

The Salesforce logo, consisting of the word "salesforce" in white lowercase letters inside a blue cloud-like shape.

salesforce

# Build and Release a CLI Plug-In LIVE!

#SalesforceCLI



# Speakers

Who are we?



**Anastasia Shpurik**  
[github.com/ashpurik](https://github.com/ashpurik)



**Thomas Dvornik**  
[@amphro](https://twitter.com/amphro)





# Forward-Looking Statement

## Statement under the Private Securities Litigation Reform Act of 1995

This presentation may contain forward-looking statements that involve risks, uncertainties, and assumptions. If any such uncertainties materialize or if any of the assumptions proves incorrect, the results of salesforce.com, inc. could differ materially from the results expressed or implied by the forward-looking statements we make. All statements other than statements of historical fact could be deemed forward-looking, including any projections of product or service availability, subscriber growth, earnings, revenues, or other financial items and any statements regarding strategies or plans of management for future operations, statements of belief, any statements concerning new, planned, or upgraded services or technology developments and customer contracts or use of our services.

The risks and uncertainties referred to above include – but are not limited to – risks associated with developing and delivering new functionality for our service, new products and services, our new business model, our past operating losses, possible fluctuations in our operating results and rate of growth, interruptions or delays in our Web hosting, breach of our security measures, the outcome of any litigation, risks associated with completed and any possible mergers and acquisitions, the immature market in which we operate, our relatively limited operating history, our ability to expand, retain, and motivate our employees and manage our growth, new releases of our service and successful customer deployment, our limited history reselling non-salesforce.com products, and utilization and selling to larger enterprise customers. Further information on potential factors that could effect the financial results of salesforce.com, inc. is included in our annual report on Form 10-K for the most recent fiscal year and in our quarterly report on Form 10-Q for the most recent fiscal quarter. These documents and others containing important disclosures are available on the SEC Filings section of the Investor Information section of our Web site.

Any unreleased services or features referenced in this or other presentations, press releases or public statements are not currently available and may not be delivered on time or at all. Customers who purchase our services should make the purchase decisions based upon features that are currently available. Salesforce.com, inc. assumes no obligation and does not intend to update these forward-looking statements.



# Agenda

## What Is This Talk About?

### Salesforce CLI & Plug-Ins

- What's new?

### Plug-In Architecture

- How does it all fit together?

### A Plug-In Use Case

- Look at all stages of the development flow

### Demo

- The LIVE part (why we are all here)

### Recap & Resources

### Roadmap



# Salesforce CLI



The Salesforce CLI is a powerful command-line interface that simplifies development and build automation when working with your Salesforce org.

- The CLI is not just for scratch orgs
  - Metadata API deploy/retrieve
  - Apex execute and tests
  - Data describes and queries
  - Anything the Ant Migration Tool can do!
  - The list goes on!
- Now with Winter '19 you can `source:deploy` and `source:retrieve`
  - Uses source format
  - Works on non-scratch orgs!
  - You don't have to worry about package.xml (although you can supply one)

```
+ pipelines git:(master) hub:(tdx) scratch:(scratch)
-> sfdx force:config:list
== Config ==
NAME                VALUE      LOCATION
defaultdevhubusername tdx        Global
defaultusername      scratch    Global
+ pipelines git:(master) hub:(tdx) scratch:(scratch)
-> sfdx force:org:list
== Orgs ==
ALIAS      USERNAME                ORG ID                CONNECTED STATUS
HubOrg     tdvornik@tdxdemo.org    00Df40000001Dh0EAU    Connected

ALIAS      SCRATCH ORG NAME      USERNAME                ORG ID                EXPIRATION DATE
(U) scratch Company     test-nlvvn1nxf3n@example.com 00DL0000002BPNWMA4    2018-03-29

+ pipelines git:(master) hub:(tdx) scratch:(scratch)
-> sfdx force:data:soql:query -q "SELECT Name FROM Account"
NAME
Marketing Row
Service Inc.
Total number of records retrieved: 2.
```

# What Are Plug-Ins?

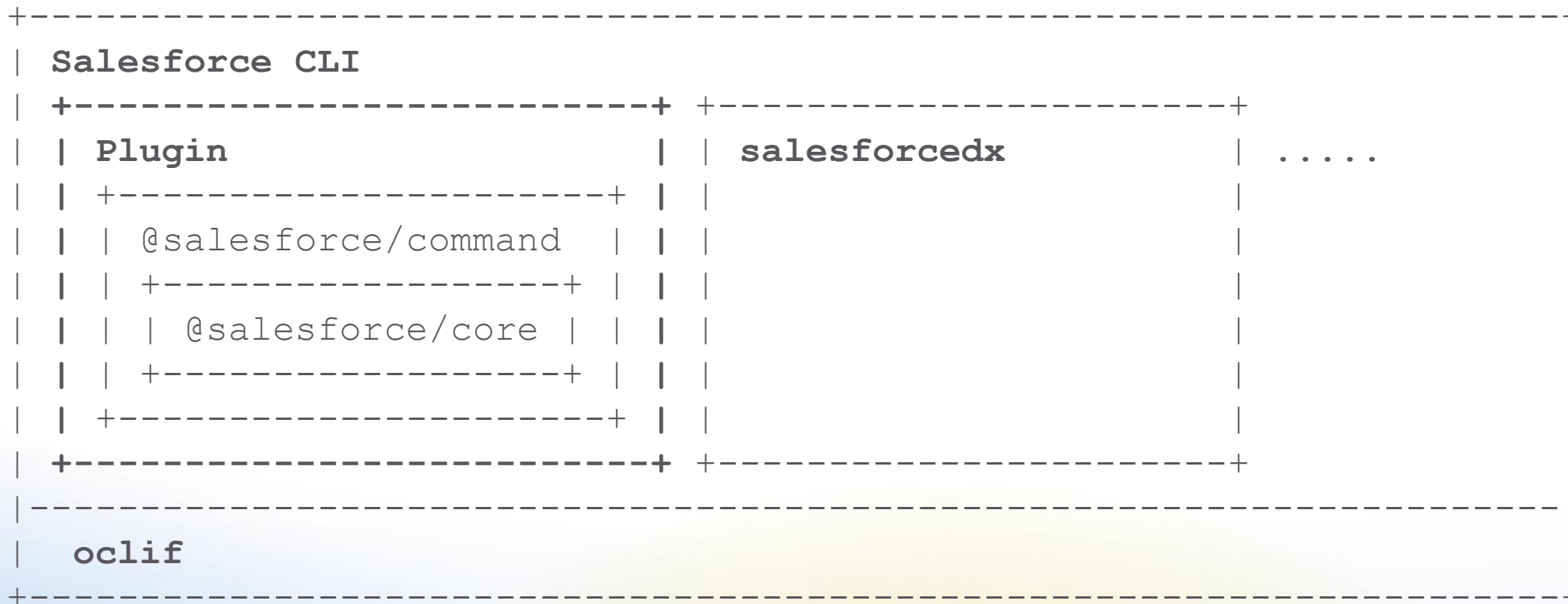


Extend the functionality of the CLI with custom commands!

- Control what you need
- Focus on building features, not the framework
  - Don't worry about terminal input/output
  - Don't worry about auth
  - ... list goes on, as we will see ...
- and... it is super easy!



# How It All Fits Together

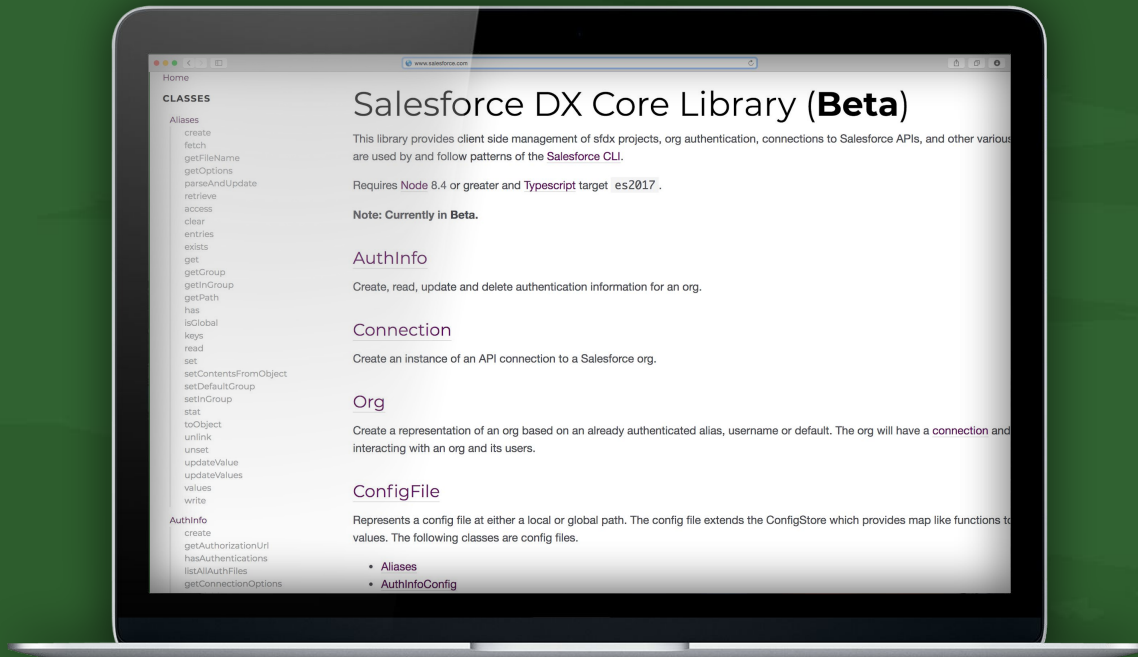


# oclif & Salesforce Core Libraries

Build Fast!



Powerful CLI framework and libraries to get you off the ground in no time!

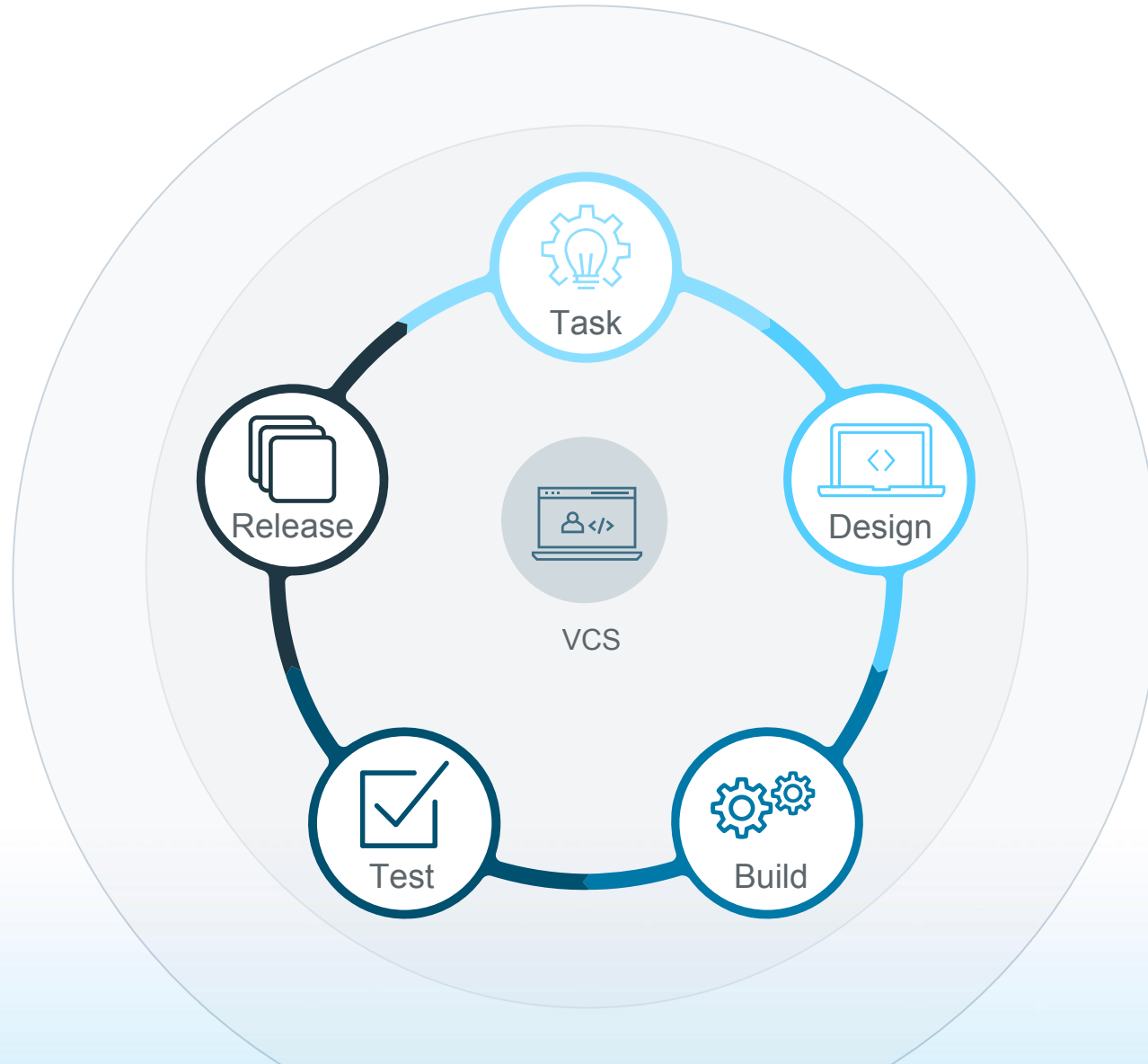


- Consistent UX
- Authentication Management
- Org Connection (jsforce)
- Project Management
- Configuration Settings
- Client-Side Logging

<https://developer.salesforce.com/media/salesforce-cli/docs/@salesforce/core/index.html>



# The Plug-In Journey



# Task



## The Use Case

What are we trying to do on the CLI?

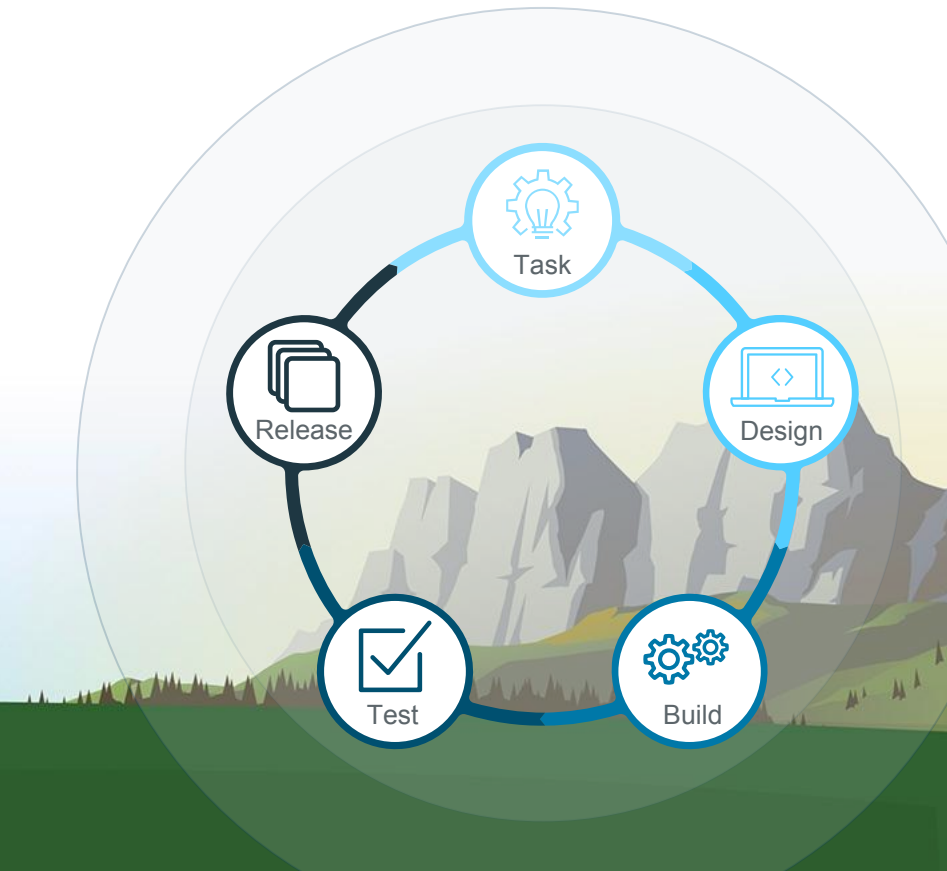
- Usually a common, repeated task
- Not already in the CLI

Is there a CLI plug-in that can do that for us?

- Search plug-ins on npm using the “sfdx-plugin” keyword
- Or, it can be specific to your use case and objects!

Example:

- A custom Salesforce app that listens to hashtags on Chatter and sends platform events to perform some sort of action
- As we were building this, we wanted to watch test events to help debug! But we had to use Workbench =(



# Design



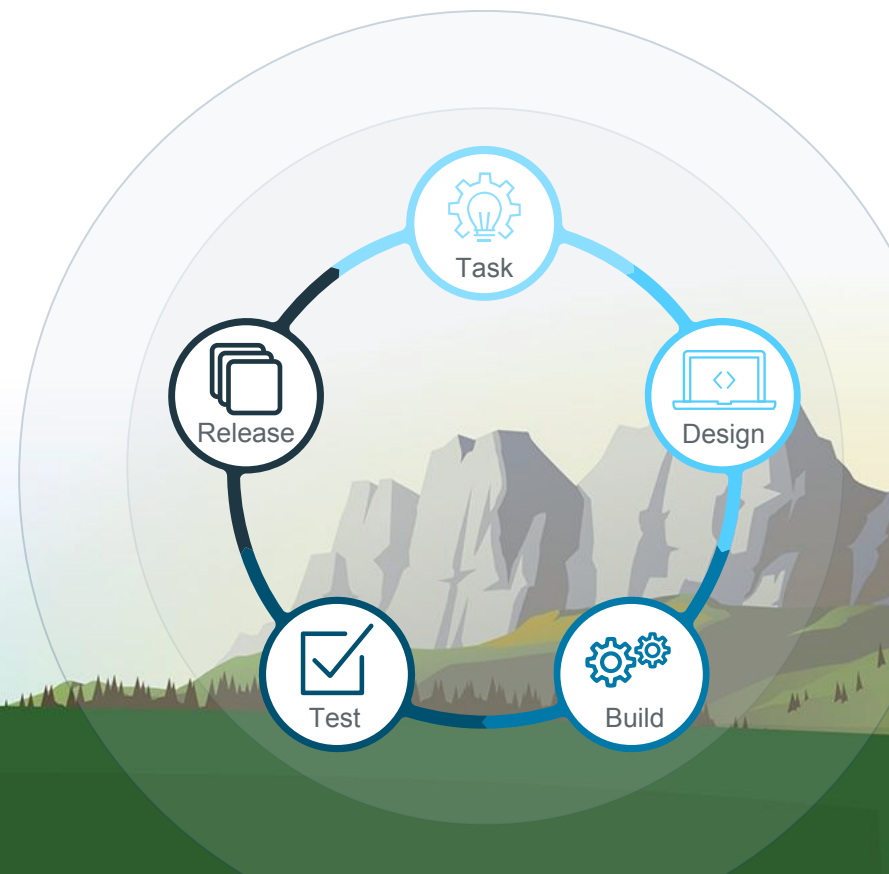
## The Commands

Think about naming!

- Topics, commands, arguments, and flags
- This sounds easy, but naming is hard!
- Keep command names verbs and flag names all lowercase
- Look at `force` commands for guidance!
- We are working on publishing guidelines

Can others use this?

- Keep it generic! If possible
- Should it be one command or many?
- Will other commands build on top of it? `--json`



# Build

## The Code

Use Salesforce Plug-In Generator

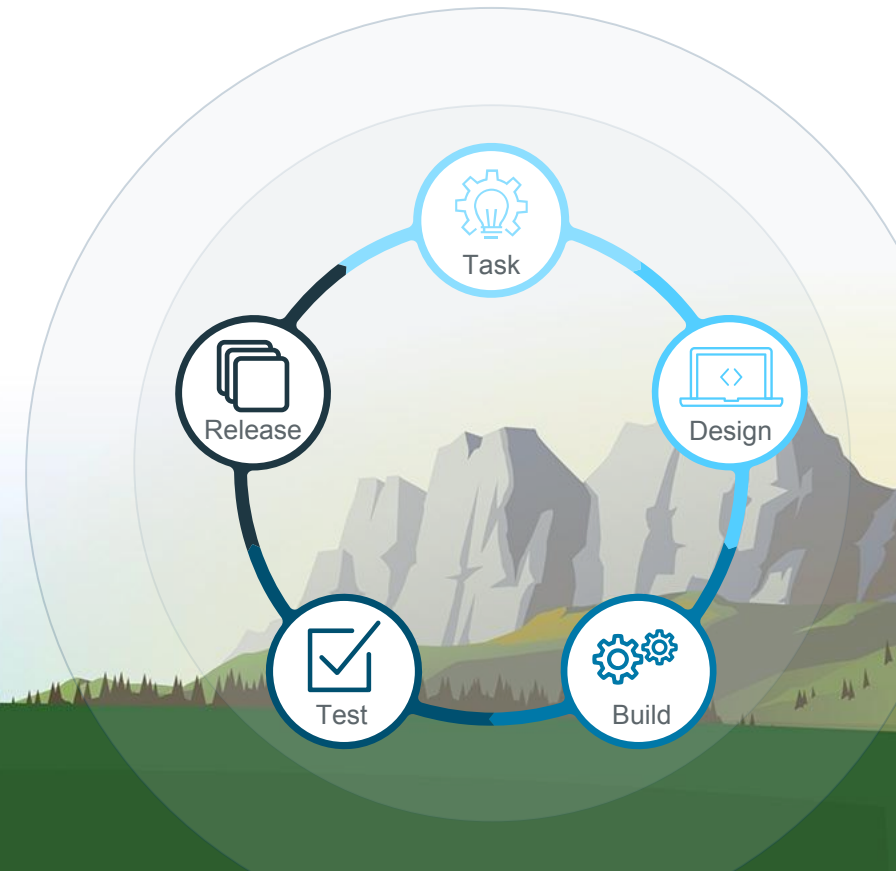
Use an editor with TypeScript support

- e.g. autocompletion and docs
- We recommend VS Code

Explore the core libraries to see what capabilities it adds!

- We will be using the StreamingClient

Take advantage of the debugger rather than console statements





# Test

## The Code

Don't save this to the end!

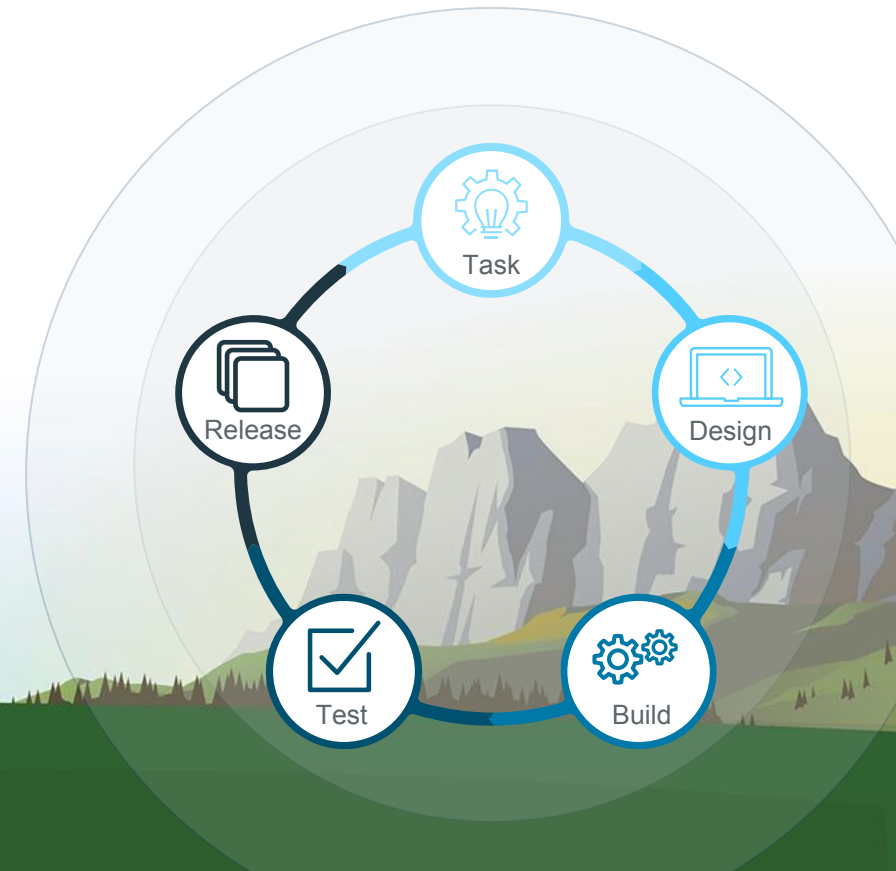
Break out code into files that the command calls

- This helps with mocking out calls, especially calls to core/command
- Try not to make API or file system calls in unit tests

## Use CI

- The plug-in generator gives you a starting point of CircleCI and AppVeyor

salesforce



# Release

## The Plug-In

Plug-ins live on npm

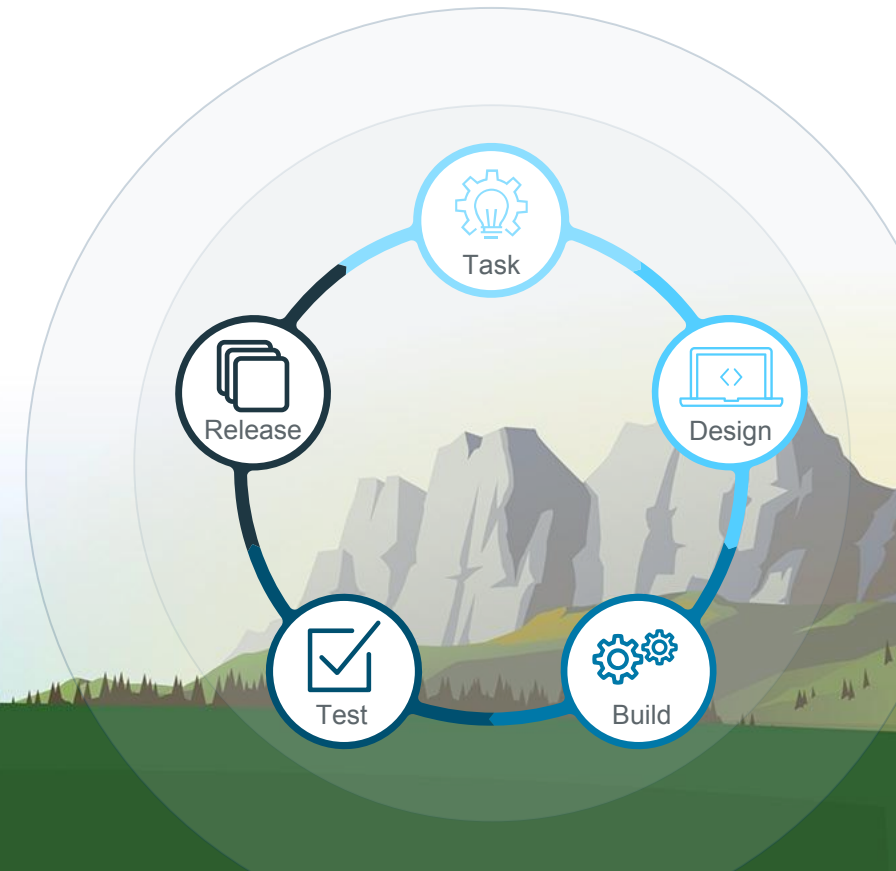
- This can be a public or private (like Verdaccio)

Easy to deploy with `npm publish`

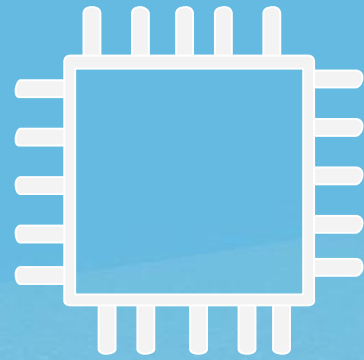
- Useful to keep the plug-in name in your scope
  - e.g. `sfdx plugins:install @amphro/my-plugin`

Open source

- Allows others to contribute or fork
- Gives open visibility to what the plug-in is doing



UP NEXT



DEMO

# Recap

## What Did We Do?

We built a CLI plug-in to listen for platform events and streaming topics!

- Used Salesforce Plug-In Generator
- Used the core libraries!
- Added tests

We used Git for a VCS

We published to npm

We installed in Salesforce CLI



## Resources

<https://github.com/ashpurik/hashtag-listener>

<https://github.com/amphro/event-listener>

<https://github.com/forcedotcom/sfdx-plugin-generate>

<https://sfdc.co/sfdx-core-docs>



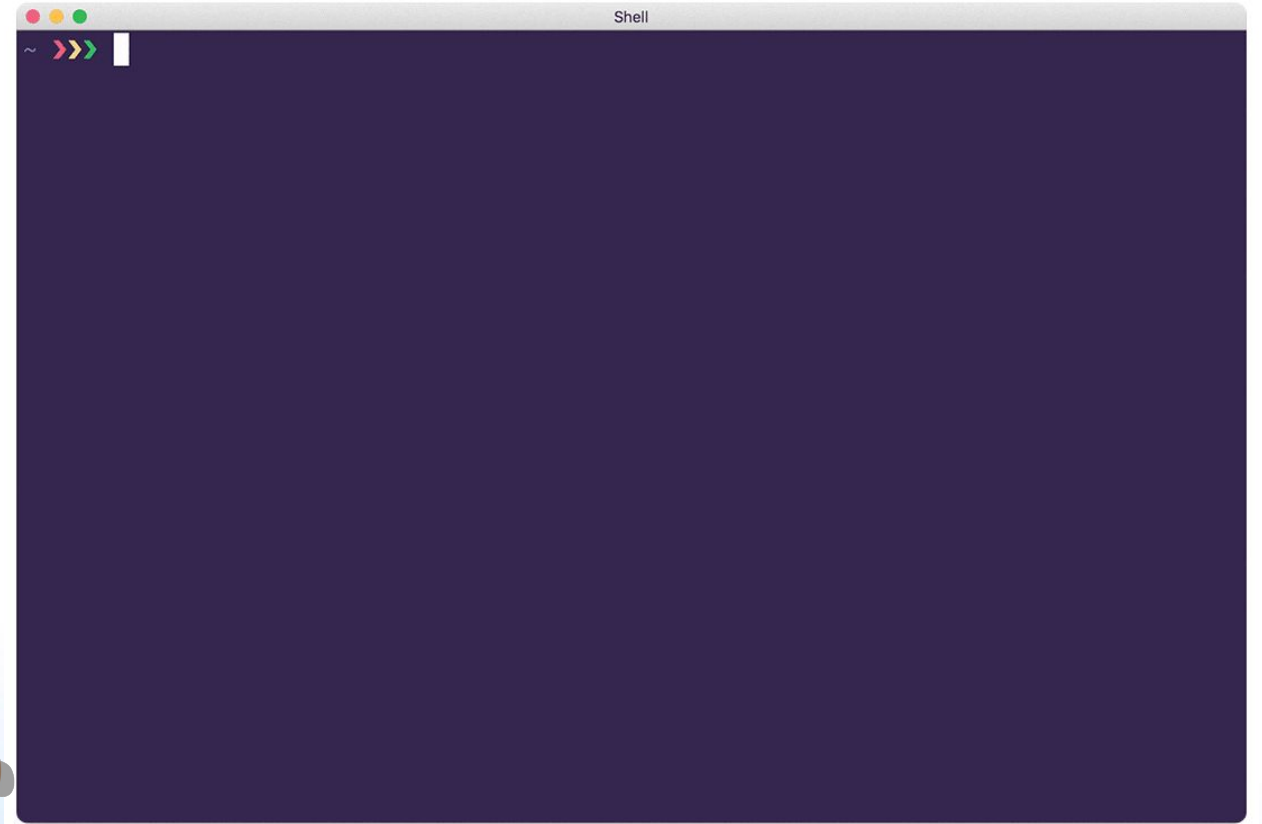
```
sfdx plugins:install @amphro/streamer
```



# Where Are We Going?

## On the Roadmap

1. Open-sourcing core/command
2. Autocompletion (GIF to the right)
3. Community signed plug-ins
4. Plug-in marketplace
5. Plug-in discoverability
6. Command hooks



# Learn More About Salesforce DX

Attend a Related Session or Visit Our Booth

Wednesday 3:00–3:20 pm

**Banish the Bugs with the Apex Replay Debugger**  
Moscone West – Developer Theater

Wednesday 4:30–5:10 pm

**Apply the Salesforce CLI to Everyday Problems**  
Moscone West – Room 2009

Wednesday 5:00–5:40 pm

**Modern App Dev: Modular Development Strategies**  
Moscone West – Room 2011

Thursday 1:15–1:35 pm

**Building Powerful Tooling for IDEs Through Language Servers**  
Moscone West – Frontier Theater

Thursday 4:00–4:20 pm

**Moving to Visual Studio Code for Force.com IDE Developers**  
Moscone West – Canyon Theater

Salesforce for  
Developers Keynote



Build the Future of Apps

Thursday | 1:00 pm  
Moscone West – Level 3

salesforce

See a  
**Demo**  
or ask a  
question  
at the  
Salesforce DX  
Booth

Explore [Lightning Rapids](#) to learn  
about app dev on the Lightning  
Platform

Visit the [IT Observatory](#) for  
Workshops and Astro's VR  
Adventure!



THANK Y☁U

