Moving bureaucracies toward modern cloud practices

Will Slack and Peter Burkholder, 18F

For FASTER Community of Practice

June 1, 2018

Agenda

1. Introductions
2. The cloud today
3. Strategies we suggest
4. A cloud.gov case study
5. A federalist case study

Introductions

Peter

- Joined GSA and 18F in August 2016
- Geophysicist turned IT modernization specialist
- Previously worked for IT automation vendors, healthcare start-ups, and major research labs, including NI

Will

- Joined GSA and 18F in Mar 2015
- Previously implemented enterprise hospital medical records systems and credit card/electronic payment processes for a large company

The cloud today

Cloud first

Lags with cloud adoption:

- In FY2016, $85B IT spend, only $2.6B on cloud
- In 2016, zero agencies at 15% cloud utilization

Progress:

- FedRAMP passed 100 authorized CSPs
- TIC modernization underway
- CoEs and USDA datacenter to cloud migration

Uneven implementation

- Naive lift-and-shift
- High spend
- Low utilization
- Re-implementing iron in the cloud

- Wide range of adoption statuses across the government, sometimes within agencies

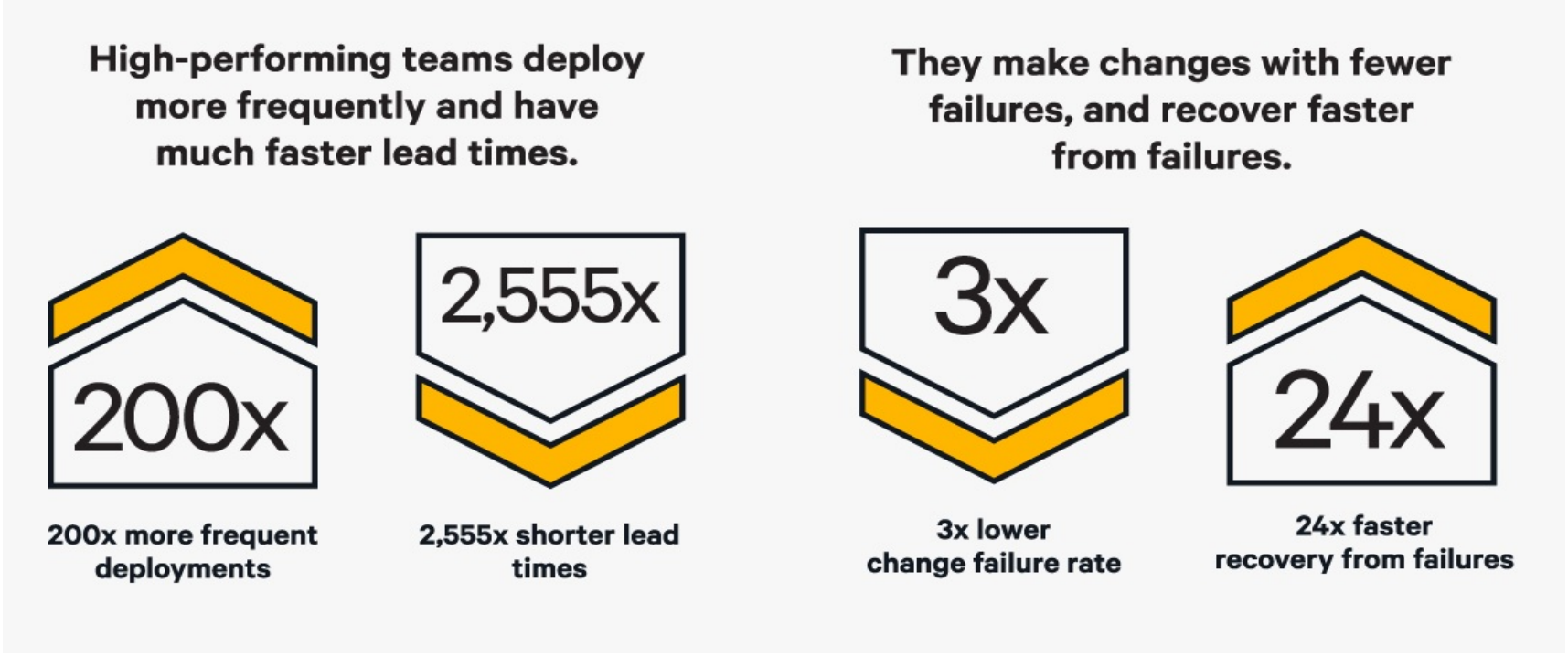
Missed opportunities

- Cost (capex v. opex, if not savings)
- Security
- Automation
- Mission-focused IT (lean/agile, innovation)

Mission-focused IT

Does IT matter?

- 2003: Nicholas Carr / HBR
  - But then: DevOps
- 2014: Gartner / Bimodal IT
  - But high-perf IT orgs still out-performed



50%

Less time spent remediating security issues

2.2X

More likely to recommend organization as a place to work

Puppet Labs 2016 State of DevOps Report

Metrics

Traditional IT measures

- Lines of Code
- Velocity
- Utilization
- MTBF
- Uptime

High-performance measures

- Release frequency
- Lead time (commit to release)
- % change fail rate
- MTTR
- SLOs

Strategies we suggest

Strategies we suggest

Start small, start core

- Don’t plan a full migration until you’ve done one and learned from it.
- Broadcast the progress through regular updates.
- Example: if you are running a program that involves tree permitting, you could start with the tree that has the lowest permit counts.

You are doing this well when:

- The team working on your first project is less than “two pizzas” in size.
- Your initial team is self-sufficient and has embedded skills in ops, security, and acquisition.
- People are excited about the project and following along unprompted.

Strategies we suggest

Live prototyping

- If your systems are public, your prototypes should be public.
- After your system is live, maintain a public staging system to test changes before pushing live.
- Deploy multiple versions of a system or feature for users and stakeholders to compare.

You are doing this well when:

- Your users default to trying out your staging environment.
- Your team and stakeholders are interested in looking at prototypes and giving feedback.

Strategies we suggest

Version control

- Habitually express everything with code and employ version control for that code.
- When you start, host locally, in your cloud, or via a SaaS solution.
- Peer review on all code changes should become normal.

You are doing this well when:

- Code updates are the *only* way that changes are made. (example: Terraform & DNS)
- People communicate over merge/pull requests.
- People take pride in their commit history.

Strategies we suggest

Be agile

- Create and iterate on working software instead of creating and following large, comprehensive plans.
- Agile is more expensive than waterfall because we expect and plan to change direction as we learn (instead of not learning until the end).
- Agile is something you are more than a specific set of practices.

You are doing this well when:

- Your team expects and appreciates whatever agile rituals and tooling you use (sprints, standups, backlogs).
- All of your team members speak up and contribute when determining commitments and priorities.

Strategies we suggest

Automated code deployment

- Release code day one.
- Demonstrate confidence in automation and testing, testing, testing.
- Demonstrate freedom from runbooks alongside portability of development to other team/vendors

You are doing this well when:

- You have a one button release process.
- Your deploy process includes linting and testing processes that confirm functionality as a part of deployment.

Strategies we suggest

Start with security

- Security should not be a exercise to pursue after development. Instead, pursue security and compliance as you develop.
- Create a team with ATO authorization personnel and your team that work together to achieve security and compliance.
- Create a “blameless postmortem” culture when there is a security issue.

You are doing this well when:

- Your cloud team and your compliance personnel are informally communicating or chatting outside of formal processes.
- Discussion of security concerns is normalized and doesn’t make team members feel nervous.

Strategies we suggest

Embrace an IaaS > PaaS > SaaS model

- Systems in the cloud don’t need to operate exactly like data center hosted systems; cloud allows a different architecture and shared services.
- Allow for authorization reuse.
- Your team can climb up the ATO mountain using a “cable car” to traverse most of the distance.

You are doing this well when:

- You are using multiple software solutions on the same platform and set of tooling.
- You don’t have to configure tooling multiple times.

Strategies we suggest

Agile procurement

- Use a challenges summary vs detailed technical requirements
- Offer smaller proposal sizes
- Plan to evaluate in weeks
- Set up contracts to reward delivery
- Early failure means work is less expensive

You are doing this well when:

- You've broken what would traditionally be a large, monolithic contract into several shorter-term, lower dollar amount contracts.
- A problem in one part of the project is isolated and can be addressed easily.

More information:

https://modularcontracting.18f.gov/

A case study: cloud.gov


tools → behavior
behavior → culture
Tools matter.
Your platform matters.

Platform is where you build, test and run:

- **Stack**: WebServer, AppServer, Database, Cache, Index
- **Environments**: (local), dev, test, stage, prod
- **User management**: admin, developers, auditors
- **Operations**: patch, logs, CDN, scanning, availability

Traditional platforms often add **friction.**

Platform as a Service can be your **best support for iterative work.**

Pre-built environment ready for deploying an application.

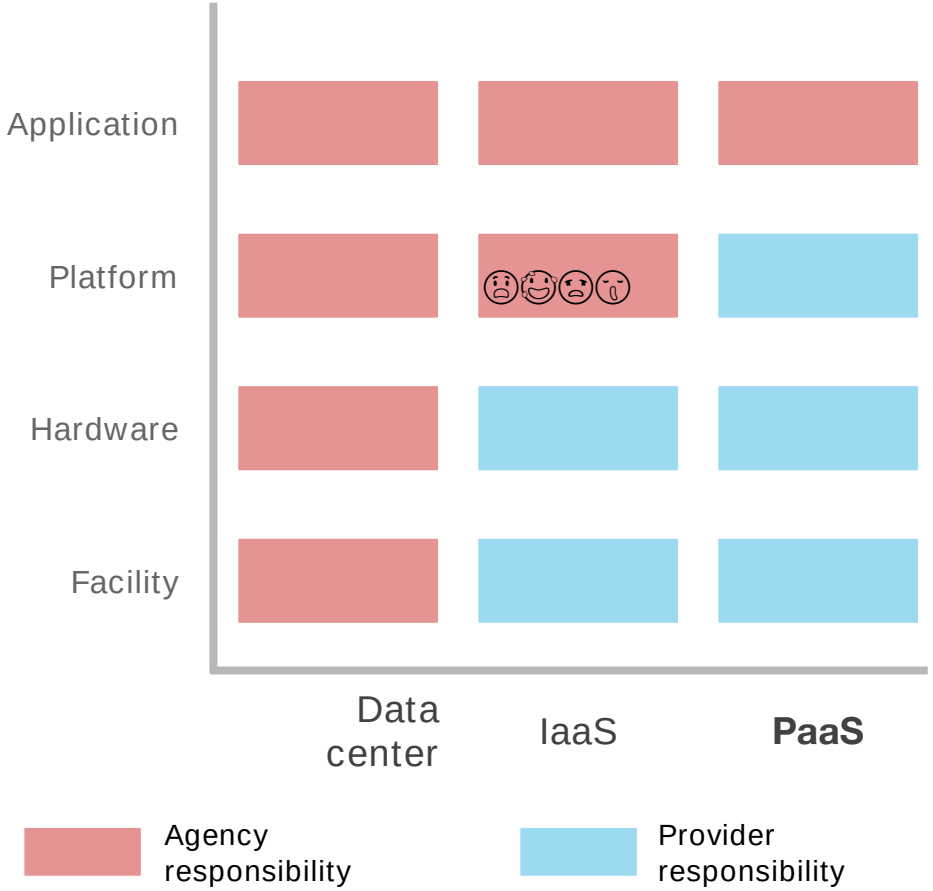
Developers can focus on mission needs.

Common technology resources are managed by an expert operations team:

- Operating system
- Databases
- Audit trails
- Authentication
- Authorization
- Load balancing
- Scaling
- Vulnerability scans
- Programming languages
- Automated updates

Reduce what you manage that's common across the government.

Platform as a Service



cloud.gov

cloud.gov is a **Platform as a Service** (PaaS).

It is based on **industry standard** Cloud Foundry and built on AWS GovCloud.

It has baked-in **federal security compliance**.

CLOUD.GOV

CLOUDFOUNDRY

AWS GovCloud (US)

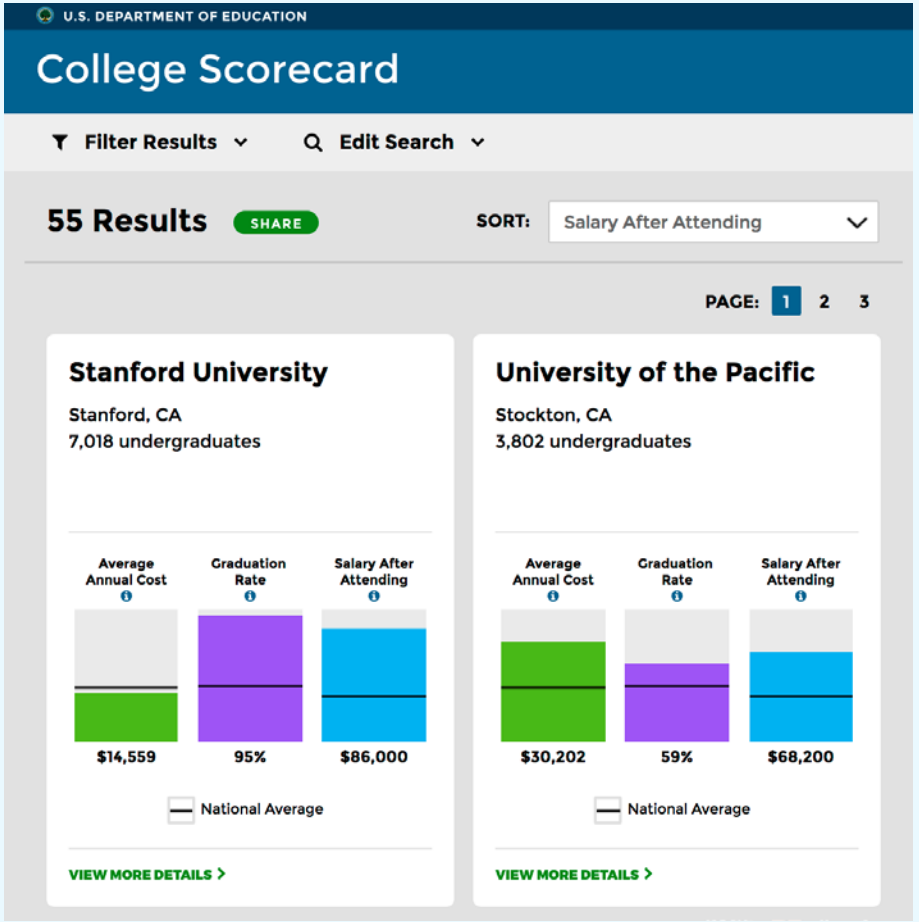
amazon web services

How it works

Your team brings **custom or COTS software**.

They use **self-service tools to configure services** for databases, storage, CDN, etc.

They **deploy** the application.



Federal Election Commission (fec.gov)

FEC spent $1.4 million annually on their data center.

With cloud.gov + AWS, initial estimates show **$1.2 million in savings annually.**

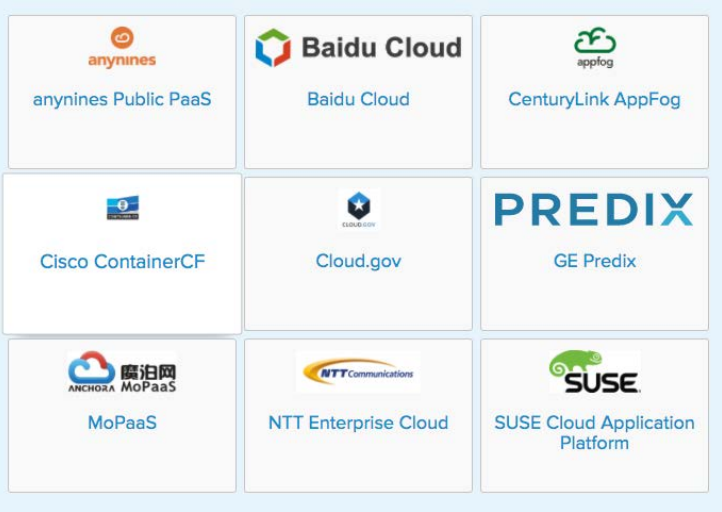
CLOUD.GOV

When I talked to a reporter I told them I'm sleeping well at night, even though it's a big project, because it's been tested for a year.

- *Deputy CIO, cloud.gov agency customer*

Cloud Foundry

- **Actively developed and updated**
  - Open source Platform as Service with many active contributors
- **Certified Provider**
- **Large community**
  - Thriving ecosystem with 400+ system integrators and consultants
- **Reduces vendor lock-in**
  - Multiple industry installations
  - Code supports multiple IaaS providers



Reducing vendor lock-in

Applications that work with cloud.gov **also work with industry Cloud Foundry providers.**

Expanding vendor choice

Third-party contractors can bid on how they build software, as most of the **operational concerns** have been offloaded.

Authorizations

FedRAMP JAB P-ATO Moderate

DISA DoD P-ATO Impact Level 2

You review authorizations, but **only assess your own application**.

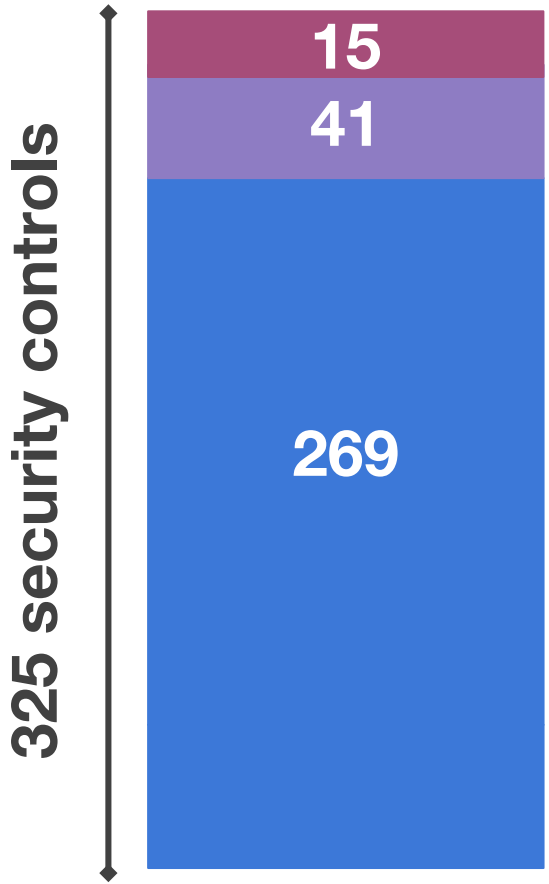
How cloud.gov reduces risk

Simplicity reduces mistakes. Plain-language configuration makes it harder to make mistakes.

cloud.gov implements the right defaults to reduce risk. Such as HTTPS and encryption at rest.

Reduce shadow IT. cloud.gov provides a modern self-service environment, so teams are less likely to use unapproved cloud infrastructure.

Many controls are handled by cloud.gov



15 customer

41 shared

269 handled or partially handled by cloud.gov
Some will be shared depending on your application

54 of these are fully inherited from AWS

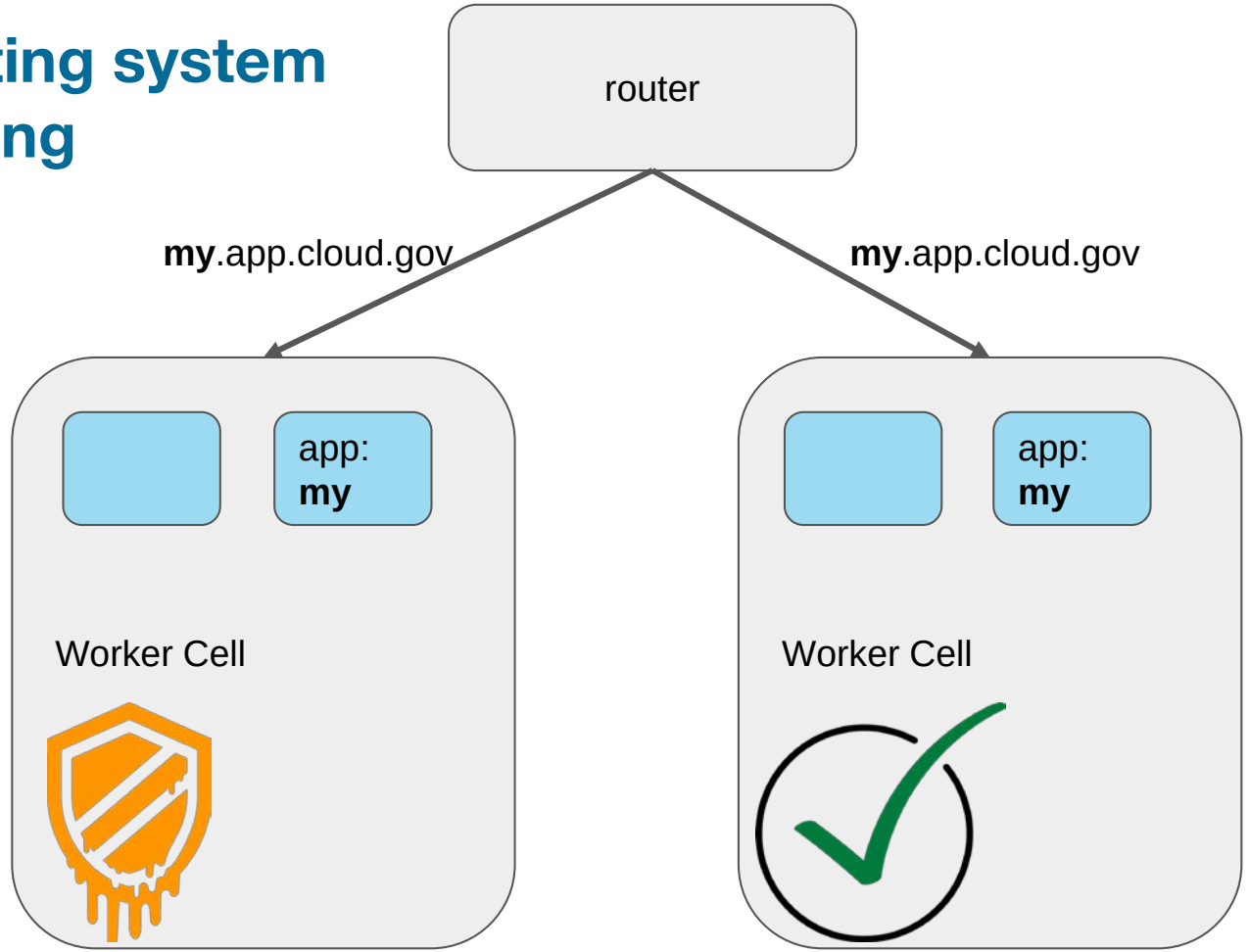
How we do security

- FedRAMP Joint Authorization Board Moderate P-ATO
  - Full, verified implementation of Moderate NIST 800-53 controls
  - **Annual third-party independent audit** of controls and penetration test
  - FedRAMP Continuous Monitoring
- Secure physical infrastructure
  - AWS GovCloud US (FedRAMP JAB High P-ATO)
- GSA operational maturity
  - Position of Public Trust background checks

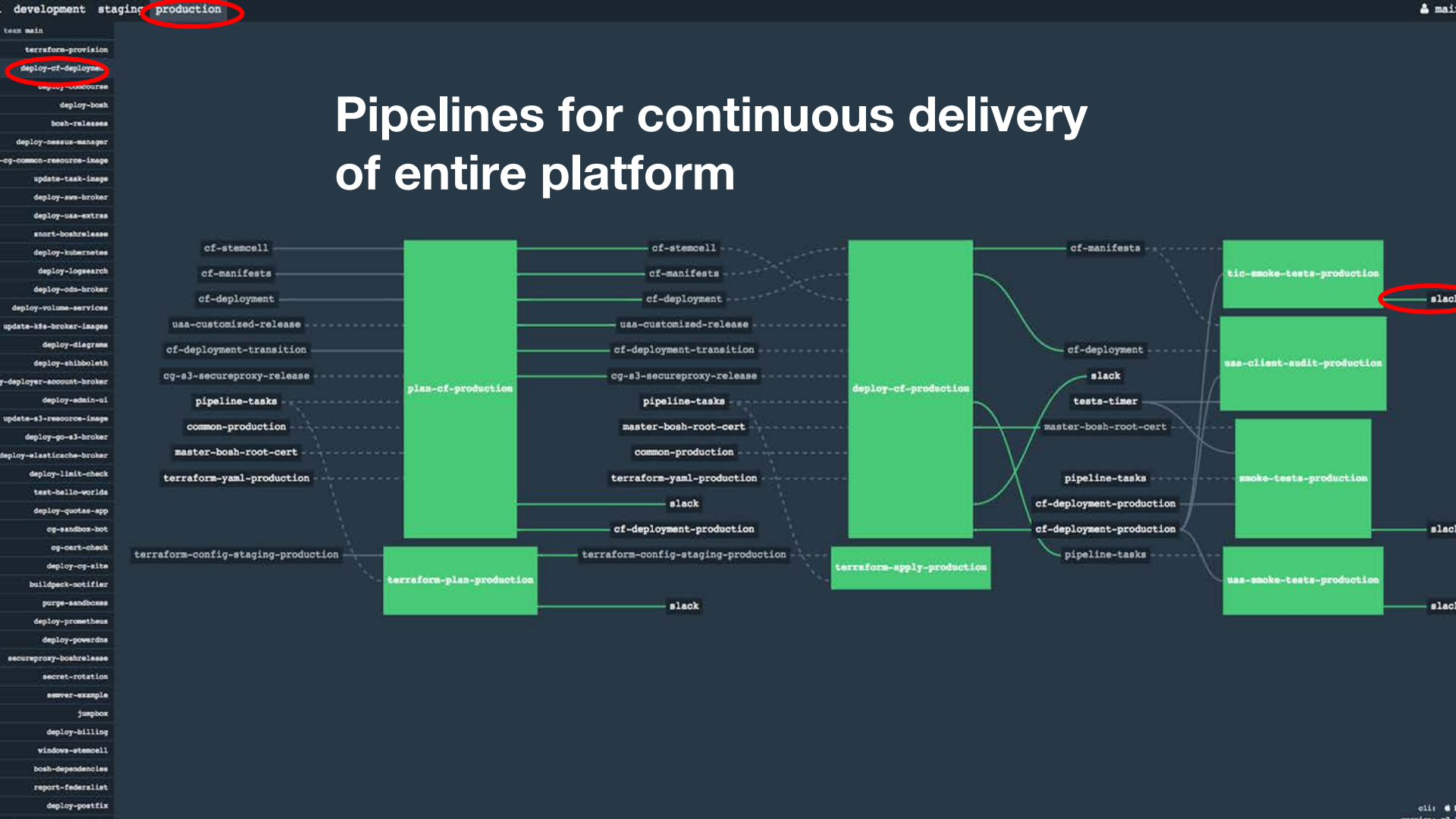
How we do security

- Architecture that isolates each customer system
- Fast, automated platform patching
  - Infrastructure as code (everything in configuration files)
  - Version control of all code and configuration
  - Continuous integration and continuous deployment
  - **Full deployment of upstream CVE patches in 12-24 hours**
  - We deploy updates several times a week
- We update without downtime or maintenance windows
  - Customer applications automatically restart, without downtime

Operating system patching



Pipelines for continuous delivery of entire platform



A case study: Federalist

Running a website in the government to inform the public can be extraordinarily difficult

1. Hosting (traffic surges)
2. Achieving authority to operate (ATO)
3. Maintaining compliance and mitigating new security issues
4. Ability to update content

Norm: 6+ months to launch, difficult updates

Many agencies have solved this problem with a CMS (Drupal, Wordpress)

- These can be great solutions, especially for large sites with many pages and a dedicated team.

- Other agencies or offices - like open data teams - can be more like a small business with a small web server, without dedicated web staff.

We made Federalist to support the government equivalent of that small business

- Federalist serves our fellow federal employees by expertly managing the backend and compliance work to launch and manage a website, allowing you to focus your expertise on your content.

- We do this by leveraging static web hosting.

We made Federalist to support teams across government and make it easy to publish

- Specifically: an individual program, office, campaign, or microagency that needs to launch and manage public web content or data.

- Examples: College Scorecard (ED), DotGov Data (GSA), BODs (DHS), itmodernization.cio.gov

Static sites in brief

- Instead of rendering web pages from a server on the fly, the pages are pre-built and stored for the public to access at incredibly low cost.

- Downsides: can't submit comments or send in information to a web server via a form (must use API or add a separate plugin like Disqus).

Federalist is built on compliant platforms

- **Infrastructure as a Service:** Amazon Web Services

- Static hosting on S3; very cost efficient
- CDN and HTTPS support from CloudFront

- **Platform as a Service:** cloud.gov

- cloud.gov brokers the AWS components; its FedRAMP JAB certification means less Federalist compliance work
- Architecture minimizes attack surface

Using federalist helps agencies adopt transformative practices

1

Live prototyping

Federalist builds out all of the versions of a site, allowing people to easily and quickly experiment, lowering QA costs.

2

Continuous integration and delivery

DevOps best practice to deploy changes as you make them instead of heavy “change control” processes

3

Open source

Partners leverage each other’s code, saving money and time.

Federalist also allowed sites to adopt features from outside sites

Using Federalist Helps Agencies Adopt Transformative Tools

1

Cloud hosting

Federalist some partners' first ever experience with cloud hosting (start small)

DHS had never put any of its sites onto modern cloud hosting (like AWS) until cyber.dhs.gov

2

Version control

All website content is managed via GitHub instead of via e-mailed word documents

Edit history easy to audit

3

IaaS > PaaS > SaaS

Federalist demonstrates the power of the model; cloud.gov manages the vast majority of our ATO controls (logging, containization, etc)

Being a Better Buyer from Industry

1

Vendors only have to focus on modern front end development

Using Federalist allows vendors to focus on what they are good at: designing quality sites.

2

Vendors don’t need FedRAMP’d hosting

Many vendors don’t have access to a FedRAMP’d hosting. To get that, the government pays a surcharge.

3

Allows for smaller contracts with small businesses

Using Federalist widens the vendor pool; single contractors can and have supported entire sites themselves.

Thank you!

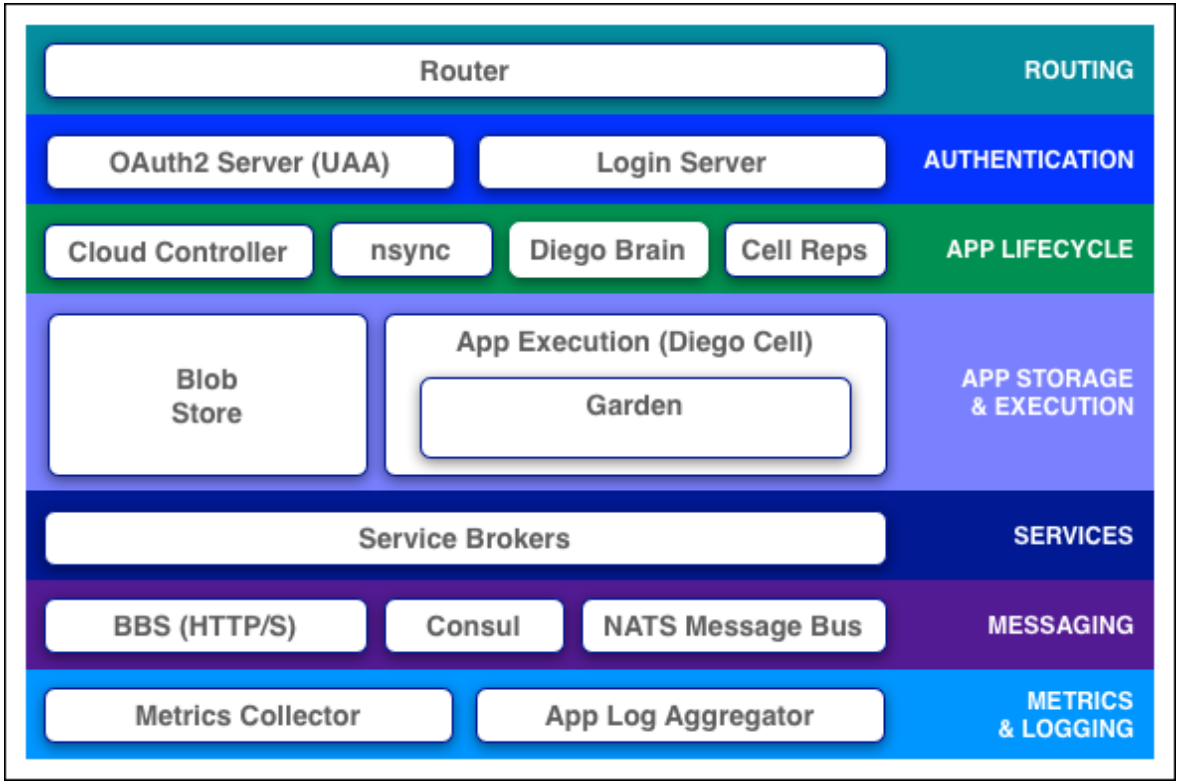
Questions?

Federalist:
Contact federalist-inquiries@gsa.gov
or william.slack@gsa.gov

Cloud.gov:
cloud-gov-inquriries@gsa.gov or
peter.burkholder@gsa.gov

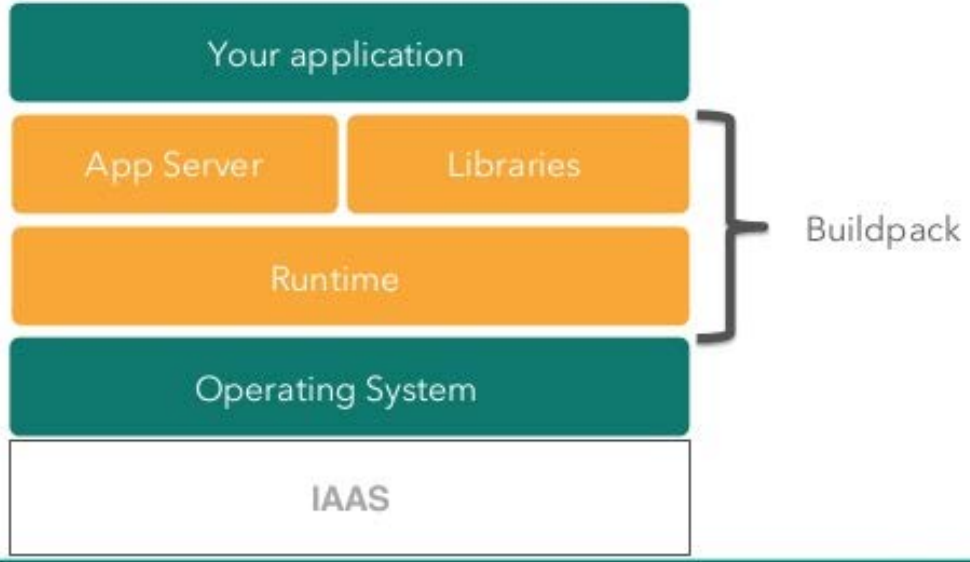
18F

Cloud Foundry system components



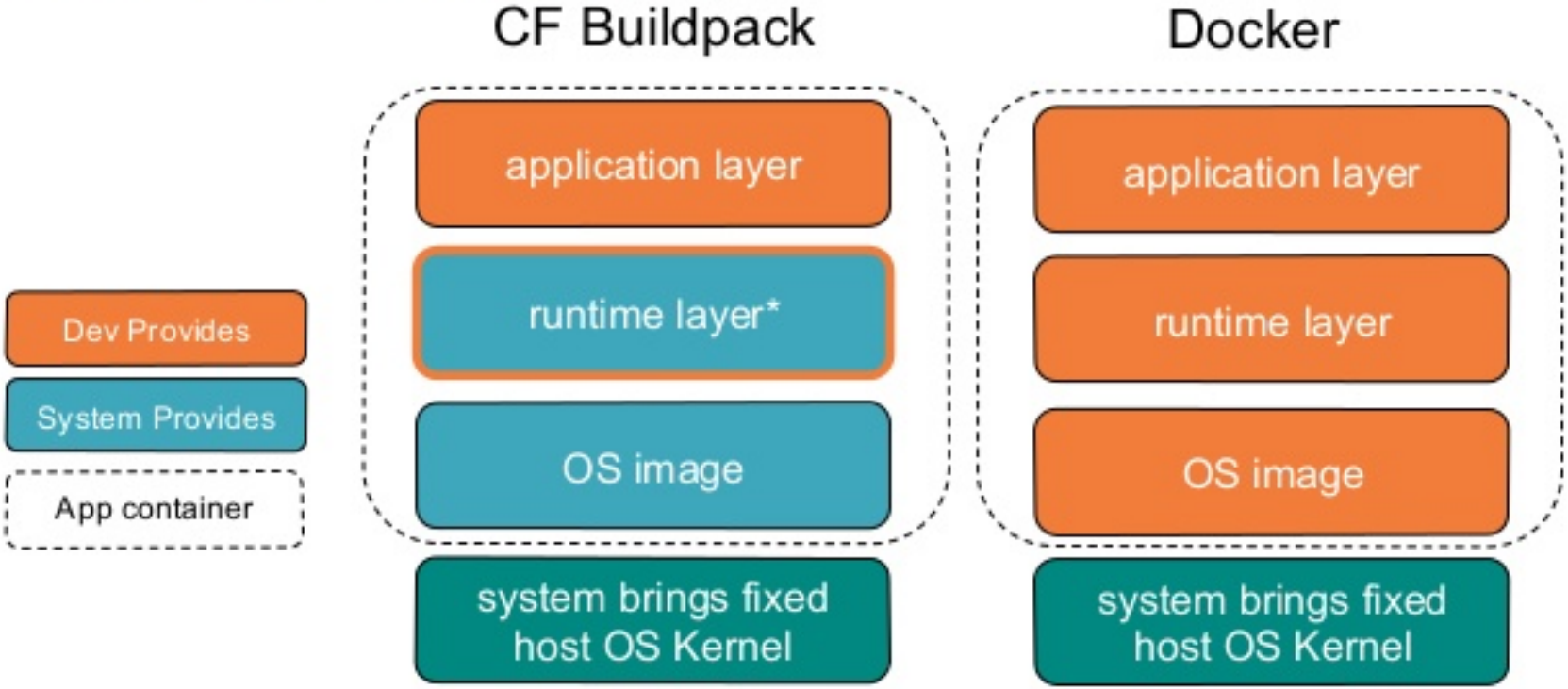
Buildpacks

Buildpacks are responsible for preparing the machine image for an application.



Pivotal

Comparison with Docker



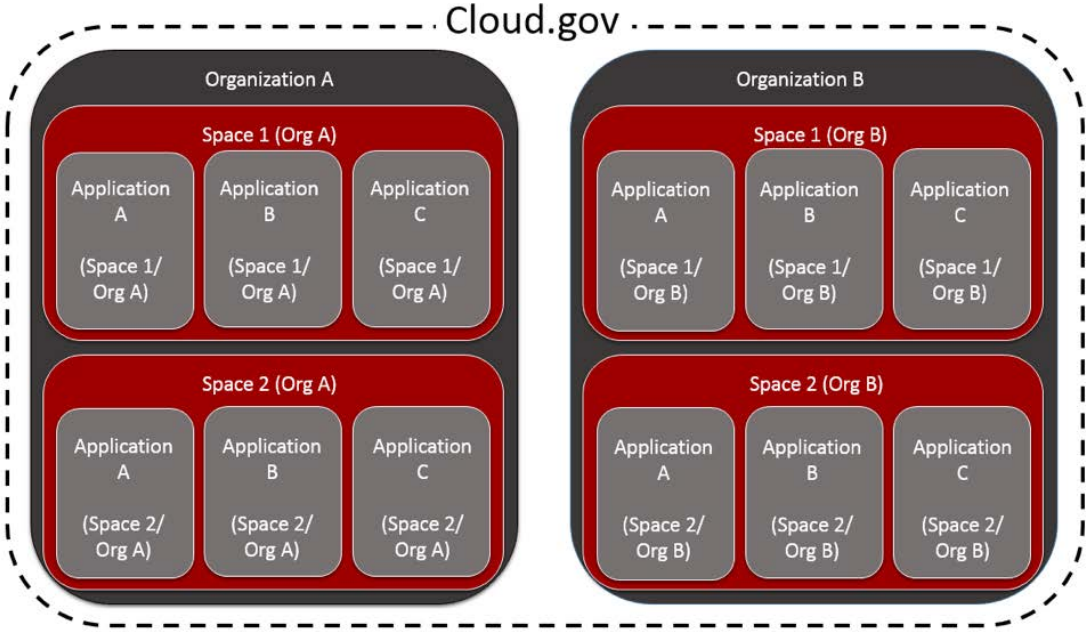
@ianhuston

* Devs may bring a custom buildpack

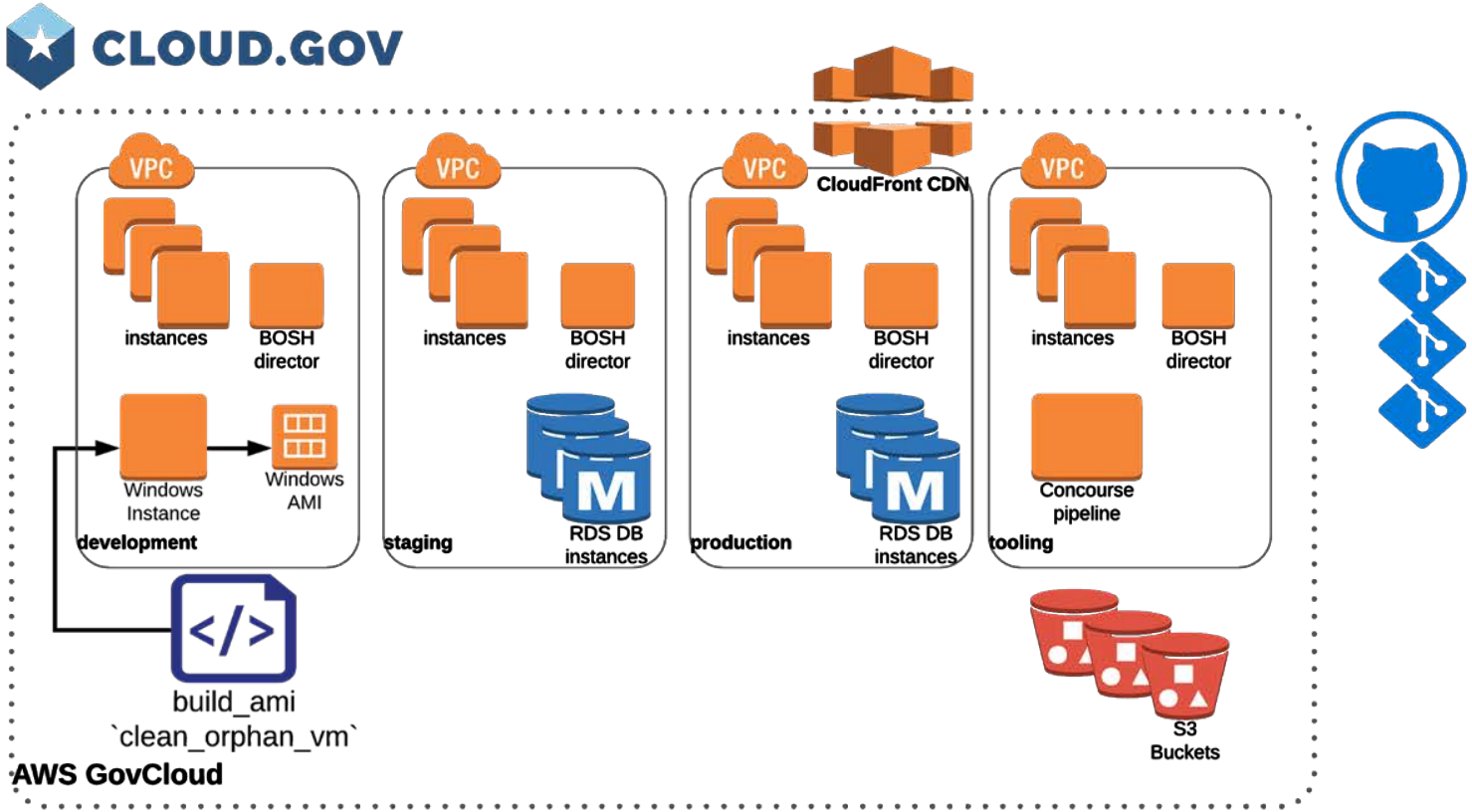
Pivotal

Platform as a Service

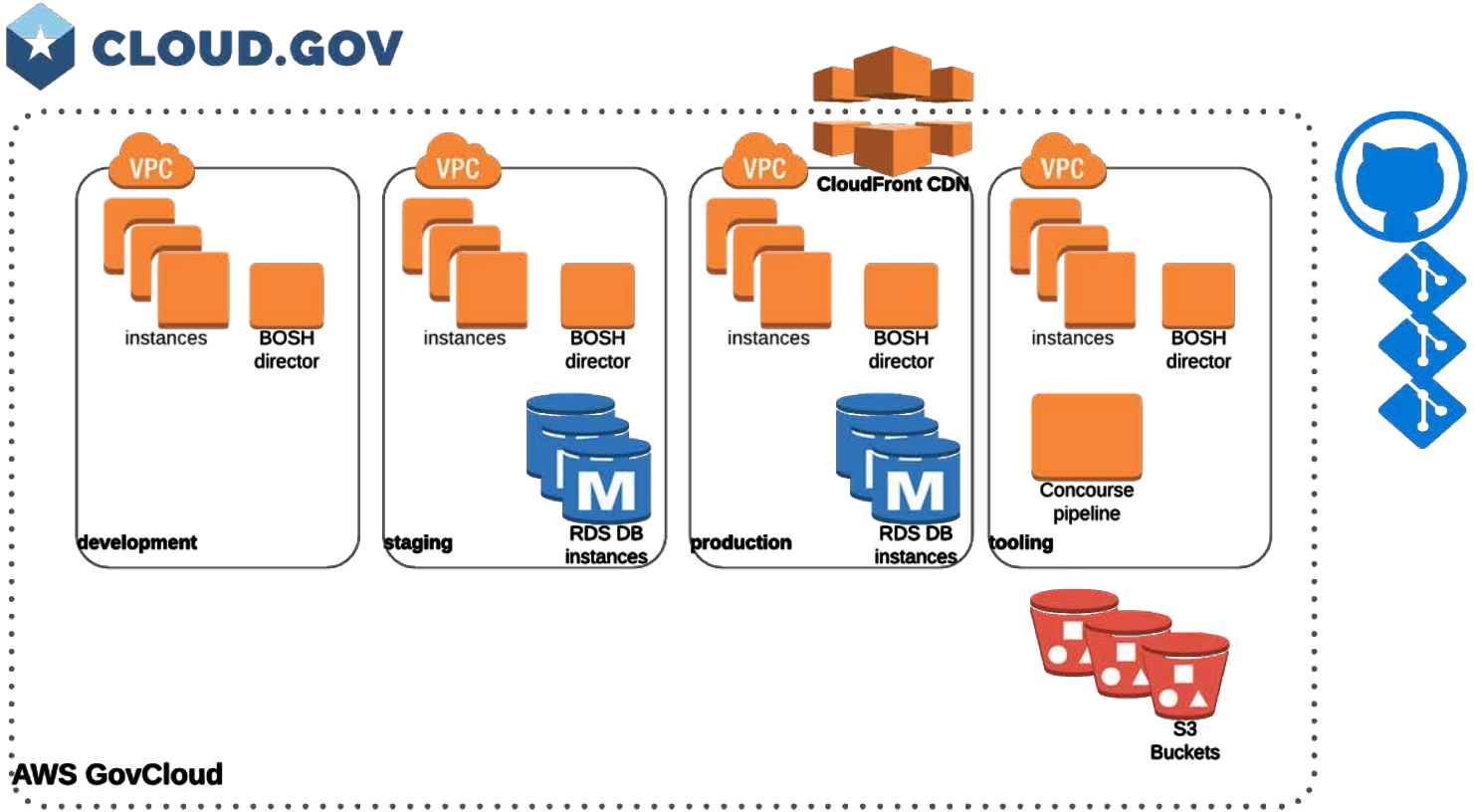
- Provides an app-hosting layer with managed back-end services
- Logical boundaries to isolate deployments for multiple customers
- Organization per customer
- Space per environment



Operations and Recovery



Operations and Recovery



Customer responsibility examples

- AC, IA: Control who has access to the system
  - **cloud.gov:** Control platform roles, provide role system for tenants
  - **Customer:** Control your tenant roles & any roles in your app
- AU: Ensure logs are stored
  - **cloud.gov:** Log the platform & provide logging infrastructure for apps
  - **Customer:** Configure your app to output logs
- RA: Scan for vulnerabilities
  - **cloud.gov:** Scan platform (including OS, databases, administrative apps)
  - **Customer:** Scan your app

18F

CLOUD.GOV

Peter Burkholder

GSA 18F

"Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Networking and Information Technology Research and Development Program."

The Networking and Information Technology Research and Development (NITRD) Program

Mailing Address: NCO/NITRD, 2415 Eisenhower Avenue, Alexandria, VA 22314

Physical Address: 490 L'Enfant Plaza SW, Suite 8001, Washington, DC 20024, USA Tel: 202-459-9674, Fax: 202-459-9673, Email: nco@nitrd.gov, Website: https://www.nitrd.gov