

# RSA® Conference 2019

San Francisco | March 4–8 | Moscone Center

## A Cloud Security Architecture Workshop

**Dave Shackleford**

Sr. Faculty  
SANS Institute  
@daveshackleford

BETTER.



# Cloud Security Architecture Overview

- There are some very well-known architectural principles for cloud that apply to security
- Sadly, there are few industry design frameworks that are accepted for secure cloud architecture
- Over time teaching and consulting, I've found a number of core best practices that can apply to any IaaS cloud design

# CSA's Enterprise Architecture

- Seeks to promote a sound reference architecture with best practices and processes for a secure cloud



# EA Guiding Principles (1)

- Define protections that enable trust in the cloud
- Develop cross-platform capabilities and patterns for proprietary and open source providers
- Will facilitate trusted and efficient access, administration, and resiliency to the customer/consumer
- Provide direction to secure information that is protected by regulations
- The Architecture must facilitate proper and efficient identification, authentication, authorization, administration, and auditability
- Centralize security policy, maintenance operation, and oversight functions

## EA Guiding Principles (2)

- Access to information must be secure yet still easy to obtain
- Delegate or Federate access control where appropriate
- Must be easy to adopt and consume, supporting the design of security patterns
- The Architecture must be elastic, flexible, and resilient, supporting multitenant, multi-landlord platforms
- The Architecture must address and support multiple levels of protection, including network, operating system, and application security needs

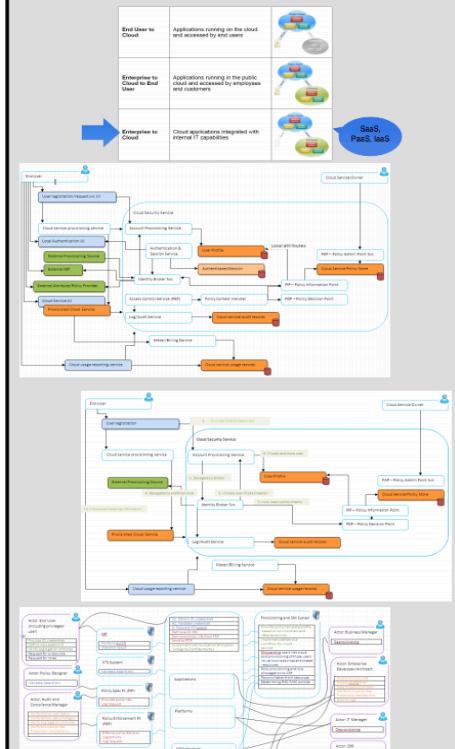
# EA Reference Architecture ... Ouch.



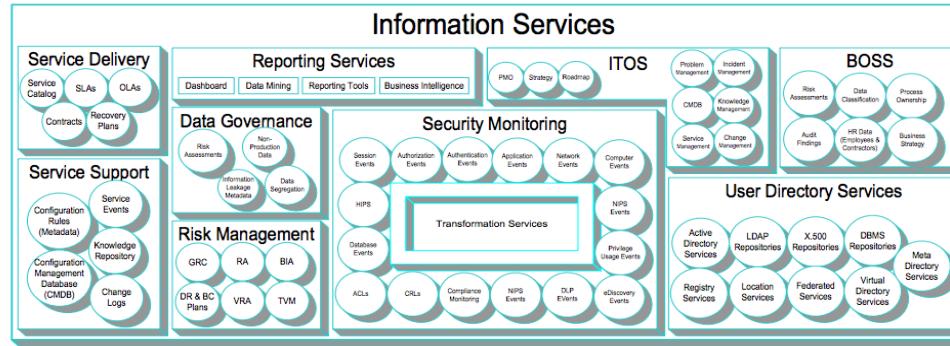
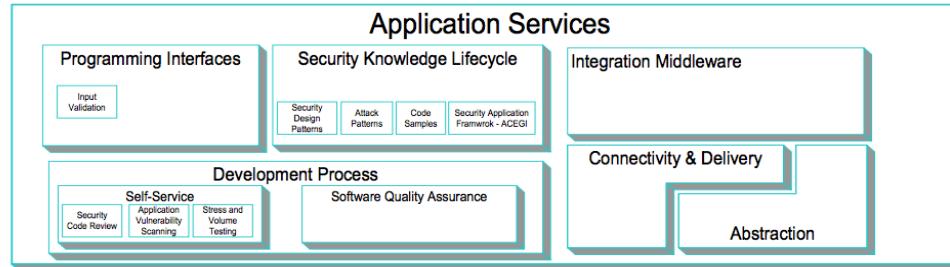
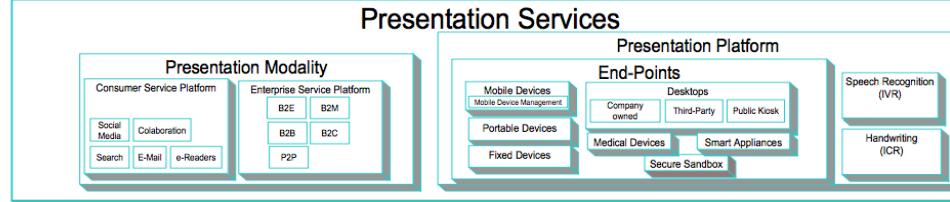
## Guiding Principles

- ❑ Define protections that enable trust in the cloud.
- ❑ Develop cross-platform capabilities and patterns for proprietary and open-source providers.
- ❑ Will facilitate trusted and efficient access, administration and residency to the customer/consumer.
- ❑ Provide direction to secure information that is protected by regulations.
- ❑ The architecture must facilitate proper and efficient identification, authentication, authorization, administration and auditability.
- ❑ Centralize security policy, maintenance operation and oversight functions.
- ❑ Access to information must be secure yet still easy to obtain.
- ❑ Delegate or Federate access control where appropriate.
- ❑ Must be easy to adopt and consume, supporting the design of security patterns
- ❑ The architecture must be elastic, flexible and resilient supporting multi-tenant, multi-landlord platforms
- ❑ The architecture must address and support multiple levels of protection, including network, operating system, and application security needs.

## High Level Use Cases



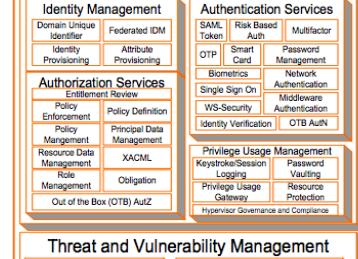
## Reference Architecture



## Security and Risk Management



## Privilege Management Infrastructure



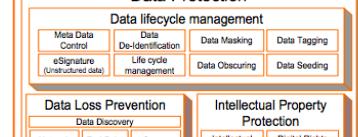
## Threat and Vulnerability Management



## Infrastructure Protection Services

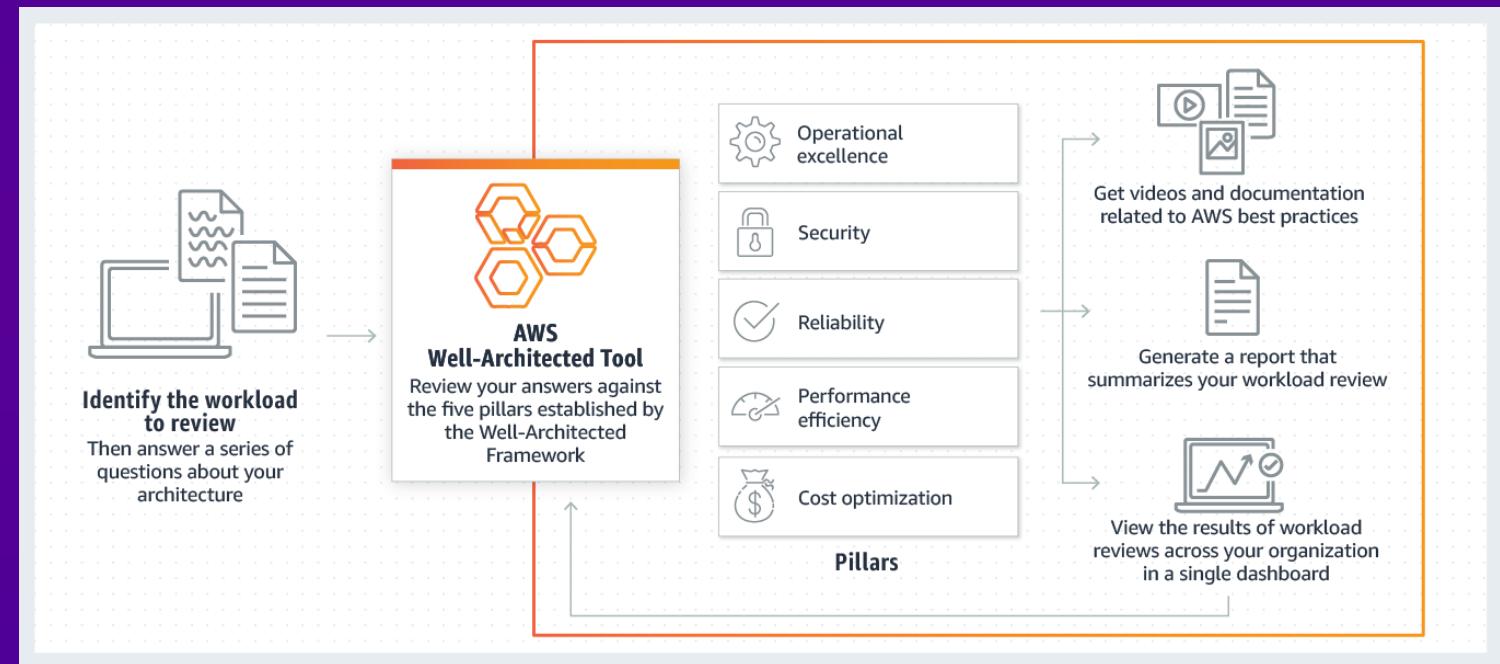


## Data Loss Prevention



# AWS Well-Architected Framework

- AWS has developed a framework called “Well-Architected” that includes five “pillars”:
  - Operational excellence
  - Security
  - Reliability
  - Performance efficiency
  - Cost optimization
- Offers specific best practices and guidance to help design and implement AWS solutions



# A Basic Approach to Sound Cloud Design

- There are many ways of defining “architecture” for the cloud
- Core considerations include:
  - Network connectivity
  - Availability and redundancy
  - Resilience
  - Scalability
  - And so on
- We've broken cloud architecture into seven themes that must be followed in all cases (next)

# SANS Cloud Architecture Principles

- SANS believes the following are the most critical security architecture principles to embed in all designs:
  - Build in security at every layer
  - Think “components”
  - Design for failure
  - Design for elasticity
  - Make use of different storage options
  - ALWAYS think of “feedback loops”
  - Focus on CSA: Centralization, Standardization, Automation

# Build in Security at Every Layer (1)

- Every cloud architecture is composed of unique layers that can be coupled and integrated (or not)
- To some degree, each layer must be “self-defending”
- As discussed on our first day, the new cloud infrastructure and application stack have a number of components
- Each layer needs some sort of security integrated and applied to build a sound “defense-in-depth” architecture for the cloud
- Depending on the layer, some will be applied in-house, others in the CSP environment

# Build in Security at Every Layer (2)

“Stack” Layer	Controls
Application Logic + Presentation	WAF, IAM, Scans/Pen tests
Operating Systems	Configuration, Vulnerability Scanning, Backups, user/privilege management
Data	Encryption, Backups, DLP
Network	Access Controls, Firewalls, Routing, DDoS Defense
Hypervisor	Configuration, access controls, user/privilege management

# Build in Security at Every Layer (3)

- Given the nature of the cloud, IT changes much more dynamically than ever before
- For this reason, all security measures should ideally be “embedded”
- This means:
  - Defining security in code internally
  - Including security configuration parameters in VM definitions
  - Automating security processes and activities
  - Building continuously monitored environments
- Many of these ideas are realized in a sound DevSecOps strategy

# Think “Components”

- While this may seem obvious at this point in the class, it's a major shift for many security professionals
- We're used to designing IT (and to some degree, security) as “systems”
- While some of that holds, in the cloud, we are actually dealing more with disparate components that can be linked and used together in different ways
- For example, each type of storage within Amazon is a different component with varied security controls

# Think “Components”: One Ring to Rule Them All

- When designing security for each individual component, the major theme you should keep in mind is centralization
- In other words, are there efficiency opportunities to define and manage security for multiple components in one place?



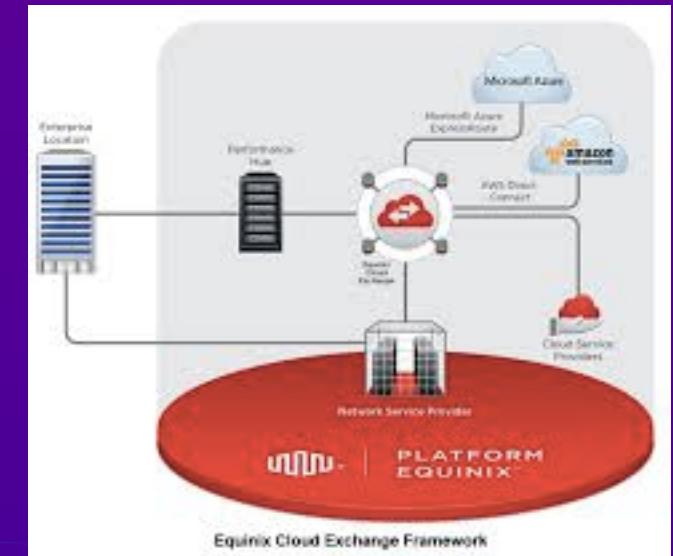
# Think “Components”: The Multi-Cloud Problem

- Many organizations use numerous cloud providers in a hybrid configuration
- What security controls are available in each environment?
- This is a perfect example of why each component needs security “baked in” if possible
- This can add to complexity and operational overhead if not managed well (and early)



# Multi-Cloud Brokers

- Another option for some organizations is using a multi-cloud brokering model
- Many traditional telecommunications carriers offer MPLS cloud access to CSPs
  - AT&T NetBond is one example
- There are other, dedicated services available too
  - Equinix offers cloud brokering with its Cloud Exchange Framework
- These options are usually very expensive
  - May offer greater flexibility and control



# Design for Failure

- Security professionals don't like the word "failure" (probably for obvious reasons)
- However, in the cloud, you are likely to encounter failure more often than you would like:
  - Elasticity issues
  - Configuration issues
  - Cloud provider issues
- Not all of these will be within your control, so you will need to plan for things to go wrong (they will)

# Design for Failure

- When designing for failure, there are two design aspects to consider:
  - Component level: Each component could fail individually or in some combination
  - Architecture level: The entire environment becomes unavailable
- While unlikely, a provider's data center could have a problem
  - Or a backbone carrier or other critical aspect could fail
- You need to design redundancy and availability into **EVERYTHING** within the cloud
  - Or at least those cloud services you care about!

# Design for Elasticity

- One of the cloud's foremost benefits is the ability to rapidly scale up and down as needed for business volume and requirements
- Designing elasticity into your models means considering the following:
  - Vertical or horizontal scaling?
  - What thresholds are appropriate for scaling up and down?
  - How will inventory management adjust to system volume changes?
  - Images new systems are spawned from
  - Where new systems will operate (network locale)
  - Host-based security + licensing

# Storage: Explore Your Options (1)

- There are many types of storage available in the cloud
- Understand each type and which are best suited for your deployment
- Each has its own security options available too
- Revisit data classification and data security policy before planning storage security design
  - Performance matters too, of course!



# Storage: Explore Your Options (2)

- When looking into storage options in the cloud, here are things to consider and evaluate:
  - Does the storage option work for operations and development?
  - Does the storage option have appropriate SLAs and uptime?
  - Does the storage option have adequate redundancy and archival?
  - Does the storage meet performance requirements?
  - Does the storage option provide native encryption capabilities?
  - Does the storage option provide access controls?
  - Does the storage option allow for adequate logging and event generation?
  - What does the storage option cost?
- Consider all the benefits and drawbacks of each before choosing!

# Security: Design in a “Feedback Loop”

- This is one of the most critical elements of cloud security design
- A huge amount of effort goes into securing resources:
  - On their way to the cloud
  - In the cloud
- Given the dynamic nature of cloud computing, things can (and will) change RAPIDLY
- While we’re building in security controls, ensure you plan for alerting and notification capabilities that continually keep us in the loop

# Security “Feedback Loops”= Logging

- Your primary source of feedback is LOGS
- Enable logging everywhere you can:
  - Within the cloud environment/account as a whole
  - For instance, OS types
  - For network platforms
  - For all identity and access management activity
  - For all interconnected services and their activity
- Be sure to secure access to logs, as well

# Security “Feedback Loops”: Other Services

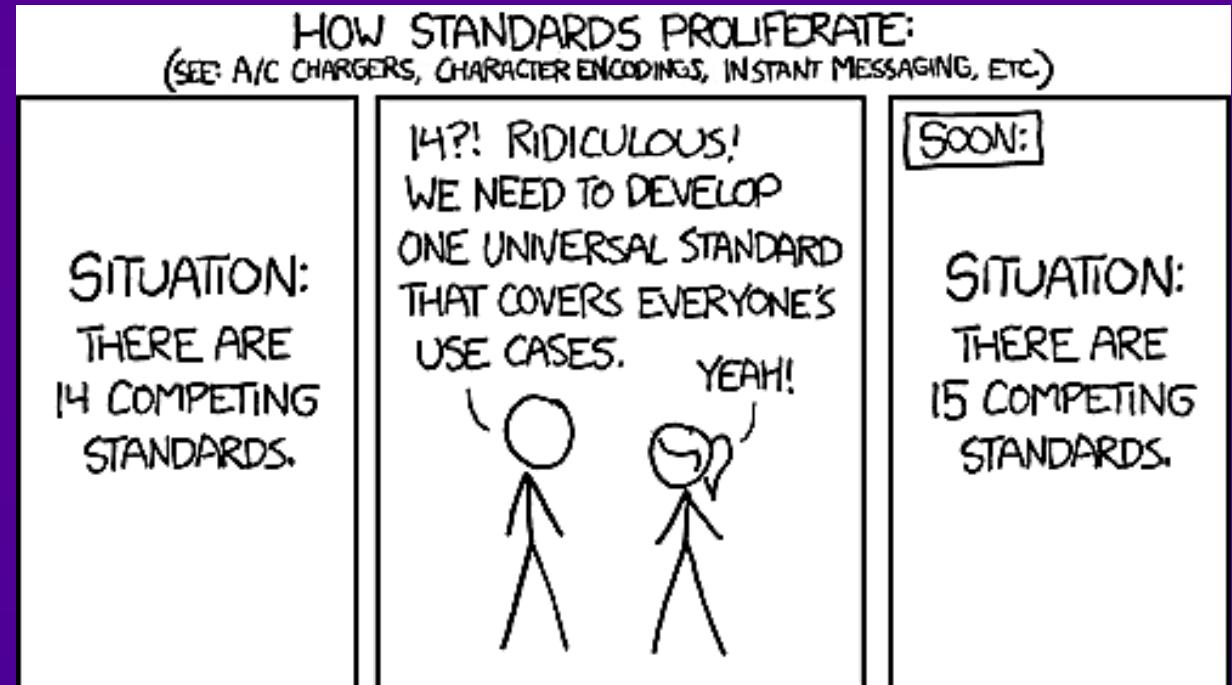
- There are numerous alerting and monitoring mechanisms in major cloud environments
  - CloudTrail logging and Azure Activity Logs
  - CloudWatch alarms
  - Simple Notification Service (SNS) alerts
  - Billing Alerts
- Google StackDriver is an alerting method within the Google cloud
- Azure Monitor is a dashboard that aggregates monitoring like activity logs, diagnostic logs, and metrics
- Azure Advanced Threat Analytics can monitor account behavior

# Centralization, Standardization, Automation: Centralization

- As a final major theme for design and architecture in the cloud, we'll touch on CSA: centralization, standardization, and automation
- Centralization is the idea that you need to look at tools and cloud services that ideally integrate into a single dashboard
- It is **very easy** in cloud deployments to end up with numerous management tools, dashboards, and interfaces to keep up with
- This is not exclusive to security tools—operations and development teams are often faced with the same problem
- Using the same vendor products across cloud environments can help with this (if possible)

# Centralization, Standardization, Automation: Standardization

- Standardization is fairly straightforward conceptually
- When designing for the cloud, look for ways to leverage well-known standards:
  - SAML and OpenID Connect for IAM
  - YAML for configs
  - AES-256+ for crypto



# Centralization, Standardization, Automation: Automation

- Automation is the core idea behind DevOps, and DevSecOps by extension
- Manual efforts in the cloud are doomed to fail in many cases, as the environment changes rapidly
- Security teams should explore ways to automate their security controls and feedback loops whenever possible
- Scripting and orchestration tools can help!

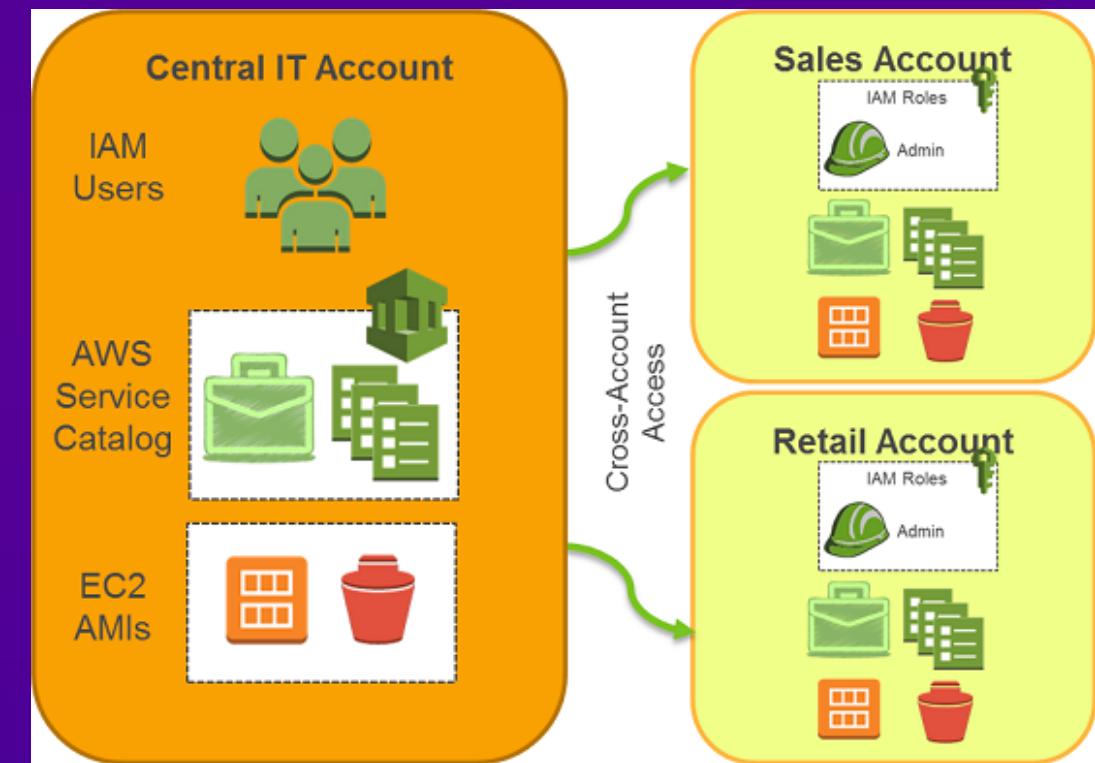


# Managing the Cloud “Blast Radius”

- One of the core security concepts in the world of DevOps and cloud computing is the “blast radius”
- The blast radius is the amount of damage that could be caused if something goes wrong
  - An account or server gets hacked
  - A component fails
- Design your security model in such a way that you limit the damage any one issue could cause

# Multiple Accounts for Limiting Blast Radius

- One cloud security strategy that has emerged in recent years is the use of multiple accounts for limiting blast radius
- Accounts can be created for:
  - Developers
  - Business units
  - Operations
  - Security
- These can then be allowed access to objects and assets in other accounts as needed
- AWS has a service called “Landing Zone” to help set this up
  - A newer service called “Control Tower” is also now available to implement this



# Applying This To Your Organization

- Next week you should:
  - Determine your level of overall architecture maturity
- In the first three months following this presentation you should:
  - Ensure you look into multi-account or subscription architectures
  - Ensure centralized, infrastructure-as-code deployments are planned
- Within six months you should:
  - Have a streamlined, central deployment incorporating DevSecOps principles
  - Ensure all feedback loop and storage controls are optimized