

# Running Node.js on customer machines

## API design for building a cloud-native application

Atishay Jain

Senior Computer Scientist, Adobe

@atishay811







# Atishay Jain

Senior Computer Scientist, Adobe

Part of many iconic software products from Photoshop and Illustrator to Adobe Capture. Creator of award winning augmented reality mobile apps from the IxDA Interaction Design award to the FWA App of the day. 10 years of industry experience working on all platforms from desktop, frontend, backend and mobile.

<https://atishay.me>

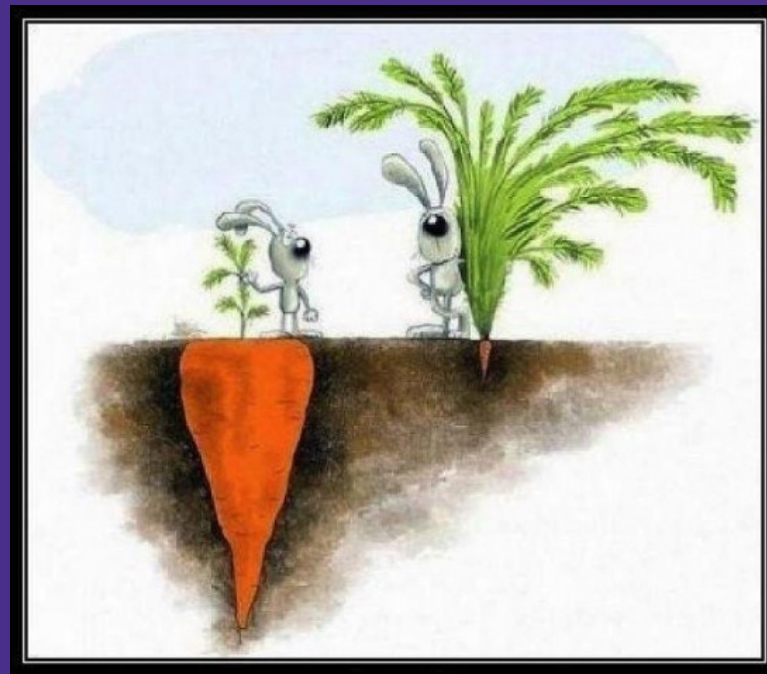
<https://linkedin.com/in/atishay>





# Agenda

- Two definitions of Cloud Native
- Making the “Cloud” Native
- History of computing
- Where does JavaScript fit in
- Running Node.js as a background process
- Its all about use cases







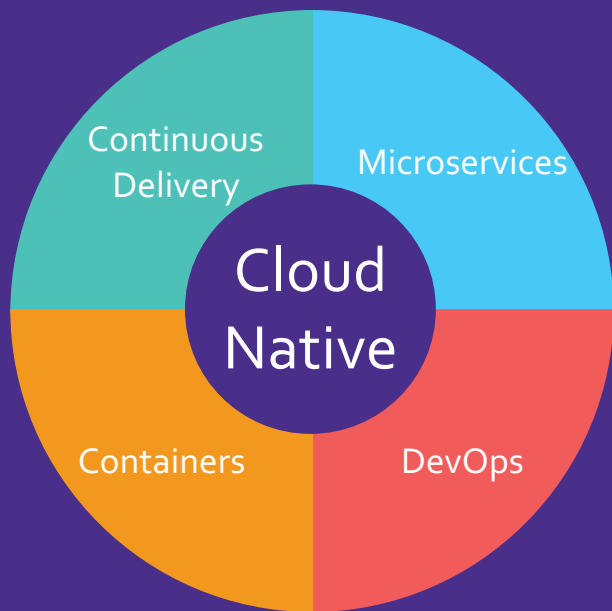
# Two definitions of Cloud Native





# Two definitions of Cloud Native

## Cloud Native for developers

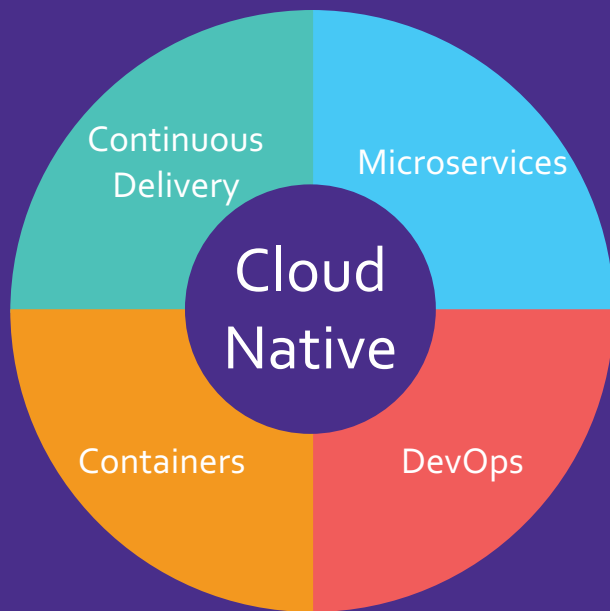




# Two definitions of Cloud Native

Cloud Native for developers

Cloud Native for users

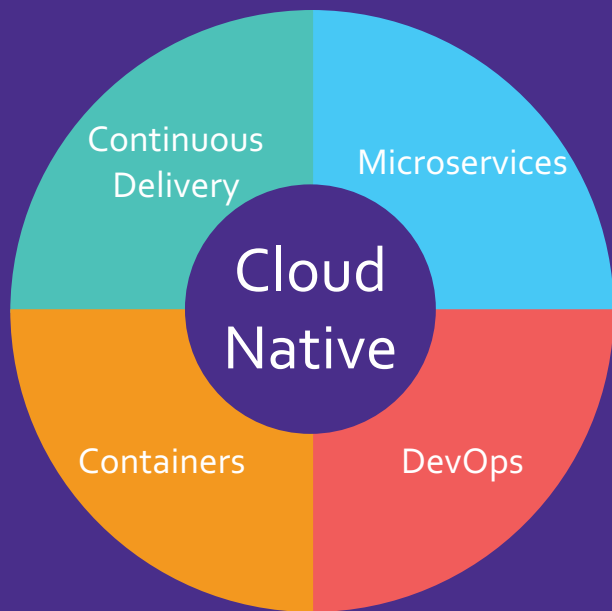




# Two definitions of Cloud Native

Cloud Native for developers

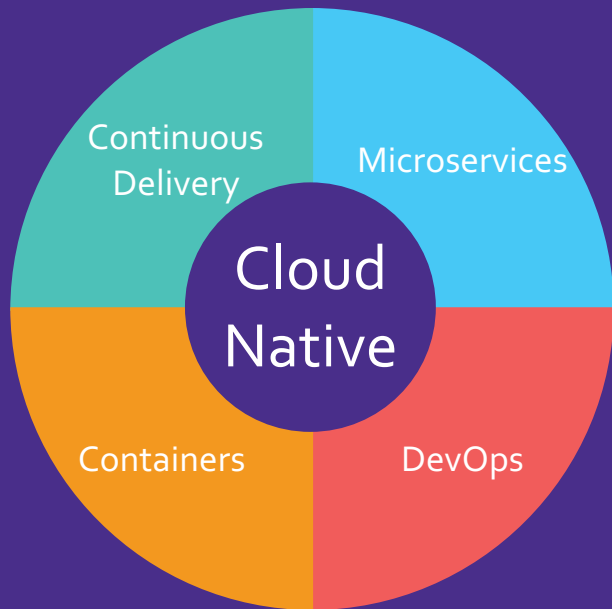
Cloud Native for users





# Two definitions of Cloud Native

Cloud Native for developers



Cloud Native for users





# What is cloud & native for the users?

## User's Cloud

- Collaboration
- Synchronization
- Automatic updates

## User's Native

- Realtime performance.
- Handle Flaky network.
- Offline support.



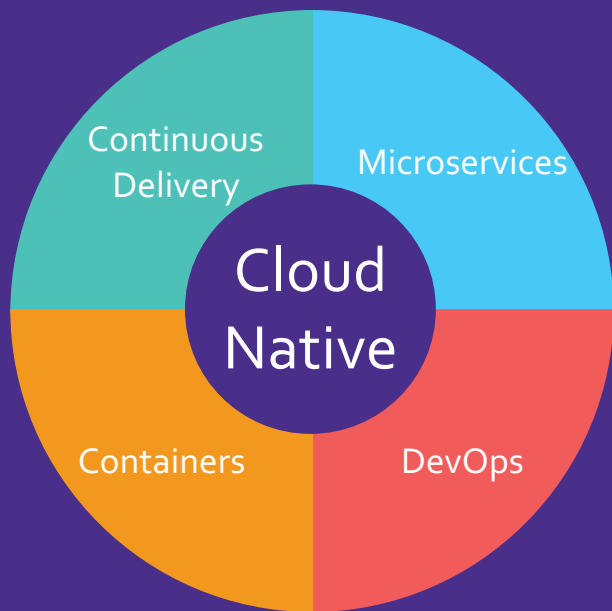


# Making the “Cloud” Native



# Making the “Cloud” Native

## Cloud Native for developers



## User's Cloud

- Collaboration
- Synchronization
- Automatic updates

## User's Native

- Realtime performance.
- Handle Flaky network.
- Offline support.



# Making the “Cloud” Native

Fighting with the laws of  
Physics.



Image from [giphy](#)



# Making the “Cloud” Native

- Multi-regions
- CDN
- Edge computing
- ??

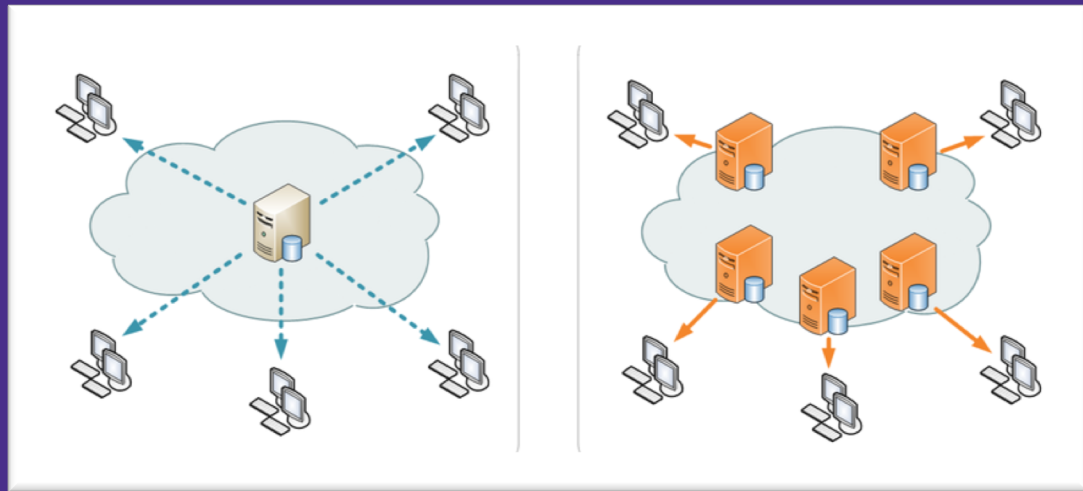


Image from [wikipedia](#)



# Making the “Cloud” Native

Does it feel native?

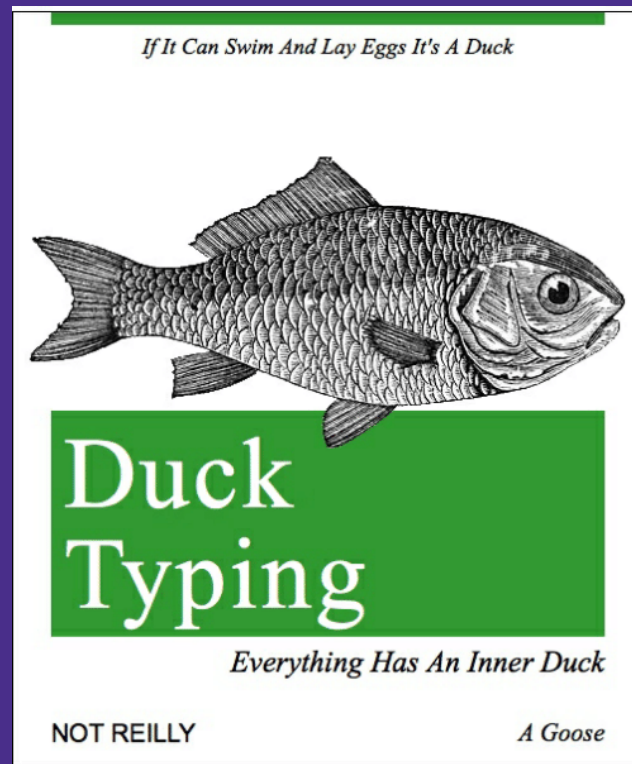
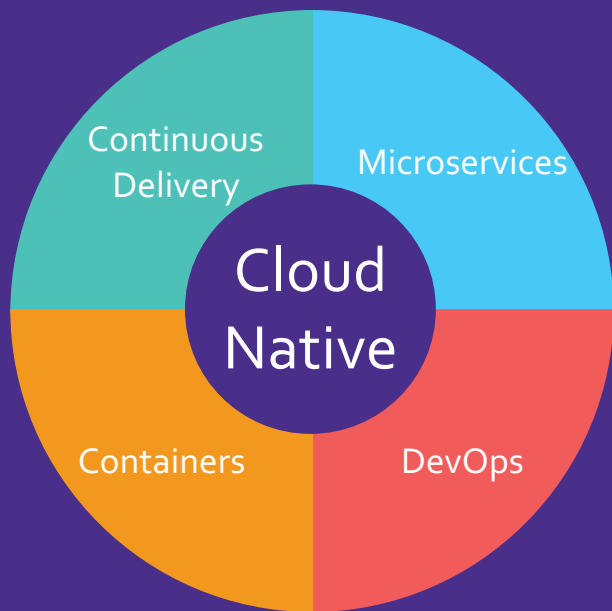


Image from [me.me](http://me.me)



# Making the “Cloud” Native

## Cloud Native for developers



## User's Cloud

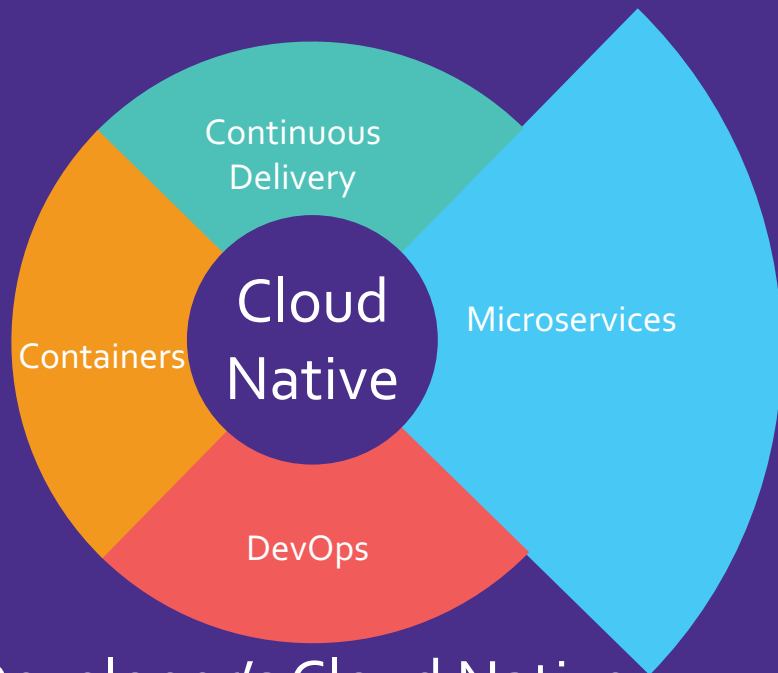
- Collaboration
- Synchronization
- Automatic updates

## User's Native

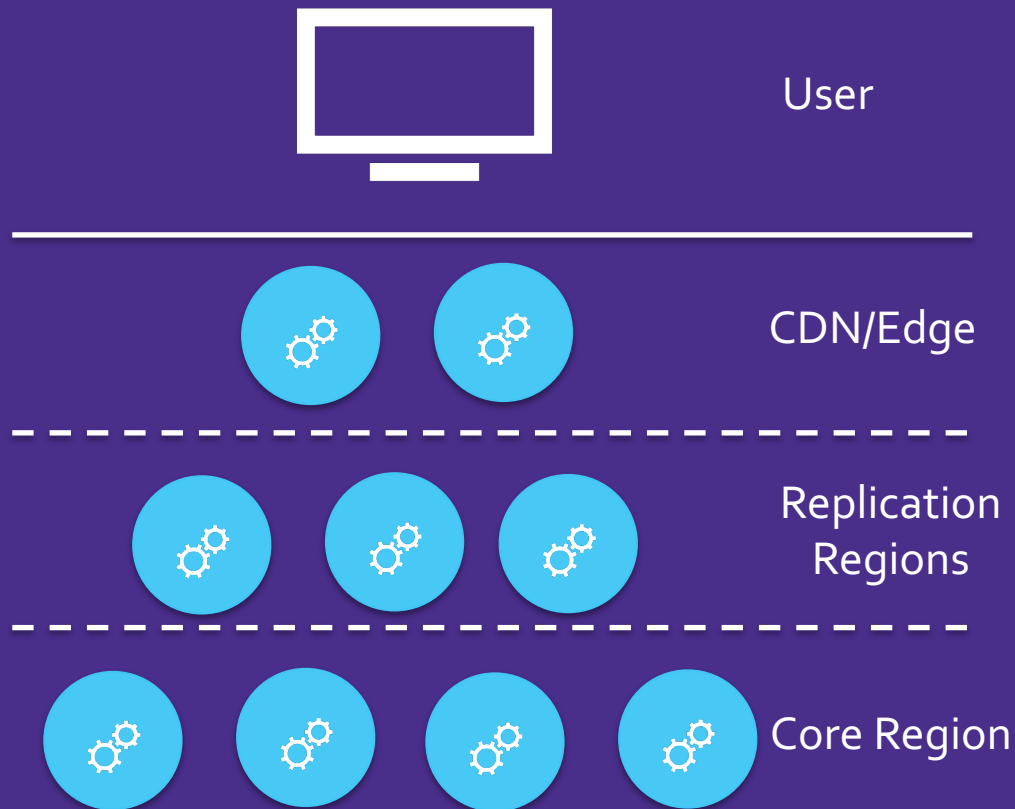
- Realtime performance.
- Handle Flaky network.
- Offline support.



# Making the “Cloud” Native



Developer's Cloud Native





# Making the “Cloud” Native

Microservice as Daemon

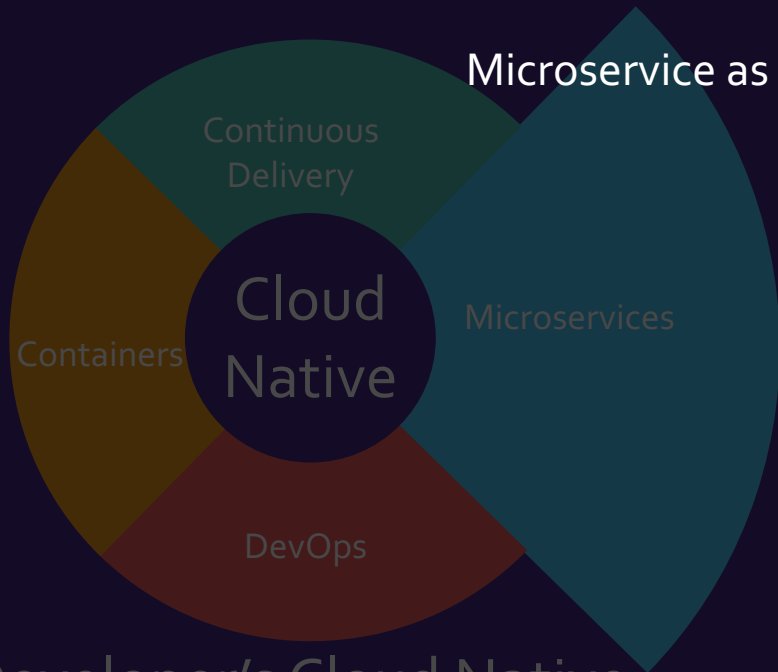


User

CDN/Edge

Replication  
Regions

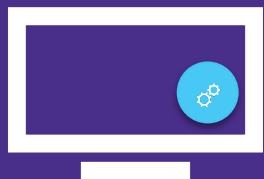
Core Region



Developer's Cloud Native



# Making the “Cloud” Native



Microservice as Daemon

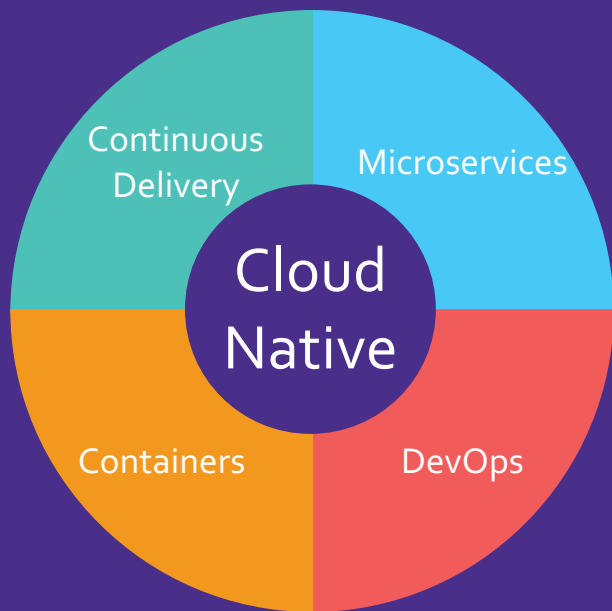
## Bad Idea

- Security?
- How about updates?
- Offline support?
- Where is the truth?



# Making the “Cloud” Native

## Cloud Native for developers



## User's Cloud

- Collaboration
- Synchronization
- Automatic updates

## User's Native

- Realtime performance.
- Handle Flaky network.
- Offline support.

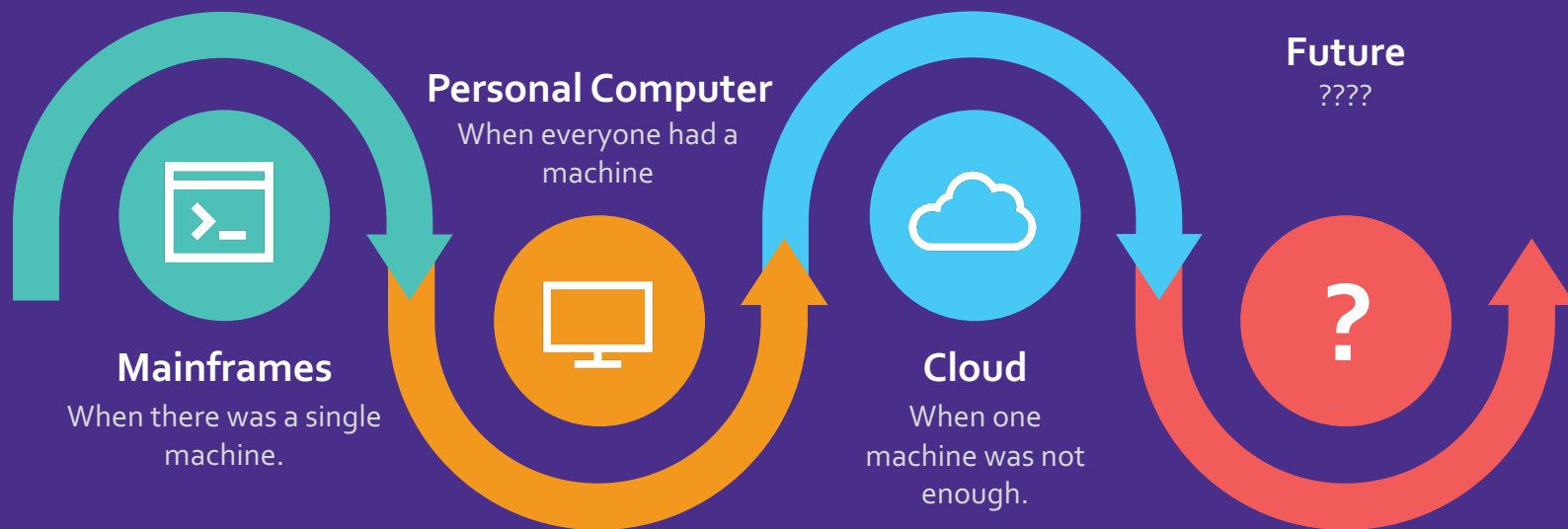




# History of Computing

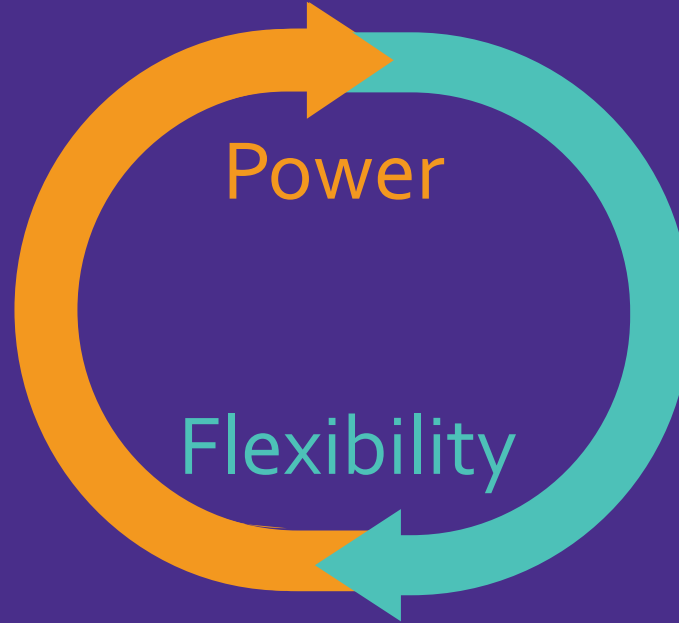


# History of Computing





# History of Computing







# Where does JavaScript fit in?



# Where does JavaScript fit in?

- Built for network based async communication.
- Heavily invested language.
- The language designers have been challenged with making the language look native.

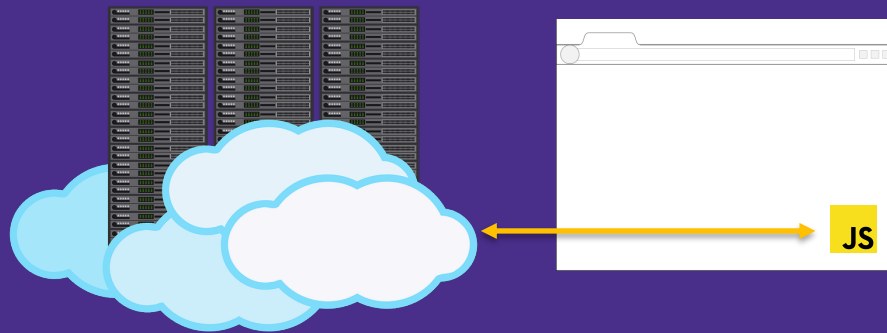


Image from [Pixabay](#), [Wikipedia](#)



# Where does JavaScript fit in?

## Service Workers

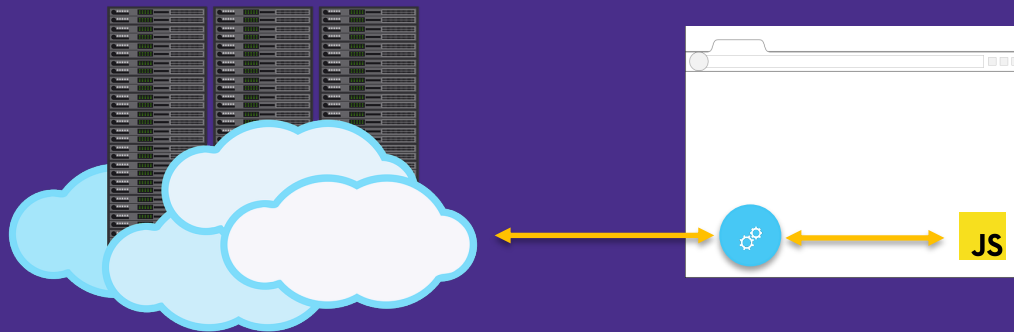


Image from [Pixabay](#), [Wikipedia](#)



# Where does JavaScript fit in?

Cloud Native for developers

Service Workers

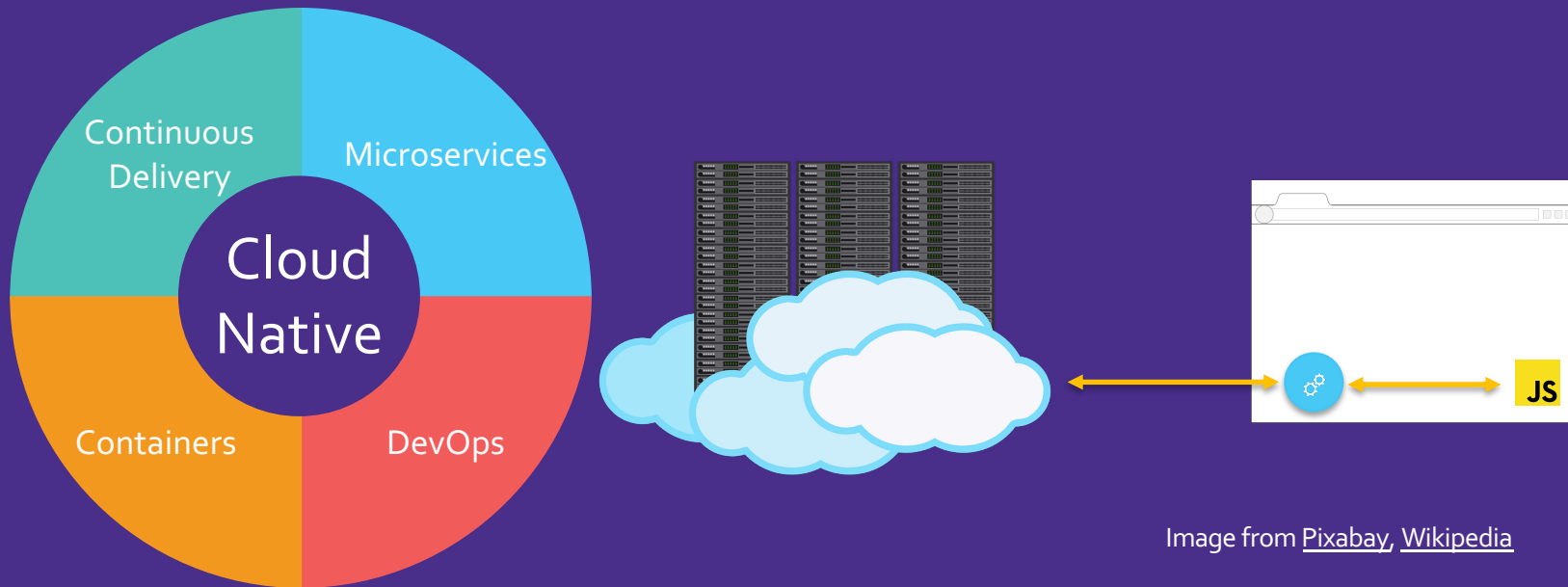


Image from [Pixabay](#), [Wikipedia](#)



# Where does JavaScript fit in?

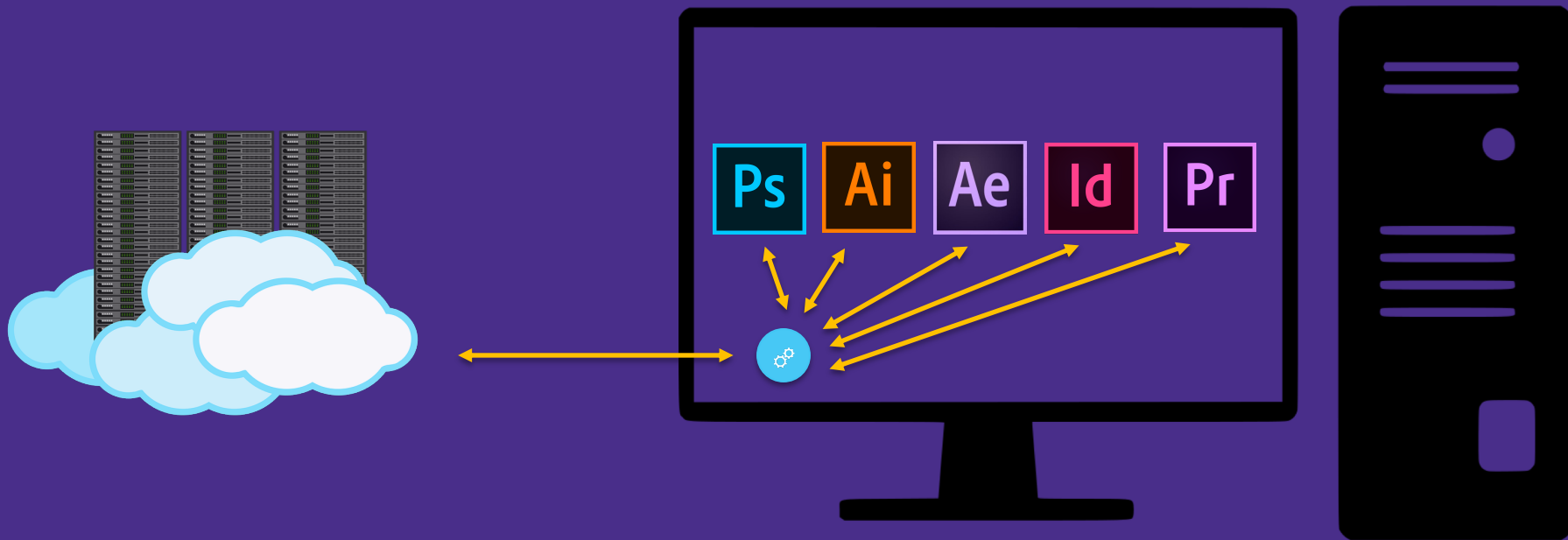
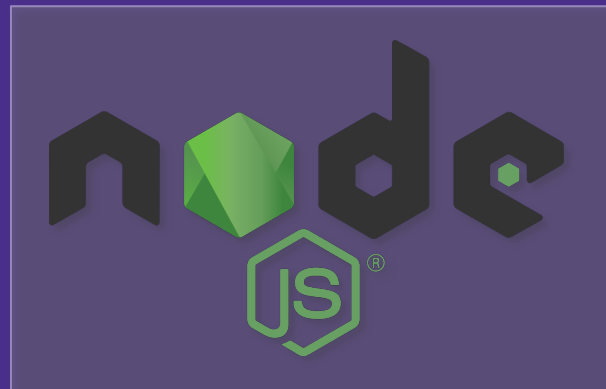


Image from [Pixabay](#), [Pixabay](#), [Wikipedia](#)



# Where does JavaScript fit in?

- Built to be run 24x7.
- Asynchronous.
- I/O Optimized.
- Mostly self-contained.



Logo from [Wikipedia](#)



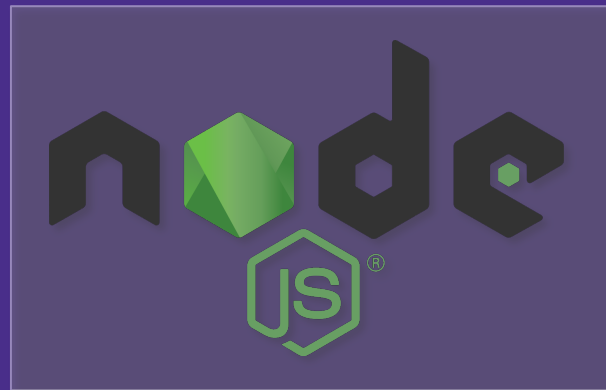
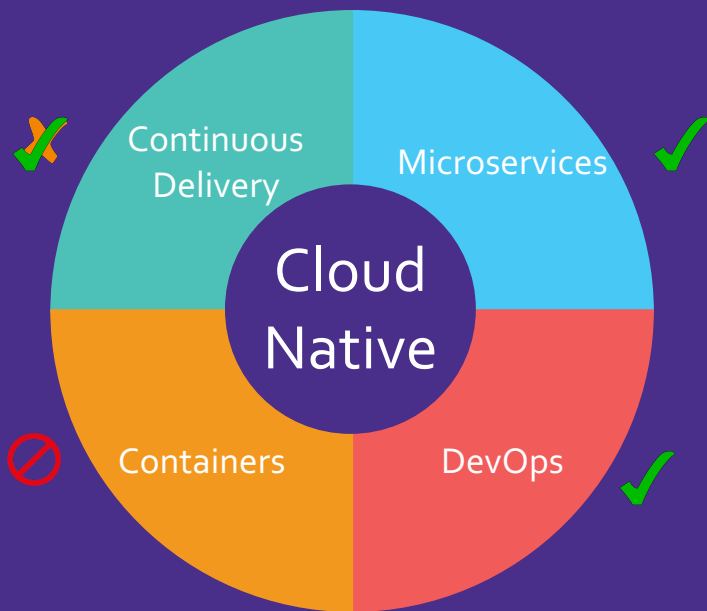


# Running Node.js as a background process



# Running Node.js as a background process

Cloud Native for developers



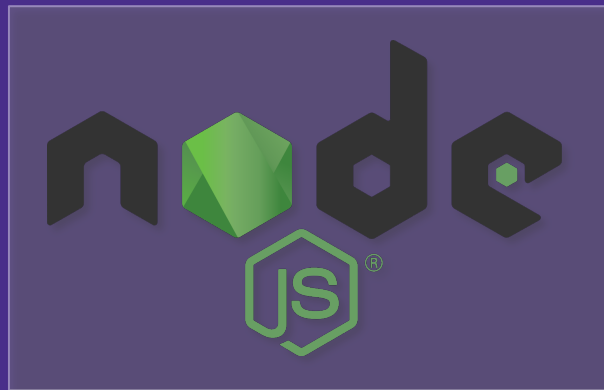
Logo from [Wikipedia](#)



# Running Node.js as a background process

## Caveats with Node.js

- Built for the server.
- No proxy support built in.
- Packaging is messy.



Logo from [Wikipedia](#)





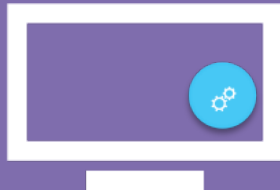
# Its all about use cases



# Conclusion



Cloud & native for user



Service as a daemon



Node.js on desktop





# DEVNET Create