Cloud-native Security and Policy: a Primer

Exploring practical and actionable solutions to modern problems

Who am I?

Evan Kwisnek
evan@ctrlplane.net
github.com/ekwisnek

Veteran → Technology Instructor

Network Engineer → System Administrator

System Engineer → System Architect

In the news…

"Through 2025, 99% of cloud security failures will be the customer's fault" – Gartner

"Though 2025, 90% of organizations that fail to control public cloud use will inappropriately share sensitive data." – Gartner

40% of respondents answered "Yes" to: "Has your organization ever experienced a data breach involving data and applications that reside in the cloud?" - Statista, 2021

Approximately 22.1 million records exfiltrated in 2014 Office of Personnel Management (OPM) breach

The same PlugX implant used as recently as 2021

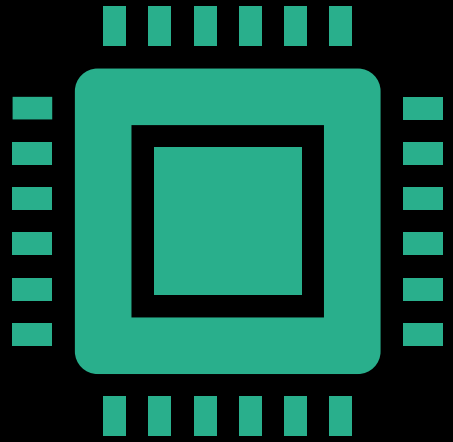
Introduction

- Modern system architectures have exceeded the bounds of traditional data centers
- Networks now span the globe, exposing critical business functions to malicious actors, unexpected outages, and costly downtime
- How do we meet the security engineering and policy governance challenges that cloud-native, hybrid, and edge deployments present?

Computing Inversions

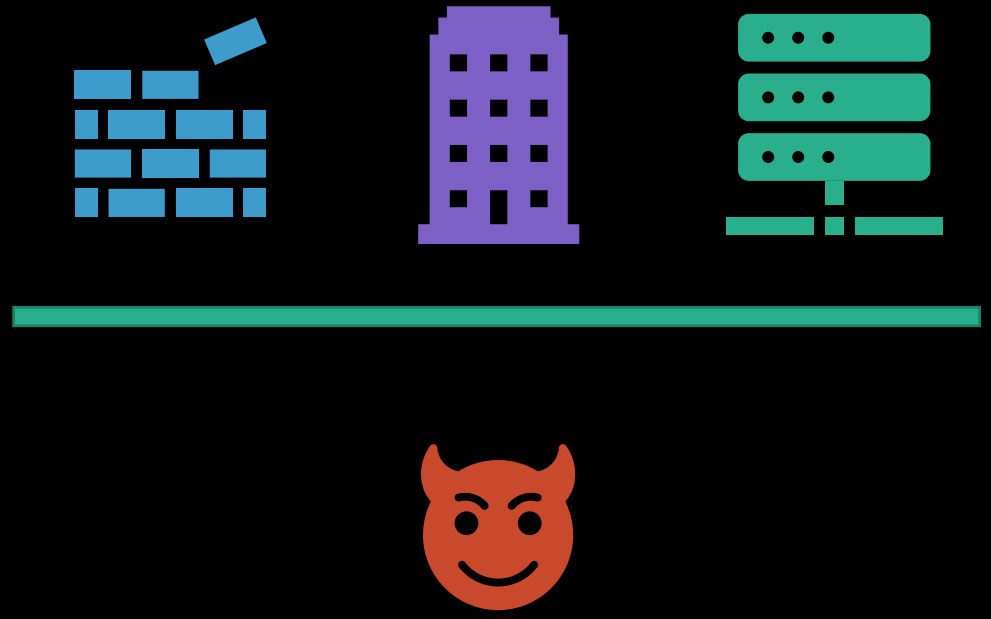
- Mainframe computers
- Personal computer revolution
  - Our first inversion
- “Pay-to-play” infrastructure (AWS, GCP)
  - Our second inversion
  - *This is what, in general, became “The Cloud™”*
- “The Edge”
  - We’re currently amid a “third inversion”

What do these bring about?

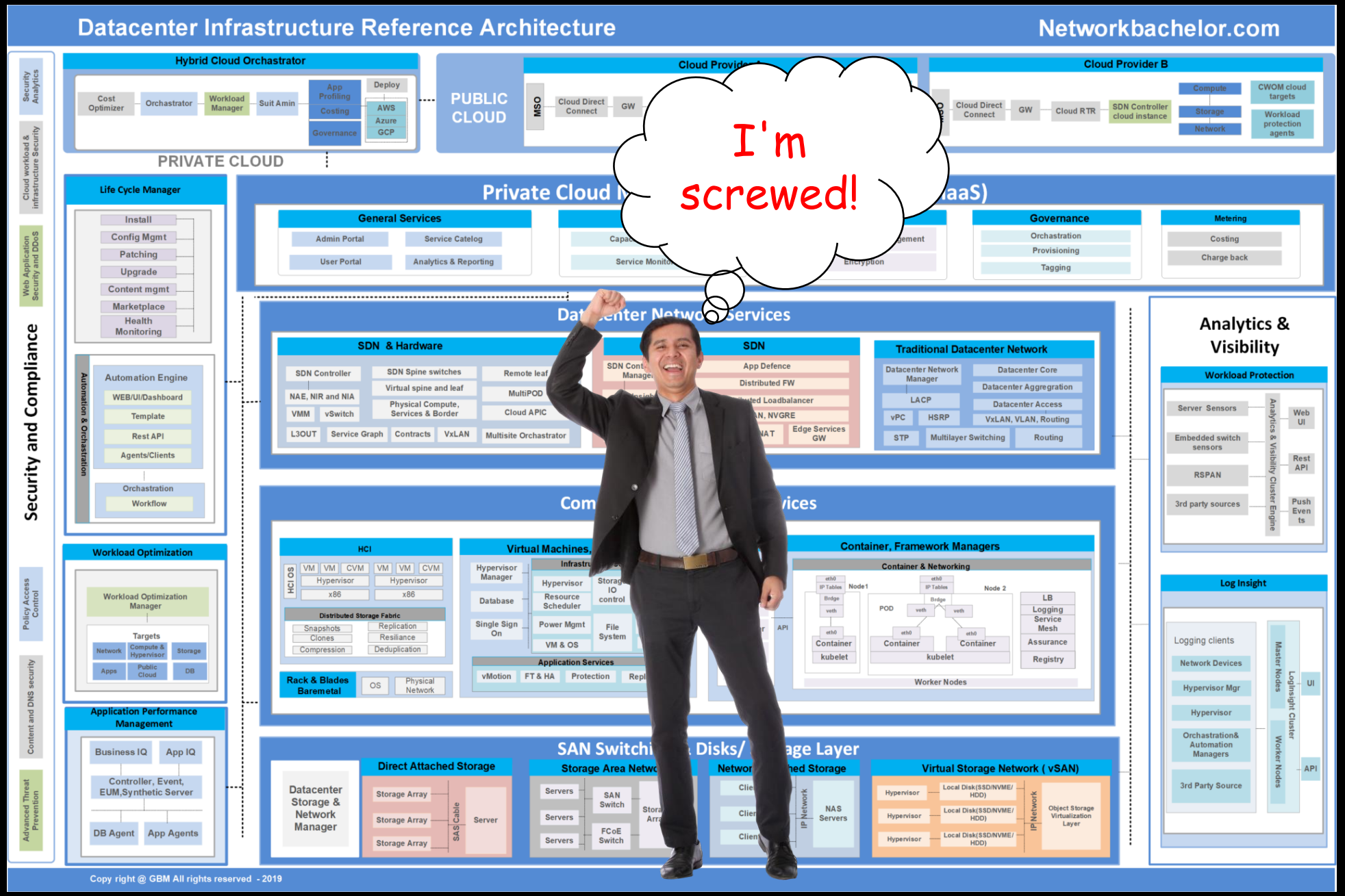


- These shifting sands bring dramatic changes in the ways our systems are designed and deployed
- They present new challenges to stay ahead by engineering secure and performant systems

Traditional Large-system Architectures



- Traditional large-system architectures were simple by comparison:
  - If I can keep bad guys out, I'm good!
- Trusted perimeter
- Trusted access
- Secured data
- …But was it trivial?


Datacenter Infrastructure Reference Architecture
Networkbachelor.com
Hybrid Cloud Orchastrator
Cost Optimizer
Orchastrator
Workload Manager
Suit Amin
App Profiling
Costing
Governance
Deploy
AWS
Azure
GCP
PUBLIC CLOUD
Cloud Provider B
Cloud Direct Connect
GW
Cloud RTR
SDN Controller cloud instance
Compute
Storage
Network
CWOM cloud targets
Workload protection agents
PRIVATE CLOUD
I'm screwed!
Security Analytics
Cloud workload & infrastructure Security
Web Application Security and DDoS
Security and Compliance
Policy Access Control
Content and DNS security
Advanced Threat Prevention
Life Cycle Manager
Install
Config Mgmt
Patching
Upgrade
Content mgmt
Marketplace
Health Monitoring
Automation Engine
WEB/UI/Dashboard
Template
Rest API
Agents/Clients
Orchestration
Workflow
Automation & Orchestration
Workload Optimization
Workload Optimization Manager
Targets
Network
Compute & Hypervisor
Storage
Apps
Public Cloud
DB
Application Performance Management
Business IQ
App IQ
Controller, Event, EUM,Synthetic Server
DB Agent
App Agents
Private Cloud
General Services
Admin Portal
Service Catelog
User Portal
Analytics & Reporting
Governance
Orchestration
Provisioning
Tagging
Metering
Costing
Charge back
SDN & Hardware
SDN Controller
SDN Spine switches
Remote leaf
NAE, NIR and NIA
Virtual spine and leaf
MultiPOD
VMM
vSwitch
Physical Compute, Services & Border
Cloud APIC
L3OUT
Service Graph
Contracts
VxLAN
Multisite Orchastrator
SDN
App Defence
Distributed FW
Edge Services GW
Traditional Datacenter Network
Datacenter Network Manager
Datacenter Core
Datacenter Aggregration
LACP
Datacenter Access
vPC
HSRP
VxLAN, VLAN, Routing
STP
Multilayer Switching
Routing
HCI
Distributed Storage Fabric
Snapshots
Clones
Compression
Replication
Resiliance
Deduplication
Rack & Blades Baremetal
OS
Physical Network
Virtual Machines
Hypervisor Manager
Database
Single Sign On
Hypervisor
Resource Scheduler
Power Mgmt
VM & OS
File System
Application Services
vMotion
FT & HA
Protection
Container, Framework Managers
Container & Networking
Node1
Node 2
POD
Container
kubelet
LB
Logging
Service Mesh
Assurance
Registry
Worker Nodes
SAN Switchi Disks/ age Layer
Datacenter Storage & Network Manager
Direct Attached Storage
Storage Array
SAS Cable
Server
Storage Area Netwo
Servers
SAN Switch
FCoE Switch
Networ ed Storage
IP Network
NAS Servers
Virtual Storage Network (vSAN)
Hypervisor
Local Disk(SSD/NVME/ HDD)
Object Storage Virtualization Layer
Analytics & Visibility
Workload Protection
Server Sensors
Embedded switch sensors
RSPAN
3rd party sources
Analytics & Visibility Cluster Engine
Web UI
Rest API
Push Even ts
Log Insight
Logging clients
Network Devices
Hypervisor Mgr
Hypervisor
Orchastration& Automation Managers
3rd Party Source
Master Nodes
Worker Nodes
LogInsight Cluster
UI
API
Copy right @ GBM All rights reserved - 2019

Scaling up and Scaling Out

- Why traditional security and policy structures and strategies don’t work:
  - Scale problem
  - Disparate networks
  - Numerous security boundaries, or “points of entry”, AKA, larger attack surface
  - More hands in the pot
    - Developers with “push-to-prod” ability
    - Infrastructure Engineers
    - SRE Team
    - DevSecOps Team

Old Problems New Again

- In what ways have the targets shifted to secure our systems under these new design and deployment paradigms?
- What are the primary challenges in securing these large-scale, distributed systems?

Who are these people?

- Dozens of developers and engineers accessing your systems daily
- Hundreds (if not thousands) of credentialed actions performed by automation and other non-persons daily
- Vendor lock-in is tempting but limiting
  - What if you need to be multi-cloud or hybrid?
- AD and LDAP aren’t going to save you now

There's a firewall for that!

- Protecting internal networks from external networks used to be the primary driver for security applications and appliances
  - Firewalls are meant to segment networks based on differing security requirements, or to only permit certain devices the ability to communicate across disparate networks
- Without implementing Virtual Private Clouds (VPC), the lines between "internal" and "external" become much more blurred

Storage of critical data

- Where are your hard drives?
- It used to be the case that you could physically secure your drives, and not worry about things like General Data Protection Regulation (GDPR) and strong data locality requirements
  - The drives were in your building!
- Now, cloud block and object storage cannot (easily) guarantee locality nor encryption at rest

Databases

- Database-as-a-Service (DBaaS) is increasingly popular
  - MongoDB Atlas
  - Amazon RDS, Dynamo, Aurora
  - Google Bigtable, Bigquery, CloudSQL, Firestore
- Attractive targets for zero-days, injection attacks – why?
  - Bang for the buck
  - What's stored in databases?
    - User PII, credit card data, etc.
- Usually, the first place ransomware hackers go with leaked or stolen credentials

To summarize all of that…

- OK, so what are all these individual security and policy considerations?
- We're talking about the three pillars of modern information security engineering:

Confidentiality

Integrity

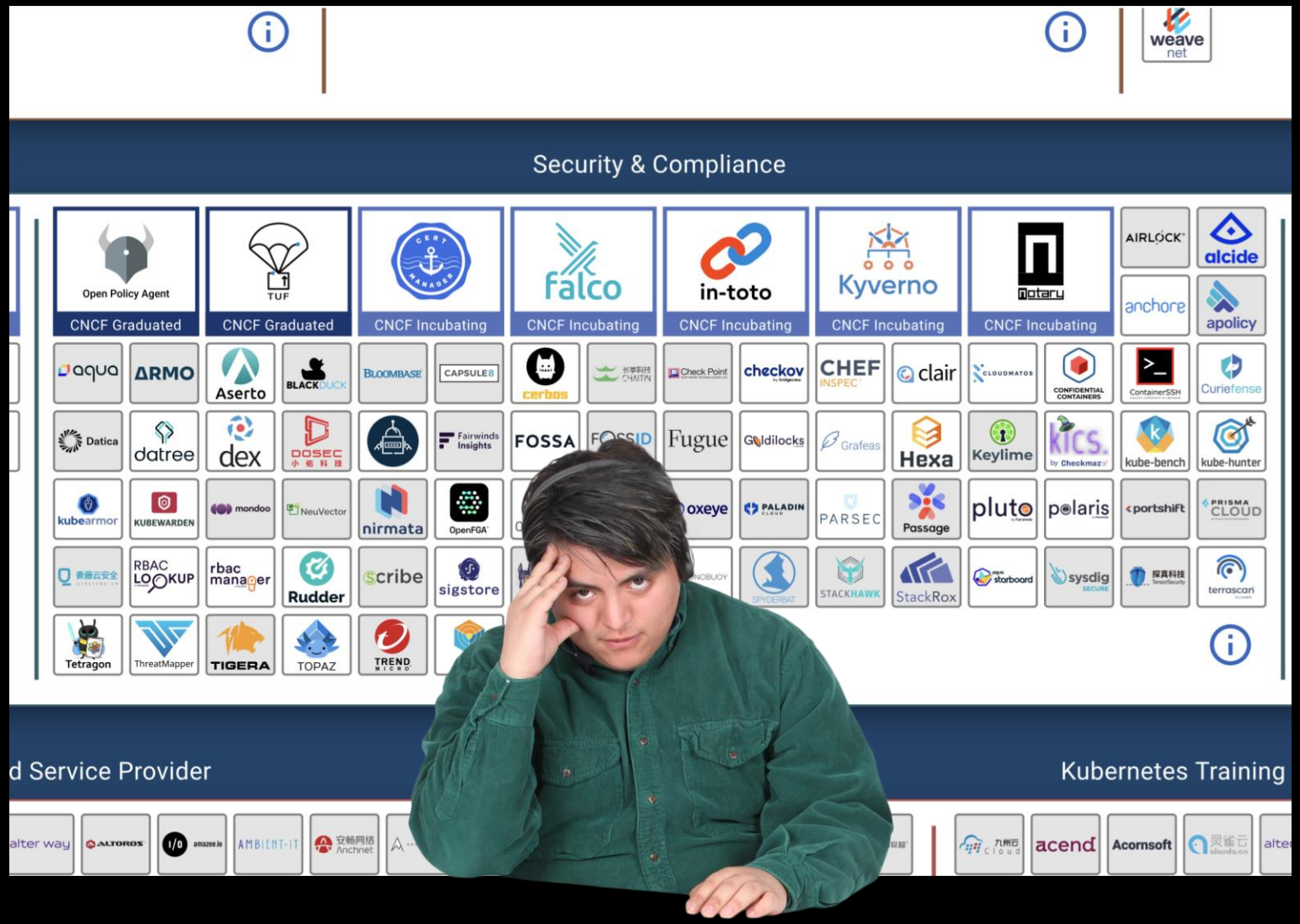
Availability

Confidentiality, Integrity, Availability (CIA)

- Confidentiality:
  - Applied to data in all forms (At-rest, In-flight, In-use)
  - Applied to communications (TLS, VPN, etc.)
- Integrity:
  - Can you verify that you're talking to the entity you think you are?
  - Can you verify the data you're being sent is the data you requested?
  - Can you verify that your data hasn't been modified in any way?
- Availability
  - Do you have a Service Level Agreement (SLA)?
  - Will your data be available on-demand?
  - Can you survive catastrophic failure of one or more of your critical systems?

Solution Time!

Here you go… …Good luck!



Policy before engineering

- Large systems require that we establish policy before we start building
- No “Waterfall” or “Big Bang” systems
  - All details, including security engineering and policy are “fully worked out” before we go to market
    - This is a troublesome and fallacious approach
  - In this case, we *can* build the ship underway
- In fact, policies should be created in a flexible and mutable way to ease adaptation for future security accreditations and improvements
- Give yourself room to grow, make the policy part of your code base

Policy As Code

- Kubernetes:
  - Kyverno, OPA Gatekeeper
- AWS:
  - AWS Policy Generator (Access Policy Language)
- GCP:
  - IAM Policy
- Should be source-controlled just as your application and DevOps code
- Separation of concerns is important here

Tackling Confidentiality First

- Con·fi·den·ti·al·i·ty
  - *noun*
  - the state of keeping or being kept secret or private.

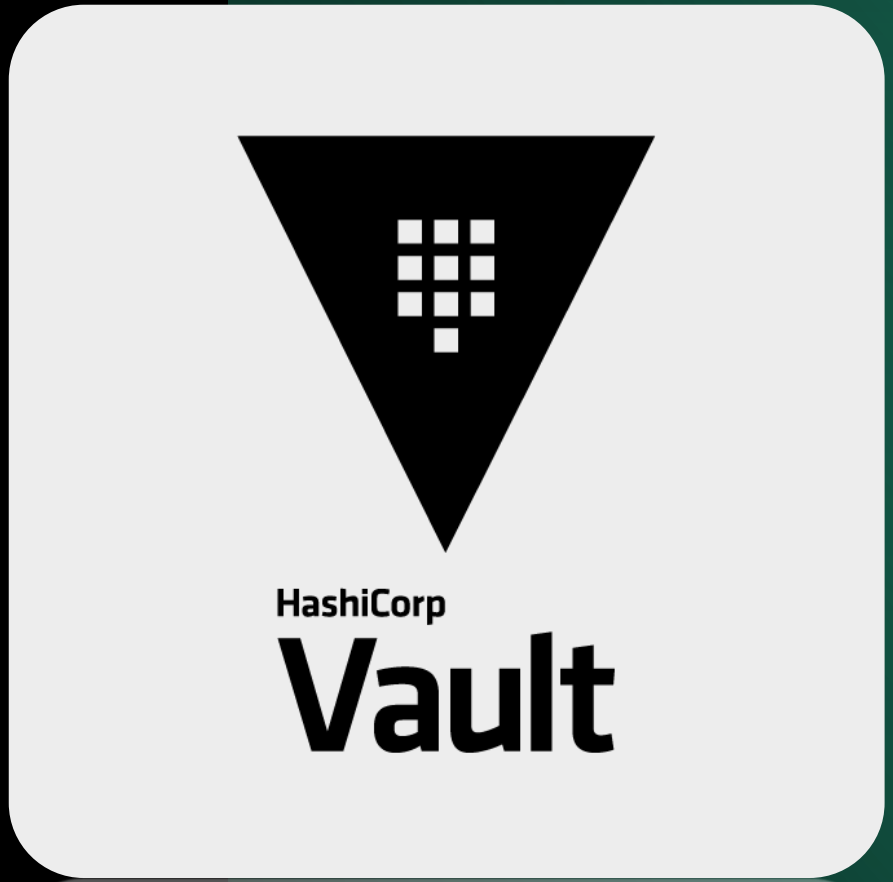
  - Encryption/Cryptography
  - Isolation of sensitive data from non-sensitive data
- You have no excuse not to encrypt your data at-rest
  - Filesystem encryption (Veracrypt, etc.)
  - Object-level encryption (S3, Minio, etc.)
  - Block-level encryption (dm-crypt, etc.)

Encryption

- At-rest
  - Full-disk encryption
  - PGP
  - AGE
- In-transit or in-flight
  - mTLS
  - VPN
- In-use
  - Intel SGX
  - AMD SEV/SME
- Don’t “roll your own” or make “in-house” encryption utilities
  - You will screw it up.

Notable Product

- Leaders in the space
- Secrets management is *key* for securing other aspects of your systems
  - Where do you get your encryption keys?
  - JIT database credentials
  - Strong auditing



Next: Integrity

- In·teg·ri·ty
  - *Noun*
  - internal consistency or lack of corruption in electronic data.
- Identity
- Code Integrity
- Data Integrity
- Non-repudiation (identity management)

Identity and Access Management (IAM)

- Identity and access management (IAM) is a centralized and consistent way to manage user identities (i.e., people, services, and servers), automate access controls, and meet compliance requirements across traditional and containerized environments.[1]

[1] https://www.redhat.com/en/topics/security/what-identity-and-access-management-iam

IAM Options

Provider if you're on a single infra

AWS

GCP

Azure

Third-party identity provider

OIDC, SAML

Github, GitLab

LDAP

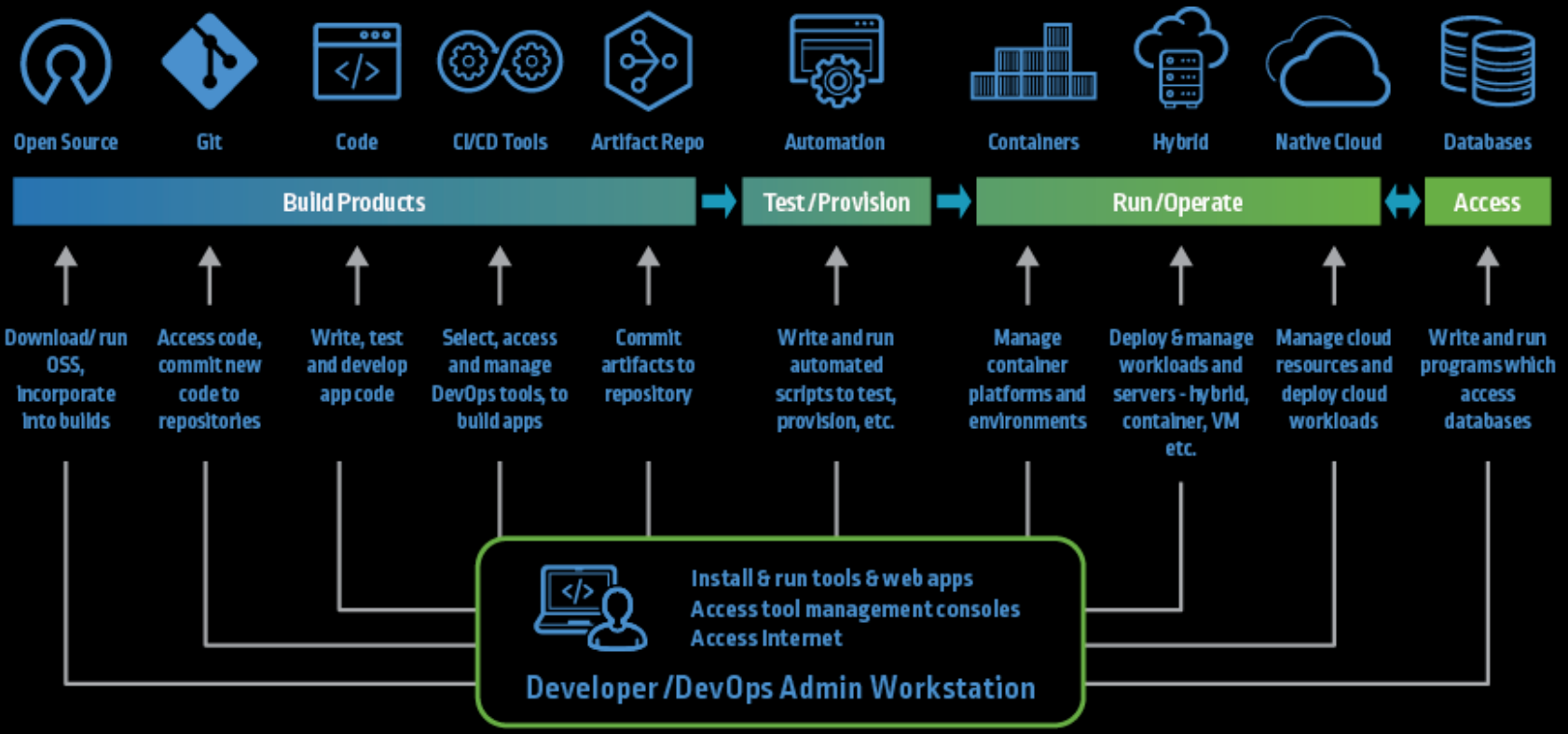
Single Sign-On (SSO)

Keycloak

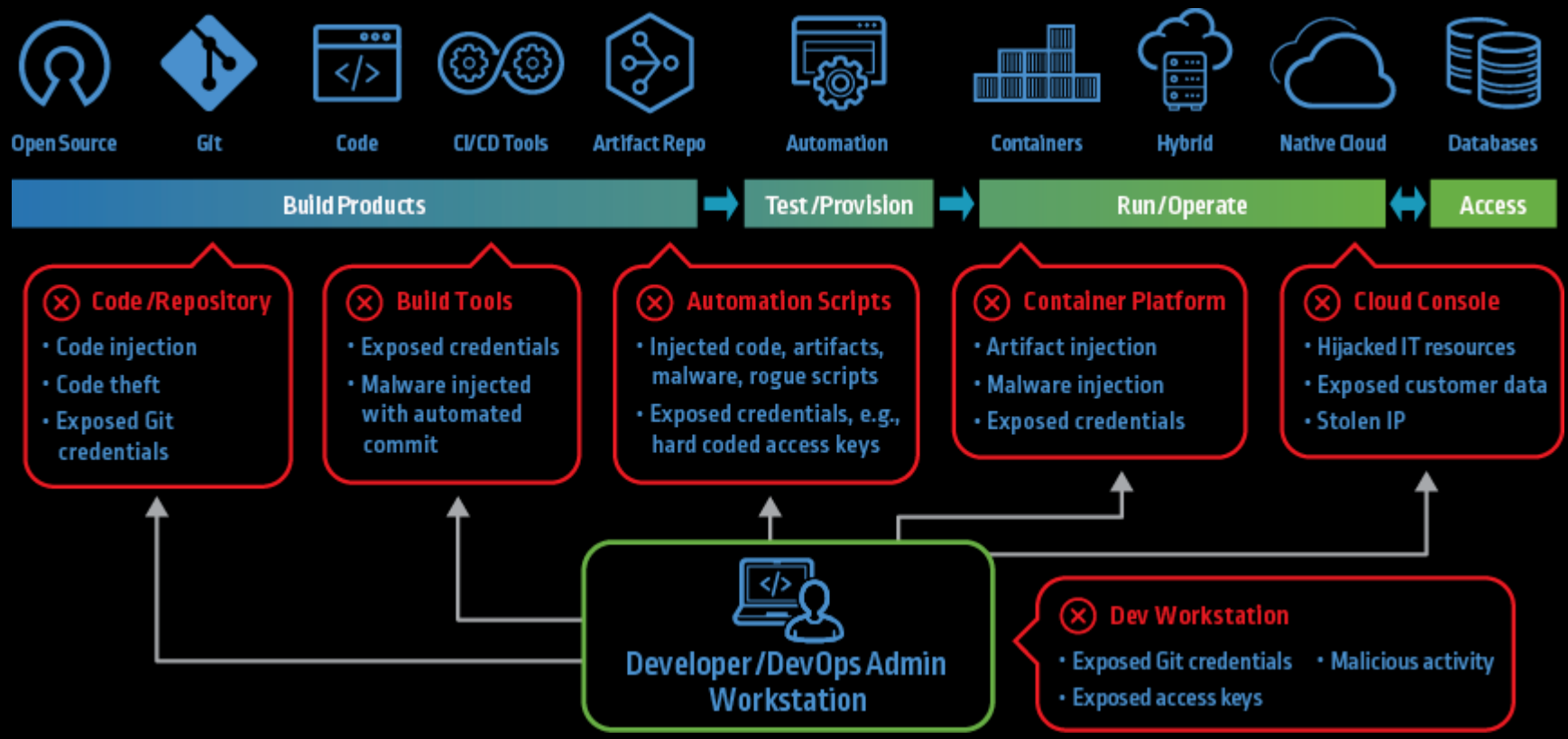
Notable Product

- Open-source IAM, SSO
- Federation
- Strong authentication
- User management
- Fine-grained authorization

Code integrity



Code integrity



Often forgotten: Availability

- High-availability
  - Two is one, one is none
- COOP
  - Continuity of Operations, multi-region, failover
- Scale
  - Often overlooked… If you have under-provisioned your server and application resources, you could suffer from a “hug of death”
  - Some ways to beat this:
    - Serverless functions scale based on inbound requests, but cost more
    - Kubernetes-specific solutions like HPA, Cluster Autoscaler

Certifications, Accreditations, and Compliance

FEDRAMP

SOC2

HIPAA

FIPS

GDPR

Additional Considerations

Certifications, Accreditations, and Compliance	Cost	Open-source risks
Software Supply Chain	Separation of concerns	Edge compute

Final Thoughts

- Good security engineering isn't about "silver bullet" security software
- Fundamental organizational changes can help move any additional security practices forward
- Awareness, training, and documentation are powerful tools for mitigating human risks
- Automation, automation, automation!

Discussion