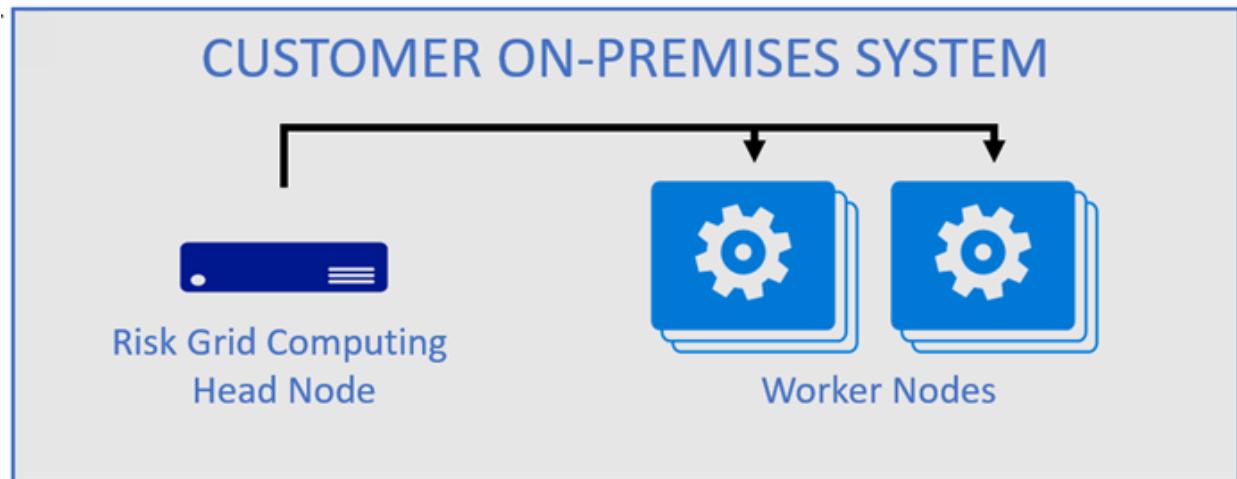
Grid computing services

Analysts need a simple and reliable way to provide their models to a batch processing pipeline. It starts with data ingestion and flows through data processing to analysis, where insights can be derived from the resulting data.

Risk model input data comes in several forms, the most common being Excel files or .csv files. These files are often restructured into formats more suitable for processing the risk model in later stages of the risk compute pipeline. A common technique for parsing and processing these files is batch processing with a grid of virtual machines working together to reach a common goal.

[Azure Batch](#) is an Azure service that allows multiple worker VMs to run in parallel, as shown below. Processing data files and submitting results to machine learning systems or data stores are common tasks for the worker nodes. The application code run by the

worker nodes is created by the customer, so almost any action may be taken in the batch job.



Azure provides an elegant solution for risk grid computing using Azure Batch. Customers can use Azure Batch to extend their existing risk computing grid, or to replace on-premises resources with a completely cloud-based solution.

Connecting directly and securely to the Azure cloud is fully supported. The Azure Batch risk processing grid worker nodes can access modeling data when they connect to data stored on-premises, when you connect to Azure with a hybrid network. The customer can also upload data to appropriate storage within Azure, allowing Batch to have direct access to the data.

Secure connectivity to Azure

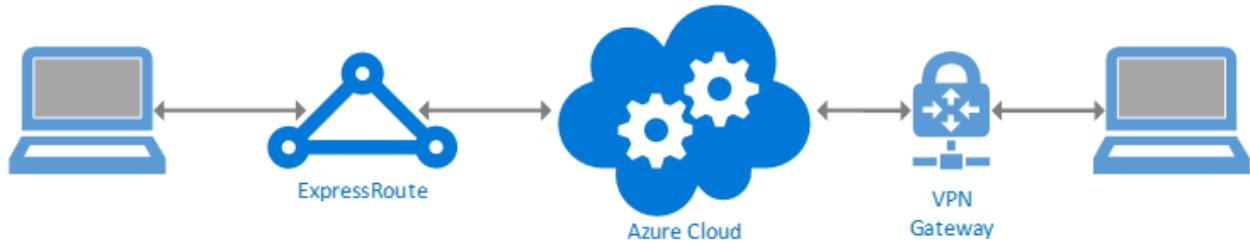
When you build a risk grid computing solution on Azure, the business will often continue to use existing on-premises applications such as trading systems, middle office risk management, risk analytics, and so on. Azure becomes an extension to those existing investments.

When you connect to the cloud, security is a primary consideration. Accounting for your current security model is the first step in connecting directly to Azure. For customers already using Active Directory on-premises, connecting to Azure can leverage existing identity resources. Service accounts can live in the on-premises AD.

Hybrid network solution

A hybrid network ties Azure directly to the customer's on-premises network. Azure offers two models for securely and reliably connecting current on-premises systems to Azure, [Microsoft Azure ExpressRoute](#) and [VPN Gateway](#). Both are trusted connectivity

solutions, although there are differences in implementation, performance, cost and other attributes.



"Burst to cloud" offloads computing jobs to cloud-based machines when existing resources spike, augmenting the customer's data center or private cloud resources. Using the hybrid network model allows for easy burst to cloud scenarios as the cloud-based risk computing grid is a simple extension of the existing network.

There are several network connectivity configurations beyond those in the simple model presented in the logical architecture above. To help with decisions and architectural guidance regarding connecting your network to Azure, see the article [Connect an on-premises network to Azure](#).

REST API solution over the internet

An alternative to creating a hybrid network is to upload data into Azure Storage and have Batch read the data files from storage. This can be achieved using a secure connection to connect to Azure, storing the documents in Azure Storage, and then managing the risk grid computing jobs via the [Batch service REST API](#) or SDK with a fit-for-purpose application, orchestrating the Batch run.

Azure Data Factory

Another solution for your scenario may be using [Azure Data Factory](#), a cloud-based data integration service, to compose large storage, movement and processing pipelines. Data can be uploaded on demand through a Data Factory pipeline. The service provides a visual designer in the Azure portal for building Extract, Transform and Load solutions in Azure. Data Factory can help ingest data into Azure for further processing.

Matching processing needs with demand

When you compute risk, whether daily or with the heavier loads at the end of the month, the calculations consume significant computational resources. These calculations don't run 24x7. When risk calculations aren't being run on the on-premises grid, the

organization leaves valuable and expensive servers running with no workload, but with ongoing costs for power, cooling, and datacenter space, along with other fixed costs.

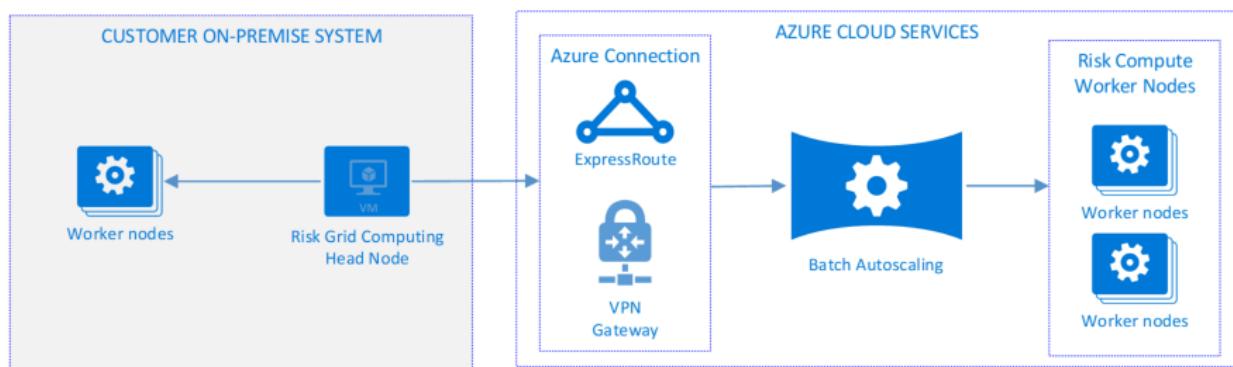
Augmenting an on-premises grid with Azure Batch

To minimize costs, a business could choose to own and manage just enough worker nodes to satisfy requirements when demand is low. High demand risk grid computing jobs can then be pushed to high performance servers in Azure, elastically scaling up and down with workload demand.

The Azure Batch processing model has several benefits for risk grid computing:

- Augments existing investments in various on-premises systems.
- Allows existing infrastructure to serve risk analysis needs when demand is low, deallocated Azure-based worker nodes.
- Provides extra capacity to the risk compute grid when demand is high.
- Enables matching machine profiles to the processing power needed by Batch workload, even when the load calls for **High Performance Computing (HPC)** configurations.

A common solution is to automatically add worker nodes in Azure when the on-premises workers are all in use. The risk grid head node simply asks for more workers. This automatically scales the number of grid worker nodes in Azure and enables an elastic demand solution.



Along with efficient use of resources, this arrangement provides other benefits. For independent tasks, adding more workers allows the load to scale linearly. Azure also provides the flexibility to try out a very large VM instance or a machine with several GPU cards. This flexibility enables experimentation and innovation.

For times when more compute capacity is needed, such as quarterly valuations, the extra capacity can also come from Azure Batch auto-scaling. Auto-scaling provides elasticity to your Batch solution. By scaling resources to match needed load, Azure provides significantly greater capacity at a lower cost than owning the hardware.

Most commercial grid products do support some form of burst to cloud, enabling easier proofs of concept for your risk analysis load. For example, [Microsoft HPC Pack](#) can run in Azure, as can products from companies like TIBCO, Univa, and others. Many of these 3rd party tools or systems are available through the [Azure Marketplace](#).

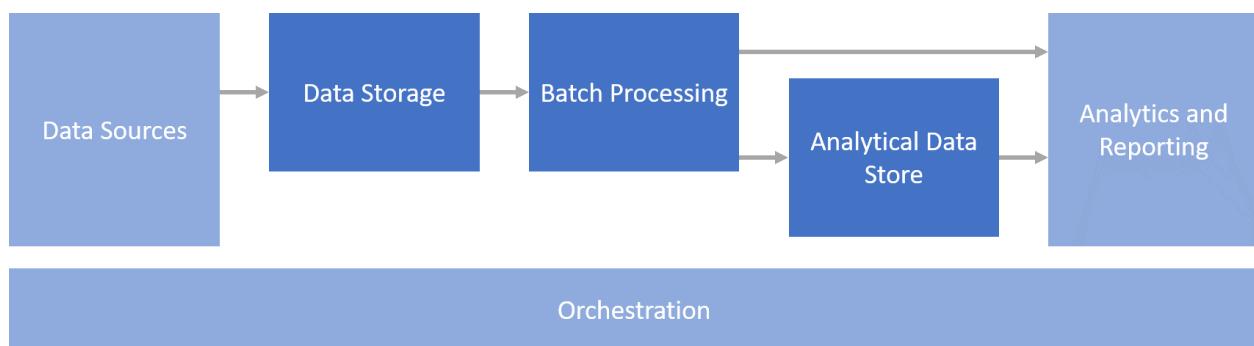
Migrating additional resources to the cloud

As workloads grow or on-premises datacenter infrastructure ages, organizations can move their entire Batch processing for risk grid computing into Azure.

Growing into Azure

As on-premises machines reach end of life, you can further distribute worker nodes into the cloud. The same can be true for the Batch head node. This inverts the relationship between the on-premises network and Azure. This may be an opportunity to decrease costs by decommissioning any network-to-network products such as Azure ExpressRoute, and any remaining on-premises worker nodes.

As part of this change, data may be made available to Azure using various file ingress techniques. Azure has many storage options to choose from, including rest endpoints to allow uploading data directly, rather than having the compute jobs pick it up from the on-premises network.



Under this model, all risk grid computing activities can take place in the cloud. Data files processed by the workers may be stored in Azure storage, data can be fed directly into the Azure Data Lake, and Azure HDInsight can take care of machine learning needs. Finally, Power BI and Azure Analytics are excellent data analysis tools and can work across all data stored in Azure.

Data security considerations for risk grid computing

While calculation data often doesn't include any customer content, most banks are still likely to conduct a security risk assessment before placing any workload in the cloud.

This assessment might require input from Microsoft and may result in security recommendations.

A notable consideration for risk grid computing is to [run the batch processes within an Azure VNet](#). This allows pool compute nodes to communicate securely with other compute nodes, or with an on-premises network. Appropriate service accounts and Network Service Groups (NSG) should be created and used by the batch compute nodes. [Azure also has solutions](#) for data encryption in transit and at rest in Azure storage.

Some areas to consider may be: Active Directory (AD) or non-AD joined compute nodes, [VM disk encryption](#), security of calculation input and output data at rest and in transit, Azure network configurations, permissions and more. Authentication may also be handled at the REST API level through a secret key.

Getting started

Many customers have an in-house risk computing grid they already use. If your company developed the grid internally, consider Azure Batch to extend the grid. A good place to start with Azure Batch is by extending any current on-premises solution by replicating the current processing application logic and running it as a Batch job in Azure. This may require a networking solution for joining the Azure Batch compute nodes to the on-premises network, depending on our application's functionality.

To mitigate any security, speed, and connection reliability concerns, consider connecting your on-premises network to Azure using Azure ExpressRoute or a VPN Gateway. From there, you may have your on-premises head node provision a cluster of Azure-based worker nodes, spinning them up and down as needed.

Lastly, you may be ready for a complete migration of your risk compute infrastructure to Azure. If this is the case, [here's an article](#) to get you started today.

Components

- [Azure Batch](#) ↗ enables augmenting on-premises risk computing worker nodes to dynamically provide compute resources based on demand.
- [Azure Data Lake](#) ↗ provides storage, processing and analytics across for your risk analysis data.
- [Azure ExpressRoute](#) ↗ extends your on-premises network to Azure over a private connection facilitated by a connectivity provider.

- [Azure HDInsight](#) is a fully managed open-source analytics service to process massive amounts of data such as the data provided in month-end batch runs.
- [Microsoft HPC Pack](#) enables provisioning High Performance Computing clusters for batch processing.
- [Power BI](#) is a suite of business analytics tools risk analysts use to gain and share insights.
- [Azure VPN Gateway](#) extends your on-premises network to the Azure cloud over the Internet.

Conclusion

The solutions covered in this article are approaches to risk grid computing in banking. Other architectures may be used given the rich capabilities of Azure products and services and the various existing client system architectures. Even so, Batch provides a reasonable model for risk grid computing given the advantages laid out in this article.

[Extending the on-premises network to Azure](#) allows Azure easy access to network resources and other processing systems already present in the on-premises network. When on-premises machines are reaching end of life, it may make more sense to use Batch compute entirely in Azure rather than supporting a hybrid model.

Uploading files to Azure Storage before the Batch job begins is another way to take advantage of Batch without the need for a hybrid network. This could be done incrementally, or as a starting process to the Batch run.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [David Starr](#) | Principal Solutions Architect

Next steps

After you select a connectivity strategy, a logical place to start with risk compute is placing your existing jobs into Azure compute worker nodes and running them in a test

environment to see if any code needs to be changed. [This article provides a starting point](#) for getting started with Azure Batch in the language or tool of your choice.

See the [Risk Grid Computing in Banking Solution Guide](#).

Product documentation:

- [What is Azure Batch?](#)
- [What is Azure Data Lake?](#)
- [What is Azure ExpressRoute?](#)
- [What is Azure HDInsight?](#)
- [What is Power BI?](#)
- [What is VPN Gateway?](#)

Related resources

- [Risk grid computing solution](#)
- [HPC risk analysis template](#)
- [Loan credit risk and default modeling](#)
- [Actuarial risk analysis and financial modeling](#)

Risk grid computing solution

Azure Batch Microsoft Entra ID Azure ExpressRoute Azure VPN Gateway

This article provides a technical overview of using Microsoft Azure to support and enhance risk grid computing in banking. The article explores recommended systems and high-level architectures.

This document is intended for Solution Architects, and in some cases Technical Decision Makers, who want a deep dive on proposed solutions for risk computing.

Introduction

Financial risk analysis models are typically processed as batch jobs. They have heavy compute loads generating high demand for computing power, data access, and analysis. Demand for risk grid computing calculations often grows over time, and the need for compute resources increases with it.

The broad range of available products and services in Azure means there can be more than one solution to most problems. This article provides an overview of the technologies, patterns, and practices that are the most effective for a risk grid computing solution in banking that uses [Microsoft Azure Batch](#).

Azure Batch is a free service that provides cost-effective and secure solutions. The solutions are for both the infrastructure and the various stages of batch processing that are typically used with risk grid computing models. Azure Batch can augment, extend, or even replace current on-premises compute resource investments using hybrid networks or by moving the entire Batch process into Azure. Data can traverse up and down from the cloud or stay on-premises. Other data can be processed by compute nodes in a burst-to-cloud model, when on-premises resources run low.

Anatomy of an Azure Batch run

There are typically at least two applications involved in a Batch run. One application, typically run on a "head node," submits the job to the pool and sometimes orchestrates the compute nodes. The orchestration can also be configured via the Azure portal. The other application is run by the compute nodes as a task (see Figure 1).

The compute node application performs the task of parallel processing risk modeling files. There can be more than one application installed and run on the compute nodes.

These applications can be uploaded via the [Batch API](#), directly through the Azure portal, or via the [Azure CLI commands for Batch](#).

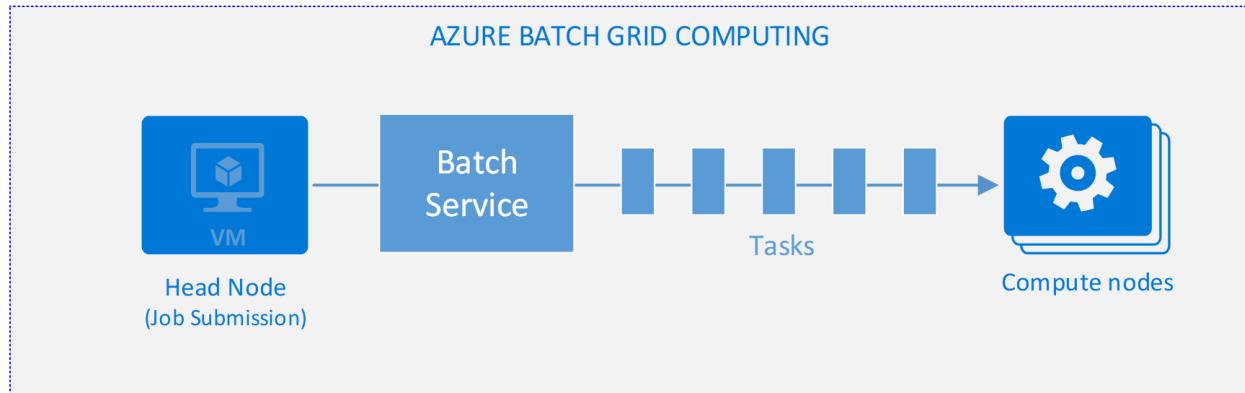


Figure 1: Azure Batch grid computing

An Azure Batch run consists of several logical elements. Figure 2 shows the logical model of a batch job. A pool is a container for the VMs involved in the Batch run and provisions the compute node VMs. A pool is also the container for the applications installed on the compute nodes. Jobs are created and run within the pool. Tasks are executed by the jobs. Tasks are a run of the worker application and are invoked by a command-line instruction.

The worker application is installed to the compute node when it's created.

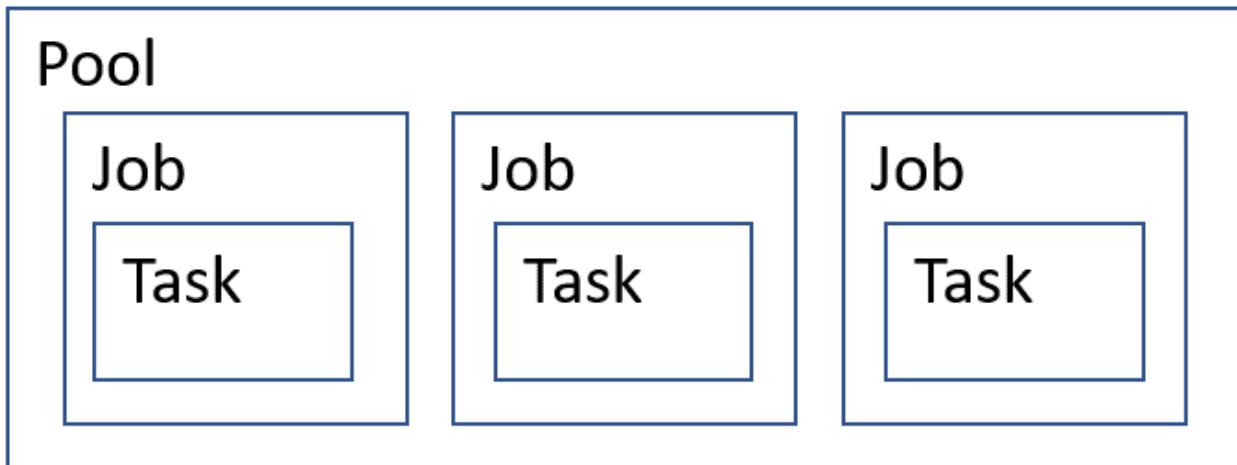


Figure 2: Logical batch concept model

When the job executes, the pool provisions any worker VMs needed and installs the worker applications. The job assigns tasks to those compute nodes, which in turn run a command-line instruction (CLI). The CLI script typically calls the installed applications or scripts.

Using Batch typically follows a prototypical pattern, described as follows:

1. Create a resource group to contain the Batch assets.

2. Within the resource group, create a Batch account.
3. Create a linked storage account.
4. Create a pool in which you can provision the worker VMs.
5. Upload the compute node application or scripts to the pool.
6. Create a job to assign tasks to the VMs in the pool.
7. Add the job to the pool.
8. Begin the Batch run.
9. The job queues tasks to run on the compute nodes.
10. Compute nodes run the tasks as the VMs become available.

An illustration of this process is show in Figure 3.

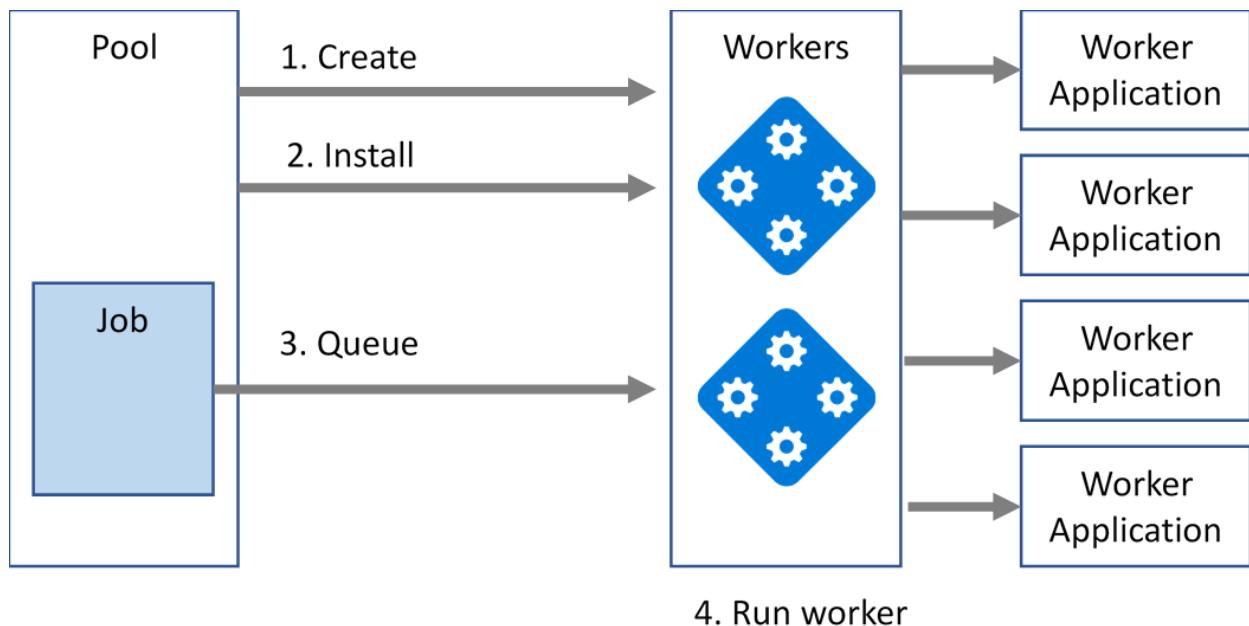


Figure 3: Logical batch concept model

Once tasks are complete, it can be useful to remove the compute nodes to not incur fees while not in use. To delete them, via code or the portal, one can delete the containing pool, which will remove the worker VMs.

For more detailed walkthroughs on how to get starting with Batch, [5-minute Quickstarts](#) take you through the process in several languages, and it also shows you how to use the Azure portal.

Batch process scheduling

Azure Batch has a scheduler built in so scheduling of each run can be defined in the portal or through APIs. The Batch job scheduler can define multiple schedules to fire multiple jobs. Each job has its own properties such as what to do when the job starts and ends. Job schedules can be set on recurring intervals or for a one-time run.

Many bank grid computing systems already have their own scheduling service. There may be no immediate need to move my scheduler to Azure. This can work seamlessly because Azure Batch can be invoked manually or through an SDK. This ability allows scheduling to still occur on-premises, and it allows workloads to be processed in Azure.

Batch processing can happen on a predetermined schedule or on demand. In either case, there's no need to keep compute node VMs alive when they aren't being used. When using hundreds, if not thousands, of VM compute nodes, significant cost savings can be realized, by de-provisioning the servers when they're done running their queued tasks.

Compute node applications

Compute nodes need an application to run when a task is invoked. These applications are written by the business to perform the processing jobs when installed on the workers. In risk grid computing for banking scenarios, this application often takes on the job of transforming data into formats especially suited for downstream analytics or other processing.

When providing the application to the pool for distribution to compute nodes, it's uploaded in an application package. An application package may be another version of a previously uploaded application package. More than one application package can be installed to one compute node. The job contains the applications packages to load onto the worker machines.

Application package deployment may also be managed by version. If multiple versions of an application package have been loaded into a pool, a specific version may be designated for use in a Batch run as shown in Figure 4. This may be necessary in audit environments or when the business wants to reproduce a prior run. It can also be used for roll-back purposes if a bug is introduced to the worker application.

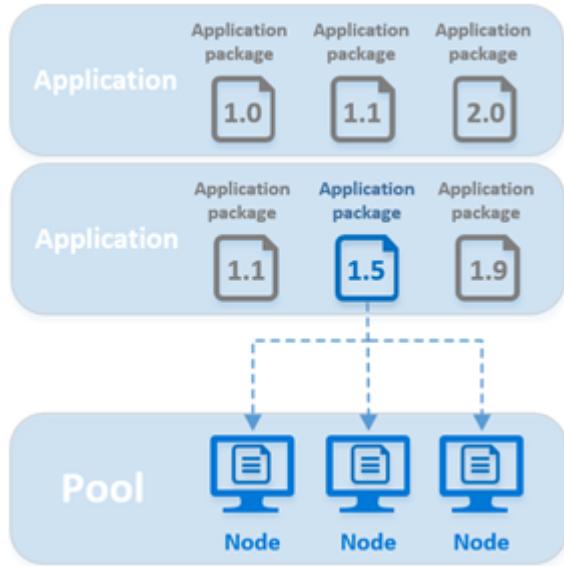


Figure 4: Versioning compute node task applications

An application package is uploaded to the pool as a .zip file. The file contains the application binaries and the supporting files that are required for tasks to run the application. There are two scopes for application packages. You can designate an application package in scope of the pool, or in the scope of tasks.

Pool application packages

These packages are deployed to every compute node in the pool. When a compute node VM is provisioned, rebooted, or reimaged, a new copy of any pool application packages is installed, if an updated application exists. One or more application packages can be assigned to a pool, which means the compute nodes will get all packages designated.

Task application packages

Application packages targeting the task level are only installed to compute nodes scheduled to run a task. Task applications packages are intended for use when more than one job is run in one pool.

Task applications are useful when aggregating data that's produced by pool-level jobs. These applications can be relevant in risk grid computing scenarios. For example, a task application can run a set of risk calculations that generate data to be used later in the risk calculation workflow.

Scaling batch jobs

Banks often perform risk analysis batch runs over weekends or at night when computing resources are underutilized. While this model works for some, it can quickly be outgrown, requiring more capital to add more worker machines to the grid.

If Azure Batch jobs take too long to run, or if you want more computing power in your Batch runs, Azure offers several options.

1. Allocate more compute node machines to scale out.
2. Allocate more powerful compute node machines to scale up. Azure machines may be provisioned to meet high performance needs of for cores and memory, and even GPU computing power.

Note: Using Microsoft HPC Pack with Batch is a more complex model and isn't discussed in this article.

In a Batch processing cluster, you might have as few as two processing VMs. Or you could have thousands of simultaneous tasks running on thousands of VM compute nodes, with tens of thousands of cores. Each VM is responsible for running a single task at a time. The number of VMs in a pool can be scaled manually or automatically, as configured when load increases or decreases.

Burst to cloud

When compute resources in an on-premises grid are running low due to executing a large analysis job, "burst to cloud" offers a way to augment those resources by adding more compute nodes in Azure. Burst to cloud is a model in which private clouds or infrastructure distribute their workload to cloud servers, when demand is high for on-premises resources.

These compute nodes can be pre-configured as Linux or Windows virtual machines to be provisioned in Azure's IaaS platform. Further, servers can be provisioned and automatically configured to work with existing investments like Tibco Gridserver and IBM Symphony.

Automatic scaling formulas

This elasticity can be configured in the Azure portal or by using [automatic scaling formulas](#). Automatic scaling formulas are scripts uploaded to the Batch processing scheduler for fine-grained control of Batch behavior. Automatic scaling on a pool of compute nodes is done by associating the nodes with autoscale formulas.

The following example is of an automatic scaling formula directing auto scaling to start with one VM and scale up to 50 VMs as needed. As tasks complete, VMs become free one by one and the auto scaling formula shrinks the pool.

Formula

```
startingNumberofVMs = 1;
maxNumberofVMs = 50;
pendingTaskSamplePercent = $PendingTasks.GetSamplePercent(180 *
TimeInterval_Second);
pendingTaskSamples = pendingTaskSamplePercent < 70 ? startingNumberofVMs :
avg($PendingTasks.GetSample(180 * TimeInterval_Second));
$TargetDedicatedNodes=min(maxNumberofVMs, pendingTaskSamples);
```

Other scaling techniques

Automatic scaling can also be enabled by the `Enable-AzureBatchAutoScale` PowerShell cmdlet. The `Enable-AzureBatchAutoScale` cmdlet enables automatic scaling of the specified pool. An example follows.

1. The first command defines a formula, and then saves it to the `$Formula` variable.
2. The second command enables automatic scaling on the pool named `RiskGridPool` using the formula in `$Formula`.

Console

```
C:\> $Formula = 'startingNumberofVMs = 1;
maxNumberofVMs = 50;
pendingTaskSamplePercent = $PendingTasks.GetSamplePercent(180 *
TimeInterval_Second?WT.mc_id=gridbanksg-docs-dastarr);
pendingTaskSamples = pendingTaskSamplePercent < 70 ? startingNumberofVMs :
avg($PendingTasks.GetSample(180 * TimeInterval_Second));
$TargetDedicatedNodes=min(maxNumberofVMs, pendingTaskSamples);';

C:\> Enable-AzureBatchAutoScale -Id "RiskGridPool" -AutoScaleFormula
$Formula -BatchContext $Context
```

Scaling can also be accomplished using the Azure CLI with the `az batch pool resize` command and through the Azure portal.

Data storage and retention

Once data is ingested and processed by a compute node, the resulting output data can be stored in a database. The output data can be further processed and analyzed or

transformed upon ingestion, before storage, in order to ensure the proper formats for downstream processing. Microsoft Azure offers several storage options. The choice of [which data store technology to use](#) depends largely on analysis and or reporting needs in downstream processes.

When using a hybrid network, the data storage target may be on-premises. When using Batch over a hybrid network, compute nodes can write data back to an on-premises data store without using an Azure-based storage location. Workers can also write to Azure File storage, which can be mounted as a disk on an on-premises machine. This setup allows easy access by any process that works with the files on-premises.

Monitoring and logging

To optimize future runs of the Batch job, data should be recorded to help identify areas of optimization. For example, if workers are running near CPU capacity, adding cores to the compute nodes may help avoid being CPU-bound and the job can finish faster. Each application run in the Batch job has its own characteristics and the optimizations made to the VMs in the Batch runs may differ. For memory-intensive tasks, more memory can be allocated by configuring the machines differently in the next run.

Logging can be done by the compute node and grid head applications or by a job using [Batch diagnostic logging](#). Logging information about the performance of the Batch runs can be configured to help identify which areas to improve for better performance and faster task completion.

Custom Batch monitoring and logging

The controlling application and compute node applications can generate this data and store it for further analysis. Data found helpful in optimizing Batch jobs includes:

- Start and end times for each task
- The time each compute node is alive and running tasks
- The time each compute node is alive and not running tasks
- The overall batch job run time

Batch diagnostic logging

There's an alternative to using the controller and compute node applications to emit instrumentation data. [Batch diagnostics logging](#) can capture a lot of the run data. Batch Diagnostic Logging isn't enabled by default and must be enabled for the Batch account.

Batch diagnostic logging provides a significant amount of data aiding in troubleshooting and optimizing Batch runs. Start and end times for job and tasks, core count, total node count, and many other metrics.

Batch logging requires a storage destination for the logs emitted, storing events produced by the Batch run such as pool creation, job execution, task execution, and so on. In addition to storing diagnostic log events in an Azure Storage account, Batch service log events can be streamed to an instance of [Azure Event Hubs](#). The events can then be sent to [Azure Log Analytics](#).

Using these data, core computing and head node applications can be optimized. This can lower costs, due to things like faster deprovisioning worker VMs, when they're no longer needed, rather than waiting for the end of the Batch run to complete.

Batch management tools

The Azure portal provides a Batch monitoring dashboard that shows information about Batch as jobs are running and even account quota usage. It's sufficient for many Batch job applications.

In addition to the Batch management and visualization tools available in the Azure portal, there's a free open-source tool, [Batch Explorer](#), for managing Batch. This is a standalone client tool to help create, debug, and monitor Azure Batch applications. Download an installation package for Mac, Linux, or Windows.

Network models

Risk analysis often requires hundreds, if not thousands, of documents to be ingested into the risk grid computing process. These files are often located on-premises in a file store, network share, or other repository. When using Azure-based VMs to access and process those files, it's often useful for the on-premises network to be seamlessly connected to the Azure network, so file access is simple and fast. This approach may even mean no code changes are needed to the code doing the processing on the compute nodes.

Azure offers two models for securely and reliably connecting current on-premises systems to Azure, [Microsoft Azure ExpressRoute](#) and [VPN Gateway](#). Both offer secure reliable connectivity, although there are differences in implementation, performance and [other attributes](#).

Alternatively, the risk grid computing head node may live on-premises and execute the Batch job through the REST APIs or SDKs in .NET and other languages.

There are other techniques for bridging the gap between Azure and on-premises resources without a hybrid network solution. More information on this is provided below.

ExpressRoute

ExpressRoute ties your on-premises or datacenter network to Azure through a private connection facilitated by a connectivity partner, such as your current Internet Service Provider. This enables both networks to see each other as the same network instance, providing seamless access between networks. Network integration is critical when you want to integrate existing on-premises systems with an Azure network, and ExpressRoute offers the fastest connection speeds possible.

Additional pricing information for Azure ExpressRoute [can be found here ↗](#).

VPN Gateway

A VPN Gateway is another way to connect your network to Azure. The downside of this model is traffic flows over the Internet. The connection can be less resilient as a result and network speeds can't reach those of ExpressRoute, however this may not be a barrier for a risk grid computing scenario as reading data files is typically a fast operation.

Additional pricing information for VPN Gateway [can be found here ↗](#).

Choices for connectivity details

There are essentially two models for extending your network to Azure, as shown in Figure 5.

- Virtual gateway – site-to-site
- ExpressRoute – Exchange or ISP provider

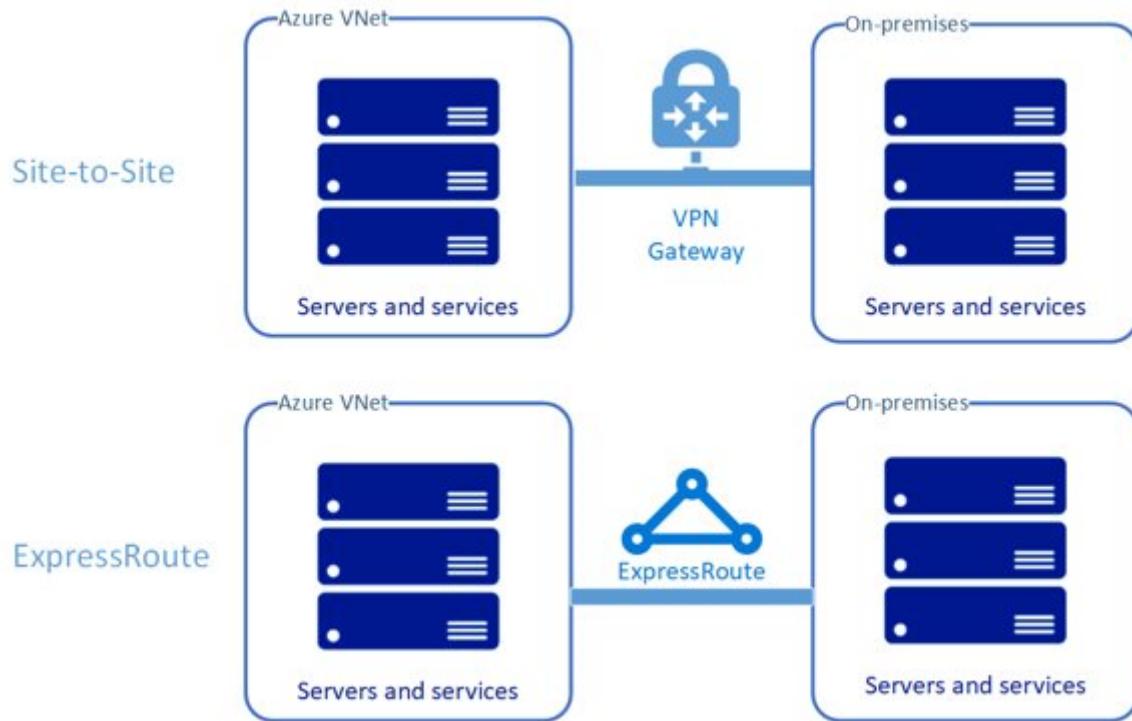


Figure 5: Site-to-Site and ExpressRoute

Virtual gateway site-to-site integration

A [Site-to-Site VPN Gateway](#) connects your on-premises network to an Azure VNet. This bridges the gap between networks essentially making them parts of the same network, with two-way access to resources, servers, and artifacts. This allows direct access to data files from the Azure worker VMs running the risk grid computing batch job.

ExpressRoute integration

An ExpressRoute connection facilitated by an Azure partner network provider realizes the same benefits as a Site-to-Site connection, but with higher speeds and reliability.

Get more information about [ExpressRoute connectivity models](#).

Batch processing without an Azure hybrid network

Another Batch scenario is uploading all data files into Azure storage for later processing by Azure-based compute machines. File storage and Blob storage are likely candidates for storing risk grid computing data.

In this scenario, the job controller and all compute nodes live in Azure as shown in Figure 6. The likely destination for processed data is an Azure data store, in preparation for further processing by Azure Machine Learning solutions or other systems. This additional processing is beyond the scope of this article.

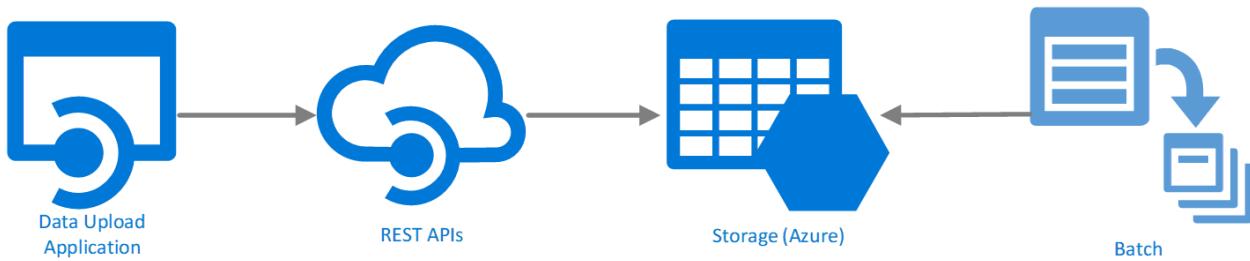


Figure 6: Batch upload to execution lifecycle

Hybrid network connectivity resources

Several configurations may be applicable in your situation. To help with decisions and architectural guidance regarding connecting network connectivity to Azure, see the article [Connect an on-premises network to Azure](#) by the patterns & practices group.

- See this article for VPN Gateway configuration alternatives.
- Learn about [ExpressRoute connectivity models](#).
- Calculate [ExpressRoute pricing](#).
- Calculate [VPN Gateway pricing](#).

Security considerations

An Azure [virtual network \(VNet\)](#) may be created and the pool's compute nodes created within it. This provides an extra level of isolation for the Batch runs and allows authentication using [Microsoft Entra ID](#). For more information, see [Pool network configuration](#).

There are two ways to authenticate a Batch application using Microsoft Entra ID:

- **Integrated authentication.** A batch application using Microsoft Entra accounts can use the account to gain resources to data stores and other resources.
- **Service principal.** Microsoft Entra service principals define access policy and permissions for users and applications. A service principal provides authentication to users using a secret key tied to that application. This allows authenticating an unattended application with a secret key. A service principal defines the policy and permissions for an application to represent the application when accessing resources at runtime. [Learn more here](#).

For more information on security in batch processing with Microsoft Entra ID, [see this article](#).

The Batch service can also authenticate with a shared key. The authentication service requires two header values to be added to the HTTP request, data and authorization. See [here for more](#) on shared key authentication.

Cost optimization

There's no charge for using Azure Batch. You only pay for the underlying resources consumed, such as virtual machine uptime, storage, and networking. However, the compute node VMs still cost money when sitting idle, so it's a good idea to deprovision machines when they're no longer needed. This is often done by deleting the pool containing them.

When creating a pool, you can specify which types of compute nodes you want and the number of each. The two types of compute nodes are as follows:

Dedicated compute nodes are reserved for your workloads. They're more expensive than low-priority nodes, but they're guaranteed to never be preempted.

Low-priority compute nodes take advantage of surplus capacity in Azure to run Batch workloads. Low-priority nodes are less expensive per hour than dedicated nodes, and enable workloads requiring a lot of compute power. For more information, see [Use low-priority VMs with Batch](#).

Dedicated and low-priority nodes may exist in the same pool.

For pricing information for both low-priority and dedicated compute nodes, see [Batch Pricing](#).

When using the Batch Diagnostic Logging service, the data emitted to Azure storage incurs a cost. This is storage data like any other data, and pricing is impacted by the amount of diagnostic data retained.

Getting started

While there are many places to get started with a complex domain like Batch computing for risk grid computing, here are some logical starting points to better understand the Batch technology.

[The Azure Batch documentation](#) is a great place to start. The documentation includes portal examples, API references, and step-by-step tutorials with code examples. The Azure Batch sample applications are also freely available [on GitHub](#).

Below are some quick tutorials to help you build a simple application to create and run batch compute jobs. Options for building the application are as follows:

- [Batch .NET API](#)
- [Batch SDK for Python](#)
- [Batch SDK for Node.js](#)
- [Batch management with PowerShell](#)
- [Batch management with the Azure CLI](#)

Consider launching a proof-of-concept initiative. What will your approach be for data ingestion into Azure? Will you use a hybrid network or upload data via an SDK or REST interface? If you're considering a hybrid network, consider launching a pilot to put this in place.

Evaluate the size of your Batch compute jobs, and then select the right scaling solution. [Autoscaling formulas](#) enable complex scheduling scenarios, while simpler scenarios are achievable by using the Azure portal.

Components

- [Azure Batch](#) provides capabilities to run large-scale parallel processing jobs in the cloud.
- [Microsoft Entra ID](#) is a multi-tenant, cloud-based directory, and identity management service combining core directory services, application access management, and identity protection into a single solution.
- [Automatic scaling formulas](#) are scripts uploaded to the batch processing scheduler for fine grained control of Batch scaling behaviors.
- [Batch Diagnostics Logging](#) is a feature of Azure Batch enabling creation of a detailed log from your Batch runs and the events generated. Logs are stored in Azure Storage.
- [Batch Explorer](#) is a standalone application for Batch monitoring and management available Windows, macOS, and Linux.
- [ExpressRoute](#) is a high speed and reliability hybrid network solution for joining on-premises and Azure networks.
- [Azure VPN Gateway](#) is a hybrid network solution using the Internet for joining on-premises and Azure networks.

Conclusion

This document provided an overview of technical solutions and considerations when using Azure Batch for risk grid computing for banking. The article covered a lot of ground from the definition of Azure Batch to [networking options](#), and even cost considerations.

Contributors

This article is maintained by Microsoft. It was originally written by the following contributors.

Principal authors:

- [David Starr](#) | Principal Solutions Architect

Next steps

When considering moving forward in evaluating Azure Batch for risk grid computing, [this page](#) is a good resource for getting started. It provides sample guided tutorials for parallel file processing, which is inherent in risk grid computing. Tutorials are provided using the Azure Portal, Azure CLI, .NET, and Python.

Product documentation:

- [What is Azure Batch?](#)
- [What is Microsoft Entra ID?](#)
- [What is Azure ExpressRoute?](#)
- [What is VPN Gateway?](#)

Related resources

- [Risk grid computing in banking](#)
- [HPC risk analysis template](#)
- [Actuarial risk analysis and financial modeling](#)

Azure security baseline for Virtual Machine Scale Sets

Article • 09/20/2023

This security baseline applies guidance from the [Microsoft cloud security benchmark version 1.0](#) to Virtual Machine Scale Sets. The Microsoft cloud security benchmark provides recommendations on how you can secure your cloud solutions on Azure. The content is grouped by the security controls defined by the Microsoft cloud security benchmark and the related guidance applicable to Virtual Machine Scale Sets.

You can monitor this security baseline and its recommendations using Microsoft Defender for Cloud. Azure Policy definitions will be listed in the Regulatory Compliance section of the Microsoft Defender for Cloud portal page.

When a feature has relevant Azure Policy Definitions, they are listed in this baseline to help you measure compliance with the Microsoft cloud security benchmark controls and recommendations. Some recommendations may require a paid Microsoft Defender plan to enable certain security scenarios.

ⓘ Note

Features not applicable to Virtual Machine Scale Sets have been excluded. To see how Virtual Machine Scale Sets completely maps to the Microsoft cloud security benchmark, see the [full Virtual Machine Scale Sets security baseline mapping file ↗](#).

Security profile

The security profile summarizes high-impact behaviors of Virtual Machine Scale Sets, which may result in increased security considerations.

Service Behavior Attribute	Value
Product Category	Compute
Customer can access HOST / OS	Full Access
Service can be deployed into customer's virtual network	True
Stores customer content at rest	True

Network security

For more information, see the [Microsoft cloud security benchmark: Network security](#).

NS-1: Establish network segmentation boundaries

Features

Virtual Network Integration

Description: Service supports deployment into customer's private Virtual Network (VNet). [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

Reference: [Virtual networks and virtual machines in Azure](#)

Network Security Group Support

Description: Service network traffic respects Network Security Groups rule assignment on its subnets. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use network security groups (NSG) to restrict or monitor traffic by port, protocol, source IP address, or destination IP address. Create NSG rules to restrict your service's open ports (such as preventing management ports from being accessed from untrusted networks). Be aware that by default, NSGs deny all inbound traffic but allow traffic from virtual network and Azure Load Balancers.

When you create an Azure virtual machine (VM), you must create a virtual network or use an existing virtual network and configure the VM with a subnet. Ensure that all deployed subnets have a Network Security Group applied with network access controls specific to your applications trusted ports and sources.

Reference: [Network security groups](#)

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Adaptive network hardening recommendations should be applied on internet facing virtual machines ↗	Azure Security Center analyzes the traffic patterns of Internet facing virtual machines and provides Network Security Group rule recommendations that reduce the potential attack surface	AuditIfNotExists, Disabled	3.0.0 ↗
All network ports should be restricted on network security groups associated to your virtual machine ↗	Azure Security Center has identified some of your network security groups' inbound rules to be too permissive. Inbound rules should not allow access from 'Any' or 'Internet' ranges. This can potentially enable attackers to target your resources.	AuditIfNotExists, Disabled	3.0.0 ↗
Internet-facing virtual machines should be protected with network security groups ↗	Protect your virtual machines from potential threats by restricting access to them with network security groups (NSG). Learn more about controlling traffic with NSGs at https://aka.ms/nsg-doc ↗	AuditIfNotExists, Disabled	3.0.0 ↗
Non-internet-facing virtual machines should be protected with network security groups ↗	Protect your non-internet-facing virtual machines from potential threats by restricting access with network security groups (NSG). Learn more about controlling traffic with NSGs at https://aka.ms/nsg-doc ↗	AuditIfNotExists, Disabled	3.0.0 ↗

NS-2: Secure cloud services with network controls

Features

Disable Public Network Access

Description: Service supports disabling public network access either through using service-level IP ACL filtering rule (not NSG or Azure Firewall) or using a 'Disable Public

Network Access' toggle switch. [Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Services installed with the operating system may be used to provide network filtering to disabled public network access.

Configuration Guidance: There is no current Microsoft guidance for this feature configuration. Please review and determine if your organization wants to configure this security feature.

Identity management

For more information, see the [Microsoft cloud security benchmark: Identity management](#).

IM-1: Use centralized identity and authentication system

Features

Azure AD Authentication Required for Data Plane Access

Description: Service supports using Azure AD authentication for data plane access.

[Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Azure Active Directory (Azure AD) as the default authentication method to control your data plane access. Azure AD protects data by using strong encryption for data at rest and in transit. Azure AD also salts, hashes, and securely stores user credentials. You can use managed identities to authenticate to any service that supports Azure AD authentication, including Key Vault, without any credentials in your code. Your code that's running on a virtual machine, can use its managed identity to request access tokens for services that support Azure AD authentication.

Reference: [Azure AD join implementation](#)

Local Authentication Methods for Data Plane Access

Description: Local authentications methods supported for data plane access, such as a local username and password. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Feature notes: Avoid the usage of local authentication methods or accounts, these should be disabled wherever possible. Instead use Azure AD to authenticate where possible.

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

IM-3: Manage application identities securely and automatically

Features

Managed Identities

Description: Data plane actions support authentication using managed identities. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Azure managed identities instead of service principals when possible, which can authenticate to Azure services and resources that support Azure Active Directory (Azure AD) authentication. Managed identity credentials are fully managed, rotated, and protected by the platform, avoiding hard-coded credentials in source code or configuration files.

Reference: [Managed identities for Azure resources](#)

Service Principals

Description: Data plane supports authentication using service principals. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Service principals may be used by applications running in Virtual Machine Scale Sets.

Configuration Guidance: There is no current Microsoft guidance for this feature configuration. Please review and determine if your organization wants to configure this security feature.

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity ↗	The Guest Configuration extension requires a system assigned managed identity. Azure virtual machines in the scope of this policy will be non-compliant when they have the Guest Configuration extension installed but do not have a system assigned managed identity. Learn more at https://aka.ms/gcpol ↗	AuditIfNotExists, Disabled	1.0.1 ↗

IM-8: Restrict the exposure of credential and secrets

Features

Service Credential and Secrets Support Integration and Storage in Azure Key Vault

Description: Data plane supports native use of Azure Key Vault for credential and secrets store. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Within the data plane or operating system, services may call Azure Key Vault for credentials or secrets.

Configuration Guidance: Ensure that secrets and credentials are stored in secure locations such as Azure Key Vault, instead of embedding them into code or configuration files.

Privileged access

For more information, see the [Microsoft cloud security benchmark: Privileged access](#).

PA-1: Separate and limit highly privileged/administrative users

Features

Local Admin Accounts

Description: Service has the concept of a local administrative account. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Feature notes: Avoid the usage of local authentication methods or accounts, these should be disabled wherever possible. Instead use Azure AD to authenticate where possible.

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

Reference: [Create virtual machines in a scale set using Azure portal](#)

PA-7: Follow just enough administration (least privilege) principle

Features

Azure RBAC for Data Plane

Description: Azure Role-Based Access Control (Azure RBAC) can be used to manage access to service's data plane actions. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Azure role-based access control (Azure RBAC) to manage Azure resource access through built-in role assignments. Azure RBAC roles can be assigned to users, groups, service principals, and managed identities.

Reference: [Built-in Role for Virtual Machine Contributor](#)

PA-8: Determine access process for cloud provider support

Features

Customer Lockbox

Description: Customer Lockbox can be used for Microsoft support access. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: In support scenarios where Microsoft needs to access your data, use Customer Lockbox to review, then approve or reject each of Microsoft's data access requests.

Data protection

For more information, see the [Microsoft cloud security benchmark: Data protection](#).

DP-1: Discover, classify, and label sensitive data

Features

Sensitive Data Discovery and Classification

Description: Tools (such as Azure Purview or Azure Information Protection) can be used for data discovery and classification in the service. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

DP-2: Monitor anomalies and threats targeting sensitive data

Features

Data Leakage/Loss Prevention

Description: Service supports DLP solution to monitor sensitive data movement (in customer's content). [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

DP-3: Encrypt sensitive data in transit

Features

Data in Transit Encryption

Description: Service supports data in-transit encryption for data plane. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Certain communication protocols such as SSH are encrypted by default. However, services such as RDP or HTTP must be configured to use TLS for encryption.

Configuration Guidance: Enable secure transfer in services where there is a native data in transit encryption feature built in. Enforce HTTPS on any web applications and services and ensure TLS v1.2 or later is used. Legacy versions such as SSL 3.0, TLS v1.0

should be disabled. For remote management of Virtual Machines, use SSH (for Linux) or RDP/TLS (for Windows) instead of an unencrypted protocol.

Reference: [In-transit encryption in VMs](#)

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Windows web servers should be configured to use secure communication protocols ↗	To protect the privacy of information communicated over the Internet, your web servers should use the latest version of the industry-standard cryptographic protocol, Transport Layer Security (TLS). TLS secures communications over a network by using security certificates to encrypt a connection between machines.	AuditIfNotExists, Disabled	4.1.0 ↗

DP-4: Enable data at rest encryption by default

Features

Data at Rest Encryption Using Platform Keys

Description: Data at-rest encryption using platform keys is supported, any customer content at rest is encrypted with these Microsoft managed keys. [Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Feature notes: In addition to the standard encryption with platform managed keys, high security sensitive customers who are concerned of the risk associated with any particular encryption algorithm, implementation, or key being compromised can now opt for additional layer of encryption using a different encryption algorithm/mode at the infrastructure layer using platform managed encryption keys and customer managed keys. This new layer can be applied to persisted OS and data disks, snapshots, and images, all of which will be encrypted at rest with double encryption.

For more information, please visit: [Double encryption at rest](#).

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

Reference: [Azure Disk Encryption for Virtual Machine Scale Sets](#)

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: Linux virtual machines should enable Azure Disk Encryption or EncryptionAtHost.	<p>By default, a virtual machine's OS and data disks are encrypted-at-rest using platform-managed keys; temp disks and data caches aren't encrypted, and data isn't encrypted when flowing between compute and storage resources. Use Azure Disk Encryption or EncryptionAtHost to encrypt all this data. Visit https://aka.ms/diskencryptioncomparison to compare encryption offerings. This policy requires two prerequisites to be deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.</p>	AuditIfNotExists, Disabled	1.1.0-preview
[Preview]: Windows virtual machines should enable Azure Disk Encryption or EncryptionAtHost.	<p>By default, a virtual machine's OS and data disks are encrypted-at-rest using platform-managed keys; temp disks and data caches aren't encrypted, and data isn't encrypted when flowing between compute and storage resources. Use Azure Disk Encryption or EncryptionAtHost to encrypt all this data. Visit https://aka.ms/diskencryptioncomparison to compare encryption offerings. This policy requires two prerequisites to be deployed to the policy assignment scope. For details, visit https://aka.ms/gcpol.</p>	AuditIfNotExists, Disabled	1.1.0-preview
Virtual machines and virtual machine scale sets should have encryption at host enabled	<p>Use encryption at host to get end-to-end encryption for your virtual machine and virtual machine scale set data. Encryption at host enables encryption at rest for your temporary disk and OS/data disk caches. Temporary and ephemeral OS disks are encrypted with platform-managed keys when encryption at host is enabled.</p>	Audit, Deny, Disabled	1.0.0

Name (Azure portal)	Description	Effect(s) (GitHub)	Version
	OS/data disk caches are encrypted at rest with either customer-managed or platform-managed key, depending on the encryption type selected on the disk. Learn more at https://aka.ms/vm-hbe .		
Virtual machines should encrypt temp disks, caches, and data flows between Compute and Storage resources	By default, a virtual machine's OS and data disks are encrypted-at-rest using platform-managed keys. Temp disks, data caches and data flowing between compute and storage aren't encrypted. Disregard this recommendation if: 1. using encryption-at-host, or 2. server-side encryption on Managed Disks meets your security requirements. Learn more in: Server-side encryption of Azure Disk Storage: https://aka.ms/disksse , Different disk encryption offerings : https://aka.ms/diskencryptioncomparison	AuditIfNotExists, 2.0.3 Disabled	

DP-5: Use customer-managed key option in data at rest encryption when required

Features

Data at Rest Encryption Using CMK

Description: Data at-rest encryption using customer-managed keys is supported for customer content stored by the service. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: If required for regulatory compliance, define the use case and service scope where encryption using customer-managed keys are needed. Enable and implement data at rest encryption using customer-managed key for those services.

Virtual disks on Virtual Machines (VM) are encrypted at rest using either Server-side encryption or Azure disk encryption (ADE). Azure Disk Encryption leverages the DM-Crypt feature of Linux to encrypt managed disks with customer-managed keys within

the guest VM. Server-side encryption with customer-managed keys improves on ADE by enabling you to use any OS types and images for your VMs by encrypting data in the Storage service.

When you specify a customer-managed key, that key is used to protect and control access to the key that encrypts your data. Customer-managed keys offer greater flexibility to manage access controls. You must use either Azure Key Vault or Azure Key Vault Managed Hardware Security Module (HSM) to store your customer-managed keys.

You can either import [your RSA keys](#) to your Key Vault or generate new RSA keys in Azure Key Vault. Azure managed disks handles the encryption and decryption in a fully transparent fashion using envelope encryption. It encrypts data using an AES 256 based data encryption key (DEK), which is, in turn, protected using your keys. The Storage service generates data encryption keys and encrypts them with customer-managed keys using RSA encryption. The envelope encryption allows you to rotate (change) your keys periodically as per your compliance policies without impacting your VMs. When you rotate your keys, the Storage service re-encrypts the data encryption keys with the new customer-managed keys.

Managed Disks and the Key Vault or managed HSM must be in the same Azure region, but they can be in different subscriptions. They must also be in the same Azure Active Directory (Azure AD) tenant, unless you're [encrypting managed disks with cross-tenant customer-managed keys](#).

Reference: [Creating and configuring a key vault for Azure Disk Encryption](#)

DP-6: Use a secure key management process

Features

Key Management in Azure Key Vault

Description: The service supports Azure Key Vault integration for any customer keys, secrets, or certificates. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Azure Key Vault to create and control the life cycle of your encryption keys, including key generation, distribution, and storage. Rotate and revoke your keys in Azure Key Vault and your service based on a defined schedule or when

there is a key retirement or compromise. When there is a need to use customer-managed key (CMK) in the workload, service, or application level, ensure you follow the best practices for key management: Use a key hierarchy to generate a separate data encryption key (DEK) with your key encryption key (KEK) in your key vault. Ensure keys are registered with Azure Key Vault and referenced via key IDs from the service or application. If you need to bring your own key (BYOK) to the service (such as importing HSM-protected keys from your on-premises HSMs into Azure Key Vault), follow recommended guidelines to perform initial key generation and key transfer.

Reference: [Creating and configuring a key vault for Azure Disk Encryption](#)

Asset management

For more information, see the [Microsoft cloud security benchmark: Asset management](#).

AM-2: Use only approved services

Features

Azure Policy Support

Description: Service configurations can be monitored and enforced via Azure Policy.
[Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Azure Policy can be used to define the desired behavior for your organization's Windows VMs and Linux VMs. By using policies, an organization can enforce various conventions and rules throughout the enterprise and define and implement standard security configurations for Azure Virtual Machine Scale Sets. Enforcement of the desired behavior can help mitigate risk while contributing to the success of the organization.

Reference: [Built-in Azure Policy Definitions for Virtual Machine Scale Sets](#)

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Virtual machines should be migrated to new Azure Resource Manager resources ↴	Use new Azure Resource Manager for your virtual machines to provide security enhancements such as: stronger access control (RBAC), better auditing, Azure Resource Manager based deployment and governance, access to managed identities, access to key vault for secrets, Azure AD-based authentication and support for tags and resource groups for easier security management	Audit, Deny, Disabled	1.0.0 ↴

AM-5: Use only approved applications in virtual machine

Features

Microsoft Defender for Cloud - Adaptive Application Controls

Description: Service can limit what customer applications run on the virtual machine using Adaptive Application Controls in Microsoft Defender for Cloud. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Microsoft Defender for Cloud adaptive application controls to discover applications running on virtual machines (VMs) and generate an application allow list to mandate which approved applications can run in the VM environment.

Reference: [Use adaptive application controls to reduce your machines' attack surfaces](#)

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Adaptive application controls for defining safe applications	Enable application controls to define the list of known-safe applications running on your machines, and alert you when other applications run. This helps harden your machines against malware. To simplify the	AuditIfNotExists, Disabled	3.0.0 ↴

Name	Description	Effect(s)	Version
(Azure portal)			(GitHub)
should be enabled on your machines ↗	process of configuring and maintaining your rules, Security Center uses machine learning to analyze the applications running on each machine and suggest the list of known-safe applications.		
Allowlist rules in your adaptive application control policy should be updated ↗	Monitor for changes in behavior on groups of machines configured for auditing by Azure Security Center's adaptive application controls. Security Center uses machine learning to analyze the running processes on your machines and suggest a list of known-safe applications. These are presented as recommended apps to allow in adaptive application control policies.	AuditIfNotExists, Disabled	3.0.0 ↗

Logging and threat detection

For more information, see the [Microsoft cloud security benchmark: Logging and threat detection](#).

LT-1: Enable threat detection capabilities

Features

Microsoft Defender for Service / Product Offering

Description: Service has an offering-specific Microsoft Defender solution to monitor and alert on security issues. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Defender for Servers extends protection to your Windows and Linux machines running in Azure. Defender for Servers integrates with Microsoft Defender for Endpoint to provide endpoint detection and response (EDR), and also provides a host of additional threat protection features, such as security baselines and OS level assessments, vulnerability assessment scanning, adaptive application controls (AAC), file integrity monitoring (FIM), and more.

Reference: [Overview of Microsoft Defender for Servers](#)

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Windows Defender Exploit Guard should be enabled on your machines ↗	Windows Defender Exploit Guard uses the Azure Policy Guest Configuration agent. Exploit Guard has four components that are designed to lock down devices against a wide variety of attack vectors and block behaviors commonly used in malware attacks while enabling enterprises to balance their security risk and productivity requirements (Windows only).	AuditIfNotExists, Disabled	2.0.0 ↗

LT-4: Enable logging for security investigation

Features

Azure Resource Logs

Description: Service produces resource logs that can provide enhanced service-specific metrics and logging. The customer can configure these resource logs and send them to their own data sink like a storage account or log analytics workspace. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Azure Monitor starts automatically collecting metric data for your virtual machine host when you create the VM. To collect logs and performance data from the guest operating system of the virtual machine, though, you must install the Azure Monitor agent. You can install the agent and configure collection using either [VM insights](#) or by [creating a data collection rule](#).

Reference: [Log Analytics agent overview](#)

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: Network traffic data collection agent should be installed on Linux virtual machines ↗	Security Center uses the Microsoft Dependency agent to collect network traffic data from your Azure virtual machines to enable advanced network protection features such as traffic visualization on the network map, network hardening recommendations and specific network threats.	AuditIfNotExists, Disabled	1.0.2-preview ↗
[Preview]: Network traffic data collection agent should be installed on Windows virtual machines ↗	Security Center uses the Microsoft Dependency agent to collect network traffic data from your Azure virtual machines to enable advanced network protection features such as traffic visualization on the network map, network hardening recommendations and specific network threats.	AuditIfNotExists, Disabled	1.0.2-preview ↗

Posture and vulnerability management

For more information, see the [Microsoft cloud security benchmark: Posture and vulnerability management](#).

PV-3: Define and establish secure configurations for compute resources

Features

Azure Automation State Configuration

Description: Azure Automation State Configuration can be used to maintain the security configuration of the operating system. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Azure Automation State Configuration to maintain the security configuration of the operating system. Azure Automation State Configuration is an Azure configuration management service that allows you to write, manage, and compile PowerShell Desired State Configuration (DSC) configurations for nodes.

Azure Automation State Configuration provides several advantages over the use of DSC outside of Azure. This service enables scalability across thousands of machines quickly and easily from a central, secure location. You can easily enable machines, assign them declarative configurations, and view reports showing each machine's compliance with the desired state you specify.

Reference: [Using Virtual Machine Scale Sets with the Azure DSC Extension](#)

Azure Policy Guest Configuration Agent

Description: Azure Policy guest configuration agent can be installed or deployed as an extension to compute resources. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Microsoft Defender for Cloud and Azure Policy guest configuration agent to regularly assess and remediate configuration deviations on your virtual machines.

Reference: [Understand the guest configuration feature of Azure Policy](#)

Custom VM Images

Description: Service supports using user-supplied VM images or pre-built images from the marketplace with certain baseline configurations pre-applied. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use a pre-configured hardened image from a trusted supplier such as Microsoft or build a desired secure configuration baseline into the VM image template

Reference: [Create and use a custom image for virtual machine scale sets with Azure PowerShell](#)

PV-4: Audit and enforce secure configurations for compute resources

Features

Trusted Launch Virtual Machine

Description: Trusted Launch protects against advanced and persistent attack techniques by combining infrastructure technologies like secure boot, vTPM, and integrity monitoring. Each technology provides another layer of defense against sophisticated threats. Trusted launch allows the secure deployment of virtual machines with verified boot loaders, OS kernels, and drivers, and securely protects keys, certificates, and secrets in the virtual machines. Trusted launch also provides insights and confidents of the entire boot chain's integrity and ensures workloads are trusted and verifiable. Trusted launch is integrated with Microsoft Defender for Cloud to ensure VMs are properly configured, by remotely attesting VM is booted in a healthy way. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature note: Trusted launch is available for generation 2 VMs. Trusted launch requires the creation of new virtual machines. You can't enable trusted launch on existing virtual machines that were initially created without it.

Configuration Guidance: Trusted launch may be enabled during the deployment of the VM. Enable all three - Secure Boot, vTPM, and integrity boot monitoring to ensure the best security posture for the virtual machine. Please note that there are a few prerequisites including onboarding your subscription to Microsoft Defender for Cloud, assigning certain Azure Policy initiatives, and configuring firewall policies.

Reference: [Deploy a VM with trusted launch enabled](#)

PV-5: Perform vulnerability assessments

Features

Vulnerability Assessment using Microsoft Defender

Description: Service can be scanned for vulnerability scan using Microsoft Defender for Cloud or other Microsoft Defender services embedded vulnerability assessment capability (including Microsoft Defender for server, container registry, App Service, SQL, and DNS). [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Follow recommendations from Microsoft Defender for Cloud for performing vulnerability assessments on your Azure virtual machines.

Reference: [Overview of Microsoft Defender for Servers](#)

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
A vulnerability assessment solution should be enabled on your virtual machines ↗	Audits virtual machines to detect whether they are running a supported vulnerability assessment solution. A core component of every cyber risk and security program is the identification and analysis of vulnerabilities. Azure Security Center's standard pricing tier includes vulnerability scanning for your virtual machines at no extra cost. Additionally, Security Center can automatically deploy this tool for you.	AuditIfNotExists, Disabled	3.0.0 ↗
Machines should have secret findings resolved ↗	Audits virtual machines to detect whether they contain secret findings from the secret scanning solutions on your virtual machines.	AuditIfNotExists, Disabled	1.0.2 ↗

PV-6: Rapidly and automatically remediate vulnerabilities

Features

Azure Automation Update Management

Description: Service can use Azure Automation Update Management to deploy patches and updates automatically. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Feature notes: Microsoft offers other capabilities to help you manage updates for your Azure VMs or Azure virtual machine scale sets that you should consider as part of your overall update management strategy.

If you are interested in automatically assessing and updating your Azure virtual machines to maintain security compliance with Critical and Security updates released each month, review Automatic VM guest patching. This is an alternative update management solution for your Azure VMs to auto-update them during off-peak hours, including VMs within an availability set, compared to managing update deployments to those VMs from Update Management in Azure Automation.

If you manage Azure virtual machine scale sets, review how to perform automatic OS image upgrades to safely and automatically upgrade the OS disk for all instances in the scale set.

For more information, please visit: [Azure Virtual Machine Scale Set automatic OS image upgrades](#).

Configuration Guidance: This feature is not supported to secure this service.

Azure Guest Patching Service

Description: Service can use Azure Guest Patching to deploy patches and updates automatically. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Services can leverage the different update mechanisms such as [Auto OS Image Upgrades](#) and [Auto Guest Patching](#). The capabilities are recommended to apply the latest security and critical updates to your Virtual Machine's Guest OS by following the Safe Deployment Principles.

Auto Guest Patching allows you to automatically assess and update your Azure virtual machines to maintain security compliance with Critical and Security updates released each month. Updates are applied during off-peak hours, including VMs within an availability set. This capability is available for VMSS Flexible Orchestration, with future support on the roadmap for Uniform Orchestration.

If you run a stateless workload, Auto OS Image Upgrades are ideal to apply the latest update for your VMSS Uniform. With rollback capability, these updates are compatible with Marketplace or Custom images. Future rolling upgrade support on the roadmap for Flexible Orchestration.

Reference: [Automatic VM Guest Patching for Azure VMs](#)

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
[Preview]: Machines should be configured to periodically check for missing system updates ↗	To ensure periodic assessments for missing system updates are triggered automatically every 24 hours, the AssessmentMode property should be set to 'AutomaticByPlatform'. Learn more about AssessmentMode property for Windows: https://aka.ms/computevm-windowspatchassessmentmode , for Linux: https://aka.ms/computevm-linuxpatchassessmentmode .	Audit, Deny, Disabled	3.3.0-preview ↗
[Preview]: System updates should be installed on your machines (powered by Update Center) ↗	Your machines are missing system, security, and critical updates. Software updates often include critical patches to security holes. Such holes are frequently exploited in malware attacks so it's vital to keep your software updated. To install all outstanding patches and secure your machines, follow the remediation steps.	AuditIfNotExists, Disabled	1.0.0-preview ↗
SQL servers on machines should have vulnerability findings resolved ↗	SQL vulnerability assessment scans your database for security vulnerabilities, and exposes any deviations from best practices such as misconfigurations, excessive permissions, and unprotected sensitive data. Resolving the vulnerabilities found can greatly improve your database security posture.	AuditIfNotExists, Disabled	1.0.0 ↗
System updates on virtual machine scale sets should be installed ↗	Audit whether there are any missing system security updates and critical updates that should be installed to ensure that your Windows and Linux virtual machine scale sets are secure.	AuditIfNotExists, Disabled	3.0.0 ↗

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
System updates should be installed on your machines ↗	Missing security system updates on your servers will be monitored by Azure Security Center as recommendations	AuditIfNotExists, Disabled	4.0.0 ↗
Vulnerabilities in container security configurations should be remediated ↗	Audit vulnerabilities in security configuration on machines with Docker installed and display as recommendations in Azure Security Center.	AuditIfNotExists, Disabled	3.0.0 ↗
Vulnerabilities in security configuration on your machines should be remediated ↗	Servers which do not satisfy the configured baseline will be monitored by Azure Security Center as recommendations	AuditIfNotExists, Disabled	3.1.0 ↗
Vulnerabilities in security configuration on your virtual machine scale sets should be remediated ↗	Audit the OS vulnerabilities on your virtual machine scale sets to protect them from attacks.	AuditIfNotExists, Disabled	3.0.0 ↗

Endpoint security

For more information, see the [Microsoft cloud security benchmark: Endpoint security](#).

ES-1: Use Endpoint Detection and Response (EDR)

Features

EDR Solution

Description: Endpoint Detection and Response (EDR) feature such as Azure Defender for servers can be deployed into the endpoint. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Azure Defender for servers (with Microsoft Defender for Endpoint integrated) provides EDR capability to prevent, detect, investigate, and respond to advanced threats. Use Microsoft Defender for Cloud to deploy Azure Defender for servers for your endpoint and integrate the alerts to your SIEM solution such as Azure Sentinel.

Reference: [Integrated license for Microsoft Defender for Endpoint](#)

ES-2: Use modern anti-malware software

Features

Anti-Malware Solution

Description: Anti-malware feature such as Microsoft Defender Antivirus, Microsoft Defender for Endpoint can be deployed on the endpoint. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: For Windows Server 2016 and above, Microsoft Defender for Antivirus is installed by default. For Windows Server 2012 R2 and above, customers can install SCEP (System Center Endpoint Protection). For Linux, customers can have the choice of installing Microsoft Defender for Linux. Alternatively, customers also have the choice of installing third-party anti-malware products.

Reference: [Microsoft Antimalware for Azure Cloud Services and Virtual Machines](#)

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Endpoint protection health issues	Resolve endpoint protection health issues on your virtual machines to protect them from latest threats and vulnerabilities. Azure Security Center supported	AuditIfNotExists, Disabled	1.0.0 ↗

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
should be resolved on your machines ↗	<p>endpoint protection solutions are documented here</p> <ul style="list-style-type: none"> - https://docs.microsoft.com/azure/security-center/security-center-services?tabs=features-windows#supported-endpoint-protection-solutions. <p>Endpoint protection assessment is documented here</p> <ul style="list-style-type: none"> - https://docs.microsoft.com/azure/security-center/security-center-endpoint-protection. 		
Endpoint protection should be installed on your machines ↗	<p>To protect your machines from threats and vulnerabilities, install a supported endpoint protection solution.</p>	AuditIfNotExists, Disabled	1.0.0 ↗
Endpoint protection solution should be installed on virtual machine scale sets ↗	<p>Audit the existence and health of an endpoint protection solution on your virtual machines scale sets, to protect them from threats and vulnerabilities.</p>	AuditIfNotExists, Disabled	3.0.0 ↗
Monitor missing Endpoint Protection in Azure Security Center ↗	<p>Servers without an installed Endpoint Protection agent will be monitored by Azure Security Center as recommendations</p>	AuditIfNotExists, Disabled	3.0.0 ↗
Windows Defender Exploit Guard should be enabled on your machines ↗	<p>Windows Defender Exploit Guard uses the Azure Policy Guest Configuration agent. Exploit Guard has four components that are designed to lock down devices against a wide variety of attack vectors and block behaviors commonly used in malware attacks while enabling enterprises to balance their security risk and productivity requirements (Windows only).</p>	AuditIfNotExists, Disabled	2.0.0 ↗

ES-3: Ensure anti-malware software and signatures are updated

Features

Anti-Malware Solution Health Monitoring

Description: Anti-malware solution provides health status monitoring for platform, engine, and automatic signature updates. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Configure your anti-malware solution to ensure the platform, engine and signatures are updated rapidly and consistently and their status can be monitored.

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Endpoint protection health issues should be resolved on your machines ↗	Resolve endpoint protection health issues on your virtual machines to protect them from latest threats and vulnerabilities. Azure Security Center supported endpoint protection solutions are documented here - https://docs.microsoft.com/azure/security-center/security-center-services?tabs=features-windows#supported-endpoint-protection-solutions . Endpoint protection assessment is documented here - https://docs.microsoft.com/azure/security-center/security-center-endpoint-protection .	AuditIfNotExists, 1.0.0 ↗ Disabled	

Backup and recovery

For more information, see the [Microsoft cloud security benchmark: Backup and recovery](#).

BR-1: Ensure regular automated backups

Features

Azure Backup

Description: The service can be backed up by the Azure Backup service. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Supported for VMSS Flex and not VMSS Uniform

Configuration Guidance: Enable Azure Backup and target Azure Virtual Machines (VM), as well as the desired frequency and retention periods. This includes complete system state backup. If you are using Azure disk encryption, Azure VM backup automatically handles the backup of customer-managed keys. For Azure Virtual Machines, you can use Azure Policy to enable automatic backups.

Reference: [How to take a snapshot of a virtual machine scale set instance and managed disk](#)

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Azure Backup should be enabled for Virtual Machines ↗	Ensure protection of your Azure Virtual Machines by enabling Azure Backup. Azure Backup is a secure and cost effective data protection solution for Azure.	AuditIfNotExists, Disabled	3.0.0 ↗

Next steps

- See the [Microsoft cloud security benchmark overview](#)
- Learn more about [Azure security baselines](#)

Azure security baseline for Virtual Machines - Linux Virtual Machines

Article • 09/20/2023

This security baseline applies guidance from the [Microsoft cloud security benchmark version 1.0](#) to Virtual Machines - Linux Virtual Machines. The Microsoft cloud security benchmark provides recommendations on how you can secure your cloud solutions on Azure. The content is grouped by the security controls defined by the Microsoft cloud security benchmark and the related guidance applicable to Virtual Machines - Linux Virtual Machines.

You can monitor this security baseline and its recommendations using Microsoft Defender for Cloud. Azure Policy definitions will be listed in the Regulatory Compliance section of the Microsoft Defender for Cloud portal page.

When a feature has relevant Azure Policy Definitions, they are listed in this baseline to help you measure compliance with the Microsoft cloud security benchmark controls and recommendations. Some recommendations may require a paid Microsoft Defender plan to enable certain security scenarios.

ⓘ Note

Features not applicable to Virtual Machines - Linux Virtual Machines have been excluded. To see how Virtual Machines - Linux Virtual Machines completely maps to the Microsoft cloud security benchmark, see the [full Virtual Machines - Linux Virtual Machines security baseline mapping file ↗](#).

Security profile

The security profile summarizes high-impact behaviors of Virtual Machines - Linux Virtual Machines, which may result in increased security considerations.

Service Behavior Attribute	Value
Product Category	Compute
Customer can access HOST / OS	Full Access
Service can be deployed into customer's virtual network	True
Stores customer content at rest	True

Network security

For more information, see the [Microsoft cloud security benchmark: Network security](#).

NS-1: Establish network segmentation boundaries

Features

Virtual Network Integration

Description: Service supports deployment into customer's private Virtual Network (VNet). [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

Reference: [Virtual networks and virtual machines in Azure](#)

Network Security Group Support

Description: Service network traffic respects Network Security Groups rule assignment on its subnets. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use network security groups (NSG) to restrict or monitor traffic by port, protocol, source IP address, or destination IP address. Create NSG rules to restrict your service's open ports (such as preventing management ports from being accessed from untrusted networks). Be aware that by default, NSGs deny all inbound traffic but allow traffic from virtual network and Azure Load Balancers.

When you create an Azure virtual machine (VM), you must create a virtual network or use an existing virtual network and configure the VM with a subnet. Ensure that all deployed subnets have a Network Security Group applied with network access controls specific to your applications trusted ports and sources.

Reference: [Network security groups](#)

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.ClassicCompute:

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
All network ports should be restricted on network security groups associated to your virtual machine ↗	Azure Security Center has identified some of your network security groups' inbound rules to be too permissive. Inbound rules should not allow access from 'Any' or 'Internet' ranges. This can potentially enable attackers to target your resources.	AuditIfNotExists, Disabled	3.0.0 ↗
Internet-facing virtual machines should be protected with network security groups ↗	Protect your virtual machines from potential threats by restricting access to them with network security groups (NSG). Learn more about controlling traffic with NSGs at https://aka.ms/nsg-doc ↗	AuditIfNotExists, Disabled	3.0.0 ↗
Non-internet-facing virtual machines should be protected with network security groups ↗	Protect your non-internet-facing virtual machines from potential threats by restricting access with network security groups (NSG). Learn more about controlling traffic with NSGs at https://aka.ms/nsg-doc ↗	AuditIfNotExists, Disabled	3.0.0 ↗

Azure Policy built-in definitions - Microsoft.Compute:

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Adaptive network hardening recommendations should be applied on internet facing virtual machines ↗	Azure Security Center analyzes the traffic patterns of Internet facing virtual machines and provides Network Security Group rule recommendations that reduce the potential attack surface	AuditIfNotExists, Disabled	3.0.0 ↗
All network ports should be restricted on network security groups associated to your virtual machine ↗	Azure Security Center has identified some of your network security groups' inbound rules to be too permissive. Inbound rules should not allow access from 'Any' or 'Internet' ranges. This can potentially enable attackers to target your resources.	AuditIfNotExists, Disabled	3.0.0 ↗
Internet-facing virtual machines should be	Protect your virtual machines from potential threats by restricting access	AuditIfNotExists, Disabled	3.0.0 ↗

Name Protected with network security groups (Azure portal)	Description Do the right thing with network security groups (NSG). Learn more about controlling traffic with NSGs at https://aka.ms/nsg-doc	Effect(s)	Version (GitHub)
Non-internet-facing virtual machines should be protected with network security groups	Protect your non-internet-facing virtual machines from potential threats by restricting access with network security groups (NSG). Learn more about controlling traffic with NSGs at https://aka.ms/nsg-doc	AuditIfNotExists, Disabled	3.0.0

NS-2: Secure cloud services with network controls

Features

Disable Public Network Access

Description: Service supports disabling public network access either through using service-level IP ACL filtering rule (not NSG or Azure Firewall) or using a 'Disable Public Network Access' toggle switch. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Services such as iptables or firewalld may be installed in the Linux OS and provide network filtering to disable public access.

Identity management

For more information, see the [Microsoft cloud security benchmark: Identity management](#).

IM-1: Use centralized identity and authentication system

Features

Azure AD Authentication Required for Data Plane Access

Description: Service supports using Azure AD authentication for data plane access. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Azure Active Directory (Azure AD) as the default authentication method to control your data plane access.

Reference: [Log in to a Linux virtual machine in Azure by using Azure AD and OpenSSH](#)

Local Authentication Methods for Data Plane Access

Description: Local authentications methods supported for data plane access, such as a local username and password. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Feature notes: A local administrator account is created by default during the initial deployment of the virtual machine. Avoid the usage of local authentication methods or accounts, these should be disabled wherever possible. Instead use Azure AD to authenticate where possible.

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

IM-3: Manage application identities securely and automatically

Features

Managed Identities

Description: Data plane actions support authentication using managed identities. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Managed identity is traditionally leveraged by Linux VM to authenticate to other services. If the Linux VM supports Azure AD authentication then managed

identity may be supported.

Configuration Guidance: Use Azure managed identities instead of service principals when possible, which can authenticate to Azure services and resources that support Azure Active Directory (Azure AD) authentication. Managed identity credentials are fully managed, rotated, and protected by the platform, avoiding hard-coded credentials in source code or configuration files.

Service Principals

Description: Data plane supports authentication using service principals. [Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Service principals may be used by applications running in the Linux VM.

Configuration Guidance: There is no current Microsoft guidance for this feature configuration. Please review and determine if your organization wants to configure this security feature.

Microsoft Defender for Cloud monitoring

Azure Policy built-in definitions - Microsoft.Compute:

Name (Azure portal)	Description	Effect(s)	Version (GitHub)
Virtual machines' Guest Configuration extension should be deployed with system-assigned managed identity	The Guest Configuration extension requires a system assigned managed identity. Azure virtual machines in the scope of this policy will be non-compliant when they have the Guest Configuration extension installed but do not have a system assigned managed identity. Learn more at https://aka.ms/gcpol	AuditIfNotExists, Disabled	1.0.1

IM-7: Restrict resource access based on conditions

Features

Conditional Access for Data Plane

Description: Data plane access can be controlled using Azure AD Conditional Access Policies. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Use Azure AD as a core authentication platform and a certificate authority to SSH into a Linux VM by using Azure AD and OpenSSH certificate-based authentication. This functionality allows organizations to manage access to VMs with Azure role-based access control (RBAC) and Conditional Access policies.

Configuration Guidance: Define the applicable conditions and criteria for Azure Active Directory (Azure AD) conditional access in the workload. Consider common use cases such as blocking or granting access from specific locations, blocking risky sign-in behavior, or requiring organization-managed devices for specific applications.

Reference: [Log in to a Linux virtual machine in Azure by using Azure AD and OpenSSH](#)

IM-8: Restrict the exposure of credential and secrets

Features

Service Credential and Secrets Support Integration and Storage in Azure Key Vault

Description: Data plane supports native use of Azure Key Vault for credential and secrets store. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Within the data plane or operating system, services may call Azure Key Vault for credentials or secrets.

Configuration Guidance: Ensure that secrets and credentials are stored in secure locations such as Azure Key Vault, instead of embedding them into code or configuration files.

Privileged access

For more information, see the [Microsoft cloud security benchmark: Privileged access](#).

PA-1: Separate and limit highly privileged/administrative users

Features

Local Admin Accounts

Description: Service has the concept of a local administrative account. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Feature notes: Avoid the usage of local authentication methods or accounts, these should be disabled wherever possible. Instead use Azure AD to authenticate where possible.

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

Reference: [Quickstart: Create a Linux virtual machine in the Azure portal](#)

PA-7: Follow just enough administration (least privilege) principle

Features

Azure RBAC for Data Plane

Description: Azure Role-Based Access Control (Azure RBAC) can be used to manage access to service's data plane actions. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Use Azure AD as a core authentication platform and a certificate authority to SSH into a Linux VM by using Azure AD and OpenSSH certificate-based

authentication. This functionality allows organizations to manage access to VMs with Azure role-based access control (RBAC) and Conditional Access policies.

Configuration Guidance: With RBAC, specify who can log in to a VM as a regular user or with administrator privileges. When users join your team, you can update the Azure RBAC policy for the VM to grant access as appropriate. When employees leave your organization and their user accounts are disabled or removed from Azure AD, they no longer have access to your resources.

Reference: [Log in to a Linux virtual machine in Azure by using Azure AD and OpenSSH](#)

PA-8: Determine access process for cloud provider support

Features

Customer Lockbox

Description: Customer Lockbox can be used for Microsoft support access. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: In support scenarios where Microsoft needs to access your data, use Customer Lockbox to review, then approve or reject each of Microsoft's data access requests.

Data protection

For more information, see the [Microsoft cloud security benchmark: Data protection](#).

DP-1: Discover, classify, and label sensitive data

Features

Sensitive Data Discovery and Classification

Description: Tools (such as Azure Purview or Azure Information Protection) can be used for data discovery and classification in the service. [Learn more](#).