



KubeCon



CloudNativeCon

Europe 2026

#KubeCon #CloudNativeCon

Dragonfly V2.4.0 - Intro, Updates, Data Distribution in AI Infrastructure

Wenbo Qi(Gaius) – Ant Group



Introduction



KubeCon



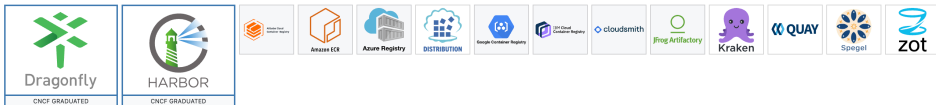
CloudNativeCon

Europe 2026



Container Registry

CNCF Projects	Keywords
• Dragonfly (graduated) • Harbor (graduated) • Distribution (sandbox) • zot (sandbox)	• Container • OCI Image • Registry



What is Dragonfly?

Delivers efficient, stable, and secure data distribution and acceleration powered by P2P technology. It aims to be the best practice and standard solution in cloud native architectures. Designed to improve the performance of large-scale delivery across files, container images, OCI artifacts, AI models, caches, logs, dependencies, etc.

It is hosted by the Cloud Native Computing Foundation(CNCF)as an **Graduated** Level Project.

There are more than **100+** contributors, and maintainers come from **Ant Group**, **Alibaba Group**, **ByteDance**, **Intel**, **Baidu Group**, **Zhipu AI** and **Dalian University of Technology**.



Introduction

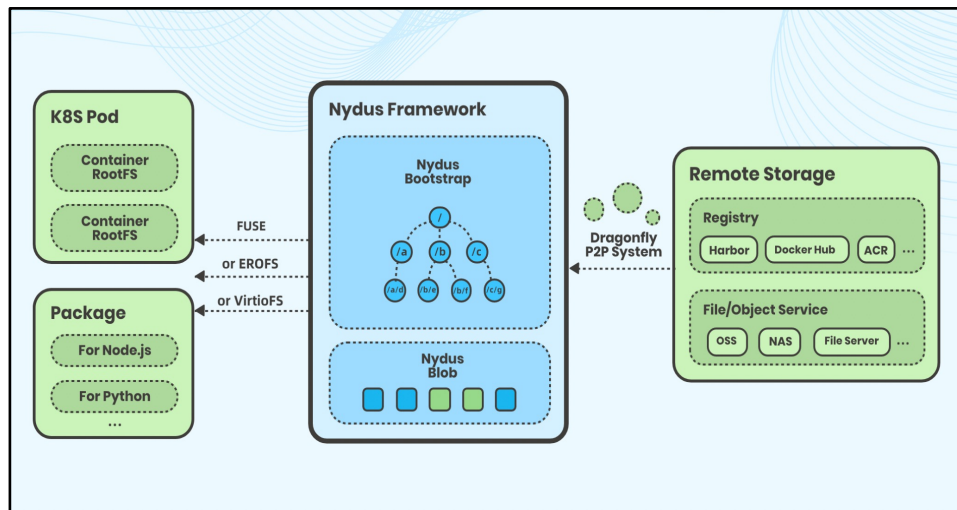


KubeCon



CloudNativeCon

Europe 2026



What is Nydus?

Provide a content-addressable file system on the RAFS format, which enhances the current OCI image specification by improving container launch speed, image space and network bandwidth efficiency, and data integrity.

It is a *sub-project of Dragonfly*. It can reduce the end-to-end cold start time of containers from *minutes to seconds*, and supports the creation of *millions of containers* every day in production.



CLOUD NATIVE
COMPUTING FOUNDATION

Introduction



KubeCon



CloudNativeCon

Europe 2026

Image Acceleration

Dragonfly supports various container clients such as **containerd**, **Docker**, **cri-o**, **ORAS**, etc. It provides three solutions for image acceleration.

The first solution is to use Dragonfly to distribute images based on P2P technology, which is suitable for **large-scale cluster**.

The second solution is to use Dragonfly and Nydus to distribute accelerated images, which is suitable for **large-scale cluster** and **faster container launching**.

The third solution is to use Nydus to distribute accelerated images, which is suitable for **faster container launching**.

File Distribution

Dragonfly supports large-scale file distribution and uses P2P technology to **eliminate the impact of origin bandwidth limitations**.

It supports file distribution of protocols including **HTTP**, **HDFS**, etc. Additionally, it also supports different object storage protocols includes **S3**, **OSS**, **OBS**, etc.

AI Infrastructure

Dragonfly supports distributing data during **AI training** and **AI inference**.

In the AI inference, Dragonfly supports **ModelPack**, **Triton Server**, **TorchServe** to use Dragonfly distribution model.

Supports downloading models and datasets from **Hugging Face Hub** by SDK through dfget command.

Introduction



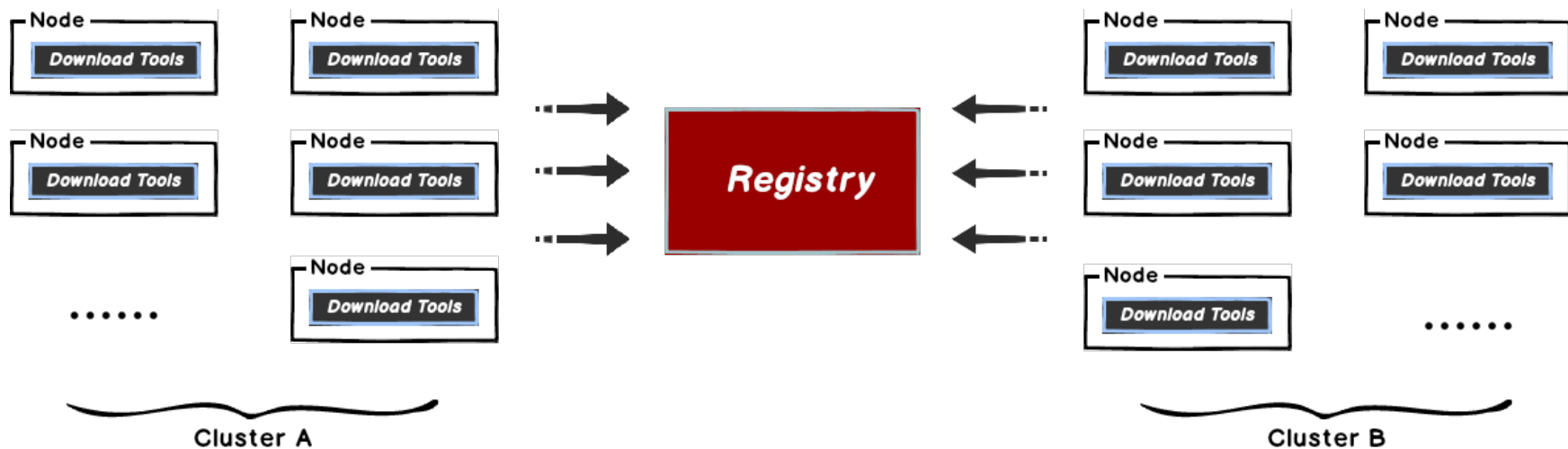
KubeCon



CloudNativeCon

Europe 2026

When downloading files in large batches, the storage bandwidth can easily reach the limit.



Introduction



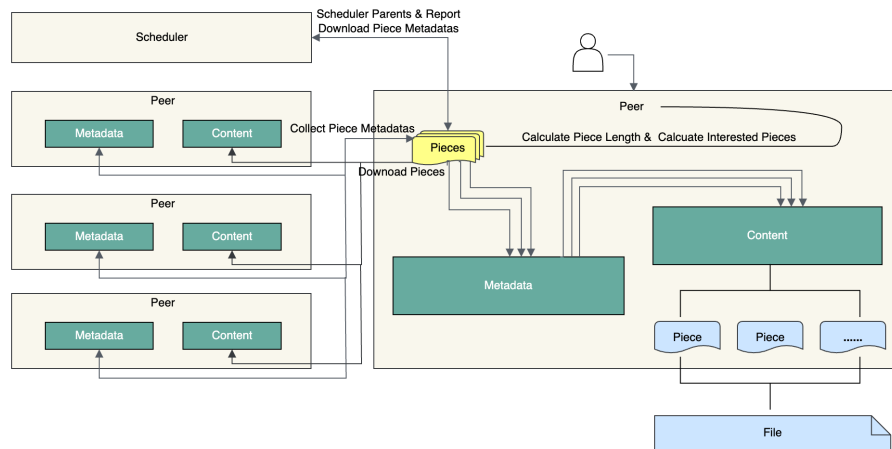
KubeCon



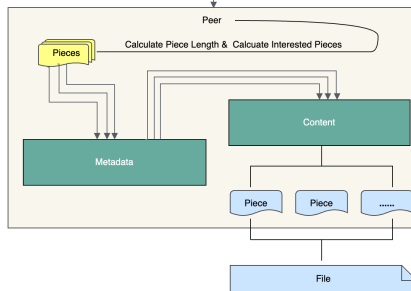
CloudNativeCon

Europe 2026

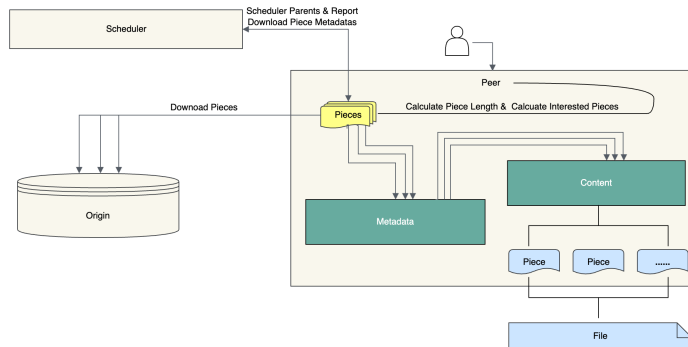
Hit Remote Peer Cache



Hit Local Peer Cache



Download From Origin



Exciting Updates



KubeCon



CloudNativeCon

Europe 2026

Vortex: A P2P File Transfer Protocol Based on TLV

Packet Format

Packet Identifier
(1 byte)

PACKET ID (1 byte)	Tag (T, 1 byte)	Length (L, 4 bytes)	Variable Length Value (V) Max 4GiB
-----------------------	--------------------	------------------------	---------------------------------------

Unsigned Integer,
up to 4GiB

TAG DEFINITIONS

TAG (Hex/Dec)	NAME	DESCRIPTION
0x00 / 0	Download Piece	Download the content of a piece from a peer. It is composed of (Task ID)(Piece ID), where the Task ID is a SHA-256 value and the Piece ID is a number.
0x01 / 1	Piece Content	The content of a piece, with a maximum size of 1 GiB per piece.
0x02 / 2	Download Persistent Piece	Download the content of a persistent piece from a peer. It is composed of (Task ID)(Piece ID), where the Task ID is a SHA-256 value and the Piece ID is a number.
0x03 / 3	Persistent Piece Content	The content of a persistent piece, with a maximum size of 1 GiB per piece.
0x04 / 4	Download Persistent Cache Piece	Download the content of a persistent cache piece from a peer. It is composed of (Task ID)(Piece ID), where the Task ID is a SHA-256 value and the Piece ID is a number.
0x05 / 5	Persistent Cache Piece Content	The content of a persistent cache piece, with a maximum size of 1 GiB per piece.
0x06 / 6	Download Cache Piece	Download the content of a cache piece from a peer. It is composed of (Task ID)(Piece ID), where the Task ID is a SHA-256 value and the Piece ID is a number.
0x07 / 7	Cache Piece Content	The content of a cache piece, with a maximum size of 1 GiB per piece.
0x08-0xFD / 8-253	Reserved	Reserved for future use.
0xFE / 254	Close	Close connection.
0xFF / 255	Error	Error message.

Example Packet

Packet ID: 0x12	Tag: 0x00 (Download Piece)	Length: 10 (bytes)	Value: "HelloWorld"
--------------------	-------------------------------	-----------------------	---------------------

Exciting Updates



KubeCon



CloudNativeCon

Europe 2026

Vortex: A P2P File Transfer Protocol Based on TLV

Vortex(TCP) VS GRPC

Performance

The test environment is a local area network with RTT less than 10ms, using the [CUBIC](#) congestion control algorithm for TCP protocols.

Protocol	File Size	Download Time from Parent Peer	CPU/MEM	Piece Concurrency	Peak Memory	Parent Peer Peak Memory
GRPC	30G	40.001s	16C32G	32	Peak Virtual Memory: 5236392 kB Peak Resident Memory: 2151256 kB	Peak Virtual Memory: 12174728 kB Peak Resident Memory: 5206904 kB
Vortex(TCP)	30G	19.869s	16C32G	32	Peak Virtual Memory: 2435020 kB Peak Resident Memory: 561364 kB	Peak Virtual Memory: 2835012 kB Peak Resident Memory: 728256 kB
GRPC	10G	12.543s	16C32G	32	Peak Virtual Memory: 3731144 kB Peak Resident Memory: 1506620 kB	Peak Virtual Memory: 7200984 kB Peak Resident Memory: 2338528 kB
Vortex(TCP)	10G	6.542s	16C32G	32	Peak Virtual Memory: 2544092 kB Peak Resident Memory: 675212 kB	Peak Virtual Memory: 2707524 kB Peak Resident Memory: 726016 kB
GRPC	1G	1.651s	16C32G	32	Peak Virtual Memory: 1257368 kB Peak Resident Memory: 425688 kB	Peak Virtual Memory: 2663160 kB Peak Resident Memory: 660344 kB
Vortex(TCP)	1G	1.562s	16C32G	32	Peak Virtual Memory: 1164812 kB Peak Resident Memory: 231308 kB	Peak Virtual Memory: 2699596 kB Peak Resident Memory: 688820 kB

Vortex(QUIC) VS GRPC

Performance

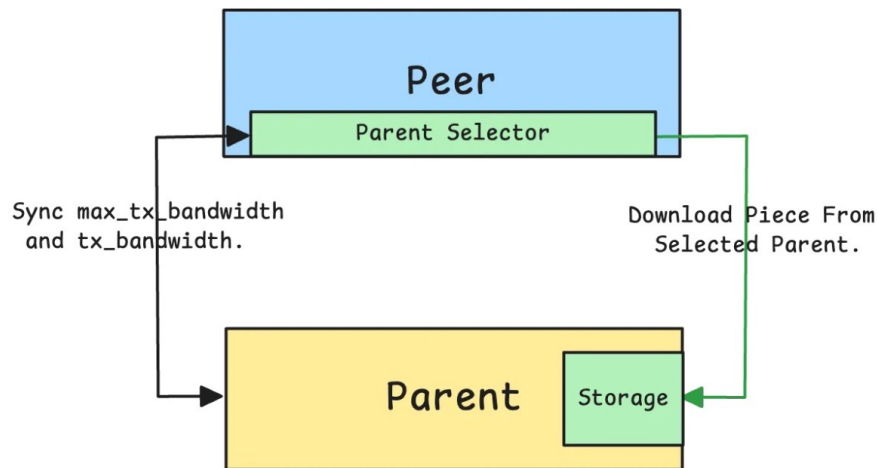
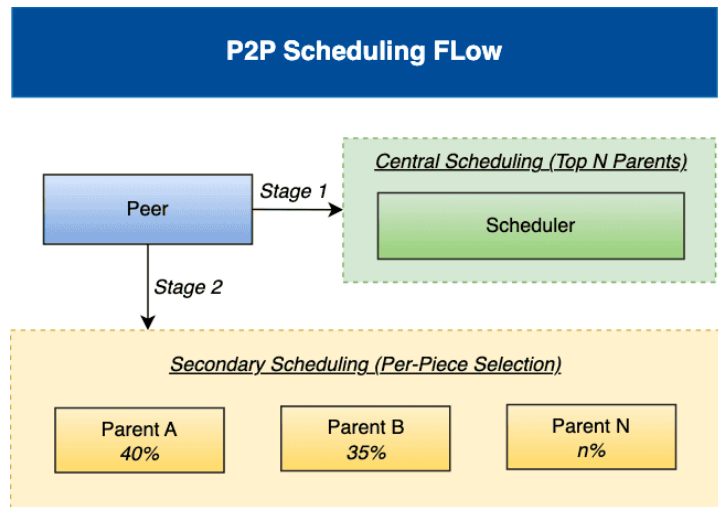
The test environment is a local area network with RTT less than 10ms, using the Bottleneck Bandwidth and Round-trip propagation time(BBR) congestion control algorithm for QUIC protocols.

Protocol	File Size	Download Time from Parent Peer	CPU/MEM	Piece Concurrency	Peak Memory	Parent Peer Peak Memory
GRPC	30G	40.001s	16C32G	32	Peak Virtual Memory: 5236392 kB Peak Resident Memory: 2151256 kB	Peak Virtual Memory: 12174728 kB Peak Resident Memory: 5206904 kB
Vortex(QUIC)	30G	28.214s	16C32G	32	Peak Virtual Memory: 3021500 kB Peak Resident Memory: 1047568 kB	Peak Virtual Memory: 3002448 kB Peak Resident Memory: 749596 kB
GRPC	10G	12.543s	16C32G	32	Peak Virtual Memory: 3731144 kB Peak Resident Memory: 1506620 kB	Peak Virtual Memory: 7200984 kB Peak Resident Memory: 2338528 kB
Vortex(QUIC)	10G	9.950s	16C32G	32	Peak Virtual Memory: 2745432 kB Peak Resident Memory: 971068 kB	Peak Virtual Memory: 2987628 kB Peak Resident Memory: 783688 kB
GRPC	1G	1.651s	16C32G	32	Peak Virtual Memory: 1257368 kB Peak Resident Memory: 425688 kB	Peak Virtual Memory: 2663160 kB Peak Resident Memory: 660344 kB
Vortex(QUIC)	1G	1.806s	16C32G	32	Peak Virtual Memory: 2595516 kB Peak Resident Memory: 294344 kB	Peak Virtual Memory: 2744732 kB Peak Resident Memory: 715704 kB

Load-aware Scheduling Algorithm

$TotalScore = CalculateParentScore(IDCAffinity, LocationAffinity, LoadQuality, HostType)$

$LoadQualityScore = CalculateLoadQualityScore(PeakBandwidthUsage, BandwidthDurationRatio, ConcurrentEfficiency)$



Exciting Updates



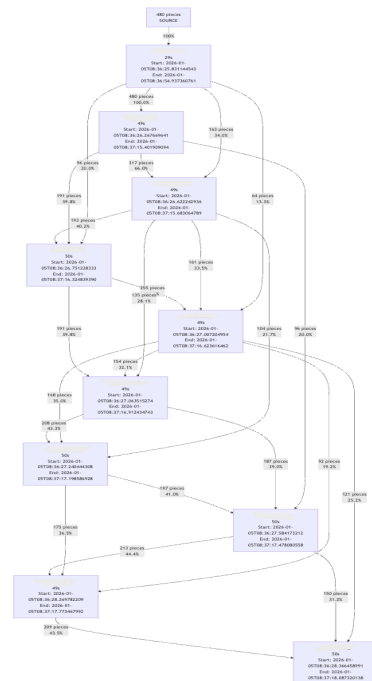
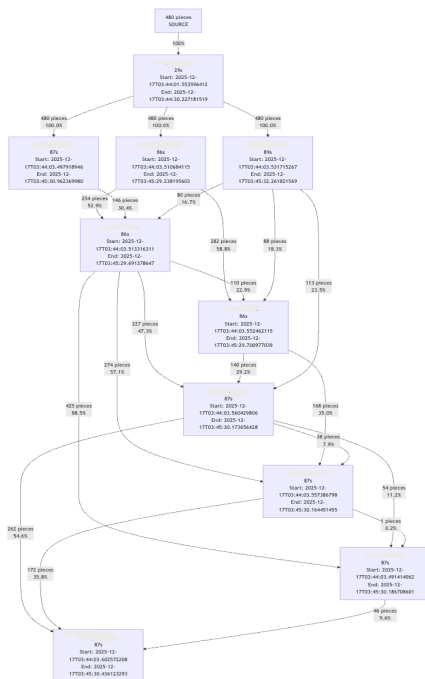
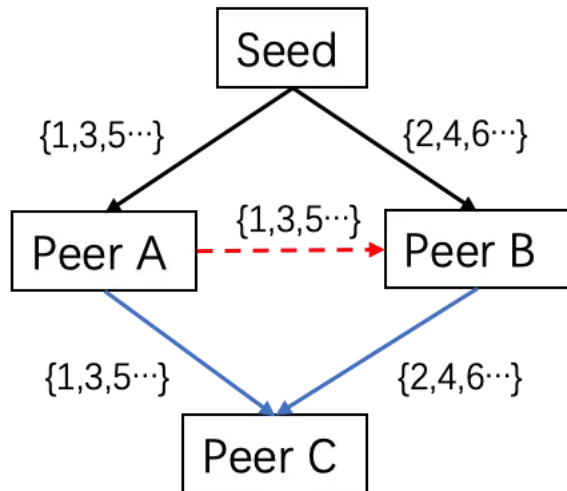
KubeCon



CloudNativeCon

Europe 2026

Load-aware Scheduling Algorithm



Exciting Updates



KubeCon



CloudNativeCon

Europe 2026

Enhanced Preheating

1. Support for IP-based peer selection in preheating jobs with priority-based selection logic where IP specification takes highest priority, followed by count-based and percentage-based selection.
2. Support for preheating multiple URLs in a single request.
3. Support for preheating file and image via Scheduler gRPC interface.

The screenshot shows the 'Create Preheat' interface in the Dragonfly web console. The interface is dark-themed and includes a sidebar with navigation options: Cluster, Job, Preheat, Resource, Audit, Garbage Collection, Developer, and User. The main content area is titled 'Create Preheat' and has a breadcrumb 'Job > Preheat > New preheat'. It features two tabs: 'Preheat File' and 'Preheat Image'. The form includes the following sections:

- Information:** A text input field for 'Description'.
- Clusters:** A dropdown menu for 'Cluster *'.
- URL:** A text input field for 'URL *' and a button '+ ADD URL'.
- Args:** A section with 'Piece Length MIB' (a dropdown) and 'Scope *' (a dropdown set to 'All Peers'). Below this are radio buttons for selection methods: 'All', 'IPs' (which is selected), 'Percentage', and 'Count'. An input field for 'IPs' is located below the radio buttons.
- Footer:** Fields for 'Tag', 'Application', and 'Filter Query Params'.

At the bottom left, there are 'Light' and 'Dark' theme toggles, and a user profile icon labeled 'root'.

Exciting Updates



KubeCon



CloudNativeCon

Europe 2026

Calculate task ID based on image blob SHA256 to avoid redundant downloads

The Client now supports calculating task IDs directly from the SHA256 hash of image blobs instead of using the download URL. This enhancement prevents redundant downloads and data duplication when the same blob is accessed from different registry domains.

Cache HTTP 307 redirects for split downloads

Support for caching HTTP 307 (Temporary Redirect) responses to optimize Dragonfly's multi-piece download performance. When a download URL is split into multiple pieces, the redirect target is now cached, eliminating redundant redirect requests and reducing latency.

Go Client Deprecated and Replaced by Rust Client

The Go client has been deprecated and replaced by the Rust Client. All future development and maintenance will focus on the Rust client, which offers improved performance, stability, and reliability.

Model Distribution

AI models serve as the central artifacts across both training and inference. This part focuses on inference, addressing how to manage and distribute these models within Kubernetes.

Large Size: Model weights have grown from hundreds of megabytes to hundreds of gigabytes, even terabytes, posing significant challenges for model storage and transfer.

Immutability: A base model used for inference is a read-only, versioned artifact. Like a container image, it follows a strict build once, distribute everywhere, never mutate in place.

Lack of Model Management Standards: The community is overly focused on distribution throughput while ignoring critical governance primitives, including standardized versioning, metadata schemas, and lifecycle management. These are exactly the prerequisites for promoting models from experimentation to production at scale.

Cloud-Native Gap: AI infrastructure is built on top of cloud-native infrastructure, yet companies widely rely on custom solutions around object storage or Git LFS for model management. This leads to non-standardized artifact management and a lack of cloud-native support.

Model Distribution



KubeCon



CloudNativeCon

Europe 2026

What If We Shipped Models the Same Way We Ship Code?

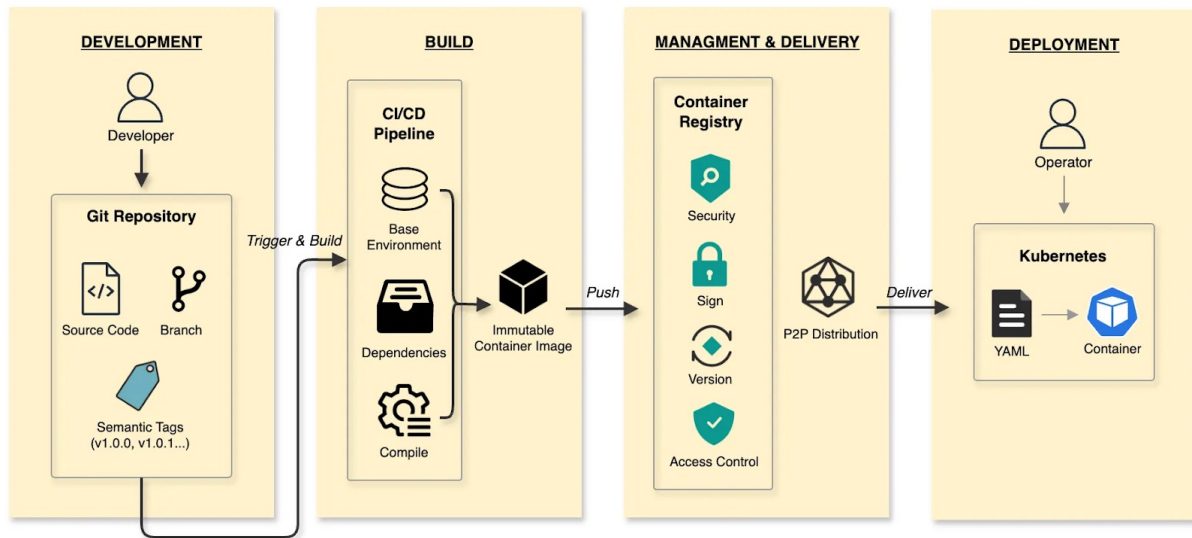
Software Delivery

Develop: Developers commit code to a Git repository, manage code changes through branches, and define versions using tags at key milestones.

Build: CI/CD pipelines compile and test, packaging the output into an immutable Container Image.

Manage & Deliver: Images are stored in a Container Registry. Supply chain security (scanning/signing), RBAC, and P2P distribution ensure safe delivery.

Deploy: DevOps engineers use declarative Kubernetes YAML to define the desired state. The Container's lifecycle is managed by Kubernetes.



Model Distribution



KubeCon



CloudNativeCon

Europe 2026

What If We Shipped Models the Same Way We Ship Code?

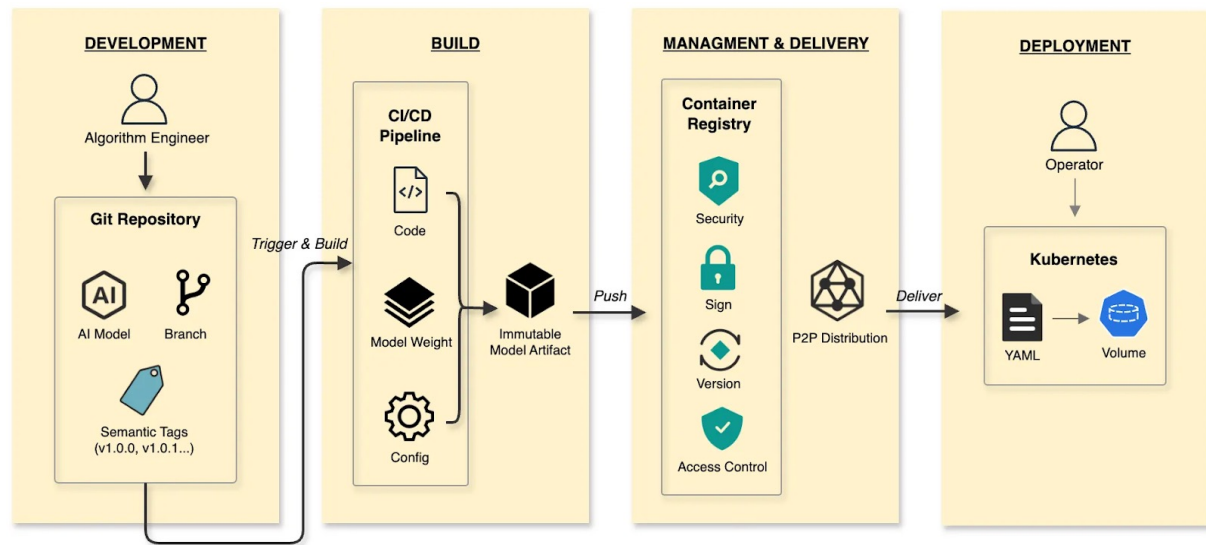
AI Model Delivery

Develop: Developers push weights and configs to the Hugging Face Hub, treating it as the Git Repository.

Build: CI/CD pipelines package weights, runtime configurations, and metadata into an immutable Model Artifact.

Manage & Deliver: The Model Artifact is managed by an Artifact Registry, reusing the existing container infrastructure and toolchain.

Deploy: Engineers use Kubernetes OCI Volumes or a Model CSI Driver. Models are mounted into the inference Container as Volumes, separate the AI model from the inference engine (vLLM, SGLang, etc.).



By approaching model lifecycle as a **supply chain**, we constructed an efficient system that resolves the challenges of managing and distributing AI models in production.

Model Distribution



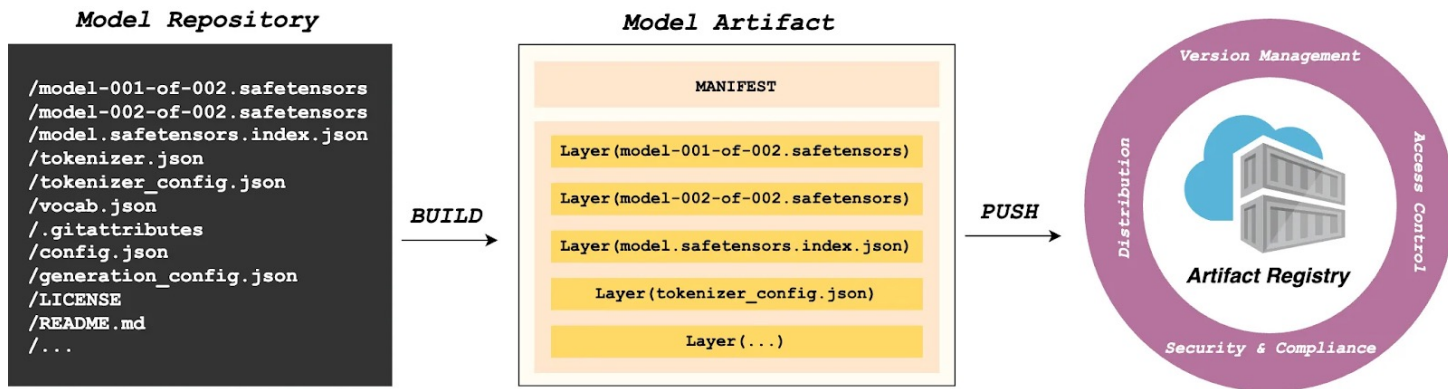
KubeCon



CloudNativeCon

Europe 2026

Build



```
$ modctl modelfile generate .
```

```
$ modctl login -u username -p password harbor.registry.com
```

```
$ modctl build -t harbor.registry.com/models/qwen2.5-0.5b:v1 -f Modelfile .
```

```
$ modctl push harbor.registry.com/models/qwen2.5-0.5b:v1
```

Model Distribution



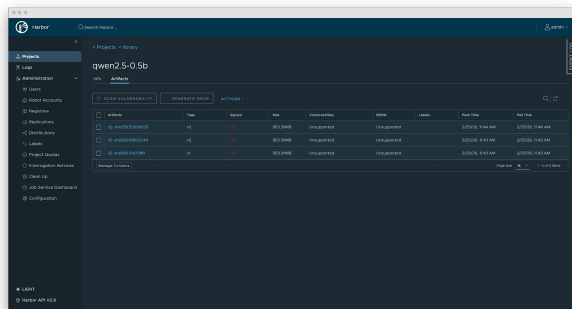
KubeCon



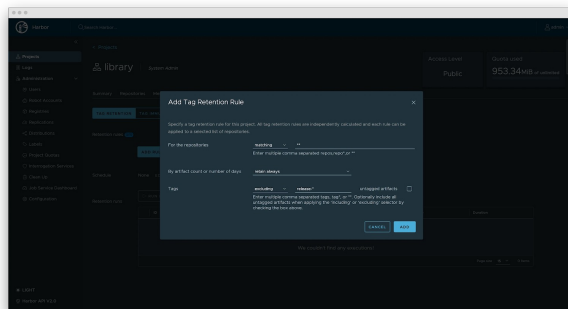
CloudNativeCon

Europe 2026

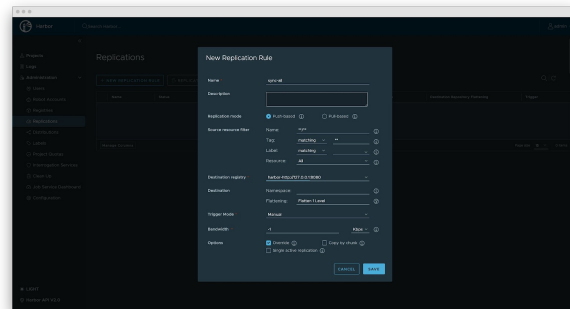
Management



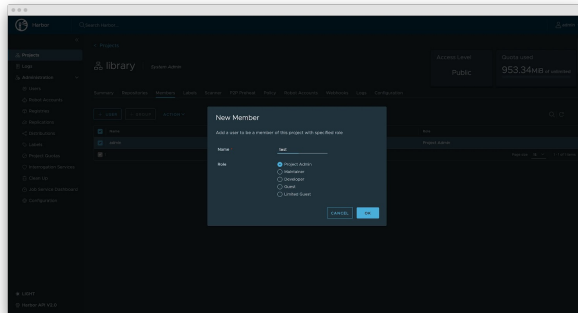
Versioning



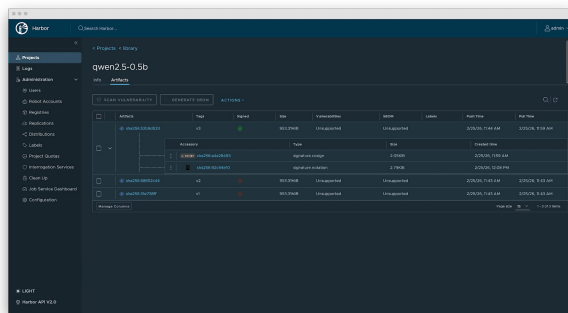
Lifecycle Management



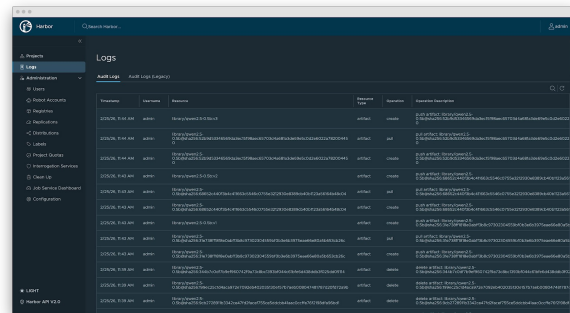
Replication



Access Control



Supply Chain Security



Audit

Model Distribution



KubeCon

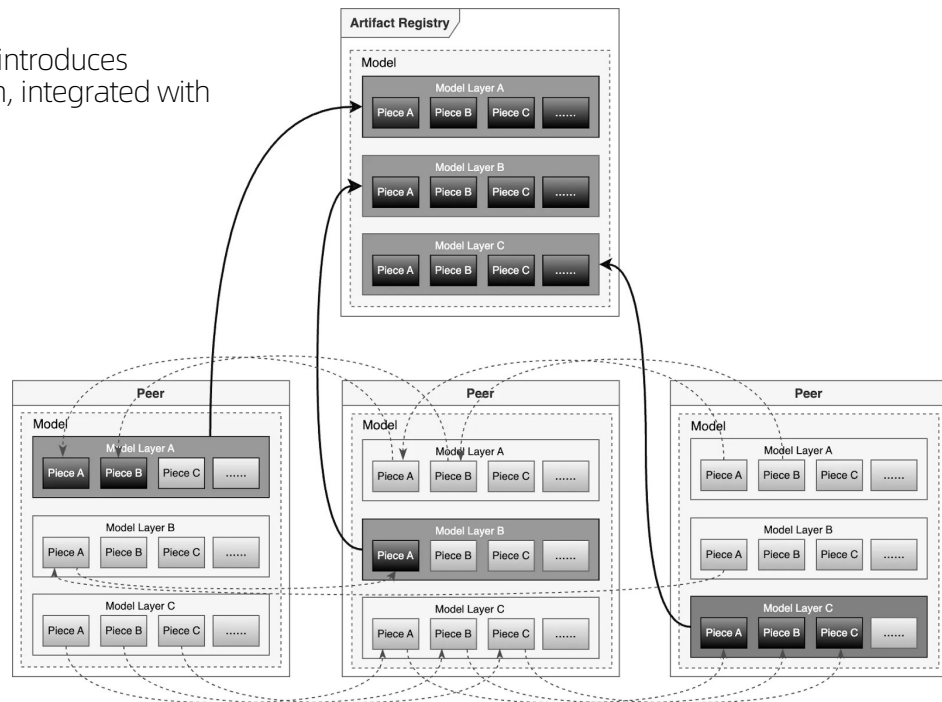
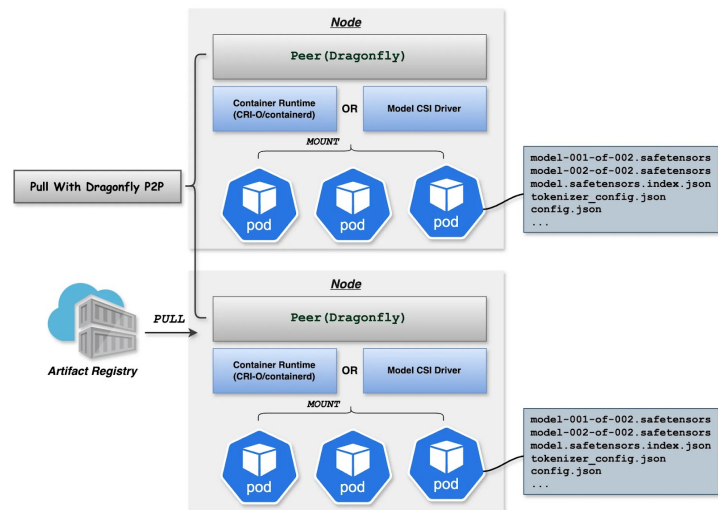


CloudNativeCon

Europe 2026

Delivery

Downloading terabyte-sized model weights directly from the origin introduces bandwidth bottlenecks. We use Dragonfly for P2P-based distribution, integrated with Harbor for preheating.



Model Distribution



KubeCon

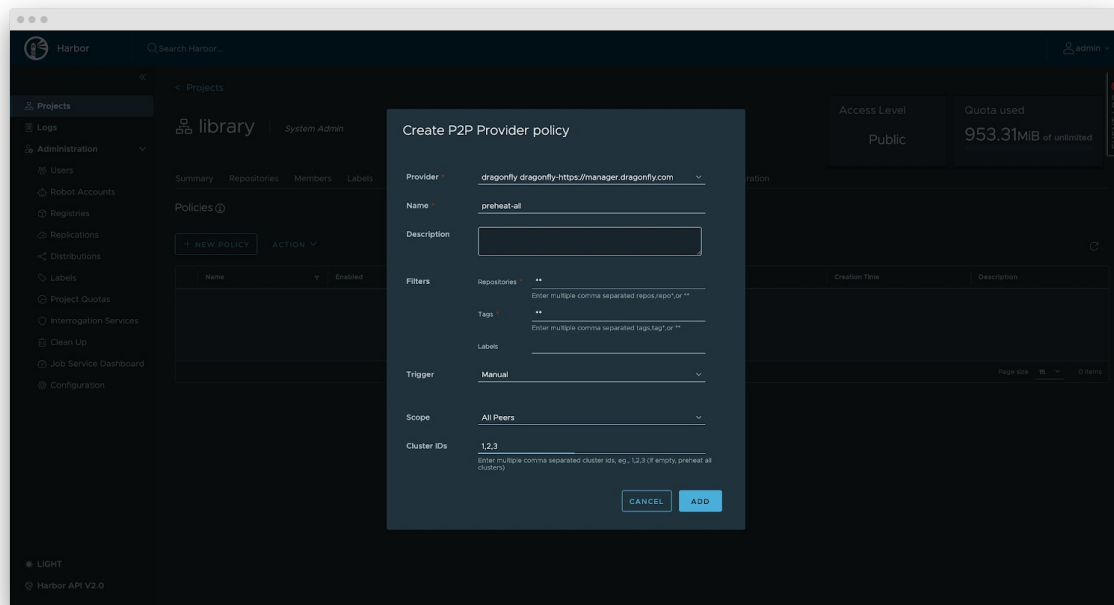


CloudNativeCon

Europe 2026

Delivery(Preheating)

For latency-sensitive inference services, Harbor triggers Dragonfly to distribute and cache data on target nodes before service scaling. When the instance starts, the model loads from the local disk, achieving zero network latency.



Model Distribution



KubeCon

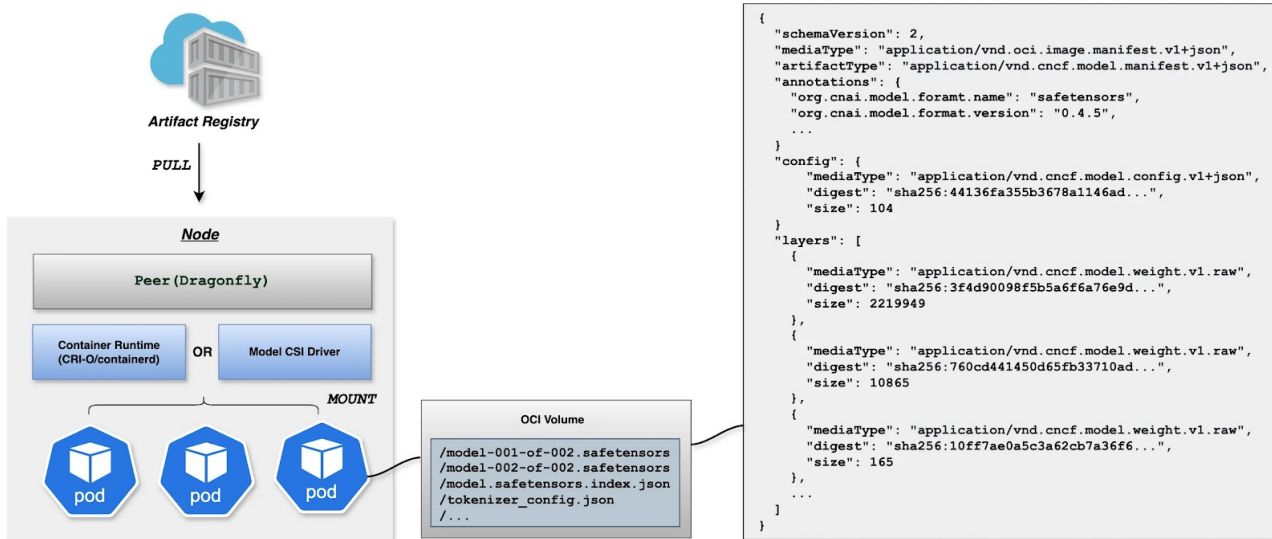


CloudNativeCon

Europe 2026

Deployment

Deployment focuses on separating the Model (Data) from the Inference Engine (Compute). By relying on Kubernetes declarative features, the Engine runs as a Container, while the Model is mounted as a Volume.



Model Distribution



KubeCon



CloudNativeCon

Europe 2026

OCI Volumes (Kubernetes 1.31+)

Native support for mounting OCI artifacts as volumes via CRI-O/containerd. This feature was introduced as **alpha** in Kubernetes 1.31 (requires enabling the **ImageVolume** feature gate) and promoted to **beta** in Kubernetes 1.33 (enabled by default, no feature gate configuration needed).

```
root@instance-20260108-1728:~/kind# k apply -f vllm.yaml
pod/vllm-cpu-inference created
service/vllm-service created
root@instance-20260108-1728:~/kind# k get po
NAME                READY   STATUS    RESTARTS   AGE
vllm-cpu-inference  1/1     Running   0           11s
```

```
[root@vllm-cpu-inference workspace]# curl http://vllm-service:8000/v1/chat/completions \
-H "Content-Type: application/json" \
-d '{
  "model": "/models",
  "messages": [
    {
      "role": "user",
      "content": "Who are u?"
    }
  ],
  "temperature": 0.7,
  "max_tokens": 512
}'
{"id":"chatcmpl-9e3e8b7a8@eccc03e","object":"chat.completion","created":1772522367,"model":"/models","choices":[{"index":0,"message":{"role":"assistant","content":"Hello user, I am an AI system designed to assist with various tasks. My name is System.🟡\n","refusal":null,"annotations":null,"audio":null,"function_call":null,"tool_calls":[],"reasoning":null,"reasoning_content":null,"logprobs":null,"finish_reason":"length","stop_reason":null,"token_ids":null},"service_tier":null,"system_fingerprint":null,"usage":{"prompt_tokens":23,"total_tokens":535,"completion_tokens":512,"prompt_tokens_details":null,"prompt_logprobs":null,"prompt_token_ids":null,"kv_transfer_params":null}}
```

```
apiVersion: v1
kind: Pod
metadata:
  name: vllm-cpu-inference
  labels:
    app: vllm
spec:
  containers:
    - name: vllm
      image: openeuler/vllm-cpu:latest
      command:
        - "python3"
        - "-m"
        - "vllm.entrypoints.openai.api_server"
      args:
        - "--model"
        - "/models"
      volumeMounts:
        - name: model-volume
          mountPath: /models
          readOnly: true
      ports:
      volumes:
        - name: model-volume
          image:
            reference: ghcr.io/chlins/qwen2.5-0.5b:v1
            pullPolicy: IfNotPresent
```

Model Distribution



KubeCon



CloudNativeCon

Europe 2026

Model CSI Driver

For compatibility with Kubernetes 1.31 and older, we offer the Model CSI Driver as an interim solution to mount and deploy models as volumes. As OCI Volumes are slated for **GA** in Kubernetes 1.36, shifting to native OCI Volumes is recommended for the long term.

```
root@instance-20260108-1728:~/kind# k apply -f vllm.yaml
pod/vllm-cpu-inference created
service/vllm-service created
root@instance-20260108-1728:~/kind# k get po
NAME          READY   STATUS    RESTARTS   AGE
vllm-cpu-inference 1/1     Running   0           11s
```

```
[root@vllm-cpu-inference workspace]# curl http://vllm-service:8000/v1/chat/completions \
-H "Content-Type: application/json" \
-d '{
  "model": "/models",
  "messages": [
    {
      "role": "user",
      "content": "Who are u?"
    }
  ],
  "temperature": 0.7,
  "max_tokens": 512
}'
{"id":"chatcmpl-9e3e8b7a80ecc83e","object":"chat.completion","created":1772522367,"model":"/models","choices":[{"index":0,"message":{"role":"assistant","content":"Hello user, I am an AI system designed to assist with various tasks. My name is System.\n","refusal":null,"annotations":null,"audio":null,"function_call":null,"tool_calls":[{"reasoning":null,"reasoning_content":null},"logprobs":null,"finish_reason":"length","stop_reason":null,"token_id":null]},"service_tier":null,"system_fingerprint":null,"usage":{"prompt_tokens":23,"total_tokens":535,"completion_tokens":512,"prompt_tokens_details":null,"prompt_logprobs":null,"prompt_token_ids":null,"kv_transfer_params":null}}
```

```
apiVersion: v1
kind: Pod
metadata:
  name: vllm-cpu-inference
  labels:
    app: vllm
spec:
  containers:
    - name: vllm
      image: openeuler/vllm-cpu:latest
      command:
        - "python3"
        - "-m"
        - "vllm.entrypoints.openai.api_server"
      args:
        - "--model"
        - "/models"
      volumeMounts:
        - name: model-volume
          mountPath: /models
          readOnly: true
      ports:
        - name: model-volume
      volumes:
        - name: model-volume
          csi:
            driver: model.csi.modelpack.org
            volumeAttributes:
              model.csi.modelpack.org/reference:
                ghcr.io/chlins/qwen2.5-0.5b:v1
```



Enhanced Preheating: Allow models to be preheated to specified nodes and querying cache distribution across nodes for model-aware pod scheduling.

Dragonfly RDMA Acceleration: Enable Dragonfly to use RoCE to improve the speed of distribution.

Lazy Loading: Implement on-demand downloading of model weights to reduce startup latency.

containerd Optimization: Enhance the OCI Volumes implementation to reduce decompression overhead for large layers.

Model Security Scanning: Introduce deep scanning feature designed for AI model weights.

Qwen/Qwen2.5-0.5B · Huggi

Harbor

huggingface.co/Qwen/Qwen2.5-0.5B

Hugging Face

Search models, datasets, users...

Models

Datasets

Spaces

Buckets

NEW

Docs

Enterprise

Pricing

Qwen/Qwen2.5-0.5B

📄

👍 like 383

👤 Follow

🌟 Qwen 75.2k

🔥 Text Generation

🤖 Transformers

🛡️ Safetensors

🌐 English

qwen2

conversational

🔗 text-generation-inference

📄 arxiv:2407.10671

📄 License: apache-2.0

📄 Model card

📄 Files and versions

🔗 xet

👤 Community 13

⋮

🚀 Deploy

💻 Use this model

🔗 Edit model card

Downloads last month
1,618,319

📄 Safetensors

Model size 0.5B params

Tensor type BF16

🔗 Chat template

📄 Files info

🔗 Inference Providers

🔗 Text Generation

This model isn't deployed by any Inference Provider.

👤 9 Ask for provider support

📄 Model tree for Qwen/Qwen2.5-0.5B

Adapters367 models

Finetunes541 models

Merges10 models

Quantizations🤖🌐📄📄95 models

🔗 Spaces using Qwen/Qwen2.5-0.5B 100

🔗 pliny-the-prompter/obliteratus

🔗 anzizdaouda0/openaiibox

🔗 eduagarcia/open_pt_llm_leaderboard

🔗 Steveeeeeee/SpeechLLM-Playbook

🔗 FallinAI/Quantize-HF-Models

🔗 openfree/LLM_Quantization

🔗 seawolf2257/LLM_Quantization

🔗 K00B404/LLM_Quantization

🔗 hf-skills/llm-trainer

Qwen2.5-0.5B

Introduction

Qwen2.5 is the latest series of Qwen large language models. For Qwen2.5, we release a number of base language models and instruction-tuned language models ranging from 0.5 to 72 billion parameters. Qwen2.5 brings the following improvements upon Qwen2:

- Significantly **more knowledge** and has greatly improved capabilities in **coding** and **mathematics**, thanks to our specialized expert models in these domains.
- Significant improvements in **instruction following**, **generating long texts** (over 8K tokens), **understanding structured data** (e.g, tables), and **generating structured outputs** especially JSON. **More resilient to the diversity of system prompts**, enhancing role-play implementation and condition-setting for chatbots.
- Long-context Support** up to 128K tokens and can generate up to 8K tokens.
- Multilingual support** for over 29 languages, including Chinese, English, French, Spanish, Portuguese, German, Italian, Russian, Japanese, Korean, Vietnamese, Thai, Arabic, and more.

This repo contains the base 0.5B Qwen2.5 model, which has the following features:

- Type: Causal Language Models
- Training Stage: Pretraining
- Architecture: transformers with RoPE, SwiGLU, RMSNorm, Attention QKV bias and tied word



KubeCon



CloudNativeCon

Europe 2026

Thanks!



Dragonfly Github



KubeCon



CloudNativeCon

Europe 2026

