

# INTRODUCTION AUX FONCTIONS AZURE

# QU'EST-CE QU'UNE FONCTION AZURE ?

# DÉFINITION

Une **fondction Azure** est une unité de code **sans état**, indépendante et **évolutive** qui peut être exécutée dans le cloud.

# AVANTAGES D'UTILISER LES FONCTIONS AZURE

- Scalabilité automatique
- Paiement à l'utilisation
- Intégration native avec d'autres **services Azure**
- Rapidité de **développement** et de **déploiement**

# CRÉATION D'UNE FONCTION AZURE

# IMPLÉMENTER LES FONCTIONS AZURE

# INTRODUCTION AUX FONCTIONS AZURE

# QU'EST-CE QU'UNE FONCTION AZURE ?

## DÉFINITION

**Azure Functions** est un service de **calcul sans serveur** qui permet d'exécuter du code à la demande sans avoir à gérer explicitement les infrastructures.

## AVANTAGES D'UTILISER LES FONCTIONS AZURE

- **Évolutivité automatique**
- Facturation à l'exécution
- Développement rapide avec des **modèles préconfigurés**

## SCÉNARIOS D'UTILISATION COURANTS

- **Microservices**
- **Intégrations d'applications**
- **Tâches planifiées**

# MODÈLES DE TARIFICATION

## CONSOMMATION

- Facturation basée sur le **temps d'exécution** et les **ressources consommées**
- Évolutivité automatique en fonction de la **demande**

## PLAN PREMIUM

- Toutes les fonctionnalités du plan de consommation +
- **Instances dédiées** et réservées
- Possibilité d'**exécution en continu**

## PLAN DÉDIÉ

- Toutes les fonctionnalités du plan **Premium +**
- Contrôle total de l'**infrastructure**

# DÉCLENCHEURS (TRIGGERS)

# HTTP TRIGGER

## SYNTAXE

```
module.exports = async function (context, req) {
    context.res = {
        status: 200,
        body: "Hello, World!"
    };
};
```

## TYPES DE REQUÊTES HTTP

- GET
- POST
- PUT
- DELETE
- PATCH

## EXEMPLES D'UTILISATION

- API REST
- Webhooks

# TIMER TRIGGER

## SYNTAXE

```
module.exports = async function (context) {
    context.log('Execution every 5 minutes');
};
```

Dans le fichier function.json:

```
{
    "schedule": "0 */5 * * *"
}
```

## PLANIFICATION

- Utilisation de la **syntaxe des chronogrammes CRON**
- Exemple: "0 \* /5 \* \* \*" pour exécuter la fonction toutes les **5 minutes**

## EXEMPLES D'UTILISATION

- Tâches de maintenance planifiées
- Nettoyage de données

# AUTRES DÉCLENCHEURS

## QUEUE TRIGGER

- Exécuter une fonction lorsque des messages sont ajoutés à une file d'attente **Azure Storage**

## BLOB TRIGGER

- Exécuter une fonction lorsqu'un fichier est **ajouté** ou **modifié** dans un stockage **Blob** d'Azure

## EVENT HUB TRIGGER

- **Exécuter** une fonction lorsqu'un **événement** est envoyé à un **Event Hub Azure**

# LIAISONS (BINDINGS)

# INTRODUCTION AUX LIAISONS

Les **liaisons** (Bindings) sont des mécanismes qui facilitent la communication entre la fonction **Azure** et les services externes ou les données.

- Entrée : lecture de données à partir de sources externes
- Sortie : envoi de données à des destinations externes
- Trigger : déclenchement de la fonction

## Type de liaison   Exemples d'utilisation

---

Entrée	Lecture de fichiers, requêtes HTTP
Sortie	Envoi d'email, écriture dans une base de données
Trigger	Réception d'un message sur un topic Azure Service Bus, réception d'une notification

# LIAISONS D'ENTRÉE

Les **liaisons d'entrée** récupèrent les données depuis un **service externe** et les transforment en objet à utiliser dans la **Fonction Azure**.

# SYNTAXE

```
{  
  "bindings": [  
    {  
      "type": "<binding_type>",  
      "direction": "in",  
      "name": "<binding_name>",  
      "parameter": "<input_parameter>"  
    }  
  ]  
}
```

# GESTION DES ERREURS ET MONITORING

# JOURNAUX D'APPLICATION (APPLICATION INSIGHTS)

Les **journals d'application** permettent de suivre et analyser les **performances** des **Fonctions Azure**.

# CONFIGURATION

Pour configurer **Application Insights**, ajoutez la **chaîne de connexion** dans les **paramètres de l'application**.

# SÉCURITÉ ET AUTHENTIFICATION

