CSAA Practise Test 6

What is AWS CloudFormation?

RSS

AWS CloudFormation is a service that helps you model and set up your AWS resources so that you can spend less time managing those resources and more time focusing on your applications that run in AWS. You create a template that describes all the AWS resources that you want (like Amazon EC2 instances or Amazon RDS DB instances), and CloudFormation takes care of provisioning and configuring those resources for you. You don't need to individually create and configure AWS resources and figure out what's dependent on what; CloudFormation handles that. The following scenarios demonstrate how CloudFormation can help.

Simplify infrastructure management

For a scalable web application that also includes a backend database, you might use an Auto Scaling group, an Elastic Load Balancing load balancer, and an Amazon Relational Database Service database instance. You might use each individual service to provision these resources and after you create the resources, you would have to configure them to work together. All these tasks can add complexity and time before you even get your application up and running.

Instead, you can create a CloudFormation template or modify an existing one. A *template* describes all your resources and their properties. When you use that template to create a CloudFormation stack, CloudFormation provisions the Auto Scaling group, load balancer, and database for you. After the stack has been successfully created, your AWS resources are up and running. You can delete the stack just as easily, which deletes all the resources in the stack. By using CloudFormation, you easily manage a collection of resources as a single unit.

Quickly replicate your infrastructure

If your application requires additional availability, you might replicate it in multiple regions so that if one region becomes unavailable, your users can still use your application in other regions. The challenge in replicating your application is that it also requires you to replicate your resources. Not only do you need to record all the resources that your application requires, but you must also provision and configure those resources in each region.

Reuse your CloudFormation template to create your resources in a consistent and repeatable manner. To reuse your template, describe your resources once and then provision the same resources over and over in multiple regions.

Easily control and track changes to your infrastructure

In some cases, you might have underlying resources that you want to upgrade incrementally. For example, you might change to a higher performing instance type in your Auto Scaling launch configuration so that you can reduce the maximum number of instances in your Auto Scaling group. If problems occur after you complete the update, you might need to roll back your infrastructure to the original settings. To do this manually, you not only have to remember which resources were changed, you also have to know what the original settings were.

When you provision your infrastructure with CloudFormation, the CloudFormation template describes exactly what resources are provisioned and their settings. Because these templates are text files, you simply track differences in your templates to track changes to your infrastructure, similar to the way developers control revisions to source code. For example, you can use a version control system with your templates so that you know exactly what changes were made, who made them, and when. If at any point you need to reverse changes to your infrastructure, you can use a previous version of your template.

```
#!/bin/bash

#update os
yum update -y
#install apache server
yum install -y httpd
# get private ip address of ec2 instance using instance metadata
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& PRIVATE_IP=`curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/local-ipv4`
# get public ip address of ec2 instance using instance metadata
TOKEN=`curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"` \
&& PUBLIC_IP=`curl -H "X-aws-ec2-metadata-token: $TOKEN" http://169.254.169.254/latest/meta-data/public-ipv4`
# get date and time of server
DATE_TIME=`date`
# set all permissions
chmod -R 777 /var/www/html
# create a custom index.html file
echo "<html>
<head>
    <title> Congratulations! You have created an instance from Launch Template</title>
</head>
<body>
    <h1>This web server is launched from launch template by YOUR_NAME</h1>
    <p>This instance is created at <b>$DATE_TIME</b></p>
    <p>Private IP address of this instance is <b>$PRIVATE_IP</b></p>
    <p>Public IP address of this instance is <b>$PUBLIC_IP</b></p>
</body>
</html>" > /var/www/html/index.html
# start apache server
systemctl start httpd
systemctl enable httpd
```

```yaml
Resources:

MyALB:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Name: Aslancfnelb
    SecurityGroups:
      - !GetAtt MySecGrp.GroupId
    Subnets: !Ref Subnets
    Type: application
Outputs:
  ShowDNS:
    Description: DNS of ALB
    Value: !Join
      - ''
      - - 'http://'
        - !GetAtt MyALB.DNSName
```

YAML

JSON

```json
{
  "Resources": {
    "MyEC2Instance": {
      "Type": "AWS::EC2::Instance",
      "Properties": {
        "ImageId": "ami-d6f32ab5"
      }
    }
  },
  "Outputs": {
    "Availability": {
      "Description": "The Instance ID",
      "Value":
      { "Fn::GetAtt" : [ "MyEC2Instance", "AvailabilityZone" ]}
    }
  }
}
```

Create volume Info

Create an Amazon EBS volume to attach to any EC2 instance in the same Availability Zone.

Volume settings

Volume type Info

General Purpose SSD (gp2)

Size (GiB) Info

100

Min: 1 GiB, Max: 16384 GiB. The value must be an integer.

IOPS Info

300 / 3000

Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS.

Throughput (MiB/s) Info

Not applicable

Availability Zone Info

us-east-1a

Snapshot ID - *optional* Info

Don't create volume from a snapshot

Encryption Info

Use Amazon EBS encryption as an encryption solution for your EBS resources associated with your EC2 instances.

Encrypt this volume

How EBS encryption works

You can encrypt both the boot and data volumes of an EC2 instance.

When you create an encrypted EBS volume and attach it to a supported instance type, the following types of data are encrypted:

- Data at rest inside the volume
- All data moving between the volume and the instance
- All snapshots created from the volume
- All volumes created from those snapshots

S3

Default encryption Info

Server-side encryption is automatically applied to new objects stored in this bucket.

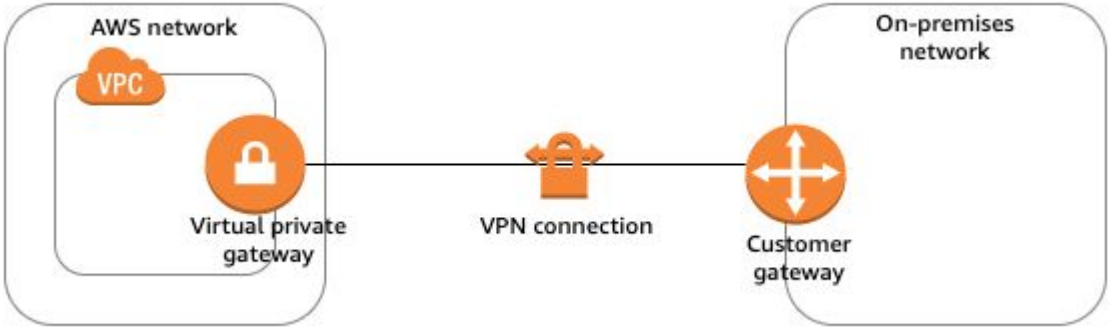
Encryption key type Info

- (•) Amazon S3 managed keys (SSE-S3)
- ( ) AWS Key Management Service key (SSE-KMS)

Bucket Key

When KMS encryption is used to encrypt new objects in this bucket, the bucket key reduces encryption costs by lowering calls to AWS KMS. Learn more

- ( ) Disable
- (•) Enable



Item	Information
Customer gateway device	The physical or software device on your side of the VPN connection. You need the vendor (for example, Cisco), platform (for example, ISR Series Routers), and software version (for example, IOS 12.4).
Customer gateway	To create the customer gateway resource in AWS, you need the following information: • The internet-routable IP address for the device's external interface • The type of routing: static or dynamic • For dynamic routing, the Border Gateway Protocol (BGP) Autonomous System Number (ASN) • (Optional) Private certificate from AWS Private Certificate Authority to authenticate your VPN For more information, see Customer gateway options for your Site-to-Site VPN connection.
(Optional) The ASN for the AWS side of the BGP session	You specify this when you create a virtual private gateway or transit gateway. If you do not specify a value, the default ASN applies. For more information, see Virtual private gateway.
VPN connection	To create the VPN connection, you need the following information: • For static routing, the IP prefixes for your private network. • (Optional) Tunnel options for each VPN tunnel. For more information, see Tunnel options for your Site-to-Site VPN connection.

Model	vCPU	Memory (GiB)	Instance Storage (GB)	Network Bandwidth (Gbps)***	EBS Bandwidth (Mbps)
c5.large	2	4	EBS-Only	Up to 10	Up to 4,750
c5.xlarge	4	8	EBS-Only	Up to 10	Up to 4,750
c5.2xlarge	8	16	EBS-Only	Up to 10	Up to 4,750
c5.4xlarge	16	32	EBS-Only	Up to 10	4,750
c5.9xlarge	36	72	EBS-Only	10	9,500
c5.12xlarge	48	96	EBS-Only	12	9,500
c5.18xlarge	72	144	EBS-Only	25	19,000
c5.24xlarge	96	192	EBS-Only	25	19,000
c5.metal	96	192	EBS-Only	25	19,000
c5d.large	2	4	1 x 50 NVMe SSD	Up to 10	Up to 4,750
c5d.xlarge	4	8	1 x 100 NVMe SSD	Up to 10	Up to 4,750
c5d.2xlarge	8	16	1 x 200 NVMe SSD	Up to 10	Up to 4,750
c5d.4xlarge	16	32	1 x 400 NVMe SSD	Up to 10	4,750
c5d.9xlarge	36	72	1 x 900 NVMe SSD	10	9,500
c5d.12xlarge	48	96	2 x 900 NVMe SSD	12	9,500
c5d.18xlarge	72	144	2 x 900 NVMe SSD	25	19,000
c5d.24xlarge	96	192	4 x 900 NVMe SSD	25	19,000
c5d.metal	96	192	4 x 900 NVMe SSD	25	19,000

- **Larger Instance for better process**
- **Provisioned IOPS/SSD for better transactional workloads**

▼ Additional configuration

1

Database options, failover, backup enabled, backtrack disabled, Performance Insights enabled, Enhanced Monitoring enabled, maintenance, CloudWatch Logs, delete protection enabled

Database options

DB parameter group Info

default.mysql8.0

Option group Info

default:mysql-8-0

Backup

Enable automated backups

Creates a point-in-time snapshot of your database

⚠ Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details here.

Backup retention period Info

Choose the number of days that RDS should retain automatic backups for this instance.

7 days

Backup window Info

Launch DB Instance

2

You are creating a new DB instance from a source DB instance at a specified time. This new DB instance will have the default DB security group and DB parameter groups.

Restore time

Point in time to restore from

Latest restorable time

May 8, 2020 at 12:40:01 PM UTC+8

Custom

Specify a custom date and time to restore from

Custom Date: May 8, 2020

Custom Time: 06 : 06 : 00 UTC+8

00
01
02
03
04
05
06
07
08
09
10

Instance specifications

DB engine

Name of the database engine to be used for this instance

PostgreSQL

License model

License type associated with the database engine

postgresql-license

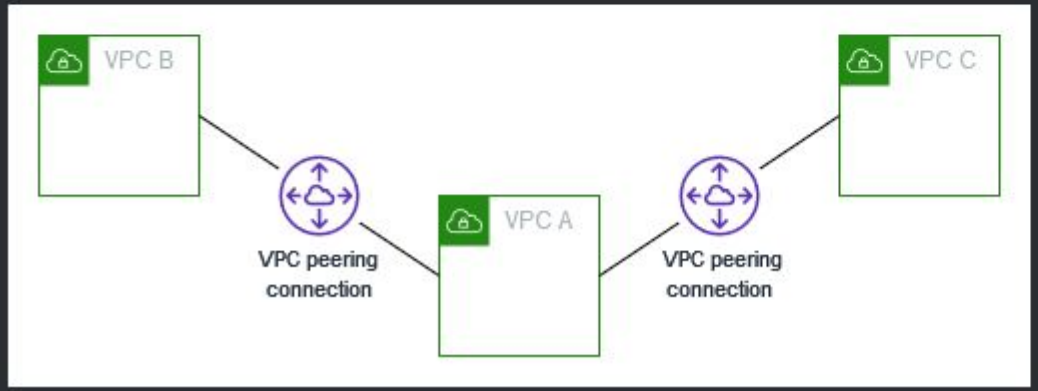
DB instance class

Contains the compute and memory capacity of the DB instance.

Multiple VPC peering connections

A VPC peering connection is a one to one relationship between two VPCs. You can create multiple VPC peering connections for each VPC that you own, but transitive peering relationships are not supported. You do not have any peering relationship with VPCs that your VPC is not directly peered with.

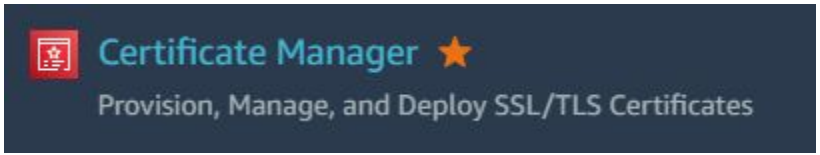
The following diagram is an example of one VPC peered to two different VPCs. There are two VPC peering connections: VPC A is peered with both VPC B and VPC C. VPC B and VPC C are not peered, and you cannot use VPC A as a transit point for peering between VPC B and VPC C. If you want to enable routing of traffic between VPC B and VPC C, you must create a unique VPC peering connection between them.



Important

You cannot install certificates with RSA keys larger than 2048-bit or EC keys on your Network Load Balancer.

To use a TLS listener, you must deploy at least one server certificate on your load balancer. The load balancer uses a server certificate to terminate the front-end connection and then to decrypt requests from clients before sending them to the targets. Note that if you need to pass encrypted traffic to the targets without the load balancer decrypting it, create a TCP listener on port 443 instead of creating a TLS listener. The load balancer passes the request to the target as is, without decrypting it.



Benefits and features

Free public certificates for ACM-integrated services

With AWS Certificate Manager, there is no additional charge for provisioning public or private SSL/TLS certificates you use with ACM-integrated services, such as Elastic Load Balancing and API Gateway. You pay for the AWS resources you create to run your application. For private certificates, AWS Private CA provides you the ability to pay monthly for the service and certificates you create. You pay less per certificate as you create more private certificates. Learn more

Managed certificate renewal

AWS Certificate Manager manages the renewal process for the certificates managed in ACM and used with ACM-integrated services, such as Elastic Load Balancing and API Gateway. ACM can automate renewal and deployment of these certificates. With AWS Private CA APIs, ACM enables you to automate creation and renewal of private certificates for on-premises resources, EC2 instances, and IoT devices.

Get certificates easily

AWS Certificate Manager removes many of the time-consuming and error-prone steps to acquire an SSL/TLS certificate for your website or application. There is no need to generate a key pair or certificate signing request (CSR), submit a CSR to a certificate authority, or upload and install the certificate once received. With a few clicks in the AWS Management Console, you can request a trusted SSL/TLS certificate from AWS. Once the certificate is created, AWS Certificate Manager takes care of deploying certificates to help you enable SSL/TLS for your website or application.

S3 cross-region replication New

Automated, fast, and reliable asynchronous replication of data across AWS regions

Use cases

Compliance - store data hundreds of miles apart

Lower latency - distribute data to regional customers)

Security - create remote replicas managed by separate AWS accounts

- Only replicates new PUTs. Once S3 is configured, all new uploads into a source bucket will be replicated
- Entire bucket or prefix based
- 1:1 replication between any 2 regions
- Versioning required

Details on Cross-Region Replication

Versioning - Need to enable S3 versioning for the source and destination buckets.

Lifecycle Rules - You can choose to use Lifecyle Rules on the destination bucket to manage older versions by deleting them or migrating them to Amazon Glacier.

Determining Replication Status - Use the HEAD operation on a source object to determine its replication status.

Region-to-Region - Replication always takes place between a pair of AWS regions. You cannot use this feature to replicate content to two buckets that are in the same region.

New Objects - Replicates new objects and changes to existing objects. Use S3 COPY to replicate existing objects

```
Metric      : change "period" to 1 minute and keep remaining as default
Conditions  :
  - Treshold Type                 : Static
  - Whenever CPUUtilization is... : Greater
  - than...                       : 60
```

```
 Notification:
   - Alarm state trigger : In alarm
   - Select an SNS topic :
     - Create new topic
       - Create a new topic… : Clarus-alarm
       - Email endpoints that will receive the notification…
 : <your email adress>
     - create topic

 EC2 action
   - Alarm state trigger
     - In alarm ---> Select "Stop Instance"
```

Making requests to the Amazon EC2 API

PDF

We provide the Query API for Amazon EC2, as well as software development kits (SDK) for AWS that enable you to access Amazon EC2 from your preferred programming language.

To monitor the calls made to the Amazon EC2 API for your account, including calls made by the AWS Management Console, command line tools, and other services, use AWS CloudTrail. For more information, see the AWS CloudTrail User Guide.

Contents

Required knowledge

If you plan to access Amazon EC2 through an API, you should be familiar with the following:

- XML
- Web services
- HTTP requests
- One or more programming languages, such as Java, PHP, Perl, Python, Ruby, C#, or C++.

Available APIs for Amazon EC2

The Amazon EC2 Query API provides HTTP or HTTPS requests that use the HTTP verb GET or POST and a Query parameter named `Action`.

AWS provides libraries, sample code, tutorials, and other resources for software developers who prefer to build applications using language-specific APIs instead of submitting a request over HTTP or HTTPS. These libraries provide basic functions that automatically take care of tasks such as cryptographically signing your requests, retrying requests, and handling error responses, so that it is easier for you to get started.

Description | **Inbound** | Outbound | Tags

Edit

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	
HTTPS	TCP	443	0.0.0.0/0	

Security group rules

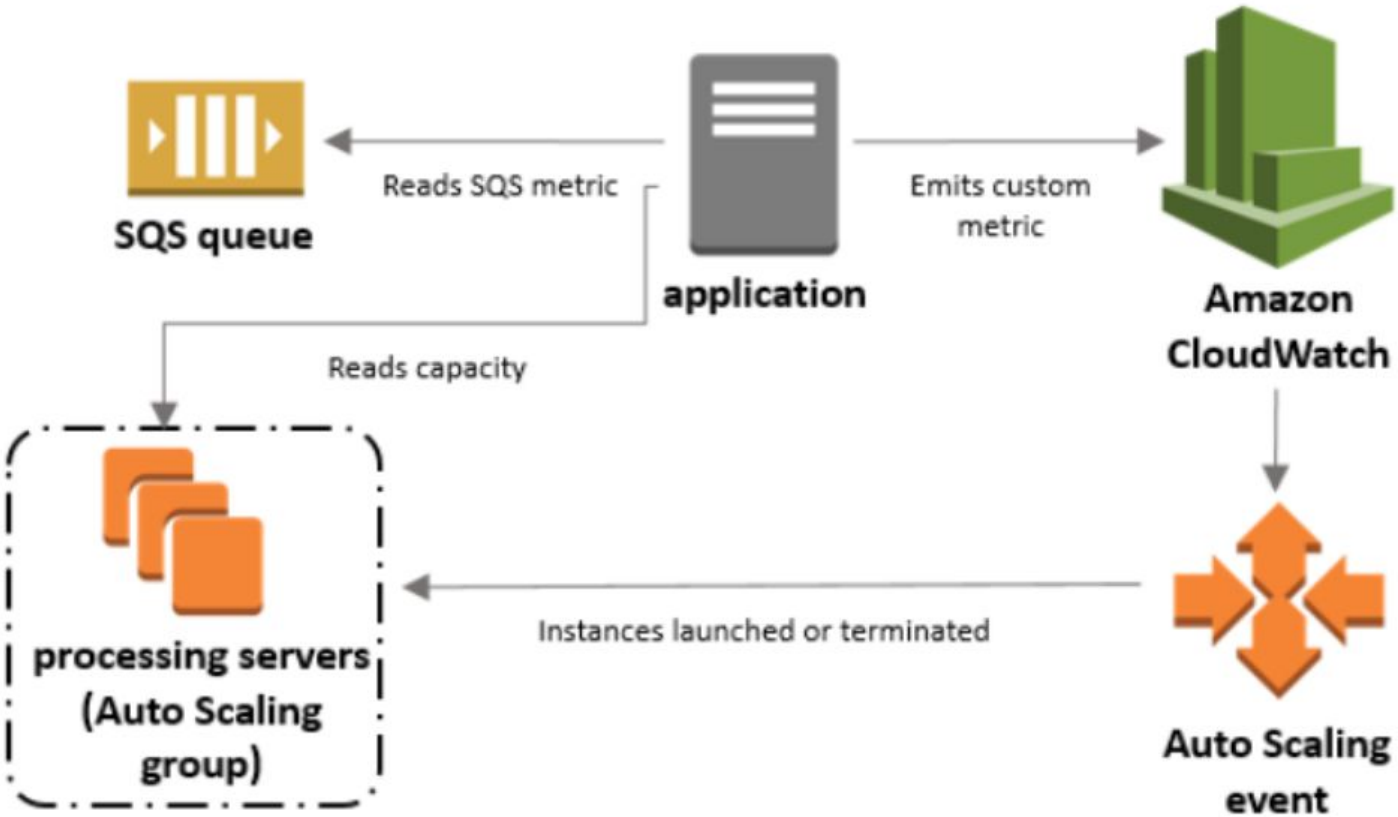
For HTTP traffic, add an inbound rule on port 80 from the source address 0.0.0.0/0.

For HTTPS traffic, add an inbound rule on port 443 from the source address 0.0.0.0/0.

These inbound rules allow traffic from IPv4 addresses. To allow IPv6 traffic, add inbound rules on the same ports from the source address ::/0. For more information on creating or modifying security groups, see Control traffic to resources using security groups.

Security groups are stateful, so the return traffic from the instance to users is allowed automatically. You don't need to modify the security group's outbound rules.

Note: The following example shows the security group rules for allowing IPv4 and IPv6 traffic on TCP port 80 (HTTP) and 443 (HTTPS). Determine if other sources of traffic, such as SSH or RDP to log in to the instance, must be allowed for your use case. Then, make sure that your SG has the relevant inbound rules to allow the needed traffic.


SQS queue
Reads SQS metric
application
Emits custom metric
Amazon CloudWatch
Reads capacity
processing servers (Auto Scaling group)
Instances launched or terminated
Auto Scaling event

Giving Aurora Access to Lambda

Before you can invoke Lambda functions from an Aurora MySQL, you must first give your Aurora MySQL DB cluster permission to access Lambda.

To give Aurora MySQL access to Lambda

1. Create an AWS Identity and Access Management (IAM) policy that provides the permissions that allow your Aurora MySQL DB cluster to invoke Lambda functions. For instructions, see Creating an IAM Policy to Access AWS Lambda Resources.

2. Create an IAM role, and attach the IAM policy you created in Creating an IAM Policy to Access AWS Lambda Resources to the new IAM role. For instructions, see Creating an IAM Role to Allow Amazon Aurora to Access AWS Services.

3. Set the `aws_default_lambda_role` DB cluster parameter to the Amazon Resource Name (ARN) of the new IAM role.

   For more information about DB cluster parameters, see Amazon Aurora DB Cluster and DB Instance Parameters.

4. To permit database users in an Aurora MySQL DB cluster to invoke Lambda functions, associate the role that you created in Creating an IAM Role to Allow Amazon Aurora to Access AWS Services with the DB cluster. For information about associating an IAM role with a DB cluster, see Associating an IAM Role with an Amazon Aurora MySQL DB Cluster.

5. Configure your Aurora MySQL DB cluster to allow outbound connections to Lambda. For instructions, see Enabling Network Communication from Amazon Aurora MySQL to Other AWS Services.

Database Options

Database Name

Note: if no database name is specified then no initial MySQL database will be created on the DB Instance.

Database Port 3306

DB Parameter Group default.mysql5.6

Option Group default:mysql-5-6

Enable Encryption Yes

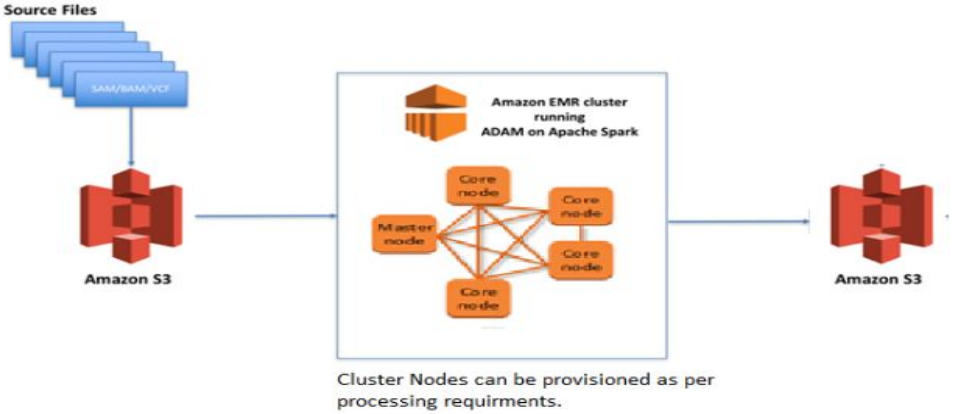
Master Key (default) aws/rds

Description Default master key that protects my RDS database volumes when no other key is defined

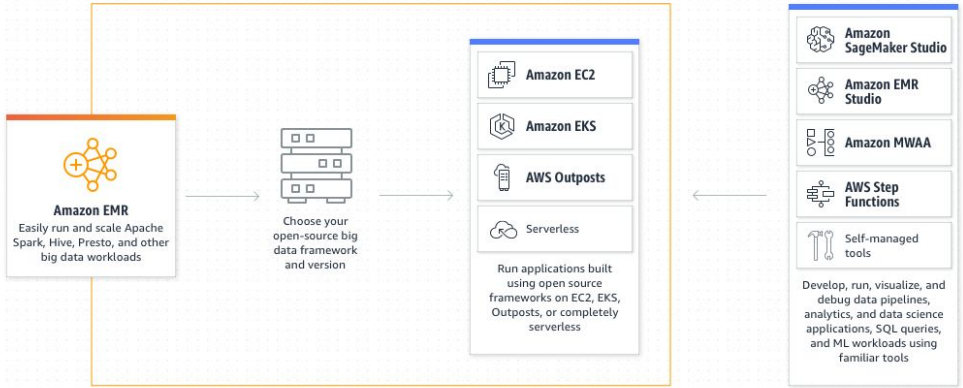
Account This account ( )

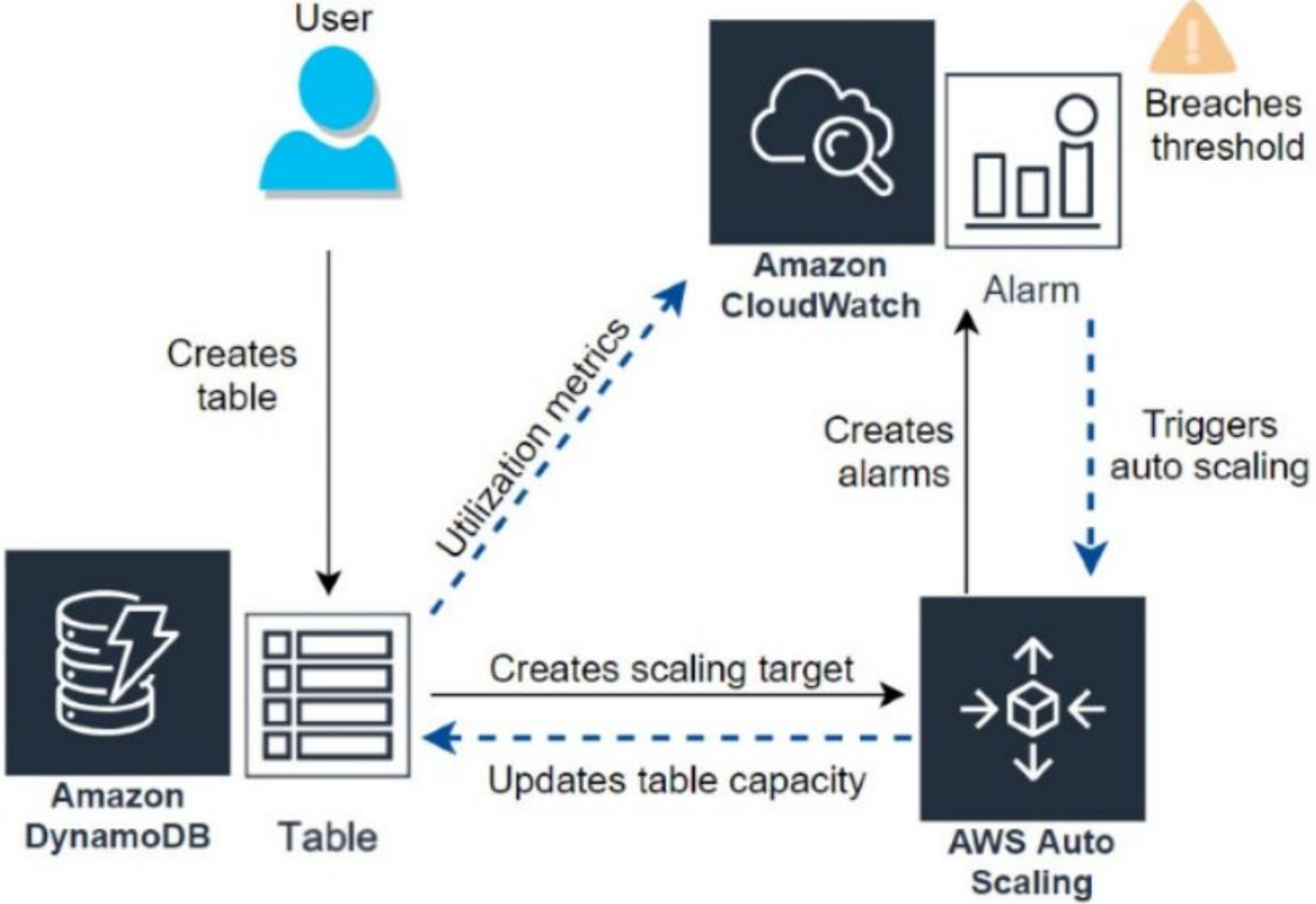
KMS Key ID alias/aws/rds

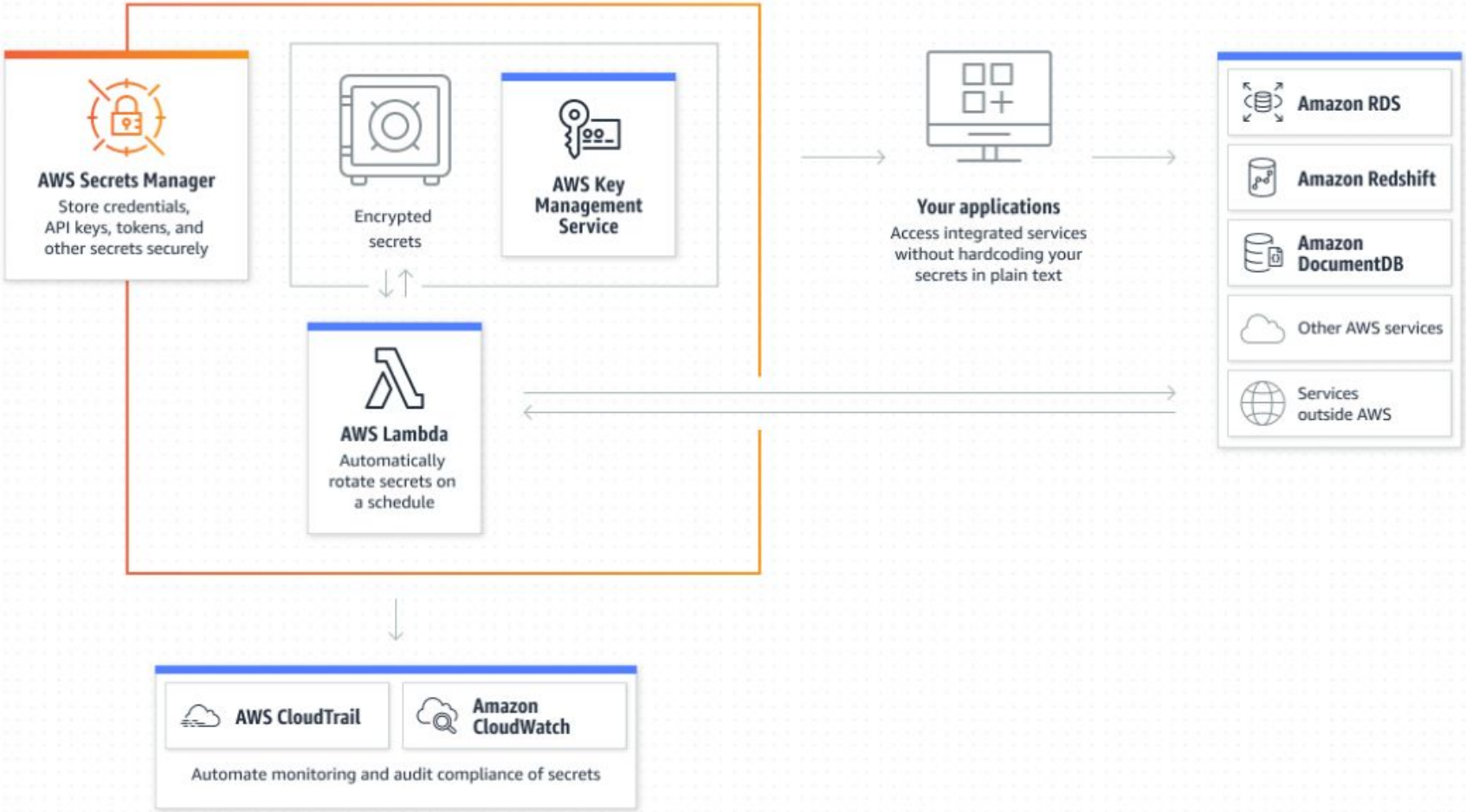
Select to encrypt the given instance. Master key ids and aliases appear in the list after they have been created using the Key Management Service(KMS) console.



EMR can be used to produce reliable and cost effective solutions by decoupling compute and storage. For compute, clusters can be launched per processing required and stopped when there is no requirement.




User
Amazon CloudWatch
Alarm
Breaches threshold
Creates table
Utilization metrics
Creates alarms
Triggers auto scaling
Amazon DynamoDB
Table
Creates scaling target
Updates table capacity
AWS Auto Scaling


AWS Secrets Manager
Store credentials, API keys, tokens, and other secrets securely
Encrypted secrets
AWS Key Management Service
AWS Lambda
Automatically rotate secrets on a schedule
Your applications
Access integrated services without hardcoding your secrets in plain text
Amazon RDS
Amazon Redshift
Amazon DocumentDB
Other AWS services
Services outside AWS
AWS CloudTrail
Amazon CloudWatch
Automate monitoring and audit compliance of secrets

AWS Kinesis Data Streams is the real-time data streaming service in Amazon Kinesis with high scalability and durability. It can help in continuously capturing multiple gigabytes of data every second from multiple sources. The higher customizability with Kinesis Data Streams is also one of the profound highlights.

As a matter of fact, it is the ideal choice for developers involved in developing custom applications or streaming data according to special needs. On the other hand, the benefits of customizability come at the price of manual provisioning and scaling. Generally, data is set up for 24 hours of availability in a stream while also ensuring that users could achieve data availability for almost 7 days.

AWS Kinesis Data Firehose provides the facility of loading data streams into AWS data stores. Kinesis Data Firehose provides the simplest approach for capturing, transforming, and loading data streams into AWS data stores.

The automatic management of scaling in the range of gigabytes per second, along with support for batching, encryption, and compression of streaming data, are also some crucial features in Amazon Kinesis Data Firehose. Firehose also helps in streaming to RedShift, S3, or ElasticSearch service, to copy data for processing by using additional services.

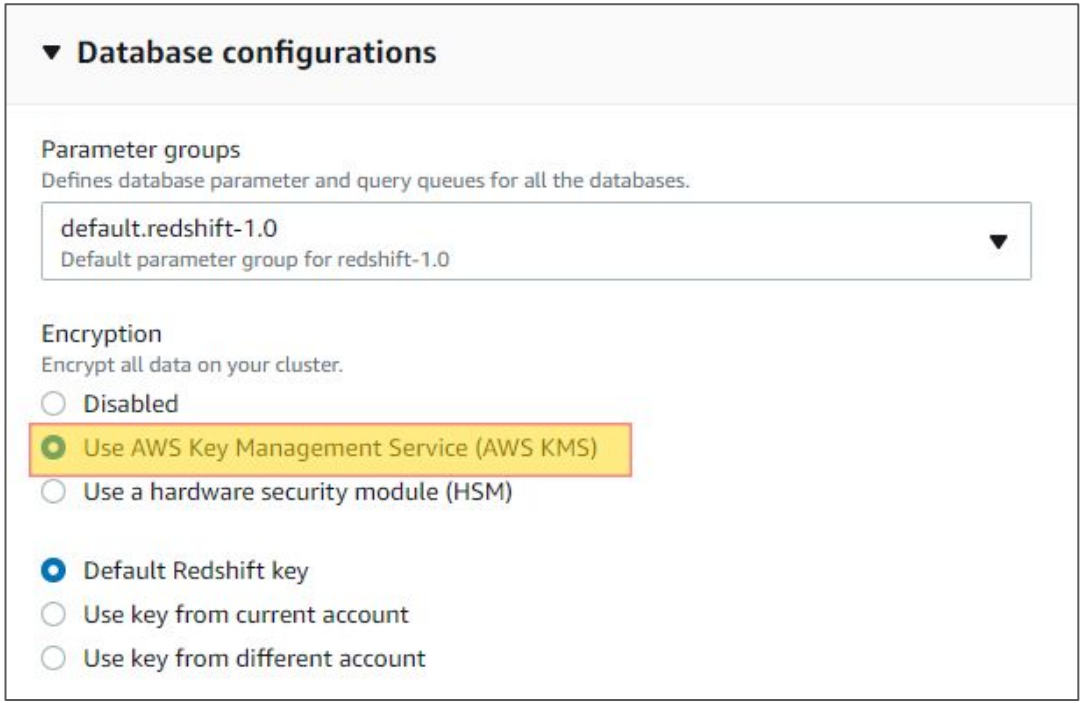
Scheduled scaling for Application Auto Scaling

PDF | RSS

Scaling based on a schedule allows you to set your own scaling schedule according to predictable load changes. For example, let's say that every week the traffic to your web application starts to increase on Wednesday, remains high on Thursday, and starts to decrease on Friday. You can configure a schedule for Application Auto Scaling to increase capacity on Wednesday and decrease capacity on Friday.

To use scheduled scaling, create *scheduled actions*, which tell Application Auto Scaling to perform scaling activities at specific times. When you create a scheduled action, you specify the scalable target, when the scaling activity should occur, a minimum capacity, and a maximum capacity. You can create scheduled actions that scale one time only or that scale on a recurring schedule.

At the specified time, Application Auto Scaling scales based on the new capacity values, by comparing current capacity to the specified minimum and maximum capacity.


Database configurations
Parameter groups
Defines database parameter and query queues for all the databases.
default.redshift-1.0
Default parameter group for redshift-1.0
Encryption
Encrypt all data on your cluster.
Disabled
Use AWS Key Management Service (AWS KMS)
Use a hardware security module (HSM)
Default Redshift key
Use key from current account
Use key from different account


Amazon EKS
Create Kubernetes clusters (powered by Amazon EKS Distro)
AWS Fargate
Deploy serverless containers
Amazon EC2
Deploy worker nodes for your EKS cluster
Run Kubernetes apps
Amazon EKS dashboard in the AWS console
View and explore running Kubernetes apps


Without Fargate
Build your container image
Define and deploy the EC2 Instances
Provision and manage compute and memory resources
Isolate applications in separate VMs
Run and manage both applications and infrastructure
Pay for EC2 Instances
With Fargate
AWS Fargate
Build container image
Define memory and compute resources required
Run and manage applications
Pay for requested compute resources when used. Application isolation by design

AWS CloudTrail – Capture AWS API Activity

by Jeff Barr | on 13 NOV 2013 | in AWS CloudTrail | Permalink | Share

Do you have the need to track the API calls for one or more AWS accounts? If so, the new AWS CloudTrail service is for you.

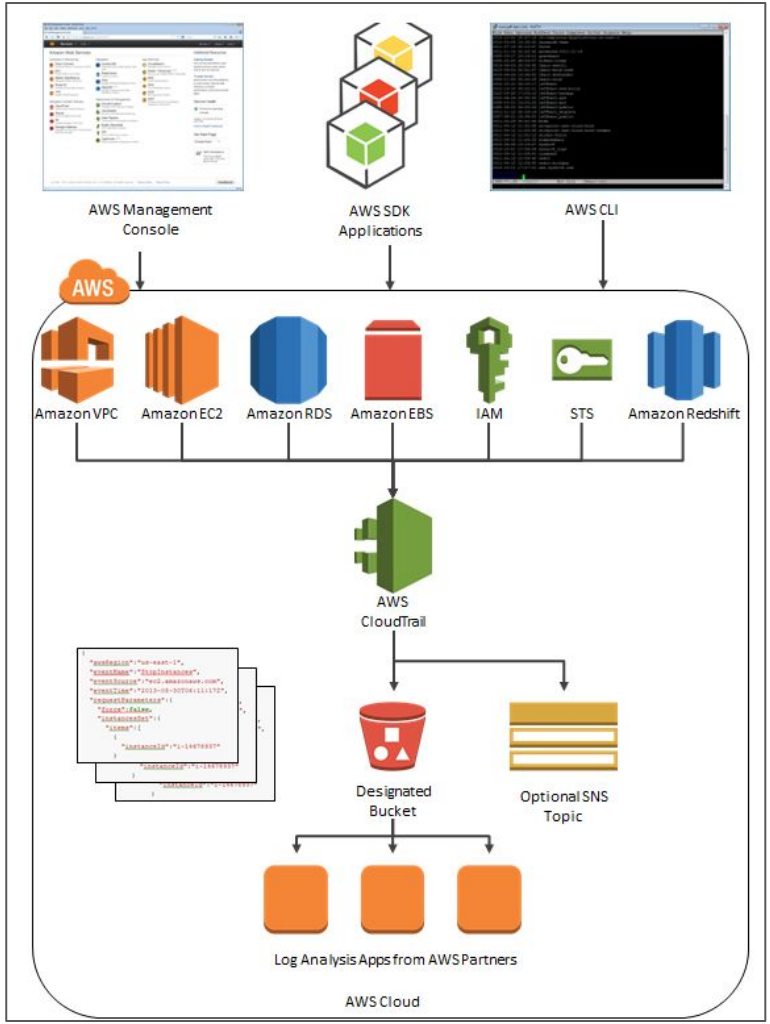
Once enabled, AWS CloudTrail records the calls made to the AWS APIs using the AWS Management Console, the AWS Command Line Interface (CLI), your own applications, and third-party software and publishes the resulting log files to the Amazon S3 bucket of your choice. CloudTrail can also issue a notification to an Amazon SNS topic of your choice each time a file is published. Each call is logged in JSON format for easy parsing and processing.

The API call history logged by CloudTrail is designed to support a wide variety of use cases. Here are some ideas to get you started:

- **Compliance Aid** – You have access to information needed to demonstrate that AWS resources were managed according to rules and regulatory standards.
- **Resource Life Cycle Tracking** -You can track an AWS resource from creation through deletion.
- **Operational Troubleshooting** – You can identify the most recent changes made to resources in your environment.
- **Security Analytics** – You can see which user activities failed due to inadequate permissions.

The data produced by CloudTrail can help you to answer questions such as:

- What actions did a given user take over a specific time period?
- For a given resource, which AWS user has taken action on it over a given time period?
- What is the source IP address of a particular activity?



151.0.0.0 /27 A

Input	Input IP	Input Long	Input Hex
151.0.0.0/27	151.0.0.0	2533359616	97.00.00.00

CIDR	CIDR IP Range	CIDR Long Range	CIDR Hex Range
151.0.0.0/27	151.0.0.0 - 151.0.0.31	2533359616 - 2533359647	97.00.00.00 - 97.00.00.1F

IPs in Range	Mask Bits	Subnet Mask	Hex Subnet Mask
32	27	255.255.255.224	FF.FF.FF.E0

151.0.0.0 /28 B

Input	Input IP	Input Long	Input Hex
151.0.0.0/28	151.0.0.0	2533359616	97.00.00.00

CIDR	CIDR IP Range	CIDR Long Range	CIDR Hex Range
151.0.0.0/28	151.0.0.0 - 151.0.0.15	2533359616 - 2533359631	97.00.00.00 - 97.00.00.0F

IPs in Range	Mask Bits	Subnet Mask	Hex Subnet Mask
16	28	255.255.255.240	FF.FF.FF.F0

151.0.0.0 /29 C

Input	Input IP	Input Long	Input Hex
151.0.0.0/29	151.0.0.0	2533359616	97.00.00.00

CIDR	CIDR IP Range	CIDR Long Range	CIDR Hex Range
151.0.0.0/29	151.0.0.0 - 151.0.0.7	2533359616 - 2533359623	97.00.00.00 - 97.00.00.07

IPs in Range	Mask Bits	Subnet Mask	Hex Subnet Mask
8	29	255.255.255.248	FF.FF.FF.F8

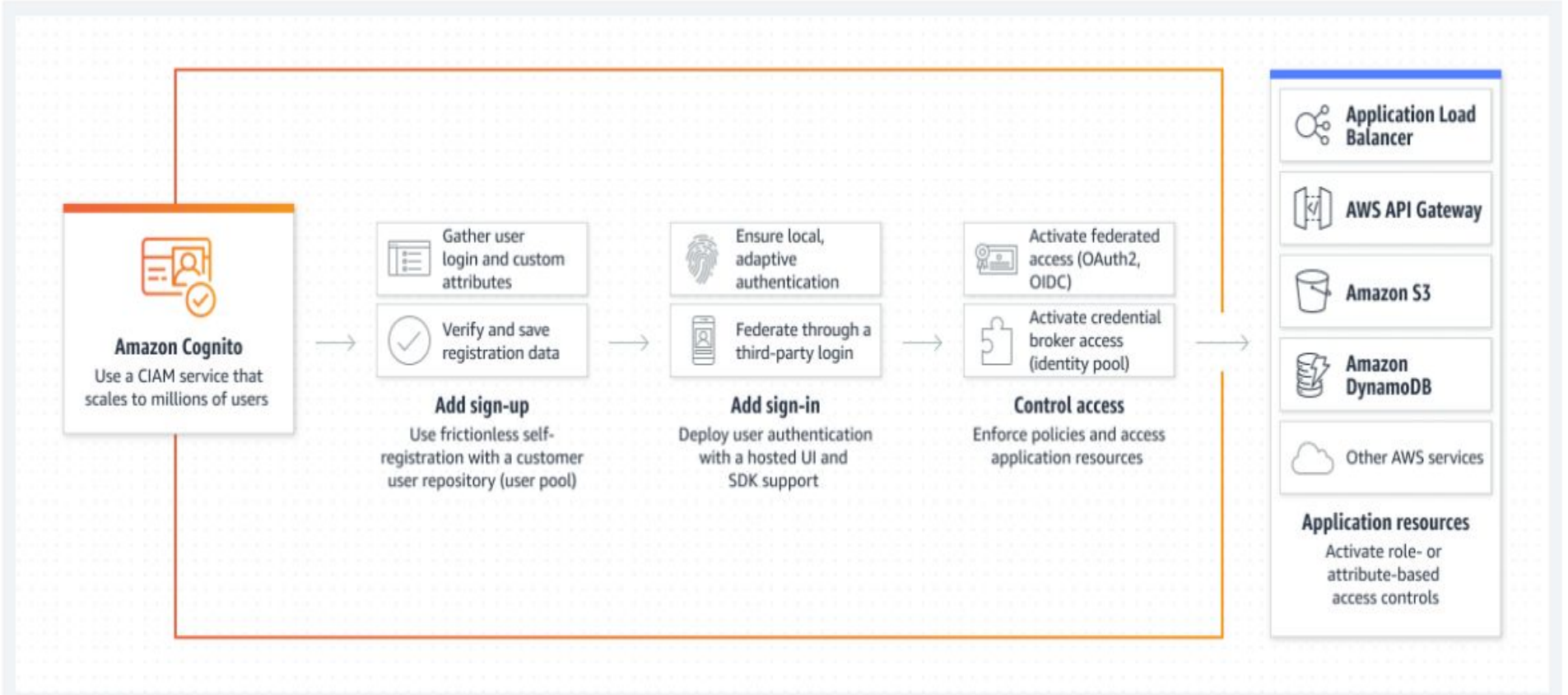
151.0.0.0 /30 D

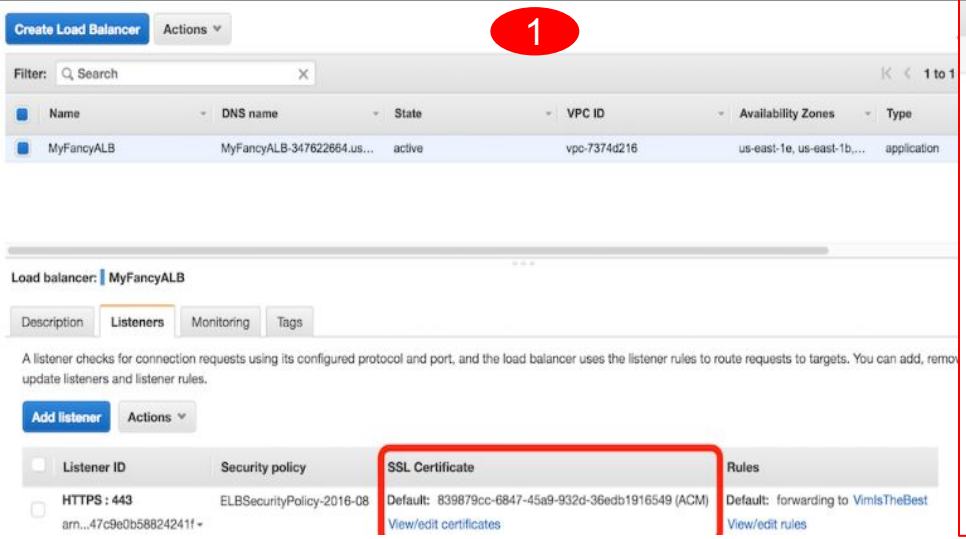
Input	Input IP	Input Long	Input Hex
151.0.0.0/30	151.0.0.0	2533359616	97.00.00.00

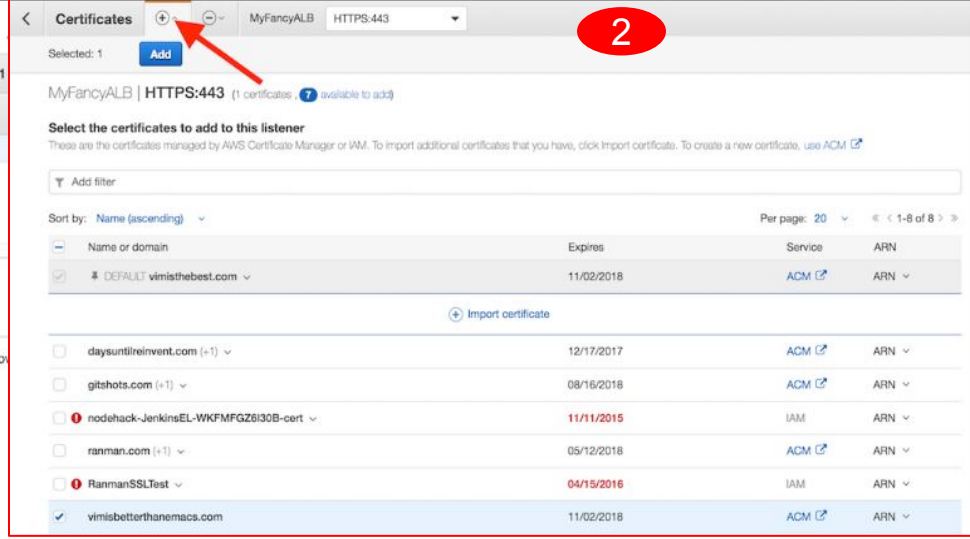
CIDR	CIDR IP Range	CIDR Long Range	CIDR Hex Range
151.0.0.0/30	151.0.0.0 - 151.0.0.3	2533359616 - 2533359619	97.00.00.00 - 97.00.00.03

IPs in Range	Mask Bits	Subnet Mask	Hex Subnet Mask
4	30	255.255.255.252	FF.FF.FF.FC

With Amazon Cognito, you can add user sign-up and sign-in features and control access to your web and mobile applications. Amazon Cognito provides an identity store that scales to millions of users, supports social and enterprise identity federation, and offers advanced security features to protect your consumers and business. Built on open identity standards, Amazon Cognito supports various compliance regulations and integrates with frontend and backend development resources.







With SNI support we're making it easy to use more than one certificate with the same ALB. The most common reason you might want to use multiple certificates is to handle different domains with the same load balancer. It's always been possible to use wildcard and subject-alternate-name (SAN) certificates with ALB, but these come with limitations. Wildcard certificates only work for related subdomains that match a simple pattern and while SAN certificates can support many different domains, the same certificate authority has to authenticate each one. That means you have reauthenticate and reprovision your certificate everytime you add a new domain.

Choosing a routing policy

PDF | RSS

When you create a record, you choose a routing policy, which determines how Amazon Route 53 responds to queries:

- **Simple routing policy** – Use for a single resource that performs a given function for your domain, for example, a web server that serves content for the example.com website. You can use simple routing to create records in a private hosted zone.
- **Failover routing policy** – Use when you want to configure active-passive failover. You can use failover routing to create records in a private hosted zone.
- **Geolocation routing policy** – Use when you want to route traffic based on the location of your users. You can use geolocation routing to create records in a private hosted zone.
- **Geoproximity routing policy** – Use when you want to route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another.
- **Latency routing policy** – Use when you have resources in multiple AWS Regions and you want to route traffic to the region that provides the best latency. You can use latency routing to create records in a private hosted zone.
- **IP-based routing policy** – Use when you want to route traffic based on the location of your users, and have the IP addresses that the traffic originates from.
- **Multivalue answer routing policy** – Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random. You can use multivalue answer routing to create records in a private hosted zone.
- **Weighted routing policy** – Use to route traffic to multiple resources in proportions that you specify. You can use weighted routing to create records in a private hosted zone.

Weighted routing

PDF | RSS

Weighted routing lets you associate multiple resources with a single domain name (example.com) or subdomain name (acme.example.com) and choose how much traffic is routed to each resource. This can be useful for a variety of purposes, including load balancing and testing new versions of software.

To configure weighted routing, you create records that have the same name and type for each of your resources. You assign each record a relative weight that corresponds with how much traffic you want to send to each resource. Amazon Route 53 sends traffic to a resource based on the weight that you assign to the record as a proportion of the total weight for all records in the group:

$$\frac{\text{Weight for a specified record}}{\text{Sum of the weights for all records}}$$

For example, if you want to send a tiny portion of your traffic to one resource and the rest to another resource, you might specify weights of 1 and 255. The resource with a weight of 1 gets 1/256th of the traffic (1/(1+255)), and the other resource gets 255/256ths (255/(1+255)). You can gradually change the balance by changing the weights. If you want to stop sending traffic to a resource, you can change the weight for that record to 0.

You can use weighted routing policy for records in a private hosted zone.

Managing your storage lifecycle

PDF | RSS

To manage your objects so that they are stored cost effectively throughout their lifecycle, configure their *Amazon S3 Lifecycle*. An *S3 Lifecycle configuration* is a set of rules that define actions that Amazon S3 applies to a group of objects. There are two types of actions:

- **Transition actions** – These actions define when objects transition to another storage class. For example, you might choose to transition objects to the S3 Standard-IA storage class 30 days after creating them, or archive objects to the S3 Glacier Flexible Retrieval storage class one year after creating them. For more information, see Using Amazon S3 storage classes.

  There are costs associated with lifecycle transition requests. For pricing information, see Amazon S3 pricing.
- **Expiration actions** – These actions define when objects expire. Amazon S3 deletes expired objects on your behalf.

  Lifecycle expiration costs depend on when you choose to expire objects. For more information, see Expiring objects.

If it seems like the APN Partner's product will need to access a customer account, I'll check to see how the APN Partner is getting credentials from the customer. If a partner is asking customers for AWS Identity and Access Management (IAM) access keys and secret keys, I halt my investigation and focus on helping the partner fix this approach.

It's not that I have a problem with partners accessing customer accounts—APN Partners can add incredible functionality and value to the resources in an AWS account. For example, they can analyze AWS CloudTrail logs, or help optimize costs by monitoring a customer's Amazon Elastic Compute Cloud (Amazon EC2) usage. The problem here is *how* the APN Partner is accessing the AWS account.

IAM access keys and secret keys could be used anywhere, by anyone who has them. If a customer gives these keys to an APN Partner, they need to be able to trust that the APN Partner is adhering to best practices to protect those keys. This should really resonate with APN Partners, who need to store and protect their customers' keys, but lack control over how customers manage those keys. Using IAM access keys and secret keys for cross-account access is not ideal for anyone. Fortunately, there is a better way.

Cross-account IAM roles allow customers to securely grant access to AWS resources in their account to a third party, like an APN Partner, while retaining the ability to control and audit who is accessing their AWS account. Cross-account roles reduce the amount of sensitive information APN Partners need to store for their customers, so that they can focus on their product instead of managing keys.

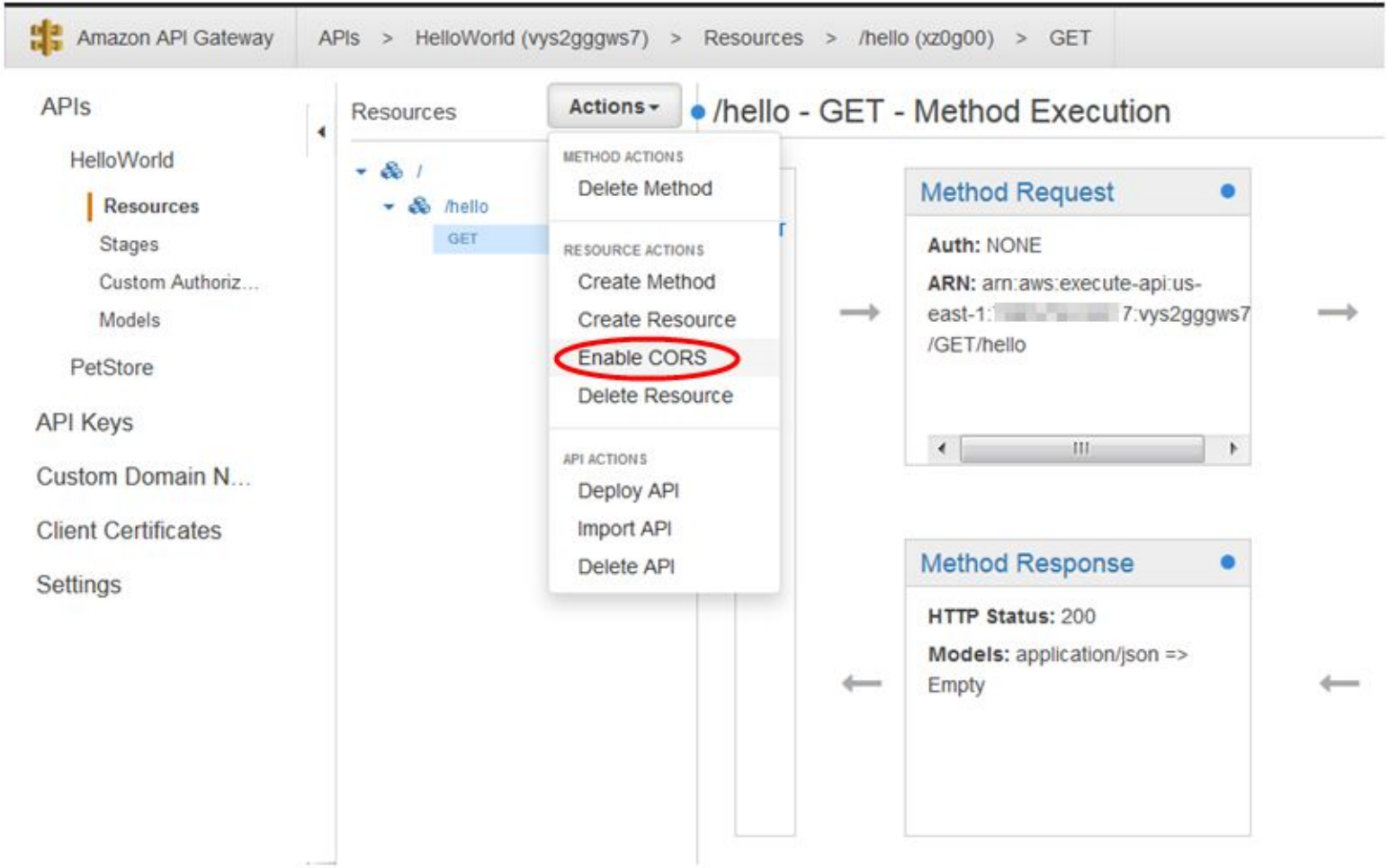
	Spot Instances	On-Demand Instances
Launch time	Can only be launched immediately if the Spot Instance request is active and capacity is available.	Can only be launched immediately if you make a manual launch request and capacity is available.
Available capacity	If capacity is not available, the Spot Instance request continues to automatically make the launch request until capacity becomes available.	If capacity is not available when you make a launch request, you get an insufficient capacity error (ICE).
Hourly price	The hourly price for Spot Instances varies based on demand.	The hourly price for On-Demand Instances is static.
Rebalance recommendation	The signal that Amazon EC2 emits for a running Spot Instance when the instance is at an elevated risk of interruption.	You determine when an On-Demand Instance is interrupted (stopped, hibernated, or terminated).
Instance interruption	You can stop and start an Amazon EBS-backed Spot Instance. In addition, the Amazon EC2 Spot service can interrupt an individual Spot Instance if capacity is no longer available, the Spot price exceeds your maximum price, or demand for Spot Instances increases.	You determine when an On-Demand Instance is interrupted (stopped, hibernated, or terminated).

On-Demand Instances

- Pay for the instances and the compute capacity used by the hour or the second, depending on which instances you run
- No long-term commitments or up-front payments
- Instances can be scaled accordingly as per the demand
- Although AWS makes effort to have the capacity to launch On-Demand instances, there might be instances during peak demand where the instance cannot be launched
- Well suited for
  - Users that want the low cost and flexibility of EC2 without any up-front payment or long-term commitment
  - Applications with **short term, spiky, or unpredictable workloads that cannot be interrupted**
  - Applications being developed or tested on EC2 for the first time

Spot Instances

- EC2 Spot instances allow access to spare EC2 computing capacity for up to 90% off the On-Demand price.
- EC2 sets up the hourly price referred to as **Spot price**, which fluctuates depending upon the demand and supply of spot instances.
- Spot instances enable bidding on unused EC2 instances and are launched whenever the bid price exceeds the current market spot price.
- Spot Instances can be interrupted by EC2 when EC2 needs the capacity back with a two minutes notification.
- Spot instances are a cost-effective choice and can bring the EC2 costs down significantly.
- Spot instances can be used for **applications flexible in the timing when they can run and also able to handle interruption** by storing the state externally *for e.g. they are well-suited for data analysis, batch jobs, background processing, and optional tasks*
- Spot instances differ from the On-Demand instances
  - they are not launched immediately
  - they can be terminated anytime
  - price varies as per the demand and supply of spot instances
- Usual strategy involves using Spot instances with On-Demand or Reserved Instances, which provide a minimum level of guaranteed compute resources, while spot instances provide an additional computation boost.
- Spot instances can also be launched with a required duration (also known as **Spot blocks**), which are not interrupted due to changes in the Spot price.
- EC2 provides a data feed, sent to an S3 bucket specified during subscription, that describes the Spot instance usage and pricing.
- T2 and HS1 instance class types are not supported for Spot instances
- Well Suited for
  - Ideal for various **stateless, fault-tolerant, or flexible applications** such as big data, containerized workloads, CI/CD, high-performance computing (HPC), web servers, and other test & development workloads
  - Applications that have **flexible start and end times**
  - Applications that are only feasible at very low compute prices
  - Users with **urgent computing needs for large amounts of additional capacity**

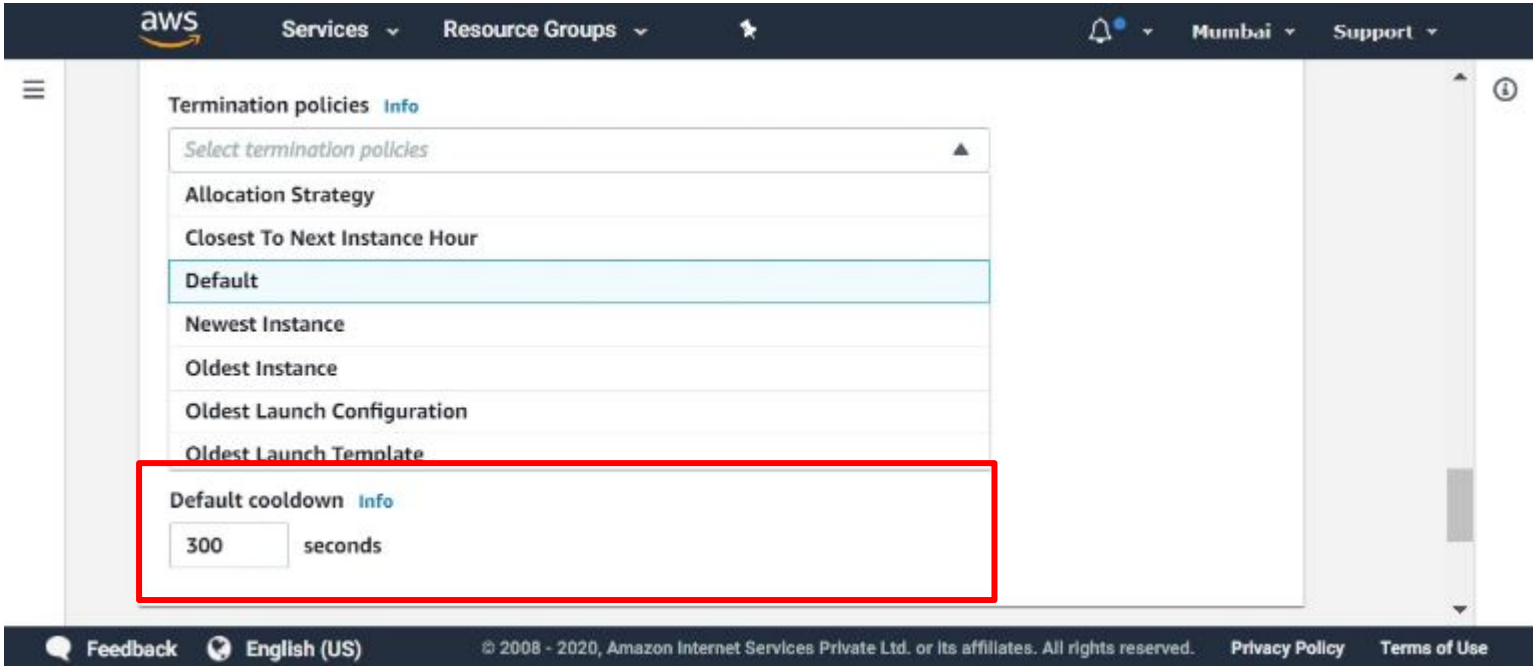

Amazon API Gateway
APIs > HelloWorld (vys2gggws7) > Resources > /hello (xz0g00) > GET
APIs
HelloWorld
Resources
Stages
Custom Authoriz...
Models
PetStore
API Keys
Custom Domain N...
Client Certificates
Settings
Resources
Actions
/hello - GET - Method Execution
/
/hello
GET
METHOD ACTIONS
Delete Method
RESOURCE ACTIONS
Create Method
Create Resource
Enable CORS
Delete Resource
API ACTIONS
Deploy API
Import API
Delete API
Method Request
Auth: NONE
ARN: arn:aws:execute-api:us-east-1:[redacted]7:vys2gggws7/GET/hello
Method Response
HTTP Status: 200
Models: application/json => Empty

After your Auto Scaling group launches or terminates instances, it waits for a cooldown period to end before any further scaling activities initiated by simple scaling policies can start. The intention of the cooldown period is to prevent your Auto Scaling group from launching or terminating additional instances before the effects of previous activities are visible.

Important

As a best practice, we recommend that you do not use simple scaling policies and scaling cooldowns.

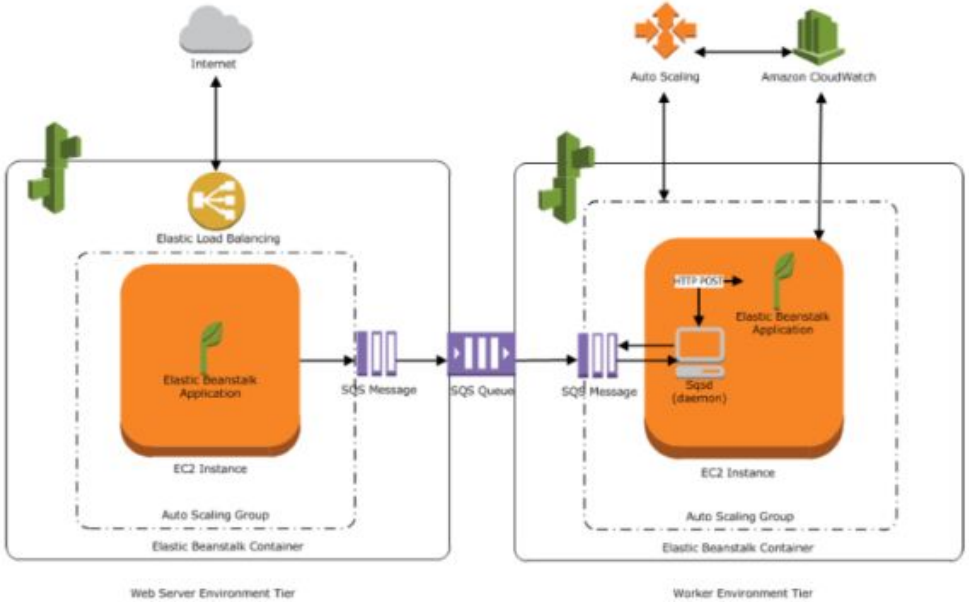
In most cases, a target tracking scaling policy or a step scaling policy is better for scaling performance. For a scaling policy that changes the size of your Auto Scaling group proportionally as the value of the scaling metric decreases or increases, we recommend target tracking over either simple scaling or step scaling.



The Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services.

For a typical web application, configuring for HA requires running multiple web servers behind a load balancer, configuring Auto Scaling to replace lost instances and launch more instances in response to surges in traffic, and having a standby database instance configured for automatic failover.

For AWS Elastic Beanstalk, production HA configuration also includes running your database instances outside of your web server environment which allows you to perform blue/green deployments and advanced database management operations.



tf-s3-app-lock-andy

Actions | Explore table items

Overview | Indexes | Monitor | Global tables | Backups | Exports and streams | Additional settings

Exports to S3 (0) Info

Showing all export jobs from the last 90 days.

View details | Export to S3

Find exports

1

Export ARN	Destination S3 bucket	Status	Start time (UTC+03:00)

No exports

Export to S3

Amazon Kinesis data stream details

Amazon Kinesis Data Streams for DynamoDB captures item-level changes in your table, and replicates the changes to a Kinesis data stream. You then can consume and manage the change information from Kinesis. Charges apply.

Turn on

Status

Off

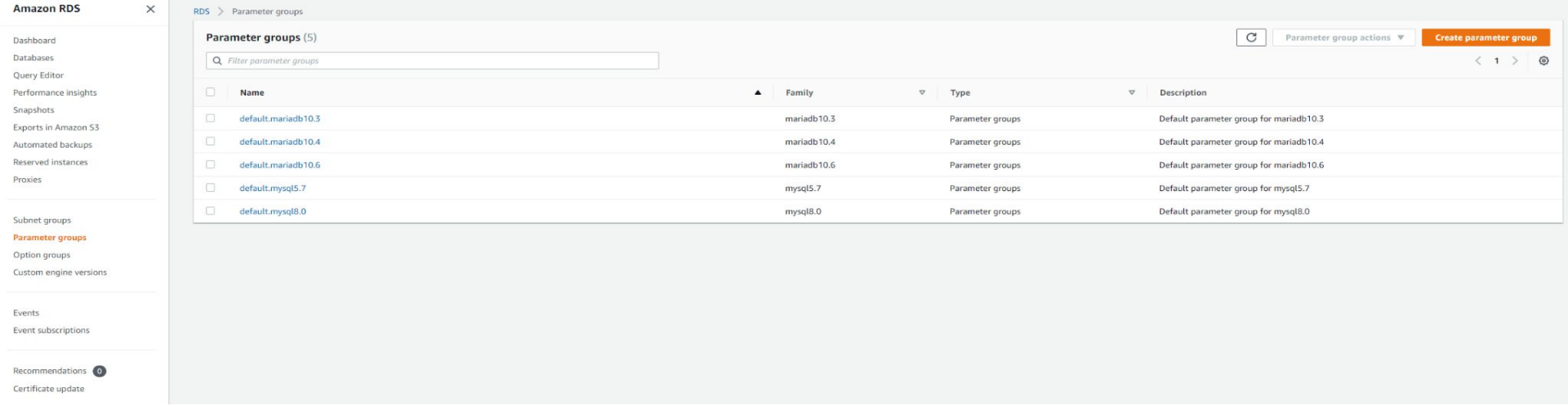
DynamoDB stream details

Capture item-level changes in your table, and push the changes to a DynamoDB stream. You then can access the change information through the DynamoDB Streams API.

Turn on

Stream status

Off



Parameters

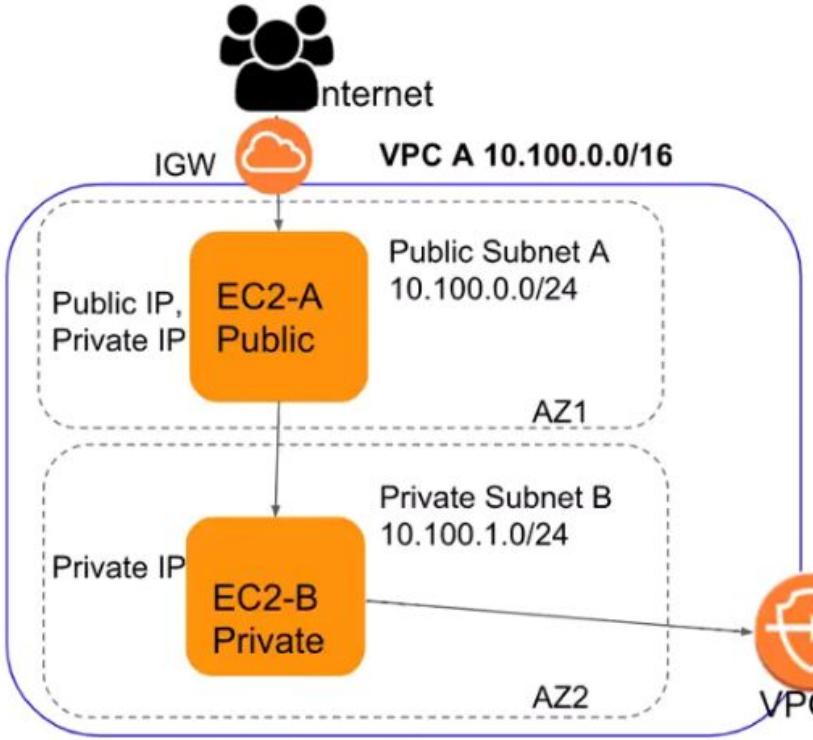
Edit parameters

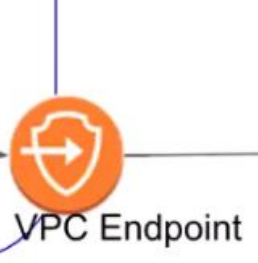
Filter parameters

1 ... 10 11 12 13 14 ... 21

Name	Values	Allowed values	Modifiable	Source	Apply type	Data type	Description
master_verify_checksum		0, 1	true	engine-default	dynamic	boolean	Enabling this variable causes the master to examine checksums when reading from the binary log.
max_allowed_packet		16777216-1073741824	true	engine-default	dynamic	integer	This value by default is small, to catch large (possibly incorrect) packets. Must be increased if using large BLOB columns or long strings. As big as largest BLOB.
max_binlog_cache_size		4096-18446744073709547520	true	engine-default	dynamic	integer	Maximum binlog cache size a transaction may use
max_binlog_size	134217728	4096-1073741824	false	system	dynamic	integer	Server rotates the binlog once it reaches this size
max_binlog_stmt_cache_size		4096-18446744073709547520	true	engine-default	dynamic	integer	If nontransactional statements within a transaction require more than this many bytes of memory, the server generates an error.
max_connect_errors		1-9223372036854775807	true	engine-default	dynamic	integer	A host is blocked from further connections if there are more than this number of interrupted connections
max_connections	{DBInstanceClassMemory/12582880}	1-100000	true	system	dynamic	integer	The number of simultaneous client connections allowed.
max_delayed_threads		0-16384	true	engine-default	dynamic	integer	Do not start more than this number of threads to handle INSERT DELAYED statements.
max_digest_length	1024	0-1048576	true	engine-default	static	integer	The maximum number of bytes available for computing statement digests.
max_error_count		0-65535	true	engine-default	dynamic	integer	The maximum number of error, warning, and note messages to be stored for display.
max_execution_time		0-18446744073709551615	true	engine-default	dynamic	integer	The execution timeout for SELECT statements, in milliseconds.
max_heap_table_size		16384-18446744073709554752	true	engine-default	dynamic	integer	Maximum size to which MEMORY tables are allowed to grow.

Assignment 5 - VPC Endpoint





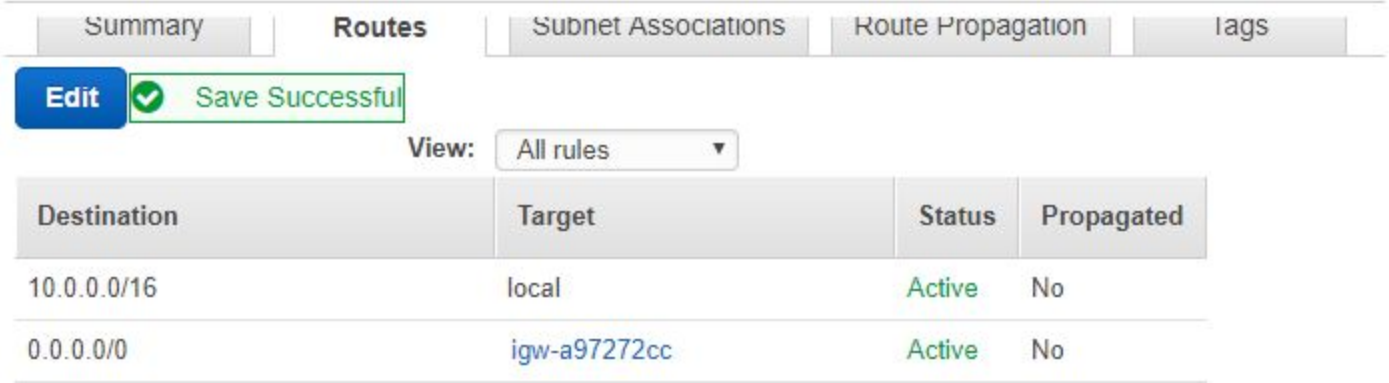
Private subnet route table

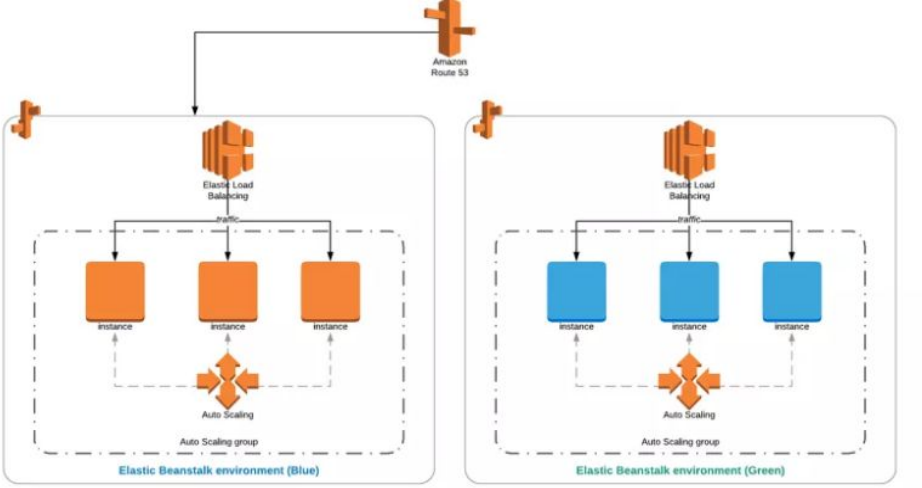
Destination	Target
10.100.0.0/16	local
s3 -endpoint	*vpce-id*

Amazon S3 Strong Consistency

Amazon S3 delivers strong read-after-write consistency automatically for all applications, without changes to performance or availability, without sacrificing regional isolation for applications, and at no additional cost. With strong consistency, S3 simplifies the migration of on-premises analytics workloads by removing the need to make changes to applications, and reduces costs by removing the need for extra infrastructure to provide strong consistency.

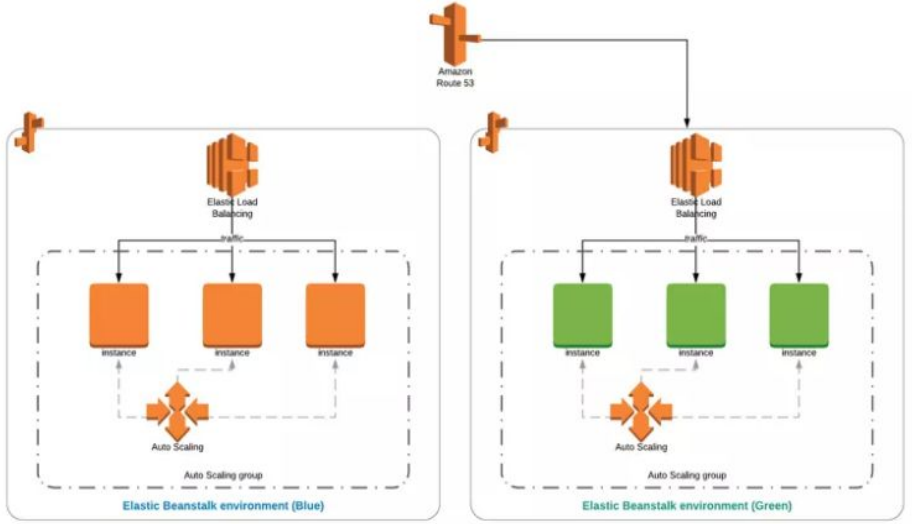
After a successful write of a new object, or an overwrite or delete of an existing object, any subsequent read request immediately receives the latest version of the object. S3 also provides strong consistency for list operations, so after a write, you can immediately perform a listing of the objects in a bucket with any changes reflected.

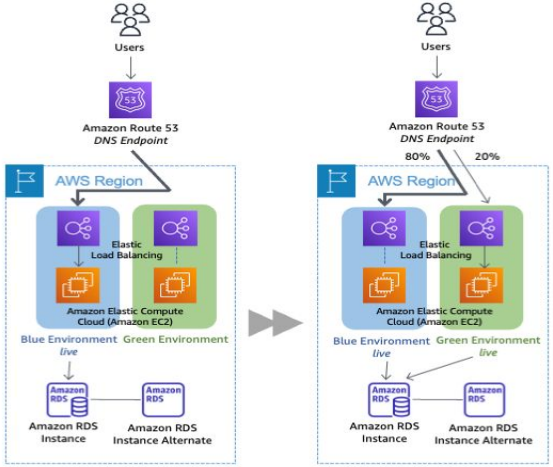

Summary
Routes
Subnet Associations
Route Propagation
Tags
Edit
Save Successful
View:
All rules
Destination
Target
Status
Propagated
10.0.0.0/16
local
Active
No
0.0.0.0/0
igw-a97272cc
Active
No



Old environment continues to serve the traffic during deployments on the cloned environment. In case of failure, it will not be effected.

Deployments are run on the cloned environment





	Throughput Optimized HDD volumes	Cold HDD volumes
Volume type	`st1`	`sc1`
Durability	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	
Use cases	• Big data • Data warehouses • Log processing	• Throughput-oriented storage for data that is infrequently accessed • Scenarios where the lowest storage cost is important
Volume size	125 GiB - 16 TiB	
Max IOPS per volume (1 MiB I/O)	500	250
Max throughput per volume	500 MiB/s	250 MiB/s
Amazon EBS Multi-attach	Not supported	
Boot volume	Not supported	

	S3 Standard	S3 Intelligent-Tiering*	S3 Standard-IA	S3 One Zone-IA†	S3 Glacier Instant Retrieval	S3 Glacier Flexible Retrieval	S3 Glacier Deep Archive
Designed for durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Designed for availability	99.99%	99.9%	99.9%	99.5%	99.9%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99%	99.9%	99.9%
Availability Zones	≥3	≥3	≥3	1	≥3	≥3	≥3
Minimum capacity charge per object	N/A	N/A	128 KB	128 KB	128 KB	40 KB	40 KB
Minimum storage duration charge	N/A	N/A	30 days	30 days	90 days	90 days	180 days
Retrieval charge	N/A	N/A	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved
First byte latency	milliseconds	milliseconds	milliseconds	milliseconds	milliseconds	minutes or hours	hours
Storage type	Object	Object	Object	Object	Object	Object	Object
Lifecycle transitions	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Attributes

Attribute name-value pairs per item

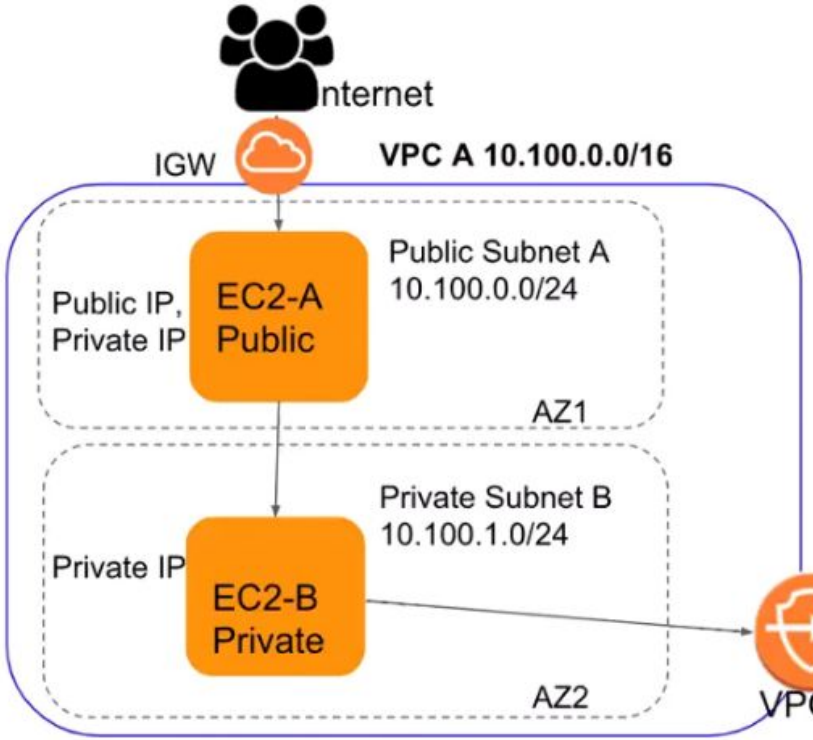
The cumulative size of attributes per item must fit within the maximum DynamoDB item size (400 KB).

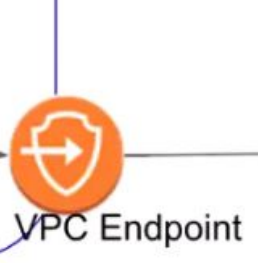
Amazon DynamoDB is a key-value and document database that delivers single-digit millisecond performance at any scale. It's a fully managed, multiregion, multimaster, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications. DynamoDB can handle more than 10 trillion requests per day and can support peaks of more than 20 million requests per second.

Many of the world's fastest growing businesses such as Lyft, Airbnb, and Redfin as well as enterprises such as Samsung, Toyota, and Capital One depend on the scale and performance of DynamoDB to support their mission-critical workloads.

Hundreds of thousands of Amazon Web Services customers have chosen DynamoDB as their key-value and document database for mobile, web, gaming, ad tech, IoT, and other applications that need low-latency data access at any scale. Create a new table for your application and let DynamoDB handle the rest.

Assignment 5 - VPC Endpoint





Private subnet route table

Destination	Target
10.100.0.0/16	local
s3 -endpoint	*vpce-id*

Choosing a routing policy

PDF | RSS

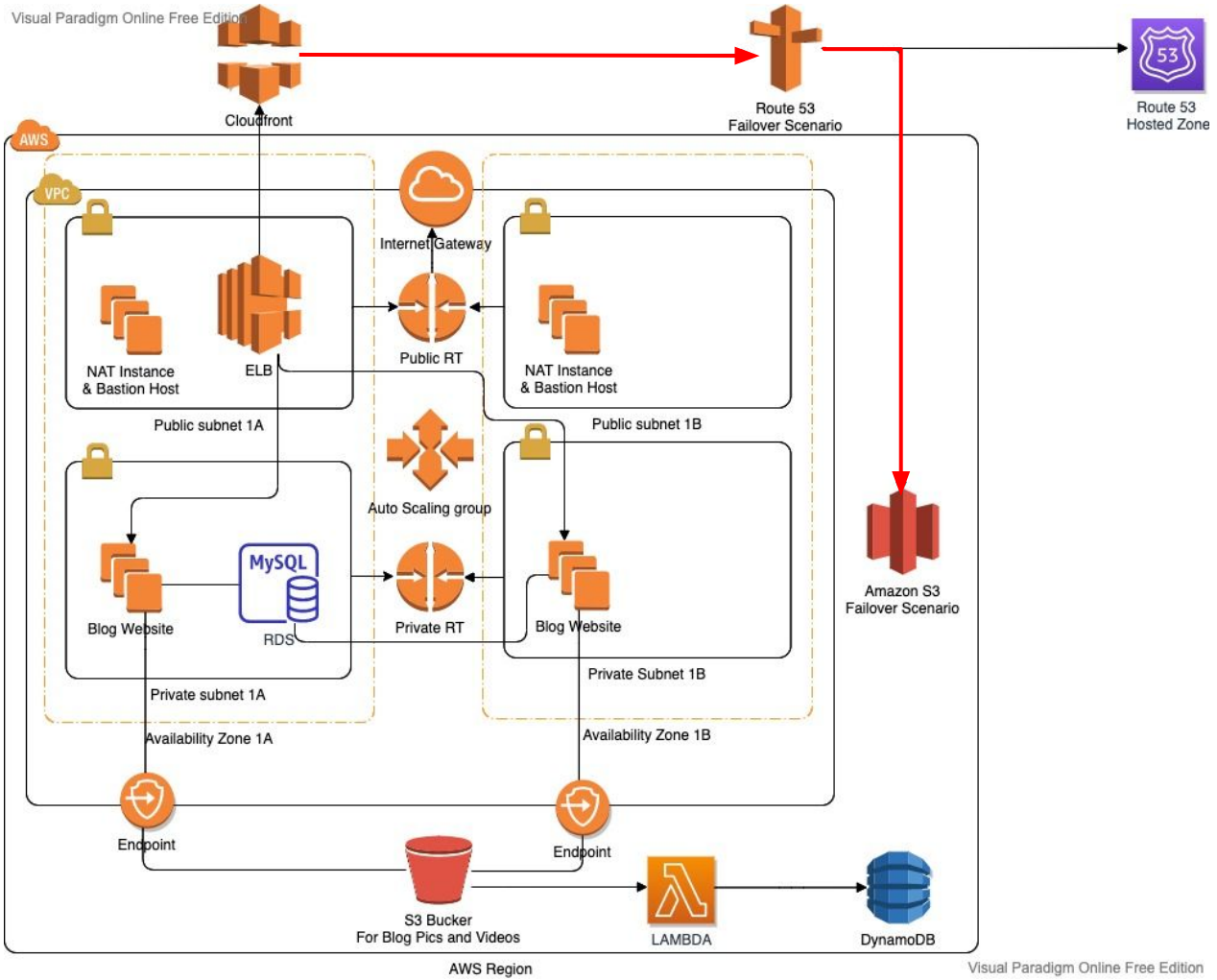
When you create a record, you choose a routing policy, which determines how Amazon Route 53 responds to queries:

- **Simple routing policy** – Use for a single resource that performs a given function for your domain, for example, a web server that serves content for the example.com website. You can use simple routing to create records in a private hosted zone.
- **Failover routing policy** – Use when you want to configure active-passive failover. You can use failover routing to create records in a private hosted zone.
- **Geolocation routing policy** – Use when you want to route traffic based on the location of your users. You can use geolocation routing to create records in a private hosted zone.
- **Geoproximity routing policy** – Use when you want to route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another.
- **Latency routing policy** – Use when you have resources in multiple AWS Regions and you want to route traffic to the region that provides the best latency. You can use latency routing to create records in a private hosted zone.
- **IP-based routing policy** – Use when you want to route traffic based on the location of your users, and have the IP addresses that the traffic originates from.
- **Multivalue answer routing policy** – Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random. You can use multivalue answer routing to create records in a private hosted zone.
- **Weighted routing policy** – Use to route traffic to multiple resources in proportions that you specify. You can use weighted routing to create records in a private hosted zone.

```
#! /bin/bash
yum update -y
wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
yum upgrade
amazon-linux-extras install java-openjdk11 -y
amazon-linux-extras install epel -y
yum install jenkins -y
systemctl enable jenkins
systemctl start jenkins
systemctl status jenkins
yum install git -y
```

```
user_data = <<-EOF
            #! /bin/bash
            sudo yum update -y
            sudo yum install java-11-amazon-corretto -y
            cd /home/ec2-user/
            wget https://ftp.itu.edu.tr/Mirror/Apache/maven/maven-3/3.6.3/binaries/
            apache-maven-3.6.3-bin.tar.gz
            tar -zxvf $(ls | grep apache-maven-*-bin.tar.gz)
            rm -rf $(ls | grep apache-maven-*-bin.tar.gz)
            echo "M2_HOME=/home/ec2-user/$(ls | grep apache-maven)" >> /home/ec2-user/.
            bash_profile
            echo 'export PATH=$PATH:$M2_HOME/bin' >> /home/ec2-user/.bash_profile
            EOF
}
```

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Amazon RDS Multi-AZ with one standby

Automatic fail over	Protect database performance	Enhance durability	Increase availability
Support high availability for your application with automatic database failover that completes as quickly as 60 seconds with zero data loss and no manual intervention.	Avoid suspending I/O activity on your primary during backup by backing up from your standby instance.	Use Amazon RDS Multi-AZ synchronous replication technologies to keep data on your standby database instance up to date with the primary.	Enhance availability by deploying a standby instance in a second AZ, and achieve fault tolerance in the event of an AZ or database instance failure.

Add rules to a security group

When you add a rule to a security group, the new rule is automatically applied to any resources that are associated with the security group.

Amazon ECS on AWS Fargate

PDF | RSS

AWS Fargate is a technology that you can use with Amazon ECS to run containers without having to manage servers or clusters of Amazon EC2 instances. With AWS Fargate, you no longer have to provision, configure, or scale clusters of virtual machines to run containers. This removes the need to choose server types, decide when to scale your clusters, or optimize cluster packing.

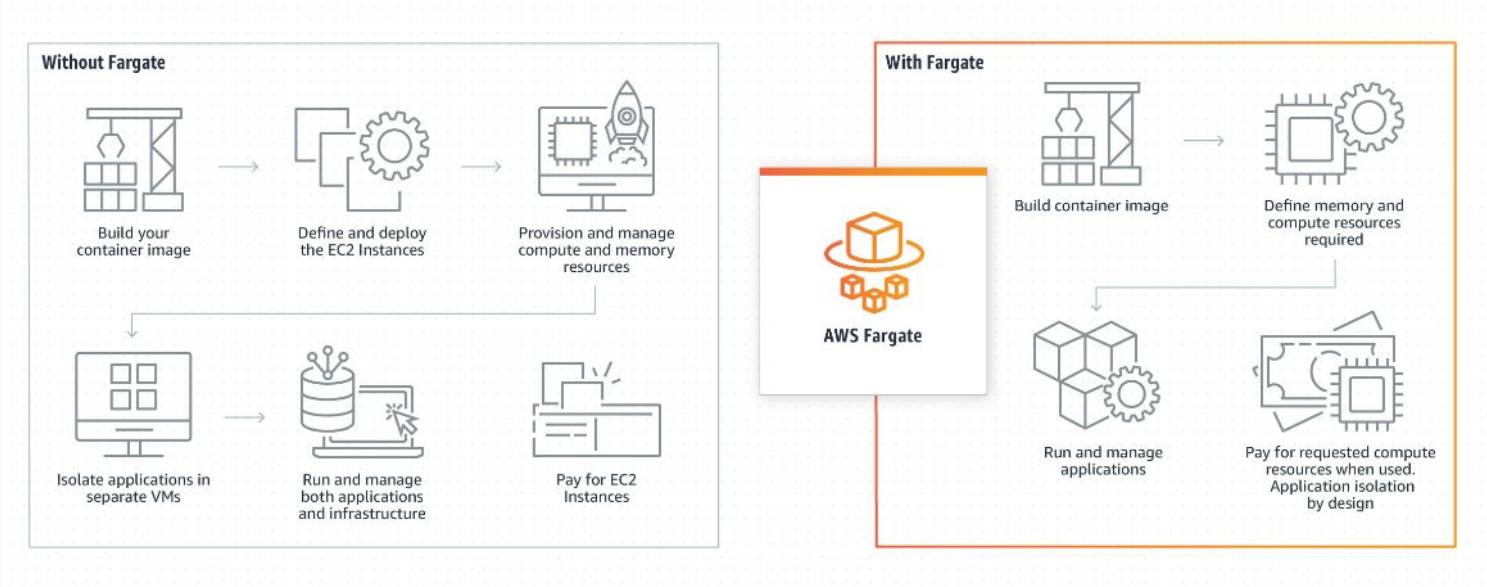
When you run your tasks and services with the Fargate launch type, you package your application in containers, specify the CPU and memory requirements, define networking and IAM policies, and launch the application. Each Fargate task has its own isolation boundary and does not share the underlying kernel, CPU resources, memory resources, or elastic network interface with another task.

Fargate offers platform versions for Amazon Linux 2 and Microsoft Windows 2019 Server Full and Core editions. Unless otherwise specified, the information on this page applies to all Fargate platforms.

This topic describes the different components of Fargate tasks and services, and calls out special considerations for using Fargate with Amazon ECS.

For information about the Regions that support Linux containers on Fargate, see Supported Regions for Linux containers on AWS Fargate.

For information about the Regions that support Windows containers on Fargate, see Supported Regions for Windows containers on AWS Fargate.



Same subnet for EC2 instances and transit gateway association

Consider a configuration where you have EC2 instances and a transit gateway association in the same subnet. The same network ACL is used for both the traffic from the EC2 instances to the transit gateway and traffic from the transit gateway to the instances.

NACL rules are applied as follows for traffic from instances to the transit gateway:

- Outbound rules use the destination IP address for evaluation.
- Inbound rules use the source IP address for evaluation.

NACL rules are applied as follows for traffic from the transit gateway to the instances:

- Outbound rules are not evaluated.
- Inbound rules are not evaluated.

Different subnets for EC2 instances and transit gateway association

Consider a configuration where you have EC2 instances in one subnet and a transit gateway association in a different subnet, and each subnet is associated with a different network ACL.

Network ACL rules are applied as follows for the EC2 instance subnet:

- Outbound rules use the destination IP address to evaluate traffic from the instances to the transit gateway.
- Inbound rules use the source IP address to evaluate traffic from the transit gateway to the instances.

NACL rules are applied as follows for the transit gateway subnet:

- Outbound rules use the destination IP address to evaluate traffic from the transit gateway to the instances.
- Outbound rules are not used to evaluate traffic from the instances to the transit gateway.
- Inbound rules use the source IP address to evaluate traffic from the instances to the transit gateway.
- Inbound rules are not used to evaluate traffic from the transit gateway to the instances.

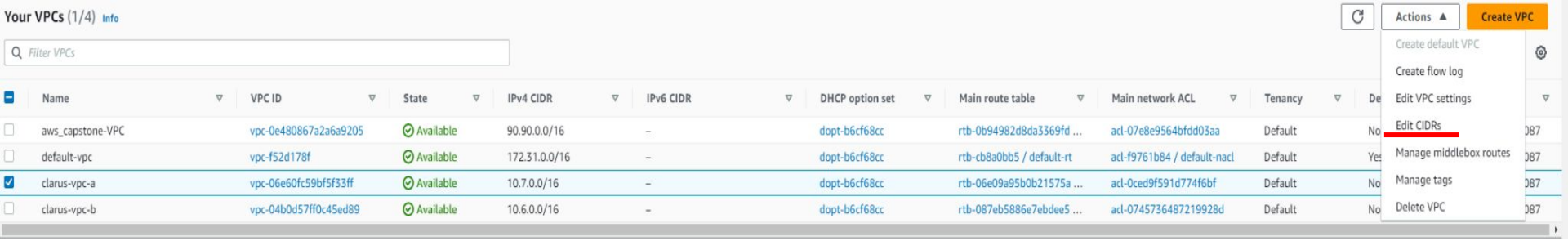
Elastic network interfaces

PDF | RSS

An *elastic network interface* is a logical networking component in a VPC that represents a virtual network card. It can include the following attributes:

- A primary private IPv4 address from the IPv4 address range of your VPC
- One or more secondary private IPv4 addresses from the IPv4 address range of your VPC
- One Elastic IP address (IPv4) per private IPv4 address
- One public IPv4 address
- One or more IPv6 addresses
- One or more security groups
- A MAC address
- A source/destination check flag
- A description

When you create a network interface, it inherits the public IPv4 addressing attribute from the subnet. If you later modify the public IPv4 addressing attribute of the subnet, the network interface keeps the setting that was in effect when it was created. If you launch an instance and specify an existing network interface as the primary network interface, the public IPv4 address attribute is determined by this network interface.



Edit CIDRs Info

Add or remove CIDR blocks for your VPC.

IPv4 CIDRs Info

CIDR	Status	
10.7.0.0/16	Associated	Remove

Add new IPv4 CIDR

IPv6 CIDRs Info

CIDR (Network border group)	Pool	Status

You have no IPv6 CIDR blocks associated with your VPC.

Add new IPv6 CIDR

Uploading Objects

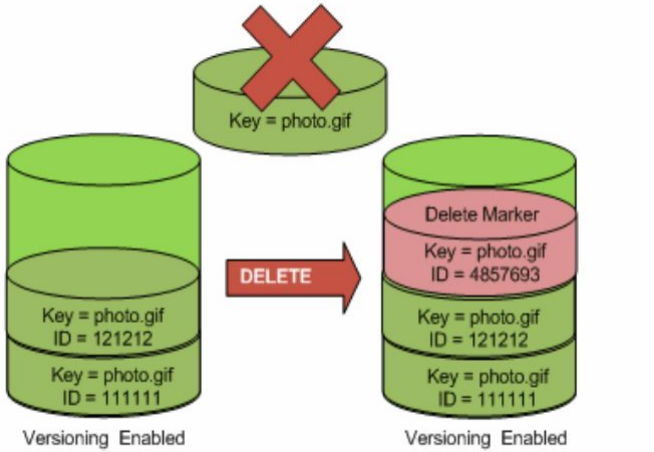
Depending on the size of the data you are uploading, Amazon S3 offers the following options:

- **Upload objects in a single operation—**With a single PUT operation, you can upload objects up to 5 GB in size. For more information, see Uploading Objects in a Single Operation.
- **Upload objects in parts—**Using the multipart upload API, you can upload large objects, up to 5 TB. The multipart upload API is designed to improve the upload experience for larger objects. You can upload objects in parts. These object parts can be uploaded independently, in any order, and in parallel. You can use a multipart upload for objects from 5 MB to 5 TB in size. For more information, see Uploading Objects Using Multipart Upload API.

We recommend that you use multipart uploading in the following ways:

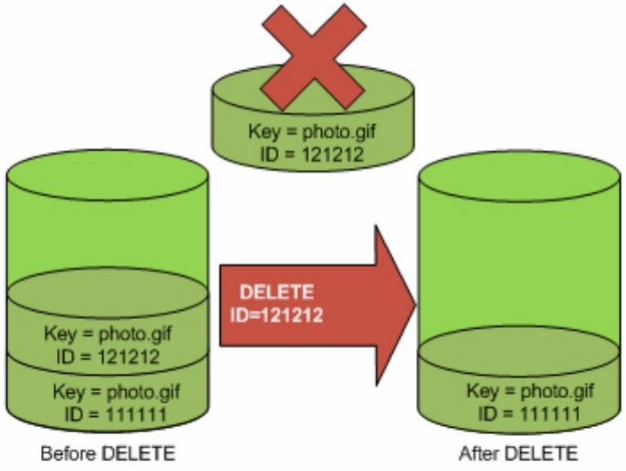
- If you're uploading large objects over a stable high-bandwidth network, use multipart uploading to maximize the use of your available bandwidth by uploading object parts in parallel for multi-threaded performance.
- If you're uploading over a spotty network, use multipart uploading to increase resiliency to network errors by avoiding upload restarts. When using multipart uploading, you need to retry uploading only parts that are interrupted during the upload. You don't need to restart uploading your object from the beginning.

	S3 Standard	S3 Intelligent-Tiering*	S3 Standard-IA	S3 One Zone-IA†	S3 Glacier Instant Retrieval	S3 Glacier Flexible Retrieval	S3 Glacier Deep Archive
Designed for durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Designed for availability	99.99%	99.9%	99.9%	99.5%	99.9%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99%	99.9%	99.9%
Availability Zones	≥3	≥3	≥3	1	≥3	≥3	≥3
Minimum capacity charge per object	N/A	N/A	128 KB	128 KB	128 KB	40 KB	40 KB
Minimum storage duration charge	N/A	N/A	30 days	30 days	90 days	90 days	180 days
Retrieval charge	N/A	N/A	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved
First byte latency	milliseconds	milliseconds	milliseconds	milliseconds	milliseconds	minutes or hours	hours
Storage type	Object	Object	Object	Object	Object	Object	Object
Lifecycle transitions	Yes	Yes	Yes	Yes	Yes	Yes	Yes



To permanently delete versioned objects, you must use DELETE Object versioned.

The following figure shows that deleting a specified object version permanently removes that obje



Supported event destinations

Amazon S3 can send event notification messages to the following destinations. You specify the Amazon Resource Name (ARN) value of these destinations in the notification configuration.

- Amazon Simple Notification Service (Amazon SNS) topics
- Amazon Simple Queue Service (Amazon SQS) queues
- AWS Lambda function

You must grant Amazon S3 permissions to post messages to an Amazon SNS topic or an Amazon SQS queue. You must also grant Amazon S3 permission to invoke an AWS Lambda function on your behalf. For information about granting these permissions, see Granting permissions to publish event notification messages to a destination.

AWS Simple Email Service (SES) is a cost-effective email service built on the reliable and scalable infrastructure that Amazon.com developed to serve its own customer base. With Amazon SES, you can send transactional emails, marketing messages, or any other type of high-quality content.

While your account is in the sandbox, you can use all of the features of Amazon SES. However, when your account is in the sandbox, we apply the following restrictions to your account:

- You can only send mail to verified email addresses and domains or to the Amazon SES mailbox simulator.
- You can only send mail from verified email addresses and domains.

When your account is out of the sandbox, you can send email to any recipient, regardless of whether the recipient's address or domain is verified. However, you still have to verify all identities that you use as "From", "Source", "Sender", or "Return-Path" addresses.

Amazon ECS Container Agent

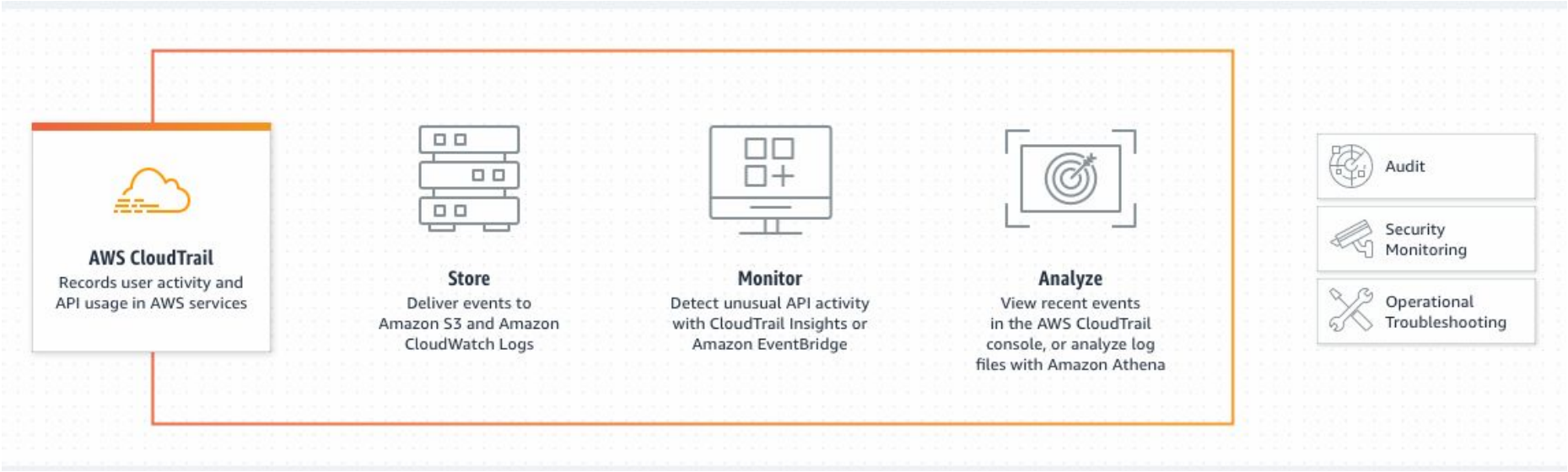
The Amazon ECS-optimized AMI looks for agent configuration data in the `/etc/ecs/ecs.config` file when the container agent starts. You can specify this configuration data at launch with Amazon EC2 user data. For more information about available Amazon ECS container agent configuration variables, see Amazon ECS Container Agent Configuration.

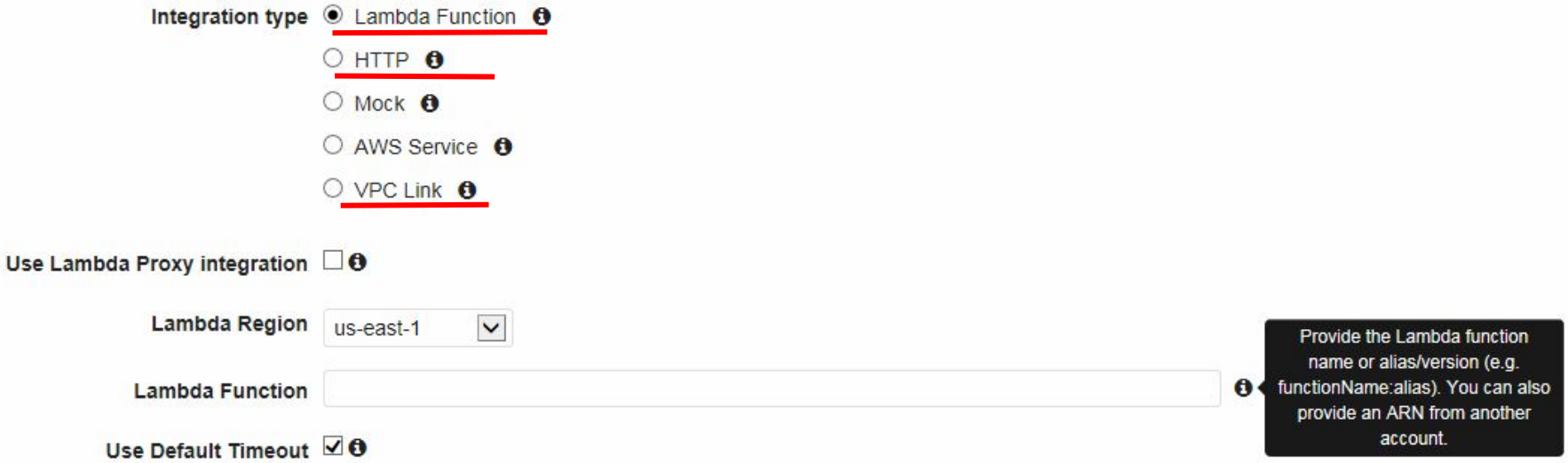
To set only a single agent configuration variable, such as the cluster name, use **echo** to copy the variable to the configuration file:

```
#!/bin/bash
echo "ECS_CLUSTER=MyCluster" >> /etc/ecs/ecs.config
```

If you have multiple variables to write to `/etc/ecs/ecs.config`, use the following heredoc format. This format writes everything between the lines beginning with **cat** and `EOF` to the configuration file.

```
#!/bin/bash
cat <<'EOF' >> /etc/ecs/ecs.config
ECS_CLUSTER=MyCluster
ECS_ENGINE_AUTH_TYPE=docker
ECS_ENGINE_AUTH_DATA={"https://index.docker.io/v1/":{"username":"my_name","password":"my_password","email":"email@example.com"}}
ECS_LOGLEVEL=debug
EOF
```


AWS CloudTrail
Records user activity and API usage in AWS services
Store
Deliver events to Amazon S3 and Amazon CloudWatch Logs
Monitor
Detect unusual API activity with CloudTrail Insights or Amazon EventBridge
Analyze
View recent events in the AWS CloudTrail console, or analyze log files with Amazon Athena
Audit
Security Monitoring
Operational Troubleshooting


Integration type
Lambda Function
HTTP
Mock
AWS Service
VPC Link
Use Lambda Proxy integration
Lambda Region
us-east-1
Lambda Function
Use Default Timeout
Provide the Lambda function name or alias/version (e.g. functionName:alias). You can also provide an ARN from another account.

← Method Execution /{proxy+} - GET - Method Request

Provide information about this method's authorization settings and the parameters it can receive.

Settings

Authorization NONE

Request Validator NONE

API Key Required true

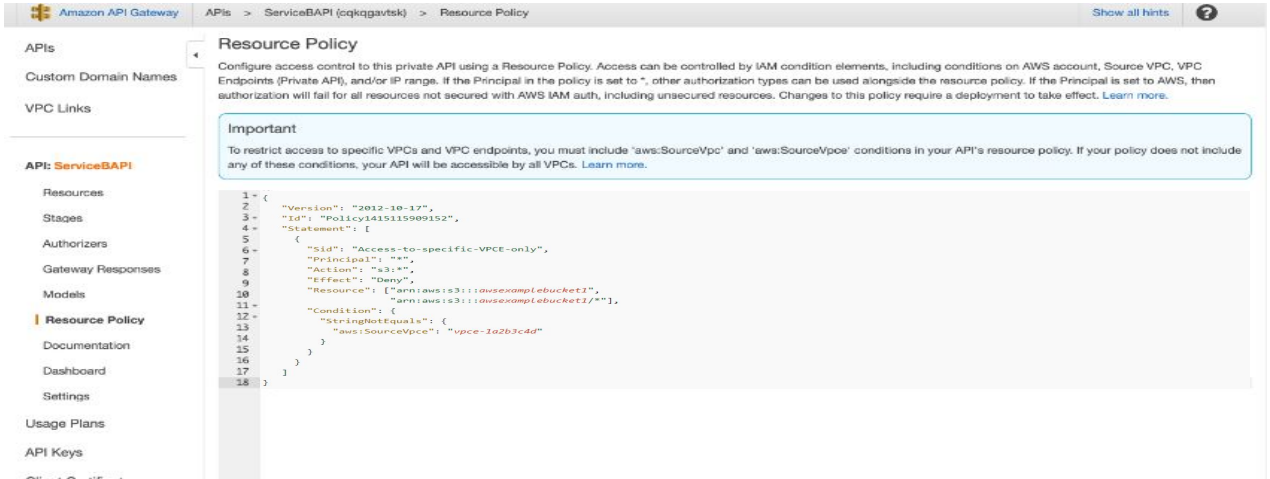
Request Paths

URL Query String Parameters

Name	Required	Caching
param1	☐	☑

Add query string

Condition keys	Criteria	Needs AuthN?	Authorization type
aws:CurrentTime	None	No	All
aws:EpochTime	None	No	All
aws:TokenIssueTime	Key is present only in requests that are signed using temporary security credentials.	Yes	IAM
aws:MultiFactorAuthPresent	Key is present only in requests that are signed using temporary security credentials.	Yes	IAM
aws:MultiFactorAuthAge	Key is present only if MFA is present in the requests.	Yes	IAM
aws:PrincipalAccount	None	Yes	IAM
aws:PrincipalArn	None	Yes	IAM
aws:PrincipalOrgID	This key is included in the request context only if the principal is a member of an organization.	Yes	IAM
aws:PrincipalOrgPaths	This key is included in the request context only if the principal is a member of an organization.	Yes	IAM
aws:PrincipalTag	This key is included in the request context if the principal is using an IAM user with attached tags. It is included for a principal using an IAM role with attached tags or session tags.	Yes	IAM
aws:PrincipalType	None	Yes	IAM
aws:Referer	Key is present only if the value is provided by the caller in the HTTP header.	No	All
aws:SecureTransport	None	No	All
aws:SourceArn	None	No	All
aws:SourceIp	None	No	All
aws:SourceVpc	This key can be used only for private APIs.	No	All
aws:SourceVpce	This key can be used only for private APIs.	No	All
aws:UserAgent	Key is present only if the value is provided by the caller in the HTTP header.	No	All
aws:userid	None	Yes	IAM
aws:username	None	Yes	IAM



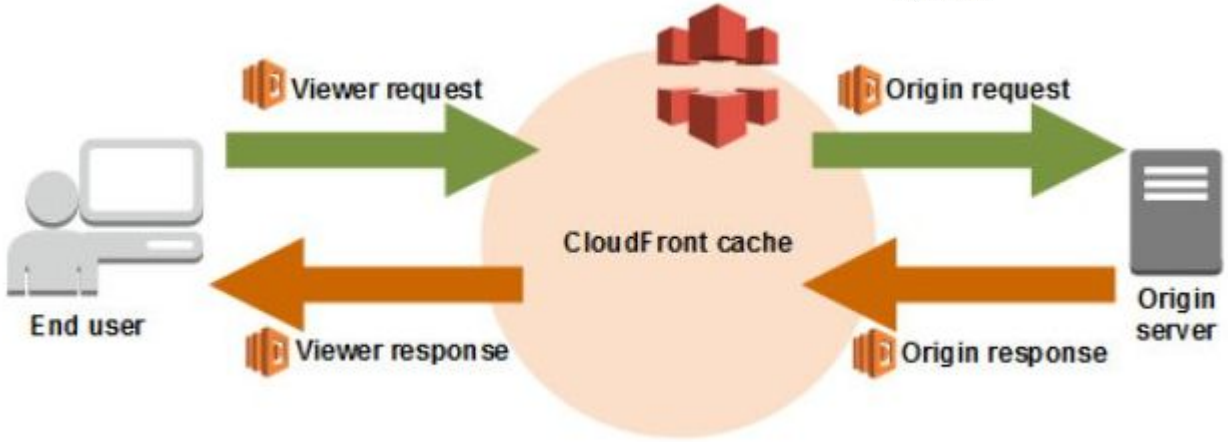
Lambda@Edge is a feature of Amazon CloudFront that lets you run code closer to users of your application, which improves performance and reduces latency. With Lambda@Edge, you don't have to provision or manage infrastructure in multiple locations around the world. You pay only for the compute time you consume - there is no charge when your code is not running.

With Lambda@Edge, you can enrich your web applications by making them globally distributed and improving their performance — all with zero server administration. Lambda@Edge runs your code in response to events generated by the Amazon CloudFront content delivery network (CDN). Just upload your code to AWS Lambda, which takes care of everything required to run and scale your code with high availability at an AWS location closest to your end user.

Q: What Amazon CloudFront events can be used to trigger my functions?

Your functions will automatically trigger in response to the following Amazon CloudFront events:

- Viewer Request - This event occurs when an end user or a device on the Internet makes an HTTP(S) request to CloudFront, and the request arrives at the edge location closest to that user.
- Viewer Response - This event occurs when the CloudFront server at the edge is ready to respond to the end user or the device that made the request.
- Origin Request - This event occurs when the CloudFront edge server does not already have the requested object in its cache, and the viewer request is ready to be sent to your backend origin webserver (e.g. Amazon EC2, or Application Load Balancer, or Amazon S3).
- Origin Response - This event occurs when the CloudFront server at the edge receives a response from your backend origin webserver.



Q: Do my AWS Lambda functions remain available when I change my code or its configuration?

Yes. When you update a Lambda function, there will be a brief window of time, typically less than a minute, when requests could be served by either the old or the new version of your function.

update-function-code

Description

Updates the code for the specified Lambda function. This operation must only be used on an existing Lambda function and cannot be used to update the function configuration.

If you are using the versioning feature, note this API will always update the $LATEST version of your Lambda function. For information about the versioning feature, see AWS Lambda Function Versioning and Aliases .

Poll-Based Invokes

This invocation model is designed to allow you to integrate with AWS Stream and Queue based services with no code or server management. Lambda will poll the following services on your behalf, retrieve records, and invoke your functions. The following are supported services:

- Amazon Kinesis
- Amazon SQS
- Amazon DynamoDB Streams

AWS will manage the poller on your behalf and perform Synchronous invokes of your function with this type of integration. The retry behavior for this model is based on data expiration in the data source. For example, Kinesis Data streams store records for 24 hours by default (up to 168 hours). The specific details of each integration are linked above.

Polling and batching streams

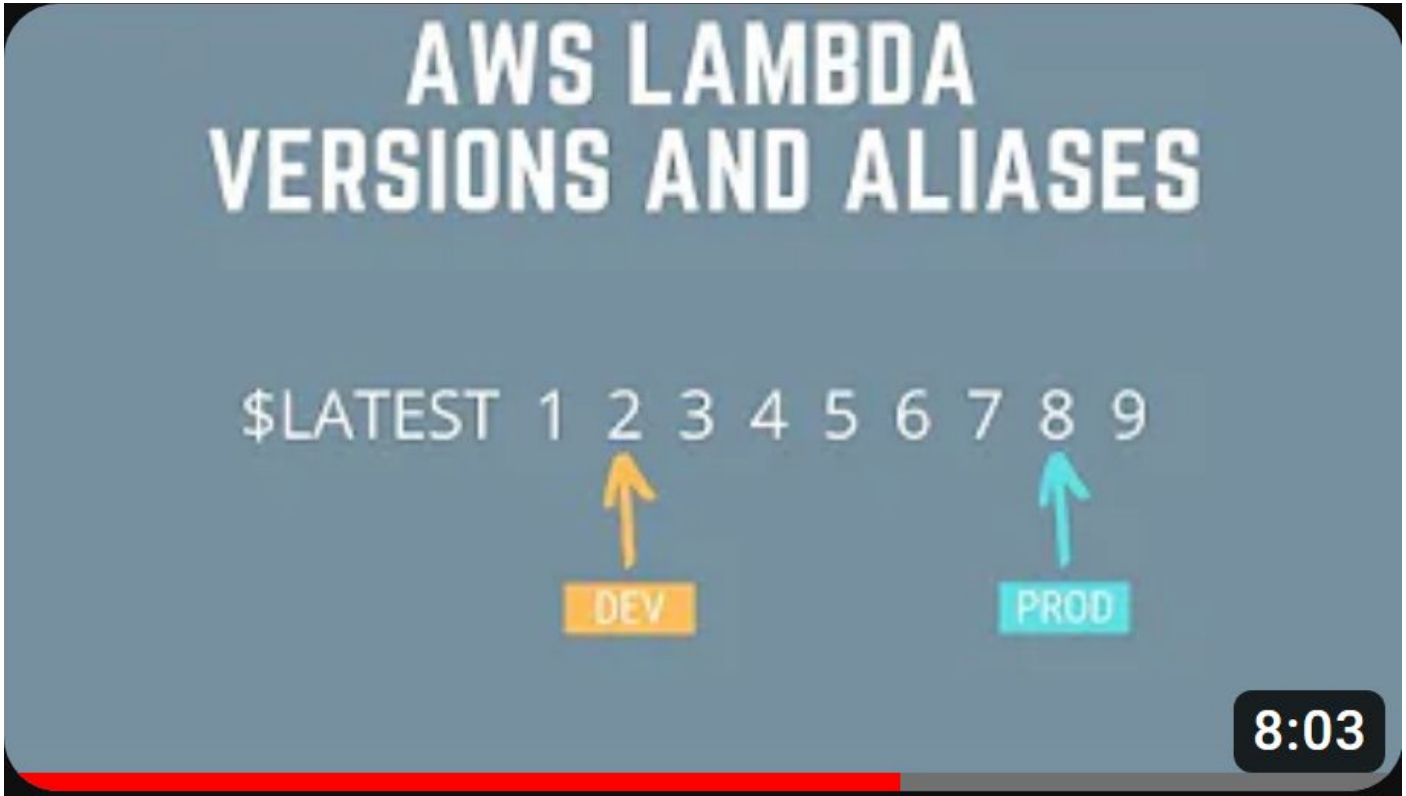
Lambda polls shards in your DynamoDB stream for records at a base rate of 4 times per second. When records are available, Lambda invokes your function and waits for the result. If processing succeeds, Lambda resumes polling until it receives more records.

By default, Lambda invokes your function as soon as records are available. If the batch that Lambda reads from the event source has only one record in it, Lambda sends only one record to the function. To avoid invoking the function with a small number of records, you can tell the event source to buffer records for up to 5 minutes by configuring a *batching window*. Before invoking the function, Lambda continues to read records from the event source until it has gathered a full batch, the batching window expires, or the batch reaches the payload limit of 6 MB. For more information, see Batching behavior.

If your function returns an error, Lambda retries the batch until processing succeeds or the data expires. To avoid stalled shards, you can configure the event source mapping to retry with a smaller batch size, limit the number of retries, or discard records that are too old. To retain discarded events, you can configure the event source mapping to send details about failed batches to a standard SQS queue or standard SNS topic.

You can also increase concurrency by processing multiple batches from each shard in parallel. Lambda can process up to 10 batches in each shard simultaneously. If you increase the number of concurrent batches per shard, Lambda still ensures in-order processing at the partition key level.

Configure the `ParallelizationFactor` setting to process one shard of a Kinesis or DynamoDB data stream with more than one Lambda invocation simultaneously. You can specify the number of concurrent batches that Lambda polls from a shard via a parallelization factor from 1 (default) to 10. For example, when you set `ParallelizationFactor` to 2, you can have 200 concurrent Lambda invocations at maximum to process 100 Kinesis data shards. This helps scale up the processing throughput when the data volume is volatile and the `IteratorAge` is high. Note that parallelization factor will not work if you are using Kinesis aggregation. For more information, see New AWS Lambda scaling controls for Kinesis and DynamoDB event sources. Also, see the Serverless Data Processing on AWS workshop for complete tutorials.



AWS Lambda Versions and Aliases Explained