

Lambda Funktionen aufsetzen:

1. Bei AWS anmelden
2. AWS Lambda auswählen
3. Auf Funktion erstellen klicken

Funktionen (9)

Zuletzt abgerufen 18.1.2026, 19:18:51 

Aktionen

Funktion erstellen

 Search by attributes or search by keyword

< 1 > | 

4. Namen geben und Laufzeit wählen
5. Auf Funktion erstellen klicken

Funktion erstellen Info

Wählen Sie eine der folgenden Optionen aus, um Ihre Funktion zu erstellen.

Ohne Vorgabe erstellen

Beginnen Sie mit einem einfachen „Hello World“-Beispiel.

Vorlage verwenden

Erstellen Sie eine Lambda-Anwendung aus Beispiel-Codes und Konfigurationseinstellungen für häufige Anwendungsfälle.

Container-Image

Wählen Sie ein Container-Image aus, das für Ihre Funktion bereitgestellt werden soll.

Grundlegende Informationen

Funktionsname

Geben Sie einen Namen zur Beschreibung Ihrer Funktion ein.

myFunctionName

Der Funktionsname muss 1 bis 64 Zeichen lang sein, muss für die Region eindeutig sein und darf keine Leerzeichen enthalten. Gültige Zeichen sind a-z, A-Z, 0-9, Bindestriche (-) und Unterstriche (_).

Laufzeit

| Info

Wählen Sie die Sprache aus, die zum Schreiben Ihrer Funktion verwendet werden soll. Beachten Sie, dass der Konsolen-Code-Editor nur Node.js, Python und Ruby unterstützt.

Node.js 24.x



Zuletzt abgerufen 18.1.2026, 19:22:09

Dauerhafte Ausführung - neu

| Info Aktivieren Sie die dauerhafte Ausführung, um die Entwicklung ausfallsicherer, mehrstufiger Anwendungen zu vereinfachen, die den Fortschritt speichern und nach Unterbrechungen fortsetzen. [Preise anzeigen](#) 

Aktivieren

Architektur

| Info

Wählen Sie die Architektur des Anweisungssatzes für Ihren Funktionscode aus.

arm64

x86_64

Berechtigungen

| Info

Lambda erstellt standardmäßig eine Ausführungsrolle mit der Berechtigung für das Hochladen von Protokollen in die Amazon CloudWatch Logs. Sie können diese Standardrolle später durch das Hinzufügen von Auslösern anpassen.

 [Standard-Ausführungsrolle ändern](#)

▶ Zusätzliche Konfigurationen

Verwenden Sie zusätzliche Konfigurationen, um Netzwerk, Sicherheit und Governance für Ihre Funktion einzurichten. Diese Einstellungen tragen dazu bei, die Bereitstellung Ihrer Lambda-Funktion zu sichern und anzupassen.

[Abbrechen](#)

 [Funktion erstellen](#)

6. Code implementieren und deployen

The screenshot shows the AWS Lambda code editor interface. On the left, there's a sidebar with icons for file operations like Open, Save, and Deploy. The main area shows a code editor with the following content:

```
1 export const handler = async (event) => {
2     // TODO implement
3     const response = {
4         statusCode: 200,
5         body: JSON.stringify('Hello from Lambda!'),
6     };
7     return response;
8 };
9 
```

Below the code editor is a deployment panel titled "DEPLOY / Current". It contains two buttons: "Deploy (Ctrl+Shift+U)" (highlighted with a red circle) and "Test (Ctrl+Shift+I)". There are also sections for "TEST EVENTS [NONE SELE..." and "+ Create new test eve...".

7. Tests erstellen und speichern

This screenshot shows the "Event Sources" section of the AWS Lambda function configuration. At the top, there are tabs for "Testereignis" (selected), "Info", and "CloudWatch Logs Live Tail". Below the tabs are buttons for "Speichern" (highlighted with a red circle) and "Test".

The main area has a heading "Testereignisaktion" with a radio button for "Neues Ereignis erstellen". To the right is a link "Gespeicherte Ereignis bearbeiten".

Under "Aufruftyp", there are two options: "Synchron" (selected) and "Asynchron". A note for "Synchron" states: "Führt die Lambda-Funktion aus und blockiert, bis die Antwort der Funktion empfangen wird, mit einem maximalen Timeout von 15 Minuten. Gibt die Funktionsausgabe oder Fehlerdetails direkt an die aufrufende Anwendung zurück." A note for "Asynchron" states: "Stellt die Lambda-Funktion zur Ausführung in die Warteschlange und gibt sofort eine Anforderungs-ID zurück. Die Funktion wird unabhängig verarbeitet, wobei die Ergebnisse optional an ein konfiguriertes Ziel wie SQS, SNS oder EventBridge gesendet werden."

The "Ereignis-Name" field is set to "MyEventName". A note says: "Maximal 25 Zeichen aus Buchstaben, Zahlen, Punkten, Bindestrichen und Unterstrichen." Below it is a section for "Einstellungen für die Ereignisfreigabe".

For "Privat", a note says: "Dieses Ereignis ist nur in der Lambda-Konsole und dem Ersteller des Ereignisses verfügbar. Sie können insgesamt 10 konfigurieren." For "Gemeinsam nutzbar", a note says: "Dieses Ereignis steht IAM-Benutzern innerhalb desselben Kontos zur Verfügung, die über Berechtigungen für den Zugriff auf und die Verwendung von gemeinsam nutzbaren Ereignissen verfügen." Below these are "Vorlage - optional" and "Ereignis-JSON".

The "Ereignis-JSON" field contains the following JSON:

```
1 + []
2   "key1": "value1",
3   "key2": "value2",
4   "key3": "value3"
5 
```

There is a "JSON formatieren" button and a "Zuletzt abgerufen 18.1.2026, 19:29:28" timestamp.

8. Auslöser (wenn benötigt) hinzufügen

This screenshot shows the "test" function overview page. At the top, there are buttons for "Drosselung", "ARN kopieren", and "Aktionen". Below the buttons are links for "Export nach Infrastructure Composer" and "Herunterladen".

The main area has a heading "Funktionsübersicht" with a "Diagramm" tab selected. It shows a diagram of the function structure with a box labeled "test" and "Layers (0)". Below the diagram is a button "+ Auslöser hinzufügen" (highlighted with a red circle).

On the right side, there are sections for "Beschreibung", "Letzte Änderung", "Funktions-ARN", and "Funktion-URL".