XPEPPERS

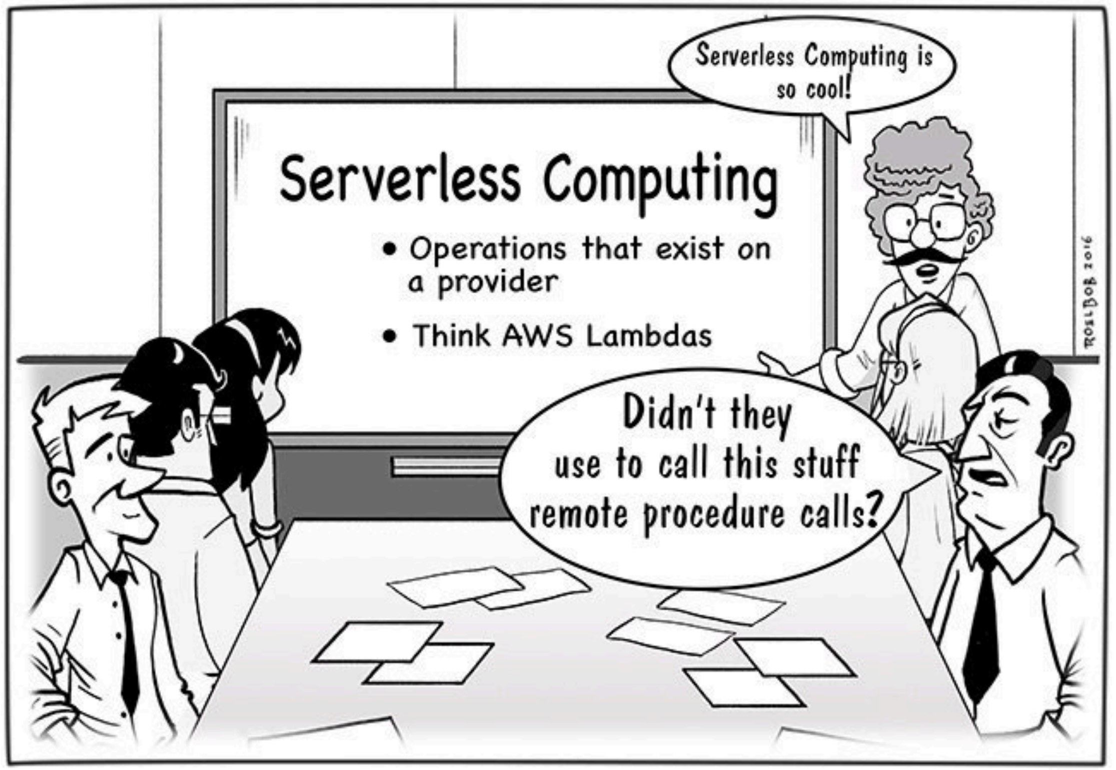
\ Serverless Architectures on AWS

Paolo Latella
@LatellaPaolo

Topics

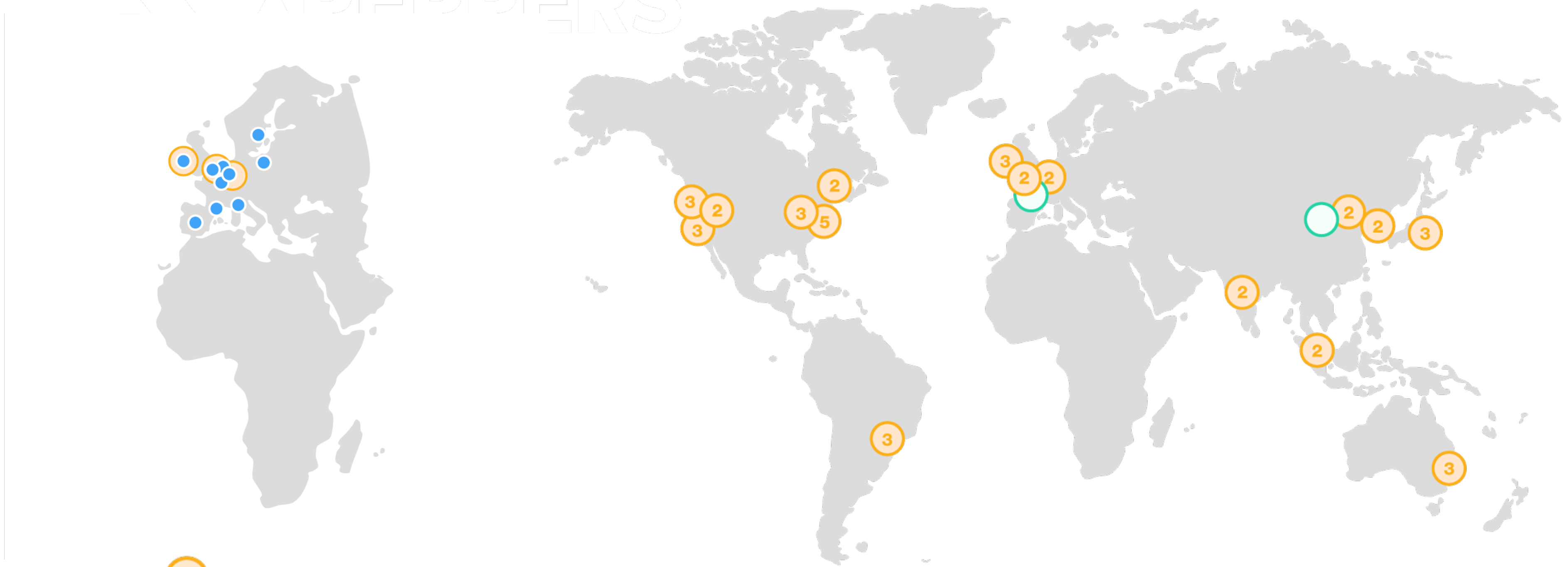
- AWS Serverless Building Block
- 3-Tier (Serverless) Architectures
    - Logic Tier - AWS Lambda and Amazon API Gateway
    - Authentication and Authorization
- Security
- Agility
- XPeppers Use Case

Serverless


Serverless Computing is so cool!
Serverless Computing
• Operations that exist on a provider
• Think AWS Lambdas
Didn't they use to call this stuff remote procedure calls?
ROSIBOB 2016

https://pbs.twimg.com/media/Cpx8XmGVIAEw2EI.jpg

AWS - Global Infrastructure



● # Regions and number of AZ

○ New Region (Paris)

● Edge Location (Milano)

AWS - Serverless Building Block

Amazon Athena

AWS WAF

Amazon API Gateway*

Amazon CloudFront

AWS IAM

Amazon Cognito

Amazon Kinesis

AWS CloudFormation

AWS CodeDeploy

SDK

Amazon SES

AWS Lambda

Amazon SNS

AWS CodePipeline

AWS CodeCommit

Amazon DynamoDB

Amazon S3

AWS CodeBuild

AWS IoT

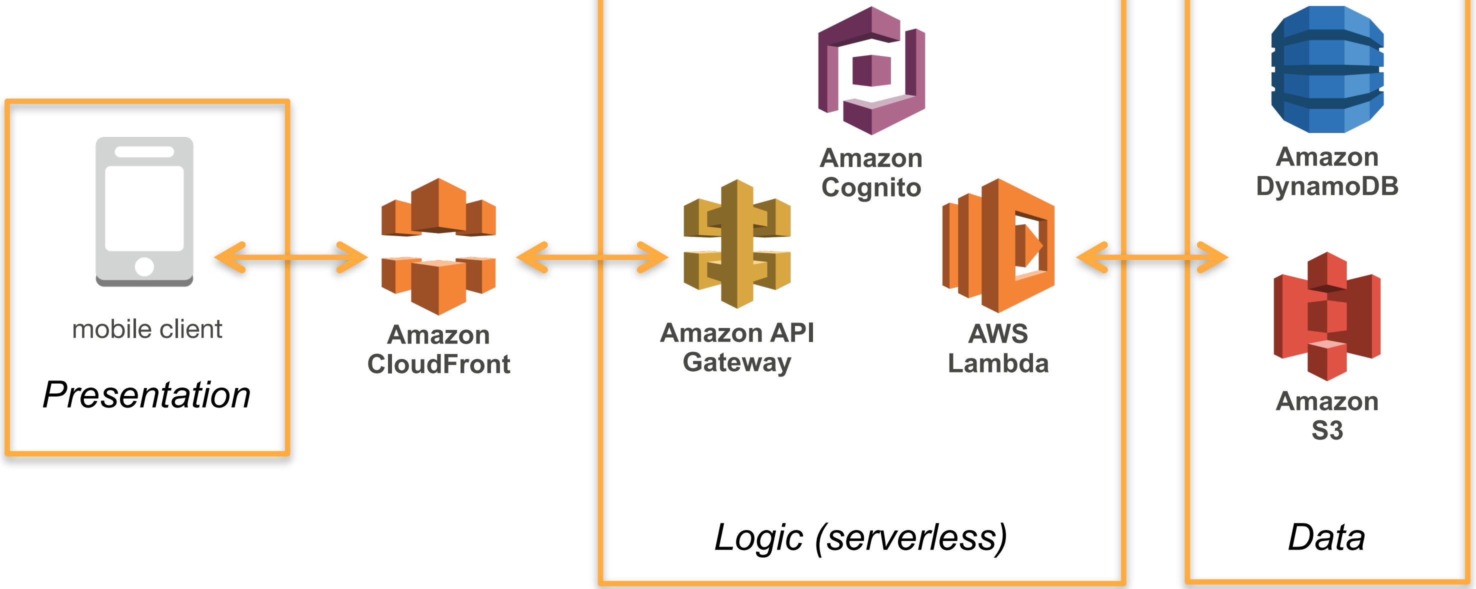
Amazon CloudWatch

Amazon SQS

AWS Step Functions

Amazon Machine Learning

AWS - 3-Tier (Serverless) architecture



The Logic Tier - Amazon API Gateway

Fully managed HTTPS service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale.

- Low-Cost
- Performance
- API Monitoring
- API Lifecycle management
- Flexible Security Controls
- Cloudfront, ELB and Lambda Integration

The Logic Tier - AWS Lambda

Serverless compute service that runs your code in response to events and automatically manages the underlying resources

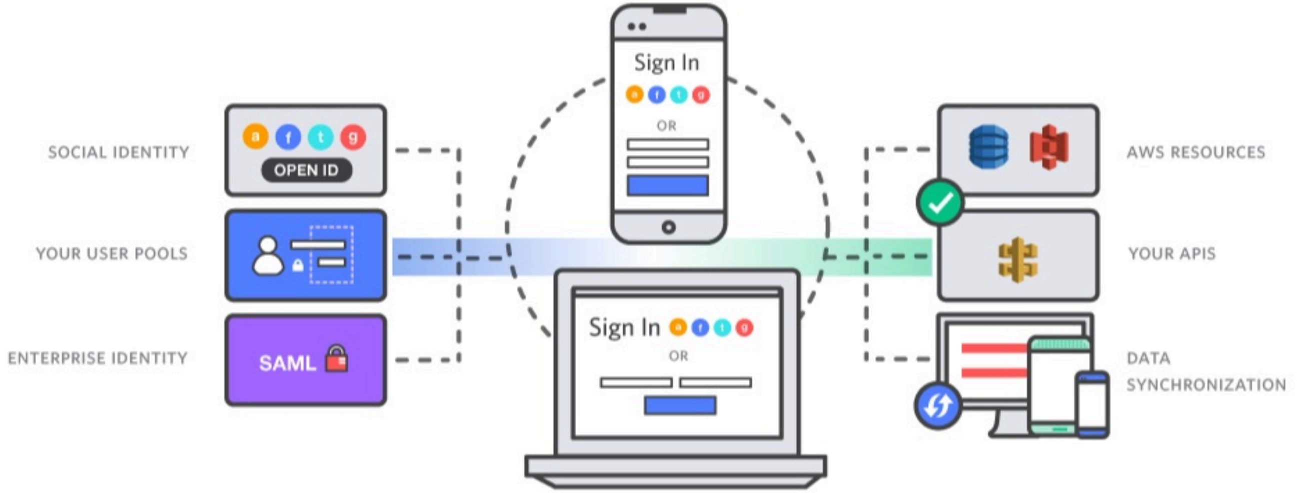
- Completely Automated Administration
- Built-in Fault Tolerance
- Run Code at Edge Locations and Step Function (Re:Invent 2016)
- Pay-per-use
- Supports Java, Node.js, C#, and Python
- VPC Integration

\ Authentication & Authorization

Authentication

\ Authentication - Amazon Cognito

Cognito is a fully managed service for *sign-up/sign-in* functionality and can scale to hundreds of millions of users



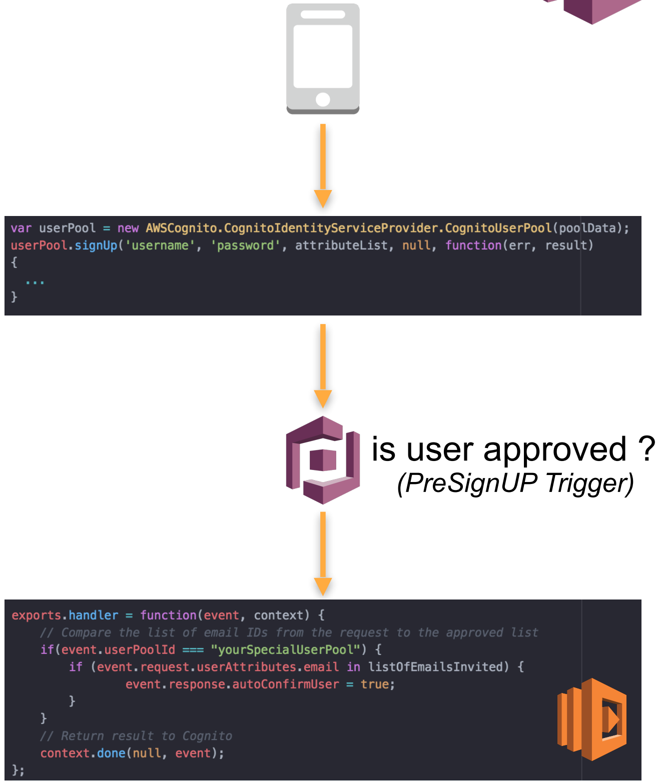
Amazon Cognito

Amazon Cognito lets you easily add user sign-up and sign-in to your mobile and web apps.

- Simple, secure, and low-cost authentication options
- Authenticate users on Facebook, Twitter, or Amazon
- Cognito Authentication Flow
- Custom Authentication Flow (AWS Lambda triggers)
- Enhanced security features, such as email and phone number verification, and multi-factor authentication

Amazon Cognito - Custom Authentication Flow

triggerSource value	Triggering event
PreSignUp_SignUp	Pre-sign up
PostConfirmation_ConfirmSignUp	Post confirmation
PreAuthentication_Authentication	Pre authentication
PostAuthentication_Authentication	Post authentication
CustomMessage_SignUp	Custom message – To send confirmation code post sign-up
CustomMessage_ResendCode	Custom message – To resend confirmation code to an existing user
CustomMessage_ForgotPassword	Custom message – To send confirmation code for Forgot Password request
CustomMessage_UpdateUserAttribute	Custom message – When a user's email or phone number is changed, this trigger sends a verification code automatically to the user. Cannot be used for other attributes.
CustomMessage_VerifyUserAttribute	Custom message – This trigger sends a verification code to the user when the they manually request it for a new email or phone number.
CustomMessage_Authentication	Custom message – To send MFA code during authentication
DefineAuthChallenge_Authentication	Define Auth Challenge
CreateAuthChallenge_Authentication	Create Auth Challenge
VerifyAuthChallengeResponse_Authentication	Verify Auth Challenge Response



```
var userPool = new AWSCognito.CognitoIdentityServiceProvider.CognitoUserPool(poolData);
userPool.signUp('username', 'password', attributeList, null, function(err, result)
{
  ...
}
```

is user approved ?

(PreSignUP Trigger)

```
exports.handler = function(event, context) {
    // Compare the list of email IDs from the request to the approved list
    if(event.userPoolId === "yourSpecialUserPool") {
        if (event.request.userAttributes.email in listOfEmailsInvited) {
                event.response.autoConfirmUser = true;
        }
    }
    // Return result to Cognito
    context.done(null, event);
};
```

http://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-identity-pools-working-with-aws-lambda-triggers.html

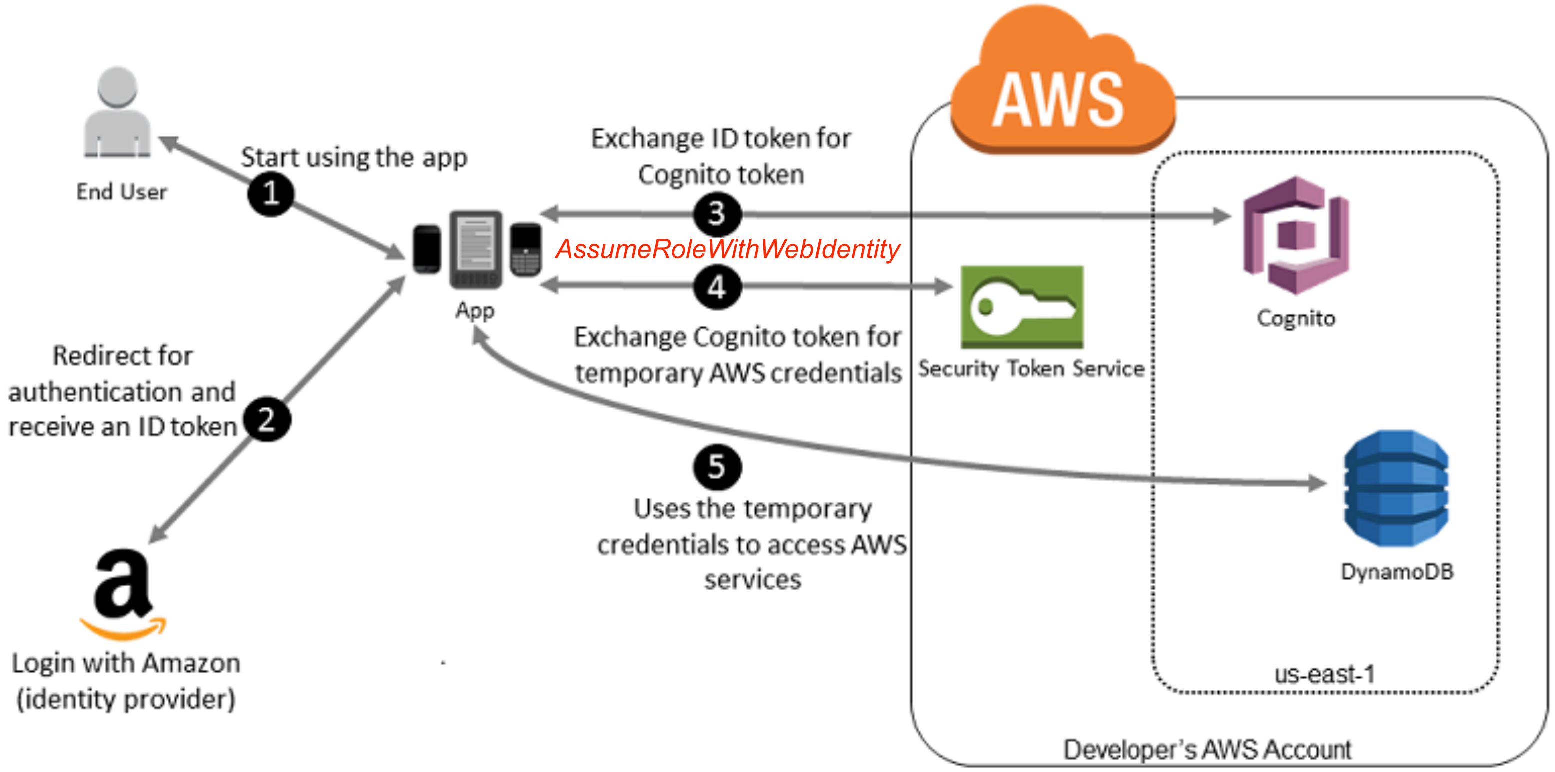
\ Authentication & Authorization

Authorization

Authorization

- AWS **Identity and Access Management** (IAM) is a web service that helps you securely control access to AWS resources
  - Roles
  - Policies
- AWS **Security Token Services** (STS) is a web service that enables you to request temporary, limited-privilege credentials for IAM
  - Use action *AssumeRoleWithWebIdentity* for federation

Authorization - Web Identity Federation


End User
1 Start using the app
App
Exchange ID token for Cognito token
3
AssumeRoleWithWebIdentity
4
Exchange Cognito token for temporary AWS credentials
Redirect for authentication and receive an ID token
2
5
Uses the temporary credentials to access AWS services
Login with Amazon (identity provider)
AWS
Security Token Service
Cognito
DynamoDB
us-east-1
Developer's AWS Account

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_oidc_cognito.html

Authorization - API Gateway Access Control

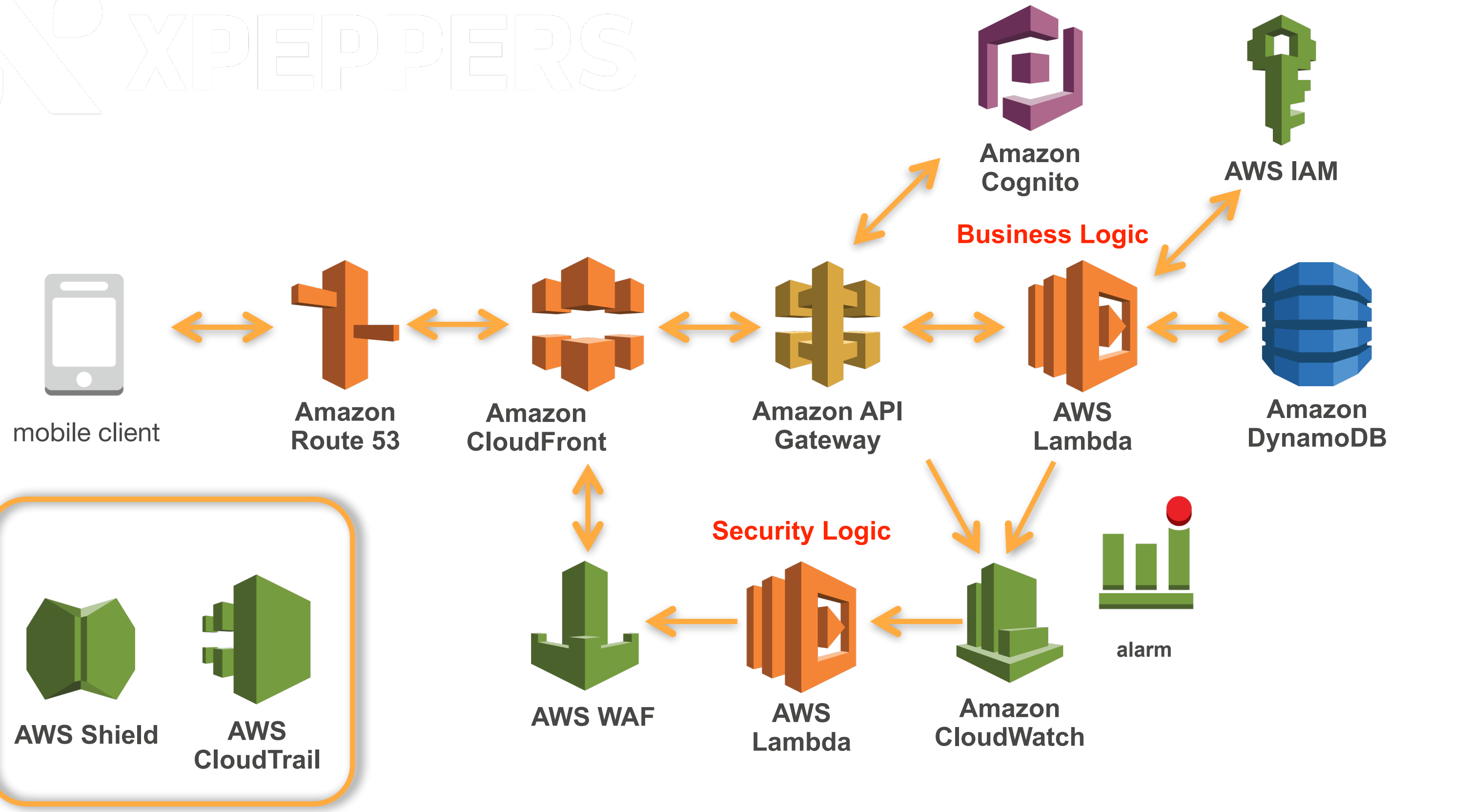
API Gateway supports multiple mechanisms of access control, including metering or tracking API uses by clients using API keys.

- Control Access for Managing an API
  - IAM Policy (`“Action": “apigateway:*”`)
- Control Access for Invoking an API
  - IAM Policy (`“Action”: “execute-api:Invoke”`)
  - Custom Authorizers (Lambda functions)
  - Amazon Cognito User Pools

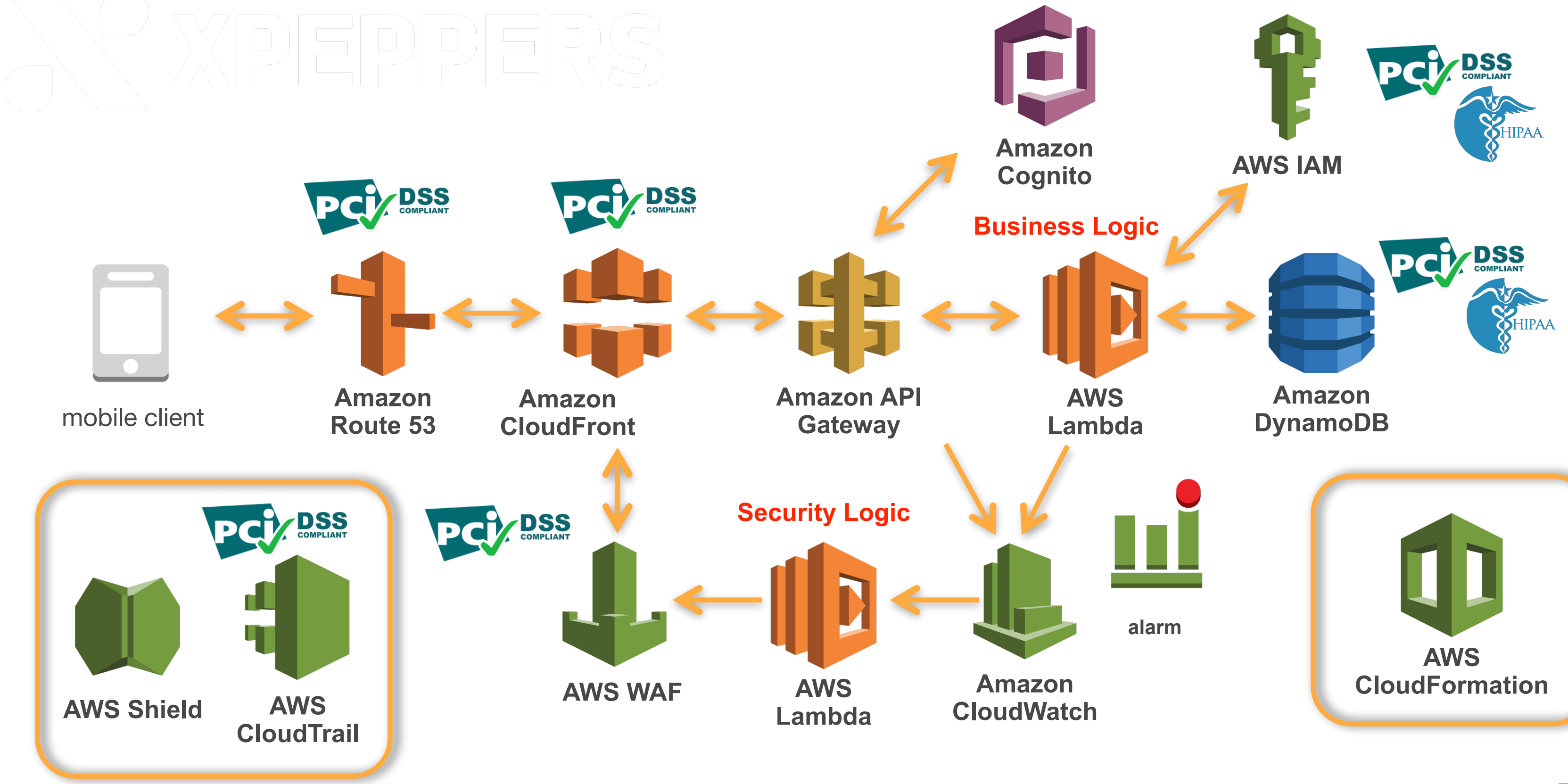
Security

- AWS IAM and Cognito
  - Authentication and Authorization
- AWS Cloudtrail
  - Monitor All APIs Management (Cognito, Lambda, API Gateway, etc)
- AWS Cloudwatch
  - Monitor All APIs Execution and related resources (Dynamodb, ELB, etc)
- AWS WAF (Web Application Firewall) and AWS Shield
  - Access Control and DDoS Mitigation
- AWS Lambda
  - Automate your security!

Security - always serverless



Security - always serverless - compliance



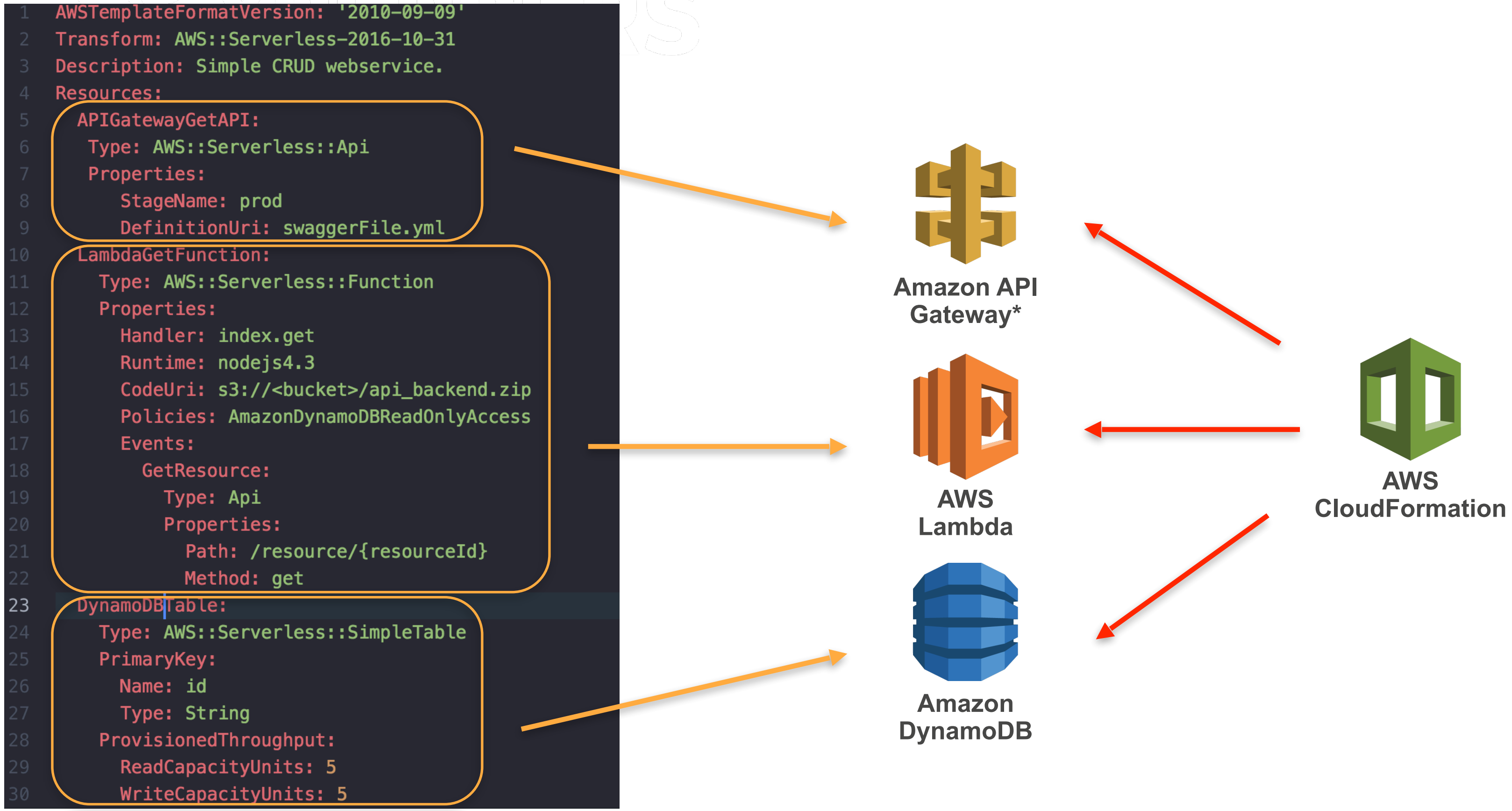
https://aws.amazon.com/compliance/services-in-scope/

Agility

- Serverless Application Model (SAM)
  - Define the application and its resources
- AWS CodeCommit
  - Your source repository
- AWS CodeBuild
  - Package your source code and SAM templates
- AWS CloudFormation
  - Deploy your Infrastructure as Code
- AWS CodePipeline
  - Orchestrate your application deployment.

All Serverless

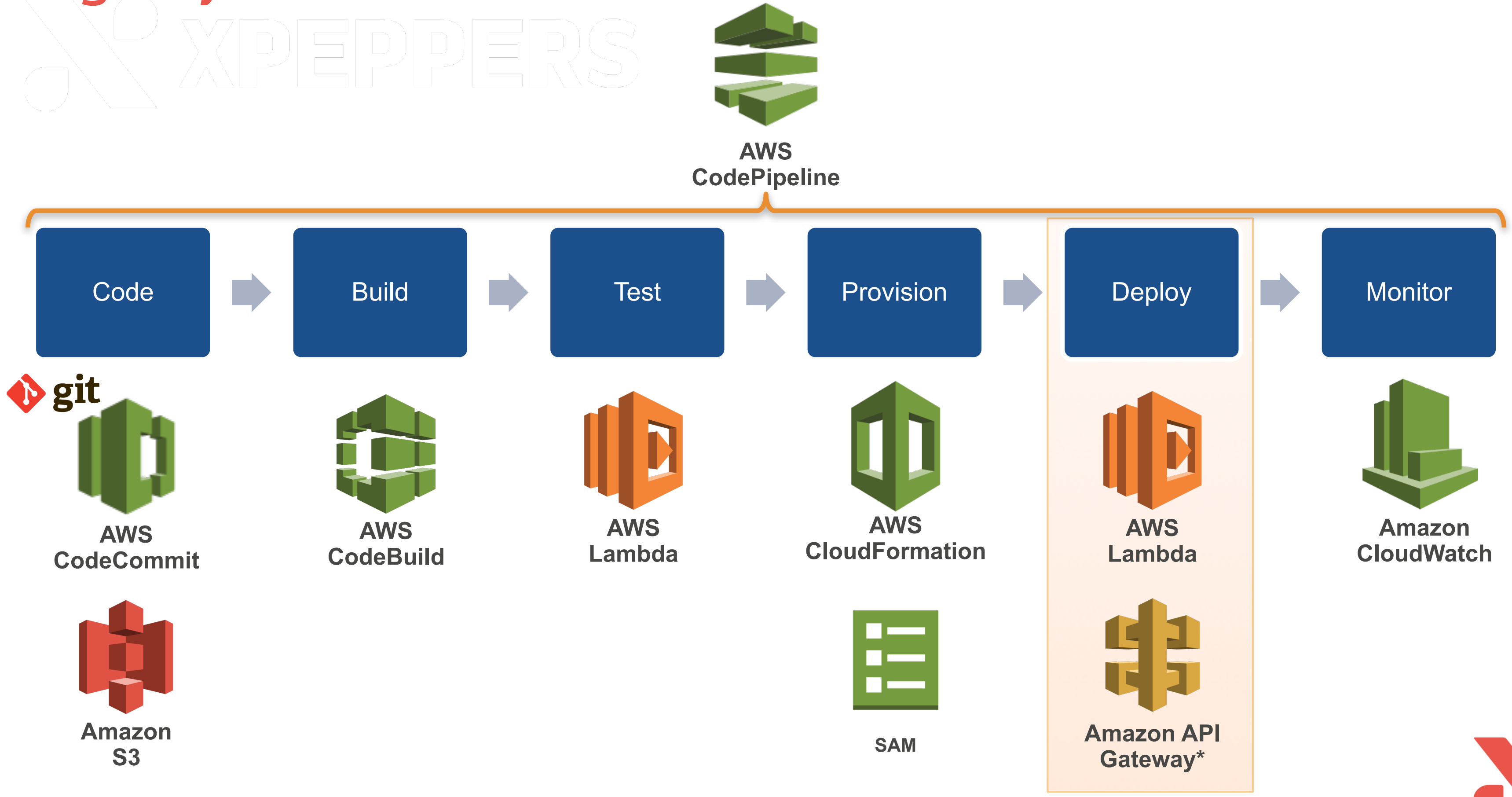
Agility - Serverless Application Model (SAM)



```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Description: Simple CRUD webservice.
Resources:
  APIGatewayGetAPI:
   Type: AWS::Serverless::Api
   Properties:
      StageName: prod
      DefinitionUri: swaggerFile.yml
  LambdaGetFunction:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.get
      Runtime: nodejs4.3
      CodeUri: s3://<bucket>/api_backend.zip
      Policies: AmazonDynamoDBReadOnlyAccess
      Events:
        GetResource:
          Type: Api
          Properties:
            Path: /resource/{resourceId}
            Method: get
  DynamoDBTable:
    Type: AWS::Serverless::SimpleTable
    PrimaryKey:
      Name: id
      Type: String
    ProvisionedThroughput:
      ReadCapacityUnits: 5
      WriteCapacityUnits: 5
```

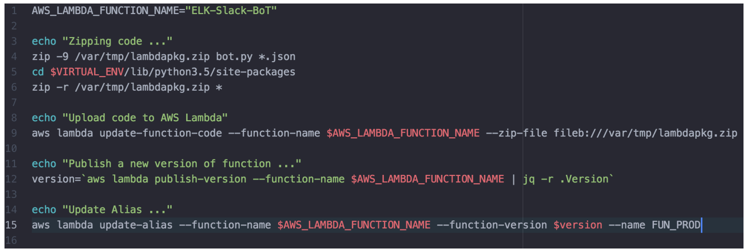
https://github.com/awslabs/serverless-application-model

Agility - CI & CD



Agility - CI&CD

AWS CodePipeline



```
AWS_LAMBDA_FUNCTION_NAME="ELK-Slack-BoT"

echo "Zipping code ..."
zip -9 /var/tmp/lambdapkg.zip bot.py *.json
cd $VIRTUAL_ENV/lib/python3.5/site-packages
zip -r /var/tmp/lambdapkg.zip *

echo "Upload code to AWS Lambda"
aws lambda update-function-code --function-name $AWS_LAMBDA_FUNCTION_NAME --zip-file fileb:///var/tmp/lambdapkg.zip

echo "Publish a new version of function ..."
version=`aws lambda publish-version --function-name $AWS_LAMBDA_FUNCTION_NAME | jq -r .Version`

echo "Update Alias ..."
aws lambda update-alias --function-name $AWS_LAMBDA_FUNCTION_NAME --function-version $version --name FUN_PROD
```

AWS Lambda Versioning

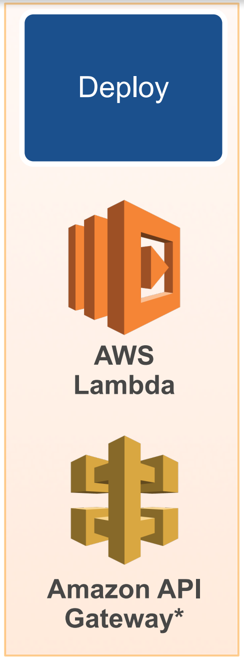
AWS Lambda Aliasing

AWS Lambda Env Variables

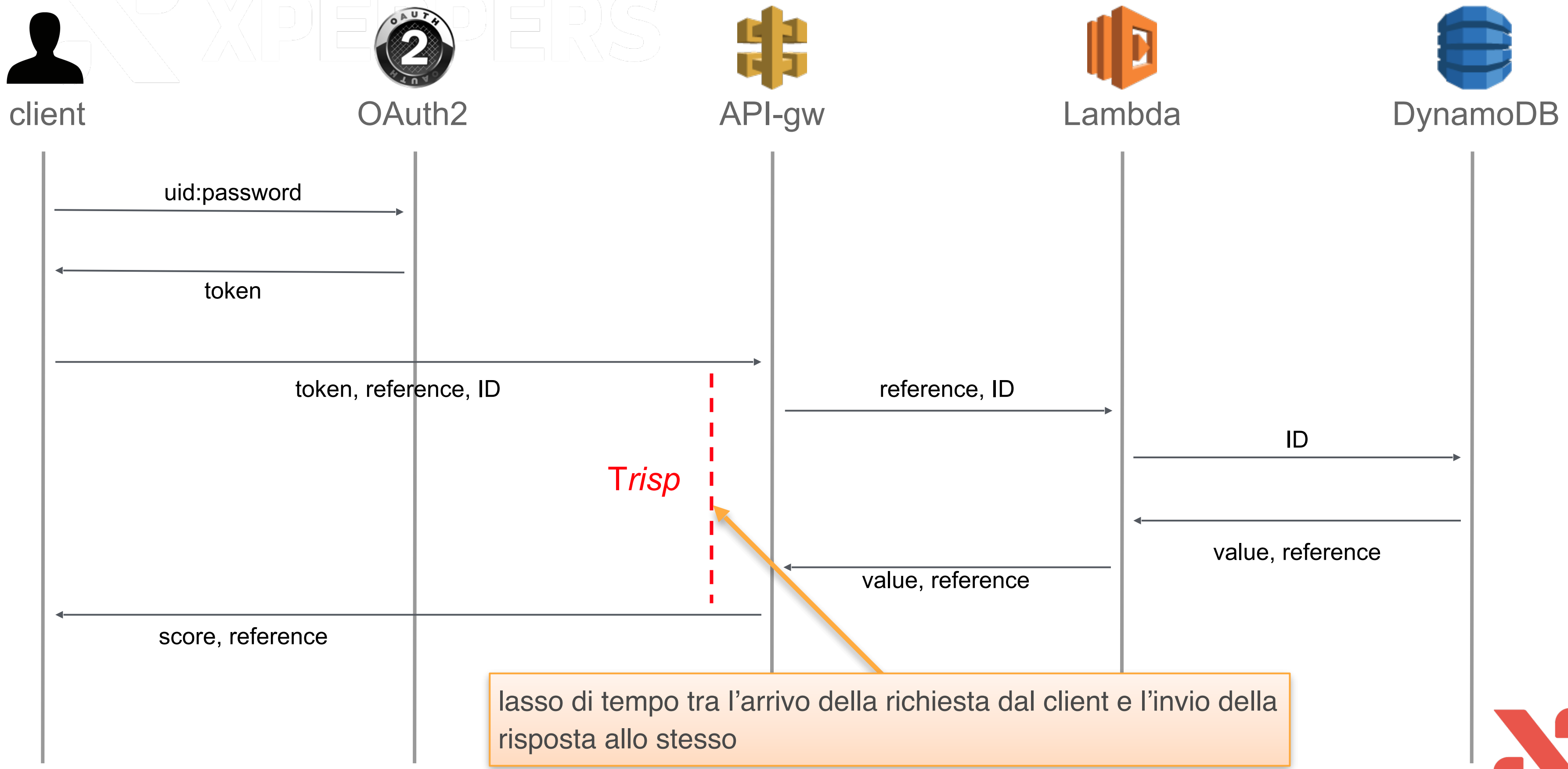
API Stage

API Stage Variables

API Swagger import/export

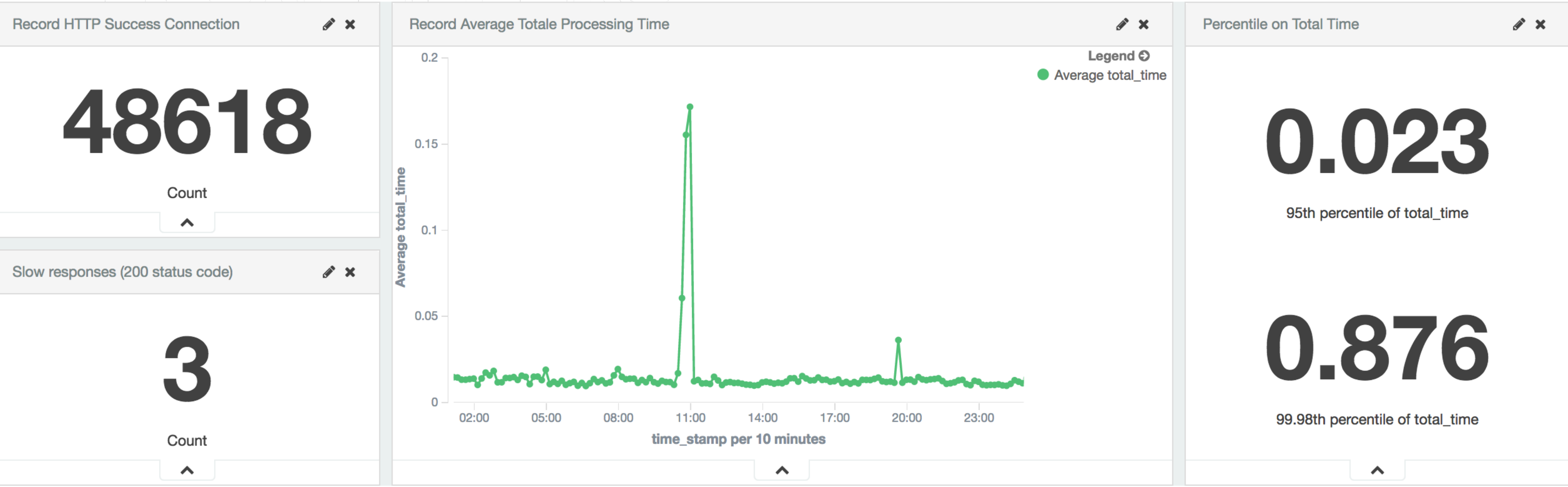


Use Case - Flusso



lasso di tempo tra l’arrivo della richiesta dal client e l’invio della risposta allo stesso

Use Case - Monitoring



Record HTTP Success Connection

48618

Count

Slow responses (200 status code)

3

Count

Record Average Totale Processing Time

Percentile on Total Time

0.023

95th percentile of total_time

0.876

99.98th percentile of total_time

ElasticSearch Service

Use Case - Costo API-Gateway

EU-West-1

Costo totale = Costo chiamate API + Costo trasferimento + Costo cache

Costo chiamate API

3,50 USD per milione di chiamate API ricevute

Costo Cache

- **0,5 GB = 0,020 USD/Ora**
- 1,6 GB = 0,038 USD
- 6,1 GB = 0,200 USD

...

Costo Trasferimento

Tariffe per il trasferimento di dati in uscita di Amazon API Gateway

- **0,09 USD/GB per i primi 10 TB**
- 0,085 USD/GB per i successivi 40 TB
- 0,07 USD/GB per i successivi 100 TB
- 0,05 USD/GB per i successivi 350 TB

Use Case - Costo API Gateway

EU-West-1

Costo Chiamate API

- 48.618 * 31 = 1.507.158 * 3,75 USD -> **5,275 USD/Mese**

Costo Cache (0,5 GB)

- 0,020 USD/ora * 24 ore * 31 giorni = **14,88 USD/Mese**

Costo trasferimento

- 48.618 * 512 Byte * 31 giorni * 0.09 USD = **0,064 USD/Mese**

Costo Totale = 20,219 USD/Mese

* = Non considerando il Free Tier

Use Case - Costo Lambda

EU-West-1

Costo totale = Costo elaborazione + Costo richieste

Elaborazione

La durata viene calcolata a partire dal momento in cui viene avviata l'esecuzione del codice e fino al momento in cui viene restituito o comunque terminato il codice, arrotondata al decimo di secondo più vicino. Il prezzo dipende dalla quantità di memoria allocata per la funzione. **Il costo è di 0,00001667 USD per ogni GB/secondo impiegato.**

Richieste

Il costo viene calcolato in base al numero totale di richieste per tutte le funzioni. Lambda conteggia una richiesta ogni volta che avvia un'elaborazione in risposta alla notifica di un evento o a una chiamata Invoke. **Il costo è di 0,20 USD ogni milione di richieste (0,0000002 USD a richiesta).** Il primo milione di richieste ogni mese è gratuito

Use Case - Costo Lambda

EU-West-1

Costo Elaborazione

- Free Tier: 400.000 GB/secondo al mese
- Il costo dell'elaborazione mensile è 0,00001667 USD per GB/s
- Elaborazione (sec) = (48.618 * 99.98%) * 0.876 sec = 42581 sec
- Elaborazione (GB/s) = 42581 secondi * (512 MB / 1024) = 21.290 GB/s
- Costi Elaborazione = 0,00001667 USD * 21.290 * 31 = **10,964 USD/Mese***

Costo Richieste

- Free Tier: 1.000.000 richieste/mese
- Dalla richiesta successiva, il costo è di 0,20 USD ogni milione di richieste
- Costi Richieste 48.618 * 0,0000002 USD = **0,30 USD/Mese***

Costo Totale = 11,264 USD/Mese

* = Non considerando il Free Tier

Use Case - Costo DynamoDB

EU-West-1

Costo totale = Costo storage + Costo capacità + Costo Trasferimento

Storage

First 25 GB stored per month is free

$0,283 per GB al mese successivamente

Trasferimento

Trasferimento IN = **$0,000 per GB**

Trasferimento OUT = **$0,090 per GB**

Capacità Read/Write

Consente di specificare il throughput per le richieste in lettura e scrittura che deve raggiungere la propria tabella:

- Throughput scrittura: **$0,00735 all'ora ogni 10 unità** di capacità in scrittura
- Throughput lettura: **$0,00735 all'ora ogni 50 unità** di capacità in lettura

Use Case - Costo DynamoDB

EU-West-1

Costo Storage

- Storage = 2.5 GB -> Costo = 0 USD/Mese*

Costo Capacità

- Totale RU = 50 -> Costo = $0.00735 * 24 ore * 31 giorni = **5,47 USD/Mese**
- Totale WU = 10 -> Costo = $0.00735 * 24 ore * 31 giorni = **5,47 USD/Mese**

Costo Trasferimento OUT

- 48.618 * 512 Byte * 31 = 0.718 GB/Mese -> Costo = 0**

Costo Totale = 10,94 USD/Mese

* = First 25 GB stored per month is free
** = Primo GB/mese is free

\ Use Case - Totale

Costo Totale API Gateway = 20,219 USD/Mese

Costo Totale Lambda = 11,264 USD/Mese

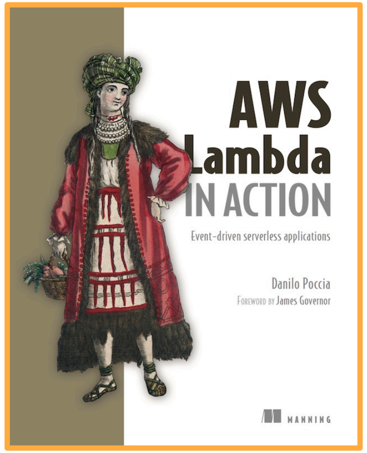
Costo Totale Dynamo = 10,94 USD/Mese

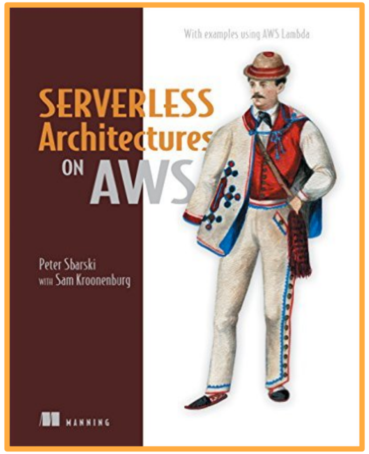
Costo Totale ElasticSearch = 15 USD/Mese

Costo Altri servizi = 10 USD/Mese

Costo Totale = 67,423 USD/Mese

Reference


AWS Lambda IN ACTION
Event-driven serverless applications
Danilo Poccia
Foreword by James Governor
MANNING


With examples using AWS Lambda
SERVERLESS Architectures ON AWS
Peter Sbarski with Sam Kroonenburg
MANNING

https://www.manning.com/books/aws-lambda-in-action

https://www.manning.com/books/serverless-architectures-on-aws

http://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-identity-pools-working-with-aws-lambda-triggers.html

http://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_providers_oidc_cognito.html

https://aws.amazon.com/compliance/services-in-scope/

https://github.com/awslabs/serverless-application-model

XPEPPERS

www.xpeppers.com

@xpeppers /xpepperssrl