# Amazon Mobile Analytics

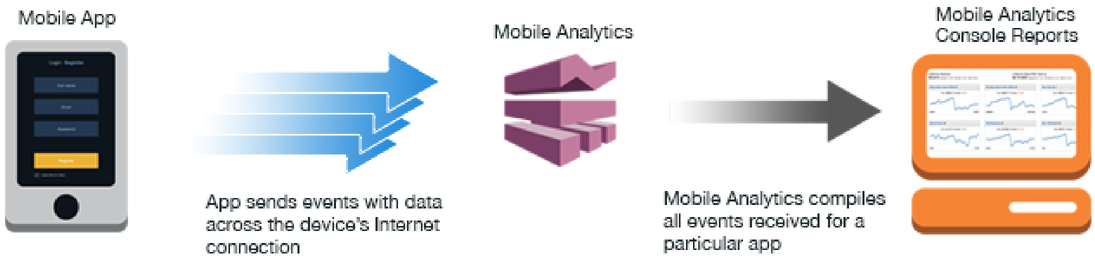
## User Guide

# Amazon Mobile Analytics: User Guide

# Table of Contents

# What Is Amazon Mobile Analytics?

Amazon Mobile Analytics is a service for collecting, visualizing, understanding, and extracting app usage data at scale. Mobile Analytics easily captures both standard device data and custom events and automatically calculates reports on your behalf.


Mobile App
App sends events with data across the device's Internet connection
Mobile Analytics
Mobile Analytics compiles all events received for a particular app
Mobile Analytics Console Reports

The following reports on key performance indicators are provided in the Mobile Analytics console:

- Daily Active Users (DAU), Monthly Active Users (MAU), and New Users
- Sticky Factor (DAU divided by MAU)
- Session Count and Average Sessions per Daily Active User
- Average Revenue per Daily Active User (ARPDAU) and Average Revenue per Paid Daily Active User (ARPPDAU)
- Day 1, 3, and 7 Retention and Week 1, 2, and 3 Retention
- Custom Events

In addition to these aggregated reports, you can also set up your data to be exported automatically to Amazon Redshift and Amazon S3 for further analysis. For more information, see Auto Export.

These reports are provided through six reporting tabs in the console:

- **Overview** – Track nine preselected reports in a simple-to-review dashboard to get a quick idea of engagement: MAU, DAU, New Users, Daily Sessions, Sticky Factor, 1-Day Retention, ARPDAU, Daily Paying Users, ARPPDAU.
- **Active Users** – Track how many users engage with your app daily and monthly and monitor its stickiness to gauge engagement, appeal, and monetization.

- **Sessions** – Track how often your app is used on a given day and how often each user opens your app during a day.
- **Retention** – Track the rate at which customers come back to your app on a daily and weekly basis.
- **Revenue** – Track in-app revenue trends to identify areas for monetization improvement.
- **Custom events** – Track custom, defined user actions specific to your app.

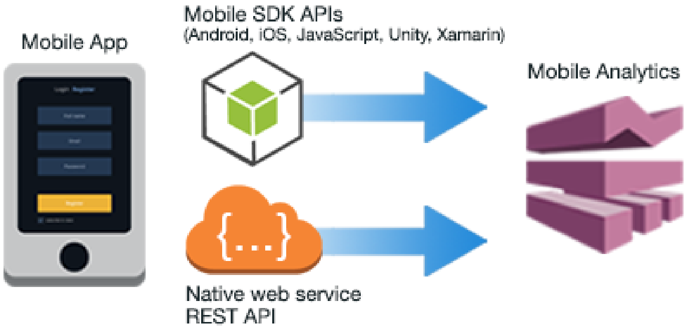
# Incorporating Mobile Analytics

Mobile Analytics can be used by developers to capture information, through analytics events created in the application logic, about how their applications are used. Individual bits of data called *attributes* and *metrics* are added to an event created in the application.

After attributes or metrics have been added, the event is then recorded so a copy of the data is preserved in the device's file storage. When an application is sent to the background by the user, recorded events are then submitted to the Mobile Analytics service; this data contributes to the reports available in the console.

There are two ways to incorporate Mobile Analytics into the code of your app:

- The Mobile Analytics web service provides a REST API your app can invoke using HTTP PUT requests. The REST API is platform- and language-independent; it requires a properly formed HTTP request only.
- For mobile app developers, the AWS Mobile SDK provides APIs that enable easier and more robust interaction with Mobile Analytics for apps running on Android, iOS, JavaScript, Unity, or Xamarin.



# Using the Mobile SDK

If you want to use Mobile Analytics in an Android, iOS, JavaScript, Unity, or Xamarin application, you will probably want to make API calls through the AWS Mobile SDK. For information about working with Mobile Analytics using the AWS Mobile SDK, see:

- Android Developer Guide
- iOS Developer Guide
- Mobile Analytics SDK for JavaScript
- Unity Developer Guide
- Xamarin Developer Guide

# Mobile Analytics Pricing

With AWS services, you pay only for what you use, with no contracts or monthly active user charges.

Mobile Analytics is free up to 100 million events per month, and costs just $1.00 per million events above the free tier. If you enable the Auto Export (p. 89) options to Amazon Simple Storage Service (Amazon S3) or Amazon Redshift (Amazon Redshift), there may be additional charges for the use of those services.

> **Important**
>
> If you use the Mobile SDK, Mobile Analytics automatically generates and submits two system events per session in addition to custom events you may create and submit. These events are required to enable collection of data compiled into the standard console charts on key performance indicators.
>
> If you use the REST API, you submit these system events (`_system.start` and `_system.stop`) manually. For more information, see Managing Sessions (p. 33).
>
> Whether submitted automatically by the Mobile SDK or manually using the REST API, these system events are included in the count of total events submitted and therefore count toward the 100 million free events each month.

# Getting Started

Before you can visualize the data in the console at https://console.aws.amazon.com/mobileanalytics/home, sign up for an AWS account, create a Cognito identity pool or an AWS Identity and Access Management (IAM) account, and then download the AWS Mobile SDK so that you can integrate it into your app.

Topics

## Step 1: Sign Up for an AWS Account

If you already have an AWS account, sign in to your account and go to the next step.

If you do not have an AWS account, use the following procedure to create one.

### To sign up for AWS

1. Open http://aws.amazon.com/ and choose **Create an AWS Account**.
2. Follow the online instructions.

## Step 2: Follow the Steps to Add an App

1. Open the Amazon Mobile Analytics console at https://console.aws.amazon.com/mobileanalytics/home.
2. If this is the first time you are using Amazon Mobile Analytics, follow the on-screen instructions to add an app.

   If you have already added an app, select the settings icon ⚙.
3. In **Manage Apps**, choose **Add App**.
4. Follow the instructions.

# Step 3: Integrate Mobile Analytics in to Your App

To integrate Mobile Analytics, use the SDK for platforms appropriate to your app:

- Using Mobile Analytics in an Android app
- Using Mobile Analytics in an iOS app
- Using Mobile Analytics in a JavaScript app
- Using Mobile Analytics in a Unity app
- Using Mobile Analytics in an Xamarin app

# Step 4: Review the Data in Amazon Mobile Analytics Reports

You can see your data in the console at https://console.aws.amazon.com/mobileanalytics/home for events reported on iOS and Android platforms. It can take up to an hour for data sent to Mobile Analytics to appear in the console reports.

# Adding and Managing Apps

The steps you follow to create an app that uses Mobile Analytics will depend on the platforms you target and the tools and programming languages associated with each. But generally speaking, to use Mobile Analytics, you must do two things:

- Add and set up your app in the console at https://console.aws.amazon.com/mobileanalytics/home.
- Add the code to your app that accesses the Mobile Analytics service as determined by the platforms your app will support.

## Accessing Mobile Analytics

In order for your app to directly access AWS it must have credentials, which you obtain through Amazon Cognito Identity. An Amazon Cognito identity pool defines user identities used with your account.

When you create your first app, the console creates a Cognito identity pool you can use to authenticate your apps with Mobile Analytics. The console also provides, for each platform, integration code snippets that include the Cognito identity pool ID your apps need to communicate with Mobile Analytics. If you select a different Cognito identity pool to use in your apps, these code snippets will be updated to reflect it.

Topics

## Adding an App to Mobile Analytics

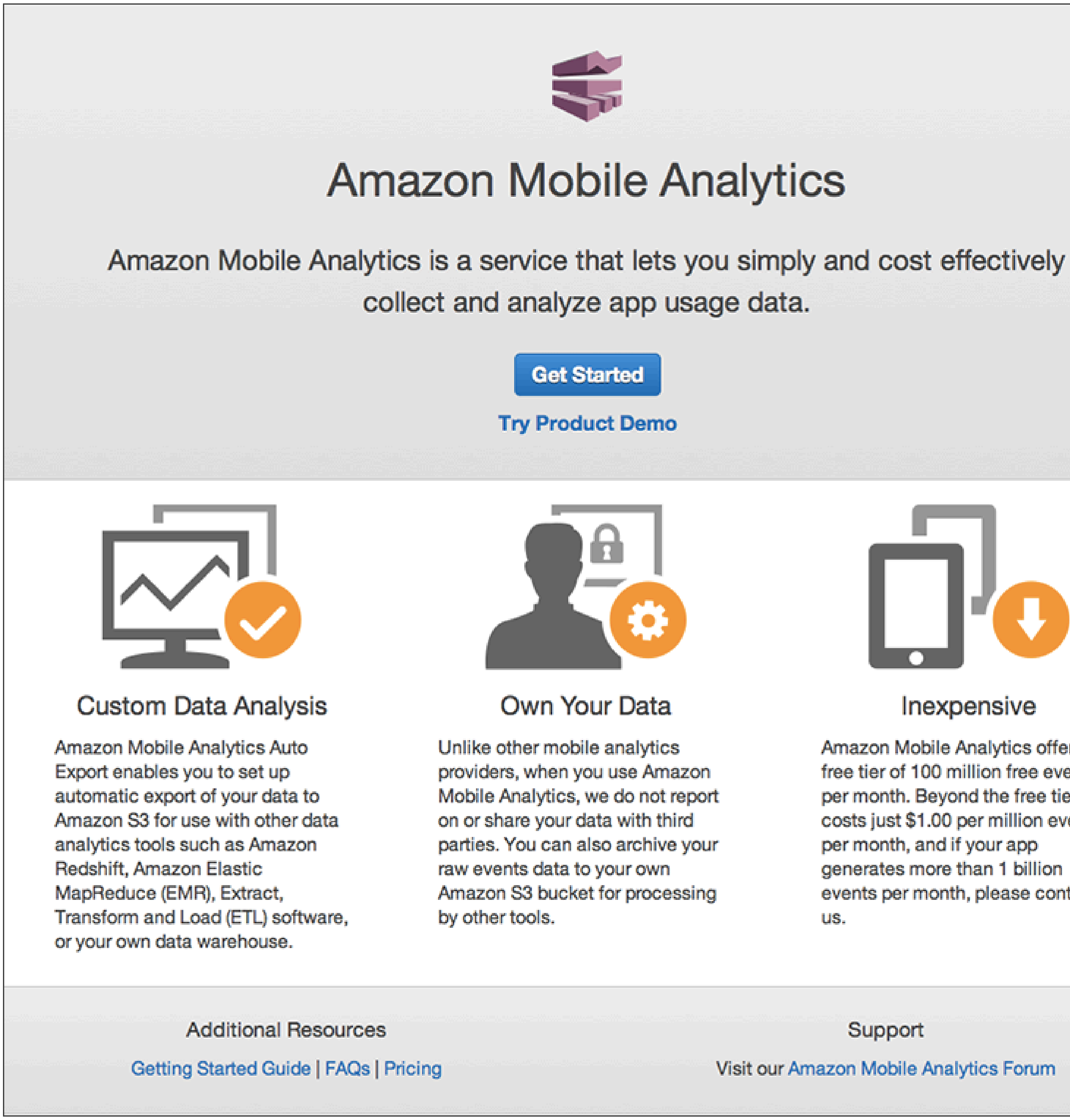
To use Mobile Analytics with an app, you must use the console to add the app to the Mobile Analytics service. The steps to add your first app to Mobile Analytics are different from the steps to add additional apps.

### If You Have No Existing Identity Pools

Use the following procedure if you are adding your first app in the Mobile Analytics console but do not yet have any Cognito identity pools defined.

**To add an app for the first time**

1. Open the Amazon Mobile Analytics console at https://console.aws.amazon.com/mobileanalytics/home.



Amazon Mobile Analytics

Amazon Mobile Analytics is a service that lets you simply and cost effectively collect and analyze app usage data.

Custom Data Analysis

Amazon Mobile Analytics Auto Export enables you to set up automatic export of your data to Amazon S3 for use with other data analytics tools such as Amazon Redshift, Amazon Elastic MapReduce (EMR), Extract, Transform and Load (ETL) software, or your own data warehouse.

Own Your Data

Unlike other mobile analytics providers, when you use Amazon Mobile Analytics, we do not report on or share your data with third parties. You can also archive your raw events data to your own Amazon S3 bucket for processing by other tools.

Inexpensive

Amazon Mobile Analytics offe free tier of 100 million free eve per month. Beyond the free tie costs just $1.00 per million eve per month, and if your app generates more than 1 billion events per month, please cont us.

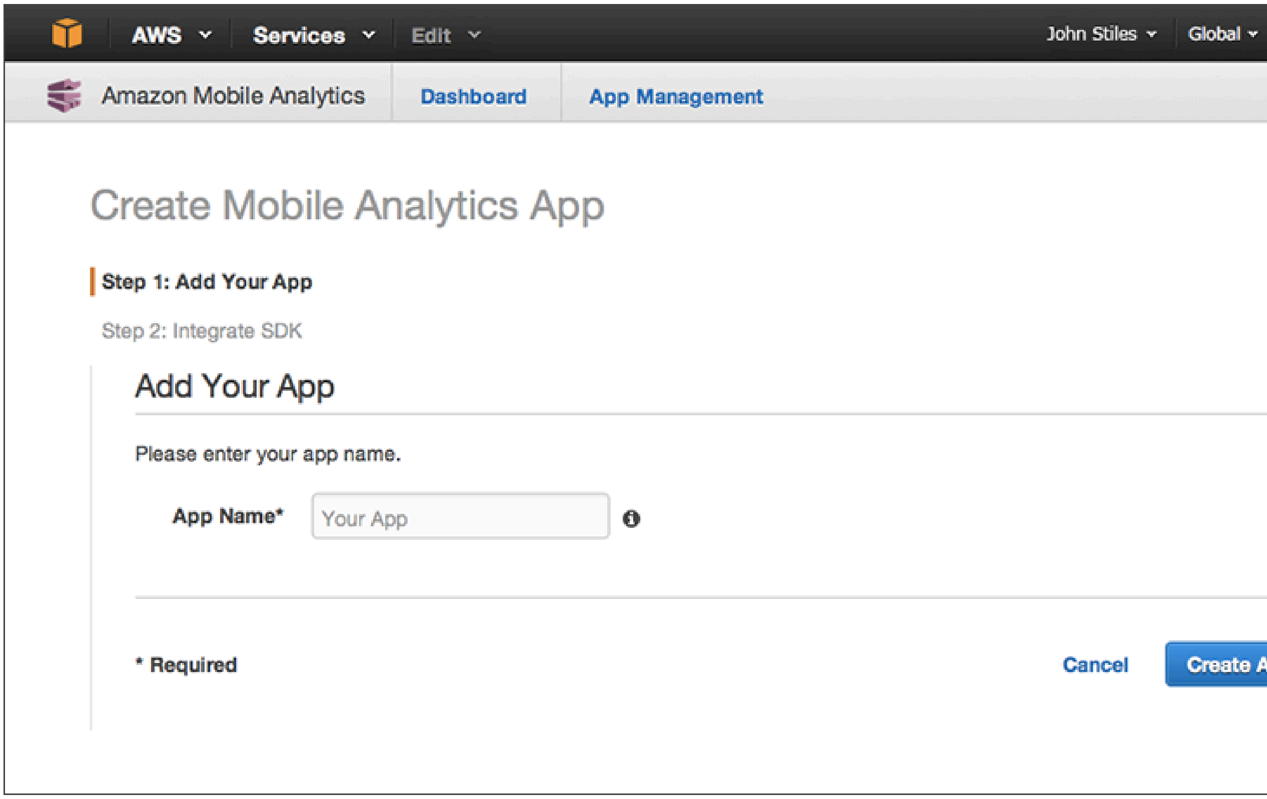
Additional Resources

Getting Started Guide | FAQs | Pricing

Support

Visit our Amazon Mobile Analytics Forum

2. Choose **Get Started**.
3. In **App Name**, type a name for your app, and then choose **Create App**.



Create Mobile Analytics App

Step 1: Add Your App

Step 2: Integrate SDK

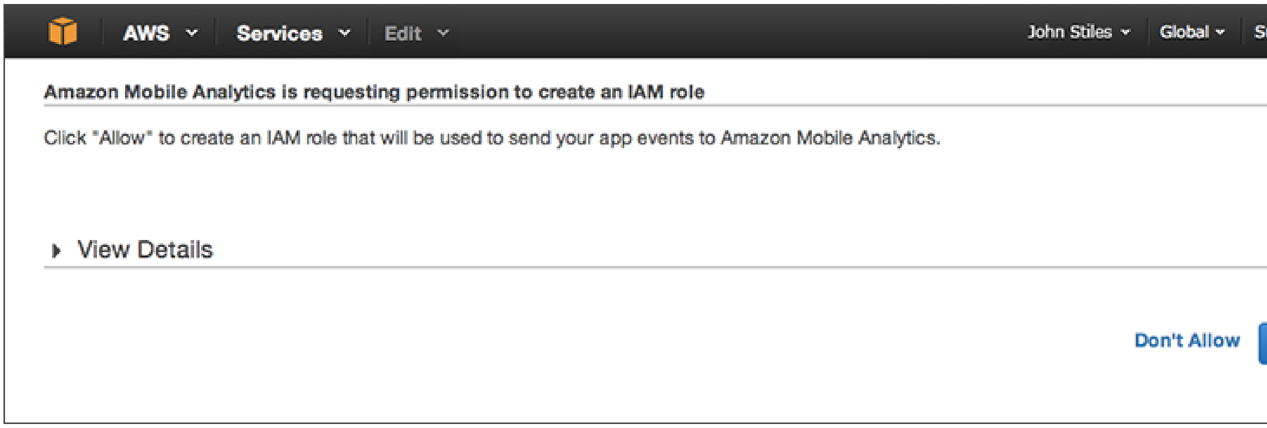
Add Your App

Please enter your app name.

App Name* Your App

* Required

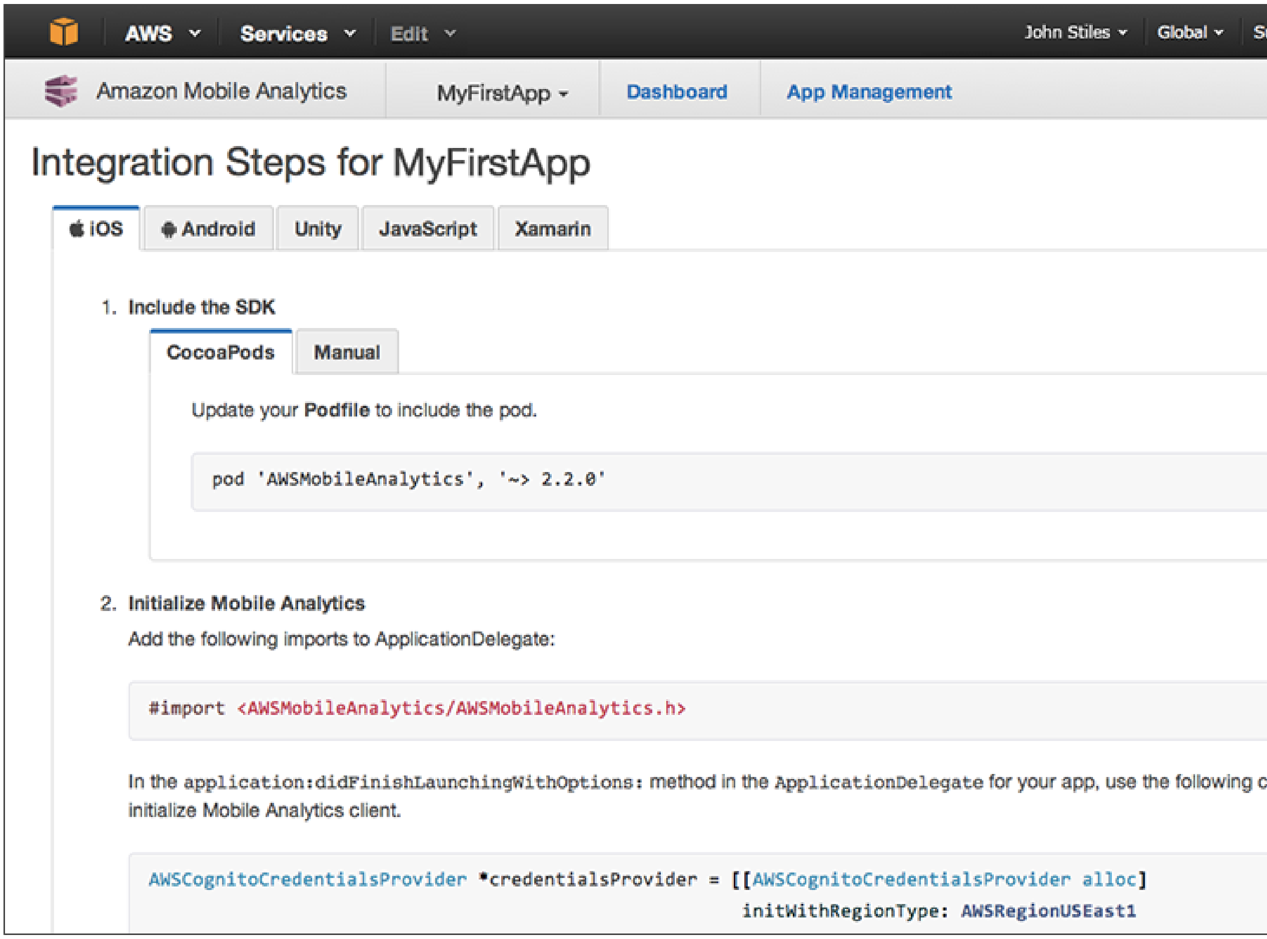
4. In the IAM role creation wizard, choose **Allow** to generate an IAM role for the Cognito identity pool.



Amazon Mobile Analytics is requesting permission to create an IAM role

Click "Allow" to create an IAM role that will be used to send your app events to Amazon Mobile Analytics.

5. In the displayed integration steps, choose the tab that corresponds to the platform targeted by your app. The Cognito identity pool ID generated by the wizard in the previous step appears in the sample code. You can easily copy and paste the code into your app.



Integration Steps for MyFirstApp

1. Include the SDK

Update your **Podfile** to include the pod.

```
pod 'AWSMobileAnalytics', '~> 2.2.0'
```

2. Initialize Mobile Analytics

Add the following imports to ApplicationDelegate:

```
#import <AWSMobileAnalytics/AWSMobileAnalytics.h>
```

In the `application:didFinishLaunchingWithOptions:` method in the `ApplicationDelegate` for your app, use the following c
initialize Mobile Analytics client.

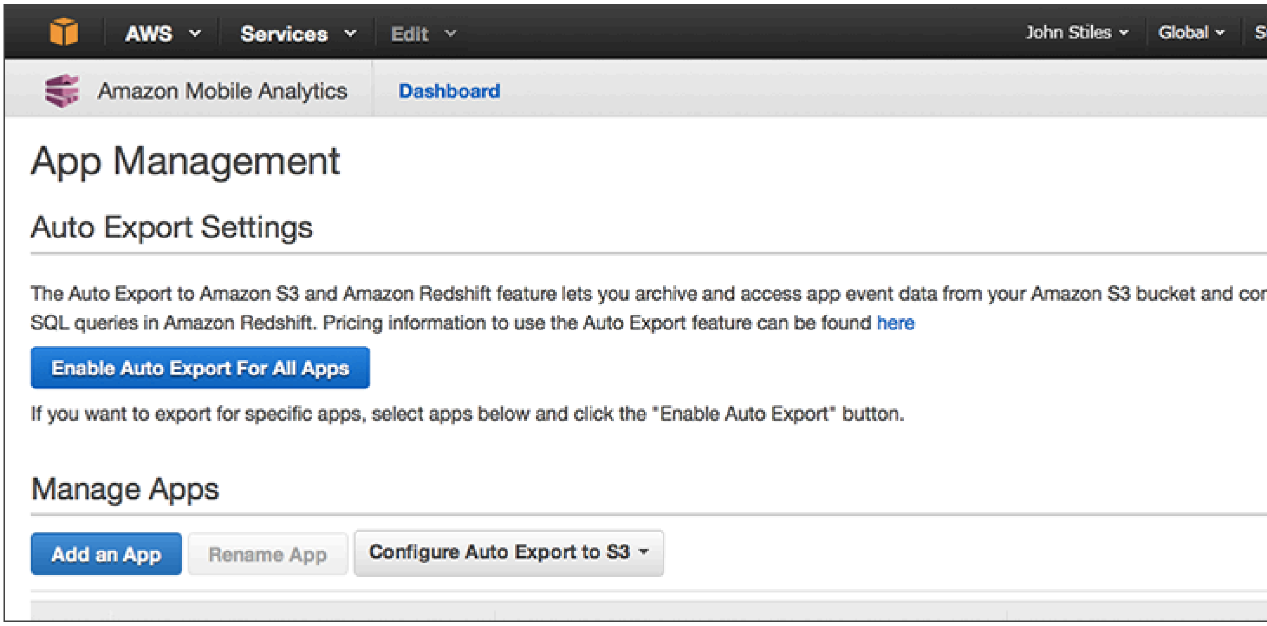
```
AWSCognitoCredentialsProvider *credentialsProvider = [[AWSCognitoCredentialsProvider alloc]
                                                      initWithRegionType: AWSRegionUSEast1
```

# Adding Additional Apps with the Default Cognito Identity Pool

Use the following procedure to add a new app if you have already added your first app and have a default Cognito identity pool.

### To add an app if you have existing apps

1. Open the Amazon Mobile Analytics console at https://console.aws.amazon.com/mobileanalytics/home.
2. Click the Settings ⚙ icon.
3. In **Manage Apps**, choose **Add an App**.



App Management

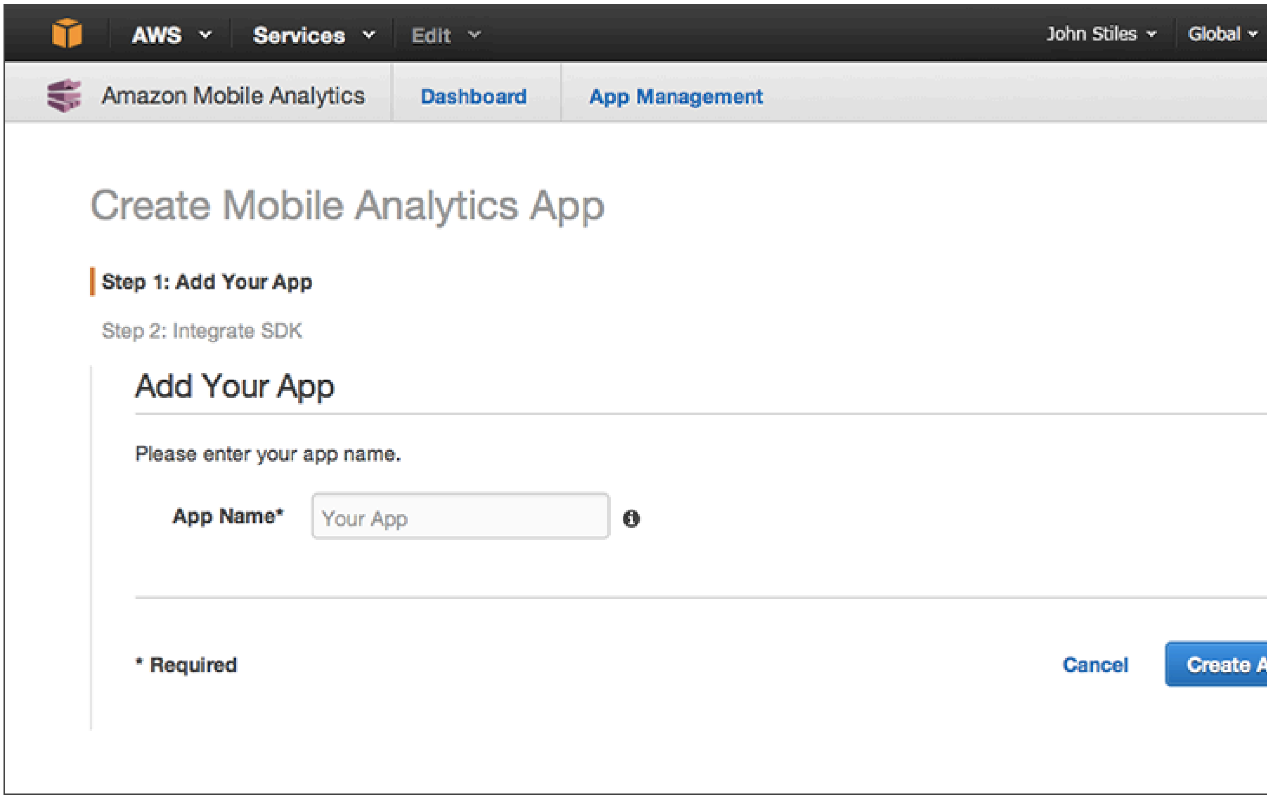
Auto Export Settings

The Auto Export to Amazon S3 and Amazon Redshift feature lets you archive and access app event data from your Amazon S3 bucket and cor SQL queries in Amazon Redshift. Pricing information to use the Auto Export feature can be found here

If you want to export for specific apps, select apps below and click the "Enable Auto Export" button.

Manage Apps

4. In **App Name**, type a name for your app, and then choose **Create App** to add the app using your default identity pool. This name will appear in the console.



Create Mobile Analytics App

Step 1: Add Your App

Step 2: Integrate SDK

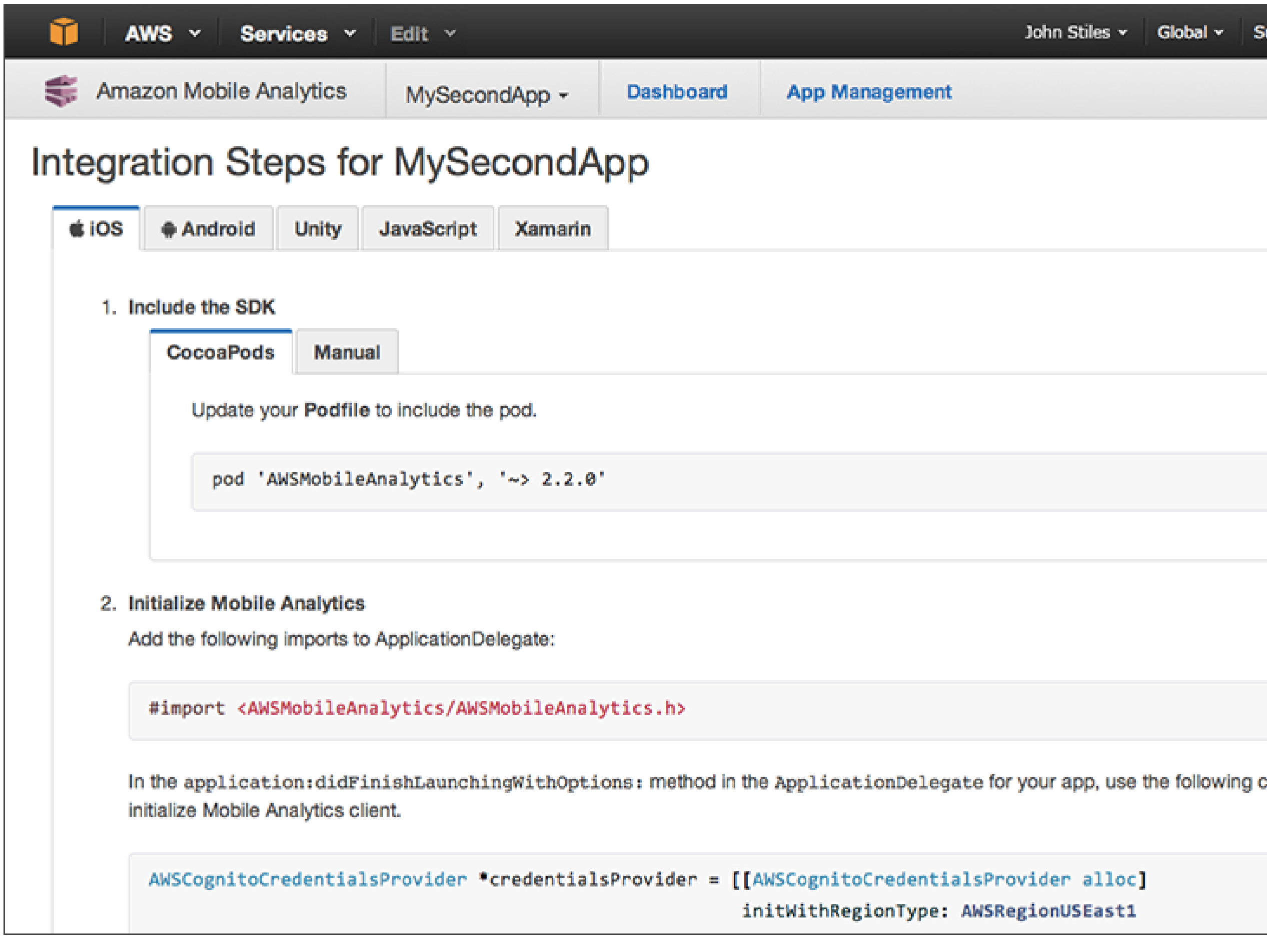
Add Your App

Please enter your app name.

App Name* Your App

* Required

5. Choose the tab that corresponds to the platform targeted by your app for details about how to integrate Mobile Analytics. The ID for your default identity pool appears in the sample code. You can easily copy and paste the code into your app. \



Integration Steps for MySecondApp

1. Include the SDK

   Update your **Podfile** to include the pod.

   ```
   pod 'AWSMobileAnalytics', '~> 2.2.0'
   ```

2. Initialize Mobile Analytics

   Add the following imports to ApplicationDelegate:

   ```
   #import <AWSMobileAnalytics/AWSMobileAnalytics.h>
   ```

   In the `application:didFinishLaunchingWithOptions:` method in the `ApplicationDelegate` for your app, use the following c
   initialize Mobile Analytics client.

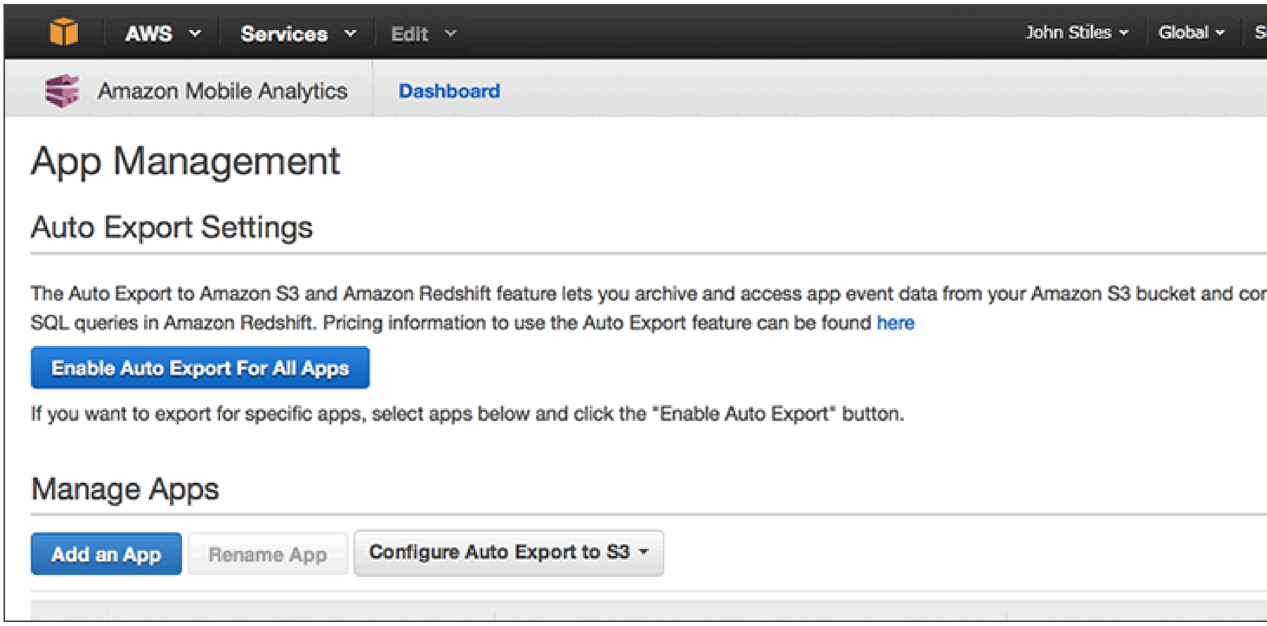
   ```
   AWSCognitoCredentialsProvider *credentialsProvider = [[AWSCognitoCredentialsProvider alloc]
                                                          initWithRegionType: AWSRegionUSEast1
   ```

# If You Have Existing Identity Pools but No Default Cognito Identity Pool

Use this procedure to add a new app if you used Mobile Analytics before July 6, 2015 or have an Cognito identity pool but have not selected a default :.

### To add an app if you have existing identity pools, but no default identity pool

1. Open the Amazon Mobile Analytics console at https://console.aws.amazon.com/mobileanalytics/home.
2. Click the Settings ⚙ icon.
3. In **Manage Apps**, choose **Add an App**.



App Management

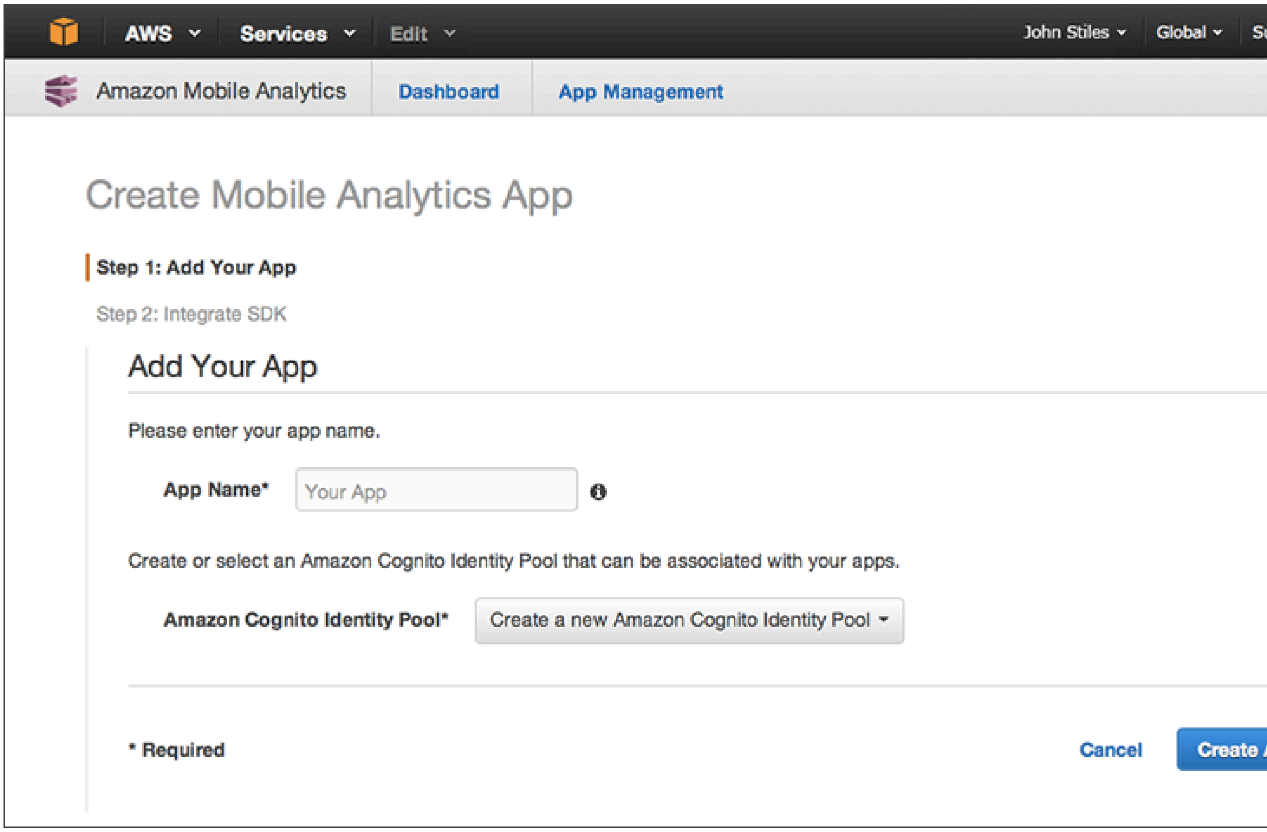
Auto Export Settings

The Auto Export to Amazon S3 and Amazon Redshift feature lets you archive and access app event data from your Amazon S3 bucket and cor
SQL queries in Amazon Redshift. Pricing information to use the Auto Export feature can be found here

If you want to export for specific apps, select apps below and click the "Enable Auto Export" button.

Manage Apps

4. In **App Name**, type a name for your app. This name will appear in the console.



Create Mobile Analytics App

Step 1: Add Your App

Step 2: Integrate SDK

Add Your App

Please enter your app name.

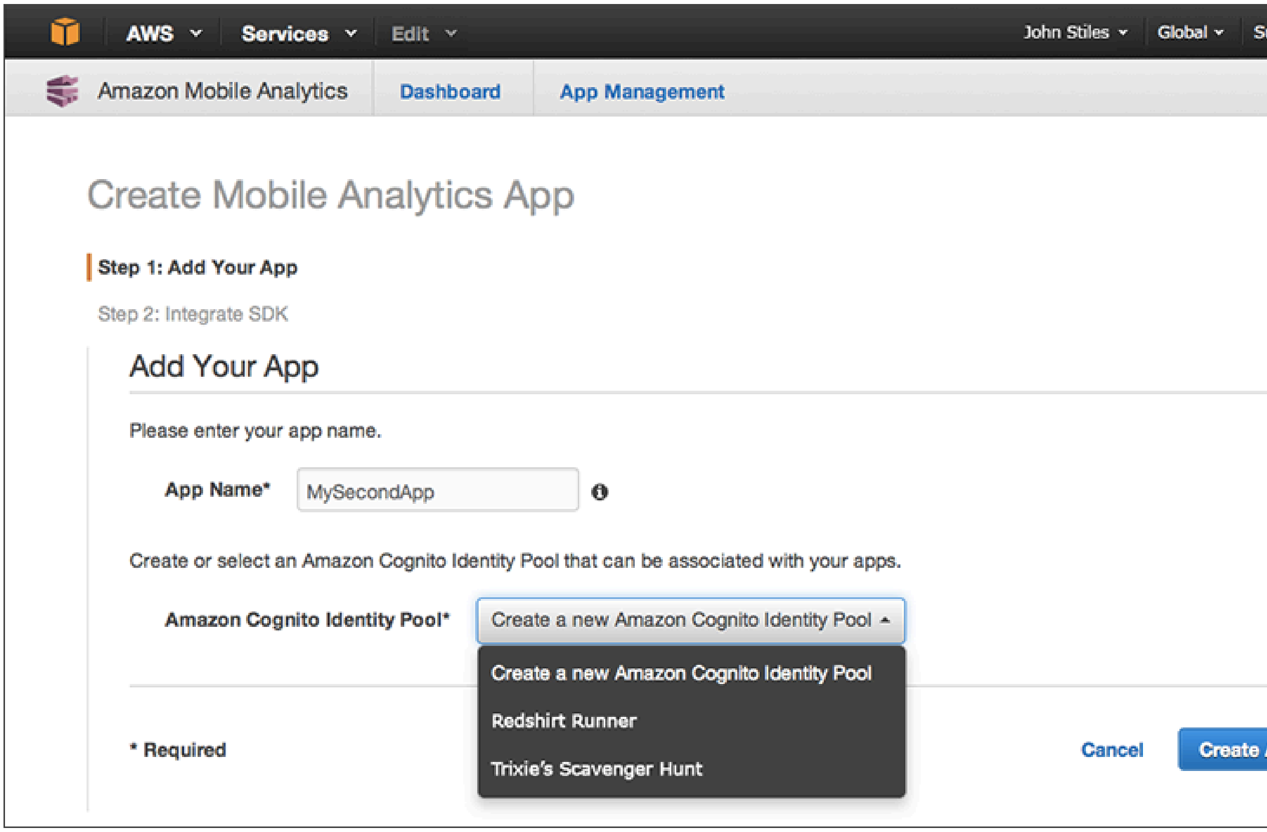
App Name* Your App

Create or select an Amazon Cognito Identity Pool that can be associated with your apps.

Amazon Cognito Identity Pool* Create a new Amazon Cognito Identity Pool

* Required

5. To use an existing identity pool, choose it from the **Amazon Cognito Identity Pool** drop-down list. To create a new identity pool, choose **Create a new Amazon Cognito Identity Pool**



Create Mobile Analytics App

Step 1: Add Your App

Step 2: Integrate SDK

Add Your App

Please enter your app name.

App Name* MySecondApp

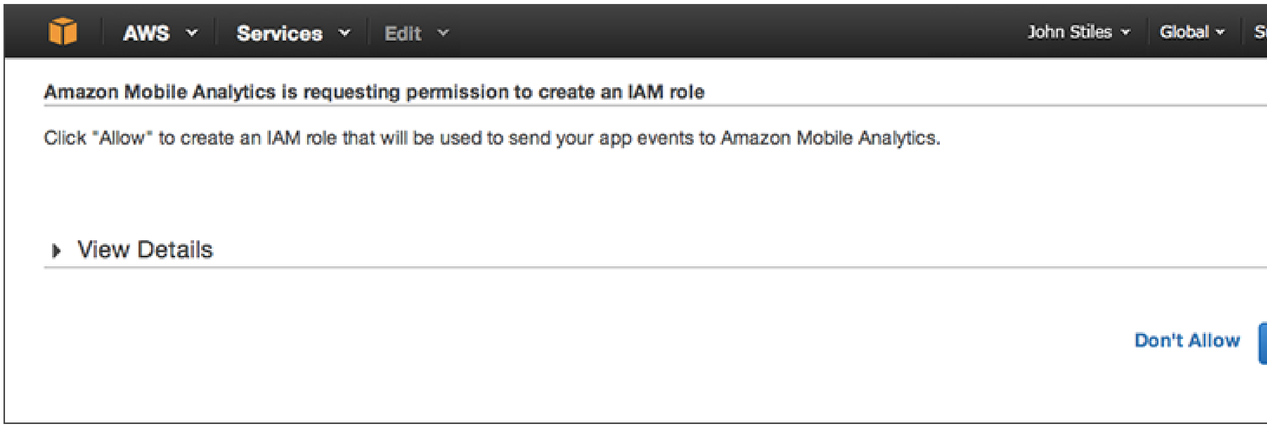
Create or select an Amazon Cognito Identity Pool that can be associated with your apps.

Amazon Cognito Identity Pool* Create a new Amazon Cognito Identity Pool

- Create a new Amazon Cognito Identity Pool
- Redshirt Runner
- Trixie's Scavenger Hunt

* Required

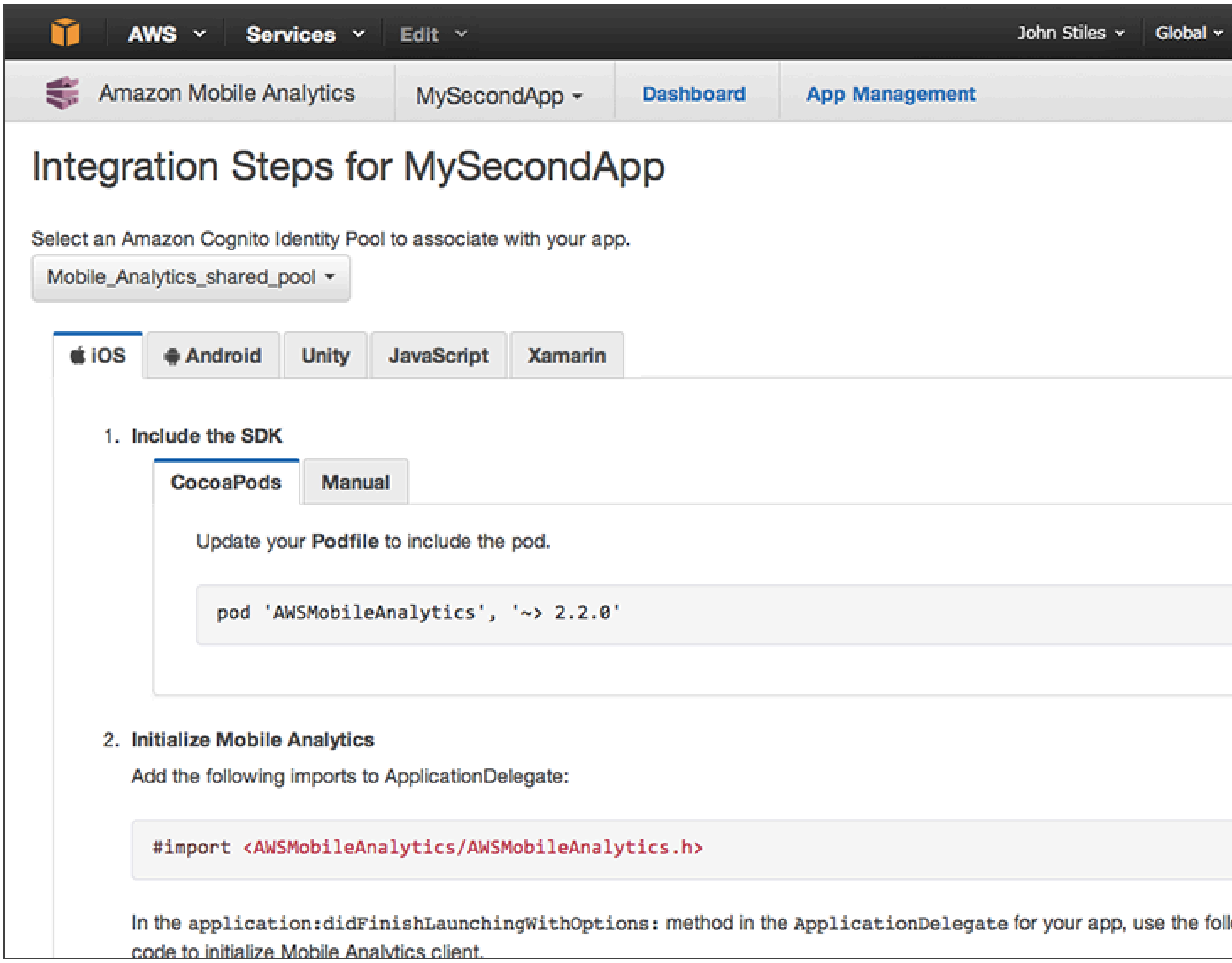
6. Choose **Create App**.
7. If you chose to create a new identity pool, the IAM role creation wizard will appear. Choose **Allow** to generate an IAM role for the Cognito identity pool.



Amazon Mobile Analytics is requesting permission to create an IAM role

Click "Allow" to create an IAM role that will be used to send your app events to Amazon Mobile Analytics.

8. Choose the tab for each platform targeted by your app for instructions and sample code. The ID for the Cognito identity pool you created or chose appears in the sample code. You can easily copy and paste the code into your app.



Integration Steps for MySecondApp

Select an Amazon Cognito Identity Pool to associate with your app.

1. **Include the SDK**

   Update your **Podfile** to include the pod.

```
pod 'AWSMobileAnalytics', '~> 2.2.0'
```

2. **Initialize Mobile Analytics**

   Add the following imports to ApplicationDelegate:

```
#import <AWSMobileAnalytics/AWSMobileAnalytics.h>
```

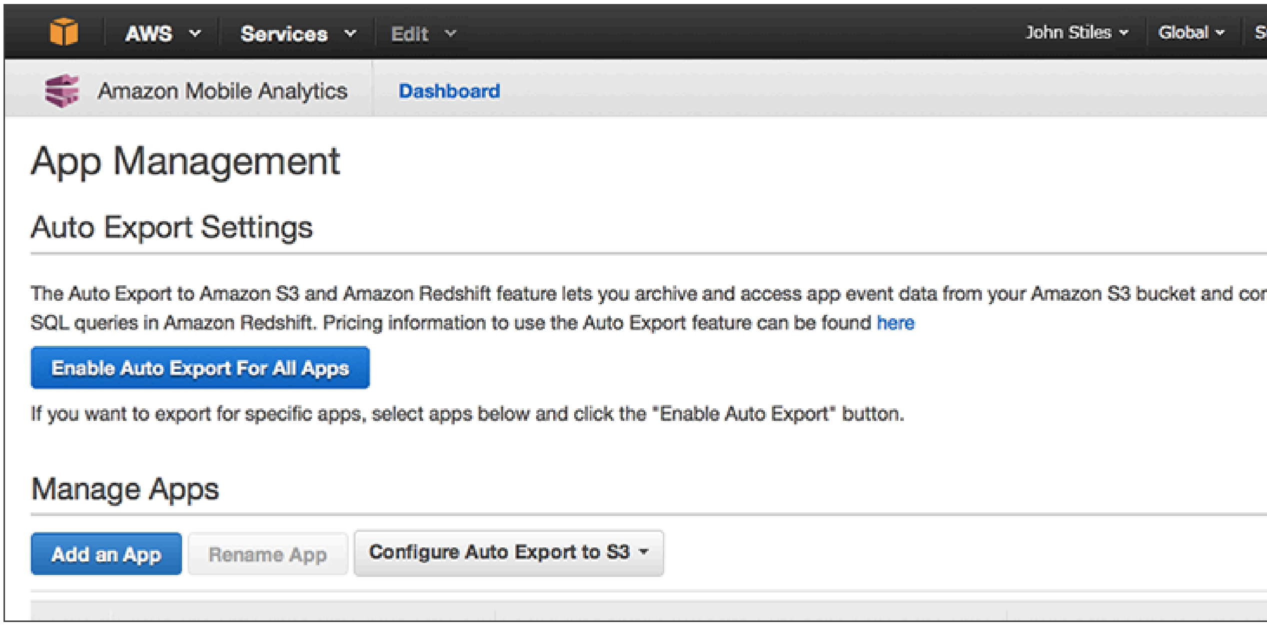
In the `application:didFinishLaunchingWithOptions:` method in the `ApplicationDelegate` for your app, use the foll
code to initialize Mobile Analytics client.

## If You Have Existing Identity Pools and a Default Cognito Identity Pool

Use this procedure if you are Amazon Cognito user who would like to start using Mobile Analytics.

### To add an app if you have existing identity pools and a default identity pool

1. Open the Amazon Mobile Analytics console at https://console.aws.amazon.com/mobileanalytics/home.
2. Click the Settings ⚙ icon.
3. In **Manage Apps**, choose **Add an App**.



App Management

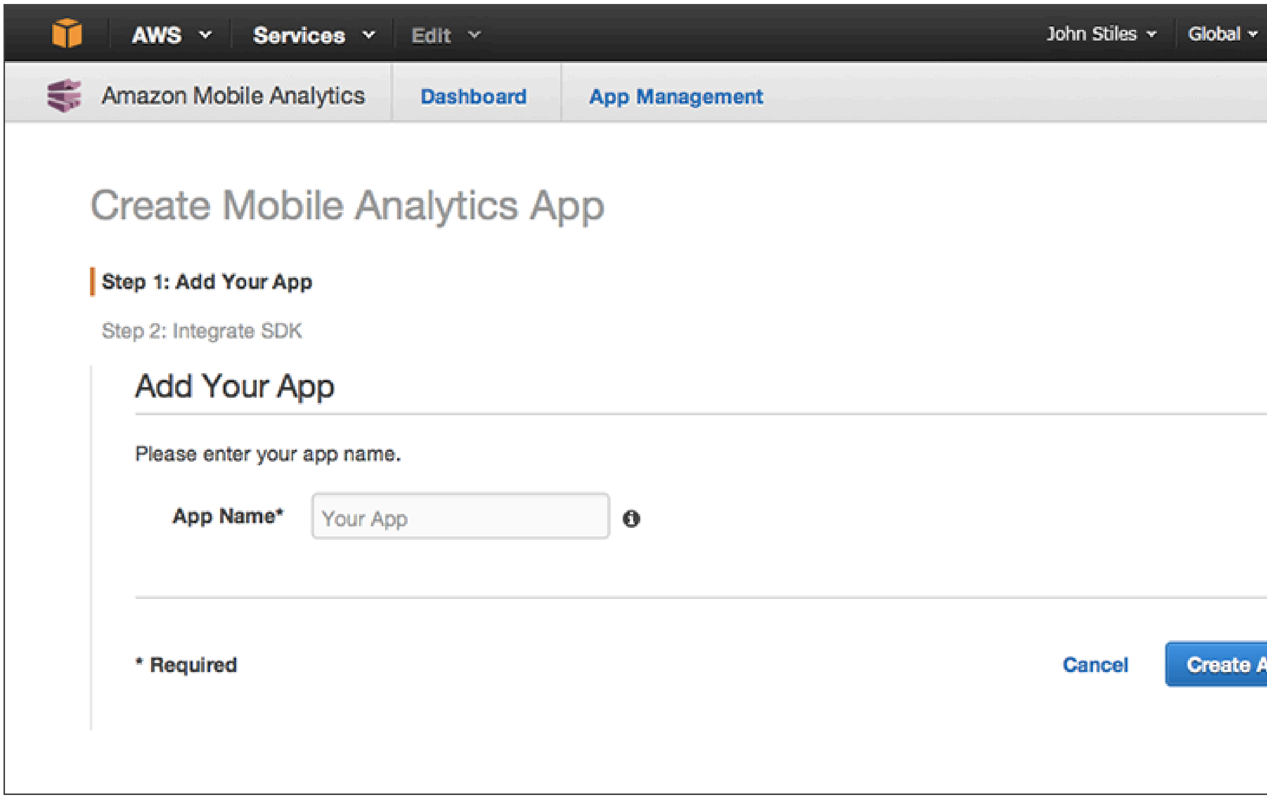
Auto Export Settings

The Auto Export to Amazon S3 and Amazon Redshift feature lets you archive and access app event data from your Amazon S3 bucket and cor
SQL queries in Amazon Redshift. Pricing information to use the Auto Export feature can be found here

If you want to export for specific apps, select apps below and click the "Enable Auto Export" button.

Manage Apps

4. In **App Name**, type a name for your app, and then choose **Create App**. This name will appear in the console.



Create Mobile Analytics App

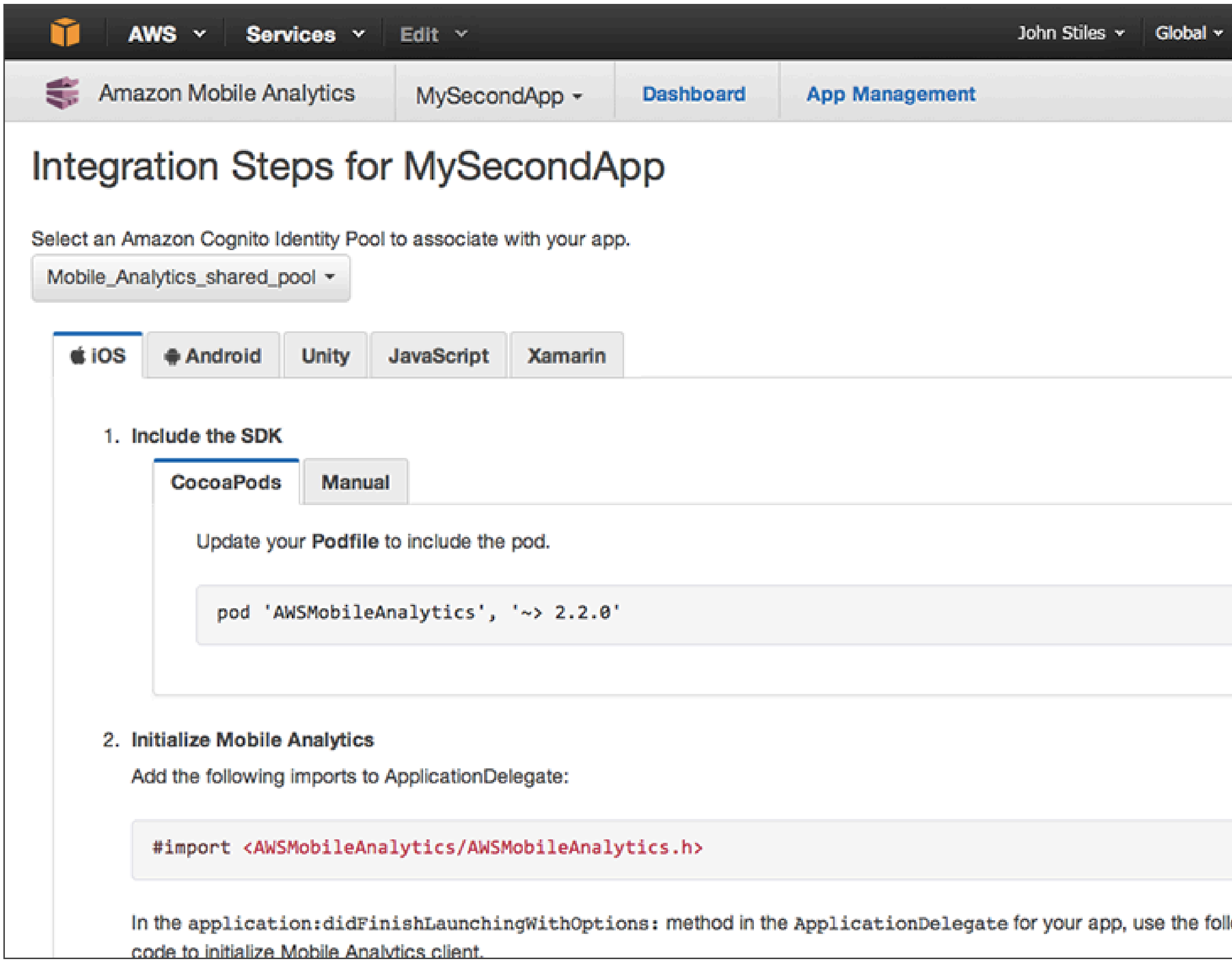
Add Your App

Please enter your app name.

App Name* Your App

* Required

5. Choose the tab for each platform targeted by your app for instructions and sample code to use. The ID for your default Cognito identity pool is automatically associated with the new app, but you can select an other identity pools from the drop-down list. The sample code contains the values appropriate to your app. You can easily copy and paste the values into your app's source code.



Integration Steps for MySecondApp

Select an Amazon Cognito Identity Pool to associate with your app.

1. Include the SDK

Update your **Podfile** to include the pod.

```
pod 'AWSMobileAnalytics', '~> 2.2.0'
```

2. Initialize Mobile Analytics

Add the following imports to ApplicationDelegate:

```
#import <AWSMobileAnalytics/AWSMobileAnalytics.h>
```

In the `application:didFinishLaunchingWithOptions:` method in the `ApplicationDelegate` for your app, use the foll
code to initialize Mobile Analytics client.

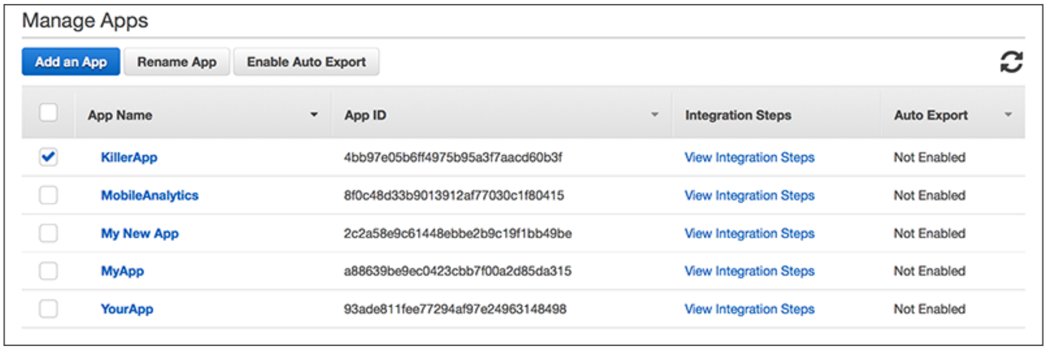
# Renaming an App

You can rename an app in the **App Management** section of the console at https://console.aws.amazon.com/mobileanalytics/home.

> **Note**
> Renaming an app can currently only be done by the root account.

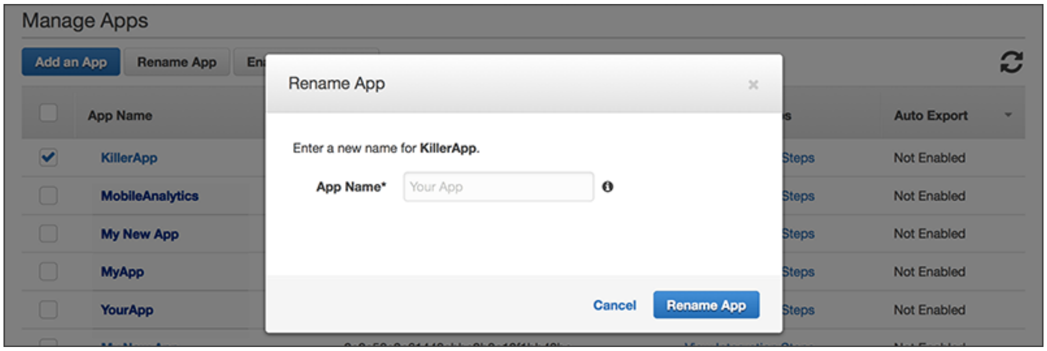
### To rename an app

1. In the Mobile Analytics console, go to **App Management**.



| | App Name | App ID | Integration Steps | Auto Export |
|---|---|---|---|---|
| ☑ | KillerApp | 4bb97e05b6ff4975b95a3f7aacd60b3f | View Integration Steps | Not Enabled |
| □ | MobileAnalytics | 8f0c48d33b9013912af77030c1f80415 | View Integration Steps | Not Enabled |
| □ | My New App | 2c2a58e9c61448ebbe2b9c19f1bb49be | View Integration Steps | Not Enabled |
| □ | MyApp | a88639be9ec0423cbb7f00a2d85da315 | View Integration Steps | Not Enabled |
| □ | YourApp | 93ade811fee77294af97e24963148498 | View Integration Steps | Not Enabled |

2. In the **Manage Apps** list, select the check box next to the app you want to rename.
3. Choose **Rename App**.
4. In the **App Name** box, type the new name for the app, and then choose **Rename App**.



Enter a new name for **KillerApp**.

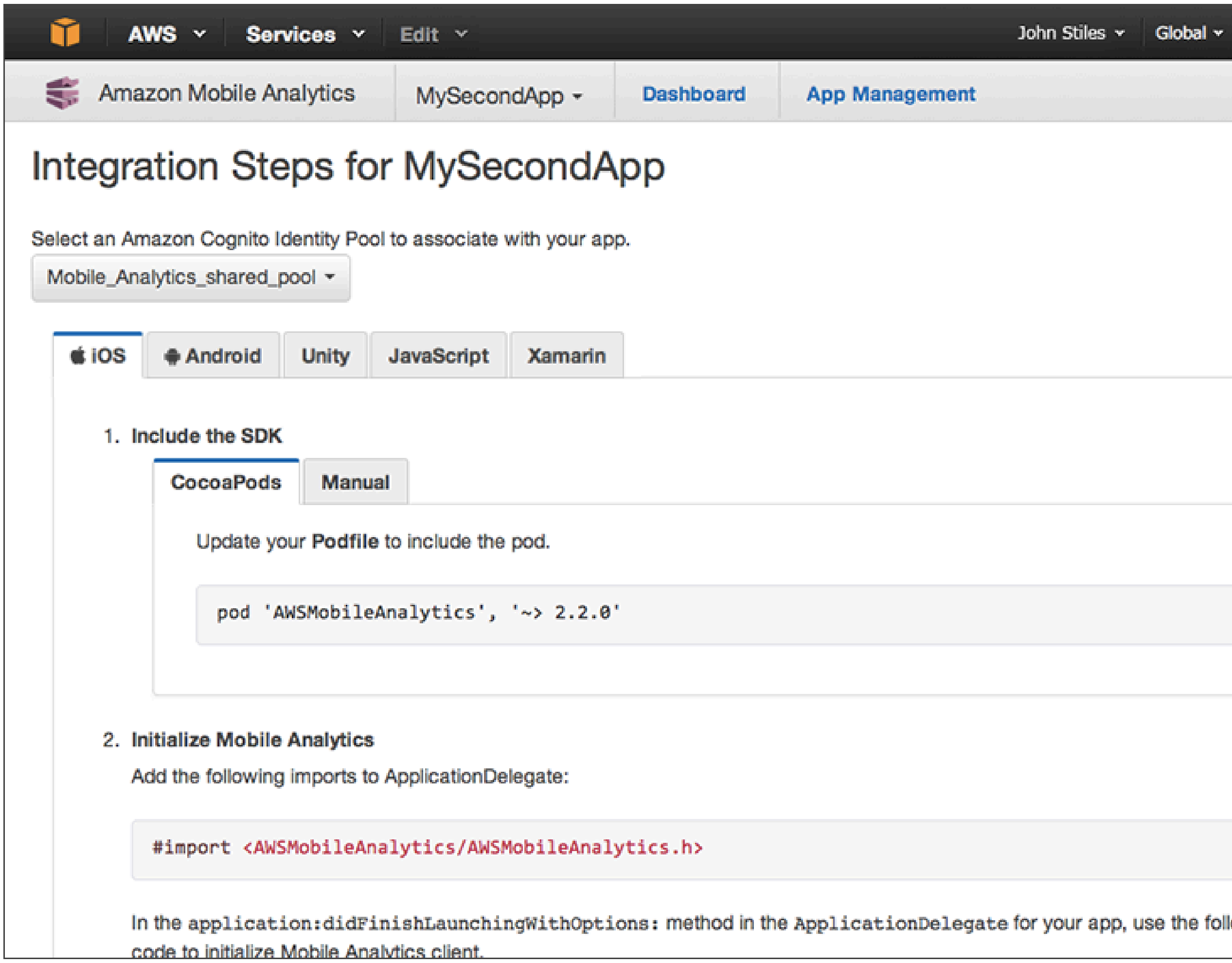
App Name* Your App

# Viewing Integration Steps

Mobile Analytics helps you integrate the service into the source code for your app. It provides blocks of code you can copy and paste into your source code as well as information about where in your app to add it. Where an integration step requires it, the code includes the `appID` to connect the data sent by the app to Mobile Analytics for generating reports and stats.

### To view the integration steps for an app

1. From the **Dashboard**, choose **Manage Apps** in the application list in the toolbar.
2. In the **Manage Apps** list, choose **View Integration Steps** next to the app whose details you want to view.



Integration Steps for MySecondApp

Select an Amazon Cognito Identity Pool to associate with your app.

Mobile_Analytics_shared_pool

1. Include the SDK

Update your **Podfile** to include the pod.

```
pod 'AWSMobileAnalytics', '~> 2.2.0'
```

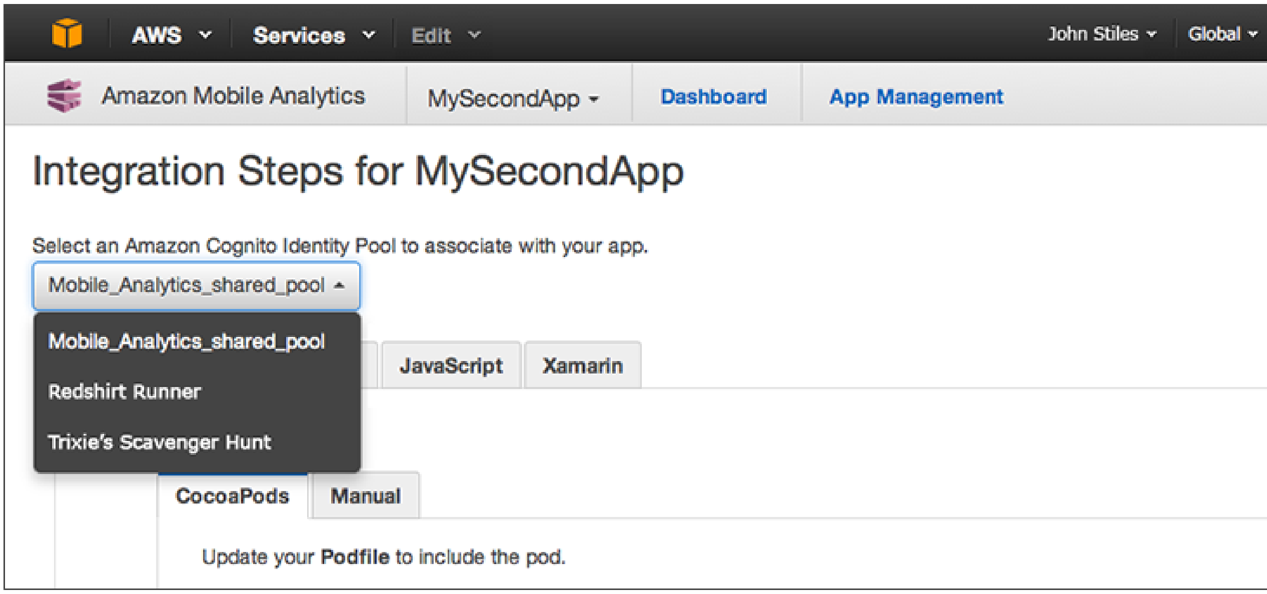
2. Initialize Mobile Analytics

Add the following imports to ApplicationDelegate:

```
#import <AWSMobileAnalytics/AWSMobileAnalytics.h>
```

In the `application:didFinishLaunchingWithOptions:` method in the `ApplicationDelegate` for your app, use the foll
code to initialize Mobile Analytics client.

3. If you have additional non-default identity pools, you can change the identity pool associated with this app. Simply choose it from the drop-down menu.



Integration Steps for MySecondApp

Select an Amazon Cognito Identity Pool to associate with your app.

Mobile_Analytics_shared_pool

Update your **Podfile** to include the pod.

# Using Mobile Analytics with the Mobile SDK

To take advantage of Mobile Analytics, your mobile app must incorporate code that generates, records, and submits events to the Mobile Analytics service. This section describes concepts you need to know to incorporate Mobile Analytics into a mobile app.

To integrate Mobile Analytics features that make data collection possible, use the APIs provided by the AWS Mobile SDK for the platforms you plan to support. When working on platforms other than those supported by the AWS Mobile SDK, use the REST API to access Mobile Analytics.

## Platforms Supported by the Mobile SDK

Mobile Analytics is supported by the AWS Mobile SDK, providing an API for each of these platforms:

- **iOS:** Track App Usage Data with Amazon Mobile Analytics.
- **Android:** Amazon Mobile Analytics.
- **JavaScript:** Mobile Analytics SDK for JavaScript.
- **Unity:** Amazon Mobile Analytics.
- **Xamarin:** Amazon Mobile Analytics.

If you are working on platforms other than those supported by the AWS Mobile SDK, use the REST API to access the Mobile Analytics service.

Topics

# Enabling SDK Logging

A very useful tool for troubleshooting issues with Mobile Analytics is a log of the interactions between your app and the Mobile SDK. How you switch on SDK logging depends on which SDK you use.

## Enabling Logging for iOS Apps

To turn on verbose logging of the AWS Mobile SDK for iOS, call the following line in the `-application:didFinishLaunchingWithOptions:` application delegate.

**Swift:**

```
@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

 var window: UIWindow?
 func application(application: UIApplication, didFinishLaunchingWithOptions
 launchOptions: [NSObject : AnyObject]?) -> Bool {
  ...
  AWSLogger.defaultLogger().logLevel = .Verbose
  ...

  return true
 }
}
```

**Objective-C:**

```
#import <AWSCore/AWSCore.h>

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
 ...
 [AWSLogger defaultLogger].logLevel = AWSLogLevelVerbose;
 ...

 return YES;
}

@end
```

For more information, see Logging in the AWS Mobile SDK iOS Developer Guide.

## Enabling Logging for Android Apps

To turn on verbose logging of the AWS Mobile SDK for Android, include the following line of code in the `onCreate` method of the main activity of the app.

```
java.util.logging.Logger.getLogger("com.amazonaws").setLevel(Level.ALL);
```

## Enabling Logging for JavaScript Apps

The Amazon Mobile Analytics SDK for JavaScript enables logging by passing a Logger object to the AMA.Client.Options object. The Logger object uses Javascript-style log levels with a separate function for each logging level. Because the console object adheres to the logger interface (`.log`, `.error`, `.info`, `.debug`) you can pass it directly to the Manager constructor.

```
var options = {
    appId    : MOBILE_ANALYTICS_APP_ID,       //Required e.g.
 'c5d69c75a92646b8953126437d92c0'
    platform : DEVICE_PLATFORM,               //Optional valid values:
 'Android', 'iPhoneOS'
    logger   : console                        //Specifying logger
};
mobileAnalyticsClient = new AMA.Manager(options);
```

For more information, see the Amazon Mobile Analytics SDK for JavaScript.

## Enabling Logging for Unity Apps

For information on how to switch on logging in apps using the AWS Mobile SDK for Unity, see Set Logging Information in the AWS Mobile SDK Unity Developer Guide.

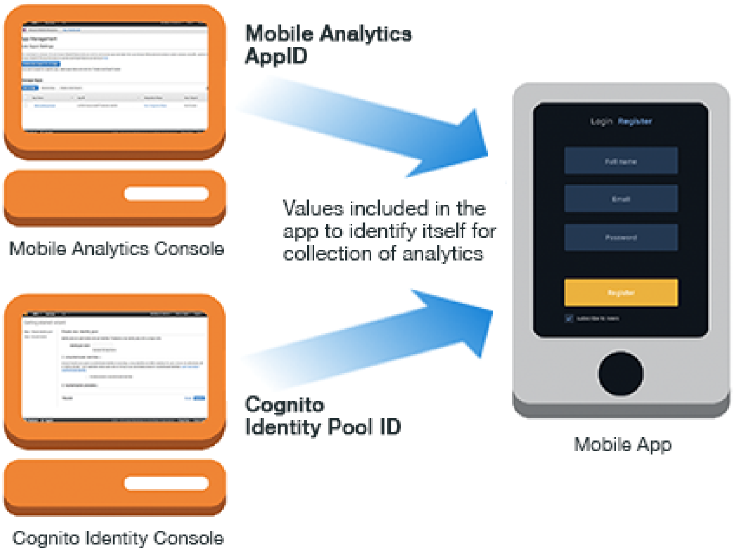
## Enabling Logging for Xamarin Apps

For information on how to switch on logging in apps using the AWS Mobile SDK for Xamarin, see Set Logging in the AWS Mobile SDK Xamarin Developer Guide.

# Identifying the App in Mobile Analytics

Before Mobile Analytics can collect and present any analytic data about your application, you must identify your app to Mobile Analytics by providing the following information:

- The Amazon Cognito identity pool created to authenticate users of your app.
- The AppID created in the Mobile Analytics console when you added your app to Mobile Analytics.



For information about providing the Mobile Analytics `appID` and the Cognito identity pool ID in the source code of your app, see the AWS Mobile SDK for your platform.

- AWS Mobile SDK Android Developer Guide
- AWS Mobile SDK iOS Developer Guide
- Mobile Analytics SDK for JavaScript
- AWS Mobile SDK Unity Developer Guide
- AWS Mobile SDK Xamarin Developer Guide

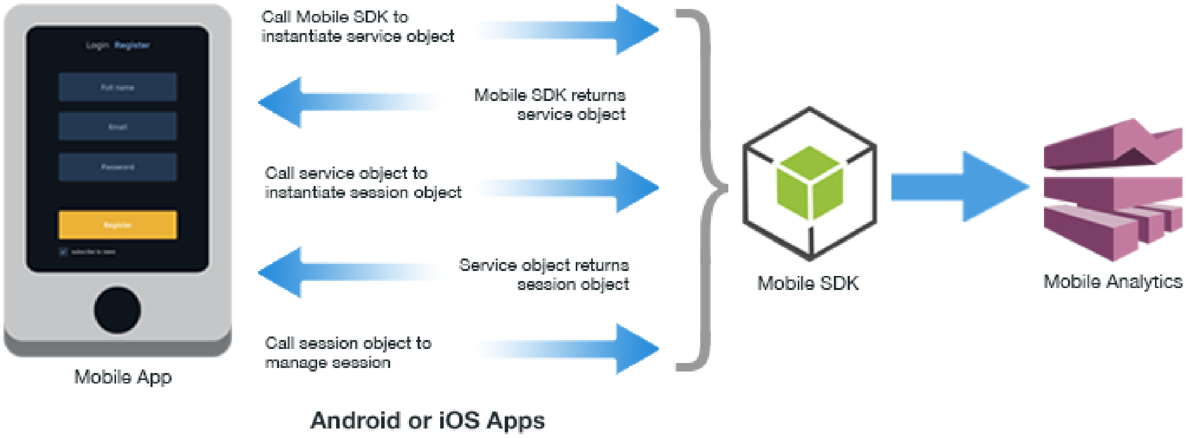
# Managing Sessions

Mobile Analytics reports app usage data based on activity captured within sessions. A *session* is a period of time during which the app is active on the device. A session begins when an app is launched or brought to the foreground, and ends when the app is terminated or goes to the background. There is an inactivity period of up to 5 seconds, so a brief interruption, like receiving a text message, does not count as a new session.

Mobile Analytics tracks sessions and events submitted during each session to generate the reports displayed in the console. Total Daily Sessions shows the number of sessions your app has each day. Average Sessions per Daily Active User shows the mean number of sessions per user per day.

## Managing Sessions in iOS or Android Apps

Apps that use the AWS Mobile SDK for iOS or AWS Mobile SDK for Android to access Mobile Analytics start a session when they create the service object for their platform. To manage a running session, the app must additionally create the session object. This session object gives your app access to session management.


Login Register
Full name
Email
Password
Register
subscribe to news
Mobile App
Call Mobile SDK to instantiate service object
Mobile SDK returns service object
Call service object to instantiate session object
Service object returns session object
Call session object to manage session
Mobile SDK
Mobile Analytics
Android or iOS Apps

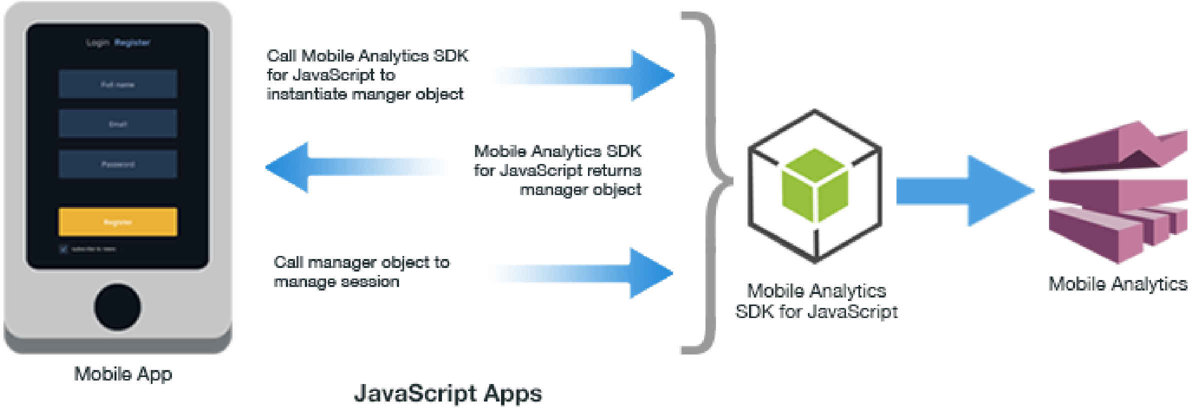
For information about managing sessions in iOS or Android apps, see:

- AWS Mobile SDK iOS Developer Guide
- AWS SDK for iOS Reference

- AWS Mobile SDK Android Developer Guide
- AWS SDK for Android Reference

## Managing Sessions in JavaScript Apps

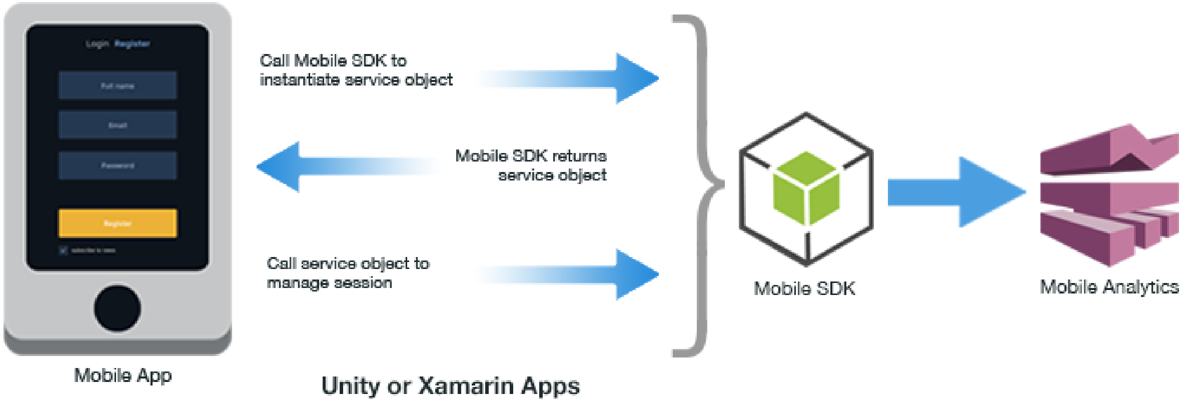
JavaScript apps must include the AWS Mobile SDK for JavaScript in the Browser as well as the Amazon Mobile Analytics SDK for JavaScript When your app starts, it must create an `AMA.Manager` object, which starts a session and gives your app access to session management.


Login Register
Full name
Email
Password
Register
subscribe to news
Mobile App
Call Mobile Analytics SDK for JavaScript to instantiate manger object
Mobile Analytics SDK for JavaScript returns manager object
Call manager object to manage session
Mobile Analytics SDK for JavaScript
Mobile Analytics
JavaScript Apps

For information about managing sessions in JavaScript apps, see the Amazon Mobile Analytics SDK for JavaScript.

## Managing Sessions in Unity or Xamarin Apps

When your app starts, it should start a session that Mobile Analytics can then begin to track. Apps that use the AWS Mobile SDK for Unity or Xamarin to access Mobile Analytics automatically start a session when they create the service object. The service object gives your app access to session management.


Login Register
Full name
Email
Password
Register
Mobile App
Call Mobile SDK to instantiate service object
Mobile SDK returns service object
Call service object to manage session
Unity or Xamarin Apps
Mobile SDK
Mobile Analytics

For information about session management in Unity and Xamarin apps, refer to

- AWS Mobile SDK Unity Developer Guide
- AWS SDK for Unity Reference

- AWS Mobile SDK Xamarin Developer Guide
- AWS SDK for Xamarin Reference

# Generating Mobile Analytics Events

Information about user engagement in your app is sent to the Mobile Analytics service using events. The events needed to produce the basic analytics reports in the console are collected and sent automatically as long as a session is active. There are two types of events your app can generate in addition to those generated automatically:

- Monetization events
- Custom events

Monetization events are specialized events used to report on monetization activities in the app, such as in-app purchases. Custom events are those you create to monitor activities specific to your app, such as completing a level in a game, posting to social media, or setting particular app preferences. An app can have up to 1,500 unique custom events.

> **Naming Custom Events**
> When naming custom events, do not begin event names with an underscore (_). Events with names beginning with an underscore are filtered out.

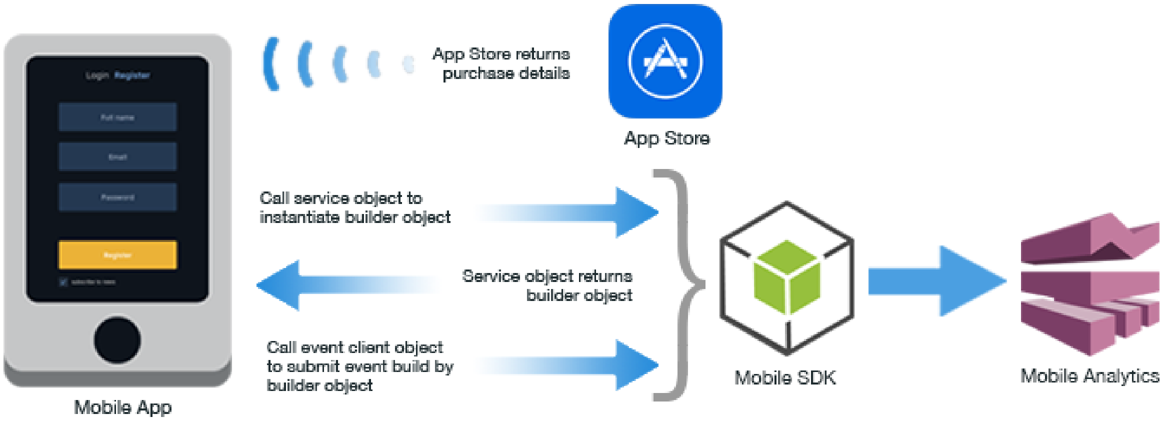
Topics

## Creating a Monetization Event

When a user of your app makes a purchase, the app code handling the purchase sends a monetization event. Data from monetization events are used to enable and populate revenue-focused reports in the console such as Average Revenue Per User (ARPU) and others.

## Creating a Monetization Event in iOS Apps

When a user of your app makes an in-app purchase, you can create a monetization event that helps you track the monetary performance of the app. To do this in an iOS app using the AWS Mobile SDK create a builder object to hold data returned by the Apple App Store. You then use the builder object to build a monetization event you submit with the event client object (p. 27).


App Store returns purchase details
App Store
Call service object to instantiate builder object
Service object returns builder object
Call event client object to submit event build by builder object
Mobile SDK
Mobile Analytics
Mobile App

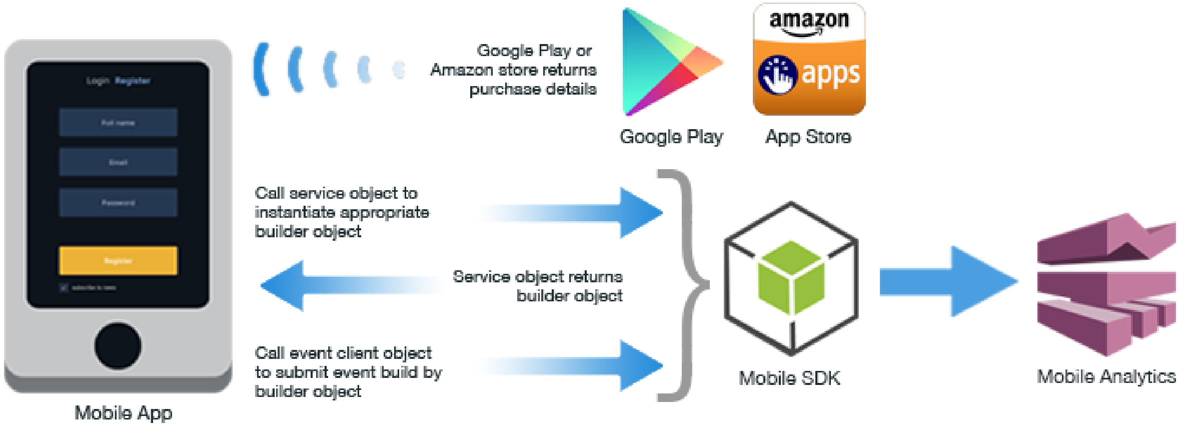
iOS Apps

For information about creating monetization events in iOS apps, see:

- AWS Mobile SDK iOS Developer Guide
- AWS SDK for iOS Reference

## Creating a Monetization Event in Android Apps

When a user makes an in-app purchase, you can create a monetization event to help you track the app's monetary performance. To do this in an Android app using the AWS Mobile SDK, create a builder object to hold data returned by the online store. There are distinct builder object classes to handle purchases from the Amazon store, the Google Play store, and from an IAP framework not defined by a specific builder. You then use the builder object to build a monetization event you submit with the event client object (p. 27).


Google Play or Amazon store returns purchase details
Google Play
amazon apps
App Store
Call service object to instantiate appropriate builder object
Service object returns builder object
Call event client object to submit event build by builder object
Mobile SDK
Mobile Analytics
Mobile App

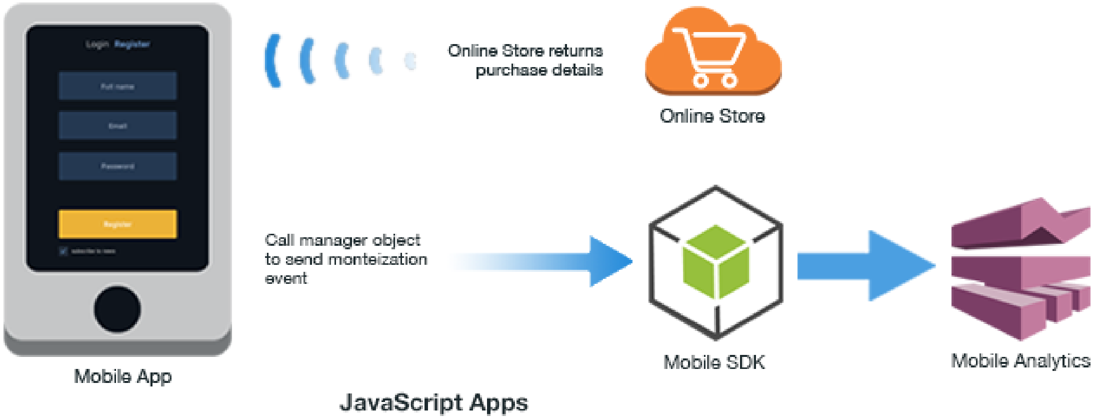
Android Apps

For information about creating monetization events in Android apps, see:

- AWS Mobile SDK Android Developer Guide
- AWS SDK for Android Reference

## Creating a Monetization Event in JavaScript Apps

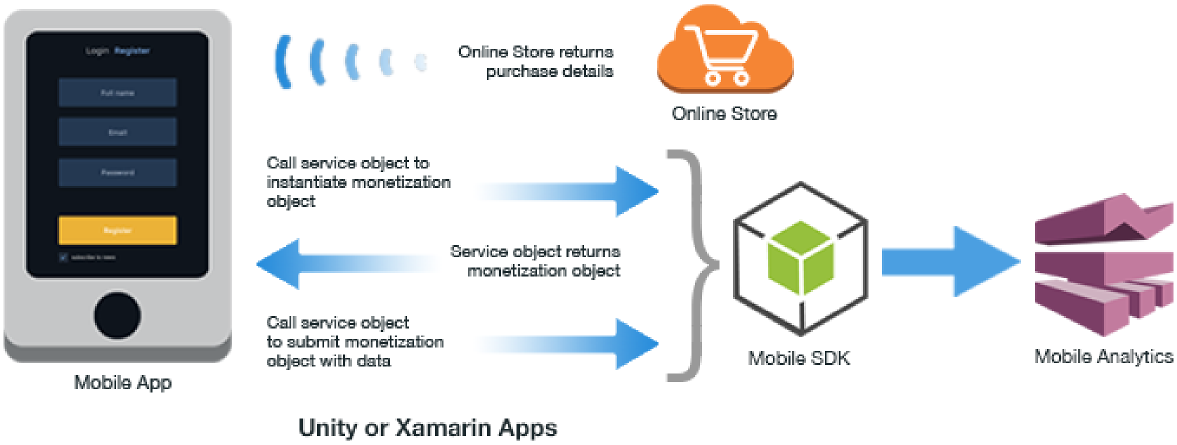
When a user of your app makes an in-app purchase, you can create a monetization event to help you track the app's monetary performance. To do this in a JavaScript app using the Mobile Analytics SDK for JavaScript, you call a method on the manager object, passing the purchase data as parameters.


Online Store returns purchase details
Online Store
Call manager object to send monteization event
Mobile SDK
Mobile Analytics
Mobile App
JavaScript Apps

For information about creating monetization events in JavaScript apps, see Mobile Analytics SDK for JavaScript.

## Creating a Monetization Event in Unity or Xamarin Apps

When a user of your app makes an in-app purchase, you can create a monetization event to help you track the app's monetary performance. To do this in a Unity app using the AWS Mobile SDK for Unity or a Xamarin app using the AWS Mobile SDK for Xamarin, create a monetization event object to hold data from the online store. Populate the purchase data, and then submit the monetization object to a method on the service object.


Online Store returns purchase details
Online Store
Call service object to instantiate monetization object
Service object returns monetization object
Call service object to submit monetization object with data
Mobile SDK
Mobile Analytics
Mobile App
Unity or Xamarin Apps

For information about creating monetization events in Unity or Xamarin apps, see:

- AWS Mobile SDK Unity Developer Guide
- AWS SDK for Unity Reference

- AWS Mobile SDK Xamarin Developer Guide
- AWS SDK for Xamarin Reference

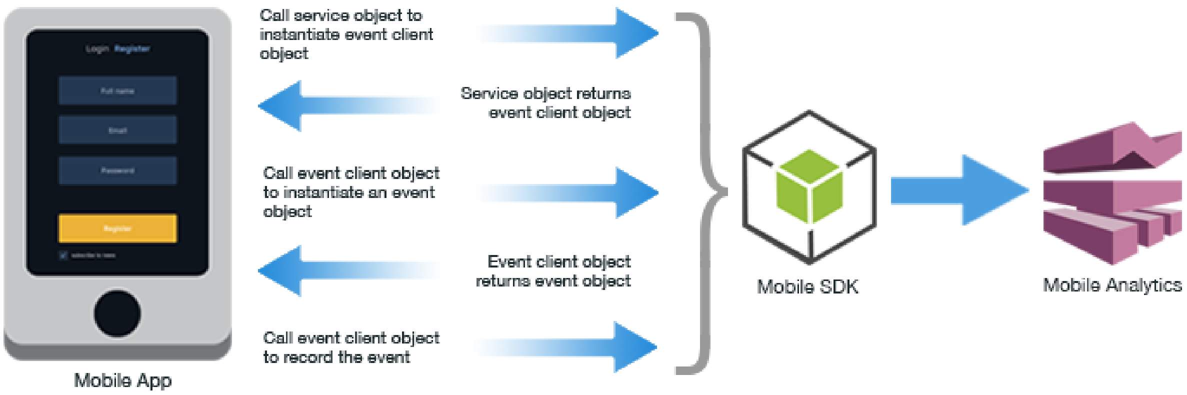
# Creating a Custom Event

You assign an eventType to your custom event . As a best practice, we recommend you give a general name to a custom event and specific names to attributes or metrics. For example, using "Item Bought" instead of "Item XYZ" as a custom event name helps keep the report from having too many distinct event names that are hard to read and aggregate.

## Creating a Custom Event in iOS or Android Apps

Custom events in iOS and Android apps are created and defined using an event object that the app submits to Mobile Analytics. To create these event objects, the app must first create an event client object, and then request individual event objects from the event client.

After the app has used an event client object to obtain a single event object, the app customizes the event object by adding attributes and metrics (p. 28) that specify data values to report. After the event object is customized, the app calls the event client object to record the event.


Call service object to instantiate event client object
Service object returns event client object
Call event client object to instantiate an event object
Event client object returns event object
Call event client object to record the event
Mobile App
Mobile SDK
Mobile Analytics

iOS or Android Apps

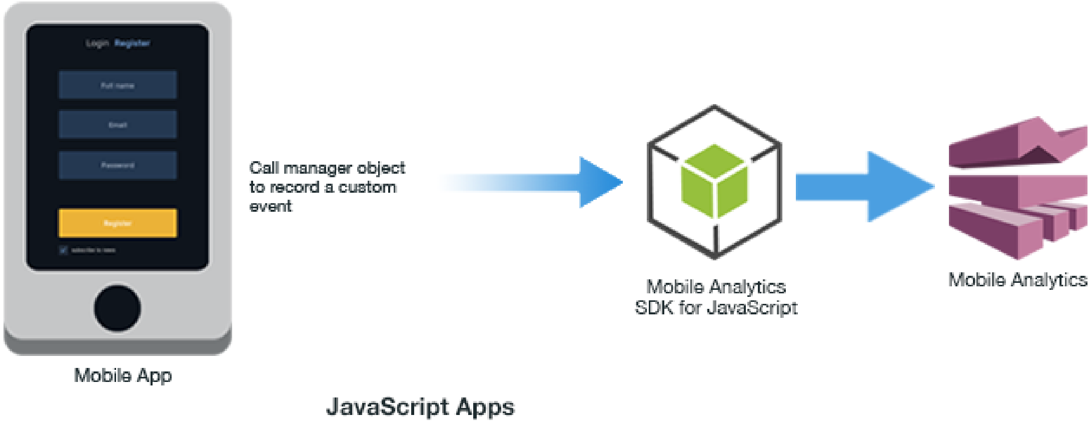
For information about creating custom events in iOS or Android apps, see:

- AWS Mobile SDK iOS Developer Guide
- AWS SDK for iOS Reference

- AWS Mobile SDK Android Developer Guide
- AWS SDK for Android Reference

## Creating a Custom Event in JavaScript Apps

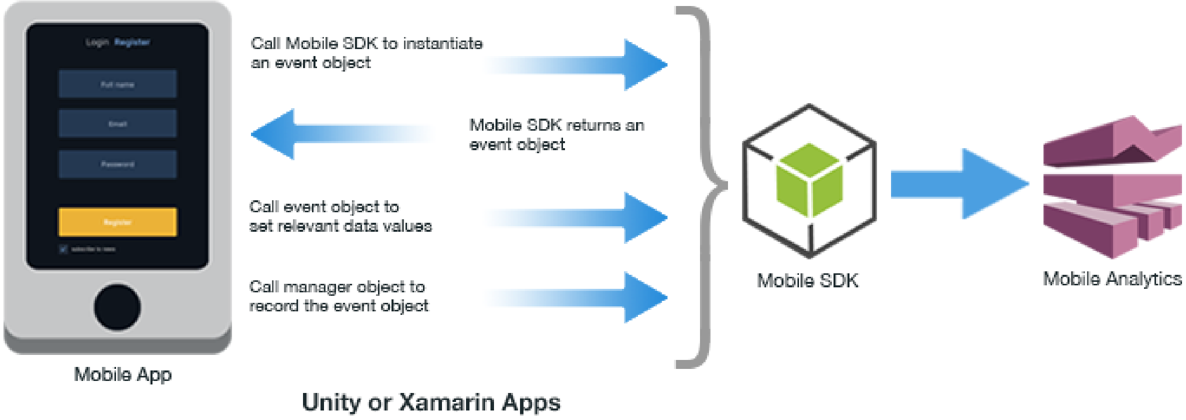
To create a custom a custom event in a JavaScript app, call the manager object to record a custom event, passing the attributes and metrics you want to capture as parameters.


Call manager object to record a custom event
Mobile App
Mobile Analytics SDK for JavaScript
Mobile Analytics
JavaScript Apps

For information about creating custom events in JavaScript apps, see Mobile Analytics SDK for JavaScript.

## Creating a Custom Event in Unity or Xamarin Apps

Custom events in Unity or Xamarin apps are created and defined using an event object the app submits to Mobile Analytics. After the app has created an event object, the app customizes the event object by adding attributes and metrics (p. 28) that specify data values to report. After the event object is customized, the app calls the manager object to record the event.


Call Mobile SDK to instantiate an event object
Mobile SDK returns an event object
Call event object to set relevant data values
Call manager object to record the event object
Mobile App
Mobile SDK
Mobile Analytics
Unity or Xamarin Apps

For information about creating custom events in Unity or Xamarin apps, see:

- AWS Mobile SDK Unity Developer Guide
- AWS SDK for Unity Reference

- AWS Mobile SDK Xamarin Developer Guide
- AWS SDK for Xamarin Reference

# Adding Attributes and Metrics

*Attributes* are data that provides context for the submitted event. For example, a game that submits an event to Mobile Analytics when the player collects a power-up bonus might include an attribute named for the type of bonus collected. You add attributes to an event as a collection of key-value pairs.

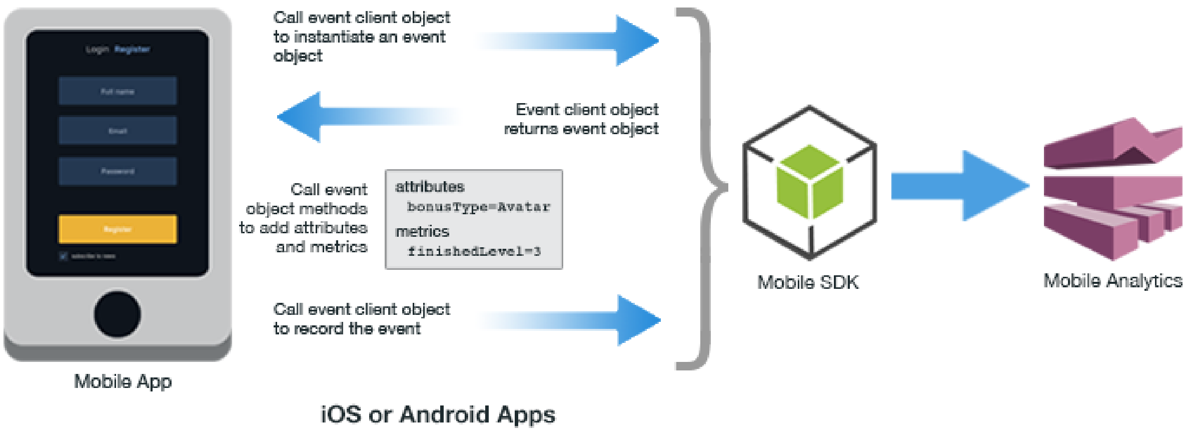
*Metrics* are data that gives measurable context to the event. For example, a photo-sharing app that submits an event when selected photos are uploaded might include a metric for the total amount of data being uploaded, in megabytes. You add metrics to an event as a collection of key-value pairs.

# Reporting Detailed Data Points

Individual data values sent to Mobile Analytics are specified by adding one or more attributes or metrics to an event before you submit it. An event can include any combination of up to 40 total attributes and metrics. As a best practice, we recommend you give a general name to a custom event and specific names to attributes or metrics. For example, using "Item Bought" instead of "Item XYZ" as a custom event name helps keep the report from having too many distinct event names that are hard to read and aggregate. This example demonstrates how to define a custom event.

## Adding Attributes and Metrics in iOS or Android Apps

Attributes and metrics are added to events in iOS and Android apps by creating a custom event object (p. 27) and then adding the required key-value pairs using the `addAttribute` or `addMetrics` methods.


Login Register
Full name
Email
Password
Register
Call event client object to instantiate an event object
Event client object returns event object
Call event object methods to add attributes and metrics
attributes
bonusType=Avatar
metrics
finishedLevel=3
Call event client object to record the event
Mobile App
Mobile SDK
Mobile Analytics
iOS or Android Apps

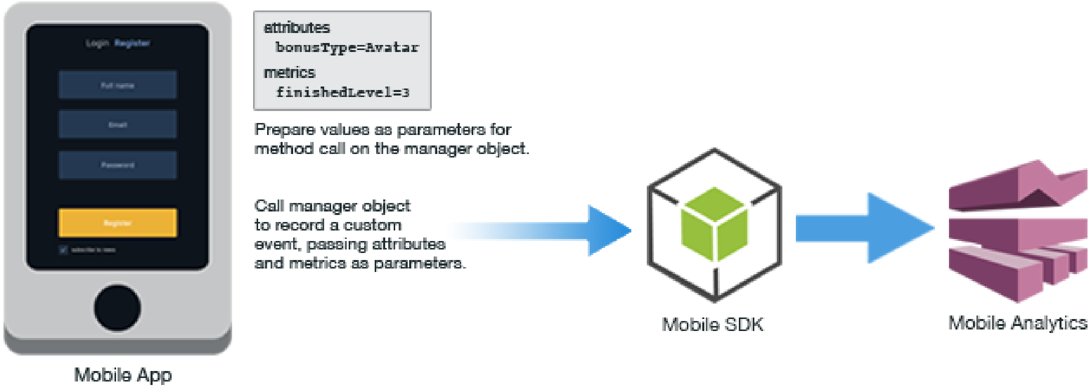
For information about adding attributes and metrics to events in iOS or Android apps, see:

- AWS Mobile SDK iOS Developer Guide
- AWS SDK for iOS Reference

- AWS Mobile SDK Android Developer Guide
- AWS SDK for Android Reference

## Adding Attributes and Metrics in JavaScript Apps

Attributes and metrics are added to custom events in JavaScript apps (p. 27) as parameter values.

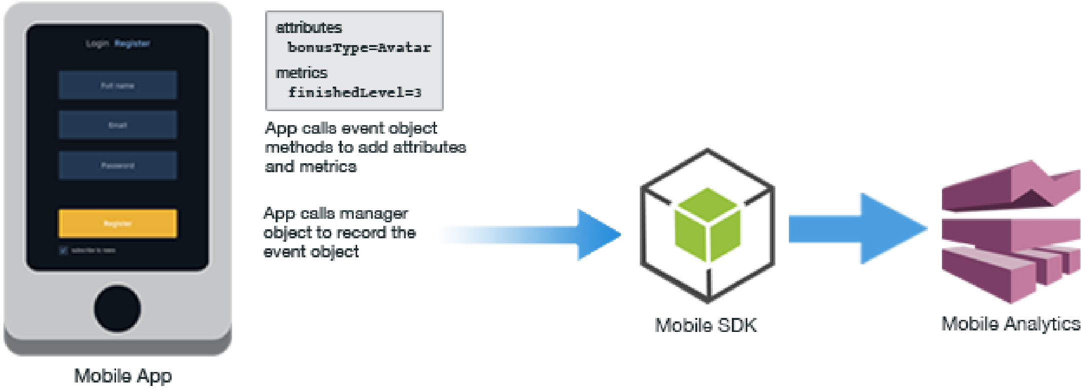

attributes
bonusType=Avatar
metrics
finishedLevel=3
Prepare values as parameters for method call on the manager object.
Call manager object to record a custom event, passing attributes and metrics as parameters.
Mobile SDK
Mobile Analytics
Mobile App

JavaScript Apps

For more information, see Mobile Analytics SDK for JavaScript.

## Adding Attributes and Metrics in Unity or Xamarin Apps

Attributes and metrics are added to events in Unity or Xamarin apps by creating a custom event object and then adding the required key-value pairs using the `addAttribute` or `addMetrics` methods.


attributes
bonusType=Avatar
metrics
finishedLevel=3
App calls event object methods to add attributes and metrics
App calls manager object to record the event object
Mobile SDK
Mobile Analytics
Mobile App

Unity or Xamarin Apps

For information about adding attributes and metrics to events in Unity or Xamarin apps, see:

- AWS Mobile SDK Unity Developer Guide
- AWS SDK for Unity Reference

- AWS Mobile SDK Xamarin Developer Guide
- AWS SDK for Xamarin Reference

# Submitting Events

Each app can have up to 1,500 unique custom events, up to 40 attributes and metrics per custom event, and an infinite number of attribute or metrics values.

# Submitting Events in iOS Apps

When you create event objects in an iOS app, you call the `recordEvent` method on the event client object to record those events in the device's local persistent storage. The client stores a maximum of 5 megabytes of event data. By default, recorded events are submitted to Mobile Analytics at the end of the current session. However you can manually submit events at any time during a session. You can call the `submitEvents` method on the event client object to submit events manually. This removes data from the client file store.

For information about submitting events in iOS apps, see:

- AWS Mobile SDK iOS Developer Guide
- AWS SDK for iOS Reference

# Submitting Events in Android Apps

When you create event objects in an Android app, you call the `recordEvent` method on the event client object to record those events in the device's local persistent storage. By default, recorded events are submitted to Mobile Analytics at the end of the current session. However you can manually submit events at any time during a session. The client will store a maximum of 5 megabytes of event data. You can call the `submitEvents` method on the event client object to submit events manually. This removes data from the client file store.

For information about submitting events in Android apps, see:

- AWS Mobile SDK Android Developer Guide
- AWS SDK for Android Reference

# Submitting Events in JavaScript Apps

When you create event objects in a JavaScript app, you call the `recordEvent` method on the manager object to record those events in the device's local persistent storage. You can call the `submitEvents` method on the manager object to submit events manually . This removes data from the client file store. As long as there is an active session, session events are submitted to Mobile Analytics every 10 seconds.

For information about submitting events in JavaScript apps, see Mobile Analytics SDK for JavaScript.

# Submitting Events in Unity or Xamarin Apps

When you create event objects in a Unity or Xamarin app, you call the `recordEvent` method on the manager object to record those events in the device's local persistent storage. All events are submitted to Mobile Analytics in a background thread.

For information about submitting events in Unity or Xamarin apps, see:

- AWS Mobile SDK Unity Developer Guide
- AWS SDK for Unity Reference

- AWS Mobile SDK Xamarin Developer Guide
- AWS SDK for Xamarin Reference

# Using the Mobile Analytics REST API

To take advantage of Mobile Analytics, your app must include code that records and submits events you want reported in the console.

App developers working on platforms supported by the AWS Mobile SDK will likely use the APIs provided for their platforms. You can use the REST API to integrate with Mobile Analytics without incorporating the AWS Mobile SDK in your app. The REST API lets you submit events from a back-end service.

The topics in this section cover key Mobile Analytics concepts, including Mobile Analytics event types and common tasks you perform with the REST API.

Topics

## Identifying the App in Mobile Analytics

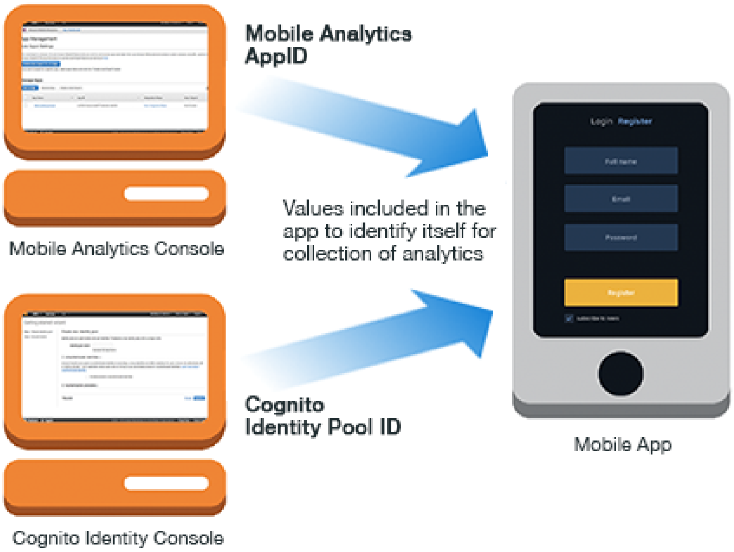
Before you identify your app, you must:

- Add your app in the Mobile Analytics console. (p. 4)
- Create an Amazon Cognito identity pool or use an identity pool created for you by Mobile Analytics.

### Tying Analytics to Users

Before Mobile Analytics can collect and present any analytic data about your app, you must identify it to Mobile Analytics. Once an app has been identified in the service, it can begin to collect data about the users of your app. There are two pieces of information you need so your application can identify itself with Mobile Analytics:

- The Amazon Cognito identity pool created to authenticate users of your application.
- The AppID created in the Mobile Analytics console when you added your app to Mobile Analytics.



When you use the Mobile Analytics REST API, your app provides the AppID as part of the client context header (p. 41) you include when calling the PutEvents action. Your app provides the Amazon Cognito identity pool ID as part of initializing Amazon Cognito Identity.

# Managing Sessions

Before you manage sessions in your app, you must:

- Add your app in the Mobile Analytics console (p. 4)
- Create an Amazon Cognito identity pool or use an identity pool created for you by Mobile Analytics.

## Users Engage in Sessions

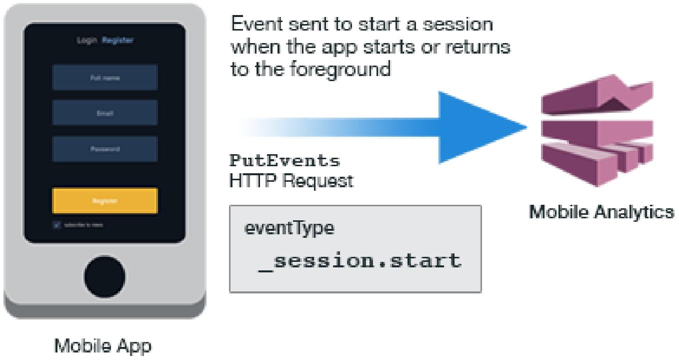
Mobile Analytics reports app usage data based on activity captured in sessions. A *session* is a period of time during which the application is active on the device. A session begins when an app is launched or brought to the foreground, and ends when the app is terminated or goes to the background. To accommodate brief interruptions, like a text message, an inactivity period of up to 5 seconds does not counted as a new session.

## Collecting KPI Data

Mobile Analytics tracks sessions and events submitted during each session to generate the collection of standard charts in the console reporting on key performance indicators (KPIs). Total Daily Sessions shows the number of sessions your app has each day. Average Sessions per Daily Active User shows the mean number of sessions per user per day.

# Starting a Session

When your app starts, a session tracked by Mobile Analytics should begin. The code that initializes the app calls the PutEvents action, sending a session start event.

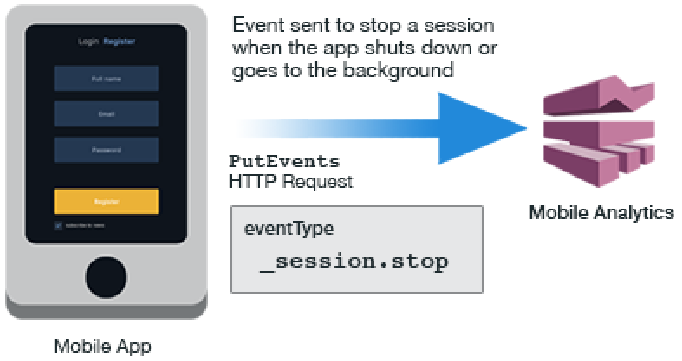


A session start event has an eventType of "`_session.start`" as shown in the following example.

```
POST /2014-06-05/events HTTP/1.1
Host: mobileanalytics.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<access_key>/20140709/us-
east-1/mobileanalytics/aws4_request, SignedHeaders=content-length;content-
type;host;user-agent;x-amz-client-context;x-amz-date;x-amz-security-token;x-
amz-target, Signature=<signature>
User-Agent: <User agent string>
x-amz-Client-Context: {"client":
{"client_id":"<client_id>","app_title":"<app_title>","app_version_name":"<app_version_name>
{},"env":
{"platform":"<platform>","model":"<model>","make":"<make>","platform_version":"<platform_ve
x-amz-security-token: <Security token>
Content-Type: application/json
Content-Length: <Payload size bytes>
Connection: Keep-Alive

{
  "events": [
    {
      "eventType": "_session.start",
      "session": {
                   "startTimestamp": "<ISO 8601 date>",
                   "id": "<session id>"
      },
      "timestamp": "<ISO 8601 date>",
      "attributes": {
                      <optional>
      },
      "metrics": {
                   <optional>
      }
    }
  ]
}
```

# Stopping a Session

When your app closes or is sent to the background, it should end the current session. To end a session, the application code that responds to notification to shut down or pause calls the PutEvents action, sending a session end event.



A session end event has an eventType of "`_session.stop`" as shown in the following example.

```
POST /2014-06-05/events HTTP/1.1
Host: mobileanalytics.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<access_key>/20140709/us-
east-1/mobileanalytics/aws4_request, SignedHeaders=content-length;content-
type;host;user-agent;x-amz-client-context;x-amz-date;x-amz-security-token;x-
amz-target, Signature=<signature>
User-Agent: <User agent string>
x-amz-Client-Context: {"client":
{"client_id":"<client_id>","app_title":"<app_title>","app_version_name":"<app_version_name>
{},"env":
{"platform":"<platform>","model":"<model>","make":"<make>","platform_version":"<platform_ve
x-amz-security-token: <Security token>
Content-Type: application/json
Content-Length: <Payload size bytes>
Connection: Keep-Alive

{
  "eventType": "_session.stop",
  "timestamp": "2014-07-09T03:26:38.719Z",
  "session": {
               "id": "<session id>",
               "duration": 497516,
               "startTimestamp": "2014-07-09T03:17:20.041Z",
               "stopTimestamp": "2014-07-09T03:25:37.557Z"
  },
  "attributes": {},
  "metrics": {}
}
```

# Generating Mobile Analytics Events

Before generating analytics events in your app, you must:

- Add your app in the Mobile Analytics console (p. 4)
- Create an Amazon Cognito identity pool or use an identity pool created for you by Mobile Analytics.
- Start a session (p. 33)

## Types of Analytics Events

Information about user engagement in your app is sent to the Mobile Analytics service using events. Generally speaking, there are two types of mobile analytics events:

- Standard events
- Custom events

Standard events include those you send to start or stop a session. They also include specialized events used to report on monetization activities in the app, such as in-app purchases. Custom events are those you create to monitor activities specific to your app, such as completing a level in a game, posting to social media, or setting particular app preferences. An app can have up to 1,500 unique custom events.

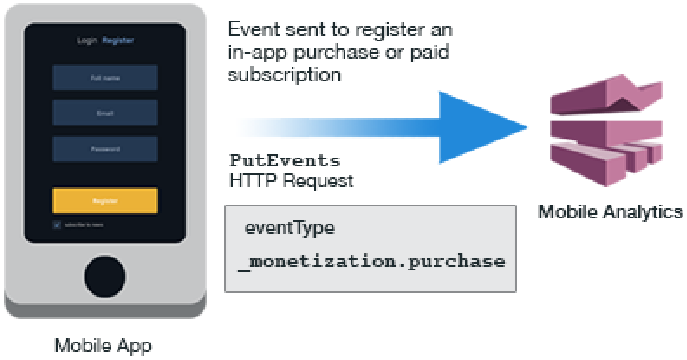
## Session Events

Standard events include those that allow the app to start and end a session (p. 33) in Mobile Analytics. Sessions enable Mobile Analytics to provide the analytics data in the console reports.

To start a session, the app calls the PutEvents action with a session start event. A session start event has an eventType of "`_session.start`". To end a session, the app sends an HTTP PutEvents request with a session end event. A session end event has an eventType of "`_session.stop`".

## Defining a Monetization Event

When a user of your app makes a purchase, the code that handles the purchase calls the PutEvents action to send a monetization event.



A monetization event has an eventType of "`_monetization.purchase`", as shown in the following example. It demonstrates how to monetize an event with a price that does not include the currency symbol.

```
POST /2014-06-05/events HTTP/1.1
Host: mobileanalytics.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<access_key>/20140709/us-
east-1/mobileanalytics/aws4_request, SignedHeaders=content-length;content-
type;host;user-agent;x-amz-client-context;x-amz-date;x-amz-security-token;x-
amz-target, Signature=<signature>
User-Agent: <User agent string>
x-amz-Client-Context: {"client":
{"client_id":"<client_id>","app_title":"<app_title>","app_version_name":"<app_version_name>
{},"env":
{"platform":"<platform>","model":"<model>","make":"<make>","platform_version":"<platform_ve
x-amz-security-token: <Security token>
Content-Type: application/json
Content-Length: <Payload size bytes>
Connection: Keep-Alive

{
  "events": [
    {
      "eventType": "_monetization.purchase",
      "session": {
                   "startTimestamp": "<ISO 8601 date>",
                   "id": "<session id>"
      },
      "timestamp": "<ISO 8601 date>",
      "attributes": {
                      "_currency": "<ISO 4217 currency code>",
                      "_product_id":"<User specified string>"
      },
      "metrics": {
                   "_quantity": <Purchase quantity, defaults to 1>,
                   "_item_price": <Decimal price>
      }
    }
  ]
}
```

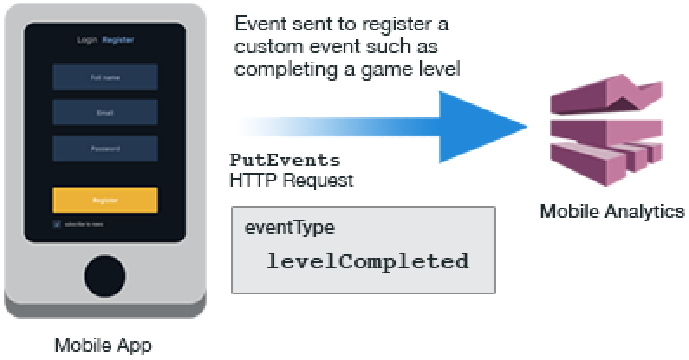
This example demonstrates how to monetize an event with a price that includes the currency symbol. You can use this example with a formatted price.

```
POST /2014-06-05/events HTTP/1.1
Host: mobileanalytics.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<access_key>/20140709/us-
east-1/mobileanalytics/aws4_request, SignedHeaders=content-length;content-
type;host;user-agent;x-amz-client-context;x-amz-date;x-amz-security-token;x-
amz-target, Signature=<signature>
User-Agent: <User agent string>
x-amz-Client-Context: {"client":
{"client_id":"<client_id>","app_title":"<app_title>","app_version_name":"<app_version_name>
{},"env":
{"platform":"<platform>","model":"<model>","make":"<make>","platform_version":"<platform_ve
x-amz-security-token: <Security token>
Content-Type: application/json
Content-Length: <Payload size bytes>
Connection: Keep-Alive
```

```
{
  "events": [
    {
      "eventType": "_monetization.purchase",
      "session": {
                   "startTimestamp": "<ISO 8601 date>",
                   "id": "<session id>"
      },
      "timestamp": "<ISO 8601 date>",
      "attributes": {
                      "_item_price_formatted": "<Price prefixed with currency
 symbol ($1.99)>",
                      "_product_id":"<User specified string>"
      },
      "metrics": {
                   "_quantity": <Purchase quantity, defaults to 1>
      }
    }
  ]
}
```

# Creating a Custom Event

In addition to the standard events, you can also define your own custom events to report on types of interaction specific to your app, such as finishing levels of a game.



You assign an eventType to your custom event. As a best practice, we recommend you give a general name to a custom event and specific names to attributes or metrics. For example, using "Item Bought" instead of "Item XYZ" as a custom event name helps keep the report from having too many distinct event names that are hard to read and aggregate. This example demonstrates how to define a custom event.

```
POST /2014-06-05/events HTTP/1.1
Host: mobileanalytics.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<access_key>/20140709/us-
east-1/mobileanalytics/aws4_request, SignedHeaders=content-length;content-
type;host;user-agent;x-amz-client-context;x-amz-date;x-amz-security-token;x-
amz-target, Signature=<signature>
User-Agent: <User agent string>
x-amz-Client-Context: {"client":
{"client_id":"<client_id>","app_title":"<app_title>","app_version_name":"<app_version_name>
```

```
{},"env":
{"platform":"<platform>","model":"<model>","make":"<make>","platform_version":"<platform_ve
x-amz-security-token: <Security token>
Content-Type: application/json
Content-Length: <Payload size bytes>
Connection: Keep-Alive

{
  "events": [
    {
      "eventType": "sampleEvent",
      "timestamp": "2014-07-09T03:17:20.041Z",
      "session": {
                    "id": "<Session id>",
                    "startTimestamp": "2014-07-09T03:15:31.041Z"
      },
      "attributes": {},
      "metrics": {}
    },
    {
      "eventType": "otherEvent",
      "timestamp": "2014-07-09T03:17:42.772Z",
      "session": {
                    "id": "<Session id>",
                    "startTimestamp": "2014-07-09T03:15:31.041Z"
                  },
      "attributes": {
                       "customAttribute": "someValue"
      },
      "metrics": {
                    "count": 1
      }
     }
  ]
}
```

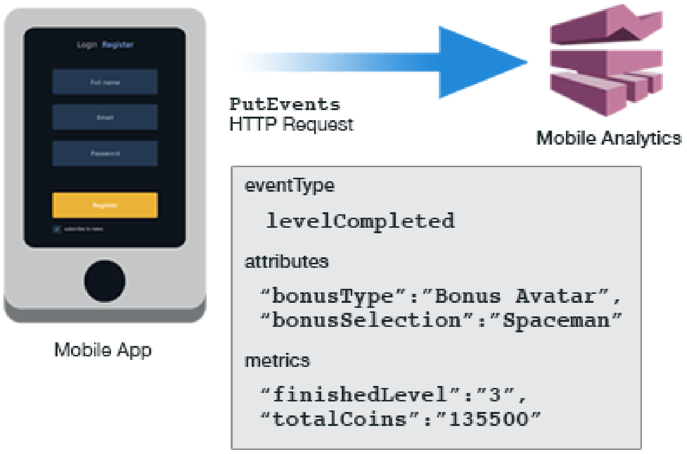
# Adding Attributes and Metrics

Before you add attributes or metrics to events in your app, you must:

- Add your app in the Mobile Analytics console (p. 4)
- Create an Amazon Cognito identity pool or use an identity pool created for you by Mobile Analytics.
- Start a session (p. 33)
- Create an event (p. 36)

## Reporting Detailed Data Points

You specify individual data values sent to Mobile Analytics by adding one or more attributes or metrics to an event before you submit it. An event can include any combination of up to 40 total attributes and metrics.



As a best practice, we recommend that names given to custom event names be broad and those given to attributes or metrics be specific. For example, using "Item Bought" instead of "Item XYZ" as the custom event name helps keep the report from having too many distinct event names that are hard to read and aggregate.

# Adding Attributes

*Attributes* are data that provide context for the submitted event. For example, a game that submits an event to Mobile Analytics when the player collects a power-up bonus might include an attribute named for the type of bonus collected.

You add attributes to an event as a collection of key-value pairs. When you use the REST API, you add each key-value pair to the attributes section of the event and then submit the event with the PutEvents action. For example:

```
"attributes": {
   "bonusType": "Bonus Avatar",
   "bonusSelection": "Spaceman"
}
```

# Adding Metrics

*Metrics* are data that provides measurable context to the event. For example, a photo-sharing app that submits an event when selected photos are uploaded might include a metric for the total amount of data being uploaded, in megabytes.

You add metrics to an event as a collection of key-value pairs. When you use the REST API, you add each key-value pair to the metrics section of the event and then submit the event with the PutEvents action. For example:

```
"metrics": {
   "finishedLevel": "3",
   "totalCoins": "135500"
}
```

# Providing a Client Context

Before you use a client context to submit events from your app, you must:

- Add your app in the Mobile Analytics console (p. 4)
- Create an Amazon Cognito identity pool or use an identity pool created for you by Mobile Analytics.

## Giving Details of Application Context

Data in a client context provides information about the client interaction with an application service; it describes the app and the environment in which the app runs. When your mobile app communicates with Mobile Analytics, your app must provide the client context for the application and device. To do this, create a JSON-formatted string with the data to include as a header (p. 127) when using the PutEvents action.

Client context contains at least two sets of properties:

- Client properties
- Environment properties

In addition to these two sets of properties, a client context can also contain custom values that provide context about the app or environment.

The client context is where you provide the values required by Mobile Analytics to identify your app. In the services section, you must include `mobile_analytics` as a key with the AppID you generated for your app in the Mobile Analytics console as the value.

# Signing Requests

The method to use to sign your request depends on where the request originates. If the request comes from a server, use AWS Identity and Access Management. If the request comes from a mobile device, use Amazon Cognito or IAM.

All requests to the Amazon Mobile Analytics REST API must be signed and the following headers must be present.

```
Accept: application/hal+json
X-Amz-Date: <TIMESTAMP in format of YYYYMMDDTHHmmSSZ e.g. 20151208T063435Z>
Authorization: AWS4-HMAC-SHA256 Credential=<YOUR ACCESS KEY ID>/20151208/us-
east-1/mobileanalytics/aws4_request,
 SignedHeaders=accept;host;x-amz-date, Signature=<Sig V4 signature>
```

Both IAM and Amazon Cognito support signature version 4. For more information, see Signature Version 4 Signing Process.

# Submitting Events

Before you submit events from your app, you must:

- Add your app in the Mobile Analytics console (p. 4)
- Create an Amazon Cognito identity pool or use an identity pool created for you by Mobile Analytics.
- Start a session (p. 33)

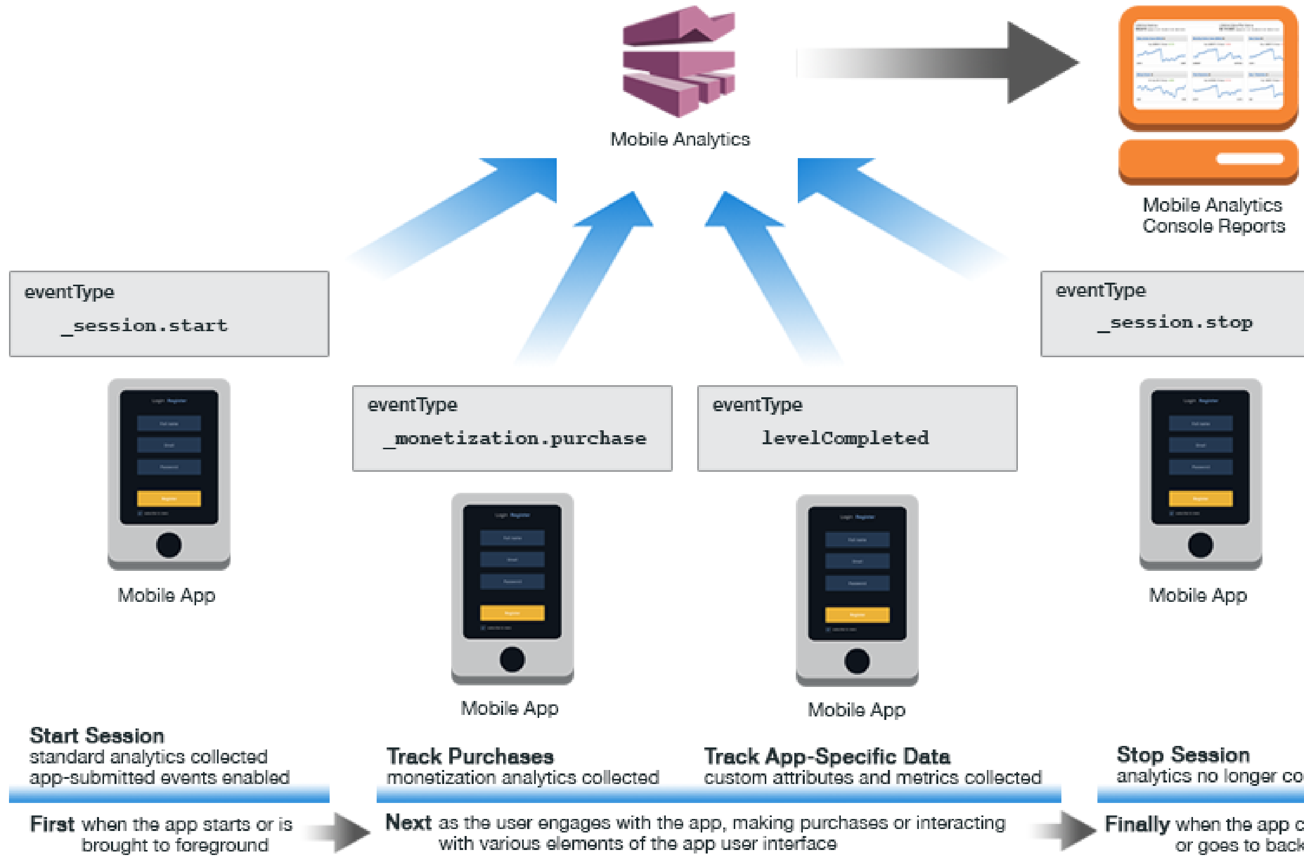
- Create an event to submit (p. 36)
- Add relevant attributes and metrics to events (p. 39)

## Event-Based Interaction

The interaction between a mobile app and Mobile Analytics takes place through a single PutEvents action in the REST API. You use the PutEvents action to:

- Start and stop sessions to collect the standard analytics displayed in the console reports.
- Send monetization data.
- Collect custom analytics specific to your app.

The following diagram shows how a mobile app sends events to Mobile Analytics at different points in the execution of the app. This interaction enables the collection of data used to produce the analytics reports.


Mobile Analytics
Mobile Analytics Console Reports
eventType
_session.start
Mobile App
eventType
_monetization.purchase
Mobile App
eventType
levelCompleted
Mobile App
eventType
_session.stop
Mobile App
Start Session
standard analytics collected
app-submitted events enabled
Track Purchases
monetization analytics collected
Track App-Specific Data
custom attributes and metrics collected
Stop Session
analytics no longer co
First when the app starts or is brought to foreground
Next as the user engages with the app, making purchases or interacting with various elements of the app user interface
Finally when the app c
or goes to back

You can have up to 1,500 unique custom events per app, up to 40 attributes and metrics per custom event, and an infinite number of attribute or metrics values.

# Querying Analytics Data

Mobile Analytics provides actions in its REST API to enable querying the accumulated analytics for your apps.

Topics

# Setting IAM Policy

To query the analytics data, you must have the correct permissions set up for the IAM user or role accessing Mobile Analytics. You can define the necessary permissions using the IAM policy generator or by adding the following policy to grant access to both standard and financial metrics for an app.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "mobileanalytics:GetReports",
                "mobileanalytics:GetFinancialReports"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
```

# Getting a List of Apps

This query returns a list of all your apps. The URL for this query is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps
```

## Example Response

Here is an example of the JSON returned in the response. The list of apps returned is in the item attribute.

```
{
    "_links" : {
        "curies" : [
            {
                "href" :  "http://docs.aws.amazon.com/mobileanalytics/latest/
ug/server-reference.html#event:{rel},
                "name" :  "app",
                "templated" :  true
            }
        ],
        "self" : {
            "href" :  "/2014-06-05/apps?page-size=3",
            "name" :  "apps",
            "title" :  "Apps Collection"
        },
        "app:by-id" : [
            {
                "href" :  "/2014-06-05/apps/{appId}",
                "name" :  "app-by-id",
                "title" :  "App By Id",
                "templated" :  true
            }
        ],
        "app:create" : [
            {
                "href" :  "/2014-06-05/apps"
            }
        ],
        "item" : [
           {
           "href" : "/2014-06-05/apps/<YOUR APP ID>",
           "name" :  "<YOUR APP NAME>",
           "title" :  "app_1"
            },
            {
            "href" :  "/2014-06-05/apps/<YOUR APP ID>",
            "name" :  "<YOUR APP NAME>",
            "title" :  "app_2"
            }
            ],
        "up" : [
        {
        "href" :  "/2014-06-05",
        "name" :  "2014-06-05"
        }
        ]
    },
    itemCount :  3,
    pageCount :  1
}
```

## Querying Key Peformance Indicators (KPIs)

Using the REST API, you can query your analytics data to obtain details about a variety of key performance indicators (KPIs).

Topics

### Querying Lifetime User Count

This KPI returns the count of all users and devices that have ever used your app. The URL for this KPI is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/new-users/lifetime-count
```

### Example Response

Here is an example of the JSON returned in the response. The value in the rows attribute contains the KPI value.

```
{
 "_links" : {
  "self" : {
   "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/new-users/lifetime-count",
   "name" : "lifetime-count"
  },
  "example" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/new-users/lifetime-count",
    "name" : "lifetime-count-all-platforms",
    "title" : "Lifetime Count All Platforms"
   }
  ],
  "up" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/new-users",
    "name" : "new-users"
   }
  ]
 },
 "columns" : {
  "NewUsers" : {
   "colIndex" : 0,
   "type" : "Quantitative",
   "dataType" : "Double",
   "unit" : null,
   "name" : "NewUsers",
   "displayName" : "New Users"
  }
 },
 "rows" : [[132456.0]],
 "scope" : {
  "appId" : "<YOUR APP ID>",
  "kpiName" : "new-users",
  "aggregate" : "lifetime-count"
 }
}
```

## Querying Active User Counts

These KPIs return the number of unique users/devices that have launched your app in a particular day or month as well as the number of first-time users of your app in that time period.

The URL for the **Daily Active Users** (DAU) KPI is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/dau/count
```

The URL for the **Monthly Active Users** (MAU) KPI is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/mau/count
```

The URL for the **New Users** KPI is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/new-users/count
```

## Example Response

Here is an example of the JSON returned in the response for the daily active users query. The value in the rows attribute contains the KPI values by day.

```
{
 "_links" : {
  "self" : {
   "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/dau/count?
start=2014-11-20&end=2014-12-01",
   "name" : "count"
  },
  "example" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/dau/count?start=7-days-
ago&end=today",
    "name" : "last-7-days-all-platforms",
    "title" : "Last 7 Days All Platforms"
   }, {
    "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/dau/count?start=30-days-
ago&end=today",
    "name" : "last-30-days-all-platforms",
    "title" : "Last 30 Days All Platforms"
   }, {
    "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/dau/count?start=60-days-
ago&end=today",
    "name" : "last-60-days-all-platforms",
    "title" : "Last 60 Days All Platforms"
   }
  ],
  "up" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/dau",
    "name" : "dau"
   }
  ]
 },
 "columns" : {
  "day" : {
   "colIndex" : 0,
   "type" : "Qualitative",
   "dataType" : "ISODateTime",
   "unit" : null,
   "name" : "day",
   "displayName" : "Date"
  },
  "DAU" : {
   "colIndex" : 1,
   "type" : "Quantitative",
   "dataType" : "Double",
   "unit" : null,
   "name" : "DAU",
   "displayName" : "Daily Active Users"
  }
 },
 "rows" :
  [["2014-11-20", 342.0],
  ["2014-11-21", 344.0],
  ["2014-11-22", 355.0],
  ["2014-11-23", 311.0],
  ["2014-11-24", 301.0],
  ["2014-11-25", 423.0],
  ["2014-11-26", 198.0],
  ["2014-11-27", 234.0],
  ["2014-11-28", 432.0],
  ["2014-11-29", 389.0],
  ["2014-11-30", 405.0],
  ["2014-12-01", 416.0]],
 "scope" : {
  "appId" : "<YOUR APP ID>",
  "kpiName" : "dau",
  "aggregate" : "count"
 }
}
```

## Querying Session Count

This KPI returns the number of sessions for your app. The URL for this KPI is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/sessions/count
```

### Example Response

Here is an example of the JSON returned in the response. The value in the rows attribute contains the KPI value.

```
{
 "_links" : {
  "self" : {
   "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/sessions/count?
start=2014-11-20&end=2014-12-01",
   "name" : "count"
  },
  "example" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/sessions/count?start=7-
days-ago&end=today",
    "name" : "last-7-days-all-platforms",
    "title" : "Last 7 Days All Platforms"
   }, {
    "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/sessions/count?start=30-
days-ago&end=today",
    "name" : "last-30-days-all-platforms",
    "title" : "Last 30 Days All Platforms"
   }, {
    "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/sessions/count?start=60-
days-ago&end=today",
    "name" : "last-60-days-all-platforms",
    "title" : "Last 60 Days All Platforms"
   }
  ],
  "up" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/sessions",
    "name" : "sessions"
   }
  ]
 },
 "columns" : {
  "day" : {
   "colIndex" : 0,
   "type" : "Qualitative",
   "dataType" : "ISODateTime",
   "unit" : null,
   "name" : "day",
   "displayName" : "Date"
  },
  "Sessions" : {
   "colIndex" : 1,
   "type" : "Quantitative",
   "dataType" : "Double",
   "unit" : null,
   "name" : "Sessions",
   "displayName" : "Sessions"
  }
 },
 "rows" : [ ["2014-11-20", 403.0],
    ["2014-11-21", 455.0],
    ["2014-11-22", 453.0],
    ["2014-11-23", 432.0],
    ["2014-11-24", 503.0],
    ["2014-11-25", 564.0],
    ["2014-11-26", 574.0],
    ["2014-11-27", 532.0],
    ["2014-11-28", 495.0],
    ["2014-11-29", 486.0],
    ["2014-11-30", 529.0],
    ["2014-12-01", 520.0]],
 "scope" : {
  "appId" : "<YOUR APP ID>",
  "kpiName" : "sessions",
  "aggregate" : "count"
 }
```

## Querying Revenue Data

These KPIs return revenue data for an app.

The URL for **daily revenue** KPI is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/daily-revenue/sum
```

The URL for **Daily Paying Active Users** (PDAU) KPI is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/pdau/count
```

The URL for **Monthly Paying Active Users** (PMAU) KPI is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/pmau/count
```

## Querying Retention Data

These KPIs return retention data for an app.

The URLs for **week 1, week 2, and week 3 retention** KPIs are:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/week-1-retention/count
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/week-2-retention/count
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/week-3-retention/count
```

The URLs for **day 1, day 3, day 5, and day 7 retention** KPIs are:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/day-1-retention/count
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/day-3-retention/count
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/day-5-retention/count
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/day-7-retention/count
```

### Example Response

Here is an example of the JSON returned in the response for the day 7 retention count query. The value in the rows attribute contains the KPI values by day.

```
{
 "_links" : {
  "self" : {
   "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/day-7-retention-per-user/
count?start=2015-12-01&end=2015-12-08",
   "name" : "count"
  },
  "example" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/day-7-retention-per-user/
count?start=7-days-ago&end=today",
    "name" : "last-7-days-all-platforms",
    "title" : "Last 7 Days All Platforms"
   }, {
    "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/day-7-retention-per-user/
count?start=30-days-ago&end=today",
    "name" : "last-30-days-all-platforms",
    "title" : "Last 30 Days All Platforms"
   }, {
    "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/day-7-retention-per-user/
count?start=60-days-ago&end=today",
    "name" : "last-60-days-all-platforms",
    "title" : "Last 60 Days All Platforms"
   }
  ],
  "up" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/kpis/day-7-retention-per-user",
    "name" : "day-7-retention-per-user"
   }
  ]
 },
 "columns" : {
  "day" : {
   "colIndex" : 0,
   "type" : "Qualitative",
   "dataType" : "ISODateTime",
   "unit" : null,
   "name" : "day",
   "displayName" : "Date"
  },
  "Day7RetentionPerUser" : {
   "colIndex" : 1,
   "type" : "Quantitative",
   "dataType" : "Double",
   "unit" : null,
   "name" : "Day7RetentionPerUser",
   "displayName" : "Day 7 Retention Per User"
  }
 },
 "rows" : [ ["2015-12-01", 0.0],
    ["2015-12-02", 0.0],
    ["2015-12-03", 0.0],
    ["2015-12-04", 0.0],
    ["2015-12-05", 0.0],
    ["2015-12-06", 0.0],
    ["2015-12-07", 0.0],
    ["2015-12-08", 0.0]],
 "scope" : {
  "appId" : "<YOUR APP ID>",
  "kpiName" : "day-7-retention-per-user",
  "aggregate" : "count"
 },
 "weightedAverage" : 0.0
}
```

# Querying Custom Events

Using the REST API, you can query your analytics data to obtain details about custom events submitted by your app. The following topics show how to obtain various data about the custom events your app submits to Mobile Analytics.

Topics

- Listing Custom Event Names (p. 51)
- Listing Custom Event Attribute Names (p. 52)
- Listing Custom Event Attribute Values (p. 53)
- Listing Custom Event Metrics Names (p. 54)

- Querying Counts of Event Occurrences (p. 55)
- Querying Counts of Event Occurrences with Specified Attribute Name (p. 57)
- Querying Counts of Event Occurrences with Specified Attribute Value (p. 59)
- Querying Counts of Event Occurrences with Specified Metric Name (p. 60)
- Querying Daily Metric Aggregates for Custom Events (p. 62)

## Listing Custom Event Names

This query lists the valid custom event names for an app. The URL for this query is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/events
```

### Example Response

Here is an example of the JSON returned in the response of a query for the names of the custom events for an app.

```
{

    _links : {
        curies : [
            {
                href :  http://docs.aws.amazon.com/mobileanalytics/latest/ug/
server-reference.html#event:{rel},
                name :  event,
                templated :  true
            }
        ],
        self : {
            href :  /2014-06-05/apps/<YOUR APP ID>/events,
            name :  events,
            title :  Events Collection
        },
        event:by-name : [
            {
                href :  /2014-06-05/apps/<YOUR APP ID>/events/{eventName},
                name :  event-by-name,
                title :  Event By Name,
                templated :  true
            }
        ],
        item : [
            {
                href :  /2014-06-05/apps/<YOUR APP ID>/events/LevelComplete,
                name :  LevelComplete
            },
            {
                href :  /2014-06-05/apps/<YOUR APP ID>/events/LevelUp,
                name :  LevelUp
            }
        ],
        up : [
            {
                href :  /2014-06-05/apps/<YOUR APP ID>,
                name :  <YOUR APP ID>
            }
        ]
    },
    appId :  <YOUR APP ID>,
    itemCount :  2,
    pageCount :  1

}
```

## Listing Custom Event Attribute Names

This query lists the valid attribute names for a custom event. The URL for this query is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/events/<CUSTOM EVENT NAME>/attributes
```

### Example Response

Here is an example of the JSON returned in the response of a query for attribute names for a "LevelComplete" custom event. It returns three example attribute names in the item attribute.

```
{
 "_links" : {
  "curies" : [{
    "href" : "http://docs.aws.amazon.com/mobileanalytics/latest/ug/server-
reference.html#event:{rel},
    "name" : "attribute",
    "templated" : true
   }
  ],
  "self" : {
   "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes",
   "name" : "attributes",
   "title" : "Attributes Collection"
  },
  "attribute:by-name" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
{attributeName}",
    "name" : "attribute-by-name",
    "title" : "Attribute By Name",
    "templated" : true
   }
  ],
  "item" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
Difficulty",
    "name" : "Difficulty"
   }, {
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
EndState",
    "name" : "EndState"
   }, {
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
LevelName",
    "name" : "LevelName"
   }
  ],
  "up" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete",
    "name" : "LevelComplete"
   }
  ]
 },
 "appId" : "<YOUR APP ID>",
 "eventName" : "LevelComplete",
 "itemCount" : 3,
 "pageCount" : 1
}
```

## Listing Custom Event Attribute Values

This query returns the valid attribute values for a specific custom event and attribute. The URL for this query is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/events/<CUSTOM EVENT NAME>/attributes/<ATTRIBUTE NAME>/values
```

### Example Response

Here is an example of the JSON returned in the response of a query for values of the "EndState" attribute of the "LevelComplete" custom event. It returns two example attribute values in the item attribute.

```
{
 "_links" : {
  "curies" : [{
    "href" : "http://docs.aws.amazon.com/mobileanalytics/latest/ug/server-
reference.html#event:{rel},
    "name" : "attributevalue",
    "templated" : true
   }
  ],
  "self" : {
   "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
EndState/values",
   "name" : "values",
   "title" : "Attribute Values Collection"
  },
  "attributevalue:by-name" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
EndState/values/{attributeValueName}",
    "name" : "attribute-value-by-name",
    "templated" : true
   }
  ],
  "item" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
EndState/values/Lose",
    "name" : "Lose"
   }, {
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
EndState/values/Win",
    "name" : "Win"
   }
  ],
  "up" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
EndState",
    "name" : "EndState"
   }
  ]
 },
 "appId" : "<YOUR APP ID>",
 "attributeName" : "EndState",
 "eventName" : "LevelComplete",
 "itemCount" : 2,
 "pageCount" : 1
}
```

## Listing Custom Event Metrics Names

This query lists the valid names of metrics available for a custom event. The URL for this query is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/events/<CUSTOM EVENT NAME>/metrics<ATTRIBUTE NAME>/values/<ATTRIBUTE
 NAME>/count
```

### Example Response

Here is an example of the JSON returned in the response of a query for the names of the metrics available on the `LevelComplete` event. The URL for this query is `https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/c4e4c37a4f304496a03d69951126af4c/events/LevelComplete/attributes/EndState/values/Lose/count`.

```
{
 "_links" : {
  "self" : {
   "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
EndState/values/Lose/count?start=2015-12-01&end=2015-12-08",
   "name" : "count"
  },
  "example" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
EndState/values/Lose/count?start=7-days-ago&end=today",
    "name" : "last-7-days-all-platforms",
    "title" : "Last 7 Days All Platforms"
   }, {
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
EndState/values/Lose/count?start=30-days-ago&end=today",
    "name" : "last-30-days-all-platforms",
    "title" : "Last 30 Days All Platforms"
   }, {
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
EndState/values/Lose/count?start=60-days-ago&end=today",
    "name" : "last-60-days-all-platforms",
    "title" : "Last 60 Days All Platforms"
   }
  ],
  "up" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
EndState/values/Lose",
    "name" : "Lose"
   }
  ]
 },
 "columns" : {
  "day" : {
   "colIndex" : 0,
   "type" : "Qualitative",
   "dataType" : "ISODateTime",
   "unit" : null,
   "name" : "day",
   "displayName" : "Date"
  },
  "count" : {
   "colIndex" : 1,
   "type" : "Quantitative",
   "dataType" : "Double",
   "unit" : null,
   "name" : "count",
   "displayName" : "Count"
  }
 },
 "rows" : [ ["2015-12-01", 98.0],
    ["2015-12-02", 33.0],
    ["2015-12-03", 37.0],
    ["2015-12-04", 89.0],
    ["2015-12-05", 65.0],
    ["2015-12-06", 45.0],
    ["2015-12-07", 77.0],
    ["2015-12-08", 64.0]],
 "scope" : {
  "attributeValue" : "Lose",
  "appId" : "<YOUR APP ID>",
  "eventName" : "LevelComplete",
  "attributeName" : "EndState",
  "aggregate" : "count"
 }
}
```

## Querying Counts of Event Occurrences

To query the number of occurrences of custom events you must know the name of the custom event. The URL to query counts of custom events is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/events/<CUSTOM EVENT NAME>/count
```

### Example Response

Here is an example of the JSON returned in the response of a query for a "LevelComplete" custom event. The value in the rows attribute contains the KPI value.

```
{
"_links" : {
 "self" : {
  "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/count?
start=2015-12-01&end=2015-12-08",
  "name" : "count"
 },
 "example" : [{
   "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/count?
start=7-days-ago&end=today",
   "name" : "last-7-days-all-platforms",
   "title" : "Last 7 Days All Platforms"
  }, {
   "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/count?
start=30-days-ago&end=today",
   "name" : "last-30-days-all-platforms",
   "title" : "Last 30 Days All Platforms"
  }, {
   "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/count?
start=60-days-ago&end=today",
   "name" : "last-60-days-all-platforms",
   "title" : "Last 60 Days All Platforms"
  }
 ],
 "up" : [{
   "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete",
   "name" : "LevelComplete"
  }
 ]
},
"columns" : {
 "day" : {
  "colIndex" : 0,
  "type" : "Qualitative",
  "dataType" : "ISODateTime",
  "unit" : null,
  "name" : "day",
  "displayName" : "Date"
 },
 "count" : {
  "colIndex" : 1,
  "type" : "Quantitative",
  "dataType" : "Double",
  "unit" : null,
  "name" : "count",
  "displayName" : "Count"
 }
},
"rows" : [ ["2015-12-01", 234.0],
   ["2015-12-02", 432.0],
   ["2015-12-03", 543.0],
   ["2015-12-04", 324.0],
   ["2015-12-05", 446.0],
   ["2015-12-06", 394.0],
   ["2015-12-07", 236.0],
   ["2015-12-08", 846.0]],
"scope" : {
 "appId" : "<YOUR APP ID>",
 "eventName" : "LevelComplete",
 "aggregate" : "count"
}
}
```

## Querying Counts of Event Occurrences with Specified Attribute Name

This query returns the number of event occurrences containing a specific attribute name. The URL for this query is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/events/<CUSTOM EVENT NAME>/attributes/<ATTRIBUTE NAME>/count
```

### Example Response

Here is an example of the JSON returned in the response of a query for the number of event occurrences containing a specified attribute name. This example shows the response for a request for the daily number of "LevelComplete" events where the "EndState" attribute had a "Lose" value:

```
{

    _links : {
        self : {
            href :  /2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/
attributes/EndState/count,
            name :  count
        },
        example : [
            {
                href :  /2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/
attributes/EndState/count?platform=all&start=7-days-ago&end=today&group-
by=none,
                name :  last-7-days-all-platforms,
                title :  Last 7 Days All Platforms
            },
            {
                href :  /2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/
attributes/EndState/count?platform=all&start=30-days-ago&end=today&group-
by=none,
                name :  last-30-days-all-platforms,
                title :  Last 30 Days All Platforms
            },
            {
                href :  /2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/
attributes/EndState/count?platform=&start=60-days-ago&end=today&group-
by=none,
                name :  last-60-days-all-platforms,
                title :  Last 60 Days All Platforms
            }
        ],
        up : [
            {
                href :  /2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/
attributes/EndState,
                name :  EndState
            }
        ]
    },
    columns : {
        day : {
            colIndex :  0,
            type :  Qualitative,
            dataType :  ISODateTime,
            unit :  null,
            name :  day,
            displayName :  Date
        },
        count : {
            colIndex :  1,
            type :  Quantitative,
            dataType :  Double,
            unit :  null,
            name :  count,
            displayName :  Count
        }
    },
    rows : [
        [
            2015-11-29,
            134
        ],
        [
            2015-11-30,
            175
        ],
        [
            2015-12-01,
            190
        ],
        [
            2015-12-02,
            143
        ],
        [
            2015-12-03,
            212
        ],
        [
            2015-12-04,
            254
        ],
        [
            2015-12-05,
            237
        ],
        [
            2015-12-06,
            275
        ],
        [
            2015-12-07,
            289
        ],
        [
            2015-12-08,
            282
        ]
    ],
    scope : {
        appId :  <YOUR APP ID>,
        eventName :  LevelComplete,
        attributeName :  EndState,
        aggregate :  count
    }

}
```

## Querying Counts of Event Occurrences with Specified Attribute Value

This query returns the number of events containing a specific attribute value. The URL for this query is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/events/<CUSTOM EVENT NAME>/attributes/<ATTRIBUTE NAME>/values/<ATTRIBUTE
 NAME>/count
```

### Example Response

Here is an example of the JSON returned in the response of a query for the daily number of `LevelComplete` events where the `EndState` attribute has a `Lose` value.

```
{
 "_links" : {
  "self" : {
   "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
EndState/values/Lose/count?start=2015-12-01&end=2015-12-08",
   "name" : "count"
  },
  "example" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
EndState/values/Lose/count?start=7-days-ago&end=today",
    "name" : "last-7-days-all-platforms",
    "title" : "Last 7 Days All Platforms"
   }, {
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
EndState/values/Lose/count?start=30-days-ago&end=today",
    "name" : "last-30-days-all-platforms",
    "title" : "Last 30 Days All Platforms"
   }, {
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
EndState/values/Lose/count?start=60-days-ago&end=today",
    "name" : "last-60-days-all-platforms",
    "title" : "Last 60 Days All Platforms"
   }
  ],
  "up" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/attributes/
EndState/values/Lose",
    "name" : "Lose"
   }
  ]
 },
 "columns" : {
  "day" : {
   "colIndex" : 0,
   "type" : "Qualitative",
   "dataType" : "ISODateTime",
   "unit" : null,
   "name" : "day",
   "displayName" : "Date"
  },
  "count" : {
   "colIndex" : 1,
   "type" : "Quantitative",
   "dataType" : "Double",
   "unit" : null,
   "name" : "count",
   "displayName" : "Count"
  }
 },
 "rows" : [ ["2015-12-01", 98.0],
    ["2015-12-02", 33.0],
    ["2015-12-03", 37.0],
    ["2015-12-04", 89.0],
    ["2015-12-05", 65.0],
    ["2015-12-06", 45.0],
    ["2015-12-07", 77.0],
    ["2015-12-08", 64.0]],
 "scope" : {
  "attributeValue" : "Lose",
  "appId" : "<YOUR APP ID>",
  "eventName" : "LevelComplete",
  "attributeName" : "EndState",
  "aggregate" : "count"
 }
}
```

## Querying Counts of Event Occurrences with Specified Metric Name

This query returns the number of event occurrences containing a specific metric name. The URL for this query is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/events/<CUSTOM EVENT NAME>/metrics/<METRIC NAME>/count
```

### Example Response

Here is an example of the JSON returned in the response of a query of the average of the `TimeToComplete` metric of the `LevelComplete` custom event. The query URL used was `https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/metrics/TimeToComplete/avg` .

```
{
 "_links" : {
  "self" : {
   "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/metrics/
TimeToComplete/avg?start=2015-12-01&end=2015-12-08",
   "name" : "avg"
  },
  "example" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/metrics/
TimeToComplete/avg?start=7-days-ago&end=today",
    "name" : "last-7-days-all-platforms",
    "title" : "Last 7 Days All Platforms"
   }, {
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/metrics/
TimeToComplete/avg?start=30-days-ago&end=today",
    "name" : "last-30-days-all-platforms",
    "title" : "Last 30 Days All Platforms"
   }, {
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/metrics/
TimeToComplete/avg?start=60-days-ago&end=today",
    "name" : "last-60-days-all-platforms",
    "title" : "Last 60 Days All Platforms"
   }
  ],
  "up" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/metrics/
TimeToComplete",
    "name" : "TimeToComplete"
   }
  ]
 },
 "columns" : {
  "day" : {
   "colIndex" : 0,
   "type" : "Qualitative",
   "dataType" : "ISODateTime",
   "unit" : null,
   "name" : "day",
   "displayName" : "Date"
  },
  "avg" : {
   "colIndex" : 1,
   "type" : "Quantitative",
   "dataType" : "Double",
   "unit" : null,
   "name" : "avg",
   "displayName" : "Avg"
  }
 },
 "rows" : [
    ["2015-12-01", 32.2],
    ["2015-12-02", 35.4],
    ["2015-12-03", 32.5],
    ["2015-12-04", 40.3],
    ["2015-12-05", 36.6],
    ["2015-12-06", 31.3],
    ["2015-12-07", 35.2],
    ["2015-12-08", 34.7]],
 "scope" : {
  "metricName" : "LevelComplete",
  "appId" : "<YOUR APP ID>",
  "eventName" : "LevelComplete",
  "aggregate" : "avg"
 },
 "weightedAverage" : 34.8
}
```

## Querying Daily Metric Aggregates for Custom Events

These queries return daily metric aggregates for your app.

The URL for the **average** query is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/events/<CUSTOM EVENT NAME>/metrics/<METRIC NAME>/avg
```

The URL for the **sum** query is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/events/<CUSTOM EVENT NAME>/metrics/<METRIC NAME>/sum
```

The URL for the **minimum** query is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/events/<CUSTOM EVENT NAME>/metrics/<METRIC NAME>/min
```

The URL for the **maximum** query is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/events/<CUSTOM EVENT NAME>/metrics/<METRIC NAME>/max
```

The URL for the **count** query is:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/events/<CUSTOM EVENT NAME>/metrics/<METRIC NAME>/count
```

The count metric contains the number of event occurrences where the metric was specified on the event.

### Example Response

Here is an example of the JSON returned in the response of a query of the average of the `TimeToComplete` metric of the `LevelComplete` custom event. The query URL used was `https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/metrics/TimeToComplete/avg` .

```
{
 "_links" : {
  "self" : {
   "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/metrics/
TimeToComplete/avg?start=2015-12-01&end=2015-12-08",
   "name" : "avg"
  },
  "example" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/metrics/
TimeToComplete/avg?start=7-days-ago&end=today",
    "name" : "last-7-days-all-platforms",
    "title" : "Last 7 Days All Platforms"
   }, {
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/metrics/
TimeToComplete/avg?start=30-days-ago&end=today",
    "name" : "last-30-days-all-platforms",
    "title" : "Last 30 Days All Platforms"
   }, {
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/metrics/
TimeToComplete/avg?start=60-days-ago&end=today",
    "name" : "last-60-days-all-platforms",
    "title" : "Last 60 Days All Platforms"
   }
  ],
  "up" : [{
    "href" : "/2014-06-05/apps/<YOUR APP ID>/events/LevelComplete/metrics/
TimeToComplete",
    "name" : "TimeToComplete"
   }
  ]
 },
 "columns" : {
  "day" : {
   "colIndex" : 0,
   "type" : "Qualitative",
   "dataType" : "ISODateTime",
   "unit" : null,
   "name" : "day",
   "displayName" : "Date"
  },
  "avg" : {
   "colIndex" : 1,
   "type" : "Quantitative",
   "dataType" : "Double",
   "unit" : null,
   "name" : "avg",
   "displayName" : "Avg"
  }
 },
 "rows" : [
    ["2015-12-01", 32.2],
    ["2015-12-02", 35.4],
    ["2015-12-03", 32.5],
    ["2015-12-04", 40.3],
    ["2015-12-05", 36.6],
    ["2015-12-06", 31.3],
    ["2015-12-07", 35.2],
    ["2015-12-08", 34.7]],
 "scope" : {
  "metricName" : "LevelComplete",
  "appId" : "<YOUR APP ID>",
  "eventName" : "LevelComplete",
  "aggregate" : "avg"
 },
 "weightedAverage" : 34.8
}
```

## Using a ResultSet

The REST API actions you can use to query the analytics for an app return that data as a `ResultSet` resource. A `ResultSet` has these properties you can use to access the data it contains:

| Property | Description |
|---|---|
| columns | A map of column names to metadata about the column. |
| column.colIndex | The index of the column this metadata describes. |
| column.dataType | The data type of the column. |
| column.name | A unique name that identifies the column. |
| column.displayName | A friendly display name for the column. |
| column.unit | The unit of measurement for the column. |
| column.type | One of these values:<br>• **Quantitative**, which indicates the column value is numeric.<br>• **Qualitative**, which indicates the column value is a string. |
| rows | A two-dimensional array containing `ResultSet` values. Values can have different data types as described by metadata in the column property. |

# Filtering Query Results

You can filter the data provided in a `ResultSet` based on optional parameters you specify when querying the data.

## Filtering Query Results by Platform

You can filter the query results to include only the data for a particular app platform, such as Android, by including the following filter parameter in your URL:

```
platform=<filter>
```

You can use the following values for the filter:

- android
- iphoneos
- kindle
- windowsphone
- blackberry
- macos
- windows
- linux

For example, here is a query URL you would use to filter the data in the returned `ResultSet` for the iOS platform:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/new-users/lifetime-count?platform=iphoneos
```

You can filter on more than one platform at a time by using a query parameter like this one:

```
?platform=<android>,<iphoneos>,<windowsphone>
```

## Filtering Query Results by Date Range

You can filter the query results to include only the data for a range of dates by including the following filter parameter in your URL:

```
start=<start date>&end=<end date>
```

Here is an example of filtering a query by specifying a date range:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/dau/count?start=2014-11-20&end=2014-12-11
```

Not specifying a date range applies a default filter that includes data for the last 30 days. Lifetime KPIs ignore this filter and will have no effect on the result.

## Filtering Query Results by Relative Date

You can filter the query results to include only the data for a relative date. Here are the filter patterns you can use when specifying relative dates:

- n-days-ago
- n-weeks-ago
- n-months-ago
- today

Here is an example of filtering a query by specifying a relative start date and the current date for the end date:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/dau/count?start=3-days-ago&end=today
```

## Combining Query Parameters

You can combine different query parameters using the ampersand (&) symbol, as shown in this example:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/dau/count?platform=android&start=2014-11-20&end=2014-12-11
```

## Grouping Data Per Platform

To group the `ResultSet` by platform, you can append the following query parameter to the end of the URL:

```
?group-by=platform
```

Here is an example of a query that returns data values per platform:

```
https://mobileanalytics.us-east-1.amazonaws.com/2014-06-05/apps/<YOUR APP
 ID>/kpis/new-users/lifetime-count?group-by=platform
```

# Mobile Analytics Console Reports Overview

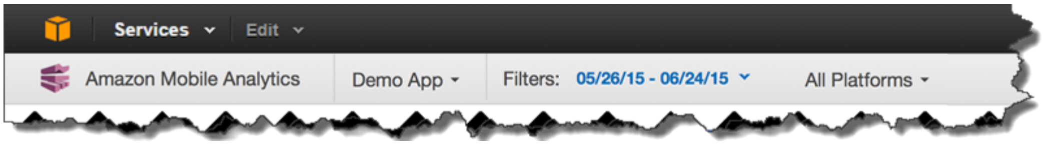
You can view or download Mobile Analytics reports from the AWS Management Console. Reports are available for metrics on active users, sessions, retention, in-app revenue, and custom events. Go to the Mobile Analytics console at https://console.aws.amazon.com/mobileanalytics/home.

Topics

## Toolbar

You can use the controls on the toolbar in the Mobile Analytics console to filter on the following :

- **Application list** – Provides a list of all apps that have submitted data to Mobile Analytics.
- **Date range** – Allows you to specify date ranges for the data to be displayed in the reports. By default, the date range is the last 30 days.
- **Platforms** – Lets you select which platform data to display in the console reports. You can choose to show data from all platforms together or data from a specific platform only.



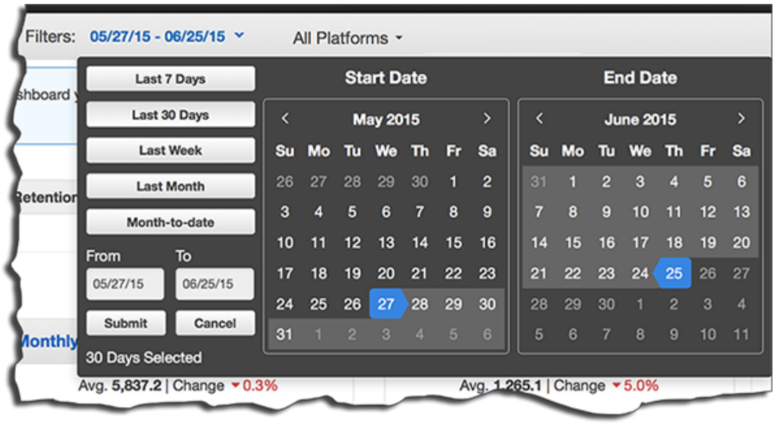
# Using the Console Reports

After you have integrated Mobile Analytics into your apps, they will relay data about user engagement to the service. You can view the currently compiled data in the console. Data reported by your apps is incorporated into the compiled data about 60 minutes after it was sent.

To view the current reports for an app, select it from the applications drop-down menu in the toolbar. The reports are organized into a set of tabs, each of which is described in this section.

## Filtering Data Shown by Date Range

You can change the date range used to filter the data shown in the reports.



Filters: 05/27/15 - 06/25/15

From 05/27/15 To 06/25/15

30 Days Selected

Avg. 5,837.2 | Change ▾0.3%

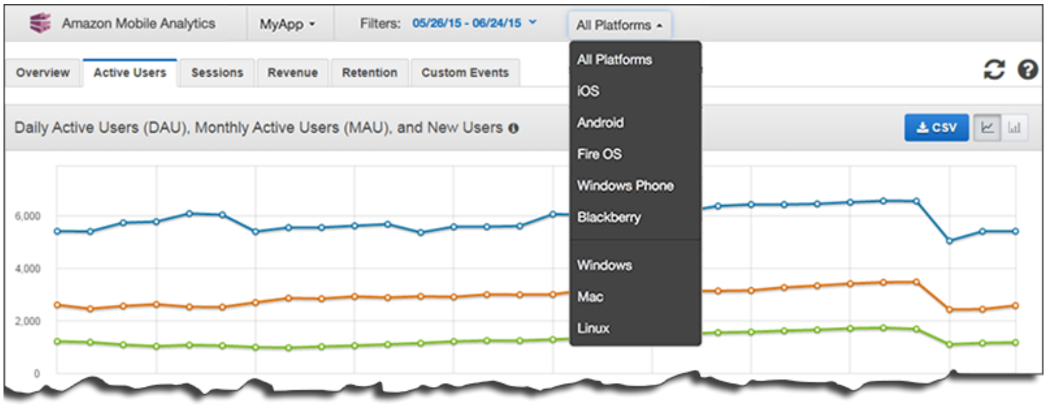
Avg. 1,265.1 | Change ▾5.0%

### To filter the data displayed in the console

1. Choose the **Filter** menu in the toolbar.
2. Choose the dates you want reflected in the console reports. You can choose one of the preset date range options or enter dates in the **From** and **To** fields.
3. Choose **Submit** to apply the date filter to the console reports.

## Selecting the Platform Displayed

You can view the data for all platforms supported by your app or for specific platforms only.



Filters: 05/26/15 - 06/24/15

### To change the platforms displayed in the reports

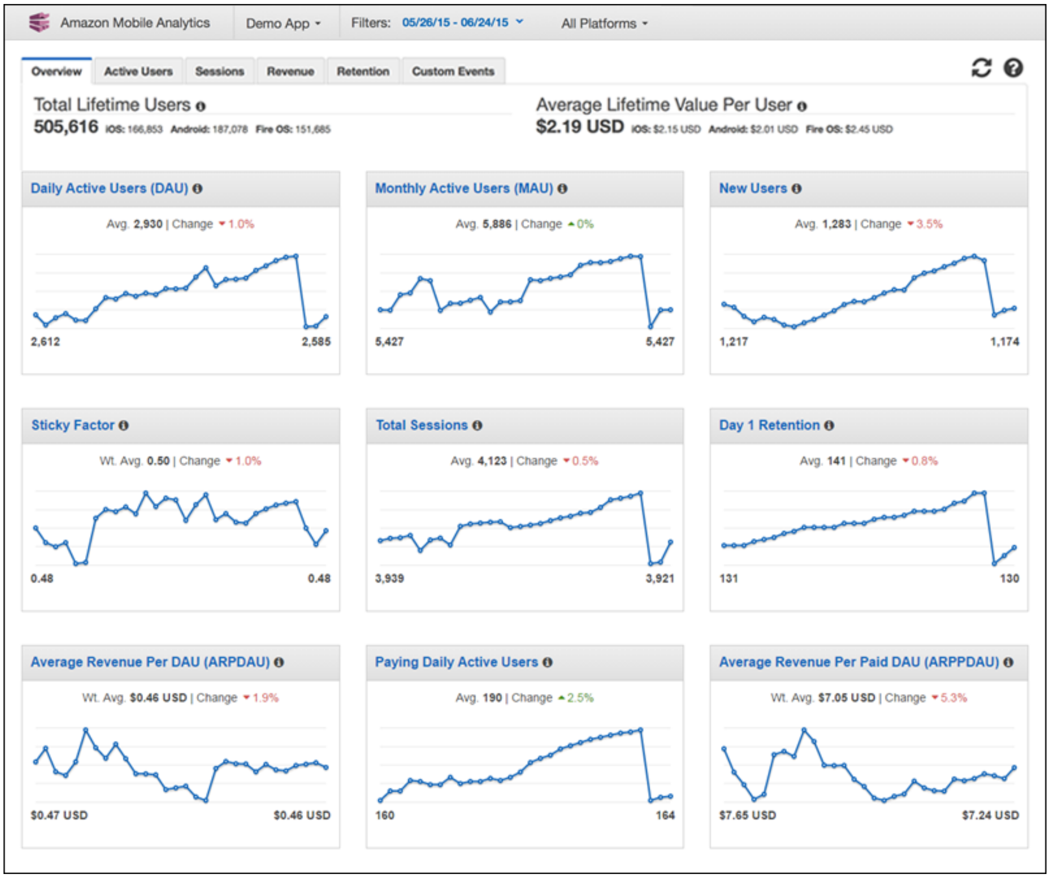
1. Choose the **All Platforms** menu in the toolbar.
2. From the drop-down list, choose **All Platforms** to display data reported by the app on all platforms or choose any of the following platforms:
   - iOS
   - Android
   - Fire OS
   - Windows Phone
   - Blackberry
   - Windows (browser-based apps)
   - Mac (browser-based apps)
   - Linux (browser-based apps)

Browser-based apps that run under one of the mobile operating systems are reported under the appropriate operating system.

# Overview Tab

The **Overview** tab displays at-a-glance summaries of:

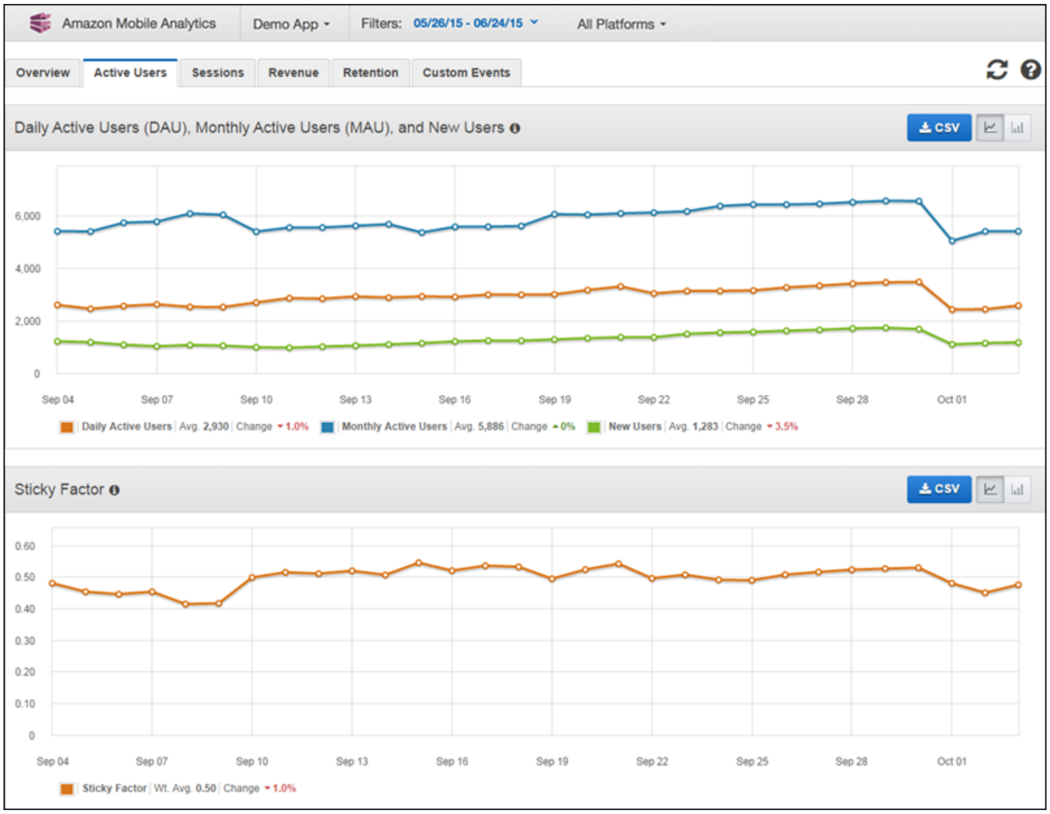
- **Daily Active Users (DAU)** – Users who used your app on a particular day.
- **Monthly Active Users (MAU)** – Users who used your app in the previous 30 days.
- **New Users** – New users who used your app on a particular day.
- **Sticky Factor** – Fraction of monthly users who used your app on a particular day (DAU/MAU).
- **Total Sessions** – Number of times your app was used on a particular day.
- **Day 1 Retention** – Percentage of new users who used your app on a specific day and then again the following day.
- **Average Revenue Per Paid Daily Active User (ARPPDAU)** – Gross revenue for in-app items per daily active user who purchased in-app items.



# Active Users Tab

The **Active Users** tab displays trend charts for the following:

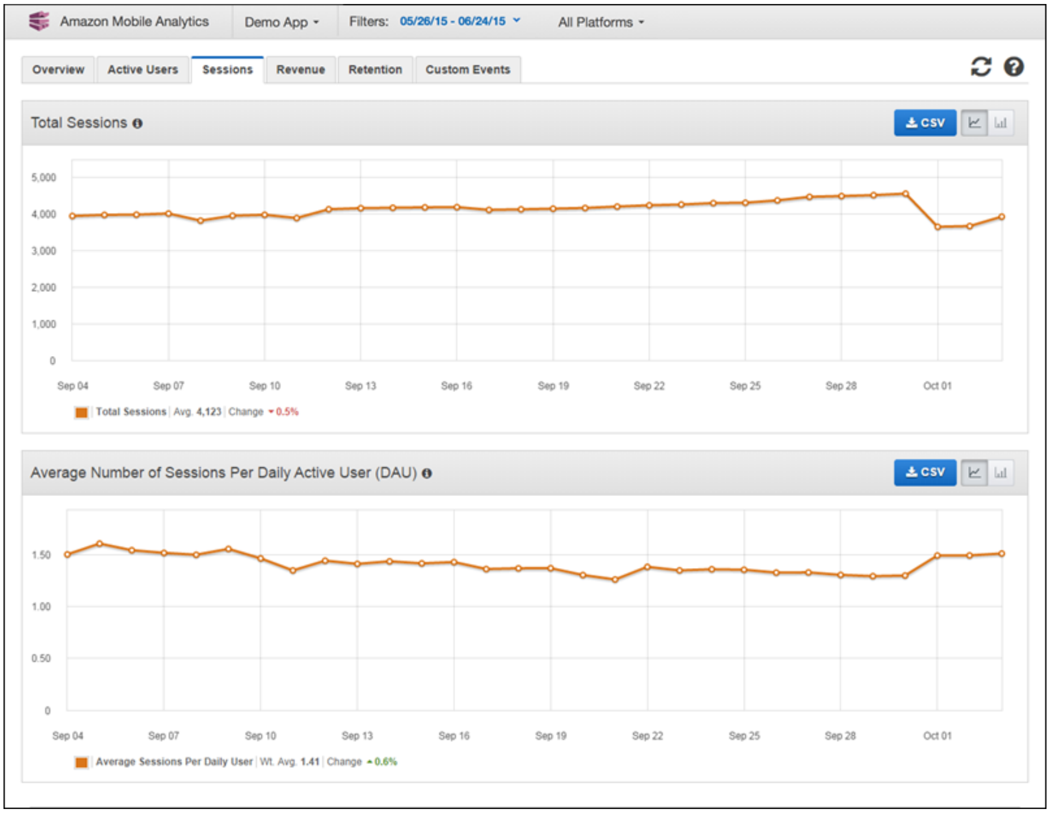
- **Daily Active Users (DAU)** – Users who used your app on a particular day.
- **Monthly Active Users (MAU)** – Users who used your app in the previous 30 days.
- **New Users** – New users who used your app on a particular day.
- **Sticky Factor** – Fraction of monthly users using your app on a particular day. For example, a Sticky Factor of .25 means that on a particular day, 25% of your users from the previous 30 days used your app.



# Sessions Tab

The **Sessions** tab displays trend charts for the following:

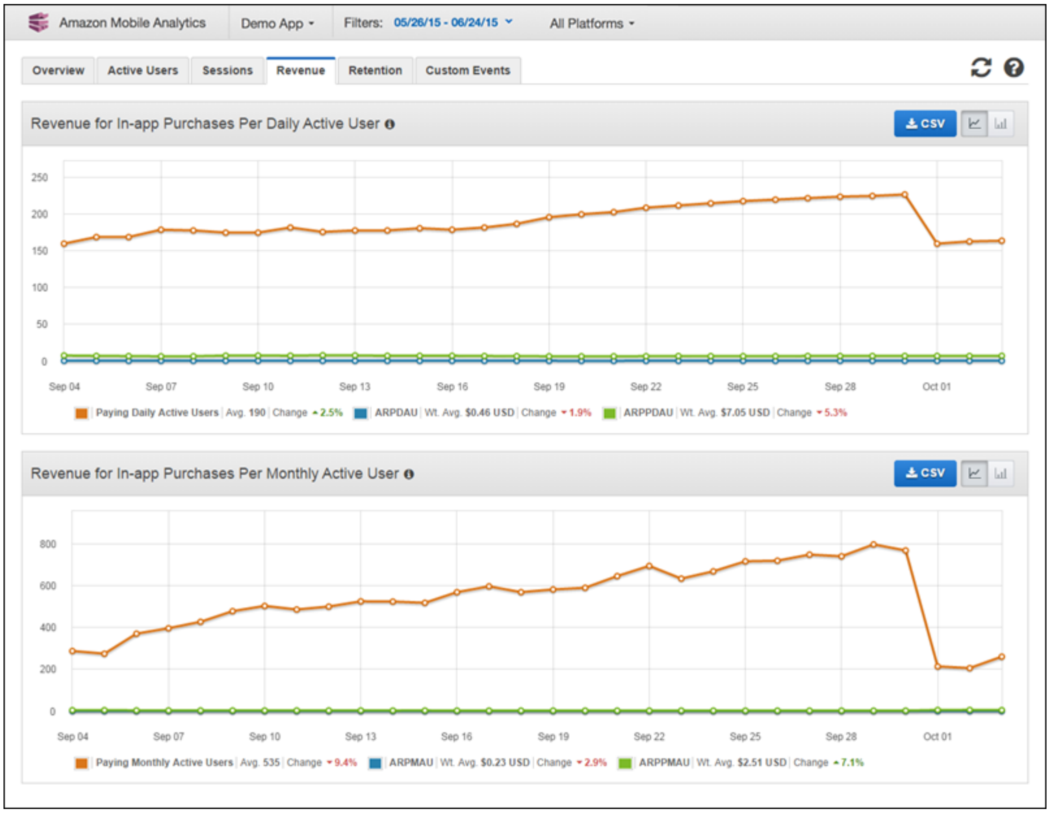
- **Total Sessions** – Number of times your app was used on a particular day.
- **Average Number of Sessions Per Daily Active User** – Average of the number of sessions per daily active user who used your app on a particular day.



# Revenue Tab

The **Revenues** tab displays trend charts for the following:

- **Average Revenue Per Daily Active User (ARPDAU)** – Gross revenue for in-app items per daily active user.
- **Average Revenue Per Paying Daily Active User (ARPPDAU)** – Gross revenue for in-app items per daily active user who purchased in-app items.
- **Paying Daily Active Users** – Users who bought an in-app item on a particular day.


Amazon Mobile Analytics
Demo App
Filters: 05/26/15 - 06/24/15
All Platforms
Overview
Active Users
Sessions
Revenue
Retention
Custom Events
Revenue for In-app Purchases Per Daily Active User
CSV
250
200
150
100
50
0
Sep 04
Sep 07
Sep 10
Sep 13
Sep 16
Sep 19
Sep 22
Sep 25
Sep 28
Oct 01
🟧 Paying Daily Active Users | Avg. 190 | Change ▲2.5%
🟦 ARPDAU | Wt. Avg. $0.46 USD | Change ▼1.9%
🟩 ARPPDAU | Wt. Avg. $7.05 USD | Change ▼5.3%
Revenue for In-app Purchases Per Monthly Active User
CSV
800
600
400
200
0
Sep 04
Sep 07
Sep 10
Sep 13
Sep 16
Sep 19
Sep 22
Sep 25
Sep 28
Oct 01
🟧 Paying Monthly Active Users | Avg. 535 | Change ▼9.4%
🟦 ARPMAU | Wt. Avg. $0.23 USD | Change ▼2.9%
🟩 ARPPMAU | Wt. Avg. $2.51 USD | Change ▲7.1%

Gross revenue does not include taxes, product returns, reimbursements, subscriptions or revenue from the purchase of the app.

Revenue for in-app purchases per monthly active user:

- **Average Revenue Per Monthly Active User (ARPMAU)** – The gross revenue divided by the number of users who bought an item from your app in the last 30 days.
- **Average Revenue Per Paid Daily Active User (ARPPMAU)** – Gross revenue for in-app items per monthly active user who purchased in-app items.
- **Paying Monthly Active Users** – Users who bought an in-app item in a particular month.

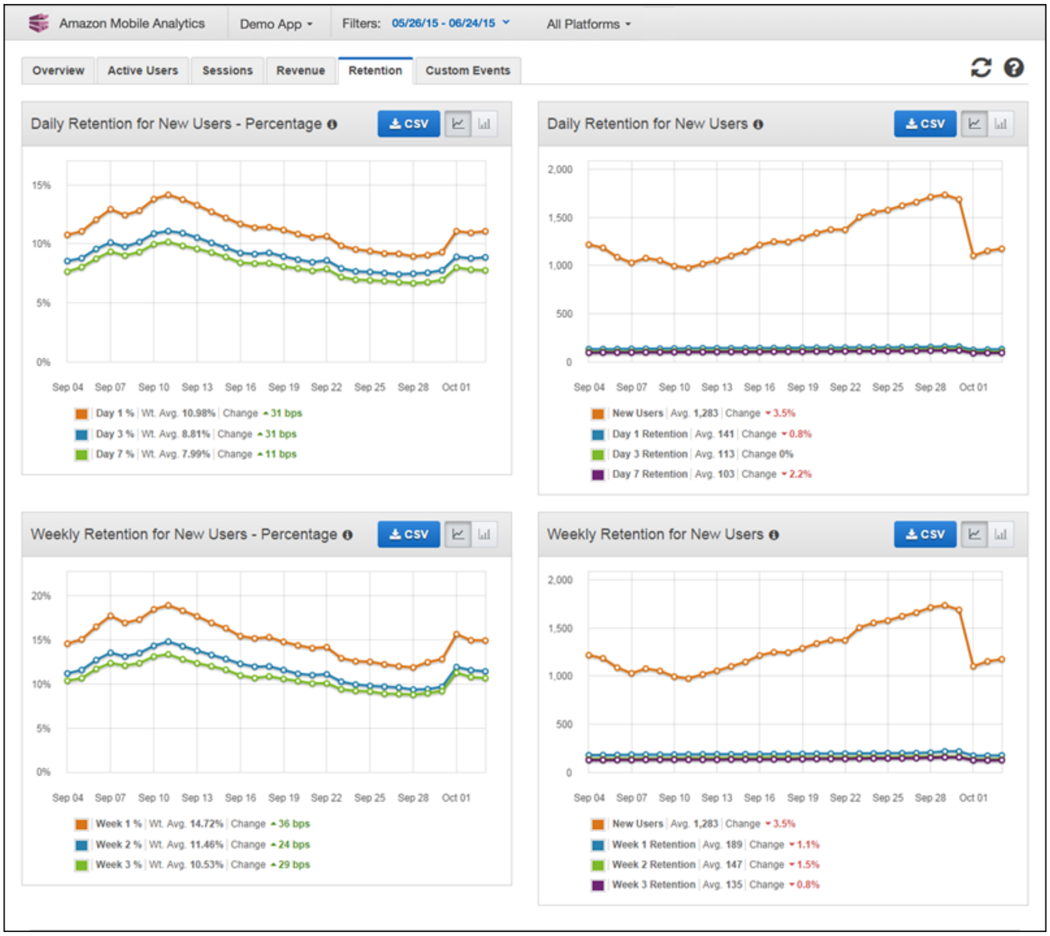
Gross revenue does not include taxes, product returns, reimbursements, subscriptions or revenue from the purchase of the app.

# Retention Tab

The **Retention** tab displays trend charts for the following:

- **Daily Retention for New Users—Percentage** – Percentage of new users who used your app on a specific day and then again on the following day, the third day, and the seventh day.
- **Weekly Retention for New Users—Percentage** –Percentage of users who used your app on a specific day and then at least once during the next 7 days inclusive (1-week retention); between 8 and 14 days inclusive (2-week retention); and between 15 and 21 days inclusive (3-week retention).
- **Daily Retention for New Users—Count** – New users who used your app on a specific day and then again on the following day, the third day, and the seventh day. This chart also shows the number of new users who used your app on a particular day.

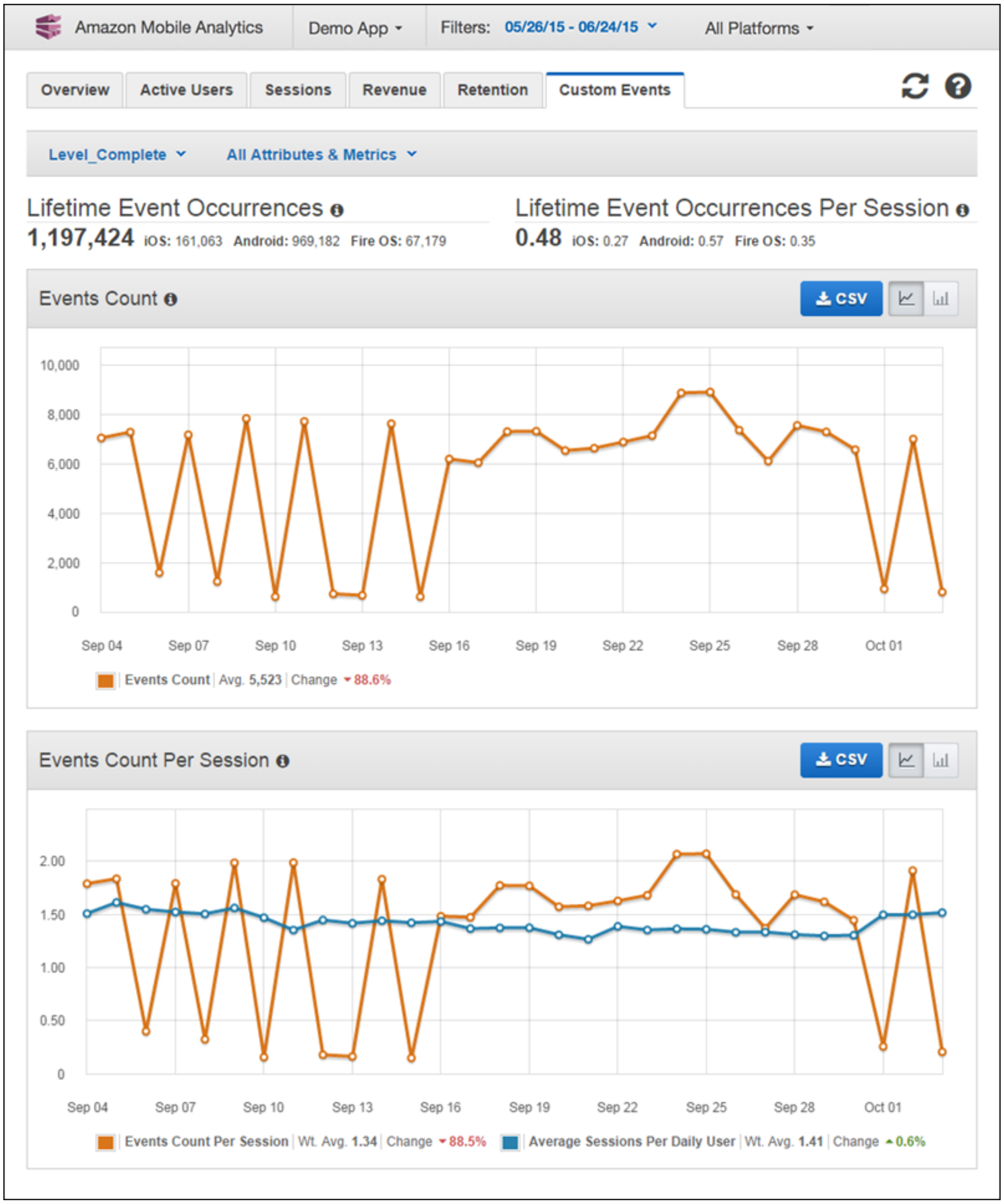
- **Weekly Retention for New Users—Count** – New users who used your app on a specific day and then at least once during the next seven days inclusive (1-week retention); between 8 and 14 days inclusive (2-week retention); and between 15 and 21 days inclusive (3-week retention). This chart also shows the number of new users who used your app on a particular day.


Amazon Mobile Analytics
Demo App
Filters: 05/26/15 - 06/24/15
All Platforms
Overview
Active Users
Sessions
Revenue
Retention
Custom Events
Daily Retention for New Users - Percentage
CSV
15%
10%
5%
0%
Sep 04
Sep 07
Sep 10
Sep 13
Sep 16
Sep 19
Sep 22
Sep 25
Sep 28
Oct 01
Day 1 % | Wt. Avg. 10.98% | Change ▴31 bps
Day 3 % | Wt. Avg. 8.81% | Change ▴31 bps
Day 7 % | Wt. Avg. 7.99% | Change ▴11 bps
Daily Retention for New Users
CSV
2,000
1,500
1,000
500
0
Sep 04
Sep 07
Sep 10
Sep 13
Sep 16
Sep 19
Sep 22
Sep 25
Sep 28
Oct 01
New Users | Avg. 1,283 | Change ▾3.5%
Day 1 Retention | Avg. 141 | Change ▾0.8%
Day 3 Retention | Avg. 113 | Change 0%
Day 7 Retention | Avg. 103 | Change ▾2.2%
Weekly Retention for New Users - Percentage
CSV
20%
15%
10%
5%
0%
Sep 04
Sep 07
Sep 10
Sep 13
Sep 16
Sep 19
Sep 22
Sep 25
Sep 28
Oct 01
Week 1 % | Wt. Avg. 14.72% | Change ▴36 bps
Week 2 % | Wt. Avg. 11.46% | Change ▴24 bps
Week 3 % | Wt. Avg. 10.53% | Change ▴29 bps
Weekly Retention for New Users
CSV
2,000
1,500
1,000
500
0
Sep 04
Sep 07
Sep 10
Sep 13
Sep 16
Sep 19
Sep 22
Sep 25
Sep 28
Oct 01
New Users | Avg. 1,283 | Change ▾3.5%
Week 1 Retention | Avg. 189 | Change ▾1.1%
Week 2 Retention | Avg. 147 | Change ▾1.5%
Week 3 Retention | Avg. 135 | Change ▾0.8%

# Custom Events Tab

The **Custom Events** tab displays charts for metrics that you define.

- **Lifetime Event Occurrences** – Aggregate number of custom events received for your app to date.
- **Lifetime Event Occurrences Per Session** – Average number of custom events received per session to date.
- **Events Count Per Session** – Average number of custom events received per session.
- **Breakdown of Metric Values** – Average, minimum, and maximum of metric values.



Lifetime Event Occurrences
1,197,424 iOS: 161,063 Android: 969,182 Fire OS: 67,179

Lifetime Event Occurrences Per Session
0.48 iOS: 0.27 Android: 0.57 Fire OS: 0.35

Events Count

Events Count | Avg. 5,523 | Change ▾88.6%

Events Count Per Session

Events Count Per Session | Wt. Avg. 1.34 | Change ▾88.5% Average Sessions Per Daily User | Wt. Avg. 1.41 | Change ▴0.6%

# Working with Charts

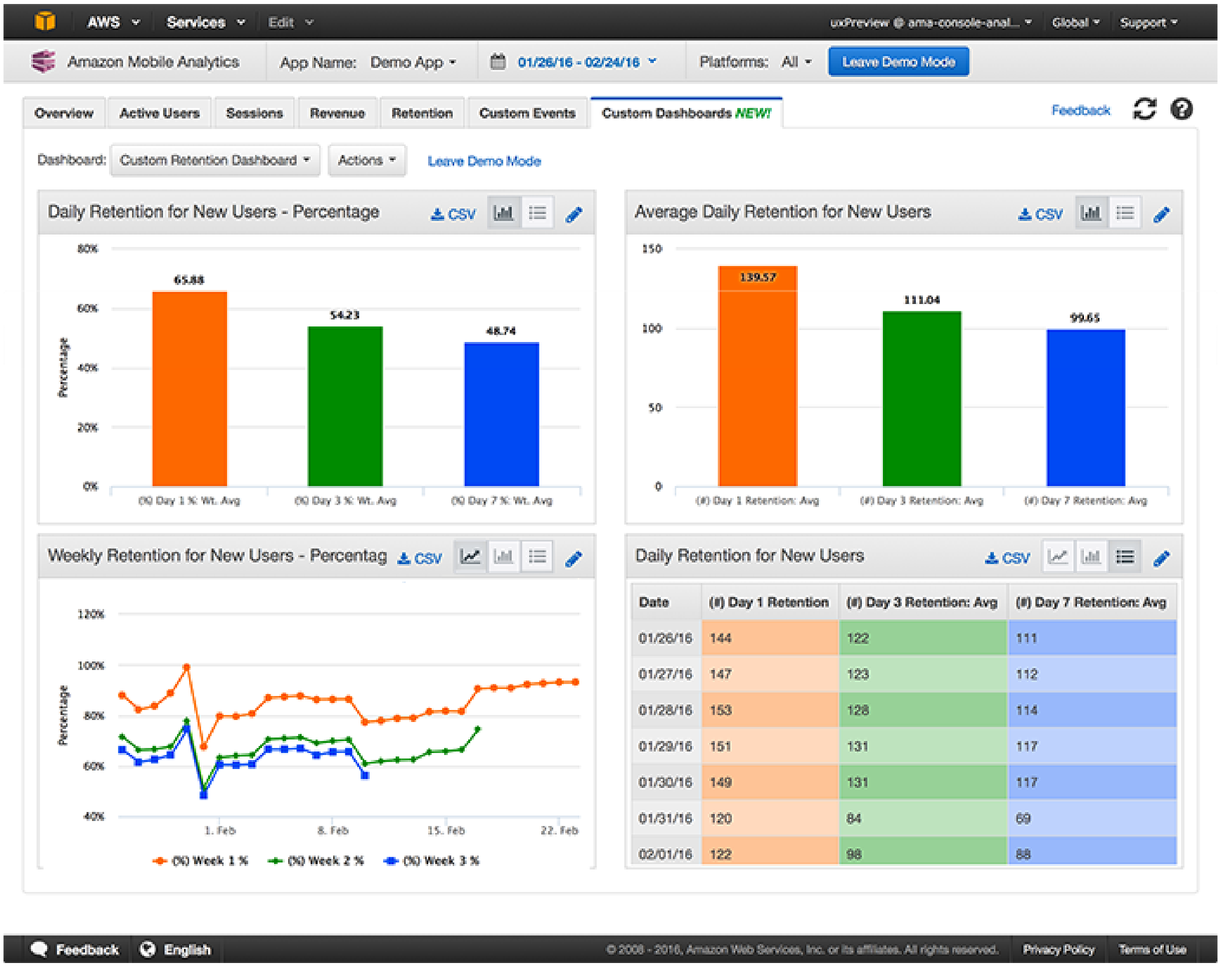
Many of the tabs on the Mobile Analytics console include controls for displaying or downloading data:

| | |
|---|---|
| ⬇ CSV | Click to download the currently displayed data to a comma-separated values file. |
| 📈 | Click to display the current data as a line chart. |

| | |
|---|---|
| 📊 | Click to display the current data as a bar chart. |

# Using Custom Dashboards

You can create custom dashboards in the Mobile Analytics console to help you quickly and easily monitor the particular metrics and analytics that are most beneficial for measuring the performance of your app.



| Date | (#) Day 1 Retention | (#) Day 3 Retention: Avg | (#) Day 7 Retention: Avg |
|---|---|---|---|
| 01/26/16 | 144 | 122 | 111 |
| 01/27/16 | 147 | 123 | 112 |
| 01/28/16 | 153 | 128 | 114 |
| 01/29/16 | 151 | 131 | 117 |
| 01/30/16 | 149 | 131 | 117 |
| 01/31/16 | 120 | 84 | 69 |
| 02/01/16 | 122 | 98 | 88 |

You can have multiple custom dashboards, each of which can include up to four charts to display details of how users interact with your app. The charts are customizable, allowing you to include standard or custom data points as well as choose whether to use line chart, bar chart, or tabular presentation formats depending on the type of chart.

You can try custom dashboards in a fully interactive demo mode. For more information, see Custom Dashboards Demo.

# Types of Dashboard Charts

There are two types of charts you can add to a custom dashboard:

- **Summary** charts, which display their data per series aggregated over the displayed date range. Summary charts can be viewed either as a bar chart or in tabular format.
- **Trended** charts, which display their data per day over the displayed date range. Trended charts can be viewed as a trendline chart, a bar chart, or in tabular format.

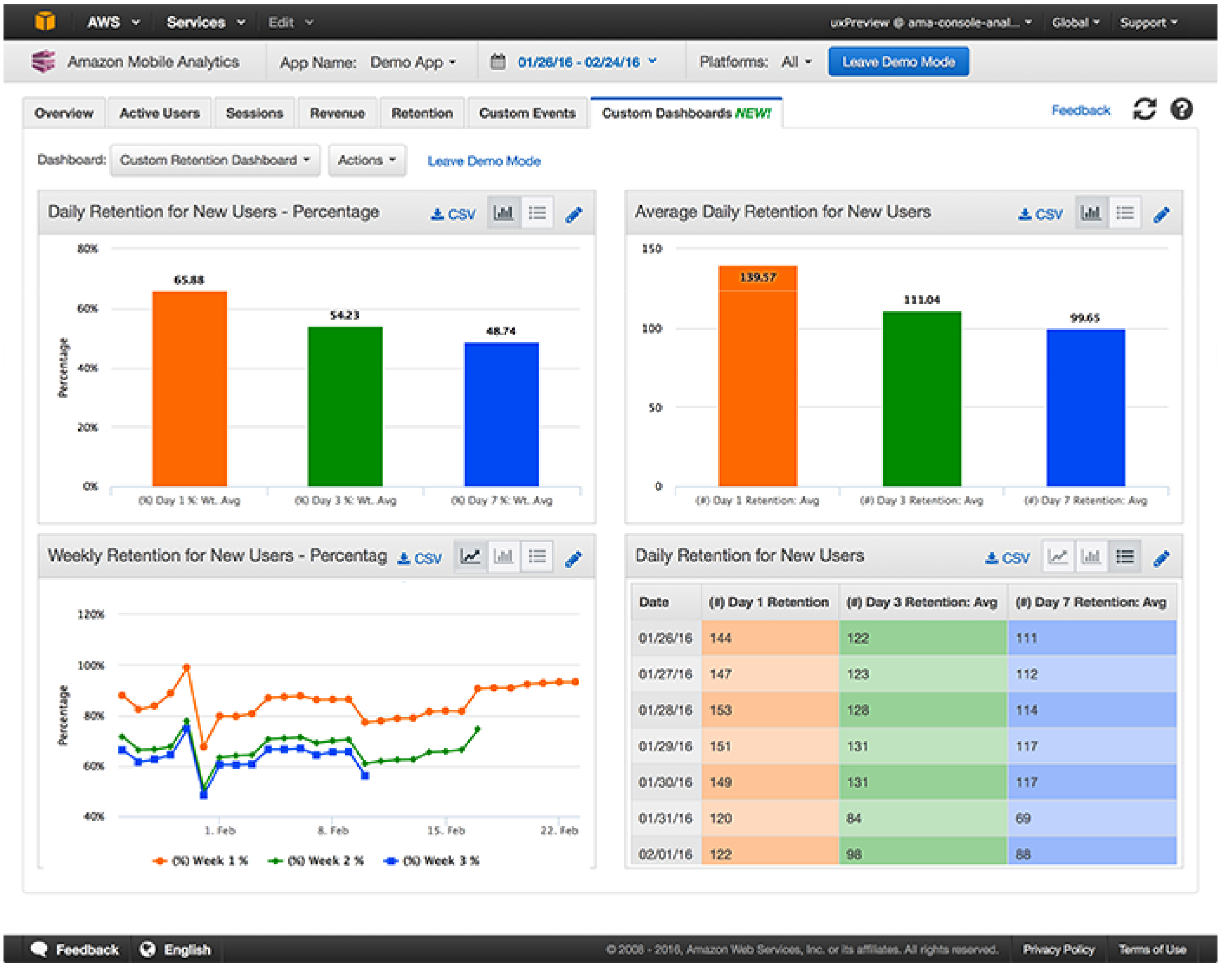
Topics

# Viewing Dashboards

Custom dashboards are available in their own tab in the Mobile Analytics console.

### To view any of your custom dashboards

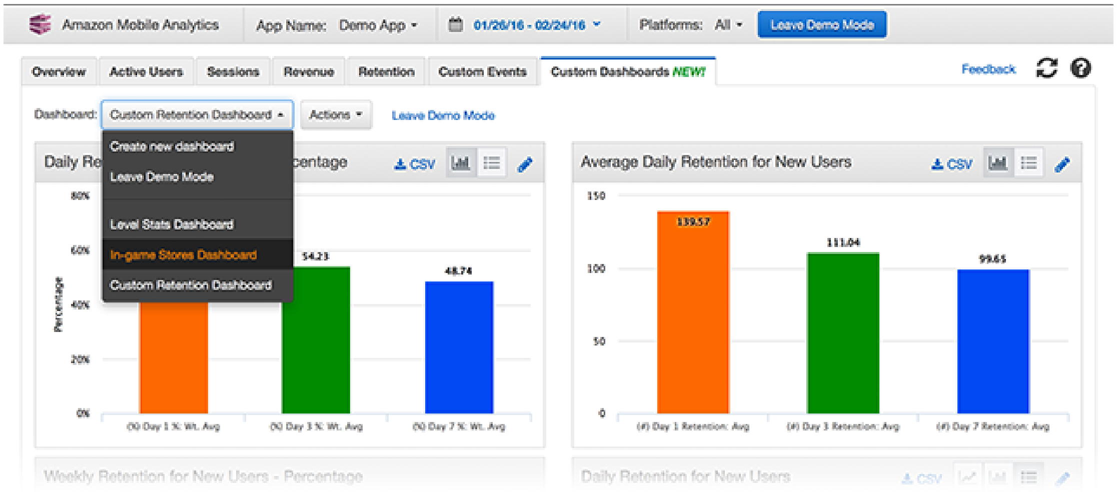
1. Open the Amazon Mobile Analytics console at https://console.aws.amazon.com/mobileanalytics/home.
2. In the main console dashboard, select the **Custom Dashboards** tab.



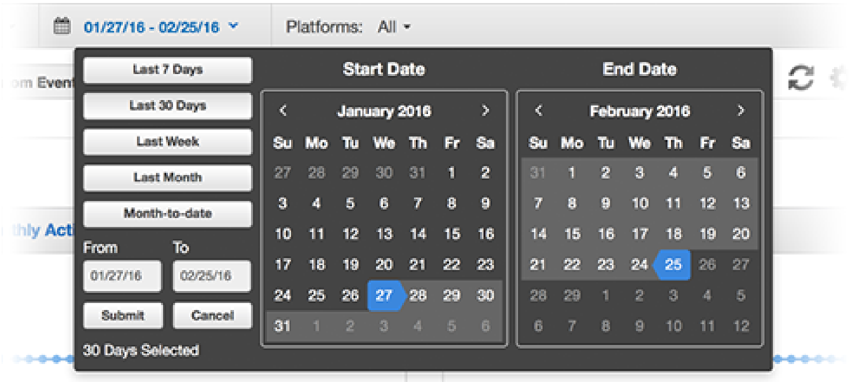
| Date | (#) Day 1 Retention | (#) Day 3 Retention: Avg | (#) Day 7 Retention: Avg |
|---|---|---|---|
| 01/26/16 | 144 | 122 | 111 |
| 01/27/16 | 147 | 123 | 112 |
| 01/28/16 | 153 | 128 | 114 |
| 01/29/16 | 151 | 131 | 117 |
| 01/30/16 | 149 | 131 | 117 |
| 01/31/16 | 120 | 84 | 69 |
| 02/01/16 | 122 | 98 | 88 |

A custom dashboard displays from one to four charts. By default, each chart has one of three views, either bar chart, or line chart, or a table. You can switch among the available views at any time to see the data presented in different ways.

3. To display a particular custom dashboard, select its name from the **Dashboard** list.

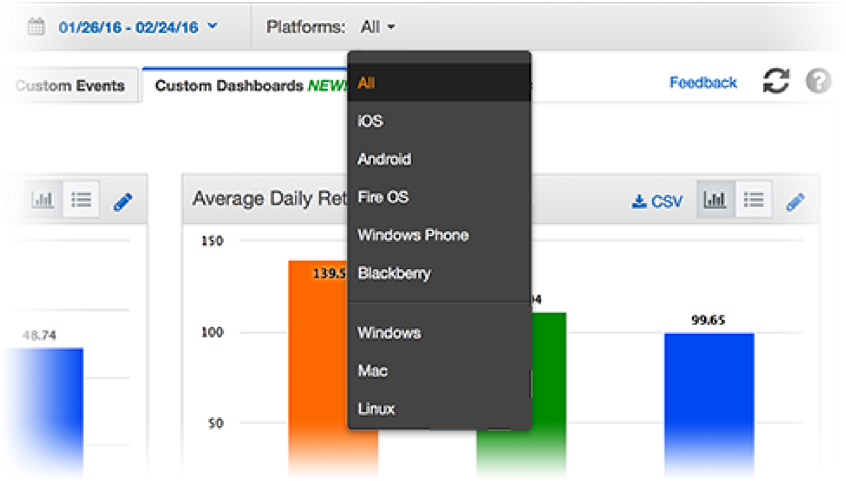


4. Select the date range you want to see from the Date filter in the Mobile Analytics toolbar.



# Filtering Charts by Platform

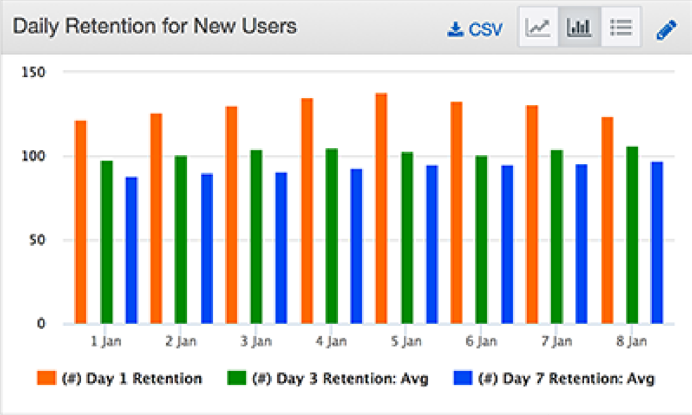
You can filter the data displayed in a custom dashboard using the Platform filter in the Mobile Analytics toolbar.



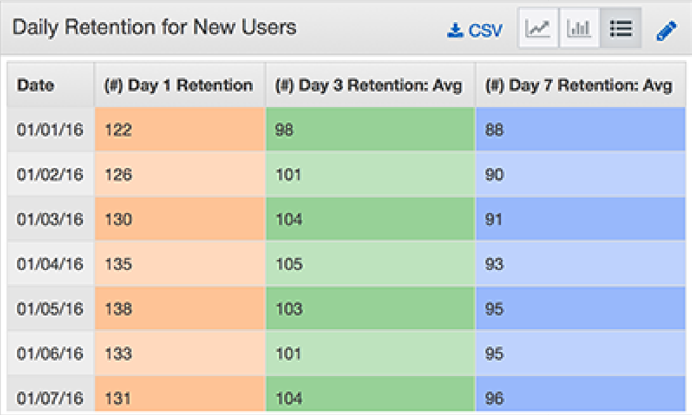
# Switching Chart Display

Each chart in a custom dashboard has a default display that was selected when the dashboard was initially created. However, you can switch between the available displays at any time when viewing a dashboard by selecting from the display options provided in the chart title bar. The options for chart display are:

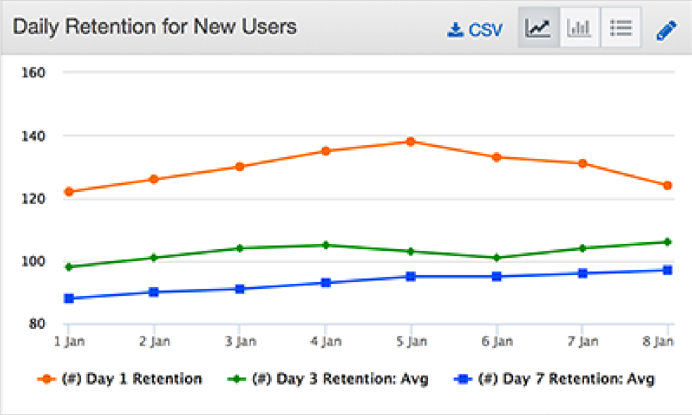
| Option | Description |
|---|---|
| 📊 | Data presented in bar chart format. Available in summary charts and trending charts. |

<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td></td><td>


</td></tr>
<tr><td>☰</td><td>
Data presented in tabular format. Available in summary charts and trending charts.



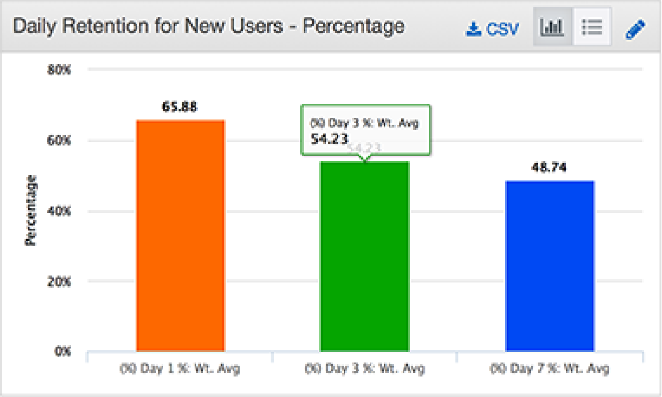
<table>
<tr><th>Date</th><th>(#) Day 1 Retention</th><th>(#) Day 3 Retention: Avg</th><th>(#) Day 7 Retention: Avg</th></tr>
<tr><td>01/01/16</td><td>122</td><td>98</td><td>88</td></tr>
<tr><td>01/02/16</td><td>126</td><td>101</td><td>90</td></tr>
<tr><td>01/03/16</td><td>130</td><td>104</td><td>91</td></tr>
<tr><td>01/04/16</td><td>135</td><td>105</td><td>93</td></tr>
<tr><td>01/05/16</td><td>138</td><td>103</td><td>95</td></tr>
<tr><td>01/06/16</td><td>133</td><td>101</td><td>95</td></tr>
<tr><td>01/07/16</td><td>131</td><td>104</td><td>96</td></tr>
</table>
</td></tr>
<tr><td>📈</td><td>
Data presented in trendline format. Available in trending charts.



</td></tr>
</table>

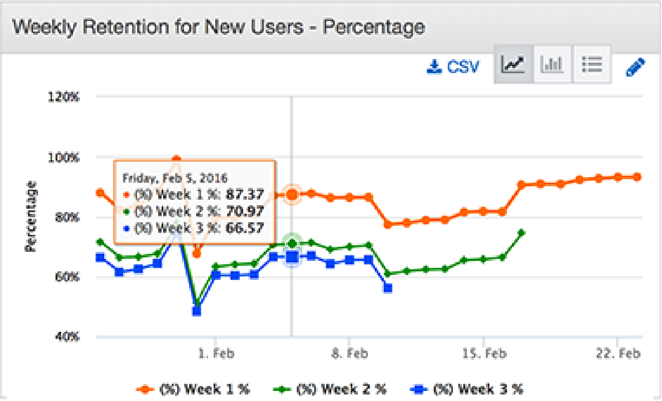
## Viewing Bar Chart Details

You can display details about any metric in a bar chart by selecting that bar.



## Viewing Line Chart Details

You can display details about any date in a line chart by selecting that date.



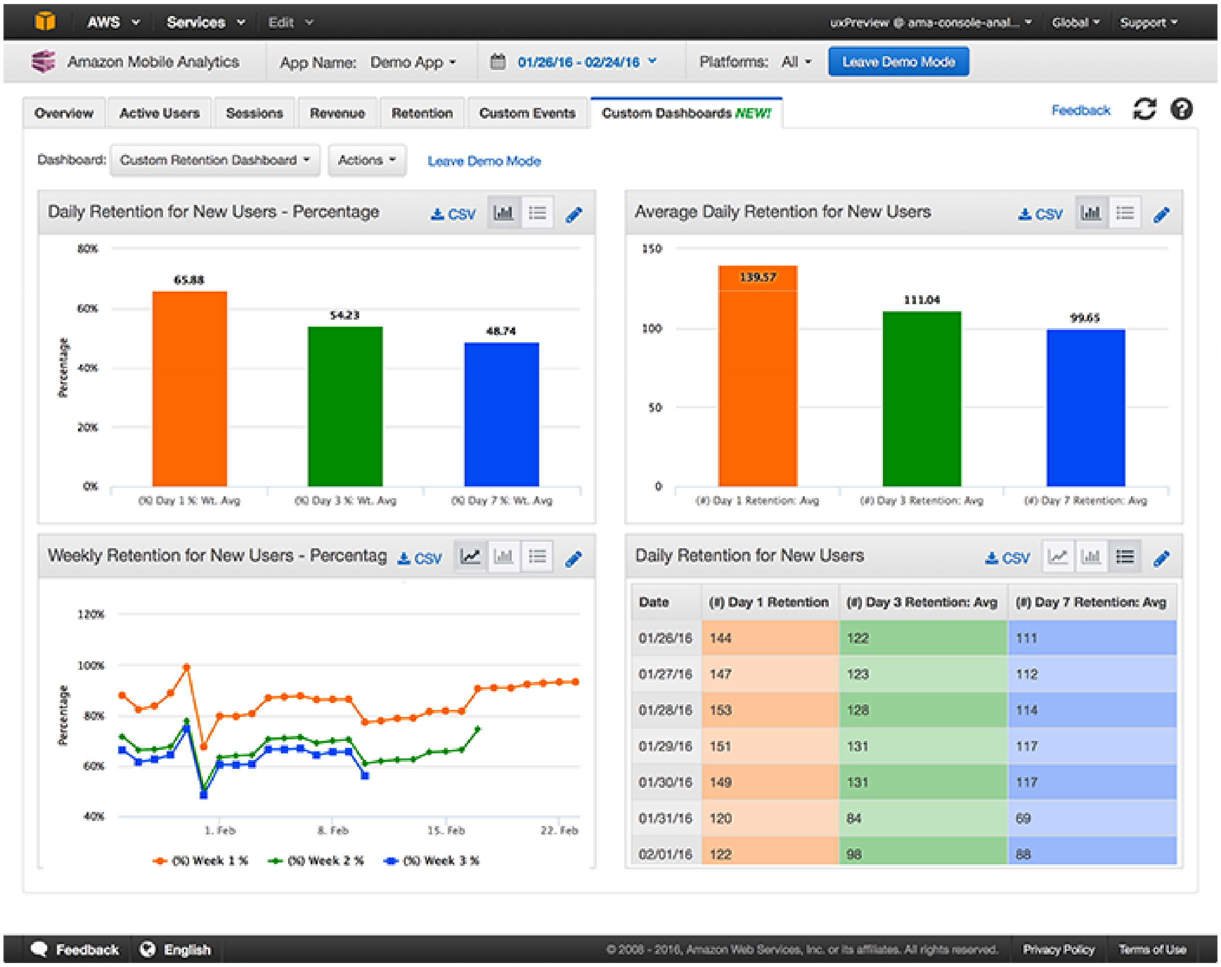
# Creating Dashboards

You can create one or more custom dashboards to provide a variety of views into the analytics reported by your apps. A custom dashboard has at least one but can have up to four charts. Each chart can display one or more series of data values, including values from standard series analytics provided by Mobile Analytics as well as custom series you define to meet the particular needs of your app and organization.

For example, tracking the financial performance of your app can be facilitated by a custom dashboard that includes data from monetization events along with data about new users.

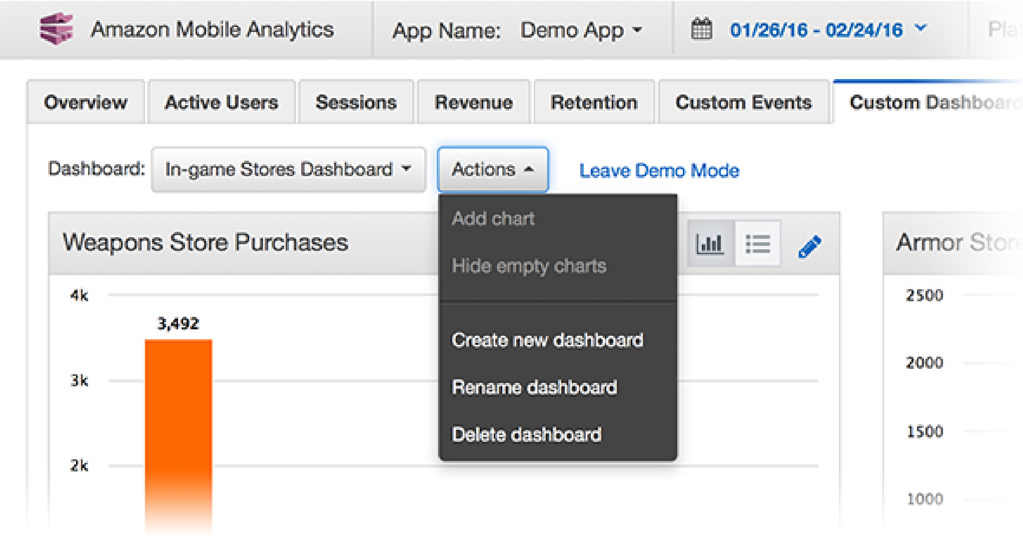
### To create a custom dashboard

1. Open the Amazon Mobile Analytics console at https://console.aws.amazon.com/mobileanalytics/home.
2. In the main console dashboard, select the **Custom Dashboards** tab.

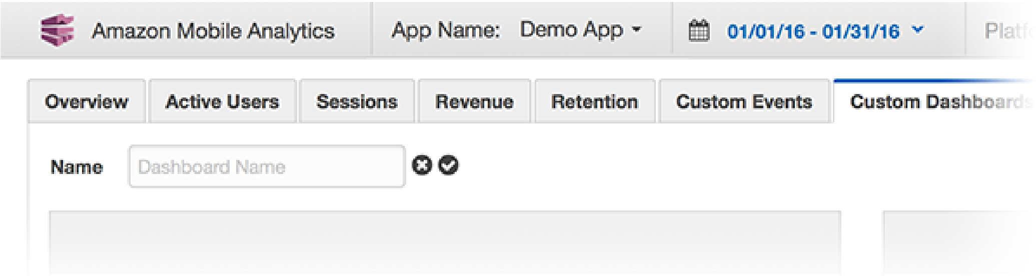


| Date | (#) Day 1 Retention | (#) Day 3 Retention: Avg | (#) Day 7 Retention: Avg |
|---|---|---|---|
| 01/26/16 | 144 | 122 | 111 |
| 01/27/16 | 147 | 123 | 112 |
| 01/28/16 | 153 | 128 | 114 |
| 01/29/16 | 151 | 131 | 117 |
| 01/30/16 | 149 | 131 | 117 |
| 01/31/16 | 120 | 84 | 69 |
| 02/01/16 | 122 | 98 | 88 |

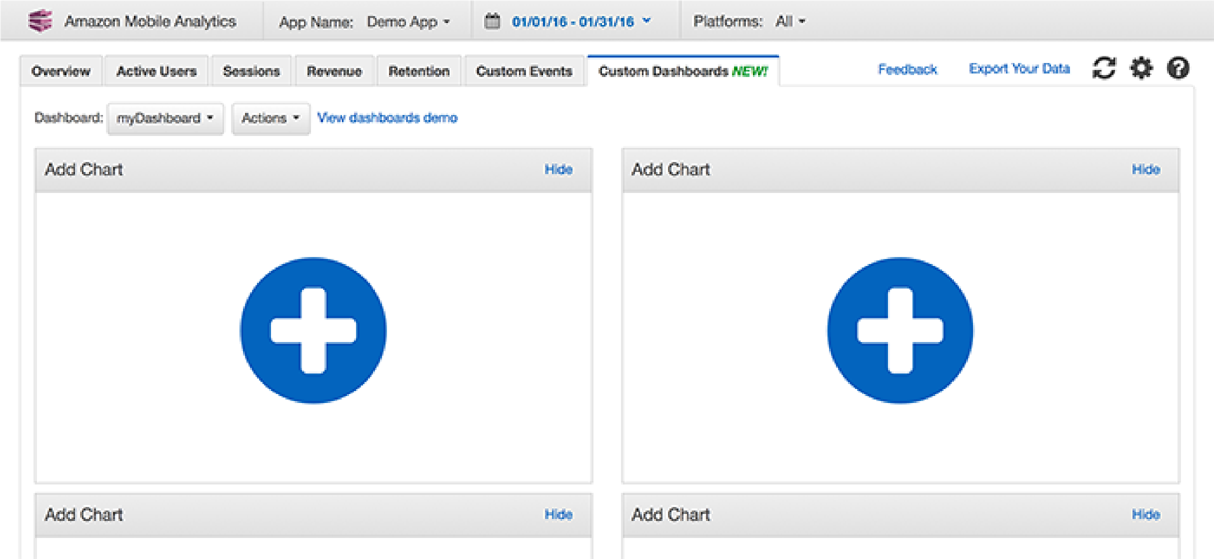
3. Select **Create new dashboard** from **Actions**.



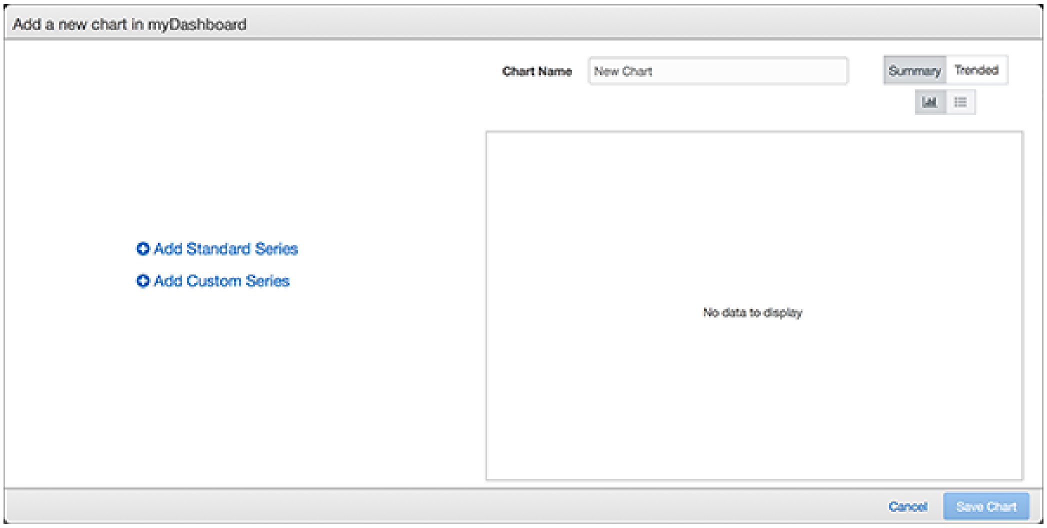
4. Type the name for the new dashboard in **Name** then select ✔ to save the blank dashboard with the name provided or ✖ to cancel.



5. Add the individual charts you want to include in this dashboard. To add a chart, select the large icon in the center of one of the four blank chart positions in the dashboard.



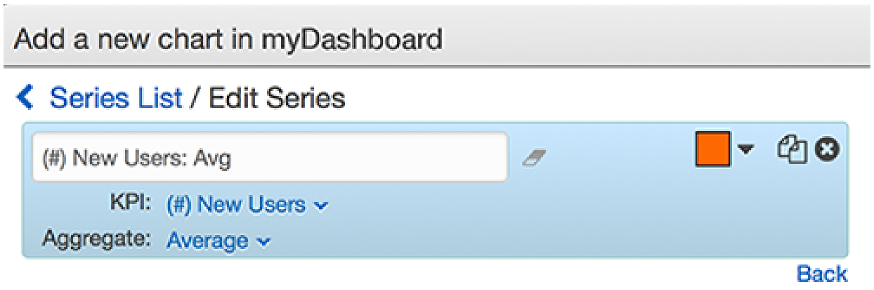
6. In the chart editing dialog, type a name for the new chart in **Chart Name**.



7. Select either **Summary** or **Trending** to specify the type of this chart.
8. Add one or more series of values you want to display in the chart. You can add any of the values from the standard series provided by Mobile Analytics or values from the custom series created from your custom events.

Select **Add Standard Series** to add one of the standard data values from Mobile Analytics.

9. In the **Series List** item that appears, you can replace the default series name with a name more appropriate for your dashboard.



Add a new chart in myDashboard

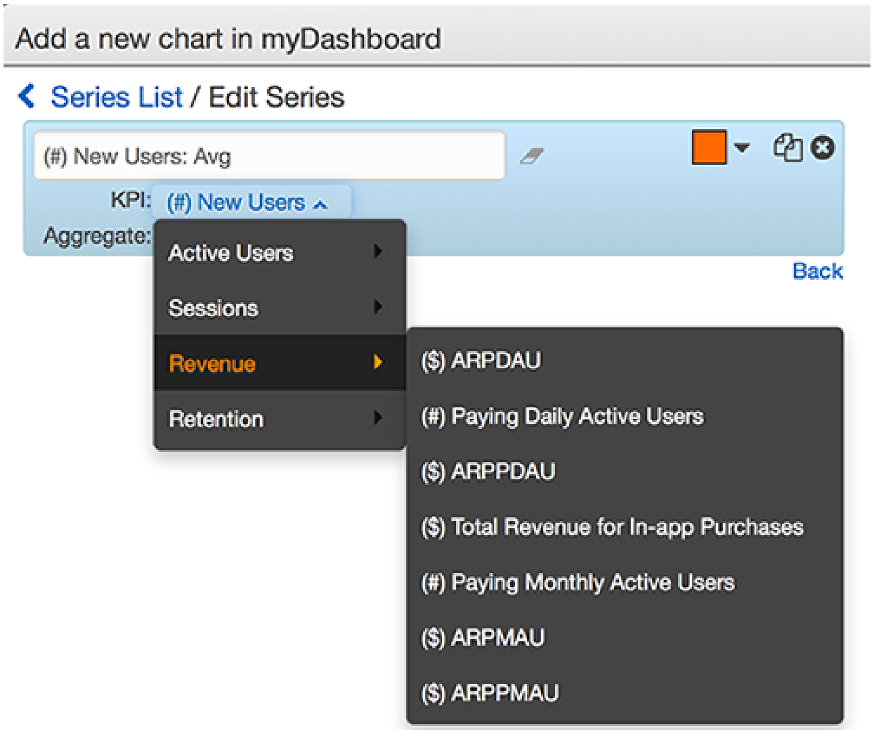
Series List / Edit Series

(#) New Users: Avg

KPI: (#) New Users

Aggregate: Average

10. In the **KPI** drop-down list, select the data value you want to chart. The KPIs are organized into categories for your convenience.



Add a new chart in myDashboard

Series List / Edit Series

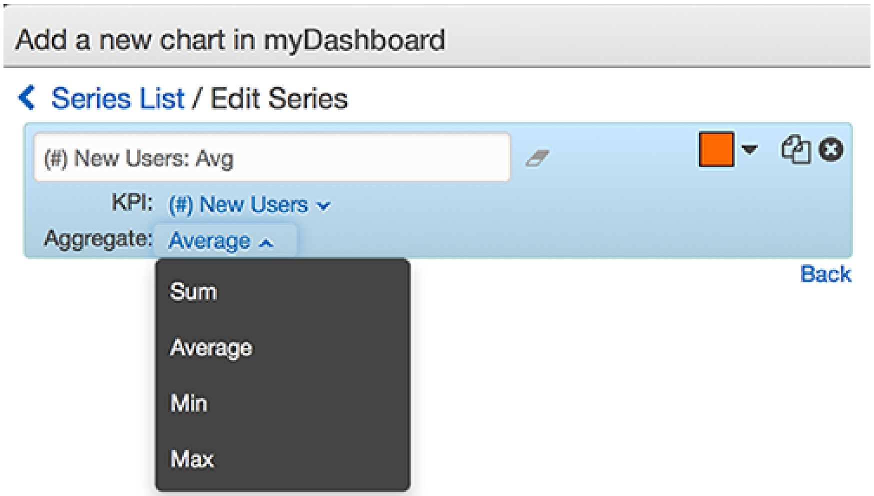
(#) New Users: Avg

KPI: (#) New Users

Aggregate:

Back
Active Users
Sessions
Revenue
Retention
($) ARPDAU
(#) Paying Daily Active Users
($) ARPPDAU
($) Total Revenue for In-app Purchases
(#) Paying Monthly Active Users
($) ARPMAU
($) ARPPMAU

11. In the **Aggregation** drop-down list, select the aggregation method you want to apply to the data values from this KPI. The selections in this list vary depending on the KPI selected for this data series.



Add a new chart in myDashboard

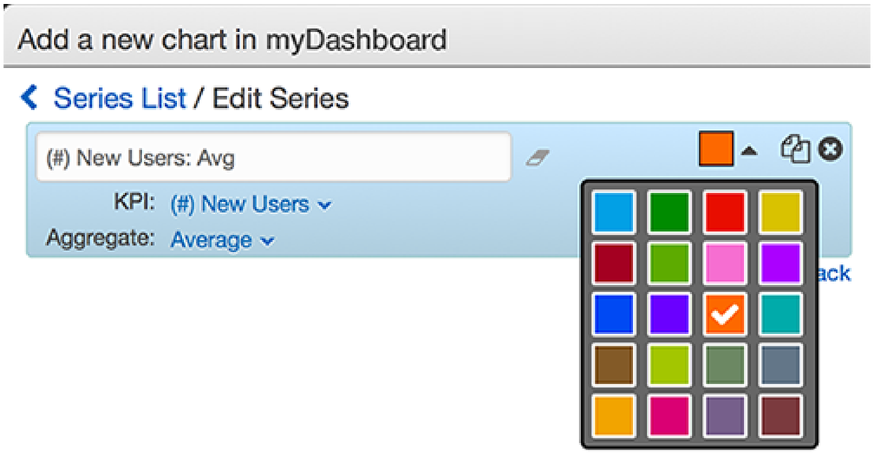
Series List / Edit Series

(#) New Users: Avg

KPI: (#) New Users

Aggregate: Average

12. Select the color you want to use to show this data series from the drop-down color swatch pallette.



Add a new chart in myDashboard

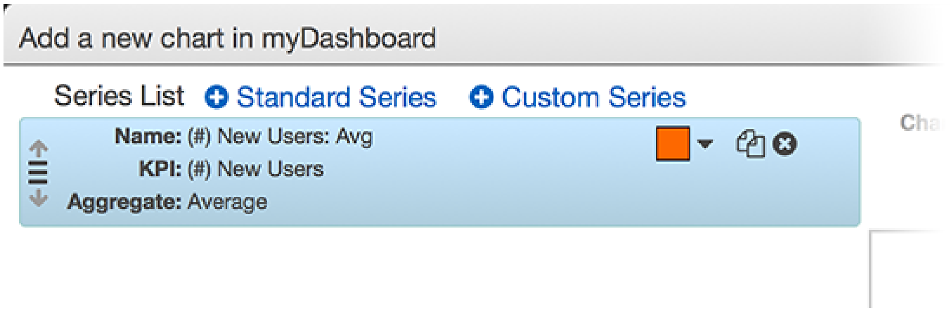
Series List / Edit Series

(#) New Users: Avg

KPI: (#) New Users

Aggregate: Average

13. When you finish defining this data series for the chart, select **Back** to add this item to the chart.
14. You can add additional data series to the chart by selecting **Standard Series** for one of the Mobile Analytics KPIs or **Custom Series** for one of your custom events.



Add a new chart in myDashboard

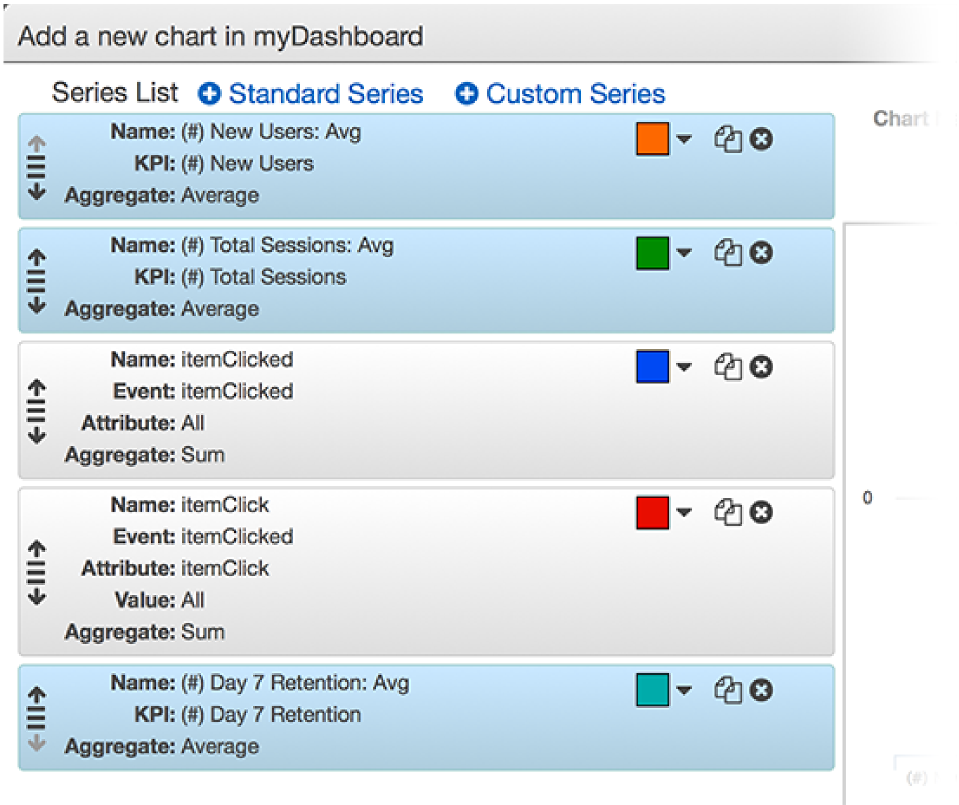
Series List

**Name:** (#) New Users: Avg

**KPI:** (#) New Users

**Aggregate:** Average

15. After you have added several data series to the chart, you can change their order by dragging and dropping them within the editor or by selecting the up or down arrows to move a data series up or down one position.



Add a new chart in myDashboard

Series List ⊕ Standard Series ⊕ Custom Series

Name: (#) New Users: Avg
KPI: (#) New Users
Aggregate: Average

Name: (#) Total Sessions: Avg
KPI: (#) Total Sessions
Aggregate: Average

Name: itemClicked
Event: itemClicked
Attribute: All
Aggregate: Sum

Name: itemClick
Event: itemClicked
Attribute: itemClick
Value: All
Aggregate: Sum

Name: (#) Day 7 Retention: Avg
KPI: (#) Day 7 Retention
Aggregate: Average

16. After you finish setting up the new chart, select **Save Chart** to save the chart to your dashboard.

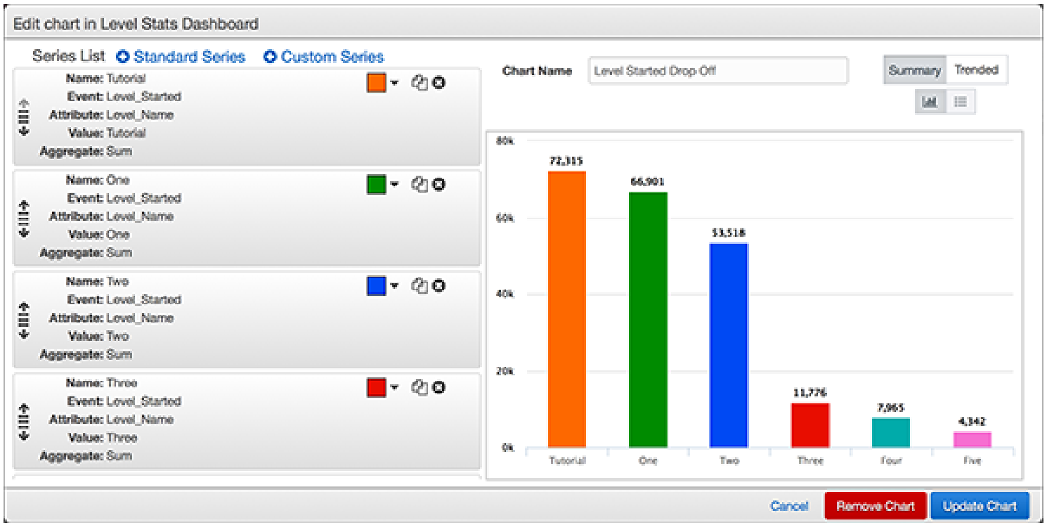
# Editing Dashboards

You can edit your dashboards to add or delete charts as needed. You can also edit individual charts in a dashboard in order to:

- Add new standard or custom data series and delete a data series
- Change the colors used to display a data series
- Reorder the presentation of data series
- Rename the chart or any of the data series in a chart
- Change the KPIs or custom event attributes displayed

### To edit a custom dashboard chart

1. In the **Custom Dashboards** tab, display the dashboard you want to edit.
2. Select the edit chart icon ✎ in the title bar of the chart you want to edit.
3. In the **Edit chart** dialog, make any changes to the data series (display color, name, order, and so on) as needed.



Edit chart in Level Stats Dashboard

Series List

**Name:** Tutorial
**Event:** Level_Started
**Attribute:** Level_Name
**Value:** Tutorial
**Aggregate:** Sum

**Name:** One
**Event:** Level_Started
**Attribute:** Level_Name
**Value:** One
**Aggregate:** Sum

**Name:** Two
**Event:** Level_Started
**Attribute:** Level_Name
**Value:** Two
**Aggregate:** Sum

**Name:** Three
**Event:** Level_Started
**Attribute:** Level_Name
**Value:** Three
**Aggregate:** Sum

**Chart Name** Level Started Drop Off

4. Select **Update Chart** whenever you want to save the changes you've made. Select **Cancel** to leave the editor without saving any unsaved changes. Select **Remove chart** to delete the chart from the dashboard.

# Amazon Mobile Analytics Auto Export

You can use Amazon Mobile Analytics to collect, visualize, and understand app usage data at scale. Mobile Analytics can automatically export event data from your apps to an Amazon S3 bucket or an Amazon Redshift cluster using an AWS CloudFormation stack created in your account. This data auto export makes it possible for you to analyze app usage in detail and combine this information with other business intelligence data.

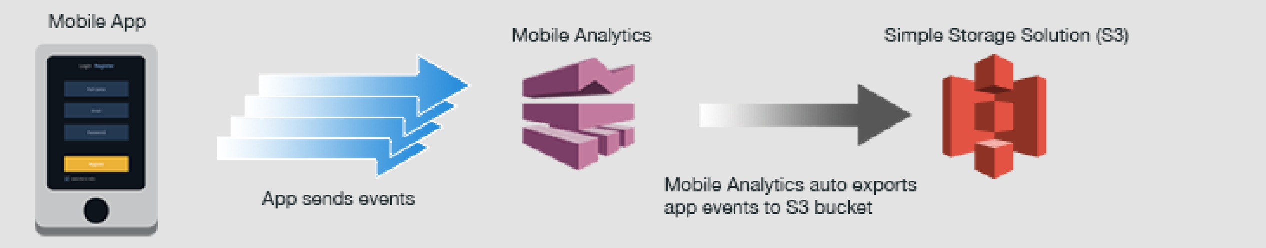
Amazon S3 buckets provide secure, durable, highly scalable object storage. Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse solution that makes it simple and cost-effective to efficiently analyze all of your data using your existing business intelligence tools. AWS CloudFormation provides an easy way to create and manage a collection of related AWS resources.

If you have not already done so, first create an app and generate events before setting up auto export. For more information about creating and app and generating events, see Getting Started (p. 4).

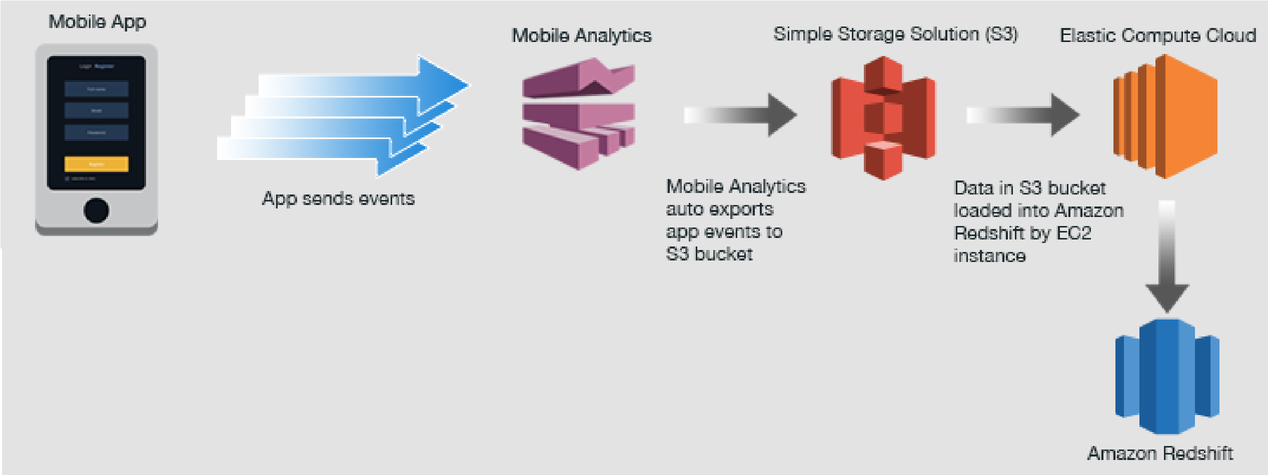
## Auto Export Overview

The analytics collected by Mobile Analytics result from events sent by your app that are collected, aggregated, and then used to populate the console reports and custom dashboards. Mobile Analytics retains this data for 25 months; however, you can't directly access this data for custom analysis. To keep this data for longer than 25 months or to perform custom analysis using your own custom queries and tools, you must export the data for storage.

Mobile Analytics provides an auto export capability that stores your incoming data, either to an Amazon S3 bucket alone or to both an Amazon S3 bucket and an Amazon Redshift cluster. Mobile Analytics will create the Amazon S3 bucket or Amazon Redshift cluster when you set up auto export, or you can use an existing Amazon S3 bucket or Amazon Redshift cluster.


Mobile App
App sends events
Mobile Analytics
Mobile Analytics auto exports app events to S3 bucket
Simple Storage Solution (S3)

OR


Mobile App
App sends events
Mobile Analytics
Mobile Analytics auto exports app events to S3 bucket
Simple Storage Solution (S3)
Data in S3 bucket loaded into Amazon Redshift by EC2 instance
Elastic Compute Cloud
Amazon Redshift

By default, auto export of your data is not enabled. You can choose to enable auto export any time, either for specific apps individually or for all of your apps. Data export for an app begins when auto export is enabled for that app, retaining your data from that time forward.

Mobile Analytics auto exports using a single configuration, so when you set up auto export for the first time you must choose between export to Amazon S3 alone or to both Amazon S3 and Amazon Redshift. That configuration, including the destination Amazon S3 bucket or Amazon Redshift cluster, then applies to the auto export of all apps for which you have enabled auto export.

When exporting to Amazon Redshift, an Amazon EC2 instance is created. This instance reads the data that is written to the Amazon S3 bucket and then loads that data into the Amazon Redshift cluster.

# Writing Events to Amazon S3

For each hour that Mobile Analytics receives event data, it writes the data as a gzip (.gz) archive file to your Amazon S3 bucket. If the volume of data for an hour is high, Mobile Analytics might write the file for that hour in multiple parts.

Mobile Analytics writes the files to Amazon S3 with the following naming convention:

```
bucket-name/awsma/events/appId/YYYY/MM/DD/hh/appId-mm-part-partNum-hexCode.gz
```

**bucket-name**
: The name of the destination Amazon S3 bucket.

**appId**
: The application ID GUID.

**YYYY**
The four-digit year.

**MM**
The two-digit month.

**DD**
The two-digit day.

**hh**
The two-digit hour of day.

**mm**
The two-digit minute.

**partNum**
The four-digit part number of the event archive.

**hexCode**
An opaque 32 character hexadecimal string.

The files are written *at least once*, which means that Mobile Analytics will usually write each file once, but in some cases it will rewrite a file with the same name and contents. Consequently, if Amazon S3 event notifications are enabled, you might receive multiple notifications for a file that is written multiple times. To handle these cases, design all processes that consume these files to handle repeated notifications.

# Getting Started with Auto Export to Amazon Redshift or Amazon S3

The S3 bucket and Amazon Redshift cluster must be in the US East (N. Virginia) Region.

## Step 1: Go to Manage Apps

From the Amazon Mobile Analytics console, choose **App Management**, and then choose **Manage Apps**.

## Step 2: Launch the Auto Export Wizard

Choose the apps for which you would like to enable auto export, and choose **Enable Auto Export**. Alternatively, you can select **Enable Auto Export for All Apps** to export all apps.

## Step 3: Choose Export to Amazon Redshift + Amazon S3

Under **Amazon S3**, choose **Get Started**.

## Step 4: Configure an Amazon S3 Bucket

Choose **Create a new S3 bucket** or use the **Select an existing bucket** drop-down list to choose an existing bucket. Regardless of which option you choose, you will own the S3 bucket and will be responsible for all charges incurred. For more information, see S3 Pricing. You can go to the S3 console at any time to delete the data in your S3 bucket.

# Step 5: Configure S3 and S3 to Amazon Redshift Permissions

Mobile Analytics uses AWS Identity and Access Management (IAM) roles to securely access your S3 bucket through a cross account role. Choose **Select/Create Role**. By default, Mobile Analytics creates a new IAM role and policy with write access to the S3 bucket you selected in the previous step. We recommend you use the default setting to create a new IAM role. If you want to specify a different role or policy, choose **View Details**, and then choose the IAM role or policy. If you choose an existing policy, be sure it provides access to the S3 bucket you will be exporting to. Choose **Allow** to allow access to your Amazon S3 bucket.

Choose **Create Export**. Upon confirmation, your auto export setup will be complete. It can take up to one hour for events to appear in your S3 bucket.

In addition to S3, auto export to Amazon Redshift requires an EC2 IAM role. This role is used by the EC2 instance created in your account to read data from S3, and communicate with other AWS resources, such as CloudWatch and Amazon Redshift. Choose **Select/Create Role**. By default, Mobile Analytics creates a new IAM role and policy that allows an EC2 instance in your account to access the S3 bucket you selected in the previous step, as well as other AWS resources. We recommend that you use the default setting to create a new IAM role. If you want to specify a different role or policy, choose **View Details** and choose the IAM role or policy. If you choose an existing policy, be sure it provides the following:

- Read and write access to the S3 bucket you will be exporting to.
- Access to "cloudwatch:PutMetricData".
- Access to "logs:*".

Choose **Allow** to use the EC2 role.

# Step 6: Configure Auto Export to Use Amazon Redshift

Amazon Mobile Analytics uses AWS CloudFormation to create a new Amazon Redshift cluster and EC2 instance in your account. Enter a password for the "master" user, which will have full administrative permissions to your Amazon Redshift cluster. Keep this password safe. Enter a password for the "`eventreader`" user, which will have read-only permissions to your Amazon Redshift cluster. Distribute these credentials to anyone who needs read-only access to your data. To prevent accidental changes, we recommend you use the `eventreader` user when you query Amazon Redshift.

If your app records custom attributes or metrics, type the name for each attribute or metric, and press **Enter**. Alternatively, you can paste a newline-separated or comma-separated list of values.

> **Note**
> You must include the names of any custom attributes or metrics when you create the auto export for their data to appear in Amazon Redshift. Any attributes or metrics you add after creating the auto export may require recreating the auto export in order to appear in Amazon Redshift, thereby losing any event data already stored in Amazon Redshift.

The EC2 instance created in your account can record metrics, such as the length of time to load events into Amazon Redshift or whether the load was successful. In addition, the EC2 instance copies all logs to CloudWatch Logs. You can enable CloudWatch Metrics and Logs by selecting **Enable CloudWatch Metrics and Logs**.

Choosing **Advanced Options** allows you to specify settings to connect to your Amazon Redshift cluster and EC2 instance:

- **IP access rules** - By default, your Amazon Redshift cluster and EC2 instance will be accessible from your machine only. To make your Amazon Redshift cluster and EC2 instance available to all internet addresses, choose **Accessible from any IP Address**. You can also specify a custom Classless Inter-Domain Routing (CIDR) rule to allow access to addresses from within your organization. For more information about CIDR notation, see Classless Inter-Domain Routing on Wikipedia.
- **Port number** - The port number used by Amazon Redshift to communicate.
- **Password for the etl_user account used for loading your data from S3 to Amazon Redshift** - This is the write-only password used by the EC2 instance to load data into Amazon Redshift. We recommend you use the generated password.
- **An EC2 key pair used for connecting to the EC2 instance using SSH** - If you want to connect to the EC2 instance by using SSH (for example, to do custom log pulling), you must specify an EC2 key pair. You can create one in the EC2 console. You cannot change the key pair after the EC2 instance has started.

Each of the advanced options has a default value, so you do not need to specify a value unless you want a value different than the default. You can choose **Advanced Options** to see the default values.

Choose **Create Export**. It can take up to 15 minutes for your Amazon Redshift cluster to be provisioned.

> **Note**
> The EC2 instance will run continuously (and incur charges) regardless of whether new event data is being sent by your app.

# Connecting to Amazon Redshift

Amazon Redshift provides multiple ways to connect to an Amazon Redshift cluster, including programmatic access via a JDBC connection, or querying Amazon Redshift directly on your workstation using SQLWorkbench/J. For more information, see Connecting to an Amazon Redshift Cluster.

When logging in to your Amazon Redshift cluster, use the Amazon Redshift connection string which is located in the Amazon Redshift console on your cluster's configuration page. For more information, see Configuring Amazon Redshift Connections. Then use the `eventreader` user name and the `eventreader` password you specified in the auto export configuration to connect to the cluster.

# Adding New Metrics and Attributes to an Auto Export to Amazon Redshift

Event data exported to Amazon S3 contains all of the attributes and metrics that were sent to the service. There are two options you can use to manage changes to event attributes and metrics in your Amazon Redshift cluster:

- Recreate auto export to a new Amazon Redshift cluster.
- Add or remove attributes that are loaded to your existing Amazon Redshift cluster with AWS CloudFormation

### Option 1: To recreate auto export to a new Amazon Redshift cluster

> **Warning**
> Using this method results in loss of any other data loaded into Amazon Redshift. Do not stop the auto export feature in the Mobile Analytics console while completing this method.

1. Sign in to the AWS Management Console and open the AWS CloudFormation console at https://console.aws.amazon.com/cloudformation/.
2. Select the stack containing the Amazon Redshift cluster. This stack is named "MobileAnalyticsAutoExportToRedshift" in most cases.
3. Choose **Delete Stack**.
4. Navigate to the Mobile Analytics console and configure the auto export to Amazon Redshift feature (p. 91). After completing this step, your Amazon Redshift cluster is created and populated with existing data in your Amazon S3 bucket.

### Option 2: To add or remove attributes being loaded into your existing Amazon Redshift cluster with AWS CloudFormation

This method updates the tables in your Amazon Redshift cluster to include new attributes and metrics columns. It also updates the AWS CloudFormation stack to include these new attributes and metrics so they are loaded into these new columns in the Amazon Redshift cluster. Any new attributes and metrics are loaded into the Amazon Redshift cluster for new events received after completing these steps. Existing events in your Amazon Redshift cluster are not updated.

> **Note**
> Using this method requires you to terminate the EC2 instance that was created with the auto export to Amazon Redshift feature.

1. Connect to your Amazon Redshift cluster with a user account that has sufficient privileges (for example, master privileges) using your SQL client of choice. For more information about connecting to an Amazon Redshift cluster using third-party tools, see Connecting to Clusters From Client Tools and Code.
2. For each new attribute column you add for the event attribute (for example, “myNewAttribute”), run these commands:

   ```
   alter table awsma.event_staging
   add column a_myNewAttribute VARCHAR(4000) NULL ENCODE LZO;
   alter table awsma.event
   add column a_myNewAttribute VARCHAR(4000) NULL ENCODE LZO;
   ```

3. For each new metric column you add for the event metric (for example, “myNewMetric”), run these commands:

   ```
   alter table awsma.event_staging
   add column m_myNewMetric FLOAT8 NULL;
   alter table awsma.event
   add column m_myNewMetric FLOAT8 NULL;
   ```

4. After all attributes and metrics are added, run these commands to update the v_event view so it contains the new columns.

   ```
   drop view AWSMA.v_event;
   CREATE OR REPLACE VIEW AWSMA.v_event AS select * from AWSMA.event;
   ALTER TABLE AWSMA.v_event OWNER to etl_user;
   grant select on AWSMA.v_event to group eventReaders;
   ```

5. Update configuration with AWS CloudFormation:

a. In the AWS CloudFormation console, choose the stack containing the Amazon Redshift Cluster (usually named "MobileAnalyticsAutoExportToRedshift").
b. Choose **Update Stack**.
c. Choose **Next**.
d. Find the `EventCustomMetrics` and `EventCustomAttributes` parameters then add your new attributes and metrics as comma separated values.
e. Find `RedshiftEtlPassword`, `RedshiftMasterPassword` and `RedshiftReadPassword` and then select **Use Existing Value** for each.
f. Complete the rest of the update wizard.

6. Terminate your Amazon EC2 instance so the Auto Scaling group creates a new instance with the updated configuration:

   a. In the Amazon EC2 console, choose **Auto Scaling Groups**.
   b. Choose the “MobileAnalyticsAutoExportToRedshift” Auto Scaling group.
   c. In the detail view select the **Instances** tab.
   d. Choose **Instance ID**.
   e. In **Actions** first choose **Instance State** and then choose **Terminate**.

The Auto Scaling group creates a new instance.

# Managing Auto Export

After you set up auto export, you may want to enable export for additional apps or disable auto export for an app. You may also want to add new metrics and attributes to an AutoExport to Amazon Redshift.

## Enabling Auto Export for Additional Apps

After you set up auto export, you can automatically export data from other apps. Go to the App Management page, choose the apps to enable, choose the **Configure Auto Export to S3** or the **Configure Auto Export to Amazon Redshift + S3** drop-down list (depending on how you configured auto export), and choose **Enable Auto Export**. The apps will be added to your auto export configuration and their data will begin appearing in your S3 bucket or Amazon Redshift cluster.

> **Note**
> Only one auto export configuration can be used at one time. If you chose the Amazon Redshift + Amazon S3 option, all of your apps configured to use auto export will write event data to Amazon Redshift and Amazon S3.

## Disabling Auto Export

You can disable auto export for one or more apps. From the App Management page, choose the apps to disable, choose **Configure Auto Export to S3** or **Configure Auto Export to Redshift + S3** (depending on how you configured auto export), and choose **Disable Auto Export**. After you've confirmed the action, data for the selected apps will stop flowing to your S3 bucket. Disabling auto export will not delete your S3 bucket, the data contained in it, your Amazon Redshift cluster, or your EC2 instance. You will continue to incur charges until you delete or terminate each resource.

To remove your Amazon Redshift cluster and EC2 instance so that you no longer incur charges, go to the CloudFormation console and delete the stack Mobile Analytics created on your behalf. This operation will delete all resources and cannot be reversed.

To remove data in S3, go to the S3 console and remove it manually.

# Exporting to an Existing Amazon Redshift Cluster

By default, Mobile Analytics auto export to Amazon Redshift creates an Amazon Redshift cluster specifically to receive exported analytics data. However, if you have an existing Amazon Redshift cluster you would rather use instead of the cluster created by Mobile Analytics, you can auto export to that cluster.

Make sure you complete the steps to set up export of your app's events to Amazon Redshift (p. 91) except for the creation of a new Amazon Redshift cluster. You don't need to create a new Amazon Redshift cluster if you plan to use an existing cluster.

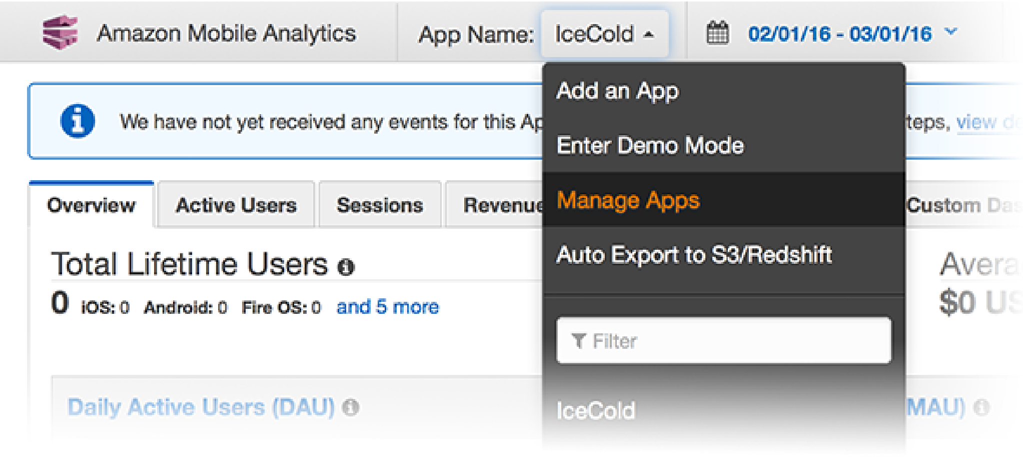
The process of setting up auto export to an existing Amazon Redshift cluster has these parts:

Topics

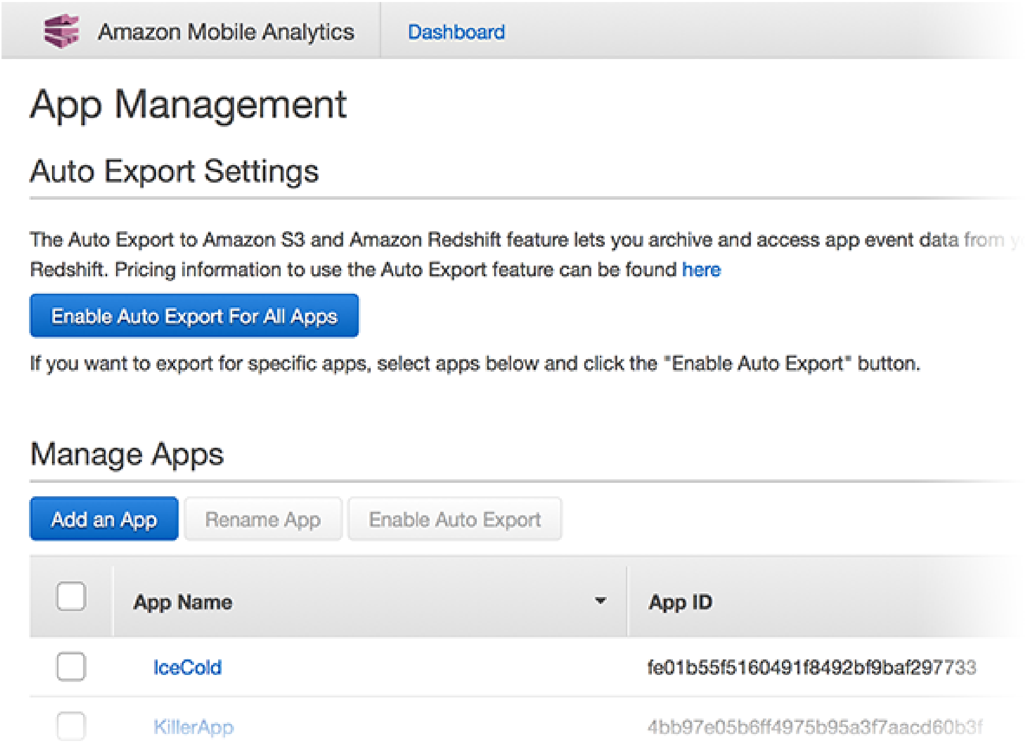
## Part 1: Configure Security Roles for Auto Export to Amazon Redshift

The first thing you need to do to auto export your app analytics to an existing Amazon Redshift cluster is configure the auto export security roles.

1. Open the Amazon Mobile Analytics console at https://console.aws.amazon.com/mobileanalytics/home.
2. Choose **Manage Apps** from the app drop-down menu.



3. Enable auto export for all apps by selecting **Enable Auto Export For All Apps**. You can enable auto export for a single app by choosing the app in the **Manage Apps** list and then choosing **Enable Auto Export**.



App Management

Auto Export Settings

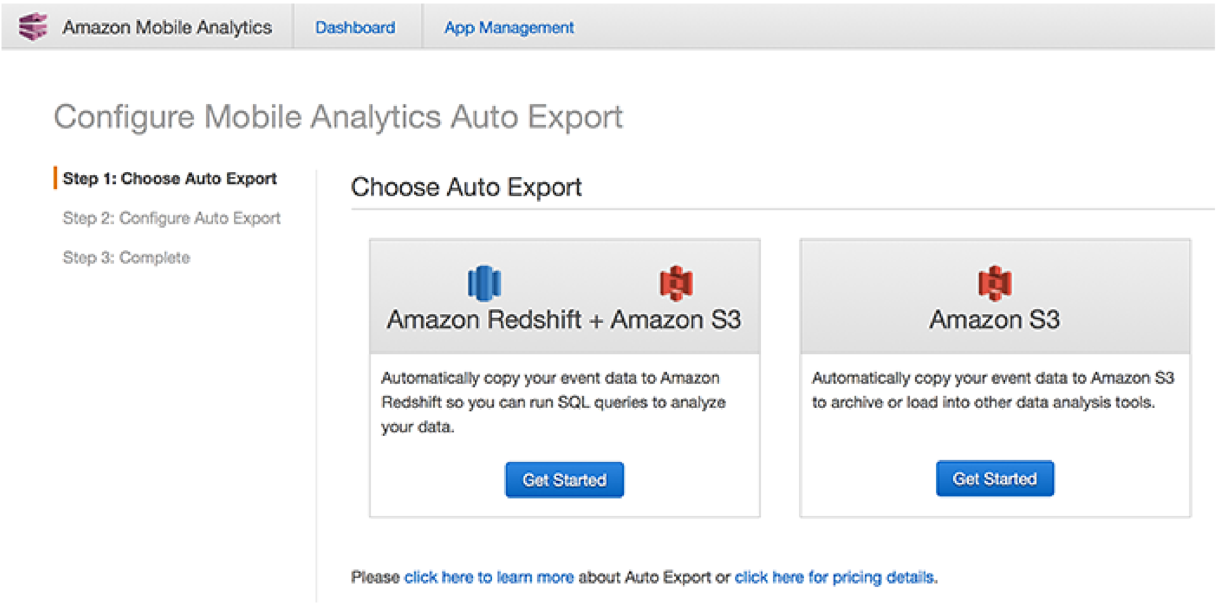
The Auto Export to Amazon S3 and Amazon Redshift feature lets you archive and access app event data from y
Redshift. Pricing information to use the Auto Export feature can be found here

If you want to export for specific apps, select apps below and click the "Enable Auto Export" button.

Manage Apps

| □ | App Name | App ID |
|---|---|---|
| □ | IceCold | fe01b55f5160491f8492bf9baf297733 |
| □ | KillerApp | 4bb97e05b6ff4975b95a3f7aacd60b3f |

4. In **Choose Auto Export**, choose **Get Started** in **Amazon Redshift + Amazon S3**.



Configure Mobile Analytics Auto Export

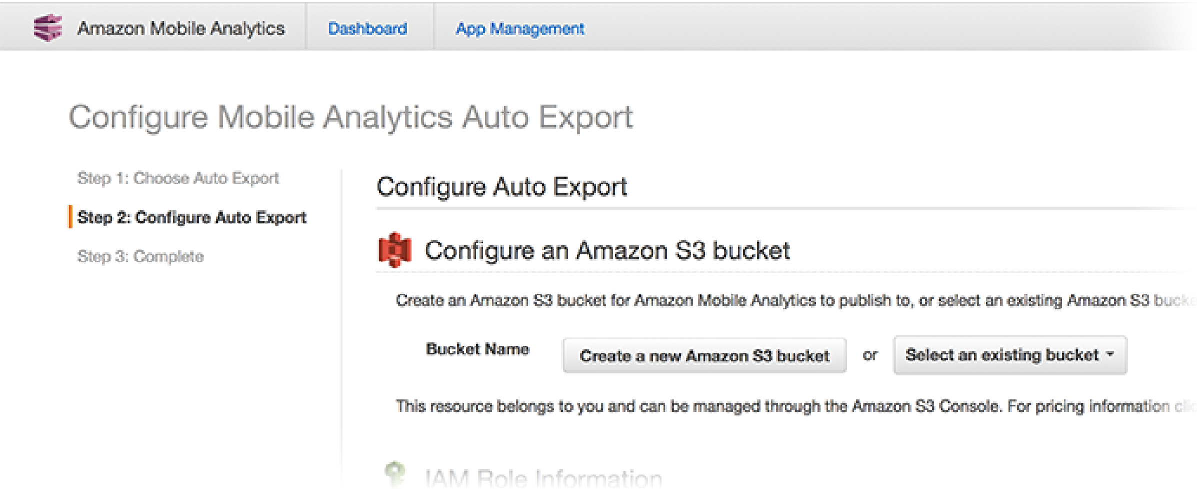
Choose Auto Export

Automatically copy your event data to Amazon Redshift so you can run SQL queries to analyze your data.

Automatically copy your event data to Amazon S3 to archive or load into other data analysis tools.

Please click here to learn more about Auto Export or click here for pricing details.

5. In **Configure Auto Export**, choose **Create a new Amazon S3 bucket** or select an existing Amazon S3 bucket from the drop-down list.


Amazon Mobile Analytics
Dashboard
App Management

Configure Mobile Analytics Auto Export

Step 1: Choose Auto Export
Step 2: Configure Auto Export
Step 3: Complete

Configure Auto Export

Configure an Amazon S3 bucket

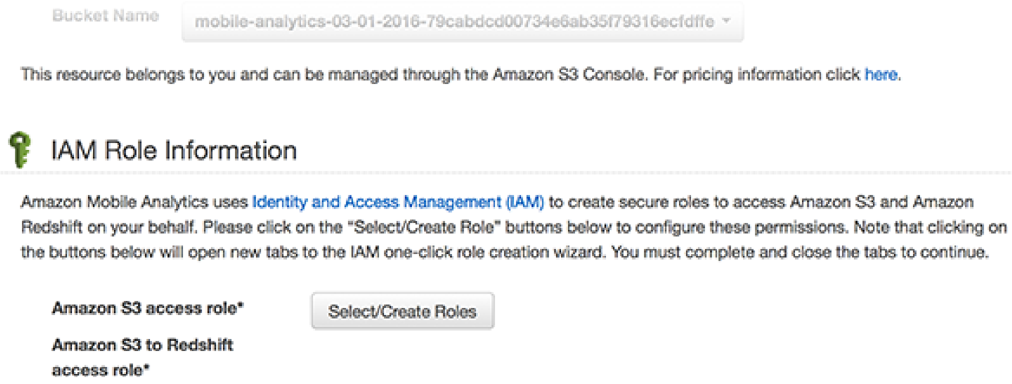
Create an Amazon S3 bucket for Amazon Mobile Analytics to publish to, or select an existing Amazon S3 buck

Bucket Name
Create a new Amazon S3 bucket
or
Select an existing bucket

This resource belongs to you and can be managed through the Amazon S3 Console. For pricing information cli

IAM Role Information

6. In **IAM Role Information**, choose **Select/Create Role**.

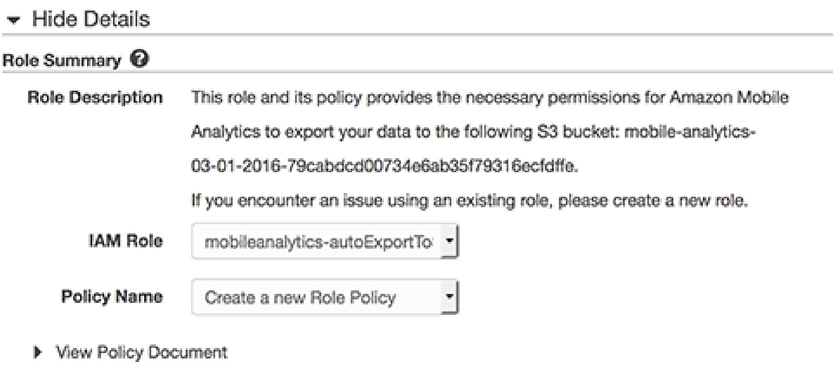

Bucket Name: mobile-analytics-03-01-2016-79cabdcd00734e6ab35f79316ecfdffe

This resource belongs to you and can be managed through the Amazon S3 Console. For pricing information click here.

IAM Role Information

Amazon Mobile Analytics uses Identity and Access Management (IAM) to create secure roles to access Amazon S3 and Amazon Redshift on your behalf. Please click on the "Select/Create Role" buttons below to configure these permissions. Note that clicking on the buttons below will open new tabs to the IAM one-click role creation wizard. You must complete and close the tabs to continue.

7. Select **View Details** and make a note of the name generated for the IAM role. Then choose **Allow**.



Role Description: This role and its policy provides the necessary permissions for Amazon Mobile Analytics to export your data to the following S3 bucket: mobile-analytics-03-01-2016-79cabdcd00734e6ab35f79316ecfdffe.

If you encounter an issue using an existing role, please create a new role.

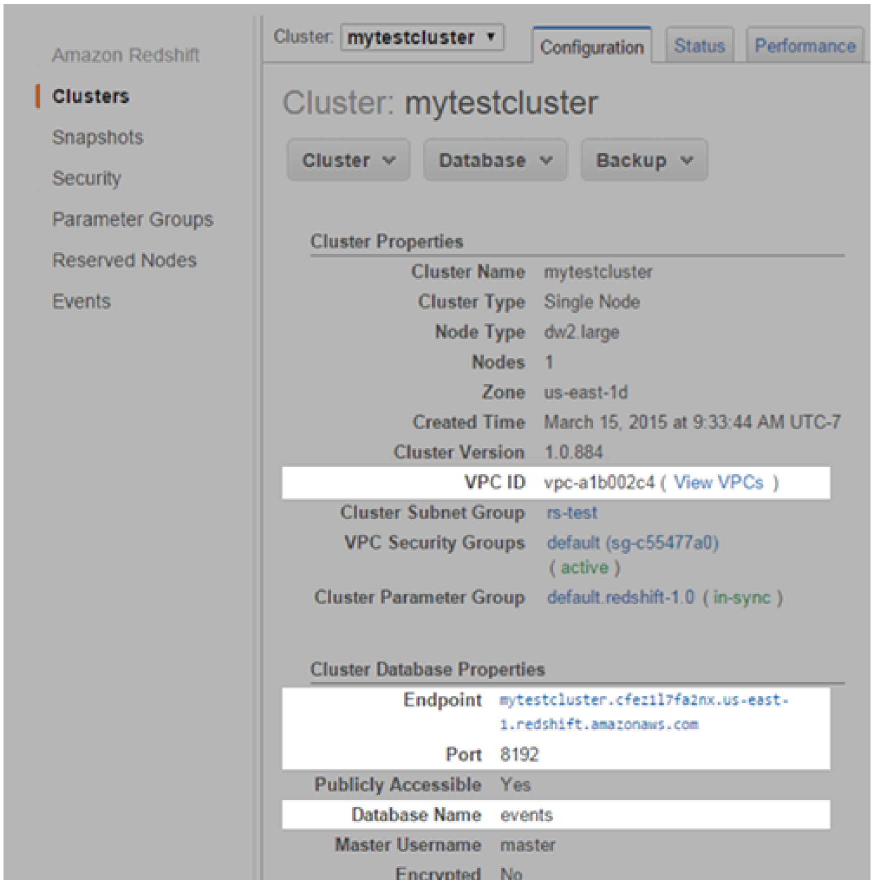
IAM Role: mobileanalytics-autoExportTo

Policy Name: Create a new Role Policy

# Part 2: Gather Details About the Amazon Redshift Cluster

After you configure the auto export security roles, you must gather details about your Amazon Redshift cluster. You will need these details later.

1. Open the Amazon Redshift console at https://console.aws.amazon.com/redshift/home.
2. Under **Clusters**, find and choose the existing Amazon Redshift cluster you want to use for auto export from Mobile Analytics to display the details of that cluster.
3. Make note of the **VPC ID**, **Endpoint**, **Port**, and **Database Name** values for use later.



Cluster: mytestcluster

Cluster Properties

| | |
|---|---|
| Cluster Name | mytestcluster |
| Cluster Type | Single Node |
| Node Type | dw2.large |
| Nodes | 1 |
| Zone | us-east-1d |
| Created Time | March 15, 2015 at 9:33:44 AM UTC-7 |
| Cluster Version | 1.0.884 |
| VPC ID | vpc-a1b002c4 ( View VPCs ) |
| Cluster Subnet Group | rs-test |
| VPC Security Groups | default (sg-c55477a0)<br>( active ) |
| Cluster Parameter Group | default.redshift-1.0 ( in-sync ) |

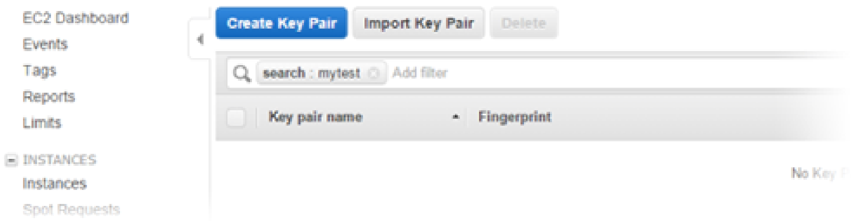
Cluster Database Properties

| | |
|---|---|
| Endpoint | mytestcluster.cfez1l7fa2nx.us-east-1.redshift.amazonaws.com |
| Port | 8192 |
| Publicly Accessible | Yes |
| Database Name | events |
| Master Username | master |
| Encrypted | No |

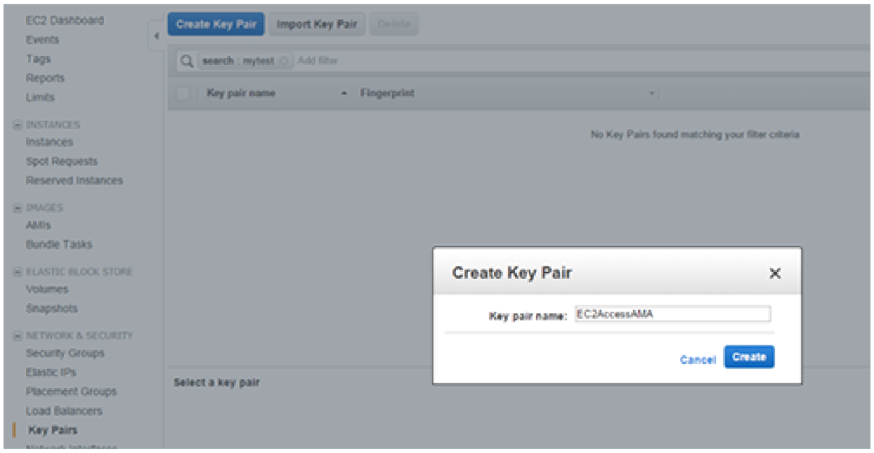
# Part 3: Create and Configure an Amazon EC2 Loader Instance

After gathering the details on your Amazon Redshift cluster, you must create and configure your Amazon EC2 loader instance.

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/v2/home.
2. Ensure that you have created a key pair.



3. Name the key pair.

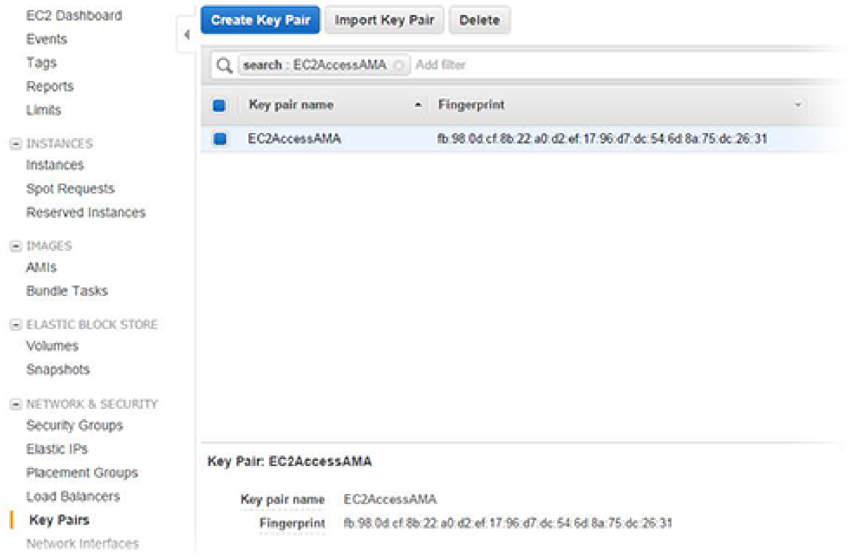


No Key Pairs found matching your filter criteria

Create Key Pair

Key pair name: EC2AccessAMA

4. Download the credentials for the key pair.



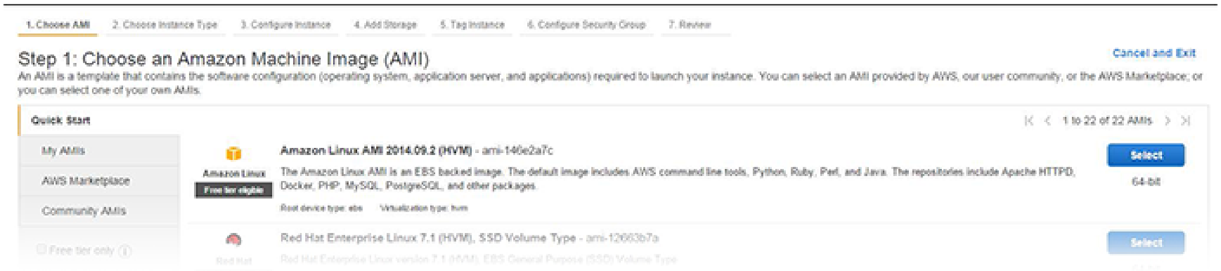
search : EC2AccessAMA

| Key pair name | Fingerprint |
|---|---|
| EC2AccessAMA | fb:98:0d:cf:8b:22:a0:d2:ef:17:96:d7:dc:54:6d:8a:75:dc:26:31 |

Key Pair: EC2AccessAMA

Key pair name EC2AccessAMA
Fingerprint fb:98:0d:cf:8b:22:a0:d2:ef:17:96:d7:dc:54:6d:8a:75:dc:26:31

5. Choose **Instances** from the Amazon EC2 console navigation. Then choose **Launch Instance** and choose **Amazon Linux AMI** from the list of available AMIs.



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Amazon Linux AMI 2014.09.2 (HVM) - ami-146e2a7c

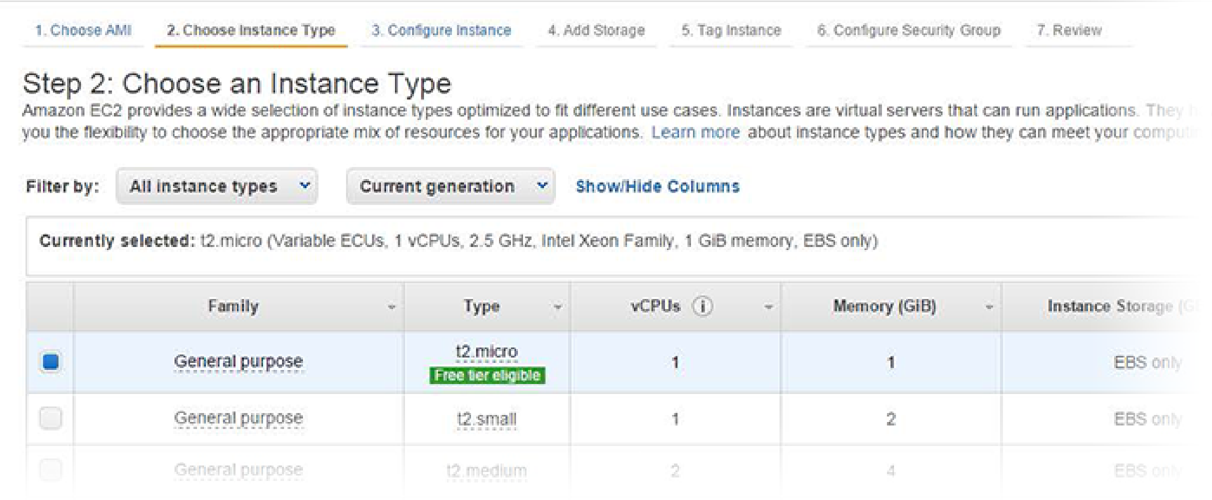
The Amazon Linux AMI is an EBS backed image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Apache HTTPD, Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs Virtualization type: hvm

Red Hat Enterprise Linux 7.1 (HVM), SSD Volume Type - ami-12663b7a

Red Hat Enterprise Linux version 7.1 (HVM), EBS General Purpose (SSD) Volume Type

6. Choose the **t2.micro** instance type.



Step 2: Choose an Instance Type

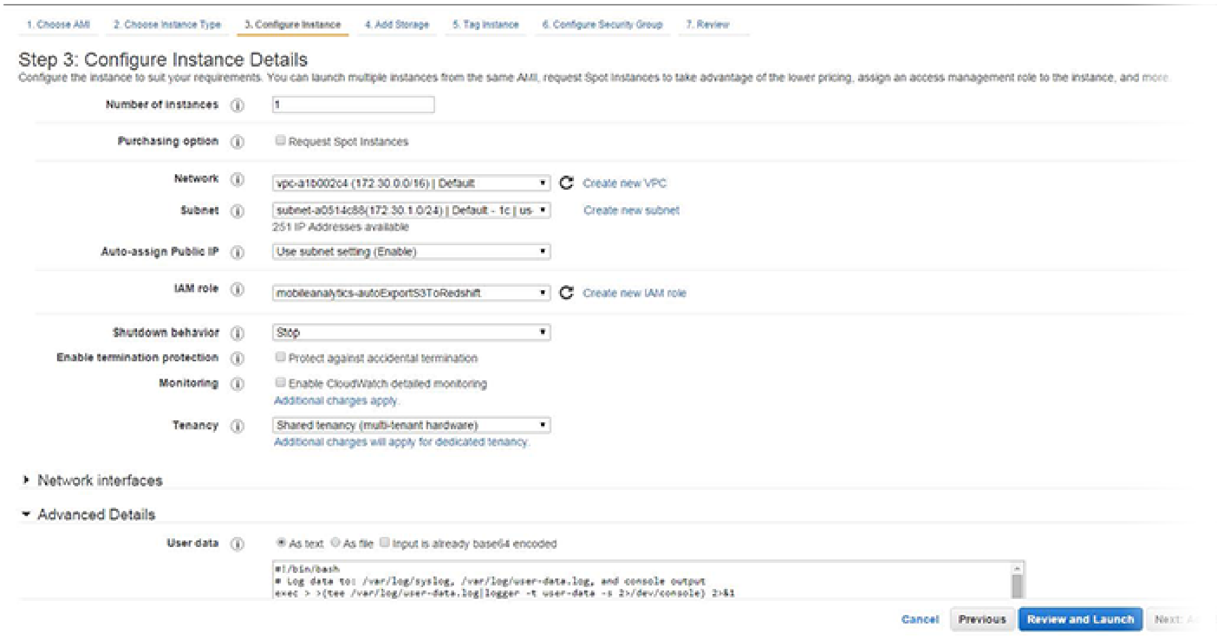
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They h you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your comput

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

| | Family | Type | vCPUs | Memory (GiB) | Instance Storage (G |
|---|---|---|---|---|---|
| ■ | General purpose | t2.micro<br>Free tier eligible | 1 | 1 | EBS only |
| □ | General purpose | t2.small | 1 | 2 | EBS only |
| □ | General purpose | t2.medium | 2 | 4 | EBS only |

7. Configure your new Amazon EC2 loader instance. Choose a VPC, ideally the VPC you set up in Part 2: Gather Details About the Amazon Redshift Cluster (p. 99). Choose **Enable** from the **Auto-assign Public IP** list. Choose the IAM role you created in Part 1: Configure Security Roles for Auto Export to Amazon Redshift (p. 96).



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more

Number of instances: 1

Purchasing option: □ Request Spot Instances

Network: vpc-a1b002c4 (172.30.0.0/16) | Default

Subnet: subnet-a0514c88(172.30.1.0/24) | Default - 1c | us-
251 IP Addresses available

Auto-assign Public IP: Use subnet setting (Enable)

IAM role: mobileanalytics-autoExportS3ToRedshift

Shutdown behavior: Stop

Enable termination protection: □ Protect against accidental termination

Monitoring: □ Enable CloudWatch detailed monitoring
Additional charges apply.

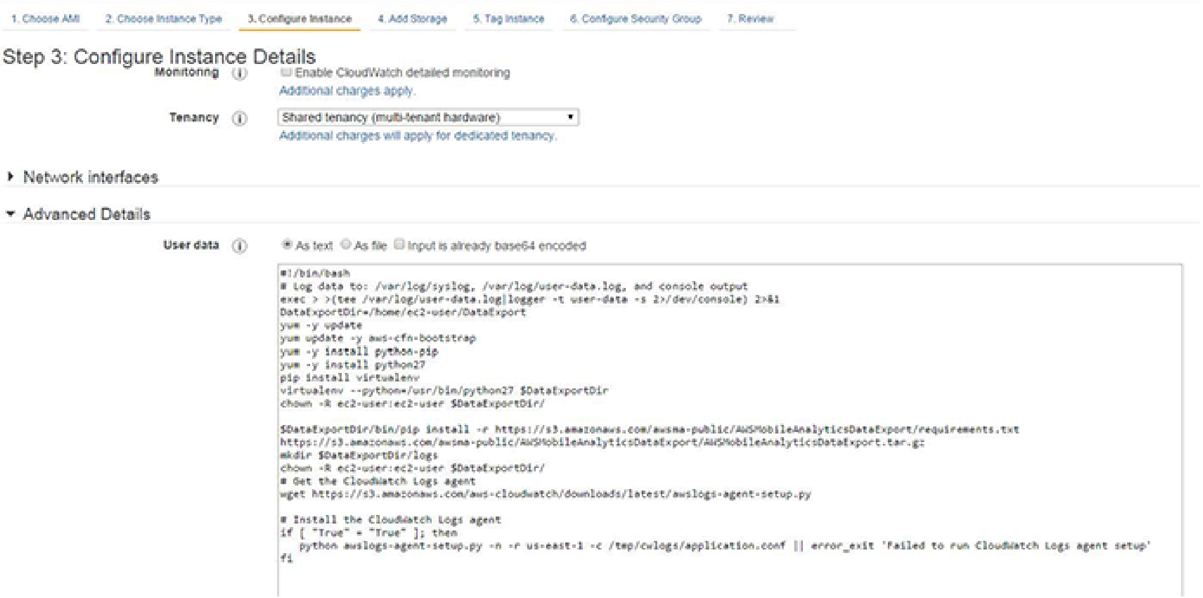
Tenancy: Shared tenancy (multi-tenant hardware)
Additional charges will apply for dedicated tenancy.

User data: ● As text ○ As file □ Input is already base64 encoded

```
#!/bin/bash
# Log data to: /var/log/syslog, /var/log/user-data.log, and console output
exec > >(tee /var/log/user-data.log|logger -t user-data -s 2>/dev/console) 2>&1
```

8. Expand **Advanced Details** and then copy and paste the following text into the **User data** section:

```
#!/bin/bash
# Log data to: /var/log/syslog, /var/log/user-data.log, and console output
exec > >(tee /var/log/user-data.log|logger -t user-data -s 2>/dev/console)
 2>&1
DataExportDir=/home/ec2-user/DataExport
yum -y update
yum -y install python-pip
yum -y install python27
pip install virtualenv
virtualenv --python=/usr/bin/python27 $DataExportDir
chown -R ec2-user:ec2-user $DataExportDir/
$DataExportDir/bin/pip install -r https://s3.amazonaws.com/awsma-
public/AWSMobileAnalyticsDataExport/1.0/latest/requirements.txt https://
s3.amazonaws.com/awsma-public/AWSMobileAnalyticsDataExport/1.0/latest/
AWSMobileAnalyticsDataExport.tar.gz
mkdir $DataExportDir/logs
chown -R ec2-user:ec2-user $DataExportDir/
# Get the CloudWatch Logs agent
wget https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-
agent-setup.py
```



Step 3: Configure Instance Details

Monitoring □ Enable CloudWatch detailed monitoring

Additional charges apply.

Tenancy: Shared tenancy (multi-tenant hardware)

Additional charges will apply for dedicated tenancy.

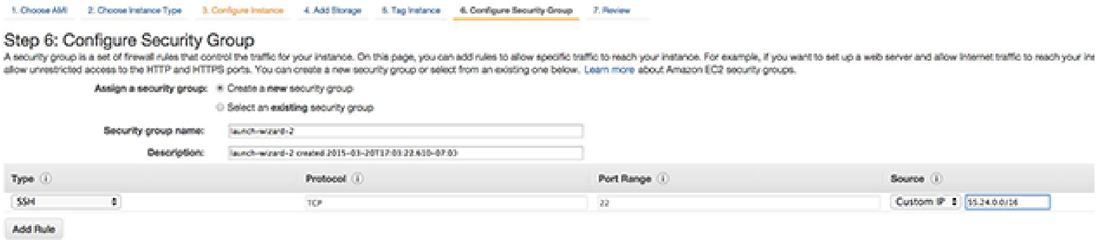
▸ Network interfaces

▾ Advanced Details

User data: ● As text ○ As file □ Input is already base64 encoded

```
#!/bin/bash
# Log data to: /var/log/syslog, /var/log/user-data.log, and console output
exec > >(tee /var/log/user-data.log|logger -t user-data -s 2>/dev/console) 2>&1
DataExportDir=/home/ec2-user/DataExport
yum -y update
yum update -y aws-cfn-bootstrap
yum -y install python-pip
yum -y install python27
pip install virtualenv
virtualenv --python=/usr/bin/python27 $DataExportDir
chown -R ec2-user:ec2-user $DataExportDir/

$DataExportDir/bin/pip install -r https://s3.amazonaws.com/awsma-public/AWSMobileAnalyticsDataExport/requirements.txt
https://s3.amazonaws.com/awsma-public/AWSMobileAnalyticsDataExport/AWSMobileAnalyticsDataExport.tar.gz
mkdir $DataExportDir/logs
chown -R ec2-user:ec2-user $DataExportDir/
# Get the CloudWatch Logs agent
wget https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py

# Install the CloudWatch Logs agent
if [ "True" = "True" ]; then
   python awslogs-agent-setup.py -n -r us-east-1 -c /tmp/cwlogs/application.conf || error_exit 'Failed to run CloudWatch Logs agent setup'
fi
```

9. On **Add Storage** and **Tag Instance** tabs, choose **Next**, accepting their default settings.
10. On the **Configure Security Group** tab, configure the Security Group CIDR rule to let your IP addresses connect to the Amazon EC2 instance on port 22. You can use an existing security group instead or change the **Source** to permit your IP address only.



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your ins
allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

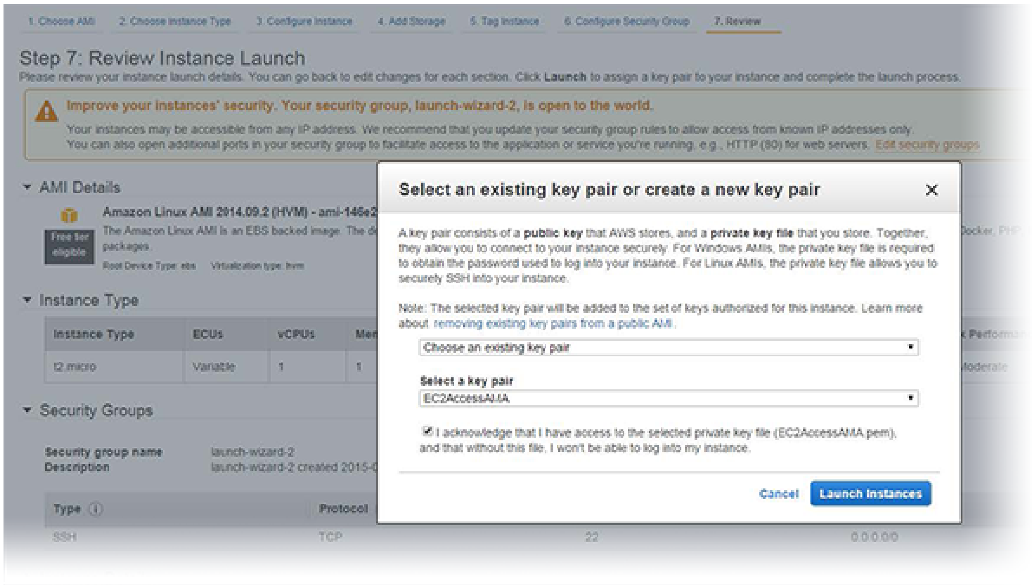
Assign a security group: ● Create a **new** security group
○ Select an **existing** security group

Security group name: launch-wizard-2

Description: launch-wizard-2 created 2015-03-20T17:03:22.610-07:00

| Type | Protocol | Port Range | Source |
|---|---|---|---|
| SSH | TCP | 22 | Custom IP 55.24.0.0/16 |

11. Choose **Review and Launch the Amazon EC2 instance**.
12. For **Select a key pair**, choose the key pair you created previously.



Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

**Improve your instances' security. Your security group, launch-wizard-2, is open to the world.**

Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only.
You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. Edit security groups

AMI Details

**Amazon Linux AMI 2014.09.2 (HVM) - ami-146e2**

The Amazon Linux AMI is an EBS backed image. The de

packages.

Root Device Type: ebs Virtualization type: hvm

Instance Type

| Instance Type | ECUs | vCPUs | Mem |
|---|---|---|---|
| t2.micro | Variable | 1 | 1 |

Security Groups

Security group name launch-wizard-2
Description launch-wizard-2 created 2015-0

| Type | Protocol | | |
|---|---|---|---|
| SSH | TCP | 22 | 0.0.0.0/0 |

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

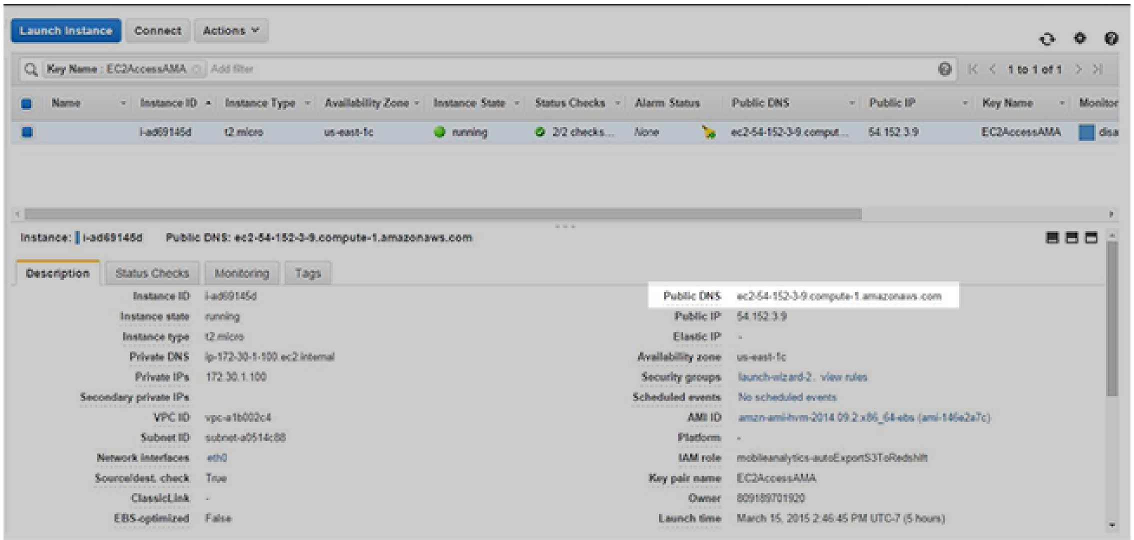
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Choose an existing key pair

**Select a key pair**
EC2AccessAMA

☑ I acknowledge that I have access to the selected private key file (EC2AccessAMA.pem), and that without this file, I won't be able to log into my instance.

13. Choose **Launch Instances**.
14. After the Amazon EC2 instance is created, make note of the value for **Public DNS**, which is required to access the Amazon EC2 instance using SSH or similar tools.



Key Name : EC2AccessAMA

| Name | Instance ID | Instance Type | Availability Zone | Instance State | Status Checks | Alarm Status | Public DNS | Public IP | Key Name | Monitor |
|---|---|---|---|---|---|---|---|---|---|---|
| | i-ad69145d | t2.micro | us-east-1c | running | 2/2 checks... | None | ec2-54-152-3-9.comput... | 54.152.3.9 | EC2AccessAMA | disa |

Instance: i-ad69145d Public DNS: ec2-54-152-3-9.compute-1.amazonaws.com

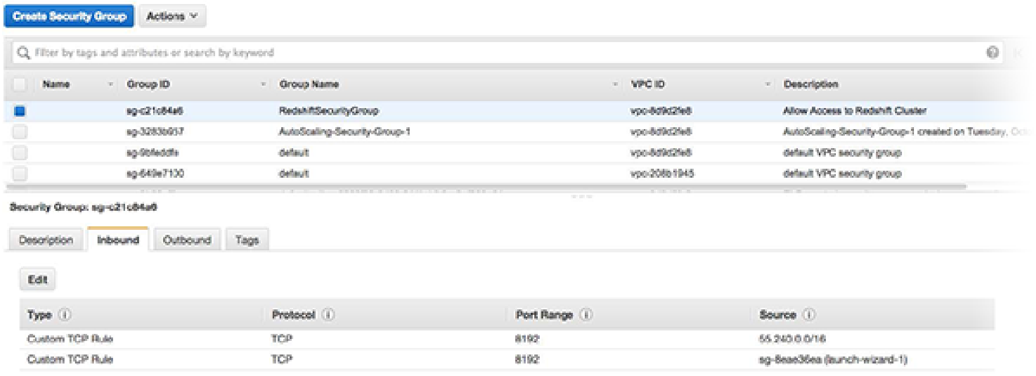
| | | | |
|---|---|---|---|
| Instance ID | i-ad69145d | Public DNS | ec2-54-152-3-9.compute-1.amazonaws.com |
| Instance state | running | Public IP | 54.152.3.9 |
| Instance type | t2.micro | Elastic IP | - |
| Private DNS | ip-172-30-1-100.ec2.internal | Availability zone | us-east-1c |
| Private IPs | 172.30.1.100 | Security groups | launch-wizard-2. view rules |
| Secondary private IPs | | Scheduled events | No scheduled events |
| VPC ID | vpc-a1b002c4 | AMI ID | amzn-ami-hvm-2014.09.2.x86_64-ebs (ami-146e2a7c) |
| Subnet ID | subnet-a0514c88 | Platform | - |
| Network interfaces | eth0 | IAM role | mobileanalytics-autoExportS3ToRedshift |
| Source/dest. check | True | Key pair name | EC2AccessAMA |
| ClassicLink | - | Owner | 809189701920 |
| EBS-optimized | False | Launch time | March 15, 2015 2:46:45 PM UTC-7 (5 hours) |

# Part 4: Configure Amazon EC2 Access to the Amazon Redshift Cluster

After creating and configuring your Amazon EC2 loader instance, you must configure Amazon EC2 access.

Configure your security groups so that the new Amazon EC2 instance can access the Amazon Redshift host and port. The ports that are accessible vary based on the configuration of your security groups.

An easy way to provide access to the Amazon Redshift cluster is to add a new inbound rule to the Amazon Redshift security group. This rule provides the Amazon EC2 instance security group access to the Amazon Redshift port. In the following example, the second of the two rules listed is a new rule that was created to give access to Amazon Redshift.



| | Name | Group ID | Group Name | VPC ID | Description |
|---|---|---|---|---|---|
| ■ | | sg-c21c84a6 | RedshiftSecurityGroup | vpc-8d9d2fe8 | Allow Access to Redshift Cluster |
| □ | | sg-3283b917 | AutoScaling-Security-Group-1 | vpc-8d9d2fe8 | AutoScaling-Security-Group-1 created on Tuesday, Oc |
| □ | | sg-9bfeddfs | default | vpc-8d9d2fe8 | default VPC security group |
| □ | | sg-649e7130 | default | vpc-208b1945 | default VPC security group |

Security Group: sg-c21c84a6

| Type | Protocol | Port Range | Source |
|---|---|---|---|
| Custom TCP Rule | TCP | 8192 | 55.240.0.0/16 |
| Custom TCP Rule | TCP | 8192 | sg-8eae36ea (launch-wizard-1) |

# Part 5: Connect to Amazon EC2 and Configure the Event Loader

After configuring your Amazon EC2 access, you must connect to Amazon EC2.

1. Log in to the Amazon EC2 instance through the AWS Management Console or through a terminal shell. For more information, see Connecting to Your Linux Instance Using SSH.
2. Using a terminal shell, enter the following:

```
$ chmod 400 YOUR_KEY_FILE.pem
$ ssh -i YOUR_KEY_FILE.pem ec2-user@PUBLIC_DNS
```

3. Configure the event loader on Amazon EC2 with the correct settings. Run the following commands, providing values for each of the properties:

```
$ ~/DataExport/bin/configure
Enter value for key: (DataReaderUser.password) -> a_password
Enter value for key: (Redshift.host) -> redshift_host
Enter value for key: (Redshift.database) -> redshift_database
Enter value for key: (Redshift.port) -> redshift_port
Enter value for key: (DataBackfill.startdate) -> s3_export_date
Enter value for key: (CloudWatch.record_duration_metric) -> True
Enter value for key: (EventDetails.custom_attributes) -> attr1,attr2
Enter value for key: (EventDetails.custom_metrics) -> metric1,metric2
Enter value for key: (S3Source.bucketname) -> s3_bucket_name
Enter value for key: (ETLUser.password) -> a_password
```

## Notes

- In the event the `configure` script is not present in `~/DataExport/bin`, you will need to log in to your Amazon EC2 instance and check the contents of `/var/log/user-data.log` to determine why the `configure` script was not downloaded during setup of the Amazon EC2 instance. For example, Python might not have installed correctly.
- Amazon Redshift passwords must contain at least 1 upper case letter and 1 number.

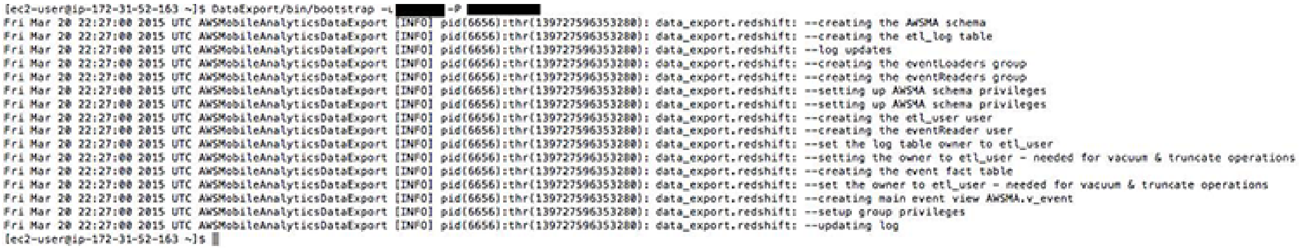
- DataReaderUser.password is the password for the eventreader Amazon Redshift user that's auto-created in the next step.
- The Redshift.host, Redshift.database, and Redshift.port are found on the Amazon Redshift console from Step 2.
- DataBackfill.startdate should be the format YYYY/MM/DD. Select the date you started exporting data to Amazon S3.
- Custom attributes and metrics should be delimited with a comma with no spaces.
- S3Source.bucketname should be just the name of Amazon S3 bucket, for example: `mobile-analytics-02-01-2015-ef5dc9e30e4a49c9a4b0cb5c31629932`
- ETLUser.password is only used by the Amazon EC2 instance.

# Part 6: Create the Schema in Amazon Redshift

Create the schema, tables, views, and users in your Amazon Redshift cluster. Run `bootstrap` to create the Mobile Analytics tables and users in your Amazon Redshift cluster. The `master_username` and `master_password` are the master username and password for your Amazon Redshift cluster.

```
$ ~/DataExport/bin/bootstrap -u master_username -P master_password
```

Here is an example of the output:



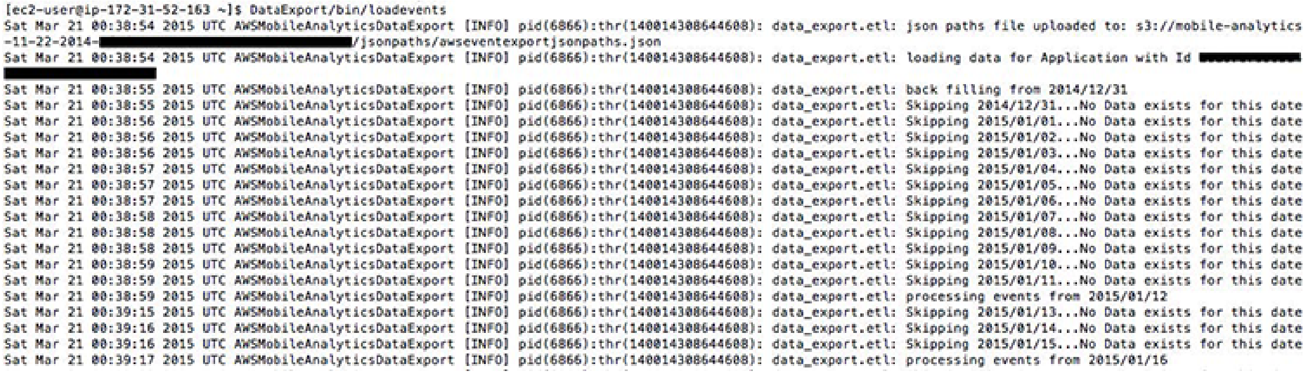
```
[ec2-user@ip-172-31-52-163 ~]$ DataExport/bin/bootstrap -u [redacted] -P [redacted]
Fri Mar 20 22:27:00 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6656):thr(139727596353280): data_export.redshift: --creating the AWSMA schema
Fri Mar 20 22:27:00 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6656):thr(139727596353280): data_export.redshift: --creating the etl_log table
Fri Mar 20 22:27:00 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6656):thr(139727596353280): data_export.redshift: --log updates
Fri Mar 20 22:27:00 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6656):thr(139727596353280): data_export.redshift: --creating the eventLoaders group
Fri Mar 20 22:27:00 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6656):thr(139727596353280): data_export.redshift: --creating the eventReaders group
Fri Mar 20 22:27:00 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6656):thr(139727596353280): data_export.redshift: --setting up AWSMA schema privileges
Fri Mar 20 22:27:00 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6656):thr(139727596353280): data_export.redshift: --setting up AWSMA schema privileges
Fri Mar 20 22:27:00 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6656):thr(139727596353280): data_export.redshift: --creating the etl_user user
Fri Mar 20 22:27:00 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6656):thr(139727596353280): data_export.redshift: --creating the eventReader user
Fri Mar 20 22:27:00 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6656):thr(139727596353280): data_export.redshift: --set the log table owner to etl_user
Fri Mar 20 22:27:00 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6656):thr(139727596353280): data_export.redshift: --setting the owner to etl_user - needed for vacuum & truncate operations
Fri Mar 20 22:27:00 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6656):thr(139727596353280): data_export.redshift: --creating the event fact table
Fri Mar 20 22:27:00 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6656):thr(139727596353280): data_export.redshift: --set the owner to etl_user - needed for vacuum & truncate operations
Fri Mar 20 22:27:00 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6656):thr(139727596353280): data_export.redshift: --creating main event view AWSMA.v_event
Fri Mar 20 22:27:00 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6656):thr(139727596353280): data_export.redshift: --setup group privileges
Fri Mar 20 22:27:00 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6656):thr(139727596353280): data_export.redshift: --updating log
[ec2-user@ip-172-31-52-163 ~]$ ▌
```

# Part 7: Loading Events from Amazon S3 to Amazon Redshift

To manually load events into the Amazon Redshift cluster from your Amazon S3 bucket, run the `loadevents` command at the command line.

```
$ DataExport/bin/loadevents
```

Watch the logs for errors. Here is an example of the output:



```
[ec2-user@ip-172-31-52-163 ~]$ DataExport/bin/loadevents
Sat Mar 21 00:38:54 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: json paths file uploaded to: s3://mobile-analytics
-11-22-2014-[redacted]/jsonpaths/awseventexportjsonpaths.json
Sat Mar 21 00:38:54 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: loading data for Application with Id [redacted]

Sat Mar 21 00:38:55 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: back filling from 2014/12/31
Sat Mar 21 00:38:55 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: Skipping 2014/12/31...No Data exists for this date
Sat Mar 21 00:38:56 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: Skipping 2015/01/01...No Data exists for this date
Sat Mar 21 00:38:56 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: Skipping 2015/01/02...No Data exists for this date
Sat Mar 21 00:38:56 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: Skipping 2015/01/03...No Data exists for this date
Sat Mar 21 00:38:57 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: Skipping 2015/01/04...No Data exists for this date
Sat Mar 21 00:38:57 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: Skipping 2015/01/05...No Data exists for this date
Sat Mar 21 00:38:57 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: Skipping 2015/01/06...No Data exists for this date
Sat Mar 21 00:38:58 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: Skipping 2015/01/07...No Data exists for this date
Sat Mar 21 00:38:58 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: Skipping 2015/01/08...No Data exists for this date
Sat Mar 21 00:38:58 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: Skipping 2015/01/09...No Data exists for this date
Sat Mar 21 00:38:59 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: Skipping 2015/01/10...No Data exists for this date
Sat Mar 21 00:38:59 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: Skipping 2015/01/11...No Data exists for this date
Sat Mar 21 00:38:59 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: processing events from 2015/01/12
Sat Mar 21 00:39:15 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: Skipping 2015/01/13...No Data exists for this date
Sat Mar 21 00:39:16 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: Skipping 2015/01/14...No Data exists for this date
Sat Mar 21 00:39:16 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: Skipping 2015/01/15...No Data exists for this date
Sat Mar 21 00:39:17 2015 UTC AWSMobileAnalyticsDataExport [INFO] pid(6866):thr(140014308644608): data_export.etl: processing events from 2015/01/16
```

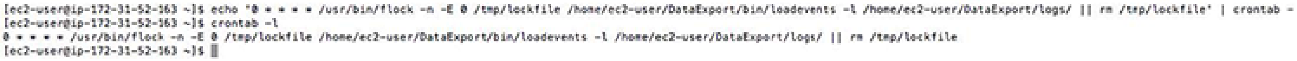
Once all of the events have successfully loaded, set up a crontab to execute the `loadevents` command every hour and write to the logs directory. Enter this at the command line:

```
$ echo '0 * * * * /usr/bin/flock -n -E 0 /tmp/lockfile /home/ec2-user/
DataExport/bin/loadevents -l /home/ec2-user/DataExport/logs/ || rm /tmp/
lockfile' | crontab -
```

Verify that the `loadevents` command runs every hour with the following command:

```
$ crontab –l
```

Here is an example of the output:



```
[ec2-user@ip-172-31-52-163 ~]$ echo '0 * * * * /usr/bin/flock -n -E 0 /tmp/lockfile /home/ec2-user/DataExport/bin/loadevents -l /home/ec2-user/DataExport/logs/ || rm /tmp/lockfile' | crontab -
[ec2-user@ip-172-31-52-163 ~]$ crontab -l
0 * * * * /usr/bin/flock -n -E 0 /tmp/lockfile /home/ec2-user/DataExport/bin/loadevents -l /home/ec2-user/DataExport/logs/ || rm /tmp/lockfile
[ec2-user@ip-172-31-52-163 ~]$ 
```

# Part 8: Configure CloudWatch

Configure Amazon CloudWatch to monitor the event loader command.

## Create an Agent Configuration File

The following example shows how to create an agent configuration file on the Amazon EC2 instance using Nano:

```
$ nano /tmp/cwlogs/application.conf
```

Add the following to the configuration file:

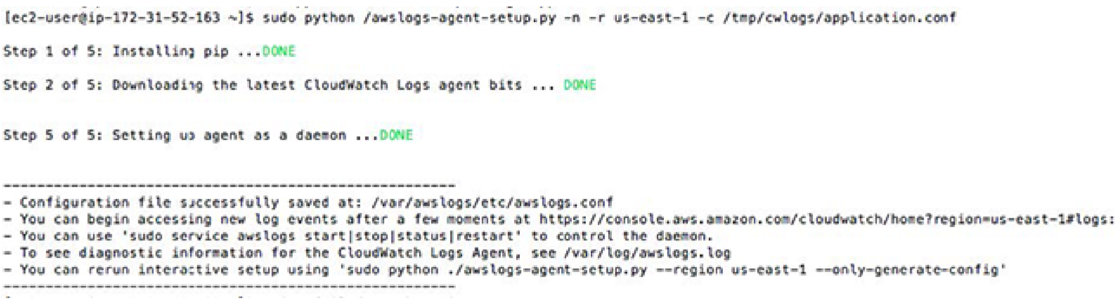
```
[general]
state_file = /var/awslogs/agent-state

[application.log]
file = /home/ec2-user/DataExport/logs/application.log*
log_group_name = MobileAnalytics.application.log
log_stream_name = {instance_id}/application.log
datetime_format = %a %b %d %H:%M:%S %Y
```

## Start the CloudWatch Agent

Start the CloudWatch agent with this command:

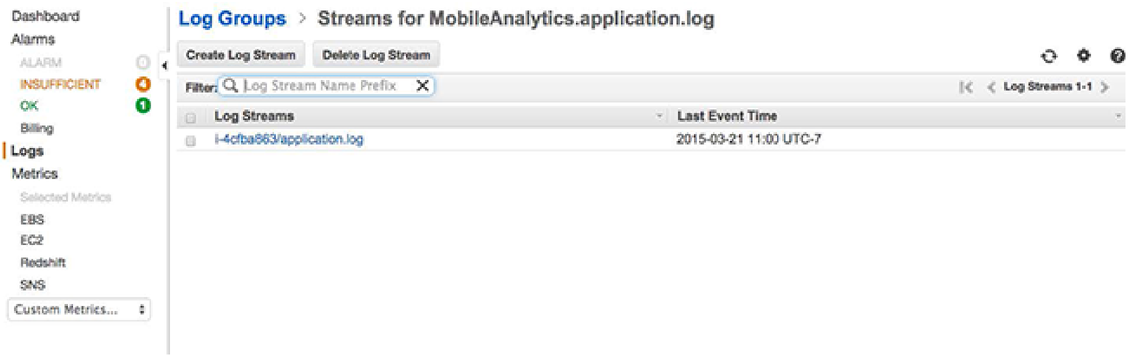
```
$ sudo python /awslogs-agent-setup.py -n -r us-east-1 -c /tmp/cwlogs/
application.conf
```

Here is an example of the output:



```
[ec2-user@ip-172-31-52-163 ~]$ sudo python /awslogs-agent-setup.py -n -r us-east-1 -c /tmp/cwlogs/application.conf

Step 1 of 5: Installing pip ...DONE

Step 2 of 5: Downloading the latest CloudWatch Logs agent bits ... DONE


Step 5 of 5: Setting up agent as a daemon ...DONE


--------------------------------------------------------
- Configuration file successfully saved at: /var/awslogs/etc/awslogs.conf
- You can begin accessing new log events after a few moments at https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#logs:
- You can use 'sudo service awslogs start|stop|status|restart' to control the daemon.
- To see diagnostic information for the CloudWatch Logs Agent, see /var/log/awslogs.log
- You can rerun interactive setup using 'sudo python ./awslogs-agent-setup.py --region us-east-1 --only-generate-config'
--------------------------------------------------------
```

Logs appear in the CloudWatch console after each run of the `loadevents` scripts.

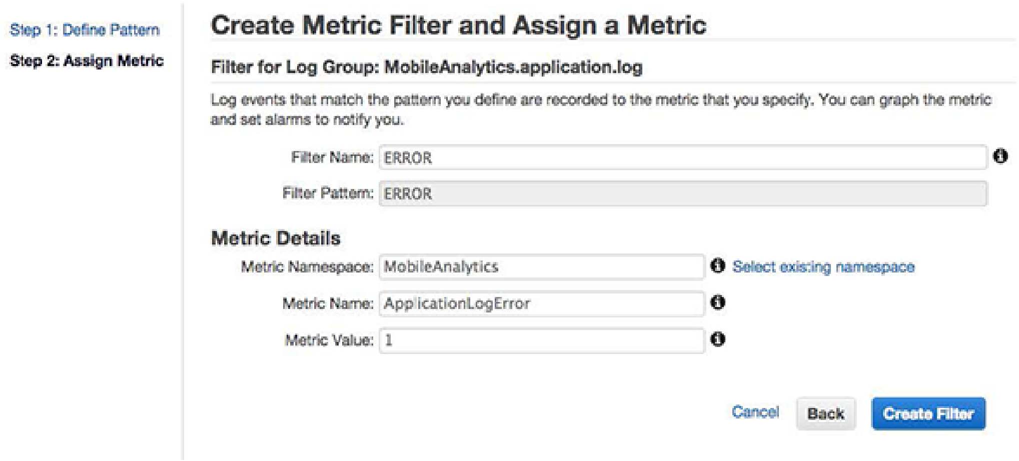


| Log Streams | Last Event Time |
|---|---|
| i-4cfba863/application.log | 2015-03-21 11:00 UTC-7 |

## Create a Metric Filter to Detect Errors

Next, you must create a CloudWatch metric filter to catch errors.

1. Navigate to the CloudWatch console and choose **Logs**.
2. Choose the MobileAnalytics.application.log.
3. Choose **Create Metric Filter**.
4. Choose ERROR as the filter pattern.
5. Type a name for the **Metric Namespace**; for example, **`MobileAnalytics`**.
6. Type the name of the **Metric**; for example, **`ApplicationLogError`**.
7. For **Metric Value**, type **`1`**.
8. Choose **Create Filter**.



Create Metric Filter and Assign a Metric

Filter for Log Group: MobileAnalytics.application.log

Log events that match the pattern you define are recorded to the metric that you specify. You can graph the metric and set alarms to notify you.

Filter Name: ERROR

Filter Pattern: ERROR

Metric Details

Metric Namespace: MobileAnalytics

Metric Name: ApplicationLogError

Metric Value: 1

# Exported Event JSON

The following JSON shows the data written for a hypothetical event that contains all elements and attributes. Not all elements are valid for all event types.

**Example**

```
{
  "event_type": "hypothetical_event",
  "event_timestamp": 1415390144920,
  "arrival_timestamp": 1415390151679,
  "event_version": "3.0",
  "application": {
    "app_id": “a376fb15e8d9460fab172014a1236e25",
    "package_name": "com.amazon.exampleapp",
    "version_name": "1.0",
    "version_code": "1",
    "title": "Example App",
    "cognito_identity_pool_id": "us-east-1:156b8012-aaaa-4da9-9c44-
a39198aebf7b",
    "sdk": {
      "name": "aws-sdk-android",
      "version": "2.1.1"
    }
  },
  "client": {
    "client_id": "47b63bb3-aaaa-bbbb-9ab7-2df4f1bccc82",
    "cognito_id": "us-east-1:4cbe503d-dd92-4224-8193-f3ba95cbec8f"
  },
  "device": {
    "model": "KFTT",
    "make": "Amazon",
    "platform": {
      "name": "ANDROID",
      "version": "4.0.3"
    },
    "locale": {
      "code": "en_US",
      "language": "en",
      "country": "US"
    }
  },
  "session": {
    "session_id": "f1bccc82-20141107-193640206",
    "start_timestamp": 1415389000206,
    "stop_timestamp": 1415389000206,
  },
  "monetization": {
    "transaction": {
      "transaction_id": "ABCDABCD-ABCD-ABCD-ABCD-ABCDABCDABCD",
      "store": "Amazon",
      "item_id": "com.yourcompany.exampleitem",
      "quantity": 1,
      "price": {
        "reported_price": "$0.99",
        "amount": 0.99,
        "currency": {
          "code": "USD",
          "symbol": "$"
        }
      }
    }
  },
  "attributes": {
    "levelName": "Level 5",
    "playerClass": "Wizard"
  },
  "metrics": {
    "score": 100400,
    "timeToCompleteMinutes": 3.5
  }
}
```

The following table describes each element.

### JSON Event Schema Data

| Attribute | Description | |
|---|---|---|
| amount | The numerical price value. Used only in a monetization event. | |
| app_id | The app ID that was used to identify the app. This ID was assigned on the App Management page of the console. | |
| arrival_timestamp | The time when the event was received by Amazon Mobile Analytics servers, expressed as an epoch time with milliseconds. | |
| client_id | Used to correlate events from the same app instance (app on a specific device). | |
| code | The locale code. | |
| code | Currency code for the purchase. Used only in a monetization event. | |
| cognito_id | If Amazon Cognito was used, this is the Amazon Cognito client ID. If not using Amazon Cognito, this attribute will not exist. | |
| cognito_identity-pool_id | If Amazon Cognito was used, this is the Amazon Cognito identity pool ID. If not using Amazon Cognito, this attribute will not exist. | |
| country | The country part of the locale. | |
| event_timestamp | The time the event occurred, expressed as an epoch time with milliseconds. | |
| event_type | The name that was specified when the event was created. Events such as _session.start and _session.stop are generated by the system. | |
| event_version | Version of the schema of the event. | |

| Attribute | Description | |
|---|---|---|
| item_id | An identifier for the item purchased. Used only in a monetization event. | |
| language | The language part of the locale. | |
| make | The make of the device. | |
| model | The model of the device. | |
| name | The name of the AWS Mobile SDK used by the app. | |
| name | If the Mobile SDK was used, this is the Android or iPhone OS () or the value you specified when using the PutEvents API. | |
| package_name | The package name. | |
| quantity | The quantity of items purchased. Used only in a monetization event. | |
| reported_price | A string representation of the price reported by the purchasing library. Used only in a monetization event. | |
| session_id | An ID for all events that occurred in the same session. | |
| start_timestamp | Time when the session began, expressed as an epoch timestamp with milliseconds. Used only on _session.start events. | |
| stop_timestamp | Time when the session ended, expressed as an epoch timestamp with milliseconds. Used only on _session.stop events. | |
| store | The name of the store where the transaction took place. Used only in a monetization event. | |
| symbol | Currency symbol for the purchase. Used only in a monetization event. | |
| title | The title of the app. | |
| transaction_id | An identifier for the transaction. Used only in a monetization event. | |

| Attribute | Description | |
|---|---|---|
| version | The platform version. | |
| version | The version of the Mobile SDK used by the app. | |
| version_code | The version code. | |
| version_name | The version name. | |

# Amazon Redshift Schema for Event Data

A new schema called AWSMA is created to contain the tables and views to access the data. To access events, use the AWSMA.v_event view. You can sign in to your Amazon Redshift cluster using the `eventreader` user and the password configured in the auto export feature. This user will have full select access to the AWSMA.v_event view.

The following table describes schema for the AWSMA.v_event view. Custom attributes and custom metric columns appear at the end.

**JSON Amazon Redshift Schema for Event Data**

| Column Name | Data Type | Description |
|---|---|---|
| event_type | varchar(256) | The name that was specified when the event was recorded in the SDK or the name of a system event that was recorded by the SDK (for example, _session.start, _session.stop). |
| event_timestamp | timestamp | The UTC timestamp (of the device) when the event occurred. |
| arrival_timestamp | timestamp | The UTC timestamp (of the server) when the event was received by the AWS server. |
| event_version | char(12) | The version of the event structure (typically 3.0). |
| application_app_id | varchar(64) | The ID used to identify the app. |
| application_package_name | varchar(256) | The package name of the app. |
| application_version_name | varchar(256) | The version name associated with the app. |
| application_version_code | varchar(256) | The version code associated with the app. |
| application_title | varchar(256) | The title of the app. |
| application_cognito_identity_pool_id | varchar(64) | This is populated if you are using Amazon Cognito authentication. It is the Amazon Cognito identity pool ID. |

| Column Name | Data Type | Description |
|---|---|---|
| application_sdk_name | varchar(256) | The AWS SDK that sent the event. |
| application_sdk_version | varchar(256) | The version of the AWS SDK that sent the event. |
| client_id | varchar(64) | An ID used to identify other events sent from the same app instance. If you used the AWS Mobile SDK to record events, all events from the same app on a device will have the same client_id. This is a GUID string. You can use this value to sample events from a random set of users. |
| client_cognito_id | varchar(64) | The Amazon Cognito client ID, this column is populated if you are using Amazon Cognito authentication. This is useful to correlate different users if you are using non-anonymous users. |
| device_model | varchar(256) | The device model. |
| device_make | varchar(256) | The device make. |
| device_platform_name | varchar(256) | The device platform (for example, iPhoneOS or Android). |
| device_platform_version | varchar(256) | The device platform version. |
| device_locale_code | varchar(256) | The locale of the device (for example, en_US). |
| device_locale_language | varchar(64) | The language part of the locale (for example, "en"). |
| device_locale_country | varchar(64) | The country part of the locale (for example, "US"). |
| session_id | varchar(64) | A GUID string used to identify events that came from the same app and device in the same session. |
| session_start_timestamp | timestamp | The timestamp when the session started. |
| session_end_timestamp | timestamp | The timestamp when the session stopped. Populated for _session.stop events only. |
| monetization_transaction_id | varchar(64) | An identifier for the transaction. |
| monetization_transaction_store | varchar(64) | The name of the app store used for the transaction. |

| Column Name | Data Type | Description |
|---|---|---|
| monetization_transaction_item_id | varchar(64) | An identifier for the item purchased. |
| monetization_transaction_quantity | FLOAT8 | The quantity of items purchased. |
| monetization_transaction_price_reported | varchar(64) | A string representation of the price reported by the purchasing library. This is different for each store. |
| monetization_transaction_price_amount | FLOAT8 | The numerical price value. |
| monetization_transaction_price_currency_code | varchar(16) | The currency code for the purchase. |
| monetization_transaction_price_currency_symbol | varchar(32) | The currency symbol for the purchase. |
| a_levelName | varchar(4000) | All custom attributes start with "a_". |
| a_playerClass | varchar(4000) | All custom attributes start with "a_". |
| m_score | FLOAT8 | All custom metrics start with "m_". |
| m_timeToCompleteMinutes | FLOAT8 | All custom metrics start with "m_". |

# Example Amazon Redshift Queries

The “users” columns in the following queries count distinct client_cognito_id values only when users are authenticated with Amazon Cognito. For unauthenticated use cases, use the client_id for app-device counts.

Top item purchases, by store (last 30 days):

```
SELECT
  application_app_id AS"app id",
  monetization_transaction_item_id AS"item id",
  monetization_transaction_store AS"store",
  COUNT(DISTINCTclient_id) AS"devices",
  COUNT(DISTINCTclient_cognito_id) AS"users",
  SUM(monetization_transaction_quantity) AS"quantity",
  SUM(monetization_transaction_price_amount) "amount (Apple only)",
  monetization_transaction_price_currency_code AS"currency (Apple only)"
  FROM
  AWSMA.v_event
  WHERE
  event_type = '_monetization.purchase'AND
  event_timestamp BETWEENgetdate() - 30 ANDgetdate() + 1
  GROUPBY
  "app id",
  "item id",
  "currency (Apple only)",
  "store"
  ORDERBY
  "app id"ASC,
  "item id"ASC,
  "quantity"DESC,
  "store",
  "devices"DESC
  ;
```

30 day active users and devices, by device language code:

```
SELECT
   application_app_id AS"app id",
   COUNT(DISTINCTclient_id) AS"devices",
   COUNT(DISTINCTclient_cognito_id) AS"users",
   device_locale_language AS"language"
   FROM
   AWSMA.v_event
   WHERE
   event_type = '_session.start'AND
   event_timestamp BETWEENgetdate() - 30 ANDgetdate() + 1
   GROUPBY
   "app id",
   "language"
   ORDERBY
   "app id"ASC,
   "devices"DESC,
   "language"
   ;
```

30 day active users and devices, by app version:

```
SELECT
  application_app_id AS"app id",
  device_platform_name AS"platform",
  application_version_name AS"version name",
  application_version_code AS"version code",
  COUNT(DISTINCTclient_id) AS"devices",
  COUNT(DISTINCTclient_cognito_id) AS"users"
  FROM
  AWSMA.v_event
  WHERE
  event_type = '_session.start'AND
  event_timestamp BETWEENgetdate() - 30 ANDgetdate() + 1
  GROUPBY
  "app id",
  "platform",
  "version name",
  "version code"
  ORDERBY
  "app id"ASC,
  "platform"ASC,
  "devices"DESC,
  "version name"DESC,
  "version code"DESC
  ;
```

# Explanation of Infrastructure

The Mobile Analytics and Amazon Redshift combination includes the following components:

- An Amazon Redshift cluster that holds all auto export events. It can start small and scale up to petabytes of information. For more information, see Amazon Redshift.
- A VPC to isolate network resources to control access. There is no additional charge for using Amazon Virtual Private Cloud (Amazon VPC), aside from the standard Amazon EC2 usage charges. For more information, see Amazon VPC.
- A t2.micro EC2 instance to facilitate the transfer of data from your S3 subscription bucket into Amazon Redshift. For more information, see Amazon EC2.
- Amazon CloudWatch Logs and metrics for information about the health of the data transfer to Amazon Redshift.
- Python scripts that run on your EC2 instance and allow you to interact with the Amazon Redshift cluster.

## Amazon Redshift Cluster

Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse solution that makes it simple and cost-effective to efficiently analyze all of your data using your existing business intelligence tools. For more information, see Amazon Redshift.

## Virtual Private Cloud

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. For more information, see Amazon Virtual Private Cloud.

## EC2 Instance

When the EC2 instance is provisioned, it downloads the Mobile Analytics Python scripts. After the EC2 instance starts, these scripts connect to the Amazon Redshift cluster and set up the users and tables required to process Mobile Analytics events. The scripts are invoked every hour to load all new event data into your Amazon Redshift cluster.

You can connect to the console using SSH with the EC2 keyname specified in the advanced settings: **ssh -i your-keyname.pem ec2-user@public-dns-of-instance**

> **Note**
> Your pem file must be set to read-only permissions. Run the following command to ensure the file will be accepted: **chmod 400 your-keyname.pem**.

For more information about EC2 instances, see Amazon EC2.

## CloudWatch Metrics and Logs

The EC2 instance records CloudWatch metrics and logs for debugging and monitoring of the EC2 instance. For more information, see Amazon CloudWatch Logs. To view metrics about the execution of the EC2 loadevents operation:

1. Go to the CloudWatch console.
2. In the left navigation pane, choose **Custom Metrics**.

3. Choose the **LoadEventsDuration** metric.

To view logs about the invocation of the EC2 instance:

1. Go to the CloudWatch console.
2. Choose **Logs**.
3. Choose the **LogGroup** for the EC2 instance.
4. Choose the application log stream.

# Troubleshooting

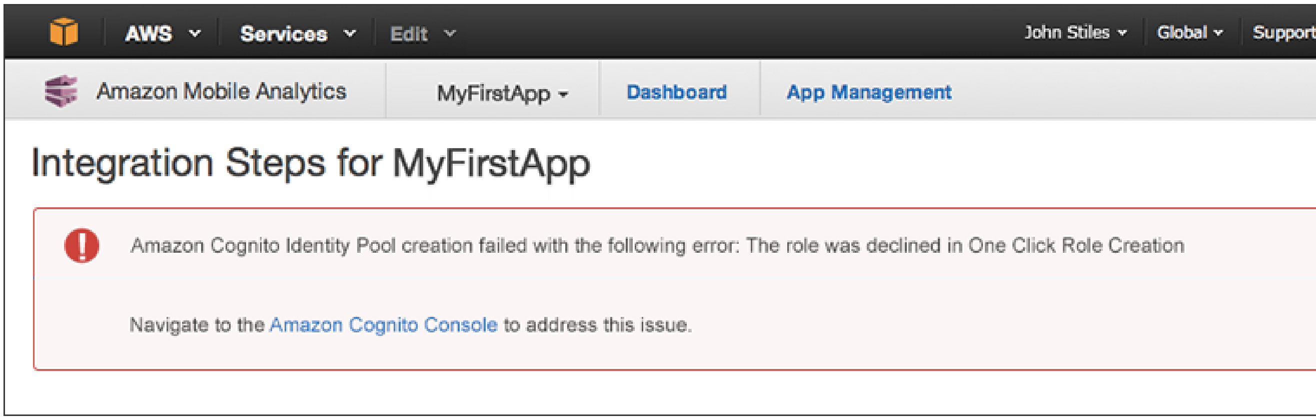
If you encounter issues or difficulties when working with Mobile Analytics, consult the topics in this section.

> **Using the Mobile SDK**
> When using the Mobile SDK to incorporate Mobile Analytics into your app, you may find it useful when troubleshooting to enable logging through the SDK. For more information, see Enabling SDK Logging (p. 20).

Topics

## If Default IAM Role Creation Fails

The following error message is displayed if Mobile Analytics is unable to create a default IAM role and Cognito identity pool when you add an app:



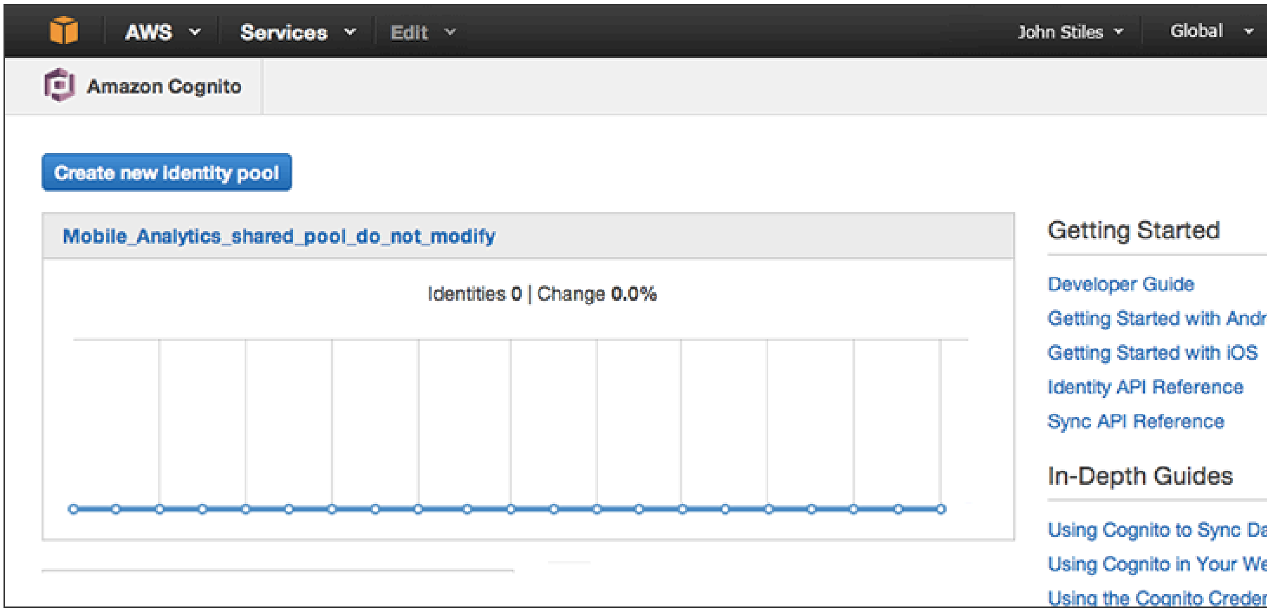
Integration Steps for MyFirstApp

Amazon Cognito Identity Pool creation failed with the following error: The role was declined in One Click Role Creation

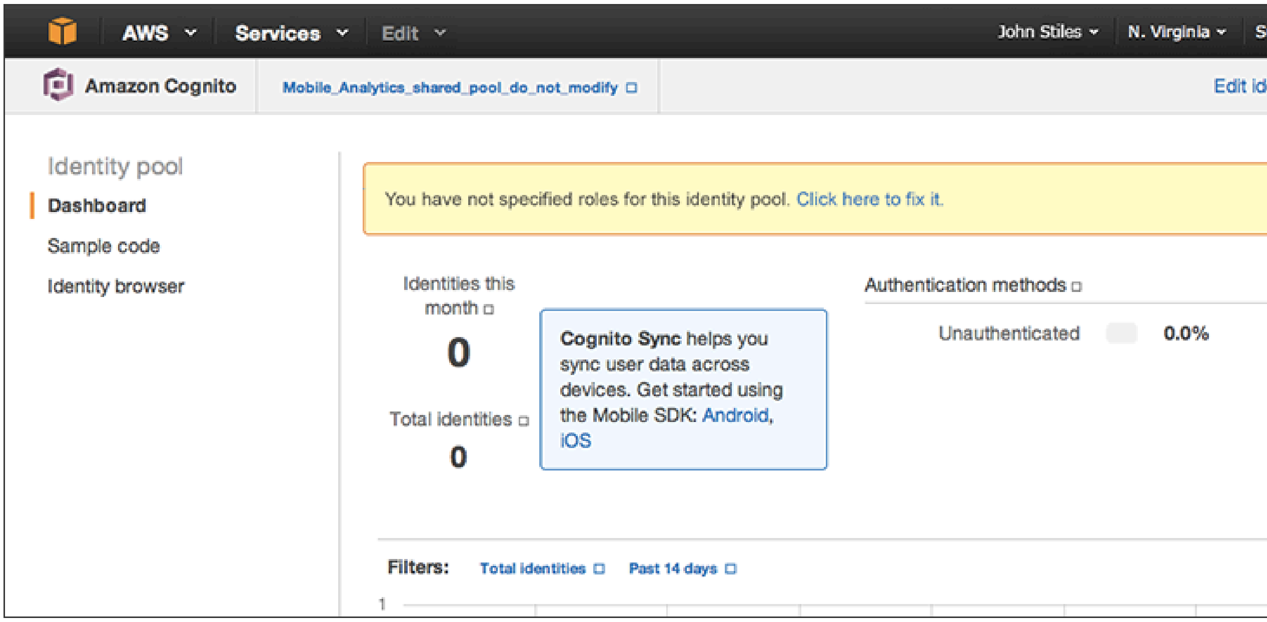
Navigate to the Amazon Cognito Console to address this issue.

### To correct a failed attempt to create a default IAM role and Cognito identity pool

1. Open the Amazon Cognito console.
2. Click **Mobile_Analytics_shared_pool_do_not_modify**.



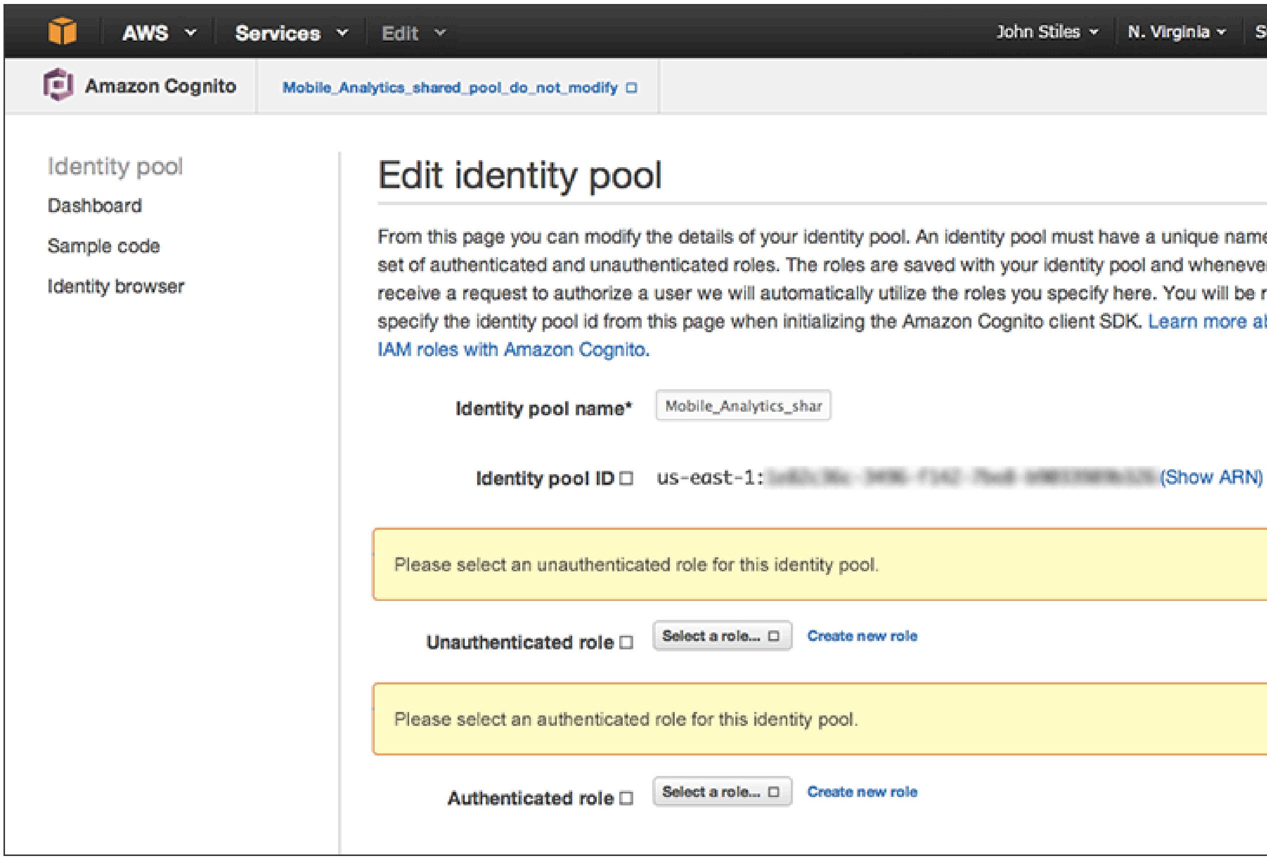
3. Choose **Click here to fix it.**



You have not specified roles for this identity pool. Click here to fix it.

Cognito Sync helps you sync user data across devices. Get started using the Mobile SDK: Android, iOS

4. For **Unauthenticated role** and **Authenticated role**, choose an existing role or create a new role.



Edit identity pool

From this page you can modify the details of your identity pool. An identity pool must have a unique name set of authenticated and unauthenticated roles. The roles are saved with your identity pool and wheneve receive a request to authorize a user we will automatically utilize the roles you specify here. You will be r specify the identity pool id from this page when initializing the Amazon Cognito client SDK. Learn more al IAM roles with Amazon Cognito.

Identity pool name* Mobile_Analytics_shar

Identity pool ID □ us-east-1:[redacted] (Show ARN)

Please select an unauthenticated role for this identity pool.

Unauthenticated role □ Select a role... □ Create new role

Please select an authenticated role for this identity pool.

Authenticated role □ Select a role... □ Create new role

5. Return to the Mobile Analytics console and the integration steps for your app.

# If Auto Export to S3 Fails

If you set up Auto Export to Amazon S3 but are not getting exported data in the S3 bucket, it could be the S3 IAM access role being used to export the data does not grant access to the correct Amazon S3 bucket. For example, if you initially set up Auto Export to Amazon Redshift and Amazon S3 but later disable that choice to set up Auto Export to S3 alone, make sure when choosing the S3 bucket that you either:

- Select the S3 bucket created originally when you set up the original Auto Export, which uses the S3 IAM access role created at the time, or
- Create a new S3 bucket, making sure to create a new S3 IAM access role for it.

# If New Attributes or Metrics Don't Appear in Auto Export to Amazon Redshift

Event data exported to Amazon S3 has all attributes and metrics that were sent to the service. However, Amazon Redshift only contains attributes or metrics that were specified when the Auto Export was created. If you expect to find particular attributes or metrics in the Auto Export to Amazon Redshift that aren't there, those attributes or metrics were likely not included in the original setup of the Auto Export.

You can resolve this circumstance in either of these ways:

- Recreate Auto Export to a new Amazon Redshift cluster
- Add or remove attributes that are loaded to your existing Amazon Redshift cluster via AWS CloudFormation

For more information, see Adding New Metrics and Attributes to an Auto Export to Amazon Redshift (p. 93).

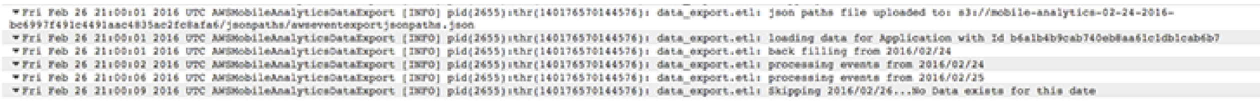
# If Data Does Not Appear in Amazon Redshift Cluster

First check to see if your data is being exported to the Amazon S3 bucket set up by Mobile Analytics. Successful export of data to your Amazon S3 bucket is a prerequisite for proper auto-export to Amazon Redshift. If your data is not appearing in Amazon S3, see If Auto Export to S3 Fails (p. 120).

If you have just set up the auto-export to Amazon Redshift, make sure you wait at least an hour before you query Amazon Redshift. By default, the Amazon EC2 instance loads events from Amazon S3 into Amazon Redshift every hour. To load events sooner than one hour, you can log into the Amazon EC2 instance and adjust the crontab settings. Amazon Redshift export can be attempted at any rate by the Amazon EC2 instance, but won't be picked up for auto-export to Amazon Redshift until the data is exported to the Amazon S3 bucket.

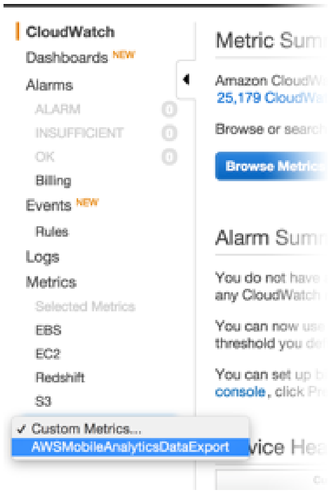
If you enabled CloudWatch metrics and logs when setting up the auto-export, follow these steps to investigate:

1. Navigate to the CloudWatch console.
2. Look at the log group with **MobileAnalyticsAutoExportToRedshift** as the prefix. If you have multiple of these log groups, look for the one with a uuid at the end matching the uuid in your latest AWS CloudFormation stack name.
3. In the selected log group, look at the latest log stream. This shows the logs produced by the Amazon EC2 instance when loading events into Amazon Redshift. Because the Amazon EC2 instance loads events every hour by default, expect to see logs on the hour boundaries. Review these logs for any errors.



```
▼Fri Feb 26 21:00:01 2016 UTC AWSMobileAnalyticsDataExport [INFO] pid(2655):thr(140176570144576): data_export.etl: json paths file uploaded to: s3://mobile-analytics-02-24-2016-
bc6997f491c4491aac4835ac2fc8afa6/jsonpaths/awseventexportjsonpaths.json
▼Fri Feb 26 21:00:01 2016 UTC AWSMobileAnalyticsDataExport [INFO] pid(2655):thr(140176570144576): data_export.etl: loading data for Application with Id b6a1b4b9cab740eb8aa61c1db1cab6b7
▼Fri Feb 26 21:00:01 2016 UTC AWSMobileAnalyticsDataExport [INFO] pid(2655):thr(140176570144576): data_export.etl: back filling from 2016/02/24
▼Fri Feb 26 21:00:02 2016 UTC AWSMobileAnalyticsDataExport [INFO] pid(2655):thr(140176570144576): data_export.etl: processing events from 2016/02/24
▼Fri Feb 26 21:00:06 2016 UTC AWSMobileAnalyticsDataExport [INFO] pid(2655):thr(140176570144576): data_export.etl: processing events from 2016/02/25
▼Fri Feb 26 21:00:09 2016 UTC AWSMobileAnalyticsDataExport [INFO] pid(2655):thr(140176570144576): data_export.etl: Skipping 2016/02/26...No Data exists for this date
```

4. Look at the CloudWatch metrics in the **AWSMobileAnalyticsDataExport** namespace. Look specifically at the **LoadEventsDuration** metric to see any trending errors or gaps in event loading to Amazon Redshift.



5. If you need assistance after reviewing these logs, you can send snippets or screenshots of these logs to Support for additional debugging help.

If you did not enable CloudWatch metrics and logs when setting up the auto-export, it will be harder to determine what the Amazon EC2 instance might experience when loading events. You will have to rely on the information found in the Amazon Redshift load logs. Connect to your Amazon Redshift cluster and query the `stl_load_errors` table as follows:

```
select * from stl_load_errors;
```

For more information, see System Tables for Troubleshooting Data Loads.

# Limits

## Aggregated Custom Events

The following is a list of limits for Mobile Analytics custom events that are aggregated into console reports or reportable values returned by direct query. Any additional values submitted beyond these limits are not aggregated, however they are still included in raw event data exported to Amazon S3 and Amazon Redshift using the Auto Export feature.

| Capacity | Limit |
|---|---|
| Total number of custom event types per application. | 1,500 |
| Total number of unique attribute or metric names per custom event type. | 2,000 |
| Total number of unique attribute values per custom event type/attribute name combination. | 1,000 |

## Other Custom Events Limits

The following is a list of additional limits for a Mobile Analytics custom event.

| Capacity | Limit |
|---|---|
| Total number of attributes and metrics per event submitted. | 40 |
| Number of characters in the key of an attribute or metric. | 50 |
| Number of characters in the value of an attribute. | 200 |
| Number of apps per AWS account. Contact AWS support to request an increase in this limit. | 100 |

# Custom Dashboards

The following is a list of limits for a Mobile Analytics custom dashboard.

| Capacity | Limit |
|---|---|
| Total number of custom dashboards. | 25 |
| Maximum number of data series in a chart. | 10 |

# REST API Reference

This is the Amazon Mobile Analytics REST API reference. It contains examples of the following events, requests, and operations.

Topics

## Making HTTP Requests to Mobile Analytics

If you don't use the AWS Mobile SDK, you can perform Mobile Analytics operations over HTTP using the POST request method. The POST method requires you to specify the operation in the header of the request and provide the data for the operation in JSON format in the body of the request.

### HTTP Header Contents

Mobile Analytics requires the following information in the header of an HTTP request:

**Host**
The Mobile Analytics endpoint. This value must be `https://mobileanalytics.us-east-1.amazonaws.com`

**X-Amz-Date**
The date. Must be specified in ISO 8601 standard format, in UTC time. For example:

```
20130315T092054Z
```

**Authorization**
The set of authorization parameters that AWS uses to ensure the validity and authenticity of the request. For more information, see Signature Version 4 Signing Process.

**User Agent**
Information about the user agent originating the request.

**X-Amz-Client-Context**
Information about the client interacting with Mobile Analytics. Data in a client context describes the app and the environment in which it runs. For details about the contents of the client context, see PutEvents (p. 127).

**X-Amz-Security-Token**

If you sign your request using temporary security credentials, you must include the corresponding security token in your request by adding the `X-Amz-Security-Token` header.

For information on signing requests using temporary security credentials in your REST API requests, see Signing and Authenticating REST Requests.

**Content-Type**

Specifies JSON and the version. For example, `Content-Type: application/json`

**Content-Length**

The payload size in bytes.

## HTTP Header Example

The following is an example header for an HTTP request for Mobile Analytics.

```
POST /2014-06-05/events HTTP/1.1
Host: mobileanalytics.us-east-1.amazonaws.com
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<access_key>/20140709/us-
east-1/mobileanalytics/aws4_request, SignedHeaders=content-length;content-
type;host;user-agent;x-amz-client-context;x-amz-date;x-amz-security-token;x-
amz-target, Signature=<signature>
User-Agent: <User agent string>
x-amz-Client-Context: {"client":
{"client_id":"<client_id>","app_title":"<app_title>","app_version_name":"<app_version_name>
{},"env":
{"platform":"<platform>","model":"<model>","make":"<make>","platform_version":"<platform_ve
x-amz-security-token: <Security token>
Content-Type: application/json
Content-Length: <Payload size bytes>
Connection: Keep-Alive
```

# HTTP Body Content

The body of an HTTP request contains the data for the operation specified in the header of the HTTP request. The data must be formatted according to the JSON data schema for Mobile Analytics. For the PutEvents operation, the body content of the HTTP request consists of an array of one or more events.

## HTTP Body Example

The following is an example of the body for an HTTP request for Mobile Analytics.

```
{
  "events": [
    {
      "eventType": "<Event type>",
      "timestamp": "<ISO 8601 date>",
      "session": {
                  "id": "<Session id>",
                  "startTimestamp": "<ISO 8601 date>"
      },
      "attributes": {
                     "<Optional string name>": "<Optional string value>",
                     ...
                     "<Optional string name>": "<Optional string value>"
      },
      "metrics": {
                  "<Optional string name>": <Optional numeric value>,
                   ...
                  "<Optional string name>": <Optional numeric value>
      }
    },
    ...
  ]
}
```

# PutEvents

The PutEvents operation records one or more events. You can have up to 1,500 unique custom events per app, any combination of up to 40 attributes and metrics per custom event, and an infinite number of attribute or metrics values.

Topics

## Requests

### Client Context Header

#### Syntax

```
x-amz-Client-Context: {
                        "client": {
                                    "client_id":"<client_id>",
                                    "app_title":"<app_title>",
                                    "app_version_name":"<app_version_name>",
                                    "app_version_code":"<app_version_code>",
                                    "app_package_name":"<app_package_name>"
                                  },

                        "custom": {},

                        "env":{
                                "platform":"<platform>",
                                "model":"<model>",
                                "make":"<make>",
                                "platform_version":"<platform_version>",
                                "locale":"<locale>”
                              },

                        "services": {
                                      "mobile_analytics": {

 "app_id":"<mobile_analytics_app_id>"
                                                          }
                                    }
                       }
```

## Description

The operation takes the following request header.

**x-amz-client-context**
The request header.

**client**
Name-value pairs that describes the client application.

**client_id**
A unique identifier representing this installation instance of your app.

Type: String

Default: None

Required: Yes

**app_title**
The title of your app. For example, **`My App`**.

Type: String

Default: None

Required: Yes

**app_version_name**
The version of your app. For example, **`V2.0`**.

Type: String

Default: None

Required: No

**app_version_code**
The version code for your app. For example, **`3`**.

Type: String

Default: None

Required: No

**app_package_name**
The name of your package. For example, **`com.example.my_app`**.

Type: String

Default: None

Required: No

**custom**
User defined name-value pairs that describe this installation of the application.

Type: Map

Default: None

Required: No

**env**
Name-value pair that describes the device that runs the event.

**platform**
The operating system of the device. For example, **`iphoneos`**.

Type: String

Valid values: `iphoneos`, `android`, `windowsphone`, `blackberry`, `macos`, `windows`, `linux`

Default: None

Required: Yes

**model**

The model of the device. For example, **`Nexus`**.

Type: String

Default: None

Required: No

**make**

The manufacturer of the device. For example, **`Samsung`**.

Type: String

Default: None

Required: No

**platform_version**

The version of the operating system of the device. For example, **`4.0.4`**.

Type: String

Default: None

Required: No

**locale**

The locale of the device. For example, **`en_US`**.

Type: String

Default: None

Required: No

**services**

Name-value pair that contains service specific sections.

**mobile_analytics**

Name-value pair that describes service specific attributes.

**app_id**

The value obtained from the Mobile Analytics console to record data to.

Type: String

Default: None

Required: Yes

## Request Body

### Syntax

```
{
  "events": [
    {
      "eventType": "<Event type>",
      "timestamp": "<ISO 8601 date>",
      "version": "v2.0",
      "session": {
                  "id": "<Session id>",
                  "startTimestamp": "<ISO 8601 date>"
      },
      "attributes": {
                     "<Optional string name>": "<Optional string value>",
                     ...
                     "<Optional string name>": "<Optional string value>"
      },
      "metrics": {
                  "<Optional string name>": <Optional numeric value>,
                   ...
                  "<Optional string name>": <Optional numeric value>
      }
    },
    ...
  ]
}
```

## Description

This operation takes the following request content.

**Events**

An array of JSON objects representing a batch of unique event occurrences in your app. Each JSON object in the array consists of the following:

**eventType**

A name signifying an event that occurred in your app. This is used for grouping and aggregating like events together for reporting purposes.

Type: String

Default: None

Required: Yes

**timestamp**

The time the event occurred in ISO 8601 standard date time format. For example, **`2014-06-30T19:07:47.885Z`**

Type: String

Constraints: Must follow ISO 8601 format

Default: None

Required: Yes

**attributes**

A collection of key-value pairs that give additional context to the event. The key-value pairs are specified by the developer.

This collection can be empty or the attribute object can be omitted.

Type: JSON object of key-value pairs (String:String)

Constraints: Key can be up to 50 characters or less and the value can be up to 200 characters.

Default: None

Required: No

**metrics**

A collection of key-value pairs that gives additional measurable context to the event. This key has the following key-values pairs. The pairs specified by the developer.

This collection can be empty or the attribute object can be omitted.

Type: JSON object of key value pairs (String:Number)

Constraints: Key can be up to 50 characters.

Default: None

Required: No

**session**

Describes the session. Session information is required on events to be aggregated in console reports. Events submitted without session information are still exported to S3 or Redshift but are not aggregated for inclusion in reports. This key has the following key-value pairs.

**id**

A unique identifier for the session.

Type: String

Default: None

Required: Yes

**startTimestamp**

The time the event occurred in ISO 8601 standard date time format. For example, **`2014-06-30T19:07:47.885Z`**

Type: String

Constraints: Must follow ISO 8601 format

Default: None

Required: Yes

**version**

Describes the version. This value must always be **`v2.0`**.

Type: String

Constraints: Must always be **`v2.0`**.

Default: None

Required: Yes

## JSON Example

```
{"events":[
  {
    "metrics":{
```

```
                "Score":12345,
                "TimeInLevel":64
    },
    "session":{
                "id" : "<session id>",
                "startTimestamp" : "2014-06-30T19:07:47.885Z"
    },
    "attributes":{
                   "LevelName":"Level1",
                   "CharacterClass":"Warrior",
                   "Successful":"True"
    },
    "eventType":"LevelComplete",
    "version":"v2.0",
    "timestamp":"2014-06-30T19:07:47.885Z"
  }
]}
```

## Responses

### Syntax

```
HTTP/1.1 202
x-amzn-RequestId: <A request id>
Content-Type: application/json
```

### Response Headers

This operation has the following response codes.

**202 Accepted**
The request has been accepted for processing, however the events have not been fully processed.

**400 Bad Request**
The x-amz-client-context header is missing or invalid.

OR

The event payload is missing or invalid.

**403 Forbidden**
The request is not authorized to perform this action.

**413 Request Entity Too Large**
The payload is too large. The payload cannot exceed 1024 KB.

# Examples

For examples of typical tasks performed with the PutEvents operation, see Using the REST API (p. 32).

# Document History

The following table describes the important changes to the documentation since the last release of Mobile Analytics.

- **API version:** 2014-06-05
- **Latest documentation update:** April 6, 2016

| Change | Description | Date Changed |
|---|---|---|
| Getting a List of Apps Added | Using the REST API to get a list of your apps is now detailed. For more information, see Getting a List of Apps (p. 43). | April 6, 2016 |
| Service Limits Revised | Several limits affecting custom events have been revised. For more information, see Limits (p. 123). | March 22, 2016 |
| Auto Export Overview Added | The auto export of analytics data to Amazon S3 or Amazon Redshift is now described in an overview. For more information, see Auto Export Overview (p. 89). | March 22, 2016 |
| Enhanced Custom Dashboards | You can create custom dashboards to easily monitor analytics that are most useful for measuring the performance of an app. For more information, see Using Custom Dashboards (p. 77). | March 8, 2016 |
| Additional Auto Export Documentation | Additional topics were added to document how to auto export analytics to an existing Amazon Redshift cluster instead of one created by Mobile Analytics. For more | March 8, 2016 |

| Change | Description | Date Changed |
|---|---|---|
| | information, see Exporting to an Existing Amazon Redshift Cluster (p. 96). | |
| Querying App Analytics | Using the REST API, you can run queries to obtain data details of the analytics for an app. For more information, see Querying Analytics Data (p. 42). | December 30, 2015 |
| Additional Troubleshooting Tips | Information has been added on how to activate SDK logging to help with troubleshooting. For more information, see Enabling SDK Logging (p. 20). Details provided on various options to add new metrics and attributes to an Auto Export to Amazon Redshift. For more information, see Adding New Metrics and Attributes to an Auto Export to Amazon Redshift (p. 93). | October 27, 2015 |
| Auto Export to Amazon S3 Troubleshooting Added | Troubleshooting information has been added to help in cases when Auto Export to Amazon S3 does not produce data in the Amazon S3 bucket. For more information, see If Auto Export to S3 Fails (p. 120). | September 15, 2015 |
| Documentation Improvements | The User Guide has been expanded and reorganized in parts. Examples of using the REST API have been moved from the reference to a series of topics that describe them in more detail. | September 4, 2015 |
| Xamarin Support Added | The AWS Mobile SDK adds support for Xamarin, including Mobile Analytics. The Mobile Analytics User Guide now includes links to the Xamarin SDK Developer Guide regarding Mobile Analytics. Displayed integration steps now include sample code and steps for integration with a Xamarin app. | July 29, 2015 |

| Change | Description | Date Changed |
|---|---|---|
| Streamlined New App Workflow | The process for creating a new app now includes a one-click wizard for generating the necessary IAM role and Cognito identity pool for users with no existing identity pools to use. Users with existing identity pools have the ability to create a new identity pool or use one of their existing identity pools with new apps. For more information, see Adding an App to Mobile Analytics. (p. 6) | July 14, 2015 |
| New Supported Platforms | The Mobile Analytics console now displays reports on an expanded set of platforms. You can now filter the events that are compiled into the console reports to display the following platform choices: All Platforms, iOS, Android, Fire OS, Windows Phone, Blackberry, Windows, Mac, and Linux. Screen shots have been updated. | June 30, 2015 |
| Auto Export | Added a topic on how to auto export app data to an Amazon S3 bucket. For more information, see Amazon Mobile Analytics Auto Export. | December 17, 2014 |