Designing an Automation Framework for Splunk

How to build source control, automate tasks, and implement continuous delivery

Alex Cain | Senior Product Manager, Data Availability

October 2018 | Rev 2

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

splunk> .conf18

ALEX CAIN

Senior Product Manager,
Data Availability

splunk>

splunk> .conf18

4 Key Takeaways

How to build a successful automation platform with Splunk

Managing Splunk in a automation friendly manner

Automating Splunk from low-level functions to complex tasks

Using an automation management platform to enable CI/CD

Building Splunk alerts to add self-service and self-healing

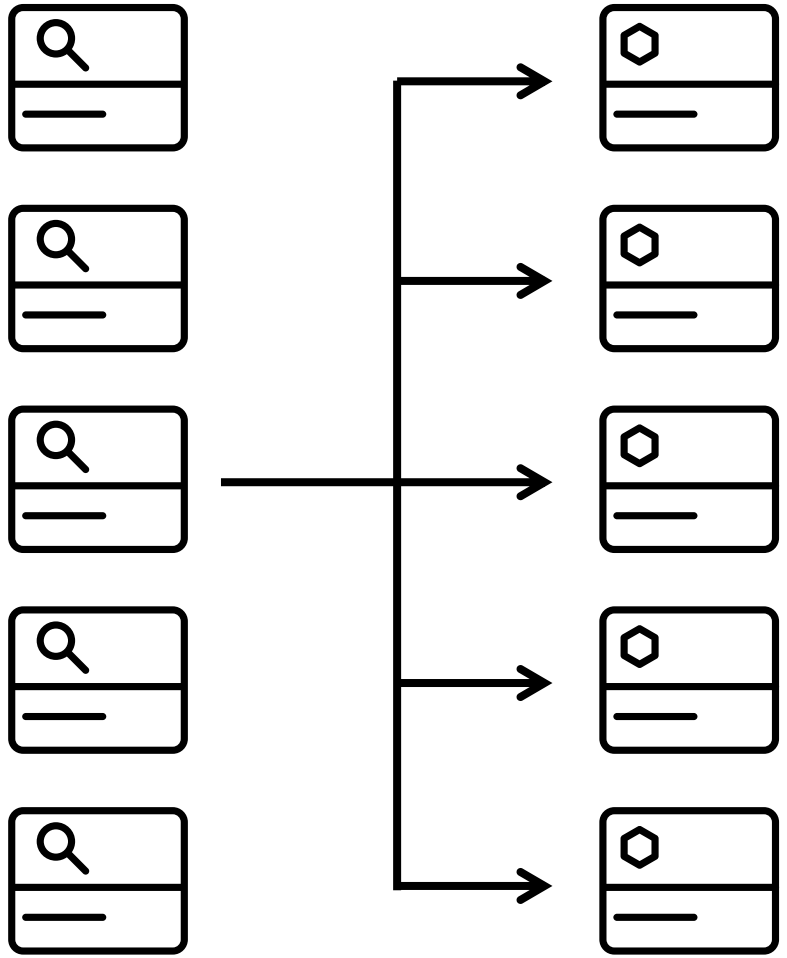
splunk> .conf18

A long Time ago…

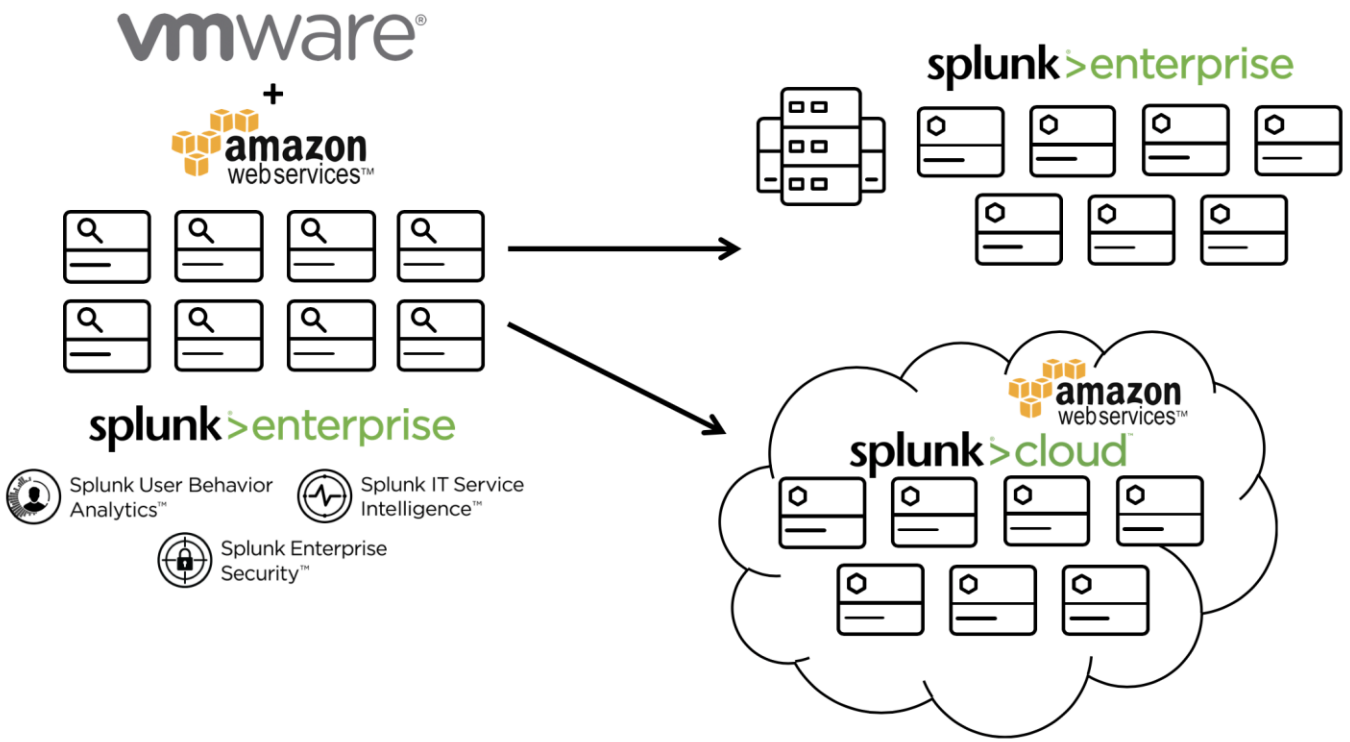
Where we were

Our deployment 4 years ago

- Small Deployment
  - 5 search heads
  - 5 indexers
  - ~500 GB per day
  - No premium use-cases
- Fragmented deployments across enterprise
  - At least 6 separate larger deployments
  - Countless small single use-case deployments



Where we are now



Our deployment today

- Medium / Large Hybrid Deployment
  - 45+ search heads
  - 35 Indexers
  - ~ 4 TB per day
  - ES and ITSI deployed
  - Hybrid Search
- Consolidated deployments across enterprise
  - Only a couple separated deployments for compliance reasons
- Future Roadmap
  - Planning large migration to Splunk Cloud

Challenges we face today

How we started down the automation path

- How do we manage applications across our deployment?
- How do we scale our deployment as we continue to grow?
- How do we make sure configurations are consistent across our infrastructure?
- How do we migrate infrastructure as it ages?
- How do we support enterprise wide usage?
- How do we support users with varying Splunk knowledge and skill levels?
- How do we run both production infrastructure and development?

"Rome ne s'est pas faite en un jour”

Rome was not built in a day

The Journey Down Automation

The key points build on each

Source Control

Control code quality to ensure deployment of known good configurations

Simple tasks

Script the basics of interacting with your deployment

Complex tasks

Combine the basic blocks and add logic to complete difficult operations

Continuous Delivery

Setup a management platform to schedule administrative tasks

Self-Healing

Hook Splunk alerts back into you automation platform to remediate issues

4 Key Takeaways

How to build a successful automation platform with Splunk

Managing Splunk in a automation friendly manner

Automating Splunk from low-level functions to complex tasks

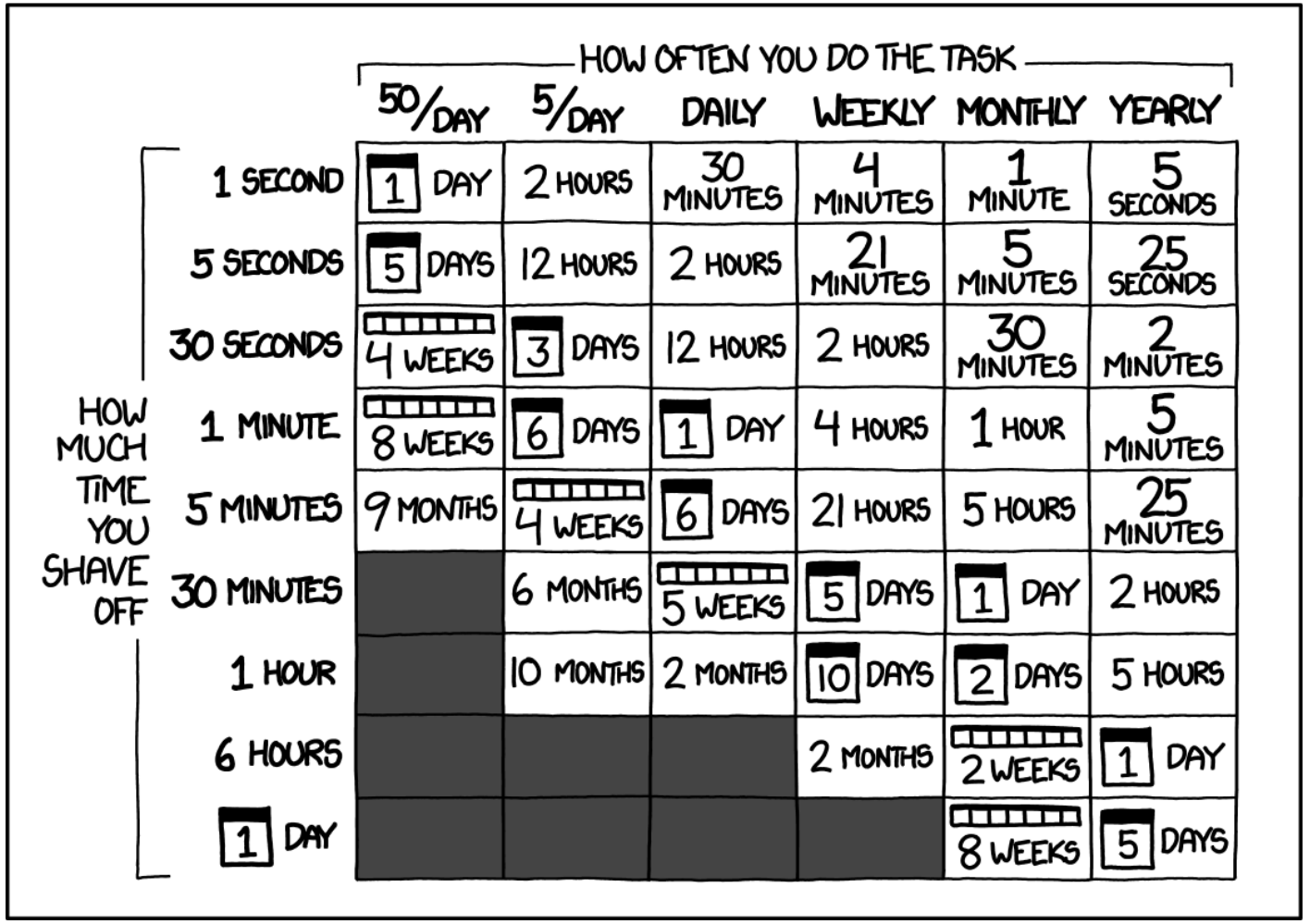
Using an automation management platform to enable CI/CD

Building Splunk alerts to add self-service and self-healing

splunk> .conf18

Is It Worth It?

HOW LONG CAN YOU WORK ON MAKING A ROUTINE TASK MORE EFFICIENT BEFORE YOU'RE SPENDING MORE TIME THAN YOU SAVE?
(ACROSS FIVE YEARS)



splunk> .conf18

Managing Splunk in a automation friendly manner

GIT your apps and configs under control

splunk> .conf18

How do you manage configuration?

Configuration journey

GitLab

Bitbucket

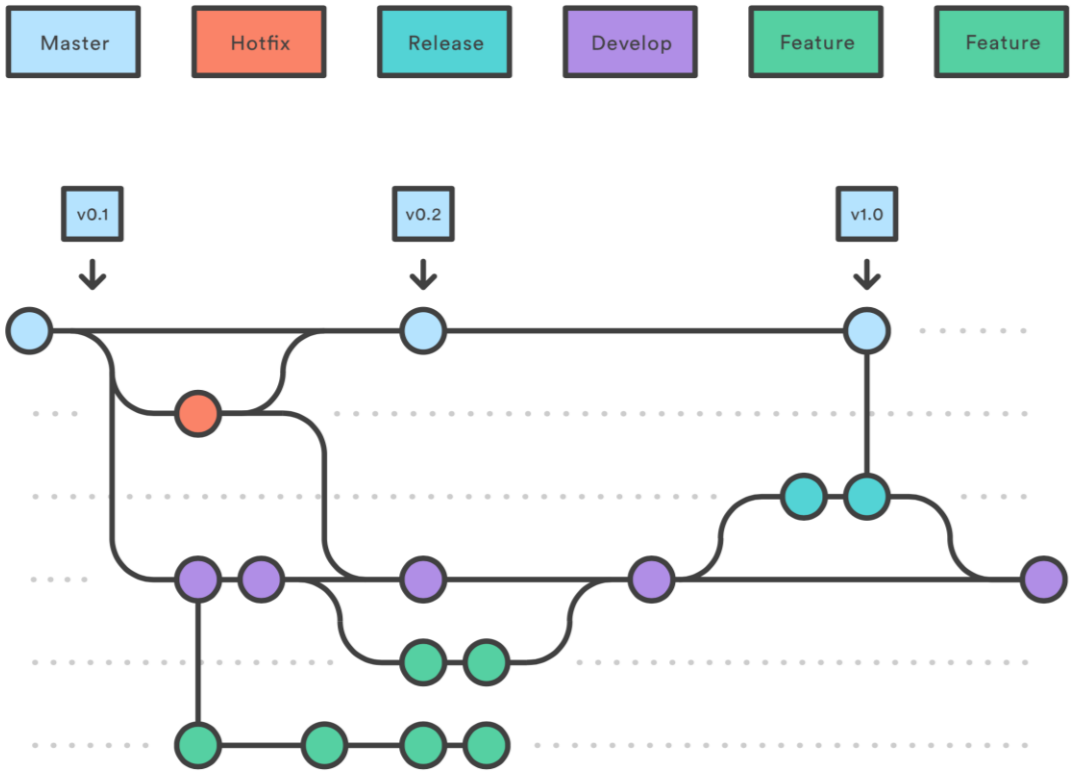
- Manage configurations?
  - Who changed that setting? Why?
  - Are you sure that code works the way you think it does?
  - Oh no the HDD failed without a backup!
- How do you manage them?
  - Deployment server?
  - FTP server?
  - Locally?
- Why Git?
  - Internally available code management system

The only wrong way to manage configs across your deployment is by not managing them…

Lessons Learned: Enabling Change Control

What we've realized through deploying applications with Git

- Repositories to manage every application
  - All administrative configurations managed via apps
  - Known good configurations kept to "master"
- Supports varying levels of control and approvals
  - Merge rules, repository permissions, and branches
  - Approvers to review any change to "master"



To enable automation applications must always be production ready

Bitbucket

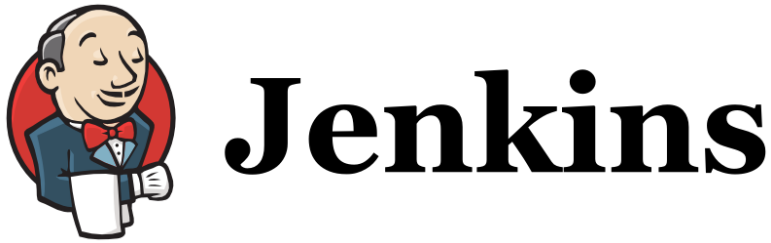
Automating Splunk from low-level functions to complex tasks

Get more stuff done, and spend less time on it

splunk> .conf18

Finding the right automation tool

Use what makes sense!



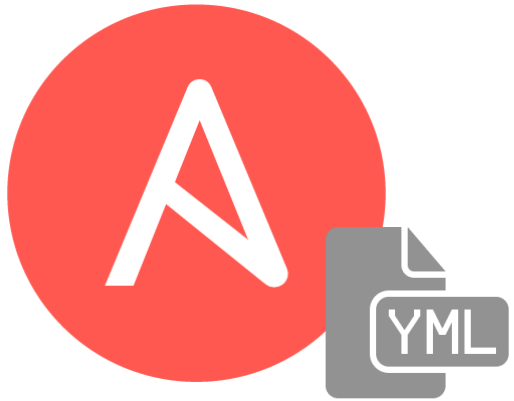
- Lots of good automation tools
  - Ansible, Chef, Jenkins, puppet, and more.. So where do you start?
  - Each of these tools are good at different things
  - Does your IT team already use an automation tool?
- Why did we choose Ansible?
  - It was already in use internally
  - Free and open source
  - Fast time to start for our engineers

There is no one right automation tool, it comes down to what is right for you.

Lessons Learned: Building Automation

Where do I start with automating splunk?

- Start with the basic blocks of work
  - Basic admin tasks (enable maintenance, restart splunkd, install splunk)
  - Repetitive pieces of work (update an app, reload serverclass, check filesystem permissions, rename bucket ids)
- Add in logic to enable advanced use-cases
  - What type of server do you need to deploy?
  - what apps need to be deployed for which server role?
  - What is the best way for me to upgrade Splunk my deployment?
  - How do I backup local changes on my server?
  - Should I back up the essence of my instance before updates?



The basic blocks or actions enable you to answer the complicated questions.

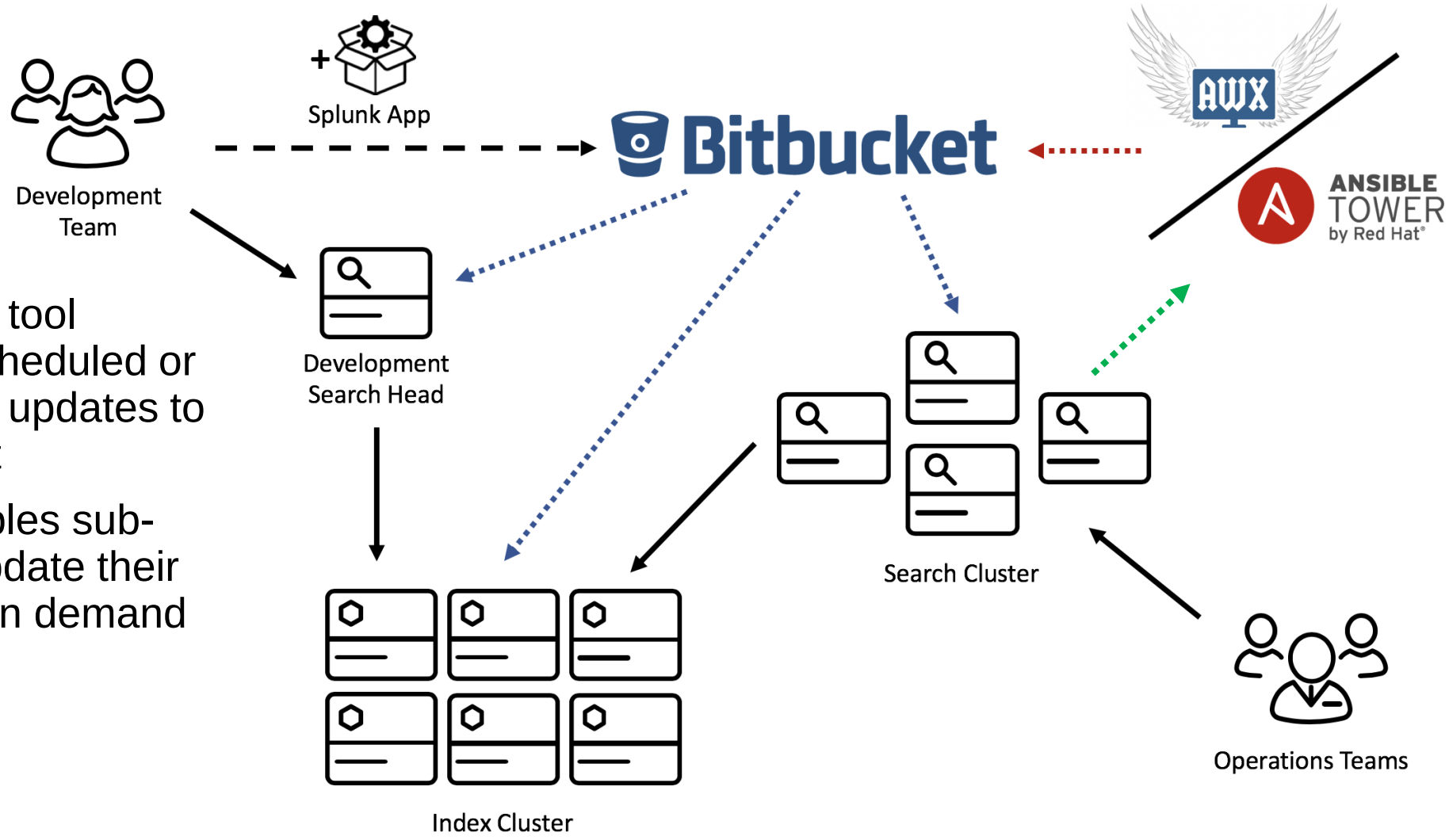
Using an automation management platform to enable CI/CD

Schedule your job so you can spend that time drinking coffee instead

splunk> .conf18

Reference Architecture

How our deployment works

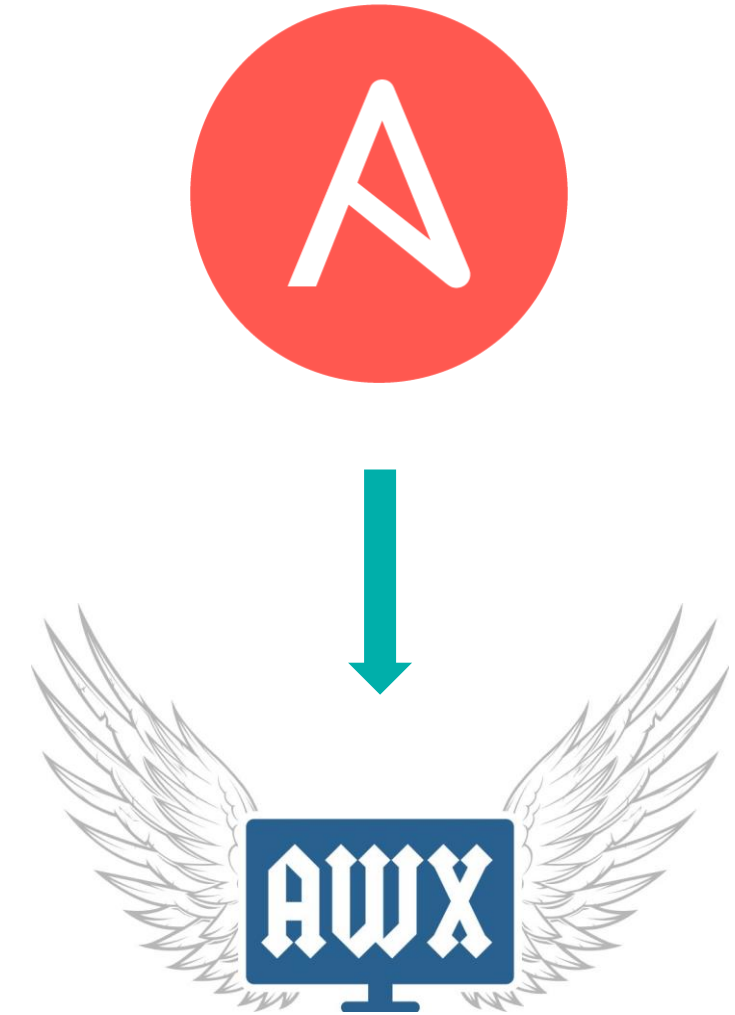


- Automation tool provides scheduled or on-demand updates to deployment
- RBAC enables sub-teams to update their own apps on demand

Lessons Learned: Enabling CD

Great I've got a bunch of ansible scripts, but I hate CLI, now what?

- Migrate ansible scripts to tower/awx templates
  - Make sure your ansible scripts are under change control
  - Use if logic to do the right actions for the right server roles
  - Use 'limit' to target a host or group
  - Setup template surveys for variables that you change often
- Setup an automation platform for success
  - Design RBAC and teams to fit how you want to operate (sub admins, read-only admins)
  - Send your logs back to splunk (AWX has HEC built in)
  - Schedule your important tasks (inventory update, scm update, templates/plays)



Taking the time to set this up right will save you lots of time later.

Using Splunk alerts to build self-healing into your deployment

Stop getting paged in the middle of the night

splunk> .conf18

Connecting it up

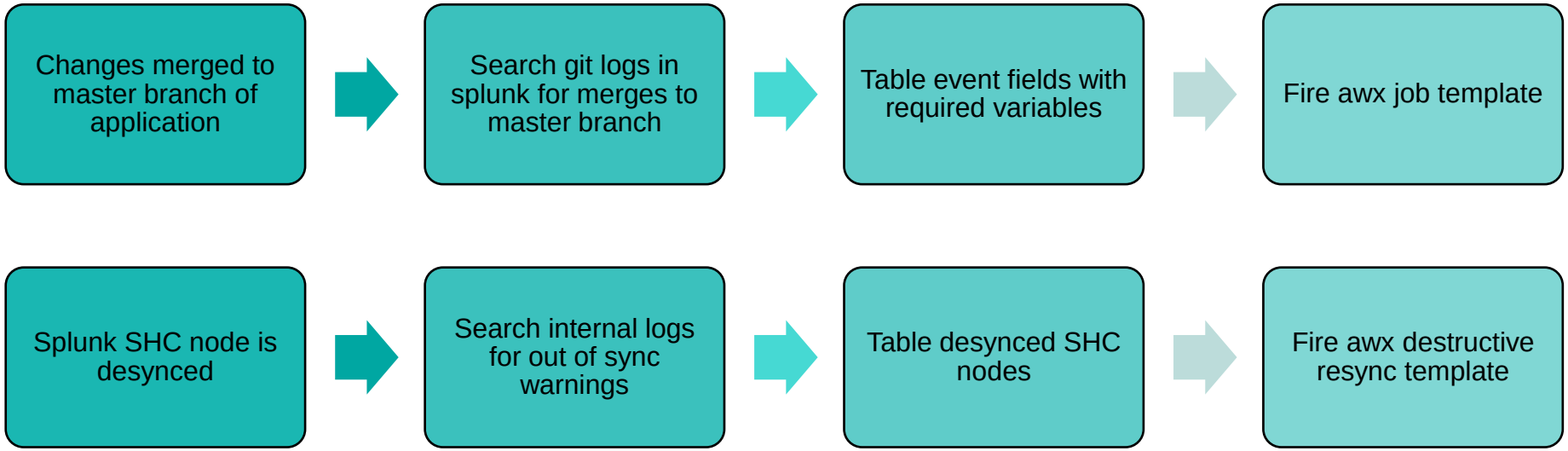
What good does it do?

- How does it work?
  - AWXLOOKUP: Lookup configured automation plays and required variables
  - AWXFIRE: Triggers automation plays and passes variables
- What do we gain?
  - Removes human variables
  - Shortened release cycles
  - Simplifies workflows/processes

There is no one right automation tool, it comes down to what is right for you.

Alert Logic

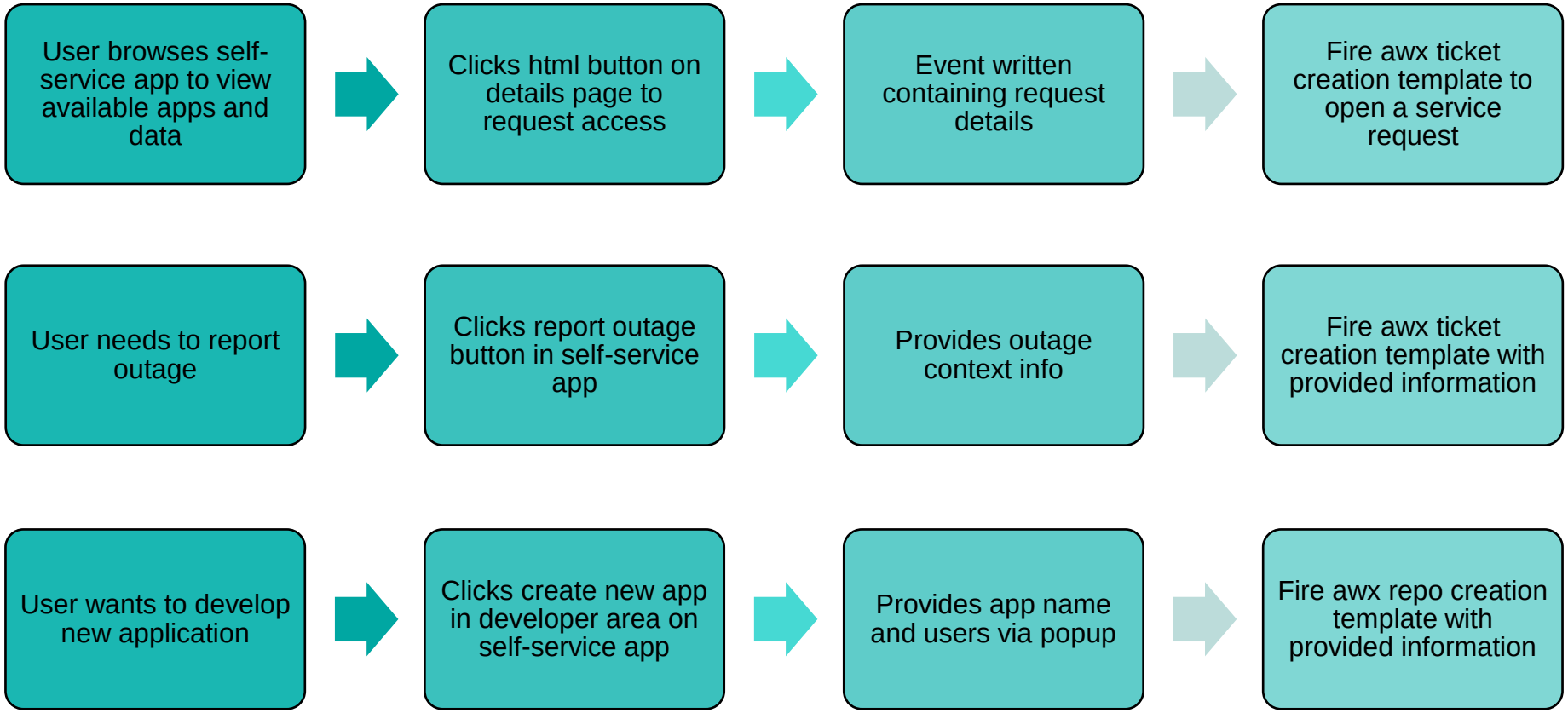
How we gained from connecting back into our automation service



splunk> .conf18

Self-Service Logic

How we envision empowering end-users (ie. Were still working on it)

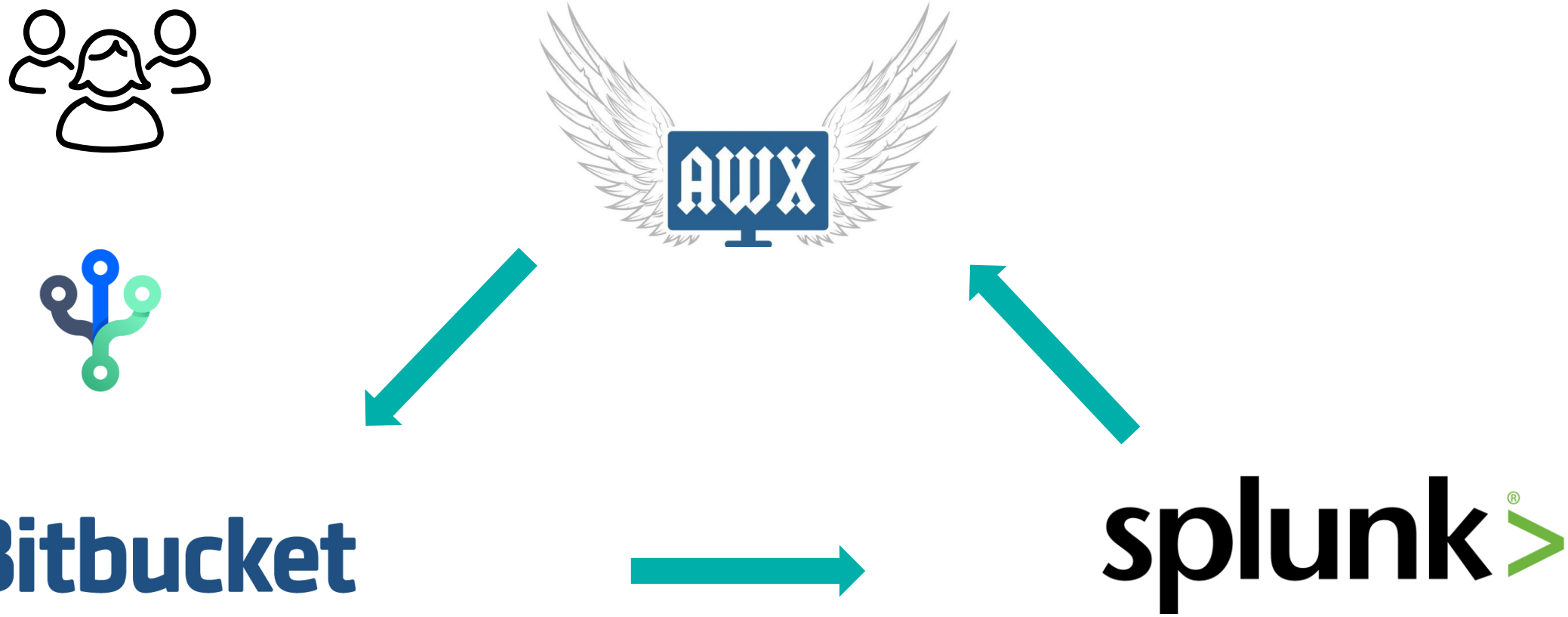


Pulling it all together

So what does it look like when all of this is put together?

Logical Layout of Technology

How our technology stack looks like now



Lets Recap

How to build a successful automation platform with Splunk

Managing Splunk in a automation friendly manner

Reduce time of management

Automating Splunk from low-level functions to complex tasks

Simplify menial workloads

Using an automation management platform to enable CI/CD

Lower admin overhead

Building Splunk alerts to add self-service and self-healing

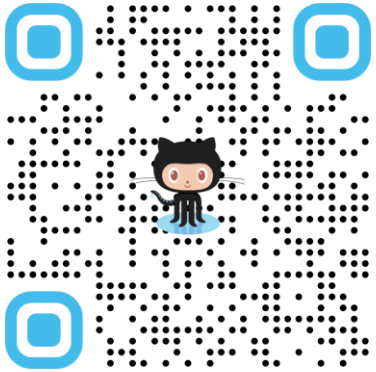
Automate L1 response

splunk> .conf18

Resources

You don't have to start from scratch!

- SplunkZero's Ansible playbooks
  - A good resource to build your own ansible library
  - https://github.com/Twistedsixty4/Splunk_Ansible



- Documentation for setting up AWX
  - https://developers.redhat.com/blog/2017/10/16/guide-starting-use-awx-top-openshift-upstream-red-hat-ansible-tower/

- AWX Splunk integration
  - Coming Soon to Splunkbase! (pending release)
  - Credit: Jack Stephenson

Questions?

splunk> .conf18

Thank You

Don't forget to rate this session in the .conf18 mobile app