Access DynamoDB from Python

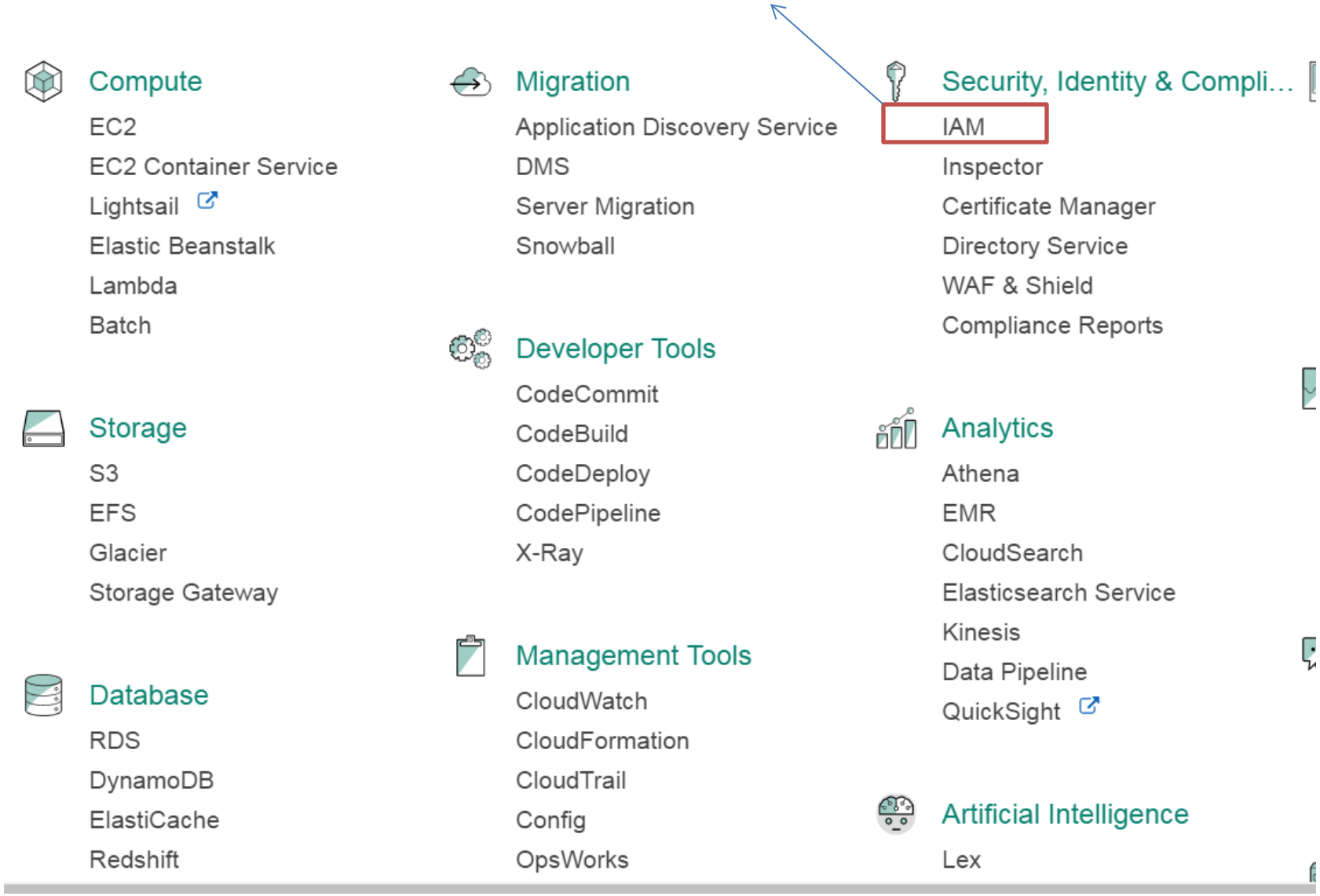
INF 551

Wensheng Wu

Steps

- Create a user in Amazon IAM
- Set permission for user to access DynamoDB
- Download access ID and key
- Configure using AWS CLI on EC2

IAM (Identity and Access Management)



Add a user & programmatic access

Set user details

You can add multiple users at once with the same access type and permissions. Learn more

User name* inf551

Add another user

Select AWS access type

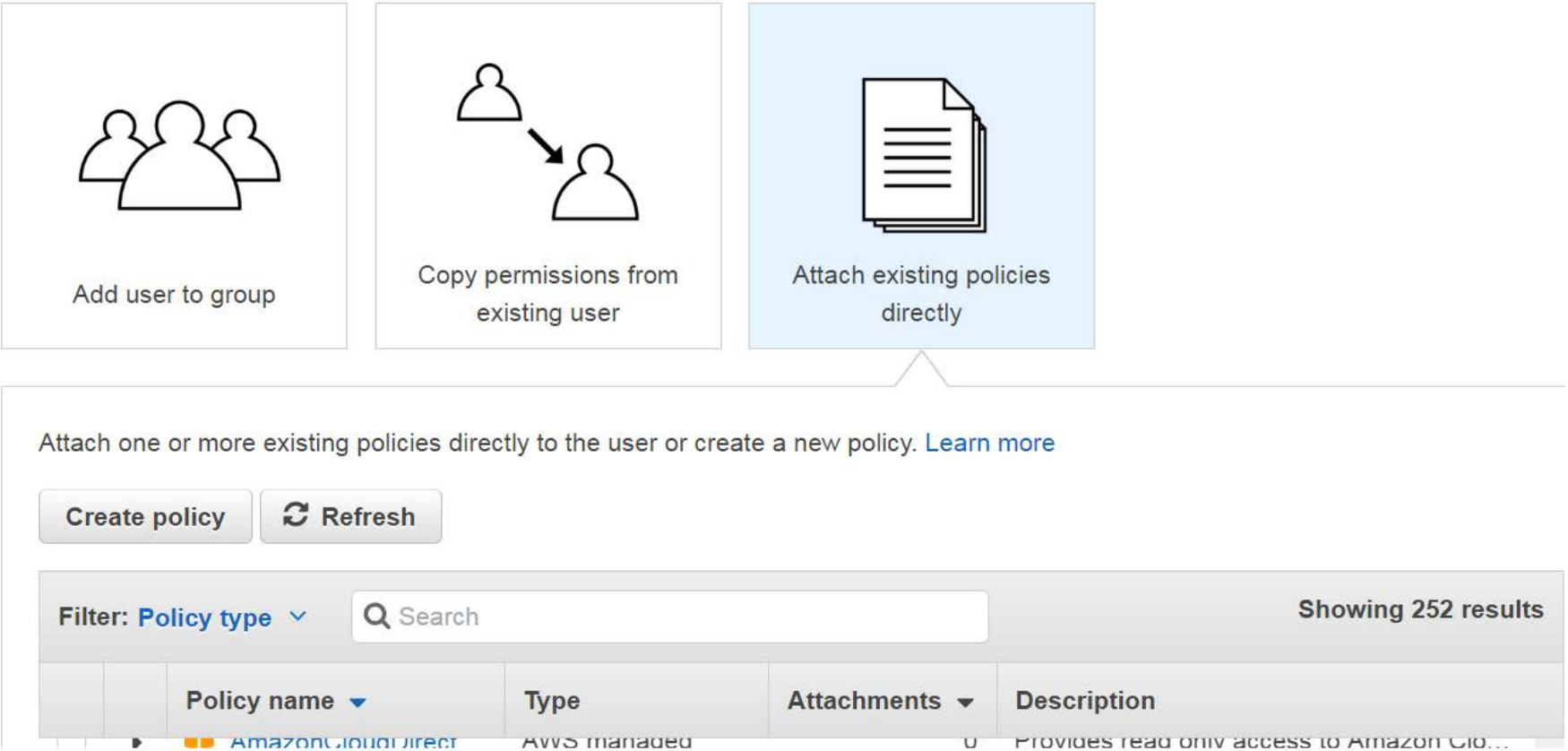
Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more

Access type*

- [x] **Programmatic access**
  Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
- [ ] **AWS Management Console access**
  Enables a **password** that allows users to sign-in to the AWS Management Console.

Set permission to access DynamoDB

Set permissions for inf551



Allow full access

Filter: Policy type | Search | Showing 252 results

Policy name	Type	Attachments	Description
AmazonDMSCloudWatchLogs...	AWS managed	0	Provides access to upload DMS replication logs to cloudwatch logs in cu...
AmazonDMSRedshiftS3Role	AWS managed	0	Provides access to manage S3 settings for Redshift endpoints for DMS.
AmazonDMSVPCManagement...	AWS managed	0	Provides access to manage VPC settings for AWS managed customer c...
AmazonDRSVPCManagement	AWS managed	0	Provides access to manage VPC settings for Amazon managed custome...
AmazonDynamoDBFullAccess	AWS managed	1	Provides full access to Amazon DynamoDB via the AWS Management C...
AmazonDynamoDBFullAccess...	AWS managed	0	Provides full access to Amazon DynamoDB including Export/Import usin...
AmazonDynamoDBReadOnlyA...	AWS managed	0	Provides read only access to Amazon DynamoDB via the AWS Manage...
AmazonEC2ContainerRegistry...	AWS managed	0	Provides administrative access to Amazon ECR resources
AmazonEC2ContainerRegistry...	AWS managed	0	Provides full access to Amazon EC2 Container Registry repositories, but...
AmazonEC2ContainerRegistry...	AWS managed	0	Provides read-only access to Amazon EC2 Container Registry repositori...

Check the settings specified

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name	inf551
AWS access type	Programmatic access - with an access key

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AmazonDynamoDBFullAccess

Create access key

- Download your id and key (& secure them)

Create access key

Success

This is the **only** time that the secret access keys can be viewed or downloaded. You cannot recover them later. However, you can create new access keys at any time.

Download .csv file

Access key ID	Secret access key
AKI[redacted]A	********* Show

Configure AWS

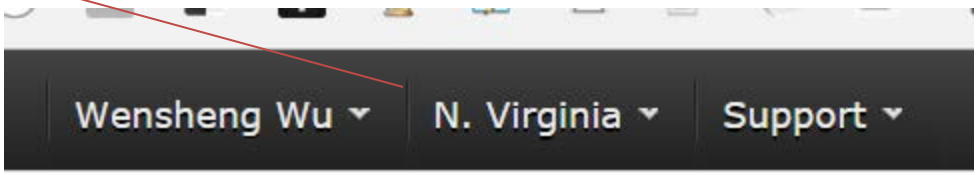
- Login to your EC2 install

- Execute "aws configure"
  - Enter id and key you downloaded
  - Enter region id (see next slide)

```
[ec2-user@ip-172-31-24-7 ~]$ aws configure
AWS Access Key ID [****************KWJA]:
AWS Secret Access Key [****************qyHm]:
Default region name [us-east-1]:
Default output format [None]:
```

Find region id

- Check region name of your account:



- Find region id here

Amazon API Gateway

Region Name	Region	Endpoint	Protocol
US East (N. Virginia)	us-east-1	apigateway.us-east-1.amazonaws.com	HTTPS
US East (Ohio)	us-east-2	apigateway.us-east-2.amazonaws.com	HTTPS
US West (N. California)	us-west-1	apigateway.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	apigateway.us-west-2.amazonaws.com	HTTPS

Install boto3

- Boto3: Python module for AWS

- sudo pip install boto3
  - If it says "pip not found", execute "sudo /usr/local/bin/pip install boto3" instead

Access DynamoDB from Python

- Enter Python interactive shell
- import boto3
- dynamodb = boto3.resource('dynamodb')

Create a table

```
table = dynamodb.create_table(
    TableName='Books',
    KeySchema=[
        {
            'AttributeName': 'Author',
            'KeyType': 'HASH'
        },
        {
            'AttributeName': 'Year',
            'KeyType': 'RANGE'
        }
    ],
    AttributeDefinitions=[
        {
            'AttributeName': 'Author',
            'AttributeType': 'S'
        },
        {
            'AttributeName': 'Year',
            'AttributeType': 'N'
        },

    ],
    ProvisionedThroughput={
        'ReadCapacityUnits': 5,
        'WriteCapacityUnits': 5
    }
)
```

Create a table

- Wait until the table exists.
  - table.meta.client.get_waiter('table_exists').wait(TableName='Books')

- Print out some data about the table.
  - print(table.item_count)

Create an item

```
table.put_item(
  Item={
      'Author': 'Bill Clinton',
      'Year': 2002,
  }
)
```

Retrieve an item

```
table = dynamodb.Table('Books')
response = table.get_item(
    Key={
        "Author": "Bill Clinton",
        "Year": 2002,
    }
)
item = response['Item']
print(item)
```

Update an item

```
table.update_item(
    Key={
        'Author': 'Bill Clinton',
        'Year': 2002
    },
    UpdateExpression='SET Title = :val1, Keywords = :val2, Ratings = :val3,
Prices = :val4, Version = :val5',
    ExpressionAttributeValues={
        ':val1': 'My Life',
        ':val2': ['History', 'Presidential2'],
            ':val3': {3, 5, 4},
            ':val4': {'Amazon': 20, 'BN': 30},
            ':val5': 3
    }
)
```

Batch write

```
with table.batch_writer() as batch:
for i in range(10):
    batch.put_item(
        Item={
            'Author': 'anonymous',
            'Year': i,
        }
    )
```

Query

- from boto3.dynamodb.conditions import Key, Attr

- response = table.query(
  KeyConditionExpression=Key('Author').eq("Bill Clinton") & Key('Year').eq(2002)
)

Key attributes

- items = response['Items']
- print(items)

Scan

- Retrieval all items
  - response = table.scan()
  - items = response['Items']
  - print(items)

Scan with a filter

- response = table.scan(

  FilterExpression=Attr("Title").eq("My Life"))

Non-key attributes

- items = response['Items']
- print(items)

Delete an item

```
table.delete_item(
    Key={
        'Author': 'Bill Clinton',
        'Year': 2002,
    }
)
```

Delete a table

- table.delete()

Resources

- Setup boto3
  - http://boto3.readthedocs.io/en/latest/guide/quickstart.html

- Access DynamoDB via boto3
  - http://boto3.readthedocs.io/en/latest/guide/dynamodb.html