Incident Response and Forensics in your Pyjamas

When security incidents happen, you often have to respond in a hurry to gather forensic data from the resources that were involved. You might need to grab a bunch of hard drives and physically visit the data centre to capture data from the systems. And that would mean getting dressed. When infrastructure is in the cloud, you have remote access and APIs for managing all your infrastructure, so you can respond to incidents with automation and do your forensic analysis in your bunny slippers. But is it as good as the capabilities you have in a data centre? Is getting dressed the price you have to pay for high quality forensics and incident response? In this talk Paco will explain the two major domains of cloud events (infrastructure domain and service domain) and describe the security and incident response techniques pioneered by AWS customers like Mozilla, Alfresco, and Netflix. He'll explain how to isolate resources to preserve the integrity of the data; get RAM dumps and disk image snapshots; and identify unauthorised changes to cloud resources using API tools and logs. And all of this while wearing pyjamas.

What Exactly Is Cloud Computing ?

"Cloud computing" is a term broadly used to define the **on-demand** delivery of IT resources and applications **via the Internet**, with **pay-as-you-go pricing**.

For more on this, see: https://aws.amazon.com/what-is-cloud-computing/

What Exactly Are Pyjamas?

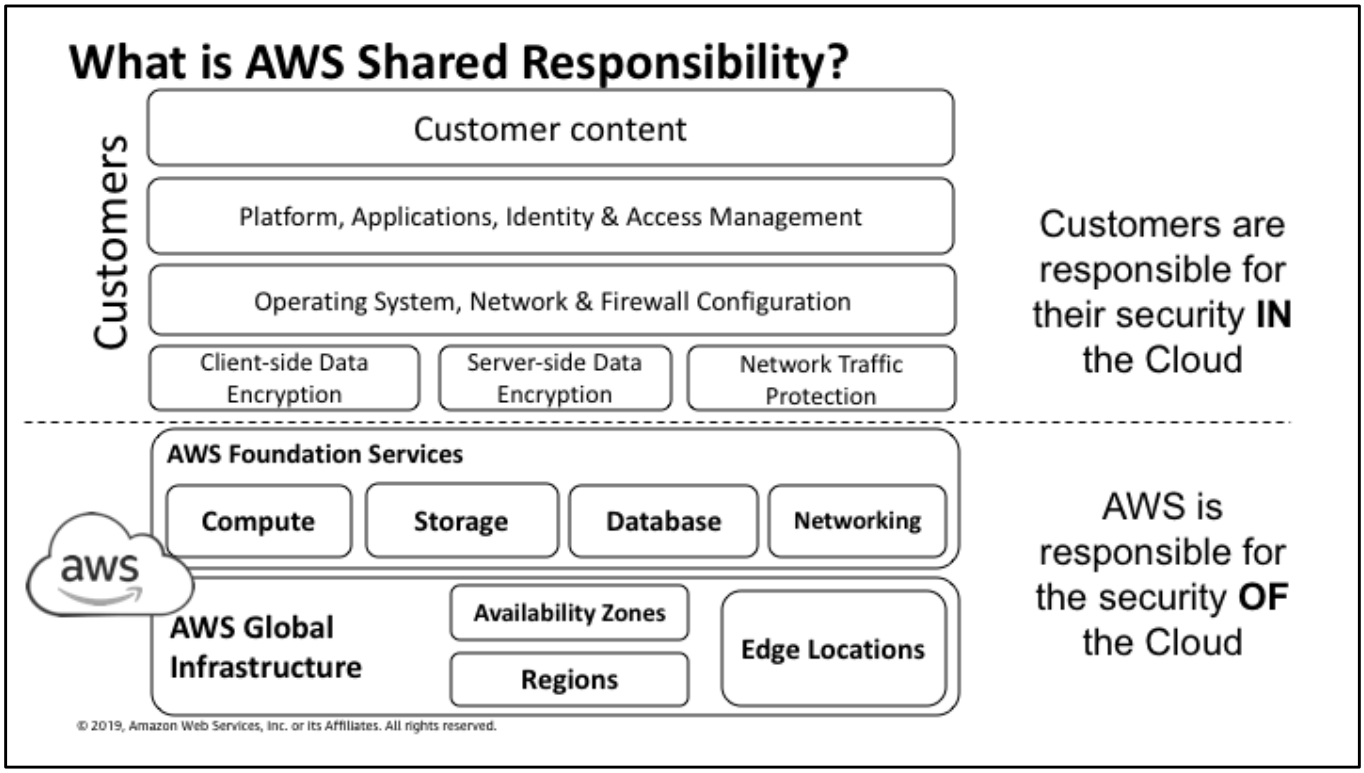
- Comfortable, loose-fitting clothes
- Imported concept from the East in the 19th century
- Worn at bed time (or while working from home)
- Not appropriate attire for data centres

Photo © 2005 monicasecas on Flickr CC BY 2.0

Photo “lazy day” by monicasecas

Image Source:
https://www.flickr.com/photos/monicasecas/3171587103

License: **Attribution 2.0 Generic (CC BY 2.0)**
https://creativecommons.org/licenses/by/2.0/



This shared model can help relieve customer's operational burden as AWS operates, manages and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the service operates.

Four Kinds of Security Controls

Directive

- What are our security goals?

Preventative

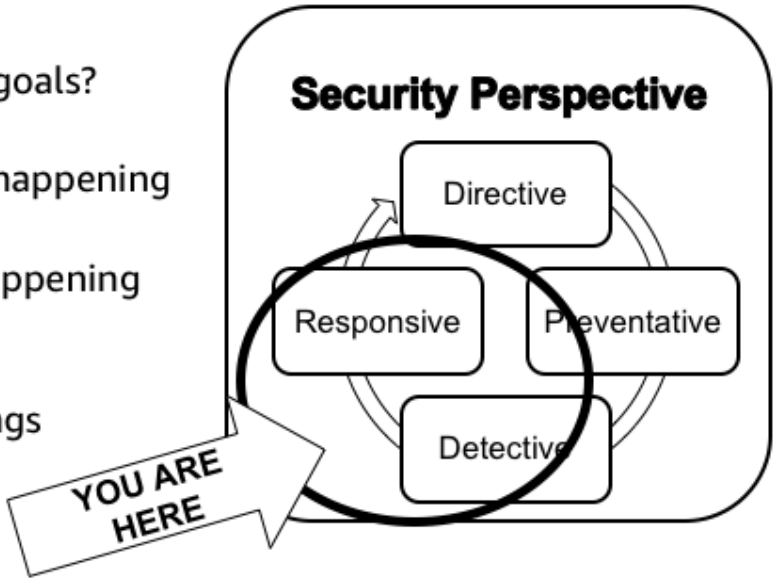
- Stop bad things from happening

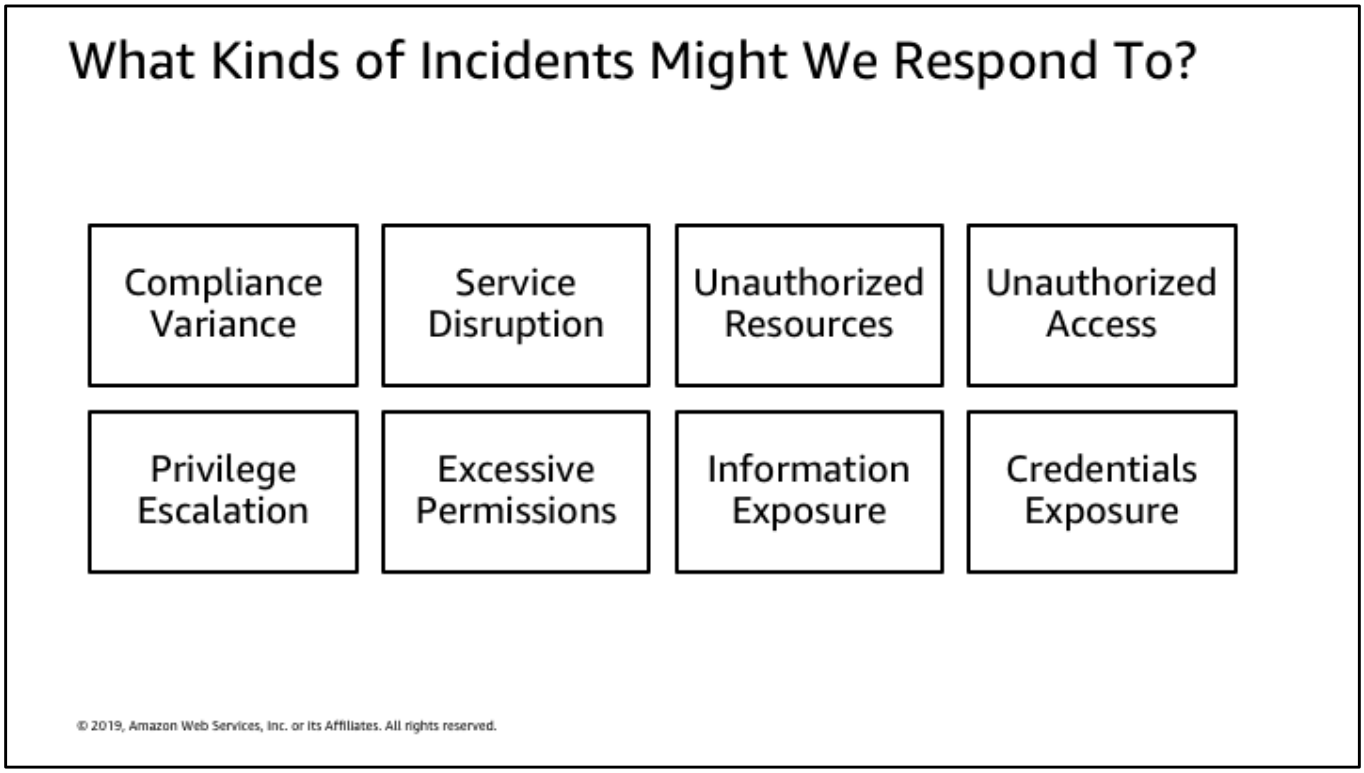
Detective

- Look for bad things happening anyway

Responsive

- Fix or alert on bad things detected





Compliance variance — data or resources configured in a way that violates your compliance policies

Service disruption — users or systems unable to access resources in your environment

Unauthorized resources — resources created in your environment that are unauthorized or unexpected

Unauthorized access — access to your resources via an IP address, user, or system, that is unauthorized

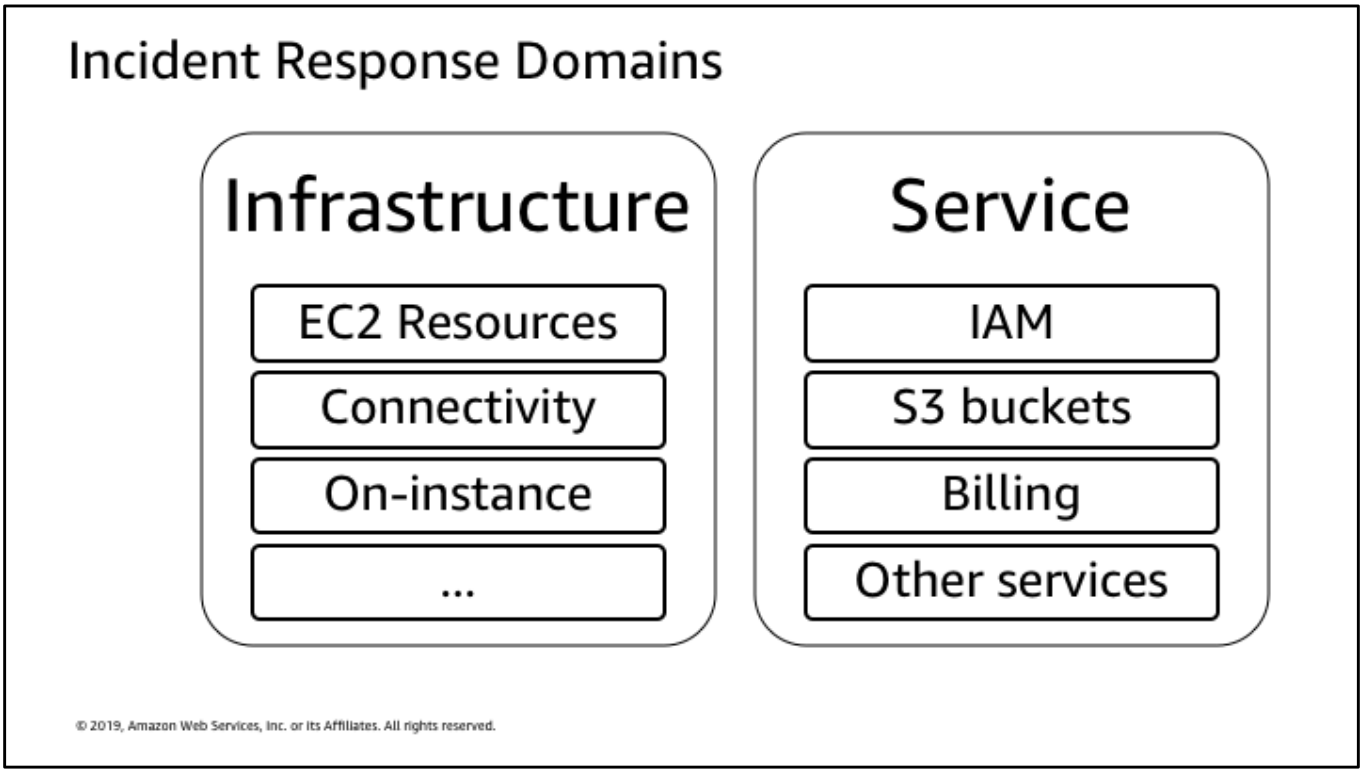
Privilege escalation — attempts to gain elevated access to resources that are normally protected from an application or user, or attempts to gain access to your system or network for an extended period of time

Persistence — attempts to establish an access mechanism allowing future access to resources

Excessive permissions — resources that have overly permissive access control mechanisms or permissions

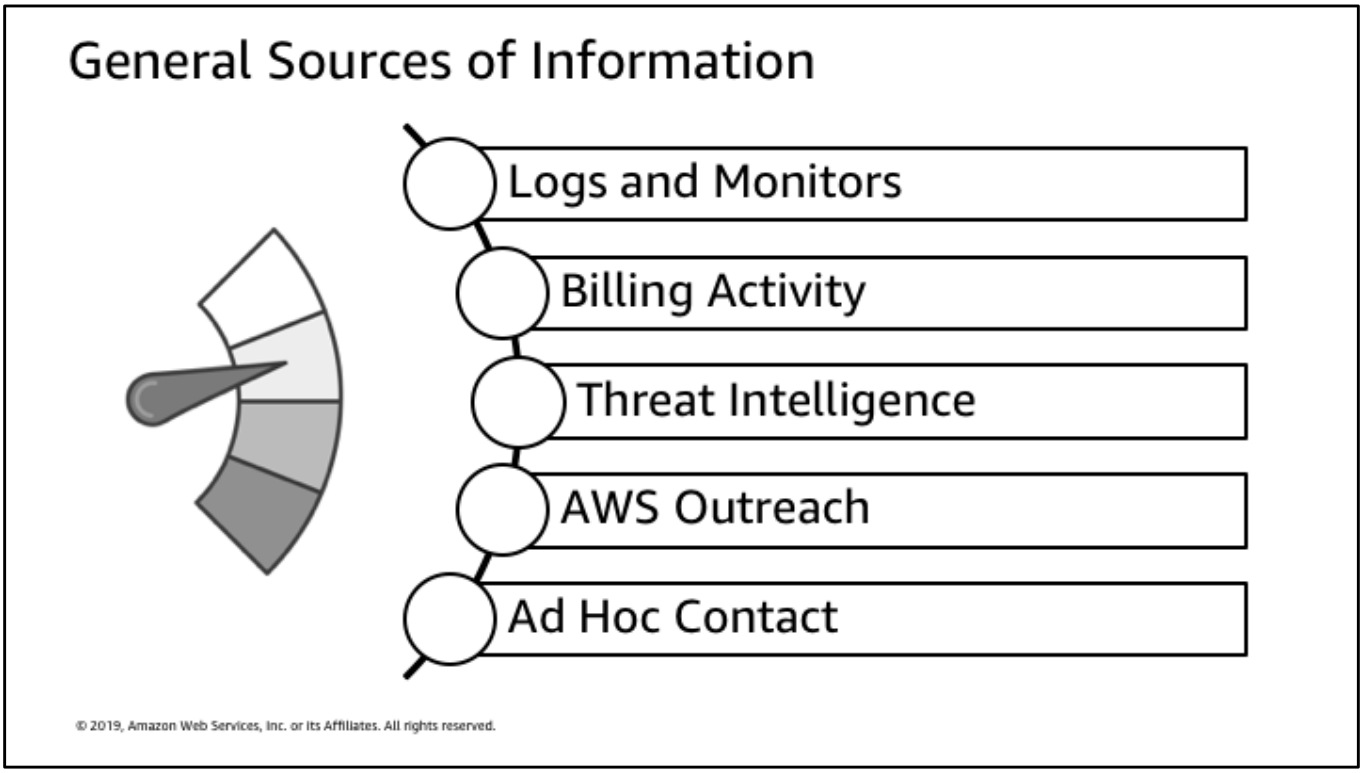
Information exposure — anomalous or unauthorised access to sensitive data

Credentials exposure — unauthorized access to AWS-specific credentials



Service Domain: Events in the service domain affect a customer's AWS account, billing, IAM permissions, resource metadata, etc. A service domain event is one that you respond to exclusively with AWS API mechanisms.

Infrastructure Domain: Events in the infrastructure domain include data or network-related activity, such as the traffic to your Amazon EC2 instances within the VPC, processes and data on your Amazon EC2 instances, etc. Your response to infrastructure domain events will often involve commands and software that executes in the operating system of an instance, but may also involve AWS API mechanisms where they can be applied.



Logs and Monitors — There are AWS logs like Amazon CloudTrail, S3 access logs, and VPC Flow Logs and security monitoring services such as Amazon GuardDuty and Amazon Macie. There are also AWS monitors like Route 53 health checks and CloudWatch Alarms. Similarly there are the Windows Events, Linux syslog logs, and other application-specific logs that you might generate in your applications.

Billing Activity — A sudden change in billing activity may indicate a security event.

Threat Intelligence — if you subscribe to a third-party threat intelligence feed, you can correlate that information with other logging and monitoring to identify potential indicators of events.

AWS Outreach — AWS Support may contact you if we identify abusive or malicious activity. See the following section, "*AWS Response to Abuse and Compromise",* for additional information.

Ad Hoc Contact — Sometimes your customers, your developers, or other staff in your organization notice something unusual. It is important to have a well-known, well-publicized "front door" for your security team to receive notifications from people. Popular choices include ticketing systems, contact email addresses,and web forms. If your organization deals with the general public, you may need to have a public-facing security contact mechanism as well.

Programmatic, Cloud-Native Sources

CloudWatch Logs — a number of services utilize CloudWatch Logs for their logging mechanism, such as Lambda. You can create filters to look for patterns in CloudWatch Logs and create CloudWatch metrics with them to trigger downstream actions
CloudWatch Events — A place that you can "listen" for specific events happening on AWS such as EC2 instance states or a specific API call. When an event is matched, it can trigger Lambda or SNS (or both)
CloudWatch Alarms — An alarm can trigger a workflow via SNS that can chain up a number of reactions including Lambda
VPC Flow Logs — Can be a telling source of nefarious network access
GuardDuty — A powerful threat detection tool that can notify you when something is off of baseline
Macie — Macie can notify you when sensitive data in S3 is being moved or accessed. Additionally, Macie reports all findings to CloudWatch.
S3 access logs — These logs could indicate possible data exfiltration
CloudTrail — CloudTrail logs all API calls on the platform, who called them, when, if it was successful or not, etc.

This is not an exhaustive list. Consider other things like:
- Health checks (ELB, Route53)
- OS and Application logs

Responding to Incidents

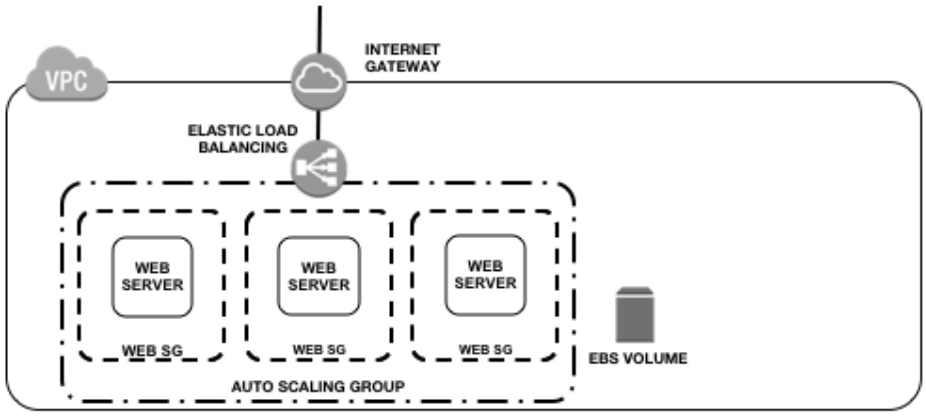
INFRASTRUCTURE DOMAIN

Two options for forensic analysis in the infrastructure domain:

- Online analysis
- Offline analysis

You can do either or both

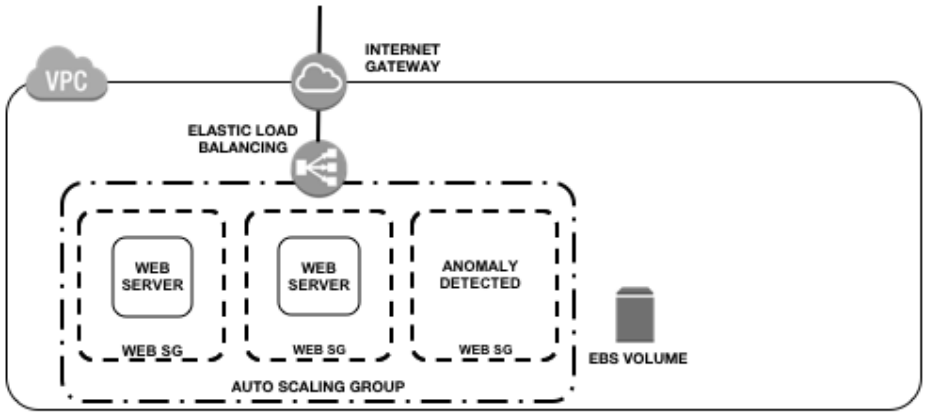
Example of Isolating an EC2 Instance



For demonstration, this is a simplified example VPC with an auto-scaling web tier. Each instance has an EBS volume attached.

So we have a VPC… web tier, each instance has a data volume

Example of Isolating an EC2 Instance

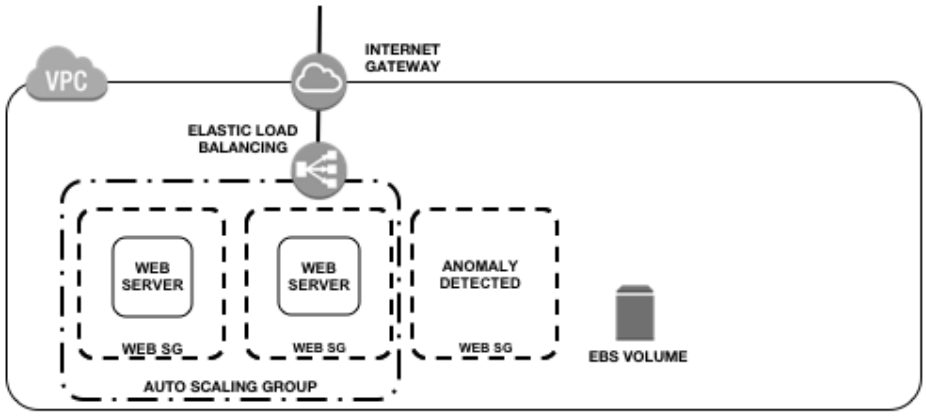


If your monitoring tools detect a security anomaly, you should respond. First, gather info on the instance and tag it for quarantine.

```
$ aws ec2 describe-instances --filters "Name=ip-address,Values=12.34.56.78"
$ aws ec2 create-tags --resources i-abcd1234 \
    --tags Key=Environment,Value=Quarantine:REFERENCE-ID
```


Let's assume that we're sending logs to S3, as well as running some monitoring tools

Example of Isolating an EC2 Instance

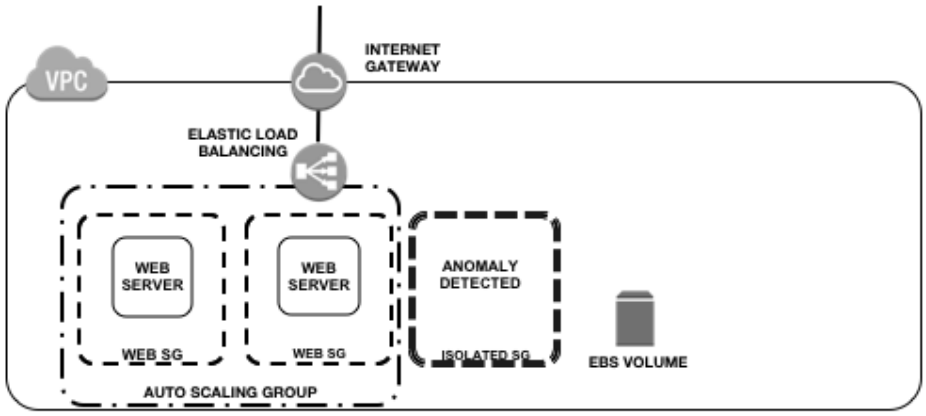


Before isolating the EC2 instance, remove it from the Auto-Scaling Group and the Elastic Load Balancer.

```
$ aws autoscaling detach-instances --instance-ids i-abcd1234 \
    --auto-scaling-group-name web-asg
$ aws elb deregister-instances-from-load-balancer --instances i-abcd1234 \
    --load-balancer-name my-load-balancer
```


Change the security group around it to stop the horizontal movement around the compromised instance.

Example of Isolating an EC2 Instance

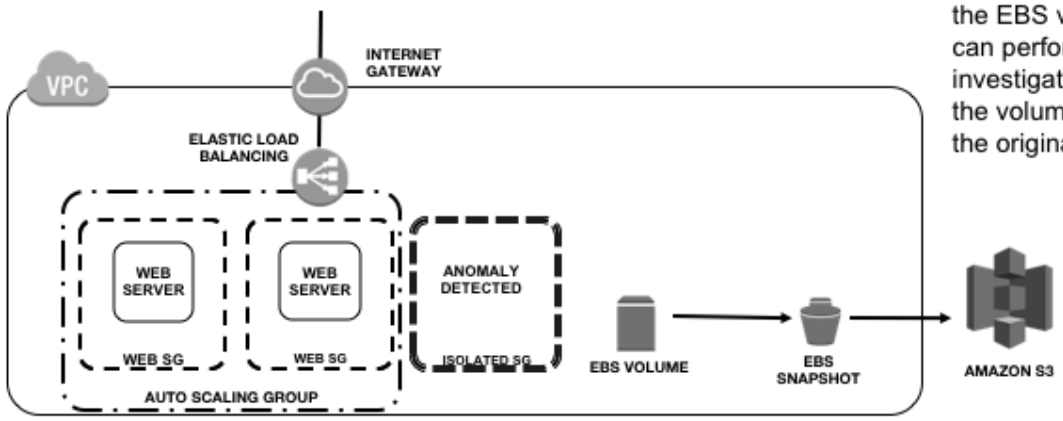


You will then change the instance's security group (SG) to an isolated SG. Also, you can mark the attribute to protect the instance from accidental termination.

```
$ aws ec2 modify-instance-attribute --instance-id i-abcd1234 --groups sg-a1b2c3d4
$ aws ec2 modify-instance-attribute --instance-id i-abcd1234 \
    --attribute disableApiTermination --value true
```


You can leave it and use it as a honeypot, but make sure the rest of your users aren't accessing it.

Example of Isolating an EC2 Instance

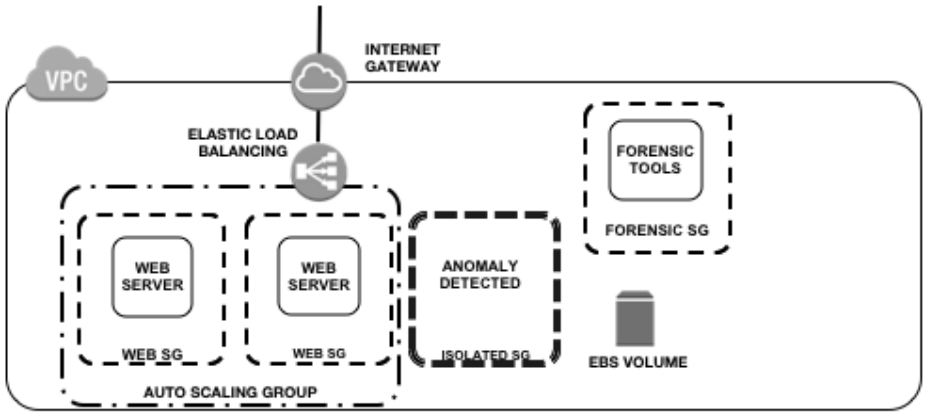


You should then snapshot the EBS volume so we can perform an investigation on a copy of the volume, rather than the original evidence.

```
$ aws ec2 create-snapshot --volume vol-12xxxx78 \
    --description "ResponderName-Date-REFERENCE-ID"
```


Take a snapshot

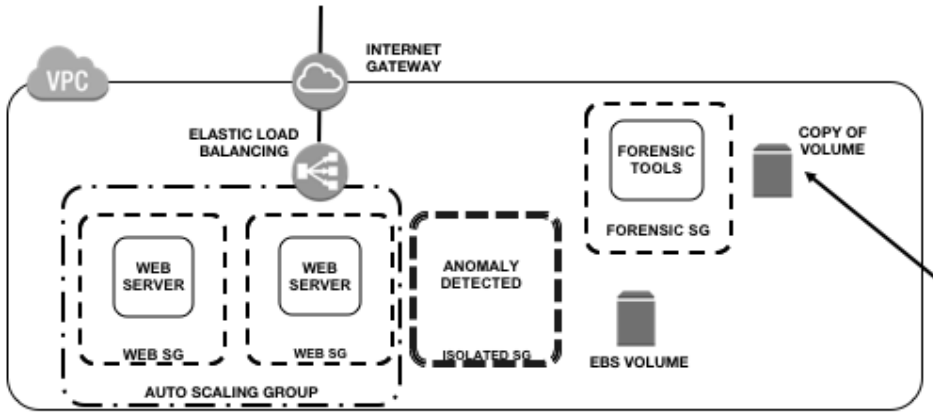
Example of Isolating an EC2 Instance



You will need a new forensic workstation, so let's launch that from our forensic Amazon Machine Image (AMI).

```
$ aws ec2 run-instances --image-id ami-4n6x4n6x --count 1 \
    --instance-type c4.8xlarge --key-name forensicKey \
    --security-group-ids sg-903004f8 --subnet-id subnet-a1b2c3d4
```

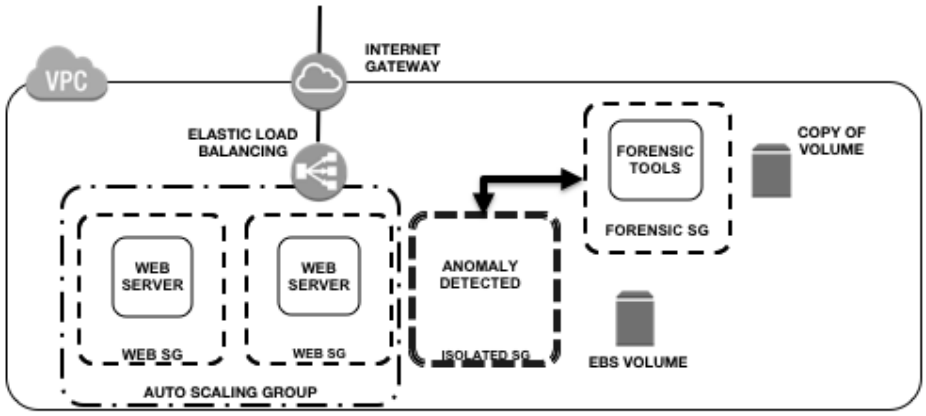
Example of Isolating an EC2 Instance




```
$ aws ec2 create-volume --region us-east-1 --availability-zone us-east-1a \
    --snapshot-id snap-abcd1234 --volume-type io1 --iops 10000
$ aws ec2 attach-volume --volume-id vol-1234abcd --instance-id i-new4n6x\
     --device /dev/sdf
```


Load the snapshot onto your forensic instance

Example of Isolating an EC2 Instance



If you need access to the original instance for a memory dump, you can authorize communication between the isolated security group and the forensic security group.

```
$ aws ec2 authorize-security-group-ingress --group-id sg-a1b2c3d4 \
    --protocol tcp --port 0-65535 --source-group sg-1a2b3c4d
```


From there, you can connect the two and run the forensics and find the threats and patterns to help you with your investigation.

…but that sounds like work…

aws professional services

I don't want to do work

Anything I can run as a **series of commands**, I can turn into a **shell script**

Anything I can write as a **shell script**, I can turn into a **python program with boto3**

Anything I can write as a **python program** I can turn into a **lambda function**

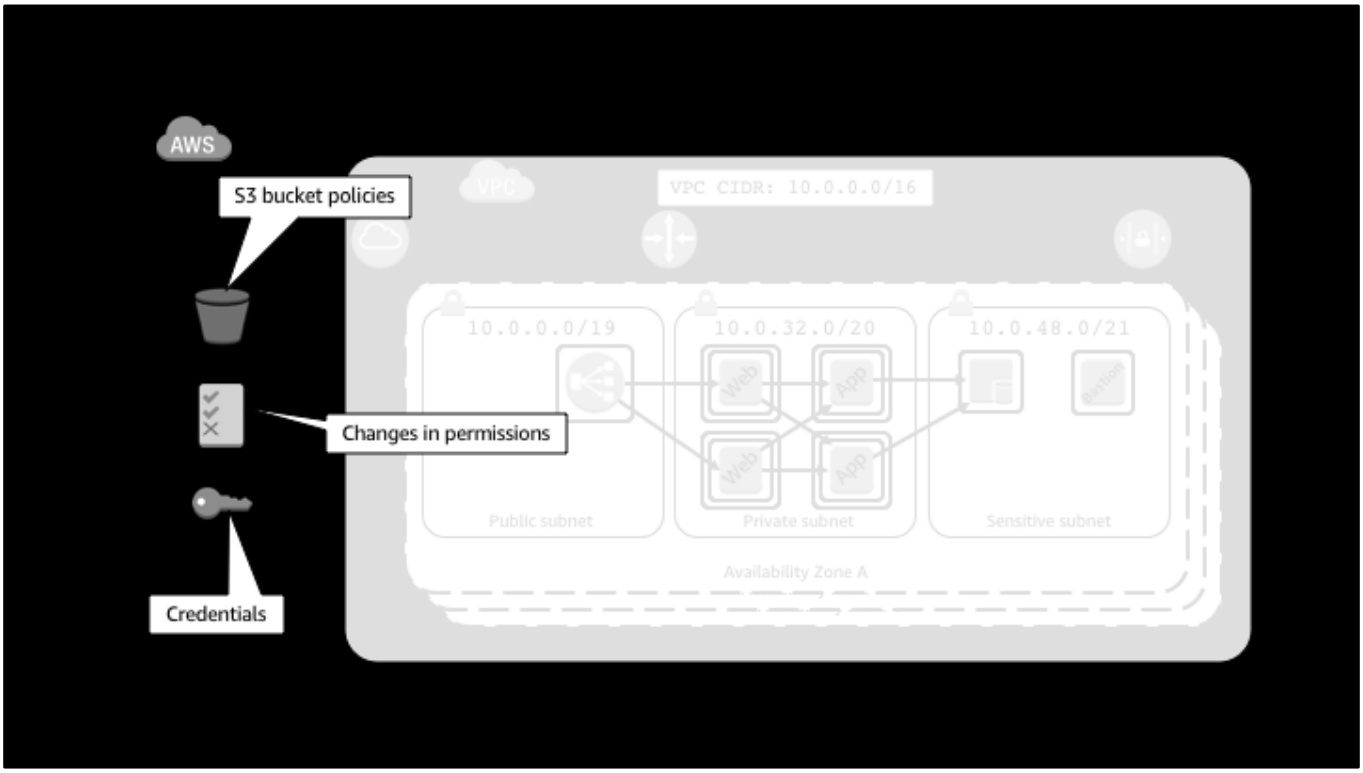
Anything I can run as a **lambda function** I can **trigger automatically** in response to events.

- ✓ Incident response
- ✓ In pyjamas
- ✓ Asleep!

Photo "Sleeping Woman" by Petr Kratochvil
Source: https://www.publicdomainpictures.net/en/view-image.php?image=208413&picture=sleeping-woman
License: CC0 Public domain: http://creativecommons.org/publicdomain/zero/1.0/

Responding to Incidents

SERVICE DOMAIN


AWS
S3 bucket policies
Changes in permissions
Credentials
VPC
VPC CIDR: 10.0.0.0/16
10.0.0.0/19
10.0.32.0/20
10.0.48.0/21
Web
App
Web
App
Bastion
Public subnet
Private subnet
Sensitive subnet
Availability Zone A

Manual or Automated Access Key Response

```
$ aws_ir key-compromise --access-key-id AKIAINAAAAAAAAAAAAA
2018-07-20T21:04:01 - aws_ir.cli - INFO - Initialization successful proceeding to incident plan.
2018-07-20T21:04:01 - aws_ir.plans.key - INFO - Attempting key disable.
2018-07-20T21:04:03 - aws_ir.plans.key - INFO - STS Tokens revoked issued prior to NOW.
2018-07-20T21:04:03 - aws_ir.plans.key - INFO - Disable complete.  Uploading results.
Processing complete for cr-18-abcdef-01234
Artifacts stored in s3://cloud-response-012345678abcdef0123456789abcdef
```

- Open source toolkit
- Disable Key
- Revoke sessions
- Log details

Responding to Incidents

HOW CUSTOMERS DO IT

Customer #1: Alfresco / Prowler

Alfresco content management platform helps customers like:

- Cisco
- NASA
- Oxford University Press

https://github.com/Alfresco/prowler
https://www.alfresco.com/

Prowler is Alfresco's open-source (github) tool

- Inspects your AWS environment
- Proactively checks against CIS benchmarks
- Reports on security issues in your environment

Customer #2: Mozilla / ThreatResponse

Mozilla foundation (Firefox web browser, Pocket)

- Half a billion people around the world use Firefox
- Firefox is free and open source software, with approximately 40% of its code written by volunteers

https://www.threatresponse.cloud/
https://www.mozilla.org/
https://github.com/ThreatResponse

ThreatResponse IR Toolkit

Open source toolkit provided as both interactive tool AND serverless lambda

Helps a foundation do more with less money. Lambda for the drudgery, humans for the important bits.

In Summary

Responding to incidents

- Infrastructure domain
  - Automate, automate, automate
- Service domain
  - Detect, respond, use native APIs
- Use the native services
- Leverage open source tools

Thank You

Paco Hope
Principal Consultant
AWS
<pacohope@amazon.co.uk>