Encryption & Data Protection

AWS Security Workshop

Agenda

- Encryption at rest
- Encryption in transit
- Data protection considerations

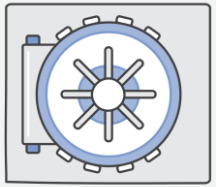
Goals

- Understand customer responsibility for data in AWS
- Learn how encryption is done in AWS
- Consider your own encryption requirements
- Discover data protection related AWS services

It is always YOUR data!

- Customers choose **where to place their data**
- AWS regions are geographically **isolated by design**
- **Data is not replicated to other AWS regions** and does not move unless the customer tell us to do so
- Customer always **own their data, the ability to encrypt it, move it, and delete it**

AWS Customer Agreement
https://aws.amazon.com/agreement/



Data Protection In-Transit and At-Rest

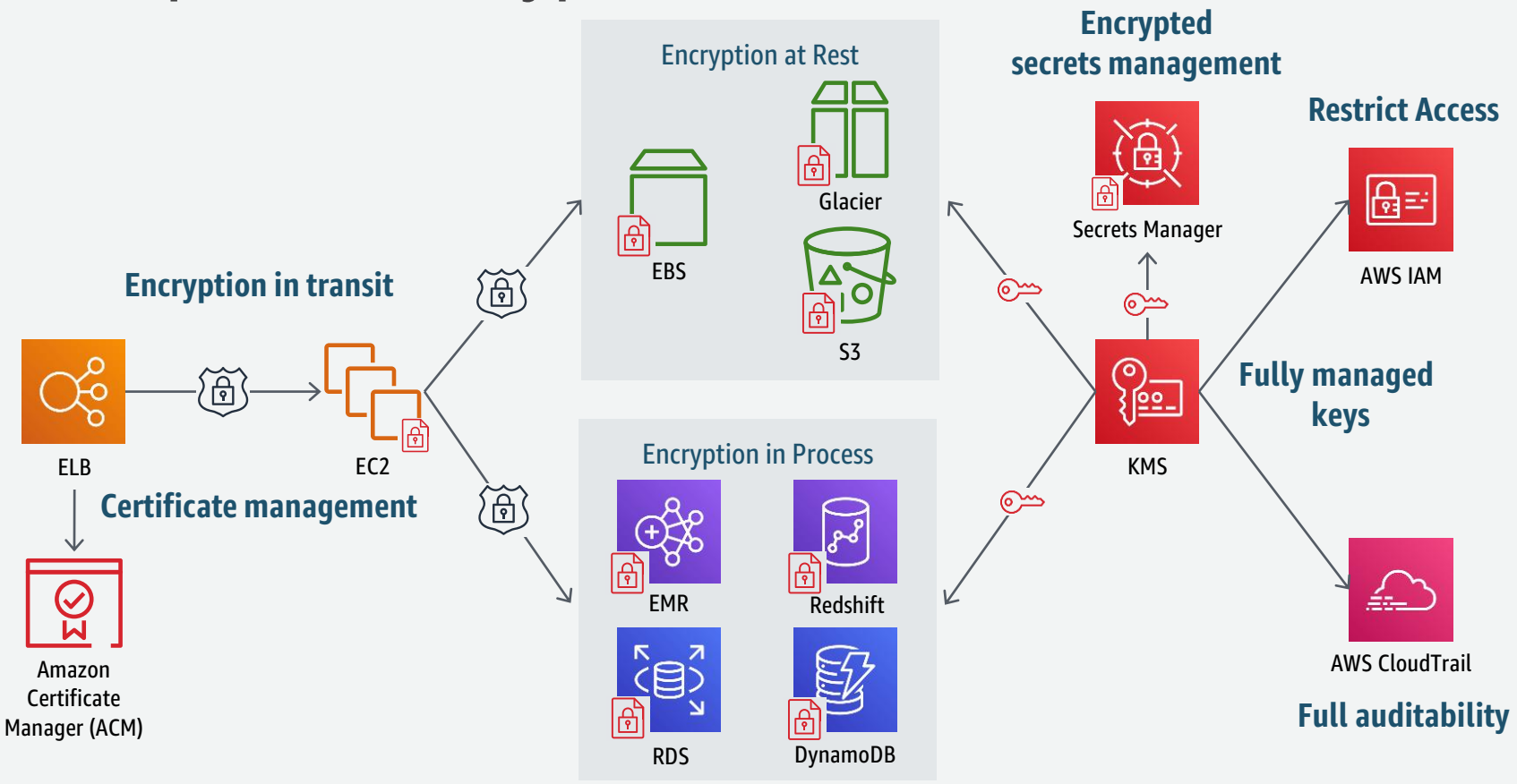
Encryption In-Transit

- SSL/TLS
- SSH
- VPN/IPSEC

Encryption At-Rest

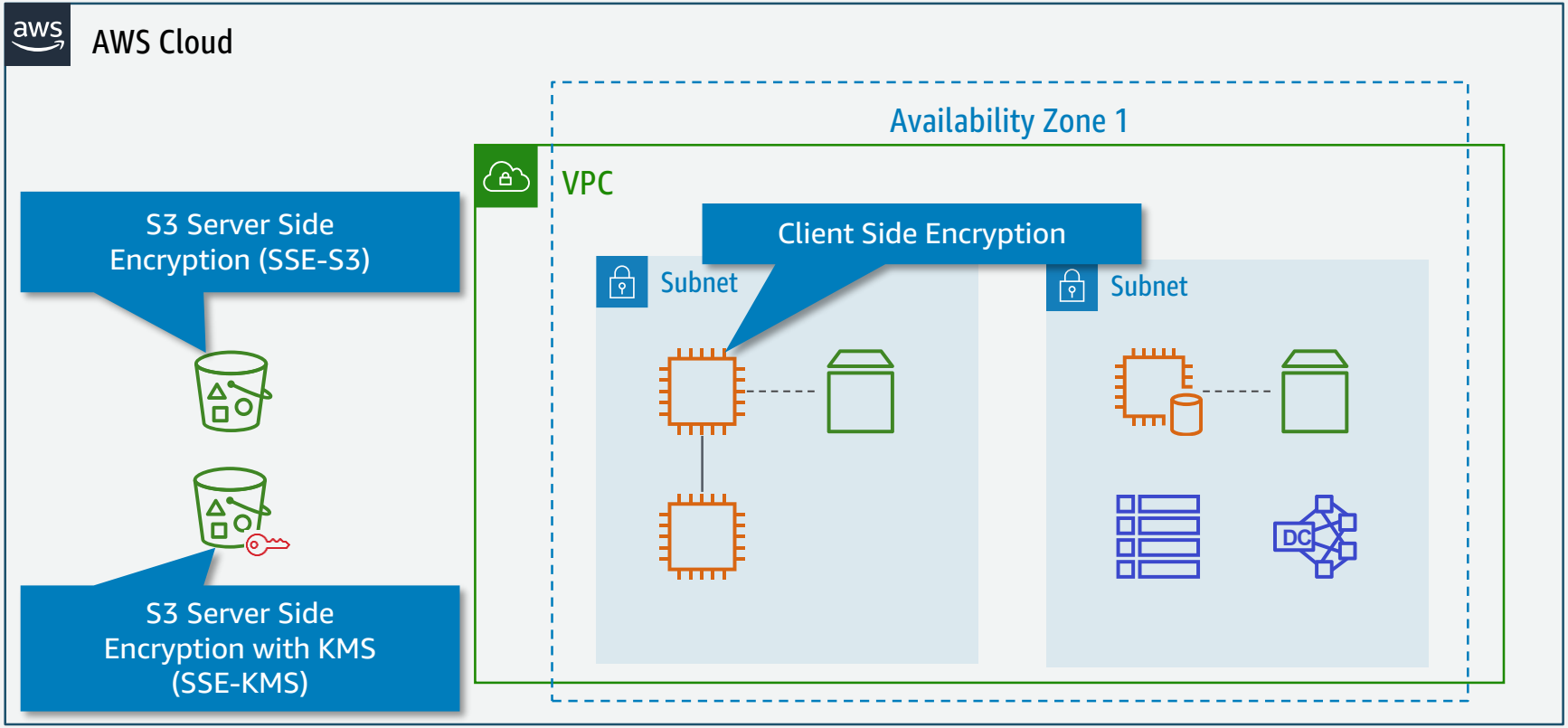
- Object
- Database
- Filesystem
- Disk

Ubiquitous Encryption

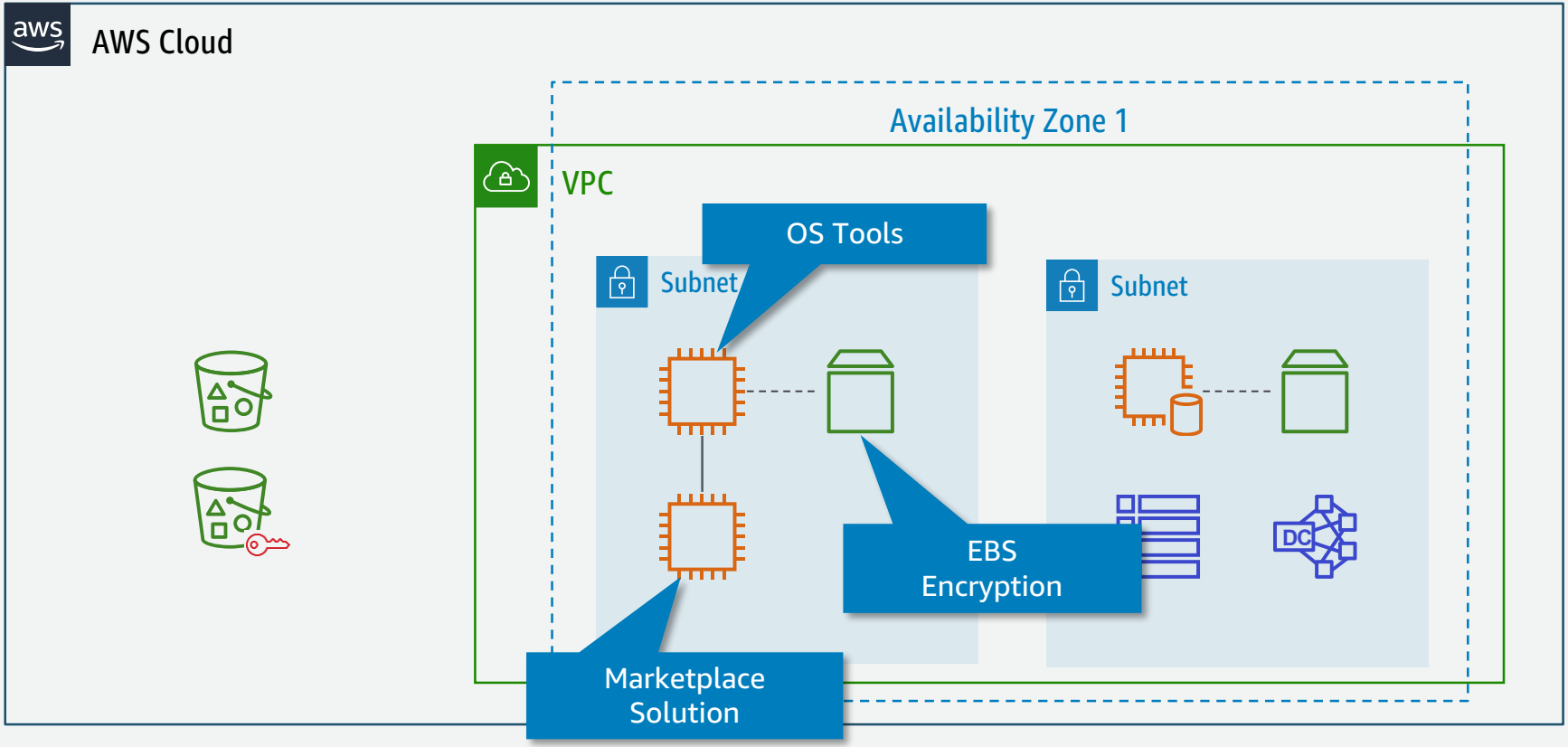


Encryption at Rest

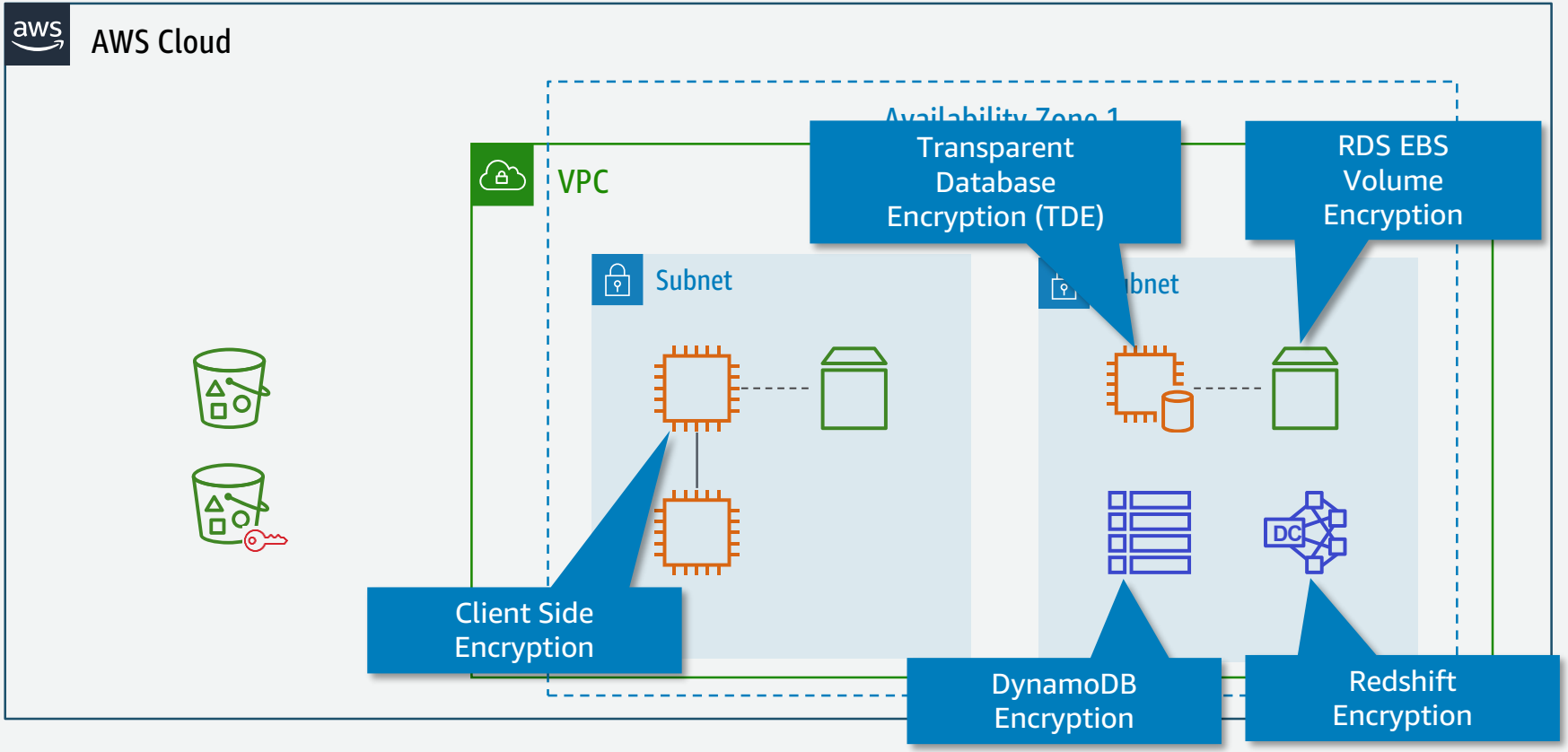
Encryption at Rest – S3



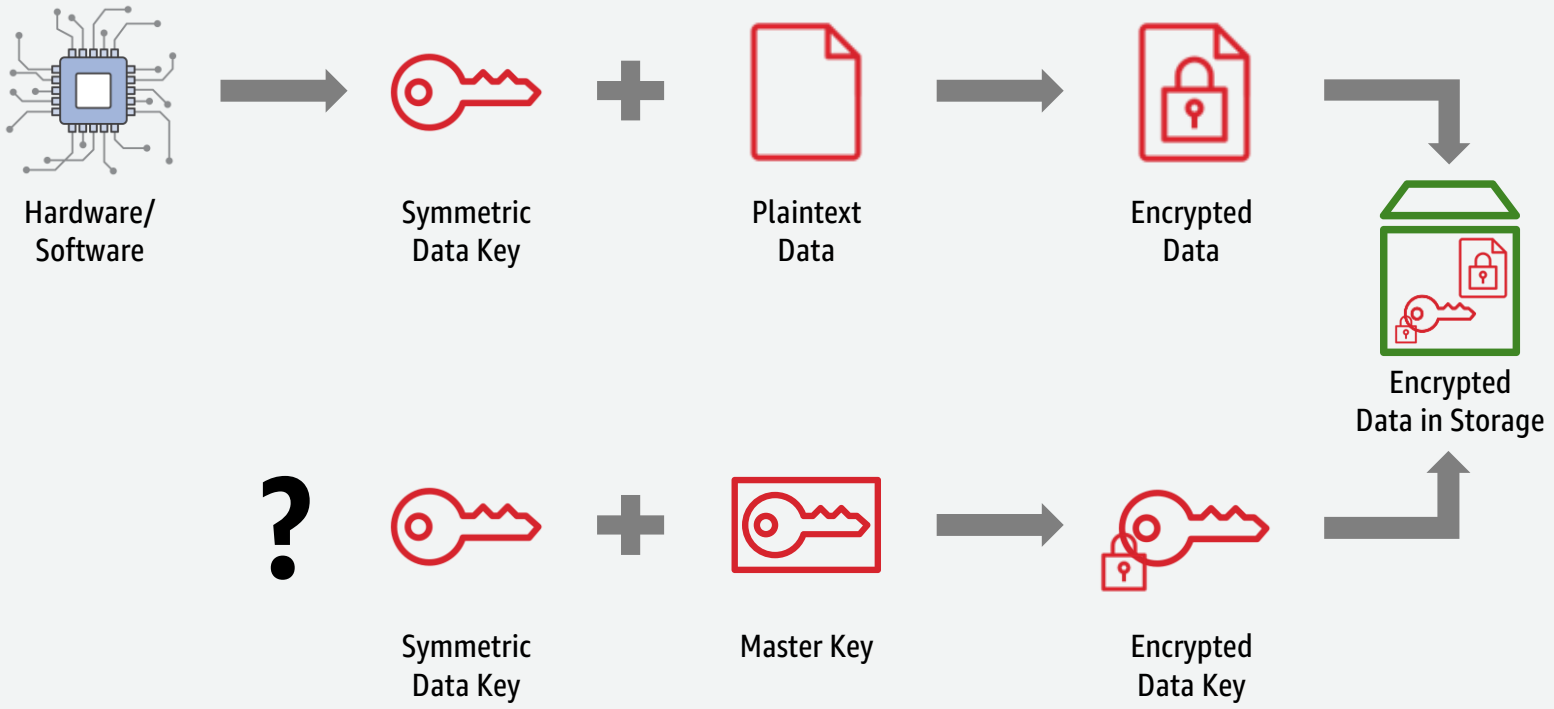
Encryption at Rest – EBS



Encryption at Rest – Databases



Encryption at Rest – Envelope Encryption Primer



Plain text keys need to exist somewhere

Encryption at Rest – Key Considerations

- Where are keys stored?
  - Hardware you own?
  - Hardware the cloud provider owns?
- Where are keys used?
  - Client software you control?
  - Server software the cloud provider controls?
- Who can use the keys?
  - Users and applications that have permissions?
  - Cloud provider applications you give permissions?
- What assurances are there for proper security around keys?

Encryption at Rest – Option in AWS

Client-side encryption

- You encrypt your data before data submitted to the service
- You supply encryption keys OR use keys in your AWS account
- Available clients:
- S3, EMR File System (EMRFS), DynamoDB, AWS Encryption SDK

Server-side encryption

- AWS encrypts data on your behalf after data is received by service
- Services with integrated encryption include S3, Snowball, EBS, RDS, Amazon Redshift, WorkSpaces, Amazon Kinesis Firehose, CloudTrail, EMR, DynamoDB, CodePipeline, AWS Secrets Manager, AWS Backup

Encryption at Rest – AWS Key Management Service

- Managed service that simplifies creation, control, rotation, deletion, and use of encryption keys in your applications
- FIPS 140-2 validated hardware security modules (HSM) and support for FIPS 140-2 validated endpoints
- Integrated with over 50 AWS services for server-side encryption
- Integrated with AWS service clients/SDKs
- S3, EMRFS, DynamoDB, AWS Encryption SDK
- Integrated with CloudTrail to provide auditable logs of key usage for regulatory and compliance activities
- Available in all commercial regions except China

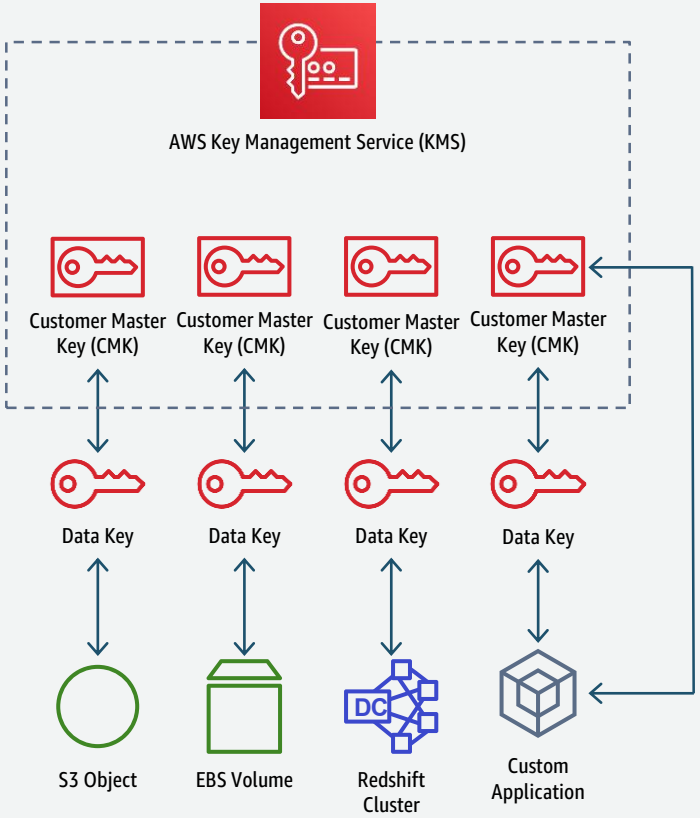
Encryption at Rest – AWS Key Management Service

AWS Key Management Service Hierarchy

- Two-tiered key hierarchy using envelope encryption
- Unique data key encrypts customer data
- KMS master keys encrypt data keys
- KMS master keys never leave the KMS HSM unencrypted

Benefits

- Limits risk of compromised data key
- Better performance for encrypting large data
- Easier to manage small number of master keys than millions of data keys
- Centralized access and audit of key activity



Encryption at Rest – AWS Key Management Service

Auditing key usage with AWS CloudTrail

```
"EventName":"DecryptResult",
```

This KMS API action was called …

```
"EventTime":"2014-08-18T18:13:07Z",
```

… at this time

```
"RequestParameters":
    "{"keyId":"2b42x363-1911-4e3a-8321-6b67329025ex”}”,
```

… in reference to this key

```
"EncryptionContext":"volumeid-12345",
```

… to protect this AWS resource

```
"SourceIPAddress":" 203.0.113.113",
```

… from this IP address

```
"UserIdentity":
    "{"arn":"arn:aws:iam:: 111122223333:user/User123"}
```

… by this AWS user in this account

Encryption at Rest – AWS Key Management Service

Bring Your Own Key Material to KMS

- You control how master keys are generated
- You store the master copy of the keys
- You import the key into KMS as key material and set an optional expiration time in the future
- Generate CMKs based on the imported key material
- You can use imported key material with all KMS-integrated services
- You can delete and re-import the key material at any time to control when AWS can use it to encrypt/decrypt data on your behalf
- Works with standards-based key management infrastructure, such as Thales e-Security

Encryption at Rest – Bring Your Own Key Material

Create customer master key (CMK) container

↓

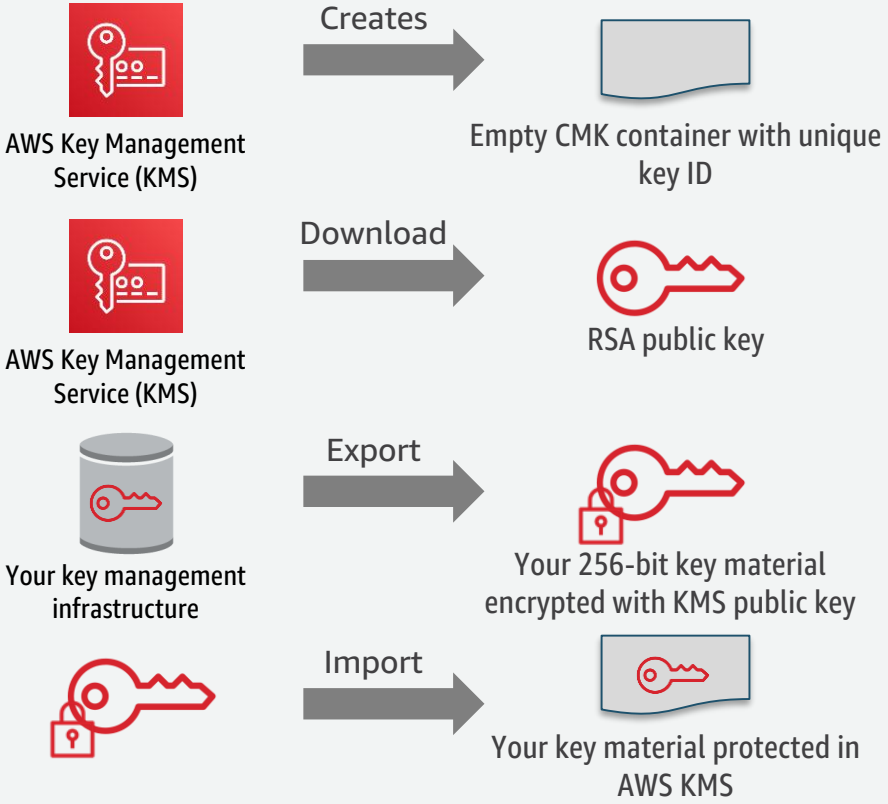
Download a public wrapping key

↓

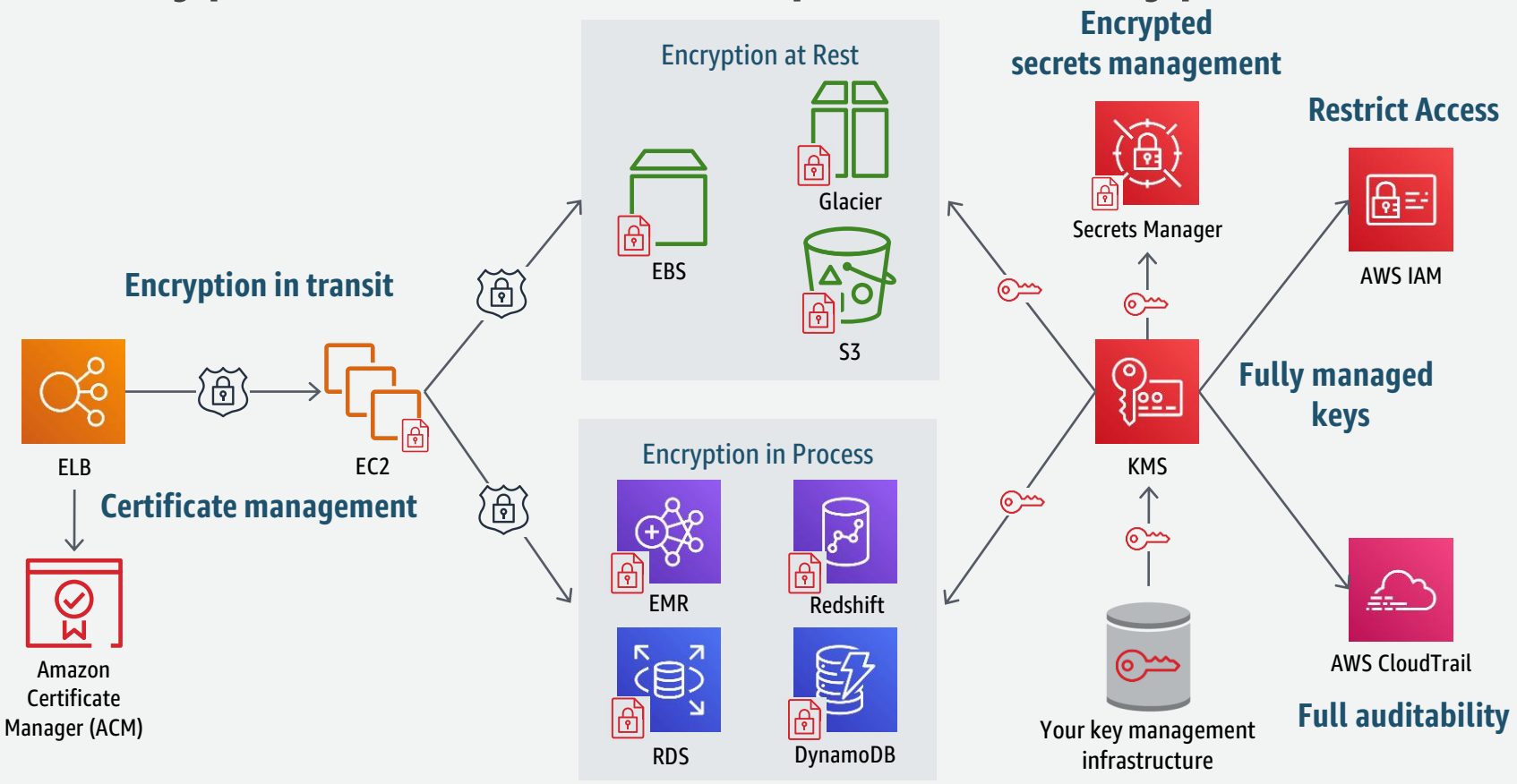
Export your key material encrypted under the public wrapping key

↓

Import the encrypted key material under the KMS CMK key ID; set optional expiration period



Encryption at Rest – Ubiquitous Encryption

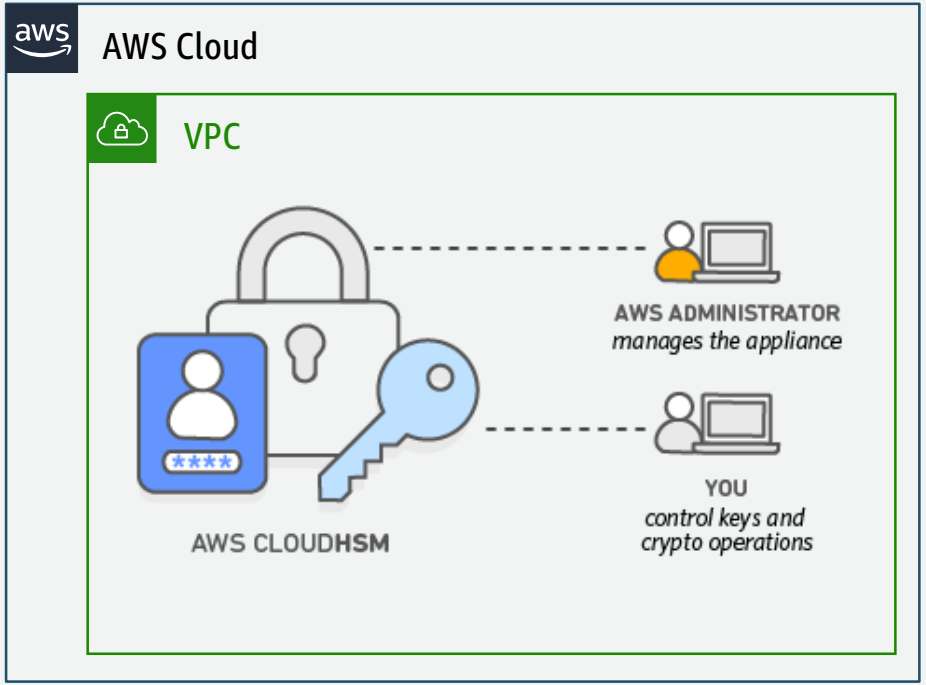


Encryption at Rest – KMS CMK Types

	AWS Owned CMK	AWS Managed CMK	Customer Managed CMK
Creation	AWS generated	AWS generated on customer's behalf	Customer generated
Rotation	Once every three years automatically	Once every three years automatically	Once a year automatically through opt-in or manually on-demand
Deletion	Can't be deleted	Can't be deleted	Can be deleted
Visible within your AWS account	No	Yes	Yes
Scope of Use	Not limited to your AWS account	Limited to a specific AWS service within your AWS account	Controlled via KMS/IAM policies

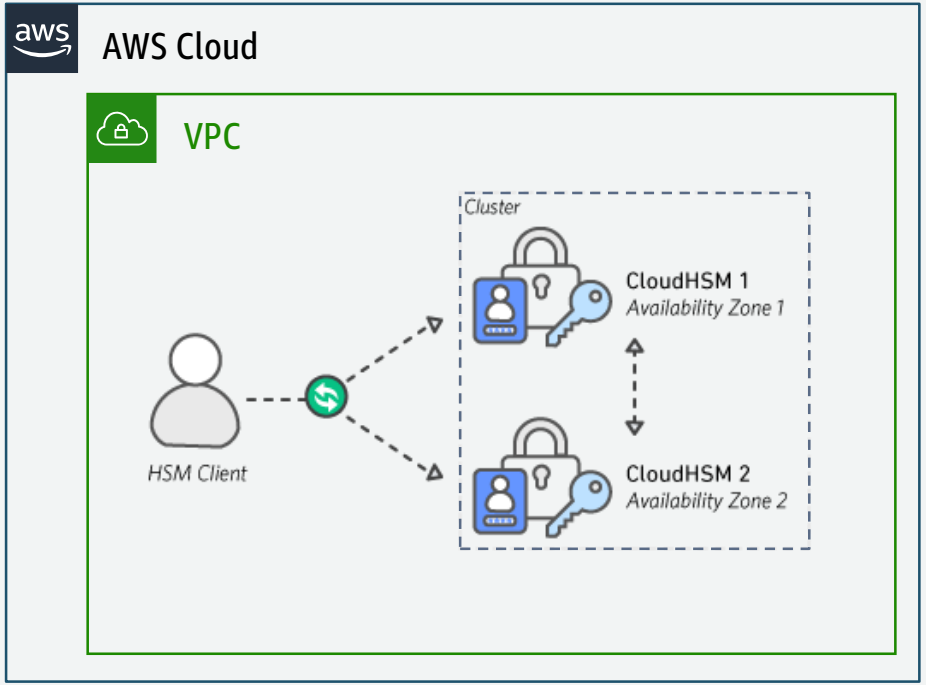
Encryption at Rest – CloudHSM

- **Dedicated access** to HSM appliances
- HSMs located in AWS data centers
- Managed and monitored by AWS
- **Only you have access to your keys and operations on the keys**
- HSMs are inside your Amazon VPC, isolated from the rest of the network
- **FIPS 140-2 level 3 certified**



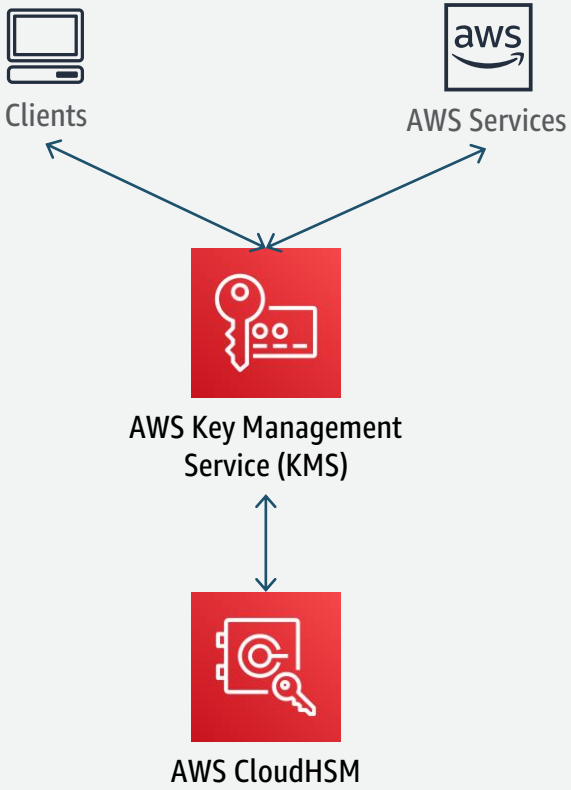
Encryption at Rest – CloudHSM

- Setup from the AWS Management Console or CLI
- Load balanced & synchronized
- Clusters can scale to meet demand
- Industry standard API's available for developers to get started
- MFA authentication available
- Capability of snapshotting CloudHSM clusters

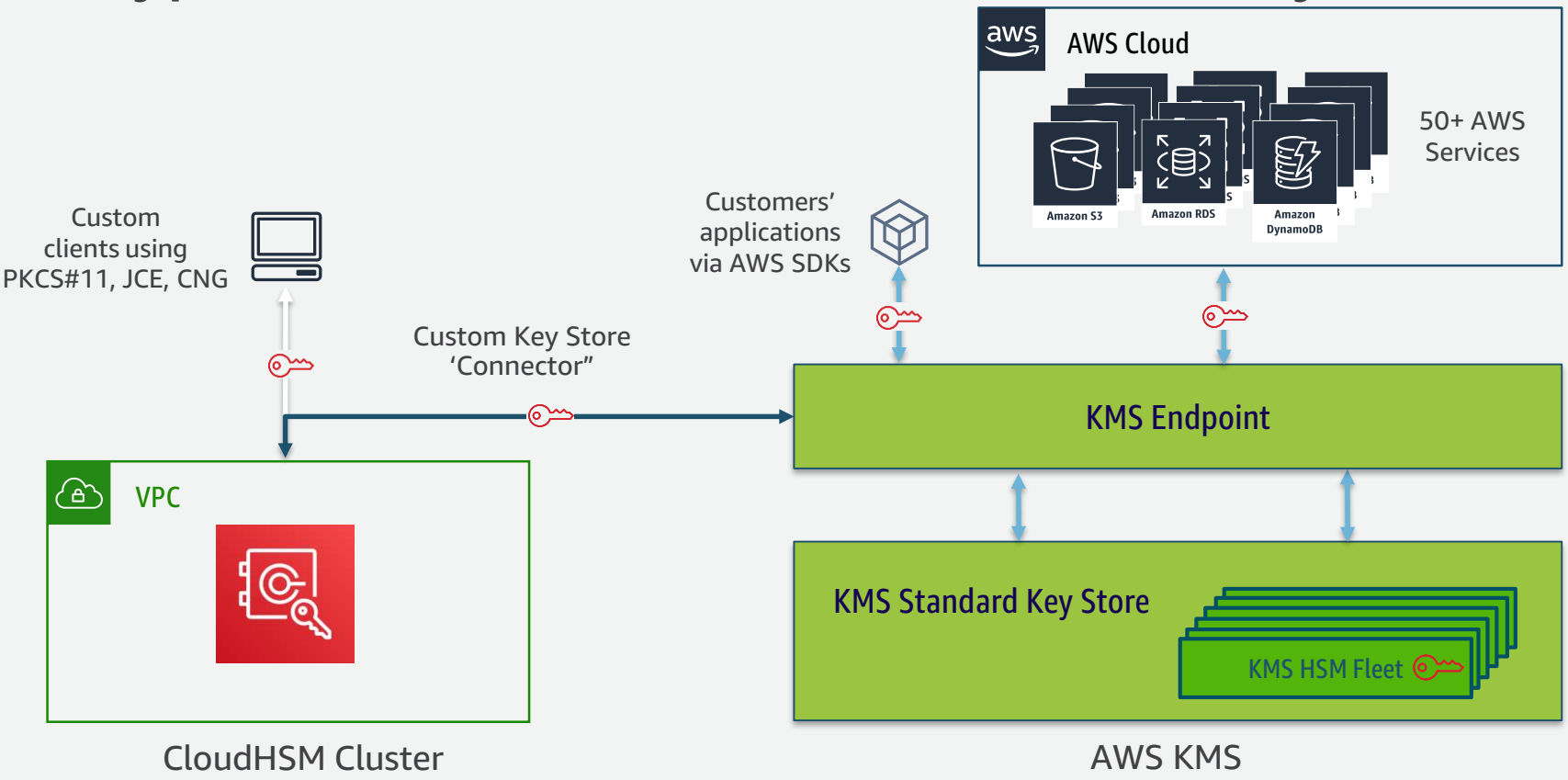


Encryption at Rest – AWS KMS Custom Key Store

- Use CloudHSM as the key store for KMS
- Combining the key management capabilities of KMS with the key storage capabilities of CloudHSM
- Use the standard KMS API's and native service integration offered by KMS



Encryption at Rest – AWS KMS Custom Key Store



Encryption at Rest – AWS KMS Custom Key Store

When to use AWS KMS Custom Key Store?

You have keys that are required to be:

- protected in a single-tenant HSM or in an HSM over which you have direct control
- stored in an HSM validated at FIPS 140-2 level 3 overall
- replicated across multiple AWS regions

Encryption at Rest – AWS KMS vs CloudHSM

AWS CloudHSM

- Dedicated access to HSM that complies with government standards (e.g. FIPS 140-2 Level 3, Common Criteria)
- High-performance in-VPC cryptographic acceleration
- You control your keys and the application software that uses them
- Supported applications:
  - Your custom software
  - Third party software
  - Symmetric or asymmetric encryption

AWS Key Management Service

- Highly available and durable key storage, management, and auditable solution (FIPS 140-2 Level 2 HSM's and support for FIPS 140-2 Level 2 endpoints)
- Easily encrypt your data across AWS services and within your own applications based on policies you define
- Supported applications:
  - Your custom software (AWS SDK)
  - Symmetric encryption
  - Integrated with multiple AWS services

Encryption at Rest – APN Partner Solutions

- You can browse, test, and buy encryption and key management solutions via the AWS Marketplace
- Pricing models vary: pay-by-the-hour, monthly, or annual
- The software fees are simply added to your AWS bill
- Some solutions offer a bring-your-own-license option

gemalto security to be free

HYTRUST

THALES e-Security

Symantec.

KeyNexus

Townsend SECURITY

Encryption at Rest – Solution Comparison

	AWS KMS	AWS KMS with Custom Key Store	AWS CloudHSM	AWS Marketplace Partner Solution	DIY
Where keys are generated and stored	AWS KMS FIPS 140-2 Level 2 HSM's (with level 3 for several other categories)	AWS CloudHSM FIPS 140-2 Level 3 HSM's	AWS CloudHSM FIPS 140-2 Level 3 HSM's	Your network or EC2 instance	Your network or EC2 instance
Where keys are used	AWS services or your applications using the AWS SDK's	AWS services or your applications using the AWS SDK's	AWS or your applications using the HSM specific SDK	Your network or EC2 instance	Your network or EC2 instance
How to control key usage	Policies you define; enforced by AWS	Policies you define; enforced by AWS – *Only for keys made available through KMS*	HSM-specific access controls	Vendor-specific access controls	You implement access controls
Responsibility for performance/scale	AWS	AWS (API's) Customer (Key Store)	Customer	Customer	Customer
Integration with AWS services?	Yes	Yes	Limited	Limited	Limited

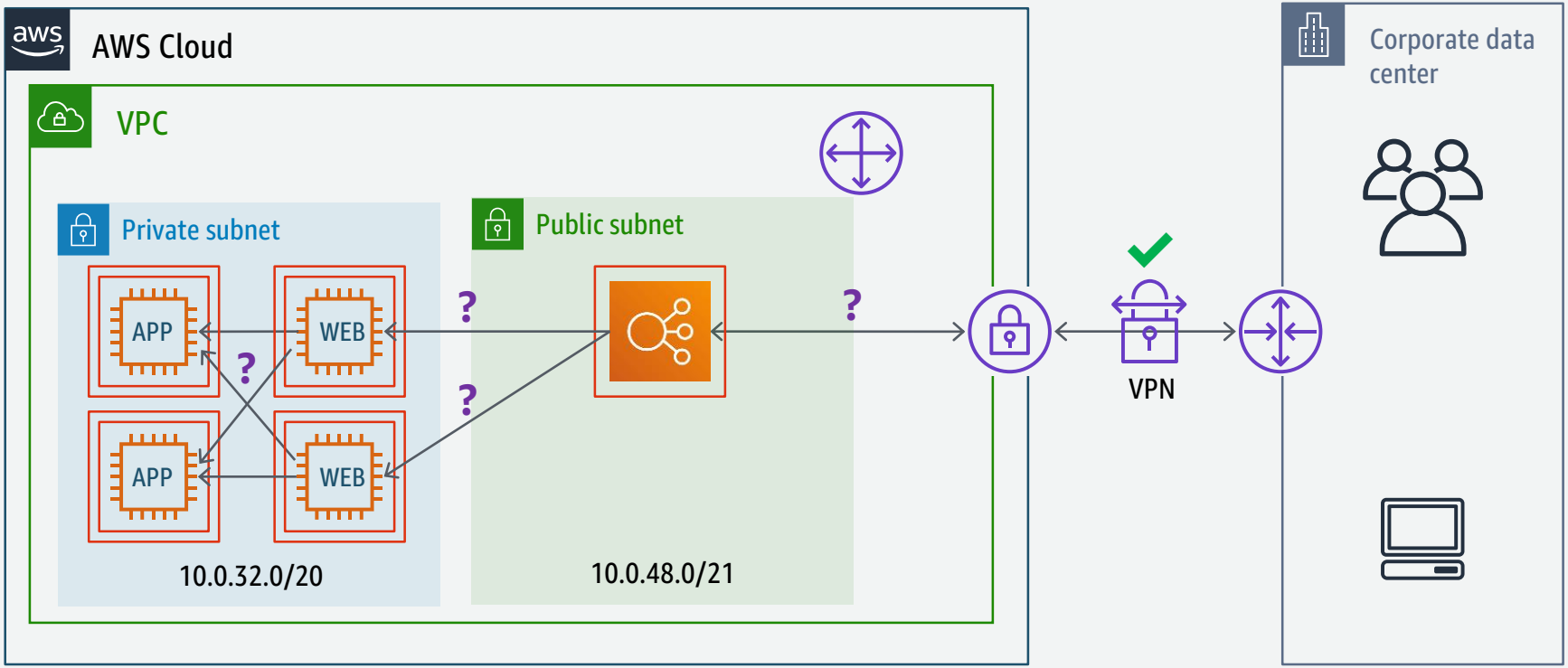
Encryption in Transit

Encryption in Transit – Inside the VPC

What is VPC (review)?

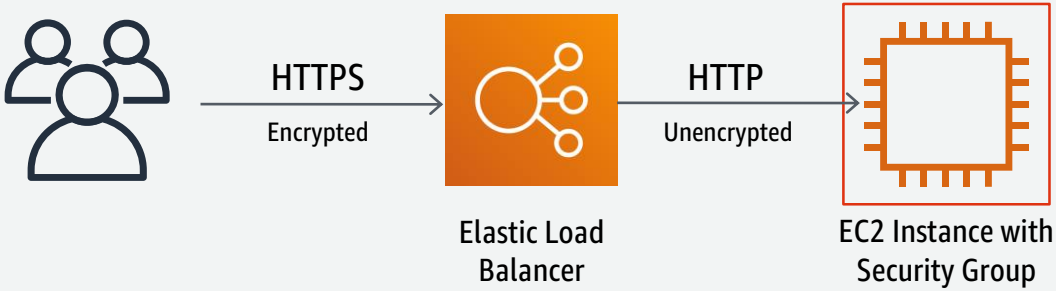
- Virtual **Private** Cloud
- **Logically isolated portion** of the AWS infrastructure
- Allows you to extend your existing data center network to the Cloud
- Can be **considered as private network** by PCI compliance
- **Audited & Certified** on SOC1/2, ISO27001, FedRAMP, HIPAA BAA, PCI
- Protected against most of L2/L3 attacks (multicast, IP/MAC/ARP spoofing, sniffing)

Encryption in Transit – Inside the VPC

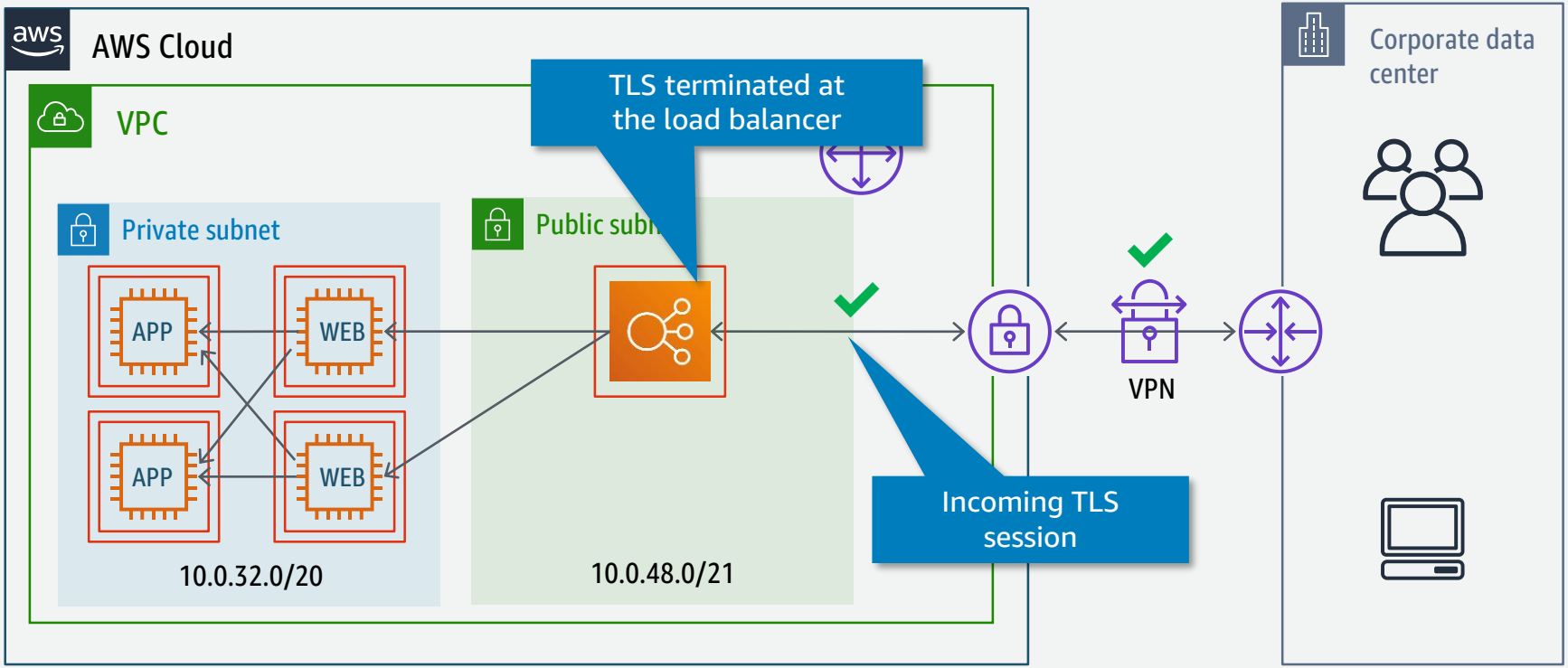


Encryption in Transit – TLS with Amazon ELB

You can use the ELB for HTTPS termination with unencrypted communication to back-end instances on port 80.

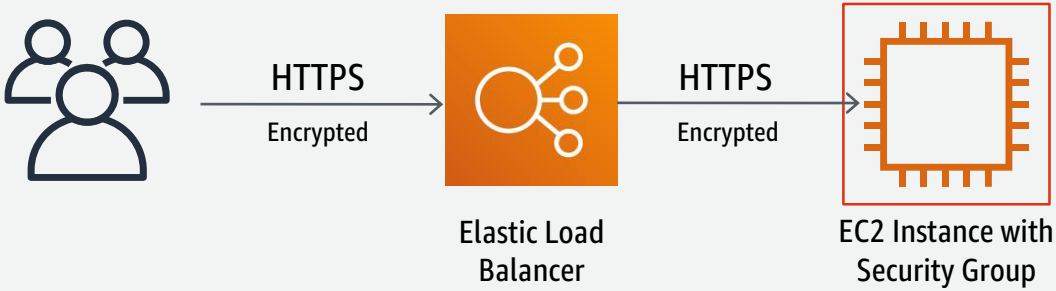


Encryption in Transit – Inside the VPC

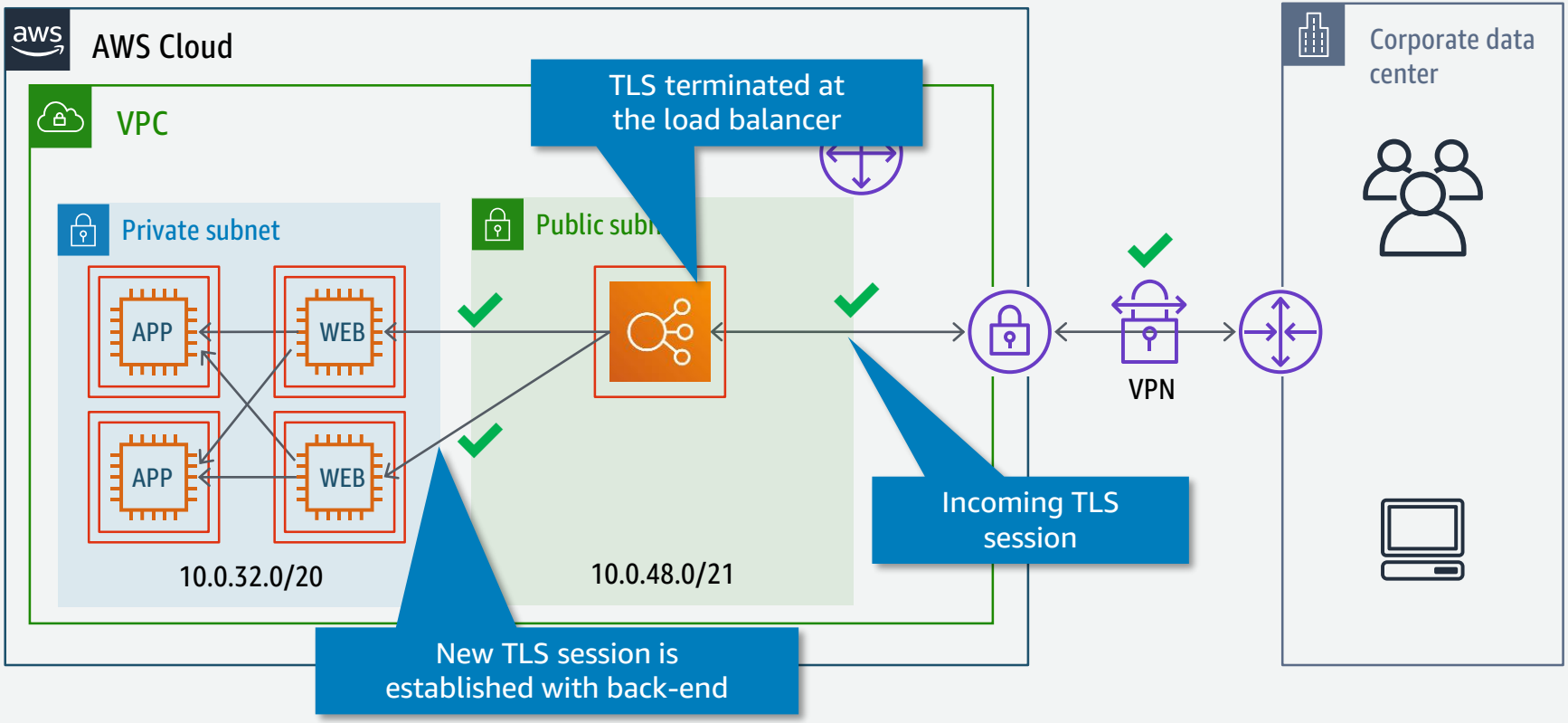


Encryption in Transit – TLS with Amazon ELB

You can use the ELB for HTTPS termination with encrypted communication to back-end instances on port 443.

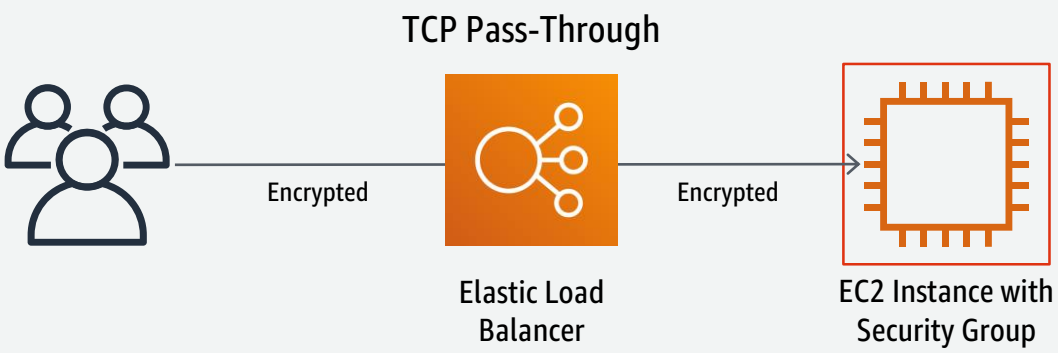


Encryption in Transit – Inside the VPC

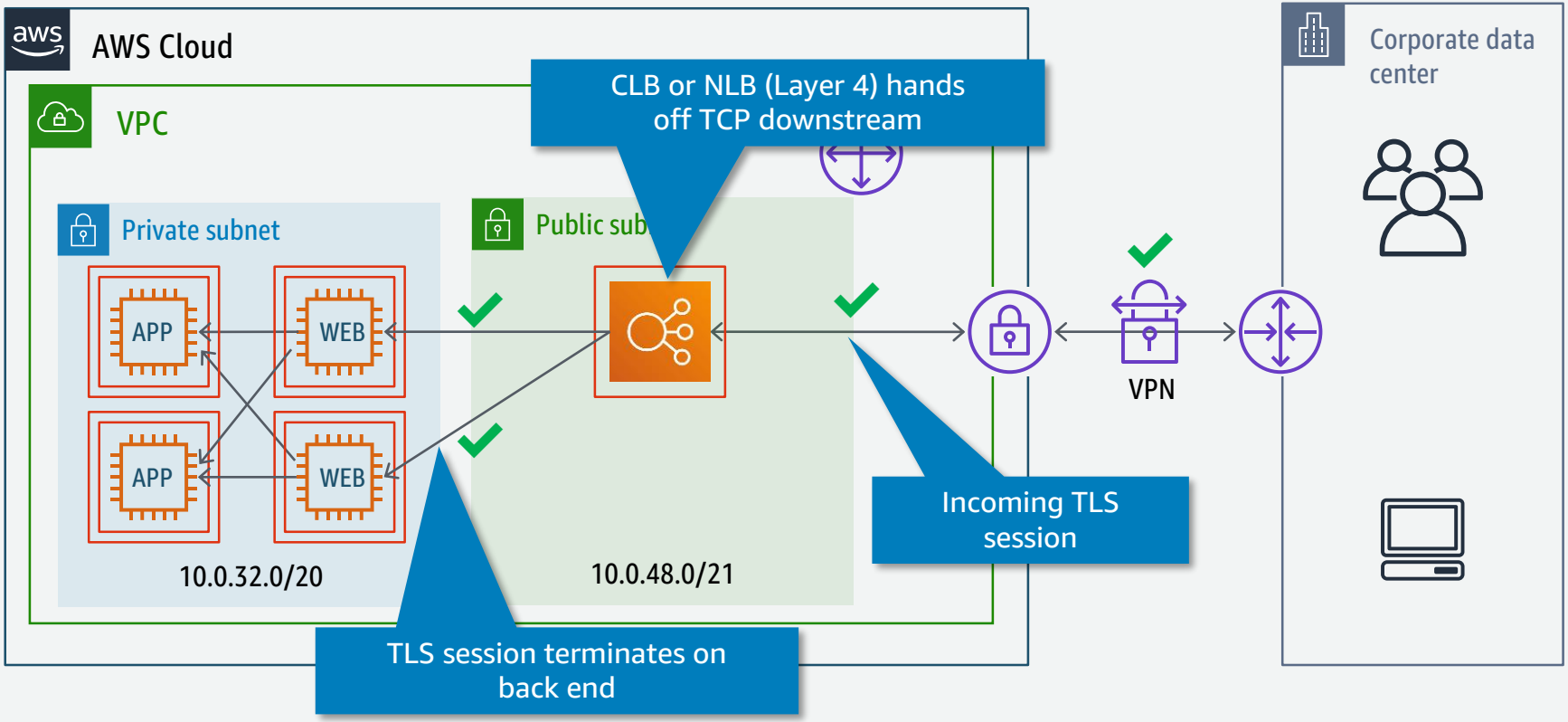


Encryption in Transit – TLS with Amazon ELB

Alternatively, you can use the Classic Load Balancer and Network Load Balancer in a TCP pass-through mode to terminate TLS connections on your EC2 instances



Encryption in Transit – Inside the VPC



Encryption in Transit – ELB Options

	Classic Load Balancer	Application Load Balancer	Network Load Balancer
Protocols	TCP, SSL/TLS, HTTP, HTTPS	HTTP, HTTPS	TCP, TLS
Network Layer	L4 – L7	L7	L4
Integration with ACM	✔	✔	✔
Back-end TLS authentication based on public-key	✔	✕	✕
Server Name Indication (SNI)	✕	✔	✕
Multiple security policies	✔	✔	✔
Custom security policy	✔	✕	✕

TLS Security Policies on Classic ELB

Security Policy	2016-08	TLS-1-1-2017-01	TLS-1-2-2017-01	2015-05	2015-03	2015-02
SSL Protocols						
Protocol-TLSv1	♦			♦	♦	♦
Protocol-TLSv1.1	♦	♦		♦	♦	♦
Protocol-TLSv1.2	♦	♦	♦	♦	♦	♦
SSL Options						
Server Order Preference	♦	♦	♦	♦	♦	♦
SSL Ciphers						
ECDHE-ECDSA-AES128-GCM-SHA256	♦	♦	♦	♦	♦	♦
ECDHE-RSA-AES128-GCM-SHA256	♦	♦	♦	♦	♦	♦
ECDHE-ECDSA-AES128-SHA256	♦	♦	♦	♦	♦	♦
ECDHE-RSA-AES128-SHA256	♦	♦	♦	♦	♦	♦
ECDHE-ECDSA-AES128-SHA	♦	♦		♦	♦	♦
ECDHE-RSA-AES128-SHA	♦	♦		♦	♦	♦

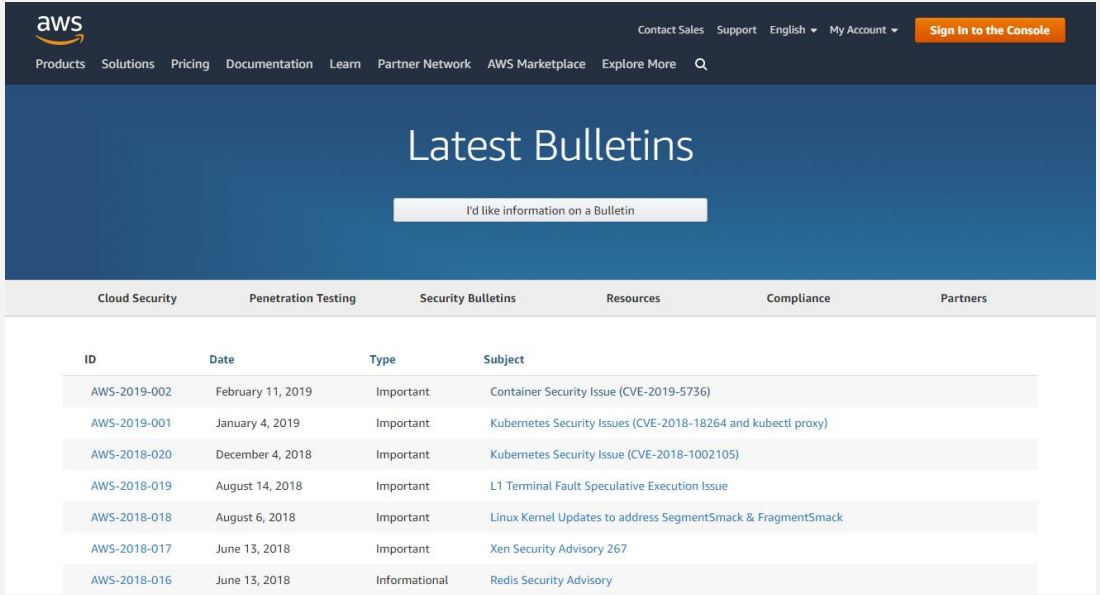
TLS Security Policies on ALB & NLB

Security Policy	2016-08 *	FS-2018-06	TLS-1-2	TLS-1-2-Ext	TLS-1-1	TLS-1-0 †
TLS Protocols						
Protocol-TLSv1	♦	♦				♦
Protocol-TLSv1.1	♦	♦			♦	♦
Protocol-TLSv1.2	♦	♦	♦	♦	♦	♦
TLS Ciphers						
ECDHE-ECDSA-AES128-GCM-SHA256	♦	♦	♦	♦	♦	♦
ECDHE-RSA-AES128-GCM-SHA256	♦	♦	♦	♦	♦	♦
ECDHE-ECDSA-AES128-SHA256	♦	♦	♦	♦	♦	♦
ECDHE-RSA-AES128-SHA256	♦	♦	♦	♦	♦	♦
ECDHE-ECDSA-AES128-SHA	♦	♦		♦	♦	♦
ECDHE-RSA-AES128-SHA	♦	♦		♦	♦	♦
ECDHE-ECDSA-AES256-GCM-SHA384	♦	♦	♦	♦	♦	♦

Encryption in Transit

Amazon was able to provide same-day mitigation for :

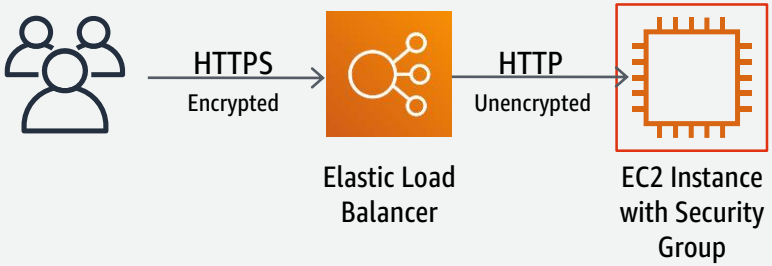
- Heartbleed
- POODLE
- LogJam

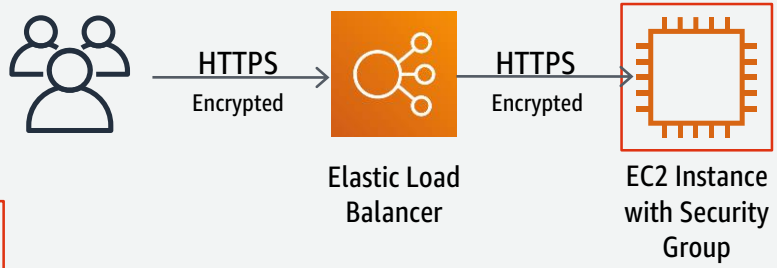


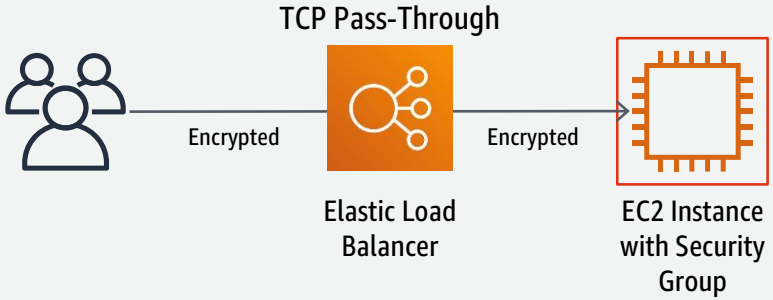
ID	Date	Type	Subject
AWS-2019-002	February 11, 2019	Important	Container Security Issue (CVE-2019-5736)
AWS-2019-001	January 4, 2019	Important	Kubernetes Security Issues (CVE-2018-18264 and kubectl proxy)
AWS-2018-020	December 4, 2018	Important	Kubernetes Security Issue (CVE-2018-1002105)
AWS-2018-019	August 14, 2018	Important	L1 Terminal Fault Speculative Execution Issue
AWS-2018-018	August 6, 2018	Important	Linux Kernel Updates to address SegmentSmack & FragmentSmack
AWS-2018-017	June 13, 2018	Important	Xen Security Advisory 267
AWS-2018-016	June 13, 2018	Informational	Redis Security Advisory

https://aws.amazon.com/security/security-bulletins/

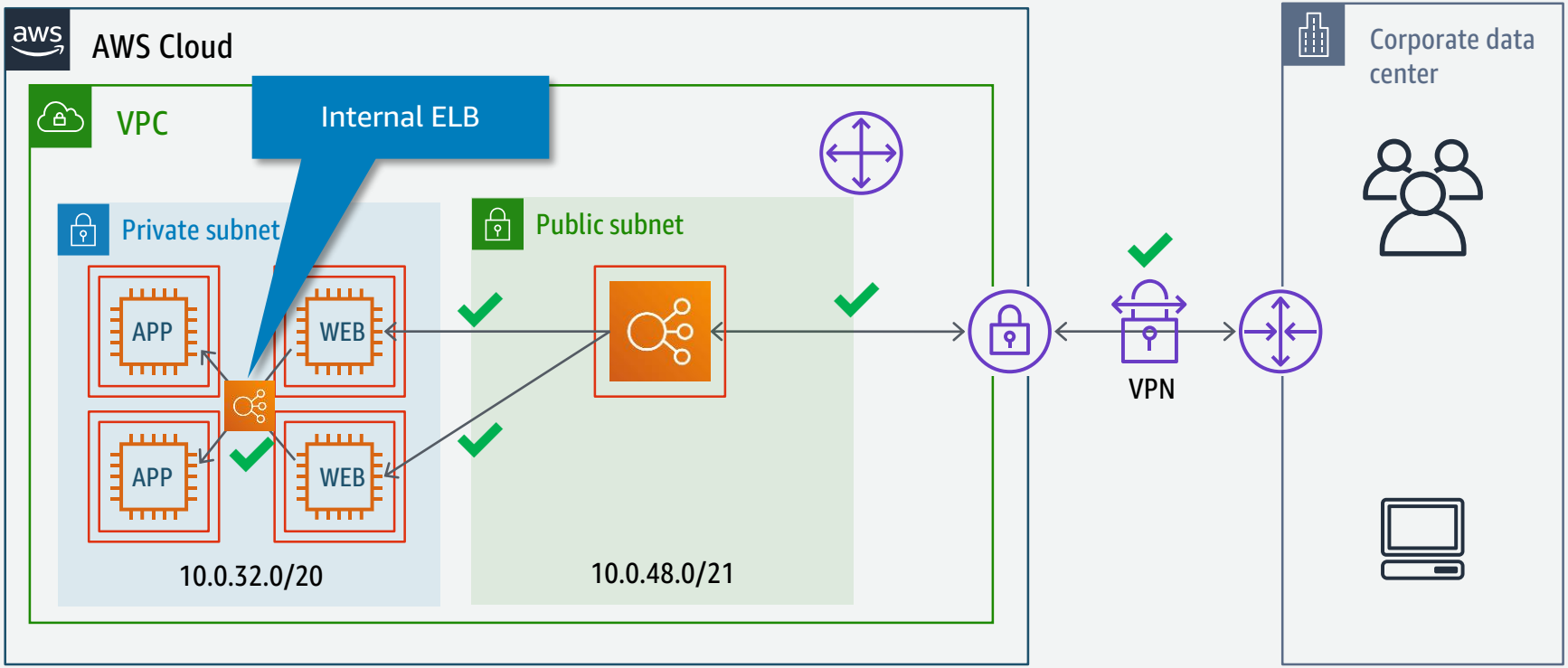
Encryption in Transit – TLS with Amazon ELB Recap



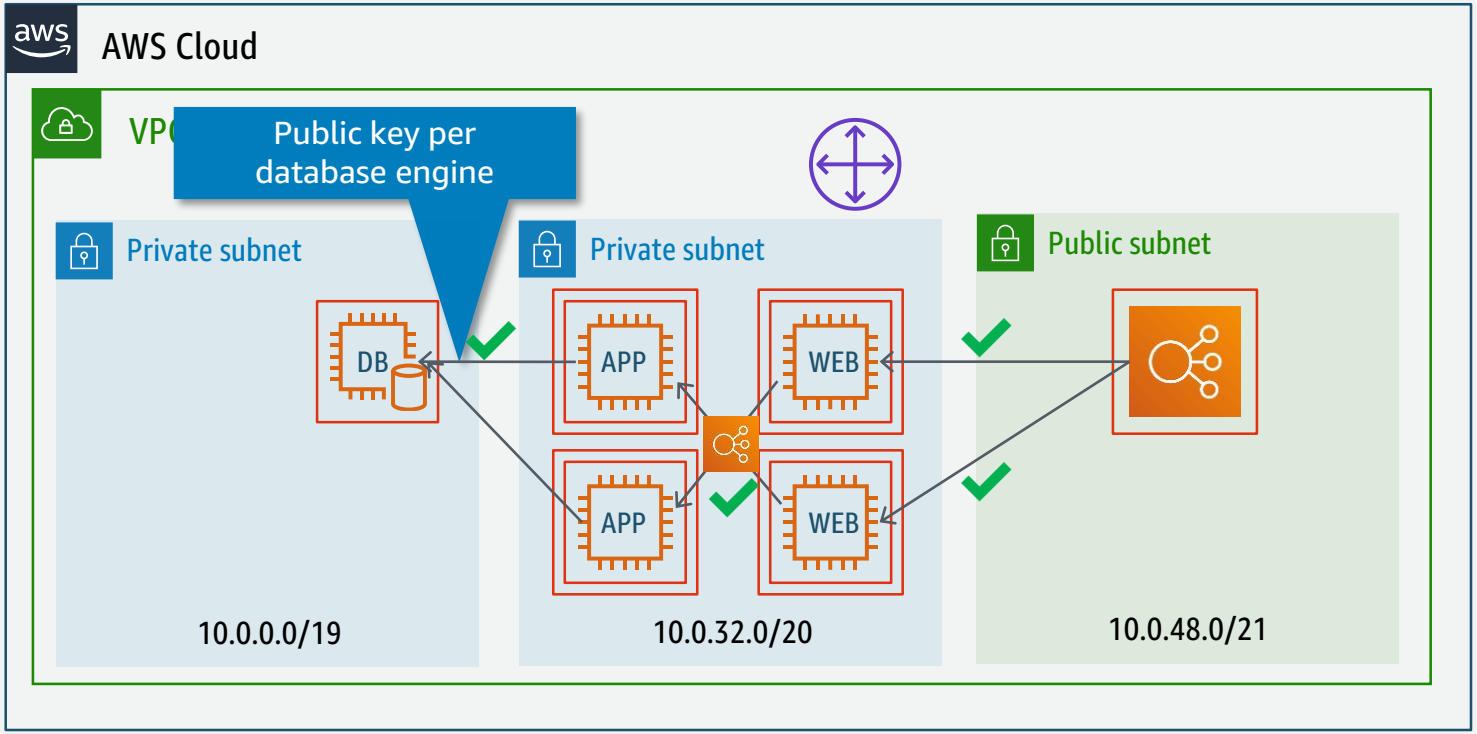




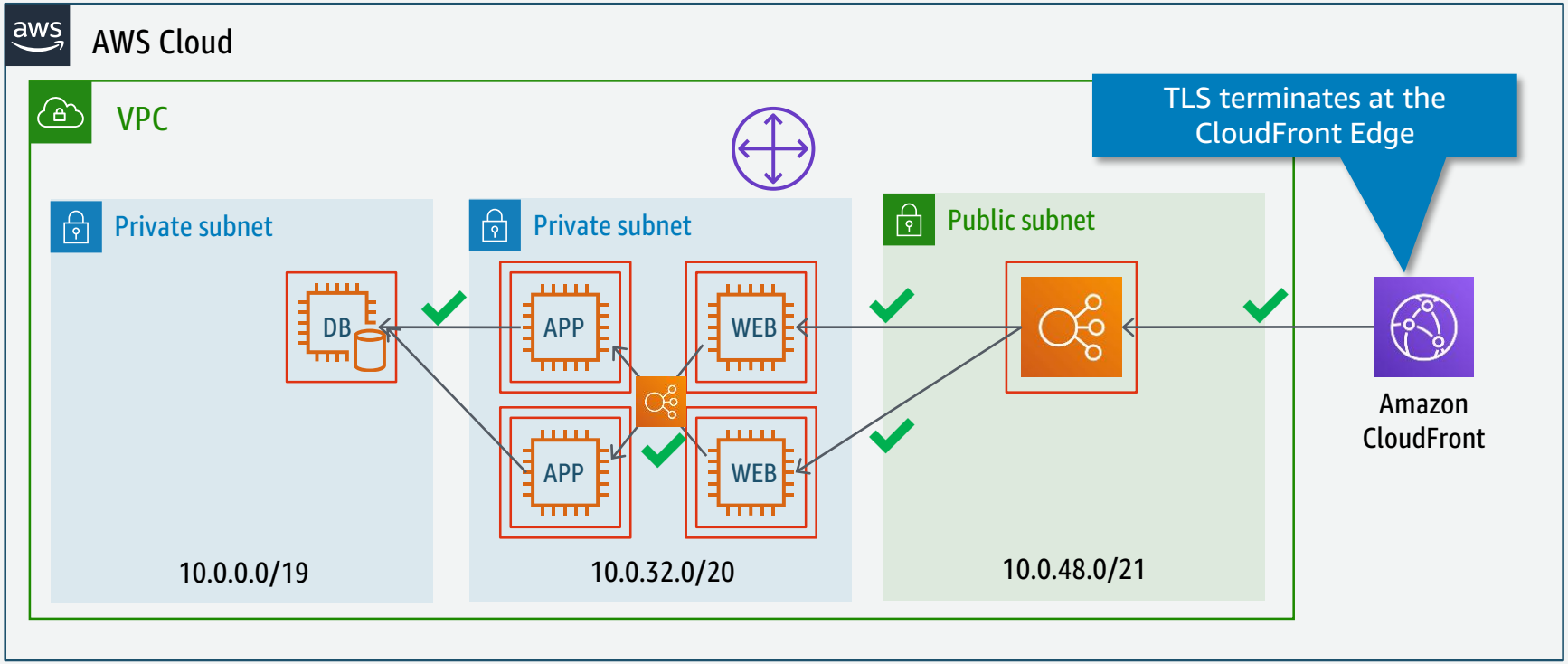
Encryption in Transit – Inside the VPC



Encryption in Transit – Inside the VPC



Encryption in Transit – Inside the VPC



Encryption in Transit – Amazon Certificate Manager

- Provision trusted SSL/TLS certificates from AWS for use with AWS resources:
  - Elastic Load Balancing
  - Amazon CloudFront distributions
- AWS handles the muck
  - Key pair and CSR generation
  - Managed renewal and deployment
- Domain validation (DV) through email or DNS (Route 53)
- Available through AWS Management Console, AWS Command Line Interface (AWS CLI), or API

What is Shared Responsibility Model (review)?

From the VPC standpoint, the Shared Responsibility Model is:

AWS Responsibility	Customer Responsibility
Security of the Cloud	**Security in the Cloud**
Provide a resilient and secure underlying infrastructure and Software Defined Network (SDN)	Configure customer-specific controls
Audited/Certified on SOC1/2, ISO27001, FedRAMP, HIPAA BAA, PCI, etc.	*Subnets, routing table, security groups, network ACL, gateways, logging, encryption, access controls*
Support the service for most of AWS 1M active customers	

Risk on Data Transmission

Risk Description	Actor
Data confidentiality is compromised during transmission → Within 1 Subnet → Between Subnets in 1 VPC → Between VPCs	Internal → Same Account External → AWS Team → Other AWS Customers → All others

Risk assessment

Data	Risk	Actor	Criticality	Strategy	Controls
Non sensitive	**Within Subnet (A)**	**Any**	**Low**	**Accept** Performance / Complexity / Value	Access Controls No encryption
Sensitive	**Within Subnet (A)** **Between Subnets (B)**	**Internal** Infrastructure Changes (Routing Table)	**Med-Low**	**Mitigate**	Access Controls Encryption if feasible Monitor Config Changes
Sensitive	**Between VPCs (C)**	**Internal** Infrastructure Changes (Routing Table/VPN Peering/vPG)	**Med**	**Mitigate**	Access Controls Encryption if feasible Monitor Config Changes
Sensitive	**Any VPC trans. type**	**AWS Team** Access to underlying	**Low**	**Mitigate**	Encryption if feasible
Sensitive	**Any VPC trans. type**	**Other AWS Customers** Bug/Spoofing/Snooping	**Low**	**Mitigate**	Encryption if feasible
Sensitive	**Any VPC trans. type**	**All others** Bug Misconfiguration	**Med-Low**	**Mitigate**	AWS Controls Access Controls Encryption if feasible

Recommendations for customers

VPC is a virtual *private* cloud, not public cloud
VPC is secure, audited and certified
Encryption-in-transit has benefits (additional assurance) and costs (complexity, performance)

Recommendations on Use cases

Use Case	Data type	Controls to be *(in-place)*	Priority
Application1	*Customer data & public facing*	Enforce encryption in transit *(Mostly done with SSL/TLS)*	**High**
Application2	*Customer data & internal*	Best effort on encryption in transit *(Not done yet)*	**Mid**
Application3	*No customer data & public facing*	Best effort on encryption in transit *(Not done yet)*	**Low**

Data Protection

Data Protection – Least Privilege Access to Data

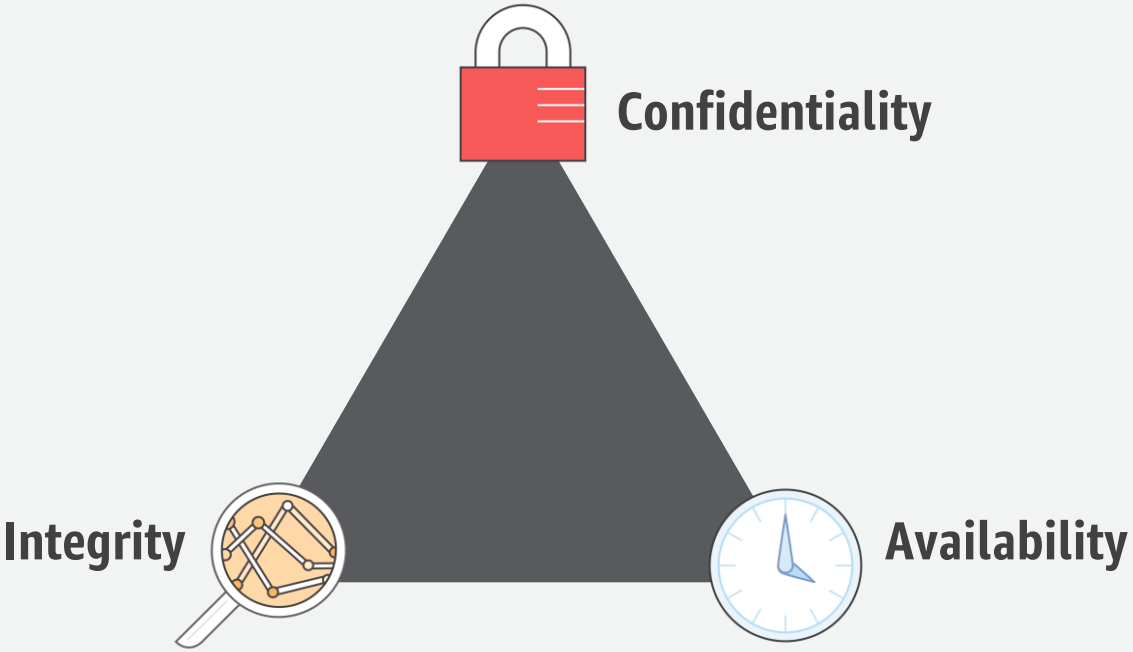
Security best practice

- Start with a minimum set of permissions
- Grant additional permissions as necessary

Define only the required set of permissions

- What actions a particular service supports
- What collection of API actions are required for the specific task
- What permissions are required to perform those actions

Data Protection



Data Protection – AWS Storage Services

Amazon S3

Confidentiality: read/write object permissions (IAM and resource policies); MFA for deleting data
Availability/Durability: S3 cross-region replication; versioning allows recovery of deleted objects
Integrity: object integrity automatically provided

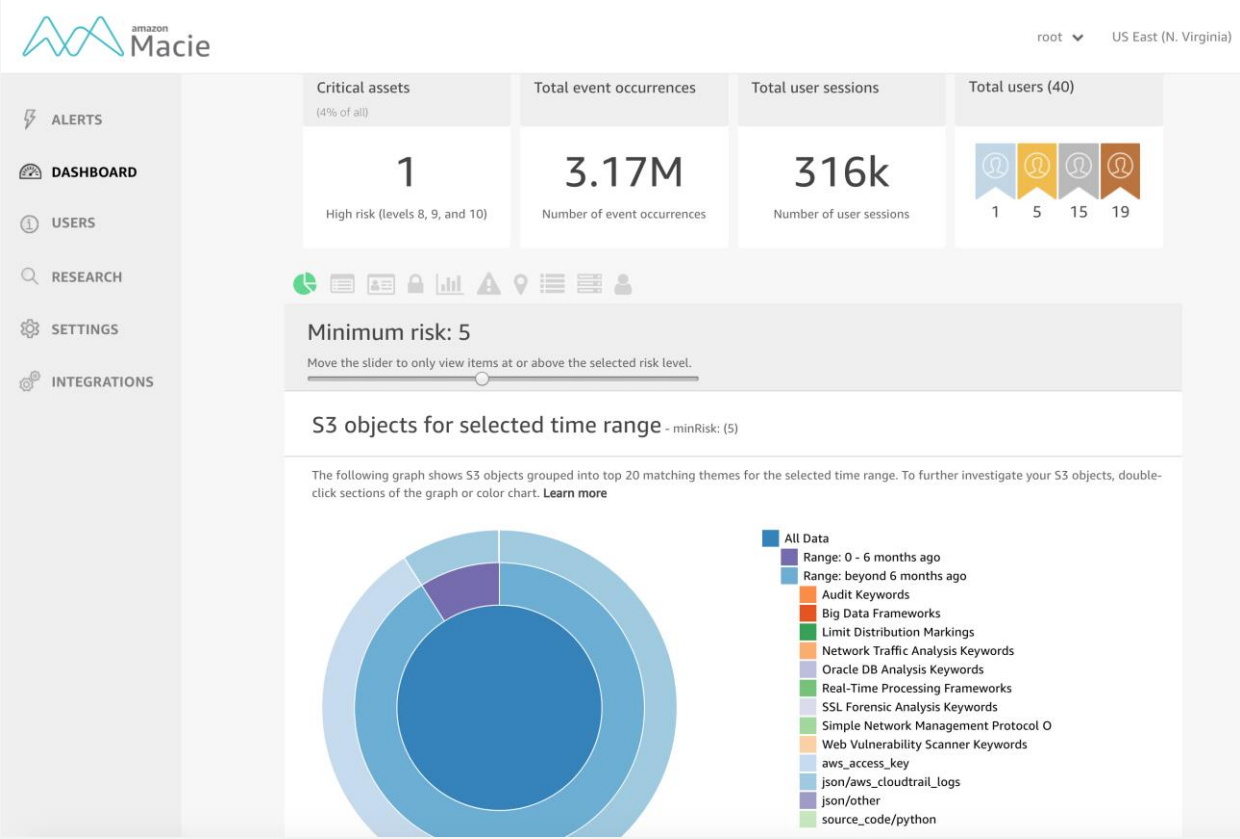
Amazon EBS

Confidentiality: tag-based IAM policies
Availability/Durability : share snapshots between accounts and copy between AWS regions
Integrity: block integrity automatically provided

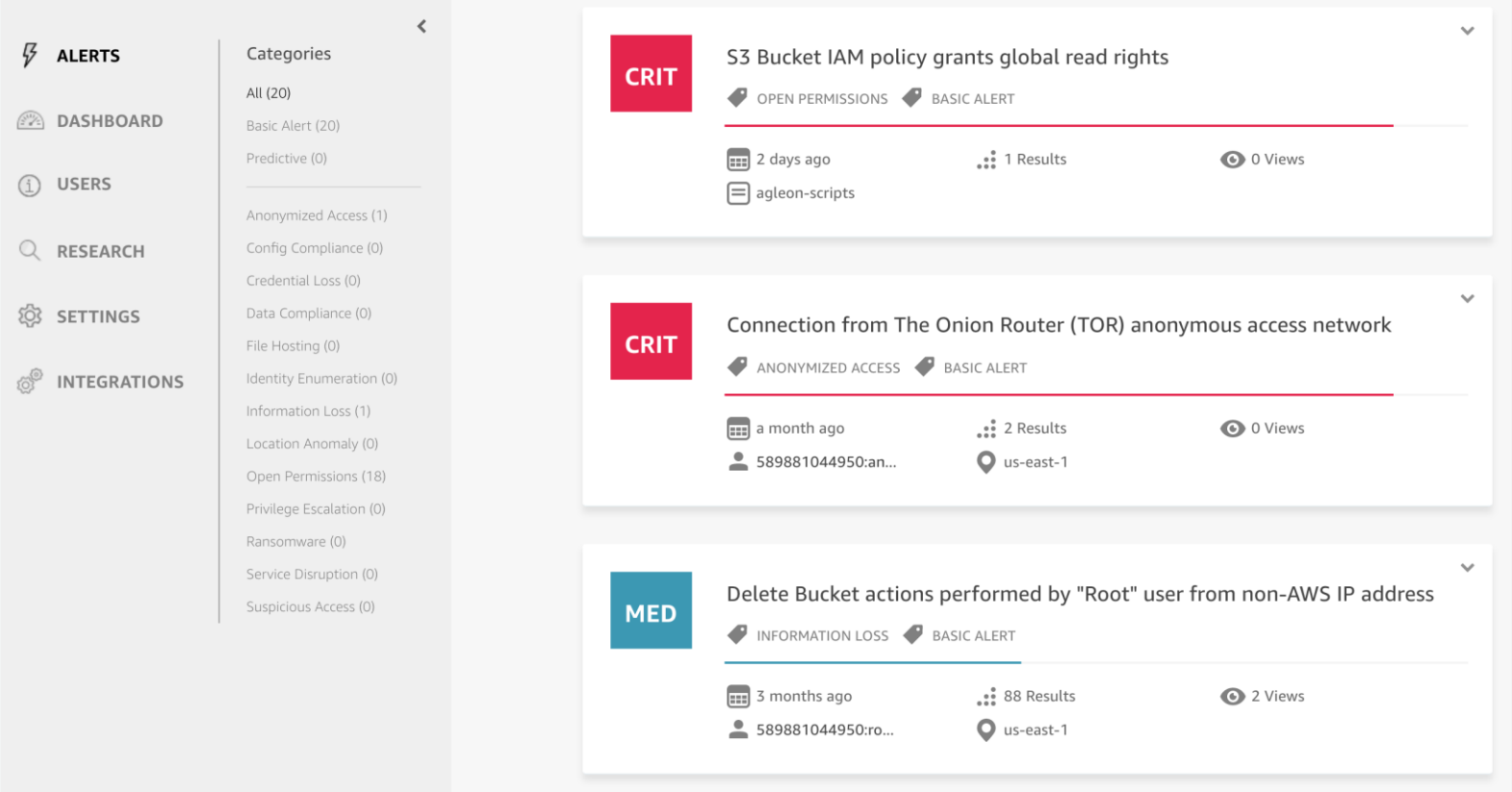
Amazon EFS

Confidentiality: IAM policies for attachment; POSIX permission for files / directories
Availability/Durability : share snapshots between accounts and copy between AWS regions
Integrity: file integrity automatically provided

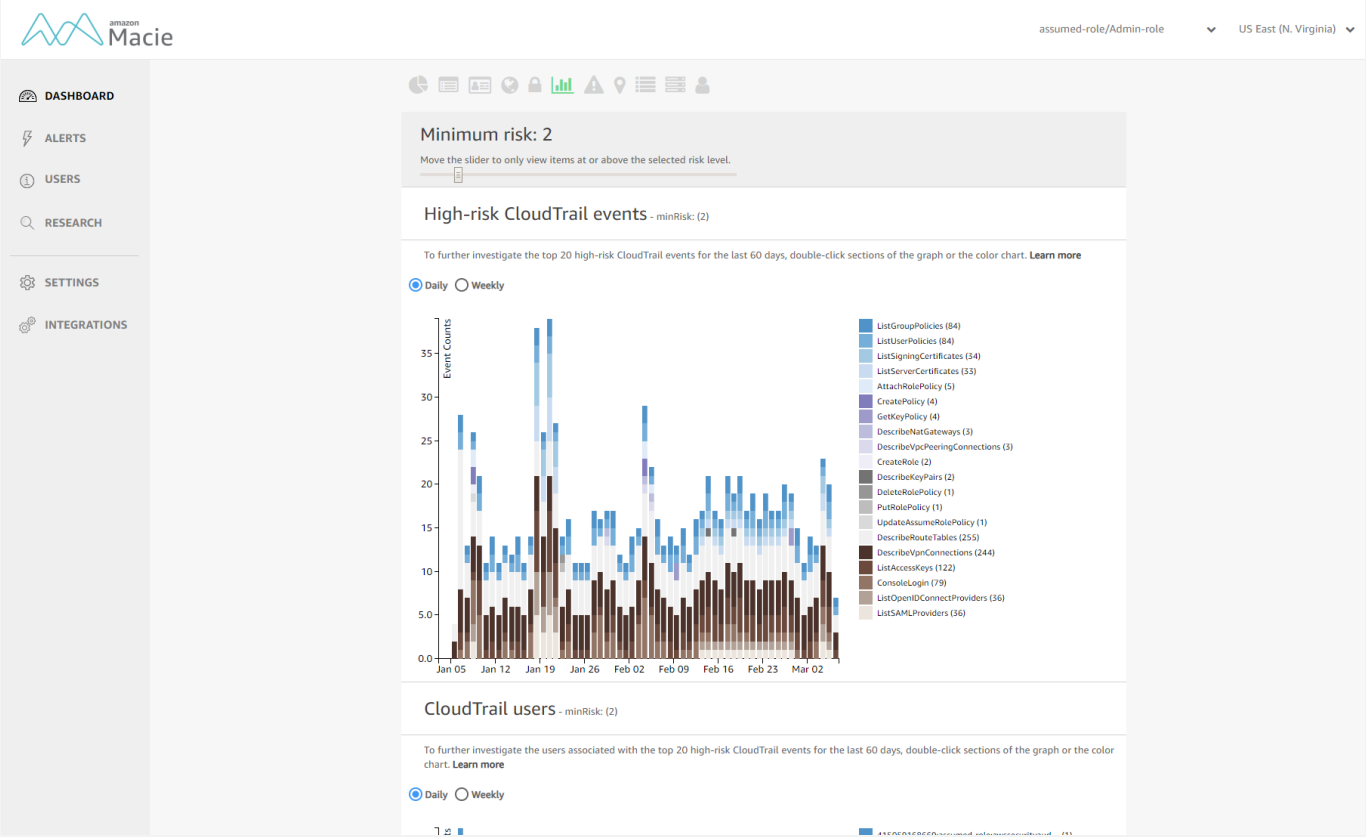
Data Protection – Amazon Macie



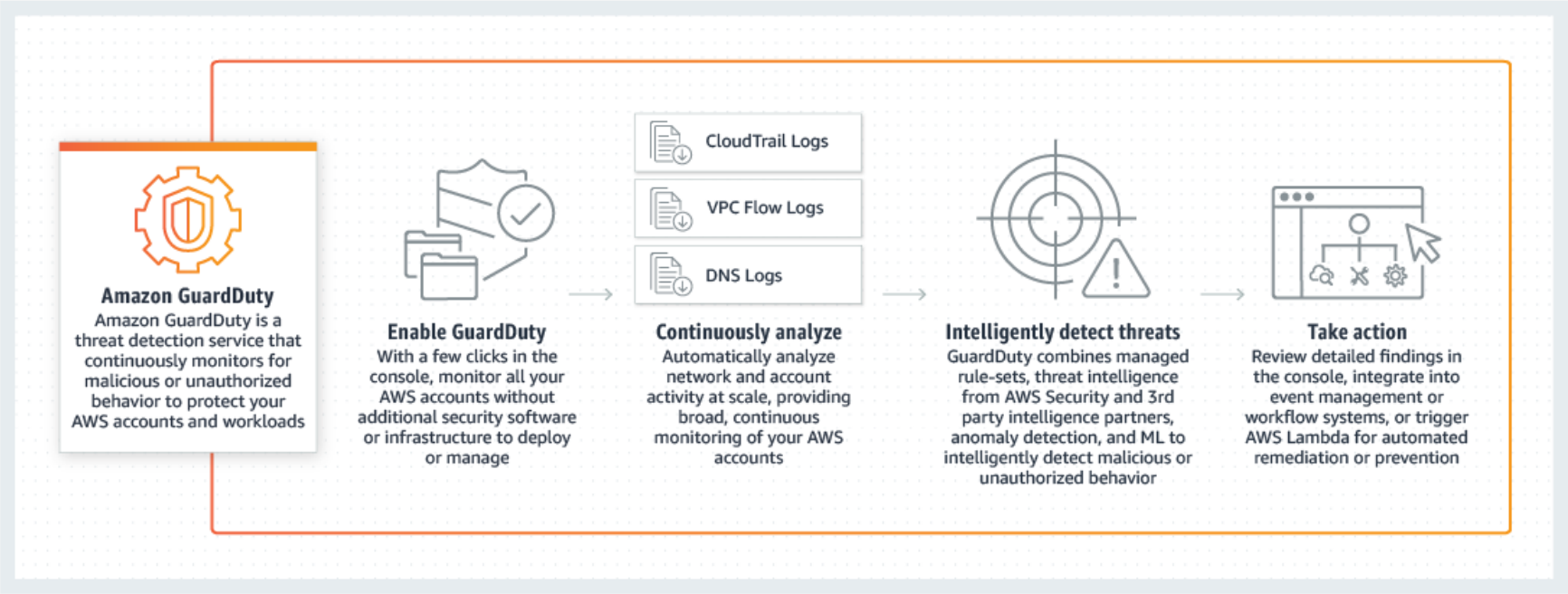
Data Protection – Amazon Macie



Data Protection – Amazon Macie



Data Protection – Amazon GuardDuty (review)



Data Protection – Amazon GuardDuty

Use GuardDuty to identify threats in your AWS environment across three major categories:

- **Reconnaissance**
- **Instance compromise**
- **Account compromise**

Data Protection Example Findings:

- EC2/TrafficVolumeUnusual
- IAMUser/ResourcePermissions
- IAMUser/UserPermissions
- IAMUser/RootCredentialUsage

https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_finding-types-active.html

Questions?