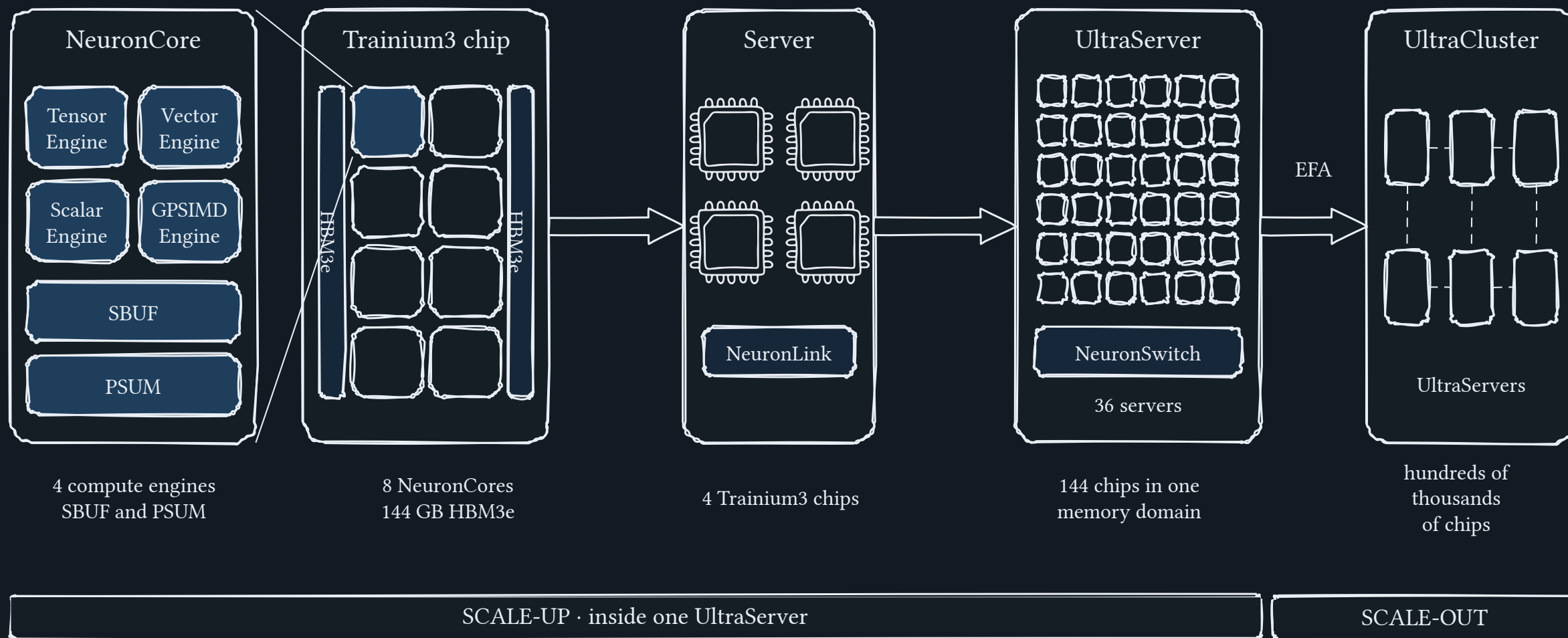


Write PyTorch. Run on *AWS* Trainium.

Robert Hundt · VP, Distinguished Engineer, Annapurna Labs, Amazon

Trainium is Amazon's AI chip, and millions of chips are deployed

Frontier models, trained and served.



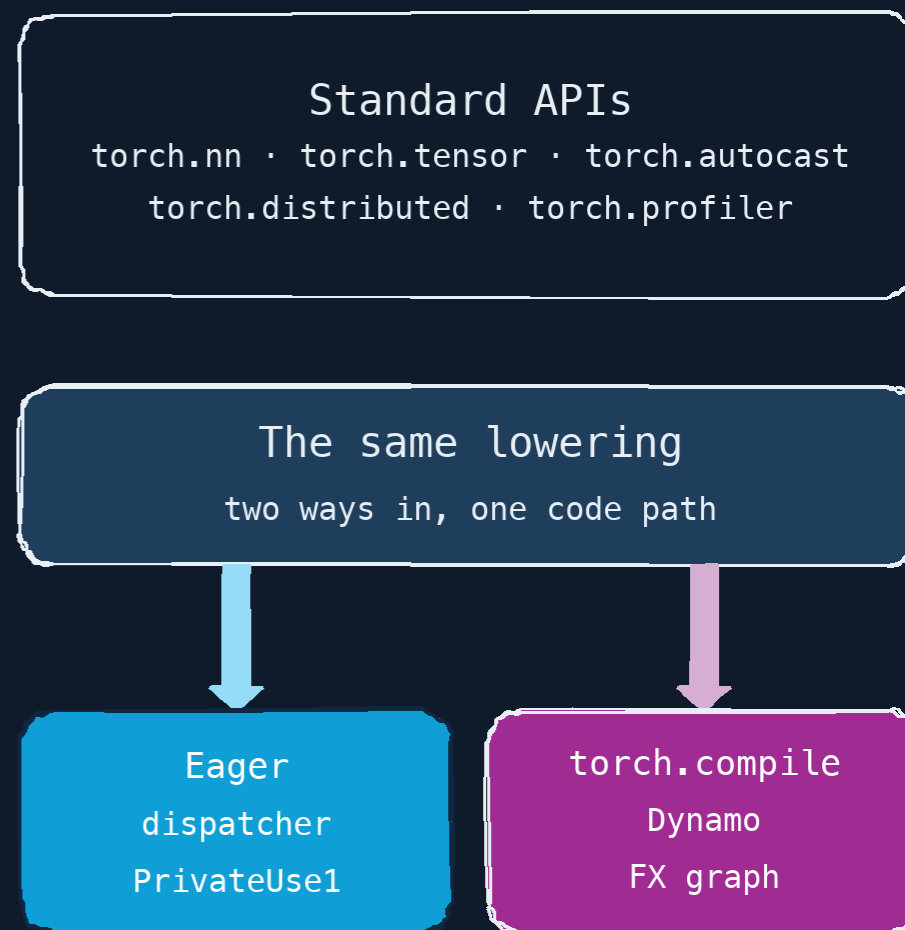
Announcing TorchNeuron, the native PyTorch backend for Trainium

```
import torch

from torch.nn.parallel import DistributedDataParallel as DDP

model = MyModel()
model.to("neuron")
model = DDP(model)
optimizer = torch.optim.AdamW(model.parameters())

for batch, targets in dataloader:
    optimizer.zero_grad()
    loss = loss_fn(model(batch), targets)
    loss.backward()
    optimizer.step()
```



Train with the libraries you already use

Built for Research

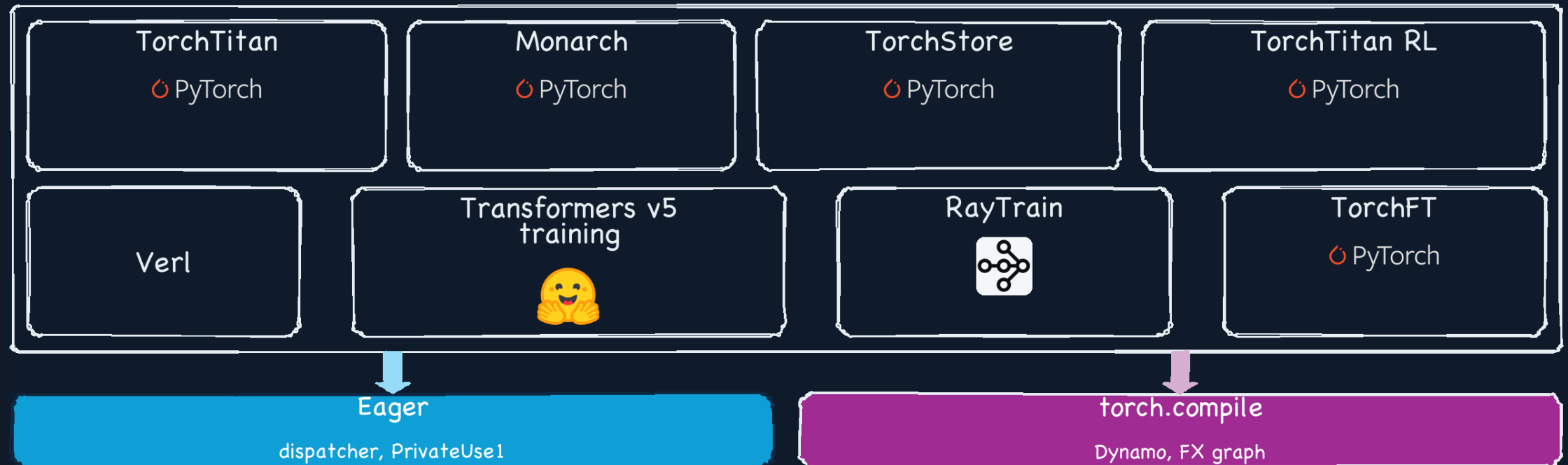
Researchers build and train on Trainium with

- the same PyTorch
- the same code and distributed idioms
- the same debugging, profiling, (and more)

Broad Model Support

Most of the models and architectures developers bring run

- Dense and MoE transformers
- Diffusion and vision-language
- State-space and encoder models



Video generation · World models · Vision-language · Robotics · MoE · LLMs · Multi-modality

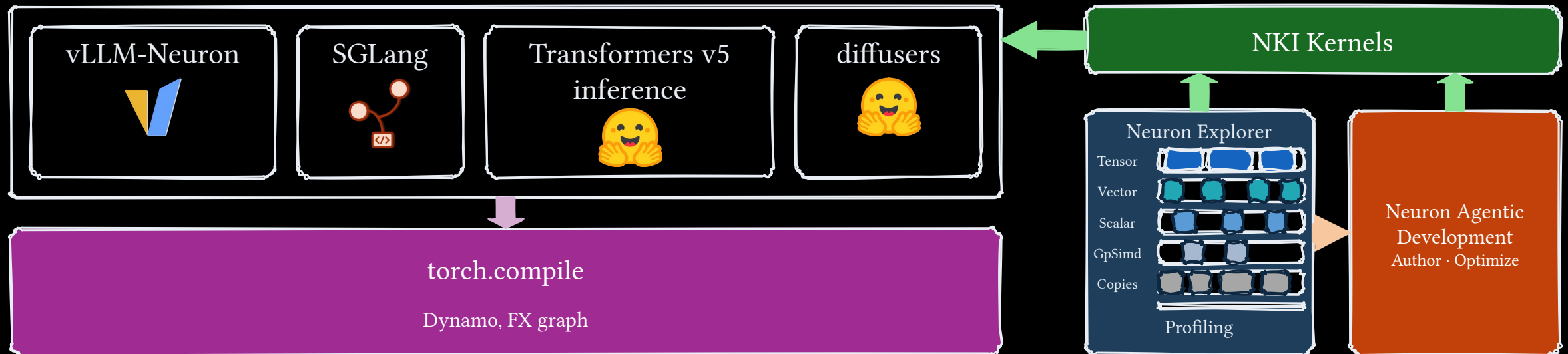
The serving software you use

NKI Kernels for best performance

- NKI kernels register as standard PyTorch custom operators
- NKI-lib ships tuned kernels for dense, MoE and multimodal
- Use Neuron Agentic Development to author and optimize them

Tune it with Neuron Explorer

- PyTorch down to ISA, time, FLOPs, data movement, at every level
- Per-engine hotspots, hardware events mapped to source in VS Code
- Distributed runs profiled with no code change



Quality and Coverage: Far beyond PyTorch test suite

Models swept

115,000

the most-downloaded, swept blind
every night

Architecture classes in them

841

tracked nightly against a GPU
reference

Training pass rate

78%

of the 841 in eager, 70% under
torch.compile

Text generation

88%

of those classes in eager, 80% under
torch.compile

Native operators

900

of the 1,300-plus registered in PyTorch

Customers already train and serve on it

Deepgram

"In our evaluation, Trainium processed approximately 60% more input audio seconds per dollar than our existing GPU baselines."

 CARTESIA

"With PyTorch native support on Trainium, we ran our Sonic and Ink inference pipeline with minimal code changes and focused custom NKI kernel work only where it mattered most. The familiar tooling made validating accuracy straightforward."

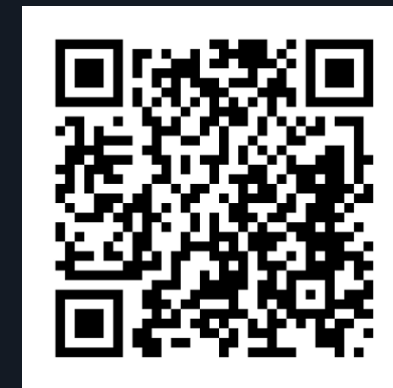
Customers already train and serve on it

PyTorch's vision is straightforward: developers should have a consistent experience across hardware platforms. Native support for AWS Trainium expands that choice, enabling researchers and engineers to experiment and iterate using the eager execution and distributed APIs they already know. Teams can take advantage of Trainium's capabilities without reinventing their workflows or leaving the PyTorch ecosystem."

It's public today: try it

`pip install torch-neuronx`

github.com/aws-neuron/torch-neuronx · open source



Attend our sessions

Day	Time	Session	Room	With
Tue Oct 20	2:15 PM	Serving: KV transfer in vLLM	LL20C	Mistral AI, Red Hat
	3:05 PM	Training: TorchTitan and Transformers v5	LL21D	Hugging Face
	5:30 PM	RL: native PyTorch on Trainium	LL21A	RadixArk
	6:00 PM	Posters: serving and profiling	Community Expo	
Wed Oct 21	10:40 AM	Live demo: reinforcement learning	Demo Theater	
	12:20 PM	Profiling: torch.profiler on Trainium	LL21A	
Both days		AWS booth	Community Expo	

It's public today

```
pip install torch-neuronx
```

github.com/aws-neuron/torch-neuronx

Apache 2.0

