

BETTER DEVEX AT NETFLIX

POLYGLOT AND CONTAINERS

Mike McGarr
[@SonOfGarr](#)

ORIGINAL DE **NETFLIX**

MARVEL **DAREDEVIL**

★★★★★ 2015 16 1 Temporada HD 5.1

Matt Murdock, ciego desde niño, lucha contra la injusticia en Hell's Kitchen, Nueva York, de día como abogado y de noche como el superhéroe Daredevil.



Popular en Netflix



Visto recientemente





Datacenter
Monolithic Java application
Oracle database



AWS
Java microservices
NoSQL



A Java shop

BETTER DEVEX AT NETFLIX

POLYGLOT AND CONTAINERS

Mike McGarr
[@SonOfGarr](#)

JAWA

AT NETFLIX

SUPPORTING JAVA

Bootcamp

Build

Dependencies

Cont. Integration

Package

Baking

Deploy

RPC

Discovery

Metrics

Persistence

Data Pipeline

Security

Dynamic Props.

Dashboards

SUPPORTING JAVA

- **Developer Tools**

Bootcamp

Build

Dependencies

Cont. Integration

Package

Baking

Deploy

RPC

Discovery

Metrics

Persistence

Data Pipeline

Security

Dynamic Props.

Dashboards

SUPPORTING JAVA

- **Developer Tools**
- **Build and CI**



SUPPORTING JAVA

- **Developer Tools**
- **Build and CI**
- **Runtime platform**

Bootcamp

Build

Dependencies

Cont. Integration

Package

Baking

Deploy

RPC

Discovery

Metrics

Persistence

Data Pipeline

Security

Dynamic Props.

Dashboards

SUPPORTING JAVA

- **Developer Tools**
- **Build and CI**
- **Runtime platform**
- **Insight Engineering**

Bootcamp

Build

Dependencies

Cont. Integration

Package

Baking

Deploy

RPC

Discovery

Metrics

Persistence

Data Pipeline

Security

Dynamic Props.

Dashboards

SUPPORTING JAVA

- **Developer Tools**
- **Build and CI**
- **Runtime platform**
- **Insight Engineering**
- **Cloud Security**

Bootcamp

Build

Dependencies

Cont. Integration

Package

Baking

Deploy

RPC

Discovery

Metrics

Persistence

Data Pipeline

Security

Dynamic Props.

Dashboards

SUPPORTING JAVA

- **Developer Tools**
- **Build and CI**
- **Runtime platform**
- **Insight Engineering**
- **Cloud Security**
- **Performance Engineering**



SUPPORTING JAVA

- **Developer Tools**
- **Build and CI**
- **Runtime platform**
- **Insight Engineering**
- **Cloud Security**
- **Performance Engineering**
- **Cloud Data Engineering**



SUPPORTING JAVA

- **Developer Tools**
- **Build and CI**
- **Runtime platform**
- **Insight Engineering**
- **Cloud Security**
- **Performance Engineering**
- **Cloud Data Engineering**
- **Demand Engineering**



SUPPORTING JAVA

- **Developer Tools**
- **Build and CI**
- **Runtime platform**
- **Insight Engineering**
- **Cloud Security**
- **Performance Engineering**
- **Cloud Data Engineering**
- **Demand Engineering**
- **Reliability Engineering**



Build and deployment

HOW DO NETFLIX ENGINEERS PUSH CODE?

HOW DO NETFLIX ENGINEERS PUSH CODE?

1. Build (w/ Gradle)

HOW DO NETFLIX ENGINEERS PUSH CODE?

- 1. Build (w/ Gradle)**
- 2. Bake (immutable servers)**

HOW DO NETFLIX ENGINEERS PUSH CODE?

- 1. Build** (w/ Gradle)
- 2. Bake** (immutable servers)
- 3. Deploy** (w/ Spinnaker)

THE PAVED ROAD



Jenkins

artifactory

bakery



Base AMI



spinnaker



A collection of Gradle plugins, built by Netflix

Repeatable builds

Going back to a previous commit of your code and building it from source doesn't guarantee exactly the same result. Your transitive dependency graph can change in subtle ways, even if you are careful to pin your dependencies. Nebula can help you lock your resolved dependency graph into source control quickly and easily.

Immutable deployments

If you've built an app and need to install it on a server, there's no better approach than a native OS package. Nebula provides a simple DSL that allows Java apps, as well as non-Java apps, to produce an RPM or Debian package easily.

Eliminate boilerplate

Nebula helps engineers remove boilerplate in Gradle build files, and makes building software the Netflix way easy. This reduces the cognitive load on developers, allowing them to focus on writing code.

Built for the Enterprise

Nebula is a collection of Gradle plugins built for Netflix engineers to eliminate boilerplate build logic and provide sane conventions. We chose Gradle for our underlying build system because we feel Gradle is the best build tool for Java applications.

In order to understand the value that Nebula provides, let's look at a Gradle build file for a typical Java web project.

```
apply plugin: 'java'
apply plugin: 'war'
apply plugin: 'application'

mainClassName = 'com.netflix.sampleApp.Application'
```

NEBULA OSPACKAGE

Package your Java application as a `.deb` or `.rpm`

```
build.gradle
1  apply plugin: 'groovy'
2  apply plugin: 'spring-boot'
3  apply plugin: 'application'
4  apply plugin: 'war'
5  apply plugin: 'os-package'
6
7  sourceCompatibility = "1.8"
8  targetCompatibility = "1.8"
9
10 version = "1.0.0"
11 mainClassName = "com.netflix.HelloWorldApplication"
12
13 buildscript {
14     repositories {
15         jcenter()
16         mavenCentral()
17     }
18     dependencies {
19         classpath 'org.springframework.boot:spring-boot-gradle-plugin:1.2.3.RELEASE'
20         classpath 'com.netflix.nebula:gradle-ospackage-plugin:2.2.6'
21     }
22 }
23
24 configurations {
25     providedRuntime
26 }
27
28 repositories {
29     jcenter()
30     mavenCentral()
31 }
32
33 dependencies {
34     compile "org.codehaus.groovy:groovy"
35     compile "org.springframework.boot:spring-boot-starter-web"
36     compile "org.springframework.boot:spring-boot-starter-actuator"
37     providedRuntime 'org.springframework.boot:spring-boot-starter-tomcat'
38     testCompile "org.springframework.boot:spring-boot-starter-test"
39 }
40
41 ospackage {
42     requires('default-jdk')
43     requires('tomcat7')
44
45     from(war) {
46         into('/var/lib/tomcat7/webapps')
47         rename(/.*/ , 'ROOT.war')
48     }
49
50     preInstall('sudo rm -rf /var/lib/tomcat7/webapps/ROOT')
51 }
```

© J. Michael McGarr, 2018


```
build.gradle
1 apply plugin: 'groovy'
2 apply plugin: 'spring-boot'
3 apply plugin: 'application'
4 apply plugin: 'war'
5 apply plugin: 'os-package'
6
7 sourceCompatibility = "1.8"
8 targetCompatibility = "1.8"
9
```

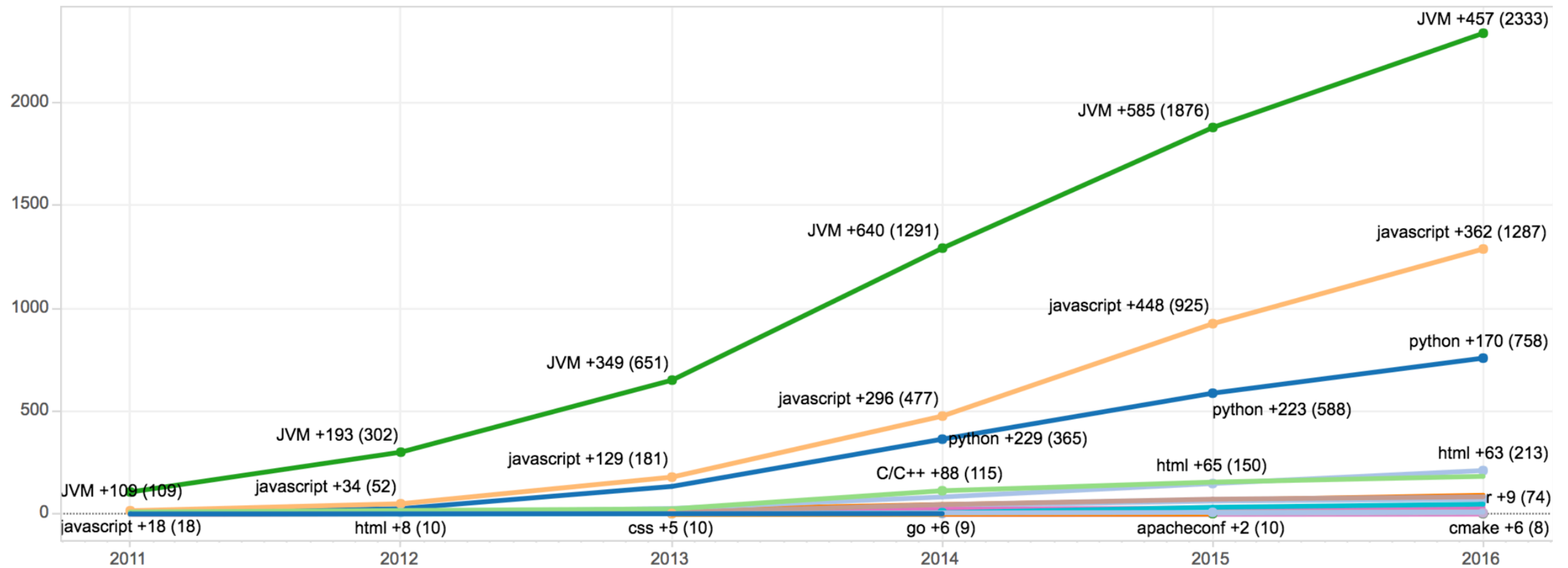
```
41 ospackage {
42     requires('default-jdk')
43     requires('tomcat7')
44
45     from(war) {
46         into('/var/lib/tomcat7/webapps')
47         rename(/.*/, 'ROOT.war')
48     }
49
50     preInstall('sudo rm -rf /var/lib/tomcat7/webapps/ROOT')
51 }
```

```
37 providedRuntime "org.springframework.boot:spring-boot-starter-tomcat"
38 testCompile "org.springframework.boot:spring-boot-starter-test"
39 }
40
41 ospackage {
42     requires('default-jdk')
43     requires('tomcat7')
44
45     from(war) {
46         into('/var/lib/tomcat7/webapps')
47         rename(/.*/, 'ROOT.war')
48     }
49
50     preInstall('sudo rm -rf /var/lib/tomcat7/webapps/ROOT')
51 }
```

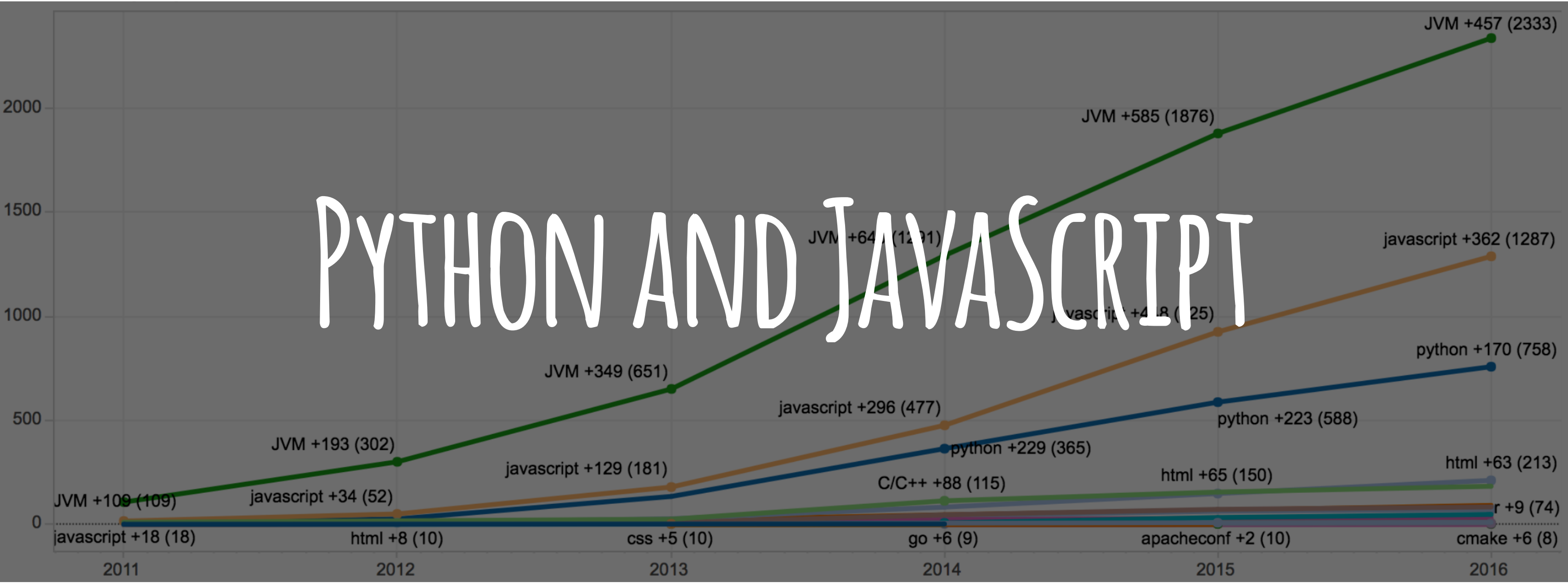

NON-NUVA

AT NETFLIX

NON-JAVA LANGUAGE GROWTH ROSE 2014



PYTHON AND JAVASCRIPT



NODE.JS APP

produce a .deb

```
build.gradle
1 apply plugin: 'nebula'
2 apply plugin: 'nebula-ospackage'
3
4 nebula {
5     moduleOwner = 'faketeam@netflix.com'
6     moduleEmail = 'faketeam@netflix.com'
7 }
8
9 ospackage {
10     requires('platform-sidecar')
11     from(project.file('dist')) {
12         into '/apps/apache/htdocs'
13     }
14
15     postInstall '/sbin/chkconfig nflx-tomcat off'
16     postInstall "sed -i -e 's/proxy_enabled=.*\n proxy_enabled=false/' /etc/default/nflx-httpd"
17 }
```

THIS WORKS

right?

Really!? I have to use
Gradle **for Node.js?**

Yes. We are a Java shop.

Java

- Bootcamp
- Build
- Dependencies
- Cont. Integration
- Package
- Baking
- Deploy
- RPC
- Discovery
- Metrics
- Persistence
- Data Pipeline
- Security
- Dynamic Props.
- Dashboards

JS

- Bootcamp
- Build
- Dependencies
- Cont. Integration
- Package
- Baking
- Deploy
- RPC
- Discovery
- Metrics
- Persistence
- Data Pipeline
- Security
- Dynamic Props.
- Dashboards

Python

- Bootcamp
- Build
- Dependencies
- Cont. Integration
- Package
- Baking
- Deploy
- RPC
- Discovery
- Metrics
- Persistence
- Data Pipeline
- Security
- Dynamic Props.
- Dashboards

other?

- Bootcamp
- Build
- Dependencies
- Cont. Integration
- Package
- Baking
- Deploy
- RPC
- Discovery
- Metrics
- Persistence
- Data Pipeline
- Security
- Dynamic Props.
- Dashboards



tipping point

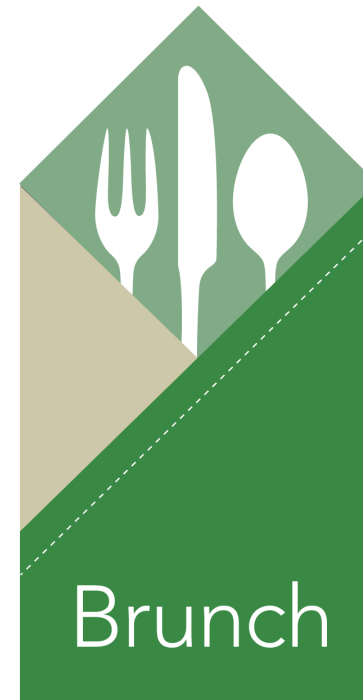
Produce Node.js .debs using Native tools

Nebula for Node.js





GRUNT



~~Build tool~~

~~Build tool~~
Developer workflow tool

Language agnostic

Native (or native-like)

reduce cognitive load



NEWT

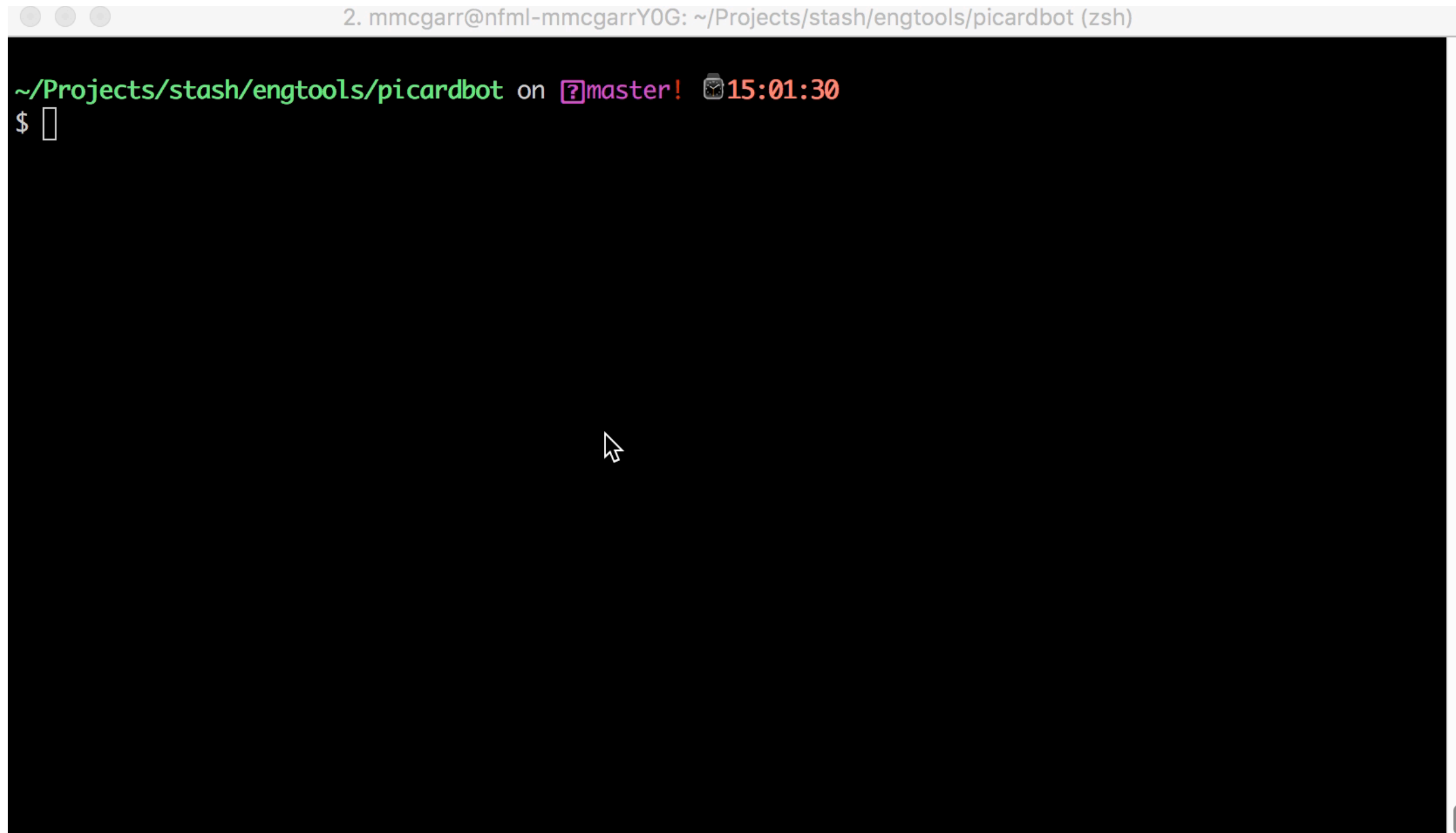
(Netflix Workflow Toolkit)



Newt for node.js

Problem # 1:
Simplify Node.js debian packaging

newt package



A terminal window with a light gray title bar containing the text "2. mmcarr@nfml-mmcarrY0G: ~/Projects/stash/engtools/picardbot (zsh)". The terminal area has a black background. The first line shows a green prompt "~ /Projects/stash/engtools/picardbot" followed by "on [?]master!" in purple, a battery icon, and the time "15:01:30" in orange. The second line shows a white prompt "\$" followed by a white cursor box. A white mouse cursor is visible in the center of the terminal area.

```
2. mmcarr@nfml-mmcarrY0G: ~/Projects/stash/engtools/picardbot (zsh)  
~/Projects/stash/engtools/picardbot on [?]master! 🔋 15:01:30  
$ █
```


Gradle/Java hidden in a Docker container

```
2. mmcgarr@nfml-mmcgarrY0G: ~/Projects/stash/engtools/picardbot (zsh)
~/Projects/stash/engtools/picardbot on [?]master! 15:02:17
$ newt package
latest: Pulling from engtools/nflx-ospackage
Digest: sha256:314387f128c7d1291729076d94d0b1e95e225712b222e86492dabedcbb85eb59
Status: Image is up to date for [REDACTED]7002/engtools/nflx-ospackage:latest
Running: /nebula-home/gradlew --offline --gradle-user-home=/gradle-cache --project-cache-dir=/gradle-
project-cache --project-dir=/src buildDeb -x lintGradle -b ./build.newt.gradle
[metrics] Build is running offline. Metrics will not be collected.
+====+
| Nebula 3.15.3 on Gradle 2.13
+====+
[buildinfo] Not using buildInfo properties file for this build.
:buildDeb

BUILD SUCCESSFUL

Total time: 16.537 secs

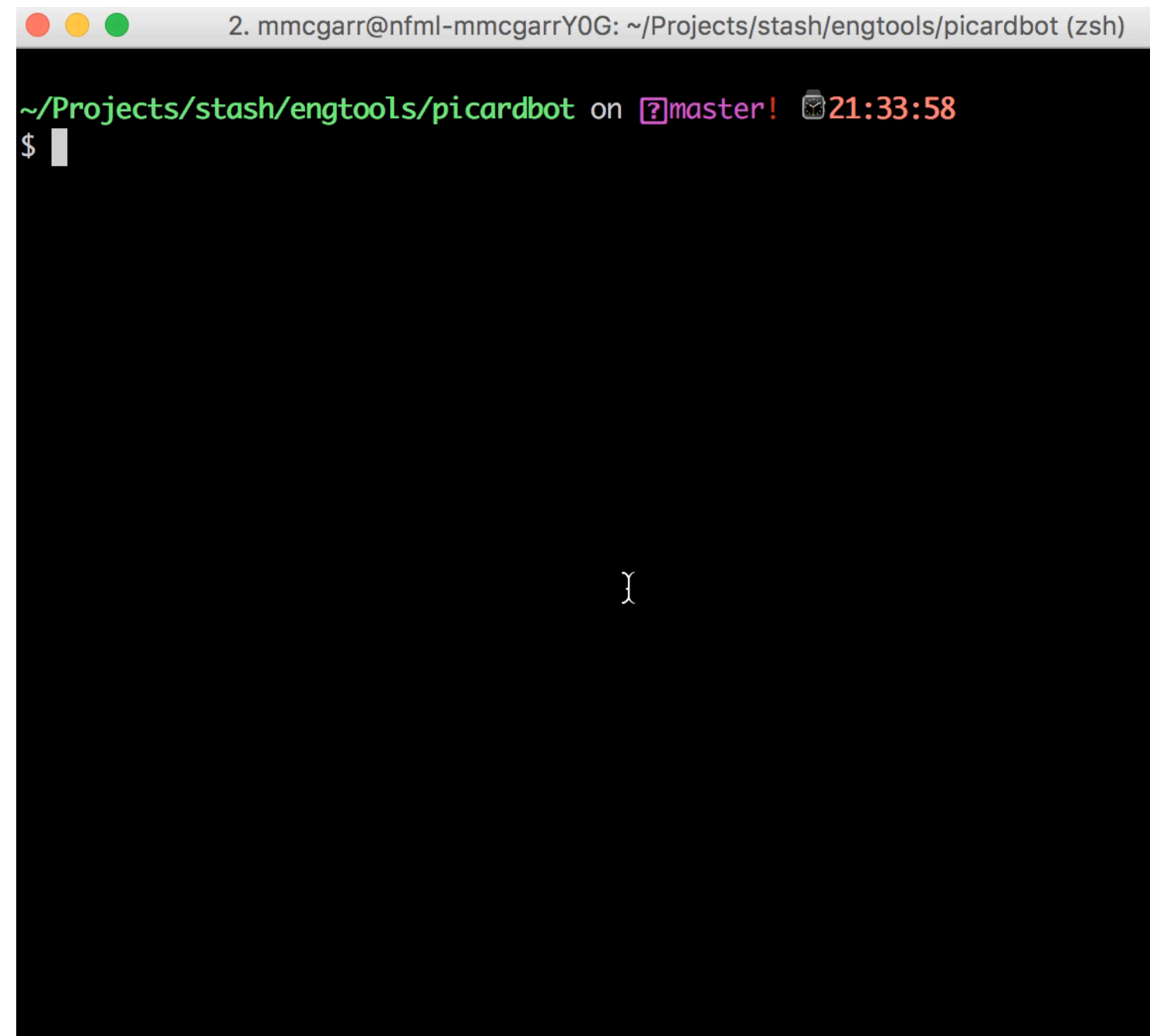
This build could be faster, please consider using the Gradle Daemon: https://docs.gradle.org/2.13/user-
guide/gradle_daemon.html

~/Projects/stash/engtools/picardbot on [?]master! 15:02:42
$ tree build
```

Node.js developers don't install Java/Gradle

Problem # 2:
Now I need Docker?

Install & configure tools per execution



```
2. mmcarr@nfml-mmcarrY0G: ~/Projects/stash/engtools/picardbot (zsh)
~/Projects/stash/engtools/picardbot on [?]master! 21:33:58
$
```

Provide consistency throughout

Problem # 3:
I still need to build!

newt build

.newt.yml

```
1 build-step: npm run-script build
```

```
2
```


Runs package.json build script:

```
npm install &&  
eslint src/** &&  
npm test
```

2. mmcarr@nfml-mmcarrY0G: ~/Projects/stash/engtools/picardbot (zsh)

~/Projects/stash/engtools/picardbot on ?master! 📧 21:50:59

\$

Problem #4:
Keep NPM/Nodejs versions consistent

DEFINE TOOL VERSIONS

```
.newt.yml
1  build-script: npm run-script build
2  tool-versions:
3    - node: 6.9.1
4    - npm: 3.10.8
5
```

newt exec

```
.newt.yml
1 build-script: newt exec npm run-script build
2 tool-versions:
3   - node: 6.9.1
4   - npm: 3.10.8
5
```


newt exec npm vs. npm

```
~/Projects/stash/engtools/picardbot on [?]master! 13:22:54
$ newt exec npm version
{ picardbot: '0.26.9',
  npm: '3.10.8',
  ares: '1.10.1-DEV',
  http_parser: '2.7.0',
  icu: '57.1',
  modules: '48',
  node: '6.9.1',
  openssl: '1.0.2j',
  uv: '1.9.1',
  v8: '5.1.281.84',
  zlib: '1.2.8' }
```

```
~/Projects/stash/engtools/picardbot on [?]master! 13:23:01
$ npm version
{ picardbot: '0.26.9',
  npm: '3.10.10',
  ares: '1.10.1-DEV',
  cldr: '30.0.2',
  http_parser: '2.7.0',
  icu: '58.1',
  modules: '51',
  node: '7.3.0',
  openssl: '1.0.2j',
  tz: '2016g',
  unicode: '9.0',
  uv: '1.10.1',
  v8: '5.4.500.45',
  zlib: '1.2.8' }
```

So now I need to type
newt exec npm
everytime?

```
~/Projects/stash/engtools/picardbot on [?]master! 13:22:54
$ newt exec npm version
{ picardbot: '0.26.9',
  npm: '3.10.8',
  ares: '1.10.1-DEV',
  http_parser: '2.7.0',
  icu: '57.1',
  modules: '48',
  node: '6.9.1',
  openssl: '1.0.2j',
  uv: '1.9.1',
  v8: '5.1.281.84',
  zlib: '1.2.8' }
```

```
~/Projects/stash/engtools/picardbot on [?]master! 13:23:01
$ npm version
{ picardbot: '0.26.9',
  npm: '3.10.10',
  ares: '1.10.1-DEV',
  cldr: '30.0.2',
  http_parser: '2.7.0',
  icu: '57.1',
  modules: '51',
  node: '7.3.0',
  openssl: '1.0.2j',
  tz: '2016g',
  unicode: '9.0',
  uv: '1.10.1',
  v8: '5.4.500.45',
  zlib: '1.2.8' }
```

alias npm="newt exec npm --"

Java

JS

Python

other?



Java	JS	Python	other?
Bootcamp	Bootcamp	Bootcamp	Bootcamp
Build	Build	Build	Build
Dependencies	Dependencies	Dependencies	Dependencies
Cont. Integration	Cont. Integration	Cont. Integration	Cont. Integration
Package	Package	Package	Package
Baking	Baking	Baking	Baking
Deploy	Deploy	Deploy	Deploy
RPC	RPC	RPC	RPC
Discovery	Discovery	Discovery	Discovery
Metrics	Metrics	Metrics	Metrics
Persistence	Persistence	Persistence	Persistence
Data Pipeline	Data Pipeline	Data Pipeline	Data Pipeline
Security	Security	Security	Security
Dynamic Props.	Dynamic Props.	Dynamic Props.	Dynamic Props.
Dashboards	Dashboards	Dashboards	Dashboards

WHAT HAVE WE GAINED?

WHAT HAVE WE GAINED?

- **Simplified packaging for all languages**

WHAT HAVE WE GAINED?


- **Simplified packaging for all languages**
- **Tooling consistency**

WHAT HAVE WE GAINED?

- **Simplified packaging for all languages**
- **Tooling consistency**
- **Tool environment isolation**

WHAT HAVE WE GAINED?

- **Simplified packaging for all languages**
- **Tooling consistency**
- **Tool environment isolation**
- **Agnostic to build tools**



UNLOCKING HIDDEN POTENTIAL

We want homogenous **apps**

app-types

app types

.newt.yml

```
1 app-type: node-ami-deploy
```

```
2
```



```
newt init --app-type node-ami-deploy
```



git



~/Projects/stash/users/mmcgarr/mmcgarr-cron 📅 13:57:04

\$ █

configurable commands

enablement through inter-sourcing

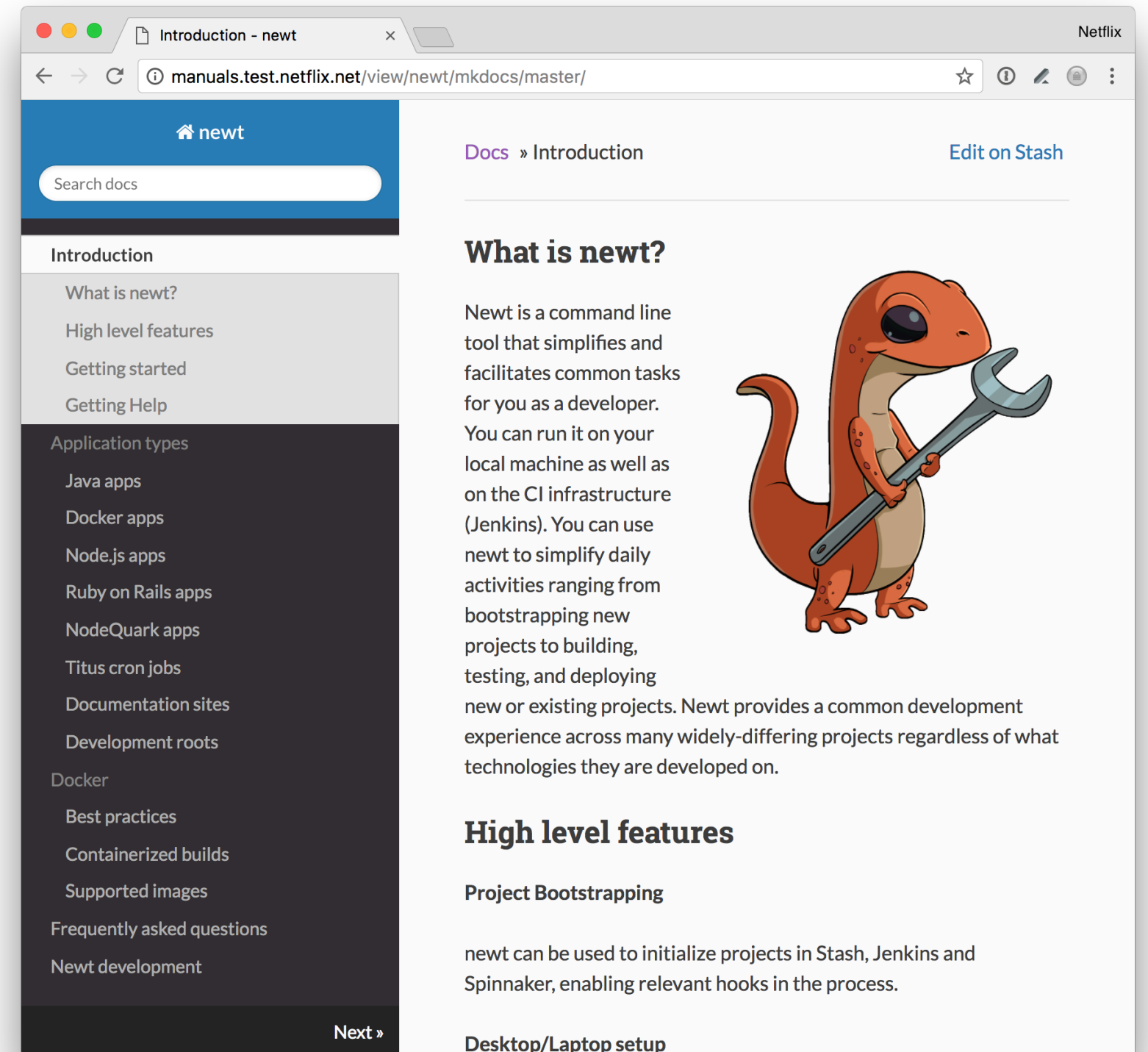
\$ newt init

APP-TYPE	DESCRIPTION	DOCS
abacus	Abacus makes it easy to develop great data visualization applications in JavaScript	http://go/abacus-docs
archer	A map reduce style platform for media processing	http://go/archernewt
cron-bash	Creates Bash job that runs once a day (1AM PDT) on Titus via a Spinnaker CRON pipeline.	http://go/newtcron
cron-groovy	Creates groovy job that runs once a day (1AM PDT) on Titus via a Spinnaker CRON pipeline.	http://go/newtcron
cron-python	Creates Python job that runs once a day (1AM PDT) on Titus via a Spinnaker CRON pipeline.	http://go/newtcron
cron-ruby	Creates Ruby job that runs once a day (1AM PDT) on Titus via a Spinnaker CRON pipeline.	http://go/newtcron
debhelper-package	Sets up a build for a Debian package using debhelper	http://go/newtdhpkg
debian-package	Sets up a build for a Debian package	http://go/newtdebpkg
devroot	create development roots to allow for ad-hoc build/testing (jdock replacement)	http://go/newtdevroot
docs-site	Uses MkDocs to create beautiful and searchable documentation sites on http://go/manuals .	(not provided)
java-project	build a Java-based service with the Runtime platform on AWS	http://go/newtjava
java-project-titus	build a Java-based service with the Runtime platform on Titus	http://go/newtjava
mantis-job	generates mantis example jobs	https://go/mantis-docs
node-ami-deploy	Painlessly build, bake and deploy node.js apps	http://go/newtjs
nodequark	Node.js container based platform for Edge UI adaptor applications talking to the Remote Service Layer (RSL)	http://go/skipperdocs
python-ami-deploy	Build/Publish Python service on AWS	http://go/newtpythonservice
python-lib	Build/Publish Python library	http://go/newtpythonlib
python-titus-deploy	Build/Publish Python service on Titus	http://go/newtpythonservice
rails-ami-deploy	Generate, build and deploy Ruby on Rails apps	http://go/newtruby

better documentation?

Developer friendly docs

```
newt init --app-type  
doc-site
```



The screenshot shows a web browser window displaying the 'Introduction' page of the 'newt' documentation. The browser's address bar shows the URL 'manuals.test.netflix.net/view/newt/mkdocs/master/'. The page has a blue header with the 'newt' logo and a search bar. A left sidebar contains a navigation menu with categories like 'Introduction', 'Application types', 'Docker', and 'Frequently asked questions'. The main content area features the heading 'What is newt?' followed by a paragraph describing the tool as a command-line utility for simplifying development tasks. To the right of the text is a cartoon illustration of an orange salamander holding a wrench. Below the main text, there are sections for 'High level features' and 'Project Bootstrapping', with the latter explaining that 'newt' can be used to initialize projects in Stash, Jenkins, and Spinnaker.

Development ecosystem complexity

ECOSYSTEM AWARENESS

ECOSYSTEM AWARENESS

- newt spinnaker start

ECOSYSTEM AWARENESS

- newt spinnaker start
- newt spinnaker browse

ECOSYSTEM AWARENESS

- newt spinnaker start
- newt spinnaker browse
- newt ci history

ECOSYSTEM AWARENESS

- newt spinnaker start
- newt spinnaker browse
- newt ci history
- newt ci list

ECOSYSTEM AWARENESS

- newt spinnaker start
- newt spinnaker browse
- newt ci history
- newt ci list
- newt ci start

ECOSYSTEM AWARENESS

- newt spinnaker start
- newt spinnaker browse
- newt ci history
- newt ci list
- newt ci start
- newt ci stop

ECOSYSTEM AWARENESS

- newt spinnaker start
- newt spinnaker browse
- newt ci history
- newt ci list
- newt ci start
- newt ci stop
- newt ci tail

ECOSYSTEM AWARENESS

- newt spinnaker start
- newt spinnaker browse
- newt ci history
- newt ci list
- newt ci start
- newt ci stop
- newt ci tail
- newt ci browse

oh yeah...support

\$ newt report-error

Newt is a ~~tool~~ platform

A stylized, cartoonish illustration of a character's face. The character has a grey, blocky head with a wide, toothy grin showing red and grey teeth. The eyes are large, circular, and red with black outlines. The character is wearing a red, textured garment. The background is a solid blue color.

TITUS

WHAT WE LEARNED!

WHAT WE LEARNED?

WHAT WE LEARNED?

- Polyglot can be expensive

WHAT WE LEARNED?

- Polyglot can be expensive
- Containers make for great tool distribution

WHAT WE LEARNED?

- Polyglot can be expensive
- Containers make for great tool distribution
- Build platforms, not just tools

WHAT WE LEARNED?

- Polyglot can be expensive
- Containers make for great tool distribution
- Build platforms, not just tools
- Provide native solutions (or native-like)

WHAT WE LEARNED?

- Polyglot can be expensive
- Containers make for great tool distribution
- Build platforms, not just tools
- Provide native solutions (or native-like)
- Reduce cognitive load

THANK YOU!

Mike McGarr (@SonOfGarr)

