

# Cloud based storage with 4D and S3

4D Method - October 12<sup>th</sup> 2016



Bruno LEGAY

A&C Consulting  
[www.ac-consulting.fr](http://www.ac-consulting.fr)

# Cloud Computing Definition

# Definition

Cloud computing : “delivery of hosted services over the Internet”



# Characteristics

- On demand
- Broad network access
- Resource pooling
- Elasticity
- Measured service

# Service Models

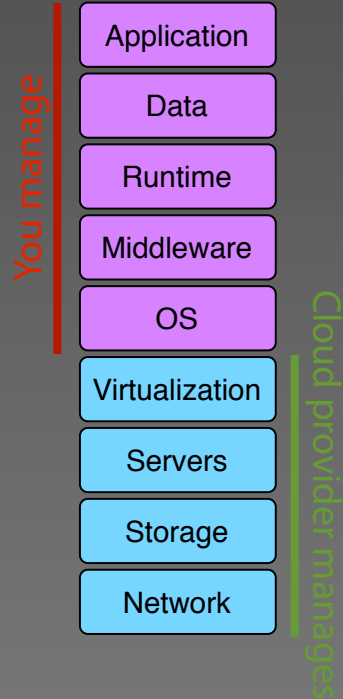
- IaaS (Infrastructure as a Service)
- PaaS (Platform as a Service)
- SaaS (Software as a Service)

# Service Models (cont.)

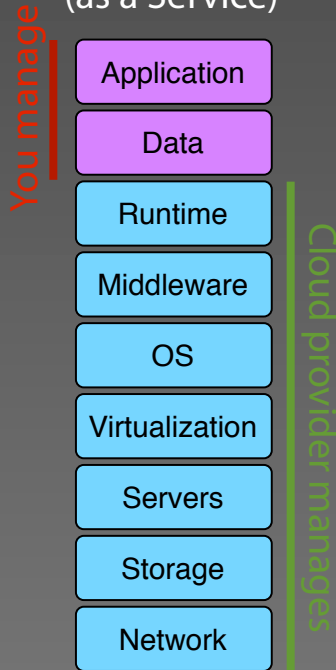
On premise  
(for comparison)



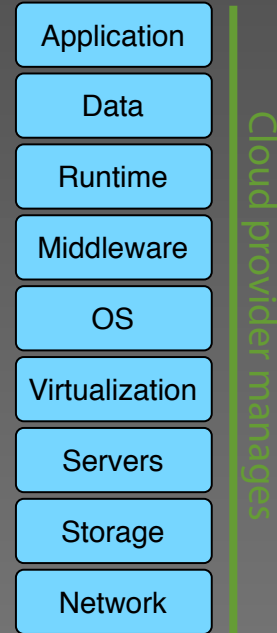
Infrastructure  
(as a Service)



Platform  
(as a Service)



Application  
(as a Service)



Why Cloud  
Computing ?



# Cloud advantages

- Reliability
- Scalability
- Cost
- Security

# Cloud drawbacks

- Cost estimation
- Vendor «lock-in»
- Ownership / security ?

# Leveraging the Cloud with 4D

# One use case

- 4D db with many large blobs (>30 Mb)
- Document with metadata
- Blob creation rate is low
- Blobs are rarely modified
- Blob read is low

# The problem

- % of blob in your data
- Managing large data files
- Backup : full vs differential

# Options

Store	Infrastructure managed by	Integrity managed by	Transaction	Security	Backup
blob (record)	na	4D	4D	na	yes
blob (data file)	na	4D	4D	na	yes
blob (disk)	na	4D	4D	na	yes
file (local disk)	you	you	you	you	you
file (shared disk)	you	you	you	you	you
cloud	cloud	you	you	you + cloud	you + cloud

# Our approach

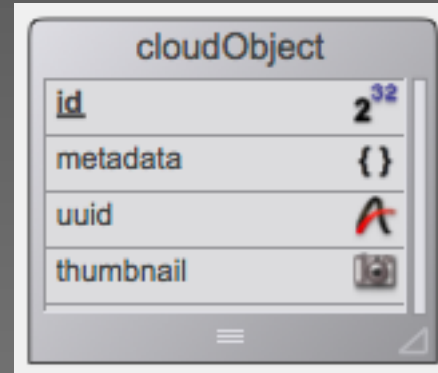
# Everything is an object


- A file is an object
- A blob is an object
- But for REST an object is a resource



# Cloud Object

- An object is identified with UUID
- Metadata
- Thumbnail



cloudObject	
<u>id</u>	2 <sup>32</sup>
metadata	{ }
uuid	A
thumbnail	

# Resource metadata

- Filename
- Creation, modification date
- Type (pdf, jpg, png, xml, json, etc...)
- Size, checksum (md5)
- Pdf : number of pages
- Image : resolution, exif, etc...

Amazon AWS

# Amazon AWS

- Visionary
- Leader
- Eat your own dog food
- Documentation

<https://aws.amazon.com/documentation/>

# Services

- S3 : object storage
- EBS : block storage
- EC2 : virtual machines
- Glacier : archiving
- DynamoDB : NoSQL database
- Beanstalk : app deployment
- RDS : managed databases
- etc

# Creating an account

- Access to AWS console
- Manage services
- Get credentials
  - «AccessKeyId» : 20 chars
  - «SecretAccessKey» : 40 chars

# AWS regions

- 11 regions
- Check availability of service
- Select your region (close to you)

Region	Continent	Town
us-east-1	US East	N. Virginia
us-west-2	US West	Oregon
us-west-1	US West	N. California
us-gov	US West	?
eu-west-1	EU	Ireland
eu-central-1	EU	Frankfurt
ap-south-1	Asia Pacific	Mumbai
ap-southeast-1	Asia Pacific	Singapore
ap-northeast-1	Asia Pacific	Tokyo
ap-southeast-2	Asia Pacific	Sydney
ap-northeast-2	Asia Pacific	Seoul
cn-north-1	Asia Pacific	Beijing
sa-east-1	South America	Sao Paulo





# Region & Number of Availability Zones  
New Region Coming Soon

# AWS endpoint

- Host/domain part of URL
- Example :

s3-eu-central-1.amazonaws.com

S3 service      EU - Frankfurt

# AWS tools and APIs

- Online : AWS Management Console
- Tool : AWS CLI
- SDKs (Java, PHP, Python, Ruby, etc)
- APIs :
  - RESTful APIs
  - SOAP APIs : deprecated

# AWS command line

- Python based tool
- Unified tool for all services
- Cross platform
- Can be used in 4D with LEP

# AWS command line (cont.)

- Download / install
- Configure

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: ENTER
```

# AWS command line (cont.)

- Syntax/args are service specific
- Example : `$ aws <service> ...`
- Output : json, text (tab / table)
- Documentation

<http://docs.aws.amazon.com/cli/latest/reference/>

# AWS APIs

- REST based
- Common authentication mechanism
- Service specific APIs

Amazon S3



# S3 cost

- 1 Gb  $\approx$  0.03 \$/month
- Cost vary depending on region
- Cost calculator

<http://calculator.s3.amazonaws.com/index.html>

# S3 resources

- Bucket (root container)
- Object
- Identified by a «key»
- Key example :

```
map/  
map/images/  
map/images/logo.png
```



# S3 object properties

- Basic (key, size, dates, ...)
- Expiry date
- Storage (standard, reduced, ...)
- Encryption (none, AES-256)
- Permissions
- Metadata

# S3 url

- Two possible syntax
- Examples with bucket «myBucket» :

`https://myBucket.s3-eu-central-1.amazonaws.com`

`https://s3-eu-central-1.amazonaws.com/myBucket`

# S3 url (cont.)

- With key

`https://myBucket.s3-eu-central-1.amazonaws.com/<object_key>`

`https://myBucket.s3-eu-central-1.amazonaws.com/map/images/logo.png`

`https://s3-eu-central-1.amazonaws.com/myBucket/map/images/logo.png`

- With key and query string

`https://s3-eu-central-1.amazonaws.com/myBucket/map/images/logo.png?acl`

# S3 cli services

- Two services

- s3

```
$ aws s3 ...
```

- s3 api

```
$ aws s3api ...
```

- Documentation

<http://docs.aws.amazon.com/cli/latest/reference/s3api/>

<http://docs.aws.amazon.com/cli/latest/reference/s3/>

# S3 REST operations

- Operations on buckets
- Operations on objects
- S3 responses format is XML or JSON
- Documentation

<http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketOps.html>

<http://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectOps.html>

# S3 auth/signature

- Two possible methods
  - HTTP headers
  - Query string parameters
- Documentation

<http://docs.aws.amazon.com/AmazonS3/latest/API/sig-v4-header-based-auth.html>

<http://docs.aws.amazon.com/AmazonS3/latest/API/sigv4-query-string-auth.html>



# AWS signature

## 1. StringToSign

A string based on select request elements

## 2. Signing Key

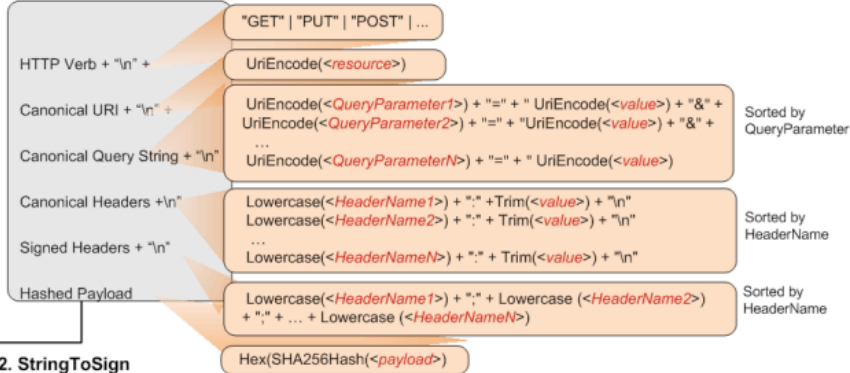
DateKey = HMAC-SHA256 ("AWS4" + "<SecretAccessKey>", "<yyyymmdd>")  
DateRegionKey = HMAC-SHA256(DateKey, "<aws-region>")  
DateRegionServiceKey = HMAC-SHA256(DateRegionKey, "<aws-service>")  
SigningKey = HMAC-SHA256(DateRegionServiceKey, "aws4\_request")

## 3. Signature

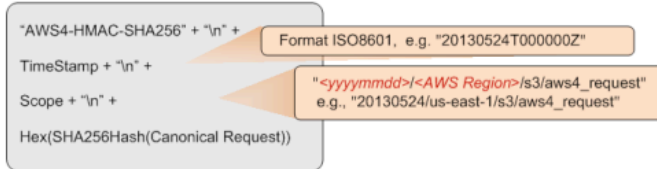
`signature = Hex(HMAC-SHA256(SigningKey, StringToSign))`

# S3 signature

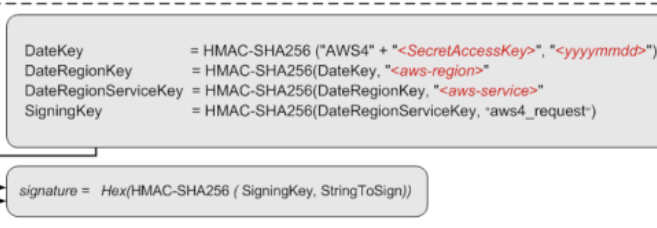
## 1. Canonical Request



## 2. StringToSign



## 3. Signature



# AWS Component

# CLI helper

```
// call once (AWS client needs to be configured)
AWS_cliPathSet ("/Users/ble/.local/lib/aws/bin/aws")

// http://docs.aws.amazon.com/cli/latest/reference/s3/
C_POINTER($vp_nil)
C_TEXT($vt_list)
AWS_cliRun ("s3 ls s3://ac-consulting-test/test/demo/"\
    ;False;$vp_nil;->$vt_list)
ALERT($vt_list)
```



# CLI helper (cont.)

```
// http://docs.aws.amazon.com/cli/latest/reference/s3api/
// we will use json :-)

// first get informations of the object with the "head-object" the json as text and parse the json
C_TEXT($vt_json)
C_POINTER($vp_nil)
AWS_cliRun ("s3api "+\
"head-object "+\
"--bucket ac-consulting-test "+\
"--key test/demo/imagePrivate.png";\
False;$vp_nil;->$vt_json)

C_OBJECT($vo_objectProperties)
$vo_objectProperties:=JSON Parse($vt_json)
// {
//   "AcceptRanges": "bytes",
//   "ContentType": "image/png",
//   "LastModified": "Mon, 04 Jan 2016 09:43:06 GMT",
//   "ContentLength": 288548,
//   "ETag": "\"43ef6cf259c8183e4c79d584ca03733c\"",
//   "StorageClass": "REDUCED_REDUNDANCY",
//   "ServerSideEncryption": "AES256",
//   "Metadata": {}
// }
```

# CLI helper (cont.)

```
// wouldn't it be nice if we could get the json directly from the AWS_cliRun ?

// list objects in the bucket
C_POINTER($vp_nil)
C_OBJECT($vo_objectProperties)
CLEAR_VARIABLE($vo_objectProperties)
AWS_cliRun ("s3api "+\
"head-object "+\
"--bucket ac-consulting-test "+\
"--key test/demo/imagePrivate.png";\
False;$vp_nil;->$vo_objectProperties)
// {
//   "AcceptRanges": "bytes",
//   "ContentType": "image/png",
//   "LastModified": "Mon, 04 Jan 2016 09:43:06 GMT",
//   "ContentLength": 288548,
//   "ETag": "\"43ef6cf259c8183e4c79d584ca03733c\"",
//   "StorageClass": "REDUCED_REDUNDANCY",
//   "ServerSideEncryption": "AES256",
//   "Metadata": {}
// }
```

# CLI helper (cont.)

```
// get a resource into a blob
C_POINTER($vp_nil)
C_BLOB($vp_blob)
AWS_cliRun ("s3 cp s3://ac-consulting-test/test/demo/imagePrivate.png -" \
;False;$vp_nil;->$vp_blob)
```



# AWS Rest API

## Managing credentials

```
C_TEXT($vt_region;$vt_AWSAccessKeyId;$vt_AWSSecretKey)
C_BOOLEAN($vb_interprocess)

$vt_region:="us-west-2"
$vt_AWSAccessKeyId:="AKIAIOSFODNN7EXAMPLE"
$vt_AWSSecretKey:="wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY"
$vb_interprocess:=True

// The parameters are set for the next S3_restApi calls
AWS_paramSet ($vt_region;$vt_AWSAccessKeyId;$vt_AWSSecretKey;$vb_interprocess)
```

# AWS Rest API

## Calling the S3 API

```
ARRAY TEXT($tt_requestHeadersArray;0)
ARRAY TEXT($tt_responseHeadersArray;0)
C_BLOB($vx_requestBodyBlob;$vx_responseBodyBlob)
C_TEXT($vt_httpVerb;$vt_bucket;$vt_uri)
$vt_httpVerb:=HTTP_get_method //"GET"
$vt_bucket:="ac-consulting-test"
$vt_uri:="/"

C_LONGINT($vl_httpReponseStatus) //200 <=> OK
$vl_httpReponseStatus:=S3_restApi ($vt_httpVerb;$vt_bucket;$vt_uri;\
->$tt_requestHeadersArray;\
->$vx_requestBodyBlob;\
->$tt_responseHeadersArray;\
->$vx_responseBodyBlob)
```

# Demonstration

Q&A ?

# Thank You !



Bruno LEGAY

A&C Consulting

[www.ac-consulting.fr](http://www.ac-consulting.fr)

[www.linkedin.com/in/brunolegay](http://www.linkedin.com/in/brunolegay)