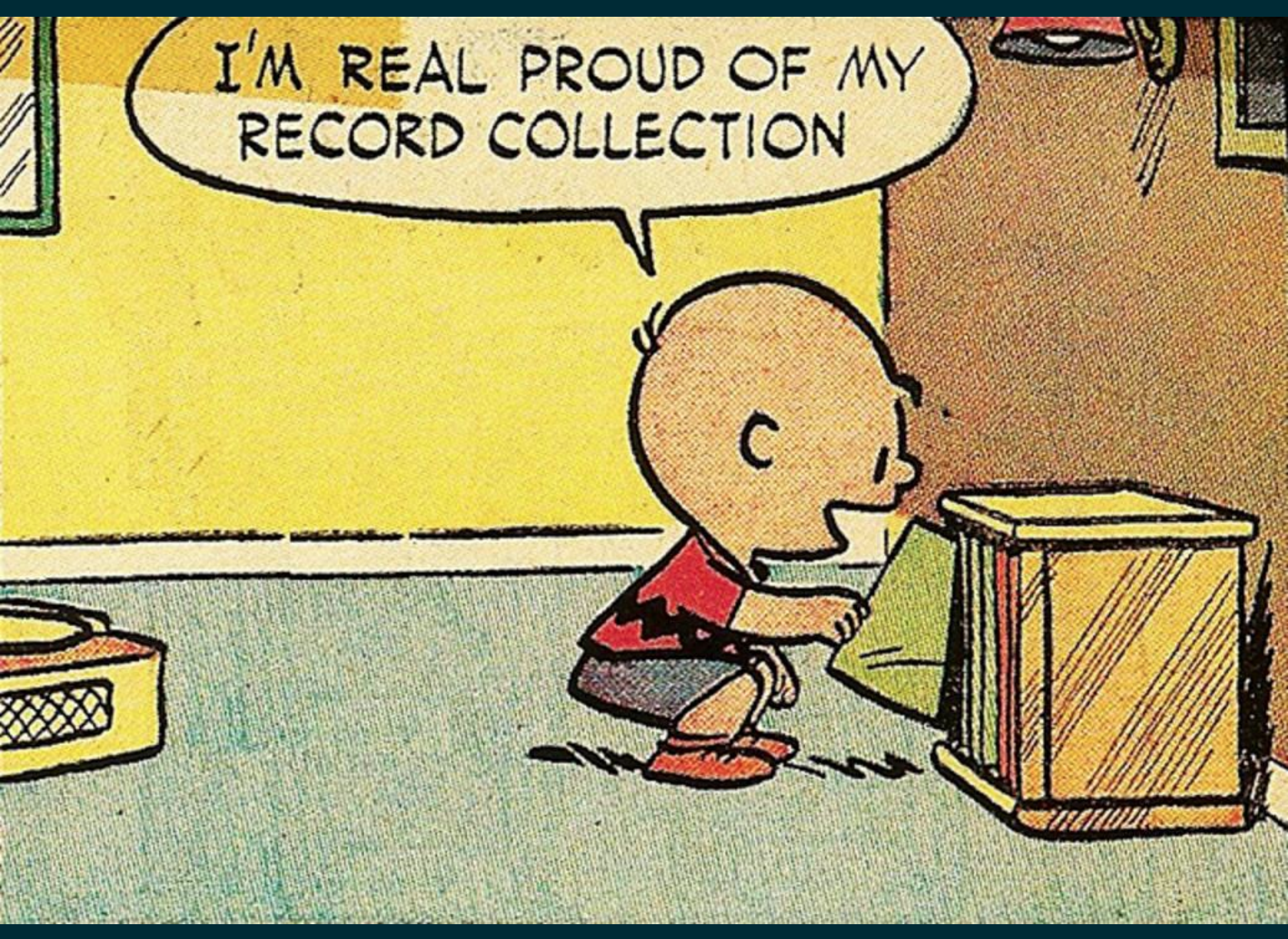
CommandBox vs Node.js

Nolan Erck
South of Shasta

About Me

- Software Consultant (southofshasta.com)
  - Software Development, Training, Design
- ColdFusion, C++, Java, jQuery, PHP, .NET, HTML5, Angular, SQL, etc...
- Manager of SacInteractive User Group
- Reformed Video Game Developer (Grim Fandango, SimPark, StarWars Rogue Squadron, etc).
- Music Junkie


I'M REAL PROUD OF MY RECORD COLLECTION

Slide deck and code samples are here:

github.com/nolanerck
commandbox-vs-node

For today's talk

- Screenshots and recorded videos
- No smoke and mirrors
- I'm merely avoiding hotel wifi

Let's get started!

We've all heard these things:

- JavaScript has all the cool toys!
- CFML is old, dying and only for legacy stuff!

Incorrect!

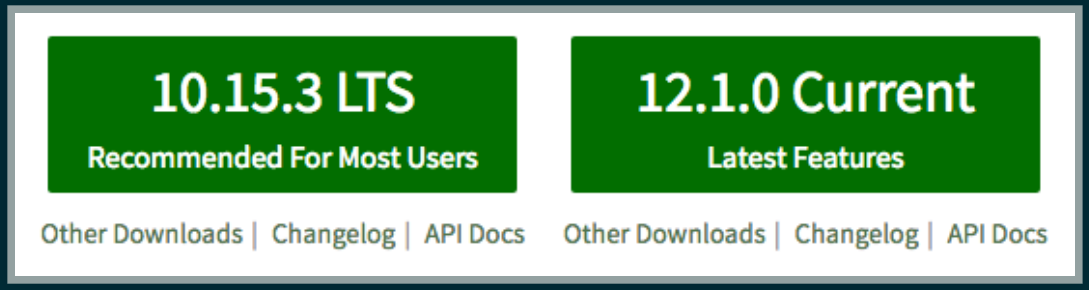
- CommandBox is here to save the day!
- All of the Node.js fundamentals are available to CFML developers

What is Node.js?

- App you install
- On laptop or server
- "JavaScript can run anywhere"
- In web browser (obviously)
- As a server side language (ala CFML)
- As a native CLI tool on your laptop

Installing Node.js

- Go to Nodejs.org
- Click one of these:



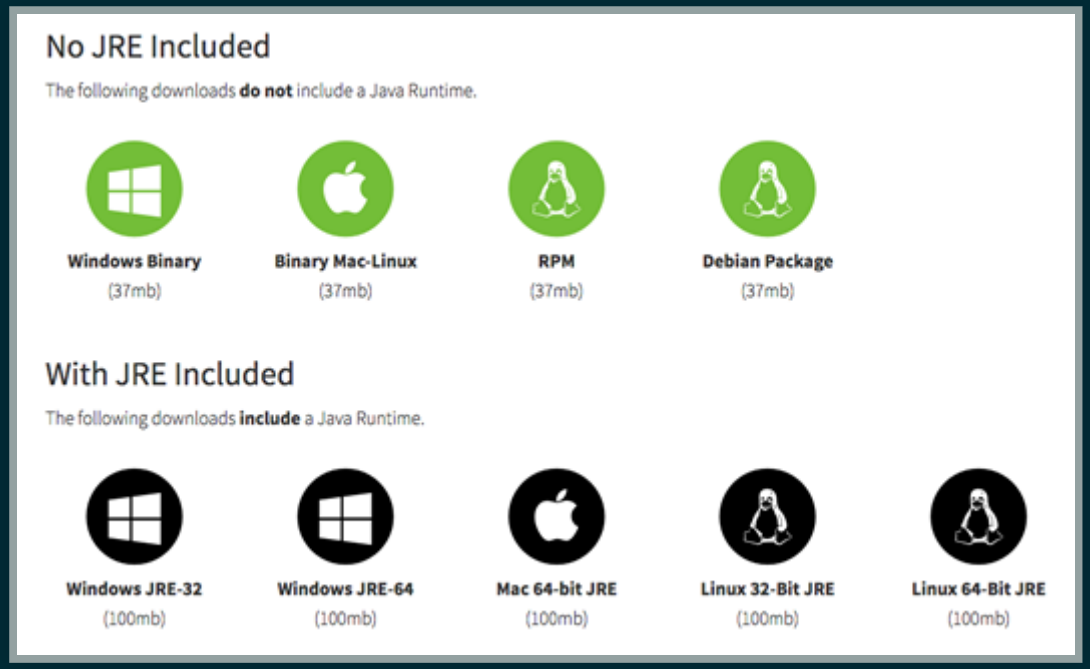
- It usually "just works"
- Regular updates are available

What is CommandBox?

- App you install
- On laptop or server
- "CFML from the command line"
- As a server side language (duh)
- As a native CLI tool on your laptop

Installing CommandBox

- Go to ortussolutions.com/products/commandbox
- Click one of these:



- It usually "just works"
- Regular updates are available

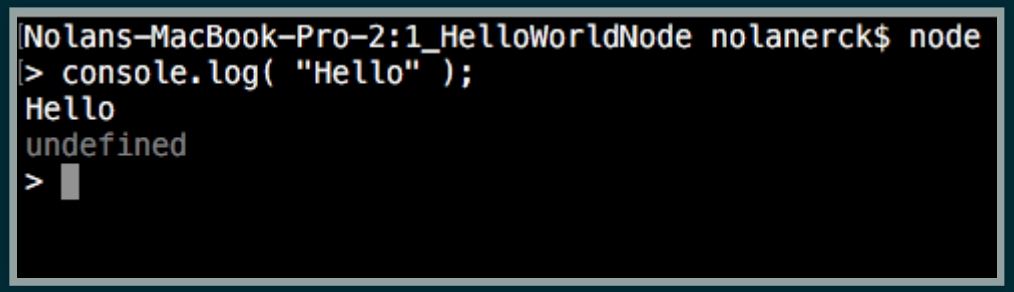
The score:

CommandBox 1, Node 1

The Command Line...

Launching Node.js

- Open a command prompt, type "node"
- This hapens:

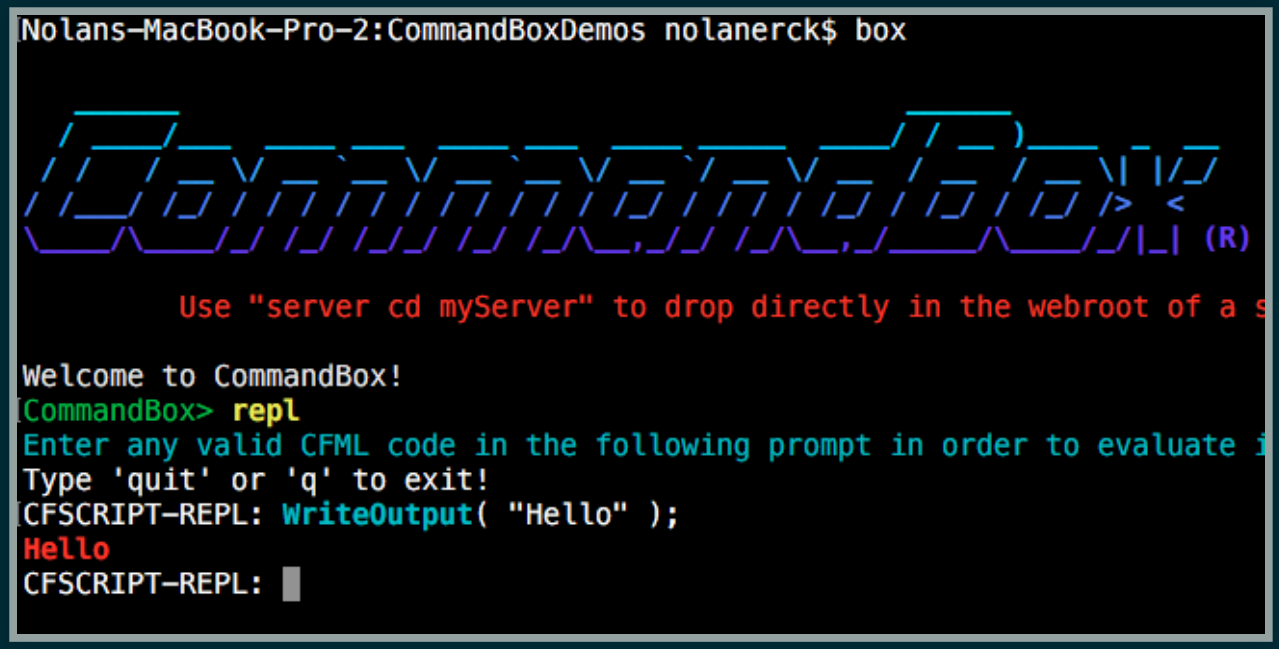


Why did it say "undefined"?

- Not a bug, working as designed
- You see 2 things in the results:
- The *output* of your code
- And the *return value* of your code
- console.log() doesn't return anything, so you get "undefined"

Launching CommandBox

- Open a command prompt, type "box"
- This hapens:

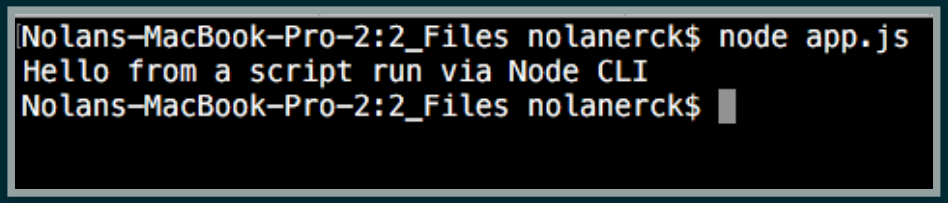


The score:

CommandBox 2, Node 2

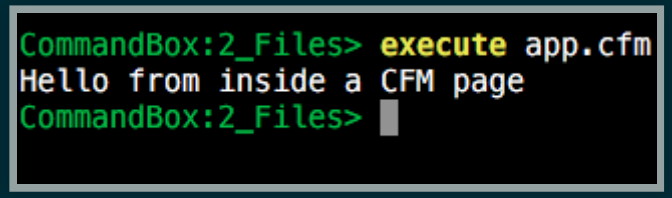
Running files via Node

- Type "node app.js", this happens:



Running files via CommandBox

- Type "execute app.cfm", this happens:

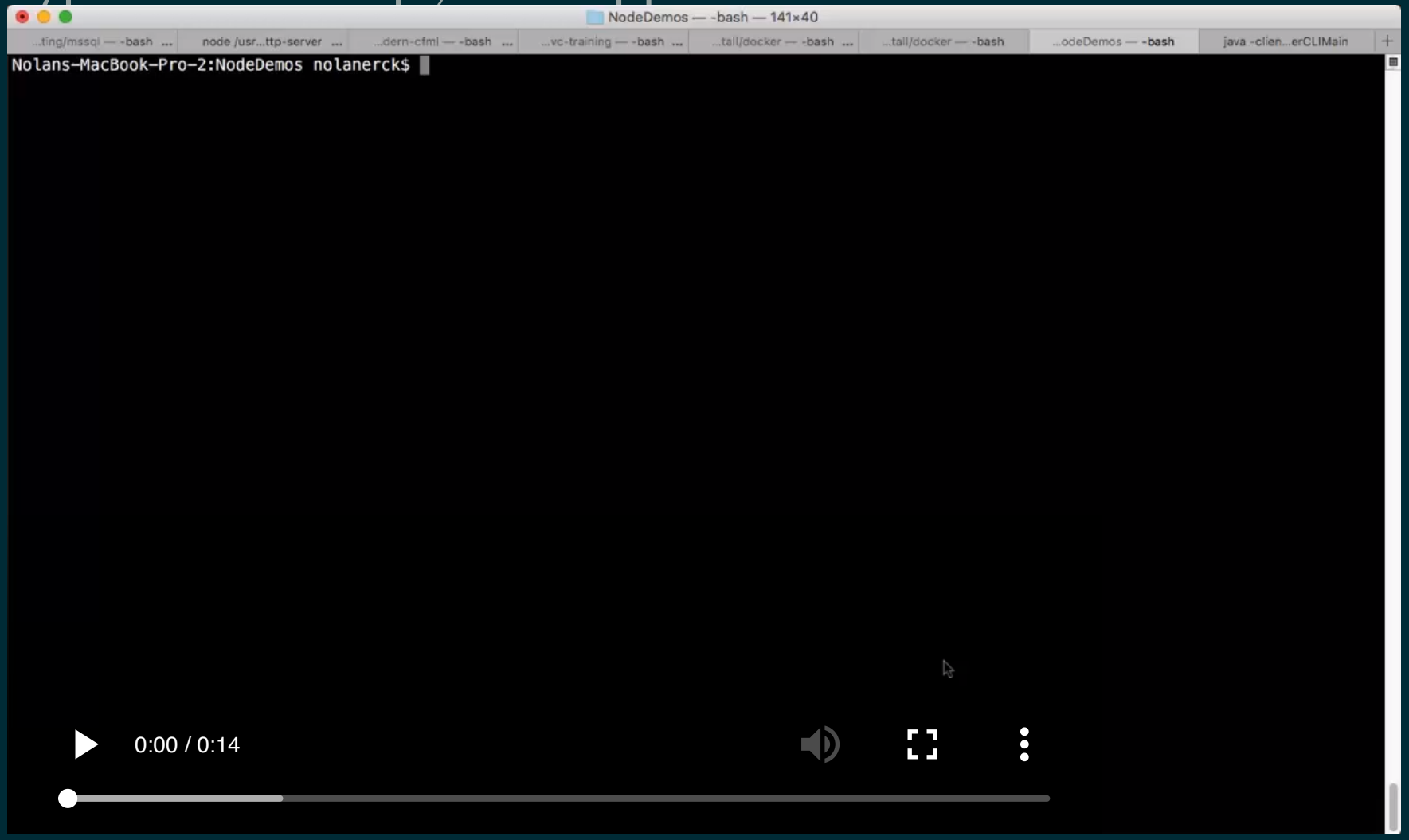


The score:

CommandBox 3, Node 3

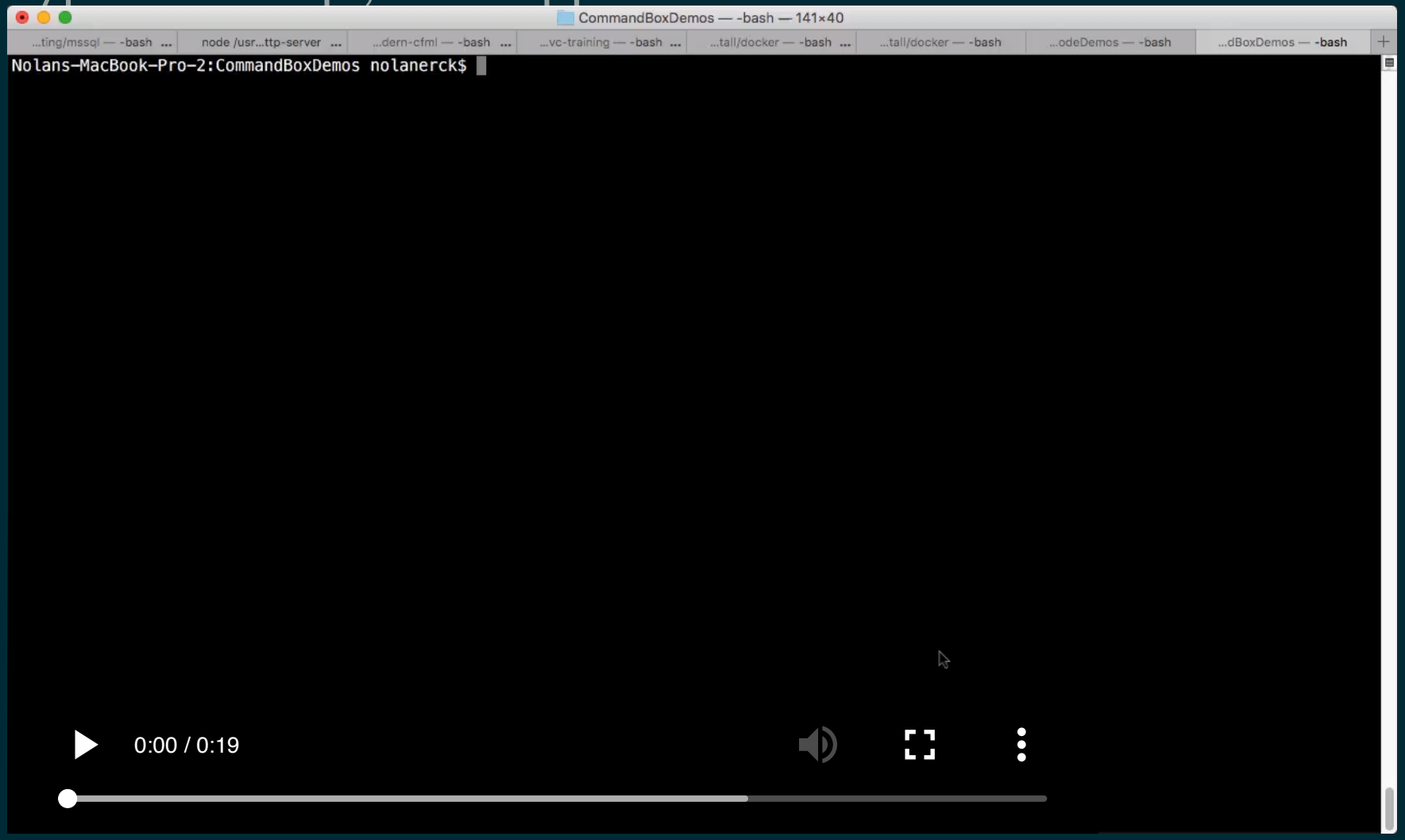
Built-in help from Node.js

- Type "node --help", this happens:



Built-in help from CommandBox

- Type "box help", this happens:



Built-in help from CommandBox

- But wait, there's more!
- CommandBox is broken up into "namespaces"
- Each "namespace" contains many different commands
- Switch to a new namespace to see more detailed help

The score:

CommandBox 4, Node 4

How does Node do it?

- Running a JavaScript application engine on your computer (or server)
- Code is processed thru the engine
- Spins up different services as needed
- Customizable per project via ".json" config files

How does CommandBox do it?

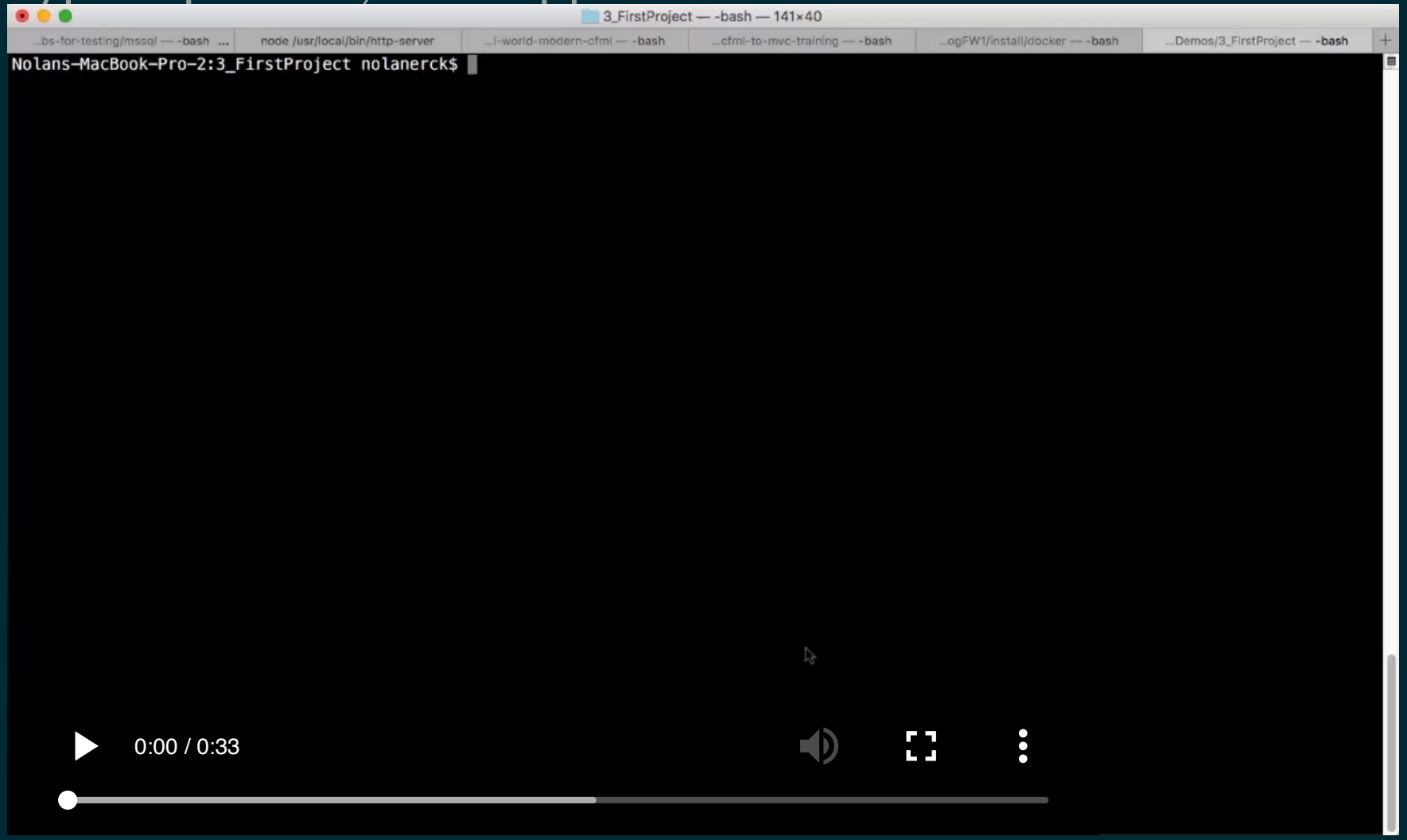
- Running a CFML application engine on your computer (or server)
- Code is processed thru the engine
- Spins up different services as needed (modules, packages)
- Customizable per project via ".json" config files

The score:

CommandBox 5, Node 5

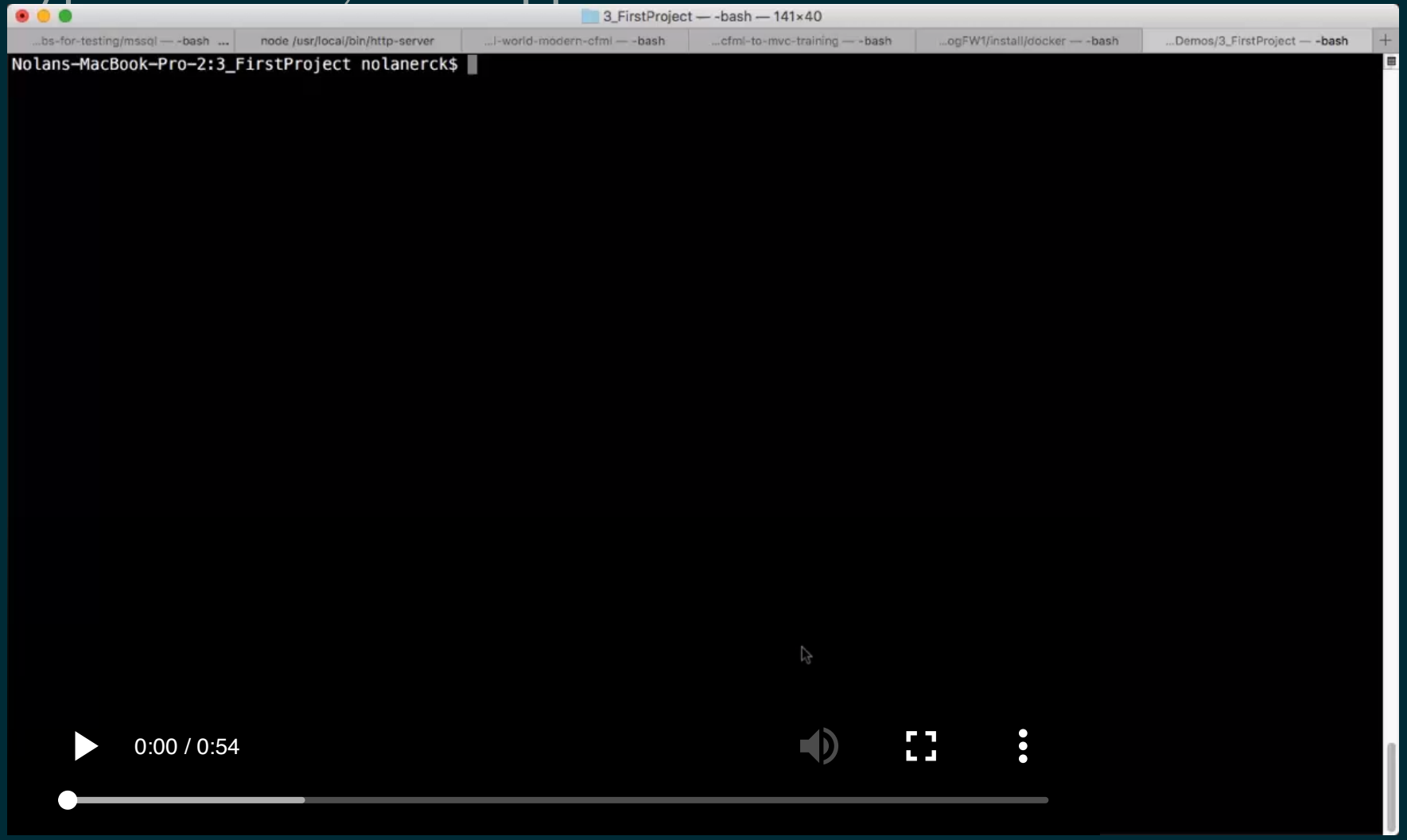
Setting up a Node Project

- Type "npm init", this happens:



Setting up a CommandBox Project

- Type "box init", this happens:



The score:

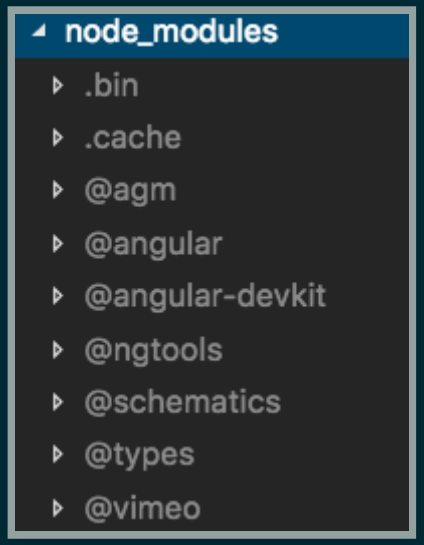
CommandBox 6, Node 6

Dependencies Witih Node

- In package.json, "dependencies" section
  Things your app needs to *run*
- jquery, lodash, Angular, libaries from your team, etc
- "npm install"
- Node goes out to "the registry" and grabs those assets
- Puts them in "node_modules" folder

Dependencies Witih Node

Dependencies Witih Node



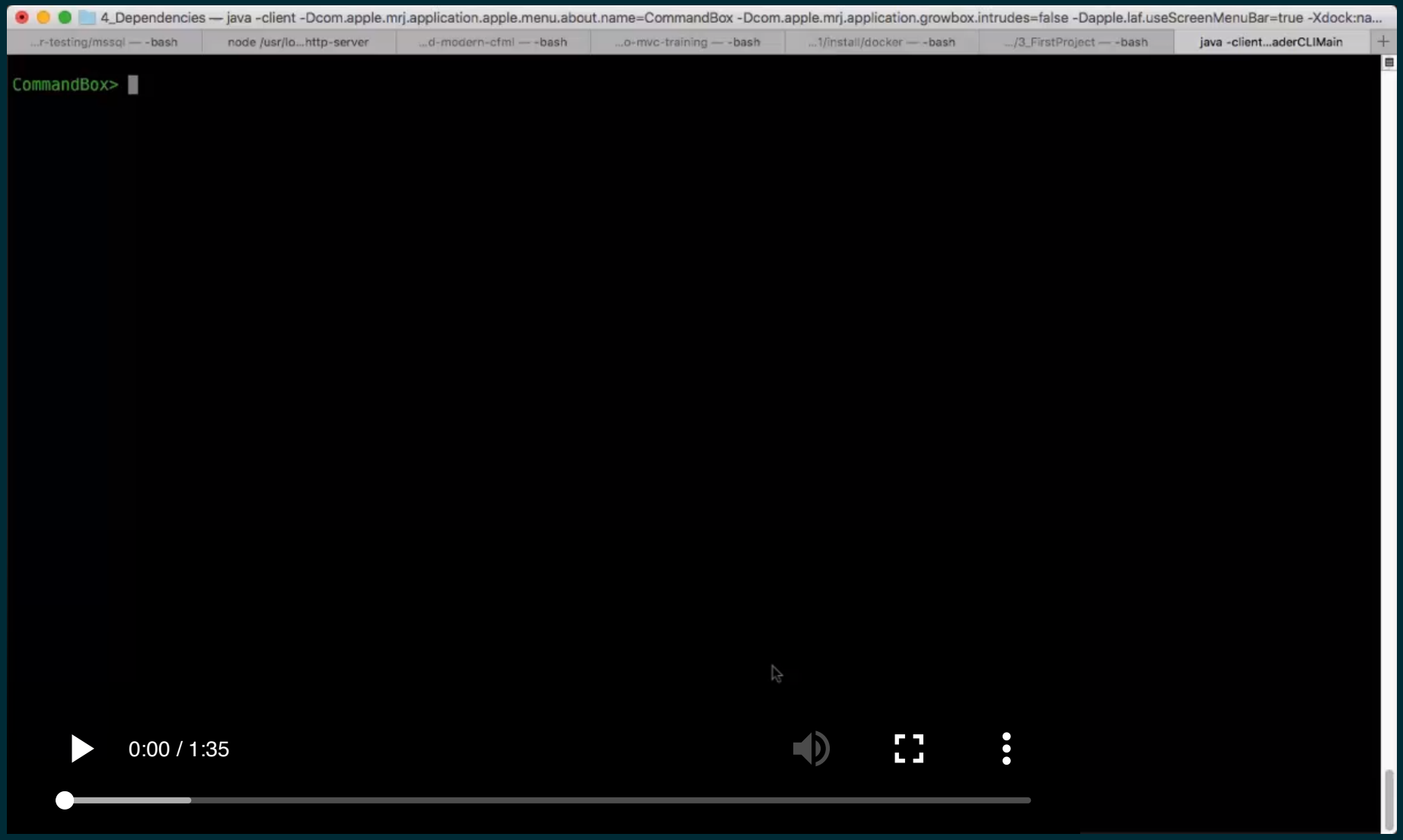
Dependencies Witih CommandBox

- In box.json, "dependencies" section
  Things your app needs to *run*
- jquery, lodash, Angular, libaries from your team, etc
- "box install"
- CommandBox goes out to "the cloud" and grabs those assets
- Puts them in "installPaths" folders

Dependencies Witih CommandBox

```
"dependencies":{
    "Font-Awesome":"git://github.com/FortAwesome/Font-Awesome.git#v4.7.0",
    "bootstrap":"git://github.com/twbs/bootstrap.git#v3.3.7",
    "jquery":"git://github.com/jquery/jquery.git#2.2.4"
},
"installPaths":{
    "Font-Awesome":"assets/Font-Awesome",
    "bootstrap":"assets/bootstrap",
    "jquery":"assets/jquery"
},
```

Dependencies Witih CommandBox



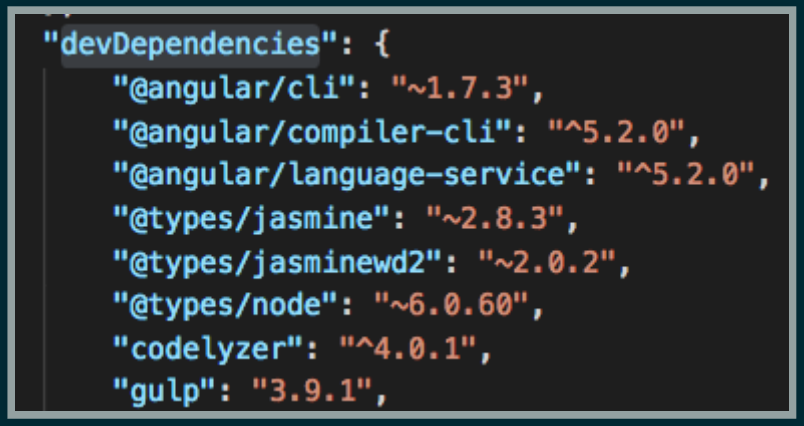
The score:

CommandBox 7, Node 7

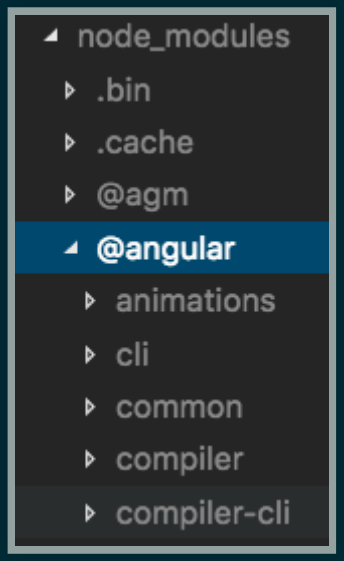
Dev Dependencies With Node

- In package.json, "devDependencies" section
  Things your app needs to *build*
- CLI Tools, Typescript transpiler, Code Analyzer, Linter, etc
- Angujlar CLI, TypeScript, Webpack, etc
- "npm install --dev [thing]"
- Node goes out to "the registry" and grabs those assets
- Puts them in "node_modules" folder

Dev Dependencies Witih Node



Dev Dependencies Witih Node



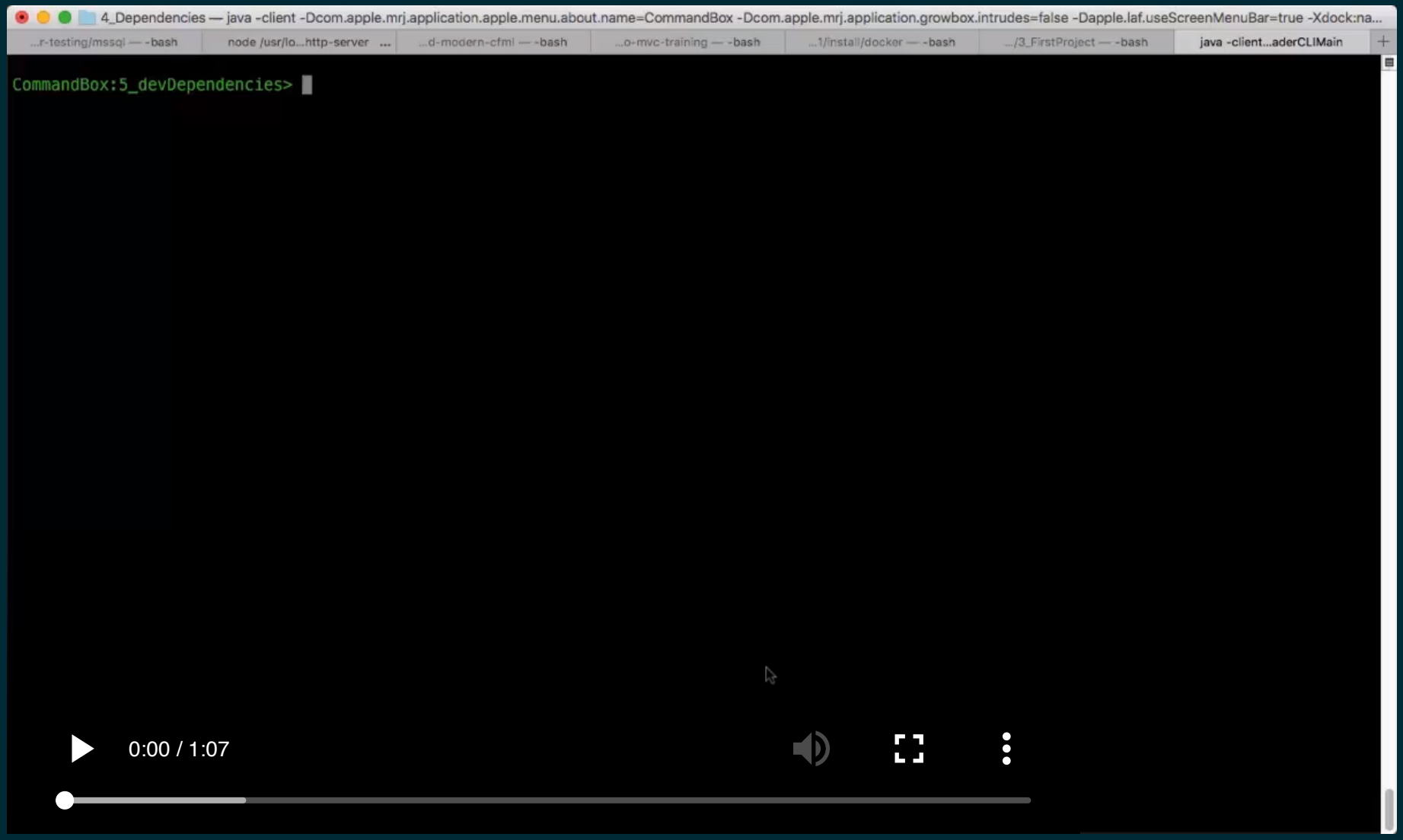
Dev Dependencies Witih CommandBox

- In box.json, "devDependencies" section
  Things your app needs to *build*
- jquery, lodash, Angular, libaries from your team, etc
- "box install --saveDev [thing]"
- CommandBox goes out to "the cloud" and grabs those assets
- Puts them in "installPaths" folders

Dev Dependencies Witih CommandBox

```
{
    "dependencies":{
        "coldbox":"^5.0.0",
        "bootstrap":"git://github.com/twbs/bootstrap.git#v3.3.7",
        "jquery":"git://github.com/jquery/jquery.git#2.2.4",
        "bootstrap-rating":"git://github.com/dreyescat/bootstrap-rating.git#v1.4.0"
    },
    "devDependencies":{
        "testbox":"^2.0.0"
    },
    "installPaths":{
        "coldbox":"coldbox",
        "testbox":"testbox",
        "bootstrap": "assets\\bootstrap",
        "jquery": "assets\\jquery",
        "bootstrap-rating": "assets\\bootstrap-rating"
```

Dev Dependencies Witih CommandBox



The score:

CommandBox 8, Node 8

Node has a Registry: npm

- Magic place in "the cloud" where reusable JavaScript lives
- "npm install [some library]"
- Node talks to "the registry", downloads the lib
- These dependencies live in the "node_modules" folder of your project

How do I add *my* project to npm?

Adding My Project to npm

- Create a package.json file
- Follow a few basic guidlines
- README, semantic version, Author, etc
- More details: https://docs.npmjs.com

CommandBox has a Registry: ForgeBox

- The "npm" of the CF world
- Not just *Box stuff!
- Can install CFWheels, Mura, FW/1, etc
- Any general CFML project can live here

How do I add *my* project to ForgeBox?

Adding My Project to npm

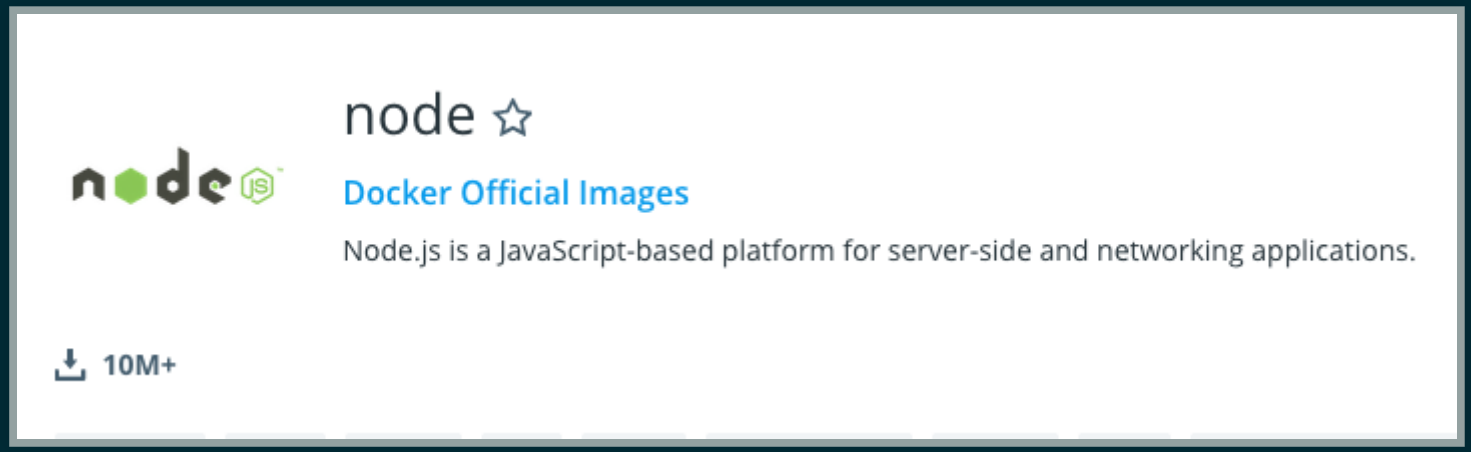
- Create a box.json file
- Follow a few basic guidlines
- README, semantic version, Author, etc
- More details: https://forgebox.io

The score:

CommandBox 9, Node 9

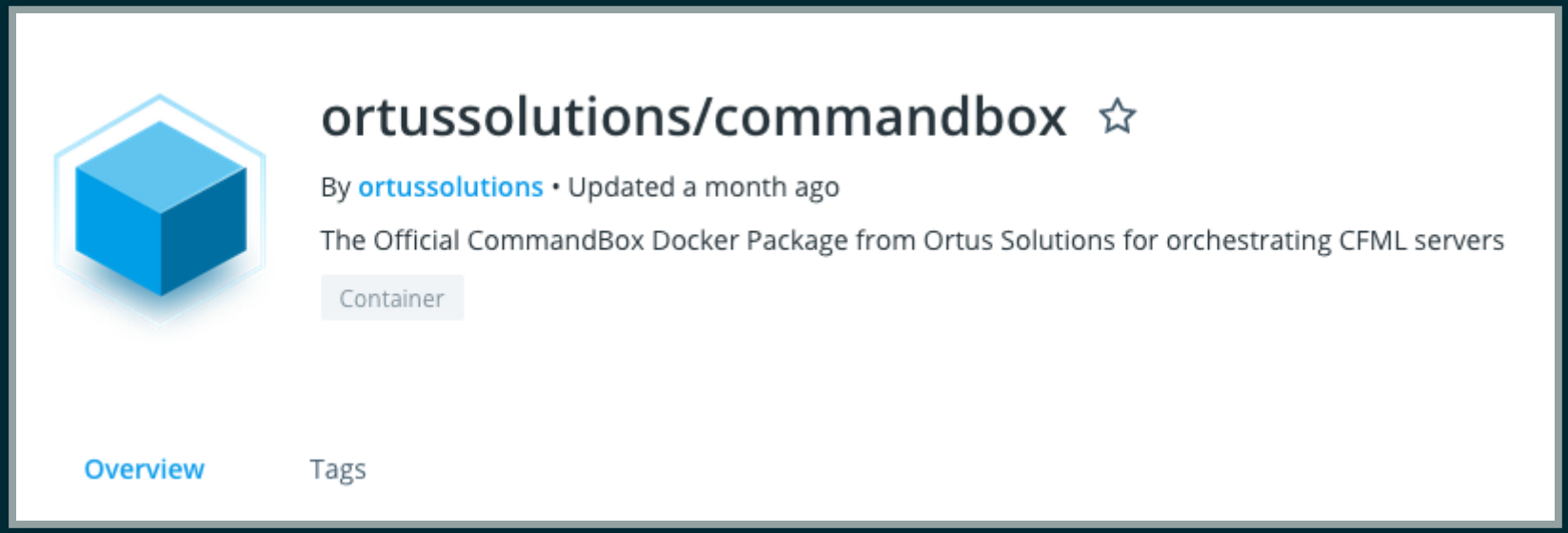
Using Node and Docker

- hub.docker.com/_/node



Using CommandBox and Docker

- Official Docker images from Ortus Solutions
- hub.docker.com/r/ortussolutions/commandbox/



The score:

CommandBox 10, Node 10

Using Node to Make Games

- Tons of resources
- Many game engines support JavaScript
- Can get as simple or advanced as you like

Using CommandBox to Make Games

Using both JavaScript *and* CFML for Games!

- box snake
  Vintage gaming at its finest!
- Minh Vo's preso on React at Gov't Summit
- draftstudios.com
- Giancarlo Gomez's preso on WebSockets:
  "Refreshing Your UI: Modern Uses for WebSockets"

The score:

CommandBox 11, Node 11

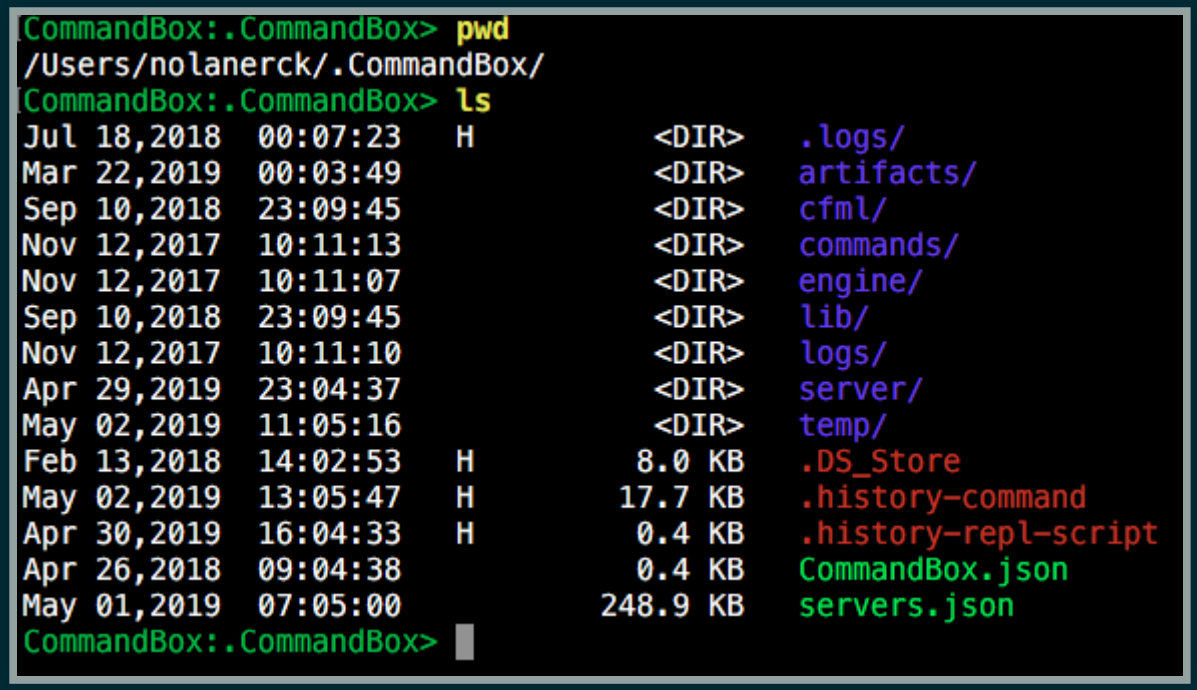
Contributing to Node.js

- Main engine is written in C++, not JavaScript
- Add-ons can be JavaScript but not the core
- e.g the Angular CLI, create-react-app

Contributing to CommandBox

- 90% of the core is CFML
- Remaining 10% is Java
- Installing CommandBox also gives you the source code!

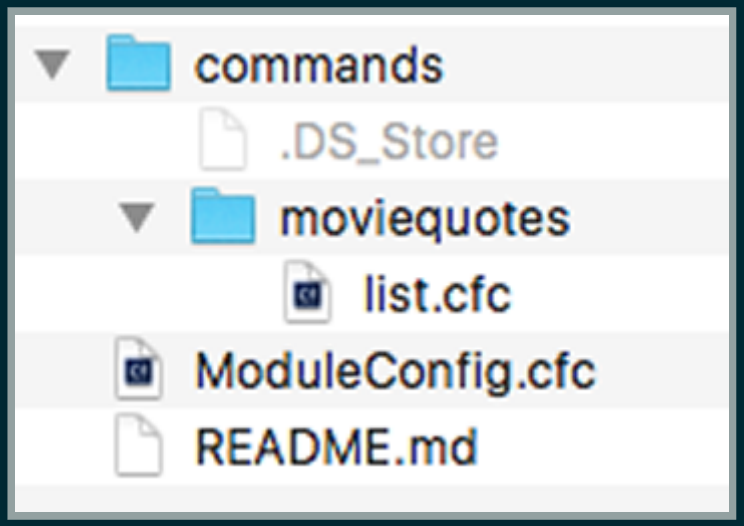
Contributing to CommandBox



But my custom commands have to be ColdBox apps, right?

Nope!

My First CommandBox Command



ModuleConfig.cfc

```
component {

    this.name = "moviequotes";
    this.author = "";
    this.webUrl = "https://github.com/nolanerck/moviequotes";

    function configure()
    {
        //
    }
}
```

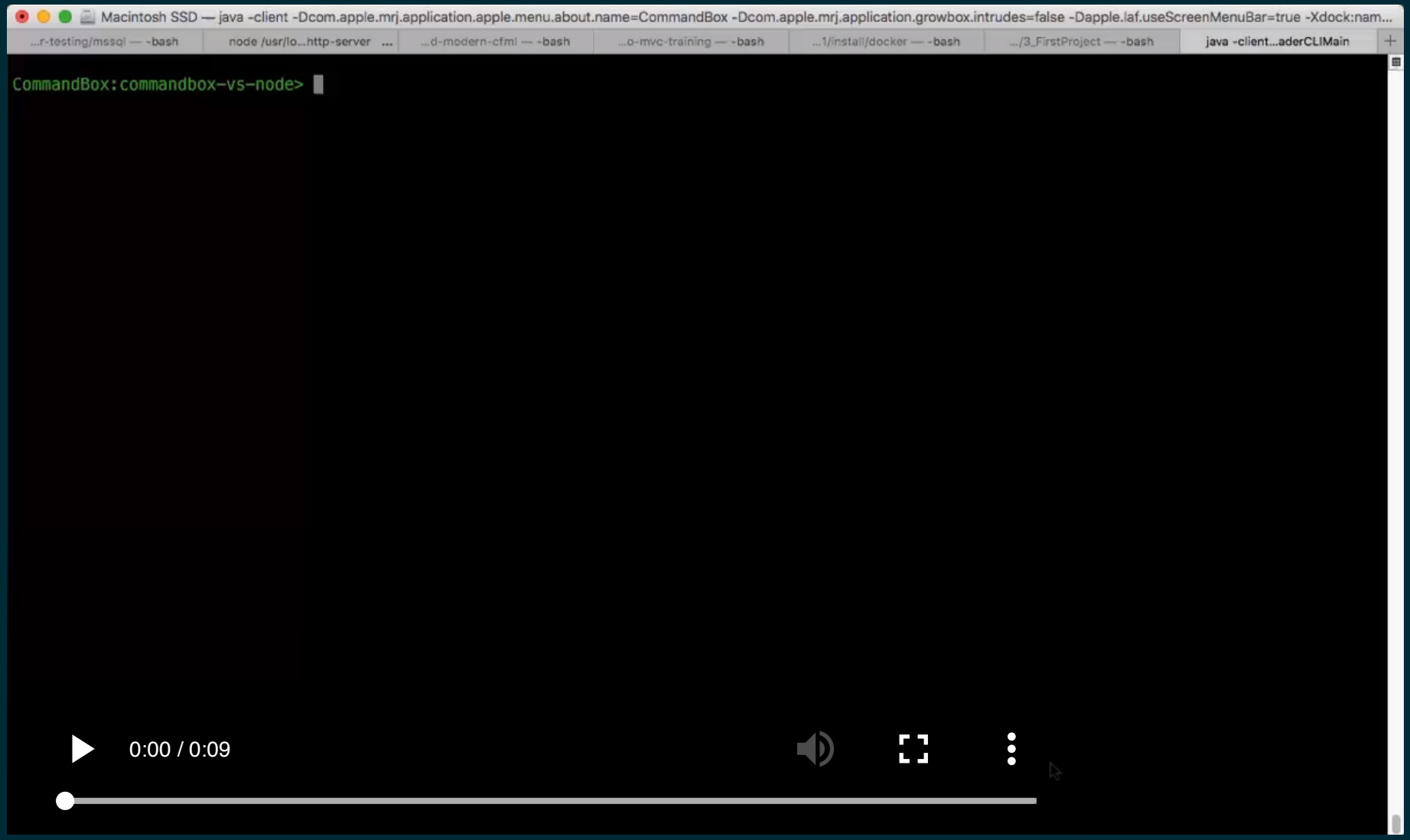
list.cfc

```
component
{
    function run()
    {
        var quotes = [ "I do not think that word means what you think it means.",
                       "Does Barry Manilow know that you raid his wardrobe?",
                       "So, what would you little maniacs like to do first?",
                       "My name is Warner Brandis. My voice is my passport. Verify me.",
                       "All of my filth is in alpabetical order."
        ];

        var num = RandRange( 1, 5 );

        return quotes[ num ];
    }
}
```

MovieQuotes In Action!



The score:

CommandBox 12, Node 12

Okay so what's the catch?

There isn't one

- CommandBox is...
- Free, open source, well supported
- Supports all CFML engines
- Lucee and Railo
- Adobe CF as far back as version 9

CommandBox Really is a Game-Changer

- You *can* do all the cool things that Node/JavaScript developers do

So which do I have installed for development?
Node or CommandBox?

Both

- They serve different purposes

Uses for Node.js

- It's ubiquitous with modern front-end development
- Front-end tooling requires Node
- Angular, Vue, React, PhoneGap, Grunt, Gulp, Stylus, SASS, SCSS, LESS, WebPack, Babel, TypeScript, etc
- This is a "given" nowadays

Uses for CommandBox

- This is *the* way to tell *modern* CFML developers from legacy programmers
- Spinning up Dev environments, testing everything, containerization, onboarding new team members
- Managing production web servers
- Building CLI tools for development AND production servers!

Other Resources

- South of Shasta - onsite and remote training
- nodejs.org
- docs.npmjs.com
- commandbox.ortusbooks.com
- Ortus Solutions
- Brad Wood's Blog
- Talk to people at the conference!

Learning in 30 minutes a day.

Questions? Comments?

- southofshasta.com
- nolan@southofshasta.com
- Twitter: @southofshasta
- Github: nolanerck
- Slides and code: github.com/nolanerck/commandbox-vs-node

Thanks!