

APPLICATIONS SANS SECRETS

LA GESTION DES SECRETS
APPLICATIFS NE DOIT PAS
NÉCESSAIREMENT ÊTRE COMPLEXE



Salut, mon nom est ...

Tidjani Belmansour



Architecte Cloud chez Cofomo (<https://www.cofomo.com>)



Microsoft Azure MVP



Co-organisateur de la Communauté Azure de Québec
(<https://meetup.com/azureqc>)



@Tidjani_B



<https://espacenuagic.com> | <https://dev.to/tidjani>



“Offrir une solution élégante et sécuritaire pour la gestion des secrets applicatifs indépendamment de son emplacement.”

Le problème

La gestion des secrets est difficile

On se retrouve souvent avec des informations sensibles dans les fichiers de configuration d'application

(e.g., appSettings.json)

```
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*",
  "AzureAd": {
    "Instance": "https://login.microsoftonline.com/",
    "Domain": "espacenuagic.com",
    "TenantId": "734baa05-7a37-470e-a15a-c6d506e09e4c",
    "ClientId": "f8f6a6ef-1f61-4360-a680-46bbcf5a5ce01",
    "CallbackPath": "/signin-oidc"
  },
  "ConnectionStrings": {
    "DefaultConnection": "Server=tcp:vstoolboxsqlsrv.data
  },
  "ServiceBusConnStr": "Endpoint=sb://vstoolbox.servicebu
  "ServiceTokenProviderConnectionString": "RunAs=Develope
  // "ServiceTokenProviderConnectionString": "RunAs=App;
}
```

Mais ça ne doit pas nécessairement l'être !

- ❖ Exclure le fichier de configuration du Source Control
- ❖ Utiliser le fichier secrets.json
- ❖ Utiliser des variables d'environnement
- ❖ Utiliser Key Vault Reference
- ❖ Utiliser Azure Service Authentication

Option #1:
exclure le fichier de configuration du Source Control

Exclure le fichier de configuration du Source Control

1. Ajouter chemin du fichier de config dans “.gitignore”
2. Retirer le fichier de config du repo Git après commit:
`git rm --cached <path-to-config-file>`

Exclure le fichier de configuration du Source Control

```
## Get latest from https://github.com/github/gitignore/blob/master/VisualStudio.gitignore

# User-specific files
*.rsuser
*.suo
*.user
*.userosscache
*.sln.docstates

# User-specific files (MonoDevelop/Xamarin Studio)
*.userprefs

# Mono auto generated files
mono_crash.*

# Exclude config file since it contains sensitive information
vstoolbox.manageSecrets/vstoolbox.manageSecrets.Web/appsettings.json
```

Exclure le fichier de configuration du Source Control

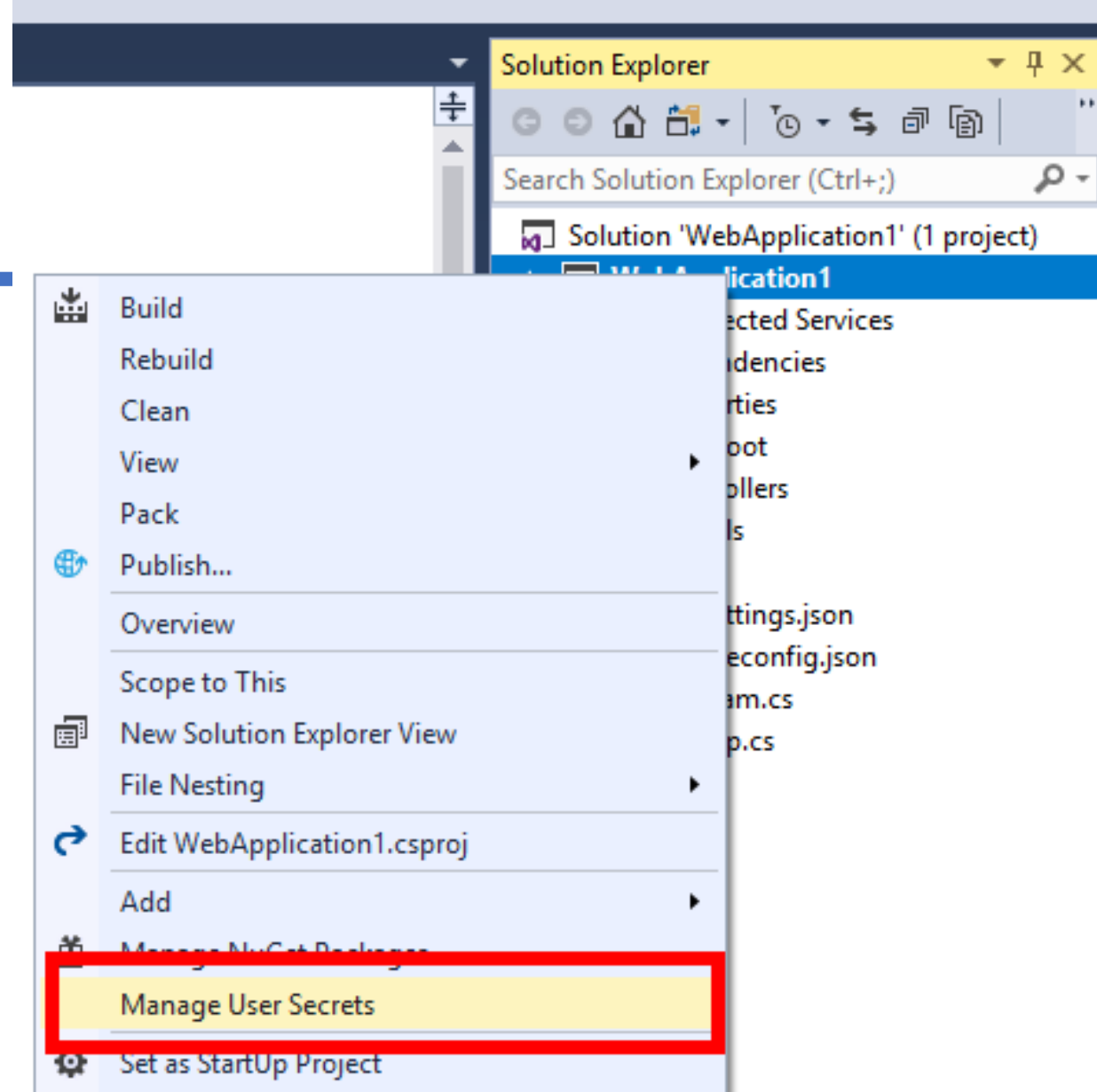
Inconvénients:

- ❖ Pas naturel d'exclure le fichier de config du contrôle de source
 - N'est pas conçu pour stocker des informations sensibles
- ❖ Si exclu, la tentation sera grande de le remettre

Option #2:
utiliser le fichier secrets.json

Utiliser le fichier secrets.json

- ❖ Fichier de config uniquement disponible sur le poste de dev
- ❖ Non archive dans le contrôle de source
- ❖ Non affiché dans VS Solution Explorer



Utiliser le fichier secrets.json

The screenshot displays the Visual Studio IDE. The main editor window shows the `secrets.json` file, which is highlighted with a red rectangle. The file path in the address bar is `C:\Users\tidja\AppData\Roaming\Microsoft\UserSecrets\d03990e9-c87c-4487-9b0e-c94a178efec1\secrets.json`. The JSON content is as follows:

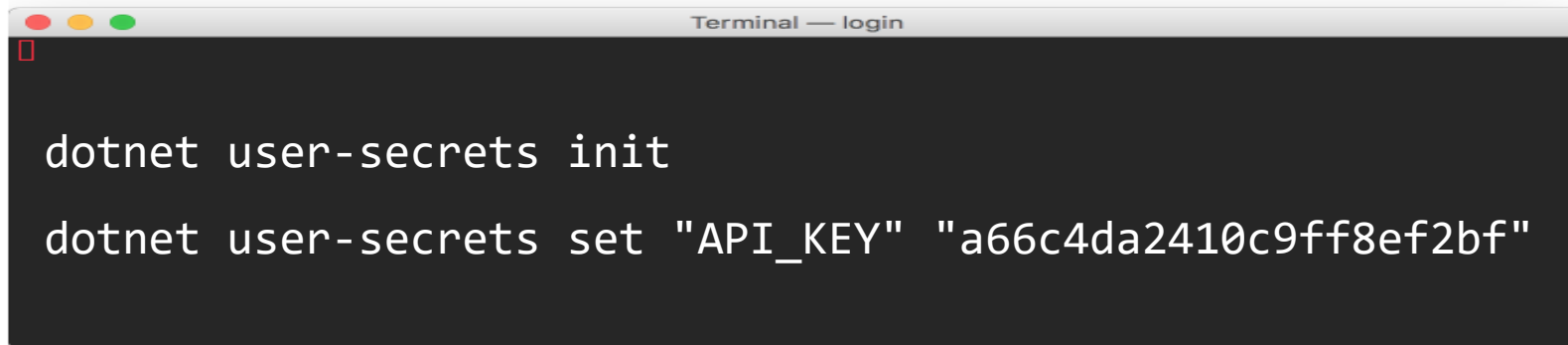
```
1  {
2    "AzureAd": {
3      "Instance": "https://login.microsoftonline.com/",
4      "Domain": "espacenuagic.com",
5      "TenantId": "734baa05-7a37-470e-a15a-c6d506e09e4c",
6      "ClientId": "f8f6a6ef-1f61-4360-a680-46bbcfa5ce01",
7      "CallbackPath": "/signin-oidc"
8    },
9
10   "ConnectionStrings": {
11     "DefaultConnection": "Server=tcp:vstoolboxsqlsrv.database.windows.net;
12
13
14   "ServiceBusConnStr": "Endpoint=sb://vstoolbox.servicebus.windows.net;
15
16 }
```

The Solution Explorer on the right side of the IDE is also highlighted with a red rectangle. It shows the project structure for `vstoolbox.manageSecrets.Web`. The files and folders listed are:

- Connected Services
- Dependencies
- Properties
- wwwroot
- Controllers
- Models
- Views
- `appsettings.json` (highlighted)
- `Program.cs`
- `Startup.cs`

Utiliser le fichier secrets.json

Si vous utilisez Visual Studio Code:

A terminal window with a dark background and a light gray title bar. The title bar contains the text "Terminal — login" and three colored window control buttons (red, yellow, green) on the left. The terminal shows two lines of white text: "dotnet user-secrets init" and "dotnet user-secrets set \"API_KEY\" \"a66c4da2410c9ff8ef2bf\"".

```
Terminal — login  
dotnet user-secrets init  
dotnet user-secrets set "API_KEY" "a66c4da2410c9ff8ef2bf"
```

Utiliser le fichier secrets.json

Inconvénients:

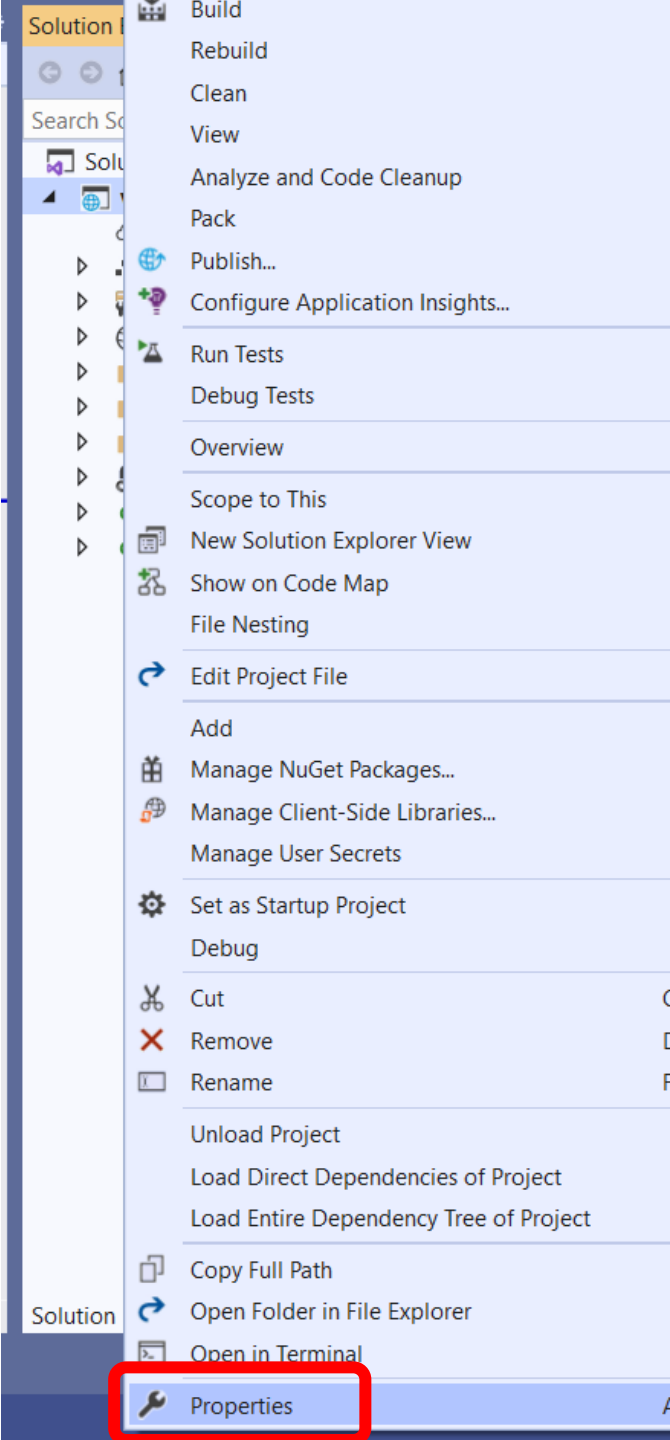
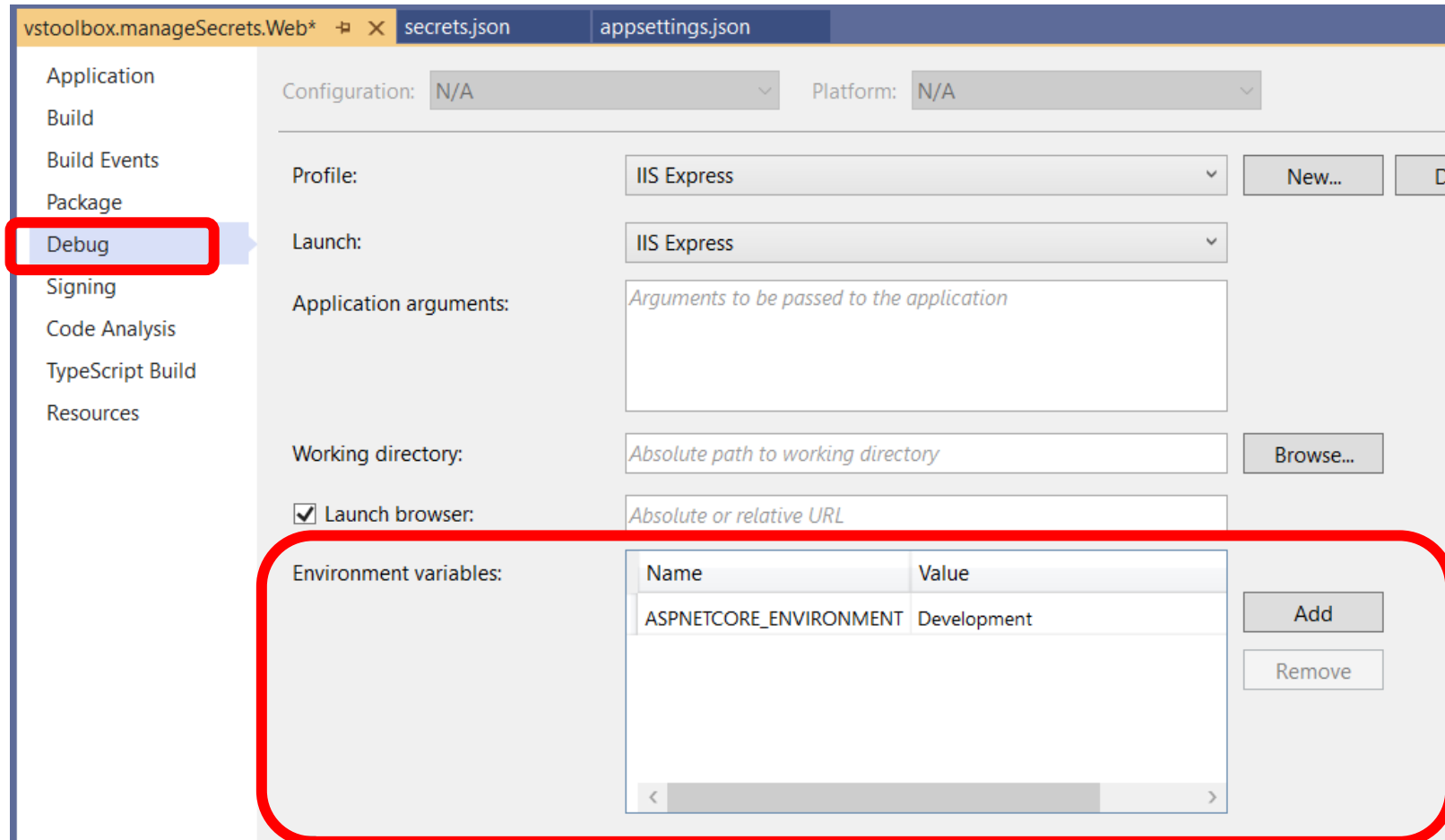
- ❖ Étant donné que le fichier secrets.json reste sur le poste du développeur, le fait de changer de poste de développement oblige le développeur à recréer ce fichier et à y remettre les informations secrètes
- ❖ Si ces informations venaient à changer, il faudrait en informer chaque développeur et il serait de la responsabilité de ce dernier de mettre à jour ces informations dans son fichier secrets.json

Option #3:
utiliser les variables d'environnement

Utiliser les variables d'environnement

- ❖ Découpler la gestion de la configuration du code applicatif
- ❖ Multiplateforme (Windows, Linux, macOS)
- ❖ Information non stockée dans le contrôle de source

Utiliser les variables d'environnement



```
{
  "iisSettings": {
    "windowsAuthentication": false,
    "anonymousAuthentication": true,
    "iisExpress": {
      "applicationUrl": "http://localhost:5010",
      "sslPort": 44317
    }
  },
  "profiles": {
    "IIS Express": {
      "commandName": "IISExpress",
      "launchBrowser": true,
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    },
    "vstoolbox.manageSecrets.Web": {
      "commandName": "Project",
      "dotnetRunMessages": "true",
      "launchBrowser": true,
      "applicationUrl": "https://localhost:5001;http://localhost:5000",
      "environmentVariables": {
        "ASPNETCORE_ENVIRONMENT": "Development"
      }
    }
  }
}
```

Search Solution Explorer (Ctrl+;)

Solution 'vstoolbox.manageSecrets' (1 of 1 project)

- ▼ vstoolbox.manageSecrets.Web
 - Connected Services
 - Dependencies
 - Properties
 - launchSettings.json
 - wwwroot
 - Controllers
 - Models
 - Views
 - appsettings.json
 - Program.cs
 - Startup.cs

Terminal — login

```
dotnet run --launch-profile "vstoolbox.manageSecrets.Web"
```

System

< > > Control Panel > System and Security > System

Search Control Panel

Control Panel Home

Device Manager

Remote settings

System protection

Advanced system settings

View basic information about your computer

Windows edition

Windows 10 Pro

© 2016 Microsoft C

System

Processor:

Installed memory (R

System type:

Pen and Touch:

Computer name, domain

Computer name:

Full computer name

Computer description

Workgroup:

Windows activation

Windows is activate

Product ID: 00331-2

System Properties

Computer Name Hardware Advanced System Protection Remote

You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

Settings...

User Profiles

Desktop settings related to your sign-in

Settings...

Startup and Recovery

System startup, system failure, and debugging information

Settings...

Environment Variables...

OK

Cancel

Apply

Windows 10

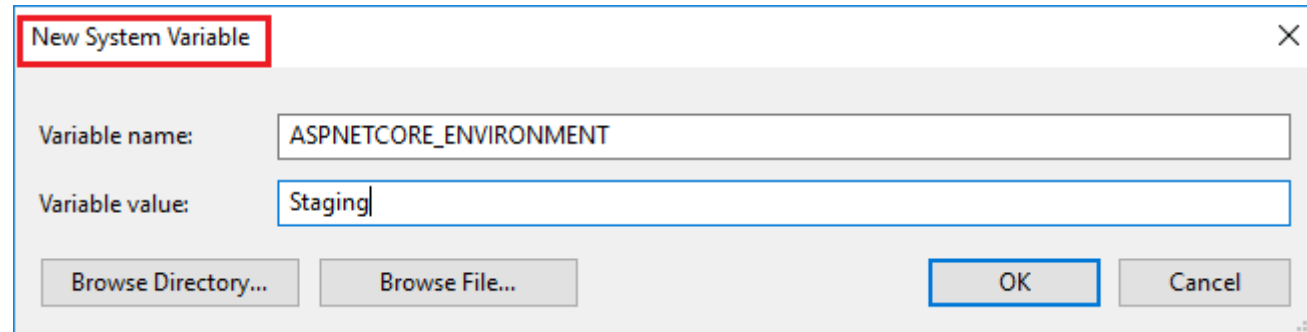
Change settings

Change product key

See also

Security and Maintenance

Utiliser les variables d'environnement



Utiliser les variables d'environnement

Console



```
set BLOG_URL=https://espacenuagic.com  
setx BLOG_URL=https://espacenuagic.com
```

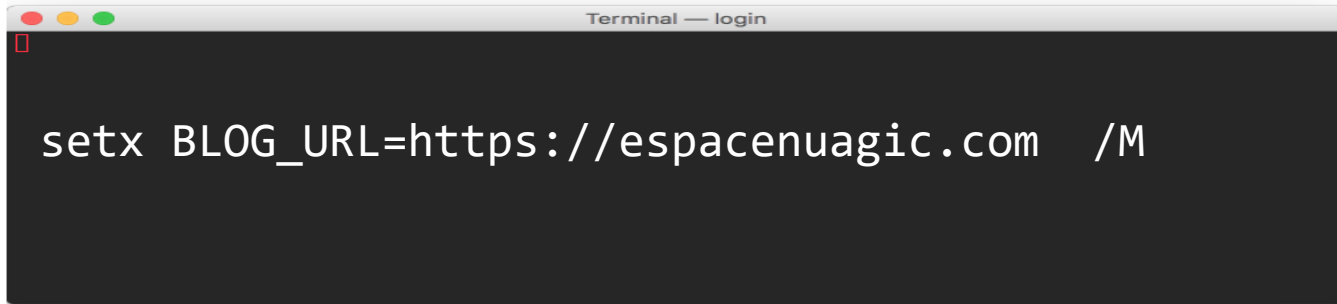
PowerShell



```
$Env:BLOG_URL = "https://espacenuagic.com"  
setx BLOG_URL "https://espacenuagic.com"
```

Utiliser les variables d'environnement (Windows)

Console



```
setx BLOG_URL=https://espacenuagic.com /M
```

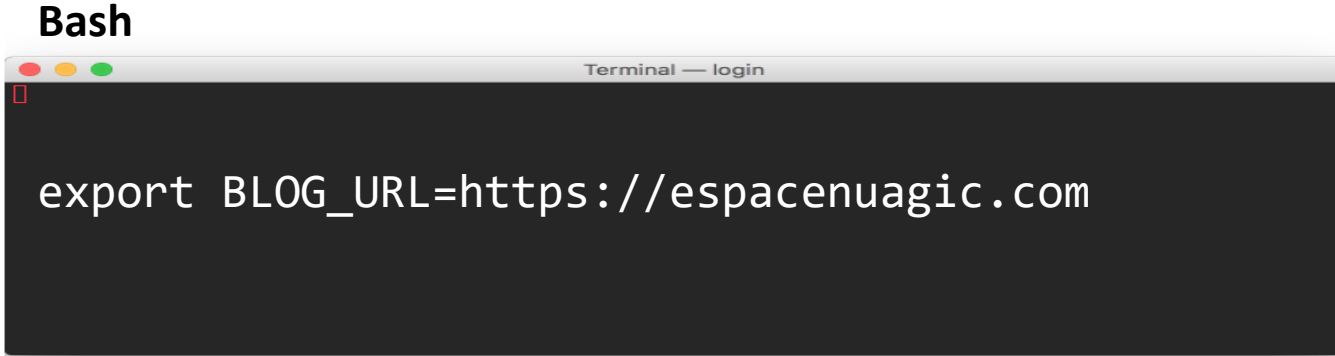
PowerShell



```
[Environment]::SetEnvironmentVariable("BLOG_URL", "https://espacenuagic.com", "Machine")
```

Utiliser les variables d'environnement (Linux & macOS)

Bash

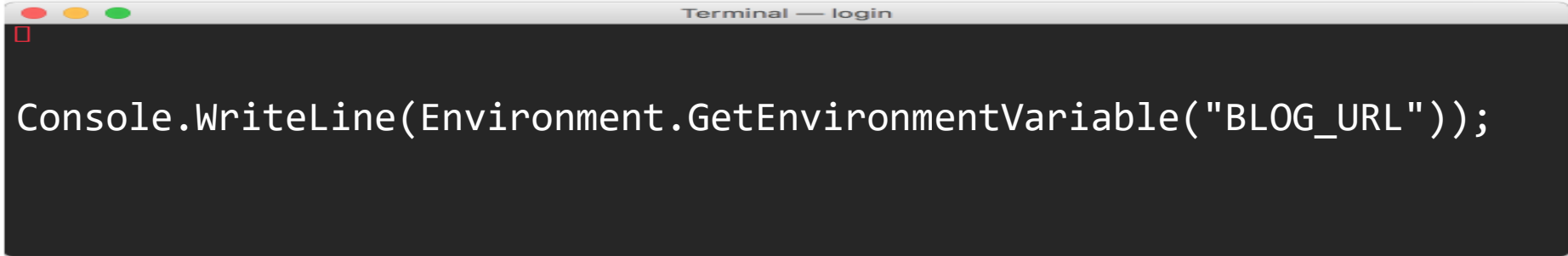
A screenshot of a macOS Terminal window. The title bar at the top reads "Terminal — login". The terminal has a dark background with white text. The command "export BLOG_URL=https://espacenuagic.com" is entered at the prompt. The window has standard macOS window controls (red, yellow, green buttons) in the top-left corner.

```
export BLOG_URL=https://espacenuagic.com
```

Pour les variables d'environnement à l'échelle de la machine, définissez-les dans le fichier *bash_profile*.

Utiliser les variables d'environnement

C#

A terminal window with a dark background and a light gray title bar. The title bar contains the text "Terminal — login" and three colored window control buttons (red, yellow, green) on the left. A small red square icon is visible in the top-left corner of the terminal area. The code "Console.WriteLine(Environment.GetEnvironmentVariable("BLOG_URL"));" is displayed in white text.

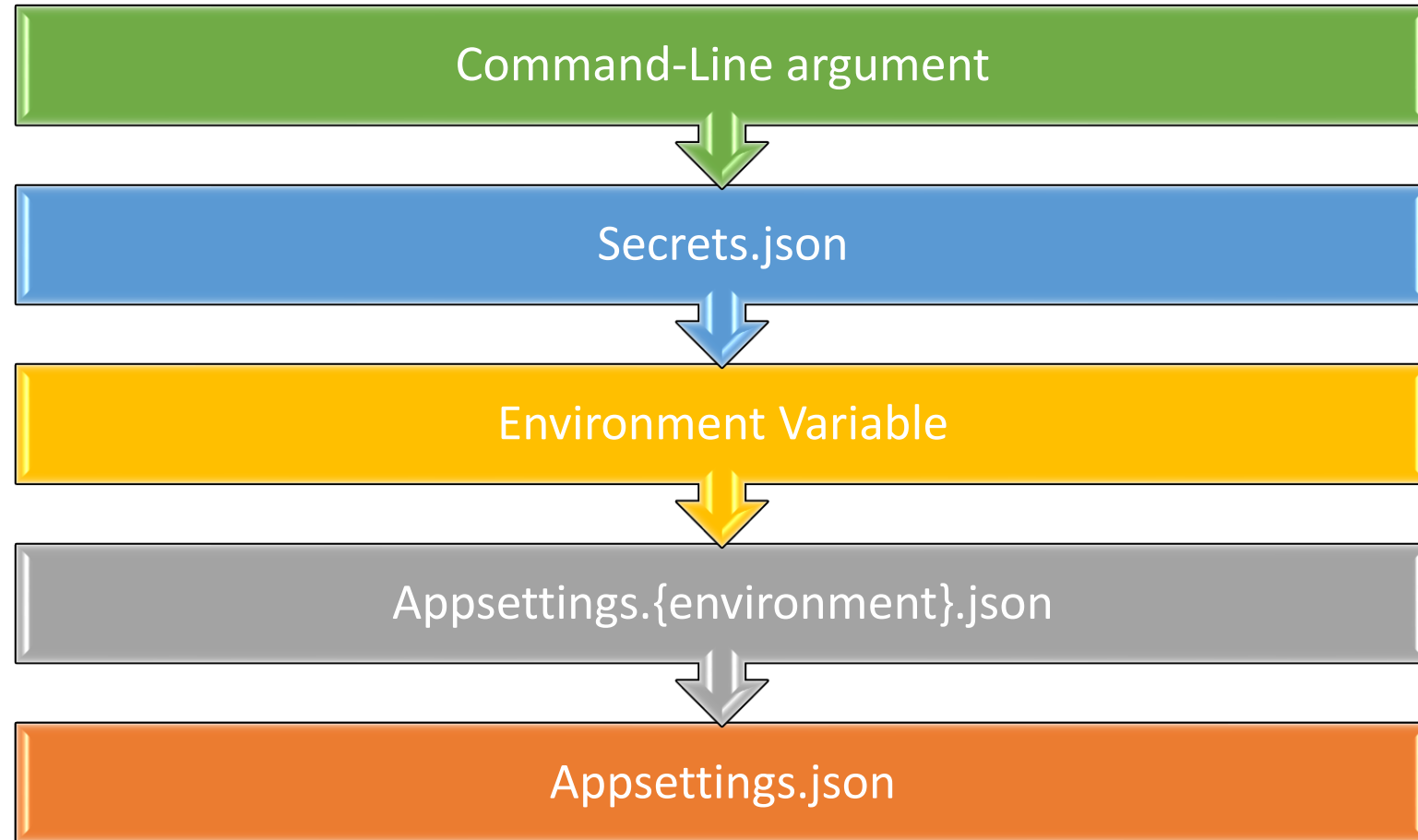
```
Terminal — login  
Console.WriteLine(Environment.GetEnvironmentVariable("BLOG_URL"));
```

Utiliser les variables d'environnement


Inconvénients:

- ❖ Configurées (et maintenues) par serveur
- ❖ Information non encryptée
- ❖ Redémarrage requis si la valeur est mise à jour








Ordre de préséance







Option #4:
utiliser Key Vault Reference


 Search (Ctrl+/)

«  Refresh  Save  Discard  Leave Feedback

-  Overview
-  Activity log
-  Access control (IAM)
-  Tags
-  Diagnose and solve problems
-  Security
-  Events (preview)


- Deployment**
-  Quickstart
 -  Deployment slots
 -  Deployment Center

- Settings**
-  Configuration
 -  Authentication



 Click here to upgrade to a higher SKU and enable additional features.

Application settings

Application settings are encrypted at rest and transmitted over an encrypted channel. You can choose to display them in plain text in your browser by using the controls below. App Settings are exposed as environment variables for access by your application at runtime. [Learn more](#)



 New application setting  Show values  Advanced edit


 Filter application settings

Name	Value	Source
superSecretMessage	 @Microsoft.KeyVault(SecretUri=https://demokeyvaultreference-kv.vault.azure.net:443/secrets/superSecretMessage/ac339852)	 Key vault Reference
WEBSITE_NODE_DEFAULT_VERSION	 Hidden value. Click to show value	App Service Config

Connection strings

Connection strings are encrypted at rest and transmitted over an encrypted channel.

 New connection string  Show values  Advanced edit

 Filter connection strings

Utiliser Key Vault Reference

@Microsoft.KeyVault(VaultName=demokeyvaultreference-kv;SecretName=superSecretMessage)

Utiliser Key Vault

Si ce n'est pas déjà fait, il faut activer le Managed Identity pour la Web App:

vstoolboxmanageSecretsWeb | Identity App Service

Search (Ctrl+/)

- Overview
- Activity log
- Access control (IAM)**
- Tags
- Diagnose and solve problems
- Security
- Events (preview)

Deployment

- Quickstart
- Deployment slots
- Deployment Center

Settings

- Configuration
- Authentication
- Application Insights
- Identity**

System assigned User assigned

A system assigned managed identity is restricted to one per resource and Azure role-based access control (Azure RBAC). The managed identity is an [Azure Managed identity](#).

Save Discard Refresh Got feedback?

Status ⓘ

Off **On**

Object (principal) ID ⓘ

e66b968c-3827-4e83-9a3e-5e015a47fdf5


Permissions ⓘ







Azure role assignments

Information ⓘ This resource is registered with Azure Active Directory. The managed identity settings for the managed identity because it can result in failures. [Learn more](#)







Utiliser Key Vault Reference

 demokeyvaultreference-kv | Access policies ...
Key vault

 Search (Ctrl+/)

-  Overview
-  Activity log
-  Access control (IAM)
-  Tags
-  Diagnose and solve problems
-  Events

Settings

-  Keys
-  Secrets
-  Certificates
-  Access policies
-  Networking
-  Security

 Save  Discard  Refresh

Enable Access to:





- ☐ Azure Virtual Machines for deployment ⓘ
- ☐ Azure Resource Manager for template deployment ⓘ
- ☐ Azure Disk Encryption for volume encryption ⓘ

Permission model


- ☒ Vault access policy
- ☐ Azure role-based access control

[+ Add Access Policy](#)

Current Access Policies

Name	Email	Key Permissions	Secret Permissions	Certificate Permissions	Action
APPLICATION					
 vstoolboxmanageSec...		0 selected 	2 selected 	0 selected 	Delete

Utiliser Key Vault Reference

 <https://vstoolboxmanagesecretsweb.azurewebsites.net>

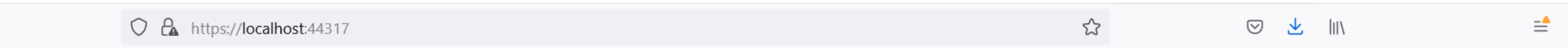


vstoolbox.manageSecrets.Web [Home](#) [Privacy](#)

dontwantyou toknow

Learn about [building Web apps with ASP.NET Core](#).

Utiliser Key Vault Reference



vstoolbox.manageSecrets.Web Home Privacy

```
@Microsoft.KeyVault(SecretUri=https://demokeyvaultrefe  
kv.vault.azure.net:443/secrets  
/superSecretMessage  
/ac339852329f4314bbe157d31b708fdd)
```

Learn about [building Web apps with ASP.NET Core](#).

Utiliser Key Vault Reference

Inconvénients:

- ❖ Solution “Azure-Only”
- ❖ Pas (encore) supportée par tous les services Azure

Option #5:
utilise Azure Service Authentication

Utiliser Azure Service Authentication

Permet:

- ❖ D'unifier la façon d'accéder aux ressources Azure, que ce soit via le poste de développement ou une fois l'application déployée sur Azure
- ❖ D'éviter de sauvegarder des informations d'infrastructure dans les fichiers de configuration du code (ex. tenantId, subscriptionId, userIdentity, ...)

Utiliser Azure Service Authentication

Ce faisant:

- ❖ On sépare encore mieux l'infrastructure du code
 - Plus besoin de mettre à jour les fichiers de configuration en cas de changement des informations d'infrastructure (tenantId, subscriptionId, userIdentity, ...)
- ❖ On renforce la sécurité
 - Ces informations n'étant pas sauvegardées dans le contrôle de code source, elles ne peuvent pas fuiter et être exploitées lors d'une attaque

Utiliser Azure Service Authentication

La solution est basée sur l'utilisation de la classe **DefaultAzureCredential** de la librairie **Azure.Identity** et qui permet de tenter une authentification dans l'ordre suivant:

- [EnvironmentCredential](#)
- [ManagedIdentityCredential](#)
- [SharedTokenCacheCredential](#)
- [VisualStudioCredential](#)
- [VisualStudioCodeCredential](#)
- [AzureCliCredential](#)
- [AzurePowerShellCredential](#)
- [InteractiveBrowserCredential](#)

Utiliser Azure Service Authentication

Paquets NuGet requis:

- ❖ Azure.Security.KeyVault.Secrets
- ❖ Azure.Identity

Utiliser Azure Service Authentication

1 référence

```
public class SecureConfigurationService : ISecureConfigurationService
```

```
{
```

```
    private SecretClient _secretClient;
```

1 référence

```
    public SecureConfigurationService()
```

```
    {
```

```
        var vaultUri = new Uri("https://demokeyvaultreference-kv.vault.azure.net/");
```

```
        _secretClient = new SecretClient(vaultUri, new DefaultAzureCredential());
```

```
    }
```

2 références

```
    public string ObtenirSecret(string nomDuSecret)
```

```
    {
```

```
        // obtenir la dernière version du secret
```

```
        //string valeurDuSecret = _secretClient.GetSecret(nomDuSecret).Value.Value;
```

```
        // obtenir une version spécifique du secret
```

```
        string valeurDuSecret = _secretClient.GetSecret(nomDuSecret, "ac339852329f4314bbe157d31b708fdd").Value.Value;
```

```
        return valeurDuSecret;
```

```
    }
```

```
}
```

Utiliser Azure Service Authentication

Home > demokeyvaultreference-kv

demokeyvaultreference-kv | Access policies ...
Key vault

Search (Ctrl+ /)

Save Discard Refresh

Soft Delete will be automatically enabled on this key vault after September 30, 2021. Click here

Enable Access to:

- ☐ Azure Virtual Machines for deployment ⓘ
- ☐ Azure Resource Manager for template deployment ⓘ
- ☐ Azure Disk Encryption for volume encryption ⓘ

Permission model

- ☐ Vault access policy
- ☒ Azure role-based access control

Please use Access Control (IAM) to configure access policy. [Learn more](#)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Events

Settings

Keys

Secrets

Certificates

Access policies

Utiliser Azure Service Authentication

[Home](#) > [demokeyvaultreference-kv](#)

 **demokeyvaultreference-kv** | Access control (IAM) ...

Key vault

<<

[+ Add](#)

[↓ Download role assignments](#)

[≡ Edit columns](#)

[↻ Refresh](#)

[✕ Remove](#)

[♥ Got feedback?](#)

[Overview](#)

[Activity log](#)

[Access control \(IAM\)](#)

[Tags](#)

[Diagnose and solve problems](#)

[Events](#)

[Settings](#)

[Keys](#)

Contributor



CesiDev_SP

App

[Contributor](#) ⓘ

[Subscription](#) (Inherited)

None



jeusreladev_sp

App

[Contributor](#) ⓘ

[Subscription](#) (Inherited)

None

Key Vault Secrets User



LocalWebAppCallingKV
/subscriptions/57e0f12d-a...

App Service or Function App

[Key Vault Secrets User](#) ⓘ

This resource

None



Steve Rogers
SRogers@tidjanibelmanso...

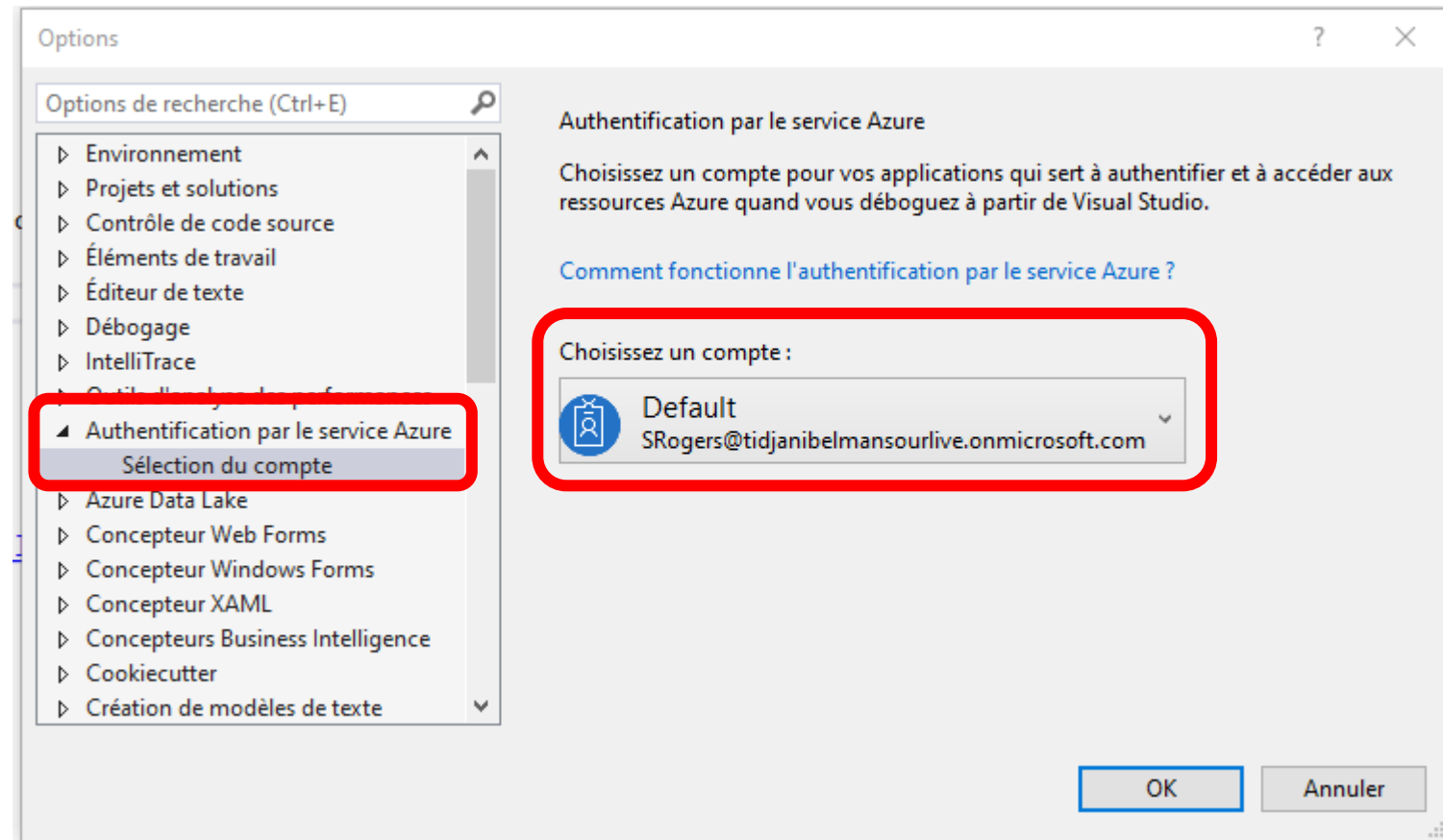
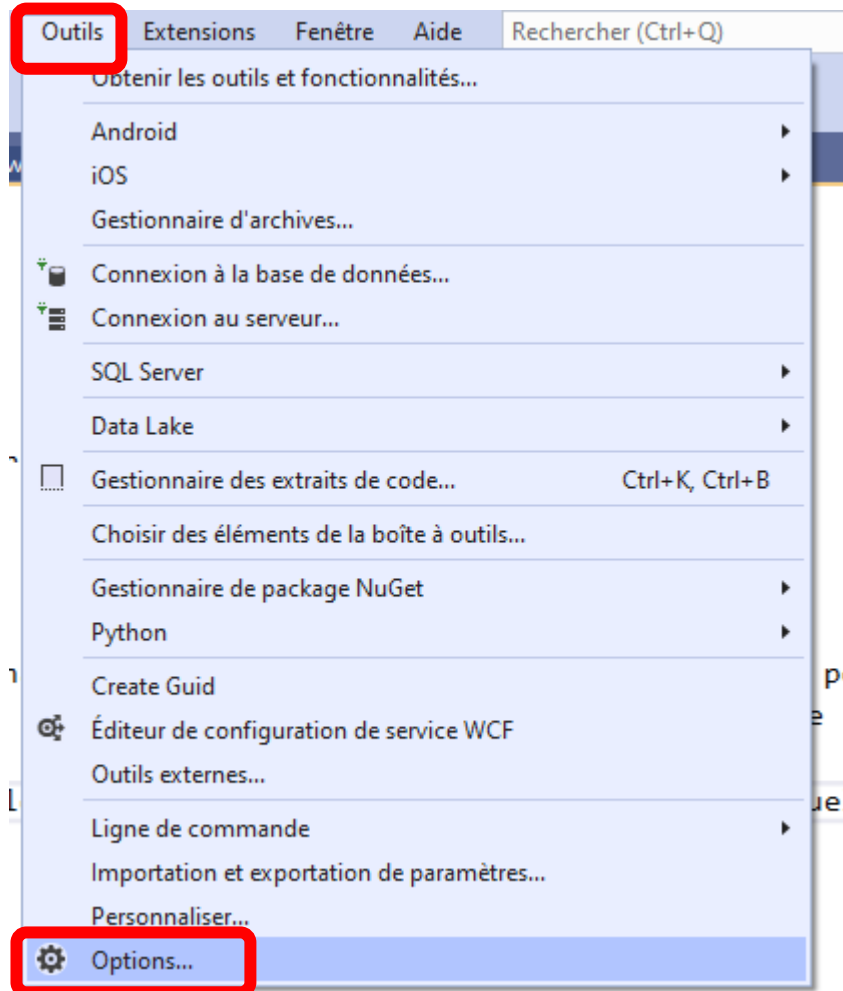
User

[Key Vault Secrets User](#) ⓘ

This resource

None

Utiliser Azure Service Authentication



Utiliser Azure Service Authentication

Startup.cs

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContext<advworksContext>(options =>
        options.UseSqlServer(Configuration.GetConnectionString("advWorksDatabase")));

    services.AddControllersWithViews();
}
```

Utiliser Azure Service Authentication

Models\advworksContext.cs

```
public advworksContext(DbContextOptions<advworksContext> options, IWebHostEnvironment env)
    : base(options)
{
    if (Database.IsSqlServer())
    {
        var connection = (SqlConnection)Database.GetDbConnection();
        var trc = new Azure.Core.TokenRequestContext(new string[] { "https://database.windows.net/" });
        connection.AccessToken = new DefaultAzureCredential().GetToken(trc).Token;
    }
}
```

Utiliser Azure Service Authentication

Activer le Active Directory

Admin sur le serveur SQL

Azure:

The screenshot displays the Azure portal interface for configuring a SQL server. The breadcrumb navigation at the top reads: Home > advworks (mibserver/advworks) > mibserver. The page title is "mibserver | Azure Active Directory" with a sub-label "SQL server".

On the left-hand navigation pane, the "Settings" section is expanded, and the "Azure Active Directory" option is highlighted with a red rectangle. Other visible options in the settings list include "Access control (IAM)", "Tags", "Diagnose and solve problems", "Quick start", "SQL databases", and "SQL elastic pools".

On the right-hand side of the page, the "Set admin" button is highlighted with a red rectangle. Below it, the "Azure Active Directory admin" section shows the "Admin name" as "DB-Admins" with a green checkmark icon, also highlighted by a red rectangle. The text "(Admin Object/App ID: d110d5)" follows. Below this, the "Azure Active Directory authentication only" section contains a checkbox labeled "Support only Azure Active Directory authentication for th".

Utiliser Azure Service Authentication

Grant sur la BD:

-- Compte pour execution dans Visual Studio

```
CREATE USER [SRogers@tidjanibelmansourlive.onmicrosoft.com] FROM EXTERNAL PROVIDER;  
GRANT SELECT TO [SRogers@tidjanibelmansourlive.onmicrosoft.com];
```

-- Managed Identity pour execution sur Azure

```
CREATE USER [LocalWebAppCallingKV] FROM EXTERNAL PROVIDER;  
GRANT SELECT TO [LocalWebAppCallingKV];
```


Utiliser Azure Service Authentication

3 références

```
public class HomeController : Controller
```

```
{
```

```
    private readonly ILogger<HomeController> _logger;
```

```
    private readonly advworksContext _context;
```

0 références

```
    public HomeController(ILogger<HomeController> logger, advworksContext context)
```

```
    {
```

```
        _logger = logger;
```

```
        _context = context;
```

```
    }
```

0 références

```
    public IActionResult Index()
```

```
    {
```

```
        var keithHarris = _context.Customers.FirstOrDefault(c => c.FirstName.ToLower() == "keith" && c.LastName.ToLower() == "harris");
```

```
        ViewBag.CustomerName = $"{keithHarris.FirstName} {keithHarris.LastName} de l'entreprise {keithHarris.CompanyName}";
```

```
        ViewBag.KvSecret = new Services.SecureConfigurationService().ObtenirSecret("superSecretMessage");
```

```
        return View();
```

```
    }
```

Utiliser Azure Service Authentication

Si le compte qui exécute l'application n'a pas les permissions requises, on aura cette erreur:

An unhandled exception occurred while processing the request.

SqlException: Login failed for user '<token-identified principal>'.

Microsoft.Data.ProviderBase.DbConnectionPool.CheckPoolBlockingPeriod(Exception e)

Stack Query Cookies Headers Routing

SqlException: Login failed for user '<token-identified principal>'.

Microsoft.Data.ProviderBase.DbConnectionPool.CheckPoolBlockingPeriod(Exception e)

Microsoft.Data.ProviderBase.DbConnectionPool.CreateObject(DbConnection owningObject, DbConnectionOptions userOptions, Dt

Microsoft.Data.ProviderBase.DbConnectionPool.UserCreateRequest(DbConnection owningObject, DbConnectionOptions userOptio

Microsoft.Data.ProviderBase.DbConnectionPool.TryGetConnection(DbConnection owningObject, uint waitForMultipleObjectsTimeo
userOptions, out DbConnectionInternal connection)

Microsoft.Data.ProviderBase.DbConnectionPool.TryGetConnection(DbConnection owningObject, TaskCompletionSource<DbConne
connection)

Microsoft.Data.ProviderBase.DbConnectionFactory.TryGetConnection(DbConnection owningConnection, TaskCompletionSource<DI
oldConnection, out DbConnectionInternal connection)

Utiliser Azure Service Authentication avec VS Code

AZURE: APP SERVICE


Waiting for Azure sign-in...

Installer l'extension

« Azure Account » et

s'authentifier à Azure

Extension: Azure Account



Azure Account

ms-vscode.azure-account

Microsoft | 2,767,380 | ★★★★★ | Repository | License

A common Sign-In and Subscription management extension for VS Code.

Disable Uninstall

Details Contributions Changelog

Azure Account and Sign-In

The Azure Account extension provides a single Azure sign-in and subscription filtering experience for all other Azure extensions. It makes Azure's Cloud Shell service available in VS Code's integrated terminal.

Commands

Command	
Azure: Sign In	Sign in to your Azure subscription.
Azure: Sign In with Device Code	Sign in to your Azure subscription with a device code. Use this in setups where the Sign In command does not work.
Azure: Sign In to Azure Cloud	Sign in to your Azure subscription in one of the sovereign clouds.
Azure: Sign Out	Sign out of your Azure subscription.

Utiliser Azure Service Authentication

En résumé:

- ❖ Solution élégante et sécuritaire de conserver les secrets
- ❖ Facile à mettre en place
- ❖ Séparation claire entre la gestion des secrets (et des permissions sur les secrets) et le code applicatif
- ❖ Pas d'impact sur l'application si la valeur du secret venait à changer

MERCI!



@tidjani_b

ConFoo.CA
DEVELOPER CONFERENCE