

# INTRODUCTION À AZURE EVENT GRID

# QU'EST-CE QU'AZURE EVENT GRID

# GÉNÉRALITÉS

**Azure Event Grid** est un service de **routage** des événements à **haute disponibilité** intégré à la plateforme Azure.

## OBJECTIFS ET AVANTAGES

- **Routage simplifié et flexible** des événements
- **Démarrage rapide et évolutivité automatique**
- Conception **orientée événements** pour les applications modernes

# COMPOSANTS D'EVENT GRID

# TOPICS

Les **topics** sont des points de terminaison où les sources envoient des **événements** à traiter.

# SUBSCRIPTIONS

Les **subscriptions** sont créées pour connecter les **topics** aux **handlers d'événements**.

# HANDLERS DE GESTION D'ÉVÉNEMENTS

Les **handlers d'événements** sont des applications de **traitement d'événements**.

# EVENT GRID DOMAINS

Un **Event Grid Domain** regroupe plusieurs **topics** sous une même organisation.

# CAS D'UTILISATION

# MICROSERVICES

- Communication entre **microservices** avec des messages basés sur des **événements**

# AUTOMATISATION DES PROCESSUS

- **Automatisation** des tâches et **orchestrations** de workflows

# INTÉGRATION DE SYSTÈMES

- Intégration d'applications et de services, même en dehors d'Azure

# EXPLORER AZURE EVENT GRID

# INTRODUCTION À AZURE EVENT GRID

# QU'EST-CE QU'AZURE EVENT GRID

## GÉNÉRALITÉS

Azure Event Grid est un service de routage d'événements qui permet de **connecter facilement** des **sources d'événements** et des **gestionnaires d'événements**.

## OBJECTIFS ET AVANTAGES

- Permet de **simplifier** et d'**améliorer** la réactivité des applications
- **Efficacité, évolutivité** et **sécurité** pour gérer un grand nombre d'événements

# COMPOSANTS D'EVENT GRID

## TOPICS

Les **Topics** sont des points de terminaison permettant d'envoyer des **événements** à **Event Grid**.

## SUBSCRIPTIONS

Les **Subscriptions** lient les **Topics** aux **Handlers** pour déclencher des actions en réponse aux événements.

## HANDLERS DE GESTION D'ÉVÉNEMENTS

Les **Handlers** sont des services qui réagissent aux événements reçus d'**Event Grid**.

## EVENT GRID DOMAINS

Un **Domaine** d'Event Grid regroupe plusieurs **Topics** et **Subscriptions** pour faciliter leur gestion.

# CRÉATION D'UN TOPIC EVENT GRID

# CRÉER UN TOPIC

Pour créer un Topic, rendez-vous sur le portail **Azure** et spécifiez les éléments suivants:

- **Emplacement**
- **Sku**
- **Tags**

Élément	Description
Emplacement	Sélectionnez la région où vous souhaitez déployer le Topic.
Sku	Choisissez le plan tarifaire pour le Topic.
Tags	Attribuez des balises pour faciliter la gestion des ressources.

# RÉCUPÉRATION D'UN ENDPOINT

Après avoir créé votre **Topic**, copiez l'URL de l'**Endpoint** disponible dans les propriétés du Topic.

# GÉNÉRATION D'UNE CLÉ

Générez une clé d'accès pour votre **Topic** dans la section des clés d'accès pour pouvoir envoyer des événements à ce Topic.

Étape	Action
1. Accéder	Ouvrez l'interface Azure et accédez à votre <b>Topic</b>
2. Sélection	Sélectionnez la section des <b>clés d'accès</b>
3. Générer	Cliquez sur <b>Générer</b> pour créer une nouvelle clé d'accès

# CRÉATION D'UNE SUBSCRIPTION EVENT GRID

# CRÉER UNE SUBSCRIPTION

Pour créer une Subscription, vous devez déterminer les éléments suivants :

- **Filtrage d'événements,**
- **Endpoint,**
- **Retries et garantie de livraison.**

# FILTRAGE D'ÉVÉNEMENTS

Vous pouvez **filtrer** les événements en fonction de leur **type** ou de leurs **données**. Par exemple :

```
{  
  "properties": {  
    "filter": {  
      "includedEventTypes": ["RecordInserted"]  
    }  
  }  
}
```

# ENDPOINT

L'**Endpoint** est la destination où les **événements filtrés** seront envoyés. Il peut s'agir d'un **Webhook**, d'une **Azure Function**, d'une **Logic App**, etc.

# RETRIES ET GARANTIE DE LIVRAISON

**Event Grid** garantit que les événements seront livrés **au moins une fois**. En cas d'échec, il réessayera avec une stratégie d'**attente progressive**.

# REST API

# APPELS POUR CRÉER UNE SUBSCRIPTION

Pour créer une nouvelle **subscription**, utilisez un appel **POST** avec les informations suivantes :

- URL :

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

- Corps de la requête : **filtres**, point de **terminaison**, calendrier d'**essai**, etc.

# APPELS POUR AJOUTER / SUPPRIMER DES FILTRES

Pour ajouter ou supprimer des filtres, utilisez un appel **PUT** avec l'**URL** de la **subscription** et mettez à jour le corps de la requête avec les filtres souhaités.

# ÉVÉNEMENTS SYSTÈME ET ÉVÉNEMENTS PERSONNALISÉS

# ÉVÉNEMENTS SYSTÈME

Les **événements système** sont des événements générés par **Azure**, tels que la création ou la suppression d'une ressource.

## DESCRIPTION

Ils sont disponibles pour de nombreux services Azure, tels que **Blob Storage, Resource Manager**, etc.

# SOURCES SUPPORTÉES

La liste complète des sources supportées se trouve dans la **documentation officielle** d'Azure Event Grid.

Source	Description
Blob Storage	Stockage d'objets non structurés (images, vidéos)
Event Hubs	Plateforme de streaming d'événements en temps réel
IoT Hub	Gestion et connexion de périphériques IoT
Service Bus	Plateforme de messagerie d'entreprise

# ÉVÉNEMENTS PERSONNALISÉS

Vous pouvez créer et envoyer des **événements personnalisés** spécifiques à votre application.

## CRÉATION D'UN ÉVÉNEMENT PERSONNALISÉ

- Utilisez `EventEmitter` pour créer un émetteur d'événements
- Définir un événement avec un nom unique

## EXEMPLE D'ÉMISSION D'UN ÉVÉNEMENT PERSONNALISÉ

```
const EventEmitter = require('events');
const monEmetteur = new EventEmitter();

monEmetteur.on('monEvenement', () => {
  console.log('Événement personnalisé déclenché');
});

monEmetteur.emit('monEvenement');
```

# CRÉATION ET ENVOI D'UN ÉVÉNEMENT PERSONNALISÉ

Utilisez la fonction **sendEvent** et spécifiez les **propriétés souhaitées** pour votre événement.

```
const { TelemetryClient, TelemetryProcessor, sendEvent } = require('applicationinsights');
const client = new TelemetryClient();

const properties = {
  'prop1': 'valeur1',
  'prop2': 'valeur2'
};

client.trackEvent({ name: 'MonEvenement', properties });
```

# PROPRIÉTÉS SUPPORTÉES

Les propriétés suivantes sont disponibles pour les **événements personnalisés** :

- eventType
- eventTime
- data (contenu principal de l'événement)
- id (optionnel)

# GESTION DES HANDLERS D'ÉVÉNEMENTS

# TYPES DE HANDLERS

Event Grid prend en charge plusieurs types de **Handlers** pour gérer différents scénarios :

- **Azure Functions**
- **Logic Apps**
- **Webhooks**
- Intégration avec d'autres services **Azure**

# AZURE FUNCTIONS

**Azure Functions** est un service de **calcul sans serveur** qui peut exécuter du code en réponse à des événements.

# EXEMPLE AVEC AZURE FUNCTIONS

1. Créer une nouvelle **fonction Azure**
2. Sélectionner le **déclencheur "Event Grid"**
3. Configurer les **paramètres du déclencheur**
4. Implémenter la **logique de gestion d'événements**

# LOGIC APPS

**Logic Apps** est un service d'intégration basé sur des **workflows** qui utilise des **connecteurs** pour communiquer avec différents services.

Services	But
Entreprise	Intégration de systèmes internes
SaaS (Software as a Service)	Automatisation des tâches et processus métier
Connecteurs natifs	Réduction du délai de configuration des services
Déclencheurs et actions	Interaction avec les services à travers les connecteurs

## EXAMPLE AVEC LOGIC APPS

1. Créer une nouvelle **Logic App**
2. Ajouter un déclencheur "**Event Grid**" à la **Logic App**
3. Configurer les paramètres du déclencheur
4. Implémenter la logique de gestion d'événements à l'aide des actions des **connecteurs**

# WEBHOOKS

Les **Webhooks** sont des appels **HTTP simples** qui envoient des données à une **URL spécifique** en réponse à un **événement**.

# CONFIGURATION DES WEBHOOKS

1. Créer un point de **terminaison HTTP** pour recevoir les événements
2. Configurer la **sécurité**, comme la validation des jetons
3. Implémenter la **logique de gestion d'événements**

# INTÉGRATION AVEC D'AUTRES SERVICES AZURE

Event Grid peut également être intégré à d'autres services Azure, tels que **Storage**, **Databases** et **Messaging**.

Services	Description
Storage	Stockage de données réparties
Databases	Gestion de bases de données
Messaging	Communication entre les services Azure

## EXEMPLES D'INTÉGRATION

- Stocker des **événements** dans **Blob Storage**
- Insérer des **événements** dans **Cosmos DB**
- Envoyer des **événements** à des **files d'attente**, comme **Service Bus**

# MONITORING ET DIAGNOSTICS

# MONITORING DES ÉVÉNEMENTS

# AZURE MONITOR

Azure Monitor est un service qui **collecte**, **analyse** et **agit** sur les **données télémétriques** provenant de diverses sources. Il aide à surveiller les événements dans Event Grid.

# OUTIL D'ANALYSE DES ÉVÉNEMENTS

Utilisez des outils d'analyse comme **Log Analytics** pour consulter les données des événements et trouver des **tendances** et des **anomalies**.

# DIAGNOSTICS ET TÉLÉMÉTRIE

# APPLICATION INSIGHTS

**Application Insights** est un service **Azure** qui offre des fonctionnalités de **diagnostic** et de **télémétrie** pour les applications. Il est intégré avec **Event Grid** pour surveiller les performances et analyser les problèmes.

# CONFIGURATION ET UTILISATION

Pour configurer et utiliser **Application Insights** avec **Event Grid** :

1. Créez une ressource **Application Insights** dans le portail Azure.
2. Ajoutez l'instrumentation à votre application.
3. Activez les **diagnostics** pour Event Grid.
4. Visualisez les données dans le portail Azure.

# BONNES PRATIQUES ET ASTUCES

# SÉCURITÉ ET AUTHENTIFICATION

# WEBHOOKS SÉCURISÉS

Utilisez des **webhooks sécurisés** pour protéger les communications entre **Event Grid** et les **points de terminaison de l'événement**.

# CERTIFICATS CLIENT

Utilisez des **certificats client** pour sécuriser l'**authentification** et l'**autorisation** entre les services **Event Grid** et les clients.

# SAS (SHARED ACCESS SIGNATURES)

Les **SAS** sont des chaînes de caractères générées par le service **Azure** qui permettent de **contrôler l'accès** aux ressources. Utilisez-les pour sécuriser les échanges d'événements avec **Event Grid**.

# OPTIMISATION ET PERFORMANCE

# GESTION DES ERREURS

Gérez correctement les erreurs en traitant les **exceptions**, en planifiant des **tentatives ultérieures** et en configurant des **alertes** pour les problèmes systémiques.

# THROTTLING ET DÉLESTAGE

Mettez en œuvre des mécanismes de **limitation** (throttling) et de **délestage** pour éviter que le service **Event Grid** ne soit surchargé et pour garantir une utilisation optimale des ressources.

# BONNES PRATIQUES ET ASTUCES

# SÉCURITÉ ET AUTHENTIFICATION

Le respect de la **sécurité** et de l'**authentification** est essentiel lors de l'utilisation d'**Azure Event Grid**. Voici quelques points importants à prendre en compte dans ce contexte.

- Principe de moindre privilège
- Utilisation de clés d'authentification
- Gestion des rôles et des permissions
- Chiffrage des données

# WEBHOOKS SÉCURISÉS

Assurez-vous d'utiliser des **webhooks sécurisés** avec **HTTPS** pour garantir la confidentialité des communications.

# CERTIFICATS CLIENT

Utilisez des **certificats client** pour établir une connexion **sécurisée** entre les **Handlers d'événements** et le **Topic Event Grid**.

# SAS

Utilisez des **signatures d'accès partagé** (SAS) pour vérifier l'accès aux ressources **Azure Event Grid**.

# OPTIMISATION ET PERFORMANCE

Voici quelques conseils pour **optimiser** les **performances** lors de l'utilisation d'**Azure Event Grid**.

- Choisissez le bon **type d'abonnement** en fonction de vos besoins (par exemple, les abonnements basés sur les filtres pour filtrer les événements)
- Utilisez des **formats de messages** compatibles avec votre système
- Pensez à mettre en cache les données fréquemment utilisées pour réduire la latence
- Surveillez et ajustez la **taille des lots d'événements** pour éviter les retards
- Utilisez des **stratégies de gestion des erreurs** pour traiter les éventuelles erreurs, comme les tentatives, les délais d'attente, etc.

# GESTION DES ERREURS

Gérez les erreurs de manière appropriée pour vous assurer que les événements sont traités correctement, même en cas d'échec.

Technique	Description
Essayer/Capturer	Encapsulez le code dans un bloc try-catch pour attraper les erreurs.
Journalisation	Logging des erreurs pour un examen ultérieur et un débogage.
Notifications	Notifier les administrateurs et les développeurs en cas d'erreurs.

- Utiliser des **méthodes spécifiques** pour gérer les erreurs en fonction du contexte.
- S'assurer de **comprendre les erreurs** potentielles pour pouvoir les gérer correctement.

# THROTTLING ET DÉLESTAGE

Anticipez et gérez le **throttling** afin de vous assurer que votre application puisse traiter correctement un grand nombre d'événements en période de charge élevée.

Gérez également le **délestage** pour réduire la charge sur votre système, en limitant le nombre d'événements traités à la fois ou en les répartissant entre plusieurs **Handlers** d'événements.

Throttling	Délestage
Gérer la vitesse	Réduire la charge
Anticiper	Répartir

