Vantalect

AWS SECURITY, COMPLIANCE & GOVERNANCE · SAMPLE DELIVERABLE

Detection and Response Everywhere

Week 5 of 6: Detection and response · prepared for StrataForge Robotics, Inc.

Prepared by Charles Sieg · charles.sieg@vantalect.com · vantalect.com

August 14, 2026 · AWS Security, Compliance & Governance Audit: The Hardening (six weeks)

SAMPLE DELIVERABLE

This is a sample deliverable. StrataForge Robotics, Inc. is a fictional company: every account, resource, finding, figure, and person in this report is simulated to show the form and depth of a Vantalect engagement. AWS prices are list prices from the AWS Price List API on October 13, 2026.

Contents

Summary **2**

Coverage, before and after **4**

By Region 5

Delegated administrators 6

Turning each service on **6**

AWS Config and the aggregator 6

GuardDuty 8

AWS Security Hub CSPM 11

IAM Access Analyzer 12

Amazon Inspector 13

Control Tower 17

Routing findings to PagerDuty **17**

The routing rule 18

The organization trail and the log archive **19**

The stacks 20

What the trail records 26

MFA delete, accepted 26

What the detection stack costs **27**

Protection plans and Amazon Inspector 28

What is not in the totals 30

What pages whom: the runbooks **31**

What is not finished **33**

Appendix: coverage by account **34**

Pull request ledger **36**

Summary

In week 5 detection went from partial to everywhere: **GuardDuty, Security Hub CSPM, AWS Config, IAM Access Analyzer, and Amazon Inspector now run in all 816 account-Regions** of the organization (48 accounts times 17 Regions), up from 258 for GuardDuty, 129 for Security Hub

CSPM, 246 for Config, 48 for Access Analyzer, and 8 for Inspector. Findings from all of them route to the people who can act: 10 runbooks name who is paged, for what, and what they do in the first 15 minutes. The organization trail was rebuilt as `sf-org-trail-v2` in a log archive that no application account can alter or delete. Closing the archive account would delete its locked logs, so closing it is denied to every administrator role in the management account and alarmed, and the archive's encryption key is retained and protected.

8 findings closed through 11 pull requests, merged August 10, 2026 through August 14, 2026. 3 more had work merged and are not closed: 2 are partly done, and one low finding was accepted in writing (`SF-SEC-049`, MFA delete), with its reasons below.

The cost is real and we show it. **The detection stack costs about $119,645 a month at list prices, up from about $58,900: an increase of $60,745 a month.** The organization trail's data events cost another $56,764 a month and did not change. What the detection stack costs shows how every line was priced, which usage is measured from the organization and which is assumed, and the one line left out.

Week 5 in numbers. Coverage is the share of the 816 account-Regions where the service is on.

Week 5 at a glance	Result
CloudTrail (organization trail) coverage	100.0% to 100.0%
GuardDuty coverage	31.6% to 100.0%
Security Hub CSPM coverage	15.8% to 100.0%
AWS Config coverage	30.1% to 100.0%
IAM Access Analyzer coverage	5.9% to 100.0%
Amazon Inspector coverage	1.0% to 100.0%
Delegated administrator services	4 to 6
Findings closed	8
Pull requests merged	11
Runbooks	10
Detection stack, monthly, on the bill's basis	$58,900 to $119,645
Trail data events, monthly	$56,764, unchanged

Decision for the sponsor

Marcus Oyelaran: please confirm the detection budget at **$176,409 a month** (the stack plus the trail's data events), or tell Elena Vasquez which service you would trade away. We do not recommend trading any of them: every one closes a finding in the register and feeds a runbook. The lever we do recommend is the trail's data events, which are the largest single line and are a question for the cost review, not a security gap (What the detection stack costs). The plan lines begin when the free trials end (GuardDuty's runs 30 days from when each plan is enabled), so read GuardDuty's and Inspector's usage pages in the last week of the trial and send us the numbers if you want them compared with this estimate in the cost review.

Coverage, before and after

Before this week, detection ran in the six Regions where StrataForge works and almost nowhere else, and not at all in the five Kestrel accounts. An attacker who enabled a Region nobody watches would have been invisible. The `sf-deny-unused-regions` policy from week 3 blocks most of that; detection everywhere is the control for what the policy does not stop.

Coverage by service, out of 816 account-Regions (48 accounts, 17 Regions). The organization trail already covered every account and Region.

Service	Account-Regions before	Coverage before	Account-Regions after	Coverage after
CloudTrail (organization trail)	816	100.0%	816	100.0%
GuardDuty	258	31.6%	816	100.0%
Security Hub CSPM	129	15.8%	816	100.0%
AWS Config	246	30.1%	816	100.0%
IAM Access Analyzer	48	5.9%	816	100.0%
Amazon Inspector	8	1.0%	816	100.0%

By Region

Accounts covered, out of 48, in each Region, before to after. The per-account view is in Appendix: coverage by account.

Region	GuardDuty	Security Hub CSPM	AWS Config	IAM Access Analyzer	Amazon Inspector
us-east-1	43 to 48	43 to 48	41 to 48	48 to 48	8 to 48
us-east-2	0 to 48	0 to 48	0 to 48	0 to 48	0 to 48
us-west-1	0 to 48	0 to 48	0 to 48	0 to 48	0 to 48
us-west-2	43 to 48	43 to 48	41 to 48	0 to 48	0 to 48
ca-central-1	0 to 48	0 to 48	0 to 48	0 to 48	0 to 48
sa-east-1	0 to 48	0 to 48	0 to 48	0 to 48	0 to 48
eu-central-1	43 to 48	43 to 48	41 to 48	0 to 48	0 to 48
eu-west-1	43 to 48	0 to 48	41 to 48	0 to 48	0 to 48
eu-west-2	0 to 48	0 to 48	0 to 48	0 to 48	0 to 48
eu-west-3	0 to 48	0 to 48	0 to 48	0 to 48	0 to 48
eu-north-1	0 to 48	0 to 48	0 to 48	0 to 48	0 to 48
ap-northeast-1	43 to 48	0 to 48	41 to 48	0 to 48	0 to 48
ap-northeast-2	0 to 48	0 to 48	0 to 48	0 to 48	0 to 48
ap-northeast-3	0 to 48	0 to 48	0 to 48	0 to 48	0 to 48
ap-southeast-1	43 to 48	0 to 48	41 to 48	0 to 48	0 to 48
ap-southeast-2	0 to 48	0 to 48	0 to 48	0 to 48	0 to 48
ap-south-1	0 to 48	0 to 48	0 to 48	0 to 48	0 to 48

Delegated administrators

Every detection service is administered from one account, `sf-security-tooling`, so that a compromised workload account cannot turn off what watches it, and so that a new account is covered the day it joins. GuardDuty, Inspector, and Security Hub CSPM name their delegated administrator per Region, so the management account designated `sf-security-tooling` in each of the 17 Regions, and the administrator then activated every existing account in each Region; auto-enable alone reaches only accounts that join later. Before this week 4 services had a delegated administrator; now 6 do.

The delegated administrator for each service, before and after week 5. `sso.amazonaws.com` is IAM Identity Center, administered from `sf-identity`.

Service principal	Delegated administrator, before	Delegated administrator, after
`guardduty.amazonaws.com`	sf-security-tooling	sf-security-tooling
`securityhub.amazonaws.com`	sf-security-tooling	sf-security-tooling
`inspector2.amazonaws.com`	sf-security-tooling	sf-security-tooling
`sso.amazonaws.com`	sf-identity	sf-identity
`config.amazonaws.com`	none	sf-security-tooling
`access-analyzer.amazonaws.com`	none	sf-security-tooling

Turning each service on

The order matters, and it is the order of the pull requests. AWS Config and Control Tower went first because everything else reads their configuration. Then GuardDuty, the trail, Security Hub CSPM, Control Tower enrollment, Access Analyzer, and Inspector. Each went to the sandbox accounts first, then to every account, Region by Region. Where this section shows code, it is the code that merged; commands you can run once are marked as scripts.

AWS Config and the aggregator

SF-SEC-022. AWS Config records nothing in 570 account-Regions. Config now records configuration changes in every governed Region, with one aggregator in `sf-security-tooling` that every account and Region reports to. It is what lets Security Hub CSPM evaluate controls and what lets you answer "what did this resource look like last Tuesday?". `platform-landing-zone#685`, reviewed by Elena Vasquez.

The Control Tower recorder fails the Config.1 service-linked-role test even though it records everything. We set the control's parameter `includeConfigServiceLinkedRoleCheck` to false in the central configuration policy, with the reason in the policy description, instead of suppressing 240 findings one by one. That is a control parameter change, not an exception: the control now checks the thing that is true.

Delegated administrator and the landing zone Region change (management account)

```bash
aws organizations enable-aws-service-access --service-principal config.amazonaws.com
aws organizations register-delegated-administrator --service-principal config.amazonaws.com --
account-id <security-tooling-account-id>

LZ=$(aws controltower list-landing-zones --query 'landingZones[0].arn' --output text)
aws controltower get-landing-zone --landing-zone-identifier "$LZ" --query
'landingZone.manifest' > lz-manifest.json
# Edit lz-manifest.json so governedRegions lists these 17 Regions:
#   us-east-1 us-east-2 us-west-1 us-west-2 ca-central-1 sa-east-1 eu-central-1 eu-west-1 eu-
west-2
#   eu-west-3 eu-north-1 ap-northeast-1 ap-northeast-2 ap-northeast-3 ap-southeast-1 ap-
southeast-2 ap-south-1
aws controltower update-landing-zone --landing-zone-identifier "$LZ" --version <landing-zone-
version> --manifest file://lz-manifest.json
aws controltower get-landing-zone-operation --operation-identifier <operation-identifier> --
query 'operationDetails.status'
```

Organization aggregator in sf-security-tooling

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Description: Aggregate AWS Config data from every account and Region.
Resources:
  AggregatorRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: sf-config-aggregator
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/service-role/AWSConfigRoleForOrganizations
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal: {Service: config.amazonaws.com}
            Action: 'sts:AssumeRole'
  Aggregator:
    Type: AWS::Config::ConfigurationAggregator
    Properties:
      ConfigurationAggregatorName: sf-org-aggregator
      OrganizationAggregationSource:
        RoleArn: !GetAtt AggregatorRole.Arn
        AllAwsRegions: true
```

How we proved it: `describe-configuration-recorder-status` shows a recording recorder in every Region of a sample account, including a Kestrel account; `describe-configuration-aggregator-sources-status` shows every account and Region as succeeded; and `aws controltower get-landing-zone` shows every governed Region and status ACTIVE. Rollback: revert the pull request to stop recording in a Region. We would not, because stopped recording also stops the Security Hub CSPM evaluations that read it.

GuardDuty

SF-SEC-023. GuardDuty is off in 558 account-Regions, including every Kestrel account; most protection plans are off. GuardDuty now runs in all 17 Regions of all 48 accounts, with `sf-security-tooling` designated as the delegated administrator in every Region, auto-enable on for every member, and every protection plan on: S3 Protection, EKS Protection, Runtime Monitoring (with the EC2, EKS, and Fargate agents managed for you), Malware Protection for EC2, RDS Protection, and Lambda Protection. Extended Threat Detection is on automatically at no extra cost and turns related signals into attack-sequence findings, which is where the runbooks for compromised clusters and instance groups come from. `platform-landing-zone#688`, reviewed by Nadia Petrova.

Delegate in every Region, then enable all members and plans (management account, then sf-security-tooling)

```bash
REGIONS="us-east-1 us-east-2 us-west-1 us-west-2 ca-central-1 sa-east-1 eu-central-1 eu-west-1 eu-west-2 eu-west-3 eu-north-1 ap-northeast-1 ap-northeast-2 ap-northeast-3 ap-southeast-1 ap-southeast-2 ap-south-1"

# 1. Management account: trusted access, the list of active accounts, and the delegated administrator in EVERY Region.
#    The designation is per Region; a Region without it rejects update-organization-configuration.
aws organizations enable-aws-service-access --profile <management-profile> --service-principal guardduty.amazonaws.com
aws organizations list-accounts --profile <management-profile> --query 'Accounts[?State==`ACTIVE`].Id' --output text | tr '\t' '\n' > active-accounts.txt
for region in $REGIONS; do
  aws guardduty enable-organization-admin-account --profile <management-profile> --region $region --admin-account-id <security-tooling-account-id>
done

# 2. sf-security-tooling: a detector in each Region, auto-enable for all members, and a count of members that came up.
#    The administrator account is not its own member, so the expected count is one less than the active accounts.
expected=$(( $(wc -l < active-accounts.txt) - 1 ))
for region in $REGIONS; do
  detector=$(aws guardduty list-detectors --profile <security-tooling-profile> --region $region --query 'DetectorIds[0]' --output text)
  [ "$detector" = "None" ] && detector=$(aws guardduty create-detector --profile <security-tooling-profile> --region $region --enable --query DetectorId --output text)
```

```
  aws guardduty update-organization-configuration --profile <security-tooling-profile> --region
$region --detector-id "$detector" --auto-enable-organization-members ALL --features '[
    {"Name": "S3_DATA_EVENTS", "AutoEnable": "ALL"},
    {"Name": "EKS_AUDIT_LOGS", "AutoEnable": "ALL"},
    {"Name": "EBS_MALWARE_PROTECTION", "AutoEnable": "ALL"},
    {"Name": "RDS_LOGIN_EVENTS", "AutoEnable": "ALL"},
    {"Name": "LAMBDA_NETWORK_LOGS", "AutoEnable": "ALL"},
    {"Name": "RUNTIME_MONITORING", "AutoEnable": "ALL", "AdditionalConfiguration": [
      {"Name": "EKS_ADDON_MANAGEMENT", "AutoEnable": "ALL"},
      {"Name": "ECS_FARGATE_AGENT_MANAGEMENT", "AutoEnable": "ALL"},
      {"Name": "EC2_AGENT_MANAGEMENT", "AutoEnable": "ALL"}]}]'
  enabled=$(aws guardduty list-members --profile <security-tooling-profile> --region $region --
detector-id "$detector" --only-associated false \
    --query 'length(Members[?RelationshipStatus==`Enabled`])' --output text)
  [ "$enabled" = "$expected" ] || echo "CHECK $region: $enabled of $expected members are
Enabled (enrollment can take several minutes; run the count again)"
done
```

High and critical GuardDuty findings

```JSON
{
  "source": [
    "aws.guardduty"
  ],
  "detail-type": [
    "GuardDuty Finding"
  ],
  "detail": {
    "severity": [
      {
        "numeric": [
          ">=",
          7
        ]
      }
    ]
  }
}
```

Route GuardDuty findings from every Region to the on-call (sf-security-tooling)

```bash
REGIONS="us-east-1 us-east-2 us-west-1 us-west-2 ca-central-1 sa-east-1 eu-central-1 eu-west-1
eu-west-2 eu-west-3 eu-north-1 ap-northeast-1 ap-northeast-2 ap-northeast-3 ap-southeast-1 ap-
southeast-2 ap-south-1"
BUS=arn:aws:events:us-east-1:<security-tooling-account-id>:event-bus/default
FORWARD_ROLE=arn:aws:iam::<security-tooling-account-id>:role/sf-events-forward

# Once: the role EventBridge uses to put events on the us-east-1 bus.
aws iam create-role --role-name sf-events-forward --assume-role-policy-document \
  '{"Version": "2012-10-17", "Statement": [{"Effect": "Allow", "Principal": {"Service":
"events.amazonaws.com"}, "Action": "sts:AssumeRole"}]}'
aws iam put-role-policy --role-name sf-events-forward --policy-name put-events-to-us-east-1 --
policy-document \
  '{"Version": "2012-10-17", "Statement": [{"Effect": "Allow", "Action": "events:PutEvents",
"Resource": "'$BUS'"}]}'

# Every Region except us-east-1 forwards matching findings; us-east-1 pages directly.
for region in $REGIONS; do
  aws events put-rule --region $region --name sf-guardduty-high-critical --state ENABLED --
event-pattern file://guardduty-high-critical-pattern.json
  if [ "$region" = "us-east-1" ]; then
    aws sns create-topic --region us-east-1 --name sf-guardduty-pagerduty
    aws events put-targets --region us-east-1 --rule sf-guardduty-high-critical \
      --targets Id=PagerDuty,Arn=arn:aws:sns:us-east-1:<security-tooling-account-id>:sf-
guardduty-pagerduty
  else
    aws events put-targets --region $region --rule sf-guardduty-high-critical --targets
Id=ToUsEast1,Arn=$BUS,RoleArn=$FORWARD_ROLE
  fi
done
# The us-east-1 rule matches both its own findings and the forwarded ones, because forwarding
keeps the source and detail.
for region in $REGIONS; do
  echo "$region: $(aws events describe-rule --region $region --name sf-guardduty-high-critical
--query State --output text)"
done
```

How we proved it: `list-detectors` returns a detector in every Region, `list-members` shows every account as Enabled, `describe-organization-configuration` shows `AutoEnableOrganizationMembers` set to ALL with every listed feature on, and a sample finding made with `create-sample-findings` reached the on-call route. Rollback: GuardDuty can be suspended per Region; the `sf-protect-security-services` policy from week 3 forbids it to everyone but the security automation role, which is why this is a decision for Nadia Petrova and the sponsor, not a console click.

AWS Security Hub CSPM

SF-SEC-038. Security Hub CSPM runs in only 3 Regions, with FSBP only and no CIS standard. AWS Security Hub CSPM now runs in every Region of every account, centrally configured from `sf-security-tooling`, which is the administrator in every Region and aggregates findings from all 17 Regions into us-east-1 through a finding aggregator: one configuration policy, `sf-baseline`, enables the AWS Foundational Security Best Practices standard v1.0.0 and the CIS AWS Foundations Benchmark v5.0.0 everywhere. Two parameters differ from the defaults, each with a reason in the policy: EC2.18 allows the TCP ports `80`, `443`, and `3478` and the UDP port `3478` that StrataForge's public services and WebRTC (TURN) relays use, and Config.1 skips the service-linked-role check for the reason above. EC2.172, the control that wants VPC Block Public Access to block internet gateway traffic, is disabled by agreement: the customer portal and the WebRTC relays have to reach the internet through gateways. `platform-landing-zone#695`, reviewed by Nadia Petrova.

Delegated administrator, aggregator, central configuration, and the sf-baseline policy

```bash
REGIONS="us-east-1 us-east-2 us-west-1 us-west-2 ca-central-1 sa-east-1 eu-central-1 eu-west-1
eu-west-2 eu-west-3 eu-north-1 ap-northeast-1 ap-northeast-2 ap-northeast-3 ap-southeast-1 ap-
southeast-2 ap-south-1"

# 1. Management account: make sf-security-tooling the administrator from the home Region.
aws securityhub enable-organization-admin-account --profile <management-profile> --region us-
east-1 --admin-account-id <security-tooling-account-id>

# 2. sf-security-tooling: turn Security Hub CSPM on without the default standards, aggregate
every Region into us-east-1,
#    and switch the organization to central configuration (it needs auto-enable off).
aws securityhub enable-security-hub --profile <security-tooling-profile> --region us-east-1 --
no-enable-default-standards
aws securityhub create-finding-aggregator --profile <security-tooling-profile> --region us-
east-1 --region-linking-mode ALL_REGIONS
aws securityhub update-organization-configuration --profile <security-tooling-profile> --region
us-east-1 --no-auto-enable --auto-enable-standards NONE \
  --organization-configuration ConfigurationType=CENTRAL

# 3. The configuration policy. Integer lists are plain numbers; EC2.18 takes TCP and UDP port
lists, and TURN relays use 3478 on both.
cat > sf-baseline-policy.json <<'JSON'
{"SecurityHub": {
  "ServiceEnabled": true,
  "EnabledStandardIdentifiers": [
    "arn:aws:securityhub:us-east-1::standards/aws-foundational-security-best-practices/v/
1.0.0",
    "arn:aws:securityhub:us-east-1::standards/cis-aws-foundations-benchmark/v/5.0.0"],
  "SecurityControlsConfiguration": {
    "DisabledSecurityControlIdentifiers": ["EC2.172"],
    "SecurityControlCustomParameters": [
```

```
      {"SecurityControlId": "EC2.18", "Parameters": {
        "authorizedTcpPorts": {"ValueType": "CUSTOM", "Value": {"IntegerList": [80, 443,
3478]}},
        "authorizedUdpPorts": {"ValueType": "CUSTOM", "Value": {"IntegerList": [3478]}}}},
      {"SecurityControlId": "Config.1", "Parameters": {"includeConfigServiceLinkedRoleCheck":
{"ValueType": "CUSTOM", "Value": {"Boolean": false}}}}]}}}
JSON
POLICY=$(aws securityhub create-configuration-policy --profile <security-tooling-profile> --
region us-east-1 --name sf-baseline \
  --configuration-policy file://sf-baseline-policy.json --query Id --output text)
aws securityhub start-configuration-policy-association --profile <security-tooling-profile> --
region us-east-1 \
  --configuration-policy-identifier "$POLICY" --target RootId=<org-root-id>

# 4. Every linked Region must list sf-security-tooling as the administrator. Where one does
not, run
#    enable-organization-admin-account in that Region too before you trust the coverage
numbers.
for region in $REGIONS; do
  echo "$region: $(aws securityhub list-organization-admin-accounts --profile <management-
profile> --region $region --query 'AdminAccounts[].[AccountId,Status]' --output text)"
done
```

How we proved it: `list-finding-aggregators` shows a finding aggregator linked to all Regions, `describe-organization-configuration` shows `ConfigurationType` CENTRAL, `list-configuration-policy-associations` shows the root associated with `sf-baseline`, both standards show as enabled in every account and Region, and a test CRITICAL finding created with `batch-import-findings` paged the on-call engineer. Rollback: a central configuration policy is edited and re-associated; it does not need to be removed to change a control.

> **Two products with nearly the same name**
>
> AWS Security Hub CSPM is the original Security Hub, renamed: it owns the controls (IAM.4, EC2.8), the FSBP and CIS standards, and the findings in ASFF format. **AWS Security Hub**, generally available since December 2, 2025, is a newer service that correlates GuardDuty, Inspector, Security Hub CSPM, and Macie findings into prioritized exposures, in OCSF format. We turned on the first. We did not turn on the second, because StrataForge's on-call is not ready to triage a second queue; we recommend evaluating it after the first quarter of Security Hub CSPM findings are at a steady state. Automation rules written for Security Hub CSPM must be migrated if you adopt it.

IAM Access Analyzer

SF-SEC-044. IAM Access Analyzer covers only us-east-1; 768 account-Regions have no external access analyzer. An organization-wide external-access analyzer, `sf-org-external-access`, now runs in every Region, administered from `sf-security-tooling`. It finds what can be reached from

outside the organization: a public bucket, a snapshot shared with another account, a role anyone can assume. Active findings route to the security queue, and each must have an owner or an archive rule with a written reason. `platform-landing-zone#707`, reviewed by Nadia Petrova.

The same analyzer as a stack set template

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Description: Organization external access analyzer.
Resources:
  ExternalAccessAnalyzer:
    Type: AWS::AccessAnalyzer::Analyzer
    Properties:
      AnalyzerName: sf-org-external-access
      Type: ORGANIZATION
```

How we proved it: `aws accessanalyzer list-analyzers --type ORGANIZATION` returns the analyzer in every Region, and a test bucket policy that granted an outside account access produced a finding within minutes. Rollback: delete the stack set; findings are lost with it, so archive rather than delete.

Amazon Inspector

SF-SEC-042. Amazon Inspector is off almost everywhere; 132 ECR repositories have no image scanning. Amazon Inspector now scans EC2 instances, container images in ECR, Lambda functions, and Lambda function code across the organization, with `sf-security-tooling` as the delegated administrator in every Region, every existing account activated in every Region, and every scan type auto-enabled for accounts that join later. ECR moved to continuous enhanced scanning, so a new CVE in an image pushed last month is found without a re-push. Critical findings route to PagerDuty at low urgency from a rule in each Region: they are patching work, not incidents. `platform-landing-zone#713`, reviewed by Nadia Petrova.

Delegate in every Region and activate all accounts (management account, then sf-security-tooling)

```bash
REGIONS="us-east-1 us-east-2 us-west-1 us-west-2 ca-central-1 sa-east-1 eu-central-1 eu-west-1
eu-west-2 eu-west-3 eu-north-1 ap-northeast-1 ap-northeast-2 ap-northeast-3 ap-southeast-1 ap-
southeast-2 ap-south-1"

# 1. Management account: trusted access, the active accounts, and the delegated administrator
in EVERY Region.
#    Inspector's delegated administrator is set per Region.
aws organizations enable-aws-service-access --profile <management-profile> --service-principal
inspector2.amazonaws.com
aws organizations list-accounts --profile <management-profile> --query 'Accounts[?
State==`ACTIVE`].Id' --output text > active-accounts.txt
for region in $REGIONS; do
  aws inspector2 enable-delegated-admin-account --profile <management-profile> --region $region
--delegated-admin-account-id <security-tooling-account-id>
done

# 2. sf-security-tooling: activate every existing account in every Region (auto-enable reaches
only accounts that join later),
#    turn on auto-enable for new members, and list any account that is not fully enabled.
for region in $REGIONS; do
  aws inspector2 enable --profile <security-tooling-profile> --region $region --account-ids $
(cat active-accounts.txt) \
    --resource-types EC2 ECR LAMBDA LAMBDA_CODE
  aws inspector2 update-organization-configuration --profile <security-tooling-profile> --
region $region \
    --auto-enable ec2=true,ecr=true,lambda=true,lambdaCode=true
  not_enabled=$(aws inspector2 batch-get-account-status --profile <security-tooling-profile> --
region $region --account-ids $(cat active-accounts.txt) \
    --query 'accounts[?state.status!=`ENABLED`].accountId' --output text)
  [ -z "$not_enabled" ] || echo "CHECK $region: not fully enabled yet: $not_enabled"
done
```

Critical Inspector findings

```JSON
{
  "source": [
    "aws.inspector2"
  ],
  "detail-type": [
    "Inspector2 Finding"
  ],
  "detail": {
    "severity": [
      "CRITICAL"
    ],
    "status": [
      "ACTIVE"
    ]
  }
}
```

Route Inspector findings from every Region to the on-call (sf-security-tooling)

```bash
REGIONS="us-east-1 us-east-2 us-west-1 us-west-2 ca-central-1 sa-east-1 eu-central-1 eu-west-1
eu-west-2 eu-west-3 eu-north-1 ap-northeast-1 ap-northeast-2 ap-northeast-3 ap-southeast-1 ap-
southeast-2 ap-south-1"
BUS=arn:aws:events:us-east-1:<security-tooling-account-id>:event-bus/default
FORWARD_ROLE=arn:aws:iam::<security-tooling-account-id>:role/sf-events-forward

# Once: the role EventBridge uses to put events on the us-east-1 bus.
aws iam create-role --role-name sf-events-forward --assume-role-policy-document \
  '{"Version": "2012-10-17", "Statement": [{"Effect": "Allow", "Principal": {"Service":
"events.amazonaws.com"}, "Action": "sts:AssumeRole"}]}'
aws iam put-role-policy --role-name sf-events-forward --policy-name put-events-to-us-east-1 --
policy-document \
  '{"Version": "2012-10-17", "Statement": [{"Effect": "Allow", "Action": "events:PutEvents",
"Resource": "'$BUS'"}]}'

# Every Region except us-east-1 forwards matching findings; us-east-1 pages directly.
for region in $REGIONS; do
  aws events put-rule --region $region --name sf-inspector-critical --state ENABLED --event-
pattern file://inspector-critical-pattern.json
  if [ "$region" = "us-east-1" ]; then
    aws sns create-topic --region us-east-1 --name sf-inspector-pagerduty
    aws events put-targets --region us-east-1 --rule sf-inspector-critical \
      --targets Id=PagerDuty,Arn=arn:aws:sns:us-east-1:<security-tooling-account-id>:sf-
inspector-pagerduty
  else
    aws events put-targets --region $region --rule sf-inspector-critical --targets
Id=ToUsEast1,Arn=$BUS,RoleArn=$FORWARD_ROLE
  fi
done
# The us-east-1 rule matches both its own findings and the forwarded ones, because forwarding
keeps the source and detail.
for region in $REGIONS; do
  echo "$region: $(aws events describe-rule --region $region --name sf-inspector-critical --
query State --output text)"
done
```

How we proved it: `batch-get-account-status` shows EC2, ECR, Lambda, and Lambda code scanning enabled for every member, `get-registry-scanning-configuration` shows ENHANCED with CONTINUOUS_SCAN in each account, and a known vulnerable test image pushed to ECR produced a finding within minutes. Instances that Systems Manager does not manage (556 of them) can only be scanned agentlessly, which finds fewer problems than agent-based scanning; getting them under Systems Manager is `SF-SEC-033`, in the roadmap.

Control Tower

SF-SEC-039. 7 accounts are outside Control Tower, and the landing zone has drifted. The landing zone had drifted: its guardrail SCPs were detached from all registered OUs in March 2023 to unblock a failed deployment and were never reattached, and some accounts sat outside Control Tower. Control Tower now governs all 17 Regions, every account is enrolled, and the drift is repaired. `platform-landing-zone#701`, reviewed by Elena Vasquez. The Kestrel accounts, `sandbox-hackathon`, and `suspended-legacy-ci` are the accounts that were outside it.

AWSControlTowerExecution role for an unenrolled account

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Description: Role Control Tower assumes to enroll this account.
Parameters:
  ManagementAccountId: {Type: String, Description: Management account ID (<management-account-
id>).}
Resources:
  ControlTowerExecutionRole:
    Type: AWS::IAM::Role
    Properties:
      RoleName: AWSControlTowerExecution
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/AdministratorAccess
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal: {AWS: !Sub 'arn:aws:iam::${ManagementAccountId}:root'}
            Action: 'sts:AssumeRole'
```

How we proved it: `aws controltower get-landing-zone` shows ACTIVE, `IN_SYNC`, and the latest version; `list-enabled-baselines` shows every account and OU as SUCCEEDED; none of the previously unenrolled accounts remain outside a registered OU.

Routing findings to PagerDuty

A finding that nobody reads is a log line. Findings reach people through one path: services emit to EventBridge in each Region, a rule filters by severity and type, and an SNS topic in us-east-1 feeds PagerDuty (GuardDuty and Inspector rules in the other Regions forward to the us-east-1 bus). Which PagerDuty service is paged, and at what urgency, is set by the runbook that matches the finding, so the page tells the engineer which procedure to open. Before this week a Slack channel received some GuardDuty findings, and nothing else was routed.

How each source reaches PagerDuty, and at what urgency. The runbook list in [What pages whom: the runbooks](#) names the owner of each.

Source	Pattern	Goes to
GuardDuty	Findings of severity HIGH or CRITICAL, including attack sequences	Security, high urgency; Fleet Cloud for EKS
Security Hub CSPM	New, failed CRITICAL and HIGH findings that Security Hub CSPM itself generated (GuardDuty and Inspector findings that it imports are routed by their own rules, so nothing pages twice)	Platform, low urgency, business hours
Amazon Inspector	New CRITICAL findings	Platform, low urgency
IAM Access Analyzer	New active findings	Security queue; high urgency for production accounts
Device Defender	Detect violations on the robot thing groups	Fleet Cloud, high urgency
EventBridge on CloudTrail	Root and break-glass sign-in, `sts:AssumeRoot`, `kms:Sign` by anyone but the OTA signer, a trail change	Security or Fleet OTA, high urgency

The routing rule

New, failed CRITICAL and HIGH findings that Security Hub CSPM itself generated

```JSON
{
  "source": [
    "aws.securityhub"
  ],
  "detail-type": [
    "Security Hub Findings - Imported"
  ],
  "detail": {
    "findings": {
      "ProductName": [
        "Security Hub"
      ],
      "Severity": {
        "Label": [
          "CRITICAL",
          "HIGH"
        ]
      },
      "Compliance": {
        "Status": [
```

```
          "FAILED"
        ]
      },
      "Workflow": {
        "Status": [
          "NEW"
        ]
      },
      "RecordState": [
        "ACTIVE"
      ]
    }
  }
}
```

Send matching findings to the PagerDuty topic

```bash
aws events put-rule --profile <security-tooling-profile> --region us-east-1 --name sf-securityhub-critical-high --state ENABLED \
  --event-pattern file://securityhub-critical-high-pattern.json
aws events put-targets --profile <security-tooling-profile> --region us-east-1 --rule sf-securityhub-critical-high \
  --targets Id=PagerDuty,Arn=arn:aws:sns:us-east-1:<security-tooling-account-id>:sf-securityhub-pagerduty
```

Test the route, don't trust it: each routing rule was proved by generating a sample or test finding (`create-sample-findings`, `batch-import-findings`, a test `StopLogging` on a scratch trail) and measuring the time to the page. Repeat that test every quarter, with the break-glass sign-in test from week 4, and log the times.

The organization trail and the log archive

SF-SEC-025. The 2019 organization trail can be stopped from the management account, and its logs can be deleted. This is Deliverable 12. The 2019 organization trail, `sf-org-trail`, had no log file validation and no KMS key, and its bucket had no Object Lock and could be written and deleted from the management account, which means anyone who took the management account could erase the evidence of doing so. We built a new bucket in `sf-log-archive`, locked in COMPLIANCE mode for 400 days, and a new organization trail, `sf-org-trail-v2`, with log file validation, a KMS key that the stack retains if it is ever deleted, and multi-Region logging. `platform-landing-zone#690`, reviewed by Nadia Petrova.

COMPLIANCE mode matters. In governance mode a sufficiently privileged principal can shorten or remove the lock; in compliance mode nobody can, the root user included, until the retention ends. That is the property an auditor wants. The cost is that a mistake is permanent for 400 days: object versions you store in the bucket cannot be deleted early, so the bucket holds the trail and the logs sent to the archive, and nothing else.

The stacks

sf-log-archive: key, access-log bucket, and the Object Lock bucket

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Description: Immutable CloudTrail archive for the organization trail.
Parameters:
  OrganizationId: {Type: String, Description: AWS Organizations ID (<org-id>).}
  ManagementAccountId: {Type: String, Description: Management account ID (<management-account-
id>).}
Resources:
  TrailKey:
    Type: AWS::KMS::Key
    DeletionPolicy: Retain
    UpdateReplacePolicy: Retain
    Properties:
      EnableKeyRotation: true
      PendingWindowInDays: 30
      KeyPolicy:
        Version: '2012-10-17'
        Statement:
          - Sid: AccountAdministration
            Effect: Allow
            Principal: {AWS: !Sub 'arn:aws:iam::${AWS::AccountId}:root'}
            Action: 'kms:*'
            Resource: '*'
          - Sid: CloudTrailDescribesKey
            Effect: Allow
            Principal: {Service: cloudtrail.amazonaws.com}
            Action: 'kms:DescribeKey'
            Resource: '*'
            Condition:
              StringEquals: {'aws:SourceArn': !Sub 'arn:aws:cloudtrail:us-east-1:$
{ManagementAccountId}:trail/sf-org-trail-v2'}
          - Sid: CloudTrailEncrypts
            Effect: Allow
            Principal: {Service: cloudtrail.amazonaws.com}
            Action: 'kms:GenerateDataKey*'
            Resource: '*'
            Condition:
              StringEquals: {'aws:SourceArn': !Sub 'arn:aws:cloudtrail:us-east-1:$
{ManagementAccountId}:trail/sf-org-trail-v2'}
              StringLike: {'kms:EncryptionContext:aws:cloudtrail:arn': !Sub
'arn:aws:cloudtrail:*:${ManagementAccountId}:trail/sf-org-trail-v2'}
          - Sid: OrganizationReadersDescribeKey
            Effect: Allow
```

```
            Principal: {AWS: '*'}
            Action: 'kms:DescribeKey'
            Resource: '*'
            Condition:
              StringEquals: {'aws:PrincipalOrgID': !Ref OrganizationId}
          - Sid: OrganizationReadersDecrypt
            Effect: Allow
            Principal: {AWS: '*'}
            Action: 'kms:Decrypt'
            Resource: '*'
            Condition:
              StringEquals: {'aws:PrincipalOrgID': !Ref OrganizationId}
              StringLike: {'kms:EncryptionContext:aws:cloudtrail:arn': !Sub
'arn:aws:cloudtrail:*:${ManagementAccountId}:trail/sf-org-trail-v2'}
  AccessLogBucket:
    Type: AWS::S3::Bucket
    Properties:
      BucketName: !Sub 'sf-log-archive-access-logs-${AWS::AccountId}-${AWS::Region}'
      BucketEncryption:
        ServerSideEncryptionConfiguration:
          - ServerSideEncryptionByDefault: {SSEAlgorithm: AES256}
      PublicAccessBlockConfiguration: {BlockPublicAcls: true, BlockPublicPolicy: true,
IgnorePublicAcls: true, RestrictPublicBuckets: true}
  AccessLogBucketPolicy:
    Type: AWS::S3::BucketPolicy
    Properties:
      Bucket: !Ref AccessLogBucket
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Sid: S3ServerAccessLogsDelivery
            Effect: Allow
            Principal: {Service: logging.s3.amazonaws.com}
            Action: 's3:PutObject'
            Resource: !Sub '${AccessLogBucket.Arn}/*'
            Condition: {StringEquals: {'aws:SourceAccount': !Ref 'AWS::AccountId'}}
          - Sid: DenyNonTlsRequests
            Effect: Deny
            Principal: '*'
            Action: 's3:*'
            Resource: [!GetAtt AccessLogBucket.Arn, !Sub '${AccessLogBucket.Arn}/*']
            Condition: {Bool: {'aws:SecureTransport': 'false'}}
  TrailBucket:
    Type: AWS::S3::Bucket
    DeletionPolicy: Retain
    UpdateReplacePolicy: Retain
    Properties:
      BucketName: sf-log-archive-org-trail-locked-us-east-1
      ObjectLockEnabled: true
      ObjectLockConfiguration:
        ObjectLockEnabled: Enabled
        Rule: {DefaultRetention: {Mode: COMPLIANCE, Days: 400}}
      BucketEncryption:
        ServerSideEncryptionConfiguration:
          - BucketKeyEnabled: true
            ServerSideEncryptionByDefault: {SSEAlgorithm: 'aws:kms', KMSMasterKeyID: !GetAtt
TrailKey.Arn}
```

```
      PublicAccessBlockConfiguration: {BlockPublicAcls: true, BlockPublicPolicy: true,
IgnorePublicAcls: true, RestrictPublicBuckets: true}
      LoggingConfiguration: {DestinationBucketName: !Ref AccessLogBucket, LogFilePrefix:
cloudtrail-v2/}
  TrailBucketPolicy:
    Type: AWS::S3::BucketPolicy
    Properties:
      Bucket: !Ref TrailBucket
      PolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Sid: CloudTrailAclCheck
            Effect: Allow
            Principal: {Service: cloudtrail.amazonaws.com}
            Action: 's3:GetBucketAcl'
            Resource: !GetAtt TrailBucket.Arn
            Condition: {StringEquals: {'aws:SourceArn': !Sub 'arn:aws:cloudtrail:us-east-1:$
{ManagementAccountId}:trail/sf-org-trail-v2'}}
          - Sid: CloudTrailWriteOrganizationLogs
            Effect: Allow
            Principal: {Service: cloudtrail.amazonaws.com}
            Action: 's3:PutObject'
            Resource: [!Sub '${TrailBucket.Arn}/AWSLogs/${ManagementAccountId}/*', !Sub '$
{TrailBucket.Arn}/AWSLogs/${OrganizationId}/*']
            Condition:
              StringEquals:
                's3:x-amz-acl': bucket-owner-full-control
                'aws:SourceArn': !Sub 'arn:aws:cloudtrail:us-east-1:$
{ManagementAccountId}:trail/sf-org-trail-v2'
          - Sid: DenyNonTlsRequests
            Effect: Deny
            Principal: '*'
            Action: 's3:*'
            Resource: [!GetAtt TrailBucket.Arn, !Sub '${TrailBucket.Arn}/*']
            Condition: {Bool: {'aws:SecureTransport': 'false'}}
Outputs:
  TrailKeyArn: {Value: !GetAtt TrailKey.Arn}
```

Management account: the new organization trail

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Description: Multi-Region organization trail with log file validation, delivered to the
immutable archive.
Parameters:
  TrailKeyArn: {Type: String, Description: KMS key ARN from the sf-log-archive stack.}
Resources:
  OrgTrail:
    Type: AWS::CloudTrail::Trail
    Properties:
      TrailName: sf-org-trail-v2
      S3BucketName: sf-log-archive-org-trail-locked-us-east-1
      KMSKeyId: !Ref TrailKeyArn
      IsLogging: true
      IsMultiRegionTrail: true
      IsOrganizationTrail: true
      IncludeGlobalServiceEvents: true
      EnableLogFileValidation: true
      EventSelectors:
        - ReadWriteType: All
          IncludeManagementEvents: true
          DataResources:
            - {Type: 'AWS::S3::Object', Values: ['arn:aws:s3']}
            - {Type: 'AWS::Lambda::Function', Values: ['arn:aws:lambda']}
```

Page when anyone tries to stop, delete, or change a trail, its key, or the archive account

```JSON
{
  "source": [
    "aws.cloudtrail",
    "aws.kms",
    "aws.organizations"
  ],
  "detail-type": [
    "AWS API Call via CloudTrail"
  ],
  "detail": {
    "eventSource": [
      "cloudtrail.amazonaws.com",
      "kms.amazonaws.com",
      "organizations.amazonaws.com"
    ],
    "eventName": [
      "StopLogging",
      "DeleteTrail",
      "UpdateTrail",
      "PutEventSelectors",
      "ScheduleKeyDeletion",
      "DisableKey",
      "CloseAccount"
    ]
  }
}
```

sf-protect-log-archive: nobody in sf-log-archive can schedule key deletion, disable a key, or delete the buckets (attach to the account itself)

```JSON
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ProtectLogArchiveKeysAndBuckets",
      "Effect": "Deny",
      "Resource": "*",
      "Action": [
        "kms:ScheduleKeyDeletion",
        "kms:DisableKey",
        "s3:DeleteBucket",
        "s3:PutBucketObjectLockConfiguration"
      ]
    }
  ]
}
```

Management account: attach to every administrator role so no one can close sf-log-archive (SCPs do not bind the management account)

```JSON
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "NobodyClosesMemberAccounts",
      "Effect": "Deny",
      "Action": "organizations:CloseAccount",
      "Resource": "*"
    }
  ]
}
```

How we proved it: `describe-trails` shows `LogFileValidationEnabled` true and a `KmsKeyId` for `sf-org-trail-v2`; `get-object-lock-configuration` on the new bucket shows COMPLIANCE for 400 days; `delete-object --version-id` on a locked object, run as an administrator of `sf-log-archive`, fails with AccessDenied (the management account has no access to the bucket, so a delete from there would prove nothing); a test `StopLogging` on a scratch trail paged the security team from the rule in the management account; and `aws cloudtrail validate-logs` passes for the last 24 hours. The old trail is stopped, not deleted, and its bucket stays for the retention you already owe; the three Kestrel single-Region trails were retired the same way. **Rollback:** there is none for the lock, by design. If the new trail must change, change the CloudFormation stack and let the stack's own alarm page you.

What the lock does not stop, and what does. Object Lock in COMPLIANCE mode resists a delete by every principal, the root user included, until retention ends. Two other paths exist, and each has a control instead of a lock. Closing the `sf-log-archive` account deletes its locked objects, and an account-level policy cannot bind the management account, so `organizations:CloseAccount` is denied to every administrator role in the management account and alarmed there. Scheduling the key for deletion would leave the logs unreadable, so the key is retained when its stack is deleted, the `sf-protect-log-archive` service control policy stops anyone in `sf-log-archive` from scheduling its deletion or disabling it, and the same alarm pages on both attempts. The tamper alarm lives in the management account in us-east-1, because that is where the API calls are recorded, and it exempts no one: a planned change pages too, and the on-call confirms it. The management account's single hardware-MFA root user is the one principal these controls do not bind, which is why its sign-in is alarmed (week 4).

What the trail records

The trail records management events in every account and Region, and S3 and Lambda data events for every bucket and function. The 2019 trail already recorded S3 data events on every bucket (since 2023); `platform-landing-zone#723` carried the same selectors onto `sf-org-trail-v2`, so the answer to "who downloaded that object?" exists for every bucket, and the data-event cost in What the detection stack costs did not change. Server access logging, which is separate, went on for 233 buckets that had it off (`SF-SEC-051`, `platform-landing-zone#729`): one log bucket per account and Region, replicated to the archive, because S3 requires the log bucket to share the source bucket's account and Region.

MFA delete, accepted

`SF-SEC-049`: MFA delete is off on every bucket. We recommend that StrataForge not enable MFA delete, and Marcus Oyelaran, Chief Information Security Officer, agreed. It requires the root user's credentials, which week 4 removed, and it blocks the lifecycle rules the data lake depends on. Object Lock is the stronger control and is in place on the trail bucket and the OTA distribution bucket.

Risk acceptance for SF-SEC-049

Accepted by: Marcus Oyelaran, Chief Information Security Officer, on August 14, 2026.
Expires: August 14, 2027.

Statement: StrataForge does not enable MFA delete: it requires the root user's credentials, which centralized root access management has removed, and it blocks the lifecycle rules the data lake depends on. Versioning, Object Lock on the log archive, and the organization trail's S3 data events protect against deletion instead.

Compensating controls.

- Versioning on every bucket that holds production data
- Object Lock (compliance mode) on the trail bucket
- S3 data events on every bucket (organization trail, since 2023)
- GuardDuty S3 Protection on every bucket

What the detection stack costs

The detection stack now costs about **$119,645 a month**, up from about $58,900. The increase is $60,745 a month, or about $728,940 a year. The prices are the AWS list prices retrieved October 13, 2026, each Region at its own rate, with volume tiers applied across the organization, less the 7% discount in StrataForge's private pricing agreement. That is the basis the monthly bill uses, so these figures can be compared with it line by line.

Monthly cost of the detection stack before and after week 5, with the trail's data events shown separately because they did not change.

Component	Before	After	Change
Config: configuration items	$20,275	$29,248	+$8,973
GuardDuty: VPC flow and DNS logs	$11,879	$16,888	+$5,009
GuardDuty: S3 Protection	$11,168	$16,442	+$5,274
GuardDuty: Runtime Monitoring, EKS nodes	$0	$11,455	+$11,455
GuardDuty: Runtime Monitoring, EC2 instances	$0	$10,718	+$10,718
Config: rule evaluations	$6,982	$10,222	+$3,240
GuardDuty: CloudTrail management events	$5,855	$8,764	+$2,909
Security Hub CSPM: security checks	$2,741	$4,125	+$1,384
Inspector: EC2 scanning	$0	$3,352	+$3,352
GuardDuty: Lambda network activity	$0	$2,863	+$2,863
GuardDuty: EKS audit log monitoring	$0	$1,632	+$1,632
GuardDuty: RDS Protection (Aurora)	$0	$1,204	+$1,204
Inspector: ECR image scanning	$0	$1,043	+$1,043
Inspector: Lambda code scanning	$0	$1,021	+$1,021
Inspector: Lambda standard scanning	$0	$507	+$507
GuardDuty: Malware Protection for EC2 (finding-initiated scans)	$0	$140	+$140
KMS: organization trail key and requests (new with sf-org-trail-v2)	$0	$21	+$21
Detection stack	**$58,900**	**$119,645**	**+$60,745**

Component	Before	After	Change
CloudTrail S3 and Lambda data events, organization trail	$56,764	$56,764	$0
With the trail's data events	**$115,664**	**$176,409**	**+$60,745**

How the "before" is measured. The July 2026 bill's lines for AWS Config, AWS Security Hub CSPM, and Amazon GuardDuty: 72% of the August run-rate in StrataForge's 43 accounts and nothing in Kestrel's five, which had none of the three. It is the same figure the cost engagement's baseline report shows for July 2026. The protection plans, Inspector, and the new trail key had no cost before week 5. The "after" is a calculation from the quantities below, which we read from the organization's accounts, at the prices above.

The monthly quantities for the three services the bill already carries. Their tiered prices are applied per account.

Monthly quantity	Amount
Config configuration items, continuous recording	10,483,000
Config configuration items, periodic recording	0
Config rule evaluations	15,724,500
Security Hub CSPM security checks	5,064,420
GuardDuty VPC flow and DNS logs, GB	38,284
GuardDuty CloudTrail management events	2,356,000,000
GuardDuty S3 Protection, S3 data events	42,005,274,000

Protection plans and Amazon Inspector

10 lines are new. Each is priced at its Region's rate from the resources we counted. Where the usage is a quantity the organization has not yet generated, we say so and state the assumption, so a different assumption can be priced in a minute.

The new GuardDuty protection plans and Amazon Inspector scan types, on the bill's basis. Each protection plan has a free trial (GuardDuty's is 30 days), so the first bill after the trials is the real test of this table.

Component	Quantity a month	Basis	Monthly cost	What it assumes
Inspector: EC2 scanning	2,026,468	Counted instance-hours	$3,352	2,813 running instances, each at its Region's rate
Inspector: Lambda standard scanning	1,306,700	Counted function-hours	$507	1,790 functions
Inspector: Lambda code scanning	1,306,700	Counted function-hours	$1,021	1,790 functions
Inspector: ECR image scanning	10,200	Assumed images a month	$1,043	340 repositories times 30 new images a month, each scanned on push and re-scanned 2 times
GuardDuty: Runtime Monitoring, EKS nodes	17,844	vCPU-months	$11,455	vCPUs of running instances in EKS node groups
GuardDuty: Runtime Monitoring, EC2 instances	22,653	vCPU-months	$10,718	vCPUs of the other running instances
GuardDuty: EKS audit log monitoring	960 million	Assumed audit events	$1,632	16 clusters times 60 million events a month
GuardDuty: RDS Protection (Aurora)	1,260	vCPU-months	$1,204	vCPUs of Aurora instances
GuardDuty: Lambda network activity	7,613 GB	Assumed	$2,863	400 bytes of network metadata per invocation, every function treated as VPC-attached
GuardDuty: Malware Protection for EC2 (finding-initiated scans)	5,000 GB	Assumed scanned	$140	10 scans a month of 500 GB

What is not in the totals

1 line is left out of the totals, and IAM Access Analyzer's external-access findings are free, so they are not in them either.

What the detection stack costs that the figures above leave out, and why.

Service	Component	Why it is not in the totals
Amazon EventBridge and Amazon SNS	Routing findings to PagerDuty	Not in the price file; a few thousand events a month cost pennies, and EventBridge events from AWS services are free.

Our recommendation on cost. Do not turn a service off to save money. Look first at the trail's data events ($56,764 a month, the largest single line): the organization trail has logged S3 data events on every bucket since 2023, and most buckets do not need it. Narrow the event selectors to the buckets that hold production data, customer data, or release artifacts, and keep management events on everywhere. That is a decision with a security trade-off, so it belongs to the monthly cost review with Marcus Oyelaran's team in the room. We left the trail's data events alone in this engagement.

What pages whom: the runbooks

Every alert above that pages a person has a runbook, written in week 5 and handed over in full in week 6. Each lives in `platform-landing-zone/runbooks/` next to the alarm that triggers it, so changing the alarm and changing the procedure happen in one pull request.

The 10 runbooks, with what triggers each, who is paged, and who owns it. The first 15 minutes of each are in the week-6 handover report.

ID	Runbook	Trigger	Pages	Owner	Linked finding types
RB-01	OTA signing key used by an unexpected principal	EventBridge: kms:Sign on alias/ota-firmware-signing-prod by any principal other than ota-release-signer	Fleet OTA (high urgency)	Sam Okafor	`ota-signing-user`, `ota-test-key-policy`
RB-02	Robot fleet authorization failures or message-volume anomaly	AWS IoT Device Defender Detect violation on a robot thing group (authorization failures or messages sent)	Fleet Cloud (high urgency)	Sam Okafor	`iot-policy-scope`, `iot-shared-certs`
RB-03	Root or break-glass sign-in	EventBridge: ConsoleLogin by the management account root user or breakglass-01/02, or any sts:AssumeRoot call	Security (high urgency)	Nadia Petrova	`break-glass`, `root-credentials`
RB-04	Credentials used from outside AWS	GuardDuty: UnauthorizedAccess:IAMUser/InstanceCredentialExfiltration.OutsideAWS	Security (high urgency)	Nadia Petrova	`imdsv2`, `workload-admin-roles`
RB-05	Access key exposed	AWS Health event AWS_RISK_CREDENTIALS_EXPOSED, or a secret-scanning alert from GitHub	Identity (high urgency)	Grace Liu	`access-key-retirement`, `lambda-env-secrets`
RB-06	Logging or detection turned off	GuardDuty: Stealth:IAMUser/CloudTrailLoggingDisabled, or an SCP denial on GuardDuty, Security Hub CSPM, AWS Config, or CloudTrail changes	Security (high urgency)	Nadia Petrova	`cloudtrail-log-archive`, `guardduty-org`, `scp-guardrails`
RB-07	Something became public	IAM Access Analyzer finding with public access, or Security Hub CSPM S3.2, RDS.1, or EC2.19 failing	Security (low urgency; high for production accounts)	Nadia Petrova	`s3-public-bucket`, `ebs-snapshot-public`, `rds-snapshot-public`, `admin-ports-internet`, `notebook-exposed`

ID	Runbook	Trigger	Pages	Owner	Linked finding types
RB-08	Kubernetes cluster compromise	GuardDuty: AttackSequence:EKS/CompromisedCluster or a Runtime Monitoring finding on an EKS node	Fleet Cloud (high urgency)	Sam Okafor	`eks-endpoint-access`, `eks-operations`
RB-09	Compromised instances in a group	GuardDuty: AttackSequence:EC2/CompromisedInstanceGroup or CryptoCurrency:EC2/BitcoinTool.B!DNS	Security (high urgency)	Nadia Petrova	`notebook-exposed`, `ci-controller-exposed`, `admin-ports-internet`
RB-10	Critical control failure on a production account	Security Hub CSPM: a CRITICAL or HIGH failed control in a production account (routed from the us-east-1 aggregation Region)	Platform (low urgency, business hours)	Elena Vasquez	`securityhub-org`, `config-org`

What is not finished

3 findings moved this week and are not closed, counting the one accepted in writing. Each has an owner and a date.

Week-5 work that did not close its finding.

ID	Severity	Finding	What remains	Owner	Due
SF-SEC-043	Medium	750 default VPCs, most in Regions nobody uses, have open default security groups and network ACLs	Replace the open default network ACLs in the working Regions' VPCs	Elena Vasquez	Oct 30
SF-SEC-054	Low	EKS control-plane logging is off on 15 of 16 clusters, and every cluster still uses the aws-auth ConfigMap	Move every cluster to access entries and upgrade 1.31 and 1.32 clusters	Sam Okafor	Nov 13
SF-SEC-049	Low	MFA delete is off on every bucket	Accepted in writing by Marcus Oyelaran; renew or retire at expiry on August 14, 2027	Nadia Petrova	Aug 14

The default VPC work (`SF-SEC-043`, `platform-landing-zone#717`) deleted the default VPCs in the Regions nobody uses and restricted the default security groups and flow logs everywhere; the open default network ACLs in the working Regions wait for owners to confirm nothing depends on them. The EKS work (`SF-SEC-054`, `fleet-infra#4373`) turned on control-plane logging and moved the production clusters to access entries; the migration of the remaining clusters and the upgrade of the 1.31 and 1.32 clusters follow the Kubernetes version calendar: 1.31 reaches the end of extended support on November 26, 2026, and EKS will then upgrade the control plane automatically, with no rollback, so that date is the one to hold.

Appendix: coverage by account

For each account, the number of the 17 Regions where the service runs, before to after.

Regions covered, out of 17, for each account and service, before to after. The organization trail covers every account in every Region and is not shown.

Account	ID	GuardDuty	Security Hub CSPM	AWS Config	IAM Access Analyzer	Amazon Inspector
sf-management	…0007	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
sf-log-archive	…2291	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
sf-security-tooling	…3374	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
sf-forensics	…3618	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
sf-network-hub	…1145	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
sf-shared-services	…1188	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
sf-identity	…1202	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
sf-cicd	…1356	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
sf-backup	…1407	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
fleet-ingest-prod	…5012	6 to 17	3 to 17	6 to 17	1 to 17	1 to 17
fleet-api-prod	…5044	6 to 17	3 to 17	6 to 17	1 to 17	1 to 17
fleet-data-prod	…5071	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
fleet-analytics-prod	…5093	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
fleet-ota-prod	…5117	6 to 17	3 to 17	6 to 17	1 to 17	1 to 17
fleet-ota-staging	…6148	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
fleet-remote-ops-prod	…5139	6 to 17	3 to 17	6 to 17	1 to 17	1 to 17
fleet-eu-prod	…5160	6 to 17	3 to 17	6 to 17	1 to 17	1 to 17

Account	ID	GuardDuty	Security Hub CSPM	AWS Config	IAM Access Analyzer	Amazon Inspector
fleet-jp-prod	…5182	6 to 17	3 to 17	6 to 17	1 to 17	1 to 17
customer-portal-prod	…5215	6 to 17	3 to 17	6 to 17	1 to 17	1 to 17
digital-twin-prod	…5238	6 to 17	3 to 17	6 to 17	1 to 17	1 to 17
fleet-staging	…6014	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
fleet-dev	…6037	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
fleet-perf	…6059	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
fleet-eu-staging	…6081	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
customer-portal-dev	…6103	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
digital-twin-dev	…6126	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
crucible-sim-prod	…7019	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
crucible-training-prod	…7042	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
crucible-data	…7065	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
crucible-research	…7088	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
crucible-dev	…7101	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
crucible-models	…7124	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
erp-sap-prod	…8016	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
erp-sap-nonprod	…8039	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
plm-prod	…8061	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
mfg-mes-prod	…8084	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
corp-data-warehouse	…8107	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17

Account	ID	GuardDuty	Security Hub CSPM	AWS Config	IAM Access Analyzer	Amazon Inspector
corp-apps	…8129	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
kestrel-prod	…9031	0 to 17	0 to 17	0 to 17	1 to 17	0 to 17
kestrel-staging	…9054	0 to 17	0 to 17	0 to 17	1 to 17	0 to 17
kestrel-dev	…9076	0 to 17	0 to 17	0 to 17	1 to 17	0 to 17
kestrel-data	…9098	0 to 17	0 to 17	0 to 17	1 to 17	0 to 17
kestrel-legacy-mgmt	…9120	0 to 17	0 to 17	0 to 17	1 to 17	0 to 17
sandbox-platform	…4011	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
sandbox-robotics	…4033	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
sandbox-ml	…4056	6 to 17	3 to 17	6 to 17	1 to 17	0 to 17
sandbox-hackathon	…4078	6 to 17	3 to 17	0 to 17	1 to 17	0 to 17
suspended-legacy-ci	…4099	6 to 17	3 to 17	0 to 17	1 to 17	0 to 17

Pull request ledger

Author for every pull request: Charles Sieg of Vantalect. Reviewers are the engineers who own the account or the system.

The week-5 pull requests that merged first: repository and number, files, reviewer, merge date, rollout, and the finding each addresses.

Pull request	Title	Files	Reviewer	Merged	Rollout	Finding
platform-landing-zone#685	Config: aggregator in sf-security-tooling; recorders in every governed Region	`stacksets/config-aggregator/template.yaml`; `control-tower/landing-zone.yaml`	Elena Vasquez	Aug 10	Enabled in the sandbox accounts first, then every account, Region by Region.	Closes SF-SEC-022
platform-landing-zone#688	GuardDuty: delegated admin in 17 Regions; auto-enable all members and plans	`security-services/guardduty.yaml`; `scripts/security-services/guardduty-org.sh`	Nadia Petrova	Aug 10	Enabled in the sandbox accounts first, then every account, Region by Region.	Closes SF-SEC-023
platform-landing-zone#690	Logging: sf-org-trail-v2 into an Object Lock (compliance) bucket in sf-log-archive	`stacksets/log-archive/org-trail-bucket.yaml`; `management/org-trail.yaml`; `stacksets/security-alarms/trail.yaml`	Nadia Petrova	Aug 10	Deployed to sf-management first and verified, then to production.	Closes SF-SEC-025
platform-landing-zone#695	Security Hub CSPM: central configuration, CIS v5.0.0 + FSBP, routing to PagerDuty	`security-services/securityhub-config-policy.json`; `stacksets/finding-routing/template.yaml`	Nadia Petrova	Aug 11	Enabled in the sandbox accounts first, then every account, Region by Region.	Closes SF-SEC-038
platform-landing-zone#701	Control Tower: govern all 17 Regions; enroll Kestrel, sandbox-hackathon, suspended-legacy-ci	`control-tower/landing-zone.yaml`; `stacksets/ct-execution-role/template.yaml`	Elena Vasquez	Aug 11	Enabled in the sandbox accounts first, then every account, Region by Region.	Closes SF-SEC-039
platform-landing-zone#707	IAM Access Analyzer: organization analyzers in 17 Regions	`security-services/access-analyzer.yaml`	Nadia Petrova	Aug 12	Enabled in the sandbox accounts first, then every account, Region by Region.	Closes SF-SEC-044

The remaining week-5 pull requests, in merge order.

Pull request	Title	Files	Reviewer	Merged	Rollout	Finding
platform-landing-zone#713	Inspector: delegated admin; EC2, ECR, Lambda, and Lambda code scanning	security-services/inspector.yaml	Nadia Petrova	Aug 12	Enabled in the sandbox accounts first, then every account, Region by Region.	Closes SF-SEC-042
platform-landing-zone#717	Network: delete default VPCs in unused Regions; restrict default security groups; flow logs	ops/delete-default-vpcs.sh; stacksets/vpc-baseline/template.yaml	Elena Vasquez	Aug 13	Deployed to sf-management first and verified, then to production.	Addresses SF-SEC-043; not yet closed
platform-landing-zone#723	Logging: S3 and Lambda data events for every bucket and function on sf-org-trail-v2	management/org-trail.yaml	Nadia Petrova	Aug 13	Deployed to sf-management first and verified, then to production.	Addresses SF-SEC-049; not yet closed
fleet-infra#4373	EKS: control-plane logging; access entries for production clusters	apps/fleet-api/lib/eks-cluster.ts; apps/ingest/lib/eks-cluster.ts; apps/fleet-api/lib/access-entries.ts	Sam Okafor	Aug 14	Deployed to fleet-staging first and verified, then to production.	Addresses SF-SEC-054; not yet closed
platform-landing-zone#729	S3: server access logging to per-account log buckets, replicated to the archive	stacksets/s3-access-logs/template.yaml; ssm/automation/enable-bucket-logging.yaml	Elena Vasquez	Aug 14	Deployed to sf-management first and verified, then to production.	Closes SF-SEC-051

© 2026 Vantalect, LLC. Sample deliverable; StrataForge Robotics is fictional.

AWS, Amazon Web Services, AWS Graviton, and the AWS service names are trademarks of Amazon.com, Inc. or its affiliates. Kubernetes is a registered trademark of The Linux Foundation. GitHub is a trademark of GitHub, Inc. NVIDIA and Isaac Sim are trademarks of NVIDIA Corporation. SAP and SAP S/4HANA are trademarks of SAP SE. Okta is a trademark of Okta, Inc. PagerDuty is a trademark of PagerDuty, Inc. SOC 2 is a registered trademark of the American Institute of Certified Public Accountants. CIS Benchmarks is a trademark of the Center for Internet Security, Inc. Vantalect is not affiliated with or endorsed by any of them. StrataForge Robotics is fictional.