

# Next Generation Ops at Standard Chartered Bank



**Shaun Norris**  
Global Head | Cloud Infrastructure  
Standard Chartered Bank

 @shaunnorris

# Shaun Norris



JPMORGAN CHASE & CO.



# IMAGINE NOT SEEING YOUR HEROES



Seeing  
is Believing

Standard  
Chartered



# 184

projects implemented across 37 countries



# 31

Innovation Fund projects to develop pioneering ideas

# 4

million sight-  
restoring  
surgeries  
supported

# 15

years of Seeing is  
Believing

# 318,833

health workers  
trained



# 167

million people reached through medical  
interventions, eye examinations, eye health  
education and training



# USD100 million

raised since 2003



We drive commerce and prosperity through our unique diversity

>150  
years old

>60  
countries

86,000  
employees

\$14,289m  
operating income (2017)

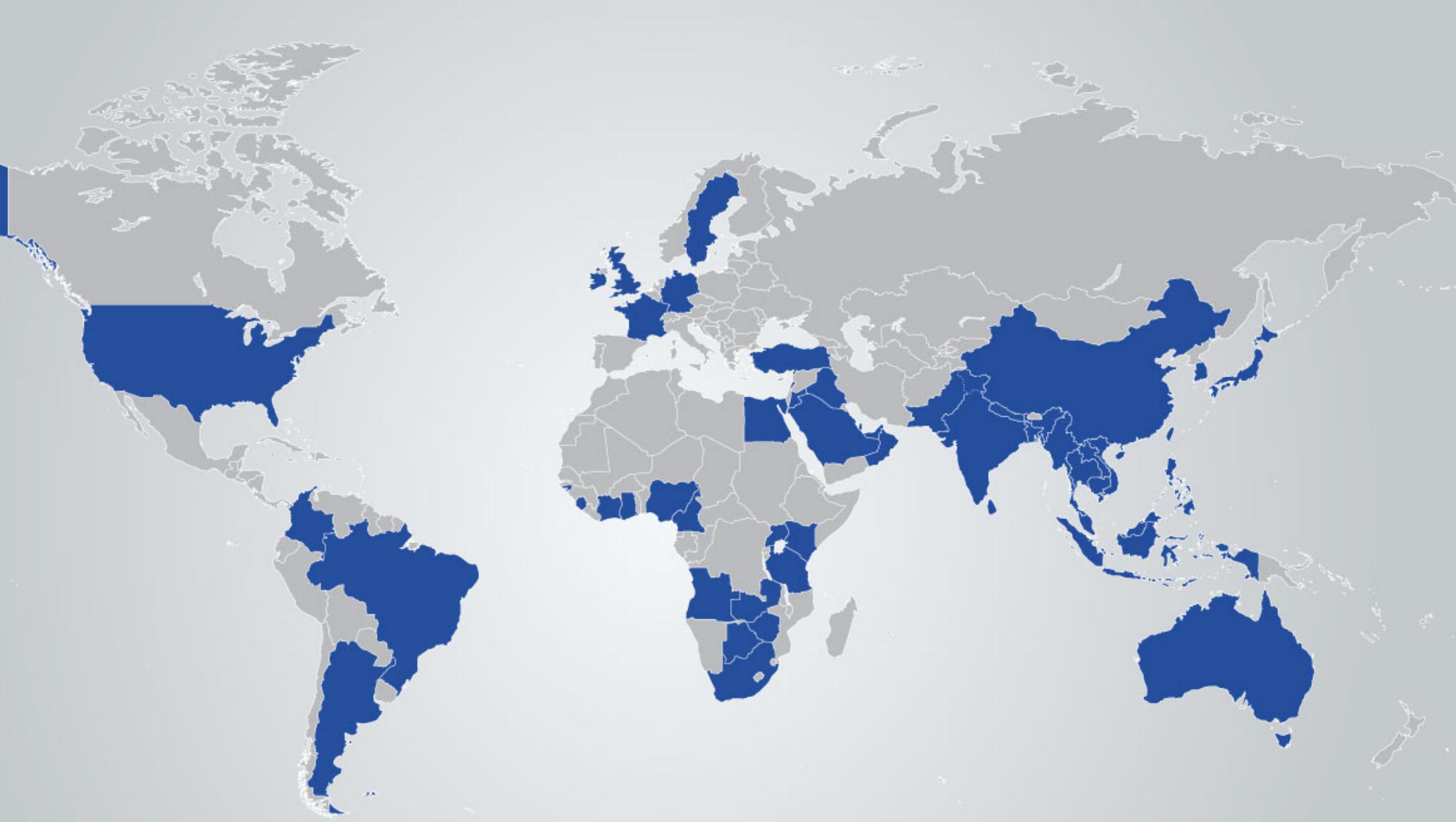
300,000  
business & commercial clients

9m  
individual clients

Chartered Bank's Royal Charter granted by Queen Victoria in 1853.

>1,000  
applications

Have printed Hong Kong banknotes since 1862

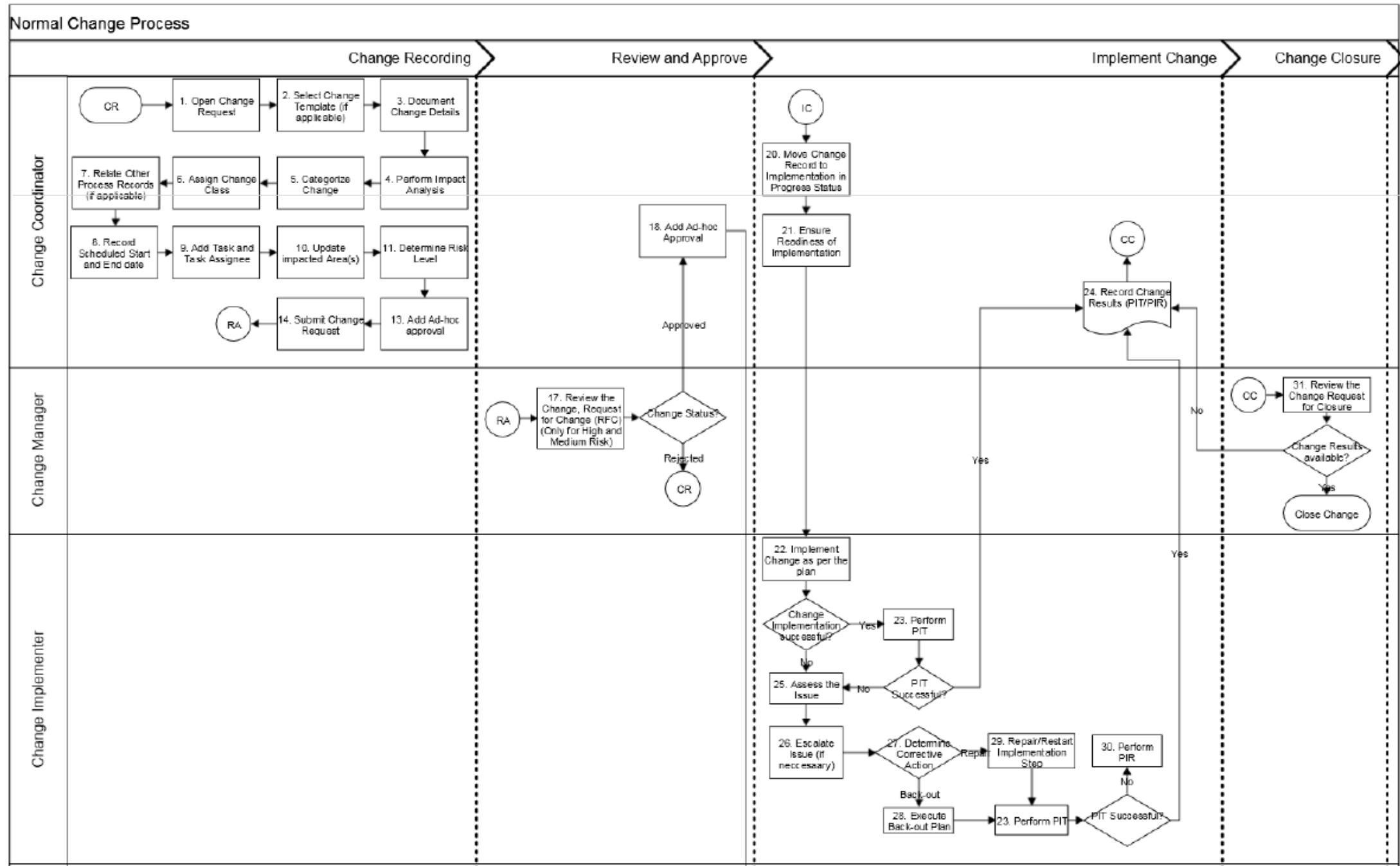




# Current Infrastructure Delivery

- Agile is a recent development in the organisation
- Risk & control management processes playing catch-up
- Servers still take weeks to provision





# During a typical incident

---

- 
- A yellow triangular warning sign with a black border and a large black exclamation mark in the center.
1. Business
  2. App Dev
  3. Major Incident Manager
  4. Problem Manager
  5. Application Support
  6. Service Management
  7. Database Support
  8. Unix Support
  9. Network Support
  10. Datacentre Support
  11. Managed Service Provider
  12. Security
  13. Country Technology

We already have one failed Cloud /  
DevOps transformation under our belt...

## Lessons learned

Bi-modal IT didn't work for us

Communication was opaque

Delivery was poor

Value realized was low

Costs were high

# DevOps and agile reputational issues

- Among operations, Agile is sometimes cynically viewed as “We want to operate without rules or constraints”
- For many, DevOps means “We’re using Jenkins”
- Some folks view agile as something that can be done before, or even without DevOps
- Agile project management is sometimes an excuse for non-delivery and missed commitments

# Reasons to be optimistic

DevOps Pipeline

Multiclouds

The last mile to Production

Pipeline Adoption

Rundeck

SRE Experiments

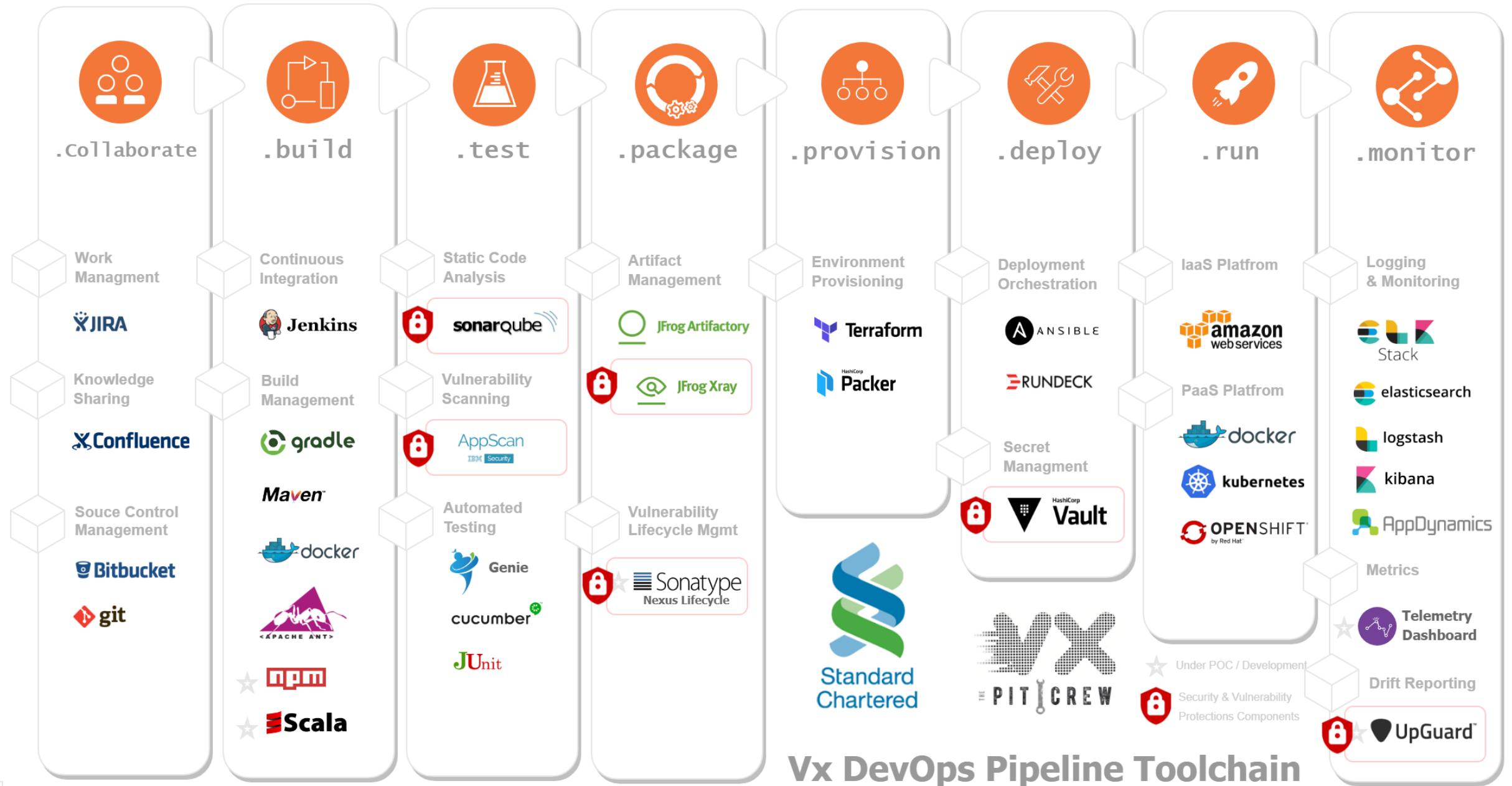
# Tenets and principles

Our management team adopted these earlier this year – at this point many are still aspirational!

- ✓ **Everything via API**
- ✓ **Everything via Patterns**
- ✓ **Everything as Code**
- ✓ **Everything via one Pipeline**
- ✓ **Everything as a Product / Service**
- ✓ **Immutable Infrastructure is the goal**
- ✓ **Everyone mapped to a Persona**
- ✓ **Strategy-based Pricing**
- ✓ **Security first, by design**
- ✓ **Eliminate reason for requests**
- ✓ **Open Source first**
- ✓ **Self-test, self-heal**
- ✓ **Declarative approach**
- ✓ **Zero-touch, Automate & orchestrate**

A photograph of a vast, bright white cumulus cloud formation against a clear, deep blue sky. The clouds are billowing and textured, appearing to float over a dark, silhouetted landscape at the bottom of the frame.

How > Where



# Paving the last mile to production



## The Challenge

We needed to orchestrate prod deployments that would run as an extension of the pipeline.

The desired outcome was to have **one click for a production deployment**



## Our Solution

We built two new integration & deployment microservices:

- **API driven and one click deployment with Rundeck**
- Can handle Ansible or Terraform
- Can deploy code to **legacy VMs, Openshift, and IaaS (AWS & Stratoscale)**
- Integrated with **JIRA and Remedy**
- Enforced automated controls around **security, controls and change management**
- Published **audit log** of all actions taken

Apps using this deployment mechanism (and DevOps pipeline) will be pre-approved for standard changes.

# Pipeline adoption so far...

>917,000 pipeline builds YTD

30% of Banks Apps on pipeline

4% Automated Deploy

1% cloud

IaaS

PaaS, CaaS, etc

Public



**kubernetes**  
(coming soonish)

Private  
(our own DCs)



 kubernetes  
(coming soon)

So far in 2018:

- >350,000 jobs executed across >400 applications
- Typically reduces TTR by 25 mins per incident when used interactively
- ~200 customer impacting incidents avoided by automated change deployment
- Projected to save ~28 people-years worth of manual work in 2019



# SRE Experiments

My colleague Venkat Raghavan has been running an experiment with SRE model in production support team handling cash apps for much of 2018



## Before SRE

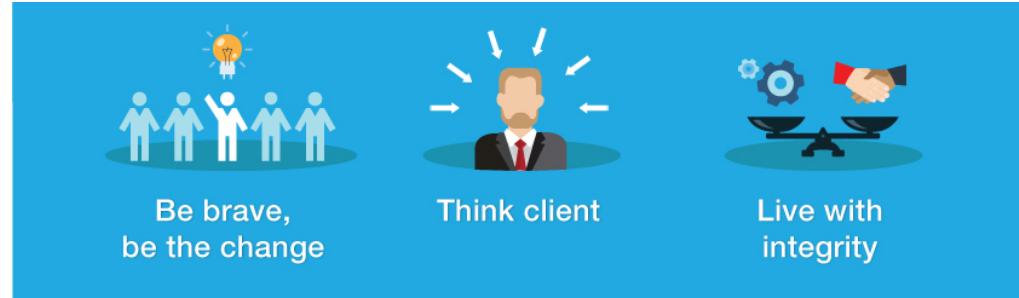
- Previously was a traditional L3 support model
- Large backlog of un(der)funded and un(der) prioritized work
- Regular priority clashes where ops-related work was left undone

## After SRE model experiment for several months

- 84% of backlog Items fixed and deployed to production
- 40% alert reduction
- 15% incident reduction
- Measurable improvements in system stability

# Our Valued Behaviours

**Do the right thing**



**Better together**



**Never settle**



# Work in Progress

## Processes

How to retool ops model for speed whilst complying with regulation and policy?

## Support model

How to support both our new platforms and the apps running on those platforms?

## SRE implementation

Still to determine how to implement and the model is not yet 'battle tested'?

## The business case

How does this impact the numbers of staff required to support legacy systems

# Things we've learned



- Start with tenets and principles
- Innovation often comes from unexpected sources
- Grassroots initiatives need senior backing eventually
- For cloud, 'How' matters more than 'where'
- Containers have much less friction than IaaS for new apps

# Help I'm looking for

- Feedback
- Kubernetes strategy – landscape changing so fast that any plan feels out of date by the time it's written down?
- Examples of how to extend our pipeline to handle data management – evolutionary DB design etc.
- Examples of compliance as code that we can emulate





@shaunnorris

Greatness has  
no finish line.

So we'll never settle for  
good enough.

