Custom Coded Workflow Actions - Workshop

Connect to a MySQL Database

Jack Coldrick, Senior Solutions Engineer @ HubSpot

Overview

- What is a custom coded workflow action?
- What will we be doing in this workshop?
- Setting up a developer account
- Practical:
  - Step 1: Setup a developer account
  - Step 2: Create the Database Instance
  - Step 3: Configure the Database Instance
  - Step 4: Create the Workflow
  - Step 5: Test the Workflow
- Useful Resources

60

minutes

What will we be doing in this workshop?

MySQL is a popular open source relational database. In this workshop we will be creating a database using Amazon Web Services (AWS) and building a HubSpot custom coded action that will check to see if a contact exists in the database when they're created in the CRM. If they do exist, no action is taken. If they don't exist we'll add a new row to the database.

View Video Walkthrough

What is a custom coded workflow action?

Custom coded workflow actions are a feature included with HubSpot's Operations Hub Professional and allow you to write code in Javascript within a Node JS Runtime environment to solve for specific use cases and processes relevant to your business.

Learn More | Use Cases | Blog Post

Step 1:
Setup a Developer Account

This is an important step, setting up a developer account. You can do this by clicking on the link below. Once you've set this up you will have a special HubSpot portal where you can create up to 10 test portals that will allow you to try and test functionality in a controlled environment. If you already have a developer account feel free to skip this section!

Create a developer account

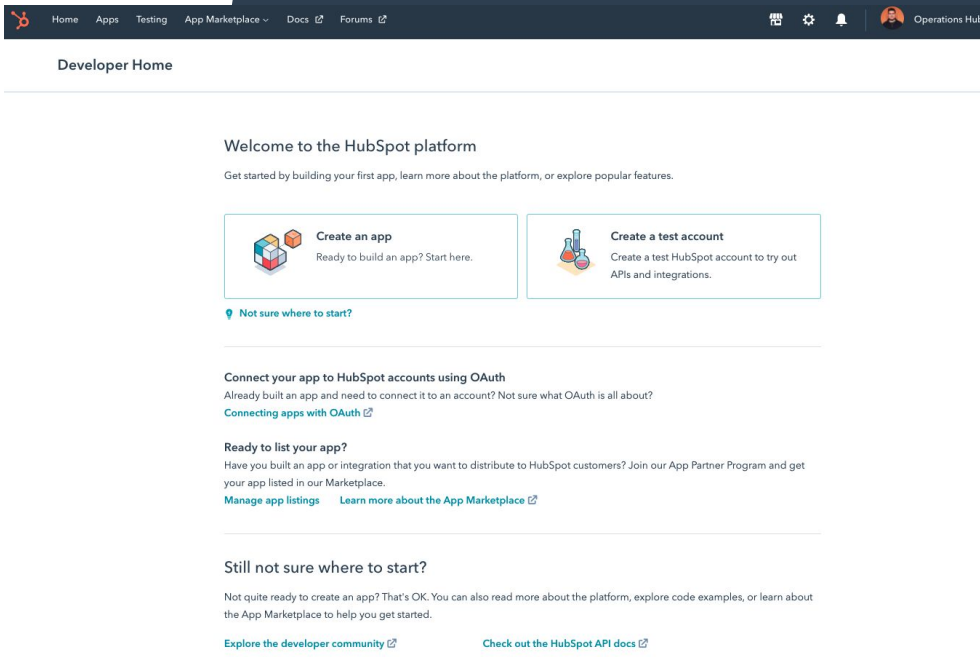
Step 1 Video Walkthrough

Step 1: Setup a Developer Account

An overview of the developer account

When you finish setting up your developer account you will see something similar to the screenshot on the left. There are a couple of items in the navigation bar:

- Apps - Register applications and manage scopes so that it can be installed in different portals.
- Testing - Create test portals to try/test HubSpot functionality. Each lasts 90 days and can be renewed manually for a further 90 days.
- App Marketplace - Manage your app listing
- Docs - Links to the developer documentation
- Forums - Links to the developer community, a great place to ask questions and converse with like minded individuals.



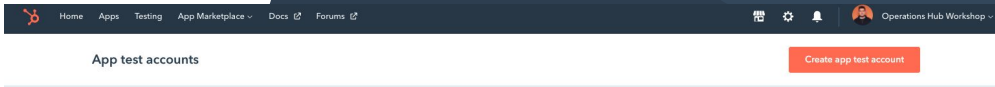
Step 1: Setup a Developer Account

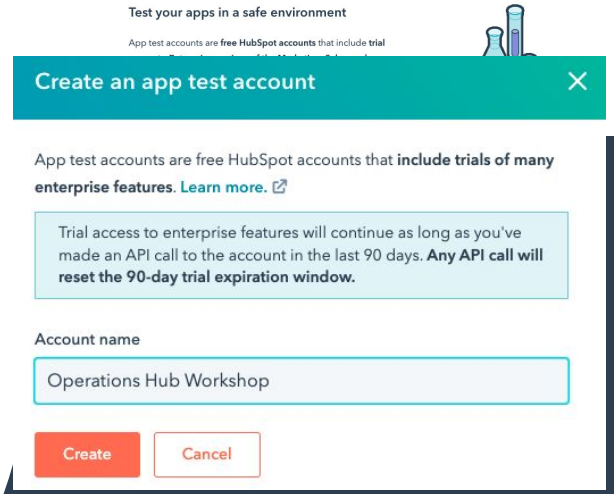
Create a test account

Test accounts are a great way to try/test functionality in an isolated environment. They provide access to the enterprise suite of HubSpot. You can create a maximum of 10 test portals, each of which lasts for a total of 90 days. They can be manually renewed for a further 90 if desired.

1. Click on “Testing” and in the window that appears click “Create app test account”
2. In the pop up give your account a name and click “Ok”
3. You should see the account in the list, click into it to access your enterprise developer environment. This is where we’ll build our custom coded workflow actions.

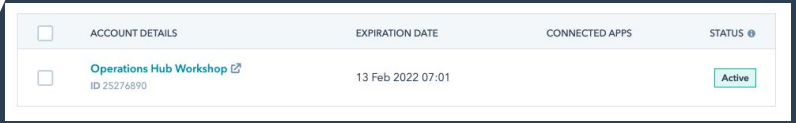
#1





#2

#3



Step 2:
Create the Database Instance

In this section we're going to setup the database that we'll be querying using our custom coded workflow action. As mentioned previously I'll be using Amazon Web Services RDB to set this up. However you're free to use whatever provider you'd like. So long as you've a database setup you are good to go.

Create AWS Free Account

Step 2 Video Walkthrough

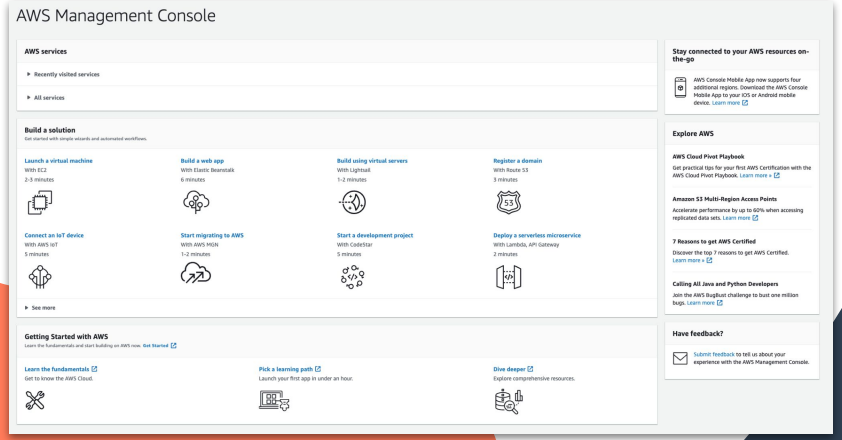
Step 2: Create the Database Instance

Setup an AWS Account

If you don't already have one an important step is to setup an AWS account. You can do this by clicking on the link below:

https://aws.amazon.com/free/

Once you've done this you should have access to your AWS management console:



What is AWS Free?

The AWS Free Tier provides Amazon customers the ability to explore and try out AWS services free of charge up to specified limits for each service. The Free Tier is comprised of three different types of offerings, a 12-month Free Tier, an Always Free offer, and short term trials. Services with a 12-month Free Tier allow customers to use the product for free up to specified limits for one year from the date the account was created. Services with an Always Free offer allow customers to use the product for free up to specified limits as long as they are an AWS customer. Services with a short term trial are free to use for a specified period of time or up to a one-time limit depending on the service selected. Details on the limits and services provided for free are detailed in each card on the Free Tier page. If your application use exceeds the free tier limits, you simply pay standard, pay-as-you-go service rates (see each service page for full pricing details). Restrictions apply; see offer terms for more details.

Step 2: Create the Database Instance

Navigate to the RDS tool

Now that your AWS account is setup and you have access to the management console you are ready to begin setting up your database. First click on “All services” and navigate to to “RDS” under the heading “Database”:

Click here

What is Amazon RDS?

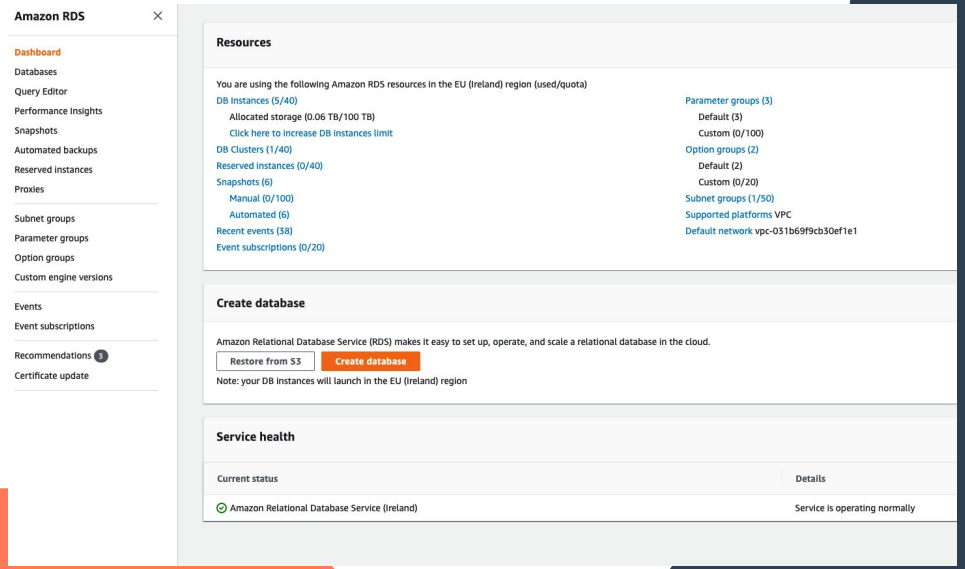
Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as hardware provisioning, database setup, patching and backups. It frees you to focus on your applications so you can give them the fast performance, high availability, security and compatibility they need.

LEARN MORE HERE

Step 2: Create the Database Instance

Create database

Once you've clicked into the RDS tool you should see an interface similar to the one below. From here we will click on the option to "Create database"

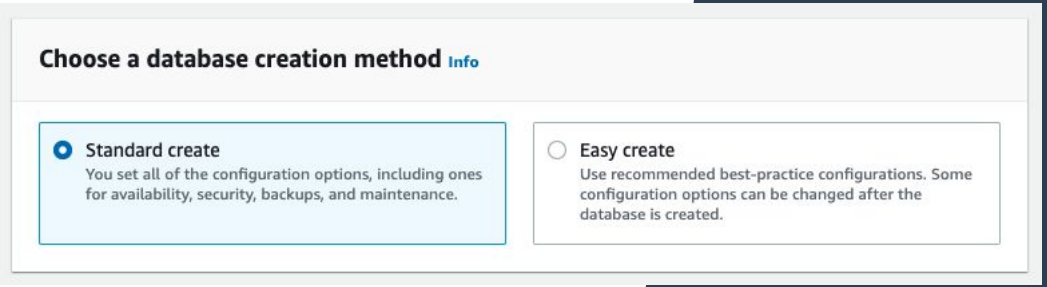


Once you click on the "Create database" option you will be brought into the database creation wizard. This is where we will configure our database and publish. There are a couple of steps involved here but don't worry we'll address them on the next few slides!

Step 2: Create the Database Instance

Choose a database creation method

There are two options here, keep the default selected “Standard create”.

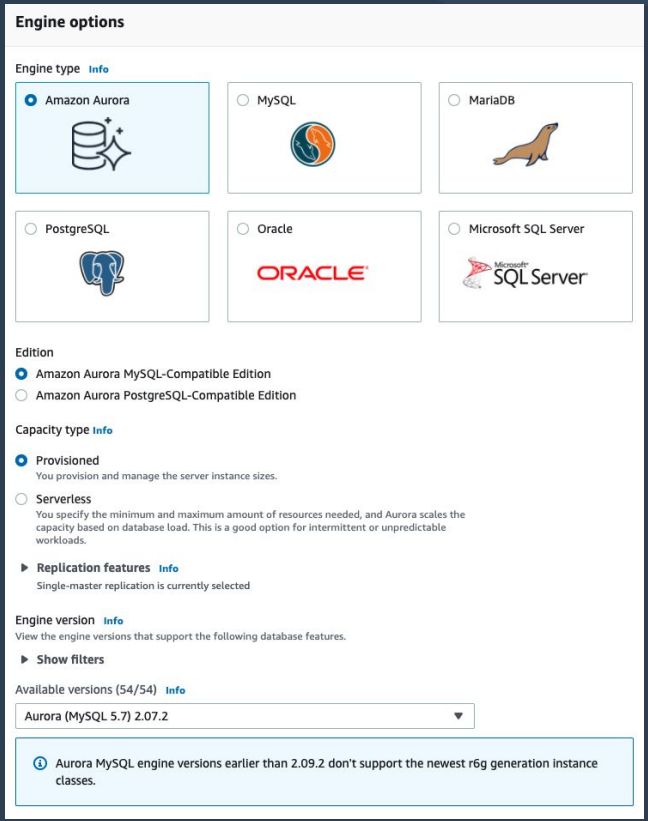


Step 2: Create the Database Instance

Engine Options

There are a number of different types of engines that power databases. AWS gives you the option to define which engine to use. We're going to be using "Amazon Aurora" which is essentially a cloud based MySQL database.

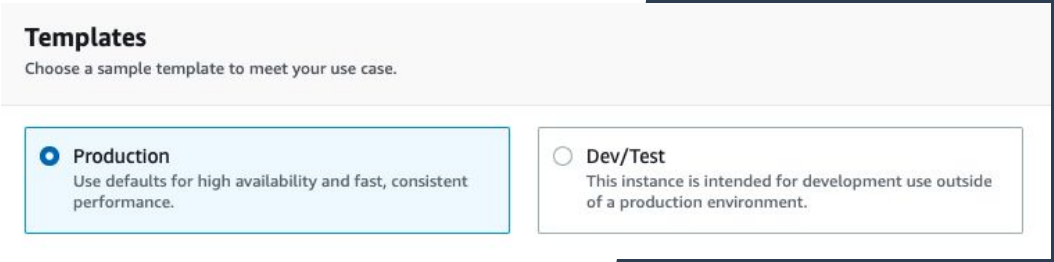
Ensure you have the settings configured as shown on the right. If you'd like to learn more about Amazon Aurora you can do so here.



Step 2: Create the Database Instance

Templates

Keep the default option of “Production” checked.

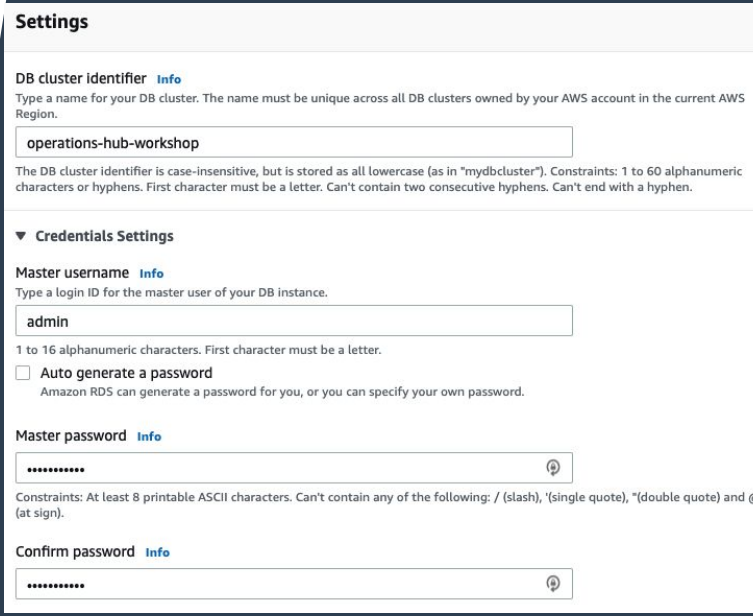


Step 2: Create the Database Instance

Settings

In this section you can give your database a name, it can be anything that you want. You will also define a username and a password to access the database.

We will be covering access to the database later on when we look at connecting via MySQL Workbench and also when we build out the custom coded workflow action.



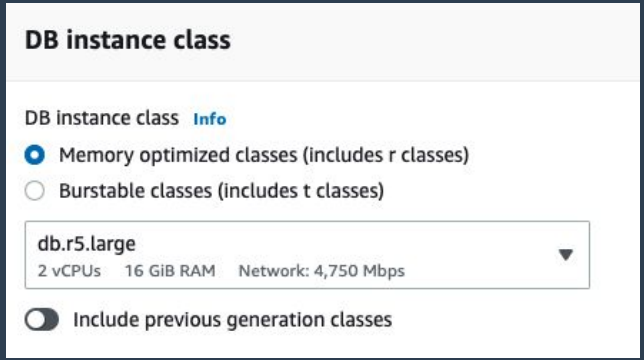
Step 2: Create the Database Instance

DB instance class

The DB instance class is essentially how powerful you need the database to be. In other words it defines the speed and memory capacity of the database.

There are several different types and some come at a cost. We can keep the default settings as we don't need anything to powerful and we're using the AWS Free Tier.

If you are interested in learning more about DB instance classes check out this article.



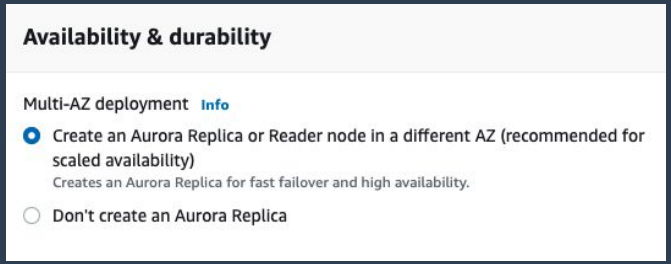
Step 2: Create the Database Instance

Availability & durability

One of the great things about AWS is that it is all in the cloud and as a result there is built in redundancy in the event your database was to go down.

What this setting ensures is that there is a replica database in a different availability zone that is kept in sync with our primary database.

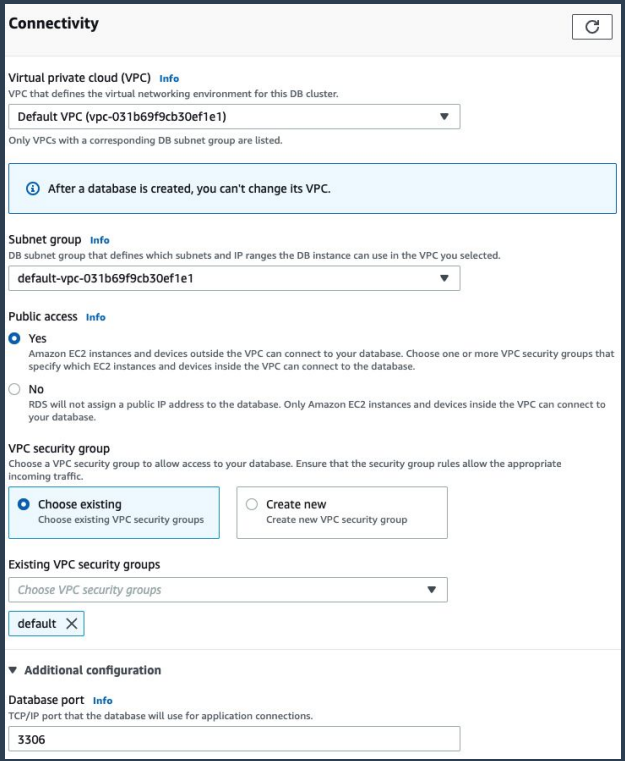
You don't have to do this but it's always good to factor in redundancy and failovers into any solution. More information here.



Step 2: Create the Database Instance

Connectivity

Quite a lot of settings here but the most important is “Public access” make sure you set this to “Yes” otherwise you will be unable to connect to the database from HubSpot.

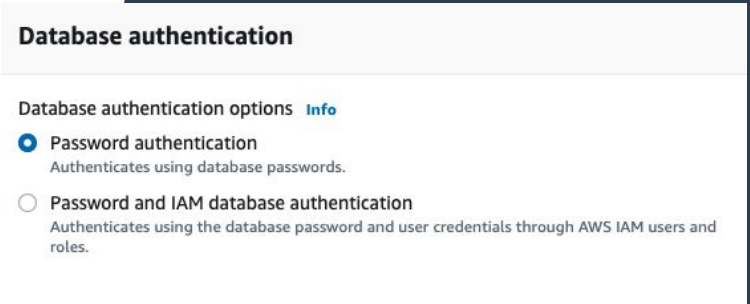


Step 2: Create the Database Instance

Database authentication

Two options here, we're only concerned with "Password authentication" - in other words to access the database you need to provide a password.

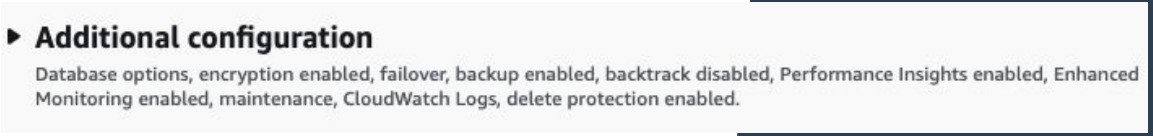
The other option is using IAM which stands for Identity Access Management and rather than using passwords to access a database you use a token provisioned for users.



Step 2: Create the Database Instance

Additional Configuration

There are a couple of options available in this section but for the purposes of this workshop we do not need to make any changes.



Step 2: Create the Database Instance

Database Overview

On the next screen you should see a table similar to the below. It might take a couple of minutes (potentially up to 20 minutes) for your database to become “Available” - don’t worry that is expected!

RDS > Databases

Databases

Group resources | Modify | Actions | Restore from S3 | Create database

oper

1

DB identifier	Role	Engine	Region & AZ	Size	Status	CPU	Current activity	Maintenance	VPC	Multi-AZ
operations-hub-workshop	Regional cluster	Aurora MySQL	eu-west-1	2 instances	Available	-		none	-	-
operations-hub-workshop-instance-1	Writer instance	Aurora MySQL	eu-west-1a	db.r5.large	Modifying	20.12%	0.00 sessions	none	vpc-031b69f9cb30ef1e1	2 Zones
operations-hub-workshop-instance-1-eu-west-1c	Reader instance	Aurora MySQL	eu-west-1c	db.r5.large	Creating	-		none	vpc-031b69f9cb30ef1e1	2 Zones

Step 2: Create the Database Instance

Database Overview

If you select a database you'll notice the section labelled "Connectivity & Security" this information is very important as we'll need it to connect and ultimately configure the database. The most important parts are:

- Endpoint
- Port
- Username
- Password

RDS > Databases > operations-hub-workshop > operations-hub-workshop-instance-1

operations-hub-workshop-instance-1

Related

oper

DB identifier	Role	Engine	Region & AZ	Size	Status	CPU	Current activity	Maintenance
operations-hub-workshop	Regional cluster	Aurora MySQL	eu-west-1	2 instances	Available	-		none
operations-hub-workshop-instance-1	Writer instance	Aurora MySQL	eu-west-1a	db.r5.large	Modifying	20.12%	0.00 sessions	none
operations-hub-workshop-instance-1-eu-west-1c	Reader instance	Aurora MySQL	eu-west-1c	db.r5.large	Creating	-		none

Connectivity & security | Monitoring | Logs & events | Configuration | Maintenance | Tags

Connectivity & security

Endpoint & port

Endpoint
operations-hub-workshop-instance-1.ckv5jtcuilgzo.eu-west-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
eu-west-1a

VPC
vpc-031b69f9cb30ef1e1

Subnet group
default-vpc-031b69f9cb30ef1e1

Subnets
subnet-0cb525a3e160555b62
subnet-0d63887f69afce4ec
subnet-0dcc8df8b370e488a

Security

VPC security groups
default (sg-0790de851dae8b45a)
Active

Publicly accessible
Yes

Certificate authority
rds-ca-2019

Certificate authority date
August 22, 2024, 06:08 (UTC+6:08)

Security group rules (3)

Filter by security group rules

Security group	Type	Rule
default (sg-0790de851dae8b45a)	CIDR/IP - Inbound	0.0.0.0/0
default (sg-0790de851dae8b45a)	EC2 Security Group - Inbound	sg-0790de851dae8b45a
default (sg-0790de851dae8b45a)	CIDR/IP - Outbound	0.0.0.0/0

Step 3:
Configure the Database Instance

Now that our database has been setup we're ready to configure it. What we mean by configuration is setting up a table to house the data. For this portion of the workshop we'll be using MySQL Workbench a popular client used for managing MySQL databases.

Download MySQL Workbench | Get the Code | Step 3 Video Walkthrough

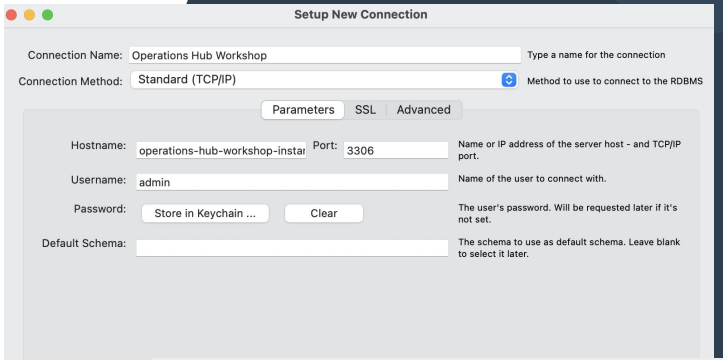
Step 3: Configure the Database Instance

Connect to the instance

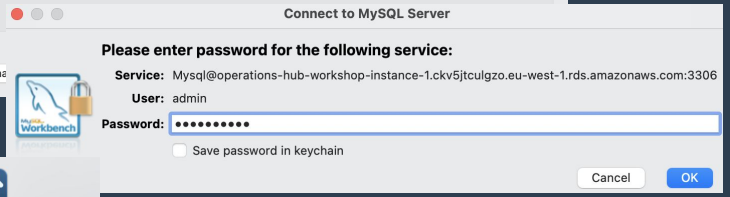
Open MySQL Workbench and click on the ⊕ icon. This will open up the wizard to connect to the database of choice.

1. Give it whatever “Connection Name” you’d like and leave “Connection Method” to “Standard (TCP/IP).Next, enter the “Hostname”. This can be found under “Endpoint & port” when viewing your AWS Database.
2. Then enter your username and click “Test connection”. In the pop up that appears enter your password.
3. You should see another pop up showing whether or not the connection was successful.

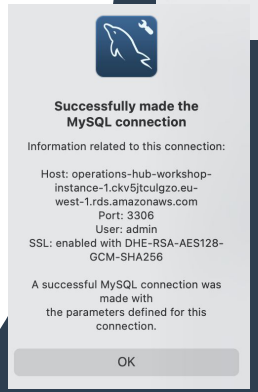
#1



#2



#3

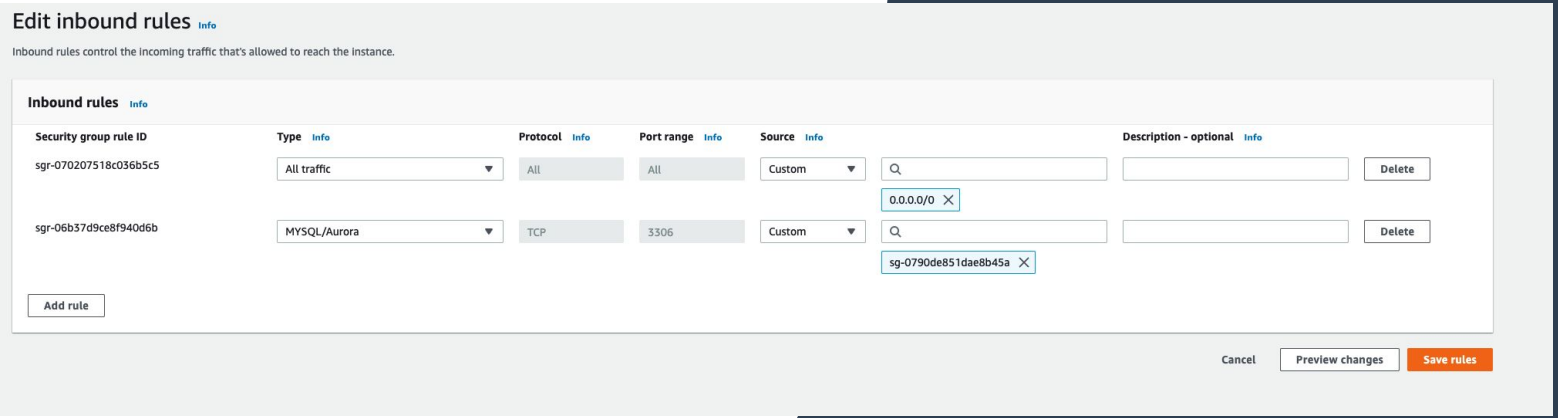


Step 3: Configure the Database Instance

Issue Connecting

If you are having trouble connecting to the database instance double check your username and password. If you are still having issues you will need to add an inbound rule to your security group.

Inbound rules are ways of managing incoming traffic to your AWS instance. Make sure you have one that allows for “All traffic”, see below. More information on inbound rules can be found here.



Step 3: Configure the Database Instance

Create the database

Now that the connection has been setup, we can easily connect to our AWS database instance and configure as needed. We are going to “Create a database” and create a table in that database to hold “customer information”. This is done using MySQL commands, you can copy the below into the MySQL query window and click ⚡ to execute.

```
CREATE DATABASE customers;
USE customers;
CREATE TABLE customer_info(
    CustomerID int AUTO_INCREMENT,
    FirstName varchar(255),
    LastName varchar(255),
    EmailAddress varchar(255),
    PRIMARY KEY(CustomerID)
)
SELECT * FROM customer_info
```

Limit to 1000 rows

```
/* create database "customers" */
CREATE DATABASE customers;

/* use the database "customers" */
USE customers;

/*
  Create table in the "customers" database called "customer_info".
  This will have columns to hold data relating to our customers namely
  their name and email address. CustomerID is the primary key and is an auto-generaged
  unique number.
*/
CREATE TABLE customer_info(
    CustomerID int AUTO_INCREMENT,
    FirstName varchar(255),
    LastName varchar(255),
    EmailAddress varchar(255),
    PRIMARY KEY (CustomerID)
);

/*
 Retrieve all of the rows within the "customer_info" table.
 Initially this should return nothing but over time it will populate when you
 run your custom coded worklfow actions in HubSpot
*/
SELECT * FROM customer_info;
```

Step 3: Configure the Database Instance

Query the database

From within MySQL workbench you can query the table to see if it contains any data. Right now it shouldn't as we've yet to build our workflow but worth noting at any stage you can use the below command to query

```
SELECT * FROM customer_info
```

Before

Result Grid | Filter Rows: Search | Edit: | Export/Impo

CustomerID	FirstName	LastName	EmailAddress
NULL	NULL	NULL	NULL

After*

Result Grid | Filter Rows: Search | Edit: | Export/Import:

CustomerID	FirstName	LastName	EmailAddress
1	Bruce	Banner	thehulk@marvel.com
2	Peter	Parker	spiderman@marvel.com
3	Steve	Rogers	captainamerica@marvel.com
4	Tony	Stark	ironman@starkindustries.com
NULL	NULL	NULL	NULL

**Results shown after 4 contacts were enrolled into the workflow*

Step 4:
Create the Workflow

Now that the database is setup, it is time to create our workflow within HubSpot. The workflow will enroll any contacts who have been created within the CRM, use a custom coded workflow action to query the database and what happens next depends on whether or not a result was found.

Get the Code

Step 4 Video Walkthrough

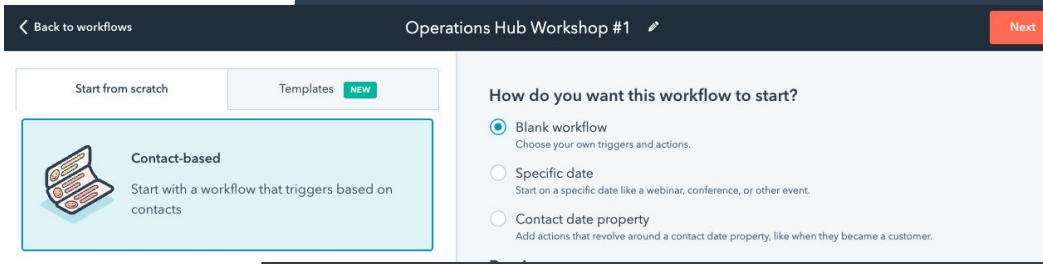
Step 4: Create the Workflow

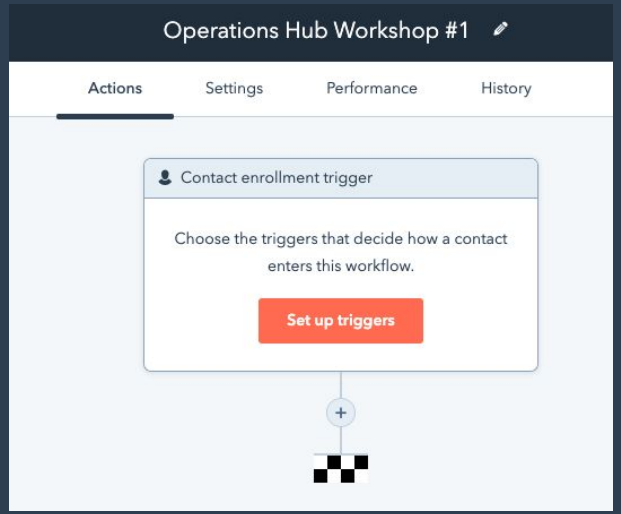
Create Workflow

Within your test account hover over “Automation” and select “Workflows”. You are now within HubSpot's automation tool.

From here select “Create workflow”. You will be prompted to select a template, instead click on the “start from scratch” tab and ensure “Contact-based” is selected.

Once this is done click on “next” and you’ll be taken into the workflow builder screen.

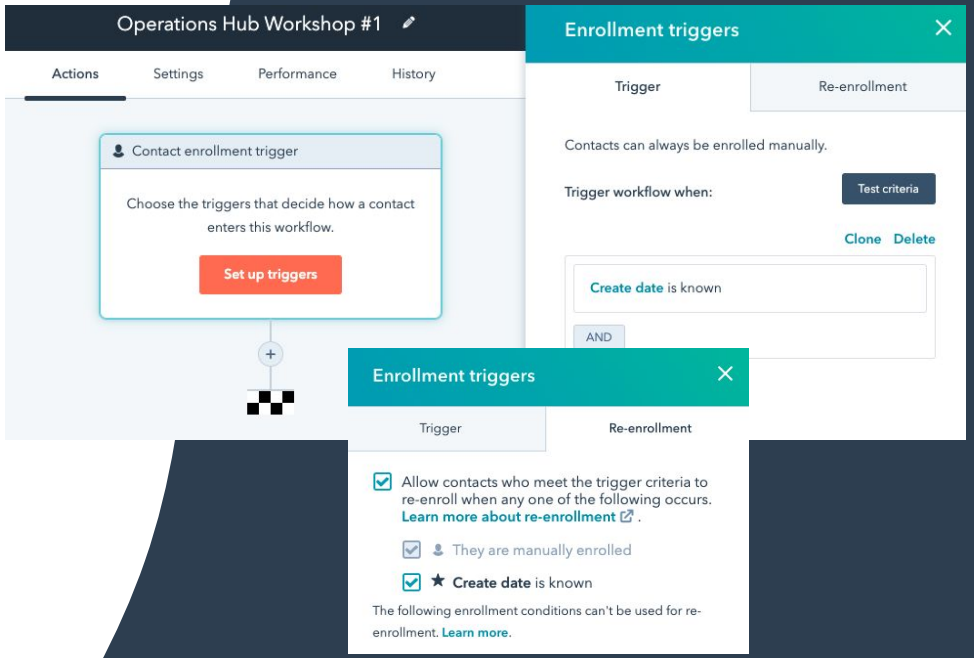




Step 4: Create the Workflow

Define enrollment Criteria

Now we must define our enrollment criteria, in theory you could set this to anything you like but for the purposes of this workshop we're simply going to use "Create date is known". In other words, any time a contact is created in the CRM this workflow will be triggered.

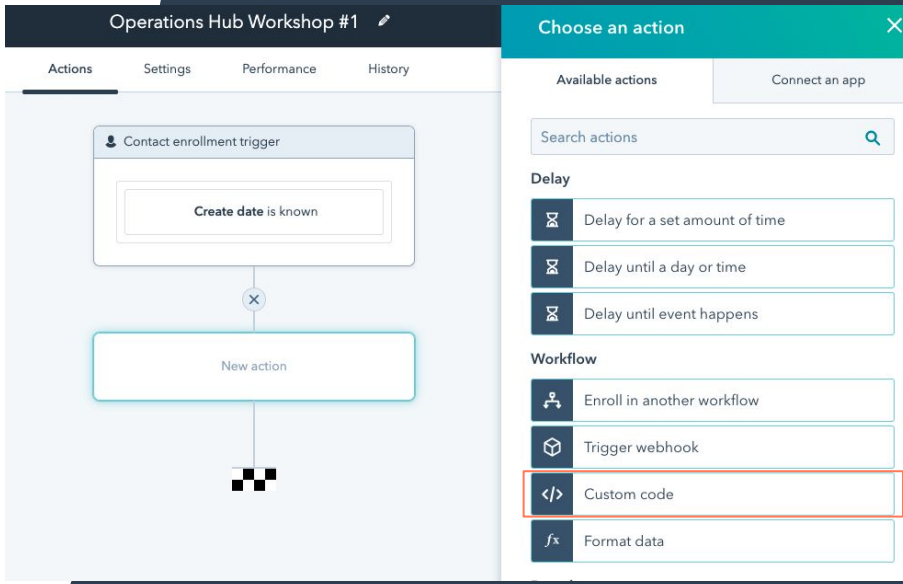


Step 4: Create the Workflow

Choose an action

There are many different types of actions you can choose to execute within a workflow. For the purposes of this workshop we're going to be using "Custom Coded Action" - exclusive to Operations Hub Professional.

Worth noting Operations Hub also gives you the "Trigger webhook" action and "Format data" action. They're not going to be leveraged in this workshop.



Step 4: Create the Workflow

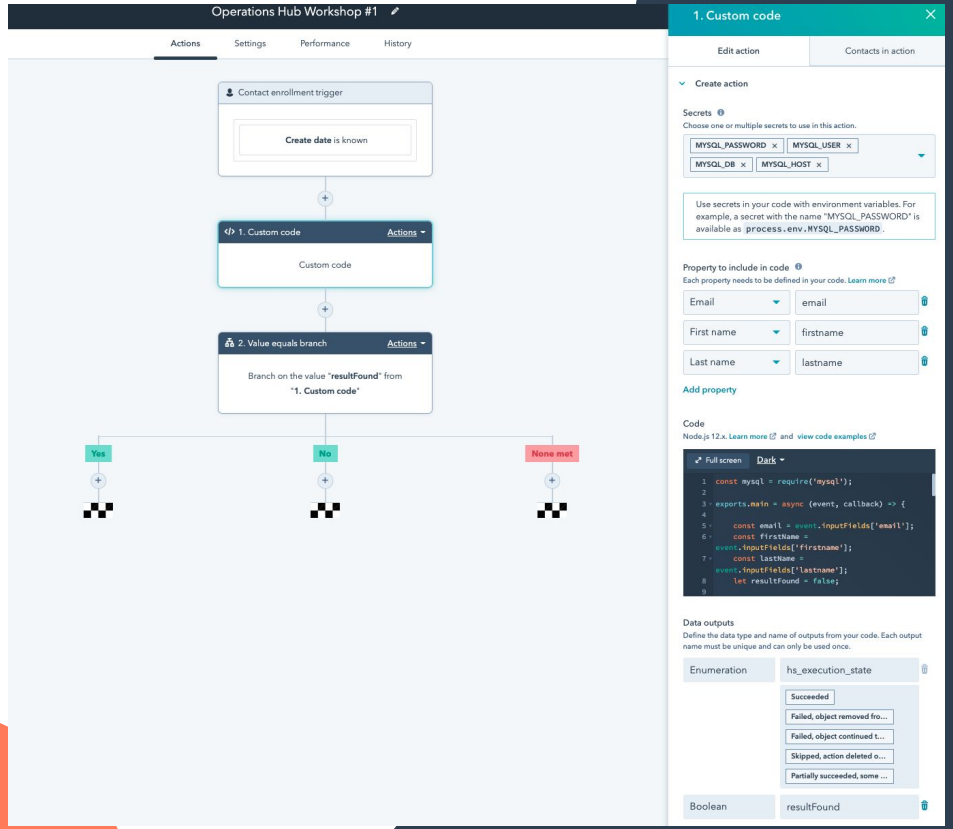
Create custom coded action

When you click on the “Custom code” action a sidebar similar to the one shown on the left will appear. Couple of things to point out…

- Secrets - define any sort of variables like usernames, passwords that you’d like to reference in your code but keep private.
- Property to include in code - define any properties relating to the enrolled object that you’d like to reference in your code.
- Code - Your javascript, we provide a template by default.
- Data outputs - define any data you’d like the custom code action to pass back to the workflow to reference at a later step.

Step 4: Create the Workflow

The finished product



Step 5:
Test the Workflow

Providing you've now setup the database and the workflow it is time to test to ensure all is working as expected. Create a contact within the CRM, they should be enrolled into the workflow.

Step 5 Video Walkthrough

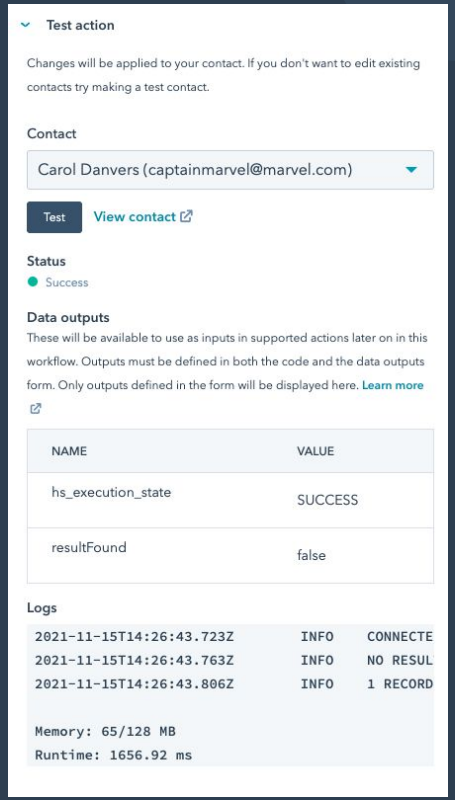
Step 4: Test the Workflow

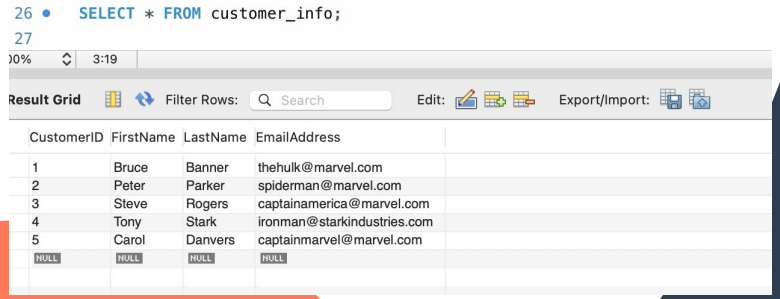
Test the Workflow

All custom coded workflow actions can be tested prior to turning the workflow on. Simply click on the “Test action” title in the sidebar. Enter the name of the contact to enroll and click “Test”. You should see something similar to the screenshot on the slide.

Also feel free to run “SELECT * FROM customer_info” in MySQL Workbench, you should see rows being added to your database.

If you are happy with the results you can turn your workflow on and any time you add contacts to the CRM it should sync to your database.





CustomerID	FirstName	LastName	EmailAddress
1	Bruce	Banner	thehulk@marvel.com
2	Peter	Parker	spiderman@marvel.com
3	Steve	Rogers	captainamerica@marvel.com
4	Tony	Stark	ironman@starkindustries.com
5	Carol	Danvers	captainmarvel@marvel.com
NULL	NULL	NULL	NULL

Congratulations!

You've successfully created a custom coded workflow action that can connect and query a MySQL database hosted on AWS.

Resources

- 3 Ways to Use Custom Coded Workflow Actions - Developer blog post
- Use Custom Coded Actions in Workflows - Knowledge base article
- Custom Coded Workflow Actions - Developer Documentation
- Programmable Automation Use Case Library
- Operations Hub Product Page
- Amazon relational Database Service (RDS) User Guide
- Community Workshops
  - Workshop #1
  - Workshop #2
  - Workshop #3

There are a ton of useful resources online relating to operations hub and custom coded workflow actions specifically. On the left are some of the resources I would recommend reviewing to learn more. I'd also highly recommend checking out my blog post below