Centralized AWS logging & Incident Response with Elastic

UCCSC 2018

8/15/2018

UC SANTA CRUZ

Team Members:

Troy Wright

Brian Hall

UCCSC 2018

UC SANTA CRUZ

Team Members:

Presentation Outline

- **Intro/Infrastructure**
- Logging pipeline
- Use cases
- End Results

Security Team Project Goals for AWS

Goals

- All AWS host & services logs sent to on-premise Elastic
- Maintain the same level of logging detail as on-premise hosts
- Provide same level of incident response in AWS as on-premise
- Provide meaningful log search to AWS developer team

UCSC Elastic Infrastructure

- 12 Elasticsearch nodes: Data store/Index/Cluster svcs
- 1 Kibana node: User interface, searching, reporting
- 11 Logstash nodes: (2 dedicated for AWS)
- 3 Kafka nodes: Log/Message queuing service.
- 900G-1.5T per day total logs for campus + AWS
- Data retention in Elastic for AWS data (90 days+)
- Virtual machines running from ESXi nodes
- Network attached storage

UCSC AWS Infrastructure

- Various enterprise level accounts - Identity Management (Shibboleth, LDAP, Grouper, Campus Directory) PeopleSoft Campus Solutions (9.0, GARP, Jazzee, PISA) Banner Advancement, ODS Dev (UCPath datastore)
- ~200 production hosts in AWS
- 100-130 log events / second (EPS) to our on-premise Elasticsearch cluster
- 15-30G logs / week (production hosts)
- 20G logs / week (vpc flow)

Terminology

- SIEM - *Security Information and Event Management system*- provides a comprehensive real-time view of your organizations IT security (heavily dependent on log collection!)
- CloudWatch - Provides centralized logging and event monitoring.
- CloudTrail - Provides governance, compliance, and operational auditing.
- ELB - *Elastic Load Balancing* - Automatically distributes incoming application traffic
- IAM - *Identity and Access Management* - create/manage AWS users, groups, permissions
- VPC - *Virtual Private Cloud* - Your own slice of the cloud (i.e., datacenter)
- EC2 Instance - *Elastic Compute Cloud* - A host/server in AWS
- S3 - *Simple Storage Service* - Provides data storage and retrieval

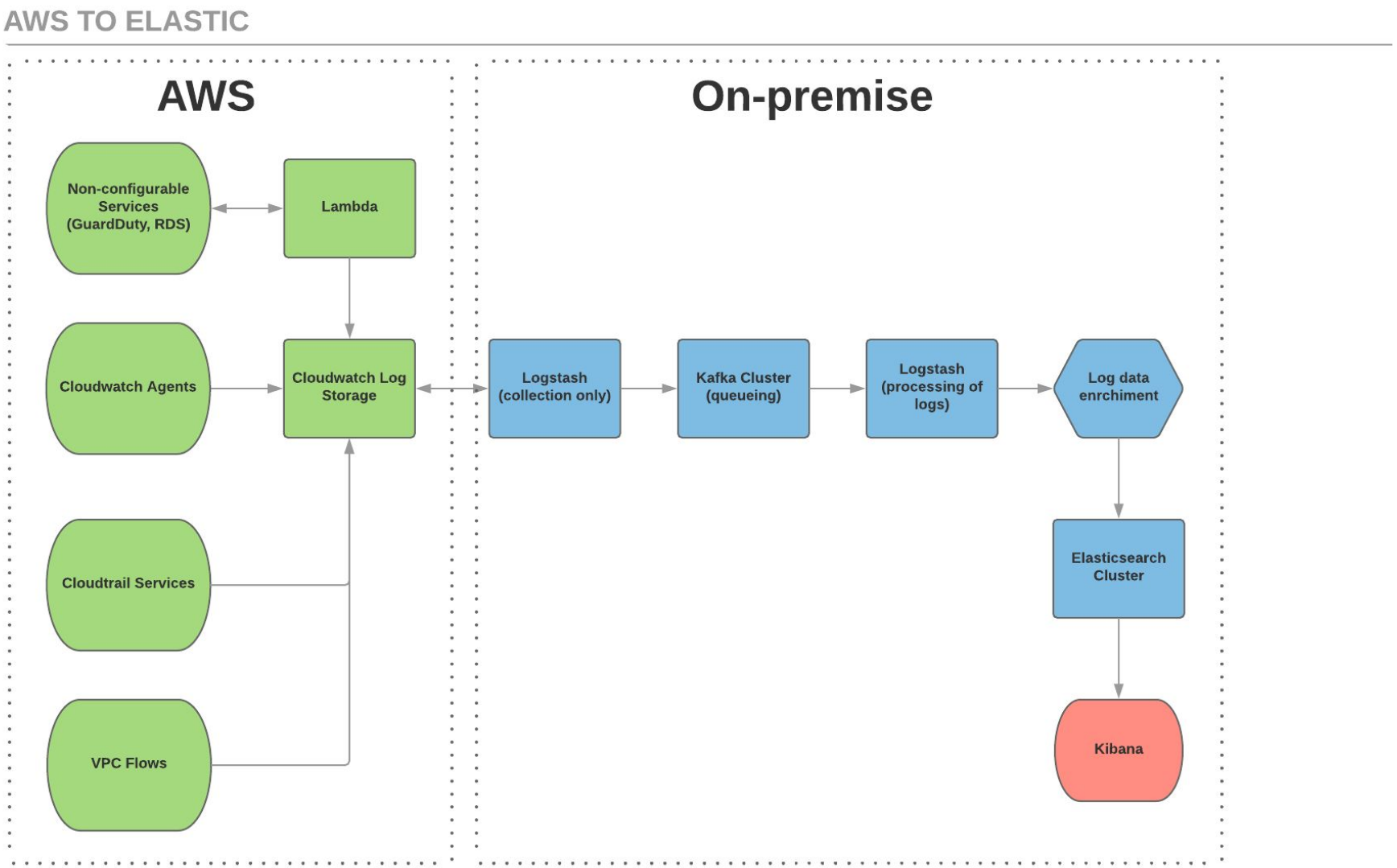
Presentation Outline

- Intro/Infrastructure
- **Logging pipeline**
- Use cases
- End Results

Our Approach
AWS to SIEM logging pipeline

- Get everything going into CloudWatch.
  - From there, pull all relevant CloudWatch data into our pipeline
- Some services in AWS can easily be configured to push log data to CloudWatch
  - Host and application logs
  - CloudTrail logs
  - VPC Flow logs
- Other services cannot be configured, so a custom solution is required
  - Examples include AWS GuardDuty, ELB, S3 logging
  - Logging for these AWS services handled through Lambda functions
  - Process requires specific roles and policies

AWS Logging Implementation



AWS Log Types

- CloudTrail
  - Captures API calls/user actions in the AWS account. Allows us to monitor who is doing what, when, and where.
- VPC Flow Logs
  - Captures network flow data within your VPC.
- ELB Access Logs
  - Detailed information about requests sent to your load balancer.
- Host (EC2 Instance) Logs
  - OS level logs: /var/log/messages, /var/log/secure, etc., and Application specific logs: /var/log/httpd/*
- S3 Logs
  - Storage access logs, ELB logs are in S3
- Threat Detection Logs: (alerts from AWS GuardDuty)

Tools for logging - CloudWatch Agents

- CloudWatch agent → your one-way ticket to getting EC2 Instance (host) logs to CloudWatch (os, application logs, etc.,)
- At UCSC, agent installed via configuration management tool (Chef Recipes)
  - We install this by default on all AWS hosts as part of our base packages deploy.
- You will need to create an IAM role/policy to create CloudWatch logs and assign that role to your EC2 Instance.
  - Amazon has straightforward documentation with an example you can copy paste for this.
- Edit the CloudWatch Agent config file to list logs you wish to capture.
  - This can also be automated using configuration management tools.

Tools for logging - AWS Lambda

- For applications and logs that don’t fit the CloudWatch mold, Lambda to the rescue!
- Available in a variety of programming languages
- Lets you run code without provisioning or managing servers. You pay only for the compute time you consume - there is no charge when your code is not running!
- Requires zero administration. Just upload your code and Lambda takes care of everything required to run and scale your code with high availability.

Lambda Details - what's needed

- Create an IAM role/policy to create CloudWatch logs and assign that role to your Lambda function.
- Create Lambda Function, add your code
  - Our functions were created using the Node.js runtime (default). You could also use other languages like Python, Java, or C#. *See our GitHub for the Lambda code.*
- Add a "trigger" to fire your Lambda function
  - In the Designer section of your Lambda function, you can pick from a set of predefined triggers.

Lambda Details - how it works

- Triggered via service:

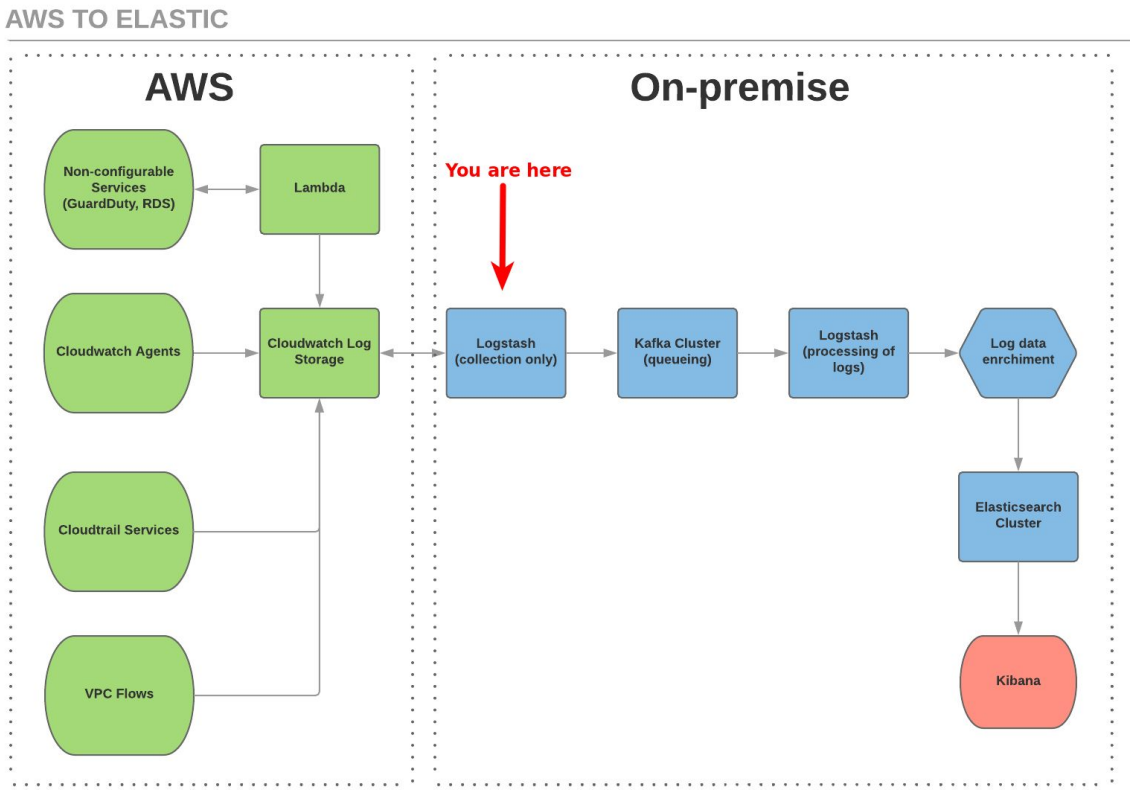
  Lambda can be directly triggered by various AWS services (S3, CloudWatch Events).

  Lambda is invoked *automatically* when events occur. For example,

  - To get S3 logs, Lambda function contains trigger to specific S3 bucket.
  - To get CloudWatch Event data (e.g., GuardDuty alerts), create a CloudWatch Event rule in the Rules section of CloudWatch. Point the rule to your Lambda function.
- Triggered via schedule:
  - You can schedule when you want your Lambda function to perform work.

Logstash

- Logstash plug-in for pulling AWS Cloudwatch logs
- Standardizing field names
- GeoIP when not natively available
- Lookup to map 12 digit AWS Account ID to AWS Account Name
- Threat intel lookups



Presentation Outline

- Intro/Infrastructure
- Logging pipeline
- **Use cases**
- End Results

Use Case: Incident Response

- AWS Security and Networking team report
  - Changes to AWS Security Groups - weekly report
  - GuardDuty alerts/reports
- Authentication & Threat Intelligence
  - Multi-country logins, threat intelligence IP hits,
- Privileged user reporting & monitoring
  - Windows/Unix team/Networking

Use Case: Incident Response Security Group changes

kibana

Discover
Visualize
Dashboard
Timelion
Machine Learning
APM
Graph
Dev Tools
Monitoring
Management

Dashboard / AWS: Security Group Changes

Full screen | Share | Clone | Edit | Reporting | Auto-refresh | Last 7 days

Search... (e.g. status:200 AND extension:PHP) — Uses lucene query syntax

Add a filter +

AWS: SecurityGroup Changes (prod accounts)

Account ID	Account Name	UserID ARN	Action	SecurityGroup ID	CidrIP	Protocol	From Port	To Port	Date/Time	Count
[redacted]		[redacted] role/xAccountAdmin[redacted]	AuthorizeSecurityGroupIngress	[redacted]	52.38.174.146/32	tcp	22	22	August 3rd 2018, 11:10:24.065	1
[redacted]		[redacted] role/xAccountAdmin[redacted]	AuthorizeSecurityGroupIngress	[redacted]	128.114.80.0/22	tcp	4,000	4,000	August 3rd 2018, 09:28:50.908	1
[redacted]		[redacted] role/xAccountAdmin[redacted]	AuthorizeSecurityGroupIngress	[redacted]	128.114.80.0/22	tcp	4,000	22	August 3rd 2018, 09:28:50.908	1
[redacted]			AuthorizeSecurityGroupIngress	[redacted]	128.114.80.0/22	tcp	22	4,000	August 3rd 2018	1

AWS: SecurityGroup Create/Deletes (prod accounts)

Account ID	UserID ARN	Action	SecurityGroup ID	Date/Time	Count
[redacted]		DeleteSecurityGroup	[redacted]	July 31st 2018, 13:24:16.445	4
[redacted]		DeleteSecurityGroup	[redacted]	July 31st 2018, 13:15:55.636	1
[redacted]		CreateSecurityGroup	[redacted]	August 3rd 2018, 09:28:50.908	1
[redacted]		CreateSecurityGroup	[redacted]	August 3rd 2018, 11:10:24.065	1

Use Case: Incident Response
Potentially compromised accounts

IR: Security Incidents 7 hits

New Save Open Share Reporting Auto-refresh Last 4 hours

Search... (e.g. status:200 AND extension:PHP) Uses lucene query syntax

Add a filter +

logstash-securityincidents-*

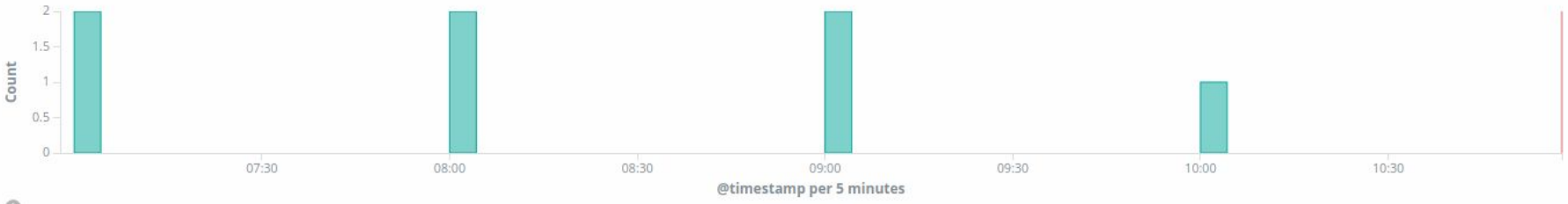
Selected Fields

- t category
- t detection_timestamp
- t reason
- t username

Available Fields

- @timestamp
- t @version
- t _id
- t _index
- # _score
- t _type
- ? logrow
- t tags

August 8th 2018, 06:57:44.188 - August 8th 2018, 10:57:44.188 — Auto



Time	username	category	reason	detection_timestamp
August 8th 2018, 10:00:19.047	[redacted]	compromised account	multi-country geo	2018-08-08 10:00:11
August 8th 2018, 09:00:26.657	[redacted]	compromised account	multi-country geo	2018-08-08 09:00:19
August 8th 2018, 09:00:19.508	[redacted]	compromised account	multi-country geo	2018-08-08 09:00:11
August 8th 2018, 08:00:26.455	[redacted]	compromised account	multi-country geo	2018-08-08 08:00:19
August 8th 2018, 08:00:19.013	[redacted]	compromised account	multi-country geo	2018-08-08 08:00:11
August 8th 2018, 07:00:26.404	[redacted]	compromised account	multi-country geo	2018-08-08 07:00:18
August 8th 2018, 07:00:18.847	[redacted]	compromised account	multi-country geo	2018-08-08 07:00:11

Use Case: Incident Response Guard Duty

- Amazon employs machine learning to act on cloudtrails, vpc flow logs and dns query logs in conjunction with Amazon threat intelligence subscription data.
- Critical in alerting on security gaps from misconfigurations
  - Alerts from GuardDuty are generated with low, medium, high severity
  - At UCSC we respond immediately and create tickets for GuardDuty messages with high severity

Use Case: AWS GuardDuty

kibana

4 hits

New Save Open Share Reporting Auto-refresh Last 7 days

Search... (e.g. status:200 AND extension:PHP)

Uses lucene query syntax

Discover
Visualize
Dashboard
Timelion
Machine Learning
APM
Graph
Dev Tools
Monitoring
Management

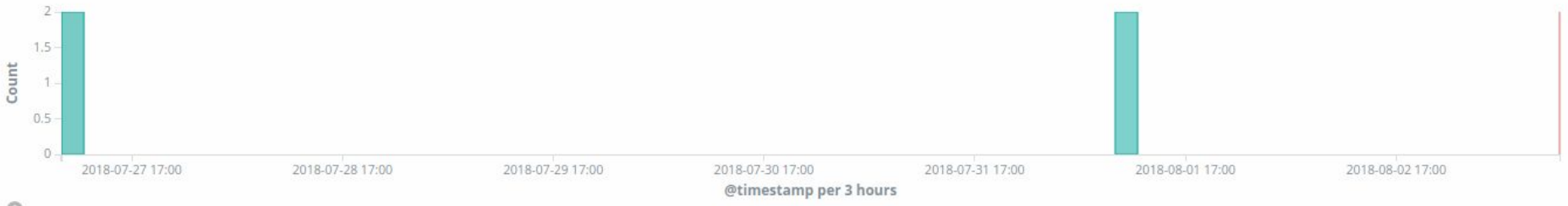
Add a filter +

logstash-guardduty-*

Selected Fields
- t detail-type
- t detail.description
- t detail.title
- t detail.type

Available Fields
- @timestamp
- t @version
- t _id
- t _index
- # _score
- t _type
- t account
- t cloudwatch_logs.event_id
- cloudwatch_logs.ingestion_time
- t cloudwatch_logs.log_group

July 27th 2018, 11:36:41.890 - August 3rd 2018, 11:36:41.890 — Auto



Time	detail-type	detail.description	detail.type	detail.title
August 1st 2018, 10:51:03.691	GuardDuty Finding	EC2 instance [redacted] is communicating with a remote host on an unusual server port 80.	Behavior:EC2/NetworkPortUnusual	Unusual outbound communication seen from EC2 instance [redacted] on server port 80.
August 1st 2018, 10:26:04.568	GuardDuty Finding	EC2 instance [redacted] is communicating with a remote host on an unusual server port 445.	Behavior:EC2/NetworkPortUnusual	Unusual outbound communication seen from EC2 instance [redacted] on server port 445.
July 27th 2018, 11:49:13.724	AWS API Call via CloudTrail	-	-	-
July 27th 2018, 11:48:32.384	AWS API Call via CloudTrail	-	-	-

Use Case: AWS Team

- Kibana interface for searching
- Shareable links to screens
- Exportable data
- Robust role-based security

Use Case: AWS Team Searches

AWS: Cloudtrail Off campus logins 17 hits

New Save Open Share Reporting Auto-refresh Last 4 hours

NOT sourceIPAddress: [128.114.0.0 TO 128.114.255.255] AND NOT sourceIPAddress: [169.233.0.0 TO 169.233.0.0]

Uses lucene query syntax

eventType: "AwsConsoleSignIn" Add a filter +

Actions

logstash-cloudtrail-*

Selected Fields

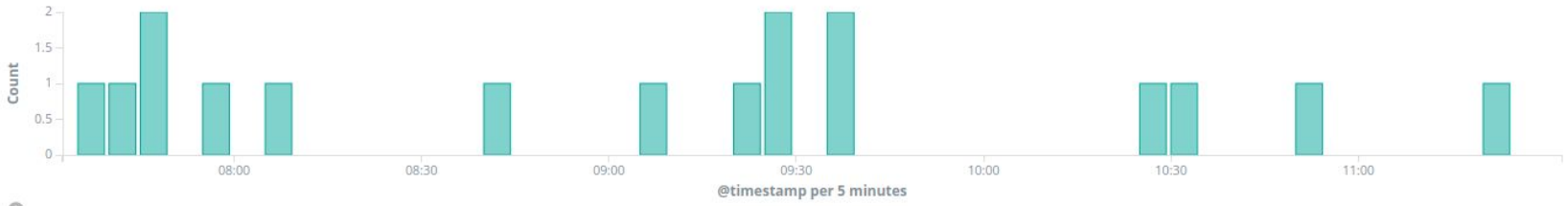
- t eventName
- t eventType
- t sourceIPAddress
- t userIdentity.arn
- t userIdentity.type

Available Fields

Popular

- t srcip
- t userIdentity.accountId
- @timestamp
- t @version
- t _id
- t _index
- # _score
- t _type
- t additionalEventData.RedirectTo
- t additionalEventData.RenewedBy
- t additionalEventData.SwitchFrom

August 7th 2018, 07:32:33.117 - August 7th 2018, 11:32:33.117 — Auto



Time	userIdentity.type	sourceIPAddress	eventName	eventType	userIdentity.arn
August 7th 2018, 11:23:21.432	AssumedRole	128.114.81.173	SwitchRole	AwsConsoleSignIn	arn:aws:sts::264087447962:assumed-role/xAccountAdmin/gholbert
August 7th 2018, 10:54:21.492	AssumedRole	128.114.81.118	RenewRole	AwsConsoleSignIn	arn:aws:sts::264087447962:assumed-role/xAccountAdmin/philme
August 7th 2018, 10:31:24.035	AssumedRole	128.114.81.172	SwitchRole	AwsConsoleSignIn	arn:aws:sts::105675313695:assumed-role/xAccountAdmin/jjendrea
August 7th 2018, 10:25:18.438	AssumedRole	128.114.81.125	RenewRole	AwsConsoleSignIn	arn:aws:sts::595663169414:assumed-role/xAccountAdmin/jkeh
August 7th 2018, 09:38:29.976	AssumedRole	128.114.81.118	RenewRole	AwsConsoleSignIn	arn:aws:sts::264087447962:assumed-role/xAccountAdmin/philme
August 7th 2018, 09:38:09.374	AssumedRole	128.114.81.196	SwitchRole	AwsConsoleSignIn	arn:aws:sts::652962613023:assumed-role/xAccountSuperAdmin/mat
August 7th 2018, 09:28:06.993	AssumedRole	128.114.81.154	SwitchRole	AwsConsoleSignIn	arn:aws:sts::652962613023:assumed-role/xAccountAdmin/tjwright
August 7th 2018, 09:26:30.259	AssumedRole	169.233.235.237	RenewRole	AwsConsoleSignIn	arn:aws:sts::595663169414:assumed-role/xAccountAdmin/tgraham
August 7th 2018, 09:20:47.637	AssumedRole	169.233.209.85	SwitchRole	AwsConsoleSignIn	arn:aws:sts::686483500175:assumed-role/xAccountAdmin/ama149
August 7th 2018, 09:08:49.163	AssumedRole	128.114.81.154	SwitchRole	AwsConsoleSignIn	arn:aws:sts::450868343536:assumed-role/xAccountAdmin/tjwright

Use Case: AWS Team Searches

Secure | https://itsec-prod-elk-1.ucsc.edu:5601/app/kibana#/discover?_g=(refreshInterval:(display:Off,pause:!f,value:0),time:(from:now%2Fd,mode:quick,to:now%2Fd))&_a=(columns:!(_source),index:'13c93960-f59b-11e7-8869-...

kibana

Discover
Visualize
Dashboard
Timelion
Machine Learning
APM
Graph
Dev Tools
Monitoring
Management
mat
Logout
Collapse

31 hits

New Save Open Share Reporting Today

cloudwatch_logs.log_group:var_log_messages AND message:stalls

Uses lucene query syntax

Add a filter +

logstash-cloudwatch-ais-*

Selected Fields

? _source

Available Fields

@timestamp
t @version
t _id
t _index
_score
t _type
t cloudwatch_logs.event_id
cloudwatch_logs.ingestion_time
t cloudwatch_logs.log_group

Top 5 values in 31 / 31 records

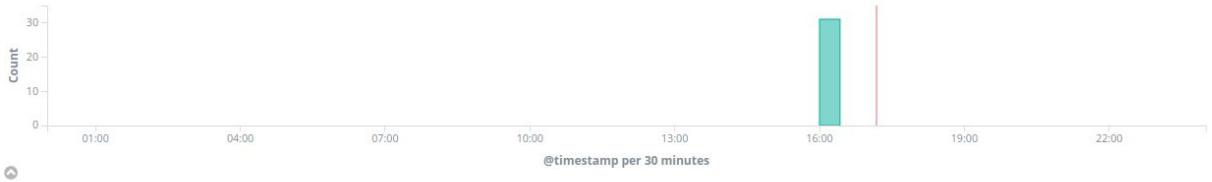
var_log_messages 100.0%

t cloudwatch_logs.log_stream
t message

Top 5 values in 31 / 31 records

Mar 5 15:11:45 cs9-prd-app-1 kerne... 3.2%
Mar 5 15:12:46 cs9-prd-app-1 kerne... 3.2%
Mar 5 15:13:46 cs9-prd-app-1 kerne... 3.2%

March 5th 2018, 00:00:00.000 - March 5th 2018, 23:59:59.999 — Auto



Time	_source
March 5th 2018, 16:10:28.559	cloudwatch_logs.log_group: var_log_messages message: Mar 5 16:03:18 cs9-prd-app-1 kernel: INFO: rcu_sched detected stalls on CPUs/tasks: { 4} (detected by 2, t=60002 jiffies, g=4414616, c=4414615, q=59003) @version: 1 cloudwatch_logs.ingestion_time: March 5th 2018, 16:10:34.328 cloudwatch_logs.event_id: 33903712058487095835635035030865934141268818076651684587 cloudwatch_logs.log_stream: cs9-prd-app-1 @timestamp: March 5th 2018, 16:10:28.559 tags: cloudwatch-ais, cloudwatch-ais _id: Rv6l-GEB2LG0Qh-D-R-A _type: logs _index: logstash-cloudwatch-ais-2018.03.06 _score: -
March 5th 2018, 16:10:28.551	cloudwatch_logs.log_group: var_log_messages message: Mar 5 16:02:04 cs9-prd-app-1 kernel: INFO: rcu_sched detected stalls on CPUs/tasks: { 4} (detected by 2, t=60002 jiffies, g=4414557, c=4414556, q=58515) @version: 1 cloudwatch_logs.ingestion_time: March 5th 2018, 16:10:34.328 cloudwatch_logs.event_id: 33903712058308689874046790045733648395087631184603841246 cloudwatch_logs.log_stream: cs9-prd-app-1 @timestamp: March 5th 2018, 16:10:28.551 tags: cloudwatch-ais, cloudwatch-ais _id: wv6m-GEB2LG0Qh-DBy2Y _type: logs _index: logstash-cloudwatch-ais-2018.03.06 _score: -
March 5th 2018, 16:10:28.538	cloudwatch_logs.log_group: var_log_messages message: Mar 5 16:01:01 cs9-prd-app-1 kernel: INFO: rcu_sched detected stalls on CPUs/tasks: { 4} (detected by 2, t=60002 jiffies, g=4414534, c=4414533, q=71635) @version: 1 cloudwatch_logs.ingestion_time: March 5th 2018, 16:10:34.328 cloudwatch_logs.event_id: 33903712058018780186465891944893684057543202485026095819 cloudwatch_logs.log_stream: cs9-prd-app-1 @timestamp: March 5th 2018, 16:10:28.538 tags: cloudwatch-ais, cloudwatch-ais _id: tf6m-GEB2LG0Qh-DBy2Y _type: logs _index: logstash-cloudwatch-ais-2018.03.06 _score: -
March 5th 2018, 16:10:28.521	cloudwatch_logs.log_group: var_log_messages message: Mar 5 15:57:09 cs9-prd-app-1 kernel: INFO: rcu_sched detected stalls on CPUs/tasks: { 4} (detected by 1, t=60002 jiffies, g=4413837, c=4413836, q=58740) @version: 1 cloudwatch_logs.ingestion_time: March 5th 2018, 16:10:34.328 cloudwatch_logs.event_id: 33903712057639667518090871351487576846908180339424428720 cloudwatch_logs.log_stream: cs9-prd-app-1 @timestamp: March 5th 2018, 16:10:28.521 tags: cloudwatch-ais, cloudwatch-ais _id: bkel-GEB1eVcv5Wj_u2Q _type: logs

Use Case: AWS Team Searches

@timestamp per 5 minutes

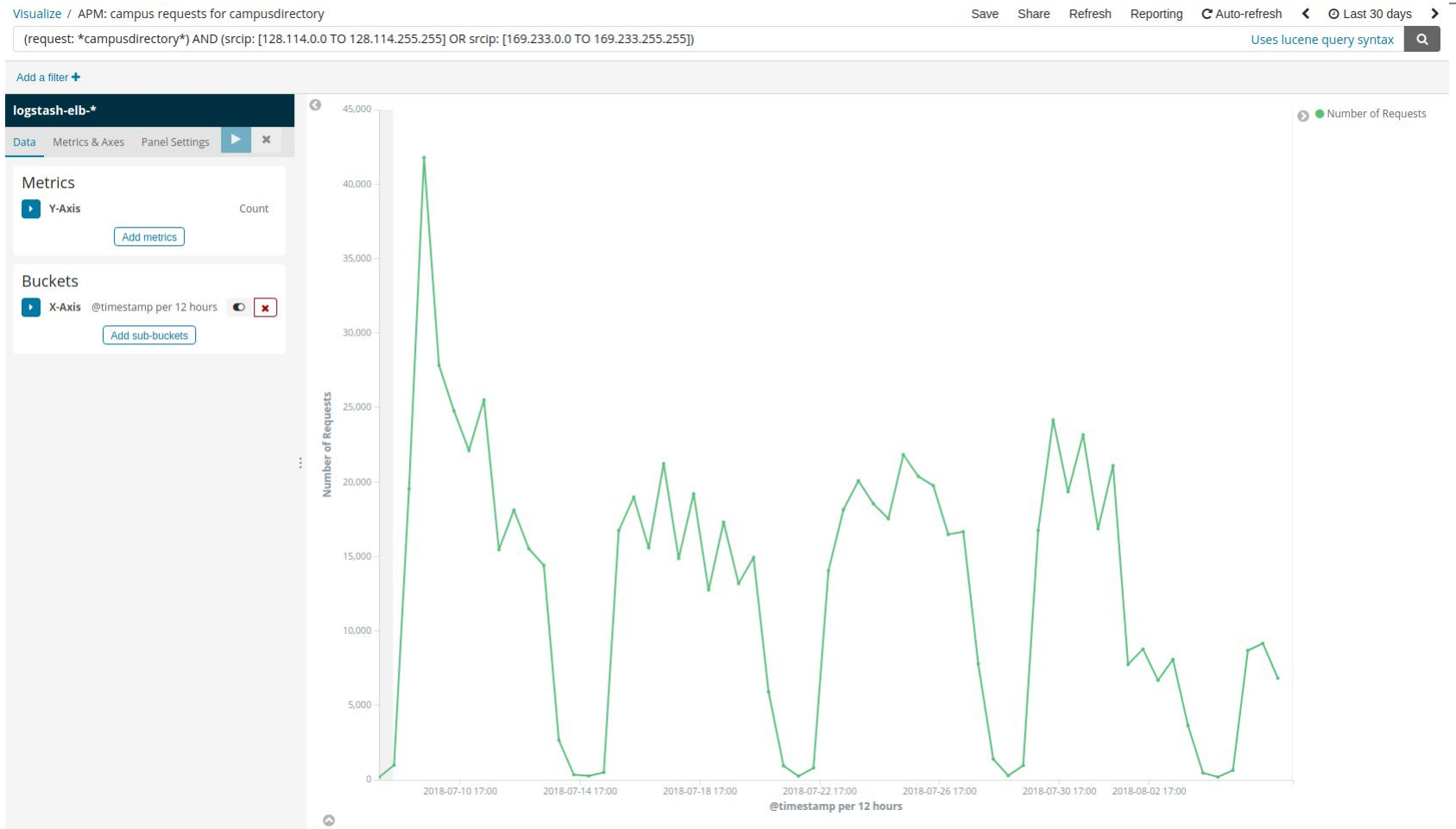
Time	userIdentity.type	sourceIPAddress	eventName	eventType	userIdentity.arn
August 8th 2018, 10:53:51.855	AssumedRole	128.114.81.156	SwitchRole	AwsConsoleSignIn	arn:aws:sts:[redacted]assumed-role/xAccountAdmi[redacted]

Table JSON

View surrounding documents | View single

	@timestamp	August 8th 2018, 10:53:51.855
t	@version	1
t	_id	INWsGmUBU3txxQ605PkI
t	_index	logstash-cloudtrail-2018.08.08
#	_score	-
t	_type	logs
t	additionalEventData.RedirectTo	https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Home:
t	additionalEventData.SwitchFrom	arn:aws:iam:[redacted]
t	awsRegion	us-east-1
t	cloudwatch_logs.event_id	34203786499232740368876112284403844909833528827804516373
	cloudwatch_logs.ingestion_time	August 8th 2018, 10:53:51.896
t	cloudwatch_logs.log_group	CloudTrail/DefaultLogGroup
t	cloudwatch_logs.log_stream	264087447962_CloudTrail_us-west-2
t	eventID	524ab10d-0936-4919-a753-bb407e3809da
t	eventName	SwitchRole

Use Case: AWS application traffic



Presentation Outline

- Intro/Infrastructure
- Logging pipeline
- Use cases
- **End Results**

End Results

- Project goals met for operational security needs in AWS
- Actionable application logs suitable for AWS developers/engineers
- Scalable logging pipeline for future AWS growth
- Some new costs associated with services and accounts vs. on-premise

Our github repo

- https://github.com/ucsc/awslogging
- Lambda functions
- Logstash configuration files
- This presentation

Thank you for coming

UCCSC 2018

8/15/2018

UC SANTA CRUZ

Team Members:

Troy Wright

tjwright@ucsc.edu

Brian Hall

brian@ucsc.edu