



# SERVERLESS BLOG WORKSHOP

Mikael Puittinen, CTO  
mikael.puittinen@sc5.io  
@mpuittinen

10.10.2016

# SC5 BRIEFLY



CLOUD  
SOLUTIONS



BUSINESS  
APPLICATIONS



DIGITAL  
DESIGN

**10**  
YEARS

**100+**  
CUSTOMERS

**400+**  
PROJECTS

**85**  
HACKERS  
DESIGNERS

**HEL**  
**JKL**

**~7**  
MEUR  
2016



A-lehdet



VISIT OUR WEB SITE FOR MORE INFO: [HTTPS://SC5.IO](https://sc5.io)

# INTRODUCTION

- Introduction to AWS & Serverless Framework

# THE SERVERLESS BLOG WORKSHOP

What we'll do

# WORKSHOP THEME: BLOG BACKEND

Create a backend for the blog application running at <http://hackathon-blog.serverless.fi/>  
(sources at <https://github.com/SC5/aws-serverless-hackathon>)

Backend must have a REST API with methods

1. POST /dev/posts
2. GET /dev/posts
3. PUT /dev/posts/{postId} - OPTIONAL
4. DELETE /dev/posts/{postId} - OPTIONAL

Use e.g. AWS DynamoDB as the database for blog posts.

**Step-by-step walkthrough available at <http://hackathon.serverless.fi/workshop.pdf>**

# RESOURCES

Blog client:

<https://github.com/SC5/aws-serverless-hackathon>

Reference implementation:

<https://github.com/SC5/serverless-blog-workshop>

This presentation:

<http://serverless.fi/docs/blog-workshop.pdf>

Code snippets:

<http://serverless.fi/docs/blog-workshop>

# GETTING READY FOR THE WORKSHOP / HACKATHON

Instructions for getting set up:

<http://serverless.fi/docs/workshop-preps>



# STEP-BY-STEP INSTRUCTIONS

# 1. CREATE SERVERLESS PROJECT

```
> sls install -u https://github.com/SC5/sc5-serverless-boilerplate -n  
serverless-blog  
  
> cd serverless-blog  
  
> npm install
```

This creates a new project serverless-blog based on sc5-serverless-boilerplate and installs the node modules required by the project.

## 2. CREATE DYNAMODB TABLE FOR POSTS (USING SERVERLESS)

- ❑ Serverless uses AWS Cloudformation to deploy resources (defined in *serverless.yml*)
- ❑ Uncomment resources and Resources in *serverless.yml* and add snippet *blog-table.yml* (check indenting)

Permissions to the table are granted by default in the boilerplate template.

```
# DynamoDB Blog table for workshop
BlogTable:
  Type: AWS::DynamoDB::Table
  DeletionPolicy: Retain
  Properties:
    AttributeDefinitions:
      - AttributeName: id
        AttributeType: S
    KeySchema:
      - AttributeName: id
        KeyType: HASH
    ProvisionedThroughput:
      ReadCapacityUnits: 1
      WriteCapacityUnits: 1
    TableName: ${self:provider.environment.TABLE_NAME}
```

### 3. CREATE FUNCTION AND SET ENDPOINTS

- ❑ Create function posts for your service

```
> sls create function -f posts --  
handler posts/index.handler
```

- ❑ Add HTTP endpoints by adding the snippet *http-events.yml* to the posts function in *serverless.yml*

```
events:  
  - http:  
    path: posts  
    method: get  
    cors: true  
    integration: lambda  
  - http:  
    path: posts  
    method: post  
    cors: true  
    integration: lambda  
  - http:  
    path: posts/{id}  
    method: put  
    cors: true  
    integration: lambda  
  - http:  
    path: posts/{id}  
    method: delete  
    cors: true  
    integration: lambda
```

## 4. IMPLEMENT THE LOGIC

- Implement the logic for the function into `posts/`. The entry point for the Lambda function is `index.js` in that folder.
- Copy the snippets `index.js` and `BlogStorage.js` from github (unless you want to code them yourself)
- If you opt to code yourself, we recommend to use.  
`AWS.DynamoDB.DocumentClient` to access the database table.  
The table name is  
`${self:provider.environment.TABLE_NAME}`

## 5. TEST THE FUNCTION

- ❑ Copy the snippet `posts.js` to `test/`
- ❑ Deploy (requires the resources) and run tests using `serverless-mocha-plugin`

```
> sls deploy  
> sls invoke test --region us-east-1 --stage dev
```

## 6. SET UP ENDPOINTS IN THE SAMPLE APP

- ❑ Launch the blog application at

`http://hackathon-blog.serverless.fi`

- ❑ Enter the endpoint URL (`https://.../dev/posts`) to the form and save (use `sls info` to get the endpoint)
- ❑ Try writing, editing, deleting posts

## 9. YOU DID IT! CONGRATS!

Next:

1. If you want to work more on serverless, check opportunities at <https://sc5.io/careers>
2. If you are interested in serverless, join the "Serverless Finland" meetup at <http://www.meetup.com/Helsinki-Serverless/> and follow <http://serverless.fi>





THANK YOU!

*mikael.puittinen@sc5.io*

*@mpuittinen*