Amazon Web Services: Overview of Security Processes

AWS Whitepaper

Amazon Web Services: Overview of Security Processes: AWS Whitepaper

Table of Contents

Amazon Web Services: Overview of Security Processes

Publication date: **March 1, 2020** (*Document Revisions* (p. 64))

Abstract

This document is intended to answer questions, such as *How does AWS help me ensure that my data is secure?* Specifically, this paper describes AWS physical and operational security processes for the network and server infrastructure under the management of AWS.

Introduction

Amazon Web Services (AWS) delivers a scalable cloud computing platform with high availability and dependability, providing the tools that enable customers to run a wide range of applications. Helping to protect the confidentiality, integrity, and availability of our customers' systems and data is of the utmost importance to AWS, as is maintaining customer trust and confidence.

Shared Security Responsibility Model

Before covering the details of how AWS secures its resources, it is important to understand how security in the cloud is slightly different than security in your on-premises data centers. When you move computer systems and data to the cloud, security responsibilities become shared between you and your cloud service provider. In this case, AWS is responsible for securing the underlying infrastructure that supports the cloud, and you're responsible for anything you put on the cloud or connect to the cloud. This shared security responsibility model can reduce your operational burden in many ways, and in some cases may even improve your default security posture without additional action on your part.

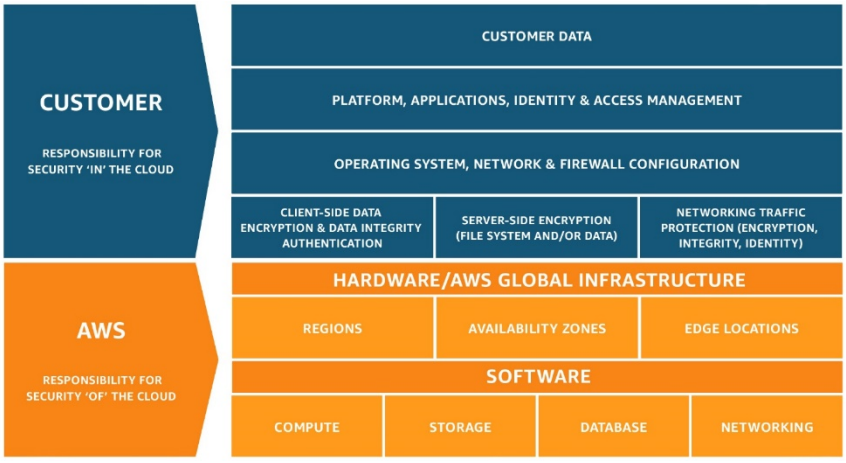

Figure 1: AWS shared security responsibility model

The amount of security configuration work you have to do varies depending on which services you select and how sensitive your data is. However, there are certain security features—such as individual user accounts and credentials, SSL/TLS for data transmissions, and user activity logging—that you should configure no matter which AWS service you use. For more information about these security features, see the AWS Account Security Features (p. 13) section.

Topics

AWS Security Responsibilities

Amazon Web Services is responsible for protecting the global infrastructure that runs all of the services offered in the AWS Cloud. This infrastructure comprises the hardware, software, networking, and facilities that run AWS services. Protecting this infrastructure is the number one priority of AWS. Although, you can't visit our data centers or offices to see this protection firsthand, we provide several reports from third-party auditors who have verified our compliance with a variety of computer security standards and regulations. For more information, visit AWS Compliance.

Note that in addition to protecting this global infrastructure, AWS is responsible for the security configuration of its products that are considered managed services. Examples of these types of services include Amazon DynamoDB, Amazon RDS, Amazon Redshift, Amazon EMR, Amazon WorkSpaces, and several other services. These services provide the scalability and flexibility of cloud-based resources with the additional benefit of being managed. For these services, AWS handles basic security tasks like guest

operating system (OS) and database patching, firewall configuration, and disaster recovery. For most of these managed services, all you have to do is configure logical access controls for the resources and protect your account credentials. A few of them may require additional tasks, such as setting up database user accounts, but overall the security configuration work is performed by the service.

Customer Security Responsibilities

With the AWS cloud, you can provision virtual servers, storage, databases, and desktops in minutes instead of weeks. You can also use cloud-based analytics and workflow tools to process your data as you need it, and then store it in your own data centers or in the cloud. The AWS services that you use determine how much configuration work you have to perform as part of your security responsibilities.

AWS products that fall into the well-understood category of Infrastructure-as-a-Service (IaaS)—such as Amazon EC2, Amazon VPC, and Amazon S3—are completely under your control and require you to perform all of the necessary security configuration and management tasks. For example, for EC2 instances, you're responsible for management of the guest OS (including updates and security patches), any application software or utilities you install on the instances, and the configuration of the AWS provided firewall (called a security group) on each instance. These are basically the same security tasks that you're used to performing no matter where your servers are located.

AWS managed services like Amazon RDS or Amazon Redshift provide all of the resources you need to perform a specific task—but without the configuration work that can come with them. With managed services, you don't have to worry about launching and maintaining instances, patching the guest OS or database, or replicating databases—AWS handles that for you. But as with all services, you should protect your AWS Account credentials and set up individual user accounts with Amazon Identity and Access Management (IAM) so that each of your users has their own credentials and you can implement segregation of duties. We also recommend using multi-factor authentication (MFA) with each account, requiring the use of SSL/TLS to communicate with your AWS resources, and setting up API/user activity logging with AWS CloudTrail. For more information about additional measures you can take, refer to the AWS Security Best Practices whitepaper and recommended reading on the AWS Security Learning webpage.

AWS Global Infrastructure Security

AWS operates the global cloud infrastructure that you use to provision a variety of basic computing resources such as processing and storage. The AWS global infrastructure includes the facilities, network, hardware, and operational software (e.g., host OS, virtualization software, etc.) that support the provisioning and use of these resources. The AWS global infrastructure is designed and managed according to security best practices as well as a variety of security compliance standards. As an AWS customer, you can be assured that you're building web architectures on top of some of the most secure computing infrastructure in the world.

Topics

AWS Compliance Program

AWS Compliance enables customers to understand the robust controls in place at AWS to maintain security and data protection in the cloud. As systems are built on top of AWS cloud infrastructure, compliance responsibilities are shared. By tying together governance-focused, audit friendly service features with applicable compliance or audit standards, AWS Compliance enablers build on traditional programs; helping customers to establish and operate in an AWS security control environment. The IT infrastructure that AWS provides to its customers is designed and managed in alignment with security best practices and a variety of IT security standards, including:

- SOC 1/SSAE 16/ISAE 3402 (formerly SAS 70)
- SOC 2
- SOC 3
- FISMA, DIACAP, and FedRAMP
- DOD CSM Levels 1-5
- PCI DSS Level 1
- ISO 9001 / ISO 27001 / ISO 27017 / ISO 27018

- ITAR
- FIPS 140-2
- MTCS Level 3
- HITRUST

In addition, the flexibility and control that the AWS platform provides allows customers to deploy solutions that meet several industry-specific standards, including:

- Criminal Justice Information Services (CJIS)
- Cloud Security Alliance (CSA)
- Family Educational Rights and Privacy Act (FERPA)
- Health Insurance Portability and Accountability Act (HIPAA)
- Motion Picture Association of America (MPAA)

AWS provides a wide range of information regarding its IT control environment to customers through white papers, reports, certifications, accreditations, and other third-party attestations. For more information, see AWS Compliance.

Physical and Environmental Security

AWS data centers are state of the art, utilizing innovative architectural and engineering approaches. Amazon has many years of experience in designing, constructing, and operating large-scale data centers. This experience has been applied to the AWS platform and infrastructure. AWS data centers are housed in facilities that are not branded as AWS facilities. Physical access is strictly controlled both at the perimeter and at building ingress points by professional security staff utilizing video surveillance, intrusion detection systems, and other electronic means.

Authorized staff must pass two-factor authentication a minimum of two times to access data center floors. All visitors are required to present identification and are signed in and continually escorted by authorized staff.

AWS only provides data center access and information to employees and contractors who have a legitimate business need for such privileges. When an employee no longer has a business need for these privileges, his or her access is immediately revoked, even if they continue to be an employee of Amazon or Amazon Web Services. All physical access to data centers by AWS employees is logged and audited routinely.

Fire Detection and Suppression

Automatic fire detection and suppression equipment has been installed to reduce risk. The fire detection system utilizes smoke detection sensors in all data center environments, mechanical and electrical infrastructure spaces, chiller rooms and generator equipment rooms. These areas are protected by either wet-pipe, double-interlocked pre-action, or gaseous sprinkler systems.

Power

The data center electrical power systems are designed to be fully redundant and maintainable without impact to operations, 24 hours a day, and seven days a week. Uninterruptible Power Supply (UPS) units provide back-up power in the event of an electrical failure for critical and essential loads in the facility. Data centers use generators to provide back-up power for the entire facility.

Climate and Temperature

Climate control is required to maintain a constant operating temperature for servers and other hardware, which prevents overheating and reduces the possibility of service outages. Data centers are conditioned to maintain atmospheric conditions at optimal levels. Personnel and systems monitor and control temperature and humidity at appropriate levels.

Management

AWS monitors electrical, mechanical, and life support systems and equipment so that any issues are immediately identified. Preventative maintenance is performed to maintain the continued operability of equipment.

Storage Device Decommissioning

When a storage device has reached the end of its useful life, AWS procedures include a decommissioning process that is designed to prevent customer data from being exposed to unauthorized individuals. AWS uses the techniques detailed in NIST 800-88 (“Guidelines for Media Sanitization”) as part of the decommissioning process.

Business Continuity Management

Amazon’s infrastructure has a high level of availability and provides customers the features to deploy a resilient IT architecture. AWS has designed its systems to tolerate system or hardware failures with minimal customer impact. Data center Business Continuity Management at AWS is under the direction of the Amazon Infrastructure Group.

Topics

Availability

Data centers are built in clusters in various global regions. All data centers are online and serving customers; no data center is “cold.” In case of failure, automated processes move customer data traffic away from the affected area. Core applications are deployed in an N+1 configuration, so that in the event of a data center failure, there is sufficient capacity to enable traffic to be load- balanced to the remaining sites.

AWS provides you with the flexibility to place instances and store data within multiple geographic regions as well as across multiple availability zones within each region. Each availability zone is designed as an independent failure zone.

This means that availability zones are physically separated within a typical metropolitan region and are located in lower risk flood plains (specific flood zone categorization varies by Region). In addition to discrete uninterruptable power supply (UPS) and onsite backup generation facilities, they are each fed via different grids from independent utilities to further reduce single points of failure. Availability zones are all redundantly connected to multiple tier-1 transit providers.

You should architect your AWS usage to take advantage of multiple regions and availability zones. Distributing applications across multiple availability zones provides the ability to remain resilient in the face of most failure modes, including natural disasters or system failures.

Incident Response

The Amazon Incident Management team employs industry-standard diagnostic procedures to drive resolution during business-impacting events. Staff operators provide 24x7x365 coverage to detect incidents and to manage the impact and resolution.

Company-Wide Executive Review

Amazon's Internal Audit group has recently reviewed the AWS services resiliency plans, which are also periodically reviewed by members of the Senior Executive management team and the Audit Committee of the Board of Directors.

Communication

AWS has implemented various methods of internal communication at a global level to help employees understand their individual roles and responsibilities and to communicate significant events in a timely manner. These methods include orientation and training programs for newly hired employees; regular management meetings for updates on business performance and other matters; and electronics means such as video conferencing, electronic mail messages, and the posting of information via the Amazon intranet.

AWS has also implemented various methods of external communication to support its customer base and the community. Mechanisms are in place to allow the customer support team to be notified of operational issues that impact the customer experience. A Service Health Dashboard is available and maintained by the customer support team to alert customers to any issues that may be of broad impact. The AWS Cloud Security Center is available to provide you with security and compliance details about AWS. You can also subscribe to AWS Support offerings that include direct communication with the customer support team and proactive alerts to any customer impacting issues.

Network Security

The AWS network has been architected to permit you to select the level of security and resiliency appropriate for your workload. To enable you to build geographically dispersed, fault-tolerant web architectures with cloud resources, AWS has implemented a world-class network infrastructure that is carefully monitored and managed.

Topics

Secure Network Architecture

Network devices, including firewall and other boundary devices, are in place to monitor and control communications at the external boundary of the network and at key internal boundaries within the

network. These boundary devices employ rule sets, access control lists (ACL), and configurations to enforce the flow of information to specific information system services.

ACLs, or traffic flow policies, are established on each managed interface, which manage and enforce the flow of traffic. ACL policies are approved by Amazon Information Security. These policies are automatically pushed using AWS's ACL- Manage tool, to help ensure these managed interfaces enforce the most up-to-date ACLs.

Secure Access Points

AWS has strategically placed a limited number of access points to the cloud to allow for a more comprehensive monitoring of inbound and outbound communications and network traffic. These customer access points are called API endpoints, and they allow secure HTTP access (HTTPS), which allows you to establish a secure communication session with your storage or compute instances within AWS. To support customers with FIPS cryptographic requirements, the SSL-terminating load balancers in AWS GovCloud (US) are FIPS 140-2-compliant.

In addition, AWS has implemented network devices that are dedicated to managing interfacing communications with Internet service providers (ISPs). AWS employs a redundant connection to more than one communication service at each Internet-facing edge of the AWS network. These connections each have dedicated network devices.

Transmission Protection

You can connect to an AWS access point via HTTP or HTTPS using Secure Sockets Layer (SSL), a cryptographic protocol that is designed to protect against eavesdropping, tampering, and message forgery.

For customers who require additional layers of network security, AWS offers the Amazon Virtual Private Cloud (VPC), which provides a private subnet within the AWS cloud, and the ability to use an IPsec Virtual Private Network (VPN) device to provide an encrypted tunnel between the Amazon VPC and your data center. For more information about VPC configuration options, see the Amazon Virtual Private Cloud (Amazon VPC) Security (p. 25) section.

Amazon Corporate Segregation

Logically, the AWS Production network is segregated from the Amazon Corporate network by means of a complex set of network security / segregation devices. AWS developers and administrators on the corporate network who need to access AWS cloud components in order to maintain them must explicitly request access through the AWS ticketing system. All requests are reviewed and approved by the applicable service owner.

Approved AWS personnel then connect to the AWS network through a bastion host that restricts access to network devices and other cloud components, logging all activity for security review. Access to bastion hosts require SSH public- key authentication for all user accounts on the host. For more information on AWS developer and administrator logical access, see AWS Access below.

Fault-Tolerant Design

Amazon's infrastructure has a high level of availability and provides you with the capability to deploy a resilient IT architecture. AWS has designed its systems to tolerate system or hardware failures with minimal customer impact.

Data centers are built in clusters in various global regions. All data centers are online and serving customers; no data center is "cold." In case of failure, automated processes move customer data traffic away from the affected area. Core applications are deployed in an N+1 configuration, so that in the event

of a data center failure, there is sufficient capacity to enable traffic to be load-balanced to the remaining sites.

AWS provides you with the flexibility to place instances and store data within multiple geographic regions as well as across multiple availability zones within each region. Each availability zone is designed as an independent failure zone. This means that availability zones are physically separated within a typical metropolitan region and are located in lower risk flood plains (specific flood zone categorization varies by region). In addition to utilizing discrete uninterruptable power supply (UPS) and onsite backup generators, they are each fed via different grids from independent utilities to further reduce single points of failure. Availability zones are all redundantly connected to multiple tier-1 transit providers.

You should architect your AWS usage to take advantage of multiple regions and availability zones. Distributing applications across multiple availability zones provides the ability to remain resilient in the face of most failure scenarios, including natural disasters or system failures. However, you should be aware of location-dependent privacy and compliance requirements, such as the EU Data Privacy Directive. Data is not replicated between regions unless proactively done so by the customer, thus allowing customers with these types of data placement and privacy requirements the ability to establish compliant environments. It should be noted that all communications between regions is across public internet infrastructure; therefore, appropriate encryption methods should be used to protect sensitive data.

Data centers are built in clusters in various global regions, including: US East (Northern Virginia), US West (Oregon), US West (Northern California), AWS GovCloud (US) (Oregon), EU (Frankfurt), EU (Ireland), Asia Pacific (Seoul) Asia Pacific (Singapore), Asia Pacific (Tokyo), Asia Pacific (Sydney), China (Beijing), and South America (Sao Paulo). For a complete list of AWS Regions, see the AWS Global Infrastructure page.

AWS GovCloud (US) is an isolated AWS Region designed to allow US government agencies and customers to move workloads into the cloud by helping them meet certain regulatory and compliance requirements. The AWS GovCloud (US) framework allows US government agencies and their contractors to comply with U.S. International Traffic in Arms Regulations (ITAR) regulations as well as the Federal Risk and Authorization Management Program (FedRAMP) requirements. AWS GovCloud (US) has received an Agency Authorization to Operate (ATO) from the US Department of Health and Human Services (HHS) utilizing a FedRAMP accredited Third Party Assessment Organization (3PAO) for several AWS services.

The AWS GovCloud (US) Region provides the same fault-tolerant design as other regions, with two Availability Zones. In addition, the AWS GovCloud (US) region is a mandatory AWS Virtual Private Cloud (VPC) service by default to create an isolated portion of the AWS cloud and launch Amazon EC2 instances that have private (RFC 1918) addresses. For more information, see AWS GovCloud (US).

Network Monitoring and Protection

AWS uses a wide variety of automated monitoring systems to provide a high level of service performance and availability. AWS monitoring tools are designed to detect unusual or unauthorized activities and conditions at ingress and egress communication points. These tools monitor server and network usage, port scanning activities, application usage, and unauthorized intrusion attempts. The tools have the ability to set custom performance metrics thresholds for unusual activity.

Systems within AWS are extensively instrumented to monitor key operational metrics. Alarms are configured to automatically notify operations and management personnel when early warning thresholds are crossed on key operational metrics. An on-call schedule is used so personnel are always available to respond to operational issues. This includes a pager system so alarms are quickly and reliably communicated to operations personnel.

Documentation is maintained to aid and inform operations personnel in handling incidents or issues. If the resolution of an issue requires collaboration, a conferencing system is used which supports communication and logging capabilities. Trained call leaders facilitate communication and progress during the handling of operational issues that require collaboration. Post-mortems are convened after any significant operational issue, regardless of external impact, and Cause of Error (COE) documents are

drafted so the root cause is captured and preventative actions are taken in the future. Implementation of the preventative measures is tracked during weekly operations meetings.

AWS Access

The AWS Production network is segregated from the Amazon Corporate network and requires a separate set of credentials for logical access. The Amazon Corporate network relies on user IDs, passwords, and Kerberos, whereas the AWS Production network requires SSH public-key authentication through a bastion host.

AWS developers and administrators on the Amazon Corporate network who need to access AWS cloud components must explicitly request access through the AWS access management system. All requests are reviewed and approved by the appropriate owner or manager.

Account Review and Audit

Accounts are reviewed every 90 days; explicit re-approval is required or access to the resource is automatically revoked. Access is also automatically revoked when an employee's record is terminated in Amazon's Human Resources system. Windows and UNIX accounts are disabled and Amazon's permission management system removes the user from all systems.

Requests for changes in access are captured in the Amazon permissions management tool audit log. When changes in an employee's job function occur, continued access must be explicitly approved to the resource or it will be automatically revoked.

Background Checks

AWS has established formal policies and procedures to delineate the minimum standards for logical access to AWS platform and infrastructure hosts. AWS conducts criminal background checks, as permitted by law, as part of pre- employment screening practices for employees and commensurate with the employee's position and level of access. The policies also identify functional responsibilities for the administration of logical access and security.

Credentials Policy

AWS Security has established a credentials policy with required configurations and expiration intervals. Passwords must be complex and are forced to be changed every 90 days.

Secure Design Principles

The AWS development process follows secure software development best practices, which include formal design reviews by the AWS Security Team, threat modeling, and completion of a risk assessment. Static code analysis tools are run as a part of the standard build process, and all deployed software undergoes recurring penetration testing performed by carefully selected industry experts. Our security risk assessment reviews begin during the design phase and the engagement lasts through launch to ongoing operations.

Change Management

Routine, emergency, and configuration changes to existing AWS infrastructure are authorized, logged, tested, approved, and documented in accordance with industry norms for similar systems. Updates to the AWS infrastructure are done to minimize any impact on the customer and their use of the services. AWS

will communicate with customers, either via email, or through the AWS Service Health Dashboard when service use is likely to be adversely affected.

Topics

Software

AWS applies a systematic approach to managing change so that changes to customer-impacting services are thoroughly reviewed, tested, approved, and well-communicated. The AWS change management process is designed to avoid unintended service disruptions and to maintain the integrity of service to the customer. Changes deployed into production environments are:

- **Reviewed** – Peer reviews of the technical aspects of a change are required.
- **Tested** – Changes being applied are tested to help ensure they will behave as expected and not adversely impact performance.
- **Approved** – All changes must be authorized in order to provide appropriate oversight and understanding of business impact.

Changes are typically pushed into production in a phased deployment starting with lowest impact areas. Deployments are tested on a single system and closely monitored so impacts can be evaluated. Service owners have a number of configurable metrics that measure the health of the service's upstream dependencies. These metrics are closely monitored with thresholds and alarming in place. Rollback procedures are documented in the Change Management (CM) ticket.

When possible, changes are scheduled during regular change windows. Emergency changes to production systems that require deviations from standard change management procedures are associated with an incident and are logged and approved as appropriate.

Periodically, AWS performs self-audits of changes to key services to monitor quality, maintain high standards, and facilitate continuous improvement of the change management process. Any exceptions are analyzed to determine the root cause, and appropriate actions are taken to bring the change into compliance or roll back the change if necessary. Actions are then taken to address and remediate the process or people issue.

Infrastructure

Amazon's Corporate Applications team develops and manages software to automate IT processes for UNIX/Linux hosts in the areas of third-party software delivery, internally developed software, and configuration management. The Infrastructure team maintains and operates a UNIX/Linux configuration management framework to address hardware scalability, availability, auditing, and security management. By centrally managing hosts through the use of automated processes that manage change, Amazon is able to achieve its goals of high availability, repeatability, scalability, security, and disaster recovery.

Systems and network engineers monitor the status of these automated tools on a continuous basis, reviewing reports to respond to hosts that fail to obtain or update their configuration and software.

Internally developed configuration management software is installed when new hardware is provisioned. These tools are run on all UNIX hosts to validate that they are configured and that software is installed in compliance with standards determined by the role assigned to the host. This configuration management software also helps to regularly update packages that are already installed on the host. Only approved personnel enabled through the permissions service may log in to the central configuration management servers.

AWS Account Security Features

AWS provides a variety of tools and features that you can use to keep your AWS Account and resources safe from unauthorized use. This includes credentials for access control, HTTPS endpoints for encrypted data transmission, the creation of separate IAM user accounts, user activity logging for security monitoring, and Trusted Advisor security checks. You can take advantage of all of these security tools no matter which AWS services you select.

Topics

AWS Credentials

To help ensure that only authorized users and processes access your AWS Account and resources, AWS uses several types of credentials for authentication. These include passwords, cryptographic keys, digital signatures, and certificates. We also provide the option of requiring multi-factor authentication (MFA) to log into your AWS Account or IAM user accounts. The following table highlights the various AWS credentials and their uses.

Table 1: Credential types and uses

Credential Type	Use	Description
Passwords	AWS root account or IAM user account login to the AWS Management Console	A string of characters used to log into your AWS account or IAM account. AWS passwords must be a minimum of 6 characters and may be up to 128 characters.
Multi-Factor Authentication (MFA)	AWS root account or IAM user account login to the AWS Management Console	A six-digit single-use code that is required in addition to your password to log in to your AWS Account or IAM user account.
Access Keys	Digitally signed requests to AWS APIs (using the AWS SDK, CLI, or REST/Query APIs)	Includes an access key ID and a secret access key. You use access keys to digitally sign programmatic requests that you make to AWS.
Key Pairs	SSH login to EC2 instances CloudFront signed URLs	A key pair is required to connect to an EC2 instance launched from a public AMI. The supported lengths are 1024, 2048, and 4096. If you connect using SSH while using the EC2 Instance Connect API,

Credential Type	Use	Description
		the supported lengths are 2048 and 4096. You can have a key pair generated automatically for you when you launch the instance or you can upload your own.
X.509 Certificates	Digitally signed SOAP requests to AWS APIs SSL server certificates for HTTPS	X.509 certificates are only used to sign SOAP-based requests (currently used only with Amazon S3). You can have AWS create an X.509 certificate and private key that you can download, or you can upload your own certificate by using the Security Credentials page.

You can download a Credential Report for your account at any time from the Security Credentials page. This report lists all of your account's users and the status of their credentials—whether they use a password, whether their password expires and must be changed regularly, the last time they changed their password, the last time they rotated their access keys, and whether they have MFA enabled.

For security reasons, if your credentials have been lost or forgotten, you cannot recover them or re-download them. However, you can create new credentials and then disable or delete the old set of credentials.

In fact, AWS recommends that you change (rotate) your access keys and certificates on a regular basis. To help you do this without potential impact to your application's availability, AWS supports multiple concurrent access keys and certificates. With this feature, you can rotate keys and certificates into and out of operation on a regular basis without any downtime to your application. This can help to mitigate risk from lost or compromised access keys or certificates. The AWS IAM API enables you to rotate the access keys of your AWS Account as well as for IAM user accounts.

Passwords

Passwords are required to access your AWS Account, individual IAM user accounts, AWS Discussion Forums, and the AWS Support Center. You specify the password when you first create the account, and you can change it at any time by going to the Security Credentials page. AWS passwords can be up to 128 characters long and contain special characters, so we encourage you to create a strong password that cannot be easily guessed.

You can set a password policy for your IAM user accounts to ensure that strong passwords are used and that they are changed often. A password policy is a set of rules that define the type of password an IAM user can set. For more information about password policies, see Managing Passwords for IAM Users.

AWS Multi-Factor Authentication (MFA)

AWS Multi-Factor Authentication (MFA) is an additional layer of security for accessing AWS services. When you enable this optional feature, you must provide a six-digit single-use code in addition to your standard user name and password credentials before access is granted to your AWS Account settings or AWS services and resources. You get this single-use code from an authentication device that you keep in your physical possession. This is called multi-factor authentication because more than one authentication factor is checked before access is granted: a password (something you know) and the precise code from your authentication device (something you have). You can enable MFA devices for

your AWS Account as well as for the users you have created under your AWS Account with AWS IAM. In addition, you add MFA protection for access across AWS Accounts, for when you want to allow a user you've created under one AWS Account to use an IAM role to access resources under another AWS Account. You can require the user to use MFA before assuming the role as an additional layer of security.

AWS MFA supports the use of both hardware tokens and virtual MFA devices. Virtual MFA devices use the same protocols as the physical MFA devices, but can run on any mobile hardware device, including a smartphone. A virtual MFA device uses a software application that generates six-digit authentication codes that are compatible with the Time-Based One-Time Password (TOTP) standard, as described in RFC 6238. Most virtual MFA applications allow you to host more than one virtual MFA device, which makes them more convenient than hardware MFA devices. However, you should be aware that because a virtual MFA might be run on a less secure device such as a smartphone, a virtual MFA might not provide the same level of security as a hardware MFA device.

You can also enforce MFA authentication for AWS service APIs in order to provide an extra layer of protection over powerful or privileged actions such as terminating Amazon EC2 instances or reading sensitive data stored in Amazon S3. You do this by adding an MFA-authentication requirement to an IAM access policy. You can attach these access policies to IAM users, IAM groups, or resources that support Access Control Lists (ACLs) like Amazon S3 buckets, SQS queues, and SNS topics.

It is easy to obtain hardware tokens from a participating third-party provider or virtual MFA applications from an AppStore and to set it up for use via the AWS website. More information is available at AWS Multi-Factor Authentication (MFA).

Access Keys

AWS requires that all API requests be signed—that is, they must include a digital signature that AWS can use to verify the identity of the requestor. You calculate the digital signature using a cryptographic hash function. The input to the hash function in this case includes the text of your request and your secret access key. If you use any of the AWS SDKs to generate requests, the digital signature calculation is done for you; otherwise, you can have your application calculate it and include it in your REST or Query requests by following the directions in Making Requests Using the AWS SDKs.

Not only does the signing process help protect message integrity by preventing tampering with the request while it is in transit, it also helps protect against potential replay attacks. A request must reach AWS within 15 minutes of the time stamp in the request. Otherwise, AWS denies the request.

The most recent version of the digital signature calculation process is Signature Version 4, which calculates the signature using the HMAC-SHA256 protocol. Version 4 provides an additional measure of protection over previous versions by requiring that you sign the message using a key that is derived from your secret access key rather than using the secret access key itself. In addition, you derive the signing key based on credential scope, which facilitates cryptographic isolation of the signing key.

Because access keys can be misused if they fall into the wrong hands, we encourage you to save them in a safe place and not embed them in your code. For customers with large fleets of elastically scaling EC2 instances, the use of IAM roles can be a more secure and convenient way to manage the distribution of access keys. IAM roles provide temporary credentials, which not only get automatically loaded to the target instance, but are also automatically rotated multiple times a day.

Key Pairs

Amazon EC2 instances created from a public AMI use a public/private key pair rather than a password for signing in via Secure Shell (SSH).The public key is embedded in your instance, and you use the private key to sign in securely without a password. After you create your own AMIs, you can choose other mechanisms to securely log in to your new instances.

You can have a key pair generated automatically for you when you launch the instance or you can upload your own. Save the private key in a safe place on your system, and record the location where you saved it.

For Amazon CloudFront, you use key pairs to create signed URLs for private content, such as when you want to distribute restricted content that someone paid for. You create Amazon CloudFront key pairs by using the Security Credentials page. CloudFront key pairs can be created only by the root account and cannot be created by IAM users.

X.509 Certificates

X.509 certificates are used to sign SOAP-based requests. X.509 certificates contain a public key and additional metadata (like an expiration date that AWS verifies when you upload the certificate), and is associated with a private key. When you create a request, you create a digital signature with your private key and then include that signature in the request, along with your certificate. AWS verifies that you're the sender by decrypting the signature with the public key that is in your certificate. AWS also verifies that the certificate you sent matches the certificate that you uploaded to AWS.

For your AWS Account, you can have AWS create an X.509 certificate and private key that you can download, or you can upload your own certificate by using the Security Credentials page. For IAM users, you must create the X.509 certificate (signing certificate) by using third-party software. In contrast with root account credentials, AWS cannot create an X.509 certificate for IAM users. After you create the certificate, you attach it to an IAM user by using IAM.

In addition to SOAP requests, X.509 certificates are used as SSL/TLS server certificates for customers who want to use HTTPS to encrypt their transmissions. To use them for HTTPS, you can use an open-source tool like OpenSSL to create a unique private key. You'll need the private key to create the Certificate Signing Request (CSR) that you submit to a certificate authority (CA) to obtain the server certificate. You'll then use the AWS CLI to upload the certificate, private key, and certificate chain to IAM.

You'll also need an X.509 certificate to create a customized Linux AMI for EC2 instances. The certificate is only required to create an instance-backed AMI (as opposed to an EBS-backed AMI). You can have AWS create an X.509 certificate and private key that you can download, or you can upload your own certificate by using the Security Credentials page.

Individual User Accounts

AWS provides a centralized mechanism called AWS Identity and Access Management (IAM) for creating and managing individual users within your AWS Account. A user can be any individual, system, or application that interacts with AWS resources, either programmatically or through the AWS Management Console or AWS Command Line Interface (CLI). Each user has a unique name within the AWS Account, and a unique set of security credentials not shared with other users. AWS IAM eliminates the need to share passwords or keys, and enables you to minimize the use of your AWS Account credentials.

With IAM, you define policies that control which AWS services your users can access and what they can do with them. You can grant users only the minimum permissions they need to perform their jobs. See the AWS Identity and Access Management (AWS IAM) (p. 56) section for more information.

Secure HTTPS Access Points

For greater communication security when accessing AWS resources, you should use HTTPS instead of HTTP for data transmissions. HTTPS uses the SSL/TLS protocol, which uses public-key cryptography to prevent eavesdropping, tampering, and forgery. All AWS services provide secure customer access points (also called API endpoints) that allow you to establish secure HTTPS communication sessions.

Several services also now offer more advanced cipher suites that use the Elliptic Curve Diffie-Hellman Ephemeral (ECDHE) protocol. ECDHE allows SSL/TLS clients to provide Perfect Forward Secrecy, which

uses session keys that are ephemeral and not stored anywhere. This helps prevent the decoding of captured data by unauthorized third parties, even if the secret long-term key itself is compromised.

Security Logs

As important as credentials and encrypted endpoints are for preventing security problems, logs are just as crucial for understanding events after a problem has occurred. And to be effective as a security tool, a log must include not just a list of what happened and when, but also identify the source. To help you with your after-the-fact investigations and near-real time intrusion detection, AWS CloudTrail provides a log of events within your account. For each event, you can see what service was accessed, what action was performed, and who made the request.

CloudTrail captures API calls, as well as other things such as console sign-in events. Once you have enabled CloudTrail, event logs are delivered about every 5 minutes. You can configure CloudTrail so that it aggregates log files from multiple regions and/or accounts into a single Amazon S3 bucket. By default, a single trail will record and deliver events in all current and future regions. In addition to S3, you can send events to CloudWatch Logs, for custom metrics and alarming, or you can upload the logs to your favorite log management and analysis solutions to perform security analysis and detect user behavior patterns. For rapid response, you can create CloudWatch Events rules to take timely action to specific events. By default, log files are stored securely in Amazon S3, but you can also archive them to Amazon S3 Glacier to help meet audit and compliance requirements.

In addition to CloudTrail's user activity logs, you can use the Amazon CloudWatch Logs feature to collect and monitor system, application, and custom log files from your EC2 instances and other sources in near-real time. For example, you can monitor your web server's log files for invalid user messages to detect unauthorized login attempts to your guest OS.

AWS Trusted Advisor Security Checks

The AWS Trusted Advisor customer support service not only monitors for cloud performance and resiliency, but also cloud security. Trusted Advisor inspects your AWS environment and makes recommendations when opportunities may exist to save money, improve system performance, or close security gaps. It provides alerts on several of the most common security misconfigurations that can occur, including leaving certain ports open that make you vulnerable to hacking and unauthorized access, neglecting to create IAM accounts for your internal users, allowing public access to Amazon S3 buckets, not turning on user activity logging (AWS CloudTrail), or not using MFA on your root AWS Account. You also have the option for a Security contact at your organization to automatically receive a weekly email with an updated status of your Trusted Advisor security checks.

The AWS Trusted Advisor service provides four checks at no additional charge to all users, including three important security checks: specific ports unrestricted, IAM use, and MFA on root account. When you sign up for Business- or Enterprise-level AWS Support, you receive full access to all Trusted Advisor checks.

AWS Config Security Checks

AWS Config is a continuous monitoring and assessment service that records changes to the configuration of your AWS resources. You can view the current and historic configurations of a resource and use this information to troubleshoot outages, conduct security attack analysis, and much more. You can view the configuration at any point in time and use that information to re-configure your resources and bring them into a steady state during an outage situation.

Using AWS Config Rules, you can run continuous assessment checks on your resources to verify that they comply with your own security policies, industry best practices, and compliance regimes such as

PCI/HIPAA. For example, AWS Config provides a managed AWS Config Rules to ensure that encryption is turned on for all EBS volumes in your account. You can also write a custom AWS Config Rule to essentially “codify” your own corporate security policies. AWS Config alerts you in real time when a resource is misconfigured, or when a resource violates a particular security policy.

AWS Service-Specific Security

Not only is security built into every layer of the AWS infrastructure, but also into each of the services available on that infrastructure. AWS services are architected to work efficiently and securely with all AWS networks and platforms. Each service provides extensive security features to enable you to protect sensitive data and applications.

Compute Services

Amazon Web Services provides a variety of cloud-based computing services that include a wide selection of compute instances that can scale up and down automatically to meet the needs of your application or enterprise.

Topics

Amazon Elastic Compute Cloud (Amazon EC2) Security

Amazon Elastic Compute Cloud (Amazon EC2) is a key component in Amazon's Infrastructure-as-a-Service (IaaS), providing resizable computing capacity using server instances in AWS's data centers. Amazon EC2 is designed to make web- scale computing easier by enabling you to obtain and configure capacity with minimal friction. You create and launch instances, which are collections of platform hardware and software.

Topics

Multiple Levels of Security

Security within Amazon EC2 is provided on multiple levels: the operating system (OS) of the host platform, the virtual instance OS or guest OS, a firewall, and signed API calls. Each of these items builds on the capabilities of the others. The goal is to prevent data contained within Amazon EC2 from being intercepted by unauthorized systems or users and to provide Amazon EC2 instances themselves that are as secure as possible without sacrificing the flexibility in configuration that customers demand.

Hypervisor

Amazon EC2 currently utilizes a highly customized version of the Xen hypervisor, taking advantage of paravirtualization (in the case of Linux guests). Because para-virtualized guests rely on the hypervisor

to provide support for operations that normally require privileged access, the guest OS has no elevated access to the CPU. The CPU provides four separate privilege modes: 0-3, called rings. Ring 0 is the most privileged and 3 the least. The host OS executes in Ring 0. However, rather than executing in Ring 0 as most operating systems do, the guest OS runs in a lesser-privileged Ring 1 and applications in the least privileged Ring 3. This explicit virtualization of the physical resources leads to a clear separation between guest and hypervisor, resulting in additional security separation between the two.

Traditionally, hypervisors protect the physical hardware and bios, virtualize the CPU, storage, networking, and provide a rich set of management capabilities. With the Nitro System, we are able to break apart those functions, offload them to dedicated hardware and software, and reduce costs by delivering all of the resources of a server to your instances.

The Nitro Hypervisor provides consistent performance and increased compute and memory resources for EC2 virtualized instances by removing host system software components. It allows AWS to offer larger instance sizes (like c5.18xlarge) that provide practically all of the resources from the server to customers. Previously, C3 and C4 instances each eliminated software components by moving VPC and EBS functionality to hardware designed and built by AWS. This hardware enables the Nitro Hypervisor to be very small and uninvolved in data processing tasks for networking and storage.

Nevertheless, as AWS expands its global cloud infrastructure, Amazon EC2's use of its Xen-based hypervisor will also continue to grow. Xen will remain a core component of EC2 instances for the foreseeable future.

Instance Isolation

Different instances running on the same physical machine are isolated from each other via the Xen hypervisor. Amazon is active in the Xen community, which provides awareness of the latest developments. In addition, the AWS firewall resides within the hypervisor layer, between the physical network interface and the instance's virtual interface. All packets must pass through this layer, thus an instance's neighbors have no more access to that instance than any other host on the Internet and can be treated as if they are on separate physical hosts. The physical RAM is separated using similar mechanisms.

Customer instances have no access to raw disk devices, but instead are presented with virtualized disks. The AWS proprietary disk virtualization layer automatically resets every block of storage used by the customer, so that one customer's data is never unintentionally exposed to another. In addition, memory allocated to guests is scrubbed (set to zero) by the hypervisor when it is unallocated to a guest. The memory is not returned to the pool of free memory available for new allocations until the memory scrubbing is complete.

AWS recommends customers further protect their data using appropriate means. One common solution is to run an encrypted file system on top of the virtualized disk device.

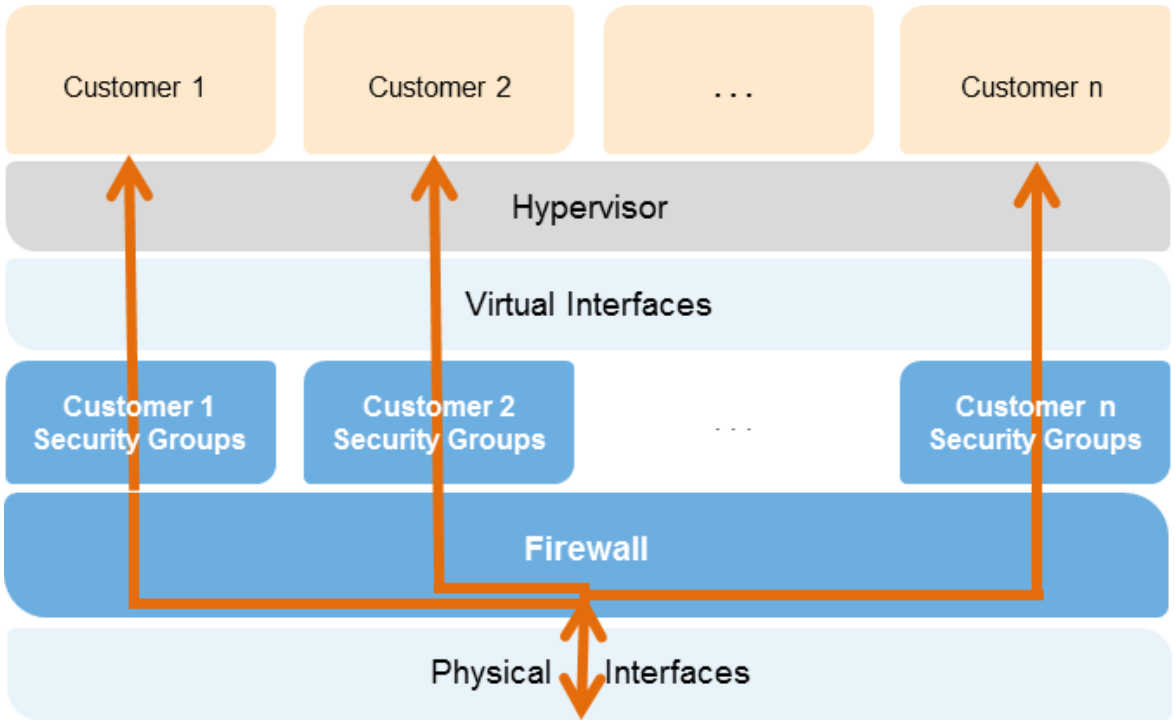

Figure 2: Amazon EC2 multiple layers of security

Host Operating System: Administrators with a business need to access the management plane are required to use multi- factor authentication to gain access to purpose-built administration hosts. These administrative hosts are systems that are specifically designed, built, configured, and hardened to protect the management plane of the cloud. All such access is logged and audited. When an employee no longer has a business need to access the management plane, the privileges and access to these hosts and relevant systems can be revoked.

Guest Operating System: Virtual instances are completely controlled by you, the customer. You have full root access or administrative control over accounts, services, and applications. AWS does not have any access rights to your instances or the guest OS. AWS recommends a base set of security best practices to include disabling password-only access to your guests, and utilizing some form of multi-factor authentication to gain access to your instances (or at a minimum certificate-based SSH Version 2 access). Additionally, you should employ a privilege escalation mechanism with logging on a per-user basis. For example, if the guest OS is Linux, after hardening your instance you should utilize certificate- based SSHv2 to access the virtual instance, disable remote root login, use command-line logging, and use 'sudo' for privilege escalation.

You should generate your own key pairs in order to guarantee that they are unique, and not shared with other customers or with AWS.

AWS also supports the use of the Secure Shell (SSH) network protocol to enable you to log in securely to your UNIX/Linux EC2 instances. Authentication for SSH used with AWS is via a public/private key pair to reduce the risk of unauthorized access to your instance. You can also connect remotely to your Windows instances using Remote Desktop Protocol (RDP) by utilizing an RDP certificate generated for your instance.

You also control the updating and patching of your guest OS, including security updates. Amazon-provided Windows and Linux-based AMIs are updated regularly with the latest patches, so if you do not need to preserve data or customizations on your running Amazon AMI instances, you can simply relaunch new instances with the latest updated AMI. In addition, updates are provided for the Amazon Linux AMI via the Amazon Linux yum repositories.

Firewall: Amazon EC2 provides a complete firewall solution; this mandatory inbound firewall is configured in a default deny-all mode and Amazon EC2 customers must explicitly open the ports needed to allow inbound traffic. The traffic may be restricted by protocol, by service port, as well as by source IP address (individual IP or Classless Inter-Domain Routing (CIDR) block).

The firewall can be configured in groups permitting different classes of instances to have different rules. Consider, for example, the case of a traditional three-tiered web application. The group for the web servers would have port 80 (HTTP) and/or port 443 (HTTPS) open to the Internet. The group for the application servers would have port 8000 (application specific) accessible only to the web server group. The group for the database servers would have port 3306 (MySQL) open only to the application server group. All three groups would permit administrative access on port 22 (SSH), but only from the customer's corporate network. Highly secure applications can be deployed using this expressive mechanism. See the following figure.

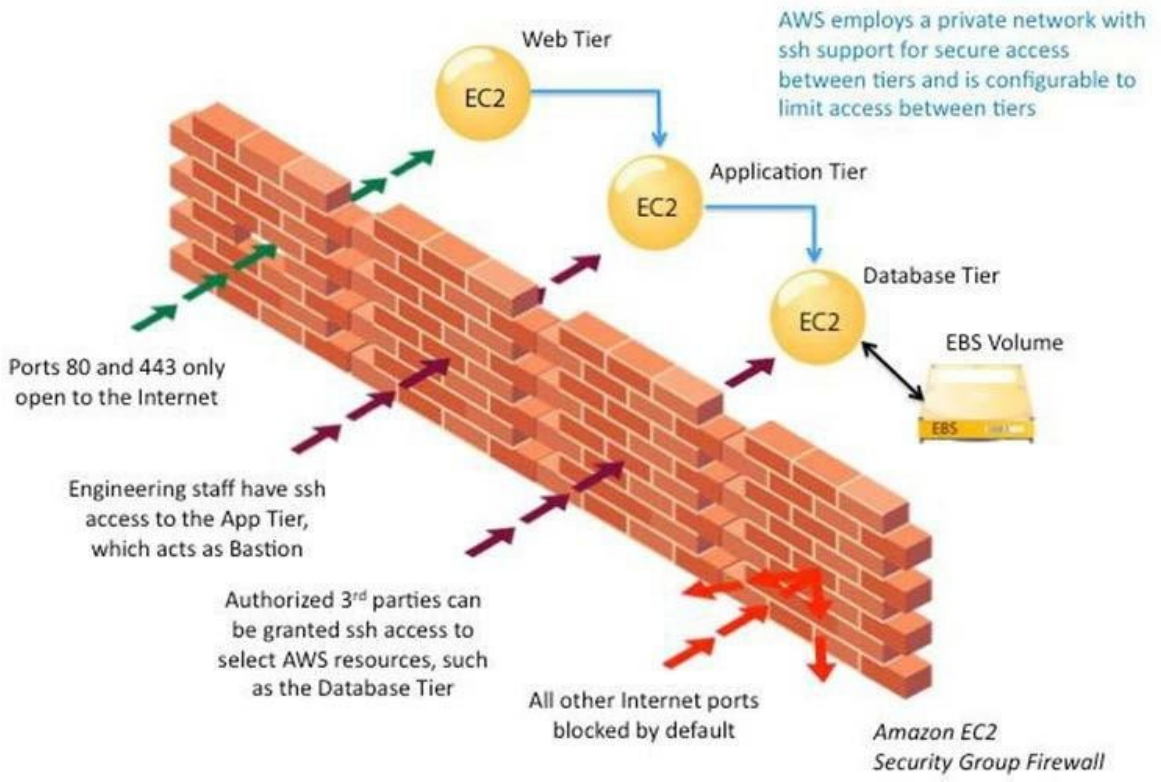

Figure 3: Amazon EC2 security group firewall

The firewall isn't controlled through the guest OS; rather it requires your X.509 certificate and key to authorize changes, thus adding an extra layer of security. AWS supports the ability to grant granular access to different administrative functions on the instances and the firewall, therefore enabling you to implement additional security through separation of duties. The level of security afforded by the firewall is a function of which ports you open, and for what duration and purpose. The default state is to deny all incoming traffic, and you should plan carefully what you will open when building and securing your applications. Well- informed traffic management and security design are still required on a per- instance basis. AWS further encourages you to apply additional per- instance filters with host-based firewalls such as IPtables or the Windows Firewall and VPNs. This can restrict both inbound and outbound traffic.

API Access: API calls to launch and terminate instances, change firewall parameters, and perform other functions are all signed by your Amazon Secret Access Key, which could be either the AWS Accounts Secret Access Key or the Secret Access key of a user created with AWS IAM. Without access to your Secret Access Key, Amazon EC2 API calls cannot be made on your behalf. In addition, API calls can be encrypted with SSL to maintain confidentiality. Amazon recommends always using SSL-protected API endpoints.

Permissions: AWS IAM also enables you to further control what APIs a user has permissions to call.

Elastic Block Storage (Amazon EBS) Security

Amazon Elastic Block Storage (Amazon EBS) allows you to create storage volumes from 1 GB to 16 TB that can be mounted as devices by Amazon EC2 instances.

Storage volumes behave like raw, unformatted block devices, with user supplied device names and a block device interface. You can create a file system on top of Amazon EBS volumes, or use them in any other way you would use a block device (like a hard drive). Amazon EBS volume access is restricted to the AWS Account that created the volume, and to the users under the AWS Account created with AWS IAM if the user has been granted access to the EBS operations, thus denying all other AWS Accounts and users the permission to view or access the volume.

Data stored in Amazon EBS volumes is redundantly stored in multiple physical locations as part of normal operation of those services and at no additional charge. However, Amazon EBS replication is stored within the same availability zone, not across multiple zones; therefore, it is highly recommended that you conduct regular snapshots to Amazon S3 for long-term data durability. For customers who have architected complex transactional databases using EBS, it is recommended that backups to Amazon S3 be performed through the database management system so that distributed transactions and logs can be checkpointed. AWS does not perform backups of data that are maintained on virtual disks attached to running instances on Amazon EC2.

You can make Amazon EBS volume snapshots publicly available to other AWS Accounts to use as the basis for creating your own volumes. Sharing Amazon EBS volume snapshots does not provide other AWS Accounts with the permission to alter or delete the original snapshot, as that right is explicitly reserved for the AWS Account that created the volume. An EBS snapshot is a block-level view of an entire EBS volume. Note that data that is not visible through the file system on the volume, such as files that have been deleted, may be present in the EBS snapshot. If you want to create shared snapshots, you should do so carefully. If a volume has held sensitive data or has had files deleted from it, a new EBS volume should be created. The data to be contained in the shared snapshot should be copied to the new volume, and the snapshot created from the new volume.

Amazon EBS volumes are presented to you as raw unformatted block devices that have been wiped prior to being made available for use. Wiping occurs immediately before reuse so that you can be assured that the wipe process completed. If you have procedures requiring that all data be wiped via a specific method, such as those detailed in NIST 800-88 (“Guidelines for Media Sanitization”), you have the ability to do so on Amazon EBS. You should conduct a specialized wipe procedure prior to deleting the volume for compliance with your established requirements.

Encryption of sensitive data is generally a good security practice, and AWS provides the ability to encrypt EBS volumes and their snapshots with AES-256. The encryption occurs on the servers that host the EC2 instances, providing encryption of data as it moves between EC2 instances and EBS storage. In order to be able to do this efficiently and with low latency, the EBS encryption feature is only available on EC2's more powerful instance types (e.g., M3, C3, R3, G2).

Auto Scaling Security

Auto Scaling allows you to automatically scale your Amazon EC2 capacity up or down according to conditions you define, so that the number of Amazon EC2 instances you are using scales up seamlessly during demand spikes to maintain performance, and scales down automatically during demand lulls to minimize costs.

Like all AWS services, Auto Scaling requires that every request made to its control API be authenticated so only authenticated users can access and manage Auto Scaling. Requests are signed with an HMAC-SHA1 signature calculated from the request and the user’s private key. However, getting credentials out to new EC2 instances launched with Auto Scaling can be challenging for large or elastically scaling fleets. To simplify this process, you can use roles within IAM, so that any new instances launched with a role will be given credentials automatically. When you launch an EC2 instance with an IAM role, temporary

AWS security credentials with permissions specified by the role are securely provisioned to the instance and are made available to your application via the Amazon EC2 Instance Metadata Service. The Metadata Service makes new temporary security credentials available prior to the expiration of the current active credentials, so that valid credentials are always available on the instance. In addition, the temporary security credentials are automatically rotated multiple times per day, providing enhanced security.

You can further control access to Auto Scaling by creating users under your AWS Account using AWS IAM, and controlling what Auto Scaling APIs these users have permission to call. For more information about using roles when launching instances, see Identity and Access Management for Amazon EC2.

Networking Services

Amazon Web Services provides a range of networking services that enable you to create a logically isolated network that you define, establish a private network connection to the AWS cloud, use a highly available and scalable DNS service and deliver content to your end users with low latency at high data transfer speeds with a content delivery web service.

Topics

Elastic Load Balancing Security

Elastic Load Balancing is used to manage traffic on a fleet of Amazon EC2 instances, distributing traffic to instances across all availability zones within a region. Elastic Load Balancing has all the advantages of an on-premises load balancer, plus several security benefits:

- Takes over the encryption and decryption work from the Amazon EC2 instances and manages it centrally on the load balancer
- Offers clients a single point of contact, and can also serve as the first line of defense against attacks on your network
- When used in an Amazon VPC, supports creation and management of security groups associated with your Elastic Load Balancing to provide additional networking and security options
- Supports end-to-end traffic encryption using TLS (previously SSL) on those networks that use secure HTTP (HTTPS) connections. When TLS is used, the TLS server certificate used to terminate client connections can be managed centrally on the load balancer, rather than on every individual instance.

HTTPS/TLS uses a long-term secret key to generate a short-term session key to be used between the server and the browser to create the ciphered (encrypted) message. Elastic Load Balancing configures your load balancer with a pre-defined cipher set that is used for TLS negotiation when a connection is established between a client and your load balancer. The pre-defined cipher set provides compatibility with a broad range of clients and uses strong cryptographic algorithms. However, some customers may have requirements for allowing only specific ciphers and protocols (such as PCI, SOX, etc.) from clients to ensure that standards are met. In these cases, Elastic Load Balancing provides options for selecting different configurations for TLS protocols and ciphers. You can choose to enable or disable the ciphers depending on your specific requirements.

To help ensure the use of newer and stronger cipher suites when establishing a secure connection, you can configure the load balancer to have the final say in the cipher suite selection during the client-server

negotiation. When the Server Order Preference option is selected, the load balancer selects a cipher suite based on the server's prioritization of cipher suites rather than the client's. This gives you more control over the level of security that clients use to connect to your load balancer.

For even greater communication privacy, Elastic Load Balancing allows the use of Perfect Forward Secrecy, which uses session keys that are ephemeral and not stored anywhere. This prevents the decoding of captured data, even if the secret long-term key itself is compromised.

Elastic Load Balancing allows you to identify the originating IP address of a client connecting to your servers, whether you're using HTTPS or TCP load balancing. Typically, client connection information, such as IP address and port, is lost when requests are proxied through a load balancer. This is because the load balancer sends requests to the server on behalf of the client, making your load balancer appear as though it is the requesting client. Having the originating client IP address is useful if you need more information about visitors to your applications in order to gather connection statistics, analyze traffic logs, or manage whitelists of IP addresses.

Elastic Load Balancing access logs contain information about each HTTP and TCP request processed by your load balancer. This includes the IP address and port of the requesting client, the backend IP address of the instance that processed the request, the size of the request and response, and the actual request line from the client (for example, GET http://www.example.com: 80/HTTP/1.1). All requests sent to the load balancer are logged, including requests that never made it to backend instances.

Amazon Virtual Private Cloud (Amazon VPC) Security

Normally, each Amazon EC2 instance that you launch is randomly assigned a public IP address in the Amazon EC2 address space. Amazon VPC enables you to create an isolated portion of the AWS cloud and launch Amazon EC2 instances that have private (RFC 1918) addresses in the range of your choice (e.g., 10.0.0.0/16). You can define subnets within your VPC, grouping similar kinds of instances based on IP address range, and then set up routing and security to control the flow of traffic in and out of the instances and subnets.

AWS offers a variety of VPC architecture templates with configurations that provide varying levels of public access:

- **VPC with a single public subnet only.** Your instances run in a private, isolated section of the AWS cloud with direct access to the Internet. Network ACLs and security groups can be used to provide strict control over inbound and outbound network traffic to your instances.
- **VPC with public and private subnets.** In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation (NAT).
- **VPC with public and private subnets and hardware VPN access.** This configuration adds an IPsec VPN connection between your Amazon VPC and your data center, effectively extending your data center to the cloud while also providing direct access to the Internet for public subnet instances in your Amazon VPC. In this configuration, customers add a VPN appliance on their corporate data center side.
- **VPC with private subnet only and hardware VPN access.** Your instances run in a private, isolated section of the AWS cloud with a private subnet whose instances are not addressable from the Internet. You can connect this private subnet to your corporate data center via an IPsec VPN tunnel.

You can also connect two VPCs using a private IP address, which allows instances in the two VPCs to communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account within a single region.

Security features within Amazon VPC include security groups, network ACLs, routing tables, and external gateways. Each of these items is complementary to providing a secure, isolated network that can be extended through selective enabling of direct Internet access or private connectivity to another network.

Amazon EC2 instances running within an Amazon VPC inherit all of the benefits described below related to the guest OS and protection against packet sniffing.

Note, however, that you must create VPC security groups specifically for your Amazon VPC; any Amazon EC2 security groups you have created will not work inside your Amazon VPC. Also, Amazon VPC security groups have additional capabilities that Amazon EC2 security groups do not have, such as being able to change the security group after the instance is launched and being able to specify any protocol with a standard protocol number (as opposed to just TCP, UDP, or ICMP).

Each Amazon VPC is a distinct, isolated network within the cloud; network traffic within each Amazon VPC is isolated from all other Amazon VPCs. At creation time, you select an IP address range for each Amazon VPC. You may create and attach an Internet gateway, virtual private gateway, or both to establish external connectivity, subject to the controls below.

API Access: Calls to create and delete Amazon VPCs, change routing, security group, and network ACL parameters, and perform other functions are all signed by your Amazon Secret Access Key, which could be either the AWS Account's Secret Access Key or the Secret Access key of a user created with AWS IAM. Without access to your Secret Access Key, Amazon VPC API calls cannot be made on your behalf. In addition, API calls can be encrypted with SSL to maintain confidentiality. Amazon recommends always using SSL-protected API endpoints. AWS IAM also enables a customer to further control what APIs a newly created user has permissions to call.

Subnets and Route Tables: You create one or more subnets within each Amazon VPC; each instance launched in the Amazon VPC is connected to one subnet. Traditional Layer 2 security attacks, including MAC spoofing and ARP spoofing, are blocked.

Each subnet in an Amazon VPC is associated with a routing table, and all network traffic leaving the subnet is processed by the routing table to determine the destination.

Firewall (Security Groups): Like Amazon EC2, Amazon VPC supports a complete firewall solution enabling filtering on both ingress and egress traffic from an instance. The default group enables inbound communication from other members of the same group and outbound communication to any destination.

Traffic can be restricted by any IP protocol, by service port, as well as source/destination IP address (individual IP or Classless Inter-Domain Routing (CIDR) block).

The firewall isn't controlled through the guest OS; rather, it can be modified only through the invocation of Amazon VPC APIs. AWS supports the ability to grant granular access to different administrative functions on the instances and the firewall, therefore enabling you to implement additional security through separation of duties. The level of security afforded by the firewall is a function of which ports you open, and for what duration and purpose. Well-informed traffic management and security design are still required on a per-instance basis.

AWS further encourages you to apply additional per-instance filters with host-based firewalls such as IP tables or the Windows Firewall.

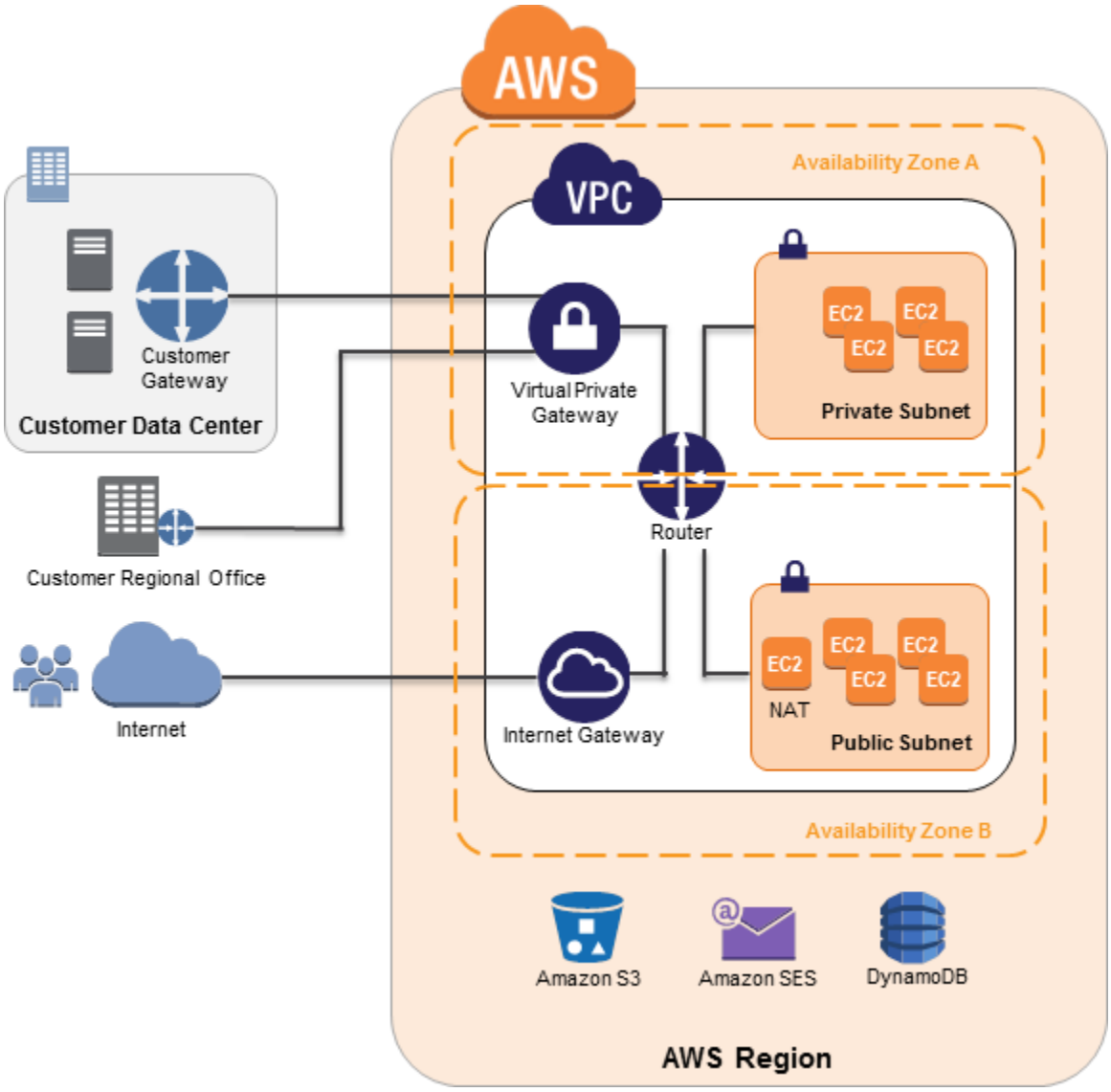

Figure 4: Amazon VPC network architecture

Network Access Control Lists: To add a further layer of security within Amazon VPC, you can configure network ACLs. These are stateless traffic filters that apply to all traffic inbound or outbound from a subnet within Amazon VPC. These ACLs can contain ordered rules to allow or deny traffic based upon IP protocol, by service port, as well as source/destination IP address.

Like security groups, network ACLs are managed through Amazon VPC APIs, adding an additional layer of protection and enabling additional security through separation of duties. The diagram below depicts how the security controls above inter-relate to enable flexible network topologies while providing complete control over network traffic flows.

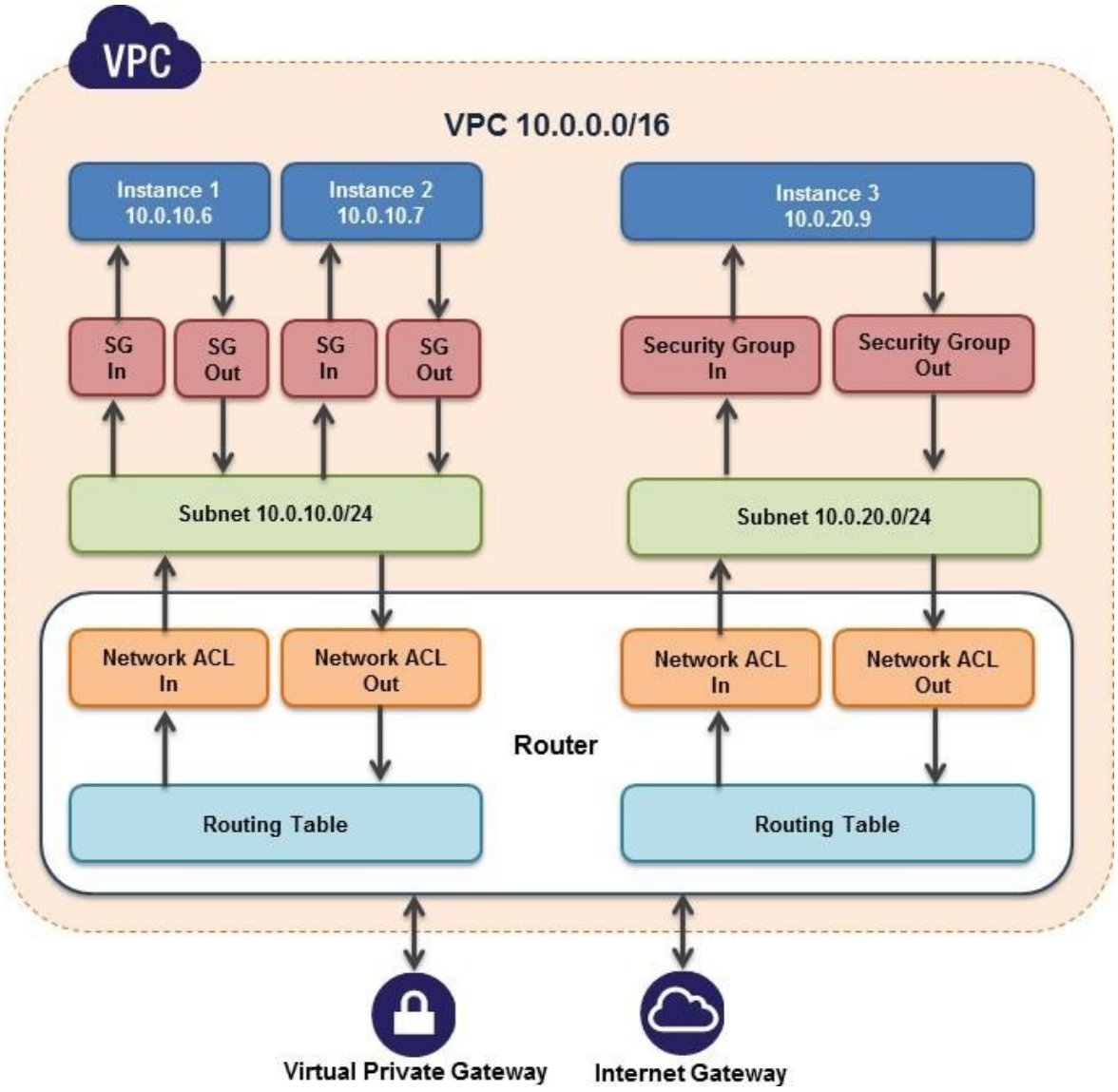

Figure 5: Flexible network topologies

Virtual Private Gateway: A virtual private gateway enables private connectivity between the Amazon VPC and another network. Network traffic within each virtual private gateway is isolated from network traffic within all other virtual private gateways. You can establish VPN connections to the virtual private gateway from gateway devices at your premises. Each connection is secured by a pre-shared key in conjunction with the IP address of the customer gateway device.

Internet Gateway: An Internet gateway may be attached to an Amazon VPC to enable direct connectivity to Amazon S3, other AWS services, and the Internet. Each instance desiring this access must either have an Elastic IP associated with it or route traffic through a NAT instance. Additionally, network routes are configured (see above) to direct traffic to the Internet gateway. AWS provides reference NAT AMIs that you can extend to perform network logging, deep packet inspection, application-layer filtering, or other security controls.

This access can only be modified through the invocation of Amazon VPC APIs. AWS supports the ability to grant granular access to different administrative functions on the instances and the Internet gateway, therefore enabling you to implement additional security through separation of duties. You can use

a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.

Dedicated Instances: Within a VPC, you can launch Amazon EC2 instances that are physically isolated at the host hardware level (i.e., they will run on single-tenant hardware). An Amazon VPC can be created with 'dedicated' tenancy, so that all instances launched into the Amazon VPC use this feature. Alternatively, an Amazon VPC may be created with 'default' tenancy, but you can specify dedicated tenancy for particular instances launched into it.

Elastic Network Interfaces: Each Amazon EC2 instance has a default network interface that is assigned a private IP address on your Amazon VPC network.

You can create and attach an additional network interface, known as an elastic network interface, to any Amazon EC2 instance in your Amazon VPC for a total of two network interfaces per instance. Attaching more than one network interface to an instance is useful when you want to create a management network, use network and security appliances in your Amazon VPC, or create dual-homed instances with workloads/roles on distinct subnets. A network interface's attributes, including the private IP address, elastic IP addresses, and MAC address, follows the network interface as it is attached or detached from an instance and reattached to another instance. For more information about Amazon VPC, see Amazon Virtual Private Cloud.

Additional Network Access Control with EC2-VPC

If you launch instances in a Region where you did not have instances before AWS launched the new EC2-VPC feature (also called Default VPC), all instances are automatically provisioned in a ready-to-use default VPC. You can choose to create additional VPCs, or you can create VPCs for instances in regions where you already had instances before we launched EC2-VPC.

If you create a VPC later, using regular VPC, you specify a CIDR block, create subnets, enter the routing and security for those subnets, and provision an Internet gateway or NAT instance if you want one of your subnets to be able to reach the Internet. When you launch EC2 instances into an EC2-VPC, most of this work is automatically performed for you.

When you launch an instance into a default VPC using EC2-VPC, we do the following to set it up for you:

- Create a default subnet in each Availability Zone
- Create an internet gateway and connect it to your default VPC
- Create a main route table for your default VPC with a rule that sends all traffic destined for the Internet to the Internet gateway
- Create a default security group and associate it with your default VPC
- Create a default network access control list (ACL) and associate it with your default VPC
- Associate the default DHCP options set for your AWS account with your default VPC

In addition to the default VPC having its own private IP range, EC2 instances launched in a default VPC can also receive a public IP.

The following table summarizes the differences between instances launched into EC2-Classic, instances launched into a default VPC, and instances launched into a non-default VPC.

Table 2: Differences between different EC2 instances

Characteristic	EC2-Classic	EC2-VPC (Default VPC)	Regular VPC
		IP address by default, unless you specify otherwise during launch.	Unless you specify otherwise during launch.
Private IP address	Your instance receives a private IP address from the EC2-Classic range each time it's started.	Your instance receives a static private IP address from the address range of your default VPC.	Your instance receives a static private IP address from the address range of your VPC.
Multiple private IP addresses	We select a single IP address for your instance. Multiple IP addresses are not supported.	You can assign multiple private IP addresses to your instance.	You can assign multiple private IP addresses to your instance.
Elastic IP address	An EIP is disassociated from your instance when you stop it.	An EIP remains associated with your instance when you stop it.	An EIP remains associated with your instance when you stop it.
DNS hostnames	DNS hostnames are enabled by default.	DNS hostnames are enabled by default.	DNS hostnames are disabled by default.
Security group	A security group can reference security groups that belong to other AWS accounts.	A security group can reference security groups for your VPC only.	A security group can reference security groups for your VPC only.
Security group association	You must terminate your instance to change its security group.	You can change the security group of your running instance.	You can change the security group of your running instance.
Security group rules	You can add rules for inbound traffic only.	You can add rules for inbound and outbound traffic.	You can add rules for inbound and outbound traffic.
Tenancy	Your instance runs on shared hardware; you cannot run an instance on single-tenant hardware.	You can run your instance on shared hardware or single-tenant hardware.	You can run your instance on shared hardware or single-tenant hardware.

Note: Security groups for instances in EC2-Classic are slightly different than security groups for instances in EC2-VPC. For example, you can add rules for inbound traffic for EC2-Classic, but you can add rules for both inbound and outbound traffic to EC2-VPC. In EC2-Classic, you can't change the security groups assigned to an instance after it's launched, but in EC2-VPC, you can change security groups assigned to an instance after it's launched. In addition, you can't use the security groups that you've created for use with EC2-Classic with instances in your VPC. You must create security groups specifically for use

with instances in your VPC. The rules you create for use with a security group for a VPC can't reference a security group for EC2-Classic, and vice versa.

Amazon Route 53 Security

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) service that answers DNS queries, translating domain names into IP addresses so computers can communicate with each other. Route 53 can be used to connect user requests to infrastructure running in AWS – such as an Amazon EC2 instance or an Amazon S3 bucket – or to infrastructure outside of AWS.

Amazon Route 53 lets you manage the IP addresses (records) listed for your domain names and it answers requests (queries) to translate specific domain names into their corresponding IP addresses. Queries for your domain are automatically routed to a nearby DNS server using anycast in order to provide the lowest latency possible. Route 53 makes it possible for you to manage traffic globally through a variety of routing types, including Latency Based Routing (LBR), Geo DNS, and Weighted Round- Robin (WRR) —all of which can be combined with DNS Failover in order to help create a variety of low- latency, fault-tolerant architectures. The failover algorithms implemented by Amazon Route 53 are designed not only to route traffic to endpoints that are healthy, but also to help avoid making disaster scenarios worse due to misconfigured health checks and applications, endpoint overloads, and partition failures.

Route 53 also offers Domain Name Registration – you can purchase and manage domain names such as example.com and Route 53 will automatically configure default DNS settings for your domains. You can buy, manage, and transfer (both in and out) domains from a wide selection of generic and country-specific top-level domains (TLDs). During the registration process, you have the option to enable privacy protection for your domain. This option will hide most of your personal information from the public Whois database in order to help thwart scraping and spamming.

Amazon Route 53 is built using AWS's highly available and reliable infrastructure. The distributed nature of the AWS DNS servers helps ensure a consistent ability to route your end users to your application. Route 53 also helps ensure the availability of your website by providing health checks and DNS failover capabilities. You can easily configure Route 53 to check the health of your website on a regular basis (even secure web sites that are available only over SSL), and to switch to a backup site if the primary one is unresponsive.

Like all AWS Services, Amazon Route 53 requires that every request made to its control API be authenticated so only authenticated users can access and manage Route 53. API requests are signed with an HMAC-SHA1 or HMAC- SHA256 signature calculated from the request and the user's AWS Secret Access key. Additionally, the Amazon Route 53 control API is only accessible via SSL- encrypted endpoints. It supports both IPv4 and IPv6 routing.

You can control access to Amazon Route 53 DNS management functions by creating users under your AWS Account using AWS IAM, and controlling which Route 53 operations these users have permission to perform.

Amazon CloudFront Security

Amazon CloudFront gives customers an easy way to distribute content to end users with low latency and high data transfer speeds. It delivers dynamic, static, and streaming content using a global network of edge locations. Requests for customers' objects are automatically routed to the nearest edge location, so content is delivered with the best possible performance. Amazon CloudFront is optimized to work with other AWS services, like Amazon S3, Amazon EC2, Elastic Load Balancing, and Amazon Route 53. It also works seamlessly with any non-AWS origin server that stores the original, definitive versions of your files.

Amazon CloudFront requires every request made to its control API be authenticated so only authorized users can create, modify, or delete their own Amazon CloudFront distributions. Requests are signed with an HMAC-SHA1 signature calculated from the request and the user's private key. Additionally, the Amazon CloudFront control API is only accessible via SSL-enabled endpoints.

There is no guarantee of durability of data held in Amazon CloudFront edge locations. The service may from time to time remove objects from edge locations if those objects are not requested frequently. Durability is provided by Amazon S3, which works as the origin server for Amazon CloudFront holding the original, definitive copies of objects delivered by Amazon CloudFront.

If you want control over who is able to download content from Amazon CloudFront, you can enable the service's private content feature. This feature has two components: the first controls how content is delivered from the Amazon CloudFront edge location to viewers on the Internet. The second controls how the Amazon CloudFront edge locations access objects in Amazon S3. CloudFront also supports Geo Restriction, which restricts access to your content based on the geographic location of your viewers.

To control access to the original copies of your objects in Amazon S3, Amazon CloudFront allows you to create one or more "Origin Access Identities" and associate these with your distributions. When an Origin Access Identity is associated with an Amazon CloudFront distribution, the distribution will use that identity to retrieve objects from Amazon S3. You can then use Amazon S3's ACL feature, which limits access to that Origin Access Identity so the original copy of the object is not publicly readable.

To control who is able to download objects from Amazon CloudFront edge locations, the service uses a signed-URL verification system. To use this system, you first create a public-private key pair, and upload the public key to your account via the AWS Management Console. Second, you configure your Amazon CloudFront distribution to indicate which accounts you would authorize to sign requests – you can indicate up to five AWS Accounts you trust to sign requests. Third, as you receive requests you will create policy documents indicating the conditions under which you want Amazon CloudFront to serve your content. These policy documents can specify the name of the object that is requested, the date and time of the request, and the source IP (or CIDR range) of the client making the request. You then calculate the SHA1 hash of your policy document and sign this using your private key. Finally, you include both the encoded policy document and the signature as query string parameters when you reference your objects. When Amazon CloudFront receives a request, it will decode the signature using your public key. Amazon CloudFront only serves requests that have a valid policy document and matching signature.

Note: Private content is an optional feature that must be enabled when you set up your CloudFront distribution. Content delivered without this feature enabled will be publicly readable.

Amazon CloudFront provides the option to transfer content over an encrypted connection (HTTPS). By default, CloudFront accepts requests over both HTTP and HTTPS protocols. However, you can also configure CloudFront to require HTTPS for all requests or have CloudFront redirect HTTP requests to HTTPS. You can even configure CloudFront distributions to allow HTTP for some objects but require HTTPS for other objects.

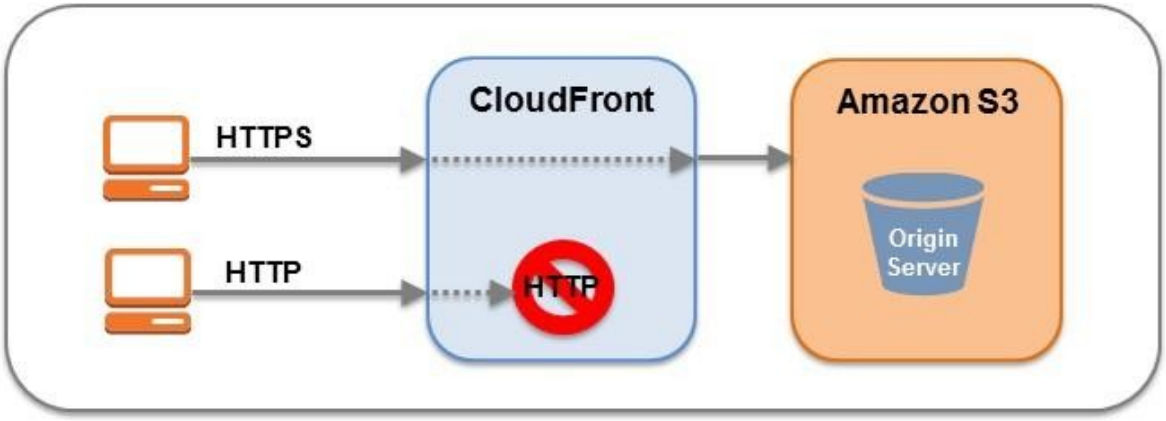

Figure 6: Amazon CloudFront encrypted transmission

You can configure one or more CloudFront origins to require CloudFront fetch objects from your origin using the protocol that the viewer used to request the objects. For example, when you use this CloudFront setting and the viewer uses HTTPS to request an object from CloudFront, CloudFront also uses HTTPS to forward the request to your origin.

Amazon CloudFront uses the SSLv3 or TLSv1 protocols and a selection of cipher suites that includes the Elliptic Curve Diffie-Hellman Ephemeral (ECDHE) protocol on connections to both viewers and the origin. ECDHE allows SSL/TLS clients to provide Perfect Forward Secrecy, which uses session keys that are ephemeral and not stored anywhere. This helps prevent the decoding of captured data by unauthorized third parties, even if the secret long- term key itself is compromised.

Note: If you're using your own server as your origin, and you want to use HTTPS both between viewers and CloudFront and between CloudFront and your origin, you must install a valid SSL certificate on the HTTP server that is signed by a third-party certificate authority, for example, VeriSign or DigiCert.

By default, you can deliver content to viewers over HTTPS by using your CloudFront distribution domain name in your URLs; for example, https://dxxxxx.cloudfront.net/image.jpg. If you want to deliver your content over HTTPS using your own domain name and your own SSL certificate, you can use SNI Custom SSL or Dedicated IP Custom SSL. With Server Name Identification (SNI) Custom SSL, CloudFront relies on the SNI extension of the TLS protocol, which is supported by most modern web browsers. However, some users may not be able to access your content because some older browsers do not support SNI. (For a list of supported browsers, visit CloudFront FAQs.) With Dedicated IP Custom SSL, CloudFront dedicates IP addresses to your SSL certificate at each CloudFront edge location so that CloudFront can associate the incoming requests with the proper SSL certificate.

Amazon CloudFront access logs contain a comprehensive set of information about requests for content, including the object requested, the date and time of the request, the edge location serving the request, the client IP address, the referrer, and the user agent. To enable access logs, just specify the name of the Amazon S3 bucket to store the logs in when you configure your Amazon CloudFront distribution.

AWS Direct Connect Security

With AWS Direct Connect, you can provision a direct link between your internal network and an AWS region using a high-throughput, dedicated connection.

Doing this may help reduce your network costs, improve throughput, or provide a more consistent network experience. With this dedicated connection in place, you can then create virtual interfaces directly to the AWS Cloud (for example, to Amazon EC2 and Amazon S3) and Amazon VPC.

With Direct Connect, you bypass internet service providers in your network path. You can procure rack space within the facility housing the AWS Direct Connect location and deploy your equipment nearby. Once deployed, you can connect this equipment to AWS Direct Connect using a cross-connect. Each AWS Direct Connect location enables connectivity to the geographically nearest AWS region as well as access to other US regions. For example, you can provision a single connection to any AWS Direct Connect location in the US and use it to access public AWS services in all US Regions and AWS GovCloud (US).

Using industry standard 802.1q VLANs, the dedicated connection can be partitioned into multiple virtual interfaces. This allows you to use the same connection to access public resources such as objects stored in Amazon S3 using public IP address space, and private resources such as Amazon EC2 instances running within an Amazon VPC using private IP space, while maintaining network separation between the public and private environments.

Amazon Direct Connect requires the use of the Border Gateway Protocol (BGP) with an Autonomous System Number (ASN). To create a virtual interface, you use an MD5 cryptographic key for message authorization. MD5 creates a keyed hash using your secret key. You can have AWS automatically generate a BGP MD5 key or you can provide your own.

Storage Services

Amazon Web Services provides low-cost data storage with high durability and availability. AWS offers storage choices for backup, archiving, and disaster recovery, as well as block and object storage.

Topics

Amazon Simple Storage Service (Amazon S3) Security

Amazon Simple Storage Service (Amazon S3) allows you to upload and retrieve data at any time, from anywhere on the web. Amazon S3 stores data as objects within buckets. An object can be any kind of file: a text file, a photo, a video, etc.

When you add a file to Amazon S3, you have the option of including metadata with the file and setting permissions to control access to the file. For each bucket, you can control access to the bucket (who can create, delete, and list objects in the bucket), view access logs for the bucket and its objects, and choose the geographical region where Amazon S3 will store the bucket and its contents.

Data Access

Access to data stored in Amazon S3 is restricted by default; only bucket and object owners have access to the Amazon S3 resources they create (note that a bucket/object owner is the AWS Account owner, not the user who created the bucket/object). There are multiple ways to control access to buckets and objects:

- **Identity and Access Management (IAM) Policies.** AWS IAM enables organizations with many employees to create and manage multiple users under a single AWS Account. IAM policies are attached to the users, enabling centralized control of permissions for users under your AWS Account to access buckets or objects. With IAM policies, you can only grant users within your own AWS account permission to access your Amazon S3 resources.
- **Access Control Lists (ACLs).** Within Amazon S3, you can use ACLs to give read or write access on buckets or objects to groups of users. With ACLs, you can only grant other AWS accounts (not specific users) access to your Amazon S3 resources.
- **Bucket Policies.** Bucket policies in Amazon S3 can be used to add or deny permissions across some or all of the objects within a single bucket. Policies can be attached to users, groups, or Amazon S3 buckets, enabling centralized management of permissions. With bucket policies, you can grant users within your AWS Account or other AWS Accounts access to your Amazon S3 resources.

Table 3: Types of access control

Type of Access Control	AWS Account Level Control	User Level Control
IAM Policies	No	Yes
ACLs	Yes	No
Bucket Policies	Yes	Yes

You can further restrict access to specific resources based on certain conditions. For example, you can restrict access based on request time (Date Condition), whether the request was sent using SSL (Boolean Conditions), a requester's IP address (IP Address Condition), or based on the requester's client application

(String Conditions). To identify these conditions, you use policy keys. For more information about action-specific policy keys available within Amazon S3, see the Amazon Simple Storage Service Developer Guide.

Amazon S3 also gives developers the option to use query string authentication, which allows them to share Amazon S3 objects through URLs that are valid for a predefined period of time. Query string authentication is useful for giving HTTP or browser access to resources that would normally require authentication. The signature in the query string secures the request.

Data Transfer

For maximum security, you can securely upload/download data to Amazon S3 via the SSL encrypted endpoints. The encrypted endpoints are accessible from both the Internet and from within Amazon EC2, so that data is transferred securely both within AWS and to and from sources outside of AWS.

Data Storage

Amazon S3 provides multiple options for protecting data at rest. For customers who prefer to manage their own encryption, they can use a client encryption library like the Amazon S3 Encryption Client to encrypt data before uploading to Amazon S3. Alternatively, you can use Amazon S3 Server-Side Encryption (SSE) if you prefer to have Amazon S3 manage the encryption process for you. Data is encrypted with a key generated by AWS or with a key you supply, depending on your requirements. With Amazon S3 SSE, you can encrypt data on upload simply by adding an additional request header when writing the object. Decryption happens automatically when data is retrieved.

Note: Metadata, which you can include with your object, is not encrypted. Therefore, AWS recommends that customers not place sensitive information in Amazon S3 metadata.

Amazon S3 SSE uses one of the strongest block ciphers available – 256-bit Advanced Encryption Standard (AES-256). With Amazon S3 SSE, every protected object is encrypted with a unique encryption key. This object key itself is then encrypted with a regularly rotated master key. Amazon S3 SSE provides additional security by storing the encrypted data and encryption keys in different hosts. Amazon S3 SSE also makes it possible for you to enforce encryption requirements. For example, you can create and apply bucket policies that require that only encrypted data can be uploaded to your buckets.

For long-term storage, you can automatically archive the contents of your Amazon S3 buckets to AWS's archival service called Amazon S3 Glacier. You can have data transferred at specific intervals to Amazon S3 Glacier by creating lifecycle rules in Amazon S3 that describe which objects you want to be archived to Amazon S3 Glacier and when. As part of your data management strategy, you can also specify how long Amazon S3 should wait after the objects are put into Amazon S3 to delete them.

When an object is deleted from Amazon S3, removal of the mapping from the public name to the object starts immediately, and is generally processed across the distributed system within several seconds. Once the mapping is removed, there is no remote access to the deleted object. The underlying storage area is then reclaimed for use by the system.

Data Durability and Reliability

Amazon S3 is designed to provide 99.999999999% durability and 99.99% availability of objects over a given year. Objects are redundantly stored on multiple devices across multiple facilities in an Amazon S3 region. To help provide durability, Amazon S3 PUT and COPY operations synchronously store customer data across multiple facilities before returning SUCCESS. Once stored, Amazon S3 helps maintain the durability of the objects by quickly detecting and repairing any lost redundancy. Amazon S3 also regularly verifies the integrity of data stored using checksums. If corruption is detected, it is repaired using redundant data. In addition, Amazon S3 calculates checksums on all network traffic to detect corruption of data packets when storing or retrieving data.

Amazon S3 provides further protection via Versioning. You can use Versioning to preserve, retrieve, and restore every version of every object stored in an Amazon S3 bucket. With Versioning, you can easily recover from both unintended user actions and application failures. By default, requests will retrieve the most recently written version. Older versions of an object can be retrieved by specifying a version in the request. You can further protect versions using Amazon S3 Versioning's MFA Delete feature. Once enabled for an Amazon S3 bucket, each version deletion request must include the six-digit code and serial number from your multi-factor authentication device.

Access Logs

An Amazon S3 bucket can be configured to log access to the bucket and objects within it. The access log contains details about each access request including request type, the requested resource, the requestor's IP, and the time and date of the request. When logging is enabled for a bucket, log records are periodically aggregated into log files and delivered to the specified Amazon S3 bucket.

Cross-Origin Resource Sharing (CORS)

AWS customers who use Amazon S3 to host static web pages or store objects used by other web pages can load content securely by configuring an Amazon S3 bucket to explicitly enable cross-origin requests. Modern browsers use the Same Origin policy to block JavaScript or HTML5 from allowing requests to load content from another site or domain as a way to help ensure that malicious content is not loaded from a less reputable source (such as during cross-site scripting attacks). With the Cross-Origin Resource Sharing (CORS) policy enabled, assets such as web fonts and images stored in an Amazon S3 bucket can be safely referenced by external web pages, style sheets, and HTML5 applications.

Amazon S3 Glacier Security

Like Amazon S3, the Amazon S3 Glacier service provides low-cost, secure, and durable storage. But where Amazon S3 is designed for rapid retrieval, Amazon S3 Glacier is meant to be used as an archival service for data that is not accessed often and for which retrieval times of several hours are suitable.

Amazon S3 Glacier stores files as archives within vaults. Archives can be any data such as a photo, video, or document, and can contain one or several files. You can store an unlimited number of archives in a single vault and can create up to 1,000 vaults per region. Each archive can contain up to 40 TB of data.

Data Upload

To transfer data into Amazon S3 Glacier vaults, you can upload an archive in a single upload operation or a multipart operation. In a single upload operation, you can upload archives up to 4 GB in size. However, customers can achieve better results using the Multipart Upload API to upload archives greater than 100 MB. Using the Multipart Upload API allows you to upload large archives, up to about 40,000 GB. The Multipart Upload API call is designed to improve the upload experience for larger archives; it enables the parts to be uploaded independently, in any order, and in parallel. If a multipart upload fails, you only need to upload the failed part again and not the entire archive.

When you upload data to Amazon S3 Glacier, you must compute and supply a tree hash. Amazon S3 Glacier checks the hash against the data to help ensure that it has not been altered en route. A tree hash is generated by computing a hash for each megabyte-sized segment of the data, and then combining the hashes in tree fashion to represent ever-growing adjacent segments of the data.

As an alternate to using the Multipart Upload feature, customers with very large uploads to Amazon S3 Glacier may consider using the AWS Snowball service instead to transfer the data. AWS Snowball facilitates moving large amounts of data into AWS using portable storage devices for transport. AWS transfers your data directly off of storage devices using Amazon's high-speed internal network, bypassing the Internet.

You can also set up Amazon S3 to transfer data at specific intervals to Amazon S3 Glacier. You can create lifecycle rules in Amazon S3 that describe which objects you want to be archived to Amazon S3 Glacier and when. You can also specify how long Amazon S3 should wait after the objects are put into Amazon S3 to delete them.

To achieve even greater security, you can securely upload/download data to Amazon S3 Glacier via the SSL-encrypted endpoints. The encrypted endpoints are accessible from both the Internet and from within Amazon EC2, so that data is transferred securely both within AWS and to and from sources outside of AWS.

Data Retrieval

Retrieving archives from Amazon S3 Glacier requires the initiation of a retrieval job, which is generally completed in 3 to 5 hours. You can then access the data via HTTP GET requests. The data will remain available to you for 24 hours.

You can retrieve an entire archive or several files from an archive. If you want to retrieve only a subset of an archive, you can use one retrieval request to specify the range of the archive that contains the files you are interested or you can initiate multiple retrieval requests, each with a range for one or more files. You can also limit the number of vault inventory items retrieved by filtering on an archive creation date range or by setting a maximum items limit. Whichever method you choose, when you retrieve portions of your archive, you can use the supplied checksum to help ensure the integrity of the files provided that the range that is retrieved is aligned with the tree hash of the overall archive.

Data Storage

Amazon S3 Glacier automatically encrypts the data using AES-256 and stores it durably in an immutable form. Amazon S3 Glacier is designed to provide average annual durability of 99.999999999% for an archive. It stores each archive in multiple facilities and multiple devices. Unlike traditional systems which can require laborious data verification and manual repair, Amazon S3 Glacier performs regular, systematic data integrity checks and is built to be automatically self-healing.

When an object is deleted from Amazon S3 Glacier, removal of the mapping from the public name to the object starts immediately, and is generally processed across the distributed system within several seconds. Once the mapping is removed, there is no remote access to the deleted object. The underlying storage area is then reclaimed for use by the system.

Data Access

Only your account can access your data in Amazon S3 Glacier. To control access to your data in Amazon S3 Glacier, you can use AWS IAM to specify which users within your account have rights to operations on a given vault.

AWS Storage Gateway Security

The AWS Storage Gateway service connects your on-premises software appliance with cloud-based storage to provide seamless and secure integration between your IT environment and the AWS storage infrastructure. The service enables you to securely upload data to AWS' scalable, reliable, and secure Amazon S3 storage service for cost-effective backup and rapid disaster recovery.

AWS Storage Gateway transparently backs up data off-site to Amazon S3 in the form of Amazon EBS snapshots. Amazon S3 redundantly stores these snapshots on multiple devices across multiple facilities, detecting and repairing any lost redundancy. The Amazon EBS snapshot provides a point-in-time backup that can be restored on-premises or used to instantiate new Amazon EBS volumes.

Data is stored within a single region that you specify. AWS Storage Gateway offers three options:

- **Gateway-Stored Volumes (where the cloud is backup).** In this option, your volume data is stored locally and then pushed to Amazon S3, where it is stored in redundant, encrypted form, and made available in the form of Amazon Elastic Block Storage (Amazon EBS) snapshots. When you use this model, the on-premises storage is primary, delivering low-latency access to your entire dataset, and the cloud storage is the backup.
- **Gateway-Cached Volumes (where the cloud is primary).** In this option, your volume data is stored encrypted in Amazon S3, visible within your enterprise's network via an iSCSI interface. Recently accessed data is cached on- premises for low-latency local access. When you use this model, the cloud storage is primary, but you get low- latency access to your active working set in the cached volumes on premises.
- **Gateway-Virtual Tape Library (VTL).** In this option, you can configure a Gateway-VTL with up to 10 virtual tape drives per gateway, 1 media changer and up to 1500 virtual tape cartridges. Each virtual tape drive responds to the SCSI command set, so your existing on-premises backup applications (either disk-to-tape or disk-to-disk-to- tape) will work without modification.

No matter which option you choose, data is asynchronously transferred from your on-premises storage hardware to AWS over SSL. The data is stored encrypted in Amazon S3 using Advanced Encryption Standard (AES) 256, a symmetric- key encryption standard using 256-bit encryption keys. The AWS Storage Gateway only uploads data that has changed, minimizing the amount of data sent over the Internet.

The AWS Storage Gateway runs as a virtual machine (VM) that you deploy on a host in your data center running VMware ESXi Hypervisor v 4.1 or v 5 or Microsoft Hyper-V (you download the VMware software during the setup process). You can also run within EC2 using a gateway AMI. During the installation and configuration process, you can create up to 12 stored volumes, 20 Cached volumes, or 1500 virtual tape cartridges per gateway. Once installed, each gateway will automatically download, install, and deploy updates and patches. This activity takes place during a maintenance window that you can set on a per-gateway basis.

The iSCSI protocol supports authentication between targets and initiators via CHAP (Challenge-Handshake Authentication Protocol). CHAP provides protection against man-in-the-middle and playback attacks by periodically verifying the identity of an iSCSI initiator as authenticated to access a storage volume target. To set up CHAP, you must configure it in both the AWS Storage Gateway console and in the iSCSI initiator software you use to connect to the target.

After you deploy the AWS Storage Gateway VM, you must activate the gateway using the AWS Storage Gateway console. The activation process associates your gateway with your AWS Account. Once you establish this connection, you can manage almost all aspects of your gateway from the console. In the activation process, you specify the IP address of your gateway, name your gateway, identify the AWS region in which you want your snapshot backups stored, and specify the gateway time zone.

AWS Snowball Security

AWS Snowball is a simple, secure method for physically transferring large amounts of data to Amazon S3, EBS, or Amazon S3 Glacier storage. This service is typically used by customers who have over 100 GB of data and/or slow connection speeds that would result in very slow transfer rates over the Internet. With AWS Snowball, you prepare a portable storage device that you ship to a secure AWS facility. AWS transfers the data directly off of the storage device using Amazon's high-speed internal network, thus bypassing the Internet. Conversely, data can also be exported from AWS to a portable storage device.

Like all other AWS services, the AWS Snowball service requires that you securely identify and authenticate your storage device. In this case, you will submit a job request to AWS that includes your Amazon S3 bucket, Amazon EBS region, AWS Access Key ID, and return shipping address. You then receive a unique identifier for the job, a digital signature for authenticating your device, and an AWS address to ship the storage device to. For Amazon S3, you place the signature file on the root directory of your device. For Amazon EBS, you tape the signature barcode to the exterior of the device. The signature file is used only for authentication and is not uploaded to Amazon S3 or EBS.

For transfers to Amazon S3, you specify the specific buckets to which the data should be loaded and ensure that the account doing the loading has write permission for the buckets. You should also specify the access control list to be applied to each object loaded to Amazon S3.

For transfers to EBS, you specify the target region for the EBS import operation. If the storage device is less than or equal to the maximum volume size of 1 TB, its contents are loaded directly into an Amazon EBS snapshot. If the storage device's capacity exceeds 1 TB, a device image is stored within the specified S3 log bucket. You can then create a RAID of Amazon EBS volumes using software such as Logical Volume Manager, and copy the image from S3 to this new volume.

For added protection, you can encrypt the data on your device before you ship it to AWS. For Amazon S3 data, you can use a PIN-code device with hardware encryption or TrueCrypt software to encrypt your data before sending it to AWS. For EBS and Amazon S3 Glacier data, you can use any encryption method you choose, including a PIN-code device. AWS will decrypt your Amazon S3 data before importing using the PIN code and/or TrueCrypt password you supply in your import manifest. AWS uses your PIN to access a PIN-code device, but does not decrypt software-encrypted data for import to Amazon EBS or Amazon S3 Glacier. The following table summarizes your encryption options for each type of import/export job.

Table 4: Encryption options for import/export jobs

Import to Amazon S3		
Source	**Target**	**Result**
• Files on a device file system • Encrypt data using PIN-code device and/or TrueCrypt before shipping device	• Objects in an existing Amazon S3 bucket • AWS decrypts the data before performing the import	• One object for each file. • AWS erases your device after every import job prior to shipping
Export from Amazon S3		
Source	**Target**	**Result**
• Objects in one or more Amazon S3 buckets • Provide a PIN code and/or password that AWS will use to encrypt your data	• Files on your storage device • AWS formats your device • AWS copies your data to an encrypted file container on your device	• One file for each object • AWS encrypts your data prior to shipping • Use PIN-code device and/or TrueCrypt to decrypt the files
Import to Amazon S3 Glacier		
Source	**Target**	**Result**
• Entire device • Encrypt the data using the encryption method of your choice before shipping	• One archive in an existing Amazon S3 Glacier vault • AWS does not decrypt your device	• Device image stored as a single archive • AWS erases your device after every import job prior to shipping
Import to Amazon EBS (Device Capacity < 1 TB)		
Source	**Target**	**Result**
• Entire device • Encrypt the data using the encryption method of your choice before shipping	• One Amazon EBS snapshot • AWS does not decrypt your device	• Device image is stored as a single snapshot • If the device was encrypted, the image is encrypted

Import to Amazon S3		
		• AWS erases your device after every import job prior to shipping
Import to Amazon EBS (Device Capacity > 1 TB)		
Source	**Target**	**Result**
• Entire device • Encrypt the data using the encryption method of your choice before shipping	• Multiple objects in an existing Amazon S3 bucket • AWS does not decrypt your device	• Device image chunked into series of 1 TB snapshots stored as objects in Amazon S3 bucket specified in manifest file • If the device was encrypted, the image is encrypted • AWS erases your device after every import job prior to shipping

After the import is complete, AWS Snowball will erase the contents of your storage device to safeguard the data during return shipment. AWS overwrites all writable blocks on the storage device with zeroes. You will need to repartition and format the device after the wipe. If AWS is unable to erase the data on the device, it will be scheduled for destruction and our support team will contact you using the email address specified in the manifest file you ship with the device.

When shipping a device internationally, the customs option and certain required subfields are required in the manifest file sent to AWS. AWS Snowball uses these values to validate the inbound shipment and prepare the outbound customs paperwork. Two of these options are whether the data on the device is encrypted or not and the encryption software's classification. When shipping encrypted data to or from the United States, the encryption software must be classified as 5D992 under the United States Export Administration Regulations.

Amazon Elastic File System Security

Amazon Elastic File System (Amazon EFS) provides simple, scalable file storage for use with Amazon EC2 instances in the AWS Cloud. With Amazon EFS, storage capacity is elastic, growing and shrinking automatically as you add and remove files. Amazon EFS file systems are distributed across an unconstrained number of storage servers, enabling file systems to grow elastically to petabyte- scale and allowing massively parallel access from Amazon EC2 instances to your data.

Data Access

With Amazon EFS, you can create a file system, mount the file system on an Amazon EC2 instance, and then read and write data from to and from your file system. You can mount an Amazon EFS file system on EC2 instances in your VPC, through the Network File System versions 4.0 and 4.1 (NFSv4) protocol.

To access your Amazon EFS file system in a VPC, you create one or more mount targets in the VPC. A mount target provides an IP address for an NFSv4 endpoint. You can then mount an Amazon EFS file system to this end point using its DNS name, which will resolve to the IP address of the EFS mount target in the same Availability Zone as your EC2 instance.

You can create one mount target in each Availability Zone in a region. If there are multiple subnets in an Availability Zone in your VPC, you create a mount target in one of the subnets, and all EC2 instances in

that Availability Zone share that mount target. You can also mount an EFS file system on a host in an on-premises datacenter using AWS Direct Connect.

When using Amazon EFS, you specify Amazon EC2 security groups for your EC2 instances and security groups for the EFS mount targets associated with the file system. Security groups act as a firewall, and the rules you add define the traffic flow. You can authorize inbound/outbound access to your EFS file system by adding rules that allow your EC2 instance to connect to your Amazon EFS file system via the mount target using the NFS port.

After mounting the file system via the mount target, you use it like any other POSIX-compliant file system. Files and directories in an EFS file system support standard Unix-style read/write/execute permissions based on the user and group ID asserted by the mounting NFSv4.1 client. For information about NFS- level permissions and related considerations, see Working with Users, Groups, and Permissions at the Network File System (NFS) Level.

All Amazon EFS file systems are owned by an AWS Account. You can use IAM policies to grant permissions to other users so that they can perform administrative operations on your file systems, including deleting a file system or modifying a mount target's security groups. For more information about EFS permissions, see Overview of Managing Access Permissions to Your Amazon EFS Resources.

Data Durability and Reliability

Amazon EFS is designed to be highly durable and highly available. All data and metadata is stored across multiple Availability Zones, and all service components are designed to be highly available. EFS provides strong consistency by synchronously replicating data across Availability Zones, with read-after-write semantics for most file operations. Amazon EFS incorporates checksums for all metadata and data throughout the service. Using a file system checking process (FSCK), EFS continuously validates a file system's metadata and data integrity.

Data Sanitization

Amazon EFS is designed so that when you delete data from a file system, that data will never be served again. If your procedures require that all data be wiped via a specific method, such as those detailed in DoD 5220.22-M ("National Industrial Security Program Operating Manual ") or NIST 800-88 ("Guidelines for Media Sanitization"), we recommend that you conduct a specialized wipe procedure prior to deleting the file system.

Database Services

Amazon Web Services provides a number of database solutions for developers and businesses—from managed relational and NoSQL database services, to in- memory caching as a service and petabyte-scale data-warehouse service.

Topics

Amazon DynamoDB Security

Amazon DynamoDB is a managed NoSQL database service that provides fast and predictable performance with seamless scalability. Amazon DynamoDB enables you to offload the administrative

burdens of operating and scaling distributed databases to AWS, so you don't have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling.

You can create a database table that can store and retrieve any amount of data, and serve any level of request traffic. DynamoDB automatically spreads the data and traffic for the table over a sufficient number of servers to handle the request capacity you specified and the amount of data stored, while maintaining consistent, fast performance. All data items are stored on Solid State Drives (SSDs) and are automatically replicated across multiple availability zones in a region to provide built-in high availability and data durability.

You can set up automatic backups using a special template in AWS Data Pipeline that was created just for copying DynamoDB tables. You can choose full or incremental backups to a table in the same region or a different region. You can use the copy for disaster recovery (DR) in the event that an error in your code damages the original table, or to federate DynamoDB data across regions to support a multi-region application.

To control who can use the DynamoDB resources and API, you set up permissions in AWS IAM. In addition to controlling access at the resource-level with IAM, you can also control access at the database level—you can create database-level permissions that allow or deny access to items (rows) and attributes (columns) based on the needs of your application. These database level permissions are called fine-grained access controls, and you create them using an IAM policy that specifies under what circumstances a user or application can access a DynamoDB table. The IAM policy can restrict access to individual items in a table, access to the attributes in those items, or both at the same time.

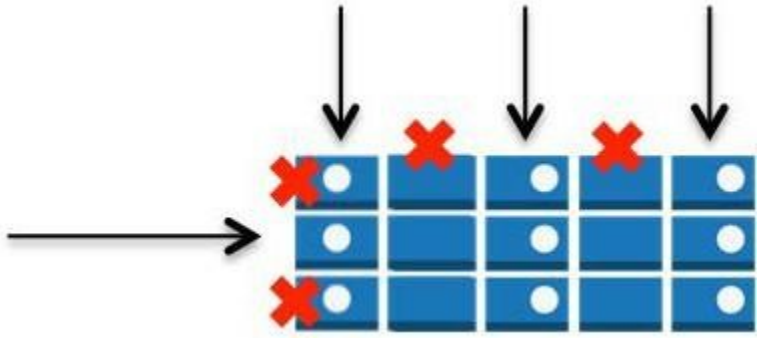

Figure 7: Database-level permissions

You can optionally use web identity federation to control access by application users who are authenticated by Login with Amazon, Facebook, or Google. Web identity federation removes the need for creating individual IAM users; instead, users can sign in to an identity provider and then obtain temporary security credentials from AWS Security Token Service (AWS STS). AWS STS returns temporary AWS credentials to the application and allows it to access the specific DynamoDB table.

In addition to requiring database and user permissions, each request to the DynamoDB service must contain a valid HMAC-SHA256 signature, or the request is rejected. The AWS SDKs automatically sign your requests; however, if you want to write your own HTTP POST requests, you must provide the signature in the header of your request to Amazon DynamoDB. To calculate the signature, you must request temporary security credentials from the AWS Security Token Service. Use the temporary security credentials to sign your requests to Amazon DynamoDB. Amazon DynamoDB is accessible via TSL/SSL-encrypted endpoints.

Amazon Relational Database Service (Amazon RDS) Security

Amazon RDS allows you to quickly create a relational database (DB) instance and flexibly scale the associated compute resources and storage capacity to meet application demand. Amazon RDS manages

the database instance on your behalf by performing backups, handling failover, and maintaining the database software. Currently, Amazon RDS is available for MySQL, Oracle, Microsoft SQL Server, and PostgreSQL database engines.

Amazon RDS has multiple features that enhance reliability for critical production databases, including DB security groups, permissions, SSL connections, automated backups, DB snapshots, and multi-AZ deployments. DB instances can also be deployed in an Amazon VPC for additional network isolation.

Topics

Access Control

When you first create a DB Instance within Amazon RDS, you will create a master user account, which is used only within the context of Amazon RDS to control access to your DB Instance(s). The master user account is a native database user account that allows you to log on to your DB Instance with all database privileges. You can specify the master user name and password you want associated with each DB Instance when you create the DB Instance. Once you have created your DB Instance, you can connect to the database using the master user credentials. Subsequently, you can create additional user accounts so that you can restrict who can access your DB Instance.

You can control Amazon RDS DB Instance access via DB Security Groups, which are similar to Amazon EC2 Security Groups but not interchangeable. DB Security Groups act like a firewall controlling network access to your DB Instance. Database Security Groups default to a “deny all” access mode and customers must specifically authorize network ingress. There are two ways of doing this: authorizing a network IP range or authorizing an existing Amazon EC2 Security Group. DB Security Groups only allow access to the database server port (all others are blocked) and can be updated without restarting the Amazon RDS DB Instance, which allows a customer seamless control of their database access.

Using AWS IAM, you can further control access to your RDS DB instances. AWS IAM enables you to control what RDS operations each individual AWS IAM user has permission to call.

Network Isolation

For additional network access control, you can run your DB Instances in an Amazon VPC. Amazon VPC enables you to isolate your DB Instances by specifying the IP range you wish to use, and connect to your existing IT infrastructure through industry-standard encrypted IPsec VPN. Running Amazon RDS in a VPC enables you to have a DB instance within a private subnet. You can also set up a virtual private gateway that extends your corporate network into your VPC, and allows access to the RDS DB instance in that VPC. Refer to the Amazon VPC User Guide for more details.

For Multi-AZ deployments, defining a subnet for all availability zones in a region will allow Amazon RDS to create a new standby in another availability zone should the need arise. You can create DB Subnet Groups, which are collections of subnets that you may want to designate for your RDS DB Instances in a VPC. Each DB Subnet Group should have at least one subnet for every availability zone in a given region. In this case, when you create a DB Instance in a VPC, you select a DB Subnet Group; Amazon RDS then uses that DB Subnet Group and your preferred availability zone to select a subnet and an IP address within that subnet. Amazon RDS creates and associates an Elastic Network Interface to your DB Instance with that IP address.

DB Instances deployed within an Amazon VPC can be accessed from the Internet or from Amazon EC2 Instances outside the VPC via VPN or bastion hosts that you can launch in your public subnet. To use a bastion host, you will need to set up a public subnet with an EC2 instance that acts as an SSH Bastion. This public subnet must have an Internet gateway and routing rules that allow traffic to be directed via the SSH host, which must then forward requests to the private IP address of your Amazon RDS DB instance.

DB Security Groups can be used to help secure DB Instances within an Amazon VPC. In addition, network traffic entering and exiting each subnet can be allowed or denied via network ACLs. All network traffic entering or exiting your Amazon VPC via your IPsec VPN connection can be inspected by your on-premises security infrastructure, including network firewalls and intrusion detection systems.

Encryption

You can encrypt connections between your application and your DB Instance using SSL. For MySQL and SQL Server, RDS creates an SSL certificate and installs the certificate on the DB instance when the instance is provisioned. For MySQL, you launch the mysql client using the --ssl_ca parameter to reference the public key in order to encrypt connections. For SQL Server, download the public key and import the certificate into your Windows operating system.

Oracle RDS uses Oracle native network encryption with a DB instance. You simply add the native network encryption option to an option group and associate that option group with the DB instance. Once an encrypted connection is established, data transferred between the DB Instance and your application will be encrypted during transfer. You can also require your DB instance to only accept encrypted connections.

Amazon RDS supports Transparent Data Encryption (TDE) for SQL Server (SQL Server Enterprise Edition) and Oracle (part of the Oracle Advanced Security option available in Oracle Enterprise Edition). The TDE feature automatically encrypts data before it is written to storage and automatically decrypts data when it is read from storage.

Note: SSL support within Amazon RDS is for encrypting the connection between your application and your DB Instance; it should not be relied on for authenticating the DB Instance itself.

While SSL offers security benefits, be aware that SSL encryption is a compute intensive operation and will increase the latency of your database connection. To learn how SSL works with SQL Server, you can read more in the Amazon Relational Database Service User Guide.

Automated Backups and DB Snapshots

Amazon RDS provides two different methods for backing up and restoring your DB Instance(s): automated backups and database snapshots (DB Snapshots).

Turned on by default, the automated backup feature of Amazon RDS enables point-in-time recovery for your DB Instance. Amazon RDS will back up your database and transaction logs and store both for a user-specified retention period. This allows you to restore your DB Instance to any second during your retention period, up to the last 5 minutes. Your automatic backup retention period can be configured to up to 35 days.

During the backup window, storage I/O may be suspended while your data is being backed up. This I/O suspension typically lasts a few minutes. This I/O suspension is avoided with Multi-AZ DB deployments, since the backup is taken from the standby.

DB Snapshots are user-initiated backups of your DB Instance. These full database backups are stored by Amazon RDS until you explicitly delete them. You can copy DB snapshots of any size and move them between any of AWS's public regions, or copy the same snapshot to multiple regions simultaneously. You can then create a new DB Instance from a DB Snapshot whenever you desire.

DB Instance Replication

Amazon cloud computing resources are housed in highly available data center facilities in different regions of the world, and each region contains multiple distinct locations called Availability Zones. Each Availability Zone is engineered to be isolated from failures in other Availability Zones, and to provide inexpensive, low-latency network connectivity to other Availability Zones in the same region.

To architect for high availability of your Oracle, PostgreSQL, or MySQL databases, you can run your RDS DB instance in several Availability Zones, an option called a Multi-AZ deployment. When you select this option, Amazon automatically provisions and maintains a synchronous standby replica of your DB instance in a different Availability Zone. The primary DB instance is synchronously replicated across Availability Zones to the standby replica. In the event of DB instance or Availability Zone failure, Amazon RDS will automatically failover to the standby so that database operations can resume quickly without administrative intervention.

For customers who use MySQL and need to scale beyond the capacity constraints of a single DB Instance for read-heavy database workloads, Amazon RDS provides a Read Replica option. Once you create a read replica, database updates on the source DB instance are replicated to the read replica using MySQL's native, asynchronous replication. You can create multiple read replicas for a given source DB instance and distribute your application's read traffic among them. Read replicas can be created with Multi-AZ deployments to gain read scaling benefits in addition to the enhanced database write availability and data durability provided by Multi-AZ deployments.

Automatic Software Patching

Amazon RDS will make sure that the relational database software powering your deployment stays up-to-date with the latest patches. When necessary, patches are applied during a maintenance window that you can control. You can think of the Amazon RDS maintenance window as an opportunity to control when DB Instance modifications (such as scaling DB Instance class) and software patching occur, in the event either are requested or required. If a "maintenance" event is scheduled for a given week, it will be initiated and completed at some point during the 30-minute maintenance window you identify.

The only maintenance events that require Amazon RDS to take your DB Instance offline are scale compute operations (which generally take only a few minutes from start-to-finish) or required software patching. Required patching is automatically scheduled only for patches that are security and durability related. Such patching occurs infrequently (typically once every few months) and should seldom require more than a fraction of your maintenance window. If you do not specify a preferred weekly maintenance window when creating your DB Instance, a 30-minute default value is assigned. If you wish to modify when maintenance is performed on your behalf, you can do so by modifying your DB Instance in the AWS Management Console or by using the ModifyDBInstance API. Each of your DB Instances can have different preferred maintenance windows, if you so choose.

Running your DB Instance as a Multi-AZ deployment can further reduce the impact of a maintenance event, as Amazon RDS will conduct maintenance via the following steps: 1) Perform maintenance on standby, 2) Promote standby to primary, and 3) Perform maintenance on old primary, which becomes the new standby.

When an Amazon RDS DB Instance deletion API (DeleteDBInstance) is run, the DB Instance is marked for deletion. Once the instance no longer indicates 'deleting' status, it has been removed. At this point the instance is no longer accessible and unless a final snapshot copy was asked for, it cannot be restored and will not be listed by any of the tools or APIs.

Event Notification

You can receive notifications of a variety of important events that can occur on your RDS instance, such as whether the instance was shut down, a backup was started, a failover occurred, the security group was changed, or your storage space is low. The Amazon RDS service groups events into categories that you

can subscribe to so that you can be notified when an event in that category occurs. You can subscribe to an event category for a DB instance, DB snapshot, DB security group, or for a DB parameter group. RDS events are published via AWS SNS and sent to you as an email or text message. For more information about RDS notification event categories, refer to the Amazon Relational Database Service User Guide.

Amazon Redshift Security

Amazon Redshift is a petabyte-scale SQL data warehouse service that runs on highly optimized and managed AWS compute and storage resources. The service has been architected to not only scale up or down rapidly, but to significantly improve query speeds – even on extremely large datasets. To increase performance, Redshift uses techniques such as columnar storage, data compression, and zone maps to reduce the amount of IO needed to perform queries. It also has a massively parallel processing (MPP) architecture, parallelizing and distributing SQL operations to take advantage of all available resources.

When you create a Redshift data warehouse, you provision a single-node or multi-node cluster, specifying the type and number of nodes that will make up the cluster. The node type determines the storage size, memory, and CPU of each node. Each multi-node cluster includes a leader node and two or more compute nodes. A leader node manages connections, parses queries, builds execution plans, and manages query execution in the compute nodes. The compute nodes store data, perform computations, and run queries as directed by the leader node. The leader node of each cluster is accessible through ODBC and JDBC endpoints, using standard PostgreSQL drivers. The compute nodes run on a separate, isolated network and are never accessed directly.

After you provision a cluster, you can upload your dataset and perform data analysis queries by using common SQL- based tools and business intelligence applications.

Topics

Cluster Access

By default, clusters that you create are closed to everyone. Amazon Redshift enables you to configure firewall rules (security groups) to control network access to your data warehouse cluster. You can also run Redshift inside an Amazon VPC to isolate your data warehouse cluster in your own virtual network and connect it to your existing IT infrastructure using industry-standard encrypted IPsec VPN.

The AWS account that creates the cluster has full access to the cluster. Within your AWS account, you can use AWS IAM to create user accounts and manage permissions for those accounts. By using IAM, you can grant different users permission to perform only the cluster operations that are necessary for their work.

Like all databases, you must grant permission in Redshift at the database level in addition to granting access at the resource level. Database users are named user accounts that can connect to a database and are authenticated when they login to Amazon Redshift. In Redshift, you grant database user permissions on a per-cluster basis instead of on a per-table basis. However, a user can see data only in the table rows that were generated by his own activities; rows generated by other users are not visible to him.

The user who creates a database object is its owner. By default, only a superuser or the owner of an object can query, modify, or grant permissions on the object. For users to use an object, you must grant the necessary permissions to the user or the group that contains the user. And only the owner of an object can modify or delete it.

Data Backups

Amazon Redshift distributes your data across all compute nodes in a cluster. When you run a cluster with at least two compute nodes, data on each node will always be mirrored on disks on another node, reducing the risk of data loss. In addition, all data written to a node in your cluster is continuously backed up to Amazon S3 using snapshots. Redshift stores your snapshots for a user-defined period, which can be from one to thirty-five days. You can also take your own snapshots at any time; these snapshots leverage all existing system snapshots and are retained until you explicitly delete them.

Amazon Redshift continuously monitors the health of the cluster and automatically re-replicates data from failed drives and replaces nodes as necessary. All of this happens without any effort on your part, although you may see a slight performance degradation during the re-replication process.

You can use any system or user snapshot to restore your cluster using the AWS Management Console or the Amazon Redshift APIs. Your cluster is available as soon as the system metadata has been restored and you can start running queries while user data is spooled down in the background.

Data Encryption

When creating a cluster, you can choose to encrypt it in order to provide additional protection for your data at rest. When you enable encryption in your cluster, Amazon Redshift stores all data in user-created tables in an encrypted format using hardware-accelerated AES-256 block encryption keys. This includes all data written to disk as well as any backups.

Amazon Redshift uses a four-tier, key-based architecture for encryption. These keys consist of data encryption keys, a database key, a cluster key, and a master key:

- ***Data encryption keys*** encrypt data blocks in the cluster. Each data block is assigned a randomly-generated AES- 256 key. These keys are encrypted by using the database key for the cluster.
- The ***database key*** encrypts data encryption keys in the cluster. The database key is a randomly-generated AES- 256 key. It is stored on disk in a separate network from the Amazon Redshift cluster and passed to the cluster across a secure channel.
- The ***cluster key*** encrypts the database key for the Amazon Redshift cluster. You can use either AWS or a hardware security module (HSM) to store the cluster key. HSMs provide direct control of key generation and management, and make key management separate and distinct from the application and the database.
- The ***master key*** encrypts the cluster key if it is stored in AWS. The master key encrypts the cluster-key-encrypted database key if the cluster key is stored in an HSM.

You can have Redshift rotate the encryption keys for your encrypted clusters at any time. As part of the rotation process, keys are also updated for all of the cluster's automatic and manual snapshots.

Note: Enabling encryption in your cluster will impact performance, even though it is hardware accelerated. Encryption also applies to backups. When restoring from an encrypted snapshot, the new cluster will be encrypted as well.

To encrypt your table load data files when you upload them to Amazon S3, you can use Amazon S3 server-side encryption. When you load the data from Amazon S3, the COPY command will decrypt the data as it loads the table.

Database Audit Logging

Amazon Redshift logs all SQL operations, including connection attempts, queries, and changes to your database. You can access these logs using SQL queries against system tables or choose to have them downloaded to a secure Amazon S3 bucket. You can then use these audit logs to monitor your cluster for security and troubleshooting purposes.

Automatic Software Patching

Amazon Redshift manages all the work of setting up, operating, and scaling your data warehouse, including provisioning capacity, monitoring the cluster, and applying patches and upgrades to the Amazon Redshift engine. Patches are applied only during specified maintenance windows.

SSL Connections

To protect your data in transit within the AWS cloud, Amazon Redshift uses hardware-accelerated SSL to communicate with Amazon S3 or Amazon DynamoDB for COPY, UNLOAD, backup, and restore operations. You can encrypt the connection between your client and the cluster by specifying SSL in the parameter group associated with the cluster. To have your clients also authenticate the Redshift server, you can install the public key (.pem file) for the SSL certificate on your client and use the key to connect to your clusters.

Amazon Redshift offers the newer, stronger cipher suites that use the Elliptic Curve Diffie-Hellman Ephemeral protocol. ECDHE allows SSL clients to provide Perfect Forward Secrecy between the client and the Redshift cluster. Perfect Forward Secrecy uses session keys that are ephemeral and not stored anywhere, which prevents the decoding of captured data by unauthorized third parties, even if the secret long-term key itself is compromised. You do not need to configure anything in Amazon Redshift to enable ECDHE; if you connect from a SQL client tool that uses ECDHE to encrypt communication between the client and server, Amazon Redshift will use the provided cipher list to make the appropriate connection.

Amazon ElastiCache Security

Amazon ElastiCache is a web service that makes it easy to set up, manage, and scale distributed in-memory cache environments in the cloud. The service improves the performance of web applications by allowing you to retrieve information from a fast, managed, in-memory caching system, instead of relying entirely on slower disk-based databases. It can be used to significantly improve latency and throughput for many read-heavy application workloads (such as social networking, gaming, media sharing, and Q&A portals) or compute-intensive workloads (such as a recommendation engine). Caching improves application performance by storing critical pieces of data in memory for low-latency access. Cached information may include the results of I/O-intensive database queries or the results of computationally-intensive calculations.

The Amazon ElastiCache service automates time-consuming management tasks for in-memory cache environments, such as patch management, failure detection, and recovery. It works in conjunction with other Amazon Web Services (such as Amazon EC2, Amazon CloudWatch, and Amazon SNS) to provide a secure, high-performance, and managed in- memory cache. For example, an application running in Amazon EC2 can securely access an Amazon ElastiCache Cluster in the same region with very low latency.

Using the Amazon ElastiCache service, you create a Cache Cluster, which is a collection of one or more Cache Nodes, each running an instance of the Memcached service. A Cache Node is a fixed-size chunk of secure, network- attached RAM. Each Cache Node runs an instance of the Memcached service, and has its own DNS name and port. Multiple types of Cache Nodes are supported, each with varying amounts of associated memory. A Cache Cluster can be set up with a specific number of Cache Nodes and a Cache Parameter Group that controls the properties for each Cache Node. All Cache Nodes within a Cache Cluster are designed to be of the same Node Type and have the same parameter and security group settings.

Amazon ElastiCache allows you to control access to your Cache Clusters using Cache Security Groups. A Cache Security Group acts like a firewall, controlling network access to your Cache Cluster. By default, network access is turned off to your Cache Clusters. If you want your applications to access your Cache Cluster, you must explicitly enable access from hosts in specific EC2 security groups. Once ingress rules are configured, the same rules apply to all Cache Clusters associated with that Cache Security Group.

To allow network access to your Cache Cluster, create a Cache Security Group and use the Authorize Cache Security Group Ingress API or CLI command to authorize the desired EC2 security group (which in turn specifies the EC2 instances allowed). IP-range based access control is currently not enabled for Cache Clusters. All clients to a Cache Cluster must be within the EC2 network, and authorized via Cache Security Groups.

ElastiCache for Redis provides backup and restore functionality, where you can create a snapshot of your entire Redis cluster as it exists at a specific point in time. You can schedule automatic, recurring daily snapshots or you can create a manual snapshot at any time. For automatic snapshots, you specify a retention period; manual snapshots are retained until you delete them. The snapshots are stored in Amazon S3 with high durability, and can be used for warm starts, backups, and archiving.

Application Services

Amazon Web Services offers a variety of managed services to use with your applications, including services that provide application streaming, queueing, push notification, email delivery, search, and transcoding.

Topics

Amazon CloudSearch Security

Amazon CloudSearch is a managed service in the cloud that makes it easy to set up, manage, and scale a search solution for your website. Amazon CloudSearch enables you to search large collections of data such as web pages, document files, forum posts, or product information. It enables you to quickly add search capabilities to your website without having to become a search expert or worry about hardware provisioning, setup, and maintenance. As your volume of data and traffic fluctuates, Amazon CloudSearch automatically scales to meet your needs.

An Amazon CloudSearch domain encapsulates a collection of data you want to search, the search instances that process your search requests, and a configuration that controls how your data is indexed and searched. You create a separate search domain for each collection of data you want to make searchable. For each domain, you configure indexing options that describe the fields you want to include in your index and how you want to us them, text options that define domain-specific stopwords, stems, and synonyms, rank expressions that you can use to customize how search results are ranked, and access policies that control access to the domain's document and search endpoints.

All Amazon CloudSearch configuration requests must be authenticated using standard AWS authentication.

Amazon CloudSearch provides separate endpoints for accessing the configuration, search, and document services:

- The configuration service is accessed through a general endpoint, such as: cloudsearch.us-east-1.amazonaws.com

- The document service endpoint is used to submit documents to the domain for indexing and is accessed through a domain-specific endpoint, such as: http://doc-*domainname*-*domainid*.us-east-1.cloudsearch.amazonaws.com/
- The search endpoint is used to submit search requests to the domain and is accessed through a domain-specific endpoint, such as: http://search-*domainname*-*domainid*.us-east-1.cloudsearch.amazonaws.com

Like all AWS Services, Amazon CloudSearch requires that every request made to its control API be authenticated so only authenticated users can access and manage your CloudSearch domain. API requests are signed with an HMAC- SHA1 or HMAC-SHA256 signature calculated from the request and the user's AWS Secret Access key. Additionally, the Amazon CloudSearch control API is accessible via SSL-encrypted endpoints. You can control access to Amazon CloudSearch management functions by creating users under your AWS Account using AWS IAM, and controlling which CloudSearch operations these users have permission to perform.

Amazon Simple Queue Service (Amazon SQS) Security

Amazon SQS is a highly reliable, scalable message queuing service that enables asynchronous message-based communication between distributed components of an application. The components can be computers or Amazon EC2 instances or a combination of both. With Amazon SQS, you can send any number of messages to an Amazon SQS queue at any time from any component. The messages can be retrieved from the same component or a different one right away or at a later time (within 4 days). Messages are highly durable; each message is persistently stored in highly available, highly reliable queues. Multiple processes can read/write from/to an Amazon SQS queue at the same time without interfering with each other.

Amazon SQS access is granted based on an AWS Account or a user created with AWS IAM. Once authenticated, the AWS Account has full access to all user operations. An AWS IAM user, however, only has access to the operations and queues for which they have been granted access via policy. By default, access to each individual queue is restricted to the AWS Account that created it. However, you can allow other access to a queue, using either an SQS-generated policy or a policy you write.

Amazon SQS is accessible via SSL-encrypted endpoints. The encrypted endpoints are accessible from both the Internet and from within Amazon EC2.

Data stored within Amazon SQS is not encrypted by AWS; however, the user can encrypt data before it is uploaded to Amazon SQS, provided that the application utilizing the queue has a means to decrypt the message when retrieved. Encrypting messages before sending them to Amazon SQS helps protect against access to sensitive customer data by unauthorized persons, including AWS.

Amazon Simple Notification Service (Amazon SNS) Security

Amazon Simple Notification Service (Amazon SNS) is a web service that makes it easy to set up, operate, and send notifications from the cloud. It provides developers with a highly scalable, flexible, and cost-effective capability to publish messages from an application and immediately deliver them to subscribers or other applications.

Amazon SNS provides a simple web services interface that can be used to create topics that customers want to notify applications (or people) about, subscribe clients to these topics, publish messages, and have these messages delivered over clients' protocol of choice (i.e., HTTP/HTTPS, email, etc.).

Amazon SNS delivers notifications to clients using a "push" mechanism that eliminates the need to periodically check or "poll" for new information and updates. Amazon SNS can be leveraged to build

highly reliable, event-driven workflows and messaging applications without the need for complex middleware and application management. The potential uses for Amazon SNS include monitoring applications, workflow systems, time-sensitive information updates, mobile applications, and many others. Amazon SNS provides access control mechanisms so that topics and messages are secured against unauthorized access. Topic owners can set policies for a topic that restrict who can publish or subscribe to a topic. Additionally, topic owners can encrypt transmission by specifying that the delivery mechanism must be HTTPS.

Amazon SNS access is granted based on an AWS Account or a user created with AWS IAM. Once authenticated, the AWS Account has full access to all user operations. An AWS IAM user, however, only has access to the operations and topics for which they have been granted access via policy. By default, access to each individual topic is restricted to the AWS Account that created it. However, you can allow other access to SNS, using either an SNS-generated policy or a policy you write.

Amazon Simple Workflow Service (Amazon SWF) Security

The Amazon Simple Workflow Service (Amazon SWF) makes it easy to build applications that coordinate work across distributed components. Using Amazon SWF, you can structure the various processing steps in an application as “tasks” that drive work in distributed applications, and Amazon SWF coordinates these tasks in a reliable and scalable manner. Amazon SWF manages task execution dependencies, scheduling, and concurrency based on a developer’s application logic. The service stores tasks, dispatches them to application components, tracks their progress, and keeps their latest state.

Amazon SWF provides simple API calls that can be executed from code written in any language and run on your EC2 instances, or any of your machines located anywhere in the world that can access the Internet. Amazon SWF acts as a coordination hub with which your application hosts interact. You create desired workflows with their associated tasks and any conditional logic you wish to apply and store them with Amazon SWF.

Amazon SWF access is granted based on an AWS Account or a user created with AWS IAM. All actors that participate in the execution of a workflow— deciders, activity workers, workflow administrators— must be IAM users under the AWS Account that owns the Amazon SWF resources. You cannot grant users associated with other AWS Accounts access to your Amazon SWF workflows. An AWS IAM user, however, only has access to the workflows and resources for which they have been granted access via policy.

Amazon Simple Email Service (Amazon SES) Security

Amazon Simple Email Service (SES), built on Amazon’s reliable and scalable infrastructure, is a mail service that can both send and receive mail on behalf of your domain. Amazon SES helps you maximize email deliverability and stay informed of the delivery status of your emails. Amazon SES integrates with other AWS services, making it easy to send emails from applications being hosted on services such as Amazon EC2.

Unfortunately, with other email systems, it's possible for a spammer to falsify an email header and spoof the originating email address so that it appears as though the email originated from a different source. To mitigate these problems, Amazon SES requires users to verify their email address or domain in order to confirm that they own it and to prevent others from using it. To verify a domain, Amazon SES requires the sender to publish a DNS record that Amazon SES supplies as proof of control over the domain. Amazon SES periodically reviews domain verification status, and revokes verification in cases where it is no longer valid.

Amazon SES takes proactive steps to prevent questionable content from being sent, so that ISPs receive consistently high-quality email from our domains and therefore view Amazon SES as a trusted email origin. Below are some of the features that maximize deliverability and dependability for all of our senders:

- Amazon SES uses content-filtering technologies to help detect and block messages containing viruses or malware before they can be sent.
- Amazon SES maintains complaint feedback loops with major ISPs. Complaint feedback loops indicate which emails a recipient marked as spam. Amazon SES provides you access to these delivery metrics to help guide your sending strategy.
- Amazon SES uses a variety of techniques to measure the quality of each user's sending. These mechanisms help identify and disable attempts to use Amazon SES for unsolicited mail, and detect other sending patterns that would harm Amazon SES's reputation with ISPs, mailbox providers, and anti-spam services.
- Amazon SES supports authentication mechanisms such as Sender Policy Framework (SPF) and DomainKeys Identified Mail (DKIM). When you authenticate an email, you provide evidence to ISPs that you own the domain. Amazon SES makes it easy for you to authenticate your emails. If you configure your account to use Easy DKIM, Amazon SES will DKIM-sign your emails on your behalf, so you can focus on other aspects of your email-sending strategy. To ensure optimal deliverability, we recommend that you authenticate your emails.

As with other AWS services, you use security credentials to verify who you are and whether you have permission to interact with Amazon SES. For information about which credentials to use, see Using Credentials with Amazon SES. Amazon SES also integrates with AWS IAM so that you can specify which Amazon SES API actions a user can perform.

If you choose to communicate with Amazon SES through its SMTP interface, you are required to encrypt your connection using TLS. Amazon SES supports two mechanisms for establishing a TLS-encrypted connection: STARTTLS and TLS Wrapper. If you choose to communicate with Amazon SES over HTTP, then all communication will be protected by TLS through Amazon SES's HTTPS endpoint. When delivering email to its final destination, Amazon SES encrypts the email content with opportunistic TLS, if supported by the receiver.

Amazon Elastic Transcoder Service Security

The Amazon Elastic Transcoder service simplifies and automates what is usually a complex process of converting media files from one format, size, or quality to another. The Elastic Transcoder service converts standard-definition (SD) or high-definition (HD) video files as well as audio files. It reads input from an Amazon S3 bucket, transcodes it, and writes the resulting file to another Amazon S3 bucket. You can use the same bucket for input and output, and the buckets can be in any AWS region. The Elastic Transcoder accepts input files in a wide variety of web, consumer, and professional formats. Output file types include the MP3, MP4, OGG, TS, WebM, HLS using MPEG-2 TS, and Smooth Streaming using fmp4 container types, storing H.264 or VP8 video and AAC, MP3, or Vorbis audio.

You'll start with one or more input files, and create transcoding jobs in a type of workflow called a transcoding pipeline for each file. When you create the pipeline, you'll specify input and output buckets as well as an IAM role. Each job must reference a media conversion template called a transcoding preset, and will result in the generation of one or more output files. A preset tells the Elastic Transcoder what settings to use when processing a particular input file. You can specify many settings when you create a preset, including the sample rate, bit rate, resolution (output height and width), the number of reference and keyframes, a video bit rate, some thumbnail creation options, etc.

A best effort is made to start jobs in the order in which they're submitted, but this is not a hard guarantee and jobs typically finish out of order since they are worked on in parallel and vary in complexity. You can pause and resume any of your pipelines if necessary.

Elastic Transcoder supports the use of SNS notifications when it starts and finishes each job, and when it needs to tell you that it has detected an error or warning condition. The SNS notification parameters are associated with each pipeline. It can also use the List Jobs by Status function to find all of the jobs with a given status (e.g., "Completed") or the Read Job function to retrieve detailed information about a particular job.

Like all other AWS services, Elastic Transcoder integrates with AWS Identity and Access Management (IAM), which allows you to control access to the service and to other AWS resources that Elastic Transcoder requires, including Amazon S3 buckets and Amazon SNS topics. By default, IAM users have no access to Elastic Transcoder or to the resources that it uses. If you want IAM users to be able to work with Elastic Transcoder, you must explicitly grant them permissions.

Amazon Elastic Transcoder requires every request made to its control API be authenticated so only authenticated processes or users can create, modify, or delete their own Amazon Transcoder pipelines and presets. Requests are signed with an HMAC-SHA256 signature calculated from the request and a key derived from the user's secret key. Additionally, the Amazon Elastic Transcoder API is only accessible via SSL-encrypted endpoints.

Durability is provided by Amazon S3, where media files are redundantly stored on multiple devices across multiple facilities in an Amazon S3 region. For added protection against users accidently deleting media files, you can use the Versioning feature in Amazon S3 to preserve, retrieve, and restore every version of every object stored in an Amazon S3 bucket. You can further protect versions using Amazon S3 Versioning's MFA Delete feature. Once enabled for an Amazon S3 bucket, each version deletion request must include the six-digit code and serial number from your multi-factor authentication device.

Amazon AppStream 2.0 Security

The Amazon AppStream 2.0 service provides a framework for running streaming applications, particularly applications that require lightweight clients running on mobile devices. It enables you to store and run your application on powerful, parallel-processing GPUs in the cloud and then stream input and output to any client device. This can be a pre-existing application that you modify to work with Amazon AppStream 2.0 or a new application that you design specifically to work with the service.

The Amazon AppStream 2.0 SDK simplifies the development of interactive streaming applications and client applications. The SDK provides APIs that connect your customers' devices directly to your application, capture and encode audio and video, stream content across the Internet in near real-time, decode content on client devices, and return user input to the application.

Because your application's processing occurs in the cloud, it can scale to handle extremely large computational loads.

Amazon AppStream 2.0 deploys streaming applications on Amazon EC2. When you add a streaming application through the AWS Management Console, the service creates the AMI required to host your application and makes your application available to streaming clients. The service scales your application as needed within the capacity limits you have set to meet demand. Clients using the Amazon AppStream 2.0 SDK automatically connect to your streamed application.

In most cases, you'll want to ensure that the user running the client is authorized to use your application before letting him obtain a session ID. We recommend that you use some sort of entitlement service, which is a service that authenticates clients and authorizes their connection to your application. In this case, the entitlement service will also call into the Amazon AppStream 2.0 REST API to create a new streaming session for the client. After the entitlement service creates a new session, it returns the session identifier to the authorized client as a single-use entitlement URL. The client then uses the entitlement URL to connect to the application. Your entitlement service can be hosted on an Amazon EC2 instance or on AWS Elastic Beanstalk.

Amazon AppStream 2.0 utilizes an AWS CloudFormation template that automates the process of deploying a GPU EC2 instance that has the AppStream 2.0 Windows Application and Windows Client SDK libraries installed; is configured for SSH, RDC, or VPN access; and has an elastic IP address assigned to it. By using this template to deploy your standalone streaming server, all you need to do is upload your application to the server and run the command to launch it.

You can then use the Amazon AppStream 2.0 Service Simulator tool to test your application in standalone mode before deploying it into production.

Amazon AppStream 2.0 also utilizes the STX Protocol to manage the streaming of your application from AWS to local devices. The Amazon AppStream 2.0 STX Protocol is a proprietary protocol used to stream high-quality application video over varying network conditions; it monitors network conditions and automatically adapts the video stream to provide a low-latency and high- resolution experience to your customers. It minimizes latency while syncing audio and video as well as capturing input from your customers to be sent back to the application running in AWS.

Analytics Services

Amazon Web Services provides cloud-based analytics services to help you process and analyze any volume of data, whether your need is for managed Hadoop clusters, real-time streaming data, petabyte scale data warehousing, or orchestration.

Topics

Amazon EMR Security

Amazon EMR is a managed web service you can use to run Hadoop clusters that process vast amounts of data by distributing the work and data among several servers. It utilizes an enhanced version of the Apache Hadoop framework running on the web-scale infrastructure of Amazon EC2 and Amazon S3. You simply upload your input data and a data processing application into Amazon S3. Amazon EMR then launches the number of Amazon EC2 instances you specify. The service begins the job flow execution while pulling the input data from Amazon S3 into the launched Amazon EC2 instances. Once the job flow is finished, Amazon EMR transfers the output data to Amazon S3, where you can then retrieve it or use it as input in another job flow.

When launching job flows on your behalf, Amazon EMR sets up two Amazon EC2 security groups: one for the master nodes and another for the slaves. The master security group has a port open for communication with the service. It also has the SSH port open to allow you to SSH into the instances, using the key specified at startup. The slaves start in a separate security group, which only allows interaction with the master instance. By default, both security groups are set up to not allow access from external sources, including Amazon EC2 instances belonging to other customers. Since these are security groups within your account, you can reconfigure them using the standard EC2 tools or dashboard. To protect customer input and output datasets, Amazon EMR transfers data to and from Amazon S3 using SSL.

Amazon EMR provides several ways to control access to the resources of your cluster. You can use AWS IAM to create user accounts and roles and configure permissions that control which AWS features those users and roles can access. When you launch a cluster, you can associate an Amazon EC2 key pair with the cluster, which you can then use when you connect to the cluster using SSH. You can also set permissions that allow users other than the default Hadoop user to submit jobs to your cluster.

By default, if an IAM user launches a cluster, that cluster is hidden from other IAM users on the AWS account. This filtering occurs on all Amazon EMR interfaces—the console, CLI, API, and SDKs—and helps prevent IAM users from accessing and inadvertently changing clusters created by other IAM users.

It is useful for clusters that are intended to be viewed by only a single IAM user and the main AWS account. You also have the option to make a cluster visible and accessible to all IAM users under a single AWS account.

For an additional layer of protection, you can launch the EC2 instances of your EMR cluster into an Amazon VPC, which is like launching it into a private subnet. This allows you to control access to the

entire subnetwork. You can also launch the cluster into a VPC and enable the cluster to access resources on your internal network using a VPN connection. You can encrypt the input data before you upload it to Amazon S3 using any common data encryption tool. If you do encrypt the data before it's uploaded, you then need to add a decryption step to the beginning of your job flow when Amazon Elastic MapReduce fetches the data from Amazon S3.

Amazon Kinesis Security

Amazon Kinesis is a managed service designed to handle real-time streaming of big data. It can accept any amount of data, from any number of sources, scaling up and down as needed. You can use Kinesis in situations that call for large-scale, real-time data ingestion and processing, such as server logs, social media or market data feeds, and web clickstream data.

Applications read and write data records to Amazon Kinesis in streams. You can create any number of Kinesis streams to capture, store, and transport data. Amazon Kinesis automatically manages the infrastructure, storage, networking, and configuration needed to collect and process your data at the level of throughput your streaming applications need. You don't have to worry about provisioning, deployment, or ongoing-maintenance of hardware, software, or other services to enable real-time capture and storage of large-scale data.

Amazon Kinesis also synchronously replicates data across three facilities in an AWS Region, providing high availability and data durability.

In Amazon Kinesis, data records contain a sequence number, a partition key, and a data blob, which is an un-interpreted, immutable sequence of bytes. The Amazon Kinesis service does not inspect, interpret, or change the data in the blob in any way. Data records are accessible for only 24 hours from the time they are added to an Amazon Kinesis stream, and then they are automatically discarded.

Your application is a consumer of an Amazon Kinesis stream, which typically runs on a fleet of Amazon EC2 instances. A Kinesis application uses the Amazon Kinesis Client Library to read from the Amazon Kinesis stream. The Kinesis Client Library takes care of a variety of details for you including failover, recovery, and load balancing, allowing your application to focus on processing the data as it becomes available. After processing the record, your consumer code can pass it along to another Kinesis stream; write it to an Amazon S3 bucket, a Redshift data warehouse, or a DynamoDB table; or simply discard it. A connector library is available to help you integrate Kinesis with other AWS services (such as DynamoDB, Redshift, and Amazon S3) as well as third-party products like Apache Storm.

You can control logical access to Kinesis resources and management functions by creating users under your AWS Account using AWS IAM, and controlling which Kinesis operations these users have permission to perform. To facilitate running your producer or consumer applications on an Amazon EC2 instance, you can configure that instance with an IAM role. That way, AWS credentials that reflect the permissions associated with the IAM role are made available to applications on the instance, which means you don't have to use your long-term AWS security credentials. Roles have the added benefit of providing temporary credentials that expire within a short timeframe, which adds an additional measure of protection. See the AWS Identity and Access Management User Guide for more information about IAM roles.

The Amazon Kinesis API is only accessible via an SSL-encrypted endpoint (kinesis.us-east-1.amazonaws.com) to help ensure secure transmission of your data to AWS. You must connect to that endpoint to access Kinesis, but you can then use the API to direct AWS Kinesis to create a stream in any AWS Region.

AWS Data Pipeline Security

The AWS Data Pipeline service helps you process and move data between different data sources at specified intervals using data-driven workflows and built-in dependency checking. When you create

a pipeline, you define data sources, preconditions, destinations, processing steps, and an operational schedule. Once you define and activate a pipeline, it will run automatically according to the schedule you specified.

With AWS Data Pipeline, you don't have to worry about checking resource availability, managing inter-task dependencies, retrying transient failures/timeouts in individual tasks, or creating a failure notification system. AWS Data Pipeline takes care of launching the AWS services and resources your pipeline needs to process your data (e.g., Amazon EC2 or EMR) and transferring the results to storage (e.g., Amazon S3, RDS, DynamoDB, or EMR).

When you use the console, AWS Data Pipeline creates the necessary IAM roles and policies, including a trusted entities list for you. IAM roles determine what your pipeline can access and the actions it can perform. Additionally, when your pipeline creates a resource, such as an EC2 instance, IAM roles determine the EC2 instance's permitted resources and actions. When you create a pipeline, you specify one IAM role that governs your pipeline and another IAM role to govern your pipeline's resources (referred to as a "resource role"), which can be the same role for both. As part of the security best practice of least privilege, we recommend that you consider the minimum permissions necessary for your pipeline to perform work and define the IAM roles accordingly.

Like most AWS services, AWS Data Pipeline also provides the option of secure (HTTPS) endpoints for access via SSL.

Deployment and Management Services

Amazon Web Services provides a variety of tools to help with the deployment and management of your applications. This includes services that allow you to create individual user accounts with credentials for access to AWS services. It also includes services for creating and updating stacks of AWS resources, deploying applications on those resources, and monitoring the health of those AWS resources. Other tools help you manage cryptographic keys using hardware security modules (HSMs) and log AWS API activity for security and compliance purposes.

Topics

AWS Identity and Access Management (IAM)

IAM allows you to create multiple users and manage the permissions for each of these users within your AWS Account. A user is an identity (within an AWS Account) with unique security credentials that can be used to access AWS Services. IAM eliminates the need to share passwords or keys, and makes it easy to enable or disable a user's access as appropriate.

IAM enables you to implement security best practices, such as least privilege, by granting unique credentials to every user within your AWS Account and only granting permission to access the AWS services and resources required for the users to perform their jobs. IAM is secure by default; new users have no access to AWS until permissions are explicitly granted.

IAM is also integrated with the AWS Marketplace, so that you can control who in your organization can subscribe to the software and services offered in the Marketplace. Since subscribing to certain software in the Marketplace launches an EC2 instance to run the software, this is an important access control

feature. Using IAM to control access to the AWS Marketplace also enables AWS Account owners to have fine-grained control over usage and software costs.

IAM enables you to minimize the use of your AWS Account credentials. Once you create IAM user accounts, all interactions with AWS Services and resources should occur with IAM user security credentials.

Topics

Roles

An IAM role uses temporary security credentials to allow you to delegate access to users or services that normally don't have access to your AWS resources. A role is a set of permissions to access specific AWS resources, but these permissions are not tied to a specific IAM user or group. An authorized entity (e.g., mobile user, EC2 instance) assumes a role and receives temporary security credentials for authenticating to the resources defined in the role.

Temporary security credentials provide enhanced security due to their short life- span (the default expiration is 12 hours) and the fact that they cannot be reused after they expire. This can be particularly useful in providing limited, controlled access in certain situations:

- **Federated (non-AWS) User Access.** Federated users are users (or applications) who do not have AWS Accounts. With roles, you can give them access to your AWS resources for a limited amount of time. This is useful if you have non-AWS users that you can authenticate with an external service, such as Microsoft Active Directory, LDAP, or Kerberos. The temporary AWS credentials used with the roles provide identity federation between AWS and your non-AWS users in your corporate identity and authorization system.

If your organization supports SAML 2.0 (Security Assertion Markup Language 2.0), you can create trust between your organization as an identity provider (IdP) and other organizations as service providers. In AWS, you can configure AWS as the service provider and use SAML to provide your users with federated single-sign on (SSO) to the AWS Management Console or to get federated access to call AWS APIs.

Roles are also useful if you create a mobile or web-based application that accesses AWS resources. AWS resources require security credentials for programmatic requests; however, you shouldn't embed long-term security credentials in your application because they are accessible to the application's users and can be difficult to rotate.

Instead, you can let users sign in to your application using Login with Amazon, Facebook, or Google, and then use their authentication information to assume a role and get temporary security credentials.

- **Cross-Account Access.** For organizations who use multiple AWS Accounts to manage their resources, you can set up roles to provide users who have permissions in one account to access resources under another account. For organizations who have personnel who only rarely need access to resources under another account, using roles helps ensures that credentials are provided temporarily, only as needed.
- **Applications Running on EC2 Instances that Need to Access AWS Resources.** If an application runs on an Amazon EC2 instance and needs to make requests for AWS resources such as Amazon S3 buckets or a DynamoDB table, it must have security credentials. Using roles instead of creating individual IAM accounts for each application on each instance can save significant time for customers who manage a large number of instances or an elastically scaling fleet using AWS Auto Scaling.

The temporary credentials include a security token, an Access Key ID, and a Secret Access Key. To give a user access to certain resources, you distribute the temporary security credentials to the user you are granting temporary access to. When the user makes calls to your resources, the user passes in the token and Access Key ID, and signs the request with the Secret Access Key.

The token will not work with different access keys. How the user passes in the token depends on the API and version of the AWS product the user is making calls to. For more information about temporary security credentials, see AWS Security Token Service API Reference.

The use of temporary credentials means additional protection for you because you don't have to manage or distribute long-term credentials to temporary users. In addition, the temporary credentials get automatically loaded to the target instance so you don't have to embed them somewhere unsafe like your code. Temporary credentials are automatically rotated or changed multiple times a day without any action on your part, and are stored securely by default.

For more information about using IAM roles to auto-provision keys on EC2 instances, see the AWS Identity and Access Management Documentation.

Amazon CloudWatch Security

Amazon CloudWatch is a web service that provides monitoring for AWS cloud resources, starting with Amazon EC2. It provides customers with visibility into resource utilization, operational performance, and overall demand patterns—including metrics such as CPU utilization, disk reads and writes, and network traffic. You can set up CloudWatch alarms to notify you if certain thresholds are crossed, or to take other automated actions such as adding or removing EC2 instances if Auto Scaling is enabled.

CloudWatch captures and summarizes utilization metrics natively for AWS resources, but you can also have other logs sent to CloudWatch to monitor. You can route your guest OS, application, and custom log files for the software installed on your EC2 instances to CloudWatch, where they will be stored in durable fashion for as long as you'd like. You can configure CloudWatch to monitor the incoming log entries for any desired symbols or messages and to surface the results as CloudWatch metrics. You could, for example, monitor your web server's log files for 404 errors to detect bad inbound links or invalid user messages to detect unauthorized login attempts to your guest OS.

Like all AWS Services, Amazon CloudWatch requires that every request made to its control API be authenticated so only authenticated users can access and manage CloudWatch. Requests are signed with an HMAC-SHA1 signature calculated from the request and the user's private key. Additionally, the Amazon CloudWatch control API is only accessible via SSL- encrypted endpoints.

You can further control access to Amazon CloudWatch by creating users under your AWS Account using AWS IAM, and controlling what CloudWatch operations these users have permission to call.

AWS CloudHSM Security

The AWS CloudHSM service provides customers with dedicated access to a hardware security module (HSM) appliance designed to provide secure cryptographic key storage and operations within an intrusion-resistant, tamper- evident device. You can generate, store, and manage the cryptographic keys used for data encryption so that they are accessible only by you. AWS CloudHSM appliances are designed to securely store and process cryptographic key material for a wide variety of uses such as database encryption, Digital Rights Management (DRM), Public Key Infrastructure (PKI), authentication and authorization, document signing, and transaction processing. They support some of the strongest cryptographic algorithms available, including AES, RSA, and ECC, and many others.

The AWS CloudHSM service is designed to be used with Amazon EC2 and VPC, providing the appliance with its own private IP within a private subnet. You can connect to CloudHSM appliances from your EC2 servers through SSL/TLS, which uses two-way digital certificate authentication and 256-bit SSL encryption to provide a secure communication channel.

Selecting CloudHSM service in the same region as your EC2 instance decreases network latency, which can improve your application performance. You can configure a client on your EC2 instance that allows your applications to use the APIs provided by the HSM, including PKCS#11, MS CAPI and Java JCA/JCE (Java Cryptography Architecture/Java Cryptography Extensions).

Before you begin using an HSM, you must set up at least one partition on the appliance. A cryptographic partition is a logical and physical security boundary that restricts access to your keys, so only you control your keys and the operations performed by the HSM. AWS has administrative credentials to the appliance, but these credentials can only be used to manage the appliance, not the HSM partitions on the appliance. AWS uses these credentials to monitor and maintain the health and availability of the appliance. AWS cannot extract your keys nor can AWS cause the appliance to perform any cryptographic operation using your keys.

The HSM appliance has both physical and logical tamper detection and response mechanisms that erase the cryptographic key material and generate event logs if tampering is detected. The HSM is designed to detect tampering if the physical barrier of the HSM appliance is breached. In addition, after three unsuccessful attempts to access an HSM partition with HSM Admin credentials, the HSM appliance erases its HSM partitions.

When your CloudHSM subscription ends and you have confirmed that the contents of the HSM are no longer needed, you must delete each partition and its contents as well as any logs. As part of the decommissioning process, AWS zeroizes the appliance, permanently erasing all key material.

AWS CloudTrail Security

AWS CloudTrail provides a log of user and system actions affecting AWS resources within your account. For each event recorded, you can see what service was accessed, what action was performed, any parameters for the action, and who made the request. For mutating actions, you can see the result of the action. Not only can you see which one of your users or services performed an action on an AWS service, but you can see whether it was as the AWS root account user or an IAM user, or whether it was with temporary security credentials for a role or federated user.

CloudTrail captures information about API calls to an AWS resource, whether that call was made from the AWS Management Console, CLI, or an SDK. If the API request returned an error, CloudTrail provides the description of the error, including messages for authorization failures. It even captures AWS Management Console sign-in events, creating a log record every time an AWS account owner, a federated user, or an IAM user simply signs into the console.

Once you have enabled CloudTrail, event logs are delivered about every 5 minutes to the Amazon S3 bucket of your choice. The log files are organized by AWS Account ID, region, service name, date, and time. You can configure CloudTrail so that it aggregates log files from multiple regions and/or accounts into a single Amazon S3 bucket. By default, a single trail will record and deliver events in all current and future regions. In addition to S3, you can send events to CloudWatch Logs, for custom metrics and alarming, or you can upload the logs to your favorite log management and analysis solutions to perform security analysis and detect user behavior patterns. For rapid response, you can create CloudWatch Events rules to take immediate action to specific events.

By default, log files are stored indefinitely. The log files are automatically encrypted using Amazon S3's Server Side Encryption and will remain in the bucket until you choose to delete or archive them. For even more security you can use KMS to encrypt the log files using a key that you own. You can use Amazon S3 lifecycle configuration rules to automatically delete old log files or archive them to Amazon S3 Glacier for additional longevity at significant savings.

By enabling the optional log file validation, you can validate that logs have not been added, deleted, or tampered with.

Like every other AWS service, you can limit access to CloudTrail to only certain users. You can use IAM to control which AWS users can create, configure, or delete AWS CloudTrail trails as well as which users can start and stop logging. You can control access to the log files by applying IAM or Amazon S3 bucket policies. You can also add an additional layer of security by enabling MFA Delete on your Amazon S3 bucket.

Mobile Services

AWS mobile services make it easier for you to build, ship, run, monitor, optimize, and scale cloud-powered applications for mobile devices. These services also help you authenticate users to your mobile application, synchronize data, and collect and analyze application usage.

Topics

Amazon Cognito

Amazon Cognito provides identity and sync services for mobile and web-based applications. It simplifies the task of authenticating users and storing, managing, and syncing their data across multiple devices, platforms, and applications. It provides temporary, limited-privilege credentials for both authenticated and unauthenticated users without having to manage any backend infrastructure.

Amazon Cognito works with well-known identity providers like Google, Facebook, and Amazon to authenticate end users of your mobile and web applications. You can take advantage of the identification and authorization features provided by these services instead of having to build and maintain your own. Your application authenticates with one of these identity providers using the provider's SDK. Once the end user is authenticated with the provider, an OAuth or OpenID Connect token returned from the provider is passed by your application to Cognito, which returns a new Amazon Cognito ID for the user and a set of temporary, limited-privilege AWS credentials.

To begin using Amazon Cognito, you create an identity pool through the Amazon Cognito console. The identity pool is a store of user identity information that is specific to your AWS account. During the creation of the identity pool, you will be asked to create a new IAM role or pick an existing one for your end users. An IAM role is a set of permissions to access specific AWS resources, but these permissions are not tied to a specific IAM user or group. An authorized entity (e.g., mobile user, EC2 instance) assumes a role and receives temporary security credentials for authenticating to the AWS resources defined in the role. Temporary security credentials provide enhanced security due to their short life-span (the default expiration is 12 hours) and the fact that they cannot be reused after they expire. The role you select has an impact on which AWS services your end users will be able to access with the temporary credentials. By default, Amazon Cognito creates a new role with limited permissions – end users only have access to the Amazon Cognito Sync service and Amazon Mobile Analytics. If your application needs access to other AWS resources such as Amazon S3 or DynamoDB, you can modify your roles directly from the IAM management console.

With Amazon Cognito, there's no need to create individual AWS accounts or even IAM accounts for every one of your web/mobile app's end users who will need to access your AWS resources. In conjunction with IAM roles, mobile users can securely access AWS resources and application features, and even save data to the AWS cloud without having to create an account or log in.

However, if they choose to do this later, Amazon Cognito merges data and identification information. Because Amazon Cognito stores data locally as well as in the service, your end users can continue to interact with their data even when they are offline. Their offline data may be stale, but anything they put into the dataset, they can immediately retrieve whether they are online or not. The client SDK manages a local SQLite store so that the application can work even when it is not connected. The SQLite store functions as a cache and is the target of all read and write operations. Cognito's sync facility compares the local version of the data to the cloud version, and pushes up or pulls down deltas as needed. Note that in order to sync data across devices, your identity pool must support authenticated identities. Unauthenticated identities are tied to the device, so unless an end user authenticates, no data can be synced across multiple devices.

With Amazon Cognito, your application communicates directly with a supported public identity provider (Amazon, Facebook, or Google) to authenticate users. Amazon Cognito does not receive or store user credentials—only the OAuth or OpenID Connect token received from the identity provider. Once Amazon Cognito receives the token, it returns a new Amazon Cognito ID for the user and a set of temporary, limited-privilege AWS credentials.

Each Amazon Cognito identity has access only to its own data in the sync store, and this data is encrypted when stored. In addition, all identity data is transmitted over HTTPS. The unique Amazon Cognito identifier on the device is stored in the appropriate secure location—on iOS for example, the Amazon Cognito identifier is stored in the iOS keychain. User data is cached in a local SQLite database within the application's sandbox; if you require additional security, you can encrypt this identity data in the local cache by implementing encryption in your application.

Amazon Mobile Analytics

Amazon Mobile Analytics is a service for collecting, visualizing, and understanding mobile application usage data. It enables you to track customer behaviors, aggregate metrics, and identify meaningful patterns in your mobile applications. Amazon Mobile Analytics automatically calculates and updates usage metrics as the data is received from client devices running your app and displays the data in the console.

You can integrate Amazon Mobile Analytics with your application without requiring users of your app to be authenticated with an identity provider (like Google, Facebook, or Amazon). For these unauthenticated users, Mobile Analytics works with Amazon Cognito to provide temporary, limited-privilege credentials. To do this, you first create an identity pool in Amazon Cognito. The identity pool will use IAM roles, which is a set of permissions not tied to a specific IAM user or group but which allows an entity to access specific AWS resources. The entity assumes a role and receives temporary security credentials for authenticating to the AWS resources defined in the role. By default, Amazon Cognito creates a new role with limited permissions – end users only have access to the Amazon Cognito Sync service and Amazon Mobile Analytics. If your application needs access to other AWS resources such as Amazon S3 or DynamoDB, you can modify your roles directly from the IAM management console.

You can integrate the AWS Mobile SDK for Android or iOS into your application or use the Amazon Mobile Analytics REST API to send events from any connected device or service and visualize data in the reports. The Amazon Mobile Analytics API is only accessible via an SSL-encrypted endpoint (https://mobileanalytics.us-east-1.amazonaws.com).

Applications

AWS applications are managed services that enable you to provide your users with secure, centralized storage and work areas in the cloud.

Topics

Amazon WorkSpaces

Amazon WorkSpaces is a managed desktop service that allows you to quickly provision cloud-based desktops for your users. Simply choose a Windows 7 bundle that best meets the needs of your users and the number of WorkSpaces that you would like to launch. Once the WorkSpaces are ready, users receive an email informing them where they can download the relevant client and log into their WorkSpace. They can then access their cloud-based desktops from a variety of endpoint devices, including PCs, laptops, and mobile devices.

However, your organization's data is never sent to or stored on the end-user device because Amazon WorkSpaces uses PC-over-IP (PCoIP), which provides an interactive video stream without transmitting actual data. The

PCoIP protocol compresses, encrypts, and encodes the users' desktop computing experience and transmits 'pixels only' across any standard IP network to end- user devices.

In order to access their WorkSpace, users must sign in using a set of unique credentials or their regular Active Directory credentials. When you integrate Amazon WorkSpaces with your corporate Active Directory, each WorkSpace joins your Active Directory domain and can be managed just like any other desktop in your organization. This means that you can use Active Directory Group Policies to manage your users' WorkSpaces to specify configuration options that control the desktop. If you choose not to use Active Directory or other type of on-premises directory to manage your user WorkSpaces, you can create a private cloud directory within Amazon WorkSpaces that you can use for administration.

To provide an additional layer of security, you can also require the use of multi- factor authentication upon sign in in the form of a hardware or software token. Amazon WorkSpaces supports MFA using an on-premise Remote Authentication Dial in User Service (RADIUS) server or any security provider that supports RADIUS authentication. It currently supports the PAP, CHAP, MS- CHAP1, and MS-CHAP2 protocols, along with RADIUS proxies.

Each Workspace resides on its own EC2 instance within a VPC. You can create WorkSpaces in a VPC you already own or have the WorkSpaces service create one for you automatically using the WorkSpaces Quick Start option. When you use the Quick Start option, WorkSpaces not only creates the VPC, but it performs several other provisioning and configuration tasks for you, such as creating an Internet Gateway for the VPC, setting up a directory within the VPC that is used to store user and WorkSpace information, creating a directory administrator account, creating the specified user accounts and adding them to the directory, and creating the WorkSpace instances. Or the VPC can be connected to an on-premises network using a secure VPN connection to allow access to an existing on-premises Active Directory and other intranet resources. You can add a security group that you create in your Amazon VPC to all the WorkSpaces that belong to your Directory. This allows you to control network access from Amazon WorkSpaces in your VPC to other resources in your Amazon VPC and on-premises network.

Persistent storage for WorkSpaces is provided by Amazon EBS and is automatically backed up twice a day to Amazon S3. If WorkSpaces Sync is enabled on a WorkSpace, the folder a user chooses to sync will be continuously backed up and stored in Amazon S3. You can also use WorkSpaces Sync on a Mac or PC to sync documents to or from your WorkSpace so that you can always have access to your data regardless of the desktop computer you are using.

Because it's a managed service, AWS takes care of several security and maintenance tasks like daily backups and patching. Updates are delivered automatically to your WorkSpaces during a weekly maintenance window. You can control how patching is configured for a user's WorkSpace. By default, Windows Update is turned on, but you have the ability to customize these settings, or use an alternative patch management approach if you desire. For the underlying OS, Windows Update is enabled by default on WorkSpaces, and configured to install updates on a weekly basis. You can use an alternative patching approach or to configure Windows Update to perform updates at a time of your choosing.

You can use IAM to control who on your team can perform administrative functions like creating or deleting WorkSpaces or setting up user directories. You can also set up a WorkSpace for directory administration, install your favorite Active Directory administration tools, and create organizational units and Group Policies in order to more easily apply Active Directory changes for all your WorkSpaces users.

Amazon WorkDocs

Amazon WorkDocs is a managed enterprise storage and sharing service with feedback capabilities for user collaboration. Users can store any type of file in a WorkDocs folder and allow others to view and download them. Commenting and annotation capabilities work on certain file types such as MS Word, and without requiring the application that was used to originally create the file.

WorkDocs notifies contributors about review activities and deadlines via email and performs versioning of files that you have synced using the WorkDocs Sync application.

User information is stored in an Active Directory-compatible network directory. You can either create a new directory in the cloud, or connect Amazon WorkDocs to your on-premises directory. When you create a cloud directory using WorkDocs' quick start setup, it also creates a directory administrator account with the administrator email as the username. An email is sent to your administrator with instructions to complete registration. The administrator then uses this account to manage your directory.

When you create a cloud directory using WorkDocs' quick start setup, it also creates and configures a VPC for use with the directory. If you need more control over the directory configuration, you can choose the standard setup, which allows you to specify your own directory domain name, as well as one of your existing VPCs to use with the directory. If you want to use one of your existing VPCs, the VPC must have an Internet gateway and at least two subnets. Each of the subnets must be in a different Availability Zone.

Using the Amazon WorkDocs Management Console, administrators can view audit logs to track file and user activity by time, IP address, and device, and choose whether to allow users to share files with others outside their organization. Users can then control who can access individual files and disable downloads of files they share.

All data in transit is encrypted using industry-standard SSL. The WorkDocs web and mobile applications and desktop sync clients transmit files directly to Amazon WorkDocs using SSL. WorkDocs users can also utilize Multi-Factor Authentication, or MFA, if their organization has deployed a Radius server. MFA uses the following factors: username, password, and methods supported by the Radius server. The protocols supported are PAP, CHAP, MS-CHAPv1, and MS- CHAPv2.

You choose the AWS Region where each WorkDocs site's files are stored. Amazon WorkDocs is currently available in the US-East (Virginia), US-West (Oregon), and EU (Ireland) AWS Regions. All files, comments, and annotations stored in WorkDocs are automatically encrypted with AES-256 encryption.

Document Revisions

To be notified about updates to this whitepaper, subscribe to the RSS feed.

update-history-change	update-history-description	update-history-date
Whitepaper updated (p. 64)	Updated compliance certifications, hypervisor, AWS Snowball.	March 1, 2020
Whitepaper updated (p. 64)	Added information about deleting objects in Amazon S3 Glacier.	February 1, 2019
Whitepaper updated (p. 64)	Edit made to the Amazon Redshift Security topic.	December 1, 2018
Whitepaper updated (p. 64)	Added section on AWS Config Security Checks.	May 1, 2017
Whitepaper updated (p. 64)	Added section on Amazon Elastic File System.	April 1, 2017
Whitepaper updated (p. 64)	Migrated into new format.	March 1, 2017
Whitepaper updated (p. 64)	Updated regions.	January 1, 2017

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.