Cloud Computing V2

IBM Cloud catalog

One of the main parts of the IBM Cloud console is the catalog that presents IBM Cloud offerings and services. The IBM Cloud catalog contains more than 190 unique services. You can check the current list of services in the catalog at the following link:

https://cloud.ibm.com/catalog

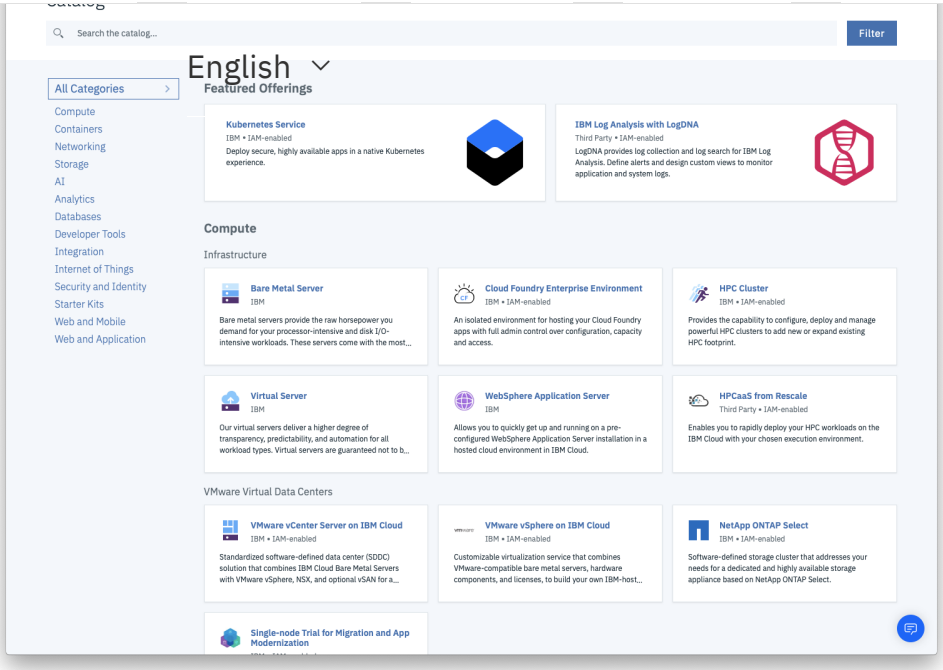


Figure 1. IBM Cloud Catalog

Working with resources

A resource is anything that can be created, managed, and contained from the list of services, run times, and templates that are available on the IBM Cloud catalog.

Some examples include:

- Cloud Foundry Apps
- Service instances

- Container clusters

- Virtual servers

English

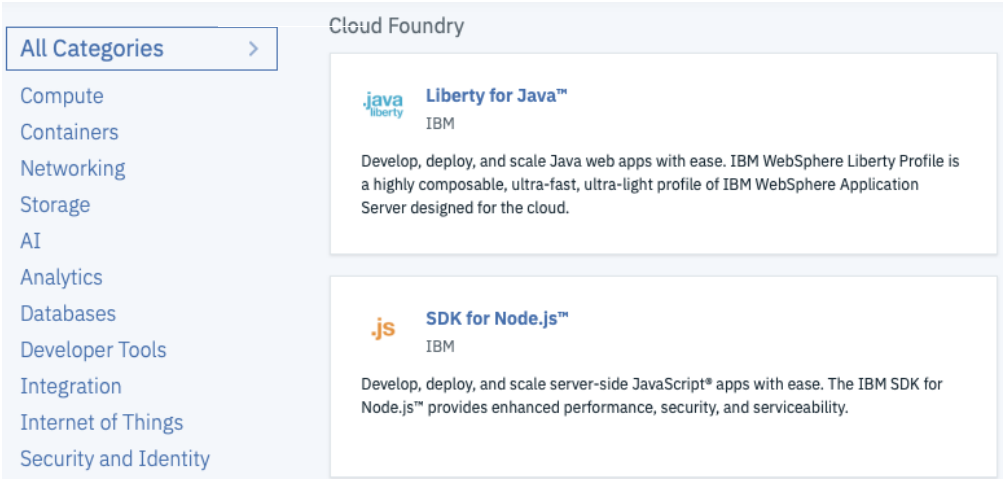


Figure 2. IBM Cloud offerings

IBM Cloud catalog: Categories

The IBM Cloud catalog displays your IBM Cloud resources worldwide. Resources in the catalog can include services, infrastructure, and templates. You can use the IBM Cloud CLI to search the catalog, view details about catalog resources, and manage the visibility of your catalog resources.

The IBM Cloud catalog provides multiple categories such as:

- Compute
- Containers
- Networking

- Storage

- Analytics
- Databases

To create a resource, click **Create** from your resource list. The catalog window appears. When you select a tile from the catalog, you can see where the resource is available. Not every resource that is listed in the catalog is available in every region.

After you click the tile for the resource that you want create, you can select the location in which you want to deploy:

- For Cloud Foundry resources, you can select a specific region and then you select the org and space to which you assign to a service instance.
- For resources that are managed by IBM Cloud IAM, you select a location in which to deploy. Then, you select an Resource Group to which to assign the service instance.

IBM Cloud catalog: Starter kits

- A *starter kit* is a production-ready pattern that can be integrated with a set of services to generate a production-ready asset that can be deployed directly into a DevOps pipeline and a Kubernetes cluster. Starter kits are great for dynamically assembling a skeleton production application in the language of your choice that is ready for cloud deployment.
- A starter kit contains metadata that describes what the kit is and does. It also contains information that informs IBM Cloud what to produce. The output is production-ready and can be iterated on for further enhancements based on IBM Cloud best practices. Starter kit content is not as complex as a demonstration and not as trivial as a snippet or sample. Apps are dynamically created based on the developer's requirements.

- Each starter kit includes a language, a framework, and a pattern for a specific use case. You can reuse code rather than reinvent it. If a starter kit requires specific services, auto-provisioned services are available so that instances for

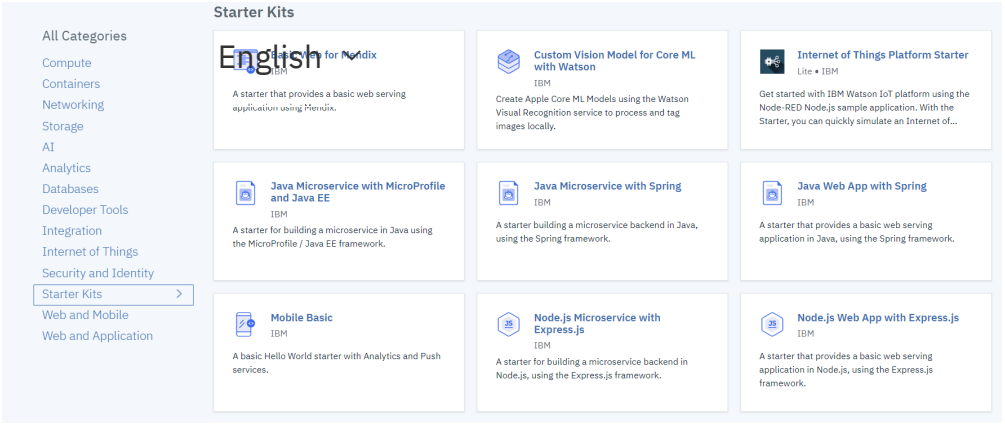


Figure 3. IBM Cloud Starter Kits

Examples of starter kits:

- Node.js Web App with Express.js

This starter kit comes pre-configured as a web app with Express.js that uses the Node.js run time.

- Node.js Microservice with Express.js

This starter kit comes pre-configured as a microservice with Express.js. It allows the user to use the IBM Cloud Developer Tools CLI to run and debug locally, then deploy to Kubernetes, Cloud Foundry, or a DevOps Pipeline.

- Java Microservice with Spring

This starter kit comes pre-configured as a Java microservice application that uses Spring Boot. It allows the user to use the IBM Cloud Developer Tools CLI to deploy to Kubernetes, Cloud Foundry, or a DevOps Pipeline.

- Java Microservice with MicroProfile and Java EE

- Internet of Things Platform Starter

This starter kit includes the IBM Watson IoT Platform that uses the Node-RED Node.js sample application. With this starter kit, the user can quickly simulate an IoT device, create cards, generate data, and begin analyzing and displaying data in the Watson IoT Platform dashboard.

- Mobile Basic

This Mobile starter app pre-provisions push services, in addition to giving a code for the mobile app as a starting point.

IBM Cloud catalog: Creating a starter kit

The figure below shows an example of the Node.js Web App with Express.js starter kit:

1. Select the Node.js Web App with Express.js starter kit from the catalog, then click Create app.
2. Fill the fields that are required in the form as the app name and the resource group, which are set to Default. Then, click Create.
3. You are redirected to the starter kit details page. You can download the code by clicking Download code.
4. There is the option of deploying this application on the Cloud Foundry runtime, IBM Kubernetes Service, or Virtual Server, and then configure the continuous delivery pipeline.
5. The starter kit also can be connected to several services.

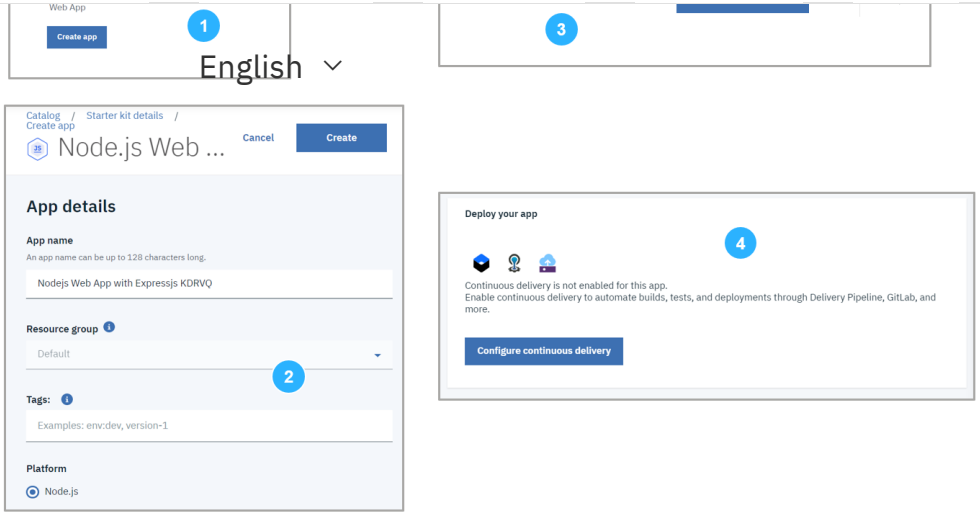


Figure 4. Example of creating a Node.js Starter Kit

IBM Cloud catalog: Cloud Foundry Apps

IBM Cloud provides Cloud Foundry runtimes that are used to deploy applications of the corresponding language on top of the runtime such as

- Liberty for Java
- SDK for Node.js
- ASP.NET core
- Runtime for Swift

What is Cloud Foundry

remain carefully coordinated with any attached services, which result in quick, consistent, and reliable iteration of applications.

Key benefits of Cloud Foundry:

- Choose your own language: IBM Cloud Foundry includes run times for Java, Node.js, PHP, Python, Ruby, Swift, and Go. Cloud Foundry community build packs are also available. Combined with DevOps services, the application runtimes enable a delivery pipeline that automates much of the iterative development process.
- Fault tolerant: Runtimes facilitate developing applications as stateless processes that quickly start and stop, replicate if an instance fails, and duplicate if sustained or increased performance requires.
- Extend apps with services: Run times link IBM Cloud services to applications as endpoints, giving any instance of an application embedded knowledge of how to manage relevant calls and data. In fact, runtimes manage all linked resources this way: SDKs, APIs (whether made available as cloud services or exposed from within a traditional enterprise as custom services), and also applications themselves when used as resources by other applications.
- Access control: Fine-grained assignment and distribution of compute capacity to development teams.
- Automatic placement: Applications are automatically placed across multiple data center Pods for maximum reliability.
- Automatic Health Management: Crashing applications restart automatically.
- Automatic routing: Internet reachable routes are automatically created for your applications.
- High availability: Supports full high availability for high application availability.
- Automatic deployment scaling: The Auto-Scaling for IBM Cloud service enables you to increase or decrease automatically the compute capacity of your application to adjust rapidly to dynamic loading needs.

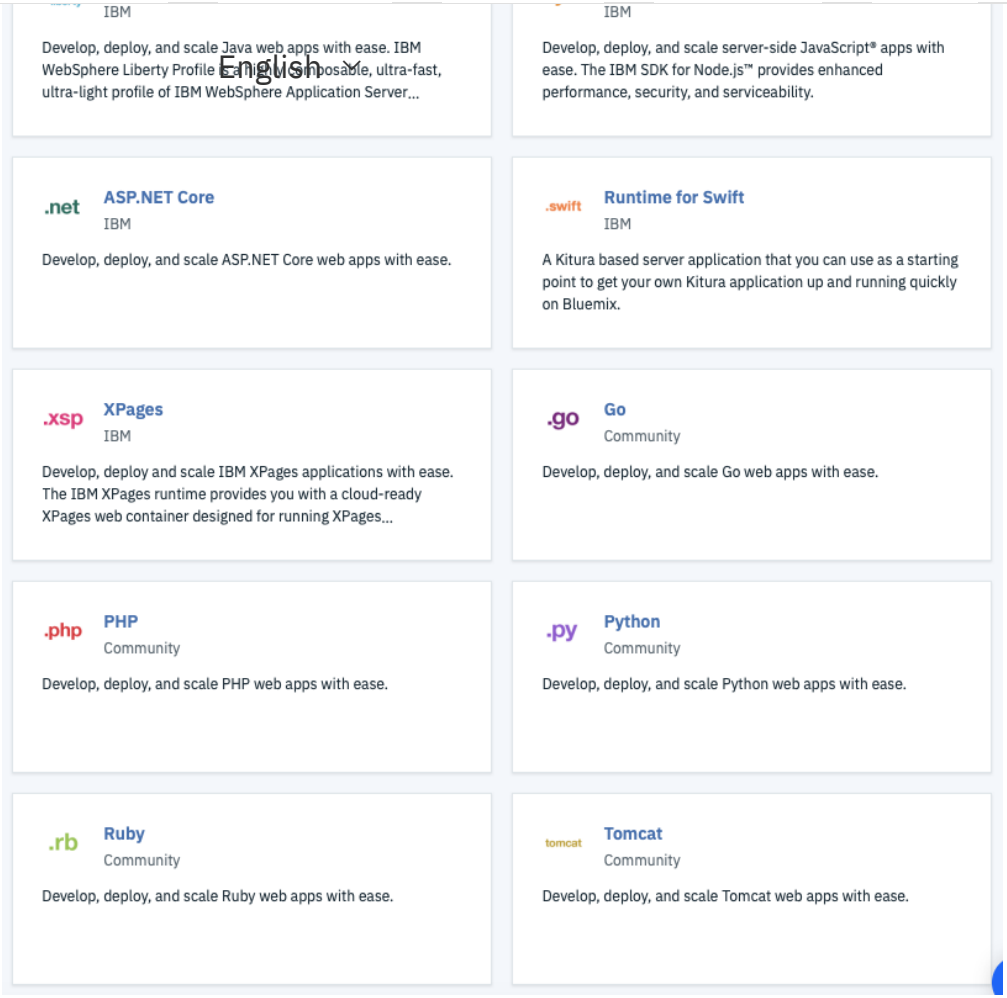


Figure 5. IBM Cloud Foundry Apps

IBM Cloud catalog: AI

IBM Cloud provides various AI services that are built on top of IBM Watson, which is the AI offering from IBM. Watson

uses Natural Language Processing (NLP), computer vision, and machine learning technologies to reveal insights from

Examples of AI services to enrich applications with cognitive features are:

- Chat bots
- Machine learning
- Language translator
- Natural language understanding

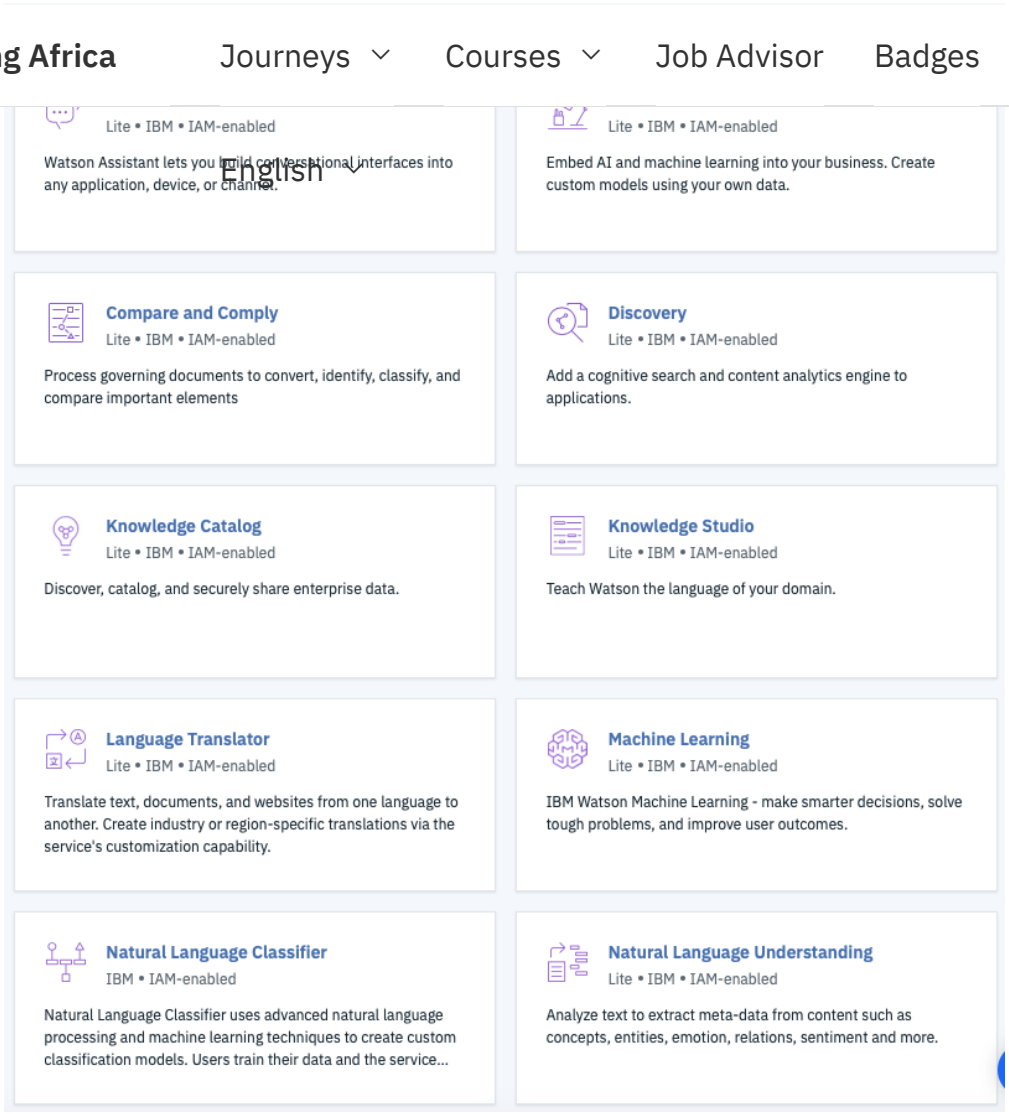


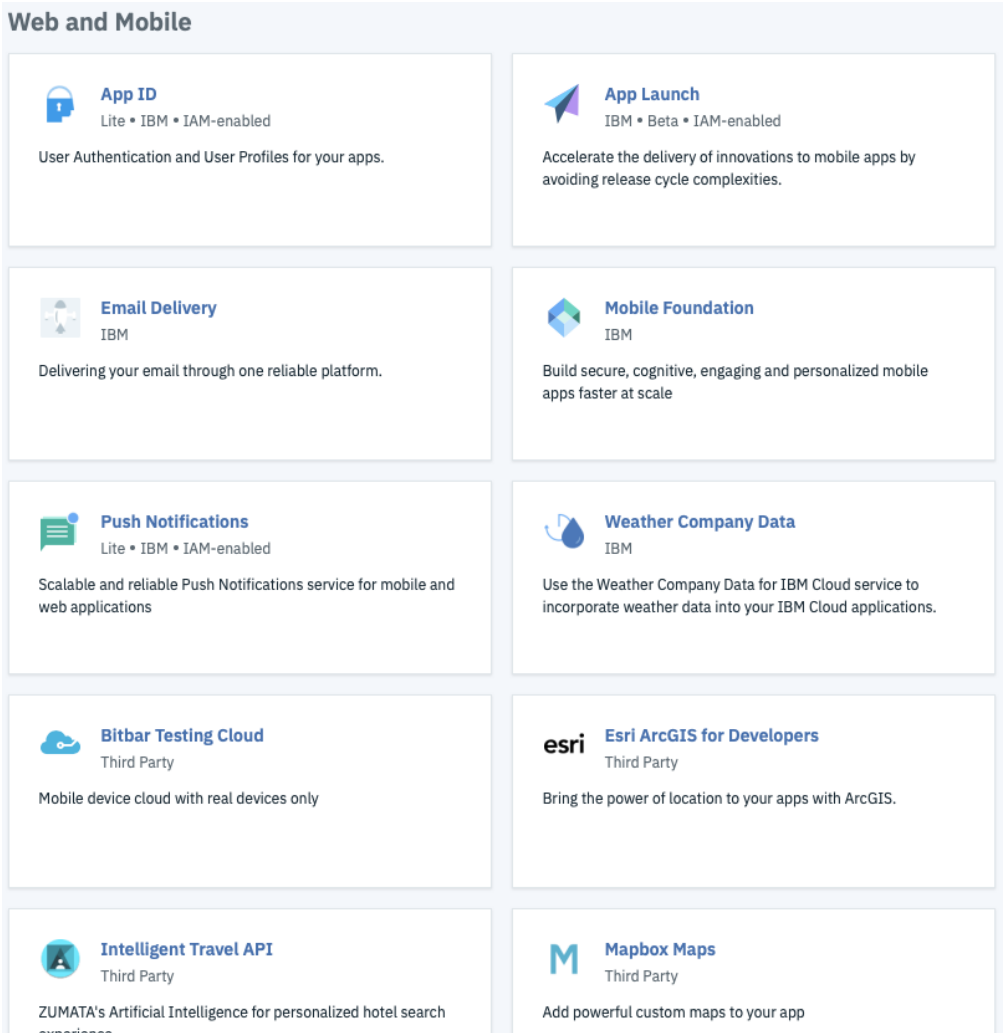
Figure 6. IBM Cloud AI Services

IBM Cloud catalog: Web and mobile

IBM Cloud offers a set of services that can ease the common tasks that are needed in building a mobile app. These

IBM Cloud offers a set of services that can ease the common tasks that are needed in building a mobile app. These services take the drudgery of routine tasks away from the app developer and provide compact SDKs with which the

IBM Cloud Mobile Foundation provides an integrated set of back-end capabilities for mobile, web, and progressive web apps (PWAs). Developers can choose to use front-end tools or frameworks of their choice and leverage the rich set of back ends that is provided by the Cloud Mobile Foundation service. Cloud Mobile Foundation SDKs are available for Cordova, iOS, Android, Xamarin, Windows 10, React Native, and Mobile Web.



experience.

English

IBM Cloud catalog: Developer tools

IBM Cloud Developer tools provide developer toolchains and APIs to help automate the tasks of developing and deploying your app. You can perform DevOps manually with simple apps, but the need for automation increases quickly as app complexity increases, and toolchain automation is necessary for continuous delivery.

The core component of a DevOps toolchain is a version control repository like GitHub. More tools might include backlog tracking, delivery pipelines, an integrated development environment (IDE), and monitoring like IBM Cloud DevOps Insights.

Examples of such tools and processes are the core set of tools that IBM Cloud Continuous Delivery provides for any DevOps toolchain: Git Repos and Issue Tracking, Delivery Pipeline, and Eclipse Orion Web IDE. Git Repos and Issue Tracking is based on the GitLab Community Edition, and offers planning boards and source code collaboration through merge requests. The Delivery Pipeline orchestrates build, test, and deployment jobs across multiple environments as changes progress from the developer to production. Applications can be deployed in minutes to the Cloud Foundry environment or to a Kubernetes cluster on IBM Cloud, to either public or private clouds.

An open toolchain integrates more tools around Cloud Continuous Delivery, such as Slack, Atlassian JIRA, Sonatype Nexus, JFrog Artifactory, Sauce Labs, PagerDuty, IBM Cloud Availability Monitoring, IBM Cloud Alert Notification, IBM Vulnerability Advisor, and IBM Globalization Pipeline. You can also substitute other tools for the Cloud Continuous Delivery capabilities, including GitHub, GitHub Enterprise, and Jenkins. Developers can also use their favorite IDEs and editors, such as Visual Studio Code, Eclipse, and more.

Code repositories, issue tracking systems, build systems, and deployment systems represent a wealth of data that can be used to help you deliver apps more efficiently and effectively. IBM Cloud DevOps Insights uses big data analysis to provide valuable insights to executives, managers, and developers. DevOps Insights aggregates and analyzes data from

your DevOps toolchain to advise you about the risk of deploying specific changes, and areas to improve both your

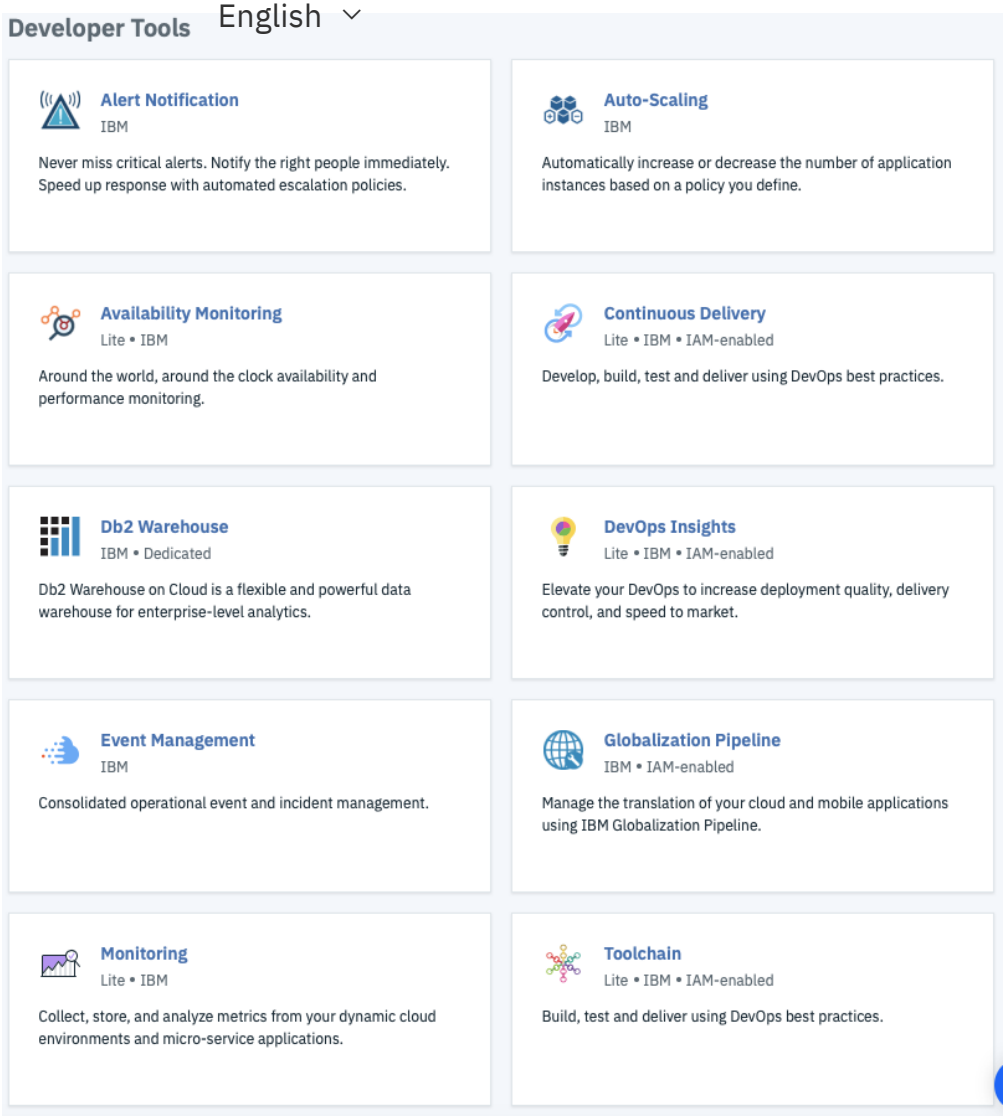


Figure 7. IBM Cloud Developer Tools

IBM Cloud Functions is an event-driven compute platform, also referred to as serverless computing or as FaaS, that runs code in response to events or direct invocations.

What is an action

An action is a small piece of code that can be invoked or set to run automatically in response to an event. In either case, each run results in a record that is identified by a unique activation ID. The input and the result of an action can be seen as key-value pairs. The key is a string and the value is a valid JSON value. An action can be written in the language of your choice and provided to the service as either source code or a Docker image. The action code runs when it is directly invoked by the Cloud Functions API, CLI, or iOS SDK. An action can automatically respond to events from IBM Cloud or third-party services.

Why would you use an action

By using actions, you limit the amount of time that your code is running, which lowers your overhead costs. For example, you can use actions to detect faces in an image, respond to changes in a database, aggregate a set of API calls, or even post a tweet.

Can you use more than one action at a time?

Yes. You can use actions to call other actions, or you can string actions together to create sequences. To make this work, the output of one action is the input for another action, which provides an output that can be used to trigger another

action, and so on. You can even bundle the group of actions that you create to form a package. With a package, you can reuse common actions or sequences by calling the package instead of configuring the action or sequence again.

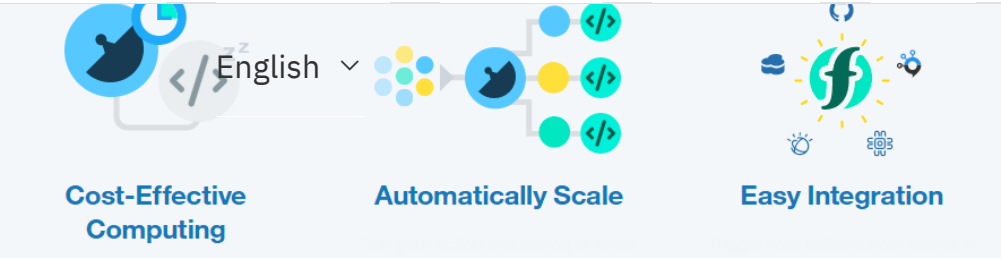


Figure 8. IBM Cloud Functions

← Previous *Completed* 17 of 21 *Modules* Next →