Amplify your Mobile App

Steven Smith

Mobile Practice Lead

Chariot Solutions

IoT on AWS

A Philly Cloud Computing Event

CHARIOT SOLUTIONS

What is the Amplify SDK?

A client framework for building mobile and web apps.

iOS

Android

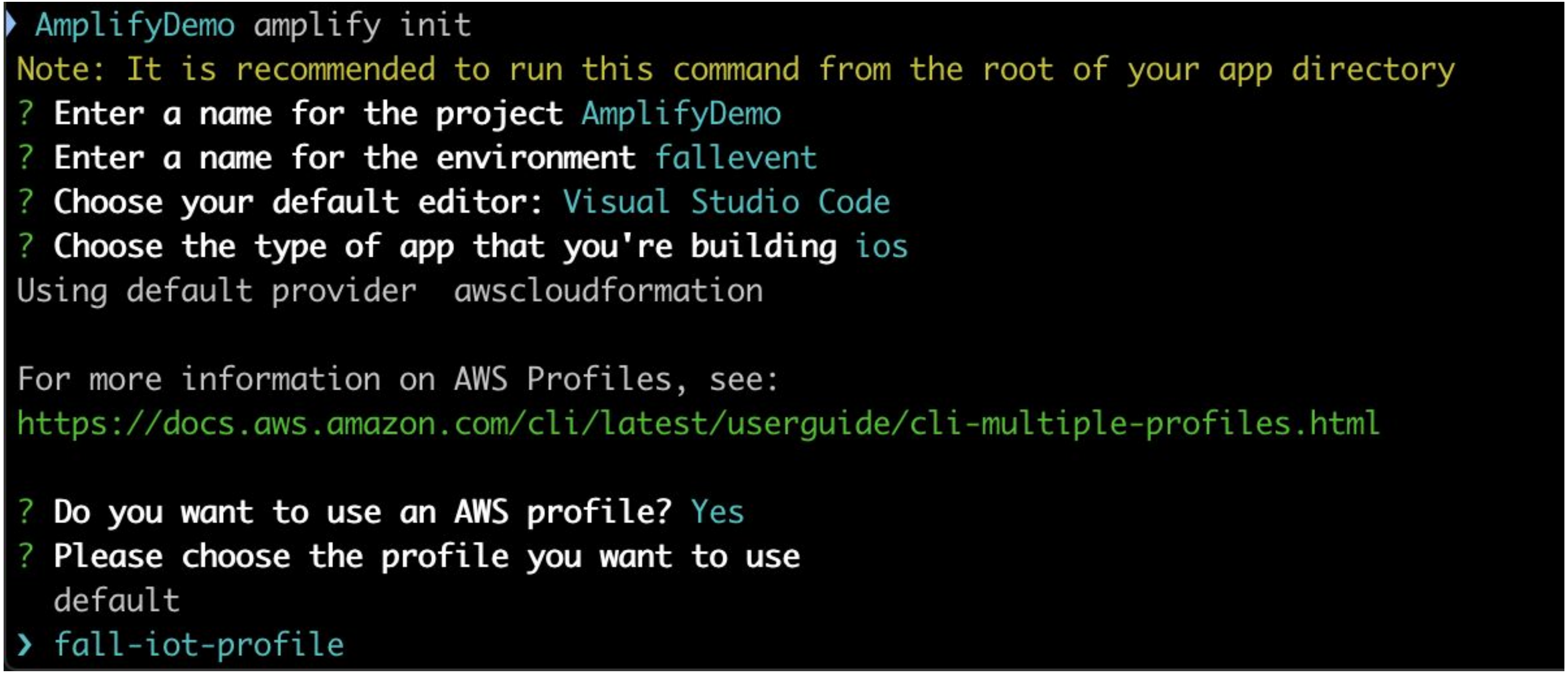
Web (JS)

React Native

How do I get started?

- Install and configure the Amplify CLI
- Create your project
- Set up your backend services using the CLI
- Add your dependencies and configuration to your App
- Code away

>amplify init

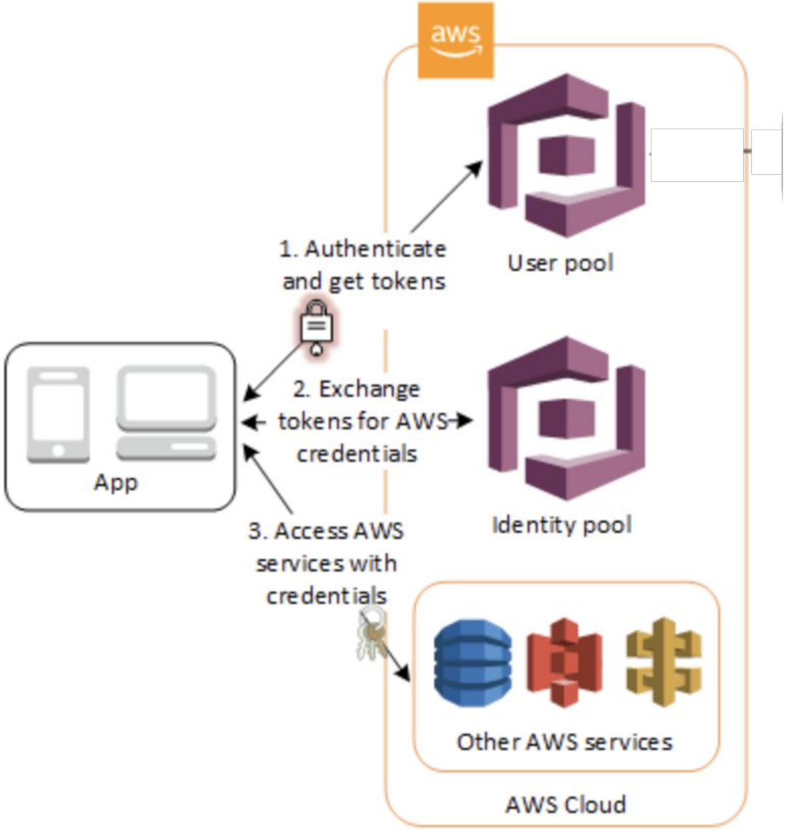


```
❯ AmplifyDemo amplify init
Note: It is recommended to run this command from the root of your app directory
? Enter a name for the project AmplifyDemo
? Enter a name for the environment fallevent
? Choose your default editor: Visual Studio Code
? Choose the type of app that you're building ios
Using default provider  awscloudformation

For more information on AWS Profiles, see:
https://docs.aws.amazon.com/cli/latest/userguide/cli-multiple-profiles.html

? Do you want to use an AWS profile? Yes
? Please choose the profile you want to use
  default
❯ fall-iot-profile
```

Accessing AWS Services


aws
User pool
1. Authenticate and get tokens
App
2. Exchange tokens for AWS credentials
Identity pool
3. Access AWS services with credentials
Other AWS services
AWS Cloud

https://docs.aws.amazon.com/en_pv/cognito/latest/developerguide/amazon-cognito-integrating-user-pools-with-identity-pools.html

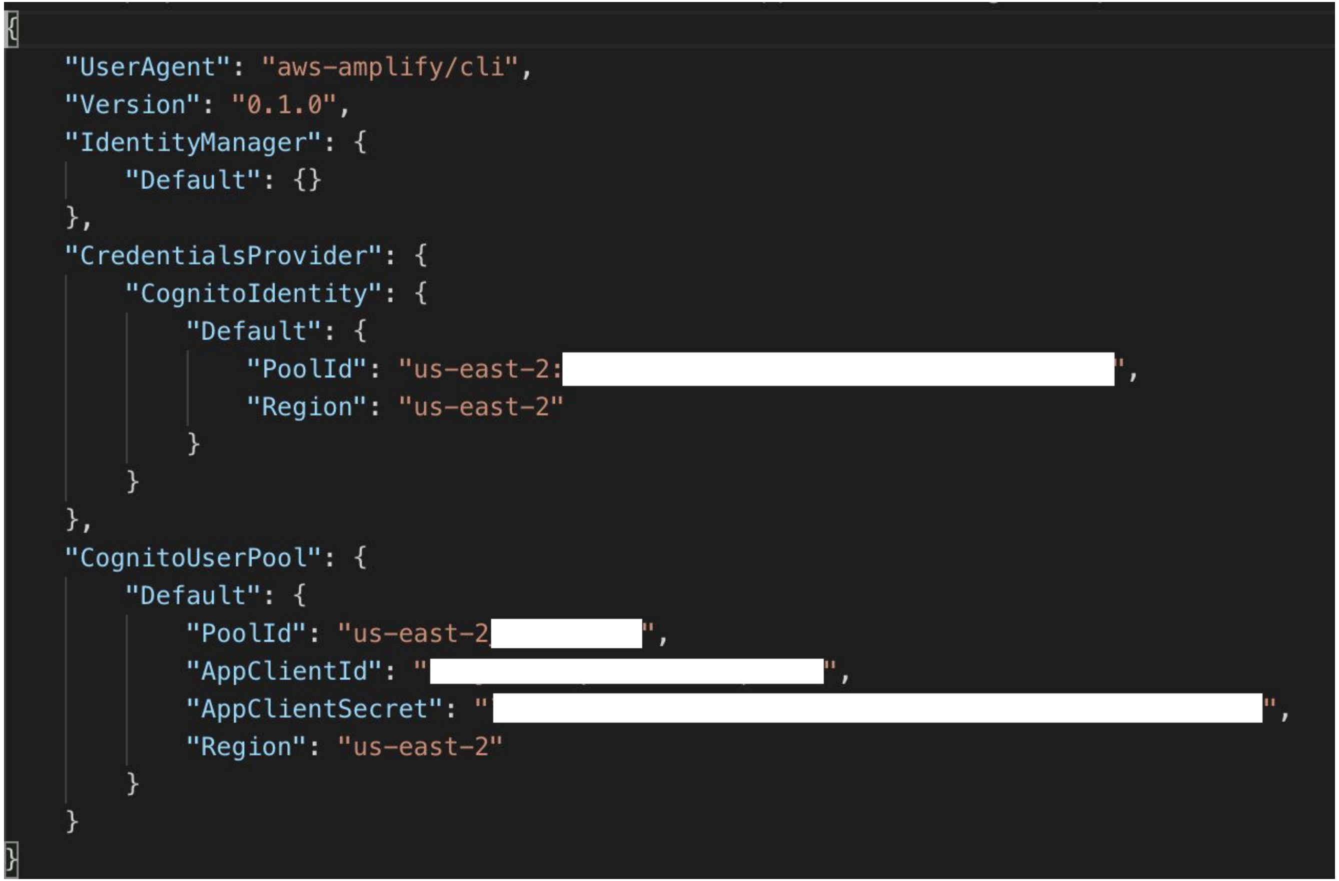
Adding Authentication

- Manual
  - User Pool
  - Client App
  - Identity Pool
  - IAM Roles
  - AWS App Config

- Amplify CLI

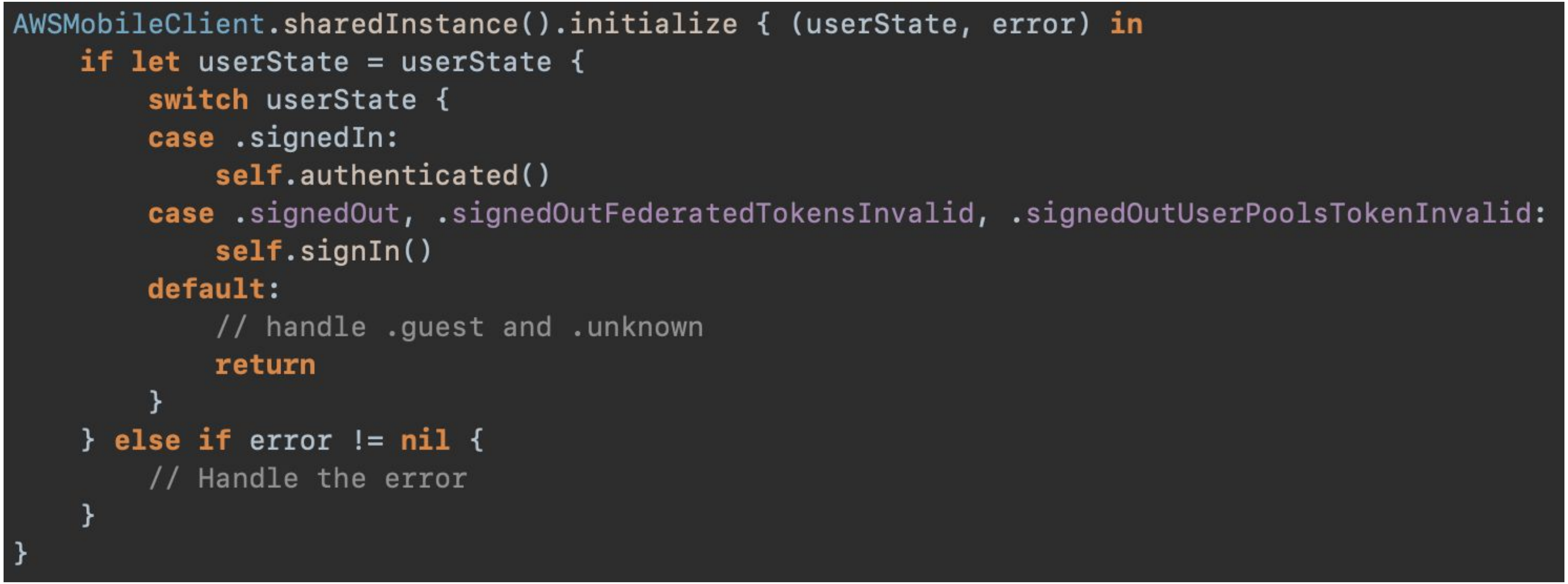
```
>amplify add auth
>amplify push
```

AWS Configuration JSON



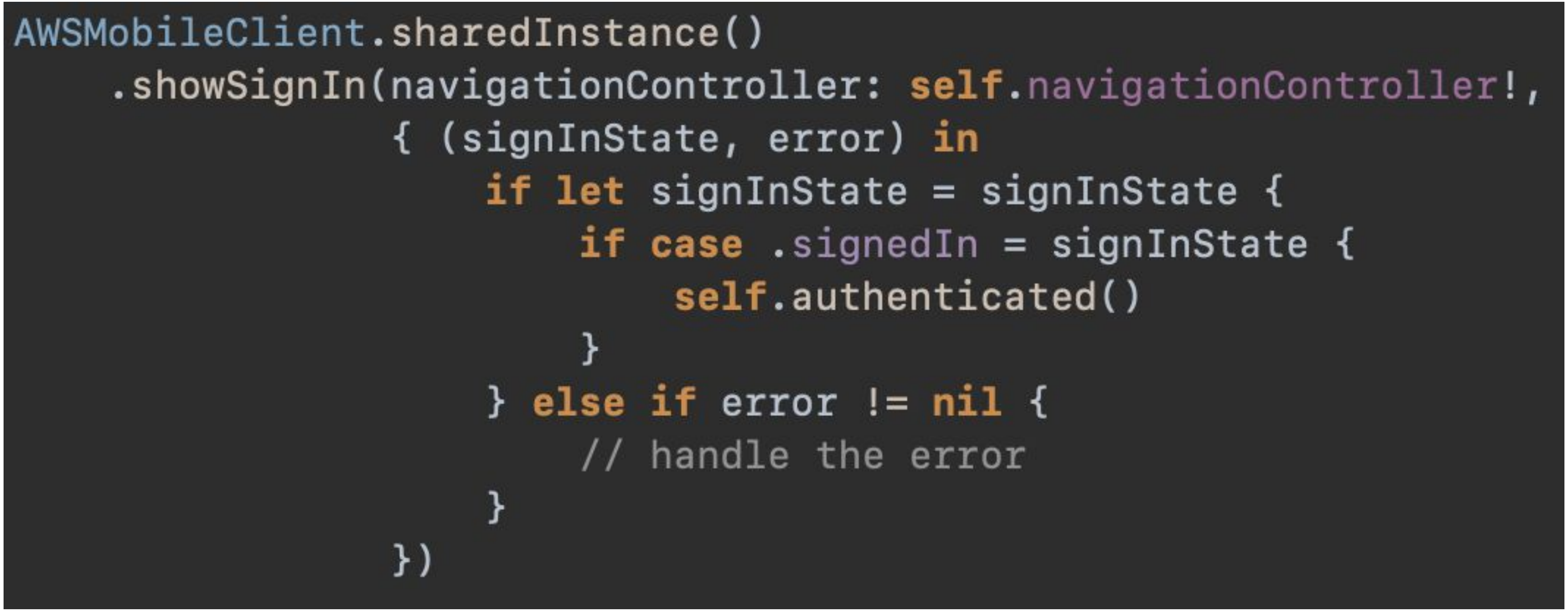
```
{
    "UserAgent": "aws-amplify/cli",
    "Version": "0.1.0",
    "IdentityManager": {
        "Default": {}
    },
    "CredentialsProvider": {
        "CognitoIdentity": {
            "Default": {
                "PoolId": "us-east-2:[redacted]",
                "Region": "us-east-2"
            }
        }
    },
    "CognitoUserPool": {
        "Default": {
            "PoolId": "us-east-2[redacted]",
            "AppClientId": "[redacted]",
            "AppClientSecret": "[redacted]",
            "Region": "us-east-2"
        }
    }
}
```

Initialization



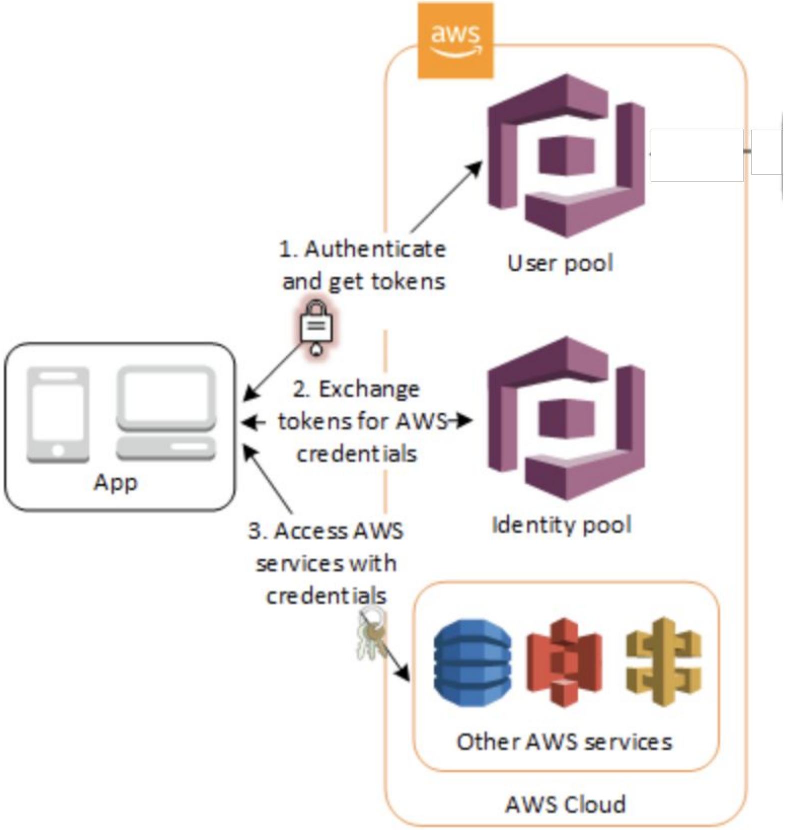
```swift
AWSMobileClient.sharedInstance().initialize { (userState, error) in
    if let userState = userState {
        switch userState {
        case .signedIn:
            self.authenticated()
        case .signedOut, .signedOutFederatedTokensInvalid, .signedOutUserPoolsTokenInvalid:
            self.signIn()
        default:
            // handle .guest and .unknown
            return
        }
    } else if error != nil {
        // Handle the error
    }
}
```

Sign Up / Sign In



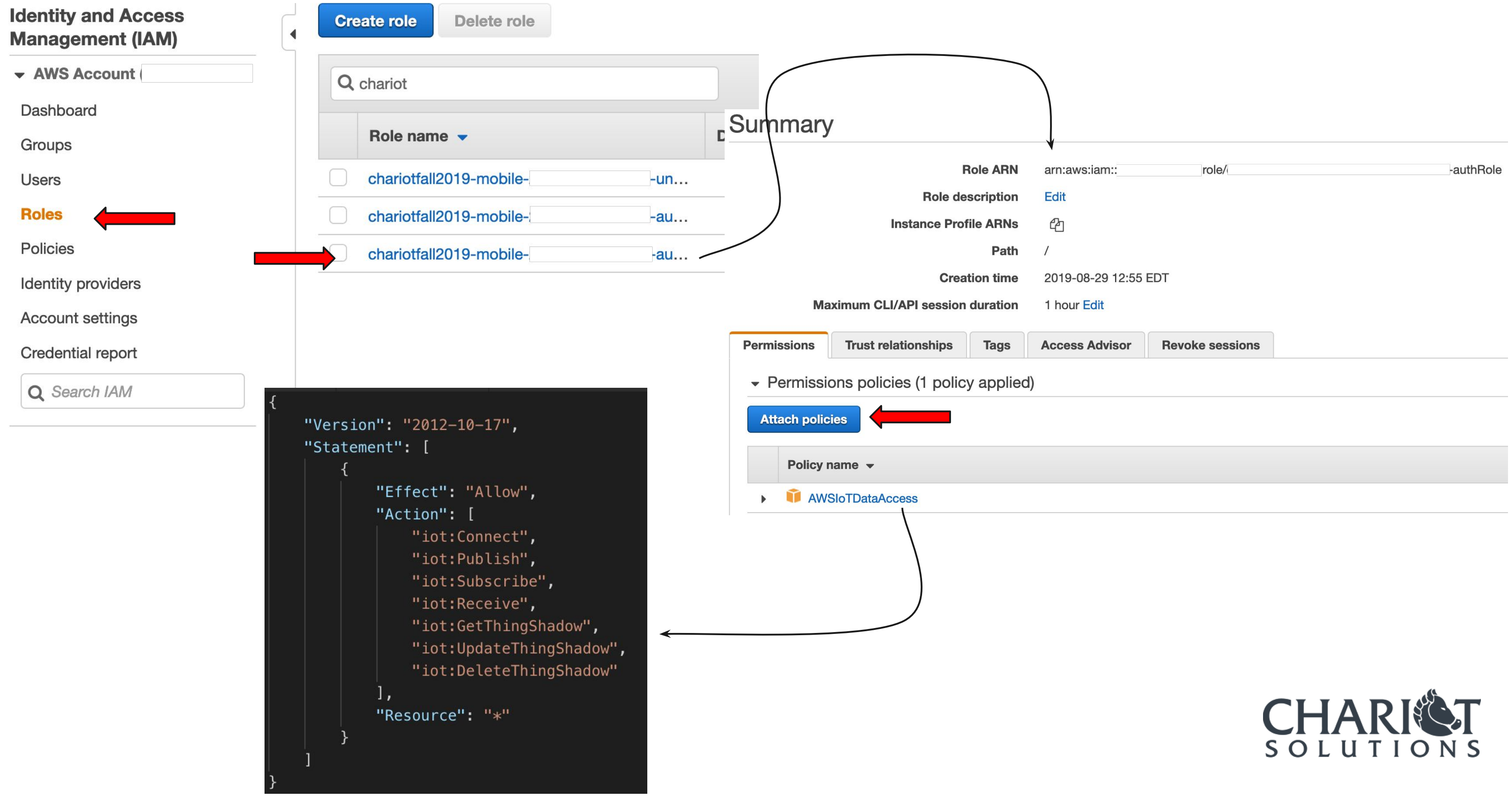
```
AWSMobileClient.sharedInstance()
    .showSignIn(navigationController: self.navigationController!,
                { (signInState, error) in
                    if let signInState = signInState {
                        if case .signedIn = signInState {
                            self.authenticated()
                        }
                    } else if error != nil {
                        // handle the error
                    }
                })
```

Accessing AWS Services


aws
User pool
1. Authenticate and get tokens
App
2. Exchange tokens for AWS credentials
Identity pool
3. Access AWS services with credentials
Other AWS services
AWS Cloud

https://docs.aws.amazon.com/en_pv/cognito/latest/developerguide/amazon-cognito-integrating-user-pools-with-identity-pools.html

Allow the IAM role access to IoT

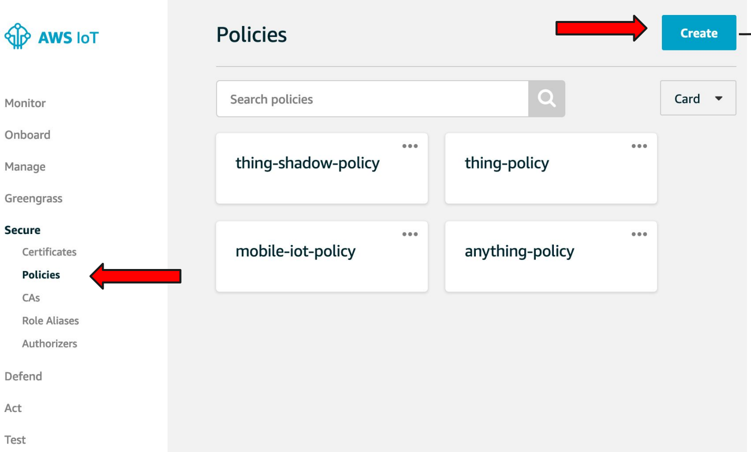


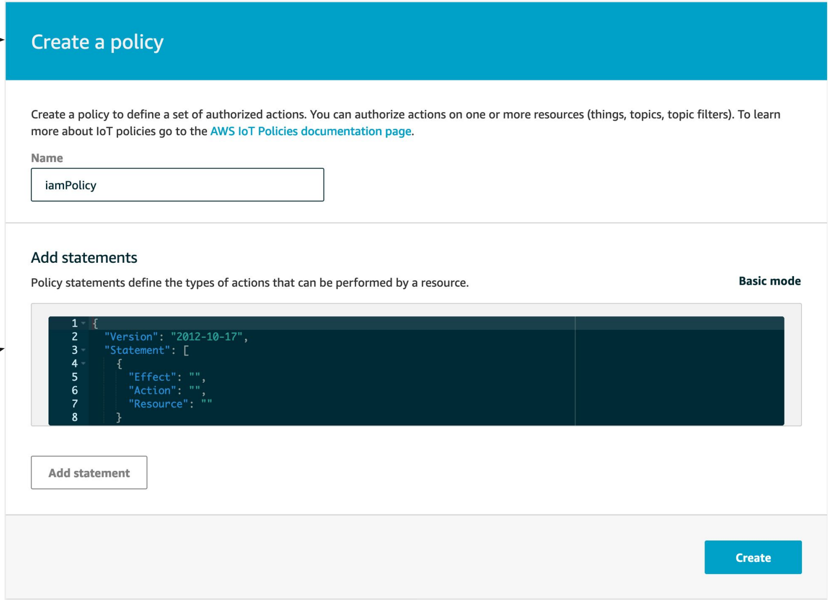
Summary

Role ARN	arn:aws:iam::[redacted]role/[redacted]-authRole
Role description	Edit
Instance Profile ARNs	
Path	/
Creation time	2019-08-29 12:55 EDT
Maximum CLI/API session duration	1 hour Edit

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect",
                "iot:Publish",
                "iot:Subscribe",
                "iot:Receive",
                "iot:GetThingShadow",
                "iot:UpdateThingShadow",
                "iot:DeleteThingShadow"
            ],
            "Resource": "*"
        }
    ]
}
```

Create the IoT Policy





Create a policy

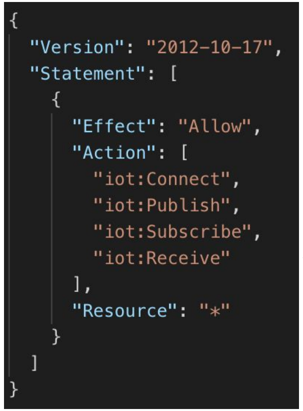
Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the AWS IoT Policies documentation page.

Name: iamPolicy

Add statements

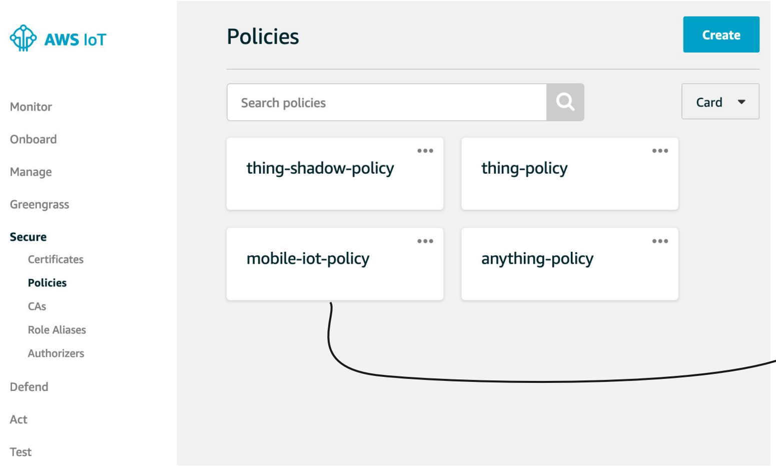
Policy statements define the types of actions that can be performed by a resource.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "",
      "Action": "",
      "Resource": ""
    }
```



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect",
        "iot:Publish",
        "iot:Subscribe",
        "iot:Receive"
      ],
      "Resource": "*"
    }
  ]
}
```

Attach the Policy to the User’s Identity



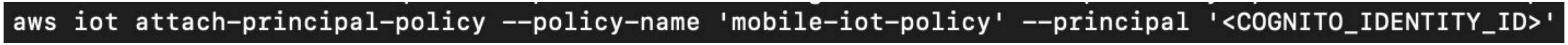
To your Cognito Identity ID

- From the SDK

  AWSMobileClient.sharedInstance().identityId

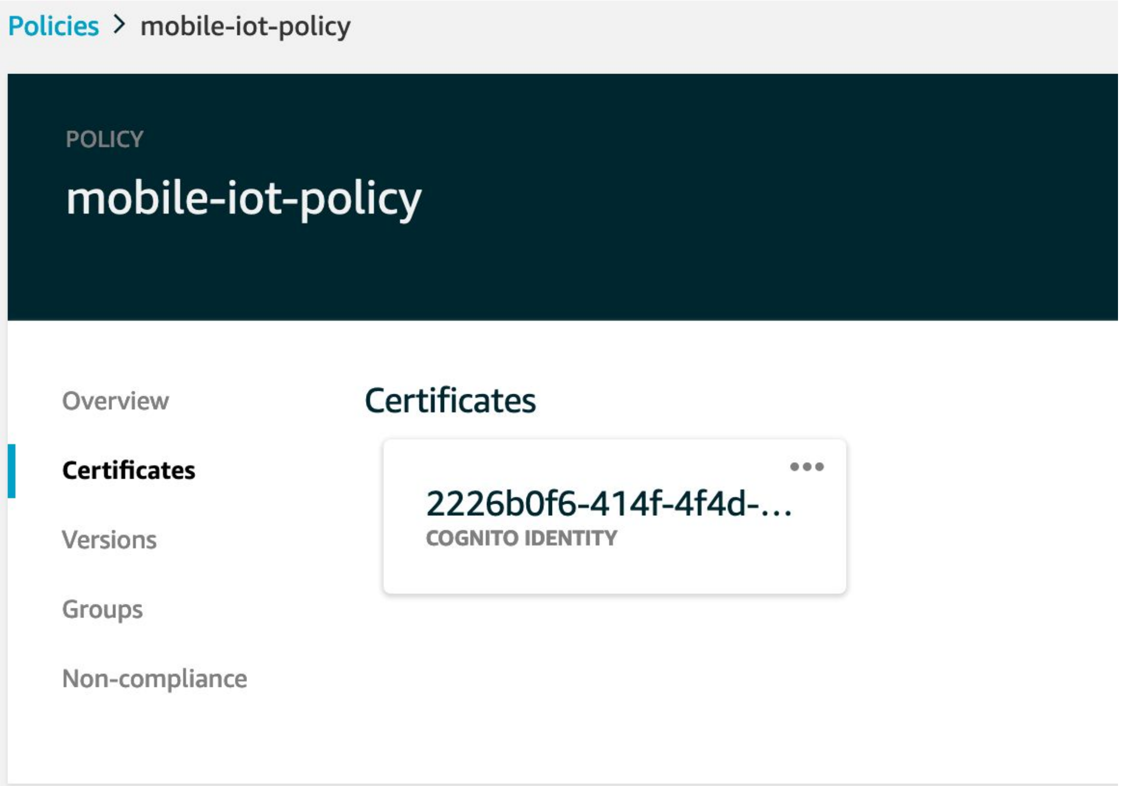
- Or Cognito Identity Pool Identity Browser

Using AWS CLI



```
aws iot attach-principal-policy --policy-name 'mobile-iot-policy' --principal '<COGNITO_IDENTITY_ID>'
```

Attached Policy



Policies > mobile-iot-policy

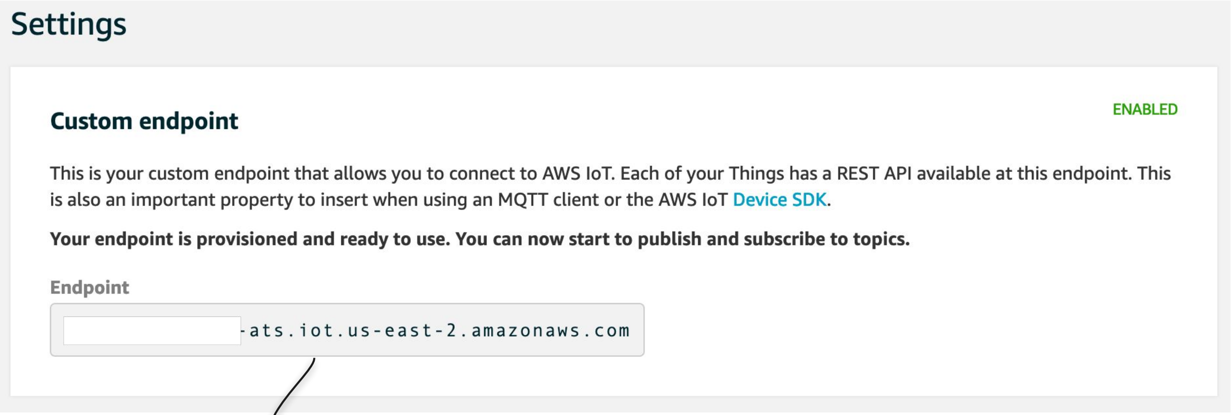
POLICY

mobile-iot-policy

Certificates

2226b0f6-414f-4f4d-…
COGNITO IDENTITY

Configure the IoT Client



Settings

Custom endpoint

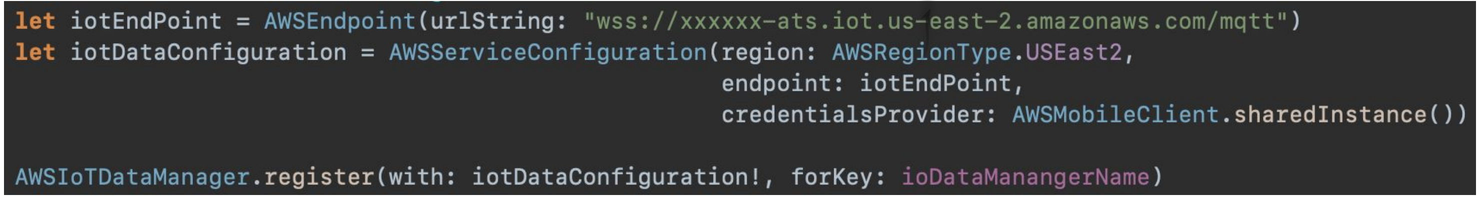
ENABLED

This is your custom endpoint that allows you to connect to AWS IoT. Each of your Things has a REST API available at this endpoint. This is also an important property to insert when using an MQTT client or the AWS IoT Device SDK.

Your endpoint is provisioned and ready to use. You can now start to publish and subscribe to topics.

Endpoint

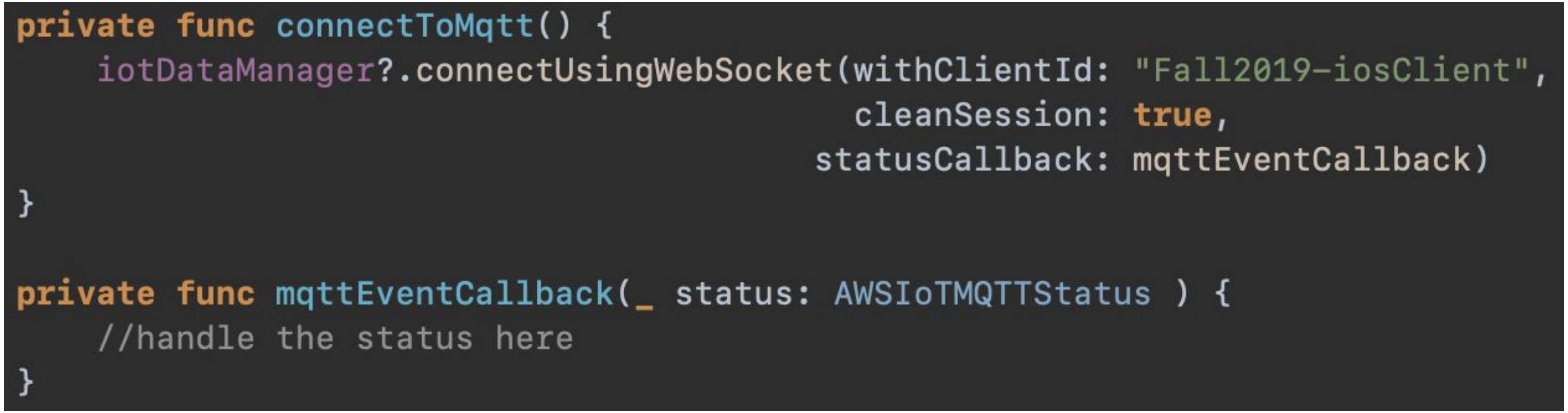
[redacted]-ats.iot.us-east-2.amazonaws.com



```
let iotEndPoint = AWSEndpoint(urlString: "wss://xxxxxx-ats.iot.us-east-2.amazonaws.com/mqtt")
let iotDataConfiguration = AWSServiceConfiguration(region: AWSRegionType.USEast2,
                                                   endpoint: iotEndPoint,
                                                   credentialsProvider: AWSMobileClient.sharedInstance())

AWSIoTDataManager.register(with: iotDataConfiguration!, forKey: ioDataManangerName)
```

Connect

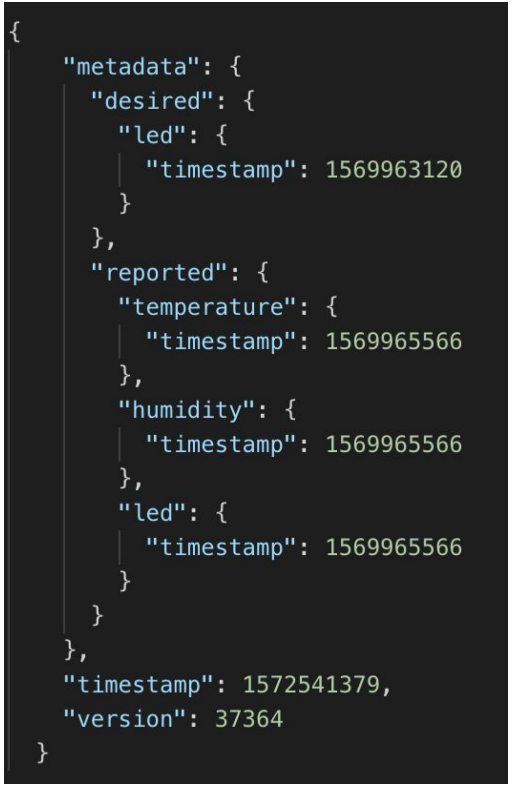


```
private func connectToMqtt() {
    iotDataManager?.connectUsingWebSocket(withClientId: "Fall2019-iosClient",
                                          cleanSession: true,
                                        statusCallback: mqttEventCallback)
}

private func mqttEventCallback(_ status: AWSIoTMQTTStatus ) {
    //handle the status here
}
```

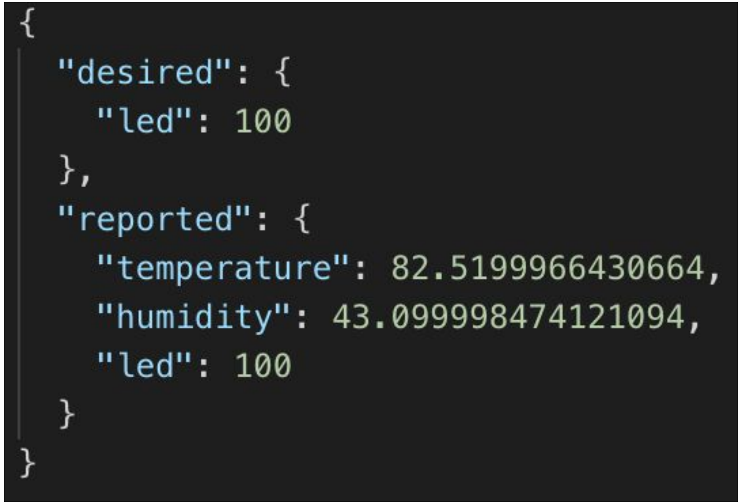
What is a device shadow and why?

Shadow Metadata:



```
{
    "metadata": {
      "desired": {
        "led": {
          "timestamp": 1569963120
        }
      },
      "reported": {
        "temperature": {
          "timestamp": 1569965566
        },
        "humidity": {
          "timestamp": 1569965566
        },
        "led": {
          "timestamp": 1569965566
        }
      }
    },
    "timestamp": 1572541379,
    "version": 37364
  }
```

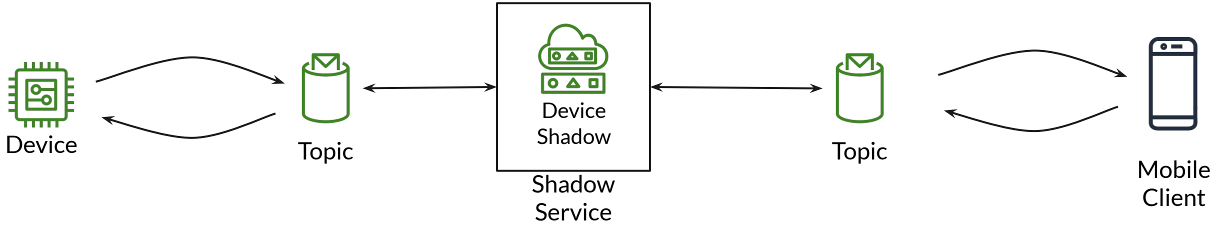
Shadow State:



```
{
  "desired": {
    "led": 100
  },
  "reported": {
    "temperature": 82.5199966430664,
    "humidity": 43.099998474121094,
    "led": 100
  }
}
```

CHARIOT SOLUTIONS

Device Shadows



← $aws/things/<thing name>/shadow/update

→ $aws/things/<thing name>/shadow/update/delta

Error Handling Topics

→ …./accepted

→ …./rejected

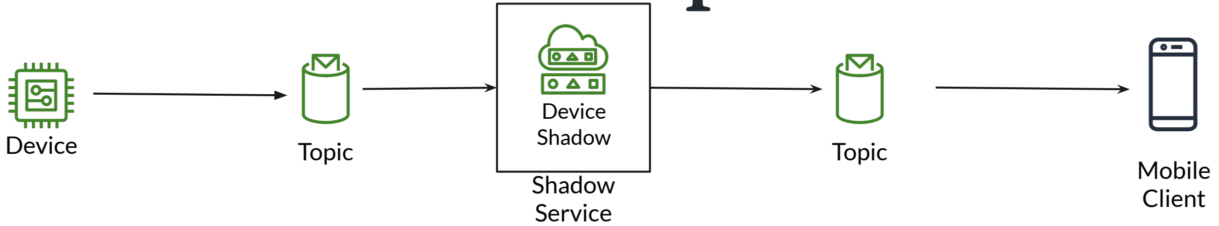
← $aws/things/<thing name>/shadow/get

← $aws/things/<thing name>/shadow/update

→ $aws/things/<thingname>/shadow/update/documents

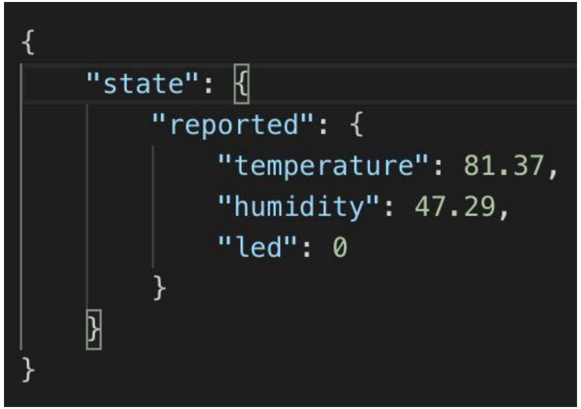
→ $aws/things/<thing name>/shadow/get/accepted

Device Shadow State Update

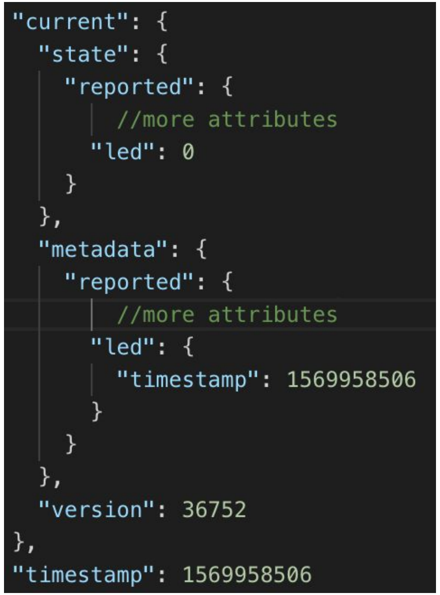


← $aws/things/<thing name>/shadow/update

→ $aws/things/<thingname>/shadow/update/documents

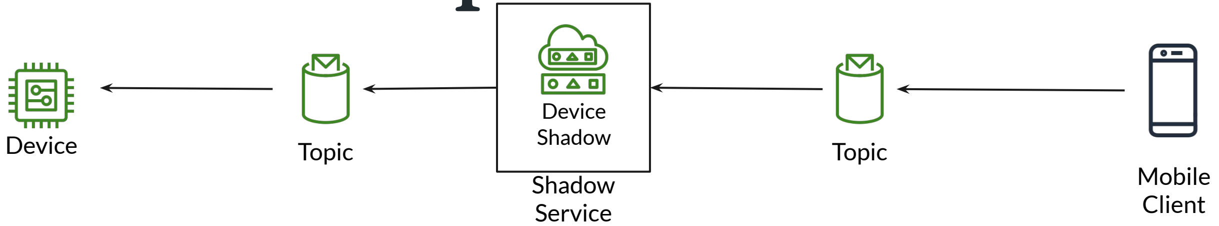


```
{
    "state": {
        "reported": {
            "temperature": 81.37,
            "humidity": 47.29,
            "led": 0
        }
    }
}
```



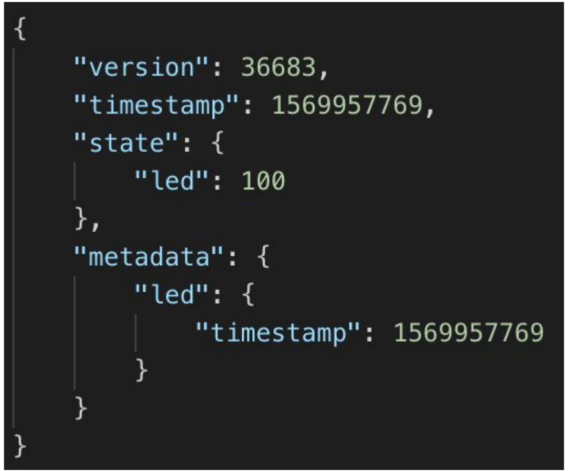
```
"current": {
  "state": {
    "reported": {
      //more attributes
      "led": 0
    }
  },
  "metadata": {
    "reported": {
      //more attributes
      "led": {
        "timestamp": 1569958506
      }
    }
  },
  "version": 36752
},
"timestamp": 1569958506
```

Device Shadow Update Device

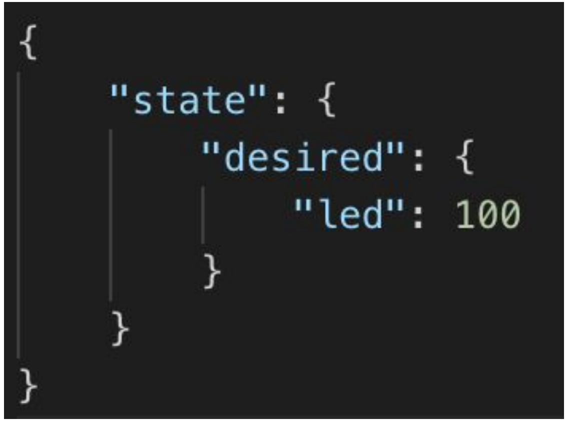


→ $aws/things/<thingname>/shadow/update/delta

← $aws/things/<thingname>/shadow/update/

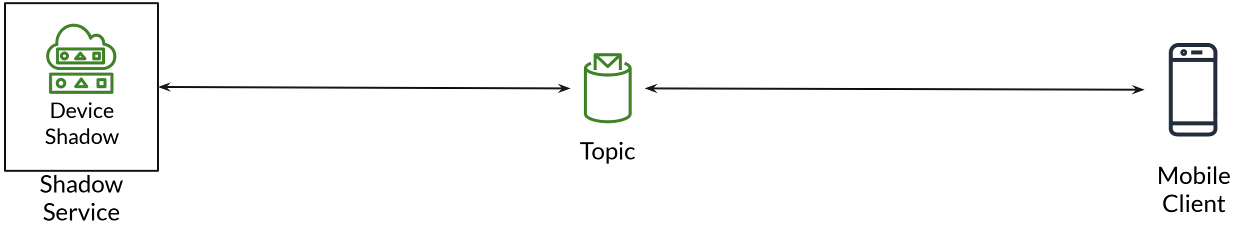


```
{
    "version": 36683,
    "timestamp": 1569957769,
    "state": {
        "led": 100
    },
    "metadata": {
        "led": {
            "timestamp": 1569957769
        }
    }
}
```



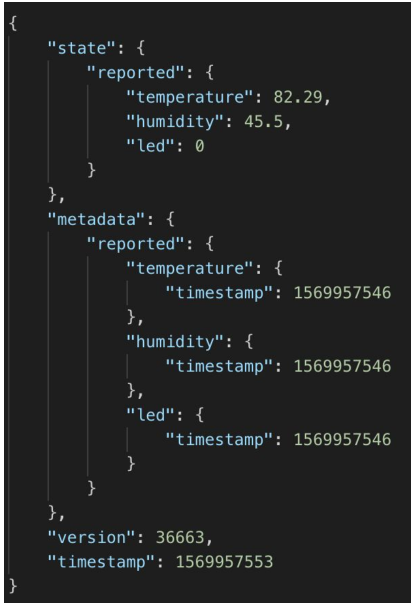
```
{
    "state": {
        "desired": {
            "led": 100
        }
    }
}
```

Device Shadow Get Device State



→ $aws/things/<thingname>/shadow/get/accepted

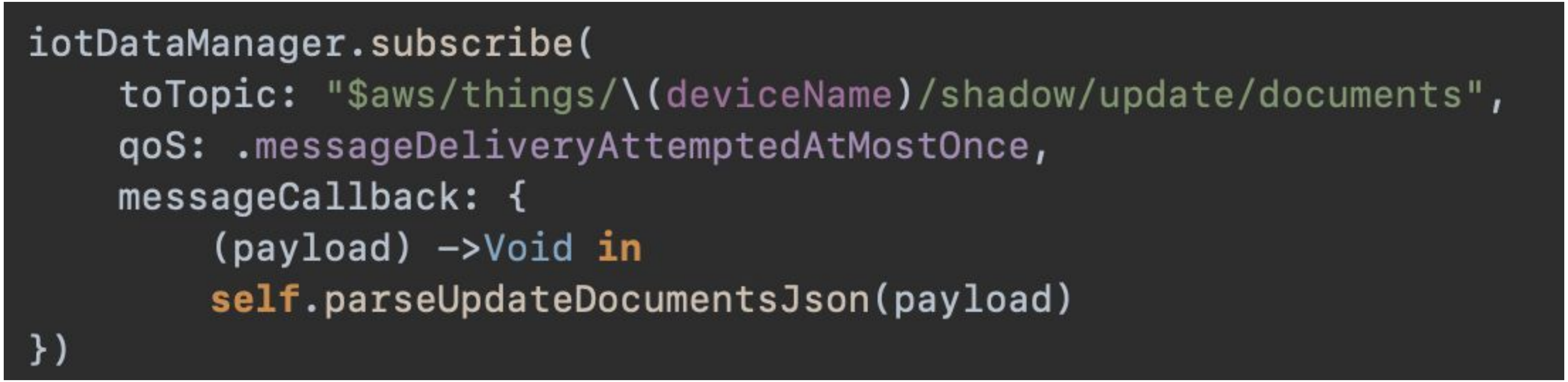
← $aws/things/<thingname>/shadow/get/



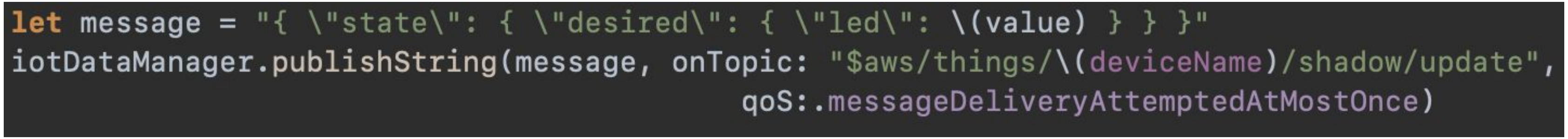
```
{
    "state": {
        "reported": {
            "temperature": 82.29,
            "humidity": 45.5,
            "led": 0
        }
    },
    "metadata": {
        "reported": {
            "temperature": {
                "timestamp": 1569957546
            },
            "humidity": {
                "timestamp": 1569957546
            },
            "led": {
                "timestamp": 1569957546
            }
        }
    },
    "version": 36663,
    "timestamp": 1569957553
}
```

<empty message>

Subscribe, Publish



```
iotDataManager.subscribe(
    toTopic: "$aws/things/\(deviceName)/shadow/update/documents",
    qoS: .messageDeliveryAttemptedAtMostOnce,
    messageCallback: {
        (payload) ->Void in
        self.parseUpdateDocumentsJson(payload)
})
```



```
let message = "{ \"state\": { \"desired\": { \"led\": \(value) } } }"
iotDataManager.publishString(message, onTopic: "$aws/things/\(deviceName)/shadow/update",
                                          qoS:.messageDeliveryAttemptedAtMostOnce)
```

References

AWS Amplify

https://aws-amplify.github.io/docs/

Device Shadow Topics

https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-mqtt.html

Accessing AWS Services with an Identity Pool

https://docs.aws.amazon.com/en_pv/cognito/latest/developerguide/amazon-cognito-integrating-user-pools-with-identity-pools.html

AWS IoT Policies

https://docs.aws.amazon.com/iot/latest/developerguide/iot-policies.html

Sample Code - Git Repo

https://github.com/chariotsolutions/chariot-fall2019-iot-shadows

Technology in the Service of Business.

Chariot Solutions is the Greater Philadelphia region's top IT consulting firm specializing in software development, systems integration, mobile application development and training.

Our team includes many of the top software architects in the area, with deep technical expertise, industry knowledge and a genuine passion for software development.

Visit us online at **chariotsolutions.com**.

CHARIOT
SOLUTIONS