Using Splunk to Increase Developer Confidence in the Pivotal Cloud Foundry Platform

Shubham Jain & Kirk Hanson

May 2018 | Version 1.0

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

Agenda

- Pivotal Cloud Foundry Introduction
- Pivotal Cloud Foundry Nozzle Overview and Architecture
- Pivotal Cloud Foundry health App Introduction and Overview
- Using Pivotal to make developers happier

Splunk & Pivotal Introduction

Overview

splunk> .conf18

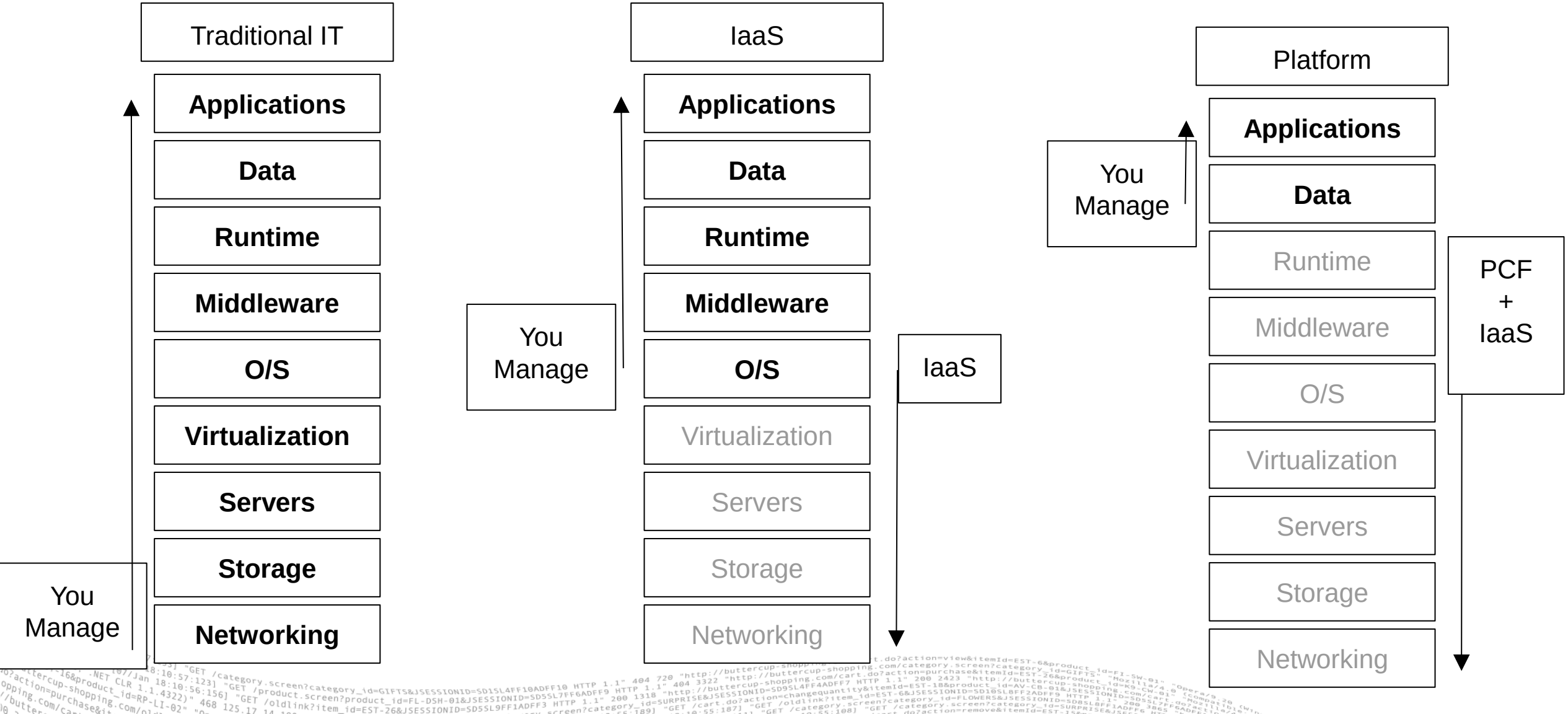
Cloud Foundry

- Open Source, multi cloud application Platform as a Service (PaaS) governed by the Cloud Foundry Foundation
- Promoted as Continuous delivery platform for full application lifecycle management
- Platform deploying and operating wide variety applications written in different languages (Java, .NET, Node.js, Python, Go etc) which can be deployed on premise or in public cloud

Pivotal Cloud Foundry

- Developer Productivity
- Operator Efficiency
- Comprehensive Security
- High Availability

The Power of the PCF Platform



Splunk Firehose Nozzle for PCF

Architecture & Overview

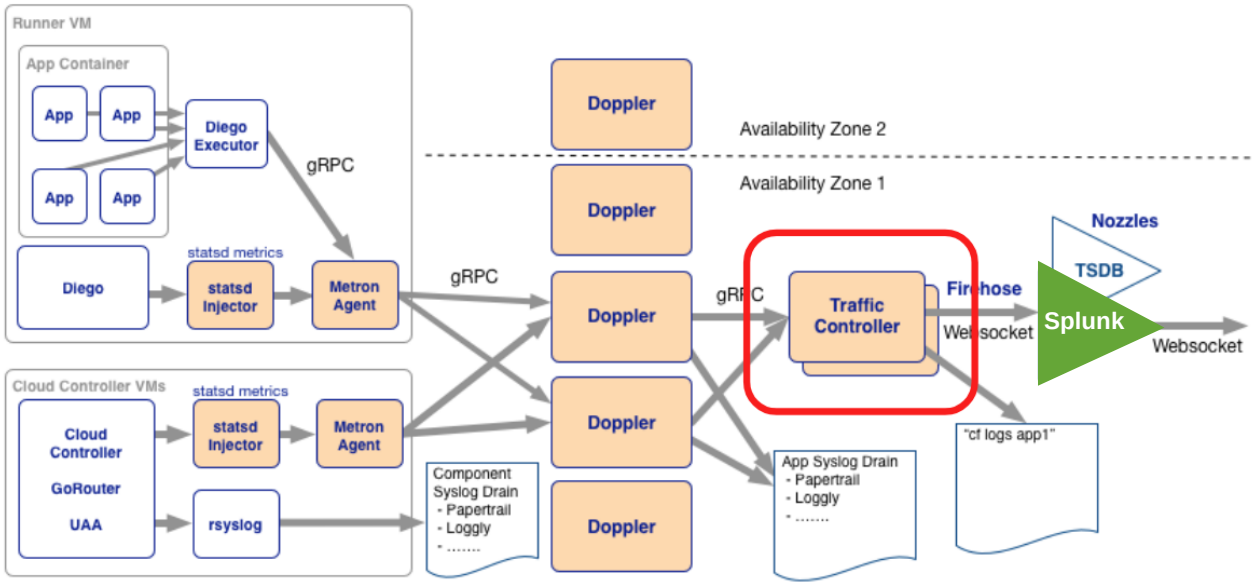
splunk> .conf18

Nozzle Overview

What is a Nozzle?

- A component dedicated to reading and processing data that streams from the Firehose Loggregator.
- Can be deployed as a managed service or an application.

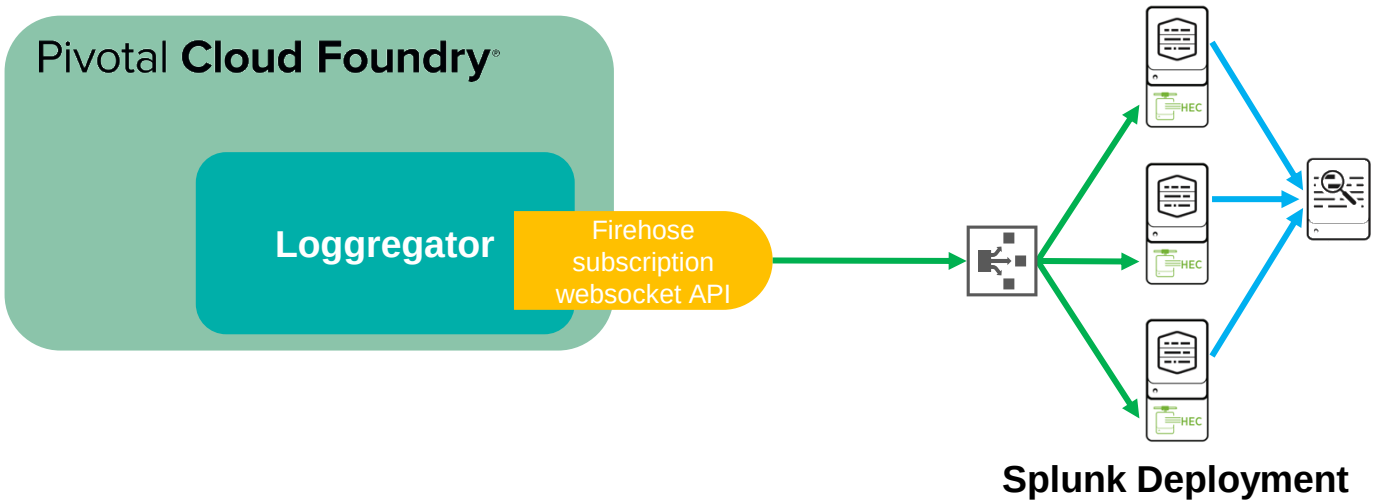
PCF Loggregator Architecture



Pivotal Cloud Foundry Integration

The Splunk Firehose Nozzle for PCF collects events from the PCF **Loggregator** endpoint and streams them to Splunk via HTTP Event Collector

- **High performance and reliability** with Nozzle's in-memory queue buffers, and parallel clients to scale out multiple ingestion channels to HEC
- **Simple deployment** natively within a PCF environment using a tile, or through CLI
- **Easy scalability** of ingest by adding more HEC data collection nodes behind a load balancer



Splunk Firehose Nozzle for PCF Features

OOTB

Setup with out-of-the-box data parsing and enrichment for various PCF event types

HEC

Reliable event delivery by leveraging Splunk's HTTP Event Collector endpoint

SSL

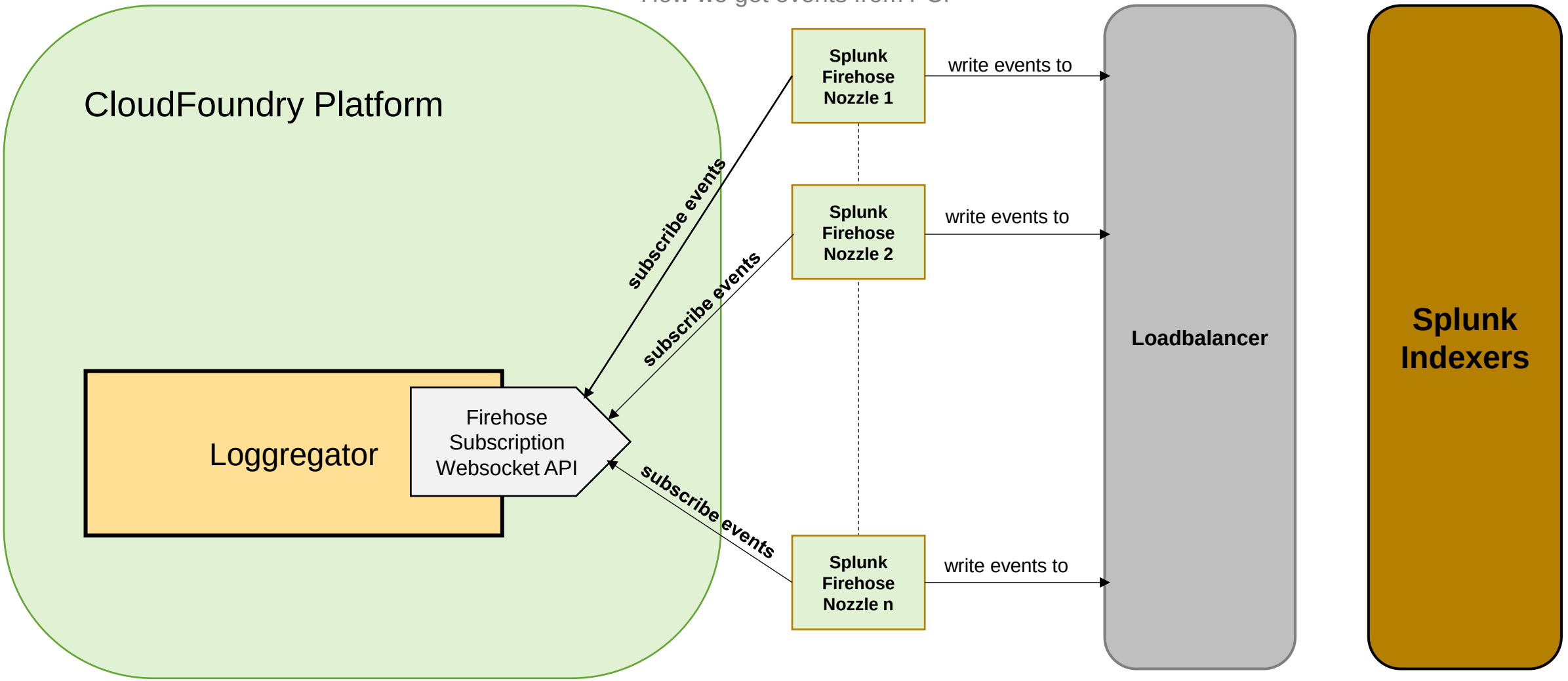
Secure forwarding from PCF into external Splunk environments using user-provided SSL certificates

Scales

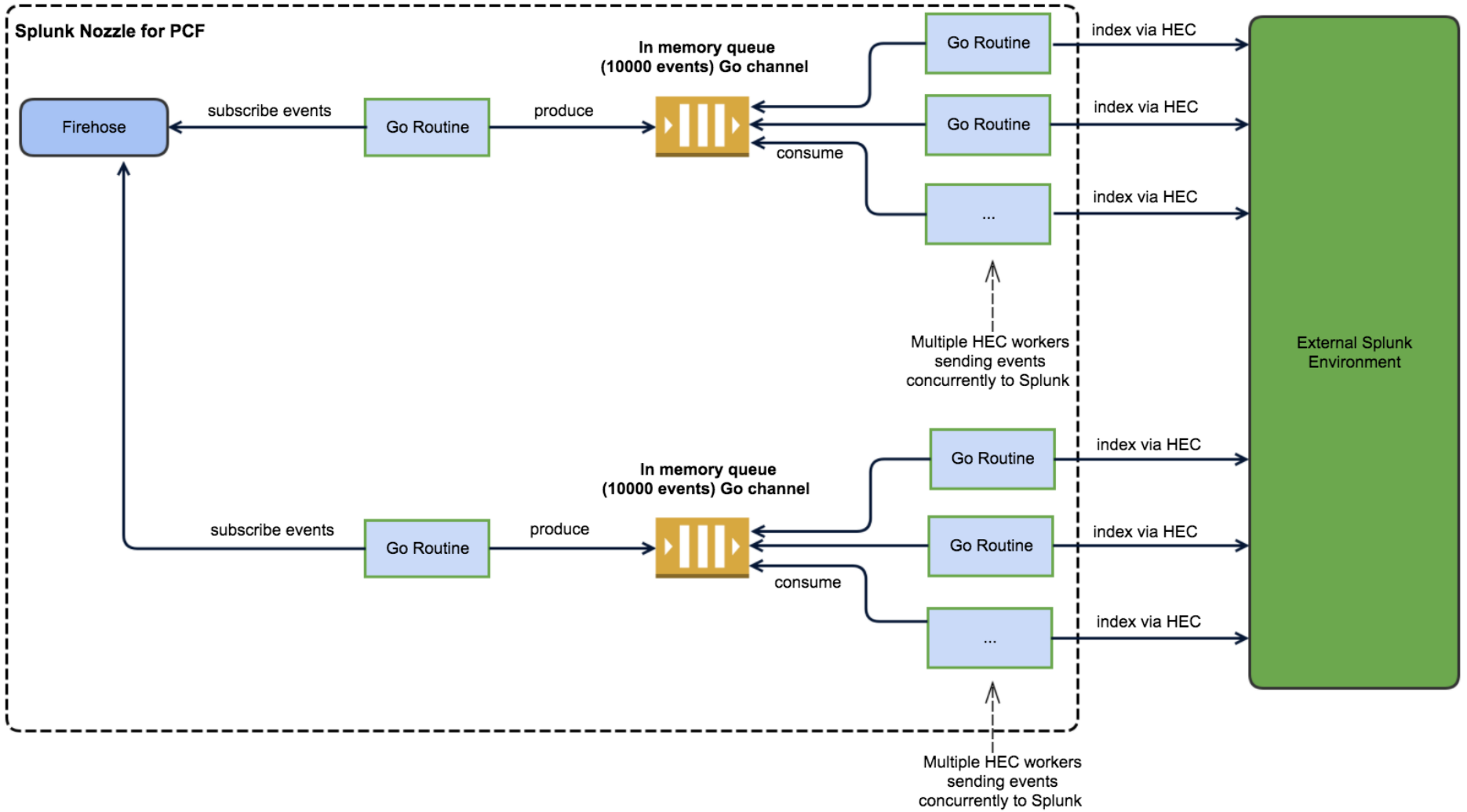
Scales out to meet increasing data volume and number of apps

PCF Nozzle Deployment Architecture

How we get events from PCF



Splunk Built Internal Components

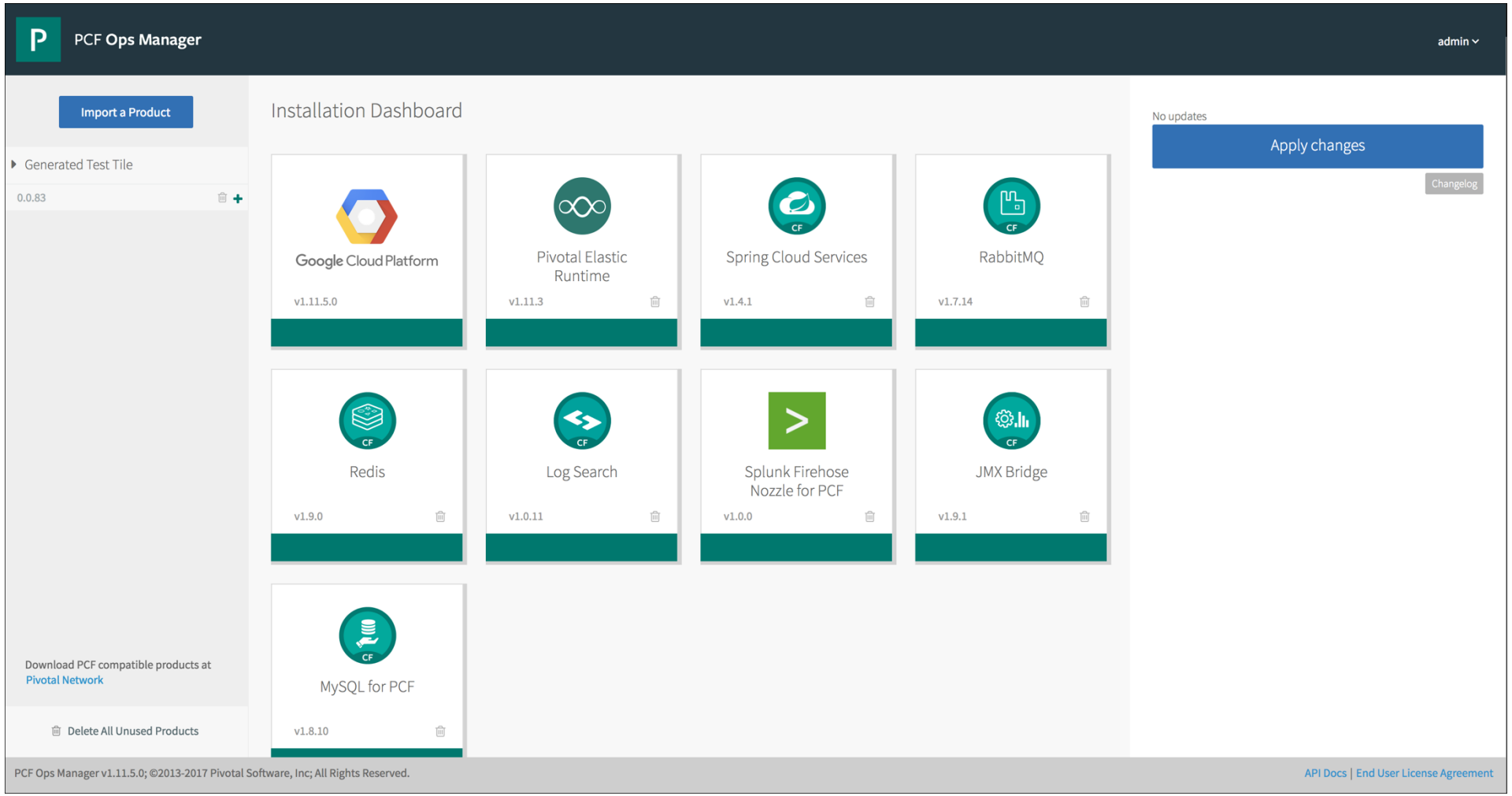


Installation and Deployment

How to install and deploy Splunk Firehose Nozzle in PCF via Tile?

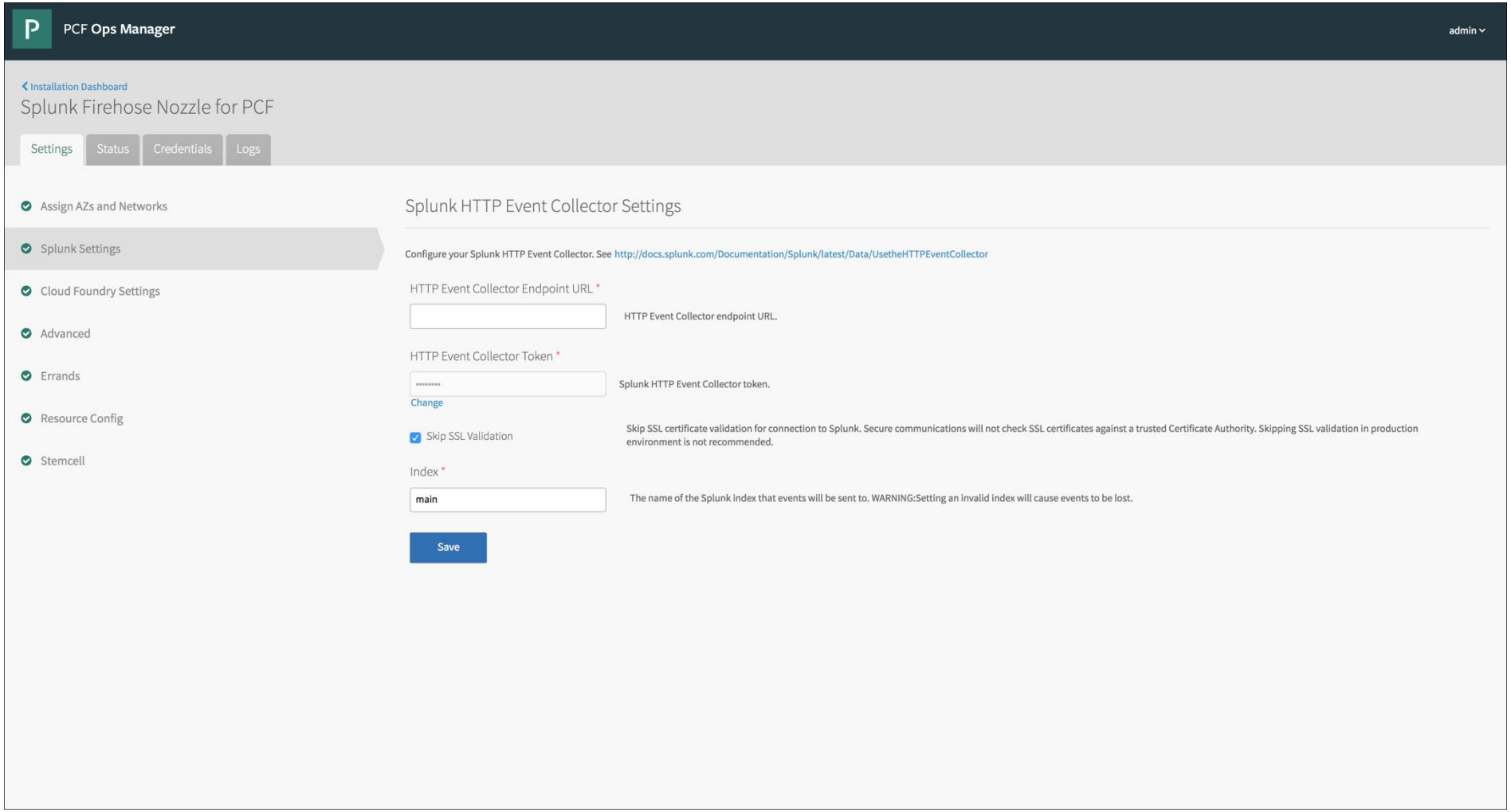
- Download and install the Splunk tile from Pivotal Network
- Navigate to the Ops Manager Installation Dashboard and click **Import a Product** to upload the product file and add it to your staging area
- Configure the Tile as per requirements
- Apply changes to deploy the nozzle

Deployment Through Tile

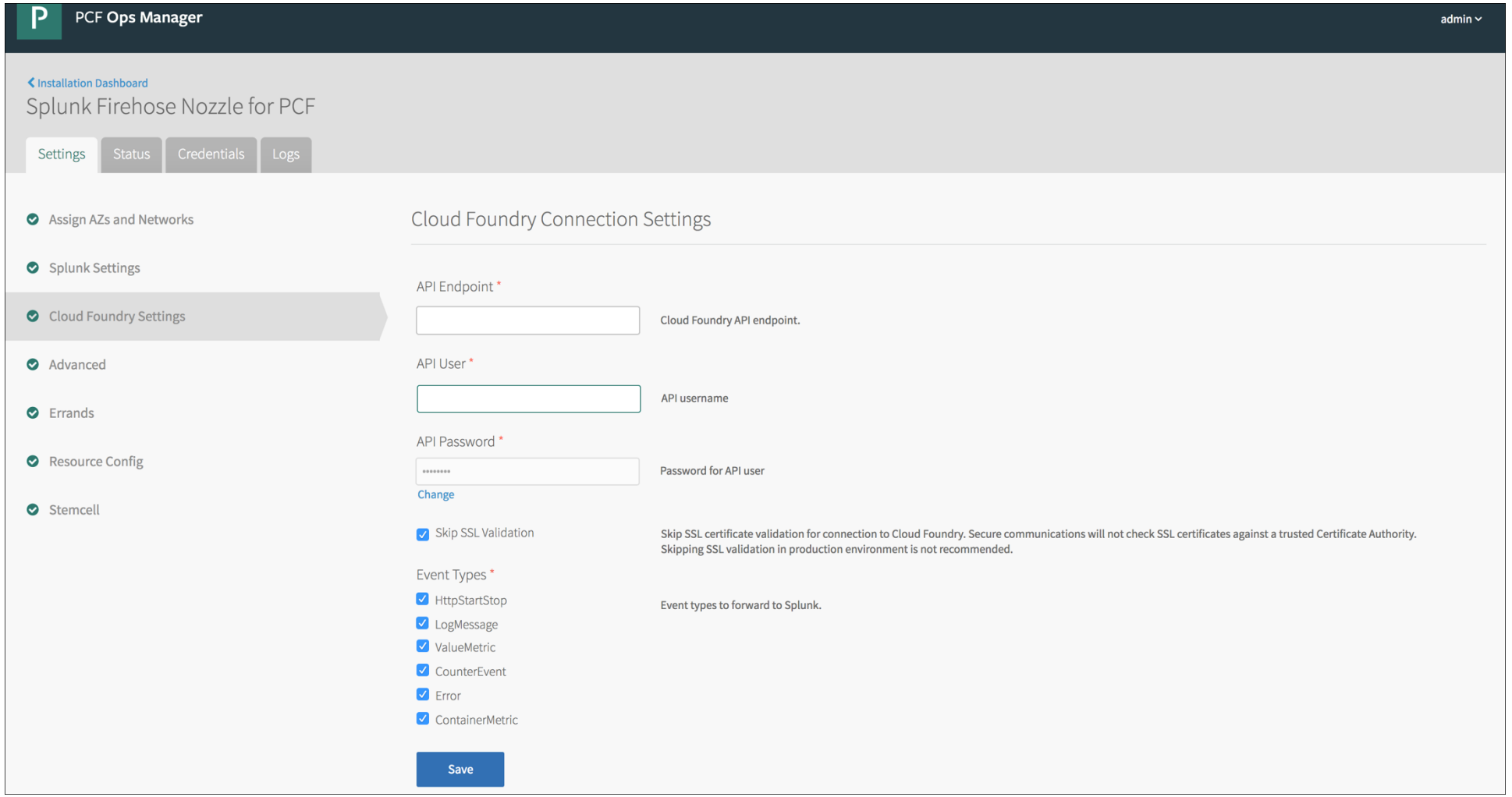


splunk> .conf18

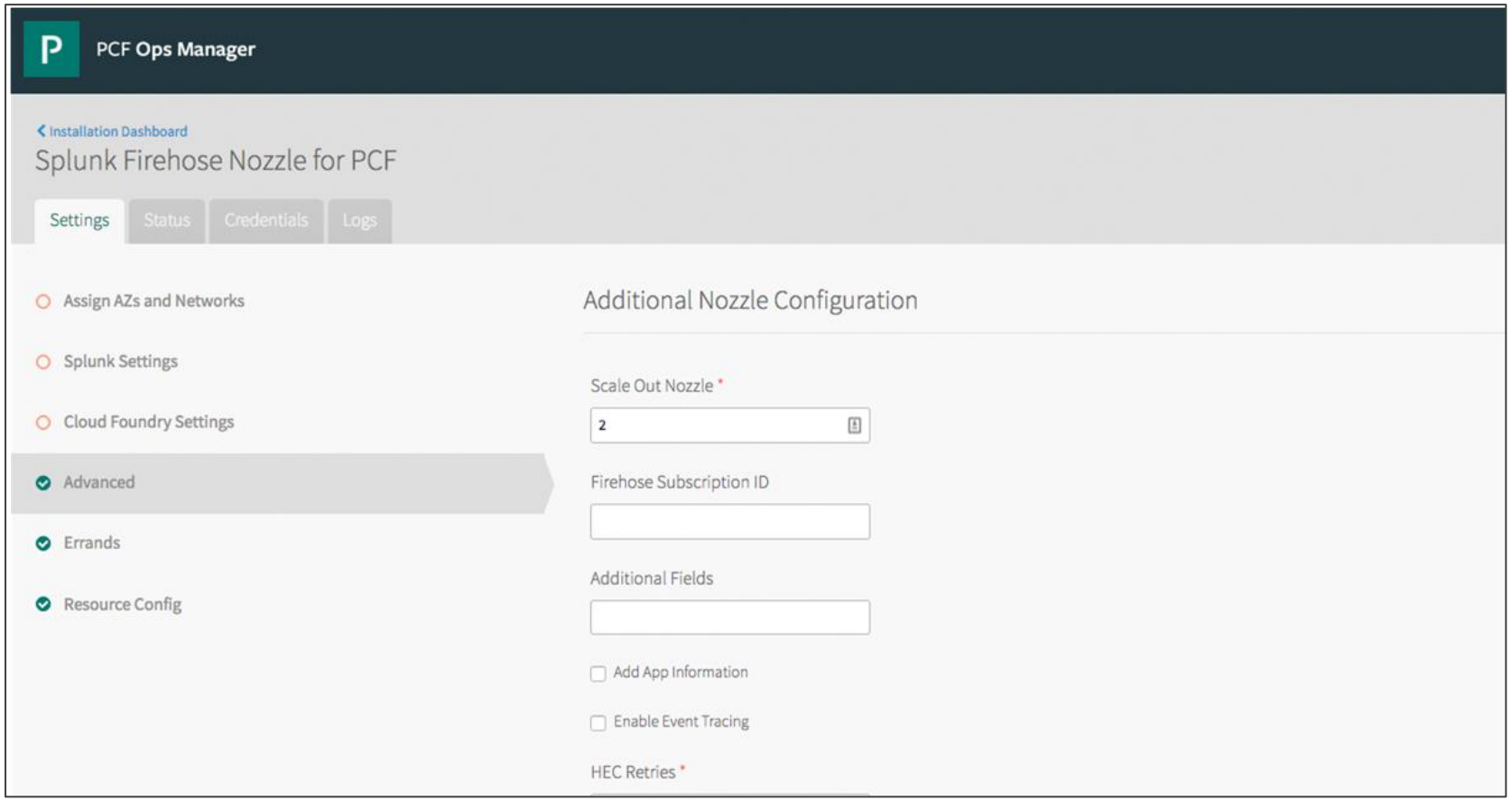
Splunk Configuration



Cloud Foundry Configuration



Advanced Configuration



Configuration options:

- **Scale Out Nozzle**
- **Firehose Subscription ID**
- **Additional Fields**:
- **Add App Information**
- **Enable Event Tracing**
- **HEC Retries**
- **HEC Batch Size**
- **HEC Workers**
- **Consumer Queue Size**
- **Flush Interval**
- **Missing App Cache Invalidate TTL**
- **App Cache Invalidate TTL**
- **App Limits**
- **Ignore Missing App**

splunk> .conf18

Splunk Pivotal Cloud Foundry Health App

Application design, overview and walkthrough

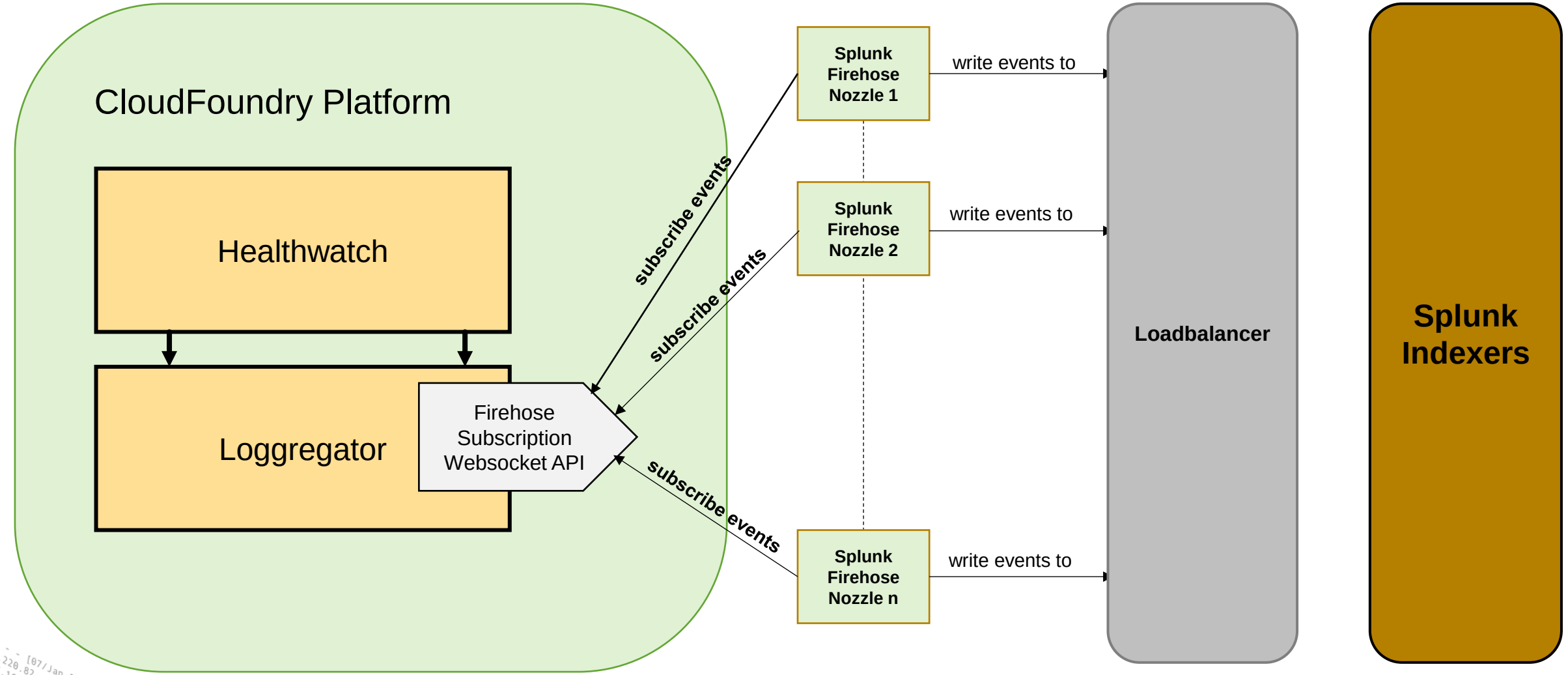
splunk> .conf18

App Introduction

- Customer Questions that drove this app creation
  - So I have the nozzle now what?
  - How do I visualize this based on PCF best practices?
- Utilize metrics generated from PCF Healthwatch
- Develop best practices for Application developer-centric dashboards

PCF Nozzle Deployment Architecture

How Healthwatch data is injected



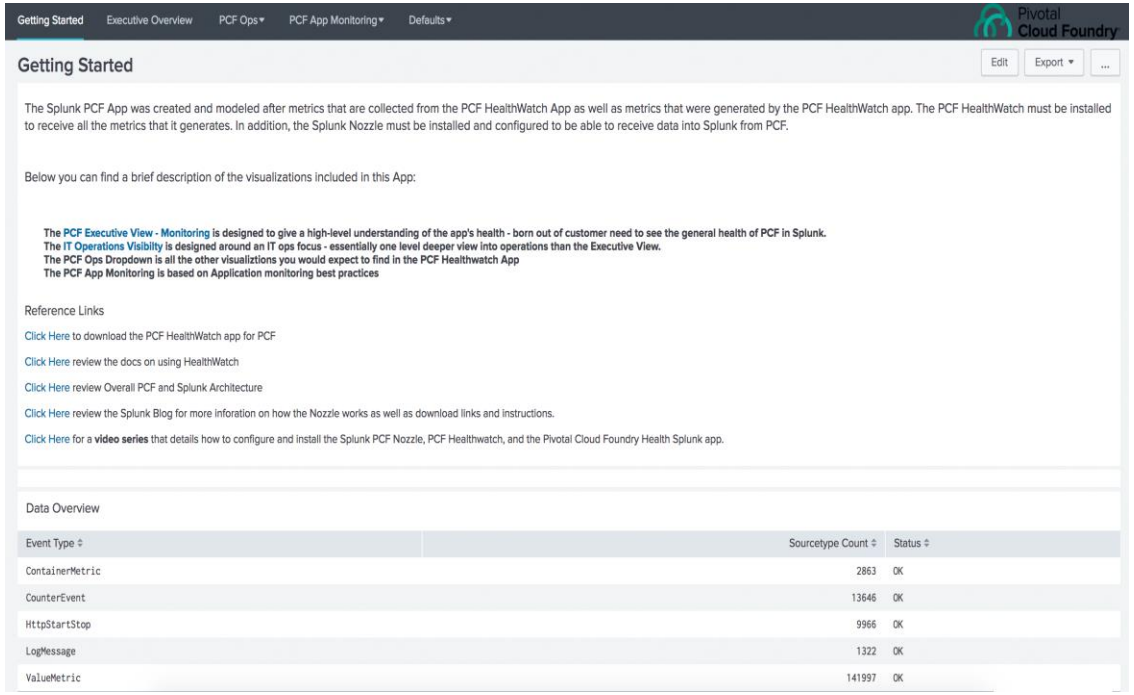
Splunk Pivotal Cloud Foundry Health App - Walkthrough

Application design, overview and walkthrough

splunk> .conf18

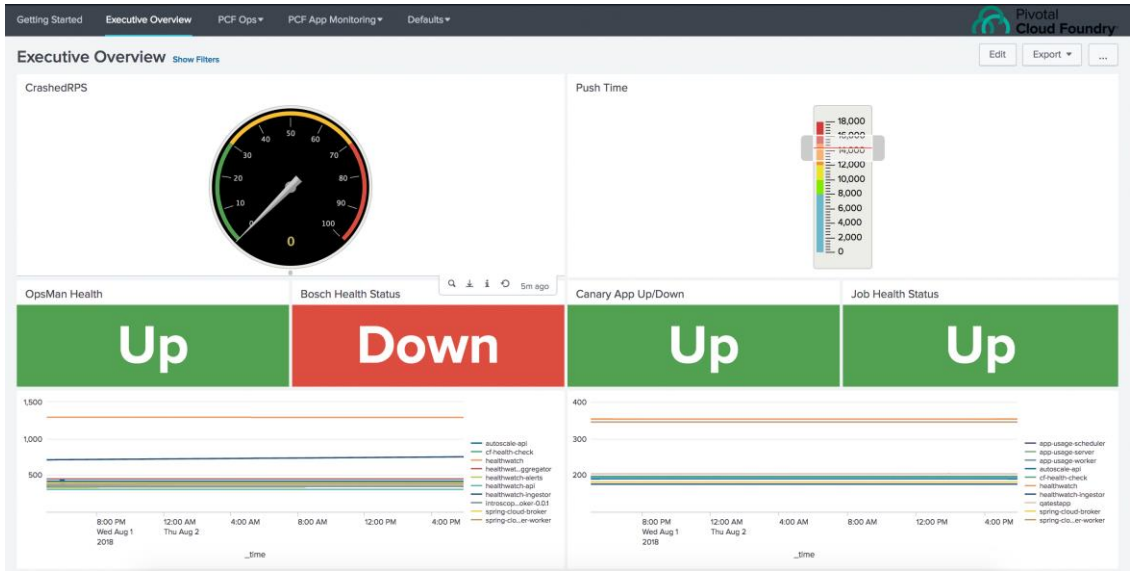
Getting Started

- Documentation
- Video Links
- Download Links
- Data Overview
  - Broken down by Event Type & Sourcetype



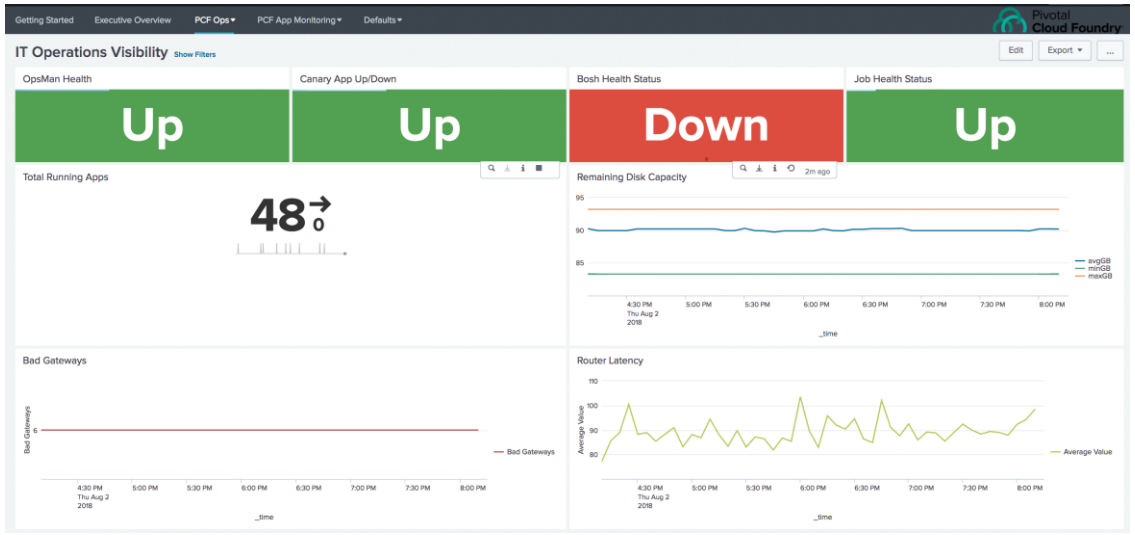
Executive Overview

- High-level metrics
- App level metrics for operators
- High-level metrics for App Developers



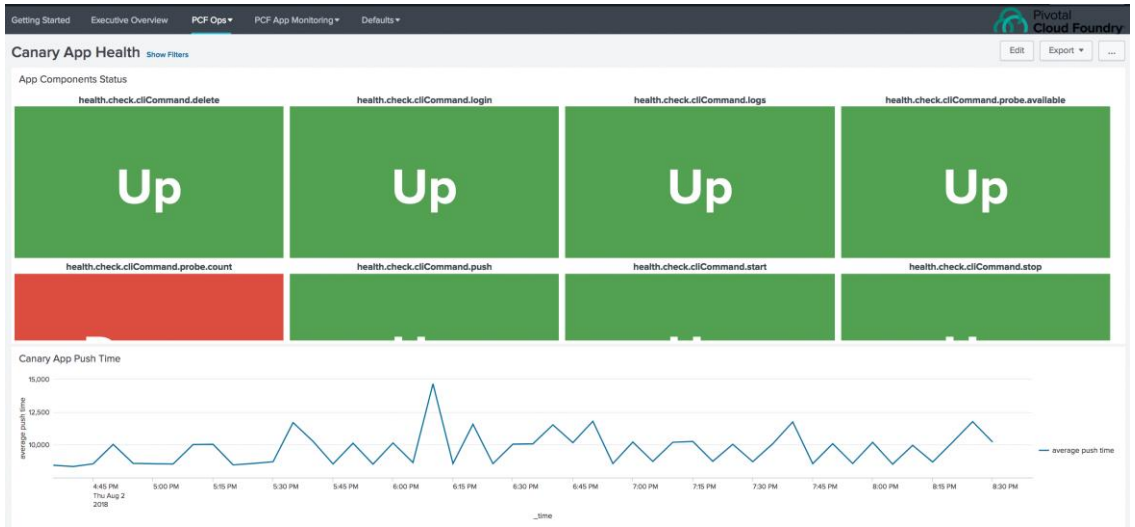
It Operations Visibility

- Same High-level metrics
- Designed to paint a picture for an Operations professional
- Consensus from customers of what matters from an IT Ops perspective



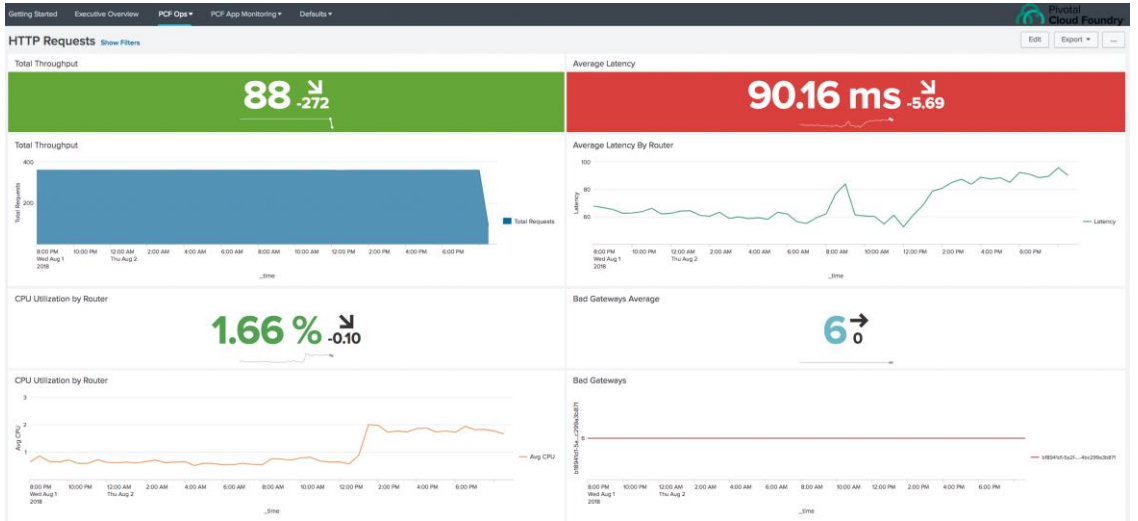
Canary App Health

- Canary Health App by component
  - Based on each component health rather than the high-level single metric
  - Detailed drill down on each metric
- Canary Push Time – average over time



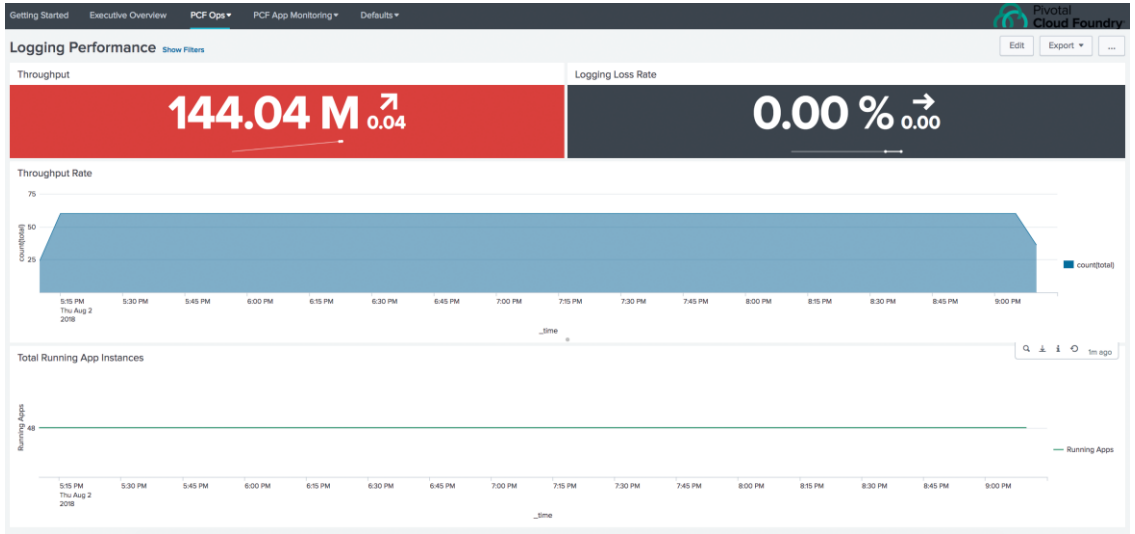
HTTP Requests

- Insights into the overall traffic flow over entire deployment
- Do you need to continue to scale the router?
- What trends are evident?
- Gorouters scale either horizontally or vertically



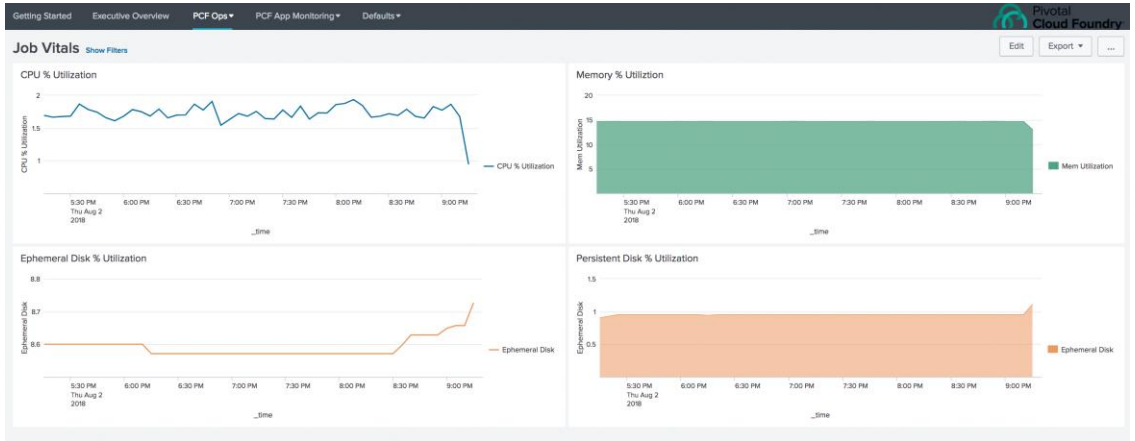
Logging Performance

- Overall Performance of Logging
- Enables you to understand the trends of your logging for example:
- Example:
  - Dropped messages can indicate that Dopplers are not processing messages fast enough and that Doppler instances need to be scaled



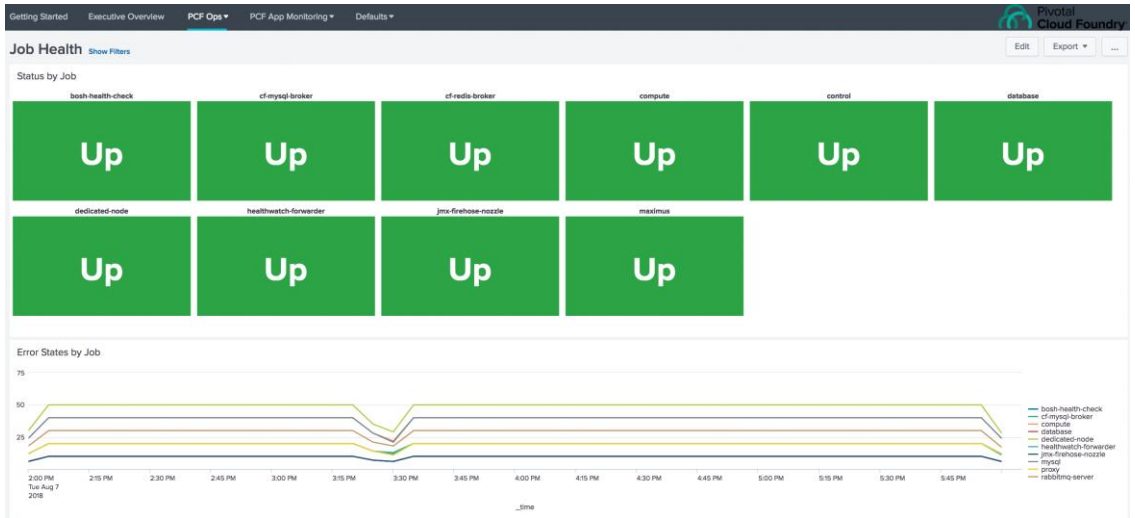
Job Vitals

- Metrics across foundation and with foundation drilldown
- High Level metrics
  - CPU, Disk Utilization, memory
- More metrics around Operational Health



Job Health

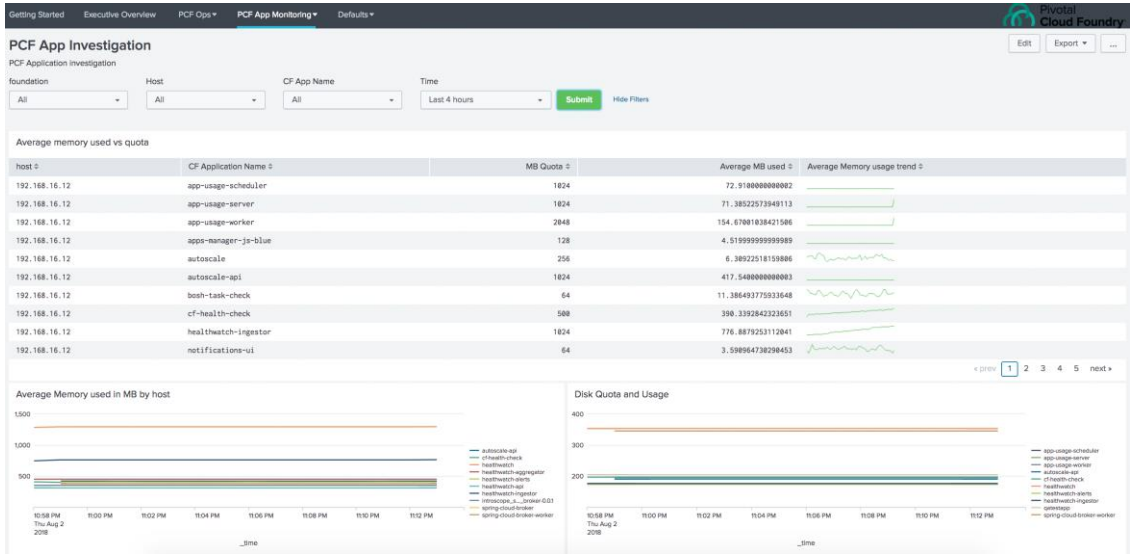
- Job Health provides you you with an understanding of job health – up or down
- In addition you can find Error states by job over time and…
- Error states by deployment



splunk> .conf18

PCF App Investigation

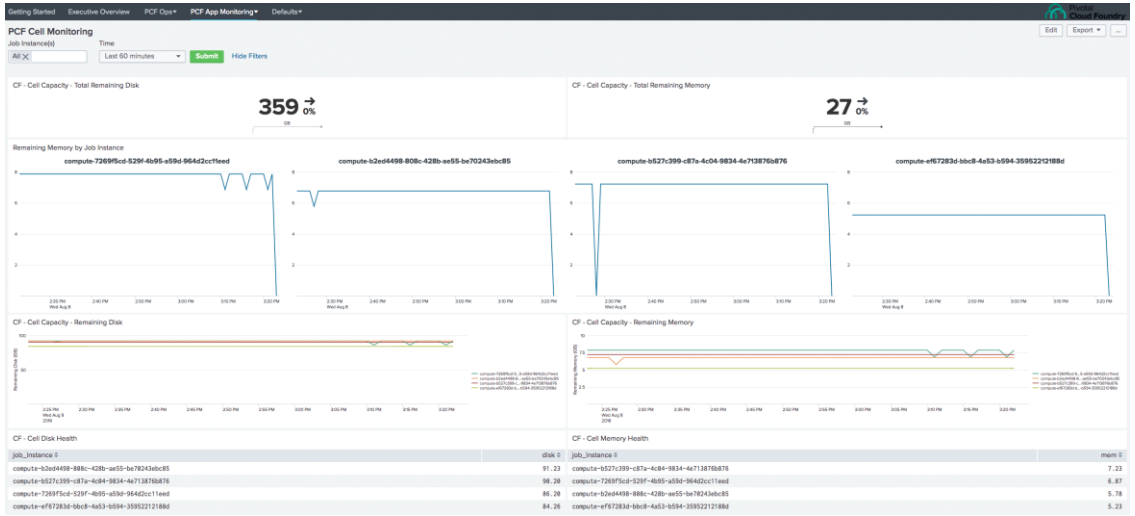
- Interactive Dashboard designed to allow you to filter down by:
  - CF App Name
  - Host
  - Foundation
- Then provides app level metrics such as:
  - Avg memory usage
  - Disk against Quota usage
  - Average Memory used by host



splunk> .conf18

PCF Cell Monitoring

- CF Cell capacity
  - Total Remaining Disk
  - Total Remaining Memory
- Memory utilization by Job Instance with Job drilldowns
- Memory and disk utilization breakdowns by:
  - Disk
  - Memory



Pivotal, PCF Firehose Nozzle & Splunk

splunk> .conf18

BP (Before PCF)

- Development time too long
  - Development time too high
- MTTR too high
- Two silo-ed development environments
- No way to have any insight past last 5 minutes

splunk> .conf18

Why PCF?

- Elastically Scale as the business requires
- Quick time to deploy
- Standardized deployment and testing
- Increased Operator efficiency
- Highly available architecture

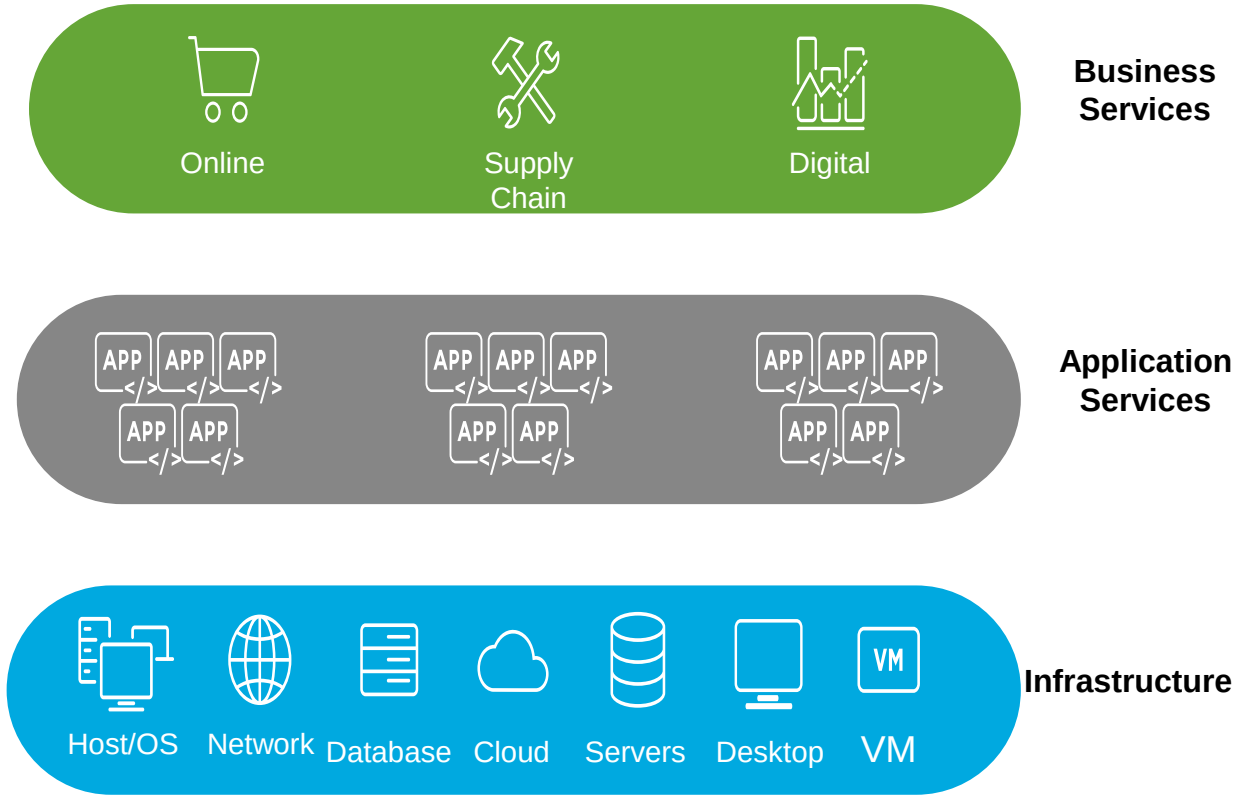
Splunk + PCF

- Splunk used for all Monitoring
- Splunk is the logical choice
- Will use to help achieve 4x9's
- PCF is the future platform for development
- App will play a critical role in that

splunk>

PCF + Splunk

- The main goal is a bottom up supportable process
- Not only inclusive of Infrastructure but App & Business Services
- A single source of truth both for development experience but operations as well
- A true dev-ops monitoring platform



Future Plans

Where do we go from here?

ITSI integration

Service-Centric views

Notable Events

MLTK Integration

Forecast future needs for the PCF platform

Predictive Analytics

Prevent downtime by using predictive analytics

splunk> .conf18

Thank You

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