CI/CD for statistical software engineering

Pravin Madhavan

PSI2025 Session: Statistical Software Engineering

FACTS

BERRY

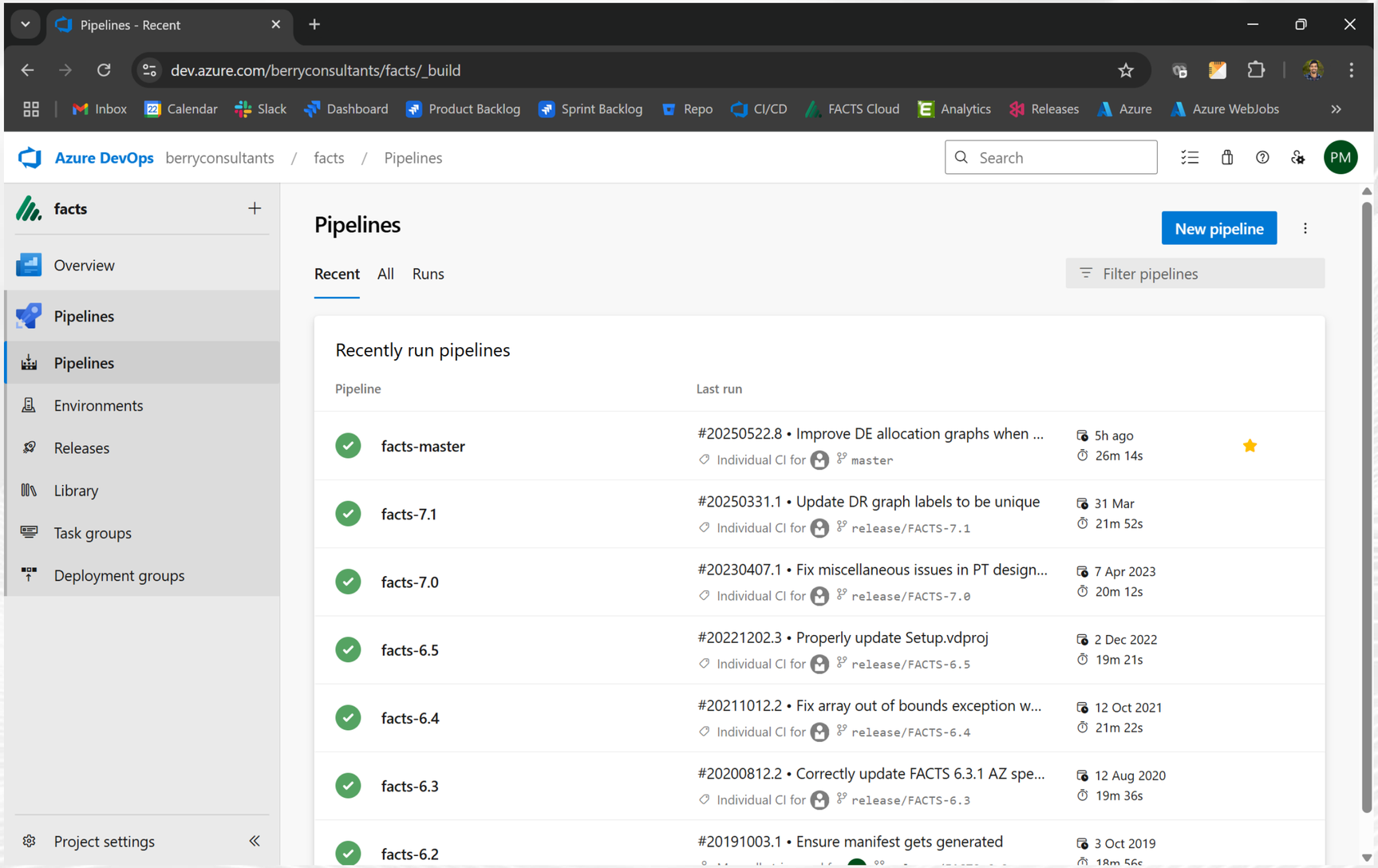
ADDPLAN

QUOTES

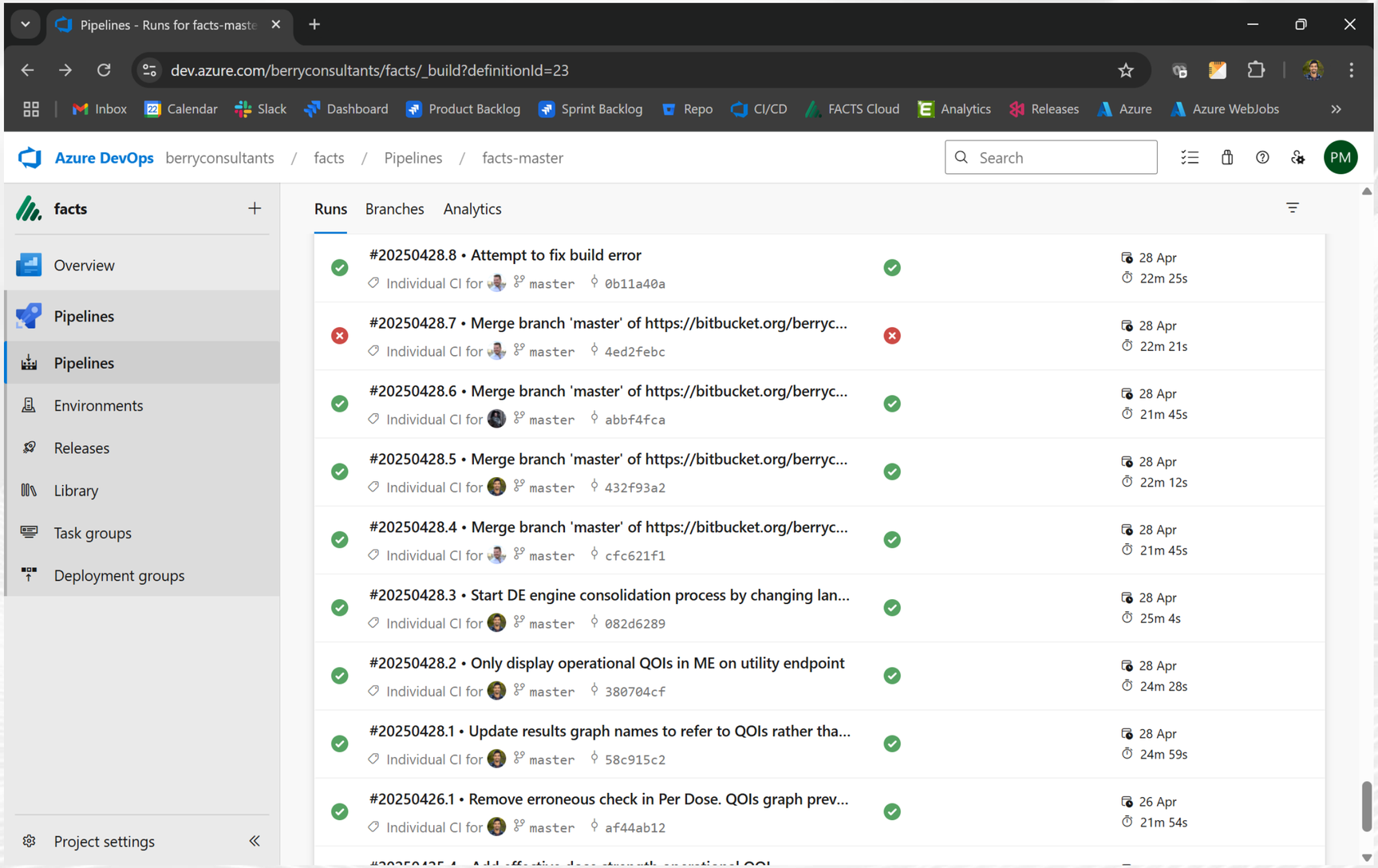
Continuous Integration (CI)

- **Frequent Code Integration**: Developers regularly merge code changes into a shared repository, often multiple times a day.
- **Automated Builds and Tests**: Each integration triggers automated builds and tests to quickly detect errors and integration issues.
- **Early Bug Detection**: CI helps catch bugs early in the development cycle, reducing the cost and effort to fix them.
- **Improved Collaboration**: CI encourages team collaboration by keeping everyone’s code up to date and compatible.
- **Faster Delivery**: By automating testing and integration, CI accelerates the development process and improves software quality.

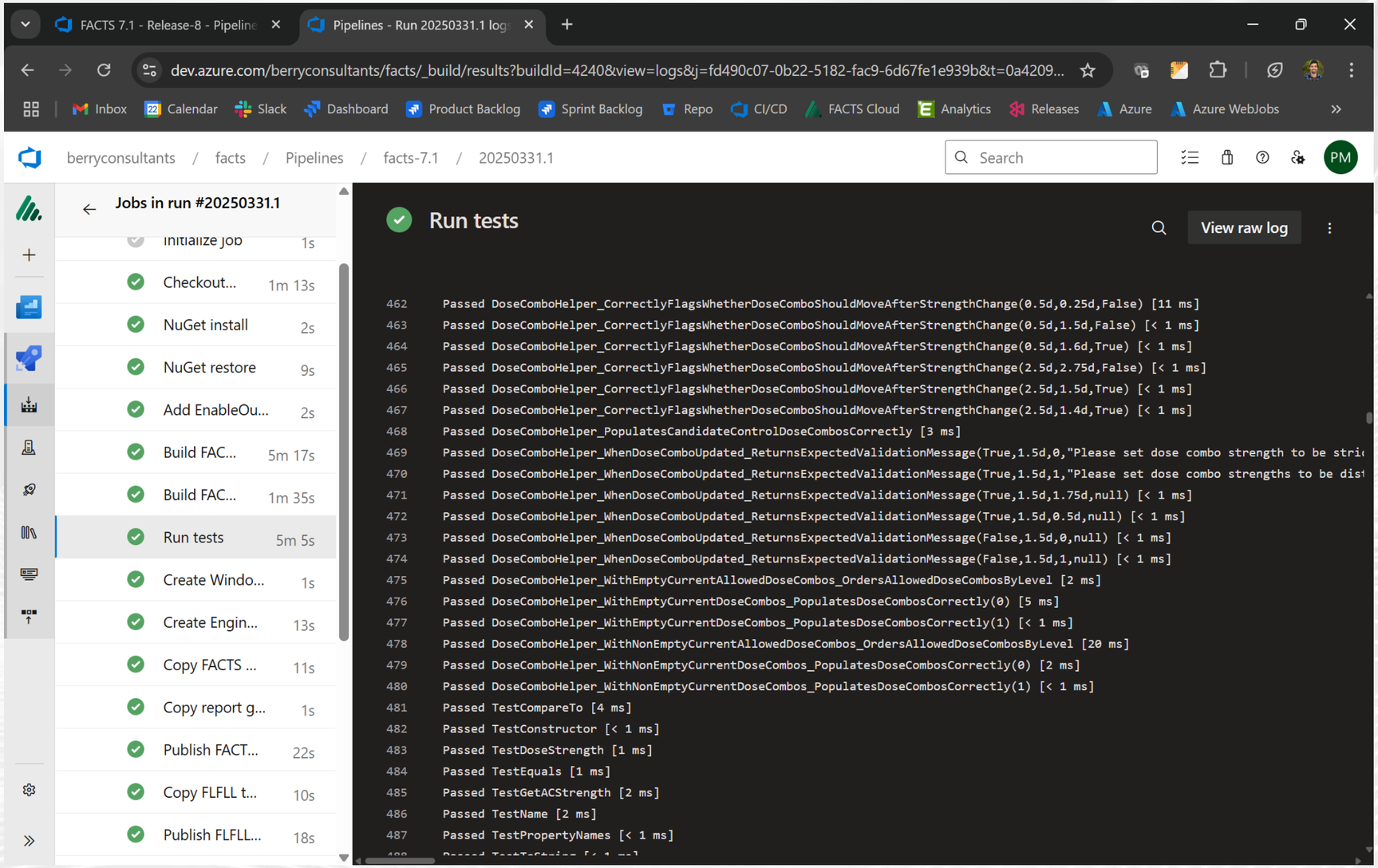
Continuous Integration (CI): Pipelines



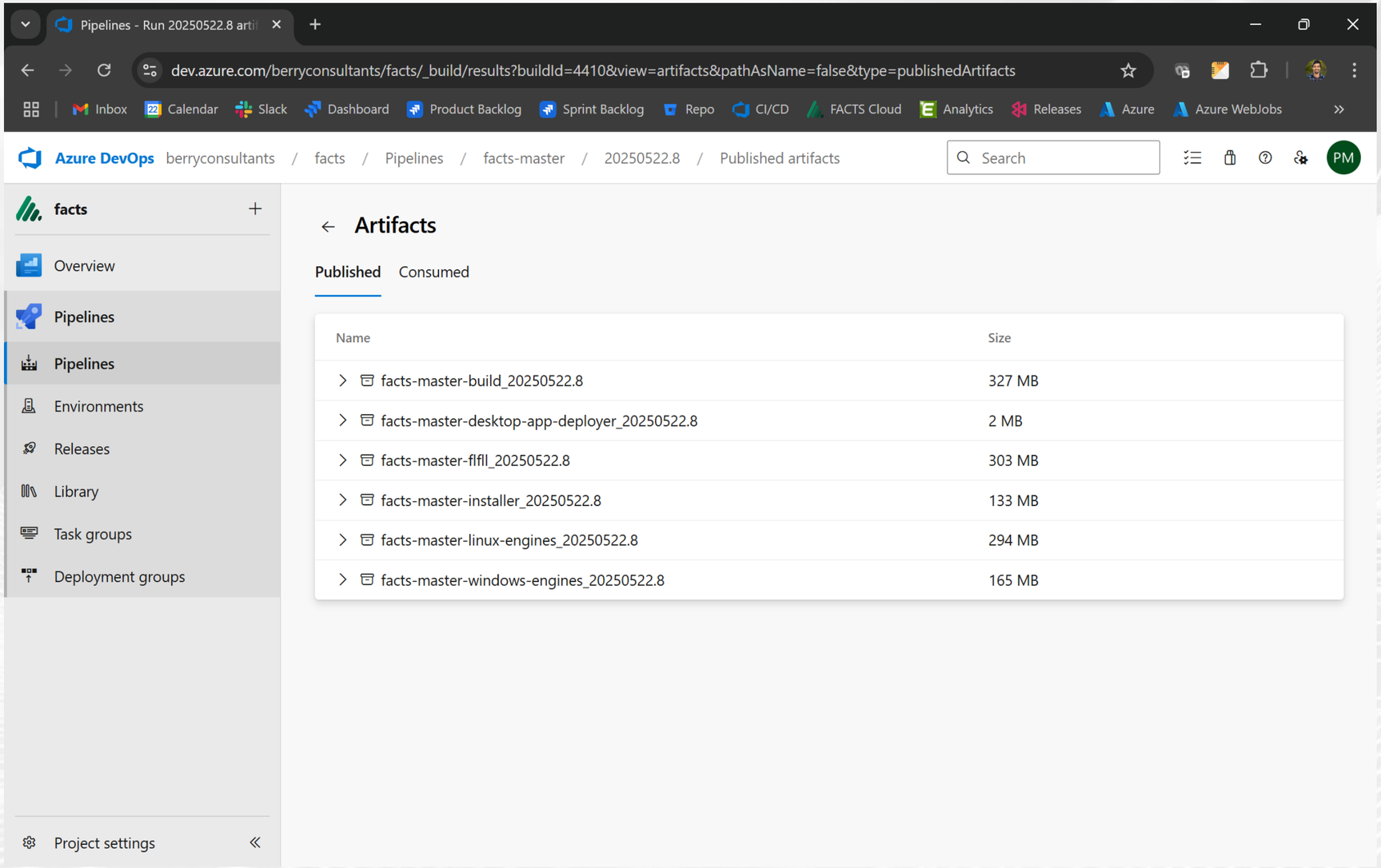
Continuous Integration (CI): Pipeline Builds



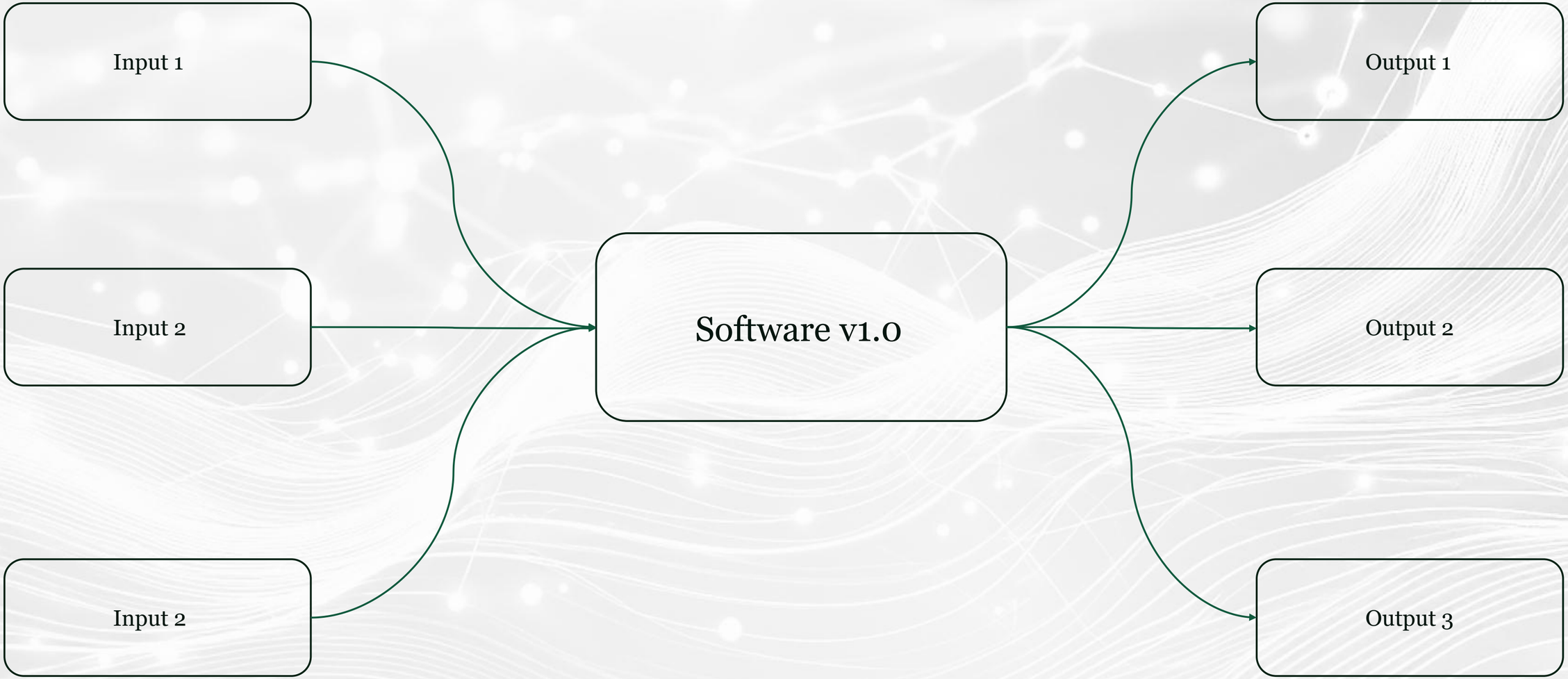
Continuous Integration (CI): Pipeline Build Steps



Continuous Integration (CI): Pipeline Build Artifacts

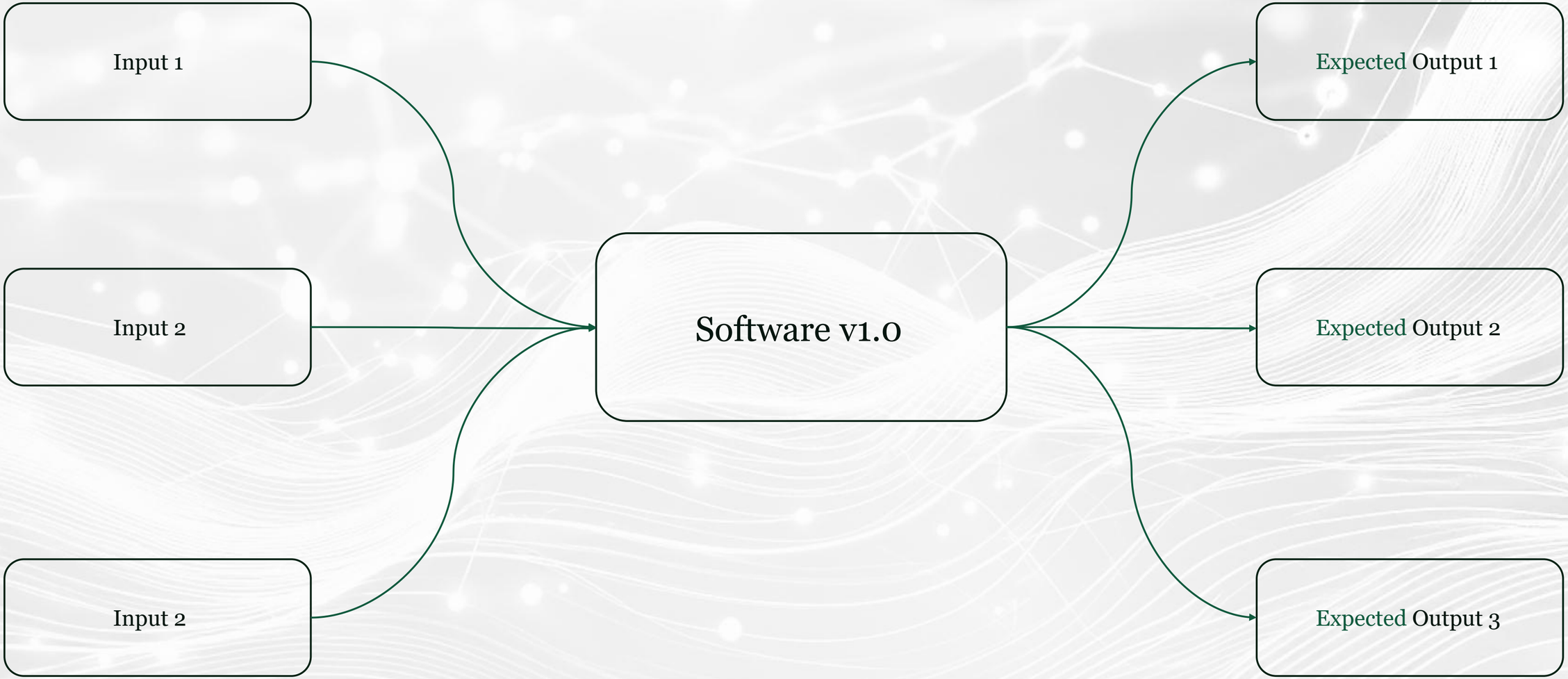


Continuous Integration (CI): End-to-End (E2E) testing

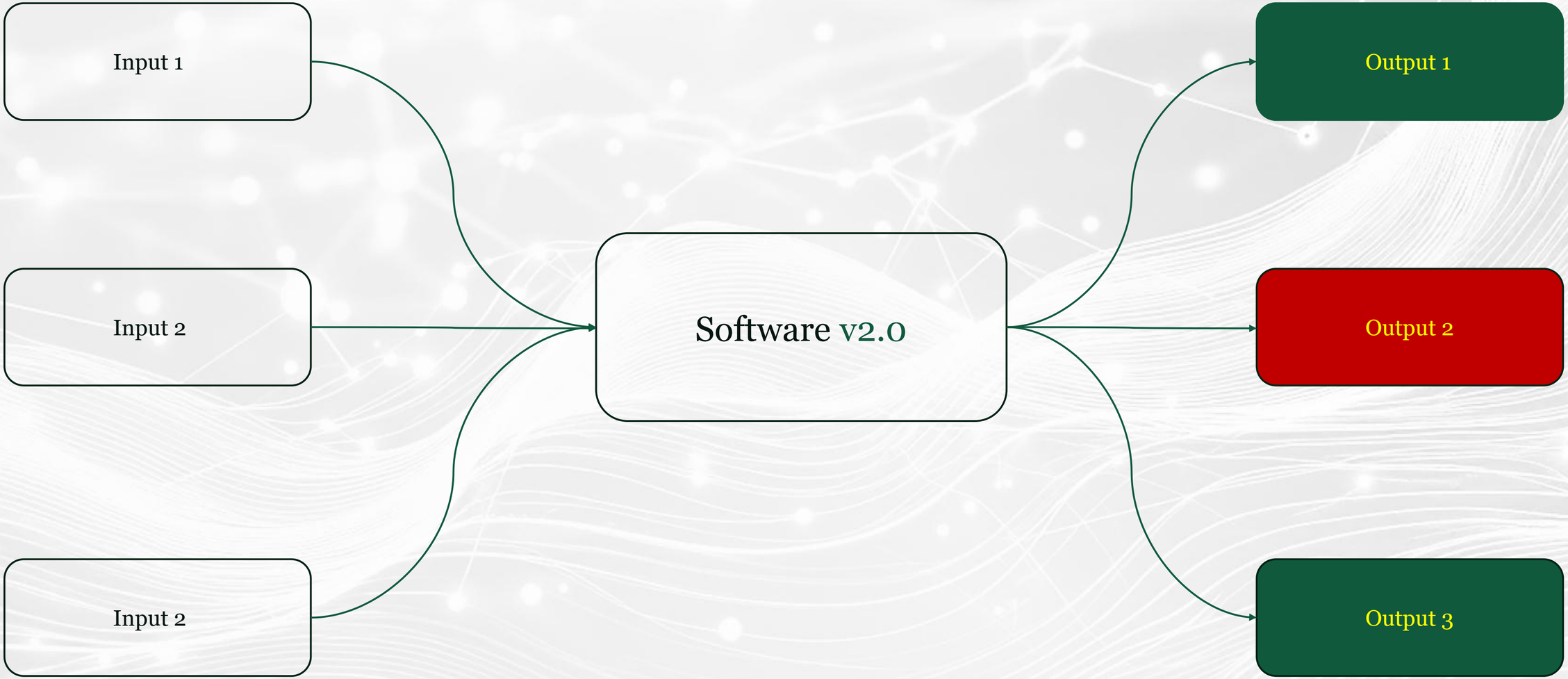


BERRY

Continuous Integration (CI): End-to-End (E2E) testing



Continuous Integration (CI): End-to-End (E2E) testing

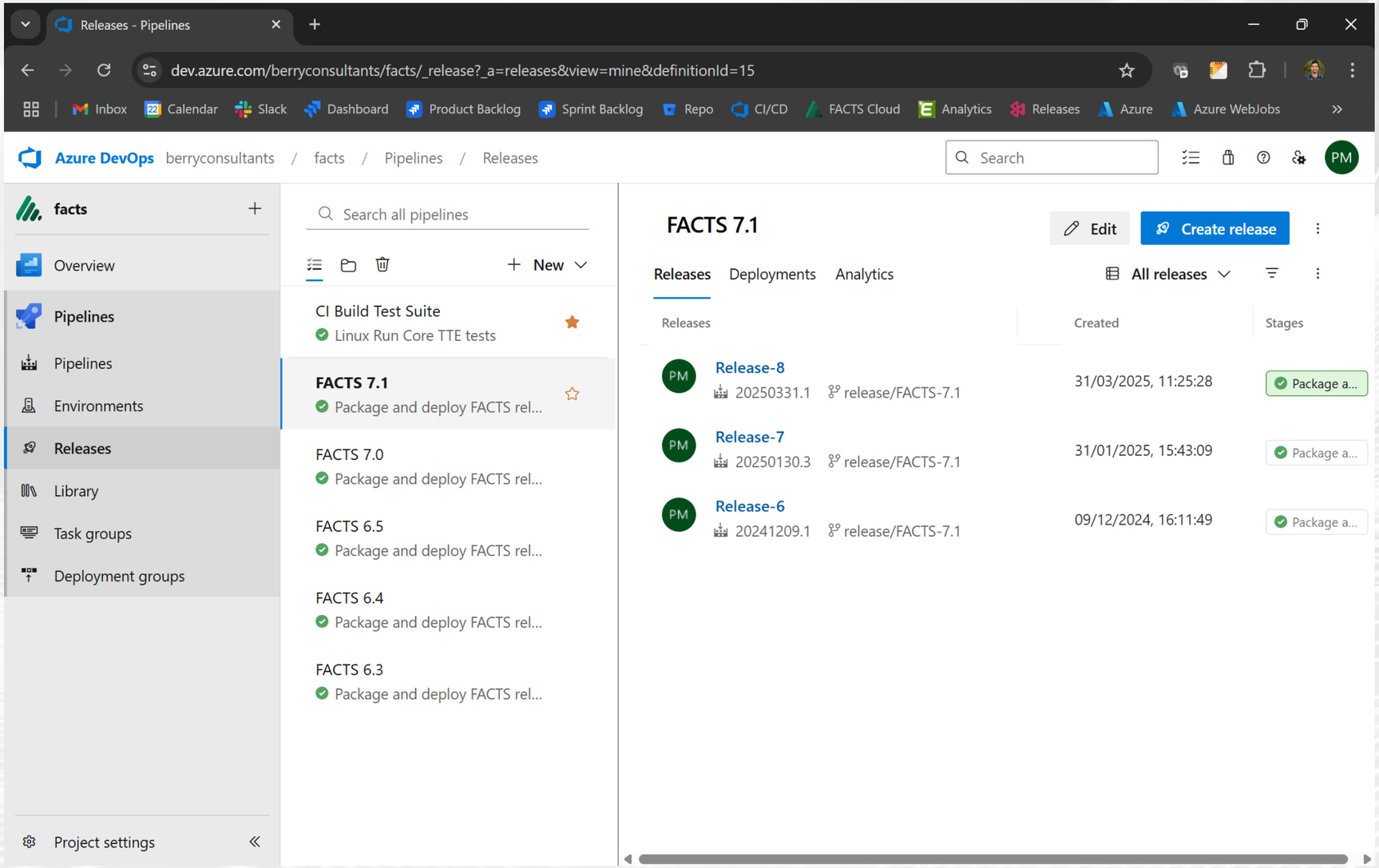


BERRY

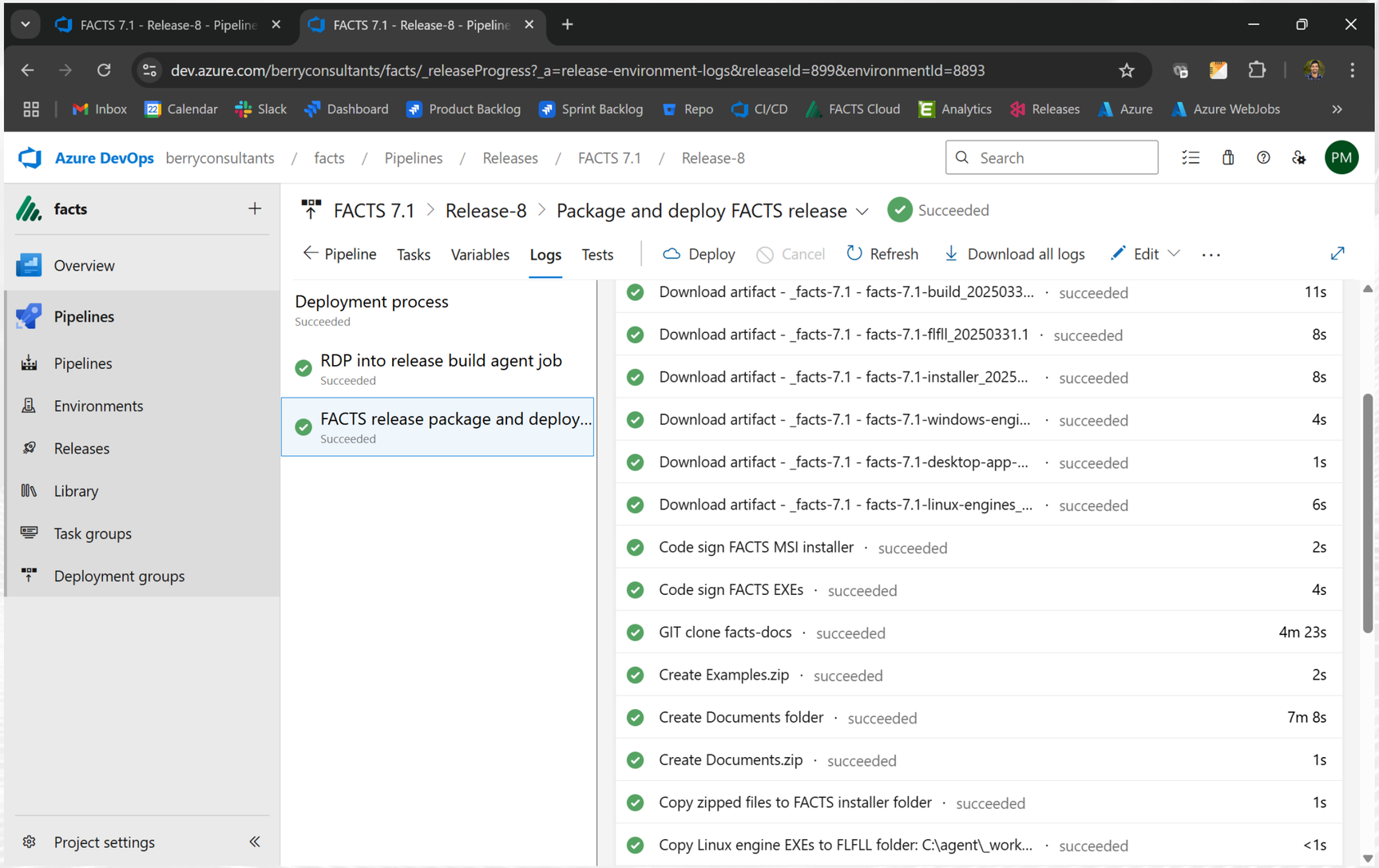
Continuous Delivery (CD)

- **Automated Release Process**: CD ensures that code changes are automatically prepared for a release to production after passing CI.
- **Production-Ready Builds**: Every code change that passes tests is considered deployable, ensuring a consistently shippable product.
- **Manual or Triggered Deployment**: Releases can be deployed to production at any time with minimal effort, often by a single click or command.
- **Faster Time to Market**: CD shortens the release cycle, enabling more frequent and reliable software updates.
- **Reduced Deployment Risk**: Frequent, smaller updates lower the chances of large-scale failures and make rollback easier if issues occur.

Continuous Delivery (CD): Release Pipelines



Continuous Delivery (CD): Release Pipeline Steps



Continuous Delivery: Release Cycles

- Issues we faced:
  - FACTS releases were previously relatively bulky, non-deterministic and rare.
  - Bulky releases take longer to validate, both internally and externally. We wanted a **quicker validation process**.
  - A non-deterministic release schedule makes it harder for IT teams to deploy FACTS across their organization. We wanted a **predictable release schedule**.
  - FACTS releases were relatively rare (once a year). We wanted to provide **more frequent value**.
- Our aim: 4 releases a year, every 3 months.

Website: www.berryconsultants.com

Knowledge Hub: docs.berryconsultants.com

Email: info@berryconsultants.com

LinkedIn: www.linkedin.com/company/berry-consultants-llc

YouTube: youtube.com/berryconsultants

X: @berryconsultant