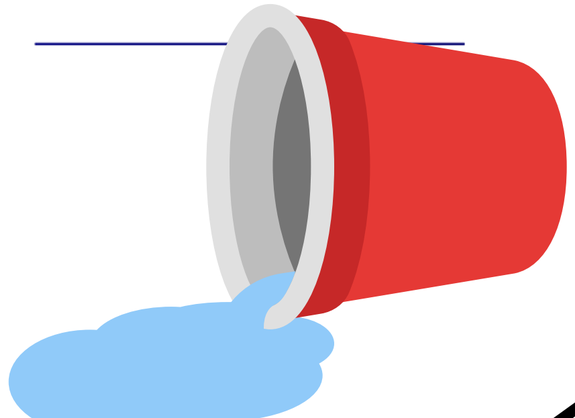
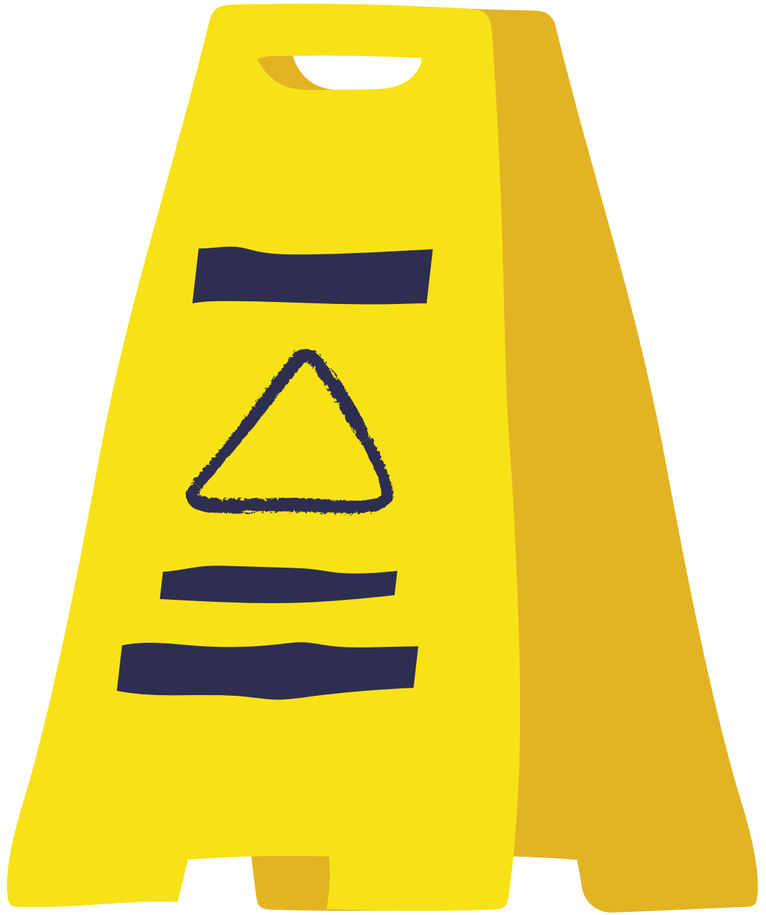
KubeCon | CloudNativeCon

North America 2023



Clean Up
On Aisle Cloud!



Presented by

Boeing Product Security

Lead Cloud Security Architect

Sara J. Johnson

This session is for everyone

(and **all levels** of experience)

BOEING'S CLOUD-NATIVE JOURNEY

Boeing Commercial Airplanes

vulnerability testing & analysis for compliance

- Machine learning log analysis
- Wireless & RF testing
- Security automation
- Platform engineering
- Cloud engineering & GitOps

FOREIGN
OBJECT
DEBRIS

Unintentional code

Unmanaged resources

Untracked resources

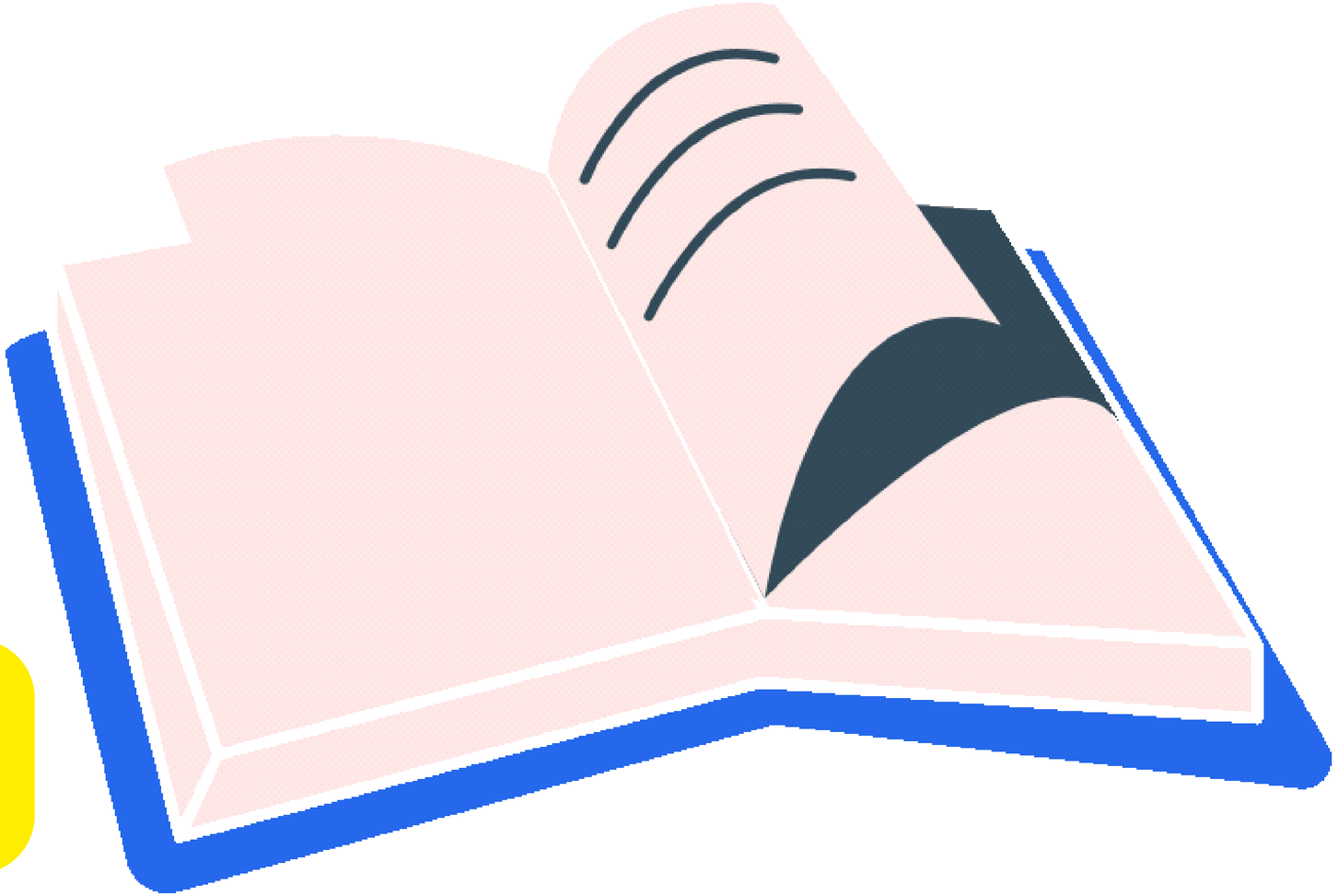
Who cleans up our software & engineering environments?

TRANSITIONING TO THE CLOUD

has rewards

and challenges

Three case studies



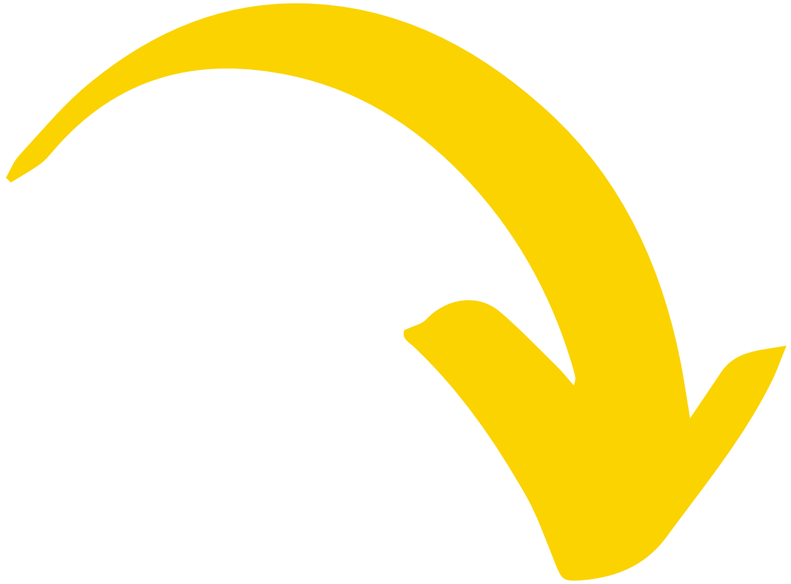
Scenario 1: Rogue Resource

- Alice is a cloud engineer for a platform team
- The team is new to the cloud and small
- The platform supports 300+ developers
- Alice notices:
  - An unnamed, untagged Windows 2012 Instance

Why **is this** a problem?

WARNING

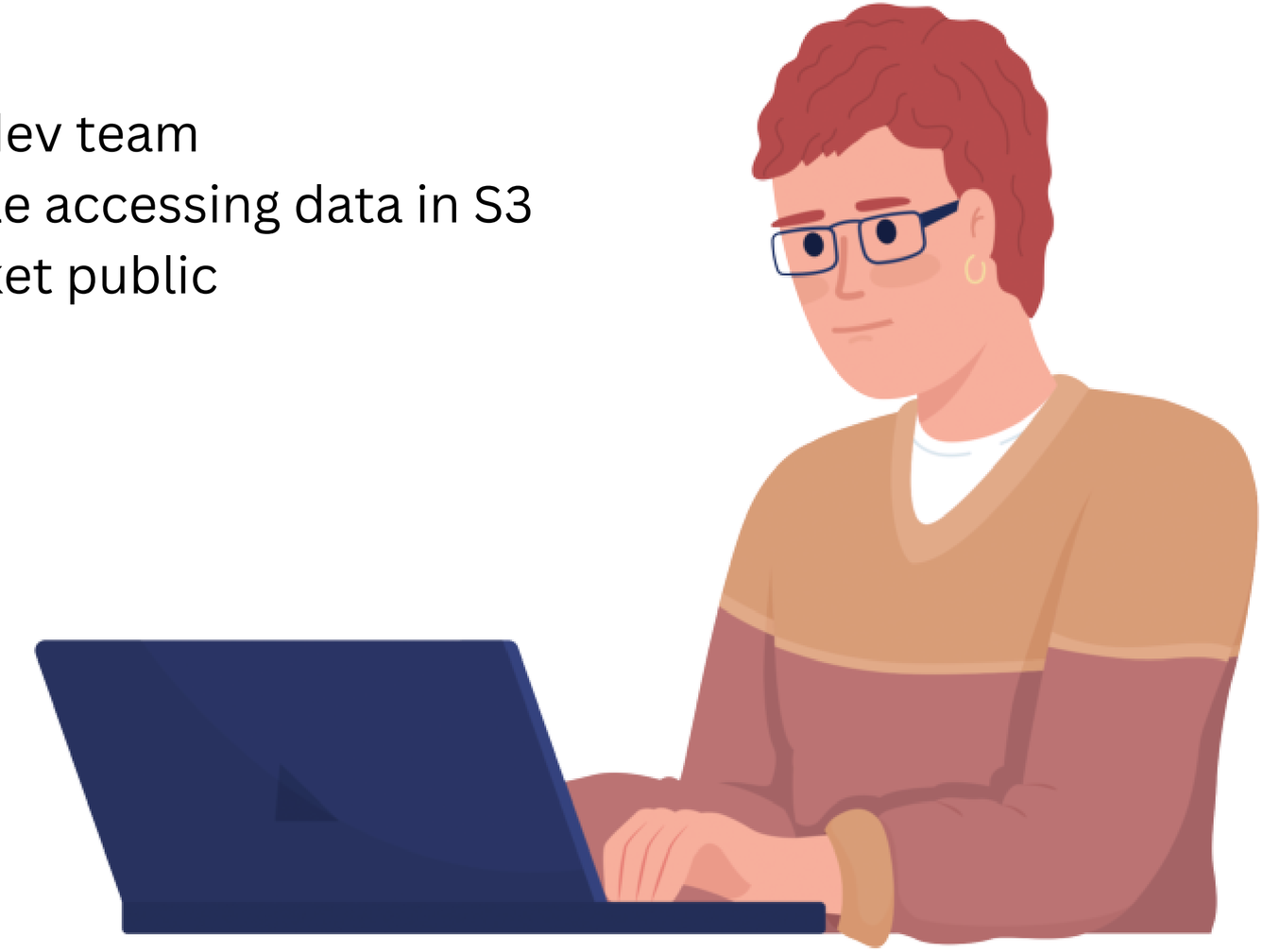
- The team uses Linux, not Windows
- Windows 2012 is an old version
- Image is from an unknown vendor
- Team’s standards weren’t followed



Example of “software FOD”

Scenario 2: Public Resource

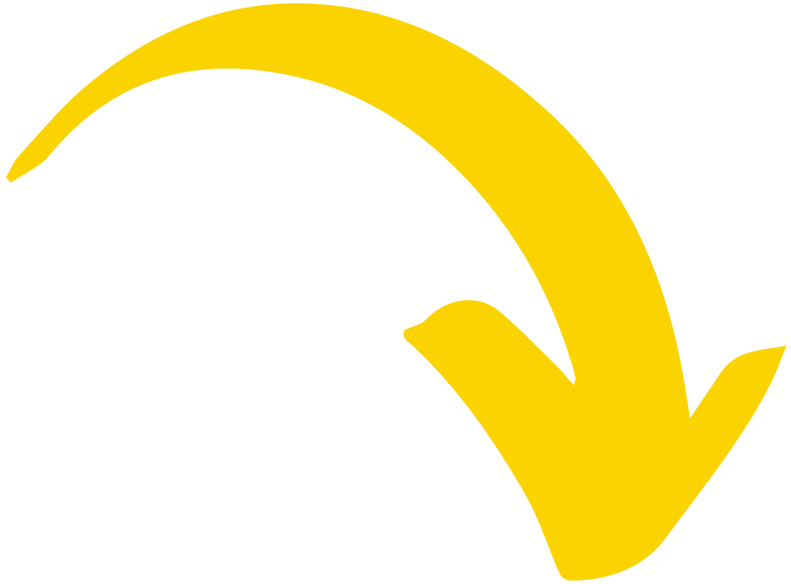
- Bob is an SRE on a dev team
- Bob is having trouble accessing data in S3
- Bob makes the bucket public



Why **is this** a problem?

WARNING

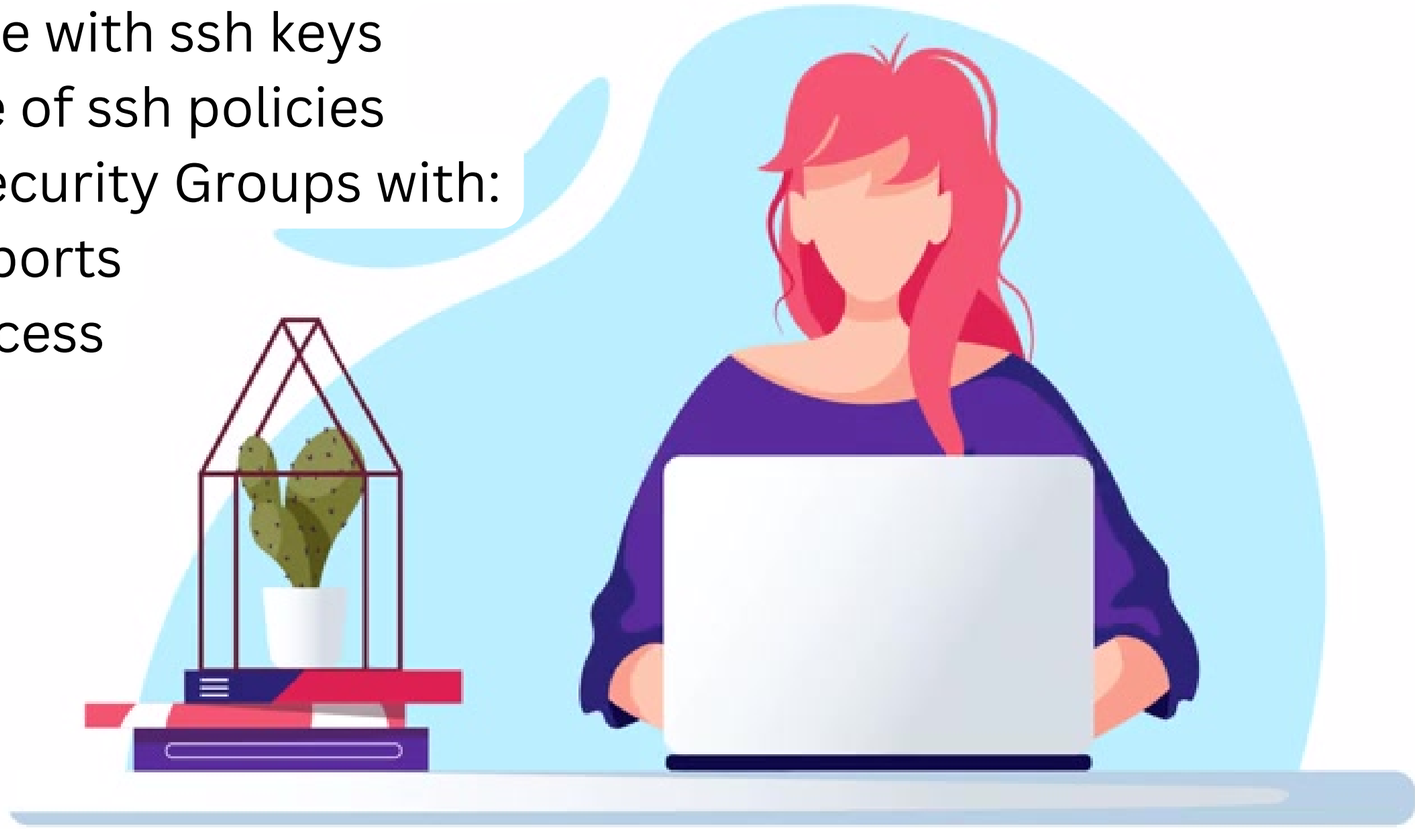
- Public S3 buckets are **public**
- Data in the bucket could be sensitive
- Bucket is not configured properly



Example of “software FOD”

Scenario 3: Exposed Sensitive Ports

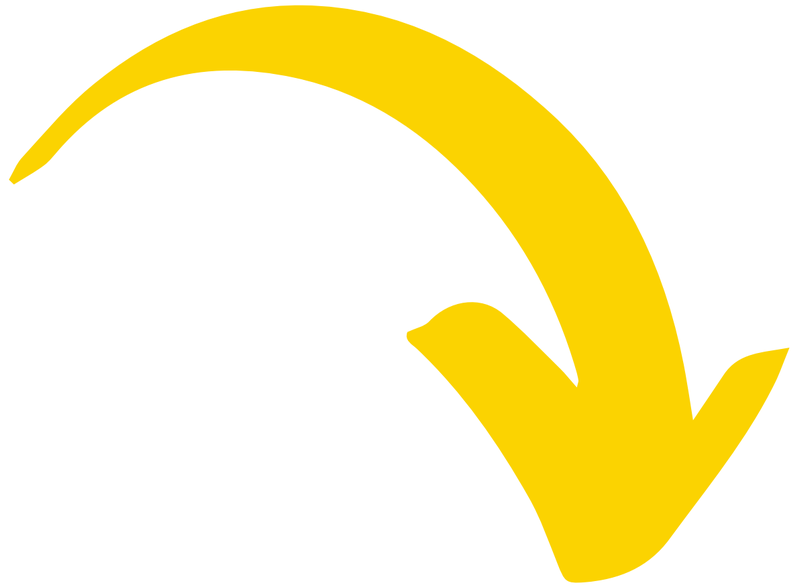
- Eve is a security engineer on an IT team
- Eve noticed an issue with ssh keys
- Teams aren’t aware of ssh policies
- Even found AWS Security Groups with:
  - open sensitive ports
  - unrestricted access



Why **is this** a problem?

WARNING

- Critical vulnerabilities
- Bad practice
- Violates NIST compliance



Example of
“software FOD”

What happens

when we

SCALE?

How **do we** solve this?

		Scenario 1 Rogue Resource	Scenario 2 Public Resource	Scenario 3 Exposed Ports
Administrative Controls				
Technical Controls	**Preventative**			
	Detective			
	Corrective			

1. Administrative Controls

(policy)

lack of policy & clear requirements causes tension

effective policy has buy-in

effective policy is communicated

		Scenario 1 Rogue Resource	Scenario 2 Public Resource	Scenario 3 Exposed Ports
Administrative Controls		Policy to use only approved vendor images	Policy to disallow public resources	Policy to remove port 22 and other sensitive ports
Technical Controls	**Preventative**			
	Detective			
	Corrective			

2. Technical Controls

A. IAM & Policies

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyRunInstanceWithNoTeamTag",
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ec2:*:*:volume/*"
      ],
      "Condition": {
        "Null": {
          "aws:RequestTag/Team": "true"
        }
      }
    }
  ]
}
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMCreateServiceRoleWithPermissionsBoundary",
      "Effect": "Allow",
      "Action": [
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPreventPublicBucket",
      "Effect": "Deny",
      "Action": [
        "s3:PutBucketPublicAccessBlock",
        "s3:PutBucketAcl",
        "s3:PutBucketPolicy"
      ],
      "Resource": "*"
    }
  ]
}
```

```
rvice-*",
```

```
-gov:iam::*:policy/core-permissions-boundary"
```

Hard “deny” blocks aren’t ideal for cases where frequent change is expected to occur

What are **other** solutions?

Amazon S3

Buckets
Access Points
Object Lambda Access Points
Batch Operations
IAM Access Analyzer for S3

Block Public Access settings for this account

Amazon S3 > Block Public Access settings for this account

Block Public Access settings for this account Info

Use Amazon S3 Block public access settings to control the settings that allow public access to your data.

Block Public Access settings for this account

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply account-wide for all current and future buckets and access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects, you can customize the individual settings below to suit your specific storage use cases. Learn more

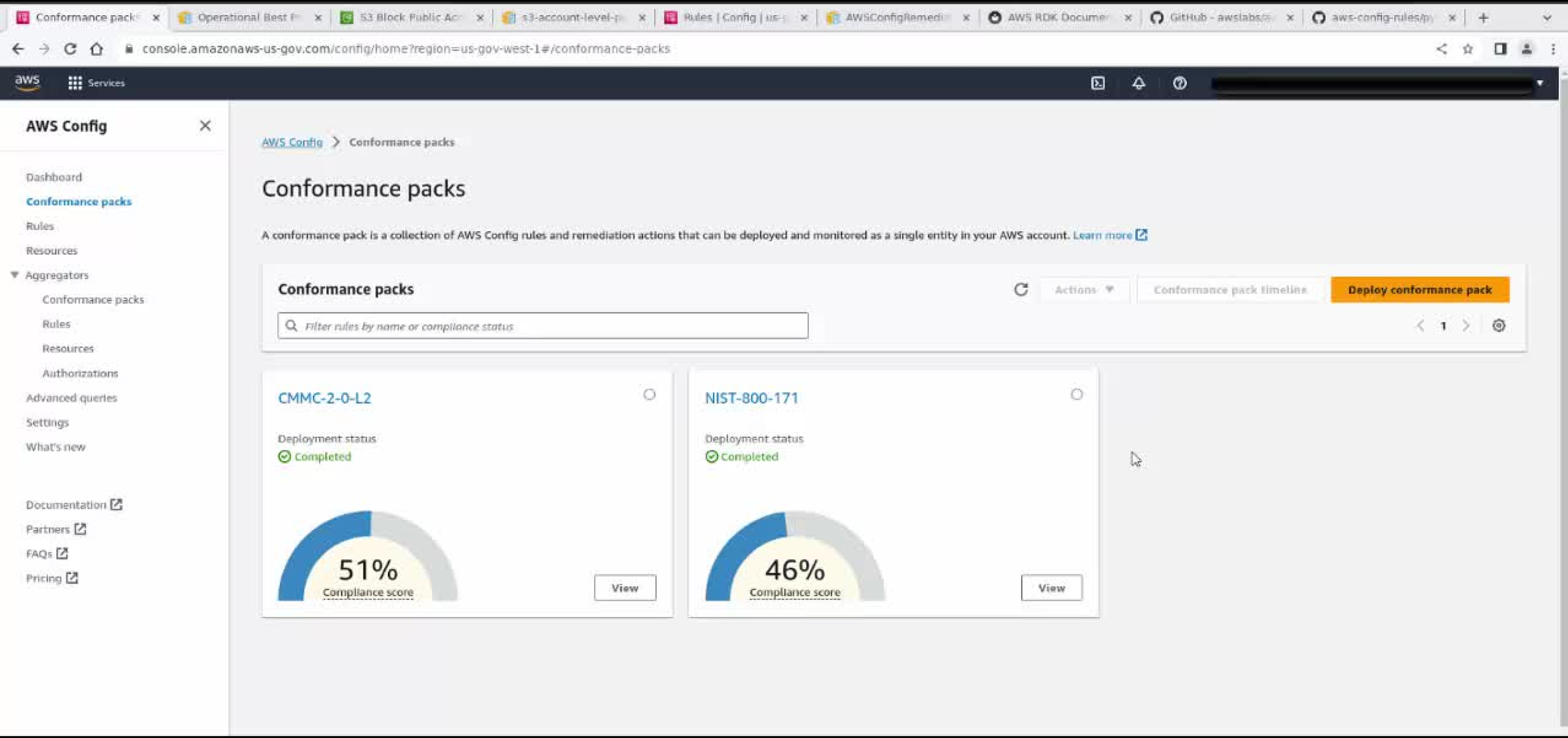
Edit

Block *all* public access
Off

- **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
  Off
- **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
  Off
- **Block public access to buckets and objects granted through *new* public bucket or access point policies**
  Off
- **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
  Off

2. Technical Controls

B. AWS Config


Conformance pack
Operational Best P
S3 Block Public Ac
s3-account-level-p
Rules | Config | us-
AWSConfigRemedi
AWS RDK Docume
GitHub - awslabs/
aws-config-rules/p
console.amazonaws-us-gov.com/config/home?region=us-gov-west-1#/conformance-packs
aws
Services
AWS Config
Dashboard
Conformance packs
Rules
Resources
Aggregators
Conformance packs
Rules
Resources
Authorizations
Advanced queries
Settings
What's new
Documentation
Partners
FAQs
Pricing
AWS Config > Conformance packs
Conformance packs
A conformance pack is a collection of AWS Config rules and remediation actions that can be deployed and monitored as a single entity in your AWS account. Learn more
Conformance packs
Actions
Conformance pack timeline
Deploy conformance pack
Filter rules by name or compliance status
1
CMMC-2-0-L2
Deployment status
Completed
51%
Compliance score
View
NIST-800-171
Deployment status
Completed
46%
Compliance score
View

Execution detail - Step 1: executeAwsApi

Automation step1: executeAwsApi

Status
Failed

Step execution ID
e4210c53-df01-4d64-a80b-0f53f338420d

Input parameters

Failure details

Failure message
Internal Server Err

FailureType
Internal

Description
Checks if the r
account level.
not match one

Config rule AR
arn:aws-us-gov:config:us-gov-west-1:1
service-rule/config-conforms.amazonaw

Warnin

Prior to rule de
than what can

1. Change di
2. Create nev

```
mkdir
```

3. Add curre

```
pip3 install
```

4. Deploy as

```
rdk d
```

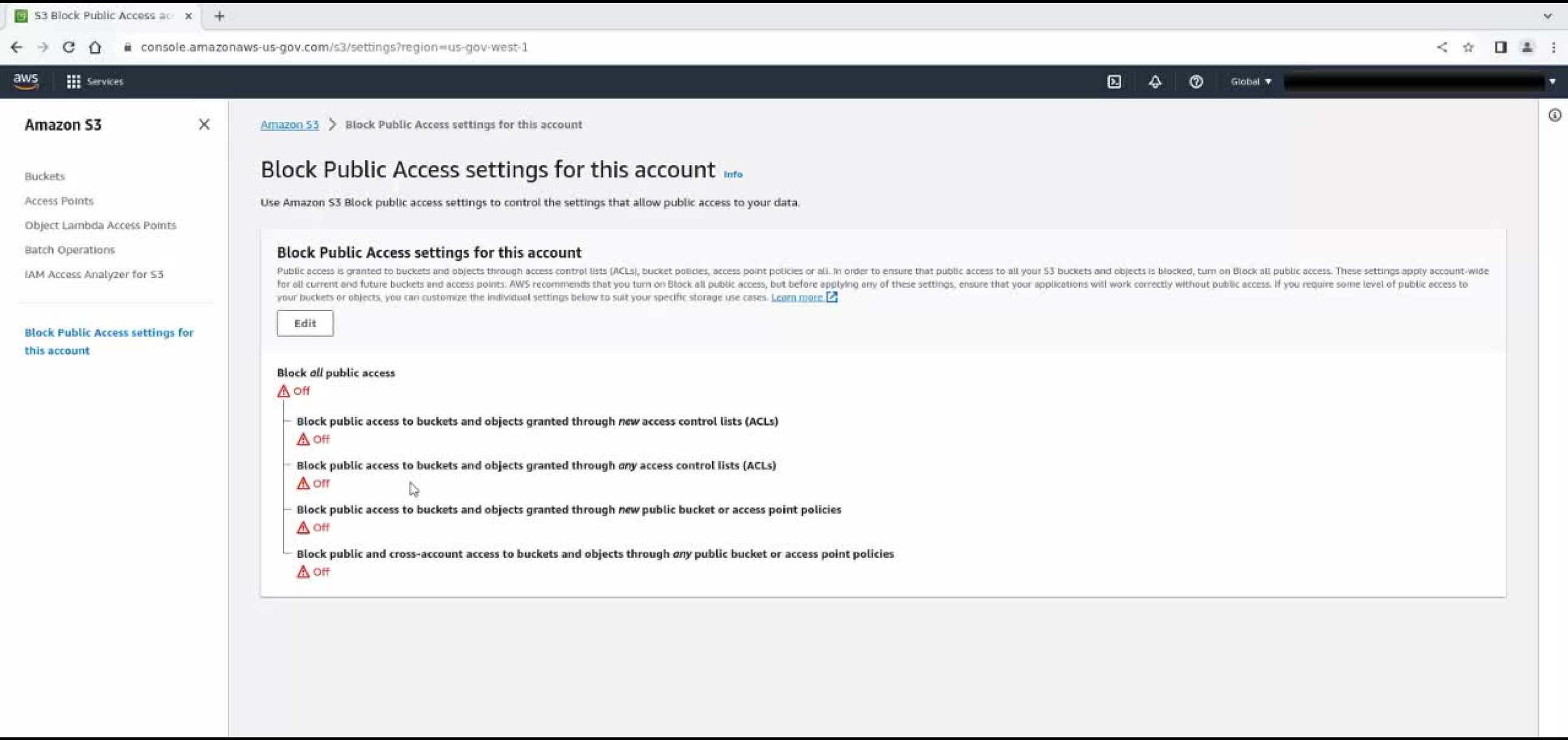
aws-config-rules / python / S3_PUBLIC_ACCESS_SETTINGS_FOR_ACCOUNT / S3_PUBLIC_ACCESS_SETTINGS_FOR_ACCOUNT.PY ↑ Top

Code | Blame | 444 lines (387 loc) · 20.9 KB | Raw

```python
240     def get_configuration(resource_type, resource_id, configuration_capture_time):
408                 evaluations.append(compliance_result)
409         else:
410             evaluations.append(build_evaluation_from_config_item(configuration_item, 'NOT_APPLICABLE'))
411
412         # Put together the request that reports the evaluation status
413         result_token = event['resultToken']
414         test_mode = False
415         if result_token == 'TESTMODE':
416             # Used solely for RDK test to skip actual put_evaluation API call
417             test_mode = True
418
419         # Invoke the Config API to report the result of the evaluation
420         evaluation_copy = []
421         evaluation_copy = evaluations[:]
422         while evaluation_copy:
423             AWS_CONFIG_CLIENT.put_evaluations(Evaluations=evaluation_copy[:100], ResultToken=result_token, TestMode=test_mode)
424             del evaluation_copy[:100]
425
426         # Used solely for RDK test to be able to test Lambda function
427         return evaluations
428
429     def is_internal_error(exception):
430         return ((not isinstance(exception, botocore.exceptions.ClientError)) or exception.response['Error']['Code'].startswith('5')
431                 or 'InternalError' in exception.response['Error']['Code'] or 'ServiceError' in exception.response['Error']['Code'])
432
433     def build_internal_error_response(internal_error_message, internal_error_details=None):
434         return build_error_response(internal_error_message, internal_error_details, 'InternalError', 'InternalError')
435
436     def build_error_response(internal_error_message, internal_error_details=None, customer_error_code=None, customer_error_message=None):
437         error_response = {
438             'internalErrorMessage': internal_error_message,
439             'internalErrorDetails': internal_error_details,
440             'customerErrorMessage': customer_error_message,
441             'customerErrorCode': customer_error_code
442         }
443         print(error_response)
444         return error_response
```

2. Technical Controls

B. Cloud Custodian

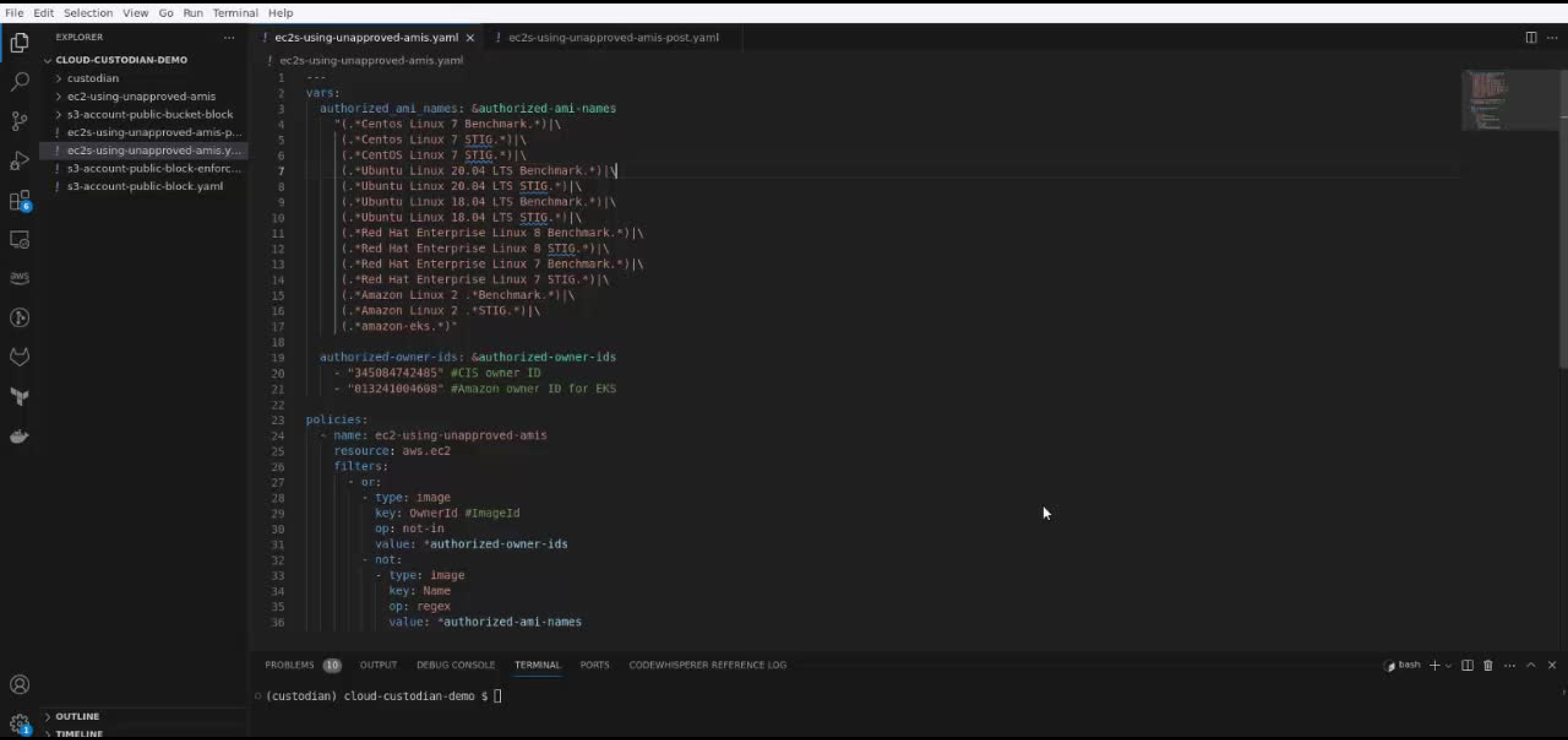

S3 Block Public Access ac
console.amazonaws-us-gov.com/s3/settings?region=us-gov-west-1
aws
Services
Global
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Amazon S3 > Block Public Access settings for this account
Block Public Access settings for this account Info
Use Amazon S3 Block public access settings to control the settings that allow public access to your data.
Block Public Access settings for this account
Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply account-wide for all current and future buckets and access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects, you can customize the individual settings below to suit your specific storage use cases. Learn more
Edit
Block all public access
Off
Block public access to buckets and objects granted through new access control lists (ACLs)
Off
Block public access to buckets and objects granted through any access control lists (ACLs)
Off
Block public access to buckets and objects granted through new public bucket or access point policies
Off
Block public and cross-account access to buckets and objects through any public bucket or access point policies
Off

		Scenario 1 Rogue Resource	Scenario 2 Public Resource	Scenario 3 Exposed Ports
Administrative Controls		Policy to use only approved vendor images	Policy to disallow public resources	Policy to remove port 22 and other sensitive ports
Technical Controls	**Preventative**		IAM, SCP AWS Config Cloud Custodian	
	Detective			
	Corrective			

What if

we want

to only monitor?


File Edit Selection View Go Run Terminal Help
EXPLORER
CLOUD-CUSTODIAN-DEMO
custodian
ec2-using-unapproved-amis
s3-account-public-bucket-block
ec2s-using-unapproved-amis-p...
ec2s-using-unapproved-amis.y...
s3-account-public-block-enforc...
s3-account-public-block.yaml
ec2s-using-unapproved-amis.yaml
ec2s-using-unapproved-amis-post.yaml
ec2s-using-unapproved-amis.yaml
---
vars:
authorized_ami_names: &authorized-ami-names
"(.*Centos Linux 7 Benchmark.*)|\
(.*Centos Linux 7 STIG.*)|\
(.*CentOS Linux 7 STIG.*)|\
(.*Ubuntu Linux 20.04 LTS Benchmark.*)|\
(.*Ubuntu Linux 20.04 LTS STIG.*)|\
(.*Ubuntu Linux 18.04 LTS Benchmark.*)|\
(.*Ubuntu Linux 18.04 LTS STIG.*)|\
(.*Red Hat Enterprise Linux 8 Benchmark.*)|\
(.*Red Hat Enterprise Linux 8 STIG.*)|\
(.*Red Hat Enterprise Linux 7 Benchmark.*)|\
(.*Red Hat Enterprise Linux 7 STIG.*)|\
(.*Amazon Linux 2 .*Benchmark.*)|\
(.*Amazon Linux 2 .*STIG.*)|\
(.*amazon-eks.*)"
authorized-owner-ids: &authorized-owner-ids
- "345084742485" #CIS owner ID
- "013241004608" #Amazon owner ID for EKS
policies:
- name: ec2-using-unapproved-amis
resource: aws.ec2
filters:
- or:
- type: image
key: OwnerId #ImageId
op: not-in
value: *authorized-owner-ids
- not:
- type: image
key: Name
op: regex
value: *authorized-ami-names
PROBLEMS 10
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
CODEWHISPERER REFERENCE LOG
bash
(custodian) cloud-custodian-demo $
OUTLINE
TIMELINE

Use a SIEM solution to:

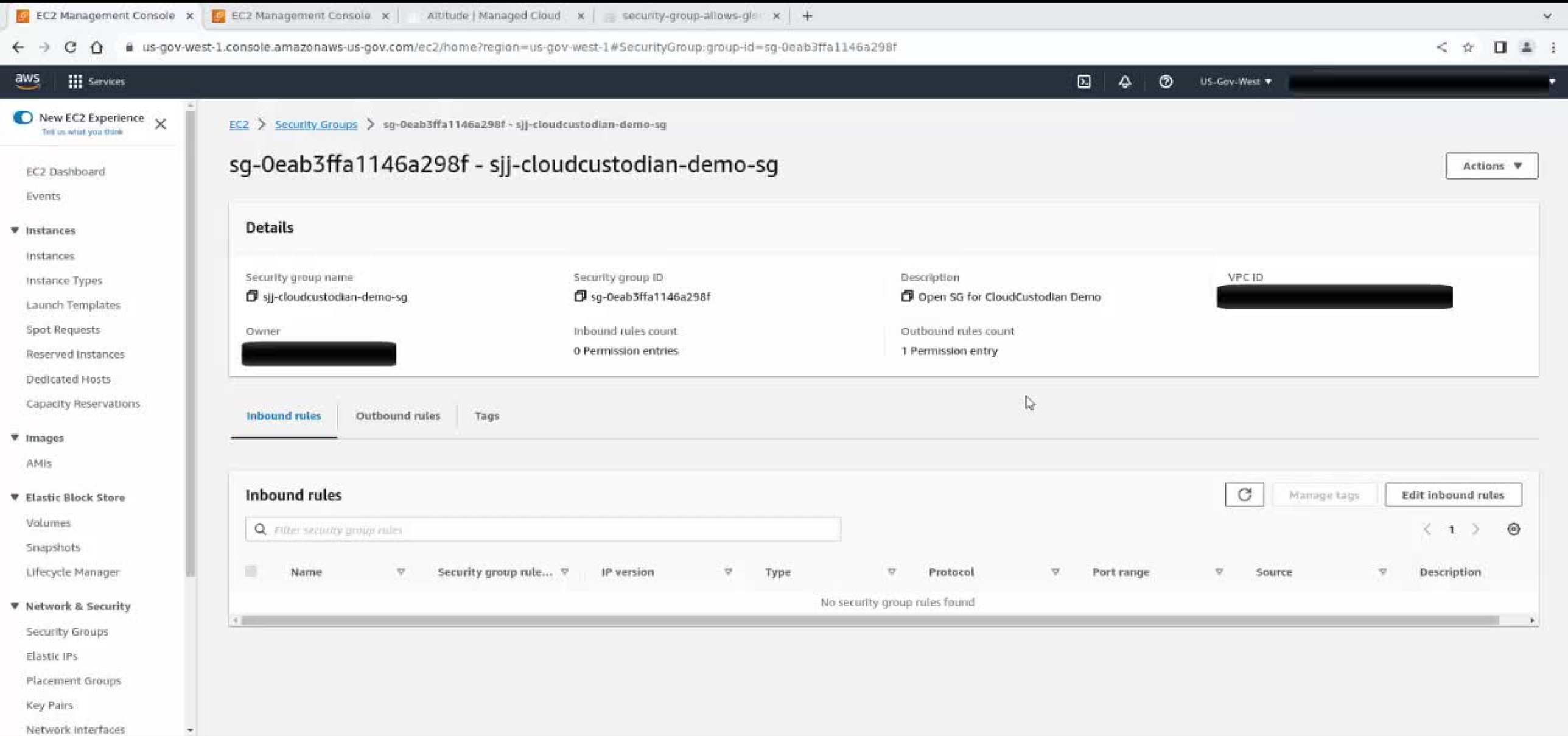
- Report findings
- Trigger alerts
- Build dashboards

		Scenario 1 Rogue Resource	Scenario 2 Public Resource	Scenario 3 Exposed Ports
Administrative Controls		Policy to use only approved vendor images	Policy to disallow public resources	Policy to remove port 22 and other sensitive ports
Technical Controls	Preventative		IAM, SCP AWS Config Cloud Custodian	
	Detective	Cloud Custodian Security Hub SIEM		
	Corrective			

What if

we want

to take action?


EC2 Management Console
EC2 Management Console
Altitude | Managed Cloud
security-group-allows-gl
us-gov-west-1.console.amazonaws-us-gov.com/ec2/home?region=us-gov-west-1#SecurityGroup:group-id=sg-0eab3ffa1146a298f
aws
Services
US-Gov-West
New EC2 Experience
Tell us what you think
EC2 Dashboard
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
EC2 > Security Groups > sg-0eab3ffa1146a298f - sjj-cloudcustodian-demo-sg
sg-0eab3ffa1146a298f - sjj-cloudcustodian-demo-sg
Actions
Details
Security group name
sjj-cloudcustodian-demo-sg
Security group ID
sg-0eab3ffa1146a298f
Description
Open SG for CloudCustodian Demo
VPC ID
[redacted]
Owner
[redacted]
Inbound rules count
0 Permission entries
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Tags
Inbound rules
Manage tags
Edit inbound rules
Filter security group rules
1
Name
Security group rule...
IP version
Type
Protocol
Port range
Source
Description
No security group rules found

		Scenario 1 Rogue Resource	Scenario 2 Public Resource	Scenario 3 Exposed Ports
Administrative Controls		Policy to use only approved vendor images	Policy to disallow public resources	Policy to remove port 22 and other sensitive ports
Technical Controls	Preventative		IAM, SCP AWS Config Cloud Custodian	
	Detective	Cloud Custodian Security Hub SIEM		
	Corrective			Cloud Custodian AWS Config (not demoed)

3 Scenarios of Misconfiguration:

1. Rogue Resources
2. (Unintended) Public Resources
3. Exposed Sensitive Ports

Administrative Controls

- Policy

Technical Controls

- IAM & SCP
- AWS Config
- Cloud Custodian
- Security Hub

Who cleans up our software & engineering environments?

Security is a perspective

on other business problems

Big Ball of Mud

Brian Foote and Joseph Yoder

Department of Computer Science
University of Illinois at Urbana-Champaign
1304 W. Springfield
Urbana, IL 61801 USA

foote@cs.uiuc.edu (217) 328-3523
yoder@cs.uiuc.edu (217) 244-4695

Saturday, June 26, 1999

BIG BALL OF MUD

alias
SHANTYTOWN
SPAGHETTI CODE

THROWAWAY CODE

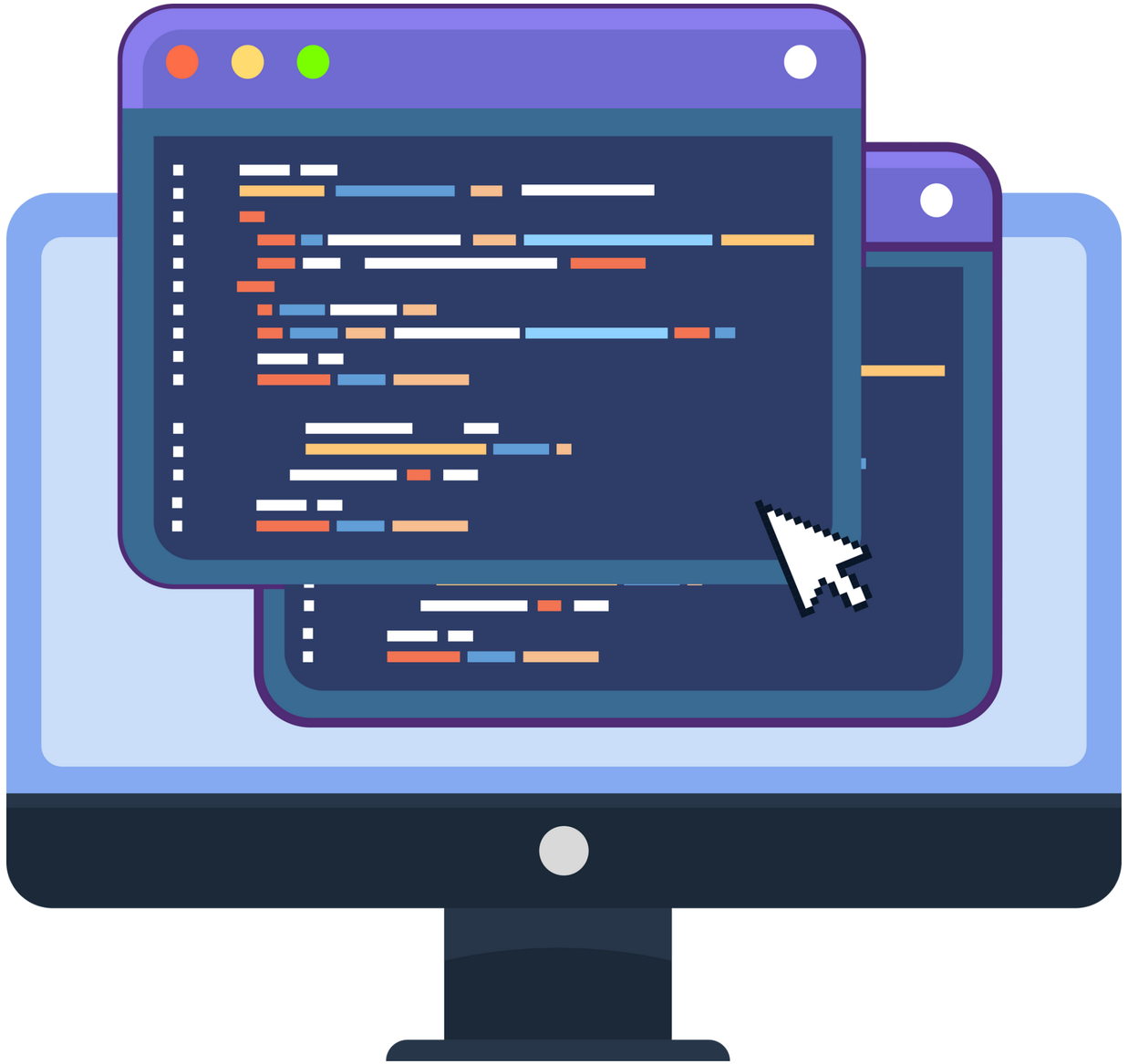
alias
QUICK HACK
KLEENEX CODE
DISPOSABLE CODE
SCRIPTING
KILLER DEMO
PERMANENT PROTOTYPE
BOOMTOWN

Security should:

- leverage automation
- use development tools
- analyze the “business” case & impacts beyond security

Devs should:

- leverage development solutions to solve joint problems
- analyze the security risk perspective



Clean up your cloud!