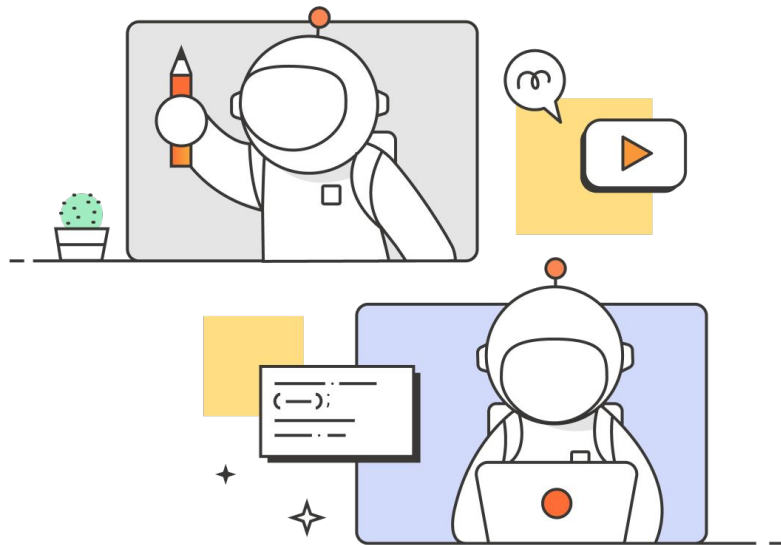




Continuous Integration/ Continuous Delivery (CI/CD) with Postman



We will be sharing the recording of this webinar as well as the slides. For previous recordings please visit: <https://www.postman.com/events/product-advocacy>

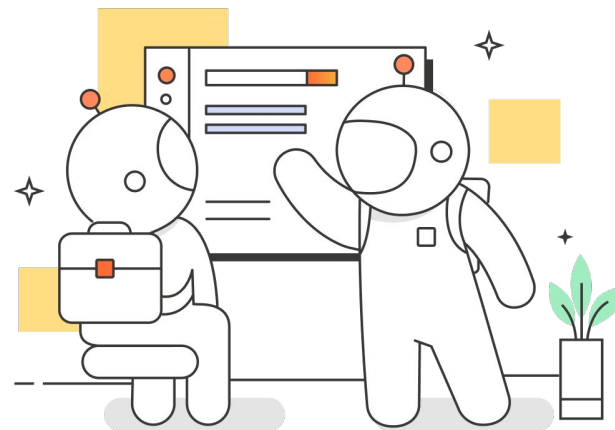


Agenda

-
- 1 Defining CI/CD

 - 2 Postman's command line tooling

 - 3 Using Postman tooling in your CI/CD setup





What is CI/CD?

Continuous Integration (CI)

- A flow is triggered that then builds, tests, and merges code

Continuous Delivery (CD)

- Code is built and ready to be deployed

Continuous Deployment (CD)

- Code is deployed to production servers



Build, Test, Package, Deploy



Postman's Command Line Tooling

CLI tool options

- Newman
- Postman CLI

Major differences

- Open source vs maintained by Postman
- Requires authentication for each command vs supporting login/logout
- Extended by reporters vs native reporting back to Postman
- API Linting options in the Postman CLI

Postman CLI	Newman
Created by Postman	Created by Postman
Maintained and supported by Postman	Open source; supported by community contributions
Supports collection runs	Supports collection runs
Automatically sends collection run results to Postman by default	Supports ingesting run results to Postman using a reporter
Package is signed by Postman	Package isn't signed by Postman
Distributed as a downloadable package	Distributed on npm
Downloadable programmatically	Downloadable programmatically
Not available as a library	Available as a library
Supports log in and log out	Doesn't support log in and log out
Checks API definition against configured API Governance and API Security rules	Doesn't check API definition against configured API Governance and API Security rules

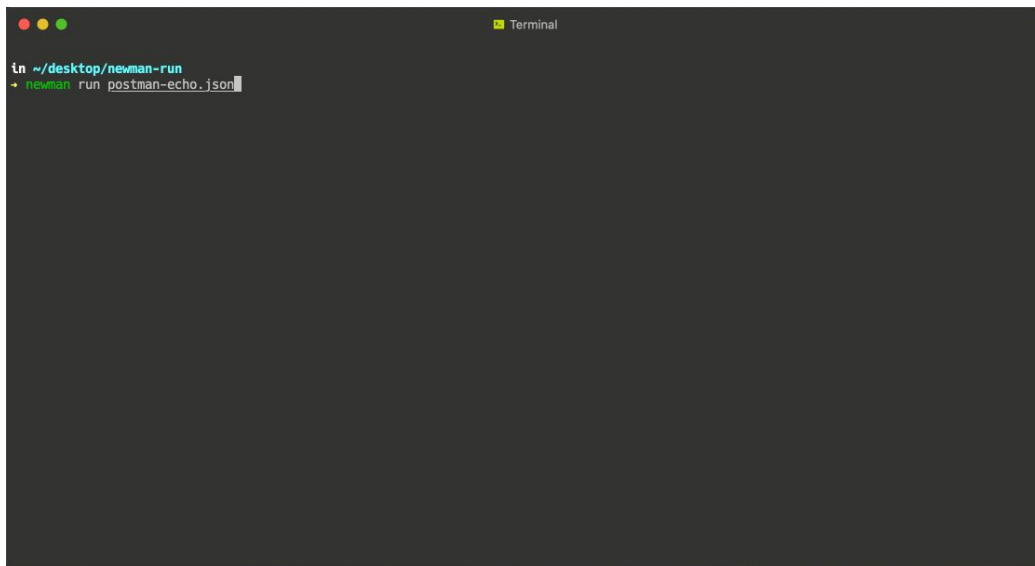
[Table Source](#)



Newman

Newman is an open source project from Postman

- Available on NPM
 - [Package located here](#)
- Extensible Collection Runner
 - Show results in console or multiple “reporter” formats
- Run Newman locally, or integrate with your CI/CD Pipeline

A screenshot of a terminal window with a dark background. The title bar at the top right says "Terminal". The prompt is "in ~/desktop/newman-run". The command entered is "newman run postman-echo.json".

```
in ~/desktop/newman-run
+ newman run postman-echo.json
```



Newman & CI/CD

Integrate your Newman Collection runner with your CI/CD Pipeline:

1. Export a collection (JSON file)
2. Store this JSON in your repository
3. Confirm file location & reference the file location in your CI tool
4. Install Node and Newman
5. Add the Newman shell command to your CI configuration file

Example CI integrations ([Travis CI](#) & [Jenkins](#))

GitLab CI example - YAML file

```
1  stages:
2  |   - test
3
4  postman_tests:
5  |   stage: test
6  |   image:
7  |     name: postman/newman
8  |     entrypoint: [""]
9  |   script:
10 |     - |
11 |       newman run postman/collections/myCollection.json
12 |       -e postman/environments/myEnvironment.json
```



Postman CLI

Postman CLI is a closed-source project, signed by Postman

- Available for [download](#) from Postman
- Secured and supported by Postman
- Collection runs and API linting
- Run data reported back to Postman UI
- Simplified command syntax
- Easily integrate Postman CLI with your CI/CD Pipeline

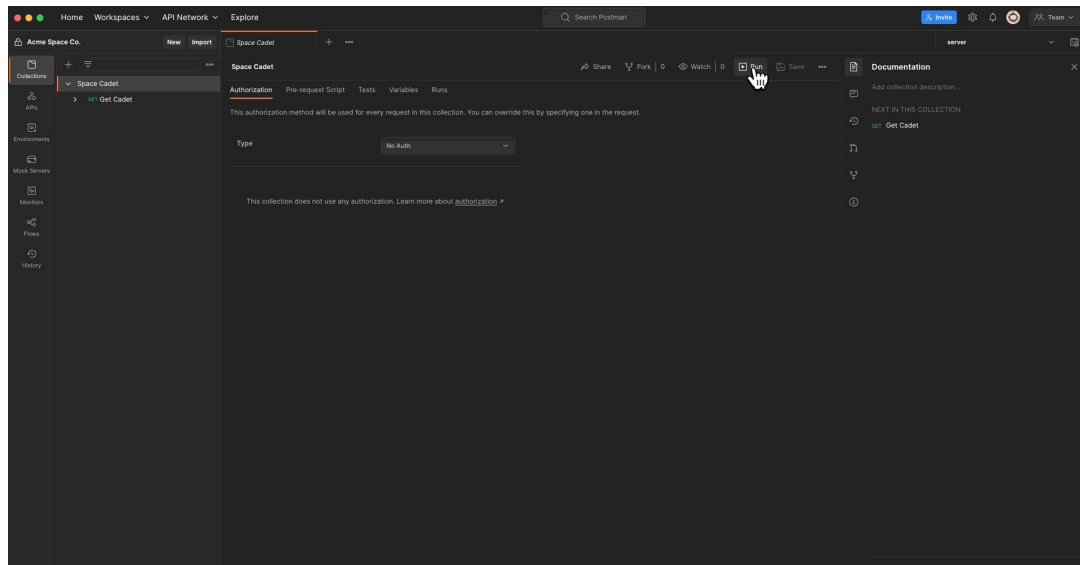
```
postman collection run 24367399-151f70df-1798-423e-ad59-867bbe97d6f9 -e 24367399-a7af28e3-d688-4d63-938b-6bce6648184f
```



Postman CLI & CI/CD

Integrate Postman CLI with your CI/CD Pipeline!

1. Go to your Collection > Click Actions > Run collection
2. Click **Automate runs via CLI** > then **Configure command** (under **Run on CI/CD**)
3. Choose your collection(s) and CI/CD pipeline options





Demo





CI/CD: Integrations with Postman

[Bitbucket Pipelines](#)

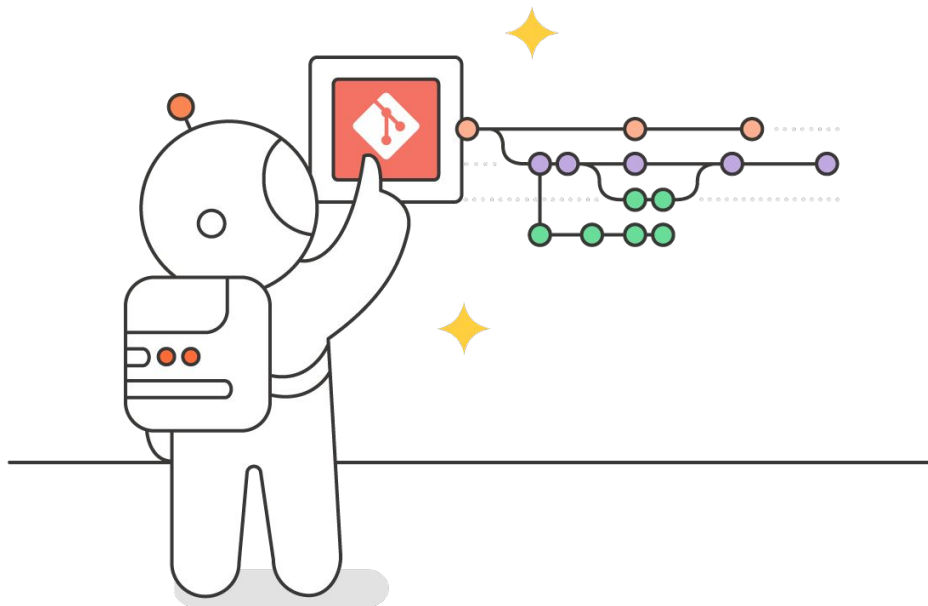
[CircleCI](#)

[GitHub Actions](#)

[GitLab CI/CD](#)

[Jenkins](#)

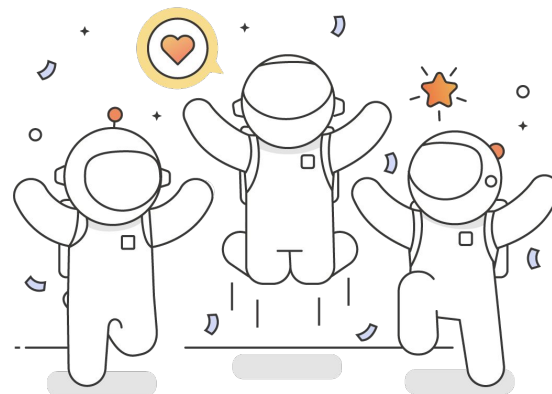
[Travis CI](#)





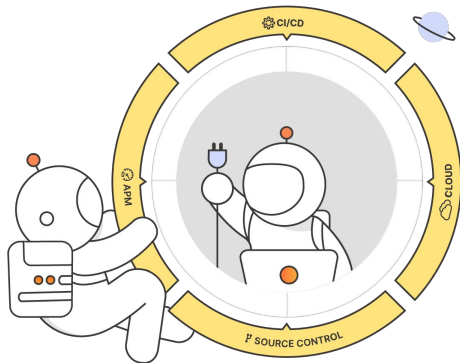
CI/CD: Overall benefits

- Up front work, but saves time in the long run
- More automation; more time spent building
- Software that's more frequently built, tested, and delivered
- Consistent and reliable programs will keep users/customers delighted





CI/CD: Continuous mindset meets Postman



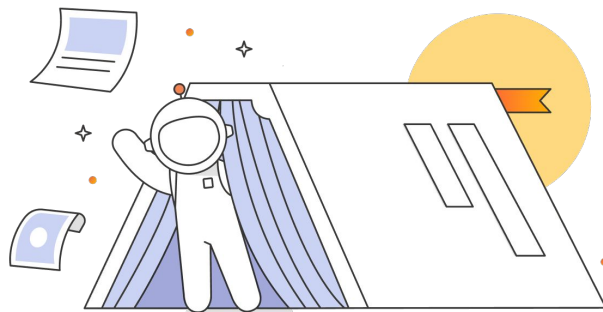
- Centralize API Design and Testing
- Integrations with major CI/CD providers
- Source control integrations provides visibility into any changes
- Reporting and APM integrations automate results
- DevOps teams can focus on deployment pipelines for the best customer experience

Postman Blog: [DevOps, Source Control, CI/CD, and API Automation](#)



CI/CD: Additional Resources

- [Integrating with Postman](#)
- [Postman CLI: Running a collection in CI/CD](#)
- [CI Integrations with Postman](#)
- [Using CI with Newman and the Postman API](#)
- [Comparing the Postman CLI and Newman](#)





Thank for participating in our Webinar!

If you have any follow-up questions, please book some time with the Product Advocacy team on our [Calendly](#) or email us at productadvocacy@postman.com.

