- Please refer to https://github.com/aws-samples/deep-learning-models/blob/master/utils/tensorflow/preprocess_imagenet.py (code) and https://docs.aws.amazon.com/ko_kr/dlami/latest/devguide/tutorial-horovod-tensorflow.html (document).
  - `python preprocess_imagenet.py \ --local_scratch_dir=[YOUR DIRECTORY] \ --imagenet_username=[imagenet account] \ --imagenet_access_key=[imagenet access key]`
  - `python tensorflow_image_resizer.py \ -d imagenet \ -i [PATH TO TFRECORD TRAINING DATASET] \ -o [PATH TO RESIZED TFRECORD TRAINING DATASET] \ --subset_name train \ --num_preprocess_threads 60 \ --num_intra_threads 2 \ --num_inter_threads 2`
- [Additional Notes] The original document uses the small number of intra-op(multiple threads within one op; for example, while doing matrix multiplication operation we can divide the op by multiple threads) and inter-op(thread-pool size per one executor) such that `--num_intra_threads 2 \ --num_inter_threads 2`. But, you can give higher number of intra-op and inter-op.

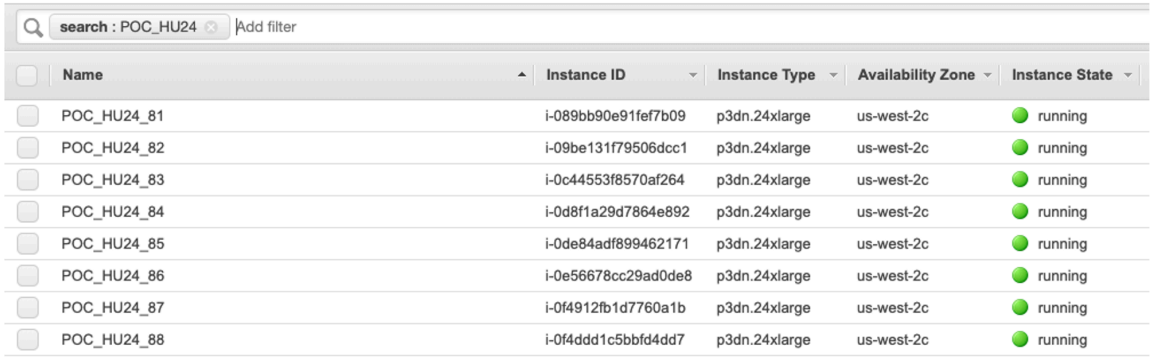
Backing up and Copying to S3

- After data transformation, create a new bucket and sync or copy feature sets to the bucket.
- Create a snapshot of the EBS volume.

Step 2. Training ResNet-50 Model with Horovod

[Before get started] If you just want to train on a single machine, you may refer to https://medium.com/@julsimon/imagenet-part-2-the-road-goes-ever-on-and-on-578f09a749f9 (RecordIO) and https://github.com/tensorflow/models/tree/master/official/r1/resnet (TFRecord)

- Create an EC2 instance for Training (Deep Learning AMI (Ubuntu 16.04) or Deep Learning AMI (Amazon Linux)). `p3.16xlarge` or `p3dn.24xlarge` is recommended if you need to do distributed GPU training using Uber's Horovod or Tensorflow's DistributedStrategy). Please also note that the default root volume size is 75GB, but I recommend you to increase 100GB since training logs and model checkpoints are stored in the root volume if you do not modify training configuration. If you not want to increase the volume size, then you can delete some conda environments such as Theano, Chainer, Caffe, and Caffe2 after logging in to the EC2 instance.
  - https://aws.amazon.com/ko/getting-started/tutorials/get-started-dlami/
- If you want to train on distributed GPUs, then you need to create multiple GPU instances with the same setting. For example, the below figure shows 8 `p3dn.24xlarge` instances.

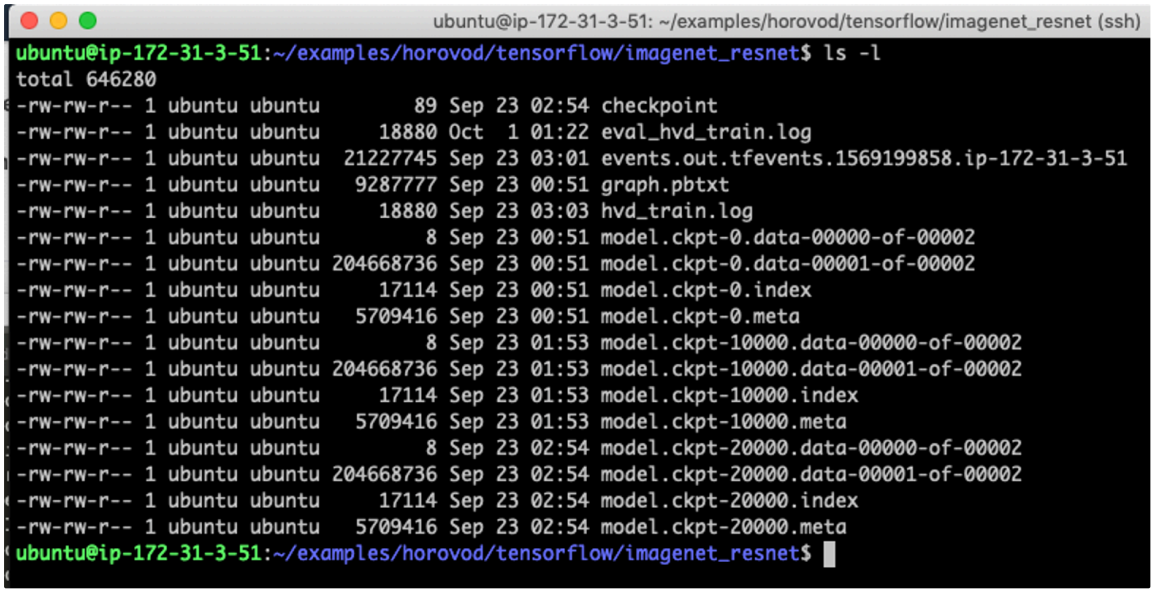


search : POC_HU24 | Add filter

Name	Instance ID	Instance Type	Availability Zone	Instance State
POC_HU24_81	i-089bb90e91fef7b09	p3dn.24xlarge	us-west-2c	running
POC_HU24_82	i-09be131f79506dcc1	p3dn.24xlarge	us-west-2c	running
POC_HU24_83	i-0c44553f8570af264	p3dn.24xlarge	us-west-2c	running
POC_HU24_84	i-0d8f1a29d7864e892	p3dn.24xlarge	us-west-2c	running
POC_HU24_85	i-0de84adf899462171	p3dn.24xlarge	us-west-2c	running
POC_HU24_86	i-0e56678cc29ad0de8	p3dn.24xlarge	us-west-2c	running
POC_HU24_87	i-0f4912fb1d7760a1b	p3dn.24xlarge	us-west-2c	running
POC_HU24_88	i-0f4ddd1c5bbfd4dd7	p3dn.24xlarge	us-west-2c	running

- Please refer to the website for the remaining steps; https://docs.aws.amazon.com/dlami/latest/devguide/tutorial-horovod-tensorflow.html. Note that all code and all feature sets(TFRecord and RecordIO) must be on the same path on each server.

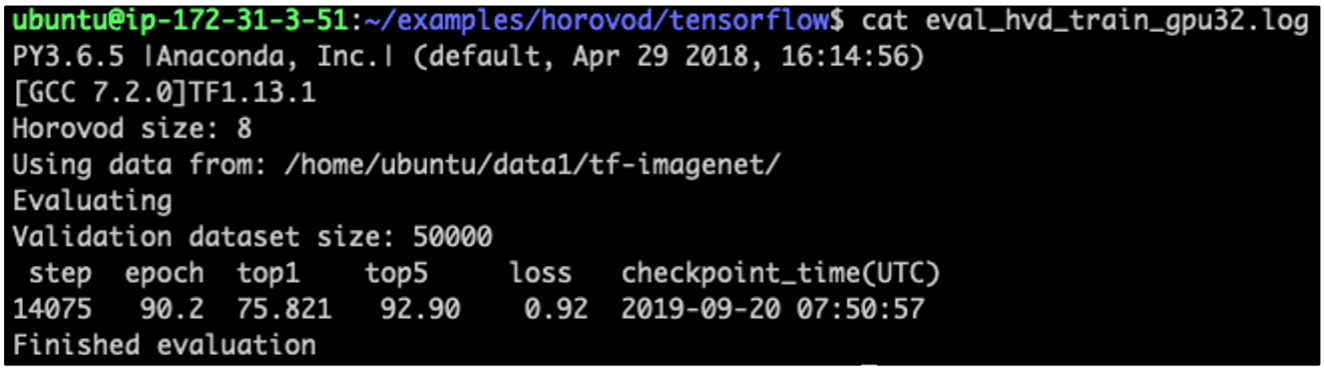
- After training, please check the training log and evaluation log by checking `imagenet_resnet` folder:



- vd_train_log (32 GPUS; 4 p3dn.24xlarge instances)

```
- Step Epoch Speed Loss FinLoss LR - 0 0.0 952.2 6.923 8.262
0.00100 - 1 0.0 2686.6 6.928 8.267 0.00305 - 50 0.3 22243.7
6.586 7.919 0.10353 - .. - 14000 89.5 21021.1 0.750 1.152
0.00012 - 14050 89.8 21818.7 0.583 0.985 0.00002 - Finished in
5289.161954164505
```

- eval_hvd_train.log (32 GPUS; 4 p3dn.24xlarge instances)



- hvd_train_log (64 GPUS; 8 p3dn.24xlarge instances)

```
- Step Epoch Speed Loss FinLoss LR - 0 0.0 1907.3 6.920 8.259
0.00100 - 1 0.0 5164.9 6.935 8.274 0.00920 - 50 0.6 43926.5
6.206 7.522 0.41119 - ... - 6950 88.9 43552.2 0.783 1.185
0.00125 - 7000 89.5 41958.4 0.624 1.027 0.00023 - Finished in
2685.1825189590454
```

- eval_hvd_train.log (64 GPUS; 8 p3dn.24xlarge instances)