

## Uygulamalı Laboratuvar - AWS Lambda Oluşturma

Tahmini Süre: 20 dakika

Bu laboratuvar, Node.js üzerinde AWS Lambda fonksiyonları oluşturma ve test etme konusunda bilgi edinmenizi sağlayacaktır.

**Önemli:** Bu laboratuvar kredi kartı kullanmanız gerektir.

### Öğrenme Hedefleri:

Bu alıştırmayı tamamladıktan sonra aşağıdaki görevleri yerine getirebilmeniziz:

- AWS Lambda fonksiyonunu oluşturmak
- AWS Lambda fonksiyonunun çıktısını test etmek

### Ön Koşullar

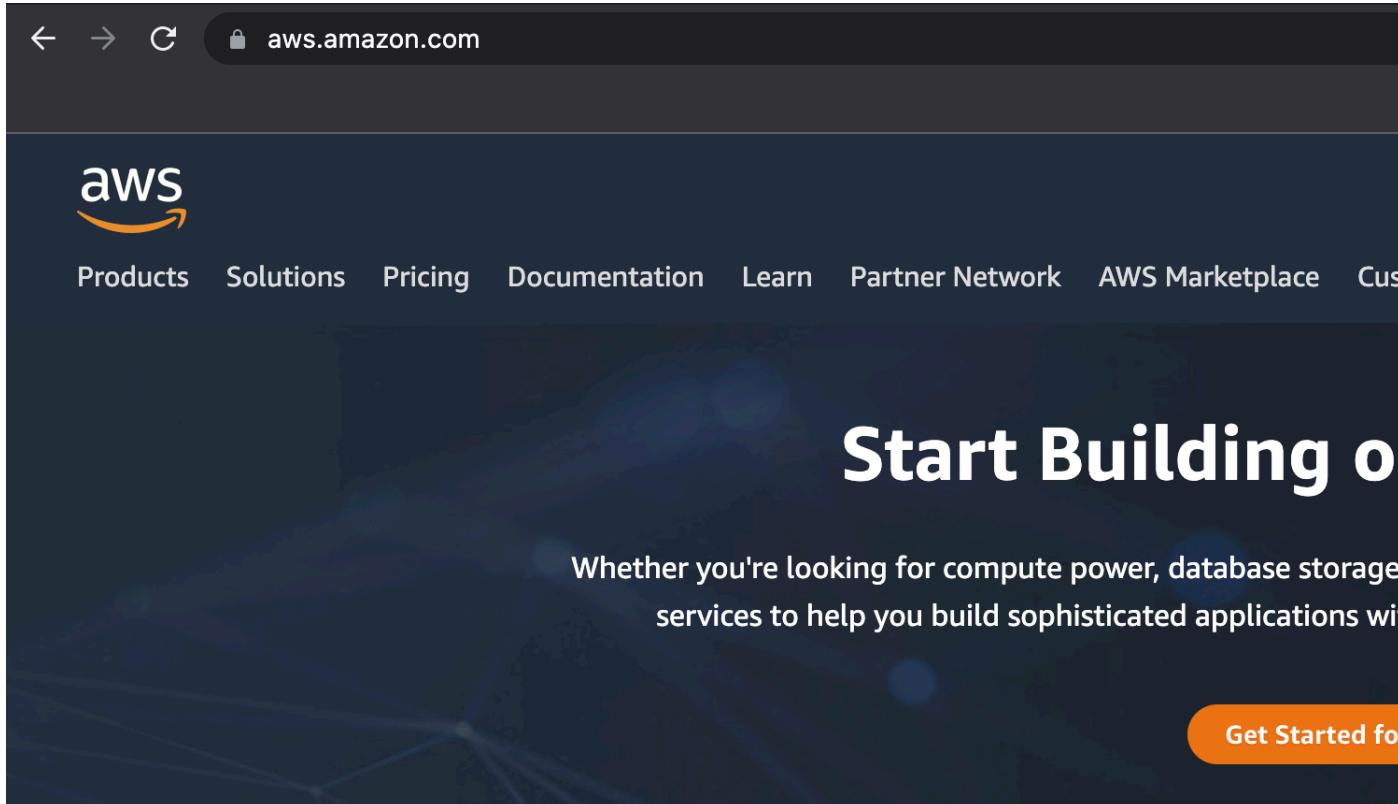
- Bir AWS hesabınız olmalıdır.
- Node.js ile aşina olmalısınız.

**Önemli:** Lütfen ücretsiz katmanın ötesindeki herhangi bir kullanımın, AWS hesabını oluşturmak için kullandığınız kredi kartına yansıtılacağını unutmayın.

### Görev 1 - AWS hesabınıza giriş yapın

1. Eğer zaten AWS hesabınıza giriş yapmışsanız, bu görevi atlayabilirsiniz. <https://aws.amazon.com> adresine gidin.

2. AWS hesabınıza giriş yapmak için **Giriş Yap** butonuna tıklayın.



3. Kök kullanıcı olarak giriş yapmak için kayıtlı e-posta adresinizi girin.



## Sign in

**Root user**  
Account owner that performs tasks requiring unrestricted access. [Learn more](#)

**IAM user**  
User within an account that performs daily tasks.  
[Learn more](#)

**Root user email address**

[Redacted]

**Next**

4. Şifreyi girin ve **Giriş Yap** butonuna tıklayın. Bu sizi AWS Console Ana Sayfası'na götürecektir.



## Root user sign in ⓘ

Email: [Redacted]@l.com

**Password** [Forgot password?](#)

[Redacted]

**Sign in**

[Sign in to a different account](#)

[Create a new AWS account](#)

### Görev 2 - AWS Lambda işlevi oluşturma

1. AWS Konsol Ana Sayfası yüklenliğinde, üstteki arama çubuğuına **Lambda** yazın ve Lambda hizmetinin ilk seçenek olarak listelendiğini göreceksiniz. **Lambda'yı** seçin.

The screenshot shows the AWS search interface with the search term 'lambda' entered in the search bar. The results are filtered under the 'Services' category. The top result is 'Lambda' with the subtext 'Run Code without Thinking about Servers'. Other visible results include 'CodeBuild', 'AWS Signer', and 'Amazon Lex'. A red box highlights the 'Lambda' service card.

Search results for 'lambda'

Services (5)

- Features (2)
- Blogs (876)
- Documentation (64,560)
- Knowledge Articles (30)
- Tutorials (4)
- Events (3)
- Marketplace (287)

Lambda ☆  
Run Code without Thinking about Servers

CodeBuild ☆  
Build and Test Code

AWS Signer ☆  
Ensuring trust and integrity of your code

Amazon Lex ☆  
Build Voice and Text Chatbots

2. Varsayılan olarak Eylem seçildir. AWS Lambda işlevinizi oluşturmaya başlamak için Fonksiyon Oluştur'a tıklayın.

The screenshot shows the AWS Lambda Functions page. The left sidebar has 'Functions' selected. The main area shows a table with zero functions. The 'Create' button in the top right corner is highlighted with a red box. The table columns are 'Function name', 'Description', and 'Package type'. A message at the bottom states 'There is no data to display.'

AWS Lambda

Dashboard

Applications

Functions

Additional resources

Code signing configurations

Layers

Replicas

Related AWS resources

Step Functions state machines

Lambda > Functions

Functions (0)

Last fetched 20 seconds ago

Actions

Create

Filter by tags and attributes or search by keyword

Function name	Description	Package type
There is no data to display.		

3. Kendi kodunuzu ekleyeceğiniz için Sıfırdan Yaz seçeneğini tercih edebilirsiniz.

## Create function Info

Choose one of the following options to create your function.

### Author from scratch

Start with a simple Hello World example.

### Use a blueprint

Build a Lambda application from sample code and configuration presets for common use cases.

### Container image

Select a container image to deploy for your function.

Bro  
rep  
Depl  
the /

4. Fonksiyonunuz için temel bilgileri sağlayın - fonksiyon adı, çalışma zamanı. Bir Node.js işlevi oluşturacaksınız. Bu nedenle çalışma zamanı **Node.js 16.x** olacaktır. Diğer ayarları varsayılan bırakın ve **Fonksiyon Oluştur** butonuna tıklayın.

## Basic information

### Function name

Enter a name that describes the purpose of your function.

helloworld

Use only letters, numbers, hyphens, or underscores with no spaces.

### Runtime Info

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Node.js 16.x

### Architecture Info

Choose the instruction set architecture you want for your function code.

x86\_64

arm64

## Permissions Info

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this

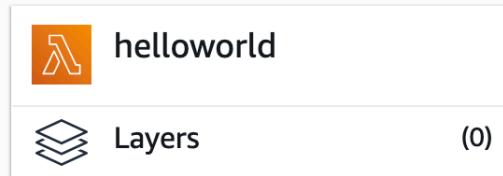
### ▶ Change default execution role

### ▶ Advanced settings

5. Fonksiyon oluşturulduktan birkaç saniye sonra, fonksiyon detayları sayfasını göreceksiniz.

# helloworld

## ▼ Function overview [Info](#)


[+ Add trigger](#)
[+ Add](#)
[Code](#)
[Test](#)
[Monitor](#)
[Configuration](#)
[Aliases](#)
[Versions](#)

6. Aynı sayfada aşağı kaydırarak **Kod** sekmesinde önceden yazılımın **Hello Lambda** kodunu görebilirsiniz.

The screenshot shows the AWS Lambda code editor interface. The tabs at the top are: Code (highlighted in orange), Test, Monitor, Configuration, Aliases, and Versions. Below the tabs, there's a toolbar with File, Edit, Find, View, Go, Tools, Window, a Test dropdown, and Deploy buttons. On the left, there's an Environment sidebar. The main area shows a file tree with a folder named 'helloworld' containing an 'index.js' file. The code in 'index.js' is:

```

1 exports.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 
```

7. Kodu aşağıdaki özel kod ile değiştirin. Bu kod, olaydan **name** parametresini alacak ve kişiselleştirilmiş bir **Merhaba** döndürecek. Scripti ekledikten sonra **Yayınla**'ya tıklayın.

```

exports.handler = async (event) => {
  const response = {
    statusCode: 200,
    body: JSON.stringify(`Hello ${event['name']} !`)
  };
  return response;
}; 
```

**Code** | **Test** | **Monitor** | **Configuration** | **Aliases** | **Versions**

## Code source Info

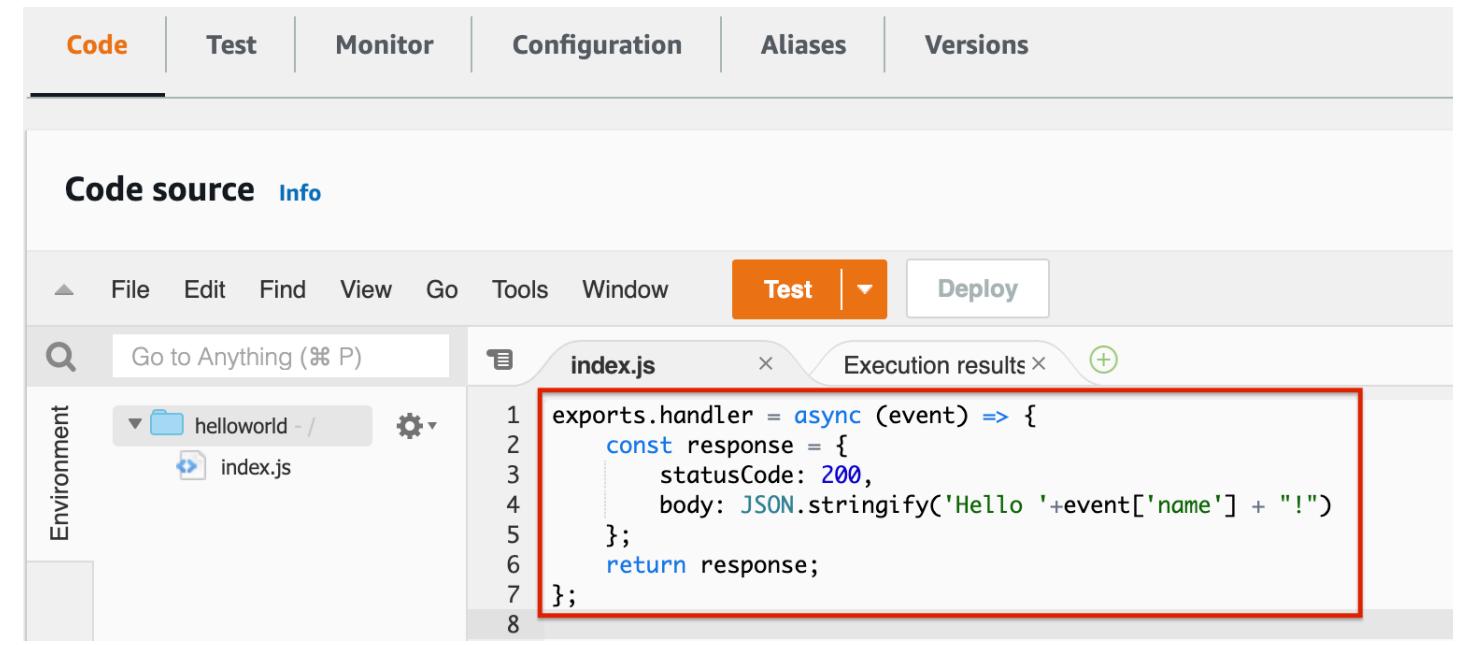
File Edit Find View Go Tools Window **Test** Deploy

Go to Anything (⌘ P)

Environment

index.js Execution results

```
1 exports.handler = async (event) => {
2     const response = {
3         statusCode: 200,
4         body: JSON.stringify('Hello ' + event['name'] + "!")
5     };
6     return response;
7 };
8
```



### Görev 3 - Lambda işlevini test et

1. Kod dağıtıldıktan sonra, bir olayı yapılandırmalı ve Lambda işlevinin çıkışını test etmelisiniz. **Test** butonunun yanında açılır menüye tıklayın ve **Test olayını yapılandır** seçeneğini seçin.

**Code** | **Test** | **Monitor** | **Configuration** | **Aliases** | **Versions**

## Code source Info

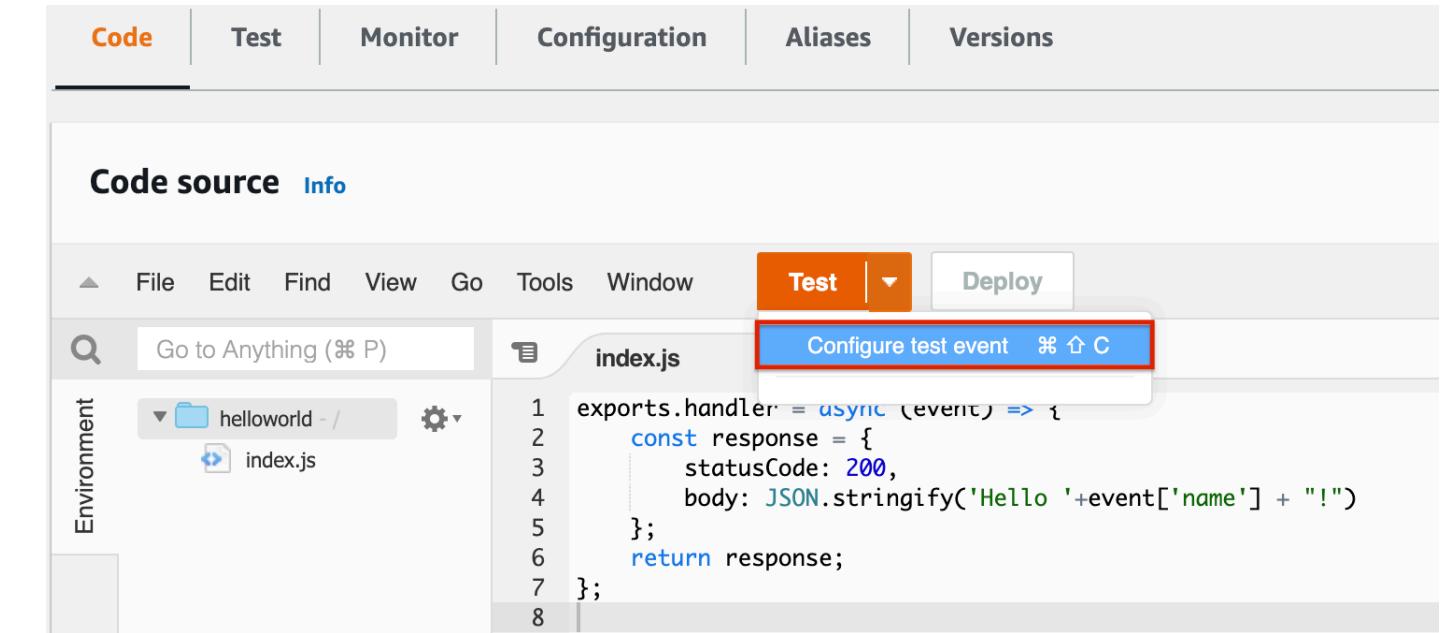
File Edit Find View Go Tools Window **Test** Deploy

Go to Anything (⌘ P)

Environment

index.js Configure test event ⌘ ⌘ C

```
1 exports.handler = async (event) => {
2     const response = {
3         statusCode: 200,
4         body: JSON.stringify('Hello ' + event['name'] + "!")
5     };
6     return response;
7 };
8
```



2. Olayınıza bir ad verin ve ardından aşağıdaki JSON'u girin veya kopyalayıp yapıştırarak olaya iletmek istediğiniz parametreyi ekleyin. Bu olay, Lambda işlevinizi test etmek istediğinizde tetiklenir. **Olay JSON'unu ekleyin** ve **Kaydet**'e tıklayın.

```
{  
    "name": "Eliot"  
}
```

### Test event action

Create new event

Edit saved event

#### Event name

sayHelloToMe

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

#### Event sharing settings

Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. [Learn more](#)

Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. [Learn more](#)

#### Template - optional

hello-world

Format JSON

#### Event JSON

```
1 {  
2   "name": "Eliot"  
3 }
```

Cancel

Save

3. Test'in yanındaki açılır menüye tekrar tıklayarak etkinliğin oluşturulup oluşturulmadığını kontrol edin.

Code    Test    Monitor    Configuration    Aliases    Versions

Code source    Info

File    Edit    Find    View    Go    Tools    Window    Test    Deploy

Configure test event    ⌘ ⌘ C

index.js

```
1 exports.handler = function(event, context) {  
2   // TODO implement  
3   const response = {  
4     statusCode: 200,  
5     body: JSON.stringify('Hello ' + event.name + '!')  
6   };  
7   return response;  
8 };
```

Environment

Go to Anything (⌘ P)

hello-world / index.js

4. Lambda işlevini çağırıkmak ve yanıtını görmek için Test'e tıklayın. Aşağıdaki görüntüde gösterildiği gibi yanıt görmelisiniz.

**Code** | Test | Monitor | Configuration | Aliases | Versions

**Code source** [Info](#)

File Edit Find View Go Tools Window **Test** Deploy

Go to Anything (⌘ P) index.js Execution result: +

Environment hello-world / index.js

Execution results

**Test Event Name** sayHelloToMe

**Response**

```
{ "statusCode": 200, "body": "\"Hello Elliot!\""}  
Function Logs  
START RequestId: cdb0ba6e-fa1f-47e0-bf60-57b6a6d63a97 Version: $LATEST  
END RequestId: cdb0ba6e-fa1f-47e0-bf60-57b6a6d63a97  
REPORT RequestId: cdb0ba6e-fa1f-47e0-bf60-57b6a6d63a97 Duration: 8.26 ms Billed Duration: 9 ms Memory Size: 128 MB  
Request ID  
cdb0ba6e-fa1f-47e0-bf60-57b6a6d63a97
```

#### Görev 4 - Lambda fonksiyonunu sil

1. Artık bir Lambda fonksiyonu oluşturduğunuza ve başarıyla test ettiğinizde onu silebilirsiniz. Sağ tıkteki **Eylem** menüsünde tıklayın ve **sil** seçeneğini seçin.

Lambda > Functions > hello-world

## hello-world

▼ Function overview [Info](#)

 **hello-world**  
 Layers (0)

+ Add trigger + Add destination

**Code** | Test | Monitor | Configuration | Aliases | Versions

**Code source** [Info](#)

File Edit Find View Go Tools Window **Test** Deploy

Go to Anything (⌘ P) index.js Execution result: +

Environment hello-world / index.js

Execution results

**Test Event Name** sayHelloToMe

**Response**

```
{ "statusCode": 200, "body": "\"Hello Elliot!\""}  
Function Logs  
START RequestId: cdb0ba6e-fa1f-47e0-bf60-57b6a6d63a97 Version: $LATEST  
END RequestId: cdb0ba6e-fa1f-47e0-bf60-57b6a6d63a97  
REPORT RequestId: cdb0ba6e-fa1f-47e0-bf60-57b6a6d63a97 Duration: 8.26 ms Billed Duration: 9 ms Memory Size: 128 MB  
Request ID  
cdb0ba6e-fa1f-47e0-bf60-57b6a6d63a97
```

2. Onay isteğinde bulunduğuanda, eylemi silmek istediğiniz onaylayabilirsiniz.

Function ARN  
arn:aws:lambda:ap-south-1:294444444444

Delete function hello-world

⚠ Deleting a function permanently removes the function code. The related logs, roles, test event schemas, and triggers are retained in your account.

Cancel Delete

Tebrikler! İlk AWS Lambda fonksiyonunuza oluşturmuş olduğunuz.

### Eğitim detayları

Yazar: Lavanaya T S

Katkıda Bulunanlar: Pallavi Rai



# Skills Network