QUICK LAB 1

NO INFRASTRUCTURE, JUST CODE. SEE THE SIMPLICITY OF SERVERLESS.

Create, build, and run a cloud-native Node.js serverless app in less than 15 minutes

INTRODUCTION

This lab walks you through the steps required to create, build and run a Serverless application using IBM Cloud Functions. **Serverless computing** refers to a model where the existence of servers is entirely abstracted away. Even though servers exist, developers are relieved from the need to care about their operation. They are relieved from the need to worry about low-level infrastructural and operational details such as scalability, high-availability, infrastructure-security, and other details. Serverless computing is essentially about reducing maintenance efforts to allow developers to quickly focus on developing code that adds value.

Serverless computing simplifies developing cloud-native applications, especially microservice-oriented solutions that decompose complex applications into small and independent modules that can be easily exchanged. Some promising solutions like Apache OpenWhisk have recently emerged that ease development approaches used in the serverless model. **IBM Cloud Functions** is a Function-as-a-Service (FaaS) platform on IBM Cloud, built using the Apache OpenWhisk open source project, that allows you to execute code in response to an event.

It provides a serverless deployment and operations model. With a granular pricing model that works at any scale, you get exactly the resources you need – not more, not less – and you are charged for code that is really running.

IBM Cloud Functions provides:

- Support for multiple languages, including JavaScript, Python, Swift, and Java
- Support for running custom logic through Docker containers

- The ability to focus more on value-adding business logic, and less on low-level infrastructural and operational details.
- The ability to easily chain together microservices to form workflows via composition.

PREREQUISITES FOR THIS LAB

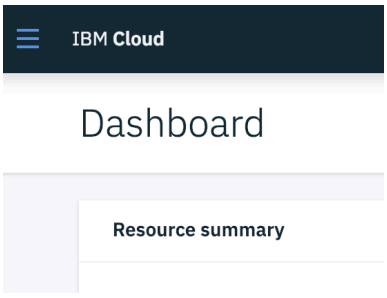
- You will need an IBM Cloud Account. Either use your existing account, or create a new account by accessing the following link: https://ibm.biz/Bd2Sfe

CREATE AN ACTION IN THE CLOUD FUNCTIONS UI

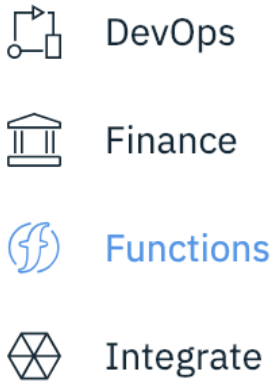
There are two main options to get started with Cloud Functions. Both allow you to work with Cloud Function's basic entities by creating, updating, and deleting actions, triggers, rules and sequences.

The CLI (command line interface) allows you to perform these basic operations from your shell. The IBM Cloud Functions UI (user interface), allows you to perform the same operations from your browser. During this lab we will use the UI to learn how to work with Cloud Functions.

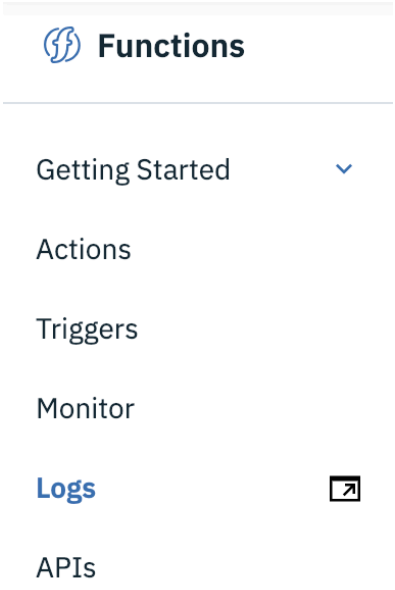
1. Start by logging into the IBM Cloud: https://cloud.ibm.com/login, and then selecting the hamburger menu in the header.



2. Then click on **Functions** to access the IBM Cloud Functions development experience on IBM Cloud.



3. The Cloud Functions UI is comprised of the following sections in the left hand side menu bar. You will learn about these basic entities in later sections.



   a. Actions – The actions section lists all actions you have created prior. An action is a small piece of code that can be explicitly invoked or set to automatically run in response to an event.
   b. Triggers – A trigger is a declaration that you want to react to a certain type of event, whether from a user or by an event source. A trigger can be fired

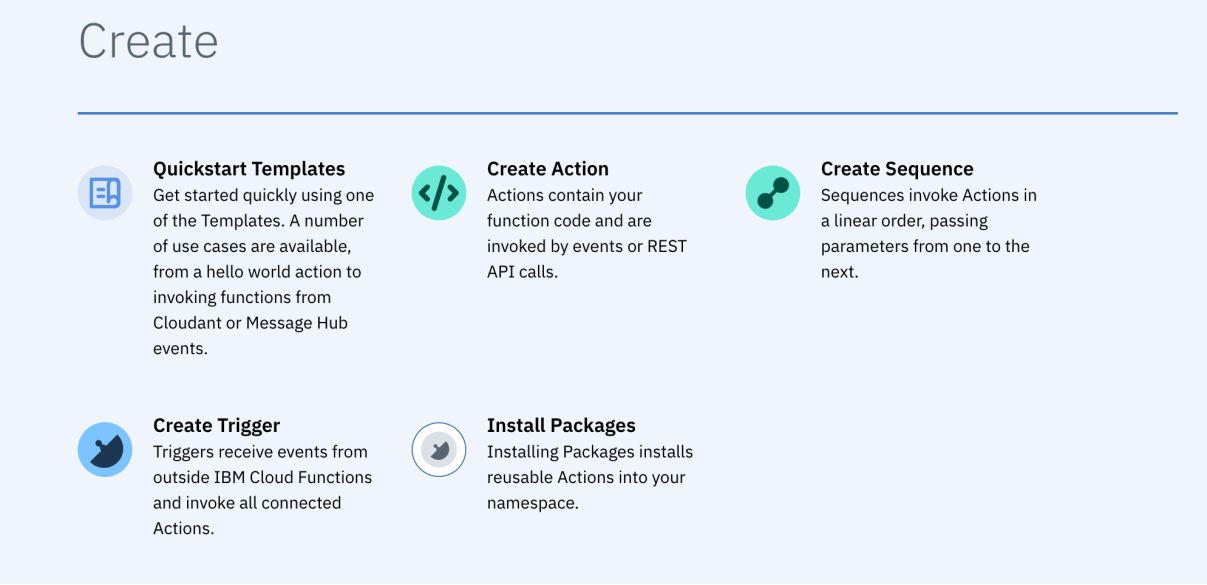
or activated. Triggers can be associated with actions, so that when the trigger is fired the action is run.

c. Monitor – This section shows you information about your actions and their activity, including an activity summary and timeline.

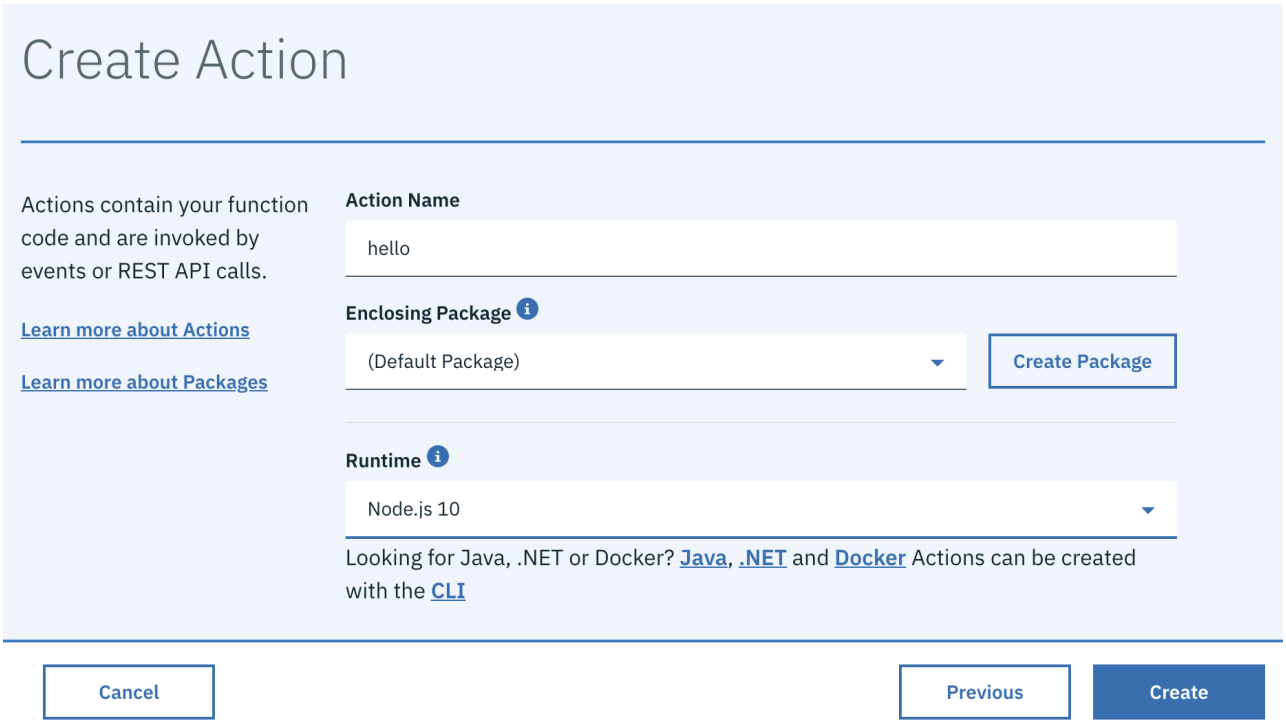
d. Logs – The logs section takes you to the IBM Cloud Logging service, which provides you with the ability to collect, analyze, and build dashboards for your logs.

e. APIs – The APIs section allows you to set up an API Gateway and API management for IBM Cloud Functions. This is beyond the scope of today's lab.

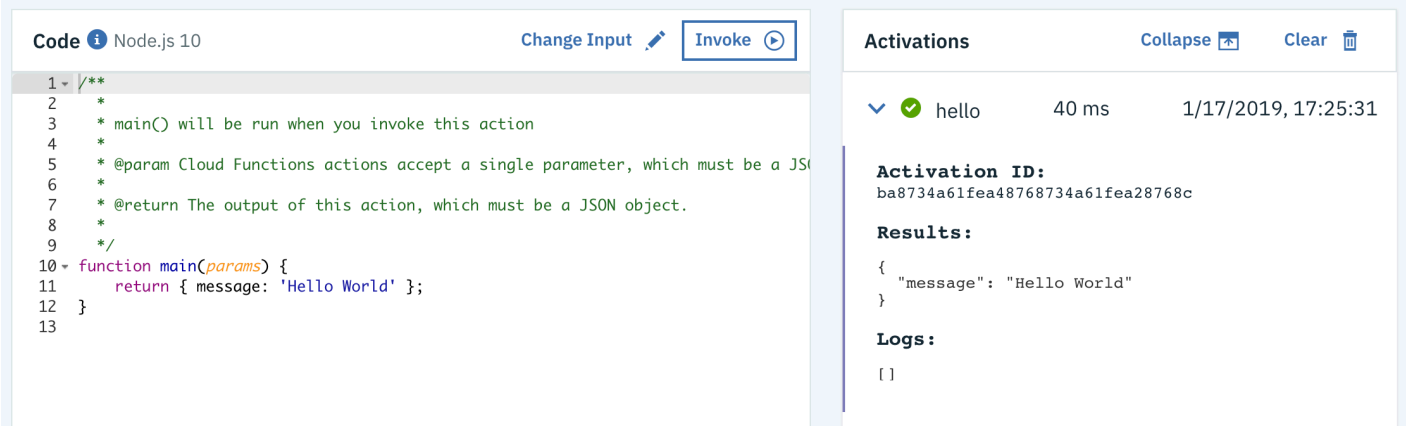
4. Start creating your first action by selecting the **Start Creating** button in the center of the UI, which opens the Create page. Then select the **Create Action** button.



5. Specify an Action Name (e.g. hello), by entering it into the text field, and then select Node.js 10 as the runtime. Leave everything else as-is and click the **Create** button at the bottom of the screen.



6. This opens a cloud-based code editor that you can use to create and extend your actions. There should already be some hello world code in the action.
7. Click **Invoke** to test this action directly from within your browser. You should see an Activations panel show up with the result.

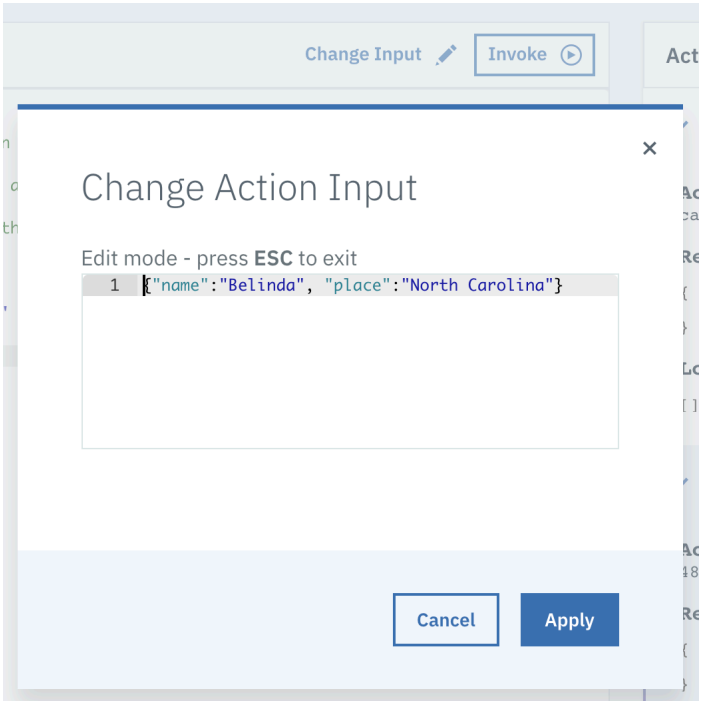


8. Actions may be invoked with a number of named parameters. Let's try out an action that accepts parameters. Update the action code by copy-pasting the following into the code section:

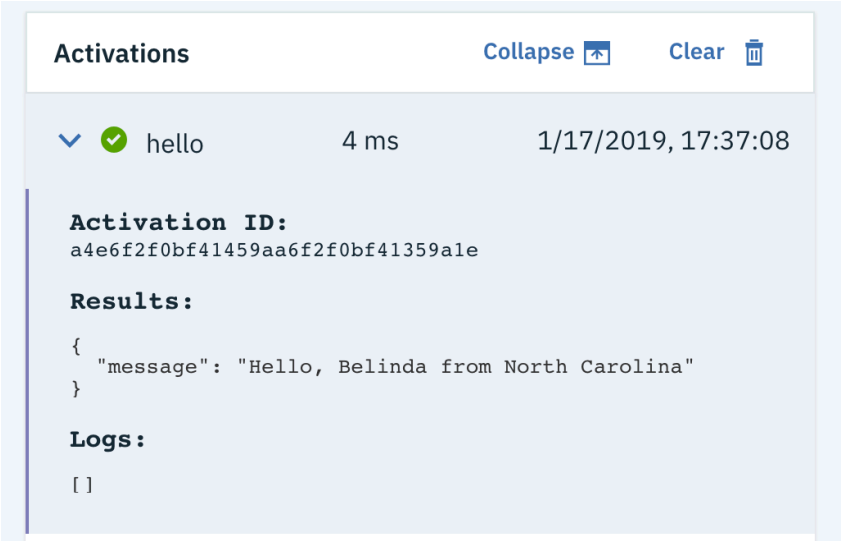
```
function main(params) {
    return { message: 'Hello, ' + params.name + ' from ' + params.place };
}
```

9. Click **Save,** then click **Invoke.** In the activations panel, you should see a result of Hello undefined from undefined. This is expected, because we didn't provide the action with any parameters. Let's do this.
10. Click on the **Change Input** button, and update the parameters with the following json:

    `{"name":"Belinda","place":"North Carolina"}`



11. Click the **Apply** button, and then click **Invoke** again to invoke your action. You should see an activation result with the name and place you provided.



CREATE A TRIGGER IN THE CLOUD FUNCTIONS UI

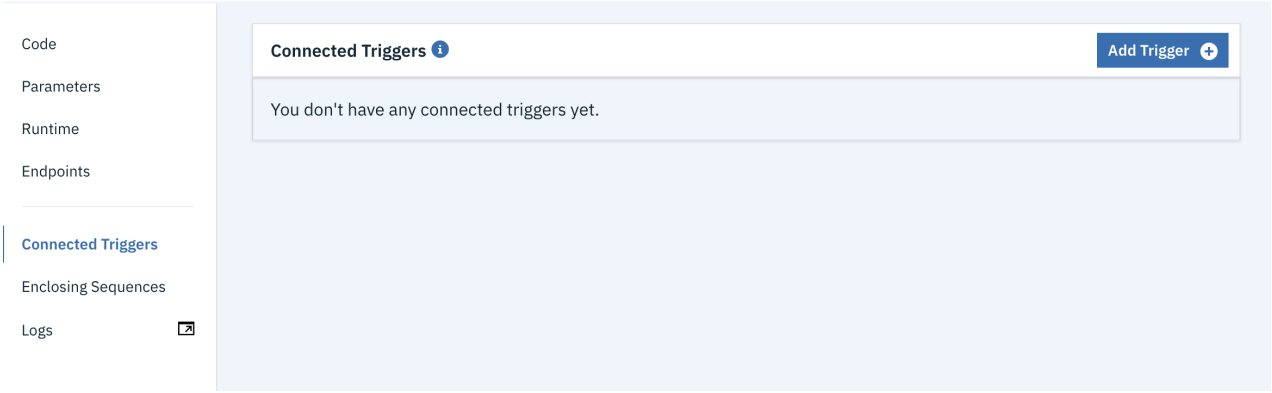
IBM Cloud Functions is a Functions-as-a-Service (FaaS) platform that executes code in response to events. Events can be emitted by services, such as other services that are a part of the IBM Cloud. These services have triggers that represent a named channel for a stream of events. Events can also be emitted in the form of API calls, fired by standard web or mobile applications, which can then trigger actions.

So far, we've only triggered our actions manually. Let's try to trigger our actions periodically.

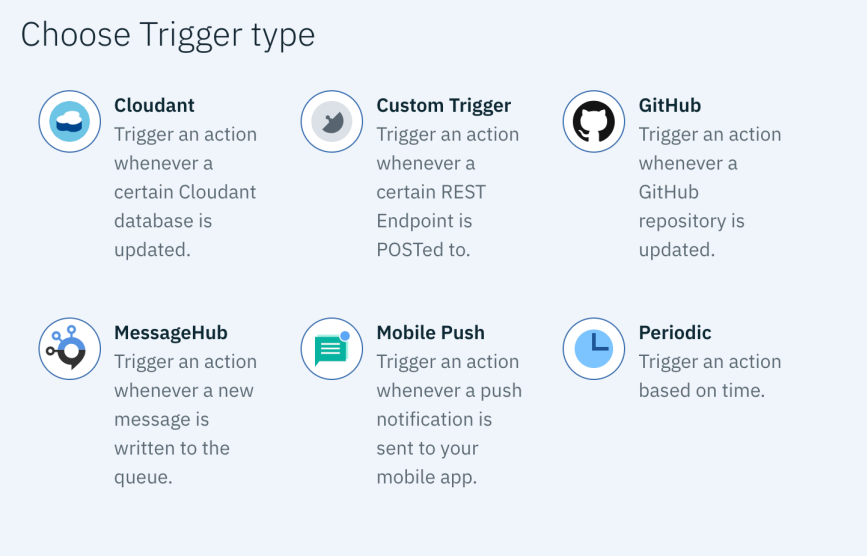
1. Update the existing action to contain the following code:

```
function main(params) {
  var date = new Date();
  var time = date.getHours() + ":" + date.getMinutes() + ":" + date.getSeconds();
  return { message: "The time is " + time };
}
```

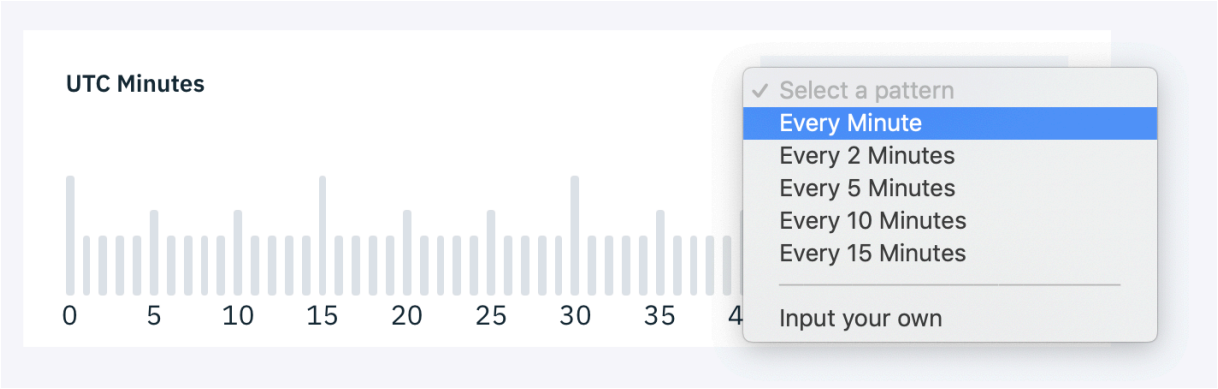
2. Click **Save.** This action returns the time. Try it out by clicking **Invoke.**
3. Next, click the **Connected Triggers** button on the left pane. This will allow you to add a Trigger that causes the action to be run.



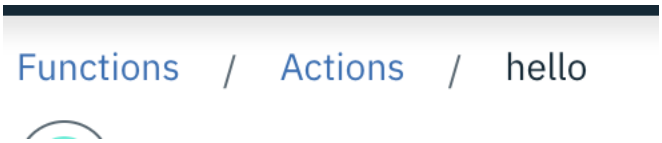
4. Click on **Add Trigger** on the right side of the panel, and select a **Periodic Trigger** as the type.



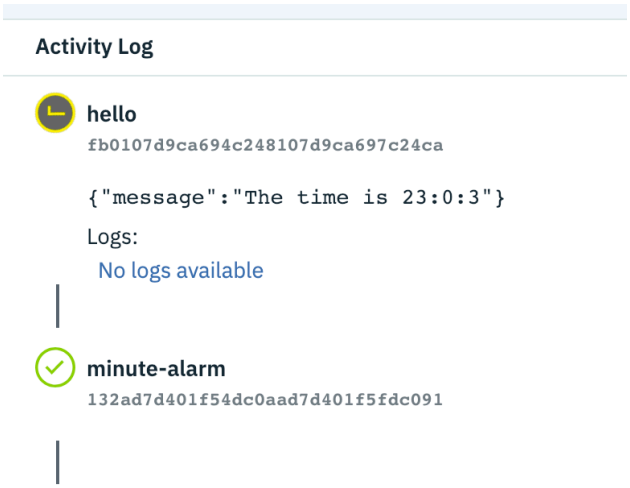
5. Give your trigger a name (e.g. “minute-alarm”)
6. scroll down to “UTC Minutes,” and select “Every Minute” from the pull down menu:



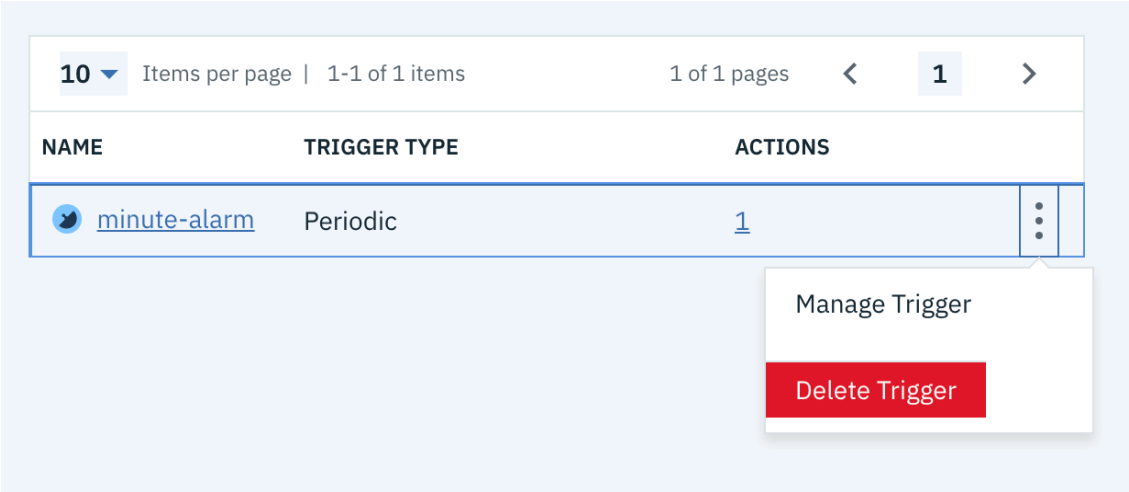
7. Click **Create & Connect** to create the trigger and connect your action to it. Return the dashboard by clicking on **Actions** in the breadcrumbs.



8. Click **Monitor** to see the activity of your actions and triggers. The Activity Log should show your action being triggered, and the time it was triggered in the result. If you don't see the result, click the check mark for more details.

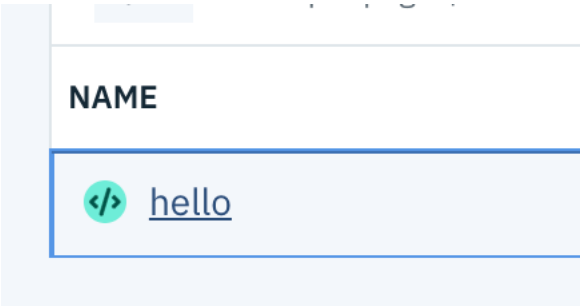


9. To delete the trigger, expand the **Triggers** section, select your **minute-alarm** trigger you just created, and click the trash bin from the pull-down menu.



CREATE A WEB ACTION

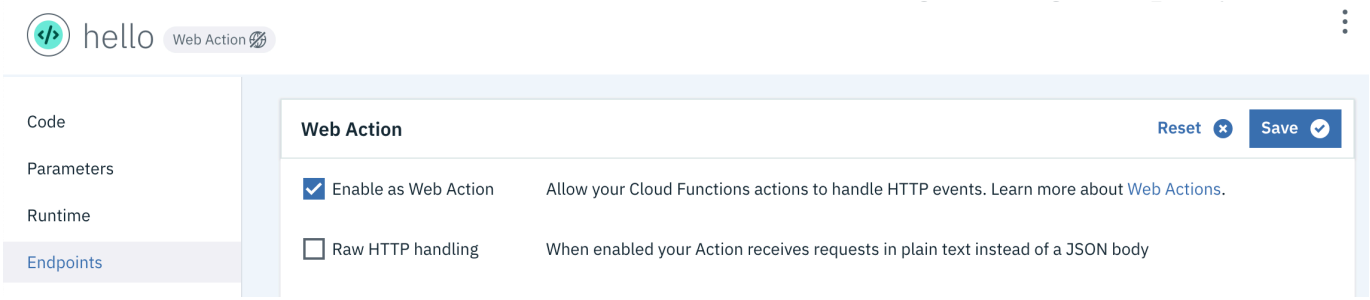
1. Select **Actions** from the left-hand panel.
2. Find your hello action, and mange it by clicking on its name.



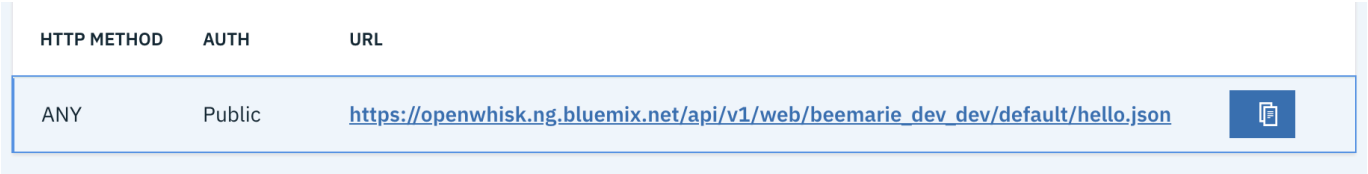
3. Change the code back to our hello action that takes name and place parameters:

```
function main(params) {

  return { message: 'Hello, ' + params.name + ' from ' + params.place };


}
```

4. Click **Save.**
5. To add a web endpoint and create a web action, select **Endpoints** from the left side menu, and select **Enable as a Web Action.**



6. Click **Save** again. Copy the URL to the clipboard, using the clipboard icon.



7. Once you have the URL copied, paste it as the URL into a browser window, and append the following parameters to the request to pass the name and place input parameters into the Action:

   `?name=Belinda&place=NC`

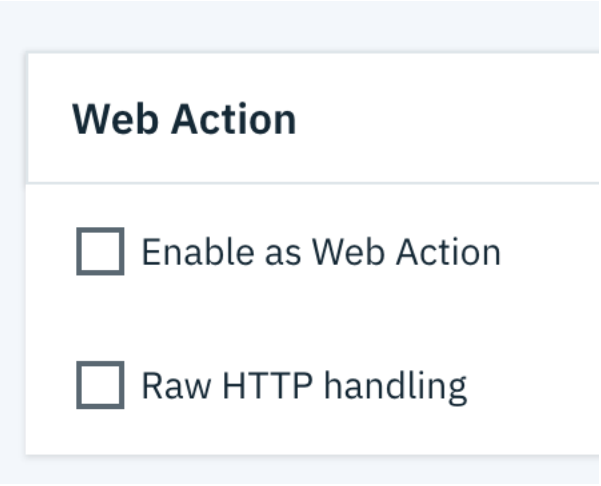
   The URL should look something like this:

   https://openwhisk.ng.bluemix.net/api/v1/web/beemarie_dev_dev/default/hello.json?name=Belinda&place=NC

8. As a response, you should then get the output of your action:

```
{
    message: "Hello, Belinda from NC"
}
```

9. This URL can be accessed by any http request. Disable the web action before you leave by deselecting the checkbox next to **Enable as Web Action** and clicking **Save.**



CONCLUSION

Congratulations! You have completed the lab. You have successfully built and deployed a number of Serverless Cloud Functions, including web actions that can be invoked from the browser or from microservices, all inside a browser! Feel free to reach out should you have any questions.