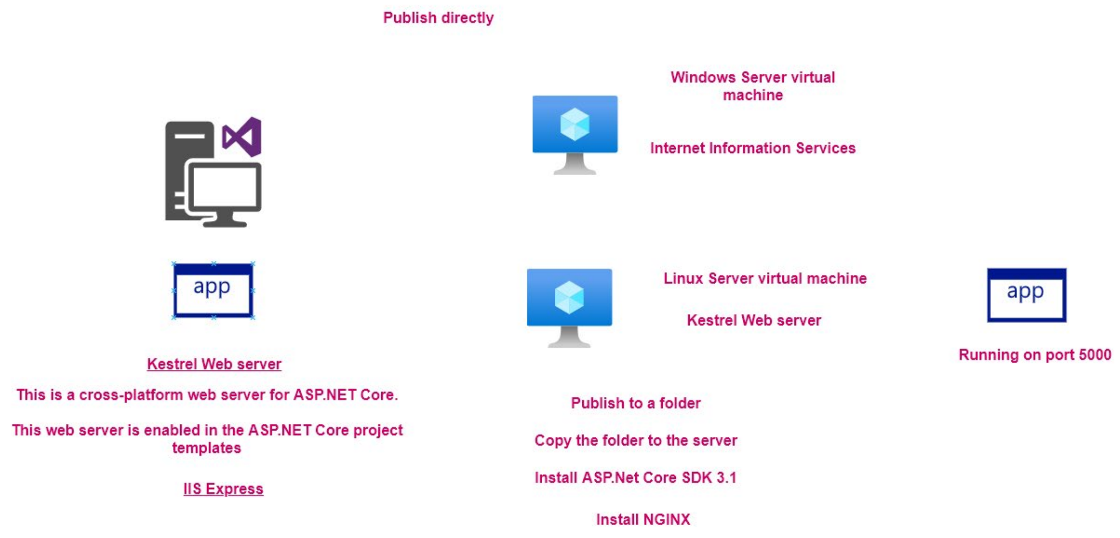
Develop Azure compute solutions – Virtual Machines

Hosting options for a web application



Publish directly

Windows Server virtual machine

Internet Information Services

Linux Server virtual machine

Kestrel Web server

Running on port 5000

Kestrel Web server

This is a cross-platform web server for ASP.NET Core.

This web server is enabled in the ASP.NET Core project templates

IIS Express

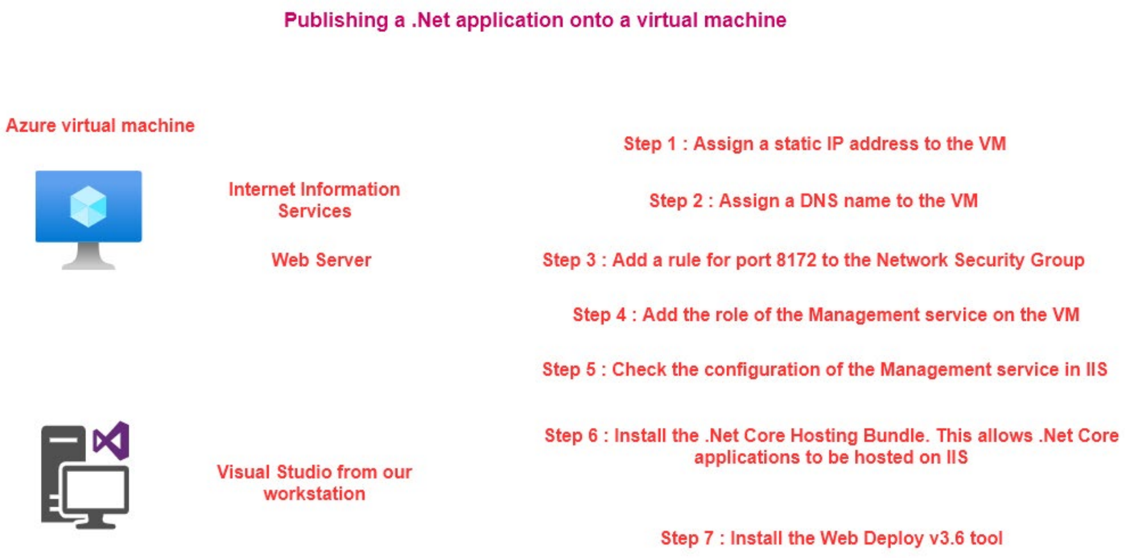
Publish to a folder

Copy the folder to the server

Install ASP.Net Core SDK 3.1

Install NGINX

Publishing a .Net project onto a Windows server



Publishing a .Net application onto a virtual machine

Azure virtual machine

Internet Information Services

Web Server

Visual Studio from our workstation

Step 1 : Assign a static IP address to the VM

Step 2 : Assign a DNS name to the VM

Step 3 : Add a rule for port 8172 to the Network Security Group

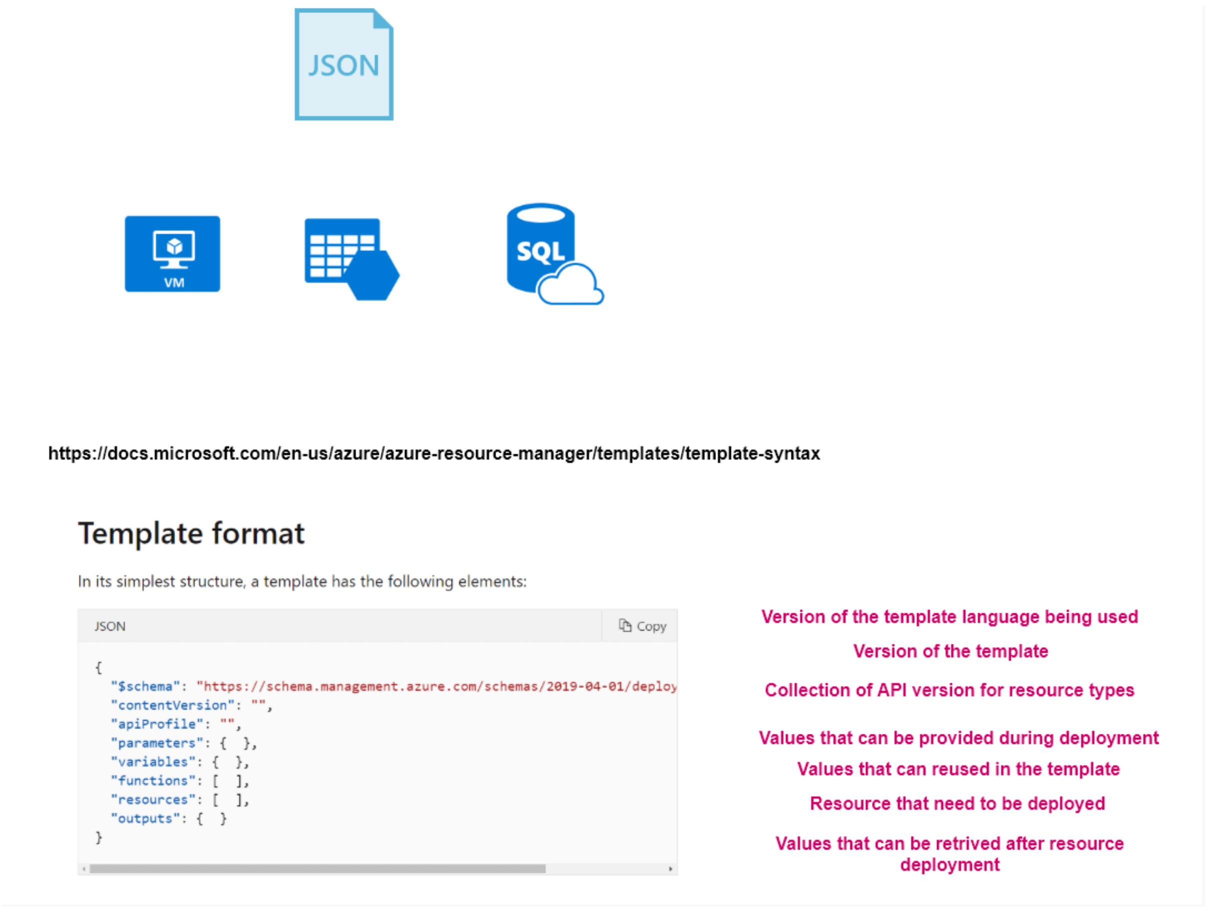
Step 4 : Add the role of the Management service on the VM

Step 5 : Check the configuration of the Management service in IIS

Step 6 : Install the .Net Core Hosting Bundle. This allows .Net Core applications to be hosted on IIS

Step 7 : Install the Web Deploy v3.6 tool

ARM templates



https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/template-syntax

Template format

In its simplest structure, a template has the following elements:

JSON

```json
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploy
  "contentVersion": "",
  "apiProfile": "",
  "parameters": {  },
  "variables": {  },
  "functions": [  ],
  "resources": [  ],
  "outputs": {  }
}
```

Version of the template language being used

Version of the template

Collection of API version for resource types

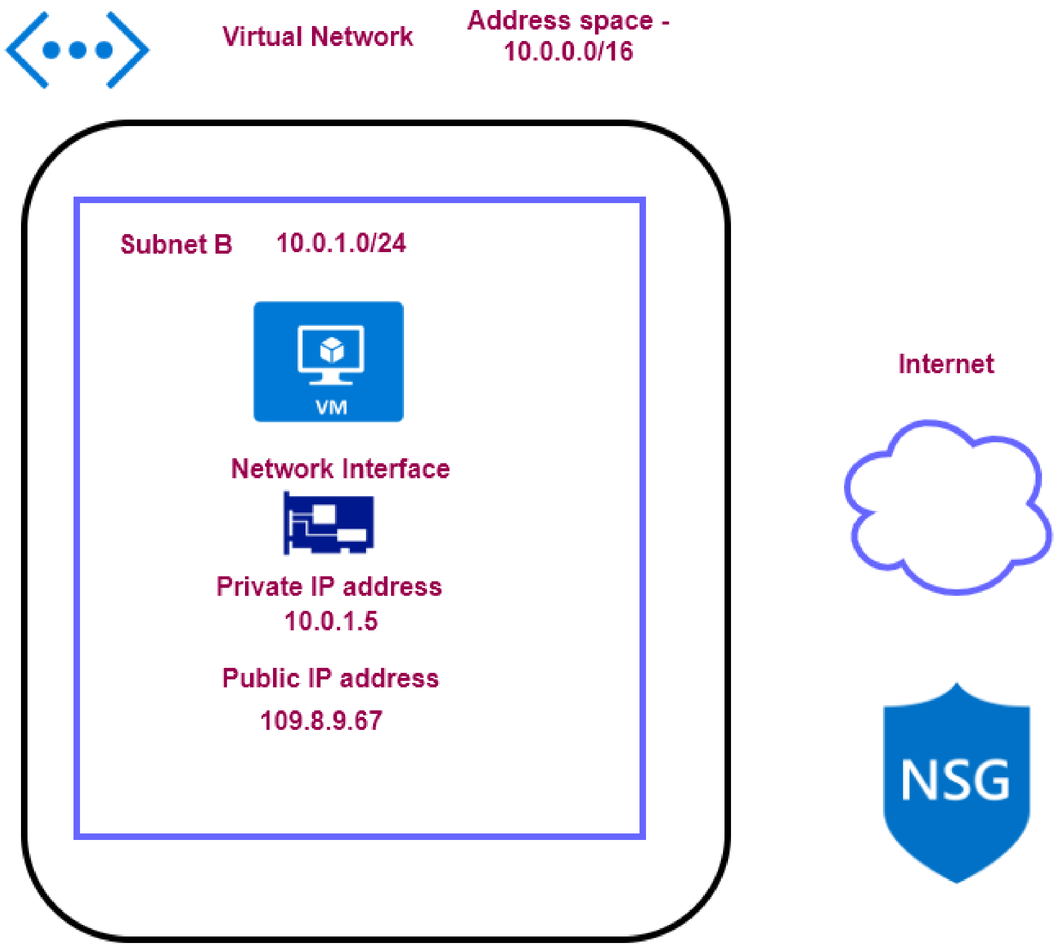
Values that can be provided during deployment

Values that can reused in the template

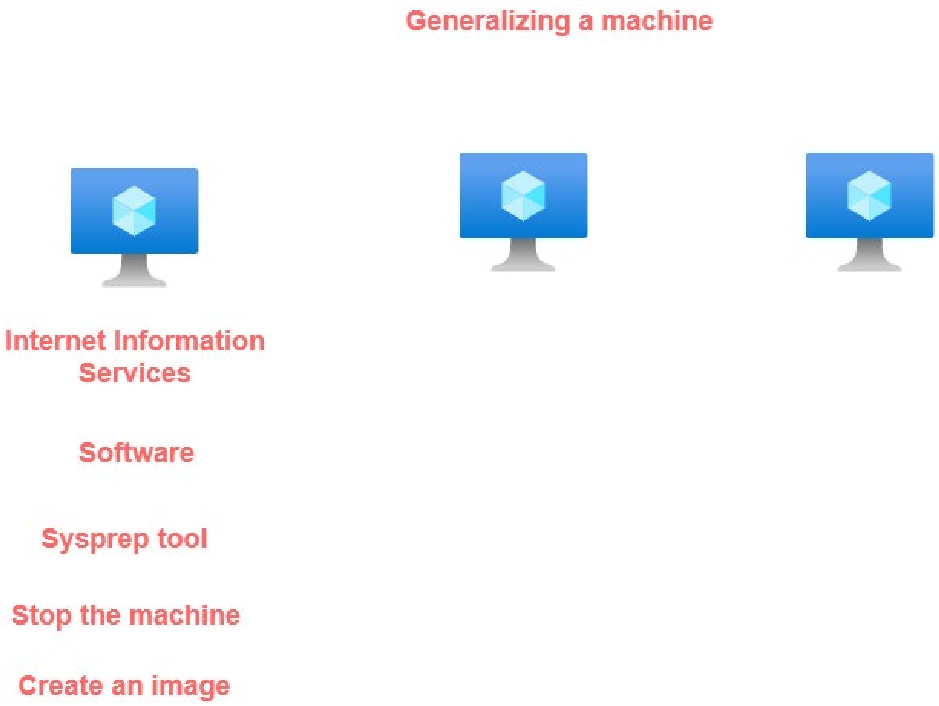
Resource that need to be deployed

Values that can be retrived after resource deployment

Understanding the different components of a VM deployment when it comes to ARM templates

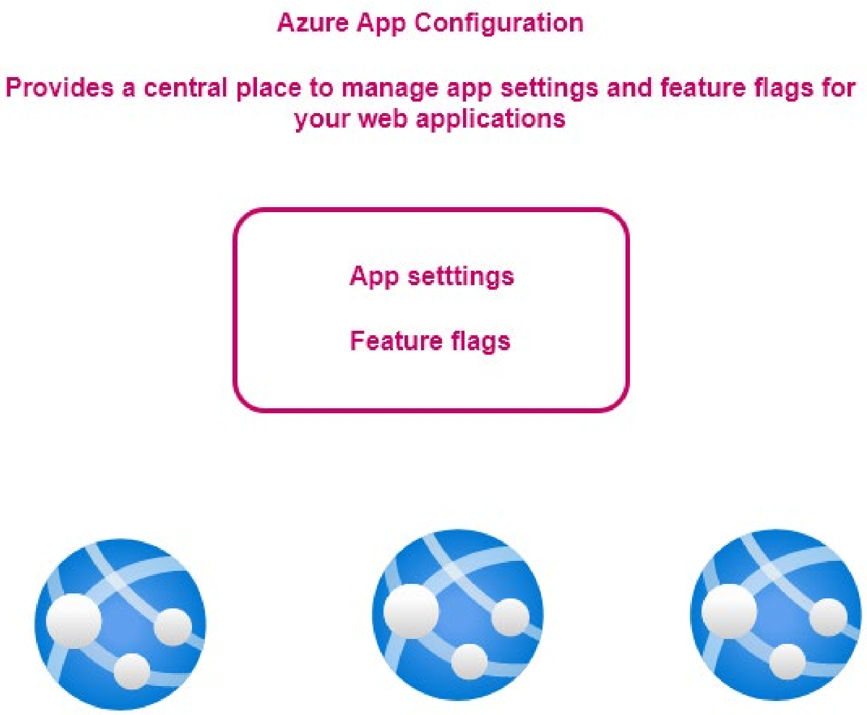

Virtual Network
Address space - 10.0.0.0/16
Subnet B 10.0.1.0/24
VM
Network Interface
Private IP address 10.0.1.5
Public IP address 109.8.9.67
Internet
NSG

Creating an image of a VM



Develop Azure compute solutions – Azure Web Apps and Azure Functions

Azure App Configuration



Azure App Configuration

Provides a central place to manage app settings and feature flags for your web applications

App setttings

Feature flags

Azure Web App Logging

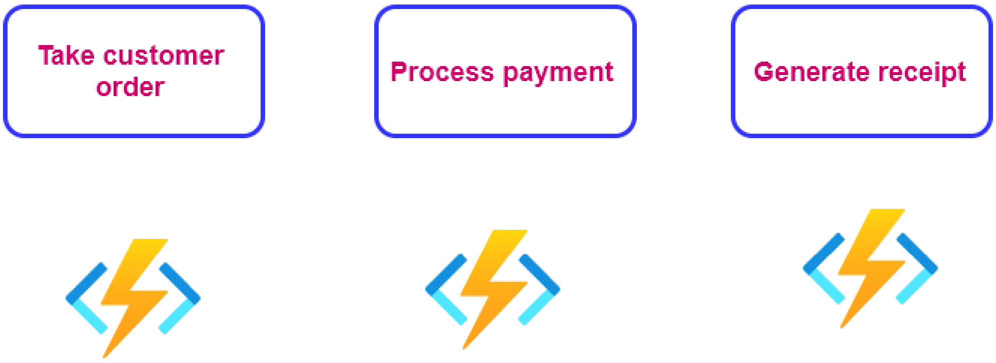
- You get a set of logging features that are available for the Azure Web App.
- The different types of logging that are available are
- **Application Logging** – This captures log messages that are generated by your application code.
- **Web server logging** – This records raw HTTP request data.
- **Detailed 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.

Durable Functions

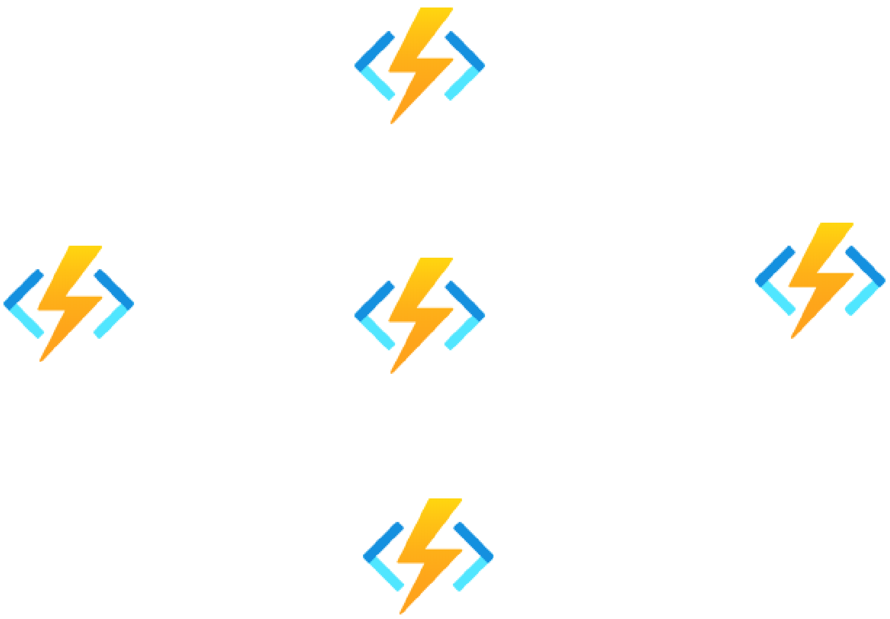
Durable Functions

Stateless in nature

Workflow



Fan-out scenario

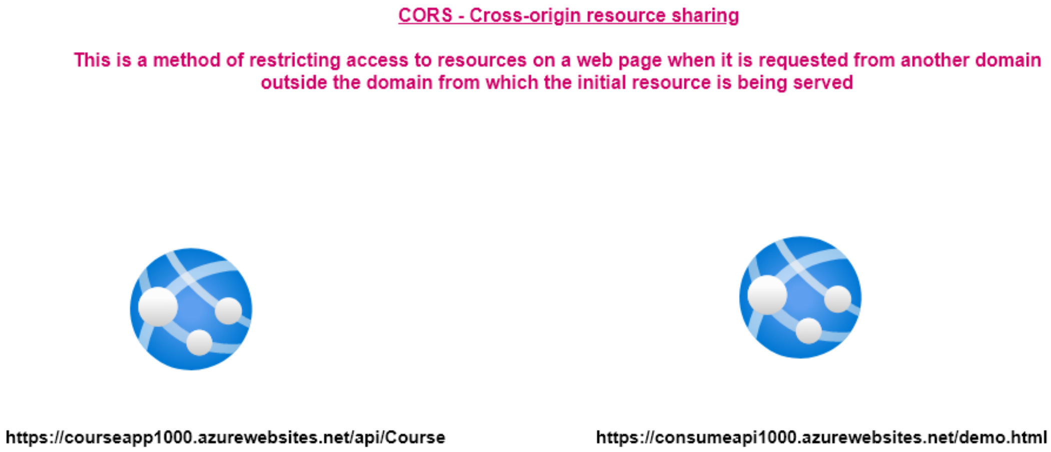


Starter Function

Orchestrator Function

Activity Function

Azure Web App – CORS



CORS - Cross-origin resource sharing

This is a method of restricting access to resources on a web page when it is requested from another domain outside the domain from which the initial resource is being served

https://courseapp1000.azurewebsites.net/api/Course

https://consumeapi1000.azurewebsites.net/demo.html

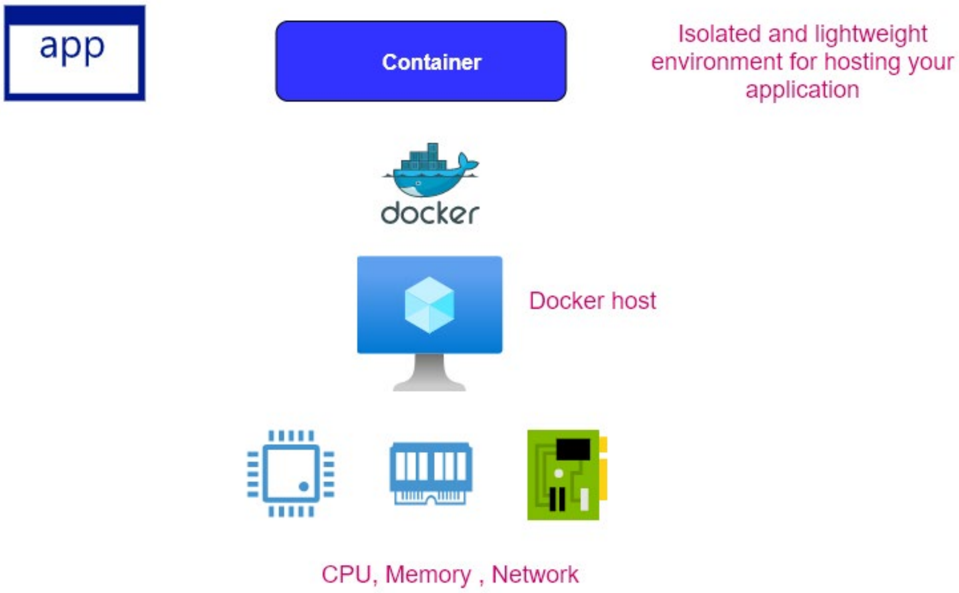
Develop Azure compute solutions – Docker, Azure Container Instances, Kubernetes

Docker Architecture

What is Docker

This is an open platform that is used for developing, shipping and running applications.

Docker has the ability to package and run an application in a loosely isolated environment called a container



Isolated and lightweight environment for hosting your application

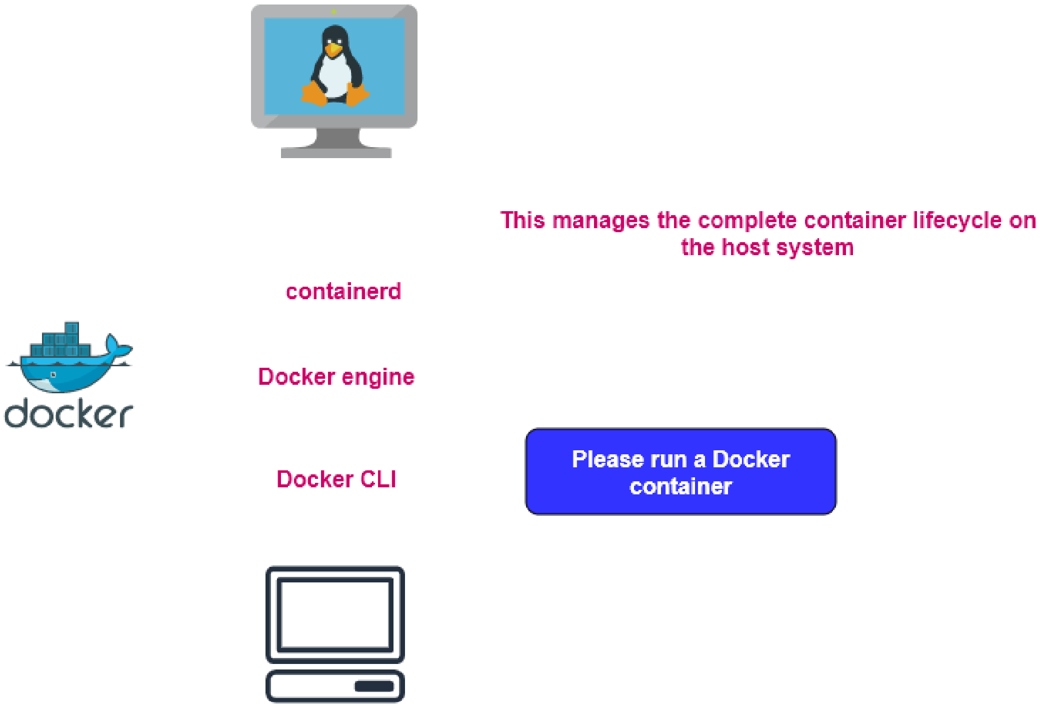
Image

This is a read-only template with instructions that are required to create the Docker container

Container

This is a runnable instance of an image

Installing Docker on a Linux VM



This manages the complete container lifecycle on the host system

containerd

Docker engine

Docker CLI

Please run a Docker container

<u>Run your Linux systems on your windows machines</u>

Windows subsystem for Linux

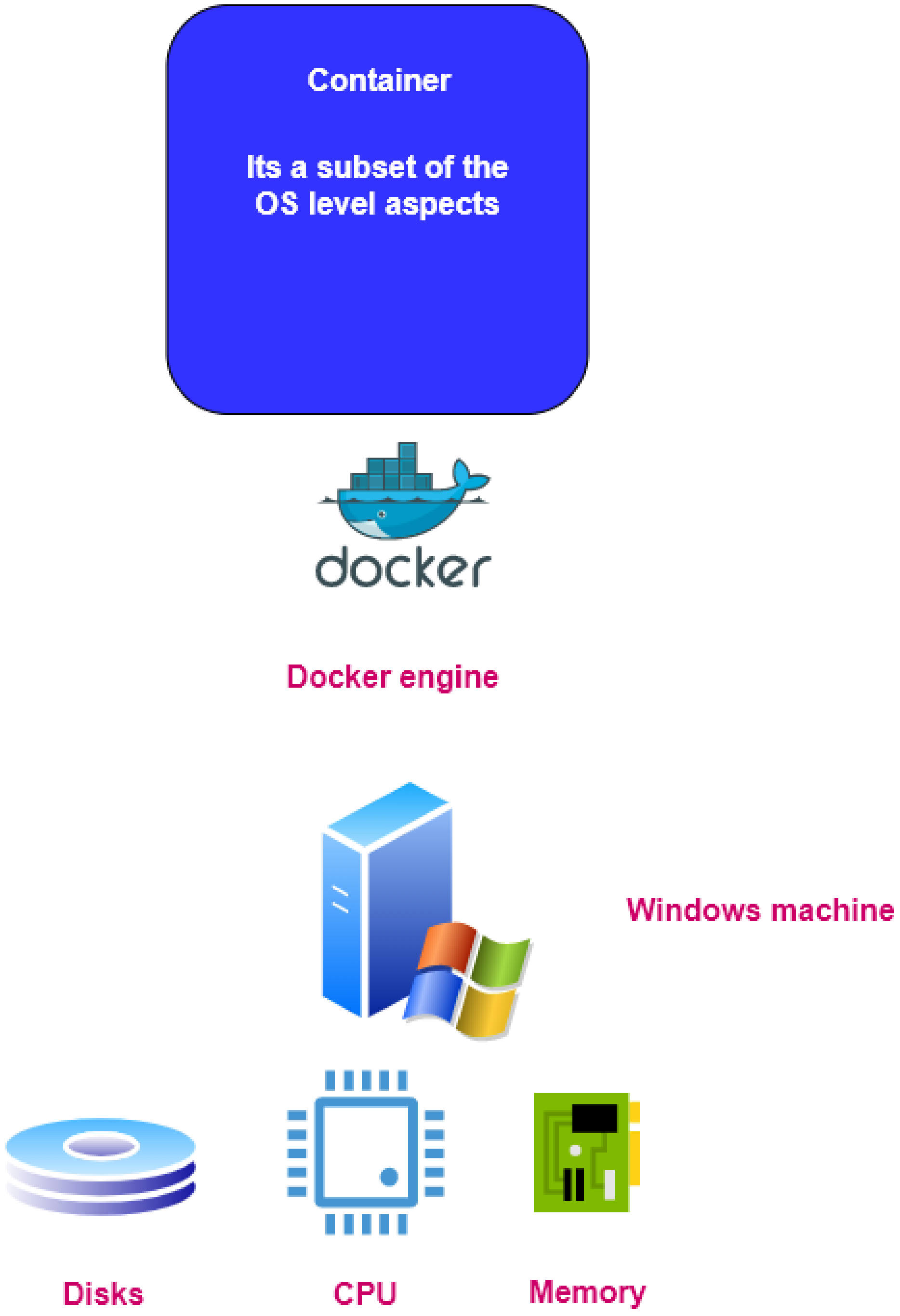
Windows system
Linux system
app
Windows system

Here you get access to a Linux environment running on Windows

Developers can test out their applications locally itself on the Linux subsystem.

When considering using Windows containers

Why won't the nginx container run as a Windows container



Container

Its a subset of the OS level aspects

Docker engine

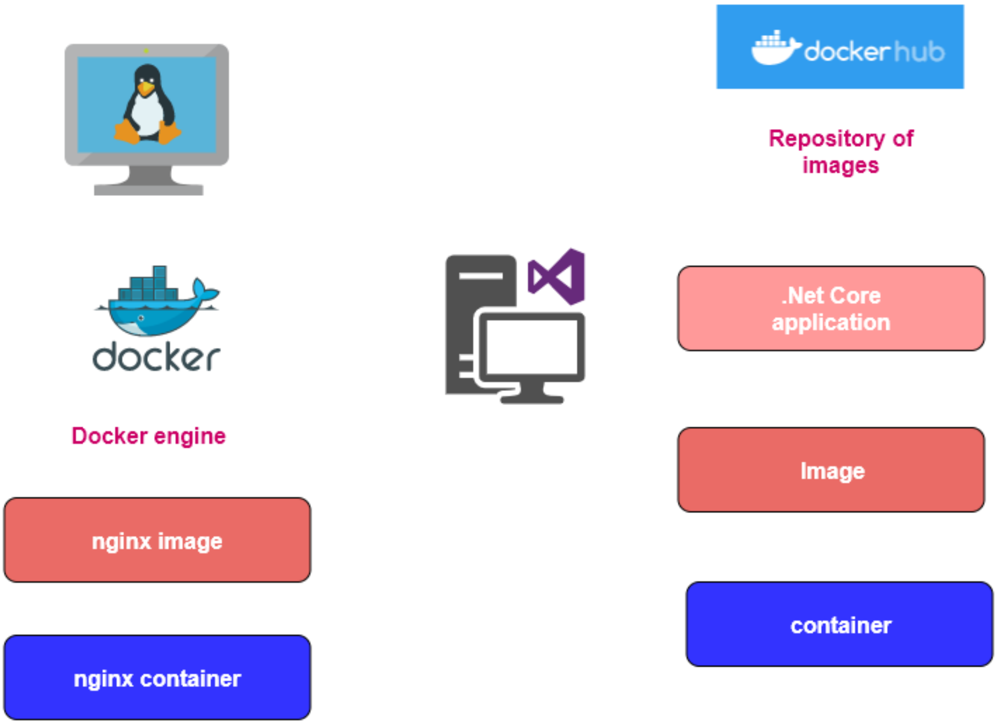
Windows machine

Disks

CPU

Memory

The need for Azure container registry



Repository of images

Docker engine

.Net Core application

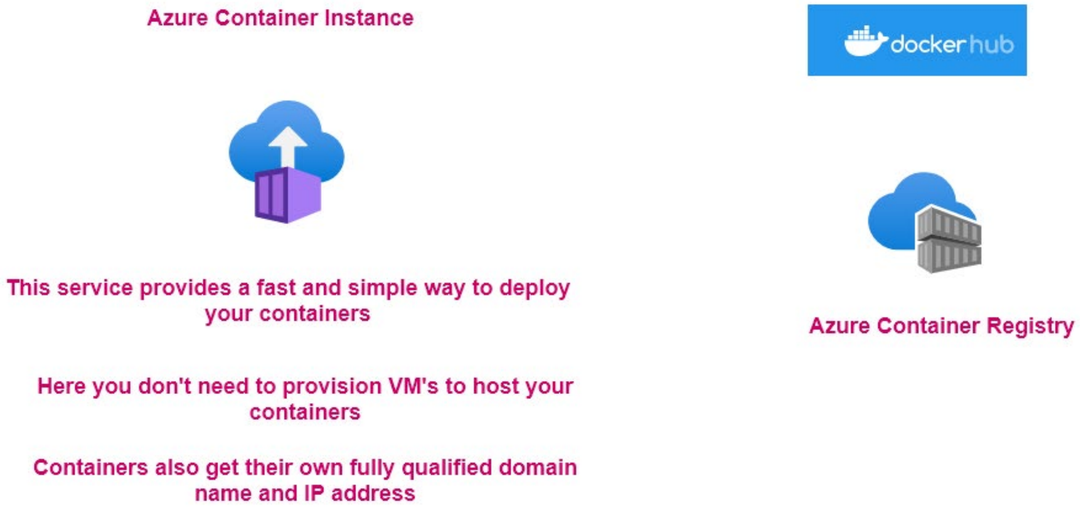
nginx image

Image

nginx container

container

Azure Container Instances



Azure Container Instance

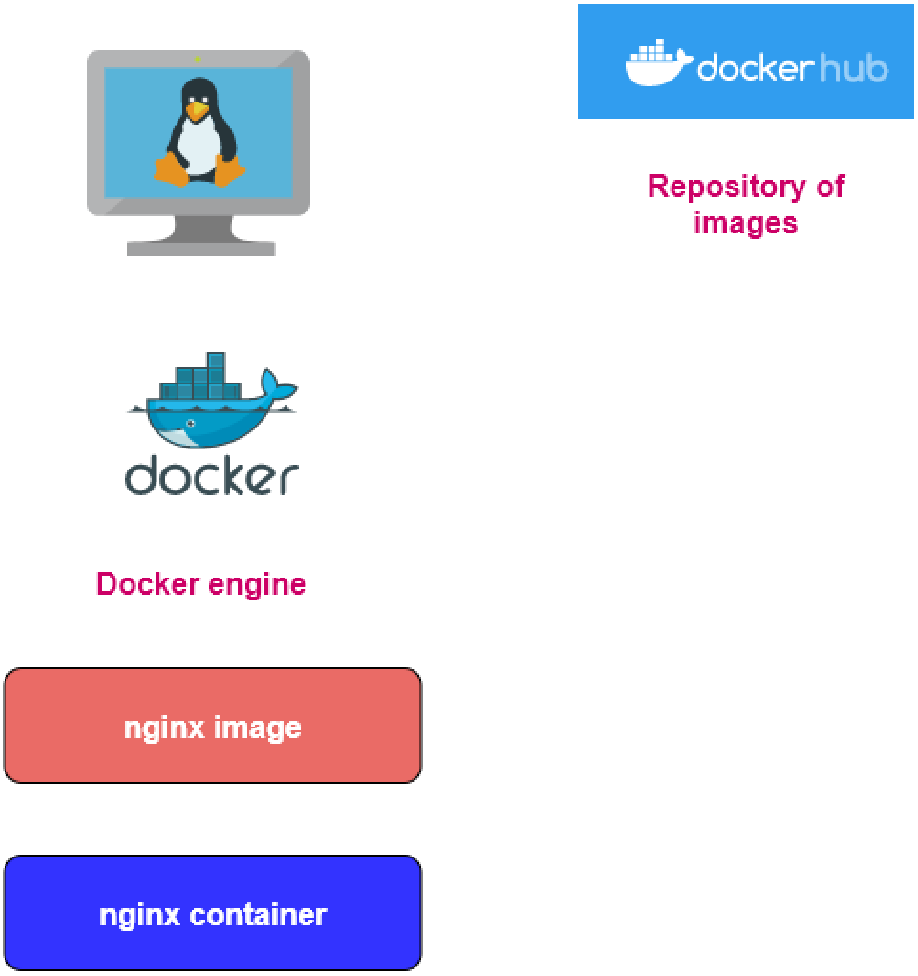
This service provides a fast and simple way to deploy your containers

Here you don't need to provision VM's to host your containers

Containers also get their own fully qualified domain name and IP address

Azure Container Registry

Understanding images

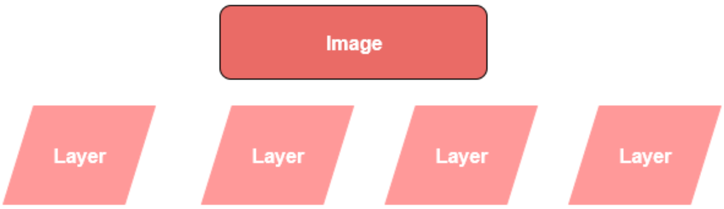


Repository of images

Docker engine

nginx image

nginx container



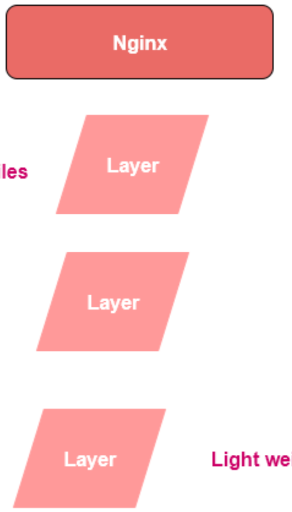
Image

Layer

Layer

Layer

Layer



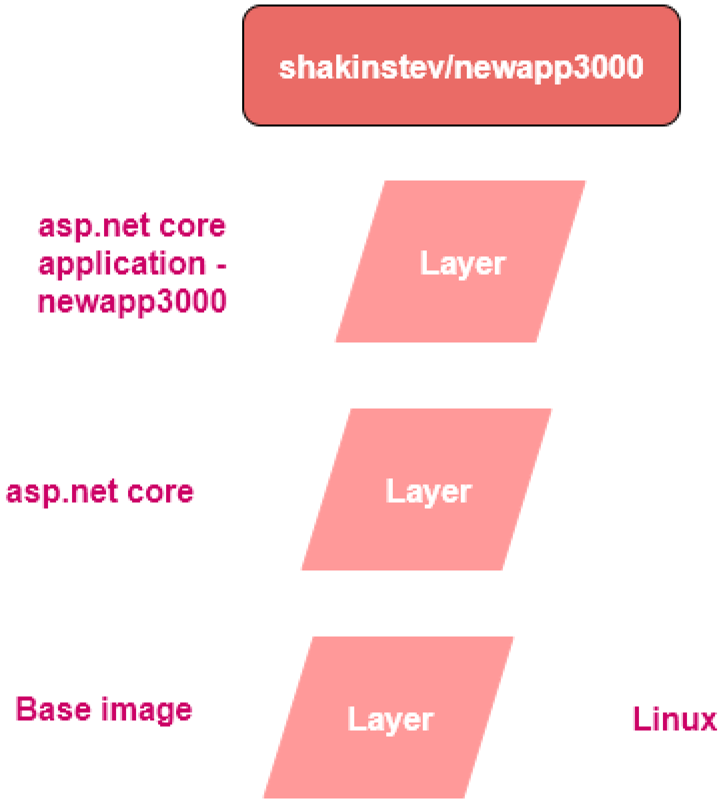
Nginx

Configuration files — Layer

nginx — Layer

Base image — Layer — Light weight OS layer - Ubuntu

Docker makes all layers seem as one seamless file system



shakinstev/newapp3000

asp.net core application - newapp3000

Layer

asp.net core

Layer

Base image

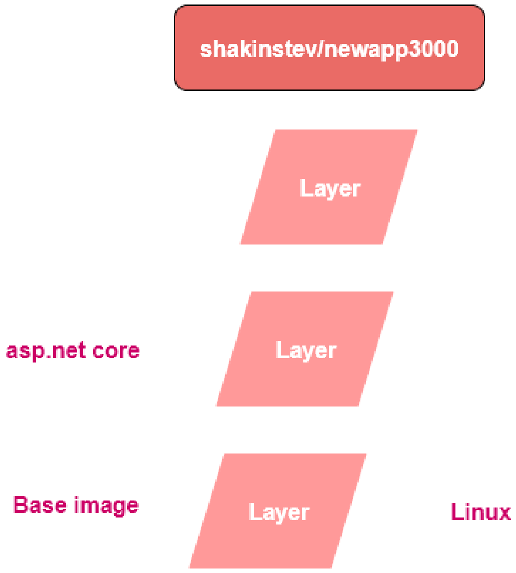
Layer

Linux

The Docker build command

docker build command

This command uses the instructions in the Dockerfile to build the image



shakinstev/newapp3000

Layer

asp.net core

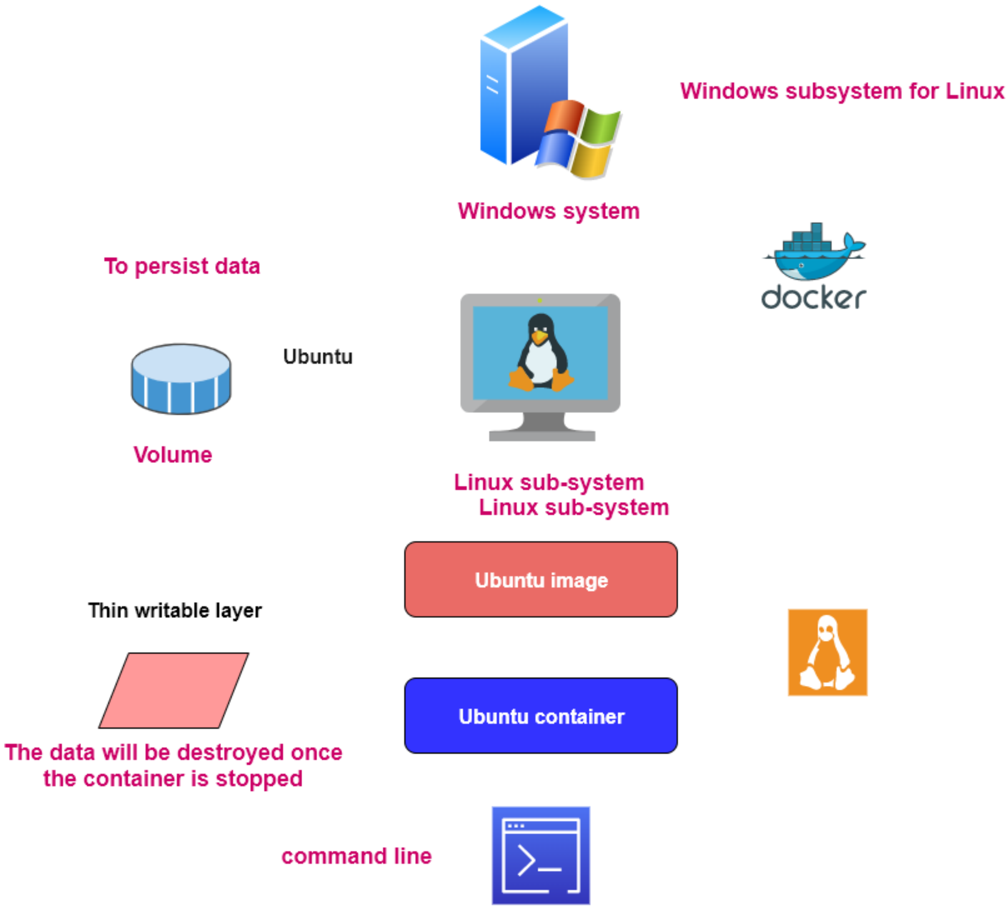
Layer

Base image

Layer

Linux

You can use volumes to persist data

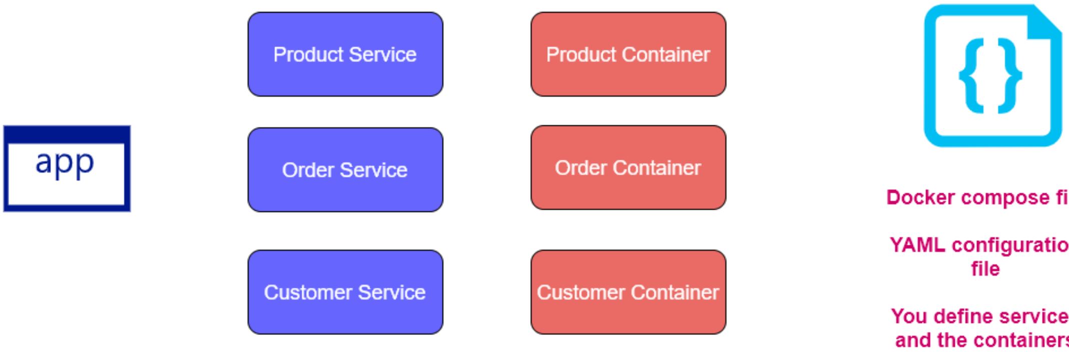


Docker compose

Docker compose

This is a tool that is used for defining and running multi-container Docker applications

You can also use a YAML file to configure the application service



app

Product Service

Order Service

Customer Service

Product Container

Order Container

Customer Container

Microservices architecture

Docker compose file

YAML configuration file

You define services and the containers that need to be deployed

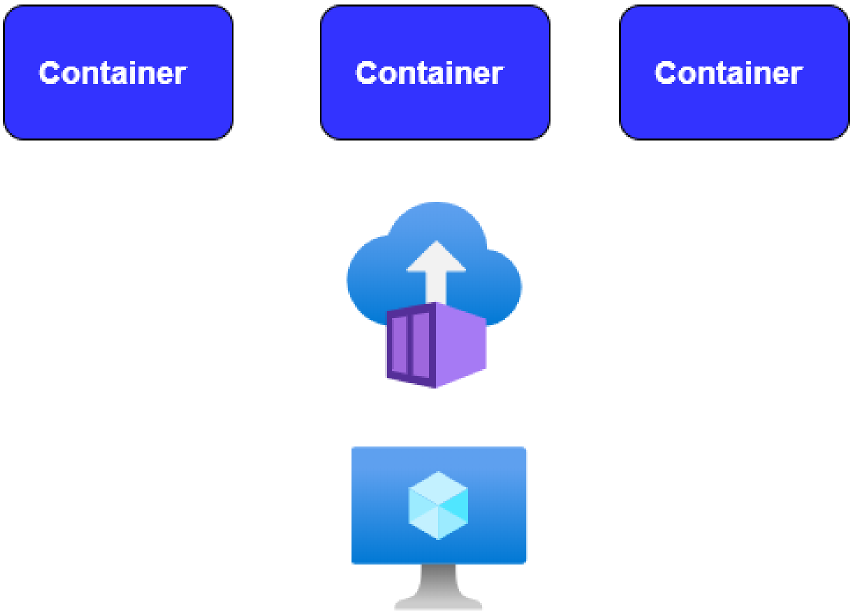
Azure Container Groups

Azure Container groups

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

The containers in the container group shared the lifecycle, resources, local network and storage volumes.

The dpeloyment of the container group can be done via a Resource Manager template of a YAML file.

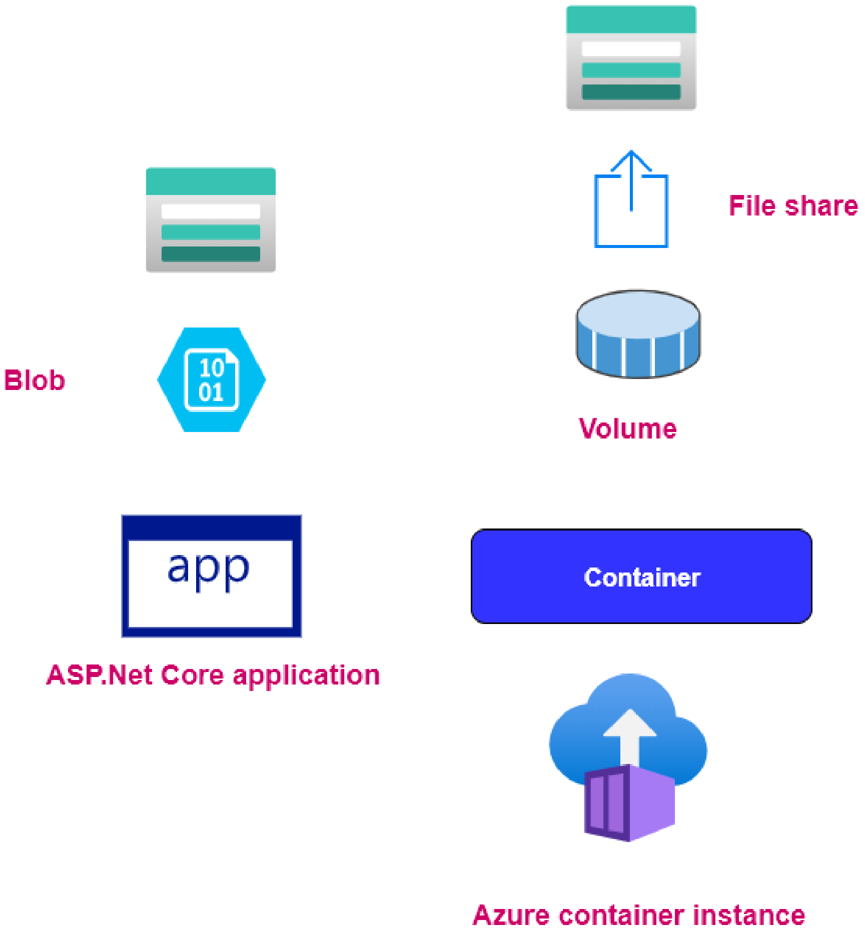


Container

Container

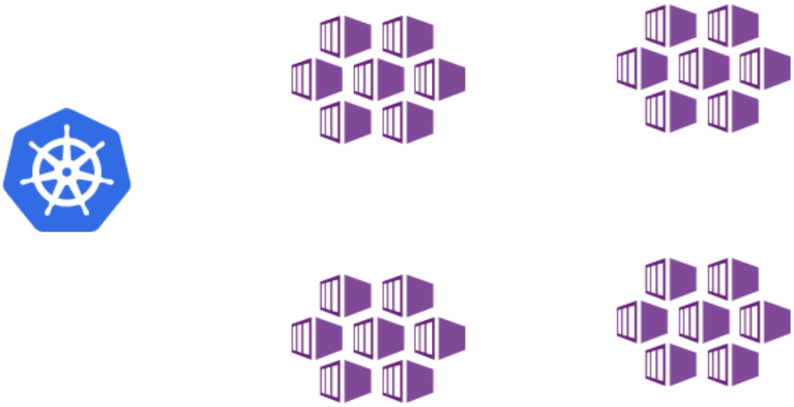
Container

Azure Container Instances – Connecting to file shares


File share
Blob
Volume
app
ASP.Net Core application
Container
Azure container instance

Kubernetes

Kubernetes



Managing containers at scale

Azure Kubernetes - Managed service for Kubernetes on Azure

Kubernetes is used to orchestrate your containers for hosting your applications

MySQL

Logging

Web Layer

Business layer

All calls to the cluster

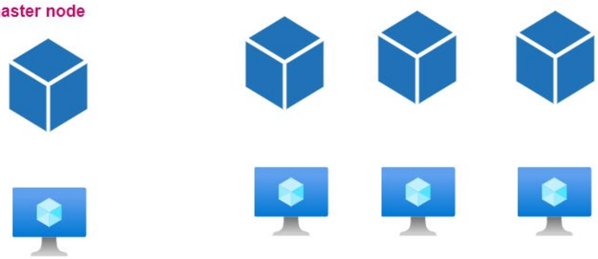
Nodes - These are used for hosting your containers

master node

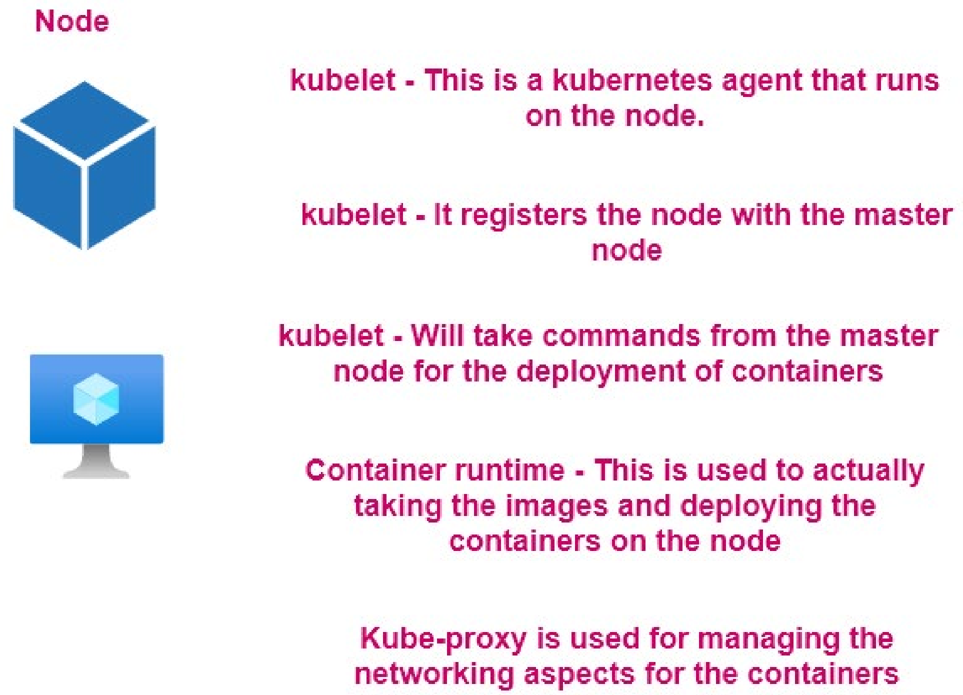
apiserver

etcd - Cluster store

State of the cluster



The master node is used to control the nodes in the cluster



Node

kubelet - This is a kubernetes agent that runs on the node.

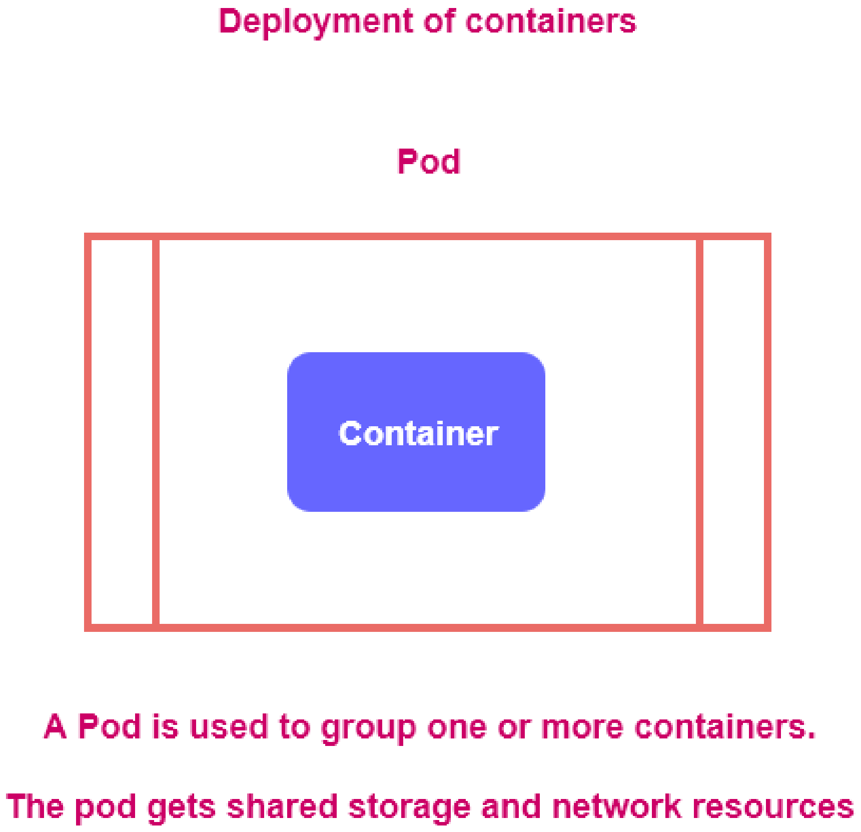
kubelet - It registers the node with the master node

kubelet - Will take commands from the master node for the deployment of containers

Container runtime - This is used to actually taking the images and deploying the containers on the node

Kube-proxy is used for managing the networking aspects for the containers

Azure Kubernetes Pods

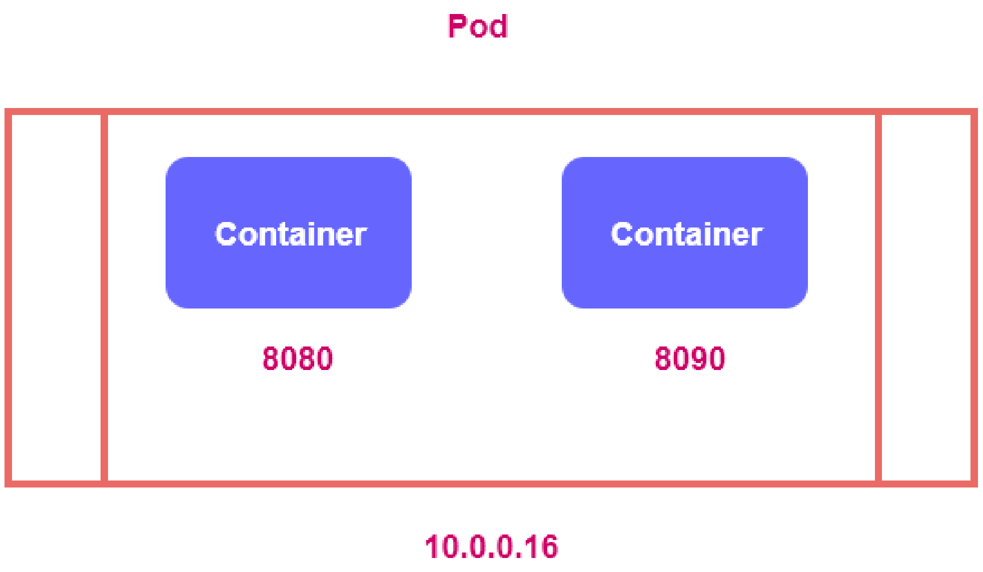


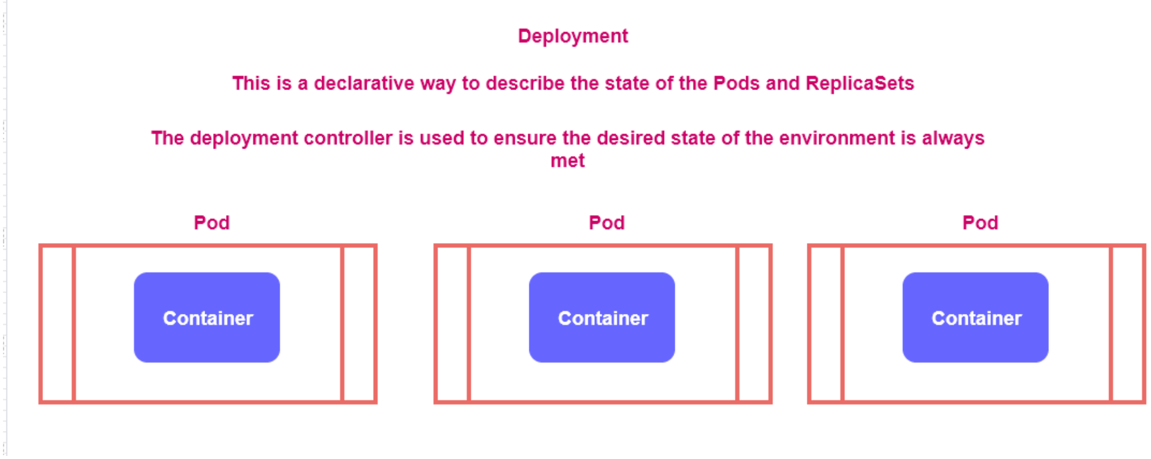
Deployment of containers

Pod

A Pod is used to group one or more containers.

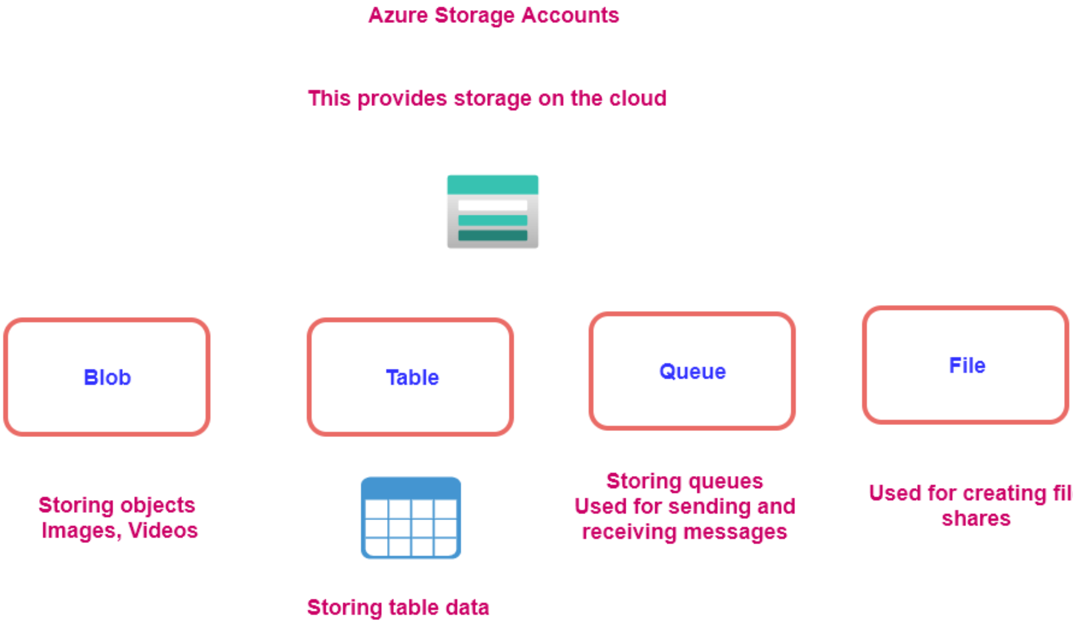
The pod gets shared storage and network resources




Deployment
This is a declarative way to describe the state of the Pods and ReplicaSets
The deployment controller is used to ensure the desired state of the environment is always met
Pod
Pod
Pod
Container
Container
Container

Develop for Azure Storage

Azure Storage Accounts



Azure Storage Accounts

This provides storage on the cloud

Blob

Table

Queue

File

Storing objects
Images, Videos

Storing table data

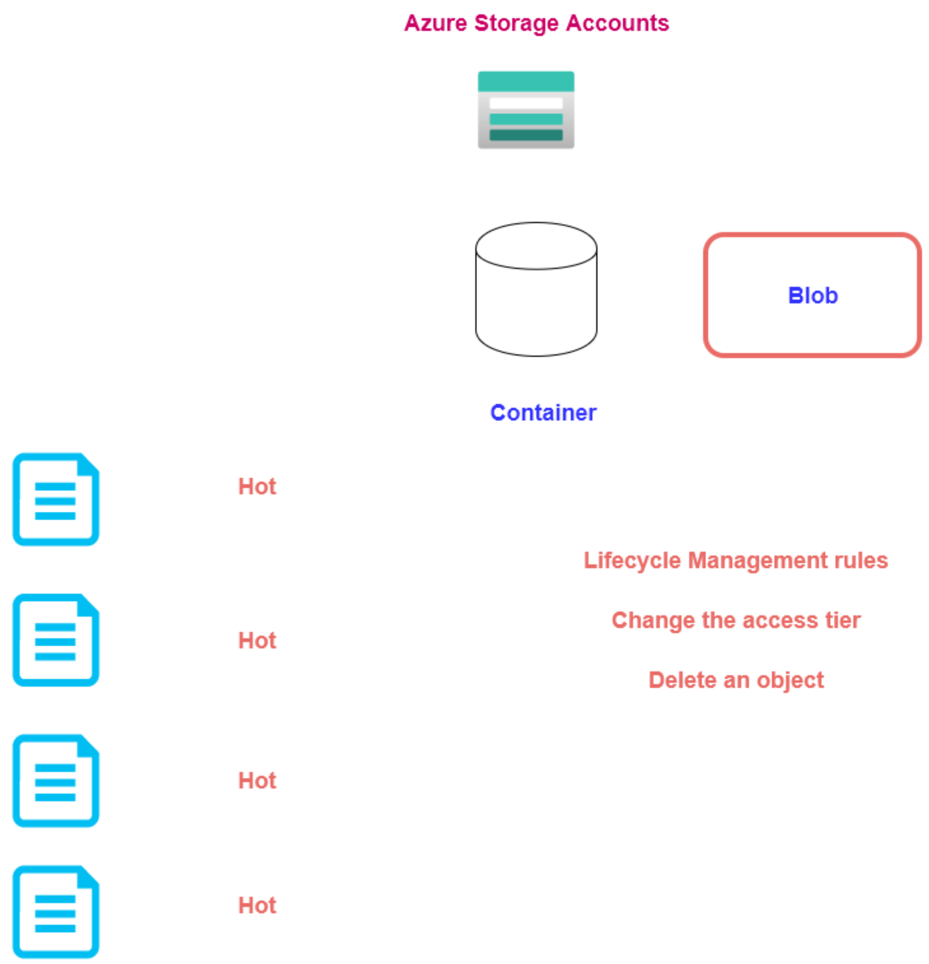
Storing queues
Used for sending and
receiving messages

Used for creating file
shares

Azure Blob versioning

- This feature allows you to maintain previous versions of an object.
- Here you can restore to an earlier version of a blob to recover data if it is erroneously modified or deleted.
- For each subsequent modification of a blob, a newer version of the blob is created.
- Here the blob versions are immutable. They can’t be modified.
- If you enable versioning, you can also disable it at any point in time. But this does not delete existing blobs, versions, or snapshots

Use lifecycle rules to transition objects from one tier to another



Azure Blob snapshots

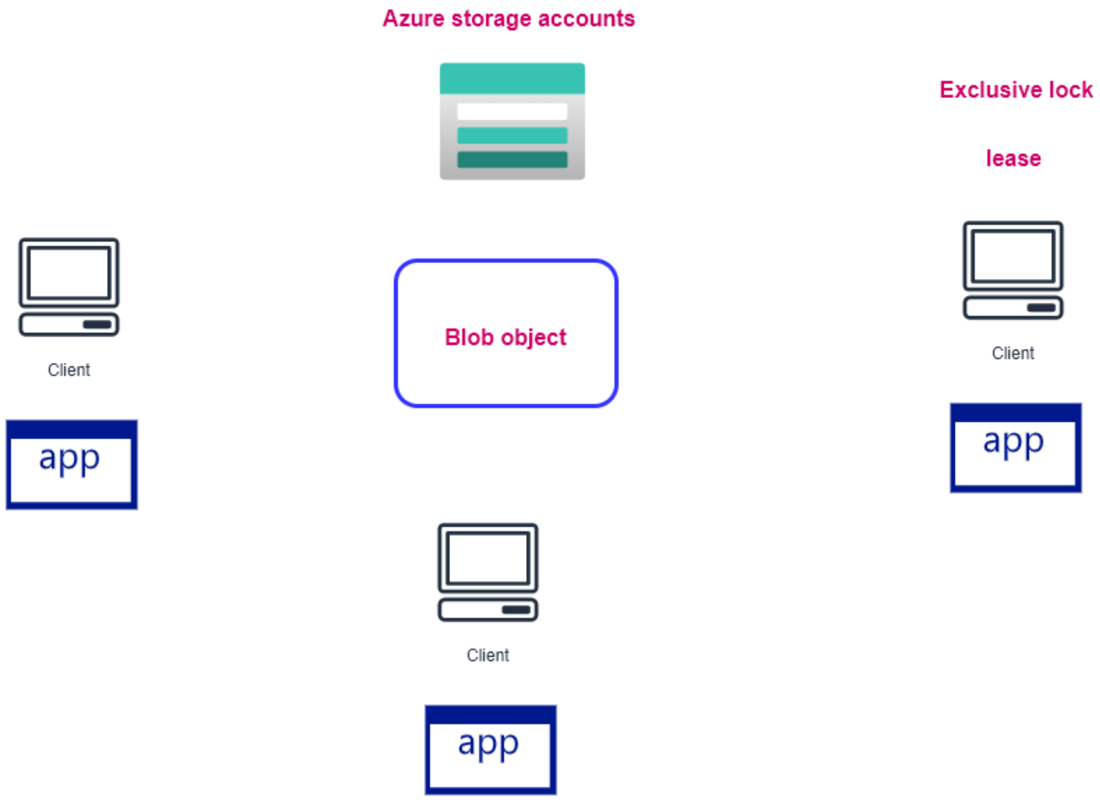
- This takes a read-only version of a blob at a particular point in time.
- You can also make use of the blob versioning feature. This provides a better way to use a previous version of a blob.
- You can create as many snapshots as required of the blob.
- When you delete the blob, the snapshots also have to be deleted.

Azure Blob – Change Feed

- This feature provides transaction logs of all of the changes that occur to the blobs and the blob metadata in the storage account.
- Here you get an ordered, guaranteed, durable, immutable and read-only log of the changes.

- The changes events are recorded in the Apache Avro format specification.
- This file format provides a compact way to store data. And also, a fast way to access the data.
- Currently General Purpose v2 and Blob storage accounts support the change feed feature.

Get locks for your objects with the help of blob leases



Azure Table storage

Azure Table storage

Table storage

Azure Storage Accounts

Data as tables

Stores non-relational structured data

NoSQL data

Here data is stored with the help of key/attributes and values

Follows a schemaless design

Azure Cosmos DB

Use Azure Cosmos DB as your NoSQL data store

Azure Cosmos DB

This is a fully managed NoSQL database

It provides fast response time

It provides high availability for your data

It is highly scalable

It carries out automatic updates and patching to the underlying database engine

SQL API

MongoDB

Gremlin API

Cassandra

Table

JSON documents

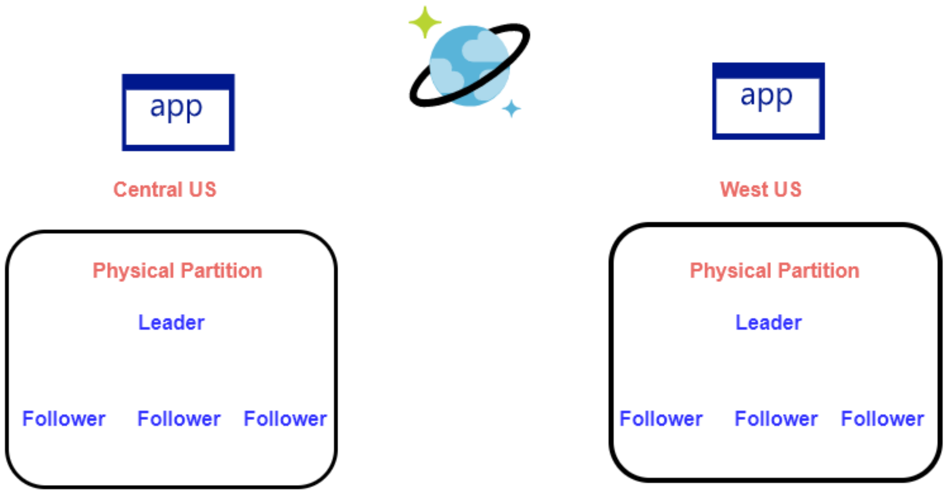
Graph-based database

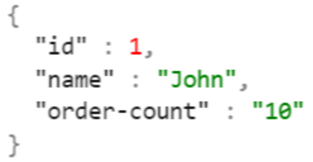
Apache Cassandra

Key-value pairs

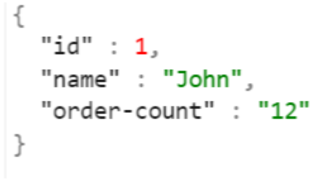
SQL commands

Consistency levels





```
{
  "id" : 1,
  "name" : "John",
  "order-count" : "10"
}
```



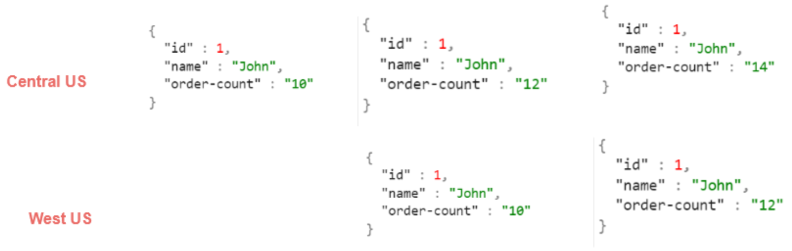
```
{
  "id" : 1,
  "name" : "John",
  "order-count" : "12"
}
```

Consistency

1. Strong - You get consistency but loose on performance

2. Eventual - You win on performance but loose on consistency

3. Bounded staleness - Here the reads can lag behind the writes by at most "K" versions of an item or by "T" time interval



```
{
  "id" : 1,
  "name" : "John",
  "order-count" : "10"
}
```

```
{
  "id" : 1,
  "name" : "John",
  "order-count" : "12"
}
```

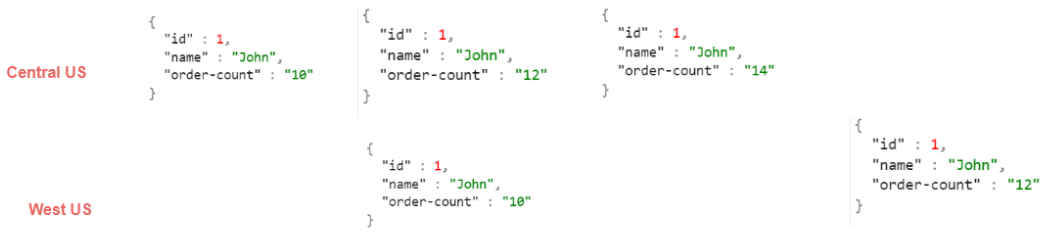
```
{
  "id" : 1,
  "name" : "John",
  "order-count" : "14"
}
```

```
{
  "id" : 1,
  "name" : "John",
  "order-count" : "10"
}
```

```
{
  "id" : 1,
  "name" : "John",
  "order-count" : "12"
}
```

4. Session - Here reads can be guaranteed for the same session

5. Consistent Prefix - Here there is a delay in the reads of the most recent data but you will never see out of order writes



```
{
  "id" : 1,
  "name" : "John",
  "order-count" : "10"
}
```

```
{
  "id" : 1,
  "name" : "John",
  "order-count" : "12"
}
```

```
{
  "id" : 1,
  "name" : "John",
  "order-count" : "14"
}
```

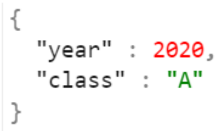
```
{
  "id" : 1,
  "name" : "John",
  "order-count" : "10"
}
```

```
{
  "id" : 1,
  "name" : "John",
  "order-count" : "12"
}
```

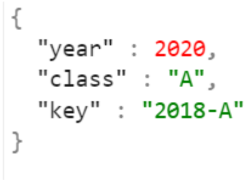
You can use synthetic keys for your partition key

Cosmos DB

Synthetic
partition key

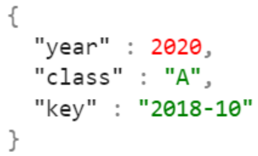


```
{
  "year" : 2020,
  "class" : "A"
}
```



```
{
  "year" : 2020,
  "class" : "A",
  "key" : "2018-A"
}
```

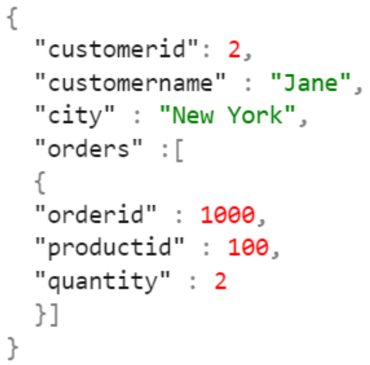
Combine keys



```
{
  "year" : 2020,
  "class" : "A",
  "key" : "2018-10"
}
```

Generate a
random suffix

Embedding data



```
{
  "customerid": 2,
  "customername" : "Jane",
  "city" : "New York",
  "orders" :[
  {
  "orderid" : 1000,
  "productid" : 100,
  "quantity" : 2
  }]
}
```

1. Here we are denormalizing
the data

When to embed data

1. When there are contained relationships between entities

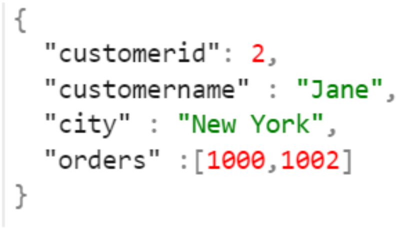
2. When there is a one to few relationship between entities

3. When the embedded data changes infrequently

4. When the embedded data is queried frequently together

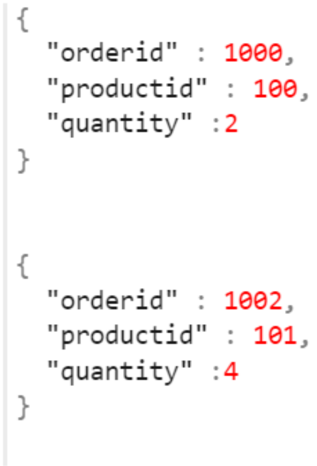
Referencing data

Seperate Document



```
{
  "customerid": 2,
  "customername" : "Jane",
  "city" : "New York",
  "orders" :[1000,1002]
}
```

Seperate Document



```
{
  "orderid" : 1000,
  "productid" : 100,
  "quantity" :2
}


{
  "orderid" : 1002,
  "productid" : 101,
  "quantity" :4
}
```

Referencing data

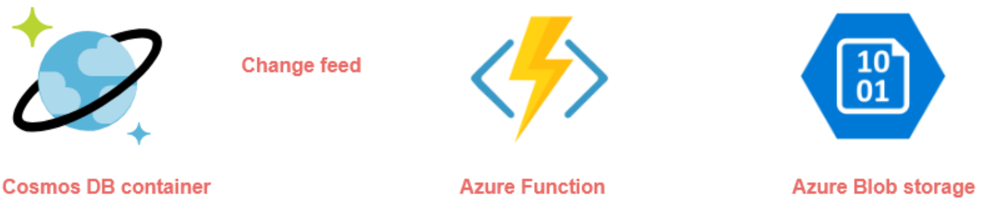
1. When representing one-to-many relationships

2. When representing many-to-many relationships

3. When there are related data changes

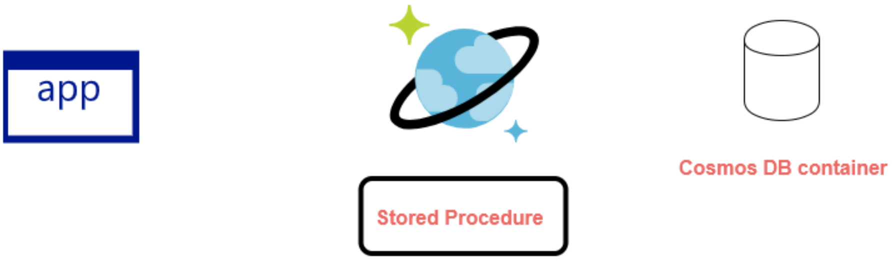
Change Feed, stored procedures and triggers

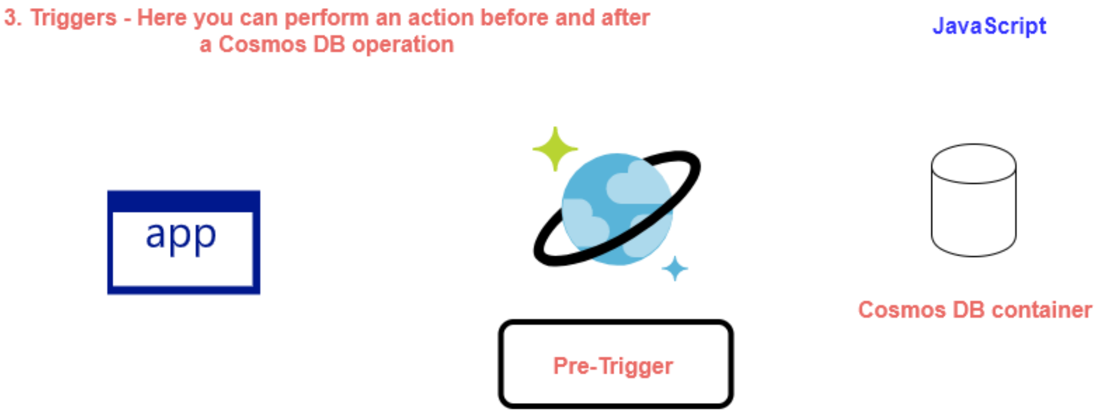
1. Change feed - This listens to all changes in a container and outputs a sorted list of documents that were changed in the order in which they were modified



2. Stored Procedure - Here you can perform an operation at the database level

JavaScript





3. Triggers - Here you can perform an action before and after a Cosmos DB operation

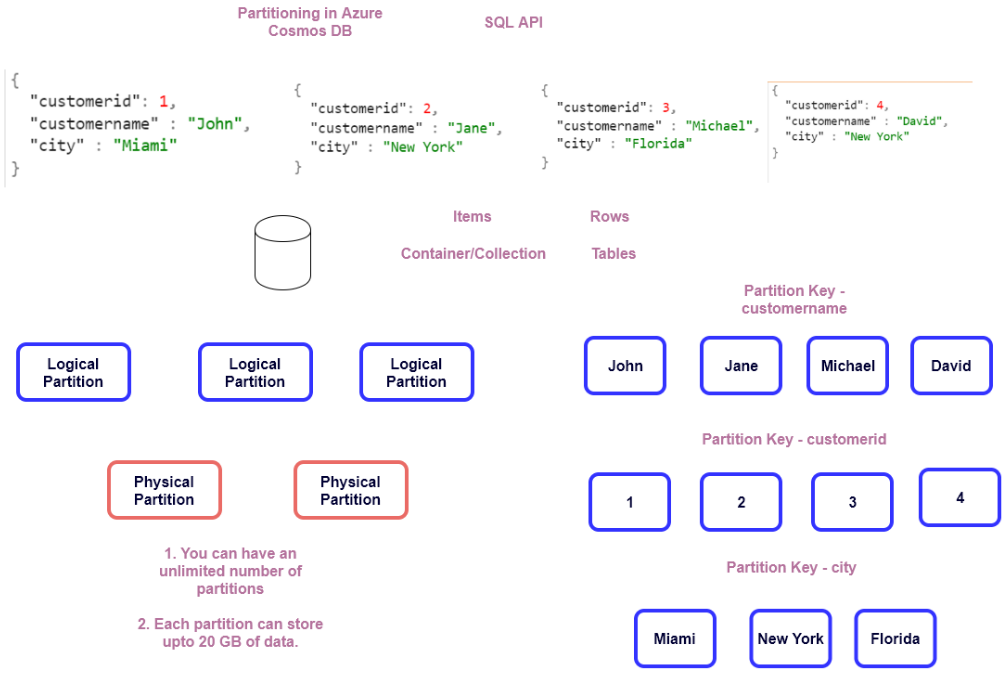
JavaScript

app

Pre-Trigger

Cosmos DB container

Partitioning in Azure Cosmos DB



Partitioning in Azure Cosmos DB

SQL API

```
{
  "customerid": 1,
  "customername" : "John",
  "city" : "Miami"
}
```

```
{
  "customerid": 2,
  "customername" : "Jane",
  "city" : "New York"
}
```

```
{
  "customerid": 3,
  "customername" : "Michael",
  "city" : "Florida"
}
```

```
{
  "customerid": 4,
  "customername" : "David",
  "city" : "New York"
}
```

Items Rows

Container/Collection Tables

Partition Key - customername

John Jane Michael David

Logical Partition Logical Partition Logical Partition

Partition Key - customerid

1 2 3 4

Physical Partition Physical Partition

Partition Key - city

Miami New York Florida

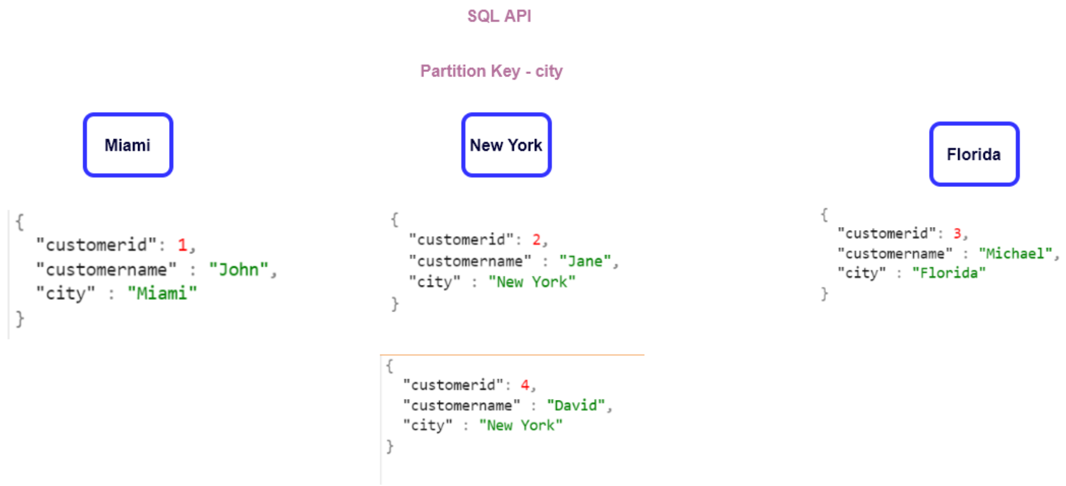
1. You can have an unlimited number of partitions

2. Each partition can store upto 20 GB of data.

Time-to-live

- You can specify a time to live for all items in a container for just a specific item.
- After the time to live setting is reached, the item will be deleted automatically.
- The item will be deleted using the left-over request units.
- If there are no left-over request units, the deletion of the item will be delayed.

- You can set the value of the time to live at the container or item level.



SQL API

Partition Key - city

```
{
  "customerid": 1,
  "customername" : "John",
  "city" : "Miami"
}
```

```
{
  "customerid": 2,
  "customername" : "Jane",
  "city" : "New York"
}
```

```
{
  "customerid": 3,
  "customername" : "Michael",
  "city" : "Florida"
}
```

```
{
  "customerid": 4,
  "customername" : "David",
  "city" : "New York"
}
```

1. Each item has an item id that uniquely identifies the item within a partition

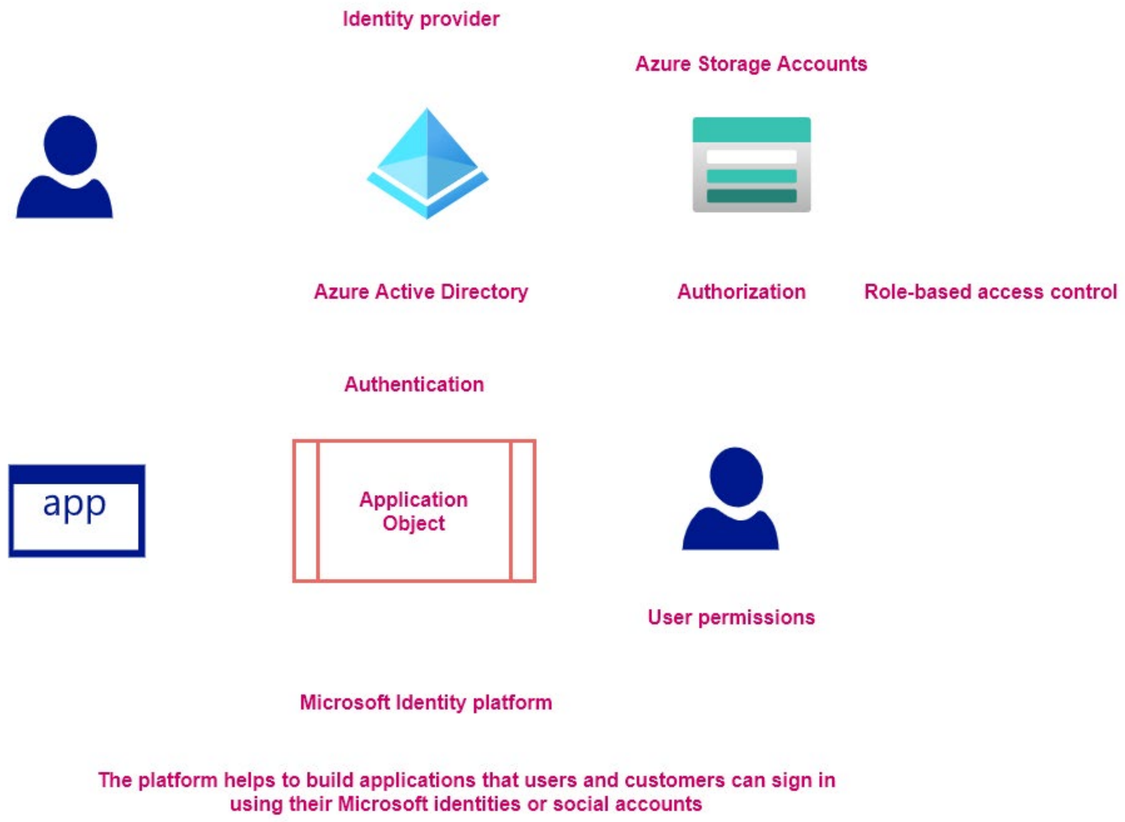
2. The item id + the partition key then uniquely identifies the item within a container

3. You can set the item id when you create the item. Or the container will generate an item id.

Implement Azure Security

Application Objects

Use Application Objects to represent your applications



Identity provider

Azure Storage Accounts

Azure Active Directory

Authorization

Role-based access control

Authentication

Application Object

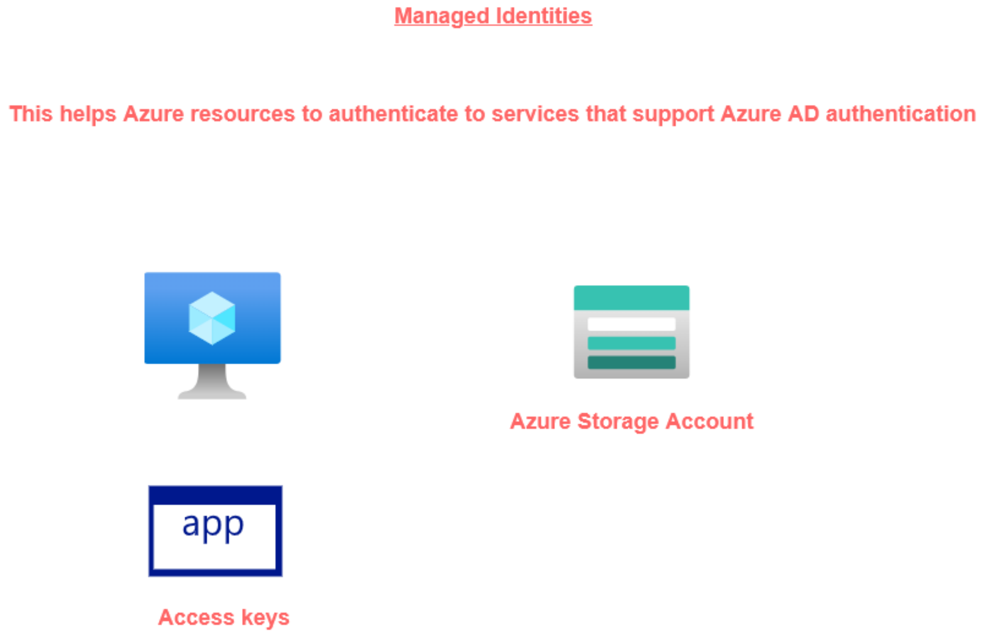
User permissions

Microsoft Identity platform

The platform helps to build applications that users and customers can sign in using their Microsoft identities or social accounts

Using Managed identities

This helps to securely access resources in Azure

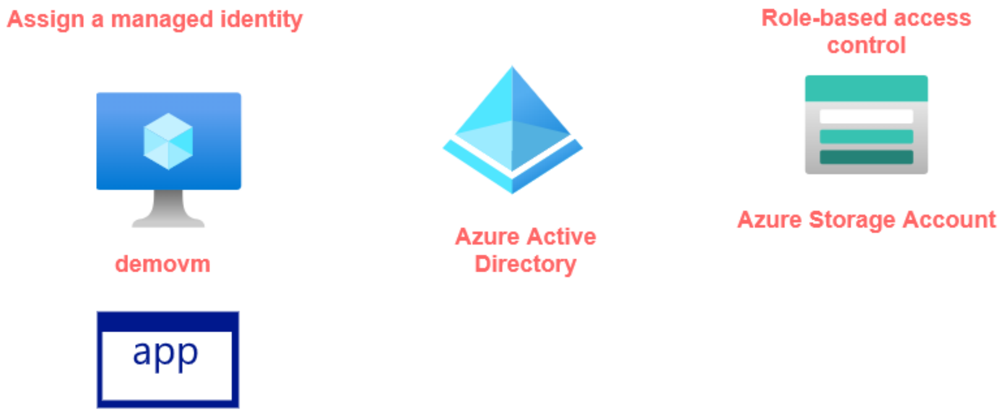


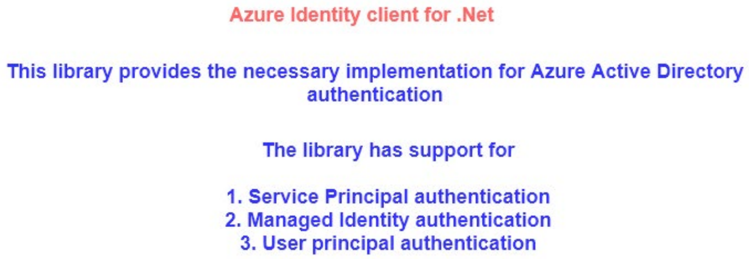
Managed Identities

This helps Azure resources to authenticate to services that support Azure AD authentication

Azure Storage Account

Access keys

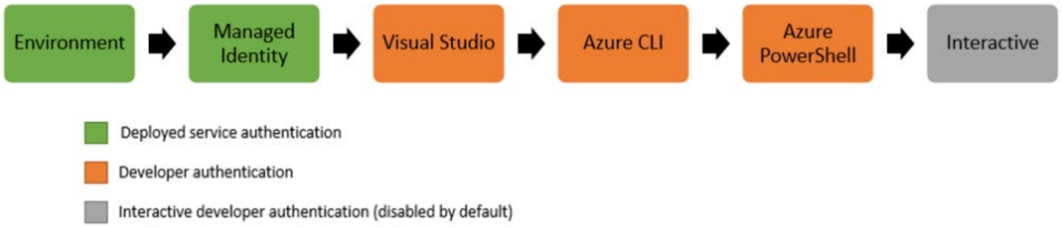



Azure Identity client for .Net

This library provides the necessary implementation for Azure Active Directory authentication

The library has support for

1. Service Principal authentication
2. Managed Identity authentication
3. User principal authentication


Environment ➡ Managed Identity ➡ Visual Studio ➡ Azure CLI ➡ Azure PowerShell ➡ Interactive

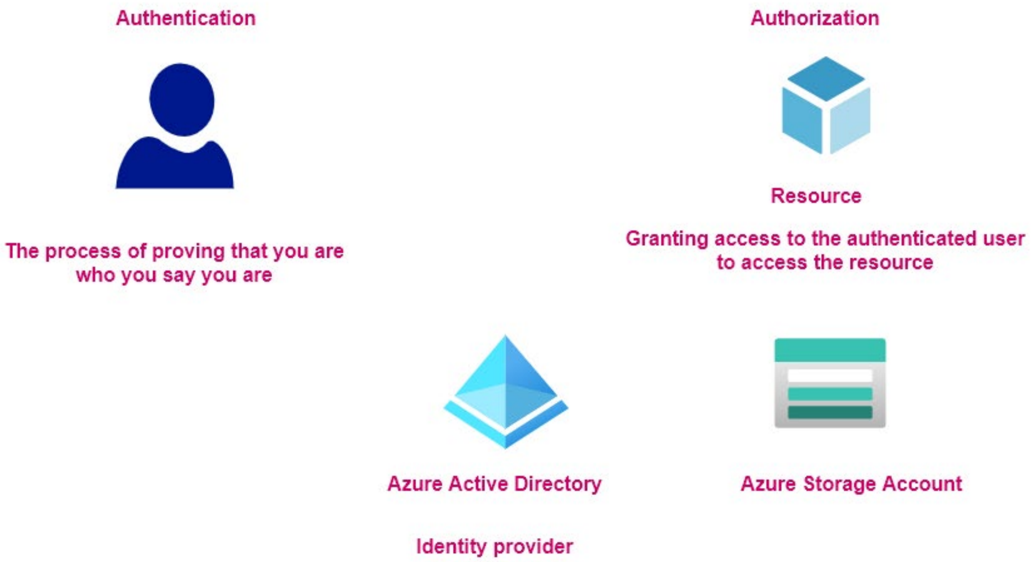
🟩 Deployed service authentication

🟧 Developer authentication

▨ Interactive developer authentication (disabled by default)

Reference link - https://docs.microsoft.com/en-us/dotnet/api/overview/azure/identity-readme

The basics of authentication



Authentication

The process of proving that you are who you say you are

Authorization

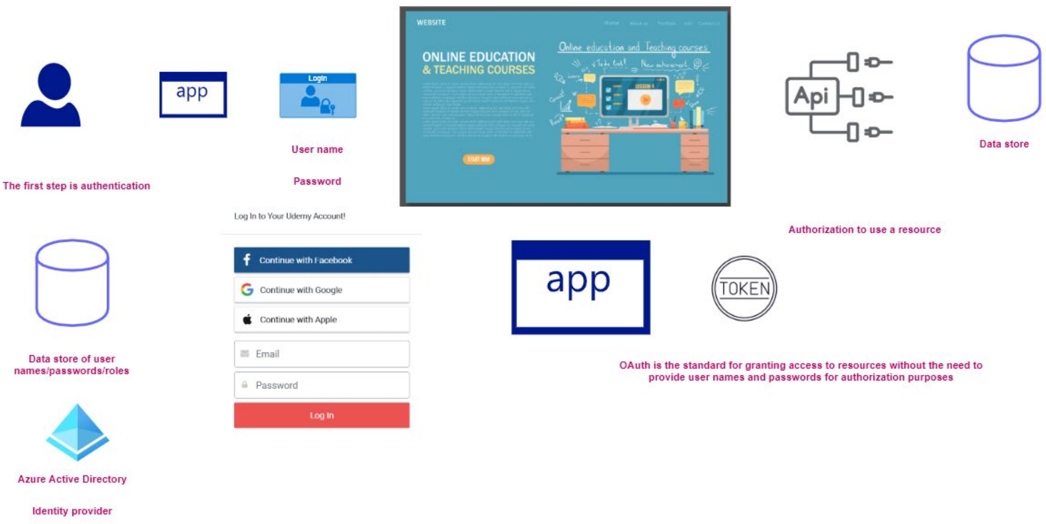
Resource

Granting access to the authenticated user to access the resource

Azure Active Directory

Identity provider

Azure Storage Account



app
Login
WEBSITE
ONLINE EDUCATION & TEACHING COURSES
Online education and Teaching courses
Api
app
TOKEN

The first step is authentication

User name

Password

Data store

Authorization to use a resource

Log In to Your Udemy Account!

Continue with Facebook

Continue with Google

Continue with Apple

Email

Password

Log In

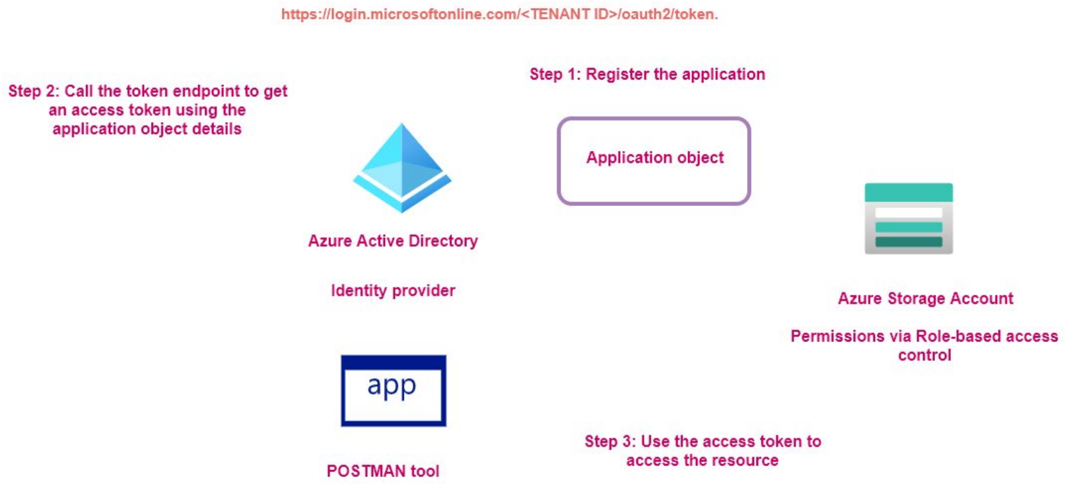
Data store of user names/passwords/roles

Azure Active Directory

Identity provider

OAuth is the standard for granting access to resources without the need to provide user names and passwords for authorization purposes

Using the POSTMAN tool to access blob storage



https://login.microsoftonline.com/<TENANT ID>/oauth2/token.

Step 2: Call the token endpoint to get an access token using the application object details

Step 1: Register the application

Application object

Azure Active Directory

Identity provider

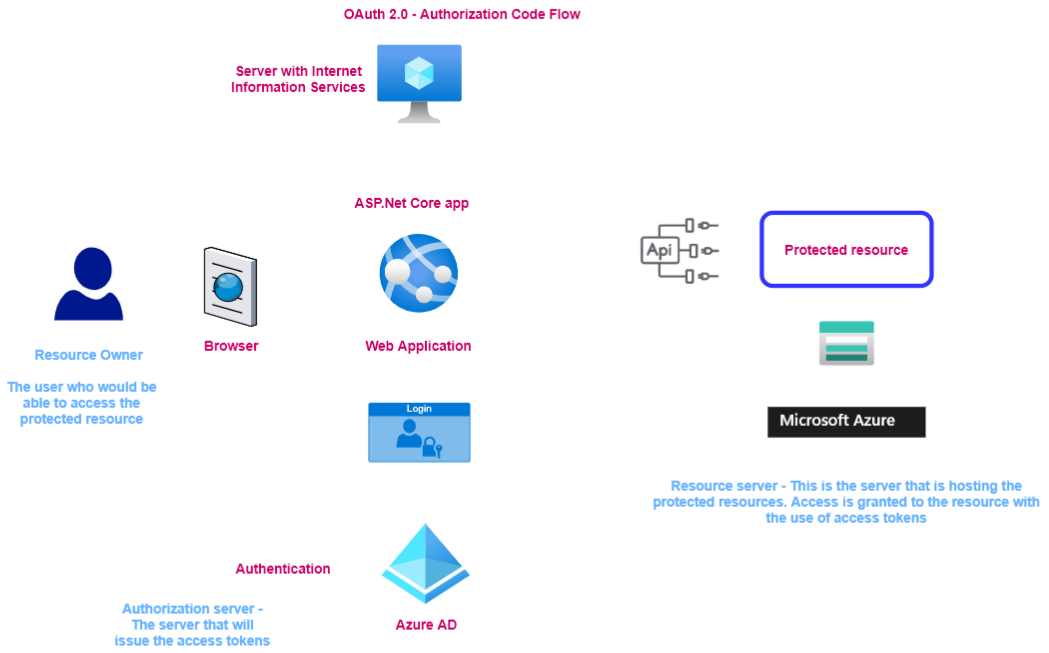
Azure Storage Account

Permissions via Role-based access control

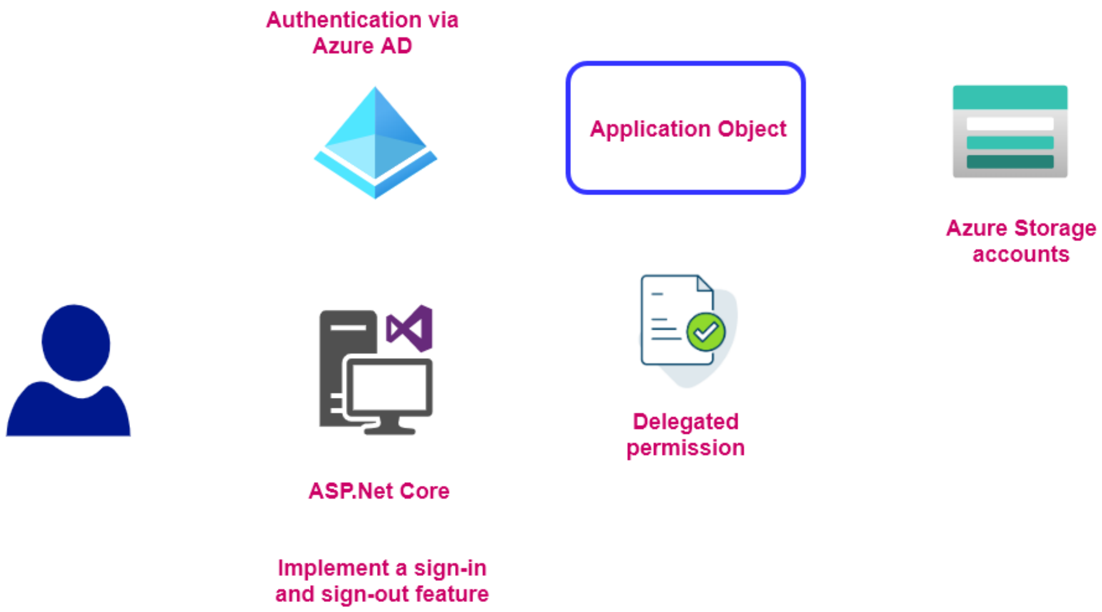
POSTMAN tool

Step 3: Use the access token to access the resource

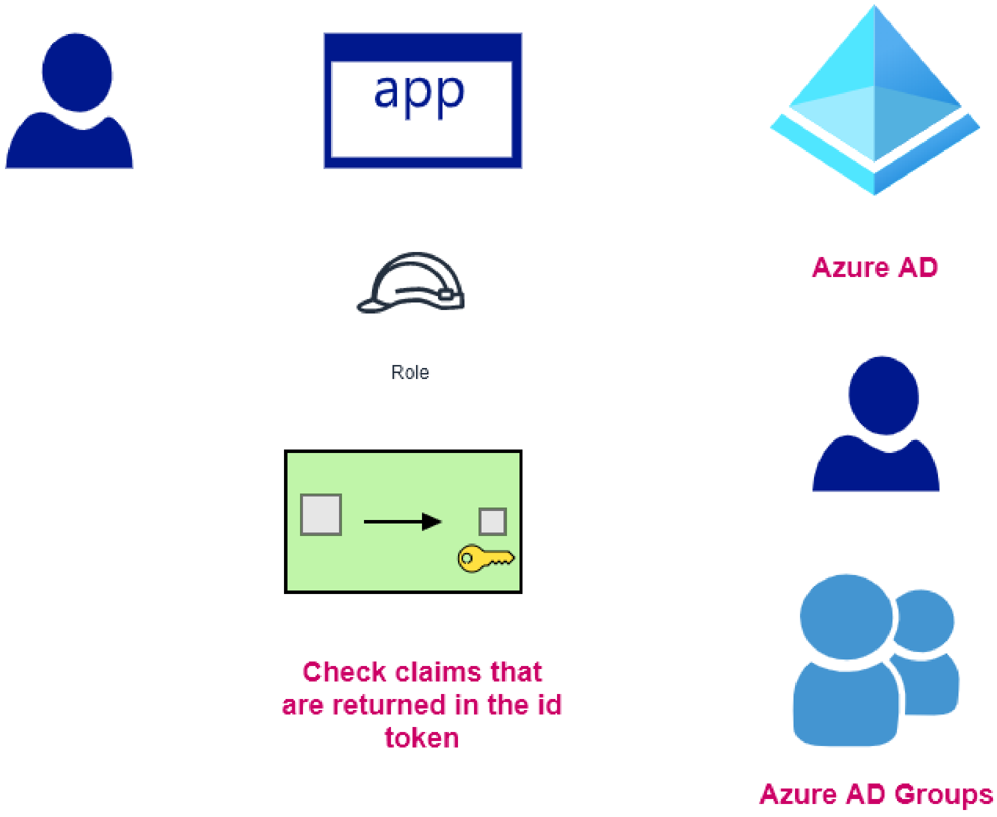
OAuth


OAuth 2.0 - Authorization Code Flow
Server with Internet Information Services
ASP.Net Core app
Web Application
Resource Owner
The user who would be able to access the protected resource
Browser
Login
Api
Protected resource
Microsoft Azure
Resource server - This is the server that is hosting the protected resources. Access is granted to the resource with the use of access tokens
Authentication
Authorization server - The server that will issue the access tokens
Azure AD

Getting the access token



Azure AD Group claims



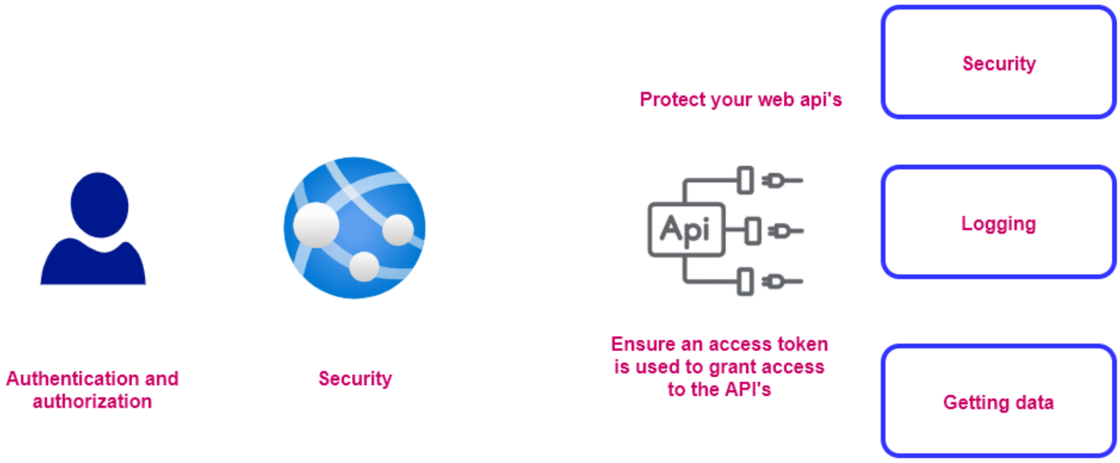
Azure AD

Role

Check claims that are returned in the id token

Azure AD Groups

Protecting your API's



Protect your web api's

Security

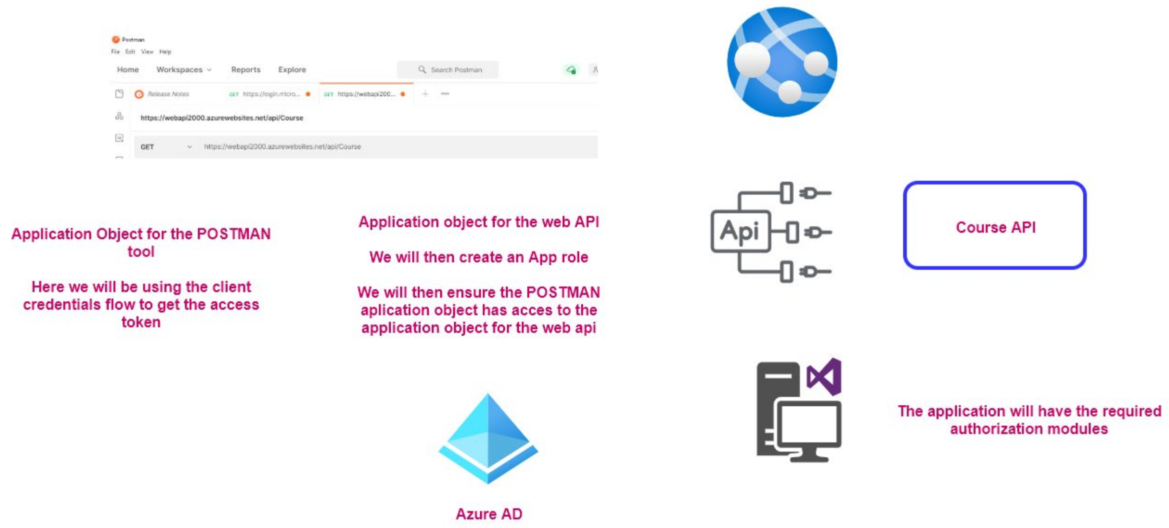
Logging

Getting data

Authentication and authorization

Security

Ensure an access token is used to grant access to the API's



https://webapi2000.azurewebsites.net/api/Course

GET https://webapi2000.azurewebsites.net/api/Course

Application Object for the POSTMAN tool

Here we will be using the client credentials flow to get the access token

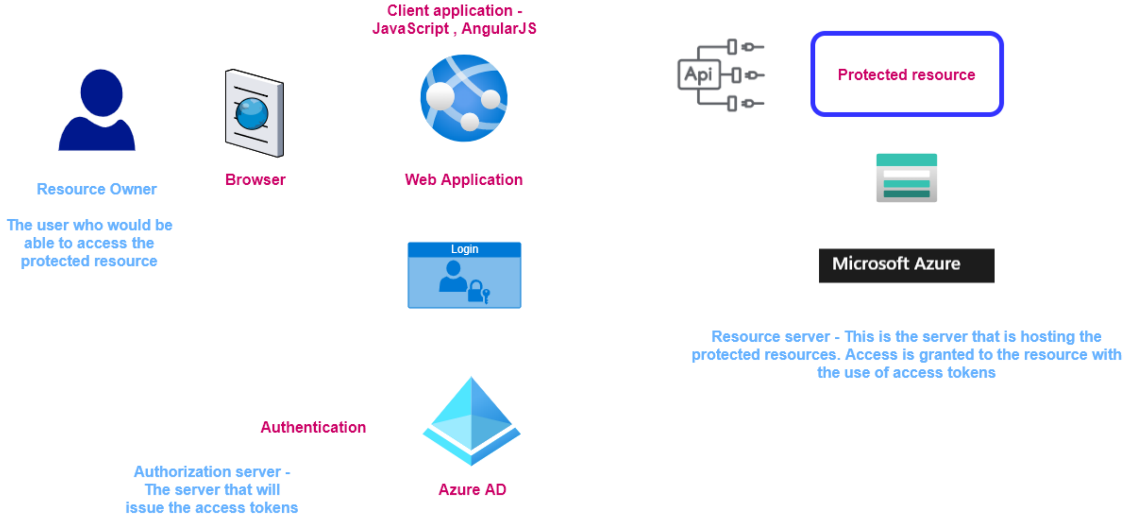
Application object for the web API

We will then create an App role

We will then ensure the POSTMAN aplication object has acces to the application object for the web api

The application will have the required authorization modules

OAuth Implicit flow


Client application - JavaScript , AngularJS

The user who would be able to access the protected resource

Resource server - This is the server that is hosting the protected resources. Access is granted to the resource with the use of access tokens

Authorization server - The server that will issue the access tokens

Monitor, troubleshoot and optimize solutions

Application Insights

- Application Performance Management service for web developers.
- You can use this tool to monitor your applications.
- It can help developers detect anomalies in the application.

- It can help diagnose issues.
- It can also help understand how users use your application.
- It also helps you improve performance and usability of your application.
- **How does it work**
- You install a small instrumentation package within your application.
- You can see the statistics of your application locally in Visual Studio as you run your application.
- You can also use the Application Insights resource in Azure to monitor your application.

- **What are the different aspects monitored by Application Insights**
- Request rates, the response times and failure rates – This is done at the page level.
- Exception recorded by your application.
- Page views and their load performance as reported from the user's browser.
- User and session counts.
- Performance counters of the underlying Windows or Linux Machines.
- Diagnostic trace logs from your application.
- Any custom events or metrics that the developer writes themselves in the code.

- Understanding how your users use the application
- **Funnels** – You can create a funnel from one stage to another stage of your application.
- You can then see how users are progressing through the stages of the funnel.

- **User Flows** – This helps visualize how users navigate between pages in your site. This can help answer question such as
  - Does the user navigate away from a page on your site
  - What do users click on a page on your site
  - Where are the places where users churn most on your site
  - Are there places where users repeat the same action over and over

- **Impact** – This helps decide if a page is having an impact on your application.
- It can help answer the question as to whether the page load time is impacting how many people convert on a page in the application.
- **Retention** – This helps you understand how many users return to your application.
- It can also help understand if users are able to perform certain tasks in your application.

Azure Front Door Caching and Compression

By default, Azure Front Door has the capability to deliver large files. If a file is large, it splits the file into chunks of 8 MB. If the file is not in the cache, the large file is retrieved from the origin. Once the first 8 MB chunk is received , it is sent to the user and then in parallel , the next 8 MB chunk is retrieved from the origin.

Azure Front Door also supports compression. It supports all of the following MIME types. But the file needs to be in size between 1 KB and 8 MB. It supports both gzip and Brotli compression.

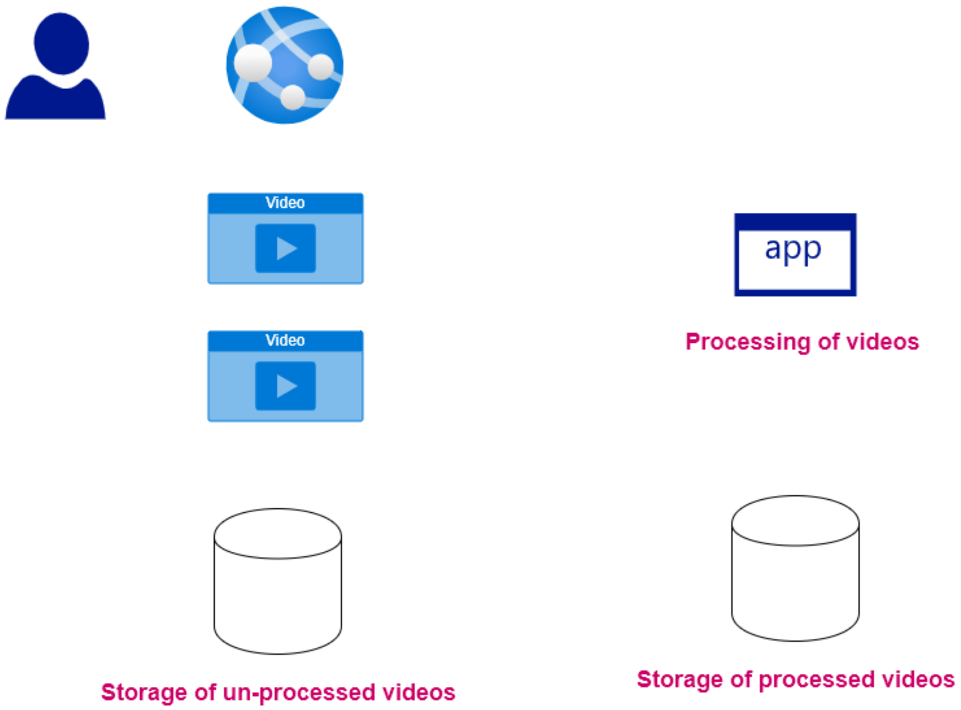
- "application/eot"
- "application/font"
- "application/font-sfnt"
- "application/javascript"
- "application/json"

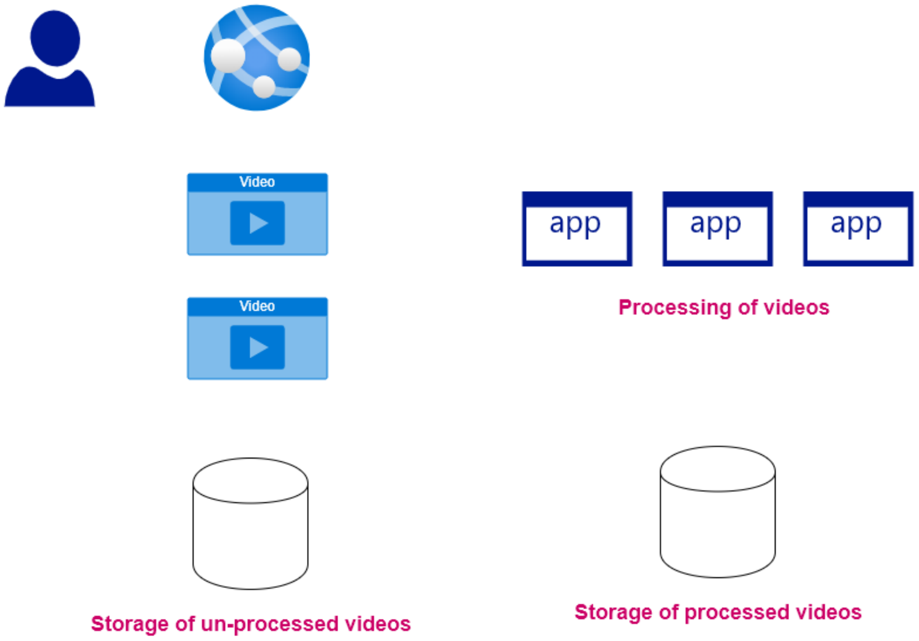
- "application/opentype"
- "application/otf"
- "application/pkcs7-mime"
- "application/truetype"
- "application/ttf",
- "application/vnd.ms-fontobject"
- "application/xhtml+xml"
- "application/xml"
- "application/xml+rss"
- "application/x-font-opentype"
- "application/x-font-truetype"
- "application/x-font-ttf"
- "application/x-httpd-cgi"
- "application/x-mpegurl"
- "application/x-opentype"
- "application/x-otf"
- "application/x-perl"
- "application/x-ttf"
- "application/x-javascript"
- "font/eot"
- "font/ttf"
- "font/otf"
- "font/opentype"
- "image/svg+xml"
- "text/css"
- "text/csv"
- "text/html"
- "text/javascript"
- "text/js", "text/plain"
- "text/richtext"
- "text/tab-separated-values"
- "text/xml"
- "text/x-script"
- "text/x-component"
- "text/x-java-source"

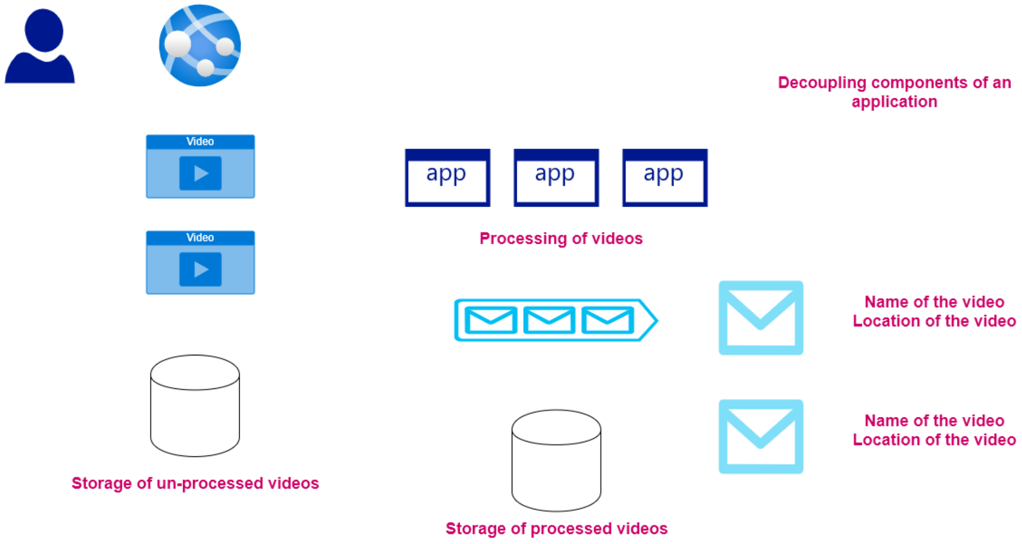
Connect to and consume Azure and third-party services

Queues

The entire purpose of queues is to decouple components of your application

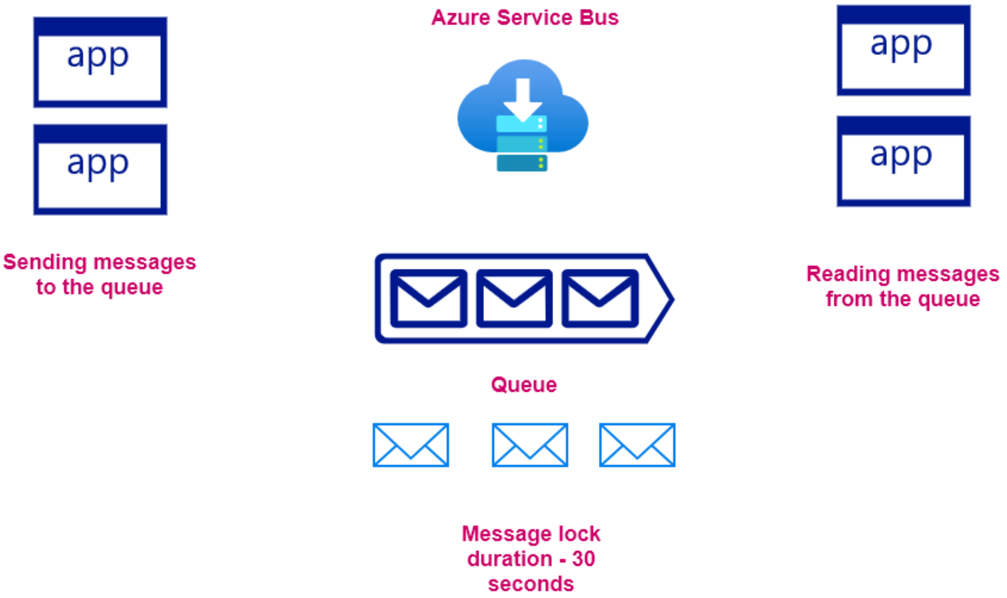




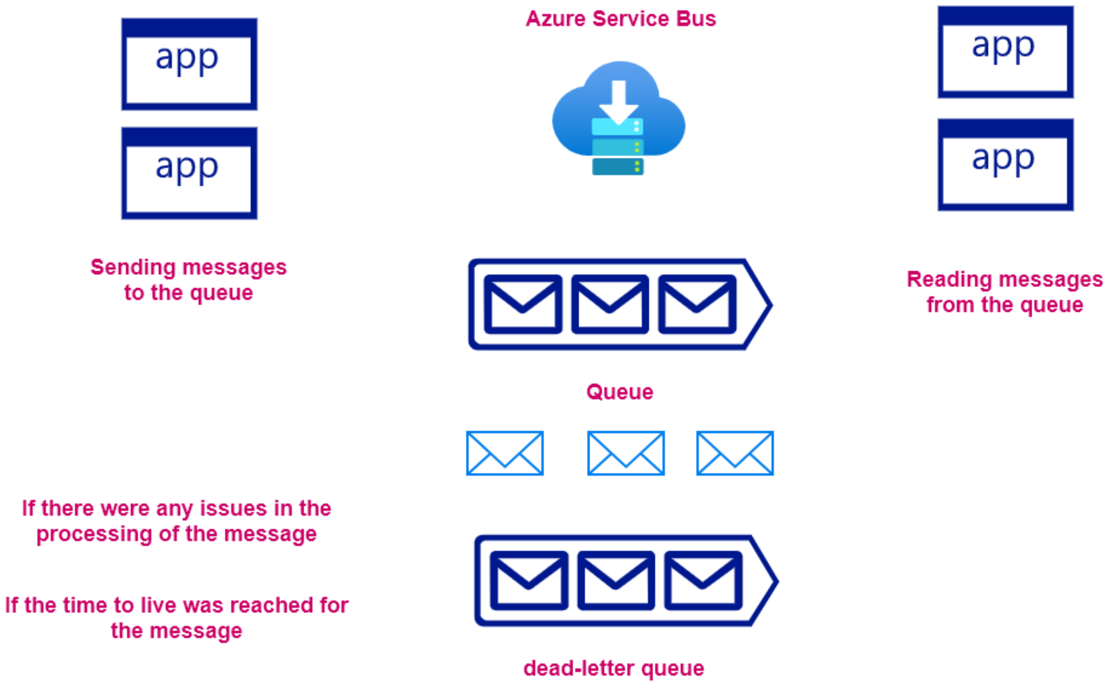


Azure Service Bus

In the PeekLock mode, you can mention what is the duration to lock messages


Azure Service Bus
app
app
Sending messages to the queue
app
app
Reading messages from the queue
Queue
Message lock duration - 30 seconds

Messages that can’t be processed , can be taken from the dead-letter queue



Azure Service Bus

Sending messages to the queue

Reading messages from the queue

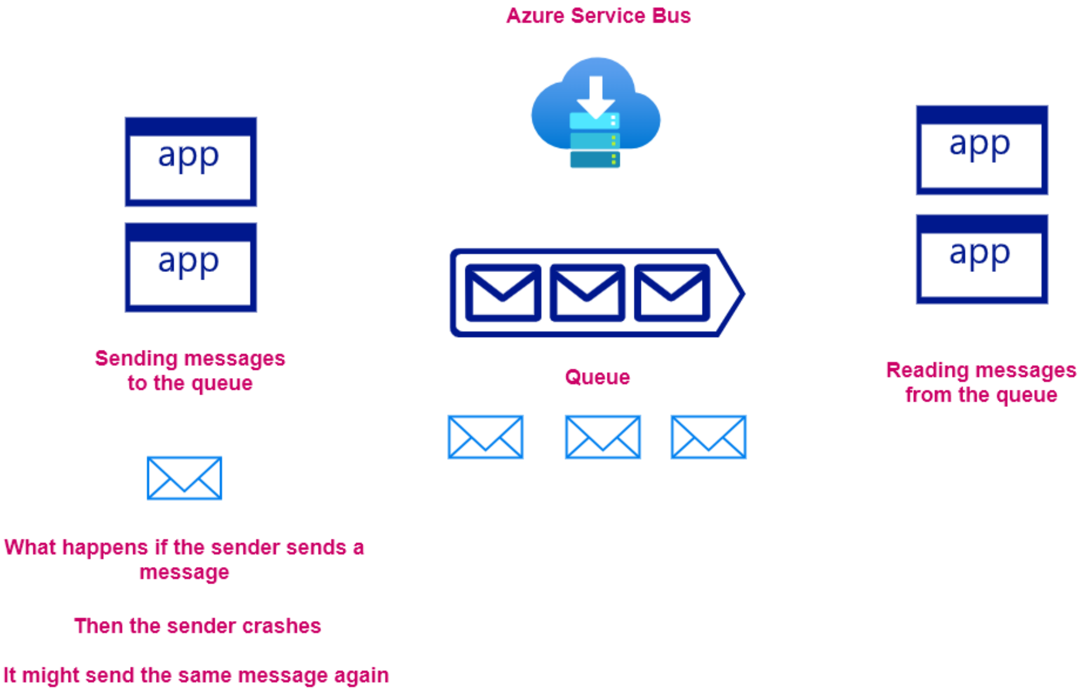
Queue

If there were any issues in the processing of the message

If the time to live was reached for the message

dead-letter queue

You can also detect duplicate messages from Azure Service Bus queues



Azure Service Bus

Sending messages to the queue

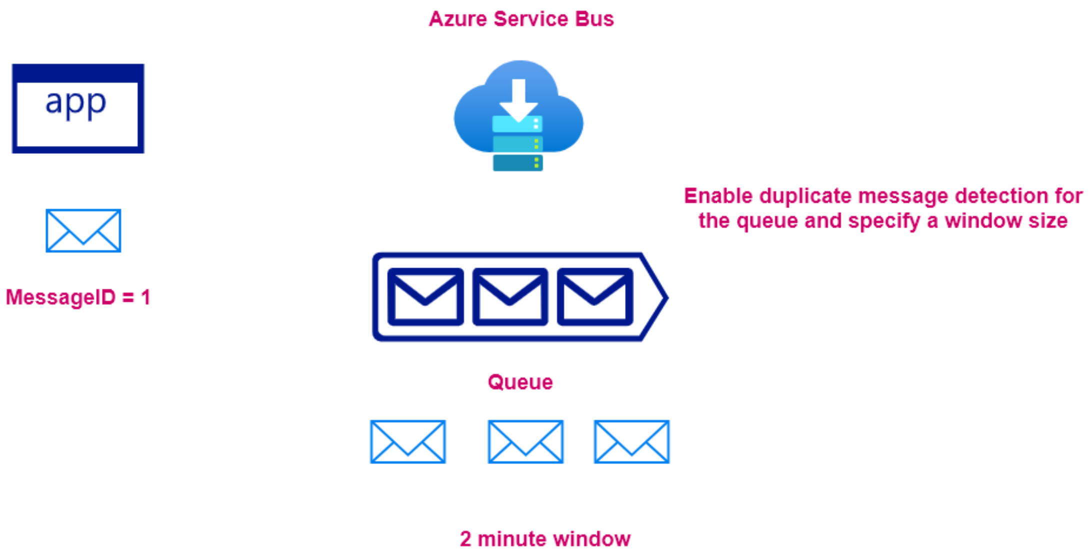
Queue

Reading messages from the queue

What happens if the sender sends a message

Then the sender crashes

It might send the same message again



Azure Service Bus

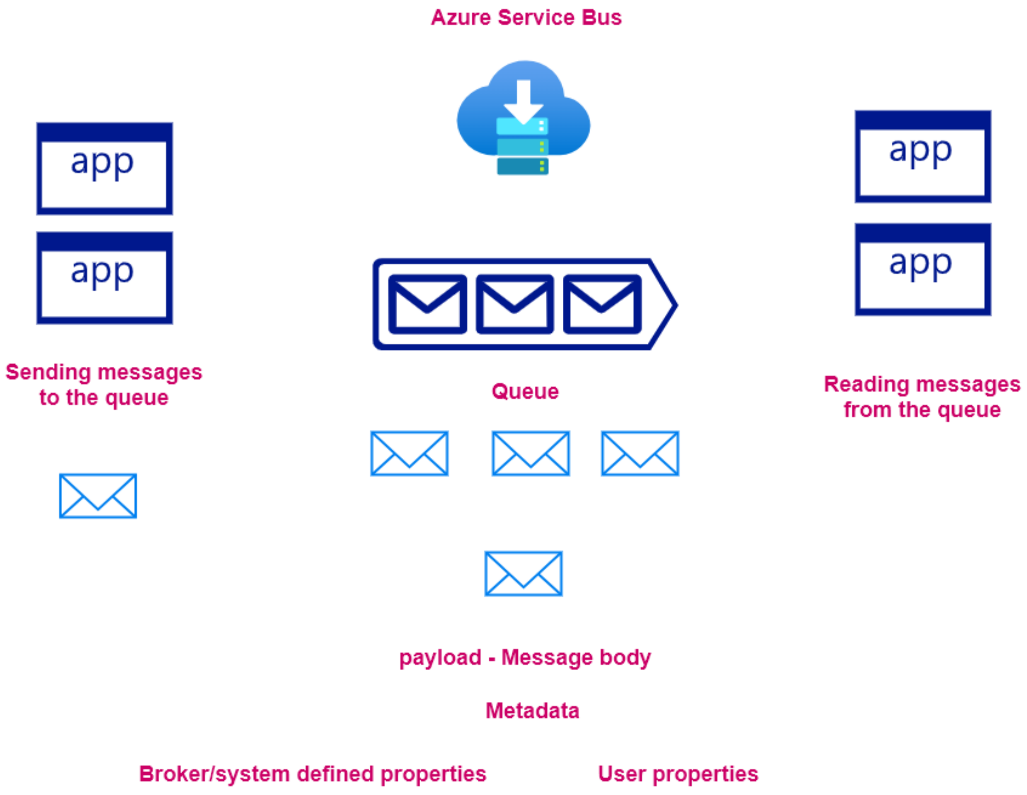
MessageID = 1

Enable duplicate message detection for the queue and specify a window size

Queue

2 minute window

Understanding the different parts of a message



Azure Service Bus

Sending messages to the queue

Queue

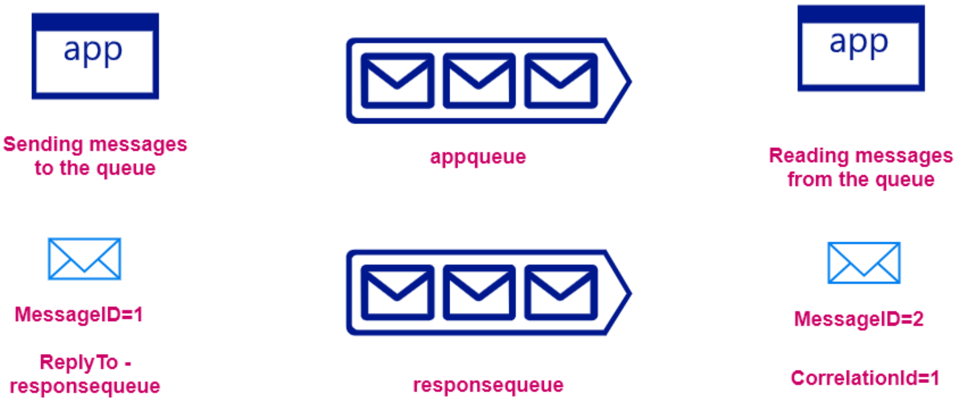
Reading messages from the queue

payload - Message body

Metadata

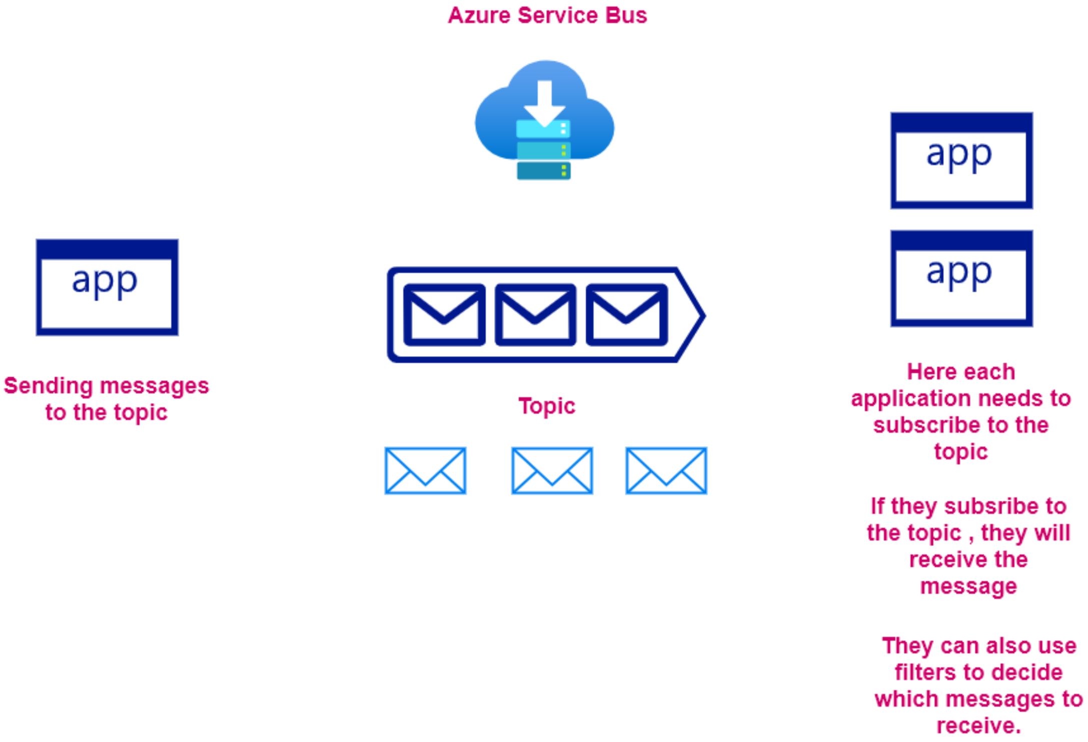
Broker/system defined properties

User properties



Azure Service Bus topics

Here subscribers can subscribe to get messages from the topic

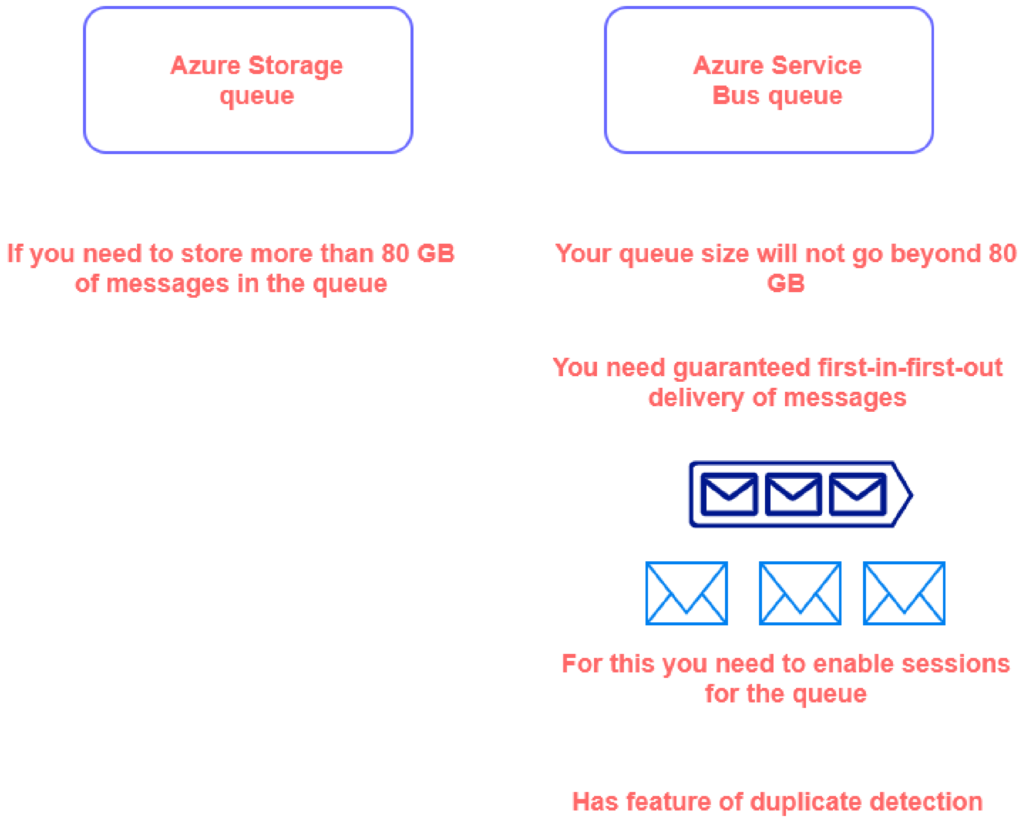

Azure Service Bus
app
Sending messages to the topic
Topic
app
app
Here each application needs to subscribe to the topic
If they subsribe to the topic , they will receive the message
They can also use filters to decide which messages to receive.

Azure Service Bus Topic filters

- Subscriptions can use filters to decide on which messages they want to receive.
- SQL Filters – Here you can use SQL-like conditions for the evaluation. This can be done against the system or user defined properties.

- For system-defined properties , the properties must be prefixed with sys.
- Boolean filters – This is a TrueFilter or FalseFilter.
- Correlation Filters – Here conditions can be used to match against the user’s or system properties.
- Note:- Correlation Filters process faster than SQL filters.

Difference when it comes to Azure storage queues and Service Bus queues



Azure Storage queue

If you need to store more than 80 GB of messages in the queue

Azure Service Bus queue

Your queue size will not go beyond 80 GB

You need guaranteed first-in-first-out delivery of messages

For this you need to enable sessions for the queue

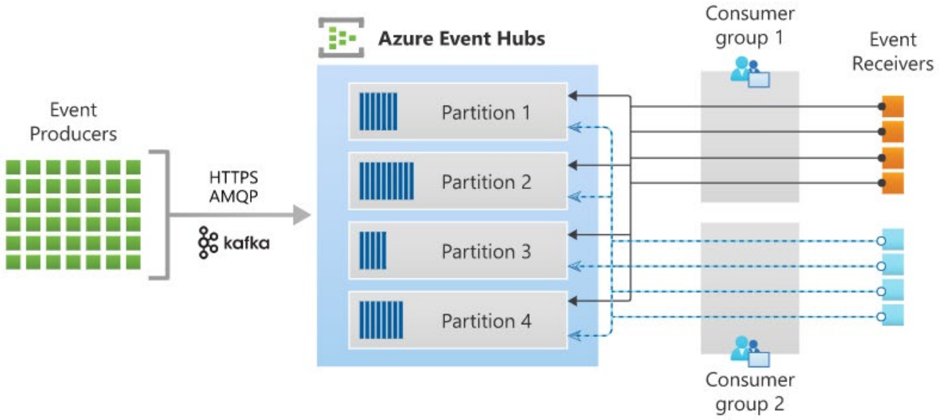
Has feature of duplicate detection

Azure Event Hubs

When you want to ingest large amounts of data , look towards using Azure Event Hubs

You can stream log data , telemetry data, any sort of events to Azure Event Hubs

Event Hubs Architecture



The different components

Event producers - This is an entity that sends data to an event hub. The events can be published using the protocols - HTTPS, AMQP, Apache Kafka

Partitions - The data is split across partitions. This allows for better throughput of your data onto Azure Event Hubs

Consumer groups - This is a view (state,position or offset) of an entire event hub

Throughput - This controls the throughput capacity of Event Hubs

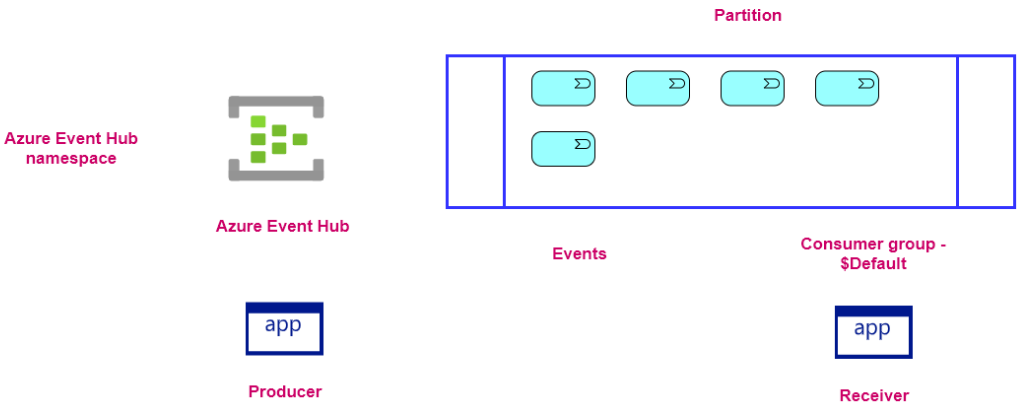
Event Receivers - This is an entity that reads event data

When you send events or data onto Azure Event Hub, they are sent to a partition. The receiver can then receive events via a consumer group

You pre-purchase throughput / billed per hour

Ingress - Up to 1 MB per second or 1000 events per second

Egress - Up to 2 MB per second or 4096 events per second



Partition

Azure Event Hub namespace

Azure Event Hub

Events

Consumer group - $Default

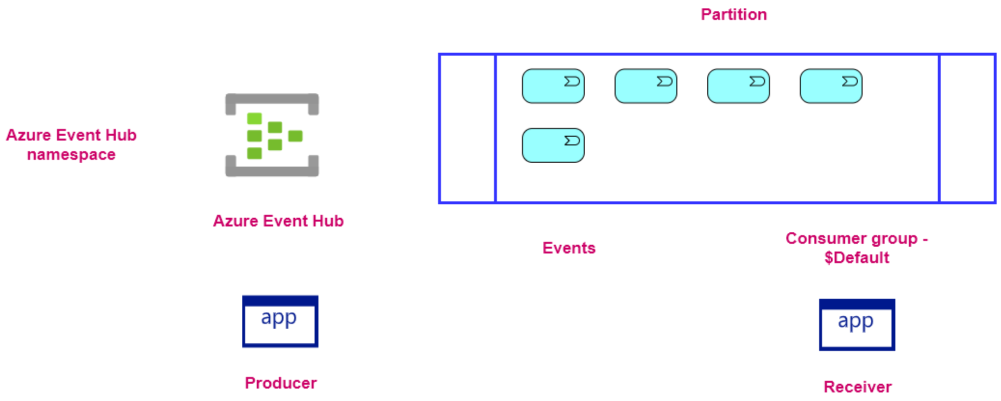
app

Producer

app

Receiver

You can use the EventProcessor class to track the events which have been consumed from Azure Event Hub



Partition

Azure Event Hub namespace

Azure Event Hub

Events

Consumer group - $Default

app

Producer

app

Receiver

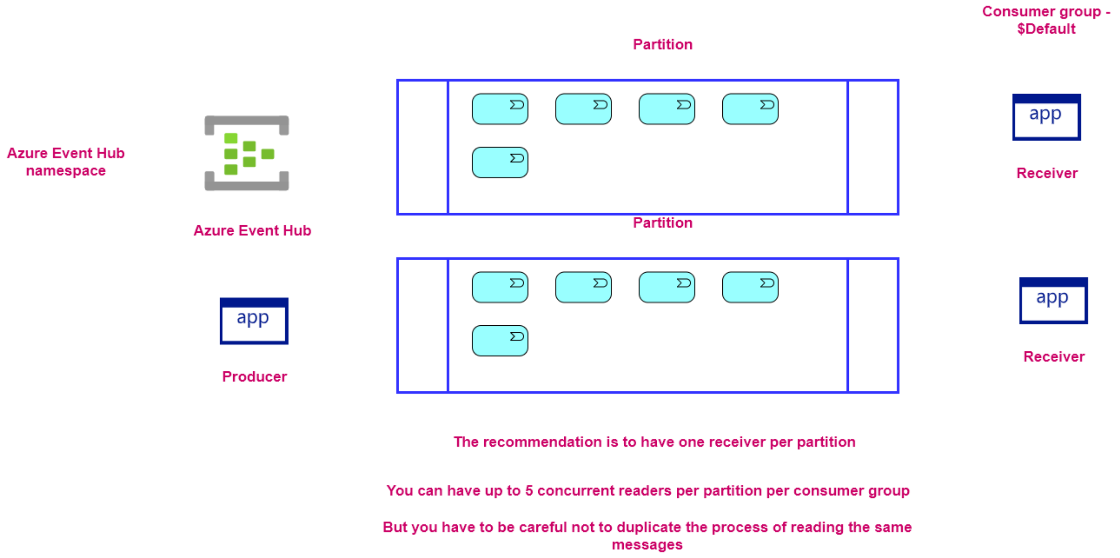
EventProcessor class

This class keeps a note or checkpoint on the events that have been processed.

It cna also receive events from multiple partitions

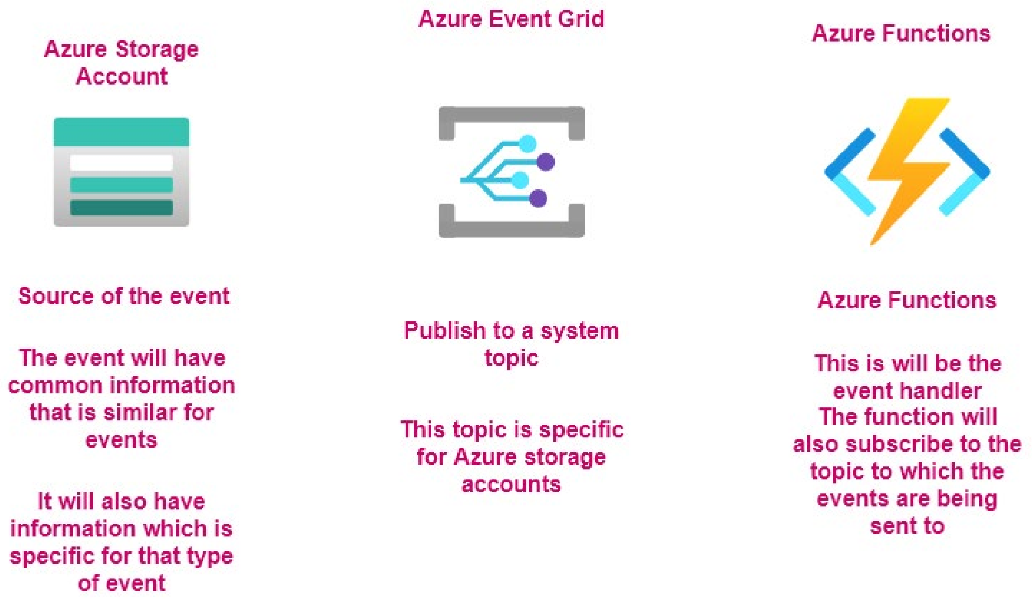
The checkpoint information is kept in a storage account

Recommendations when it comes to readers and partitions


Consumer group - $Default
Partition
Azure Event Hub namespace
Azure Event Hub
Partition
app
Receiver
app
Producer
app
Receiver
The recommendation is to have one receiver per partition
You can have up to 5 concurrent readers per partition per consumer group
But you have to be careful not to duplicate the process of reading the same messages

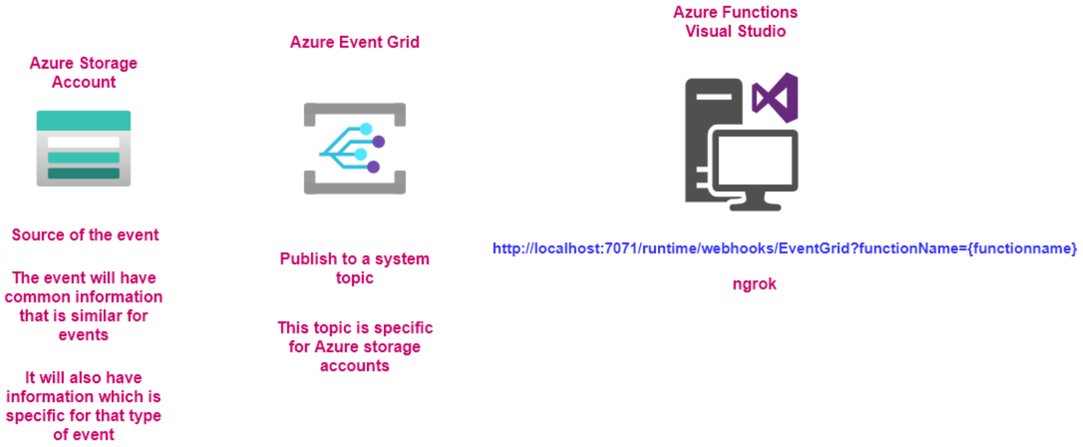
Azure Event Grid

Capture events from various Azure resources with the use of Azure Event Grid


Azure Storage Account
Azure Event Grid
Azure Functions
Source of the event
The event will have common information that is similar for events
It will also have information which is specific for that type of event
Publish to a system topic
This topic is specific for Azure storage accounts
Azure Functions
This is will be the event handler
The function will also subscribe to the topic to which the events are being sent to

Use Azure Event Grid to capture events from an Azure storage account and then use Azure functions as the endpoint to capture the events.

To debug your Azure Event Grid code from Visual Studio in Azure Functions, make sure to use the ngrok tool https://ngrok.com/



Azure Storage Account

Source of the event

The event will have common information that is similar for events

It will also have information which is specific for that type of event

Azure Event Grid

Publish to a system topic

This topic is specific for Azure storage accounts

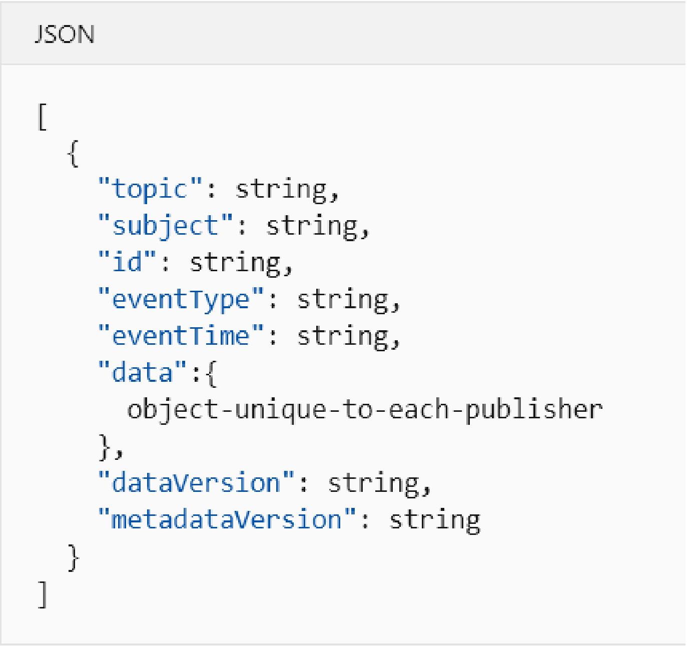
Azure Functions Visual Studio

http://localhost:7071/runtime/webhooks/EventGrid?functionName={functionname}

ngrok

When working with Azure Event Grid, its important to know the schema. Here you know the format of the data that is sent from Azure Event Grid

Reference - https://docs.microsoft.com/en-us/azure/event-grid/event-schema



JSON

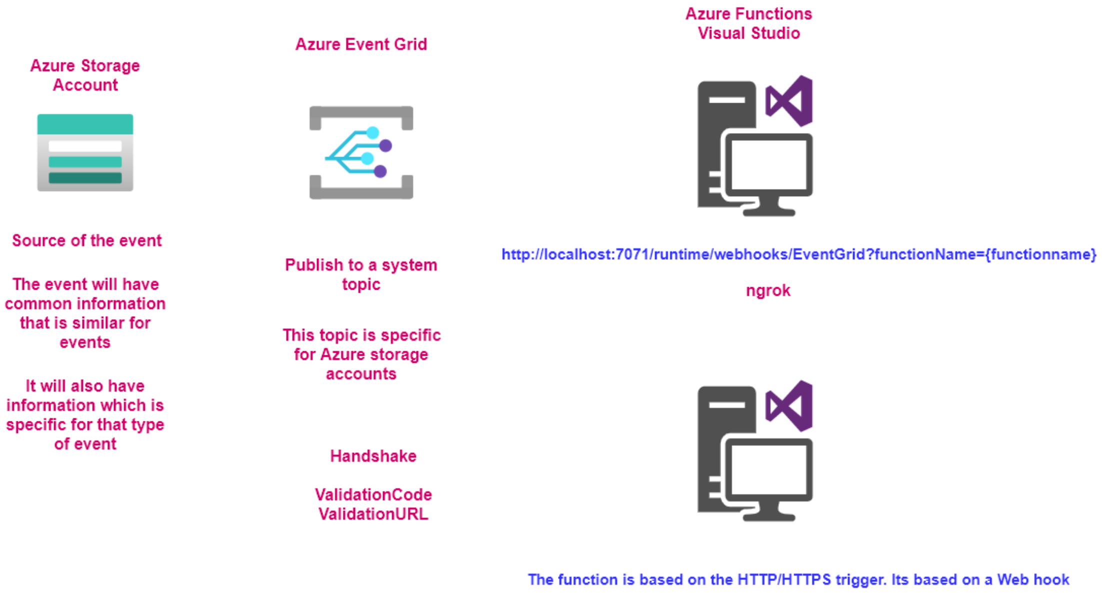
```
[
  {
    "topic": string,
    "subject": string,
    "id": string,
    "eventType": string,
    "eventTime": string,
    "data":{
      object-unique-to-each-publisher
    },
    "dataVersion": string,
    "metadataVersion": string
  }
]
```

The event sources send the events to the Azure Event Grid service in an array

The array can have several event objects

The total size of the array will be up to 1 MB

Remember that if you have your own custom Webhook, then you need to have the handshake code in place to take the validationcode and send it back to the validationURL.



Azure Storage Account

Source of the event

The event will have common information that is similar for events

It will also have information which is specific for that type of event

Azure Event Grid

Publish to a system topic

This topic is specific for Azure storage accounts

Handshake

ValidationCode
ValidationURL

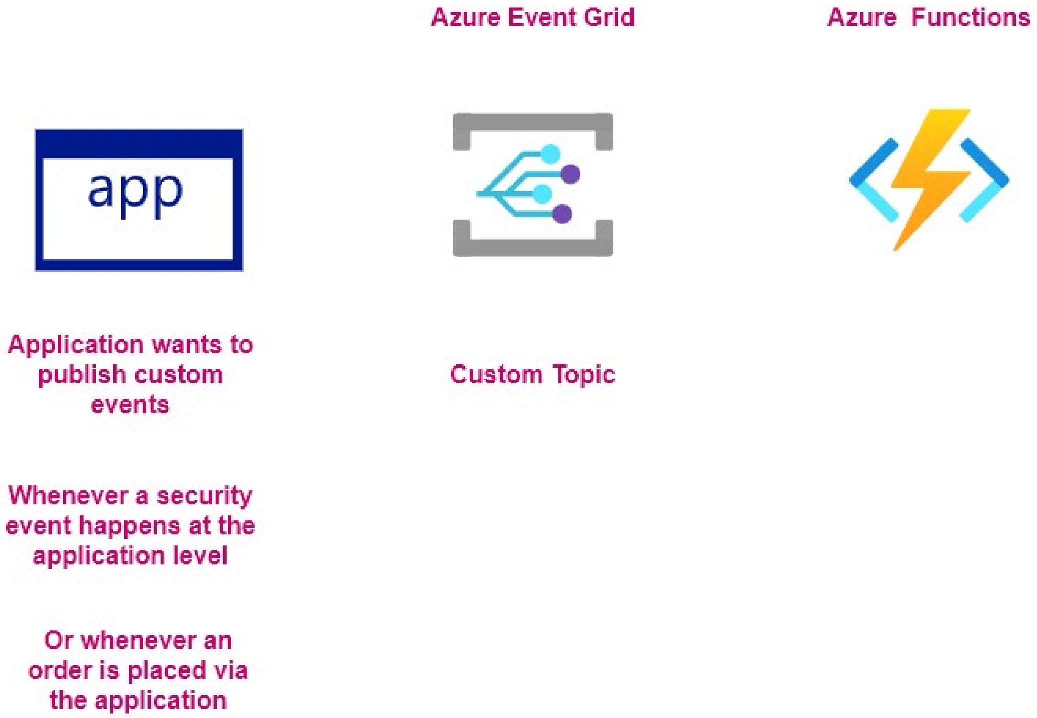
Azure Functions
Visual Studio

http://localhost:7071/runtime/webhooks/EventGrid?functionName={functionname}

ngrok

The function is based on the HTTP/HTTPS trigger. Its based on a Web hook

You can also create your custom topics



Azure Event Grid

Azure Functions

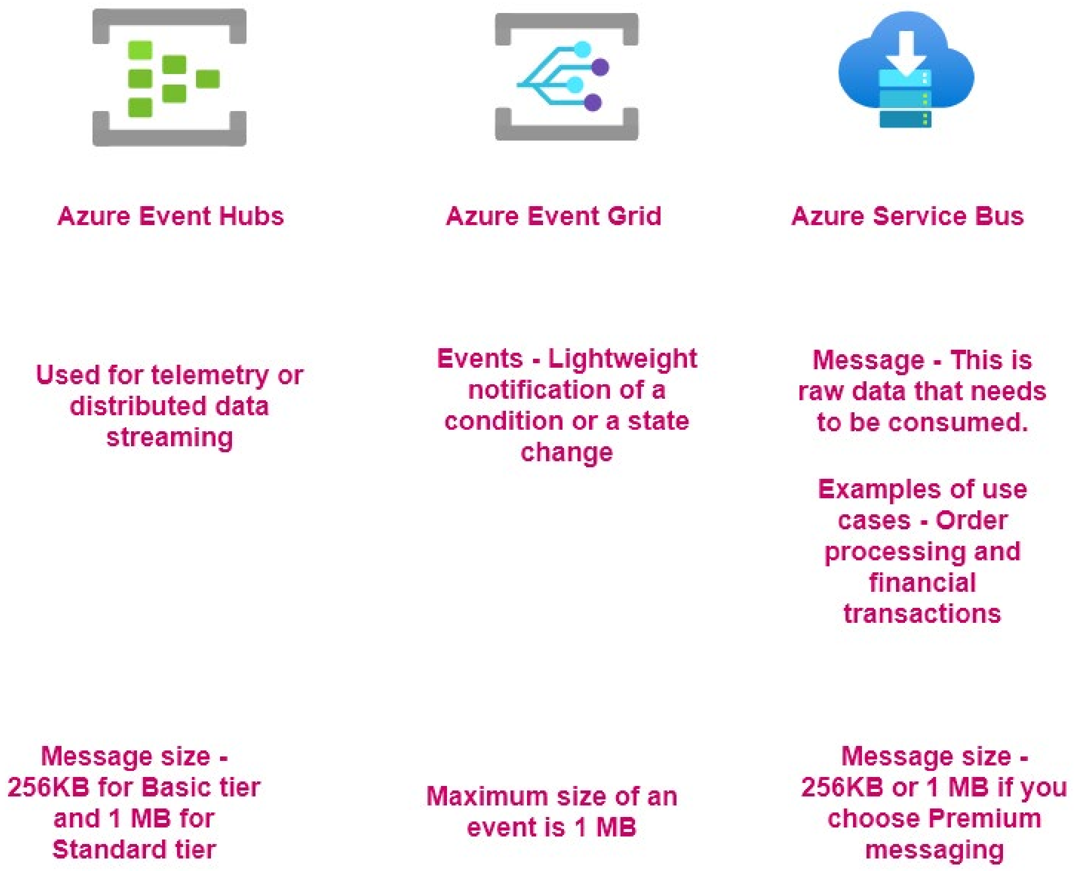
Application wants to publish custom events

Whenever a security event happens at the application level

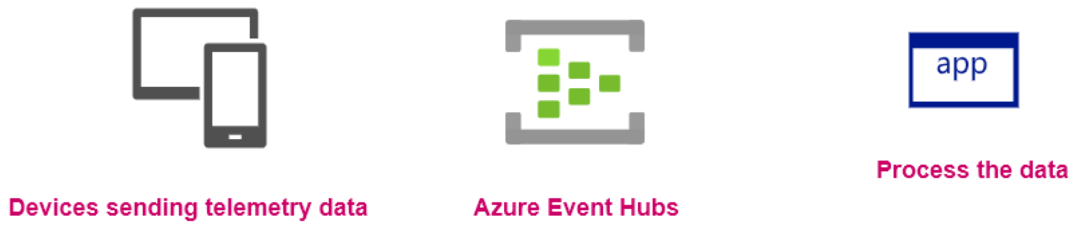
Or whenever an order is placed via the application

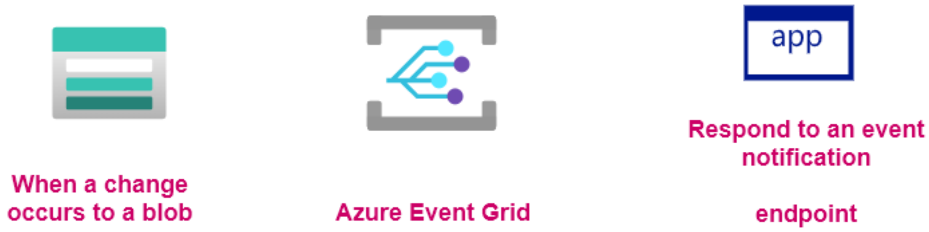
Custom Topic

Comparison between the different messaging services

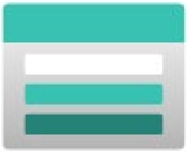


Azure Event Hubs	Azure Event Grid	Azure Service Bus
Used for telemetry or distributed data streaming	Events - Lightweight notification of a condition or a state change	Message - This is raw data that needs to be consumed. Examples of use cases - Order processing and financial transactions
Message size - 256KB for Basic tier and 1 MB for Standard tier	Maximum size of an event is 1 MB	Message size - 256KB or 1 MB if you choose Premium messaging




When a change occurs to a blob
Azure Event Grid
app
Respond to an event notification
endpoint

Some differences when it comes to Storage Blob triggers and Azure Event Grid



Azure storage accounts Blob service

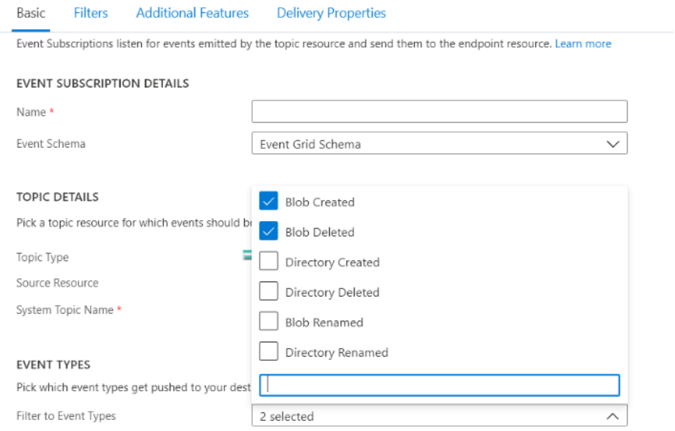
When a new or updated blob is detected

Blob storage trigger

Azure functions

Azure Event Grid

Azure Event Grid



Event Subscriptions listen for events emitted by the topic resource and send them to the endpoint resource. Learn more

EVENT SUBSCRIPTION DETAILS

Name *

Event Schema: Event Grid Schema

TOPIC DETAILS

Pick a topic resource for which events should b

Topic Type

Source Resource

System Topic Name *

☑ Blob Created
☑ Blob Deleted
□ Directory Created
□ Directory Deleted
□ Blob Renamed
□ Directory Renamed

EVENT TYPES

Pick which event types get pushed to your dest

Filter to Event Types: 2 selected

Azure Event Grid also has support for Blob-only storage accounts

Also if you need high-scale, consider using Event Grid over the Blob trigger - This is if you have more than 100,000 blobs in the storage account or if there are 100 blob updates per second.

Azure Logic Apps

Helps you to build workflows in Azure

Azure Logic Apps

Azure Functions

Developers define logic to perform some steps

When an administrative action is performed on a virtual machine

Email an administrator