COMP 6244 Coursework

Cloud Applications

Project Management Portal

Janani Sundaresan - 30575389
ECS Department, University of Southampton
js10n18@soton.ac.uk

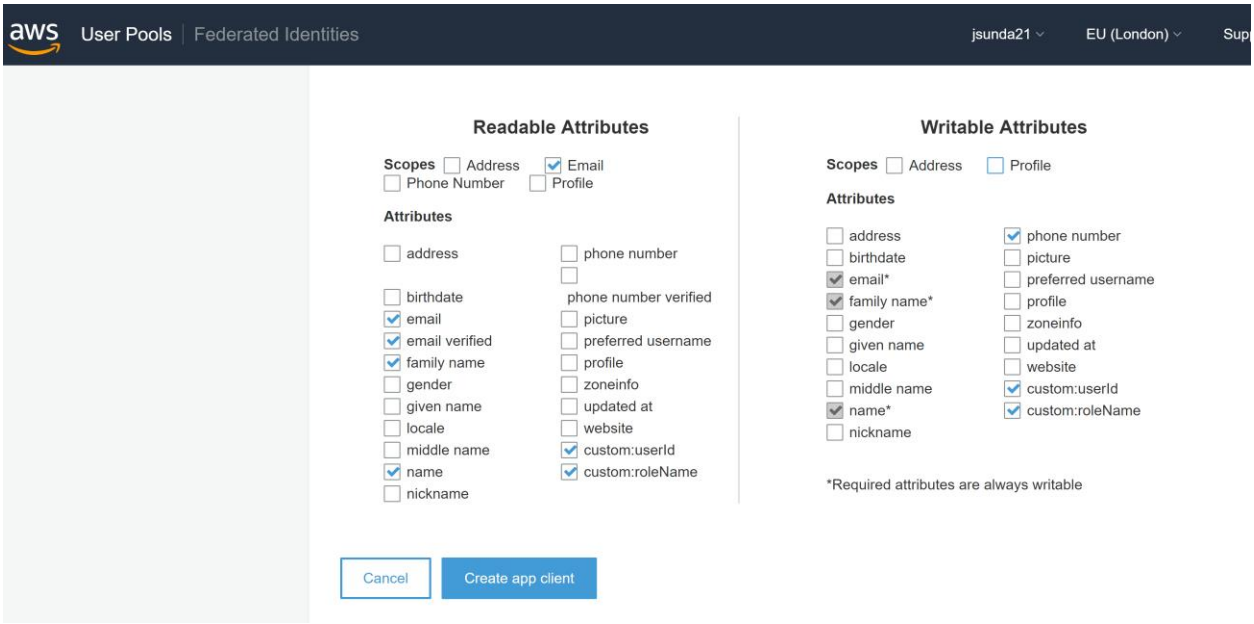
- Link to static website - http://janani-southampton.s3-website.eu-west-2.amazonaws.com
- Link to Github Repository - https://github.com/jananisundaresan/projectManagement.git
- Configured cloudfront for s3 bucket - d3w1fqi7l2w3j7.cloudfront.net

Project Management Portal Credentials

Role	Username	Password
Admin	janani	Southampton100$
Project Manager	ProjectManager	Cloud123!
Developer	Developer	Cloud1234!

UserManagement – AWS Cognito

1. Created UserPool projectManagement.
2. Each user has the following attributes.
   name
   family_name
   userId
   roleName
3. Generated app client



4. Created federated Identities for ‘projectManagement’ userpool

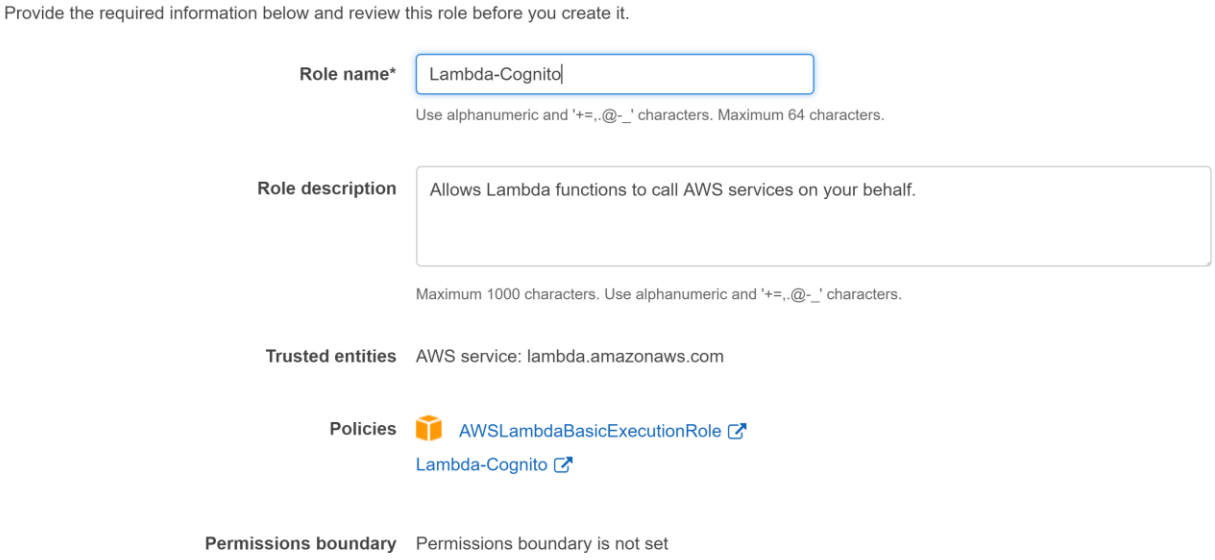
We have the following cognito values

'cognito': {
'REGION': ’ eu-west-2’,
'USER_POOL_ID': 'eu-west-2_t3Um8HkrK',
'APP_CLIENT_ID': '5qerodl71ajnqutadg7ap07to0',
'IDENTITY_POOL_ID': 'eu-west-2:d2a529ae-3444-4394-be7c-cacc0796844d'
}

5. Created Policy –‘ Lambda Cognito’ – to access AWS Cognito for User Management

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "cognito-idp:ListUsers",
      "Resource": "arn:aws:cognito-idp:eu-west-2:*:userpool/eu-west-2:d2a529ae-3444-4394-be7c-cacc0796844d"
    }
  ]
}
```

6. Created Role ‘Lambda-Cognito’



ProjectManagement - DynamoDB

1. Created two DynamoDB tables customerRoles, projectManagement

customerRoles -

Amazon Resource Name (ARN)	arn:aws:dynamodb:eu-west-2:551021002244:table/customerRoles

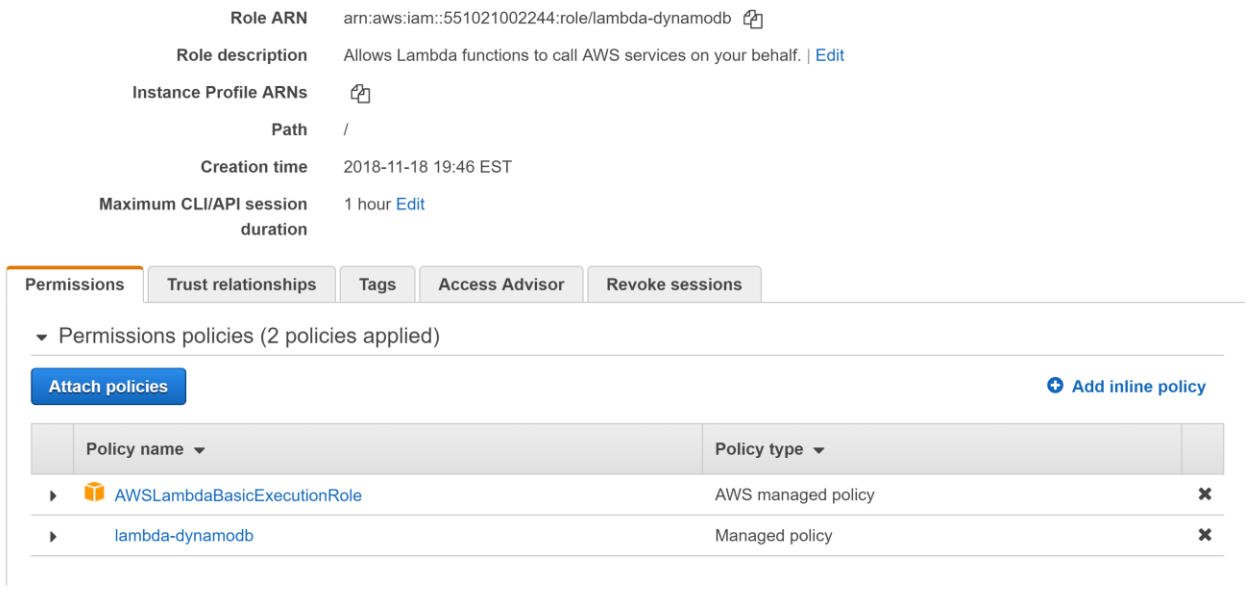
projectManagement

Amazon Resource Name (ARN) – arn:aws:dynamodb:eu-west-2:551021002244:table/projectManagement

2. Created policy lambda-dynamodb to perform CRUD operations on both datatables

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "dynamodb:BatchGetItem",
        "dynamodb:PutItem",
        "dynamodb:DescribeTable",
        "dynamodb:DeleteItem",
        "dynamodb:GetItem",
        "dynamodb:Scan",
        "dynamodb:Query",
        "dynamodb:UpdateItem"
      ],
      "Resource": [
        "arn:aws:dynamodb:eu-west-2:551021002244:table/projectManagement",
        "arn:aws:dynamodb:eu-west-2:551021002244:table/customerRoles"
      ]
    }
  ]
}
```

3. Created role lambda-dynamodb.



AWS Lambda

getUsers

- Created Lambda function ‘getUsers’ and add role ‘Lambda-Cognito’
- This function is used to get the list of users

getRole

- Created Lambda function ‘getRole’ and add role ‘lambda-dynamodb’
- This function is used to get the role of a particular user

assignRole

- Created Lambda function ‘assignRole’ and add role ‘lambda-dynamodb’
- This function is used to assign a role to a particular user

getProject

- Created Lambda function ‘getProject’ and add role ‘lambda-dynamodb’
- This function is used to get projects created by/assigned to a particular user

assignProject

- Created Lambda function ‘assignProject’ and add role ‘lambda-dynamodb’
- This function is used to create/update projects

API Gateway

/users

- Integrated with ‘getUsers’ lambda function

/project

- Integrated with ‘assignProject’ lambda function

/project/{userId}

- Integrated with ‘getProject’ lambda function

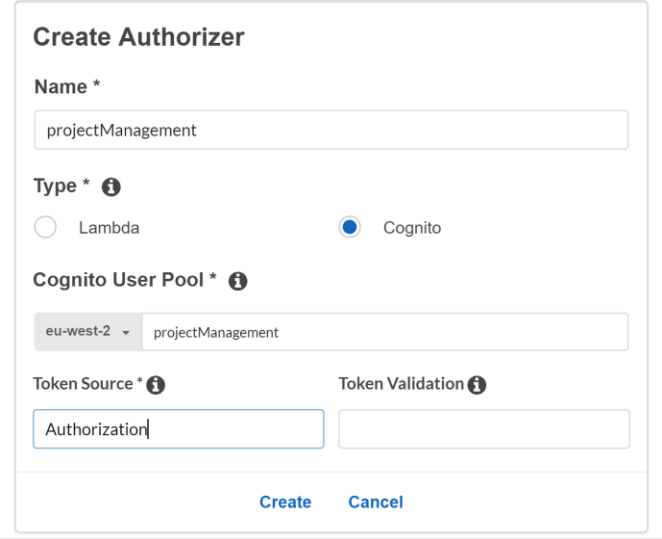
/roles

- Integrated with ‘assignRole’ lambda function

/roles/{userId}

- Integrated with 'getRole' lambda function
-

Added Authorizer to secure the API



Enable CORS and deploy APIs

Invoke URL: https://1ndav6mxce.execute-api.eu-west-2.amazonaws.com/prod

S3 bucket

Created janani-southampton s3 bucket with policy:

```
{
  "Id": "Policy1542605166569",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1542605164948",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::janani-southampton/*",
      "Principal": "*"
    }
  ]
```

```
}
```

Deploying Single Page Application – Angular into s3 and cloudfront

- Uploaded production build of the project into S3 – janani-southampton
- Created a static website
- Hosted a static website - http://janani-southampton.s3-website.eu-west-2.amazonaws.com
- Configured cloudfront for s3 bucket - d3w1fqi7l2w3j7.cloudfront.net

Time Spent:

Totally spent 70 hours in learning the project, doing all the coding in the local and finally pushing into GitHub. Took the help of materials mentioned in the References section.

User Manual:

Created a User Manual and included all the screenshots and functionalities of the portal into the manual which is available as “PM User Manual.pdf” in Github Reposistory.

References:

[1] https://github.com/youssefsharief/aws-cognito-angular-serverless-app-sample

[2] Build a Serverless App with AWS Lambda - Hands On! - https://www.oreilly.com/library/view/build-a-serverless/9781789348149/

[3] Angular UI Development with PrimeNG - https://www.oreilly.com/library/view/angular-ui-development/9781788299572/