

# Developer Operations

---

## Security Considerations

# Security in Cloud / DevOps

---

- Security often considered an **inhibitor** to cloud deployment
  - Isolation failure (*nosy neighbour* problem)
  - Credential leakage
  - Legal & regulatory issues
  - Loss of control
  - Data loss
  - Transition of legacy applications / models
  - Punctured perimeter (so network security devices like firewalls and intrusion detection lose effectiveness)

# Security in Cloud / DevOps

---

- But security can also be an **enabler**
  - Security more easily outsourced to specialists
    - Especially relevant to PaaS and SaaS
  - New deployments allow security to be built in from the start
  - Specialist cloud security services (“security as a service”) can improve security – e.g.:
    - Encryption accelerators
    - Secure random number generators
    - Secure containers
    - Security monitoring
    - Online penetration testing

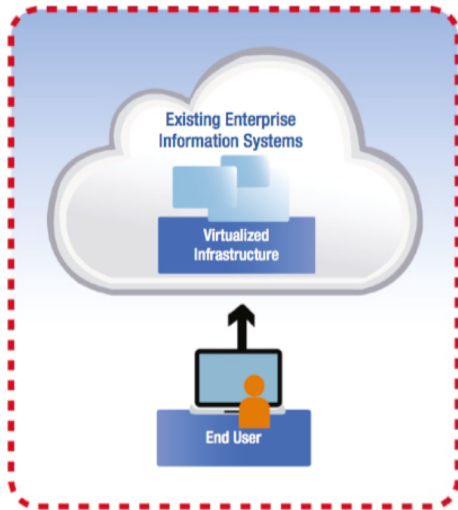
# Trends impacting cloud security

---

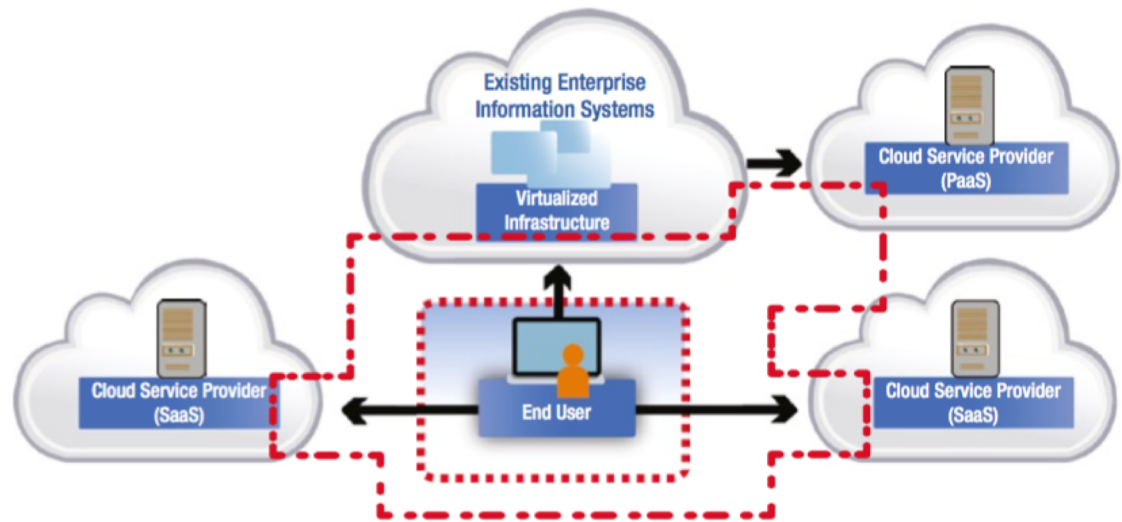
- Virtualisation and cloud architectures
  - Multi-tenancy: hardware shared by different lines of business and even different businesses
- Increased attack sophistication
  - No longer just OS and system/application software
    - BIOS, firmware, hypervisor
  - Targeted attacks
  - More covert
- Legal and regulatory compliance
  - Increased regulation and audit of personally identifiable data, financial data, etc.



# Perimeter evolution



Traditional perimeter



Cloud security perimeter

---

# AWS Security

# AWS Shared Responsibility Model

---

Responsible for  
security  
*'in'* the Cloud

AWS services ***you*** utilize to run  
your workloads (EC2 instances, S3  
buckets, etc.)

Responsible for  
security  
*'of'* the Cloud

Policies and mechanisms ***AWS***  
uses to protect the cloud itself












# AWS Shared Responsibility Model



# AWS Shared Responsibility Model



# AWS Principal Services for Security

Areas	Key Services				
Data protection					
	Elastic Load Balancing	Amazon EBS	Amazon S3	Amazon RDS	AWS Key Management Service (KMS)
Privilege management					
	AWS IAM	MFA token			
Infrastructure protection					
	Amazon VPC				
Detective controls					
	AWS CloudTrail	AWS Config	Amazon CloudWatch		

# AWS Data Protection

---

- Many AWS services include encryption capabilities to protect data in transit and at rest, such as:
  - Elastic Load Balancing
  - Elastic Block Store (EBS),
  - Simple Storage Service (S3)
  - Relational Database Service (RDS).
- AWS Key Management Service (KMS)
  - Secure key storage, creation, rotation, usage
- Cloud HSM
  - Cloud-based hardware security module – managed by customer

# AWS Privilege Management

---

- Identity & Access Management (IAM)
  - Users & Groups
  - Roles
  - Permissions (policies)
  - Multi-factor authentication
  - API keys



AWS IAM



# IAM Authentication: Management Console

## ■ AWS Management Console

- User Name and Password
- MFA



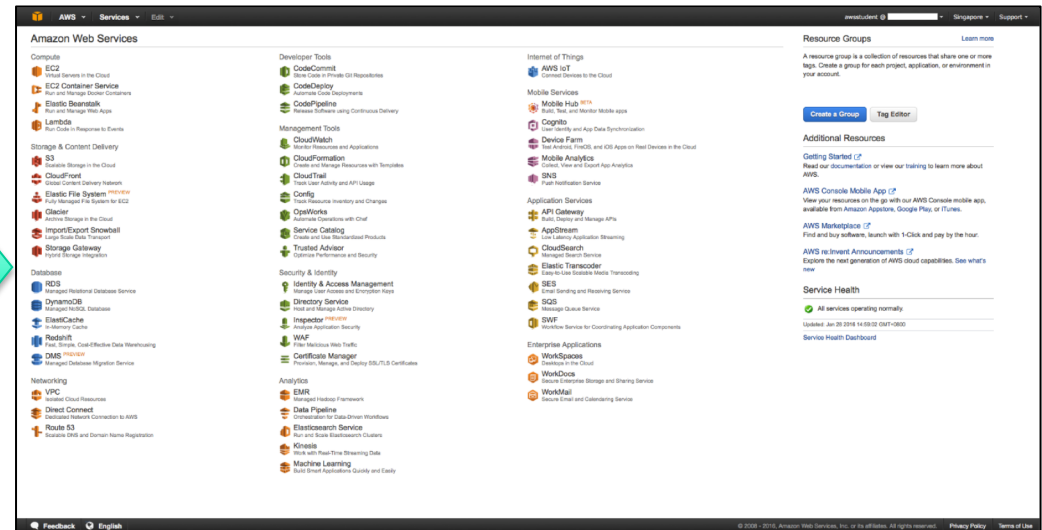
**Account:**

**User Name:**

**Password:**

MFA users, enter your code on the next screen.

[Sign In](#)



# IAM Authentication: CLI / API

- **AWS CLI or SDK API**

- Access Key and Secret Key



IAM User

**Access Key ID:** AKIAIOSFODNN7EXAMPLE  
**Secret Access Key:** wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY

## AWS CLI

```
~$ aws configure
AWS Access Key ID [*****O22A]:
AWS Secret Access Key [*****4m8i]:
Default region name [ap-southeast-1]:
Default output format [json]:
```

## AWS SDK & API



Java



Python



.NET

# IAM Authorisation

---

## Policies:

- ✓ Are JSON documents to describe permissions.
- ✓ Are assigned to users, groups or roles.



**IAM  
User**



**IAM  
Group**



**IAM  
Roles**

# IAM Authorisation: Roles

- IAM roles are a secure way to grant permissions to specific entities
  - e.g. application code running on an EC2 instance that needs to perform some actions on AWS resources

aws IAM Management Console

Services Resource Groups jmcgibney @ witdev Global Support

## Create role

1 Trust — 2 Permissions — 3 Review

### Attach permissions policies

Choose one or more policies to attach to your new role.

Create policy Refresh

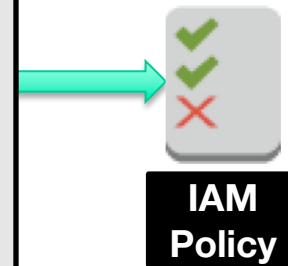
Filter: Policy type s3 Showing 4 results

	Policy name	Attachments	Description
<input type="checkbox"/>	AmazonDMSRedshiftS3Role	0	Provides access to manage S3 settings for Redshift endpoint...
<input type="checkbox"/>	AmazonS3FullAccess	1	Provides full access to all buckets via the AWS Management...
<input checked="" type="checkbox"/>	AmazonS3ReadOnlyAccess	0	Provides read only access to all buckets via the AWS Manag...
<input type="checkbox"/>	AmazonS3ReadOnlyAccess	0	Provides read only access to all buckets via the AWS Manag...

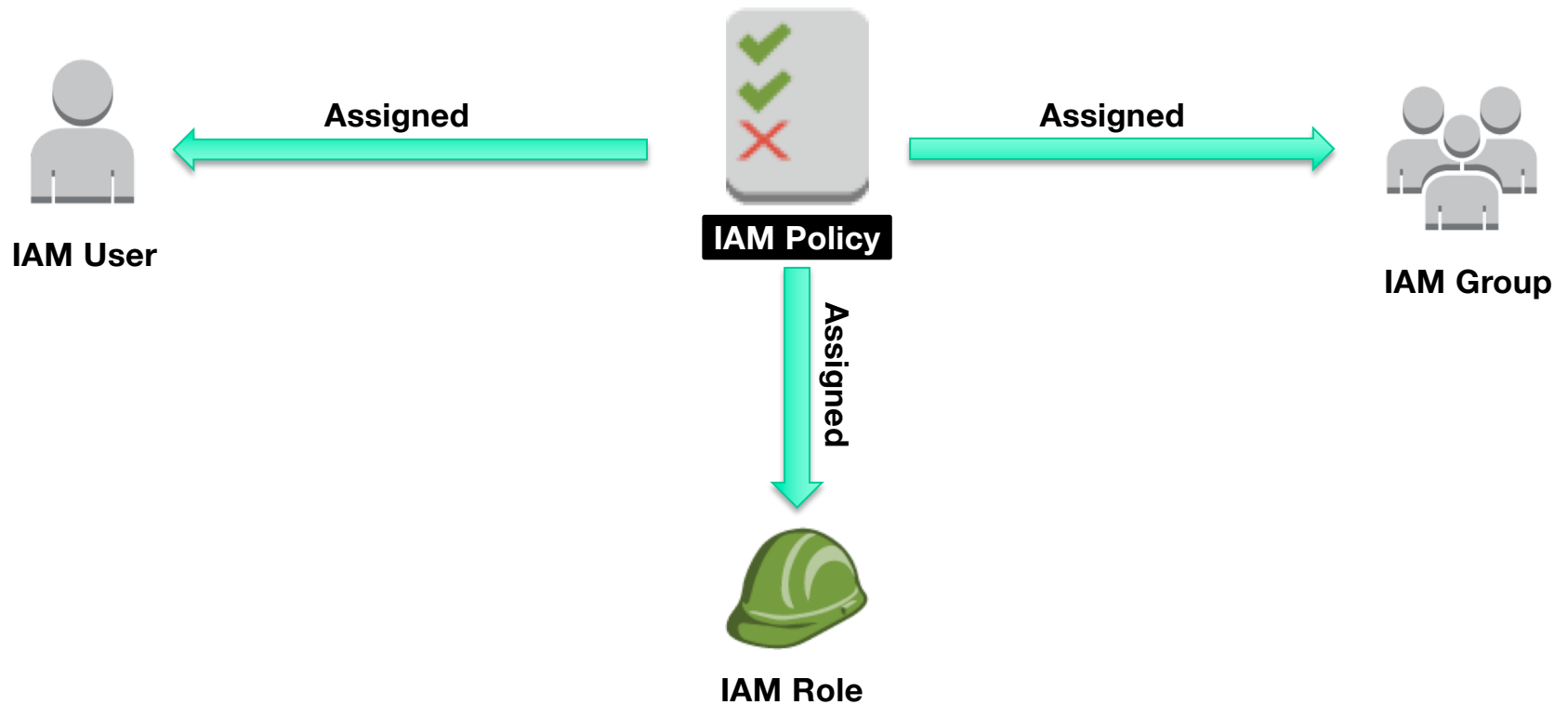
\* Required Cancel Previous Next: Review

# IAM Authorisation: Policy elements

```
{
  "version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1453690971587",
      "Action": [
        "ec2:Describe*",
        "ec2:StartInstances",
        "ec2:StopInstances"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "54.64.34.65/32"
        }
      }
    },
    {
      "Sid": "Stmt1453690998327",
      "Action": [
        "s3:GetObject*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::example_bucket/*"
    }
  ]
}
```



# IAM Authorisation: Policy assignment

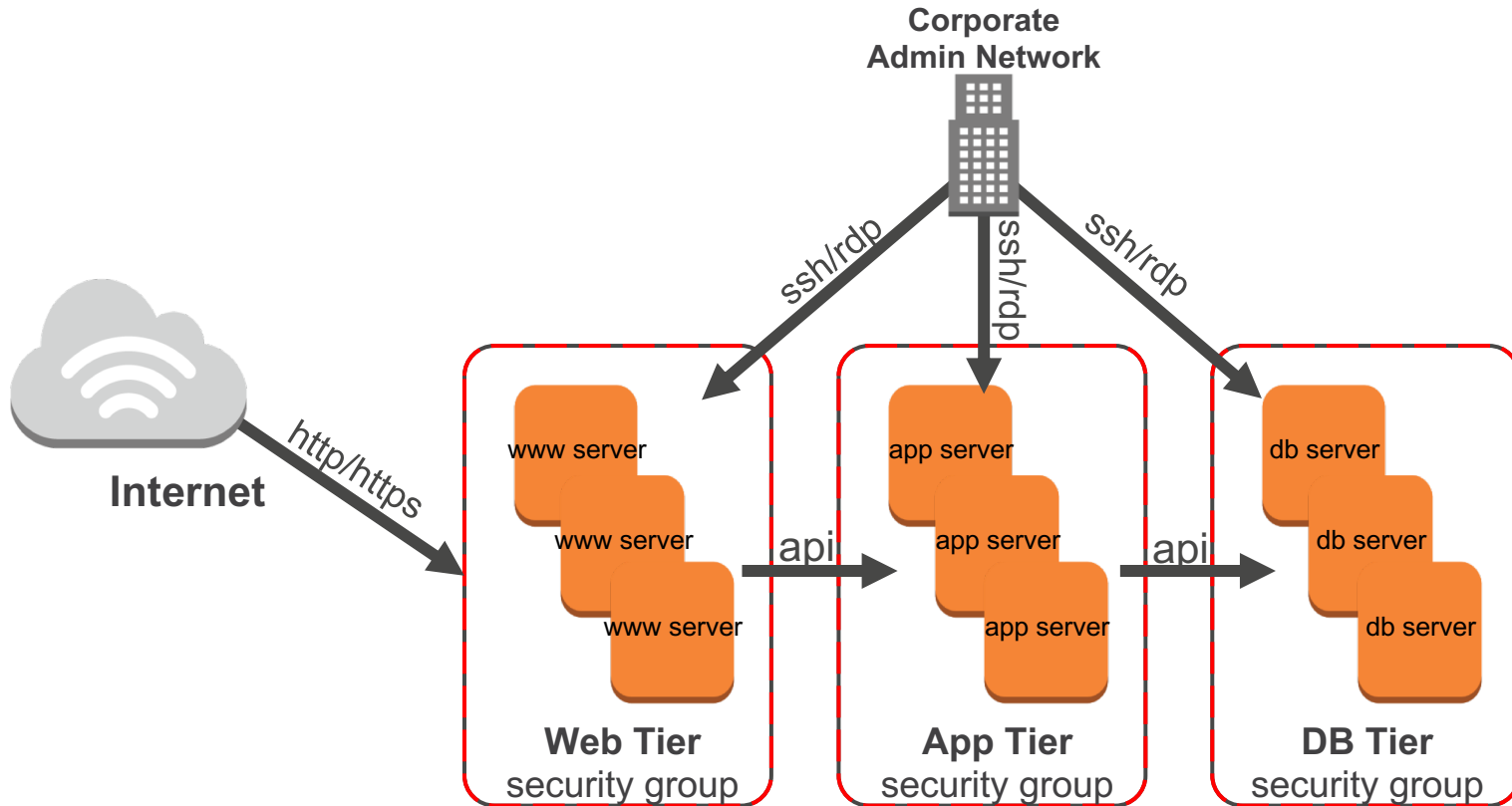


# AWS Infrastructure protection

---

- Amazon Virtual Private Cloud (VPC) lets you provision a private, isolated section of the AWS cloud where you can launch AWS resources in a virtual network
- Public and private subnets
- NAT
- Security Groups
  - Control traffic to/from EC2 instances and RDS databases
- Network Access Control Lists (NACLs)
  - Control traffic to/from subnets
- VPN connections – for connecting to other networks, e.g. customer on-premises network

# AWS Infrastructure protection



*(all other ports are blocked)*



# Security Group configuration

### Edit inbound rules

HTTP	TCP	80	Custom IP	193.1.34.15/	
HTTP	TCP	80	Custom IP	193.1.34.0/2	
HTTP	TCP	80	Custom IP	176.61.0.0/1	
HTTP	TCP	80	Custom IP	46.7.0.0/16	
Custom TCP Rule	TCP	8080	Custom IP	193.1.184.2/	
Custom TCP Rule	TCP	8080	Custom IP	109.76.0.0/1	
Custom TCP Rule	TCP	8080	Custom IP	89.100.0.0/1	
Custom TCP Rule	TCP	8080	Custom IP	193.1.184.0/	
Custom TCP Rule	TCP	8080	Custom IP	86.40.0.0/13	
Custom TCP Rule	TCP	8080	Custom IP	79.97.0.0/16	
Custom TCP Rule	TCP	8080	Custom IP	188.141.0.0/	
Custom TCP Rule	TCP	8080	Custom IP	178.73.195.1	
Custom TCP Rule	TCP	8080	Custom IP	95.83.192.0/	
Custom TCP Rule	TCP	8080	Custom IP	109.125.0.0/	
Custom TCP Rule	TCP	8080	Custom IP	193.1.184.2/	
Custom TCP Rule	TCP	8080	Custom IP	193.1.34.15/	
Custom TCP Rule	TCP	8080	Custom IP	193.1.34.0/2	
Custom TCP Rule	TCP	8080	Custom IP	176.61.0.0/1	
Custom TCP Rule	TCP	8080	Custom IP	46.7.0.0/16	
SSH	TCP	22	Anywhere	0.0.0.0/0	

Add Rule

CancelSave

# AWS Detective Controls

---

- **AWS CloudTrail**
  - records AWS API calls
- **AWS Config**
  - provides a detailed inventory of AWS resources and configuration
- **Amazon CloudWatch**
  - monitoring service for AWS resources

# Some other AWS Security Features

---

- EC2 authentication
  - Key pair for Linux instances; username & password for Windows
- Web application firewall (WAF)
  - Monitors and filters HTTP/HTTPS requests to protect web apps
- Inspector
  - Application behaviour monitoring (installed on instances)
- Certificate Manager
  - TLS certificates deployment, management, renewal
- AWS Shield
  - Protection against DDoS (distributed denial of service)
- Trusted Advisor
  - Dashboard for monitoring AWS resources; includes security

# Web Application Firewall

**Set up a web access control list (web ACL)**

[Concepts overview](#)

[Step 1: Name web ACL](#)

**Step 2: Create conditions**

[Step 3: Create rules](#)

[Step 4: Choose AWS resource](#)

[Step 5: Review and create](#)

### Create conditions

Conditions specify the filters that you want to use to allow or block requests that are forwarded to AWS resources such as Amazon CloudFront distributions.

**Cross-site scripting match conditions**

Name	Create condition
You don't have any cross-site scripting match conditions. Choose <b>Create XSS match condition</b> to get started.	

A cross-site scripting match condition specifies the parts of a web request (such as a User-Agent header) that you want AWS WAF to inspect for cross-site scripting threats. [Learn more](#)

**IP match conditions**

Name	Create condition
You don't have any IP match conditions. Choose <b>Create IP match condition</b> to get started.	

### Concepts overview

**Web ACL example**  
if requests match

**Rule 1, Bad User-Agents, then block**

**IP match condition**  
Suspicious IPs

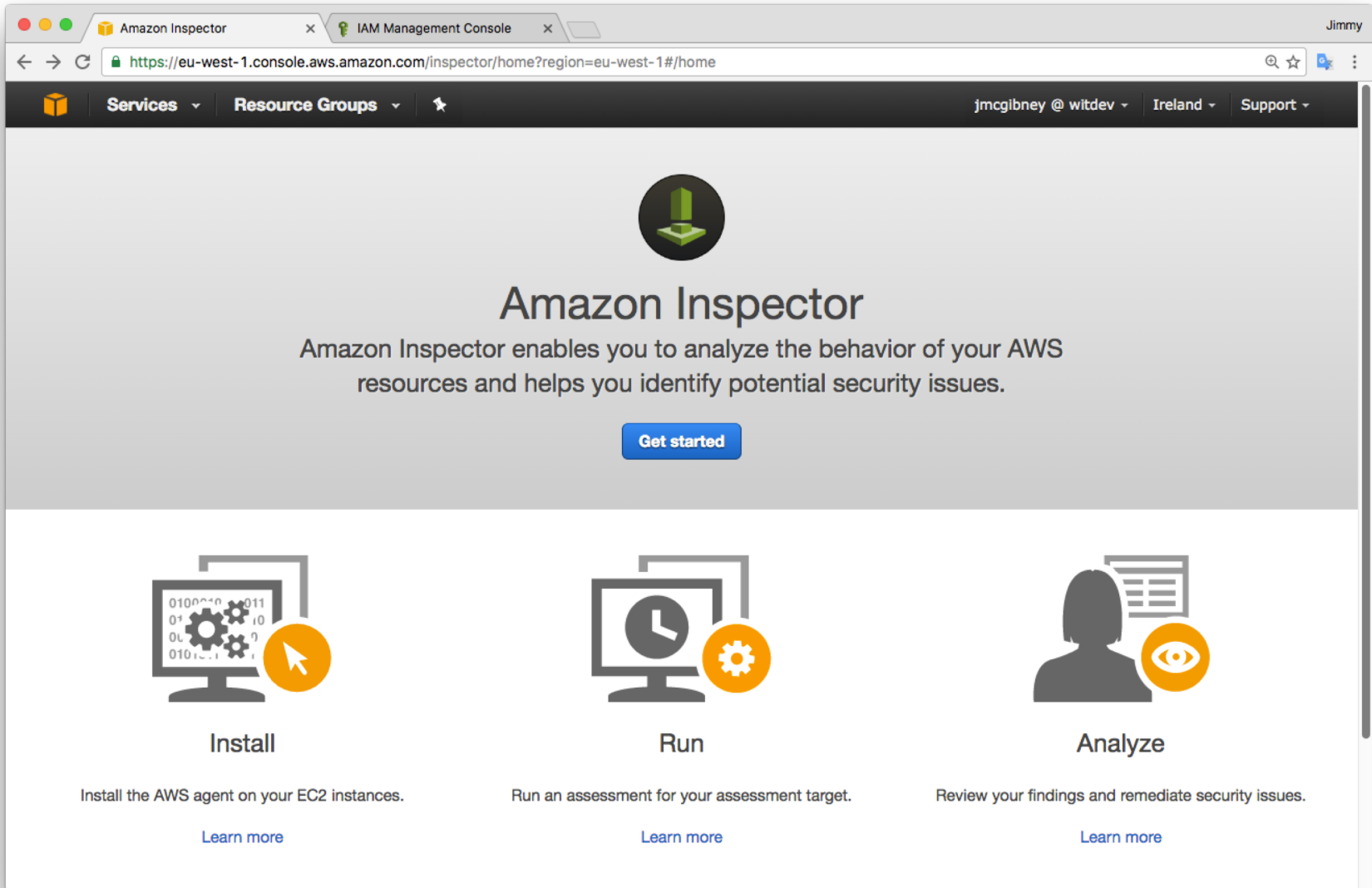
and

**String match condition**  
Bad bots

or if requests match

**Rule 2, Detect SQLi, then block**

# Inspector



The screenshot shows the Amazon Inspector console interface. At the top, there's a navigation bar with 'Services' and 'Resource Groups' dropdowns, and a user profile 'jmcglibney @ witdev' in the 'Ireland' region. The main heading is 'Amazon Inspector' with a subtext: 'Amazon Inspector enables you to analyze the behavior of your AWS resources and helps you identify potential security issues.' A 'Get started' button is prominently displayed. Below this, three main steps are outlined: 'Install' (with an icon of a monitor and gears), 'Run' (with an icon of a monitor and a clock), and 'Analyze' (with an icon of a person and an eye). Each step includes a brief description and a 'Learn more' link.

Amazon Inspector

Amazon Inspector enables you to analyze the behavior of your AWS resources and helps you identify potential security issues.

[Get started](#)

**Install**

Install the AWS agent on your EC2 instances.

[Learn more](#)

**Run**

Run an assessment for your assessment target.

[Learn more](#)

**Analyze**

Review your findings and remediate security issues.

[Learn more](#)

# Certificate Manager

The screenshot shows the AWS Certificate Manager console in the eu-west-1 region. The 'Import a certificate' wizard is active, with Step 1: Import certificate selected. A blue box indicates that AWS Certificate Manager certificates are only used with Elastic Load Balancing. The 'Select certificate' section prompts the user to paste the PEM-encoded certificate body, private key, and certificate chain. The 'Certificate body' field contains a sample certificate, the 'Certificate private key' field contains a sample private key, and the 'Certificate chain' field contains a sample certificate chain. A 'Create Listener' dialog is overlaid on the right, showing the 'Protocol' set to HTTPS (Secure HTTP), the 'Port' set to 443, and the 'Default target group' set to temp-tg. Below the dialog, the 'Select Certificate' section explains that an SSL Certificate allows configuring HTTPS/SSL listeners. It offers three options: 'Choose an existing certificate from AWS Certificate Manager (ACM)', 'Choose an existing certificate from AWS Identity and Access Management (IAM)', and 'Upload a new SSL certificate to AWS Identity and Access Management (IAM)'. A blue box highlights the 'Request a new certificate from ACM' option, which lists several certificates, including 'temp-lb-955413093.eu-west-1.elb.amazonaws.com' and 'temp-lb-955413093.eu-west-1.elb.amazonaws.com'.

**Import a certificate**

Step 1: Import certificate  
Step 2: Review and import

You can use AWS Certificate Manager certificates only with Elastic Load Balancing. [Learn more.](#)

**Select certificate**

Paste the PEM-encoded certificate body, private key, and certificate chain below. [Learn more.](#)

**Certificate body\***

```
-----BEGIN CERTIFICATE-----
MIIDIDCCAnACCCQCCVeY9vDWDQzANBgkqhkiG9w0BAQsFADCBhTElMAkGA1UEBhMC
SUUxEADAQBgNVBAgTB0lyZWxhbmQxElAQBgNVBAcTCVdhdGVyZm9yZDELMAkGA1UE
ChMCU1cxClZlBgNVBAStAINXMTYwNAYDVQQDEy10ZW1wLWxhLTk1NTQxMzA5My5l
dS13ZXN0LTFuZWxhLTk1NTYxMTE2MjI1MjIzWWhcNMTCx
-----END CERTIFICATE-----
```

**Certificate private key\***

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEArlNo1A0rrDuOnewEcl8aK3Tc1Z.I4S9hizsfIO7/.liCFUTGFv
eQ4/CeccHrwzmo1
-----END RSA PRIVATE KEY-----
```

**Certificate chain**

```
-----BEGIN TRUSTED CERTIFICATE-----
MIIDMDCCAhgCCC
RTEQMA4GA1UECg
EwJDQTElMAkGA1
-----END TRUSTED CERTIFICATE-----
```

\* Required

**Create Listener**

Protocol: HTTPS (Secure HTTP)  
Port: 443  
Default target group: temp-tg

**Select Certificate**

An SSL Certificate allows you to configure the HTTPS/SSL listeners of your load balancer. You may select an existing SSL certificate or create a new one below. [Learn more](#) about setting up HTTPS load balancers and certificate management.

**Certificate type**

- ☒ Choose an **existing** certificate from AWS Certificate Manager (ACM)
- ☐ Choose an **existing** certificate from AWS Identity and Access Management (IAM)
- ☐ Upload a **new** SSL certificate to AWS Identity and Access Management (IAM)

**Request a new certificate from ACM**

Choose a certificate

Certificate name	ARN
temp-lb-955413093.eu-west-1.elb.amazonaws.com	(arn:aws:acm:eu-west-1:808146113457:certificate/7255fdb4-f069-46a3-9606-52.17.134.20)
temp-lb-955413093.eu-west-1.elb.amazonaws.com	(arn:aws:acm:eu-west-1:808146113457:certificate/7255fdb4-f069-46a3-9606-52.17.134.20)
temp-lb-955413093.eu-west-1.elb.amazonaws.com	(arn:aws:acm:eu-west-1:808146113457:certificate/7255fdb4-f069-46a3-9606-52.17.134.20)
temp-lb-955413093.eu-west-1.elb.amazonaws.com	(arn:aws:acm:eu-west-1:808146113457:certificate/7255fdb4-f069-46a3-9606-52.17.134.20)

EC2

(load balancer config)

# Finally ... protect credentials!

 sign in  become a supporter | subscribe  search

find a job dating more ▾ International edition ▾

# theguardian

 UK world sport football opinion culture business lifestyle fashion environment tech travel

≡ all sections

home > tech

Uber

## Uber concealed massive hack that exposed data of 57m users and drivers

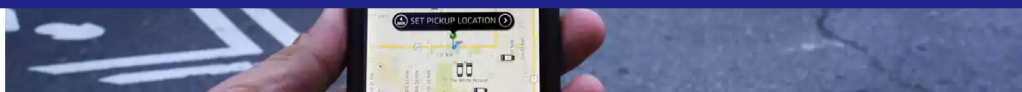
- Firm paid hackers \$100,000 to delete data and keep breach quiet
- Chief security officer Joe Sullivan fired for concealing October 2016 breach



Advertisement

According to Bloomberg, the breach occurred when two hackers obtained login credentials to access data stored on Uber's Amazon Web Services account. Paul Lipman, CEO of cybersecurity firm BullGuard, said that the fact that the data was being stored unencrypted was “unforgivable”.

Wednesday 22 November  
2017 11.16 GMT



BOMGAR

FREE TRIAL