Grundbildung

gbssg.ch

Modul 346 » Cloud .

bla

Inhaltsverzeichnis

1	AWS CLI & AWS Lamda			
	1.1	Ziele	3	
	1.2	Vorbereitung	4	
	1.3	Aufgaben	4	

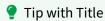
1 AWS CLI & AWS Lamda

1.1 Ziele

- Sie erstellen lokal eine AWS Lambdafunktion und veröffentlichen diese via AWS CLI.
- Sie lassen sich alle bereits existierenden Lambdafunktionen Ihres Accounts via AWS CLI anzeigen.
- Sie führen eine Lambdafunktion via AWS CLI aus.
- Sie finden gezielt nach Lambda Logmeldungen über AWS CloudWatch.
- Sie löschen eine Lambdafunktion über AWS CLI.

Mit Virtualisierung sind Sie sicherlich bereits während Ihrer bisherigen Lehre in Berührung gekommen. Versuchen Sie den nachfolgenden Arbeitsauftrag zuerst alleine zu bearbeiten. Wenn Sie fertig sind, können Sie sich mit Ihrem Banknachbarn austauschen:

- 1. Was bedeutet Virtualisierung grundsätzlich?
- 2. Wie funktioniert Virtualisierung grob?
- 3. Wo und wann macht Virtualisierung Sinn?



This is an example of a callout with a title.

Note

Note that there are five types of callouts, including: note, tip, warning, caution, and important.

Tip

Note that there are five types of callouts, including: note, tip, warning, caution, and important.

Warning

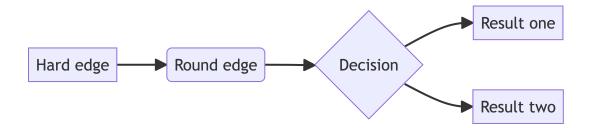
Note that there are five types of callouts, including: note, tip, warning, caution, and important.

Caution

Note that there are five types of callouts, including: note, tip, warning, caution, and important.

Important

Note that there are five types of callouts, including: note, tip, warning, caution, and important.



1.2 Vorbereitung

In dieser Übung wird das AWS CLI Tool benötigt. Das Tool kann entweder lokal installiert und konfiguriert werden oder direkt über die CloudShell nutzen.

1.3 Aufgaben

Aufgabe 1: Lamdafunktion erstellen | Lamdafunktion erstellen erstelle erstellen erstelle erstelle

Als Grundlage einer Lambdafunktion wird ein Script verwendet. Dieses Script wird beim Aufruf von AWS Lambda ausgeführt. Erstellen Sie eine index.js Datei mit nachfolgendem Inhalt:

```
exports.handler = async function(event, context) {
  console.log("EVENT\n" + JSON.stringify(event, null, 2))
  return context.logStreamName
}
```

Bevor Sie jetzt die neue Funktion erstellen. Lassen Sie sich alle vorhanden Lambdafunktionen ausgeben:

```
aws lambda list-functions
```

Für das Erstellen der Lambdafunktion wird ein deployment package verwendet. Dies ist ein zip-Archiv, das alle für die Ausführung der Funktion notwendigen Abhängigkeiten enthält.

```
zip function.zip index.js
```

Die Datei function.zip enthält nun alles, was AWS Lambda benötigt. Dieses deployment package wird nun freigegeben. Ersetzen Sie zuvor <lab-function-name> durch einen Namen Ihrer Wahl. Das Erstellen der Lambdafunktion funktioniert nur, wenn der gewählte Name nicht bereits von einer anderen Lambdafunktion verwendet wird. Zudem wird noch die Role-ARN benötigt:

AWS Management Console -> IAM -> Rollen -> LabRole -> ARN

Kopieren Sie die ARN. Sie ist im nachfolgenden Befehl für <lab-role-arn> einzusetzen.

```
aws lambda create-function --function-name <lab-function-name> --zip-file

→ fileb://function.zip --handler index.handler --runtime nodejs16.x --role

→ <lab-role-arn>
```

Die Lambdafunktion ist nun erstellt. Überprüfen Sie diese anhand der AWS Management Console oder des obigen Befehls.



Aufgabe 1.1: Advanced - Update einer Lambdafunktion | 🚣 Einzelarbeit | 🕓 5'

Oben haben wir gesehen, dass eine Funktion im Prinzip nur genau ein Mal erstellt werden kann. Ein zweiter Aufruf von aws lambda create-function ... schlägt fehl weil der Name bereits existiert. Wie kann man diesem Problem begegnen? Suchen Sie hierfür eine Lösung. Stichwort hierzu lautet update-function-code.



Das Anpassen der Funktion ist natürlich wichtig, um beispielsweise Bugs oder Optimierungen am Code zu veröffentlichen.

Aufgabe 2: Lamdafunktion ausführen | Linzelarbeit | U 10'

Eine Lambdafunktion kann über AWS CLI ausgeführt werden:

```
aws lambda invoke --function-name <lab-function-name> response.txt
```

Die Datei response.txt enthält Daten, die von der Lambdafunktion zurückgegeben wurden. Analyiseren Sie den Funktionscode und den Inhalt der Datei response.txt. Was enthält die Datei?



Die Lambdafunktion kann beliebig oft aufgerufen werden. Amazon verrechnet datei die Laufzeit der Funktion in ms.

Logausgaben sind zu finden unter:

AWS Management Console -> CloudWatch -> Protokolle -> Protokollgruppen -> /aws/lambda/<lab-function-name>

```
Aufgabe 2.1: Advanced - Übergabe von Daten | Linzelarbeit | U 10'
```

Auch via AWS CLI ist es möglich, der Funktion gewisse Daten mitzugeben. Diese Daten landen dann im Objekt event (siehe Code der Lambdafunktion).

```
aws lambda invoke --function-name <lab-function-name> --cli-binary-format

→ raw-in-base64-out --payload '{"name": "John Smith"}' response.txt
```

Versuchen Sie herauszufinden, was mit den Daten {"name": "John Smith"} genau passiert. Tipp: Die Daten werden in der Lambdafunktion geloggt.



Spielen Sie ein wenig dem Parameter -- payload und schauen Sie sich jeweils das Resultat im Log an.

Aufgabe 3: Lamdafunktion löschen 🚨 Einzelarbeit | 🕓 5'

Nicht mehr benötigte Lambdafunktionen können entfernt werden:

aws lambda delete-function <lab-function-name>

Überprüfen Sie die Aktion, indem Sie alle vorhanden Lambdafunktionen anzeigen.

