



# Select the right ML instance for your training and inference jobs

Shashank Prasanna

Sr. Developer Advocate, AI/ML

AWS





Shashank Prasanna  
**Sr. Developer Advocate, AI/ML  
AWS**

 @shshnkp

 [linkedin.com/in/shashankprasanna](https://www.linkedin.com/in/shashankprasanna)

 [medium.com/@shashankprasanna](https://medium.com/@shashankprasanna)

# What are we going to talk about today?

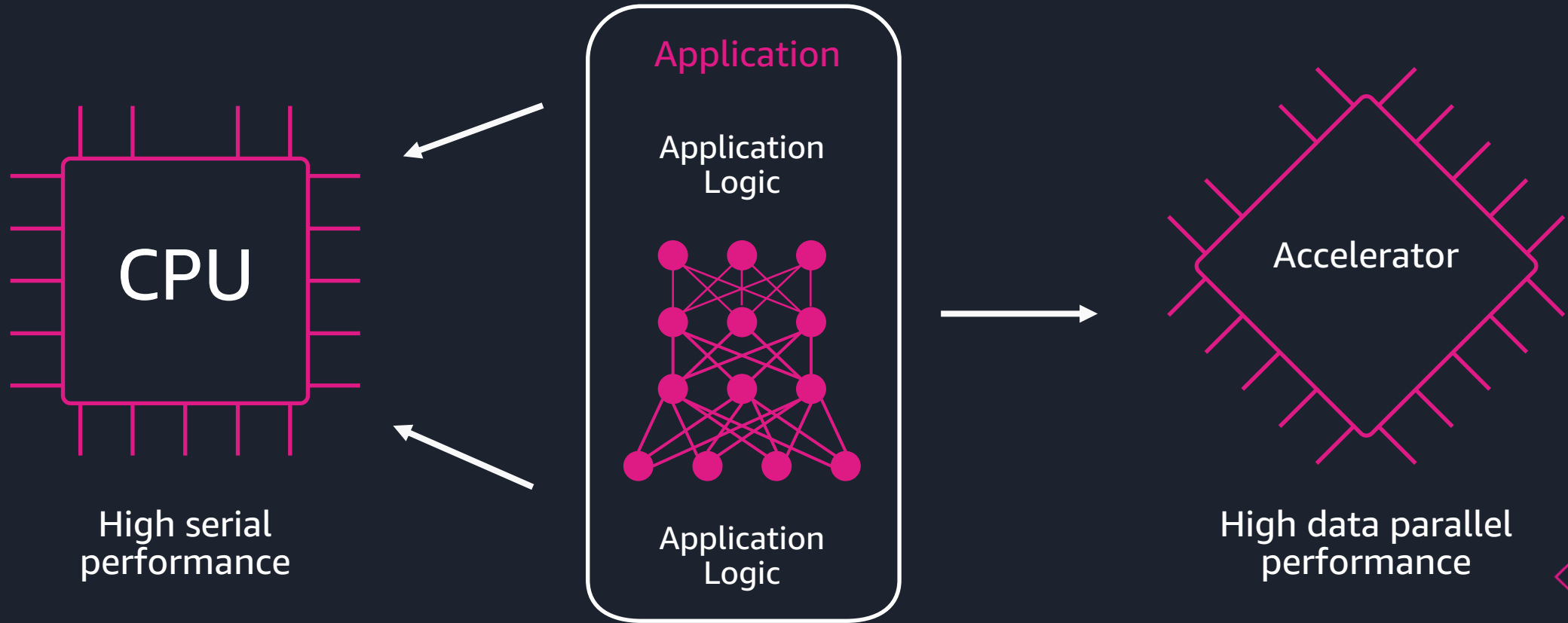


?



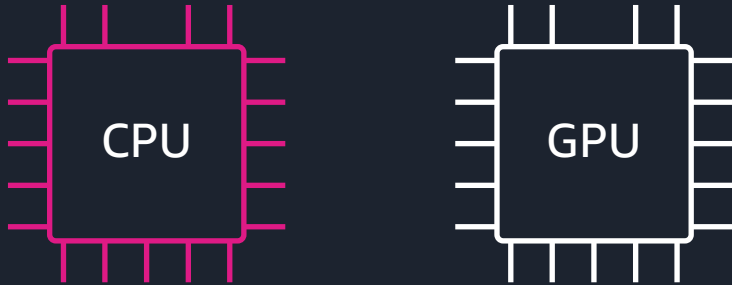
?

# How AI hardware accelerators work

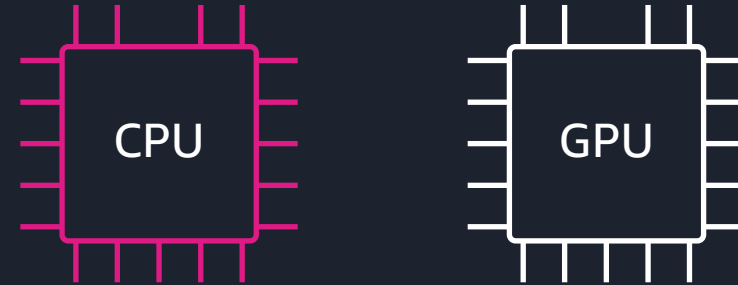


# Hardware needs for training and inference are different

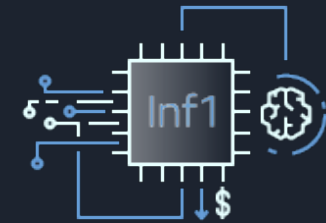
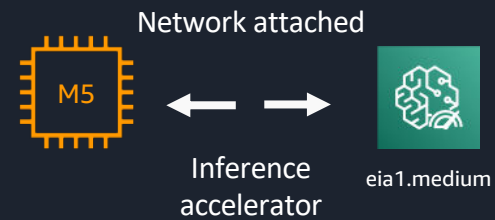
## TRAINING



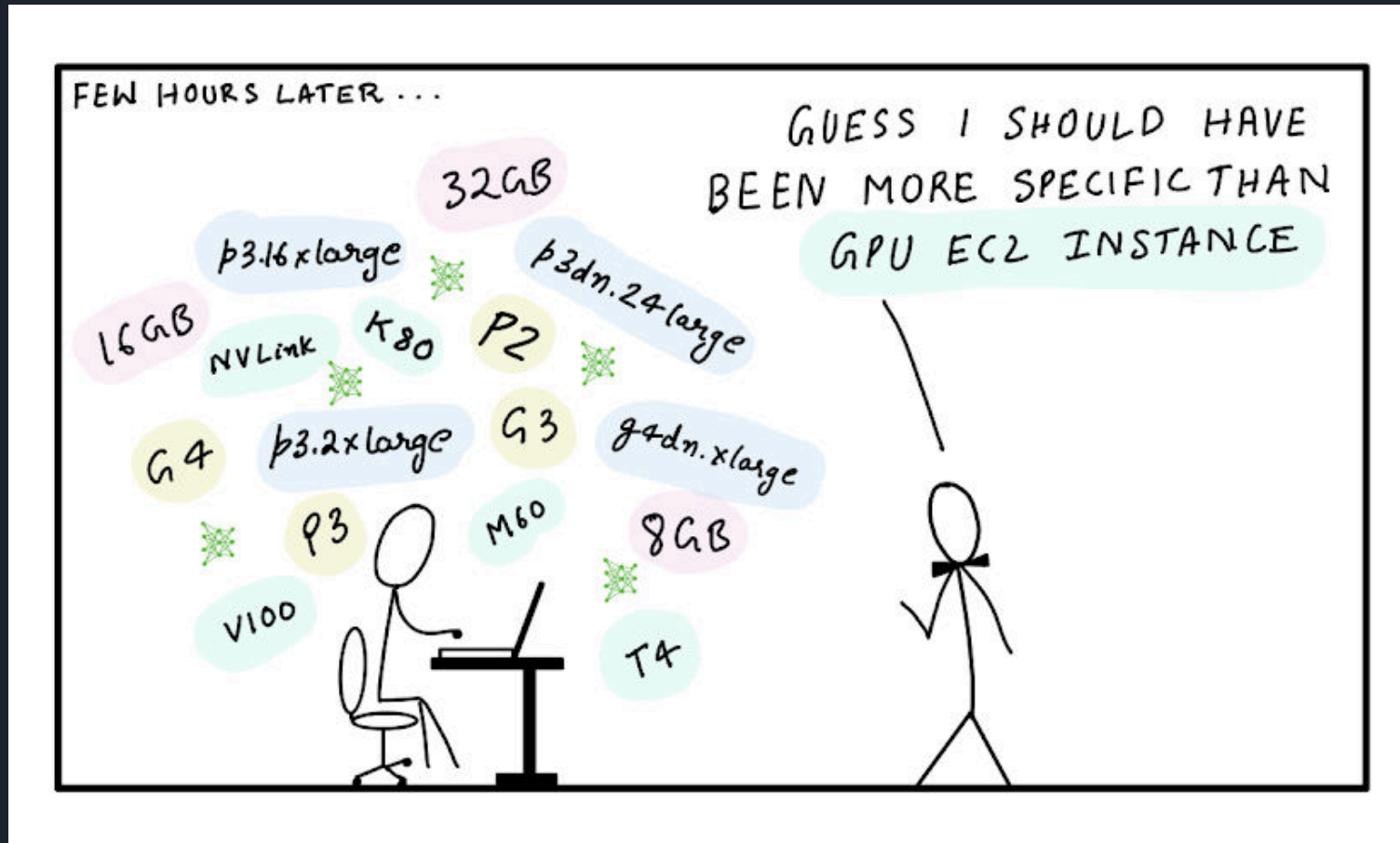
## INFERENCE DEPLOYMENT



### Elastic Inference



# Choosing instances for training - where do you start?

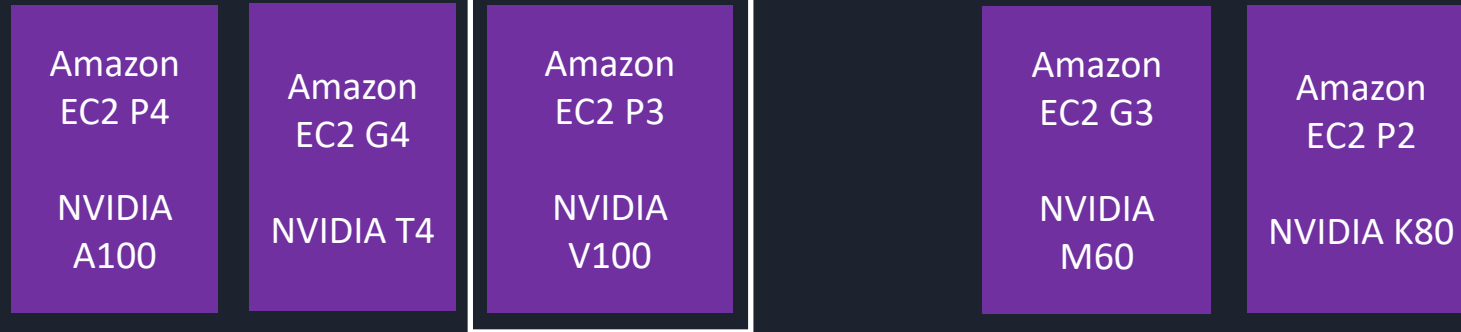


# GPUs for deep learning training in the cloud

NVIDIA GPU Architecture



Amazon EC2 instances



NVIDIA GPU

# Amazon EC2 P3 instance family at a glance

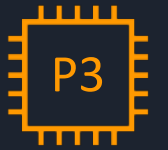
- GPU Generation: NVIDIA Volta
- Supported precision types: FP64, FP32, FP16, Tensor Cores (mixed-precision)
- GPU memory: 16 GB, 32 GB only on p3dn.24xlarge
- GPU interconnect: NVLink high-bandwidth interconnect, 2nd generation

## Single GPU Instance:

- p3.2xlarge

## Multi-GPU Instances:

- p3.8xlarge (4 GPUs)
- p3.16xlarge (8 GPUs)
- p3dn.24xlarge (8 GPUs, 32 GB version)



# How do I choose the right P3 instance size?

	P3.2xlarge	P3.8xlarge	P3.16xlarge	P3dn.24xlarge
GPUs	1 x V100	4 x V100	8 x V100	8 x V100
GPU memory	16 GB / GPU	16 GB / GPU	16 GB / GPU	32 GB / GPU
vCPUs	8	32	64	96
Mem	61	244	488	768

Highest performance optimized for distributed training

- 32 GB memory
- 100 Gbps bandwidth

Distributing training and large-scale experiments

Distributing training and multiple experiments

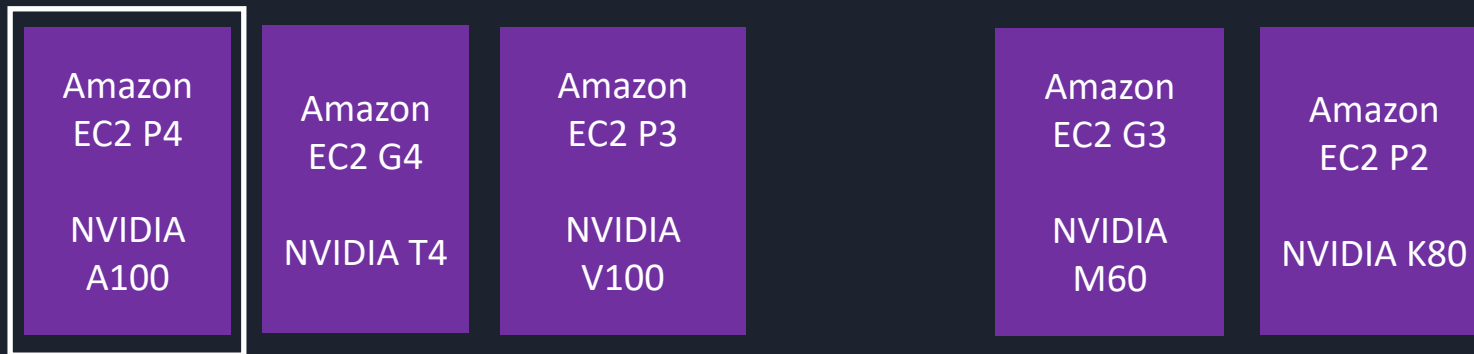
Local mode training and prototyping

# GPUs for deep learning training in the cloud

NVIDIA GPU Architecture



Amazon EC2 instances



NVIDIA GPU

# Amazon EC2 P3 instance family at a glance

- GPU Generation: NVIDIA Ampere
- Supported precision types: FP64, FP32, FP16, Tensor Cores (mixed-precision), Bfloat 16, TensorFloat-32
- GPU memory: 40 GB
- GPU interconnect: NVLink high-bandwidth interconnect, 3rd generation

## Single GPU Instance:

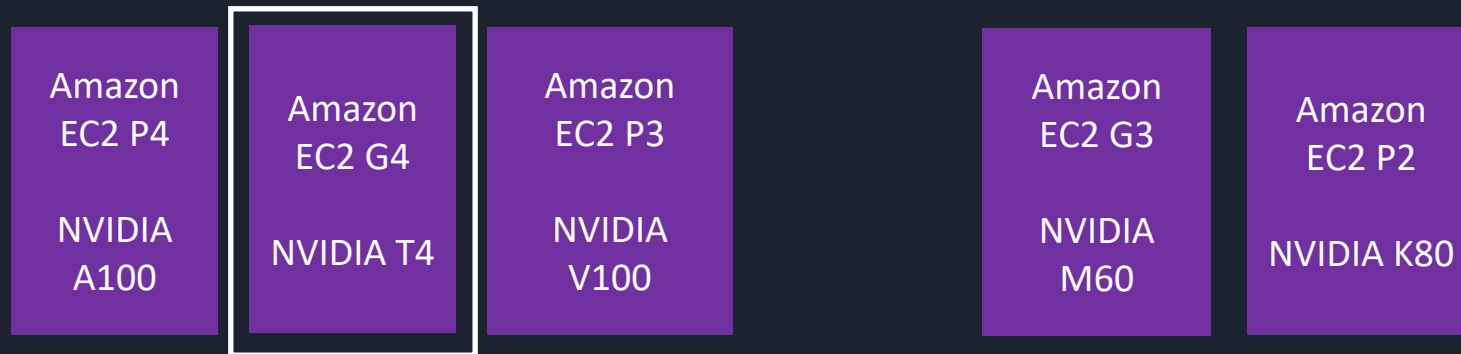
- p3.2xlarge

# GPUs for deep learning training in the cloud

NVIDIA GPU Architecture



Amazon EC2 instances



NVIDIA GPU

# Large training datasets: What are my options?



Fully managed and optimized Amazon SageMaker cluster



1. Moderate and large datasets



Amazon S3

**File mode:** Copy entire dataset to local volume

**Pipe mode:** Stream dataset from Amazon S3

2. Scalable shared file system



Amazon EFS

No downloading or streaming  
Share file system with other services

3. High-performance file system



FSx for Lustre file system

Optimized for high-performance computing  
Natively integrated with Amazon S3

# DEMO: GPU-accelerated training

## 1. Update Training Script

```
if __name__ == "__main__":  
  
    # Change: Update script to accept hyperparameters as command line arguments  
    parser = argparse.ArgumentParser()  
  
    # Hyper-parameters  
    parser.add_argument('--epochs', type=int, default=15)  
    parser.add_argument('--learning-rate', type=float, default=0.001)  
    parser.add_argument('--batch-size', type=int, default=256)  
    parser.add_argument('--weight-decay', type=float, default=2e-4)  
    parser.add_argument('--momentum', type=float, default=0.9)  
    parser.add_argument('--optimizer', type=str, default='adam')  
    parser.add_argument('--model-type', type=str, default='resnet')  
  
    # SageMaker parameters  
    parser.add_argument('--model_dir', type=str)  
  
    # Data directories and other options  
    parser.add_argument('--train', type=str, default=os.environ['SM_CHANNEL_TRAIN'])  
    parser.add_argument('--validation', type=str, default=os.environ['SM_CHANNEL_VALIDATION'])  
    parser.add_argument('--eval', type=str, default=os.environ['SM_CHANNEL_EVAL'])  
  
    args = parser.parse_args()  
  
    main(args)
```

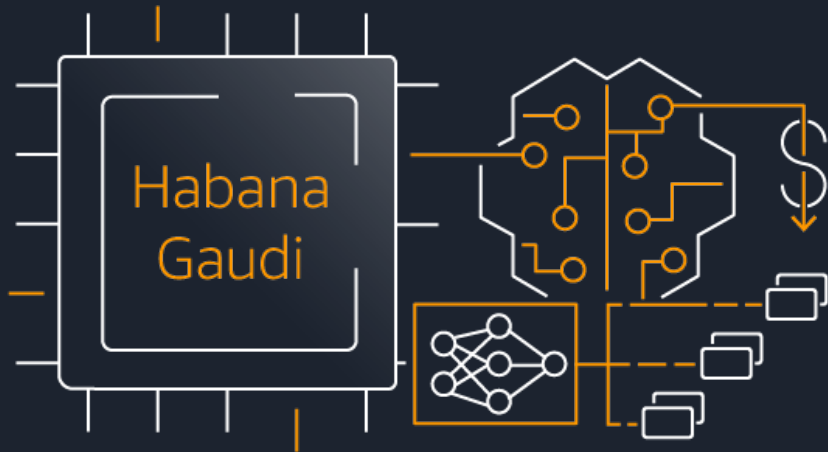
## 2. Submit Training Job

```
job_name = f'tf-single-gpu-{time.strftime("%Y-%m-%d-%H-%M-%S-%j", time.gmtime())}'  
output_path = f's3://{bucket_name}/{jobs_folder}'  
  
metric_definitions = [{'Name': 'Validation Accuracy', 'Regex': 'Validation Accuracy: ([0-9\\.]+)'}]  
  
hyperparameters = {'epochs': 50,  
                   'learning-rate': 0.01,  
                   'momentum': 0.95,  
                   'weight-decay': 2e-4,  
                   'optimizer': 'adam',  
                   'batch-size': 256,  
                   'model-type': 'custom'}  
  
from sagemaker.tensorflow import TensorFlow  
smdp_estimator = TensorFlow(entry_point = 'cifar10-tf2.py',  
                           source_dir = 'code',  
                           output_path = output_path + '/',  
                           code_location = output_path,  
                           role = role,  
                           instance_count = 1,  
                           instance_type = 'ml.p3.2xlarge',  
                           framework_version = '2.3.1',  
                           py_version = 'py37',  
                           metric_definitions = metric_definitions,  
                           hyperparameters = hyperparameters)
```

# Coming in 2021

## Habana Gaudi-based instances

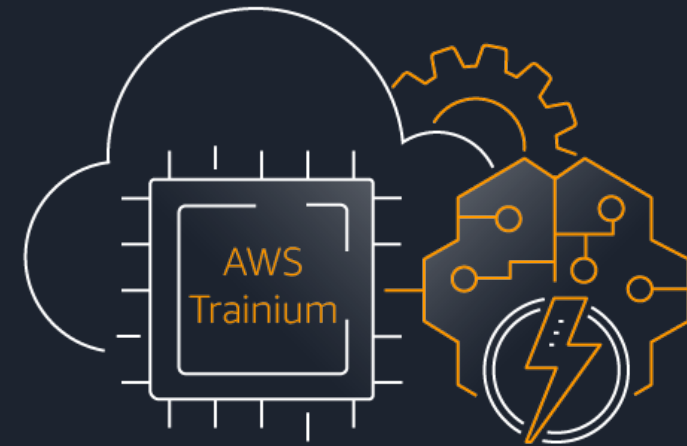
EC2 instances powered by new Habana Gaudi processors from Intel



[aws.amazon.com/ec2/instance-types/habana-gaudi/](https://aws.amazon.com/ec2/instance-types/habana-gaudi/)

## AWS Trainium

High performance machine learning training chip, custom designed by AWS

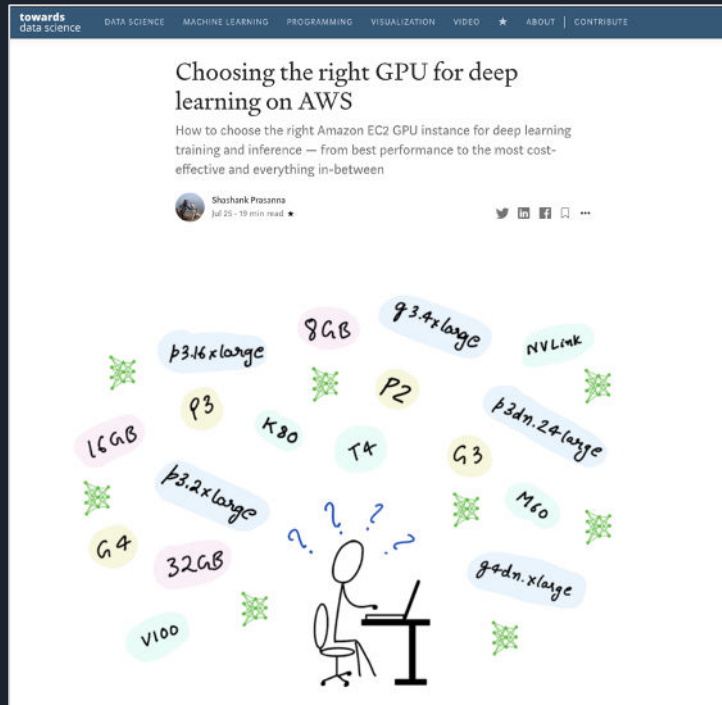


[aws.amazon.com/machine-learning/trainium/](https://aws.amazon.com/machine-learning/trainium/)

Architecture	NVIDIA GPU	Instance type and family	Instance size	Number of GPUs	GPU Interconnect (NVLink / PCIe)	GPU Memory	Tensor Cores (mixed-precision)	Precision: FP64	Precision: FP32	Precision: FP16	Precision: INT8	Precision Bfloat (BF16)	Precision: TensorFloat-32 (TF32)	Nitro based
Ampere	A100	P4	p4d.24xlarge	8	NVLink gen 3 (600 GB/s)	40 GB	Tensor Cores gen 3	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Volta	V100	P3	p3.2xlarge	1	NA	16 GB	Tensor Cores gen 2	Yes	Yes	Yes	Yes	No	No	No
	V100	P3	p3.8xlarge	4	NVLink gen 2 (300 GB/s)	16 GB	Tensor Cores gen 2	Yes	Yes	Yes	Yes	No	No	No
	V100	P3	p3.16xlarge	8	NVLink gen 2 (300 GB/s)	16 GB	Tensor Cores gen 2	Yes	Yes	Yes	Yes	No	No	No
	V100 (32 GB)	P3	p3dn.24xlarge	8	NVLink gen 2 (300 GB/s)	32 GB	Tensor Cores gen 2	Yes	Yes	Yes	Yes	No	No	Yes
Turing	T4	G4	g4dn.xlarge	1	PCIe	16 GB	Tensor Cores gen 2	No	Yes	Yes	Yes	No	No	Yes
	T4	G4	g4dn.2xlarge	1	PCIe	16 GB	Tensor Cores gen 2	No	Yes	Yes	Yes	No	No	Yes
	T4	G4	g4dn.4xlarge	1	PCIe	16 GB	Tensor Cores gen 2	No	Yes	Yes	Yes	No	No	Yes
	T4	G4	g4dn.8xlarge	1	PCIe	16 GB	Tensor Cores gen 2	No	Yes	Yes	Yes	No	No	Yes
	T4	G4	g4dn.16xlarge	1	PCIe	16 GB	Tensor Cores gen 2	No	Yes	Yes	Yes	No	No	Yes
	T4	G4	g4dn.12xlarge	4	PCIe	16 GB	Tensor Cores gen 2	No	Yes	Yes	Yes	No	No	Yes
	T4	G4	g4dn.metal	8	PCIe	16 GB	Tensor Cores gen 2	No	Yes	Yes	Yes	No	No	Yes
Kepler	K80	P2	p2.xlarge	1	NA	12 GB	No	Yes	Yes	No	No	No	No	No
	K80	P2	p2.8xlarge	8	PCIe	12 GB	No	Yes	Yes	No	No	No	No	No
	K80	P2	p2.16xlarge	16	PCIe	12 GB	No	Yes	Yes	No	No	No	No	No
Maxwell	M60	G3	g3s.xlarge	1	PCIe	8 GB	No	No	Yes	No	No	No	No	No
	M60	G3	g3.4xlarge	1	PCIe	8 GB	No	No	Yes	No	No	No	No	No
	M60	G3	g3.8xlarge	2	PCIe	8 GB	No	No	Yes	No	No	No	No	No
	M60	G3	g3.16xlarge	4	PCIe	8 GB	No	No	Yes	No	No	No	No	No

All GPU Amazon EC2 instances and features at a glance

# Blog post: Choosing the right GPUs for deep learning on AWS



## Blog post

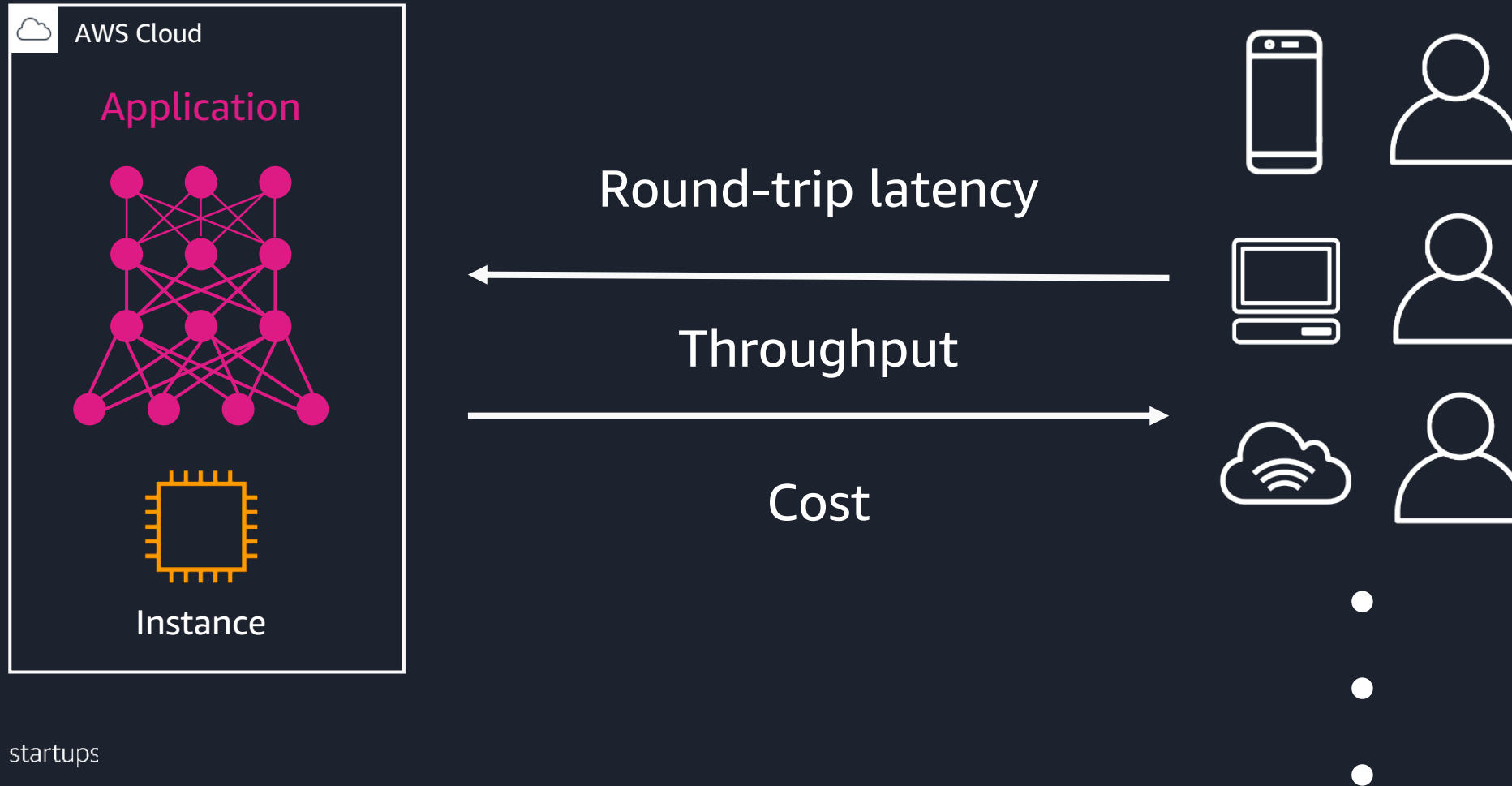
<https://towardsdatascience.com/choosing-the-right-gpu-for-deep-learning-on-aws-d69c157d8c86>

# What are we going to talk about today?

Accelerate ML  
development with  
training hardware

Accelerate ML  
deployment with  
inference hardware

# Inference performance affects customer experience



# Instances and accelerators for ML inference

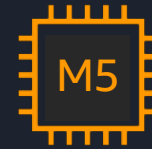
CPU instances



GPU instances



Elastic Inference



Network attached

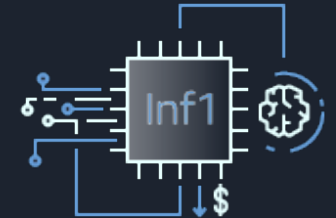


Inference accelerator



eia1.medium

AWS Inferentia



1. Target performance

3. Model and framework support

2. Cost efficiency

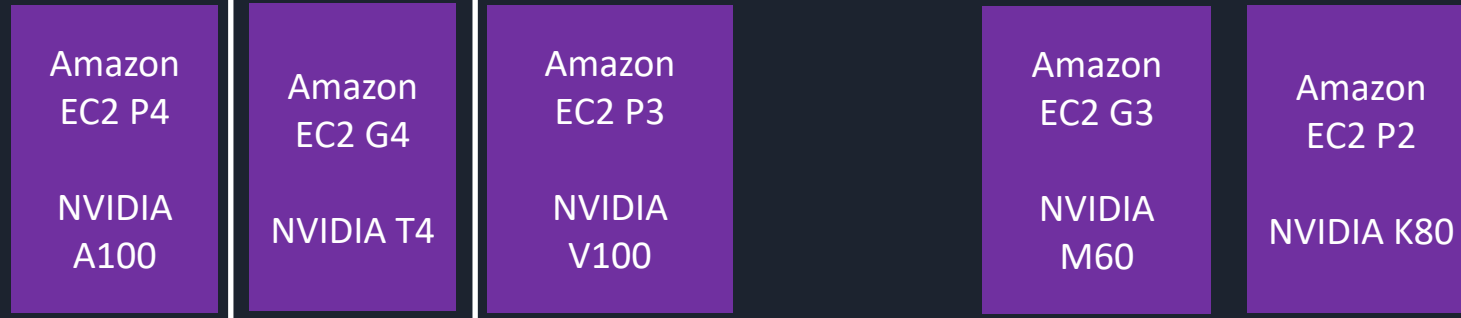
4. Ease of use and deployment

# GPUs for deep learning training in the cloud

NVIDIA GPU Architecture



Amazon EC2 instances



NVIDIA GPU

# Amazon EC2 G4 instance family at a glance

- GPU memory: 16 GB  
Supported precision types:  
FP32, FP16, INT8, Tensor Cores (mixed-precision)
- **Performance:** GPUs are throughput processors, and can deliver very throughput at desired latency
- **Cost:** GPUs may be under-utilized for small model, small batch, or sporadic inference request (fluctuating customer demand)
- **Ease of use:** Native GPU acceleration on all popular deep learning frameworks

## Single GPU Instance

- g4dn.xlarge
- g4dn.2xlarge
- g4dn.4xlarge
- g4dn.8xlarge
- g4dn.16xlarge

## Multi-GPU Instances

- g4dn.12xlarge (4 GPUs)
- g4dn.metal (8 GPUs)

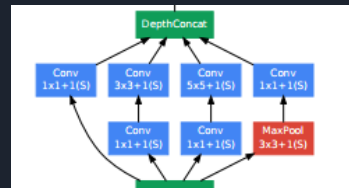
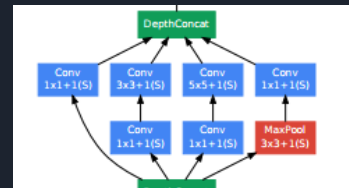
# GPU key strength: Programmability

GPUs are usually first to get support for novel models and operators  
You can write custom GPU accelerated model using NVIDIA CUDA

CPU

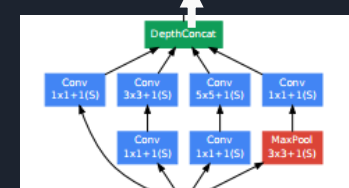
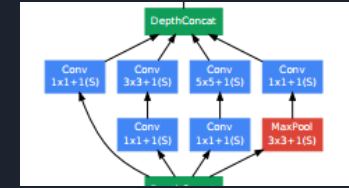
GPU

Custom ops  
written in  
Python



GPU

Custom ops  
written in  
CUDA



# Demo: Inference deployment with Amazon EC2 G4 instances

**Problem** – Classification (1,000 classes)

**Dataset** – ImageNet2012 validation dataset with 50,000 images

**Model** – Resnet50 (expected top-1 accuracy on test set – ~74.5%)

## Approach

Measure **throughput**, **latency**, and **accuracy** for

- GPU-accelerated inference with TensorFlow / Keras
- GPU-accelerated inference with NVIDIA TensorRT optimizations
- Host a GPU-accelerated inference endpoint with Amazon SageMaker hosting

# Instances and accelerators for ML inference

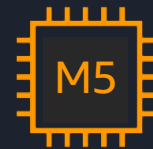
CPU instances



GPU instances



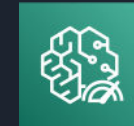
Elastic Inference



Network attached

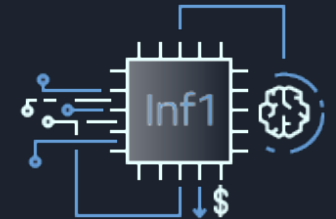


Inference accelerator



eia1.medium

AWS Inferentia



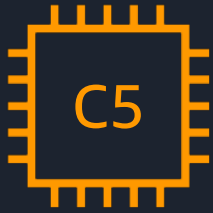
1. Target performance

3. Model and framework support

2. Cost efficiency

4. Ease of use and deployment

# What if you can't maximize GPU utilization?



Lower compute power

- Low cost / inference for
- Small DL models
  - Traditional ML models

What about?

Mid-sized models

Need acceleration but not a dedicated GPU

Lower throughput and higher latency tolerance

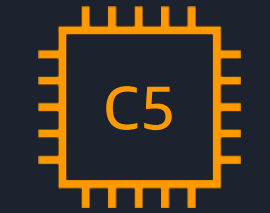
Cost sensitive



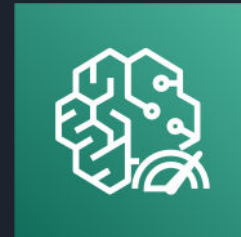
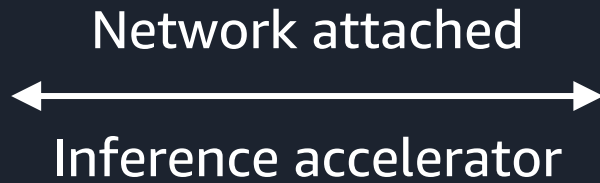
Higher compute power

- Low cost / inference for
- Large DL models
  - Large batch sizes
  - High demand

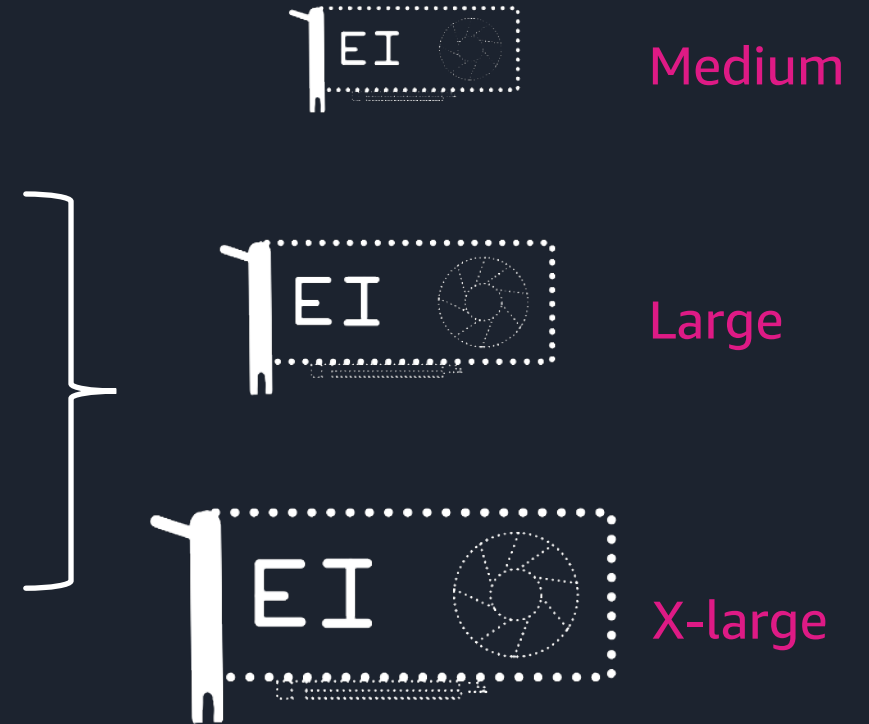
# Lower cost with access to variable-size GPU acceleration



CPU instance



eia1.medium



Variable-size GPU acceleration



# Elastic Inference accelerators at a glance

EI accelerator memory – 2 GB, 4 GB, 8 GB

EI accelerator (in TFLOPS)

- 1, 2, 4 (FP32)
- 8, 16, 32 (FP16)

**Performance:** Amazon EI delivers lower performance than a dedicated GPU instance, but at much lower cost

**Cost:** Amazon EI can save you up to 75% in inference cost vs. GPU;

**Ease of use:** EI enabled frameworks make it easy to deploy with almost no code changes

## EIA1 family

- eia1.medium
- eia1.large
- eia1.xlarge

## EIA2 family

- eia2.medium
- eia2.large
- eia2.xlarge

# Demo: Inference deployment with EI accelerators

**Problem** – Classification (1,000 classes)

**Dataset** – ImageNet2012 validation dataset with 50,000 images

**Model** – Resnet50 (expected top-1 accuracy on test set – ~74.5%)

## Approach

Measure throughput, latency, and accuracy for

1. CPU-only inference with TensorFlow / Keras
2. EI accelerated inference with EI-enabled TensorFlow / Keras
3. Host a EI accelerated inference endpoint with Amazon SageMaker hosting

# Instances and accelerators for ML inference

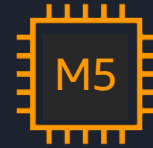
CPU instances



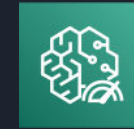
GPU instances



Elastic Inference

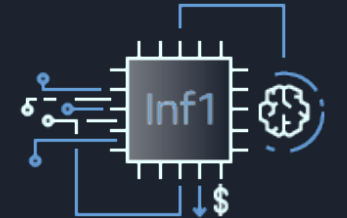


Network attached  
↔  
Inference accelerator



eia1.medium

AWS Inferentia



1. Target performance

3. Model and framework support

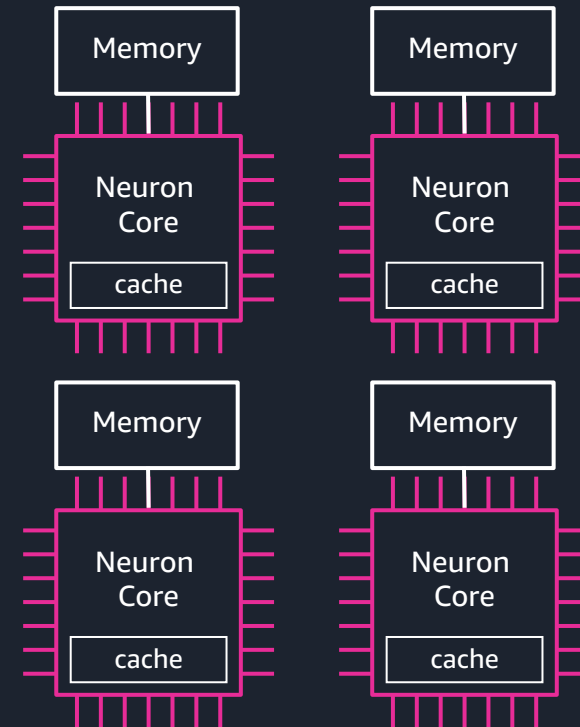
2. Cost efficiency

4. Ease of use and deployment

# AWS Inferentia: Custom silicon for ML inference

- 4 NeuronCores
- Up to 128 TOPS
- 2-stage memory hierarchy  
Large on-chip cache and commodity DRAM
- Supports FP16, BF16, INT8 data types with mixed precision
- Fast chip-to-chip interconnect

## AWS Inferentia



# Amazon EC2 Inf1 instance family at a glance

Accelerators – 1–16 AWS Inferentia chips

Cores – 4–64 NeuronCores

Up to 192 GiB of Memory

Up to 100 Gbps networking bandwidth

**Performance:** Amazon Inf1 instances with AWS Inferentia can deliver high throughput and at lower cost compared to GPUs

**Cost:** Inf1 instances delivers lower cost vs. GPU for popular models: YOLOv4, OpenPose, BERT and SSD

**Ease of use:** AWS Neuron SDK offers a compiler and runtime as well as profiling tools

Single Inferentia chip instance

- inf1.xlarge
- inf1.2xlarge

Multi-Inferentia chip Instances

- inf1.6xlarge (4 chips)
- inf1.24xlarge (16 chips)

# Demo: Inference deployment with Inf1 instances and AWS Inferentia

**Problem** – Classification (1,000 classes)

**Dataset** – ImageNet2012 validation dataset with 50,000 images

**Model** – Resnet50 (expected top-1 accuracy on test set: ~74.5%)

## Approach

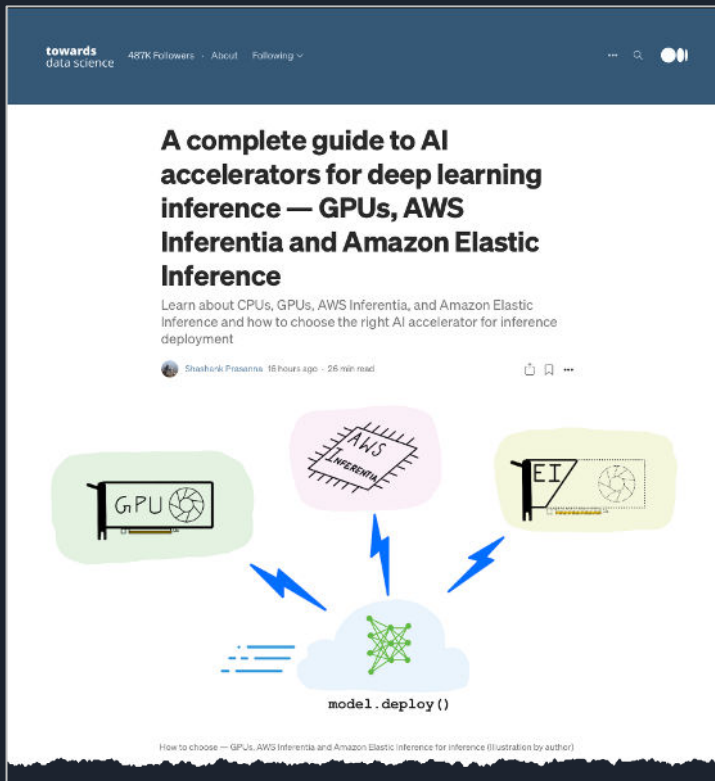
Measure throughput, latency, and accuracy

- Batch size = 1 and number of cores = 1
- Batch size = 5 and number of cores = 1
- Batch size = 1 and number of cores = 4

# Instances and accelerators for ML inference

Inference accelerator instance type	Throughput	Latency	Cost efficiency	Model support and programmability	Ease of use	Framework support
CPU-only C4, C5 instance types	○	○	● Smaller models	●	●	●
GPU G4 instance type	●	●	● High-utilization	●	◐	●
Elastic Inference CPU instances + EI accelerator	◐	◐	●	◐	◐	◐
AWS Inferentia Inf1 instance type	●	●	●	◐	◐	◐

# Blog post: Complete guide to AI accelerators on AWS



## Blog post

<https://towardsdatascience.com/a-complete-guide-to-ai-accelerators-for-deep-learning-inference-gpus-aws-inferentia-and-amazon-7a5d6804ef1c>

## Examples

<https://github.com/shashankprasanna/ai-accelerators-examples>





# Thank you

Twitter: @shshnkp

LinkedIn: [linkedin.com/in/shashankprasanna](https://www.linkedin.com/in/shashankprasanna)

Medium: [medium.com/@shashankprasanna](https://medium.com/@shashankprasanna)

