EP57: How ChatGPT works technically

ALEX XU
APR 29, 2023

♡ 368   💬 2   ↻ 8   Share   ⋯

This week's system design refresher:

- How ChatGPT works technically (Youtube video)

- 18 Key Design Patterns Every Developer Should Know

- Netflix Tech Stack - Part 1 (CI/CD Pipeline)

- How does Docker work?

- Job openings

[Postman's 2023 State of the API survey is open (Sponsored)](#)

The biggest, most comprehensive survey on APIs is open for a fourth year, with new questions on monetization, generative AI, and more. Take the survey and share how you work with APIs, the challenges you're overcoming, and what you see for the future of APIs (plus there are prizes, including a PS5, Steam Deck, and Amazon gift cards).

How ChatGPT works technically

We attempted to explain how it works in this video. We will cover:

- Large Language Model

- GPT-3.5

- Fine-tuning

- Prompt engineering

- How to answer a prompt

18 Key Design Patterns Every Developer Should Know

Patterns are reusable solutions to common design problems, resulting in a smoother, more efficient development process. They serve as blueprints for building better software structures. These are some of the most popular patterns:

- Abstract Factory: Family Creator - Makes groups of related items.

- Builder: Lego Master - Builds objects step by step, keeping creation and appearance

- Prototype: Clone Maker - Creates copies of fully prepared examples.

- Singleton: One and Only - A special class with just one instance.

- Adapter: Universal Plug - Connects things with different interfaces.

- Bridge: Function Connector - Links how an object works to what it does.

- Composite: Tree Builder - Forms tree-like structures of simple and complex parts.

- Decorator: Customizer - Adds features to objects without changing their core.

- Facade: One-Stop-Shop - Represents a whole system with a single, simplified interface.

- Flyweight: Space Saver - Shares small, reusable items efficiently.

- Proxy: Stand-In Actor - Represents another object, controlling access or actions.

- Chain of Responsibility: Request Relay - Passes a request through a chain of objects until handled.

- Command: Task Wrapper - Turns a request into an object, ready for action.

- Iterator: Collection Explorer - Accesses elements in a collection one by one.

- Mediator: Communication Hub - Simplifies interactions between different classes.

- Memento: Time Capsule - Captures and restores an object's state.

- Observer: News Broadcaster - Notifies classes about changes in other objects.

- Visitor: Skillful Guest - Adds new operations to a class without altering it.

Netflix Tech Stack - Part 1 (CI/CD Pipeline)
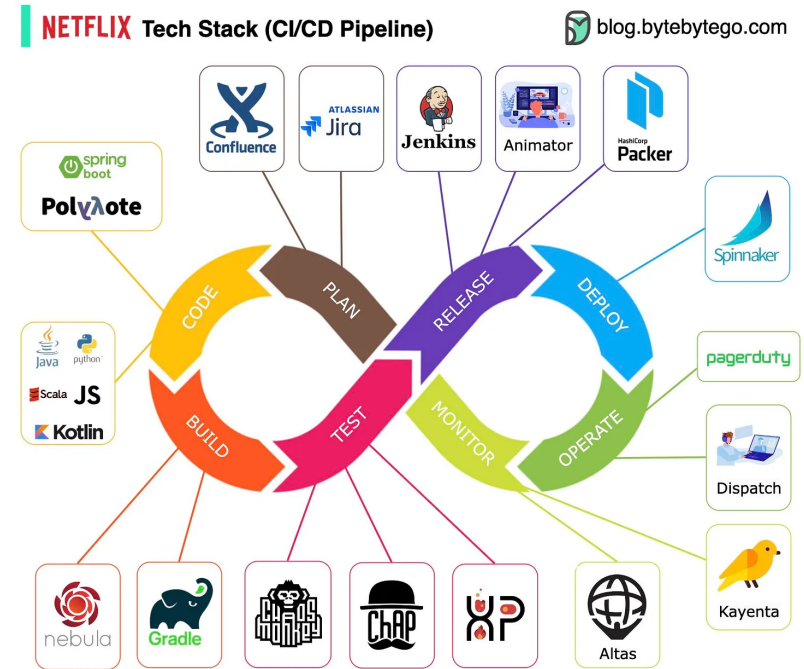
- Planing: Netflix Engineering uses JIRA for planning and Confluence for documentation.

- Coding: Java is the primary programming language for the backend service, while other languages are used for different use cases.

- Build: Gradle is mainly used for building, and Gradle plugins are built to support various use cases.

- Packaging: Package and dependencies are packed into an Amazon Machine Image (AMI) for release.

- Testing: Testing emphasizes the production culture's focus on building chaos tools.

- Deployment: Netflix uses its self-built Spinnaker for canary rollout deployment.

- Monitoring: The monitoring metrics are centralized in Atlas, and Kayenta is used to detect anomalies.

- Incident report: Incidents are dispatched according to priority, and PagerDuty is used for incident handling.

Thanks, [Xiong Wang](), for putting it together!

—

We plan to develop detailed illustrations of tech stacks used by various companies for educational purposes. If you would like to contribute to this project, kindly complete the short survey. All responses will be kept anonymous. Thank you for your help.
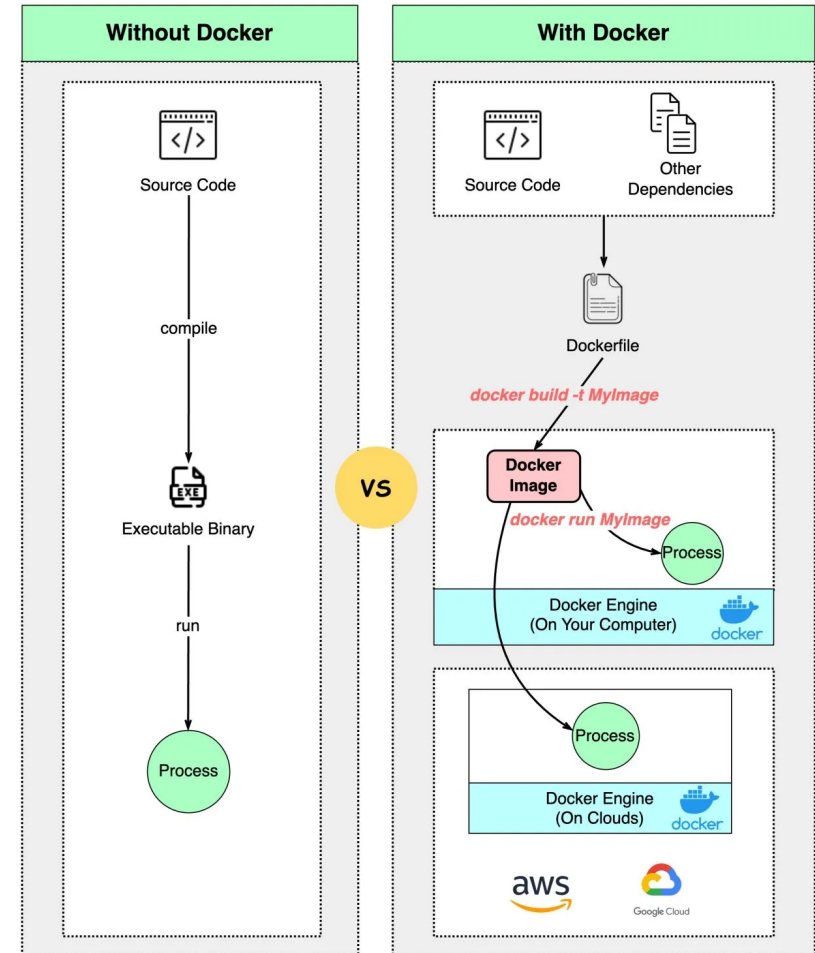
Share the tech stack you know here: [https://lnkd.in/eiEbH7tX]()

How does Docker work?

A comparison of Docker-based and non-Docker-based development is shown below.

With Docker, we can develop, package, and run applications quickly:

- The developers can write code locally and then build a Docker image and push it to a dev environment. In this way, the development work is shared within the team. The tests are run automatically in the dev environment.

An image is a read-only template with instructions for creating a Docker container. We use a Dockerfile to define the steps to create the image and run it.

◆ The above process can run incrementally when bugs are found or improvements are needed.

◆ When dev testing is complete, the Docker image is pushed to the production environment (often on the cloud).

Compared with traditional development without Docker, Docker is quite lightweight and fast, because only the changed part of Dockerfile is rebuilt every time we make a change.

Over to you: Where are Docker images stored?

Join the ByteByteGo Talent Collective

If you're looking for a new gig, join the collective for customized job offerings from selected companies. Public or anonymous options are available. Leave anytime.

If you're **hiring**, join the ByteByteGo Talent Collective to start getting bi-monthly drops of world-class hand-curated engineers who are open to new opportunities.

Featured job openings

X1 Card: Engineering Leader - Card Platform (United States, remote)

X1 Card: Software Engineer, Infrastructure (United States, remote)

(**catch**) **Health:** Senior Frontend Engineer, React + Typescript (United States, remote)
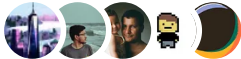
368 Likes · 8 Restacks

2 Comments

Write a comment...

John Crickett Coding Challenges Apr 30, 2023

I'm not sure I follow the reasoning for docker making development faster wrt to testing locally, but it's great for avoiding the 'works on my machine' issue.

♡ LIKE (4)   💬 REPLY   ⬆ SHARE

Ravi May 2, 2023

Not able to upgrade to paid. I am having HDFC diners club card. Every time after entering OTP it says unable to authenticate. Any idea ...?

♡ LIKE   💬 REPLY   ⬆ SHARE