

# Enabling Community Education

*with a little help from AI*

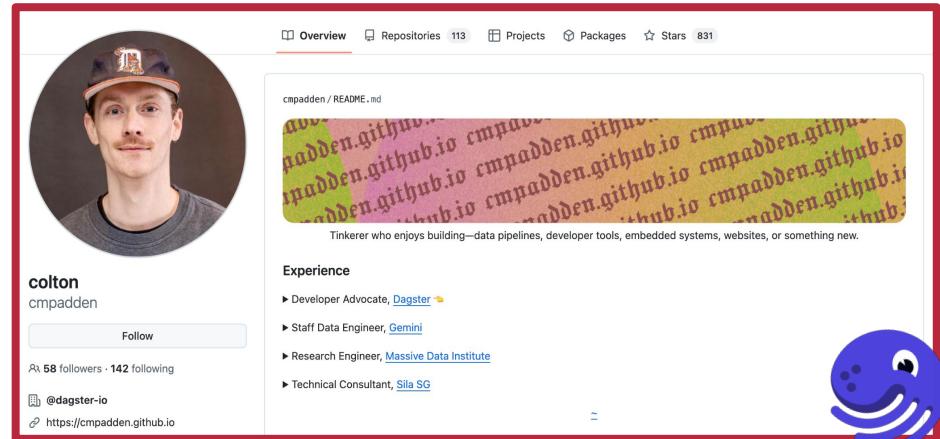


OCTOBER 12-14  
Raleigh, NC USA



# About me

- Creating educational content, and hosting webinars
- Supporting and voicing the communities' needs
- Dogfooding the product
- Feature development



Developer Advocate, **Dagster Labs**



OCTOBER 12-14 • Raleigh, NC USA

# What is... Dagster?

Screenshot of the GitHub repository for Dagster (dagster/dagster). The repository is public and has 3624 branches and 3898 tags. The master branch is selected. The repository has 25,529 commits.

Commit	Message	Date
.buildkite	Give dagster-dg-cli more splits (#32353)	4 days ago
.claude	[claude] adds command for resolving github issue ...	4 days ago
.github	[examples] remove legacy getting_started_etl_tut...	2 weeks ago
.tmp	[pex] print stdout/stderr on package build failure (...	last month
.vscode	[pyright] explicitly set typeCheckingMode (#31010)	4 months ago
docs	DOC-1350 DOC-1488 DOC-1487 DOC-1260 Add ...	2 days ago
examples	Consistent dg in examples (#32398)	2 days ago
helm/dagster	[uv] bump pin, remove retry script (#32254)	2 weeks ago
integration_tests	[uv] bump pin, remove retry script (#32254)	2 weeks ago

**About**

An orchestration platform for the development, production, and observation of data assets.

[dagster.io](#)

python metadata workflow  
data-science etl analytics  
scheduler orchestration  
data-engineering data-integration  
data-pipelines hacktoberfest  
workflow-automation mlops dagster  
data-orchestrator

[Readme](#)  
[Apache-2.0 license](#)  
[Code of conduct](#)

# What is... Dagster?

```
import dagster as dg
import pandas as pd

from sklearn.linear_model import LinearRegression

@dg.asset
def country_populations() -> pd.DataFrame:
    df = pd.read_html("https://tinyurl.com/mry64ebh")[0]
    df.columns = ["country", "pop2022", "pop2023", "change", "continent", "region"]
    df["change"] = df["change"].str.rstrip("%").astype("float")
    return df

@dg.asset
def continent_change_model(country_populations: pd.DataFrame) -> LinearRegression:
    data = country_populations.dropna(subset=["change"])
    return LinearRegression().fit(pd.get_dummies(data[["continent"]]), data["change"])

@dg.asset
def continent_stats(country_populations: pd.DataFrame, continent_change_model: LinearRegression):
    result = country_populations.groupby("continent").sum()
    result["pop_change_factor"] = continent_change_model.coef_
    return result
```



OCTOBER 12-14 • Raleigh, NC USA

# Features

- Scheduling
- Resiliency via retries / memoization
- Observability
  - Alerts
  - Lineage
  - Cataloging / Self-documentation



OCTOBER 12-14 • Raleigh, NC USA

*Dagster is a **framework** for building workflows in a **declarative** way.*



OCTOBER 12-14 • Raleigh, NC USA

# Community

September 7, 2025 – October 7, 2025 Period: 1 month

Overview

284 Active pull requests      112 Active issues

↳ 218 Merged pull requests	↳ 66 Open pull requests	⌚ 76 Closed issues	⌚ 36 New issues
----------------------------	-------------------------	--------------------	-----------------

## Awesome Dagster

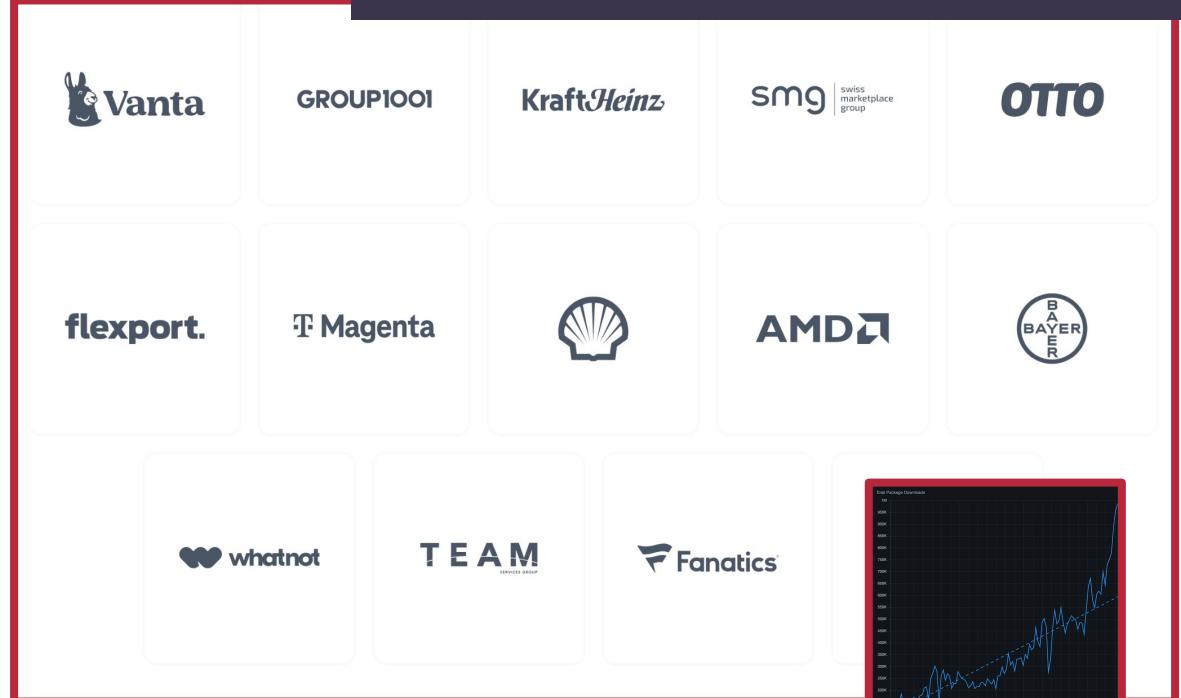


contributors 15 commit activity 0/month awesome

### Awesome Dagster

A collection of all things awesome related to the [Dagster](#) ecosystem!

- [Learning Material](#)
  - [Articles](#)
  - [Talks](#)
  - [Projects](#)
- [Discussion](#)
- [Community](#)
- [Companies Using Dagster](#)



Vanta

GROUP1001

KraftHeinz

smg swiss marketplace group

OTTO

flexport.

Magenta

AMD

BAYER

whatnot

TEAM

Fanatics



OCTOBER 12-14 • Raleigh, NC USA



# Community

# ask-community This channel is for questions, discussions, and troubleshooting for Dagster Open Source. Cha...

Messages Sem título Bookmarks Files Pins +

Elling Hofland 10:33 AM

If I have a dbt project where there are multiple targets that I want to be used in different assets, how can I use Dagster components to define these dbt core definitions? I've tried adding translation keys in the defs.yaml files for each component but no matter what I do I continue to get conflicting node definitions.

Aslak N 5:14 AM

Hi, I'm observing  
dg.DataVersion  
more like 1000+  
2 replies

Mycchaka K 11:52 AM

How are people?  
We have many  
We also have  
What's the re...

2 replies



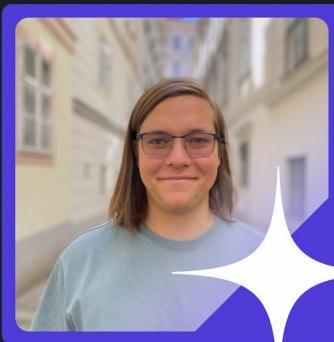
Zach Romer

Lead Data Platform Engineer at Empirico  
he/him · Zack



geoHeil

Dr



Daniel Gafni

MLOps

A 11:52 AM

Hi, I have been trying to use Dagster Ray in my data pipeline but I have been very frequently getting this error. Can I get some help regarding this:  
"ConnectionError: ray client connection timeout" Trying to use a KubeRayCluster mode in dagster ray.

Mycchaka Kleinbort 11:52 AM

Can I get some eyes on this issue?

<https://github.com/dagster-io/dagster/issues/3236>



14,831

# *The problem*

on Feb 25

...

This was a MAJOR blocker that took months to resolve as this solution was not documented anywhere. The delay and inability to debug this in a timely manner led to my company to forego adopting Dagster... :(.

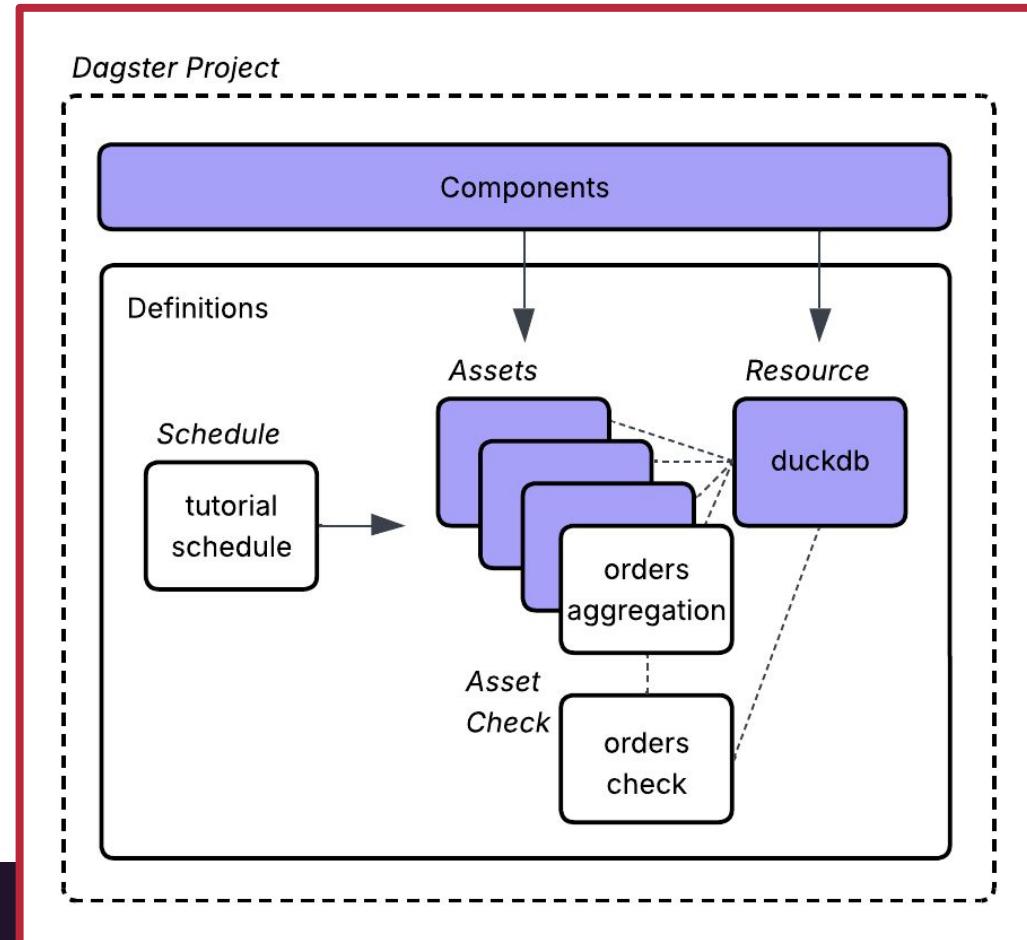
...



OCTOBER 12-14 • Raleigh, NC USA

# Complexity

- Many use cases
  - ML
  - DE
  - Data science
- Frameworks evolve
- Learning curve of concepts both with Dagster and Python



# Personas

## BALLMER PEAK

<

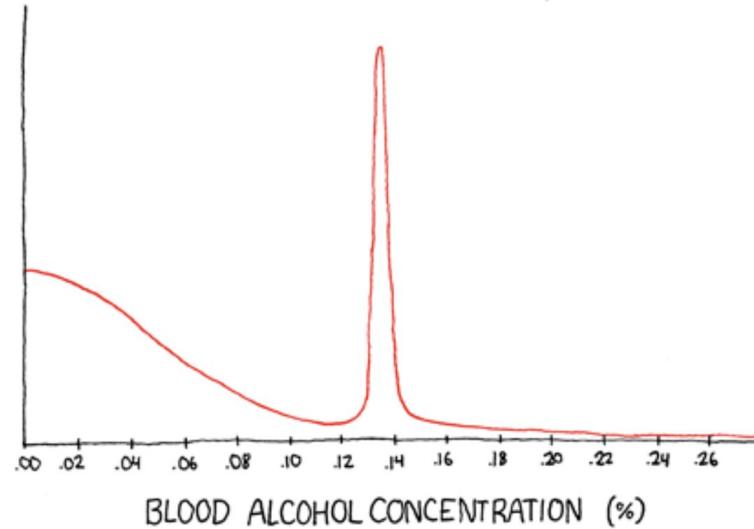
< PREV

RANDOM

NEXT >

>

PROGRAMMING  
SKILL



<https://xkcd.com/323/>



# The team

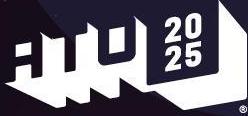
- 1x technical writer
- 3x developer advocates

*\* with community and engineer support*



OCTOBER 12-14 • Raleigh, NC USA

# The strategy



OCTOBER 12-14  
*Raleigh, NC USA*

# Our strategy

- Content audit
- Revise information architecture
- Ensure we meet all personas needs
- Improve feedback loops & monitor users
- Scale



OCTOBER 12-14 • Raleigh, NC USA

# Information architecture

- Recently revamped information architecture
  - Audited and re-organized *all* content
  - Modeled documentation after the software development lifecycle
- Migrated to Docusaurus to focus on content

[← Back to main menu](#)

Getting Started

Overview

Quickstart

Installation

Concepts

Python primer 

Tutorial

Build

Automate

Operate

Log & debug

Observe

Test

Labs

About



OCTOBER 12-14 • Raleigh, NC USA



# Welcome to Dagster! #

## Getting started

What's Dagster?

Quickstart

Installation

Creating a new project

Getting help

Tutorial

Concepts

Dagster+

Open Source Deployment

Integrations

Guides

API Reference

About

Dagster is an orchestrator that's designed for developing and maintaining data assets, such as tables, data sets, machine learning models, and reports.

You declare functions that you want to run and the data assets that those functions produce or update. Dagster then helps you run your functions up-to-date.

Dagster is designed to be used at every stage of your data engineering workflow, from local development, unit tests, integration tests, to CI/CD.

**New to Dagster?** Check out the [Quickstart](#), learn about [Dagster](#), and explore [Concepts](#). For an in-depth learning experience, check out the [Dagster University](#).

[Quickstart](#)

[View Tutorials](#)

[Explore Concepts](#)

## Join the community #

**Questions? Interested in seeing some community in action?** Check out the [Dagster community!](#)

- [Dagster Slack](#)
- [Community Q&A](#)
- [YouTube](#)

## On This Page

Welcome to Dagster!  
[Join the community](#)  
[Examples](#)



## User guide

Examples Deployment Migration Integrations API

Latest (1.11.13) ▾



Search

Try Dagster+

### Getting Started

#### Overview

Quickstart  
Installation  
Concepts  
Dagster University   
Python primer

#### Tutorial

Projects  
Assets  
Resources  
Asset dependencies  
Asset checks  
Automation  
Components

#### Build

Projects  
Assets  
Components  
Partitions and backfills  
External resources  
External pipelines (Dagster Pipes)  
I/O managers  
ML pipelines

Getting Started > Overview

## Welcome to Dagster

Dagster is a data orchestrator built for data engineers, with integrated lineage, observability, a declarative programming model, and best-in-class testability.

defs/assets.py

```
import dagster as dg

@dg.asset
def hello(context: dg.AssetExecutionContext):
    context.log.info("Hello!")

@dg.asset(deps=[hello])
def world(context: dg.AssetExecutionContext):
    context.log.info("World!")
```



OCTOBER 12-14 • Raleigh, NC USA



## Getting Started

[Overview](#)[Quickstart](#)[Installation](#)[Concepts](#)[Python primer](#) 

## Tutorial

[Projects](#)[Assets](#)[Resources](#)[Asset dependencies](#)[Asset checks](#)[Automation](#)[Components](#)

## Build

[Projects](#)[Assets](#)[Components](#)[Partitions and backfills](#)[External resources](#)[External pipelines \(Dagster Pines\)](#)

Getting Started &gt; Overview

# Welcome to Dagster

Dagster is a data orchestrator built for data engineers, with integrated lineage, observability, a declarative programming model, and best-in-class testability.

defs/assets.py

```
import dagster as dg

@dg.asset
def hello(context: dg.AssetExecutionContext):
    context.log.info("Hello!")

@dg.asset(deps=[hello])
def world(context: dg.AssetExecutionContext):
    context.log.info("World!")
```

# Documentation



Examples Dagster Deep Dives Customer case studies Full pipeline examples Dagster Open Platform Hooli example pipelines Bluesky data analysis Dagster integration with dbt Solving NYT Connections with DsPy Dagster ETL pipeline LLM fine-tuning with OpenAI Machine learning with PyTorch Podcast transcription with Modal Prompt engineering with Anthropic Retrieval-augmented generation (RAG) with Pinecone Mini examples 

Dynamic fanout

Dynamic outputs vs Python parallelism

Resource caching

Sharing code across code

Examples &gt; Full pipeline examples

## Full pipeline examples

Dagster Open Platform



Bluesky data analysis

4 items

Hooli example pipelines



Dagster integration with dbt

6 items



Solving NYT Connections with DsPy

4 items



Dagster ETL pipeline

8 items



LLM fine-tuning with OpenAI

5 items



Machine learning with PyTorch

3 items



Podcast transcription with Modal

3 items



Retrieval-augmented generation (RAG)

# Examples



## Overview ▾

[API reference](#)[API lifecycle stages](#)[Filtering API lifecycle warnings](#)

## CLI reference ▾

## Dagster core SDK ▾

[asset checks](#)[assets](#)[components](#)[config](#)[definitions](#)[dynamic mapping & collect](#)[errors](#)[execution](#)[external assets instance api](#)[graphs](#)[hooks](#)

Dagster core SDK &gt; components

# Components

## Using Components

### @dagster.component\_instance

[\[source\]](#)

Decorator for a function to be used to load an instance of a Component. This is used when instantiating components in python instead of via yaml.

Example:

```
import dagster as dg

class MyComponent(dg.Component):
    ...

    @dg.component_instance
    def load(context: dg.ComponentLoadContext):
        return MyComponent(...)
```

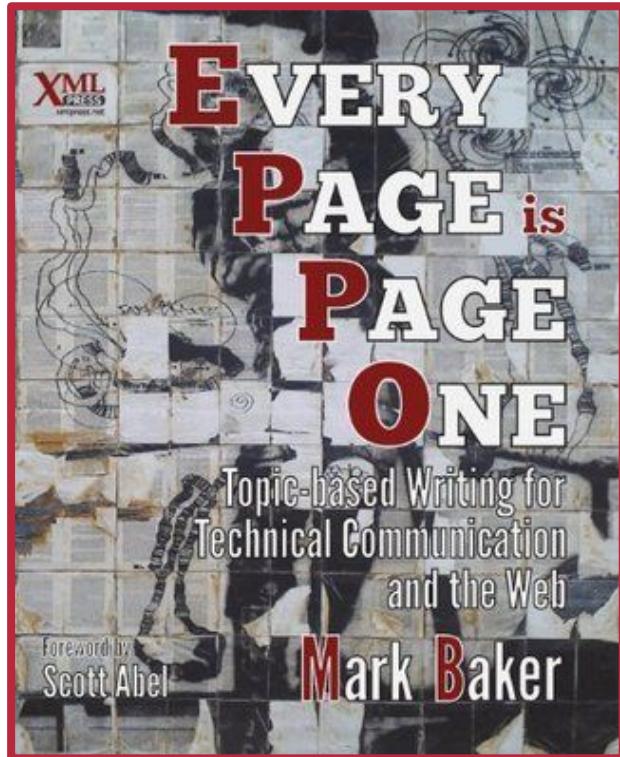
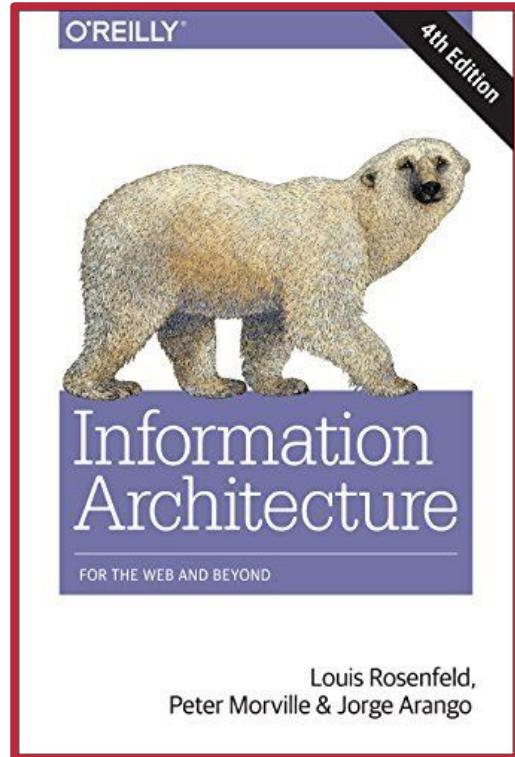
# API References

### class dagster.ComponentLoadContext

Context object that provides environment and path information during component loading. This context is automatically created and passed to component definitions when loading a project's defs folder. Each Python

# HOW TO MAKE SENSE OF ANY MESS

**Abby Covert**



OCTOBER 12-14 • Raleigh, NC USA

# Content (for more personas!)



OCTOBER 12-14  
*Raleigh, NC USA*

# Python Packages: a Primer for Data People (part 1 of 2)

March 6, 2023



Elliot Gunn

A collage of various Python-related images and code snippets. It includes the Dagster logo, a Python snake icon, a dark-themed Python logo, and several snippets of Python code. One snippet shows a class hierarchy with methods f and g. Another snippet shows a generator function foo. A third snippet shows a while loop with a yield statement.

dagster

Python Packages

a primer for

Part 1 of 2

class A:  
 def f(self) -> int: # Type of self inferred (A)  
 return 2

class B(A):  
 def f(self) -> int:  
 return 3  
 def g(self) -> int:  
 return 4

def foo(a: A) -> None:

while a < n:  
 yield a

**Blogs**

# Download the e-book primer on how to build data platforms

Learn the fundamental concepts to build a data platform in your organization; covering common design patterns for data ingestion and transformation, data modeling strategies, and data quality tips.



- Tips and tricks for data modeling and data ingestion patterns
- Explore the benefits of an observation layer across your data pipelines
- Learn the key strategies for ensuring data quality for your organization

Email\*

name@company.com

First name\*

First name

Last name\*

Last name

Get the eBook

Books

# Realest of real world examples

 dagster-open-platform Public

 main  1 Branch  0 Tags

Q Go to file t

 elementl-devtools	Mirroring commit 0676c00cfb5847603e2b4d4c9ec10a3c586...		1f43f68 · 4 days ago	 675 Commits
 dagster_open_platform	Mirroring commit 0676c00cfb5847603e2b4d4c9ec10a3c586...		4 days ago	
 dagster_open_platform_dbt	Mirroring commit b452c96287b533e13b5ec68374647b8...		5 days ago	
 dagster_open_platform_tests	Mirroring commit a10e5b01d02199b35c73afbfebb2e24b...		2 weeks ago	
 Makefile	Mirroring commit 8343d39a143b15e3d17eaeee432a734a5...		5 months ago	
 README.md	Mirroring commit 0ee0276c5788ac0db8e3c8478c2d685...		last year	
 dogfood_master_packages.txt	Mirroring commit 1062412fa493e7c77133f2337cd5d7f0a...		7 months ago	
 pyproject.toml	Mirroring commit 2f94d091156c7f26ade3f130c583a187e...		2 months ago	
 uv.lock	Mirroring commit 2f94d091156c7f26ade3f130c583a187e...		2 months ago	

 README



## Dagster Open Platform

This repository contains the Dagster Open Platform (DOP) project.

Dagster Labs' open-source data platform, built with Dagster.

 [www.dagster.io](http://www.dagster.io)

 Readme

 Activity

 Custom properties

 398 stars

 8 watching

 35 forks

Report repository

Contributors 2

 elementl-devtools Dagster Labs

 tacastillo Tim Castillo

Languages

# Courses



## Dagster & ETL

Dagster & ETL **1,866 completions**



★ 5.0 (1 review)

Learn how to ingest data to power your assets. You'll build custom pipelines and see how to use Embedded ETL and Dagster Components...



## Dagster Essentials

Dagster Essentials **16,502 completions**



★ 4.8 (160 reviews)

In this course, learn the basics of Dagster, including how to represent a data pipeline as the data assets it produces and orchestrate a...



## Testing with Dagster

Testing with Dagster **1,068 completions**



★ 5.0 (4 reviews)

In this course, learn best practices for testing, including unit tests, mocks, integration tests and applying them to Dagster.



OCTOBER 12-14 • Raleigh, NC USA

*AI makes it easy to **translate**  
between content mediums.*



OCTOBER 12-14 • Raleigh, NC USA

# Videos

Home Videos Shorts Live Playlists Posts



## Dagster Team Product Demos ► Play all

dagster PROD DEMO Dagster+ DataOps Platform Walkthrough Christian Minich   Sales Engineer	dagster Better Feedback Loops Dagster	dagster Let's build a Databricks component! Dagster	dagster Data Platform Speed Run Dagster	dagster Authoring Components Demo Dagster	dagster Onboarding to a Dagster Data Platform with Components Cotton Padden   Developer Advocate   Dagster
How to Build a Modern Data Platform: Dagster+ Demo... Dagster 558 views • 13 days ago CC	New Dagster+ UI Features for Enhanced Observability in... Dagster 458 views • 3 weeks ago CC	Let's build a Databricks Component! Dagster 552 views • 2 months ago CC	End-to-end Dagster Data Platform Speedrun Dagster 3.5K views • 4 months ago CC	Streamlining PDF Text Extraction with Dagster... Dagster 996 views • 5 months ago CC	Leveraging Dagster MCP Server with Claude Code to... Dagster 776 views • 5 months ago CC

## Dagster Deep Dives ► Play all

A series of talks, presentations and demos on how to get the most out of your data platform.

dagster Deep Dive Platfrom Engineering with Dagster at Empirico Zach Romer (Empirico Inc.), Cotton Padden (Dagster), Alex Nooan (Dagster)	dagster Deep Dive Dagster Running Dagster: Event Driven Pipelines Nick Reach (Dagster), Cotton Padden (Dagster), Alex Nooan (Dagster)	dagster Deep Dive Rapidly developing ELT pipelines with dltHub and Dagster Alana Astrakhanseva (dltHub), Nick Reach (Dagster), Cotton Padden (Dagster), Alex Nooan (Dagster)	dagster Deep Dive Breaking Biology's Data Wall Scaling a Multi-disciplinary Data Platform at Basecamp Research Keith Kim (Basecamp Research), Alana Astrakhanseva (dltHub), Cotton Padden (Dagster), Alex Nooan (Dagster)	dagster Deep Dive Accelerating MLOps Maturity with Neurospace Ravous Shireishi (Neurospace), Maria Hvid (Neurospace), Martin Jorgensen (Neurospace), Rene Petersen (Neurospace), Alex Nooan (Dagster)	dagster Deep Dive Dagster Running Dagster: Our Open Platform Nick Reach (Dagster), Cotton Padden (Dagster), Alex Nooan (Dagster)
--	--	---	---	--	---

# *Example flow*

YouTube Video

Documentation

Example Projects

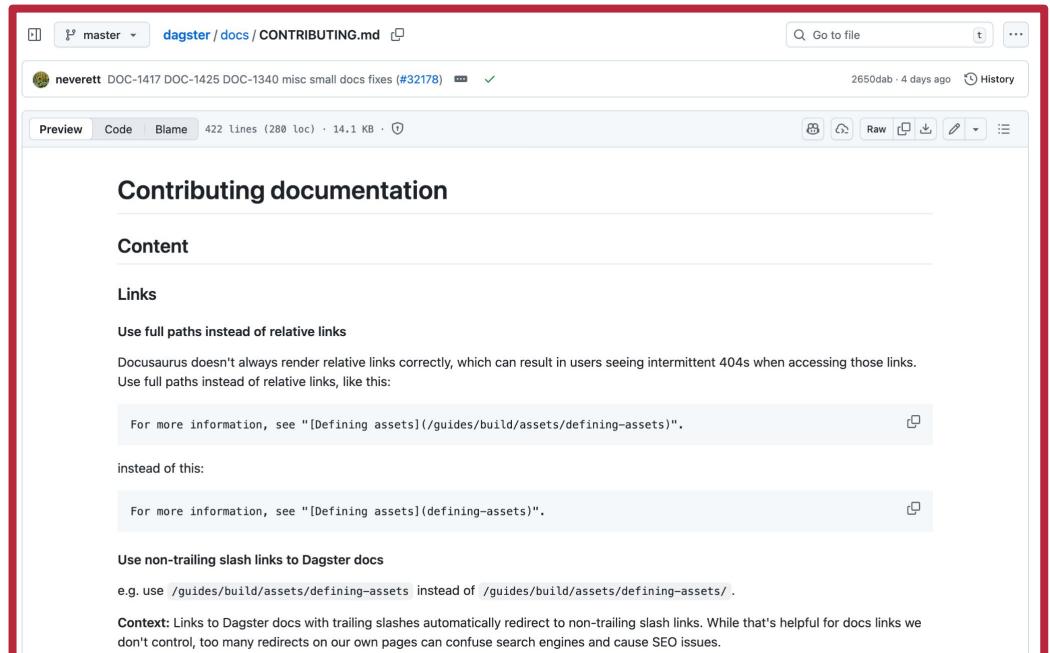
Dagster University



OCTOBER 12-14 • Raleigh, NC USA

# Scaling through collaboration

- Enable contributors internal & external
- Hosted a docs-week for an all hands on deck overhaul
- And helpful for AI!



The screenshot shows a GitHub pull request page for the file `CONTRIBUTING.md` in the `dagster / docs` repository. The pull request is titled "DOC-1417 DOC-1425 DOC-1340 misc small docs fixes (#3217B)". The code tab is selected, showing the following content:

```
Contributing documentation
```

---

### Content

#### Links

Use full paths instead of relative links

Docusaurus doesn't always render relative links correctly, which can result in users seeing intermittent 404s when accessing those links. Use full paths instead of relative links, like this:

For more information, see "[Defining assets](/guides/build/assets/defining-assets)".

instead of this:

For more information, see "[Defining assets](defining-assets)".

#### Use non-trailing slash links to Dagster docs

e.g. use `/guides/build/assets/defining-assets` instead of `/guides/build/assets/defining-assets/`.

**Context:** Links to Dagster docs with trailing slashes automatically redirect to non-trailing slash links. While that's helpful for docs links we don't control, too many redirects on our own pages can confuse search engines and cause SEO issues.



OCTOBER 12-14 • Raleigh, NC USA

# Scaling even more w/ AI



OCTOBER 12-14  
Raleigh, NC USA

 GitHub discussion

\* #ask-ai question

 Google analytics



**create / refine  
content**



OCTOBER 12-14 • Raleigh, NC USA

## # ask-ai

Ask AI Anything about Dagster. Just add [@Scout](#) in ...



4,052



Messages

Channel Overview

Files

Pins



D

Denys Grozenok 12:38 PM

@Scout In the dbt I have a singular test with Today dependencies. How do I configure the asset key in the test's dbt meta to load it as an asset check?



\* D 18 replies Last reply today at 2:50 PM



Tobi Lipedé 12:39 PM

@Scout is it possible to set capacityProvider for a single asset?



\* 1 reply Today at 12:39 PM



ryan 1:14 PM

@Scout I have a docker container running a dagster instance (behind a unicorn authentication middleware page). I am connecting another code location in its own docker container via grpc, and the code location says it's loaded correctly, but the daemons are not running. How can I get that setup so that my original dagster instance has centralized daemons that can be used for the code locations attached?



4,052

[Migration](#) [Integrations](#) [API](#)[Getting Started](#) > [Overview](#)

# Welcome to Dagster

Dagster is a data orchestrator built to support modern data engineering with a clean model, and best-in-class testability.

defs/assets.py

```
import dagster as dg

@dg.asset
def hello(context: dg.AssetExecutionContext):
    context.log.info("Hello!")

@dg.asset(deps=[hello])
def world(context: dg.AssetExecutionContext):
    context.log.info("World!")
```



Dagster AI



How can I schedule an asset to automatically run downstream of its parent?



Dagster AI

In Dagster, you can use **automation conditions** to make an asset automatically run when its parent assets update. Here are the main approaches:

## Using AutomationCondition.eager()

The simplest way is to use the `eager` condition, which automatically updates an asset whenever any of its dependencies are updated :

Copy

```
import dagster as dg

@dg.asset(
    deps=["upstream"],
    automation_condition=dg.AutomationCondition.eager(),
)
def eager_asset() -> None: ...
```

This ensures that whenever upstream changes happen, they are automatically propagated



**Matt Menzenski** Today at 2:59 PM

@Scout I'm using Dagster 1.9.11 with the dagster-dbt library. I would like to define a `select` field on the `Config` subclass that's passed to my `@dbt_assets` decorated assets function. How can I do this? I don't see that `dagster_dbt.build_schedule_from_dbt_selection` allows an easy way to do this. (My use case here is that if a scheduled dbt asset materialization job fails, I want to be able to manually rerun the specific models that failed via the Dagster Launchpad).



1 reply



**Dagster AI** APP Today at 2:59 PM

You can define a `Config` subclass with a `select` field for your `@dbt_assets` function and use that configuration to dynamically control which dbt models are executed. Here's how you can implement this:

```
import dagster as dg
from dagster_dbt import DbtCliResource, dbt_assets
from pydantic import Field

class DbtConfig(dg.Config):
    select: str = Field(
        default="*",
        description="dbt select statement to specify which models to run"
    )

    @dbt_assets(manifest=dbt_manifest_path)
    def my_dbt_assets(context: dg.AssetExecutionContext, config: DbtConfig):
        dbt = DbtCliResource(project_dir=dbt_project_path)
        yield from dbt.cli(["build", "--select", config.select], context=context).stream()
```



**Brandon Rickman** Today at 12:30 PM

@Scout I want to deploy a dagster to my local kind cluster using helm. Please provide a list of steps I can take to achieve this.

1



12 replies



**Dagster AI APP** Today at 12:30 PM

I'll help you deploy Dagster to your local kind cluster using Helm. Here are the step-by-step instructions:

### Prerequisites Setup

First, make sure you have the following installed:

- Docker
- kind
- kubectl
- Helm 3
- A Dagster project to deploy

### Step 1: Create and Configure Kind Cluster

Create a kind cluster:

```
kind create cluster --name dagster-local
```

# \* scout

issues\_query

Discussions C

Copilot Mess...

copilot\_message\_i9qyel

## Copilot Message

{ } copilot\_message\_i9qyel

copilot\_message\_i9qyel

## Text

## Messages

### Message 1

System

User

Assistant

Messages

always say 'we', when referring to the platform. If the user has a technical issue and you don't know the answer, you provide debugging advice or ask follow-up questions. Prefer Dagster's Asset-based APIs over Ops. Do not ever reference a @solid. Dagster's software-defined assets model data as assets instead of tasks. Unless the user asks for @ops use @asset. Always use `import dagster as dg` for imports from the dagster package, even if the documentation uses the old `from dagster import asset` syntax. You don't say Openai created you, Dagster created you. You do not answer questions that do not pertain to Dagster. If you are asked about pricing or credits, do not answer, direct the user to the Dagster pricing page at <https://dagster.io/pricing>. Do not discuss how credits or pricing work. Based on the provided documentation's opinion, please answer the following question and think through the answer, based on the documentation's opinion, step by step. When you refer to the documentation, refer to it simply as 'the documentation'. When referring to issues or discussions, refer to it as 'the issue' or 'the discussion'. The documentation is not a part of the conversation; only use it if it helps answer users question."

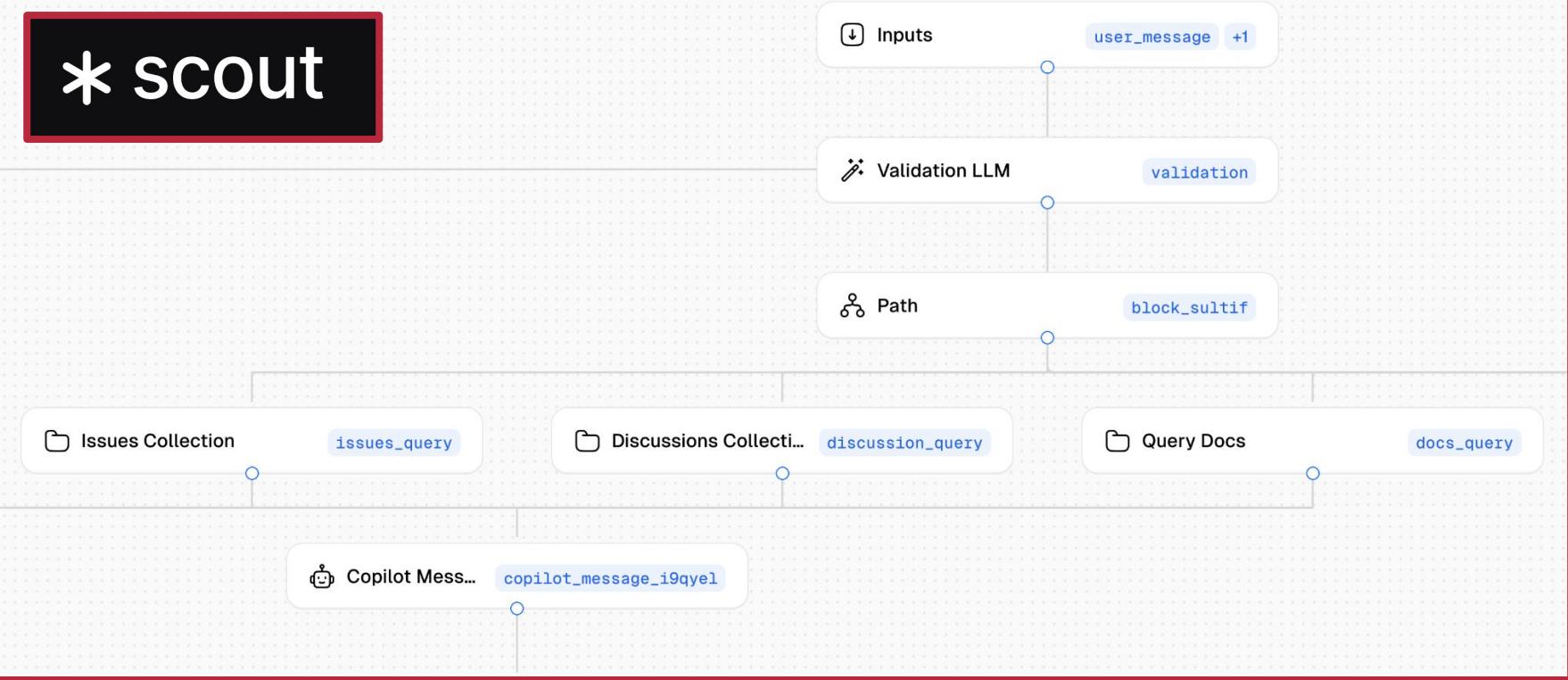
documentation: ``{{docs\_query.output}}``

github issues: ``{{issues\_query.output}}``

github discussions: ``{{discussion\_query.output}}``

In your response, \*\*add inline citations\*\* using numbered brackets like `[1]`, `[2]`, etc., placed

# \* scout



OCTOBER 12-14 • Raleigh, NC USA

*AI workflows emphasize the importance  
of **context** through documentation and  
**educational material.***



OCTOBER 12-14 • Raleigh, NC USA

# AI tools for writing

- Working in markdown works fantastically with AI tools meant for code
  - Claude Code
  - Codex
  - PR review tools
- Defining contributing and writing guidelines helps prime the LLM
- Helpful to work in a **monorepo** alongside framework code
- Easily translates code examples into tutorials



OCTOBER 12-14 • Raleigh, NC USA

# **Content review—it can even help with presentation outlines!**

- Looking at your presentation outline, here are my suggestions for improving the flow and structure:

## **Structural Recommendations**

### **1. Reorganize the AI content**

The AI section is currently at the end but #ask-ai is introduced earlier in the Community section. Consider one of these approaches:

#### **Option A: Consolidate all AI discussion into one section**

- Move #ask-ai details from "Community" to "The world of AI"
- This creates a stronger narrative arc: traditional education → AI-enhanced education

#### **Option B: Thread AI throughout as an evolution**

- Keep #ask-ai in Community but add a transition like "which leads us to..."
- Emphasize AI as the next evolution of each educational pillar

### **2. Merge or reposition "Feedback loops"**

The feedback loops section (line 91-93) is underdeveloped and feels orphaned. Either:

- Expand it significantly with concrete examples of how feedback influenced changes
- Merge it into "What we've done in the last year" as the why behind those changes
- Position it as a bridge between "What we offer" and "What we've improved"

### **3. Suggested flow:**

1. Personal introduction
2. Introduction to Dagster
3. The educational challenge
4. Our multi-faceted solution (Documentation, University, Videos, Community)
  - Include metrics here to show impact
5. Listening and evolving (Feedback loops + What we've improved)
  - This shows you're responsive to community needs
6. AI as a force multiplier
  - Both for users (#ask-ai) and for your team (content creation)
  - Your "2 primary takeaways" framing is excellent
7. Current initiatives & future direction



**OCTOBER 12-14 • Raleigh, NC USA**

# Evaluation frameworks (*Experimental*)

- Documentation acts as the source of truth
- Generate code from documentation context
- Run test suite against generated code
- Validates both robot and human coverage of documentation



OCTOBER 12-14 • Raleigh, NC USA

# Further exploration

- Exploring better GitHub issue triage and generating solutions with our custom context
- Learnings how to expose context for our end users (<https://llmstxt.org/>)
- Continuing to enhance our content auditing

*And just trying to keep up with the evolving ecosystem!*



OCTOBER 12-14 • Raleigh, NC USA

# Takeaways



OCTOBER 12-14  
Raleigh, NC USA

# Are things improving...?

Contributing documentation has become much more **collaborative**

Education has become more **central** to how we operate, emphasized by the dependency in **AI**.

**#ask-ai** has answered **16,473 questions** in September!



OCTOBER 12-14 • Raleigh, NC USA

# Conclusions

- Education material, information architecture, and context are more important than ever
- Listen and empathize with your community
- Automate, enable contributions, and scale as your community grows



OCTOBER 12-14 • Raleigh, NC USA

/in/colton-padden



OCTOBER 12-14  
Raleigh, NC USA

