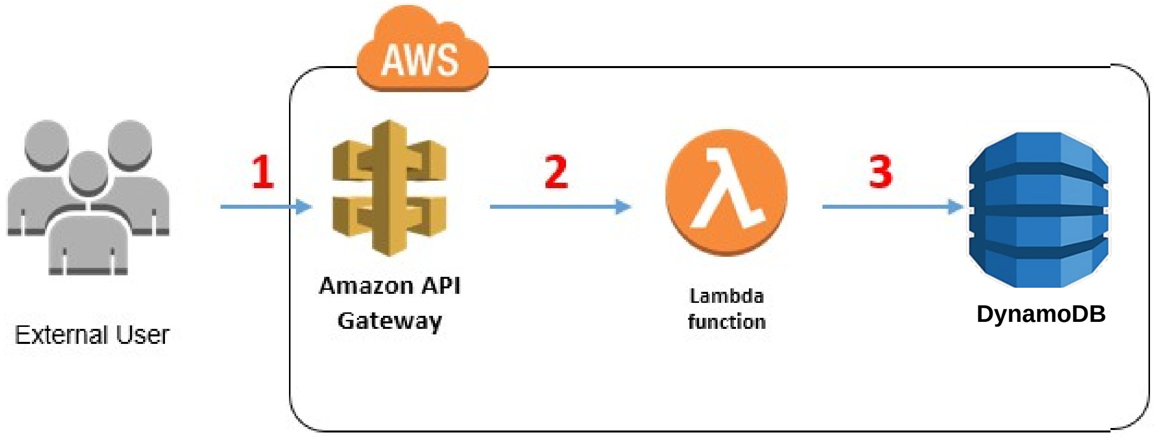
CLOUD COMPUTING PROJECT

WATCHES WEBSERVICES

PART III

Part III - Objectives

- Deploy a Serverless API in the cloud
  - Using AWS PaaS/FaaS !
    - API Gateway
    - AWS Lambda
    - DynamoDB



Tasks Overview

- Create Info Service v2 in Python
  - Replace MySQL by AWS DynamoDB (NoSQL)
- Configure AWS DynamoDB
  - Insert the DB (`watches.json` in repository)
- Deploy the code on AWS Lambda
- Deploy API on AWS API Gateway
  - AWS API Gateway → AWS Lambda → AWS DynamoDB
- One primary goal is to automate all the deployment process from command line
  - However you can still use the GUI to test/learn how things work

Info Service v2

- `info_openapi_v2.yaml`
  - Re-use v1, only 2 methods kept
- Replace MySQL by DynamoDB
  - https://aws.amazon.com/sdk-for-python/

    ```
    $ pip install boto3
    ```
  - https://boto3.amazonaws.com/v1/documentation/api/latest/guide/dynamodb.html
  - Local DynamoDB for development (Docker)
    - https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/DynamoDBLocal.Docker.html

Info Service v2 - Lambda Deploy

- You can deploy all your code in a single Python file
  - In fact you will have to create a Zip bundle that contains the Python code and also all the dependencies (if any)
    - https://docs.aws.amazon.com/lambda/latest/dg/lambda-python-how-to-create-deployment-package.html
- When you deploy the file to Lambda, you specify the handler function
  - The handler function is the specific function that will be called by that particular Lambda Function
  - You can deploy the same bundle for each Lambda Function, just setting a different handler

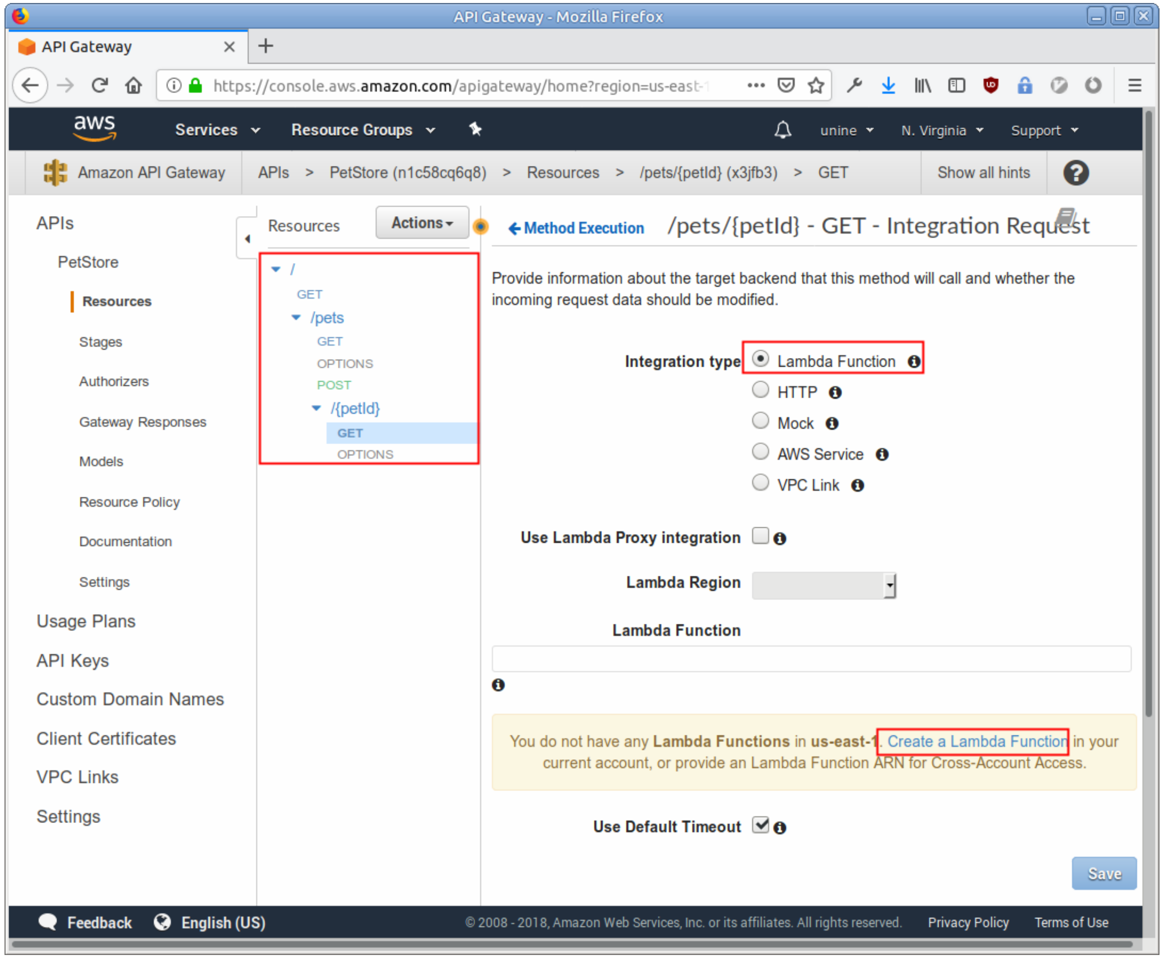
Info Service v2 - Lambda Deploy #2

- Solution 1
  - Adapt the code of Info Service v2 for Lambda
    - Remove Flask/routing stuff
    - Extract parameters from event
  - Command line deployment using AWS CLI
- Solution 2 (recommended)
  - Don't change the Python code
  - Use Zappa to automatically adapt your Python/Flask code and deploy it in AWS
    - https://www.zappa.io/
    - Zappa will generate some binding code to call your Flask code without changing it

AWS API Gateway

- Create Info Service v2 API endpoints
  - Create it directly from OpenAPI spec
    - Ignore import warning
  - No SSL
  - No Authentication
  - No cache

    → Keep default options
- Connect it with AWS Lambda
  - Select Lambda function to call
  - Use Lambda Proxy integration
- Deploy your API
  - To have a public endpoint



/pets/{petId} - GET - Integration Request

Provide information about the target backend that this method will call and whether the incoming request data should be modified.

Integration type [✓ Lambda Function] HTTP Mock AWS Service VPC Link

Use Lambda Proxy integration ☐

Lambda Region

Lambda Function

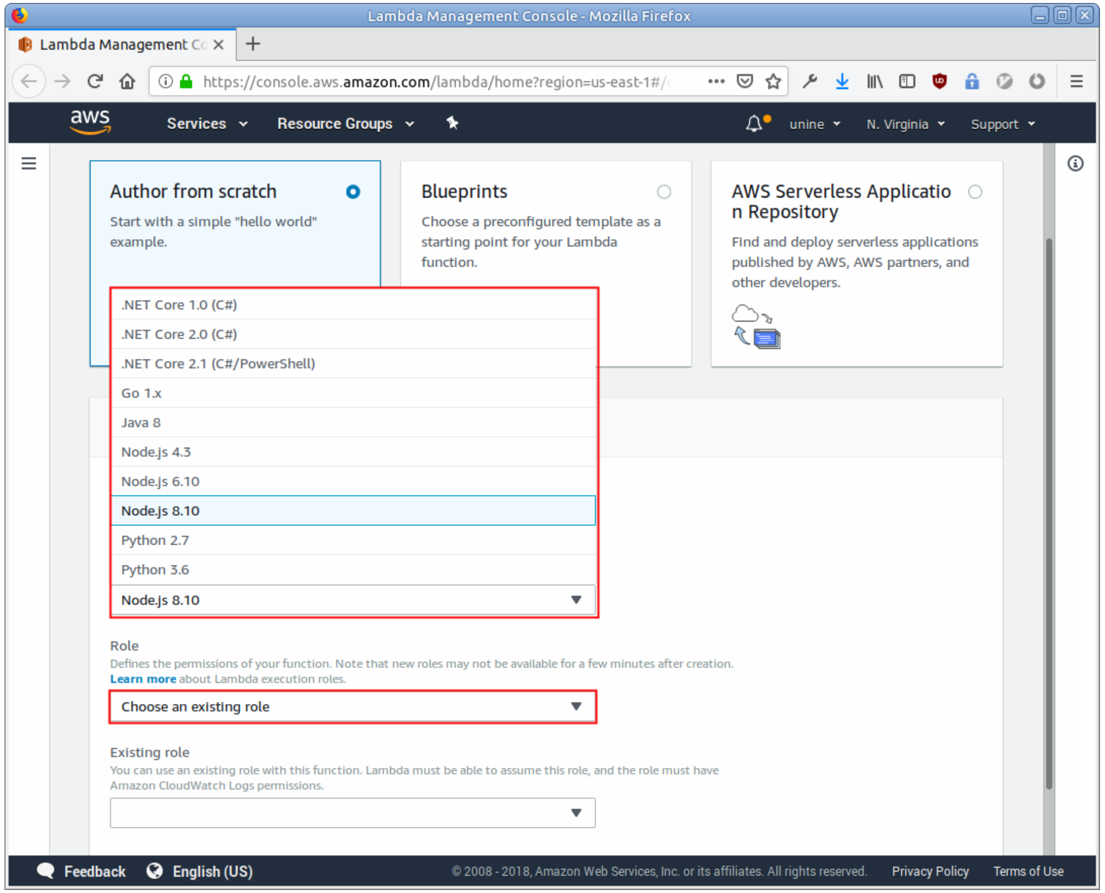
You do not have any **Lambda Functions** in **us-east-1**. Create a Lambda Function in your current account, or provide an Lambda Function ARN for Cross-Account Access.

Use Default Timeout ✓

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Lambda

- Lambda will be the binding between the API Gateway and DynamoDB
  - Lambda function must have the rights to access DynamoDB
  - Create a new IAM role for Lambda (auto)
    - Then edit the role to add DynamoDB read access


Lambda Management Console - Mozilla Firefox

https://console.aws.amazon.com/lambda/home?region=us-east-1#/

Author from scratch
Start with a simple "hello world" example.

Blueprints
Choose a preconfigured template as a starting point for your Lambda function.

AWS Serverless Application Repository
Find and deploy serverless applications published by AWS, AWS partners, and other developers.

.NET Core 1.0 (C#)
.NET Core 2.0 (C#)
.NET Core 2.1 (C#/PowerShell)
Go 1.x
Java 8
Node.js 4.3
Node.js 6.10
Node.js 8.10
Python 2.7
Python 3.6

Node.js 8.10

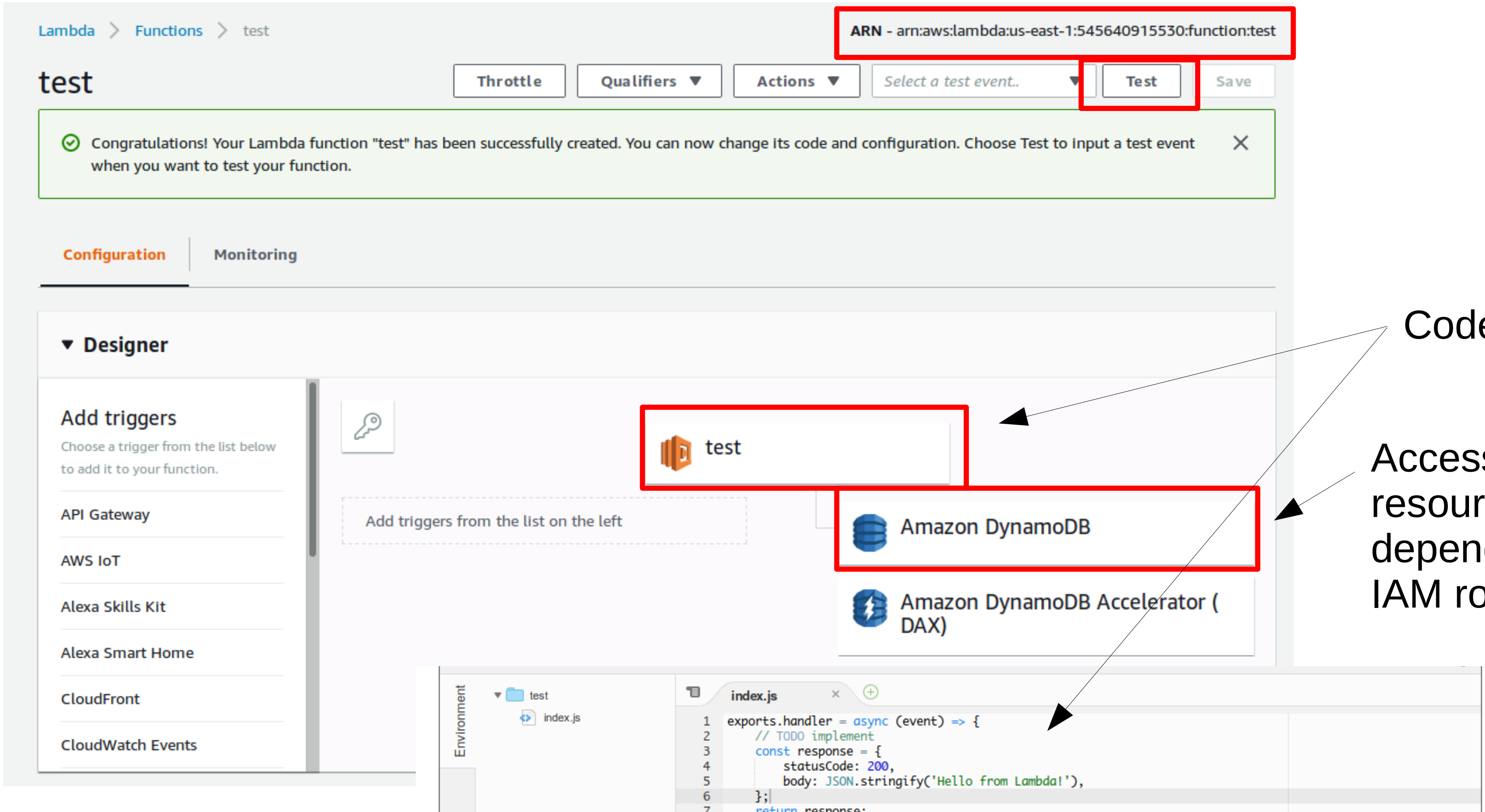
Role
Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation.
Learn more about Lambda execution roles.

Choose an existing role

Existing role
You can use an existing role with this function. Lambda must be able to assume this role, and the role must have Amazon CloudWatch Logs permissions.

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Lambda #2



Lambda > Functions > test

ARN - arn:aws:lambda:us-east-1:545640915530:function:test

test

Throttle | Qualifiers ▼ | Actions ▼ | Select a test event.. ▼ | Test | Save

Congratulations! Your Lambda function "test" has been successfully created. You can now change its code and configuration. Choose Test to input a test event when you want to test your function.

Configuration | Monitoring

▼ Designer

Add triggers

Choose a trigger from the list below to add it to your function.

- API Gateway
- AWS IoT
- Alexa Skills Kit
- Alexa Smart Home
- CloudFront
- CloudWatch Events

test

Add triggers from the list on the left

Amazon DynamoDB

Amazon DynamoDB Accelerator (DAX)

Environment

test
index.js

index.js

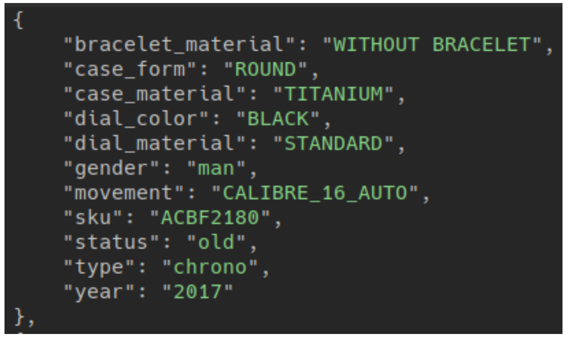
```
exports.handler = async (event) => {
    // TODO implement
    const response = {
        statusCode: 200,
        body: JSON.stringify('Hello from Lambda!'),
    };
    return response;
```

Code

Accessible resources depends of IAM role

DynamoDB

- NoSQL DB → Store JSON documents
  - https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.CoreComponents.html
  - https://aws.amazon.com/blogs/database/choosing-the-right-dynamodb-partition-key/
- Create a table 'project1'
  - With watches fields
    - sku → primary key (partition key)
    - All attributes as string type 'S', except year 'N'
  - https://docs.aws.amazon.com/cli/latest/userguide/cli-dynamodb.html



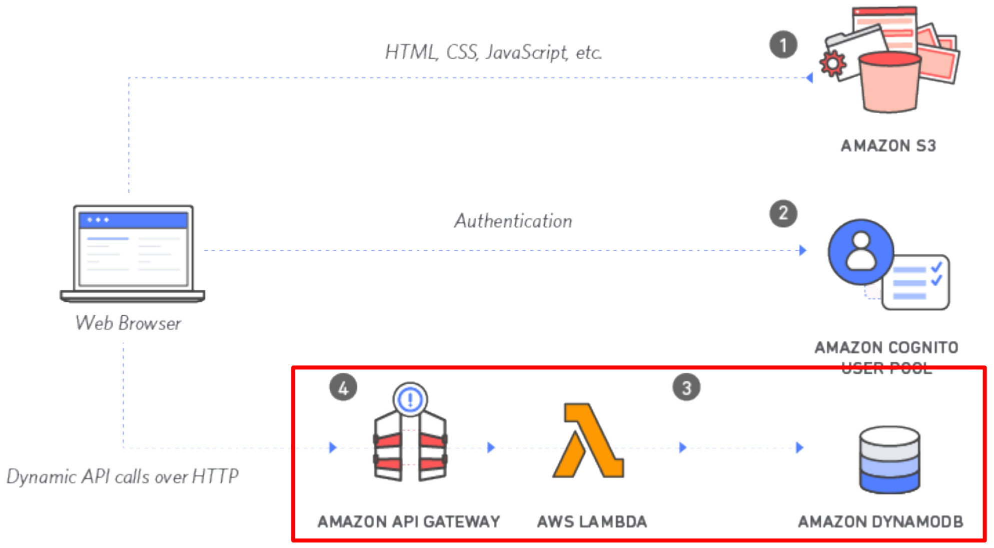
```
{
    "bracelet_material": "WITHOUT BRACELET",
    "case_form": "ROUND",
    "case_material": "TITANIUM",
    "dial_color": "BLACK",
    "dial_material": "STANDARD",
    "gender": "man",
    "movement": "CALIBRE_16_AUTO",
    "sku": "ACBF2180",
    "status": "old",
    "type": "chrono",
    "year": "2017"
},
```

DynamoDB - Import watches data

- Table exported from MySQL in JSON → **watches.json provided on git**
  - Write a small app that read it and send records in DynamoDB
    - Use AWS SDK (available in many languages)
      - Ruby
        - https://docs.aws.amazon.com/sdk-for-ruby/v3/developer-guide/dynamo-example-load-table-items-from-json.html
      - Javascript
        - https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/dynamodb-example-table-read-write.html
        - https://stackoverflow.com/questions/32944920/how-to-insert-json-in-dynamodb
  - Or use a script and CLI (see link on previous page)
  - **I recommend using the ruby example, it can be straightforwardly be modified to fit your needs**

Similar tutorial

- https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/
  - Module 3: Serverless Service Backend

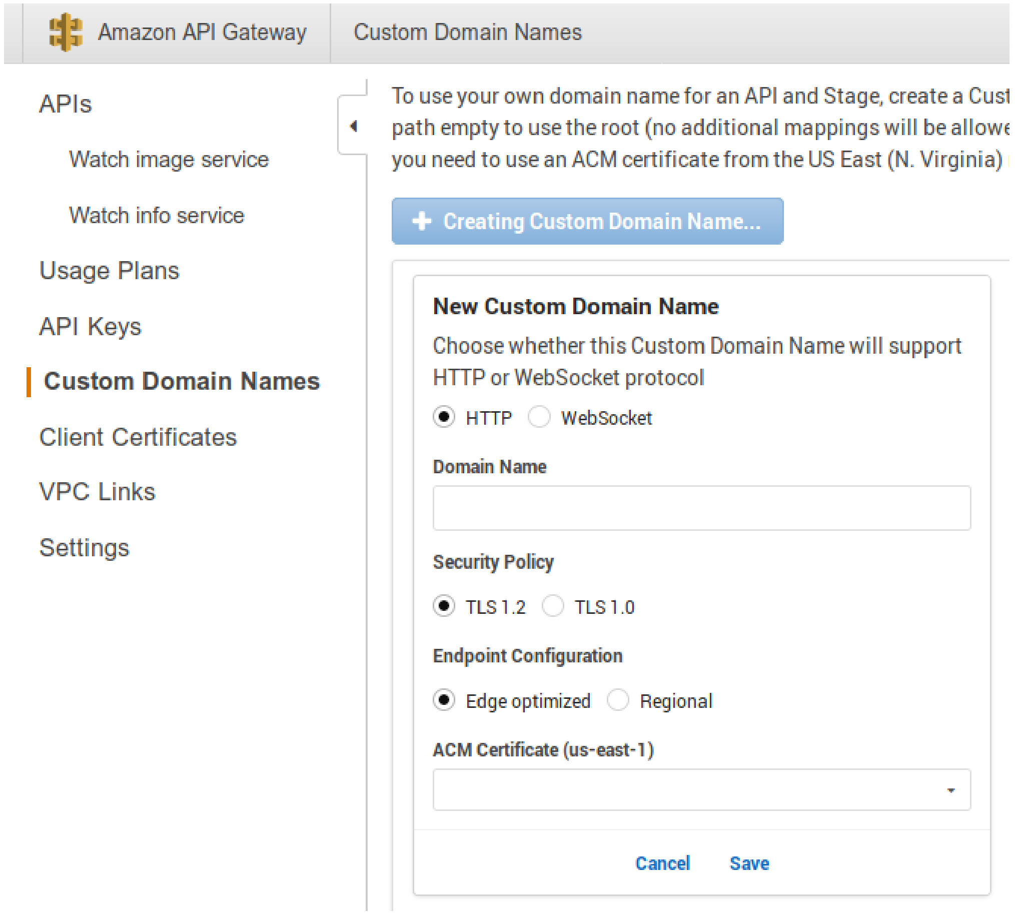


Domain (optional)

- Additional optional steps to have a production API
- If you have an available domain name
  - Consider `my-domain.ch` as an example
  - If you want to buy one it costs < 10$ / year
- Set AWS as the name servers (DNS) for your domain
  - AWS Route 53
- Use AWS to generate a free SSL certificate for your domain
  - AWS Certificate Manager (ACM)

Domain (optional) #2

- In AWS API Gateway select use custom domain name for your API
  - Indicate `api-watches.my-domain.ch`
  - Select the SSL certificate
  - Setup prefix routing to route to your Info v2 API or Image v1 API
- Final result
  - `https://api-watches.my-domain.ch`
    - `/info/v2`
    - `/image/v1`
- With that architecture it is extremely simple to create new services or new versions in the future while conserving access to the previous ones



To use your own domain name for an API and Stage, create a Cust
path empty to use the root (no additional mappings will be allowe
you need to use an ACM certificate from the US East (N. Virginia)

New Custom Domain Name

Choose whether this Custom Domain Name will support HTTP or WebSocket protocol

● HTTP ○ WebSocket

Domain Name

Security Policy

● TLS 1.2 ○ TLS 1.0

Endpoint Configuration

● Edge optimized ○ Regional

ACM Certificate (us-east-1)

Deliverables - Info Service v2

- `/info-v2/`
  - `server.py`
    - Also files for local testing: `requirements.txt`, `run.sh`
  - `create_lambdas.sh`
    - Script doing the initial creation of the Lambda Functions
    - Set name, handler and execution resources
  - `update_lambdas.sh`
    - Script to execute in order to update the Lambda function each time you do a code change
- Notes
  - `create_lambdas.sh` and `update_lambdas.sh` must update all your Lambda Functions
  - If not using Zappa
    - They also must create/update the package bundle (`lambdas.zip`)
  - If using Zappa
    - Put the Zappa commands inside the deployments scripts
    - With Zappa there may be additional config files

Deliverables - Other

- `/info-v2/`
  - `create_api_gateway.sh`
    - Create the AWS API Gateway from the OpenAPI spec
    - Bind the REST endpoints to Lambda Functions
  - `initialize_dynamodb.sh`
    - Create an AWS DynamoDB table
    - Fill it with watches data (using `watches.json`)
  - README
    - The API endpoint (generated by AWS API Gateway when you deploy the API)
    - Additional infos as usual if something doesn't execute as expected
- Note
  - All the deployment scripts will use your AWS credentials from ENV vars or personal config, your credentials should not appear in the git repository

Delay - Grading

- Delay: 4 weeks
    - Deadline: 2019-12-11T23:59:59+02:00
- Grading:
    - Participation
        - 1 point
    - Working Architecture (tests via API public endpoint)
        - 3 points
    - Deployment scripts
        - 2 points

CLOUD COMPUTING

READ ASSIGNMENT

Lorenzo Leonini
lorenzo.leonini@unine.ch

Read assignment

- Read a paper from a recent top-tier conference
  - Mix of academic and industry authors
- Learn about cutting-edge research
- Prepare a presentation with other students as the target audience
- Improve presentation and summarization skills
- Have fun

Paper selection and assignment

- Each student picks 1 to 3 choices in decreasing order of interest
  - We assign papers in a first-come first-serve basis
  - If first choice already taken or paper too close already taken, we assign your second choice (or 3rd etc.)
  - Sending multiple choices directly reduces email overhead
- Selection of paper
  - From the conferences on the next page
  - Or from full-length papers who appeared at a top-level conference
    - Should be recent (max 2 years old)
- Send your selection by email

List of papers

- Papers
  - https://dblp.org/db/conf/cloud/socc2018
  - https://dblp.org/db/conf/eurosys/eurosys2019
- The selected paper should have a relation to Cloud Computing

Planning

- Propositions of conferences and papers
  - Today 2019-11-14
- Collection of students preferences
  - Deadline next Thursday 2019-11-21
- Confirmation of assignment
  - Thursday 2019-11-28
- Presentations
  - Thursday 2019-12-12 and 2019-12-19

Grading

- Grading by Lorenzo and Rémi
  - 1/8 of your final grade (project parts I, II and III = 3/8 → 50%)
    - We will send you the consolidated 50% grade as soon as Rémi has finished reviewing the project
- Clarity of the speech
  - Context and clear problem definition
  - Conciseness and simplicity
  - Ability to explain solution from the right level of abstraction (do not present any detail from the paper)
- Quality of the slides
  - Flow and structure
  - Use of appropriate diagrams
- Respect of time
- Answers to questions

Presentation rules

- Initial presentation order will be alphabetically
  - You can swap with someone else if you cannot be present at some specific date/time
- You must use your own deck of slides
  - Forbidden to copy/paste from paper or existing presentation, with exception for evaluation plots
- Presentation time will be 10 minutes
  - Then a few minutes for questions

Recommendations

- Your three main goals
  - Convince the audience this is an important problem
    - What is the context and why does it matter?
    - Why is there something to fix / improve? What is the problem?
    - Why is it not trivial problem to solve?
  - Give an idea of the solution and make the audience want to read the paper
    - What is the core idea of the authors?
    - What are the key challenges?
    - What are the tools and techniques used, and why?
  - Discuss limitations and future work

Suggestions for slides

- 10 minutes = about 7-8 slides
  - You will not be able to present every single detail of the paper
    - Keep a high level of abstraction
    - Focus on what you believe are the most interesting aspects
    - Clearly separate authors’ contributions and your opinion
- One possible way to do it
  - 1 slide: background
  - 2 slides: problem definition
  - 2 slides: solution
  - 2 slides: results
  - 1 slide: limitations and future work