

Services → iam → roles → create roles → AWS service → ec2 → select policy type (dynamodb full access)

Services → ec2 instance → launch instance → select amazon linux ami → AWS service → ec2 → select policy type (dynamodb full access) → configure instance (select IAM role created above, have the bootstrap scripts under advance) → add storage → add tags → configure security group (have http and ssh access) → select / create key pair for the region → launch instances

Select the instance and copy the public ip

```
Fayes-iMac:Downloads fayeellis$ ssh -i mylondonkeypair.pem ec2-user@35.176.23.146
```

```
  _ | _ | _ )  
  _ | ( _ | /  
  _ | \ _ | _ |
```

Amazon Linux AMI

```
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/  
[ec2-user@ip-172-31-25-86 ~]$ sudo su  
[root@ip-172-31-25-86 ec2-user]# cd /var/www/html  
[root@ip-172-31-25-86 html]# ls  
dynamodb  test.php  
[root@ip-172-31-25-86 html]#
```

## SDK for php – composer - installation steps

```
[ec2-user@ip-172-31-87-185 html]$ sudo su
[root@ip-172-31-87-185 html]# php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
[root@ip-172-31-87-185 html]# php -r "if (hash_file('sha384', 'composer-setup.php') === '93b54496392c062774670ac18b134c3b3a95e5a5e5c8f1a9f115f203b75bf9a129d5daa8ba6a13e2cc8a1da0806388a8') { echo 'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php'); } echo PHP_EOL;"
Installer verified
[root@ip-172-31-87-185 html]# php composer-setup.php
All settings correct for using Composer
Downloading...

Composer (version 1.8.0) successfully installed to: /var/www/html/composer.phar
Use it: php composer.phar

[root@ip-172-31-87-185 html]# php -r "unlink('composer-setup.php');"
[root@ip-172-31-87-185 html]#
```

## php - installation steps

```
[root@ip-172-31-87-185 html]# php -d memory_limit=-1 composer.phar require aws/aws-sdk-php
```

```
[root@ip-172-31-25-86 html]# ls
composer.json  composer.lock  composer.phar  dynamodb  test.php  vendor
[root@ip-172-31-25-86 html]# cd dynamodb
[root@ip-172-31-25-86 dynamodb]# ls
createtables.php  README.md  uploaddata.php
[root@ip-172-31-25-86 dynamodb]# 
[root@ip-172-31-25-86 dynamodb]# vi createtables.php
```

Createtables.php → create tables in that region  
Uploaddata.php → uploads the data in that region

```
// Date now needs to be set, which I guess is a good thing!
date_default_timezone_set('Europe/London');

// Find out what the issues are:
ini_set('display_errors',1);
ini_set('display_startup_errors',1);
error_reporting(-1);

require '/var/www/html/vendor/autoload.php';
use Aws\DynamoDb\DynamoDbClient;

$client = DynamoDbClient::factory(array(
    'region' => 'eu-west-2', // replace with your desired region visit
    'version' => '2012-08-10' // Now needs a version
));

$tableNames = array();
```

set the region / availability zone

Ec2 instance public ip

Php script to create table  
in dynamodb

35.176.23.146/dynamodb/createtables.php

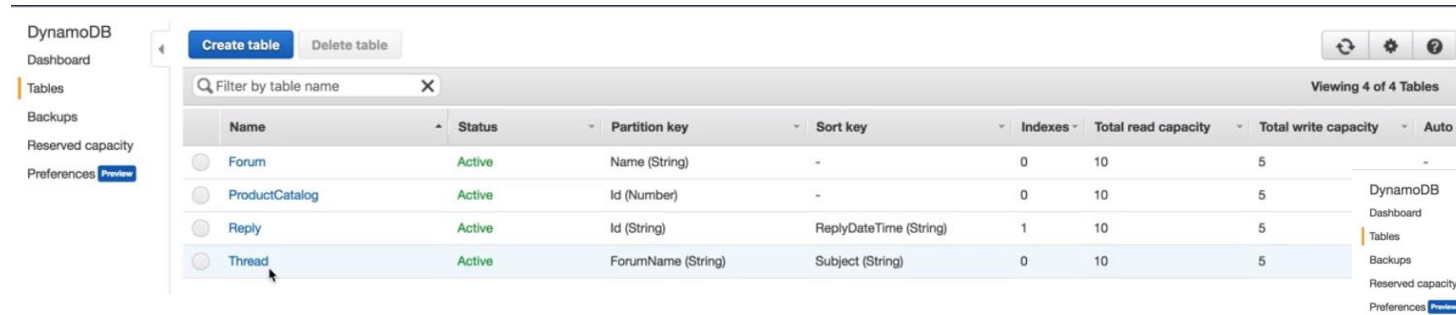
Creating table ProductCatalog... Creating table Forum... Creating table Thread... Creating table Reply... Waiting for table ProductCatalog to be created. Table ProductCatalog has been created. Waiting for table Forum to be created. Table Forum has been created. Waiting for table Thread to be created. Table Thread has been created. Waiting for table Reply to be created. Table Reply has been created.

Php script to upload data  
to the table in dynamodb

35.176.23.146/dynamodb/uploaddata.php

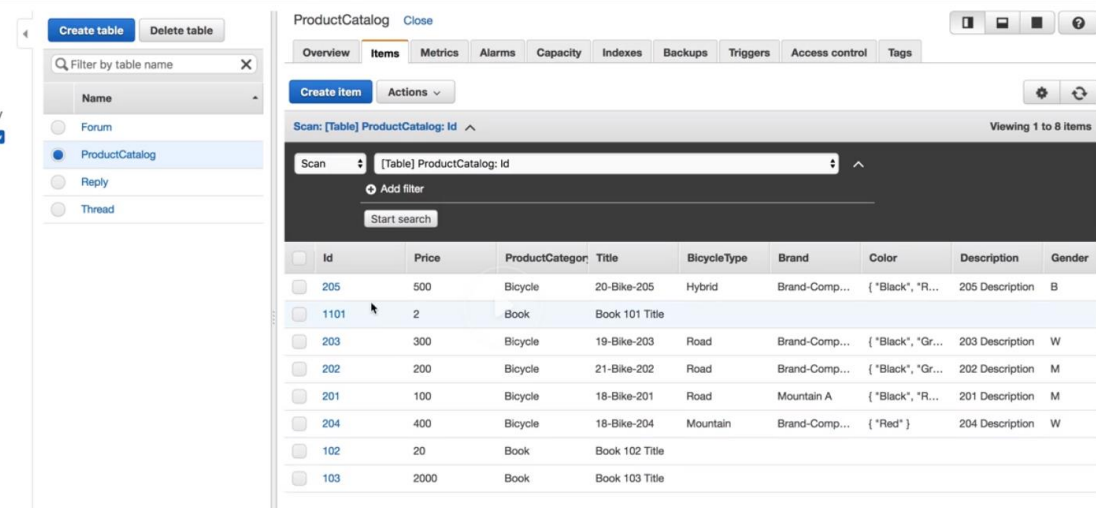
Adding data to the ProductCatalog table... done. Adding data to the Forum table... done. Adding data to the Reply table... done.

Services → database → dynamodb → tables



The screenshot shows the AWS DynamoDB console 'Tables' page. It displays a list of four tables: Forum, ProductCatalog, Reply, and Thread. Each table is in an 'Active' state. The 'Thread' table is selected, and its details are shown on the right side of the console.

Name	Status	Partition key	Sort key	Indexes	Total read capacity	Total write capacity	Auto
Forum	Active	Name (String)	-	0	10	5	-
ProductCatalog	Active	Id (Number)	-	0	10	5	-
Reply	Active	Id (String)	ReplyDateTime (String)	1	10	5	-
Thread	Active	ForumName (String)	Subject (String)	0	10	5	-



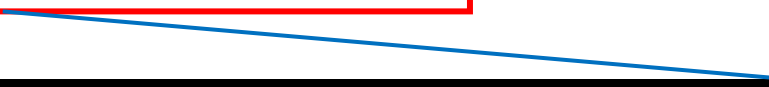
The screenshot shows the details of the 'ProductCatalog' table in the AWS DynamoDB console. The 'Items' tab is selected, displaying a list of items. The table has a primary key 'Id' and a secondary index 'ProductCatalog: Id'. The items are displayed in a table format.

Id	Price	ProductCategory	Title	BicycleType	Brand	Color	Description	Gender
205	500	Bicycle	20-Bike-205	Hybrid	Brand-Comp...	{ 'Black', 'R...	205 Description	B
1101	2	Book	Book 101 Title					
203	300	Bicycle	19-Bike-203	Road	Brand-Comp...	{ 'Black', 'Gr...	203 Description	W
202	200	Bicycle	21-Bike-202	Road	Brand-Comp...	{ 'Black', 'Gr...	202 Description	M
201	100	Bicycle	18-Bike-201	Road	Mountain A	{ 'Black', 'R...	201 Description	M
204	400	Bicycle	18-Bike-204	Mountain	Brand-Comp...	{ 'Red' }	204 Description	W
102	20	Book	Book 102 Title					
103	2000	Book	Book 103 Title					



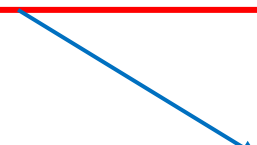
Interacting with dynamodb via CLI

This is possible as we have set IAM role with policy dynamo db full access



```
[root@ip-172-31-29-76 html]# aws dynamodb get-item --table-name ProductCatalog --region eu-west-2 --key '{"Id":{"N":"205"}}'
```

If we want ec2 instance access dynamodb, we need to create IAM service role. Otherwise, ec2 instance wouldn't connect with dynamodb



```
[root@ip-172-31-29-76 html]# aws dynamodb get-item --table-name ProductCatalog --region eu-west-2 --key '{"Id":{"N":"205"}}'
```

```
An error occurred (UnrecognizedClientException) when calling the GetItem operation: The security token included in the request is invalid.  
[root@ip-172-31-29-76 html]#
```