

Pivotal

Raytheon

Hyper-automated software platform

Jeff Glenn

jglenn@pivotal.io

What Raytheon Wants To Achieve

- Provide modern application platform service to business units
- Drive DevOps efficiencies, culture change through self-service & hyper-automation
- Embrace new cloud native patterns and constructs to speed business innovation
- Reduce ongoing operations costs and complexity
- Improve application security, resiliency, elasticity, manageability, speed of patching
- Support cloud choice and portability (vSphere, OpenStack, Azure, AWS, Google, etc)

What's Important to Raytheon's Desired Outcomes

- GEA - Kerry Finn - "A platform that supports continuous everything DevOps is a key theme of Raytheon's end to end SDLC value stream enablement that helps us move from a variety of SDLC methodologies (e.g. RAD, iterative, waterfall) towards one common agile business outcome based (e.g. "Deliver Mission/Business solutions with critical capabilities, 50% cheaper, 50% faster with 100% show stopper and critical defects resolved") approach ."
- RMS – Matt Kehret - move from current (25-30) IT Ticket based app deployment process, reduce or eliminate the IBM middleware management outsourcing service
- GBS – Eric Babjac – currently building homegrown platform to enable application and data microservices to allow for BU innovation
- IDS – Rick LaRowe team exploration of platform to optimize DevOps for C5i solutions

INTELLIGENCE, INFORMATION & SERVICES



Cyber



Environmental Intelligence



Training



Classified



Modernization
Through Sustainment

Current – Virtualization. Chef CI repository. Docker. Tripwire. People. Firefighting.

Vision - Make delivery a science instead of an art. Immutability allows move to predictable container based deployment. Constrains the attack surface & enhances security posture. Reduces the effort for certification/accreditation.

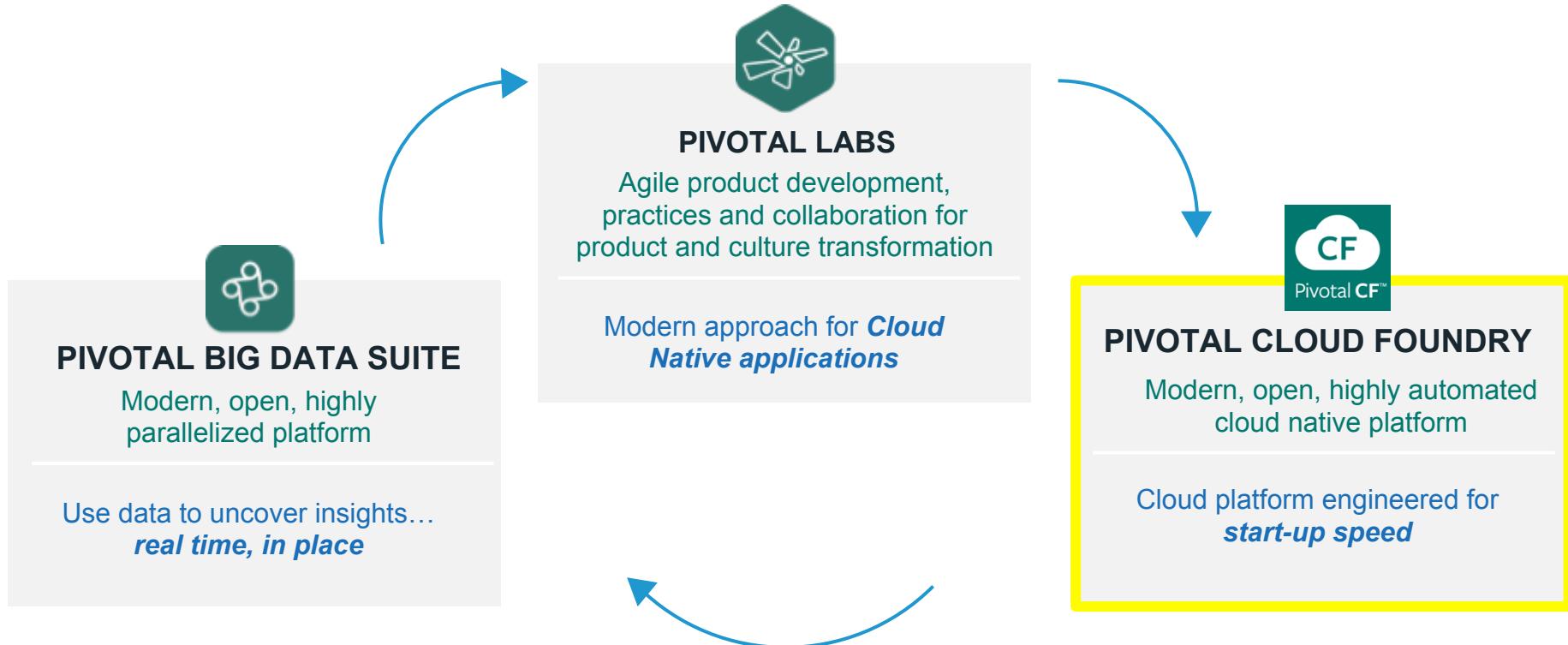
Challenges - Can't rearchitect, have to pull forward old tech, but use modern infrastructure. Accrediting authority certification process. High spend on security.

Solution - Common platform for deploying, operating, iterating applications to achieve: Immutability / Ease of configuration management / Upgradability / Configuration Abstraction / Automated test

Approach – Modern, industry standard Platform-as-a-Service. **COTS lift and shift build book**. And enable groups already all-in with cloud native paradigm.

Hands-on workshops – Boston August (Kerry Finn). Agenda on <http://raytheon.cfapps.io/> (raytheon/patriot)

Pivotal's Approach





**LOCKHEED
MARTIN**

“App in 10 weeks, instead of 9 months”

Humana.

“From 2 months for provisioning servers to delivering products in less than 5 weeks”

Allstate®

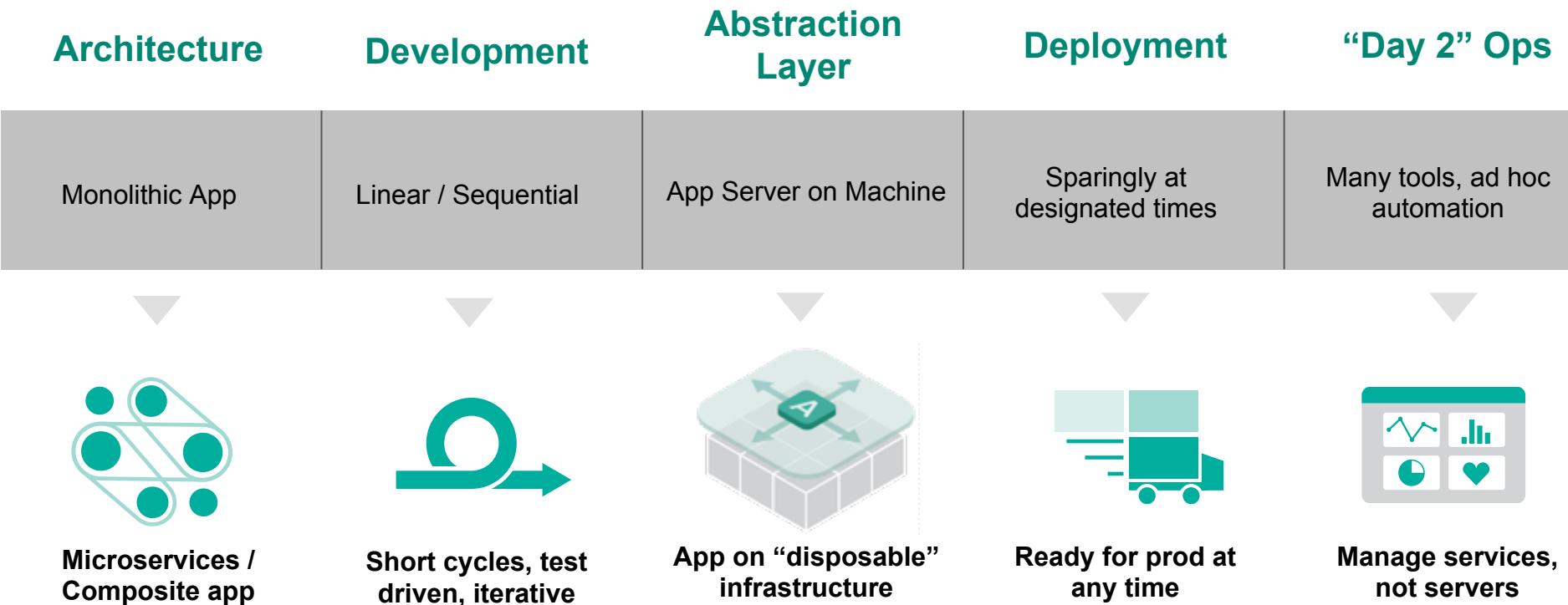
“Developers used to be 20% productive, now they’re 80–90% productive.”

 CLOUD.GOV

The Cloud.gov logo features a blue hexagonal icon with a white star in the center, followed by the word "CLOUD.GOV" in a lowercase, sans-serif font.

Deploys new applications and achieves production ATO-compliant levels in fewer than three days, when it used to take more than nine months.

Changing Model for Application Delivery





DEVELOPMENT

Multiple Languages



...

Microservices Support



Native

User Provided

Partner

Services Marketplace

OPERATIONS

App Deployment & Management



CI/CD Tools,
ID, Security

Availability



Health,
Metrics
Patching

Visibility & Administration



Apps &
Platform
Dashboards

VALUE LINE

Operating System



Container Orchestration



Cloud API



Google

AWS

Azure

VMW

Openstack

Expected outcome for Raytheon

- Developer and IT Operation Productivity
 - Average Developer is able to Provision and Install 50% faster
 - Average IT Operator can Configure, Deploy, Monitor and Scale App Lifecycle 90% faster.
 - Support multiple development languages
- Software License Savings
 - Based on Open Source platform and designed to avoid vendor lock-in
 - Single platform removes the need for complex, multi-vendor contracts
 - Future-Proof your IT environment
- Infrastructure Utilization
 - App-centric instead of infra-centric approach
 - Higher packing density equals better workload utilization
 - Spend time optimizing for business needs, instead of configuration management

| Customer Data | |
|---|--|
| Number of Customer Applications | |
| Number of App Instances per Customer App | |
| Number of IT Ops + IT Staff per Customer App | |
| Number of Developers per Customer App | |
| Number of software Releases per app/year | |
| Number of OS, DB, APP, WEb Server & Middleware upgrades per year | |
| Number of OS, DB, APP, WEb Server & Middleware critical patches and security fixes per year | |
| Number of times scaled per year | |
| Number of Apps per VM | |
| VM Medium (monthly) | |
| VM Large (monthly) | |
| PCF Pricing | |
| PCF Application Instance Cost | |
| PCF Foundation Cost | |
| No. PCF Foundations | |
| TOTAL ROI | |
| Cost of PCF | |
| Developer/Operator Productivity Savings | |
| Software License and IaaS savings | |
| Total ROI | |

Pivotal track record of success



Honeywell



comcast



Google



Pivotal



Air Force Research Laboratory Makes Leap to Transform Department of Defense Software Development and Deployment



Pivotal Solution includes: Pivotal Cloud Foundry

Challenge:

- Build cloud native platforms and provide them as a service to U.S. Department of Defense entities.
- Lack of its own development arm.
- DOD IT services is unreliable and has expensive, very long deployment cycles.
- Desire to drastically transform how the DOD runs software.

Solution:

- Introduce the concept of cloud native platform development and deployment to Department of Defense entities.
- Offer platform to the broader DOD community and use profit to drive research.
- Customers will build apps on top of PCF. One example is an intelligence system that will pull technological and scientific data in multiple languages, index it and making it searchable.

Lockheed Martin Re-Platforms App Dev & Delivery

<https://www.youtube.com/watch?v=CvpAqflUj0o>



Pivotal Solution includes: Pivotal Cloud Foundry, HAWQ and Pivotal Labs

Challenge:

- Manage legacy (.Net and J2EE) and create new framework to allow them to get through cycle time faster and effectively
- Increase efficiencies with tools to improve project management and reduce bureaucracy
- Envision the ability to have data insight and analytics

Solution:

- First step in transformation towards cloud services and cloud delivery
- Develop initial web application around tracking projection with project managers and customer using PaaS
- Leveraging HAWQ to help deliver software applications that are data and analytics driven

DIY Cloud Native Platform Reference Architecture

| Infrastructure | | | | Visualization & Reporting | | | Release Management | | |
|--|--|---|--|--|---|---|---|--|---|
| Capability | Description | Solution | | Capability | Description | Solution | Capability | Description | Priority |
| Multi-Region Deployment | Deployment across multiple, disparate geographies; i.e. east and west coasts | AWS us-east-1  AWS us-west-2  | | Metrics Collection | Collection of time series metrics for alerting and reporting | Sensu  | Security / Account Abstraction & Portal Access | Single unified access to all CSP accounts | Custom  |
| Multi-AZ Deployment | Deployment across multiple datacenters within a geography; i.e. | AWS AZs   | | Metrics Storage | Storage of time series metrics for alerting and reporting | Graphite  | Security / RBAC Abstraction | Role-based access by product team to CSP accounts | Custom  |
| Network Circuits / VPN | Dedicated network connectivity between CSP and Datacenters | AWS DirectConnect   | | Dashboards | Single view of disparate operational health and metrics | Grafana  | Execution of Orchestration Templates | Automated provisioning of stacks based off of orchestration templates | Custom  |
| Private Networks / SDN | Implementation of dedicated, private RFC 1918 subnets within CSP | AWS VPC   | | Logging | Aggregated collection and visualization of log data | ELK   | Versioning and Bundling of Deployment Artifacts (Chef, CFN, Binaries) | Bundling and versioning of all "full-stack" deployment artifacts. | Custom  |
| Routing & Segmentation of Private Networks | Routing and integration of traffic between multiple private networks | AWS VPC   | | Financial Reporting | Detailed financial reporting by product and team | Netflix ICE  | Application Deployment Abstraction of CSP | Parameterized deployment for applications to deploy consistently across multiple cloud providers | Custom  |
| Firewall / Segmentation | Automated provisioning and configuration of firewall rules | AWS VPC & Sec. Groups   | | Alerts | Routing of events and notifications by on-call schedule | Pager Duty  | Automated Blue-Green Traffic Switching | Automated shifting of traffic from one application release to another in a full-stack model | Custom  |
| DNS | DNS switching capability to bridge external/public DNS and internal DNS | AWS Route 53  BIND DNS  | | Continuous Delivery | | | Service Catalog of Approved Versions of Deployment Artifacts | Catalog of tested and approved production deployment artifacts | Custom  |
| Outbound Network Traffic | Ability for compute instances to initiate network connections to the Internet | iptables  | | CI Orchestration | Continuous build, test, and integration of application and infrastructure code | Jenkins  | Deployment Policy Enforcement (Network, Capacity, etc...) | Enforcement of placement of application stacks based on policy: compliance, capacity, etc... | Custom  |
| Configuration Management (Run-Time) | Configuration management of compute instances in production runtime | Chef  | | Infrastructure Orchestration | Orchestrated provisioning and configuration of stack-specific infra. | Cloud Formation  Terraform  | DNS Provisioning | Automated provisioning of DNS records for application components and stacks | Custom  |
| Artifact Repository (Binaries) | Repository of successfully tested application binaries | Artifactory   | | Configuration Management (Image Build) | Continuous build, test, and integration of cloud instance images by application component | Packer  | Security | | |
| Service Discovery | Ability for services to register themselves, discover each other, & integrate autonomously | Consul  | | Artifact Repository (Infracode) | Repository of successfully tested infrastructure code. | Chef Supermarket   | Middleware | | |
| Container Format | Standardized format for packaging and delivery of Linux containers | Docker  | | Network Security Monitoring | Automated monitoring and alerting of internal and external network for malicious traffic | BRO  | HTTP Proxy | Reverse proxy and routing to backend micro services and applications | Nginx  |
| Container Scheduler | Automated scheduling, deployment, and recovery of Linux containers | Kubernetes and/or ECS  | | Secrets Management | Automated creation, management, and expiration of security "secrets." | Vault  | Java Container | Standard container for running Spring-based Java applications | Tomcat  |
| Data Platform | | | | Vulnerability Audit | Proactive scanning for threats and vulnerabilities | Tenable  | Object Cache | In-memory object cache | Memcached  |
| Geo-Replicated Messaging | High-throughput, distributed messaging platform replicated across AZs/Regions | Kafka   | | Incident Response | Collection of forensic data in the event of a security event or compromise | Google Rapid Response   | Document Store | High performance No-SQL document store | MongoDB  |
| Geo-Replicated NoSQL | Distributed NoSQL key/value store replicated across AZs/Regions | Cassandra  | | CSP Management Console Audit | Automated security auditing of CSP admin console | Evident.io  | Messaging | Traditional asynchronous messaging | RabbitMQ  |

Same Reference Arch with Cloud Foundry

| Infrastructure | | | Visualization & Reporting | | | Release Management | | |
|--|--|---|--|--|---|--|--|--|
| Capability | Description | Solution | Capability | Description | Solution | Capability | Description | Priority |
| Multi-Region Deployment | Deployment across multiple, disparate geographies; i.e. east and west coasts | Pivotal Ops Manager  | Metrics Collection | Collection of time series metrics for alerting and reporting | Pivotal Ops Metrics  | Security / Account Abstraction & Portal Access | Single unified access to all CSP accounts | Elastic Runtime Apps Manager  |
| Multi-AZ Deployment | Deployment across multiple datacenters within a geography; i.e. | Pivotal Ops Manager  | Metrics Storage | Storage of time series metrics for alerting and reporting | Pivotal Ops Metrics  | Security / RBAC Abstraction | Role-based access by product team to CSP accounts | Elastic Runtime & OpsManager  |
| Network Circuits / VPN | Dedicated network connectivity between CSP and Datacenters | AWS DirectConnect   | Dashboards | Single view of disparate operational health and metrics | Pivotal PCF Metrix  | Execution of Orchestration | Automated provisioning of stacks based off of orchestration templates | Elastic Runtime and BOSH  |
| Private Networks / SDN | Implementation of dedicated, private RFC 1918 subnets within CSP | Pivotal Ops Manager  | Logging | Aggregated collection and visualization of log data | Pivotal PCF Metrix  | Versioning and Bundling of Deployment Artifacts | Bundling and versioning of all "full-stack" deployment artifacts. | Elastic Runtime and BOSH  |
| Routing & Segmentation of Private Networks | Routing and integration of traffic between multiple private networks | Pivotal Ops Manager  | Financial Reporting | Detailed financial reporting by product and team | Netflix ICE  | Application Deployment Abstraction of CSP | Parameterized deployment for applications to deploy consistently across multiple cloud providers | Elastic Runtime and BOSH  |
| Firewall / Segmentation | Automated provisioning and configuration of firewall rules | Application Security Groups  | Alerts | Routing of events and notifications by on-call schedule | Pager Duty  | Automated Blue-Green Traffic Switching | Automated shifting of traffic from one application release to another in a full-stack model | Elastic Runtime (go)Router Routes  |
| DNS | DNS switching capability to bridge external/public DNS and internal DNS | Elastic Runtime (go)Router  | Continuous Delivery | | | Service Catalog of Approved Versions of Deployment Artifacts | Catalog of tested and approved production deployment artifacts | Elastic Runtime & Marketplace  |
| Outbound Network Traffic | Ability for compute instances to initiate network connections to the Internet | Elastic Runtime (go)Router  | CI Orchestration | Continuous build, test, and integration of application and infrastructure code | Jenkins Enterprise for PCF  | Deployment Policy Enforcement (Network, Capacity, etc...) | Enforcement of placement of application stacks based on policy: compliance, capacity, etc... | Elastic Runtime Apps Manager & Diego Scheduler  |
| Configuration Management (Run-Time) | Configuration management of compute instances in production runtime | Elastic Runtime Stemcells & Build Packs  | Infrastructure Orchestration | Orchestrated provisioning and configuration of stack-specific infra. | Elastic Runtime Buildpacks  | DNS Provisioning | Automated provisioning of DNS records for application components and stacks | Elastic Runtime (go)Router Routes  |
| Artifact Repository (Binaries) | Repository of successfully tested application binaries | Artifactory for PCF  | Configuration Management (Image Build) | Continuous build, test, and integration of cloud instance images and containers by application component | Stem Cells & Elastic Runtime Buildpacks  | Security | | |
| Service Discovery | Ability for services to register themselves, discover each other, & integrate autonomously | Spring Cloud Services: Eureka  | Artifact Repository (Infracode) | Repository of successfully tested infrastructure code. | Elastic Runtime Blobstore  | Middleware | | |
| Container Format | Standardized format for packaging and delivery of Linux containers | Elastic Runtime Garden  | Network Security Monitoring | Automated monitoring and alerting of internal and external network for malicious traffic | BRO  | HTTP Proxy | Reverse proxy and routing to backend micro services | Elastic Runtime (go)Router  |
| Container Scheduler | Automated scheduling, deployment, and recovery of Linux containers | Elastic Runtime Diego  | Secrets Management | Automated creation, management, and expiration of security "secrets" | Vault  | Java Container | Standard container for running Spring-based Java applications | Tomcat (tcServer) for PCF  |
| Data Platform | | | Vulnerability Audit | Proactive scanning for threats and vulnerabilities | Tenable  | Object Cache | In-memory object cache | Memcached for PCF  |
| Capability | | | Incident Response | Collection of forensic data in the event of a security event or compromise | Google Rapid Response  | Document Store | High performance No-SQL document store | MongoDB for PCF  |
| Geo-Replicated Messaging | High-throughput, distributed messaging platform replicated across AZs/Regions | Kafka for PCF  | CSP Management Console Audit | Automated security auditing of CSP admin console | Evident.io  | Messaging | Traditional asynchronous messaging | RabbitMQ for PCF  |
| Geo-Replicated NoSQL | Distributed NoSQL key/value store replicated across AZs/Regions | Cassandra for PCF  | | | | | | |
| Data Search and Analytics | Search, analysis, and visualization of data in real time | Elasticsearch for PCF  | | | | | | |



Pivotal[®]

Transforming How The World Builds Software