

wenn cloud, dann richtig

...und mit Java

**"It's not work if you like it"
...so I never worked. #java**

adambien.blog

airhacks.io => online video courses

airhacks.news => stay tuned

airhacks.fm => podcast

NEW discord.gg/airhacks

NEW youtube.com/@bienadam/shorts



#29 Jakarta EE / MicroProfile in the Clouds: Runtimes not Servers

[episode link] Listen on

#185 A Cloud Migration Story: From J2EE to Serverless Java

An airhacks.fm conversation with

servers science, modern

An airhacks.fm

ZX Spectru

CPC 64, De

in 1993, usi

Lambda, Cl

clouds the

services, n

quarkus in

the cloud h



[RSS]

#219 Java, CraC and Reducing Cold Start Duration with AWS Lambda SnapStart

[episode link]



[RSS]

An airhacks.fm conversation with Mark Sailes (@MarkSailes3) about:

CRaC API, C1 and C2 compilers, GraalVM and Random, CRaC and Stateful EJB beans, Lambda SnapStart and snapshotting the Firecracker VM, the CraC resource interface and listener methods, priming the critical path, Quarkus with MicroProfile AWS on Lambda CDK template, Plain Java AWS Lambda with CDK template, SDKs calls in the beforeCheckpoint hook, SnapStart state never leaves the region, SnapStart state is cached in caches within Availability Zones, SnapStart is available within VPCs, only versioned AWS Lambdas can be optimized, Provisioned Concurrency and SnapStart, The Other Feature of AWS Lambda Provisioned Concurrency – Saving Money, A serverless journey: AWS Lambda under the hood provisioned concurrency and EC 2 reserved instances, AWS Lambda function starts at bare metal,

airhacks.TV

with the time machine, “100 episodes ago segment”

...any questions left?

airhacks.live

NEW online, live virtual workshops

Continuous coding, explaining, interacting and sharing with [Adam Bien](#)

Live, Virtual Online Workshops, Summer 2023:

[AWS Java Bootstrap, June 15th, 2023](#)

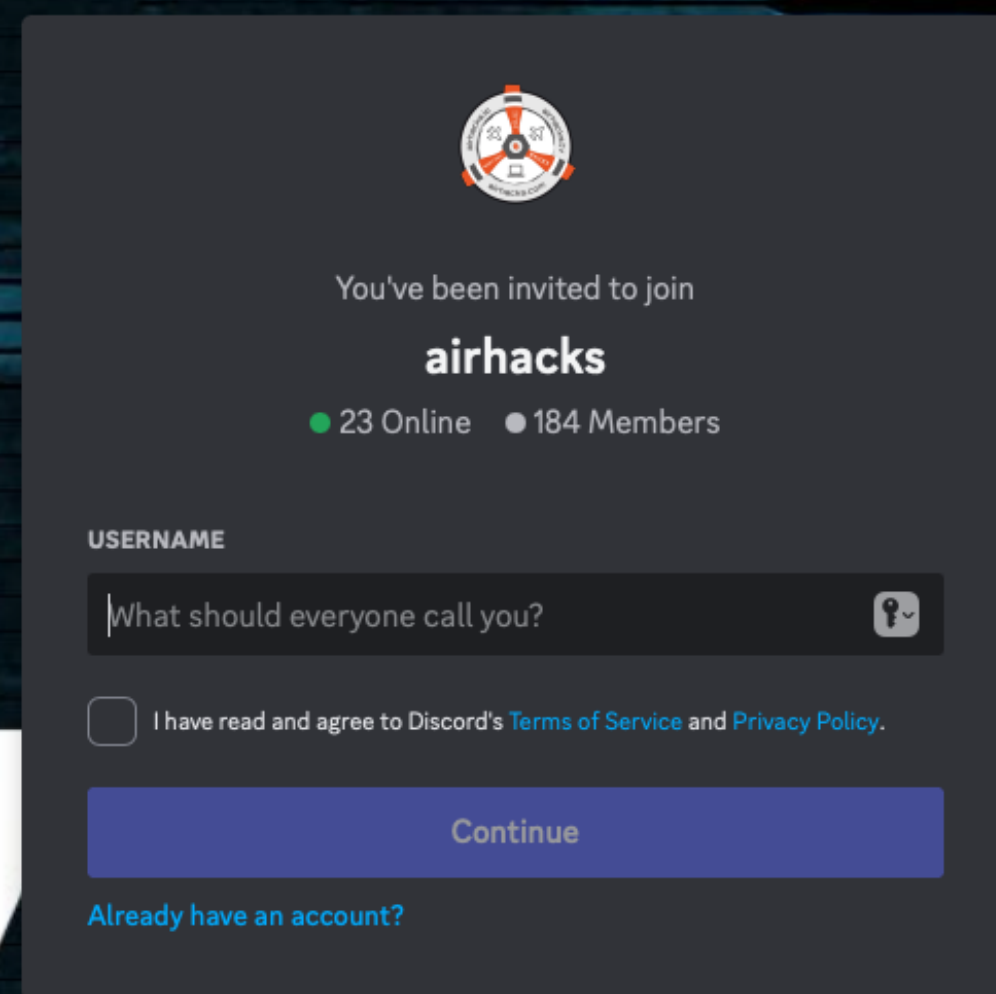
[Cost Driven Architectures with Java on AWS, June 22nd, 2023](#)


Tickets are also available from: airhacks.eventbrite.com and meetup.com/airhacks

by [Adam Bien](#)

You don't like live, interactive virtual workshops? Checkout video courses: airhacks.io


WELCOME TO airhacks





You've been invited to join
airhacks
● 23 Online ● 184 Members

USERNAME

What should everyone call you? 

I have read and agree to Discord's [Terms of Service](#) and [Privacy Policy](#).

[Continue](#)

[Already have an account?](#)

from J2EE to serverless clouds



Sun Java Web Server

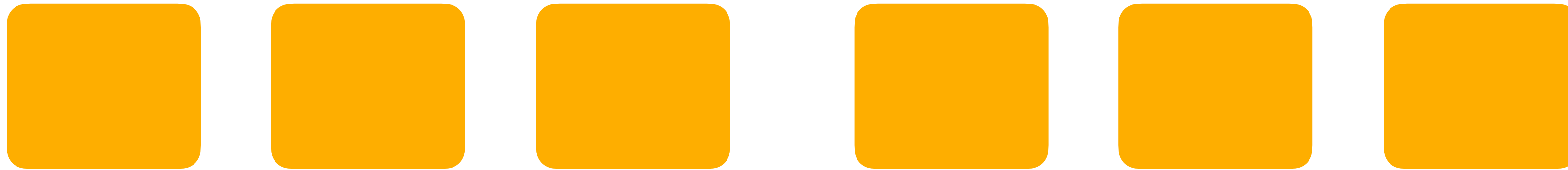


web container



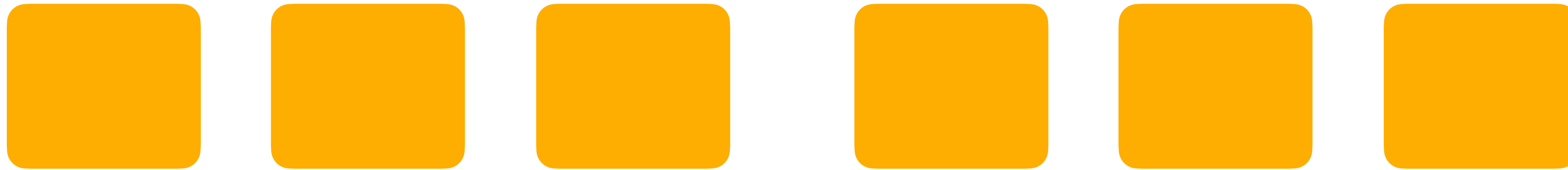
EJB container

J2EE application server



web + EJB container

Java EE application server



CDI + JAX-RS container

Jakarta EE application server



(Docker) container runtime



MicroProfile runtime

(AWS Lambda) serverless runtime



MicroProfile runtime

(Docker) container runtime



MicroProfile runtime

(AWS Lambda) serverless runtime

business logic

MicroProfile

AWS Lambda

business logic

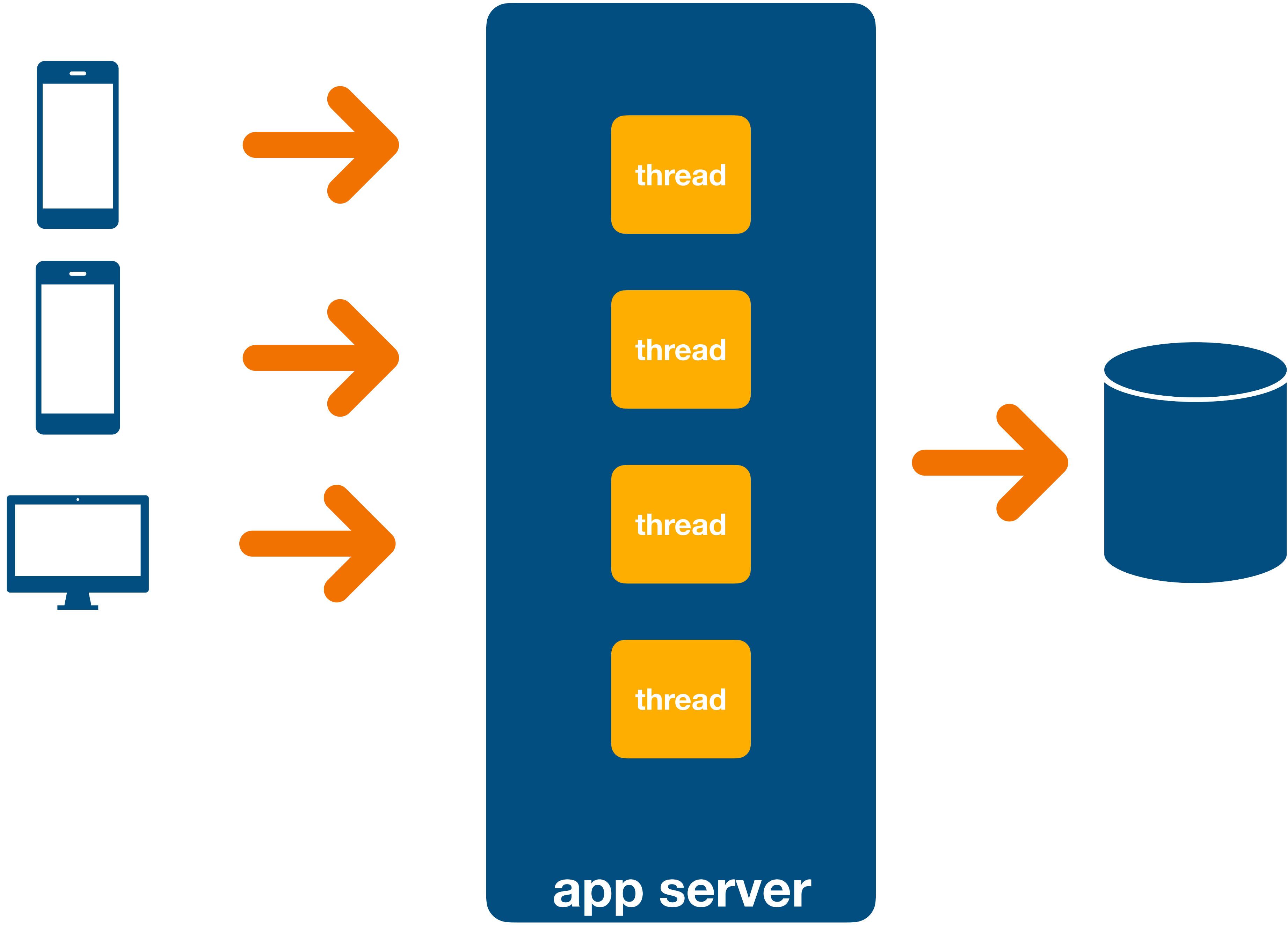
MicroProfile

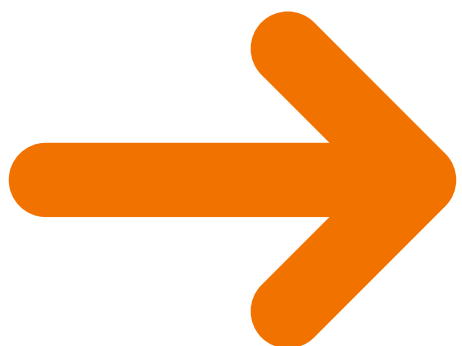
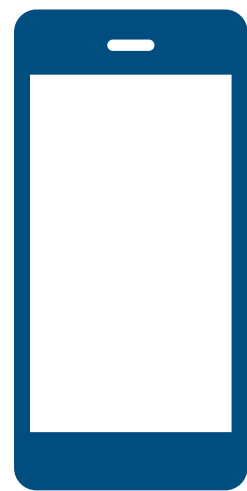
Java SE / bare metal

business logic

MicroProfile

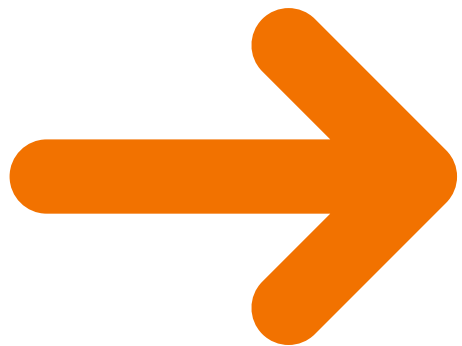
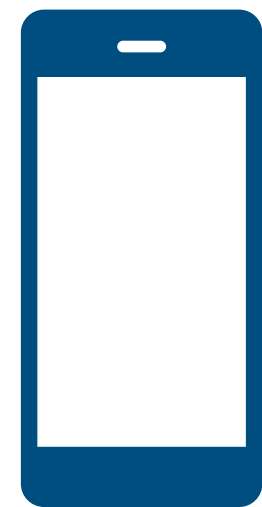
containerized





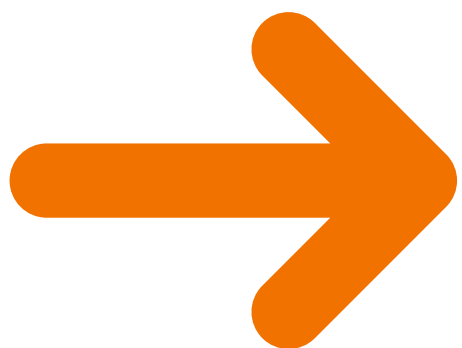
function.zip

**AWS Lambda
Process**



function.zip

**AWS Lambda
Process**

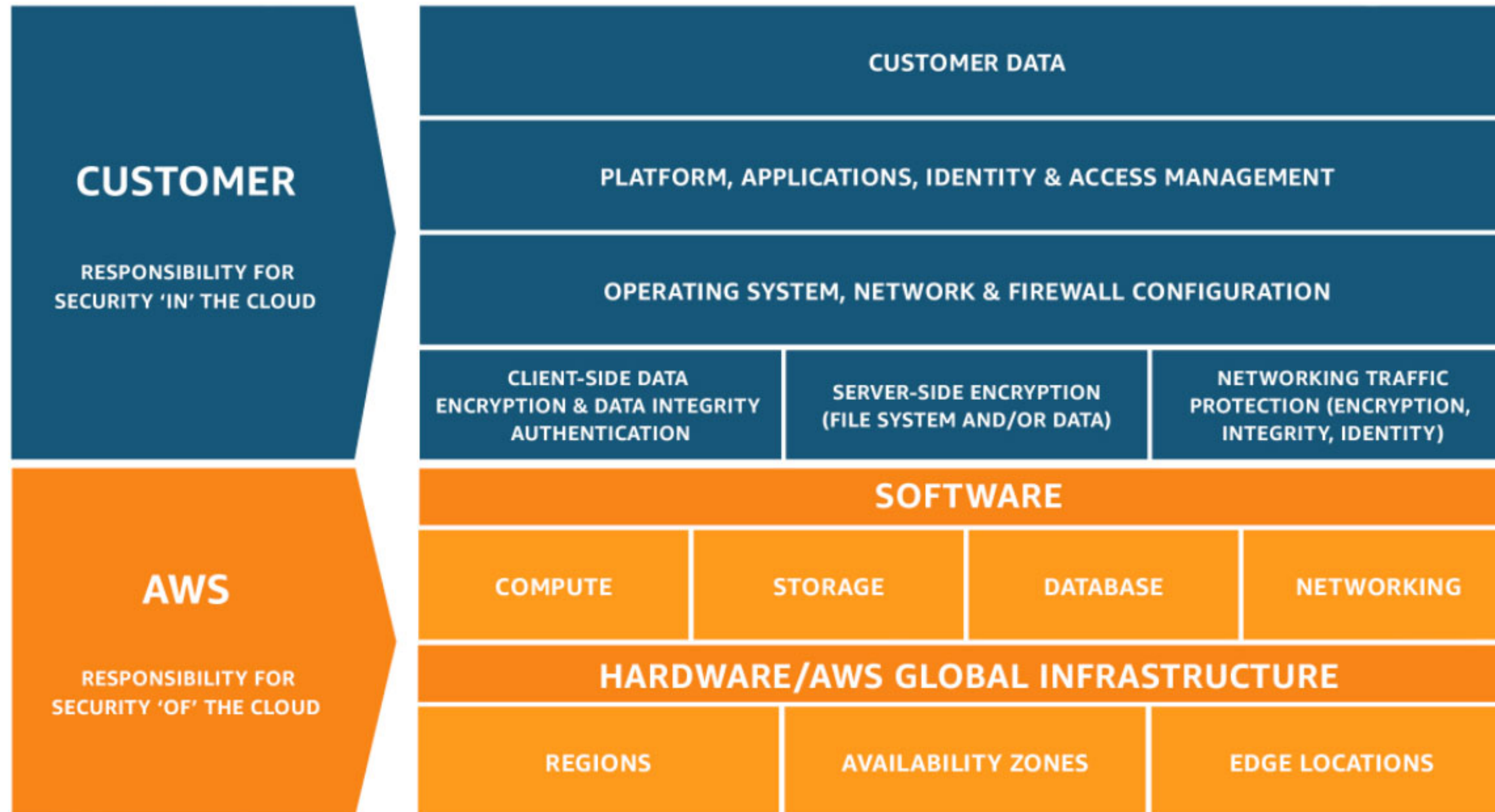


function.zip

**AWS Lambda
Process**

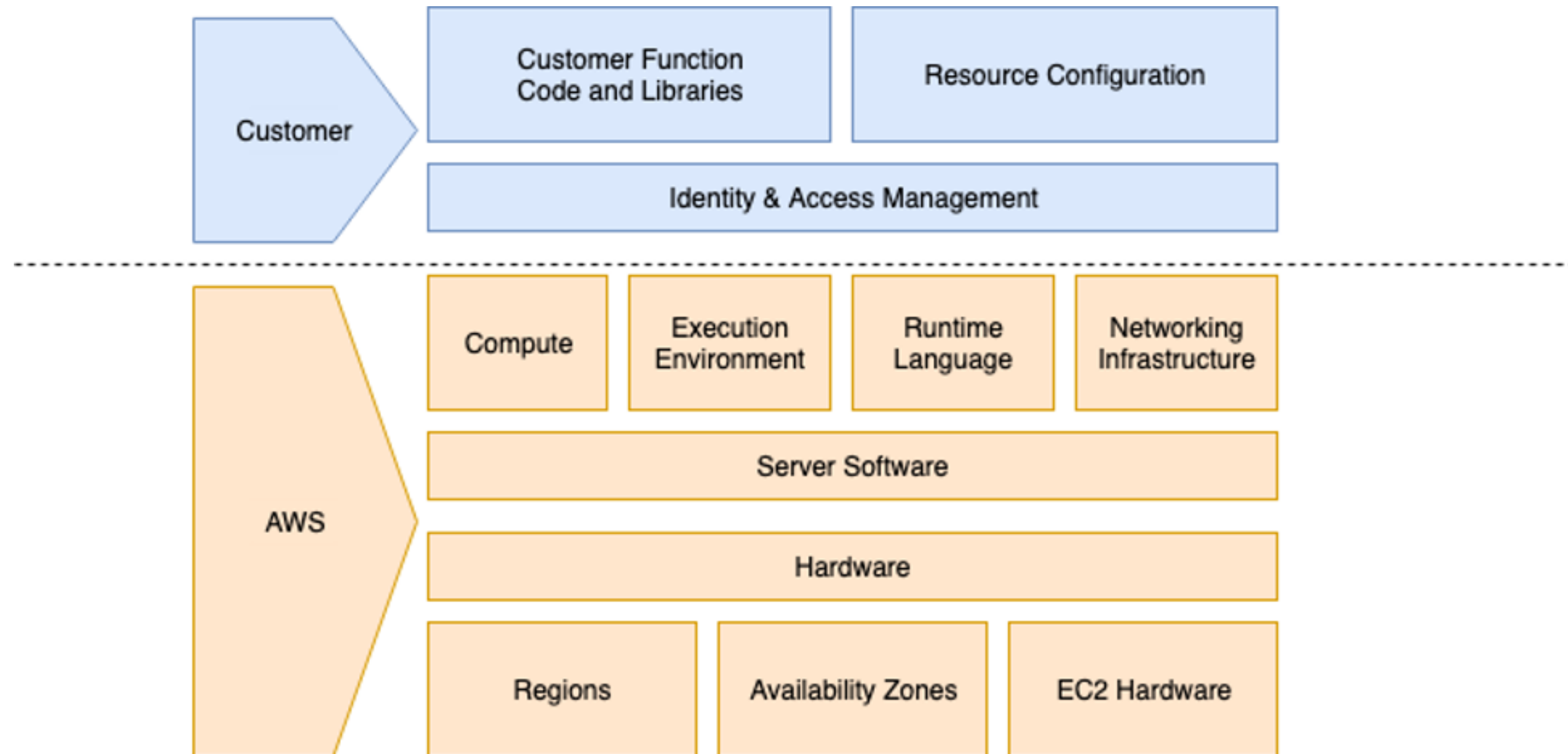


boring...



<https://aws.amazon.com/compliance/shared-responsibility-model/>

...next step



<https://docs.aws.amazon.com/whitepapers/latest/security-overview-aws-lambda/the-shared-responsibility-model.html>

J2EE / Java EE development roles

Java EE Product Provider

“The Java EE product provider is the company that designs and makes available for purchase the Java EE platform APIs and other features defined in the Java EE specification. Product providers are typically application server vendors that implement the Java EE platform according to the Java EE 6 Platform specification.”

<https://docs.oracle.com/javaee/6/tutorial/doc/bnaca.html>

Tool Provider

“The tool provider is the company or person who creates development, assembly, and packaging tools used by component providers, assemblers, and deployers.”

<https://docs.oracle.com/javaee/6/tutorial/doc/bnaca.html>

Application Component Provider

“The application component provider is the company or person who creates web components, enterprise beans, applets, or application clients for use in Java EE applications.”

<https://docs.oracle.com/javaee/6/tutorial/doc/bnaca.html>

Enterprise Bean Developer

“An enterprise bean developer performs the following tasks to deliver an EJB JAR file that contains one or more enterprise beans:

- Writes and compiles the source code
- Specifies the deployment descriptor (optional)
- Packages the `.class` files and deployment descriptor into the EJB JAR file”

<https://docs.oracle.com/javaee/6/tutorial/doc/bnaca.html>

Web Component Developer (Application Client Developer)

“A web component developer performs the following tasks to deliver a WAR file containing one or more web components:

- Writes and compiles servlet source code
- Writes JavaServer Faces, JSP, and HTML files
- Specifies the deployment descriptor (optional)
- Packages the `.class`, `.jsp`, and `.html` files and deployment descriptor into the WAR file

<https://docs.oracle.com/javaee/6/tutorial/doc/bnaca.html>

Application Assembler

“The application assembler is the company or person who receives application modules from component providers and may assemble them into a Java EE application EAR file. The assembler or deployer can edit the deployment descriptor directly or can use tools that correctly add XML tags according to interactive selections.

A software developer performs the following tasks to deliver an EAR file containing the Java EE application:

- Assembles EJB JAR and WAR files created in the previous phases into a Java EE application (EAR) file
- Specifies the deployment descriptor for the Java EE application (optional)
- Verifies that the contents of the EAR file are well formed and comply with the Java EE specification

Application Deployer and Administrator

The application deployer and administrator is the company or person who configures and deploys application clients, web applications, Enterprise JavaBeans components, and Java EE applications, administers the computing and networking infrastructure where Java EE components and applications run, and oversees the runtime environment. Duties include setting transaction controls and security attributes and specifying connections to databases.

During configuration, the deployer follows instructions supplied by the application component provider to resolve external dependencies, specify security settings, and assign transaction attributes. During installation, the deployer moves the application components to the server and generates the container-specific classes and interfaces.

A deployer or system administrator performs the following tasks to install and configure a Java EE application or components:

- Configures the Java EE application or components for the operational environment
- Verifies that the contents of the EAR, JAR, and/or WAR files are well formed and comply with the Java EE specification
- Deploys (installs) the Java EE application or components into the Java EE server

references

- “Ten Years After: From Java EE 6 to Quarkus and AWS Lambda”: <http://youtube.com/@bienadam/>
- “Hey Enterprise EJB Developers Now Is The Time To Go Serverless”: <http://youtube.com/@bienadam/>
- “Past, Present and Future: Serverside Java on Premise and in the Clouds”: <http://youtube.com/@bienadam/>
- “Saving Costs with Pragmatic Serverless Java Architectures”: <http://youtube.com/@bienadam/>
- <https://github.com/AdamBien/x-ray> (~15 years old code)
- <https://github.com/AdamBien/lightfish> (~15 years old code)

on-premise

- direct access to machines possible
- compute and storage are either nearly free, or extremely expensive
- automation is hard
- there are no APIs
- “root” or elevated access is common
- “always on”
- intra-department dependencies

on-premise (contd.)

resources like:

- RDBMS
- MoM
- Idp / auth
- NAS
- backup

are already configured and can be conveniently used

clouds

- everything is an API
- everything can be automated
- storage can be cheap
- CPU is expensive
- you are responsible for everything
- opensource products can be hard to use and run

**if you go to the cloud
...use the cloud!**

airhacks.live

NEW online, live virtual workshops

Continuous coding, explaining, interacting and sharing with [Adam Bien](#)

Live, Virtual Online Workshops, Summer 2023:

[AWS Java Bootstrap, June 15th, 2023](#)

[Cost Driven Architectures with Java on AWS, June 22nd, 2023](#)

Tickets are also available from: airhacks.eventbrite.com and meetup.com/airhacks

by [Adam Bien](#)

You don't like live, interactive virtual workshops? Checkout video courses: airhacks.io

Thank YOU!