

The screenshot shows the AWS homepage with a navigation bar at the top. The main content area displays several service categories: Compute (EC2, Lambda), Storage & Content Delivery (S3, CloudFront, Elastic File System, CloudTrail, Config, Snowball, Storage Gateway), Database (RDS, DynamoDB, ElastiCache, Redshift, DMS), and others like Developer Tools, Internet of Things, Game Development, Management Tools, Mobile Services, Application Services, Security & Identity, and Service Health.

Sign In or Create an AWS Account

What is your email (phone for mobile accounts)?

E-mail or mobile number:

I am a new user.

I am a returning user and my password is:

[Sign in using our secure server](#)

[Forgot your password?](#)

AWS Accounts Include
12 Months of Free Tier Access

Including use of Amazon EC2, Amazon S3, and Amazon RDS

Visit aws.amazon.com/free for full offer terms.

Learn more about [AWS Identity and Access Management](#) and [AWS Multi-Factor Authentication](#), features that provide additional security for your AWS Account. View full [AWS Free Usage Tier](#) offer terms.

About Amazon.com Sign In

Amazon Web Services uses information from your Amazon.com account to identify you and allow access to Amazon Web Services on this site. You can change your account settings for how AWS Customer Support links back to your Amazon.com account or [View your account settings](#).

Amazon recommends that you use your root user to set up other users that should have varying access to do certain things in your AWS account and not use the root user for working.

The screenshot shows the AWS Identity and Access Management (IAM) dashboard. The left sidebar includes links for Dashboard, Search IAM, Details, Groups, Users, Roles, Policies, Identity Providers, Account Settings, Credential Report, and Encryption Keys. The main content area features a "Welcome to Identity and Access Management" header and a link to the IAM sign-in page. It displays IAM Resources: 0 users, 1 group, 0 roles, 0 identity providers, and 5 customer managed policies. A "Security Status" bar indicates 2 out of 5 items are complete. Below are five tasks: "Delete your root access keys" (completed), "Activate MFA on your root account" (in progress), "Create individual IAM users" (in progress), "Use groups to assign permissions" (completed), and "Apply an IAM password policy" (in progress). To the right, a "Feature Spotlight" box shows a video thumbnail titled "Introduction to AWS IAM". The bottom right contains additional information links: IAM documentation, Web Identity Federation Playground, Policy Simulator, Videos, IAM release history, and additional resources.

The screenshot shows the AWS IAM service interface. The top navigation bar includes 'AWS Services' and 'Edit' dropdowns, and user information 'Matt Thomas - Global Support'. On the left, a sidebar lists 'Dashboard', 'Search IAM', 'Details', 'Groups', 'Users' (which is selected and highlighted in orange), 'Roles', 'Policies', 'Identity Providers', 'Account Settings', and 'Credential Report'. The main content area has tabs 'Create New Users' and 'User Actions'. A search bar with 'Filter' placeholder text is present. Below it is a table header with columns: 'User Name', 'Groups', 'Password', 'Password Last Used', 'Access Keys', and 'Creation Time'. The message 'Showing 0 results' is displayed above the table, and 'No records found.' is shown below it.

Note that the root user does not come up under the Users list in your account. The Users list contains the list of users that you have created as the root account.

Create User

Enter User Names:

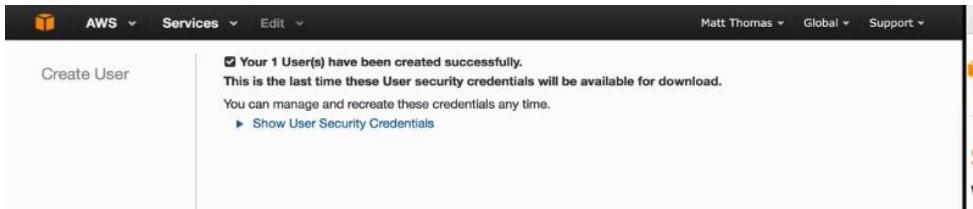
1. BillyBob
2. []
3. []
4. []
5. []

Maximum 64 characters each

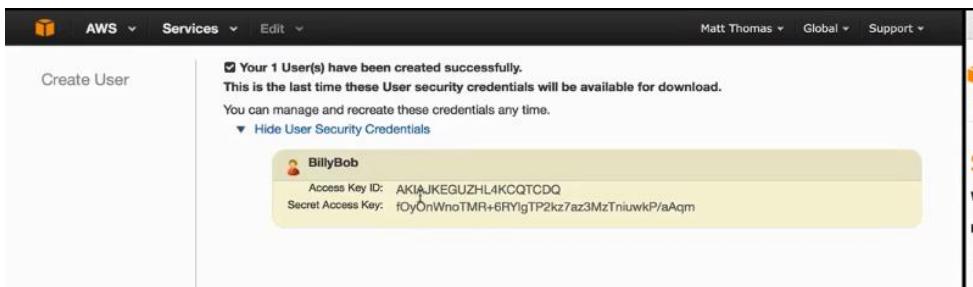
Generate an access key for each user

Users need access keys to make secure REST or Query protocol requests to AWS service APIs.

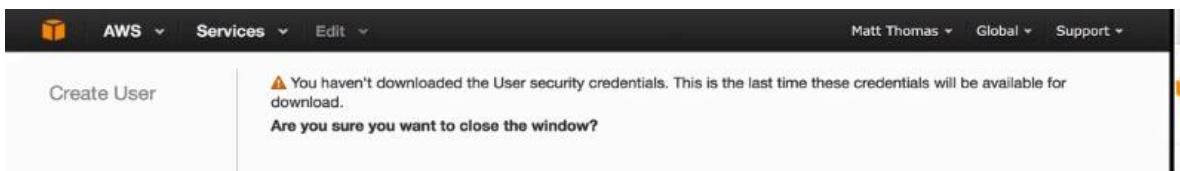
For users who need access to the AWS Management Console, create a password in the Users panel after completing this wizard.



You have now created a new user called BillyBob



You can now download the **access credentials** for this user for logging into the AWS Console or AWS CLI to do work. This **access credentials** of the (**Access Key ID and the Secret Access Key**) can be used for signing API requests when using it for an API-level access for your app by using the **Secret Access Key** in your application code to sign automated requests to AWS resources for authentication.



Dashboard

Services

Edit

Matt Thomas Global Support

Create New Users						User Actions
	User Name	Groups	Password	Password Last Used	Access Keys	Creation Time
<input type="checkbox"/>	BillyBob	0	N/A	1 active	2016-07-07 08...	
BillyBob						

Showing 1 results

Details

Groups

Users

Roles

Policies

Identity Providers

Account Settings

Credential Report

Encryption Keys

We have created a user called BillyBob, let us now set username and passwords for this user to use.

AWS Services Edit Matt Thomas Global Support

IAM > Users > BillyBob

Summary

User ARN: arn:aws:iam::515331423217:user/BillyBob
Has Password: No
Groups (for this user): 0
Path: /
Creation Time: 2016-07-07 08:10 UTC+0100

Groups Permissions Security Credentials Access Advisor

This user does not belong to any groups.

Add User to Groups

Dashboard Search IAM Details Groups Users Roles Policies Identity Providers Account Settings Credential Report Encryption Keys

AWS Services Edit Matt Thomas Global Support

IAM > Users > BillyBob

Summary

User ARN: arn:aws:iam::515331423217:user/BillyBob
Has Password: No
Groups (for this user): 0
Path: /
Creation Time: 2016-07-07 08:10 UTC+0100

Groups Permissions Security Credentials Access Advisor

Access Keys

Use access keys to make secure REST or Query protocol requests to any AWS service API. For your protection, you should never share your secret keys with anyone. In addition, industry best practice recommends frequent key rotation. Learn more about Access Keys.

Create Access Key

Access Key ID	Created	Last Used	Last Used Service	Last Used Region	Status	Actions
AKIAJKEGUZHL4KCQTCDO	2016-07-07 08:10 UTC+0100	N/A	N/A	N/A	Active	Make Inactive Delete

Sign-In Credentials

User Name: BillyBob Manage Password
Password: No
Last Used: N/A

Multi-Factor Authentication Device Manage MFA Device

Signing Certificates Manage Signing Certificates

SSH keys for AWS CodeCommit

Use SSH public keys to authenticate to AWS CodeCommit repositories. Learn more about SSH keys.
No SSH public keys are associated with this user.

Upload SSH public key

Dashboard Search IAM Details Groups Users Roles Policies Identity Providers Account Settings Credential Report Encryption Keys

AWS Services Edit Matt Thomas Global Support

Manage Password

Users who will be using the AWS Management Console require a password. Select from the options below to manage the password for user BillyBob.

Assign an auto-generated password
 Assign a custom password

Password:
Confirm Password:

Require user to create a new password at next sign-in

User ARN: arn:aws:iam::515331423217:user/BillyBob

Has Password: Yes

Groups (for this user): 0

Path: /

Creation Time: 2016-07-07 08:10 UTC+0100

Groups **Permissions** **Security Credentials** **Access Advisor**

Access Keys

Use access keys to make secure REST or Query protocol requests to any AWS service API. For your protection, you should never share your secret keys with anyone. In addition, industry best practice recommends frequent key rotation. Learn more about Access Keys

Create Access Key

Access Key ID	Created	Last Used	Last Used Service	Last Used Region	Status	Actions
AKIAJKEGUZHL4KCQTCGQ	2016-07-07 08:10 UTC+0100	N/A	N/A	N/A	Active	Make Inactive Delete

Sign-In Credentials

User Name: BillyBob

Manage Password

Password: Yes

Last Used: Never

Multi-Factor Authentication Device: No

Manage MFA Device

Signing Certificates: None

Manage Signing Certificates

SSH keys for AWS CodeCommit

Use SSH public keys to authenticate to AWS CodeCommit repositories. Learn more about SSH keys.

No SSH public keys are associated with this user.

Upload SSH public key

BillyBob now has a password

Welcome to Identity and Access Management

IAM users: 1 press Command + C to copy <https://515331423217.signin.aws.amazon.com/console> Customize | Copy Link

IAM Resources

- Users: 1
- Groups: 1
- Customer Managed Policies: 5

Security Status 3 out of 5 complete.

- Checkmark** Delete your root access keys
- Warning** Activate MFA on your root account
- Checkmark** Create individual IAM users
- Checkmark** Use groups to assign permissions
- Warning** Apply an IAM password policy

Feature Spotlight

Introduction to AWS IAM

Additional Information

- IAM documentation
- Web Identity Federation
- Playground
- Policy Simulator
- Videos, IAM release history and additional resources

We need to send this special link to the Users that we created as the root user, they will use this link to log in as below

AWS Services Edit Matt Thomas Global Support

Welcome to Identity and Access Management

IAM users sign-in link: <https://515331423217.signin.aws.amazon.com/console> Customize | Copy Link

IAM Resources

Users: 1 Roles: 10 Groups: 1 Identity Providers: 0 Customer Managed Policies: 5

Security Status 3 out of 5 complete.

- ✓ Delete your root access keys**
- ⚠ Activate MFA on your root account**
- ✓ Create individual IAM users**
- ✓ Use groups to assign permissions**
- ⚠ Apply an IAM password policy**

Feature Spotlight

Introduction to AWS IAM

Additional Information

IAM documentation Web Identity Federation Playground Policy Simulator Videos, IAM release history and additional resources

AWS Services Edit BillyBob @ 5153-3142-3217 Oregon

Amazon Web Services

Compute	Developer Tools	Internet of Things
EC2 Virtual Servers in the Cloud	CodeCommit Store Code in Private Git Repositories	AWS IoT Connect Devices to the Cloud
EC2 Container Service Run and Manage Docker Containers	CodeDeploy Automate Code Deployments	Game Development
Elastic Beanstalk Run and Manage Web Apps	CodePipeline Release Software using Continuous Delivery	GameLift Deploy and Scale Session-based Multiplayer Games
Lambda Run Code in Response to Events		
Storage & Content Delivery	Management Tools	Mobile Services
S3 Scalable Storage in the Cloud	CloudWatch Monitor Resources and Applications	Mobile Hub Build, Test, and Monitor Mobile Apps
CloudFront Global Content Delivery Network	CloudFormation Create and Manage Resources with Templates	Cognito User Identity and App Data Synchronization
Elastic File System Fully Managed File System for EC2	CloudTrail Track User Activity and API Usage	Device Farm Test Android, iOS, and Web Apps on Real Devices in the Cloud
Glacier Archive Storage in the Cloud	Config Track Resource Inventory and Changes	Mobile Analytics Collect, View and Export App Analytics

Resource Groups Learn more

A resource group is a collection of resources that share one or more tags. Create a group for each project, application, or environment in your account.

Create a Group Tag Editor

Additional Resources

Getting Started Read our documentation or view our training to learn more about AWS.

AWS Console Mobile App View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.

We are now logged in as BillyBob on the right side of the screen.

AWS Services Edit BillyBob @ 5153-3142-3217 Oregon Support

Amazon Web Services

Compute	Developer Tools	Internet of Things
EC2 Virtual Servers in the Cloud	CodeCommit Store Code in Private Git Repositories	AWS IoT Connect Devices to the Cloud
EC2 Container Service Run and Manage Docker Containers	CodeDeploy Automate Code Deployments	Game Development
Elastic Beanstalk Run and Manage Web Apps	CodePipeline Release Software using Continuous Delivery	GameLift Deploy and Scale Session-based Multiplayer Games
Lambda Run Code in Response to Events		
Storage & Content Delivery	Management Tools	Mobile Services
S3 Scalable Storage in the Cloud	CloudWatch Monitor Resources and Applications	Mobile Hub Build, Test, and Monitor Mobile Apps
CloudFront Global Content Delivery Network	CloudFormation Create and Manage Resources with Templates	Cognito User Identity and App Data Synchronization
Elastic File System Fully Managed File System for EC2	CloudTrail Track User Activity and API Usage	Device Farm Test Android, iOS, and Web Apps on Real Devices in the Cloud
Glacier Archive Storage in the Cloud	Config Track Resource Inventory and Changes	Mobile Analytics Collect, View and Export App Analytics
Database	Services Catalog	SNS Push Notification Service
RDS Managed Relational Database Service	Service Catalog Create and Use Standardized Products	
DynamoDB Managed NoSQL Database	Trusted Advisor Optimize Performance and Security	
ElasticCache		
Security & Identity	Application Services	
	API Gateway Build, Deploy and Manage APIs	
	AppStream Low Latency Application Streaming	
	CloudSearch Managed Search Service	
	Elastic Transcoder Convert Video and Audio Files	

Resource Groups Learn more

A resource group is a collection of resources that share one or more tags. Create a group for each project, application, or environment in your account.

Create a Group Tag Editor

Additional Resources

Getting Started Read our documentation or view our training to learn more about AWS.

AWS Console Mobile App View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.

AWS Marketplace Find and buy software, launch with 1-Click and pay by the hour.

AWS re:Invent Announcements Explore the next generation of AWS cloud capabilities. See what's new

When you create a new User in AWS, they will be restricted from doing anything inside AWS until they are granted access by the root user in the form of IAM Roles.

The screenshot shows the AWS DynamoDB console at <https://us-west-2.console.aws.amazon.com/dynamodb/home?region=us-west-2#tables>. The user is BillyBob. The interface includes a sidebar with 'DynamoDB', 'Dashboard', 'Tables', and 'Reserved capacity'. A 'Create table' button is visible. A search bar at the top says 'Filter by table name'. Below it, a table header row shows 'Name', 'Status', 'Partition key', 'Sort key', 'Indexes', and 'Total read capac'. A red warning message '⚠ Not Authorized' is displayed below the table.

We can confirm that BillyBob is not able to even see the Comments table that we created in DynamoDB as the root user because BillyBob does not have any permissions for that yet.

The screenshot shows the AWS Reserved Capacity console at <https://us-west-2.console.aws.amazon.com/dynamodb/home?region=us-west-2#reserved-capacity>. The user is BillyBob. The interface includes a sidebar with 'DynamoDB', 'Dashboard', 'Tables', and 'Reserved capacity'. A 'Purchase reserved capacity' button is visible. A red warning message '⚠ Not Authorized' is displayed below the button.

The screenshot shows the AWS S3 console at <https://console.aws.amazon.com/s3/home?region=us-west-2#>. The user is BillyBob. The interface includes a sidebar with 'AWS', 'Services', and 'Edit'. A message states 'You don't have permissions to use the Amazon S3 Console. If you need assistance, contact your System Administrator.'.

The screenshot shows the AWS IAM Users console at <https://console.aws.amazon.com/iam/home?region=us-west-2#/users>. The user is Matt Thomas. The interface includes a sidebar with 'Dashboard', 'Search IAM', 'Details', 'Groups', 'Users' (which is selected), 'Roles', 'Policies', 'Identity Providers', 'Account Settings', and 'Credential Report'. A table lists users with columns: 'User Name', 'Groups', 'Password', 'Password Last Used', 'Access Keys', and 'Creation Time'. One user, 'BillyBob', is listed with 0 groups, a checked password status, and active access keys.

IAM > Users > BillyBob

Summary

User ARN: arn:aws:iam::515331423217:user/BillyBob
Has Password: Yes
Groups (for this user): 0
Path: /
Creation Time: 2016-07-07 08:10 UTC+0100

Groups **Permissions** **Security Credentials** **Access Advisor**

Access Keys

Use access keys to make secure REST or Query protocol requests to any AWS service API. For your protection, you should never share your secret keys with anyone. In addition, industry best practice recommends frequent key rotation. Learn more about Access Keys.

Create Access Key

Access Key ID	Created	Last Used	Last Used Service	Last Used Region	Status	Actions
AKIAJKEGUZHL4KCQTCDO	2016-07-07 08:10 UTC+0100	N/A	N/A	N/A	Active	Make Inactive Delete

Sign-In Credentials

User Name: BillyBob
Password: Yes
Last Used: 2016-07-07 08:14 UTC+0100

Manage Password

Multi-Factor Authentication Device: No
Manage MFA Device

Signing Certificates: None
Manage Signing Certificates

SSH keys for AWS CodeCommit

Use SSH public keys to authenticate to AWS CodeCommit repositories. Learn more about SSH keys.
No SSH public keys are associated with this user.

Upload SSH public key

IAM > Users > BillyBob

Summary

User ARN: arn:aws:iam::515331423217:user/BillyBob
Has Password: Yes
Groups (for this user): 0
Path: /
Creation Time: 2016-07-07 08:10 UTC+0100

Groups **Permissions** **Security Credentials** **Access Advisor**

Managed Policies

There are no managed policies attached to this user.

Attach Policy

Inline Policies

As the root user on the left, we can assign **permissions managed by Policies** to any of our users like BillyBob to allow them to be able to do things in our AWS account. A **Policy** is a JSON value/document defines an action that can be done by a User on some Resource, whether the user has **Allow** or **Deny** actions over some resources.

The screenshot shows the 'Attach Policy' page in the AWS IAM console. At the top, there's a search bar labeled 'Filter: Policy Type' and a 'Q' search icon. Below it, a table displays 206 results, with the first few rows visible:

	Policy Name	Attached Entities	Creation Time	Edited Time
<input type="checkbox"/>	Comments_PutItem	2	2016-05-21 11:38 UT...	2016-05-22 23:0...
<input type="checkbox"/>	AmazonDynamoDB...	1	2015-02-06 18:40 UT...	2015-11-11 23:3...
<input type="checkbox"/>	AmazonElasticMap...	1	2015-02-06 18:41 UT...	2015-05-13 22:2...
<input type="checkbox"/>	AmazonElasticMap...	1	2015-02-06 18:41 UT...	2016-02-10 22:4...
<input type="checkbox"/>	AWSMobileHub_Ser...	1	2015-12-10 22:04 UT...	2016-05-24 20:4...
<input type="checkbox"/>	get_comments_policy	1	2016-05-28 14:44 UT...	2016-06-06 08:4...
<input type="checkbox"/>	AdministratorAccess	0	2015-02-06 18:39 UT...	2015-02-06 18:3...
<input type="checkbox"/>	AmazonAPIGatewa...	0	2015-07-09 18:34 UT...	2015-07-09 18:3...
<input type="checkbox"/>	AmazonAPIGatewa...	0	2015-07-09 18:36 UT...	2015-07-09 18:3...
<input type="checkbox"/>	AmazonAPIGatewa...	0	2015-11-11 23:41 UT...	2015-11-11 23:4...
<input type="checkbox"/>	AmazonAppStream...	0	2015-02-06 18:40 UT...	2015-02-06 18:4...
<input type="checkbox"/>	AmazonAppStream...	0	2015-02-06 18:40 UT...	2015-02-06 18:4...

When you click the Attach Policy button, you see the screen above where AWS lists all the policies available including the ones you created. You can attach any of these policies to any user to grant them access to certain resources in your AWS account.

The screenshot shows the 'Attach Policy' page in the AWS IAM console with a filter applied. The search bar contains 'dynamo'. The table displays 5 results, with the second row selected:

	Policy Name	Attached Entities	Creation Time	Edited Time
<input type="checkbox"/>	AmazonDynamoDB...	1	2015-02-06 18:40 UT...	2015-11-11 23:3...
<input checked="" type="checkbox"/>	AmazonDynamoDB...	0	2015-02-06 18:40 UT...	2015-11-12 02:1...
<input type="checkbox"/>	AmazonDynamoDB...	0	2015-02-06 18:40 UT...	2015-11-12 02:1...
<input type="checkbox"/>	AWSLambdaDyna...	0	2015-04-09 16:09 UT...	2015-04-09 16:0...
<input type="checkbox"/>	AWSLambdaInvo...	0	2015-02-06 18:40 UT...	2015-02-06 18:4...

You can filter to see list of policies for DynamoDB that we can use and assign to our users.

The screenshot shows the 'Attach Policy' page in the AWS IAM console with a policy selected for attachment. The search bar contains 'dynamo'. The table displays 5 results, with the first row selected:

	Policy Name	Attached Entities	Creation Time	Edited Time
<input checked="" type="checkbox"/>	AmazonDynamoDB...	0	2015-02-06 18:40 UT...	2015-11-12 02:1...
<input type="checkbox"/>	AmazonDynamoDB...	0	2015-02-06 18:40 UT...	2015-11-12 02:1...
<input type="checkbox"/>	AWSLambdaDyna...	0	2015-04-09 16:09 UT...	2015-04-09 16:0...
<input type="checkbox"/>	AWSLambdaInvo...	0	2015-02-06 18:40 UT...	2015-02-06 18:4...
<input type="checkbox"/>	AmazonDynamoDB...	1	2015-02-06 18:40 UT...	2015-11-11 23:3...

We can select a Policy as above and click the Attach Policy button to give this User that specific access as above

The screenshot shows the AWS IAM User Details page for a user named 'BillyBob'. The left sidebar has 'Users' selected. The main content area shows the 'Summary' tab with details like User ARN, Has Password, Groups, Path, and Creation Time. Below this is a tab navigation bar with 'Groups', 'Permissions' (which is selected), 'Security Credentials', and 'Access Advisor'. Under 'Permissions', there are sections for 'Managed Policies' and 'Inline Policies'. The 'Managed Policies' section lists a single policy: 'AmazonDynamoDBFullAccess'.

BillyBob now has full DynamoDB access using this preconfigured AWS Policy that grants certain Allow or Deny actions

The screenshot shows a 'Show Policy' modal dialog. It displays the JSON code for the 'AmazonDynamoDBFullAccess' policy. The policy grants full access to all DynamoDB resources. The JSON code is as follows:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "dynamodb:*",
                "cloudwatch:DeleteAlarms",
                "cloudwatch:DescribeAlarmHistory",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:DescribeAlarmsForMetric",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "cloudwatch:PutMetricAlarm",
                "datapipeline:ActivatePipeline",
                "datapipeline>CreatePipeline",
                "datapipeline:DeletePipeline",
                "datapipeline:DescribeObjects",
                "datapipeline:DescribePipelines",
                "datapipeline:GetPipelineDefinition",
                "datapipeline>ListPipelines",
                "datapipeline:PutPipelineDefinition",
                "datapipeline:QueryObjects",
                "iam>ListRoles",
                "sns>CreateTopic",
                "sns>DeleteTopic",
                "sns>ListSubscriptions",
                "sns>ListSubscriptionsByTopic",
                "sns>ListTopics",
                "sns:Subscribe",
                "sns:Unsubscribe",
                "sns:SetTopicAttributes",
                "lambda>CreateFunction",
                "lambda>ListFunctions",
                "lambda>ListEventSourceMappings",
                "lambda>CreateEventSourceMapping",
                "lambda>DeleteEventSourceMapping",
                "lambda:GetFunctionConfiguration",
                "lambda>DeleteFunction"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

These are all the things that this Policy grants the user. You can copy this policy and edit to grants fine-grained access to only certain resources like only to a specific table(s) instead of all DynamoDB tables using the * wildcard as above.

The screenshot shows the AWS IAM User Details page for a user named 'BillyBob'. The left sidebar includes links for Dashboard, Details, Groups, Users (which is selected), Roles, Policies, Identity Providers, Account Settings, Credential Report, and Encryption Keys. The main content area displays the user's ARN (arn:aws:iam::515331423217:user/BillyBob), password status (Yes), groups (0), path (/), and creation time (2016-07-07 08:10 UTC+0100). Below this, tabs for Groups, Permissions (selected), Security Credentials, and Access Advisor are shown. The 'Managed Policies' section lists a single policy: 'AmazonDynamoDBFullAccess' with actions for Show Policy, Detach Policy, and Simulate Policy.

The screenshot shows the AWS DynamoDB Dashboard for the US West (Oregon) region. The left sidebar has links for Dashboard, Tables, and Reserved capacity. The main content area features a 'Create table' button, a 'Recent alerts' section (empty), and a 'Total capacity for US West (Oregon)' summary table. The table shows Provisioned read capacity at 5, Provisioned write capacity at 5, Reserved read capacity at 0, and Reserved write capacity at 0. To the right, there are sections for 'Best practices' (Amazon DynamoDB January 2016), 'What's new' (list of enhancements), 'Related services' (Amazon ElastiCache), and 'Additional resources' (links to guide, hands-on lab, FAQ, release notes, developer guide, forums, and report an issue).

The user called BillyBob can now see the DynamoDB service on his screen and is able to do things.

← → C https://us-west-2.console.aws.amazon.com/dynamodb/home?region=us-west-2#create-table: star

AWS Services Edit BillyBob @ 5153-3142-3217 Oregon Support

Create DynamoDB table

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name*

Primary key* Partition key
 String info

Add sort key

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

Cancel Create

BillyBob can even create tables and so on

← → C https://us-west-2.console.aws.amazon.com/dynamodb/home?region=us-west-2#create-table: star

AWS Services Edit BillyBob @ 5153-3142-3217 Oregon Support

Create DynamoDB table

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* BillyBobTable

Primary key* Partition key
 id String info

Add sort key

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

Cancel Create

IAM > Users > BillyBob

Summary

User ARN: arn:aws:iam::515331423217:user/BillyBob
Has Password: Yes
Groups for this user:
Path: /
Creation Time: 2016-07-07 08:10 UTC+0100

Managed Policies

The following managed policies are attached to this user. You can attach up to 10 managed policies.

Policy Name	Actions
AmazonDynamoDBFullAccess	Show Policy Detach Policy Simulate Policy

DynamoDB

Create table

BillyBobTable

OregonTable

Let us now start playing around with the Policies attached to the User BillyBob to show the concepts of Roles, Groups, etc. but let us create another User below

Welcome to Identity and Access Management

IAM users sign-in link: <https://515331423217.signin.aws.amazon.com/console>

IAM Resources

- Users: 1
- Groups: 1
- Roles: 10
- Identity Providers: 0

Customer Managed Policies: 5

Security Status

3 out of 5 complete.

- Delete your root access keys
- Activate MFA on your root account
- Create individual IAM users
- Use groups to assign permissions
- Apply an IAM password policy

Feature Spotlight

Introduction to AWS IAM

Dashboard

Create New Users

User Actions

User Name	Groups	Password	Password Last Used	Access Keys	Creation Time
BillyBob	0	✓	2016-07-07 08:14 UTC+0100	1 active	2016-07-07 08:10 ...

Create User

Enter User Names:

- Angelina
-
-
-
-

Maximum 64 characters each

Generate an access key for each user

Users need access keys to make secure REST or Query protocol requests to AWS service APIs.
For users who need access to the AWS Management Console, create a password in the Users panel after completing this wizard.

Create a new User called Angelina

AWS Services Edit Matt Thomas Global Support

Create User

Your 1 User(s) have been created successfully.
This is the last time these User security credentials will be available for download.
You can manage and recreate these credentials any time.
► Show User Security Credentials

AWS Services Edit Matt Thomas Global Support

Create User

You haven't downloaded the User security credentials. This is the last time these credentials will be available for download.
Are you sure you want to close the window?

AWS Services Edit Matt Thomas Global Support

Dashboard Search IAM

Details Groups **Users** Roles Policies Identity Providers Account Settings Credential Report Encryption Keys

Create New Users User Actions

Showing 2 results

User Name	Groups	Password	Password Last Used	Access Keys	Creation Time
Angelina	0	N/A	2016-07-07 08:14 UTC+0100	1 active	2016-07-07 08:27 ...
BillyBob	0	✓	2016-07-07 08:14 UTC+0100	1 active	2016-07-07 08:10 ...

AWS Services Edit Matt Thomas Global Support

Dashboard Search IAM

Details Groups **Users** Roles Policies Identity Providers Account Settings Credential Report Encryption Keys

IAM > Users > Angelina

Summary

User ARN: arn:aws:iam::515331423217:user/Angelina
Has Password: No
Groups (for this user): 0
Path: /
Creation Time: 2016-07-07 08:27 UTC+0100

Groups Permissions Security Credentials Access Advisor

Managed Policies

There are no managed policies attached to this user.

Attach Policy

Inline Policies

AWS Services Edit Matt Thomas Global Support

Dashboard Search IAM

Details Groups Users Roles Policies Identity Providers Account Settings Credential Report Encryption Keys

IAM > Users > Angelina

Summary

User ARN: arn:aws:iam::515331423217:user/Angelina
Has Password: No
Groups (for this user): 0
Path: /
Creation Time: 2016-07-07 08:27 UTC+0100

Groups Permissions Security Credentials Access Advisor

Access Keys

Use access keys to make secure REST or Query protocol requests to any AWS service API. For your protection, you should never share your secret keys with anyone. In addition, industry best practice recommends frequent key rotation. Learn more about Access Keys

Create Access Key

Access Key ID	Created	Last Used	Last Used Service	Last Used Region	Status	Actions
AKIAISIV2HECWVF6FW5A	2016-07-07 08:27 UTC+0100	N/A	N/A	N/A	Active	Make Inactive Delete

Sign-In Credentials

User Name: Angelina Manage Password
Password: No
Last Used: N/A

Multi-Factor Authentication Device Manage MFA Device

Signing Certificates: None Manage Signing Certificates

SSH keys for AWS CodeCommit

Use SSH public keys to authenticate to AWS CodeCommit repositories. Learn more about SSH keys.
No SSH public keys are associated with this user.

Upload SSH public key

AWS Services Edit Matt Thomas Global Support

Manage Password

Users who will be using the AWS Management Console require a password. Select from the options below to manage the password for user Angelina.

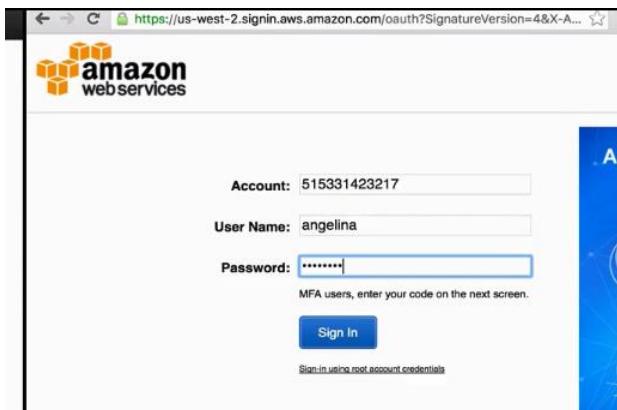
Assign an auto-generated password
 Assign a custom password

Password:
Confirm Password:

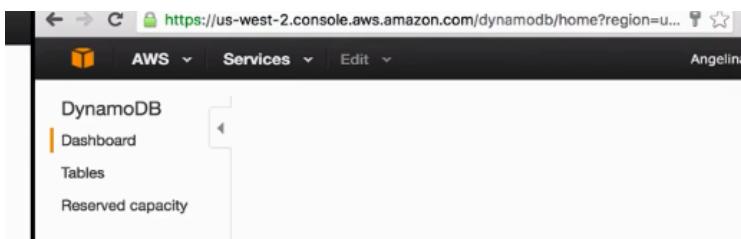
Require user to create a new password at next sign-in

The screenshot shows the AWS IAM User details page for a user named 'Angelina'. The left sidebar lists navigation options like Dashboard, Details, Groups, Users (which is selected), Roles, Policies, Identity Providers, Account Settings, Credential Report, and Encryption Keys. The main content area displays the user's ARN (arn:aws:iam::515331423217:user/Angelina), password status (Yes), groups (0), path (/), and creation time (2016-07-07 08:27 UTC+0100). Below this, tabs for Groups, Permissions, Security Credentials (selected), and Access Advisor are visible. The 'Security Credentials' tab contains sections for Access Keys, Sign-In Credentials, and SSH keys for AWS CodeCommit. The Access Keys section shows one key: AKIAISIV2HECWVF6FW5A, created on 2016-07-07 08:27 UTC+0100, with 'Status' set to 'Active'. The Sign-In Credentials section shows the User Name as Angelina, Password as Yes, and Last Used as Never. The SSH keys section indicates no keys are associated with this user.

The User Angelina now has a password set



We then log into another screen as the User Angelina



Note that we have not yet assigned any policy for accessing resources to Angelina and she wouldn't be able to do work. Assuming we are building a database team and Angelina and BillyBob are the current team members, we will have to attach the same policy that granted BillyBob access to the DynamoDB service under our AWS account to Angelina and all other future team members. We can instead create a new Group for the database team, attach that policy to the group, and just add team member Users to that group for access.

The screenshot shows the AWS IAM Groups page. The left sidebar has a 'Groups' section selected. The main area has a 'Create New Group' button and a 'Group Actions' dropdown. A search bar and filter options ('Group Name', 'Users', 'Inline Policy', 'Creation Time') are present. The message 'Showing 0 results' indicates no groups are currently listed.

A **Group** is a logical group of users to which we can assign certain permissions and properties.

The screenshot shows the 'Create New Group Wizard' at Step 1: Group Name. It asks for a group name ('MyDBTeam') and provides examples like 'Developers' or 'ProjectAlpha'. A note says group names can be edited any time. Step 2: Attach Policy and Step 3: Review are also shown.

We create the new Group as above

The screenshot shows the 'Attach Policy' step of the wizard. It lists policies available for attachment, including 'AmazonDynamoDBFullAccess' (selected), 'AmazonDynamoDBReadOnly', 'AmazonElasticMapReduce*', 'AmazonElasticMapReduce...', 'AWSMobileHub_ServiceU...', 'Comments_PutItem', 'get_comments_policy', and 'AdministratorAccess'. A note says each group can have up to 10 policies attached.

We can now attach all the needed Policies to this new group.

The screenshot shows the 'Review' step of the wizard. It summarizes the group creation information: Group Name is 'MyDBTeam' and Policies are 'arn:aws:iam::aws:policy/AmazonDynamoDBFullAccess'. Buttons for 'Create Group' and 'Edit Group Name' are present.

The screenshot shows the AWS IAM Groups page. On the left, there's a sidebar with options like Dashboard, Details, Groups (which is selected), Users, Roles, Policies, Identity Providers, and Account Settings. The main area has a search bar and a table with columns: Group Name, Users, Inline Policy, and Creation Time. One result is listed: MyDBTeam, 0 users, no inline policy, created on 2016-07-07 08:30 UTC+0100.

This screenshot shows the details for the 'MyDBTeam' group. It includes a summary section with Group ARN, Users (0), Path (/), and Creation Time (2016-07-07 08:30 UTC+0100). Below this are tabs for Users, Permissions (selected), and Access Advisor. Under the Permissions tab, there's a 'Managed Policies' section with a table showing one policy: 'AmazonDynamoDBFullAccess'. There's also an 'Inline Policies' section which is currently empty.

We currently have a single policy attached to this group at the moment for full DynamoDB access

This screenshot shows the same 'MyDBTeam' group details as before, but now the 'Users' tab is selected. It displays a message: 'This group does not contain any users.' and a blue 'Add Users to Group' button.

We can also start adding Users to the Group,

This screenshot shows the 'Add Users to Group' dialog. It lists two users: Angelina and BillyBob. Both have checkboxes next to their names, and the first one is checked. The table columns are User Name, Groups, Password, Password Last Used, Access Keys, and Creation Time.

User Name	Groups	Password	Password Last Used	Access Keys	Creation Time
Angelina	0	✓	2016-07-07 08:28 UTC+0100	1 active	2016-07-07 08:27...
BillyBob	0	✓	2016-07-07 08:14 UTC+0100	1 active	2016-07-07 08:10...

Screenshot of the AWS IAM Groups page for the 'MyDBTeam' group.

Summary

Group ARN: arn:aws:iam::515331423217:group/MyDBTeam
Users (in this group): 2
Path: /
Creation Time: 2016-07-07 08:30 UTC+0100

Users **Permissions** **Access Advisor**

This view shows all users in this group: **2 Users**

User Actions

User	Actions
BillyBob	Remove User from Group
Angelina	Remove User from Group

Remove Users from Group **Add Users to Group**

Screenshot of the AWS IAM Groups page for the 'MyDBTeam' group, showing the 'Permissions' tab.

Summary

Group ARN: arn:aws:iam::515331423217:group/MyDBTeam
Users (in this group): 2
Path: /
Creation Time: 2016-07-07 08:30 UTC+0100

Users **Permissions** **Access Advisor**

Managed Policies

The following managed policies are attached to this group. You can attach up to 10 managed policies.

Attach Policy

Policy Name	Actions
AmazonDynamoDBFullAccess	Show Policy Detach Policy Simulate Policy

Inline Policies

Screenshot of the AWS DynamoDB service dashboard.

DynamoDB

Recent alerts
No CloudWatch alarms have been triggered.
[View all in CloudWatch](#)

Total capacity for US West (Oregon)

Provisioned read capacity	10
Provisioned write capacity	10
Reserved read capacity	0
Reserved write capacity	0

Service health

Current Status
Loading ...

Best practices
[Amazon DynamoDB Best Practices](#)

What's new

- Enhanced metrics
- Titan graph database integration
- Elasticsearch integration

Related services

- Amazon ElastiCache

Additional resources

- Getting started guide
- Getting started hands-on lab
- FAQ
- Release notes
- Developer guide
- Forums
- Report an issue

Angelina now has access to the DynamoDB service and can start using it as below

<https://us-west-2.console.aws.amazon.com/dynamodb/home?region=us-...>

AWS Services Edit Angelina

Create DynamoDB table Tutorial ?

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* AngelinaTable i

Primary key* Partition key i

String

Add sort key

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".

<https://us-west-2.console.aws.amazon.com/dynamodb/home?region=us-...>

AWS Services Edit Angelina

DynamoDB Dashboard Tables Reserved capacity

Create table Actions ▾

Filter by table name X

Name
AngelinaTable
BillyBobTable
OregonTable

AngelinaTable Overview Item

Table is being created

Recent alerts

No CloudWatch alarms have been triggered for this table.

Stream details

Table details

<https://console.aws.amazon.com/iam/home?region=us-...>

AWS Services Edit Matt Thomas Global Support

Dashboard Search IAM

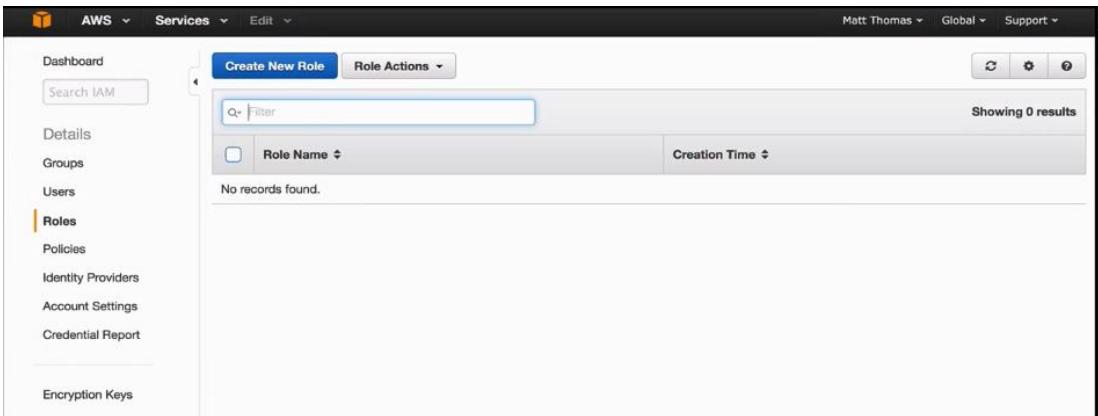
Details Groups Users Roles Policies Identity Providers Account Settings Credential Report Encryption Keys

Create New Role Role Actions

Showing 10 results

Role Name	Creation Time
Comment_role	2016-05-21 11:42 UTC+0100
dynamodbreadonly	2016-07-05 10:08 UTC+0100
EMR_DefaultRole	2016-06-21 16:18 UTC+0100
EMR_EC2_DefaultRole	2016-06-21 16:18 UTC+0100
get_comments_role	2016-05-28 14:46 UTC+0100
ios_lambdaexecutionrole_MOBILEHUB_643227571	2016-06-10 11:45 UTC+0100
ios_unauth_MOBILEHUB_643227571	2016-06-10 11:43 UTC+0100
MobileHub_Service_Role	2016-06-10 10:07 UTC+0100
mobile_unauth_MOBILEHUB_1815289795	2016-06-10 10:07 UTC+0100
mobile_unauth_MOBILEHUB_325533447	2016-06-10 10:38 UTC+0100

IAM Roles are slightly different from Groups, they are also used to assign permissions to Users. **A Role is an entity that you define in AWS IAM service which allows certain User to temporarily adopt/assume that Role, perform actions that are defined for that Role, then release the Role once finished.**



Let us create a new Role that will allow a User temporary permission to access and use the DynamoDB service.



This is a **Role** that different users may or may not be allowed to assume during company operations. We are going to grant anybody operating under the **DatabaseMasterGuru Role** above some very awesome permissions.

There are 3 different types of Roles that we can create shown above, AWS Service Roles, Cross-Account Access Roles, and Identity Provider Access Roles. When you have an EC2 hosted app instance running some application and you want to grant this EC2 hosted instance permissions to call other services in your AWS account like DynamoDB, you can give this instance a Role that enables it to do this. You can create a User for this EC2 app and somehow embed that user's public and private key in your application logic in the EC2 instance, that is not the most efficient way to do this.

The recommended way is to define a Role that can do certain things in DynamoDB and grant the EC2 instance the ability to assume that Role to access DynamoDB.

Welcome to Identity and Access Management

We can also see the URL that people can use to log in when they want to assume this DatabaseMasterGuru Role. We then click the Create button to create the Role

The screenshot shows the AWS IAM Roles page. On the left, there's a sidebar with options like Dashboard, Details, Groups, Users, Roles (which is selected), Policies, Identity Providers, Account Settings, and Credential Report. The main area has tabs for 'Create New Role' and 'Role Actions'. A search bar at the top says 'Search IAM'. Below it, a table lists one role: 'DatabaseMasterGuru' created on '2016-07-07 08:41 UTC+0100'. There are icons for edit, delete, and details.

This screenshot shows the 'Permissions' tab for the 'DatabaseMasterGuru' role. It displays managed policies attached to the role. One policy, 'AmazonDynamoDBFullAccess', is listed with options to 'Show Policy', 'Detach Policy', or 'Simulate Policy'. There's also a section for inline policies, which is currently empty.

This screenshot shows the 'Trust Relationships' tab for the 'DatabaseMasterGuru' role. It indicates that the role can be assumed by entities from account 515331423217. A button labeled 'Edit Trust Relationship' is present. The 'Conditions' section below states that no conditions are associated with this role.

We can click the Edit Trust Relationship button to see the Policy Document as below

This screenshot shows the 'Show Policy' page for the 'DatabaseMasterGuru' role. It displays the following JSON policy document:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::515331423217:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Here is the Policy Document is the trust relationship that defines who from outside our account can assume this Role. Principal is an identity with the specified account shown in the document

Role ARN: arn:aws:iam::515331423217:role/DatabaseMasterGuru

Instance Profile ARN(s): /

Path: /

Creation Time: 2016-07-07 08:41 UTC+0100

Give this link to users who can switch roles in the console: <https://signin.aws.amazon.com/switchrole?account=515331423217&roleName=DatabaseMasterGuru>

Permissions | **Trust Relationships** | **Access Advisor**

You can view the trusted entities that can assume the role and the access conditions for the role. Show policy document

Trusted Entities

The following trusted entities can assume this role.

Conditions

The following conditions define how and when trusted entities can assume the role.

Trusted Entities

The account 515331423217

The **515331423217** account on its end will also have to set up/manage Users that it is allowing to assume this Role when they want to do work

Role ARN: arn:aws:iam::515331423217:role/DatabaseMasterGuru

Instance Profile ARN(s): /

Path: /

Creation Time: 2016-07-07 08:41 UTC+0100

Give this link to users who can switch roles in the console: <https://signin.aws.amazon.com/switchrole?account=515331423217&roleName=DatabaseMasterGuru>

Permissions | **Trust Relationships** | **Access Advisor**

Managed Policies

The following managed policies are attached to this role. You can attach up to 10 managed policies.

Attach Policy

Policy Name	Actions
AmazonDynamoDBFullAccess	Show Policy Detach Policy Simulate Policy

Inline Policies

We can now copy think link and send to the Users to switch Roles.

Role ARN: arn:aws:iam::515331423217:role/DatabaseMasterGuru

Instance Profile ARN(s): /

Path: /

Creation Time: 2016-07-07 08:41 UTC+0100

Give this link to users who can switch roles in the console: <https://signin.aws.amazon.com/switchrole?account=515331423217&roleName=DatabaseMasterGuru>

Permissions | **Trust Relationships** | **Access Advisor**

Managed Policies

The following managed policies are attached to this role. You can attach up to 10 managed policies.

Attach Policy

Policy Name	Actions
AmazonDynamoDBFullAccess	Show Policy Detach Policy Simulate Policy

Inline Policies

Angelina @ 5153-3142-3217 | Oregon | Support

IAM User: Angelina
Account: 5153-3142-3217

Groups Learn more

group is a collection of users that share one or more tags for each project, team, or environment in your account.

My Account

Billing & Cost Management

Security Credentials

Switch Role

Sign Out

AWS Console Mobile App View your resources on the go with the AWS mobile app, available from Amazon Appstore, Google Play, and iTunes.

AWS Marketplace Find and buy software, launch with 1-Click and pay by the hour.

AWS re:Invent Announcements Explore the next generation of AWS services.

While logged in as Angelina on the right, she can switch Roles as above

https://signin.aws.amazon.com/switchrole?src=nav&redirect_uri=https%3A%2F%2Fus-west-2.console.aws.amazon.com...

The page shows the 'Switch role' interface. It includes a 'Switch Role' button, a section titled 'Get started in 3 simple steps' with three icons: 'Create role', 'Role access', and 'Switch roles', and a note about administrator permissions.

<https://signin.aws.amazon.com/switchrole>

The left panel shows the AWS IAM Roles page for 'DatabaseMasterGuru'. The right panel shows the 'Switch Role' dialog with fields for Account, Role, Display Name, and Color, along with a note about account settings and a 'Switch Role' button.

We then specify which account to use. Recall that this is a 2-way process. We need to administer Angelina explicitly or implicitly using a group or User to allow her be able to switch to this Role

IAM Management Console Granting a User Permissions to Switch Roles - AWS Identity and Access Management

AWS Services Edit Matt Thomas Global Support

Dashboard IAM > Users > BillyBob

Search IAM

Details Groups **Users** Roles Policies Identity Providers Account Settings Credential Report

Encryption Keys

User ARN: arn:aws:iam::515331423217:user/BillyBob
Has Password: Yes
Groups (for this user): 1
Path: /
Creation Time: 2016-07-07 08:10 UTC+0100

Groups Permissions **Security Credentials** Access Advisor

Managed Policies

The following managed policies are attached to this user. You can attach up to 10 managed policies.

Attach Policy

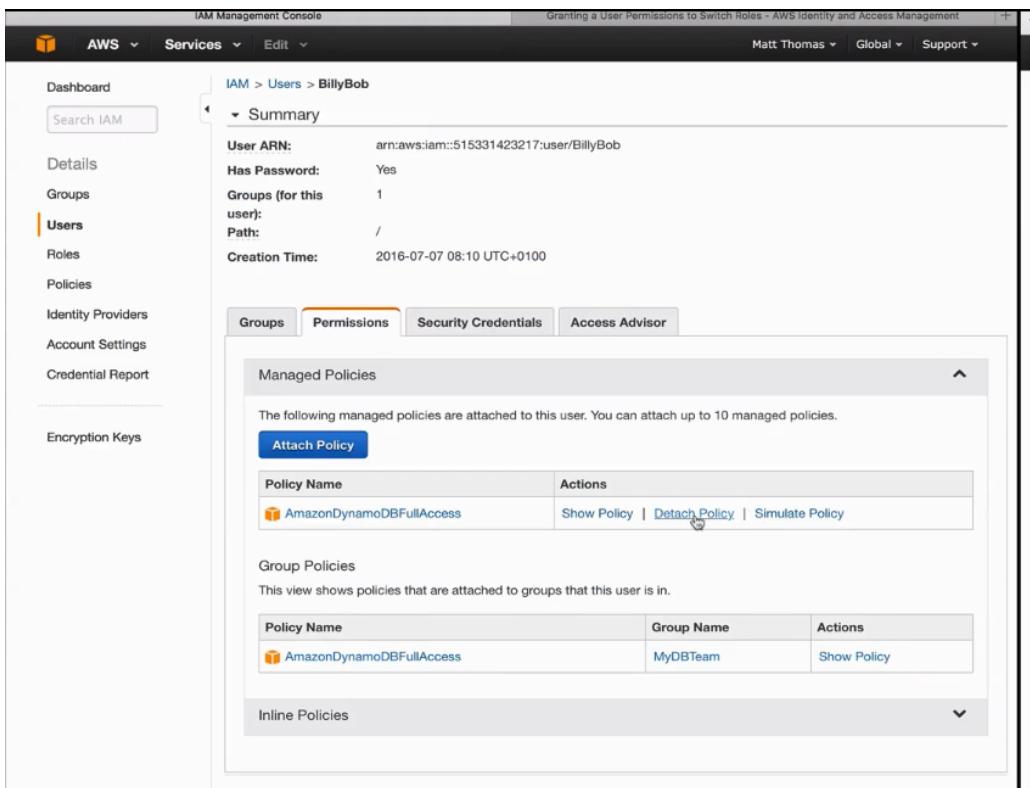
Policy Name	Actions
AmazonDynamoDBFullAccess	Show Policy Detach Policy Simulate Policy

Group Policies

This view shows policies that are attached to groups that this user is in.

Policy Name	Group Name	Actions
AmazonDynamoDBFullAccess	MyDBTeam	Show Policy

Inline Policies



Let us remove BillyBob's DynamoDB access

IAM Management Console Granting a User Permissions to Switch Roles - AWS Identity and Access Management

AWS Services Edit Matt Thomas Global Support

Dashboard IAM > Users > BillyBob

Search IAM

Details Groups **Users** Roles Policies Identity Providers Account Settings Credential Report

Encryption Keys

User ARN: arn:aws:iam::515331423217:user/BillyBob
Has Password: Yes
Groups (for this user): 1
Path: /
Creation Time: 2016-07-07 08:10 UTC+0100

Groups Permissions **Security Credentials** Access Advisor

Managed Policies

The following managed policies are attached to this user. You can attach up to 10 managed policies.

Attach Policy

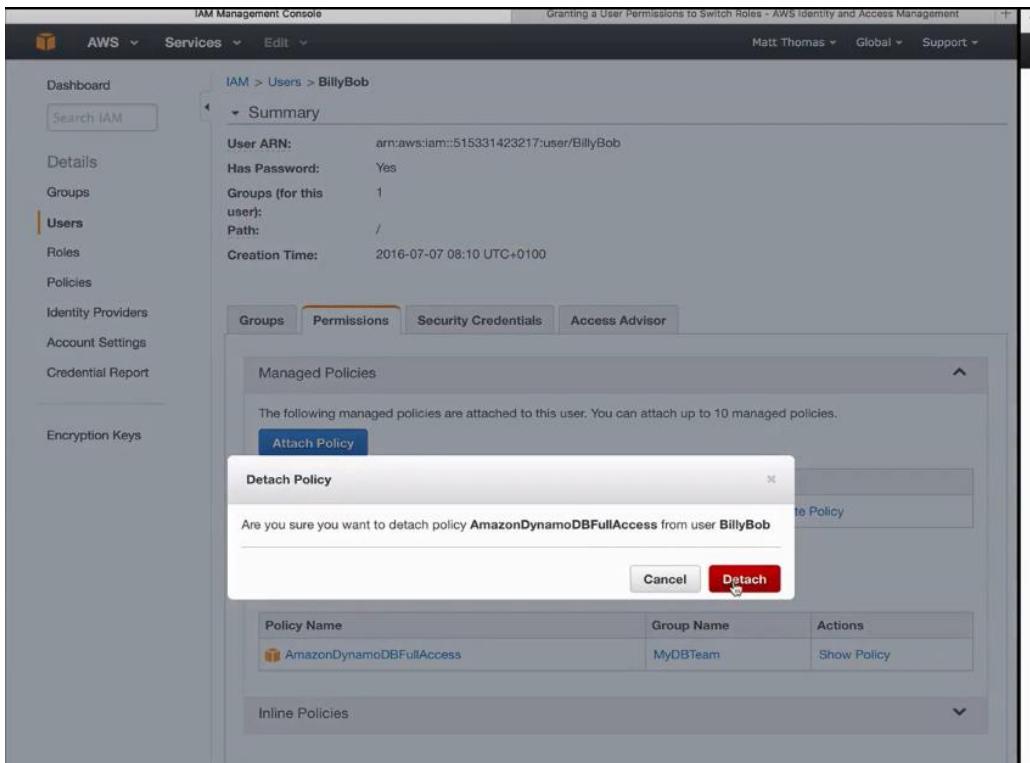
Detach Policy

Are you sure you want to detach policy AmazonDynamoDBFullAccess from user BillyBob?

Cancel **Detach**

Policy Name	Group Name	Actions
AmazonDynamoDBFullAccess	MyDBTeam	Show Policy

Inline Policies



IAM Management Console Granting a User Permissions to Switch Roles - AWS Identity and Access Management

AWS Services Edit Matt Thomas Global Support

Dashboard IAM > Users > BillyBob

Search IAM

Details Groups **Users** Roles Policies Identity Providers Account Settings Credential Report Encryption Keys

User ARN: arn:aws:iam::515331423217:user/BillyBob
Has Password: Yes
Groups (for this user): 1
Path: /
Creation Time: 2016-07-07 08:10 UTC+0100

Groups Permissions **Security Credentials** Access Advisor

Managed Policies

There are no managed policies attached to this user.

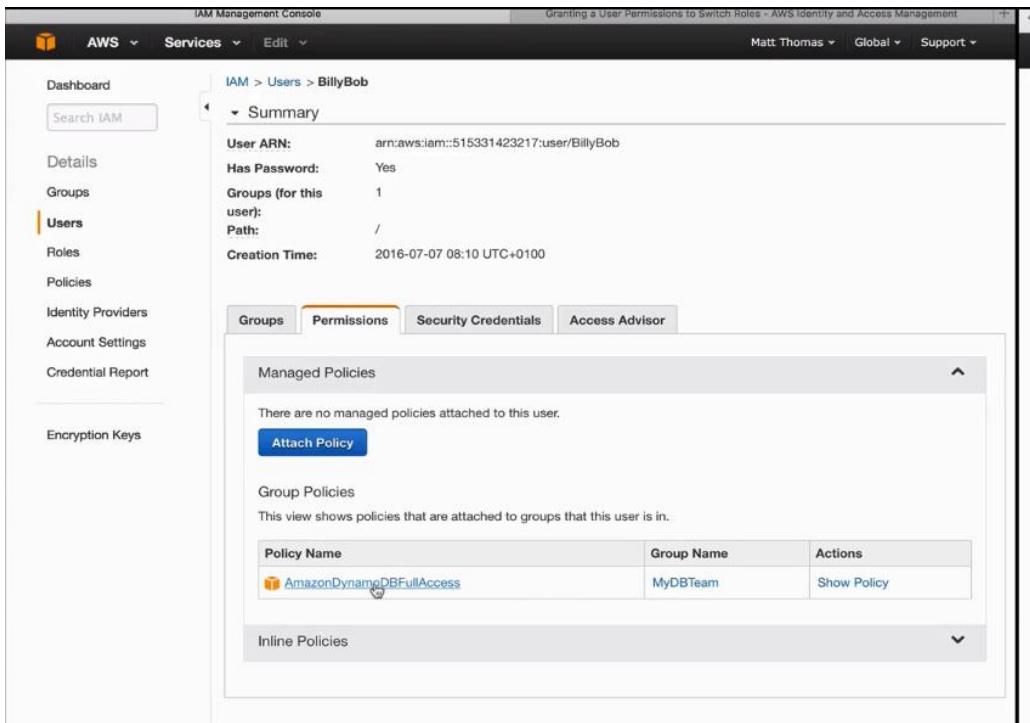
Attach Policy

Group Policies

This view shows policies that are attached to groups that this user is in.

Policy Name	Group Name	Actions
AmazonDynamoDBFullAccess	MyDBTeam	Show Policy

Inline Policies



IAM Management Console Granting a User Permissions to Switch Roles - AWS Identity and Access Management

AWS Services Edit Matt Thomas Global Support

Dashboard IAM > Groups > MyDBTeam

Search IAM

Details **Groups** Users Roles Policies Identity Providers Account Settings Credential Report Encryption Keys

Group ARN: arn:aws:iam::515331423217:group/MyDBTeam
Users (in this group): 2
Path: /
Creation Time: 2016-07-07 08:30 UTC+0100

Users Permissions Access Advisor

Managed Policies

The following managed policies are attached to this group. You can attach up to 10 managed policies.

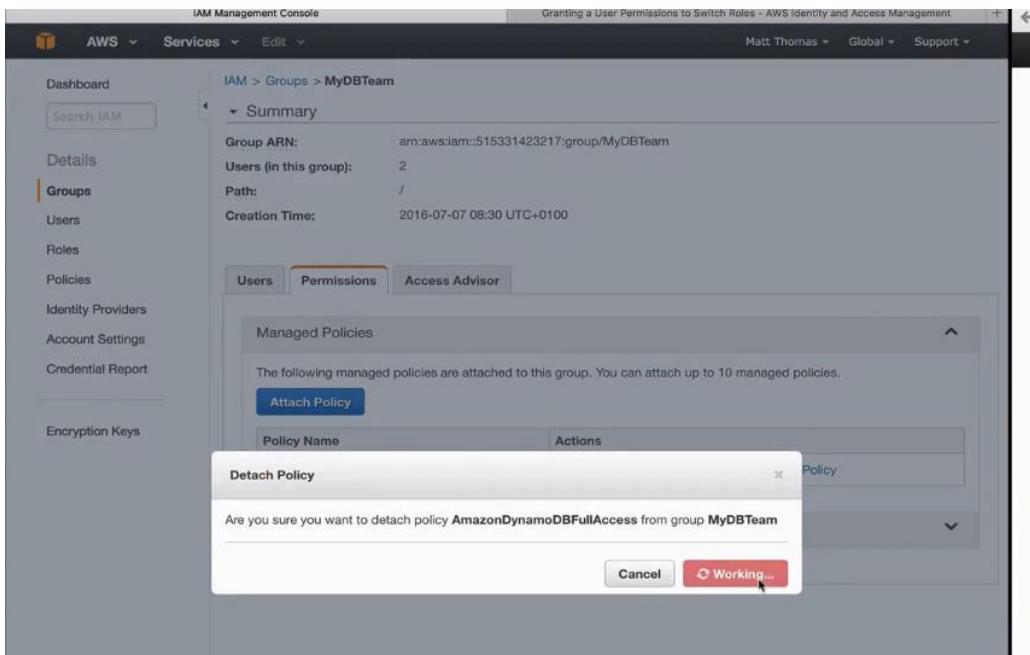
Attach Policy

Policy Name	Actions
AmazonDynamoDBFullAccess	Policy

Detach Policy

Are you sure you want to detach policy **AmazonDynamoDBFullAccess** from group **MyDBTeam**

Cancel Working...



Now this removes the group members from accessing DynamoDB too

IAM Management Console Granting a User Permissions to Switch Roles - AWS Identity and Access Management

AWS Services Edit Matt Thomas Global Support

Dashboard Search IAM Details Groups Users Roles Policies Identity Providers Account Settings Credential Report Encryption Keys

IAM > Groups > MyDBTeam

Summary

Group ARN: arn:aws:iam::515331423217:group/MyDBTeam
Users (in this group): 2
Path: /
Creation Time: 2016-07-07 08:30 UTC+0100

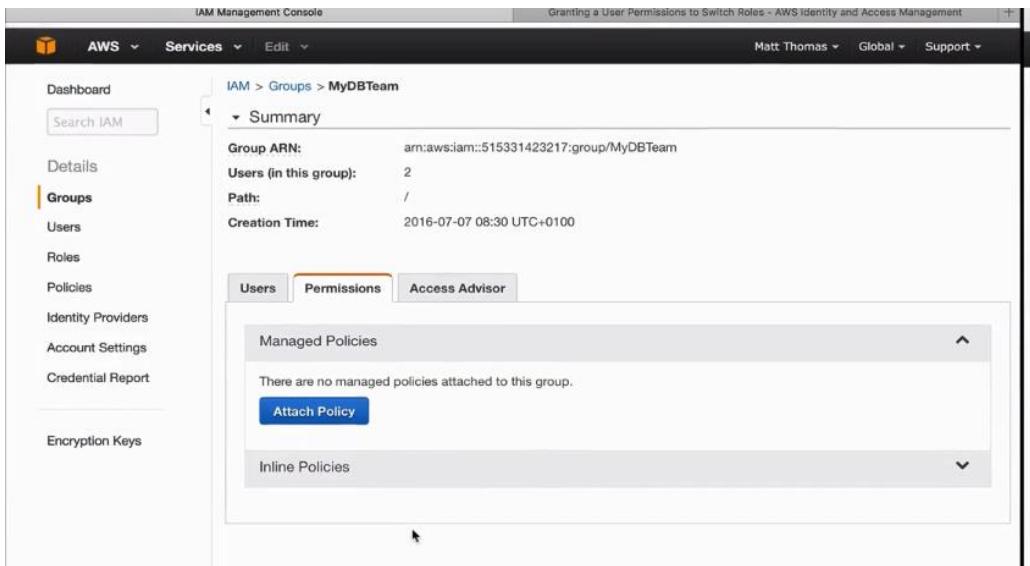
Users Permissions Access Advisor

Managed Policies

There are no managed policies attached to this group.

Attach Policy

Inline Policies



IAM Management Console Granting a User Permissions to Switch Roles - AWS Identity and Access Management

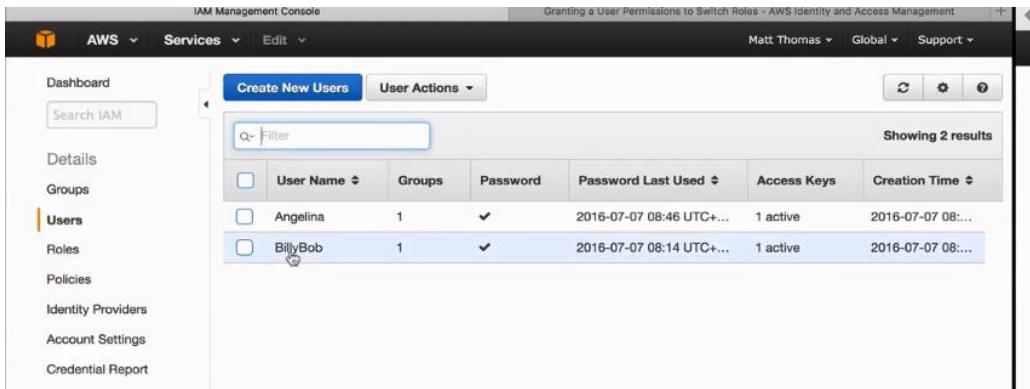
AWS Services Edit Matt Thomas Global Support

Dashboard Search IAM Details Groups Users Roles Policies Identity Providers Account Settings Credential Report Encryption Keys

Create New Users User Actions

Showing 2 results

User Name	Groups	Password	Password Last Used	Access Keys	Creation Time
Angelina	1	✓	2016-07-07 08:46 UTC+...	1 active	2016-07-07 08:...
BillyBob	1	✓	2016-07-07 08:14 UTC+...	1 active	2016-07-07 08:...



Now BillyBob and Angelina both have no access to DynamoDB anymore

IAM Management Console Granting a User Permissions to Switch Roles - AWS Identity and Access Management

AWS Services Edit Matt Thomas Global Support

Dashboard Search IAM Details Groups Users Roles Policies Identity Providers Account Settings Credential Report Encryption Keys

IAM > Users > Angelina

Summary

User ARN: arn:aws:iam::515331423217:user/Angelina
Has Password: Yes
Groups (for this user): 1
Path: /
Creation Time: 2016-07-07 08:27 UTC+0100

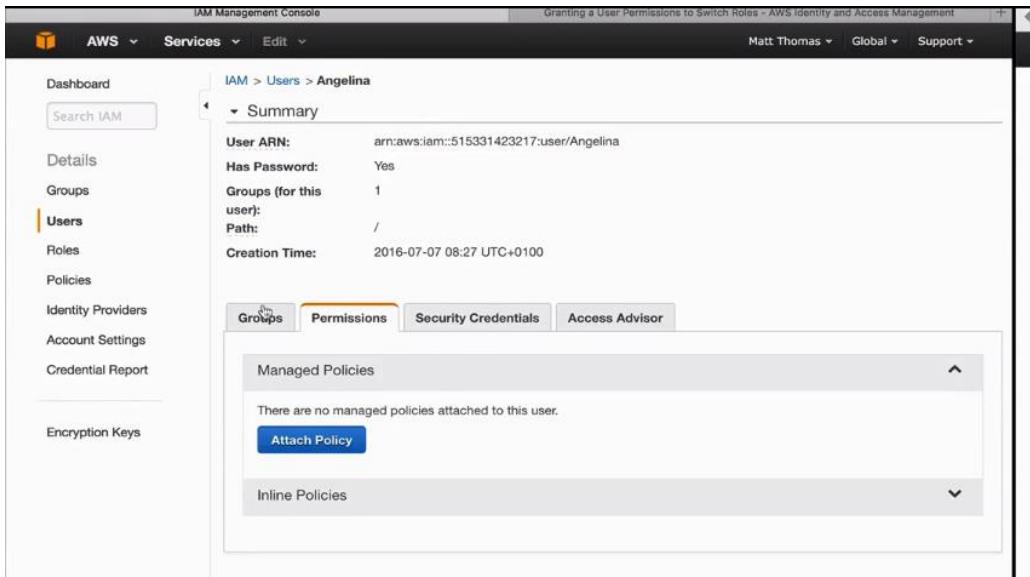
Groups Permissions Security Credentials Access Advisor

Managed Policies

There are no managed policies attached to this user.

Attach Policy

Inline Policies



Dashboard

IAM > Users > Angelina

Summary

User ARN: arn:aws:iam::515331423217:user/Angelina

Has Password: Yes

Groups (for this user): 1

Path: /

Creation Time: 2016-07-07 08:27 UTC+0100

Groups Permissions Security Credentials Access Advisor

This view shows all groups the User belongs to: 1 Group

Add User to Groups

Group	Actions
MyDBTeam	Remove from Group

Angelina is a member of the MyDBTeam but the group also does not have access to DynamoDB

Dashboard

IAM > Users > Angelina

Summary

User ARN: arn:aws:iam::515331423217:user/Angelina

Has Password: Yes

Groups (for this user): 1

Path: /

Creation Time: 2016-07-07 08:27 UTC+0100

Groups Permissions Security Credentials Access Advisor

Managed Policies

There are no managed policies attached to this user.

Attach Policy

Inline Policies

There are no inline policies to show. To create one, click here.

We are now going to grant Angelina the permission to assume the DatabaseMasterGuru Role. **Inline Policies** are policies that you create which are directly attached to the entity (like user/Angelina) that you are currently working with. Managed Policies on the other can be attached to any User, Group, or Role we want

Manage User Permissions

IAM Management Console

AWS Services Edit Matt Thomas Global Support

Set Permissions

Select a policy template, generate a policy, or create a custom policy. A policy is a document that formally states one or more permissions. You can edit the policy on the following screen, or at a later time using the user, group, or role detail pages.

Policy Generator

Use the policy generator to create your own set of permissions.

Custom Policy

Select

IAM Management Console Granting a User Permissions to Switch Roles - AWS Identity and Access Management

AWS Services Edit Matt Thomas Global Support

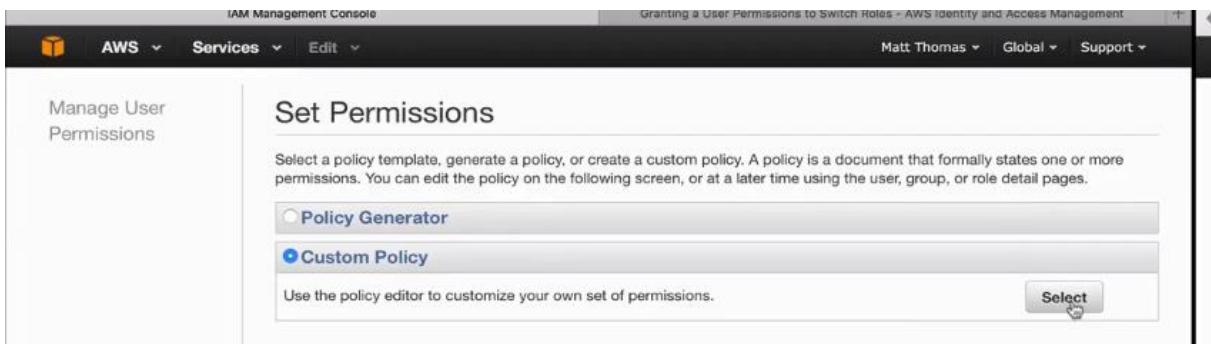
Manage User Permissions

Set Permissions

Select a policy template, generate a policy, or create a custom policy. A policy is a document that formally states one or more permissions. You can edit the policy on the following screen, or at a later time using the user, group, or role detail pages.

Policy Generator
 Custom Policy

Use the policy editor to customize your own set of permissions. [Select](#)



IAM Management Console Granting a User Permissions to Switch Roles - AWS Identity and Access Management

AWS Services Edit Matt Thomas Global Support

Manage User Permissions

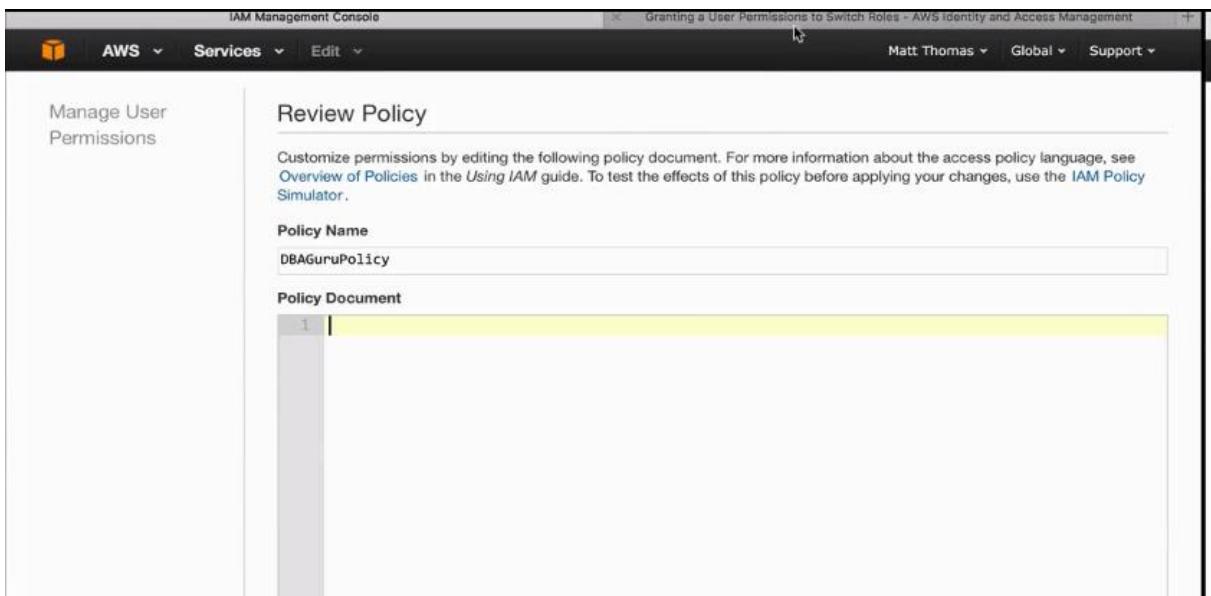
Review Policy

Customize permissions by editing the following policy document. For more information about the access policy language, see Overview of Policies in the Using IAM guide. To test the effects of this policy before applying your changes, use the IAM Policy Simulator.

Policy Name
DBAGuruPolicy

Policy Document

```
1 |
```



IAM Management Console | Granting a User Permissions to Switch Roles - AWS Identity and Access Management

Menu  English Sign In to the Console

AWS Identity and Access Management

User Guide

Documentation - This Guide

+ What Is IAM?
□ Getting Set Up
+ Getting Started
+ Tutorials
+ Best Practices and Use Cases
+ IAM Console and Sign-In Page
- Identities
 + Users
 + Groups
- Roles
 □ Terms and Concepts
 + Common Scenarios
 + Identity Providers and Federation
+ Creating Roles
- Using Roles
 □ **Granting Permissions to Switch Roles**
 □ Switching Roles (AWS Console)
 □ Switching Roles (AWS CLI)

permissions to a user to accomplish a task. However, it is [best practice not to grant permissions directly to an individual user](#). For easier management, we recommend assigning policies and granting permissions to IAM groups and then making the users members of the appropriate groups.

Creating or Editing the Policy

A policy that grants a user permission to assume a role must include a statement with the `Allow` effect on the `sts:AssumeRole` action and the Amazon Resource Name (ARN) of the role in a `Resource` element, as shown in the following example. Users that get the policy (either through group membership or directly attached) are allowed to switch to the specified role.

Note

Note that if `Resource` is set to `*`, the user can assume any role in any account that trusts the user's account (the role's trust policy specifies the user's account as `Principal`). As a best practice, we recommend that you follow the [principle of least privilege](#) and specify the complete ARN for only the role(s) that the user needs.

The following example shows a policy that lets the user assume roles in only one account and additionally specifies by wildcard (*) that the user can only switch to a role in that account if the role name begins with the letters "Test" followed by any other combination of characters.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "sts:AssumeRole",  
      "Resource": "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:role/Test*"  
    }  
  ]  
}
```

Note

The permissions that the role grants to the user do not add to the permissions already granted to the user. When a user switches to a role, the user

IAM Management Console | Granting a User Permissions to Switch Roles - AWS Identity and Access Management

AWS Services Edit Matt Thomas Global Support

Manage User Permissions

Review Policy

Customize permissions by editing the following policy document. For more information about the access policy language, see [Overview of Policies](#) in the [Using IAM](#) guide. To test the effects of this policy before applying your changes, use the [IAM Policy Simulator](#).

Policy Name
DBAGuruPolicy

Policy Document

```
1- {  
2-   "Version": "2012-10-17",  
3-   "Statement": [  
4-     {  
5-       "Effect": "Allow",  
6-       "Action": "sts:AssumeRole",  
7-       "Resource": "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:role/Test*"  
8-     }  
9-   ]  
10 }
```

IAM Management Console Granting a User Permissions to Switch Roles - AWS Identity and Access Management

AWS Services Edit Matt Thomas Global Support

Manage User Permissions

Review Policy

Customize permissions by editing the following policy document. For more information about the access policy language, see Overview of Policies in the Using IAM guide. To test the effects of this policy before applying your changes, use the IAM Policy Simulator.

Policy Name

DBAGuruPolicy

Policy Document

```

1  {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "sts:AssumeRole",
7       "Resource": "arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:role/Test"
8     }
]

```

IAM Management Console Granting a User Permissions to Switch Roles - AWS Identity and Access Management

AWS Services Edit Matt Thomas Global Support

Manage User Permissions

Review Policy

Customize permissions by editing the following policy document. For more information about the access policy language, see Overview of Policies in the Using IAM guide. To test the effects of this policy before applying your changes, use the IAM Policy Simulator.

This policy is valid.

Policy Name

DBAGuruPolicy

Policy Document

```

1  {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "sts:AssumeRole",
7       "Resource": "arn:aws:iam::515331423217:role/DatabaseMasterGuru"
8     }
]

```

We then copy the needed values from the Angelina browser as above, then we validate the Policy to make sure the JSON is valid. Then click the Apply Policy button

IAM Management Console https://signin.aws.amazon.com/switchrole

AWS Services Edit Matt Thomas Global Support

Dashboard IAM > Users > Angelina Summary

Details Groups **Users** Roles Policies Identity Providers Account Settings Credential Report Encryption Keys

Search IAM

User ARN: arn:aws:iam::515331423217:user/Angelina

Has Password: Yes

Groups (for this user): 1

Path: /

Creation Time: 2016-07-07 08:27 UTC+0100

Groups **Permissions** Security Credentials Access Advisor

Managed Policies

There are no managed policies attached to this user.

Attach Policy

Inline Policies

This view shows all inline policies that apply to this user, including policies that are embedded in this user and policies that are embedded in groups that this user is in.

Create User Policy

Policy Name	Actions
DBAGuruPolicy	Show Policy Edit Policy Remove Policy Simulate Policy

Switch Role

Allows management of resources across AWS accounts using a single user ID and password. You can switch roles after an AWS administrator has configured a role and given you the account and role details. Learn more.

Could not switch roles using the provided information. Please check your settings and try again. If you continue to have problems, contact your administrator.

Account* 515331423217 Role* DatabaseMasterGuruData@515331423217 Display Name DatabaseMasterGuru @ 515331423217 Color a a a a a a

*Required Cancel Switch Role

English Terms of Use Privacy Policy © 1996-2016, Amazon Web Services, Inc. or its affiliates.

The User Angelina now has the DBAGuruPolicy attached to it as an inline Policy, this means that it can only be attached to this User only

AWS Services

Compute

- EC2** Virtual Servers in the Cloud
- EC2 Container Service** Run and Manage Docker Containers
- Elastic Beanstalk** Run and Manage Web Apps
- Lambda** Run Code in Response to Events

Storage & Content Delivery

- S3** Scalable Storage in the Cloud
- CloudFront** Global Content Delivery Network
- Elastic File System** Fully Managed File System for EC2
- Glacier** Archive Storage in the Cloud
- Snowball** Large Scale Data Transport
- Storage Gateway** Hybrid Storage Integration

Developer Tools

- CodeCommit** Store Code in Private Git Repositories
- CodeDeploy** Automate Code Deployments
- CodePipeline** Release Software using Continuous Delivery

Management Tools

- CloudWatch** Monitor Resources and Applications
- CloudFormation** Create and Manage Resources with Templates
- CloudTrail** Track User Activity and API Usage
- Config** Track Resource Inventory and Changes
- OpsWorks** Automate Operations with Chef
- Service Catalog** Create and Use Standardized Products

Internet of Things

- AWS IoT** Connect Devices to the Cloud

Game Development

- GameLift** Deploy and Scale Session-based Multiplayer Games

Mobile Services

- Mobile Hub** Build, Test, and Monitor Mobile Apps
- Cognito** User Identity and App Data Synchronization
- Device Farm** Test Android, iOS, and Web Apps on Real Devices in the Cloud
- Mobile Analytics** Collect, View and Export App Analytics
- SNS** Push Notification Service

Application Services

- API Gateway** Build, Deploy and Manage APIs
- AppStream** Low Latency Application Streaming
- CloudSearch** Managed Search Service

Resource Groups

A resource group is a collection of resources that share one or more tags. Create a group for each project, application, or environment in your account.

Create a Group Tag

AWS Services

Compute

- EC2** Virtual Servers in the Cloud
- EC2 Container Service** Run and Manage Docker Containers
- Elastic Beanstalk** Run and Manage Web Apps
- Lambda** Run Code in Response to Events

Storage & Content Delivery

- S3** Scalable Storage in the Cloud
- CloudFront** Global Content Delivery Network
- Elastic File System** Fully Managed File System for EC2
- Glacier** Archive Storage in the Cloud
- Snowball** Large Scale Data Transport
- Storage Gateway** Hybrid Storage Integration

Developer Tools

- CodeCommit** Store Code in Private Git Repositories
- CodeDeploy** Automate Code Deployments
- CodePipeline** Release Software using Continuous Delivery

Management Tools

- CloudWatch** Monitor Resources and Applications
- CloudFormation** Create and Manage Resources with Templates
- CloudTrail** Track User Activity and API Usage
- Config** Track Resource Inventory and Changes
- OpsWorks** Automate Operations with Chef
- Service Catalog** Create and Use Standardized Products

Internet of Things

- AWS IoT** Connect Devices to the Cloud

Game Development

- GameLift** Deploy and Scale Session-based Multiplayer Games

Mobile Services

- Mobile Hub** Build, Test, and Monitor Mobile Apps
- Cognito** User Identity and App Data Synchronization
- Device Farm** Test Android, iOS, and Web Apps on Real Devices in the Cloud
- Mobile Analytics** Collect, View and Export App Analytics
- SNS** Push Notification Service

Application Services

- API Gateway** Build, Deploy and Manage APIs
- AppStream** Low Latency Application Streaming
- CloudSearch** Managed Search Service

IAM User: Angelina

Account: 5153-3142-3217

My Account

Billing & Cost Management

Security Credentials

Role History:

- DatabaseMasterGuru @ 515331423217

Switch Role

Sign Out

from Amazon Appstore, Google iTunes.

AWS Marketplace

Find and buy software, launch services, and more. Click and pay by the hour.

AWS re:Invent Announcements

Explore the next generation of AWS cloud capabilities. See what's new.

Service Health

Angelina is now operating within the AWS account as the DatabaseMasterGuru Role

We can click the **Back to Angelina** link and that would take us back to the Angelina Role and we would not be able to do the things that the DatabaseMasterGuru Role can only do.

<https://us-west-2.console.aws.amazon.com/dynamodb/home?region=us-west-2#>

AWS Services Edit DatabaseMasterGuru @ 515331423217 Oregon Support

DynamoDB

Dashboard Tables Reserved capacity

Create table

Amazon DynamoDB is a fully managed non-relational database service that provides fast and predictable performance with seamless scalability.

Create table

Recent alerts

No CloudWatch alarms have been triggered.

[View all in CloudWatch](#)

Total capacity for US West (Oregon)

Provisioned read capacity	15	Reserved read capacity	0
Provisioned write capacity	15	Reserved write capacity	0

Service health

Current Status	Details
● Amazon DynamoDB (Oregon)	Service is operating normally

[View complete service health details](#)

Best practices

Amazon DynamoDB January 2016 Day at... ↗

How to Design NoSQL Tables and Avoid Hot Keys

Excerpt from:  **DynamoDB Day** January 20, 2016

What's new

- Enhanced metrics
- Titan graph database integration
- Elasticsearch integration

Related services

- Amazon ElastiCache

Additional resources

- Getting started guide
- Getting started hands-on lab
- FAQ
- Release notes
- Developer guide

<https://us-west-2.console.aws.amazon.com/dynamodb/home?region=us-west-2#create-table>

AWS Services Edit DatabaseMasterGuru @ 515331423217 Oregon Support

Create DynamoDB table

[Tutorial](#) [?](#)

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name*

Primary key* Partition key String [Edit](#) [Help](#)

Add sort key

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

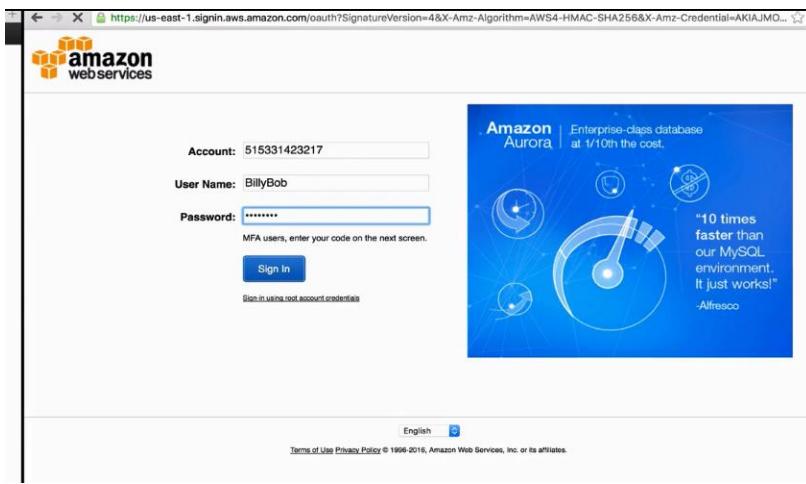
[Cancel](#) **Create**

The screenshot shows the AWS DynamoDB console. On the left, the navigation bar includes 'AWS', 'Services', 'Edit', and a user dropdown 'DatabaseMasterGuru @ 515331423217'. The main area displays a table list with 'AngelinaGuru' selected. The 'Overview' tab is active, showing a message 'Table is being created'. Under 'Stream details', it says 'Stream enabled: No'. In the 'Table details' section, the table name is 'AngelinaGuru', primary partition key is 'id (String)', and status is 'Creating'.

This screenshot is similar to the previous one, but a modal window is open over the table details. The modal shows 'Role History' with 'DatabaseMasterGuru @ 515331423217' listed. It also shows 'My Account' and 'Billing & Cost Management' links, and a 'Back to Angelina' button.

The screenshot shows the same AWS DynamoDB interface, but the table list now displays a red warning icon and the text 'Not Authorized' for all tables. The user dropdown at the top shows 'Angelina @ 5153-3142-3217'. The overall status is 'Not Authorized'.

We are now back to Angelina Role that does not have access to DynamoDB.



We can also log in as the user BillyBob who does not have access to DynamoDB too

IAM Management Console - User Details:

User ARN	Has Password	Groups (for this user)	Path	Creation Time
arn:aws:iam::515331423217:user/Angelina	Yes	1	/	2016-07-07 08:27 UTC+0100

IAM Management Console - Policies:

Policy Name	Actions
DBAGuruPolicy	Show Policy Edit Policy Remove Policy Simulate Policy

DynamoDB Management Console - BillyBob:

How would you rate your experience with the DynamoDB management console? ★ ★ ★ ★ ★

BillyBob @ 5153-3142-3217 • Oregon • Support

No Tables

⚠ Not Authorized

IAM Management Console

Granting a User Permissions to Switch Roles – AWS Identity and Access Management

Matt Thomas Global Support

Users

IAM > Users > Angelina

Summary

User ARN: arn:aws:iam::51531423217:user/Angelina
Has Password: Yes
Groups (for this user): 1
Path: /
Creation Time: 2016-07-07 08:27 UTC+0100

Groups **Permissions** **Security Credentials** **Access Advisor**

Managed Policies
There are no managed policies attached to this user.

Inline Policies
This view shows all inline policies that apply to this user, including policies that are embedded in this user and policies that are embedded in groups that this user is in.

Create User Policy

Policy Name	Actions
DBAGuruPolicy	Show Policy Edit Policy Remove Policy Simulate Policy

DynamoDB

How would you rate your experience with the DynamoDB management console? ★ ★ ★ ★ ★

BillyBob @ 5153-3142-3217

No Tables

Name Status Partition key

Not Authorized

Instead of assigning the DBAGuruPolicy Policy to only the User Angelina so that they can assume the DatabaseMasterGuru Role. We are going to attach this DBAPolicy to the MyDBTeam Group so that all its members can assume that DatabaseMasterGuru Role and do things in DynamoDB.

IAM Management Console

Granting a User Permissions to Switch Roles – AWS Identity and Access Management

Matt Thomas Global Support

Users

IAM > Users > Angelina

Summary

User ARN: arn:aws:iam::51531423217:user/Angelina
Has Password: Yes
Groups (for this user): 1
Path: /
Creation Time: 2016-07-07 08:27 UTC+0100

Groups **Permissions** **Security Credentials** **Access Advisor**

Managed Policies

Show Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::51531423217:role/DatabaseMasterGuru"
    }
  ]
}
```

DynamoDB

How would you rate your experience with the DynamoDB management console? ★ ★ ★ ★ ★

BillyBob @ 5153-3142-3217

No Tables

Name Status Partition key

Not Authorized

Make a copy of the Policy from Angelina as above

IAM Management Console

Granting a User Permissions to Switch Roles – AWS Identity and Access Management

Matt Thomas Global Support

Groups

Dashboard Services Edit

Create New Group Group Actions

Showing 1 results

Group Name	Users	Inline Policy	Creation Time
MyDBTeam	2		2016-07-07 08:30 UTC+0100

DynamoDB

How would you rate your experience with the DynamoDB management console? ★ ★ ★ ★ ★

BillyBob @ 5153-3142-3217

No Tables

Name Status Partition key

Not Authorized

The screenshots illustrate the process of granting permissions to a group. In the IAM Groups section, a custom policy named 'GroupAssumeGuru' is attached to the 'MyDBTeam' group. This policy allows members of the group to assume the 'DatabaseMasterGuru' role. The policy document is as follows:

```

1. {
2.     "Version": "2012-10-17",
3.     "Statement": [
4.         {
5.             "Effect": "Allow",
6.             "Action": "sts:AssumeRole",
7.             "Resource": "arn:aws:iam::515331423217:role/DatabaseMasterGuru"
8.         }
9.     ]
}

```

Now any member of the Group MyDBTeam have the privileges to assume the DatabaseMasterGuru Role via the attached GroupAssumeGuru Policy above

IAM > Groups > MyDBTeam

Summary

Group ARN: arn:aws:iam::515331423217:group/MyDBTeam

Users (in this group): 2

Path: /

Creation Time: 2016-07-07 08:30 UTC+0100

Permissions

Managed Policies

There are no managed policies attached to this group.

Attach Policy

Inline Policies

This view shows all inline policies that are embedded in this group.

Create Group Policy

Policy Name	Actions
GroupAssumeGuru	Show Policy Edit Policy Remove Policy Simulate Policy

Amazon Web Services

- Compute: EC2, Lambda
- Developer Tools: CodeCommit, CodeDeploy, CodePipeline
- Internet of Things: AWS IoT
- Game Development: GameLift
- Management Tools: CloudWatch Metrics, CloudFormation, CloudTrail, Config, OpsWorks
- Mobile Services: Mobile Hub, Cognito, Device Farm
- Storage & Content Delivery: S3, CloudFront, CloudFront Network
- Database: Elastic File System, Amazon Glacier, Snowball, Storage Gateway, RDS
- Security & Identity: Trusted Advisor, Identity & Access Management

BillyBob @ 5153-3142-3217

My Account, Billing & Cost Management, Security Credentials, Role History, DatabaseMasterGuru @ 515331423217, Switch Role, Sign Out, AWS Marketplace, AWS re:Invent Announcements, Service Health.

We can see that the BillyBob User now has the DatabaseMasterGuru Role available that he can switch into to do DynamoDB related work

IAM > Groups > MyDBTeam

Summary

Group ARN: arn:aws:iam::515331423217:group/MyDBTeam

Users (in this group): 2

Path: /

Creation Time: 2016-07-07 08:30 UTC+0100

Permissions

Managed Policies

There are no managed policies attached to this group.

Attach Policy

Inline Policies

This view shows all inline policies that are embedded in this group.

Create Group Policy

Policy Name	Actions
GroupAssumeGuru	Show Policy Edit Policy Remove Policy Simulate Policy

Switch Role

Allows management of resources across AWS accounts using a single user ID and password. You can switch roles after an AWS administrator has configured a role and given you the account and role details. [Learn more](#).

Account*

Role*

Display Name

Color a a a a a a a

*Required Cancel

English [Change](#)

[Terms of Use](#) [Privacy Policy](#) © 1996-2016, Amazon Web Services, Inc. or its affiliates.

IAM > Groups > MyDBTeam

Summary

Group ARN: arn:aws:iam::515331423217:group/MyDBTeam

Users (in this group): 2

Path: /

Creation Time: 2016-07-07 08:30 UTC+0100

Permissions

Managed Policies

Show Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::515331423217:role/DatabaseMasterGuru"
    }
  ]
}
```

Cancel

Switch Role

Allows management of resources across AWS accounts using a single user ID and password. You can switch roles after an AWS administrator has configured a role and given you the account and role details. [Learn more](#).

Account* 515331423217

Role* DatabaseMasterGuru

Display Name DatabaseMasterGuru @ 5

Color a a a a a a a

*Required Cancel

English [Change](#)

[Terms of Use](#) [Privacy Policy](#) © 1996-2016, Amazon Web Services, Inc. or its affiliates.

<https://us-west-2.console.aws.amazon.com/console/home?region=us-west-2#>

AWS Services

Amazon Web Services

Compute

- EC2 Virtual Servers in the Cloud
- EC2 Container Service Run and Manage Docker Containers
- Elastic Beanstalk Run and Manage Web Apps
- Lambda Run Code in Response to Events

Storage & Content Delivery

- S3 Scalable Storage in the Cloud
- CloudFront Global Content Delivery Network
- Elastic File System Fully Managed File System for EC2
- Glacier Archive Storage in the Cloud
- Snowball Large Scale Data Transport
- Storage Gateway Hybrid Storage Integration

Database

- RDS

Developer Tools

- CodeCommit Store Code in Private Git Repositories
- CodeDeploy Automate Code Deployments
- CodePipeline Release Software using Continuous Delivery

Management Tools

- CloudWatch Monitor Resources and Applications
- CloudFormation Create and Manage Resources with Templates
- CloudTrail Track User Activity and API Usage
- Config Track Resource Inventory and Changes
- OpsWorks Automate Operations with Chef
- Service Catalog Create and Use Standardized Products
- Trusted Advisor Optimize Performance and Security

Internet of Things

- AWS IoT Connect Devices to the Cloud

Game Development

- GameLift Deploy and Scale Session-based Multiplayer Games

Mobile Services

- Mobile Hub Build, Test, and Monitor Mobile Apps
- Cognito User Identity and App Data Synchronization
- Device Farm Test Android, iOS, and Web Apps on Real Devices in the Cloud
- Mobile Analytics Collect, View and Export App Analytics
- SNS Push Notification Service

Application Services

- API Gateway Build, Deploy and Manage APIs
- AppStream Low Latency Application Streaming

Security & Identity

- Identity & Access Management

Resource Groups

A resource group is a collection of resources that share one or more tags. Create a group for each project, application, or environment in your account.

Create a Group Tag Editor

Additional Resources

Getting Started Read our documentation or view our training to learn more about AWS.

AWS Console Mobile App View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.

AWS Marketplace Find and buy software, launch with 1-Click and pay by the hour.

AWS re:Invent Announcements Explore the next generation of AWS cloud

<https://us-west-2.console.aws.amazon.com/console/home?region=us-west-2#>

AWS Services

Amazon Web Services

Compute

- EC2 Virtual Servers in the Cloud
- EC2 Container Service Run and Manage Docker Containers
- Elastic Beanstalk Run and Manage Web Apps
- Lambda Run Code in Response to Events

Storage & Content Delivery

- S3 Scalable Storage in the Cloud
- CloudFront Global Content Delivery Network
- Elastic File System Fully Managed File System for EC2
- Glacier Archive Storage in the Cloud
- Snowball Large Scale Data Transport
- Storage Gateway Hybrid Storage Integration

Database

- RDS

Developer Tools

- CodeCommit Store Code in Private Git Repositories
- CodeDeploy Automate Code Deployments
- CodePipeline Release Software using Continuous Delivery

Management Tools

- CloudWatch Monitor Resources and Applications
- CloudFormation Create and Manage Resources with Templates
- CloudTrail Track User Activity and API Usage
- Config Track Resource Inventory and Changes
- OpsWorks Automate Operations with Chef
- Service Catalog Create and Use Standardized Products
- Trusted Advisor Optimize Performance and Security

Internet of Things

- AWS IoT Connect Devices to the Cloud

Game Development

- GameLift Deploy and Scale Session-based Multiplayer Games

Mobile Services

- Mobile Hub Build, Test, and Monitor Mobile Apps
- Cognito User Identity and App Data Synchronization
- Device Farm Test Android, iOS, and Web Apps on Real Devices in the Cloud
- Mobile Analytics Collect, View and Export App Analytics
- SNS Push Notification Service

Application Services

- API Gateway Build, Deploy and Manage APIs
- AppStream Low Latency Application Streaming

Security & Identity

- Identity & Access Management

Learn more

Logged in as: [BillyBob](#)

Currently active as: [DatabaseMasterGuru](#)

Account: 5153-3142-3217

Action of or more tags. Project, application, count.

Role History:

- My Account
- Billing & Cost Management
- Switch Role Back to BillyBob

Sign Out

Read our documentation or view our training to learn more about AWS.

AWS Console Mobile App View your resources on the go with our AWS Console mobile app, available from Amazon Appstore, Google Play, or iTunes.

AWS Marketplace Find and buy software, launch with 1-Click and pay by the hour.

AWS re:Invent Announcements Explore the next generation of AWS cloud

<https://us-west-2.console.aws.amazon.com/dynamodb/home?region=us-west-2#>

AWS Services

DynamoDB

Dashboard

Tables

Recent alerts

No CloudWatch alarms have been triggered.

Create table

Amazon DynamoDB is a fully managed non-relational database service that provides fast and predictable performance with seamless scalability.

Total capacity for US West (Oregon)

Provisioned read capacity	20	Reserved read capacity	0
Provisioned write capacity	20	Reserved write capacity	0

Service health

Current Status	Details
Green checkmark Amazon DynamoDB (Oregon)	Service is operating normally

View complete service health details

Best practices

Amazon DynamoDB January 2016 Day at the ...

How to Design NoSQL Tables and Avoid Hot Keys

Excerpt from:  January 20, 2016

What's new

- Enhanced metrics
- Titan graph database integration
- Elasticsearch integration

Related services

- Amazon ElastiCache

<https://us-west-2.console.aws.amazon.com/dynamodb/home?region=us-west-2#tables>

DynamoDB

Dashboard

Tables

Reserved capacity

Create table Actions

Filter by table name X

Viewing 4 of 4 Tables

Name	Status	Partition key	Sort key	Indexes	Total read capacity	Total write capacity
AngelinaGuru	Active	id (String)	-	0	5	5
AngelinaTable	Active	id (String)	-	0	5	5
BillyBobTable	Active	id (String)	-	0	5	5
OregonTable	Active	id (String)	-	0	5	5

<https://us-west-2.console.aws.amazon.com/dynamodb/home?region=us-west-2#create-table>

Create DynamoDB table

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* ⓘ

Primary key* Partition key
 String ⓘ
 Add sort key

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

Cancel **Create**

<https://us-west-2.console.aws.amazon.com/dynamodb/home?region=us-west-2#tables:selected=BillyBobGuruTable>

DynamoDB

Dashboard

Tables

Reserved capacity

Create table Actions

Filter by table name X

BillyBobGuruTable Close

Overview Items Metrics Alarms Capacity Indexes Triggers Access control

Table is being created

Recent alerts

No CloudWatch alarms have been triggered for this table.

Stream details

Stream enabled	No
View type	-
Latest stream ARN	-

Manage Stream

Table details

Table name	BillyBobGuruTable
Primary partition key	id (String)
Primary sort key	-
Table status	Creating
Creation date	July 7, 2016 at 9:03:10 AM UTC+1
Provisioned read capacity units	5
Provisioned write capacity units	5
Last decrease time	-
Last increase time	-

The screenshot shows the AWS DynamoDB console with the URL <https://us-west-2.console.aws.amazon.com/dynamodb/home?region=us-west-2#tables:selected=BillyBobGuruTable>. The user is logged in as **DatabaseMasterGuru @ 515331423217** in the Oregon region. On the left, the navigation bar shows **DynamoDB**, **Dashboard**, **Tables** (selected), and **Reserved capacity**. The main area displays the **BillyBobGuruTable** overview, which is currently being created. The table details are as follows:

Table name	BillyBobGuruTable
Primary partition key	id (String)
Primary sort key	-
Table status	Creating
Creation date	July 7, 2016 at 9:03:10 AM UTC+1
Provisioned read capacity units	5
Provisioned write capacity units	5
Last decrease time	-

A context menu is open over the table name, listing options: **Logged in as:** DatabaseMasterGuru, **Currently active as:** DatabaseMasterGuru, **Account:** 5153-3142-3217, **Role History:** DatabaseMasterGuru @ 515331423217, **Switch Role**, **Sign Out**, and **Manage Stream**. The option **Back to BillyBob** is highlighted with a mouse cursor.

If we go back to BillyBob User as above

The screenshot shows the AWS DynamoDB console with the URL <https://us-west-2.console.aws.amazon.com/dynamodb/home?region=us-west-2#tables:selected=BillyBobGuruTable>. The user is now logged in as **BillyBob @ 5153-3142-3217** in the Oregon region. The interface is identical to the previous screenshot, but a red error message **⚠ Not Authorized** is displayed prominently.

We no longer have access to DynamoDb anymore. This gives us an idea about setting up Users, Groups and Roles in AWS.