Serverless Computing

with AWS for Data Science

Christoph Bodner & Thomas Laber

AGENDA

	01	02	03
Topics	What is AWS Lambda	Build R Runtime	Build Custom Function
	AWS - Serverless concept - AWS Lambda	R runtime - What is a runtime? - Build a runtime	Lambda functions - Deploy a function - Invoke a function

AGENDA

	01	02	03
Topics	What is AWS Lambda	Build R Runtime	Build Custom Function
	AWS - Serverless concept - AWS Lambda	R runtime - What is a runtime? - Build a runtime	Lambda functions - Deploy a function - Invoke a function

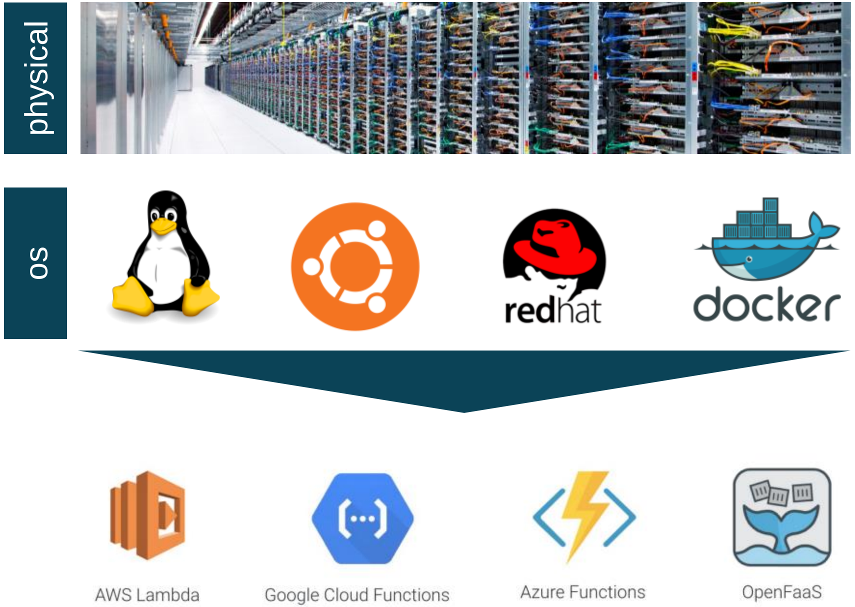
WHY SERVERLESS

SERVERLESS HAS SERVERS, BUT WE DON‘T HAVE TO CARE ☺

Just like wireless internet has wires somewhere, serverless architectures still have servers somewhere.

What ‘serverless’ really means is that, as a developer you don’t have to think about those servers. You just focus on code.

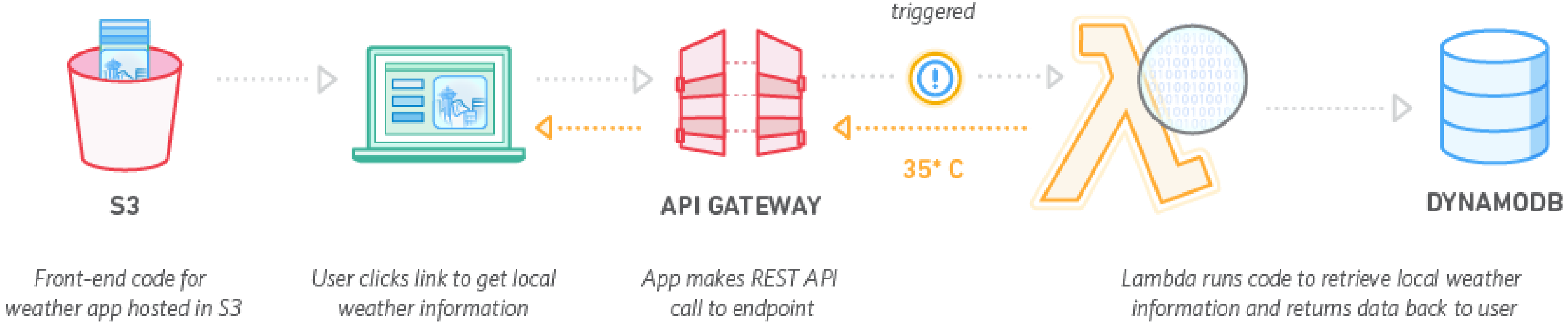
serverless



WHY SERVERLESS

FLOWCHART - EXAMPLE APP

Example: *Weather Application*



Front-end code for weather app hosted in S3

User clicks link to get local weather information

App makes REST API call to endpoint

Lambda runs code to retrieve local weather information and returns data back to user

WHY SERVERLESS

THE PROMISE: FOCUS ON CODING, NOT MAINTENANCE

No administration

No server provisioning and maintenance is necessary. Hardware and OS are abstracted away

Scale on demand

Scaling is automatic and part of the service.

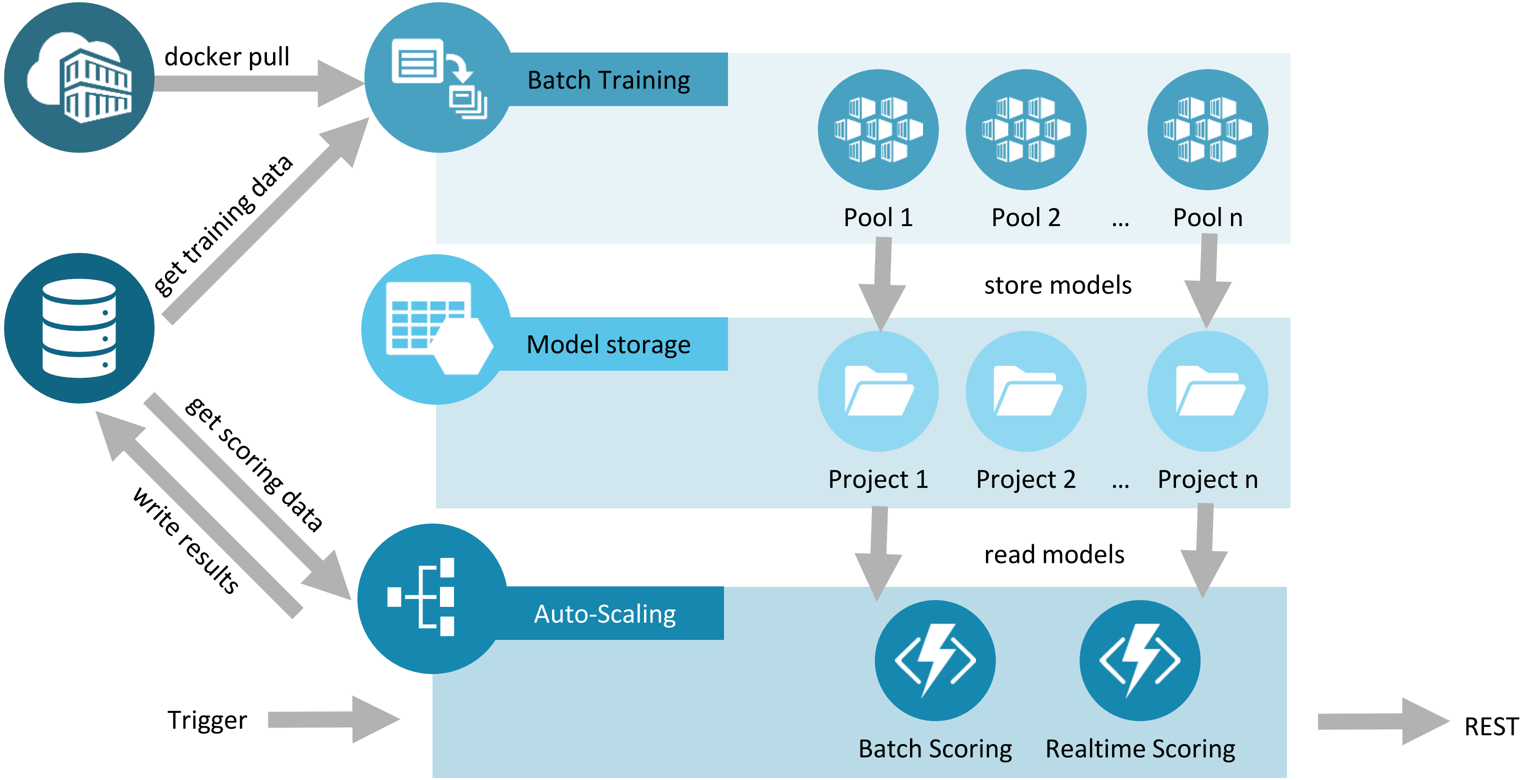
Pay-per-use

Billing is based on actual compute resources used. No compute used, no costs.

Faster turnaround

No server provisioning and maintenance is necessary. Hardware and OS are abstracted away

WHY SERVERLESS
SERVERLESS DATA SCIENCE ARCHITECTURE



CUSTOM RUNTIME
AMAZONS DEFINITION

Custom AWS Lambda Runtimes

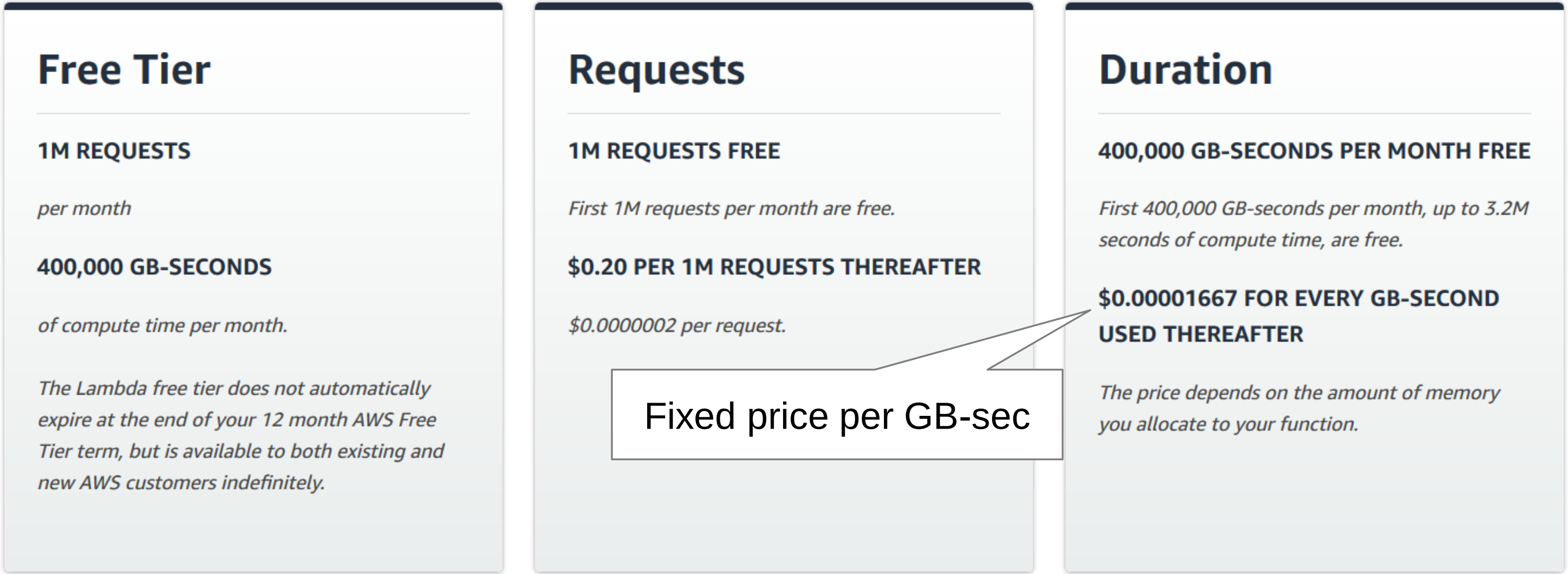
„You can implement an AWS Lambda runtime in **any programming language**. A runtime is a program that runs a Lambda **function's handler method** when the function is invoked. You can include a runtime in your function's deployment package in the form of an executable file named **bootstrap**.

A runtime is responsible for running the function's **setup code**, reading the **handler name** from an environment variable, and reading **invocation events** from the Lambda runtime API. The runtime passes the event data to the **function handler**, and **posts the response** from the handler back to Lambda.”

https://docs.aws.amazon.com/lambda/latest/dg/runtimes-custom.html

AWS LAMBDA PRICING MODEL

REQUESTS AND GB-SECONDS



Free Tier

1M REQUESTS

per month

400,000 GB-SECONDS

of compute time per month.

The Lambda free tier does not automatically expire at the end of your 12 month AWS Free Tier term, but is available to both existing and new AWS customers indefinitely.

Requests

1M REQUESTS FREE

First 1M requests per month are free.

$0.20 PER 1M REQUESTS THEREAFTER

$0.0000002 per request.

Duration

400,000 GB-SECONDS PER MONTH FREE

First 400,000 GB-seconds per month, up to 3.2M seconds of compute time, are free.

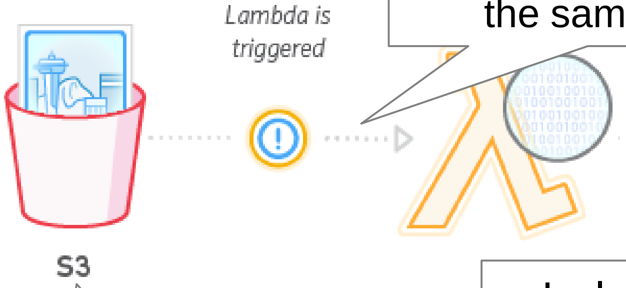
$0.00001667 FOR EVERY GB-SECOND USED THEREAFTER

The price depends on the amount of memory you allocate to your function.

Fixed price per GB-sec

Source: https://aws.amazon.com/lambda/pricing/ as of 2019-02-11

AWS LAMBDA PRICING MODEL
EXAMPLE CALCULATION



Transfer is free in the same region

Storage is extra (but pretty cheap)

Included in monthly free tier

Included in monthly free tier

AWS Lambda Billing Example

Use Lambda to score R models once a day using 512mb RAM. Scoring all models takes 10min

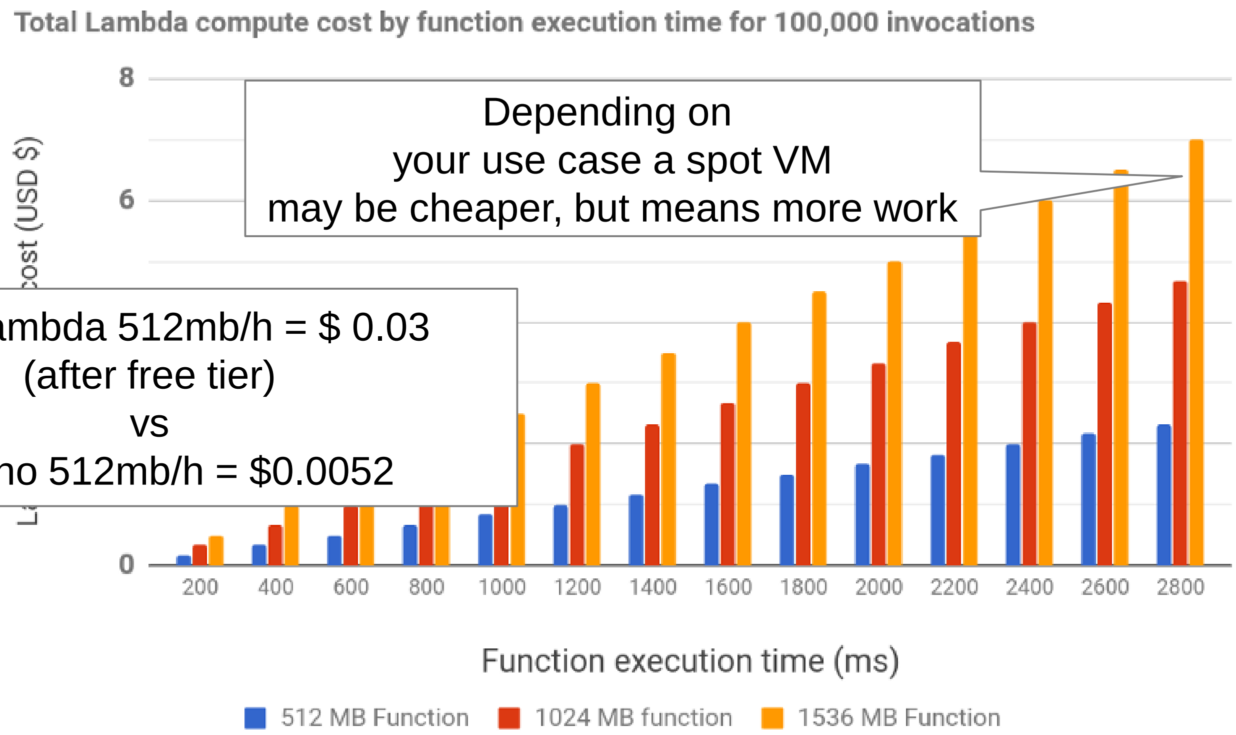
# Requests (i.e. Function triggers)

- 1 x 30 = 30

# GB/sec

$$\underbrace{\underbrace{512/1024}_{\text{0.5 GB}} \text{ x } \underbrace{\text{10min x 60sec x 30days}}_{\text{18,000 sec / month}}}_{\text{9,000 GB-sec}}$$

WHAT'S IT GONNA COST?
IN MANY USE CASES SERVERLESS IS GOOD VALUE FOR MONEY



Depending on
your use case a spot VM
may be cheaper, but means more work

Cost Lambda 512mb/h = $ 0.03
(after free tier)
vs
t3.nano 512mb/h = $0.0052

Source: https://serverless.com/blog/understanding-and-controlling-aws-lambda-costs/ 2019-02-11

AGENDA

	01	02	03
Topics	What is AWS Lambda	Build R Runtime	Build Custom Function
	AWS - Serverless concept - AWS Lambda	R runtime - What is a runtime? - Build a runtime	Lambda functions - Deploy a function - Invoke a function

CODE ADAPTED FROM
INFORMATION SOURCE

Medium

Running R on AWS Lambda
by Philipp Schirmer

Links
- https://medium.com/bakdata/running-r-on-aws-lambda-9d40643551a6
- https://github.com/bakdata/aws-lambda-r-runtime

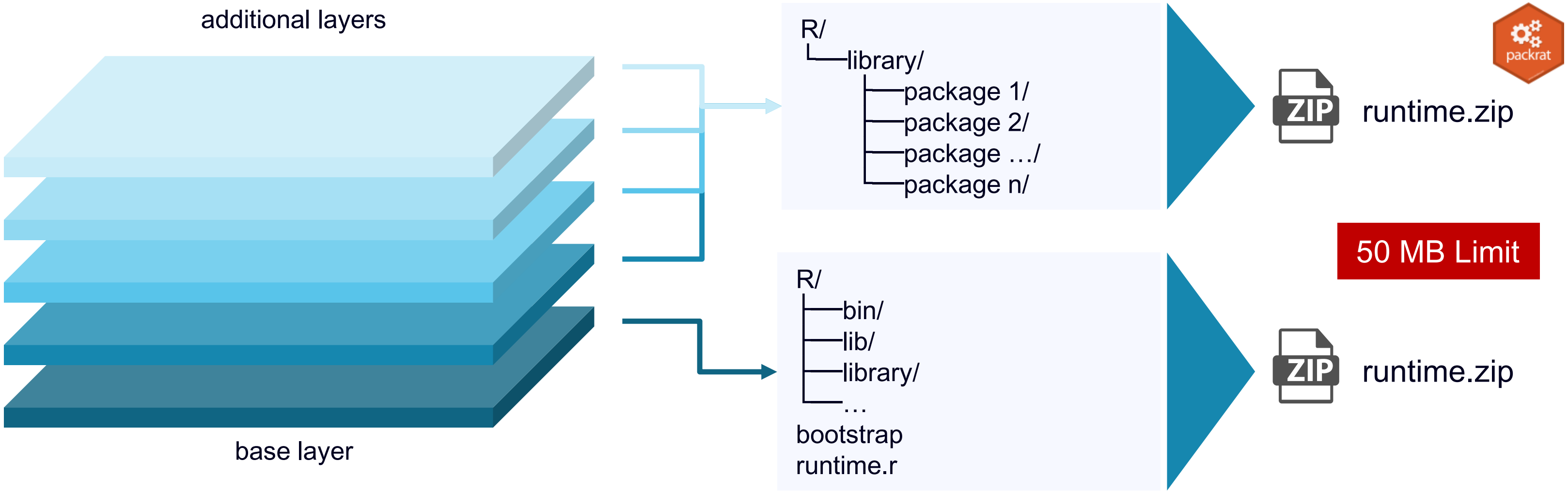
Additionally:
- https://medium.com/veltra-engineering/running-r-script-on-aws-lambda-custom-runtime-3a87403dcb

SERVERLESS R IN AWS

OVERVIEW SERVERLESS SERVICES FOR R

Layer

Layer content



additional layers

base layer

```
R/
└─library/
    ├─package 1/
    ├─package 2/
    ├─package …/
    └─package n/
```

runtime.zip

50 MB Limit

```
R/
├─bin/
├─lib/
├─library/
└─…
bootstrap
runtime.r
```

runtime.zip

A function can use up to 5 layers at a time. The total unzipped size of the function and all layers can't exceed the unzipped deployment package size limit of 250 MB.

BOOTSTRAP
EXECUTABLE FILE

Initialization Tasks

RETRIEVE SETTINGS

- _HANDLER – The location to the handler, from the function's configuration. The standard format is *file.method*, where file is the name of the file without an extension, and method is the name of a method or function that's defined in the file.
- LAMBDA_TASK_ROOT – The directory that contains the function code.
- AWS_LAMBDA_RUNTIME_API – The host and port of the runtime API.

INITIALIZE THE FUNCTION

Load the handler file and run any global or static code that it contains. Functions should create static resources like SDK clients and database connections once, and reuse them for multiple invocations.

HANDLE ERROS

If an error occurs, call the initialization error API and exit immediately.

AWS LAMBDA FUNCTIONS
OVERVIEW

bootstrap

```
# Processing
while true

do
    HEADERS="$(mktemp)"

    # Get an event
    EVENT_DATA=$(curl -sS -LD "$HEADERS" -X GET "http://${AWS_LAMBDA_RUNTIME_API}/2018-06-01/runtime/invocation/next")

    REQUEST_ID=$(grep -Fi Lambda-Runtime-Aws-Request-Id "$HEADERS" | tr -d '[:space:]' | cut -d: -f2)

    /opt/R/bin/Rscript /opt/runtime.r $EVENT_DATA $REQUEST_ID
done
```

AWS LAMBDA FUNCTIONS
OVERVIEW

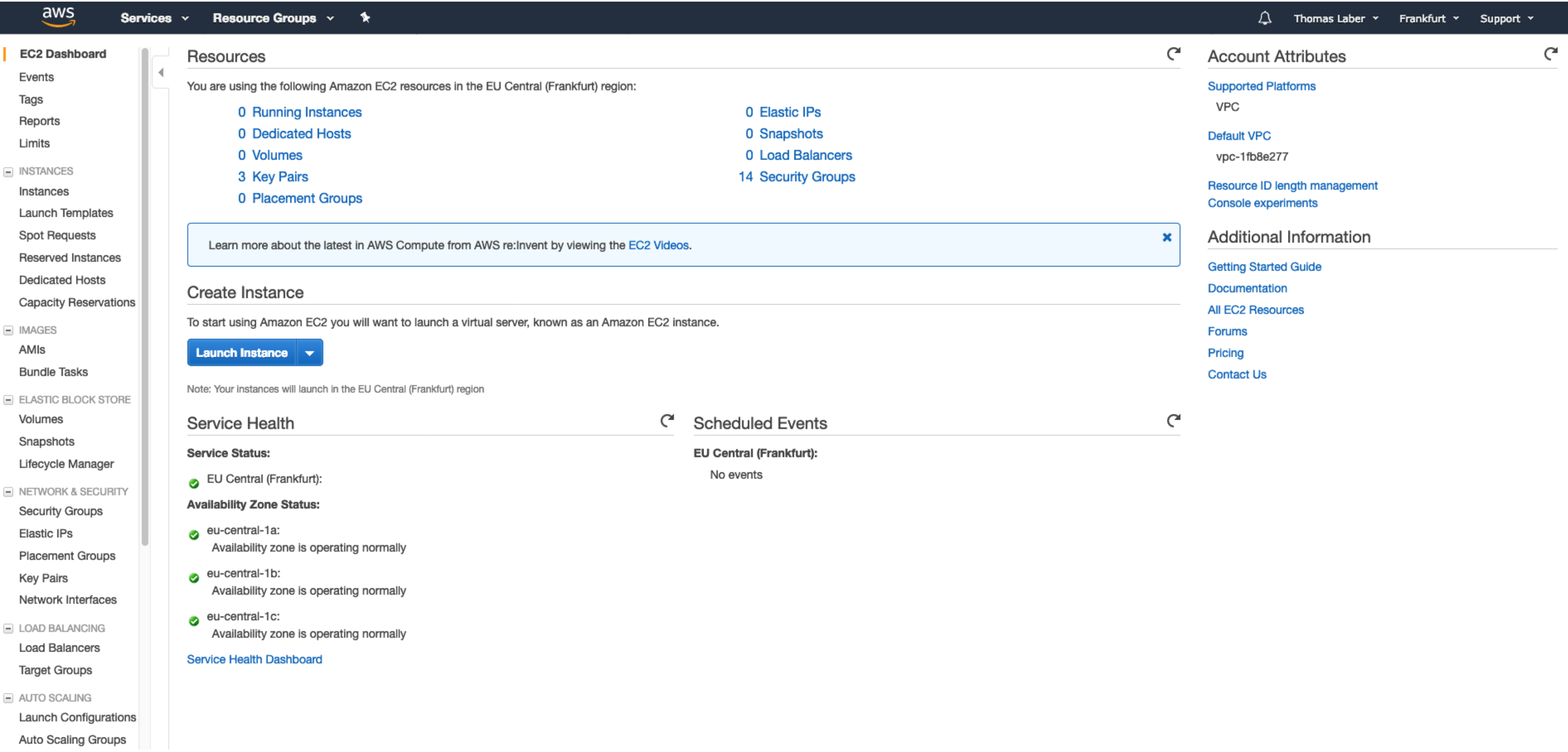
runtime.R

```r
library(httr); library(jsonlite);

HANDLER <- Sys.getenv("_HANDLER")
AWS_LAMBDA_RUNTIME_API <- Sys.getenv("AWS_LAMBDA_RUNTIME_API")
args = commandArgs(trailingOnly = TRUE)
EVENT_DATA <- args[1]
REQUEST_ID <- args[2]

HANDLER_split <- strsplit(HANDLER, ".", fixed = TRUE)[[1]]
file_name <- paste0(HANDLER_split[1], ".r")
function_name <- HANDLER_split[2]
print(paste0("Sourcing '", file_name, "'"))
source(file_name)
print(paste0("Invoking function '", function_name, "' with parameters:"))
params <- fromJSON(EVENT_DATA)
print(params)
result <- do.call(function_name, params)
print("Function returned:"); print(result);
url <- paste0("http://", AWS_LAMBDA_RUNTIME_API, "/2018-06-01/runtime/invocation/", REQUEST_ID, "/response")
res <- POST(url, body = list(result = result), encode = "json")
print("Posted result:"); print(res);
```

DEMO



Resources

You are using the following Amazon EC2 resources in the EU Central (Frankfurt) region:

0 Running Instances	0 Elastic IPs
0 Dedicated Hosts	0 Snapshots
0 Volumes	0 Load Balancers
3 Key Pairs	14 Security Groups
0 Placement Groups	

Learn more about the latest in AWS Compute from AWS re:Invent by viewing the EC2 Videos.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the EU Central (Frankfurt) region

Service Health

Service Status:

✓ EU Central (Frankfurt):

Availability Zone Status:

✓ eu-central-1a:
Availability zone is operating normally

✓ eu-central-1b:
Availability zone is operating normally

✓ eu-central-1c:
Availability zone is operating normally

Service Health Dashboard

Scheduled Events

EU Central (Frankfurt):

No events

Account Attributes

Supported Platforms
VPC

Default VPC
vpc-1fb8e277

Resource ID length management
Console experiments

Additional Information

Getting Started Guide
Documentation
All EC2 Resources
Forums
Pricing
Contact Us

AGENDA

	01	02	03
Topics	What is AWS Lambda	Build R Runtime	Build Custom Function
	AWS - Serverless concept - AWS Lambda	R runtime - What is a runtime? - Build a runtime	Lambda functions - Deploy a function - Invoke a function

AWS LAMBDA WORKFLOW

OVERVIEW SERVERLESS SERVICES FOR R

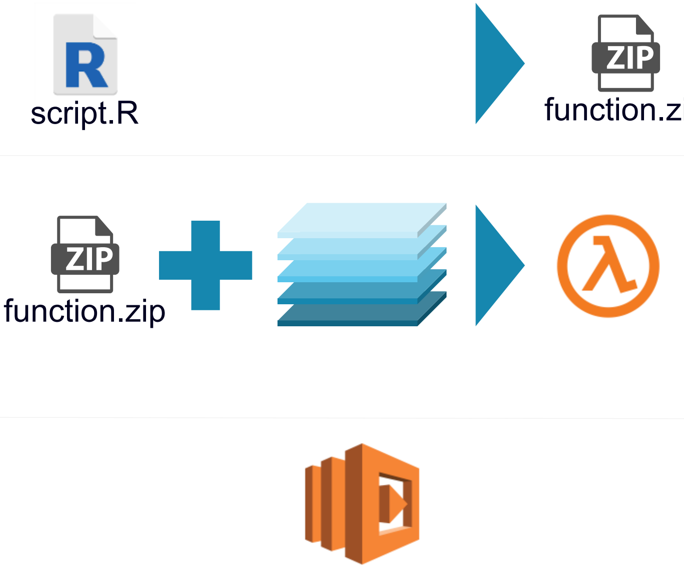
Code

```r
increment <- function(x) {
  return(x + 1)
}
```

```
aws lambda create-function --function-name r-example \
  --zip-file fileb://function.zip --handler script.increment \
  --runtime provided --timeout 60 \
  --layers <layer-arn> \
  --role <role-arn> --region eu-central-1
```

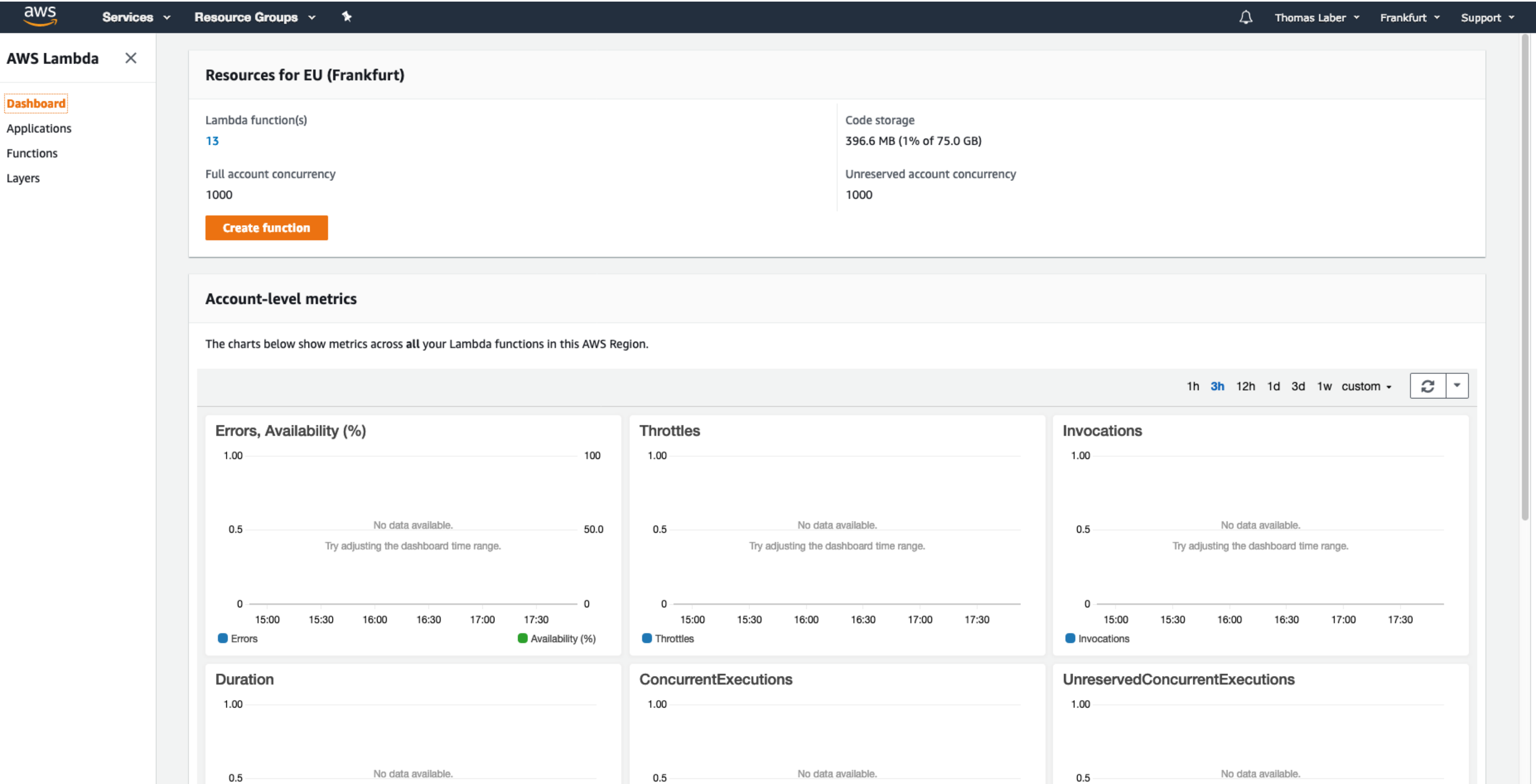
```
aws lambda invoke --function-name r-example \
  --payload '{"x":1}'--region eu-central-1 response.txt
```

Layer content



Function memory allocation is limited to 3 GB and maximum duration is 15 minutes.

DEMO



Resources for EU (Frankfurt)

Lambda function(s)
13

Code storage
396.6 MB (1% of 75.0 GB)

Full account concurrency
1000

Unreserved account concurrency
1000

Account-level metrics

The charts below show metrics across **all** your Lambda functions in this AWS Region.

PACKRAT
DEPENDENCY MANAGEMENT SYSTEM FOR R

Private Package Library

ISOLATED

Installing a new or updated package for one project won’t break other projects.

That’s because packrat gives each project its own private package library.

PORTABLE

Easily transport your projects from one computer to another, even across different platforms. Packrat makes it easy to install the packages your project depends on.

REPRODUCABLE

Packrat records the exact package versions you depend on, and ensures those exact versions are the ones that get installed wherever you go.

THANK YOU FOR YOUR ATTENTION!

Questions?