STP201

Modular

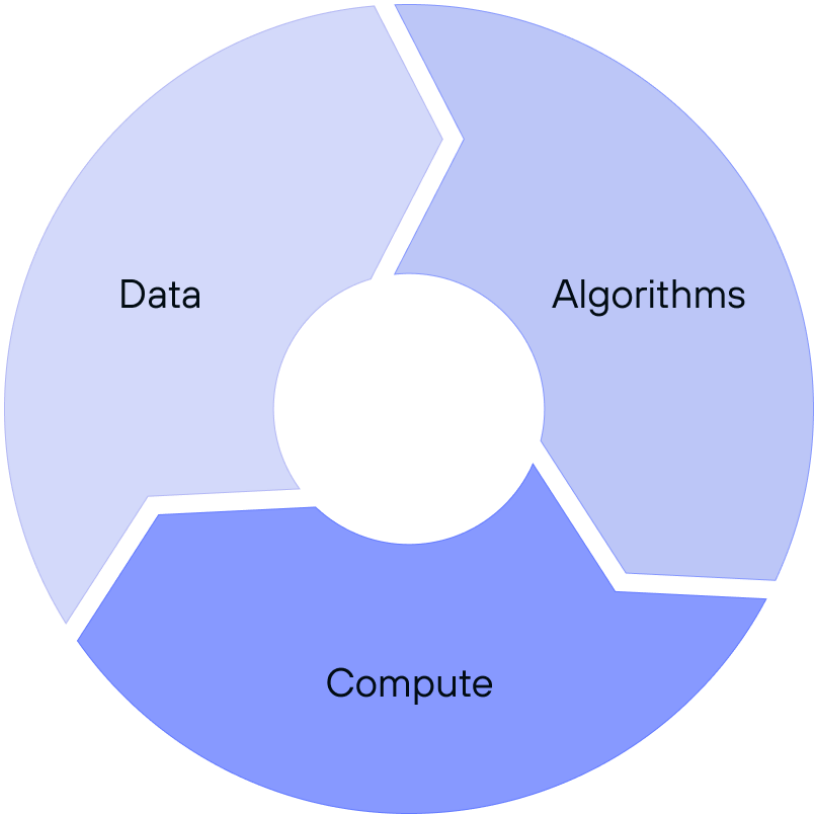
Accelerating the pace of AI

“Generative AI is poised to unleash the next wave of productivity.”

MCKINSEY 2023 AI ECONOMIC REPORT

But how did we get here?

AI is a flywheel



Let’s take a quick look through history

Pre-Internet

Before 2000

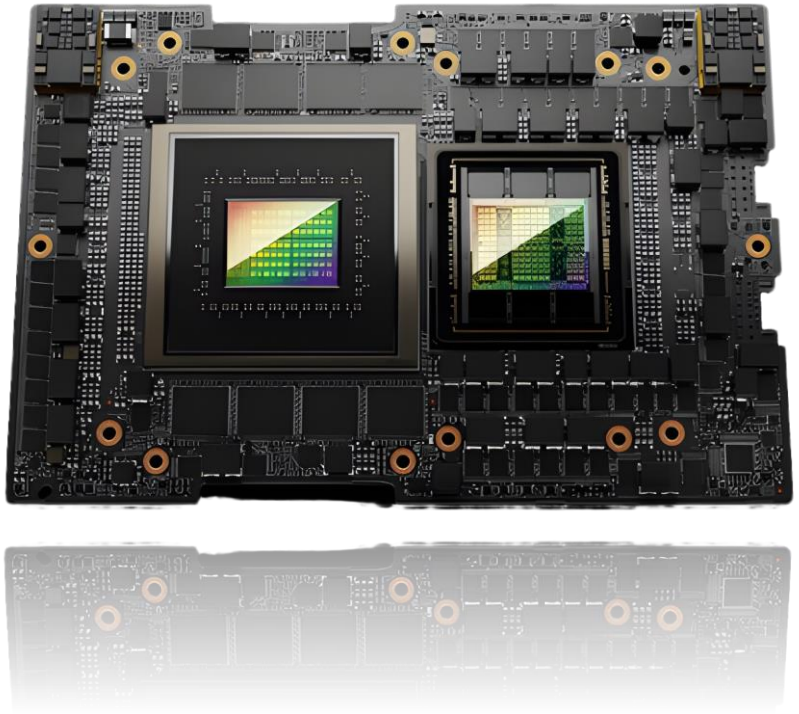
Traditional computing relied on single-threaded CPU performance and programming languages like C & C++.

Internet + Mobile

2000-2010

Rise of the Internet, Mobile compute and the “Big Data” era prompted a shift towards cloud and **multi-core hardware**.

Huge numbers of CPUs were installed in massive data centers.



Dawn of AI

2010-2021

AI started to exploded onto the scene, deep learning revolution began.

Enormous demand for parallel programming, and top tech companies heavily invest in AI ahead of the world.

GenAI Era

2022-Today

ChatGPT changed the world in 2022, despite being in BigTech for years.

Demand for GenAI compute, consumer and enterprise AI adoption sky rockets.

And today, NVIDIA powers AI

With 17 year old software called CUDA

In fact, most AI software used in production is now old and fragmented

And it's getting worse!

2015-17	2018-20	2021-Present
Model Training	Model Inference	GenAI & LLMs
Caffe	TensorRT OpenVINO	NVIDIA TensorRT-LLM text-generation-inference LLM DeepSpeed ggml gemma.cpp

And you might wonder, why?

Ensembles

MoEs

Flash Attention

Chain of Thought

Customized Operators

Mixed Precision

AI research is evolving too fast
Frameworks aren't keeping up

Complex Numerics

New Datatypes

LLMs

KV Caching

Multi-device compute

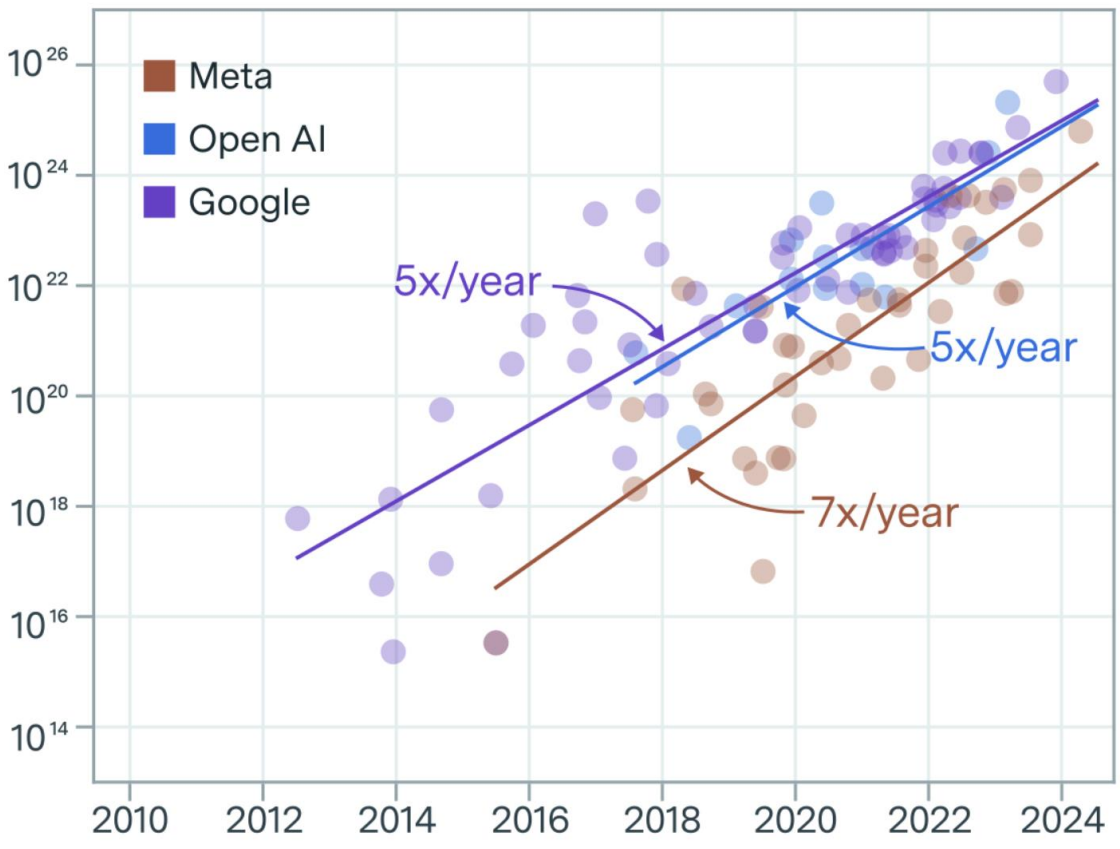
Multi-modal

None of these were designed for GenAI

Stack	Tools	Languages
DATA	SQL, SciPy, NumPy, pandas	Python
TRAINING	JAX, PyTorch, TensorFlow, XGBoost, scikit-learn	Python
INFERENCE	TensorRT, OpenVINO, ML.NET, Windows ML, ONNX	C++
SERVING	NVIDIA Triton Inference Server, TensorFlow Serving, TorchServe, mlflow, Kubernetes	C++

Only the largest and most sophisticated AI companies are leading model development

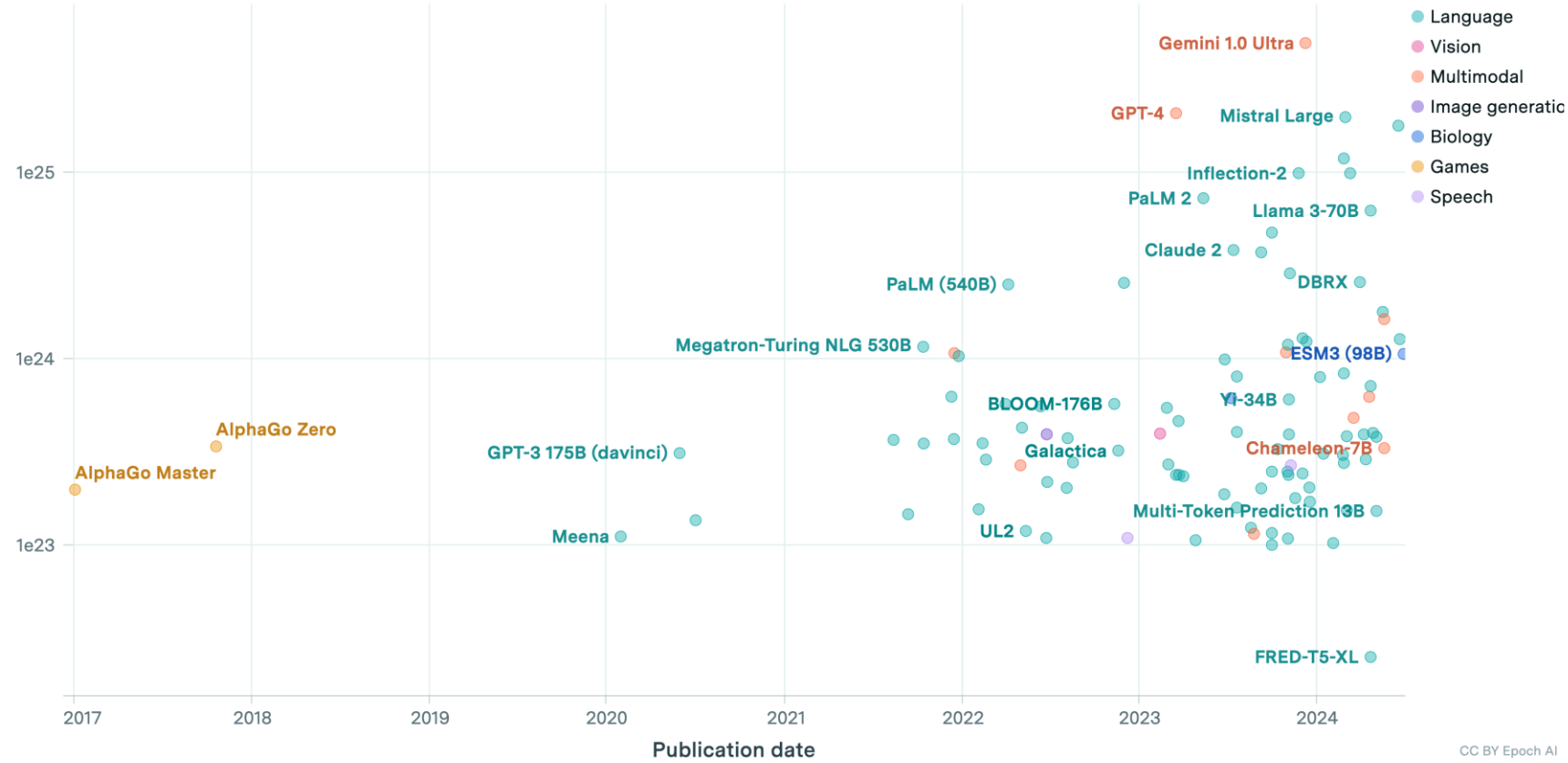
Training compute growing by 4-5x per year



Large-Scale AI Models

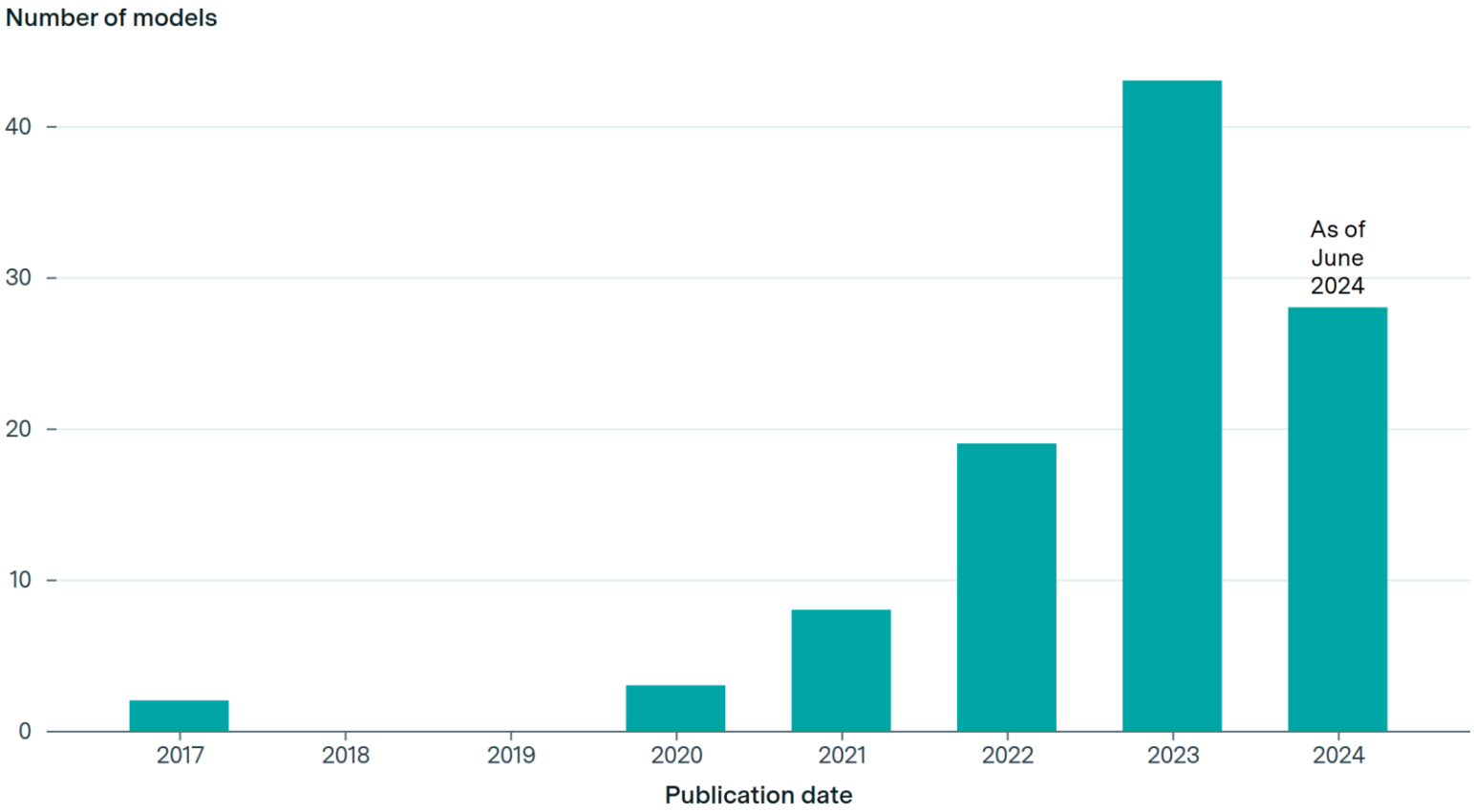
EPOCH AI

Training compute (FLOP)



Number of models larger than 10^{23} FLOP released by year

EPOCH AI



Compute demands continue to explode … for NVIDIA

1.35x/year

Computational performance

8x

Lower-precision number formats

1.2x/year

Memory capacity

1.18x/year

Memory Bandwidth

How can the rest of us keep up?

Right now, we're all struggling too

AI ENGINEER

MAIN MOTIVATION

Build at scale

1 New models & optimizations every week

2 Every product needs to be GenAI enhanced

3 No time to deal with new hardware

4 Costs prohibit graduating PoC’s into production

The result?

The industry has converged to endpoint solutions


OpenAI


Gemini


MISTRAL
AI_


Fireworks AI


together.ai


perplexity


Claude

Despite so many frontier models being incredible

Top model rankings

LMSYS Chatbot Arena

Rank	Model	Elo	Open
1	GPT-4o	1287	No
2	Claude 3.5	1271	No
3	Gemini-Advanced	1263	No
…			
11	Gemma2	1212	Yes
12	Llama3	1207	Yes
12	Nemotron	1205	Yes

https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard

Most enterprises don’t need SoTA reasoning & planning

At the cost of giving up data,
security, ownership, control
& more

We need a better way

The future of AI can’t be owned by a few

You want to …

… control your data & security

… control your data & security

… integrate into your own cloud VPC

… control your data & security

… integrate into your own cloud VPC

… customize your own model

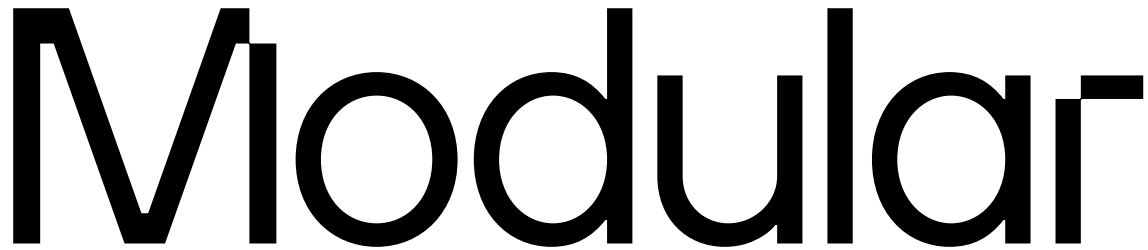
… control your data & security

… integrate into your own cloud VPC

… customize your own model

… save money on inference costs

It’s time to own your AI future


Modular

Who we are

Team with decades of experience from leading engineers who have built AI infrastructure across the current stack.

INFRASTRUCTUR	aws	Google Cloud	Meta
ML FRAMEWORKS			
RUNTIME		android	
COMPILERS			

We help you own, control, deploy GenAI into your products with ease

Our approach

1. Unify AI software with next generation infrastructure

2. Interop with existing software to make migration simple

3. Scale PoC’s to production faster

MAX

Modular Accelerated Xecution Platform

Framework

A new framework for Gen AI, best way to deploy PyTorch

Cloud

2025

Coming soon

Managed services to deploy MAX in the cloud

Framework

A new framework for Gen AI, best way to deploy PyTorch

Cloud

2025

Coming soon

Managed services to deploy MAX in the cloud

MAX

A new framework for Gen AI, and the best way to deploy PyTorch

SERVING LIBRARY

INFERENCE RUNTIME

MODEL PIPELINE

CLI

EXTEND EASILY IN



SCALE IN THE CLOUD

MAX
Framework

1. Works with PyTorch out of the box

2. Native MAX APIs unlock GenAI Inference

3. Easily scale across CPU+GPUs

Memory Planning

Layout Selection

Device Assignment

Compiler Fusion

Optimized Kernels

MAX brings the best innovations together into one place

Async Execution

Full User Extensibility

Constant Folding

Arithmetic Simplifications

Parametric Dynamic Shapes

Framework

A new framework for Gen AI, best way to deploy PyTorch

Cloud

Managed services to deploy MAX in the cloud

2025

Coming soon

Cloud

The easiest way to deploy MAX in the cloud across CPUs & GPUs

- OBSERVABILITY
- JOBS & WORKFLOWS
- PROVISIONING & SCALING
- BYOC

Why use MAX

1 **Performance**

2-5x over PyTorch out of the box, Setting new SoTA

2 **Portability**

Write once, deploy across CPUs and GPUs

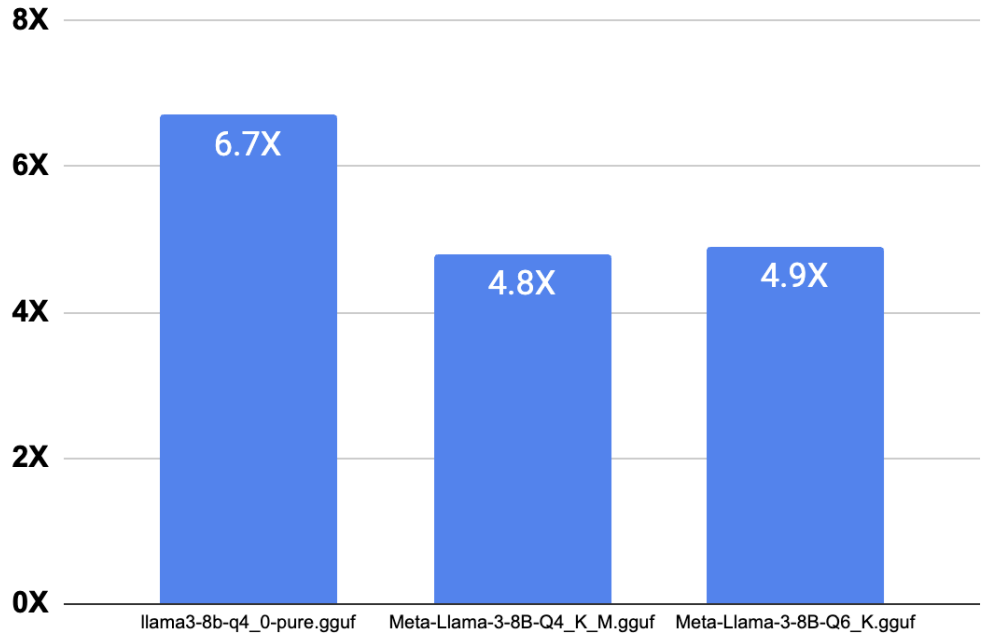
3 **Interoperability**

Meets you where you are (Python + PyTorch)

4 **Developer Velocity**

Unify local and cloud development, Move faster

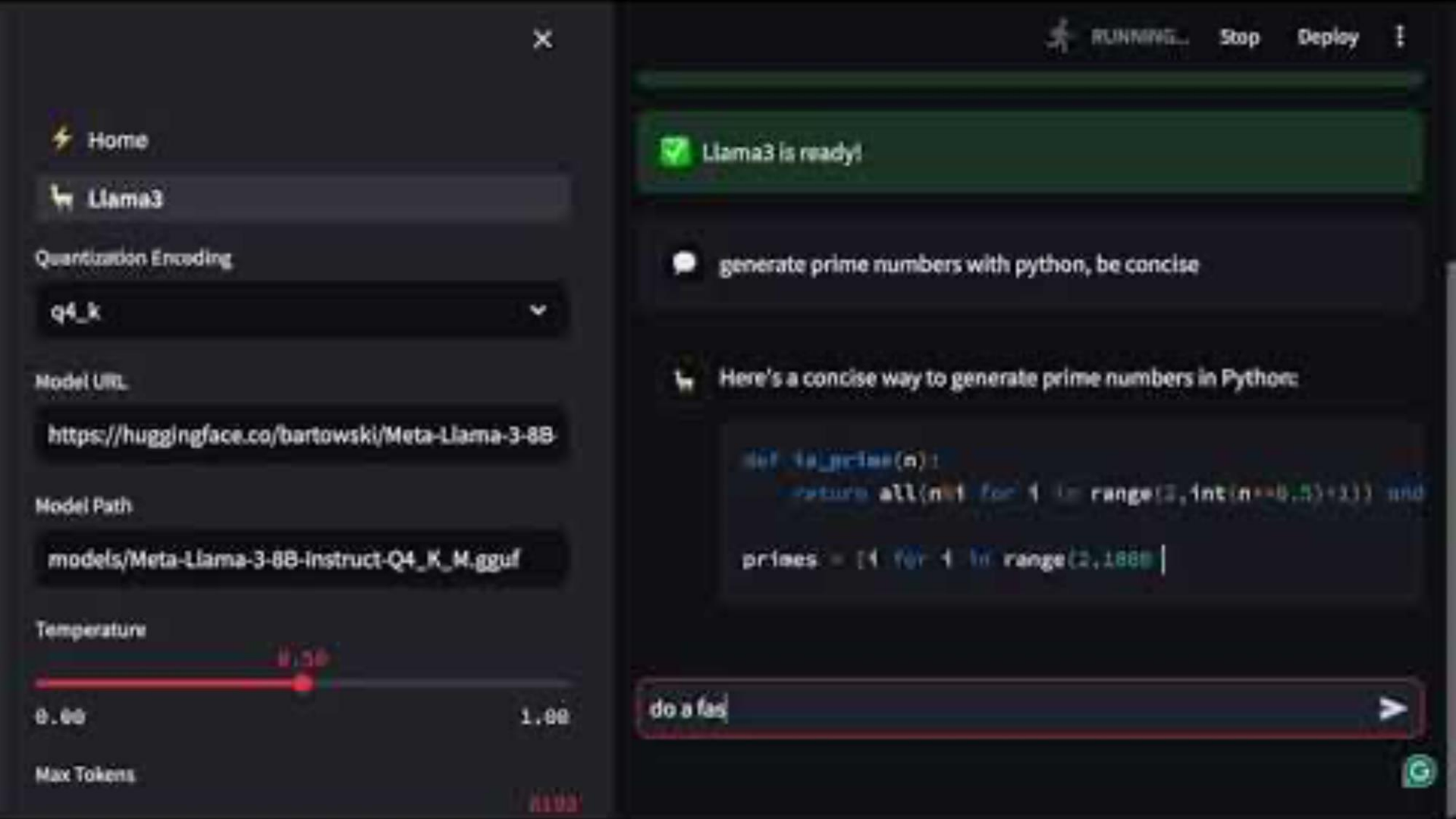
MAX Llama3 vs llama.cpp



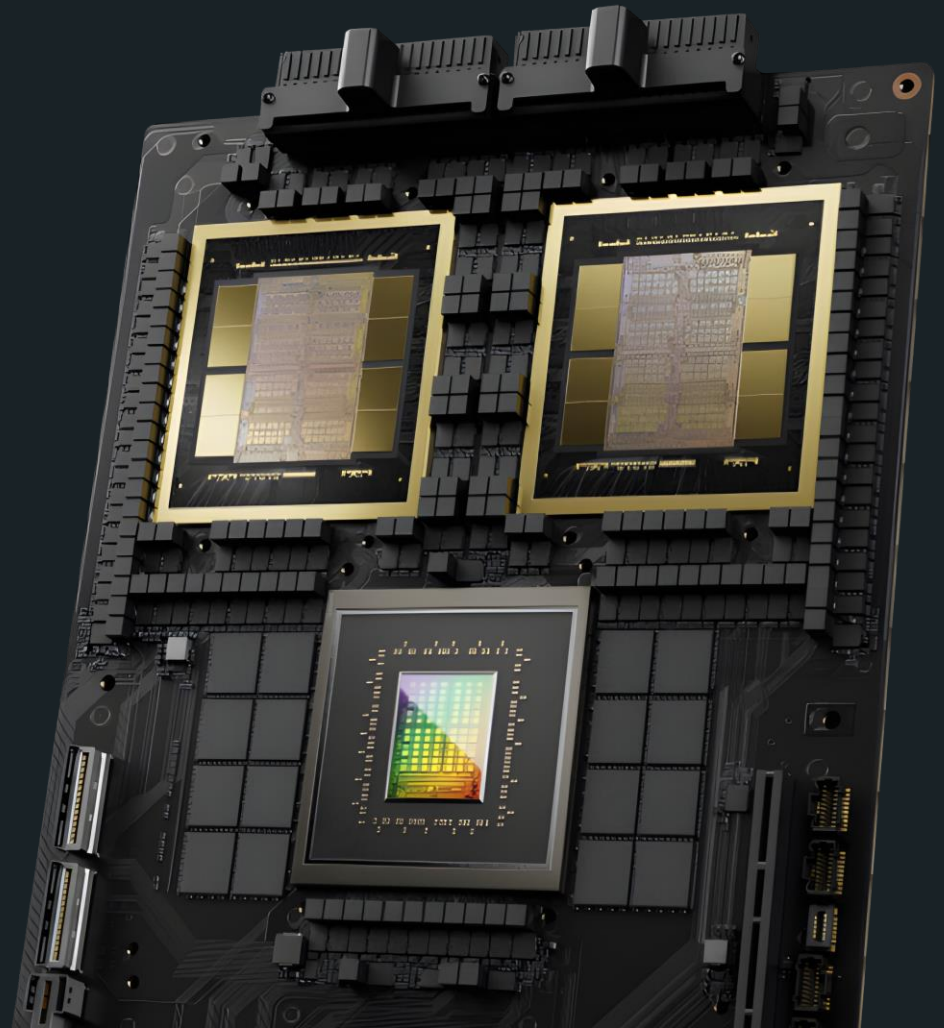
AWS c6i.16xlarge

Time to First Token, llama.cpp 2867; built with Intel(R) oneAPI DPC++/C++ Compiler 2024.0.2


Demo


RUNNING...
Stop
Deploy
Home
Llama3
Quantization Encoding
q4_k
Model URL
https://huggingface.co/bartowski/Meta-Llama-3-8B
Model Path
models/Meta-Llama-3-8B-Instruct-Q4_K_M.gguf
Temperature
0.00
1.00
Max Tokens
Llama3 is ready!
generate prime numbers with python, be concise
Here's a concise way to generate prime numbers in Python:
def is_prime(n):
return all(n%i for i in range(2,int(n**0.5)+1)) and
primes = [i for i in range(2,1000
do a fas

MAX GPU



NVIDIA

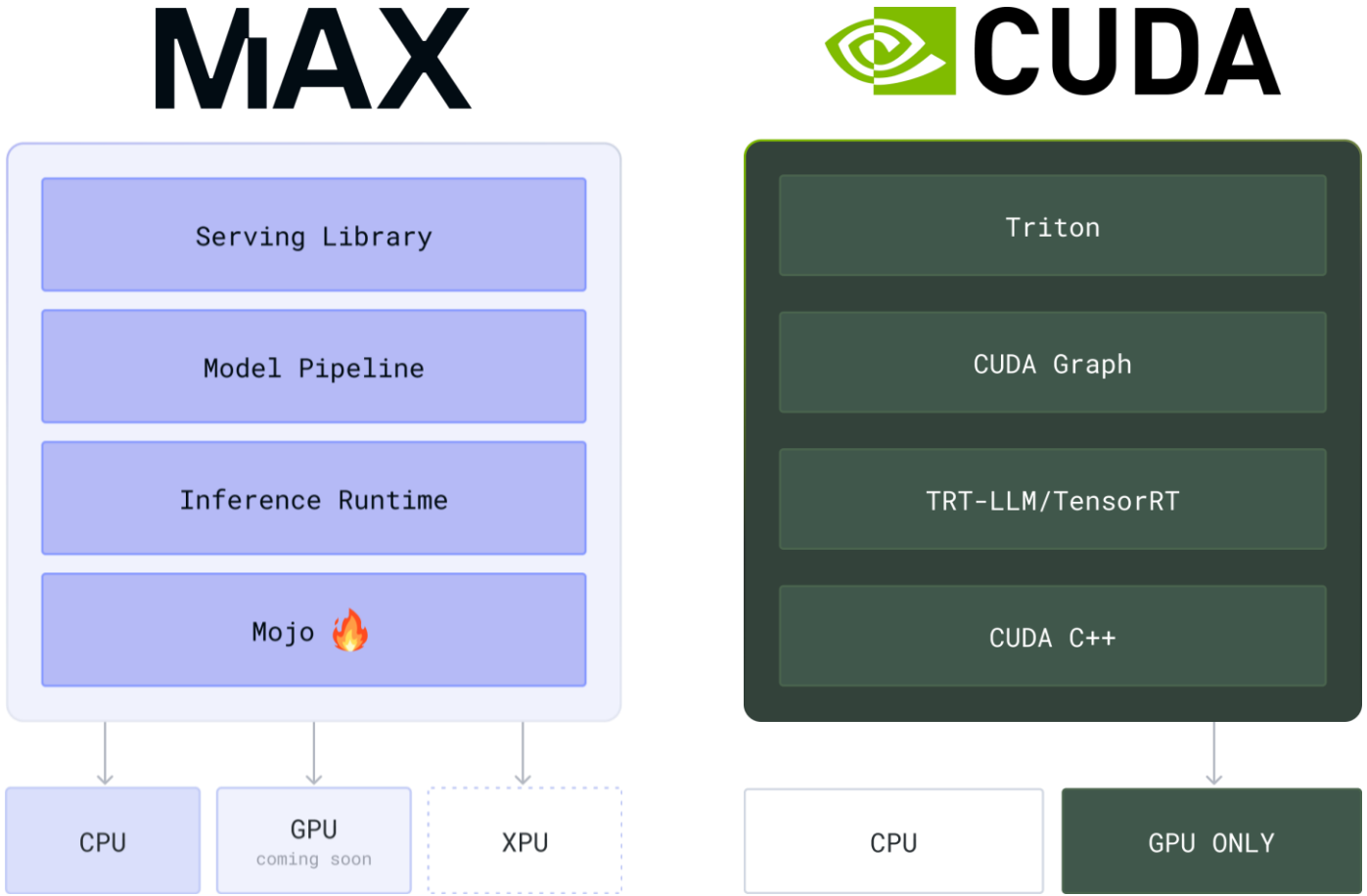
A better world for GPUs

1. You can program a GPU as easily as you program a CPU using Python

2. Your GenAI workloads achieve much higher utilization

3. You can move to other types of GPUs, enabling you to scale to **NVIDIA** and **AMD**

```
print("Building model...")
graph = model.build_graph("llama_model")

session_options = SessionOptions(
    cuda_device() if config.use_gpu else cpu_device()
)
session = InferenceSession(session_options)

compiled_model = compile_graph(session, graph)

generate_text(tokenizer, compiled_model, params, config, metrics)
```


MAX
Serving Library
Model Pipeline
Inference Runtime
Mojo 🔥
CPU
GPU
coming soon
XPU
CUDA
Triton
CUDA Graph
TRT-LLM/TensorRT
CUDA C++
CPU
GPU ONLY

MAX GPU

Full power of CUDA, much easier to use

- Autofusion for ease + performance

Exposes advanced GPU features:

- TensorCores, TMA, etc…

```
for i in range(16, K, 16):
    next_a_frag = a[i, aCol, layout=a_layout]
    next_b_frag = b[bRow, i, layout=b_layout]
    acc_frag += (await a_frag) @ (await b_frag)
    swap(a_frag, next_a_frag)
    swap(b_frag, next_b_frag)

acc_frag += (await a_frag) @ (await b_frag)
c.store(cRow, cCol, acc_frag, layout=c_layout)
```

TensorCores

MAX GPU

Full power of CUDA, much easier to use

- Autofusion for ease + performance

Exposes advanced GPU features:

- TensorCores, TMA, etc…

Easy and Pythonic:

- Predictable, scalable, debug-able
- Works with NVIDIA tools

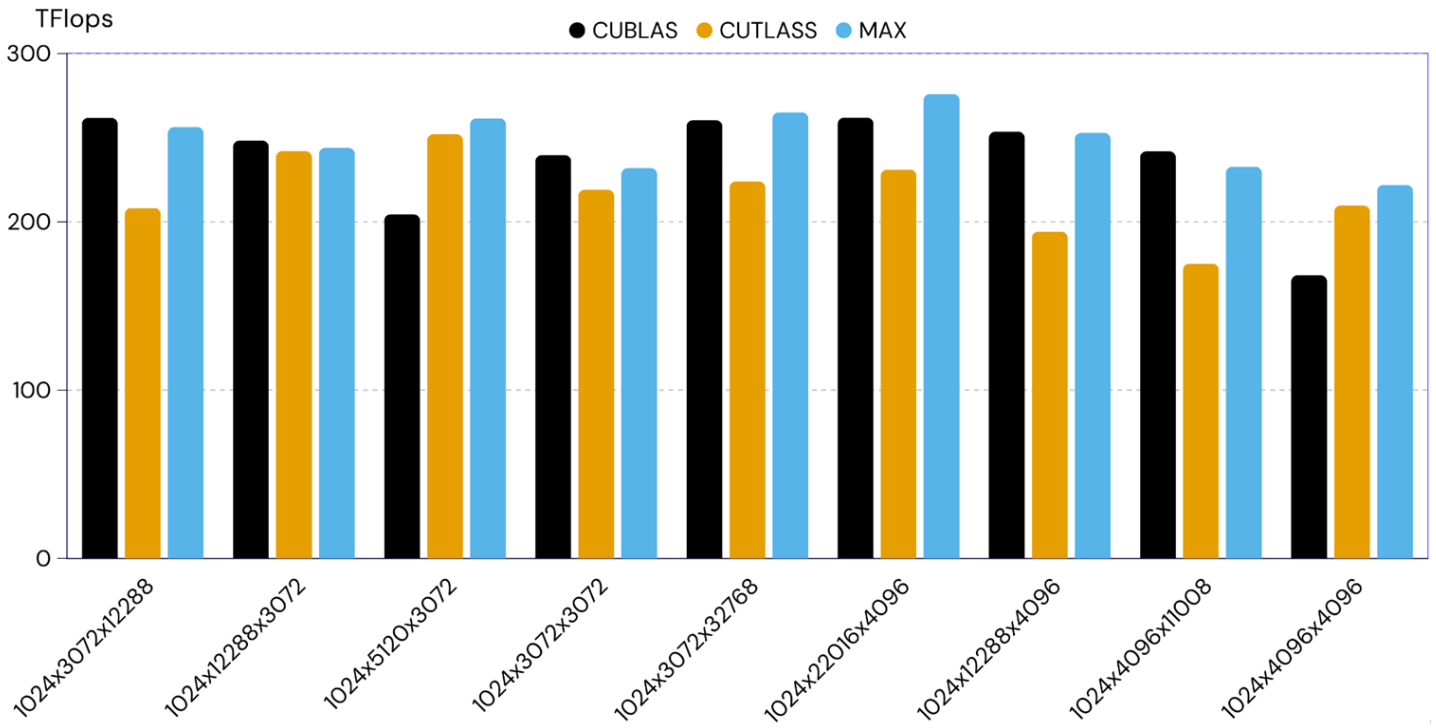
```
for i in range(16, K, 16):
    next_a_frag = a[i, aCol, layout=a_layout]
    next_b_frag = b[bRow, i, layout=b_layout]
    acc_frag += (await a_frag) @ (await b_frag)
    swap(a_frag, next_a_frag)
    swap(b_frag, next_b_frag)

acc_frag += (await a_frag) @ (await b_frag)
c.store(cRow, cCol, acc_frag, layout=c_layout)
```

TensorCores

But what about performance ...

MAX can meet & beat vendor performance

Matrix Multiplication Performance vs. NVIDIA



BF16 input/output F32 accum on A100 shapes from Llama + Replit models

MAX will provide you with GPU optionality

Sign up for early access to GPUs!

https://modular.com

Mojo🔥

The best way to extend Python
for CPU & GPU programming

What is Mojo 🔥?

Pythonic system programming language

- Full CLI toolchain + VS Code support

Freely available on Linux, Mac and Windows

- Progressively open sourcing

Vibrant and growing community:

- 200K+ users overall, 20K+ users on Discord

```
def mandelbrot_kernel[
    width: Int  # SIMD Width
](c: ComplexSIMD[float, width]) ->
   SIMD[int, width]:
   """A vectorized implementation of
   the inner mandelbrot computation."""
   z = ComplexSIMD[float, width](0, 0)

   iters = SIMD[DType.index, width](0)
   mask = SIMD[DType.bool, width](True)
   for i in range(MAX_ITERS):
     if not any(mask):
       break
     mask = z.squared_norm() <= 4
     iters = mask.select(iters + 1, iters)
     z = z.squared_add(c)
   return iters
```

Mojo 🔥 is fast!

100-1000x faster than Python!

Benefits:

- No need to learn C/C++
- Lower cost by default
- “Full stack” hackability
- Don’t split your eng team

Faster than Rust!

```
while merge_options:
    merge = merge_options.pop()
    # Check whether the best merge is still valid
    if merge.left not in tokens or merge.right not in tokens:
        continue  # outdated merge option
    merged = tokens[merge.left] + tokens[merge.right]
    if len(merged) != merge.checksum:
        continue  # outdated merge option
    # Merge the right token into the left token, then
    # add any new valid merge options to the priority queue.
    left = tokens.prev(merge.left)
    right = tokens.next(merge.right)
    tokens[merge.left] = merged
    tokens.remove(merge.right)
    …
```

MAX on aws

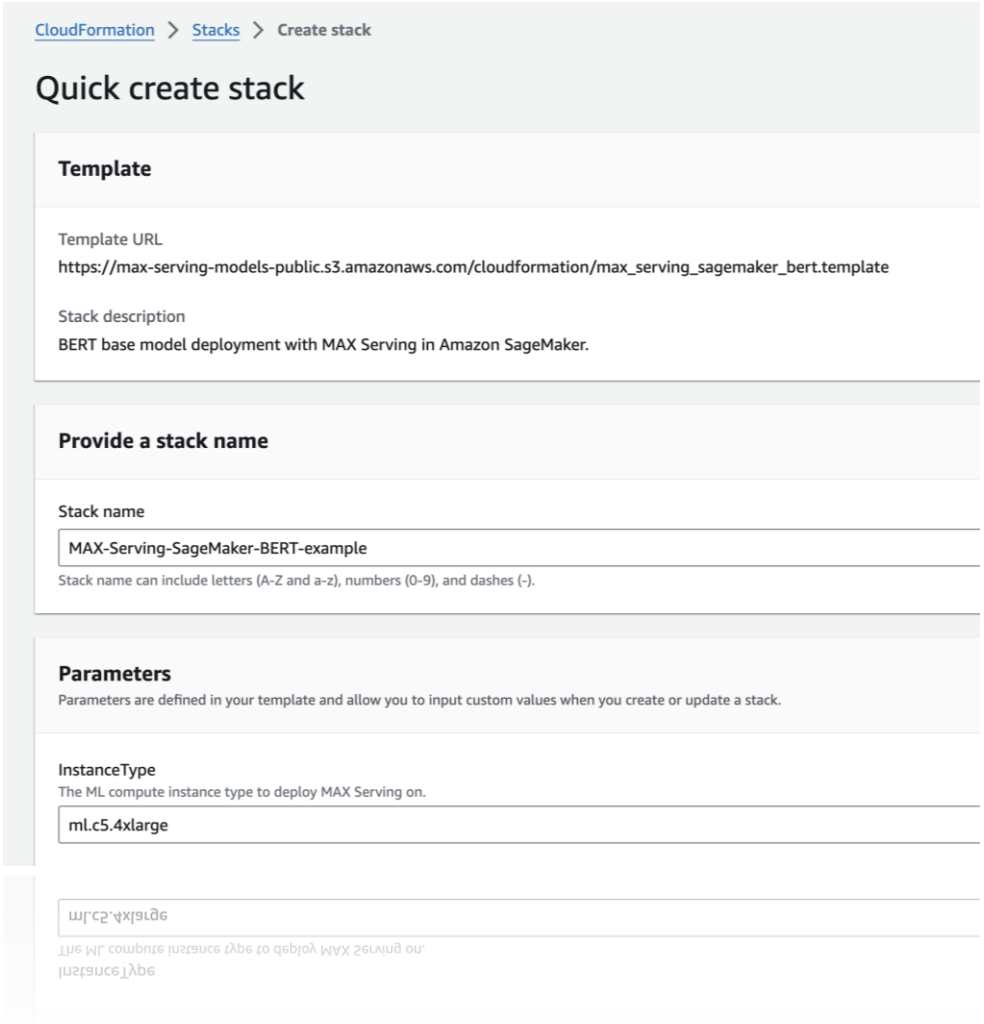
Scale to the cloud with MAX

1. Seamlessly interop into AWS cloud environments
2. Deploy through containers to SageMaker and EKS
3. Enable local to cloud deployment seamlessly

Deploying to SageMaker is easy

1. Download a pretrained model, or use your model, ready to use
2. Upload model to S3 so MAX & SageMaker can access it
3. Pull the latest MAX Serving container image, push to ECR
4. Create a SageMaker model and deploy to specified instance
5. Invoke the endpoint to test

Even faster with our Cloud Formation templates



Get started today

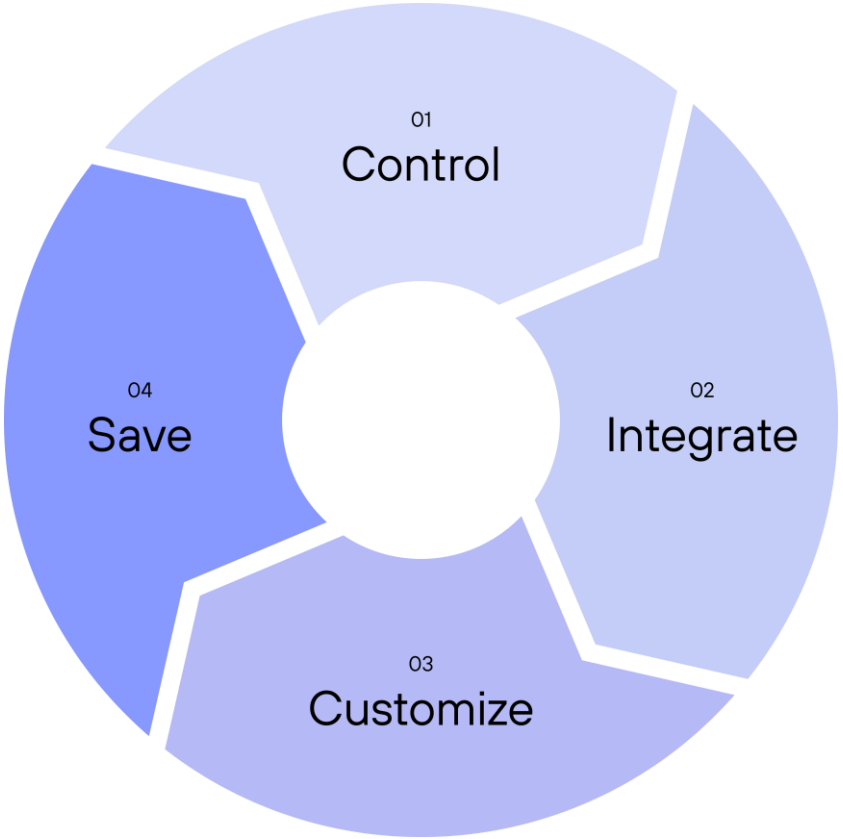
https://modul.ar/sagemaker-max

Let’s wrap up!

The future of AI can’t be owned by just a few

We want to help scale AI to the world ... for the world

Control your AI future



Download
MAX
free now

https://modular.com/max

Come join the revolution!