AWS re:Invent

DECEMBER 2 – 6, 2024 | LAS VEGAS, NV

AIM301-NEW

Accelerate gen AI: Amazon SageMaker HyperPod training plans & recipes

Gal Oshri

Prin. Product Manager,
Amazon SageMaker
AWS

Giuseppe A. Porcelli

Sr. Manager,
Amazon SageMaker
AWS

Babak Pahlavan

Founder & CEO
NinjaTech AI

Agenda

01 Challenges with training large-scale models

02 Amazon SageMaker HyperPod training plans

03 Amazon SageMaker HyperPod recipes

04 Demo

05 How NinjaTech AI has used Amazon SageMaker HyperPod training plans

Generative AI model computational demand is growing

Source: Our World in Data

Challenges with training large-scale models

Hardware

Faults

Timelines

Performance

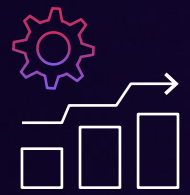
Cost

Why Amazon SageMaker HyperPod

REDUCE TRAINING TIME BY UP TO 40% THROUGH RESILIENCY AND PERFORMANCE OPTIMIZATIONS

Resilient environment

Self-healing clusters reduce training time



Streamline distributed training

SageMaker distributed training libraries improve performance

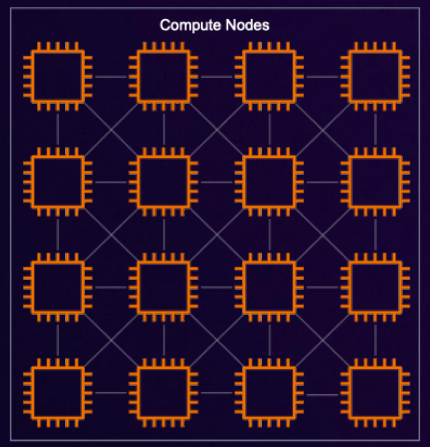
Customizable UX and persistent cluster

Control over computing environment and workload scheduling

Setting up compute resources

On–demand capacity

- Flexible
- Variable availability
- Higher-cost option



Long-term reservations

- Utilization
- Predictable availability
- Lower cost

Challenging to plan training workloads within timelines and budgets

NEW

Amazon SageMaker HyperPod flexible training plans

Save weeks of training time and help meet timelines and budgets

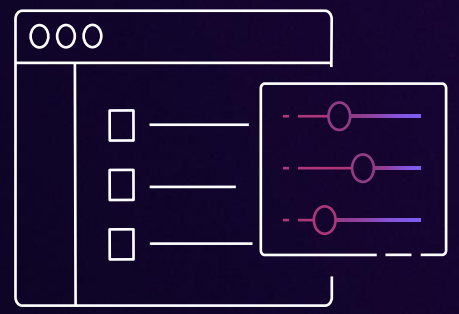
- Remove the uncertainty and manual process of capacity reservation
- Automatically set up training infrastructure and clusters
- Ensure FM training meets budgets and timelines

Get plans that meet your requirements

Specify your needs

- Instance type
- Number of instances
- Duration of plan
- Start date

Create a training plan



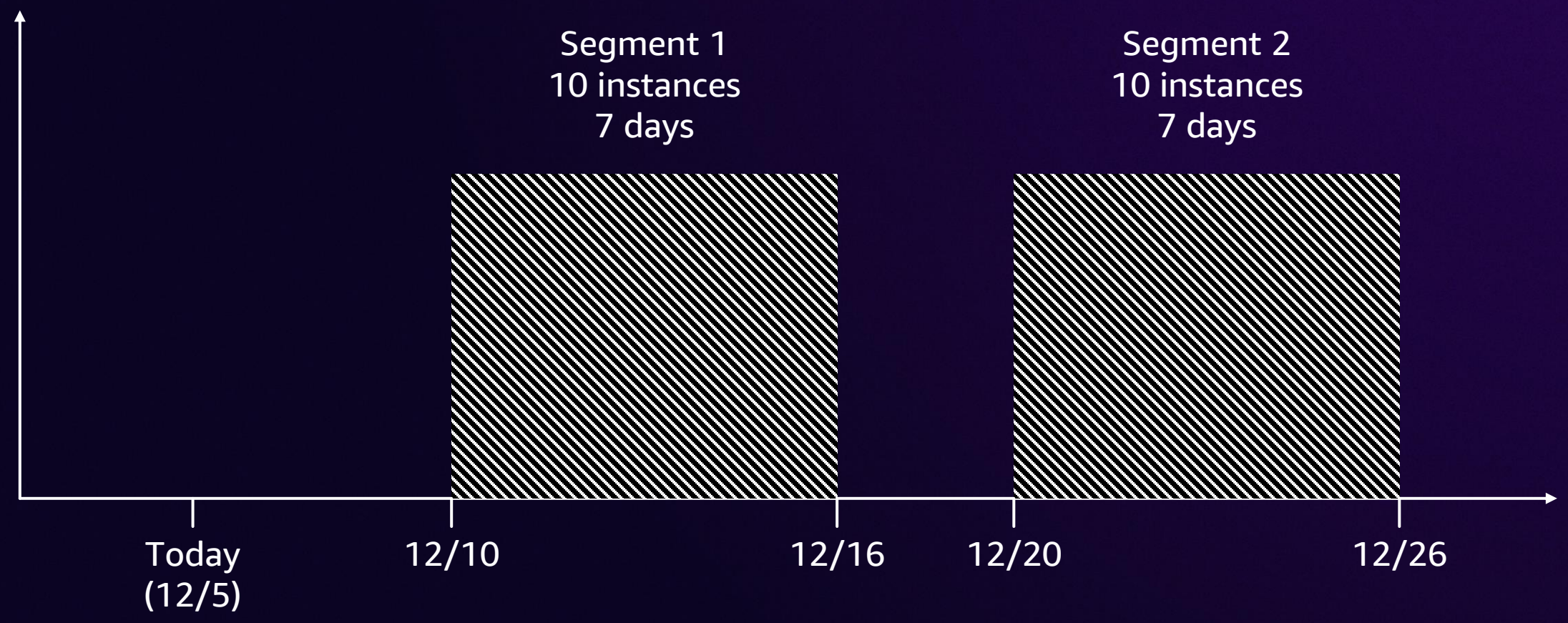
- Pick recommended offering
- Pay upfront

Use the training plan

- Use the plan in Amazon SageMaker HyperPod to create clusters
- Use the plan in Amazon SageMaker training jobs

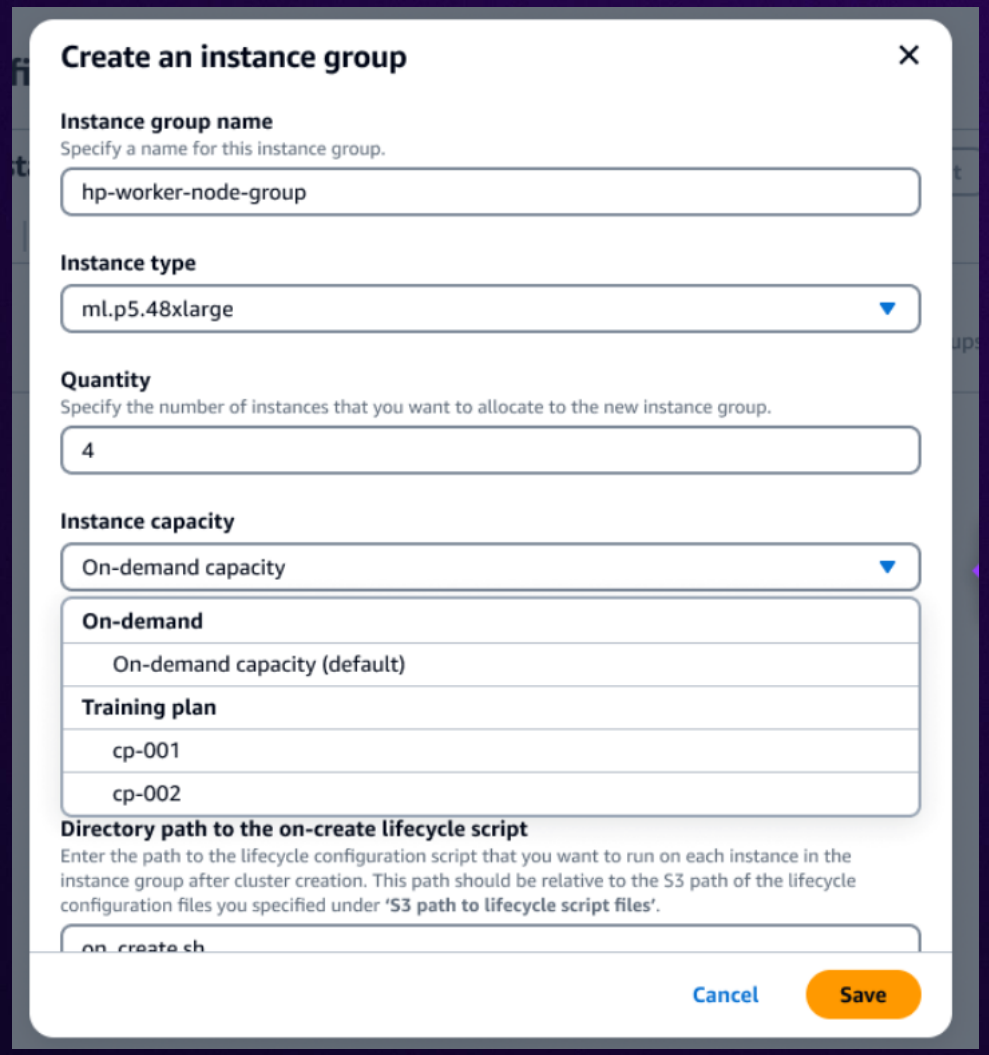
Training plans

“Create a training plan with 10 instances of ml.p5.48xlarge for 14 days starting 12/10”



Training plans with HyperPod

1. Create clusters with multiple instance groups
2. Specify instance groups to use training plans for compute
3. Instance groups scale up when training plan is active
   - ✓ Scale to requested count
   - ✓ Lifecycle scripts executed
   - ✓ Deep health checks
   - ✓ Resiliency



Benefits of using training plans

Compute access

Resiliency

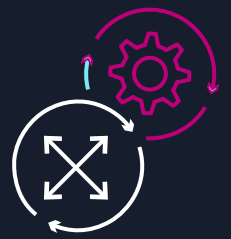
Predictable timelines

High performance

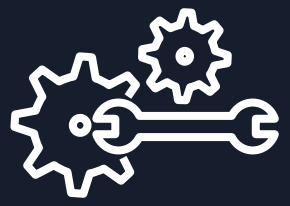
Upfront cost

What generative AI customers are **asking for**

Which model should I use?

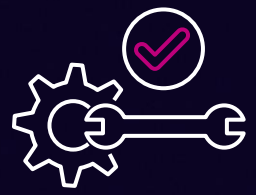


How do I customize my model?



How can I optimize training performance?

Customizing FMs for your business



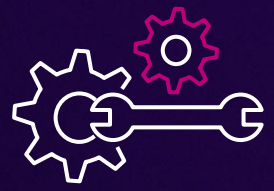
Fine-tuning

PURPOSE

Maximizing accuracy for specific tasks

DATA NEED

Small number of labeled examples



Pre-training

PURPOSE

Maintaining model accuracy for your domain

DATA NEED

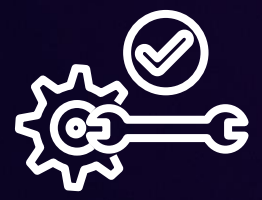
Large number of unlabeled datasets

Optimizing FM pre-training and fine-tuning can take weeks of effort

~Minutes

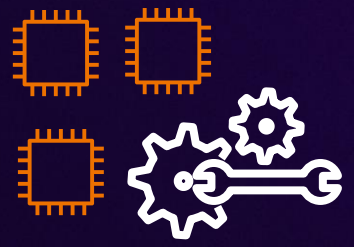
Select a model

~Hours



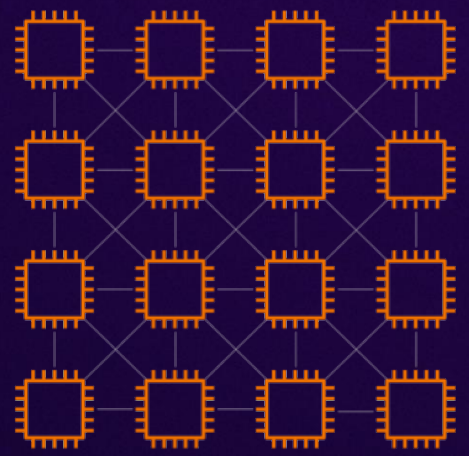
Configure framework

~Weeks



Optimize model training

~Months



Production pre-training & fine-tuning

Training FMs with billions of parameters spans thousands of different training stack configs

3-5 popular training frameworks

Dozens of different tunable parameters

Hundreds of configurable settings

Other performance optimizations like SMP, SMDDP

NOW AVAILABLE

Amazon SageMaker HyperPod recipes

Amazon SageMaker HyperPod recipes

Curated, ready-to-use recipes for pre-training and fine-tuning popular publicly available FMs

Tested and validated for foundational models such as Llama and Mistral

Automatic checkpoints for faster fault recovery and managed end-to-end training loop

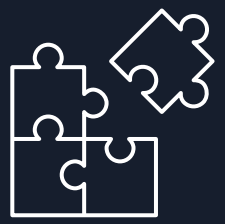
Easily switch between GPU-based or Trainium-based instances

Getting started in minutes

RUN FM PRE-TRAINING AND FINE-TUNING WITH A SINGLE LINE OF CODE

Select

Model training and fine-tuning recipe on GitHub



Setup Prerequisites

Resource limits, AWS credentials, training cluster

Run the recipe

on Amazon SageMaker HyperPod
or
Amazon SageMaker training jobs

Getting started

RUN FM PRE-TRAINING AND FINE-TUNING WITH A SINGLE LINE OF CODE

Recipes on Amazon SageMaker HyperPod (Slurm)

```
python3 main.py recipes=recipe-name
```

NeMo-style launcher

or

```
run_<model_name>.sh
```

Launcher script

Recipes on Amazon SageMaker HyperPod (EKS)

```
hyperpod start-job --recipe recipe-name
```

SageMaker HyperPod CLI

Recipes on Amazon SageMaker training jobs

```
model_trainer = ModelTrainer.from_recipe(
    training_recipe="<recipe_path>",
    ...
)

model_trainer.train(wait=False)
```

SageMaker Python SDK

How it works

AMAZON SAGEMAKER HYPERPOD RECIPES

aws / sagemaker-hyperpod-recipes

Type / to search

Code | Issues | Pull requests 8 | Actions | Projects | Wiki | Security 3 | Insights

sagemaker-hyperpod-recipes

Unwatch 18

main | 71 Branches | 0 Tags | Go to file t | Add file | Code

julianhr Remove setuptools upgrade from installation instructions (#130) ✓ dc5deef · 3 days ago 134 Commits

.github	Add repolinter to scan source code (#117)	last week
launcher	Only kill Kandinsky docker containers (#127)	4 days ago
launcher_scripts	Mixtral pretrain recipes (#124)	3 days ago
recipes_collection	Mixtral pretrain recipes (#124)	3 days ago
scripts	Add repolinter to scan source code (#117)	last week
template	Merge from bugbash. (#108)	2 weeks ago
tests	Only kill Kandinsky docker containers (#127)	4 days ago
.coveragerc	Increase coverage threshold from 80 to 90 (#105)	2 weeks ago

Open source implementation

Launcher scripts and recipes collection

Built on NVIDIA NeMo foundations (launcher, configuration hierarchy)

Over 30 recipes to get started

- SageMaker-optimized models (GPU)
- Neuron-optimized models (Trainium)
- Native NeMo models
- Custom models

How it works

EXAMPLE RECIPE

sagemaker-hyperpod-recipes / recipes_collection / recipes / training / llama / **hf_llama3_8b_seq16384_gpu_p5x16_pretrain.yaml**

Code | Blame | 142 lines (128 loc) · 3.54 KB

```yaml

# Model training configs
model:
  model_type: llama_v3
  # Base configs
  train_batch_size: 4
  val_batch_size: 1
  seed: 12345
  grad_clip: 1.0
  log_reduced_training_loss: True

  # Memory saving /distributed training configs
  tensor_model_parallel_degree: 4
  expert_model_parallel_degree: 1
  context_parallel_degree: 2
  moe: False
  activation_checkpointing: False
  activation_loading_horizon: 1
  delayed_param: True
  offload_activations: False
```

Hydra-based configuration

sagemaker-hyperpod-recipes / recipes_collection / **config.yaml**

Code | Blame | Executable File · 35 lines (27 loc) · 1003 Bytes

```yaml
defaults:
  - _self_
  - cluster: slurm  # set to `slurm`, `k8s` or `sm_jobs`, depending on the desired cluster
  - recipes: training/llama/hf_llama3_8b_seq8192_gpu # select desired config inside the training directory
  - override hydra/job_logging: stdout

cluster_type: slurm  # bcm, bcp, k8s or sm_jobs. If bcm, k8s or sm_jobs, it must match - cluster above.
# If using sm_jobs cluster_type, set sm_jobs_config. See cluster/sm_jobs.yaml for example.

hydra:
  run:
    dir: .
  output_subdir: null

debug: False

instance_type: p5.48xlarge
# TODO: remove
data_dir: null  # Location to store and read the data.
base_results_dir: null  # Location to store the results, checkpoints and logs.

container: null
```

How it works

AMAZON SAGEMAKER HYPERPOD TRAINING ADAPTER FOR NEMO

aws / sagemaker-hyperpod-training-adapter-for-nemo

Type / to search

Code | Issues 1 | Pull requests 3 | Actions | Projects | Wiki | Security | Insights

sagemaker-hyperpod-training-adapter-for-nemo

Unwatch 17

main | 99 Branches | 0 Tags | Go to file t | Add file | Code

ArjunKrishnak Fixing adaptor import in custom_pretrain (#195) 4861594 · 2 days ago 191 Commits

.github	Rename Adapter (#192)	3 days ago
examples	Fixing adaptor import in custom_pretrain (#195)	2 days ago
requirements	Add profiling dependencies file, install all dependencies (i...	last week
scripts	Cleanups for Adaptor (#185)	5 days ago
src/hyperpod_nemo_adapter	Add option to save the final full checkpoint directly (#193)	3 days ago
tests	fix test_patched_LFA2__init__ (#194)	3 days ago
.gitignore	unit tests for sagemaker_base_model	4 months ago
.pre-commit-config.yaml	Moe support (#57)	2 months ago

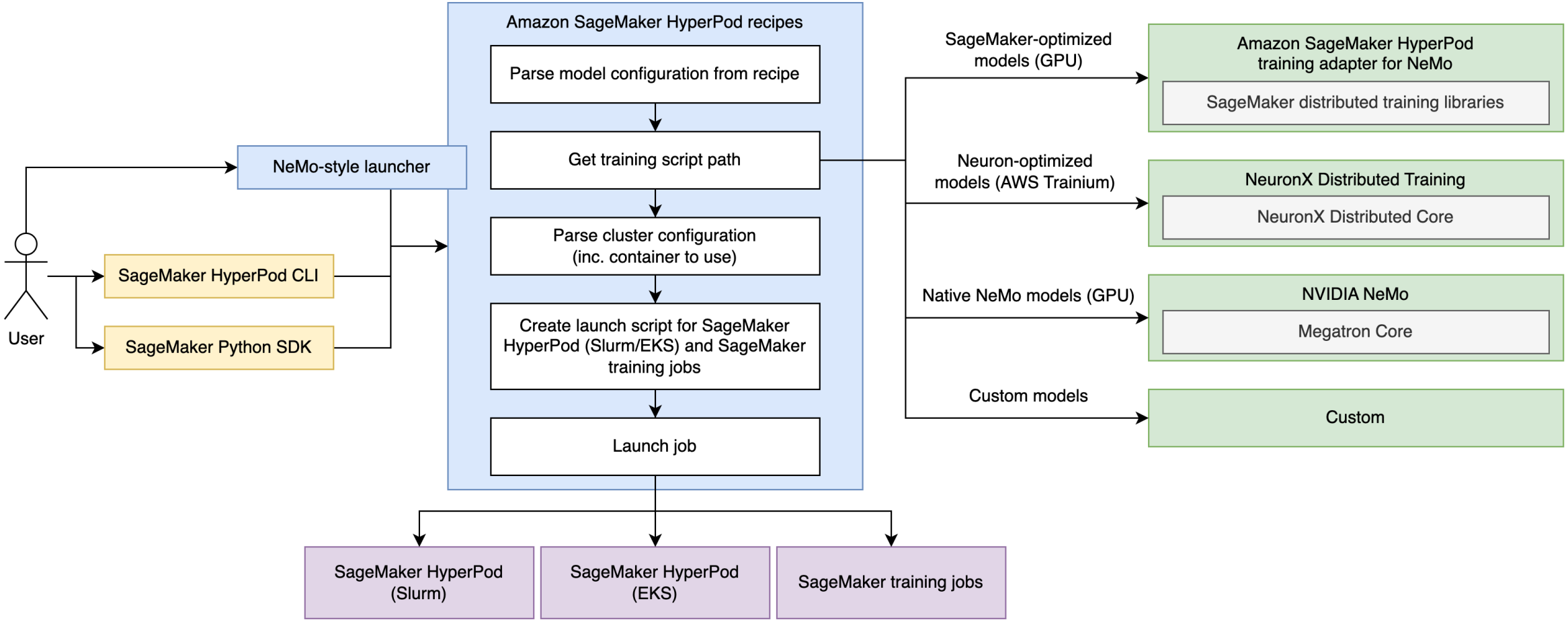
Defines training loop, data loading, and automatic checkpointing code for SageMaker-optimized models (GPU)

Implements optimized SageMaker distributed training strategies

Built on NVIDIA NeMo backend

How it works

JOB SUBMISSION WORKFLOW



Amazon SageMaker HyperPod training plans and recipes DEMO

How NinjaTech AI has used SageMaker HyperPod training plans

NinjaTech AI

Your All-In-One AI Agent for Unlimited Productivity

“Netflix of gen AI”: Democratizing access to world’s best AI models & AI skills

Babak Pahlavan
Founder, CEO & CPO
December 2024

Ninja

All-in-one unlimited AI skills

Market Today

Multiple subscriptions & models

- Code generation
- Image creation
- Online research

And more

One subscription

Unlimited skills & model access

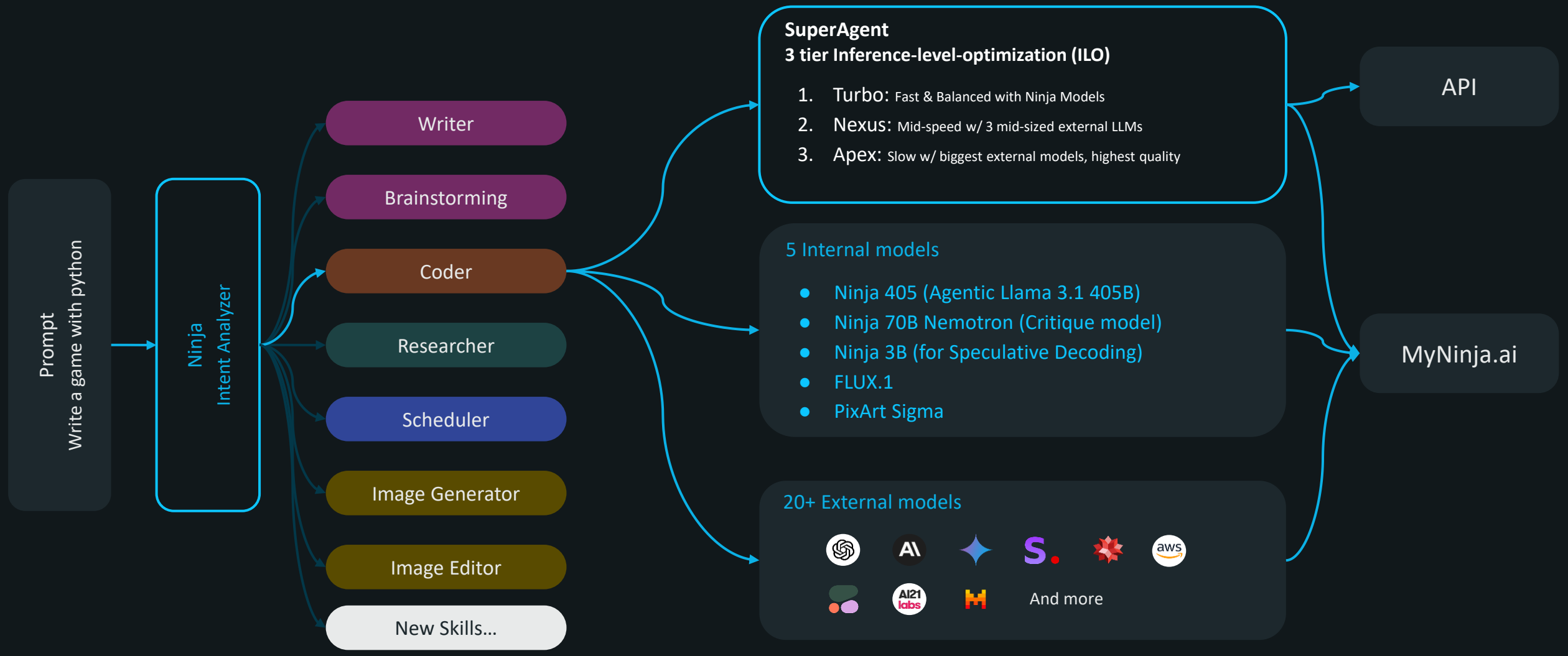
- Advance reasoning
- Image creation
- Code generation
- Online research
- File analysis
- Vision analysis
- Multilingual processing
- Fast inference
- More skills
  Scheduling, Travel and etc
- Access to external models

+more

SuperAgent AI Assistant

Platform & Model agnostic all-in-one solution

Get tasks done using the state of art AI agents and world's best LLMs



Our challenges for training: Fine-tuning our 405B model, reducing costs, and automating

Mission: To be the one-stop shop for all gen AI solutions, providing a seamless experience for users to access multiple AI agents.

Challenges

Critical Need

- A model that can automatically detect user intent and determine which AI agent to use.
- Can be updated to incorporate customer feedback and new features. A process involving 10m-100m tokens at each round of LoRA fine tune.

Resource & Operational

- **Hardware Access**: Acquiring high-performance GPUs (e.g., H100 or H200) is challenging due to their high demand – specifically in multi-node clusters.
- **Training Process**: Complex, time-consuming, and multi-step training process involving model downloading, distributed training, checkpoint, monitoring, auto-remediation, merging, and quantization.

Ninja + HyperPod training plans

Planning reliable + affordable compute in advance – HyperPod training plans delivered on this challenge

Llama 3.1 405b

Fine-Tuning on HyperPod –
Utilized HyperPod training plans to fine-tune our 405B model on HyperPod, leveraging the power of predictable access to Nvidia H100 GPU capacity.

Efficient

Cost-Effective

High-Performance

Customized Training Plans: Tailored training plan that matched our specific compute and timeline needs.

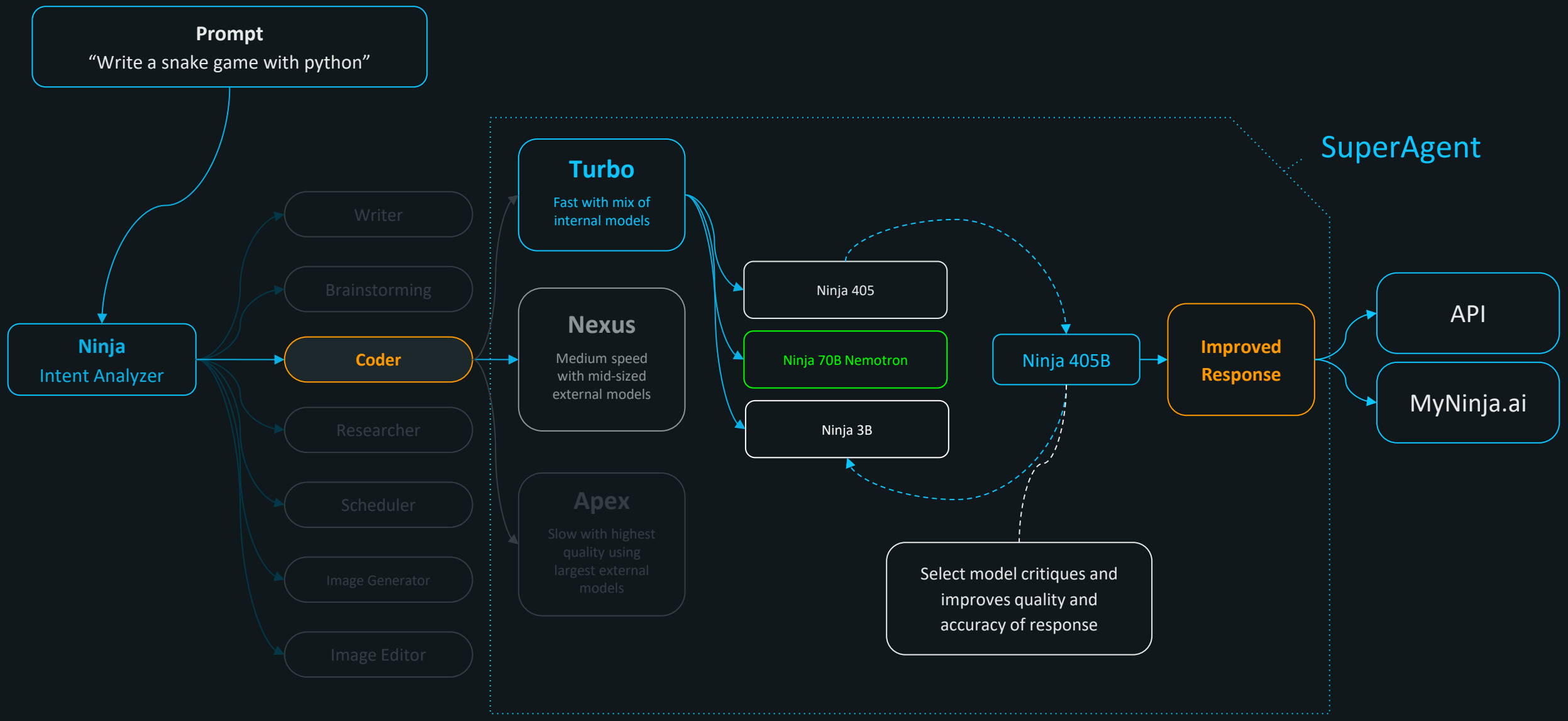
Automated Execution + Resource Management: SageMaker HyperPod training plans automated the execution of our training plan, provisioning required resources, setting up of infrastructure, and recovering from failures.

Flexible Reservation Options: Flexible reservation options – duration and instance quantity that suited our needs.

Effortless Integration With HyperPod Clusters: Seamlessly connected SageMaker HyperPod training plans with Amazon SageMaker HyperPod clusters, which offers expertly managed GPU clusters with advanced monitoring and automatic recovery capabilities.

SuperAgent

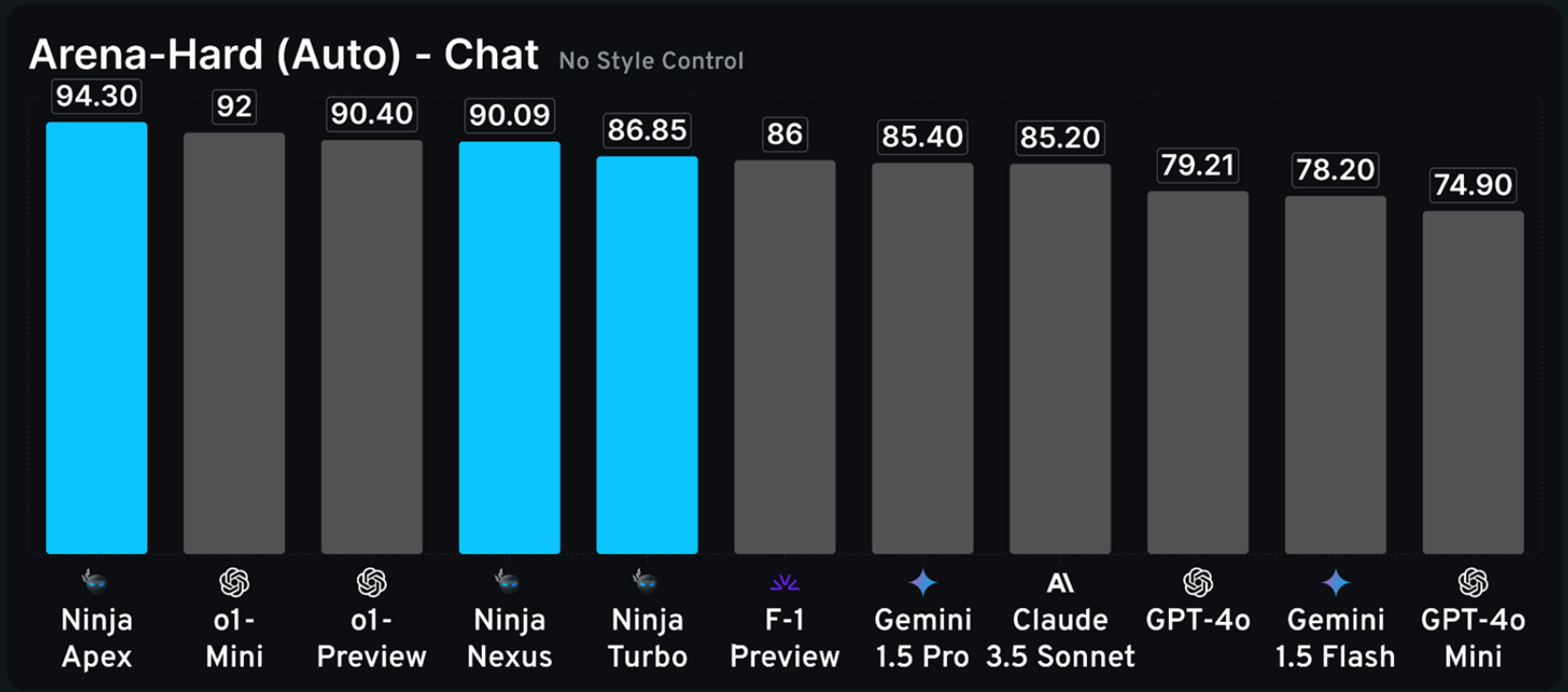
Real-time Inference Level Optimization (ILO) technology



SuperAgent: Agentic Inference level optimization (More tokens → higher accuracy and better results)

Ninja achieves SOTA (State-of-the-art)

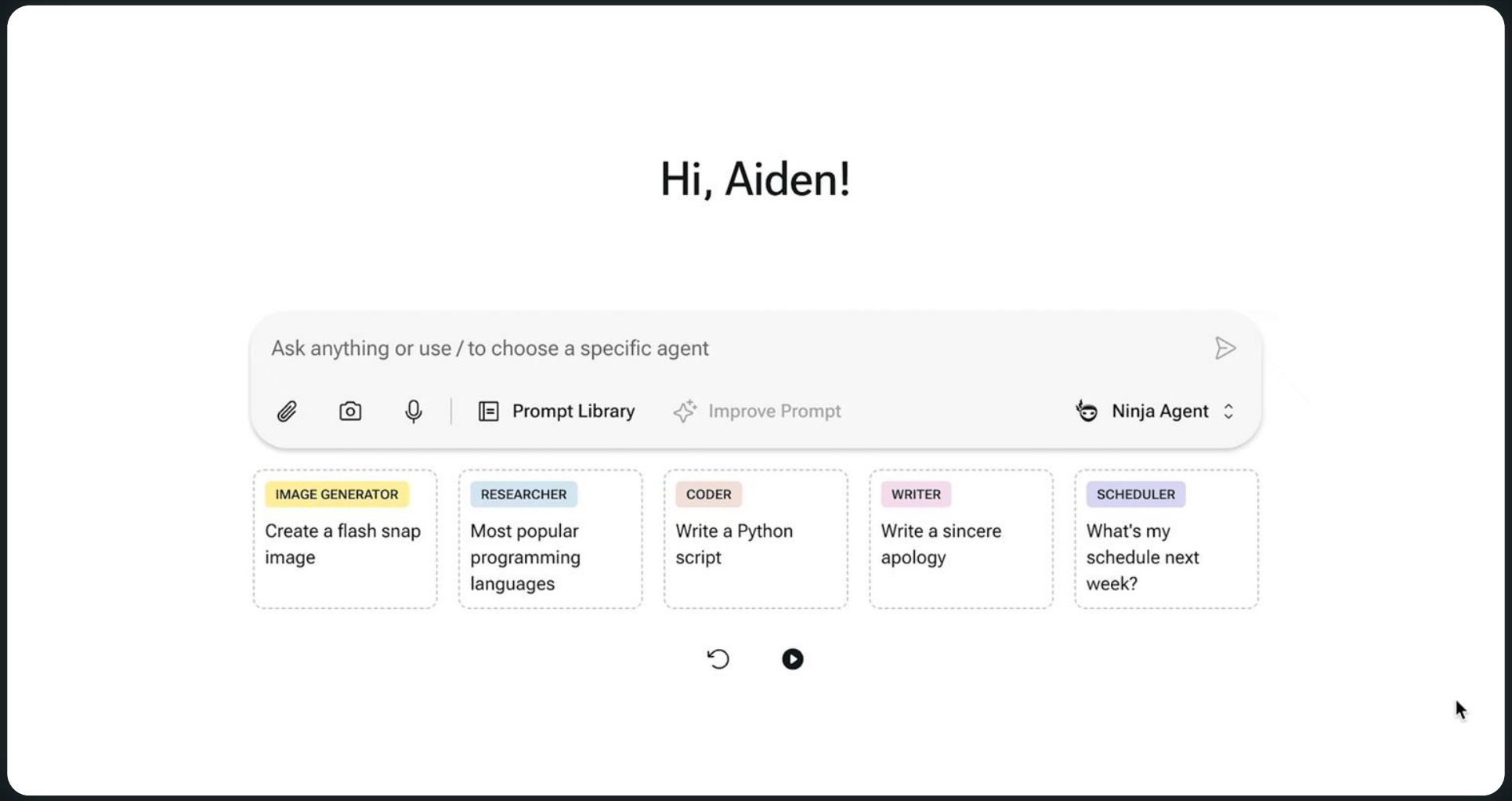
1. SuperAgent variations
    a. Turbo (Fast with internal models)
    b. Nexus (Medium speed with mid-sized external models)
    c. Apex (Slow with largest external models)
2. Accuracy optimizations: Skill based accuracy improvement algorithms (Example: Coding questions will use a different algorithms compared to writing or Research questions).



Updated on (12/03/2024)

Demo

Multi-node trained: Making Llama models talkative



Recap

HyperPod is a game changer, especially for start-ups: It allows seamless multi-node large-scale training.

Steps

- Prerequisites: SageMaker HyperPod VPC stack, cluster observability stack, SageMaker Studio stack
- Create SageMaker training plan
- Cluster setup: config on Instances, training plan
- SSH to cluster instances
- Train model with cluster instances

HyperPod

Advantages

- Capacity Reservation => Saving Time & money!
- Terminate instance automatically when reservation finished
- Manage the VPC: enable seamless communication between each instances in cluster
- Automation via recipes
- Single UI
- Support

Try Ninja today!

MyNinja.ai

Resources & references

Service page	https://aws.amazon.com/sagemaker/hyperpod/
Documentation	https://docs.aws.amazon.com/sagemaker/latest/dg/reserve-capacity-with-training-plans.html https://docs.aws.amazon.com/sagemaker/latest/dg/sagemaker-hyperpod-recipes.html
Blogs	https://aws.amazon.com/blogs/aws/meet-your-training-timelines-and-budgets-with-new-amazon-sagemaker-hyperpod-flexible-training-plans/ https://aws.amazon.com/blogs/aws/accelerate-foundation-model-training-and-fine-tuning-with-new-amazon-sagemaker-hyperpod-recipes/
Announcements	https://aws.amazon.com/about-aws/whats-new/2024/12/amazon-sagemaker-hyperpod-flexible-training-plans/ https://aws.amazon.com/about-aws/whats-new/2024/12/amazon-sagemaker-hyperpod-recipes/

Thank you!

Please complete the session survey in the mobile app

Gal Oshri
galoshr@amazon.com

Giuseppe Angelo Porcelli
gianpo@amazon.com

Babak Pahlavan
babak@ninjatech.ai