

DoD-Compliant Implementations in AWS

First Published April 2015

Updated November 3, 2021



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.

© 2021 Amazon Web Services, Inc. or its affiliates. All rights reserved.

Contents

Overview	1
Getting started	1
Shared responsibilities and governance.....	2
Shared responsibility model.....	2
Compliance and governance	13
AWS global infrastructure.....	17
Architecture.....	19
Traditional DoD data center	19
DoD compliant cloud environment.....	20
AWS services	26
Compute	26
Networking.....	30
Storage	35
Management.....	40
Services in scope	44
Reference architecture	45
Impact level 2	45
Impact level 4	49
Impact level 5	51
Conclusion	53
Contributors	54
Further reading	54
Document revisions	54

Abstract

This whitepaper is intended for Department of Defense (DoD) mission owners who are designing the security infrastructure and configuration for applications running in Amazon Web Services (AWS). It provides security best practices and architectural recommendations that can help you properly design and deploy DoD-compliant infrastructure to host your mission applications and protect your data and assets in the AWS Cloud. The paper is designed for Information Technology (IT) decision makers and security personnel and assumes that mission owners are familiar with basic security concepts in the areas of networking, operating systems, data encryption, and operational controls.

AWS provides a secure hosting environment for mission owners in which to deploy their applications. Mission owners retain the responsibility to securely deploy, manage and monitor their systems and applications in accordance with DoD security and compliance policies. When operating an application or system on AWS, the mission owner is responsible for network configuration and security of their AWS environment, including [Amazon Elastic Compute Cloud](#) (Amazon EC2) guest operating systems and management of user access.

Overview

In January 2015, the Defense Information Systems Agency (DISA) released the DoD Cloud Computing (CC) Security Requirements Guide (SRG), which provided guidance for cloud service providers and for DoD mission owners in support of running workloads in cloud environments. The DoD CC SRG is the primary guidance for cloud computing in the DoD community.

This whitepaper provides high-level guidance for DoD mission owners and partners in designing and deploying solutions in the AWS Cloud that are able to be accredited at Impact Level (IL) 2, IL 4, and IL 5. Although there are many design permutations that can meet CC SRG requirements on AWS, this document presents sample reference architectures to consider that will address many of the common use cases for IL2, IL4, and IL5.

Getting started

When considering an application deployment or migration to the AWS Cloud, DoD mission owners must first make sure that their IT plans align with their organization's business model. A solid understanding of the mission and core competencies of your organization will help you identify opportunities for modernization and innovation by migrating to the AWS Cloud.

You must think through key technology questions, including:

- How can the AWS Cloud advance your mission objectives?
- Do you have legacy applications and systems that need greater scalability, reliability, or security than you can afford to maintain in your own environment?
- What are your compute, storage, and network capacity requirements?
- How will you be prepared to scale up (and down) to support the mission?

As you answer each question, apply the lenses of *flexibility*, *cost effectiveness*, *scalability*, *elasticity*, and *security*. Taking advantage of AWS services allows you to focus on your core competencies and leverage the resources and experience that AWS provides.

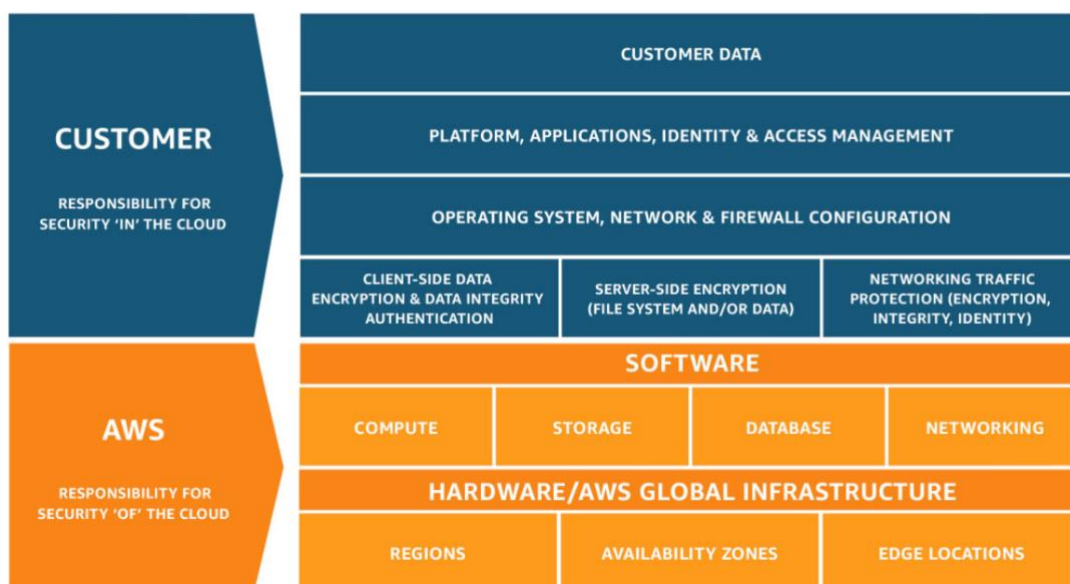
Shared responsibilities and governance

As mission owners build systems on top of AWS Cloud infrastructure, the responsibility for implementing operational, maintenance and security measures are shared: mission owners provide operational, maintenance, and security support for their software-defined cloud components, and AWS provides operational, maintenance, and security for its infrastructure. Mission owners can also inherit or use security controls provided by AWS.

Shared responsibility model

Security and compliance are shared responsibilities between AWS and mission owners. This shared model can help relieve your operational burden because AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the service operates.

The mission owner assumes responsibility and management of the guest operating system (including updates and security patches) and other associated application software, as well as the configuration of the AWS-provided security group firewall. Mission owners should carefully consider the services they choose as their responsibilities vary depending on the services used, the integration of those services into their IT environment, and applicable laws and regulations.¹



Security responsibilities in the Cloud and of the Cloud

It is possible for mission owners to enhance security and/or meet their more stringent compliance requirements by leveraging AWS services like [Amazon GuardDuty](#), [AWS Key Management Service](#) (AWS KMS), and encrypted [Amazon Simple Storage Service](#) (Amazon S3) buckets, as well as network firewalls and centralized log aggregation. The nature of this shared responsibility also provides the flexibility and mission owner control that permits the deployment of solutions that meet industry-specific certification requirements.

This mission owner and AWS shared responsibility model also extends to compliance controls. Just as the responsibility to operate the IT environment is shared between AWS and its mission owners, so is the management, operation, maintenance, and verification of shared compliance controls. AWS manages security controls associated with AWS physical infrastructure. Mission owners can then use the AWS control and compliance documentation available to them at [AWS Artifact](#) to perform their control evaluation and verification procedures.

AWS offers services and features that can ease management of the customer's portion of the shared responsibility model. Refer to [AWS Cloud Security](#).

Mission owner responsibilities

Service instance management

Mission owners are responsible for managing their instantiations of Amazon S3 bucket storage and objects, [Amazon Relational Database Service](#) (Amazon RDS) database instances, EC2 compute instances and their associated storage, and Virtual Private Cloud (VPC) network environments. This includes mission owner-installed operating systems, databases, and applications running on EC2 instances that are within their authorization boundary.

Mission owners are also responsible for managing specific controls relating to shared interfaces and services within their security authorization boundary, such as customized security control solutions. Examples include, but are not limited to, configuration and patch management, vulnerability scanning, disaster recovery, protecting data in transit and at rest, host firewall management, credential management, identity and access management, and VPC network configurations.

Mission owners provision and configure their AWS compute, storage, and network resources using API calls to AWS API endpoints or by using the [AWS Management Console](#). Using these methods, the mission owner is able to launch and shut down EC2

and RDS instances, change firewall parameters, and perform other management functions.

Application management

Applications that run on AWS services are the responsibility of each mission owner to configure and maintain. Mission owners should address the controls relevant to each application in the applicable System Security Plan (SSP).

Operating system maintenance

AWS provides [Amazon Machine Images](#) (AMIs) for standard OS releases that include Amazon Linux 2, Microsoft Windows Server, Red Hat Enterprise Linux, SUSE Linux, and Ubuntu Linux, with no additional configuration applied to the image. An AMI provides the information required to launch an EC2 instance, which is a virtual server in the cloud. The mission owner specifies the AMI used to launch an instance and the mission owner can launch as many instances from the AMI as needed. An AMI includes the following:

- A template for the root volume for the instance. The root volume of an instance is either an [Amazon Elastic Block Store](#) (Amazon EBS) volume or an instance store volume.
- Launch permissions that control which AWS accounts can use the AMI to launch instances.
- A block device mapping that specifies the volumes to attach to the instance when it's launched.

The OS that is installed on an AMI provided by AWS is patched to a point in time. In general, AMIs include a minimal install of a guest operating system. AWS does not perform any systems administration, operations, or maintenance duties such as patching. DoD mission owners are responsible for properly hardening, patching, and maintaining their AMIs in accordance with DoD Security Technical Implementation Guides (STIGs) and the Information Assurance Vulnerability Management process.

To aid mission owners in compliance and configuration management, consider implementing AWS Systems Manager. AWS Systems Manager can scan your instances against your patch, configuration, and custom policies. You can define patch baselines, maintain up-to-date antivirus definitions, and enforce firewall policies. You can also remotely manage your servers at scale without manually logging in to each server. Systems Manager also provides a centralized store to manage your

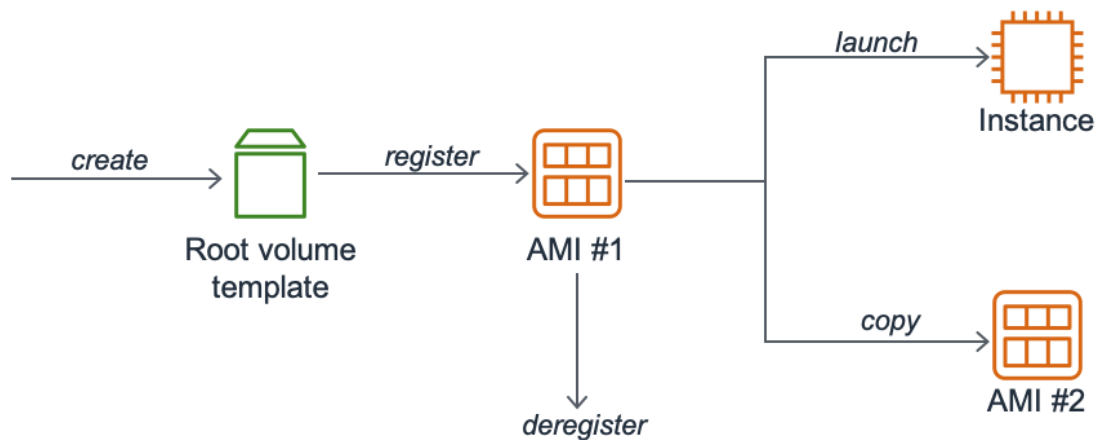
configuration data, whether in plaintext, such as database strings, or secrets, such as passwords. This allows you to separate your secrets and configuration data from code.

Amazon EC2 provides an [AWS Systems Manager](#) (SSM) document, `AWSEC2-ConfigureSTIG`, to apply Security Technical Information Guide (STIG) controls to an instance to help you quickly build compliant images following STIG standards. The STIG SSM document scans for misconfigurations and runs a remediation script. The STIG SSM document installs InstallRoot on Windows AMIs, which is a utility produced by the Department of Defense (DoD), designed to install and update DoD certificates and remove unnecessary certificates to maintain STIG compliance. There are no additional charges for using the STIG SSM document. For more information, refer to [AWSEC2-ConfigureSTIG](#).

In 2019, AWS released new AMIs for Microsoft Windows Server to help you meet STIG compliance standards. Amazon EC2 Windows Server AMIs for STIG Compliance are pre-configured with more than 160 required security settings. STIG-compliant operating systems include Windows Server 2012 R2, Windows Server 2016, and Windows Server 2019. The STIG-compliant AMIs include updated DoD certificates to help you get started and achieve STIG compliance. For instructions on how to deploy these AMIs, consult [Amazon EC2 documentation](#) or search on the [AWS Marketplace](#).

AWS does not guarantee a specific patch level or control configuration settings. Mission owner responsibility includes updating any EC2 instance to a recent patch level and configuring the instance to suit specific mission needs. Upon deployment of EC2 instances, the mission owner can assume full administrator access and is responsible for performing additional configuration, patching, security hardening, vulnerability scanning, and application installation. AWS does not maintain administrator access to mission owner EC2 instances.

Mission owners can customize the instance launched from a public AMI and then save that configuration as a custom AMI for the mission owner's own use. After mission owners create and register an AMI, they can use it to launch new instances. This concept is analogous to creating virtual machine templates in a traditional data center environment. Instances launched from this customized AMI contain all of the customizations that mission owner has made. The mission owner can deregister the AMI when finished. After the AMI is deregistered, mission owners cannot use it to launch new instances.



Creating custom Amazon Machine Images (AMIs)

Workload migration

Mission owners also have several options to assist in bulk virtual machine migration to AWS, commonly referred to as lift-and-shift.

One such option is the [AWS Server Migration Service](#) (SMS). AWS SMS is an agentless service which makes it easier and faster for mission owners to migrate thousands of on-premises workloads to AWS. AWS SMS lets mission owners automate, schedule and track incremental replications of live server volumes, making it easier to coordinate large-scale server migrations. Although agentless, SMS does require privileged access to the source servers' hypervisor.

A second option is AWS Application Migration Service, formerly known as CloudEndure Migration, which is an agent-based approach. AWS Application Migration Service simplifies, expedites, and reduces the cost of cloud migration by offering a highly automated lift-and-shift solution. With AWS Application Migration Service, you can maintain normal business operations throughout the replication process. It nearly continuously replicates source servers, which means little to no performance impact. When you're ready to launch the production machines, your machines are automatically converted from their source infrastructure into the AWS infrastructure so they can boot and run natively in AWS.

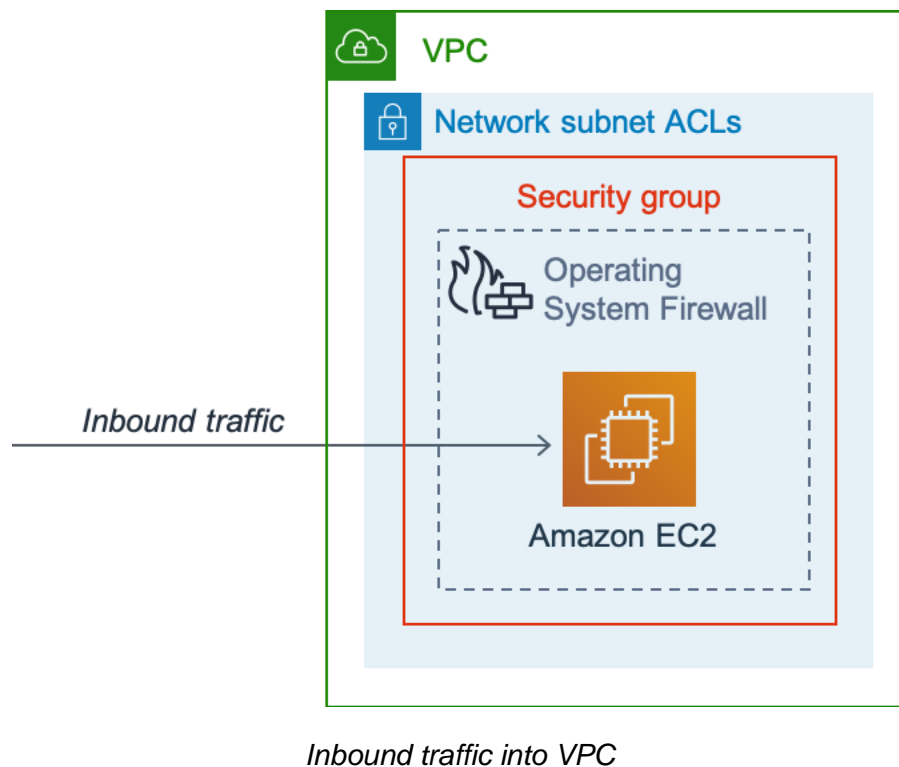
Security group configuration

Mission owners are responsible for properly configuring their security groups in accordance with their organization's networking policies. A security group acts as a virtual firewall for an instance to control inbound and outbound traffic. As part of ongoing operations and maintenance, mission owners must regularly review their security group

configuration and instance assignment to maintain a secure baseline. Security groups are not a solution that can be deployed using a one-size-fits-all approach. They should be carefully tailored to the intended functionality of each class of instance deployed within the mission owner's AWS environment.

VPC configuration

[Amazon Virtual Private Cloud](#) (VPC) provides enhanced capabilities that AWS mission owners can use to secure their AWS environment through the deployment of traditional networking constructs, such as demilitarized zones (DMZs), Virtual Local Area Networks (VLANs), and subnets that are segregated by functionality. Network Access Control Lists (NACLs) provide stateless filtering that can be used similarly to a firewall to defend against malicious traffic at the subnet level. This adds another layer of network security in addition to the mission owner's security group implementation.



Backups

Mission owners are responsible for establishing a backup strategy using AWS services or third-party tools that meet the retention goals identified for their application. Through the use of Amazon EBS snapshots, mission owners can ensure their data is backed up to Amazon S3 on a regular basis. Mission owners are responsible for setting and maintaining proper access permissions to their Amazon EBS volumes and Amazon S3

buckets and objects. Amazon S3 objects and Amazon EBS snapshots can also be configured with lifecycle policies to meet retention requirements, and can be aged off to Amazon S3 Glacier for lower-cost long-term deep storage.

Host-based security tools

Mission owners should install and manage anti-malware and host-based intrusion detection systems in accordance with their organization's security policies. Host-based security tools can be included within the mission owner's AML, installed via bootstrapping services when the instance is launched, or deployed using configuration management and automation tools like AWS Systems Manager.

Vulnerability scanning and penetration testing

Mission owners are responsible for conducting regular vulnerability scanning and penetration testing of their systems in accordance with their organization's security policies. All vulnerability and penetration testing must be properly coordinated with AWS Security, in accordance with AWS policy. For more information, refer to [AWS Penetration Testing page](#).

Vulnerability scanning of EC2 instances can be accomplished using third-party tools, or via [Amazon Inspector](#). Amazon Inspector is an automated security assessment service that assesses applications for exposure, vulnerabilities, and deviations from best practices. After performing an assessment, Amazon Inspector produces a detailed list of security findings prioritized by level of severity. These findings can be reviewed directly or as part of detailed assessment reports which are available via the Amazon Inspector console or API.

High availability and disaster recovery

Mission owners also have the responsibility to architect their applications and systems so they are highly available and are routinely backed up. Applications and systems should use multiple [Availability Zones within an AWS Region](#) for fault tolerance. 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. Mission owners also have the option of automating recovery in case of failures of systems or processes.

With APIs and automation in place, mission owners can launch and test the Disaster Recovery (DR) solution on a recurring, periodic basis to endure proper functionality of the solution, and be prepared ahead of time. Mission owners can reduce recovery times by quickly provisioning pre-configured resources (such as AMLs) when they are needed

or cutover to already provisioned DR site (and then scaling gradually as you need). Security best practices can be enumerated within an [AWS CloudFormation](#) template and provision resources within a VPC.²

AWS Identity and Access Management

Mission owners are responsible for properly managing their AWS accounts, including AWS account credentials, as well as any IAM users, groups or roles that they have associated with their accounts. This includes configuring multi-factor authorization (MFA), password complexity, and password retention requirements as applicable by accreditation policy. Through the use of the [AWS Identity and Access Management](#) (IAM) service, mission owners can implement role-based access control that properly separates users by their identified roles and responsibilities, thereby establishing least privilege and helping to ensure that users have only the permissions necessary to perform their assigned tasks.

To manage multiple AWS accounts, mission owners should leverage [AWS Organizations](#). AWS Organizations is an account management service that enables you to consolidate multiple AWS accounts into an *organization* that you create and centrally manage. AWS Organizations includes account management and consolidated billing capabilities that enable you to better meet your mission's budgetary, security, and compliance needs.

Identity federation

AWS offers multiple options for federating your identities in AWS. You can use IAM to enable users to sign in to their AWS accounts with their existing corporate credentials. AWS supports identity federation with on-premises authentication stores such as Lightweight Directory Access Protocol (LDAP) and Active Directory. With federation, you can use single sign-on (SSO) to access your AWS accounts using credentials from your organization's directory. Federation uses open standards, such as Security Assertion Markup Language 2.0 (SAML), to exchange identity and security information between an identity provider (IdP) and an application.

Multi-factor and CAC authentication

At a minimum, AWS mission owners should implement multi-factor authentication (MFA) for their AWS account credentials, as well as any privileged IAM accounts associated with AWS accounts. MFA can be used to add an additional layer of security in Amazon S3, through activation of the MFA delete feature.

The DoD has standardized on MFA through the use of the Common Access Card (CAC) or U.S. Government Personal Identity Verification (PIV) token. You can require your AWS users to authenticate to the AWS Management Console with a smart card by implementing SAML identity federation. You can also implement a RADIUS server to handle authentication requests to an AWS Managed Microsoft Active Directory instance. Mission owner applications that are migrated to AWS that currently require CAC authentication at the application layer operate exactly the same as they do within an on-premises data center environment.

Privileged remote access

Mission owners should implement privileged remote access for application and systems administrators to manage their AWS environments. There are several options for privileged remote access:

- [Amazon WorkSpaces](#)
- [Amazon AppStream 2.0](#)
- [AWS Systems Manager Session Manager](#)
- [EC2-based bastion hosts](#)

Amazon WorkSpaces is a managed, secure Desktop-as-a-Service (DaaS) solution that helps you decrease the complexity in managing hardware inventory, OS versions and patches, and Virtual Desktop Infrastructure (VDI).

Amazon WorkSpaces can be configured to restrict access to specific resources within designated VPC subnets and specific AWS services, controlled by the user's IAM policy or role. Users are able to access their WorkSpaces through an installable desktop client or using the Remote Desktop client. Both of these deployment options can be configured for CAC/PIV authentication. Amazon WorkSpaces is accredited at IL2, IL4, and IL5.

Amazon AppStream 2.0 is a fully managed application streaming service. You centrally manage your desktop applications on AppStream 2.0 and securely deliver them to any computer. For example, you are able to stream database management tools such as SQL Server Management Studio, web browsers such as Firefox and Chrome (restricted to certain URLs, if desired), as well as common office software.

Applications and data are not stored on users' computers. Your applications are streamed as encrypted pixels and access data secured within your network. Users are

able to authenticate with their CAC/PIV tokens through the use of identity federation. The Amazon AppStream 2.0 service is accredited at IL2, IL4, and IL5.

AWS Systems Manager Session Manager is a fully managed AWS Systems Manager capability that lets you manage your EC2 instances, on-premises instances, and virtual machines (VMs) through an interactive one-click browser-based shell or through the AWS CLI. Session Manager provides secure and auditable instance management without the need to open inbound ports, maintain bastion hosts, or manage SSH keys.

Session Manager also allows you to comply with security policies that require controlled access to instances, strict security practices, and fully auditable logs with instance access details, while still providing end users with simple one-click access to your managed instances across multiple operating systems. Users are able to authenticate with their CAC/PIV tokens through the use of AWS Management Console identity federation. Session Manager is a feature of AWS Systems Manager which is accredited at IL2, IL4, IL5 and IL6.

EC2-based bastion hosts are hardened instances used for administrative tasks within the AWS environment. Rather than allowing shell access to all EC2 instances from the public internet, access can be restricted to a single EC2 instance, thereby limiting the attack surface from possible compromise. Access to the bastion host should be through whitelisted IP addresses within the mission owner's organization, require valid SSH keys, and require multi-factor authentication. Auditing capabilities within the OS of the bastion host should be configured to record all administrative activity. These bastion hosts must be patched, hardened, and scanned in the same way as all other EC2 instances deployed within the mission environment.

Auditing

Mission owners are responsible for properly configuring their AWS services to ensure that required audit logs are generated. Audit logs should be forwarded to a dedicated log server instance or tool located within the mission owner's VPC management subnet or written to a secured and encrypted Amazon S3 bucket, ensuring that sensitive data is properly protected. The mission owner should enable the use of [AWS CloudTrail](#), a managed service that enables governance, compliance, operational auditing, and risk auditing of your AWS account.

With CloudTrail, you can log, nearly continuously monitor, and retain account activity related to actions across your AWS infrastructure. CloudTrail provides event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command line tools, and other AWS services. This event history

simplifies security analysis, resource change tracking, and troubleshooting. In addition, you can use CloudTrail to detect unusual activity in your AWS accounts.

Data protection and spillage

Following the Shared Responsibility Model, AWS customers are responsible for encryption and access control of their data within their AWS environments. According to published DISA guidance, all data at rest must also be encrypted. You can use [AWS Key Management Service](#) (AWS KMS) to help ensure that your data is encrypted at rest. For more information, refer to [AWS KMS Keys](#).

To provide protection against data spills, all mission owner data stored on Amazon EBS volumes and Amazon S3 must be encrypted using AES 256 encryption in accordance with DoD guidance. The mission owner is responsible for implementing FIPS 140-2 validated encryption for data at rest, with customer managed encryption keys in accordance with DoD policy. The combination of the mission owner's encryption and the automated wipe functionality that AWS provides can ensure that any spilled data is illegible ciphertext, greatly limiting the risk of accidental disclosure. AWS degausses and destroys all decommissioned media in accordance with National Institute of Standards and Technology (NIST) and National Security Agency (NSA) standards.

Intrusion detection

Mission owners are responsible for properly implementing host-based intrusion detection systems on their instances, as well as any required network-based intrusion detection. To assist mission owners in this endeavor, AWS provides native services like Amazon GuardDuty.

[Amazon GuardDuty](#) is a threat detection service that nearly continuously monitors for malicious activity and unauthorized behavior to protect your AWS accounts and workloads. With the cloud, the collection and aggregation of account and network activities is simplified, but it can be time consuming for security teams to nearly continuously analyze event log data for potential threats. With GuardDuty, you now have an intelligent and cost-effective option for nearly continuous threat detection in the AWS Cloud. GuardDuty is accredited at IL2, IL4, and IL5.

Mission owners are responsible for coordinating deployment of their intrusion detection capabilities with their Cyber Security Service Provider (CSSP). Mission owners can implement the Secure Cloud Computing Architecture (SCCA) within AWS to help them meet their compliance and security requirements. More information about the SCCA architecture can be found in the IL2, IL4, and IL5 sample reference architecture section of this document.

Compliance and governance

Mission owners are required to maintain strong governance over the entire IT environment, regardless of whether it is deployed in a traditional data center or in the AWS Cloud. Best practices include:

- Understanding your workload's required compliance objectives
- Establishing a control environment that meets those objectives and requirements
- Understanding the requirements for validation based on the organization's risk tolerance
- Verifying the operating effectiveness of the control environment

Deployment of workloads in the AWS Cloud gives you options to apply various types of controls and utilize multiple verification methods.

To help mission owners meet DoD compliance and governance requirements, a mission owner must perform the following basic steps:

1. Review information from AWS and other sources to understand how their cloud environment is architected and configured.
2. Document all relevant DoD compliance requirements that may be in scope for their workloads in the cloud.
3. Design and implement control objectives to meet the organization's security and compliance requirements.
4. Identify and document controls owned by outside or third parties.
5. Verify that all control objectives are met and all key controls are designed and operating effectively.

Approaching compliance and governance in this manner will help mission owners gain a better understanding of their environment and will help clearly delineate any verification activities that need to be performed.

FedRAMP

The Federal Risk and Authorization Management Program (FedRAMP) is a U.S. government-wide program that provides a standardized approach to security assessment, authorization, and nearly continuous monitoring for cloud products and services.³

The DoD SRG uses the FedRAMP program to establish a standardized approach for DoD entities that are utilizing commercial cloud services. AWS has been assessed and approved under FedRAMP, and has been issued two Agency Authority to Operate (ATO) authorizations covering all 48 Contiguous States and the District of Columbia (CONUS) [Regions](#) which include AWS GovCloud (US), US East and US West. For more information on FedRAMP compliance of the AWS Cloud, visit our [FedRAMP FAQ](#) page. All cloud service providers must demonstrate compliance with FedRAMP standards before they can be considered for a provisional authorization under the CC SRG by DoD.

Cloud Computing Security Requirements Guide

The DoD CC SRG provides a formalized assessment and authorization process for cloud service providers to obtain a DoD Provisional Authorization (PA), which can then be leveraged by mission owners. These provisional authorizations provide reusable certifications that attest to the compliance of specific AWS Regions and services in alignment with DoD standards, reducing the time necessary for a DoD mission owner to assess and authorize their workloads for migration to AWS.

The CC SRG supports the overall goal of the U.S. federal government to increase the utilization of commercial cloud computing and it provides a means for the DoD to support this goal. The CC SRG requires the categorization of mission systems and their workloads at one of four (4) Impact Levels. Each level represents a determination of the data sensitivity of a particular system and the controls required to protect it, starting at level 2 (lowest) through level 6 (highest).

The following table summarizes the impact levels with a description of a typical workload, connectivity restrictions, Boundary Cloud Access Point (BCAP) requirements, and Computer and Network Defense (CND) requirements.

Table 1 – Security requirements

Impact Level	Information Sensitivity	Security Controls	Location	Off-Premises Connectivity	Separation	Personnel Requirements
2	PUBLIC or non-critical mission information	FedRAMP Moderate	US / US outlying areas or DoD on-premises	Internet	Virtual / Logical Public Community	National Agency Check and Inquiries (NACI)

Impact Level	Information Sensitivity	Security Controls	Location	Off-Premises Connectivity	Separation	Personnel Requirements
4	CUI or Non-CUI	Level 2 +	US / US	NIPRNet via	Virtual / Logical	US Persons
	Non-critical mission information	CUI-specific Tailored Set	outlying areas or DoD on-premises	CAP	Limited "Public" Community	ADP-1 Single Scope Background Information (SSBI)
	Non-National Security Systems				Strong virtual separation between tenant systems and information	ADP-2
5	Higher Sensitivity CUI	Level 4 +	US / US	NIPRNet via	Virtual / Logical	National Agency Check with Law and Credit (NACLC)
	Mission critical information	NSS and CUI-specific Tailored Set	outlying areas or DoD on-premises	CAP	Federal Government Community	Nondisclosure Agreement (NDA)
	National Security Systems				Dedicated multi-tenant infrastructure physically separate from non-federal systems Strong virtual separation between tenant systems and information	
6	Classified SECRET	Level 5 +	US / US	SIPRNET	Virtual / Logical	US citizens with favorably adjudicated SSBI and SECRET clearance
	National Security Systems	Classified Overlay	outlying areas or DoD on-premises	DIRECT	Federal Government Community	NDA
			CLEARED/CLASSIFIED FACILITIES	With DoD SIPRNet Enclave Connection Approval	Dedicated multi-tenant infrastructure physically separate from non-federal and unclassified systems	

AWS holds a provisional authorization for Impact Level 2 workloads within US East and US West, which permits mission owners to deploy public, unclassified information in these AWS Regions with both the AWS authorization and the mission application's ATO.

AWS GovCloud holds a provisional authorization for Impact Levels 2, 4, and 5, and permits mission owners to deploy the full range of controlled, unclassified information categories covered by these levels. The AWS Secret Region holds a provisional authorization for Impact Level 6 and permits workloads up to and including Secret classification.

To begin planning for the deployment of a DoD mission system in AWS, it is critical that the CC SRG impact level categorization be made in advance. Systems designated at Impact Level 2 can begin deployments relatively quickly. Conversely, a designation at Impact Level 4 or 5 requires that the mission application on AWS be connected to the Non-secure Internet Protocol Router Network (NIPRNet) by means of AWS Direct Connect, Internet Protocol Security (IPsec) virtual private network (VPN), or both. This NIPRNet connection also requires that the traversal of all inbound and outbound traffic to and from the mission owner's VPC be routed through a Border Cloud Access Point (BCAP) or equivalent DoD CIO approved boundary and its associated CND suite.

The provisioning of circuits for an AWS Direct Connect to NIPRNet connection typically has a substantial lead-time, so mission owners should plan accordingly. Mission owners can also take advantage of existing Cloud Access Points that have been set up by various DoD agencies or CSSPs, including DISA.

For more information regarding the Department of Defense CC SRG, refer to the DISA cyber.mil website for the latest [Cloud Security announcements and requirements](#) or the latest [CC SRG v1.3](#) document. For more information on the DISA Cloud Access Point, refer to the [DISA Cloud Connection Process Guide](#).

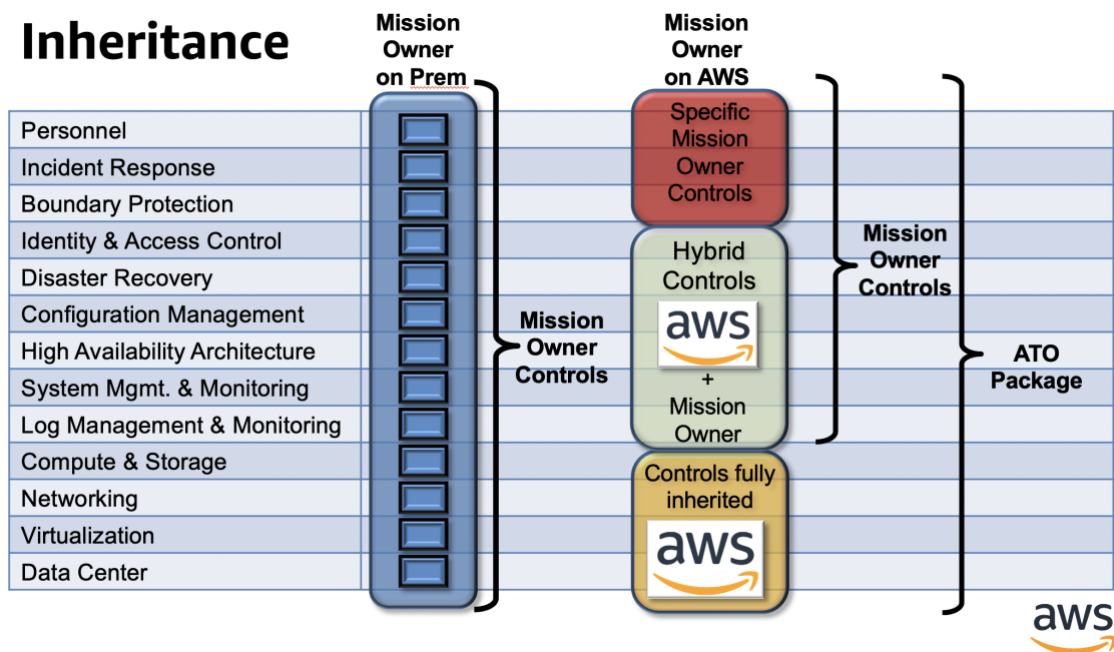
FedRAMP + CC SRG compliance = the path to AWS

For DoD application owners to obtain an Authority to Operate (ATO) for their cloud-deployed applications from their approving authority, they must select a cloud service provider that has obtained a provisional authorization from DoD. Gaining authorization under FedRAMP is the first step toward gaining authorization from DoD. There are four paths into the FedRAMP repository; the Joint Authorization Board (JAB) and Agency ATO paths are the most common.

If a CSP wants to go beyond FedRAMP and become a DoD CSP, the CSP must go through the DoD CC SRG assessment process. Currently, attaining a FedRAMP Moderate authorization enables a CSP to be considered for Impact Level 2 of the CC SRG, while an additional assessment is required against the FedRAMP+ controls of Impact Levels 4 and 5 prior to being granted a provisional authorization at those levels.

Regardless of whether the Designated Accrediting Authority (DAA) is using the DoD Information Assurance and Certification Accreditation Process (DIACAP) or the Risk Management Framework (RMF) process, the DAA has the ability to leverage and inherit the Provisional Authorization package(s) as part of its assessment toward a final ATO which only it grants (not the Defense Information Systems Agency (DISA)). The RMF process has been formally adopted by the DoD. Mission owners can request the AWS

FedRAMP package to get a better understanding of compliance and security processes that AWS abides by.



Controls inheritance and responsibilities

AWS global infrastructure

AWS provides facilities and hardware in support of mission owners with security features controlled by AWS at the infrastructure level. In the infrastructure as a service (IaaS) model, AWS is responsible for applicable service delivery layers including:

- **Infrastructure** (hardware and software that comprise the infrastructure)
- **Service management processes** (the operation and management of the infrastructure and the system and software engineering lifecycles)

Mission owners use AWS to manage the cloud infrastructure including the network, data storage, system resources, data centers, security, reliability, and supporting hardware and software.

Across the globe, the infrastructure of AWS is organized into Regions. [Each Region contains Availability Zones](#), which are located within a particular geographic area that allows for low-latency communication between the zones. Customer data resides within a particular Region and does not move to a different Region unless the customer explicitly takes this action.

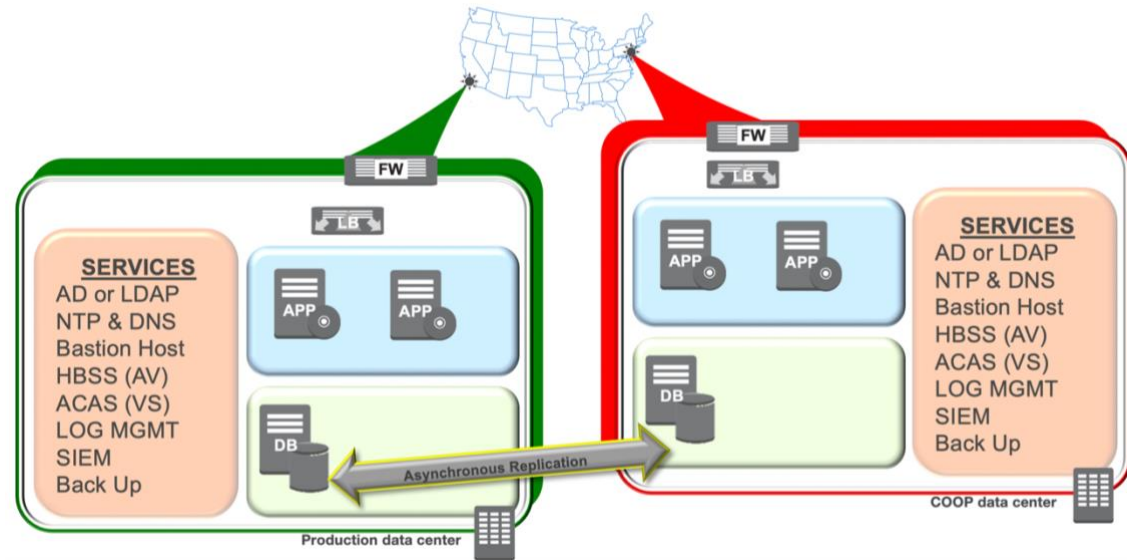
Currently, there are seven Regions available within CONUS that are permitted for use by the DoD. They are:

- **US East (IL2)**
 - us-east-1 (Northern Virginia)
 - us-east-2 (Ohio)
- **US West (IL2)**
 - us-west-1 (Northern California)
 - us-west-2 (Oregon)
- **AWS GovCloud (IL4, IL5, ITAR and export-controlled workloads)**
 - us-gov-west-1 (Oregon)
 - us-gov-east-1 (Ohio)
- **AWS Secret Region (IL6)**

Each Availability Zone has an identical cloud services, offering compute, storage and networking, among other functionality, that enables mission owners to deploy applications and services with flexibility, scalability, and reliability. AWS provides mission owners with the option to choose only the services they require and the ability to provision or release them as needed.

Architecture

Traditional DoD data center



Traditional three-tier data center architecture

A typical DoD three-tier data center architecture might consist of the following:

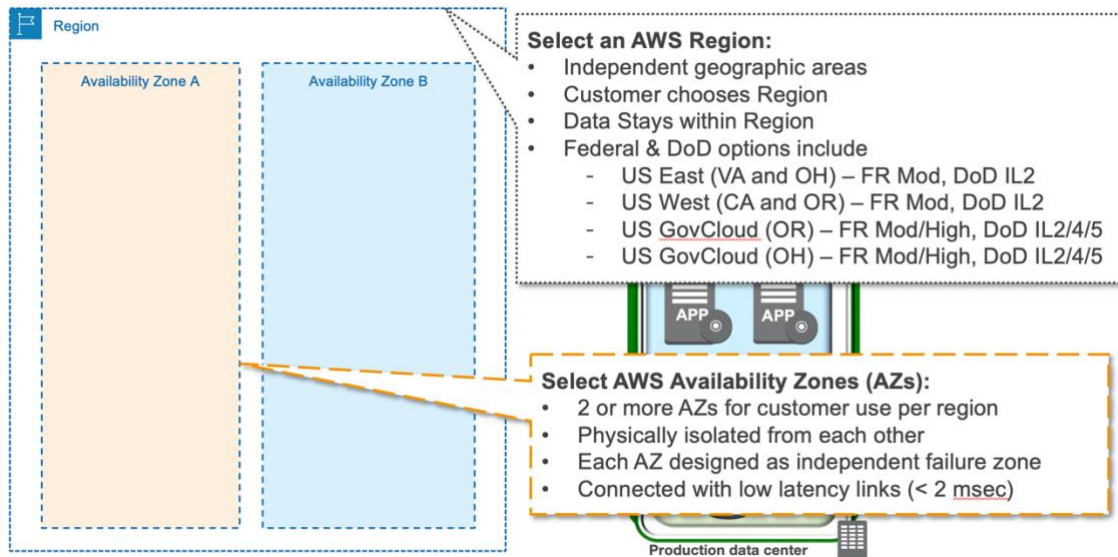
- Two data center locations; one hosting the production environment and one hosting the COOP or DR environment.
- Each system consists of three distinct tiers or network enclaves. Each enclave is defined by separate subnets or VLANs.
- Network isolation and control between enclaves is maintained by a firewall. This isolation allows the web tier to communicate with the application tier and the application tier to communicate with the database tier. Direct external access to the application and web tiers is prohibited.
- A load balancer is used to distribute traffic across the web servers and may also provide SSL/TLS offloading.

Because of the distance between these systems and the network connectivity, the data replication between databases is asynchronous. In addition to the three-tier web, application and database components, additional “shared” or common services are needed to support the infrastructure as a whole. These services may be dedicated to this application or may be leveraged to support multiple applications.

DoD compliant cloud environment

Migrating mission workloads to a DoD compliant environment in the AWS Cloud is achievable through the following high-level steps.

Step 1 – Find a “home” in the AWS Cloud

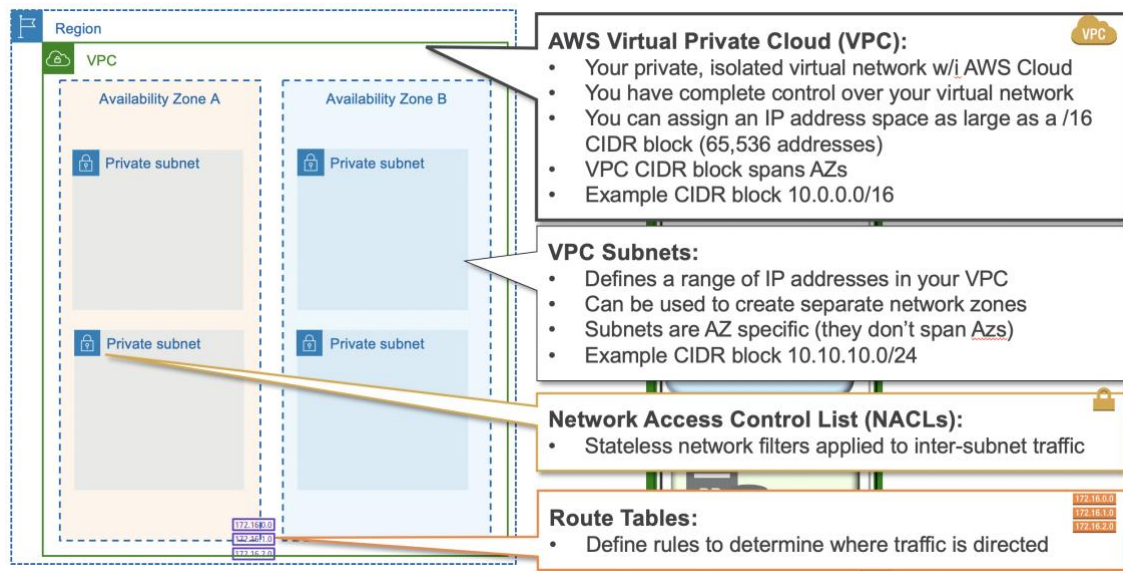


Planning migration to AWS Regions and Availability Zones

Concepts

- [AWS Region](#)
- [AWS Availability Zone](#)

Step 2 – Define your network in AWS

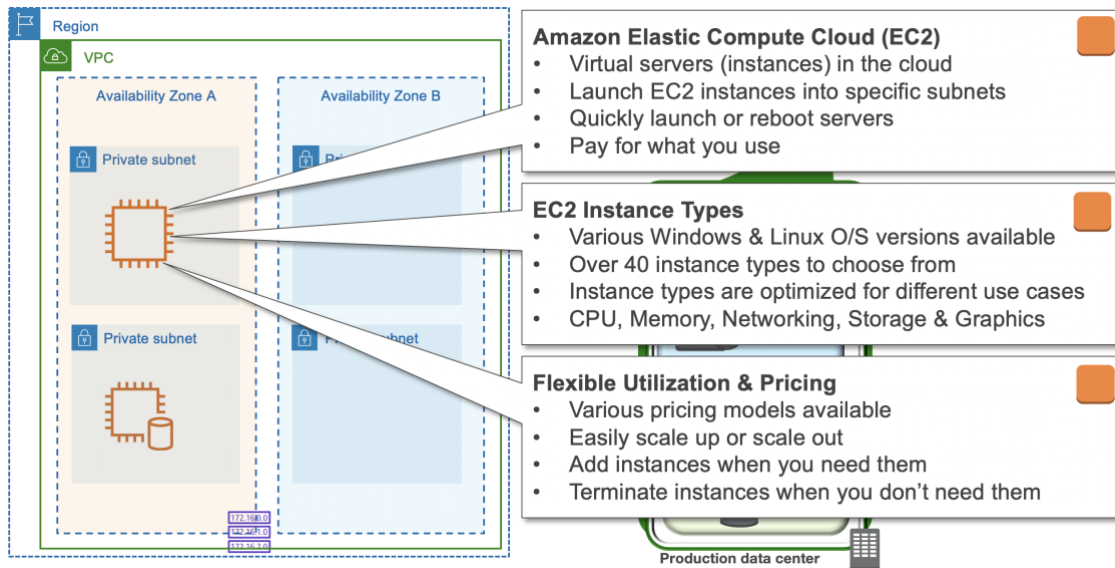


Configuring VPC, subnets, NACLs, and route tables

Concepts

- [Virtual Private Cloud \(VPC\)](#)
- [VPC Subnet](#)
- [Network Access Control List \(Network ACL\)](#)
- [VPC Route Table](#)

Step 3 – deploy servers (or containers or serverless infrastructure)

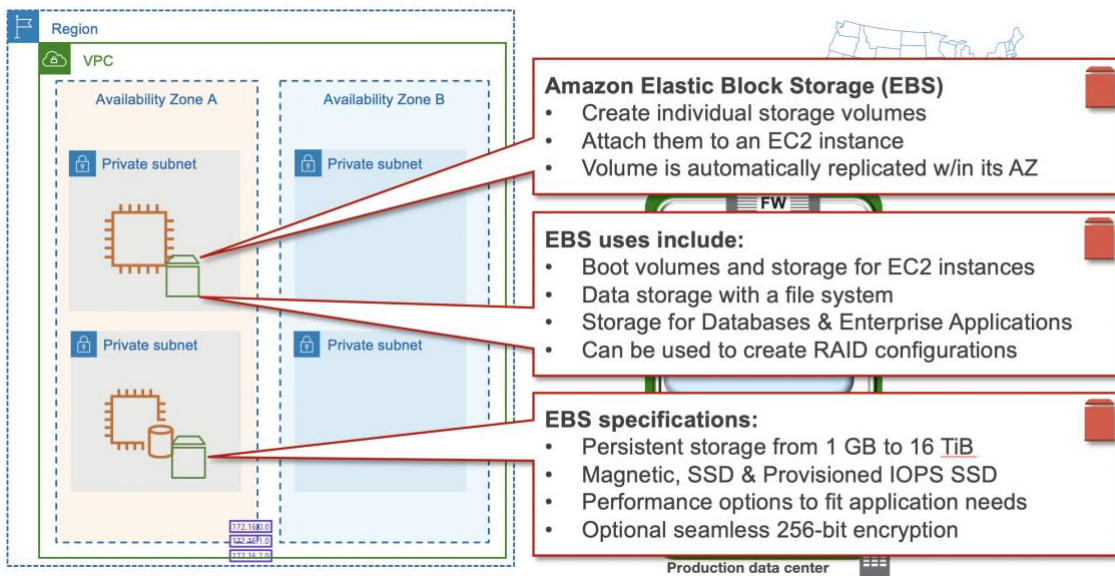


Deploying Amazon EC2 instances in your subnets

Concepts

- [Amazon Elastic Compute Cloud \(EC2\)](#)

Step 4 – Add storage

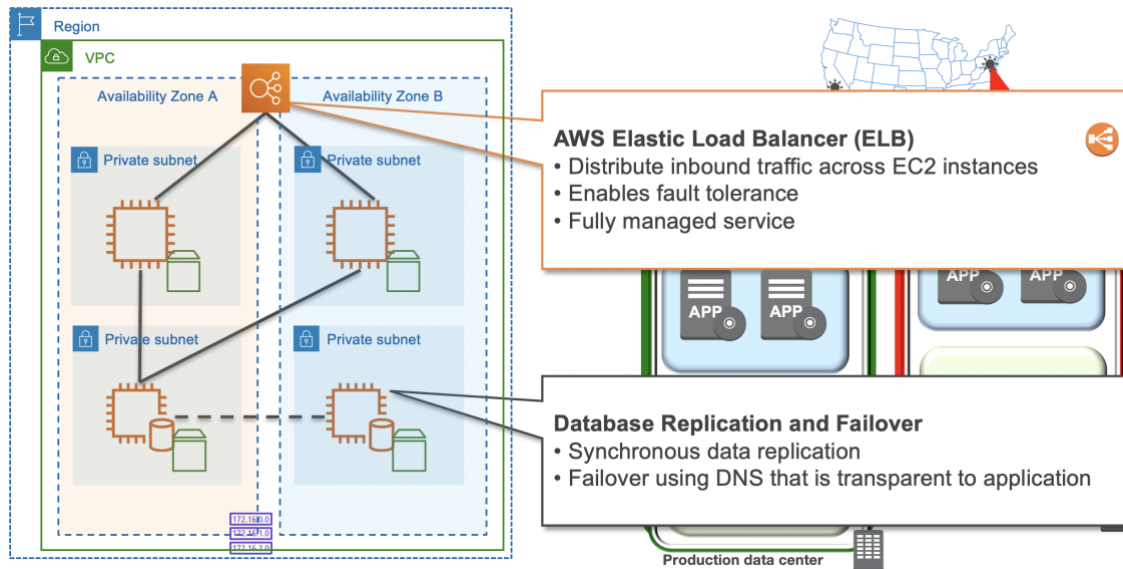


Creating and attaching Amazon EBS volumes to your Amazon EC2 instances

Concepts

- [Amazon EBS](#)
- [Amazon S3](#)
- [Amazon S3 Glacier](#)

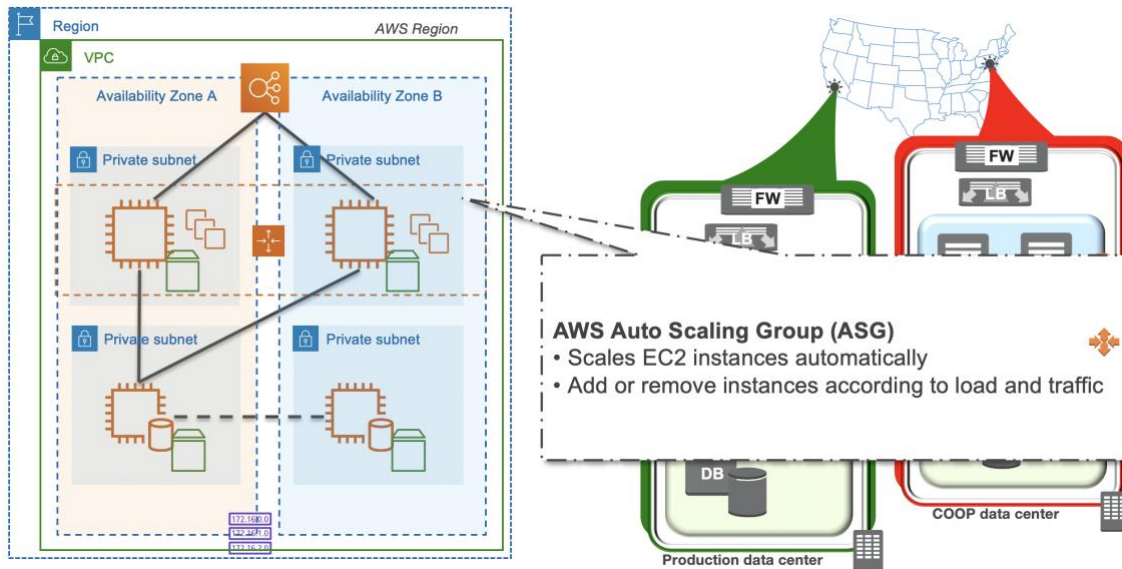
Step 5 – Add scalability, redundancy and failover



Adding ELB to handle traffic coming to your EC2 instances

Concepts

- [Multi-AZ Architecture](#)
- [Elastic Load Balancing](#) (ELB)

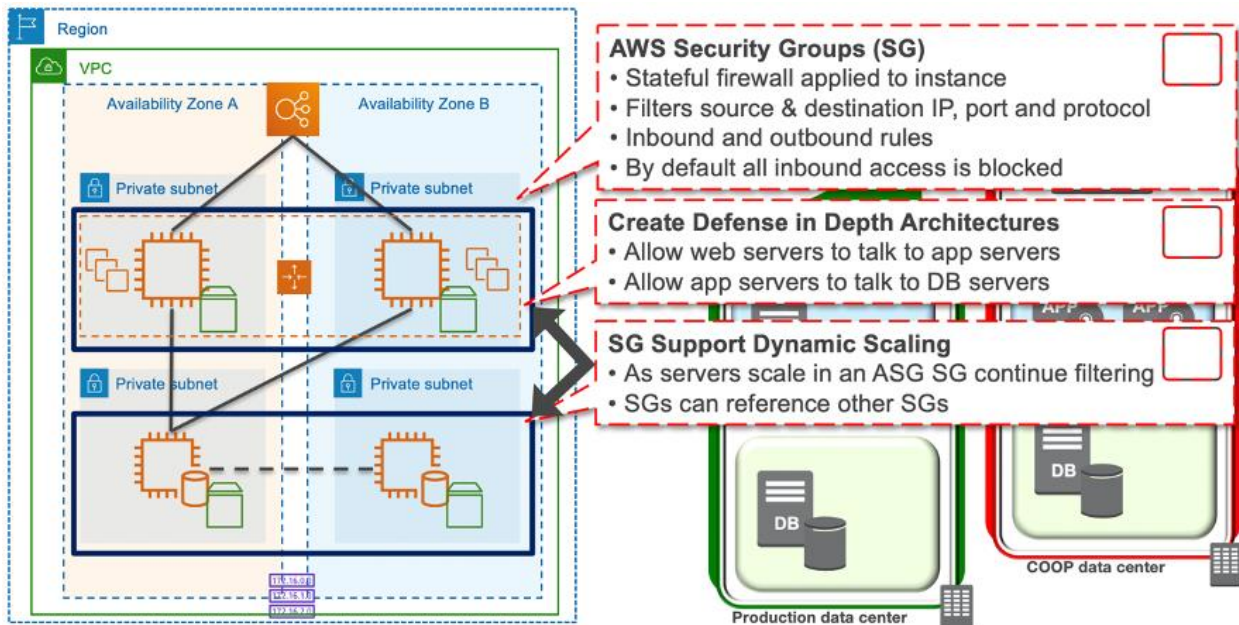


Adding an Auto Scaling Group to increase and decrease your compute capacity

Concepts

- [Amazon EC2 Auto Scaling](#)

Step 6 – Implement network traffic filtering

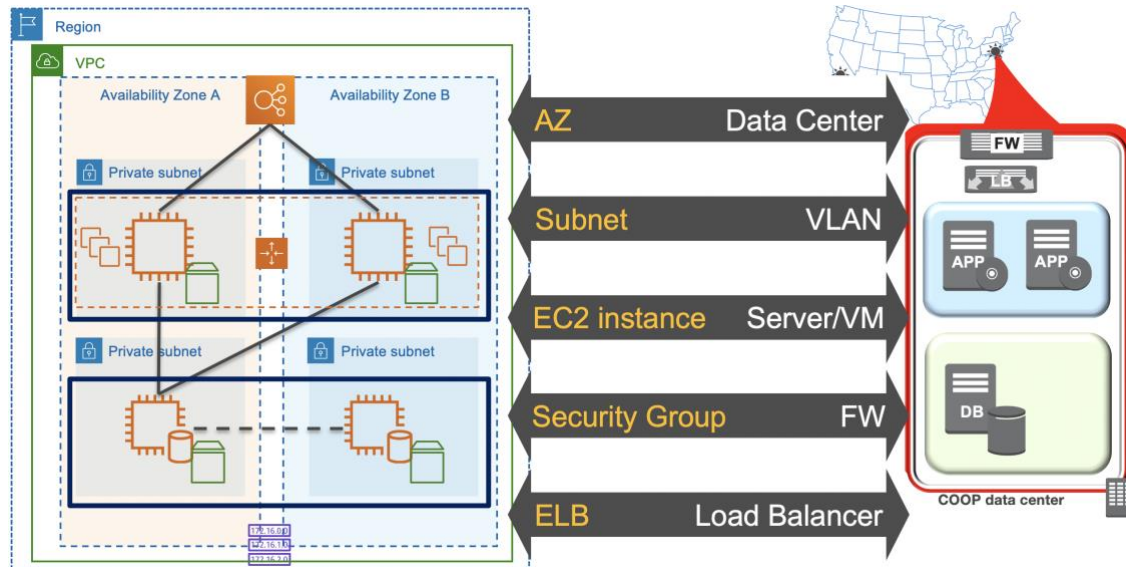


Adding security groups to your VPC

Concepts

- [AWS security groups](#)
- [Defense in depth](#)

Recap

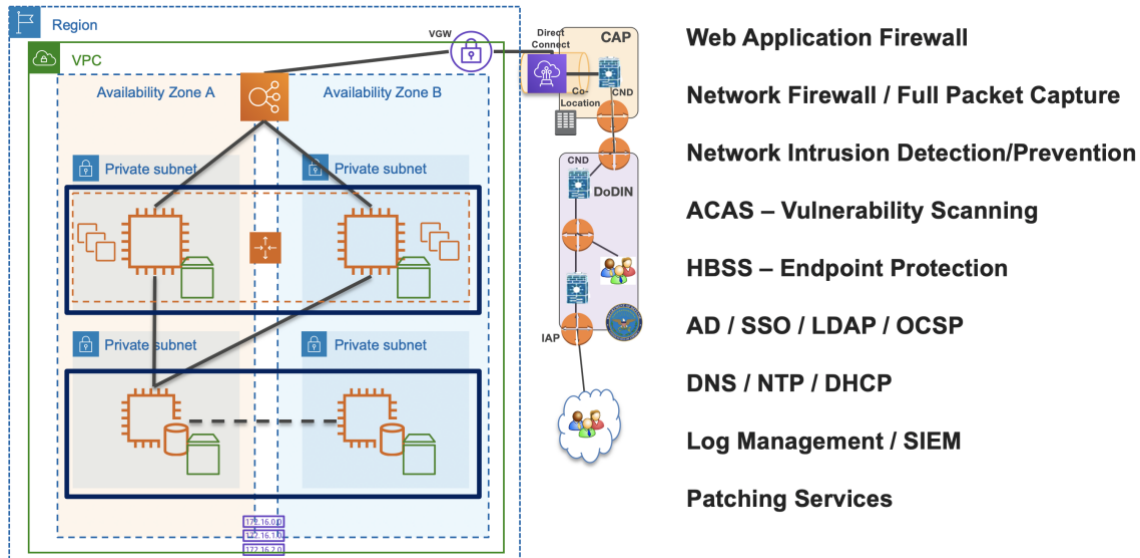


Comparison of AWS Cloud architecture and on-premises data centers

Availability Zones are analogous to data centers. Subnets are analogous to layer 3 VLANs. EC2 instances are analogous to servers or virtual machines. Security **groups** are analogous to stateful firewalls.

Shared services

There are several other components that are required to support a DoD compliant environment in AWS. The DoD CC SRG stipulates that IL4+ workloads require protection by a web application firewall including network intrusion detection/prevention, full packet capture functionality, vulnerability scanning, endpoint protection, identity and access control (including public key infrastructure (PKI)), common services (DNS/NTP), as well as log management and patching capabilities.



Additional components required to support a DoD-compliant environment in AWS

AWS services

Compute

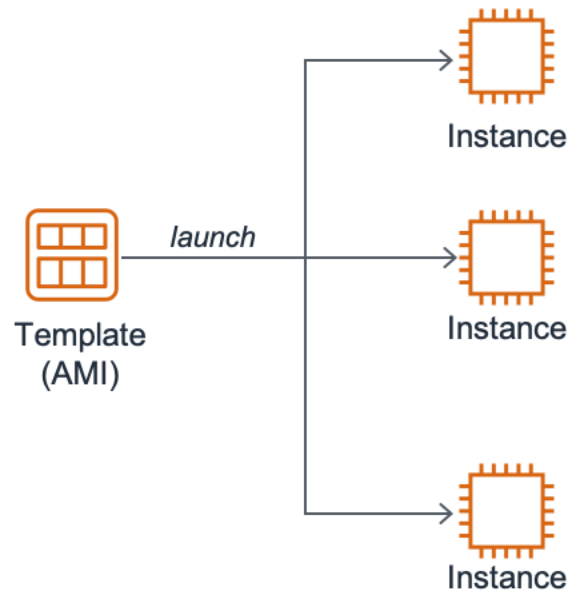
Amazon Elastic Compute Cloud (EC2)

[Amazon EC2](#) is a web service that provides virtual server instances that can be used to build and host software systems. Amazon EC2 facilitates web-scale computing by enabling mission owners to deploy virtual machines on demand. The simple web service interface allows mission owners to obtain and configure capacity with minimal friction, and it provides complete control over-computing resources. Amazon EC2 changes the economics of computing by allowing organizations to avoid large capital expenditures and instead pay only for capacity that is actually used.

Amazon EC2 functionality and features include:

- **Elastic** – Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing mission owners to quickly scale capacity, both up and down, as computing requirements change.

- **Flexible** – The mission owner can choose among various options for number of CPUs, memory size, and storage size. A highly reliable and fault-tolerant system can be built using multiple EC2 instances. EC2 instances are very similar to traditional virtual machines or hardware servers. EC2 instances use operating systems such as Windows or Linux. They can accommodate most software that can run on those operating systems. EC2 instances have IP addresses, so the usual methods of interacting with a remote machine, such as Secure Shell (SSH) and Remote Desktop Protocol (RDP), can be used.
- **Amazon Machine Image (AMI)** – AMI templates are used to define an EC2 server instance. Each AMI contains a software configuration, including operating system, application server, and applications applied to an instance type. Instance types in Amazon EC2 are essentially hardware archetypes matched to the amount of memory (RAM) and computing power (number of CPUs) needed for the application.



Using AMI templates to launch Amazon EC2 instances

- **Custom AMI** – The first step toward building applications in AWS is to create a library of customized AMIs. Starting an application then becomes a matter of launching the AMI. For example, if an application is a website or web service, the AMI should be configured with a web server (for example, Apache, Nginx or Microsoft Internet Information Services), the associated static content, and the code for all dynamic pages.

Alternatively, the AMI could be configured to install all required software components and content by running a bootstrap script as soon as the instance is launched. As a result, after launching the AMI, the web server will start and the application can begin accepting requests. After an AMI has been created, replacing a failing instance is very simple; a replacement instance can easily be launched that uses the same AMI as its template.

- **EC2 local instance store volumes** – These volumes provide temporary block-level storage for EC2 instances. When an EC2 instance is created from an AMI, in most cases, it comes with a preconfigured block of pre-attached disk storage. Unlike Amazon EBS volumes, data on instance store volumes persists only during the life of the associated EC2 instance, and they are not intended to be used as durable disk storage.

Data on EC2 local instance store volumes is persistent across orderly instance reboots, (following OS vendor procedure for rebooting the underlying operating system), but not in situations where the EC2 instance shuts down or goes through a failure/restart cycle. *Local instance store volumes should not be used for any data that must persist over time, such as permanent file or database storage.* Although local instance store volumes are not persistent, the data can be persisted by periodically copying or backing it up to Amazon EBS or Amazon S3.

- **Mission-owner controlled** – Mission owners have complete control of their instances. They have root access to each one and can interact with them as they would any machine. Mission owners can stop their instance while retaining the data on a boot partition and then subsequently restart the same instance using web service APIs. Instances can be rebooted remotely using web service APIs. Mission owners also have access to the AWS Management Console to view and control their instances.
- **API Management** – Managing instances can be done through an API call, scriptable command line tools, or the AWS Management Console. Being able to quickly launch replacement instances based on a custom AMI is a critical first step towards fault tolerance. The next step is storing persistent data that these server instances use.

- **Multiple Availability Zones** – Amazon EC2 provides the ability to place instances in multiple Availability Zones. Availability Zones are distinct locations that are engineered to be insulated from failures in other Availability Zones. They provide inexpensive, low latency network connectivity to other zones in the same Region. By launching instances in separate Availability Zones, mission owners can protect their applications from failure of a single location. Regions consist of one or more Availability Zones.
- **Reliable** – The [Amazon EC2 Service Level Agreement](#) (SLA) commitment is 99.95% availability for each EC2 Region.
- **Elastic IP Addresses** – Elastic IP addresses are static IP addresses designed for dynamic cloud computing. An Elastic IP address is associated with a mission owner account and not with a particular instance. So, mission owners control that address until they choose to explicitly release it. Unlike traditional static IP addresses, however, Elastic IP addresses can be programmatically remapped to any instance in their account in the event of an instance or Availability Zone failure.

Mission owners don't need to wait for a network technician to reconfigure or replace a host or wait for the Domain Name System (DNS) to propagate. In addition, mission owners can optionally configure the reverse DNS record of any of their Elastic IP addresses.

Scalable (Durability) – [AWS Auto Scaling](#) is a web service that enables mission owners to automatically launch or terminate Amazon EC2 instances based on user-defined policies, health status checks, and schedules. For applications configured to run on a cloud infrastructure, scaling is an important part of cost control and resource management.

Scaling is the ability to increase or decrease the compute capacity of an application by either changing the number of servers (horizontal scaling) or changing the size of the servers (vertical scaling). In a typical situation, when the web application starts to get more traffic, the mission owner either adds more servers or increases the size of existing servers to handle the additional load.

Similarly, if the traffic to the web application starts to slow down, the under-utilized servers can be shut down or the size of the existing servers can be decreased. Depending on the infrastructure involved, vertical scaling might involve changes to server configurations every time the application scales. With horizontal scaling, AWS simply increases or decreases the number of servers according to the application's demands.

The decision when to scale vertically and when to scale horizontally depends on factors such as the mission owner's use case, cost, performance, and infrastructure.

When using Auto Scaling, mission owners can increase the number of servers in use automatically when the user demand goes up to ensure that performance is maintained and can decrease the number of servers when demand goes down to minimize costs. Auto Scaling helps make efficient use of compute resources by automatically doing the work of scaling for the mission owner. This automatic scaling is the core value of the Auto Scaling service.

Auto Scaling is well suited for applications that experience hourly, daily, or weekly variability in usage and need to automatically scale horizontally to keep up with changes in usage. Auto Scaling frees users from having to predict traffic spikes accurately and plan for provisioning resources in advance of them. With Auto Scaling, mission owners can build a fully scalable and affordable infrastructure in the cloud.

Networking

Amazon Virtual Private Cloud (VPC)

AWS enables a mission owner to create the equivalent of a "virtual private enclave" with the Amazon VPC service. Amazon VPC is used to provision a logically isolated section of the AWS Cloud where a customer can launch AWS resources in a virtual network that is defined by the mission owner.

This logically separate space within AWS contains compute and storage resources that can be connected to a mission owner's existing infrastructure through a virtual private network (VPN) connection, AWS Direct Connect (private) connection, and/or the internet. With Amazon VPC, it is then possible to extend existing DoD directory services, management tools, monitoring/security scanning solutions, and inspection capabilities, thus maintaining a consistent means of protecting information whether it is residing on internal DoD IT resources or in AWS.

Network isolation and the ability to demonstrate separation of infrastructure and data is applicable at Impact Levels 2, 4, and 5, and it is a key requirement of the CC SRG for Impact Levels 4 and 5.

The CC SRG requires Impact Level 4 and 5 mission applications to be connected to NIPRNet without direct internet access from the VPC.

Mission owners have complete control over the definition of the virtual networking environment within their VPC, including the selection of a private (RFC 1918) address range of their choice (for example, 10.0.0.0/16), the creation of subnets, the configuration of route tables, and the inclusion or exclusion of network gateways. Further, mission owners can define the subnets within their VPC in a way that enables them to group similar kinds of instances based on IP address range.

Mission owners can use VPC functionality and features in the following ways:

- Mission owners can define a VPC on scalable infrastructure and specify its private IP address range from any range they choose.
- Mission owners can sub-divide a VPC's private IP address space further into one or more public or private subnets according to application requirements and security best practices. This can facilitate running applications and services in a customer's VPC.
- Mission owners define inbound and outbound access to and from individual subnets using network access control lists.
- Data can be stored in Amazon S3 with set permissions ensuring that the data can only be accessed from within a mission owner's VPC.
- An Elastic IP address can be attached to any instance in a mission owner's VPC so it can be reached directly from the internet (Impact level 2 only).
- A mission partner's VPC can be bridged with their onsite DoD IT infrastructure (encapsulated in an encrypted VPN connection) to extend existing security and management policies to the VPC instances as if they were running within the mission partner's physical infrastructure.

Amazon VPC provides advanced security features, such as security groups and network access control lists, to enable inbound and outbound filtering at the instance level and subnet level. When building a VPC, mission owners must define the subnets, routing rules, security groups, and network access control lists (NACLs) that comply with the networking and security requirements of the DoD and their organization.

Subnets

VPCs can span multiple Availability Zones. After creating a VPC, mission owners can add one or more subnets in each Availability Zone. Each subnet must reside entirely within one Availability Zone, cannot span zones, and is assigned a unique ID by AWS.

Routing

By design, each subnet must be associated with a route table that specifies the allowed routes for outbound traffic leaving the subnet. Every subnet is automatically associated with the main route table for the VPC. By updating the association, mission owners can change the contents of the main route table.

Mission owners should know the following basic things about VPC route tables:

- The VPC has an implicit router.
- The VPC comes with a main route table that mission owners can modify.
- Mission owners can create additional custom route tables for their VPC.
- Each subnet must be associated with a route table, which controls the routing for the subnet. If a mission owner does not associate a subnet with a particular route table, the subnet uses the main route table.
- Mission owners can replace the main route table with a custom table that they have created (this table becomes the default table each new subnet is associated with).
- Each route in a table specifies a destination Classless Inter-Domain Routing (CIDR) block and a target (for example, traffic destined for 172.16.0.0/12 is targeted for the virtual private gateway). Amazon VPC uses the most specific route that matches the traffic to determine how to route the traffic.

Security groups and network ACLs

AWS provides two features that mission owners can use to increase security in their VPC: security groups and network access control lists (NACLs). Both features enable mission owners to control the inbound and outbound traffic for their instances.

Security groups work at the instance level, and access control lists (ACLs) work at the subnet level. security groups default to deny all and must be configured by the mission owner to permit traffic.

Security groups provide stateful filtering at the instance level and can meet the network security needs of many AWS mission owners. However, VPC users can choose to use both security groups and network ACLs to take advantage of the additional layer of security that network ACLs provide.

An ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet. Mission owners can set up network ACLs with rules similar to those implemented in security groups to add a layer of stateless filtering to their VPC.

Mission owners should know the following basic things about network ACLs:

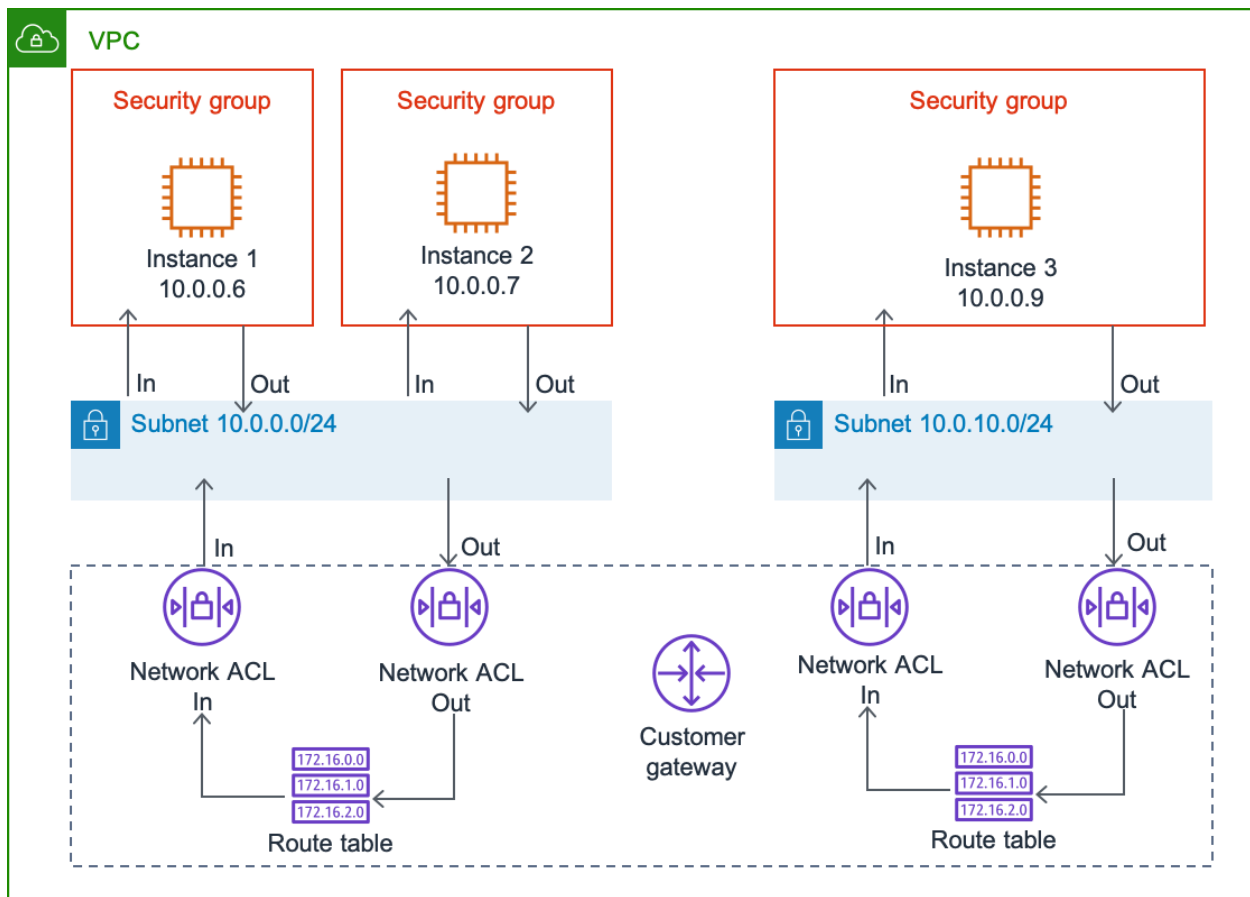
- A network ACL is a numbered list of rules that is evaluated in order, starting with the lowest-numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL. The highest rule number available for use is 32766. We suggest that mission owners start by creating rules with rule numbers that are multiples of 100, so that new rules can be inserted later on.
- A network ACL has separate inbound and outbound rules, and each rule can either allow or deny traffic.
- Each VPC automatically comes with a modifiable default network ACL; by default, it allows all inbound and outbound traffic.
- Each subnet must be associated with a network ACL; if mission owners don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default network ACL.
- Mission owners can create custom network ACLs; each custom network ACL starts out closed (permits no traffic) until the mission owner adds a rule.
- Network ACLs are stateless; responses to allow inbound traffic are subject to the rules for outbound traffic (and vice versa).

The following table summarizes the basic differences between security groups and network ACLs. Inbound traffic will first be processed according to the rules of the network ACL applied to a subnet, and subsequently by the security group applied at the instance level.

Table 2 — Differences between security groups and network ACLs

Security Group	Network ACL
Operates at the instance level (first layer of defense)	Operates at the subnet level (additional layer of defense)
Supports allow rules only	Supports allow rules and deny rules
Is stateful; return traffic is automatically allowed, regardless of any rules	Is stateless; return traffic must be explicitly allowed by rules
All rules are evaluated before deciding whether to allow traffic	Rules are processed in order when deciding whether to allow traffic
Applies to an instance only if someone specifies the security group when launching the instance, or associates the security group with the instance after the instance is launched	Automatically applies to all instances in the subnets it's associated with (backup later of defense, so you don't have to rely on someone specifying the security group)

The following diagram illustrates the layers of security provided by security groups and network ACLs. For example, traffic from an internet gateway is routed to the appropriate subnet using the routes in the routing table. The rules of the network ACL associated with the subnet control which traffic is allowed to the subnet. The rules of the security group associated with an instance control which traffic is allowed to the instance.



Security layers provided by security groups and network ACLs

Storage

There are three common storage options for instances and/or resources that can be utilized in conjunction with a system hosted within an Amazon VPC. The three storage types are Amazon S3, Amazon EBS, and instance storage, each of which has distinct use cases.

Amazon S3

[Amazon S3](#) is a highly durable repository designed for mission critical and primary data storage for mission owner data. It enables mission owners to store and retrieve any amount of data, at any time, from within Amazon EC2 or anywhere on the web.

Amazon S3 stores data objects redundantly on multiple devices across multiple facilities and allows concurrent read or write access to these data objects by many separate clients or application threads. Amazon S3 is designed to protect data and allow access to it even in the case of a failure of a data center.

Additionally, mission owners can use the redundant data stored in Amazon S3 to recover quickly and reliably from instance or application failures. The Amazon S3 versioning feature allows the retention of prior versions of objects stored in Amazon S3 and also protects against accidental deletions initiated by staff or software error. Versioning can be enabled on any Amazon S3 bucket.

Mission owners should know the following basic things about Amazon S3 functionality and features:

- Mission owners can write, read, and delete objects containing from 1 byte to 5 terabytes of data each. The number of objects mission owners can store is unlimited.
- Each object is stored in an Amazon S3 bucket and retrieved via a unique, developer-assigned key.
- Objects stored in an AWS Region never leave unless the mission owner transfers them out.
- Authentication mechanisms are provided to ensure that data is kept secure from unauthorized access. Objects can be made private or public, and rights can be granted to specific users.
- Options for secure data upload and download and encryption of data at rest are provided for additional data protection.
- Amazon S3 uses standards-based REST and SOAP interfaces designed to work with any internet-development toolkit.
- Amazon S3 is built to be flexible so that protocol or functional layers can easily be added.
- Amazon S3 includes options for performing recurring and high-volume deletions. For recurring deletions, rules can be defined to remove sets of objects after a pre-defined time period. For efficient one-time deletions, up to 1,000 objects can be deleted with a single request.

For more information on these Amazon S3 features, consult the [Amazon S3 documentation](#).

Amazon Elastic Block Store

[Amazon EBS](#) provides block-level storage volumes for use with Amazon EC2 instances.

Amazon EBS volumes are highly available and reliable storage volumes that can be attached to any running instance that is in the same Availability Zone. Amazon EBS volumes that are attached to an Amazon EC2 instance are exposed as storage volumes that persist independently from the life of the instance. With Amazon EBS, users pay only for what they use.

Amazon EBS is recommended when data changes frequently and requires long-term persistence. Amazon EBS volumes are particularly well suited for use as the primary storage for file systems, databases, or for any applications that require fine granular updates and access to raw, unformatted, block-level storage. Amazon EBS is particularly helpful for database-style applications that frequently encounter many random reads and writes across the dataset.

Mission owners can attach multiple volumes to the same instance within the limits specified by their AWS account. Currently, an AWS account is limited 300 TiB of total storage within EBS volumes.

Amazon EBS volumes store data redundantly, making them more durable than a typical hard drive. The annual failure rate for an Amazon EBS volume is 0.1% to 0.5%, compared to 4% for a commodity hard drive.

Amazon EBS and Amazon EC2 are often used in conjunction with one another when building an application on AWS. Any data that needs to persist can be stored on Amazon EBS volumes, not on the temporary storage associated with each EC2 instance.

If the EC2 instance fails and needs to be replaced, the Amazon EBS volume can simply be attached to the new EC2 instance. Because this new instance is a duplicate of the original, there is no loss of data or functionality.

EBS volumes are highly reliable, but to further mitigate the possibility of a failure, backups of these volumes can be created using a feature called snapshots. A robust backup strategy will include an interval between backups, a retention period, and a recovery plan. Snapshots are stored for high-durability in Amazon S3. Snapshots can be used to create new EBS volumes, which are an exact copy of the original volume at the time the snapshot was taken. These EBS operations can be performed through API calls.

Mission owners should know the following basic things about Amazon EBS functionality and features:

- Amazon EBS allows mission owners to create storage volumes from 1 GB to 16 TB that can be mounted as devices by EC2 instances. Multiple volumes can be mounted to the same instance.
- Storage volumes behave like raw, unformatted block devices, with user supplied device names and a block device interface. Mission owners can create a file system on top of EBS volumes or use them in any other way they would use a block device (like a hard drive).
- Amazon EBS volumes are placed in a specific Availability Zone and can then be attached to instances also in that same Availability Zone.
- Each storage volume is automatically replicated within the same Availability Zone. This prevents data loss due to failure of any single hardware component.
- Amazon EBS also provides the ability to create point-in-time snapshots of volumes, which are persisted to Amazon S3. These snapshots can be used as the starting point for new Amazon EBS volumes and protect data for long-term durability. The same snapshot can be used to instantiate as many volumes as desired.

For more information on these Amazon EBS features, refer to the [Amazon EBS documentation](#).

Instance storage

An instance store provides volatile, temporary block-level storage for use with an EC2 instance and consists of one or more instance store volumes. Instance store volumes must be configured using block device mapping at launch time and mounted on the running instance before they can be used. Instances launched from an instance store-backed AMI have a mounted instance store volume for the virtual machine's root device volume, and can have other mounted instances store volumes, depending on the instance type.

The data in an instance store is temporary and only persists during the lifetime of its associated instance. If an instance reboots (intentionally or unintentionally), data in the instance store persists. However, data on instance store volumes is lost under the following circumstances:

- Failure of an underlying drive
- Stopping an Amazon EBS-backed instance
- Ending an instance

Therefore, AWS mission owners should not rely on instance store volumes for important, long-term data. Instead, keep data safe by using a replication strategy across multiple instances storing data in Amazon S3, or using Amazon EBS volumes.

Encryption

AWS supports multiple encryption mechanisms for data stored within a mission owner's VPC.

The following is a summary of the encryption methods:

- **Amazon EBS encryption** — For Amazon EBS volumes, encryption is managed by OS-level encryption (for example, BitLocker or Encrypted File System (EFS)), by third-party products, or by Amazon EBS encryption. For Amazon EBS encryption, when customers create an encrypted Amazon EBS volume and attach it to a supported instance type, the data stored at rest on the volume, the disk I/O, and the snapshots created from the volume are all encrypted. The encryption occurs on the servers that host Amazon EC2 instances, providing encryption of data in transit from EC2 instances to Amazon EBS storage.
- **Amazon S3 encryption** — Provides added security for object data stored in buckets in Amazon S3. Mission owners can encrypt data on the client side and upload the encrypted data to Amazon S3. In this case, mission owners manage the encryption process, the encryption keys, and related tools. Optionally, mission owners can use the Amazon S3 server-side encryption feature. Amazon S3 encrypts object data before saving it on disks in its data centers, and it decrypts the object data when objects are downloaded, freeing mission owners from the tasks of managing encryption, encryption keys, and related tools. Mission owners can also use their own encryption keys with the Amazon S3 server-side encryption feature.
- **AWS Key Management Service (AWS KMS)** — [AWS KMS](#) is a managed service that makes it easy for mission owners to create and control the encryption keys used to encrypt your data. Learn more information about AWS KMS in the [Management](#) section of this paper.

- **AWS CloudHSM** — [AWS CloudHSM](#) is a cloud-based hardware security module (HSM) that allows you to easily add secure key storage and high-performance cryptographic operations to your AWS applications. CloudHSM has no upfront costs and proves the ability to start and stop HSMs on demand, allowing you to provision capacity when and where it is needed quickly and cost-effectively. CloudHSM is a managed service that automates the time-consuming administrative tasks, such as hardware provisioning, software patching, high availability and backups.

CloudHSM is one of several AWS services, including AWS KMS, which offers a high level of security for your cryptographic keys. AWS KMS provides an easy, cost-effective way to manage encryption keys on AWS that meets the security needs for the majority of customer data. CloudHSM offers customers the option of single-tenant access and control over their HSMs.

Management

AWS Identity and Access Management (IAM)

[IAM](#) is a web service that enables mission owners to manage users and permissions in AWS. The service is targeted at organizations with multiple users or systems that use products such as Amazon EC2, Amazon RDS, and the AWS Management Console. With IAM, mission owners can centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users can access.

Without IAM, organizations with multiple users and systems must either create multiple AWS accounts, each with its own billing and subscriptions to AWS products, or employees must all share the security credentials of a single AWS account. Also, without IAM, mission owners have no control over the tasks a particular user or system can do and what AWS resources they might use.

IAM addresses this issue by enabling organizations to create multiple users (each user is a person, system, or application) who can use AWS products, each with individual security credentials, all controlled by and billed to a single AWS account. With IAM, each user is allowed to do only what they need to do as part of the user's job.

IAM includes the following features:

- **Central control of users and security credentials**—Mission owners control creation, rotation, and revocation of each user's AWS security credentials (such as access keys).

- **Central control of user access**—Mission owners control what data users can access and how they access it.
- **Shared resources** – Users can share data for collaborative projects.
- **Permissions based on organizational groups** – Mission owners can restrict users' AWS access based on their job duties (for example, admin, developer, etc.) or departments. When users move inside the organization, mission owners can easily update their AWS access to reflect the change in their role.
- **Central control of AWS resources** – A mission owner's organization maintains central control of the data the users create, with no breaks in continuity or lost data as users move around within or leave the organization.
- **Control over resource creation** – Mission owners can help make sure that users create data only in sanctioned places.
- **Networking controls** – Mission owners can restrict user access to AWS resources to only from within the organization's corporate network, using SSL.

AWS Key Management Service (AWS KMS)

[AWS Key Management Service](#) allows mission owners to create and control encryption keys used to encrypt their data. It utilizes FIPS 140-2 validated cryptographic modules.

AWS KMS runs with other AWS services, like AWS CloudTrail to provide mission owners with logs of all key usage to help meet regulatory and compliance needs.

AWS KMS gives mission owners more control over the access to data that is encrypted. Mission owners have control over who can use the AWS KMS keys and gain access to encrypted data.

AWS KMS uses envelope encryption to protect data. Envelope encryption is the practice of encrypting plaintext data with a data key, and then encrypting the data key with another key. Envelope encryption allows several benefits.

For example, when rotating keys, instead of re-encrypting the raw data multiple times with different keys, mission owners can only re-encrypt the data keys that protect the raw data.

AWS KMS includes the following features:

- [AWS KMS keys](#) are the primary resource within AWS KMS. AWS KMS keys are used to generate, encrypt and decrypt data keys that are used outside of AWS KMS to encrypt data. AWS KMS stores, tracks and protects AWS KMS keys, and when an individual wants to use an AWS KMS key, the key is accessed through AWS KMS. An AWS KMS key never leaves AWS KMS unencrypted, nor does AWS KMS store, manage or track data keys.
- There are two types of AWS KMS keys within a mission owners AWS account:
 - **Customer-managed** AWS KMS keys, in which the mission owner creates, manages and uses the AWS KMS keys. In this case, the mission owner is responsible for enabling and disabling AWS KMS keys, establishing IAM and key policies to grant other permissions to use the keys.
 - **AWS managed** AWS KMS keys. In this case, keys are managed by the AWS service that runs with AWS KMS.
- Data keys are encryption keys for encrypting data, including large amounts of data. AWS KMS is used to generate, encrypt and decrypt data keys.
- Mission owners can import their own key material from their own infrastructure and use to encrypt their data. They can also use AWS KMS to manage the lifecycle of the key material.
- Key policies are used to control access to AWS KMS in AWS. Each AWS KMS has its own policy that has permissions and enables access to the key. For a user to access a resource, he or she must have access to the key and permissions to use the key.
- Mission owners can add an additional layer of security by limiting permissions to AWS KMS by using encryption context. The encryption context is another key-value pair of data that can be associated with the information protected by AWS KMS.
- AWS also offers an Encryption Software Development Kit (SDK), which is a library to implement encryption and follow best practices within an application.

AWS CloudTrail

[AWS CloudTrail](#) is a service that enables governance, compliance, operational and risk auditing of a mission owner's AWS account. CloudTrail nearly continuously logs and monitors actions taken by a user, role or another AWS service within an account.



CloudTrail monitors actions as an event in the AWS Management Console, AWS CLI, SDKs and APIs. Mission owners can use CloudTrail to view, search, download, archive, analyze and respond to account activity across the mission owner's AWS infrastructure. Mission owners have granularity to identify who or what took the action, what resources were acted upon, when the event occurred and other details.

CloudTrail includes the following features:

- Mission owners have the ability to aggregate all logs in Amazon S3 and restrict access to the Amazon S3 buckets to prevent tampering and deletion of log data.
- Mission owners can turn on CloudTrail in all AWS Regions, even if they aren't operating in other Regions. This way, suspicious activity in an account is always logged.
- CloudTrail can be enabled to audit usage of AWS KMS keys.

Amazon CloudWatch

[Amazon CloudWatch](#) is a monitoring service for AWS Cloud resources and applications running within AWS. CloudWatch is near real-time stream of system events and can be used to monitor for specific events and performs actions in an automated manner. Amazon CloudWatch is different from AWS CloudTrail, as the latter records API calls for an AWS account and deliver logs.

Amazon CloudWatch includes the following features:

- Mission owners can collect and track metrics, like CPU usage and disk reads/writes of Amazon EC2 instances or other Key Performance Indicators (KPIs).
- CloudWatch alarms send notifications or automatically make changes to resources that mission owners are monitoring based on the rules they have defined.
- Mission owners can create custom metrics to monitor application resources to gain visibility into resource utilization, application performance and operational health.

- The CloudWatch service also includes [CloudWatch Logs](#), which can be used to monitor, store, and access your log files from Amazon EC2 instances, AWS CloudTrail, Route 53, Lambda, and other sources. This can be used for log aggregation and consolidation, to support log reduction and auditing security operations functions.

AWS Config

[AWS Config](#) is a service that lets mission owners assess, audit and evaluate the configurations of all of their AWS resources. AWS Config monitors AWS resources and captures the configuration of these resources. It can automatically evaluate the configuration against desired configurations and helps simplify compliance auditing, security analysis, change management and operational troubleshooting. Mission owners can use AWS Config to keep an inventory of their AWS resources as well as software configurations within EC2 instances.

AWS Config includes the following features:

- Mission owners can keep track of all of their AWS resources and determine when a change to a certain resource has been made.
- AWS Config can be used to assess overall compliance. Mission owners can define rules for provisioning AWS resources, for example, only allow Amazon EBS volumes to be created if they are encrypted.
- Mission owners can also track the relationships among the resources and review dependencies prior to making changes.
- AWS Config can also capture a comprehensive history of AWS resource configuration. Mission owners can obtain the details of the event API call that invoked the change.
- AWS Config also allows viewing of compliance status across the enterprise, over multiple accounts and multiple Regions. This way, it is easier to identify non-compliant accounts or resources and view the data from the AWS Config console in a central account.

Services in scope

As stated previously, hosting a workload requires classification of data and determination of DoD SRG Impact Level of the data. The impact level may require the mission owner to choose AWS Regions carefully for their workloads.

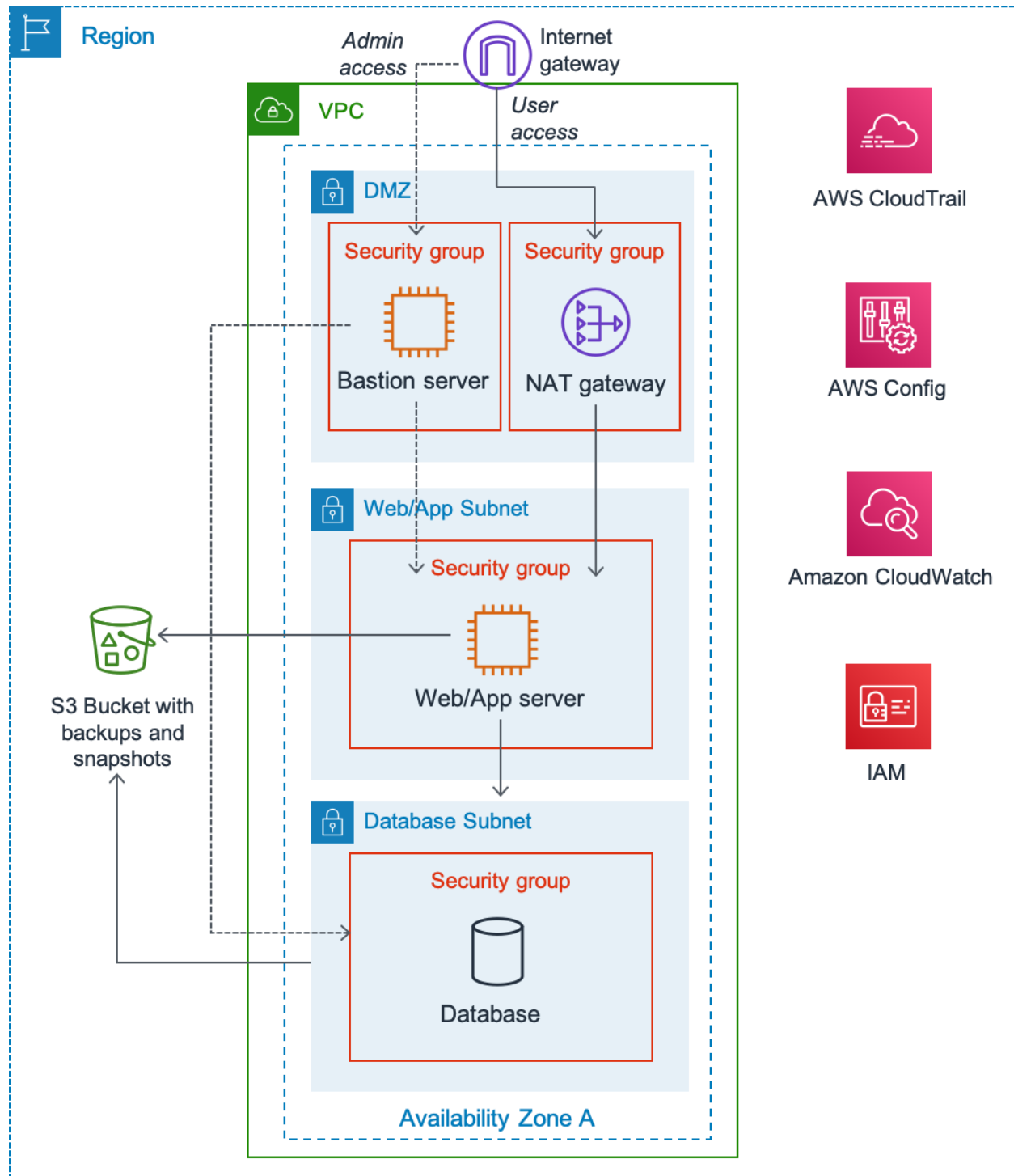
CONUS Regions (US-East/US-West) within the US have a provisional authorization to host IL2 data, whereas the AWS GovCloud (US) Regions may be used to host IL4 and IL5 data.

Within each Region, there are also a variety of services that mission owners can use that have gone through the DoD SRG accreditation process. AWS is constantly working with 3rd party auditors and with DoD accreditation agencies to get more services accredited at different impact levels. For an updated list of services that are currently undergoing or have already undergone various accreditation processes, refer to the [AWS Services in Scope page](#).

Reference architecture

Impact level 2

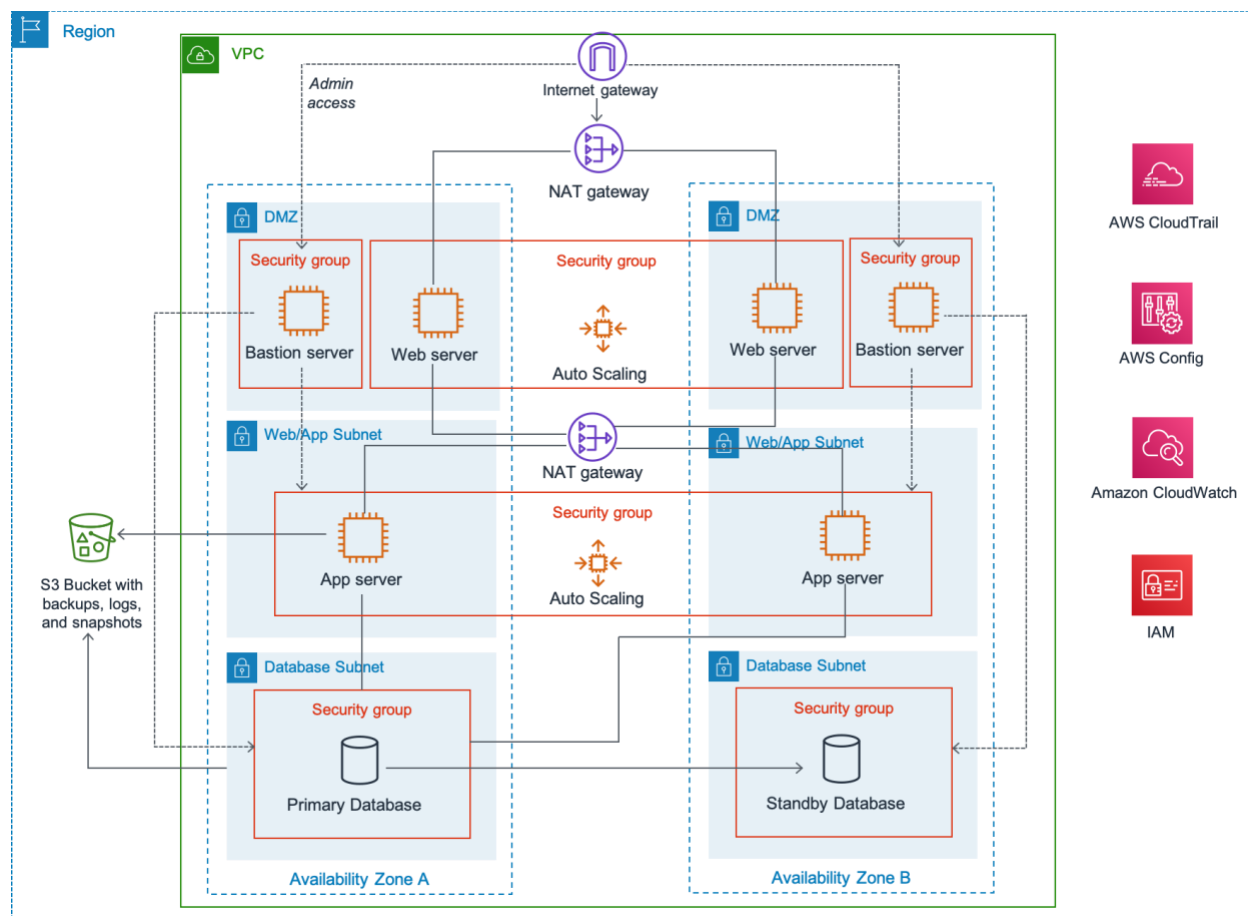
CC SRG Impact Level 2 (IL2) systems are appropriate for hosting public or limited access information. IL2 systems are not required to be fully segregated from internet traffic, and they can connect directly to the internet. The following is a sample reference architecture for an IL2 system with a recovery time objective (RTO) of greater than or equal to one day.



Sample Impact Level 2 Architecture with RTO $\geq 1\text{day(s)}$

The following is an IL2 sample reference architecture with a recovery time objective (RTO) of less than or equal to one hour.

The following reference architecture is an example of how to both meet application RTO requirements and maintain CC SRG compliance.



Sample Impact Level 2 Architecture with RTO \geq 1 hour

The following are some key attributes:

- Access to and from the internet traverses an internet gateway.
- A layer 7 reverse web proxy may reside in the DMZ for protection against application-level attacks targeting web infrastructures. Similarly, mission owners have the option of using native AWS services like AWS Web Application Firewall and AWS Shield, to protect against web-based attacks.
- Web and application instances are deployed in Auto Scaling groups across multiple Availability Zones.

- Each impact level 2 infrastructure should be adequately stratified to limit access to the web/application and database assets to either authorized traffic (by strata), or to administrative traffic initiated from an authorized bastion host contained within the infrastructure.
- Static web-addressable content is stored in secured Amazon S3 buckets (using bucket policies) and directly addressable from the internet.
- Infrastructure backups, images, and volume snapshots are securely stored in the Amazon S3 infrastructure in separate buckets, so they are not publicly addressable from the internet.
- Application database utilizes Amazon RDS, which is a managed offering for many flavors of commercial databases. The Amazon RDS instance is deployed in a Multi-AZ configuration with primary and secondary databases and synchronous replication between the two.

By default, the AWS infrastructure operates in a “zero trust” security model. Access to an instance, regardless of the strata on which it resides, must be implicitly allowed.

The enforcement of this model is enabled through the use of security groups (SG), which are addressable by other security groups.

For administrative access to any instance in the infrastructure, the use of a bastion host is defined as the only host instance that is authorized to access infrastructure assets within a designated infrastructure.

These hosts are typically, Windows Server instances (RDP via port 3389), Remote Desktop Gateway servers, and/or Linux instances for SSH access to Linux hosts. Any instance designated as a bastion host should be included in a bastion security group.

This should be the only security group granted access to the reverse web proxy, web/application instances, and database instances (via ports 22 and/or 3389).

Additionally, to further bolster the defensive posture of the infrastructure, the bastion host(s) should be powered off when administration activities are not being performed.

The following table is a sample summary of security group behavior by traffic flow:

Table 3 — Security group behavior by traffic flow

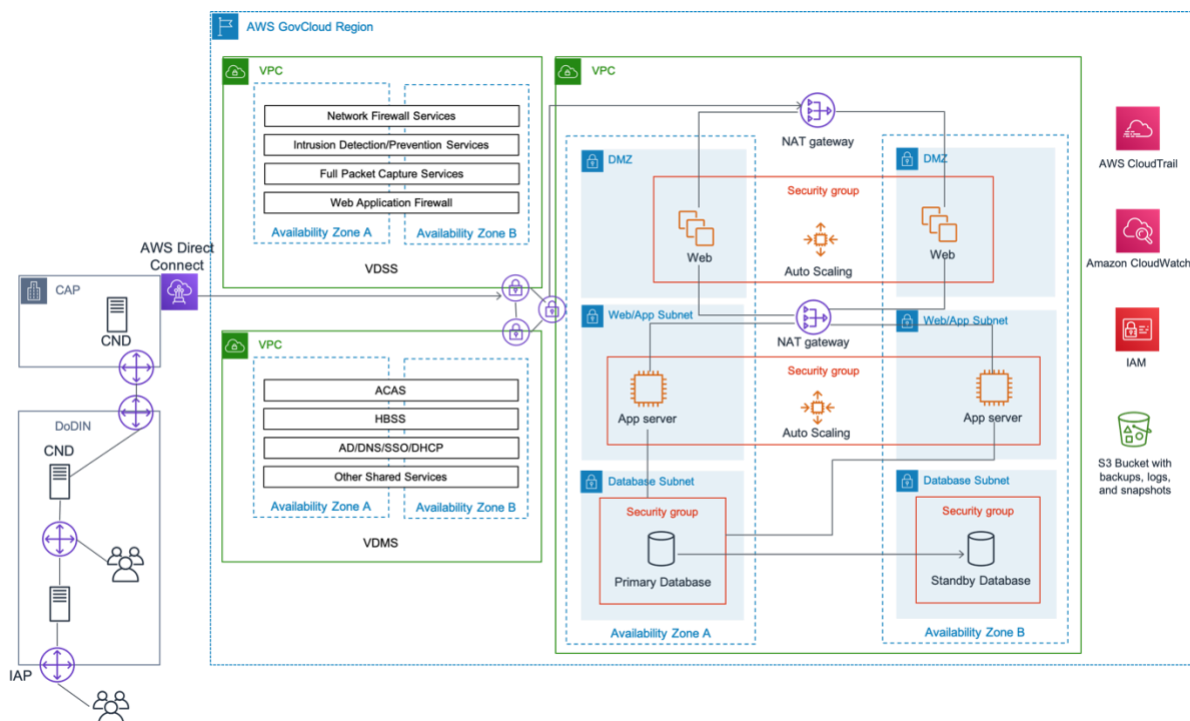
Traffic From Security Group (SG)	Traffic to SG	Security Group Rule
Internet	Reverse web proxy (<i>reverse-proxy-SG</i>)	Allow 80/443 from internet (all)
Reverse web proxy (<i>reverse-proxy-SG</i>)	Web/application server(s) (<i>web-server-SG</i>)	Allow 80/443 from <i>reverse-proxy-SG</i>
Web/application server(s) (<i>web-server-SG</i>)	Database server(s) (<i>db-server-SG</i>)	Allow appropriate database port
Administrator (internet, trusted admin IP)	Bastion host (<i>bastion-host-SG</i>)	Allow 3389/22 from trusted remote administration host (host IP address range)
Bastion host (<i>bastion-host-SG</i>)	Proxy, web, application, database instances (<i>reverse-proxy-SG</i> , <i>web-server-SG</i> , <i>db-server-SG</i>)	Allow 3389/22 from <i>bastion-host-SG</i>

Impact Level 4

DoD systems hosting data categorized at IL4 and IL5 of the CC SRG must attain complete separation from systems hosting non-DoD data, and route traffic entirely through dedicated connections to the DoD information networks (DoDIN) through a VPN or an AWS Direct Connect.

To achieve full separation of network traffic, the current approved DoD reference architecture is to establish an AWS Direct Connect connection from DoDIN to AWS, including BCAP with a Computer Network Defense (CND) Suite hosted in a colocation facility associated with AWS.

The following illustration is a sample reference architecture for an IL4 system. The following architecture utilizes best practices according to the DoD SRG, which creates guidelines for a Secure Cloud Computing Architecture (SCCA). In addition, AWS has published additional [SCCA reference architecture](#) guidance.



Sample Impact Level 4 architecture

The following list contains the CC SRG requirements for IL4 that are added to those already defined for IL2:

- No direct access to/from the public internet – All traffic in/out of AWS must traverse the DoDIN through a virtual private gateway.
- Security and management of the environment is separated from the application environment using different VPCs in accordance with the SCCA architecture.
- A Virtual Data Center Security Stack (VDSS) VPC is utilized for performing security functionality in accordance with the SCCA and all traffic will flow through the VDSS VPC before reaching the Mission Owner VPC, which contains the application. The VDSS may contain approved third-party security components to meet the security requirements of the mission owner (for example, performing full packet capture or adding intrusion detection or prevention services).
- A Virtual Data Center Management Stack (VDMS) VPC is established for performing management functionality and offering shared services. This VPC may host shared services for hosting multiple mission owner application VPCs. The VDMS may also perform host management via bastion hosts, security scans and other services deemed necessary by the mission owner.

- Connection to the DoDIN – This can be accomplished through the use of AWS Direct Connect, IPsec VPN, and/or a combination of the two. All traffic traversing between DoDIN and the DoD application must use a BCAP.
- Access to Amazon S3 is restricted to AWS Direct Connect – Access to Amazon S3 is, while internet addressable by default, only accessible through a private route introduced as part of the AWS Direct Connect service.
- All traffic to/from the VPC is scanned on DoDIN – All traffic entering and/or exiting the Amazon VPC is required to pass through a hardware-based Computer Network Defense suite of tools. This infrastructure is both owned and operated by the government (or on behalf of the government by a Mission Partner organization).
- Host-Based Security System (HBSS) servers are deployed in the VDMS VPC – All DoD EC2 instances will have HBSS installed, and they will communicate with an orchestrator server hosted in the VDMS VPC.
- Assured Compliance Assessment Solution (ACAS) tool is deployed in the VDMS VPC – All DoD instances will be scanned by an ACAS tool that is located in the VDMS, with full access to the subnets of the mission owner VPC.

Impact Level 5

Data that is classified at IL5 has additional controls that must be placed on top of impact level 4 controls. One of the controls required for IL5 is that all data must be encrypted in flight and at rest. Any components of the architecture that process IL5 data requires physical separation from unencrypted data. Within AWS, all IL5 workloads must be deployed in the AWS GovCloud (US) Region within an Amazon VPC, and network traffic must flow through an approved CAP or DoD SCCA compliant solution.

As previously stated, all data must be encrypted in flight and at rest. Decryption of data at certain points of the traffic flow (for example, decrypting to perform compute operations) requires an Amazon EC2 Dedicated Host or Dedicated Instance to meet the requirements for physical separation.

The AWS services that require dedicated tenancy are EC2, Elastic MapReduce, Elastic Beanstalk, Amazon WorkSpaces, Elastic Kubernetes Service without Fargate, and Elastic Container Service without AWS Fargate. If architecting a three-tier web application, like in the examples used so far, all three tiers of the application's compute must use Dedicated Hosts or Instances.

It is also possible to have the web tier instances as On-Demand Instances (IL4) if the web servers are only passing encrypted traffic. The application and database tiers will always require Dedicated Instances or Hosts.

Multi-tenant

By default, EC2 instances are multi-tenant. This means that the mission owner pays for the compute capacity by the hour or second. Mission owners can increase or decrease their capacity based on demand.

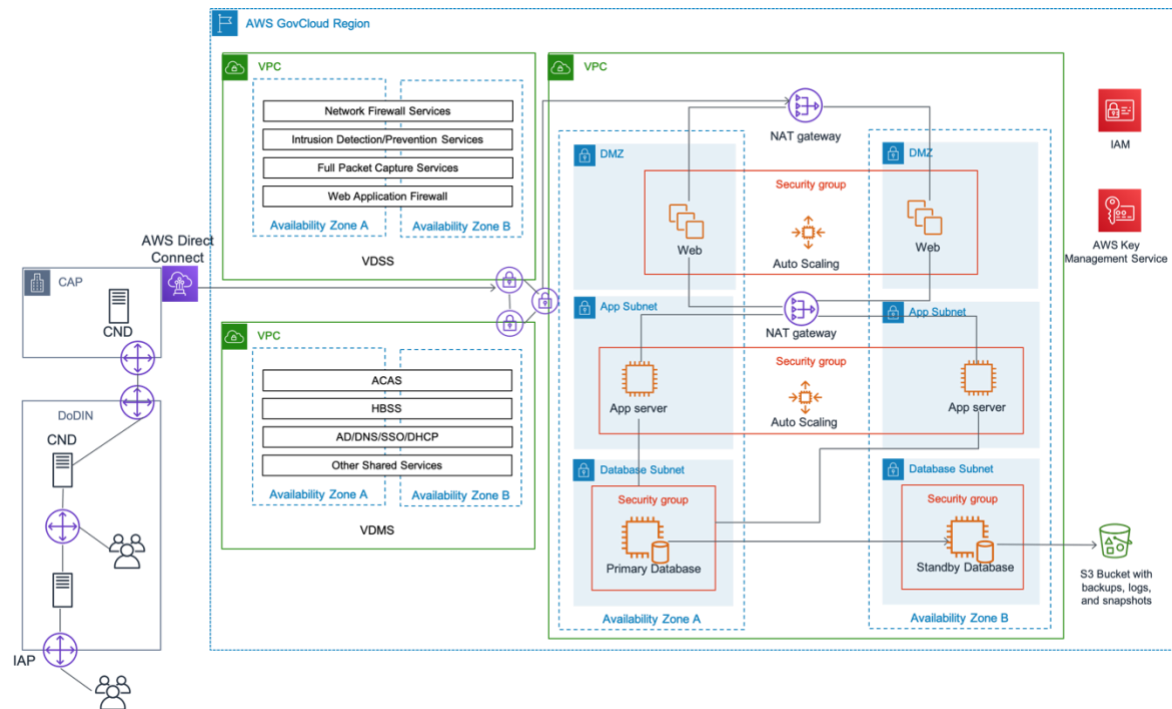
Dedicated Instances

Dedicated Instances are EC2 instances that run inside a VPC on hardware that is dedicated to a single customer. The Dedicated Instances of a mission owner 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. More information can be found on the [Dedicated Instances pricing page](#).

Dedicated Hosts

A Dedicated Host is a physical server with Amazon EC2 instance capacity fully reserved for one AWS account. Dedicated Hosts are designed to meet compliance requirements and allow mission owners to utilize their server-bound software licenses.

The following diagram is an example of an IL5 architecture hosted in AWS.



Sample Impact Level 5 architecture hosted in AWS

The following are some key attributes:

- All EC2 instances must use Dedicated Instances or Dedicated Hosts if handling unencrypted data
- Mission Owners can use AWS KMS for managing their encryption keys, or they may bring their own encryption keys. Key policies must be utilized to control and grant other resources or individual access to encryption keys.
- This architecture also utilizes the SCCA guidelines by incorporating a VDSS and VDMS, like in the IL4 environment.

Conclusion

AWS provides a number of important benefits to DoD mission owners, including flexibility, elasticity, utility billing, and reduced time-to-market. It provides a range of security services and features that you can use to manage the security of your assets and data in AWS.

Although AWS provides an excellent service management layer for infrastructure or platform services, mission owners are still responsible for protecting the confidentiality, integrity, and availability of their data in the cloud, and for meeting specific mission requirements for information protection.

Conventional security and compliance concepts still apply in the cloud. Using the various best practices highlighted in this whitepaper, we encourage you to build a set of security policies and processes for your organization so you can deploy applications and data.

Contributors

The following individuals contributed to this document:

- Paul Bockelman, Lead Architect, AWS Worldwide Public Sector
- Andrew McDermott, Solutions Architect
- Nabil Merchant, Security Consultant, AWS Worldwide Public Sector
- Jim Collins, Principal Consultant, AWS Professional Services
- Michael Alpaugh, Senior Security Architect, AWS Worldwide Public Sector

Further reading

For additional information, refer to the following:

- [AWS Whitepapers](#)
- [AWS Documentation](#)

Document revisions

Date	Description
November 3, 2021	Major structural update and additional content; updated diagrams; compliance updates.
April 2018	Updated diagrams. IL5 reference architecture section added. Added description of additional services
April 2015	First publication

Notes

- ¹ [Department of Defense Cloud Computing Security Requirements Guide](#)
- ² [Using AWS for Disaster Recovery](#)
- ³ [FedRAMP: About Us](#)