Managing security in AWS

Viral Desai, AWS Solutions Architect

virdesai@amazon.com

10/16/2018

The most sensitive workloads run on AWS

"We determined that security in AWS is superior to our on-premises data center across several dimensions, including patching, encryption, auditing and logging, entitlements, and compliance."

—John Brady, CISO, FINRA (Financial Industry Regulatory Authority)

"The fact that we can rely on the AWS security posture to boost our own security is really important for our business. AWS does a much better job at security than we could ever do running a cage in a data center."

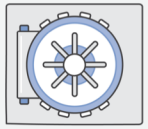
— Richard Crowley, Director of Operations, Slack

"With AWS, DNAnexus enables enterprises worldwide to perform genomic analysis and clinical studies in a secure and compliant environment at a scale not previously possible."

— Richard Daly, CEO DNAnexus

aws

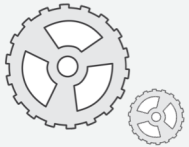
Security is Job ZERO at AWS
Strengthen your security posture



Inherit global security and compliance controls

Scale with superior visibility and control

Highest standards for privacy and data security



Automate with deeply integrated security services

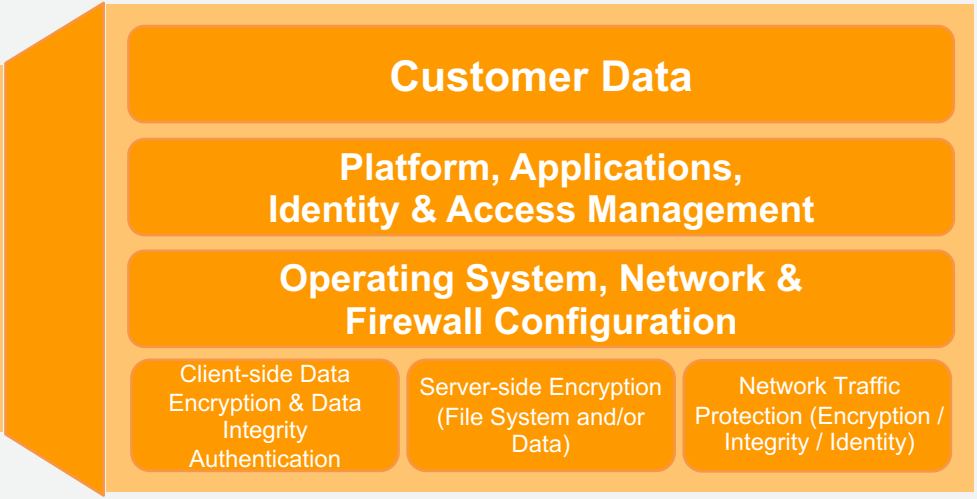
Largest network of security partners and solutions

aws

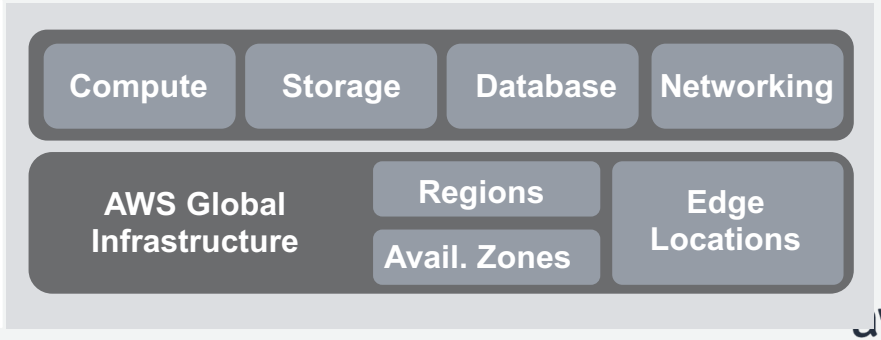
The AWS Shared Responsibility

Hopefully, you've seen this before…

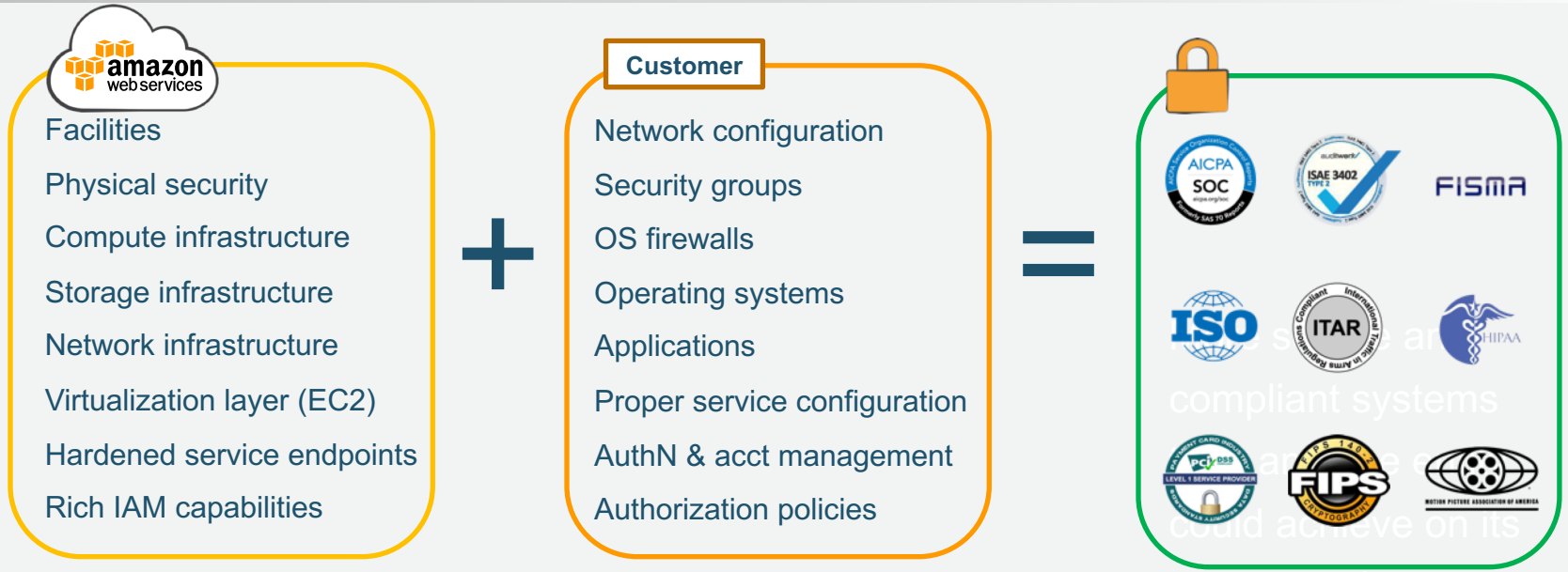
Customers are responsible for security *'in'* the Cloud



AWS is responsible for security *'of'* the Cloud



AWS Shared Responsibility Model



- Scope of responsibility depends on the type of service offered by AWS: **Infrastructure, Container, Abstracted Services**
- Understanding who is responsible for what is critical to ensuring your AWS data and systems are secure!

aws

Access a deep set of cloud security tools

Networking & Security


Amazon GuardDuty


Amazon VPC


AWS Direct Connect


VPN connection


Security Groups


Flow logs


AWS Shield


AWS WAF

Route table


AWS Firewall Manager

Compliance & Governance


AWS Service Catalog


AWS Trusted Advisor


AWS CloudFormation


AWS CloudTrail


AWS Systems Manager


Amazon CloudWatch


AWS Config


AWS Artifact


Amazon Inspector


AWS OpsWorks

Identity


Amazon Cognito


IAM


AWS Directory Service


AWS Organizations


Active Directory integration

AWS Single Sign-on

Temporary Security credential

SAML Federation

Encryption


AWS KMS


AWS Secrets Manager


AWS CloudHSM


Client-side encryption


AWS Certificate Manager

aws

What does this mean?

- You benefit from an environment built for the most security sensitive organizations
- AWS manages a multitude of security controls **so you don’t have to**
- You get to define the right security controls for your workload sensitivity
- You always have full ownership and control of your data

aws

Identity and Access Management (IAM)

Policy specification basics

- JSON-formatted documents
- Contain a statement (permissions) that specifies:
  - Which actions a principal can perform
  - Which resources can be accessed

```
{
 "Statement":[{
  "Effect":"effect",
  "Principal":"principal",
  "Action":"action",
  "Resource":"arn",
  "Condition":{
   "condition":{
    "key":"value" }
   }
  }
 ]
}
```

Principal
Action
Resource
Condition

You can have multiple statements and each statement is comprised of PARC.

aws

Policy specification basics

- JSON-formatted documents
- Contain a statement (permissions) that specifies:
  - Which actions a principal can perform
  - Which resources can be accessed

```
{
 "Statement":[{
  "Effect":"effect",
  "Principal":"principal",
  "Action":"action",
  "Resource":"arn",
  "Condition":{
   "condition":{
    "key":"value" }
   }
  }
 ]
}
```

Principal
Action
Resource
Condition

You can have multiple statements and each statement is comprised of PARC.

aws

Principal – Examples

Principal
Action
Resource
Condition

- An entity that is allowed or denied access to a resource
- Indicated by an Amazon Resource Name (ARN)
- With IAM policies, the principal element is implicit (i.e., the user, group, or role attached)

```
<!-- Everyone (anonymous users) -->
"Principal":"AWS":"*.*"

<!-- Specific account or accounts -->
"Principal":{"AWS":"arn:aws:iam::123456789012:root" }
"Principal":{"AWS":"123456789012"}

<!-- Individual IAM user -->
"Principal":"AWS":"arn:aws:iam::123456789012:user/username"

<!-- Federated user (using web identity federation) -->
"Principal":{"Federated":"accounts.google.com"}

<!-- Specific role -->
"Principal":{"AWS":"arn:aws:iam::123456789012:role/rolename"}

<!-- Specific service -->
"Principal":{"Service":"ec2.amazonaws.com"}
```

aws

Action – Examples

Principal
Action
Resource
Condition

- Describes the type of access that should be allowed or denied
- You can find actions in the docs or use the policy editor to get a drop-down list
- Statements must include either an `Action` or `NotAction` element

```
<!-- EC2 action -->
"Action":"ec2:StartInstances"

<!-- IAM action -->
"Action":"iam:ChangePassword"

<!- Amazon S3 action -->
"Action":"s3:GetObject"

<!-- Specify multiple values for the Action element-->
"Action":["sqs:SendMessage","sqs:ReceiveMessage"]

<-- Wildcards (* or ?) in the action name. Below covers create/delete/list/update-->
"Action":"iam:*AccessKey*"
```


aws

Understanding `NotAction`

- Lets you specify an exception to a list of actions
- Could result in shorter policies than using `Action` and exclude many actions
- Example: Let's say you want to allow everything but IAM APIs

```
{
  "Version": "2012-10-17",
  "Statement": [ {
      "Effect": "Allow",
      "NotAction": "iam:*",
      "Resource": "*"
    }
  ]
}
```


```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Action": "*",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": "iam:*",
      "Resource": "*"
    }
  ]
}
```


aws

Understanding `NotAction`

- Lets you specify an exception to a list of actions
- Could result in shorter policies than using `Action` and exclude many actions
- Example: Let's say you want to allow everything but IAM APIs

```
{
  "Version": "2012-10-17",
  "Statement": [ {
      "Effect": "Allow",
      "NotAction": "iam:*",
      "Resource": "*"
    }
  ]
}
```

This is not a **`Deny`**. A user could still have a separate policy that grants **`IAM:*`**


```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Action": "*",
      "Resource": "*"
    },
    {
      "Effect": "Deny",
      "Action": "iam:*",
      "Resource": "*"
    }
  ]
}
```

If you want to prevent the user from ever being able to call IAM APIs, use an explicit **`Deny`**.

Policy specification basics

- JSON-formatted documents
- Contain a statement (permissions) that specifies:
  - Which actions a principal can perform
  - Which resources can be accessed

```
{
 "Statement":[{
  "Effect":"effect",
  "Principal":"principal",
  "Action":"action",
  "Resource":"arn",
  "Condition":{
   "condition":{
    "key":"value" }
   }
  }
 ]
}
```

Principal
Action
Resource
Condition

You can have multiple statements and each statement is comprised of PARC.

aws

Resource – Examples

Principal
Action
Resource
Condition

- The object or objects being requested
- Statements must include either a `Resource` or a `NotResource` element

```
<-- S3 bucket -->
"Resource":"arn:aws:s3:::my_corporate_bucket/*"

<-- All S3 buckets, except this one -->
"NotResource":"arn:aws:s3:::security_logging_bucket/*"

<-- Amazon SQS queue-->
"Resource":"arn:aws:sqs:us-west-2:123456789012:queue1"

<-- Multiple Amazon DynamoDB tables -->
"Resource":["arn:aws:dynamodb:us-west-2:123456789012:table/books_table",
            "arn:aws:dynamodb:us-west-2:123456789012:table/magazines_table"]

<-- All EC2 instances for an account in a region -->
"Resource": "arn:aws:ec2:us-east-1:123456789012:instance/*"
```

Replace with your account number

aws

Condition example

Principal
Action
Resource
Condition

What if you wanted to restrict access to a time frame and IP address range?

```
“Condition” : {
  "DateGreaterThan" : {"aws:CurrentTime" : "2017-01-01T11:00:00Z"},
  "DateLessThan": {"aws:CurrentTime" : "2017-12-31T15:00:00Z"},
  "IpAddress" : {"aws:SourceIp" : ["192.0.2.0/24", "203.0.113.0/24"]}
}
```

AND

OR

- Allows a user to access a resource under the following conditions:
  - The time is after 11:00 A.M. on 01/01/2017 AND
  - The time is before 3:00 P.M. on 12/31/2017 AND
  - The request comes from an IP address in the 192.0.2.0 /24 OR 203.0.113.0 /24 range
- All of these conditions must be met in order for the statement to evaluate to TRUE.

The anatomy of a policy with variables

```
{
  "Version": "2012-10-17",
  "Statement": [{
     "Effect": "Allow",
     "Action": ["s3:ListBucket"],
     "Resource": ["arn:aws:s3:::myBucket"],
     "Condition":
       {"StringLike":
          {"s3:prefix":["home/${aws:username}/*"]}
        }
    },
    {
      "Effect":"Allow",
      "Action":["s3:*"],
      "Resource": ["arn:aws:s3:::myBucket/home/${aws:username}",
                   "arn:aws:s3:::myBucket/home/${aws:username}/*"]
    }
  ]
}
```

Grants a user access to a home directory in S3 that can be accessed programmatically

aws

Controlling access to EC2

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:RebootInstances",
                "ec2:RunInstances",
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:TerminateInstances"
            ],
            "Condition": {
                "StringLikeIfExists": {
                        "ec2:Region": [ "us-east-1", "us-west-2“ ],
                        "ec2:InstanceType": [ "t1.*, "t2.*", "m3.*“ ]
                }
             },
              "Resource": "*"
         }
    ]
}
```

Basic policy hygiene

Basic control actions for EC2

Use of `IfExists` makes sure your policy works the way you expect it to.

aws

Policy enforcement

1. Decision starts at Deny
2. Evaluate all applicable policies
3. Is there an explicit Deny?
4. Is there an Allow?
5. Final decision ="Deny" (default Deny)

- AWS retrieves all policies associated with the user and resource.
- Only policies that match the action and conditions are evaluated.

Yes

Final decision ="Deny" (explicit Deny)

- If a policy statement has a Deny, it trumps all other policy statements.

No

Yes

Final decision ="Allow"

- Access is granted if there is an explicit Allow and no Deny.

No

- By default, an implicit (default) Deny is returned.

aws

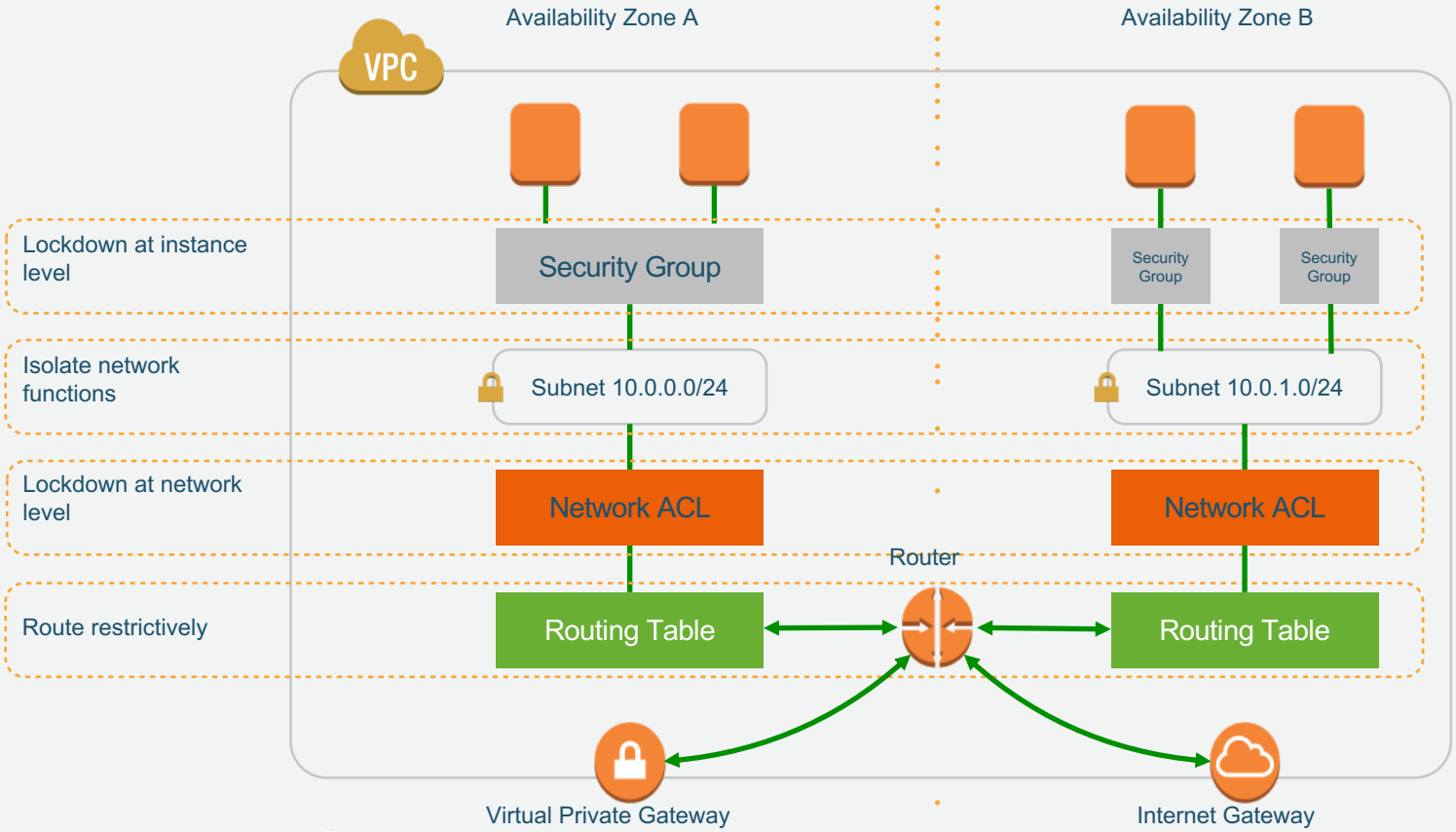
IAM Best Practices

- Lock away your AWS account access keys
- Create individual IAM users
- Use groups to assign permissions to IAM users
- Grant least privilege
- Configure a strong password policy for your users
- Enable MFA for privileged users
- Use roles for applications that run on Amazon EC2 instances
- Delegate by using roles instead of by sharing credentials
- Rotate credentials regularly
- Remove unnecessary credentials
- Use policy conditions
- Keep an audit log

aws

How do you secure the network in AWS?

VPC security layers



How are you currently encrypting your data?

Encryption

Protecting data in-transit and at-rest.

Encryption In-Transit

- HTTPS
- SSL/TLS
- VPN / IPSEC
- SSH

Encryption At-Rest

- Object
- Database
- Filesystem
- Disk

Details about encryption can be found in the AWS Whitepaper, "Securing Data at Rest with Encryption".

aws

AWS Certificate Manager

Provision trusted SSL/TLS certificates from AWS for use with integrated AWS resources

- Elastic Load Balancing
- Amazon CloudFront distributions
- APIs on API Gateway

AWS handles the muck

- Key pair and CSR generation
- Managed renewal and deployment

Domain validation (DV) through email

ACM-provided Certificates

Domain names

- Single domain name: www.example.com
- Wildcard domain names: *.example.com
- Combination of wildcard and non-wildcard names
- Multiple domain names in the same certificate (up to 10)

ACM-provided certificates are **managed**

- Private keys are generated, protected, and managed
- ACM-provided certificates cannot be used on Amazon EC2 instances or on-premises servers
- Can be used with AWS services, such as Elastic Load Balancing and Amazon CloudFront, API Gateway

Algorithms

- RSA 2048 and SHA-256

Free

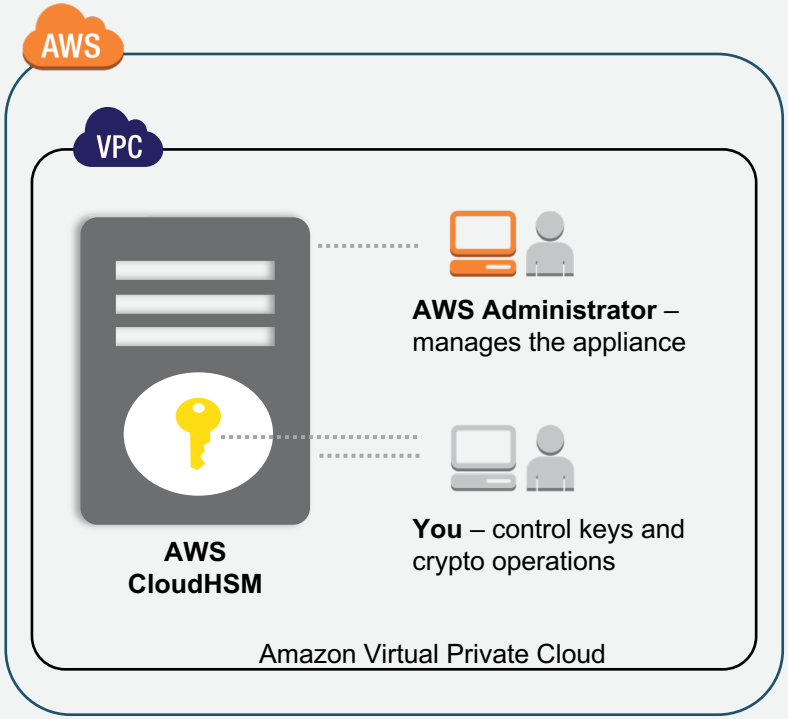
aws

AWS CloudHSM

Help meet compliance requirements for data security by using a dedicated Hardware Security Module appliance with AWS.

- Dedicated, single-tenant hardware device
- Can be deployed as HA and load balanced

- Customer use cases:
  - Oracle TDE
  - MS SQL Server TDE
  - Setup SSL connections
  - Digital Rights Management (DRM)
  - Document Signing



AWS Key Management Service

- Managed service that simplifies creation, control, rotation, deletion, and use of encryption keys in your applications
- Integrated with 19 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

aws

Encryption at Rest

Volume Encryption

- EBS Encryption
- Filesystem Tools
- AWS Marketplace/Partner

Object Encryption

- S3 Server Side Encryption (SSE)
- S3 SSE w/ Customer Provided Keys
- Client-Side Encryption

Database Encryption

- RDS MSSQL TDE
- RDS ORACLE TDE/HSM
- RDS MYSQL KMS
- RDS PostgreSQL KMS
- Redshift Encryption

KMS integration with AWS services

- **Storage:** EBS, S3, Snowball, ECS
- **Database:** All RDS engines, DMS
- **Data analytics:** Redshift, EMR, Kinesis Firehose
- **Enterprise apps:** WorkMail, WorkSpaces
- **Developer tools:** AWS CodeCommit, AWS CodePipeline
- **Management:** CloudTrail, CloudWatch Logs
- **App svcs:** Elastic Transcoder, Simple Email Service, CloudSearch
- **AWS IoT**

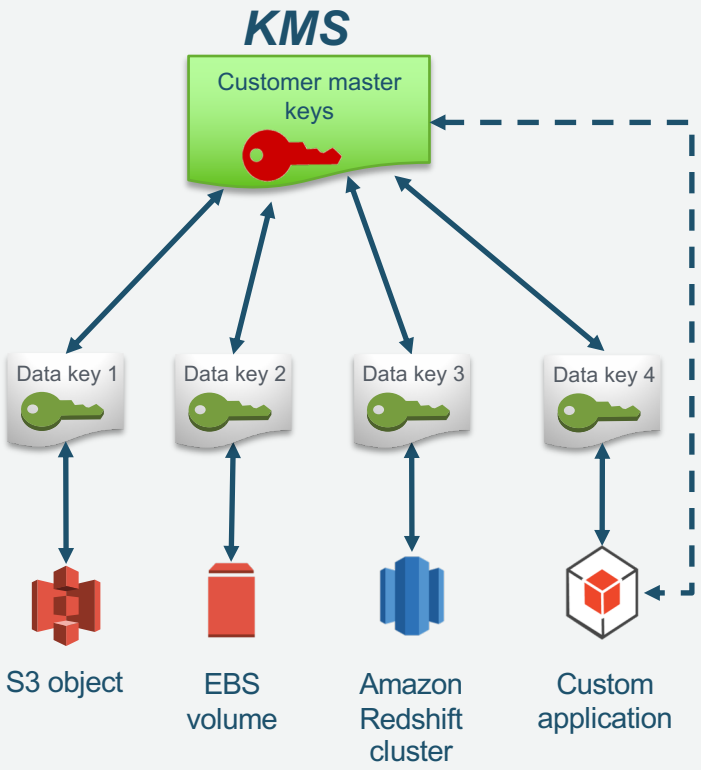
How clients and AWS services typically integrate with KMS

Two-tiered key hierarchy using envelope encryption

- Unique data key encrypts customer data
- KMS master keys encrypt data keys

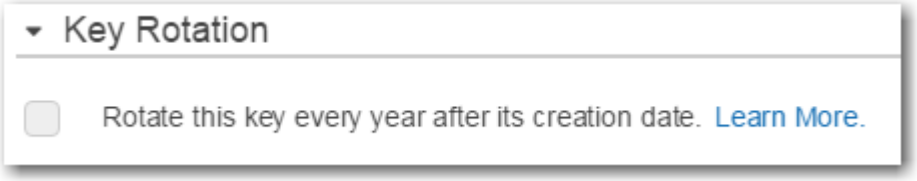
Benefits

- 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



Rotating master keys in KMS

Console (Key Summary page)



AWS CLI

enable-key-rotation --key-id <value>

What key rotation means:

- A new version of a master key is created, but mapped to the same key ID (or alias)
- New encryption requests use the new version
- Previous versions of master keys are kept to perform decryption on older ciphertexts
- No version management needed by you – the same key ID or alias just works

aws

Auditability of KMS key usage through AWS CloudTrail

```
"EventName":"DecryptResult",
"EventTime":"2014-08-18T18:13:07Z",
"RequestParameters":
        "{\"keyId\":\"2b42x363-1911-4e3a-8321-6b67329025ex"}",
"EncryptionContext":"volumeid-12345",
"SourceIPAddress":" 203.0.113.113",
"UserIdentity":
      "{\"arn\":\"arn:aws:iam:: 111122223333:user/User123"}
```

This KMS API action was called…

….at this time

…in reference to this key

…to protect this AWS resource

…from this IP address

…by this AWS user in this account

aws

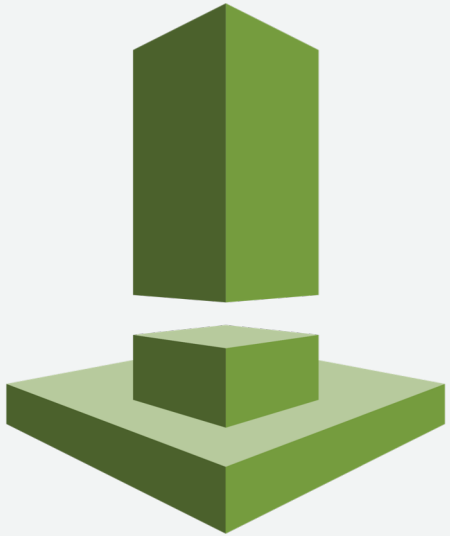
Bring Your Own Key

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

aws

Amazon Inspector

- Vulnerability Assessment Service
  - Built from the ground up to support DevSecOps
  - Automatable via APIs
  - Integrates with CI/CD tools
  - On-Demand Pricing model
  - Static & Dynamic Rules Packages
  - Generates Findings



AWS Web Application Firewall

Web Traffic Filtering with Custom Rules

Create custom rules that can block, allow or monitor requests based on IP address, HTTP headers, or a combination of both.

Malicious Request Blocking

AWS WAF can recognize and block common web application security risks like SQL injection (SQLi) and cross-site scripting (XSS).

Active monitoring & tuning

Monitor and configure the requests that are being blocked and allowed by the Web ACL rules.

Auditing made easy with CloudTrail

Who made the API call?

When was the API call made?

What was the API call?

What were the resources that were acted up on in the API call?

Where was the API call made from?

AWS CloudTrail

Web service that records AWS API calls for your account and delivers logs.

Who?	When?	What?	Where to?	Where from?
Bill	3:27pm	Launch Instance	us-west-2	72.21.198.64
Alice	8:19am	Added Bob to admin group	us-east-1	127.0.0.1
Steve	2:22pm	Deleted DynamoDB table	eu-west-1	205.251.233.176


```
{
"Records": [
        {
        "eventVersion": "1.0",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/Alice",
            "accountId": "123456789012",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice",
            "sessionContext": {
                "attributes": {
                    "mfaAuthenticated": "false",
                    "creationDate": "2014-03-25T18:45:11Z"
                }
            }
        },
        "eventTime": "2014-03-25T21:08:14Z",
        "eventSource": "iam.amazonaws.com",
        "eventName": "AddUserToGroup",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "AWSConsole",
        "requestParameters": {
            "userName": "Bob",
            "groupName": "admin"
        },
        "responseElements": null
    },
    ...additional entries
  ]
}
```

AWS CloudWatch

Monitoring services for AWS Resources and AWS-based Applications.

What does it do?

- Collect and Track Metrics
- Monitor and Store Logs
- Set Alarms (react to changes)
- View Graphs and Statistics

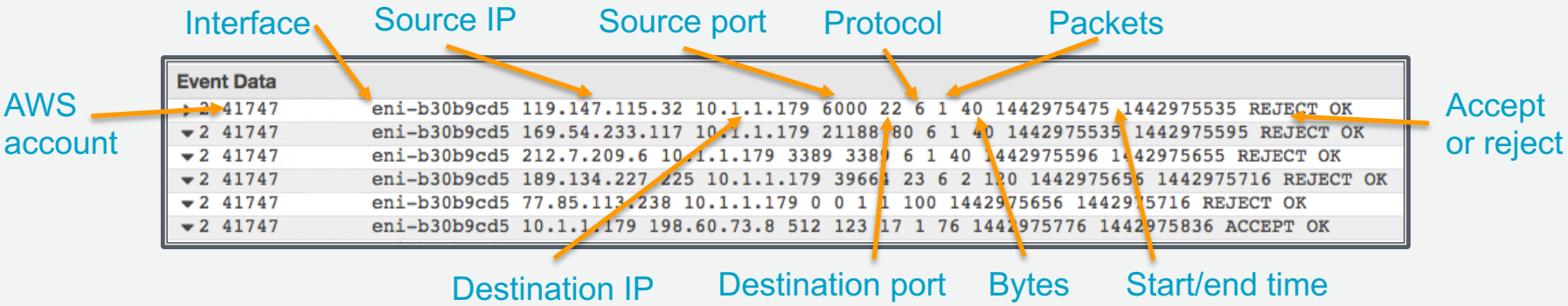
How can you use it?

Monitor CPU, Memory, Disk I/O, Network, etc.	← CloudWatch Metrics
React to application log events and availability	← CloudWatch Logs / CloudWatch Events
Automatically scale EC2 instance fleet	← CloudWatch Alarms
View Operational Status and Identify Issues	← CloudWatch Dashboards

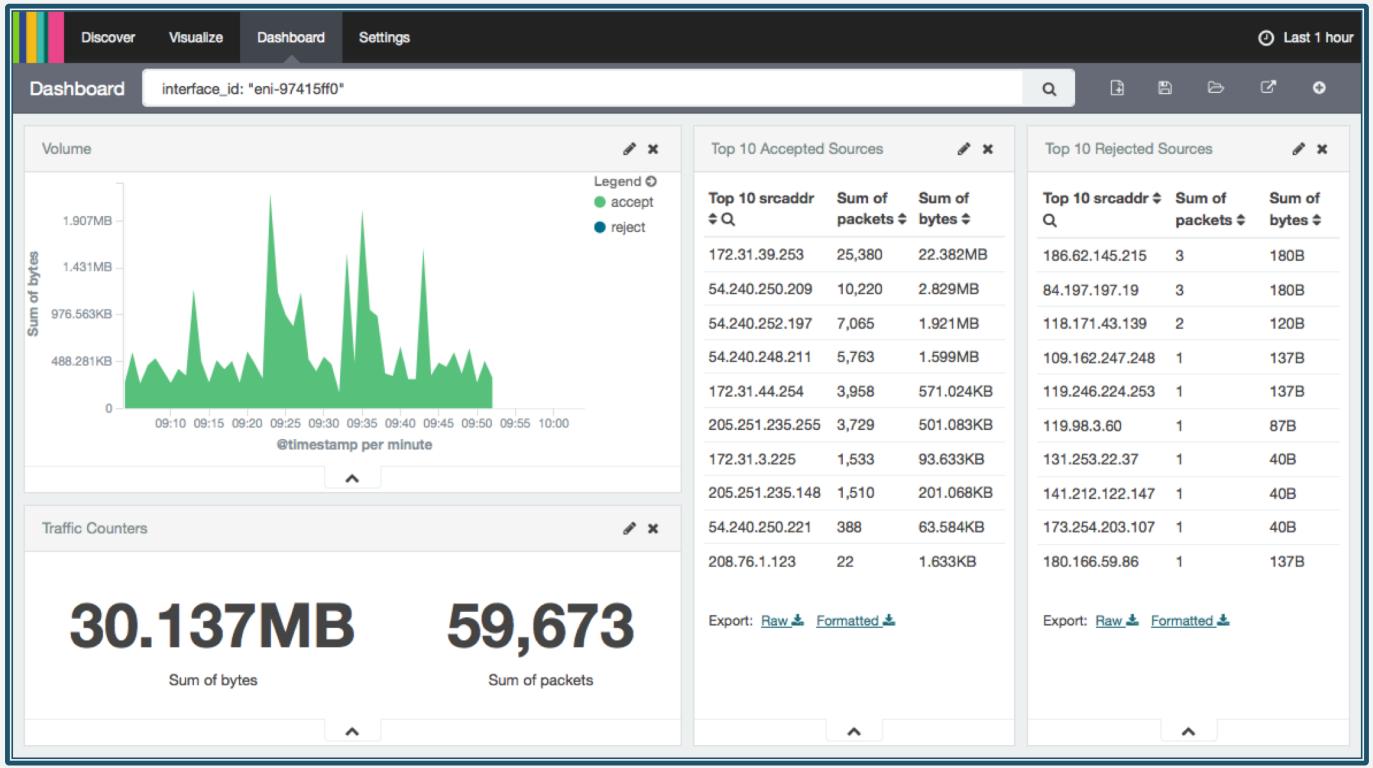
aws

VPC Flow Logs

- Agentless
- Enable per ENI, per subnet, or per VPC
- Logged to AWS CloudWatch Logs
- Create CloudWatch metrics from log data
- Alarm on those metrics



VPC Flow Logs

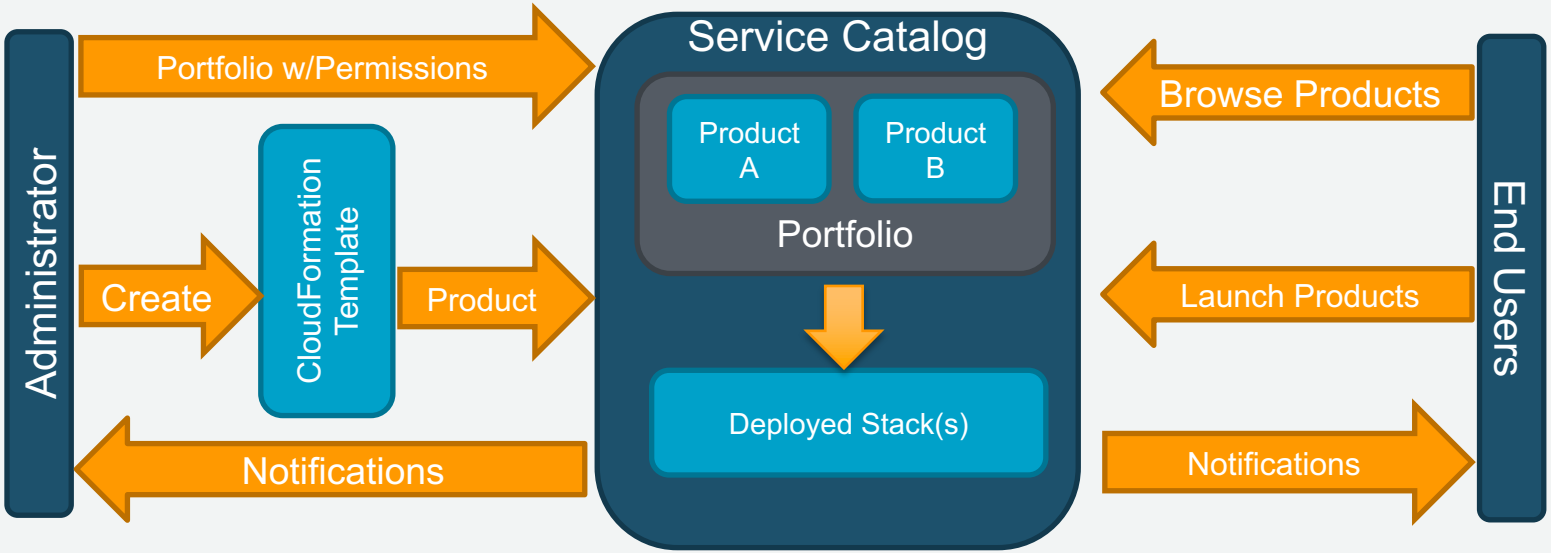


- Amazon Elasticsearch Service
- Amazon CloudWatch Logs subscriptions

aws

AWS Service Catalog

Self-service portal for creating and managing resources in AWS.

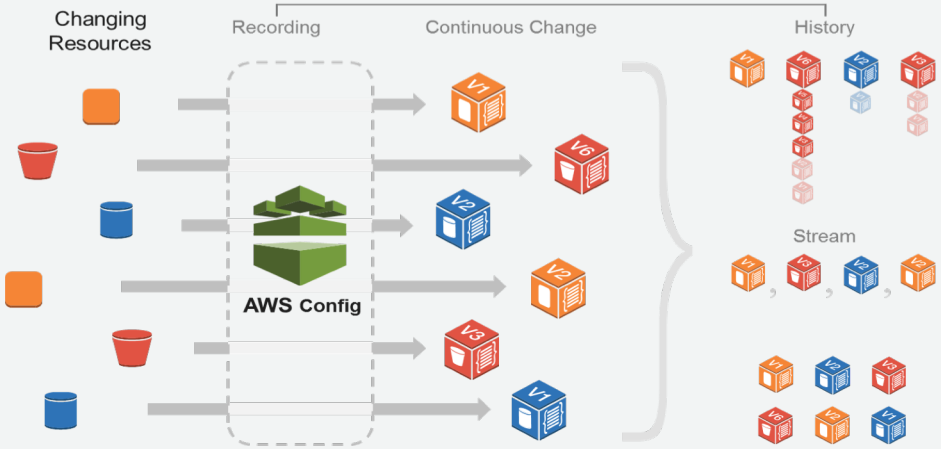


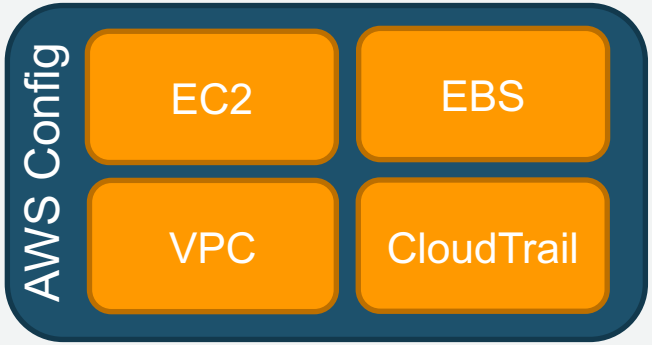
- Create and manage approved catalogs of resources.
- End users browse and launch products via self-service portal.
- Control user access to applications or AWS resources per compliance needs.
- Extensible via API to existing self-service frameworks.

aws

AWS Config

Managed service for tracking AWS inventory and configuration, and configuration change notification.





Security Analysis

Audit Compliance

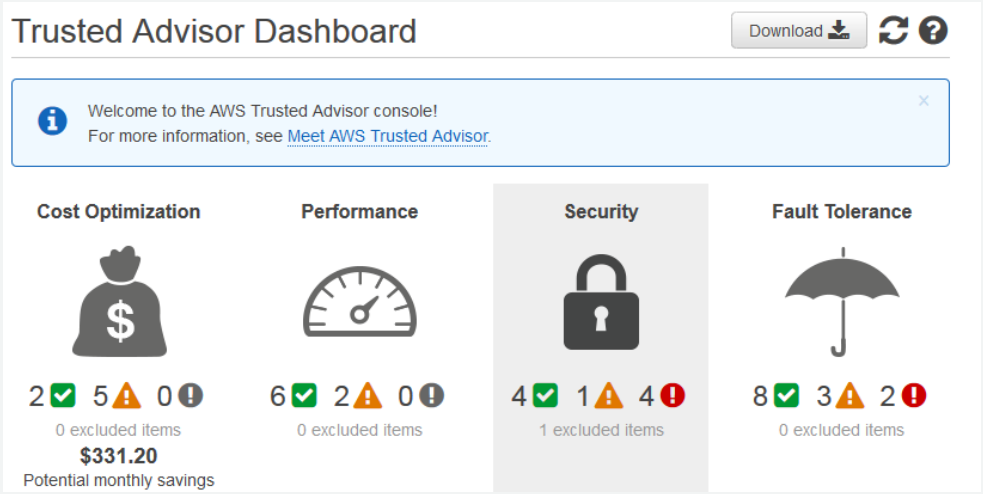
Change Management

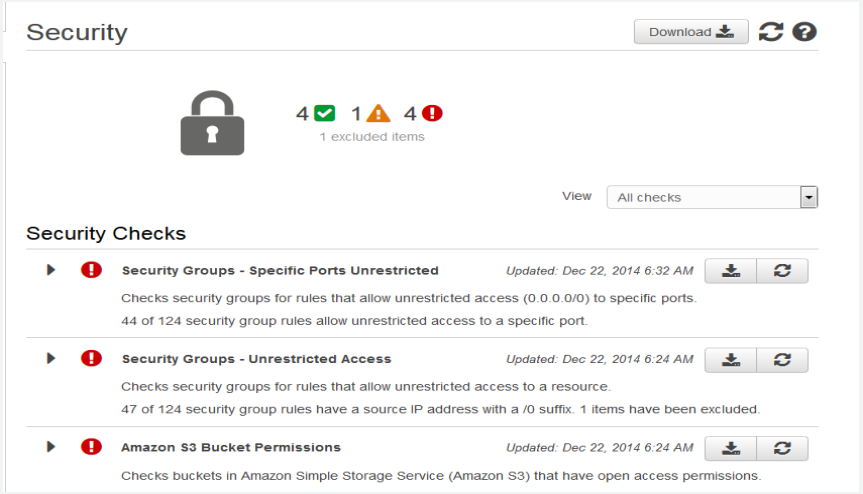
Troubleshooting

Discovery

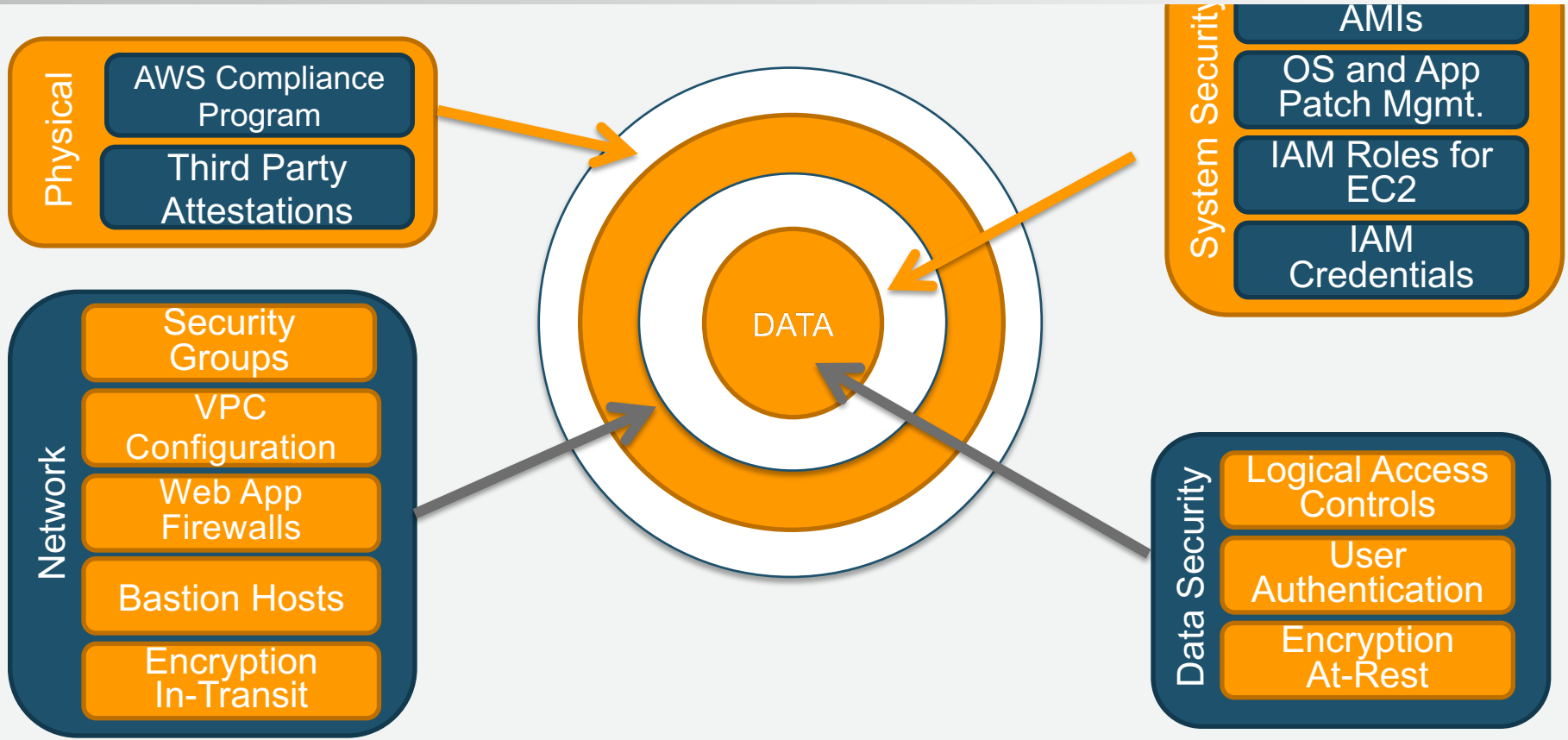
AWS Trusted Advisor

Leverage Trusted Advisor to analyze your AWS resources for best practices for availability, cost, performance and security.

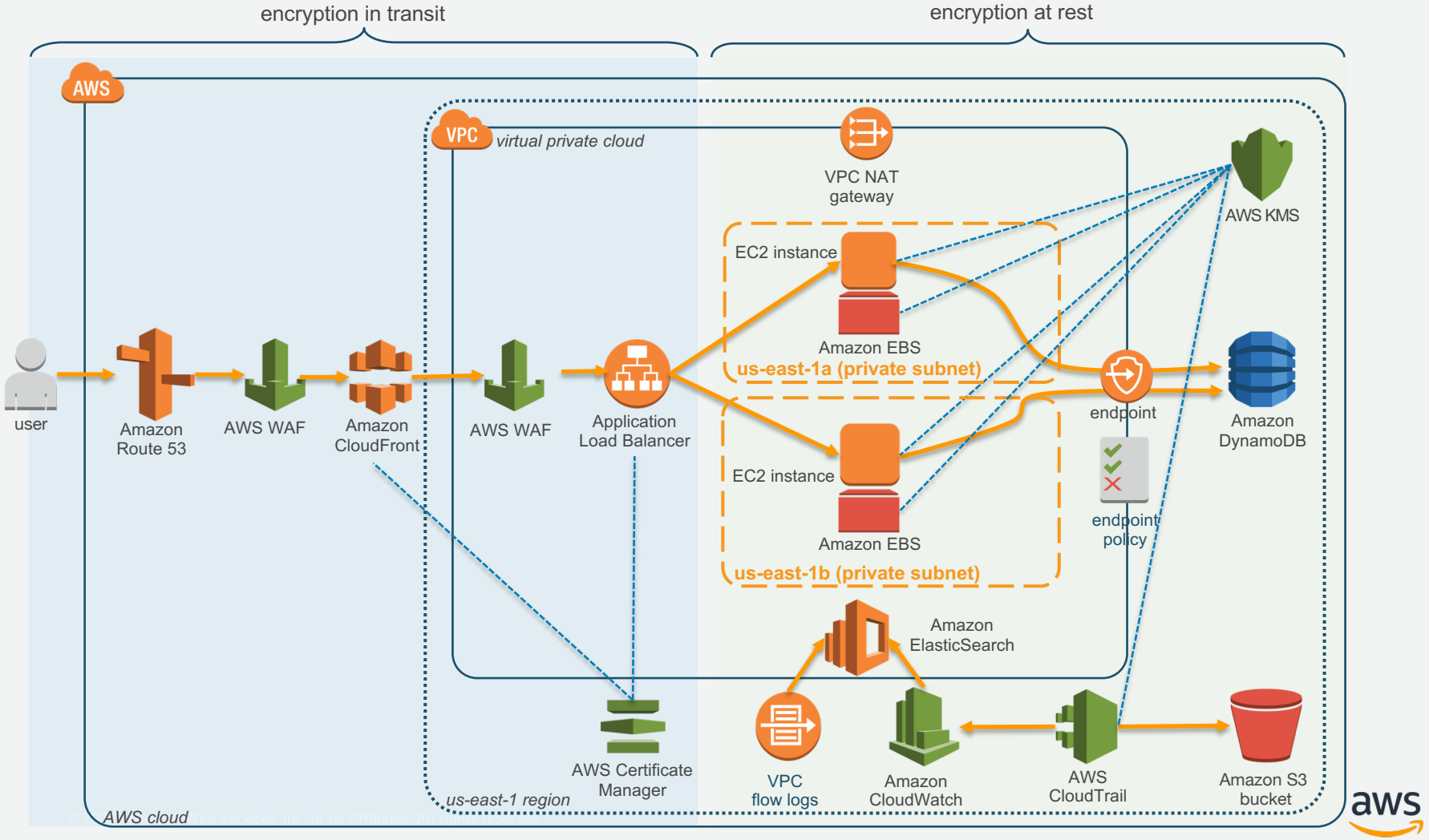




Defense-in-Depth



aws


encryption in transit
encryption at rest
AWS
VPC
virtual private cloud
VPC NAT gateway
AWS KMS
EC2 instance
Amazon EBS
us-east-1a (private subnet)
user
Amazon Route 53
AWS WAF
Amazon CloudFront
AWS WAF
Application Load Balancer
endpoint
Amazon DynamoDB
EC2 instance
Amazon EBS
us-east-1b (private subnet)
endpoint policy
Amazon ElasticSearch
AWS Certificate Manager
VPC flow logs
Amazon CloudWatch
AWS CloudTrail
Amazon S3 bucket
us-east-1 region
AWS cloud

Largest ecosystem of security partners and solutions

Infrastructure security

Infrastructure security

Identity & access control

Data protection

Configuration & vulnerability analysis

Logging & monitoring

aws

Consulting competency partners with demonstrated expertise

Security engineering	Security engineering	Governance, risk & compliance	Security operations & automation
8K Miles	Deloitte.	Booz | Allen | Hamilton	CLOUDREACH
accenture High performance. Delivered.	GuidePoint SECURITY LEADING SECURITY INNOVATION	stackArmor	logicworks
AllCloud	logicworks	REĀN CLOUD	
CLOUDZONE by matrix	NRI 未来創発 Dream up the future.		
ctp Cloud Technology Partners a Hewlett Packard Enterprise Company	REĀN CLOUD		

aws

AWS Security Center

Comprehensive security portal to provide a variety of security notifications, information and documentation.

http://aws.amazon.com/security

Security Whitepapers

- Overview of Security Process
- AWS Risk and Compliance
- AWS Security Best Practices

Security Bulletin

Security Resources

Vulnerability Reporting

Penetration Testing

Requests

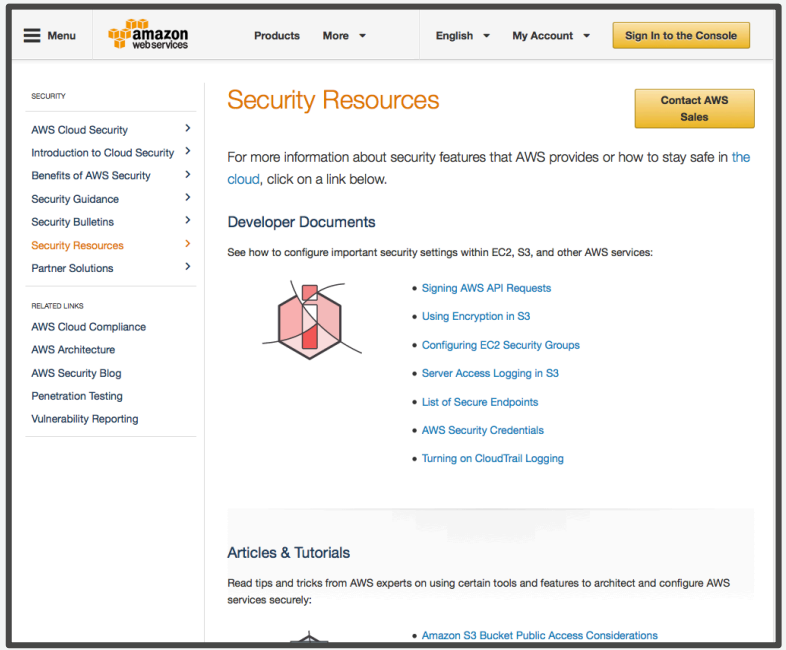
Report Suspicious Emails

Documentation and Blogs

Security Resources

http://aws.amazon.com/security/security-resources/

Developer Information, Articles and Tutorials, Security Products, and Whitepapers



AWS Security Blog

http://blogs.aws.amazon.com/security/

Subscribe to the blog – it's a great way to stay up-to-date on AWS security and compliance.

Video Resources

- The AWS Shared Responsibility Model in Detail https://youtu.be/RwUSPklR24M
- IAM Recommended Practices https://youtu.be/R-PyVnhxx-U
- Encryption Options on AWS https://youtu.be/9bn7p2tdym0
- Compliance, Logging, Analysis and Alerting https://youtu.be/42-1xpT-s6U
- Securing Serverless Architectures https://www.youtube.com/watch?v=lKVp8d45HSU
- Account Separation and Mandatory Access Control https://www.youtube.com/watch?v=bFMkxIAhFv8

Thank You!