NODE.JS FOR BUILDING MODERN APPLICATIONS

Rammohan Chitimilla
Senior Consultant
May 3rd, 2017
ABOUT NODE.JS
WHAT IS NODE.JS?

- JavaScript based servers side app development platform
- Built on top of Google’s V8 JavaScript Engine
- Over 300,000 Node.js modules
- Open source and community driven
  - Node.js foundation
  - Mission: enable widespread adoption and help accelerate development of Node.js and other related modules
WHAT IS NODE.JS?

• Asynchronous / Event-Driven
  • Allows applications to scale, serve multiple simultaneous connections, by efficiently using CPU cycles and without requiring more memory.
• Fast Code Execution
  • JavaScript code is compiled into native code, native code is executed rather than interpreting the JavaScript at runtime.
• Highly Scalable
WHAT IS NODE.JS?

Node.JS Processing Model

Clients send HTTP requests to Node.JS server...

Event loop returns result to client

Responses are sent to main thread via callback

Event loop is woken up by OS, passes request + response objects as JavaScript closures to worker functions with callbacks

Long-running jobs run on worker threads..

Non-blocking Worker
(internal C++ threadpool)

Stackoverflow.com
WHAT IS NODE.JS?

- Faster application development cycles
- Allows for full stack development using JavaScript for both front and back ends
- Lower hardware resource usage due to asynchronous nature
- Support for integration with various backend systems and protocols
- Exponential growth in the usage of the platform (Web, IOT, Robotics and for Tooling)
WHY NODE.JS WAS DEVELOPED?

• Address challenges when using existing web servers for building concurrency into the platform
• Simplify the development of applications that can support concurrency
  • Without getting deep into threads
MODERN APPS
MODERN APPS

What are these?

- Low latency
- Highly concurrent
- Support Microservices
- Faster/Shorter development cycles
- Support DevOps
- Built around a container platform (OCP)
SUPPORTING ECOSYSTEM

What are these?

• Node.js packages
  www.npmjs.com

• Frameworks
  Express.js
  Hapi
  Koa
  +Others

• Testing
  Mocha
  Chai
  Sinon.JS
  Node Inspector
REFERENCE ARCHITECTURE

MEAN STACK

Web
• Angular JS

Mobile
• Hybrid
• Native (iOS/Android)

Middleware
• Node.js
• Redis (Caching)
• MongoDB (persistence)

Backend
• Any other system (Authentication,..)
WHO IS USING NODE.JS

AND MANY MORE
CONSIDERATIONS IN USING NODE.JS

• Development skills
  • JavaScript
CONSIDERATIONS IN USING NODE.JS

• **Learning curve**
  • There is a learning curve associated with using frameworks and Node.js modules
  • Architecting the applications to leverage the platform capabilities
  • Callback hell
CONSIDERATIONS IN USING NODE.JS

• Security
  • Anyone can publish modules
  • Local repo (Sinopia)
  • Managing module dependencies
    Shrinkwrap
CONSIDERATIONS IN USING NODE.JS

- DevOps and Microservices
  - Supporting infrastructure
  - Leveraging OCP
  - Seneca
CONSIDERATIONS IN USING NODE.JS

• **Performance**
  • High I/O applications vs CPU intensive applications
  • Add-ons for C/C++
CONSIDERATIONS IN USING NODE.JS

Scaling of applications
- PM2
- OCP
RED HAT AND NODE.JS
RED HAT & NODE.JS

• Platinum member of the Node.js foundation
  • https://nodejs.org/en/blog/announcements/welcome-redhat/
• Leveraging the Node.js for over 5 years for building mobile and web applications
  • Has the expertise in helping enterprises deploy Node.js in building next generation mobile and web applications
• Red Hat Mobile Application Platform (RHMAP) is built on top of Node.js and supports developing Node.js based applications
• Helped clients in various sectors in building Node.js based applications
  • Pharmaceuticals
  • Healthcare
  • Engineering
  • Financial
  • Energy
RED HAT AND NODE.JS

- Best Practices
- Application Architecture and Development
- Security
- Driving Digital Transformation, DevOps, and Microservices
  - OCP
  - Ansible
  - 3 Scale
  - Open Innovation Labs
  - RHMAP
START WITH DISCOVERY

Node.js Discovery Session

1. Identify business drivers, use cases, and challenges
2. Identify potential technologies and solutions
3. Create an action plan to address opportunities
RED HAT CONSULTING ENTERPRISE

THE RED HAT CONSULTING APPROACH

**DISCOVERY SESSION**
Review and capture requirements, processes, workload, and environment details as it pertains to developing Node.js applications.

**WORKSHOPS**
Provide target-state design patterns and architecture for a Node.js solution based on Red Hat expertise and help build the roadmap identifying levels of effort and timelines for organizational adoption.

**IMPLEMENTATION**
Deploy target Node.js application architecture within client. Develop, test, and deploy target app. Integrate with customer’s environment and mentor client organization.
APPROACH

Phase 0
- Discovery Session

Phase 1
- Node.js Pilot or Smart Start
- Cloud and Back-End Service Development
- Full App Dev
- Node.js Based Solution Development
- Microservices CoE Development and Enterprise Architecture Strategy Guidance
- Open Innovation Labs Residency

Phase 2
- Innovation Labs Off-ramp to Production
- Custom Consulting