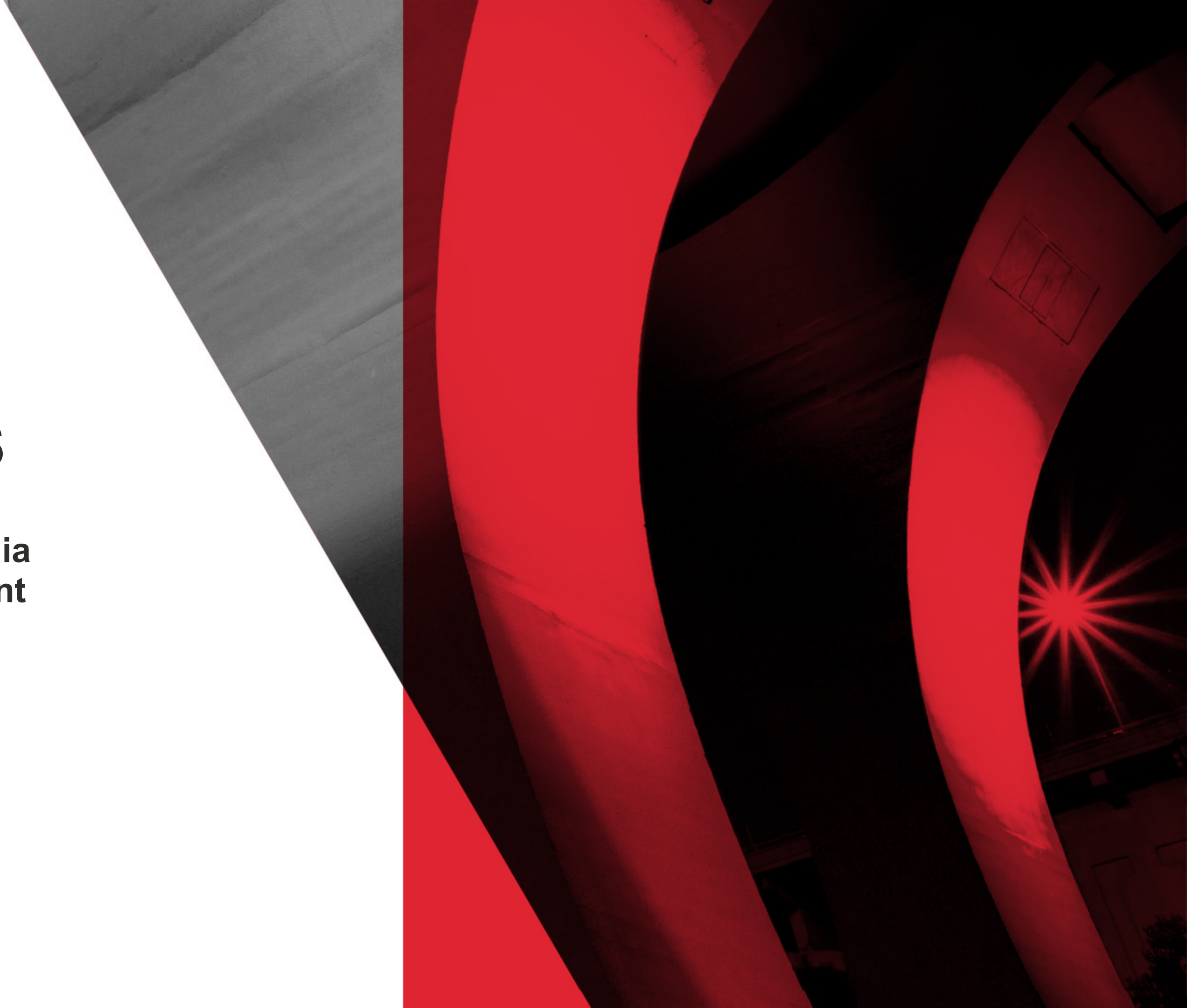
2023 INTERNET2 TECHNOLOGY exchange

Cloud Security by Default
The Rewards of Standards and Infrastructure as Code

George Holbert, Matthew Stout
University of California, Office of the President

ABOUT US

University of California
Office of the President



About Us

University of California:
- 10 Campuses - undergraduate/graduate
- 6 Academic Health Centers
- 3 National Laboratories
- >230,000 employees
- >280,000 students

University of California, Office of the President (UCOP):
- Systemwide infrastructure services
- Local infrastructure services
- >2000 employees
- >$2M annual cloud provider utility
- > 50 cloud accounts

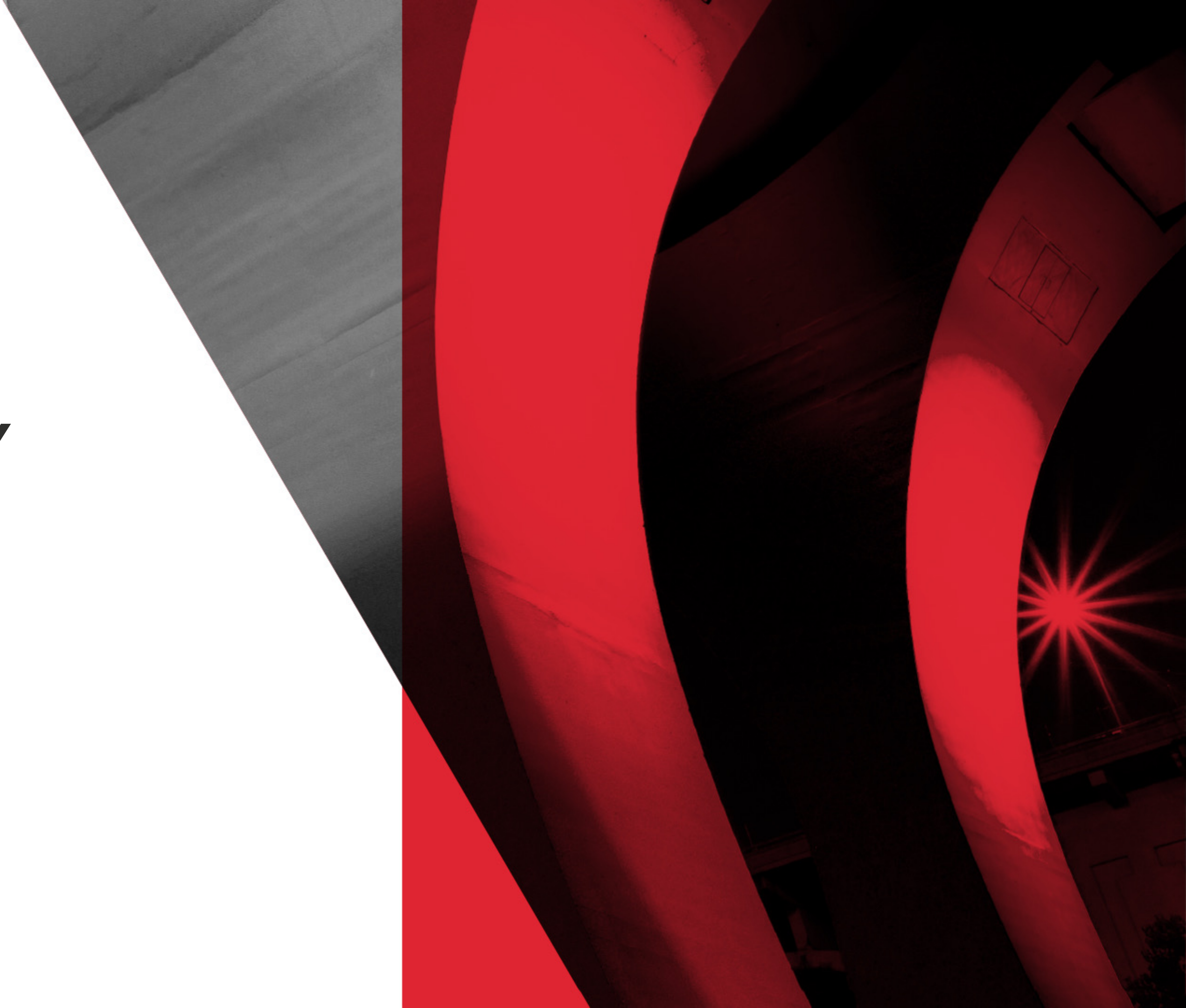
George Holbert
Cloud Engineer

Matt Stout
Cloud Architect

Our Cloud Journey

< 2019	2020	2021	2022	2023
Cloud Exploration	**Assessment**	**Foundations**	**Scaling**	**<u>Even More</u>**
• Training for all • Freeform builds	• Established FinOps • Analyzed Existing Workloads • Easy Optimizations Migrations: • UC Systemwide Payroll System	• Cloud Standards • Automated Compliance • Security Framework Migrations: • Data Center (secondary)	• Central IT Alignment • Retrofits • Intermediate Optimizations Migrations: • UC Systemwide Student Application System • Mainframe Workloads • ECS Cluster Containers	• Containers • Cloudfront • Cloud team and Infrastructure both supporting Cloud services Migrations: • UC Systemwide Retirement and Pension Adminstration System • Starting to move one of DCs in full to Cloud in 2024

SECURITY BY DEFAULT



Security By Default

- We now have ~50 accounts and more staff working in the Cloud

- The Cloud is now our default location for new services
  - More teams than ever with access to cloud accounts
  - More teams building resources
  - More potential for chaos and security incidents

- We also require consistency
  - We have centralized teams database administration, networking, middleware, infrastructure, security, cloud and fin ops that support all managed accounts
  - Even if security was not a concern we cannot be effective if we let every account owner choose their own standards

Security by Default

- With Security by Default we get
  - Consistency across accounts
  - Ability to audit or review risk faster per account or service
  - Most baseline standards met before anything is built
  - Builds are faster once we have a standard and reuse lessons learned or we have standard solutions
  - New solutions and services deal with Security standards or controls from the beginning not later just before going live

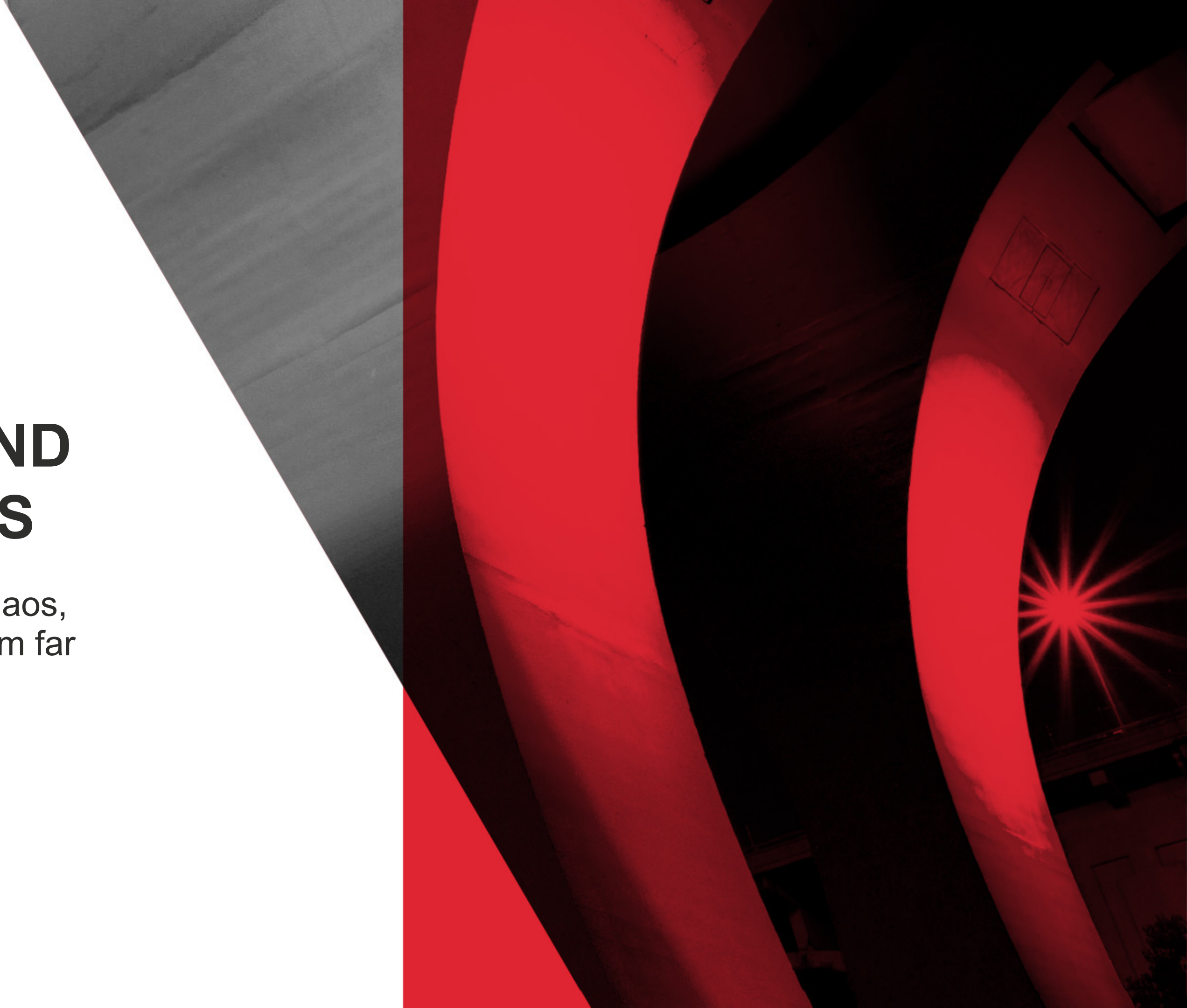
Security by Default

- Today we will look at three of the main ways we implement our Security by Default

- However, first lets look at the complexity of Security and just how much we are trying build into our cloud offerings

SECURITY AND STANDARDS

They can be their own chaos, even while saving you from far worse chaos



Our Tool Box

DATADOG

AWS Trusted Advisor

QRadar IBM

RAPID7

FireEye

AWS Organizations

AWS Security Hub

Trellix

Amazon Inspector

AWS Well-Architected Tool

Amazon GuardDuty

AWS IAM Identity Center

aws

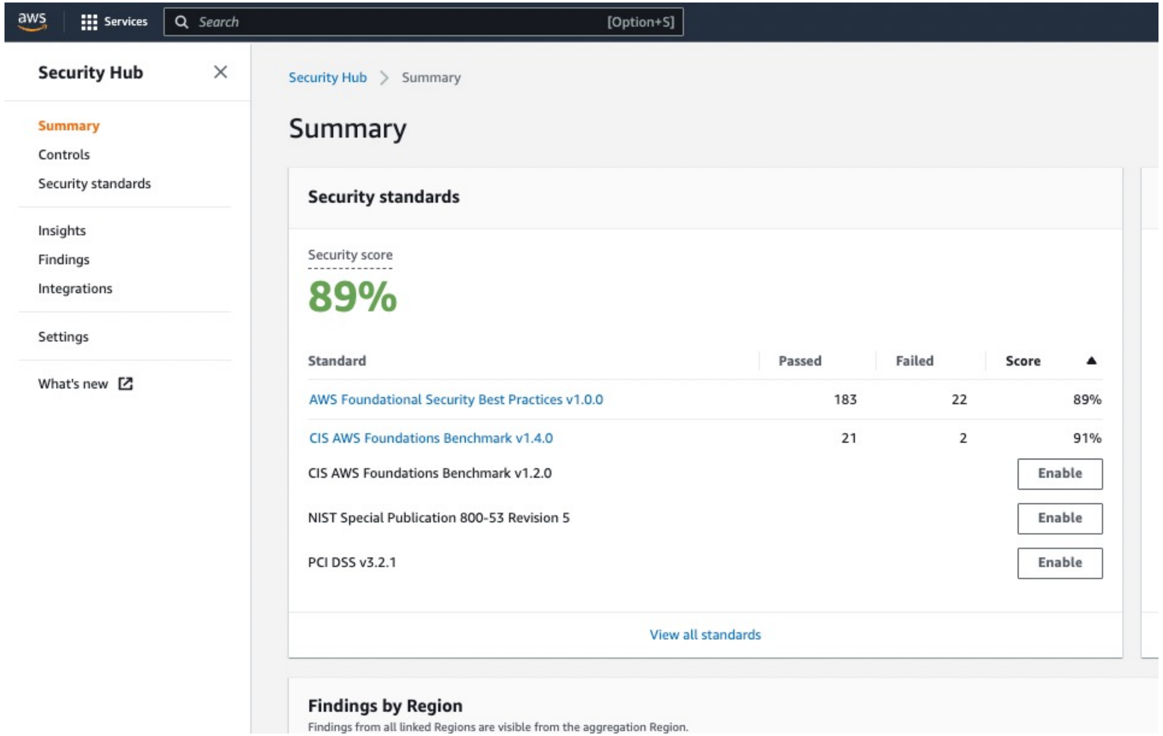
Our Tool Box

- That previous slide is not an attempt to dazzle, nor confess that we might have too many tools, rather we want to make a case for how difficult all this would be without our standardization and security setup in all accounts

- Next, we will look at just a few of those tools and how they are useful to us, but also how complex they can be

Security Hub

- AWS Security Hub is a cloud security posture management (CSPM)
- AWS maintains standards with hundreds of checks
- Helps prioritize the most sensitive issues
- Gives your overall status
- ~250 checks (just the ones we have in use!



Summary

Security standards

Security score

89%

Standard	Passed	Failed	Score ▲
AWS Foundational Security Best Practices v1.0.0	183	22	89%
CIS AWS Foundations Benchmark v1.4.0	21	2	91%
CIS AWS Foundations Benchmark v1.2.0			Enable
NIST Special Publication 800-53 Revision 5			Enable
PCI DSS v3.2.1			Enable

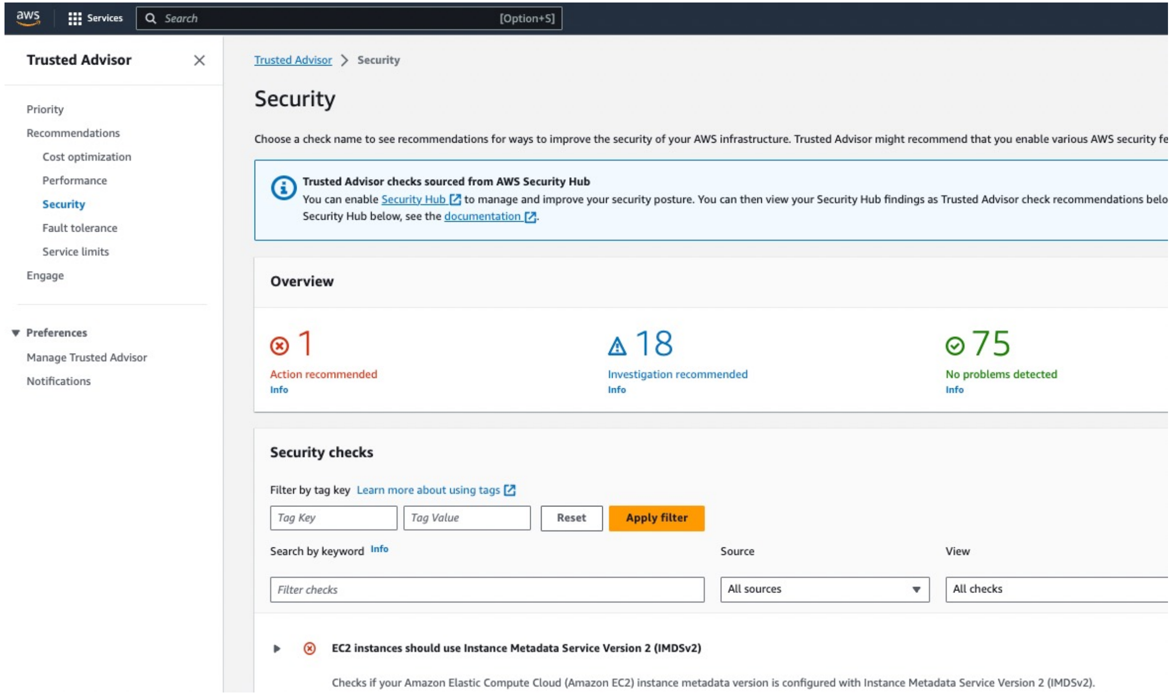
View all standards

Findings by Region

Findings from all linked Regions are visible from the aggregation Region.

Trusted Advisor

- Checks to aid in following AWS best practices
- ~200 checks
- Enterprise Support customers: AWS Trusted Advisor Priority
- Does require support to get all checks



Security

Choose a check name to see recommendations for ways to improve the security of your AWS infrastructure. Trusted Advisor might recommend that you enable various AWS security fe

Trusted Advisor checks sourced from AWS Security Hub

You can enable Security Hub to manage and improve your security posture. You can then view your Security Hub findings as Trusted Advisor check recommendations belo Security Hub below, see the documentation.

Overview

1 Action recommended Info

18 Investigation recommended Info

75 No problems detected Info

Security checks

Filter by tag key Learn more about using tags

Search by keyword Info: Filter checks

Source: All sources

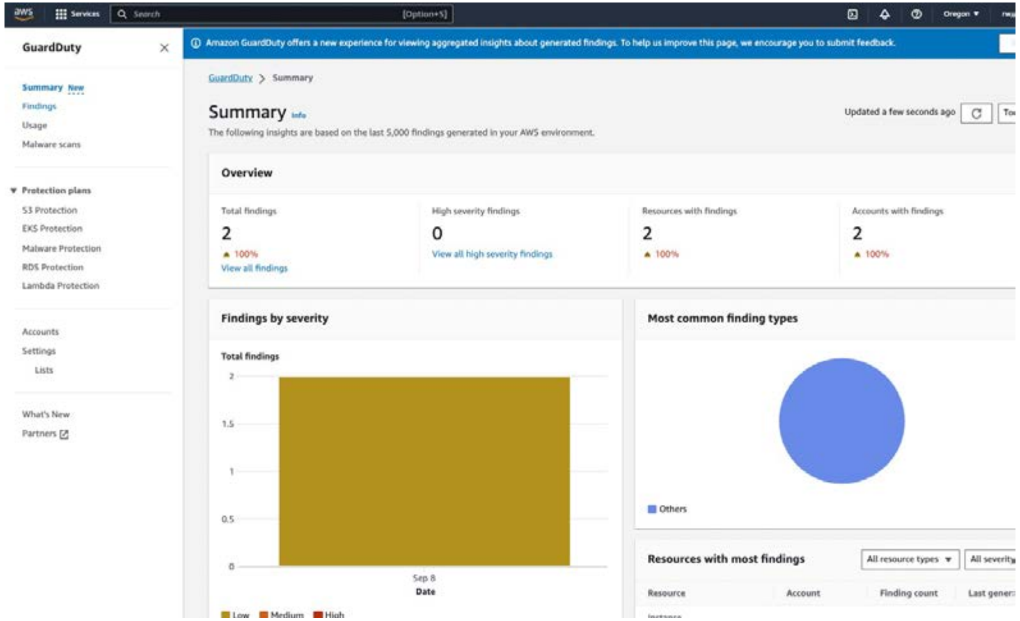
View: All checks

EC2 instances should use Instance Metadata Service Version 2 (IMDSv2)

Checks if your Amazon Elastic Compute Cloud (Amazon EC2) instance metadata version is configured with Instance Metadata Service Version 2 (IMDSv2).

GuardDuty

- Amazon GuardDuty is a threat detection service that continuously monitors your AWS accounts and workloads for malicious activity and delivers detailed security findings for visibility and remediation
- We see variable findings but it can be dozens a month across accounts at times



Amazon GuardDuty offers a new experience for viewing aggregated insights about generated findings. To help us improve this page, we encourage you to submit feedback.

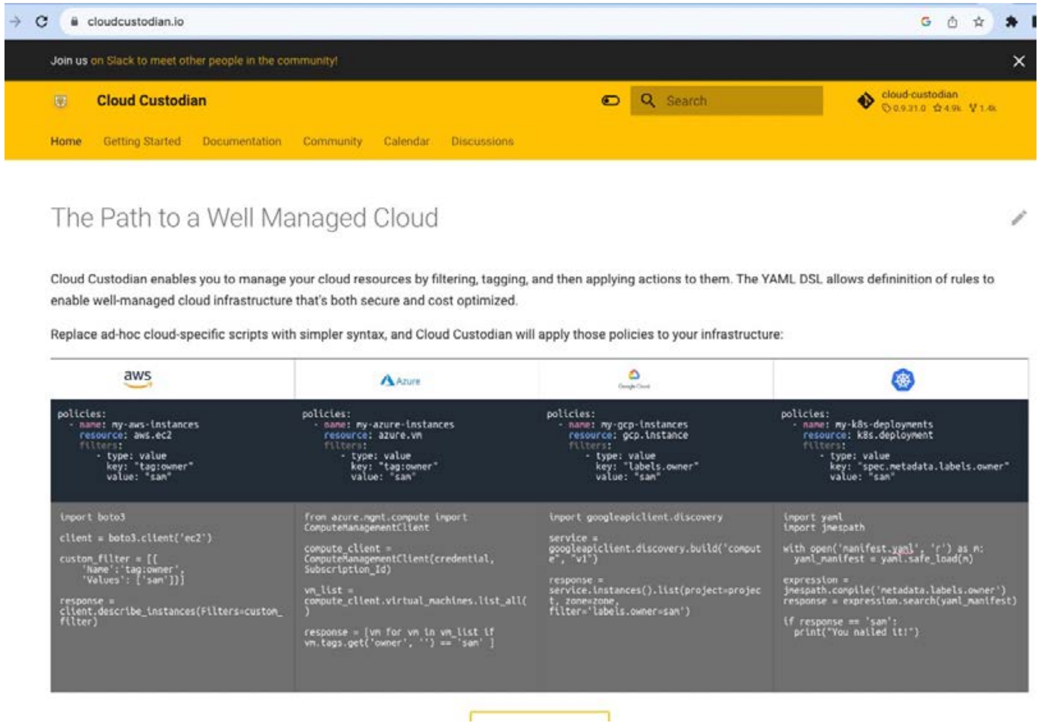
Summary

The following insights are based on the last 5,000 findings generated in your AWS environment.

Total findings	High severity findings	Resources with findings	Accounts with findings
2	0	2	2
▲ 100%	View all high severity findings	▲ 100%	▲ 100%
View all findings			

Cloud Custodian

- Flexible, simple configuration syntax to find resources and apply actions.
- Unlike other tools you can easily take actions on a resource
  - Such as set IMDS v2 on in all accounts
- We have ~30 policies to set encryption on if not, automate S3 access logging, and much much more



cloudcustodian.io

Join us on Slack to meet other people in the community!

Cloud Custodian

Search

cloud-custodian 0.9.31.0 4.9k 1.4k

Home Getting Started Documentation Community Calendar Discussions

The Path to a Well Managed Cloud

Cloud Custodian enables you to manage your cloud resources by filtering, tagging, and then applying actions to them. The YAML DSL allows defininition of rules to enable well-managed cloud infrastructure that's both secure and cost optimized.

Replace ad-hoc cloud-specific scripts with simpler syntax, and Cloud Custodian will apply those policies to your infrastructure:

aws

```yaml
policies:
  - name: my-aws-instances
    resource: aws.ec2
    filters:
      - type: value
        key: "tag:owner"
        value: "sam"
```

```python
import boto3

client = boto3.client('ec2')

custom_filter = [{
    'Name':'tag:owner',
    'Values': ['sam']}]

response =
client.describe_instances(Filters=custom_
filter)
```

Azure

```yaml
policies:
  - name: my-azure-instances
    resource: azure.vm
    filters:
      - type: value
        key: "tag:owner"
        value: "sam"
```

```python
from azure.mgmt.compute import
ComputeManagementClient

compute_client =
ComputeManagementClient(credential,
Subscription_Id)

vm_list =
compute_client.virtual_machines.list_all(
)

response = [vm for vm in vm_list if
vm.tags.get('owner', '') == 'sam' ]
```

Google Cloud

```yaml
policies:
  - name: my-gcp-instances
    resource: gcp.instance
    filters:
      - type: value
        key: "labels.owner"
        value: "sam"
```

```python
import googleapiclient.discovery

service =
googleapiclient.discovery.build('comput
e", "v1")

response =
service.instances().list(project=projec
t, zone=zone,
filter='labels.owner=sam')
```

```yaml
policies:
  - name: my-k8s-deployments
    resource: k8s.deployment
    filters:
      - type: value
        key: "spec.metadata.labels.owner"
        value: "san"
```

```python
import yaml
import jmespath

with open('manifest.yaml', 'r') as m:
  yaml_manifest = yaml.safe_load(m)

expression =
jmespath.compile('metadata.labels.owner')
response = expression.search(yaml_manifest)

if response == 'sam':
  print("You nailed it!")
```

https://cloudcustodian.io/

Security and Operations Tools

- In addition to those there are more than we can cover in this talk
- They included
    - Host vulnerability management solution
    - Web application Scanning
    - Endpoint Detection and Response EDR
    - Security Information and Event Management SIEM
    - Central authentication and access logging and alerting
    - Central Performance Monitoring Solution
    - CloudWatch
    - And more

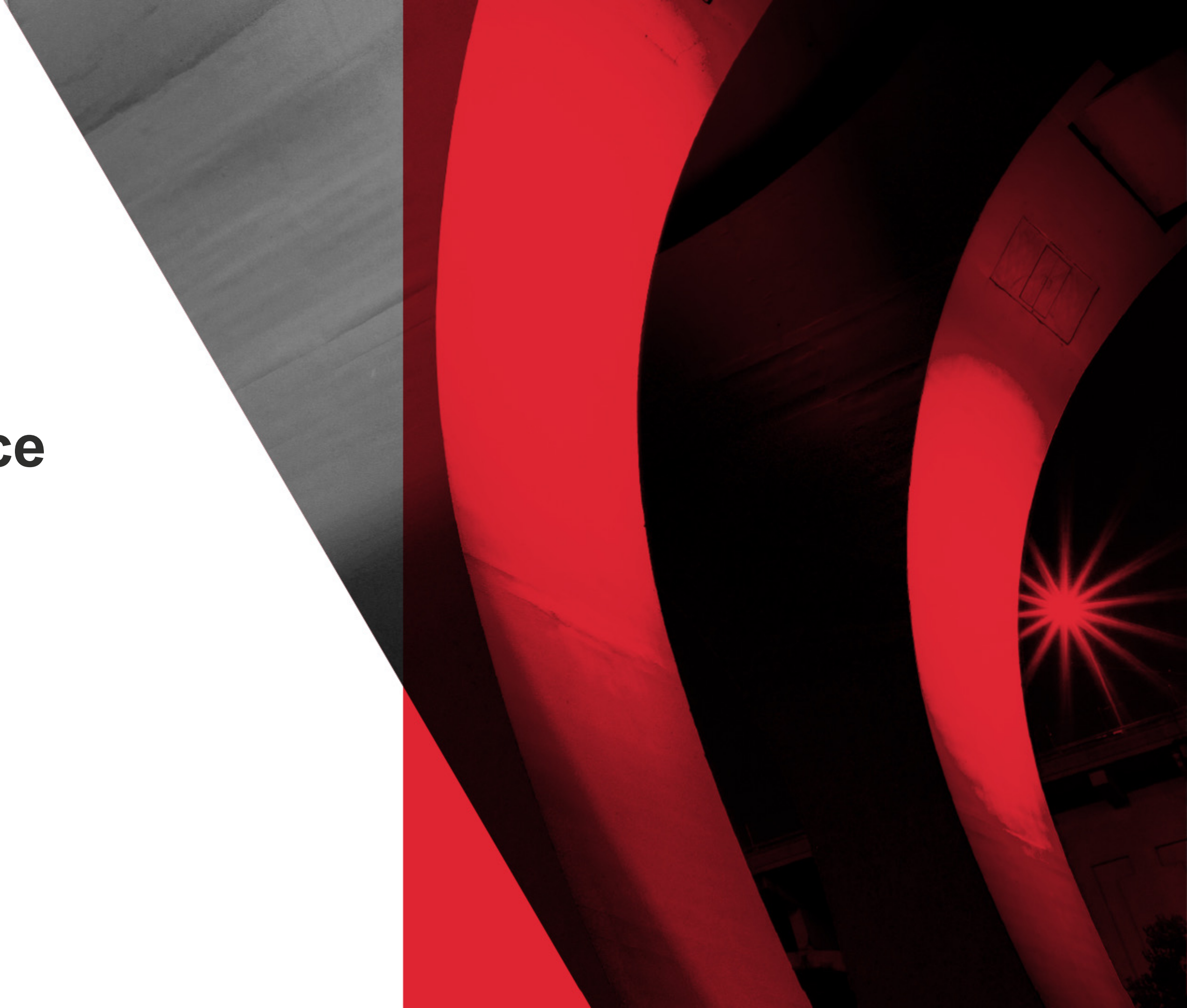
Summary By the Numbers

- Security By Default Means from the start we want to meet all of these:
  - Security Hub: ~250 checks
  - Trusted Advisor: ~200 checks
  - Cloud Custodian: ~30 policies active

- Over 500 Checks not including vulnerability scanning, automation to install agents for scanning, etc.

- Up Next: Three ways we handle Security by Default
  - AWS Account Setup
  - Infrastructure as Code (IaC)

1) Governance



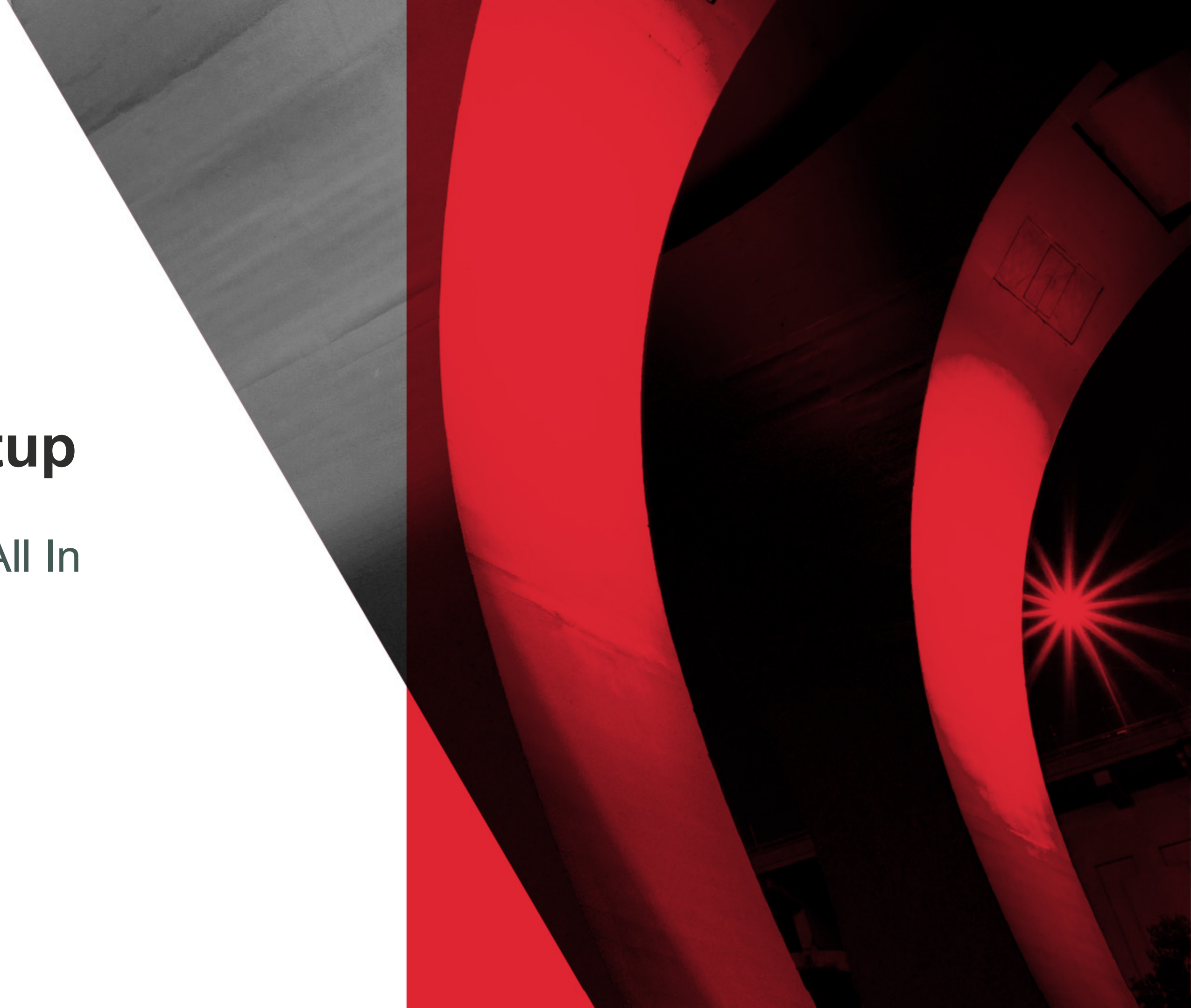
Account Creation and Setup

- A must for
    - managing so many accounts
    - shared teams and
    - adopting standards...

- As important as any software or technical solution
    - See that timeline at the beginning of this slide
    - Our cloud journey had many important steps of setting up standards, roles and responsibilities and building our different teams we never had before

2) Account Setup

Start with Security All In



Account Creation and Setup

- We use AWS Organizations
  - Our automation is a bit lacking; several separate processes and a few manual items, however, is very similar to Control Tower
    - Separate network, logging, security accounts
    - AWS SSO, SCPs, etc.
- We start from day one in all new managed accounts with all of standard controls and setup
  - Security Hub, AWS GuardDuty, Enterprise Support, and all the rest
  - We rarely do POC accounts and if we do we still do all the same setup
  - Rarely allow unmanaged accounts, mostly legacy or special use
  - This means as new services are created or resources deployed we catch configuration or security issues at the start

Infrastructure as Code (IaC)

How we build in AWS

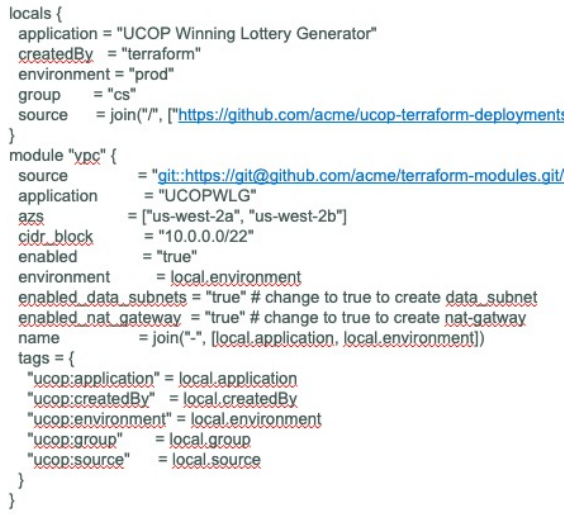
What Is Infrastructure as Code (IaC)?

- Infrastructure as Code (IaC) is the managing and provisioning of infrastructure through code instead of through manual processes.
- Examples include:
  - AWS CloudFormation, Red Hat Ansible, Chef, Puppet, SaltStack and Terraform.

- With IaC we can build:
  - VPCs in a few minutes
  - Same for EC2, Load Balancers, ECS+Fargate, and much more
  - This allows a few experts to build faster when they are the ones that must
  - We are working to make more of our standard builds things all on the Cloud and Infrastructure teams can build

What We Use

- Terraform
  - Terraform is our primary tool of choice for all our AWS Cloud resources
  - Terraform codifies cloud APIs into declarative configuration files
  - Great public examples, modules, and code
    - https://registry.terraform.io/
  - Use Modules
    - Modules have the main code and do all the creations and we only need a small block of code to supply the variables and options we want on a resource
  - See our slides from last your for more!



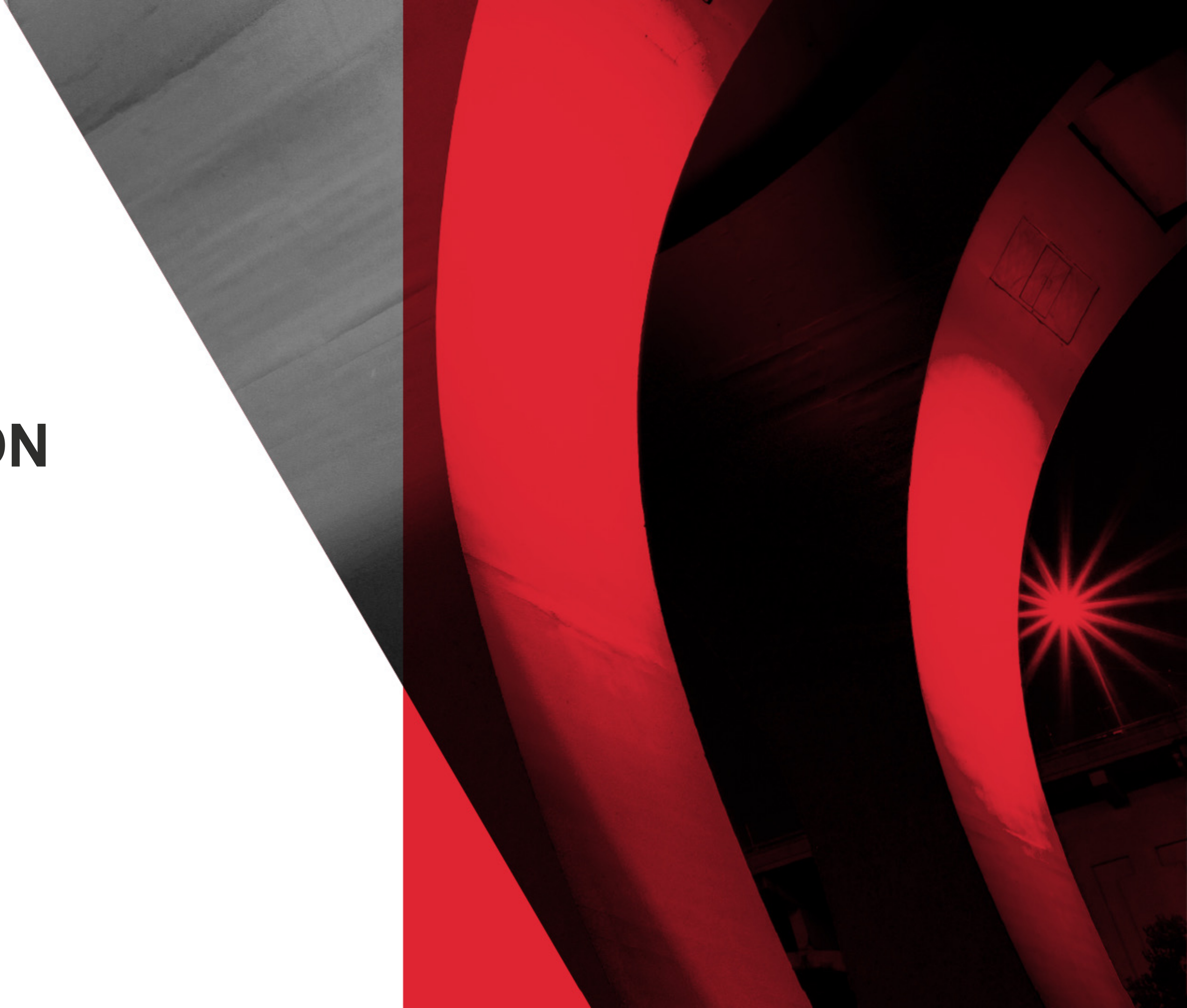
```
locals {
  application = "UCOP Winning Lottery Generator"
  createdBy   = "terraform"
  environment = "prod"
  group       = "cs"
  source      = join("/", ["https://github.com/acme/ucop-terraform-deployment
}
module "vpc" {
  source              = "git::https://git@github.com/acme/terraform-modules.git/
  application         = "UCOPWLG"
  azs                 = ["us-west-2a", "us-west-2b"]
  cidr_block          = "10.0.0.0/22"
  enabled             = "true"
  environment         = local.environment
  enabled_data_subnets = "true" # change to true to create data_subnet
  enabled_nat_gateway  = "true" # change to true to create nat-gatway
  name                = join("-", [local.application, local.environment])
  tags = {
    "ucop:application" = local.application
    "ucop:createdBy"   = local.createdBy
    "ucop:environment" = local.environment
    "ucop:group"       = local.group
    "ucop:source"      = local.source
  }
}
```

What We Use

- IaC is not just terraform or similar solutions!

- Other Examples – Other ways we to add IaC:
  - Automated container deployments via CodePipelines/commits to version control
    - Allows app developers to write code and deploy it
  - Sytems manager Documents
    - Automate security agent installs, join domain, and more. Mostly windows for us now
  - Puppet for Linux
    - Harden OS, install security agents, and more

CONCLUSION



Security By Default

- Reduce the time involved in reworking or redesigning solutions
- Create in an environment with all of our standards and enforcment tools
    - Sometimes slows us down at the start, however, this resolves issues early and not while rushing to release production

- Security we can mostly forget and still have in place
    - Allow more hands to start building while still keeping us secure
    - Security by Default!

Questions?

UCOP @ Technology Exchange

- Join us for our 2023 Technology Exchange presentations by UCOP team members:
  - **Moving from VM to Cloud Native Containers** with Khalid Ahmadzai, Tuesday 11:20am-12:10pm
  - **Cloud Security By Default** with Matthew Stout and George Holbert, Thursday 10:20am-11:10am
  - **Control Chaos with IaC & Automation** with Josh Whitlock, Thursday 1:40pm-2:30pm
- 2022 Technology Exchange presentation by UCOP's own Khalid Ahmadzai, Kari Robertson, Matt Stout
  - **Moving from Cloud Chaos to Standards:**
  - https://internet2.edu/wp-content/uploads/2022/12/techex22-Cloud-MovingfromCloudChaostoStandards-AhmadzaiStoutRobertson.pdf

QUESTIONS