

# AWS re:Invent

## Intrusion Detection in the Cloud

Greg Roth, AWS Identity & Access Management

Don Bailey, AWS Security

November 14<sup>th</sup>, 2013



# Why should you care about this?

- Change management / monitoring is a good thing™
- Traditional intrusion detection may not detect AWS-specific environment changes
- Figure it out now, rather than later
- News flash: The bad guys know about the cloud

# So let's geek for a bit

- Intrusion detection in your AWS environment
- Universal adversary tactics to focus on
- AWS-specific security features to build with
- AWS-specific intrusion detection mechanisms w/ demos!
- Other tips, resources, Q&A



# Can you have your IDS in AWS?



- Short answer: YES!
- What IS an intrusion detection system?
  - System that monitors environment; alerting to detected intrusions.
- On premise, your IDS takes advantage of that environment's features.
- Within your AWS environment, you should do the same: Your AWS-specific IDS will likely NOT look like your traditional IDS. That's OK!



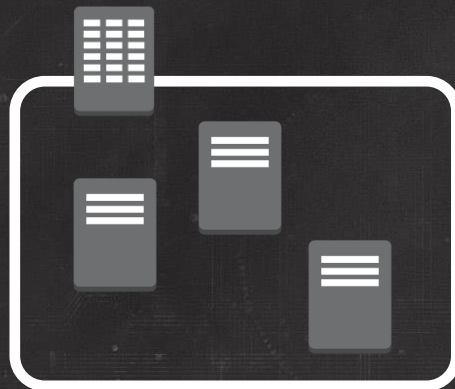
- Operating System
- Processes
- Files

- Packets
- Flows

?



Server



Network



Cloud



## Traditional IDS



Customer



Objects



Amazon S3 Bucket



Instances



Security Group



Internet Gateways



VPC Subnet



Groups,  
Users,  
Credentials



Amazon RDS  
DB Instances



Applications

This Talk



Configuration

AWS



Amazon S3



Amazon EC2



Amazon VPC



IAM



Amazon RDS



Elastic Beanstalk

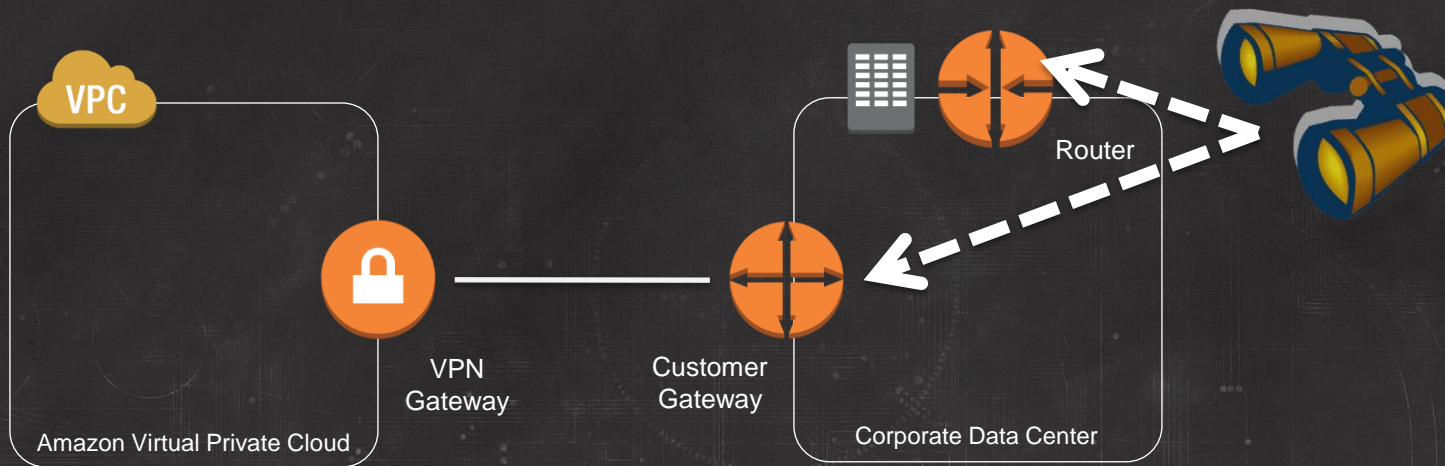


# Wait! I want MY IDS in AWS!

- I.e., “traditional” IDS – rackable, stackable, network-sniffing box that streams alerts night & day
- You ♥ traditional IDS, for a number of reasons, not all of them your own, eg., compliance.
- No worries! You can still do that too in AWS

# Traditional IDS in AWS

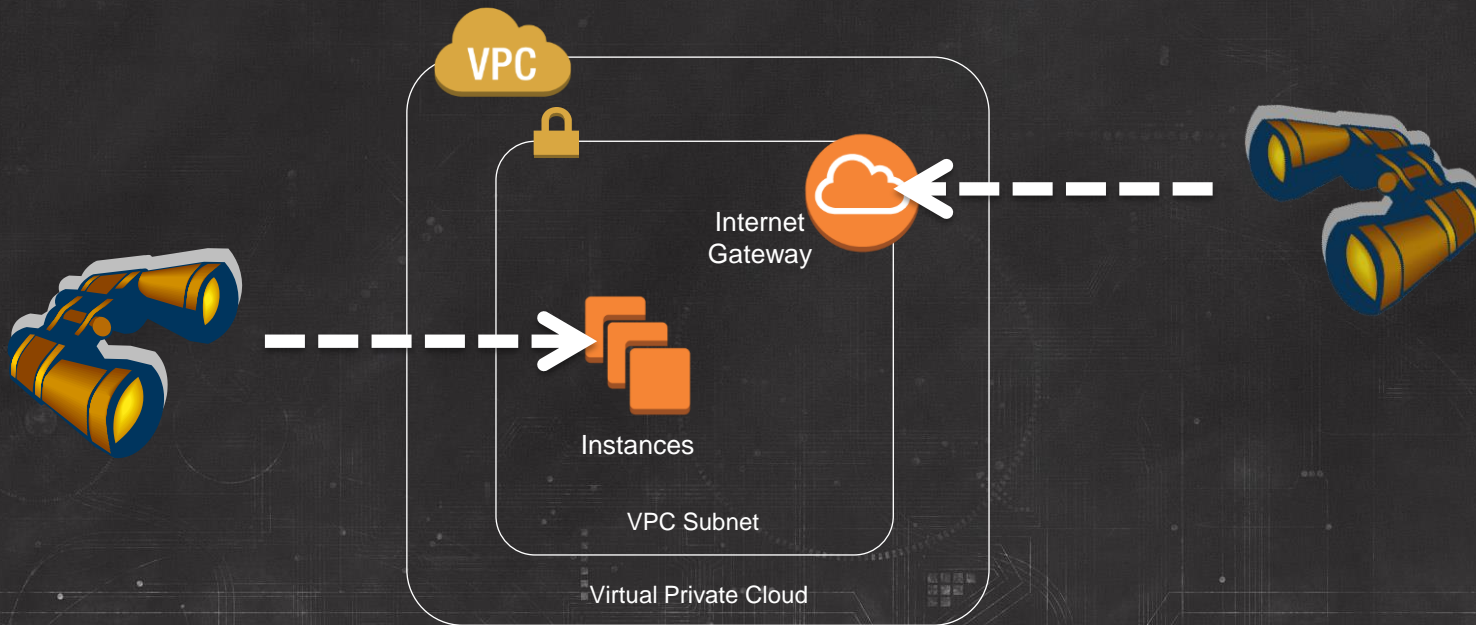
- On premises, VPC endpoint





# Traditional IDS in AWS

- In cloud, as VPC NAT gateway or on-instance



# Traditional IDS in AWS

- On premises, VPC endpoint
- In cloud, as VPC NAT gateway or on-instance
- Numerous AWS technology partners here
- Visit their booths ... or the AWS Marketplace!

**SOPHOS**



**ALERTLOGIC**  
Security. Compliance. Cloud.



**TREND**  
MICRO



**CloudPassage**

## Categories

[All Categories](#)
[Software Infrastructure](#)
[Security](#)

## Filters

[Operating System](#) ▾

Windows releases:

All

Linux/UNIX distributions:

All

## Software Infrastructure >

**Security (152 results)** showing 1 - 10

1 2 3 4 5 ... 16 ▶



### NGINX Plus - Amazon Linux AMI

★★★★★ (2) | Version 1.2 | Sold by [Nginx Inc.](#)
**\$0.04 to \$0.68/hr** for software + AWS usage fees

NGINX Plus AMI for AWS is provided by the original creators of NGINX web server. Run by over 40% of web sites hosted on AWS (ref. Netcraft's October 2012 Web Server Survey), ...

Linux/Unix, Amazon Linux 2013.03 | 64-bit Amazon Machine Image (AMI)



### Alert Logic Threat Manager for EC2

★★★★★ (2) | Version 1.2 | Sold by [Alert Logic](#)
**\$0.48 to \$1.64/hr** for software + AWS usage fees

Threat Manager for EC2 is the first Network Intrusion Detection (IDS) service specifically designed for AWS. Using this service you can now cost effectively protect security ...

Linux/Unix, Debian 6 | 32-bit Amazon Machine Image (AMI)



### Vyatta Virtual Router/Firewall/VPN

★★★★★ (1) | Version VSE6.6R2 | Sold by [Vyatta Inc.](#)
**\$0.30 to \$1.20/hr** for software + AWS usage fees

The Brocade Vyatta vRouter delivers advanced routing, firewall and VPN in a cloud-ready, software appliance. Much more than a simple gateway or firewall solution, Vyatta ...

Linux/Unix, Other 6.5R1 | 64-bit Amazon Machine Image (AMI)



### Check Point Virtual Appliance for AWS

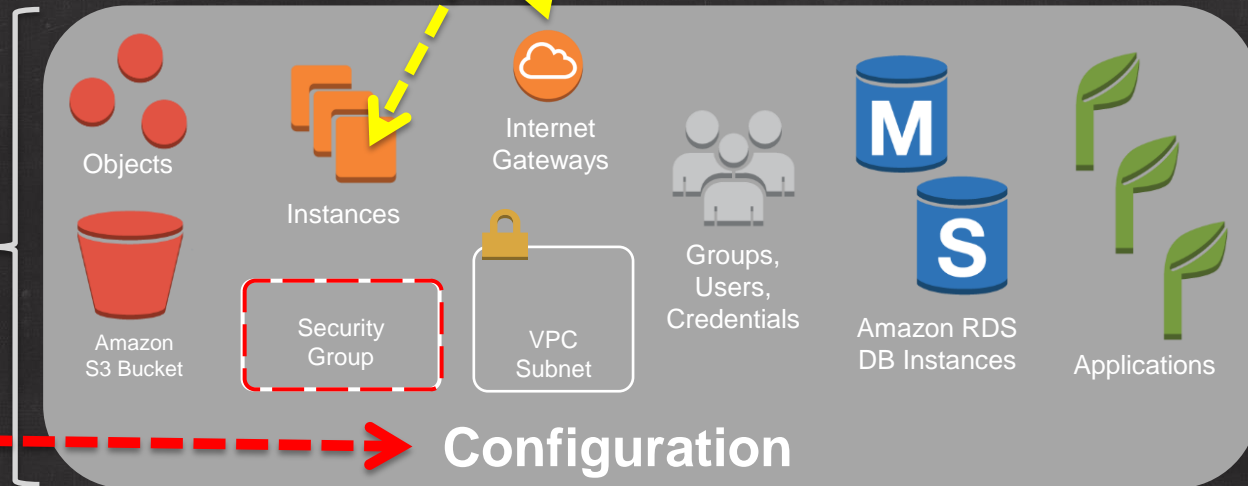
★★★★★ (2) | Version R75 | Sold by [Check Point Software Technologies](#)
**Bring Your Own License** + AWS usage fees

Check Point Virtual Appliance for Amazon Web Services enables customers to extend their security to the cloud with the full range of protections using Check Point Software





## Traditional IDS



Customer

This Talk



Configuration

AWS



Amazon S3



Amazon EC2



Amazon VPC



IAM



Amazon RDS



Elastic Beanstalk

# School of r00t

- Gain access
- Maintain access
- Steal stuff

# Prerequisites

- AWS Identity and Access Management (IAM)  
<http://docs.aws.amazon.com/IAM/latest/UserGuide/IAMGettingStarted.html>
- Multi-Factor Authentication (MFA)  
[http://docs.aws.amazon.com/IAM/latest/UserGuide/Using\\_ManagingMFA.html](http://docs.aws.amazon.com/IAM/latest/UserGuide/Using_ManagingMFA.html)
- Amazon S3 Bucket Logging  
<http://docs.aws.amazon.com/AmazonS3/latest/UG/ManagingBucketLogging.html>
- And THREE more ...



# Security Role

- You need insight when managing the security of many AWS accounts
- Create a “security audit role” with “read” access to policies and configurations you want to monitor.
- For more info or getting started, check out <http://docs.aws.amazon.com/IAM/latest/UserGuide/WorkingWithRoles.html>

# What's a Role

- Named IAM entity (name isn't a secret)
- Set of permissions
- No credentials: Policy specifies who can assume

[illegible]



# Security Role (Snippet of Example Policy)

```
{
  "sid": "stmt1382474270211",
  "Action": [
    "s3:GetBucketAcl",
    "s3:GetBucketLocation",
    "s3:GetBucketLogging",
    "s3:GetBucketNotification",
    "s3:GetBucketPolicy",
    "s3:GetLifecycleConfiguration",
    "s3:GetObjectAcl",
    "s3:GetObjectVersionAcl",
    "s3:ListAllMyBuckets"
  ],
  "Effect": "Allow",
  "Resource": "*"
}

"s3:GetBucketRequestPayment",
"s3:GetBucketVersioning",
"s3:GetBucketWebsite",
```

# Demonstration: Creating Security Role

Services

Edit

Greg Roth

Global

Help

Dashboard

Details

Groups

Users

Roles

Identity Providers

Password Policy

Create New Role

Role Actions

Viewing:

X

K < 1 to 6 of 6 Items > |

Role Name	Creation Time
<input type="checkbox"/> bototest	2013-11-14 00:17 PST
<input type="checkbox"/> ClassicRTTRole	2013-06-25 11:28 PDT
<input type="checkbox"/> jenkins	
<input type="checkbox"/> ltest	
<input type="checkbox"/> security_a	

0 Roles Selected

Select a role

Create Role

Cancel

CONFIGURE ROLE

ESTABLISH TRUST

SET PERMISSIONS

REVIEW

Specify a role name. You cannot edit the role name after the role is created.

Role Name:

Maximum 64 characters. Use alphanumeric and '+=, @-.' characters

Continue

© 2008 - 2013, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Terms of Use

Feedback





Services ▾

Edit ▾

Greg Roth ▾

Global ▾

Help ▾

Dashboard

Details

Groups

Users

**Roles**

Identity Providers

Password Policy

Create New Role

Role Actions ▾

Viewing: 🔍

1 to 6 of 6 items

Create Role

Cancel ✕

CONFIGURE ROLE

ESTABLISH TRUST

SET PERMISSIONS

REVIEW

## Select Role Type

☐ AWS Service Roles

☒ Role for Cross-Account Access

› Provide access between AWS accounts you own

Allows IAM users from one of your other AWS accounts to access this account.

Select

› Allows IAM users from a 3rd party AWS account to access this account.

Allows IAM users from a 3rd party AWS account to access this account.

Select

☐ Role for Identity Provider Access



Services ▾

Edit ▾

Greg Roth ▾

Global ▾

Help ▾

Dashboard

Details

Groups

Users

**Roles**

Identity Providers

Password Policy

Create New Role

Role Actions ▾

Viewing: 🔍



⏪ < 1 to 6 of 6 Items > ⏩

Role Name	Creation Time
<input type="checkbox"/> bototest	2013-11-14 00:17 PST
<input type="checkbox"/> ClassicRTTRole	2013-06-25 11:29 PDT
<input type="checkbox"/> jenkins	
<input type="checkbox"/> ltest	
<input type="checkbox"/> security_aud	

0 Roles Selected

Select a role above

**Create Role** Cancel

Enter the ID of the AWS account whose IAM users will be able to access this account.

Account ID:

[< Back](#) [Continue](#)



Services ▾

Edit ▾

Greg Roth ▾

Global ▾

Help ▾

Dashboard

Details

Groups

Users

Roles

Identity Providers

Password Policy

Create New Role

Role Actions ▾



1 to 6 of 6 items

### Create Role

Cancel ✕

CONFIGURE ROLE

ESTABLISH TRUST

SET PERMISSIONS

REVIEW

## Set Permissions

You can customize permissions by editing the following policy document. For more information about the access policy language, see [Overview of Policies](#) in Using IAM.

### Policy Name

secaudit

### Policy Document

```
{
  "Statement": [
    {
      "Sid": "Stmnt1382473313140",
      "Action": [
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStackResource",
        "cloudformation:DescribeStackResources",
        "cloudformation:DescribeStacks",
        "cloudformation:GetTemplate",
        "cloudformation:ListStacks",
      ]
    }
  ]
}
```

Back

Continue



# Write-Once Storage

- What is it good for
  - Tripwire
  - Configuration audits
  - Logs
- Integrity for records of activity, historical configurations
- Further enhanced by moving off-system or limiting availability to a VERY select few

# Configuring Write-Once Storage

- Bucket versioning

<http://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html>

- MFA delete

<http://docs.aws.amazon.com/AmazonS3/latest/dev/MultiFactorAuthenticationDelete.html>

- Go for the gusto! Create a SECOND account
  - Bucket policy
  - Role

## Bucket Policy Editor

Cancel X

### Policy for Bucket : "writeonce"

Add a [new policy](#) or edit an existing bucket policy in the text area below.

```
{
  "Version": "2008-10-17",
  "Id": "Policy1382581126724",
  "Statement": [
    {
      "Sid": "Stmt1382581121416",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::923022406781:root"
      },
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::writeonce/*"
    }
  ]
}
```

[AWS Policy Generator](#) | [Sample Bucket Policies](#)

Save

Delete

Close



## ▼ Versioning

**Versioning** allows you to preserve, retrieve, and restore every version of every object stored in this bucket. This provides an additional level of protection by providing a means of recovery for accidental overwrites or deletions.

Once enabled, Versioning cannot be disabled.

☒ **Enabled**    ☐ **Suspended**

**Save**

**Cancel**

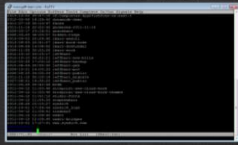
# Audit Logs via AWS CloudTrail



AWS Management Console



AWS SDKs



AWS CLI

- AWS CloudTrail records API calls in your account and delivers logs to your S3 bucket.
- Typically, delivers an event within 15 minutes of the API call.
- Log files are delivered approximately every 5 minutes.
- Currently in us-east-1 and us-west-2

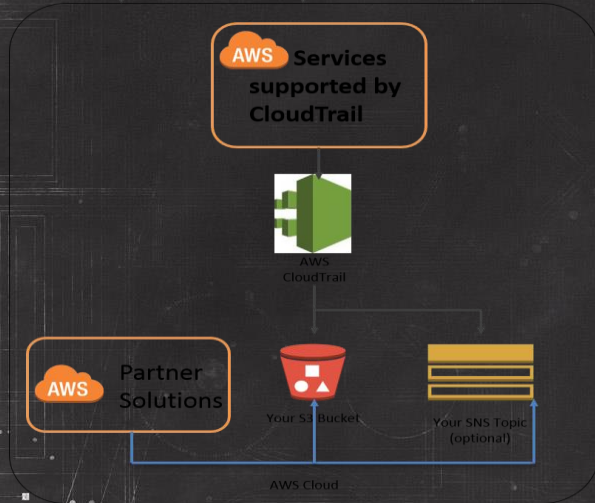


Image Source: Jeff Barr

# AWS Services Supported by AWS CloudTrail

- Currently, records API call made to these AWS services.



Amazon EC2



Amazon Redshift



AWS IAM



Amazon EBS



Amazon VPC



AWS STS (Security Token Service)



Amazon RDS

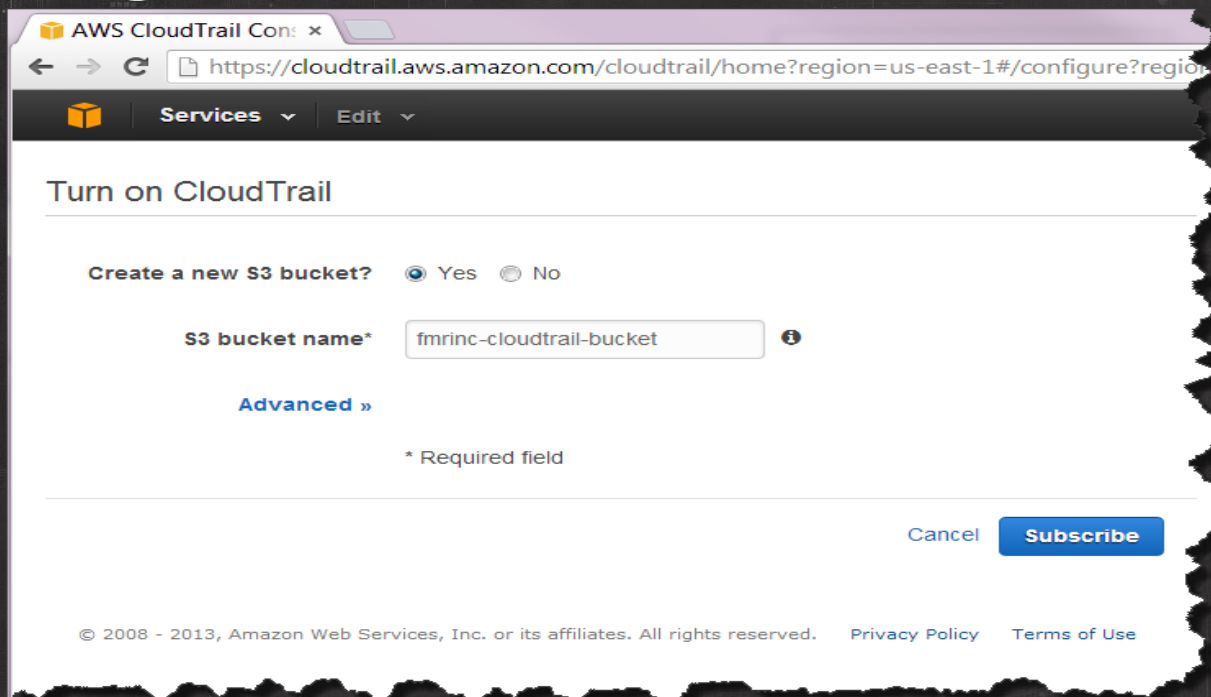


AWS CloudTrail

- Includes API calls made by higher-level AWS services such as AWS CloudFormation, AWS Elastic Beanstalk and AWS OpsWorks



# Turning on AWS CloudTrail



The screenshot shows the AWS CloudTrail console interface. At the top, the browser address bar displays the URL: `https://cloudtrail.aws.amazon.com/cloudtrail/home?region=us-east-1#/configure?region=us-east-1`. Below the address bar, there's a navigation bar with the AWS logo, a 'Services' dropdown menu, and an 'Edit' dropdown menu. The main heading is 'Turn on CloudTrail'. Under this heading, there's a section 'Create a new S3 bucket?' with two radio buttons: 'Yes' (selected) and 'No'. Below this, there's a text input field labeled 'S3 bucket name\*' containing the text 'fmrinc-cloudtrail-bucket'. To the right of the input field is an information icon. Below the input field, there's a link 'Advanced »'. At the bottom of the form, there's a note '\* Required field'. On the right side of the form, there are two buttons: 'Cancel' and 'Subscribe'. At the very bottom of the page, there's a footer with copyright information: '© 2008 - 2013, Amazon Web Services, Inc. or its affiliates. All rights reserved.' and links to 'Privacy Policy' and 'Terms of Use'.

- Have a centralized write-only store? Use it!

# What is in the logs?

- **Who** made the API call?
- **When** was the API call made?
- **What** was the API call?
- **What** were the resources that were acted up on in the API call?
- **Where** was the API call made from?

# Who? Example 1: API Call by IAM User Bob

```
"userIdentity": {  
    "accessKeyId": "AKEXAMPLE123EJVA",  
    "accountId": "123456789012",  
    "arn": "arn:aws:iam::123456789012:user/Bob",  
    "principalId": "AIEXAMPLE987ZKLALD3HS",  
    "type": "IAMUser",  
    "userName": "Bob"  
}
```

Anonymized data



# Who? Example 2: API Call by Federated User Alice

```
"userIdentity":{  
  "type":"FederatedUser",  
  "principalId":"123456789012:Alice",  
  "arn":"arn:aws:sts::123456789012:federated-user/Alice",  
  "accountId":"123456789012",  
  "accessKeyId":"ASEXAMPLE1234WTROX8F",  
  "sessionIssuer":{  
    "type":"IAMUser",  
    "accountId":"123456789012",  
    "userName":"Bob"  
  }  
}
```

Anonymized data; Partial Output

# Who? Example 3: AWS Service Creating Resource, on Behalf of a User

- Elastic Beanstalk creating AWS resources on behalf of IAM user Bob

```
"userIdentity": {  
  "accountId": "123456789012",  
  "arn": "arn:aws:iam::123456789012:user/Bob",  
  "invokedBy": "elasticbeanstalk.amazonaws.com",  
  "principalId": " ASEXAMPLE123XWTROX8F ",  
  "type": "IAMUser",  
  "userName": "Bob"  
}
```

Anonymized data

# When was the API call made?

- Start time and date of the event in ISO 8601 format.
- Unambiguous and well-defined method of representing date and time.
- AWS services sync all system clocks with centralized Network Time Protocol (NTP) servers

"eventTime": "2013-10-23T23:30:42Z"



# What was the API call?

## What resources were acted up on?

- API call and the service the API call belongs to.  
"eventName": "RunInstances"  
"eventSource": "EC2"
- Request parameters provided by the requester and Response elements returned by the AWS service.
- Response elements for read only API calls (Describe\*, Get\*, List\*) are not recorded to prevent event size inflation.

# School of r00t

- Gain access
- Maintain access
- Steal stuff

# Detecting Unauthorized Access

- Types of access
  - Credentials
  - Publicly accessible resources
  - Cross account access



# Detecting Unauthorized Access – Credentials

- Types of credentials
  - Login profile
  - Access key
  - X509
  - Cloudfront
  - Temporary Security Credentials
- Attachment points
  - Root account
  - IAM users
- You want to know what credentials are out there with access to your account.

# Demonstration: Checking Credentials

[Dashboard](#)[Details](#)[Groups](#)[Users](#)[Roles](#)[Identity Providers](#)[Password Policy](#)[Create New Users](#)[User Actions ▾](#)

Viewing:



1 to 6 of 6 items



User Name	Groups	Password	Access Keys	Creation Time
<input type="checkbox"/> ClassicRTTUser	1		2 active	2013-06-25 11:28 PDT
<input checked="" type="checkbox"/> ec2test	0		2 active	2013-06-14 13:41 PDT

1 Users Selected



User: ec2test

[Groups](#)[Permissions](#)[Security Credentials](#)[Summary](#)

### Access Credentials

#### Access Keys:

AKIAISNKP5NBWJRQTBWA

Active

2013-06-14 13:41 PDT

AKIAMWFQHOLKE3ARKOQ

Active

2013-10-29 10:54 PDT

[Manage Access Keys](#)

#### Signing Certificates:

None

[Manage Signing Certificates](#)

### Sign-In Credentials

#### User Name:

ec2test

#### Password:

Yes

[Manage Password](#)

#### Multi-Factor Authentication Device:

No

[Manage MFA Device](#)



# Detecting Unauthorized Access – Public

- Publically accessible resources (NOT by default, but could be configured as such)
  - Amazon S3 Bucket
  - Amazon S3 Anonymous Objects
  - Amazon SQS Open / Public Queues
- You want to keep track of which resources are readable (or writable even) to the world

# Detecting Unauthorized Access – Cross Account

- Resources that support resource policies
  - Amazon S3 Buckets
  - Amazon SQS queues
  - Amazon SNS topics
- You want to pay particular attention to any resources that have resource policies allowing cross account access.

# Demonstration: Checking for Cross-Account Access to Resources





Upload

Create Folder

Actions ▾

None

Properties

Transfers



All Buckets / gbr-billreport

Name	Storage Class	Size	Last Modified
Bucket Policy Editor			21:48 GMT-800
Policy for Bucket : "gbr-billreport"			29:25 GMT-800
Add a new policy or edit an existing bucket policy in the text area below.			2:54:27 GMT-800
			28:43 GMT-800
			00:08:33 GMT-700

Cancel

```
{
  "Version": "2008-10-17",
  "Id": "Policy1335892530063",
  "Statement": [
    {
      "Sid": "Stmt1335892150622",
      "Effect": "Allow",
      "Principal": {
        "AWS":
"arn:aws:iam::386209384616:root"
      },
      "Action": [
        "s3:GetBucketAcl",
        "s3:GetBucketPolicy"
      ],
      "Resource": "arn:aws:s3:::gbr-
billreport"
    },
    {
      "Sid": "Stmt1335892526596",
      "Effect": "Allow"
```

[AWS Policy Generator](#) | [Sample Bucket Policies](#) Save Delete Close

## Bucket: gbr-billreport

Bucket: gbr-billreport  
Region: US Standard  
Creation Date: Tue Oct 22 17:21:53 GMT-700 2013  
Owner: Me

## Permissions

Grantee: gregrataws

☒ List ☒ Upload/Delete ☒ View Permissions ☒ x

Edit Permissions

+ Add more permissions

Edit bucket policy

Add CORS Configuration

Save

Cancel

▸ Static Website Hosting

▸ Logging

▸ Notifications

▸ Lifecycle

# Detecting Unauthorized Access – Roles

- What is a role
  - Name
  - AssumeRole Policy
  - Capabilites
- You want to look at what roles are present in the account and who can assume them

# Demonstration: Checking for Roles



[Dashboard](#)[Details](#)[Groups](#)[Users](#)[Roles](#)[Identity Providers](#)[Password Policy](#)[Create New Role](#)[Role Actions ▾](#)

Viewing:



&lt; &lt; 1 to 6 of 6 items &gt; &gt;

Role Name	Creation Time
<input type="checkbox"/> bototest	2013-11-14 00:17 PST
<input type="checkbox"/> ClassicRTTRole	2013-06-25 11:29 PDT
<input type="checkbox"/> jenkins	2013-03-29 18:01 PDT
<input type="checkbox"/> ltest	2013-03-28 15:19 PDT
<input checked="" type="checkbox"/> security_audit	2013-11-06 19:48 PST

1 Roles Selected



Role: security\_audit

[Permissions](#)[Trust Relationships](#)[Summary](#)

You can view the trusted entities that can assume the role and the access conditions for the role. [Show policy document](#)

[Edit Trust Relationship](#)

### Trusted Entities

The following trusted entities can assume this role.

#### Trusted Entities

The account 923022406781

### Conditions

The following conditions define how and when trusted entities can assume the role.

There are no conditions associated with this role.

# Detecting Unauthorized Access – Effective Access

- Ways of expressing \* (IMPLICIT \*)
  - PutUserPolicy
  - Credential creation
  - PassRole \*
- You want to look out for policies that could be used to GAIN all access (IAM APIs)
- IAM Policy Simulator ...

<https://policysim.aws.amazon.com/>

# A

# B

# C

```
{
  "Statement": [
    {
      "Sid":
"Stmt1383555181147",
      "Action": "sns:*",
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Sid":
"Stmt1383555193395",
      "Action": ["s3:*", "*"],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

```
{
  "Statement": [
    {
      "Sid":
"Stmt1383555181147",
      "NotAction": "*",
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Sid":
"Stmt1383555193395",
      "Action":
["iam:PutUserPolicy"],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

```
{
  "Statement": [
    {
      "Sid":
"Stmt1383555181147",
      "Action": "ec2:*",
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Sid":
"Stmt1383555193395",
      "Action":
["s3:*", "iam:PassRole"],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```



**YES!**

# A

# B

# C

```
{
  "Statement": [
    {
      "Sid":
"Stmt1383555181147",
      "Action": "sns:*",
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Sid":
"Stmt1383555193395",
      "Action": ["s3:*", "*"],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

```
{
  "Statement": [
    {
      "Sid":
"Stmt1383555181147",
      "NotAction": "*",
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Sid":
"Stmt1383555193395",
      "Action":
["iam:PutUserPolicy"],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

```
{
  "Statement": [
    {
      "Sid":
"Stmt1383555181147",
      "Action": "ec2:*",
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Sid":
"Stmt1383555193395",
      "Action":
["s3:*", "iam:PassRole"],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

# Detecting Unauthorized Access – Effective Access

- Dump the output of various configuration APIs into write-once storage
- Pay attention to changes
- Some examples for grabbing this data ...

<https://s3.amazonaws.com/reinvent2013-sec402/SecConfig.py>



# Using Security Role for Amazon S3 Audit (Bucket Policies)

```
s3 = boto.connect_s3(access_key_id,secret_access_key)
bucket_info=[]
buckets=s3.get_all_buckets()
for bucket in buckets:
    try:
        policy=bucket.get_policy()
        bucket_info.append(config_line_policy("s3:bucketpolicy",bucket.name,"",policy))
    except boto.exception.S3ResponseError as e:
        bucket_info.append(config_line("s3:bucketpolicy",bucket.name,"",e.code))
output_lines(bucket_info)
```

<https://s3.amazonaws.com/reinvent2013-sec402/SecConfig.py>

# Using Security Role for IAM Audit (Users)

```
user_info=[]
users=iam.get_all_users().list_users_response.list_users_result.users
debug(users)
for user in users:
    policies=iam.get_all_user_policies(user.user_name)
    policies=policies.list_user_policies_response.list_user_policies_result.policy_names
    for policy_name in policies:
        policy=iam.get_user_policy(user.user_name, policy_name)\
            .get_user_policy_response.get_user_policy_result.policy_document
        policy=urllib.unquote(policy)
        user_info.append(config_line_policy("iam:userpolicy", user.user_name,
        policy_name, policy))
output_lines(user_info)
```

<https://s3.amazonaws.com/reinvent2013-sec402/SecConfig.py>

# Account Configuration Change Security Alerts

- Dump all the users, groups, roles, attached permissions, creds for all users
- Amazon S3 bucket, Amazon SQS queue, Amazon SNS topic policies
- Amazon EC2 security group configuration
- All goes to flat file, write-once Amazon S3 object
- Diff and detect changes

<https://s3.amazonaws.com/reinvent2013-sec402/SecConfig.py>



# Demonstration: Intrusion Detection Script

<https://s3.amazonaws.com/reinvent2013-sec402/SecConfig.py>

# Example Usage

```
SecConfig.py [-h] -a ACCESS_KEY_ID -k SECRET_ACCESS_KEY \  
[-t SECURITY_TOKEN] [-r ROLE] [-v] [-d]
```

-h, --help	show this help message and exit
-a ACCESS_KEY_ID, --access_key_id ACCESS_KEY_ID	access key id
-k SECRET_ACCESS_KEY, --secret_access_key SECRET_ACCESS_KEY	secret access key
-t SECURITY_TOKEN, --security_token SECURITY_TOKEN	security token (for use with temporary security credentials)
-r ROLE, --role ROLE	role to assume
-v, --verbose	enable verbose mode
-d, --debug	enable debug mode

```
iam:accountsummary, AccountMFAEnabled, 1
iam:accesskey, ClassicRTTUser, Active, AKIAJQF4G2Z0ZBL3FYKQ
iam:accesskey, ClassicRTTUser, Active, AKIAJVZ456L2HVERGIQ
iam:accesskey, audit, Active, AKIAJJ7D5VQ2KAC4RX6Q
iam:accesskey, ec2test, Active, AKIAIWMFQHOLKE3ARKQQ
iam:accesskey, ec2test, Active, AKIAISNKP5NBWJRQTPBWA
iam:usingergroup, ClassicRTTUser, , ClassicRTTGrp
iam:userpolicy, ClassicRTTUser, PowerUserAccess-ClassicRTTUser-201306251128,
3be1369a6334b59ecbe24496a45a6c792ae8468bf29f31d30f5deffc645b2197
iam:userpolicy, audit, ReadOnlyAccess-audit-201310221803,
02bc4680f269c2949a2da250e6c2b430e3f2a6c1f9e665fce58b6d94de27001d
iam:userpolicy, ec2test, AdministratorAccess-ec2test-201306141348,
08504c15956913f7a75aadcb895ef2b92368826916f95027a128388e60cda61d4
iam:userpolicy, ec2test, AdministratorAccess-ec2test-201306141416,
76c7d1e7027c934815dd4c69bd072992cd2912af59a513ddc633223b7fe01ebb
iam:userpolicy, ec2test, ReadOnlyAccess-ec2test-201310231957,
02bc4680f269c2949a2da250e6c2b430e3f2a6c1f9e665fce58b6d94de27001d
iam:userpolicy, mbp-r-managed, one,
e3e0211e865b5cac2a57241edcb8aeb9d546764abba2f325b694ec840985c2ff
iam:userpolicy, quux, mypolicy,
2ad665ca145f5d107be53beecc7c0092461d76c1b9588cae4e0b0f4cbdbdc5083
iam:grouppolicy, test, CloudFrontFullAccess-test-201310291053,
3036fb93022a9f4146d6ccc67ff953d2be25c5ae3d0241b8b983442b577e5b73
iam:assumerolepolicy, ClassicRTTRole,
iam:aws:iam:923022406781:role/ClassicRTTRole,
3036fb93022a9f4146d6ccc67ff953d2be25c5ae3d0241b8b983442b577e5b73
iam:assumerolepolicy, jenkins, arn:aws:iam:923022406781:role/jenkins,
e3e0211e865b5cac2a57241edcb8aeb9d546764abba2f325b694ec840985c2ff
iam:assumerolepolicy, ltest, arn:aws:iam:923022406781:role/ltest,
6e676d8b13e140781b56775c5e2894d8b8b38e15a12b64bf128a9794931b80
iam:assumerolepolicy, security_audit,
arn:aws:iam:923022406781:role/security_audit,
6e676d8b13e140781b56775c5e2894d8b8b38e15a12b64bf128a9794931b80
```

```
iam:assumerolepolicy,uasrc,arn:aws:iam::923022406781:role/uasrc,
b675543c022ca9bce21414468a7b6e2e07116f1f77e722ae2f65fed7e69ffbb
iam:rolepolicy,ClassiCRTRole,PowerUserAccess-ClassicRTRole-201306251129,
e3e0211e865b5cac2a57241edcb8aeb9d546764abba2f325b694ec840985c2ff
iam:rolepolicy,jenkins,ReadOnlyAccess-jenkins-201303291802,
6e676d8b13e140781b56775c55e2894d8b8b83be15a12b64bf128a9794931b80
iam:rolepolicy,security_audit,ReadOnlyAccess-security_audit-201311061949,
b675543c022ca9bce21414468a7b6e2e07116f1f77e722ae2f65fed7e69ffbb
iam:rolepolicy,uasrc,AmazonDynamoDBFullAccess-uasrc-201210111714,
75cc727843ed2bc783bf9c325300ff307d9b2594b2a5d3d88b59e609e39af1a89
s3:bucketpolicy,caec.us, , NoSuchBucketPolicy
s3:bucketpolicy,cf-templates-g5zg6nnc0317-us-east-1, , NoSuchBucketPolicy
s3:bucketpolicy,dcslides, , NoSuchBucketPolicy
s3:bucketpolicy,elasticbeanstalk-us-east-1-923022406781, , NoSuchBucketPolicy
s3:bucketpolicy,gb-r-billreport, , NoSuchBucketPolicy
s3:bucketpolicy, , eef9f053353a1c6bb3f7bec968d6851679e9694757c8f8e18ae3588e7334e2a20
s3:bucketpolicy,gb-r-testv, , NoSuchBucketPolicy
s3:bucketpolicy,gbrcrypto, , NoSuchBucketPolicy
s3:bucketpolicy,gbrcrypto-logs, , NoSuchBucketPolicy
s3:bucketpolicy,gregroth.desktop.amazon.com, , NoSuchBucketPolicy
s3:bucketpolicy,logs.s3.caec.us, , NoSuchBucketPolicy
s3:bucketpolicy,s3.caec.us, , NoSuchBucketPolicy
sqs:queuepolicy,https://queue.amazonaws.com/923022406781/deletemetoo, , NoPolicy
sqs:queuepolicy,https://queue.amazonaws.com/923022406781/deletetme, ,
21fbfa969788e8675e540c1fb0114f1a5d20863d5c4e4e976c106af8bffc9
sns:topicpolicy,arn:aws:sns:us-east-1:923022406781:test,
c5f96939702f70124b7e2af14ed07034d155fa56bf043f187d5d6d2d1c9521c0
sns:topicpolicy,arn:aws:sns:us-east-1:923022406781:test2,
27f459b59b384b38c92458a4c2ea7268be7c73db687cfba52ac7521770541cb8
```



# Example Output (Snippet)

```
iam:accountsummary, AccountMFAEnabled, , 1
iam:accesskey, ClassicRTTUser, Active, AKIAJQF4G2Z0ZBL3FYKQ
iam:accesskey, ClassicRTTUser, Active, AKIAJVZ456L2HVERGIQ
iam:accesskey, audit, Active, AKIAJJ7D5VQ2KAC4RX6Q
iam:accesskey, ec2test, Active, AKIAIMWFQHOLKE3ARKOQ
iam:accesskey, ec2test, Active, AKIAISNKP5NBWJRQTBWA
iam:accesskey, mbp-r-managed, Active, AKIAJKVVGIG7L5UC50GQ
iam:accesskey, quux, Active, AKIAJR7ZICS26032EPBQ
iam:accesskey, test, Active, AKIAINTUMS4ITD5CJVSA
iam:useringroup, ClassicRTTUser, , ClassicRTTGrp
```

# Example Output (Snippet)

```
s3:bucketpolicy, dcslides, , NoSuchBucketPolicy  
s3:bucketpolicy, elasticbeanstalk-us-east-1-923022406781, ,  
NoSuchBucketPolicy  
s3:bucketpolicy, gbr-billreport, ,  
ee9f053535a1c6bb3f7becc968d6851679e9694757c8fe18ae3588e7334e2a20
```

```
sqs:queuepolicy, https://queue.amazonaws.com/923022406781/deletemetoo, ,  
NoPolicy  
sqs:queuepolicy, https://queue.amazonaws.com/923022406781/deletme, ,  
21fbfa969788e8675e540c1fb0114f1a5d280863d5c4e4e9476ec106af8bffc9
```

```
sns:topicpolicy, arn:aws:sns:us-east-1:923022406781:test, ,  
c5f96939702f70124b7e2af14ed07034d155fa56bf043f187d5d6d2d1c9521c0  
sns:topicpolicy, arn:aws:sns:us-east-1:923022406781:test2, ,  
27f459b59b384b38c92458a4c2ea7268be7c73db687cfba52ac7521770541cb8
```

# Example Diff, Something to Look Into

```
< iam:userpolicy, mbp-r-managed, one,  
e3e0211e865b5cac2a57241edcb8aeb9d546764abba2f325b694ec840985c2ff  
---  
> iam:userpolicy, mbp-r-managed, ReadOnlyAccess-mbp-r-managed-  
201311111559,  
b675543c022ca9bce21414468a7b62e207116f11f77e722ae2f65fed7e69ffbb  
> iam:userpolicy, mbp-r-managed, one,  
1cc602178f7e876c6d38cbaa8c4adde19b1c3e5a89e6f13c29df5688eb73f50f
```

<https://s3.amazonaws.com/reinvent2013-sec402/SecConfig.py>



# School of r00t

- Gain access
- Maintain access
- Steal stuff

●●●●○ Verizon LTE

10:22 AM



< Messages

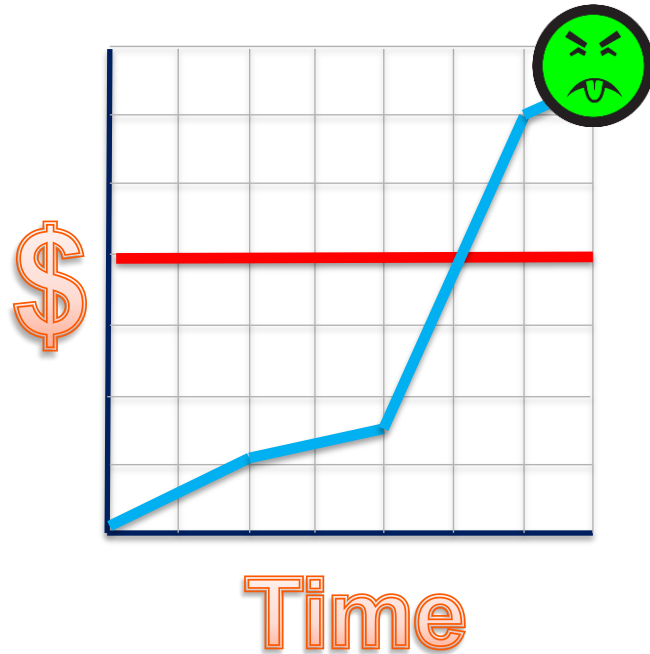
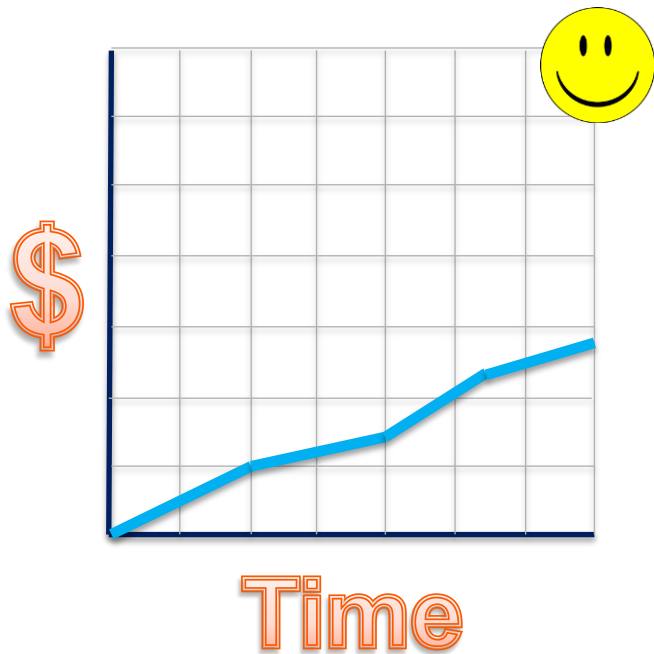
Verizon

Contact

Fri, Oct 4, 10:00 AM

DATA ALERT: Your acct  
used abt 75% of its data  
allowance for the bill  
ending the 25th. Monitor at  
[vzw.com](http://vzw.com). As of 10/04  
01:00 PM EDT. FREE MSG

# Example OK vs UH-OH Billing Trend / Graph





# Billing Alerts!

- No need to wait until end of month to become aware of unexpected utilization
  - Establish a baseline of known good billing over time; set your thresholds (overall or service specific)
  - Investigate alerts to determine root (?) cause
  - Simplest cloud IDS mechanism, and FREE\*
- \* Setup of 10 alarms and receipt of 1 K notifications

[Sign Up](#)[My Account / Console](#) ▼[English](#) ▼[AWS Products & Solutions](#) ▼[AWS Product Information](#) ▼[Developers](#) ▼[Support](#) ▼

## Account

### ■ Account Activity

- [AWS Identity and Access Management](#)
- [AWS Management Console](#)
- [Consolidated Billing](#)
- [DevPay](#)
- [Manage Your Account](#)
- [Payment Method](#)
- [Personal Information](#)

## Account Activity

**Welcome**[| Sign Out](#)

Account Number



You are eligible for the [AWS Free Usage Tier](#). See the [Getting Started Guide AWS Free Usage Tier](#) to learn how to get started with the free usage tier.



Monitor your estimated charges. [Enable Now](#) to begin setting billing alerts that automatically e-mail you when charges reach a threshold you define. [Learn More](#)

## 1. Select Metric

## 2. Define Alarm

Back

Next

Cancel

To create an alarm, first **select a metric** by browsing or searching on the right. Once you find the metric you want, select it and then click **Next**.

Browse Metrics

Search Metrics

X

## CloudWatch Metrics by Category

Your CloudWatch metric summary has loaded. Total metrics: **1,014**

### Billing Metrics : 35

Total Estimated Charge : 1

By Service : 13

By Linked Account : 3

By Linked Account and Service : 18

### DynamoDB Metrics : 4

Table Metrics : 4

### EC2 Metrics : 272

Per-Instance Metrics : 181

By Auto Scaling Group : 56

By Image (AMI) Id : 14

Aggregated by Instance Type : 14

Across All Instances : 7

### ELB Metrics : 95

Per-LB Metrics : 29

Per LB, per AZ Metrics : 37

By Availability Zone : 20

Across All LBs : 9



## 1. Select Metric

## 2. Define Alarm

Back

Next

Cancel

To create an alarm, first **select a metric** by browsing or searching on the right. Once you find the metric you want, select it and then click **Next**.

Browse Metrics

Search Metrics

X Billing > By Service

1 to 13 of 13 Metrics

Select All | Clear

Billing > By Service

	ServiceName	Currency	Metric Name
<input type="checkbox"/>	AWSDataTransfer	USD	EstimatedCharges
<input type="checkbox"/>	AWSQueueService	USD	EstimatedCharges
<input type="checkbox"/>	AWSSupportDeveloper	USD	EstimatedCharges
<input type="checkbox"/>	AmazonDynamoDB	USD	EstimatedCharges
<input checked="" type="checkbox"/>	AmazonEC2	USD	EstimatedCharges
<input type="checkbox"/>	AmazonElastiCache	USD	EstimatedCharges

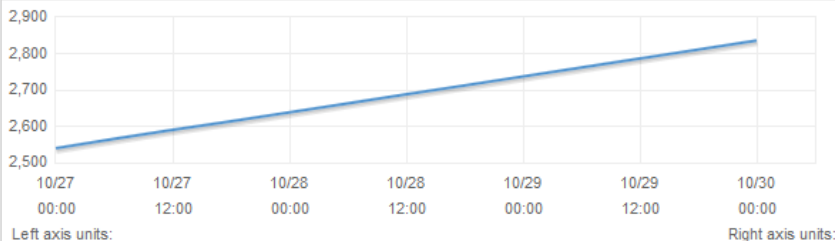
EstimatedCharges (None)

Maximum

6 Hours



Update Graph



Time Range

Relative

Absolute

UTC (GMT)

From: 3

days ago

To: 0

days ago

Zoom: 1h | 3h | 6h | 12h | 1d | 3d | 1w | 2w

1. Select Metric

2. Define Alarm

Back

Next

Cancel

Please set the alarm threshold, actions and click **Create Alarm** below.

Create Alarm

## Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name: My Estimated Charges

Description: Estimated Monthly Charges

Whenever charges for: EstimatedCharges

is:  $\geq$  USD \$ 200

for: 1 consecutive period(s)

## Actions

Define what actions are taken when your alarm changes state.

Notification

Delete

Whenever this alarm: State is ALARM

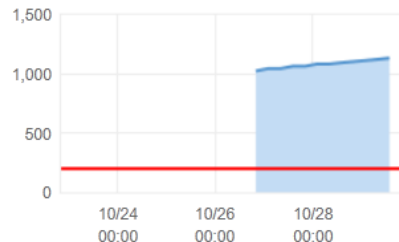
Send notification to: Please select an SNS topic

Create topic

## Alarm Preview

This alarm will trigger when the **blue line** goes up to or above the **red line** for a duration of **6 hours**

EstimatedCharges  $\geq$  200



Namespace: AWS/Billing

Currency: USD

Metric Name: EstimatedCharges

Period: 6 Hours

Statistic: Maximum

1. Select Metric

2. Define Alarm

Back

Next

Cancel

Please set the alarm threshold, actions and click **Create Alarm** below.

Create Alarm

## Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name: My Estimated Charges

Description: Estimated Monthly Charges

Whenever charges for: EstimatedCharges

is:  $\geq$  USD \$ 200

for: 1 consecutive period(s)

## Actions

Define what actions are taken when your alarm changes state.

Notification

Delete

Whenever this alarm: State is ALARM

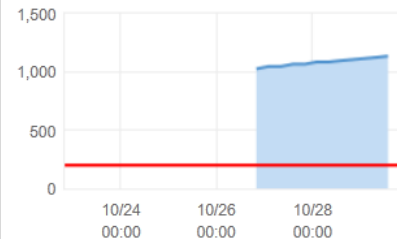
Send notification to:

Email list: john.stiles@example.com

## Alarm Preview

This alarm will trigger when the **blue line** goes up to or above the **red line** for a duration of **6 hours**

EstimatedCharges  $\geq$  200



Namespace: AWS/Billing

Currency: USD

Metric Name: EstimatedCharges

Period: 6 Hours

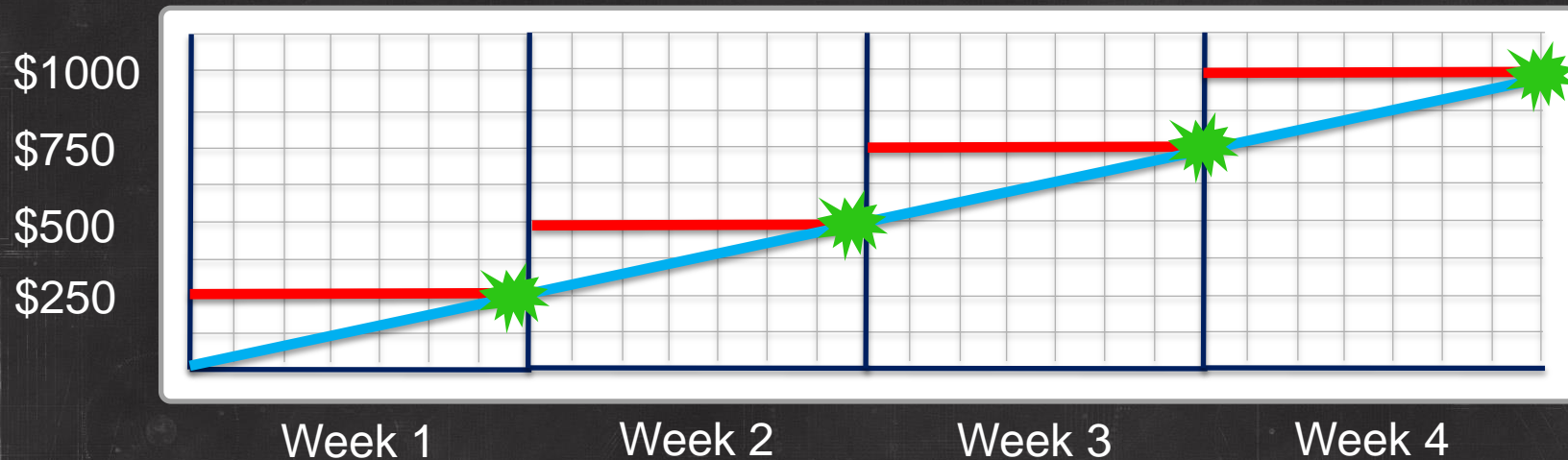
Statistic: Maximum




# Example Billing Alert via CLI

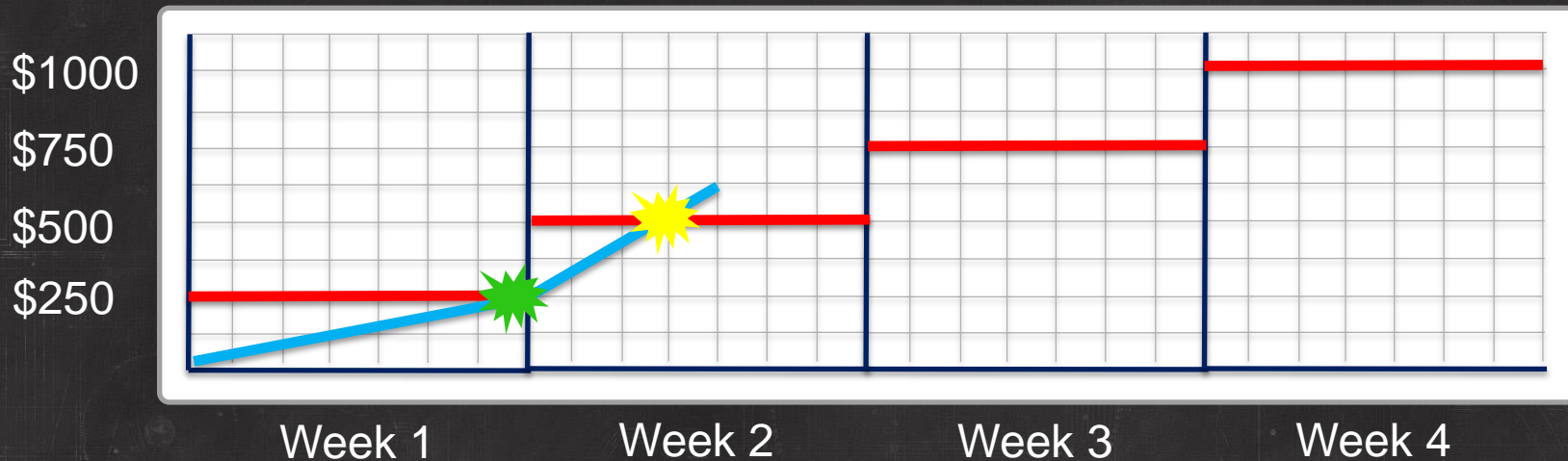
```
mon-put-metric-alarm ec2billing --comparison-operator  
GreaterThanOrEqualToThreshold --evaluation-periods 1 --metric-name  
EstimatedCharges --namespace AWS/Billing --dimensions "Currency=USD" --  
period 21600 --statistic Maximum --threshold 200 --actions-enabled true --  
alarm-actions arn:aws:sns:us-east-1:111111111111:NotifyMe
```

# Assuming You Anticipate ~ \$1 K / Month ...



 = OK!

# “Early” Alerts Are “Interesting” ...

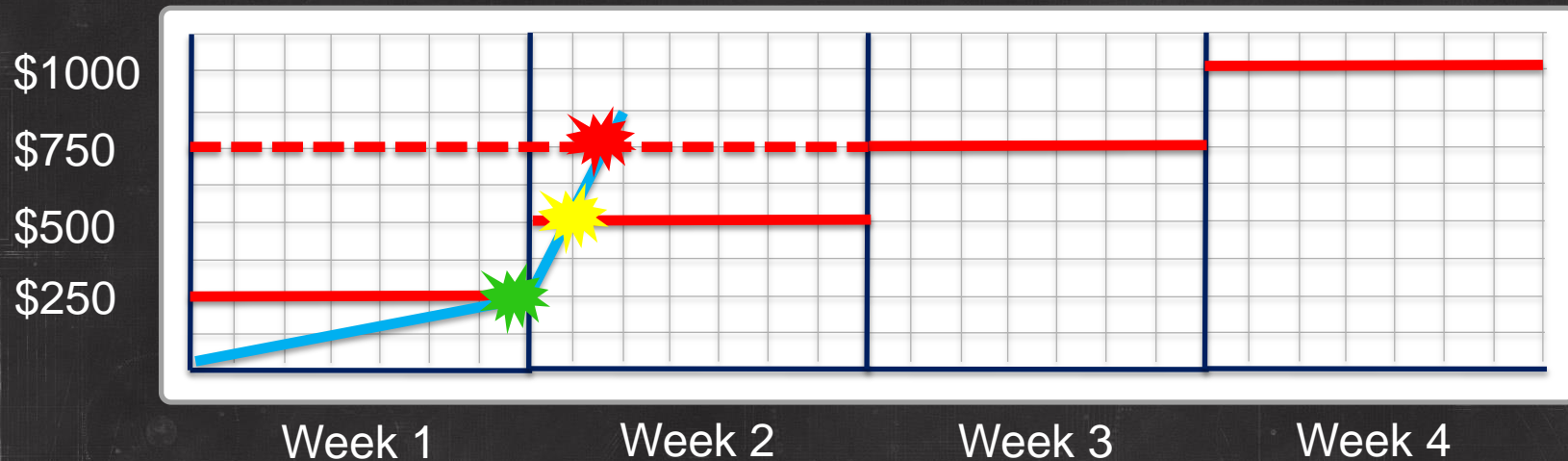


= OK!



= Hmm ...

# More than One “Early” Alert ...?



= OK!



= Hmm ...



= Uh-Oh!



# More Resources on Billing Alerts Setup ...

- Monitoring your AWS charges

[http://docs.amazonwebservices.com/AmazonCloudWatch/latest/DeveloperGuide/monitor\\_estimated\\_charges\\_with\\_cloudwatch.html](http://docs.amazonwebservices.com/AmazonCloudWatch/latest/DeveloperGuide/monitor_estimated_charges_with_cloudwatch.html)

- Amazon CloudWatch Command Line Interface Reference

<http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/CLIRreference.html>

# Create Your Own Meter-based Alerts?

- Use: programmatic access to billing data
- You have more info about the types and locations of charges
- Allows for looking for unexpected usage per region

<http://docs.aws.amazon.com/awsaccountbilling/latest/about/programmatic-access.html>

# Another Tactic? Rebuild Frequently

- Breaking in is noisy and the holes tend to get patched [intrusion lifecycle]
- Auditing a system is easiest after creation
- Rebuild everything every day



# Premium Support / Trusted Advisor

- Inspects AWS environment; can identify and help close security gaps, enable security features, examine permissions
  - Open security groups
  - Bucket policy
  - IAM, passwords, MFA

<https://aws.amazon.com/premiumsupport/trustedadvisor/>





No issue detected



Investigation Recommended



Action Recommended



Not Available

## Summary

## Cost Optimizing

## Security

## Fault Tolerance

## Performance

**\$1,821,868**

In potential annual savings



**Cost Optimizing**  
Suppressed (0)

**116**

Opportunities to enhance security



**Security**  
Suppressed (0)

**30**

Recommendations to improve availability



**Fault Tolerance**  
Suppressed (0)

**138**

Opportunities to improve performance



**Performance**  
Suppressed (0)

## Recently Launched Checks



**New**

Amazon Route 53 High TTL Resource Record Sets



0 of 1461 resource record sets have TTL values that are too large.



**New**

Amazon Route 53 Name Server Delegations



0 of 5 hosted zones do not have all four name server delegations configured.



**New**

Amazon Route 53 Alias Resource Record Sets



18 of 1491 resource record sets can be changed to alias resource record sets.

### Security Group - Open Ports

- ▶ 215 of 753 Security Group port rules create potential security vulnerabilities by granting global access

### IAM Use

- ▶ IAM is configured for this account

### S3 Bucket Permissions

- ▶ 6 of 42 S3 Buckets have permission properties that grant global access

### MFA On Root

- ▶ Root user does not have MFA enabled.

### IAM Password Policy

- ▶ Password policy is not configured

# Support for Security

- AWS support is the one-stop shop for AWS customers, for ANY concerns, including security-related
- If support can not immediately address your concern, they will escalate internally to the appropriate technical team, AWS security included

<https://aws.amazon.com/support>



# Other Resources

- AWS Security Blog  
<http://blogs.aws.amazon.com/security/>
- AWS Security Center  
<https://aws.amazon.com/security>
- Contact the AWS security team  
[aws-security@amazon.com](mailto:aws-security@amazon.com)

# NEW! Security Best Practices Whitepaper

- Help for designing security infrastructure and configuration for your AWS environment
- High-level guidance for ...
  - Managing accounts, users, groups roles
  - Managing OS-level access to instances
  - Securing your data, OS, apps, infrastructure
  - Managing security monitoring, auditing, alerting, incident response

[http://media.amazonwebservices.com/AWS\\_Security\\_Best\\_Practices.pdf](http://media.amazonwebservices.com/AWS_Security_Best_Practices.pdf)

# Key Takeaways

- Beyond traditional host- or network-based intrusion detection, there is intrusion detection for the cloud
- AWS provides a variety of mechanisms and support that you can and should leverage to monitor key security controls
- Tinker, give us feedback, and approach our partners about incorporating some ideas here



# Downloads

<https://s3.amazonaws.com/reinvent2013-sec402/secaudit.json>

<https://s3.amazonaws.com/reinvent2013-sec402/SecConfig.py>



# AWS re:Invent

Please give us your feedback on this presentation

## SEC402

As a thank you, we will select prize winners daily for completed surveys!

