

# How to use Intel® DAAL K-Means Clustering via SageMaker web interface

## Description of algorithm:

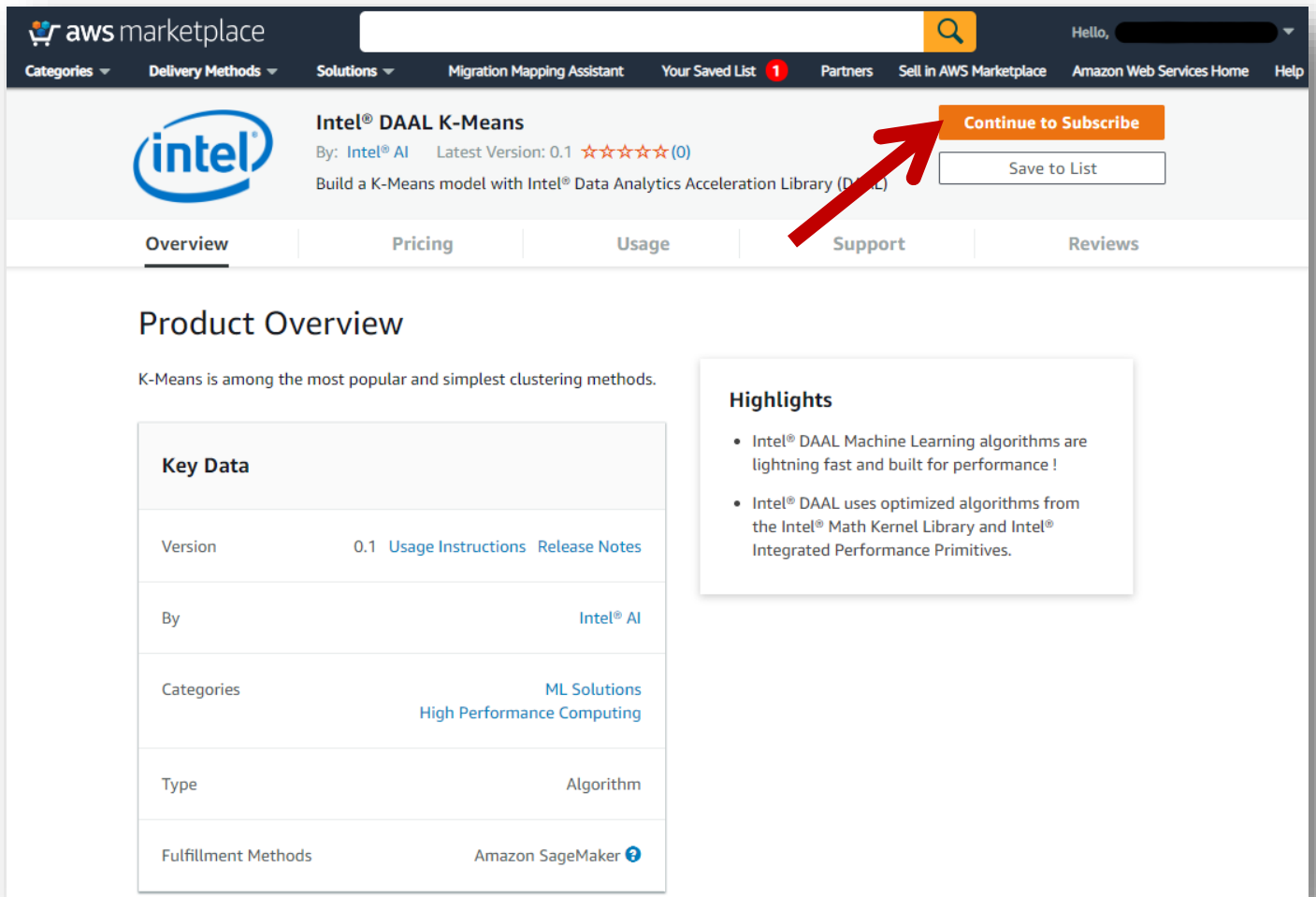
K-Means is among the most popular and simplest clustering methods. It is intended to partition a data set into a small number of clusters such that feature vectors within a cluster have greater similarity with one another than with feature vectors from other clusters. Each cluster is characterized by a representative point, called a centroid, and a cluster radius. In other words, the clustering methods enable reducing the problem of analysis of the entire data set to the analysis of clusters. There are numerous ways to define the measure of similarity and centroids. For K-Means, the centroid is defined as the mean of feature vectors within the cluster.

[Intel® DAAL developer guide](#)

[Intel® DAAL documentation for K-Means](#)

## Instruction:

1. Visit page on SageMaker Marketplace and click “Continue to Subscribe”



The screenshot shows the AWS Marketplace page for Intel® DAAL K-Means. The page header includes the AWS Marketplace logo, a search bar, and navigation links. The main content area features the Intel logo, the product name 'Intel® DAAL K-Means', and a 'Continue to Subscribe' button highlighted by a red arrow. Below the product name, there are tabs for Overview, Pricing, Usage, Support, and Reviews. The 'Overview' tab is selected, showing a 'Product Overview' section with a description of K-Means and a 'Key Data' table. To the right of the table is a 'Highlights' section with two bullet points.

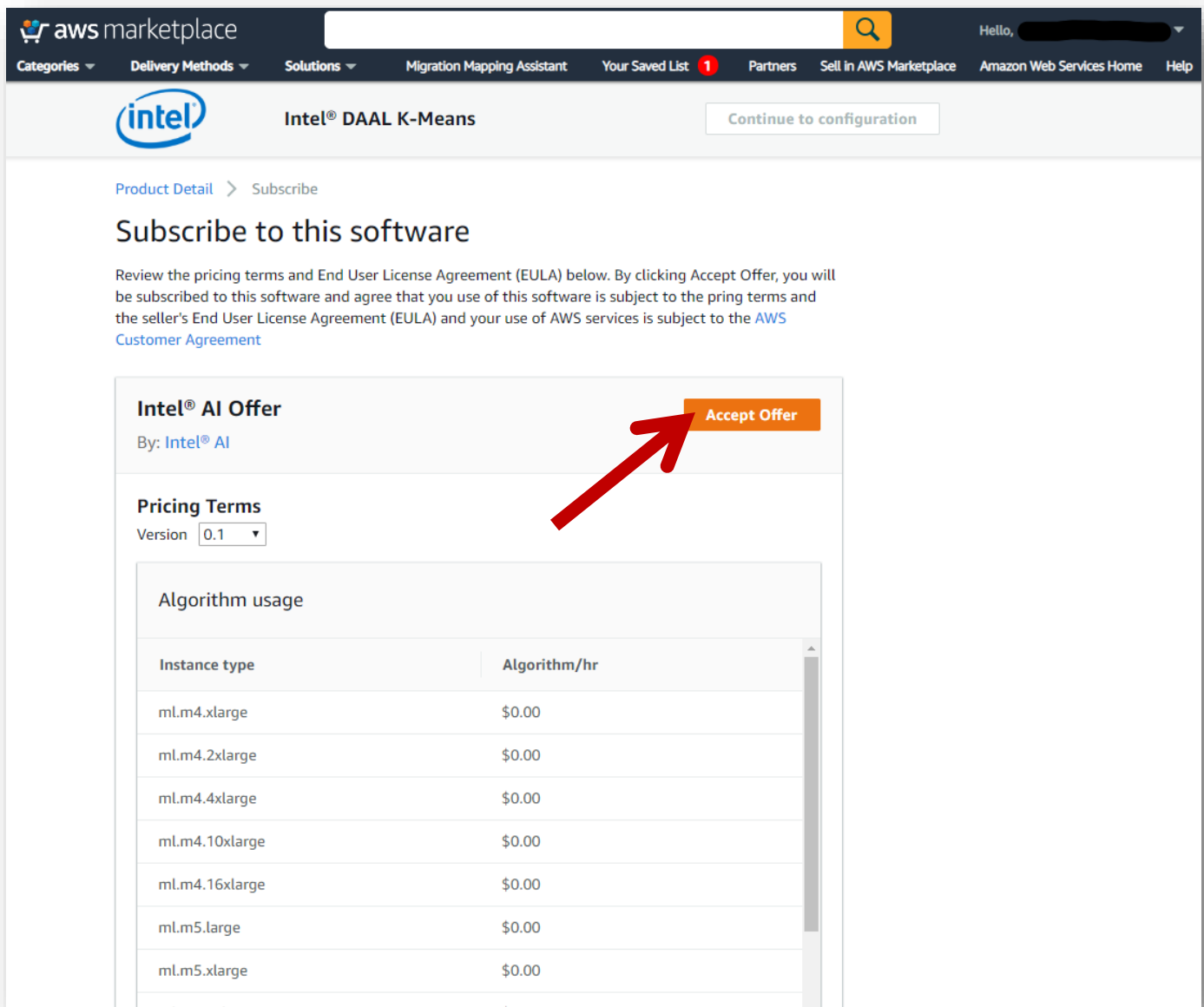
**Key Data**

Version	0.1 <a href="#">Usage Instructions</a> <a href="#">Release Notes</a>
By	<a href="#">Intel® AI</a>
Categories	<a href="#">ML Solutions</a> <a href="#">High Performance Computing</a>
Type	Algorithm
Fulfillment Methods	<a href="#">Amazon SageMaker</a>

**Highlights**

- Intel® DAAL Machine Learning algorithms are lightning fast and built for performance !
- Intel® DAAL uses optimized algorithms from the Intel® Math Kernel Library and Intel® Integrated Performance Primitives.

- Click "Accept Offer" if you agree with EULA at end of page. If you already subscribed on algorithm on Marketplace this step will be skipped.



The screenshot shows the AWS Marketplace interface for the Intel® DAAL K-Means product. The top navigation bar includes the AWS Marketplace logo, a search bar, and links to Categories, Delivery Methods, Solutions, Migration Mapping Assistant, Your Saved List (with a red notification icon), Partners, Sell in AWS Marketplace, Amazon Web Services Home, and Help. The product header features the Intel logo, the product name 'Intel® DAAL K-Means', and a 'Continue to configuration' button. Below the header, the page title is 'Subscribe to this software'. A paragraph of text explains the subscription process and mentions the EULA. A red arrow points to the 'Accept Offer' button. Below the button, the 'Pricing Terms' section shows a version dropdown set to '0.1'. The 'Algorithm usage' table lists various instance types and their corresponding pricing.

Instance type	Algorithm/hr
ml.m4.xlarge	\$0.00
ml.m4.2xlarge	\$0.00
ml.m4.4xlarge	\$0.00
ml.m4.10xlarge	\$0.00
ml.m4.16xlarge	\$0.00
ml.m5.large	\$0.00
ml.m5.xlarge	\$0.00
ml.m5.2xlarge	\$0.00

3. Click "Continue to configuration"

The screenshot shows the AWS Marketplace interface for the Intel DAAL K-Means software. The top navigation bar includes the AWS Marketplace logo, a search bar, and links for Categories, Delivery Methods, Solutions, Migration Mapping Assistant, Your Saved List (with a red notification badge), Partners, Sell in AWS Marketplace, Amazon Web Services Home, and Help. The main header features the Intel logo and the product name 'Intel® DAAL K-Means'. A red arrow points to the 'Continue to configuration' button in the top right corner of the product header.

Product Detail > Subscribe

## Subscribe to this software

Review the pricing terms and End User License Agreement (EULA) below. By clicking Accept Offer, you will be subscribed to this software and agree that you use of this software is subject to the pring terms and the seller's End User License Agreement (EULA) and your use of AWS services is subject to the [AWS Customer Agreement](#)

✔ Thank you for subscribing. You can now view your product. ✕

**Intel® AI Offer** Already Subscribed

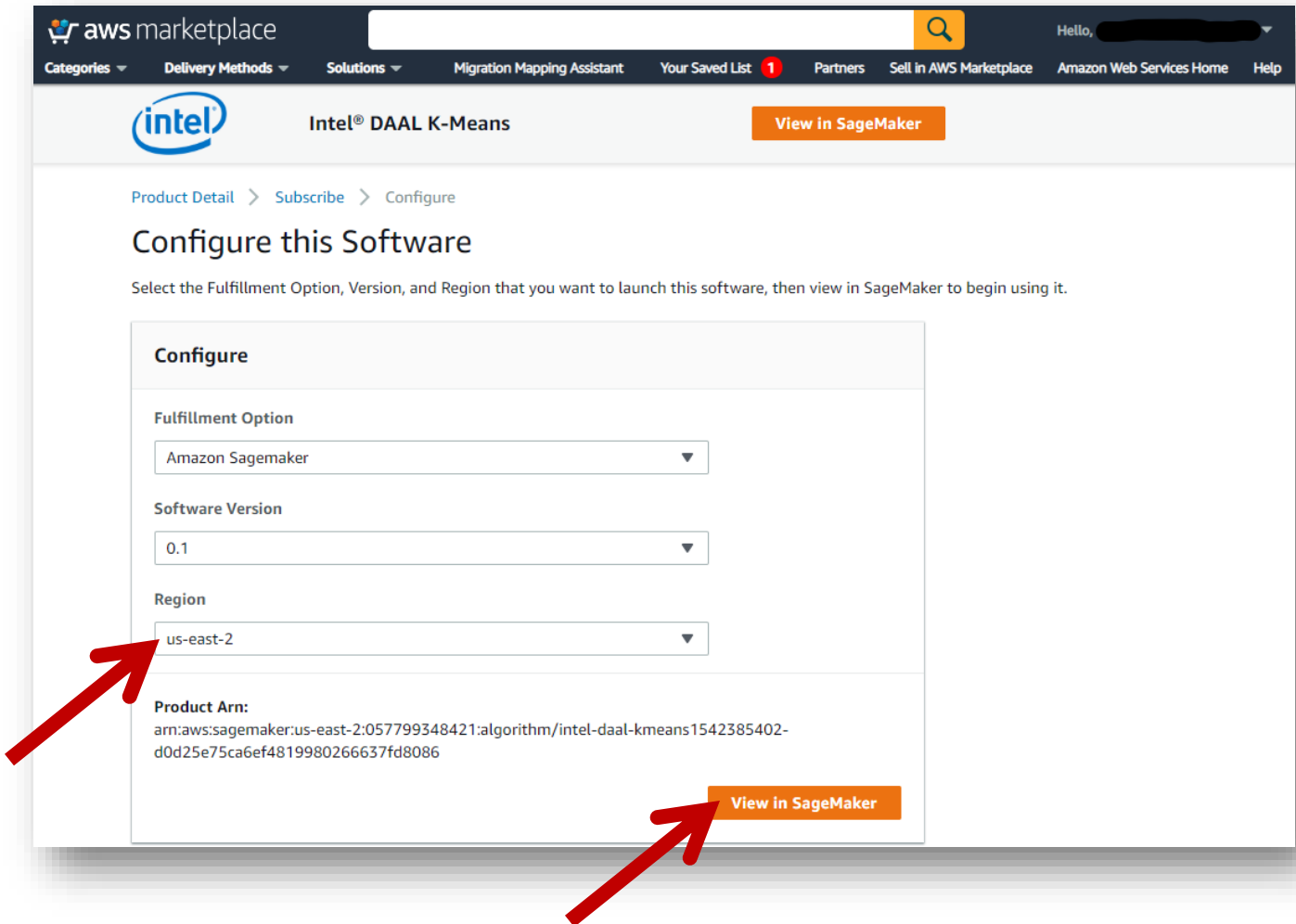
By: Intel® AI

**Pricing Terms**

Version

Algorithm usage	
Instance type	Algorithm/hr
mL.m4.xlarge	\$0.00
mL.m4.2xlarge	\$0.00
mL.m4.4xlarge	\$0.00
mL.m4.10xlarge	\$0.00
mL.m4.16xlarge	\$0.00
mL.m5.large	\$0.00
mL.m5.xlarge	\$0.00

#### 4. Choose the Region and click “View in SageMaker”



K-Means algorithm on AWS SageMaker is divided into two stages: training job and getting inference from endpoint.

Training job is computing clusters centroids and other values from provided training data.

After that, you should create model with computed values and endpoint based on it.

Sending data to endpoint gives you numbers of clusters for provided data in response.

5. Select needed algorithm version and click “Create training job”

The screenshot shows the Amazon SageMaker console interface. On the left is a navigation sidebar with options like Dashboard, Search, Ground Truth, Notebook, Training, and Inference. The main content area is titled 'Intel® DAAL K-Means' and shows a table of 'Algorithm versions'. The table has columns for Title, Version, and Algorithm ARN. The first row is selected, showing 'Intel® DAAL K-Means' with version '0.1' and a specific ARN. To the right of the table are two buttons: 'Create hyperparameter tuning job' and 'Create training job'. A red arrow points from the 'Create training job' button towards the right.

Title	Version	Algorithm ARN
Intel® DAAL K-Means	0.1	arn:aws:sagemaker:us-east-2:057799348421:algorithm/intel-daal-kmeans1542385402-d0d25e75ca6ef4819980266637fd8086

6. Type job name, select IAM role and instance type

The screenshot shows the 'Job settings' form in the Amazon SageMaker console. The form has several sections: 'Job name' with a text input field and a description; 'IAM role' with a dropdown menu showing 'AmazonSageMaker-ExecutionRole-'; 'Algorithm options' with a section for 'Algorithm source' and 'Choose an algorithm subscription'; and 'Resource configuration' with fields for 'Instance type' (set to 'ml.m4.xlarge'), 'Instance count' (set to '1'), and 'Additional storage volume per instance (GB)' (set to '1'). Three red arrows point to the 'Job name' input field, the 'IAM role' dropdown, and the 'Instance type' dropdown.

## 7. Choose hyperparameters

### Hyperparameters

You can use hyperparameters to finely control training. We've set default hyperparameters for the algorithm you've chosen.

Key	Value
fptype	double
nClusters	2
initMethod	defaultDense
oversamplingFactor	0.5
nRounds	5
seed	777
method	lloydDense
maxIterations	100
accuracyThreshold	0
gamma	1
distanceType	euclidean
assignFlag	True

Parameter name	Type	Default value	Description
fptype	str	"double"	The floating-point type that the algorithm uses for intermediate computations. Can be "float" or "double"
nClusters	int	2	The number of clusters
initMethod	str	"defaultDense"	Available initialization methods for K-Means clustering: defaultDense - uses first nClusters points as initial clusters, randomDense - uses random nClusters points as initial clusters, plusPlusDense - uses K-Means++ algorithm; parallelPlusDense - uses parallel K-Means++ algorithm
oversamplingFactor	float	0.5	A fraction of nClusters in each of nRounds of parallel K-Means++. $L = nClusters * oversamplingFactor$ points are sampled in a round
nRounds	int	5	The number of rounds for parallel K-Means++. $(L * nRounds)$ must be greater than nClusters
seed	int	777	The seed for random number generator
method	str	"lloydDense"	Computation method for K-Means clustering
maxIterations	int	100	The number of iterations
accuracyThreshold	float	0	The threshold for termination of the algorithm
gamma	float	1	The weight to be used in distance calculation for binary categorical features
distanceType	str	"euclidean"	The measure of closeness between points (observations) being clustered. The only distance type supported so far is the Euclidian distance
assignFlag	bool	True	A flag that enables computation of assignments, that is, assigning cluster indices to respective observations

## 8. Specify S3 location of input data for training

### Input data configuration

Create up to 8 channels of input sources. If the algorithm you chose supports multiple input channels, you can specify those here. See [Algorithms Provided by Amazon SageMaker: Common Parameters](#)

#### Channels

##### ▼ training

Remove

Channel name

training

Input mode - optional

Pipe

Content type - optional

text/csv

Choose one of the formats below

- text/csv

Compression type

None

Record wrapper

None

S3 data type

S3Prefix

S3 data distribution type

FullyReplicated

S3 location

s3://bucket/path-to-your-data/

Add channel



9. Specify S3 output path (model will be stored here) and click “Create training job”

### Output data configuration

S3 output path

s3://bucket/path-to-your-data/

Encryption key - *optional*

If you want Amazon SageMaker to encrypt the output of your training job using your own AWS KMS encryption key instead of the default S3 service key, provide its ID or ARN.

### ▼ Tags - *optional*

Key

Value

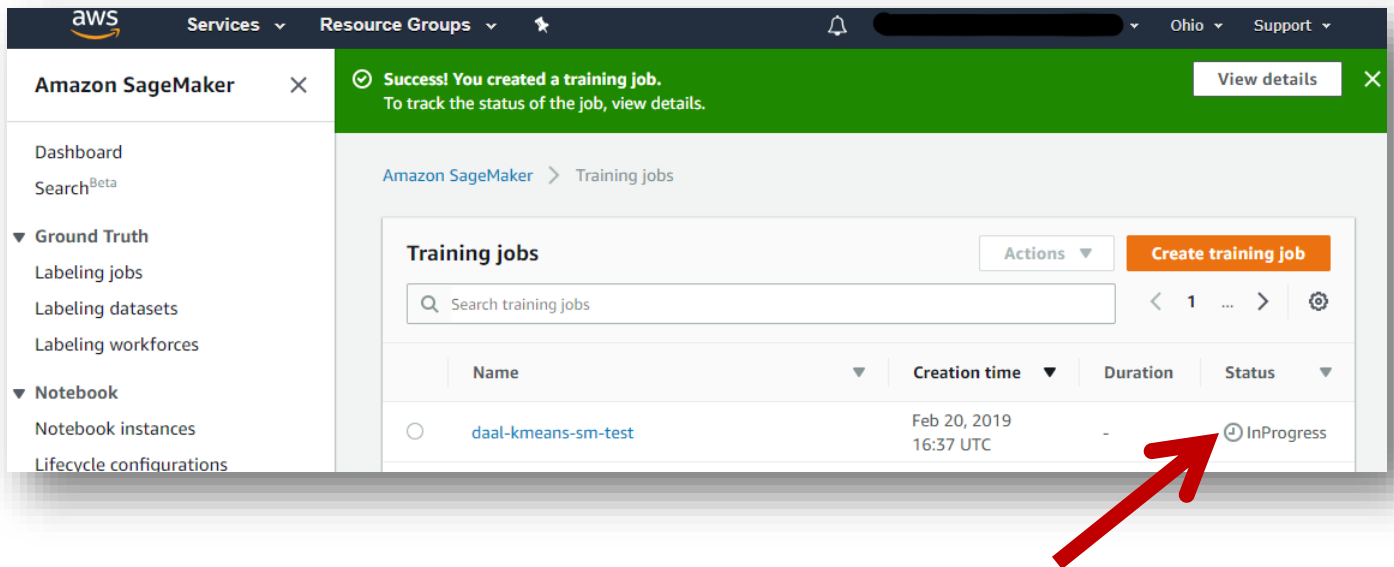
Remove

[Add tag](#)

Cancel

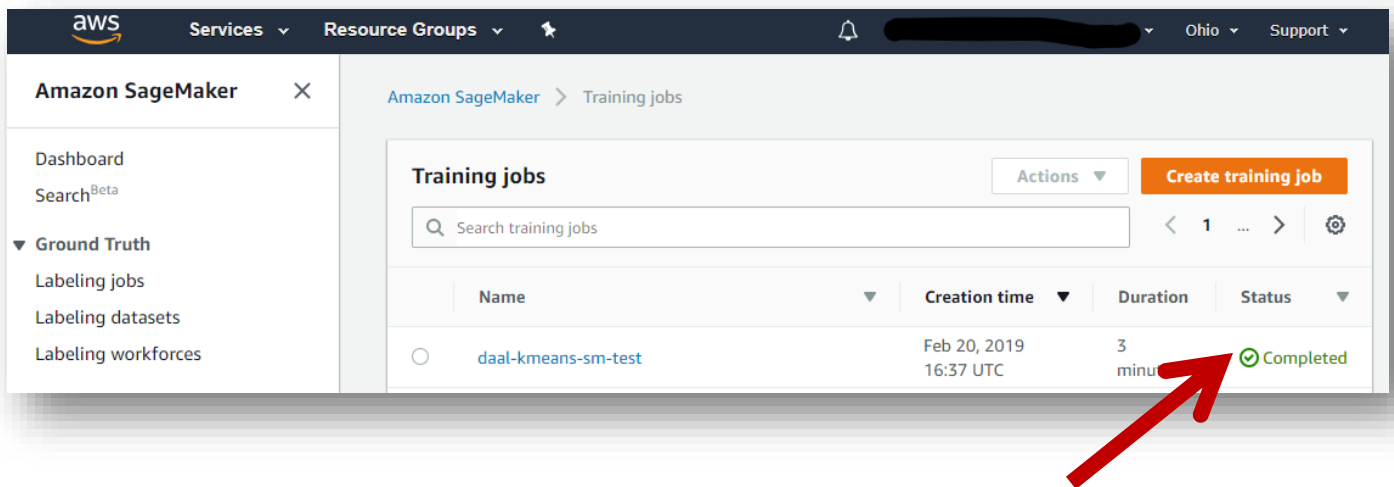
Create training job

## 10. Wait until finish of training job



The screenshot shows the Amazon SageMaker console with a green notification banner at the top stating "Success! You created a training job. To track the status of the job, view details." The left sidebar contains navigation links for Dashboard, Search, Ground Truth, and Notebook. The main content area displays the "Training jobs" table. A red arrow points to the "InProgress" status of the job named "daal-kmeans-sm-test".

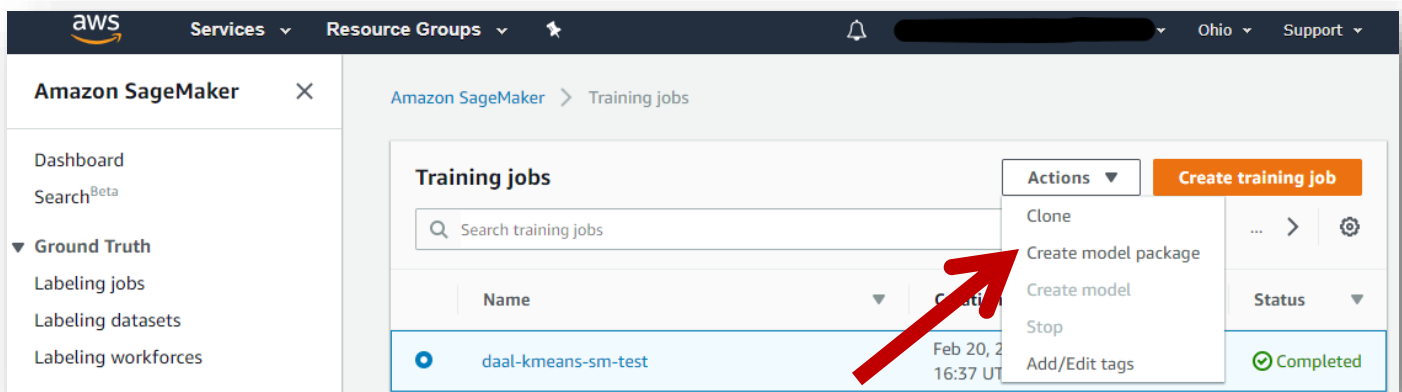
Name	Creation time	Duration	Status
daal-kmeans-sm-test	Feb 20, 2019 16:37 UTC	-	InProgress



The screenshot shows the Amazon SageMaker console with the "Training jobs" table. A red arrow points to the "Completed" status of the job named "daal-kmeans-sm-test".

Name	Creation time	Duration	Status
daal-kmeans-sm-test	Feb 20, 2019 16:37 UTC	3 min	Completed

## 11. Select training job and take action "Create model package"



The screenshot shows the Amazon SageMaker console with the "Training jobs" table. A red arrow points to the "Create model package" action in the dropdown menu for the job named "daal-kmeans-sm-test".

Name	Creation time	Status
daal-kmeans-sm-test	Feb 20, 2019 16:37 UTC	Completed

- Clone
- Create model package
- Create model
- Stop
- Add/Edit tags

12. Type model package name and click "Next"

## Create model package

### Inference specifications

#### Model package name and description

Model package name



The model package name must be unique in your account and in the AWS Region and can have up to 63 characters. Valid characters: a-z, A-Z, 0-9, and - (hyphen)

Description - *optional*

The description can be up to 1024 characters.

#### Inference specification options

- ☐ Provide the location of the inference image and model artifacts  
Choose this option if your model was trained using an algorithm stored in ECR.
- ☒ Provide the algorithm used for training and its model artifacts  
Choose this option if you are using a model trained by an algorithm resource or subscription algorithm from AWS Marketplace.

#### Algorithm and model artifacts

Algorithm ARN

Enter the Amazon Resource Name (ARN) used to create the training job and model artifacts.

arn:aws:sagemaker:us-east-2-██████████:algorithm/intel-daal-kmeans-60222020

Location of model artifacts - *optional*

If you want buyers to use the model artifacts from a specific model, enter the path to the S3 bucket where they are stored.

s3://daal-validation/kmeans/outputs/daal-kmeans-sm-test/output/model.tar.gz

To find a path, [go to Amazon S3](#)

Cancel **Next**



### 13. Click "Create model package"

Step 1  
Inference specifications

Step 2  
Validation specifications

## Create model package

### Validation specifications

To list your model package on AWS Marketplace, you must have it validated by Amazon SageMaker. Provide the information that Amazon SageMaker needs to run transform jobs to validate your product.

#### Validation and scanning

Publish this model package on AWS Marketplace  
AWS SageMaker requires successful validation before you can publish to AWS Marketplace.

☐ Yes ☒ No

Validate this resource  
AWS SageMaker will create a training job and/or transform job based on your validation profiles below.

☐ Yes ☒ No

Cancel Previous **Create model package**

### 14. Wait until package is created

Amazon SageMaker

Dashboard  
Search<sup>Beta</sup>

- Ground Truth
  - Labeling jobs
  - Labeling datasets
  - Labeling workforces
- Notebook
  - Notebook instances
  - Lifecycle configurations
  - Git repositories
- Training
  - Algorithms

**New model package created successfully.**

Amazon SageMaker > My model packages

My model packages | AWS Marketplace subscriptions

#### Model packages

Actions Find model packages Create model package

Search model packages

	Name	Status	ARN	Creation time
<input type="radio"/>	daal-kmeans-sm-test	Completed	arn:aws:sagemaker:us-east-2:[redacted]:model-package/daal-kmeans-sm-test	Feb 20, 2019 16:48 UTC

## 15. Select package and take action "Create endpoint"

The screenshot shows the Amazon SageMaker console interface. On the left is a navigation sidebar with options like Dashboard, Search, Ground Truth, and Notebook. The main content area is titled 'My model packages' and shows a table of model packages. One package, 'daal-kmeans-sm-test', is selected. An 'Actions' dropdown menu is open for this package, showing options: Clone model package, Create model, Create endpoint, Publish new ML Marketplace listing, Delete, and Add/edit tags. A red arrow points to the 'Create endpoint' option.

Name	Creation time
daal-kmeans-sm-test	Feb 20, 2019 16:48 UTC

## 16. Type model name and click "Next"

The screenshot shows the 'Create model and endpoint' wizard in the Amazon SageMaker console. The wizard has two steps: 'Step 1: Create model' and 'Step 2: Create endpoint'. The 'Model settings' section is visible, with a 'Model name' input field containing 'daal-kmeans-sm-test'. A red arrow points to this input field. Below the input field, there is a description: 'Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.' Below this, there is an 'IAM role' section with a dropdown menu showing 'AmazonSageMaker-ExecutionRole-...'.

17. Type endpoint name, edit and create endpoint configuration and click "Submit"

Model was successfully updated. Continue to create an endpoint.

Step 1  
[Create model](#)

Step 2  
**Create endpoint**

Amazon SageMaker > Using model package "daal-kmeans-sm-test"

### Create model and endpoint

**Endpoint**

Endpoint name  
Your application uses this name to access this endpoint.

Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.

**Attach endpoint configuration**

☐ Use an existing endpoint configuration  
Use an existing endpoint configuration or clone an endpoint configuration.

☒ Create a new endpoint configuration  
Add models and configure the instance and initial weight for each model.

New endpoint configuration

To deploy models to Amazon SageMaker, first create an endpoint configuration. In the configuration, specify which models to deploy, and the relative traffic weighting and hardware requirements for each.

Endpoint configuration name  
endpointConfig-[redacted]

Maximum of 63 alphanumeric characters. Can include hyphens (-), but not spaces. Must be unique within your account in an AWS Region.

Encryption key - *optional*  
Encrypt your data. Choose an existing KMS key or enter a key's ARN.

No Custom Encryption

**Production variants**

Model name	Training job	Variant name	Instance type	Elastic Inference	Initial instance count	Initial weight	Actions
daal-kmeans-sm-test		variant-name-1	ml.m4.xlarge	none	1	1	<a href="#">Edit</a>   <a href="#">Remove</a>

[Add model](#)

[Create endpoint configuration](#)

**Tags - optional**

Key	Value	
		<a href="#">Remove</a>

[Add tag](#)

Cancel [Previous](#) [Submit](#)

## 18. Wait until endpoint is ready

Amazon SageMaker

Success! You created an endpoint.  
To track the status of the endpoint, view details.

Amazon SageMaker > Endpoints

Endpoints

Update endpoint Actions Create endpoint

Search endpoints

	Name	ARN	Creation time	Status	Last updated
<input type="radio"/>	daal-kmeans-sm-test	arn:aws:sagemaker:us-east-2:123456789012:endpoint/daal-kmeans-sm-test	Feb 21, 2019 12:36 UTC	Creating	Feb 21, 2019 12:36 UTC
<input type="radio"/>	daal-knn-sm-test	arn:aws:sagemaker:us-east-2:123456789012:endpoint/daal-knn-sm-test	Feb 21, 2019 12:36 UTC	InService	Feb 21, 2019 12:42 UTC

## 19. Use AWS CLI to get real-time prediction

Type command:

```
aws sagemaker-runtime invoke-endpoint --endpoint-name <endpoint-name> --body "$(cat <prediction_data_file_name>)" --content-type text/csv --accept text/csv <output_data_file_name>
```

```
(base) ubuntu@ip-172-31-22-46:~$ aws sagemaker-runtime invoke-endpoint --endpoint-name daal-knn-sm-test --body "$(cat probe_data.csv)" --content-type text/csv --accept text/csv output.txt
```

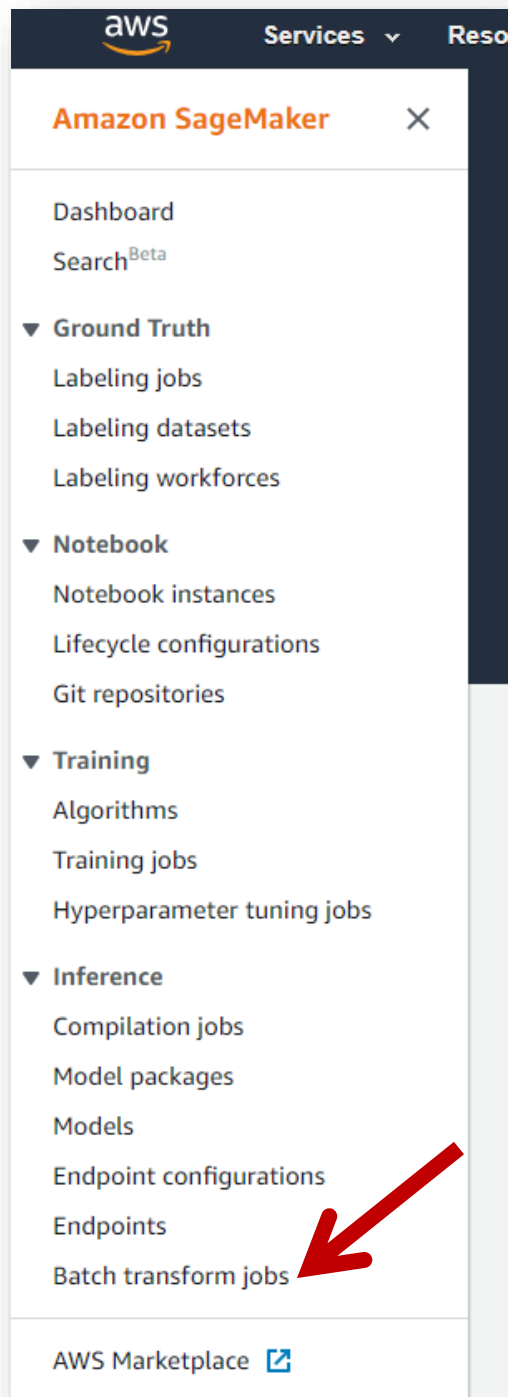
Then, see content of output file:

```
(base) ubuntu@ip-172-31-22-46:~$ cat output.txt
1
4
0
2
3
4
1
1
0
2
4
```

## Batch transform job as alternative to endpoint

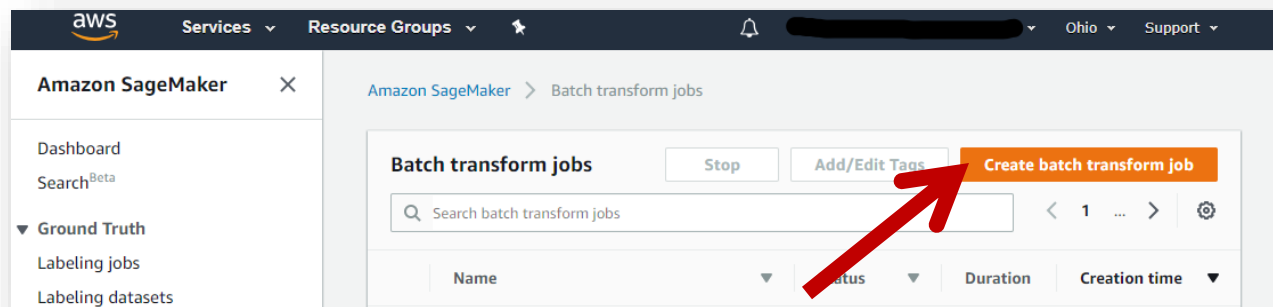
You can use batch transform job if you need compute predictions once.

1. Go to “Batch transform job” page

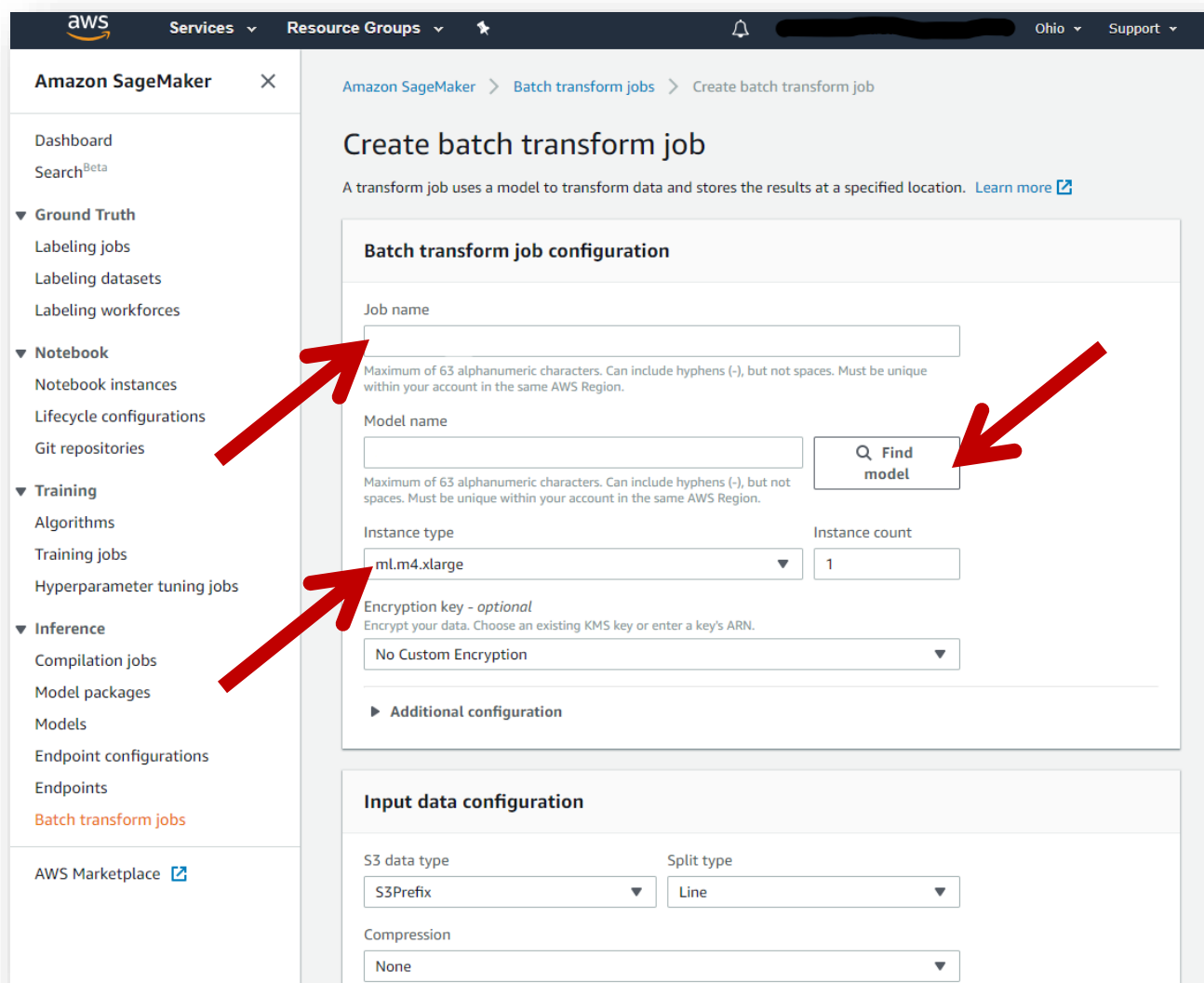




2. Click "Create batch transform job"



3. Enter job name, select previously created model, instance and set instance count to 1



- Specify S3 location of data for prediction, S3 output path (predictions will be stored here) and click "Create job"

**Amazon SageMaker**

Services ▾ Resource Groups ▾ Ohio ▾ Support ▾

Dashboard  
Search Beta

▼ **Ground Truth**  
Labeling jobs  
Labeling datasets  
Labeling workforces

▼ **Notebook**  
Notebook instances  
Lifecycle configurations  
Git repositories

▼ **Training**  
Algorithms  
Training jobs  
Hyperparameter tuning jobs

▼ **Inference**  
Compilation jobs  
Model packages  
Models  
Endpoint configurations  
Endpoints  
**Batch transform jobs**

AWS Marketplace [↗](#)

### Input data configuration

S3 data type:  Split type:

Compression:

Content type - *optional*:   
For content types that are available in built-in algorithms, [view our documentation](#)

S3 location:   
To find a path, [go to Amazon S3](#)

### Output data configuration

S3 output path:   
To find a path, [go to Amazon S3](#)

Encryption key - *optional*:   
An encryption key protects your data. Type the key ID or key ARN that you want to use.

Accept - *optional*:  Assemble with:

► **Tags - optional**

[Cancel](#) [Create job](#)

5. Wait until job is completed and find predictions in previously specified S3 output path

The screenshot shows the Amazon SageMaker console with a green notification banner at the top stating "Batch transform job daal-sm-test was successfully created". The left sidebar contains navigation links for Dashboard, Search<sup>Beta</sup>, Ground Truth, Labeling jobs, Labeling datasets, Labeling workforces, Notebook, and Notebook instances. The main content area displays "Batch transform jobs" with a search bar and a table. The table has columns for Name, Status, Duration, and Creation time. A single job, "daal-sm-test", is listed with a status of "InProgress", a duration of "a few seconds", and a creation time of "Feb 17, 2019 14:32 UTC". A red arrow points to the "InProgress" status.

Name	Status	Duration	Creation time
daal-sm-test	InProgress	a few seconds	Feb 17, 2019 14:32 UTC

The screenshot shows the same Amazon SageMaker console interface, but the status of the "daal-sm-test" job has changed to "Completed". The duration is now "3 minutes" and the creation time remains "Feb 17, 2019 14:32 UTC". A red arrow points to the "Completed" status.

Name	Status	Duration	Creation time
daal-sm-test	Completed	3 minutes	Feb 17, 2019 14:32 UTC