

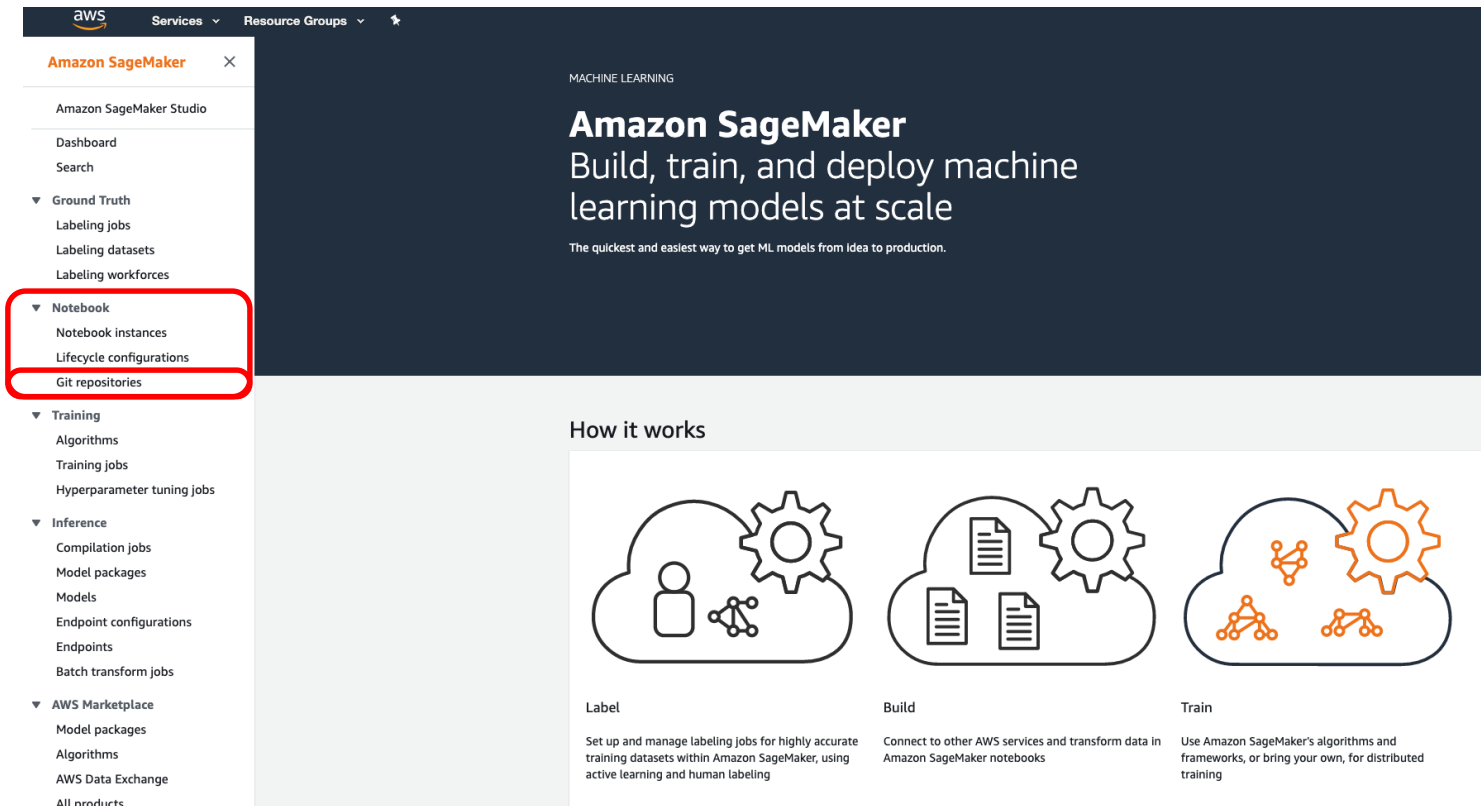


# Deep Learning for Autonomous Driving

## Git in SageMaker, Training Instances, and SageMaker Hints

# AWS – How to use your personal Git in Sagemaker

- This is a quick guide how to use a Git with Sagemaker
- It is assumed that you already know how to set up a sagemaker notebook



# First Steps – Add Private Repository

Name of the repository in Sagemaker

Create new secret to access your GitHub/GitLab

Name of the secret (the secret can be reused for other repositories)

**Add repository**

**Repository settings**

What type of repository do you want to add?

☐ AWS CodeCommit  
Highly available Git source control from AWS. Includes encryption, IAM integration, and more.

☒ GitHub/Other Git-based repo  
Connect GitHub or other Git-based repository. Store credentials with AWS Secrets Manager.

Amazon SageMaker repository name  
DLAD-Ex2-Git

The repository 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)

Git Repository URL  
https://github.com/alexliniger/DLAD-Ex2

Repository branch name - optional

Git credentials  
Amazon SageMaker saves your credentials using [AWS Secret Manager](#).

☐ Use existing secret  
☒ Create secret  
☐ No Secret

Secret name  
For security, Amazon SageMaker manages your credentials with [AWS Secret Manager](#).  
github-secret

The secret name must be ASCII letters, digits, or the following characters : / \_ + = . @ -

Username  
alexliniger@gmail.com

Password  
\*\*\*\*\*

Create secret

If you use GitHub or GitLab

Link to your personal Git

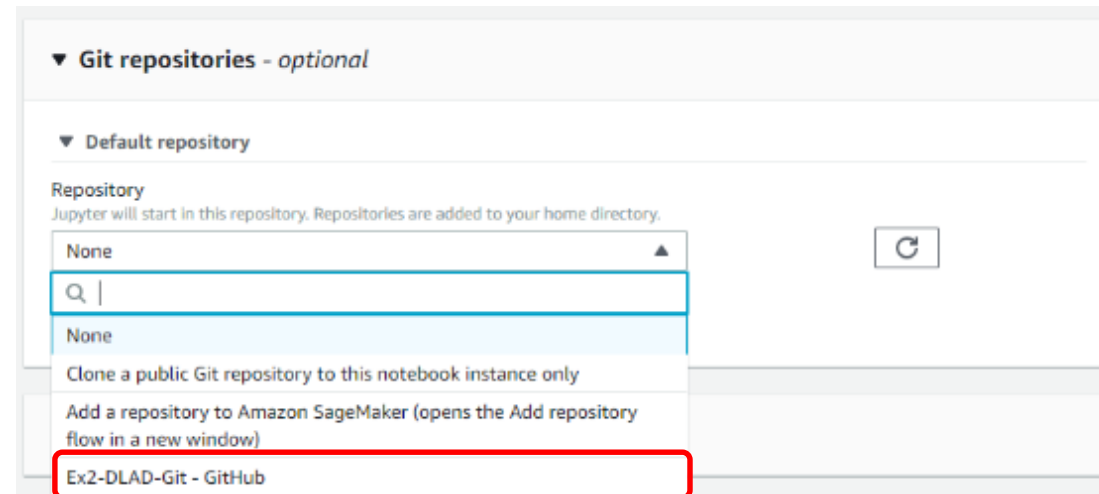
1. If you use gitlab.ethz.ch you need to use the https clone link:  
[https://gitlab.ethz.ch/user/dlad\\_ex2\\_multitask.git](https://gitlab.ethz.ch/user/dlad_ex2_multitask.git)

Username and Password of your Git

1. You can use a token instead of your password
2. GitHub needs your email as username, ETH-GitLab only your nethz username

# Create Notebook

- Create notebook identical to tutorial but use your own Git



▼ Git repositories - optional

▼ Default repository

Repository  
Jupyter will start in this repository. Repositories are added to your home directory.

None

Q |

None

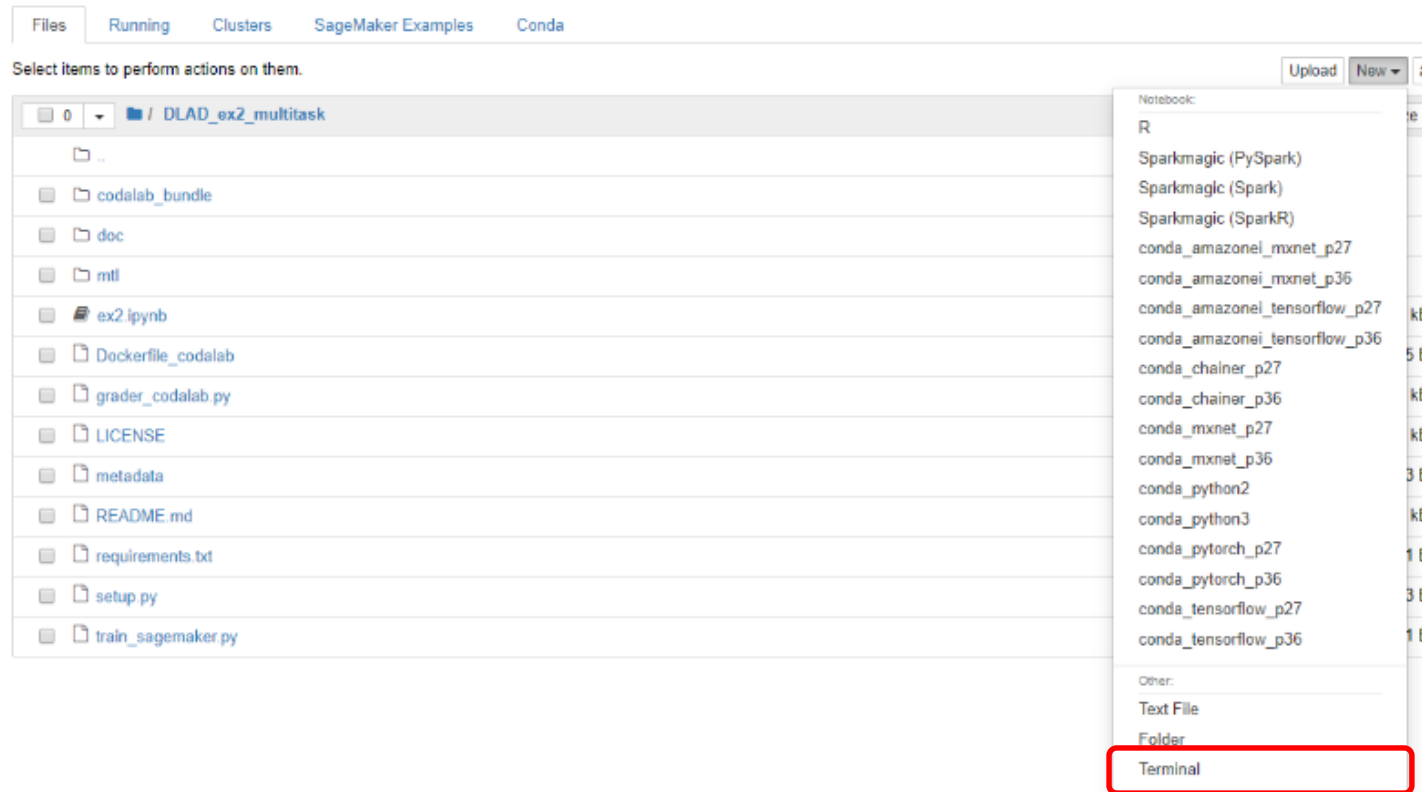
Clone a public Git repository to this notebook instance only

Add a repository to Amazon SageMaker (opens the Add repository flow in a new window)

Ex2-DLAD-Git - GitHub

# Work with your Git within your Notebook Instance

- Open notebook instance
- New -> Terminal



# Work with Terminal

- This opens a terminal in a new browser windows
- Go to project folder -> `cd SageMaker/<project-name>`
- Use git in the terminal

```
jupyter

sh-4.2$ cd SageMaker/DLAD_ex2_multitask/
sh-4.2$ ls
codalab_bundle  Dockerfile_codalab  grader_codalab.py  metadata  README.md  setup.py
doc             ex2.ipynb           LICENSE           mtl       requirements.txt  train_sagemaker.py
sh-4.2$ git status
On branch master
Your branch is behind 'origin/master' by 7 commits, and can be fast-forwarded.
(use "git pull" to update your local branch)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   ex2.ipynb

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .ipynb_checkpoints/
        mtl/utlils/.ipynb_checkpoints/
        mtl/utlils/Untitled.ipynb

no changes added to commit (use "git add" and/or "git commit -a")
sh-4.2$ git stash
Saved working directory and index state WIP on master: 03369a4 Update ex2.ipynb
sh-4.2$ git pull
Updating 03369a4..e60f6d
Fast-forward
 .gitignore                               | 1 +
 README.md                               | 2 +-
 codalab_bundle/bundle.zip                | Bin 0 -> 148703 bytes
 codalab_bundle/competition.yaml          | 2 +-
 grader_codalab.py                       | 8 +++++
 mtl/experiments/experiment_senseq_with_depth.py | 2 +-
 mtl/scripts/export_assignment.py         | 3 +-
 mtl/scripts/(codalab_create_program.sh => export_grader.sh) | 3 +-
 mtl/scripts/grader.py                   | 2 +-
 mtl/scripts/train.py                     | 18 ++++++++
 mtl/utlils/config.py                     | 15 ++++++++
 mtl/utlils/daemon_ngrok.py               | 71 ++++++++
 mtl/utlils/(tensorboard_daemon.py => daemon_tensorboard.py) | 18 ++++++++
 ngrok                                     | Bin 0 -> 26633198 bytes
14 files changed, 123 insertions(+), 22 deletions(-)
create mode 100644 codalab_bundle/bundle.zip
rename mtl/scripts/(codalab_create_program.sh => export_grader.sh) (64%)
create mode 100644 mtl/utlils/daemon_ngrok.py
rename mtl/utlils/(tensorboard_daemon.py => daemon_tensorboard.py) (82%)
create mode 100755 ngrok
sh-4.2$
```

# Applying for SageMaker training instances

- To train with GPU instances you need to apply for p2.xlarge training instances
- Navigate to **Support** and **Create Case** a new support case

**Create case** [Info](#)

☐ **Account and billing support**  
Assistance with account and billing-related inquiries

☒ **Service limit increase**  
Requests to increase the service limit of your AWS resources

☐ **Technical support**  
Service-related technical issues and third-party applications  
Unavailable under the Basic Support Plan

**Case classification**

Limit type  
SageMaker ▼

Severity [Info](#)  
The severity levels available are determined by your support subscription.

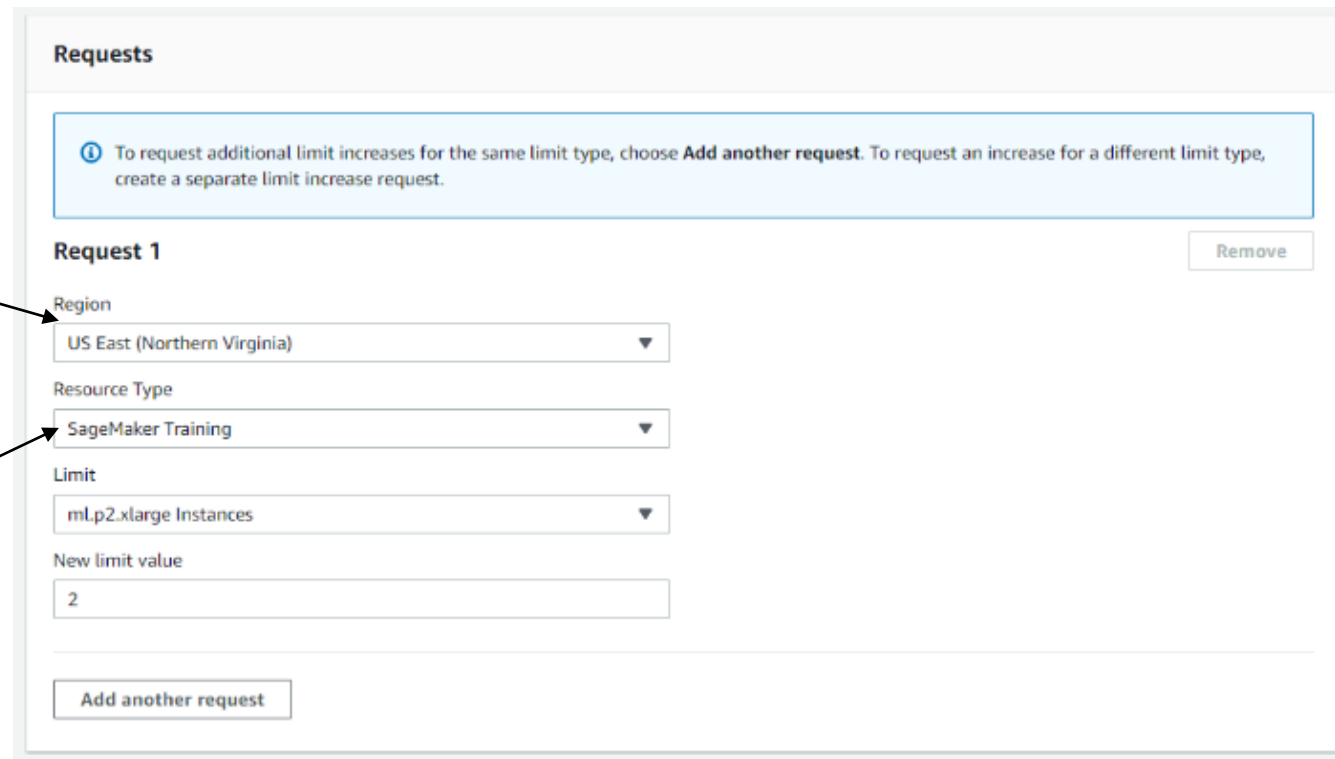
General question ▼

# Applying for SageMaker training instances

- To train with GPU instances you need to apply for p2.xlarge training instances
- Navigate to **Support** and **Create Case** a new support case

Instances are only available in this region

Important that you apply for **Training**



The screenshot shows the 'Requests' section of the AWS SageMaker console. At the top, there is a blue information box with a warning icon and text: 'To request additional limit increases for the same limit type, choose **Add another request**. To request an increase for a different limit type, create a separate limit increase request.' Below this, 'Request 1' is listed with a 'Remove' button. The request details are as follows:

Field	Value
Region	US East (Northern Virginia)
Resource Type	SageMaker Training
Limit	ml.p2.xlarge Instances
New limit value	2

At the bottom of the request form, there is a button labeled 'Add another request'.



# Applying for SageMaker training instances

- To train with GPU instances you need to apply for p2.xlarge training instances
- Navigate to **Support** and **Create Case** a new support case

Change name



**Case description**

Use case description

Dear Support,

We need AWS sagemaker for a lecture about deep learning, therefore I would like to apply for two p2.xlarge training instances. Could you also increase my limit to include two p2.xlarge "Managed Spot Training" instances.

Best regards,  
Alex Liniger

Maximum 5000 characters (4738 remaining)

**▼ Contact options**

Preferred contact language

English

Contact methods [Info](#)

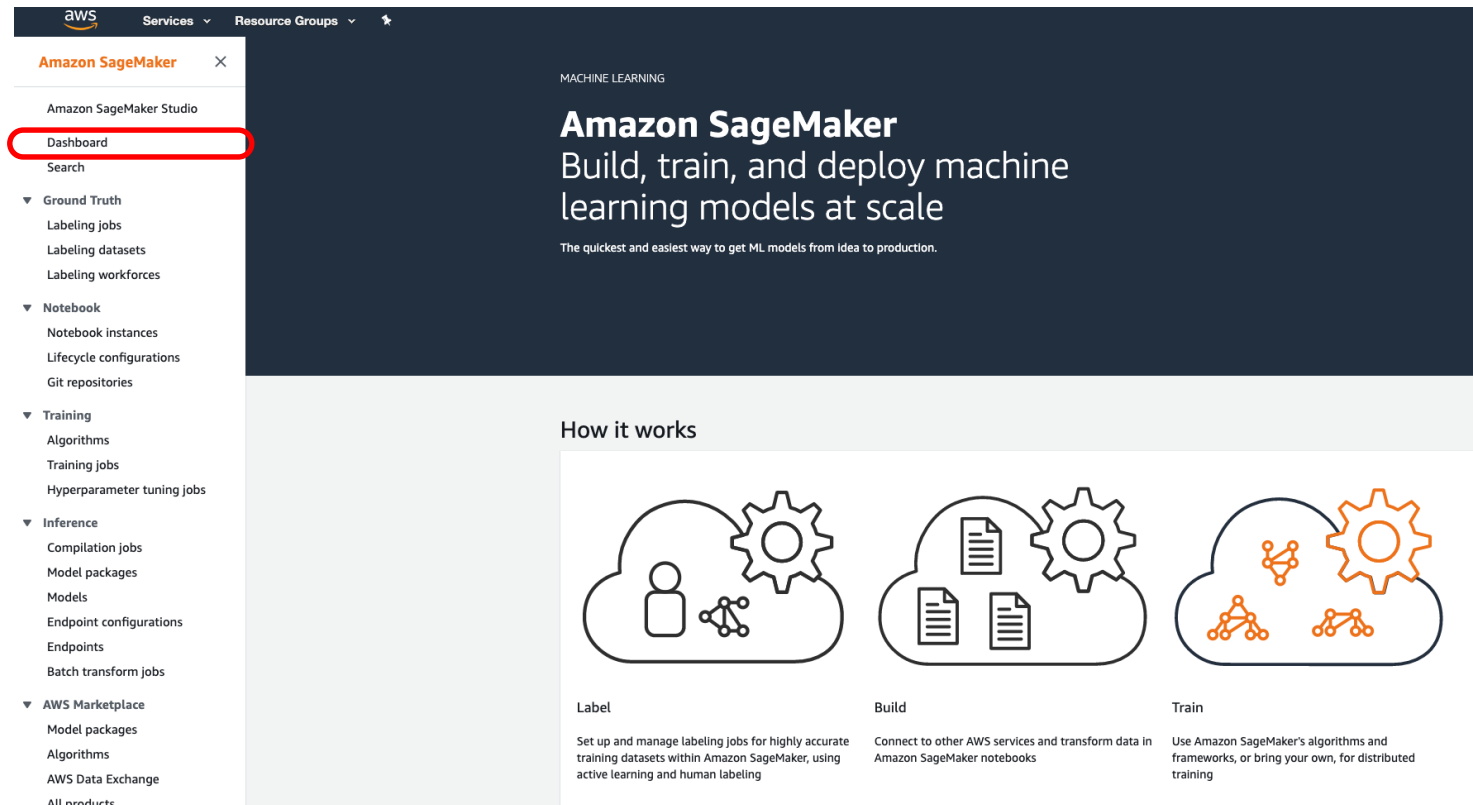
**Web** ☒   
Via email and Support Center

**Phone** ☐   
We call you back at your number

Cancel Submit

# SageMaker DashBoard

- See an overview of training jobs and notebooks



The screenshot shows the Amazon SageMaker console interface. On the left, a sidebar menu lists various services. The 'Dashboard' link is highlighted with a red rectangle. The main content area displays the 'Amazon SageMaker' header, a subtitle 'Build, train, and deploy machine learning models at scale', and a brief description. Below this, a section titled 'How it works' illustrates the SageMaker workflow through three stages: Label, Build, and Train, each represented by an icon and a short description.

**Amazon SageMaker**  
Build, train, and deploy machine learning models at scale  
The quickest and easiest way to get ML models from idea to production.

**How it works**

**Label**  
Set up and manage labeling jobs for highly accurate training datasets within Amazon SageMaker, using active learning and human labeling

**Build**  
Connect to other AWS services and transform data in Amazon SageMaker notebooks

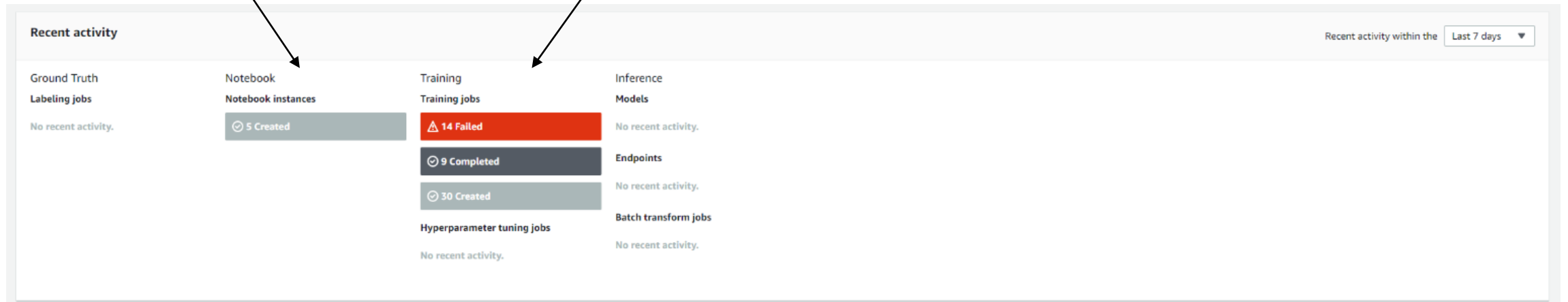
**Train**  
Use Amazon SageMaker's algorithms and frameworks, or bring your own, for distributed training

# SageMaker DashBoard

- See an overview of training jobs and notebooks

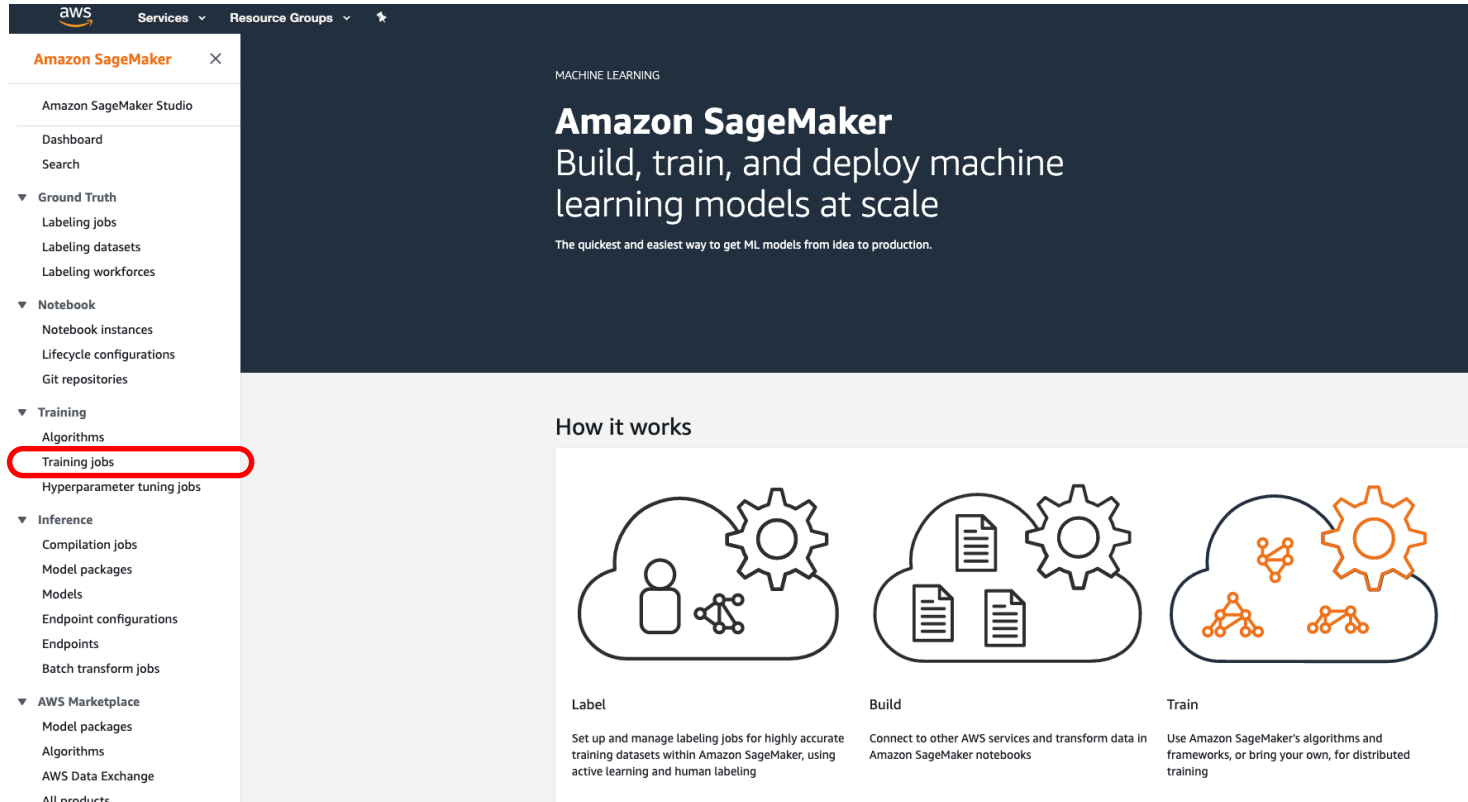
Shows running notebooks

Shows how many training Jobs are running  
Can be helpful if you do not have enough instances and you do not know why



# SageMaker Training Jobs

- Overview of all past and current training jobs



The screenshot shows the Amazon SageMaker console interface. On the left, a navigation sidebar lists various services. Under the 'Training' category, the 'Training jobs' option is highlighted with a red rectangle. The main content area displays the 'Amazon SageMaker' header with the tagline 'Build, train, and deploy machine learning models at scale'. Below this, a section titled 'How it works' illustrates the SageMaker workflow in three steps: 'Label' (setting up and managing labeling jobs), 'Build' (connecting to other AWS services and transforming data), and 'Train' (using SageMaker's algorithms and frameworks for distributed training).

**Amazon SageMaker**

MACHINE LEARNING

**Amazon SageMaker**  
Build, train, and deploy machine learning models at scale

The quickest and easiest way to get ML models from idea to production.

**How it works**

**Label**  
Set up and manage labeling jobs for highly accurate training datasets within Amazon SageMaker, using active learning and human labeling

**Build**  
Connect to other AWS services and transform data in Amazon SageMaker notebooks

**Train**  
Use Amazon SageMaker's algorithms and frameworks, or bring your own, for distributed training

# SageMaker Training Jobs

- Info about all training job

	Name	Creation time	Duration	Status
<input type="radio"/>	<a href="#">pytorch-training-2020-03-30-15-21-13-968</a>	Mar 30, 2020 15:21 UTC	an hour	✔ Completed
<input type="radio"/>	<a href="#">pytorch-training-2020-03-27-21-35-42-590</a>	Mar 27, 2020 21:35 UTC	3 hours	✔ Completed
<input type="radio"/>	<a href="#">pytorch-training-2020-03-27-17-18-40-609</a>	Mar 27, 2020 17:18 UTC	3 hours	✔ Completed
<input type="radio"/>	<a href="#">pytorch-training-2020-03-27-15-50-59-480</a>	Mar 27, 2020 15:51 UTC	5 hours	⊖ Stopped
<input type="radio"/>	<a href="#">pytorch-training-2020-03-27-13-28-13-038</a>	Mar 27, 2020 13:28 UTC	an hour	✔ Completed
<input type="radio"/>	<a href="#">pytorch-training-2020-03-27-13-08-15-559</a>	Mar 27, 2020 13:08 UTC	7 minutes	✖ Failed
<input type="radio"/>	<a href="#">pytorch-training-2020-03-27-13-03-46-915</a>	Mar 27, 2020 13:03 UTC	3 minutes	⊖ Stopped
<input type="radio"/>	<a href="#">pytorch-training-2020-03-27-13-03-20-707</a>	Mar 27, 2020 13:03 UTC	5 minutes	⊖ Stopped
<input type="radio"/>	<a href="#">pytorch-training-2020-03-27-10-55-50-897</a>	Mar 27, 2020 10:55 UTC	an hour	✔ Completed
<input type="radio"/>	<a href="#">pytorch-training-2020-03-27-10-20-38-936</a>	Mar 27, 2020 10:20 UTC	39 minutes	⊖ Stopped

# SageMaker Training Jobs

- Detailed info about training job
- Including hyperparameters, run time, ...

Job settings			
Job name	Status	SageMaker metrics time series	IAM role ARN
pytorch-training-2020-03-30-15-21-13-968	✔ Completed <a href="#">View history</a>	Enabled	[REDACTED]
ARN	Creation time	Training time (seconds)	
arn:aws:sagemaker:us-east-1:[REDACTED]:training-job/pytorch-training-2020-03-30-15-21-13-968	Mar 30, 2020 15:21 UTC	4288	
	Last modified time	Billable time (seconds)	
	Mar 30, 2020 16:35 UTC	4288	
		Managed spot training savings	
		0%	
		Tuning job source/parent	
		-	

# SageMaker Training Jobs

- Detailed info about training job
- Also GPU and memory usage

