Dagster Deep Dive

Orchestrating ML Workloads with Dagster & Modal

dagster × Modal

Colton Padden

Developer Advocate @ Dagster

coltonpadden

cmpadden

Charles Frye

AI Engineer Extraordinaire @ Modal

charles_irl

charlesfrye

dagster × Modal

Overview

Machine learning workloads often require a unique set of infrastructure requirements, and **Modal** offers scalable infrastructure to meet these demands, while **Dagster** offers way to robustly orchestrate these pipelines and manage state.

dagster × Modal

Strengths of Dagster

- Full data lineage
- End-to-end state management
  - Partitions
  - Provides context to sub-processes
- Resiliency through observability and alerts
  - Checks, Alerts, Retries

dagster × Modal

Strengths of Modal

- Self-provisioning runtime
- Blazingly fast custom container stack
- Auto-scaling distributed apps without all the agonizing pain
- GPUs, but only when you want them

dagster × Modal

Modal is a self-provisioning serverless runtime.

compute+storage

```
@app.function(
    image=image,
    gpu=modal.gpu.A100(
        count=1, size="80GB"
    ),
    volumes={MODEL_DIR: volume},  # stores fine-tuned model
    timeout=1800,  # 30 minutes
    secrets=[
        modal.Secret.from_name("my-wandb-secret"),
        modal.Secret.from_name("huggingface"),
    ]
    if USE_WANDB
    else [modal.Secret.from_name("huggingface")],
)
def train(instance_example_urls, config):
```

URL+deployment

```
@app.function(
    image=image,
    concurrency_limit=1,
    allow_concurrent_inputs=1000,
    ...
)
@modal.asgi_app()
def fastapi_app():
    import gradio as gr
```

modal.com/docs/examples/dreambooth_app

dagster × Modal

Stronger Together

- Developer-friendly, cloud-native orchestration on top of a developer-friendly, cloud-native infrastructure tool
- Focus on your work, not your infra

dagster × Modal

Several hours of yapping

“That’s crazy man.

Do you believe in aliens?”

“No.”

dagster × Modal

Making artificial intelligence practical, productive & accessible to everyone

Chris Benson

Daniel Whitenack

---

colton@dagsterlabs.com <colton@dagsterlabs.com>
to me

3:01 PM (0 minutes ago)

Podcaset Summary

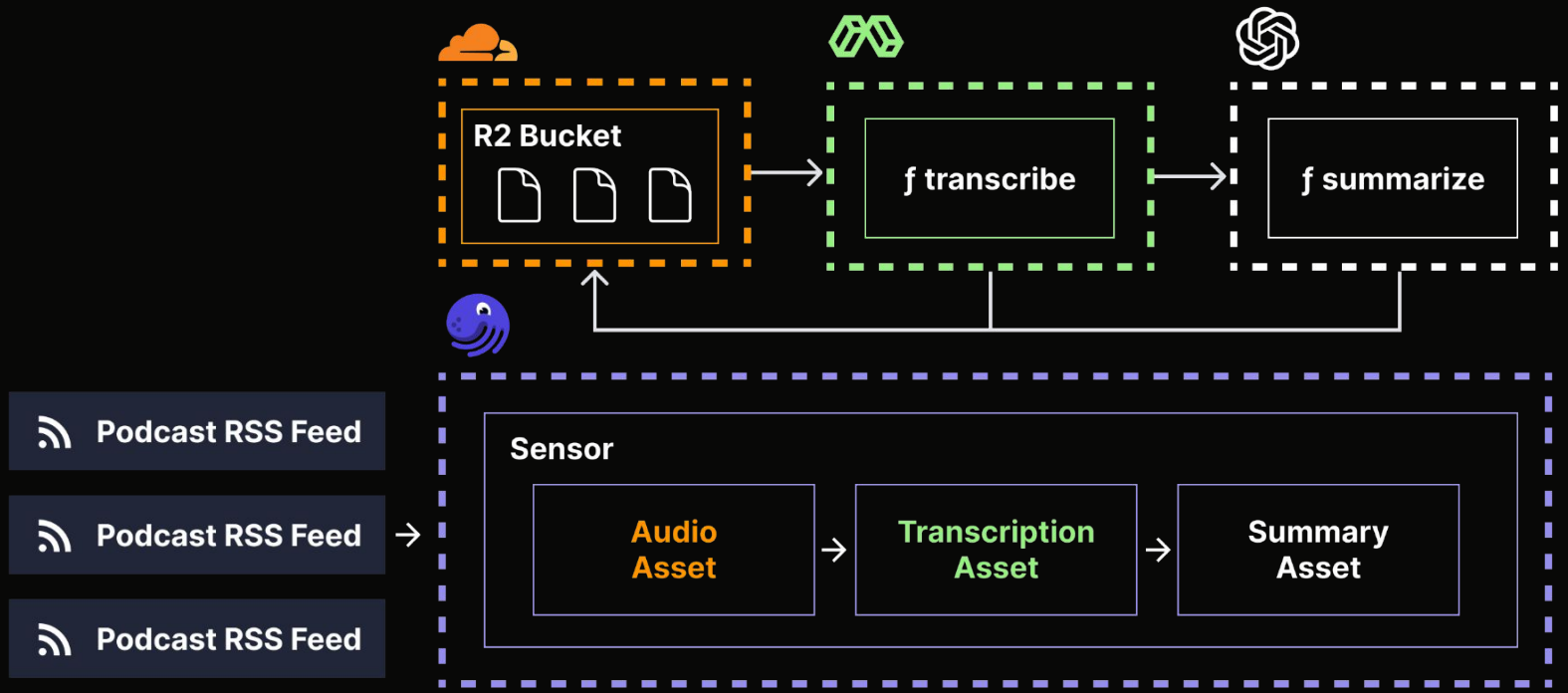
changelog_com_7_2554

...

Practical AI is a podcast for those involved or interested in artificial intelligence, discussing its impact and developments. The show is supported by partners like fly.io, which offers microvms for apps, and SpeakEasy, providing a platform for API development with features like SDK generation and API testing. Hosts Daniel Weitnack and Chris Benson, along with guest Dennis Cruz, discuss the security aspects of AI, emphasizing the need for deterministic AI systems that can be trusted and verified. They explore how AI can be used to enhance cybersecurity and the importance of understanding and controlling the interaction between data and AI models to prevent vulnerabilities.

Reply Forward

dagster × Modal



dagster × Modal

github.com/dagster-io/dagster-modal-demo

dagster × Modal

Project Structure

```
$ tree -L 2
.
├── Makefile
├── README.md
├── dagster_modal_demo
│   ├── __init__.py
│   ├── constants.py
│   ├── dagster_modal/
│   ├── definitions.py
│   ├── pipeline_factory.py
│   ├── resources.py
│   └── utils/
├── modal_project
│   ├── config.py
│   └── transcribe.py
├── pyproject.toml
├── setup.cfg
└── setup.py
```

Integration

Pipeline / Assets

Modal Script

dagster × Modal

Pipeline Factory and Sensors

- A pipeline is created for each RSS feed
- A sensor polls each RSS feed for entries
  - An `etag` is used to only fetch new podcasts
- If a new entry is found, a run request is triggered to materialize our pipelines
  - Download audio, Transcribe, Summarize, and E-mail

dagster × Modal

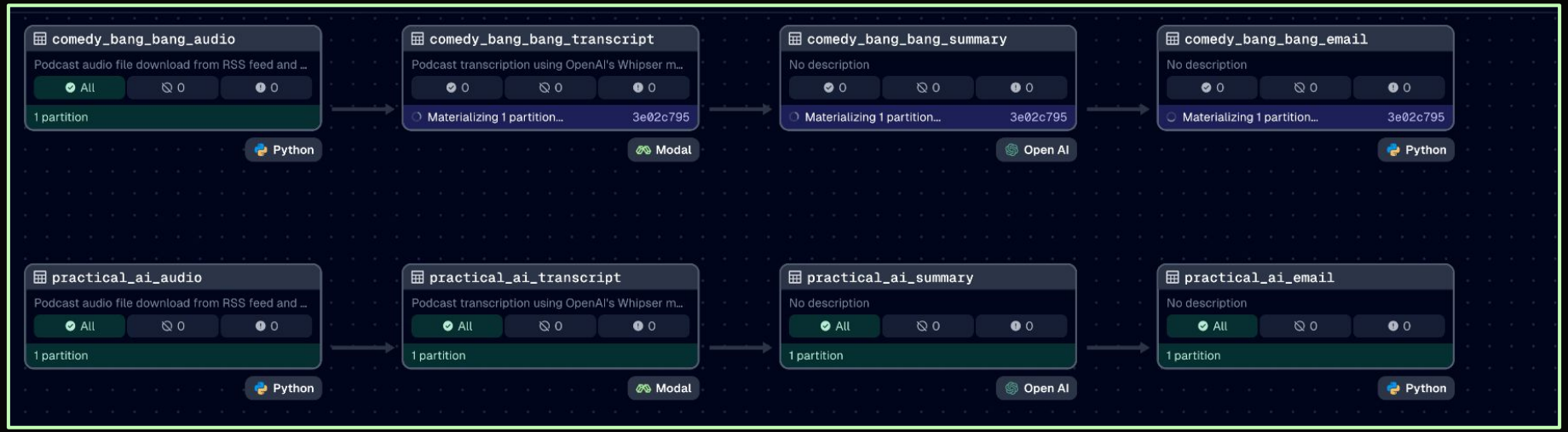
dchhaddendum.libsyn.com/rss


```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0" xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cc="http://web.resource.org/cc/" xmlns:itunes="http://ww
>-------<channel>
>-------->-------<atom:link href="https://dchhaddendum.libsyn.com/rss" rel="self" type="application/rss+xml"/>
>-------->-------<title>Dan Carlin's Hardcore History: Addendum</title>
>-------->-------<pubDate>Thu, 01 Aug 2024 05:59:00 +0000</pubDate>
>-------->-------<lastBuildDate>Thu, 12 Sep 2024 06:45:23 +0000</lastBuildDate>
>-------->-------<generator>Libsyn WebEngine 2.0</generator>
>-------->-------<link>http://dchhaddendum.libsyn.com/website</link>
>-------->-------<language>en</language>
>-------->-------<copyright><![CDATA[dancarlin.com]]></copyright>
>-------->-------<docs>http://dchhaddendum.libsyn.com/website</docs>
>-------->-------<managingEditor>dan@dancarlin.com (dan@dancarlin.com)</managingEditor>
>-------->-------<itunes:summary><![CDATA[Interviews, musings and extra material from the makers of Dan Carlin's Hardcore Hi
>-------->-------<image>
>-------->-------->-------<url>https://static.libsyn.com/p/assets/6/b/f/e/6bfe939ed4336498/HHA-1400px_b.jpg</url>
>-------->-------->-------<title>Dan Carlin's Hardcore History: Addendum</title>
>-------->-------->-------<link><![CDATA[http://dchhaddendum.libsyn.com/website]]></link>
>-------->-------</image>
>-------->-------<itunes:author>Dan Carlin</itunes:author>
>-------->-------<itunes:keywords>History,ancient,archival,classical,discussion,military,modern,rome,war</itunes:keywords>
>-------->-------<itunes:category text="History">
>-------->-------</itunes:category>
>-------->-------<itunes:image href="https://static.libsyn.com/p/assets/6/b/f/e/6bfe939ed4336498/HHA-1400px_b.jpg" />
>-------->-------<itunes:explicit>false</itunes:explicit>
>-------->-------<itunes:owner>
>-------->-------->-------<itunes:name><![CDATA[Dan Carlin]]></itunes:name>
>-------->-------->-------<itunes:email>dan@dancarlin.com</itunes:email>
>-------->-------</itunes:owner>
>-------->-------<description><![CDATA[Interviews, musings and extra material from the makers of Dan Carlin's Hardcore Histo
>-------->-------<itunes:type>episodic</itunes:type>
>-------->-------



>-------->-------<podcast:locked owner="dan@dancarlin.com">no</podcast:locked>
>-------->-------
>-------->-------<item>
>-------->-------->-------<title>EP30 So, you say you want a revolution?</title>
```

dagster × Modal



dagster × Modal

Filter

Type an asset subset... (ex: practical_ai_audio+)

Materialize sele

practical_ai_summary

No description

All 0 0

1 partition

Open AI

pra

No desc

1 partiti

practical_ai_summary

View in Asset Catalog

Latest materialization

Run	Run 91caa8f3 View logs practical_ai_job @ 728e6d9f practical_ai_summary
Partition	changelog_com_7_2565
Timestamp	Sep 17, 3:12 PM
summary	Practical AI is a podcast for those involved or interested in AI, featuring discussions on how AI technology is transforming the world. In a recent episode, Dylan Fox, founder and CEO of Assembly AI, shared insights on their speech AI models which convert voice data into text and extract valuable information from it. Assembly AI offers a simple API that developers can use for free initially to build applications that leverage voice data. The episode also touched on the vast opportunities for developers given the abundance of voice data available online. Additionally, the podcast hosts, Daniel White-Nack and Chris Benson, discussed various AI topics and the importance of staying updated in the fast-evolving field of AI. They also highlighted the role of community platforms in learning and sharing AI-related knowledge.
summary_key	data/changelog_com_7_2565-summary.txt
openai.calls	1
openai.total_tokens	9421
openai.prompt_tokens	9272
openai.completion_tokens	149

Materialization tags

Metadata plots

openai.calls

2.0

dagster × Modal

Overview of Pipes

- Pass context to subprocesses with **Pipes**
  - Environment variables
  - Dagster `context`
- Wrapper around `subprocess`
- The subprocess can emit events
  - Materialization results
  - Structured logs

dagster × Modal

Overview of Modal Code

- Infrastructure-from-code
  - Container images from method chaining
  - Hardware from `@decorators`

- Plan of attack
  - Read audio from R2, segment it
  - Fan out over segments
    - Transcribe with Whisper
  - Upload back to R2

```python
app_image = (
    modal.Image.debian_slim(python_version="3.10")
    .apt_install("git")
    .pip_install(
        "git+https://github.com/openai/whisper.git",
        "dacite",
        "jiwer",
        "ffmpeg-python",
        "gql[all]~=3.0.0a5",
        "python-multipart~=0.0.9",
        "pandas",
        "loguru==0.6.0",
        "torchaudio==2.1.0",
        "python-dotenv",
    )
    .apt_install("ffmpeg")
    .pip_install("ffmpeg-python")
)
```

```python
@app.function(
    image=app_image,
    timeout=900,
    volumes={
        "/mount": cloud_bucket_mount,
    },
)
def transcribe_episode(
```

```python
@app.function(
    image=app_image,
    cpu=2,
    timeout=400,
    volumes={
        "/mount": cloud_bucket_mount,
    },
)
def transcribe_segment(
```

dagster × Modal

<code exploration>

dagster × Modal

?s

Next Steps & Resources

Join the Dagster Slack

Connect with other data practitioners. Share knowledge or find help

dagster.io/slack

Sign up for Dagster Cloud

Sign up for Dagster Cloud and get started with a free 30 day trial

dagster.io

Sign up for Modal

Get up and running in no-time with Modal's blazing fast container stack. $30/month of free compute!

modal.com

Join the Modal Slack

Ask Modal questions, get Modal answers. Talk nerdy about GPUs and more.

modal.com/slack