Building an IoT Backend with AWS and Serverless

August 21, 2018

@carlosaln_dev
/in/carlosaln

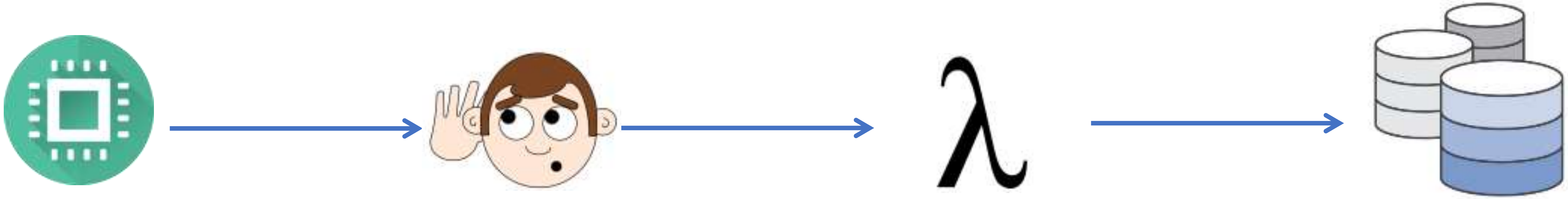
Carlos Lemus

Senior Cloud Engineer @ trek10

The Three Steps to Building an IoT Backend on AWS

1. Prototyping through the console
2. Using a framework
3. Simulating your devices

A Typical IoT Backend



IoT Device

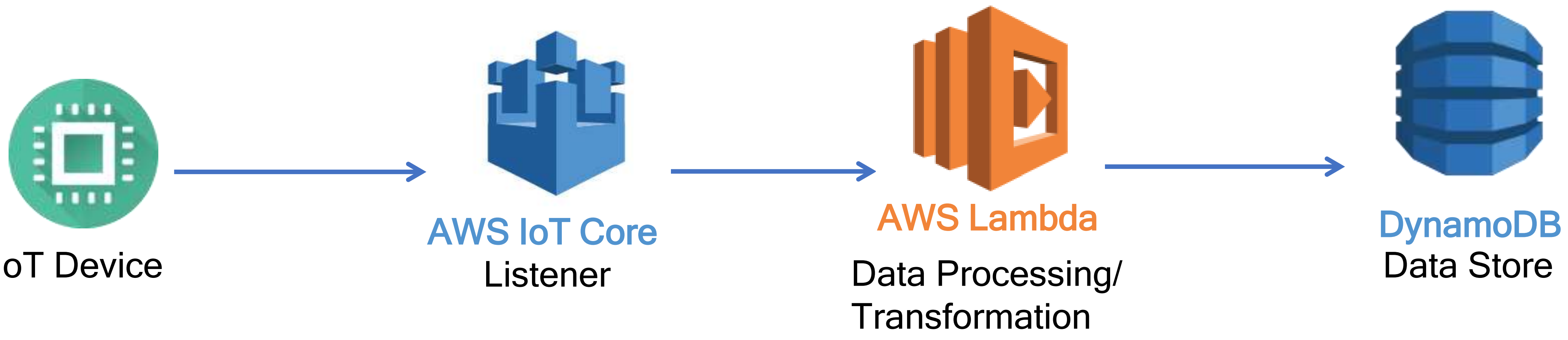
Listener

Data Processing/
Transformation

Data Store

Level 1:
Simple Console Prototyping

A Typical (AWS) IoT Backend



IoT Device

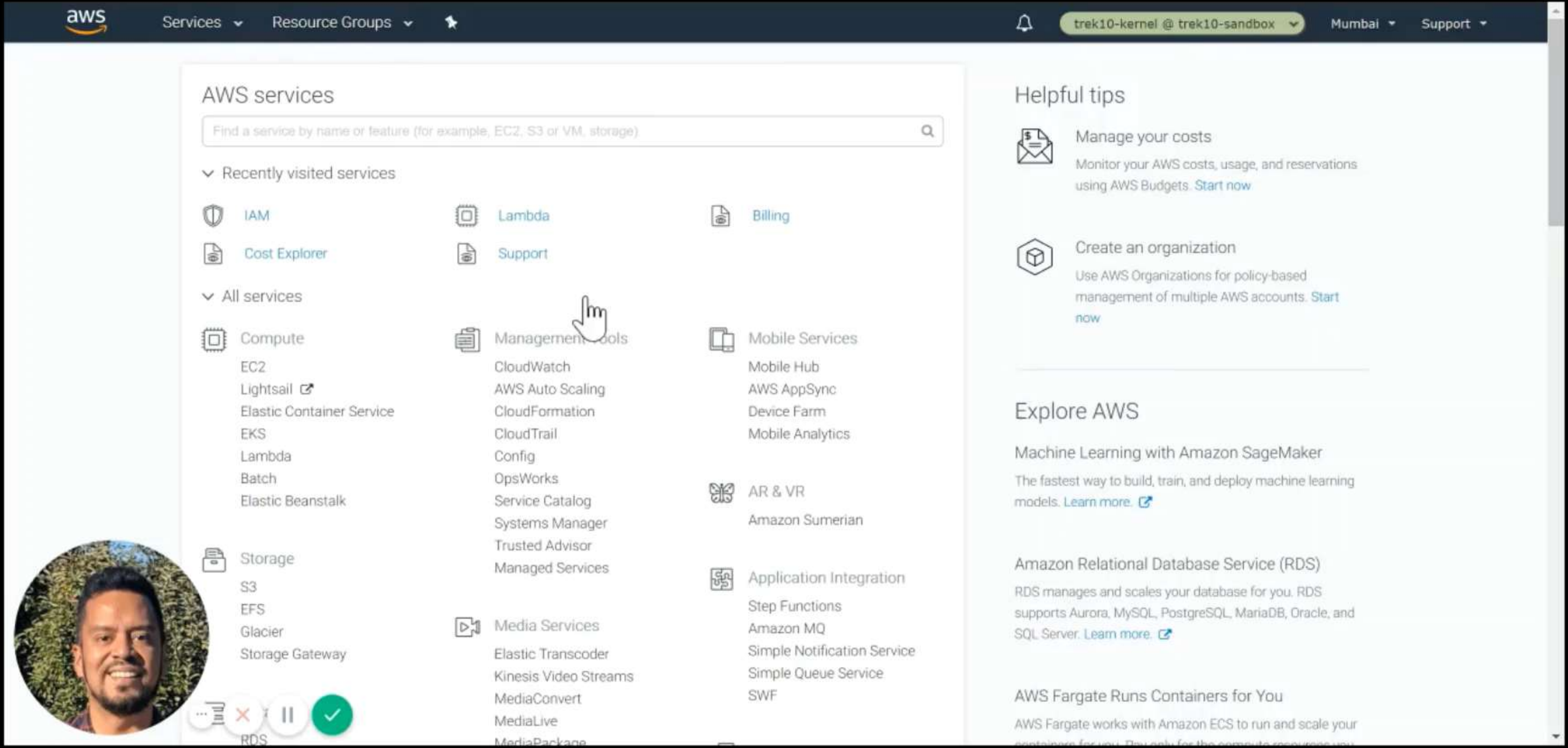
AWS IoT Core
Listener

AWS Lambda
Data Processing/
Transformation

DynamoDB
Data Store

AWS Lambda

- Serverless compute platform for code execution in response to events.
- Stateless
- Pay-per-use
- No hardware/OS/platform configuration



AWS services

Find a service by name or feature (for example, EC2, S3 or VM, storage)

Recently visited services

IAM
Lambda
Billing
Cost Explorer
Support

All services

Compute
EC2
Lightsail
Elastic Container Service
EKS
Lambda
Batch
Elastic Beanstalk

Storage
S3
EFS
Glacier
Storage Gateway

RDS

Management Tools
CloudWatch
AWS Auto Scaling
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Systems Manager
Trusted Advisor
Managed Services

Media Services
Elastic Transcoder
Kinesis Video Streams
MediaConvert
MediaLive
MediaPackage

Mobile Services
Mobile Hub
AWS AppSync
Device Farm
Mobile Analytics

AR & VR
Amazon Sumerian

Application Integration
Step Functions
Amazon MQ
Simple Notification Service
Simple Queue Service
SWF

Helpful tips

Manage your costs

Monitor your AWS costs, usage, and reservations using AWS Budgets. Start now

Create an organization

Use AWS Organizations for policy-based management of multiple AWS accounts. Start now

Explore AWS

Machine Learning with Amazon SageMaker

The fastest way to build, train, and deploy machine learning models. Learn more.

Amazon Relational Database Service (RDS)

RDS manages and scales your database for you. RDS supports Aurora, MySQL, PostgreSQL, MariaDB, Oracle, and SQL Server. Learn more.

AWS Fargate Runs Containers for You

AWS Fargate works with Amazon ECS to run and scale your

AWS IoT Core

- Data ingestion service
- Securely connect IoT devices to AWS resources

AWS IoT Core - Concepts

Message

→ The Payload, typically in JSON

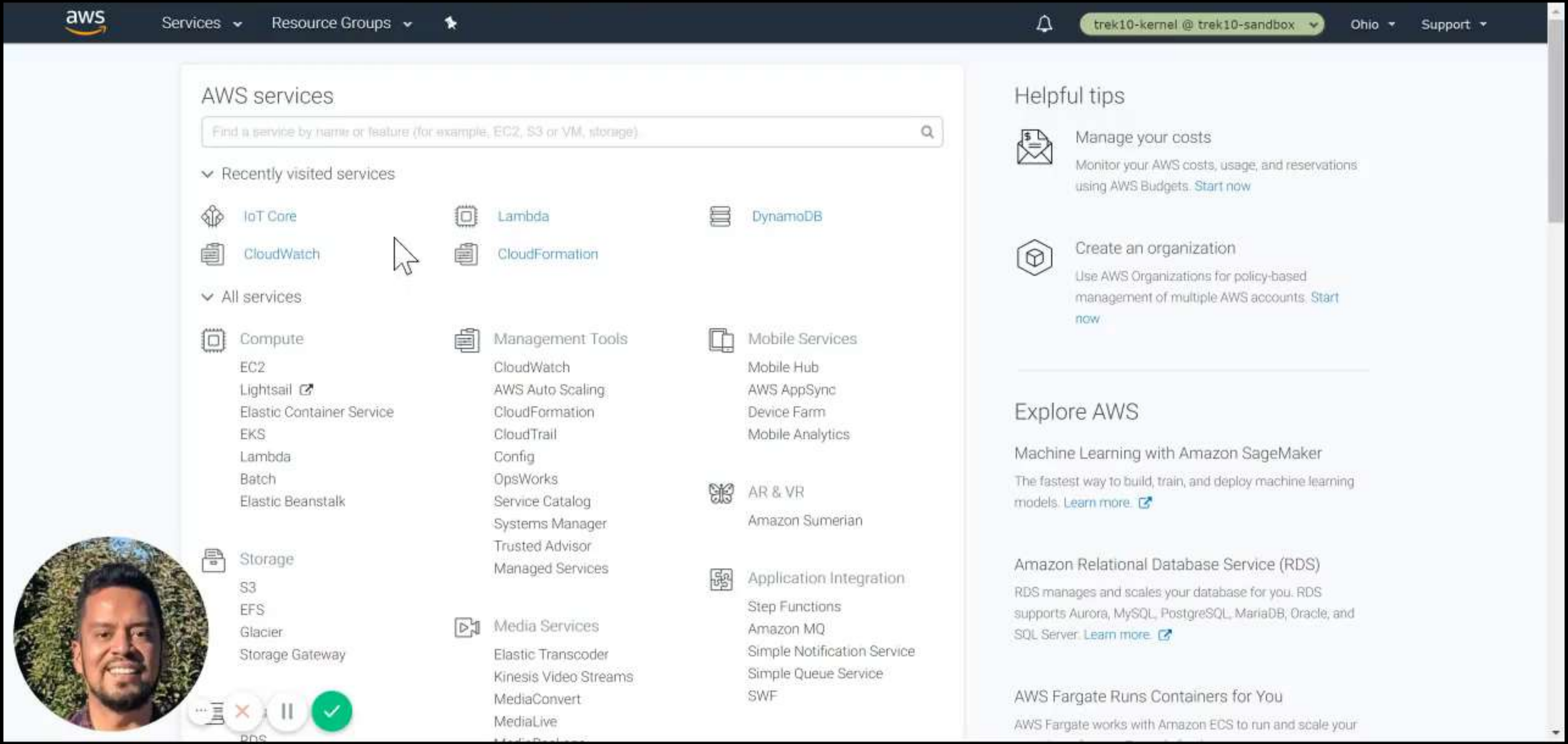
```
{
"timestamp": "2018-07-19 19:46:42.178000000",
"trip_id": "2b140fea-2e55-4d1f-9d27-474c493e7566",
"vin": "Y8CHZ0R48Z96YV0RM",
"name": "location",
"latitude": 38.8668,
"longitude": -77.44119
}
```

Topic

→ The “destination” where the message should go.

Rule

→ What to do with messages that hit a certain topic.



AWS services

Find a service by name or feature (for example, EC2, S3 or VM, storage)

Recently visited services

IoT Core
Lambda
DynamoDB
CloudWatch
CloudFormation

All services

Compute
EC2
Lightsail
Elastic Container Service
EKS
Lambda
Batch
Elastic Beanstalk

Storage
S3
EFS
Glacier
Storage Gateway

RDS

Management Tools
CloudWatch
AWS Auto Scaling
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Systems Manager
Trusted Advisor
Managed Services

Media Services
Elastic Transcoder
Kinesis Video Streams
MediaConvert
MediaLive

Mobile Services
Mobile Hub
AWS AppSync
Device Farm
Mobile Analytics

AR & VR
Amazon Sumerian

Application Integration
Step Functions
Amazon MQ
Simple Notification Service
Simple Queue Service
SWF

Helpful tips

Manage your costs

Monitor your AWS costs, usage, and reservations using AWS Budgets. Start now

Create an organization

Use AWS Organizations for policy-based management of multiple AWS accounts. Start now

Explore AWS

Machine Learning with Amazon SageMaker

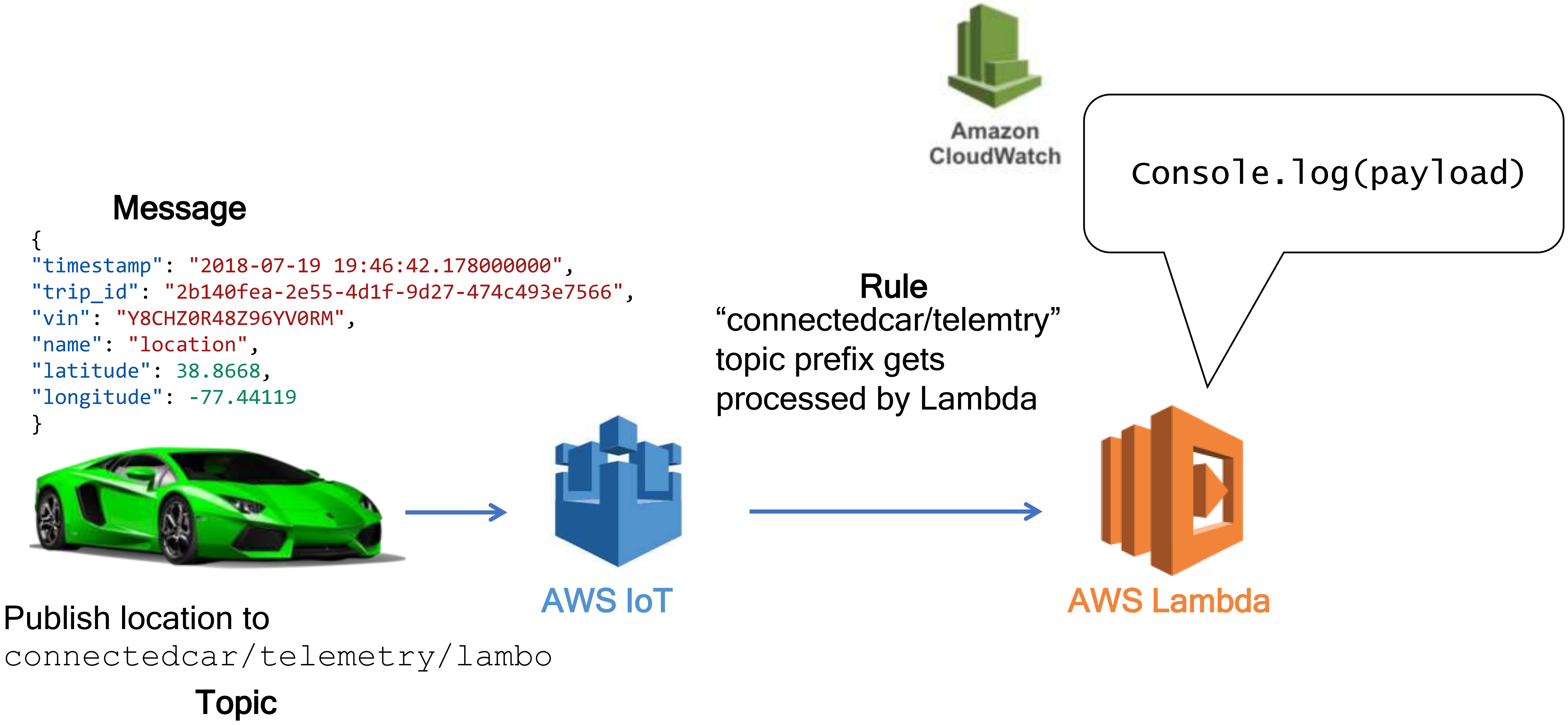
The fastest way to build, train, and deploy machine learning models. Learn more.

Amazon Relational Database Service (RDS)

RDS manages and scales your database for you. RDS supports Aurora, MySQL, PostgreSQL, MariaDB, Oracle, and SQL Server. Learn more.

AWS Fargate Runs Containers for You

AWS Fargate works with Amazon ECS to run and scale your



Message

```
{
"timestamp": "2018-07-19 19:46:42.178000000",
"trip_id": "2b140fea-2e55-4d1f-9d27-474c493e7566",
"vin": "Y8CHZ0R48Z96YV0RM",
"name": "location",
"latitude": 38.8668,
"longitude": -77.44119
}
```

Publish location to
`connectedcar/telemetry/lambo`

Topic

AWS IoT

Rule
“connectedcar/telemtry” topic prefix gets processed by Lambda

AWS Lambda

Amazon CloudWatch

`Console.log(payload)`

Testing your lambda function

Test by invoking the function

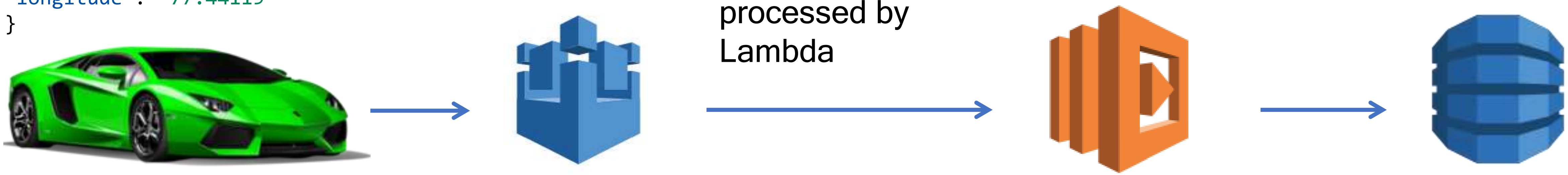
Test by pushing dummy data through IoT Core

Level 2:

Use a Framework deploy and test your code

Message

```
{
"timestamp": "2018-07-19 19:46:42.178000000",
"trip_id": "2b140fea-2e55-4d1f-9d27-474c493e7566",
"vin": "Y8CHZ0R48Z96YV0RM",
"name": "location",
"latitude": 38.8668,
"longitude": -77.44119
}
```



Rule

“connectedcar” topic prefix gets processed by Lambda

AWS IoT Core

AWS Lambda

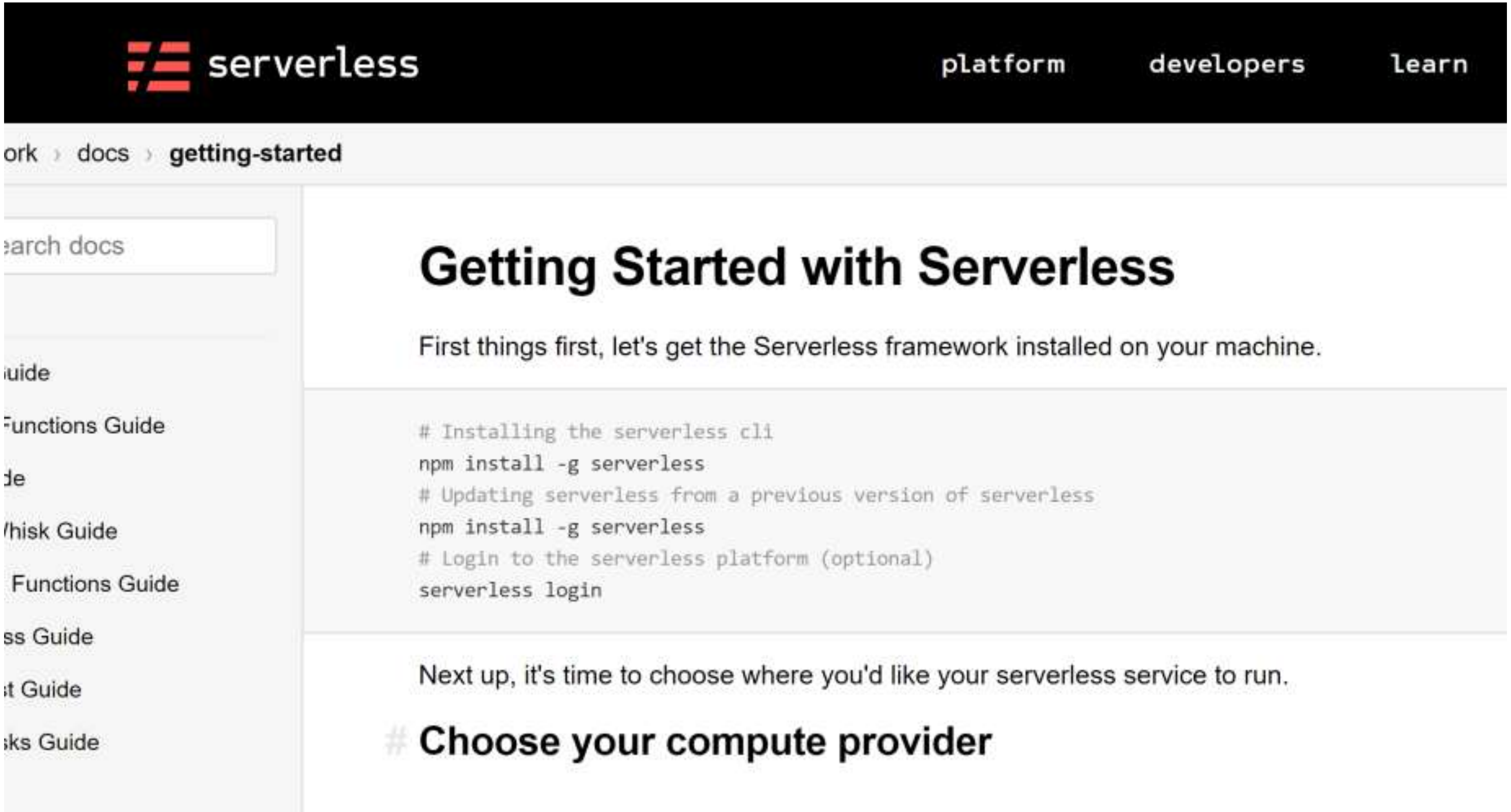
DynamoDB Table

Publish location to
`connectedcar/telemetry/lambo`

Topic

Serverless Framework Template and Code

http://serverless.com



Getting Started with Serverless

First things first, let's get the Serverless framework installed on your machine.

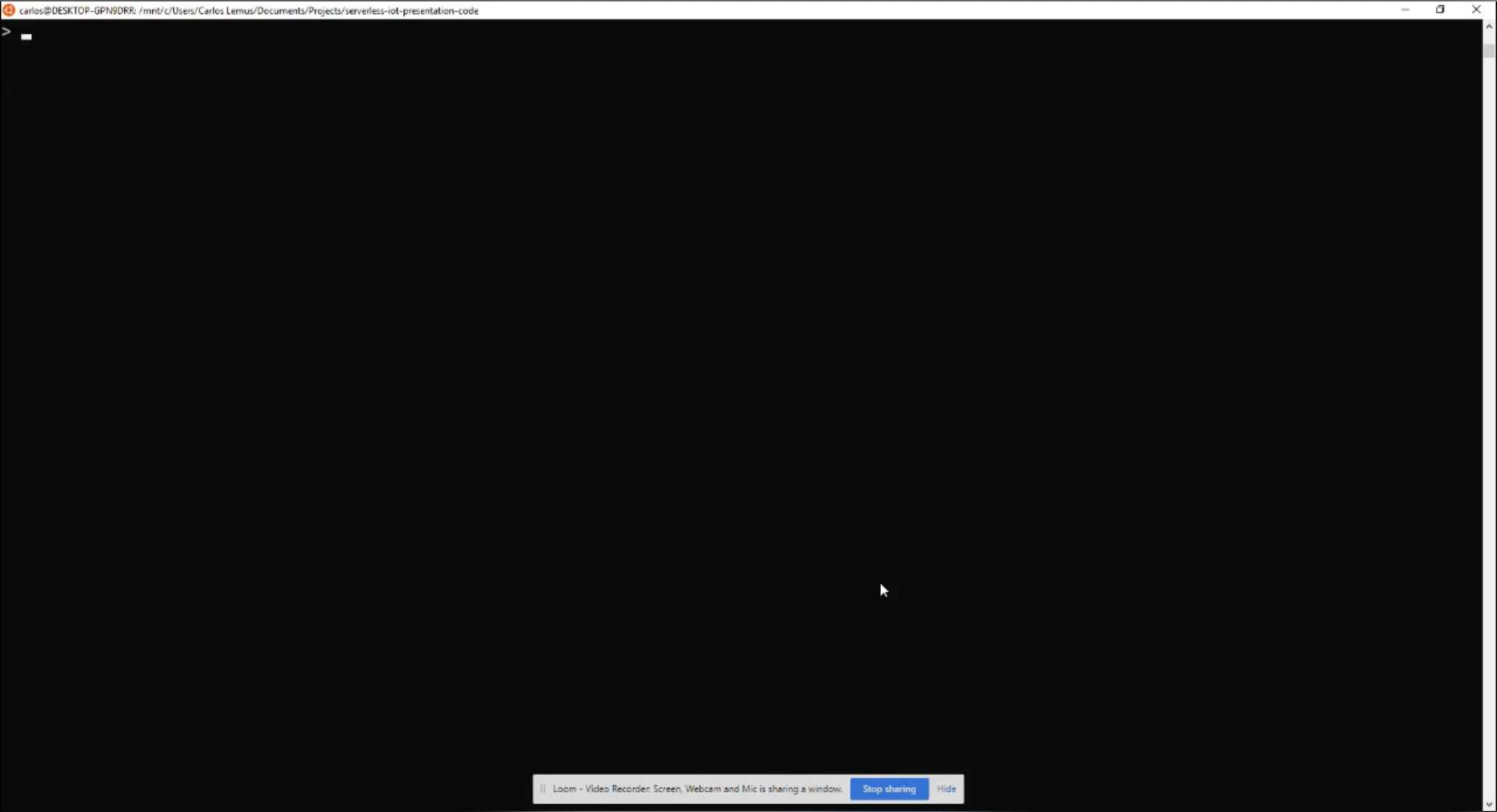
```
# Installing the serverless cli
npm install -g serverless
# Updating serverless from a previous version of serverless
npm install -g serverless
# Login to the serverless platform (optional)
serverless login
```

Next up, it's time to choose where you'd like your serverless service to run.

Choose your compute provider

Deploying a Serverless Project

```
serverless deploy –v --stage dev --region us-east-2
```



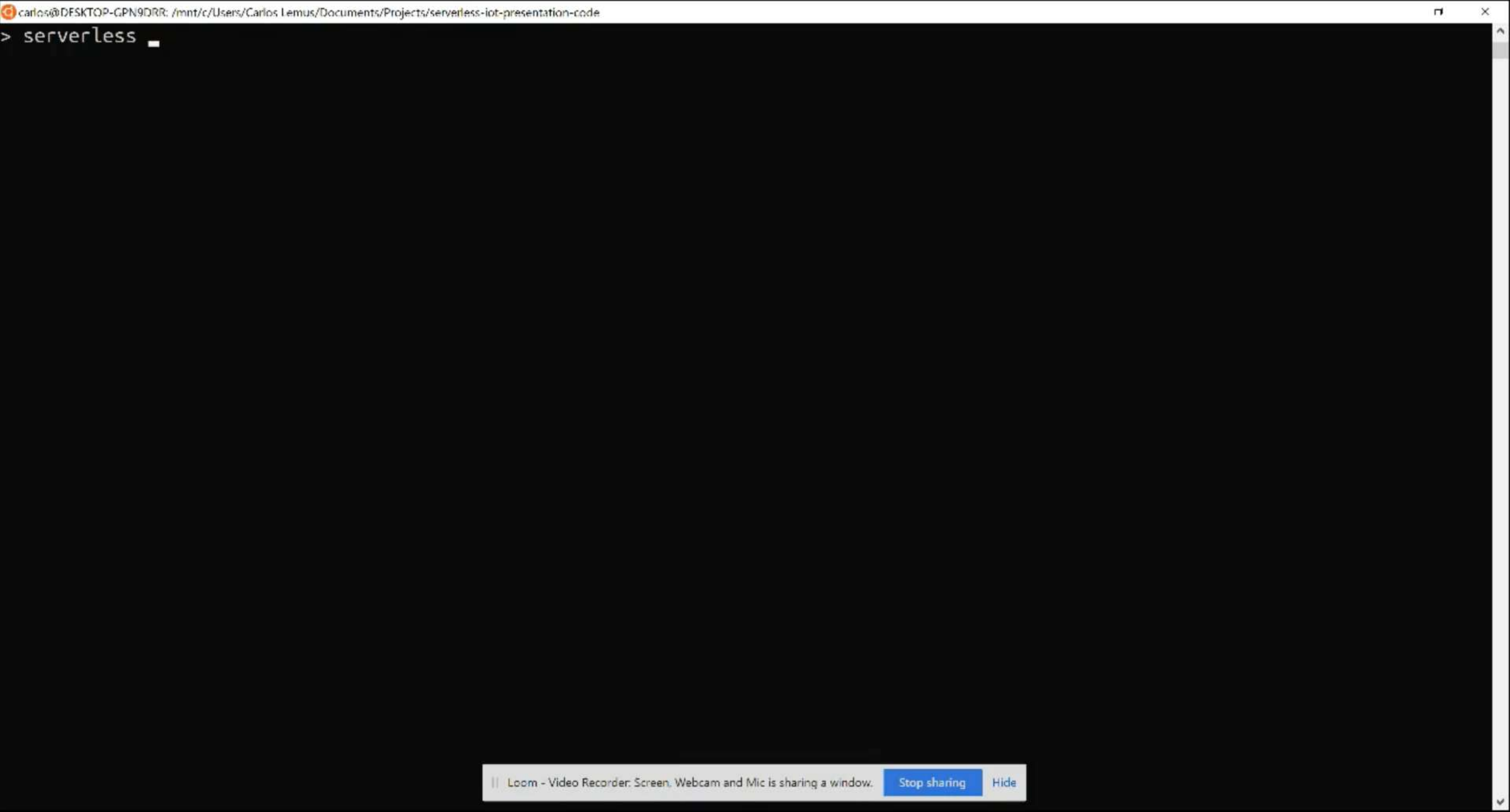
carlos@DESKTOP-GPN9DRR: /mnt/c/Users/Carlos Lemus/Documents/Projects/serverless-iot-presentation-code

```
>
```

Loom - Video Recorder: Screen, Webcam and Mic is sharing a window.

Invoking a Lambda Function

```
serverless invoke -f <function_name> -p <payload_file> -l
```



carlos@DESKTOP-GPN9DRR: /mnt/c/Users/Carlos Lemus/Documents/Projects/serverless-iot-presentation-code

```
> serverless _
```

Loom - Video Recorder: Screen, Webcam and Mic is sharing a window.

AWS IoT Device Simulator

Create and simulate 100s of virtual devices without physical devices or complicated scripts.

AWS IoT Device Simulator

- Simulates random JSON payloads
- Publishes payloads to your AWS IoT Endpoint
- Automotive module

In Summary

- How to build a simple backend through the console
- How to use a framework to automate and manage development
- How to use Device Simulator

More from us

trek10

Our blog and podcast offer more on Serverless, IoT, and other cutting edge AWS Service offerings:

http://www.trek10.com/blog

Want to talk infrastructure design, architecture reviews, managed services, and more?

http://www.trek10.com/contact

@trek10inc
linkedin.com/company/trek10

aws partner network

Advanced
Consulting Partner

IoT Competency

DevOps Competency

MSP Partner

Public Sector Partner