1.split function:

Split():

split splits a string into an array of substrings, based on a delimiter that you choose, and returns the item specifieby the position. You can only add split to a calculated field during data preparation, not to an analysis. This function is not supported in direct queries to Microsoft SQL Server.

**Syntax**

split(*expression*, *delimiter* , *position*)

**Example**The following example splits a string into an array, using the semicolon character (;) as the delimiter, and returns the third element of the array.

split('123 Test St;1402 35th Ave;1818 Elm Ct;11 Janes Lane', ';', 3)

output:

The following item is returned.

1818 Elm Ct

This function skips items containing null values or empty strings.

2.Locate function:

locate locates a substring that you specify within another string, and returns the number of characters until the first

character in the substring. The function returns 0 if it doesn't find the substring. The function is 1-based.

**Syntax:**

locate(*expression*, *substring*, *start*)

**Examples:**The following example returns information about where the first occurrence of the substring 'and' appears in a string.

locate('1 and 2 and 3 and 4', 'and')

The following value is returned.

0/p:

3

The following example returns information about where the first occurrence of the substring 'and' appears in a string after the fourth character.

locate('1 and 2 and 3 and 4', 'and', 4)

The following value is returned.

o/p:

9

**Substring function:** substring returns the characters in a string, starting at the location specified by the *start* argument and proceeding for the number of characters specified by the *length* arguments.

**Syntax**

substring(*expression*, *start*, *length*)

**Example**

The following example returns the 13th through 19th characters in a string. The beginning of the string is index 1, so you begin counting at the first character.

substring('Fantasy and Science Fiction',13,7)

The following value is returned.

o/p:

Science

**Adding or updating DAGs:**

Directed Acyclic Graphs (DAGs) are defined within a Python file that defines the DAG's structure as code. You can use the AWS CLI, or the Amazon S3 console to upload DAGs to your environment. This page describes the steps to add or update Apache Airflow DAGs on your Amazon Managed Workflows for Apache Airflow environment using the dags folder in your Amazon S3 bucket.

**Uploading DAG code to Amazon S3**

You can use the Amazon S3 console or the AWS Command Line Interface (AWS CLI) to upload DAG code to your Amazon S3 bucket. The following steps assume you are uploading code (.py) to a folder named dags in your Amazon S3 bucket.

**Using the AWS CLI**

The AWS Command Line Interface (AWS CLI) is an open source tool that enables you to interact with AWS services using commands in your command-line shell. To complete the steps on this page, you need the following:

* [AWS CLI – Install version 2](https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html).
* [AWS CLI – Quick configuration with aws configure](https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-configure.html).

**To upload using the AWS CLI**

1. Use the following command to list all of your Amazon S3 buckets.

aws s3 ls

1. Use the following command to list the files and folders in the Amazon S3 bucket for your environment.

aws s3 ls s3://*YOUR\_S3\_BUCKET\_NAME*

1. The following command uploads a dag\_def.py file to a dags folder.

aws s3 cp dag\_def.py s3://*YOUR\_S3\_BUCKET\_NAME*/dags/

If a folder named dags does not already exist on your Amazon S3 bucket, this command creates the dags folder and uploads the file named dag\_def.py to the new folder.

**Using the Amazon S3 console**

The Amazon S3 console is a web-based user interface that allows you to create and manage the resources in your Amazon S3 bucket. The following steps assume you have a DAGs folder named dags.

**To upload using the Amazon S3 console**

1. Open the [Environments page](https://console.aws.amazon.com/mwaa/home#/environments) on the Amazon MWAA console.
2. Choose an environment.
3. Select the **S3 bucket** link in the **DAG code in S3** pane to open your storage bucket on the Amazon S3 console.
4. Choose the dags folder.
5. Choose **Upload**.
6. Choose **Add file**.
7. Select the local copy of your dag\_def.py, choose **Upload**.