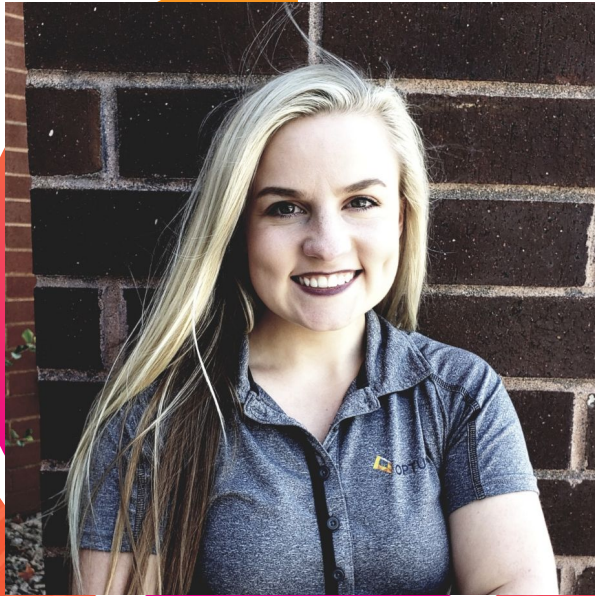


Can't We Just Have a Bot Run Our Deployments?

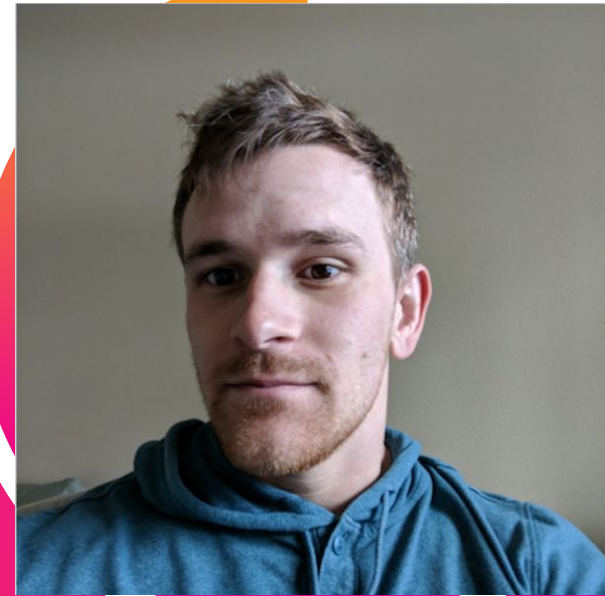
Yes, Yes We Can.

Shelbie Wise & Mitchell
Peters
Peter's Stuff | Optum



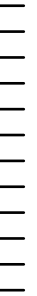
Shelbie Wise

Architect Analyst | Optum
Splunk Admin



Mitchell Peters

Sr IT Architect Analyst | Optum
Automation Engineer



Forward-Looking Statements



During the course of this presentation, we may make forward-looking statements regarding future events or plans 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 may differ materially. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, it may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements made herein.

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 functionalities described or to include any such feature or functionality in a future release.

Splunk, Splunk>, Turn Data Into Doing, The Engine for Machine Data, Splunk Cloud, Splunk Light and SPL are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2019 Splunk Inc. All rights reserved.



Who is Optum?

UNITEDHEALTH GROUP[®]



The Optum Splunk Project Scope

Started from the bottom
now we're here

1. In 2018 our team was given the task to onboard **all of Optum's top critical** business applications into Splunk.
 - 100+ Clients at 10-100+ hosts each
 - Each client will receive their own index, and search app which means we will be pushing conf files per client on a regular basis

Limitations at Our Company

Lack of root access for all Unix hosts

Adding Mounts/file systems

- Orchestration tools and daemons

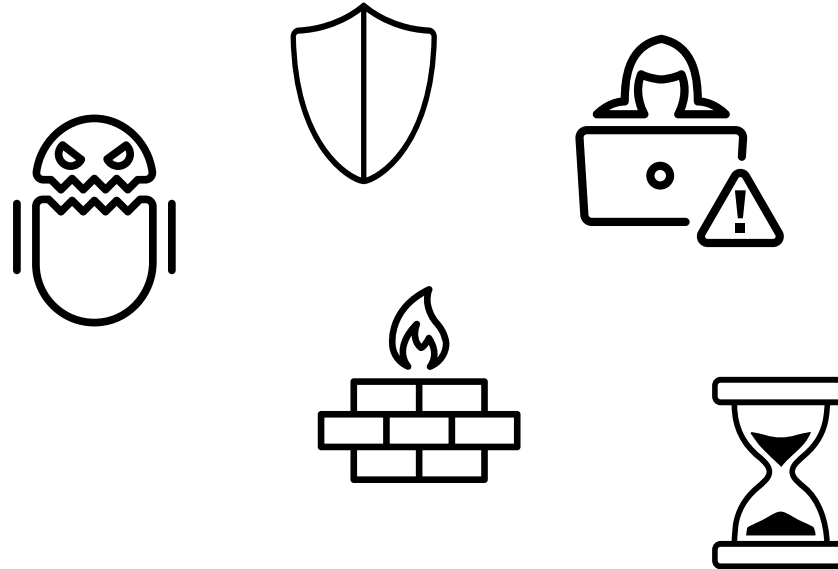
Segregated networks

- Talking is forbidden

Acquisitions

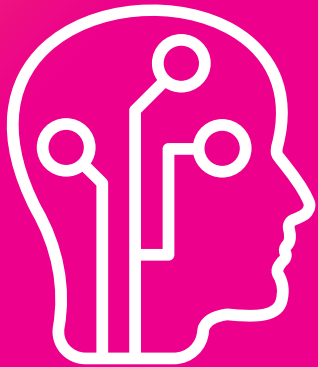
- Different infrastructures

Authentication



Goals:

Solutions > Problems



1. What is the fastest way we can install 1000+ forwarders
2. How can we schedule our deployments, so they can be hands free
3. How are we going to fix missing forwarders without boot start enabled
4. Who's going to manage the cluster bundle updates?

The Current Optum Splunk Environment

EST 2018



Indexers



Search Heads



Universal Forwarders



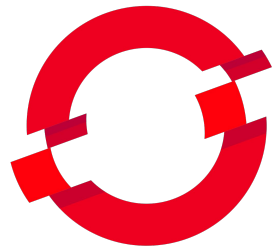
End Users



Unlimited



The Technology We Use

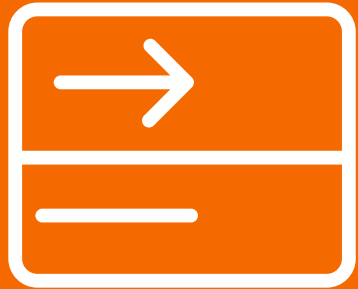


OPENSIFT

bash-\$

Bot Technology and Automation Breakdown

What and how we use our bot and automation on a regular basis



Deploy-UF

How we install and turn on our forwarders to thousands of hosts



Missing Forwarders

An alternative to enable boot start...because we don't have that privilege



Config Deployments

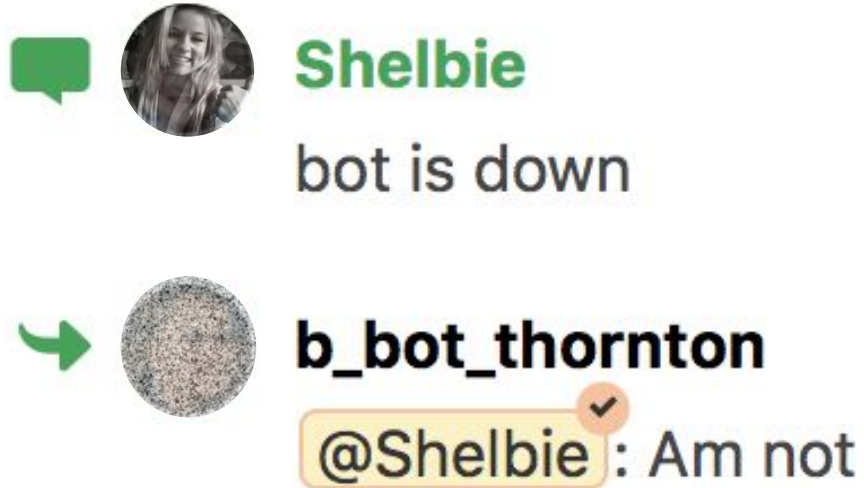
In case you wanted to see how we manage our bundle updates



Meet Our Bot

Meet Our Bot

Legal made us cover his face



- Name: b_bot_thornton
- Breed: Hubot
- Home: OpenShift Origin
- Age: 1.5 years old

Setting Up the Bot






How we did it, how you can too

1. Set up the environment for the bot install
 - We chose to use a container because it was easily available to us at the time...not saying you *have* to do this
2. Install **Hubot** (open source)
3. Install the **Authentication Module**
 - Had to re-write some of the core module to allow configs to be stored in **OpenShift Origin**
4. Write your hooks

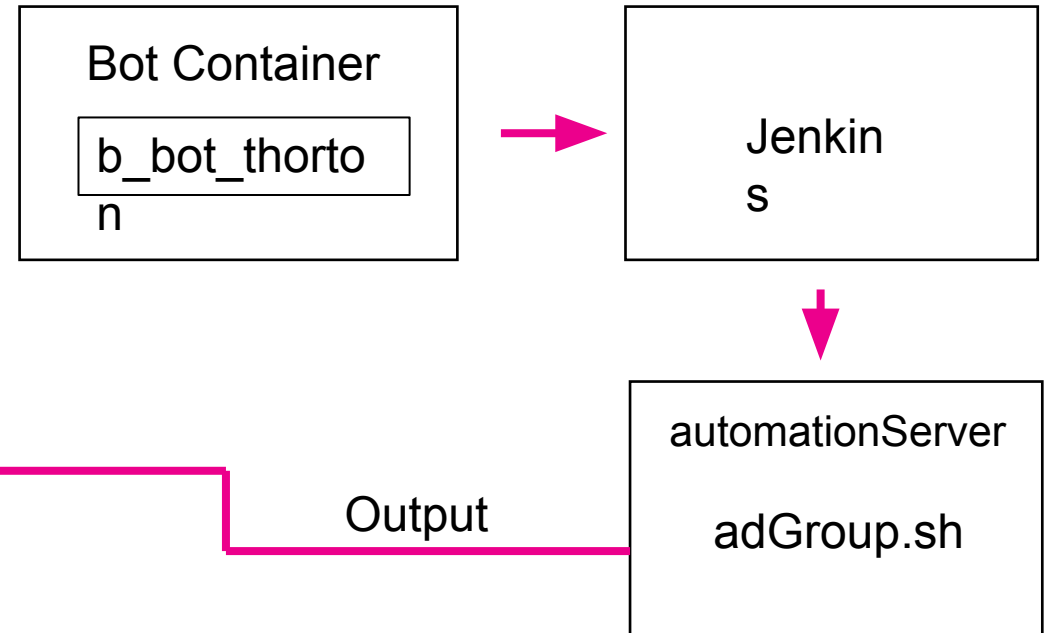
Bot Technology Flow

User posts their command

Flowdo

-  **Pete_Grady**
please make-splunk-groups pmalive odds
-  **b_bot_thornton**
making splunk groups owner: pmalive, group name: odds
-  **b_bot_thornton**
SSMOSplunk_odds_Prod created
-  **b_bot_thornton**
SSMOSplunk_odds_Poweruser created
-  **b_bot_thornton**
SSMOSplunk_odds_Nonprod created

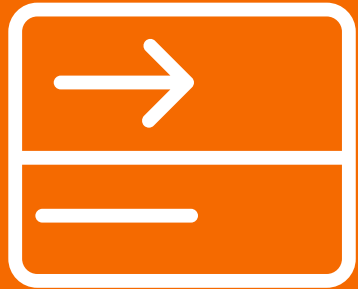
Duration = ~30 seconds



Example Above: Make-Groups command – our bot grabs the arguments and creates ad groups via api, groups are then used to map to Splunk roles

Bot Technology and Automation Breakdown

What and how we use our bot and automation on a regular basis



Deploy-UF

How we install and turn on our forwarders to thousands of hosts



Missing Forwarders

An alternative to enable boot start...because we don't have that privilege



Config Deployments

In case you wanted to see how we manage our bundle updates



deploy-uf

deploy-uf – Prerequisites (optum pre-reqs)

Linux/Sun/AIX

- Key authentication
- File System/mount point needs to be in place with proper permissions

Windows

- Need to have admin access
- WinRm must be enabled

deploy-uf – Prerequisites

More details [Splunk nuid & key authentication](#)

Splunk NUID

We created one non user id account (nuid) to own the forwarder installs

- Git repo holds our teams public keys
- This allows our team to all have access via keys
- No Passwords

keyPush.sh

Nightly cron runs the above script and pushes keys to hosts

Scripts adds our nuid account and pushes keys

Uses Ansible, Bash, and Jenkins

deploy-uf.sh

What does it do?

- Manages the clients and forwarder installs on the hosts

What technology does it use?

- Flowdock (chat ops – splunk admins)
- GitHub (holds the host list)
- Jenkins (runs the job)
- Shell Scripting (controls logic checks & installs)
- Ansible (manages the connections to the hosts)

Logic Checks:

- Operating system
- Is telnet installed, if so:
 - Test connectivity to our splunk deployment server & indexers (some lol)
- File System
 - 10 GB available
 - Permissions (can we write)
- Does the Forwarder already exist?
 - Checksum command against the splunk binary
 - Is NTP running

Deploy-uf: How Do We Run It?

From a Splunk admin's point of view

Splunk Admin Steps:

1. Create the host list in Github
2. Run the bot command in Flowdock
3. Validate and trouble shoot any failures




Flowdock – Bot deploy-uf run command

 **Shelbie**
Date & Time ↓
Host list ↓
On/Of f ↓
at 2019-06-25 15:01:00 deploy-uf OrxTibcoSW.inv on

Notice the run finished and posted back by 15:01

 **b_bot_thornton**
Here's the plan: date: Tue Jun 25 2019 15:01:00 GMT-0500 (CDT), file: OrxTibcoSW.inv, command: on .. to cancel, say:
`deploy-uf cancel OrxTibcoSW.inv_on_1561492860000`

Jun 25, 2019 15:01 # 

 **b_bot_thornton**
[tibco]
`#badufdeployer` run by: splunknu
tickets: none
hostname: , success: yes
hostname: , success: yes
hostname: , success: yes
hostname: , success: yes
hostname: , success: yes
hostname: , success: yes
hostname: , success: yes
hostname: , success: yes
hostname: , success: yes
hostname: , success: yes

Script Output

From the logging/splunk perspective

List ▾ ✎ Format 50 Per Page ▾		
☰ All Fields	i Time	Event
>	8/27/19 11:01:23.000 AM	<pre>{ [-] date: Tue Aug 27 11:01:23 CDT 2019 deploymentSum: cksum-is-good hostname: ██████████ isItReally: Its-running monitorGood: /monitor-exists-is-readable-and-is-writable-splunknu running: no success: yes time-manager: ntp-running }</pre> <p>Show as raw text</p> <p>host = ██████████ source = /deployer/npe-runstatus.stuff sourcetype = deployer_status</p>
>	8/27/19 10:30:35.000 AM	<pre>{ [-] date: Tue Aug 27 10:30:35 CDT 2019 deploymentSum: cksum-is-good</pre>

deploy-uf – Command Break Down

Tells bot to listen

at 2019-07-24-02:50:00
hostList.inv off

Splunk admins enter the desired time they want it to run

deploy-uf – prod install
npe-deploy-uf = non prod install

deploy-uf

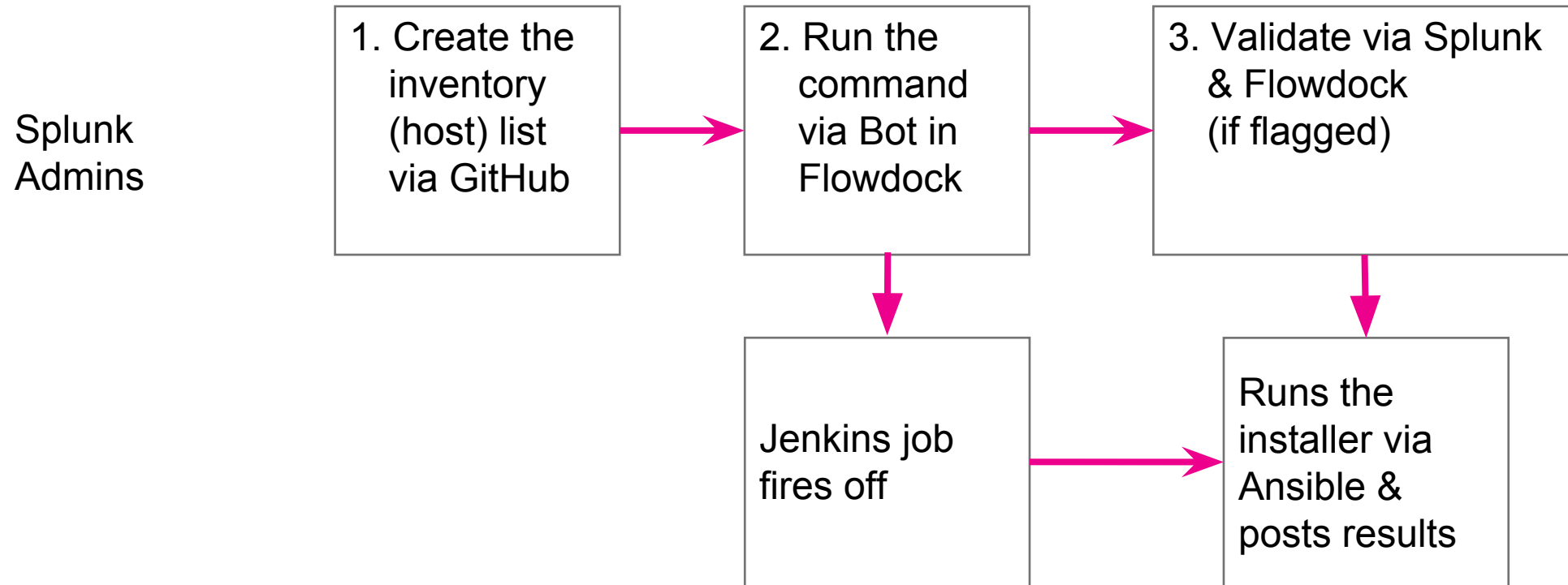
Host list name created from GitHub

on = ./splunk
start
off = ./splunk
stop

NOTE: we use “off” to install the UF without starting it

deploy-uf – Overall Flow

What is happening behind the scenes



deploy-uf – How Did This Help Our Team?

It **saved us LOADS** of time

It allowed us to **schedule** the run

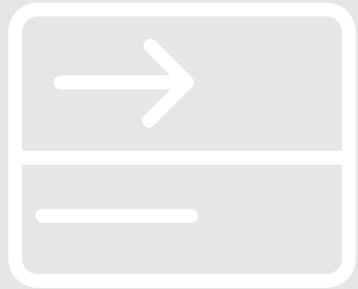
It allowed us to **grant offshore access** to run automation

We make **less mistakes**

Because of the success with the **automation** our team was able to prove that we no longer were considered a **“risky”** install and were able to **achieve a pre approval process** so that we could schedule CRs whenever **we** needed.

Bot Technology and Automation Breakdown

What and how we use our bot and automation on a regular basis



Deploy-UF

How we install and turn on our forwarders to thousands of hosts



Missing Forwarders

An alternative to enable boot start...because we don't have that privilege



Config Deployments

In case you wanted to see how we do it



missing_forwards

Missing Forwarders

A work around solution for not having root privs

What does it do?

Utilizes our deploy-uf script to run every 30 minutes against all hosts that report missing from the Monitoring Console

Why did we need this?

- We don't have root access to enable boot-start on our linux/aix/sun hosts
- Helps us cleanse out retired hosts
- Assists in identifying corrupt file systems

Missing Forwarders

General flow – this process runs every 30 minutes

Monitoring Console

Generate the missing forwarder list via **dmc_forwarder_assets** lookup

Custom alert is set up via splunk to trigger a script to the ansible server

ansibleServer

Takes the list from the **dmc_forwarder_assets** and checks it against the **serverclass.conf**

IF a host is in BOTH lists, **missing_forwarders.sh** will call Jenkins to run **deploy-uf.sh**







deploy-uf.sh

Attempts to restart/reinstall UF

Posts output to Flowdock

Missing Forwarders

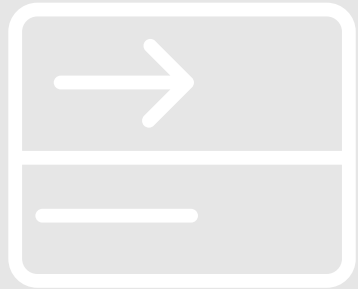
Output in Flowdock

-   **b_bot_thornton**
[missing-hosts]
#npe-ufdeployer run by: splunknu
tickets: none
hostname: ██████████, success: yes
-   **b_bot_thornton**
[missing-hosts]
#npe-ufdeployer run by: splunknu
tickets: none
hostname: ██████████, success: yes
-   **b_bot_thornton**
[missing-hosts]
#badufdeployer run by: splunknu
tickets: none
hostname: ██████████, success: yes
hostname: ██████████, success: UNREACHABLE!

Note: Hostnames and images have been covered for personal reasons

Bot Technology and Automation Breakdown

What and how we use our bot and automation on a regular basis



Deploy-UF

How we install and turn on our forwarders to thousands of hosts



Missing Forwarders

An alternative to enable boot start...because we don't have that privilege



Config Deployments

In case you wanted to see how we do it



Config Deployments

Deployment Configs

How we have our backend configs set up in git

Deployment Server

DS Repo

- serverclass.conf
- inputs.conf

Cluster Master

CM Repo

- indexes.conf
- props.conf
- transforms.conf

Search Head Deployer

SHD Repo

- app.conf
- authorize.conf
- authentication.conf
- local.meta
- user-prefs

Each repo has their own Jenkins job with a cron

Config Deployments – Technology Used

Git hooks post notifications to Jenkins

- Set some notifications in git, set Jenkins to listen to them

Jenkins executes the scripts, and notifies of success and failure

- Define your next steps either in a Jenkins or leverage existing scripts and save some re-work for another day

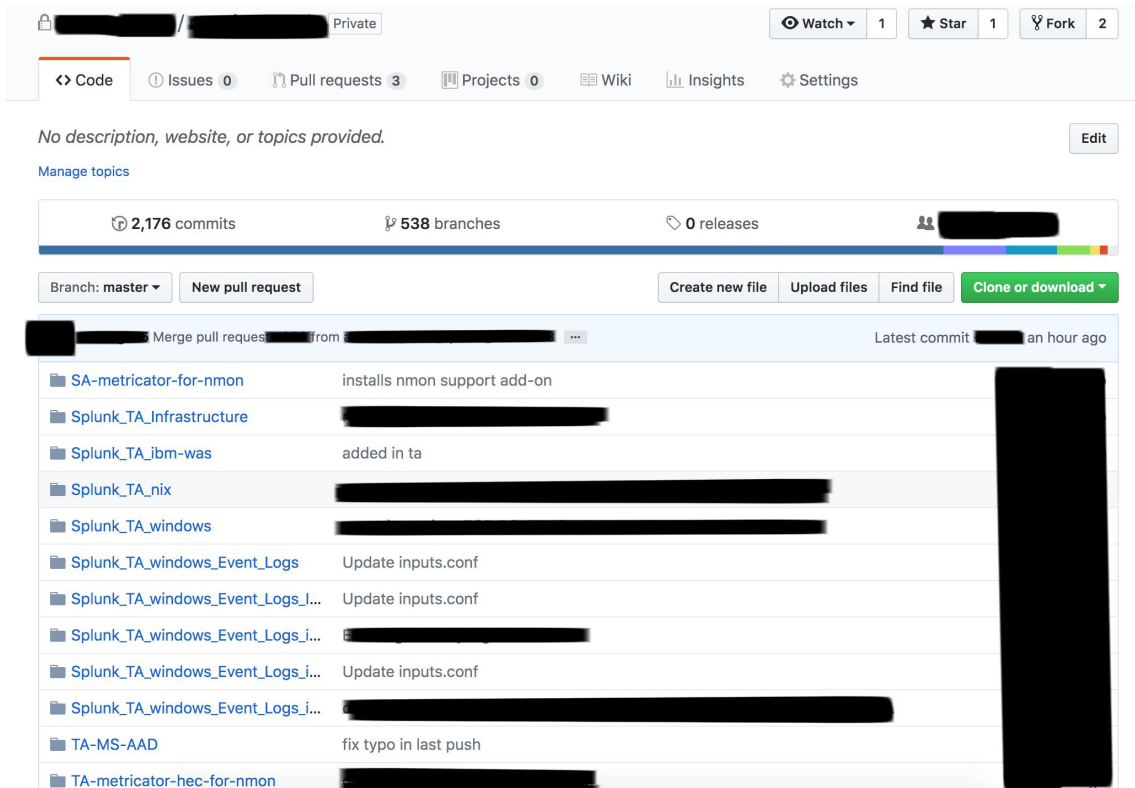
Script runs on splunk cluster member, reports status back to chat

- Make sure you enable hooks where you need them, (see next slide)

You can run it all in one place if your environments support it, ours does not

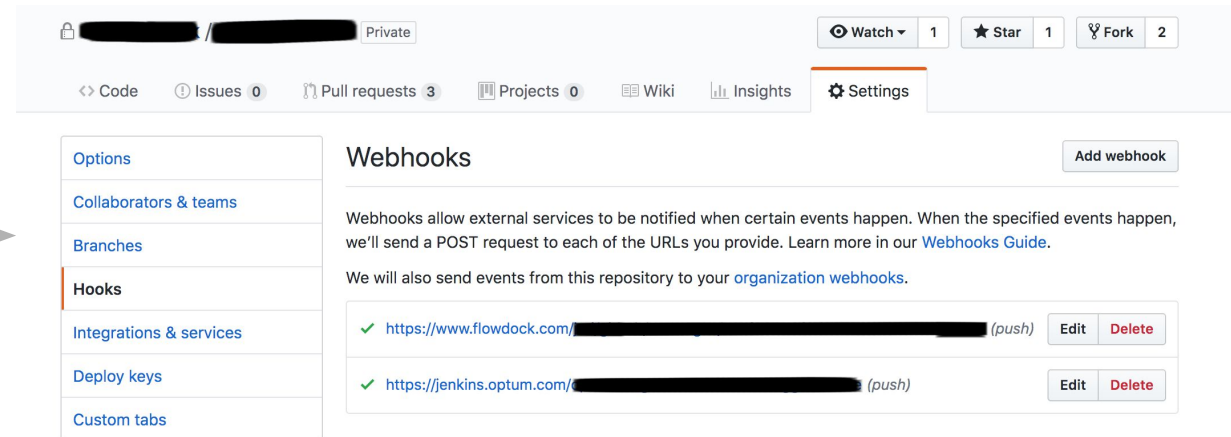
What it Looks Like

From a GitHub perspective – Repo names blurred



This screenshot shows the main view of a GitHub repository. The repository name and owner are blurred. The page includes navigation tabs for Code, Issues (0), Pull requests (3), Projects (0), Wiki, Insights, and Settings. Below the navigation, there is a description field with the text "No description, website, or topics provided." and an "Edit" button. A statistics bar shows 2,176 commits, 538 branches, and 0 releases. Below this, there are buttons for "Branch: master", "New pull request", "Create new file", "Upload files", "Find file", and "Clone or download". A list of recent commits is visible, with repository names blurred. The commit list includes:

- SA-metricator-for-nmon: installs nmon support add-on
- Splunk_TA_Infrastructure: [blurred]
- Splunk_TA_ibm-was: added in ta
- Splunk_TA_nix: [blurred]
- Splunk_TA_windows: [blurred]
- Splunk_TA_windows_Event_Logs: Update inputs.conf
- Splunk_TA_windows_Event_Logs_I...: Update inputs.conf
- Splunk_TA_windows_Event_Logs_i...: [blurred]
- Splunk_TA_windows_Event_Logs_i...: Update inputs.conf
- Splunk_TA_windows_Event_Logs_i...: [blurred]
- TA-MS-AAD: fix typo in last push
- TA-metricator-hec-for-nmon: [blurred]



This screenshot shows the "Settings" page for the same repository. The repository name and owner are blurred. The "Webhooks" section is active, showing a list of configured webhooks. The page includes navigation tabs for Code, Issues (0), Pull requests (3), Projects (0), Wiki, Insights, and Settings. The "Webhooks" section contains the following information:

- Webhooks** (Add webhook)
- Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).
- We will also send events from this repository to your [organization webhooks](#).
- Two webhooks are listed:

Webhook URL	Event	Actions
https://www.flowdock.com/[blurred]	(push)	Edit Delete
https://jenkins.optum.com/[blurred]	(push)	Edit Delete

What it Also Looks Like

The screenshot shows the Jenkins web interface for the 'npe-splunk' job. The breadcrumb navigation shows 'Jenkins > npe-splunk'. On the left, there is a sidebar with navigation options: Up, Status, Configure, New Item, Delete Folder, People, Build History, and Project Relationship. The main content area shows the job name 'npe-splunk' and a table of build history. The table has columns for 'S' (Success), 'W' (Warnings), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. Two builds are listed, both with success status and warning icons.

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	[redacted]merge	50 min - #829	2 mo 10 days - #719	19 sec
✓	☀	[redacted]curl	47 min - #727	1 mo 19 days - #649	30 sec

```
[redacted]@ [redacted]:/home/[redacted]/inputs
$ ls -a [redacted]/splunk/etc/deployment-apps/
.
..
.git
om_kvstore_enable
optum_ace_inputs
optum_acis_inputs
optum_acq_inputs
optum_apollo_cmcp_inputs
optum_apollo_portal_inputs
optum_apollo_qflpr_inputs
optum_asp_inputs
optum_azure_pack_inputs
optum_bank_inputs
optum_beportal_inputs
optum_brx_patientportal_inputs
optum_btb_na_inputs
optum_cap_inputs
optum_cdb_bdpaas-hec_props
optum_cdb_hec_props
optum_clmintegrations-hec_props
optum_npe_fw_limits
optum_npe_fw_outputs
optum_npe_hf_admin
optum_npe_hf_http_inputs
optum_npe_hf_limits
optum_npe_hf_outputs
optum_npe_hyperion_inputs
optum_npe_iib_inputs
optum_npe_lean_internal_inputs
optum_npe_lean_tesales_inputs
optum_npe_myuhc_inputs
optum_npe_odx_inputs
optum_npe_oil_inputs
optum_npe_ommswy_inputs
optum_npe_omni_inputs
optum_npe_optum_navigator_inputs
optum_npe_ose_outputs
optum_npe_pcl_inputs
optum_npe_qlikview_inputs
optum_oc_fullwell_inputs
```

Technology Flow – Bundle Updates



Note: ALL backend configs are held in Github – Splunk admins absolutely never log into ANY splunk hardware to update/make changes

Config Deployments

Why did we set it up this way aka how does it help?

Manages all of the cluster bundle updates for us

Prevents admins from logging into boxes

Cron allows us to keep pushing configs knowing the order of configs will be taken care of (example props before inputs)

Tips: Set your cron jobs off of order of configs and low traffic hours



Q&A

Shelbie Wise | Architect Analyst

Mitchell Peters | Sr IT Architect Analyst

Final Tips

Notes you can save and read later

Key authentication – always push keys when you can. This prevents user accounts from getting locked

Consider using one **NUID** for your forwarder installs – this promotes consistency among hosts

Take advantage of **open source technology**

Use **git/version control** for ALL of and everything you do, this eliminates the need to have access to splunk hardware



splunk>

Thank

You



Go to the .conf19 mobile app to

RATE THIS SESSION

