Quick Lab: No Infrastructure, Just Code. See the Simplicity of Serverless

Chris Bailey, IBM Runtimes, baileyc@uk.ibm.com
Carlos Santana, IBM Cloud Functions, csantana@us.ibm.com

No Infrastructure, Just Code: See the Simplicity of Serverless

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. I.e. that even though servers still 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 so forth. Hence, serverless computing is essentially about reducing maintenance efforts to allow developers to quickly focus on developing value-adding code.

Serverless computing simplifies developing cloud-native applications, especially micro-service-oriented solutions that decompose complex applications into small and independent modules that can be easily exchanged.

Serverless computing does not refer to a specific technology; instead if refers to the concepts underlying the model described prior. Nevertheless some promising solutions have recently emerged easing development approaches that follow the serverless model – such as OpenWhisk.

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 you with the previously mentioned serverless deployment and operations model, with a granular pricing model at any scale that provides you with exactly the resources – not more not less – you need and only charges you for code really running. It offers a flexible programming model. incl. support for languages like JavaScript, Swift, Python, and Java and even for the execution of custom logic via Docker containers. This allows small agile teams to reuse existing skills and to develop in a fit-for-purpose fashion. It also provides you with tools to declaratively chain together the building blocks you have developed. It is open and can run anywhere to avoid and kind of vendor lock-in.

In summary, IBM Cloud Functions provides...

- ... a rich set of building blocks that they can easily glue/stitch together
- ... the ability to focus more on value-add business logic and less on low-level infrastructural and operational details

... the ability to easily chain together microservices to form workflows via composition

This Lab Requires:

1. An account with IBM Cloud.
    - You may use your existing account or create a new account (Section 2)

Setting Up

Before starting this lab, please do the following:

- Go to http://ibm.biz/startmylab

- Select **No Infrastructure, Just Code. See the Simplicity of Serverless** lab from the dropdown and click Ok
    - You will be brought to the sign up page to register for an IBM Cloud Platform account. If you do not have an account, please register for one.
    - This lab does not require you to have an IBM Cloud Platform account but it allows you to do some optional steps.
- Clone the project for this lab:
    - Click on the Launchpad (rocket logo) in the toolbar, search for "Terminal" and click on the icon.
    - Clone the GitHub project containing the FoodTracker application we are going to extend:

      ```
      git clone http://github.com/seabaylea/Serverless
      ```

      A copy of these instructions are available included in the project. You can open the instructions using:

      ```
      open "Serverless/ Serverless.pdf"
      ```

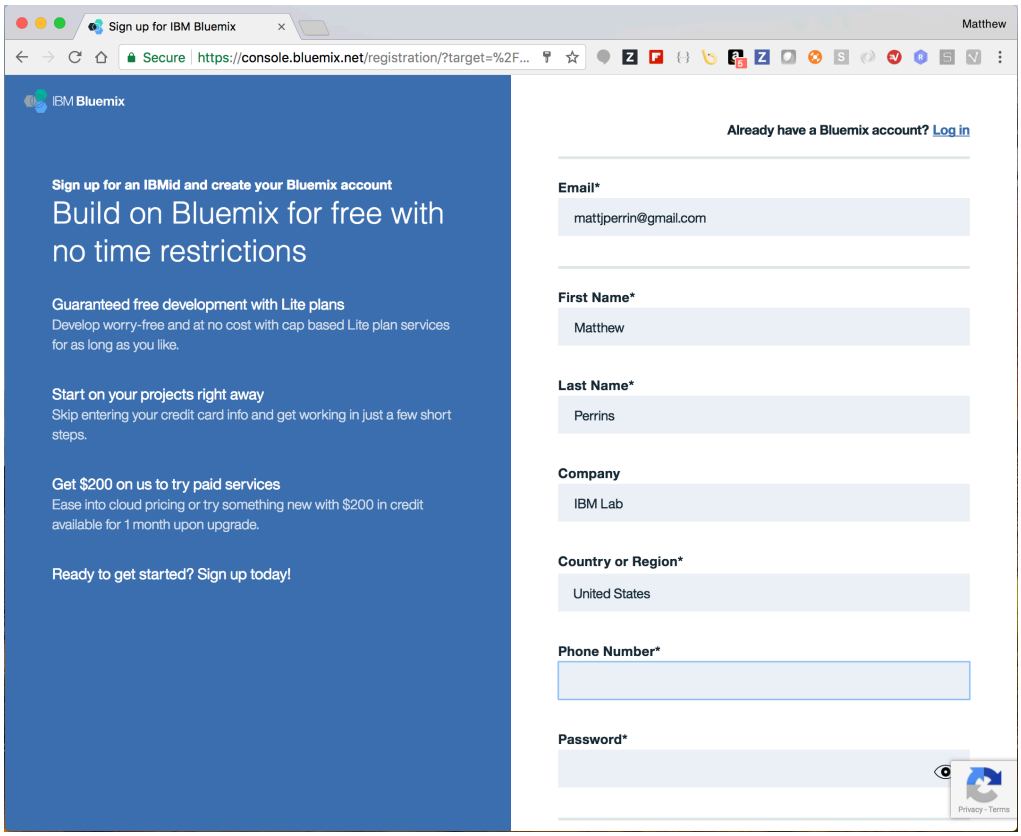
NEXT STEPS:
If you do not have an IBM Cloud account proceed to Section 2 (Create an Account)

If you do have an IBM Cloud account, proceed to section 3.

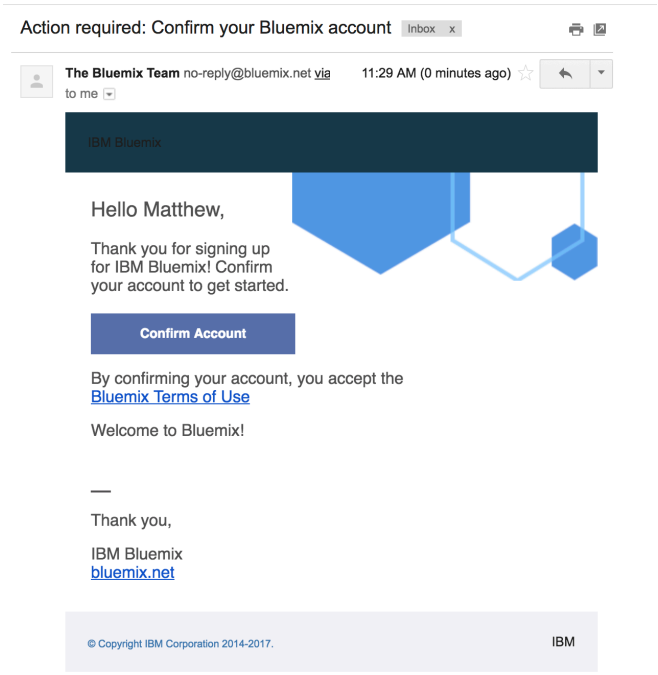
1. Create an IBM Cloud Account

If you **do** have an existing IBM Cloud account, skip to the next section.

1. To run these lab instructions, you will need to create a trial account on the IBM Cloud. You can do this free of charge and all the instructions contained in this lab do not incur any additional costs.
2. Create an IBM Cloud Trial account by navigating to this link. https://console.bluemix.net, and click on **Create a free Account.**
3. Complete the Registration Form with your details and a valid email address.

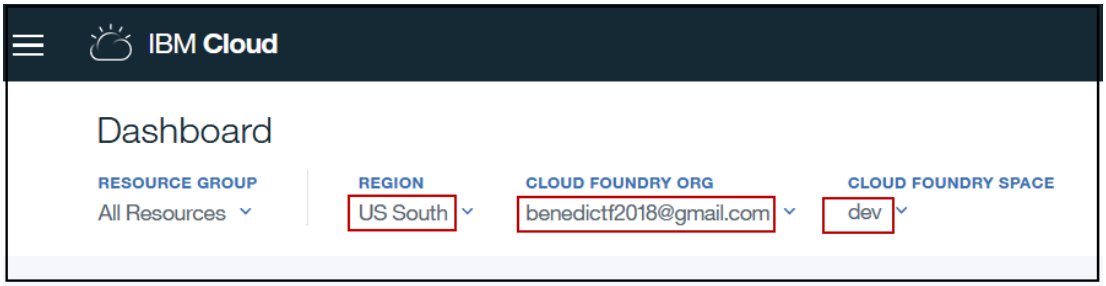


4. Click on **Create Account.**
5. Confirm your registration by accessing your email and clicking on **Confirm Account.**



6. You can now log into IBM Cloud.
7. Click on **Login** and enter your email and password
8. You will finally see the Dashboard View, which will be empty as you have not created any services or apps at this point. Please note the **Region**, **Org** and **Space** settings.

   Typically this will be **US South**, the Org setting will be same as your email address and by default you are placed in a Space called **dev**, as shown below.



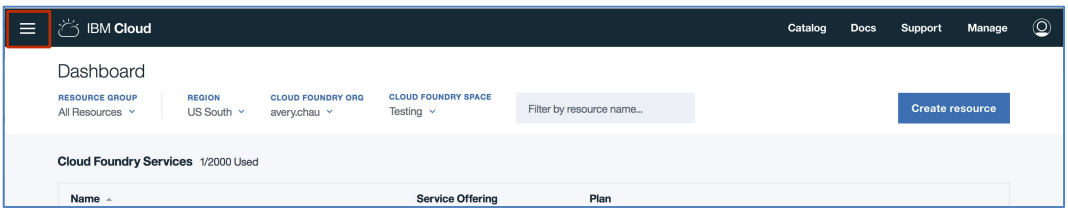
2. Create an Action in the Cloud Functions UI

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.

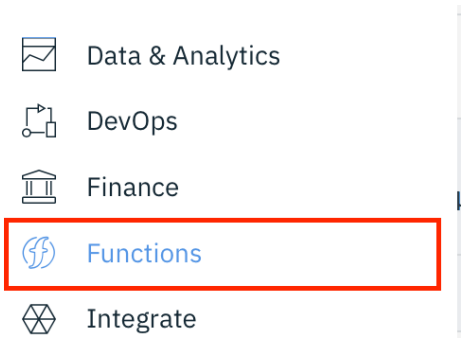
The CLI (command line interface) allows you to perform these basic operations from your shell. The IBM Cloud Functions UI (user interface), allows you to perform the same operations from your browser. During this lab we will use the UI to learn how to work with Cloud Functions.

1. Start by logging in to IBM Cloud.

   While logged into the IBM Cloud, click on the **Navigation** menu in the header:



   Then click on **Functions** to access the **IBM Cloud Functions** cloud-native development experience on IBM Cloud.



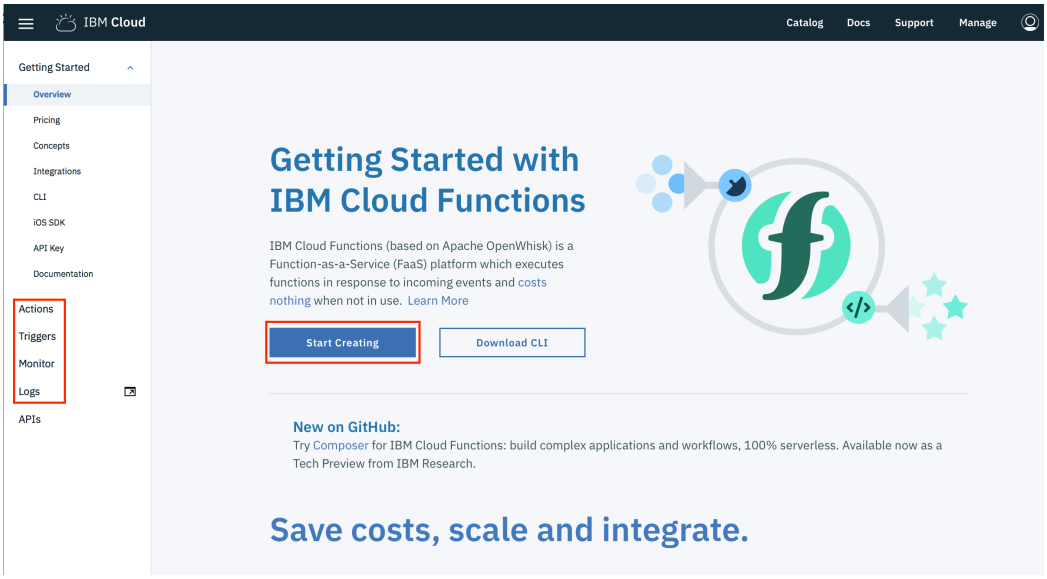
2. The Cloud Functions UI is comprised of the following sections in the left hand side menu bar(you will learn more about these basic entities like actions in the subsequent sections):

   a. Actions
      The Actions section lists all actions you have created prior.
      Clicking an action loads its code into the code editor.
      Hovering over an action lets a trash bin appear allowing to delete the action.

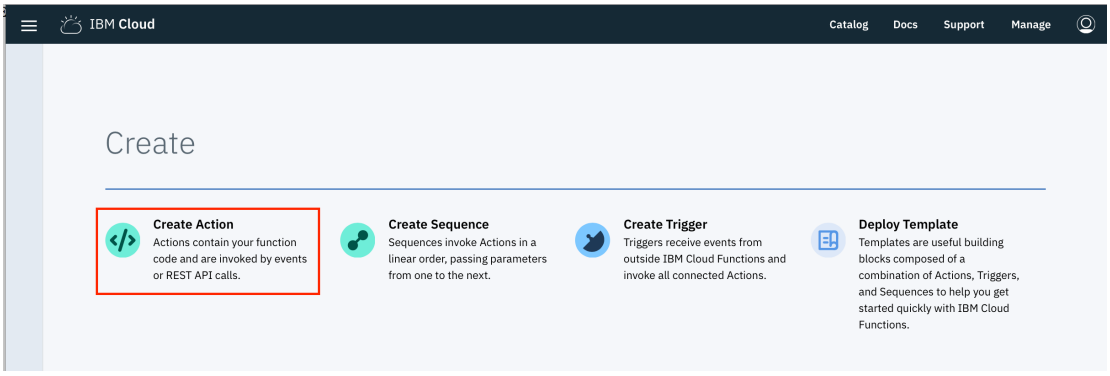
b. Triggers
The Triggers section lists all the triggers you have created prior.
Hovering over a trigger lets a flash icon appear allowing to fire the trigger as well as a trash bin allowing to delete the trigger.

c. Monitor
The Monitor section show you information about your actions and their activity, including an activity summary and timeline.

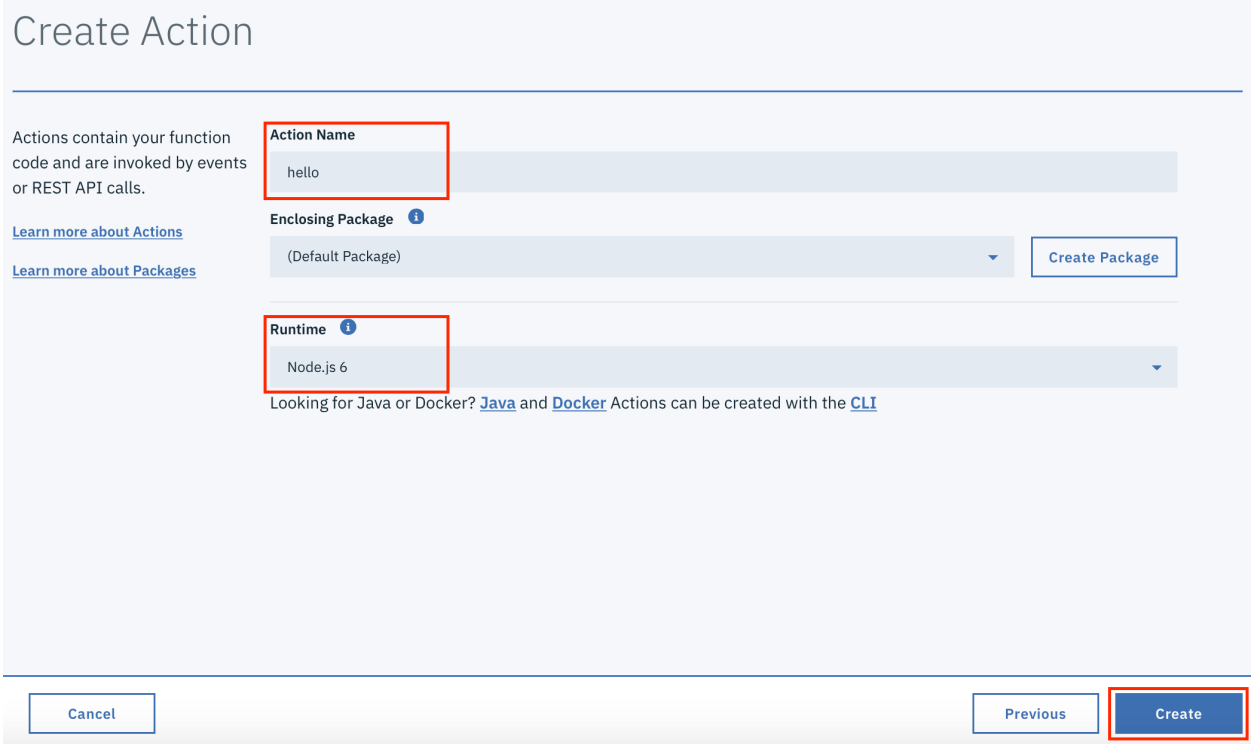
d. Logs
The Logs sections takes you to the IBM Cloud Logging service, which prvides you with the ability to collect, analyze and build dashboards for your logs.



3. Start creating your first action by selecting the “Start Creating” button in the center of the UI, which opens the Create page. Start by selecting the “Create Action” option:



Specify an Action Name (e.g. hello) by entering it into the text field, and select a Runtime of Node.js 6. Leave everything else as-is and click the Create Action button at the bottom of the screen.



This opens a cloud based code editor that you can use to create and extend your actions.

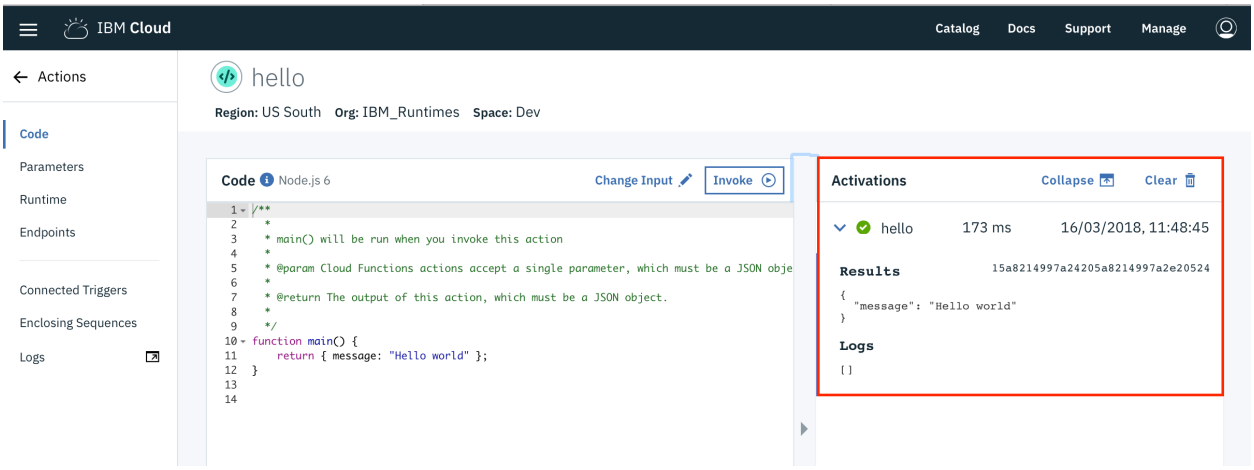
Copy the following code snippet into the code editor replacing any existing code:

```
function main() {
    return { message: "Hello world" };
}
```

Next click Save. This Save button will then we replaced with a Invoke button. Click the Invoke button to t est this action directly from within your browser.

This should provide a result directly in your browser:

```
{
  "message": "Hello world"
}
```

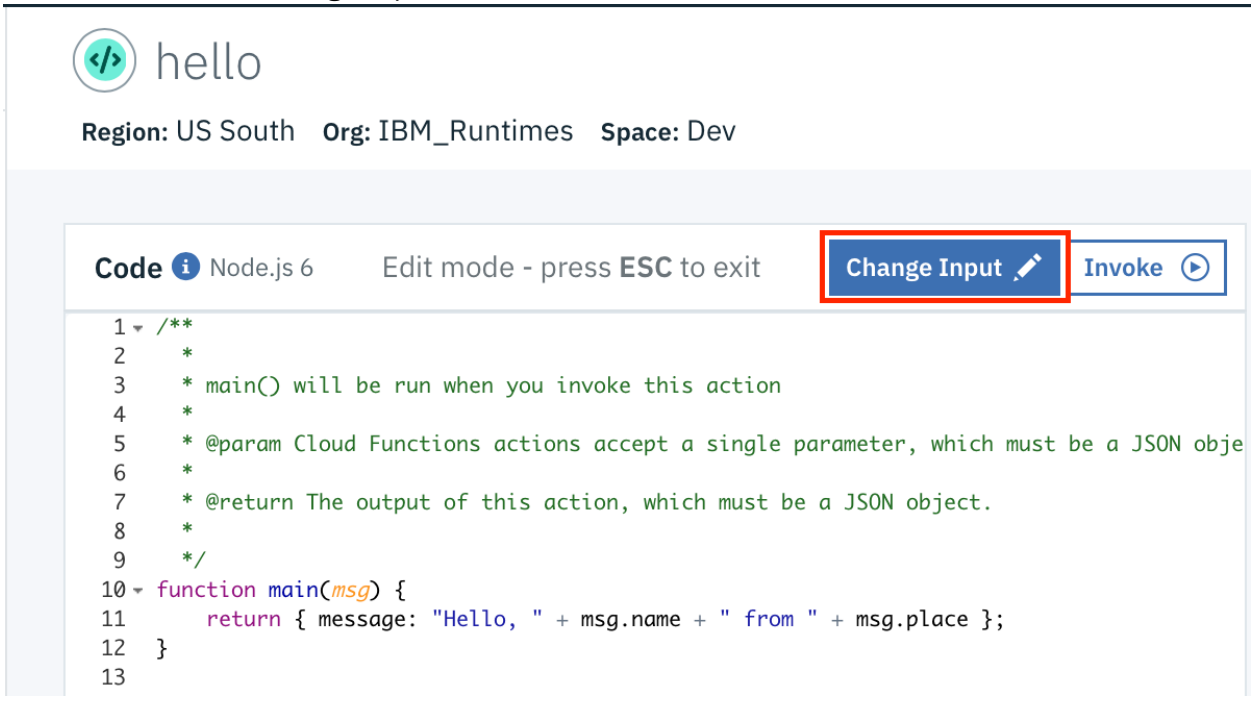


Actions may also be invoked with a number of named *parameters*. To see how things work when working with an action accepting parameters click the Change Input button and update the action code to be the following:

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

Once again, click the Save button to save the action.

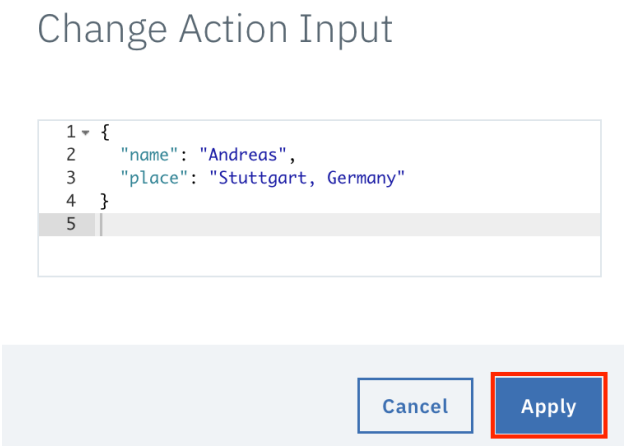
As this action now requires some input parameters, we can add those before we run invoke the action. Click on the Change Input button:



And add the following JSON data into the Change Action Input dialog box:

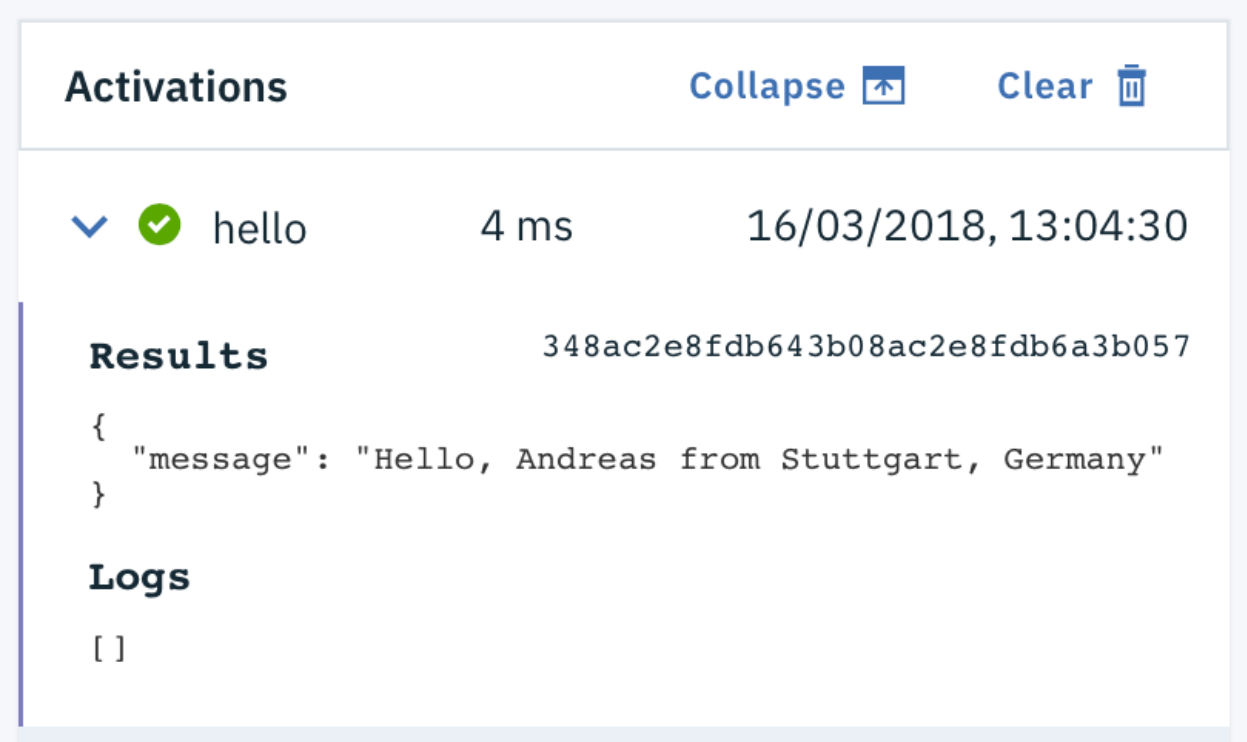
```
{
  "name": "Andreas",
  "place": "Stuttgart, Germany"
}
```

And click on the Apply button:



You can now run the action with the new input data by clicking the Invoke button. You should receive the following result:

```
{
  "message": "Hello, Andreas from Stuttgart, Germany"
}
```



3. Create a Trigger in the Cloud Functions UI

IBM Cloud Functions is a Function-as-a-Service (FaaS) platform supposed to execute code in response to events. Such events can be emitted in different ways: On the one hand side, events can be emitted by services (like other services part of the IBM Cloud platform) by what we call *triggers* which represent a named "channel" for a stream of events. On the other side, events can also be emitted in the form of API calls fired by standard web- or mobile applications which can then trigger actions, too.

So far we have triggered our actions "only" manually. Now, let's trigger actions another way, namely periodically.

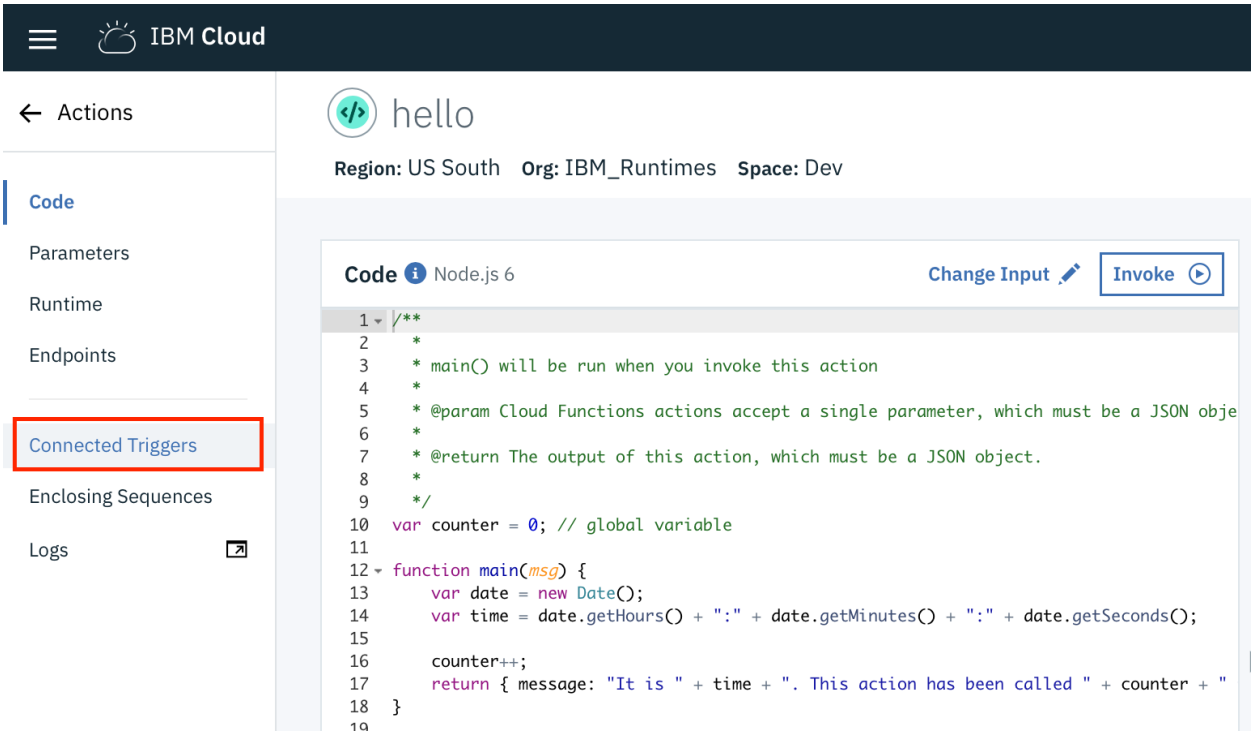
To test this create the following action the way you have learned it before:

Update your existing action to contain the following:

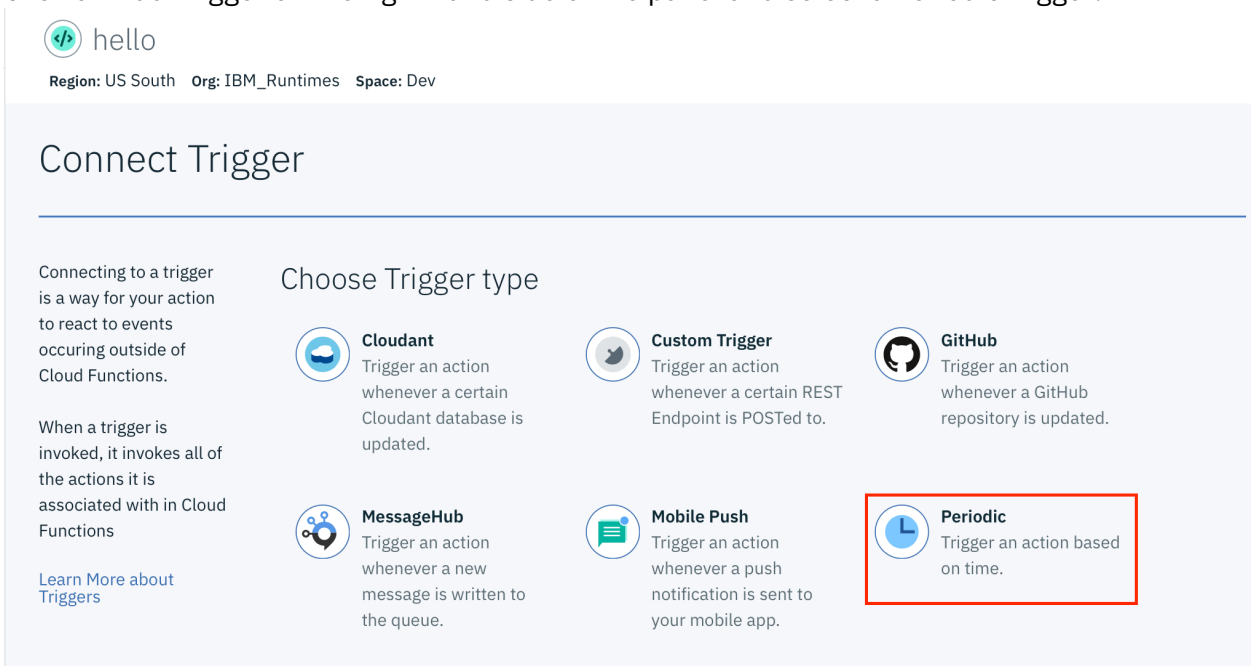
```
var counter = 0; // global variable

function main(msg) {
    var date = new Date();
    var time = date.getHours() + ":" + date.getMinutes() + ":" +
date.getSeconds();

    counter++;

    return { message: "It is " + time + ". This action has been called " +
counter + " time(s)." };

}
```

When being invoked the action returns the time and the amount of times it has been invoked. For the latter a so called global variable is being used which allows to hold data even beyond a single invocation of the same action.

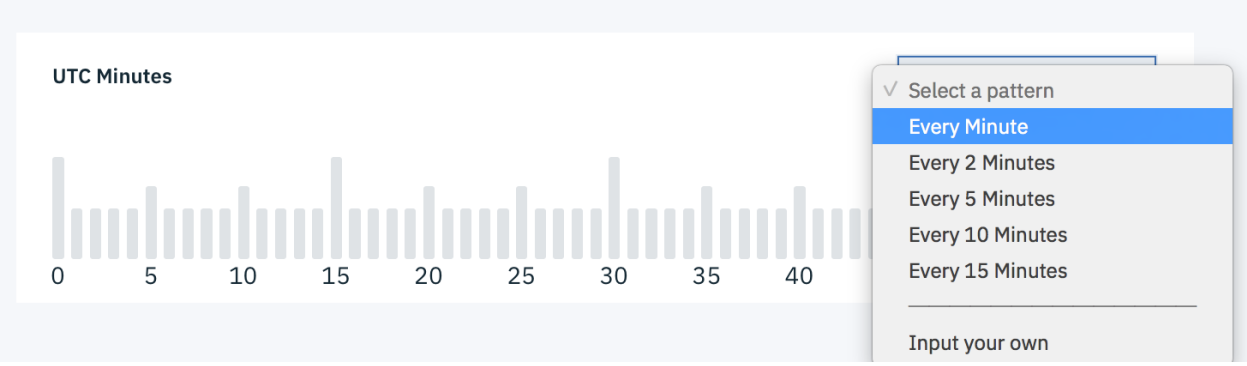
Next, click the `Connected Triggers` button on the left hand side of the pane. This will allow you to add a Trigger that causes the action to be run.



Click on Add Trigger on the right hand side of the panel and select a Periodic trigger:

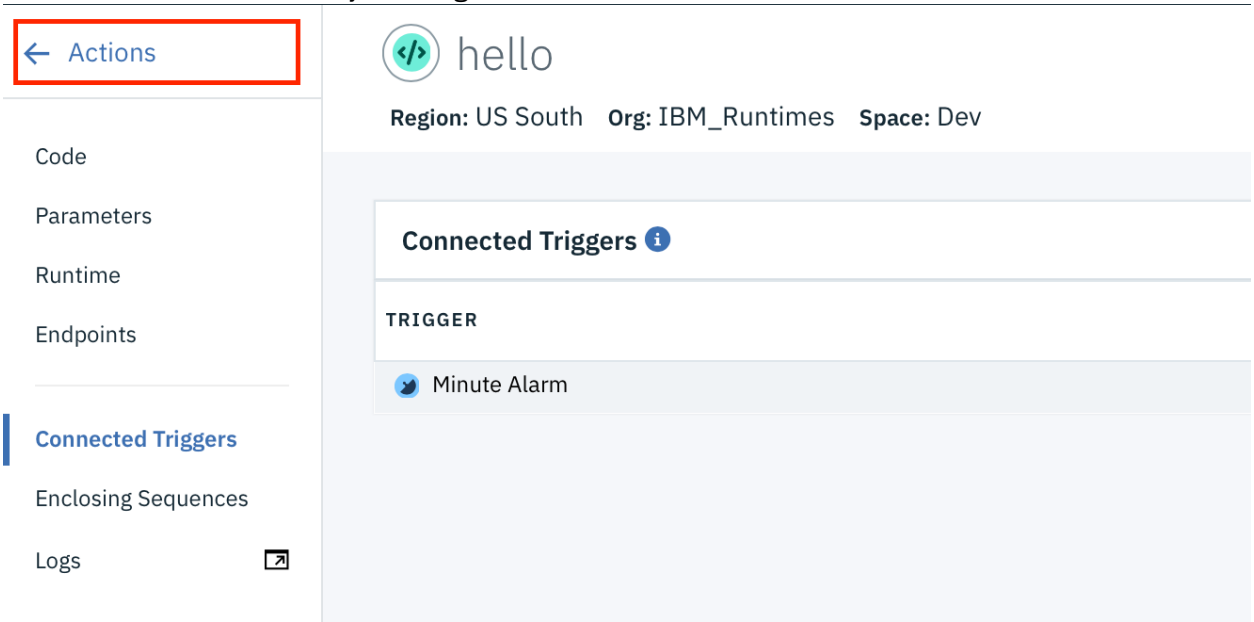


Give your trigger a name (eg. “minute alarm”), scroll down to “UTC Minutes”, and select Every Minute from the pull down menu:

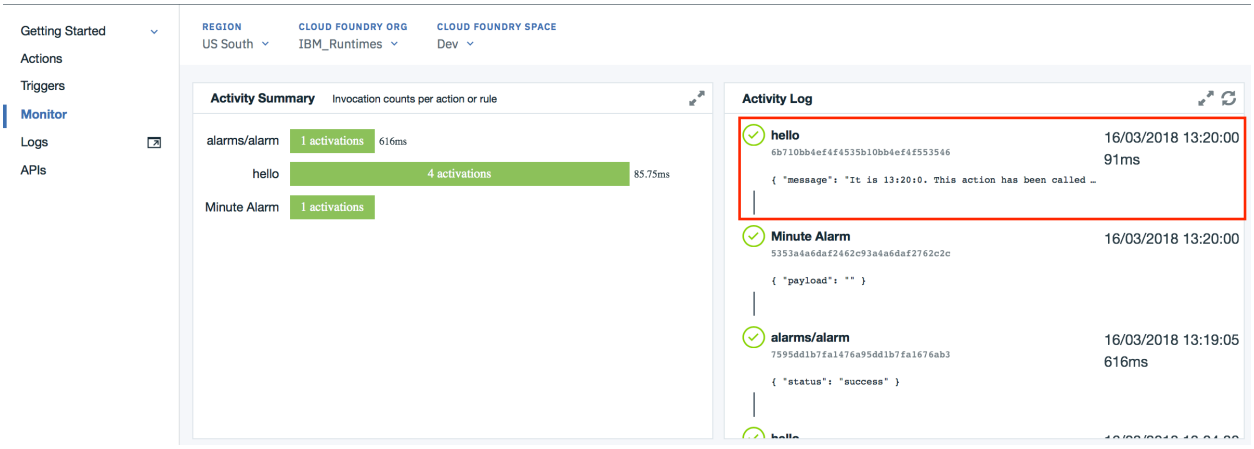


Click on “`Create & Connect`” in order to create the trigger and connect your action to it.

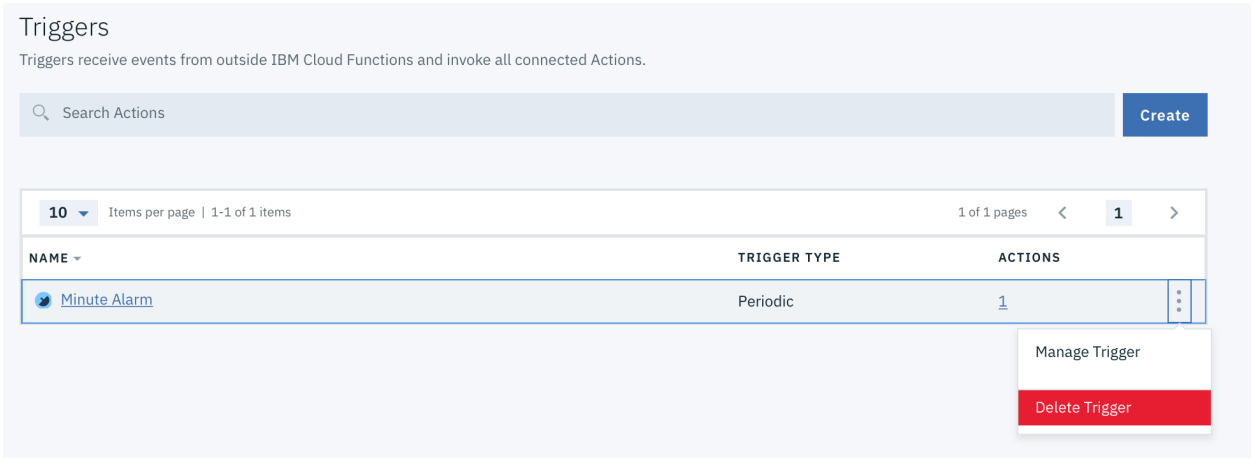
Return to the dashboard by clicking on the “`<- Actions`” button:



And select Monitor from the left hand menu 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:

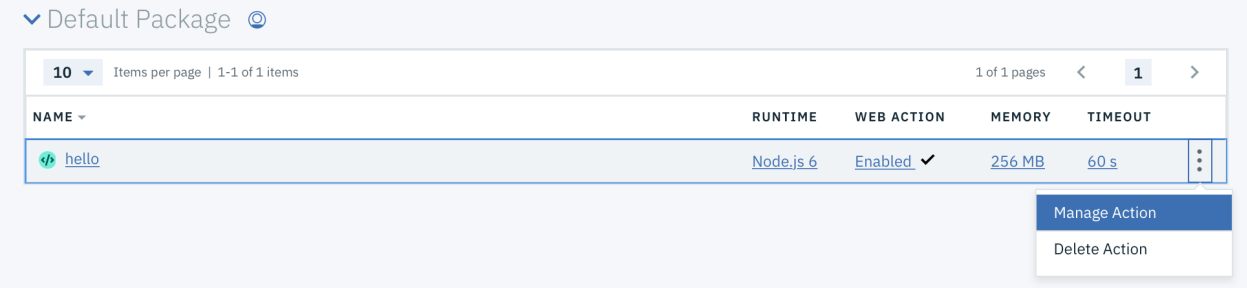


To delete the trigger, expand the `Triggers` section, select your Minute Alarm trigger you just created and click the trash bin from the pull down menu:



4. Create a Web Action

Select Actions from the left hand panel and select your “hello” action and choose Manage Action from the pull down menu:

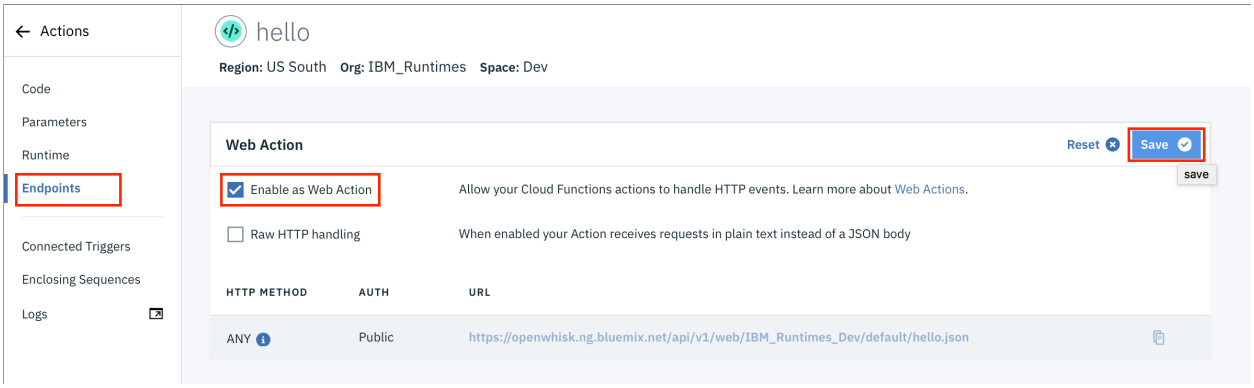


Change the code in the action back to our Hello action that takes name and place parameters:

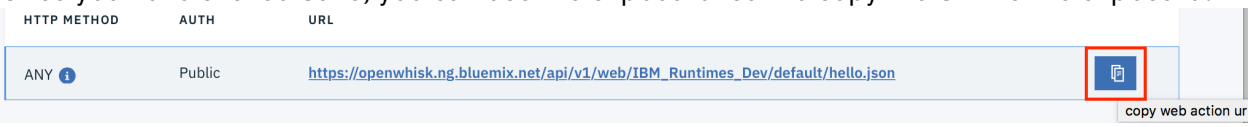
```
var counter = 0; // global variable
function main(msg) {
    return { message: "Hello, " + msg.name + " from " + msg.place };
}
```

and click save.

To add a web endpoint and create a web action, select Endpoints from the left hand menu and select “`Enable as a Web Action`”:



Once you have clicked Save, you can use the clipboard icon to copy the URL to the clipboard:



Copy this and then paste it as the URL into the browser window, appending the following query parameters to the request:

```
?name=Andreas&place=Stuttgart
```

This passes the name and place input parameters into the Action.

You should then get the output of your action as a response:

```
{
  "message": "Hello, Andreas from Stuttgart"
}
```

5. 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!

6. Resources

- IBM Cloud App Service: Build a Node.js app to deploy on Containers to IBM Cloud - https://www.youtube.com/watch?v=1qAjtduM2TY
- IBM Cloud App Service: http://bluemix.net/developer/appservice
- IBM Cloud Container Service: https://www.ibm.com/cloud/container-service
- IBM Cloud Garage Method: https://www.ibm.com/cloud/garage/
- IBM Cloud Continuous Delivery Service: https://www.ibm.com/cloud/continuous-delivery
- IBM Cloud DevOps: https://bluemix.net/devops
- IBM Cloud Developer Tools: https://www.youtube.com/watch?v=z-ByHuI41dU&t=76s
- IBM DevOps Overview: https://www.youtube.com/watch?v=psiMSUfn9oA
- Develop a Kubernetes App Toolchain: https://www.ibm.com/cloud/garage/tutorials/use-develop-kubernetes-app-toolchain?task=2
- IBM Cloud App Service: https://www.youtube.com/watch?v=VBZTwvLkGIk
- IBM Cloud Developer Tools (IDT): https://console.bluemix.net/docs/cloudnative/idt/index.html
- Using the IDT: https://www.youtube.com/watch?v=z-ByHuI41dU&t=76s
- Jump Start Your Microservice Development: https://www.youtube.com/watch?v=1HdtILoL6O4
- Spring@IBM Developer Center: http://developer.ibm.com/java/spring
- Node@IBM Developer Center: http://developer.ibm.com/cloud/node

Acknowledgements and Disclaimers (v8)