

# How AWS handles security

Aurelijus Banelis





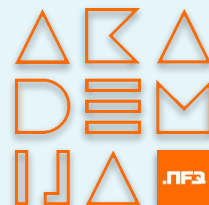
# Aurelijus Banelis

## Backend/DevOps

**aurelijus.banelis.lt**

**aurelijus@banelis.lt**

PGP 0x320205E7**539B6203**  
130D C446 1F1A 2E50 D6E3  
3DA8 3202 05E7 539B 6203



The background of the slide is a dark, grayscale photograph of a bridge's structural framework, featuring a complex network of steel beams and trusses. The lighting is dramatic, with some parts of the structure highlighted against a dark sky.

# Security patterns in AWS

# **Introduction**

**What is AWS**  
**Cloud vs Hosting**  
**Core security tools**

# **By comparison**

**Monolithic vs distributed**  
**Traditional vs cloud-native**  
**Hierarchical vs graph-based**

# **By example**

**Upload from frontend**  
**Automation without root**



# **Introduction**

**What is AWS**  
**Cloud vs Hosting**  
**Core security tools**

## **By comparison**

**Monolithic vs distributed**  
**Traditional vs cloud-native**  
**Hierarchical vs graph-based**

## **By example**

**Upload from frontend**  
**Automation without root**

## Monthly costs by service

Daily

Bar

Group by: **API Operation** Service Linked Account Region Instance Type Usage Type Tag Availability Zone Platform Purchase Option Tenancy More

Costs (\$)

# AWS

To see usage data, filter by "Usage Type" or "Usage Type Group" filters with matching units (e.g., hours).

Download CSV

Operation

Region

Region

Region

Operation Total

Total cost (\$)

CreateVolume-Gp2 (\$)

RunInstances (\$)

Daily

Bar

Group By All Operations Service Linked Account Region Instance Type Usage Type Tag Availability Zone Platform Purchase Option Tenancy More

Costs

# Infrastructure as a service

# AWS

CreateVolume-Gp2 RunInstances Storage Standard PublicIP-Out Others

To see usage data, filter by "Usage Type" or "Usage Type Group" filters with matching units (e.g., hours)

Download CSV

Operation Operation Total

Total cost (\$)

CreateVolume-Gp2 (\$)

RunInstances (\$)

# Pay on demand

# Cloud vs Hosting

## ▼ All services

### Compute

- EC2
- Lightsail
- ECR
- ECS
- EKS
- Lambda
- Batch
- Elastic Beanstalk
- Serverless Application Repository

### Storage

- S3
- EFS
- FSx
- S3 Glacier
- Storage Gateway
- AWS Backup

### Database

- RDS
- Amazon ElastiCache
- Amazon Neptune
- Amazon Redshift
- Amazon QLDB
- Amazon DocumentDB

### Migration & Transfer

- AWS Migration Hub
- Application Discovery Service
- Database Migration Service
- Server Migration Service
- AWS Transfer for SFTP
- Snowball
- DataSync

### Networking & Content Delivery

- VPC
- CloudFront
- Route 53
- API Gateway
- Direct Connect

### Developer Tools

- CodeStar
- CodeCommit
- CodeBuild
- CodeDeploy
- CodePipeline
- Cloud9
- X-Ray

### Robotics

- AWS RoboMaker

### Blockchain

- Amazon Managed Blockchain

### Satellite

- Ground Station

### Management & Governance

- AWS CloudFormation
- CloudWatch
- AWS IAM
- CloudTrail
- Config
- OpsWorks
- Service Catalog
- Systems Manager
- Trusted Advisor
- Managed Services
- Control Tower
- AWS License Manager
- AWS Well-Architected Tool
- Personal Health Dashboard
- AWS Chatbot

### Media Services

- Elastic Transcoder
- Kinesis Video Streams
- MediaConnect
- MediaConvert
- MediaLive
- MediaPackage

### Machine Learning

- Amazon SageMaker
- Amazon Comprehend
- AWS DeepLens
- Amazon Lex
- Machine Learning
- Amazon Polly
- Rekognition
- Amazon Transcribe
- Amazon Translate
- Amazon Personalize
- Amazon Forecast
- Amazon Texttract
- AWS DeepRacer

### Analytics

- Athena
- EMR
- CloudSearch
- Amazon Kinesis
- Amazon Redshift
- Amazon EMRFS
- AWS Glue
- AWS Lake Formation
- MSK

### Security, Identity, & Compliance

- IAM
- Resource Access Manager
- Cognito
- Secrets Manager
- GuardDuty
- Inspector
- Amazon Macie
- AWS Single Sign-On
- Certificate Manager
- Key Management Service
- CloudHSM
- Directory Service
- WAF & Shield
- Artifact
- Security Hub

### Mobile

- AWS Amplify
- Mobile Hub
- AWS AppSync
- Device Farm

### AR & VR

- Amazon Sumerian

### Application Integration

- Step Functions
- Amazon EventBridge
- Amazon MQ
- Simple Notification Service
- Simple Queue Service
- SWF

### Customer Engagement

- Amazon Comprehend
- Amazon Rekognition
- Simple Email Service

### Business Applications

- Alexa for Business
- Amazon Chime
- WorkMail

### End User Computing

- WorkSpaces
- AppStream 2.0
- WorkDocs
- WorkLink

### Internet of Things

- IoT Core
- Amazon FreeRTOS
- IoT 1-Click
- IoT Analytics
- IoT Device Defender
- IoT Device Management
- IoT Events
- IoT Greengrass
- IoT SiteWise

Machine learning for every developer and data scientist. [Learn more](#)

## Register for re:Invent

Join us in Las Vegas December 2 – 6 for 2,500+ sessions, bootcamps, hackathons, workshops, and chalk talks. [View session catalog](#)

## EC2 Spot Instances

Run fault-tolerant workloads on Spot Instances and save up to 90% on compute. [Learn more](#)

## Amazon RDS

Set up, operate, and scale your relational database in the cloud. [Learn more](#)

## Have feedback?

Let us know what you think about AWS services and the AWS website. We'll use your feedback to improve our services and website. [Give us feedback](#)



Innovate with provider

Cloud vs Hosting

Thinking model

# Security tools





The background features two faded diagrams. The left diagram is an AWS IAM console screenshot showing a 'Principal' (User), a 'Request' (Action: iam:CreateUser), and 'Authorization' (Policies: Identity-based, Resource-based). The right diagram is an AWS VPC console screenshot showing a VPC with a Public subnet (10.0.0.0/24) containing Web servers and a Private subnet (10.0.1.0/24) containing a Database. It also shows a NAT gateway and an Availability Zone A.

# Complex system

# Security tools

Network, storage, auditing, reaction,  
application level



The background features two faded diagrams. The left diagram is an AWS IAM console screenshot showing a 'Principal' (User), a 'Request' (Action: iam:CreateUser), and an 'Authorization' section with 'Identity-based policies' and 'Resource-based policies'. The right diagram is an AWS VPC console screenshot showing a VPC with a 'Public subnet' (10.0.0.0/24) and a 'Private subnet' (10.0.0.0/24), along with 'Web servers' and a 'Database'.

**Complex system**

# Security tools

**Network, storage, auditing, reaction,  
application level**

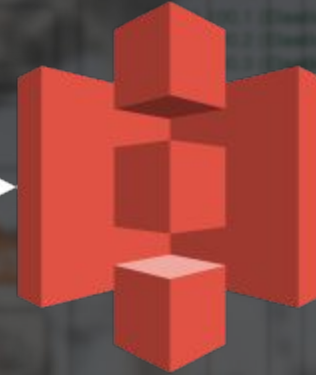
Principal



Action



Resource



**Who**

Identified  
by  
HTTPS  
signing

**What**

Differs by  
AWS service

**Where**

Uniquely  
identified by  
ARN  
(URL-like name)



# **Introduction**

**What is AWS**  
**Cloud vs Hosting**  
**Core security tools**

## **By comparison**

**Monolithic vs distributed**  
**Traditional vs cloud-native**  
**Hierarchical vs graph-based**

## **By example**

**Upload from frontend**  
**Automation without root**



# **Introduction**

**What is AWS**  
**Cloud vs Hosting**  
**Core security tools**

## **By comparison**

**Monolithic vs distributed**  
**Traditional vs cloud-native**  
**Hierarchical vs graph-based**

## **By example**

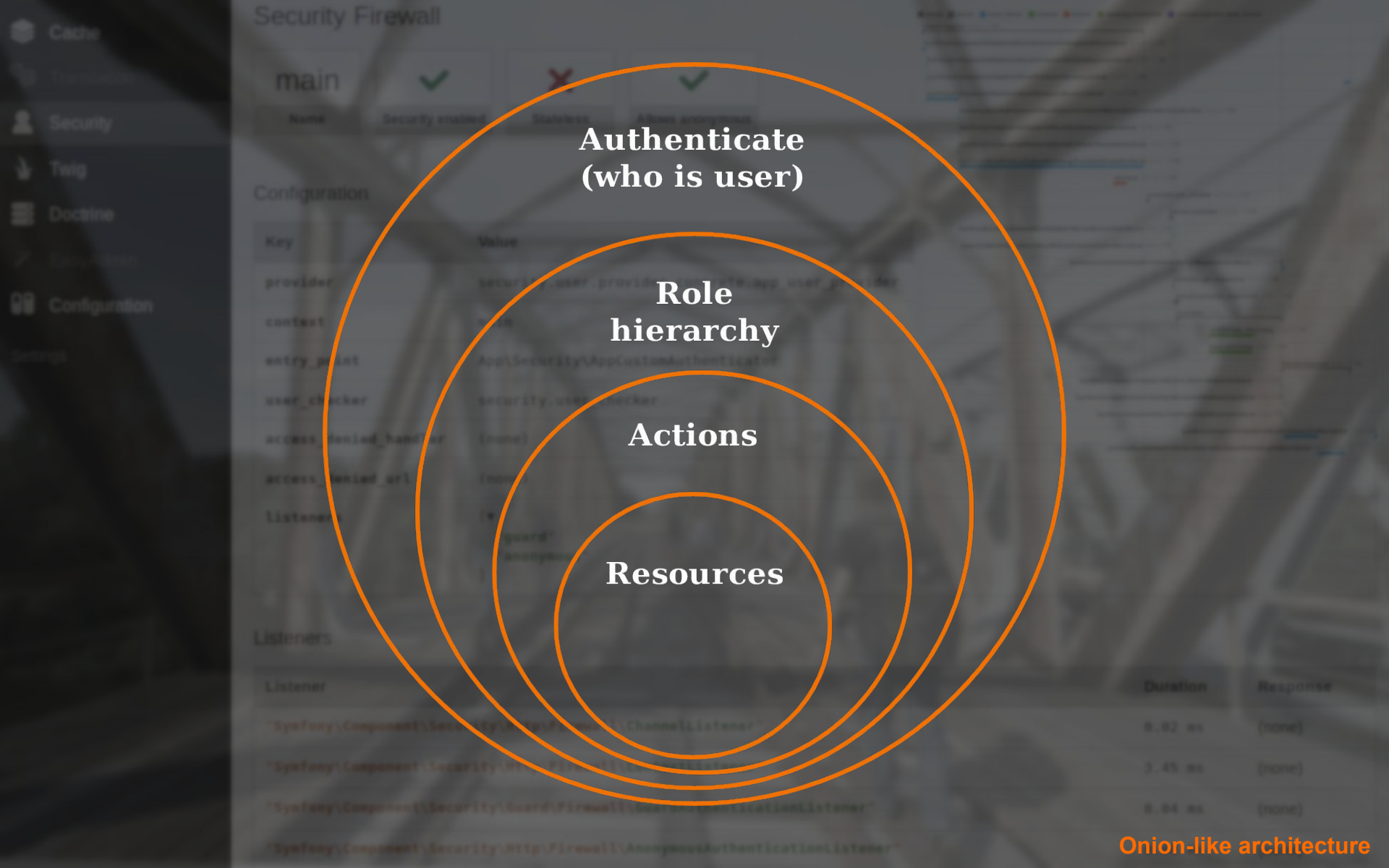
**Upload from frontend**  
**Automation without root**



# **Monolithic vs Distributed**

# Monolithic





**Authenticate  
(who is user)**

**Role  
hierarchy**

**Actions**

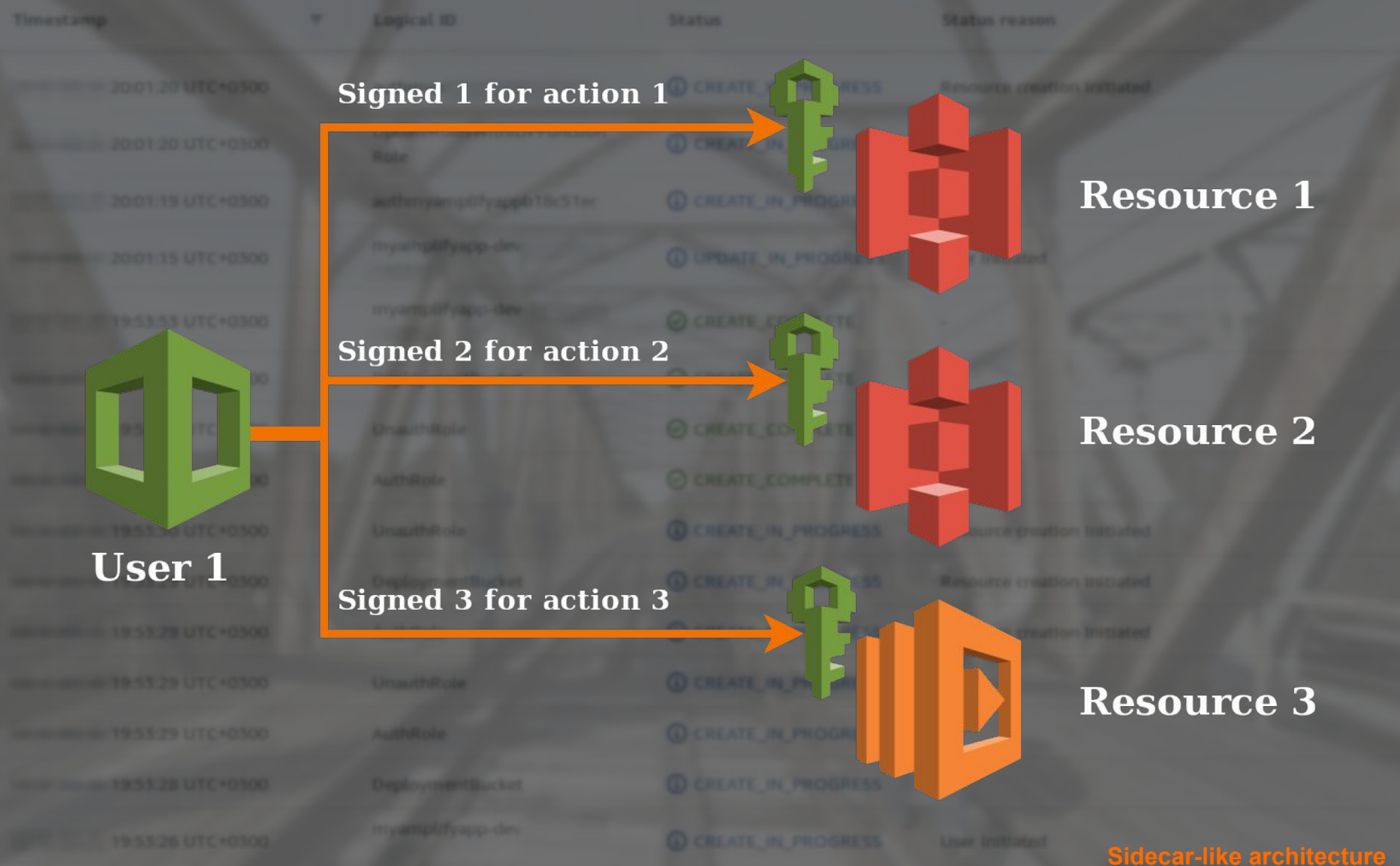
**Resources**

**Onion-like architecture**

# Distributed

Timestamp	Logical ID	Status	Status reason
20:01:20 UTC+0300	authmya	CREATE_IN_PROGRESS	Resource creation initiated
20:01:20 UTC+0300	UpdateRolesWithIDPFunction Role	CREATE_IN_PROGRESS	-
20:01:19 UTC+0300	authmyamplifyappb18c51er	CREATE_IN_PROGRESS	-
20:01:15 UTC+0300	myamplifyapp-dev	UPDATE_IN_PROGRESS	User initiated
19:53:53 UTC+0300	myamplifyapp-dev	CREATE_COMPLETE	-
19:53:50	DeploymentBucket	CREATE_COMPLETE	-
19:53:48 UTC+0300	UnauthRole	CREATE_IN_PROGRESS	-
19:53:47	AuthRole	CREATE_IN_PROGRESS	-
19:53:30 UTC+0300	UnauthRole	CREATE_IN_PROGRESS	Resource creation initiated
19:53:29 UTC+0300	DeploymentBucket	CREATE_IN_PROGRESS	Resource creation initiated
19:53:29 UTC+0300	AuthRole	CREATE_IN_PROGRESS	Resource creation initiated
19:53:29 UTC+0300	UnauthRole	CREATE_IN_PROGRESS	-
19:53:29 UTC+0300	AuthRole	CREATE_IN_PROGRESS	-
19:53:28 UTC+0300	DeploymentBucket	CREATE_IN_PROGRESS	-
19:53:26 UTC+0300	myamplifyapp-dev	CREATE_IN_PROGRESS	User initiated



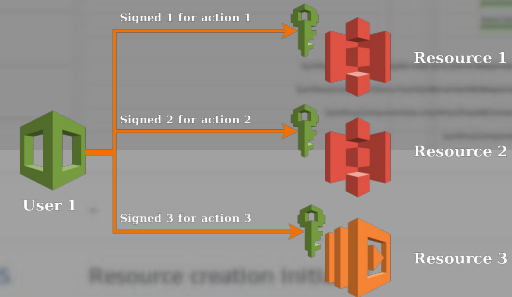
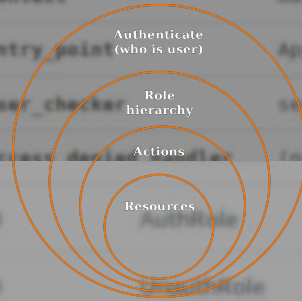




# Monolithic

## vs

# Distributed





# **Traditional vs Cloud-native**

Consent

Data Protection Officer

Email Marketing

Encryption

Fines / Penalties

Personal Data

Privacy by Design

Privacy Impact Assessment

Processing

Records of Processing Activities

Right of Access

Right to be Forgotten

Right to be Informed

Third Countries

## GDPR Encryption

# Traditional

Companies can reduce the probability of a data breach and thus reduce the risk of fines in the future. They choose to use encryption of personal data. The processing of personal data is secure, as it is not possible to access the data without the correct key. However, where cyber-attacks are frequent, it is necessary to take additional measures to protect the data. In particular, a company should ensure that the data is encrypted and that the encryption key is stored securely. The company should also ensure that the data is encrypted and that the encryption key is stored securely. The company should also ensure that the data is encrypted and that the encryption key is stored securely.

In general, encryption refers to the procedure that converts clear text into a hashed code using a key, where the outgoing information only becomes readable again by using the correct key. This minimises the risk of an incident during data processing, as encrypted contents are basically unreadable for third parties who do not have the correct key. Encryption is the best way to protect data during transfer and one way to secure stored personal data. It also reduces the risk of abuse within a company, as access is limited only to authorised people with the right key.

The Regulation also recognizes these risks when processing personal data and places the responsibility on the controller and the processor in Art. 32(1) of the General Data Protection Regulation to implement appropriate technical and organisational measures to secure personal data. The GDPR deliberately does not define which specific technical and organisational measures are considered suitable in each case, in order to accommodate individual factors.

Consent

Data Protection Officer

Email Marketing

Encryption

Fines / Penalties

Personal Data

Privacy by Design

Privacy Impact Assessment

Processing

Records of Processing Activities

Right of Access

Right to be Forgotten

Right to be Informed

Third Countries

## GDPR Encryption

Companies can reduce the probability of a data breach in the future, if they chose to use encryption. Encryption is a risk naturally associated with a certain level of complexity, which is nearly unavoidable for companies. Encryption plays an ever-larger role in IT security and data protection for companies.

In general, encryption refers to the process of converting data into a key, where the outgoing information is encrypted. Encryption minimises the risk of an incident during data processing, as encrypted contents are typically unreadable for third parties who do not have the correct key. Encryption is the best way to protect data during transfer and one way to secure stored personal data. It also reduces the risk of abuse within a company, as access is limited only to authorised people with the right key.

The Regulation also recognizes these risks when processing personal data and places the responsibility on the controller and the processor in Art. 32(1) of the General Data Protection Regulation to implement appropriate technical and organisational measures to secure personal data. The GDPR deliberately does not define which specific technical and organisational measures are considered suitable in each case, in order to allow companies to choose the appropriate measures based on the state of the art and the nature, scope, context and purposes of the processing.



aurelijusb reviewed on

[View changes](#)

/Services

```
12 + public function testFormatEur()  
13 + {  
14 +     $moneyFormatter = new MoneyFormatter(new NumberFormatter());  
15 +     // $numberFormatter = $this->createMock(NumberFormatter::class);
```



aurelijusb on May 26 Author Owner

...



aurelijusb commented on

Author

Owner

...



Enforced and validated by humans

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

Visual editor

JSON

[Import managed policy](#)

# Cloud-native

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "DenyIncorrectEncryptionHeader",  
6       "Effect": "Deny",  
7       "Action": "s3:PutObject",  
8       "Resource": "*",  
9       "Condition": {  
10        "StringNotEquals": {  
11          "s3:x-amz-server-side-encryption": "AES256"  
12        }  
13      }  
14    },  
15    {  
16      "Sid": "DenyUnencryptedObjectUploads",  
17      "Effect": "Deny",  
18      "Action": "s3:PutObject",  
19      "Resource": "*",  
20      "Condition": {  
21        "Null": {  
22          "s3:x-amz-server-side-encryption": true  
23        }  
24      }  
25    }  
26  ]  
27 }
```

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

Visual editorJSON

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Sid": "DenyIncorrectEncryptionHeader",
6       "Effect": "Deny",
7       "Action": "s3:PutObject",
8       "Resource": "*",
9       "Condition": {
10        "StringNotEquals": {
11          "s3:x-amz-server-side-encryption": "AES256"
12        }
13      }
14    },
15    {
16      "Sid": "DenyUnEncryptedObjectUploads",
17      "Effect": "Deny",
18      "Action": "s3:PutObject",
19      "Resource": "*",
20      "Condition": {
21        "Null": {
22          "s3:x-amz-server-side-encryption": true
23        }
24      }
25    }
26  ]
27 }
28
29
30
```

## Welcome to CloudTrail

With CloudTrail, you can view events for your AWS account. Create a trail to retain a record of these events. With a trail, you can also create event metrics, trigger alerts, and create event workflows. You can also create a trail for an organization by logging in with the master account for AWS Organizations. [Learn more](#)

Create trail

## Recent events

These are the most recent events recorded by CloudTrail. To view all events for the last 90 days, go to Event history.

	Event time	User name	Event name
▶	08:03:41 PM	root	ConsoleLogin
▶	08:13:55 PM	amplify- -cli	PutBucketWebsite
▶	08:13:54 PM	amplify- -cli	PutBucketCors
▶	08:13:54 PM	amplify- -cli	CreateBucket
▶	08:13:45 PM	amplify- -cli	UpdateStack

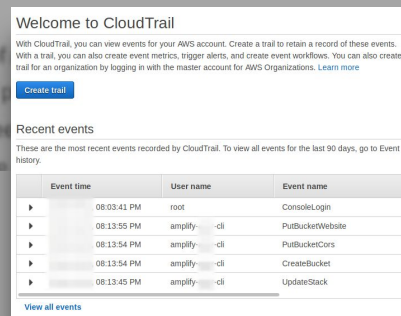
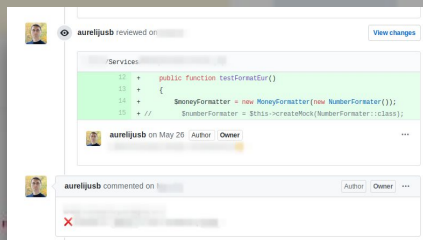
[View all events](#)

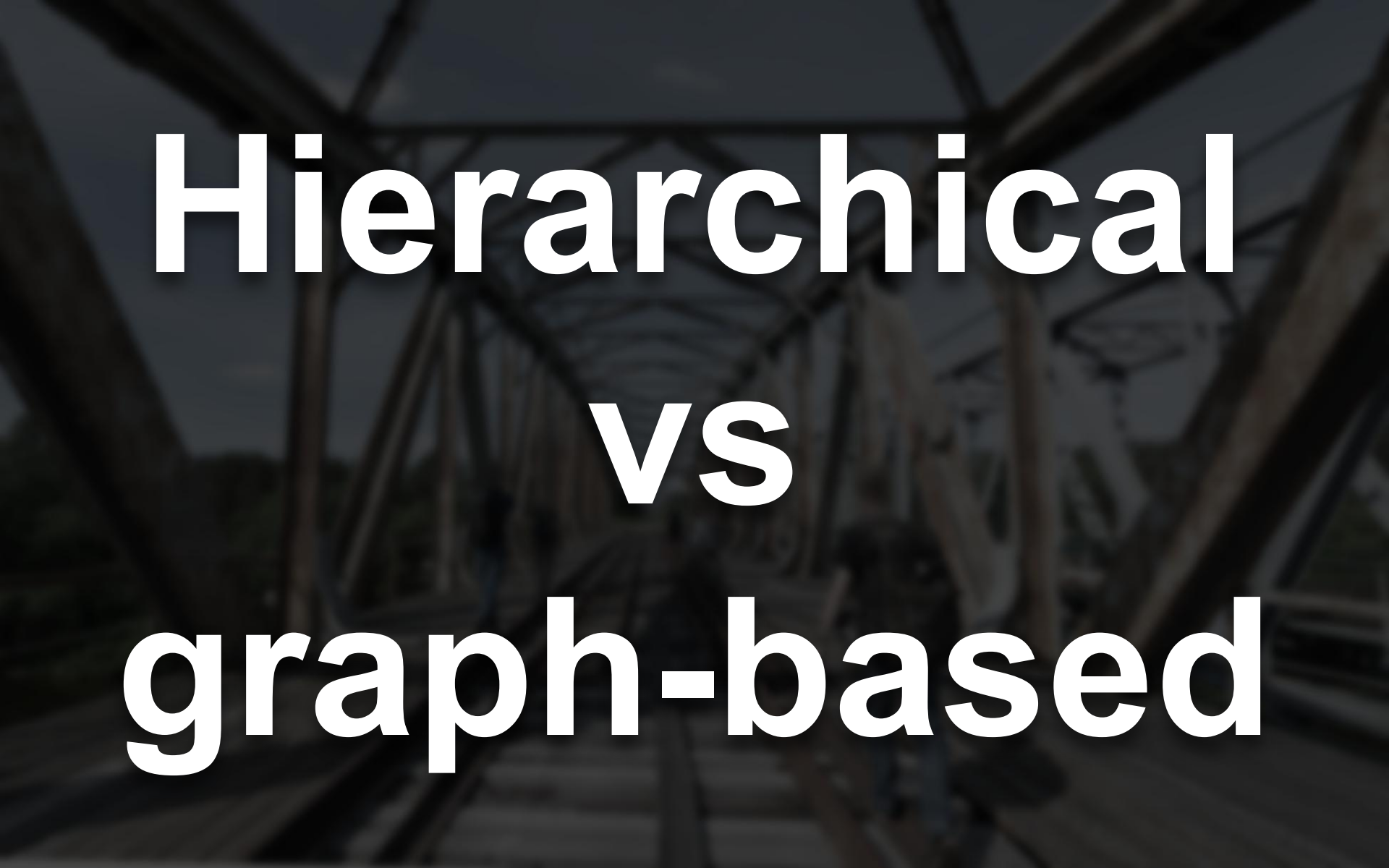


# Traditional

vs

# Cloud-native





# **Hierarchical vs graph-based**



# Hierarchical

## Synopsis

```
assume-role
--role-arn <value>
--role-session-name <value>
[--policy-arns <value>]
[--policy <value>]
[--duration-seconds <value>]
[--external-id <value>]
[--profile-name <value>]
[--key-id <value>]
[--cli-binary-name <value>]
[--generate-cli-skeleton <value>]
```

# Graph-based

## Options

--role-arn (string)

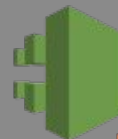
The Amazon Resource Name (ARN) of the role to assume.

--role-session-name (string)

## Synopsis

```
assume-role
--role-arn <value>
--role-session-name <value>
[--policy-arns <value>]
[--policy <value>]
[--duration-seconds <value>]
[--expiration <value>]
[--profile <value>]
[--output <value>]
[--output-json <value>]
[--generate-cli-skeleton <value>]
```

Granular  
auditing



Expiration



**assume  
role**



## Options

--role-arn (string)

The Amazon Resource Name (ARN) of the role to assume.

--role-session-name (string)

# Hierarchical vs graph-based

```
assume-role
--role-arn <value>
--role-session-name <value>
[--policy-arns <value>]
[--policy <value>]
[--external-id <value>]
[--token-code <value>]
[--cli-input-json <value>]
[--generate-cli-skeleton <value>]
```



# **Introduction**

**What is AWS**  
**Cloud vs Hosting**  
**Core security tools**

## **By comparison**

**Monolithic vs distributed**  
**Traditional vs cloud-native**  
**Hierarchical vs graph-based**

## **By example**

**Upload from frontend**  
**Automation without root**

# **Introduction**

**What is AWS**  
**Cloud vs Hosting**  
**Core security tools**

# **By comparison**

**Monolithic vs distributed**  
**Traditional vs cloud-native**  
**Hierarchical vs graph-based**

# **By example**

**Upload from frontend**  
**Automation without root**

```
use Aws\S3\S3Client;  
use Aws\Exception\AwsException;
```

## Sample Code

```
$s3Client = new Aws\S3\S3Client([  
    'profile' => 'default',  
    'region' => 'us-east-2',  
    'version' => '2006-03-01',  
]);  
  
$cmd = $s3Client->getCommand('GetObject', [  
    'Bucket' => 'my-bucket',  
    'Key' => 'testKey'  
]);  
  
$request = $s3Client->createPresignedRequest($cmd, '+20 minutes');
```



# Upload from frontend

## Creating a Pre-Signed URL

You can create pre-signed URLs for any Amazon S3 operation using the `getCommand()` method for creating a command object, and then calling the `createPresignedRequest()` method with the command object. When using pre-signed URLs, the request is the same as the original request, and the same headers as the returned request.

## Sample Code

```
//Creating a presigned URL  
$cmd = $s3Client->getCommand('GetObject', [  
    'Bucket' => 'my-bucket',  
    'Key' => 'testKey'  
]);  
  
$request = $s3Client->createPresignedRequest($cmd, '+20 minutes');
```

```
// Get the actual presigned-url  
$presignedUrl = (string)$request->getUri();
```



```
use Aws\S3\S3Client;  
use Aws\Exception\AwsException;
```

Sample Code



Frontend

HTTP POST



Backend

```
$s3Client = new Aws\S3\S3Client([  
    'profile' => 'default',  
    'region' => 'us-east-2',  
    'version' => '2006-03-01',  
]);  
  
$cmd = $s3Client->getCommand('GetObject', [  
    'Bucket' => 'my-bucket',  
    'key' => 'testKey'  
]);  
  
$request = $s3Client->createPresignedRequest($cmd, '+20 minutes');
```



Creating a Presigned URL

You can create pre-signed URLs for any Amazon S3 operation using the `getCommand` method for creating a command object, and then calling the `createPresignedRequest()` method with the command. When ultimately sending the request, be sure to use the same method and the same headers as the returned request.

Sample Code

```
//Creating a Presigned URL  
$cmd = $s3Client->getCommand('GetObject', [  
    'Bucket' => 'my-bucket',  
    'key' => 'testKey'  
]);  
  
$request = $s3Client->createPresignedRequest($cmd, '+20 minutes');
```

```
// Get the actual presigned-url  
$presignedUrl = (string)$request->getUri();
```



```
use Aws\S3\S3Client;  
use Aws\Exception\AwsException;
```

## Sample Code

Frontend



HTTP POST

```
$s3Client = new Aws\S3\S3Client([  
    'profile' => 'default',  
    'region' => 'us-east-2',  
    'version' => '2006-03-01',  
]);  
  
$cmd = $s3Client->getCommand('GetObject', [  
    'Bucket' => 'my-bucket',  
    'Key' => 'testKey'  
]);  
  
$request = $s3Client->createPresignedRequest($cmd, '+20 minutes');
```

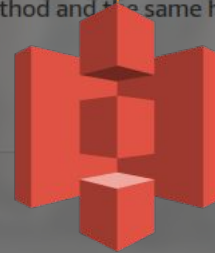
Backend



Signed URL



HTTP PUT



## Creating a Pre-Signed URL

You can create pre-signed URLs for any Amazon S3 operation using the `getCommand` method for creating a command object, and then calling the `createPresignedRequest()` method with the command. When ultimately sending the request, be sure to use the same method and the same headers as the returned request.

## Sample Code


```
//Creating a presigned URL  
$cmd = $s3Client->getCommand('GetObject', [  
    'Bucket' => 'my-bucket',  
    'Key' => 'testKey'  
]);  
  
$request = $s3Client->createPresignedRequest($cmd, '+20 minutes');
```

```
// Get the actual presigned-url  
$presignedUrl = (string)$request->getUri();  
https://gist.github.com/aurelijusb/527c07e0f47b6dcbd1bdca27d265ac72
```

# Automation without root

SignalResource	Sends a signal to the specified resource with a success or failure status.	Write	stack*		
StopStackSetOperation	Stops an in-progress operation on a stack set and its associated stack instances.	Write	stackset*		
UpdateStack	Updates a stack as specified in the template.	Write	stack*		
				cloudformation:ResourceTypes	
				cloudformation:RoleArn	
				cloudformation:StackPolicyUrl	
				cloudformation:TemplateUrl	
				aws:RequestTag/\${TagKey}	
				aws:TagKeys	
UpdateStackInstances	Updates the parameter values for stack instances for the specified accounts, within the specified region.	Write	stackset*		
UpdateStackSet	Updates a stack set specified in the template.	Write	stackset*		
				cloudformation:RoleArn	
				cloudformation:TemplateUrl	
				aws:RequestTag/\${TagKey}	
				aws:TagKeys	
UpdateTerminationProtection	Updates termination protection for the specified stack.	Write	stack*		
ValidateTemplate	Validates a specified template.	Write			



StopStackResources	Sends a signal to the specified resource with a success or failure status.	Write	stack*			
StopStackSetOperation	Stops an in-progress operation on a stack set and its associated stack instances.	Write	stackset*			
UpdateStack	Updates a stack as specified in the template.	Write	stack*			
	<b>PolicyName:</b> AllowToDeployNewVersion <b>PolicyDocument:</b> Version: "2012-10-17" Statement: - Effect: "Allow" Action: - "cloudformation:DescribeStacks" - "cloudformation:DescribeStackEvents" - "cloudformation:DescribeStackResources" - "cloudformation:CreateChangeSet" - "cloudformation:DescribeChangeSet" - "cloudformation>DeleteChangeSet" - "cloudformation:ExecuteChangeSet" - "cloudformation>ListChangeSets" - "cloudformation:CancelUpdateStack" - "cloudformation:ContinueUpdateRollback" - "cloudformation>DeleteChangeSet" - "cloudformation:UpdateStack" - "cloudformation>ListStackResources"		cloudformation:ResourceTypes cloudformation:RoleArn cloudformation:StackPolicyUrl cloudformation:TemplateUrl aws:RequestTag/\${TagKey} aws:TagKeys	<div>Principal</div> <div>Action</div> <div>Resource</div> 		
UpdateStackInstances	Updates the parameter values for stack instances of the specified stack.	Write	stackset*	<b>Who</b> Identified by HTTPS signing	<b>What</b> Differs by AWS service	<b>Where</b> Uniquely identified by ARN (URL-like name)
UpdateStackSet	Updates the parameter values for stack instances of the specified stack.	Write	stackset*		cloudformation:RoleArn cloudformation:TemplateUrl aws:RequestTag/\${TagKey} aws:TagKeys	
UpdateTerminationProtection	Updates termination protection for the specified stack.	Write	stack*			
ValidateTemplate	Validates a specified template.	Write				

# **Introduction**

**What is AWS**  
**Cloud vs Hosting**  
**Core security tools**

# **By comparison**

**Monolithic vs distributed**  
**Traditional vs cloud-native**  
**Hierarchical vs graph-based**

# **By example**

**Upload from frontend**  
**Automation without root**

# **Introduction**

**What is AWS**  
**Cloud vs Hosting**  
**Core security tools**

# **By comparison**

**Monolithic vs distributed**  
**Traditional vs cloud-native**  
**Hierarchical vs graph-based**

# **By example**

**Upload from frontend**  
**Automation without root**

# Conclusion







**Problems**  
**harder**  
**Perspective**  
**wider**

# References and further reading

- AWS Best practices:  
<https://aws.amazon.com/architecture/well-architected/>
- Summaries as illustrations:  
<https://www.awsgeek.com/>
- Community managed resources:  
<https://github.com/open-guides/og-aws#security-and-iam>
- Thinking about the Cloud: from application perspective:  
<http://shop.oreilly.com/product/0636920072768.do>
- Thinking about the Cloud: from infrastructure tools perspective:  
<http://shop.oreilly.com/product/0636920075837.do>



# How AWS

**Thank you**  
**Discussion?**

Aurelijus Banelis

