



EBook Gratis

APRENDIZAJE ibm-bluemix

Free unaffiliated eBook created from
Stack Overflow contributors.

#ibm-

bluemix

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con ibm-bluemix.....	2
Observaciones.....	2
Examples.....	3
Instalación o configuración.....	3
Obtención de la instalación de IBM Bluemix para desplegar una aplicación de nodo.....	3
Paso 1: Crea una cuenta bluemix.....	3
Paso 2: Instale Bluemix y las herramientas de línea de comandos de Cloud Foundry.....	3
Paso 2a: Conectar a Bluemix.....	3
Paso 2b: Iniciar sesión en Bluemix.....	3
Implementación continua de RESTful API (Node.js) a través de GitHub utilizando IBM Bluemix.....	3
Paso 1: Crea tu cuenta de GitHub.....	3
Paso 2: Crea tu cuenta de IBM Bluemix.....	4
Paso 3: Implementa tu paquete de API Rest en GitHub.....	4
Paso 4: implemente su repositorio GitHub en IBM Bluemix usando Toolchain (herramienta de i.....	5
Paso 5: Probar la API Rest con PostMan.....	26
Renuncia:.....	27
Capítulo 2: Mobile Analytics para Bluemix.....	28
Introducción.....	28
Examples.....	28
Introducción a las aplicaciones iOS Swift y Mobile Analytics para Bluemix.....	28
Capítulo 3: Unión.....	30
Examples.....	30
Acceso a credenciales a través de la variable de entorno VCAP_SERVICES.....	30
Creditos.....	32

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [ibm-bluemix](#)

It is an unofficial and free ibm-bluemix ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official ibm-bluemix.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con ibm-bluemix

Observaciones

Bluemix es una plataforma como servicio (PaaS) desarrollada por IBM para crear, ejecutar, implementar y administrar aplicaciones en la nube. Bluemix ofrece un gran catálogo de servicios para integrar con sus aplicaciones. Los lenguajes de programación y tiempos de ejecución admitidos incluyen Java, Node.js, Swift, Go, PHP, Python y Ruby, con lenguajes adicionales compatibles mediante el uso de paquetes de compilación.

IBM Bluemix está basado en la tecnología abierta Cloud Foundry. Además de crear aplicaciones web para varios marcos, incluido iOS, puede crear contenedores de Docker o iniciar instancias de servidores virtuales. Bluemix se basa en un modelo de nube híbrida. Sus aplicaciones, contenedores y máquinas virtuales son portátiles a través de nubes públicas, dedicadas y locales.

La consola Bluemix proporciona placas de aplicación para ayudarte a comenzar. Las consolas Bluemix en Dallas, Londres o Sydney ofrecen un creciente catálogo de más de 100 servicios para construir sus aplicaciones.

Categorías de servicio

- Compute - Múltiples tiempos de ejecución, contenedores basados en Docker, acceso a servidores virtuales
- Red - Conexiones VPN seguras
- Almacenamiento - Almacenes de datos en la nube no estructurados
- Datos y análisis: opciones de base de datos y análisis de big data
- Watson - edificio de aplicación cognitiva
- Integrar: administración de API, integración en la nube, puerta de enlace segura
- DevOps: canalización de entrega, escalado automático, seguimiento y planificación, notificaciones de alerta
- Seguridad - Inicio de sesión único, monitoreo de seguridad
- Servicios de aplicaciones: reglas de negocio, flujo de trabajo, almacenamiento en caché, servidor de aplicaciones y más
- Mobile: servicios para aplicaciones móviles, como notificaciones push, pruebas, ajustes y análisis
- Internet de las cosas: base de IoT e información de datos en tiempo real

Enlaces Útiles

- [Documentación de Bluemix](#) - información técnica extensa y detallada
- Preguntas de [identificación y facturación: preguntas](#) frecuentes sobre problemas comunes de Bluemix
- [Estado de Bluemix](#) : notificaciones sobre eventos que afectan a la plataforma y los servicios
- [Empieza con Bluemix](#)
- Consolas Bluemix - [EE](#) . [UU. Sur](#) , [EU-GB](#) , [AU-SYD](#)

Examples

Instalación o configuración

Instrucciones detalladas sobre cómo configurar o instalar ibm-bluemix.

Obtención de la instalación de IBM Bluemix para desplegar una aplicación de nodo

Paso 1: Crea una cuenta bluemix

Cree una cuenta en <https://console.ng.bluemix.net/registration/>

Esto te preparará con una prueba de 30 días. No tiene que pagar nada por los recursos gratuitos y no tiene que configurar la facturación hasta el final de su prueba (aunque no todos los servicios estarán disponibles).

Paso 2: Instale Bluemix y las herramientas de línea de comandos de Cloud Foundry

Esto es opcional, puede realizar la mayoría de las acciones a través de la consola web desde el paso 1. Las utilidades de CLI se pueden descargar desde https://new-console.ng.bluemix.net/docs/starters/install_cli.html y debe instalar ambos El cli bluemix y el cli nube.

Paso 2a: Conectar a Bluemix

```
$ bluemix api https://api.ng.bluemix.net
```

Paso 2b: Iniciar sesión en Bluemix

```
$ bluemix login -u username -o org_name -s space_name
```

Debe especificar su nombre de usuario, nombre de organización y nombre de espacio que puede obtener desde la consola web en el Paso 1.

Después de iniciar sesión correctamente, ahora ha configurado bluemix. Puede obtener más información sobre la CLI y obtener complementos adicionales en <http://clis.ng.bluemix.net/ui/home.html>

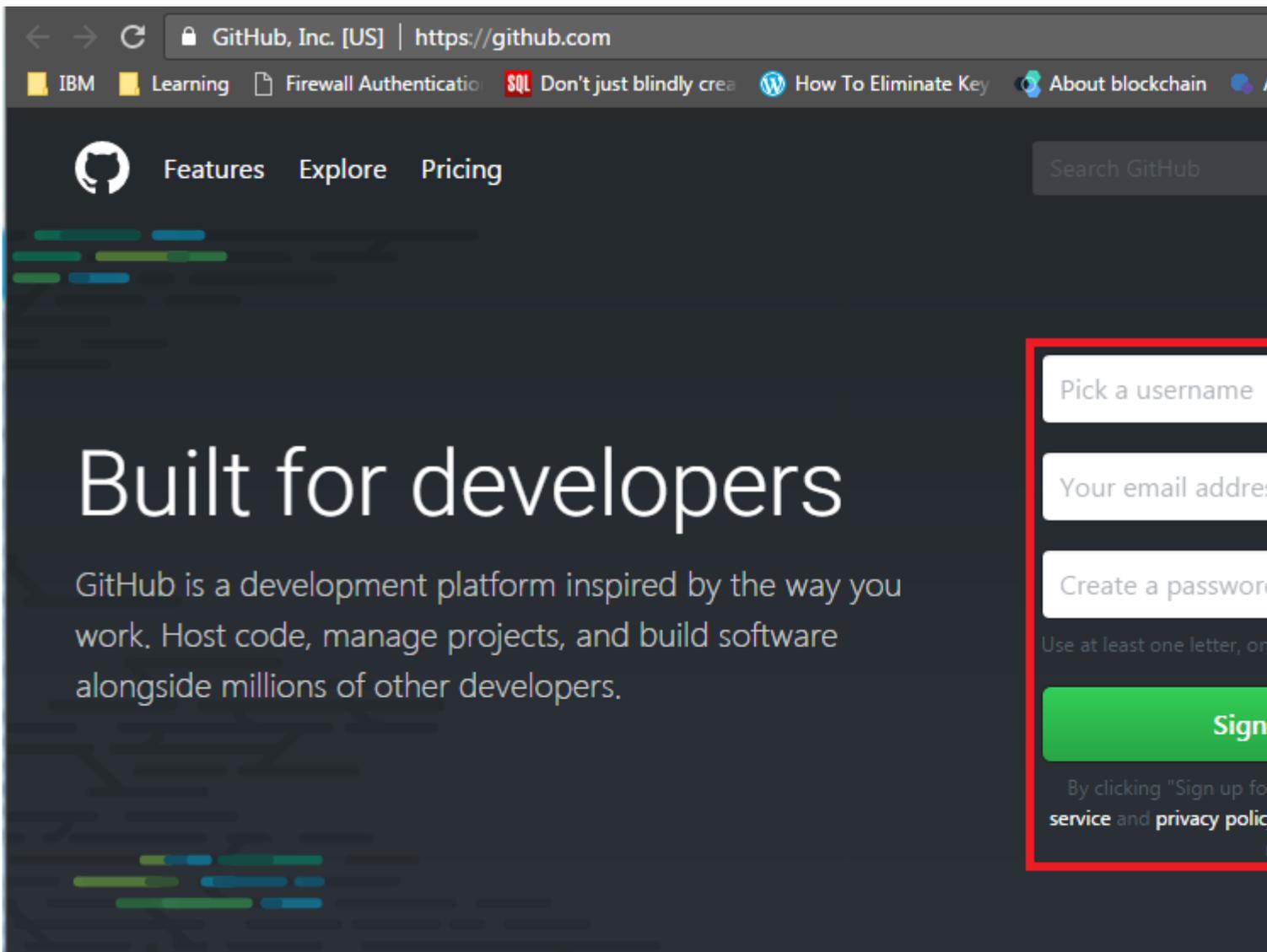
Implementación continua de RESTful API (Node.js) a través de GitHub utilizando IBM Bluemix Toolchain

Paso 1: Crea tu cuenta de GitHub

Si ya tiene una cuenta de GitHub, continúe con el Paso 2. De lo contrario, siga estos pasos:

1.a Ir a la página de [Github](#) .

1.b Ingrese su nombre de usuario deseado, su dirección de correo electrónico y luego su contraseña deseada. Luego, haga clic en el botón **Registrarse en GitHub** .



Paso 2: Crea tu cuenta de IBM Bluemix

Consulte [Bluemix Get Started](#) sobre cómo crear una cuenta o siga el ejemplo anterior sobre cómo crear su cuenta Bluemix en esta página.

Paso 3: Implementa tu paquete de API Rest

en GitHub

3.a Si no sabe cómo usar Git, lea el Tutorial de Git o puede usar [SourceTree](#), una implementación de Git basada en GUI. Lea el [Tutorial de SourceTree](#) para obtener más información sobre SourceTree.

3.b Suba su código de API Rest a GitHub usando el comando **git push** . En contraste, puede bifurcar o clonar mi [API Rest](#) en GitHub y luego aplicar sus propios cambios según sea necesario y luego cargar el código en su cuenta de GitHub. Nota: Mi API Rest de muestra contiene una implementación de ejemplo de MongoDB y APICache.

Paso 4: implemente su repositorio GitHub en IBM Bluemix usando Toolchain (herramienta de implementación continua)

4.a Inicie sesión en su cuenta de [Bluemix](#) .

Sign in to IBM

Enter IBMid or email

[Forgot your IBMid?](#)

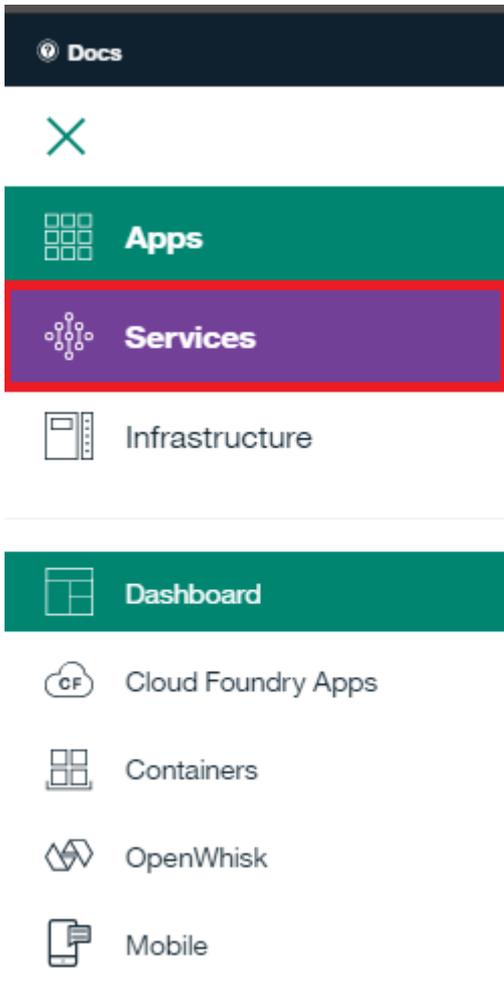
Continue

New? [Create an IBMid.](#)

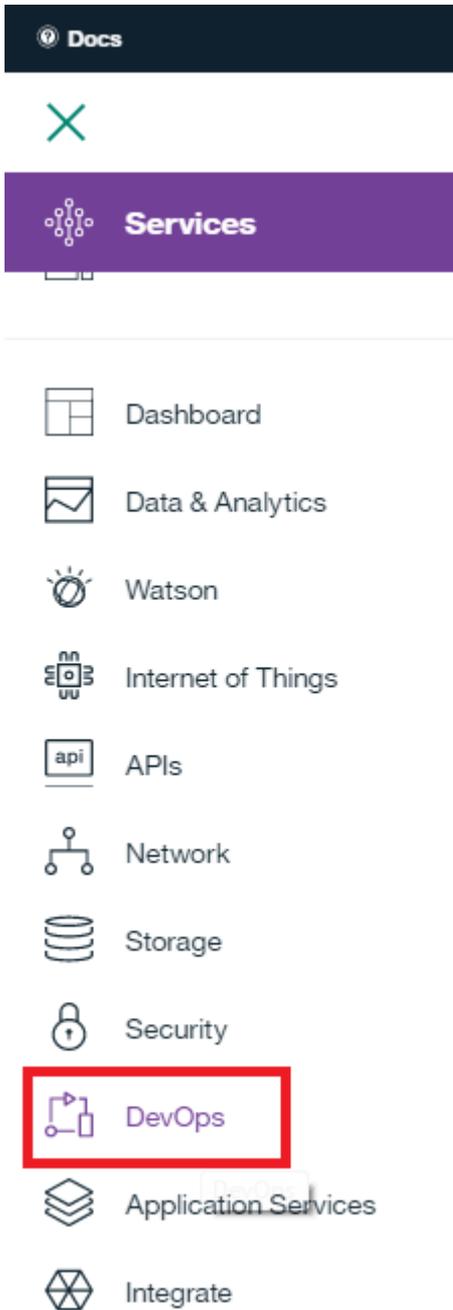
4.b Haga clic en Menú para mostrar otros elementos.



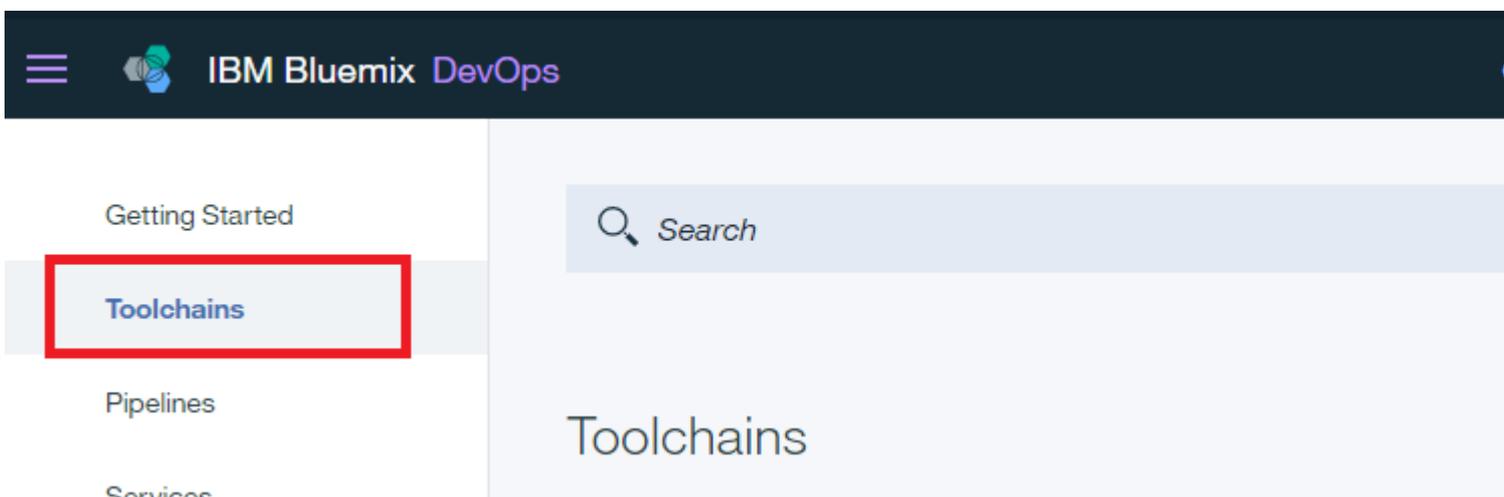
A continuación, seleccione "Servicios".



Y luego, haga clic en "DevOps".



4.c Seleccionar Toolchains. Luego, haga clic en el botón **Crear una cadena de herramientas** .



4.d Seleccionar otras plantillas

The screenshot shows the IBM Bluemix DevOps interface. At the top, there is a dark blue header with the IBM logo and the text "IBM Bluemix DevOps". Below the header, the main content area is light blue. The title "DevOps Insights Templates" is centered. There are three template cards visible. The first card is "Delivery Insights with IBM UrbanCode Deploy" with a lightbulb icon containing code symbols and a plus sign. The second card is "Developer Insights and Team Dynamics with GitHub and JIRA" with a lightbulb icon containing code symbols and a plus sign. The third card is "Build your own toolchain" with a circular icon containing various colored dots and a plus sign. This third card is highlighted with a red rectangular border. Below the "DevOps Insights Templates" section, there is a section titled "Other Templates" which contains the "Build your own toolchain" card.

to deploy.

DevOps Insights Templates

-  **Delivery Insights with IBM UrbanCode Deploy**
View deployment metrics.
-  **Developer Insights and Team Dynamics with GitHub and JIRA**
Analyze GitHub source code, and GitHub or JIRA issues.
-  **Deployment Insights with GitHub and JIRA**
Use analytics to optimize your deployment process.

Other Templates

-  **Build your own toolchain**
For advanced users, create your toolchain from scratch.

4.e Introduzca deseada **Toolchain Nombre**, tenga en cuenta que esto mostrará en su dirección URL de la API.



This toolchain has no preconfigured tools. If you are already familiar with toolchains, you can set up your own toolchain. To get started, click **Create**.

TEMPLATE INFO

GIT URL

<https://github.com/open-toolchain/en>

GIT BRANCH

master

Organization

Toolchain Name

4.f Espere unos segundos a que Bluemix cree su cadena de herramientas. Luego, haga clic en **Agregar una herramienta** .

← Toolchains

Overview

Connections

Manage

sample-rest-api-bluemix-toolch...

Your toolchain is ready!



Quick start: You can now add tool integrations. For step-by-step instructions, see the [Quick start](#) page for this toolchain.



No Tool Integrations

4.g Elige **GitHub**



[Learn More](#)



Alert Notification

Never miss critical issues.

IBM

Experimental



Artifactory

Store build artifacts in Artifactory repository.

Third Party



Availability Monitoring

Test, monitor, and improve your application as you build it.

IBM



Delivery Pipeline

Automate your build and deployment process and more.

IBM



DevOps Insights

Use analytics to determine whether to deploy.

IBM

Beta



Eclipse Orion

A browser-based IDE for cloud development.

IBM



Git Repos and Issue Tracking

IBM hosted repos and issue tracking based on GitLab

IBM

Experimental



GitHub

Store and manage your code the easy way.

Third Party

4.h Ahora, necesitamos vincular nuestro repositorio existente en GitHub a esta cadena de herramientas. En **Tipo de repositorio** , seleccione **Existente** . Luego, elija la URL del repositorio correcta para su fuente de Rest Rest. Después de eso, depende de usted si desea realizar un seguimiento de los cambios del repositorio a través de Toolchain. Para hacerlo, marque la casilla de verificación Realizar **seguimiento de cambios de código** .



← Tool Catalog

Configure the Integration



GitHub

Store your source code in a new or existing repository on GitHub.com and engage in social coding through wikis, issue tracking, and pull requests.

Third Party

[View Docs](#)

PROVIDER **IBM**
TOOLCHAIN **sample-rest-api-bluemix-toolchain**

Repository type:

Existing

Link to the repository that is specified in the Source repository URL field

Source repository URL:

`https://github.com/ariescamitan/node-api-with-express`

Track deployment of code changes

4.i Ahora, lo que necesitamos es algo para construir e implementar nuestro repositorio API Rest de forma automática. Así que tenemos que añadir otra herramienta.

sample-rest-api-bluemix-toolchain

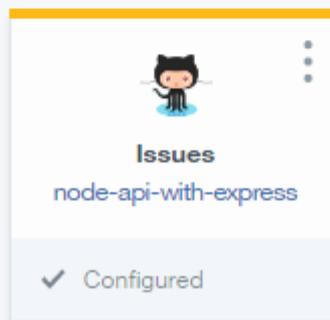
Your toolchain is ready!



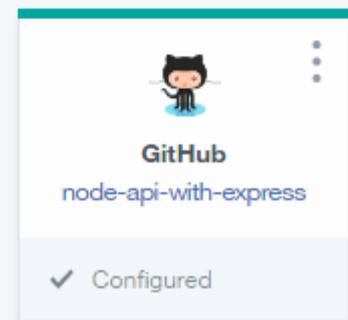
Quick start: You can now add tool integrations. For step-by-step instructions, see the documentation for this toolchain.

THINK

CODE



Issues
node-api-with-express
✓ Configured



GitHub
node-api-with-express
✓ Configured

4.j Agreguemos el Pipeline de Entrega.



properly connect the tool integration with the toolchain, you might need to provide account information.

[Learn More](#)

Tool Integrations



Alert Notification

Never miss critical issues.

IBM

Experimental



Artifactory

Store build artifacts in Artifactory repository.

Third Party



Availability Monitoring

Test, monitor, and improve your application as you build it.

IBM



Delivery Pipeline

Automate your build and deployment process and more.

IBM



DevOps Insights

Use analytics to determine whether to deploy.

IBM

Beta



Eclipse Orion

A browser-based IDE for cloud development.

IBM

Luego, asígnele el nombre **Build and Deploy** .



Tool Catalog

Configure the Integration



Delivery Pipeline

The Delivery Pipeline service automates continuous deployment.

IBM

[View Docs](#)

PROVIDER	IBM
TOOLCHAIN	sample-rest-api-bluemix-toolchain

Pipeline name:

Build and Deploy

Show apps in the View app menu

Después de unos segundos, aparecerá el recién agregado Pipeline de entrega. Pinchalo.

- ← Toolchains
- Overview
- Connections
- Manage

sample-rest-api-bluemix-toolchain

Your toolchain is ready!
✔ **Quick start:** You can now add tool integrations. For step-by-step instructions for this toolchain.

THINK

Issues
node-api-with-express
✔ Configured

CODE

GitHub
node-api-with-express
✔ Configured

4.k Necesitamos agregar etapas a nuestro Pipeline de entrega. Haga clic en **Agregar etapa**.

← Toolchain

Build and Deploy | Delivery Pipeline

Add Stage +

4.l Crea la **etapa de compilación** en la que construiremos automáticamente nuestro paquete. Siga los pasos en la figura a continuación:



Build

1. Name it "Build".

INPUT

JOBS

ENVIRONMENT PROPERTIES

5. Click "JOBS" tab.

Input Settings

Input Type

2. Choose SCM Repository.

SCM Repository

Git Repository

node-api-with-express

3. Verify the Git Repository.

Git URL

https://github.com/ariescamitan/node-api-with-express.git

Branch

master

Stage Trigger



Run jobs whenever a change is pushed to Git



Run jobs only when this stage is run manually

4. Select "Run jobs whenever a change is pushed to Git."

SAVE

CA

Necesitamos agregar un **TRABAJO** que ejecute el **comando de instalación npm** .



Build and Deploy | Stage Configuration

Build

INPUT

JOBS

ENVIRONMENT PROPERTIES

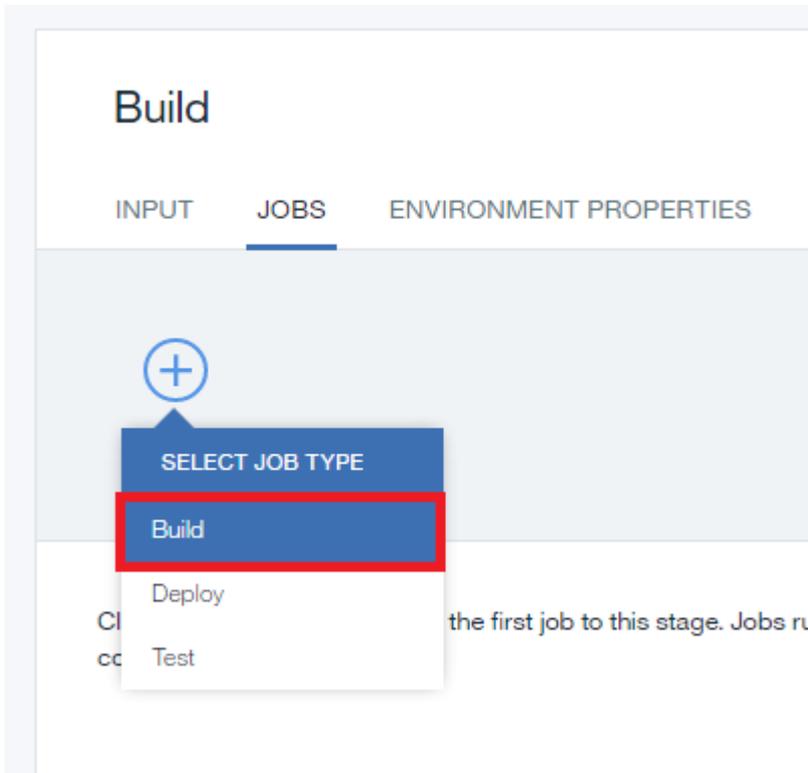


Click "ADD JOB" above to add the first job to this stage. Jobs runs sequentially in a stage, and each job runs in a container environment.

SAVE

CA

Seleccione la opción **Construir** .



Luego, seleccione **npm** bajo el **Tipo de Generador** . Y luego, bajo el **comando Build Shell**, configure automáticamente que ejecutará el comando **npm install** .

Build

REMOVE

Build Configuration

Builder Type

npm

Build Shell Command

```
#!/bin/bash
# The default Node.js version is 0.10.40
# To use Node.js 0.12.7, uncomment the following line:
#export PATH=/opt/IBM/node-v0.12/bin:$PATH
# To use Node.js 4.2.2, uncomment the following line:
#export PATH=/opt/IBM/node-v4.2/bin:$PATH
npm install
```

Don't have a build script? Create a new one from a template. [+ ADD](#)

Working Directory

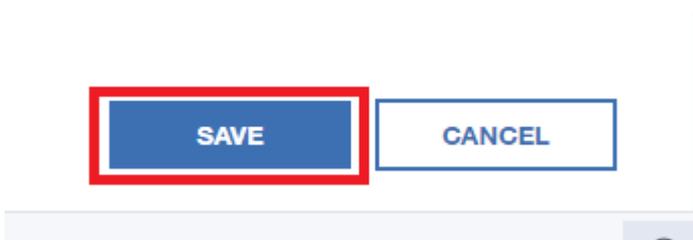
Build Archive Directory

 Enable Test Report

Run Conditions

 Stop running this stage if this job fails

Asegúrese de hacer **click en el botón Guardar** para guardar los cambios.



4.m Después de crear la etapa de compilación, ahora necesitamos crear la etapa de **implementación** . Ve y agrega otra etapa haciendo clic en **Agregar etapa** .

Build and Deploy | Delivery Pipeline

The screenshot shows a 'Build' stage in a delivery pipeline. The stage is currently 'NOT RUN'. The interface includes a play button and a settings gear icon. Below the stage name, there is a section for 'LAST INPUT' with a 'Git URL' link and the text 'Not yet run'. A 'JOBS' section contains a single job named 'Build' with a status of 'Not yet run' and a 'View logs and history' link. The 'LAST EXECUTION RESULT' section shows 'No results'. To the right of the stage card, a blue button labeled 'Add Stage +' is highlighted with a red border.

Siga los pasos definidos en la siguiente figura:

Build and Deploy | Stage Configuration

The screenshot shows the 'Stage Configuration' interface. At the top, the stage name 'Deploy' is highlighted with a red box and labeled '1. Rename to "Deploy".'. Below it are three tabs: 'INPUT', 'JOBS', and 'ENVIRONMENT PROPERTIES'. The 'JOBS' tab is highlighted with a red box and labeled '4. Click "JOBS" tab.'. Under the 'Input Settings' section, the 'Input Type' dropdown is set to 'None', which is highlighted with a red box and labeled '2. Choose "None".'. Under the 'Stage Trigger' section, the radio button for 'Run jobs when the previous stage is completed' is selected and highlighted with a red box, labeled '3. Leave it as is.'. At the bottom right, there are 'SAVE' and 'CANCEL' buttons.

Luego, en la pestaña **TRABAJOS** , haga clic en **Agregar trabajo** .

Deploy

INPUT

JOBS

ENVIRONMENT PROPERTIES



Click "ADD JOB" above to add the first job to this stage. Jobs runs sequentially in a stage, and each job runs in a clean container environment.

Y luego, seleccione la opción **Implementar** .

Deploy

INPUT

JOBS

ENVIRONMENT PROPERTIES



SELECT JOB TYPE

Build

Deploy

Click "ADD JOB" above to add the first job to this stage. Jobs runs sequentially in a stage, and each job runs in a clean container environment.

En **Tipo de implementación**, seleccione **Cloud Foundry** (Servicio de nube predeterminado de IBM Bluemix).

Deploy

INPUT

JOBS

ENVIRONMENT PROPERTIES



Deploy



ADD JOB

Deploy

REMOVE

Deploy Configuration

Deployer Type



Cloud Foundry

Target



US South - https://api.ng.bluemix.net

Organization



En este caso, seleccioné el espacio **dev** . Para un proyecto real, es posible que desee seleccionar un mejor espacio que pueda manejar su uso de producción real. Después, haga clic en el botón **Guardar** .

Space (i)

dev

Application Name (i)

Build and Deploy

Deploy Script (i)

```
#!/bin/bash
cf push "${CF_APP}" -c "node server.js"
```

Run Conditions (i)

Stop running this stage if this job fails

SAVE
CANCEL

4.n Ahora todas las etapas están configuradas. Por defecto, todas las etapas están en estado de **parada** . Necesitamos **ejecutar** todas nuestras etapas haciendo clic en los botones **Jugar** como.

Build 1. ▶ ⚙️

STAGE RUNNING...

LAST INPUT [Git URL](#)

Last commit by Aries B. Camitan 13d ago
[Added MongoDB on readme file.](#)

JOBS [View logs and history](#)

Build Running

LAST EXECUTION RESULT

No results

Deploy 2. ▶ ⚙️

STAGE NOT RUN

INPUT

No input configured

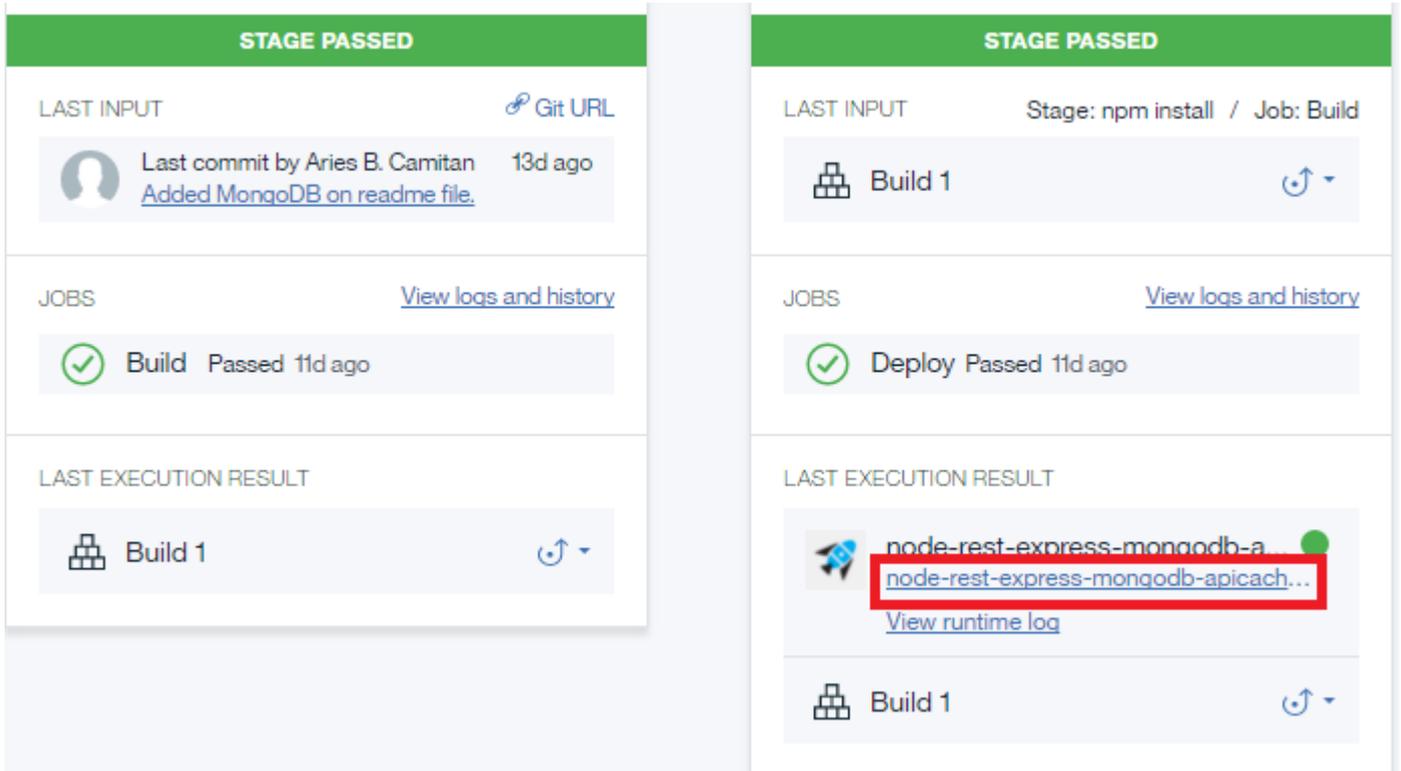
JOBS [View logs and history](#)

Deploy Not yet run

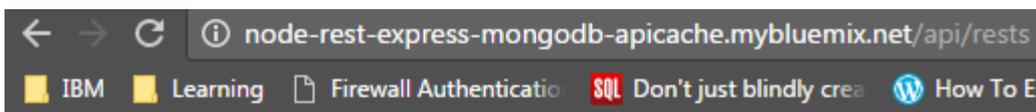
LAST EXECUTION RESULT

No results

4.o Si ve una figura similar a continuación, significa que configuró e implementó exitosamente nuestra **Implementación Continua** de nuestras API RESTful a través de IBM Bluemix usando ToolChain.



Intente hacer clic en el enlace como se define en la figura anterior para ver si la API se está ejecutando como se esperaba.



```
[
- {
  _id: "58b69aa7957c5a0528ab8fbc",
  name: "Virgo",
  __v: 0
},
- {
  _id: "58b7d844f5bc8825bcca9327",
  name: "Ram",
  __v: 0
},
- {
  _id: "58b7d97b6a80521c78526505",
  name: "Josephine",
  __v: 0
}
]
```

Paso 5: Probar la API Rest con PostMan

Postman es una potente herramienta de prueba de API y se puede instalar a través de Google Chrome como una extensión. Asegúrese de que en su prueba siga la convención de ruta que definió en su código Express. Para saber más sobre cómo utilizar la herramienta Postman, consulte el [Blog de Postman](#).

Renuncia:

No poseo ningún enlace externo dado en este ejemplo. Crédito a quienes poseen esos enlaces.

Lea **Empezando con ibm-bluemix en línea**: <https://riptutorial.com/es/ibm-bluemix/topic/1110/empezando-con-ibm-bluemix>

Capítulo 2: Mobile Analytics para Bluemix

Introducción

El servicio Mobile Analytics en IBM Bluemix le permite analizar el uso de sus aplicaciones móviles. Instale su aplicación con el SDK de Mobile Analytics y obtenga información sobre sus usuarios, el comportamiento de su aplicación y las alertas de configuración en eventos específicos

Examples

Introducción a las aplicaciones iOS Swift y Mobile Analytics para Bluemix

1. Cree una instancia de [Mobile Analytics para Bluemix](#) .
2. [Agregue el SDK de servicios móviles de Bluemix](#) a su proyecto de iOS.
3. Después de instalar el SDK, agregue estas declaraciones de importación en la parte superior de su archivo AppDelegate.swift:

```
import BMSCore
import BMSAnalytics
```

4. A continuación, deberá iniciar y enviar análisis móviles en su método `didFinishLaunchingWithOptions`:

```
func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[NSObject: AnyObject]?) -> Bool {

    // Set api key, Bluemix region. These are available in the Mobile Analytics console
    after you create an instance
    let api_key="xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx"
    let bmixRegion=BMSClient.Region.usSouth

    //Provide a name for your app that will appear in the Mobile Analytics console.
    let appName="MyAppName"

    //Initialize Mobile Analytics in your Bluemix region
    BMSClient.sharedInstance.initialize(bluemixRegion: bmixRegion)
    Analytics.initialize(appName: appName, apiKey: api_key, hasUserContext: false,
deviceEvents: DeviceEvent.LIFECYCLE)

    //Send analytics
    //Analytics.send()

    //Alternately send analytics and log to your xCode console
    Analytics.send { (response: Response?, error: NSError?) in
        if response?.statusCode == 201 {
            print("Successfully sent analytics: \(response?.responseText)")
        }
        else {
            print("Failed to send analytics: \(response?.responseText). Error:
\((error?.localizedDescription)")
```

```
    }  
  }  
  
  return true  
}
```

5. Agregue `Analytics.send()` al código de su aplicación en cualquier lugar donde desee que los análisis almacenados en la aplicación se envíen al servicio de Mobile Analytics para Bluemix.
6. Ejecute su aplicación. Verá un nuevo usuario y una nueva sesión en su consola de análisis.

Lea Mobile Analytics para Bluemix en línea: <https://riptutorial.com/es/ibm-bluemix/topic/6680/mobile-analytics-para-bluemix>

Capítulo 3: Unión

Examples

Acceso a credenciales a través de la variable de entorno VCAP_SERVICES

Cuando vincula un servicio a las credenciales de su aplicación, estará disponible a través de la variable de entorno VCAP_SERVICES.

Esta variable de entorno contiene JSON que contiene las credenciales para todos los servicios enlazados.

Ejemplo de variable de entorno VCAP_SERVICES

```
{
  "push-reappt": [
    {
      "name": "Reappt from Push Technology",
      "label": "push_reappt",
      "plan": "reappt:pushtechnology:free",
      "credentials": {
        "principal": "service-binding-abcd1234",
        "credentials": "XYZlmnop456",
        "host": "sniffingitchyPythagoras.eu.bluemix.reappt.io",
        "port": 443
      }
    }
  ]
}
```

A continuación, puede acceder a estas credenciales a través de su aplicación.

Javascript

En una aplicación Node puedes hacer lo siguiente:

```
var reappt_credentials = JSON.parse(process.env.VCAP_SERVICES) ["push-reappt"] [0].credentials;

diffusion.connect({
  host : reappt_credentials.host,
  principal : reappt_credentials.principal,
  credentials : reappt_credentials.credentials
}).then(connected, error);
```

Java

En una aplicación Java, lo mismo podría hacerse de la siguiente manera:

```
private static final JsonParser PARSER = new JsonParser();
private static final JsonObject VCAP_SERVICES =
PARSER.parse(System.getenv("VCAP_SERVICES")).getAsJsonObject();
```

```
private static final JsonObject REAPPT_CREDENTIALS = VCAP_SERVICES.getAsJsonArray("push-  
reappt").get(0)  
    .getAsJsonObject().getAsJsonObject("credentials");  
protected static final String HOST =  
REAPPT_CREDENTIALS.getAsJsonPrimitive("host").getAsString();  
protected static final String PRINCIPAL =  
REAPPT_CREDENTIALS.getAsJsonPrimitive("principal").getAsString();  
protected static final String CREDENTIALS =  
REAPPT_CREDENTIALS.getAsJsonPrimitive("credentials").getAsString();
```

Lea Unión en línea: <https://riptutorial.com/es/ibm-bluemix/topic/2542/union>

Creditos

S. No	Capítulos	Contributors
1	Empezando con ibm-bluemix	ariescamitan , Community , Joshua Smith , ralphearle
2	Mobile Analytics para Bluemix	mdarden , Srikanth K M
3	Unión	Freddie Coleman