Logical Separation on AWS

Moving Beyond Physical Isolation in the Era of Cloud Computing

July 2020

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.

Contents

Abstract

This paper examines the topic of logical separation for customers using Amazon Web Services (AWS). The paper discusses using a multi-pronged approach — for example, leveraging virtualization, encryption, and programmatic policies — to build logical security mechanisms that meet and often exceed the security results of physical separation and other on-premises security approaches. Public sector and commercial organizations worldwide can leverage these mechanisms to more confidently migrate sensitive workloads to the cloud without the need for physically dedicated infrastructure.

Introduction

Cloud technology takes advantage of transformative techniques in information technology (IT). One fundamental technique is to offer multi-tenant services that place multiple customers' applications and data on the same physical infrastructure. This architecture allows cloud service providers (CSPs), like AWS, to maximize use of physical resources so they can offer the value of those resources at a lower cost to customers. It also allows customers to easily update and migrate their workloads with minimal disruption to the latest technology as it continually makes its way into the CSP's infrastructure. This architectural choice is enabled by the development of powerful and flexible logical security controls that create strong isolation boundaries between customers. Since launching its first cloud services in 2006, AWS has been steadily enhancing its features and controls so that customers can achieve the security posture required to meet their data classification requirements. Customers often find that CSPs, like AWS, enable them to effectively optimize security configurations in the cloud compared to their on-premises solutions.

Customers leveraging AWS can benefit from a data center, network, and software architecture built to satisfy the requirements of the most security-sensitive organizations in the world. AWS provides highly available services and supports a combination of traditional and novel security mechanisms that are intrinsic to its service design and operation.

AWS gives customers rich control over their content and provides tools to determine where their content will be stored and how it will be protected. AWS features provide customers the ability to secure their content in transit and at rest, to tightly control access to AWS services and resources for their users, and to monitor access as well as the evolving state of their systems. Customers of AWS maintain full control over access to their content, which enables architecture to prevent unauthorized users from accessing customer data. All this occurs within a framework of multi-tenant services with strict logical isolation. The logical isolation between customer environments provided by AWS can be more effective and reliable than security seen in dedicated physical infrastructure.

Drivers for Physical Separation Requirements

Requirements for physically dedicated environments are primarily driven by concerns around third-party or unauthorized access to systems, applications, or data. There is a general misunderstanding that physically separated environments will provide better protection against unintended information or system disclosure, tampering, and unauthorized access compared to logically separated multi-tenant cloud environments. However, when examining the most common attack vectors for unauthorized access — such as remote exploitation, human error, and insider threat — a physically separated environment does not reduce the risk profile. In fact, for any system that is accessible over a network or the Internet, physical separation — such as placing them in a locked cage or a separate data center facility — does not inherently provide added security or control over the most important forms of access.

Additionally, smaller physically separated environments do not have parity with generally available cloud environments; hence any physical separation requirement can limit or delay a customer's ability to leverage innovative investments (including security feature innovations) made on behalf of all customers using AWS services. Disadvantages may include higher cost structure, extensive compliance timelines, and limited redundancy options and features compared with the geo-diversity of commercial data center regions.

AWS addresses the concerns driving physical separation requirements through the logical security capabilities we provide customers and the security controls we have in place to help protect customer data. The strength of that isolation combined with the automation and flexibility that it provides is on par with or better than the security controls seen in traditional, physically separated environments.

Logical Separation Compared to Physical Separation

Customers can leverage some or all aspects of the AWS capabilities below to meet or exceed the security of their on-premises physical separation requirements.

- **Unified authentication and authorization** – A robust and granular authentication and authorization model common across all AWS services that integrates with on-premises user identity management systems.
- **Rich monitoring and logging** – Deep and granular logging services for visibility of all API calls and resource state across AWS services. Current configuration and application events are logged in a centralized fashion to quickly understand both current security posture as well as a record of previous configuration states.
- **Virtual private cloud (VPC) and accompanying features** — VPC is a software-defined network that allows customers to create segmented or micro-segmented network domains to isolate traffic flow between different compute environments and AWS services as well as to join together segments when needed in safe and limited ways.
- **Encrypting data at-rest and in-transit** — Encryption options for all AWS storage services, powerful certificate creation and lifecycle management for encrypting data in transit. Key management via AWS Key Management Service (AWS KMS) or optionally using AWS CloudHSM for key generation and storage.
- **Host and instance isolation** — Options to provision dedicated hypervisor-enabled or bare-metal architectures to maintain customer data on a physical compute host is not shared with others.
- **Serverless and container architecture** — Isolated execution environments offer a smaller, ephemeral runtime environment to simplify security controls.

Unified Authentication and Authorization Mechanisms

The security mechanisms that define and manage identity and access management are among the most critical parts of an information security program. They serve to ensure that only authenticated principals (users, roles, groups, applications, and other identities) are authorized to access the targeted resource in the manner intended and with least privilege. A major feature that many organizations strive for is unified authentication across enterprise services. This feature allows for identity validation that is applicable to the entire portfolio of services. Executing on this functionality is difficult especially when dealing with diverse systems that require custom credential formats or have incompatible authorization models.

With AWS, customers gain the ability for unified authentication and authorization across all AWS services to enforce least privilege. AWS Identity and Access Management (IAM) allows customers to authenticate to any AWS service using the same credential format. IAM supports multiple means of authentication including API access keys,

console-based user passwords, and federation using external identity providers. Customers can configure the modes of authentication in IAM to require multifactor. AWS enables customers control to access to their resources in AWS services using policy-based authentication mechanisms. Each policy is populated either by the customer or by AWS, if using AWS managed policies, with definable elements that work together to construct granular and conditional “allow” or “deny” actions on particular resources. Customers can share or reuse policies across identities both within and between accounts, regardless of how those identities are authenticated. The robustness of these capabilities allows customers to architect a wide range of isolation and enforcement mechanisms of cloud resources and application level elements. This can be done using role-based access control (RBAC) methods or attribute-based access control (ABAC) methods or both.

Customers can use policies in multiple ways including 1) controlling which resources a set of users can access, 2) controlling which users can access a given resource, 3) controlling which AWS services can be used, and 4) controlling which users are allowed to modify policies. All policies allow the use of conditions to further scope access. For example, a customer could enforce a policy that only allows access to contents in an Amazon Simple Storage Service (Amazon S3) bucket if the user also has access to the decryption key managed in the AWS Key Management Service and the request is made over a specific VPC. Any ability to modify such a policy could be scoped down to a limited set of modifications that could only be made by a privileged set of administrators, all of whom must authenticate using multiple factors. Policies can be enforced across multiple accounts using AWS Organizations.

This level of control, deep integration, and wide interoperability would be exceedingly difficult to implement and manage in a traditional on-premises enterprise environment with physically separated and disparate systems. Most organizations use a combination of access and identity management solutions that vary across business unit and applications, but also across different layers of the infrastructure “stack” — network devices, virtualization, operating systems, and applications. This leads to a large set of identity services that need to be bound together and managed in a unified way. Adding to the management complexity, integration of these systems usually requires significant manual work coupled with continual care and attention as other parts of the service portfolio are brought into the fold. Additionally, uniform access policies still have to be crafted to ensure enforcement cascades down to the system and data levels across an enterprise.

With AWS, policy-based security management gives customers several distinct advantages. Security policies can be crafted to be both human and machine readable. This means that, while treating policy as code, it can also be a representative artifact for governance, risk and compliance efforts. This vastly improves clarity, accuracy, and transparency by letting stakeholders readily see actions that can and cannot be taken while being able to execute that policy directly in the service. Policies can be programmatically built and managed in a pipeline as code. This enables the same configuration management control over policy that an organization has with their application code. Another distinct benefit is the ability to utilize test automation in a pipeline, just like you would with software development, in order to verify and validate that policies function as expected. An example of this is IAM Access Analyzer which customers can be enable to continuously evaluate permissions granted in policies to identify resources that can be accessed from outside a customer's AWS account. IAM Access Analyzer uses automated reasoning, which applies logic and mathematical inference to evaluate hundreds or even thousands of policies across a customer's environment in seconds.

Rich Monitoring and Logging

A cornerstone of detecting and protecting one's environment and data is the ability to granularly monitor configurations across an enterprise and robust logging of activities occurring within an IT infrastructure. Visibility and traceability within IT environments is often hard to achieve for large on-premises operations that focus on physical separation based security controls. This design can result in fragmentation of operational views due to the lack of integration across services. This situation makes threat detection and root cause analysis challenging. AWS builds core security services that are highly integrated throughout AWS services, including monitoring and logging. AWS CloudTrail, Amazon CloudWatch, VPC Flow Logs, and AWS Config integrate across AWS services offerings, providing clear records of activities and configuration changes. The information provided by these services paints a multi-dimensional view of the operational state of the systems and data from functional, performance, and security perspectives. This comprehensive visibility can also be achieved at a lower cost compared to on-premises enterprise systems.

AWS CloudTrail provides the option to log AWS API requests for customers, regardless of whether the requests were made through the AWS Management Console, AWS SDKs, command line tools, or via other AWS services on the customer's behalf. Each

log event identifies the caller identity and called AWS API, the source IP address of the call, when the call occurred, and other parameters specific to the API. The logs can be ingested into a customer's local security information and event management (SIEM) system for analysis or sent to other AWS analytics services like CloudWatch Logs Insights. AWS CloudTrail logs are digitally signed to prevent tampering before they are stored in Amazon S3 for customers to access. Logs can also be retained using S3 Object Lock to create strong policies that makes all users, even root users, unable to delete the object. Logs are encrypted in storage, optionally under keys the customer controls in AWS KMS.

Amazon CloudWatch is used to monitor AWS resources and applications in near real-time. It can collect, track, and alarm based on metrics that are accessible via customizable dashboards or APIs. CloudWatch data are encrypted in transit and at rest. In addition, Amazon EventBridge delivers a near real-time stream of system events that describe changes to AWS resources to customers, which can set alarms and be notified of potentially unauthorized access. Rules can be implemented to match events and routed to one or more target functions or streams for further monitoring or even execution of corrective actions. For example, rules can examine incoming events, parse the incoming values, and properly route the event to any number of targets, such as email or mobile devices, ticketing queues, and issue management systems.

Configuration management is at the heart of controlling changes to an environment. Configurations that drift from their intended state present a risk to a system's security posture. Managing and enforcing configuration states across an on-premises environment is usually difficult because the tooling to measure a system's present state often lacks enough points of integration to offer a holistic view of the enterprise. In AWS, customers can address configuration management in multiple ways. One of the best options is to move toward an infrastructure as code (IaC) model for your environment. IaC allows you to provision, de-provision, and maintain infrastructure configuration state in consistent, repeatable, and automated manner using code. This includes being able to use secure code management practices and test automation directly on infrastructure components. One way to accomplish this with AWS is using AWS CloudFormation.

AWS CloudFormation templates can create, configure, and manage resources through use JavaScript Object Notation (JSON) or YAML. You manage these resources declared in the templates in units called AWS CloudFormation Stacks. Stacks can be composed as StackSets to manage resources across regions and accounts from single

templates or sets of templates. From a monitoring standpoint, CloudFormation is integrated with CloudTrail for recording actions performed by the service. Additionally, CloudFormation can detect configuration drift between the current resource configuration from a StackSets against the expected configuration declared in the StackSets. This level of configuration management can detect unmanaged changes and allow the user to reapply the template to return the resources to the declared state.

Often deeper and broader configuration management capabilities are needed by customers to handle the many ways AWS resources can be provisioned, changed, and managed. AWS Config fills this need by providing a detailed, continuous view of the configuration of AWS resources in a customer's AWS accounts. This includes how the resources are related to one another and how they were configured in the past so that a customer can see how the configurations and relationships change over time. AWS Config provides an AWS resource inventory, configuration history, and configuration change notifications across regions and accounts. These capabilities, along with advanced querying and customizable rules, enables security and governance insights and workflow automation for AWS resources.

Another linchpin for deep monitoring and logging is traffic flow visibility. VPC Flow Logs are a feature whereby a customer can capture information about the IP traffic going to and from network interfaces in their VPC. Flow log data can be published as records to Amazon CloudWatch Logs and Amazon S3 for further analysis. A flow log can be created for an entire VPC, a subnet, or a single network interface. In addition to Flow Logs, VPC also allows full packet capture when useful or necessary using its Traffic Mirroring feature. These two features work well together, VPC Flow Logs for routine network logging, and temporarily enabling Traffic Mirroring when circumstances require it.

Dealing with the volumes of logging data can be cumbersome for some customers so many choose to ease monitoring and analysis of logs by using Amazon GuardDuty the AWS managed threat detection offering. GuardDuty is a service that provides threat detection and continuous network security monitoring by consuming and analyzing many of the data sources mentioned here such as Flow Logs and CloudTrail logs, plus internal AWS DNS logs and threat intelligence feeds. GuardDuty applies machine learning, behavioral anomaly analysis, and other detection techniques to identify threats across network activity.

VPC and Accompanying Features

Amazon Virtual Private Cloud (Amazon VPC) enables the creation of a logically separate network enclave within the Amazon Elastic Cloud Compute (Amazon EC2) network that can house compute and storage resources. This environment can be connected to a customer's existing infrastructure through various means including a virtual private network (VPN) connection over the Internet, or through AWS Direct Connect, a service that provides private connectivity into the AWS Cloud. Use of a VPC provides organizations with flexibility, security, and complete control of their network presence in the cloud. The customer controls the private environment including IP addresses, subnets, network access control lists, security groups, operating system firewalls, route tables, VPNs, and internet gateways. Amazon VPC provides robust logical isolation of all customer resources, including their access paths to each other and with AWS services.

Every packet flow on the network is individually authorized against a rule to validate the correct source and destination before it is transmitted and delivered. It is highly improbable for information to arbitrarily pass between entities without specifically being authorized by both the transmitting and receiving entity. If a packet is being routed to a destination without a rule that matches it, the packet is dropped. Reply addresses must be valid or the packet is dropped. Moreover, while address resolution protocol (ARP) packets trigger an authenticated database look-up, ARP packets never hit the network as they are not needed for discovery of the virtual network topology. This means ARP spoofing is highly improbable on the AWS network. Also, promiscuous mode does not reveal any traffic other than traffic bound to and from the customer operating system. Customers can set precise rules for traffic ingress and egress which allow for increased connectivity flexibility, and enable more customer control over traffic segmentation and routing.

VPC connectivity options include the ability for the customer to:

- Connect to the Internet using Network Address Translation (private subnets) — Private subnets can be used for instances that should not have direct access to or from the Internet. Instances in a private subnet can access the Internet without exposing their private IP address by routing their traffic through a Network Address Translation (NAT) gateway in a public subnet.

- Connect securely to the corporate data center — All traffic to and from instances in a VPC can be routed to the customer's corporate data center over an industry standard, encrypted IPsec hardware VPN connection.
- Connect privately to other VPCs — Peer VPCs together to share resources across multiple virtual networks across multiple AWS accounts.
- Privately connect internal services across different accounts and VPCs within an AWS Organization, significantly simplifying internal network architecture.
- Use AWS Transit Gateway as a single, unified central gateway where connections can be created to many VPCs and on-premises systems while being able to manage authentication and access to the services with AWS IAM.
- Use VPC features like AWS PrivateLink to create private connections to resources outside of the customer's VPC. These private connections do not traverse the public Internet and can provide secure connectivity between VPCs, AWS services, and on-premises applications.

Additionally, all traffic within a VPC and inter-region peering is transparently encrypted when using supported instance types. From an infrastructure standpoint, physical network encryption is used by AWS to encrypt network traffic on any link outside of AWS physical control such as between data-centers.

Encrypting Data-at-Rest and -in-Transit

AWS recommends encryption as an additional access control to complement the identity, resource, and network-oriented access controls already described. AWS provides a number of features that enable customers to easily encrypt data and manage the keys. All AWS services offer the ability to encrypt data at rest and in transit. AWS KMS integrates with the majority of services to let customers control the lifecycle of and permissions on the keys used to encrypt data on the customer's behalf. Customers can enforce and manage encryption across services integrated with AWS KMS through the use of policy and configuration tools.

AWS services' use of server-side encryption is the easiest way for a customer to ensure encryption is implemented correctly and applied consistently. Customers can control when data is decrypted, by whom, and under which conditions as it passed to and from their applications and AWS services. Because access to encrypt or decrypt the data within the service is independently controlled by AWS KMS policies under the customer's control, customers can isolate control over access to the data, from access to the keys. This isolation model is a powerful additional logical separation control that can be applied across a customer's AWS environment.

In addition to controlling how server-side encryption happens within AWS services, customers can choose to encrypt data within their own application environment using AWS KMS with client-side encryption, thereby taking AWS services out of their trust boundary. Application-level, client-side encryption can be used to ensure a consistent security posture as data traverses within a customer's own service architecture, whether in AWS, on-premises, or in a hybrid model. The use of AWS KMS to manage the lifecycle of and permissions on keys provides a consistent access control mechanism for all encryption keys, regardless of where they are used.

In order to prevent unauthorized use of encryption keys outside the boundary of AWS KMS, the service utilizes hardware security modules (HSMs) to protect customer key material while in use. These HSMs are validated under Federal Information Processing Standard (FIPS) 140-2 with physical tamper response controls. The HSMs are designed so that plaintext keys cannot be used outside the HSM by anyone, including AWS employees. The only way keys can be used is when an authenticated and authorized customer request is received by the service. In response to the request, AWS KMS enables the customer's key to be used within the HSM for an encryption or decryption operation. Customer keys can only be used within the AWS region in which they were created. The HSMs in AWS KMS are designed as multi-tenant in the sense that any customer's key could be used in any HSM within the region. Like other AWS services that utilize multi-tenancy, AWS KMS is designed to isolate usage of keys only to the customer that owns the keys. There is no mechanism for an unauthorized user to cause a customer's key to be used. AWS KMS transparently manages the durability and availability of customer keys and can scale to support any number of keys at the rate customers' applications need to use them. Customers simply manage the lifecycle and permissions on keys using the same authentication and authorization controls available to every other AWS service. Every request made of AWS KMS is logged to AWS CloudTrail to provide an audit of when keys were used and under what circumstances. AWS KMS is in scope for all accreditation programs supported by AWS that relate to data protection.

For customers with requirements to directly manage the HSM device that generates, stores, and uses their encryption keys, AWS CloudHSM is available an as option. AWS CloudHSM offers a dedicated FIPS 140-2 Level 3 validated HSM and affords the flexibility of integrating with customer applications using industry-standard APIs such as PKCS#11, Java Cryptography Extensions (JCE), and Microsoft CryptoNG (CNG) libraries. It enables organizations to export keys to most other commercially available

HSMs for use in hybrid architectures. AWS automates the time-consuming administrative tasks around these HSMs such as hardware provisioning, software patching, network routing, and creating encrypted backups of key stores. Customers are responsible for scaling their CloudHSM environment and managing the crypto user accounts and credentials within the HSM. Like AWS KMS, CloudHSM is designed so that plaintext keys cannot be used outside the HSM by anyone, including AWS employees.

Customers can combine the ease-of-use and integration with AWS services offered by AWS KMS with AWS CloudHSM by using the AWS KMS custom key store option. Customers logically attach an AWS CloudHSM cluster to an AWS KMS key identifier so that requests made to the key are authorized by AWS KMS, but executed on the customer's dedicated CloudHSM.

To protect data in transit, AWS encourages customers to leverage a multi-level approach. All network traffic between AWS data centers is transparently encrypted at the physical layer. All traffic within a VPC and between peered VPCs across regions is transparently encrypted at the network layer when using supported Amazon EC2 instance types. At the application layer, customers have a choice about whether and how to use encryption using a protocol like Transport Layer Security (TLS). All AWS service endpoints support TLS to create a secure HTTPS connection to make API requests.[1] For customer-managed infrastructure within AWS that needs to terminate TLS, AWS offers several options including load balancing services (e.g., Elastic Load Balancing, Network Load Balancer, and Application Load Balancer), Amazon CloudFront (a content delivery network), and Amazon API Gateway. In order to implement a TLS connection, each of these endpoint services allows customers to upload their own digital certificates to bind a cryptographic identity to the endpoint. Digital certificates are notoriously difficult to manage at scale because they expire and need to be rotated. AWS simplifies the process of generating, distributing, and rotating digital certificates with AWS Certificate Manager (ACM). ACM offers publicly trusted certificates at no cost that can be used in AWS services that require them to terminate TLS connections to the Internet. ACM also offers the ability to create a private certificate authority to automatically generate, distribute and rotate certificates to secure internal communication among customer-managed infrastructure.

Using services like AWS KMS, AWS CloudHSM, and AWS ACM, customers can implement a comprehensive data at rest and data in transit encryption strategy across

their AWS ecosystem to ensure all data of a given classification shares the same security posture.

Host and Instance Features

AWS is constantly evolving its security capabilities at both the host and instance level of operations. These features provide isolation and separation of operations for host hardware and the instances running on those hosts. With the introduction of AWS Nitro System, AWS provides industry defining security mechanisms for firmware and hypervisor operations. AWS Nitro System is comprised of a family of Peripheral component Interconnect Express (PCIe) cards with custom integrated circuits (ASICs) that control distinct functions such as access to storage, virtual networking, and a Nitro Security Chip that continuously monitors and protects hardware resources and independently verifies firmware each time a system boots. These, in conjunction with the Nitro hypervisor, a lightweight kernel virtual machine (KVM)-based hypervisor, provide the backbone for many AWS instance families. This allows AWS to constrain operator-host interactions to a small set of functions that can only be called through an API. There is no interactive shell access. Virtual instances operating on these hosts also have numerous additional security mechanisms enforced, such as memory and CPU isolation.

In addition to providing highly secure, logically isolated, multi-tenant compute services, AWS also provides means of deploying compute to dedicated hardware using Dedicated Instances, Dedicated Hosts, and Bare Metal. These deployment options can be used to launch Amazon EC2 instances onto physical servers that are dedicated for customer use. Dedicated Instances are hypervised Amazon EC2 instances that run in a VPC on hardware that's dedicated to a single customer. Dedicated Instances are physically isolated at the host hardware level from instances that belong to other AWS accounts. Dedicated Instances may share hardware with other instances from the same AWS account that are not Dedicated Instances. A Dedicated Host is also a physical server that's dedicated for customer use. With a Dedicated Host, customers have visibility and control over how hypervised instances are placed on the server. Bare Metal instances are non-hypervised host hardware devices. Using the AWS Nitro technology for network and storage offload, as well as the Nitro Security Chip to address the risks associated with serial single-tenancy on Bare Metal, customers have direct access to Amazon EC2 hardware. These Bare Metal instances are full-fledged

members of the Amazon EC2 service and have access to services such as Amazon VPC and Amazon Elastic Block Store (Amazon EBS).

There are little to no performance, security, or physical differences between Dedicated Instances and instances deployed on Dedicated Hosts. However, Dedicated Hosts give customers additional control over how instances are placed on a physical server and how that server is utilized. When customers use Dedicated Hosts, they have control over instance placement on the host using the Host Affinity and Instance Auto-placement settings. If customers want to use AWS, and have an existing software license that requires that the software be run on a particular piece of hardware for some minimum amount of time, Dedicated Hosts allow visibility into the host's hardware, enabling customers to meet licensing requirements.

Serverless and Containers

The ability to seamlessly incorporate serverless technology, container technology, and microservice designs in AWS enables customers to build multiple levels of isolation for workloads. AWS services use multiple layers of security to achieve isolated operations. Many of the security features of services like AWS Lambda and AWS Fargate, while operating behind the scenes, are based on the functionality provided by capabilities of AWS services and features already discussed in this paper. For example, the set of security services and capabilities included with the EC2 Nitro architecture, VPC networking, and IAM, (e.g., ACLs, Security Groups, and IAM Policies) apply here as well.

AWS approaches logical isolation with its serverless service, AWS Lambda, and its managed container service, AWS Fargate, in a multilayered fashion. These layers start with bare metal instances, the same ones that any customer can provision, using the same underlying Nitro architecture and its security benefits that were previously discussed. Then, at a subsequent layer, there is the purpose-built lightweight virtual machine monitor called Firecracker which was created by AWS to securely manage containers and serverless functions. Firecracker functions as an isolated environment that provides secure runtime execution for serverless functions and containers. Lambda operates in EC2 as micro virtual machines (micro-VMs) and offers similar protections for logical isolation as other EC2 instances. Each function executes in a sandbox that is contained in the micro-VM. The sandbox offers secure Linux kernel isolation using cgroups, namespaces, seccomp, and other features. Additionally, techniques such as

process jailing and static linking are used to securely isolate runtime. Firecracker presents multiple security features such as a simple guest model — in other words, a virtualized device model that presents a minimal surface area allowing just enough features for operation. These concentric levels of protection allow for rapid, fraction-of-a-second invocations while securely isolating the micro-VM to a customer account. The source code for Firecracker has been provided as open source to the community at large to support full transparency with its operational configuration and capabilities.

Customers can build their own logical isolation and separation practices tailored to their organization using capabilities such as serverless resources. For example, customers can build event driven architectures which have multiple automation focused use cases from incident response to fleet management. Lambda in combination with other AWS services, such as Amazon CloudWatch Events or Amazon EventBridge, AWS Step Functions, Amazon GuardDuty, and others to create new security capabilities. With these services, operations can be designed to auto-remediate security issues without the need for human intervention. For example, a finding in Amazon GuardDuty can be sent to CloudWatch Events which can then trigger a Lambda function to initiate a remediation activity, such as updating security groups, web application firewall, or changing IAM policies. AWS Step Functions and additional Lambda functions can be add to the workflow for more complex logic such as calling AWS Systems Manager to execute commands on an EC2 instance to capture or modify configurations. This concept can be used to build similar isolation practices that keep direct human access away from important workloads — something that would be highly difficult in a traditional on-premises environment. For more information, see the AWS Security Incident Response Guide.

AWS container orchestration service, Amazon Elastic Container Service (Amazon ECS), provides its own security separation and isolation properties whether you are using it to manage container services such as AWS Fargate or in a self-managed environment on EC2. Amazon ECS Task Definitions allows customers to define security functionality and isolation parameters using the security features of their own VPC. One or more containers can operate within prescribed constraints using an Amazon ECS Task Definition. A customer can define granular container communication rules because each task definition can receive its own elastic network interface in a customer VPC. This gives containers the same VPC network security features as seen in EC2 instances. Customers can apply IAM policies dto each task furthering access and operational bounds for each container or sets of containers. Security and isolation

mechanisms related to upstream functions such as a container registry are addressed with Amazon Elastic Container Registry (Amazon ECR). When a container is built or pulled, it is critical that protections are around those source images both as they are being housed and transmitted. Amazon ECR automatically encrypts container images at rest and in transit. Using IAM policies, access to images in Amazon ECR can be constrained to only the principals that have a need for that access. When used together, the suite of AWS container services creates an end-to-end isolated and secure environment for fleets of containers or microservices.

AWS's cloud services offer customers with a growing list of capabilities to make security "in the cloud" robust and easy to implement while maintaining a high security bar. Ever expanding security services and features minimize cumbersome processes, improve confidentiality, and expand accessibility to democratize security and the benefits of modern techniques and innovation. Applying foundational security practices, such as encryption, with proper customer implementation can effectively address the security risks associated with the demand for physical separation.

Mitigating Unauthorized Access to Data

Preventing unauthorized access requires practicing proper security hygiene and implementing robust preventive and detective capabilities. For example, systems should be designed to limit the "scope of impact" of security events so that one node with unauthorized access has minimal impact on any other node in the enterprise. Hyperscale CSPs, such as AWS, provide a full security tooling environment to enable customers to maintain encrypted communications and implement tampering protections to mitigate the risk of unauthorized access. AWS does not have visibility into, or knowledge of, the content or data inside a customer account, including whether or not that content includes any personal information. AWS customers are empowered to use various techniques such as encryption, tokenization, data decomposition, and cyber deception to render content unintelligible to AWS or other parties seeking access to its content.

- **Encryption** — Appropriately encrypting data can make the data unreadable. This means storing encrypted data in the cloud, regardless of location, can provide adequate protection against the vast majority of exfiltration attempts. It is crucial that the encryption keys for the data are carefully managed to ensure strong protections are maintained against any intercepting party. AWS provides services that can deliver these capabilities at an enterprise level with AWS CloudHSM or AWS KMS.[8] The amount of control that customers wish to have over the encryption method, storage of cryptographic keys, and management of cryptographic keys used with their data is up to the customer.
- **Tokenization** – Tokenization is a process that allows you to define a sequence of data to represent an otherwise sensitive piece of information (e.g., a token to represent a customer's credit card number). A token is meaningless on its own and cannot be mapped back to the data it represents without use of the tokenization system. Token vaults can be constructed in VPCs to store sensitive information in an encrypted form while sharing tokens out to approved services for transmitting obfuscated data. In addition, AWS has a number of partners that specialize in providing tokenization services that integrate with popular databases and other storage services.
- **Data Decomposition** – This is a process that reduces data sets into unrecognizable elements that have no significance on their own.[10] These elements or fragments are then stored in a distributed fashion so that any unauthorized access to one node would yield only an insignificant data fragment. A particular advantage of this technique is it requires an unauthorized user to access all nodes, obtain all fragments, and know the algorithm (or fragmentation scheme) to piece together the data in a coherent way.
- **Cyber Deception Defense** – Cyber deception architectures and solutions can be a key component for mitigating advanced security events. Deception solutions can use highly sophisticated traps and decoys to present an unauthorized party with the perception that they have infiltrated the system while in reality diverting them to a highly controlled environment. Intelligence about the unauthorized party is gathered in order to mitigate future attempts and the issue is neutralized.

AWS also monitors for unauthorized remote management and expeditiously disconnects or disables unauthorized remote access once it is detected. All remote administrative access attempts are logged, and the logs are reviewed, not just by

humans for suspicious activity, but also by automated machine-learning systems built by the AWS Security team to detect unusual access patterns that may indicate unauthorized attempts to access data. If suspicious activity is detected, the incident response procedures are initiated. Further, AWS has established formal policies and procedures to delineate standards for logical access to the AWS infrastructure and hosts. The policies also identify functional responsibilities for the administration of logical access and security. Unless prohibited by law, AWS requires that all employees undergo a background investigation commensurate with their position and level of access. Finally, customer virtual instances are solely controlled by the customer who has full root access or administrative control over accounts, services, and applications. AWS personnel do not have the ability to log into customer EC2 instances or ECS/EKS containers.

Duties and areas of responsibility (for example, access request and approval, change management request and approval) must be segregated across different individuals to reduce opportunities for an unauthorized or unintentional modification or misuse of AWS systems. AWS personnel with a business need to access the management plane are required to first use multi-factor authentication, distinct from their normal corporate Amazon credentials, to gain access to purpose-built administrative hosts. These administrative hosts are systems that are specifically designed, built, configured, and hardened to protect the management plane. All 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 are revoked. AWS has implemented a session lock out policy that is systematically enforced. The session lock is retained until established identification and authentication procedures are performed.

AWS enables organizations to retain audit records that support after-the-fact investigations of security events and the ability to meet regulatory and organizational information retention requirements. Customers can retrieve cloud audit logs and reports by leveraging CloudTrail and CloudWatch Logs, which they can then provide to the appropriate authorities. These solutions enable AWS customers to respond directly to law enforcement requests for information, enabling government officials to get the information that they require without accessing underlying customer content. For additional information on “compelled disclosure” or law enforcement access to data, see the AWS Data Residency whitepaper.

Case Study

US Defense Department accepts logical storage separation approach for sensitive unclassified workloads

In December 2011, the U.S. Federal Chief Information Officer established a government-wide policy mandating federal agencies use the Federal Risk and Authorization Management Program (FedRAMP) — a standardized, federal-wide program for the security authorization of cloud services. FedRAMP maintains three standardized security baselines — Low, Moderate, and High impact — based on Federal Information Processing Standards Publication (FIPS) 199 categorizations. These baselines were developed through the collaboration of cybersecurity experts across private industry and the U.S. Government (including the Department of Defense (DoD)). While the DoD established reciprocity with the FedRAMP Moderate baseline, it has not established reciprocity with the FedRAMP High baseline. Instead, the DoD developed and implemented what is effectively a "FedRAMP plus" set of security controls and requirements via the DoD Cloud Computing Security Requirements Guide (SRG).

In particular, DoD through the SRG requires separation between DoD and Federal government tenants/missions either via physical or logical means. More specifically, the SRG states that "CSPs must provide evidence of strong virtual separation controls and monitoring, and the ability to meet 'search and seizure' requests without the release of DoD information and data." Even further, for Impact Level 5 systems (IL5),[2] DoD requires "physical separation (e.g., dedicated infrastructure) from non-DoD/non-Federal Government tenants." These DoD requirements are intended to address DoD concerns regarding the co-mingling of DoD data with other tenant data from unintended data disclosure and the unauthorized access or tampering of DoD data by a non-DoD tenant.

To implement an outcome-focused best practice, the SRG acknowledged the use of logical separation as a viable approach to meet DoD IL5 separation requirements:

"A CSP may offer alternate solutions that provide equivalent security to the stated requirements. Approval will be assessed on a case by case basis during the PA [provisional authorization] assessment process."

> Through DoD's cloud computing SRG assessment and authorization (i.e., accreditation) process, AWS demonstrated the sufficiency of logical separation *combined with dedicated tenancy*[3] to meet the intent behind a requirement for dedicated, physically isolated infrastructure for DoD's most sensitive unclassified workloads**.** Our accepted approach confirms that multi-tenant logically separated environments that meet robust security controls can provide a level of security superior to dedicated private cloud deployments, while providing significant advantages in availability, scalability, and lower cost. Modern cloud technology from established providers can offer novel solutions that can meet the objective of traditional technology security as long as accreditation approaches are flexible enough to accommodate alternative implementations.

Conclusion

The AWS approach shows that properly configured, multi-tenant, logically separated environments can provide a level of security superior to dedicated private cloud deployments, while providing significant advantages in availability, scalability, and lower cost. Modern cloud technology from leading providers offers novel solutions that can meet the objective of traditional security based on physical isolation as long as accreditation approaches are flexible enough to accommodate alternative implementations.

Although reviewing security controls can be valuable for demonstrating compliance, experience has shown that organizations that focus primarily (and in some instances exclusively) on traditional controls implementation can inadvertently limit their access to best-in-class security solutions. As public and private sector organizations evaluate whether CSPs meet requirements based on legacy concepts and on-premises architectures, they should step back and first clearly articulate desired security outcomes. Mapping those outcomes to CSP capabilities and understanding how to properly address those needs leads to a deeper understanding of how to most efficiently design a solution as well as clarifies the risk that needs to be accepted while operating in the cloud.

As security assurance programs mature and scale to keep up with the rapid pace of cloud feature and security innovation, traditional control implementation details will become increasingly irrelevant relative to the capabilities CSPs have in place today and

will likely enhance very quickly. The desired end-state – robust cloud security, based on a framework defined by customer security outcomes and CSP-determined security capabilities to meet those outcomes – can only come about as a result of continuous dialogue across the cloud assurance stakeholder community. AWS believes this approach will continue to provide significant improvements in maintaining assurance of a CSP's security posture.

Contributors

Contributors to this document include:

- Tim Anderson, Senior Security Advisor, Growth Strategies, Security
- Ken Beer, General Manager, Cryptography
- Min Hyun, Global Lead, Growth Strategies, Security
- Mark Ryland, Director, Office of the CISO, Security

Further Reading

For additional information, see:

- AWS Whitepapers page
- AWS Data Residency whitepaper
- AWS Security Incident Response Guide

Document Revisions

Date	Description
July 2020	First publication

Notes

1 AWS is updating all AWS FIPS endpoints to a minimum Transport Layer Security (TLS) version of 1.2 across all AWS Regions, with a targeted completion date of March 31, 2021. Once completed, these updates will revoke the ability to use TLS 1.0 and TLS 1.1 on all FIPS endpoints. No other AWS endpoints will be affected by this change.

2 5.2.2.2 Impact Level 5 Location and Separation Requirements

Information that must be processed and stored at Impact Level 5 can only be processed in a dedicated infrastructure, on-premises or off-premises in any cloud deployment model that restricts the physical location of the information as described in section 5.2.1, “Jurisdiction/ Location Requirements.” This excludes public service offerings.

The following applies:

- Only DoD private, DoD community or Federal Government community clouds are eligible for Impact Level 5.
- Each deployment model may support multiple missions or tenants / missions from each customer organization.
- Virtual/logical separation between DoD and Federal Government tenants / missions is permitted.
- Virtual/logical separation between tenant/mission systems is minimally required.
- Physical separation (e.g. Dedicated Infrastructure) from non-DoD/non-Federal Government tenants is required.

NOTE: A CSP may offer alternate solutions that provide equivalent security to the stated requirements. Approval will be assessed on a case by case basis during the PA assessment process.

https://iasecontent.disa.mil/cloud/Downloads/Cloud_Computing_SRG_v1r3.pdf

3 Refer to the previous section on "Host and Instance Features".