



No Regrets

A primer to getting setup in AWS for the long haul

Chris Farris
PrimeHarbor Technologies

Who Am I?



- Built the cloud security programs for some media companies
- Founder: fwd:cloudsec conference
- Rants a lot on ~~Twitter~~ Mastodon and Bluesky
- Somehow was named a Security Hero by AWS
- Cloud Security Consultant

— aws —
security
HERO

THAT'S WHAT I DO:
I DRINK AND
I KNOW THINGS.





And I'm moving to Porto Area



Agenda & Roadmap

- Why?
- Get Organization
- Avoid Identity Crisis
- Manage your Workloads
- Manage your Networks
- GuardRails!

But Why?



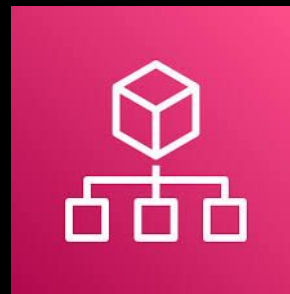
Welcome to your new AWS Account



AWS Organizations



- Free!
- Do it right away!
- Always use a fresh account!
- Everything else depends on this!



What About Control Tower?

Friends don't let friends run control tower

CloudTrail

- Audit Log of everything you do
- Free (mostly)
- Do it once and forget about it



```
{
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROREDACTEDL6IX:chris@primeharbor.com",
    "arn": "arn:aws:sts::xxxxx:assumed-role/AWSRes",
    "accountId": "112233445566",
    "accessKeyId": "ASIAREDACTEDZKK",
  },
  "eventTime": "2025-09-25T20:29:58Z",
  "eventSource": "signin.amazonaws.com",
  "eventName": "ConsoleLogin",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "77.54.99.99",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS",
    "Safari/537.36",
  "responseElements": {
    "ConsoleLogin": "Success"
  },
  "additionalEventData": {
    "MobileVersion": "No",
    "MFAUsed": "No"
  },
  "readOnly": false,
  "eventType": "AwsConsoleSignIn",
  "managementEvent": true,
  "recipientAccountId": "112233445566",
  "eventCategory": "Management",
}
```

Avoid Billing Surprises



- Enable Billing Alerts
- Use multiple levels
 - \$100, \$250, \$500, \$1000, \$5000
- Don't warn about credit depletion
- Consider AWS Budgets



Yes, leaking your keys to a cryptominer was a bad move

Disable Root Users



Identity and Access Management (IAM)

Search IAM

Dashboard

▼ **Access management**

- User groups
- Users
- Roles
- Policies
- Identity providers
- Account settings
- Root access management** New

▼ **Access reports**

- Access Analyzer

Root access management

☐	pht-research [redacted] aws+research@primeharbor.com	✔ Not present
☐	pht-cloudgoat [redacted] aws+pht-cloudgoat@primeharbor.com	✔ Not present
☐	pht-canarytoken1 [redacted] aws+pht-canarytoken1@primeharbor.com	✔ Not present
☐	pht-payer management account [redacted] aws+pht-payer@primeharbor.com	
☐	pht-ss0 [redacted] aws+pht-ss0@primeharbor.com	✔ Not present
☐	RealEstateAgent [redacted]	⚠ Present
☐	pht-minecraft [redacted]	✔ Not present

Set your account contacts

- So you know when you commit an access key to GitHub



Account [Info](#) [Close account](#)

Account details [Actions](#)

Name pht-payer	Service provider Amazon Web Services, Inc.
ID 	ARN 

Account display settings - new [Edit](#)

Account color
☒ Unset

Contact information [Info](#) [Edit](#)

Full name Chris Farris	Company name PrimeHarbor Technologies, LLC	Address  Marietta, GA 30068 US
Phone number +14045383275	Website URL https://primeharbor.com	

Alternate contacts

You can add alternate contacts for Billing, Operations, and Security communications to ensure the correct people are notified for each topic. As a primary account holder, you will continue to receive all email communications. [Learn more](#)

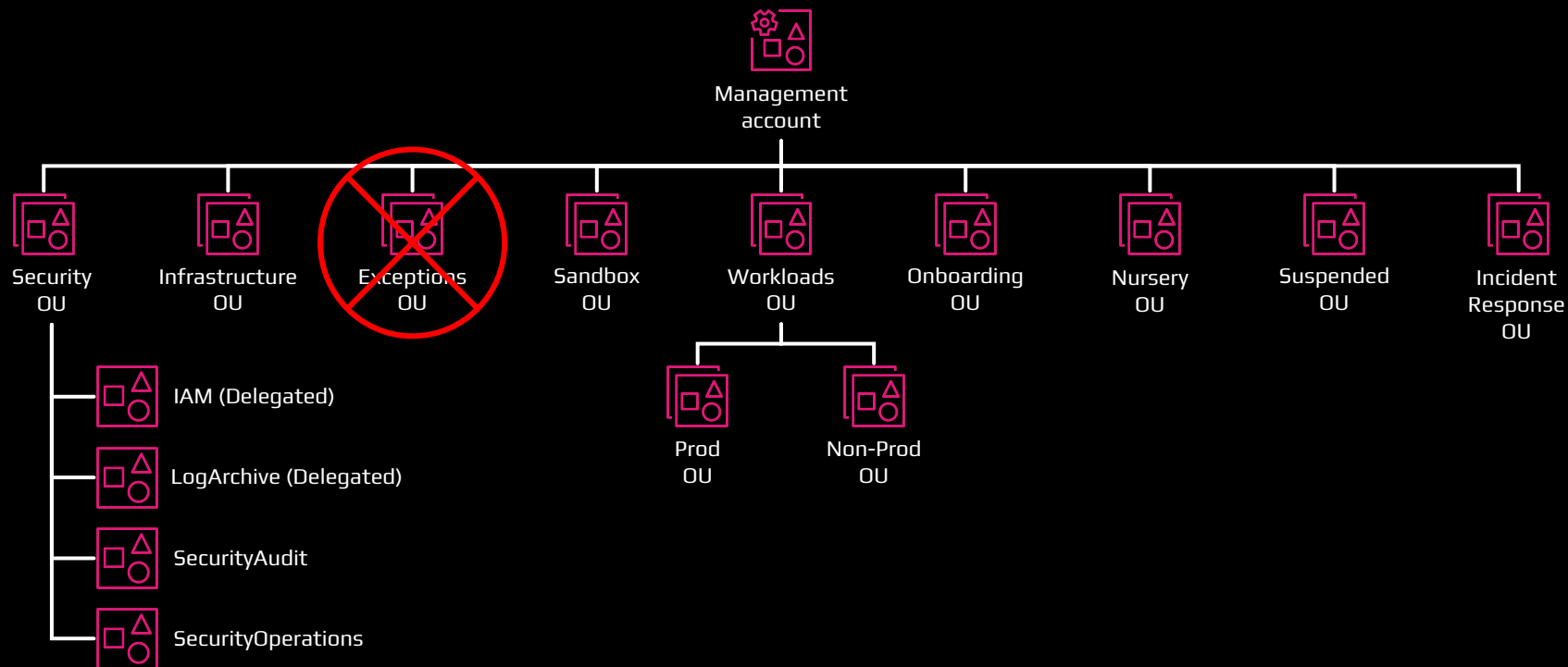
Billing contact	Operations contact	Security contact
Full name Chris Farris	Full name J. Chris Farris	Full name Chris Farris
Title CFO	Title CEO	Title Global CISO
Email address chris@primeharbor.com	Email address chris@primeharbor.com	Email address chris@room17.com
Phone number +14048199737	Phone number +14045383275	Phone number +14045383275

MultiAccount Strategy

- Blast Radius
- Security Boundary
- Accountability
- Financial Reporting
- Easy to clean up



Organization hierarchy





Manage Your Identities



IAM Users Not Even Once



Attackers use which initial access method most often?

Lost/leaked access
keys/credentials

#4



66 %

Valid IAM Credentials

↑
1/3

Of those are **root credentials**

[20% of all IAM credential access method use]



13 %

Public facing EC2 instance



© 2023, Amazon Web Services, Inc. or its affiliates.

AWS Identity Center



- Grants access to all the accounts
- Works for Console & CLI
- One login & password
 - And MFA!
- Easily integrates to Google Workspace and Entra ID

Do this now, even if you're just one person!



AWS access portal

Accounts

Applications

AWS accounts (27)

🔍 *Filter accounts by name, ID, or email address*

▶  linuxshowcase
[REDACTED] | aws+linuxshowcase@primeharbor.com

▶  pht-aws-iq
([REDACTED] | [REDACTED])

▶  pht-backups
[REDACTED] | aws+backups@primeharbor.com

▼  pht-bedrock
([REDACTED] | aws+pht-bedrock@primeharbor.com)

AdministratorAccess | [Access keys](#) 🔗

CloudAdminAccess | [Access keys](#) 🔗

CloudAdminReadOnly | [Access keys](#) 🔗



Separate your workloads

- Dev, Test, Prod
- Sandbox accounts
- Use "Plus Addressing"
 - aws+<name>@yourcompany.ai
- Have an existing account?
 - Invite it to the Organization

What about VPCs?



Identity is the new perimeter

or

You need to defend three dimensionally

or

*"Cute network controls you have there
if would be a shame if someone just
routed around them"*





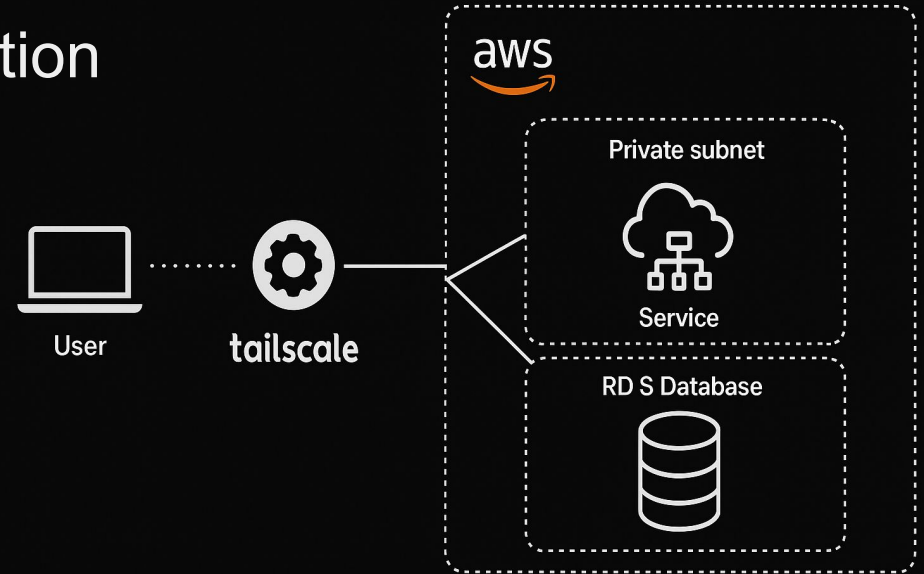
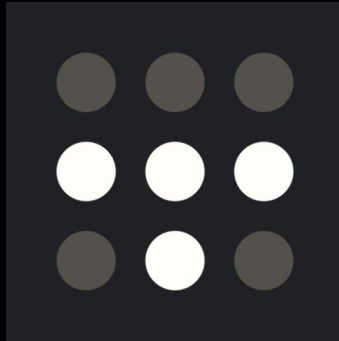
Manage Your VPCs

- Delete Default VPCs
- Allocate IPs from a google sheet.
- Big Ranges, but not stupid big

Don't put things on the internet!!!!



- #2 way things get compromised
- Choose Private Subnets!
- tailscale is a free(ish) solution



Choose your guardrail



laC scanning



Auto remediation



Service control policies
Resource control policies



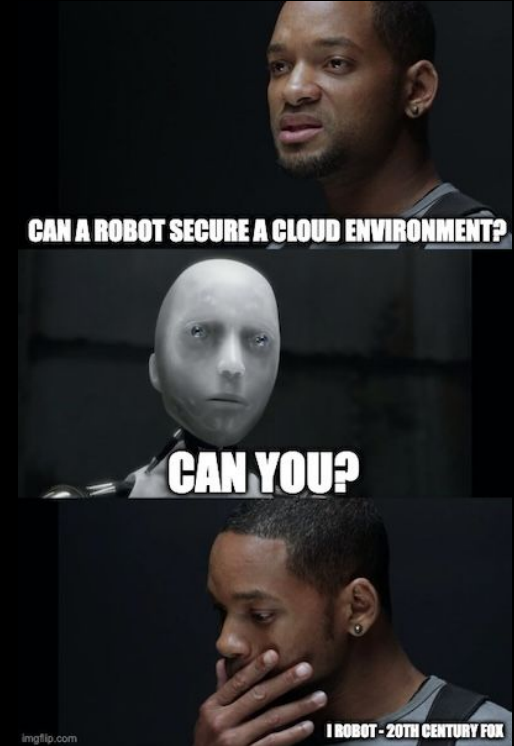
Auto-Remediation like nuclear power. One accident, and suddenly everyone is against the idea.

Chris Farris

Three Laws of Auto Remediation



1. A bot must never harm stateful data or allow stateful data to come to harm.
2. A bot must act with utmost haste so functionality doesn't become dependent on a misconfiguration.
3. A bot must announce its existence and tell a carbon-based life form what it did and why.





A security invariant is a system property that relates to the system's ability to prevent security issues from happening. Security invariants are statements that will always hold true for your business and applications. — AWS

. . . will always hold true . . .



“No one can create a VPC”

vs.

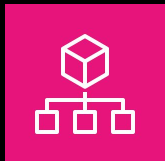
“Only the network engineering team can create a VPC”

Examples



- “Only the network engineering team may create a VPC, alter route tables, or attach an IGW”
- “Only the security and privacy team may make an S3 bucket public”
- “Only procurement may subscribe to or accept an offer in AWS Marketplace”
- “Only cloud engineering can enable new opt-in regions”

Enforcing invariants



Organization-based policies

- Service control policies
- Resource control policies
- Declarative policies



Identity-based policies

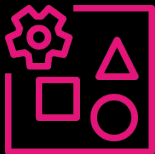
- Permission policies
- Permission boundaries



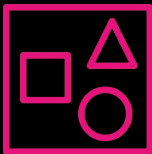
Automation/guardrails

- Declarative controls (Block Public Access)
- Automated remediation

Service control policies



Managed via the
AWS Organizations
management account
(aka “payer”)



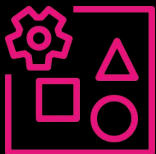
Defines the
“maximum
permissions of the
account”

(This includes the root user)



Applies to **your
identities**

Resource control policies



Managed via the Organizations Management Account (aka “payer”)



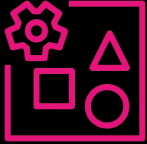
Applies to all principals – every AWS Customer



Only some services for now:

S3, STS/IAM, SQS, Secrets Manager

Declarative policies



Managed via
Organizations

But not IAM policies



Enforced at the
service's control
plane

This exists outside
of IAM



Supports:

- EBS Snapshots
- AMI
- VPC
- IMDSv2



*SCPs, RCPs, and permissions
boundaries **don't grant permissions**,
they define the **maximum
permissions available***



GuardRail Resources

Blog: [Defining Security Invariants](#)

GitHub Repo: [aws-organizational-policies](#)

Blog: [Implementing Security Invariants in an AWS Management Account](#)

GitHub Repo: [pht-payer-invariants](#)

AWS Docs: [Service Authorization Reference](#)

Actions, resources, and condition keys for AWS services

That's too much work!



<https://github.com/primeharbor/org-kickstart>

<https://github.com/primeharbor/pht-account-configurator>

I'm available for consulting!
<https://pht.us>

QUESTIONS?



@jcfarris [@infosec.exchange]



<https://github.com/jchrisfarris>



<https://www.linkedin.com/in/jcfarris>



<http://www.chrisfarris.com>

<https://www.primeharbor.com>

<https://pht.us/invariants>

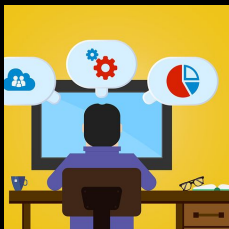
<https://pht.us/CDPT25>



Security spectrum



Invariants live here



Educated and empowered developers



Architectural and design reviews



IaC scanning



Prevention



Auto remediation



Spreadsheet hell

Time before appearing in production

Time after appearing in production

How to build an IAM Invariant



1. Define invariant – plain language
2. Determine actions
3. Determine resources
4. Determine “principals” (if SCP)
5. Determine conditions/define the exceptions

Define invariant in plain language



“Only the security and privacy team may make an Amazon S3 bucket public”

- Specific – “. . . make an Amazon S3 bucket public”
- Enforceable – Use S3 Block Public Access with SCP
- Realistic – Teams can create buckets,
they cannot remove the default BPA
- Avoids exceptions – “Only the security and privacy team . . .”