



Azure Web App Logging

Azure Web App Logging

You get a set of logging features that are available for the Azure Web App.

The different types of logging that are available are

Application Logging – This captures log messages that are generated by your application code.

Web server logging – This records raw HTTP request data.

Azure Web App Logging

Detailed Error Messages – This stores copies of the .htm error pages that would have been sent to the client browser.

Deployment logging – These are logs when you publish content to an application.

You can also stream logs in real time.

Azure *App Configuration*

Azure App Configuration

Central Service

Provides a central service for managing application settings and feature flags .

Managed

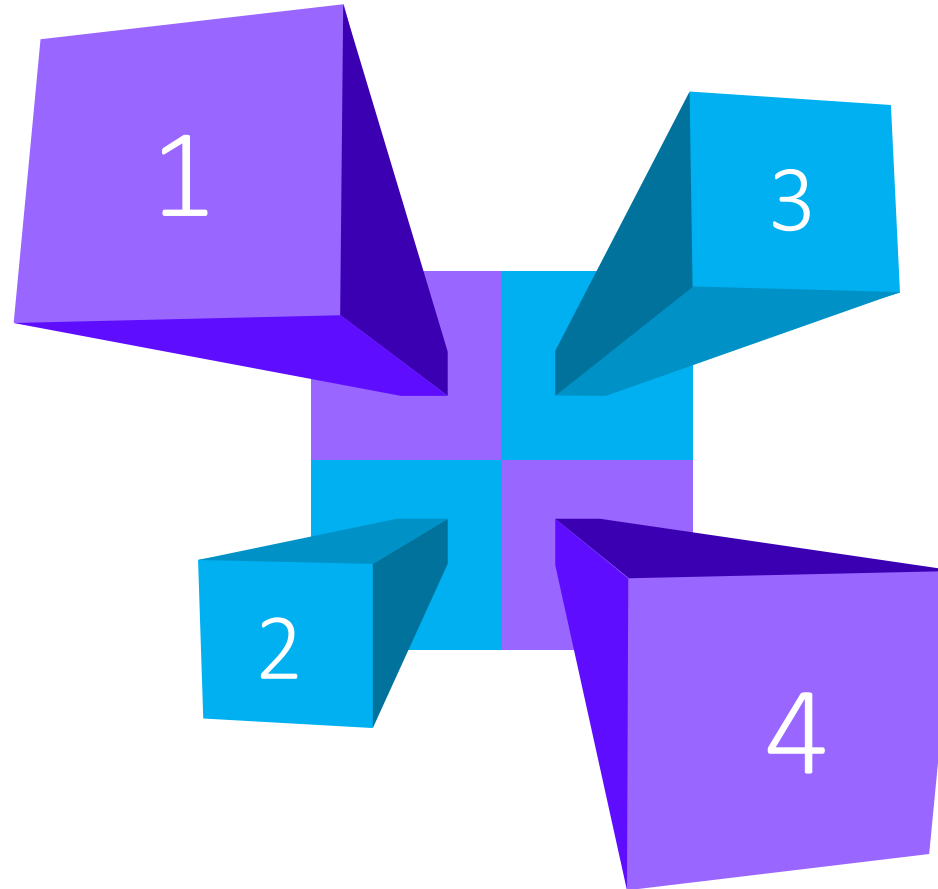
This is a fully managed service.

Encryption

It provides encryption of sensitive information at rest and in transit.

Frameworks

It integrates with popular frameworks.



Azure *Container Instances*

Azure Container Instances

Deployment

Provides a fast and easy way to deploy containers.

Infrastructure

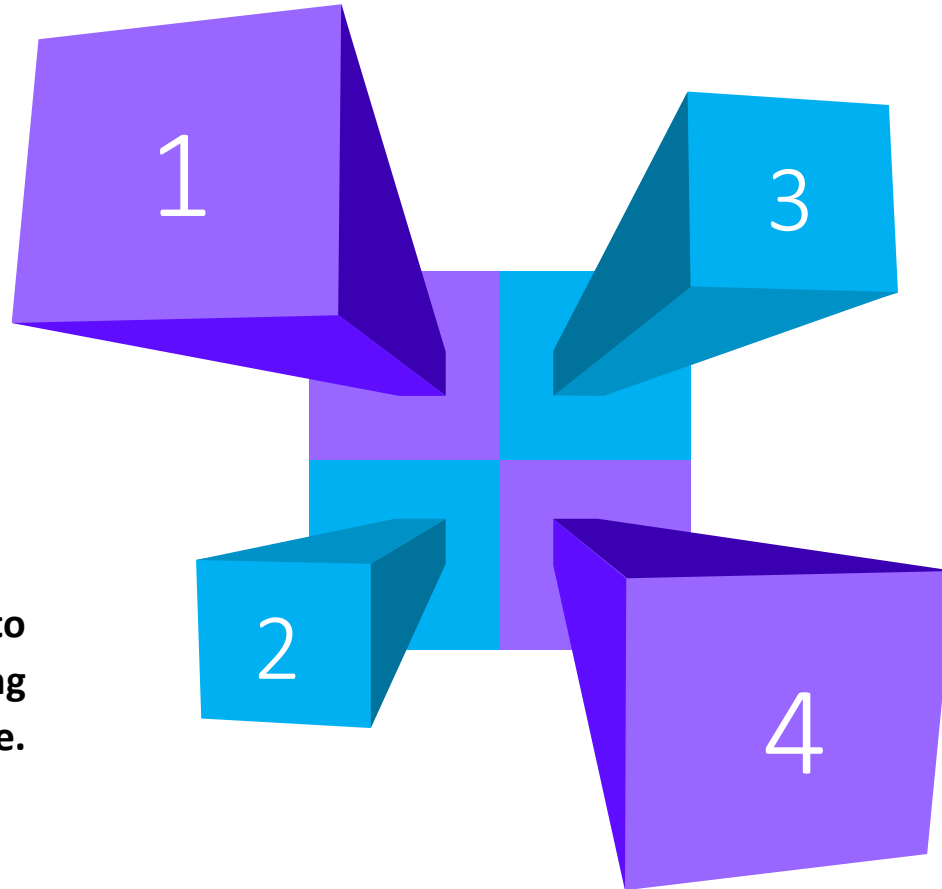
Here you don't need to manage the underlying infrastructure.

Public Access

Azure Container Instances can also get their own Public IP and DNS name.

File shares

You can also persist data via the use of Azure File shares.





Azure *Container Groups*

Azure Container Groups

Collection

This is a collection of containers that get scheduled on the same host machine.

Resources

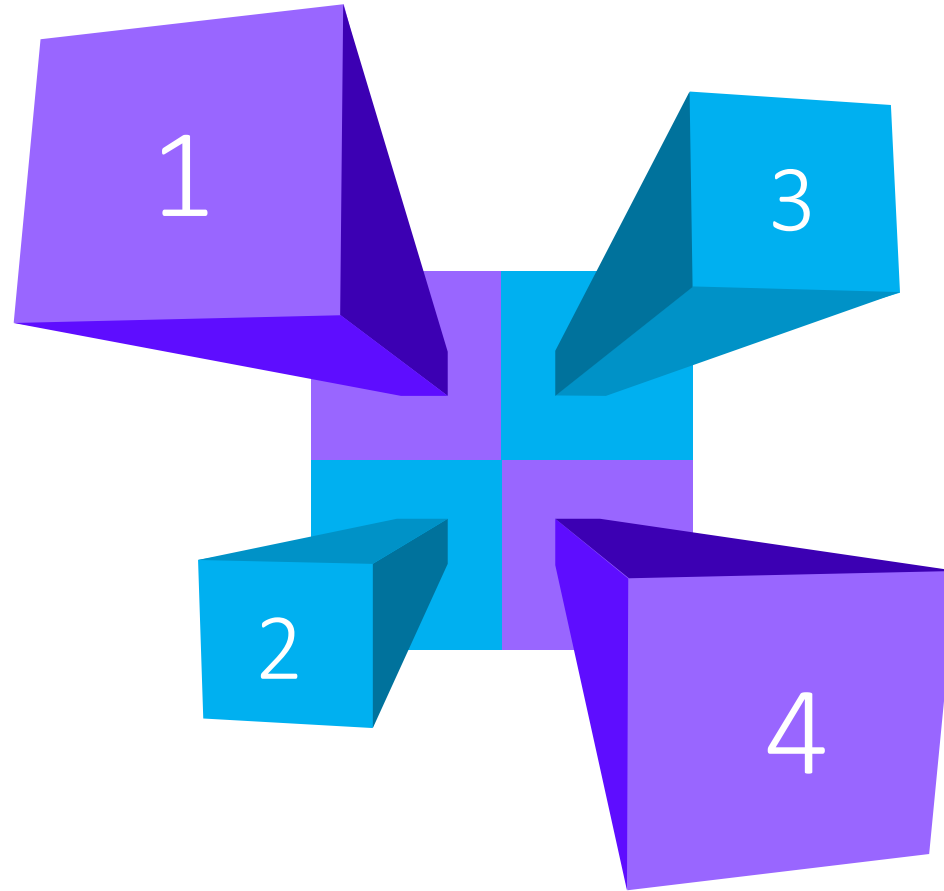
The containers then share the lifecycle, resources, local network and storage volumes.

Deployment

The deployment of a container group is done via a Resource Manager template or a YAML file.

File shares

You can also persist data via the use of Azure File shares.



Azure *Blob change feed*

Azure Blob change feed

Purpose

It provides an ordered ,
guaranteed, durable,
immutable, read-only log of
changes.

Operations

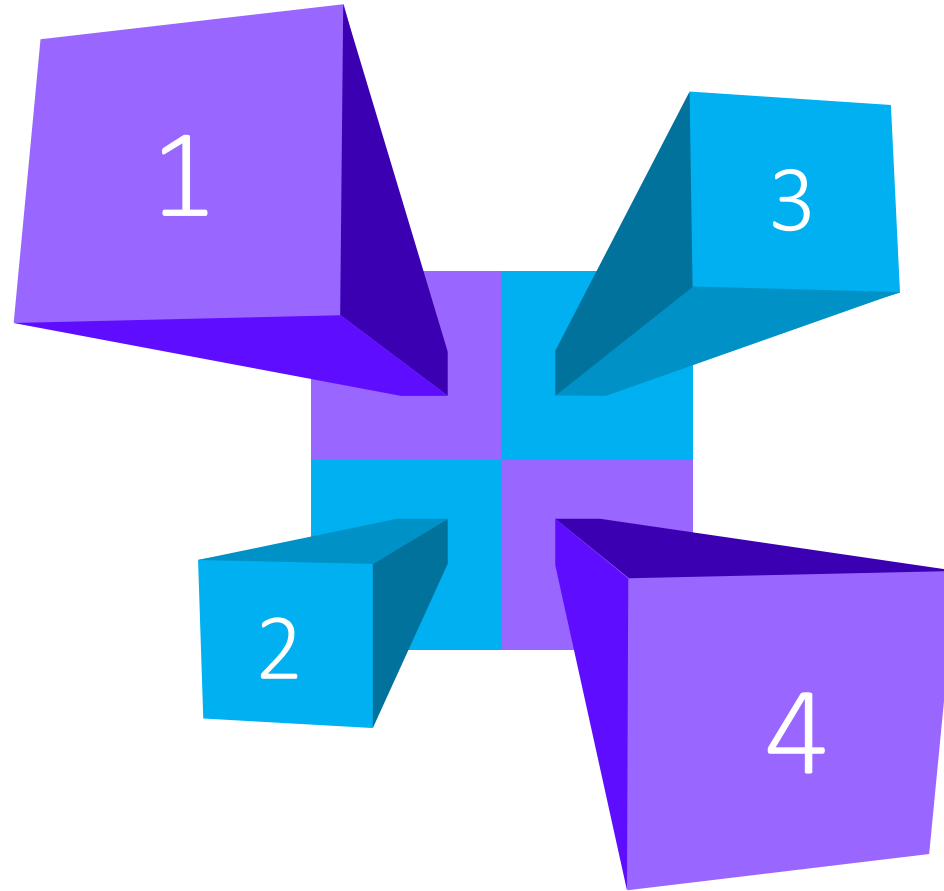
Audit log of events to Blob
data – Create, Update,
Delete.

Records

The change feed data is
stored in a contained
named \$blobchangeFeed.

Record format

The records are stored in
Apache Avro format.



Azure *Blob versioning*

Azure Blob versioning

Versions

Allows to maintain previous versions of a blob.

Version ID

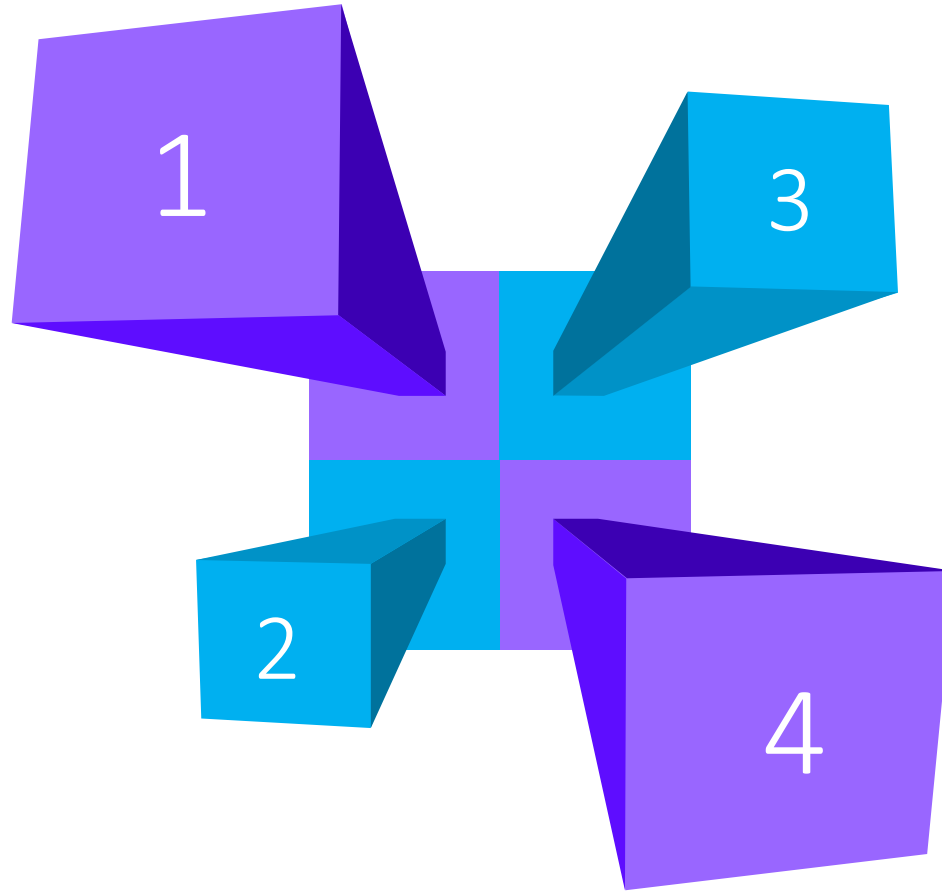
Each blob gets an initial version ID. A new version is set whenever the blob is updated.

Restore

You can then restore an earlier version of the blob.

Feature Enable

You can enable or disable the feature at any point in time.



Azure *Cosmos DB Costing*

Request Units



The cost of database operations is measured in terms of Request Units.



Request Units is a blended measure of CPU, IOPS and memory.



Request Units are used no matter which API you are using for the Azure Cosmos DB account.

Request Units



The cost of reading a single item (1 KB) is 1 Request Unit.



Other operations have their own measure of charged request units.



For each operation, you can see the amount of request units that were used for that operation.

Provisioned Throughput

1

Here you provision the number of Request Units. You can always increment or decrement the number of request units provisioned.

2

You are billed on an hourly basis.

3

You can provision throughput at the container or database level.

Serverless Mode

1

You don't provision any throughput.

2

This is managed by Azure Cosmos DB.

3

You will be billed based on the Request Units you consume.

Autoscale Mode

1

Here the Request Units can automatically scale based on demand.

2

The demand is checked both at the container and database level.

3

Great for mission critical workloads

Azure *Cosmos DB Partitions*

Azure Cosmos DB Partitions

Logical Partitions

Items in a container are divided into subsets called logical partitions.

Partition Key

The partition for the item is decided by the partition key that is associated with the item in the container.

Item id

Each item also has an item id. This helps to uniquely identify an item in the partition

Identification

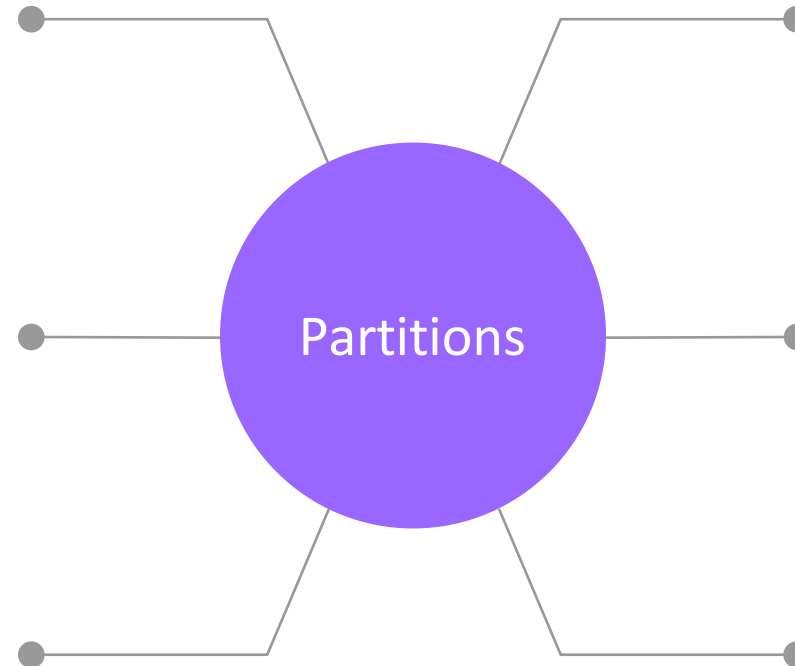
The combination of the partition key and the item id helps to uniquely identify the item within the container.

Size

Each logical partition can grow up to 20GB.

Limit

There is no limit when it comes to the number of logical partitions within a container.



Partition Key



Choose a property to be a partition key in which the value does not change.



The property should have a wide range of possible values.



Once you decide on the partition key for a container, you can't change it.

Change *Feed*

Azure Cosmos DB Change Feed

Record Changes

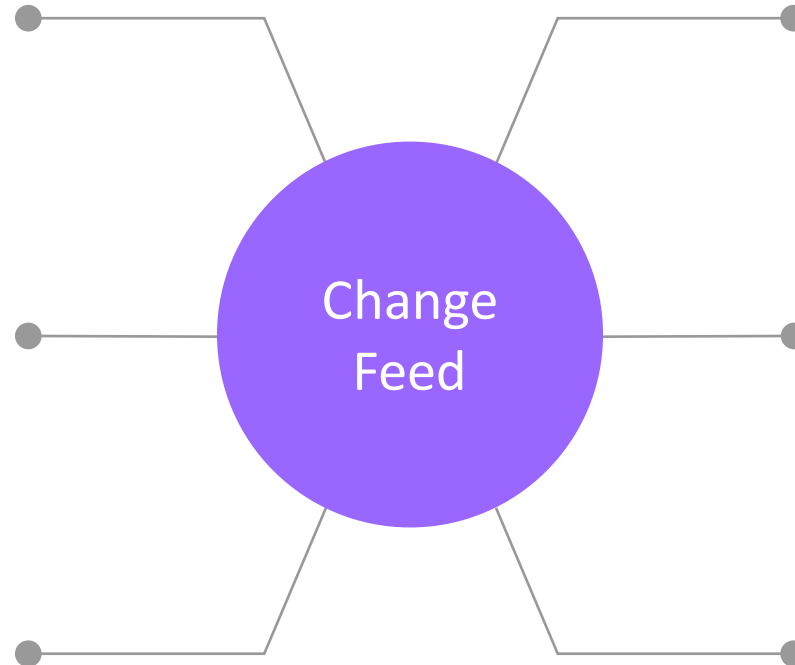
Here changes to the container are recorded in the order that they occur.

Feed

The feed consists of inserts and updates. Deletes are not recorded.

Process Change Feed

You can process the change feed with the use of Azure Functions or a change feed processor.



Records

The records for the change feed are written to another container.

Sorting

The change feed is sorted in the order of modification within each logical partition key value.

Throughput

The same provisioned throughput can be used to read from the container containing the changes.

Time *to Live*

Azure Cosmos DB - Time to Live

Delete items

Here items from the container can be deleted automatically after a certain period of time.

Level

This can be set at the item level or the container level.

Deletion Process

Items will be deleted based on the Time to live that is set in seconds.

Time to Live

Background task

The deletion process is done as a background task that uses the left-over Request Units.

For the item

For the TTL to work at the item level, the setting should also be defined at the container level.

Benefit

Here the process of deletion is managed by Azure Cosmos DB.



Azure *Active Directory*

Azure Active Directory

What is it

This is a cloud-based identity and access management service.

Usage

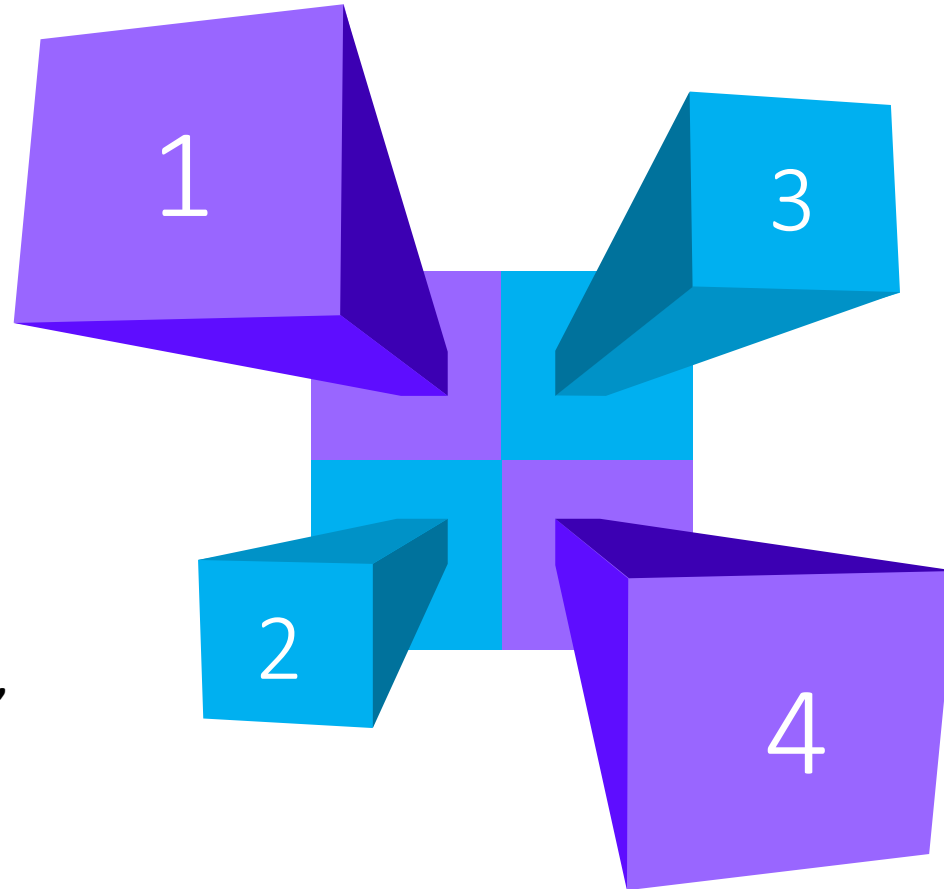
This can be used as an identity provider for Azure, Microsoft 365 and other SaaS products as well.

Identities

You can manage identities such as users, groups and even applications.

Features

Manage security aspects when it comes to your identities.



Azure Active Directory

1

Free

User and group management.

2

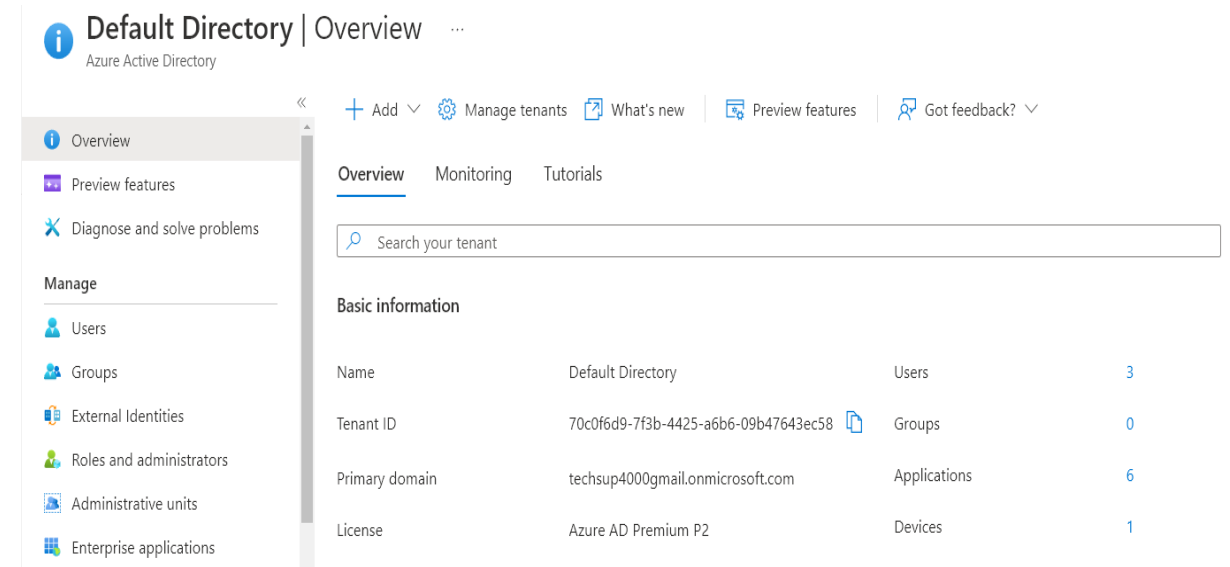
Premium P1

Dynamic groups, hybrid identities, self-service password reset for on-premise users.

3

Premium P2

Azure AD Identity protection and Privileged Identity Management.



The screenshot shows the 'Default Directory | Overview' page in the Azure Active Directory portal. The left sidebar contains a navigation menu with 'Overview' selected, followed by 'Preview features', 'Diagnose and solve problems', and a 'Manage' section with links to 'Users', 'Groups', 'External Identities', 'Roles and administrators', 'Administrative units', and 'Enterprise applications'. The main content area has tabs for 'Overview', 'Monitoring', and 'Tutorials'. Below the tabs is a search bar labeled 'Search your tenant'. The 'Basic information' section displays a table with tenant details.

Basic information			
Name	Default Directory	Users	3
Tenant ID	70c0f6d9-7f3b-4425-a6b6-09b47643ec58	Groups	0
Primary domain	techsup4000gmail.onmicrosoft.com	Applications	6
License	Azure AD Premium P2	Devices	1

Dynamic *Thresholds*

Azure Monitor

Dynamic thresholds

Here Azure Monitor uses machine learning to check the historical behavior of metrics.

Based on the historical data, it can then identify patterns and anomalies that could indicate possible issues.



Azure Monitor

Sensitivity

High – Here the alert rule will be triggered even for the smallest deviation.

Medium – Here you have more balanced thresholds and fewer alerts will be generated.

Low – Here alerts will only triggered on large deviations.





Application *Insights*

Application Insights

Monitoring

This provides the feature of application performance management and monitoring of live web applications.

Aspects

Here you can see aspects such as detecting performance issues or any other issues.

Support

There is support for .NET, Node.js, Java and Python.

Applications

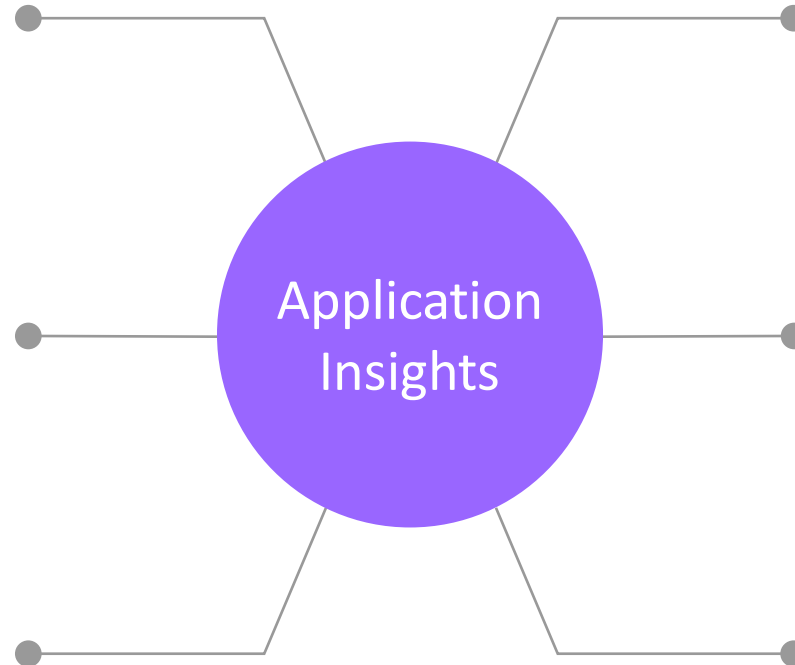
This works for applications hosted in Azure, on-premises environments, or other cloud platforms.

Integration

It has Integration with the Visual Studio IDE.

Users

You can also see how users interact with your application.



Application Insights

How does it work

You can install a small instrumentation package (SDK) for your application. Or use the Application Insights agent.

You can instrument web applications, background components and JavaScript in web pages.

The telemetry data sent by Application Insights has very little impact on the performance of your application.





Application *Insights*

Application Insights

Users, Sessions and Events

Users – Here you can see how many people have used your application and its features.

Session – You can see sessions of user activity. This includes certain pages and features of the application.

Events – This gives a view of how often certain pages and features have been used in the application.



Application Insights

Funnels – Here you can have multiple stages like a pipeline. And then you can see how users are progressing through your application as an entire process.

Cohorts – This is a set of users, sessions, events or operations that have something in common. It helps to analyze a particular set of users or events.

Impact – Here you can see how load times and other aspects of your application impact the conversion rate for your application.



Application Insights

Retention – Here you can analyze how many users return back to your application.

User flows – This can help in answering useful questions such as

- 1. What do users click on a page within the application*
- 2. Where are the places within the application that users churn the most from the site.*
- 3. Are there places in the application where the users repeat the same action over and over again.*





Application *Insights*

Availability Tests

Tests

You can define tests that can monitor the availability and responsiveness of an application.

Web Requests

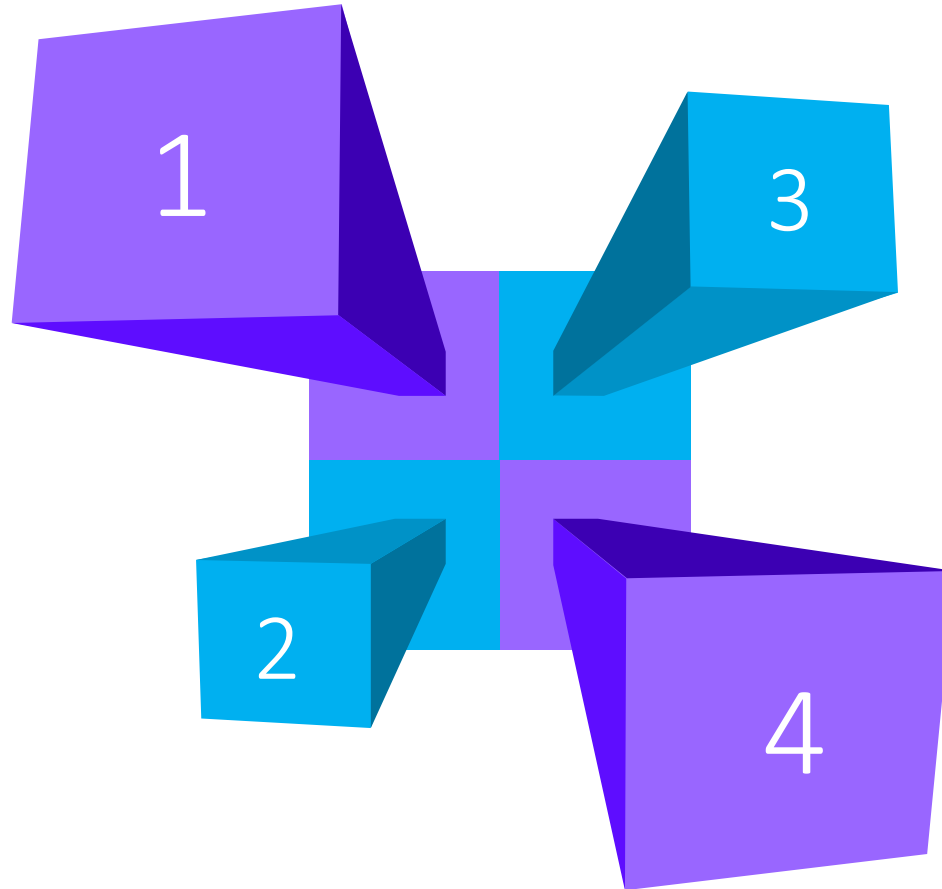
This feature can send web requests to your web application from different points across the world.

Endpoint

You can setup tests against an HTTP or HTTPS endpoint that's accessible over the public Internet.

API

You can also test your REST API's as well.



Azure *Cache for Redis*

Azure Cache for Redis

Data store

This is an in-memory data store that is based on the Redis software.

Server memory

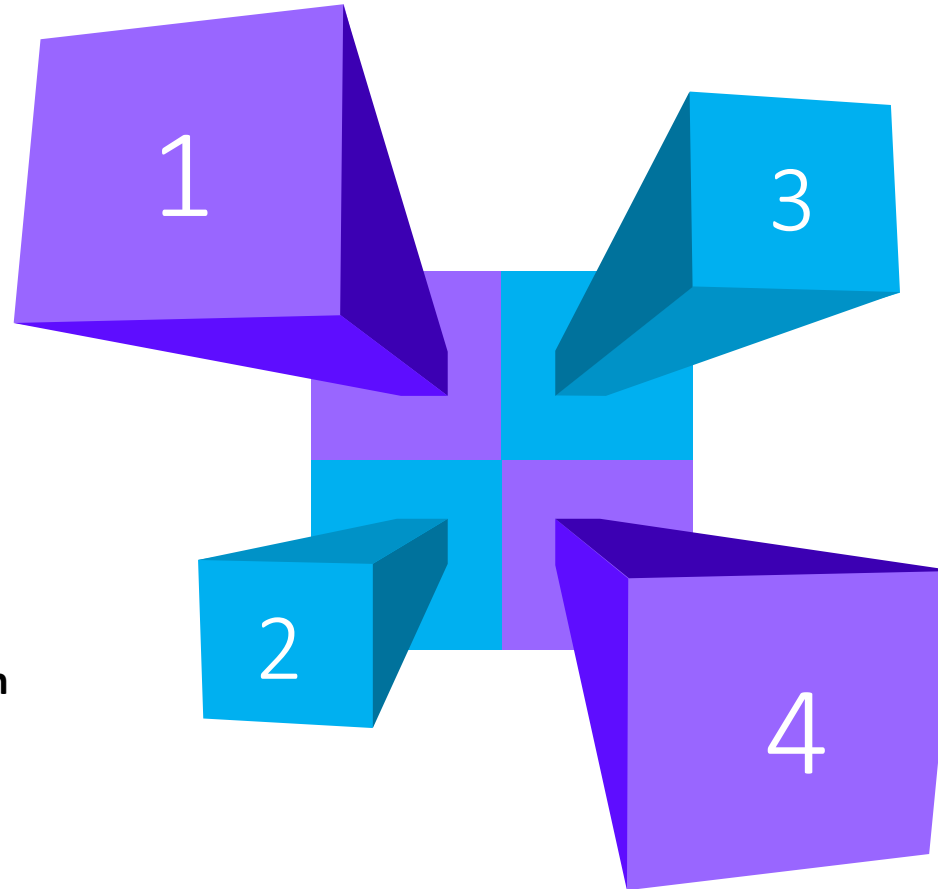
Here you can store frequently accessed data in server memory.

Advantage

The advantage of server memory is that you can write and read data quickly.

For the application

It helps to provide low latency to your data and high throughput.



Azure *Front Door*

Caching

You can enable caching for your endpoint routes.

When delivering large files, Azure Front Door request the file from the origin in chunks of 8 MB.

When a chunk of data is received, the data chunk is cached and immediately sent to the user.



Caching

You have different query string behaviors when it comes to caching

Ignore query strings – Here the query strings from the requestor is passed to the origin for the first request and then the asset is cached. All further requests ignore the query strings until the expiry of the cached asset.

Cache every unique URL – Here each URL is treated as unique and cached separately.

Specify cache key query string – You can decide to include/exclude specified parameters to determine what gets cached.



Topic *filters*

Azure Service Bus Topic filters

With the help of filters, subscribers can decide which messages they want to receive.

This can be defined with the use of rules.

Each rule has a filter condition.

The message will contain a property called RuleName that shows the matching rule.



Azure Service Bus Topic filters

Boolean filter – This is either the TrueFilter or FalseFilter. This can cause all messages to be sent to the subscription or for the subscription to receive no messages.

SQL filters – Here SQL like language can be used to evaluate the messages user-defined or system properties.

Correlation filters – Here also conditions can be matched against the messages' user or system defined properties.

Correlation filters are more efficient than SQL filters.

