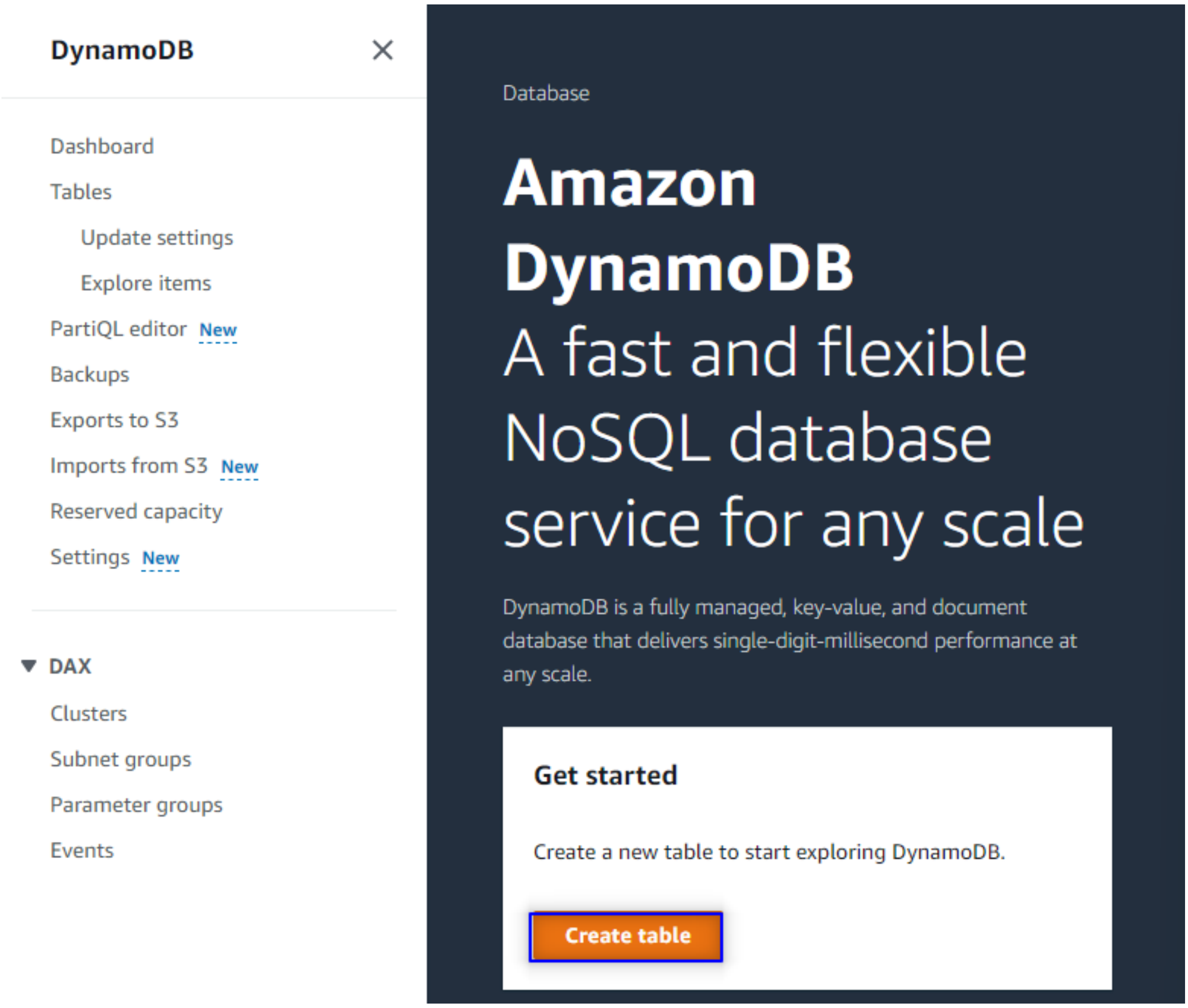
Phase 1: Create the DynamoDB Table.

Step 1: Navigate to the **DynamoDB Dashboard** and click *Create table*.



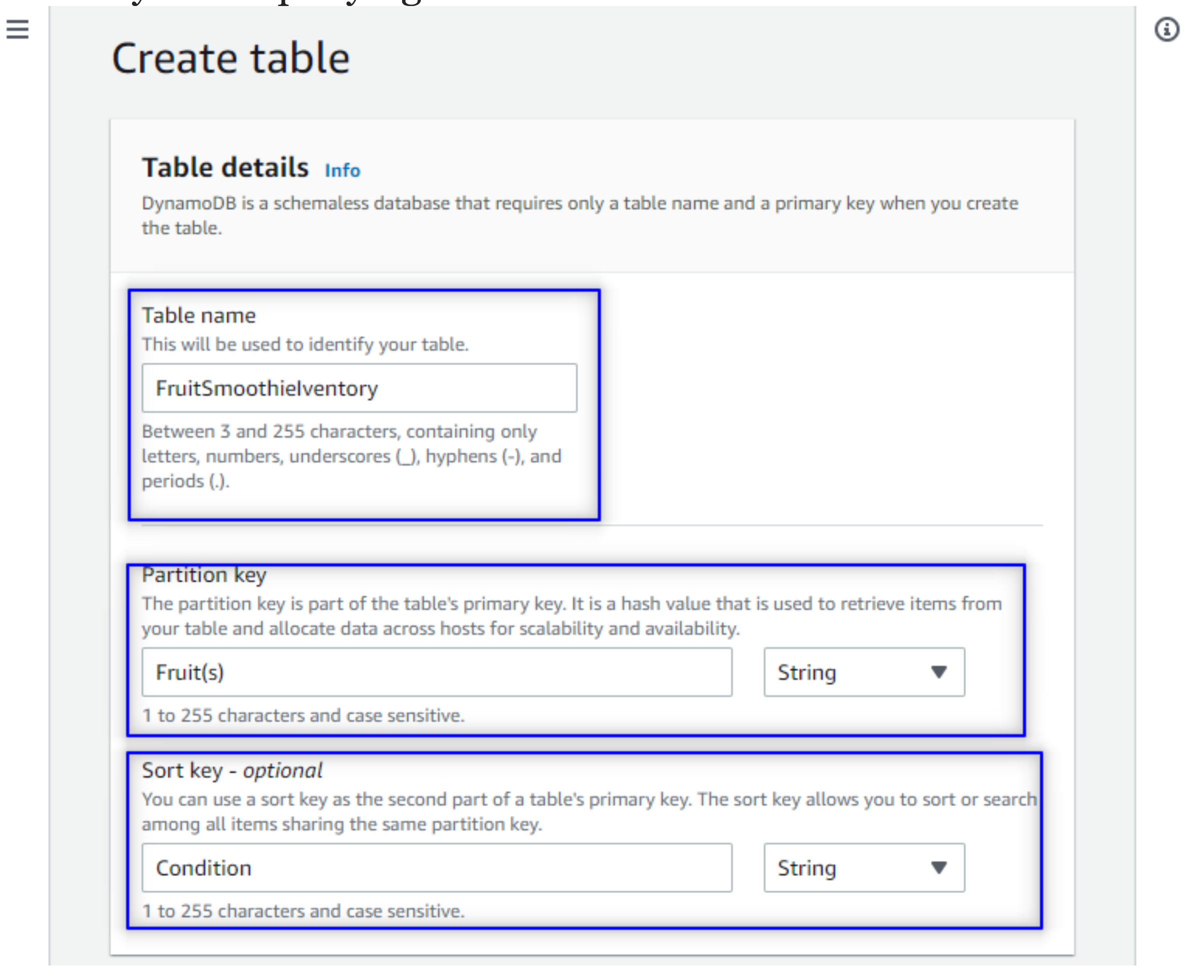
A fast and flexible NoSQL database service for any scale

DynamoDB is a fully managed, key-value, and document database that delivers single-digit-millisecond performance at any scale.

Create a new table to start exploring DynamoDB.

Step 2: Now we'll create the table by putting in names for the *Table*, *Partition key* and *Sort key*. Leave all the default settings then click *Create Table*.

What are these keys? A ***Partition key*** is essentially the primary key of the table. It must be unique because it logically separates the table into different partitions (physical storage internal to Dynamo DB). A ***Sort key*** is used when duplicate partition keys exist. The sort keys must be different for the duplicate partitions and together they make a ***Composite key***. A composite key gives additional flexibility when querying data.



Create table

Table details Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name

This will be used to identify your table.

FruitSmoothieIventory

Between 3 and 255 characters, containing only letters, numbers, underscores (_), hyphens (-), and periods (.).

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

Fruit(s) String ▼

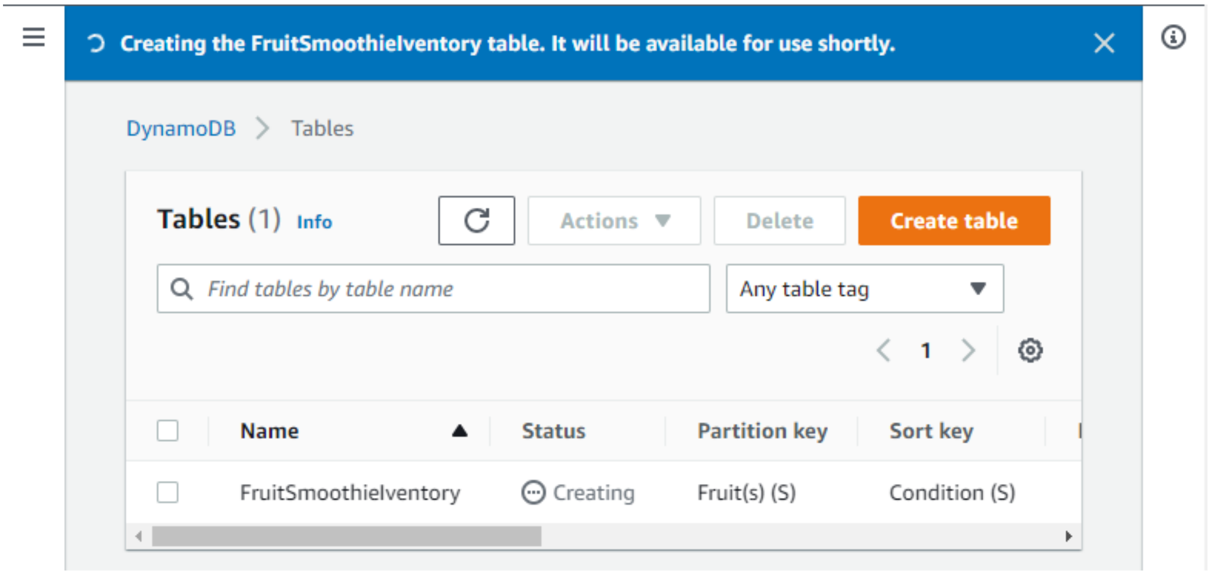
1 to 255 characters and case sensitive.

Sort key - *optional*

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

Condition String ▼

1 to 255 characters and case sensitive.



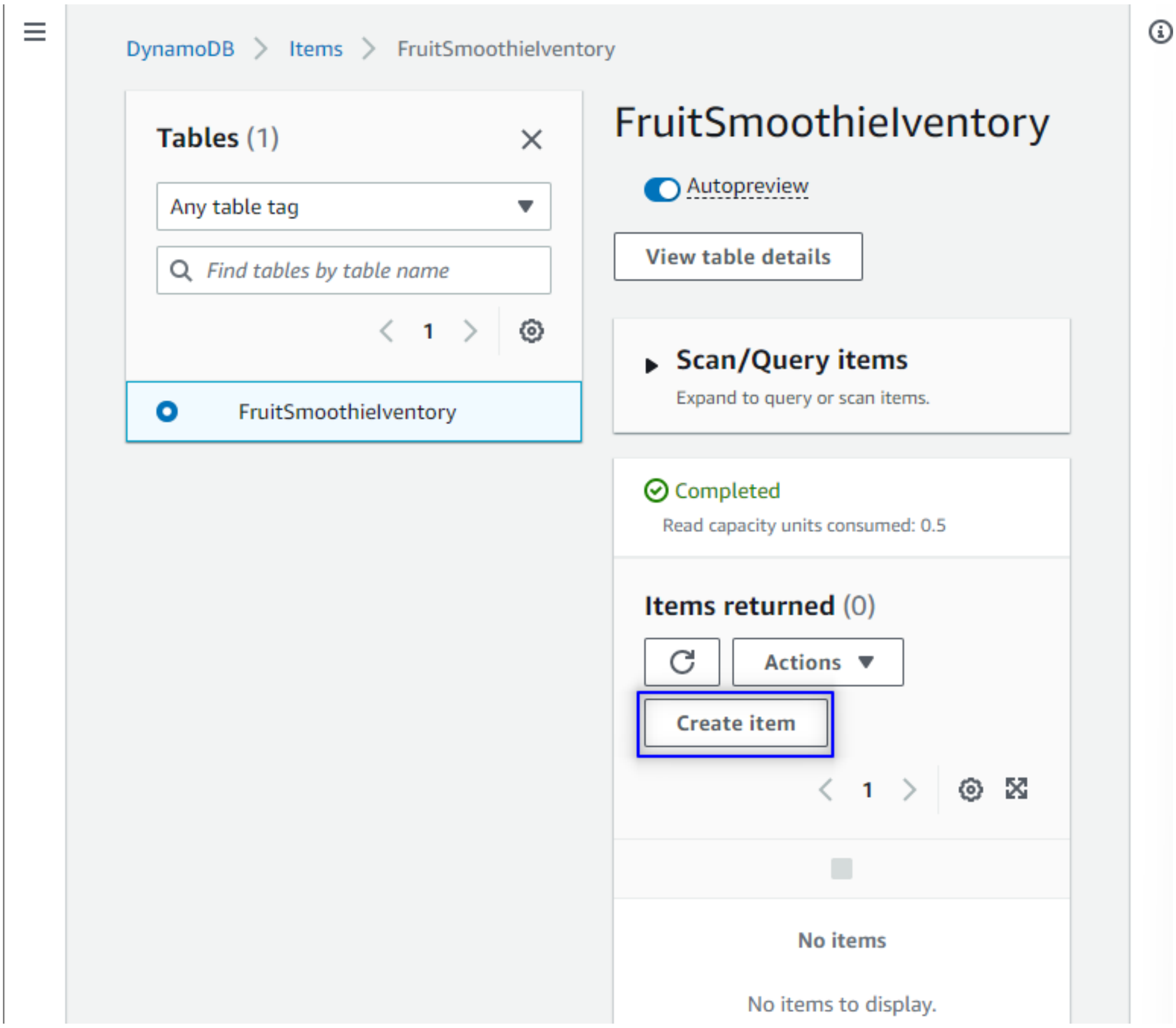
Creating the FruitSmoothieIventory table. It will be available for use shortly.

DynamoDB > Tables

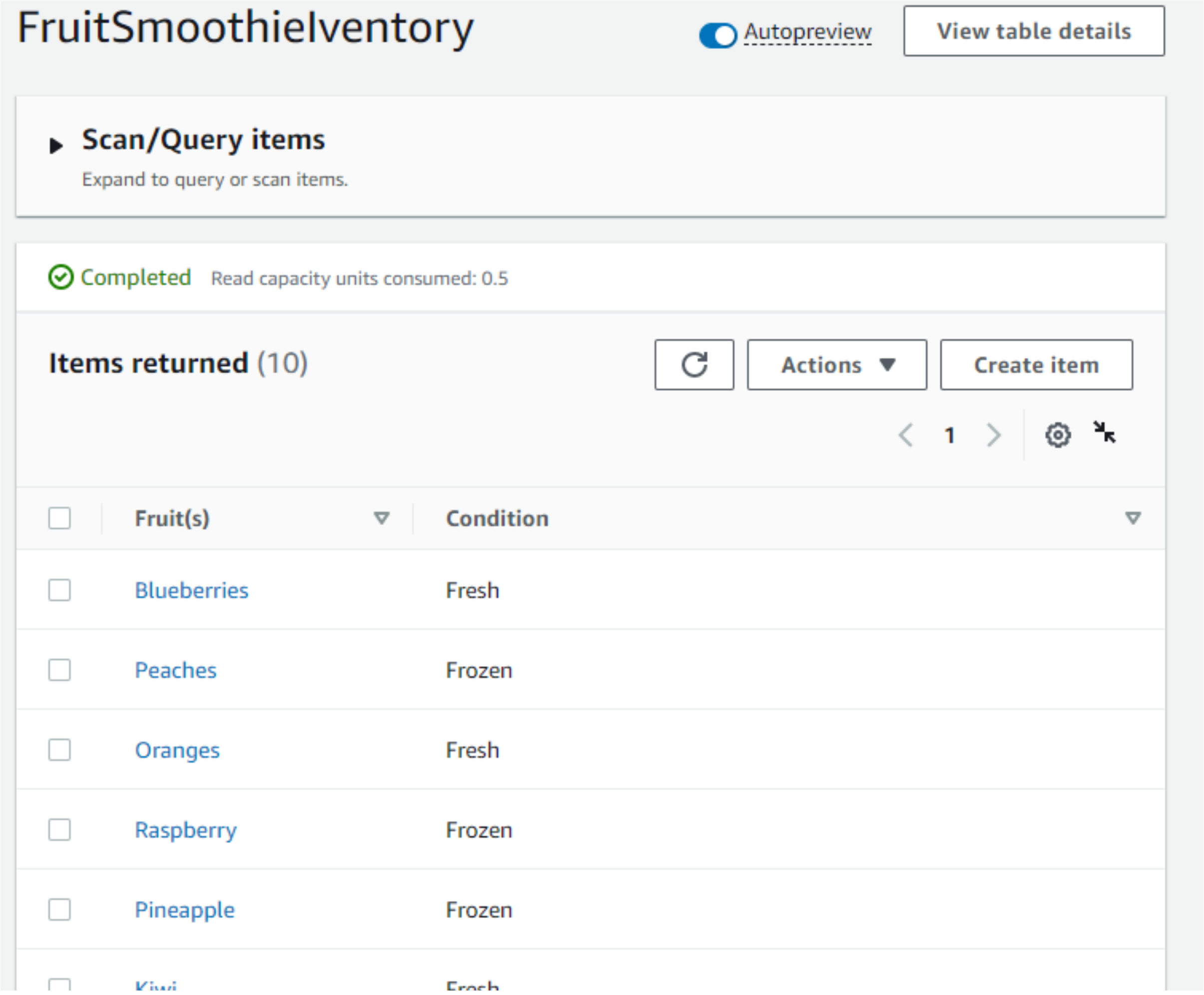
Tables (1) Info

□	Name ▲	Status	Partition key	Sort key
□	FruitSmoothieIventory	Creating	Fruit(s) (S)	Condition (S)

Step 3: In the DynamoDB menu click *Explore Items*, select the table then click *Create Item*.



Step 4: Input all 10 of your items into the table.



Completed Read capacity units consumed: 0.5

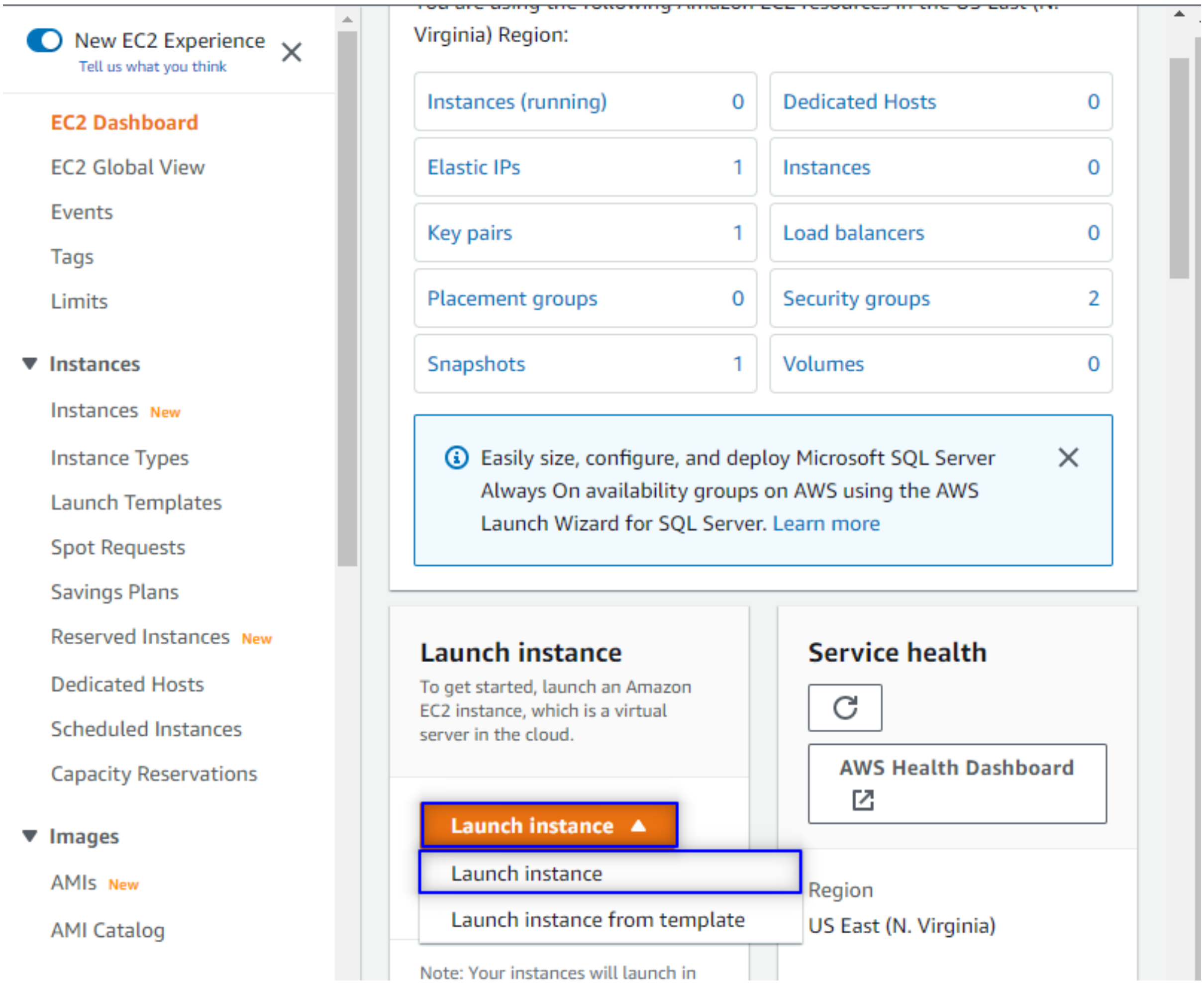
Items returned (10)

	Fruit(s)	Condition
□	Blueberries	Fresh
□	Peaches	Frozen
□	Oranges	Fresh
□	Raspberry	Frozen
□	Pineapple	Frozen
□	Kiwi	Fresh

Completed Table

Phase 2: Create a t2.micro EC2 Instance along with IAM role.

Step 1: In the AWS Management Console head over to the **EC2 Dashboard** and click *Launch Instance.*



New EC2 Experience
Tell us what you think

- EC2 Dashboard
- EC2 Global View
- Events
- Tags
- Limits
- ▼ Instances
  - Instances New
  - Instance Types
  - Launch Templates
  - Spot Requests
  - Savings Plans
  - Reserved Instances New
  - Dedicated Hosts
  - Scheduled Instances
  - Capacity Reservations
- ▼ Images
  - AMIs New
  - AMI Catalog

Virginia) Region:

Instances (running)	0	Dedicated Hosts	0
Elastic IPs	1	Instances	0
Key pairs	1	Load balancers	0
Placement groups	0	Security groups	2
Snapshots	1	Volumes	0

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS Launch Wizard for SQL Server. Learn more

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance ▲
- Launch instance
- Launch instance from template

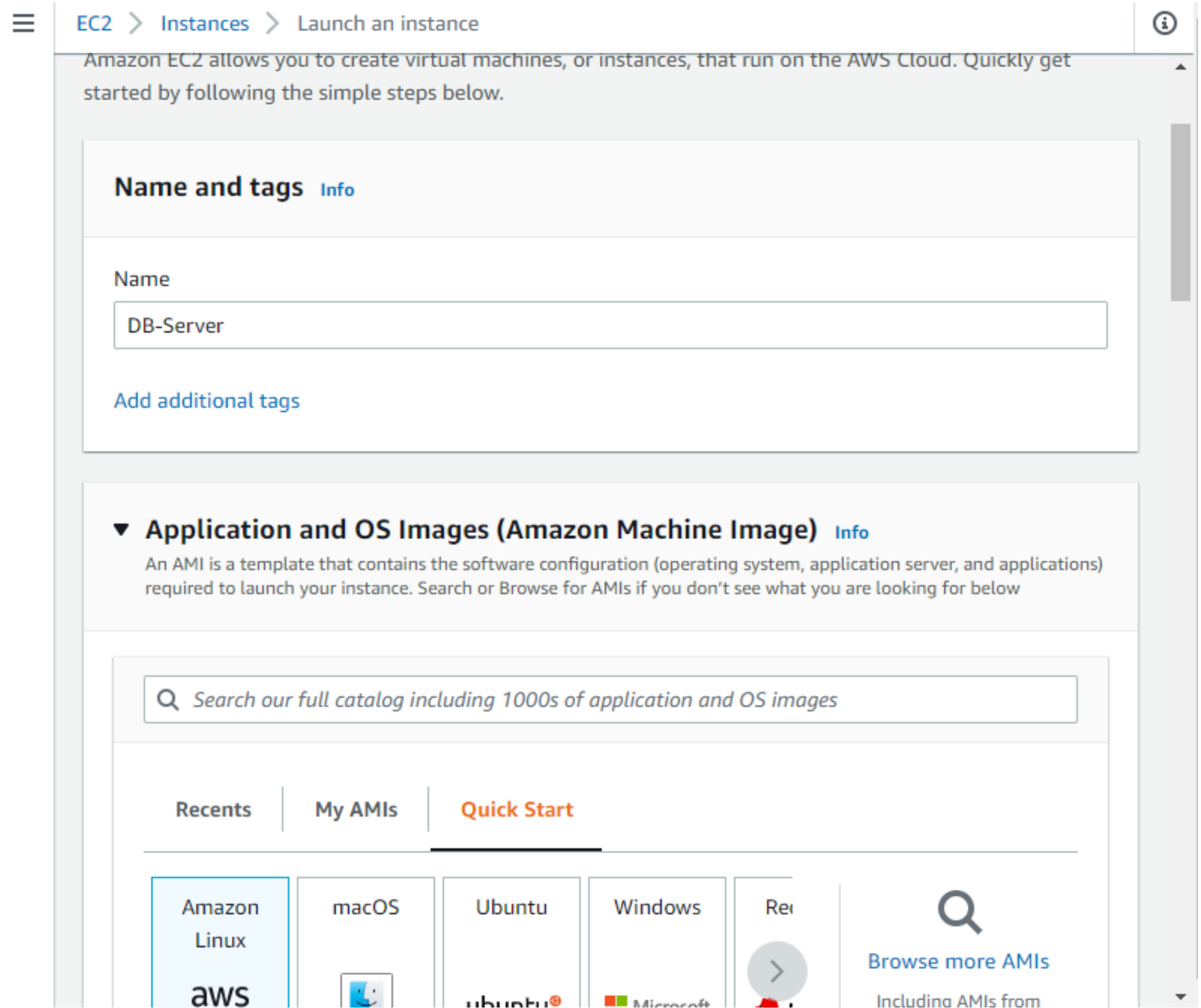
Note: Your instances will launch in

Service health

AWS Health Dashboard

Region
US East (N. Virginia)

Step 2: Name your server and choose the appropriate configurations. You can reference this article to see how it’s done Installing Apache Web Server on AWS EC2.



Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name

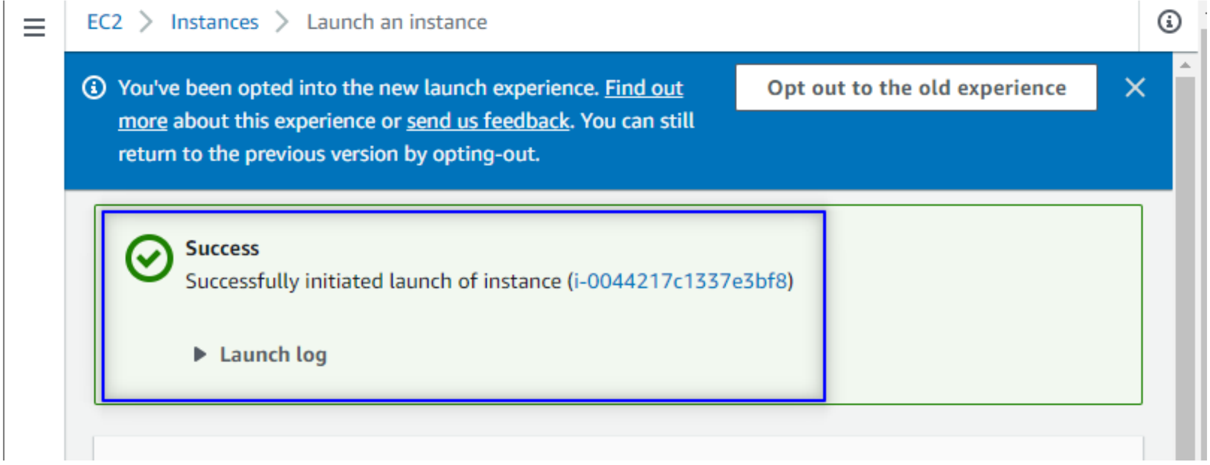
DB-Server

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

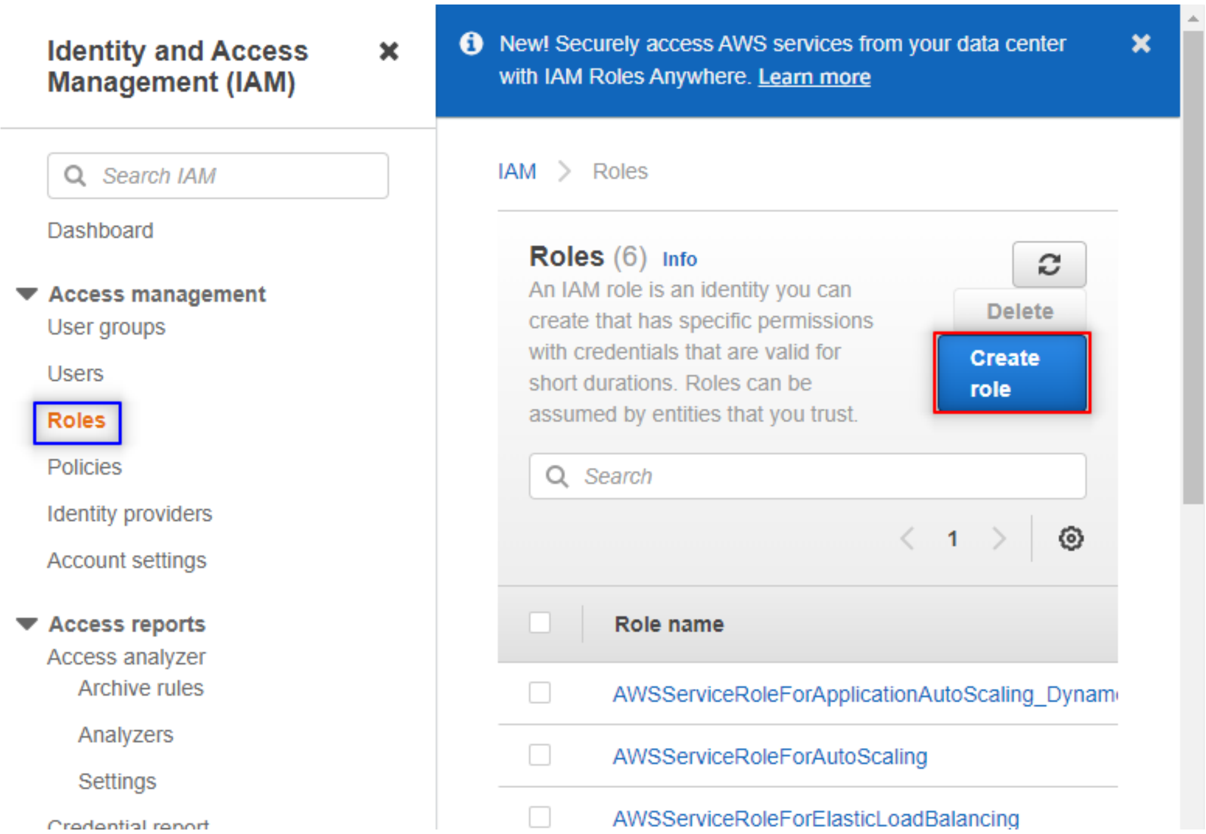


You've been opted into the new launch experience. Find out more about this experience or send us feedback. You can still return to the previous version by opting-out.

Success

Successfully initiated launch of instance (i-0044217c1337e3bf8)

Step 3: Navigate to IAM Dashboard and select *Roles* then click *Create Role*.



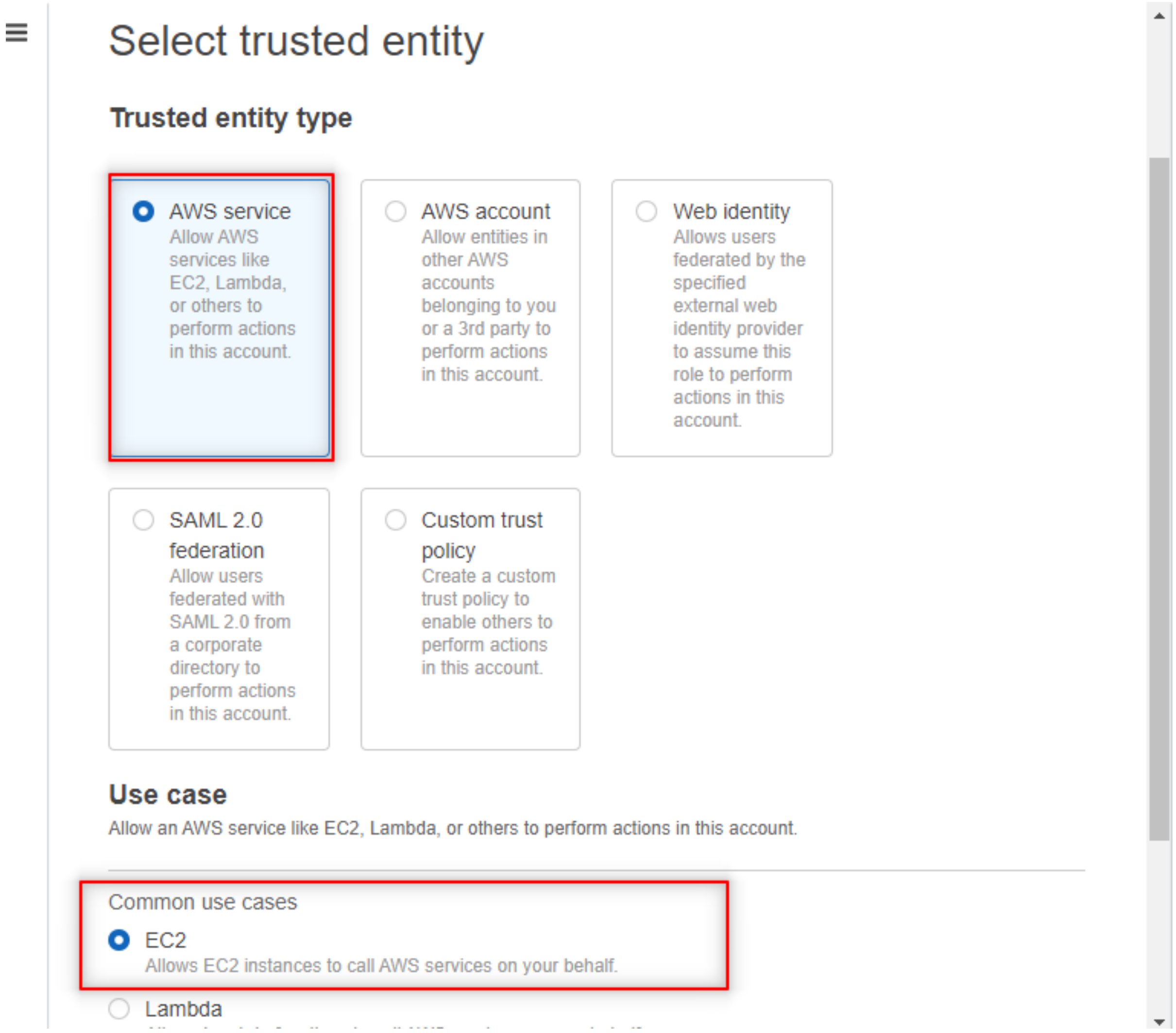
New! Securely access AWS services from your data center with IAM Roles Anywhere. Learn more

Roles (6) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

□	Role name
□	AWSServiceRoleForApplicationAutoScaling_Dynam
□	AWSServiceRoleForAutoScaling
□	AWSServiceRoleForElasticLoadBalancing

Step 4: Select *AWS service* as the **Trusted entity type** and *EC2* as the **Use Case**.



Select trusted entity

Trusted entity type

● AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

○ AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

○ Web identity
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

○ SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

○ Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

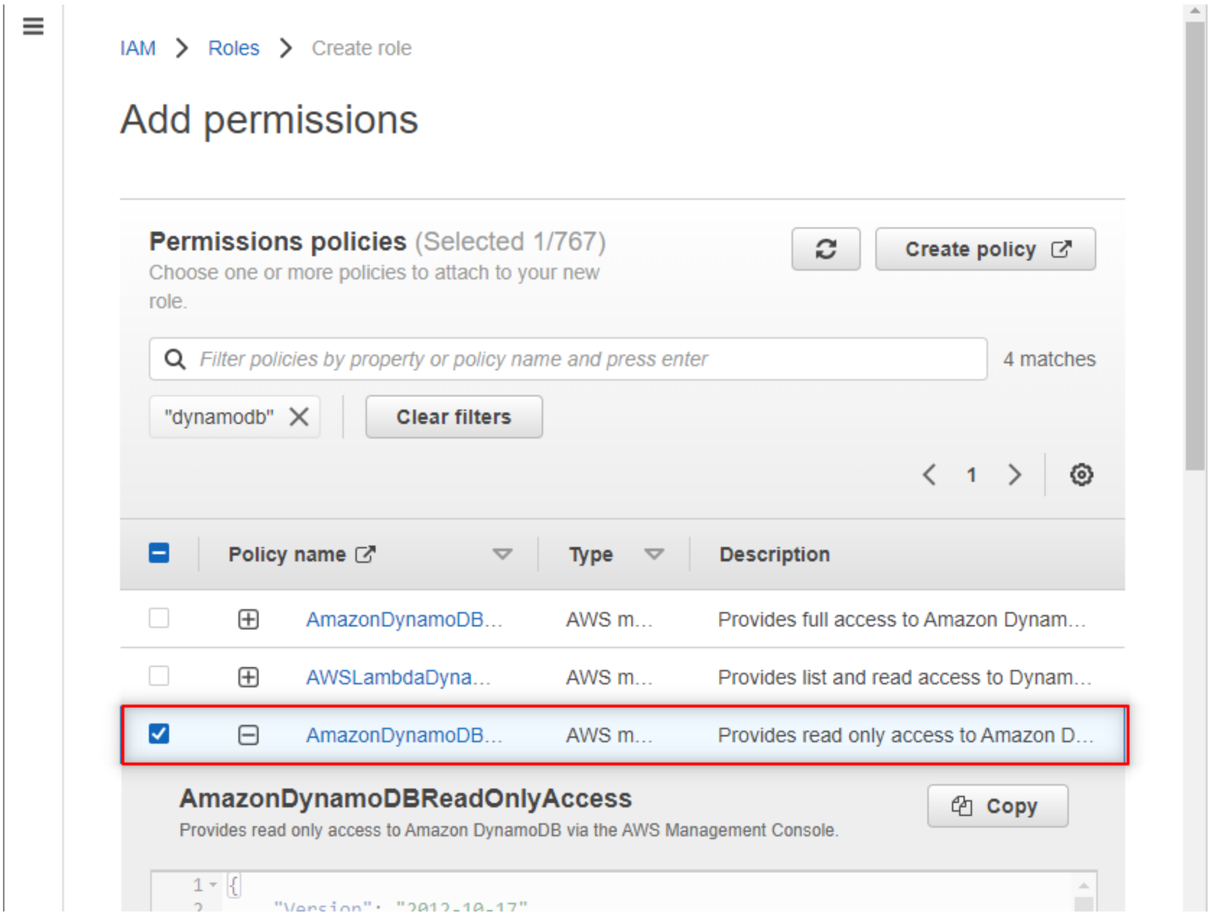
Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

● EC2
Allows EC2 instances to call AWS services on your behalf.

○ Lambda

Step 5: Choose the necessary permission for this case (we only need read access to the DynamoDB table from the EC2 Instance). {AmazonDynamoDBReadOnlyAccess}



IAM > Roles > Create role

Add permissions

Permissions policies (Selected 1/767)

Choose one or more policies to attach to your new role.

Filter policies by property or policy name and press enter

4 matches

"dynamodb"

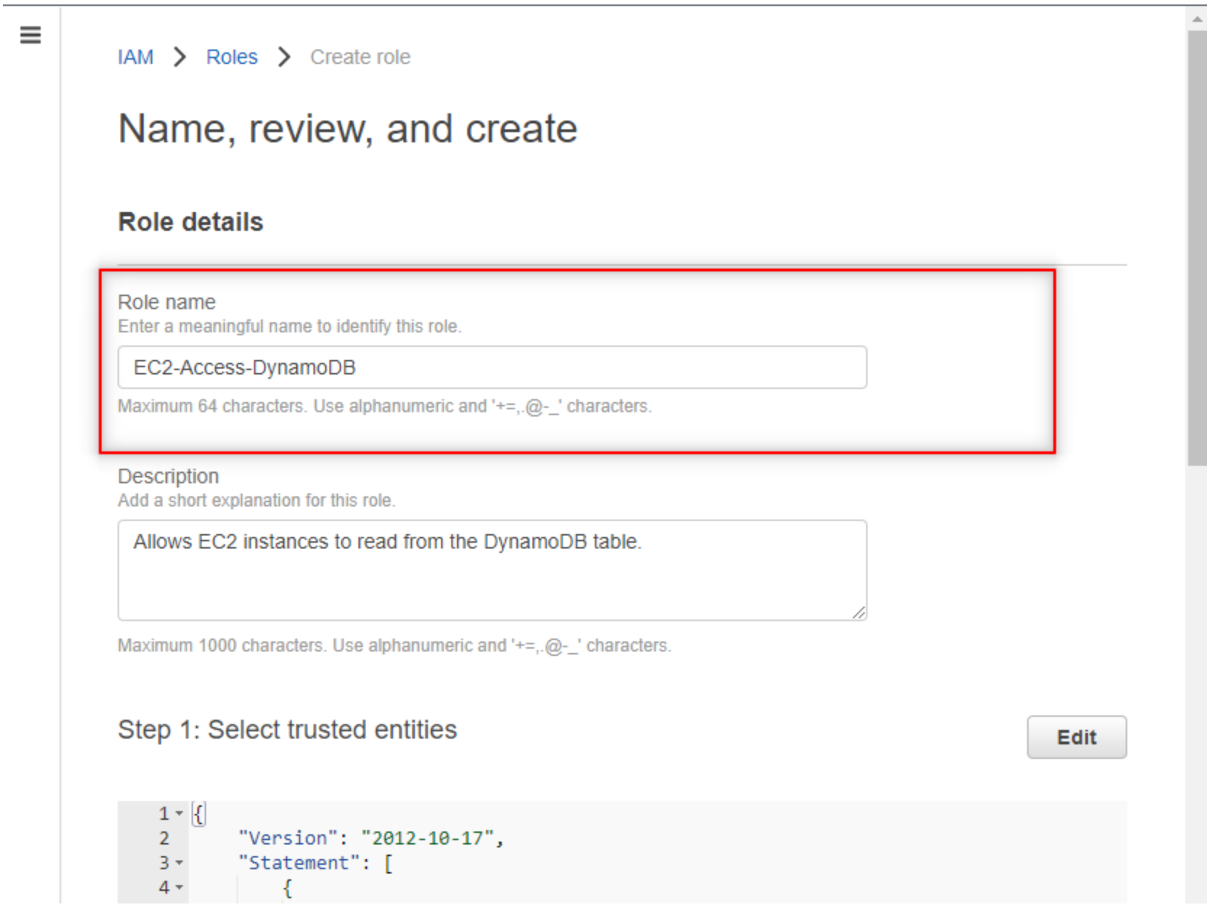
		Policy name	Type	Description
□	⊞	AmazonDynamoDB...	AWS m...	Provides full access to Amazon Dynam...
□	⊞	AWSLambdaDyna...	AWS m...	Provides list and read access to Dynam...
☑	⊟	AmazonDynamoDB...	AWS m...	Provides read only access to Amazon D...

AmazonDynamoDBReadOnlyAccess

Provides read only access to Amazon DynamoDB via the AWS Management Console.

```
{
    "Version": "2012-10-17"
```

Step 6: Name the role then click *Create role*.



Name, review, and create

Role details

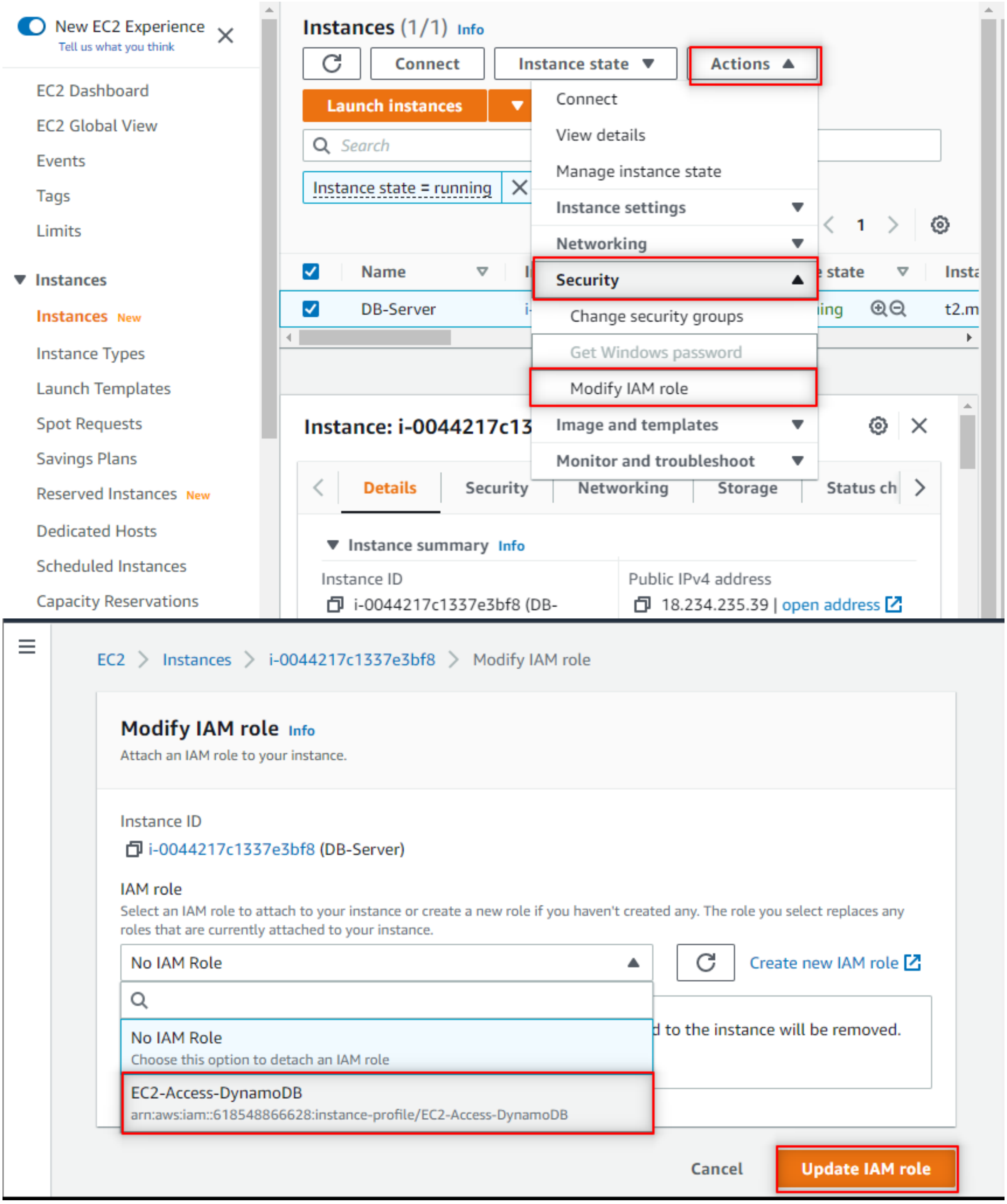
Role name
Enter a meaningful name to identify this role.
EC2-Access-DynamoDB
Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.
Allows EC2 instances to read from the DynamoDB table.
Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Step 1: Select trusted entities

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
```

Step 7: Head back to the EC2 Dashboard and select the newly created instance. Click *Actions* > *Security* > *Modify IAM role*.



Modify IAM role Info

Attach an IAM role to your instance.

Instance ID

i-0044217c1337e3bf8 (DB-Server)

IAM role

Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.

No IAM Role

No IAM Role
Choose this option to detach an IAM role

EC2-Access-DynamoDB
arn:aws:iam::618548866628:instance-profile/EC2-Access-DynamoDB

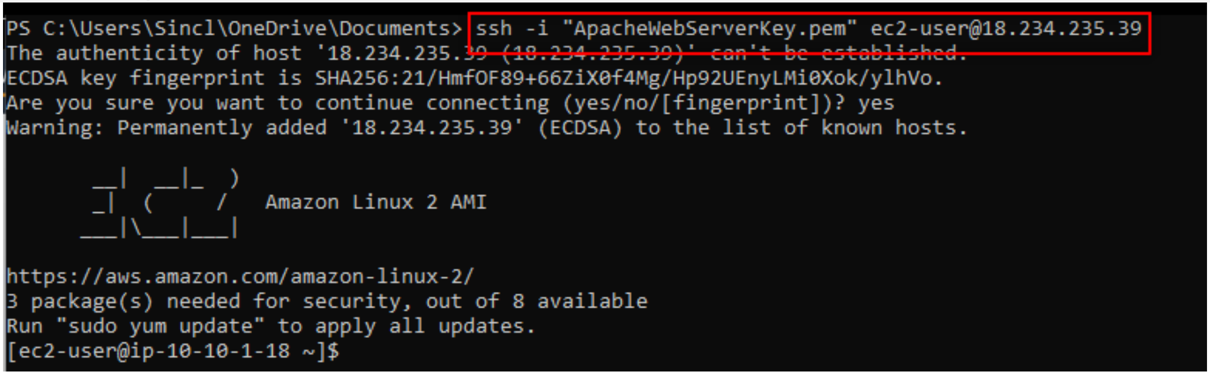
d to the instance will be removed.

Select the IAM Role

Phase 3: From the CLI scan the table and confirm permissions.

Step 1: With windows Powershell, ssh into the instance from your local machine.

```
cd (Directory containing Key Pair)ssh –i "Key Pair name"
ec2user@(Public IP address)
```



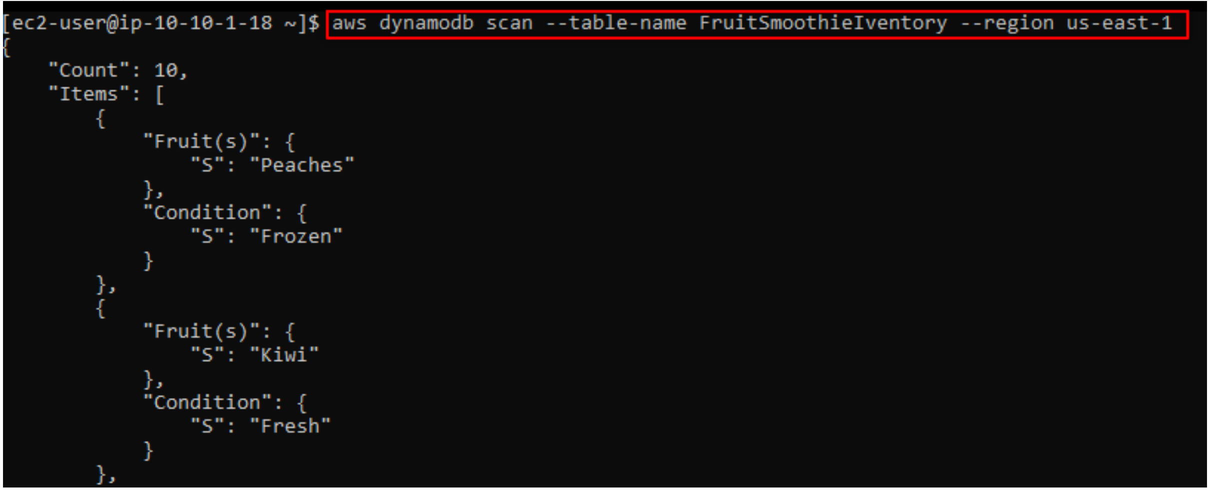
```
PS C:\Users\Sincl\OneDrive\Documents> ssh -i "ApacheWebServerKey.pem" ec2-user@18.234.235.39
The authenticity of host '18.234.235.39 (18.234.235.39)' can't be established.
ECDSA key fingerprint is SHA256:21/HmfOF89+66ZiX0f4Mg/Hp92UEnyLMi0Xok/ylhVo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '18.234.235.39' (ECDSA) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
3 package(s) needed for security, out of 8 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-10-1-18 ~]$
```

Step 2: Run command to scan the DynamoDB Table we created earlier

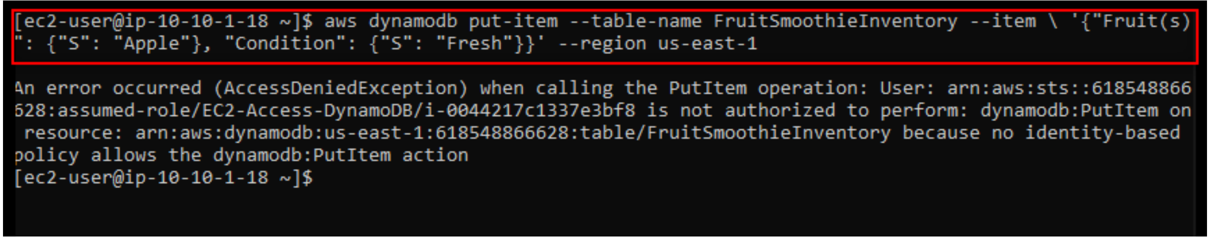
```
aws dynamodb scan ––table–name (name of table) ––region (Region
housing database)
```



```
[ec2-user@ip-10-10-1-18 ~]$ aws dynamodb scan --table-name FruitSmoothieIventory --region us-east-1
{
    "Count": 10,
    "Items": [
        {
            "Fruit(s)": {
                "S": "Peaches"
            },
            "Condition": {
                "S": "Frozen"
            }
        },
        {
            "Fruit(s)": {
                "S": "Kiwi"
            },
            "Condition": {
                "S": "Fresh"
            }
        },
```

Final Step: Let's attempt to write an item to the table to validate the IAM Role is operating correctly

```
aws dynamodb put–item ––table–name (Table Name) ––item \
'{"(Partition key name)": {"S": “(Value)"}, "(Sort key name)":
{"S": "(Value)"}}’ ––region (Region housing database)
```



```
[ec2-user@ip-10-10-1-18 ~]$ aws dynamodb put-item --table-name FruitSmoothieInventory --item \ '{"Fruit(s)
": {"S": "Apple"}, "Condition": {"S": "Fresh"}}' --region us-east-1

An error occurred (AccessDeniedException) when calling the PutItem operation: User: arn:aws:sts::618548866
628:assumed-role/EC2-Access-DynamoDB/i-0044217c1337e3bf8 is not authorized to perform: dynamodb:PutItem on
 resource: arn:aws:dynamodb:us-east-1:618548866628:table/FruitSmoothieInventory because no identity-based
policy allows the dynamodb:PutItem action
[ec2-user@ip-10-10-1-18 ~]$
```

Error message

Because we made an ***API call*** to create an item in the table, we received an error message. The IAM Role attached to the EC2 instance has only read permissions associated, so the Instance will fail to do anything but read the database.