How to deploy GIGA – Backend with Azure Pipelines

Introduction

Welcome to our concise guide designed for operations teams.

Learn key steps in setting up Azure DevOps pipelines for efficient software deployment and implementing changes in Azure Active Directory B2C for enhanced user management. This guide equips you to navigate modern development and security landscapes effectively.

Azure DevOps Pipelines: Discover how to automate build and deployment processes using Azure DevOps for Backend and Frontend GIGA Applications. Streamline collaboration and continuous integration with practical pipeline creation techniques.

Azure AD B2C Claims enrichment: Enhance user security and experiences by customizing Azure AD B2C claims enrichment by the GIGA Backend Application providing a JWT Token with the necessary claims to perform both authorization on Backend and display options on Frontend based on fetched permissions.

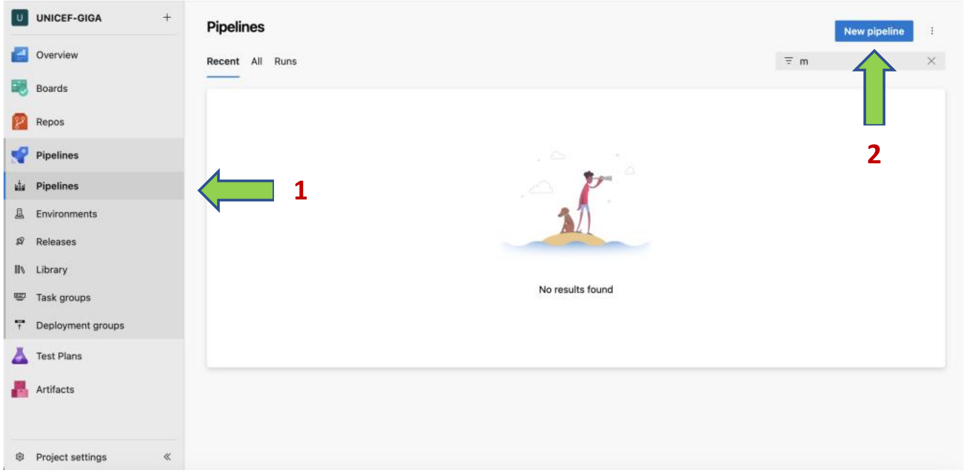
Empower our operations with these vital skills to drive excellence in our ever-evolving technological journey.

Requirements

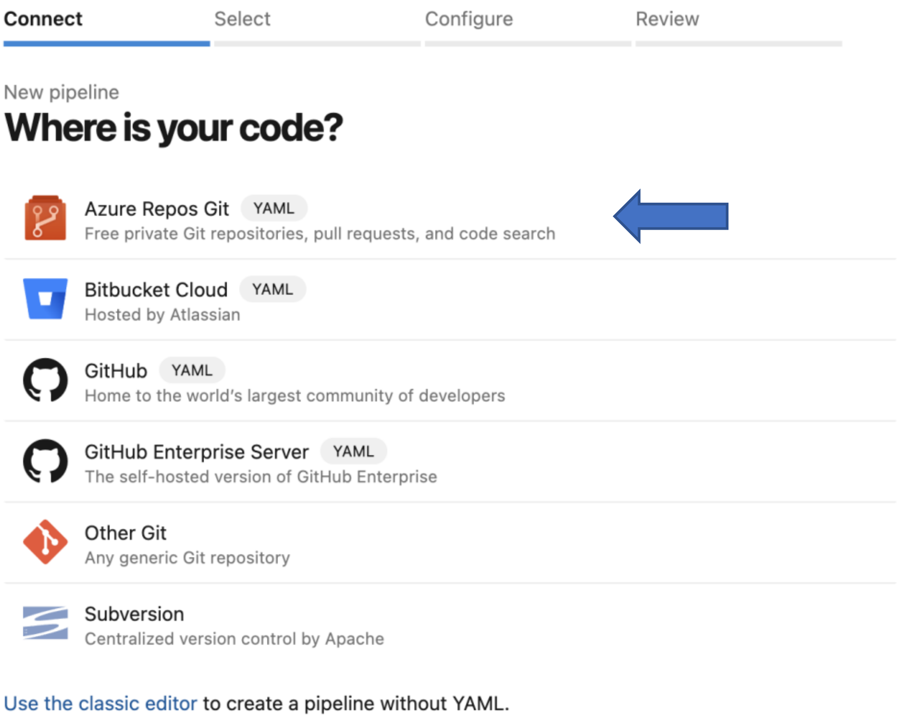
Before to create a pipeline, we need to create a service connection to Azure subscription and had “Build Administrator” permissions in Az DevOps.

Steps for deployment

1. Create a pipeline from GiGA-Backend repository.



2. Select “Azure Repos Git” option and select GIGA-BACK repository. If no exists, create from Azure Repos option. Then select “Use the classic editor” option.



Connect Select Configure Review

New pipeline

Where is your code?

Azure Repos Git YAML
Free private Git repositories, pull requests, and code search

Bitbucket Cloud YAML
Hosted by Atlassian

GitHub YAML
Home to the world’s largest community of developers

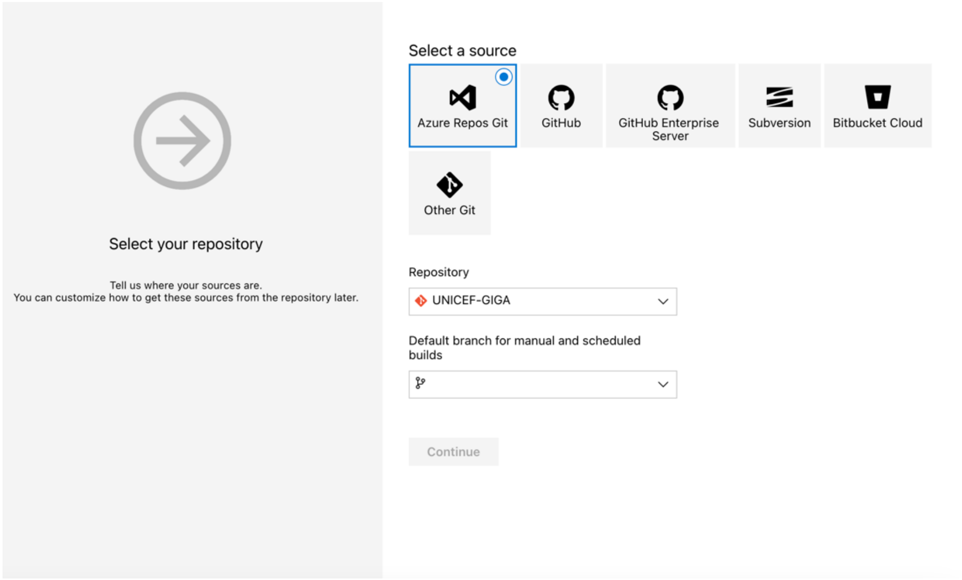
GitHub Enterprise Server YAML
The self-hosted version of GitHub Enterprise

Other Git
Any generic Git repository

Subversion
Centralized version control by Apache

Use the classic editor to create a pipeline without YAML.

3. Now select “Azure Repos Git” option and GIGA-Backend repo with the **develop** branch. Then press “Continue” button and press “Empty Job” as template



Select your repository

Tell us where your sources are.
You can customize how to get these sources from the repository later.

Select a source

● Azure Repos Git | GitHub | GitHub Enterprise Server | Subversion | Bitbucket Cloud | Other Git

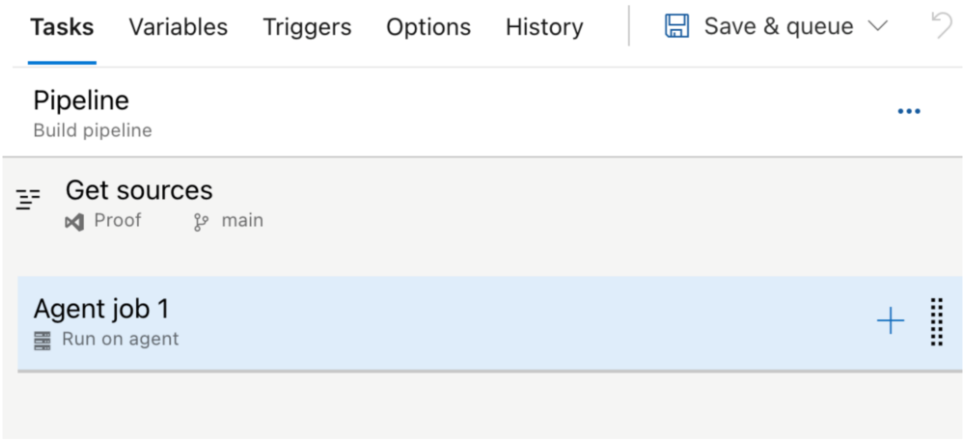
Repository
UNICEF-GIGA

Default branch for manual and scheduled builds

Continue

Select a template
Or start with an Empty job
Search

After press the empty job option, you can view the following screen: ￼



4. For configure “Agent Job” set the following properties:

   Display name: Agent Job Giga Backend

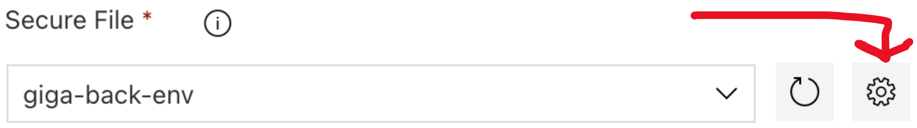
   Agent pool: Azure Pipelines

   Agent Specification: ubuntu-22.04

   *Rest of properties leave it in default

5. Add “Download secure file” task and configure with the following properties:

Secure file: <upload .env file>



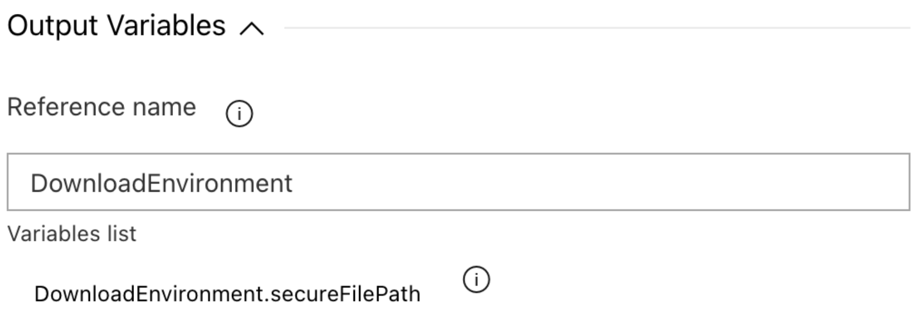
Secure File *

giga-back-env

```
# Variables - .env file
HOST=0.0.0.0
PORT=3333
NODE_ENV=dev
TZ=America/Sao_Paulo
ORIGINS_ALLOWED=http://localhost:3000
URL_FRONTEND=http://127.0.0.1:3000
APP_KEY={The secret to encrypt and sign different values in your application}
DRIVE_DISK=local
DB_CONNECTION=pg
```

```
DB_DEBUG=true
PG_HOST=localhost
PG_PORT=5432
PG_USER=postgres
PG_PASSWORD={PG_USER Password to connect to database}
PG_DB_NAME=unicef-giga
PGSSLMODE=allow
UNICEF_API_TOKEN={UNICEF API TOKEN}
UNICEF_API=https://uni-connect-services.azurewebsites.net/api/v1/
AZURE_STORAGE_CONNECTION_STRING=
AZURE_CONTAINER_NAME=attachments
CRON_TASK_EMAIL='*/5 * * * *'
CRON_TASK_MEASURES='* * 1 * *'
CRON_TASK_CONTRACTS_STATUS='* */10 * * *'
CRON_TASK_CASHBACK='*/30 * * * *'
CRON_TASK_AUTOMATIC_PAYMENTS='*/10 * * * *'
CRON_TASK_EMAIL_ENABLED=false
CRON_TASK_MEASURES_ENABLED=false
CRON_TASK_CONTRACTS_STATUS_ENABLED=false
CRON_TASK_CASHBACK_ENABLED=false
CRON_TASK_AUTOMATIC_PAYMENTS_ENABLED=false
EMAIL_FROM=no-reply@giga.dev.com
EMAIL_CLIENT_TO_USE=ETHEREAL {Options ETHEREAL, MAILJET}
EMAIL_MAILJET_API_KEY=
EMAIL_MAILJET_API_SECRET=
EMAIL_MAILJET_ADDRESS_TO_FAKE={If you get an free mailjet account, set here the email sender
validated in mailjet}
EMAIL_ETHEREAL_KEY=
EMAIL_ETHEREAL_SECRET=
JWT_PRIVATE_KEY=
JWT_PUBLIC_KEY=
URL_FRONTEND=http://127.0.0.1:3000
TENANT_NAME=
POLICY_NAME=
TENANT_ID=
APPLICATION_ID=
WEB3_NETWORK_ID=80001
WEB3_NODE_PROVIDER_URL=https://polygon-mumbai.g.alchemy.com/v2/
WEB3_NODE_PROVIDER_KEY=
WEB3_OWNER_SK=
WEB3_CONTRACTS_HANDLER_ADR=
```

And set Reference name: DownloadEnvironment



Output Variables

Reference name

DownloadEnvironment

Variables list

DownloadEnvironment.secureFilePath

*Rest of properties leave it in default

6. Add "Command Line" task and configure with the following properties:

Script:

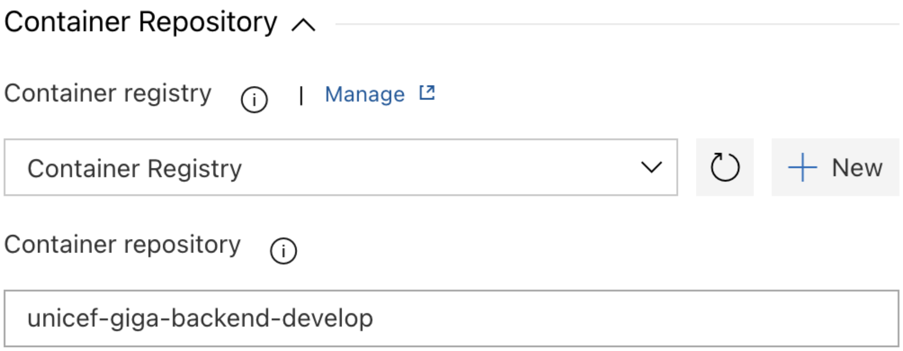
```
mv $(DownloadEnvironment.secureFilePath) ./.env
```

*Rest of properties leave it in default

7. Add “Docker” task and configure with the following properties:

Display name: buildAndPush

Select your container registry.



Container Repository

Container registry ⓘ | Manage

Container Registry

Container repository ⓘ

unicef-giga-backend-develop

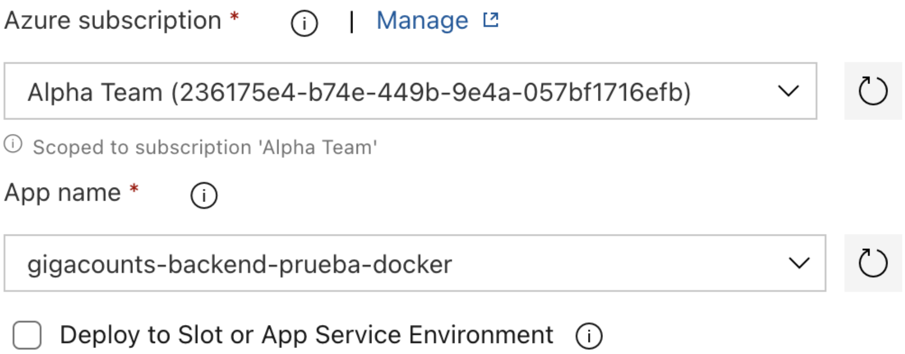
Container repository: unicef-giga-backend-develop

*Rest of properties leave it in default

8. Now, add “Azure Web App for Containers” task and configure with the following properties:

Display name: Azure Web App on Container Deploy: gigacounts-backend-prueba-docker

Select your webapp



Azure subscription * ⓘ | Manage

Alpha Team (236175e4-b74e-449b-9e4a-057bf1716efb)

ⓘ Scoped to subscription 'Alpha Team'

App name * ⓘ

gigacounts-backend-prueba-docker

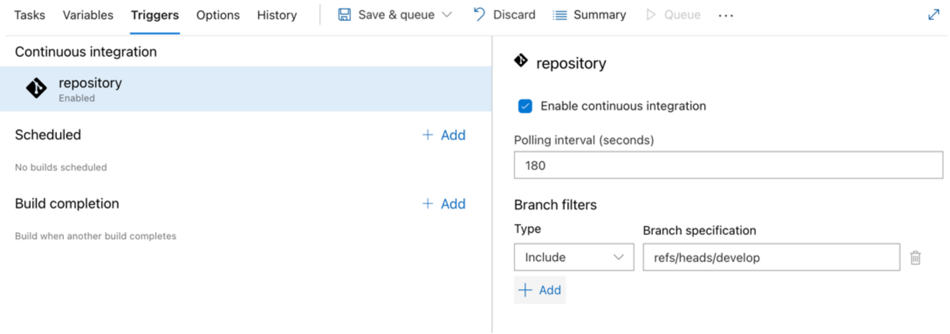
□ Deploy to Slot or App Service Environment ⓘ

Image name:

```
acralcatrazautomation.azurecr.io/unicef-giga-backend-develop:$(Build.BuildId)
```

*Rest of properties leave it in default

9. Finally enter to "Triggers” tab and set de following configuration:



Continuous integration

repository
Enabled

Scheduled + Add

No builds scheduled

Build completion + Add

Build when another build completes

repository

☑ Enable continuous integration

Polling interval (seconds)

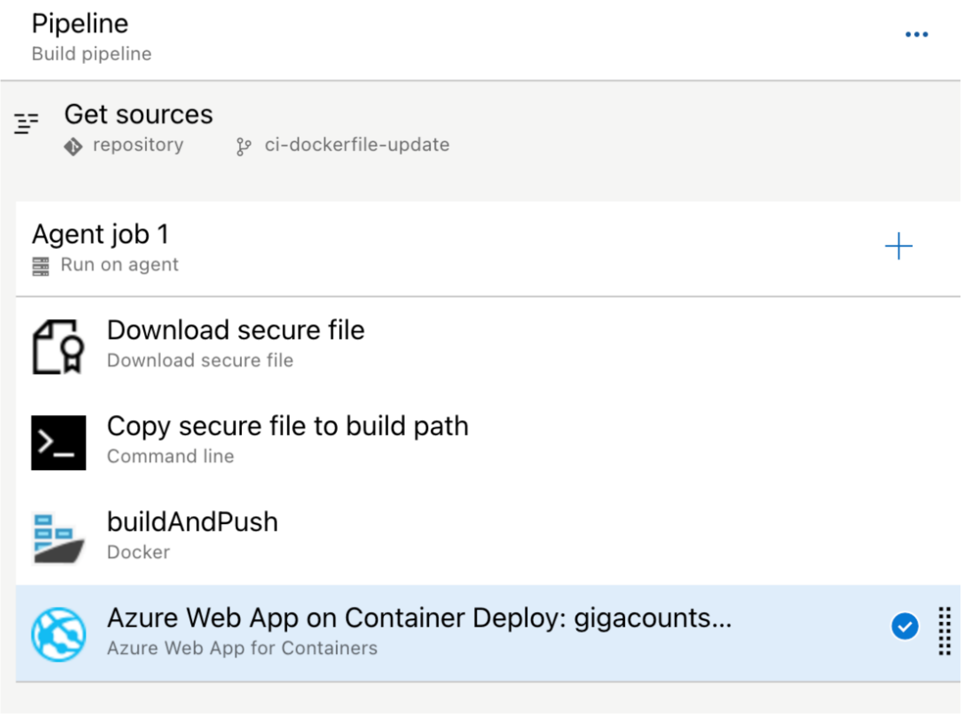
180

Branch filters

Type	Branch specification
Include	refs/heads/develop

+ Add

Result


Pipeline
Build pipeline

Get sources
repository ci-dockerfile-update

Agent job 1
Run on agent

Download secure file
Download secure file

Copy secure file to build path
Command line

buildAndPush
Docker

Azure Web App on Container Deploy: gigacounts...
Azure Web App for Containers

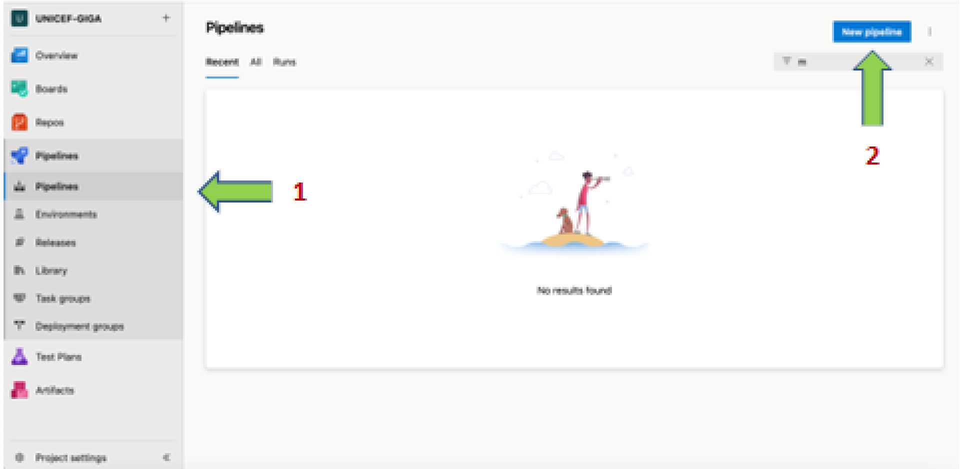
How to deploy GIGA – Frontend with Azure Pipeline

Requirements

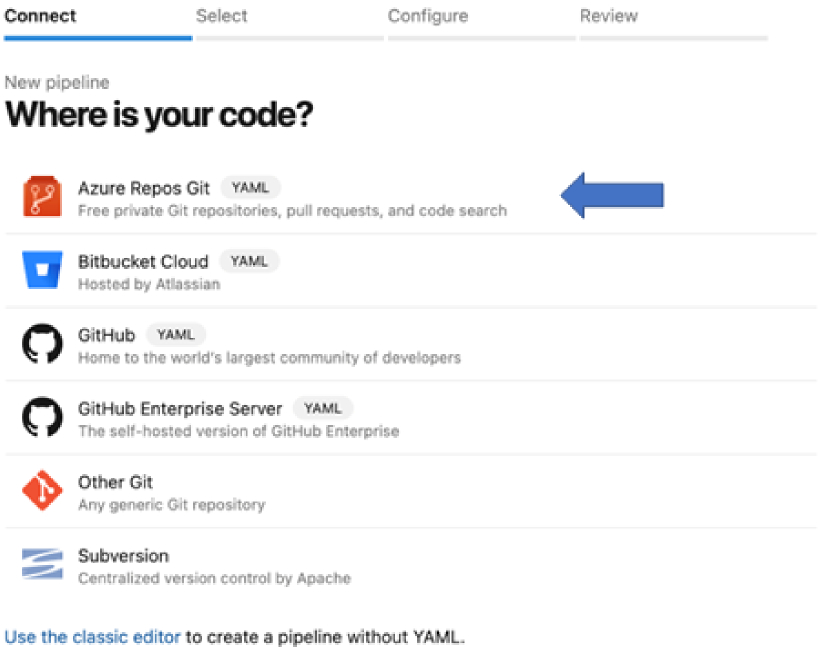
Before to create a pipeline, we need to create a service connection to Azure subscription and had “Build Administrator” permissions in Az DevOps.

Steps for deployment

1. Create a pipeline from GiGA-Frontend repository



2. Select “Azure Repos Git” option and select GIGA-FRONT repository. If no exists, create from Azure Repos option. Then select “Use the classic editor” option.



New pipeline

Where is your code?

Azure Repos Git YAML
Free private Git repositories, pull requests, and code search

Bitbucket Cloud YAML
Hosted by Atlassian

GitHub YAML
Home to the world's largest community of developers

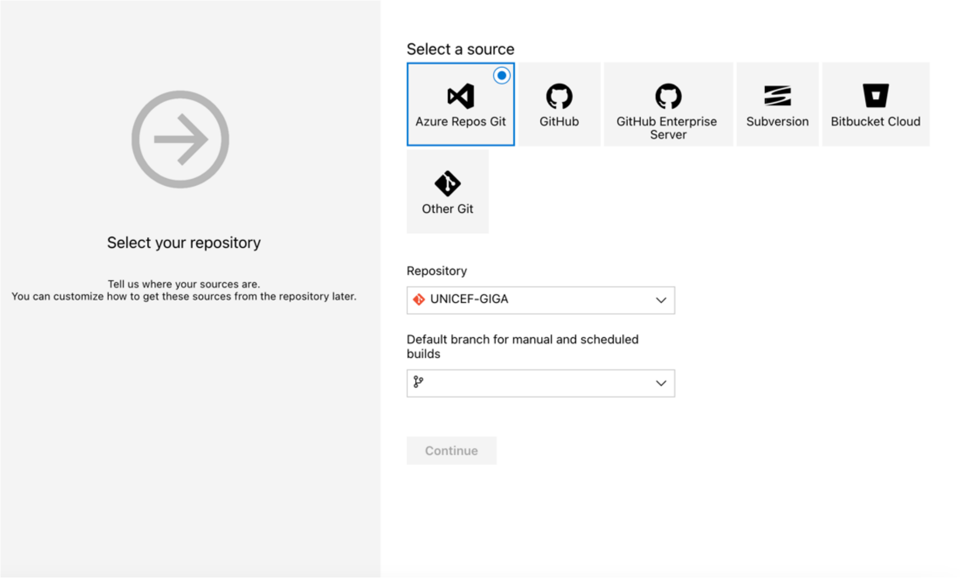
GitHub Enterprise Server YAML
The self-hosted version of GitHub Enterprise

Other Git
Any generic Git repository

Subversion
Centralized version control by Apache

Use the classic editor to create a pipeline without YAML.

3. Now select “Azure Repos Git” option and GIGA-Frontend repo with the develop branch. Then press “Continue” button and press “Empty Job” as template



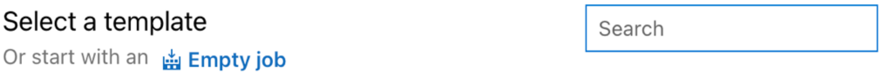
Select your repository

Tell us where your sources are.
You can customize how to get these sources from the repository later.

Select a source

Repository: UNICEF-GIGA

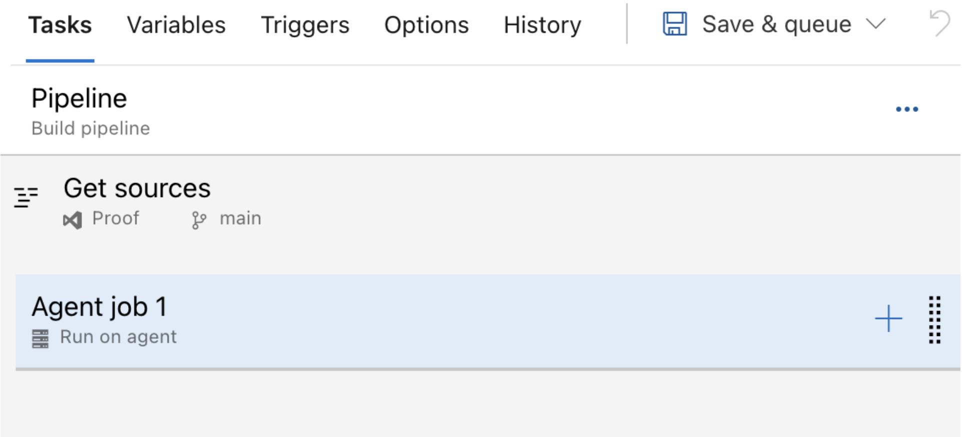
Default branch for manual and scheduled builds



Select a template

Or start with an Empty job

After press the empty job option, you can view the following screen: ￼



Pipeline
Build pipeline

Get sources
Proof main

Agent job 1
Run on agent

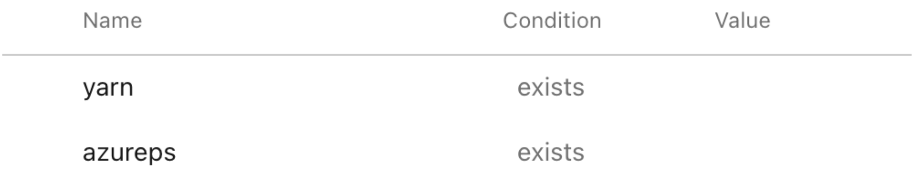
4. For configure “Agent Job” set the following properties:

   Display name: Agent Job Giga Frontend

   Agent pool: Azure Pipelines

   Agent Specification: windows-2022

   Demands:



Name	Condition	Value
yarn	exists	
azureps	exists	

*Rest of properties leave it in default

5. Add “Node JS tools installer” task and configure with the following properties:

   Display name: Use Node 16.x

   Version Spec: 16.14.0

   *Rest of properties leave it in default

6. Add “NPM” task and configure with the following properties:

   Display name: npm install

   Command: install

   *Rest of properties leave it in default

7. Add other “NPM” task and configure with the following properties:

   Display name: npm run lint:fix

   Command: custom

   Command and arguments: run lint:fix

   *Rest of properties leave it in default

8. Add “Yarn” task and configure with the following properties:

   Display name: Yarn build

   Arguments: build

   *Rest of properties leave it in default

9. Add “Archive file copy” task and configure with the following properties:

   Task version: 5.*

   Display name: AzureBlob File Copy

   Source: $(System.DefaultWorkingDirectory)/build/*

   Azure Subscription: <Select your subscription>

   Destination Type: Azure Blob

   RM Storage Account: <Select your storage account>

   Container Name: $web

   SAS Token Expiration Period In Minutes: 240

*Rest of properties leave it in default

10. Now, add “Publish build artifacts” task and configure with the following properties:

    Task version: 1.*

    Display name: Publish artifacts: drop

    Path to publish: $(Build.ArtifactStagingDirectory)

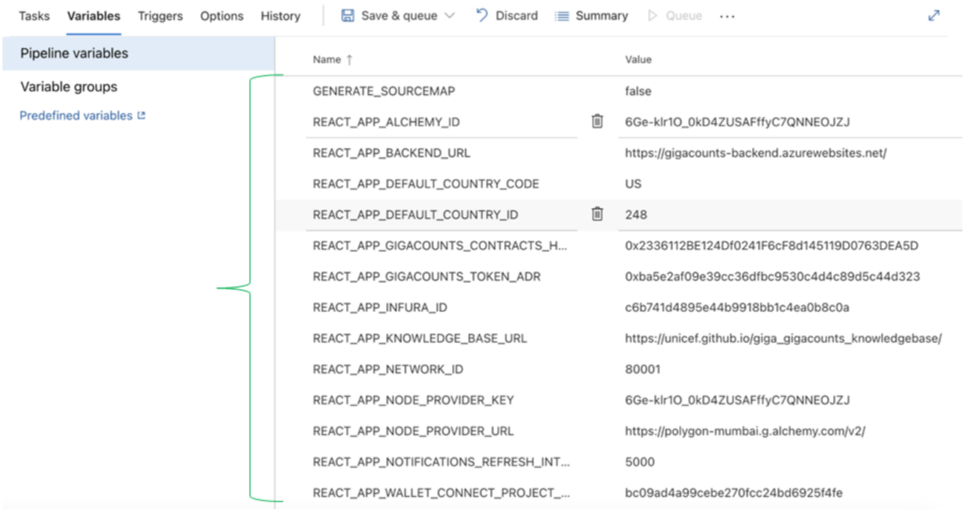
    Artifact name: drop

    Artifact publish location: Azure Pipelines

    Max Artifact Size: 0

    *Rest of properties leave it in default

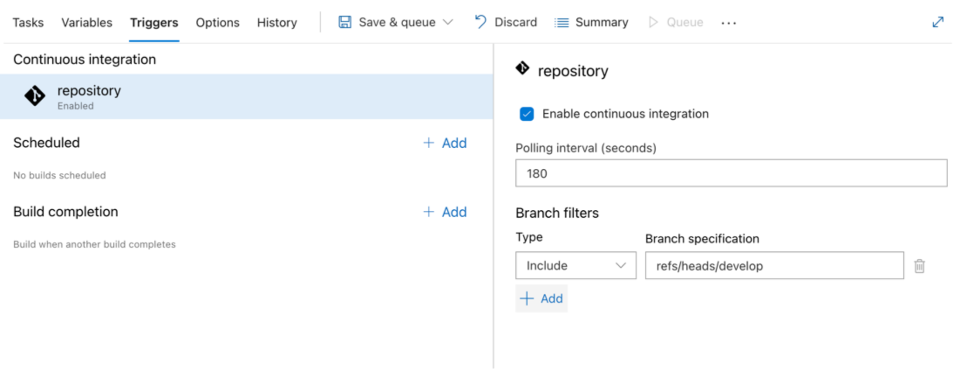
11. Enter to "Variables” tab and set de following “pipeline variables”



Name ↑	Value
GENERATE_SOURCEMAP	false
REACT_APP_ALCHEMY_ID	6Ge-klr1O_0kD4ZUSAFffyC7QNNEOJZJ
REACT_APP_BACKEND_URL	https://gigacounts-backend.azurewebsites.net/
REACT_APP_DEFAULT_COUNTRY_CODE	US
REACT_APP_DEFAULT_COUNTRY_ID	248
REACT_APP_GIGACOUNTS_CONTRACTS_H...	0x2336112BE124Df0241F6cF8d145119D0763DEA5D
REACT_APP_GIGACOUNTS_TOKEN_ADR	0xba5e2af09e39cc36dfbc9530c4d4c89d5c44d323
REACT_APP_INFURA_ID	c6b741d4895e44b9918bb1c4ea0b8c0a
REACT_APP_KNOWLEDGE_BASE_URL	https://unicef.github.io/giga_gigacounts_knowledgebase/
REACT_APP_NETWORK_ID	80001
REACT_APP_NODE_PROVIDER_KEY	6Ge-klr1O_0kD4ZUSAFffyC7QNNEOJZJ
REACT_APP_NODE_PROVIDER_URL	https://polygon-mumbai.g.alchemy.com/v2/
REACT_APP_NOTIFICATIONS_REFRESH_INT...	5000
REACT_APP_WALLET_CONNECT_PROJECT_...	bc09ad4a99cebe270fcc24bd6925f4fe

```
SKIP_PREFLIGHT_CHECK=true
GENERATE_SOURCEMAP=false
npm_config_user_agent=yarn
PORT=3000
REACT_APP_BACKEND_URL=http://127.0.0.1:3333
REACT_APP_DEFAULT_COUNTRY_CODE=US
REACT_APP_NOTIFICATIONS_REFRESH_INTERVAL_MS=5000
REACT_APP_KNOWLEDGE_BASE_URL=
REACT_APP_WALLET_CONNECT_PROJECT_ID=
REACT_APP_NETWORK_ID=80001
REACT_APP_NODE_PROVIDER_URL=https://polygon-mumbai.g.alchemy.com/v2/
REACT_APP_NODE_PROVIDER_KEY=
REACT_APP_GIGACOUNTS_TOKEN_ADR=
REACT_APP_GIGACOUNTS_CONTRACTS_HANDLER_ADR=
REACT_APP_DRAFT_ID_OFFSET=9000
REACT_APP_B2C_SIGNUP_SIGNIN_NAME=
REACT_APP_B2C_FORGOT_PASSWORD_NAME=
REACT_APP_B2C_EDIT_PROFILE_NAME=
REACT_APP_B2C_URL=
REACT_APP_B2C_DOMAIN=
REACT_APP_B2C_CLIENT_ID=
```

12. Finally enter to "Triggers” tab and set de following configuration:



Continuous integration

repository
Enabled

Scheduled + Add

No builds scheduled

Build completion + Add

Build when another build completes

repository

☑ Enable continuous integration

Polling interval (seconds)

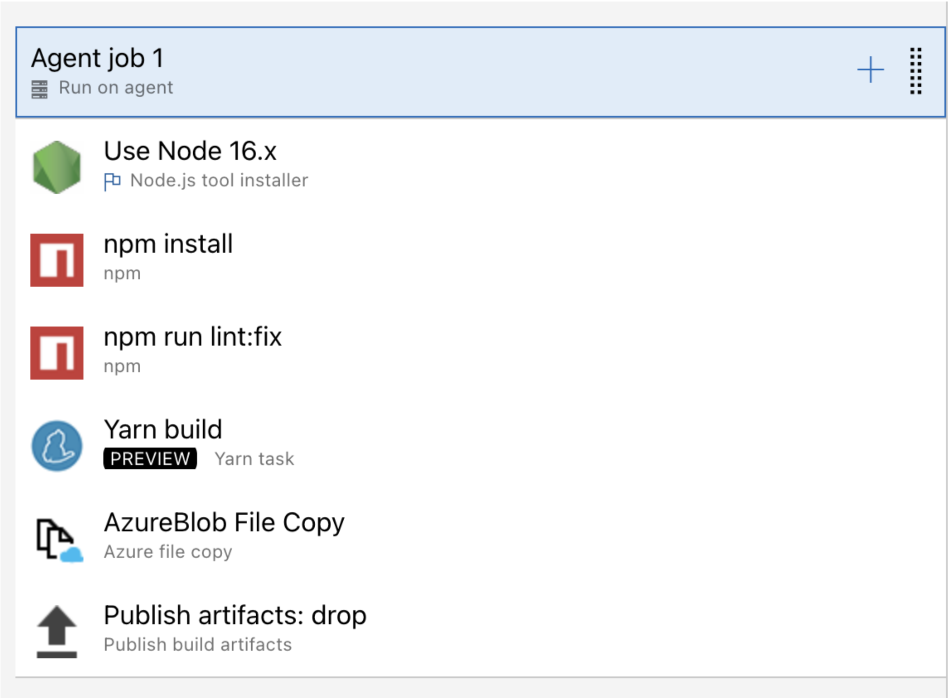
180

Branch filters

Type	Branch specification
Include	refs/heads/develop

+ Add

Result


Agent job 1
Run on agent

Use Node 16.x
Node.js tool installer

npm install
npm

npm run lint:fix
npm

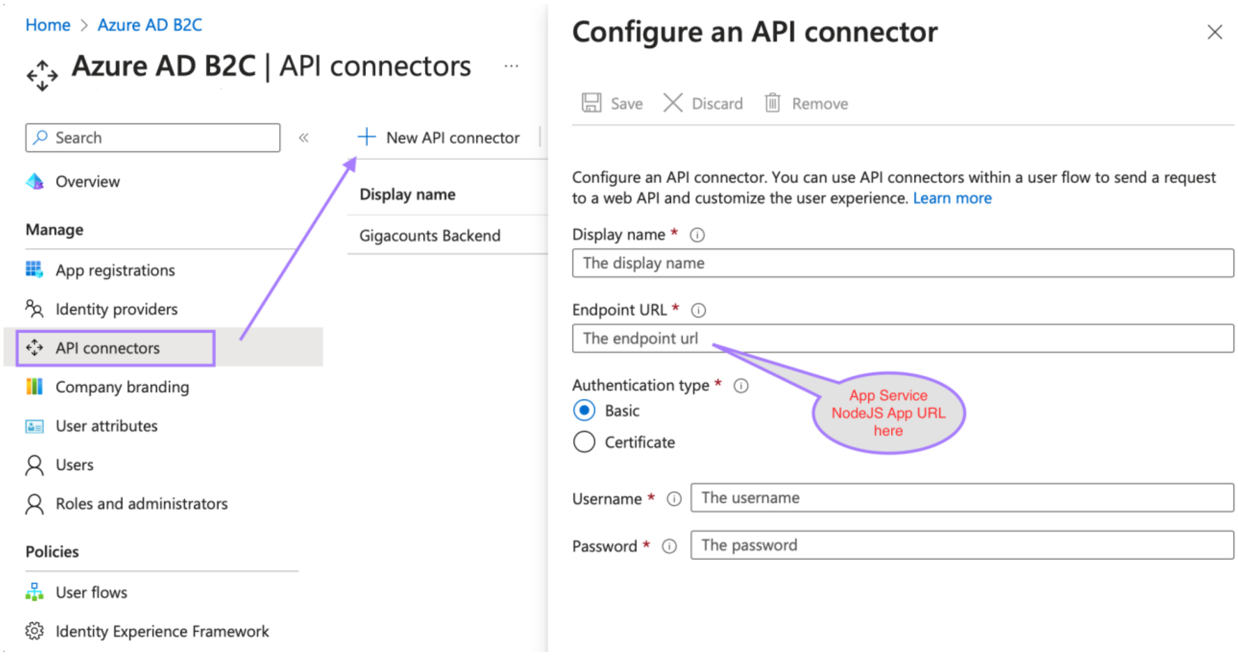
Yarn build
PREVIEW Yarn task

AzureBlob File Copy
Azure file copy

Publish artifacts: drop
Publish build artifacts

Azure AD B2C Instructions

1. Create a new API Connector that aims to Backend application endpoint. The endpoint will be shared later.



Home > Azure AD B2C

Azure AD B2C | API connectors

Search

Overview

Manage

- App registrations
- Identity providers
- API connectors
- Company branding
- User attributes
- Users
- Roles and administrators

Policies

- User flows
- Identity Experience Framework

+ New API connector

Display name
Gigacounts Backend

Configure an API connector

Save Discard Remove

Configure an API connector. You can use API connectors within a user flow to send a request to a web API and customize the user experience. Learn more

Display name * : The display name

Endpoint URL * : The endpoint url

Authentication type *

● Basic

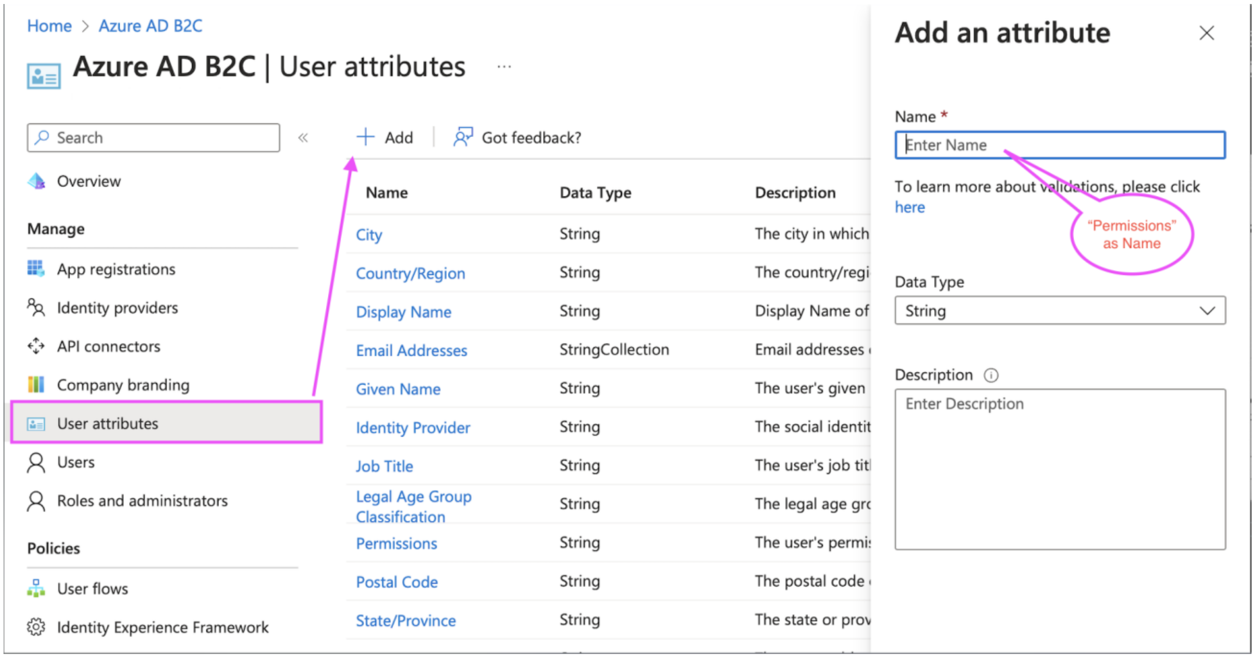
○ Certificate

Username * : The username

Password * : The password

App Service NodeJS App URL here

2. Add a new attribute called **Permissions** of type String.



Home > Azure AD B2C

Azure AD B2C | User attributes

Search

Overview

Manage

- App registrations
- Identity providers
- API connectors
- Company branding
- User attributes
- Users
- Roles and administrators

Policies

- User flows
- Identity Experience Framework

+ Add Got feedback?

Name	Data Type	Description
City	String	The city in which
Country/Region	String	The country/regi
Display Name	String	Display Name of
Email Addresses	StringCollection	Email addresses
Given Name	String	The user's given
Identity Provider	String	The social identit
Job Title	String	The user's job tit
Legal Age Group Classification	String	The legal age gr
Permissions	String	The user's permis
Postal Code	String	The postal code
State/Province	String	The state or prov

Add an attribute

Name * : Enter Name

To learn more about validations, please click here

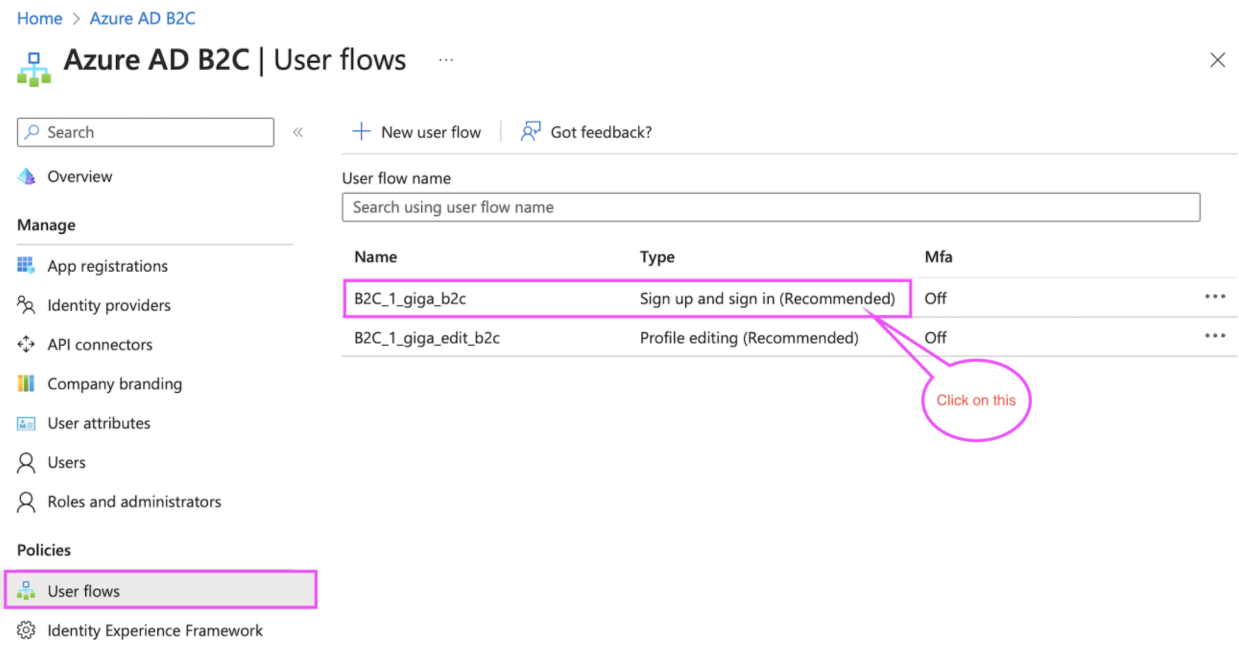
Data Type: String

Description: Enter Description

“Permissions” as Name

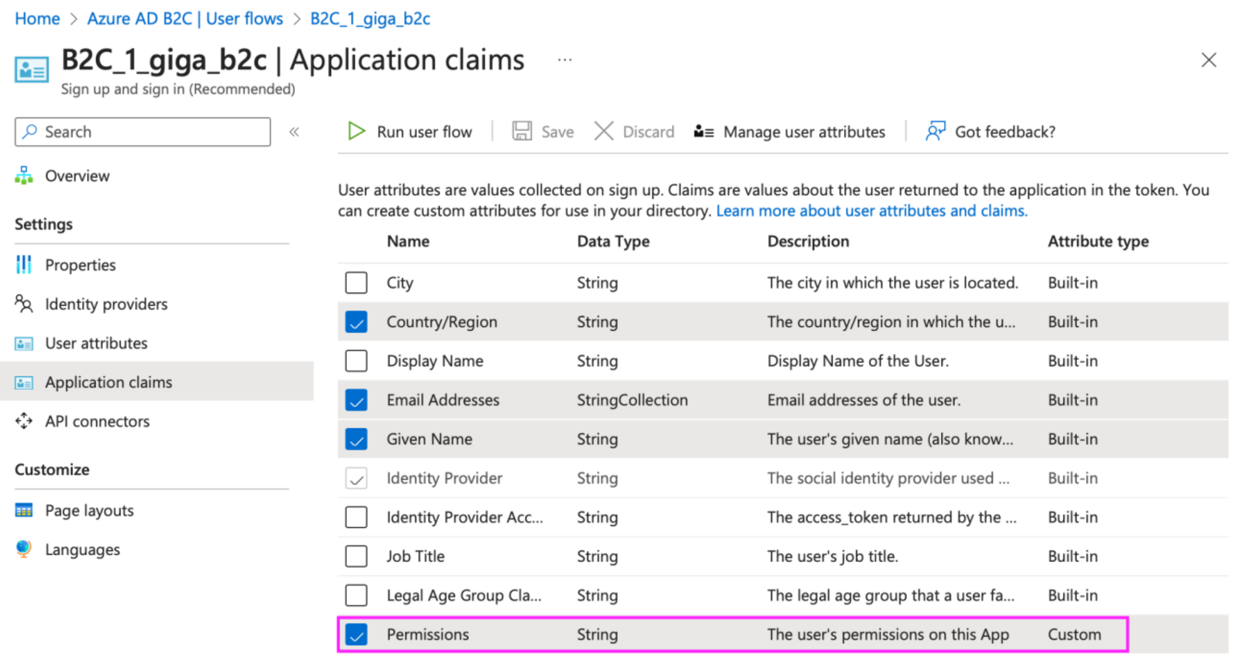
2.1. Repeat the same process for two additional attributes: **UserId** of type Int and **CountryId** of type Int.

3. Modify the existing sign-in user flow. Go to User Flows on left sidebar and click on default *Sign up and Sign in* flow.



Name	Type	Mfa
B2C_1_giga_b2c	Sign up and sign in (Recommended)	Off
B2C_1_giga_edit_b2c	Profile editing (Recommended)	Off

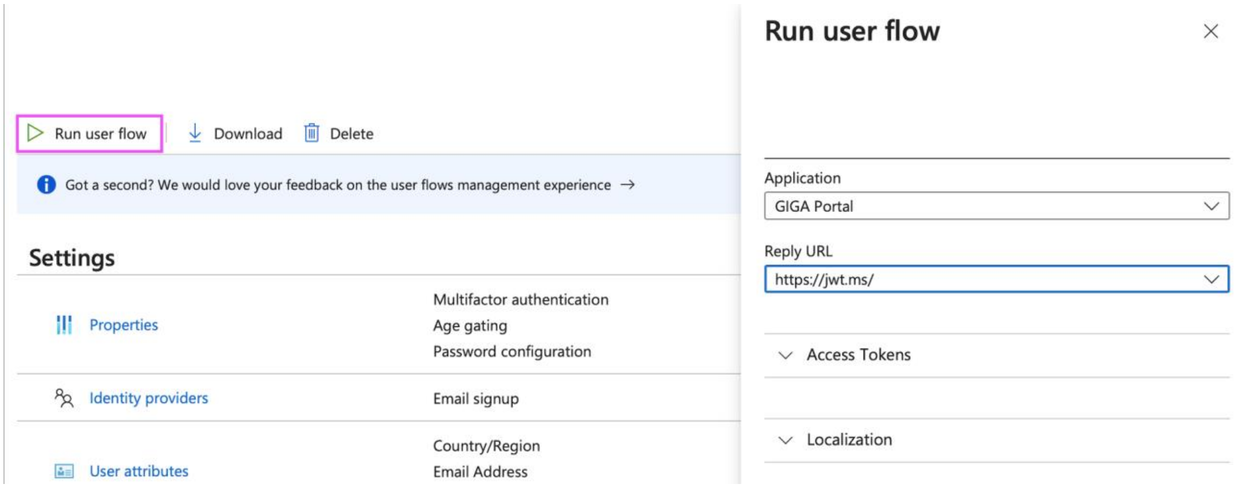
3.1. On the selected user flow, check the **Permissions** attribute, as well as **UserId** and **CountryId**.



User attributes are values collected on sign up. Claims are values about the user returned to the application in the token. You can create custom attributes for use in your directory. Learn more about user attributes and claims.

	Name	Data Type	Description	Attribute type
☐	City	String	The city in which the user is located.	Built-in
☑	Country/Region	String	The country/region in which the u...	Built-in
☐	Display Name	String	Display Name of the User.	Built-in
☑	Email Addresses	StringCollection	Email addresses of the user.	Built-in
☑	Given Name	String	The user's given name (also know...	Built-in
✓	Identity Provider	String	The social identity provider used ...	Built-in
☐	Identity Provider Acc...	String	The access_token returned by the ...	Built-in
☐	Job Title	String	The user's job title.	Built-in
☐	Legal Age Group Cla...	String	The legal age group that a user fa...	Built-in
☑	Permissions	String	The user's permissions on this App	Custom

Once created, please share with us the link generated by the **Run user Flow** option.



Got a second? We would love your feedback on the user flows management experience →

Settings

Properties	Multifactor authentication Age gating Password configuration
Identity providers	Email signup
User attributes	Country/Region Email Address

Application: GIGA Portal

Reply URL: https://jwt.ms/

User Creation on Azure AD B2C

Create these users with the some password: e.g: p1ssw0rd#198341# and these users:

"admin@giga.com" "GIGA Super Admin"
"view.only@giga.com" "GIGA View-only"
"isp.cm1.brazil@giga.com" "ISP Contract Manager"
"isp.cm2.brazil@giga.com" "ISP Contract Manager"
"isp.cm1.botswana@giga.com" "ISP Contract Manager"
"isp.cm2.botswana@giga.com" "ISP Contract Manager"
"isp.sa1.brazil@giga.com" "ISP Customer Service Agent"
"isp.sa2.brazil@giga.com" "ISP Customer Service Agent"
"isp.sa1.botswana@giga.com" "ISP Customer Service Agent"
"isp.sa2.botswana@giga.com" "ISP Customer Service Agent"
"cc1.brazil@giga.com" "Country Contract Creator"
"cc2.brazil@giga.com" "Country Contract Creator"
"cc1.botswana@giga.com" "Country Contract Creator"
"cc2.botswana@giga.com" "Country Contract Creator"
"ca1.brazil@giga.com" "Country Accountant"
"ca2.brazil@giga.com" "Country Accountant"
"ca1.botswana@giga.com" "Country Accountant"
"ca2.botswana@giga.com" "Country Accountant"
"admin1.brazil@giga.com" "Country Super Admin"
"admin1.botswana@giga.com" "Country Super Admin"
"monitor1.brazil@giga.com" "Country Monitor"
"monitor1.botswana@giga.com" "Country Monitor"
"sch.manager1.brazil@giga.com" "School Connectivity Manager"
"sch.manager2.brazil@giga.com" "School Connectivity Manager"
"sch.manager3.brazil@giga.com" "School Connectivity Manager"
"sch.manager1.botswana@giga.com" "School Connectivity Manager"
"sch.manager2.botswana@giga.com" "School Connectivity Manager"
"sch.manager3.botswana@giga.com" "School Connectivity Manager"