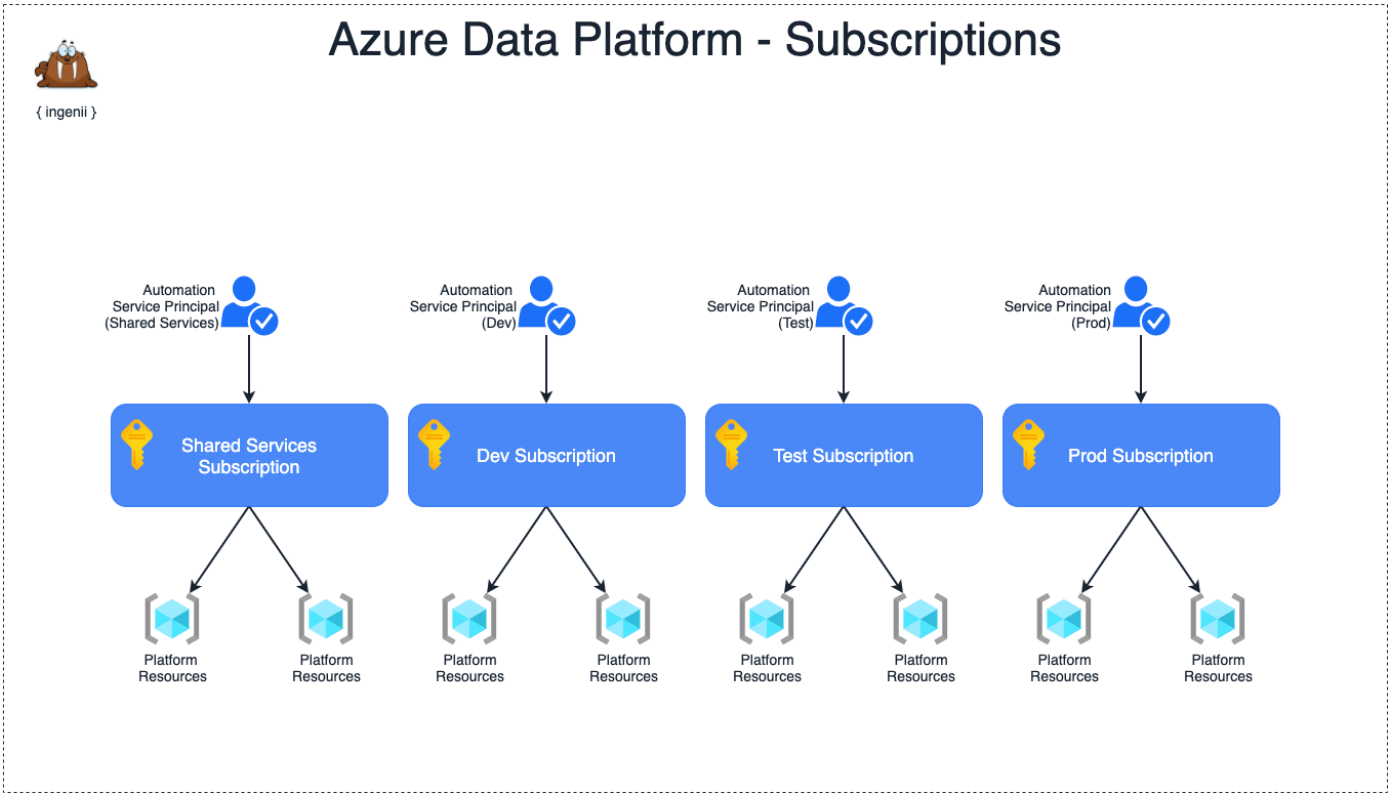
Ingenii Azure Data Platform Requirements

Table of Contents

1. Introduction

The consumer of this platform will have to complete specific pre-requisite steps to ensure all requirements are in place before the deployment can start.

2. Azure Subscriptions



The platform requires four Azure Subscriptions to match the resource environments. Segregating environment resources in different Subscriptions provides better access and billing control.

- **Shared Subscription** - Contains resources for the Shared Environment
- **Dev Subscription** - Contains resources for the Dev Environment
- **Test Subscription** - Contains resources for the Test Environment
- **Prod Subscription** - Contains resources for the Prod Environment

> Note: It is possible to create all resources in the same Subscription. Ingenii does not recommend this approach unless this is the only viable option.

2.1 Create Azure Subscriptions

Please follow the official Microsoft documentation on how to create Azure Subscriptions.

Suggested Subscription Names

- **ADPShared**
- **ADPDev**
- **ADPTest**
- **ADPProd**

ADP stands for Azure Data Platform.

3. Azure Service Principals

A service principal is an application within Azure Active Directory. When created, the application generates authentication tokens. Terraform uses these tokens as a way to authenticate against Azure and deploy the infrastructure resources.

We need to create a Service Principal per Subscription.

Suggested Service Principal Names

- ADPSharedAutomation
- ADPDevAutomation
- ADPTestAutomation
- ADPProdAutomation

3.1 Create Azure Service Principal

The steps below outline the method of creating a Service Principal using the Azure CLI. You need to repeat the steps for each Service Principal.

Step 1 - Log in to the Azure CLI

```
$ az login
```

Step 2 - List All Subscriptions

```
$ az account list
```

The output should return all Subscriptions that we have created. Choose the Subscription that you are creating Service Principal for and make a note of the `id` value. That is the `Subscription Id`.

For this example, the Subsciprtion Id is `bd36c548-3b15-4570-ae81-08d9c4c480cb`. We will use that id in subsequent commands.


```
{
  "cloudName": "AzureCloud",
  "homeTenantId": "360a8067-cf57-4e52-bacb-83702821f1f1",
  "id": "bd36c548-3b15-4570-ae81-08d9c4c480cb",
  "isDefault": true,
  "managedByTenants": [
    {
      "tenantId": "a27c5a35-9c0e-4b06-adb0-05e82f48947f"
    }
  ],
  "name": "ADPDev",
  "state": "Enabled",
  "tenantId": "360a8067-cf57-4e52-bacb-83702821f1f1",
```

```
  "user": {
    "name": "user@example.com",
    "type": "user"
  }
}
```

Step 3 - Specify Subscription

Tell the Azure CLI what Subscription Id we are operating with:

```
$ az account set --subscription="bd36c548-3b15-4570-ae81-08d9c4c480cb"
```

Step 4 - Create Service Principal

We are creating a new Service Principal called `ADPDevAutomation`, assigning it the role `Owner` to the Subscription `ADPDev` (bd36c548-3b15-4570-ae81-08d9c4c480cb).

```
$ az ad sp create-for-rbac --name="ADPDevAutomation" --role="Owner" --
scopes="/subscriptions/bd36c548-3b15-4570-ae81-08d9c4c480cb" --years 1
```

> The Service Principal credentials will be valid for a year from the time we create them. You can increase the lifetime by setting the `--years` switch in Step 4 to a different value than 1.

The command will output values like these:

```
{
  "appId": "b4dce5a6-c62a-4a6f-b6d6-110d695cab42",
  "displayName": "ADPDevAutomation",
  "name": "b4dce5a6-c62a-4a6f-b6d6-110d695cab42",
  "password": "mZ2Q0-j6manfKj5ZQJHSJSYhZnQSqaQPBY",
  "tenant": "a27c5a35-9c0e-4b06-adb0-05e82f48947f"
}
```

> **Please keep the values somewhere safe.**

Here is how they map to the Terraform credentials:

- `appId` is the `ARM_CLIENT_ID`
- `password` is the `ARM_CLIENT_SECRET`
- `tenant` is the `ARM_TENANT_ID`

We also need the subscription id from Step 2.

- `subscription` is the `ARM_SUBSCRIPTION_ID`

Make sure to save these credentials to a password manager. The Service Principal password will not be displayed again after you close the terminal window.

3.2 Assign Azure Active Directory Permissions

The Azure Data Platform creates Azure AD Groups and Azure AD Applications (Service Principals) as part of the deployment.

We need to grant access to our Service Principal to create Azure AD Groups and Applications.

Step 1 - Grant API Permissions

Navigate to the Azure Active Directory overview within the Azure Portal and select the App Registrations blade. You should see the Service Principals (applications) we have created. Click on the display name of the Service Principal you want to manage.

Go to the `API Permissions` blade and click `Add a permission`. In the new pane that opens, select `Azure Active Directory Graph` (under the Supported Legacy APIs subheading). **Do not select** "Microsoft Graph" as the Azure Terraform provider does not currently make use of it.

Choose `Application Permissions` for the permission type, and check the permissions below:

- `Application.ReadWrite.All`
- `Directory.ReadWrite.All`

These permissions will only allow us to create/manage Applications and Groups but not delete them.

Step 2 - Consent the API Permissions

Once you have assigned the permissions, you will need to grant admin consent. This requires that you are signed to the Azure Portal as a Global Administrator.
Click the "Grant admin consent" button and confirm this action.

4. Azure DevOps

Azure DevOps is used as a CI/CD platform by some of the Azure Data Platform services. (e.g., Azure Data Factory, Azure Databricks, Data Build Tool, etc.)

The Terraform provider for Azure DevOps currently supports authentication via Personal Access Tokens only.

4.1 Create Personal Access Token

Step 1 - Go to Azure DevOps portal

> If you have never used Azure DevOps, you will have to sign up using your Microsoft credentials and create a new Organization.

Step 2 - Click on your User icon at the top right corner

Step 3 - Select "Personal access tokens"

Step 4 - Create a new token

Select `+ New Token` to open the Token Creation pane.

Suggested Token Name: ADPSharedAutomation

Set the expiration to 1 year or more. Match the expiration time to your Service Principal expiration time.

Click on `Scopes` and choose `Custom defined`.

> You might need to click on `Show more scopes` to see all scopes.

Select the following permissions:

Agent Pools

- ✅ Read

Build

- ✅ Read & Execute

Code

- ✅ Full
- ✅ Status

Environment

- ✅ Read & Manage

Graph

- ✅ Read

Identity

- ✅ Read & Manage

Member Entitlement Management

- ✅ Read & Write

Project and Team

- ✅ Read, Write & Manage

Secure Files

- ✅ Read, Create & Manage

Security

- ✅ Manage

Variable Groups

- ✅ Read, Create & Manage

Click `Create` to generate the new token.

Keep the token safe in your password manager.

---

5. Additional Questions

- Which is the primary Azure region we should use for the deployment? - e.g. **UKSouth**
- What Azure VNET address space (CIDR) should we use?
  - We require 3x /16 (255.255.0.0) ranges. One for each environment (Dev, Test, Prod)
  - We require 1x /20 (255.255.240.0) range. Reserved for the Shared services environment.
- What resource prefix can we use? - The resource prefix will be added to every deployed resource. E.g. `prefix`**-resource-name**. For example, if your company name is **Fabrikam**, the prefix can be **fbrkm**. *The prefix is limited to 5 alphanumerical characters*.

6. Requirements Checklist

- **Azure Subscriptions**
  - ✅ Shared
  - ✅ Dev
  - ✅ Test
  - ✅ Prod
- **Azure Service Principals**
  - ✅ Shared
  - ✅ Dev
  - ✅ Test
  - ✅ Prod
- **Azure DevOps Personal Access Token**
  - ✅ Shared
- **Additional Questions**
  - ✅ Azure Region
  - ✅ Azure VNET Address Space
  - ✅ 3x /16 ranges
  - ✅ 1x /20 range
  - ✅ Resource Prefix