Configuring GCN, AWS Cognito and AWS Sync

The below article describes how to configure GCN with AWS Cognito using AWS SNS in applications like multiplayer games that require data exchange between players.

GCM -> Sends Push Notification to an Android app.

AWS Cognito -> Data Store in cloud.

AWS SNS ->Web Service that coordinates and manages the delivery or sending of messages to subscribing endpoints or clients.

**The process consists of below.**

**Step 1**: GCM setup

**Step 2**. GCM integration with SNS

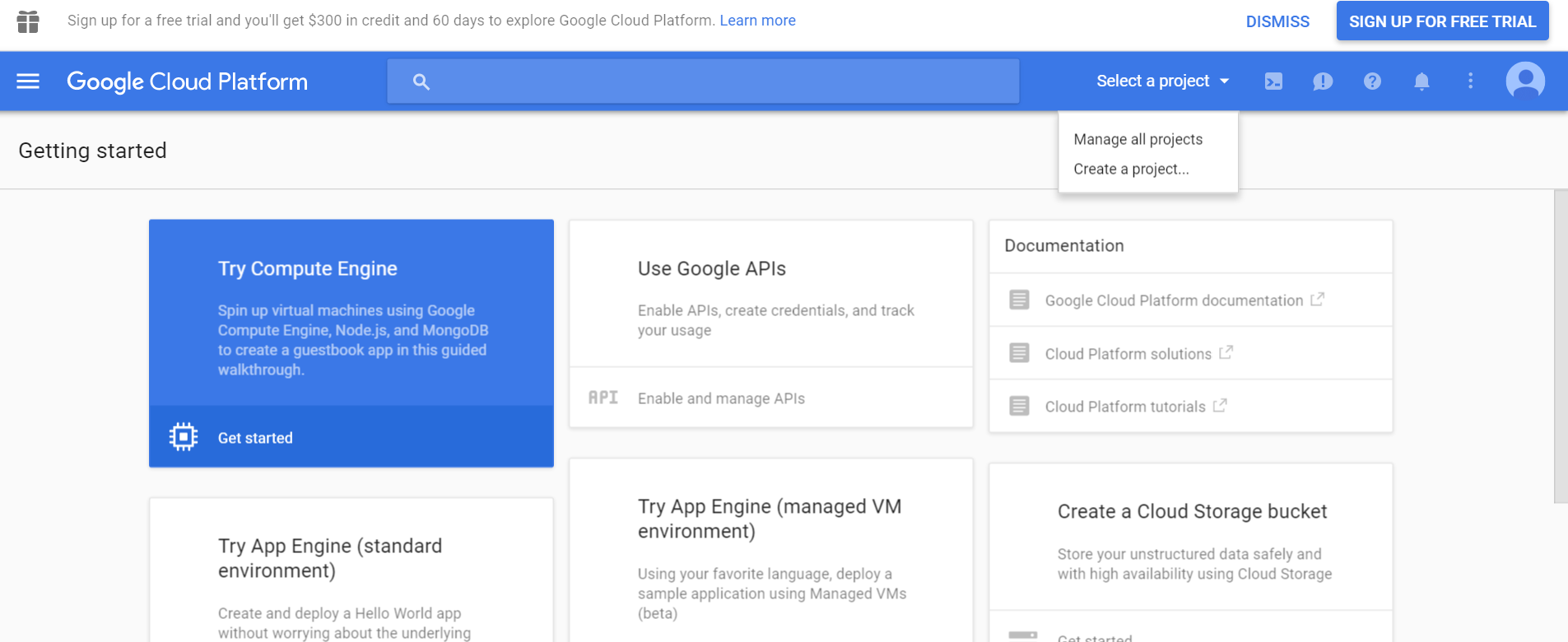
**Step 3**: AWS Cognito Pool creation.

**Step 4**: AWS Cognito to AWS SNS integration

**Step 5**: Creating IAM users and group that are authorized to make communicate with Cognito.

**Step 1: GCM setup** (Reference: <http://docs.aws.amazon.com//sns/latest/dg/mobile-push-gcm.html>)

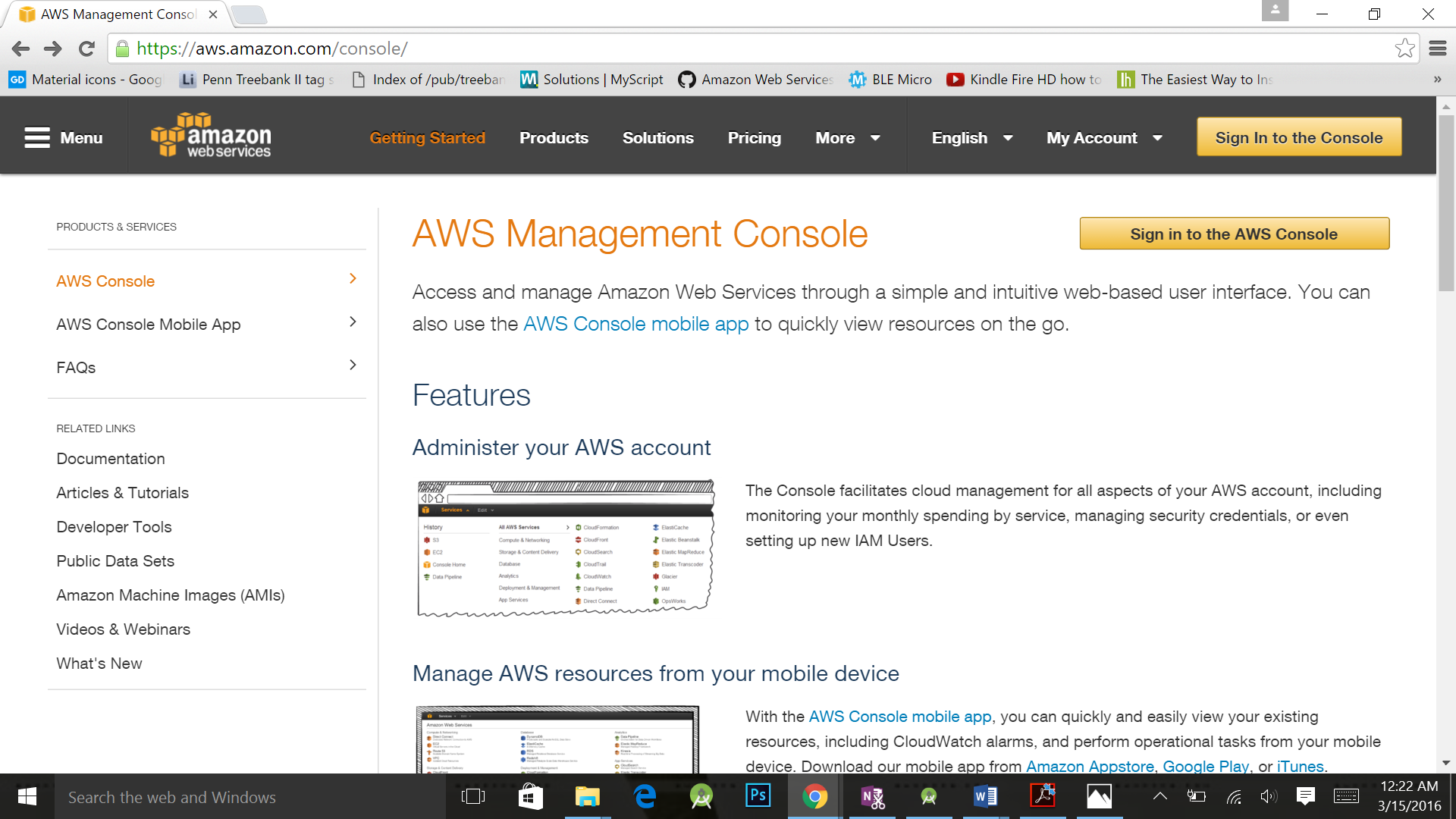
1. Go to <https://console.developers.google.com/project>
2. Click on create project



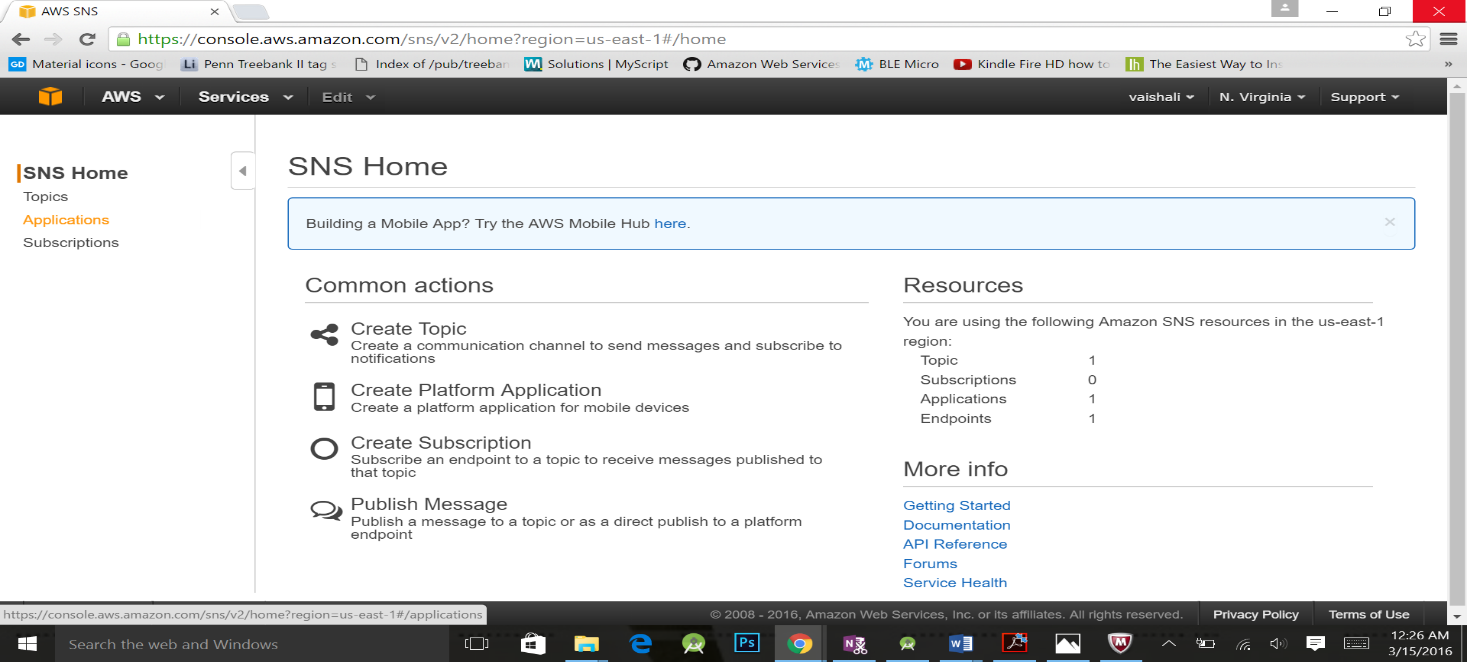
1. Add Project name and click create.
2. Go to Use Google APIs and click on “Enable and manage APIs”.
3. Under **Mobile APIs** click on “Google Cloud Messaging". Click on the **Enable** button.
4. Navigate to the credentials panel. Click “create credentials.” Select API key. Select Android key. Click create. This is the “server key” that is reference in the AWS documentation.

**Step 2. GCM integration with SNS**

1. Go the <https://console.aws.amazon.com/> and sign in to the console.



1. Select the region as **US East (N. Virginia). DO NOT SELECT ANY OTHER REGION.**
2. Select **SNS** from the Services dropdown.
3. We now need to register the application created in Step 1 with SNS.
4. Click on Applications from the left panel.

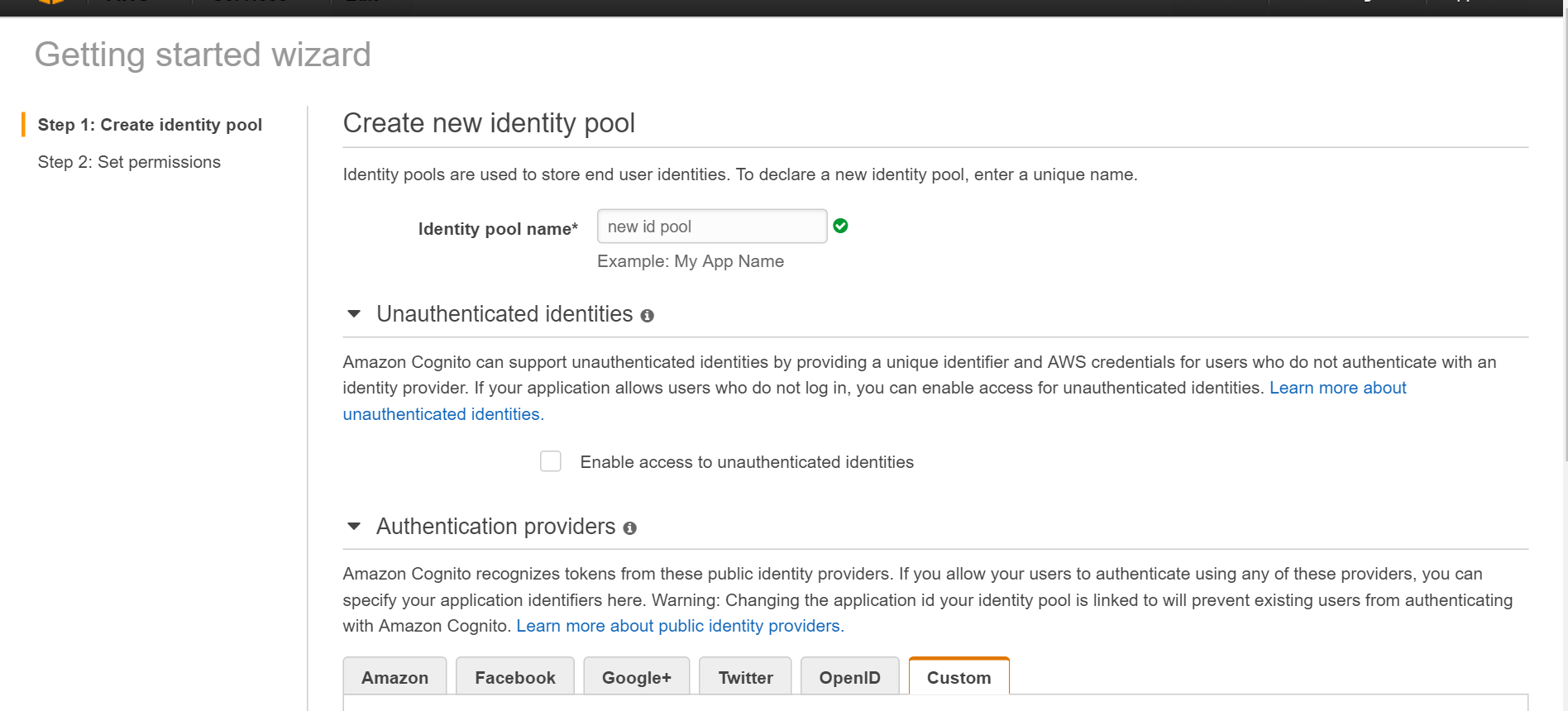


1. Click Create Platform Application. We use an Application as our endpoint and not Topic.
2. Enter the below details.
   1. Application Name: Same as that was entered in Google cloud platform.
   2. Push Notification Platform: GCM.
   3. API key: the server key that was created in Step 1.
   4. Click on create.
   5. This will generate an application ARN for SNS. Make a note of this. Ex: [arn:aws:sns:us-east-1:186727481815:app/GCM/XYZ](https://console.aws.amazon.com/sns/v2/home?region=us-east-1#/applications/arn:aws:sns:us-east-1:186727481815:app/GCM/NUMAD16S)

**Step 3: AWS Cognito Pool creation.**

**A. Creating an Identity Pool.**

1. Go to the Services and select cogito.
2. Click on “**Create new identity pool**” button.



1. Enter id pool name.
2. Click on custom under “Authenticated provider” ex given underneath.
3. Click on “**Create Pool**”.
4. Expand View Details.
5. 2 Roles are assigned by Default. One is an Auth Role for authenticated users and the other is a UnAuth Role for non-authenticated users.
6. Don’t make modification for these role names. Click on “**Allow**”.
7. Your new Identity Pool has been created under **US East (N. Virginia).**

**Step 4: AWS Cognito to AWS SNS integration**

1. Click on “**Edit Identity Pool”** at the top right hand side.
2. Expand “**Push Synchronization**”.
3. Click **create role.**
4. The application created under SNS will now show up here. Click the check box.
5. We now need to make the newly created role in SNS having permissions for both. 1. SNS communication 2. Auth Role for CRUD operations to the Cognito datastore.
6. Under the Authenticated role select this role from the drop down. Click on save.

**Step 5: Creating IAM users and group that are authorized to make communicate with Cognito.**

1. Click on services. Select IAM.
2. We have three operations.
   1. Modify the Role Policies.
   2. Create a user group with Policies.
   3. Add users to the user group.
3. Modify the Role Policies
4. We need to make modification to the newly created SNS Auth role so that it has permissions for both CRUD to Cognito and data transfer to SNS App.
5. Click Roles.
6. Select the Roles from the Grid.
7. Select the Tab **Trust Relationships.** Click on **Edit Trust Relationship.**
8. Add one more entry to the **Statement** option**.** The part highlighted in red is nothing but **Identity pool ID.** (Click on Edit Identity Pool and see this value.).

"Statement": [

{

"Effect": "Allow",

"Principal": {

"Service": "cognito-sync.amazonaws.com"

},

"Action": "sts:AssumeRole"

},

**{**

**"Effect": "Allow",**

**"Principal": {**

**"Federated": "cognito-identity.amazonaws.com"**

**},**

**"Action": "sts:AssumeRoleWithWebIdentity",**

**"Condition": {**

**"StringEquals": {**

**"cognito-identity.amazonaws.com:aud": "us-east-1:aea64ea2-2f5c-4c25-a9fd-26989536b5e3"**

**},**

**"ForAnyValue:StringLike": {**

**"cognito-identity.amazonaws.com:amr": "authenticated"**

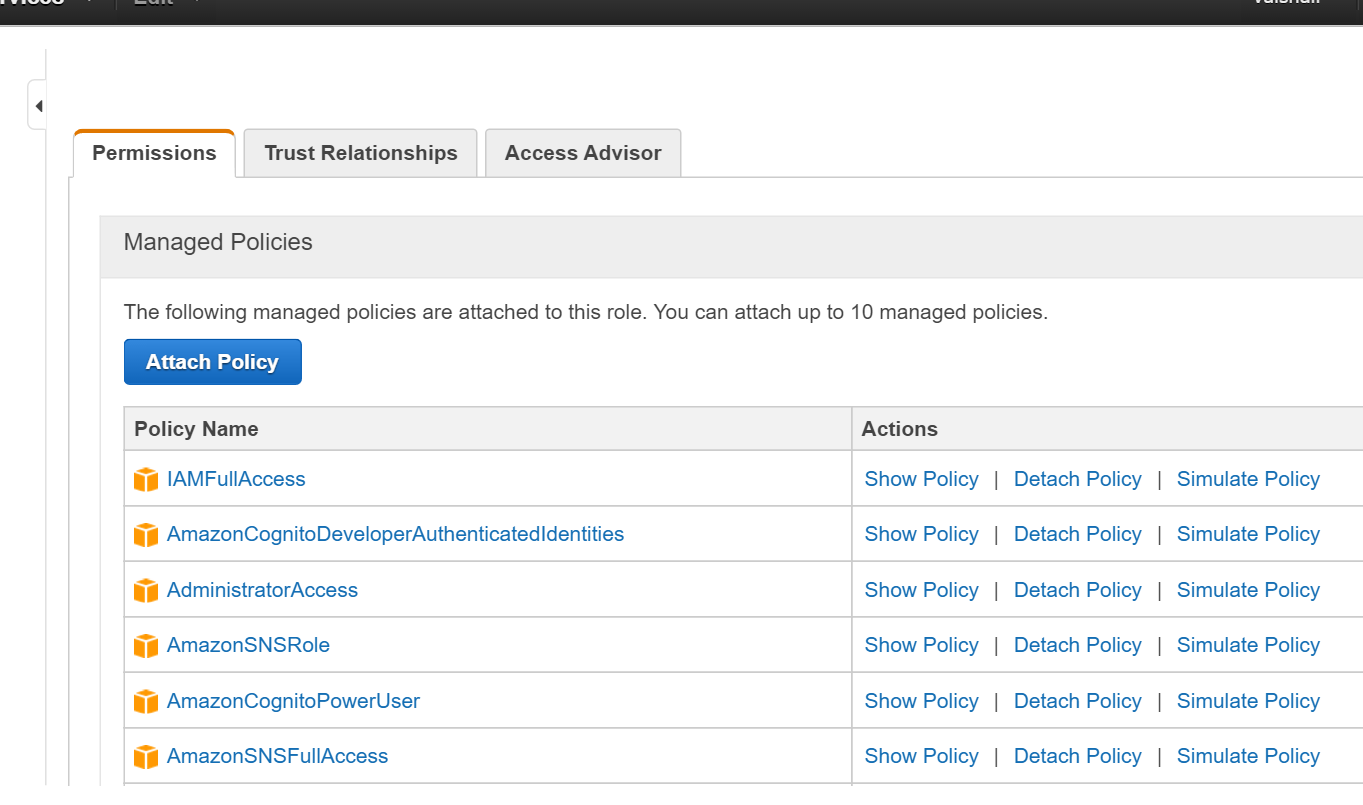
**}**

**}**

**}**

]

1. Click on save.
2. Select **Permission Tab. Click Attach Policy.** Attach the below set of policies.



1. **Create a user group with Policies.**
2. Add a Group Name. Click Next.
3. Attach the same policies to the Group as given above to the Role.
4. Click Next. Create Group.
5. **Create a user and Attach to User Group.**
6. Click on Create New Users.
7. Enter user name. Click Create.
8. **Download the credentials. These need to be entered into the application level code.**
9. Click on User Actions Dropdown. Click Add user to Group. Select the group that has been created.

**That’s IT. We are done with the Integration on the console.**

**From the Google API Project we’ll need the below values.**

1. Google **project ID** that was created.
2. **Server API key** from Google project.
3. **User’s key and secret key** that was generated while creating a new user.

**From the AWS Console.**

1. AWS Account ID.
2. AWS Pool ID of the pool that was created. ex: us-east-1:aea64ea2-2f5c-4c25-a9fd-26989536b5e3
3. Auth Role ARN from the Identity Pool. Ex: arn:aws:iam::186727481815:role/Cognito\_XYZ\_NVirginiaUnauth\_Role
4. UnAuth Role ARN from the Identity Pool. Ex: arn:aws:iam::186727481815:role/Cognito\_PushSync\_Role
5. AWS Developer provider name. From the Authenticated Providers -> Custom -> developer provider name.

Refer to the sample code that has been uploaded to GITHUB to make and test your connections.

**The flow of the code is as follows.**

1. We first register our device with GCN and obtain a registration ID.
2. We then register this device ID with AWS SNS so that the service can send “push” notifications to the register mobile device.
3. Next we set up a Cognito Provider and connect to the Cognito Identity Pool using the configuration details.
4. We then register the device with the Cognito Identity Pool.
5. The Broadcast receiver is set up to receiver any push notifications to GCM.
6. We run this code on multiple devices. Now the devices can communicate with each other via push notifications.
7. Example run would be: Device 1 -> saves data to Cognito -> Sync up with AWS SNS -> send a push notification via GCM to second registered device.