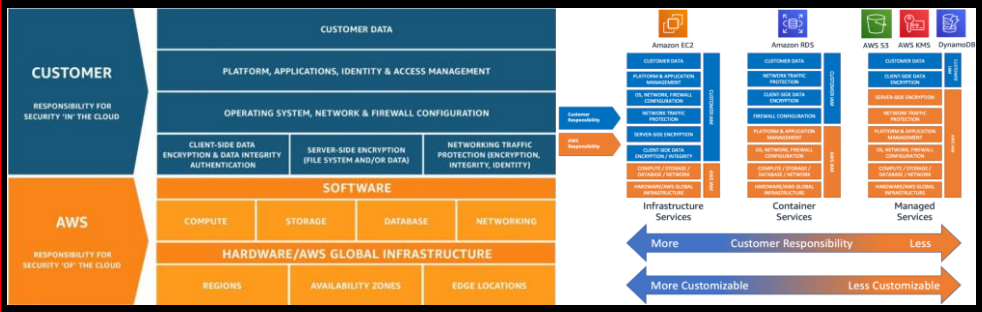
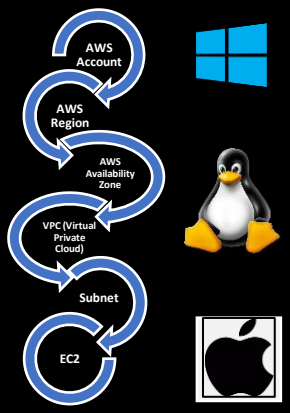
AWS EC2 Incident Response Cheat sheet

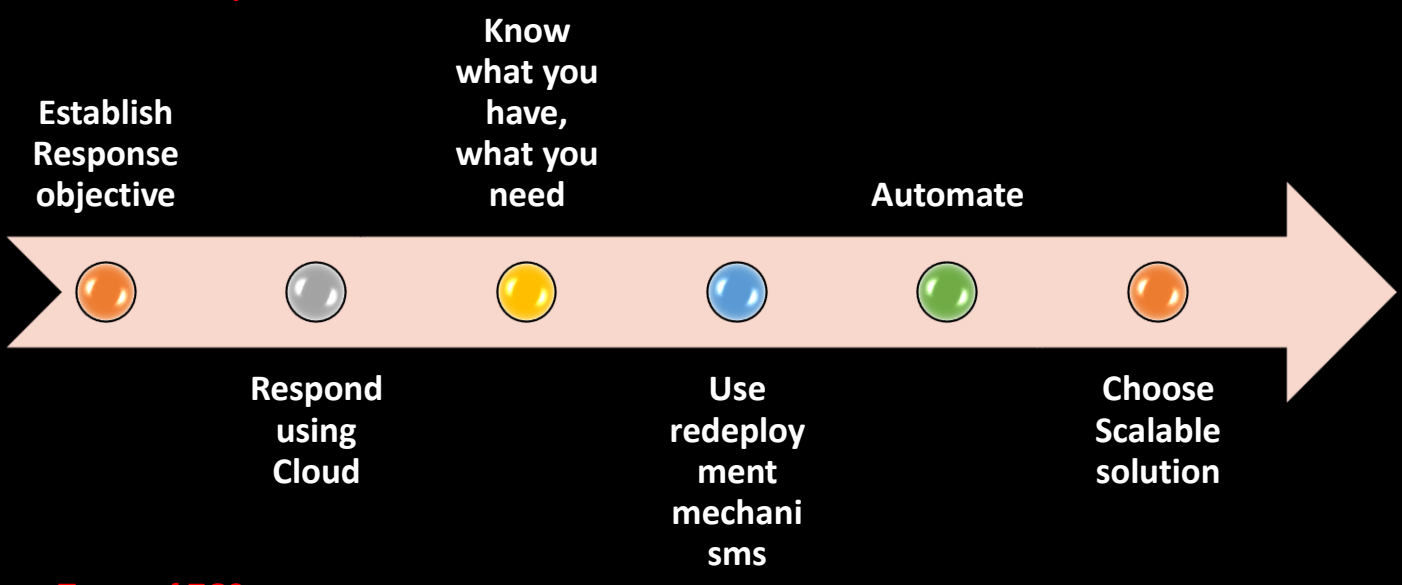
AWS Shared Responsibility Model:



Basics of EC2 (Elastic Compute Cloud) :

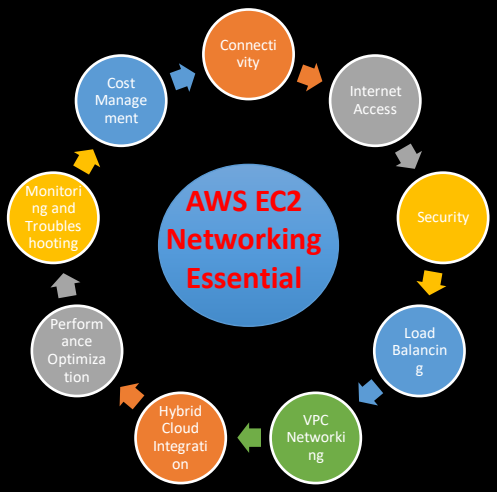


Incident Response Process in Cloud:

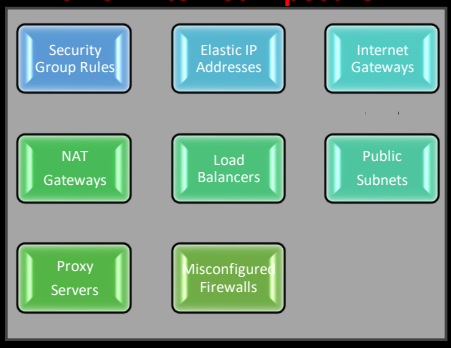


Types of EC2:

General Purpose	Compute Optimized	Memory Optimized	Accelerated Computing	Storage Optimized
A1 – ARM Based Core and custom silicon	C4 – Compute- CPU intensive apps and DBs	R4 – RAM- Memory intensive apps and DBs	P2 – Processing Optimized- Machine Learning	H1 – High Disk Throughput- Big data clusters
T2 – Tiny- Web Servers and small DBs		X1 – Xtreme RAM- For SAP and Spark	G3 – Graphics Intensive- Video and streaming	H1 – IOPS- NoSQL DBs
M4 – Main- App Servers and general purpose		z1d – High Compute and High Memory- Gaming	F1 – Field Programming- Hardware acceleration	D2 – Dense Storage- Data Warehousing



AWS EC2 Internet Exposure:



AWS EC2 Lateral Movement:

1. Lateral Movement within the Same Subnet within the Same VPC:
If an attacker gains access to an EC2 instance within a subnet in a VPC and the subnet's security group rules or Network Access Control Lists (NACLs) allow unrestricted communication within the subnet, the attacker can potentially move laterally to other instances within the same subnet.

2. Lateral Movement between Different Subnets within the Same VPC:
If an attacker compromises an EC2 instance in one subnet of a VPC, and there are overly permissive security group rules or NACLs allowing communication between the subnets within the same VPC, the attacker can traverse to other subnets.

3. Lateral Movement between Different Subnets in Different VPCs:
If multiple VPCs are connected through VPC peering or VPN connections, an attacker who gains access to an EC2 instance in one VPC can attempt to move laterally to other VPCs if there are permissive peering or VPN rules in place.

Important of AWS EC2 Security Logging Essentials:

1.System Logs (Syslog):
System logs provide information about the operating system and application-level events, such as startup and shutdown times, software installations, and application errors.
Need: System logs are essential for troubleshooting issues with your EC2 instances, identifying software errors, and understanding system behavior.

2.Access Logs (Web Server Logs):
Access logs generated by web servers (e.g., Apache, Nginx) record every HTTP request made to your EC2 instances, including the requested URLs, response codes, user agents, and IP addresses of clients.
Need: Access logs help monitor web traffic, analyze user behavior, and detect potential security threats like DDoS attacks or suspicious access patterns.

3.Security Logs:
Security logs contain information about security-related events, such as login attempts, authentication failures, and changes to security groups or firewall settings.
Need: Security logs are crucial for detecting and investigating security incidents, identifying unauthorized access attempts, and ensuring compliance with security policies.

4.Application Logs:
Application logs record application-specific events, errors, and activities related to your software and services running on EC2 instances.
Need: Application logs are vital for diagnosing and resolving issues within your applications, identifying performance bottlenecks, and optimizing application behavior.

5.CloudTrail Logs:
AWS CloudTrail logs track API activity within your AWS account, recording actions taken by users, roles, or services.
Need: CloudTrail logs are critical for auditing and compliance purposes, tracking changes to your AWS resources, and investigating suspicious or unauthorized activities.

6.Performance Metrics (CloudWatch Metrics):
CloudWatch metrics provide performance data on CPU utilization, memory usage, disk I/O, network traffic, and more.
Need: Performance metrics help monitor the health and performance of your EC2 instances, enabling you to optimize resource allocation and plan for scaling based on demand.

7.Instance Console Output:
The instance console output provides real-time information during instance launch, showing boot logs and potential errors.
Need: The instance console output is valuable for debugging boot issues and understanding the instance's initial configuration and setup.

Athena Query to Investigate EC2 Compromise:

-- List all EC2-related API calls in CloudTrail

```
SELECT
  eventTime,
  eventSource,
  eventName,
  userAgent,
  sourceIpAddress,
  userIdentity.userName,
  userIdentity.arn
FROM
  my_ec2_logs.cloudtrail_logs
WHERE
  eventSource = 'ec2.amazonaws.com'
ORDER BY
  eventTime DESC;
```

-- List all security group modifications for EC2 instances

```
SELECT
  eventTime,
  eventName,
  sourceIpAddress,
  userIdentity.userName,
  requestParameters,
  responseElements
FROM
  my_ec2_logs.cloudtrail_logs
WHERE
  eventSource = 'ec2.amazonaws.com'
  AND eventName LIKE '%SecurityGroup%';
```

-- List all EC2 instances and their associated IAM roles and instance profiles

```
SELECT
  instanceid,
  iaminstanceprofileid,
  iaminstanceprofilearn
FROM
  my_ec2_logs.ec2_instances;
```

Athena Query to Investigate EC2 Compromise:

```
-- List all network traffic to and from an EC2 instance based on its private IP address

SELECT
  date_format(from_iso8601_timestamp(starttime), 'yyyy-MM-dd HH:mm:ss') as timestamp,
  srcaddr,
  dstaddr,
  srcport,
  dstport,
  protocol,
  packets,
  bytes
FROM
  my_ec2_logs.vpc_flow_logs
WHERE
  (srcaddr = 'EC2_INSTANCE_PRIVATE_IP' OR dstaddr = 'EC2_INSTANCE_PRIVATE_IP')
  -- Optionally, add other filters such as a specific time range or subnet ID
ORDER BY
  timestamp DESC;
```

```
-- List user login activity from EC2 instance logs (Assuming Your EC2 instances have been configured to log login attempts

SELECT
  date_format(from_iso8601_timestamp(eventTime), 'yyyy-MM-dd HH:mm:ss') as timestamp,
  username,
  src_ip,
  event
FROM
  my_ec2_logs.ec2_instance_logs
WHERE
  event = 'UserLogin'  -- Adjust the event type based on your log configuration
ORDER BY
  timestamp DESC;
```

Disk Analysis Runbook:

```
-- This query will show the top disk-consuming directories and files
-- Choose the appropriate log group name for your CloudWatch Logs that contain disk space metrics
-- Replace "/aws/ecs/containerinsights/<CLUSTER_NAME>/performance" with the correct log group name
-- For EC2 instances, it might be something like "/var/log/messages" if you've configured CloudWatch Agent to collect disk metrics in this log group.

-- Identify the top directories consuming disk space
WITH logdata AS (
  SELECT
    time,
    message
  FROM
    "/aws/ecs/containerinsights/<CLUSTER_NAME>/performance"
  -- For EC2 instances, replace <CLUSTER_NAME> with the appropriate log group name containing disk space metrics
  WHERE
    metric_name = 'disk_used_percent'
    AND mount_point != '/'
)

SELECT
  date_format(from_unixtime(time/1000), '%Y-%m-%d %H:%i:%s') AS timestamp,
  regexp_extract(message, '.*\"filesystem\":\"(.+?)\".*', 1) AS filesystem,
  regexp_extract(message, '.*\"mountpoint\":\"(.+?)\".*', 1) AS mount_point,
  avg(regexp_extract(message, '.*\"value\":(.+?),.*', 1)::double) AS avg_disk_usage_percent,
  max(regexp_extract(message, '.*\"value\":(.+?),.*', 1)::double) AS max_disk_usage_percent
FROM
  logdata
GROUP BY
  1, 2, 3
ORDER BY
  max_disk_usage_percent DESC
LIMIT
  10;
```