RSAConference2020

San Francisco | February 24 – 28 | Moscone Center

HUMAN ELEMENT

SESSION ID: CSV-W01

Logging in the Cloud: From Zero to (Incident Response) Hero

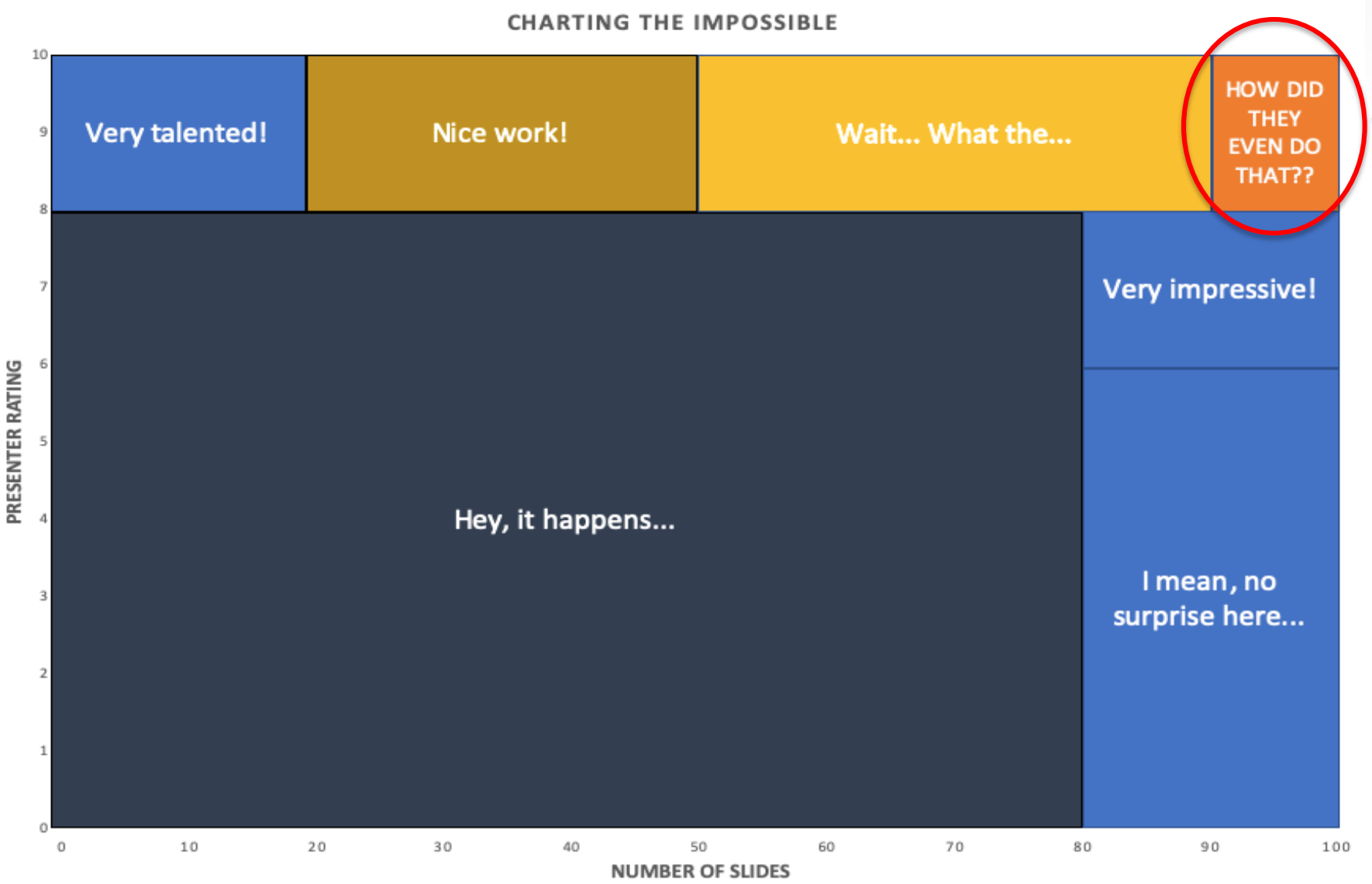
Jonathon Poling

Managing Principal Consultant
Secureworks
@JPoForenso

Agenda

```
for those in [ aws , Azure , Google Cloud Platform ]:
    print("What Should I Be Logging?")
    print("How *Specifically* Should I Configure it?")
    print("What Should I Be Monitoring?")
else:
    print("Questions?")
```

Today, We (Attempt to) Make History…



I plan to live here…

Why Me?

- Cloud (AWS) SME for Secureworks
- Developed Secureworks' AWS Incident Response Service Line
- Help SMB through Fortune 10 Customers…
  - Intelligently Configure/Instrument Their Environments
  - Protect Their Infrastructure
  - Effectively Respond to Incidents

Why This Presentation?

- Too many clouds, too little time
  - Many of us are still lacking foundational understanding of Cloud operations and security
  - It's extremely hard to master one cloud, let alone multiple
- Tired of presentations with no actionable takeaways
  - People need prescriptive actions to take that can help them to *immediately* start getting/operating/securing their Cloud(s) better
- Helping us to help you (to help us and help you)

How Will This Help You?

In this talk you will (hopefully) learn:

- Core log capabilities of each Cloud provider
- Which core logs should be configured (specifically how)
- Tips for Monitoring core logs
- A few tips/tricks for Incident Response along the way

Get Ready for a LOT of Material…

RSAConference2020

Amazon Web Services (AWS)

Overview of Logging

Core Logs

- CloudTrail
  - Your account's syslog on steroids
  - Enabled by Default for 90 days of retention BUT…
    - Each region's logs are kept ONLY in that region's bucket (ROYAL PAIN for response)
    - Only "Global" (IAM/STS) service events will be logged across all regions/buckets
      - But… some aren't… (DON'T @ ME "ConsoleLogin"!)

Core Logs

- CloudWatch
  - System performance metrics
    - Enabled by default (metrics sent every 15 minutes)
    - Enabling “Detailed Monitoring” will send metrics every 1 minute
  - OS/Application Logs
    - Send to CloudWatch via EC2 Systems Manager (SSM) and/or CloudWatch Logs Agent
      - Both require installation of additional agent on each Instance
  - Additional stuff you’re also sending (CloudTrail, VPC Flow Logs, etc.)

Core Logs

- Config
  - Track Resource “Compliance” against a set of rules
  - Easy setup via Console or CLI
  - Deliver config logs to SNS Topic and/or S3
  - Config Rules
    - Enable various default Config Rules to monitor/alert on configuration changes as they occur or on a schedule
    - Create custom rules according to your environment and policies
    - AWS Managed Rules provided/enabled by default
  - Now with Multi-Account Multi-Region Data Aggregation

Core Logs

- Config
  - (BONUS) Software Monitoring
    - Monitor/record software inventory/changes
      - Requires Instances to be configured as “Managed Instances”

Core Logs

- S3
  - Bucket-Level (aka Management Event) Logs
    - Delete/Get/Put Bucket* type actions
    - Enabled by default
  - Object-Level (aka Data Event) Logs
    - Delete/Get/Put Object* type actions
    - Must be manually configured
  - Server Access Logs
    - Apache-ish type logs (Remote IP, URI, Bytes Sent, Referer, User-Agent, etc.)
    - Must be manually configured

Core Logs

- VPC Flow Logs
  - Netflow(ish) type connection logs
  - Can be enabled for VPC, VPC Subnet, or Elastic Network Interface (ENI)
    - Enable for anything of which you might even remotely care about the incoming/outgoing traffic
  - Logged to CloudWatch Logs as a new Log Group with a Stream for each associated ENI
    - Create CloudWatch Metric Filters/Alarms for traffic you care about

Core Logs

- Load Balancer Logs
  - Elastic Load Balancer (ELB) Logs
    - Now referred to as “Classic Load Balancer” (CLB)
    - Logs the details of each request made to the load balancer
      - Timestamp, Client/Backend IP/Port, Processing Time, Sent/Received Bytes, User Agent, etc.
    - Publishes a log file for each ELB node every 5 or 60 (default) minutes
    - Disabled by default

Core Logs

- Load Balancer Logs
  - Application Load Balancer (ALB) Logs
    - Logs requests (*as best effort*) sent to the load balancer, including requests that never made it to the targets (malformed requests, requests with no target response)
    - Logs the details of each request/connection made to the Load Balancer
      - Connection Type, Timestamp, Client/Target IP/Port, Status Code, Sent/Received Bytes, User Agent, etc.
    - Publishes a log file for each ALB node every 5 minutes
    - Disabled by default

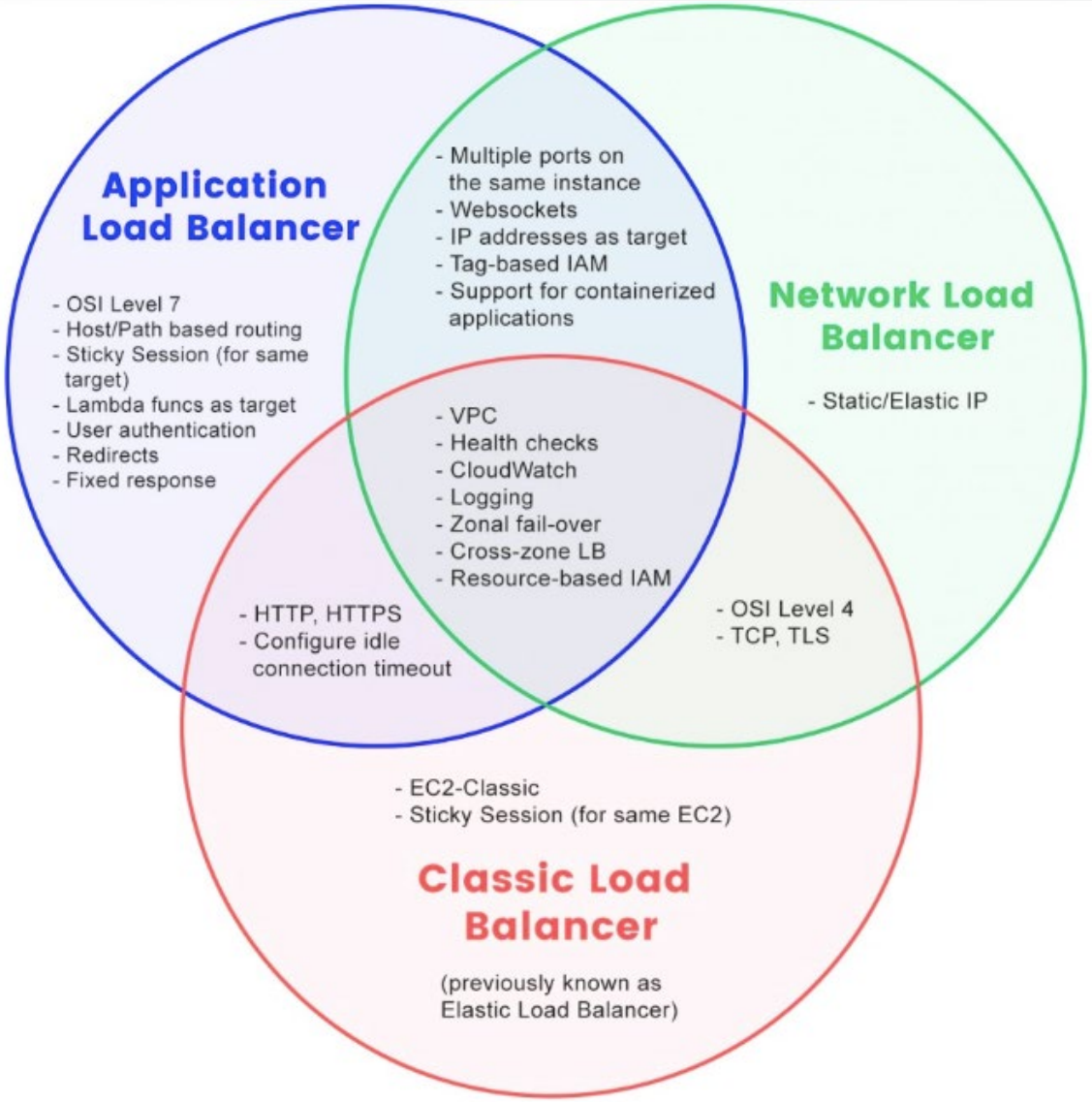
Core Logs

- Load Balancer Logs
  - Network Load Balancer (NLB) Logs
    - Logs detailed information about the *TLS requests* sent to your NLB
      - **Access logs are created only if the load balancer has a TLS listener and they contain information only about TLS requests!**
    - Logs the details of each TLS single request/connection made to the Load Balancer
      - Timestamp, Client/Target IP/Port, Sent/Received Bytes, TLS Cipher, TLS Protocol Version, etc.
    - Publishes a log file for each NLB node every 5 minutes
    - Disabled by default

Core Logs

	CLB	ALB	NLB
Protocols	TCP, SSL/TLS, HTTP, HTTPS	HTTP, HTTPS	TCP, TLS
Performance (a higher number is slower): the ability to handle more traffic	2	3	1 (fastest)
Host/Path-based routing	No	Yes	No
Sticky Session (for session-based applications)	Yes (redirect to the same machine)	Yes (redirect to the same target)	No
Static/Elastic IP	No	No	Yes
Load balancing to multiple ports on the same instance	No	Yes	Yes
Configurable idle connection timeout	Yes	Yes	No

Core Logs


Application Load Balancer
- OSI Level 7
- Host/Path based routing
- Sticky Session (for same target)
- Lambda funcs as target
- User authentication
- Redirects
- Fixed response
- Multiple ports on the same instance
- Websockets
- IP addresses as target
- Tag-based IAM
- Support for containerized applications
Network Load Balancer
- Static/Elastic IP
- VPC
- Health checks
- CloudWatch
- Logging
- Zonal fail-over
- Cross-zone LB
- Resource-based IAM
- HTTP, HTTPS
- Configure idle connection timeout
- OSI Level 4
- TCP, TLS
- EC2-Classic
- Sticky Session (for same EC2)
Classic Load Balancer
(previously known as Elastic Load Balancer)

RSAConference2020

Amazon Web Services (AWS)

Configuring Logging

CloudTrail

- Configuring Global/Central Logging to a single bucket
  - Navigate to **CloudTrail**
  - Ensure you're in the Region where you'd like your CT logs centralized
  - Select **Trails**
  - Click **Create Trail**
  - Input the **Trail Name**
  - Select **Apply trail to all regions**
- Note: IAM Events will be duplicated across all regions
  - Used to be able to disable Global Events in all Buckets except one
  - Documentation no longer references how to do this, so… YMMV

CloudWatch

- Certain Logs automatically sent to CloudWatch
  - CloudFront, Config, GuardDuty
- Enabling Detailed Monitoring (per Instance)
  - New Instances
    - In **Step 3** of your **Instance Configuration**, select **Enable Cloudwatch detailed monitoring**
  - Existing Instances
    - Navigate to **EC2**
    - Select **Instances**
    - Right-click the **Instance**
    - Select **CloudWatch Monitoring** -> **Enable Detailed Monitoring**

CloudWatch

- Configuring CloudWatch Logs Agent
  - Configure IAM Role to Allow Instance to write to CloudWatch
    - Either create a new Role or modify existing Role(s) to have the permissions specified in the **CloudWatchAgentServerPolicy** Policy
  - Configure Linux Instance to send OS/Host logs to CloudWatch
    - Download and Install the CloudWatch Logs Agent

```
$ wget <link_to_proper_package>
$ sudo rpm -U ./amazon-cloudwatch-agent.rpm
```

OR

```
$ sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
```

CloudWatch

- Configuring CloudWatch Logs Agent
  - Configure Linux Instance to send OS/Host logs to CloudWatch (Cont.)
    - Configure the CloudWatch Logs Agent Configuration File
      - Modify the config the collect the appropriate metrics and logs from your system(s)
    - Start the CloudWatch Logs Agent

```
$ sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:configuration-file-path -s
```

CloudWatch

- Configuring CloudWatch Logs Agent
  - Configure Windows Instance to send OS/Host logs to CloudWatch
    - Download and Install the CloudWatch Logs Agent

      Link: `https://s3.amazonaws.com/amazoncloudwatch-agent/windows/amd64/latest/amazon-cloudwatch-agent.msi`

      ```
      > msiexec /i amazon-cloudwatch-agent.msi
      ```
    - Configure the CloudWatch Logs Agent Configuration File
      - Modify the config the collect the appropriate metrics and logs from your system(s)
    - Start the CloudWatch Logs Agent (via PowerShell)

      ```
      > & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -a fetch-config -m ec2 -c file:configuration-file-path -s
      ```

CloudWatch

- Configuring CloudWatch Logs Agent
  - Can also:
    - Install CloudWatch Logs Agent using SSM (if Instances are instrumented)
    - Install CloudWatch Logs Agent on on-premises systems to send to CW in AWS

CloudWatch

- Configuring CloudTrail to send logs to CloudWatch
  - Navigate to **CloudTrail**
  - Select the appropriate **Trail**
  - Within the **CloudWatch Logs** section, click **Configure**
  - Specify a **New or existing log group**
  - Click **Continue**
  - Create a New or select an Existing **IAM Role** and **Policy Name**
  - Click **Allow**

CloudWatch

- Configuring VPC Flow Logs to send to CloudWatch
  - Create a VPC Flow Logs IAM Role

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogGroups",
        "logs:DescribeLogStreams"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "vpc-flow-logs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

CloudWatch

- Configuring VPC Flow Logs to send to CloudWatch
  - Create a VPC Flow Logs IAM Role (Cont.)
    - Users will also need **PassRole** permissions for the Role

```
{
  "Version": "2012-10-17",
  "Statement": [ {
      "Effect": "Allow",
      "Action": ["iam:PassRole"],
      "Resource": "arn:aws:iam::account-id:role/flow-log-role-name"
      } ]
}
```

CloudWatch

- Configuring VPC Flow Logs to send to CloudWatch
  - Configure VPC Flow Log to publish to CloudWatch
    - Navigate to **EC2**
    - Select **Network Interfaces**
    - Right-click on the appropriate network Interface and select **Create Flow Log**
    - Select the appropriate traffic **Filter** (Accept, Deny, All)
    - Select the **Maximum aggregation interval** (1 or 10 minutes)
    - Select the **Destination** to **Send to CloudWatch Logs**
    - Enter the **Destination log group**
    - Select the previously created **IAM Role**
    - Click **Create**

Config

- Configuring Multi-Region Aggregation
  - Set up an Aggregator for all Regions
    - Navigate to **AWS Config**
    - Select **Aggregated View** -> **Aggregators**
    - Click **Add Aggregator**
    - Select **Allow AWS Config to replicate data from source account(s) into an aggregator account. You must select this checkbox to continue to add an aggregator.**
    - Input a unique **Aggregator Name**
    - Select either:
      - **Add individual account IDs** (input Account ID's to include)
      - **Add my organization** (create/choose the appropriate IAM Role)
    - Select all available **Region(s)**
    - Select **Allow AWS Config to aggregate data from all future AWS regions where AWS Config is enabled.**
    - Click **Save**

Config

- Configuring Multi-Region Aggregation
  - Authorize Aggregators for Regions
    - Navigate to **AWS Config**
    - Select **Authorizations**
    - Click **Add authorization**
    - Input **Aggregator Account**
    - Select **Aggregator Region**
    - Click **Add authorization**

Config

- Configuring Config Rules (that sounds weird*)
  - Adding Managed Rules
    - Navigate to **AWS Config**
    - Select **Rules**
    - Click **Add rule**
    - Search/filter based on rule name or description
    - Select the appropriate **Rule**
    - Configure the **Rule** as needed
    - Click **Save**

*But not as weird as AWS Systems Manager Session Manager…

Config

- Configuring Config Rules (that sounds weird*)
  - Adding Custom Rules
    - Navigate to **AWS Config**
    - Select **Rules**
    - Click **Add rule**
    - Click **Add custom rule**
    - Configure the **Custom Rule** as needed
      - Name, Description, Lambda, Trigger, Rule Parameters, and Remediation Action
    - Click **Save**

S3

- Enabling MFA Delete
  - Can only be configured via the AWS CLI (unless I am missing something)
  - Configuring MFA Delete for a Bucket via the AWS CLI

```
$ aws s3api put-bucket-versioning --bucket my_bucket
--versioning-configuration '{"MFADelete":"Enabled"}'
```

  - Consider using **S3 Object Lock** as an alternative and/or added measure for preventing unintended/malicious data deletion

S3

- Enabling Object-Level Logging
  - Via S3 (for Specific Bucket)

    *Can also configure upon Bucket Creation in **Configure options**

    - Navigate to **S3**
    - Select the appropriate **Bucket**
    - Navigate to the **Properties** tab
    - Click **Obect-level logging**
    - Select the **Bucket** for recording the activity
    - Select **Read** and **Write** for Events
    - Click **Create**

S3

- Enabling Object-Level Logging
  - Via CloudTrail (For All Buckets)
    - Navigate to **CloudTrail**
    - Select **Trails**
    - Click the appropriate **Trail**
    - Under **Data events**, click **Configure** under the **S3** tab
    - Click **Select all S3 buckets in your account**
    - Click **Save**

S3

- Enabling Server Access Logs
  - Navigate to **S3**
  - Create Target Bucket for collecting the Server Access Logs
    - Click **Create bucket**
      - Within the **Set permissions** tab, under **Manage system permissions**, ensure **Grant Amazon S3 Log Delivery Group write access to this bucket** is selected from the drop-down list

S3

- Enabling Server Access Logs
  - Configure Server Access Logging (per Bucket)
    - Click the **Bucket** for which you'd like to enable Server Access Logs
    - Navigate to the **Properties** tab
    - Select **Server access logging**
    - Click **Enable logging**
    - Input the previously created **Target Bucket**
    - (Optional) Enter a **Target prefix** (e.g., "ServerAccessLogs")
    - Click **Save**

VPC Flow Logs

- Configuring per ENI
  - Navigate to **EC2**
  - Right-click the appropriate **ENI**, select **Create flow log**
- Configuring per Subnet
  - Navigate to **VPC** -> **Subnets**
  - Right-click the appropriate **Subnet**, select **Create flow log**
- Configuring per VPC
  - Navigate to **VPC** -> **Your VPCs**
  - Right-click the appropriate **VPC**, select **Create flow log**

Load Balancer Logs

- Configuring ALB/NLB Access Logs
  - Navigate to **EC2** -> **Load Balancers**
  - Select the appropriate **Load Balancer**
  - Scroll to the bottom of the **Description** tab
  - Click **Edit Attributes**
  - Check the **Access logs** box
  - Input the appropriate **S3 location**
    - Select **Create this location for me** if it does not yet exist
  - Click **Save**

Load Balancer Logs

- Configuring ELB (Classic) Access Logs
  - Navigate to **EC2** -> **Load Balancers**
  - Select the appropriate **Load Balancer**
  - Scroll to the bottom of the **Description** tab
  - Click **Configure Access Logs**
  - Check the Enable **Access logs** box
  - Select the appropriate **Interval**
  - Input the appropriate **S3 location**
    - Select **Create this location for me** if it does not yet exist
  - Click **Save**

CloudFront Logs

- Configuring CloudFront Access Logs (per Distribution)
  - Navigate to **CloudFront -> Distributions**
  - Select the appropriate **Distribution**
  - Under the **General** tab, click **Edit**
  - Within the **Distribution Settings** tab, scroll down to the **Logging** section
  - Select **On** for **Logging**
  - Input the appropriate target **Bucket for Logs**
  - (Optional) Input a **Log Prefix**
  - Click **Yes, Edit**

RSAConference2020

Amazon Web Services (AWS)

Tips for Monitoring

CloudWatch Alarms

- Create CloudWatch Alarms for various Metrics:
  - CloudFront
    - Inordinate number of 4xx/5xx errors, anomalous bytes downloaded/uploaded, …
  - EC2 Instances
    - High CPU/Memory utilization, high CPU Credit Usage, StatusCheckFailed's, …
  - Load Balancers
    - High number of active or rejected connections, auth errors, high response times, …
  - VPC Flow Logs
    - Anomalous traffic increases/spikes or inbound/outbound data transfer, …
  - …

CloudWatch Events

- Create CloudWatch Events for:
  - Config Rules
    - Disable accounts when/where MFA is disabled
  - CloudTrail Actions/API Calls
    - Alert and re-enable CloudTrail Logging if ever stopped/deleted
  - GuardDuty Alerts
    - Shut down Instances found to be compromised with CryptoMiners
  - TrustedAdvisor Findings
    - Alert/respond (lambda) to MFA disable for root account, public EBS Snapshots, service limits hit, …
  - VPC Flow Logs
    - Alert on known malicious IP's, SSH Brute Force attacks, RDP traffic, …
  - …

Log Analysis in Athena

- Athena provides a super easy and scalable option for log analysis
- Query any data (directly) that resides in S3
- Create tables/queries on the fly
- Perform highly parallelized and efficient searches across massive amounts of data*

* With the proper data partitioning!

Tons More Tips for AWS Alerting/Monitoring…

If you're interested in learning more about AWS Alerting and Monitoring, check out my other talks on the subjects (links on my website)…

RSAConference2020

Microsoft Azure

Overview of Logging

Core Logs

- Activity Logs
  - Management Plane events (Operations performed against your subscription)
    - All Create, Update, List, or Delete actions performed
      - Create Virtual Machine, Delete Network Security Group (NSG), …
- Resource (Diagnostics) Logs
  - Data Plane events (Operations your Resource itself performed)
    - Getting a Secret from a Key Vault, Querying a DB, VM Metrics/Operations, …
- Azure Active Directory Logs
  - Active Directory activities/events (with built-in reports)

Core Logs

- Windows Azure Diagnostics (WAD)
  - Collects host/system logs
- Application Logs/Insights
  - Monitor Application Health and Performance
  - Collect and Monitor Application/Server Logs
- Storage Analytics Logs
  - Detailed information about requests to Storage service

Core Logs

- Network Security Group (NSG) Flow Logs
  - Netflow(ish) Logs
    - Source/Dest IP, Source/Dest Port, Protocol, Allowed/Denied, Bytes/Packets Sent
  - Diagnostic Logs
    - See which (and how) firewall rules were triggered/applied to traffic
- Security Center
  - Provides a variety of endpoint and account-based monitoring and threat detections
  - Endpoint log analytics agent (Microsoft Monitoring Agent) must be specifically configured

RSAConference2020

Microsoft Azure

Configuring Logging

Activity Logs

- Activity Logs
  - Enabled by default
  - Configure via:
    - Navigate to **Azure Monitor**
    - Select **Activity Log**
    - Select **Diagnostic Settings**
    - Configure + send to:
      - Storage
      - Log Analytics Workspace (for Azure Monitor)
      - Event Hub

Resource Logs

- Resource (Diagnostic) Logs
  - Each Resource requires its own configuration
  - Configuration for a single resource:
    - Select **Monitoring** -> **Diagnostic Settings**
    - Select **Add diagnostic setting**
    - Configure + send to:
      - Storage
      - Log Analytics Workspace (for Azure Monitor)
      - Event Hub
  - Configuration for multiple resources:
    - Navigate to **Azure Monitor**
    - Select **Settings** -> **Diagnostic Settings**

Active Directory Logs

- Active Directory Logs
  - Enabled by default with the following logs/reports:
    - Audit Logs
    - Sign-in Logs
    - Risky Sign-in Logs
    - Users Flagged for Risk Logs
    - Provisioning Logs
  - Configure via:
    - Navigate to **Azure Active Directory** -> **Diagnostic Settings**
    - Select **Add diagnostic setting**
    - Configure **AuditLogs** and/or **SignInLogs** to send to:
      - Storage
      - Log Analytics Workspace (for Azure Monitor)
      - Event Hub

Windows Azure Diagnostics (WAD) Logs

- Windows Azure Diagnostics
  - Configuration via:
    - Windows Azure Diagnostics (send to Storage, Log Analytics, Azure Monitor)
    - Windows Event Forwarding (send to your SIEM)
  - Configuration for VM's:
    - Configure diagnostics at run/build time manually or using templates

Application (Diagnostic) Logs

- Configure Application Logging (Windows) – per App:
  - Navigate to **App Service Logs**
  - Select **On** for:
    - **Application Logging (Filesystem) – Temporary (12-hour) storage for debugging purposes**
    - **Application Logging (Blob) – Long term storage**
  - Select the (Log) **Level**
- Configure Application Logging (Linux/Container) – per App:
  - Navigate to **App Service Logs**
  - Select **Application Logging** -> **File System**
  - Configure:
    - **Quota (MB)**
    - **Retention Period (Days)**

Application (Diagnostic) Logs

- Configure Web Server Logging – per App:
  - Navigate to **App Service Logs**
  - Select **Web Server Logging**
  - Select to send to:
    - **Storage**
    - **File System**
  - Configure **Retention Period (Days)**
- Configure Detailed Error Messages – per App:
  - Navigate to **App Service Logs**
  - Set **Detailed Error Logging** to **On**

Application (Diagnostic) Logs

- Configure Failed Request Tracing – per App:
  - Navigate to **App Service Logs**
  - Set **Failed Request Tracing** to **On**
- Configure Deployment Logging – per App:
  - Enabled by default
    - "Happens automatically and there are no configurable settings for deployment logging. It helps you determine why a deployment failed."

Storage Analytics Logs

- Storage Analytics
  - Configure via Azure Portal – per Storage Account:
    - Navigate to **Storage Accounts**
    - Select the appropriate **Storage Account**
    - Select **Monitoring (Classic)** -> **Diagnostics Settings (Classic)**
    - Select the appropriate **Metrics**:
      - **API Metrics, Delete Data**
    - Select the appropriate **Logging**:
      - **Read, Write, Delete, Delete Data**
    - Set the **Retention (Days)**

Network Security Group (NSG) Logs

- NSG Flow Logs
  - Pre-Requisites:
    - Register Microsoft.Insights Provider – per Subscription:
      - Navigate to **Subscriptions**
      - Select the appropriate **Subscription**
      - Select **Settings** -> **Resource Provider**
      - Select **Register**
    - Enable Network Watcher – per Region:
      - Navigate to **Network Watcher**
      - Click the "**>**" next to the Regions to expand them
      - Select the "**…**" next to each appropriate Region
      - Select **Enable Network Watcher**

Network Security Group (NSG) Logs

- NSG Flow Logs
  - Configure NSG Flow Logs – per NSG:
    - Navigate to Network Watcher
    - Select **Logs** -> **NSG Flow Logs**
    - Select the appropriate **NSG**
    - Under **Flow Logs**, select **On**
    - Select **Version 2** for Flow Logs version (includes bytes/packets count + flow state)
    - Select the appropriate **Storage Account**
    - Select the appropriate **Retention Period (Days)** – for Storage v2 Accounts

Network Security Group (NSG) Logs

- NSG Flow Logs
  - Configure NSG Flow Logs – per NSG:
    - Optional
      - Under **Traffic Analytics Status**, select **On**
      - Select **Processing Interval** (1 Hour, 10 Minutes)
      - Select existing (or new) **Log Analytics Workspace** as a log destination (for later analysis)

Security Center

- Security Center
  - Configure endpoint log analytics agent via:
    - Automatic Provisioning (for all Azure VM's)
      - Select **Pricing & Settings**
      - Select the appropriate **Subscription**
      - Select **Data Collection**
      - Set **Auto Provisioning** to **On**
      - Select the appropriate **Workspace** for log destination

Security Center

- Security Center
  - Configure endpoint log analytics agent via:
    - Automatic Provisioning (for all Azure VM's)
      - Optional – **Store Additional Raw Data**
        - **None** (not recommended)
        - **Minimal** ("This set covers only events that might indicate a successful breach and important events that have a very low volume.") – 4624 / 4625 / 4688 / …
        - **Common** ("Provide a full user audit trail in this set.") – 4634 / ...
        - **All Events** (All Windows Security and AppLocker events)

Security Center

- Security Center
  - Configure endpoint log analytics agent via:
    - Manual Provisioning
      - Ensure Auto Provision is set to Off
      - Select **Pricing & Settings**
      - Select the appropriate **Subscription**
      - Ensure the **Pricing Tier** is set to **Standard**
      - Deploy Monitoring Agents to:
        - New VM's via a Resource Manager Template
        - Existing VM's via

Security Center

- Security Center
  - Configure endpoint log analytics agent via:
    - Manual Provisioning
      - Deploy Monitoring Agents to:
        - New VM's via a Resource Manager Template
        - Existing VM's via **Log Analytics Workspace** -> **Virtual Machines** -> Select **VM** -> Click **Connect**
        - Existing VM's via PowerShell

RSAConference2020

Microsoft Azure

Tips for Monitoring

Azure Monitor

- Activity Logs
  - Review for anomalous CREATE / DELETE / UPDATE actions
    - New Accounts
    - New resources created in unapproved methods / regions
- Network Activity
  - Review for anomalous traffic
    - After-hours traffic spikes
    - Heartbeat (C2)
    - Possible DDoS

Azure Monitor

- Resource Diagnostics (OS-level Logs)
  - Run queries for:
    - Host-level authentications
    - Process executions
    - Command-line/PowerShell activity
    - ..
- Use “Insights” Features for Anomaly Discovery

Network Watcher

- Analyze NSG Flow Logs in Network Watcher
  - Identify “Top Talkers”
  - Visualize Activity by Geographic Map
  - Statistics of Allowed vs. Blocked traffic
  - Identify “badness”:
    - Connection initiated inbound w/ large outbound data (web shell or just web server?)
    - Connection initiated outbound w/ large outbound data (reverse shell?)
    - Regular X byte connection started every Y minutes (C2?)
  - Query for known malicious IP’s

Active Directory

- Utilize Built-In Auditing and Reports to Review Authentications
  - Security Reports
    - “Users At Risk” Report
      - A “risky” user is an indicator for a user account that might have been compromised
    - “Risky Sign-In” Report
      - A “risky sign-in” is an indicator for a sign-in attempt that might have been performed by someone who is not the legitimate owner of a user account

Active Directory

- Utilize Built-In Auditing and Reports to Review Authentications
  - Activity Reports
    - Audit Logs
      - Audit all AD activities (New Users/Groups, Password Changes, New/Modified Admin Groups New/Modified Service Accounts)
    - Sign-In Report
      - Identify sign-in patterns of specific users (signing in from new location out of nowhere?)

Security Center

- Security Center
  - Use this as a force multiplier for your monitoring/security efforts
  - Secure Score
    - Review, investigate, and remediate findings
    - Start with highest impact Recommendations
  - Security Alerts
    - Monitor for, and investigate, these alerts
    - Can be early (or only) indicators of compromise

Azure Sentinel

- Azure-based native SIEM
- Connect/send all your logs to Sentinel to:
  - Use built-in (and custom) analytics for searching/alerting
  - Use built-in (or custom) workbooks to search/investigate
  - Use built-in Investigations capability (and graphs) to investigate possible incidents
  - Use Playbooks to build and automate responses to incidents

RSAConference2020

Google Cloud Platform (GCP)

Overview of Logging

Core Logs

- Activity Logs
  - API calls or other administrative actions that modify the configuration or metadata of resources
  - Enabled by default (at no charge)
  - Always written – you cannot configure/disable them
  - Automatically retained for 400 days

Core Logs

- Data Access Logs
  - API calls that create, modify, or read user-provided data
  - Disabled by default
  - Automatically retained for 30 days

Core Logs

- System Event Audit Logs
  - Log entries for Google Cloud administrative actions that modify the configuration of resources
  - Generated by Google systems (not driven by direct user action)
  - Always written – you cannot configure/disable them
  - Automatically retained for 400 days

Core Logs

- Application/Host/OS Logs
  - Collect Application and Host/OS-level logs via the Stackdriver Logging Agent
    - GCP's customized version of Fluentd
  - Monitors/collects the following logs by default:
    - Linux
      - Syslog, nginx, apache2, apache-error
    - Windows
      - Windows Event Logs

Core Logs

- VPC Flow Logs
  - Per-VM or Per-VPC network flow logs
  - Allow you to:
    - Monitor the VPC network
    - Perform network diagnosis
    - Filter the flow logs by VMs and by applications to understand traffic changes
    - Understand traffic growth for capacity forecasting
  - Built into the networking stack of the VPC network infrastructure
    - No extra delay or performance penalty in enabling

Core Logs

- Cloud Storage Logs
  - Access Logs
    - Provides info for all of the requests made on a specified bucket
      - Access to public objects
      - Changes made by the Object Lifecycle Management feature
    - Server access style logs (client/dest IP, port, method, uri, bytes, etc.)
    - Created Hourly, when there is activity (typically created 15 minutes after the end of the hour)
  - Storage Logs
    - Provide info about the storage size (in “byte_hours”) of buckets per 24 hour period
    - Created Daily with previous day’s info (typically created before 10:00 am PST)
    - Not generally recommended to use - suggested to use **Monitoring** -> **Metrics Explorer** instead

RSAConference2020

Google Cloud Platform (GCP)

Configuring Logging

Data Access Logs

- Configure Data Access Logs (logging per Service)
  - Navigate to **IAM & Admin** -> **Audit Logs**
  - Select the appropriate Project/Folder/Organization
  - Select a **Service**
  - Turn on/off the following logging for the selected **Service**:
    - **Admin Read**
    - **Data Read**
    - **Data Write**
  - Click **Save**

Data Access Logs

- Configure Data Access Logs (default logging for All New/Existing Services)
  - Navigate to **IAM & Admin** -> **Audit Logs**
  - Select the appropriate Project/Folder/Organization
  - Click **Default Audit Config**
  - Turn on/off the following logging for the **All Services**:
    - **Admin Read**
    - **Data Read**
    - **Data Write**
  - Click **Save**

Application Logs

- Stackdriver Logging Agent

  *Note: Installed by default on VM's running in **Google Kubernetes Engine** or **App Engine**

  - Installing the Agent
    - Linux (via Command-Line)

```
$ curl -sSO https://dl.google.com/cloudagents/install-logging-agent.sh
$ sudo bash install-logging-agent.sh
```

      - (Optional) - Edit Proxy config in `/etc/default/google-fluentd` to export `http_proxy`, `https_proxy`, and `no_proxy` environment variables

```
$ sudo service google-fluentd restart
```

Application Logs

- Stackdriver Logging Agent
  - Installing the Agent
    - Windows (via Command Line)
      - (Optional) – Export proxy variables via Admin Command Prompt

```
> setx http_proxy http://<PROXY_IP>:<PROXY_PORT> /m
> setx https_proxy http://<PROXY_IP>:<PROXY_PORT> /m
> setx no_proxy 169.254.169.254 /m
```

      - Open PowerShell terminal (No Admin Needed)

```
> cd $env:UserProfile;
> (New-Object Net.WebClient).DownloadFile("https://dl.google.com/cloudagents/windows/StackdriverLogging-v1-10.exe", ".\StackdriverLogging-v1-10.exe")
> .\StackdriverLogging-v1-10.exe /S /D="C:\Preferred\Install\Dir\"
```

URL may change over time

Specify Silent Install

Set Install Dir

Application Logs

- Stackdriver Logging Agent
  - Installing the Agent
    - Windows (via GUI)
      - Simply download + install the Stackdriver Logging Agent executable

Application Logs

- Stackdriver Logging Agent
  - Configuring the Agent
    - "The Logging agent comes with a default configuration; in most common cases, no additional configuration is required." (YMMV)
      - Due to GCP's implementation/inclusion of a `fluentd-catch-all-config`
    - Agent configuration files locations:
      - Linux

        `/etc/google-fluentd/google-fluentd.conf`
      - Windows

        `C:\Program Files (x86)\Stackdriver\LoggingAgent\fluent.conf`

Application Logs

- Stackdriver Logging Agent
  - Customizing the Agent to collect additional (non-standard) logs
    - Create a new config file (e.g. `new-log.conf`) within the following directory:
      - Linux

        `/etc/google-fluentd/config.d/`
      - Windows

        `C:\Program Files (x86)\Stackdriver\LoggingAgent\`
    - Set the appropriate `path`, `format`, `tag`, … in the config file
    - Restart the service

Container (GKE) Logs

- Stackdriver Logging for Kubernetes (GKE)
  - Metrics (CPU/Mem Utilization, Incidents, etc.) for GKE Clusters/Nodes
  - Configuring Stackdriver (New Cluster)
    - Navigate to **Kubernetes Engine** -> **Clusters**
    - Click Create Cluster
    - Click **Availability, networking, security, and additional features**
    - Select Enable Stackdriver Kubernetes Engine Monitoring
    - Click **Create**
  - Configuring Stackdriver (Existing Cluster)

Container (GKE) Logs

- Stackdriver Logging for Kubernetes (GKE)
  - Configuring Stackdriver (Existing Cluster)

    *Requires cluster to version 1.12.7 or higher (will need to manually upgrade if not)

    - Navigate to **Kubernetes Engine** -> **Clusters**
    - Click the **Edit (pencil)** icon on the appropriate Cluster
    - In the **Stackdriver Kubernetes Engine Monitoring** drop down, select **Enabled**
    - Click **Save**
  - (Optional) Configuring Prometheus Monitoring Support
    - Stackdriver configured as sidecar, exports metrics as “External Metrics”

Container (GKE) Logs

- Enabling Auditd Logs on GKE Nodes
  - Provides OS/Host-level auditing logs (errors, logins, binary execution, etc.) to provide info on the state of your cluster/workloads
  - Requires use of a Kubernetes DaemonSet**

    **Works only on nodes running Container-Optimized OS

    - Manages groups of replicated Pods
    - Runs one Pod on each cluster node with 2 Containers to configure auditd:
      - First is an `init-container` that starts the cloud-audit-setup systemd service
      - Second is `fluentd-gcp-cos-auditd` Container that configures auditd

Container (GKE) Logs

- Enabling Auditd Logs on GKE Nodes
  - Configuring Auditd Logging (per Cluster)**

    **As always with configuring auditd – *be aware of performance implications!*

    - Download the example manifests

      ```
      $ curl https://raw.githubusercontent.com/GoogleCloudPlatform/k8s-node-tools/master/os-audit/cos-auditd-logging.yaml > cos-auditd-logging.yaml
      ```

    - Deploy the logging DaemonSet and ConfigMap

      ```
      $ kubectl apply -f cos-auditd-logging.yaml
      ```

    - Verify logging pods have started

      ```
      $ kubectl get pods --namespace=cos-auditd
      ```

VPC Flow Logs

- Configuring VPC Flow Logs (per Subnet*)

  *Note: VPC Flow logs may only be enabled per-Subnet

  - New Subnet
    - Navigate to **Networking** -> **VPC Networks**
    - Select the appropriate **Network**
    - Click **Add Subnet**
    - Under **Flow Logs**, select **On**
    - Click **Configure Logs** to set **Aggregation Interval**, **Include Metadata**, and **Sample rate**
    - Click **Add**

VPC Flow Logs

- Configuring VPC Flow Logs (per Subnet*)

  *Note: VPC Flow logs may only be enabled per-Subnet

  - Existing Subnet
    - Navigate to **Networking** -> **VPC Networks**
    - Select the appropriate **Subnet**
    - Under **Flow Logs**, select **On**
    - Click **Configure Logs** to set **Aggregation Interval**, **Include Metadata**, and **Sample rate**
    - Click **Add**

Cloud Storage Logs

- Configure Log Delivery for Access and Storage Logs

  *Requires use of `gsutil` tool (or XML/JSON API's)

  – Create a Bucket to store the logs (if not already created)

  ```
  $ gsutil mb gs://example-logs-bucket
  ```

  – Configure Bucket to allow Cloud Storage `WRITE` permissions

  ```
  $ gsutil acl ch -g cloud-storage-analytics@google.com:W
  gs://example-logs-bucket
  ```

  – (Optional) Configure default object ACL

  ```
  $ gsutil defacl set project-private gs://example-logs-bucket
  ```

Cloud Storage Logs

- Configure Log Delivery for Access and Storage Logs
  - Enable Logging for each Bucket in scope

```
$ gsutil logging set on -b gs://example-logs-bucket [-o
log_object_prefix ] gs://example-bucket
```

    - Optionally can specify `log_object_prefix`
    - By default, the object prefix is the name of the bucket for which the logs are enabled

Exporting Logs

- Can export logs to 3 destination types:
  - Cloud Storage Bucket (for simple retention)
  - BigQuery Datasets (to stage for queries/investigations)
    - Ideal for native investigation and response capabilities
  - Pub/Sub Topics (to send to another application/SIEM)
    - Useful if you're using a separate/dedicated SIEM for log retention, monitoring, and querying

Exporting Logs

- Exporting Logs to BigQuery with Log Viewer

  *You can also use the `gloud` tool or Stackdriver Logging API

  - Per-Project Sink (All Logs, No Filtering)
    - Navigate to **Stackdriver** -> **Logging** -> **Logs Router**
    - Click **Create Sink**
      - Enter **Sink Name**
      - Select **BigQuery** as the **Sink Service**
      - Select **Use Partitioned Tables**
      - For **Sink Destination**, select **Create New BigQuery Dataset**
      - Enter the **BigQuery Dataset Name** and click **Create**
      - Click **Create Sink**

Exporting Logs

- Exporting Logs to BigQuery with Log Viewer
  - Organization-Level Sink (Aggregate Sink of all Admin Activity)

```
$ gcloud logging sinks create my-bq-sink
bigquery.googleapis.com/projects/my-project/datasets/my_dataset
--log-filter='logName: "logs/cloudaudit.googleapis.com%2Factivity"'
--organization=<org_ID> --include-children
```

Exporting Logs

- Exporting Logs to BigQuery with Log Viewer
  - Folder-Level Sink (Aggregate Sink of all Data Access Activity)

```
$ gcloud logging sinks create my-bq-sink
bigquery.googleapis.com/projects/my-project/datasets/my_dataset
--log-filter='logName: "logs/cloudaudit.googleapis.com%2Fdata_access"'
--folder=<folder_ID> --include-children
```

Log Sink Cheat Sheet

Log Types Supported by the GCP Sensor

Log Type	Filter to Capture This Log	Notes
Audit Logs at the Organization Level	organizations/<organization-id>/logs/cloudaudit.googleapis.com	To filter these logs further, append: • **%2Factivity**: For activity logs • **%2Fdata_access**: For data access logs • **%2Fsystem_event**: For system events
Audit Logs at the Project Level	projects/<project-id>/logs/cloudaudit.googleapis.com	To filter these logs futher, append: • **%2Factivity**: For activity logs • **%2Fdata_access**: For data access logs • **%2Fsystem_event**: For system events
VPC Flow Logs	projects/<project-id>/logs/compute.googleapis.com%2Fvpc_flows	
Firewall Logs	projects/<project-id>/logs/compute.googleapis.com%2Ffirewall	
Syslog	projects/<project-id>/logs/syslog	These logs are delivered via the Stackdriver logging agent
Apache Logs	projects/<project-id>/logs/apache	• **-access**: For access logs • **-error**: For error logs
Nginx Logs	projects/<project-id>/logs/nginx	• **-access**: For access logs • **-error**: For error logs

Source Link

RSAConference2020

Google Cloud Platform (GCP)

Tips for Monitoring

Stackdriver Monitoring/Alerting

- Utilize Stackdriver Monitoring to create alerts
  - Metrics-Based Alerts
    - Create Alerts based on:
      - High CPU Usage (bitcoin miner? ransomware encryption?)
      - High Memory Usage (resource exhaustion?)
      - Uptime (something recently rebooted? why?)
  - Application Log-Based Alerts
    - Gratuitous 404 errors

Using Stackdriver Logs Viewer for Investigations

- Utilize Stackdriver Logs query service to perform regular queries for anomalies
- Define log(s) to search:

```
log_name:"/logs/cloudaudit.googleapis.com%2Factivity" AND...
log_name:"/logs/cloudaudit.googleapis.com%2Fdata_access" AND...
log_name:"/logs/cloudaudit.googleapis.com%2Fsystem_event" AND...
```

- Search a specific resource:

```
logName:"projects/[PROJECT_ID]/logs" AND
resource.type=[RESOURCE_TYPE] AND
resource.labels.instance_id=[INSTANCE_ID]
```

Using Stackdriver Logs Viewer for Investigations

- Perform targeted searches
  - HTTP Error Logs

```
resource.type="gae_app" AND proto_payload.status >= 400 AND
sample(insertId, 0.1)
```

  - Service Account Creation

```
resource.type="service_account" AND
log_name="projects/[PROJECT_ID]/logs/cloudaudit.googleapi
s.com%2Factivity" AND
proto_payload.method_name="google.iam.admin.v1.CreateServ
iceAccount”
```

Using Stackdriver Logs Viewer for Investigations

- Perform targeted searches
  - Firewall Rule Deletion

```
resource.type="gce_firewall_rule" AND
log_name="projects/[PROJECT_ID]/logs/cloudaudit.googleapis.com%2Fact
ivity" AND proto_payload.method_name:"firewalls.delete"
```

  - Bucket Creation

```
resource.type="gcs_bucket" AND
log_name="projects/[PROJECT_ID]/logs/cloudaudit.googleapis.com%2Fact
ivity" AND proto_payload.method_name="storage.buckets.create"
```

Using Stackdriver Logs Viewer for Investigations

- Perform targeted searches
  - All Inbound SSH Activity (VPC Flow Logs)

```
resource.type="gce_subnetwork" AND
log_name="projects/[PROJECT_ID]/logs/compute.googleapis.com%2Fvpc_fl
ows" AND json_payload.connection.dst_port=”22"
```

GKE Monitoring

- Native Tooling
  - Stackdriver Kubernetes Engine Monitoring
    - Dashboard interface to your Kubernetes Clusters
    - View alerts, metrics, logs, and details surrounding them
    - Can view by Aggregation categories:
      - Infrastructure (Aggregate by Cluster -> Node -> Pod -> Container)
      - Workloads (Aggregate by Cluster -> Namespace -> Workload -> Pod -> Container)
      - Service (Aggregate by Cluster -> Namespace -> Service -> Pod -> Container)

GKE Monitoring

- Native(ish*) Tooling
  - Prometheus
    - *Technically third-party, but GCP has built a Stackdriver Prometheus sidecar
    - Utilize standard Monitoring console's Metrics Explorer
    - Select Kubernetes Container as Resource Type
    - Specify external Metric fields with "external/prometheus/" prefix

GKE Monitoring

- Third-Party Tooling
  - Falco
    - Dedicated security auditing/monitoring solution for Kubernetes
    - "Falco lets you continuously monitor and detect container, application, host, and network activity, all in one place, from one source of data, with one set of rules."
    - Behavior monitoring/analytics (via SysCall monitoring) to help identify/alert when:
      - A shell is run inside a container
      - A server process spawns a child process of an unexpected type
      - A sensitive file, like /etc/shadow, is unexpectedly read
      - A non-device file is written to /dev
      - A standard system binary (like ls) makes an outbound network connection

Using BigQuery for Investigations

- Query BigQuery DataSets established previously
  - Utilize Log Sinks to aggregate/segregate certain types of data into certain DataSets (i.e. Tables) as the source(s) for queries
- Can run Active and Scheduled Queries
  - Manually run queries if/when needed
  - Run Scheduled Queries and regularly review results

Using BigQuery for Investigations

- Identify Virtual Machine Deletions in Activity Logs

```
SELECT timestamp, resource.labels.instance_id,
protopayload_auditlog.authenticationInfo.principalEmail,
protopayload_auditlog.resourceName, protopayload_auditlog.methodName

FROM (TABLE_DATE_RANGE(
[PROJECT].[DATASET].cloudaudit_googleapis_com_activity,
DATE_ADD(CURRENT_TIMESTAMP(),-7,'DAY'), CURRENT_TIMESTAMP()) )

WHERE resource.type = "gce_instance" AND operation.first IS TRUE AND
protopayload_auditlog.methodName = "v1.compute.instances.delete"

ORDER BY timestamp, resource.labels.instance_id

LIMIT 1000
```

Using BigQuery for Investigations

- Identify Most Common Actions in Data Access Logs

```
SELECT protopayload_auditlog.methodName, resource.type, COUNT(*) AS
counter

FROM (TABLE_DATE_RANGE(
[PROJECT].[DATASET].cloudaudit_googleapis_com_data_access,
DATE_ADD(CURRENT_TIMESTAMP(),-30,'DAY'), CURRENT_TIMESTAMP()) )

GROUP BY protopayload_auditlog.methodName, resource.type

ORDER BY COUNTER DESC

LIMIT 1000
```

In Conclusion…

(TL;DR)

TL;DR

There is no TL;DR…

Too. Much. Material.

How Can You Apply This Starting Right Now?

- Next week you should:
  - Begin getting familiar with the core logs in each provider
    - I'd suggest assigning one (or more) SME's to each Cloud
    - Or accept that one person is about to be extremely busy form here on out…
  - Start poking around the Consoles and playing with configurations
  - Start identifying and testing multiple access and logging configuration methods
    - Console
    - CLI
    - Custom (and/or Open Source) Scripts

How Can You Apply This Starting Right Now?

- In the first three months following this presentation you should:
  - Have the core logs enabled and centralized
  - Begin testing and verifying the log configurations and contents:
    - How easy is it to access the logs?
    - Do the logs contain all the information needed to perform comprehensive investigations?
    - If not… (in this order)
      - How can those gaps be addressed with native tooling?
      - How can those gaps be address with third-party tooling?
    - Do we have an effective and efficient way to aggregate and analyze the logs?

How Can You Apply This Starting Right Now?

- Within six months you should:
  - Identify any gaps in log collection methodologies and/or content
  - Have a roadmap for fixing the identified gaps
  - Be planning several tabletop exercises to test your logging configuration, content, and access with real-world scenarios
    - Compromised Access Key
    - Compromised Instance(s) involving SSRF
    - Unauthorized S3 Data Access/Transfer
    - DDoS
    - …
  - Get creative – you know what needs testing

The End

Please feel free to reach out!

Email: jpoling@secureworks.com
Twitter: @JPoForenso
Blog: https://www.ponderthebits.com