Journey Through the Cloud

Security Best Practices

Ian Massingham — Technical Evangelist

ianmas@amazon.com

@IanMmmm

Journey Through the Cloud

1. Common use cases and adoption models for the AWS Cloud

2. Learn from the journeys taken by other AWS customers

3. Discover best practices that you can use to bootstrap your projects

Security Best Practices

Architected to be one of the most flexible and secure cloud environments

Removes many of the security headaches that come with infrastructure

Built in Security Features

Agenda

Sharing the Security Responsibility

Overview of AWS Security Features

Current Recommendations

Verifying our Security

Case Studies & Useful Resources

Increasing your Security Posture in the Cloud

People & Process

System

Network

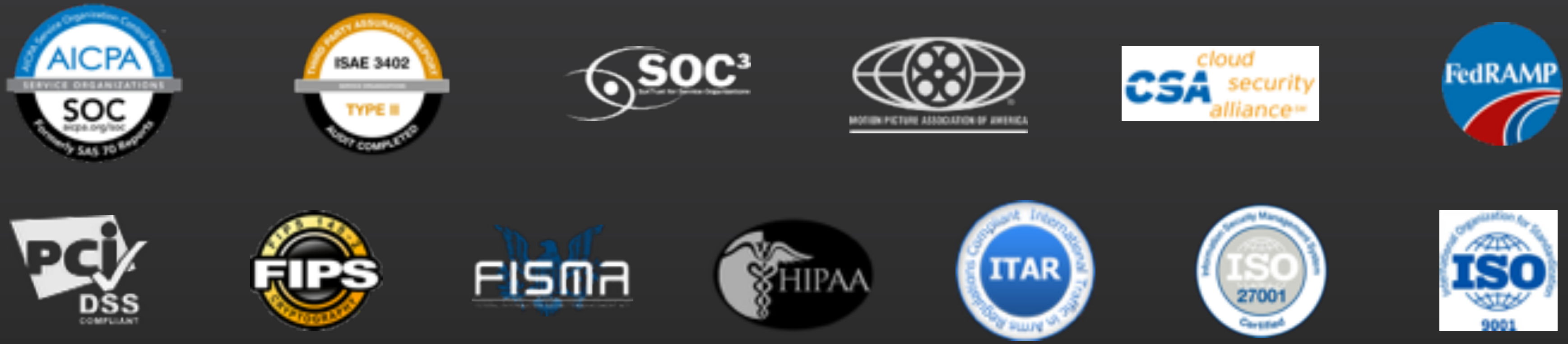
Physical

AWS security
approach

Size of AWS
security team

Visibility into
usage & resources

Broad Accreditations & Certifications


AICPA
SERVICE ORGANIZATIONS
SOC
aicpa.org/soc
Formerly SAS 70 Reports
THIRD PARTY ASSURANCE REPORT
ISAE 3402
TYPE II
AUDIT COMPLETED
SOC3
MOTION PICTURE ASSOCIATION OF AMERICA
CSA cloud security alliance
FedRAMP
PCI DSS COMPLIANT
FIPS 140-2
FIPS
CRYPTOGRAPHY
FISMA
HIPAA
Compliant International Traffic in Arms Regulations
ITAR
ISO
27001
Certified
International Organization for Standardization
ISO
9001

Security Benefits from Community Network Effect

TREND MICRO

riverbed f5

bmcsoftware

ALERTLOGIC
Security Compliance Cloud

Partner ecosystem

GE NASA

Bristol-Myers Squibb

Pinterest

Customer ecosystem

Everyone benefits

SHARING THE SECURITY RESPONSIBILITY

Shared Security Model

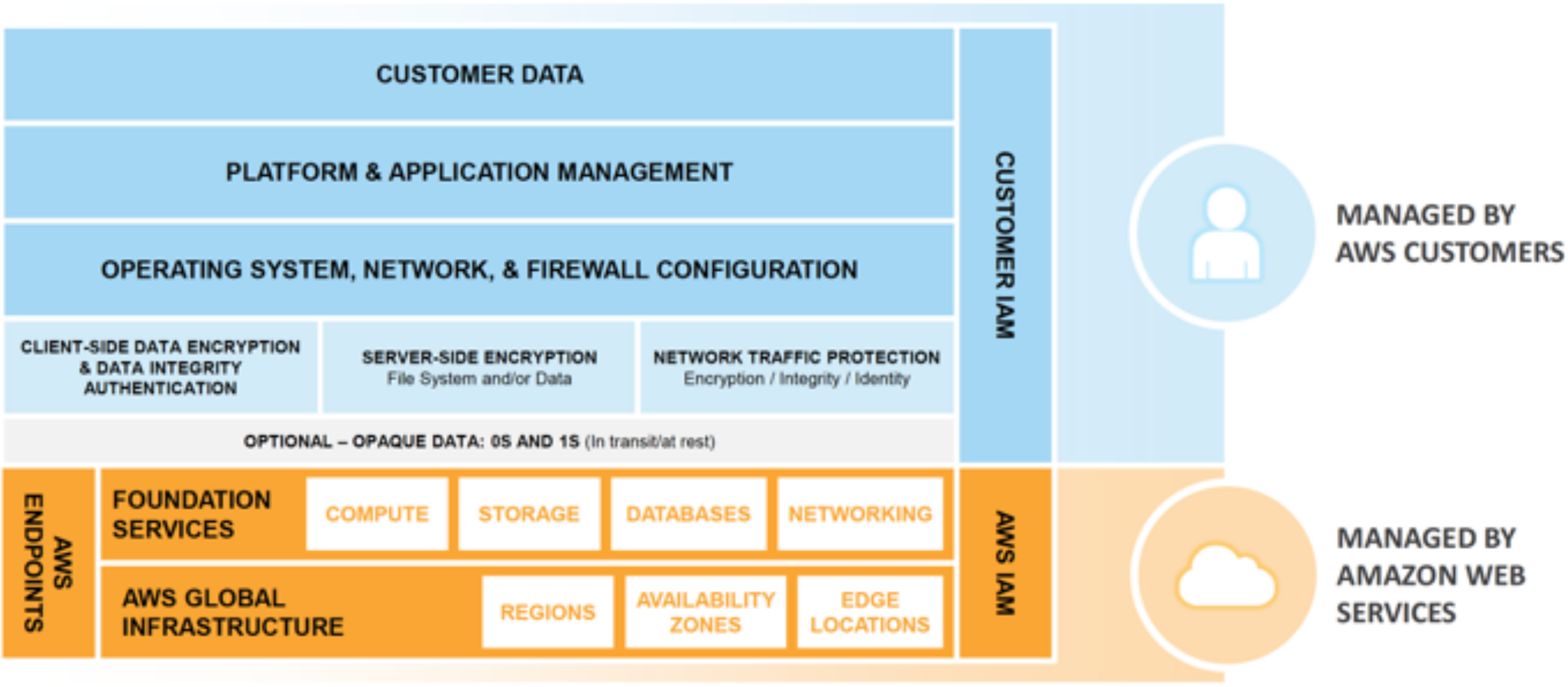
- Shared Responsibility
  - Let AWS do the heavy lifting
  - Focus on what's most valuable to your business

- AWS
  - Facility operations
  - Physical Security
  - Physical Infrastructure
  - Network Infrastructure
  - Virtualisation Infrastructure
  - Hardware lifecycle management

- Customer
  - Choice of Guest OS
  - Application Configuration Options
  - Account Management flexibility
  - Security Groups
  - ACLs
  - Identity Management

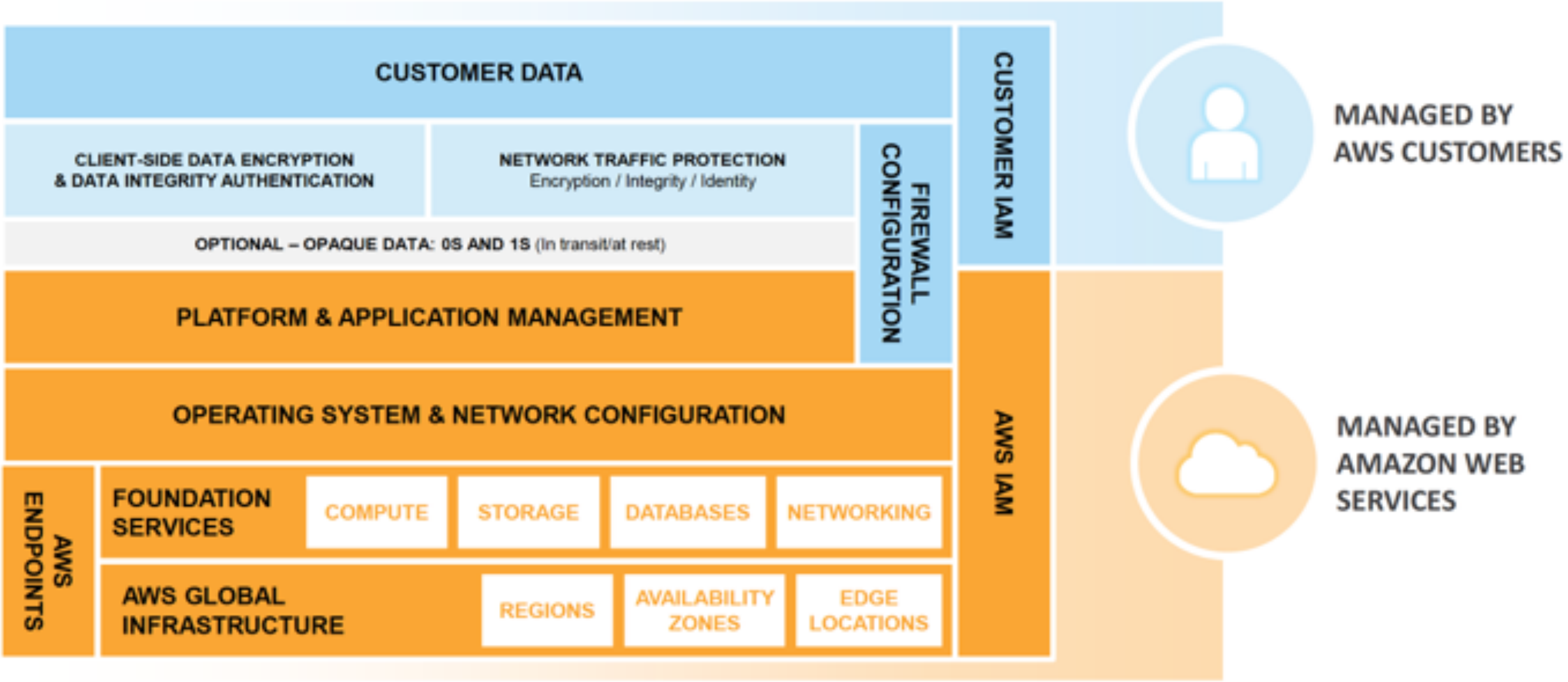
Shared Security Model: Infrastructure Services

Such as Amazon EC2, Amazon EBS, and Amazon VPC


CUSTOMER DATA
PLATFORM & APPLICATION MANAGEMENT
OPERATING SYSTEM, NETWORK, & FIREWALL CONFIGURATION
CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
SERVER-SIDE ENCRYPTION
File System and/or Data
NETWORK TRAFFIC PROTECTION
Encryption / Integrity / Identity
OPTIONAL – OPAQUE DATA: 0S AND 1S (In transit/at rest)
CUSTOMER IAM
MANAGED BY AWS CUSTOMERS
AWS ENDPOINTS
FOUNDATION SERVICES
COMPUTE
STORAGE
DATABASES
NETWORKING
AWS GLOBAL INFRASTRUCTURE
REGIONS
AVAILABILITY ZONES
EDGE LOCATIONS
AWS IAM
MANAGED BY AMAZON WEB SERVICES

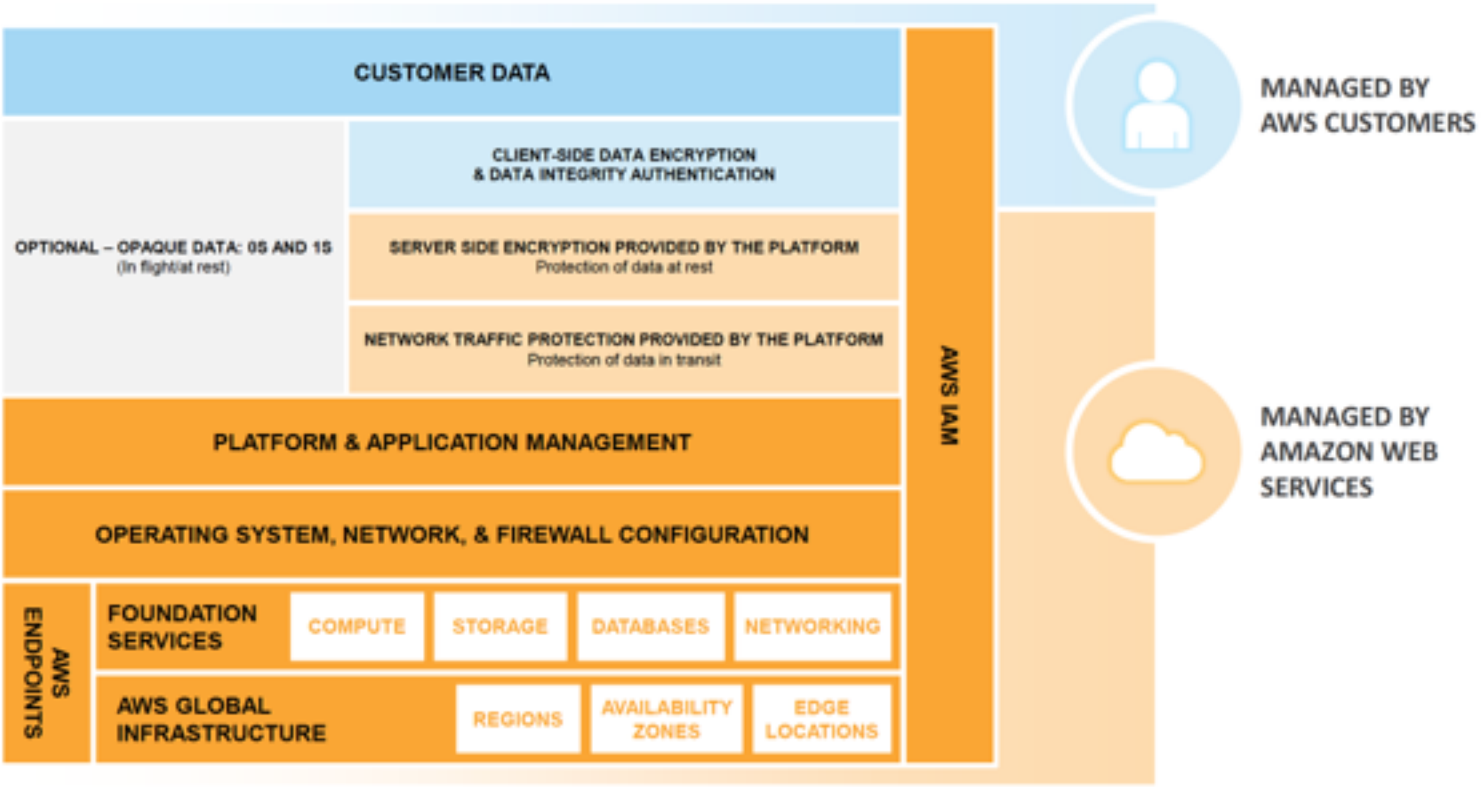
Shared Security Model: Container Services

Such as Amazon RDS and Amazon EMR


CUSTOMER DATA
CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
NETWORK TRAFFIC PROTECTION
Encryption / Integrity / Identity
OPTIONAL – OPAQUE DATA: 0S AND 1S (In transit/at rest)
FIREWALL CONFIGURATION
CUSTOMER IAM
PLATFORM & APPLICATION MANAGEMENT
OPERATING SYSTEM & NETWORK CONFIGURATION
AWS IAM
AWS ENDPOINTS
FOUNDATION SERVICES
COMPUTE
STORAGE
DATABASES
NETWORKING
AWS GLOBAL INFRASTRUCTURE
REGIONS
AVAILABILITY ZONES
EDGE LOCATIONS
MANAGED BY AWS CUSTOMERS
MANAGED BY AMAZON WEB SERVICES

Shared Security Model: Abstracted Services

Such as Amazon S3 and Amazon DynamoDB



CUSTOMER DATA

OPTIONAL – OPAQUE DATA: 0S AND 1S
(In flight/at rest)

CLIENT-SIDE DATA ENCRYPTION
& DATA INTEGRITY AUTHENTICATION

SERVER SIDE ENCRYPTION PROVIDED BY THE PLATFORM
Protection of data at rest

NETWORK TRAFFIC PROTECTION PROVIDED BY THE PLATFORM
Protection of data in transit

PLATFORM & APPLICATION MANAGEMENT

OPERATING SYSTEM, NETWORK, & FIREWALL CONFIGURATION

AWS ENDPOINTS

FOUNDATION SERVICES

COMPUTE

STORAGE

DATABASES

NETWORKING

AWS GLOBAL INFRASTRUCTURE

REGIONS

AVAILABILITY ZONES

EDGE LOCATIONS

AWS IAM

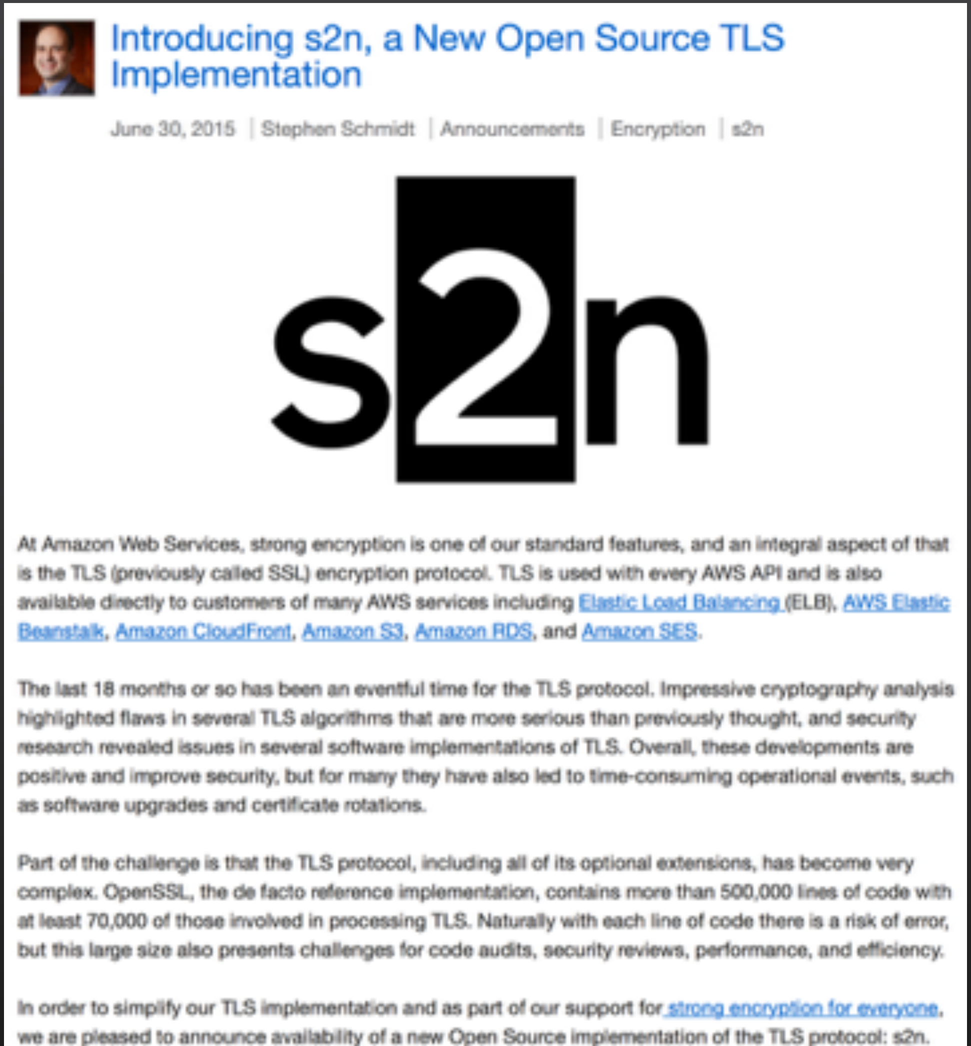
MANAGED BY AWS CUSTOMERS

MANAGED BY AMAZON WEB SERVICES

AWS SECURITY FEATURES

SECURE ACCESS

API ENDPOINTS USE TLS



Introducing s2n, a New Open Source TLS Implementation

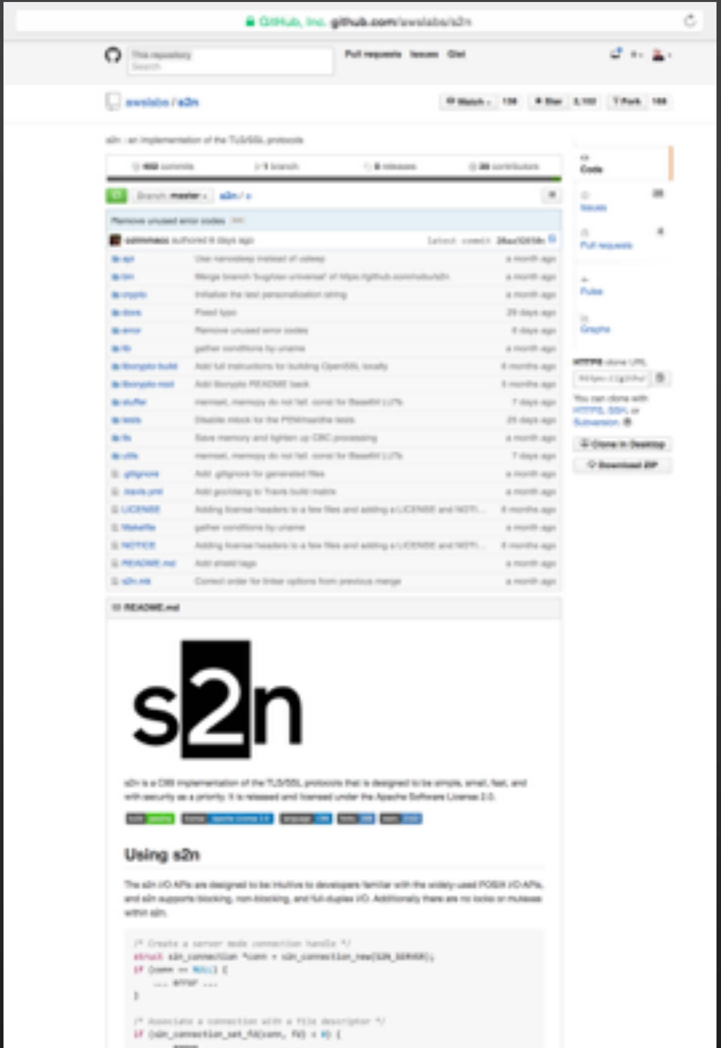
June 30, 2015 | Stephen Schmidt | Announcements | Encryption | s2n

At Amazon Web Services, strong encryption is one of our standard features, and an integral aspect of that is the TLS (previously called SSL) encryption protocol. TLS is used with every AWS API and is also available directly to customers of many AWS services including Elastic Load Balancing (ELB), AWS Elastic Beanstalk, Amazon CloudFront, Amazon S3, Amazon RDS, and Amazon SES.

The last 18 months or so has been an eventful time for the TLS protocol. Impressive cryptography analysis highlighted flaws in several TLS algorithms that are more serious than previously thought, and security research revealed issues in several software implementations of TLS. Overall, these developments are positive and improve security, but for many they have also led to time-consuming operational events, such as software upgrades and certificate rotations.

Part of the challenge is that the TLS protocol, including all of its optional extensions, has become very complex. OpenSSL, the de facto reference implementation, contains more than 500,000 lines of code with at least 70,000 of those involved in processing TLS. Naturally with each line of code there is a risk of error, but this large size also presents challenges for code audits, security reviews, performance, and efficiency.

In order to simplify our TLS implementation and as part of our support for strong encryption for everyone, we are pleased to announce availability of a new Open Source implementation of the TLS protocol: s2n.



s2n : an implementation of the TLS/SSL protocols

402 commits | 1 branch | 0 releases | 20 contributors

Branch: master | s2n / +

Remove unused error codes

colmmacc authored 8 days ago — latest commit 26ad1045dc

api	Use nanosleep instead of usleep	a month ago
bin	Merge branch 'bugfixes universal' of https://github.com/nebulas2n	a month ago
crypto	Initialize the test personalization string	a month ago
docs	Fixed typo	29 days ago
error	Remove unused error codes	8 days ago
lib	gather conditions by uname	a month ago
libcrypto-build	Add full instructions for building OpenSSL locally	4 months ago
libcrypto-root	Add libcrypto README back	4 months ago
stuffer	memset, memcpy do not fail. const for Base64 LUTs	7 days ago
tests	Disable mlock for the PEM/sanity tests	25 days ago
tls	Save memory and tighten up CBC processing	a month ago
utils	memset, memcpy do not fail. const for Base64 LUTs	7 days ago
.gitignore	Add .gitignore for generated files	a month ago
.travis.yml	Add gcc/clang to Travis build matrix	a month ago
LICENSE	Adding license headers to a few files and adding a LICENSE and NOTI…	4 months ago
Makefile	gather conditions by uname	a month ago
NOTICE	Adding license headers to a few files and adding a LICENSE and NOTI…	4 months ago
README.md	Add shield tags	a month ago
s2n.mk	Correct order for linker options from previous merge	a month ago

README.md

s2n is a C99 implementation of the TLS/SSL protocols that is designed to be simple, small, fast, and with security as a priority. It is released and licensed under the Apache Software License 2.0.

Using s2n

The s2n I/O APIs are designed to be intuitive to developers familiar with the widely used POSIX I/O APIs, and s2n supports blocking, non-blocking, and full-duplex I/O. Additionally there are no locks or mutexes within s2n.

```
/* Create a server mode connection handle */
struct s2n_connection *conn = s2n_connection_new(S2N_SERVER);
if (conn == NULL) {
    ... error ...
}

/* Associate a connection with a file descriptor */
if (s2n_connection_set_fd(conn, fd) < 0) {
```

BUILT-IN FIREWALLS

YOU CONTROL ACCESS TO YOUR INSTANCES

ROLE-BASED ACCESS CONTROL

WITH FINE-GRAINED PERMISSIONS

MULTI-FACTOR AUTHENTICATION

BUILT IN

PRIVATE SUBNETS

WITHIN YOUR AWS VIRTUAL PRIVATE CLOUD

ENCRYPT YOUR DATA AT REST

USING AES 256 BIT ENCRYPTION KEYS

CLOUD HSM

A HIGHLY SECURE WAY TO STORE KEYS

DEDICATED CONNECTION

AN OPTION WITH AWS DIRECT CONNECT

SECURITY LOGS

AWS CLOUDTRAIL, AWS CONFIG & AMAZON CLOUDWATCH LOGS

TRUSTED ADVISOR

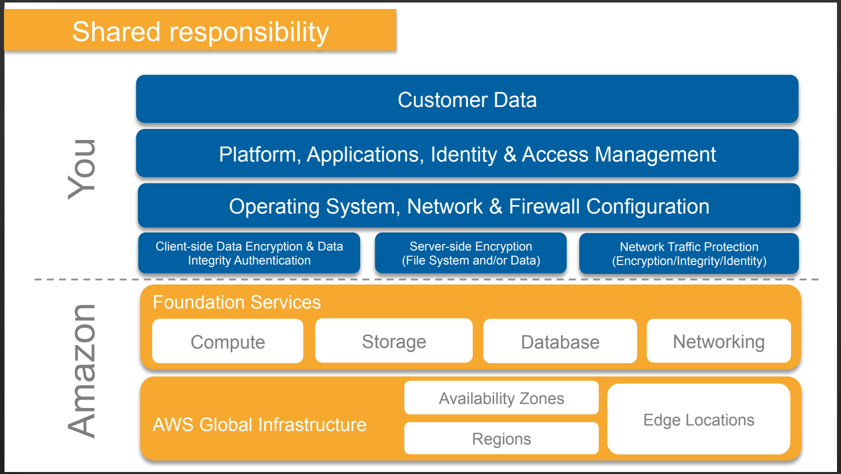
YOUR CUSTOMISED CLOUD EXPERT

CURRENT RECOMMENDATIONS

1

Know the AWS Shared Responsibility Model

Build your systems using AWS as the foundation & architect using an ISMS that takes advantage of AWS features



Shared responsibility

You

Customer Data

Platform, Applications, Identity & Access Management

Operating System, Network & Firewall Configuration

Client-side Data Encryption & Data Integrity Authentication

Server-side Encryption (File System and/or Data)

Network Traffic Protection (Encryption/Integrity/Identity)

Amazon

Foundation Services

Compute

Storage

Database

Networking

AWS Global Infrastructure

Availability Zones

Regions

Edge Locations

2

Understand the AWS Secure Global Infrastructure

Regions, Availability Zones and Endpoints

Regions

An independent collection of AWS resources in a defined geography

A solid foundation for meeting location-dependent privacy and compliance requirements

Availability Zones

Designed as independent failure zones

Physically separated within a typical metropolitan region

2

Understand the AWS Secure Global Infrastructure

Using the IAM service

AWS Identity and Access Management (IAM) enables you to securely control access to AWS services and resources for your users.

Using IAM, you can create and manage AWS users and groups and use permissions to allow and deny their access to AWS resources via credentials such as access keys, passwords and multi-factor authentication devices.

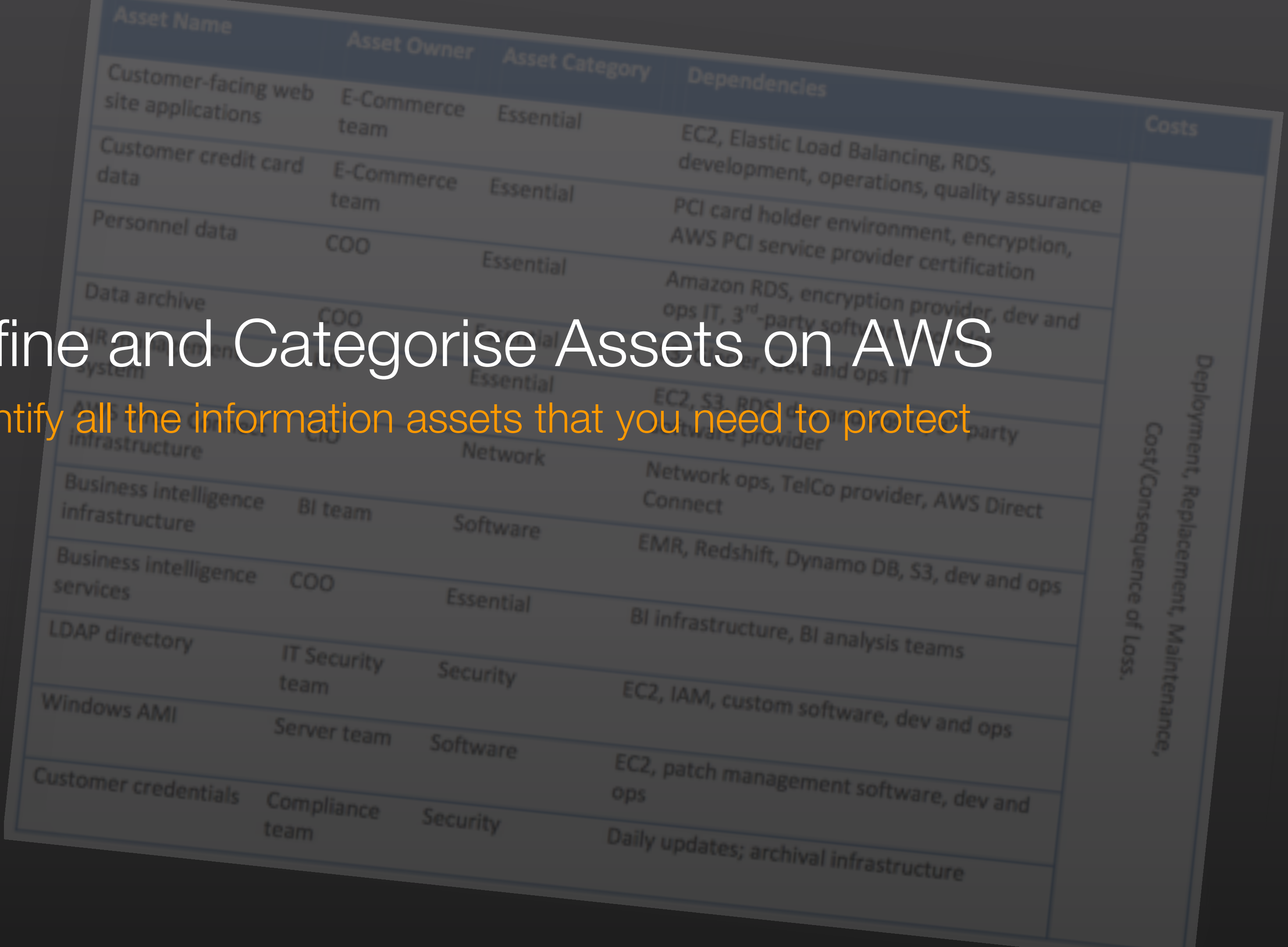
You can also federate with SAML to your own pre-existing directories of user account information, such as OpenLDAP or Active Directory

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

3

Define and Categorise Assets on AWS

Identify all the information assets that you need to protect



Asset Name	Asset Owner	Asset Category	Dependencies	Costs
Customer-facing web site applications	E-Commerce team	Essential	EC2, Elastic Load Balancing, RDS, development, operations, quality assurance	Deployment, Replacement, Maintenance, Cost/Consequence of Loss.
Customer credit card data	E-Commerce team	Essential	PCI card holder environment, encryption, AWS PCI service provider certification	
Personnel data	COO	Essential	Amazon RDS, encryption provider, dev and ops IT, 3rd-party software provider	
Data archive	COO	Essential	S3, Glacier, dev and ops IT	
HR management system	HR	Essential	EC2, S3, RDS, dev and ops IT, 3rd-party software provider	
AWS Direct Connect infrastructure	CIO	Network	Network ops, TelCo provider, AWS Direct Connect	
Business intelligence infrastructure	BI team	Software	EMR, Redshift, Dynamo DB, S3, dev and ops	
Business intelligence services	COO	Essential	BI infrastructure, BI analysis teams	
LDAP directory	IT Security team	Security	EC2, IAM, custom software, dev and ops	
Windows AMI	Server team	Software	EC2, patch management software, dev and ops	
Customer credentials	Compliance team	Security	Daily updates; archival infrastructure	

4

Design Your ISMS to Protect Your Assets on AWS

Establish a standard for implementing, operating, monitoring, reviewing, maintaining & improving your information security management system

Amazon Web Services – AWS Security Best Practices November 2013

Design Your ISMS to Protect Your Assets on AWS

After you have determined assets, categories, and costs, establish a standard for implementing, operating, monitoring, reviewing, maintaining, and improving your information security management system (ISMS) on AWS. Security requirements differ in every organization, depending on the following factors:

- Business needs and objectives
- Processes employed
- Size and structure of the organization

All these factors can change over time, so it is a good practice to build a cyclical process for managing all of this information.

[illegible] suggests a phased approach to designing and building an ISMS in AWS. You might also find standard frameworks, such as ISO 27001, helpful with ISMS design and implementation.

Phase	Title	Description
1	**Define scope and boundaries.**	Define which regions, Availability Zones, instances and AWS resources are "in scope." If you exclude any component (for example, AWS manages facilities, so you can leave it out of your own management system), state what you have excluded and why explicitly.
2	**Define an ISMS policy.**	Include the following: • Objectives that set the direction and principles for action regarding information security • Legal, contractual, and regulatory requirements • Risk management objectives for your organization • How you will measure risk • How management approves the plan
3	**Select a risk assessment methodology.**	Select a risk assessment methodolo organization about th

5

Manage AWS Accounts, IAM Users, Groups & Roles

Operate under the principle of Least Privilege

AWS Account

Your AWS account represents a business relationship between you and AWS. AWS accounts have root permissions to all AWS resources and services, so they are very powerful.

IAM Users

With IAM you can create multiple users, each with individual security credentials, all controlled under a single AWS account.
IAM users can be a person, service, or application that needs access to your AWS resources through the management console, CLI, or directly via APIs.

5

Manage AWS Accounts, IAM Users, Groups & Roles

Strategies for using multiple AWS accounts

Business Requirement	Proposed Design	Comments
Centralised security management	Single AWS Account	Centralize information security management and minimize overhead.
Separation of production, development & testing accounts	Three AWS Accounts	Create one AWS account for production services, one for development and one for testing
Multiple autonomous departments	Multiple AWS Accounts	Create separate AWS accounts for each autonomous part of the organization. You can assign permissions and policies under each account
Centralized security management with multiple autonomous independent projects	Multiple AWS Accounts	Create a single AWS account for common project resources (such as DNS services, Active Directory, CMS etc.). Then create separate AWS accounts per project. You can assign permissions and policies under each project account and grant access to resources across accounts.

5

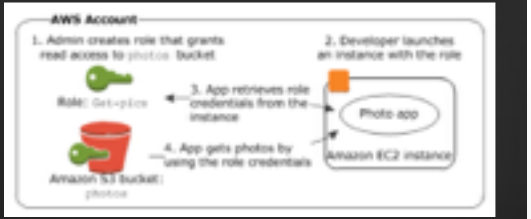
Manage AWS Accounts, IAM Users, Groups & Roles

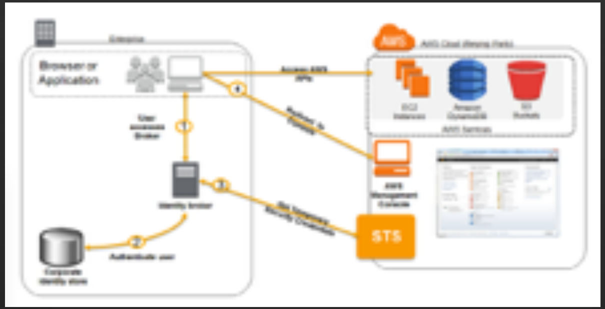
Delegation using IAM Roles and Temporary Security Credentials

Applications on Amazon EC2 that need to access AWS resources

Cross Account Access

Identity Federation


AWS Account
1. Admin creates role that grants read access to photos bucket
2. Developer launches an instance with the role
Role: Get-pics
3. App retrieves role credentials from the instance
Photo app
Amazon EC2 instance
4. App gets photos by using the role credentials
Amazon S3 bucket: photos



http://docs.aws.amazon.com/STS/latest/APIReference/Welcome.html

6

Manage OS-level Access to Amazon EC2 Instances

You own the credentials, but AWS helps you bootstrap initial access to the OS

Amazon EC2 Key Pairs

Used to authenticate SSH access to Linux instances and to generate the initial administrator password on Windows instances.

If you have higher security requirements, you are free to implement alternative authentication mechanisms and disable Amazon EC2 Key Pair Authentication

7

Secure Your Data

At rest & in transit

Resource Access Authorisation

Users or IAM Roles can only access resources after authentication

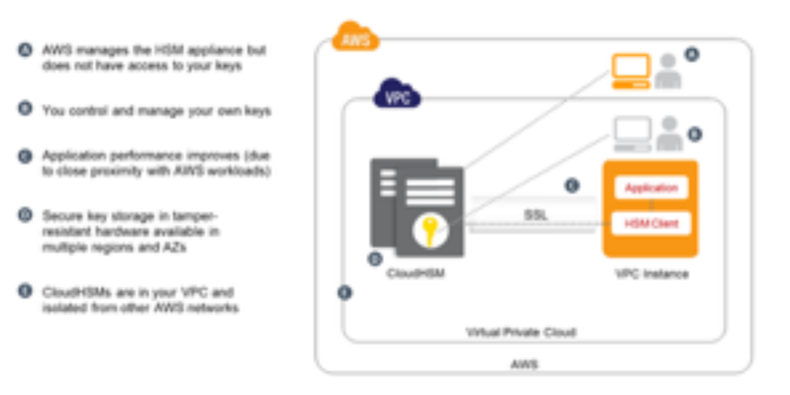
Fine-grained resources policies can restrict users or permit users to access only the resources that you specify

```
{
      "Effect": "Allow”,
      "Action": ["s3:GetObject”,"s3:PutObject”],
      "Resource": ["arn:aws:s3:::myBucket/amazon/snakegame/${cognito-identity.amazonaws.com:sub}"]
}
```

7

Secure Your Data

At rest & in transit



A AWS manages the HSM appliance but does not have access to your keys

B You control and manage your own keys

C Application performance improves (due to close proximity with AWS workloads)

D Secure key storage in tamper-resistant hardware available in multiple regions and AZs

E CloudHSMs are in your VPC and isolated from other AWS networks

Storing and Managing Encryption Keys

We recommend you store your keys in tamper-proof storage, such as Hardware Security Modules. AWS CloudHSM is one option available to help you do this, and the best option if you need third-party assurance that AWS doesn't have access to your keys; for a more easily-integrated solution, also see KMS.

As an alternative, you can store keys on your premises (eg using your own HSMs) and access these over secure links, such as via AWS Direct Connect with Ipsec, or IPsec VPNs over the Internet.

aws.amazon.com/kms/
aws.amazon.com/cloudhsm/

7

Secure Your Data

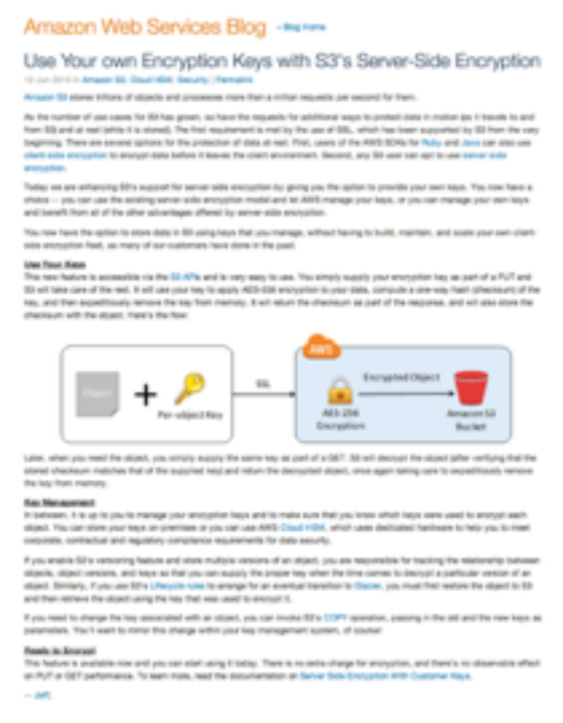
At rest & in transit

Protecting Data at Rest

Options differ by AWS Service.
Amazon S3 – Server side encryption with Amazon S3 managed keys, your own encryption keys with Customer-Provided Keys (SSE-C), or keys managed by KMS

Amazon EBS – use volume encryption provided by your operating system or KMS. For example, Windows EFS or Microsoft Windows Bitlocker, Linux dm-crypt, CloudHSM or on-premise HSM with SafeNet ProtectV

Amazon RDS – use database specific cryptographic functions, or KMS
EMR/DynamoDB – see Security Best Practices Whitepaper for options



Amazon Web Services Blog – Blog Home

Use Your own Encryption Keys with S3's Server-Side Encryption

12 Jun 2014 in Amazon S3, Cloud HSM, Security | Permalink

Amazon S3 stores trillions of objects and processes more than a million requests per second for them.

As the number of use cases for S3 has grown, so have the requests for additional ways to protect data in motion (as it travels to and from S3) and at rest (while it is stored). The first requirement is met by the use of SSL, which has been supported by S3 from the very beginning. There are several options for the protection of data at rest. First, users of the AWS SDKs for Ruby and Java can also use client side encryption to encrypt data before it leaves the client environment. Second, any S3 user can opt to use server side encryption.

Today we are enhancing S3's support for server side encryption by giving you the option to provide your own keys. You now have a choice — you can use the existing server side encryption model and let AWS manage your keys, or you can manage your own keys and benefit from all of the other advantages offered by server-side encryption.

You now have the option to store data in S3 using keys that you manage, without having to build, maintain, and scale your own client-side encryption fleet, as many of our customers have done in the past.

Use Your Keys
This new feature is accessible via the S3 APIs and is very easy to use. You simply supply your encryption key as part of a PUT and S3 will take care of the rest. It will use your key to apply AES-256 encryption to your data, compute a one-way hash (checksum) of the key, and then expeditiously remove the key from memory. It will return the checksum as part of the response, and will also store the checksum with the object. Here's the flow:

Later, when you need the object, you simply supply the same key as part of a GET. S3 will decrypt the object (after verifying that the stored checksum matches that of the supplied key) and return the decrypted object, once again taking care to expeditiously remove the key from memory.

Key Management
In between, it is up to you to manage your encryption keys and to make sure that you know which keys were used to encrypt each object. You can store your keys on-premises or you can use AWS Cloud HSM, which uses dedicated hardware to help you to meet corporate, contractual and regulatory compliance requirements for data security.

If you enable S3's versioning feature and store multiple versions of an object, you are responsible for tracking the relationship between objects, object versions, and keys so that you can supply the proper key when the time comes to decrypt a particular version of an object. Similarly, if you use S3's Lifecycle rules to arrange for an eventual transition to Glacier, you must first restore the object to S3 and then retrieve the object using the key that was used to encrypt it.

If you need to change the key associated with an object, you can invoke S3's COPY operation, passing in the old and the new keys as parameters. You'll want to mirror this change within your key management system, of course!

Ready to Encrypt
This feature is available now and you can start using it today. There is no extra charge for encryption, and there's no observable effect on PUT or GET performance. To learn more, read the documentation on Server Side Encryption With Customer Keys.

— Jeff;

8

Secure Your Operating Systems & Applications

With the shared responsibility model you manage operating systems & application security

OS Hardening and Updates

Use of Amazon Machine Images (AMIs) makes it easy to deploy standardized operating system and application builds

Amazon provides and maintains a preconfigured set of AMIs, but you are also free to create your own and use these as the basis for EC2 instances that you deploy

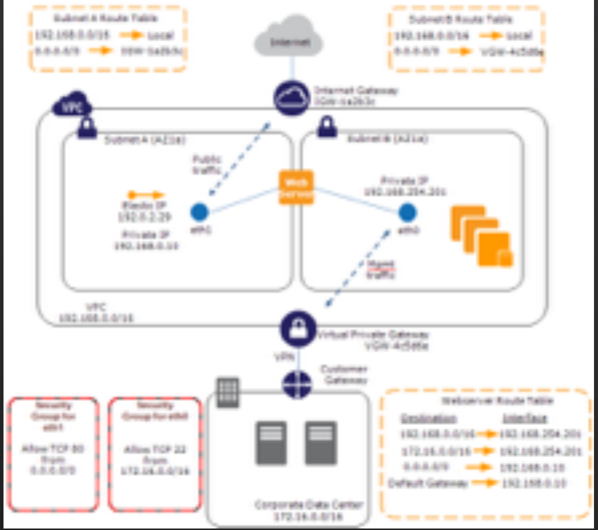
Standard OS hardening principles (eg CIS Benchmarks, DISA STIGs) can and should be applied to the operating systems that you chose to run on EC2 instances

There are lots more detailed recommendations for securing your OS environment in the AWS Security Best Practices Whitepaper

9

Secure Your Infrastructure

Using AWS platform features



Amazon Virtual Private Cloud (VPC)

Create private clouds with Layer 2 separation, within the AWS Cloud

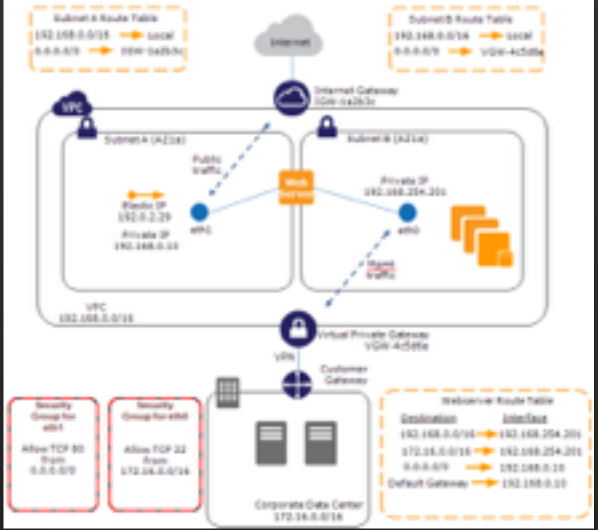
Use your own IP address space, allocated by you. Use RFC1918 private address space for non-internet-routable networks

Connect to your VPC via the Internet, IPsec over the Internet, AWS Direct Connect, AWS Direct Connect with IPsec or a combination of these.
Define your own subnet topology, routing table and create custom service instances such as DNS or time servers

9

Secure Your Infrastructure

Using AWS platform features



Security Zoning and Network Segmentation

Network segmentation simply isolates one network from another

Security zones are groups of system components with similar security levels that have common controls applied to them

Combine AWS platform security features with your own overlay infrastructure components such as repositories, DNS & time servers to segment networks and create security zones

The AWS elastic cloud infrastructure & automated deployment tools mean that you can apply the same security controls across all AWS regions
Repeatable and uniform deployments improve your overall security posture

10

Monitoring, Alerting, Audit Trail & Incident Response

Adapt existing processes, tools & methodologies for use in the cloud

Area	Consideration
Log collection	Note how log files are collected. Often operating system, application, or third-party/middleware agents collect log file information
Log transport	When log files are centralized, transfer them to the central location in a secure, reliable, and timely fashion
Log storage	Centralize log files from multiple instances to facilitate retention policies, as well as analysis and correlation
Log taxonomy	Present different categories of log files in a format suitable for analysis
Log analysis/ correlation	Log files provide security intelligence after you analyze them and correlate events in them. You can analyze logs in real time, or at scheduled intervals.
Log protection/ security	Log files are sensitive. Protect them through network control, identity and access management, protection/ encryption, data integrity authentication, and tamper-proof time-stamping

Implement OS & Higher Level Monitoring

Logs may be generated by a variety of network components as well as operating systems, platforms and applications

We recommend logging and analysis of the following event types:

- Actions taken by any individual with root or administrative privileges
- Access to all audit trails
- Invalid logical access attempts
- Use of identification and authentication mechanisms
- Initialisation of audit logs
- Creation, deletion and modification of system level objects

10

Monitoring, Alerting, Audit Trail & Incident Response

Adapt existing processes, tools & methodologies for use in the cloud

Area	Consideration
Log collection	Note how log files are collected. Often operating system, application, or third-party/middleware agents collect log file information
Log transport	When log files are centralized, transfer them to the central location in a secure, reliable, and timely fashion
Log storage	Centralize log files from multiple instances to facilitate retention policies, as well as analysis and correlation
Log taxonomy	Present different categories of log files in a format suitable for analysis
Log analysis/ correlation	Log files provide security intelligence after you analyze them and correlate events in them. You can analyze logs in real time, or at scheduled intervals.
Log protection/ security	Log files are sensitive. Protect them through network control, identity and access management, protection/ encryption, data integrity authentication, and tamper-proof time-stamping

Use CloudWatch Logs to Centralise Your Logs

CloudWatch Logs enables you to monitor and troubleshoot your systems and applications using your existing system, application, and custom log files.

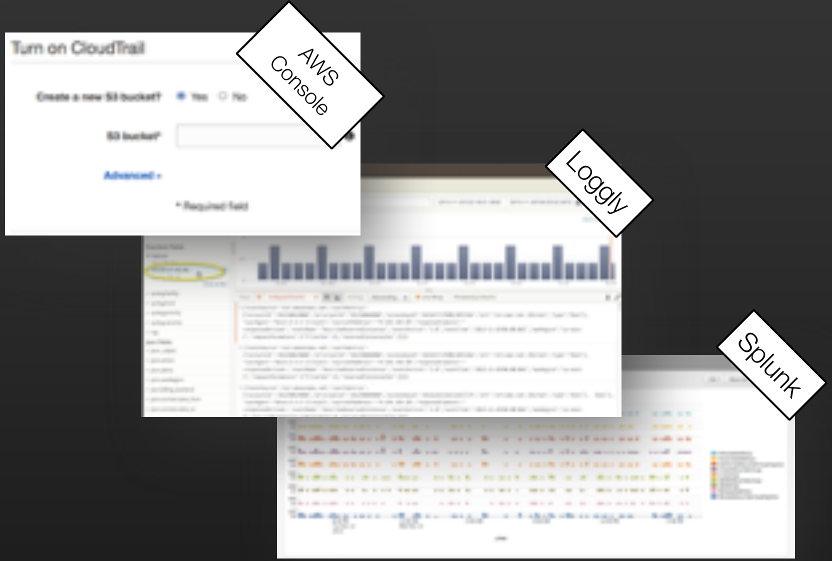
Send your existing system, application, and custom log files to CloudWatch Logs via our agent, and monitor these logs in near real-time.

This can help you better understand and operate your systems and applications, and you can store your logs using highly durable, low-cost storage for later access

10

Monitoring, Alerting, Audit Trail & Incident Response

Adapt existing processes, tools & methodologies for use in the cloud



Turn on CloudTrail

Create a new S3 bucket? ● Yes ○ No

S3 bucket*

Advanced ›

* Required field

Use CloudTrail to Record AWS API Calls

AWS CloudTrail is a web service that records AWS API calls for your account and delivers log files to you.

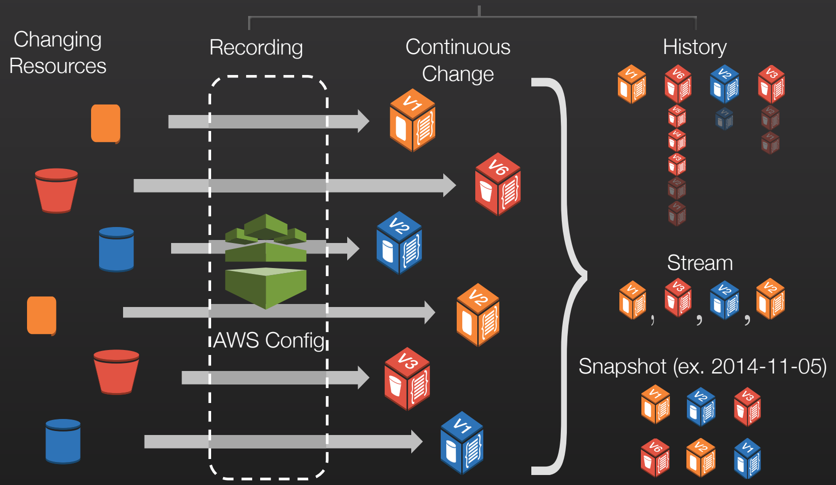
The recorded information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by the AWS service.

With CloudTrail, you can get a history of AWS API calls for your account. The AWS API call history produced by CloudTrail enables security analysis, resource change tracking, and compliance auditing.

10

Monitoring, Alerting, Audit Trail & Incident Response

Adapt existing processes, tools & methodologies for use in the cloud



Use AWS Config to Record AWS Environment Changes

AWS Config is a service that records AWS environment configurations, changes and relationships for your account and delivers log files to you.

The recorded information includes the configuration and metadata for VPCs, Subnets, NACLS, Security Groups, VGWs, Internet Gateways, Elastic IPs etc and the relationships between them, and the time of the change.

Snapshots answer the question “What did my environment look like, at time t?”

History answers the question “What changes have happened, to infrastructure element I over time?”

10

Monitoring, Alerting, Audit Trail & Incident Response

Adapt existing processes, tools & methodologies for use in the cloud

VERIFYING OUR SECURITY

Compliance at AWS

AWS is Level 1 compliant under the Payment Card Industry (PCI) Data Security Standard (DSS). Customers can run applications on our PCI-compliant technology infrastructure for storing, processing, and transmitting credit card information in the cloud.

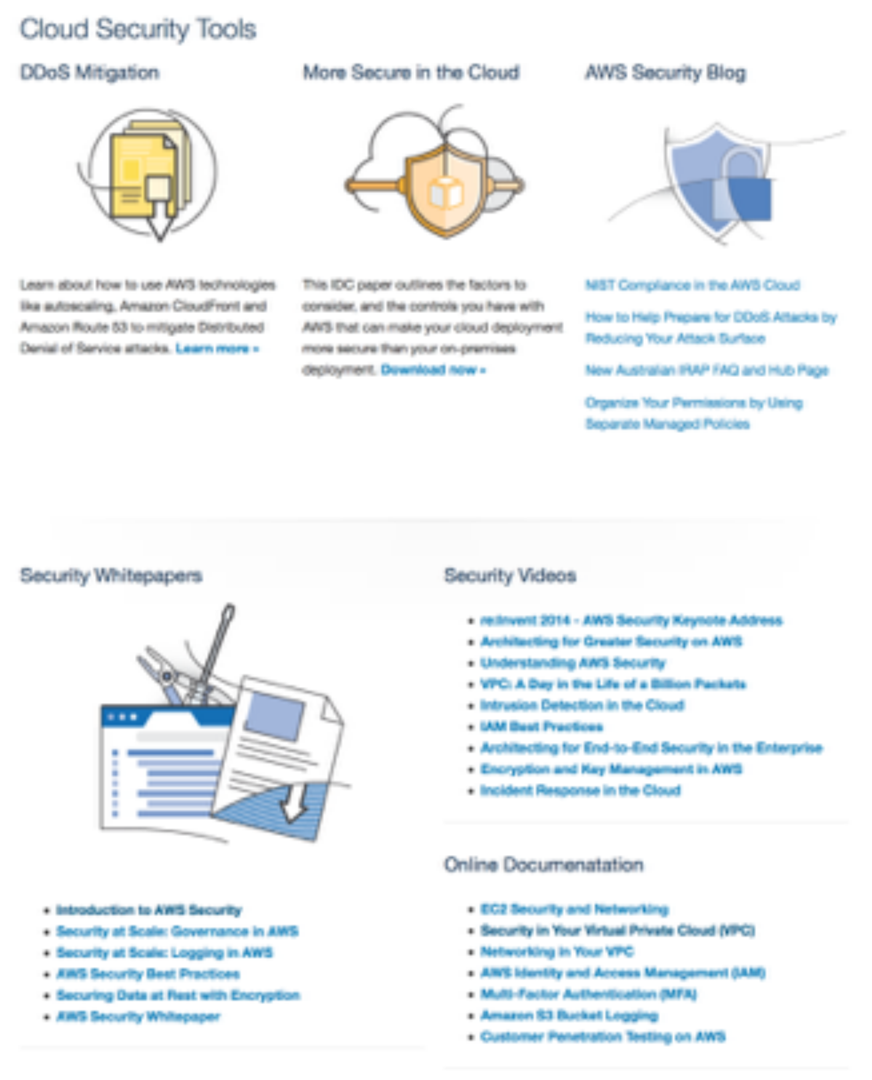
AWS is ISO 27001 certified under the International Organization for Standardization (ISO) 27001 standard. ISO 27001 is a widely-adopted global security standard that outlines the requirements for information security management systems.

Many other government and industry compliance requirements are also met by AWS. Find more at:

aws.amazon.com/compliance

RESOURCES YOU CAN USE TO LEARN MORE

aws.amazon.com/security/



Cloud Security Tools

DDoS Mitigation

Learn about how to use AWS technologies like autoscaling, Amazon CloudFront and Amazon Route 53 to mitigate Distributed Denial of Service attacks. Learn more »

More Secure in the Cloud

This IDC paper outlines the factors to consider, and the controls you have with AWS that can make your cloud deployment more secure than your on-premises deployment. Download now »

AWS Security Blog

NIST Compliance in the AWS Cloud

How to Help Prepare for DDoS Attacks by Reducing Your Attack Surface

New Australian IRAP FAQ and Hub Page

Organize Your Permissions by Using Separate Managed Policies

Security Whitepapers

- Introduction to AWS Security
- Security at Scale: Governance in AWS
- Security at Scale: Logging in AWS
- AWS Security Best Practices
- Securing Data at Rest with Encryption
- AWS Security Whitepaper

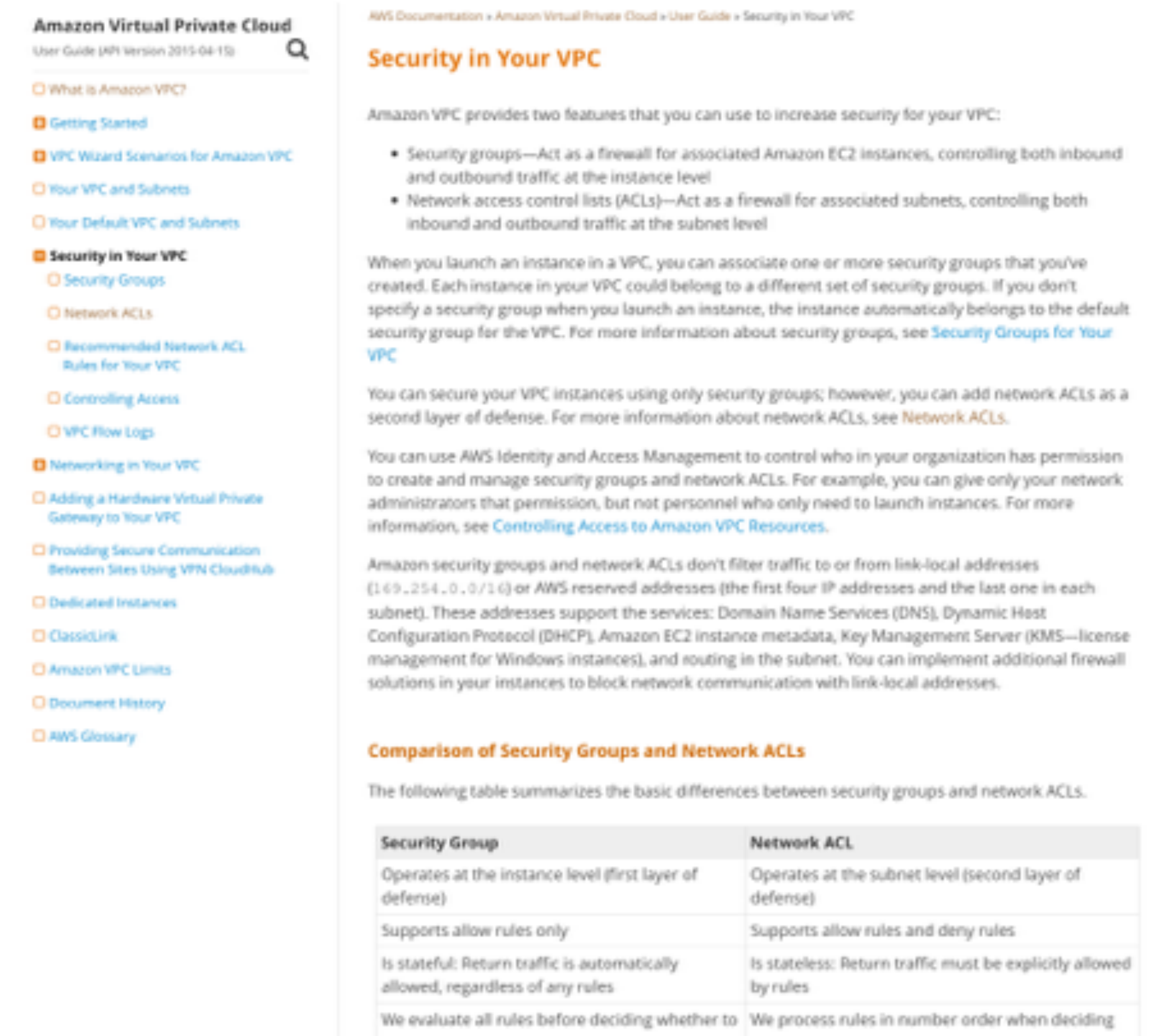
Security Videos

- re:Invent 2014 - AWS Security Keynote Address
- Architecting for Greater Security on AWS
- Understanding AWS Security
- VPC: A Day in the Life of a Billion Packets
- Intrusion Detection in the Cloud
- IAM Best Practices
- Architecting for End-to-End Security in the Enterprise
- Encryption and Key Management in AWS
- Incident Response in the Cloud

Online Documenatation

- EC2 Security and Networking
- Security in Your Virtual Private Cloud (VPC)
- Networking in Your VPC
- AWS Identity and Access Management (IAM)
- Multi-Factor Authentication (MFA)
- Amazon S3 Bucket Logging
- Customer Penetration Testing on AWS

AWS Technical Documentation



Amazon Virtual Private Cloud

User Guide (API Version 2015-04-15)

Security in Your VPC

Amazon VPC provides two features that you can use to increase security for your VPC:

- Security groups—Act as a firewall for associated Amazon EC2 instances, controlling both inbound and outbound traffic at the instance level
- Network access control lists (ACLs)—Act as a firewall for associated subnets, controlling both inbound and outbound traffic at the subnet level

When you launch an instance in a VPC, you can associate one or more security groups that you've created. Each instance in your VPC could belong to a different set of security groups. If you don't specify a security group when you launch an instance, the instance automatically belongs to the default security group for the VPC. For more information about security groups, see Security Groups for Your VPC

You can secure your VPC instances using only security groups; however, you can add network ACLs as a second layer of defense. For more information about network ACLs, see Network ACLs.

You can use AWS Identity and Access Management to control who in your organization has permission to create and manage security groups and network ACLs. For example, you can give only your network administrators that permission, but not personnel who only need to launch instances. For more information, see Controlling Access to Amazon VPC Resources.

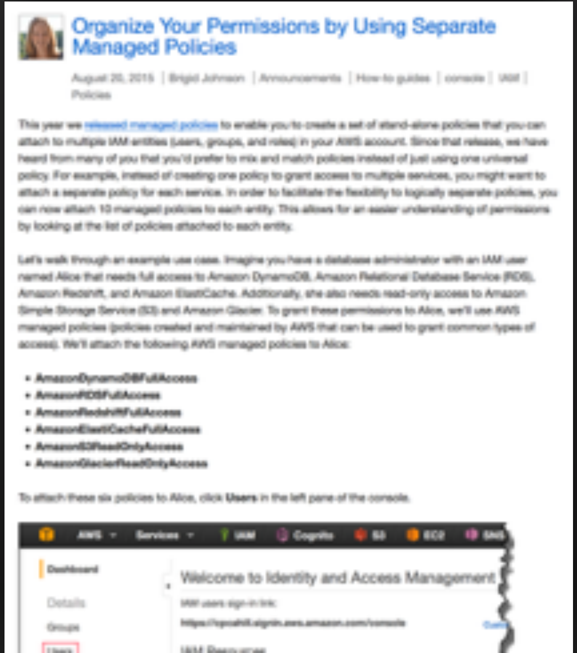
Amazon security groups and network ACLs don't filter traffic to or from link-local addresses (`169.254.0.0/16`) or AWS reserved addresses (the first four IP addresses and the last one in each subnet). These addresses support the services: Domain Name Services (DNS), Dynamic Host Configuration Protocol (DHCP), Amazon EC2 instance metadata, Key Management Server (KMS—license management for Windows instances), and routing in the subnet. You can implement additional firewall solutions in your instances to block network communication with link-local addresses.

Comparison of Security Groups and Network ACLs

The following table summarizes the basic differences between security groups and network ACLs.

Security Group	Network ACL
Operates at the instance level (first layer of defense)	Operates at the subnet level (second layer of defense)
Supports allow rules only	Supports allow rules and deny rules
Is stateful: Return traffic is automatically allowed, regardless of any rules	Is stateless: Return traffic must be explicitly allowed by rules
We evaluate all rules before deciding whether to	We process rules in number order when deciding

blogs.aws.amazon.com/security



Organize Your Permissions by Using Separate Managed Policies

August 20, 2015 | Brigid Johnson | Announcements | How-to guides | console | IAM | Policies

This year we released managed policies to enable you to create a set of stand-alone policies that you can attach to multiple IAM entities (users, groups, and roles) in your AWS account. Since that release, we have heard from many of you that you'd prefer to mix and match policies instead of just using one universal policy. For example, instead of creating one policy to grant access to multiple services, you might want to attach a separate policy for each service. In order to facilitate the flexibility to logically separate policies, you can now attach 10 managed policies to each entity. This allows for an easier understanding of permissions by looking at the list of policies attached to each entity.

Let's walk through an example use case. Imagine you have a database administrator with an IAM user named Alice that needs full access to Amazon DynamoDB, Amazon Relational Database Service (RDS), Amazon Redshift, and Amazon ElastiCache. Additionally, she also needs read-only access to Amazon Simple Storage Service (S3) and Amazon Glacier. To grant these permissions to Alice, we'll use AWS managed policies (policies created and maintained by AWS that can be used to grant common types of access). We'll attach the following AWS managed policies to Alice:

- **AmazonDynamoDBFullAccess**
- **AmazonRDSFullAccess**
- **AmazonRedshiftFullAccess**
- **AmazonElastiCacheFullAccess**
- **AmazonS3ReadOnlyAccess**
- **AmazonGlacierReadOnlyAccess**

To attach these six policies to Alice, click **Users** in the left pane of the console.

Welcome to Identity and Access Management

IAM users sign-in link:

https://opcahill.signin.aws.amazon.com/console

IAM Resources

How to Manage Identities in Simple AD Directories

August 18, 2015 | Chen Wong | How-to guides | Amazon Linux | Directory Service | Simple AD

As I said in yesterday's blog post, How to Migrate Your Microsoft Active Directory Users to Simple AD, AWS Directory Service allows you to create a standalone, highly available AWS-managed directory called Simple AD in a matter of minutes. With Simple AD, you can centrally manage user accounts and group memberships for Amazon EC2 instances joined to a domain. It also allows you to use a single set of credentials to log in across all EC2 instances as well as provide authentication to your applications. For more information about Simple AD, see What is AWS Directory Service?

In yesterday's post, I showed you how to migrate your identities from Microsoft Active Directory to Simple AD. In today's post, I will talk about the commands you can use to help manage those identities in Linux and Windows environments.

Important note: Before making changes to your Simple AD directory, it is important to keep snapshots as a backup. If you need to create a snapshot of your directory now, follow these instructions.

Managing Simple AD

The following commands enable you to manage the user accounts and group memberships for your Simple AD directory. The following links take you to instructions about how to install and use Active Directory Users and Computers on EC2 instances running Microsoft Windows:

- Installing the Active Directory Administration Tools
- Creating Users and Groups

Equivalent commands for Linux are described in this post.

Note: The following instructions refer to using EC2 instances running Amazon Linux. Other Linux distributions may have different commands but should be similar. Launch and join the instance to the domain by following these instructions. Connect to the instance with a user that has rights to create objects in the domain (in other words, a Domain Admin user) using any SSH client.

These are the values used in the commands in this post:

- User name: `johndoe`

How to Address the PCI DSS Requirements for Data Encryption in Transit Using Amazon VPC

July 23, 2015 | Balaji Palanisamy | Compliance | Encryption | Amazon VPC | PCI DSS

The PCI requirements for encryption for data in transit are different for private networks than they are for public networks. When correctly designed, Amazon Virtual Private Cloud (Amazon VPC), a logically isolated portion of the AWS infrastructure that allows you to extend your existing data center network to the cloud, can be considered a private network, as qualified by the Payment Card Industry Data Security Standards (PCI DSS).

In this blog post, I will review the importance of understanding the logical isolation provided by Amazon VPC and then review some of the key points to consider when designing for PCI workloads that need to transmit sensitive data within or outside the AWS infrastructure. I will also demonstrate how you can use the native isolation provided by Amazon VPC for additional security.

Amazon VPC is the architectural construct of choice for AWS customers deploying workloads that are in scope for a PCI DSS assessment. Within Amazon VPC, Amazon EC2 instances must have an Internet gateway or a virtual private gateway in order to communicate with hosts outside Amazon VPC. Additionally, AWS-designed Layer 2 networking features include the mapping service, which performs checks to ensure that even packets with malformed or modified addresses cannot hop across Amazon VPC boundaries. Network access control lists (NACLs) and security groups may be used to filter inbound and outbound traffic to hosts within Amazon VPC. These controls make it difficult for data to be intercepted or diverted while in transit, and demonstrate the private nature of Amazon VPC.

Encryption of sensitive data in motion is addressed in PCI DSS version 3.1 via Requirement 4 and its corresponding subrequirements. The DSS is clear that the requirements apply to the transmission of payment card data across "open, public networks" that are susceptible to unauthorized access. The PCI DSS and the PCI Glossary describe public networks as network transport providers that connect an organization's networks to each other over a wide area network (WAN), to the Internet, or to partner networks—and not software-defined cloud constructs such as Amazon VPC.

Typically, such public networks exhibit managed ingress and egress points that act as gateways to a shared network, with the provider managing the routing within the shared network. It is also possible that the ingress and egress points may be represented by dedicated physical hardware called the customer-premises equipment (CPE). On the other hand, the software-defined Amazon VPC abstracts any underlying hardware and allows for logical isolation. Additionally, PCI DSS testing procedures such as 4.1.c require the PCI Qualified Security Assessor (QSA) to "observe a sample of inbound and outbound transmissions as they

AWS Security White Papers

Amazon Web Services – Overview of Security Processes July 2015

amazon web services

Introduction to AWS Security

July 2015

Page 1 of 7

Introduction to AWS Security

Security at Scale: Governance in AWS

Security at Scale: Logging in AWS

AWS Security Best Practices

Securing Data at Rest with Encryption

AWS Security Whitepaper

aws.amazon.com/iam

aws.amazon.com/vpc

aws.amazon.com/kms

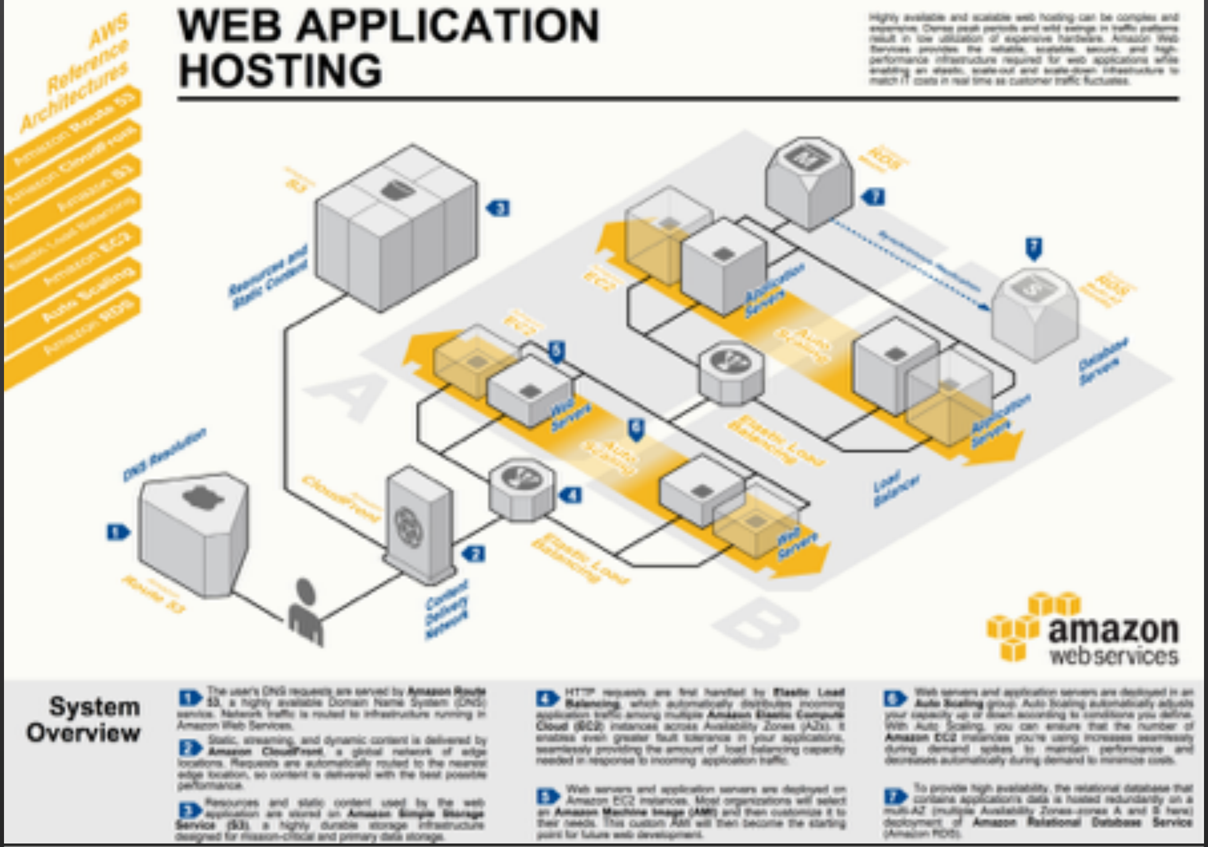
aws.amazon.com/config

aws.amazon.com/cloudtrail

aws.amazon.com/cloudhsm

aws.amazon.com/cloudwatch

aws.amazon.com/trustedadvisor



WEB APPLICATION HOSTING

Highly available and scalable web hosting can be complex and expensive. Dense peak periods and wild swings in traffic patterns result in low utilization of expensive hardware. Amazon Web Services provides the reliable, scalable, secure, and high-performance infrastructure required for web applications while enabling an elastic, scale-out and scale-down infrastructure to match IT costs in real time as customer traffic fluctuates.

System Overview

1 The user's DNS requests are served by **Amazon Route 53**, a highly available Domain Name System (DNS) service. Network traffic is routed to infrastructure running in Amazon Web Services.

2 Static, streaming, and dynamic content is delivered by **Amazon CloudFront**, a global network of edge locations. Requests are automatically routed to the nearest edge location, so content is delivered with the best possible performance.

3 Resources and static content used by the web application are stored on **Amazon Simple Storage Service (S3)**, a highly durable storage infrastructure designed for mission-critical and primary data storage.

4 HTTP requests are first handled by **Elastic Load Balancing**, which automatically distributes incoming application traffic among multiple **Amazon Elastic Compute Cloud (EC2)** instances across Availability Zones (AZs). It enables even greater fault tolerance in your applications, seamlessly providing the amount of load balancing capacity needed in response to incoming application traffic.

5 Web servers and application servers are deployed on Amazon EC2 instances. Most organizations will select an **Amazon Machine Image (AMI)** and then customize it to their needs. This custom AMI will then become the starting point for future web development.

6 Web servers and application servers are deployed in an **Auto Scaling** group. Auto Scaling automatically adjusts your capacity up or down according to conditions you define. With Auto Scaling, you can ensure that the number of **Amazon EC2** instances you're using increases seamlessly during demand spikes to maintain performance and decreases automatically during demand to minimize costs.

7 To provide high availability, the relational database that contains application's data is hosted redundantly on a multi-AZ (multiple Availability Zones–zones A and B here) deployment of **Amazon Relational Database Service (Amazon RDS)**.

aws.amazon.com/architecture/

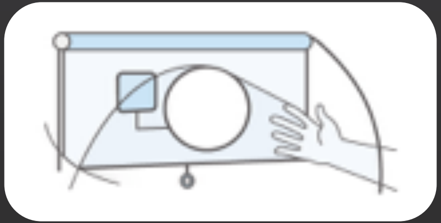
AWS Training & Certification

Self-Paced Labs

Try products, gain new skills, and get hands-on practice working with AWS technologies

aws.amazon.com/training/self-paced-labs

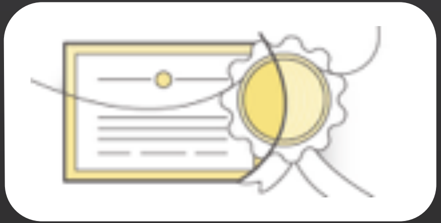
Training



Build technical expertise to design and operate scalable, efficient applications on AWS

aws.amazon.com/training

Certification



Validate your proven skills and expertise with the AWS platform

aws.amazon.com/certification

Follow us for more events & webinars

amazon web services

Ian Massingham — Technical Evangelist

🐦 @IanMmmm

🐦 @AWS_UKI for local AWS events & news

🐦 @AWScloud for Global AWS News & Announcements