Creating New Environment

Last updated by | Ross Tappin | Dec 21, 2022 at 2:25 AM MST

Azure Active Directory B2C Tenant

In Azure portal, make sure you are in the SparkChange tenant and create an Azure Active Directory B2C Tenant for the new environment:

1. In the search bar, search for 'Azure Active Directory'
2. Click on Manage Tenants
3. Click on Create
4. Select 'Azure Active Directory (B2C)' as the tenant type
5. Click on Configuration
6. Enter the organization name - this is the name of the new directory to be created (for example, sparkchangedev)
7. Enter the initial domain name - this is usually the same as the organization name
8. Select the country/region - this is usually United Kingdom
9. Select Subscription - If you are setting up a non production environment, select 'Spark Dev/Test Deployments' from the drop down list
10. Select the resource group - select 'CoreSupport' from the drop down list
11. Click Create + Review
12. Review the configuration and click Create
13. Register a new application called 'Terraform'
    1. Switch to the directory of the new Azure AD B2C Tenant
    2. In the search bar, search for and click on 'Azure AD B2C'
    3. Go to 'App Registration' by clicking it from the menu
    4. Click on 'New Registration'
    5. Enter 'Terraform' as the name of the application
    6. For the supported account type, select 'Accounts in any identity provider or organizational directory (for authenticating users with user flows)'
    7. Ensure the 'Grant admin consent to openid and offline_access permissions' is checked
    8. Click Register

Terraform Application Registration

1. Create a new certificate for Terraform application

    1. In the Terraform application, click Certificates & secrets
    2. Create a new secrete by clicking New client section
    3. In the description enter 'Terraform certificate'
    4. Select expiry from the drop down list - this is usually the max available from the drop down list (24 months) and click Add
    5. Take note of the value (make sure you keep a temp copy of this as it will be required to be added to the terraform variables file later)
2. Give permissions to 'Terraform' application

    1. In the search bar, search for 'Azure Active Directory'
    2. From the menu, select Roles and administrators

3. Search for 'Cloud application administrator' and click it
4. Click on Add assignment
5. Search for Terraform and click it
6. Click Add

Prepare and Run Provisioning Scripts

1. Clone the sparkbuild project:
   - git clone https://sparkchange@dev.azure.com/sparkchange/spark/_git/sparkbuild
2. Ensure you have the latest from the sparkbuild project by running:
   - git pull
3. In the sparkbuild project, under /deployments/environments create a new folder. Name the folder the name of the environment you want to set up. For example, if you want to set up a new dev environment call the folder "dev".

4. Add an environment variables file under the new environment folder you created called "terraform.tfvars". This will hold the environment variables that Terraform will use to set up the new environment.

5. Open the new terraform.tfvars file you created. In here you will add the following variables:
   - subscription_id
     1. In the search bar on Azure portal, search for 'Subscriptions'
     2. For any non production environment find and use the subscription id of 'Spark Dev/Test Deploymnets' subscription
   - tenant_id
     1. In the search bar on Azure portal, search for 'Azure Active Directory'
     2. Click on 'Manage Tenants'
     3. Find and use the organization ID for sparkchange.io domain
   - client_id
     If the environment you are setting up is a non production environment then we use 'TerraformDevTest'. Since we already have 'TerraformDevTest' created we will go ahead and use that client id.

     1.
   - client_secret
   - client_principal_id
   - container_registry
   - log_analytics_workspace_id
   - environment
   - sparkstorage_accountkey
   - rbac_cluster_admin_id
   - sql_sys_admin_id
   - vnet_range
   - dev_vnet_resource_group_id
   - dev_vnet_name
   - dev_vnet_id

  - infrastructure_subscription_id
  - sqlserver_private_dns_id
  - sqlserver_private_dns_name
  - b2c_tenant_id
  - b2c_client_id
  - b2c_client_secret
  - b2c_domain
  - spark_keyvault_id