

# WISE-PaaS/ DeviceOn Azure Kubernetes Service (AKS) Version

## Quick Start Guide: How to Retrieve My Azure Account Information?

English v-1.0.1

### Table of Content

- Method 1: Create & Get Information on Azure Portal..... 2
  - Step 1. Create Your Application ..... 2
  - Step 2. Get Subscription ID..... 3
  - Step 3. Get Application & Tenant ID ..... 5
  - Step 4. Add & Get Client Secret..... 6
- Method 2: Create via Azure CLI (Command-line Tool)..... 7
  - Step 1. Install Azure CLI ..... 7
  - Step 2. Sign in Account..... 7
  - Step 3. Get Subscription ID & Copy Output ..... 7
  - Step 4. Create service principal and get Application ID, Tenant ID and Client Secret..... 7

To deploy DeviceOn to your Azure subscription, a set of Azure account information is required. You will be prompted to enter required information on the WISE-PaaS/Marketplace when chose to Create Deployment. WISE-PaaS Marketplace will use the given information to create deployment to your Azure subscription.

If you already have DeviceOn installed in your Azure service and do not wish to install DeviceOn again, you can skip this document.

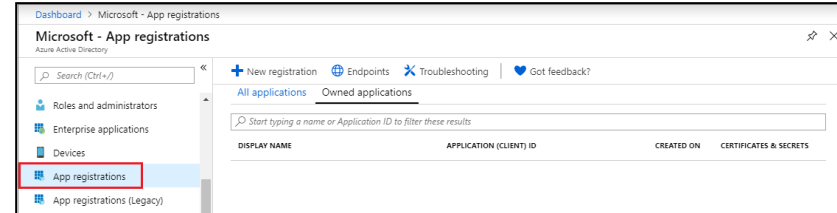
## Method 1:

# Create & Get Information on Azure Portal

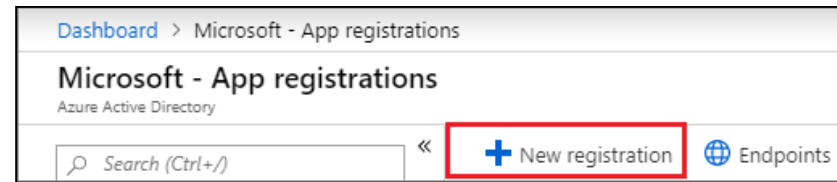
## Step 1. Create Your Application

- 1.1. Log into your [Azure Portal](#) >.
- 1.2. Select **[Azure Active Directory]**
- 1.3. Select **[App registrations]**
- 1.4. Add **[New Registration]**
- 1.5. Setup your **Application Name** then click **[Register]**.
  - Enter your Application display name in **Name** filed.
  - Setup **Supported account types** by selecting account type to access this API.
  - Under **Redirect URI**, select Web for the type of application you want to create. Enter the URI where the access token is sent to. **Note: You cannot create Native application credential nor use the type for an automated application.**

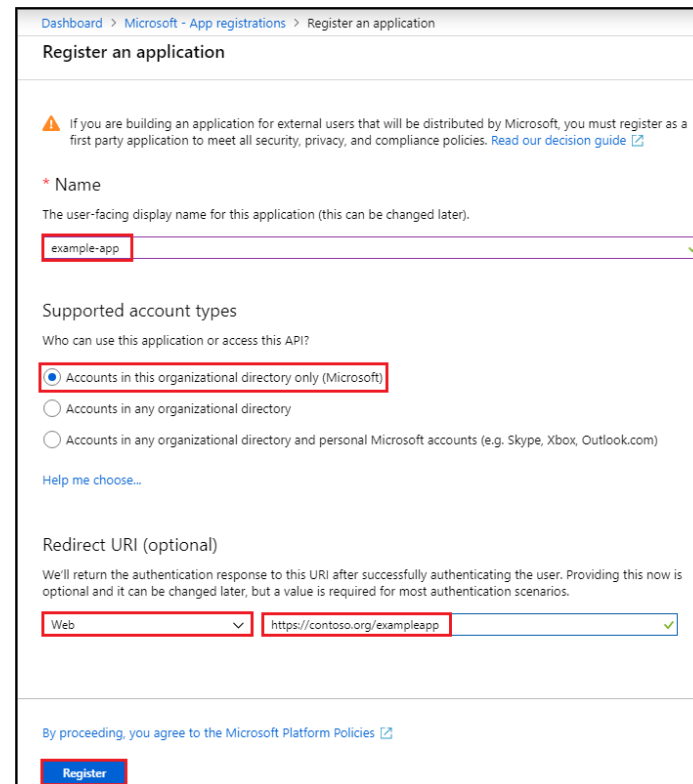
### Step 1.3 Select App registrations



### Step 1.4 Add New registration



### Step 1.5 Setup Application name to Register



## Step 2. Get Subscription ID

To access resources in your subscription, you must assign Application to a role. You can set the scope at the level of Subscription, Resource Group, or Resource; Permissions are inherited to lower levels of scope.

[For more details, see RBAC: Built in Roles >](#)

2.1. Select **All services** then select **Subscriptions** to setup the level of scope you wish to assign this application.

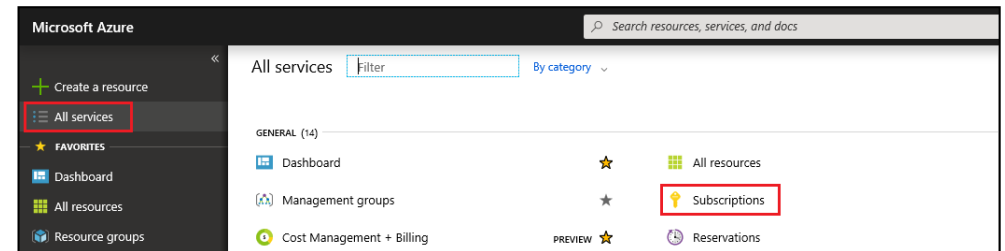
2.2. Find the **Subscription** you would like to assign to the Application created in the Step 1 and copy the **Subscription ID** for the Azure information requested on the Marketplace later.

**! Troubleshoot:** If you do not see the subscription you're looking for, select **global subscriptions** filter. Make sure the subscription you want is selected for the portal.

2.3. Select **Access control (IAM)** then **Add role assignment**

Fig1. Marketplace Fields

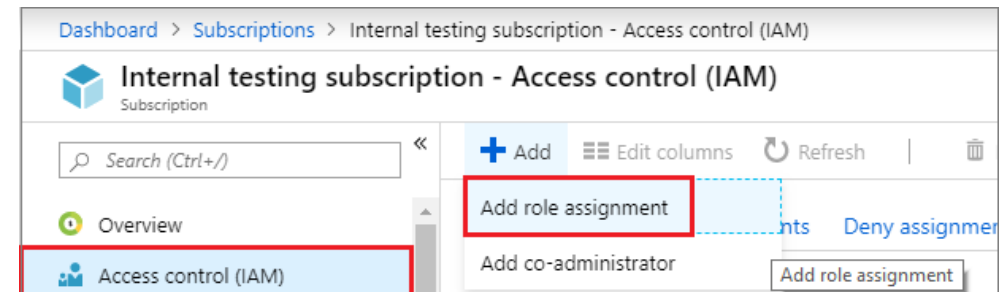
### Step 2.1 Select Subscriptions



### Step 2.2 Copy Subscription ID

SUBSCRIPTION	SUBSCRIPTION ID
Visual Studio Enterprise - MPN	[REDACTED]

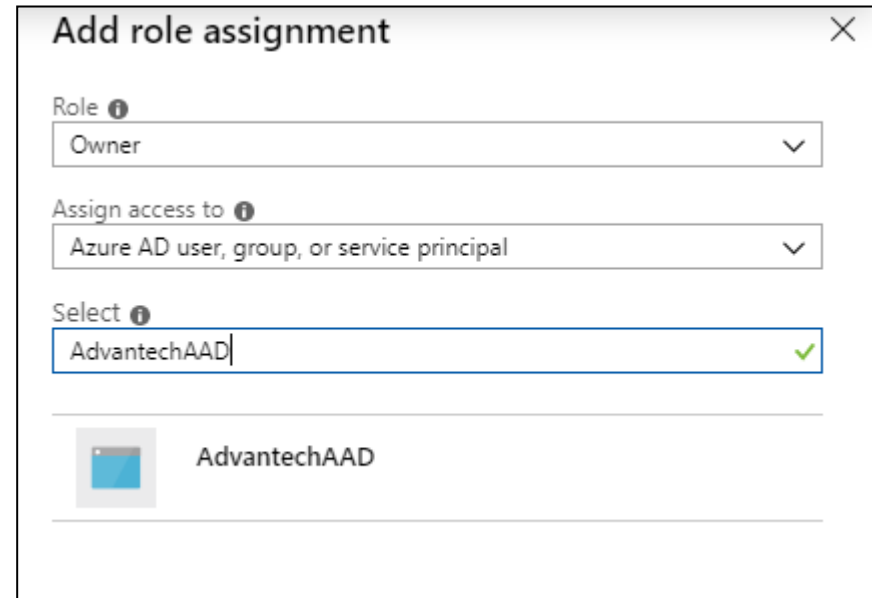
### Step 2.3 Setup Access Control



2.4. Select the **Owner** role. By default, Azure AD applications aren't displayed in the available options. To find your application, search for the name.

2.5. Click **Save** to finish assigning the role. You see your application in the list of users assigned to a role for that scope.

## Step 2.4 Add Role Assignment




**Add role assignment** ✕

Role ⓘ  
Owner

Assign access to ⓘ  
Azure AD user, group, or service principal

Select ⓘ  
AdvantechAAD ✓

---

 AdvantechAAD

### Step 3. Get Application & Tenant ID

#### 3.1. Select Azure Active Directory

3.2. From **App registrations** in Azure AD, select your application

3.3. Copy the **Directory (tenant) ID** for the Azure information requested on the Marketplace later.

3.4. Copy the **Application (client) ID** for the Azure information requested on the Marketplace later.

Would You Like To Deploy This App To Microsoft Azure?  
Please enter your Microsoft Azure information to initiate deployment  
(don't have Azure account yet? [contact us to create one](#))

**Azure Subscription ID:**  
enter your subscription ID a

**Application (client) ID:**  
enter your service principal application ID b Application ID

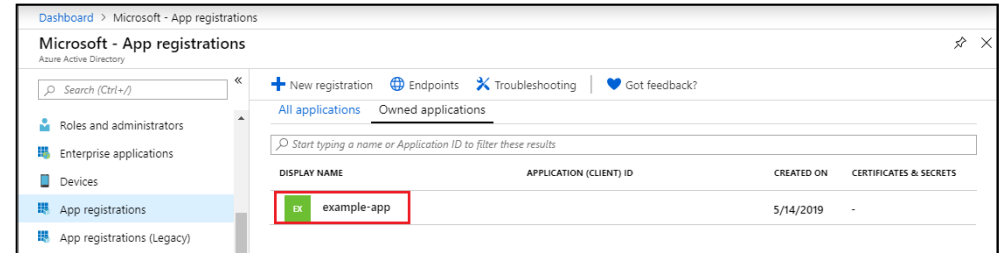
**Directory (tenant) ID:**  
enter your tenant ID c Tenant ID

**Client Secret:**  
enter your service principal password d

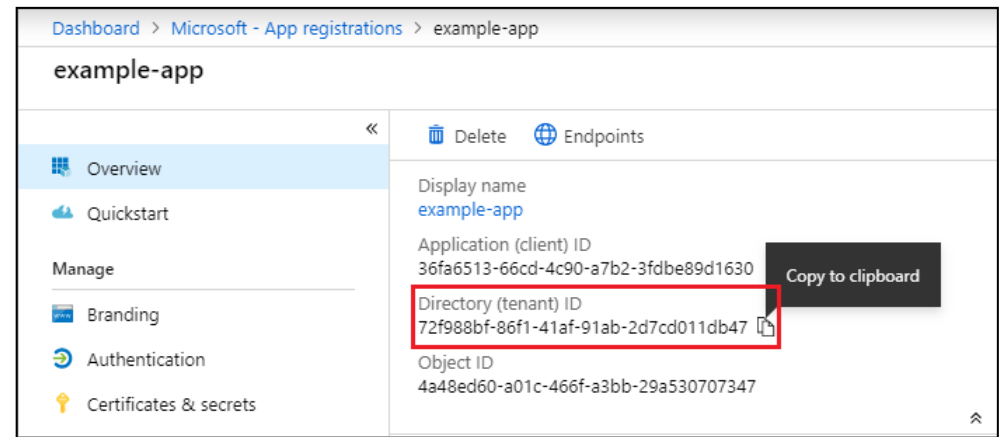
Fig1. Marketplace Fields

Done. Skip Deployment
Create Deployment

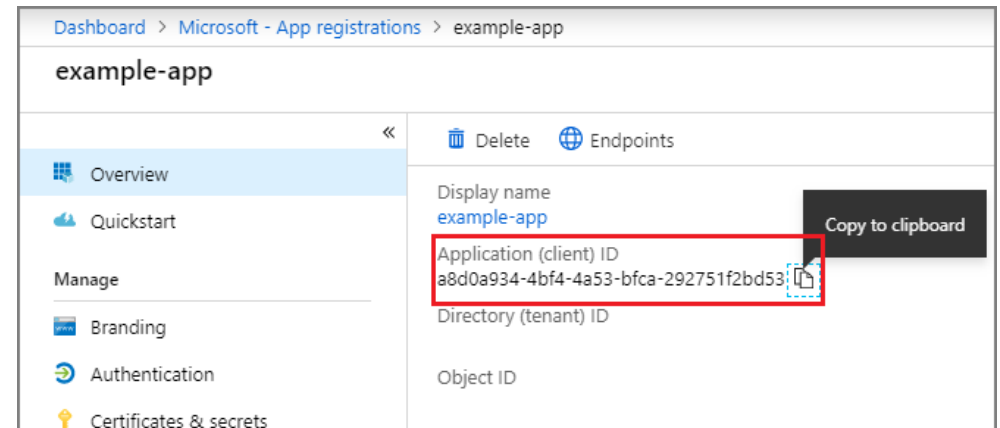
#### Step 3.2 Find Your Application



#### Step 3.3 Copy Directory ID (ref. Marketplace filed #C)



#### Step 3.4 Copy Application ID (ref: Marketplace field #B)



## Step 4. Add & Get Client Secret

4.1. Select **[Certificates & secrets]**

4.2. Select **Client secrets** then **New client secret**

4.3. Give new client secret description, setup expiration period. Then Click **[Add]**

Would You Like To Deploy This App To Microsoft Azure?  
Please enter your Microsoft Azure information to initiate deployment  
(don't have Azure account yet? contact us to create one)

**Azure Subscription ID:**  
enter your subscription ID **a**

**Application (client) ID:**  
enter your service principal application ID **b**

**Directory (tenant) ID:**  
enter your tenant ID **c**

**Client Secret:**  
enter your service principal password **d** ← **Client Secret**

Fig1. Marketplace Fields

Done. Skip Deployment    Create Deployment

### Step 4.2 Set Client Secret for your Application

Dashboard > Microsoft - App registrations > example-app

**example-app**

Overview    Quickstart    Manage    Authentication    **Certificates & secrets**    Delete    Endpoints

Display name: **example-app**  
Application (client) ID: a8d0a934-4bf4-4a53-bfca-292751f2bd53  
Directory (tenant) ID:  
Object ID:

### Step 4.3 Add Client Secret

Add a client secret

Description: **demo secret**

Expires:  
 In 1 year  
 In 2 years  
 Never

**Add**    Cancel

### Step 3.4 Copy Client Secret (ref: Marketplace field #D)

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

[+ New client secret](#)

DESCRIPTION	EXPIRES	VALUE
demo secret	5/14/2020	nWu9HVZ7Rnj.2y7XSkVyUngZ][x9Ze [📄]

**Method 2:**  
**Create via Azure CLI (Command-line Tool)**

**Step 1. Install Azure CLI**

[For details, please view step by step guide >](#)

**Step 2. Sign in Account**

```
1. C:\>az login
```

Note: If the CLI can open your default browser, it will do so and load a sign-in page. Otherwise, you need to open a browser page and follow the instructions on the command line to enter an authorization code after navigating to <https://aka.ms/device/login> in your browser. Sign in with your account credentials in the browser.

**Step 3. Get Subscription ID & Copy Output**

```
2. C:\>az account show --query id
```

**Step 4. Create service principal and get Application ID, Tenant ID and Client Secret**

```
3. C:\>az ad sp create-for-rbac --name ServicePrincipalName
```

Reference: [Create an Azure service principal with Azure CLI >](#)

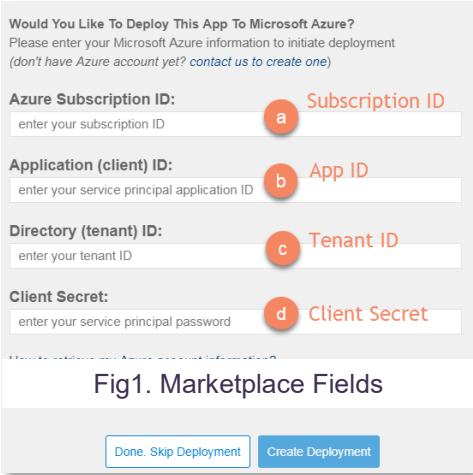


Fig1. Marketplace Fields

**Step 3 Get Subscription ID**

```
C:\>az login
Note, we have launched a browser for you to login. For old experience with device code, use "az login --use-device-code"
You have logged in. Now let us find all the subscriptions to which you have access...
{
  "cloudName": "AzureCloud",
  "id": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "isDefault": true,
  "name": "Visual Studio Enterprise \u00d02013 MPN",
  "state": "Enabled",
  "tenantId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "user": {
    "name": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
    "type": "user"
  }
}
```

**Step 4 Get Application, Tenant IDs & Client Secret**

```
C:\>az ad sp create-for-rbac --name AdvantechAD
Retrying role assignment creation: 1/36
Retrying role assignment creation: 2/36
{
  "appId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "displayName": "AdvantechAD",
  "name": "http://AdvantechAD",
  "password": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "tenant": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
}
```