

































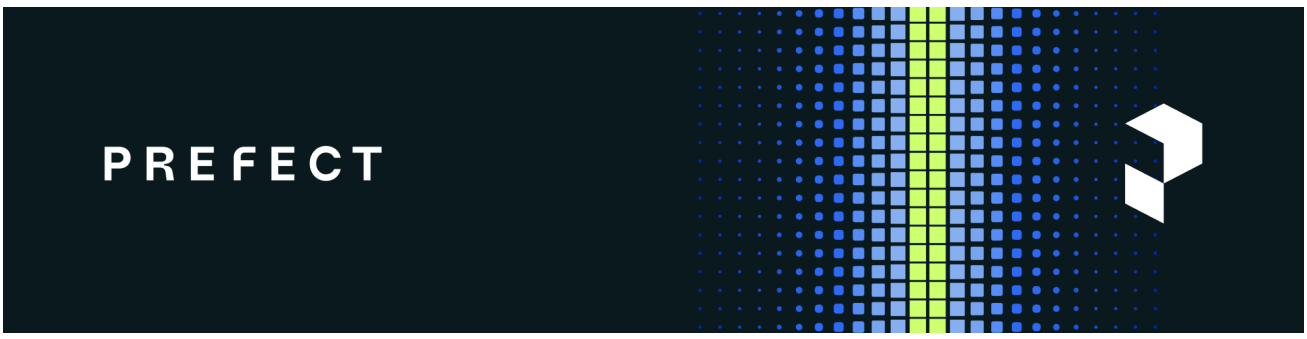
 github-actions[bot] docs: document integration settings pattern (PrefectB... ... ✓
02847b6 · 3 hours ago 
 .claude docs: raise quality bar for auto... 2 days ago
 .github docs: raise quality bar for auto... 2 days ago
 benches perf: replace correlated EXIST... last week
 client chore(deps): bump the python... 3 days ago
 compat-tests @ 3c5ec01 Add additional websocket deb... last year
 docs docs: Add release notes for 3... 2 days ago
 examples docs(prefect-dbt): reorganize p... last week
 integration-tests docs: add AGENTS.md for inte... 2 weeks ago
 load_testing Fix script to run server with loc... last year
 plans Add opt-in BuildKit/buildx supp... 2 months ago
 schemas docs: Add release notes for 3... last month
 scripts chore(deps): bump uv from 0.9... 3 weeks ago
 src docs: document integration set... 3 hours ago
 tests Fix #21768 : Use now() helper i... 2 days ago
 tools Add SDK instrumentation mod... 3 months ago
 ui-v2 chore(deps): bump the ui-v2-d... 3 days ago
 ui chore(deps): bump postcss fro... last week
 .dockerignore omit nested venv from docker ... 11 months ago
 .gitignore fix(prefect-dbt): silence dbt con... 2 weeks ago

 .gitmodules	Run compat - tests as part of ...	2 years ago
 .nvmrc	chore(deps-dev): bump the esl...	3 months ago
 .pre-commit-config.yaml	chore(deps): bump the python...	3 days ago
 .prefectignore	Removes experimental status ...	3 years ago
 AGENTS.md	Document uv run --project for l...	2 weeks ago
 CODE_OF_CONDUCT....	copy CODE_OF_CONDUCT.md to r...	2 years ago
 Dockerfile	chore(deps): bump continuumi...	4 days ago
 Dockerfile.sqlite-builder	perf(docker): replace inline SQ...	3 months ago
 LICENSE	Update license to apache	4 years ago
 README.md	add bsky link to readme (#204...	4 months ago
 SECURITY.md	Updates the contact info for ou...	2 years ago
 justfile	Add just unreleased-integr...	2 months ago
 pyproject.toml	chore(deps): bump the python...	3 days ago
 uv.lock	chore(deps): bump the uv-dep...	2 days ago

[README](#)
[Code of conduct](#)
[Contributing](#)
[Apache-2.0 license](#)
[Security](#)



pypi **v3.6.29** | downloads **12M/month** | Stars **22k** | commit activity **221/month**
[slack](#) | [join community](#) | [youtube](#) | [watch videos](#)

[Installation](#) · [Quickstart](#) · [Build workflows](#) · [Deploy workflows](#) · [Prefect Cloud](#)

Prefect

Prefect is a workflow orchestration framework for building data pipelines in Python. It's the simplest way to elevate a script into a production workflow. With Prefect, you can build resilient, dynamic data pipelines that react to the world around them and recover from unexpected changes.

With just a few lines of code, data teams can confidently automate any data process with features such as scheduling, caching, retries, and event-based automations.

Workflow activity is tracked and can be monitored with a self-hosted [Prefect server](#) instance or managed [Prefect Cloud](#) dashboard.

💡 Tip

Prefect flows can handle retries, dependencies, and even complex branching logic

[Check our docs](#) or see the example below to learn more!

Getting started

Prefect requires Python 3.10+. To [install the latest version of Prefect](#), run one of the following commands:

```
pip install -U prefect
```



```
uv add prefect
```



Then create and run a Python file that uses Prefect `flow` and `task` decorators to orchestrate and observe your workflow - in this case, a simple script that fetches the number of GitHub stars from a repository:

```
from prefect import flow, task
import httpx

@task(log_prints=True)
def get_stars(repo: str):
    url = f"https://api.github.com/repos/{repo}"
    count = httpx.get(url).json()["stargazers_count"]
    print(f"{repo} has {count} stars!")

@flow(name="GitHub Stars")
def github_stars(repos: list[str]):
    for repo in repos:
        get_stars(repo)
```



```
# run the flow!  
if __name__ == "__main__":  
    github_stars(["PrefectHQ/prefect"])
```

Fire up a Prefect server and open the UI at <http://localhost:4200> to see what happened:

```
prefect server start
```



To run your workflow on a schedule, turn it into a deployment and schedule it to run every minute by changing the last line of your script to the following:

```
if __name__ == "__main__":  
    github_stars.serve(  
        name="first-deployment",  
        cron="* * * * *",  
        parameters={"repos": ["PrefectHQ/prefect"]}  
    )
```



You now have a process running locally that is looking for scheduled deployments! Additionally you can run your workflow manually from the UI or CLI. You can even run deployments in response to [events](#).

Tip

Where to go next - check out our [documentation](#) to learn more about:

- [Deploying flows to production environments](#)
- [Adding error handling and retries](#)
- [Integrating with your existing tools](#)
- [Setting up team collaboration features](#)

Prefect Cloud

Prefect Cloud provides workflow orchestration for the modern data enterprise. By automating over 200 million data tasks monthly, Prefect empowers diverse organizations — from Fortune 50 leaders such as Progressive Insurance to innovative disruptors such as Cash App — to increase engineering productivity, reduce pipeline errors, and cut data workflow compute costs.

Read more about Prefect Cloud [here](#) or sign up to [try it for yourself](#).






prefect-client

If your use case is geared towards communicating with Prefect Cloud or a remote Prefect server, check out our [prefect-client](#). It is a lighter-weight option for accessing client-side functionality in the Prefect SDK and is ideal for use in ephemeral execution environments.





Connect & Contribute

Join a thriving community of over 25,000 practitioners who solve data challenges with Prefect. Prefect's community is built on collaboration, technical innovation, and continuous improvement.

Community Resources

-  [Explore the Documentation](#) - Comprehensive guides and API references
-  [Join the Slack Community](#) - Connect with thousands of practitioners
-  [Contribute to Prefect](#) - Help shape the future of the project
-  [Support or create a new Prefect integration](#) - Extend Prefect's capabilities
-  [Tail the Dev Log](#) - Prefect's open source development blog

Stay Informed

-  [Subscribe to our Newsletter](#) - Get the latest Prefect news and updates
-  [X](#) and [Bluesky](#) - Latest updates and announcements
-  [YouTube](#) - Video tutorials and webinars
-  [LinkedIn](#) - Professional networking and company news

Your contributions, questions, and ideas make Prefect better every day. Whether you're reporting bugs, suggesting features, or improving documentation, your input is invaluable to the Prefect

Releases 805

 **3.6.29 - ON CONFLICT DO BETTER** Latest
2 days ago

[+ 804 releases](#)

Packages

No packages published

Contributors 427



[+ 413 contributors](#)

Languages

