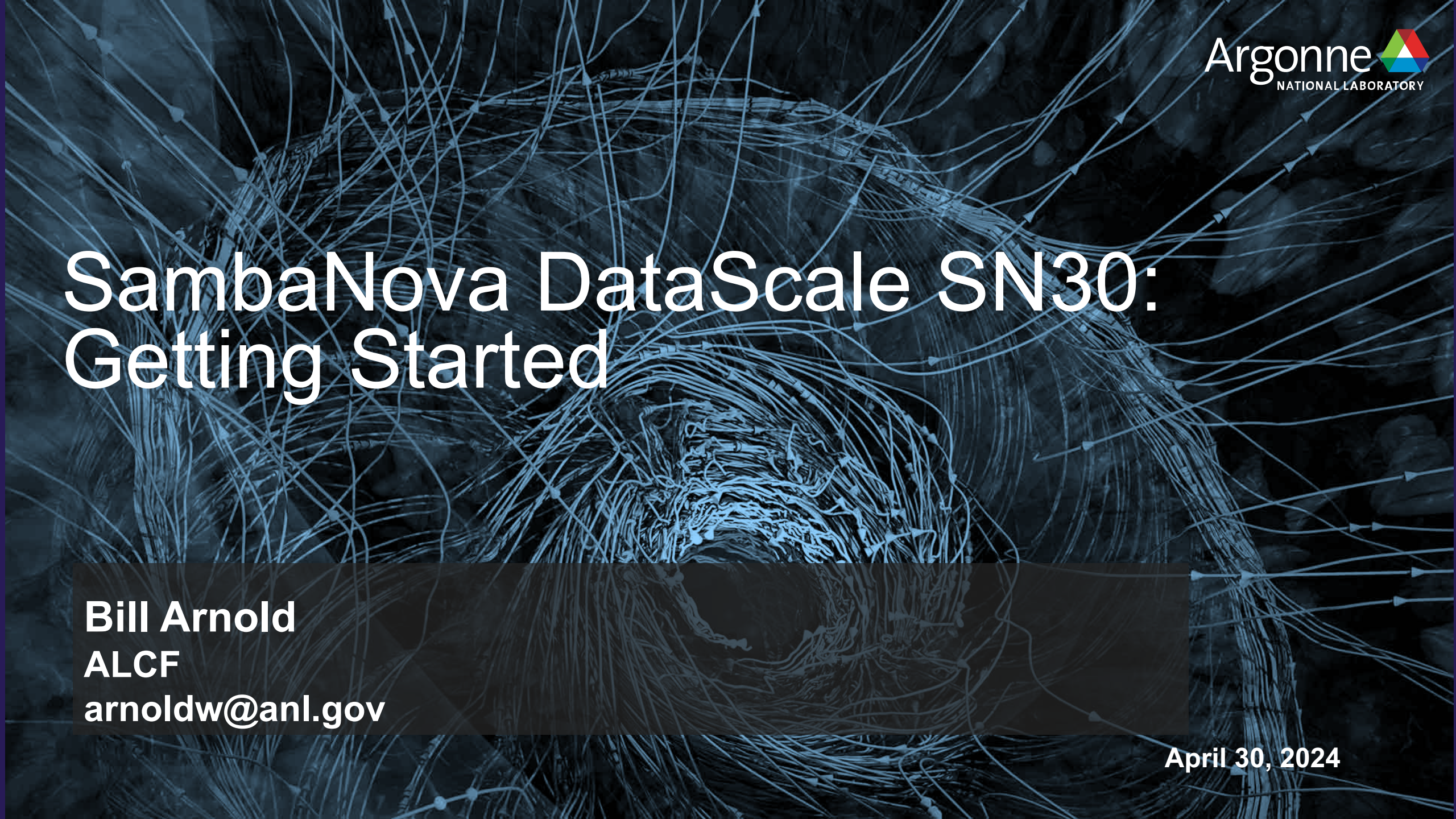


SambaNova DataScale SN30: Getting Started

Bill Arnold
ALCF
arnoldw@anl.gov

April 30, 2024

OVERVIEW

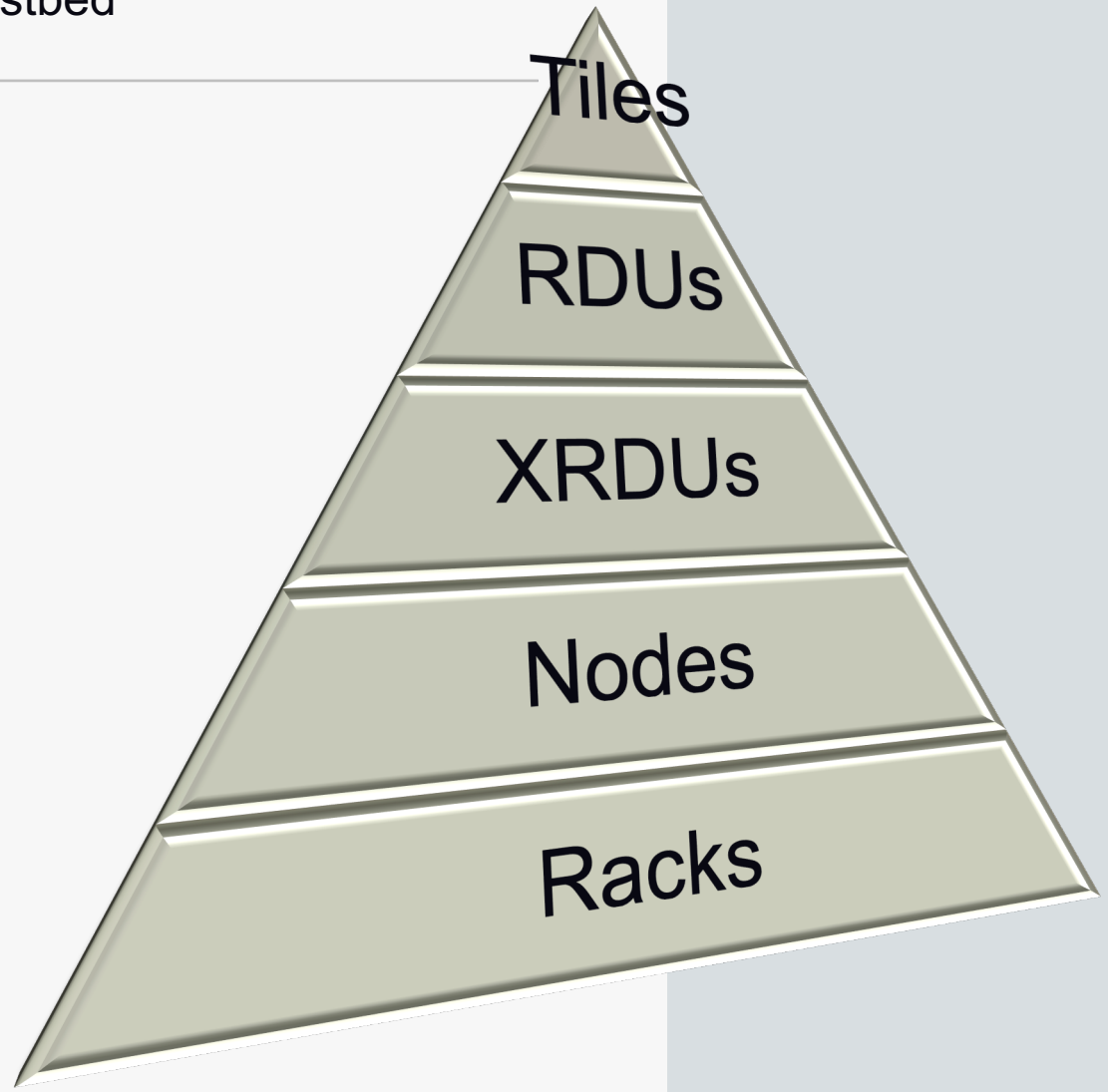
- System Overview
- Account access
- Steps to Login
- Environment setup
- Workflow
- Example programs

SambaNova Datascale SN30

https://www.alcf.anl.gov/alcf-ai-testbed

SambaNova Datascale SN30

- 4 Racks
- 8 nodes of SN30
- 8 RDUs or 4 XRDUs per node
- 8 Tiles per RDU
- Group of 4 tiles



SambaNova Datascale SN30

https://www.alcf.anl.gov/alcf-ai-testbed

SambaNova Datascale SN30

- 4 Racks
- 8 nodes of SN30
- 8 RDUs or 4 XRDUs per node
- 8 Tiles per RDU
- Group of 4 tiles

sn30-r1-h1	sn30-r1-h2
sn30-r2-h1	sn30-r2-h2
sn30-r3-h1	sn30-r3-h2
sn30-r4-h1	sn30-r4-h2

ACCOUNT ACCESS

- **Step 1 : Request for an ALCF account**

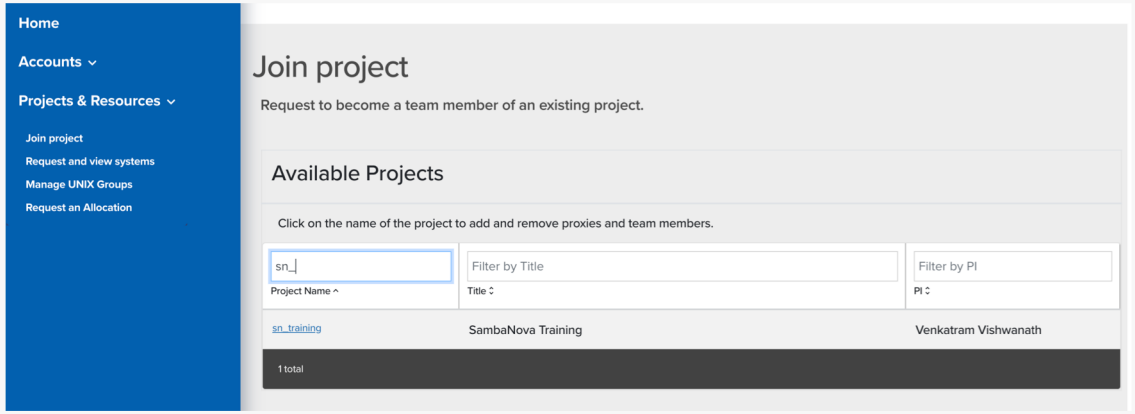
Request an ALCF account on our accounts page. (https://accounts.alcf.anl.gov)

Need an active project account.

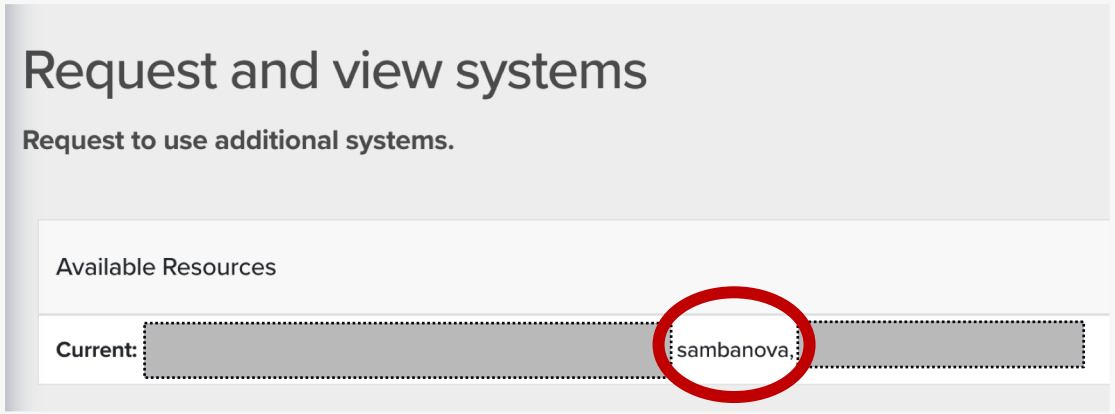
Help : **accounts@alcf.anl.gov**

ACCOUNT ACCESS

- **Step 2 : Request to Join project**



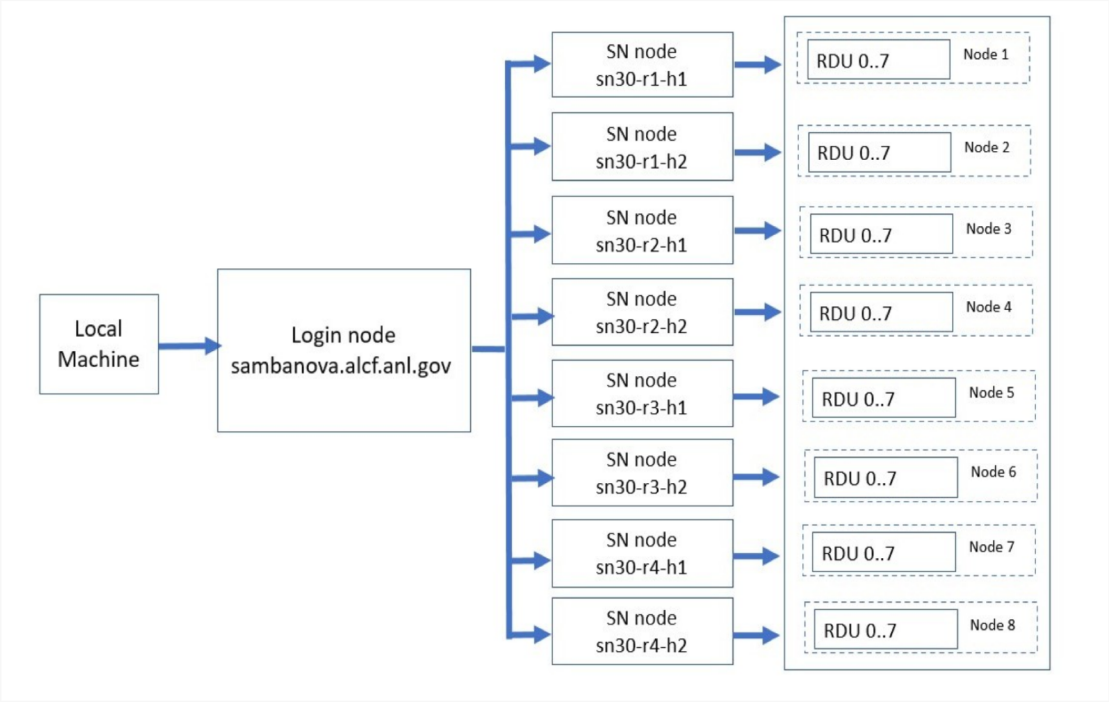
Join project under "sn_training".



Check for "sambanova" under Request and view systems

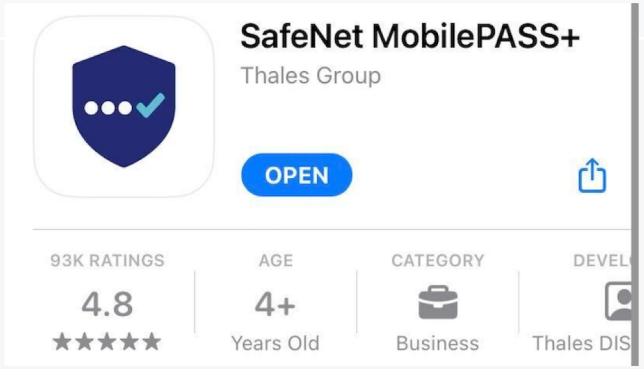
Connection and Login

- **Step 3 : Login to SN30 node.**



From local machine to login node :

```
ssh ALCFUserID@sambanova.alcf.anl.gov
Password: < MobilePASS+ code >
```



From login node to a SN30 node :

```
ALCFUserId@sm-login-1:~$ ssh sn30-r1-h1
```

Environment setup

- SDK (SambaFlow software stack and the associated environmental variables) is set up at login.
- Pre-built environments under /opt/sambaflow/apps/
- Create new virtual environment and install packages

```
python -m venv --system-site-packages my_env
source my_env/bin/activate

python3 -m pip install <package>
```

```
./private/anl/venv
./recommender/dlrm/venv
./recommender/deepinterest/venv
./recommender/ncf/venv
./starters/lenet/venv
./starters/upscalenet/venv
./starters/mlp/venv
./starters/ffn_mnist/venv
./starters/power_pca/venv
./starters/logreg/venv
./micros/venv
./nlp/tapas/venv
./nlp/data_processing/venv
./nlp/transformers_on_rdu/blocksparse/venv
./nlp/transformers_on_rdu/genslm/venv
./nlp/transformers_on_rdu/gpt13b/venv
./nlp/transformers_on_rdu/venv
./speech/speaker_diarization/venv
./speech/hubert/venv
./image/deepvit/venv
./image/object_detection/venv
./image/segmentation_3d/venv
./image/segmentation/venv
./image/classification/venv
```

Workflow

- **Compile**
  - Model compilation and ' **.pef** ' generation.
  - Maps the compute and memory resources required to run an application on RDUs
  - Re-compile only when model parameters change.
  - Significant compile times for large models.

```
srun python lenet.py compile -b=1 --pef-name="lenet" --output-folder="pef"
```

- **Run**
  - Model trained on RDU using the ".pef" generated as part of compile process and the training dataset.

```
srun python lenet.py run --pef="pef/lenet/lenet.pef"
```

Example programs

- /opt/sambaflow/apps/

```
drwxr-xr-x 7 root root  4096 Mar 27 04:15 image
drwxr-xr-x 5 root root  4096 Mar 27 04:16 nlp
drwxr-xr-x 6 root root  4096 Mar 27 04:17 micros
drwxr-xr-x 5 root root  4096 Mar 27 04:17 recommender
drwxr-xr-x 4 root root  4096 Mar 27 04:17 speech
drwxr-xr-x 8 root root  4096 Mar 27 04:18 starters
```

```
drwxr-xr-x 8 root root 4096 Mar 27 04:15 deepvit
drwxr-xr-x 5 root root 4096 Mar 27 04:15 object_detection
drwxr-xr-x 5 root root 4096 Mar 27 04:15 classification
drwxr-xr-x 8 root root 4096 Mar 27 04:16 segmentation_3d
drwxr-xr-x 5 root root 4096 Mar 27 21:09 segmentation
```

- /data/ANL/scripts

```
-rwxrwxr-x 1 root    SambaNova_Deployment   4284 Mar 31 18:13 Gpt1.5B_compile.sh
-rwxrwxr-x 1 nobody  SambaNova_Deployment   4391 Mar 31 18:13 Gpt1.5B_compile_single.sh
-rwxrwxr-x 1 root    SambaNova_Deployment   4491 Mar 31 18:13 Gpt1.5B_run.sh
-rwxrwxr-x 1 root    SambaNova_Deployment   4658 Mar 31 18:13 Gpt1.5B.sh
-rwxrwxr-x 1 nobody  SambaNova_Deployment   2160 Mar 31 18:14 Stream-DDR-BW.sh
-rwxrwxr-x 1 root    SambaNova_Deployment   2183 Mar 31 18:14 Stream-Host-BW.sh
-rwxrwxr-x 1 root    SambaNova_Deployment   2732 Mar 31 18:14 Unet3D-512.sh
-rwxrwxr-x 1 root    SambaNova_Deployment   2935 Mar 31 18:14 Unet3d.sh
-rwxrwxr-x 1 root    SambaNova_Deployment   2283 Mar 31 18:14 Uno.sh
```

Example program - mnist

- Path : "/opt/sambaflow/apps/starters/ffn_mnist
  ***Note : Make a copy of the apps directory into the home directory.**

- Activate the environment.

```
source /opt/sambaflow/apps/starters/ffn_mnist/venv/bin/activate
```

- Compile and Run

```
srun python ffn_mnist.py  compile -b 1 --pef-name="ffn_mnist" --mac-v2
srun python ffn_mnist.py  run -b 1 -p out/ffn_mnist/ffn_mnist.pef
```

- See https://docs.alcf.anl.gov/ai-testbed/sambanova/example-programs/

Example program – multinode

- Runs multiple instances of training on multiple tiles / RDUs / nodes in a data-parallel fashion.
- Gradient accumulation on the RDU.
- Refer example :
  - /data/ANL/scripts/Unet2d.sh
  - /data/ANL/scripts/unet_batch.sh
  - https://docs.alcf.anl.gov/ai-testbed/sambanova/example-multi-node-programs/multi-node-programs/
- Environmental Variables
  - OMP_NUM_THREADS (8/16/32)
  - SF_RNT_NUMA_BIND (2)
  - SF_RNT_FSM_POLL_BUSY_WAIT (1)
  - SF_RNT_DMA_POLL_BUSY_WAIT (1)
  - SAMBA_CCL_USE_PCIE_TRANSPORT (1)

Example program – multinode

For single instance run :

```
./Unet2d.sh compile <image size> <batch_size> <num of instances> <RunID>
./Unet2d.sh run <image size> <batch_size> <num of instances> <RunID>
```

```
./Unet2d.sh compile 256 256 1 unet2d_single_compile
./Unet2d.sh run 256 256 1 unet2d_single_run
```

For multi instance run :

```
./Unet2d.sh pcompile <image size> <batch_size> <num of instances> <RunID>
./Unet2d.sh prun <image size> <batch_size> <num of instances> <RunID>
```

```
./Unet2d.sh pcompile 256 256 8 unet2d_8inst_pcompile
./Unet2d.sh prun 256 256 8 unet2d_8inst_prun
```

Note : Run the scripts after copying to your home directory

Example program – multinode

- Activate environment

```
source /opt/sambaflow/apps/image/segmentation/venv/bin/activate
```

- Compile

```
python /opt/sambaflow/apps/image/segmentation/compile.py compile --mac-v2 --in-channels=3 --in-width=${2} --in-height=${2} --batch-size=${BS} --enable-conv-tiling --num-tiles=4 --pef-name=unet_train_${BS}_${2}_NP_${NUM_TILES}  --data-parallel -ws 2 --output-folder=${OUTDIR}
```

- Run

```
sbatch --gres=rdu:1 --tasks-per-node ${NP} --nodes 1 --nodelist $(hostname) --cpus-per-task=${cpus} $(pwd)/unet_batch.sh ${NP} ${NUM_WORKERS} ${BS} ${2} ${5}
```

```
srun --mpi=pmi2 python /opt/sambaflow/apps/image/segmentation//hook.py run --data-cache=${CACHE_DIR}  --data-in-memory --num-workers=${NUM_WORKERS} --enable-tiling  --min-throughput 395 --in-channels=3 --in-width=${IM} --in-height=${IM} --init-features 32 --batch-size=${BS} --epochs 10 --data-dir ${DS} --log-dir log_dir_unet_${IM}_${BS}_${NP} --data-parallel --reduce-on-rdu --pef=${OUTDIR}/unet_train_${BS}_${IM}_NP_4/unet_train_${BS}_${IM}_NP_4.pef
```

Utility commands

srun / sbatch : Run individual Python scripts in parallel with other scripts on cluster assigned by Slurm.

```
srun --nodelist=sn30-r1-h1 python lenet.py compile -b=1 --pef-name="lenet" --output-folder="pef"
```

squeue : Check the scheduled jobs

```
@sn30-r1-h1:~$ squeue
      JOBID PARTITION                NAME     USER ST       TIME  NODES NODELIST(REASON)
      14340 sambanova              python [redacted]  R       0:06      1 sn30-r1-h1
```

sntilestat : Check the process on each tile / node.

```
[redacted]@sn30-r1-h1:~$ sntilestat
TILE                %idle %exec %pload %aload %chkpt %quiesce    PID     USER COMMAND
/XRDU_0/RDU_0/TILE_0  82.7   3.7   10.7    2.9    0.0      0.0 492994          /opt/sambaflow/apps/starters/ffn_mnist/venv/bin/py
/XRDU_0/RDU_0/TILE_1  83.0   2.9   11.2    3.0    0.0      0.0 492994          /opt/sambaflow/apps/starters/ffn_mnist/venv/bin/py
/XRDU_0/RDU_0/TILE_2  83.9   2.7   10.9    2.4    0.0      0.0 492994          /opt/sambaflow/apps/starters/ffn_mnist/venv/bin/py
/XRDU_0/RDU_0/TILE_3  82.3   3.5   11.3    2.9    0.0      0.0 492994          /opt/sambaflow/apps/starters/ffn_mnist/venv/bin/py
/XRDU_0/RDU_0/TILE_4  83.9   2.4   10.9    2.7    0.0      0.0 492994          /opt/sambaflow/apps/starters/ffn_mnist/venv/bin/py
/XRDU_0/RDU_0/TILE_5  84.2   1.5   11.2    3.1    0.0      0.0 492994          /opt/sambaflow/apps/starters/ffn_mnist/venv/bin/py
/XRDU_0/RDU_0/TILE_6  84.1   1.1   11.7    3.1    0.0      0.0 492994          /opt/sambaflow/apps/starters/ffn_mnist/venv/bin/py
/XRDU_0/RDU_0/TILE_7  83.8   2.0   12.0    2.2    0.0      0.0 492994          /opt/sambaflow/apps/starters/ffn_mnist/venv/bin/py
/XRDU_0/RDU_1/TILE_0 100.0   0.0    0.0    0.0    0.0      0.0
/XRDU_0/RDU_1/TILE_1 100.0   0.0    0.0    0.0    0.0      0.0
```

sinfo , scancel

Important directory paths and links

- Sambaflow models path : /opt/sambaflow/apps/
- Model scripts : /data/ANL/scripts/
- Important datasets : /software/sambanova/dataset/
- /software /projects
- AI Testbed User Guide
- Sambanova Documentation.
- SambaTune.

Allocation programs

SN30 system is available for Director's Discretionary (DD) allocations and NAIRPilot program.

Allocation Request Form

THANK YOU

BACKUP

ACCOUNT ACCESS

- **Step 2 : Request for an ALCF account**

Request an ALCF account on our accounts page.

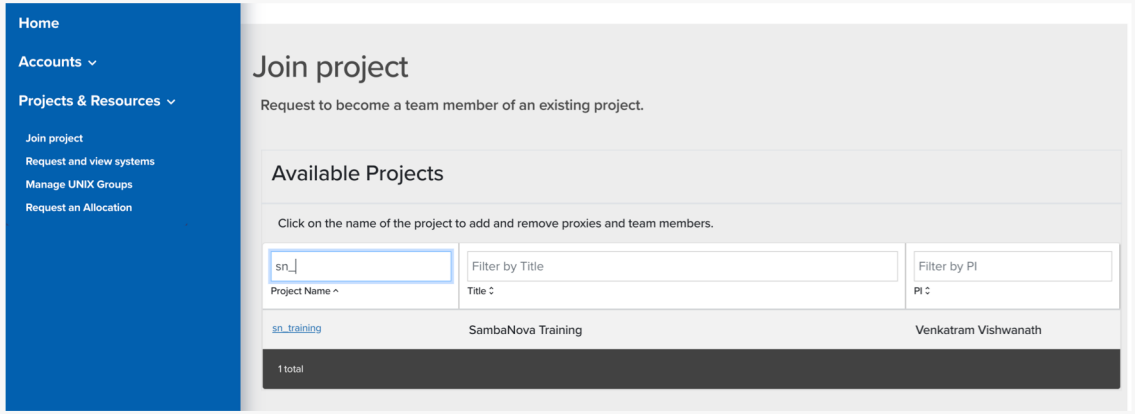
Need an active project account.

Help : **accounts@alcf.anl.gov**

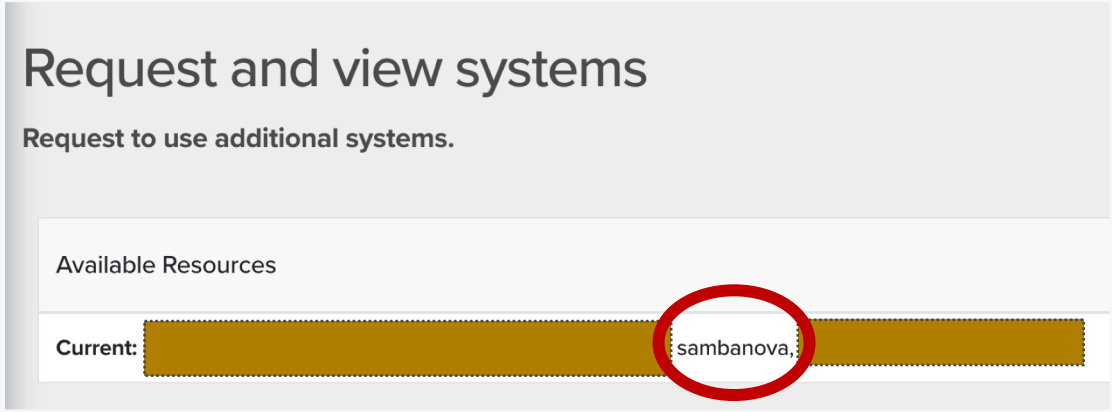
Argonne NATIONAL LABORATORY

ACCOUNT ACCESS

- **Step 2 : Request for an ALCF account**



Join project under "sn_training".

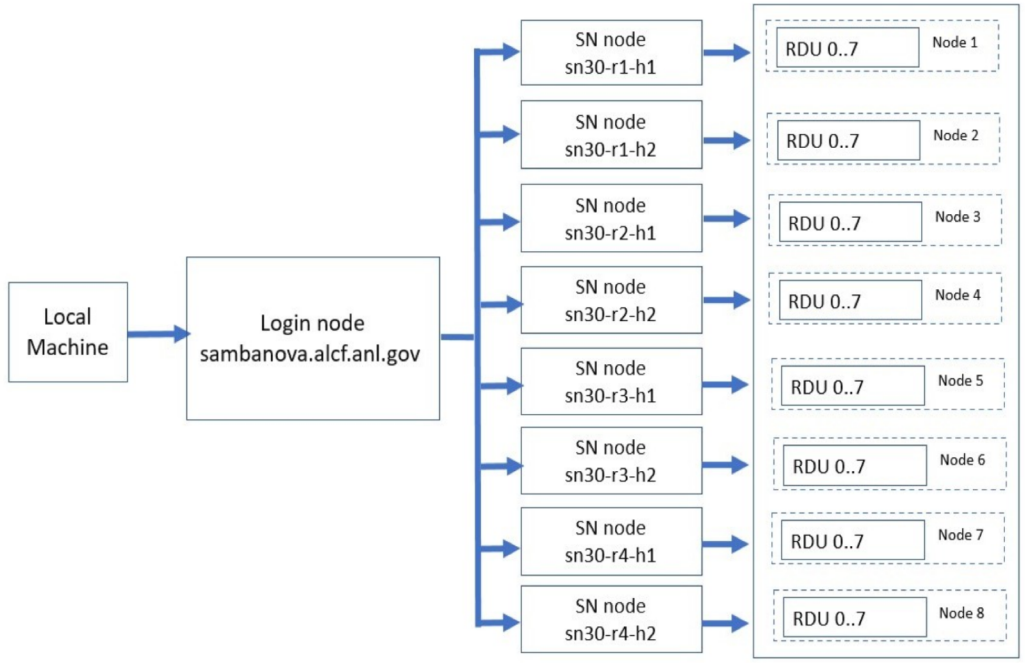


Check for "sambanova" under Request and view systems

Argonne NATIONAL LABORATORY

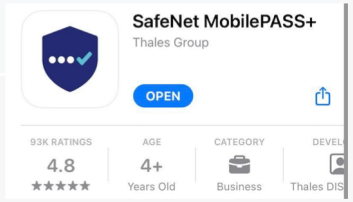
Connection and Login

- **Step 3 : Login to SN30 node.**



From local machine to login node :

```
ssh ALCFUserID@sambanova.alcf.anl.gov
Password: < MobilePASS+ code >
```



From login node to a SN30 node :

```
ALCFUserId@sm-login-1:~$ ssh sn30-r1-h1
```