How to use Intel® DAAL PCA via SageMaker web interface

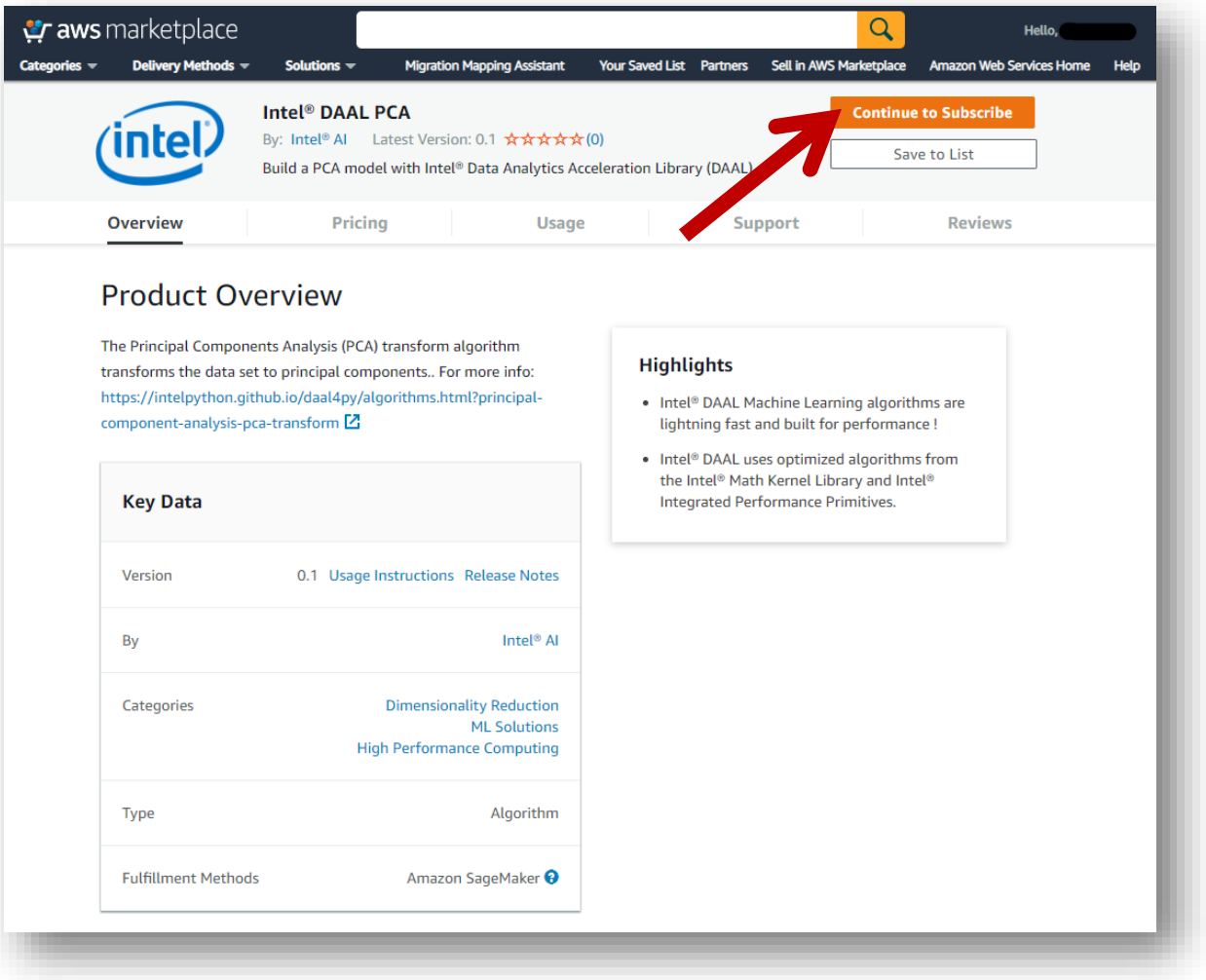
Description of algorithm:

Principal Component Analysis (PCA) is a method for exploratory data analysis. PCA transforms a set of observations of possibly correlated variables to a new set of uncorrelated variables, called principal components. Principal components are the directions of the largest variance, that is, the directions where the data is mostly spread out. Because all principal components are orthogonal to each other, there is no redundant information. This is a way of replacing a group of variables with a smaller set of new variables. PCA is one of powerful techniques for dimension reduction.

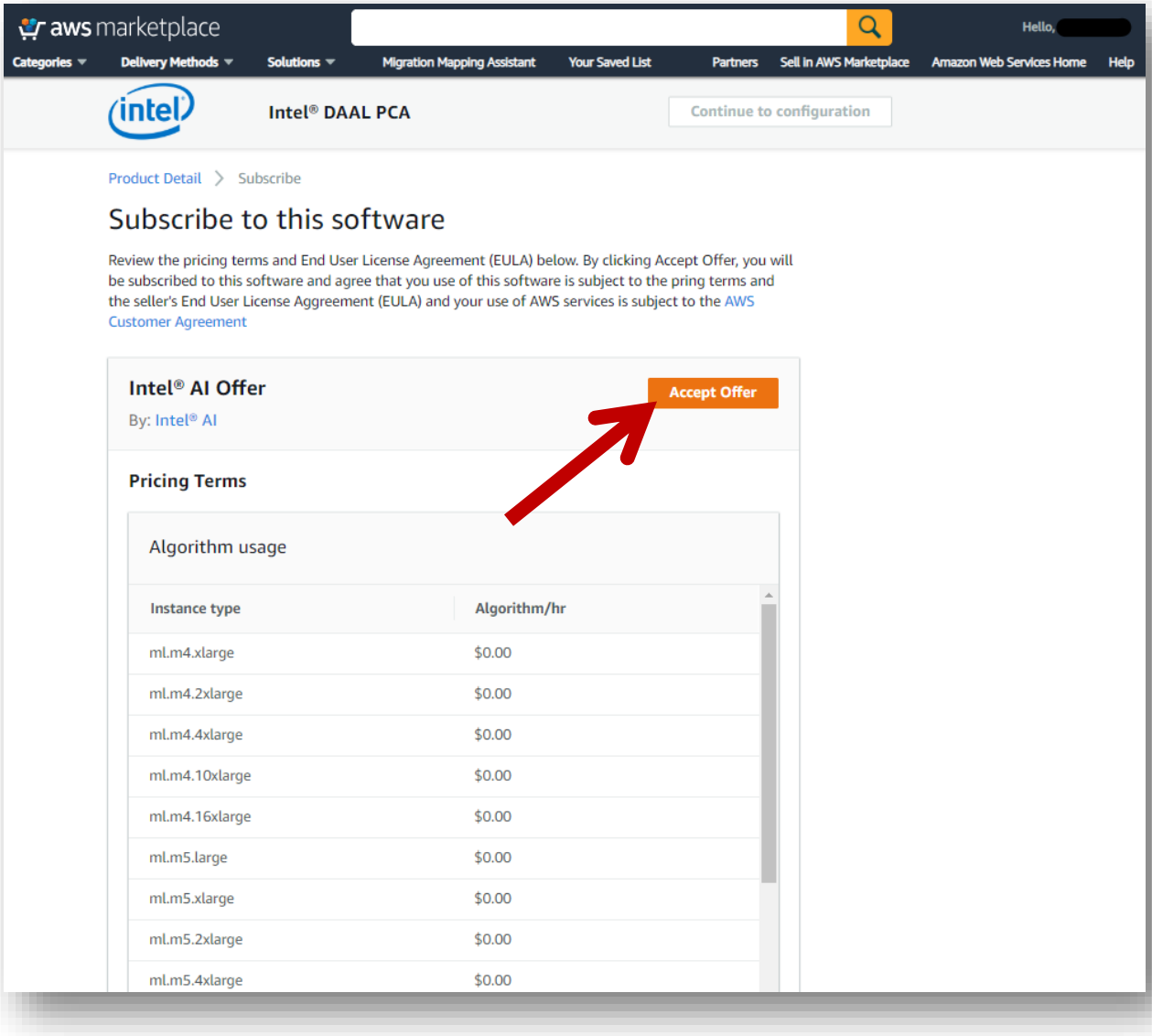
Intel® DAAL developer guide Intel® DAAL documentation for PCA

Instruction:

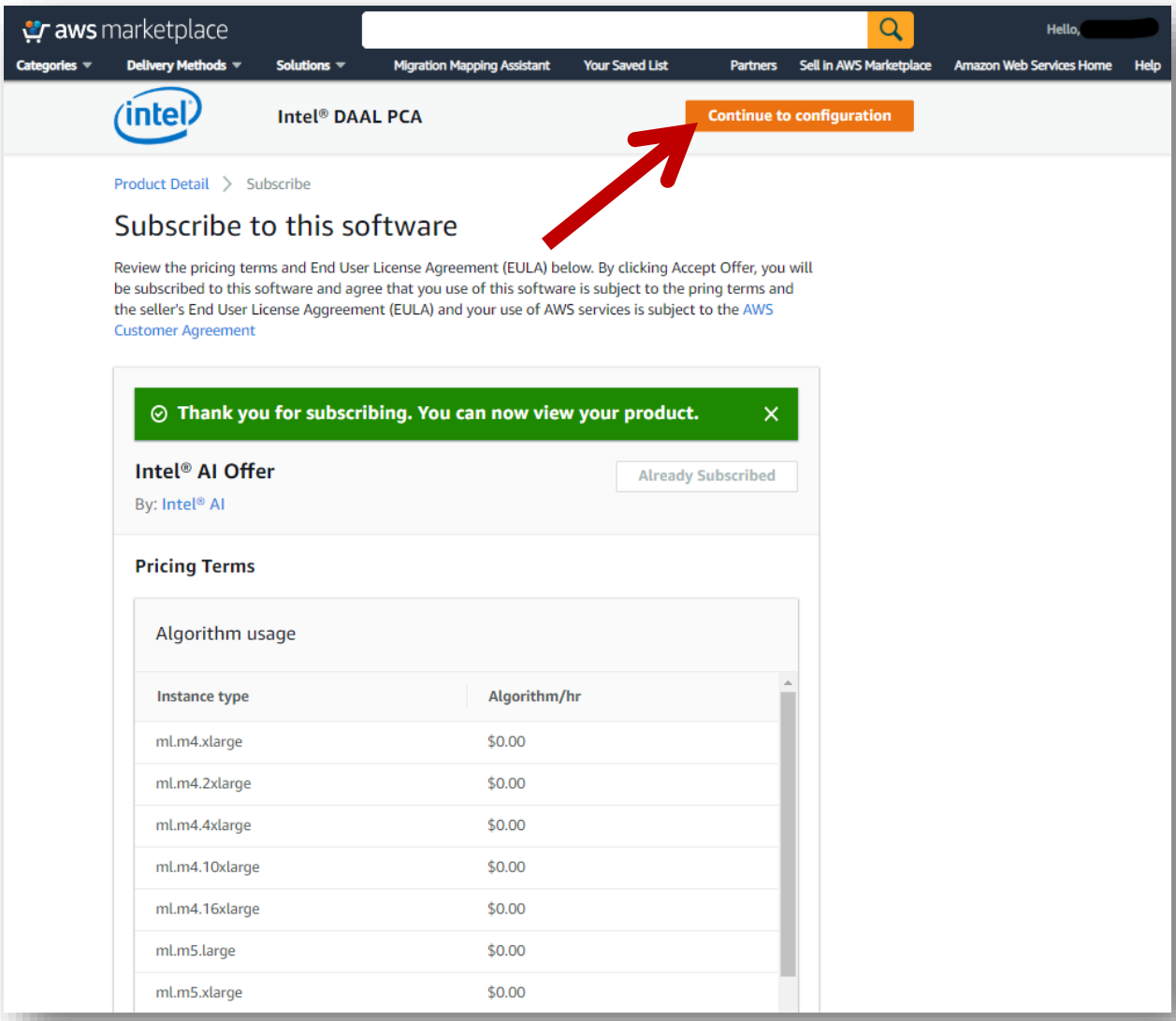
1. Visit page on SageMaker Marketplace and click “Continue to Subscribe”



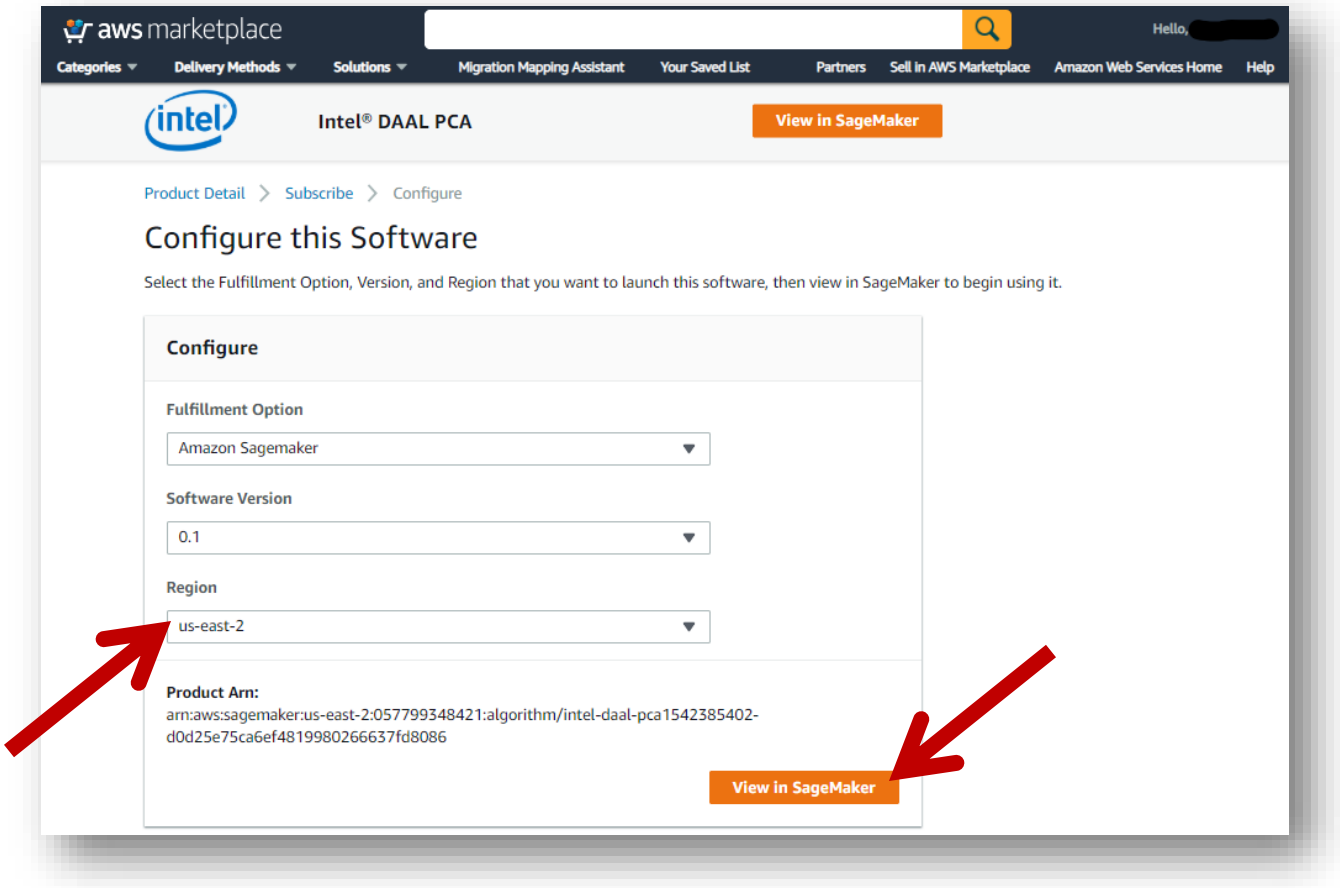
2. Click “Accept Offer” if you agree with EULA at end of page. If you already subscribed on algorithm on Marketplace this step will be skipped.



3. Click “Continue to configuration”



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



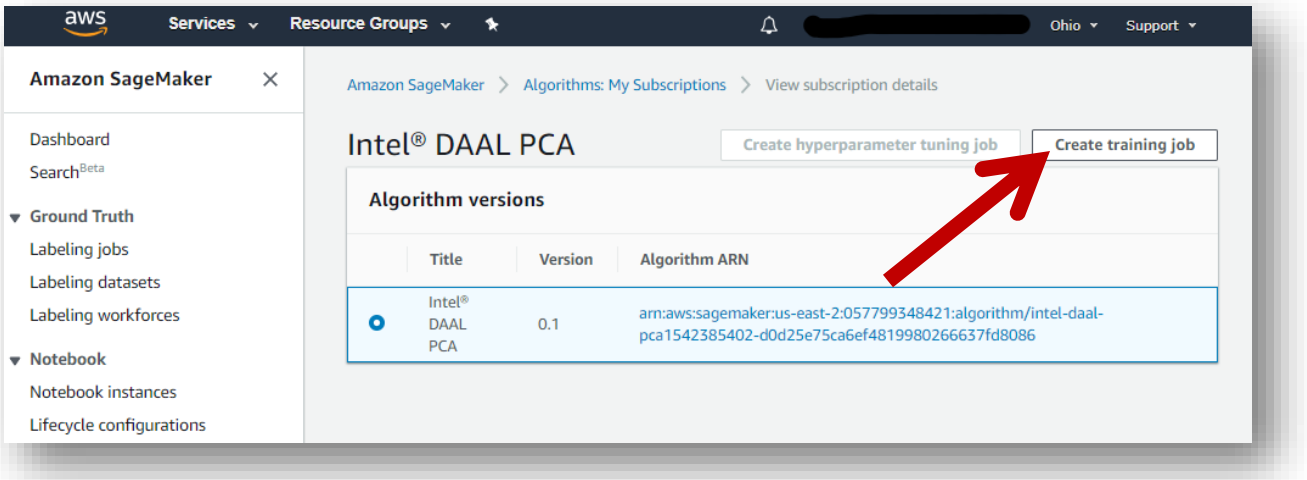
PCA algorithm on AWS SageMaker is divided into two stages: training job and getting inference from endpoint.

Training job is computing principal components and other values from provided training data.

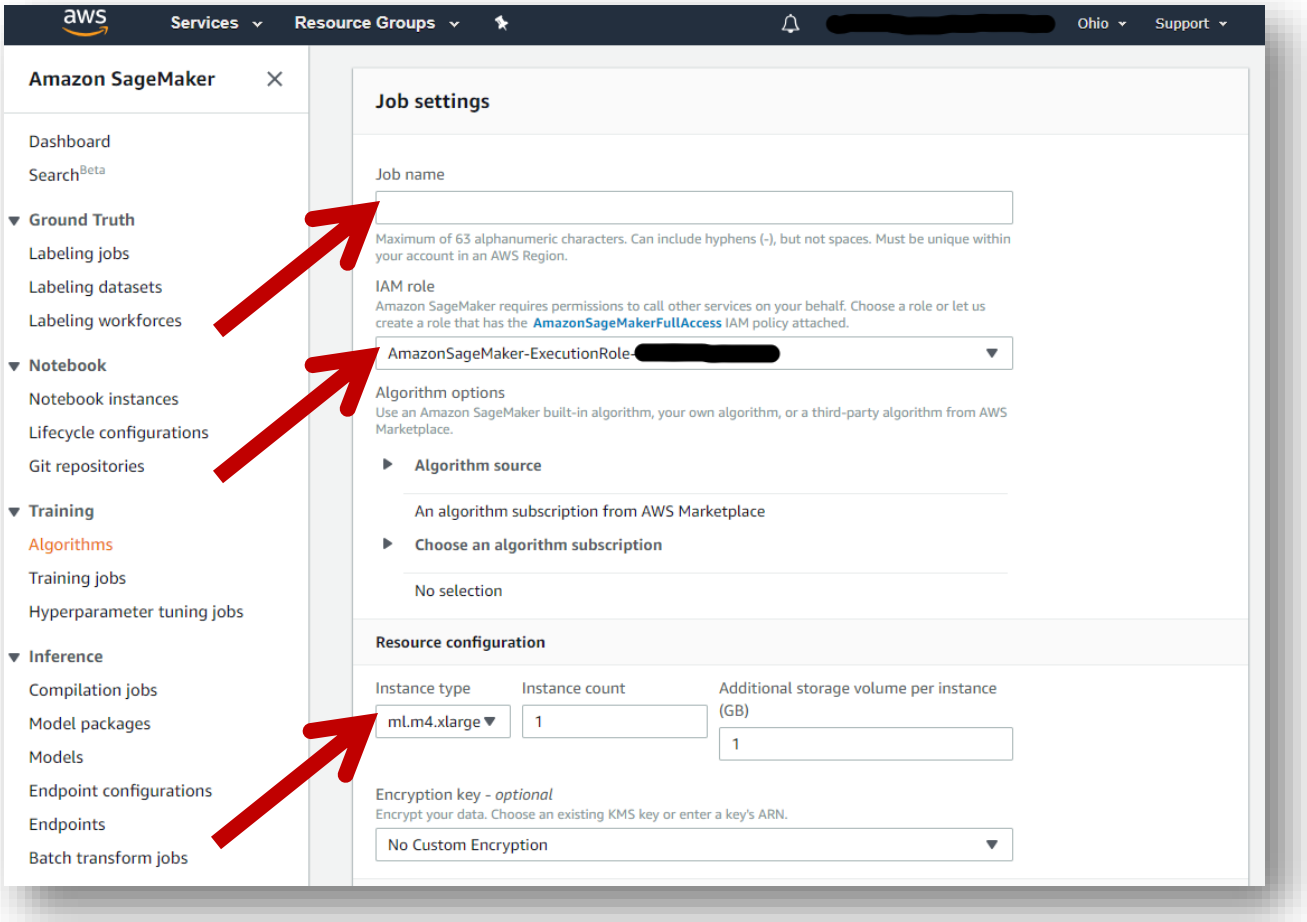
After that, you should create model with computed values and endpoint based on it.

Sending data to endpoint gives you transformed (dimension reduced) data in response.

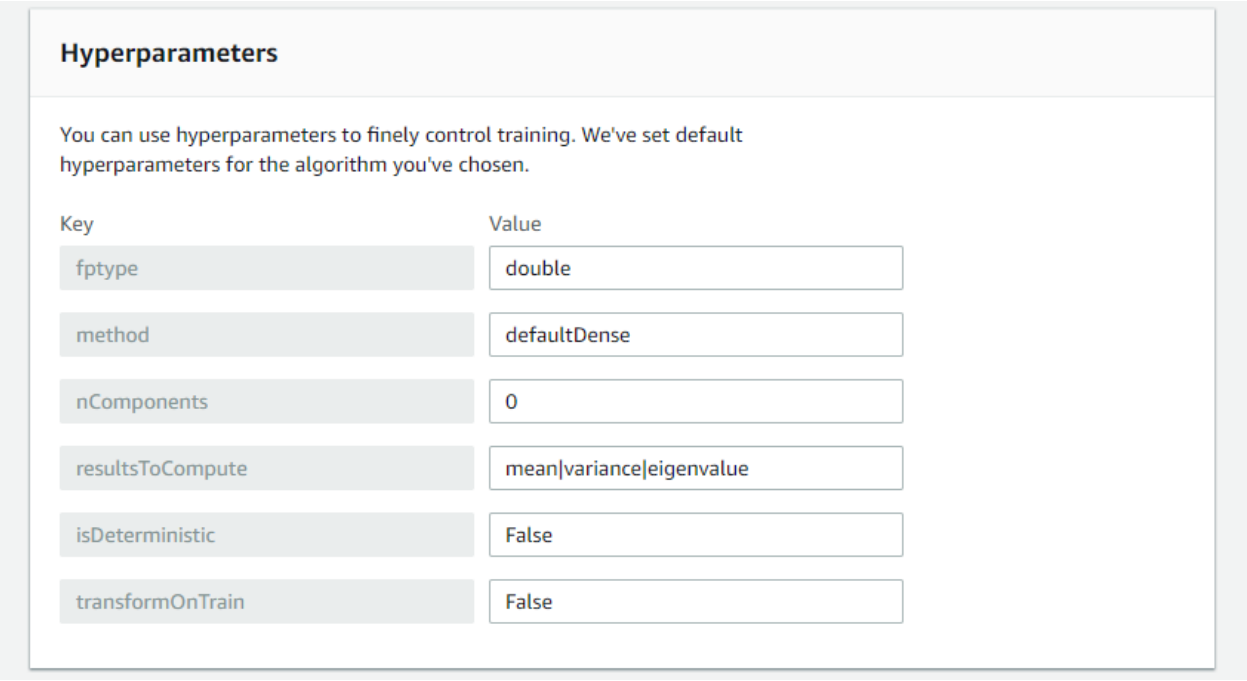
5. Select needed algorithm version and click “Create training job”



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

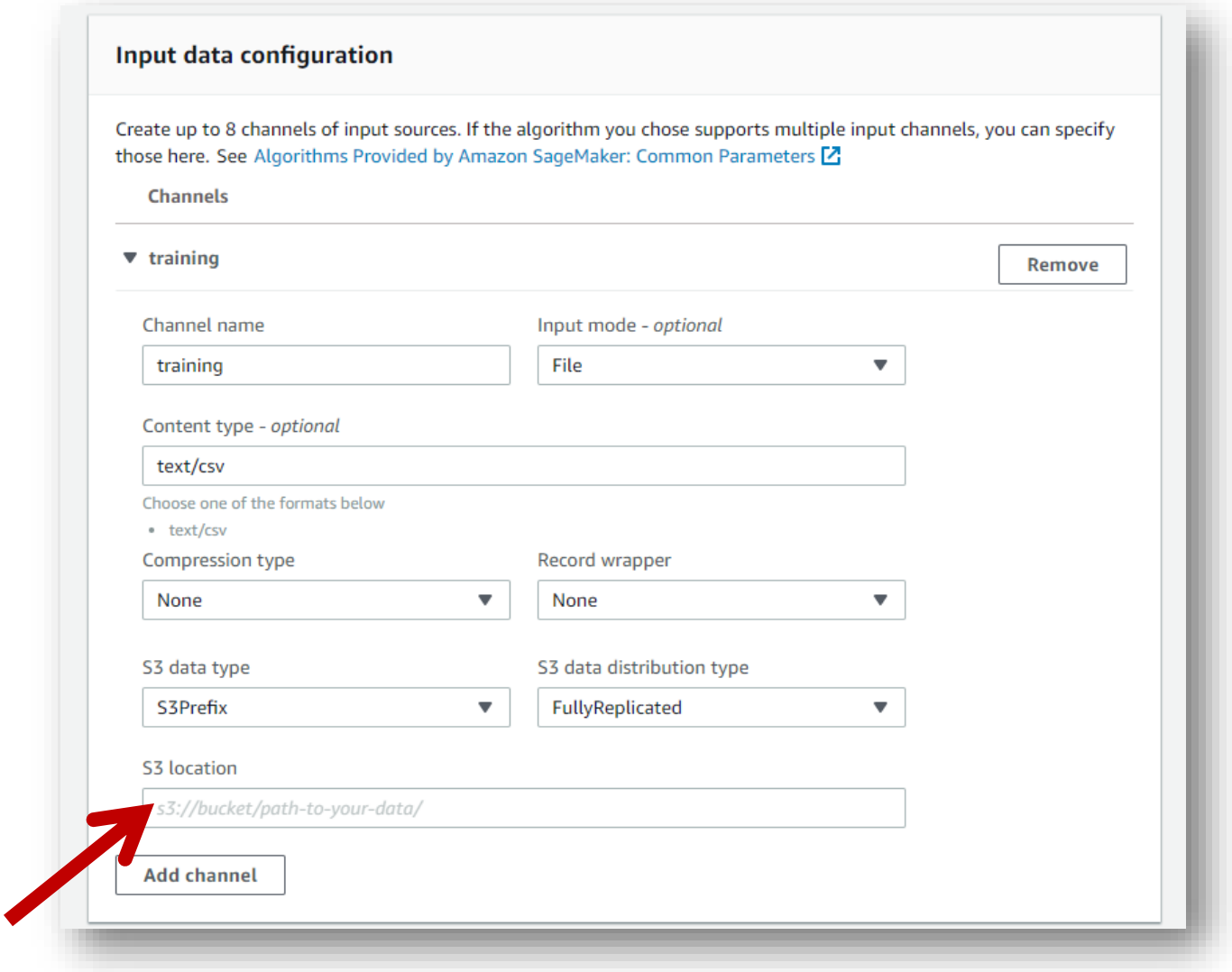


7. Choose hyperparameters

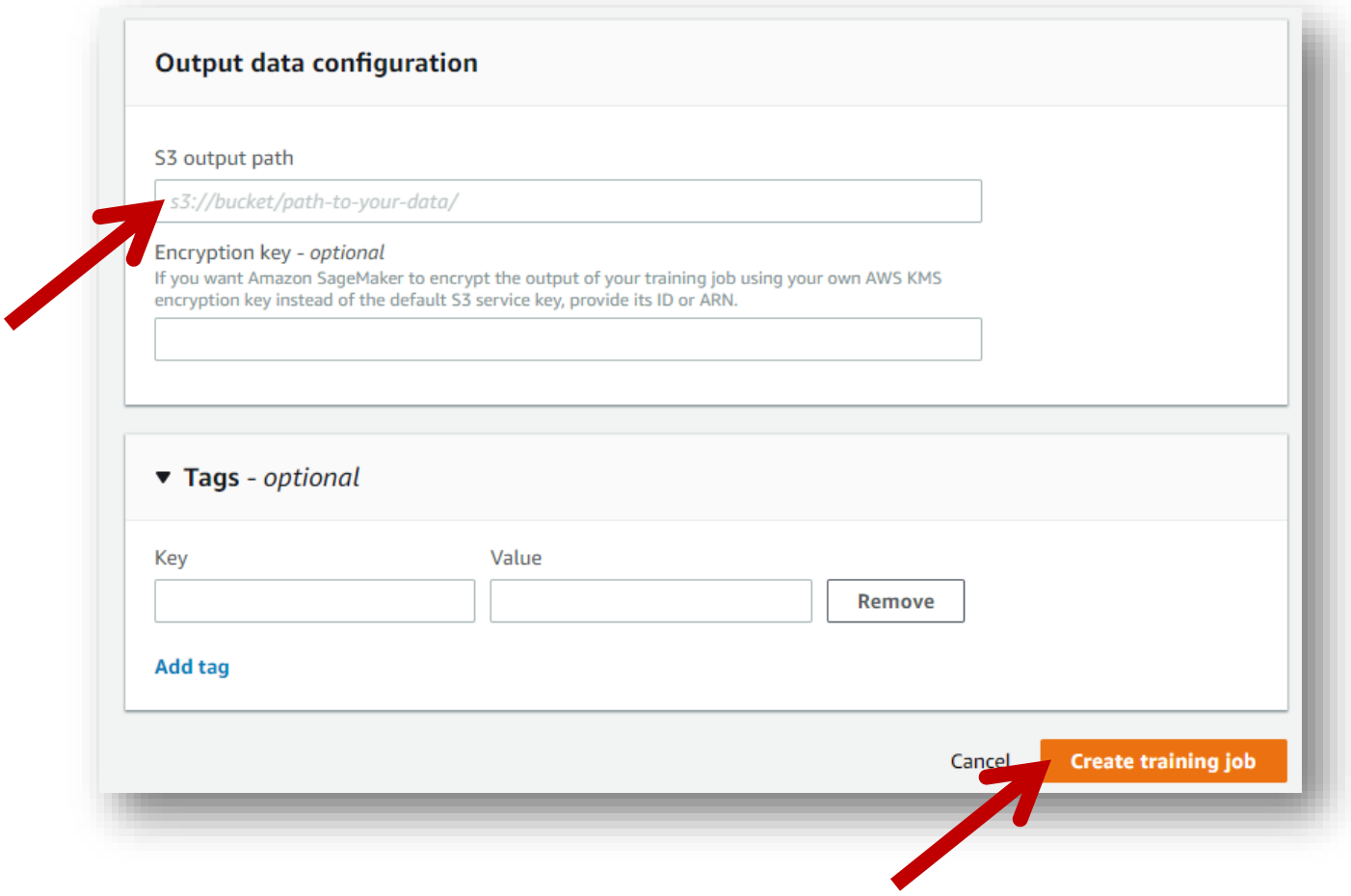


Parameter name	Type	Default value	Description
fptype	str	"double"	The floating-point type that the algorithm uses for intermediate computations. Can be "float" or "double"
method	str	"correlationDense"	Available methods for PCA computation: "correlationDense" ("defaultDense") or "svdDense"
resultsToCompute	str	"none"	Provide one of the following values to request a single characteristic or use bitwise OR to request a combination of the characteristics: mean, variance, eigenvalue. For example, combination of all is "mean|variance|eigenvalue"
nComponents	int	0	Number of principal components. If it is zero, the algorithm will compute the result for number of principal components = number of features. Remember that number of components must be equal or less than number of features for PCA algorithm
isDeterministic	bool	False	If True, the algorithm applies the "sign flip" technique to the results
transformOnTrain	bool	False	If True, training data will be transformed and saved in model package on training stage

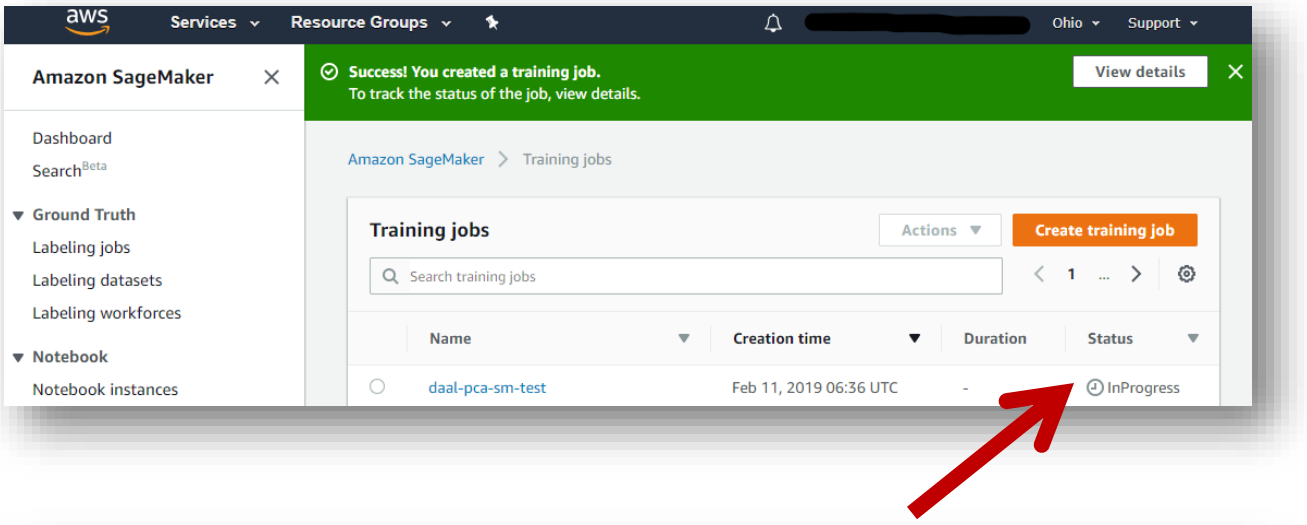
8. Specify S3 location of input data for training

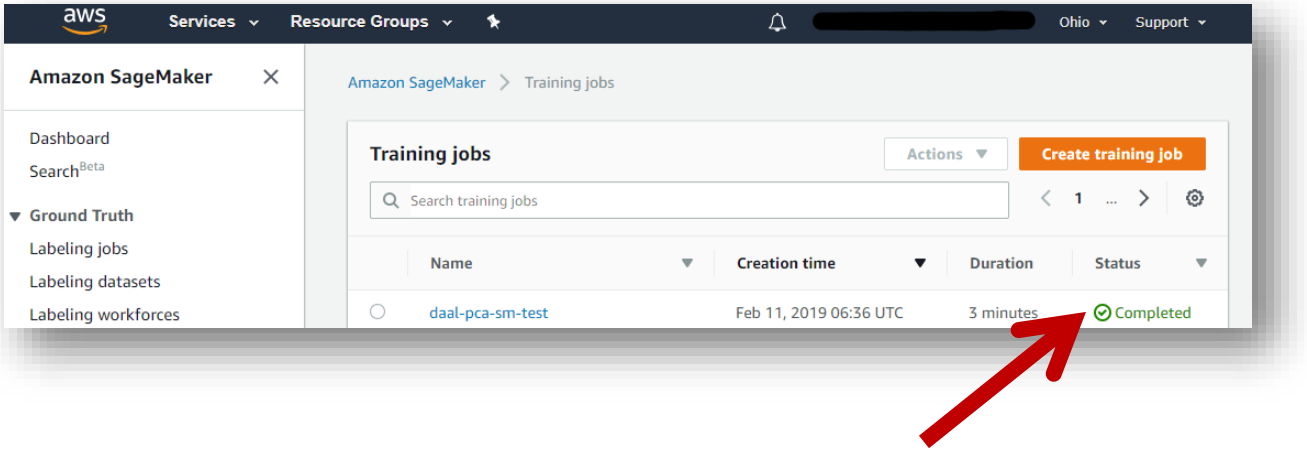


9. Specify S3 output path (model will be stored here) and click “Create training job”

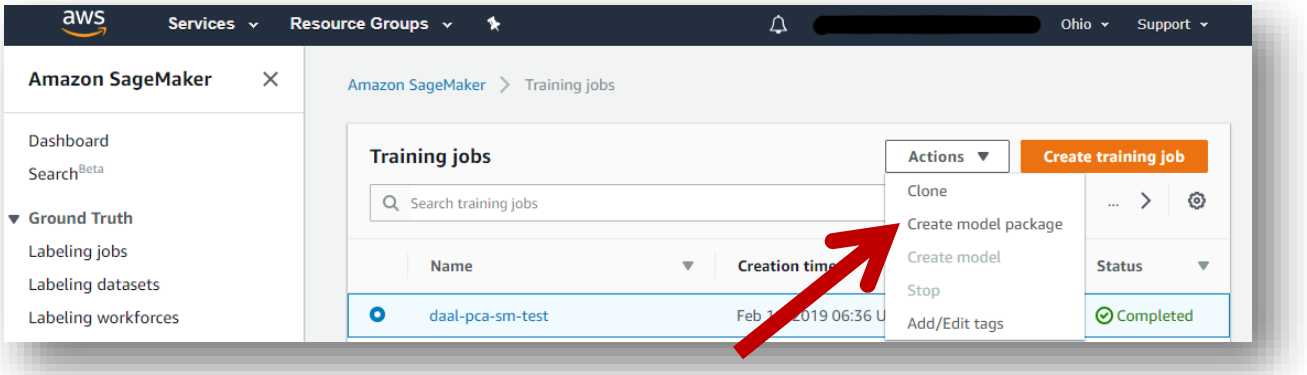


10. Wait until finish of training job

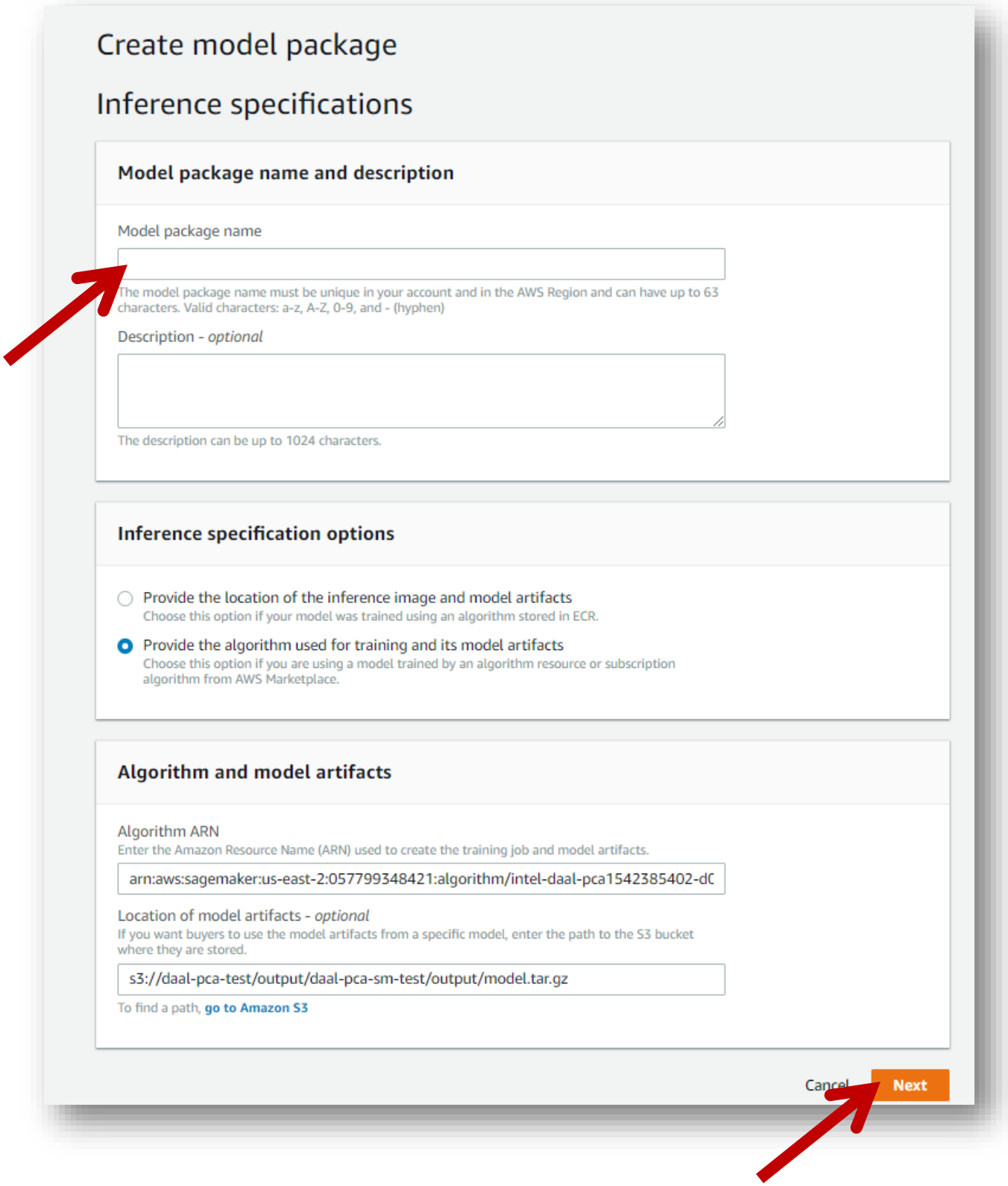




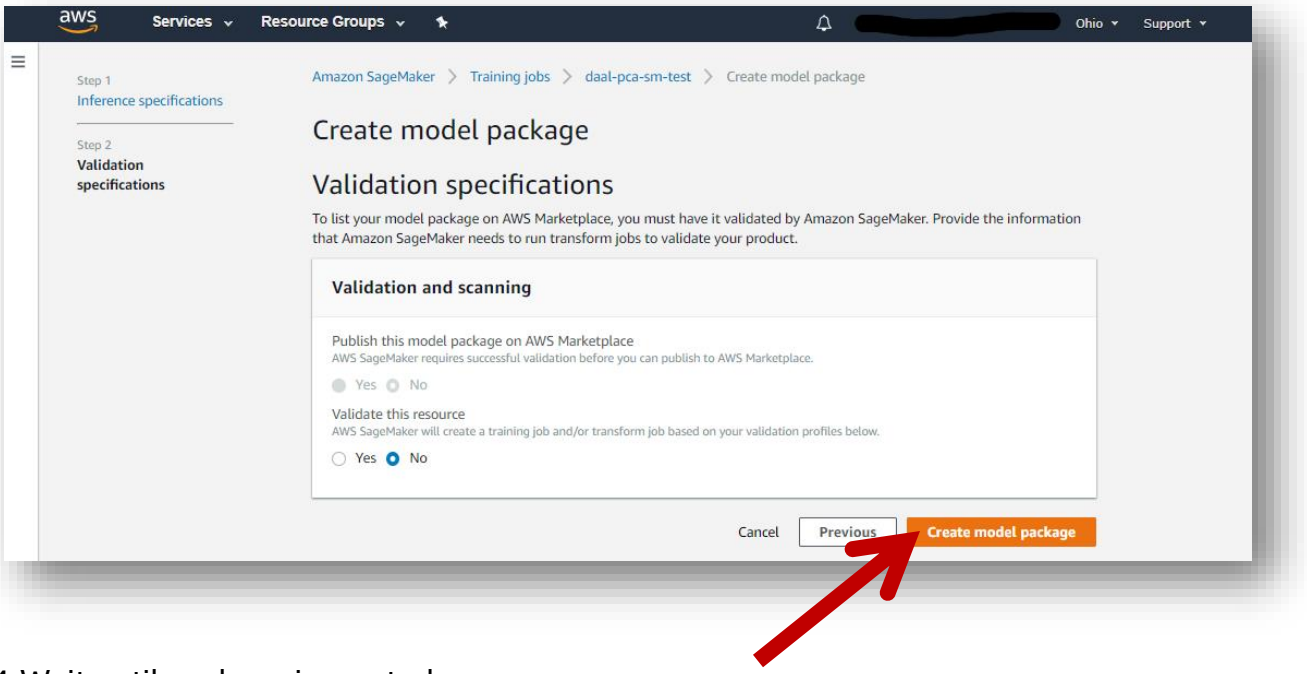
11. Select training job and take action “Create model package”



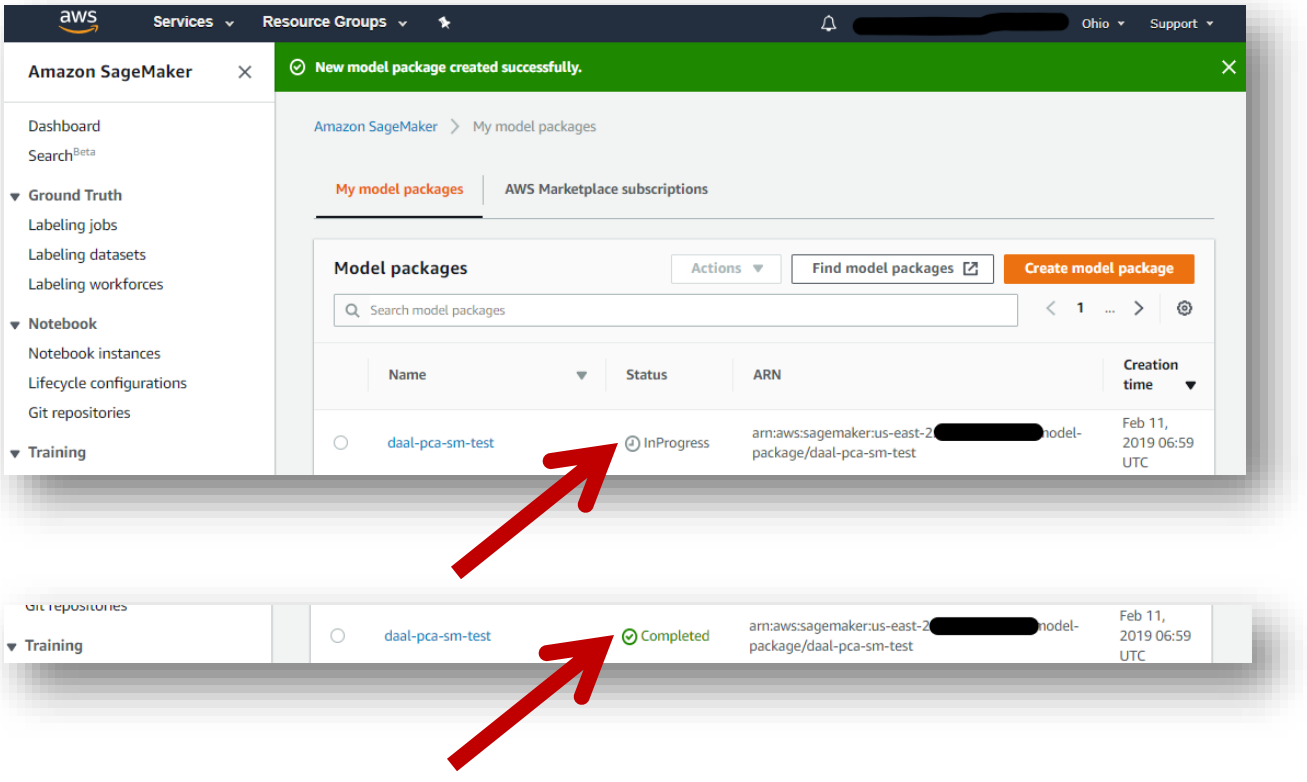
12. Type model package name and click “Next”



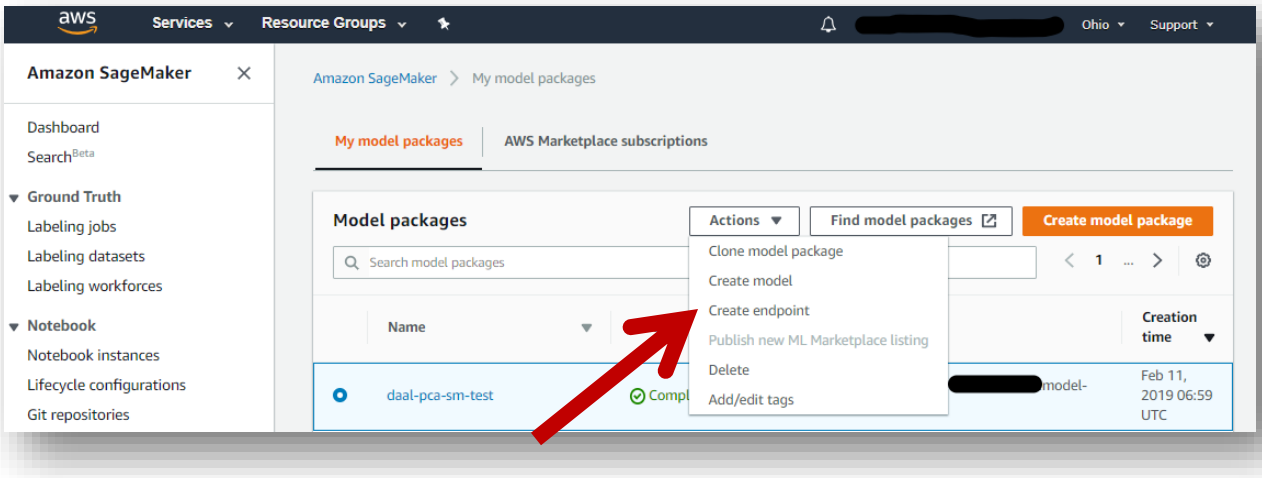
13. Click “Create model package”



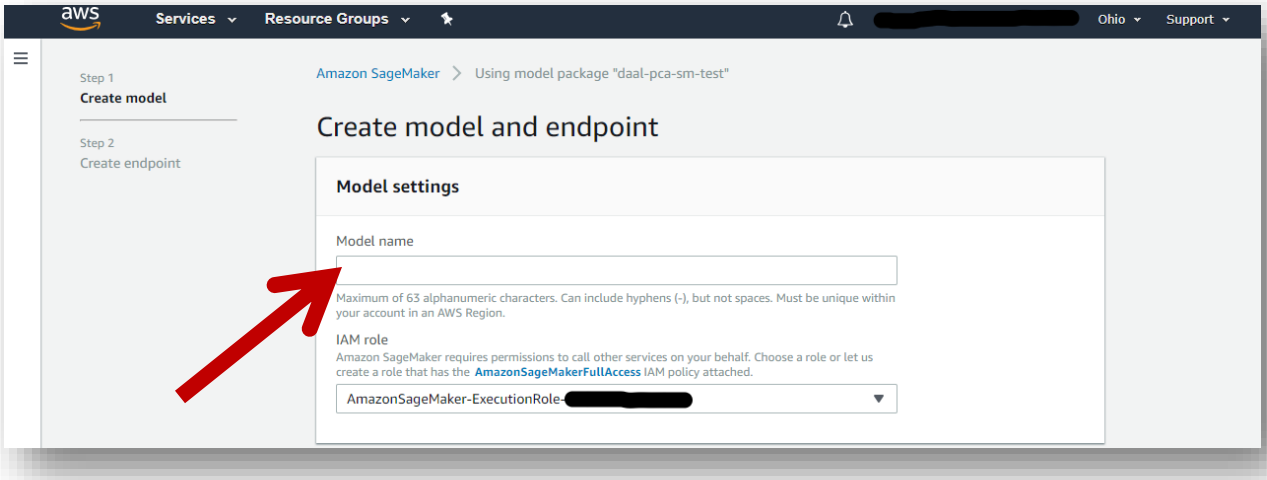
14. Wait until package is created



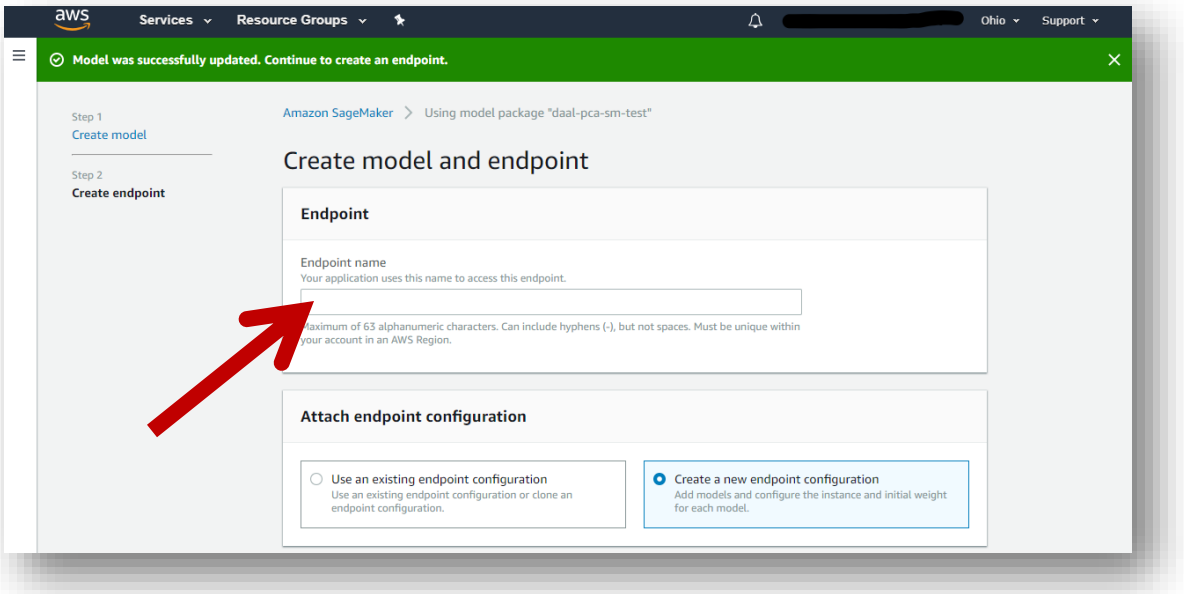
15. Select package and take action “Create endpoint”

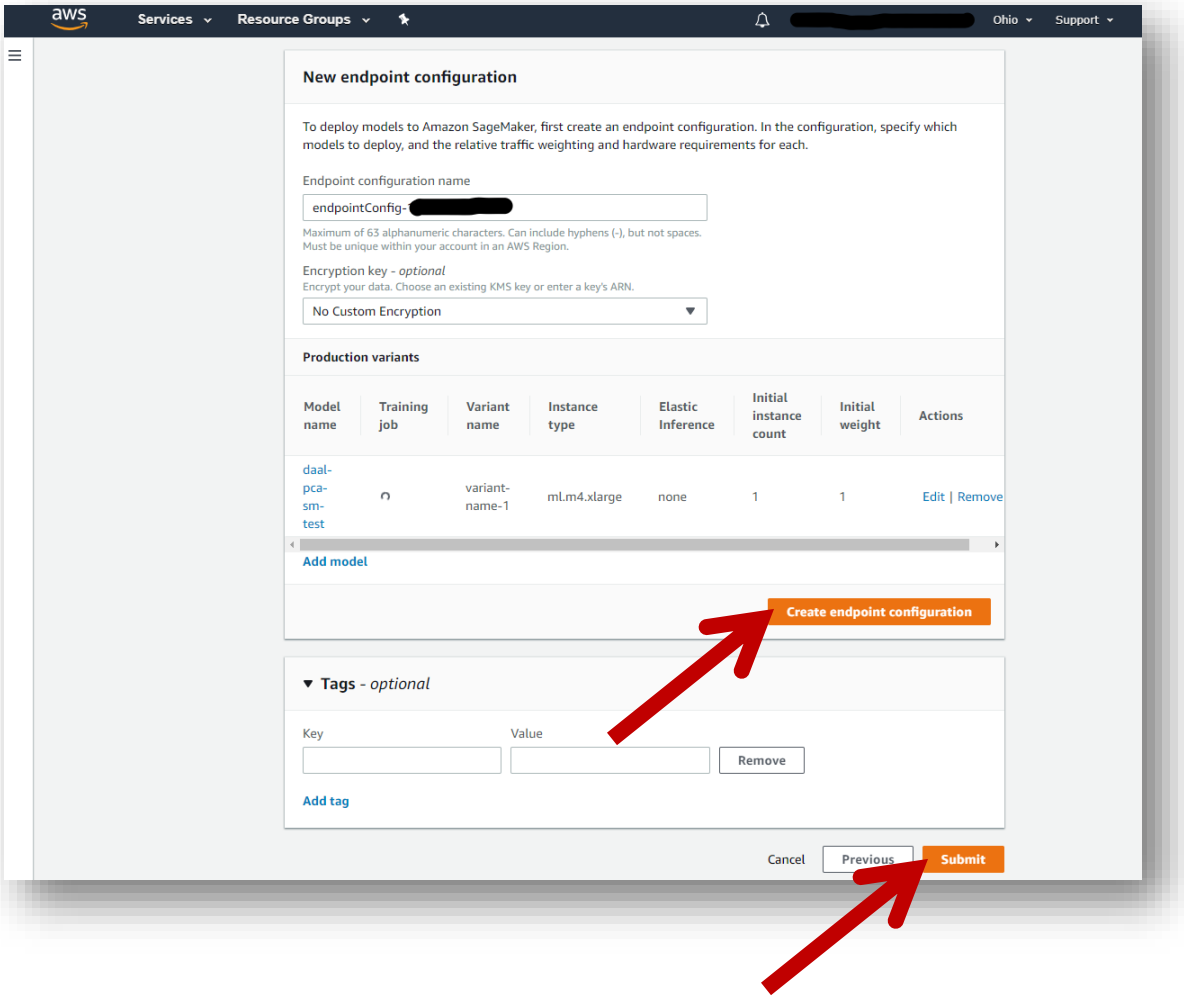


16. Type model name and click “Next”

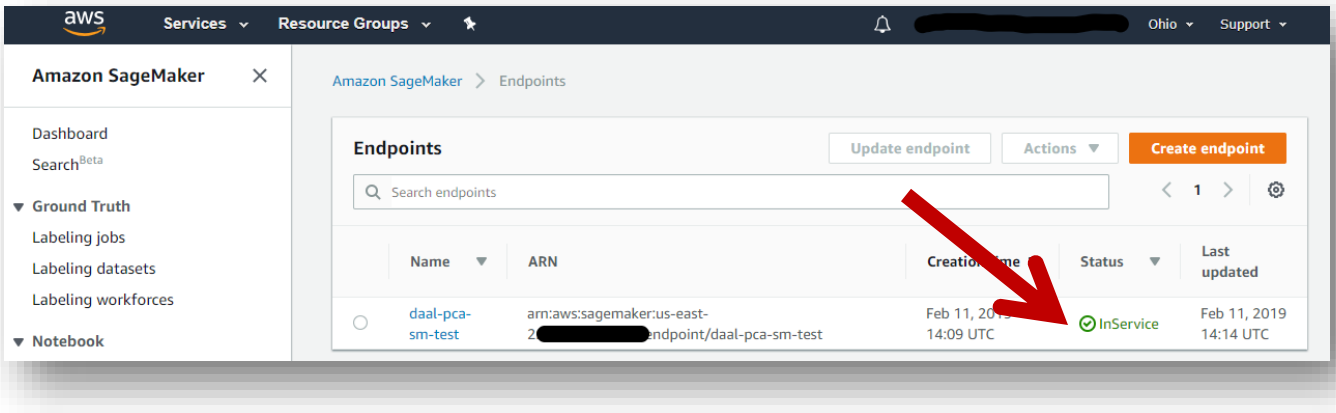


17. Type endpoint name, edit and create endpoint configuration and click “Submit”





18. Wait until endpoint is ready



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

NOTE: training data and data for transformation should have same number of columns

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-172-31-11-84:~$ aws sagemaker-runtime invoke-endpoint --endpoint-name daal-pca-sm-test
--body "$(cat probe_data.csv)" --content-type text/csv --accept text/csv output.txt
```

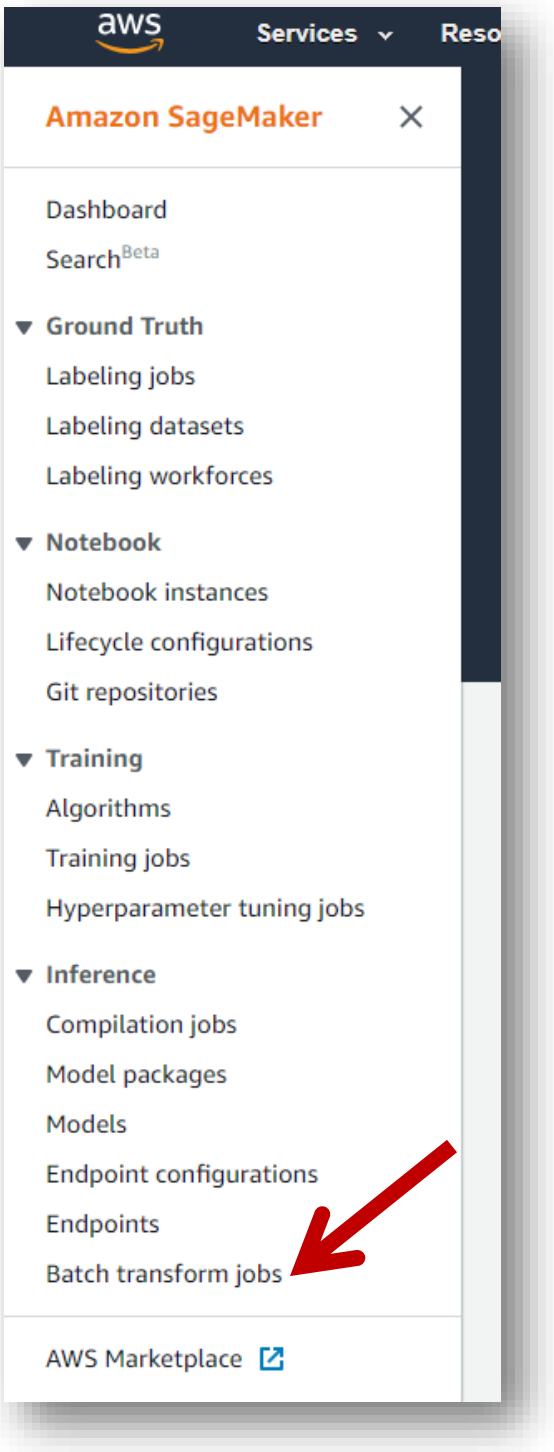
Then, see content of output file:

```
(base) ubuntu@ip-172-31-11-84:~$ cat output.txt
2.5424888493446657 91e+00 -2.8549636629255958 91e+00 -2.2711279521589324 84e+00 -4.3349025911318683 77e+00 1
.7583313877117574 54e+00 -7.7876024530941982 96e+00 4.0884462775354929 45e-02 6.7649529633228535 63e+00 -2.0
6842835665674185 1e+00 -2.1208139870194142 60e+00
-4.2316281926524927 93e+00 4.5134667849533620 65e+00 -5.7952193354050729 47e+00 5.9128213510181240 05e+00 -2
.5299646370790891 40e+00 -1.0065286974685621 50e+00 2.5965215515253357 03e+00 6.1077734748531229 41e+00 1.55
6518408950566812e+00 -5.840441330756375393e+00
3.5996203070043568 67e+00 -3.2838621237514109 55e+00 1.9216316019398489 43e+00 6.4521013413142132 01e+00 2.9
3644071801353490 1e-01 -2.1522312380561237 08e+00 2.8401166378126330 16e+00 2.0003042937801995 83e+00 1.8024
98395592655678e+00 -1.459109835559649682e+00
```

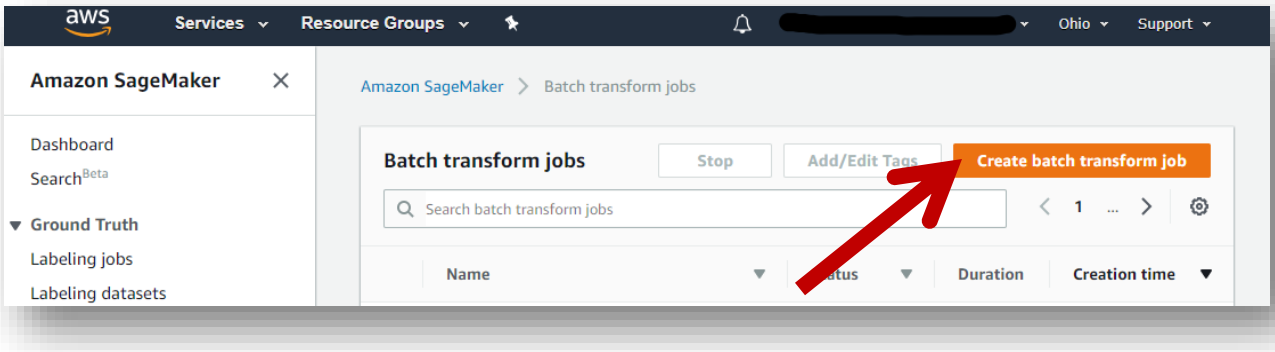
Batch transform job as alternative to endpoint

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

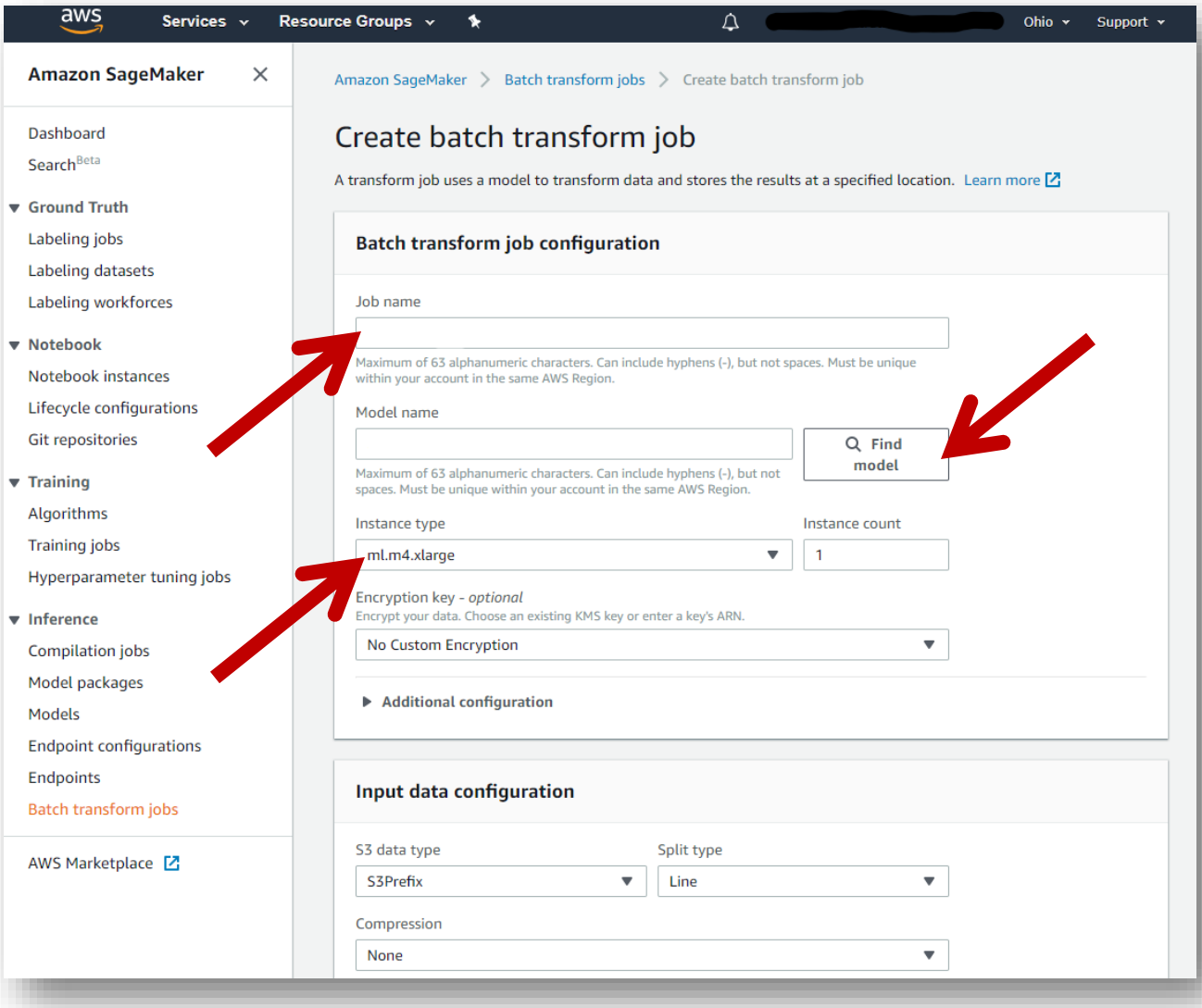
1. Go to “Batch transform job” page



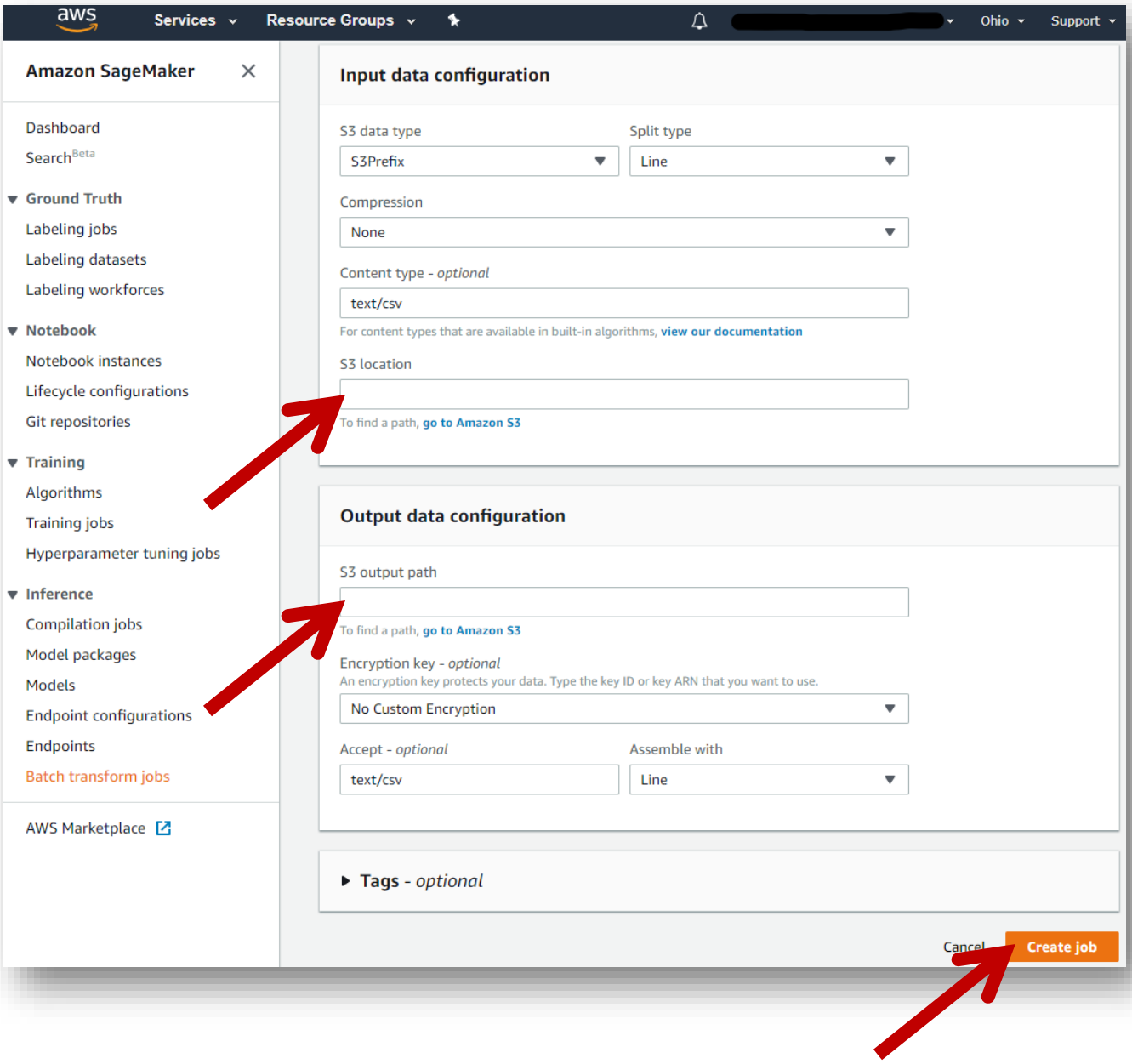
2. Click “Create batch transform job”



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



4. Specify S3 location of input data for transformation, S3 output path (transformed data will be stored here) and click “Create job”



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