### Smart TV and AWS IOT

The smart tv backend on initialization:

1. Creates the certificate and keys, save it in the local java key store and upload the certificate to the AWS IoT platform
   1. Reason for the creation of certificates is to allow authentication of the app so that it can use the aws services
   2. For the application to upload the certificate to AWS, it requires Cognito Identity with access to the AWS IoT.
   3. Using Amazon Cognito to create a new identity pool and ensure that it has access to unauthenticated identities. It allows the application to assumed the unauthenticated role associated with this identity pool. By creating an identity pool, Cognito will setup 2 roles in IAM. The auth and unauthenticated role.
   4. Create a policy that allows iot functions (creation of certs and keys as well as policy attaching) and attach it to the unauthenticated role.
   5. The configuration we have setup up to this point will enable the App to connect to the AWS IoT platform using Cognito and upload certificates and policies.
   6. A new policy is created under AWS IoT that is attached to the device certificate uploaded earlier to AWS IoT. This policy authorizes the certificate to connect to the AWS IoT message broker and perform pub, sub and receive operations.

2. Once the certs and keys are created, the app will try to connect to the endpoint provided by the AWS IoT customer endpoint, using the pool id given by cognito and the region.

1. Once connected, in the MQTT client test found in AWS IoT, one can now publish to a topic and, on the smart tv app side, it can subscribe to the topic and receive the content push to it.