DevOps Tools

Continuous Integration

Jenkins

- It is an open-source tool with great community support.
- It is easy to install.
- It has 1000+ plugins to ease your work. If a plugin does not exist, you can code it and share it with the community.
- It is free of cost.
- It is built with Java and hence, it is portable to all the major platforms.

Jenkins integrates development life-cycle processes of all kinds, including build, document, test, package, stage, deploy, static analysis, and much more.

"Jenkins is interconnected with well over 1,000 plugins that allow it to integrate with most of the development, testing and deployment tools." *Edureka*

Created in 2004, really popular amongst the DevOPs community, with the market share being 71% in 2019 2019 blogpost. 1+ million users.

Actively ongoing project **Last Commit 2 days ago.** With the first commit to the project in 2004
List of popular companies utilizing it for their apps include but not limited to:

- Facebook
- Netflix
- LinkedIn
- Twitch
- Robinhood
- Udemy
- Instacart
- etc...

Cited Source.

Real Time Error Monitoring

Raygun

- See the exact line of code that caused an error
- Monitor deployments to determine what caused a spike or decrease in error count
- Full stack trace information and diagnostic details for every error occurrence or crash
- Filter through your errors by date, time, version, tag, host, OS, browser, custom tags, and more
- Reduce noise with configurable filters for machine name, version, IP address, hostname, and more
- Groups errors by affected users (this also helps reduce noise)
- 180-day data retention
- Support for all major languages and frameworks
- Easy setup using lightweight SDKs
- Works seamlessly alongside Real User Monitoring and APM for full visibility into your users digital experience.

Few Integrations

- GitHub
- Jira Software
- Slack
- Amazon SQS
- Asana
- Azure DevOps Services

Pricing on demand, cheap option per log handling.
$4/month for 10,000 error events.

Really simple installation and getting started guide, it offers many different functions and performance watching functions to check different things on our site such as custom timing events for performance tracking, like rendering time.
React Docs
Node.js APM

Runtime Analysis

extraLargeArray

- **doublerAppend : 4.565305 ms**
- **doublerInsert : 930.737197 ms**

	INSERT	APPEND
Tiny	**48.441 µs**	**116.25 µs**
Small	**59.719 µs**	**123.069 µs**
Medium	**196.979 µs**	**183.988 µs**
Large	**7.832142 ms**	**756.311 µs**
ExtraLarge	**930.737197 ms**	**4.565305 ms**

Key: Tiny: 10 Small: 100 Medium: 1000 Large: 10000 ExtraLarge: 100000

The reasoning behind why Insert is O(n) and Append is O(1) is: append just pushes the next number to the end of the array without changing any of the indexes of the following array elements while inserting with unshift, has to increment all of the elements that are already in the Array, making it O(n) as it has to iterate over n elements and push is O(1) because it only has to deal with its own element.