java and clouds

AWS

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

airhacks.io => online video courses

airhacks.news => stay tuned

airhacks.fm => podcast

adambien.blog

airhacks.live

airhacks.TV

airhacks.live

NEW online, live virtual workshops

"like airhacks.com, without leaving your home"

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

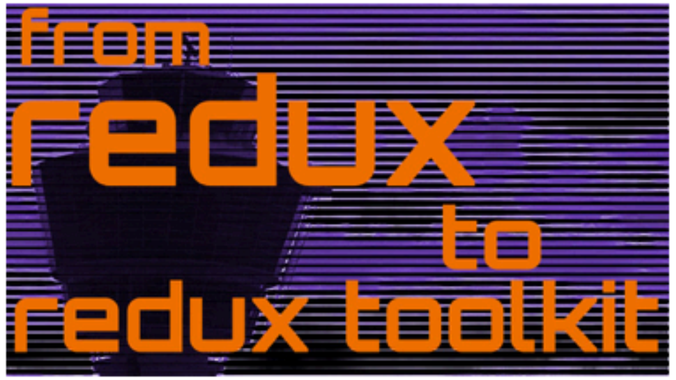
Live, Virtual Online Workshops, Winter 2021, [backends]:

Testing, Monitoring, Resiliency December 9th, 2021

Serverless Java on AWS Cloud, December 16th, 2021

by and with adam-bien.com

From redux to redux toolkit



From redux to redux toolkit

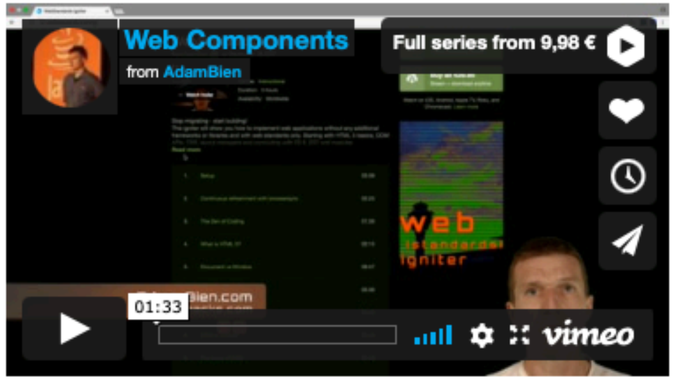
Web Components, lit-html and redux

webcomponents-with-redux.training

Effective Web Apps with Web Standards (only)

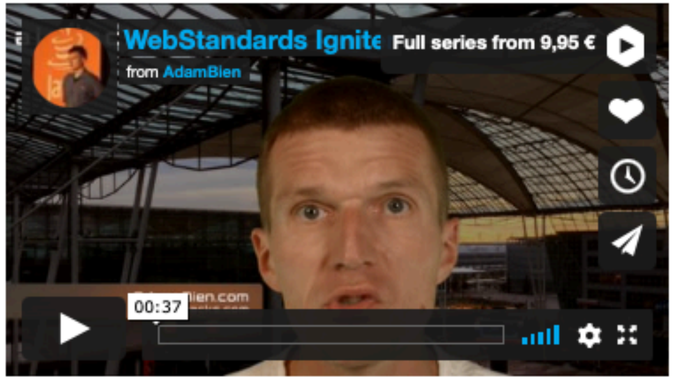
effectiveweb.training

Web Components Igniter



webcomponents.training

WebStandards Igniter



webstandards.training

>>java workshops

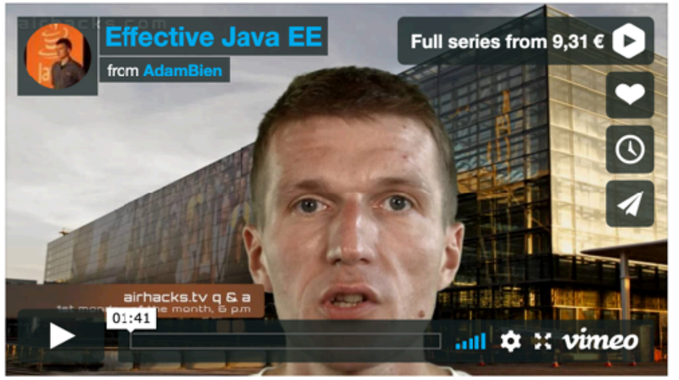
Apps with MicroProfile

microprofile.training

Java EE 7 bootstrap

javaeebootstrap.com

Effective Java EE 7



effectivejavaee.com

Java EE 7 Testing

javaeetesting.com

Java EE 7 Microservices

javaeemicro.services

airhacks.live

NEW

coupon code: redux4free

airhacks.live

Cloud computing is the on-demand availability of computer system resources,especially data storage (cloud storage) and computing power, without direct active management by the user. The term is generally used to describe data centers available to many users over the Internet. Large clouds, predominant today, often have functions distributed over multiple locations from central servers. If the connection to the user is relatively close, it may be designated an edge server.

https://en.wikipedia.org/wiki/Cloud_computing

(clients) cloud expectations

- no plumbing
- “single click deployments”
- automatic scalability
- agility
- low costs
- ease of use
- microservices are the preferred deployment model

lift and shift / rehost

“Most migrations happen in phases to minimize risk and speed up time to production. The most common approach is to lift-and-shift (also known as "rehost") an application and its data with as few changes as possible. This enables the fastest time to production.”

https://aws.amazon.com/products/storage/lift-and-shift/

are clouds killing microservices?

- microservice is a process
- every process comes with an overhead
- you are paying for the overhead
- projects are starting to merging their microservices into larger monoliths

monthly costs driven development

- minimise the amount of API calls
- use the least amount of vCPU possible
- use the least amount of RAM
- minimise the amount of communication
- user cheaper storage
- software quality, simplicity, lean code become less and less important

effective cloud strategies

- estimate the costs in a PoC
- automate everything, ignore the console (harder to forget things)
- no one cares about boring CRUD in the clouds
- non-functional requirements (beyond scalability :-))
- additional cloud services can provide significant added value
- the future is serverless
- FaaS (functions) are the ultimate integrator and trigger

on-premise / colocation hosting

- should be cheap
- years in production
- perfectly suitable for boring Jakarta EE / MicroProfile workloads
- additional features like: image recognition, distributed storage, certs rotation, encryption at rest, fine grained security model, high availability disaster recovery
- containers are a must
- kubernetes can help to run a large amount of applications

killer use cases for the clouds

significant added-value with pay-as-you-go, e.g.:

image recognition, text extraction, fraud detection, global caching, static hosting, certificate management, DNS, burstable scalability, your favourite OpenSource project as managed solution, IoT integration, CI/CD, backup

Java EE / Jakarta EE-cloud mapping

- Servlets
- CDI
- Managed Beans / EJBs
- JPA
- JMS
- Bean Validation
- JAX-RS
- Transactions
- WebSockets
- Jakarta Security, Authentication, Authorization

MicroProfile 4.0

AWS history

During 6 years at Amazon he witnessed the transformation of the company from a bookseller to the almost $1B, Infrastructure as a Service (IaaS) API, cloud computing leader. As Yegge's recalls that one day Jeff Bezos issued a mandate, sometime back around 2002 (give or take a year):

- All teams will henceforth expose their data and functionality through service interfaces.
- Teams must communicate with each other through these interfaces.
- There will be no other form of inter-process communication allowed: no direct linking, no direct reads of another team's data store, no shared-memory model, no back-doors whatsoever. The only communication allowed is via service interface calls over the network.
- It doesn't matter what technology they use.
- All service interfaces, without exception, must be designed from the ground up to be externalizable. That is to say, the team must plan and design to be able to expose the interface to developers in the outside world. No exceptions.

The mandate closed with:

> ***Anyone who doesn't do this will be fired. Thank you; have a nice day!***

Everyone got to work and over the next couple of years, Amazon transformed itself, internally into a service-oriented architecture (SOA), learning a tremendous amount along the way.

Think about what Bezos was asking! Every team within Amazon had to interact using web services. If you were human resources and you needed some numbers from marketing, you had to get them using an API. He was asking every team to decouple, define what resources they had, and make them available through an API. Every team within your company essential becomes a partner of the other.

https://apievangelist.com/2012/01/12/the-secret-to-amazons-success-internal-apis/

AWS history

SOAP requests

PDF

We have deprecated the SOAP API for Amazon EC2. As of 1 December 2015, we no longer support SOAP requests for any version of the API. If you use a SOAP request you receive the following response:

```
Client.UnsupportedProtocol: SOAP is no longer supported.
```

Similarly, the AWS Software Development Kits (SDKs) no longer support SOAP requests for any version of the API.

We recommend that you use the Query API for Amazon EC2, or the AWS SDKs. For more information, see Making requests to the Amazon EC2 API.

- Eucalyptus / RightScale

AWS News Blog

REST vs. SOAP

by Jeff Barr | on 13 FEB 2006 | Permalink | Share

Kevin Daily offers a brief thought on REST vs. SOAP in the case of the Amazon Web Services.

We are still seeing an 80% REST / 20% SOAP usage pattern. Kevin correctly notes that the ease with which developers can invoke an XSLT transformation as the concluding step in a REST request makes for easier and simpler web development.

— Jeff;

AWS Release Services

- 2006: S3, SQS, EC 2
- 2007: Amazon SimpleDB
- 2008: EIP, AZs, EBS, CloudFront
- 2009: VPC,EMR, ELB, AutoScaling, CloudWatch, RDS, EC 2 Spot
- 2010: SNS, CloudFormation, Route 53, Elastic Beanstalk
- 2011: SES
- 2012: DynamoDB, IaM, Glacier, Redshift

AWS Release Services

- 2013: CloudTrail, Kinesis
- 2014: eu-central-1, Amazon Aurora, KMS, ECS, Lambda, Resource Groups
- 2015: CodePipeline, API Gateway, Elasticsearch, ECR
- 2016: ACM, EFS, AWS Snowmobile, Lightsail, Polly, Rekognition, Lex, Pinpoint, StepFunctions, SystemsManager
- 2017: Amazon Glue, Aurora PostgreSQL, AWS Privatelink, Fargate, SageMaker, Translate, Transcribe
- 2018: SecretsManager, EKS, Textract, Personalize, Forecast, MSK, AppMesh

AWS Release Services

- 2019: AWS Outpost
- 2020: Amazon AppFlow
- 2020: Amazon Kendra
- 2020: Amazon Honeycode

kubernetes

- de-factor standard container orchestrator in private clouds

but:

Staff Developer Advocate, Google Cloud Platform

payara.cloud

- Payara Cluster -> Payara Kubernetes Operator
- payara server -> kubernetes operator
- payara micro -> pod
- WAR: packaging

AWS

- aws cli
- sam
- terraform
- java cdk
- java sdk
- cloudformation
- quarkus

networking

networking

- VPC
- global infrastructure
- region
- AZ
- CIDR
- subnet

networking

- Amazon Route 53
- Amazon CloudFront
- AWS Certificate Manager (ACM)
- Amazon CloudFront
- AWS Global Accelerator
- VPC Endpoint

VPC

- default VPC
- CIDR ranges (5 IP addresses are reserved)
  - 10.0.0.0/16
  - 172.31.0.0/16
  - 192.168.0.0/16

compute

compute

- EC 2, ELB
- ECS, ECR
- ECS Fargate
- EKS
- Amazon Lightsail
- AWS Lambda
- AWS Outposts

storage

storage

- S3
- instance storage
- EBS
- EFS
- RDS, Redshift
- Aurora
- DynamoDB, Athena, Neptune

IAM

AWS Identity and Access Management

- users
- groups
- roles
- policy
- resources
- identities
- permissions

sample policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListObjectsInBucket",
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": ["arn:aws:s3:::bucket-name"]
        },
        {
            "Sid": "AllObjectActions",
            "Effect": "Allow",
            "Action": "s3:*Object",
            "Resource": ["arn:aws:s3:::bucket-name/*"]
        }
    ]
}
```

IaC

Infrastructure as Code

- AWS Command Line Interface (CLI)
- AWS CloudFormation
- AWS Cloud Development Kit (AWS CDK)

CI/CD

continuous integration and deployment

- AWS CodeCommit
- AWS CodeBuild
- AWS CodeDeploy
- AWS CodePipeline

monitoring / observability

monitoring / observability

- Amazon CloudWatch
- CloudTrail
- SNS
- Amazon Managed Service for Grafana
- Amazon OpenSearch Service
- Amazon Managed Service for Prometheus
- AWS Distro for OpenTelemetry

airhacks.live

NEW online, live virtual workshops

"like airhacks.com, without leaving your home"

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

Live, Virtual Online Workshops, Winter 2021, [backends]:

Testing, Monitoring, Resiliency December 9th, 2021

Serverless Java on AWS Cloud, December 16th, 2021

by and with adam-bien.com

Thank You!