AuburnHacks

Introduction to AWS

Cloud Background

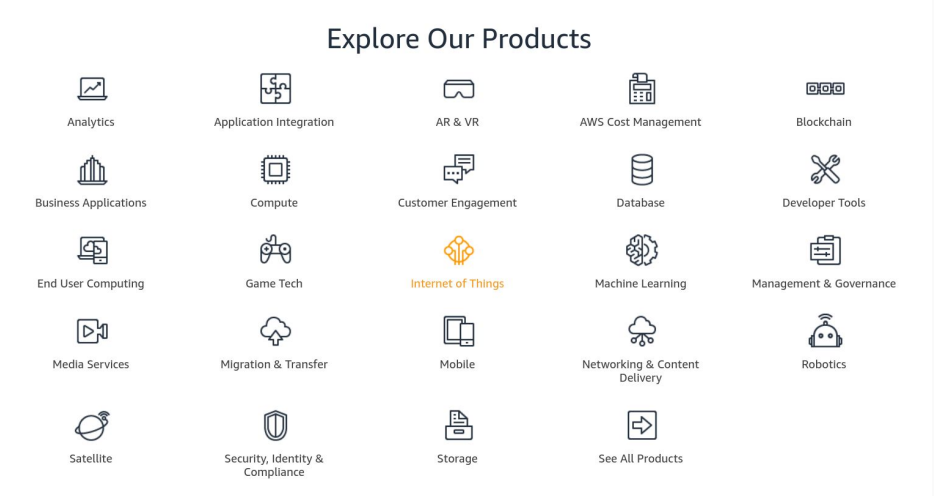
- Resilient: redundancy and failover systems
- On-demand: resources whenever you need them
- Elastic: able to scale resources on demand

What is AWS?

Amazon Web Services (AWS) is one of the largest cloud providers and offer many services for your cloud solutions

There is a lot of documentation and numerous tutorials for you to get started on something today.

Some of their clients include big names like Netflix, Airbnb, and Adobe



Getting started

1. Sign up for a new account:

https://portal.aws.amazon.com/billing/signup#/start

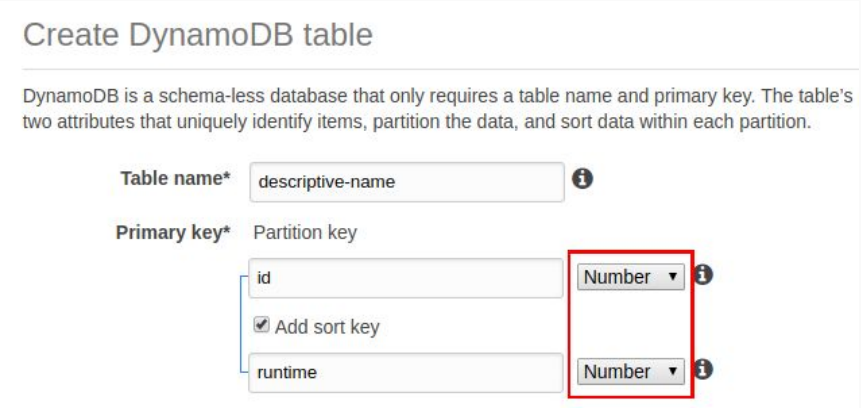
Make sure to write down or save your Access key ID and secret access key

2. Download the code that is being used in this tutorial:

https://github.com/auburnhacks/hacker-resources/tree/master/aws_techtalk

DynamoDB

1. Click ‘Create table’
2. Give your table a descriptive name
3. Name the Partition key ‘id’
4. Name the sort key ‘runtime’
5. Make sure to make both of these Numbers

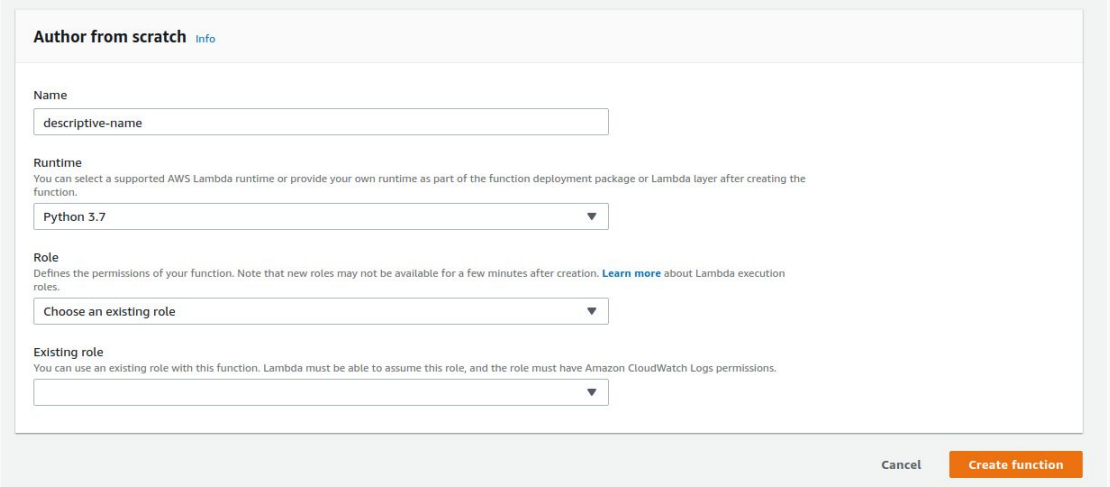


Adding items to DynamoDB

1. You can either add an item via the console or through the command line
2. First install the AWS CLI:
   `sudo apt-get install awscli` (or with your other installer if not on linux)
3. Configure:
   `aws configure`
   Then enter your access key ID, secret, region name, and output format
4. Write the items to the table
   `aws dynamodb batch-write-item --request-items file://items.json`

Lambda

1. Click ‘Create function’
2. Give your function a descriptive name
3. Select language
4. Select ‘Create a custom role’. A new window will pop up taking you to the IAM console

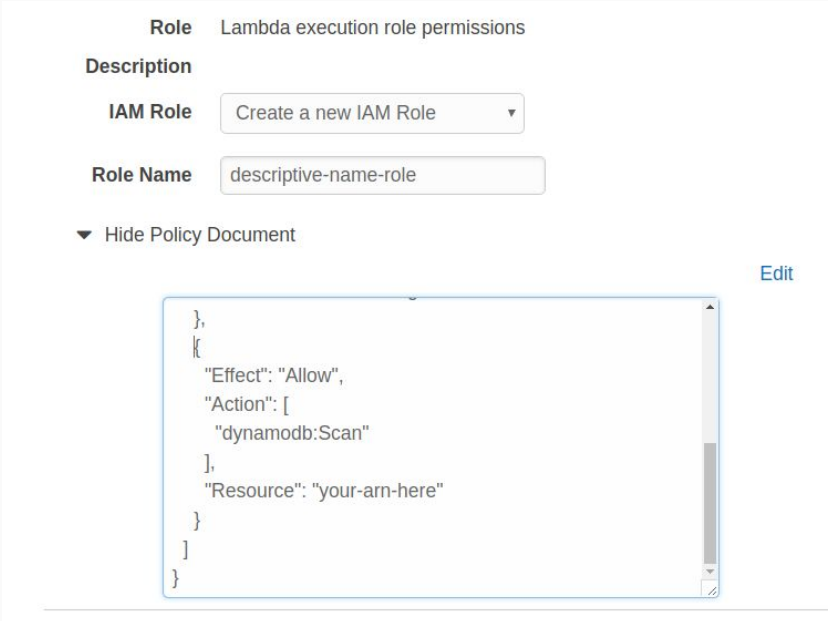


Create new execution role

1. Give your role a descriptive name
2. Edit the policy document
3. Add this in the ‘Statement’ array:

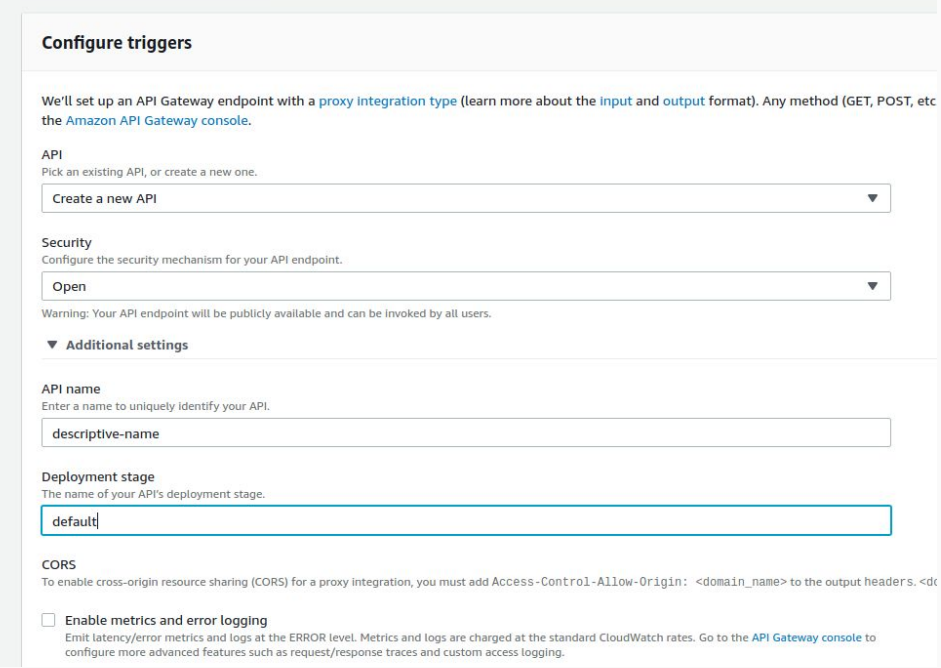
```
{
        "Effect": "Allow",
        "Action": [
                "dynamodb:Scan"
        ],
        "Resource": "your-arn-here"
}
```

(You can find your resource arn in the overview section of your table)



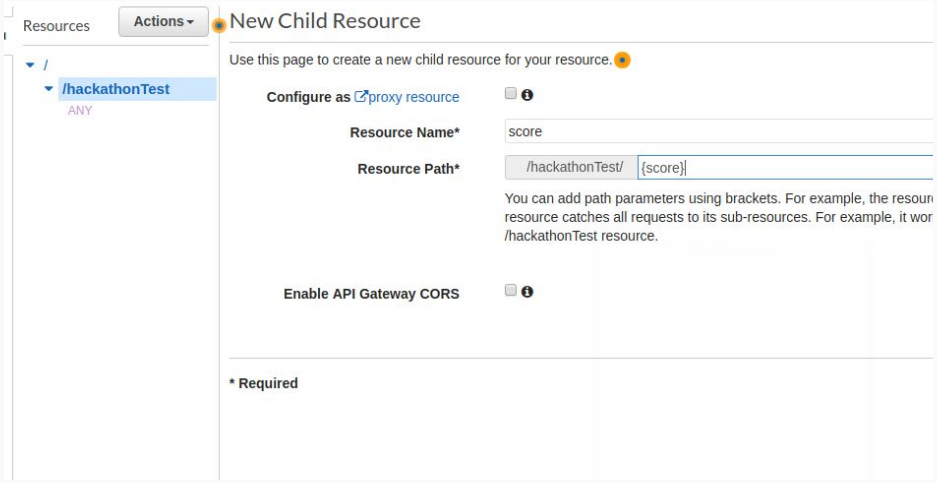
Lambda

1. Select APIGateway from the left
2. For ‘Security’ just select Open for now
3. Give your API a descriptive name
4. The rest can stay default for now
5. Click ‘Add’, then click ‘Save’ at the top
6. Click on the blue link to open up your new API in APIGateway



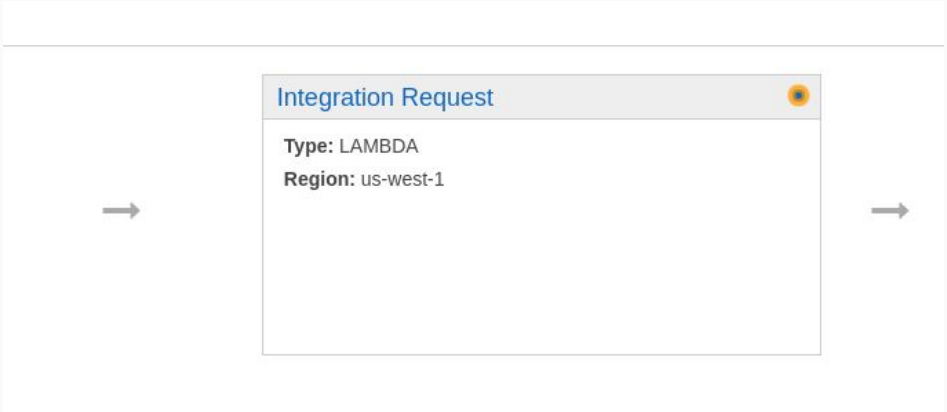
APIGateway

1. Click on ‘Actions’ then ‘Create Resource’
2. Name your resource ‘score’
3. Make the ‘Resource Path’ ‘{score}
4. Click ‘Create Resource’



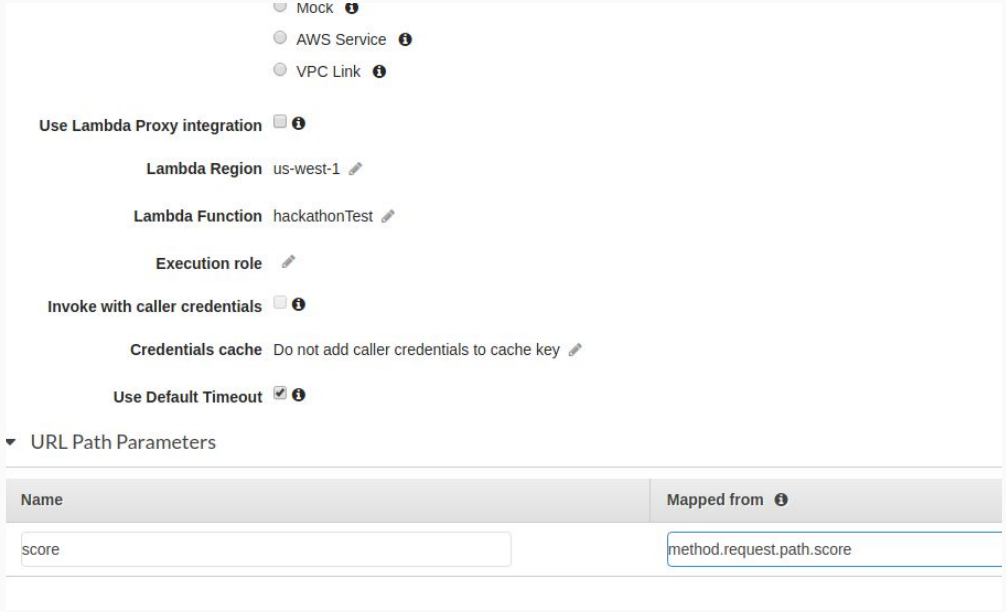
APIGateway

1. Click on ‘Actions’ then ‘Create Method’ and select ‘GET’
2. In ‘Lambda Function’ type the name of your Lambda function
3. Click Save and then OK
4. Next, click ‘Integration Request’



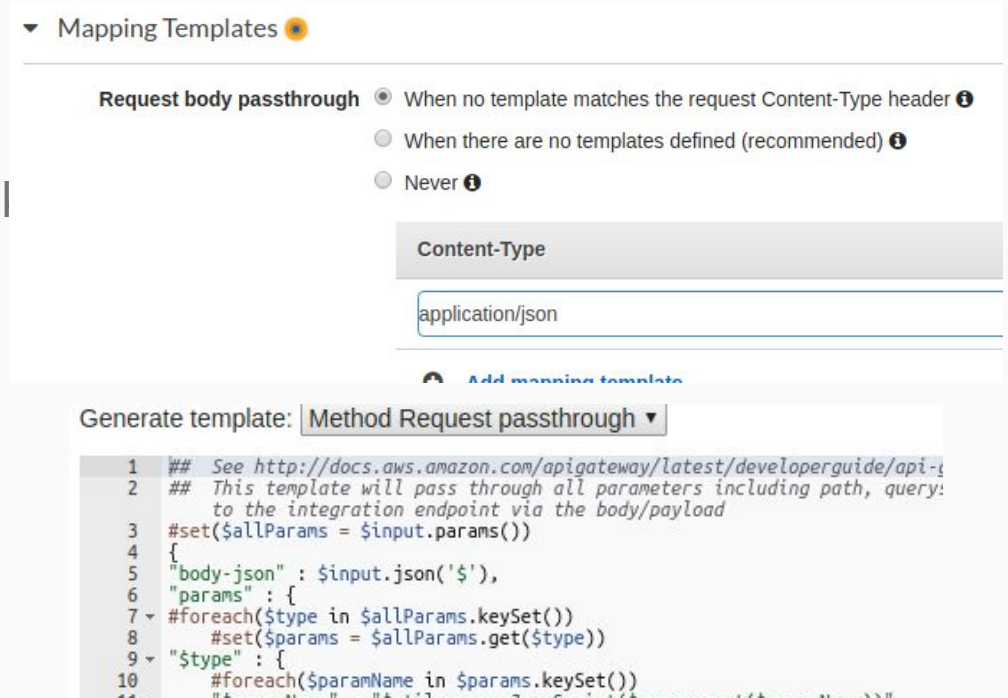
APIGateway

1. Click on ‘URL Path Parameters’
2. Type ‘score’ under Name
3. Type the path that you want to capture the path parameter from ‘method.request.path.score’
4. Click the check mark



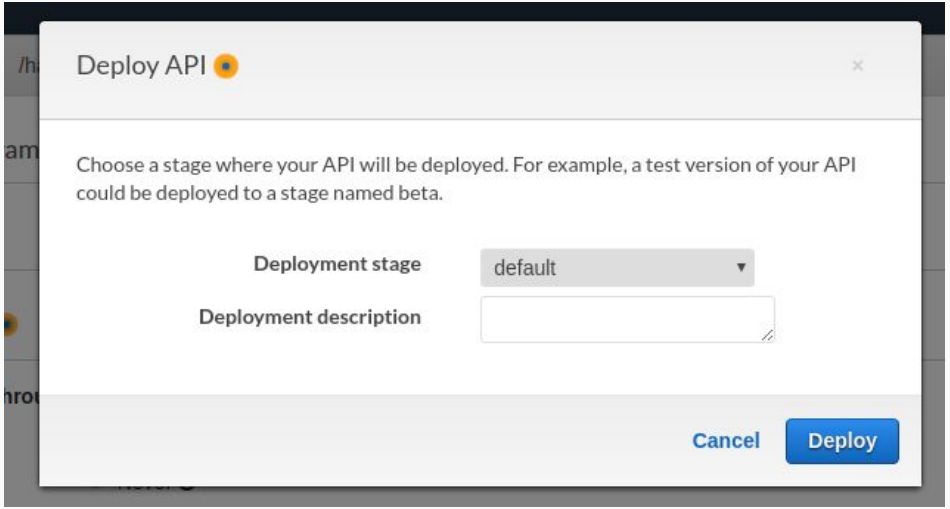
APIGateway

1. Click on ‘Mapping Templates’
2. Click ‘Add mapping template’
3. Type ‘application/json’ and clicl the check mark
4. Click ‘Yes, secure this integration’
5. Next, click on ‘Generate template:’ and select ‘Method Request passthrough’
6. Click Save



APIGateway

1. Click on ‘Actions’ then click on ‘Deploy API’
2. Choose ‘default’ as the deployment stage or create a new one for testing or something
3. Click Deploy
4. Now you can click on your blue invoke URL



Code

Remember that the code for this project can be found here:
https://github.com/auburnhacks/hacker-resources/tree/master/aws_techtalk

Thanks for attending!