



**IBM Developer Advocacy México** 

## Pre-requisitos

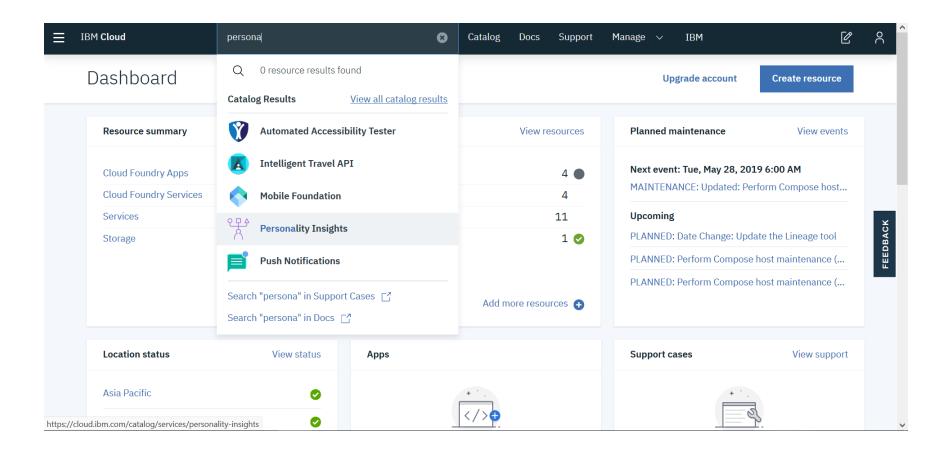
- · Cuenta de IBM Cloud
- Editor de código(<a href="https://code.visualstudio.com/download">https://code.visualstudio.com/download</a>)
- Node.js (<a href="https://nodejs.org/en">https://nodejs.org/en</a>)

## **Personality Insights**

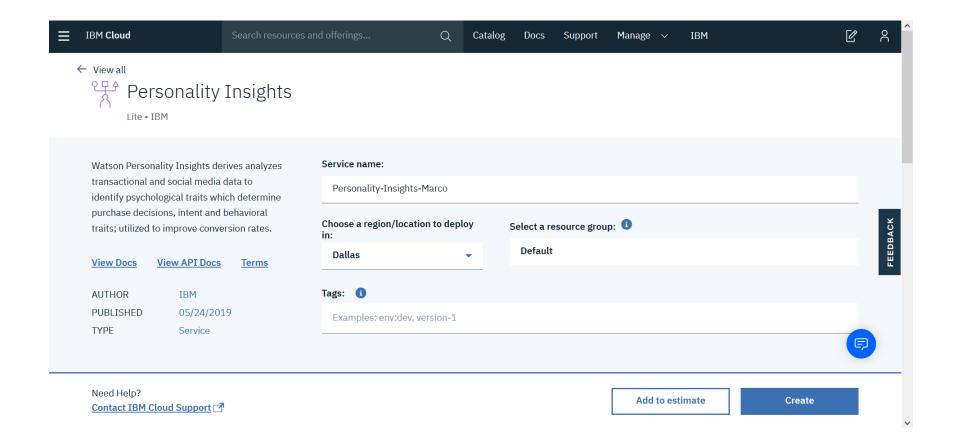
## Pre-requisitos

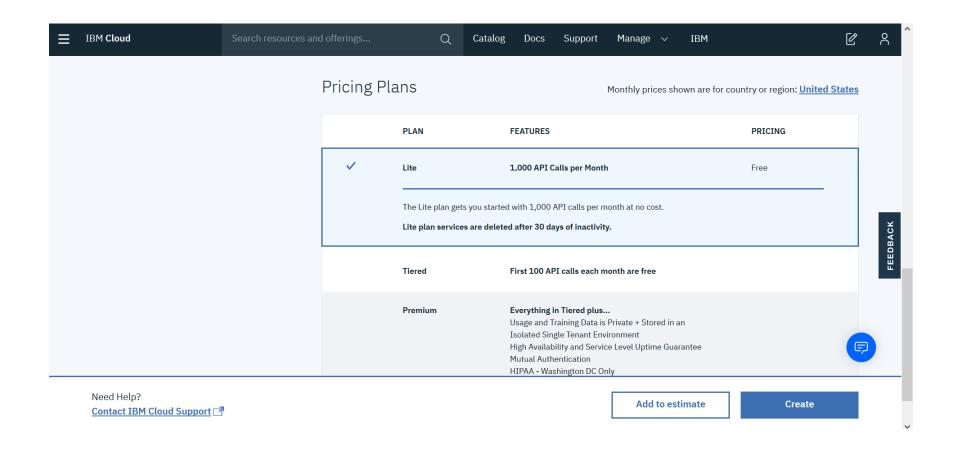
• Clonar Repositorio de Github(<a href="https://github.com/ibmdbgmx/bootcampMx-PersonalityInsights">https://github.com/ibmdbgmx/bootcampMx-PersonalityInsights</a>) desde la pagina o desde el cli con el comando **git clone**.

Crearemos los servicios de IBM. Lo buscaremos en el catálogo.



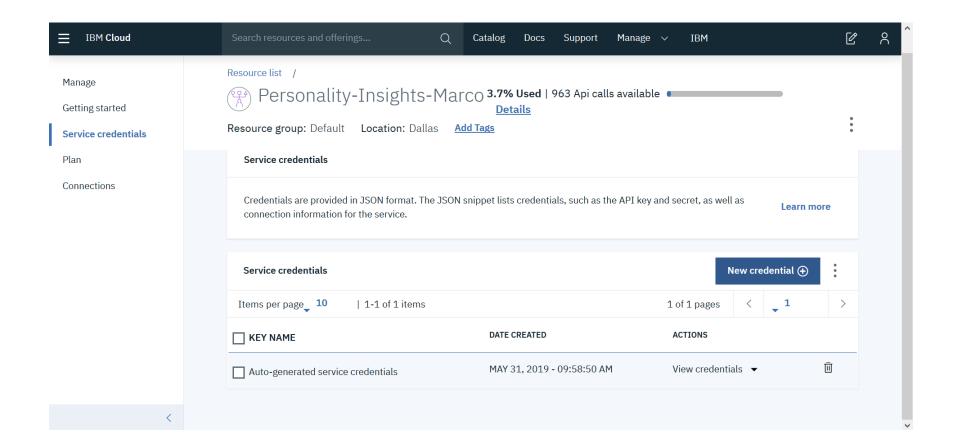
Le asignamos un nombre y seleccionamos un plan para el servicio.



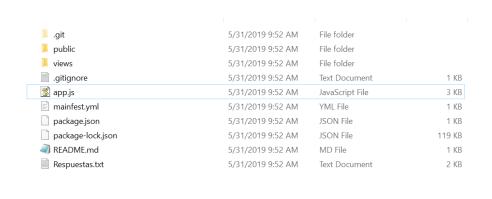


Daremos click en el botón Create.

Nos vamos a la pestaña de Service Credentials, si no nos creo una, damos click New credential



En nuestra carpeta que clonamos, buscamos nuestro archivo app.js y ubicamos **USERNAME:** y **PASSWORD:** 



```
//set IBM cloud credentials in this section
var personalityInsights = new PersonalityInsightsV3({
   version_date: '2017-10-13',
   username: '',
   password: ''
});
```

Las configuraremos con las credenciales que obtuvimos en el servicio creado y salvamos nuestro archivo.

En la terminal ejecutamos el comando **npm install** y luego **node app** ubicandonos en el directorio que clonamos.

```
PS C:\Users\MARCOANTONIOALVAREZG\Documents\workshop\bootcampMx-PersonalityInsights> npm install
> nodemon@1.17.5 postinstall C:\Users\MARCOANTONIOALVAREZG\Documents\workshop\bootcampMx-PersonalityInsights\node_modules\nodemon
> node bin/postinstall || exit 0

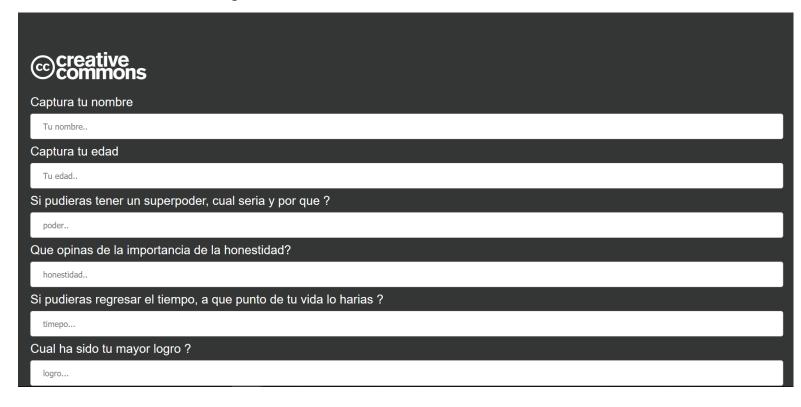
cove_modument? You can now support the project via the open collective:
> https://opencollective.com/nodemon/donate

npm MARN nodeuploads@1.0.0 No repository field.

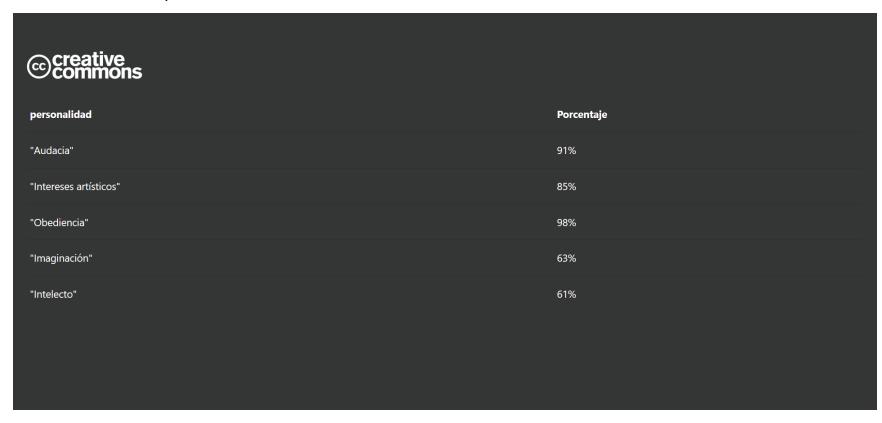
added 31 packages from 35 contributors and audited 1658 packages in 3.844s
found 3 vulnerabilities (1 moderate, 2 html)
run `npm audit fix` to fix them, or `npm audit` for details

PS C:\Users\MARCOANTONIOALVAREZG\Documents\workshop\bootcampMx-PersonalityInsights> node app
Server started on port 8080
```

Listo. Nuestra aplicación esta desplegada de manera local en el puerto 8080... nos dirigiremos a localhost:8080 en el navegador.



Después de responder el formulario, y enviar las respuestas. Nos regresara una pantalla como la siguiente. Donde ya hace el análisis de nuestras respuestas y nos devuelve específicamente 5 características de personalidad.



En esta parte del código es done se hace el llenado de los parámetros y del perfil de las 5 características que nos arroja la página.

```
personalityInsights.profile(profileParams, function(error, profile) {
(%)
               if (error) {
                 console.log(error);
x = profile;
                 var audaciaN = JSON.stringify(x.personality[0].children[0].name, null, 2);
                 var audaciaP = JSON.stringify(x.personality[0].children[0].percentile, null, 2);
                 var inAr = JSON.stringify(x.personality[0].children[1].name, null, 2);
                 var inArP = JSON.stringify(x.personality[0].children[1].percentile, null, 2);
                 var emocional = JSON.stringify(x.personality[1].children[2].name, null, 2);
                 var emocionalP = JSON.stringify(x.personality[1].children[2].percentile, null, 2);
                 var imagin = JSON.stringify(x.personality[0].children[3].name, null, 2);
                 var imaginP = JSON.stringify(x.personality[0].children[3].percentile, null, 2);
                 var intelec = JSON.stringify(x.personality[0].children[4].name, null, 2);
                 var intelecP = JSON.stringify(x.personality[0].children[4].percentile, null, 2);
                 var y=Estructurador.cleansing(x);
                 console.log(y);
               res.render('profile', {
               au: audaciaN,
               aup: audaciaP,
               inar: inAr,
               inarp: inArP.
```

El servicio nos despliega mucha más información que se puede visualizar en command line interface de la siguiente manera gracias a la función de Estructurador.js que toma esta información y la muestra de manera más explicita de la siguiente manera.

```
Estabilidad: { N:
Audacia: { N:
                                                                      Estructura: { N:
Disciplina: { N: '0.57202
Autodisciplina: { N:
Autoeficacia: { N: '@
Amabilidad: { N:
Altruismo: { N:
Modestia: { N:
Intransigencia: { N:
Confianza: { N:
Vehemencia: { N:
                                                                                                                                  ': { N:
Desmesura: { N:
Timidez: { N: '
Familiaridad: { N:
Curiosidad: { N:
Ideal: { N:
Libertad: { N:
Practicidad: { N:
```