Headquarters U.S. Air Force

*Integrity - Service - Excellence*


# How did the Department of Defense move to Kubernetes and Istio?

**Mr. Nicolas Chaillan**

**Chief Software Officer, U.S. Air Force**

**Co-Lead, DoD Enterprise DevSecOps Initiative**

**v1.81 – UNCLASSFIED**

U.S. AIR FORCE

# Must Rapidly Adapt To Challenges


Work as a Team!

# A Large Team!


With Various Technologies


# Bring It With Us!


Even To Space!

# With a Few Sensors!

# With Their Help!

U.S. AIR FORCE

# What is the DoD Enterprise DevSecOps Initiative?

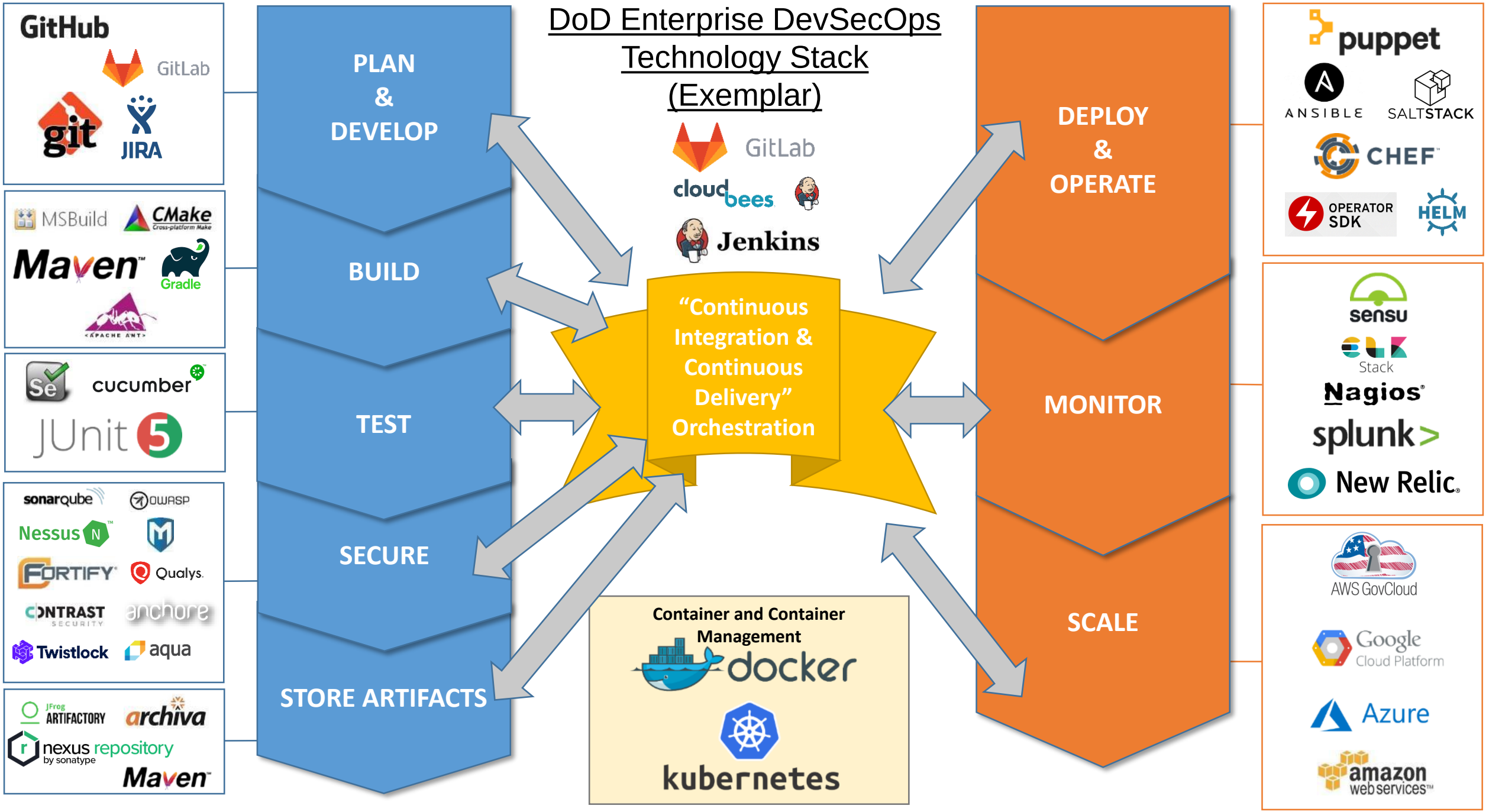
- Joint Program with OUSD(A&S), DoD CIO, U.S. Air Force, DISA and the Military Services.
- Technology excellence in execution is enabled by
  - **Avoid vendor lock-in** at the Infrastructure and Platform Layer by leveraging FOSS with **Kubernetes and OCI containers**,
  - Creating the DoD Centralized Artifacts Repository (DCAR) of hardened and centrally accredited containers,
  - **Baked-in Zero Trust Security** with our **Sidecar Container Security Stack (SCSS)** leveraging **behavior detection, zero trust down to the container/function level**,
  - Leveraging a Scalable Microservices Architecture with **Service Mesh** baked-in security, and the adoption of automation and services for platform, infrastructure, configuration, and continuous risk assessment.
- Bringing **Enterprise IT Capabilities with Cloud One and Platform One** – Cloud and DevSecOps as Managed Services capabilities, on-boarding and support!
- Standardizing metrics and define acceptable thresholds for **DoD-wide continuous Authority to Operate.**
- Massive **Scale Training with Self Learning Capabilities** (train over 100K people within a year) and bring state of the art DevSecOps curriculum
- Creating new Agile contracting language to enable and incentivize the use of DevSecOps

U.S. AIR FORCE

# CSO Website – Continuously Updated!

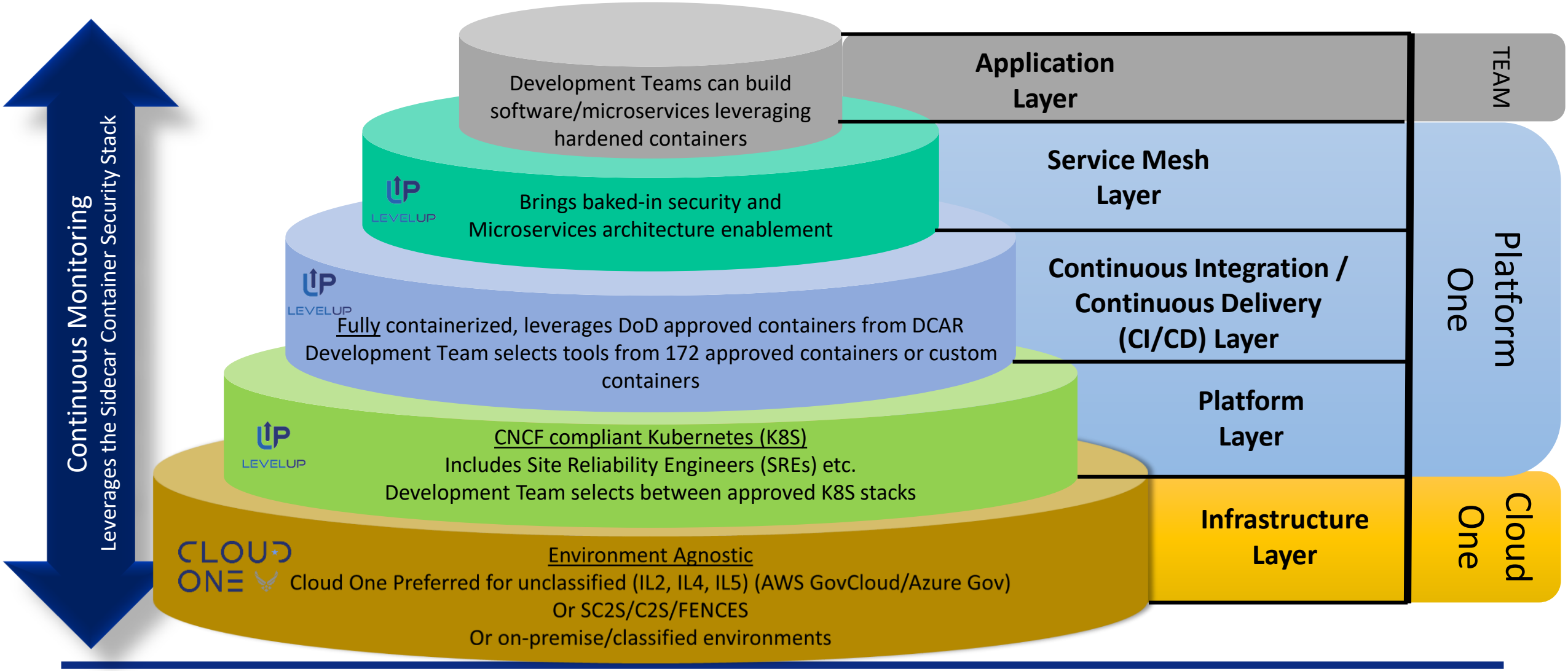
- Want to find information about the DevSecOps initiative and the CSO?
  - https://software.af.mil/
  - **Our latest documents/videos:** https://software.af.mil/dsop/documents/
  - **Our latest training videos from DAU available at:** https://software.af.mil/training/
  - More information about
    - Cloud One
    - Platform One
    - DevSecOps
    - Training including videos selection
    - Software Factories
    - Our Events/News!

# DoD Enterprise DevSecOps Technology Stack (Exemplar)



U.S. AIR FORCE

# Understanding the DevSecOps Layers



U.S. AIR FORCE

# Why Kubernetes / Containers?

- One of the most critical aspect of the DevSecOps initiative is to ensure we **avoid any vendor lock-in** so the DoD mandated:
  - **Open Container Initiative (OCI) containers** (no lock-in to containers/container runtimes/builders)
  - **Cloud Native Computing Foundation (CNCF) Kubernetes compliant cluster** for container orchestration, no lock-in to orchestration options/networking/storage APIs.
- Containers are **immutable** and will allow the DoD to centrally accredit and harden containers (FOSS, COTS, GOTS) (think of a true gold disk concept but that actually scale and works).
- Kubernetes will provide:
  - **Resiliency**: Self-healing so containers that crash can automatically be restarted,
  - **Baked-in security**: thanks to **automatic injection** of our Sidecar Container Security Stack (SCSS) to any K8S cluster with Zero Trust,
  - **Adaptability**: containers are “Lego” blocks and can be swapped with no downtime thanks to load balancing and modern routing (A/B testing, canary release etc.),
  - **Automation**: thanks to our Infrastructure as Code (IaC) and GitOps model,
  - **Auto-scaling**: if load requires more of the same container, K8S will automatically scale based on compute/memory needs,
  - **Abstraction layer**: ensure we don’t get locked-in to Cloud APIs or to a specific platform as K8S is managed by CNCF and dozens of products are compliant with its requirements.

# Key "DevSecOps" Ingredients

- **Abstracted**: to avoid drifts, be agnostic to environment (Cloud/on-premise/classified/disconnected…) and prevent lock-ins with Cloud or Platform layers, we leverage CNCF compliant Kubernetes and OCI compliant containers - open source stacks with U.S eyes on code and continuous scanning,
- **GitOps / Infrastructure as Code (IaC):** no drift, everything is code (including configuration, networking etc.) Instantiate entire stack automatically,
- **Continuous Integration/Continuous Delivery pipeline (CI/CD)**: fully containerized and using Infrastructure as Code (IaC),
- **Hardened Containers**: hardened "Lego blocks" to bring options to development teams (one size fits all lead to shadow IT)
- **Software Testing**: mandated high test coverage,
- **Baked-in Security**: mandated static/dynamic code analysis, container security, bill of material (supply chain risk) etc.
- **Continuous Monitoring:**
  - **Centralized logging and telemetry**,
  - Automated alerting,
  - **Zero trust**, leveraging Service Mesh as Sidecar (part of SCSS), down to the container level,
  - **Behavior detection** (automated prevention),
  - CVE scanning,
- **Chaos engineering**: Dynamically kills/restarts container with moving target defense.

U.S. AIR FORCE

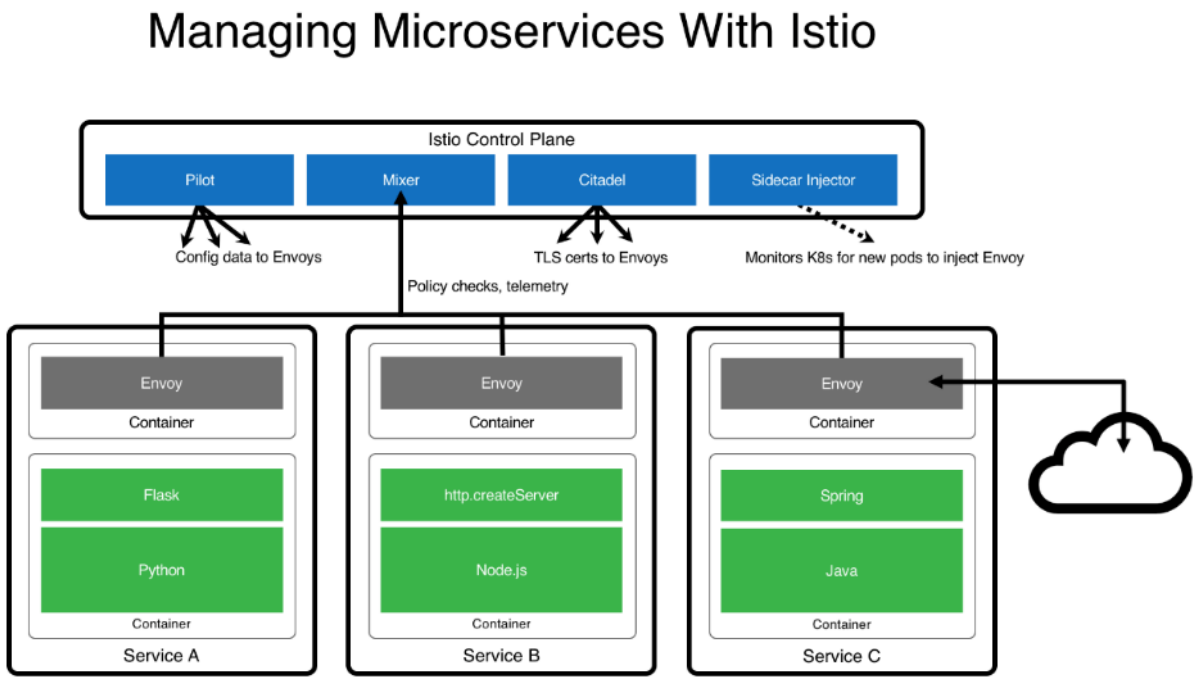
# Questions about the Agile / SAFe Memo?

- The CSO signed a Memorandum for Record on Nov 26th 2019, sent to all PEOs and PMs regarding the use of DevSecOps and Agile and **highly discouraging from using rigid, prescriptive frameworks such as the Scaled Agile Framework (SAFe)**.
- Why?
  - DoD is still using Waterfall or Water-Agile-Fall so until we can truly implement basic Scrum/Kanban, there is nothing to « SCALE ». Agile should be applied across the entire Program, not just the development team, that includes: Contracting, Program Management, Reporting to leadership (no EVM) etc!
    - You cannot scale if you don't have the "basics" right. At best, such frameworks put us at risk to fall back to what we know and go back to Waterfall because of their "mapping".
  - SAFe might potentially be an useful framework for teams that do not use DevOps/DevSecOps but a key principle of DevSecOps is to decouple work and teams and the only synchronization required should be across Product Owners. Teams shouldn't have to coordinate if they use a Service Mesh/Domain Driven Design/Microservices model. This doesn't require a rigid framework. If you're having issues implement this, you're not implementing a true DevSecOps model.
  - Take what is best from any framework and make it work for your team! Certifications aren't always the answer!
  - Fundamentally, the main "goal" of Software development is NOT to be « SAFE », it is to INNOVATE and CREATE. You do not create by not taking risks… unless you're part of the far less than 5% of AF software that implements safety critical functions… it is quite the opposite:
    - « Continuous Learning: Fail Fast but don't Fail twice for the same reason! » - Small incremental changes which mitigate risks and create safe conditions to implement rapid changes.
  - SAFe isn't used by any successful software commercial organization (Facebook, Google, Netflix, etc.).
  - Looking to coordinate your Product Owners' work? Multiple models exist. This shouldn't impact the developers.
  - Don't believe us? Listen to the Agile fathers: http://www.smharter.com/blog/safe-a-collection-of-comments-from-leading-experts/

U.S. AIR FORCE

# Microservices Architecture (ISTIO)

- Turnkey Service Mesh (ISTIO) architecture
- ISTIO side car proxy, baked-in security, with visibility across containers, by default, without any developer interaction or code change
- Benefits:
  - API Management, service discovery, authentication…
  - Dynamic request routing for A/B testing, gradual rollouts, canary releases, resilience, observability, retries, circuit breakers and fault injection
  - Layer 7 Load balancing
  - Zero Trust model: East/West Traffic Whitelisting, ACL, RBAC…
  - TLS encryption by default, Key management, signing…



U.S. AIR FORCE

# "Infrastructure as Code" Benefits

The "Infrastructure as Code" concept is a critical DevSecOps ingredient to ensure that production environments do not drift from development/testing environments. No human should make changes in production environments. Changes should only be made in source code and redeployed by the CI/CD pipeline.

- No drift between environments, whether classified/disconnected/Cloud/on-premise
- Immutable,
- Replicable,
- Automated,
- No human in production environments: reduces attack surface (disable SSH etc.), insider threat and configuration drifts,
- Everything is code: including playbooks, networking, tests, configuration etc.

# What is GitOps?

- Based on Infrastructure as Code concepts, makes Git the single source of truth of the desired state of your Infrastructure, Platform and Applications.
- Benefits:
  - Everything is code: infrastructure, networking, configuration, sealed secrets etc.
  - Auditability & Compliance
  - Consistent deployments and rollback (no drifts between environment)
  - Configuration Management enforcement
  - Disaster Recovery
  - Baked-in security: Kubernetes clusters **pulls** from Git. CI/CD won't have access to production clusters. Removing human from production environments
  - Declarative manifests and playbooks
- Options:
  - Argo CD, Flux as FOSS. Projects are merging into a single FOSS and be part of CNCF.

U.S. AIR FORCE


# Thank You!

**Nicolas Chaillan**
**Chief Software Officer, U.S. Air Force**
usaf.cso@mail.mil

# Backup Slides

U.S. AIR FORCE

# Nicolas Chaillan - Presenter

Chief Software Officer

- Nicolas M. Chaillan is the Chief Software Officer at the U.S. Air Force and the Co-Lead for the DoD Enterprise DevSecOps Initiative.
- He is the former Special Advisor for Cloud Security and DevSecOps at OSD, A&S.
- He was the Special Advisor for Cybersecurity at the Department of Homeland Security and the Chief Architect for Cyber.gov, the new robust, innovative and holistic .Gov cyber security architecture for all .gov agencies.
- Chaillan is a technology entrepreneur, software developer, cyber expert and inventor. He is recognized as one of France's youngest entrepreneurs after founding his first company at 15 years of age.
- With 19 years of international tech, entrepreneurial and management experience, Chaillan is the founder of more than 12 companies, including AFTER-MOUSE.COM, Prevent-Breach, anyGuest.com, and more.
- Over the last eight years alone, he has created and sold over 180 innovative software products to 40 Fortune 500 companies.
- Chaillan is recognized as a pioneer of the computer language PHP.

2018
OFFICIAL MEMBER
Forbes
Technology Council

U.S. AIR FORCE

# DCCSCR/DCAR (DoD Container Repository)

- Containers are centrally accredited by the DSOP team in the DoD repository:
  - **DoD Centralized Containers Source Code Repository (DCCSCR)**: https://dccscr.dsop.io/dsop
  - DCCSCR Infrastructure as Code (IaC): https://dccscr.dsop.io/levelup-automation/aws-infrastructure
    - Allows DoD programs to reuse DevSecOps stack and CI/CD pipelines to ensure pre-hardened deployments.
- **DoD Centralized Artifacts Repository (DCAR)** (Container binaries): https://dcar.dsop.io
- Containers are signed and continuously monitored.
- Community can contribute code merge requests, reviewed by the DSOP team.
- Vendors/DoD Programs can contribute containers that have enterprise benefits to DCCSCR/DCAR and DSOP team will accredit them and maintain them.

U.S. AIR FORCE

# Key "Continuous Security" Ingredients

- **Kubernetes hardening.**
  - Automated injection of Sidecar Container Security Stack (SCSS) into all containers/pods running without manual action.
  - RBAC/SSO/SELinux enabled
  - Compliant with CIS Kubernetes Benchmark, mapped to NIST 800-53
  - Nodes, master, etcd are hardened.
  - Automated backups of cluster and persistent storage!
- **Sidecar Container Security Stack (SCSS):**
  - Automated centralized logging and telemetry with Elasticsearch, Fluentd, Kibana (EFK),
  - Service Mesh (Istio):
    - Baked-in **zero trust model** down to the container level!
      - Strong identities automatically generated using certificates.
      - mTLS tunnel injected across all container communication
    - Whitelist enforcement, Layer 7 load balancer etc.
  - Container security: Continuous Scanning, Alerting, CVE scanning, **Behavior detection** both in development and production (Build, Registry, Runtime) with Twistlock (looking into StackRox and Sysdig),
  - Container security and insider threat (custom policies detecting unapproved changes to Dockerfiles) with Anchore;
  - Automated STIG compliance with OpenSCAP.

U.S. AIR FORCE

# DevSecOps Stack implements Zero Trust!

- **Identities**:
  - strong NPE identities are automatically managed by Istio (Service Mesh) for each container to enable zero trust down to the container level.
  - Non-NPE identities are using strong identities with DoD PKI
- **Devices**:
  - Developer endpoints are using VDI options or approved endpoints images
- **Applications**:
  - Apps are containerized and behind the Service Mesh which enforces Zero trust with strong identities per pod/container and .
- **Infrastructure**:
  - Kubernetes is centrally hardened and continuously monitored with centralized logs and telemetry.
  - SCSS monitors container signatures and container state
  - SCSS brings Behavior detection and CVE continuous scanning
- **Network**:
  - mTLS tunnels are automatically injected across all containers/pods by SCSS.
- **Data**:
  - Data is always encrypted in transit and leverages FIPS encryption at rest.

U.S. AIR FORCE

# What is a Continuous ATO?

- A Continuous ATO is very different from a traditional ATO or a Fast-Track/Accelerated ATO:
  - Platforms have to be compliant with the DoD Enterprise DevSecOps Ref Design to ensure DoD-wide reciprocity, including the use of the Sidecar Container Security Stack (SCSS). Platform controls are mapped to NIST-800-53.
  - We accredit the Platform's **PROCESS** (Continuous Integration/Continuous Delivery (Software Factory)) **with mandated testing and security gates.** The software coming out of the factory and that is RUNNING IN PRODUCTION **on the Platform** (Kubernetes with SCSS) also benefits from the cATO.
  - We accredit **TEAMS** using the Platform so they can produce quality software and be trained to move to DevSecOps
  - A key principle of DevSecOps is the **baked-in security** with:
    - Zero Trust
    - Automation
    - Removal of environment drifts
    - Behavior Detection
    - Continuous Monitoring
    - Pen-testing

U.S. AIR FORCE

# Value for DoD Programs

- Enables any DoD Program across DoD Services deploy a DoD hardened Software Factory, on their existing or new environments (including classified, disconnected and Clouds), within days instead of a year. Tremendous cost and time savings.
- Multiple DevSecOps pipelines are available with various options (no one-size-fits-all)
- Enables rapid prototyping (in days and not months or years) for any Business, C4ISR and Weapons system. Deployment in PRODUCTION!
- Enables learning and continuous feedback from actual end-users (warfighters).
- Enables **bug and security fixes in minutes** instead of weeks/months.
- Enables automated testing and security.
- Enables **continuous Authorization to Operate (c-ATO)** process. Authorize ONCE, use MANY times!
- Brings a holistic and baked-in cybersecurity stack, gaining complete visibility of all assets, software security state and infrastructure as code.

U.S. AIR FORCE

CLOUD ONE

# Cloud One

- Air Force Cloud Office with turnkey access to AWS GovCloud and Azure Government at IL2, 4 and 5. IL6 available by December 2019.
- Simple “Pay per use” model with ability to instantiate your own Development and Production VPCs at various Impact Levels within days with full compliance/security and a baked-in ATO.
- Enterprise Solution: we provide the guardrails to the cloud in a standard manner so you can focus on your mission
- Fully Automated: All environmental stand-up is managed by Infrastructure as Code, drastically speeding up deployment, reducing manual work, and human error
- Centralized Identities and Single-Sign-On (SSO): one login across the Cloud stack
- Internet facing Cloud based VPN to connect to IL5 enclaves with a Virtual Internet Access Point (coming within January 2020).
- DevSecOps Focused: secure, mission driven deployments are built into the framework to ensure self-service and seamless deployments. Leverages Zero Trust model.
- Proactive Scaling and System Monitoring: Mission Owners can see all operational metrics and provide rules and alerts to manage each mission their way
- Accreditation Inheritance has been identified in the AF-Cloud One eMASS accounts (AWS & Azure) to include inheritance from the CSP, USAF, DoD and CSSP. All that’s left for the mission is the controls that are unique to them.

# "Platform One by LevelUP"
# The Air Force Software Factory Team

- Merged top talent across U.S. Air Force from various Factories (Kessel Run, SpaceCAMP and UP.
- Helps instantiate DevSecOps CI/CD pipelines / Software Factories within days at various classification levels.
- Manages Software Factories for Development teams so they can focus on building mission applications.
- Provides Blanket Purchase Agreement (BPA) DoD-wide DevSecOps contracts for Cloud Service, Talent and Licenses. Enables awards every 15/30 days with bulk discounts.
- Decouples Development Teams from Factory teams with DevSecOps and Site Reliability Engineer (SRE) expertise.
- Partners with Cloud One to provide IL2, 4, 5 and 6 access but also uses C2S/SC2S and various on-premise environments!
- Self-learning and training capabilities to enable teams move to Scrum/Kanban/eXtreme Programming (XP) Agile practices.
- Leverages the DoD hardened containers while avoiding one-size-fits-all architectures.
- Fully compliant with the DoD Enterprise DevSecOps Initiative (DSOP) with DoD-wide reciprocity and an ATO. Leverages Zero Trust model.
- Hardens the 172 DoD enterprise containers (databases, development tools, CI/CD tools, cybersecurity tools etc.).
- Provides Software Enterprise Services with Collaboration tools, Cybersecurity tools, Source code repositories, Artifact repositories, Development tools, DevSecOps as a Service, Chats etc. These services will be MANAGED services on Cloud One.

U.S. AIR FORCE

LEVELUP CODE WORKS

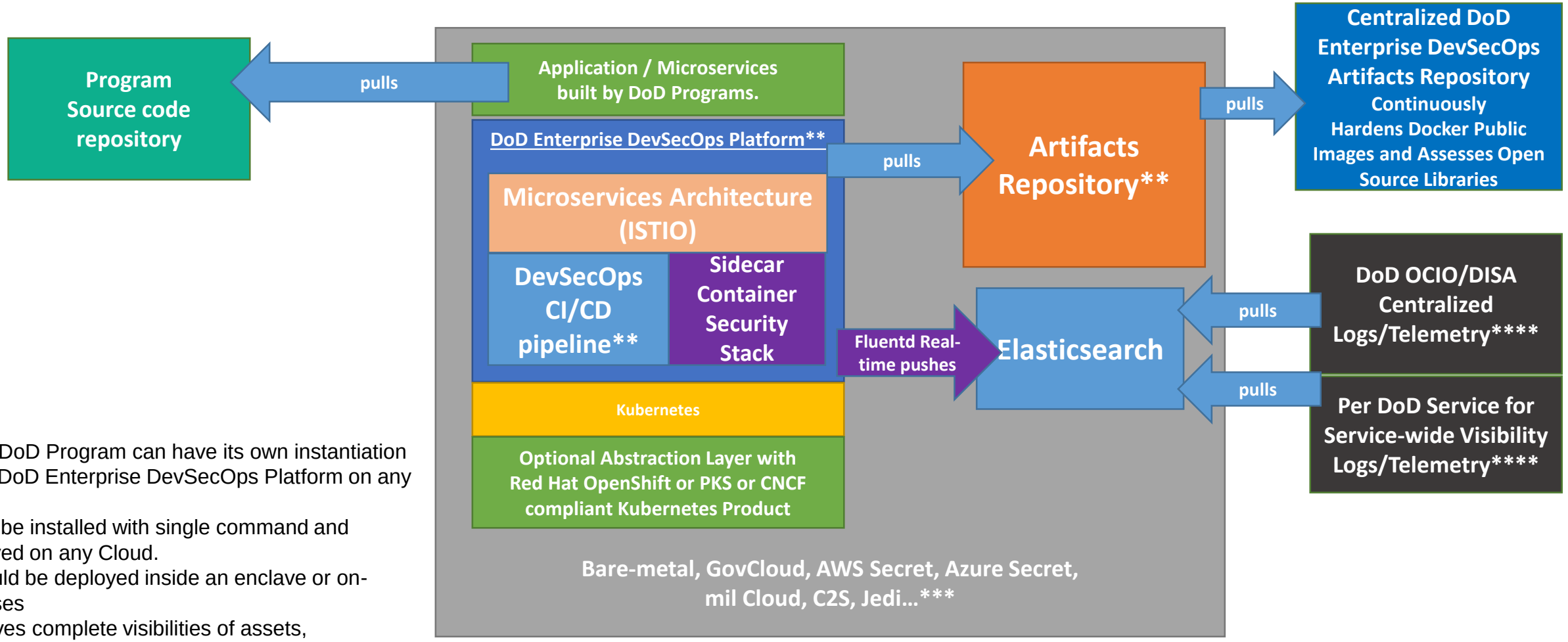
# "Platform One by LevelUP" Managed Services "A La Carte"

- Hardened Containers Options
  - Delivery of hardened enterprise containers with accreditation reciprocity (existing containers only).
  - Delivery of custom hardened containers as needed.
- Continuous Integration / Continuous Delivery (CI/CD) Options
  - Delivery of existing hardened Kubernetes/OpenShift/PKS playbooks (full Infrastructure as Code).
  - Delivery of a **turnkey CI/CD pipeline** (Software Factory) with complete « Infrastructure as Code » to instantiate on any environment (development teams picks the tools from the approved hardened containers) on various classified/unclassified environment.
- Training/On-Boarding Options
  - 1-day training Session: introduction to DevSecOps. Overview and understanding of the vision and activities.
  - A 3 day introduction to LevelUP DevSecOps tech stack. Hands on code and User-Centered Design (UCD) to deploy your first demo app to production.
  - A several week full on-boarding, that concludes with an MVP ready for production.
  - A several month full on-boarding, that concludes with your platform team being able to support your own DevSecOps applications for development and production.
  - Customized training options (both at our locations or on your premises).
- Contracting Support Options
  - Ability to leverage the DevSecOps BOAs (Cloud Services, Talent and Licenses).
  - Enable access to DevSecOps engineers/SREs Full-Time-Equivalent (FTEs) (Medics/Counselors) to assist Programs.



# DoD Enterprise DevSecOps Architecture

# DoD Enterprise DevSecOps Architecture*




*each DoD Program can have its own instantiation of the DoD Enterprise DevSecOps Platform on any Cloud.
** can be installed with single command and deployed on any Cloud.
*** could be deployed inside an enclave or on-premises
**** gives complete visibilities of assets, security/vulnerability state etc. can be integrated to existing cybersecurity shared services.



# DevSecOps Platform Stack (continuously evolving)


# DevSecOps Product Stack (1)

| Source Repository | API Gateways | Programming Languages | Databases |
|---|---|---|---|
| GitHub Government | Kong | C/C++ | SQL Server |
| GitLab | Azure API | C#/.NET | MySQL |
| | AWS API | .NET Core | PostgreSQL |
| **Container Management technologies:** | Axway | Java | MongoDB |
| Kubernetes | 3Scale | PHP | SQLite |
| Openshift | Apigee | Python | Redis |
| VMWare Tanzu | ISTIO (service mesh) | Groovy | Elasticsearch |
| PKS | | Ruby | Oracle |
| OKD | **Artifacts** | R | etcd |
| Rancher (K8S only) | Artifactory | Rust | Hadoop/HDInsight |
| D2IQ (K8S only) | Nexus | Scala | Cloudera |
| Docker EE (K8S only) | Maven | Perl | Oracle Big Data |
| | Archiva | Go | Solr |
| **Container Packagers:** | S3 bucket | Node.JS | Neo4J |
| Helm | | Swift | Memcached |
| Kubernetes Operators | | | Cassandra |
| | | | MariaDB |
| | | | CouchDB |
| | | | InfluxDB (time) |

U.S. AIR FORCE

# DevSecOps Product Stack (2)

**Message bus/Streams**
Kafka
Flink
Nats
RabbitMQ
ActiveMQ

**Proxy**
Oauth2 proxy
nginx ldap auth proxy
openldap
HA Proxy

**Visualization**
Tableau
Kibana

**Logs**
Logstash
Splunk Forwarder
Fluentd
Syslogd
Filebeat
rsyslog

**Webservers**
Apache2
Nginx
IIS
Lighttpd
Tomcat

**Docker base images OS:**
~~Alpine~~
~~Busybox~~
~~Ubuntu~~
~~Centos~~
~~Debian~~
~~Fedora~~
Universal Base Image

**Serverless**
Knative

U.S. AIR FORCE

# DevSecOps Product Stack (3)

**Build**
MSBuild
CMake
Maven
Gradle
Apache Ant

**Tests suite**
Cucumber
J-Unit
Selenium
TestingWhiz
Watir
Sahi
Zephyr
Vagrant
AppVerify
nosetests
SoapUI
LeanFT

**Test coverage**
JaCoCo
Emma
Cobertura
codecov

**CI/CD Orchestration**
Jenkins (open source)
CloudBees Jenkins
GitLab

**Jenkins plugins**
Dozens (Need to verify security).

**Configuration Management / Delivery**
Puppet
Chef
Ansible
Saltstack

**Security**
Tenable / Nessus Agents
Fortify
Twistlock
Aqua
SonarQBE
Qualys
StackRox
Aporeto
Snort
OWASP ZAP
Contrast Security
OpenVAS
Metasploit
ThreadFix
pylint
JFrog Xray
OpenSCAP (can check against DISA STIG)
OpenControl for compliance documentation

**Security (2)**
Snyk
Code Climate
AJAX Spider
Tanaguru (508 compliance)
InSpec
OWASP Dependency-Check
Burp
HBSS
Anchore
Checkmarx
SD Elements
Clair
Docker Bench Security
Notary
Sysdig
Layered Insight
BlackDuck
Nexus IQ/Lifecycle/Firewall

U.S. AIR FORCE

# DevSecOps Product Stack (4)

**Monitoring**
Sensu
EFK (Elasticsearch, Fluentd, Kibana)
Splunk
Nagios
New Relic
Sentry
Promotheus
Grafana
Kiali

**Collaboration**
Rocket.Chat
Matter.Most
PagerDuty

**Plan**
Jira
Confluence
Rally
Redmine
Pivotal Tracker

**Secrets**
Kubernetes Secrets
Vault
Credentials (Jenkins)
CryptoMove

**SSO**
Keycloak

**Documentation**
Javadoc
RDoc
Sphinx
Doxygen
Cucumber
phpDocumentator
Pydoc

**Performance**
Apache AB
Jmeter
LoadRunner

U.S. AIR FORCE

# Self-Learning (1)

- Recommended Videos (Part 1)
  - Watch our playlists, available at different expertise levels and continuously augmented!
  - Kafka / KSQL (message bus, pub/sub, event driven):
    - Beginners: https://www.youtube.com/playlist?list=PLSIv_F9TtLlzz0zt03Ludtid7icrXBesg
    - Intermediate: https://www.youtube.com/playlist?list=PLSIv_F9TtLlxxXX0oCzt7laO6mD61UIQw
    - Advanced: N/A
  - Kubernetes
    - Beginners: https://www.youtube.com/playlist?list=PLSIv_F9TtLlydFzQzkYYDdQK7k5cEKubQ
    - Intermediate: https://www.youtube.com/playlist?list=PLSIv_F9TtLlx8dSFH_jFLK40Tt7KUXTN_
    - Advanced: https://www.youtube.com/playlist?list=PLSIv_F9TtLlytdAJiVqbHucWOvn5LrTNW

U.S. AIR FORCE

# Self-Learning (2)

- Recommended Videos (Part 2)
  - Watch our playlists, available at different expertise levels and continuously augmented!
  - Service Mesh
    - Beginners: https://www.youtube.com/playlist?list=PLSIv_F9TtLlxtC4rDIMQ8QiG5UBCjz7VH
    - Intermediate: https://www.youtube.com/playlist?list=PLSIv_F9TtLlwWK_Y_Cas8Nyw-DsdbH6vl
    - Advanced: https://www.youtube.com/playlist?list=PLSIv_F9TtLlx8VW2MFONMRwS_-2rSJwdn
  - Microservices
    - Beginners: https://www.youtube.com/playlist?list=PLSIv_F9TtLlz_U2_RaONTGYLkz0lh-A_L
    - Intermediate: https://www.youtube.com/playlist?list=PLSIv_F9TtLlxqjuAXxoRMjvspaEE8L2cB
    - Advanced: https://www.youtube.com/playlist?list=PLSIv_F9TtLlw4CF4F4t3gVV3j0512CMsu

U.S. AIR FORCE

# Self-Learning (3)

- Recommended Books
  - A Seat at the Table – by Mark Schwartz (former CIO of USCIS, leader in Agile)

    This book is highly recommended for ALL leadership as it is not technical but focused on the challenges around business, procurement and how leadership can enable DevOps across the organization and remove impediments.
  - The Phoenix Project – by the founders of DevOps
  - The DevOps Handbook – by Gene Kim, Patrick Debois.

For those who drive to work like me (for hours), please note that these books are available as Audiobooks.

U.S. AIR FORCE

# Legacy to DevSecOps => Strangler Pattern

- Martin Fowler describes the Strangler Application:
  - *One of the natural wonders of this area are the huge strangler vines. They seed in the upper branches of a fig tree and gradually work their way down the tree until they root in the soil. Over many years they grow into fantastic and beautiful shapes, meanwhile strangling and killing the tree that was their host.*
- To get there, the following steps were followed:
  - First, add a proxy, which sits between the legacy application and the user. Initially, this proxy doesn't do anything but pass all traffic, unmodified, to the application.
  - Then, add new service (with its own database(s) and other supporting infrastructure) and link it to the proxy. Implement the first new page in this service. Then allow the proxy to serve traffic to that page (see below).
  - Add more pages, more functionality and potentially more services. Open up the proxy to the new pages and services. Repeat until all required functionality is handled by the new stack.
  - The monolith no longer serves traffic and can be switched off.
- Learn more: https://www.ibm.com/developerworks/cloud/library/cl-strangler-application-pattern-microservices-apps-trs/index.html and https://www.michielrook.nl/2016/11/strangler-pattern-practice/