Exercise 4. Securing a web application with single sign-on (optional)

Estimated time

00:45

Overview

In this exercise, you secure an application by using the App ID service for single sign-on by authenticating your application through trusted server providers.

Objectives

After completing this exercise, you should be able to:

- Create an App ID service.
- Bind the App ID service to an application to add single sign-on capability to the application.
- Describe different configurations in the App ID service.

Introduction

In all web and mobile applications, security is important to protect all your sensitive data. You can use App ID to add authentication to your mobile and web apps and protect your APIs and back ends running on IBM Cloud. You also can use it to add an email and password-based sign-up and sign-in process, multi-factor authentication (MFA) with the App ID scalable user registry Cloud Directory, or a social media login, with Google or Facebook.

For employee apps, App ID uses SAML 2.0 federation so that users can sign in by using their enterprise credentials.

For all app users, you can enrich their profiles with additional information so that you can build engaging experiences. For example, if you know the age of the user, you can filter content in your application, or if you know the gender of the user, you can customize the application UI according to their preferences.

In this exercise, you create an App ID service, learn about different configurations, and bind the service to a Node.js application to explore the service capabilities.

Requirements

This exercise requires:

- Access to the internet and the latest version of a modern web browser, such as Chrome, Firefox, and Safari.
- IBM Cloud account.
- IBM Cloud CLI.
- A valid Google account.

Exercise instructions

In this exercise, you will complete the following tasks:

__ 1. Create an App ID service instance.

__ 2. Explore the App ID service.

__ 3. Retrieve the App ID service credentials.

__ 4. Create a Node.js app and connect it to the App ID service.

__ 5. Configure the application to integrate with the App ID service.

__ 6. Secure the sample application with the App ID service.

__ 7. Clean up the environment.

Part 1: Creating an App ID service instance

In this part, you create an App ID service to learn about its security capabilities and how to increase user interaction capabilities. Integrating an App ID service into your application can secure resources and add authentication even if you do not have a security background.

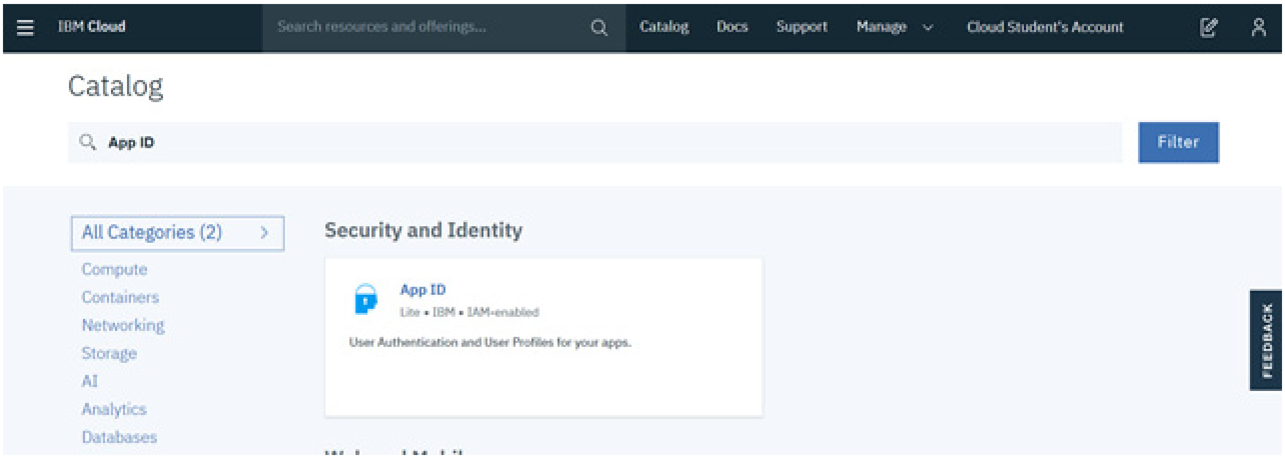
Complete the following steps:

__ 1. Log in to IBM Cloud at https://cloud.ibm.com/login.

__ 2. Click **Catalog**.

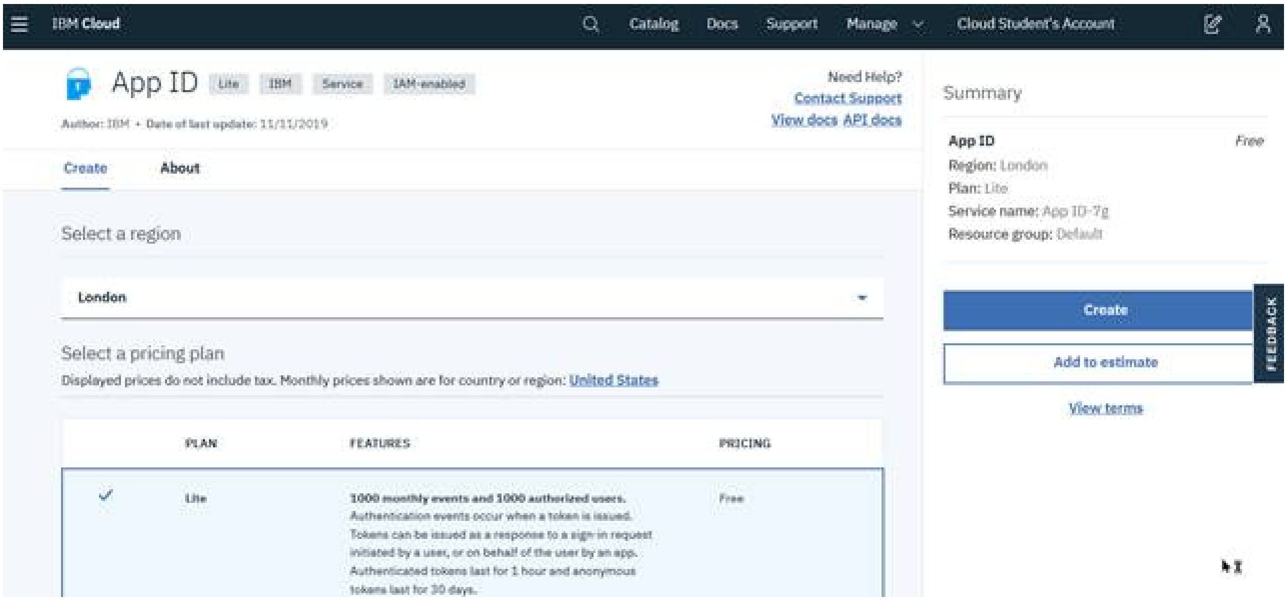
__ 3. In the search field, enter “App ID”.

__ 4. The App ID service is listed, as shown in the following figure.



__ 5. Click the **App ID** service under Security and Identity.

__ 6. Create an App ID service instance by accepting the default values and then click **Create**, as shown in the following figure.



Author: IBM • Date of last update: 11/11/2019

Select a region

London

Select a pricing plan

Displayed prices do not include tax. Monthly prices shown are for country or region: United States

	PLAN	FEATURES	PRICING
✓	Lite	**1000 monthly events and 1000 authorized users.** Authentication events occur when a token is issued. Tokens can be issued as a response to a sign-in request initiated by a user, or on behalf of the user by an app. Authenticated tokens last for 1 hour and anonymous tokens last for 30 days.	Free

Summary

App ID Free
Region: London
Plan: Lite
Service name: App ID-7g
Resource group: Default

Create

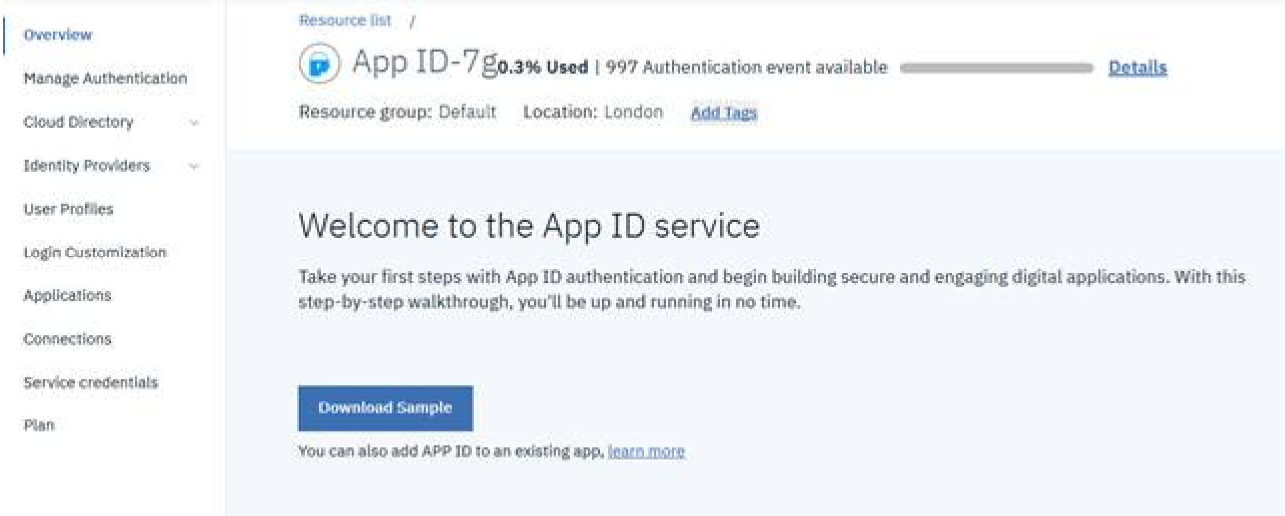
Add to estimate

View terms

Note

Save the service name and region for later use.

__ 7. After you click **Create**, you are redirected to the App Service Overview window, as shown in the following figure.



App ID-7g**0.3% Used** | 997 Authentication event available

Resource group: Default Location: London

Welcome to the App ID service

Take your first steps with App authentication and begin building secure and engaging digital applications. With this step-by-step walkthrough, you'll be up and running in no time.

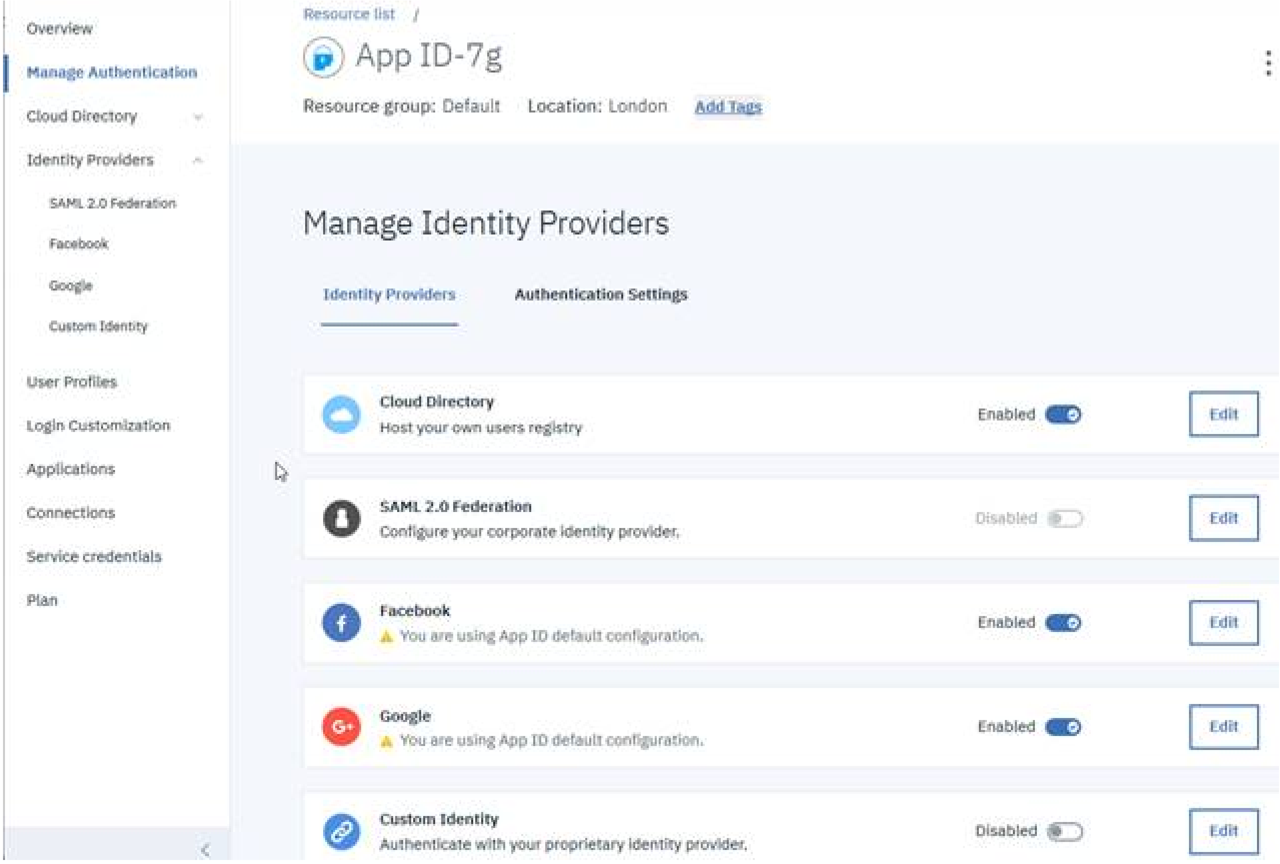
You can also add APP ID to an existing app, learn more

Part 2: Exploring the App ID service

In this part, you discover more about App ID configuration terms and concepts, such as identity providers and user profiles. You must configure them when you use the service to secure your application.

Complete the following steps:

__ 1. In the side bar menu, click **Manage Authentication** to configure identity providers. Two tabs, which are called Identity Providers and Authentication Settings, are displayed, as shown in the following figure.



App ID-7g

Resource group: Default Location: London

Manage Identity Providers

Cloud Directory — Host your own users registry — Enabled

SAML 2.0 Federation — Configure your corporate identity provider. — Disabled

Facebook — ⚠ You are using App ID default configuration. — Enabled

Google — ⚠ You are using App ID default configuration. — Enabled

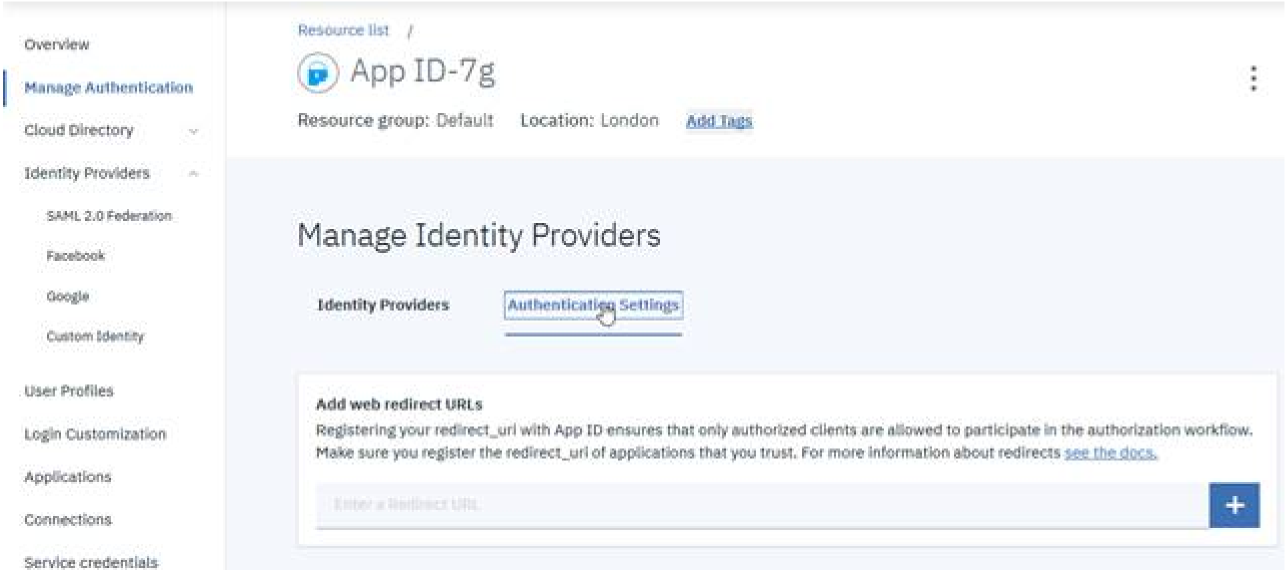
Custom Identity — Authenticate with your proprietary identity provider. — Disabled

An identity provider creates and manages information about an entity, such as a user, a functional ID, or an application. The provider verifies the identity of the entity by using credentials, such as a password. Then, the identity provider sends the identity information to another service provider. Because the identity provider authenticates the entity, the App ID can authorize it and grant access to your apps.

Note

Make sure that you have a Google account because you are going to use it in this exercise. If you do not have a Google account, use Facebook instead.

__ 2. Click the **Authentication Settings** tab. You can add your application URLs in the **Add web redirect URLs** field to “whitelist” your application to redirect to the App ID service. See the following figure. You add your application URL later in this exercise.



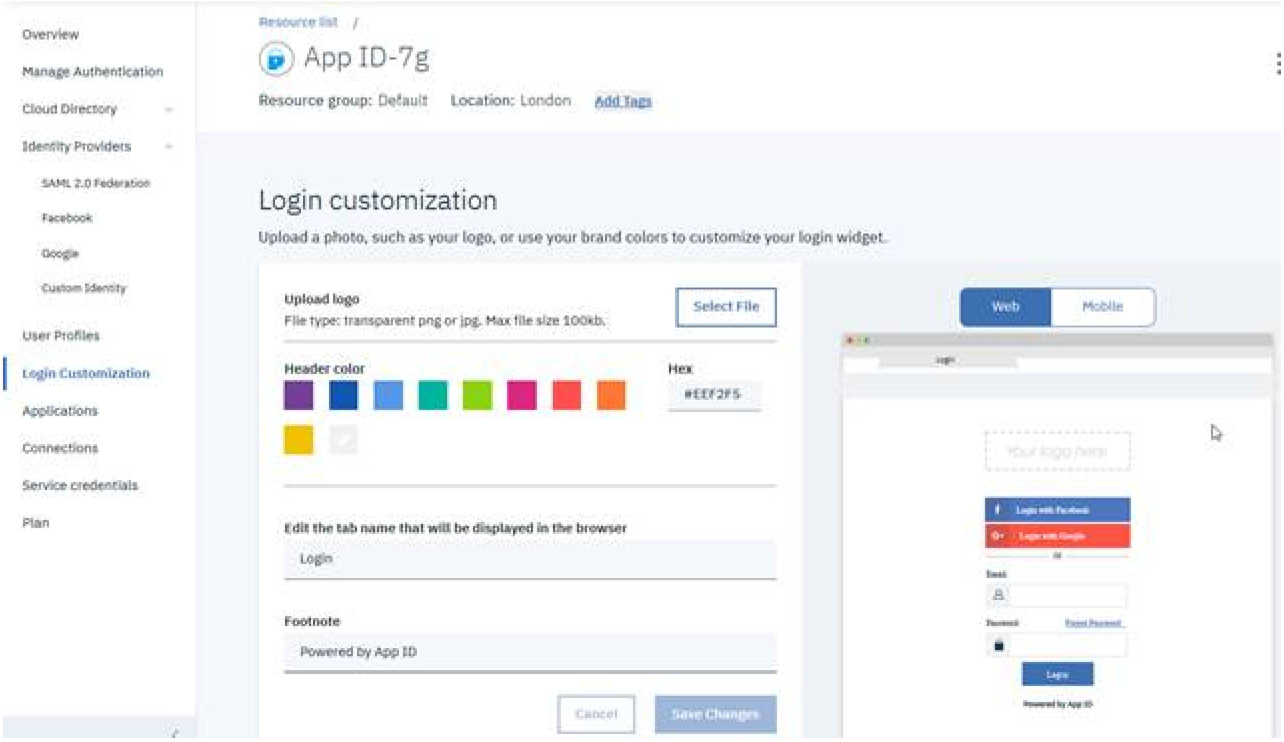
Registering your redirect_url with App ID ensures that only authorized clients are allowed to participate in the authorization workflow. Make sure you register the redirect_url of applications that you trust. For more information about redirects see the docs.

Information

A *redirect* *URL* is a technique that is used to send a user from one URL to another. For example, if the user enters the URL `website.com/page-a`, a redirect can cause the browser to open the web page `website.com/page-b`.

This function is useful when, for example, you move your website and want to shut down the old one. To prevent users from receiving a `404 Not Found` error, you redirect the old URLs to the new ones.

__ 3. App ID provides a login window that the user uses to log in to one of the configured identity providers. Click **Login Customization** to change the UI of the redirected login window, as shown in the following figure.



Overview
Manage Authentication
Cloud Directory
Identity Providers
SAML 2.0 Federation
Facebook
Google
Custom Identity
User Profiles
Login Customization
Applications
Connections
Service credentials
Plan

Resource list /

App ID-7g

Resource group: Default Location: London Add Tags

Login customization

Upload a photo, such as your logo, or use your brand colors to customize your login widget.

Upload logo
File type: transparent png or jpg. Max file size 100kb.
Select File

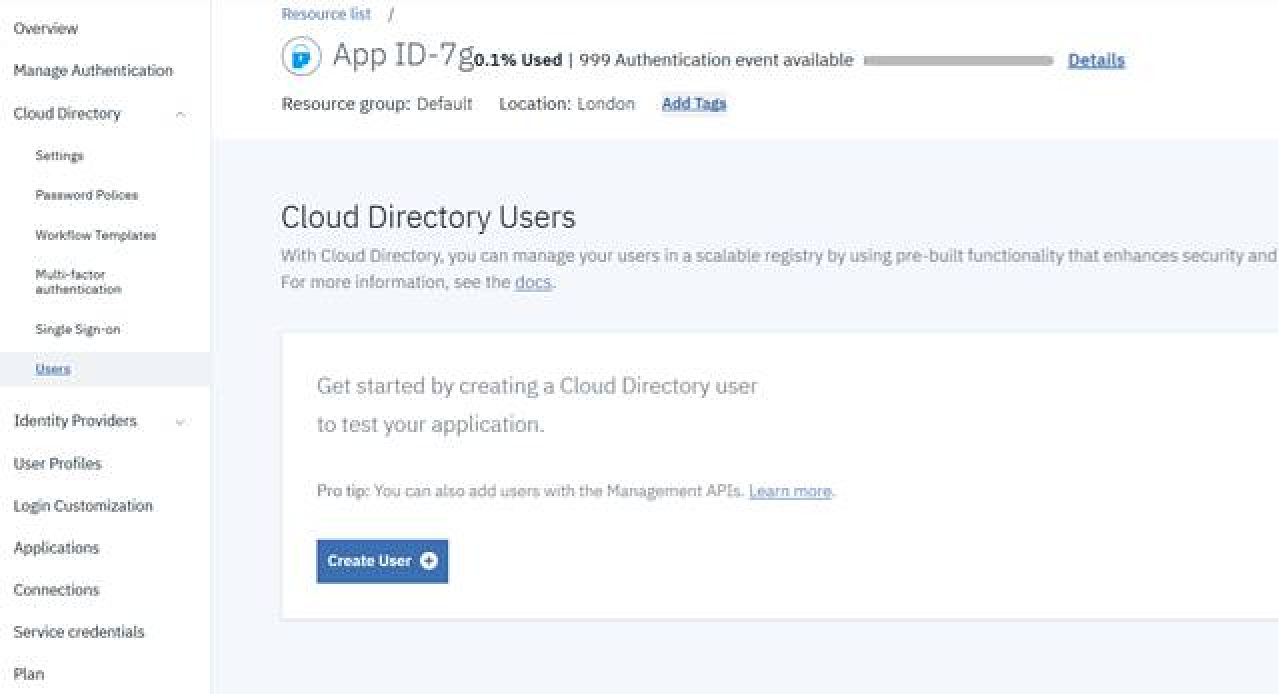
Header color
Hex
#EEF2F5

Edit the tab name that will be displayed in the browser
Login

Footnote
Powered by App ID

Cancel Save Changes

__ 4. Expand **Cloud Directory** and click **Users**. This tab is used to add Cloud Directory users that can log in by using a user name and password.



Overview
Manage Authentication
Cloud Directory
Settings
Password Polices
Workflow Templates
Multi-factor authentication
Single Sign-on
Users
Identity Providers
User Profiles
Login Customization
Applications
Connections
Service credentials
Plan

Resource list /

App ID-7g 0.1% Used | 999 Authentication event available Details

Resource group: Default Location: London Add Tags

Cloud Directory Users

With Cloud Directory, you can manage your users in a scalable registry by using pre-built functionality that enhances security and
For more information, see the docs.

Get started by creating a Cloud Directory user
to test your application.

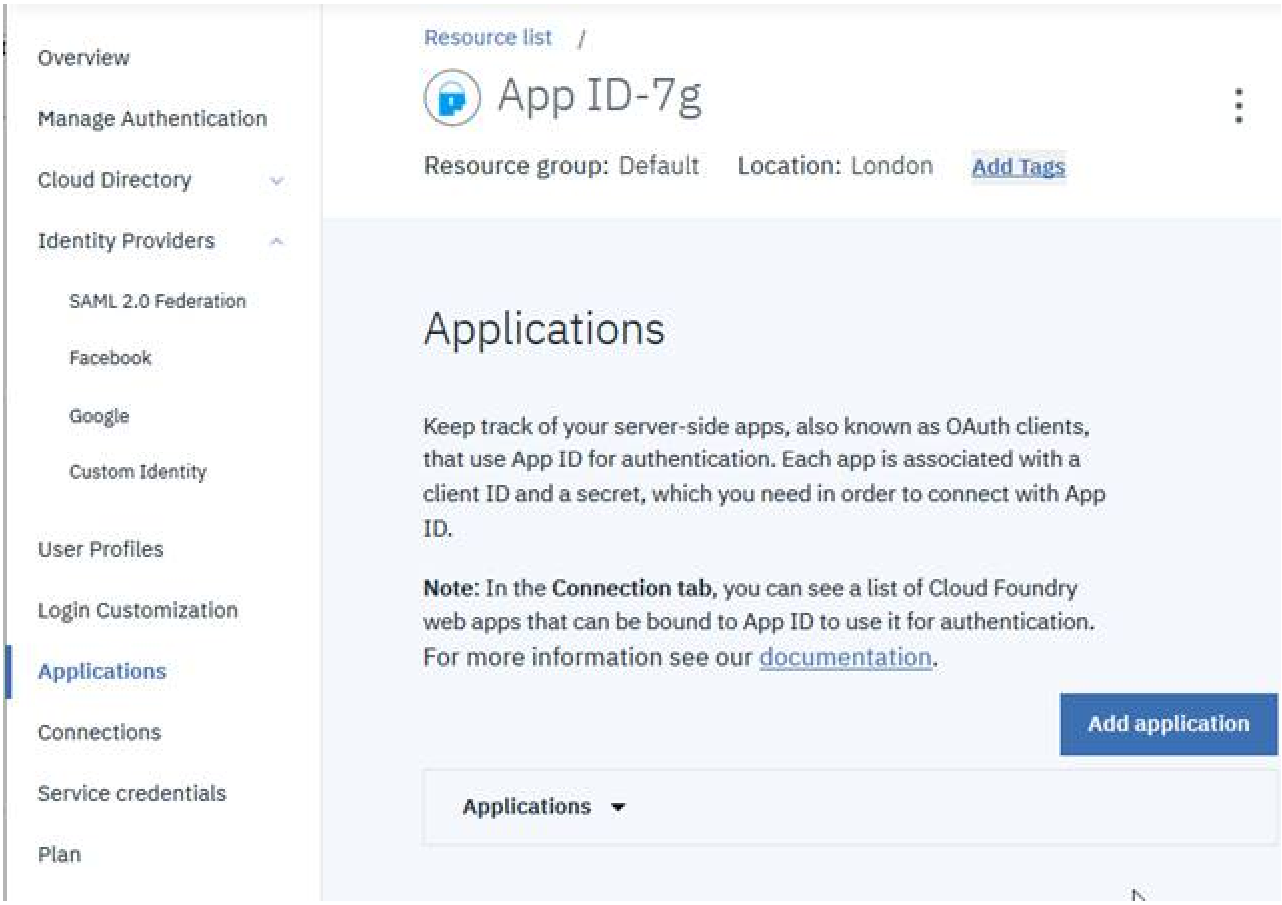
Pro tip: You can also add users with the Management APIs. Learn more.

Create User ⊕

Note

A cloud directory is a user registry that is maintained in the cloud. When users sign up for your app, they are added to your directory of users. The directory acts as another identity provider if you are not depending on Facebook, Google, or SAML for your logins.

__ 5. Click **Applications**. In this tab, you see all the Cloud Foundry applications that are connected to your service, as show in the following figure.



Applications

Keep track of your server-side apps, also known as OAuth clients, that use App ID for authentication. Each app is associated with a client ID and a secret, which you need in order to connect with App ID.

Note: In the **Connection tab**, you can see a list of Cloud Foundry web apps that can be bound to App ID to use it for authentication. For more information see our documentation.

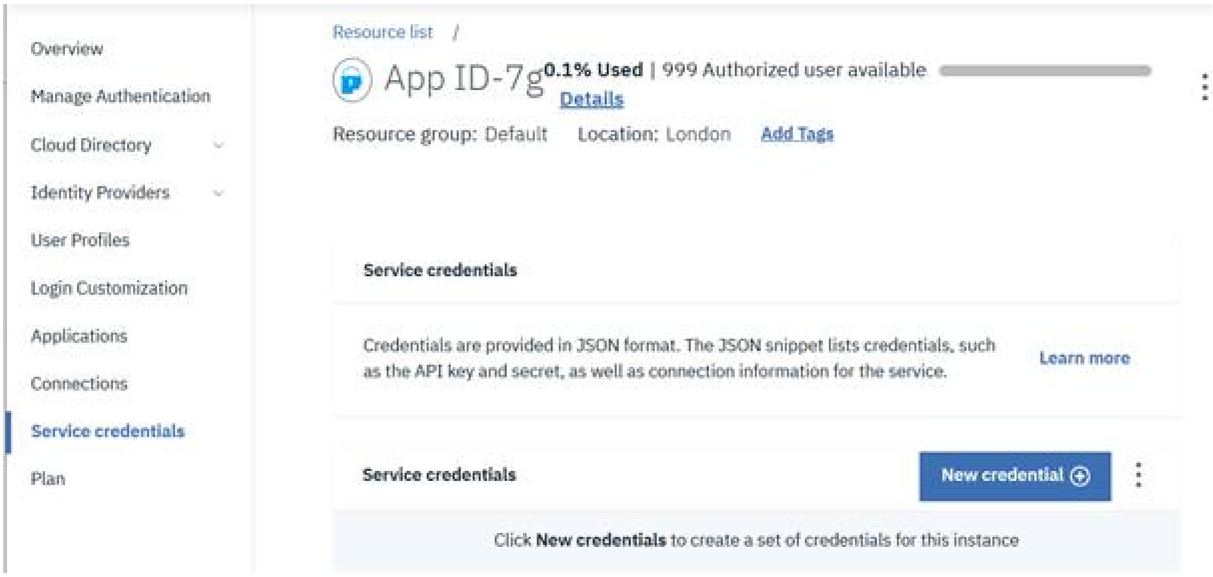
Part 3: Retrieving the App ID service credentials

In this part, you retrieve the App ID service credentials. You use this information to customize the sample app for integration with the App ID service.

Complete these steps:

__ 1. Return to the App ID dashboard and click **Service credentials.**

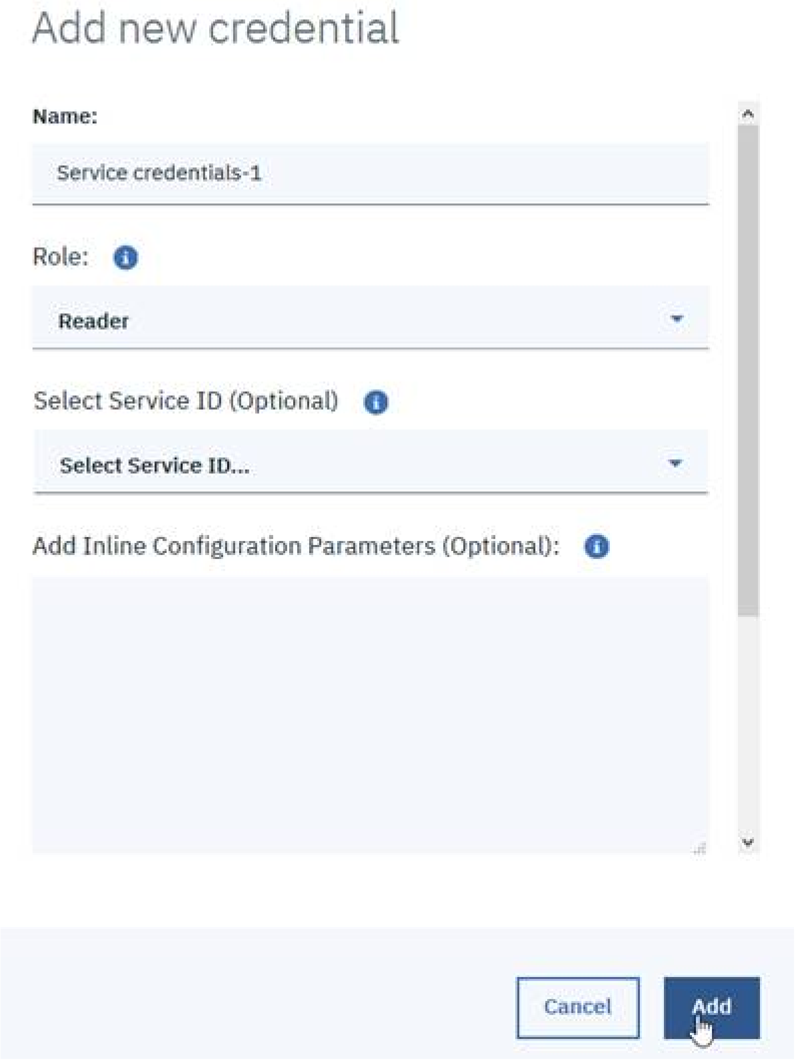
__ 2. If there are no service credentials, click **New credentials +**. If the service credentials exist, skip to View credentials.



Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service.

Click **New credentials** to create a set of credentials for this instance

__ 3. At the Add new credentials window, keep the default values, and click **Add**.


Add new credential

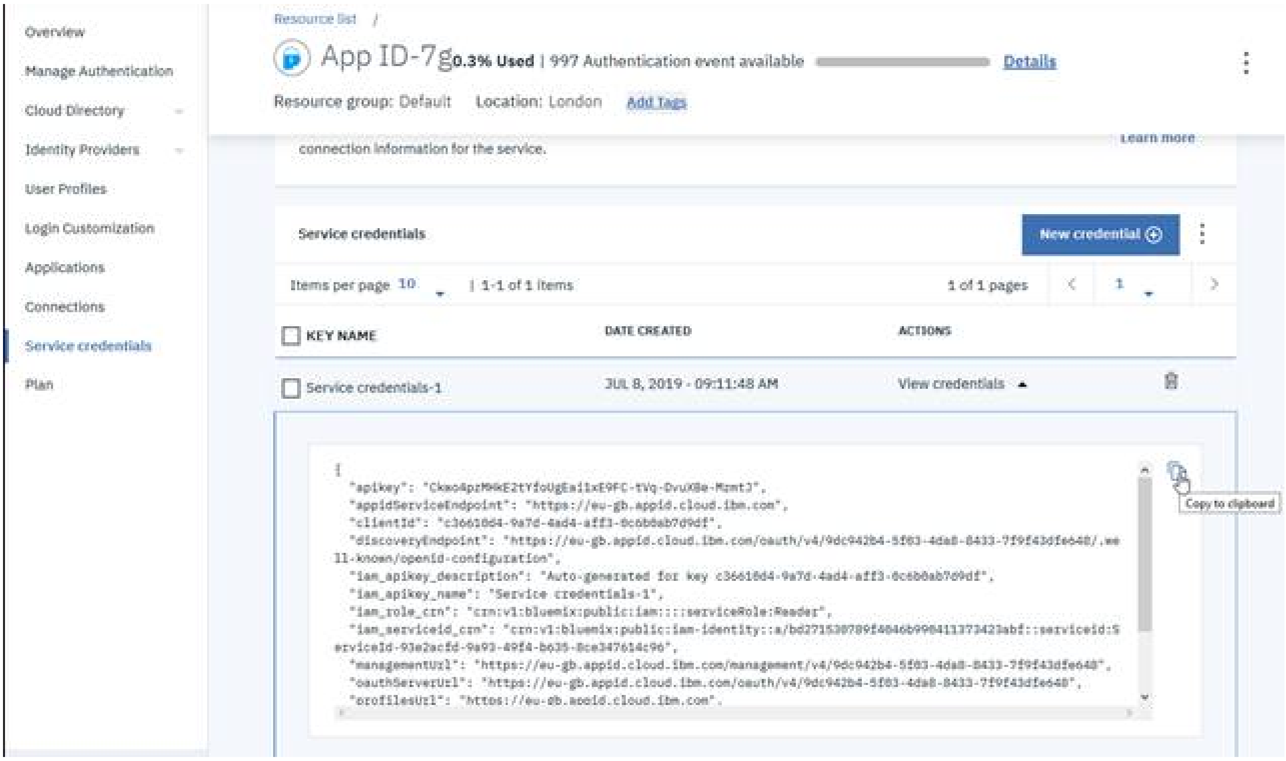
Name: Service credentials-1

Role: Reader

Select Service ID (Optional): Select Service ID...

Add Inline Configuration Parameters (Optional):

__ 4. Expand **View credentials** under ACTIONS. The App ID service credentials are displayed in JSON format, as shown in the following figure.



App ID-7g 0.3% Used | 997 Authentication event available

Resource group: Default Location: London

connection information for the service.

Service credentials

Items per page 10 | 1-1 of 1 items 1 of 1 pages 1

KEY NAME	DATE CREATED	ACTIONS
Service credentials-1	JUL 8, 2019 - 09:11:48 AM	View credentials

```
{
  "apikey": "Ckmo4przMHkE2tYfoUgEai1xE9FC-tVq-DvuX8e-Mzmt3",
  "appidServiceEndpoint": "https://eu-gb.appid.cloud.ibm.com",
  "clientId": "c36610d4-9a7d-4ad4-aff3-0c6b0ab7d9df",
  "discoveryEndpoint": "https://eu-gb.appid.cloud.ibm.com/oauth/v4/9dc942b4-5f03-4da8-8433-7f9f43dfe648/.we
ll-known/openid-configuration",
  "iam_apikey_description": "Auto-generated for key c36610d4-9a7d-4ad4-aff3-0c6b0ab7d9df",
  "iam_apikey_name": "Service credentials-1",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Reader",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam-identity::a/bd271530789f4046b99041137342 3abf::serviceid:S
erviceId-93e2acfd-9a93-49f4-b635-8ce347614c96",
  "managementUrl": "https://eu-gb.appid.cloud.ibm.com/management/v4/9dc942b4-5f03-4da8-8433-7f9f43dfe648",
  "oauthServerUrl": "https://eu-gb.appid.cloud.ibm.com/oauth/v4/9dc942b4-5f03-4da8-8433-7f9f43dfe648",
  "profilesUrl": "https://eu-gb.appid.cloud.ibm.com",
```

__ 5. Click the Copy to clipboard icon and paste the credentials in a text editor. You use this information to configure your application to integrate with the App ID service.

Part 4: Creating a Node.js app and connecting it to the App ID service

In this part, you create a Node.js app on IBM Cloud and connect it to the App ID service that you created in Part 1.

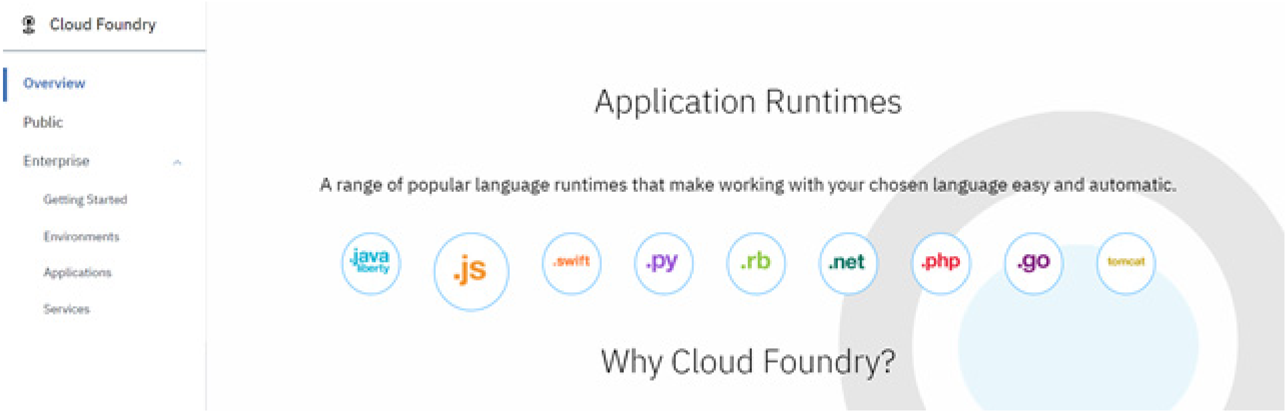
Creating a Node.js application on IBM Cloud

Create a Node.js application by performing the following steps:

__ 1. From the IBM Cloud catalog search for **Cloud Foundry**.

__ 2. Select the **Cloud Foundry** service.

__ 3. In the Cloud Foundry overview page, select **.js** under **Application Runtimes** to create a Node.js application as shown in the following figure.

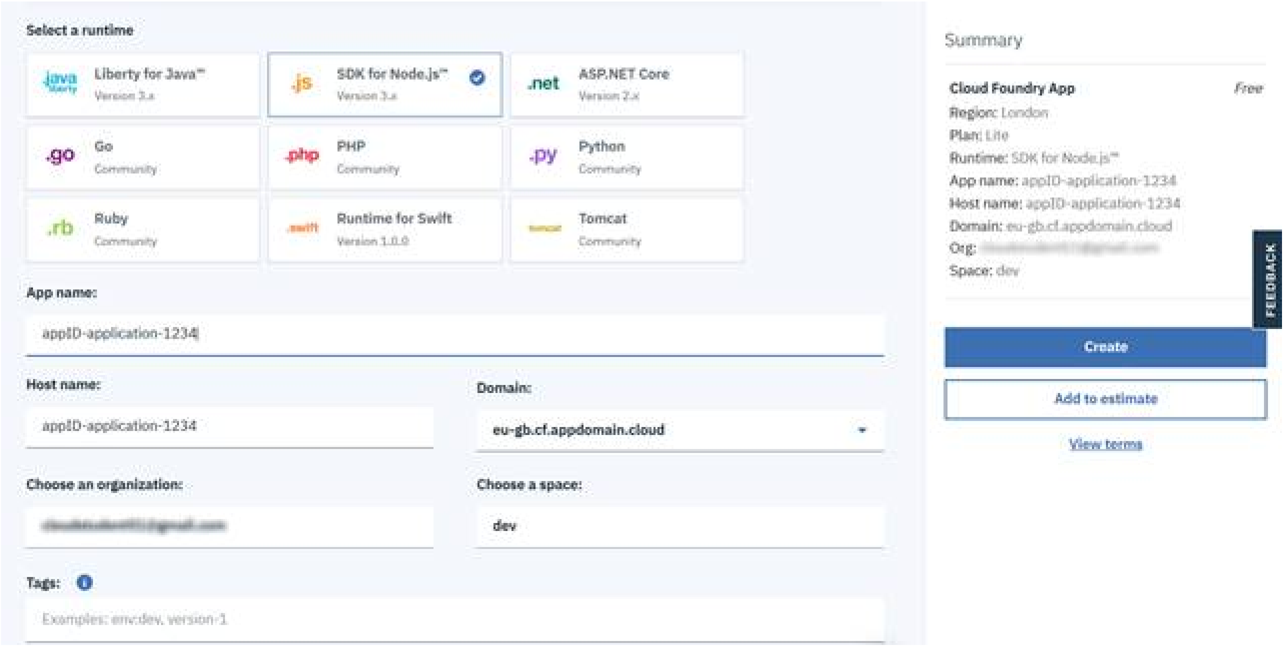


Application Runtimes

A range of popular language runtimes that make working with your chosen language easy and automatic.

Why Cloud Foundry?

__ 4. Enter a unique app name, for example, **appID-application-xxxxx** where “xxxxx” is a unique identifier for your application. Keep the default values for the other fields as shown in the following figure.



Select a runtime

App name: appID-application-1234

Host name: appID-application-1234

Domain: eu-gb.cf.appdomain.cloud

Choose an organization: [redacted]

Choose a space: dev

Tags: Examples: env:dev, version-1

Summary

Cloud Foundry App *Free*

Region: London
Plan: Lite
Runtime: SDK for Node.js™
App name: appID-application-1234
Host name: appID-application-1234
Domain: eu-gb.cf.appdomain.cloud
Org: [redacted]
Space: dev

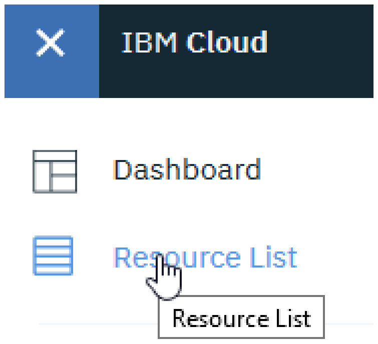
__ 5. Click **Create**.

Connecting the application to the App ID service

In this part, you connect the Node.js Cloud Foundry app that you created on IBM Cloud to the App ID service that you created in Part 1.

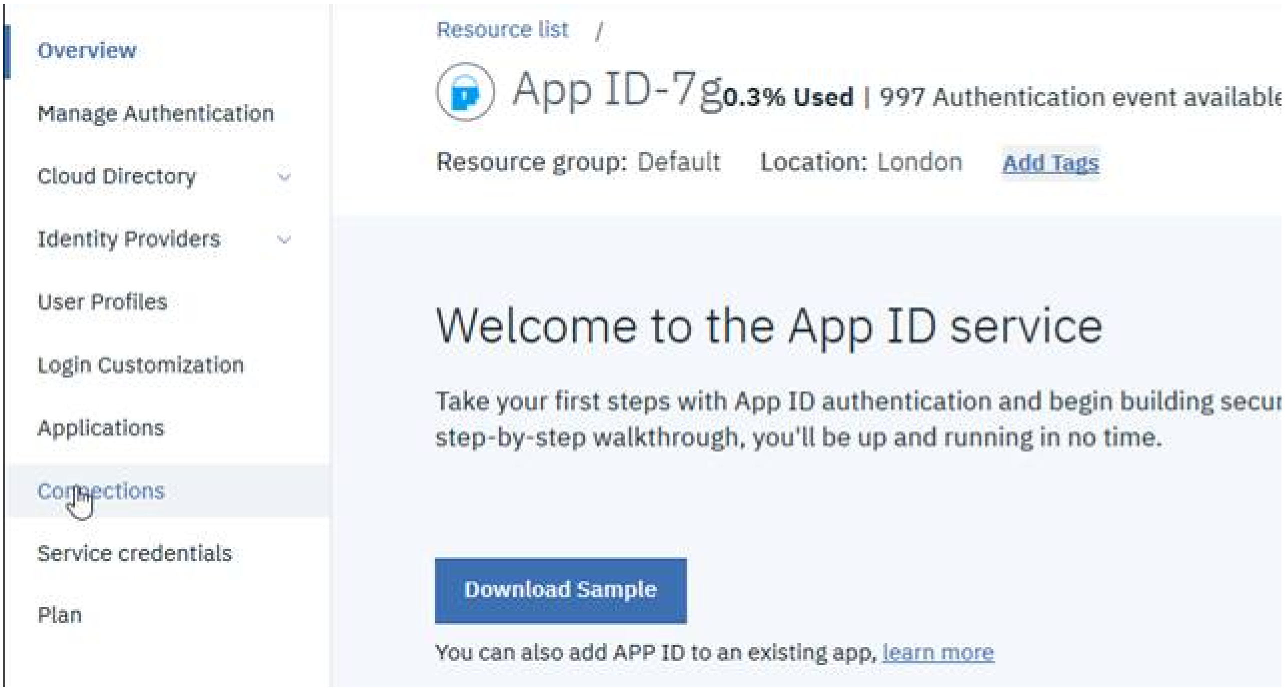
Perform the following steps:

__ 1. Access the App ID service. From the Navigation Menu (upper left) select Resource List.



__ 2. Expand **Services** and click your App ID service. The Welcome to the App ID service page is displayed.

__ 3. From the options on the left, click **Connections** as shown in the following figure.

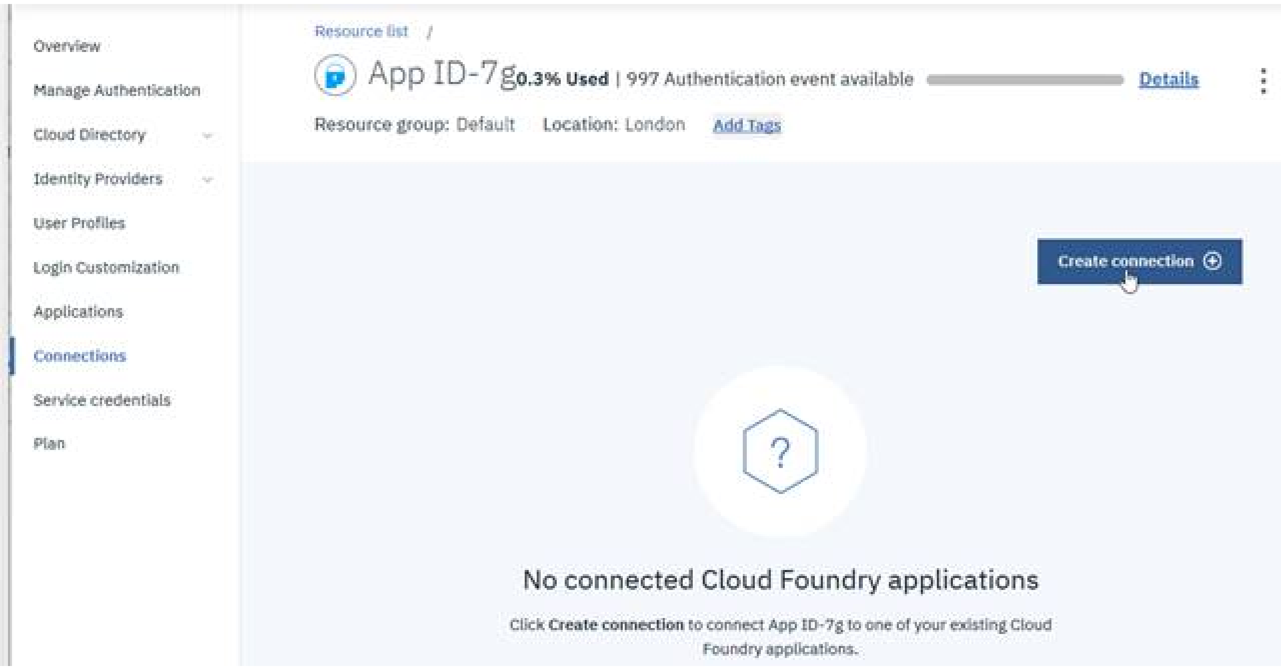


Welcome to the App ID service

Take your first steps with App ID authentication and begin building secu
step-by-step walkthrough, you'll be up and running in no time.

You can also add APP ID to an existing app, learn more

__ 4. Click **Create connection**.



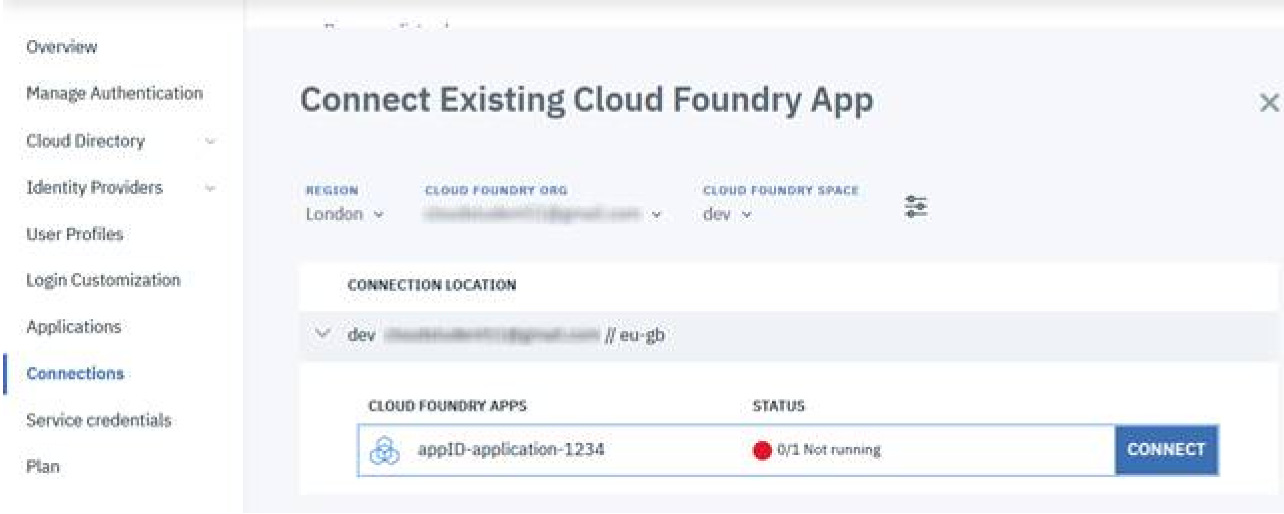
App ID-7g 0.3% Used | 997 Authentication event available

Resource group: Default Location: London

No connected Cloud Foundry applications

Click **Create connection** to connect App ID-7g to one of your existing Cloud Foundry applications.

__ 5. Make sure that the correct region and organization (your IBM Cloud ID) are selected and choose your Node.js application, as shown in the following figure.



Connect Existing Cloud Foundry App

REGION: London

CLOUD FOUNDRY ORG: [redacted]

CLOUD FOUNDRY SPACE: dev

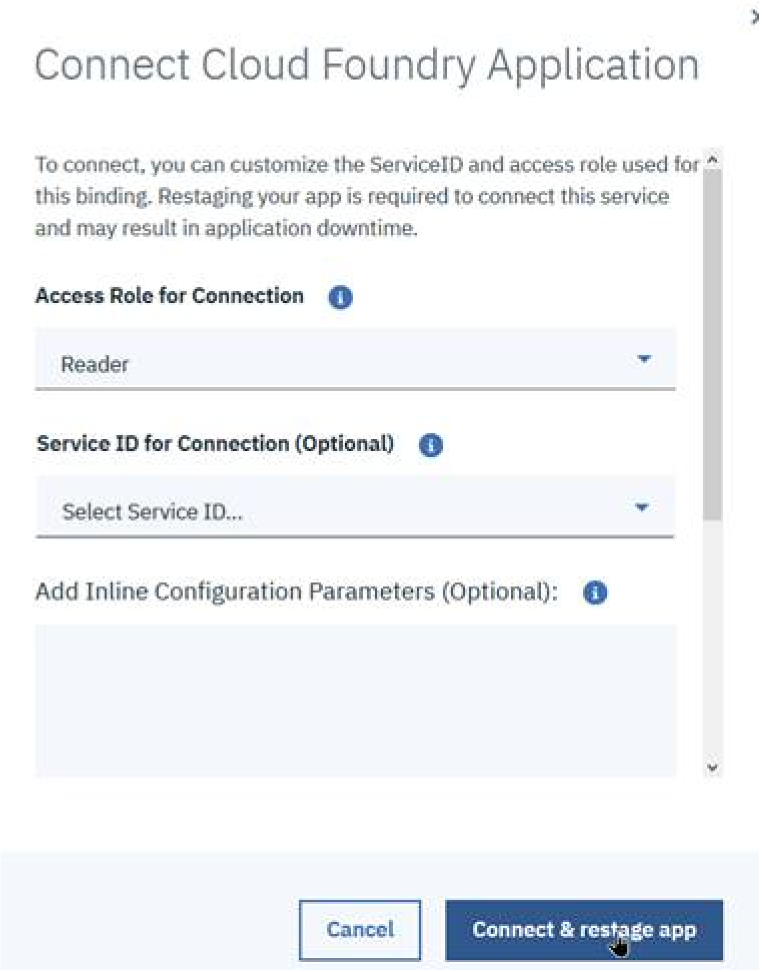
CONNECTION LOCATION

dev [redacted] // eu-gb

CLOUD FOUNDRY APPS	STATUS
appID-application-1234	● 0/1 Not running

__ 6. Click **CONNECT**.

__ 7. In the next window, click **Connect & restage app**, as shown in the following figure.



Connect Cloud Foundry Application

To connect, you can customize the ServiceID and access role used for this binding. Restaging your app is required to connect this service and may result in application downtime.

Access Role for Connection

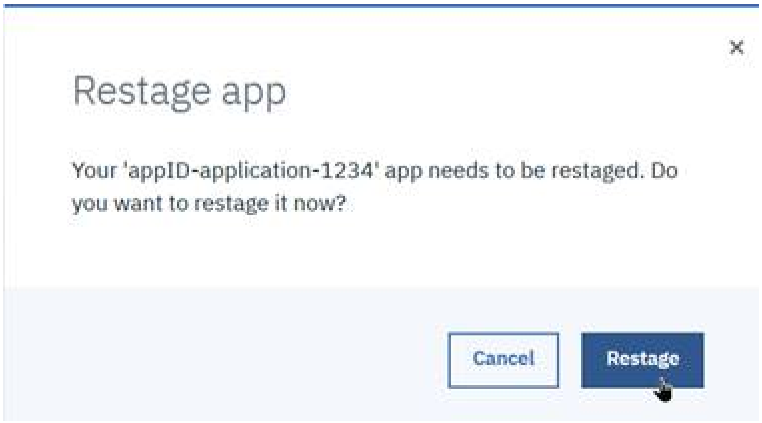
Reader

Service ID for Connection (Optional)

Select Service ID...

Add Inline Configuration Parameters (Optional):

__ 8. Click **Restage**.



Restage app

Your 'appID-application-1234' app needs to be restaged. Do you want to restage it now?

Part 5: Configuring the application to integrate with the App ID service

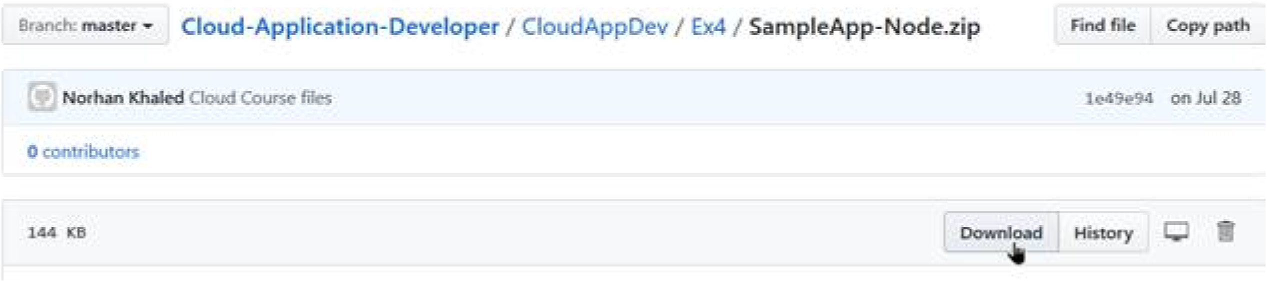
In this part, you will:

- Download a sample Node.js application code from a Git repo.
- Modify application files to configure the sample app to integrate it with the App ID service.
- Deploy the app by pushing the changes out to Cloud Foundry on IBM Cloud with IBM Cloud CLI.

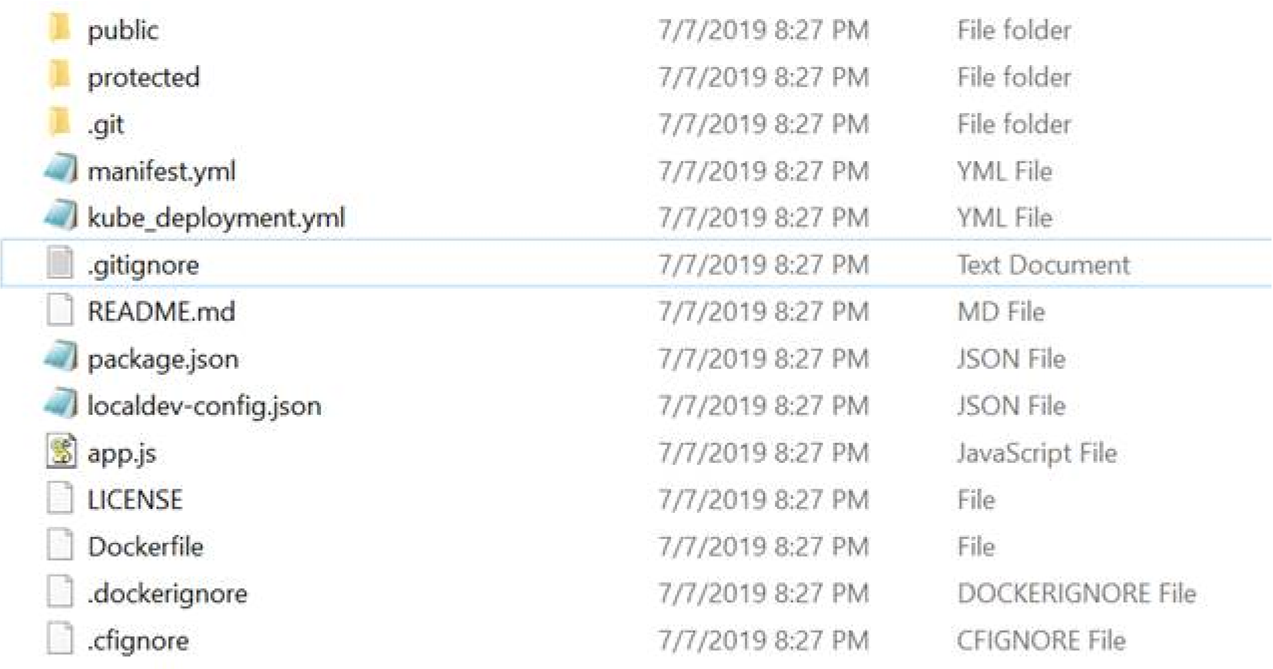
Downloading the sample Node.js app code

Complete these steps:

__ 1. Download the sample app. Access the sample app at this link https://github.com/IBM-SkillsAcademy/Cloud-Application-Developer/blob/master/CloudAppDev/Ex4/SampleApp-Node.zip and click **Download**.


Branch: master
Cloud-Application-Developer / CloudAppDev / Ex4 / SampleApp-Node.zip
Find file
Copy path
Norhan Khaled Cloud Course files
1e49e94 on Jul 28
0 contributors
144 KB
Download
History

__ 2. Open the file and extract its content into your preferred folder. The sample app files are shown in the following figure.



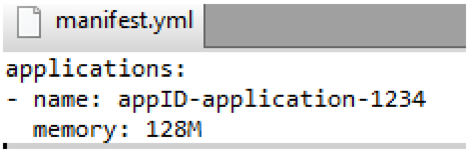
public	7/7/2019 8:27 PM	File folder
protected	7/7/2019 8:27 PM	File folder
.git	7/7/2019 8:27 PM	File folder
manifest.yml	7/7/2019 8:27 PM	YML File
kube_deployment.yml	7/7/2019 8:27 PM	YML File
.gitignore	7/7/2019 8:27 PM	Text Document
README.md	7/7/2019 8:27 PM	MD File
package.json	7/7/2019 8:27 PM	JSON File
localdev-config.json	7/7/2019 8:27 PM	JSON File
app.js	7/7/2019 8:27 PM	JavaScript File
LICENSE	7/7/2019 8:27 PM	File
Dockerfile	7/7/2019 8:27 PM	File
.dockerignore	7/7/2019 8:27 PM	DOCKERIGNORE File
.cfignore	7/7/2019 8:27 PM	CFIGNORE File

Modifying the sample app files

In this section, you modify the `manifest.yaml` file to customize your app name and you modify the `localdev-config.json` file to integrate the sample app with the App ID service that you created in Part 1.

Complete these steps:

__ 1. Edit the **manifest.yaml** file and change the application name to match the unique name of the application that you created in Part 4. For example, **appID-application-xxxx** where “xxxx” is your unique identifier. The following figure shows an example.

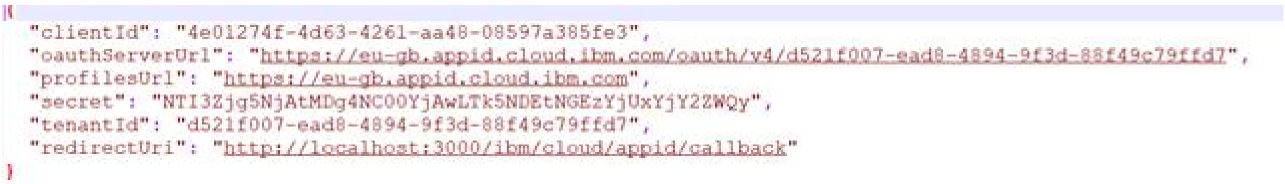


manifest.yml

```
applications:
- name: appID-application-1234
  memory: 128M
```

__ 2. Edit the **localdev-config.json** file with your preferred editor.

__ 3. Replace the values for `clientId`, `oauthServerUrl`, `profilesUrl`, `secret`, and `tenantId` that are shown in the following figure, with the corresponding values from the App ID service credentials that you saved in Part 3.



```
{
  "clientId": "4e01274f-4d63-4261-aa48-08597a385fe3",
  "oauthServerUrl": "https://eu-gb.appid.cloud.ibm.com/oauth/v4/d521f007-ead8-4894-9f3d-88f49c79ffd7",
  "profilesUrl": "https://eu-gb.appid.cloud.ibm.com",
  "secret": "NTI3Zjg5NjAtMDg4NC00YjAwLTk5NDEtNGEzYjUxYjY2ZWQy",
  "tenantId": "d521f007-ead8-4894-9f3d-88f49c79ffd7",
  "redirectUri": "http://localhost:3000/ibm/cloud/appid/callback"
}
```

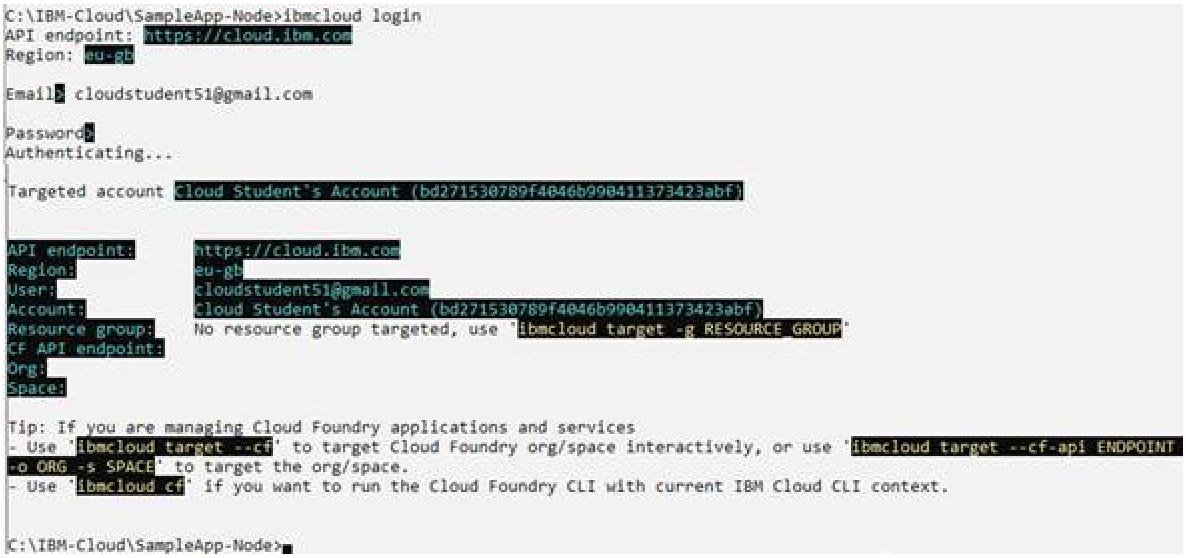
Deploying the sample app to IBM Cloud

To deploy the sample application to IBM Cloud, complete the following steps:

__ 1. Open the Command Prompt and set the current directory to the folder where the sample app code is.

__ 2. Login to IBM Cloud by running the following command:

```
ibmcloud login
```



```
C:\IBM-Cloud\SampleApp-Node>ibmcloud login
API endpoint: https://cloud.ibm.com
Region: eu-gb

Email> cloudstudent51@gmail.com

Password>
Authenticating...

Targeted account Cloud Student's Account (bd271530789f4046b990411373423abf)


API endpoint:      https://cloud.ibm.com
Region:            eu-gb
User:              cloudstudent51@gmail.com
Account:           Cloud Student's Account (bd271530789f4046b990411373423abf)
Resource group:    No resource group targeted, use 'ibmcloud target -g RESOURCE_GROUP'
CF API endpoint:
Org:
Space:

Tip: If you are managing Cloud Foundry applications and services
- Use 'ibmcloud target --cf' to target Cloud Foundry org/space interactively, or use 'ibmcloud target --cf-api ENDPOINT
-o ORG -s SPACE' to target the org/space.
- Use 'ibmcloud cf' if you want to run the Cloud Foundry CLI with current IBM Cloud CLI context.


C:\IBM-Cloud\SampleApp-Node>
```

Note

If the *incorrect* region is selected by default, you can change it by running the command `ibmcloud target -r <region>`, for example `ibmcloud target -r eu-gb`. For the complete list of available regions, see *Regions* at https://cloud.ibm.com/docs/cloud-foundry-public?topic=cloud-foundry-public-endpoints#endpoints_regions

__ 3. Use `--cf-api` to specify the Cloud Foundry API endpoint to which to deploy the application. Select it based on the region where the application was created. Run the following command:

```
ibmcloud target --cf-api <CF API ENDPOINT> -o <ORG> -s <SPACE>
```

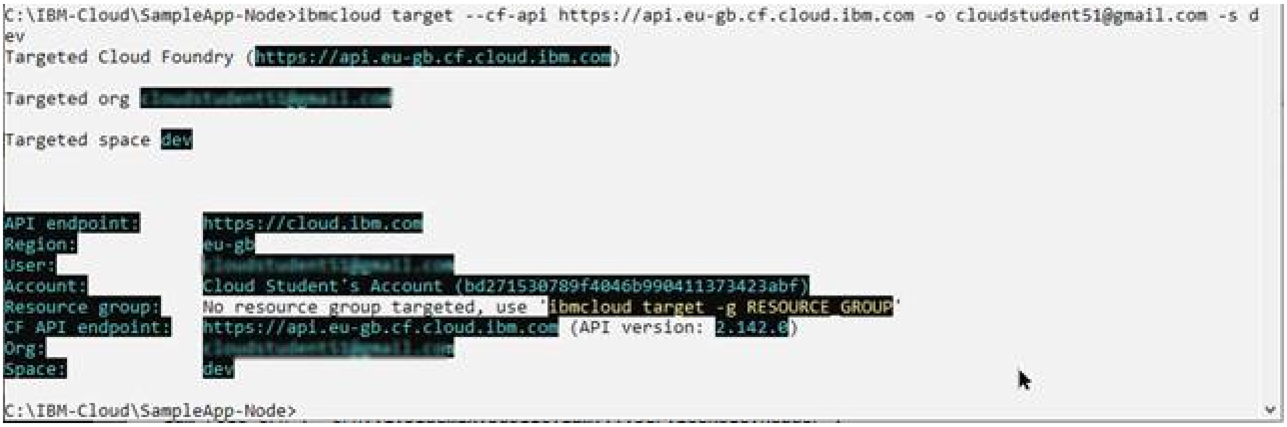
In this example, the command is:

```
ibmcloud target --cf-api https://api.eu-gb.cf.cloud.ibm.com -o <your-email>
-s dev
```

Note

The organization is set by default to your IBMid, which is the email that you use to log in to IBM Cloud, and the space is set by default to `dev`. For the complete list of API endpoints see API Endpoints at https://cloud.ibm.com/docs/cloud-foundry-public?topic=cloud-foundry-public-endpoints#api-endpoint-options

The following figure shows an example of the command output.

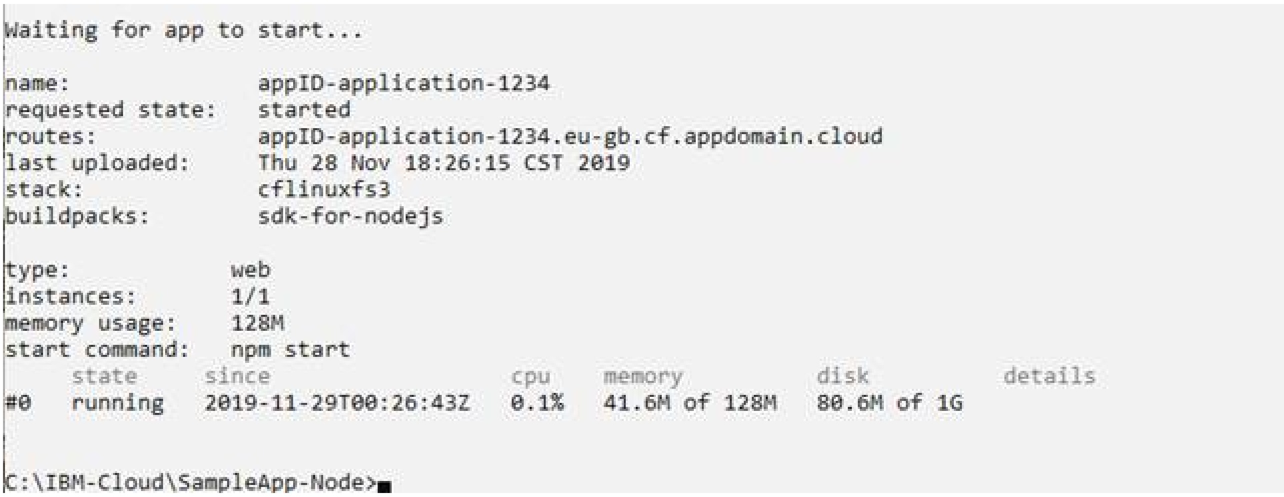


```
C:\IBM-Cloud\SampleApp-Node>ibmcloud target --cf-api https://api.eu-gb.cf.cloud.ibm.com -o cloudstudent51@gmail.com -s d
ev
Targeted Cloud Foundry (https://api.eu-gb.cf.cloud.ibm.com)

Targeted org [redacted]

Targeted space dev


API endpoint:      https://cloud.ibm.com
Region:            eu-gb
User:              [redacted]
Account:           Cloud Student's Account (bd271530789f4046b990411373423abf)
Resource group:    No resource group targeted, use 'ibmcloud target -g RESOURCE_GROUP'
CF API endpoint:   https://api.eu-gb.cf.cloud.ibm.com (API version: 2.142.0)
Org:               [redacted]
Space:             dev

C:\IBM-Cloud\SampleApp-Node>
```

__ 4. Deploy the application on IBM Cloud by running the following command:

```
ibmcloud cf push
```

The following figure shows the output of the command.



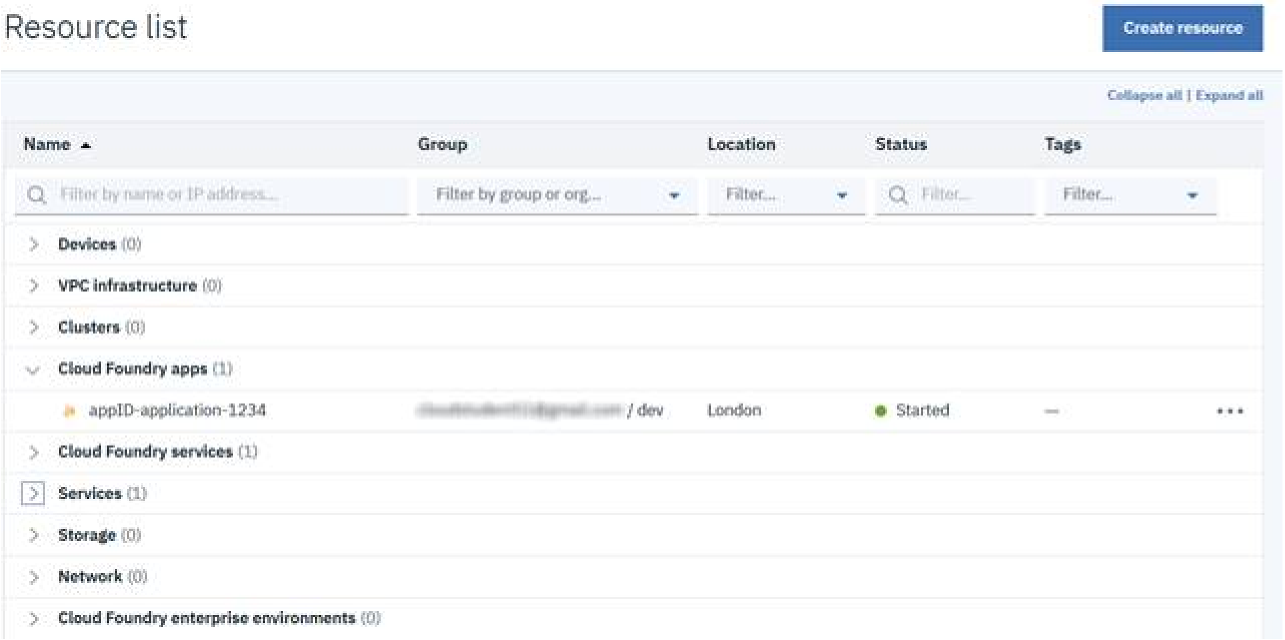
```
Waiting for app to start...

name:              appID-application-1234
requested state:   started
routes:            appID-application-1234.eu-gb.cf.appdomain.cloud
last uploaded:     Thu 28 Nov 18:26:15 CST 2019
stack:             cflinuxfs3
buildpacks:        sdk-for-nodejs

type:            web
instances:       1/1
memory usage:    128M
start command:   npm start
     state     since                  cpu    memory         disk          details
#0   running   2019-11-29T00:26:43Z   0.1%   41.6M of 128M  80.6M of 1G


C:\IBM-Cloud\SampleApp-Node>
```

Copying the sample app URL

In this section, you copy the sample app URL to use it later in the configuration.

Complete these steps:

__ 1. Open the application in the browser. Display IBM Cloud **Resource List**, and expand **Cloud Foundry Apps**, as shown in the following figure.



Resource list

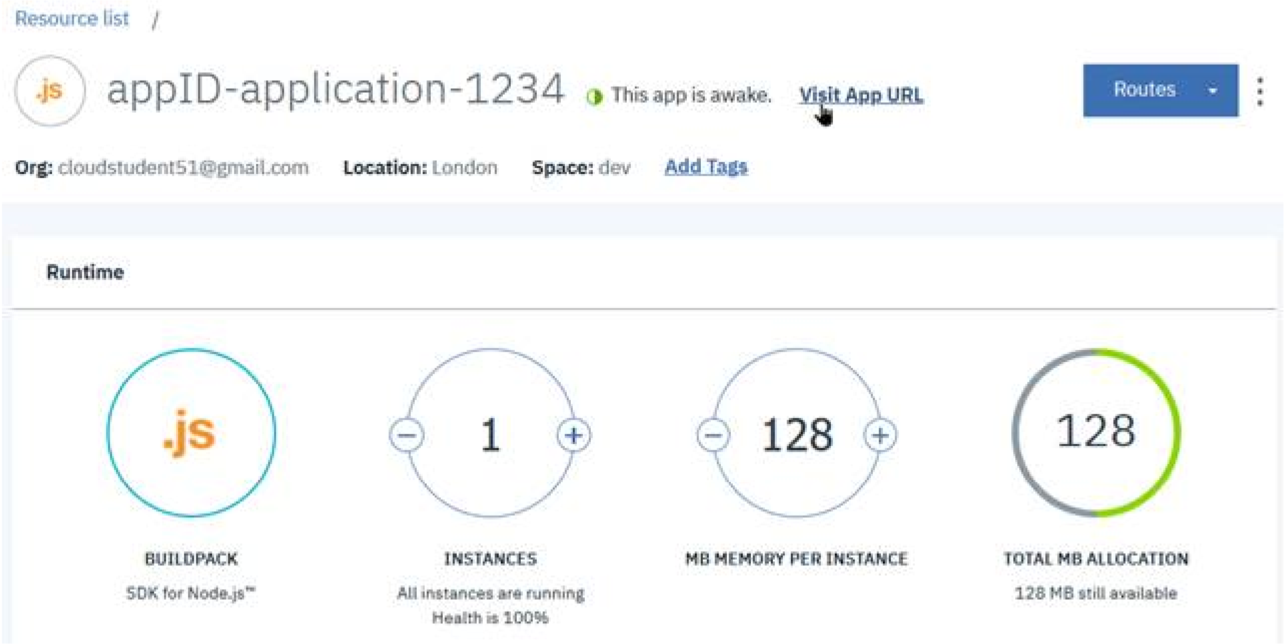
Create resource

Collapse all | Expand all

Name ▴	Group	Location	Status	Tags
Filter by name or IP address...	Filter by group or org...	Filter...	Filter...	Filter...
› Devices (0)				
› VPC infrastructure (0)				
› Clusters (0)				
⌄ Cloud Foundry apps (1)				
appID-application-1234	[redacted] / dev	London	● Started	—
› Cloud Foundry services (1)				
› Services (1)				
› Storage (0)				
› Network (0)				
› Cloud Foundry enterprise environments (0)				

__ 2. Click your application name under Cloud Foundry Apps.

__ 3. Click **Visit App URL**, as shown in the following figure.



Resource list /

appID-application-1234 This app is awake. Visit App URL

Routes

Org: cloudstudent51@gmail.com Location: London Space: dev Add Tags

Runtime

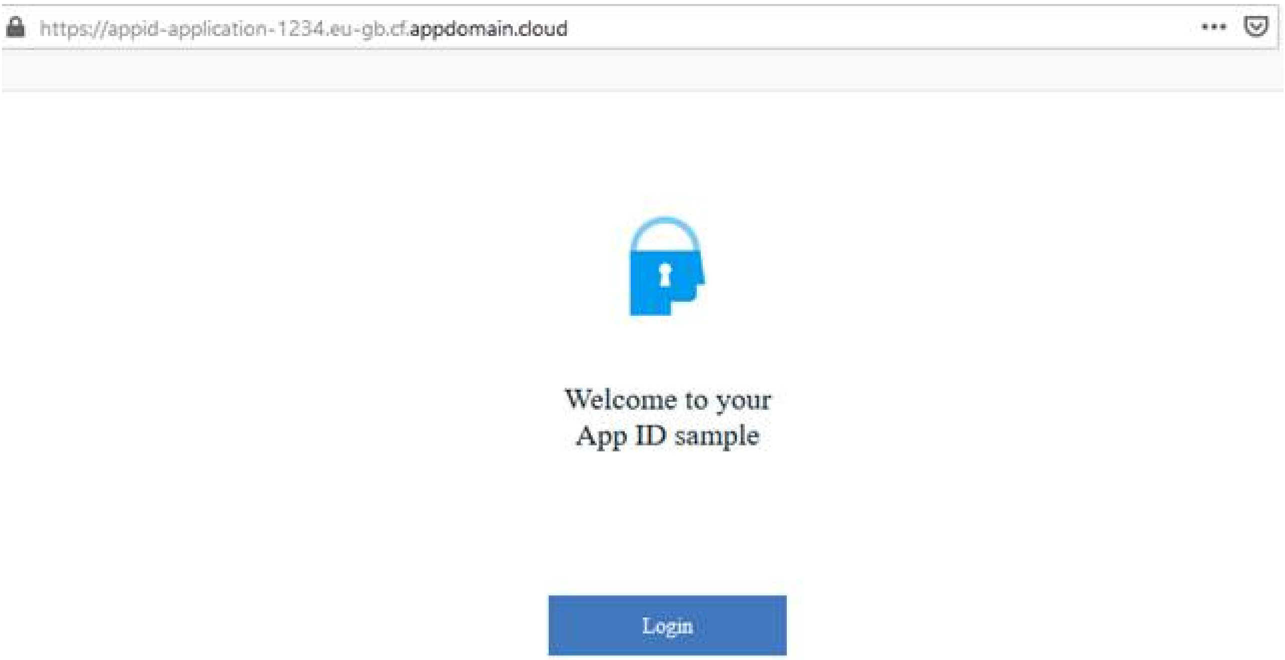
BUILDPACK
SDK for Node.js™

INSTANCES
All instances are running
Health is 100%

MB MEMORY PER INSTANCE

TOTAL MB ALLOCATION
128 MB still available

- The application opens in another tab, as shown in the following figure.



https://appid-application-1234.eu-gb.cf.appdomain.cloud

Welcome to your
App ID sample

__ 4. Copy and save the URL for later use.

Part 6: Securing the sample application with the App ID service

In this part, you learn how to use the App ID service and bind it to your application to secure your application and authenticate through different identity providers.

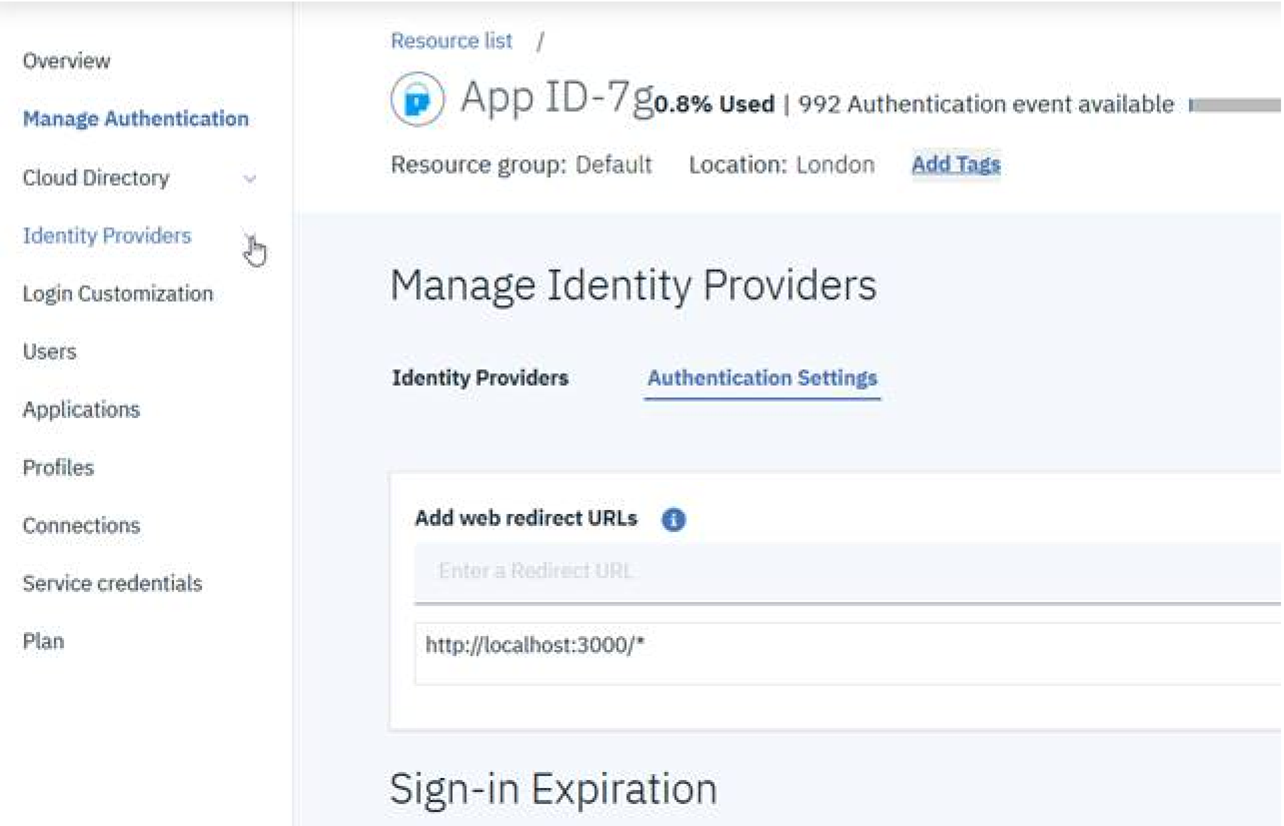
Configuring the application URL

Now, you configure the URL of the deployed application in the service instance. You explored this process in Part 2 when you looked at Managing Identity Providers.

Complete the following steps:

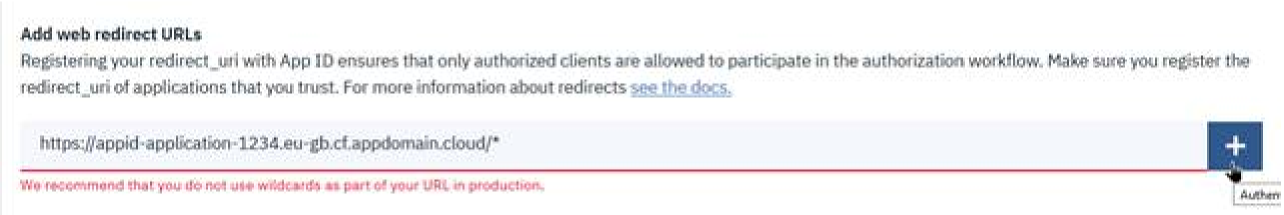
__ 1. Open a new browser tab and access your App ID service dashboard.

__ 2. Click **Manage Authentication** and then click **Authentication Settings**, as shown in the following figure.



http://localhost:3000/*

__ 3. Add the application URL that you saved previously to **Add web redirect URLs** so that the application can redirect the user to the App ID for authentication. Add /* after the URL and then click **+** as shown in the following figure.


Add web redirect URLs

Registering your redirect_uri with App ID ensures that only authorized clients are allowed to participate in the authorization workflow. Make sure you register the redirect_uri of applications that you trust. For more information about redirects see the docs.

https://appid-application-1234.eu-gb.cf.appdomain.cloud/*

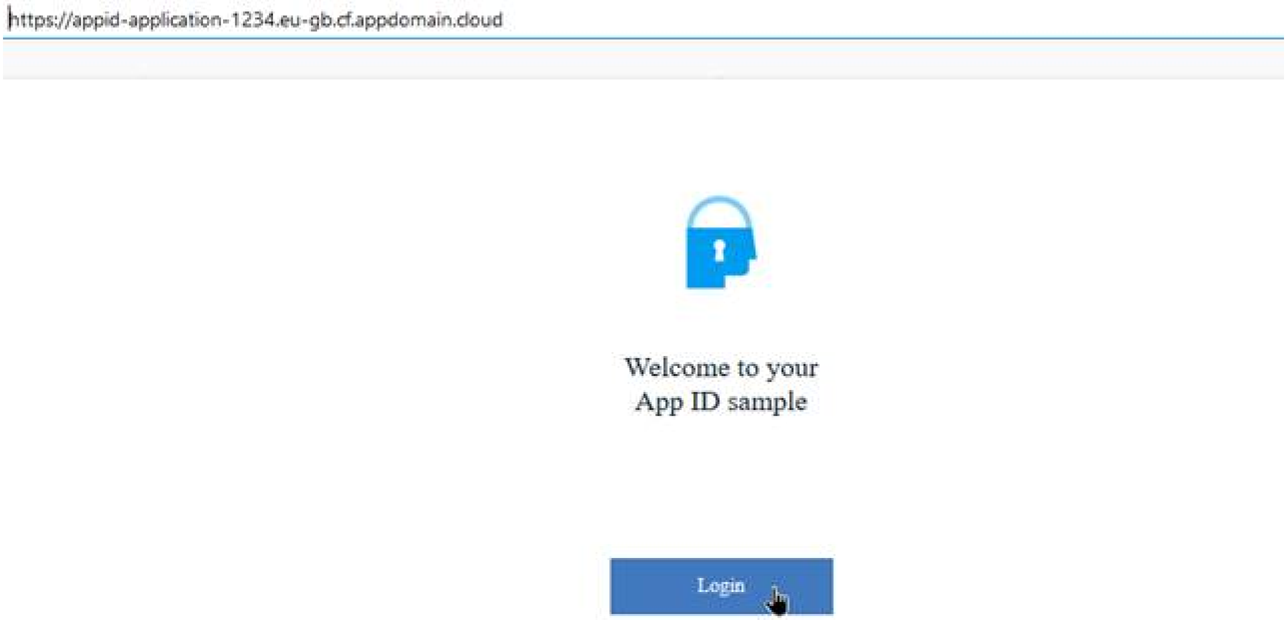
We recommend that you do not use wildcards as part of your URL in production.

Now, the App ID accepts requests from only this URL.

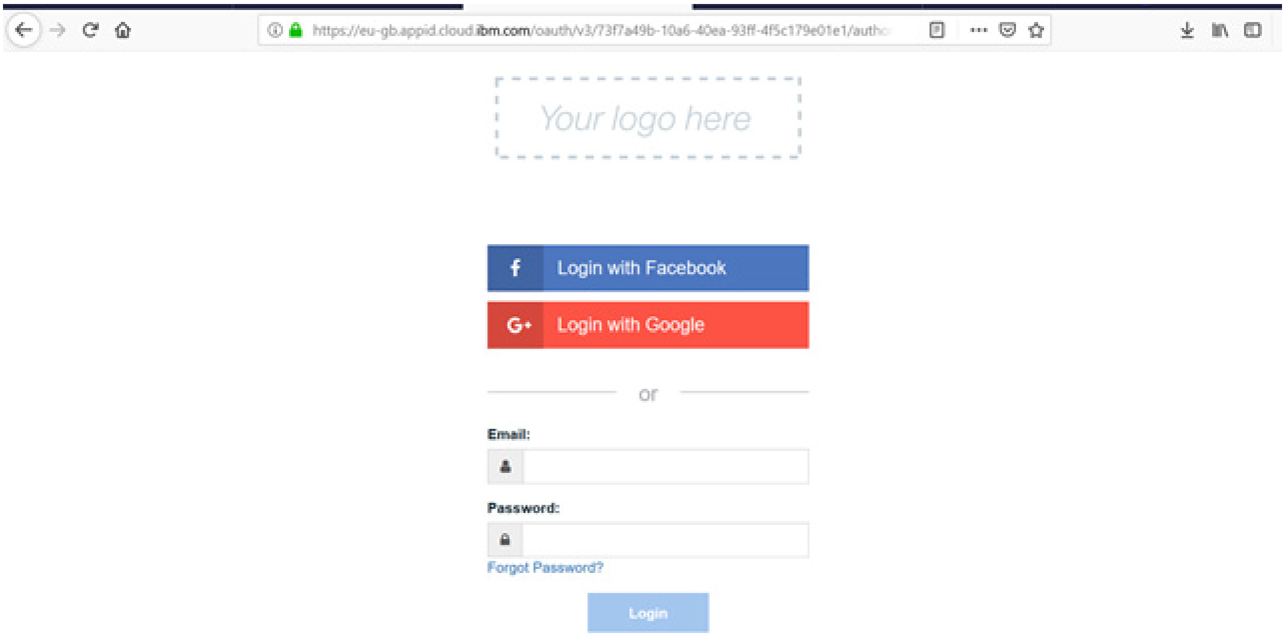
Signing in to the secured application

To verify that authentication is now required to access your application, complete the following steps:

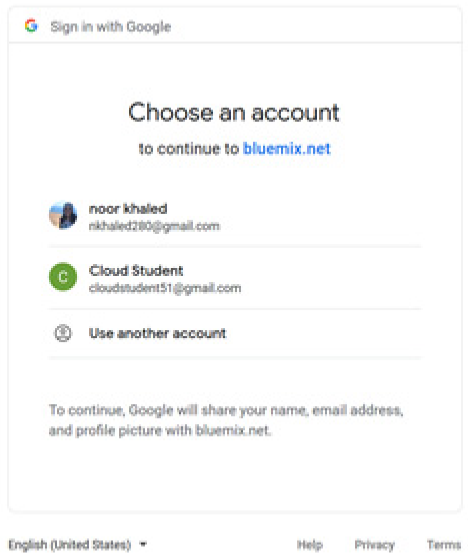
__ 1. On the opened application tab, take note of the URL of the application and click **Login**, as shown in the following figure.


https://appid-application-1234.eu-gb.cf.appdomain.cloud
Welcome to your App ID sample
Login

__ 2. You are redirected to the App ID login to verify your identity, as shown in the following figure.



The identity providers that you saw in Part 2 are listed. If you have a Gmail account, login with your Google credentials. If you want to log in by using your Facebook account, click Login with Facebook. In this exercise, we chose to log in by using Google, as shown in the following figure.



Sign in with Google

Choose an account

to continue to bluemix.net

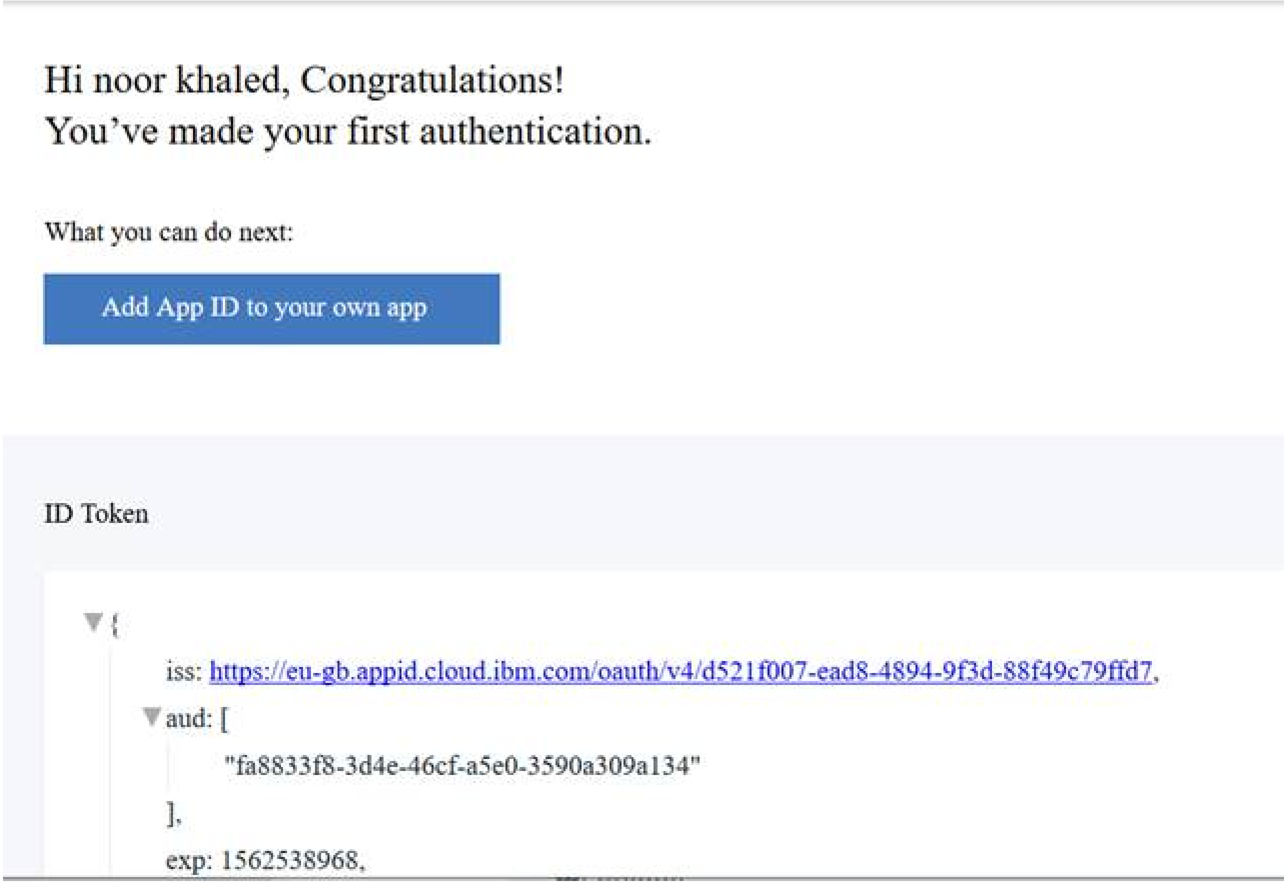
noor khaled
nkhaled280@gmail.com

Cloud Student
cloudstudent51@gmail.com

Use another account

To continue, Google will share your name, email address, and profile picture with bluemix.net.

__ 3. After you enter your Google account credentials, you are logged in to your application, as shown in the following figure. The application received the user's name and image from the identity provider (Google in this case).



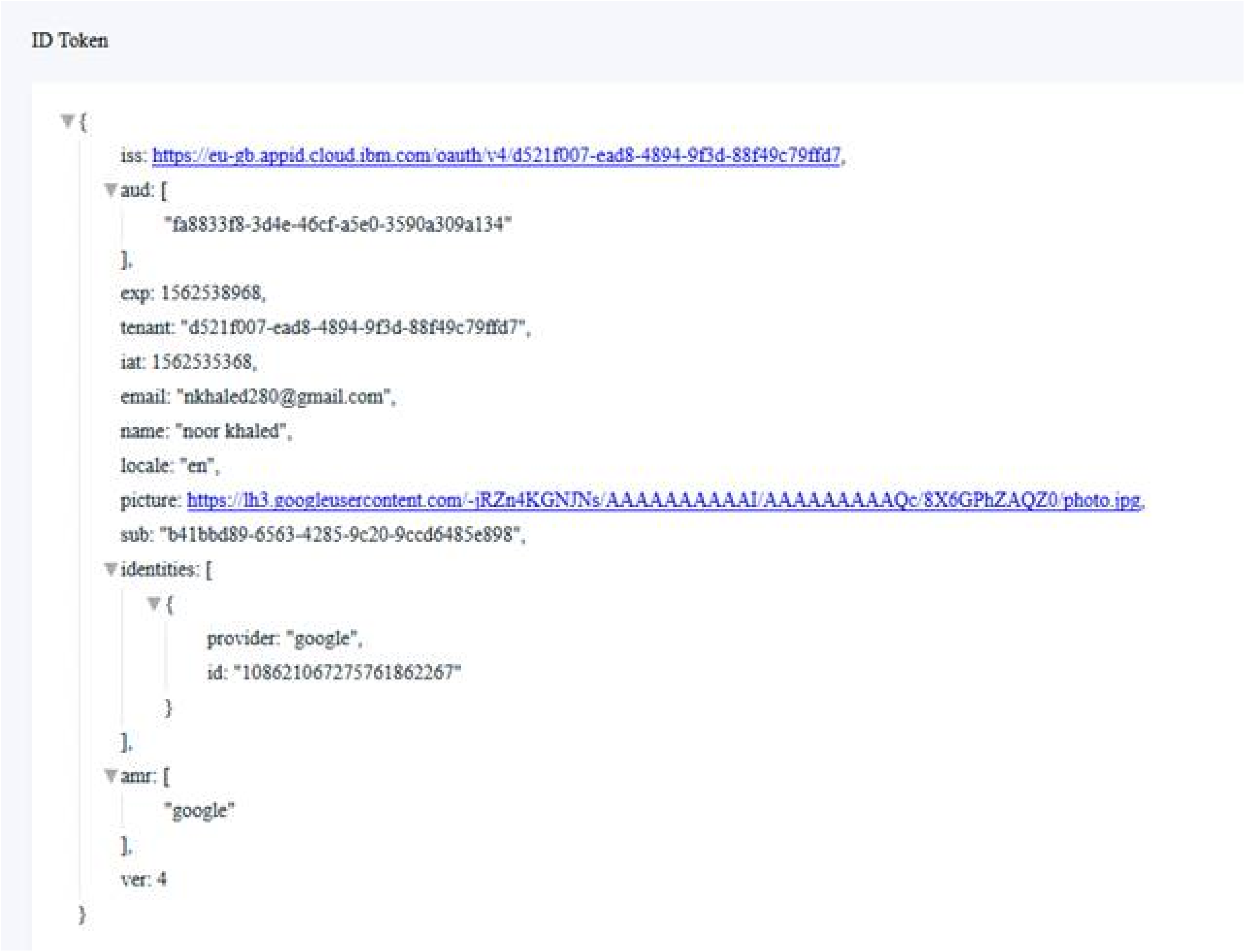
Hi noor khaled, Congratulations!
You've made your first authentication.

What you can do next:

ID Token

```
▼{
    iss: https://eu-gb.appid.cloud.ibm.com/oauth/v4/d521f007-ead8-4894-9f3d-88f49c79ffd7,
  ▼aud: [
        "fa8833f8-3d4e-46cf-a5e0-3590a309a134"
    ],
    exp: 1562538968,
```

__ 4. Browse through the ID Token to display the user's verified token and the basic user's information as shown in the following figure.



ID Token

```
{
    iss: https://eu-gb.appid.cloud.ibm.com/oauth/v4/d521f007-ead8-4894-9f3d-88f49c79ffd7,
    aud: [
        "fa8833f8-3d4e-46cf-a5e0-3590a309a134"
    ],
    exp: 1562538968,
    tenant: "d521f007-ead8-4894-9f3d-88f49c79ffd7",
    iat: 1562535368,
    email: "nkhaled280@gmail.com",
    name: "noor khaled",
    locale: "en",
    picture: https://lh3.googleusercontent.com/-jRZn4KGNJNs/AAAAAAAAAAI/AAAAAAAAAQc/8X6GPhZAQZ0/photo.jpg,
    sub: "b41bbd89-6563-4285-9c20-9ccd6485e898",
    identities: [
        {
            provider: "google",
            id: "108621067275761862267"
        }
    ],
    amr: [
        "google"
    ],
    ver: 4
}
```

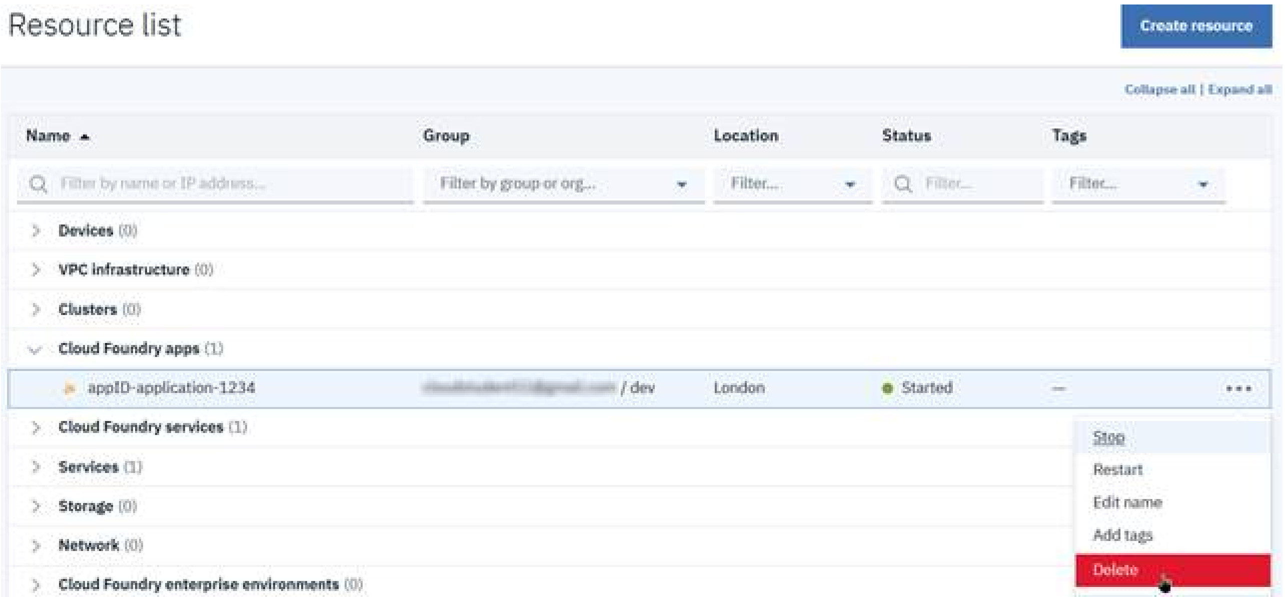
Part 7: Cleaning up the environment

In this part, you delete the instance of the application and App ID service that you created.

Complete the following steps:

__ 1. Open the IBM Cloud dashboard.

__ 2. Under Cloud Foundry Apps, click **Actions** (the three dots) for your application, as shown in the following figure.

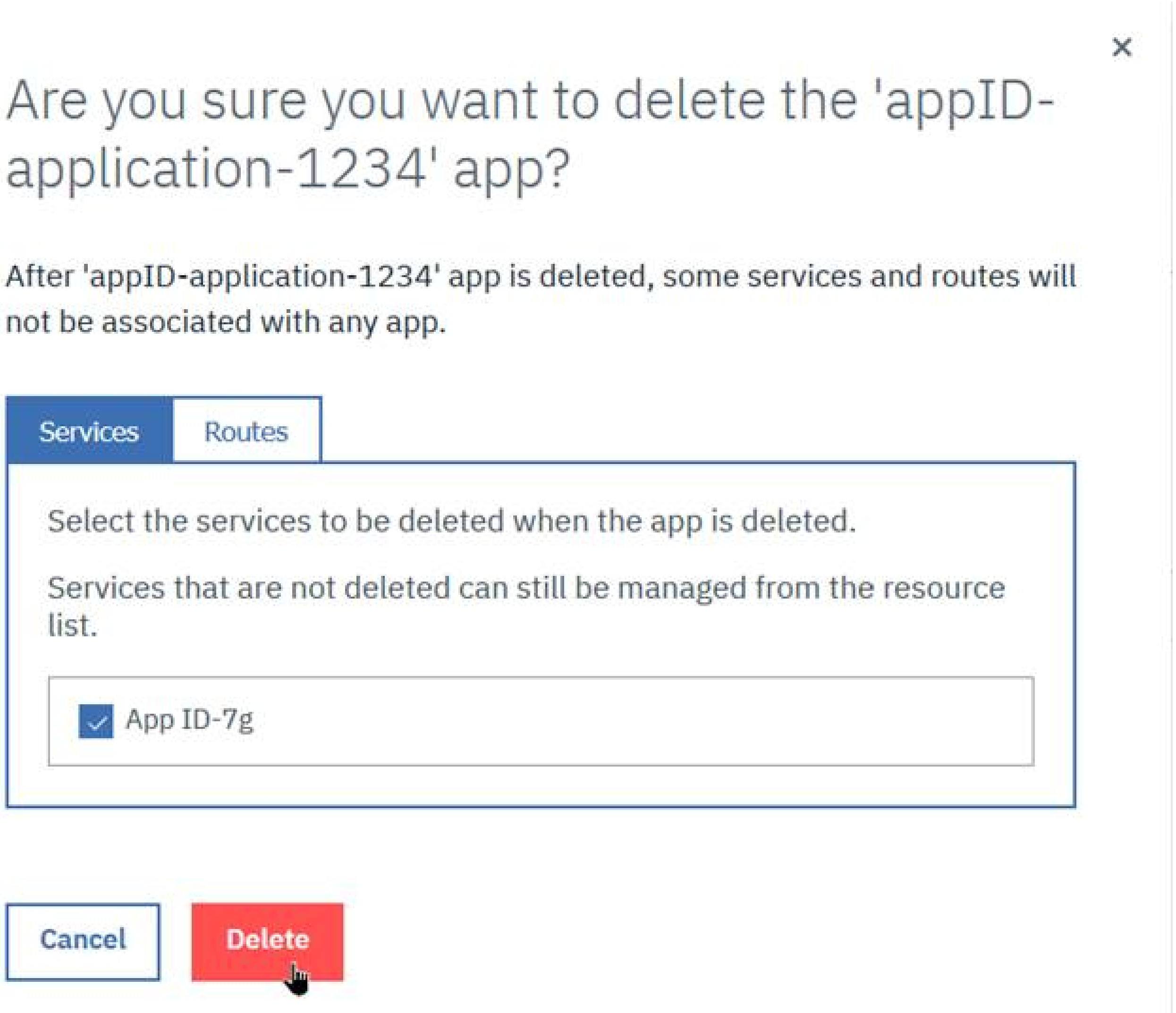


Resource list

Name ▲	Group	Location	Status	Tags
Filter by name or IP address...	Filter by group or org...	Filter...	Filter...	Filter...
> Devices (0)				
> VPC infrastructure (0)				
> Clusters (0)				
⌄ Cloud Foundry apps (1)				
appID-application-1234	[redacted] / dev	London	● Started	—
> Cloud Foundry services (1)				
> Services (1)				
> Storage (0)				
> Network (0)				
> Cloud Foundry enterprise environments (0)				

__ 3. Click **Delete**.

__ 4. Select all the associated services and routes and click **Delete**, as shown in the following figure.



Are you sure you want to delete the 'appID-application-1234' app?

After 'appID-application-1234' app is deleted, some services and routes will not be associated with any app.

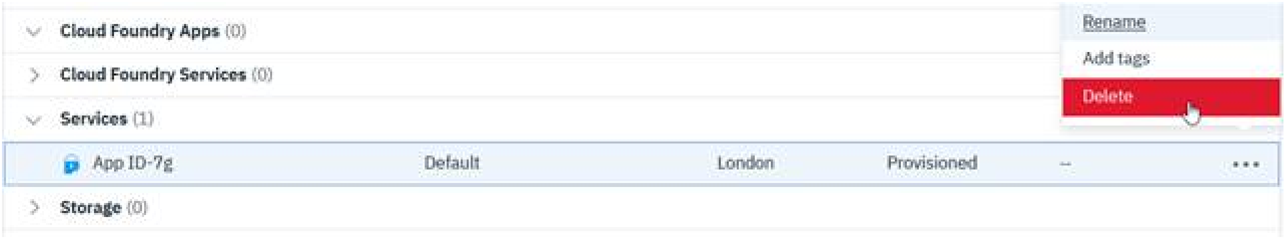
Select the services to be deleted when the app is deleted.

Services that are not deleted can still be managed from the resource list.

☑ App ID-7g

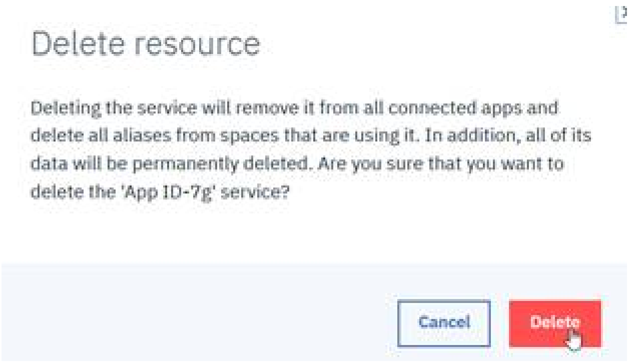
__ 5. Expand **Services**.

__ 6. Click the **Actions** menu (three dots) for your **App ID** service and select **Delete**.



Cloud Foundry Apps (0)					
Cloud Foundry Services (0)					
Services (1)					
App ID-7g	Default	London	Provisioned	–	•••
Storage (0)					

__ 7. At the Delete resource prompt, click **Delete** to confirm.



Delete resource

Deleting the service will remove it from all connected apps and delete all aliases from spaces that are using it. In addition, all of its data will be permanently deleted. Are you sure that you want to delete the 'App ID-7g' service?

End of exercise

Exercise review and wrap-up

In this exercise, you learned how to create an App ID service and explore the various tabs of the service. You also learned some terms, such as identity providers and cloud directory users.

You downloaded a sample Node.js application, bound it to the App ID service, deployed it on IBM Cloud, and tested the authentication by using an identity provider (in this example, Google).

You saw how Google verified the user and sent all the details about this user to the application to provide a better user experience.

You saw how App ID supports user profiles when you logged in with Google and clicked the user information. You found all information about the user, which you can use to enhance the user experience in your application.

You deleted the application instance to perform other exercises.