**Amazon Inspector**

**Definition**

Amazon Inspector is a vulnerability management service that continuously scans your AWS workloads for software vulnerabilities and unintended network exposure. Amazon Inspector automatically discovers and scans running Amazon EC2 instances, container images in Amazon Elastic Container Registry (Amazon ECR), and AWS Lambda functions for known software vulnerabilities and unintended network exposure.

**Roles**

AWS Inspector having the following roles

{

2 "Version": "2012-10-17",

3 "Statement": [

4 {

5 "Effect": "Allow",

6 "Action": [

7 "directconnect:DescribeConnections",

8 "directconnect:DescribeDirectConnectGatewayAssociations",

9 "directconnect:DescribeDirectConnectGatewayAttachments",

10 "directconnect:DescribeDirectConnectGateways",

11 "directconnect:DescribeVirtualGateways",

12 "directconnect:DescribeVirtualInterfaces",

13 "directconnect:DescribeTags",

14 "ec2:DescribeTags",

15 "ec2:DescribeAvailabilityZones",

16 "ec2:DescribeCustomerGateways",

17 "ec2:DescribeInstances",

18 "ec2:DescribeInternetGateways",

19 "ec2:DescribeManagedPrefixLists",

20 "ec2:DescribeNatGateways",

21 "ec2:DescribeNetworkAcls",

22 "ec2:DescribeNetworkInterfaces",

23 "ec2:DescribePrefixLists",

24 "ec2:DescribeRegions",

25 "ec2:DescribeRouteTables",

26 "ec2:DescribeSecurityGroups",

27 "ec2:DescribeSubnets",

28 "ec2:DescribeTransitGatewayAttachments",

29 "ec2:DescribeTransitGatewayConnects",

30 "ec2:DescribeTransitGatewayPeeringAttachments",

31 "ec2:DescribeTransitGatewayRouteTables",

32 "ec2:DescribeTransitGatewayVpcAttachments",

33 "ec2:DescribeTransitGateways",

34 "ec2:DescribeVpcEndpointServiceConfigurations",

35 "ec2:DescribeVpcEndpoints",

36 "ec2:DescribeVpcPeeringConnections",

37 "ec2:DescribeVpcs",

38 "ec2:DescribeVpnConnections",

39 "ec2:DescribeVpnGateways",

40 "ec2:GetManagedPrefixListEntries",

41 "ec2:GetTransitGatewayRouteTablePropagations",

42 "ec2:SearchTransitGatewayRoutes",

43 "elasticloadbalancing:DescribeListeners",

44 "elasticloadbalancing:DescribeLoadBalancerAttributes",

45 "elasticloadbalancing:DescribeLoadBalancers",

46 "elasticloadbalancing:DescribeRules",

47 "elasticloadbalancing:DescribeTags",

48 "elasticloadbalancing:DescribeTargetGroups",

49 "elasticloadbalancing:DescribeTargetHealth",

50 "network-firewall:DescribeFirewall",

51 "network-firewall:DescribeFirewallPolicy",

52 "network-firewall:DescribeResourcePolicy",

53 "network-firewall:DescribeRuleGroup",

54 "network-firewall:ListFirewallPolicies",

55 "network-firewall:ListFirewalls",

56 "network-firewall:ListRuleGroups",

57 "tiros:CreateQuery",

58 "tiros:GetQueryAnswer"

59 ],

60 "Resource": [

61 "\*"

62 ]

63 },

64 {

65 "Effect": "Allow",

66 "Action": [

67 "ecr:BatchGetImage",

68 "ecr:BatchGetRepositoryScanningConfiguration",

69 "ecr:DescribeImages",

70 "ecr:DescribeRegistry",

71 "ecr:DescribeRepositories",

72 "ecr:GetAuthorizationToken",

73 "ecr:GetDownloadUrlForLayer",

74 "ecr:GetRegistryScanningConfiguration",

75 "ecr:ListImages",

76 "ecr:PutRegistryScanningConfiguration",

77 "organizations:DescribeAccount",

78 "organizations:DescribeOrganization",

79 "organizations:ListAccounts",

80 "ssm:DescribeAssociation",

81 "ssm:DescribeAssociationExecutions",

82 "ssm:DescribeInstanceInformation",

83 "ssm:ListAssociations",

84 "ssm:ListResourceDataSync"

85 ],

86 "Resource": "\*"

87 },

88 {

89 "Effect": "Allow",

90 "Action": [

91 "lambda:ListFunctions",

92 "lambda:GetFunction",

93 "lambda:GetLayerVersion",

94 "cloudwatch:GetMetricData"

95 ],

96 "Resource": "\*"

97 },

98 {

99 "Effect": "Allow",

100 "Action": [

101 "ssm:CreateAssociation",

102 "ssm:StartAssociationsOnce",

103 "ssm:DeleteAssociation",

104 "ssm:UpdateAssociation"

105 ],

106 "Resource": [

107 "arn:\*:ec2:\*:\*:instance/\*",

108 "arn:\*:ssm:\*:\*:document/AmazonInspector2-\*",

109 "arn:\*:ssm:\*:\*:document/AWS-GatherSoftwareInventory",

110 "arn:\*:ssm:\*:\*:managed-instance/\*",

111 "arn:\*:ssm:\*:\*:association/\*"

112 ]

113 },

114 {

115 "Effect": "Allow",

116 "Action": [

117 "ssm:CreateResourceDataSync",

118 "ssm:DeleteResourceDataSync"

119 ],

120 "Resource": [

121 "arn:\*:ssm:\*:\*:resource-data-sync/InspectorResourceDataSync-do-not-delete"

122 ]

123 },

124 {

125 "Effect": "Allow",

126 "Action": [

127 "events:PutRule",

128 "events:DeleteRule",

129 "events:DescribeRule",

130 "events:ListTargetsByRule",

131 "events:PutTargets",

132 "events:RemoveTargets"

133 ],

134 "Resource": [

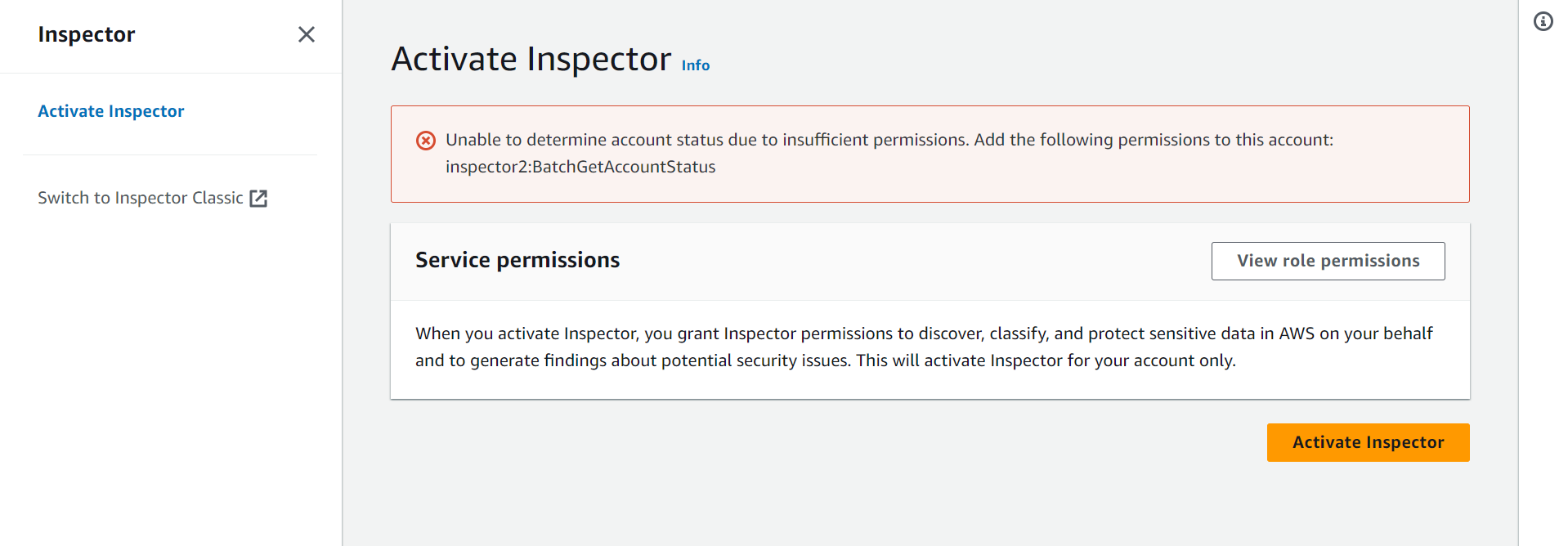
135 "arn:\*:events:\*:\*:rule/DO-NOT-DELETE-AmazonInspector\*ManagedRule"

136 ]

137 }

138 ]

139}



Source Log: Elastic cloud Computing.

Destination: S3.

Log Type:

1. Network related logs.
2. Package Vulnerabilities in Ec2-instance.

Assessment Type:

**Network Assessment (Inspector Agent is not required)**

1. Assessments Performed: Network configuration analysis to checks for ports reachable from outside the VPC.
2. Optional Agent: If the inspector Agent is installed on your Ec2-instances, the assessments also finds processes reachable on port.
3. Pricing: Pricing for network Assessments is based on the monthly volume of instance-assessments, where an instance-assessments denotes a successful assessment of an instance.

**Host Assessments (Inspector Agent is required)**

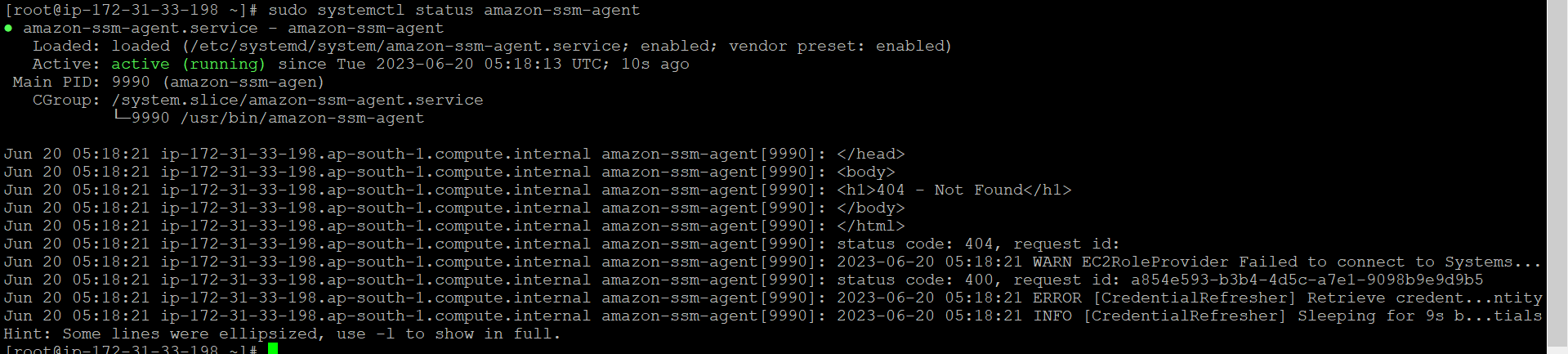
1. Assessments performed: Vulnerable software (CVE), host hardening (CIS Benchmarks), and security best practices.
2. Agent Deployment: Inspector Assessment require an agent to be installed on your Ec2-instance. We will automatically install the agent for instance that allow System Manager run Command.
3. Pricing: Pricing for host assessment is based on the mostly volume of agent-assessments denotes a successful assessment of an instance.

**Installation of SSM Agents.**

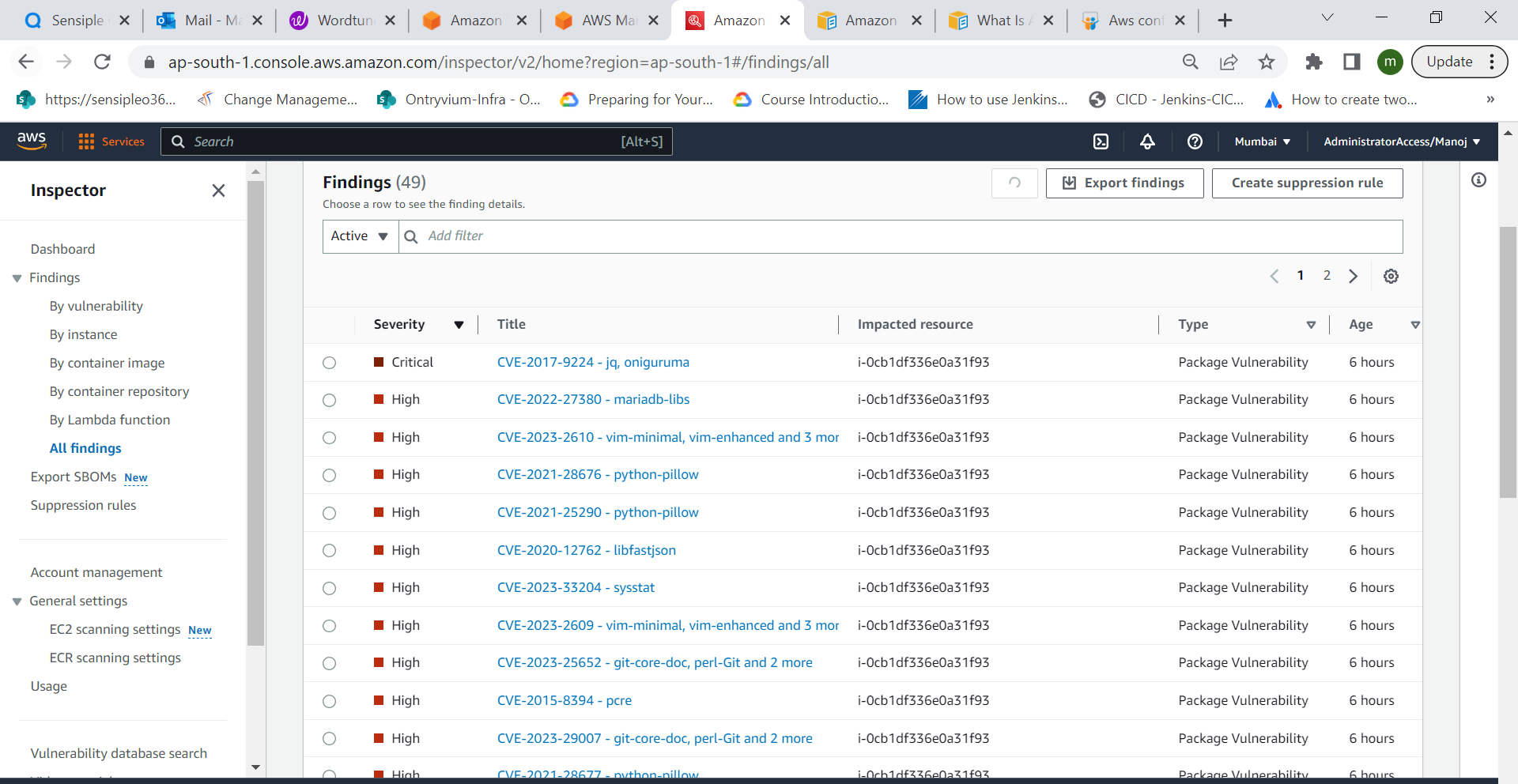
Installation Commands

sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux\_amd64/amazon-ssm-agent.rpm

sudo systemctl status amazon-ssm-agent

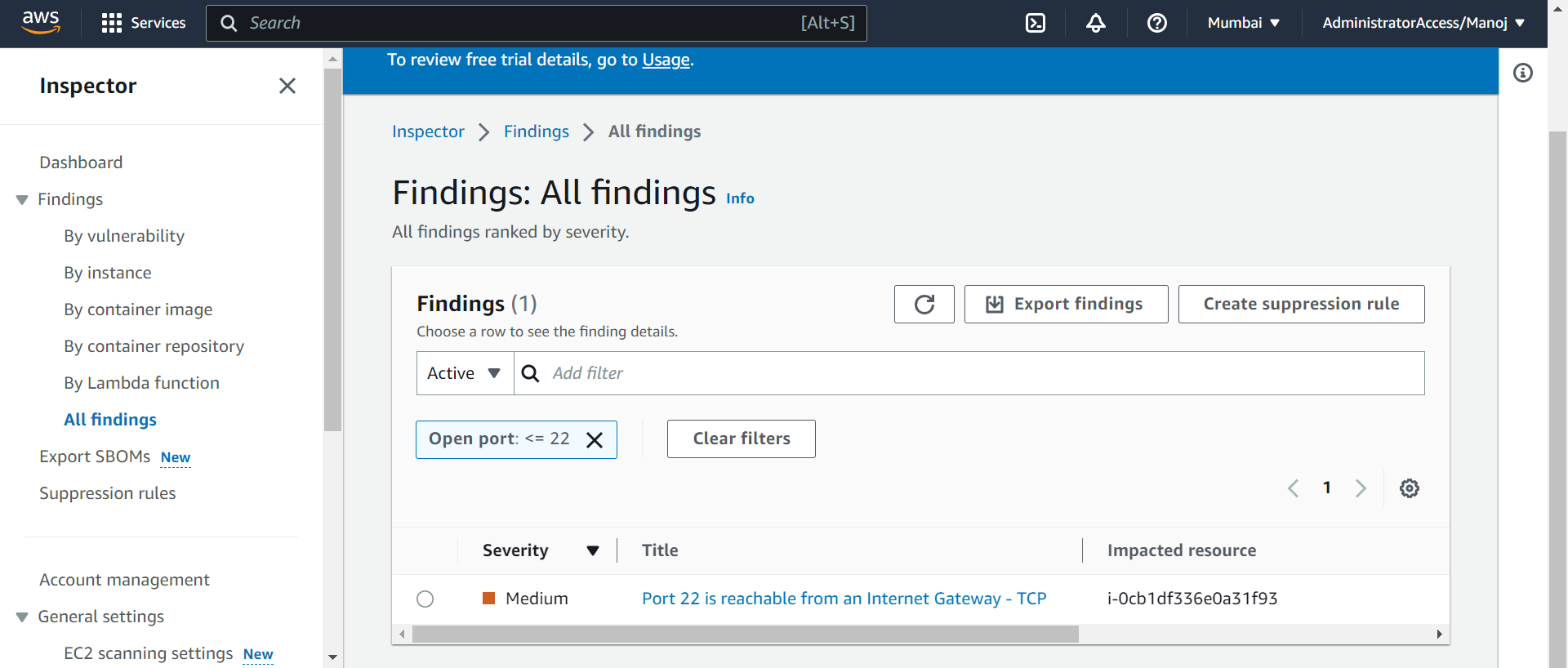


Once we installed the AWS SSM agent on that particular EC-2 instance we could see the host level vulnerabilities.

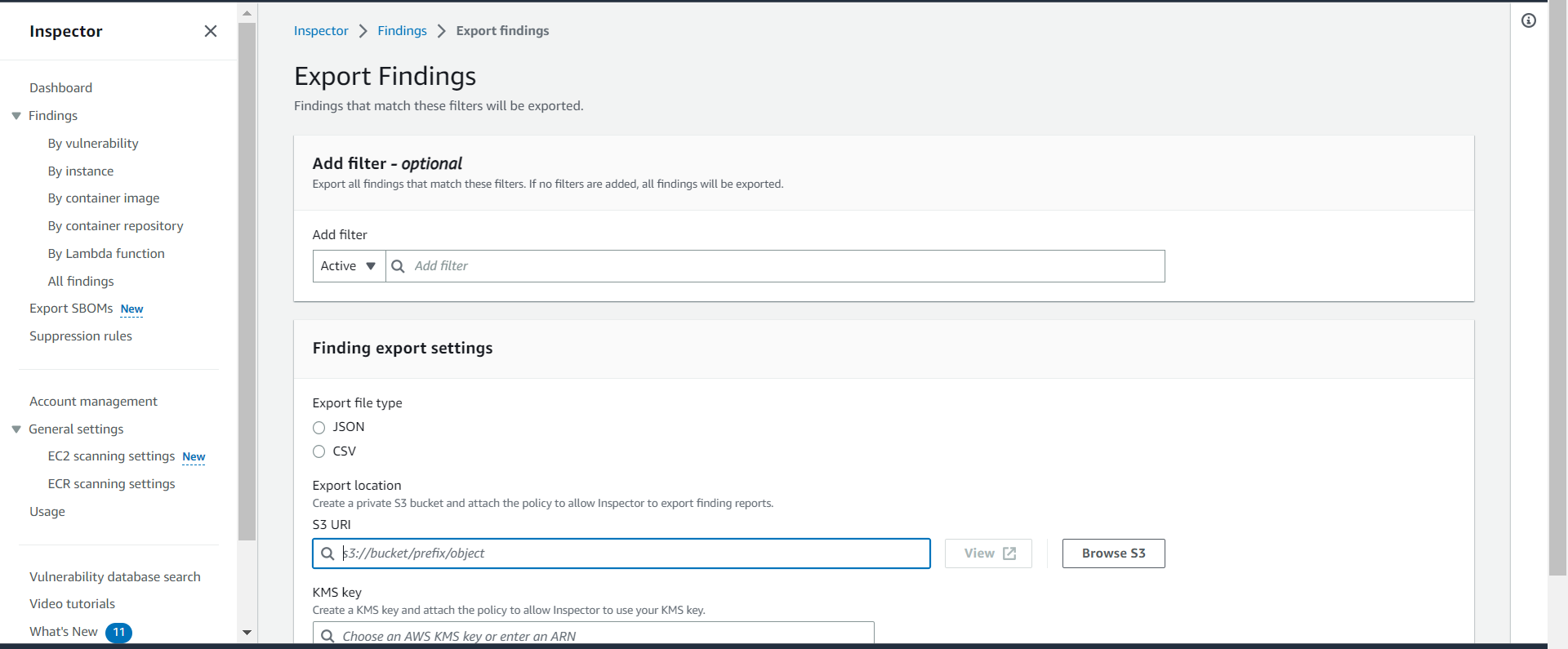


We can able to create a customized suspension rule based on our convinces.

**Example**

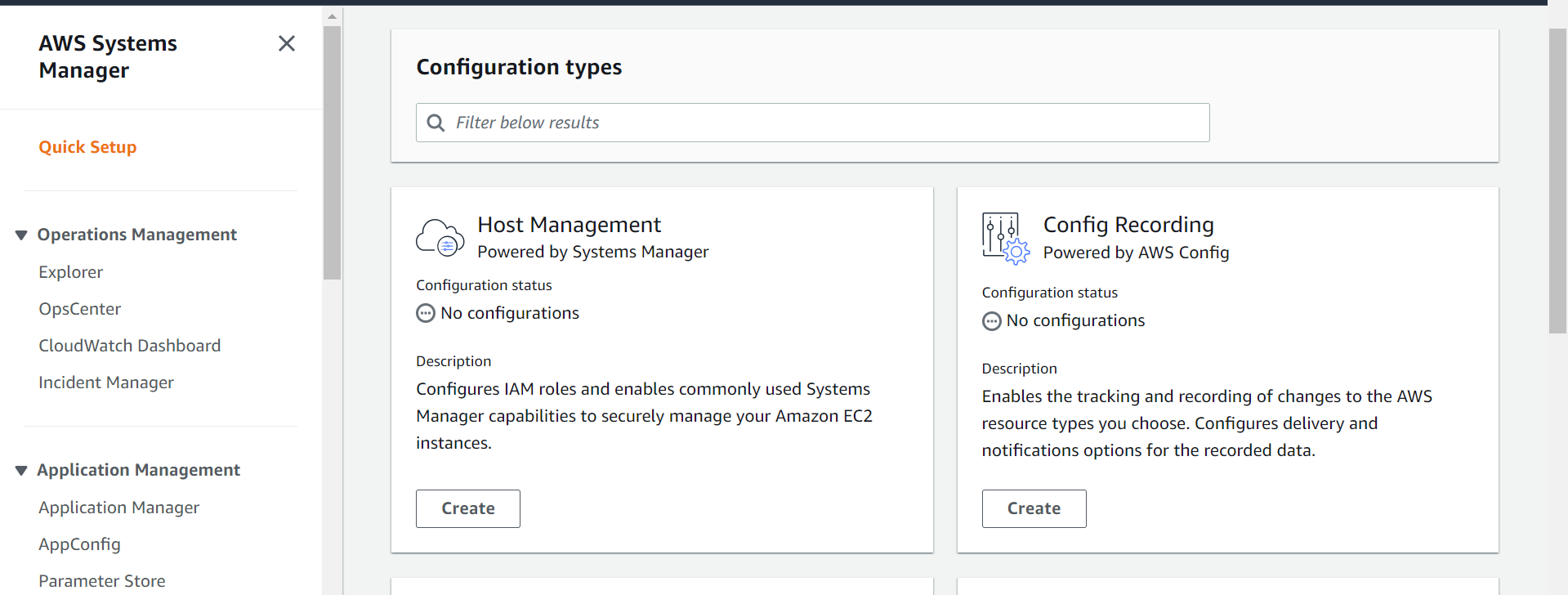


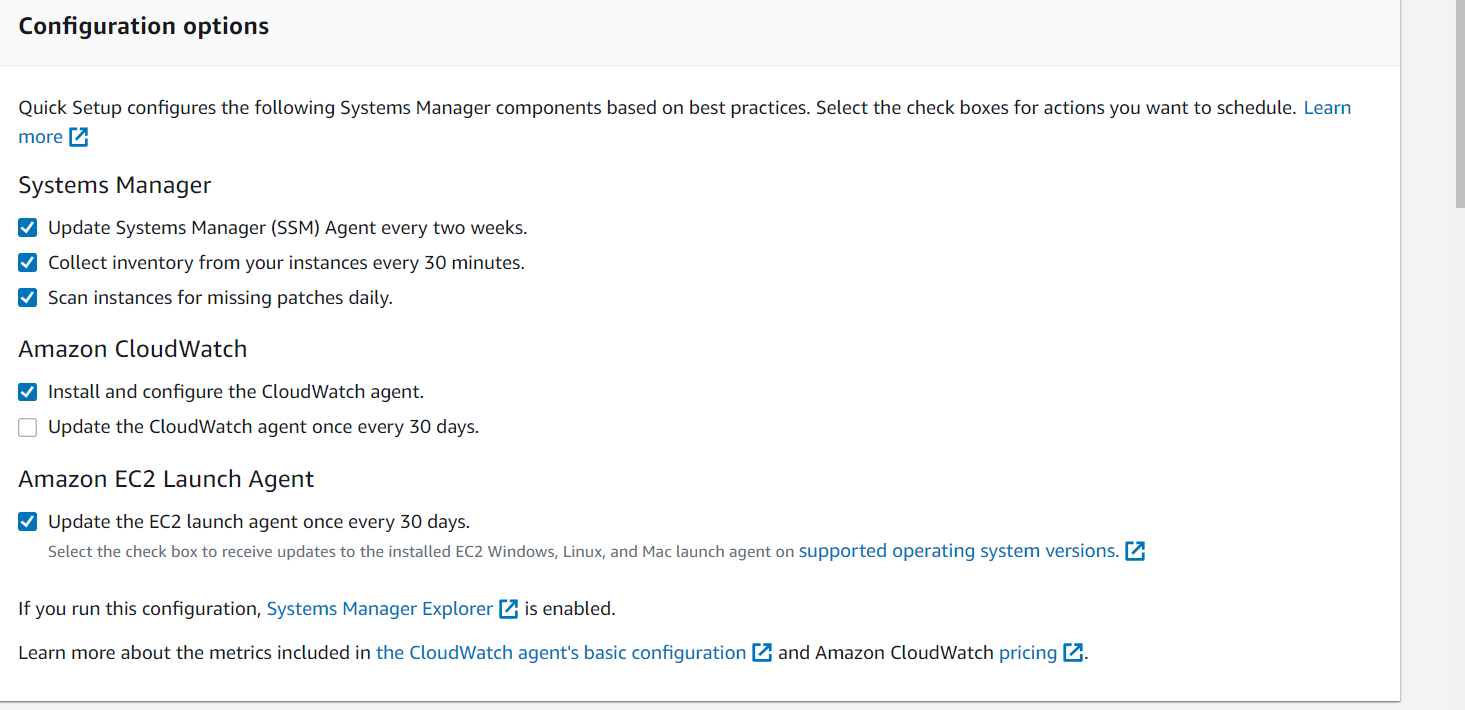
Using the export option, we can manually store the findings in S3-Bucket using the AWS Key management for security purpose.

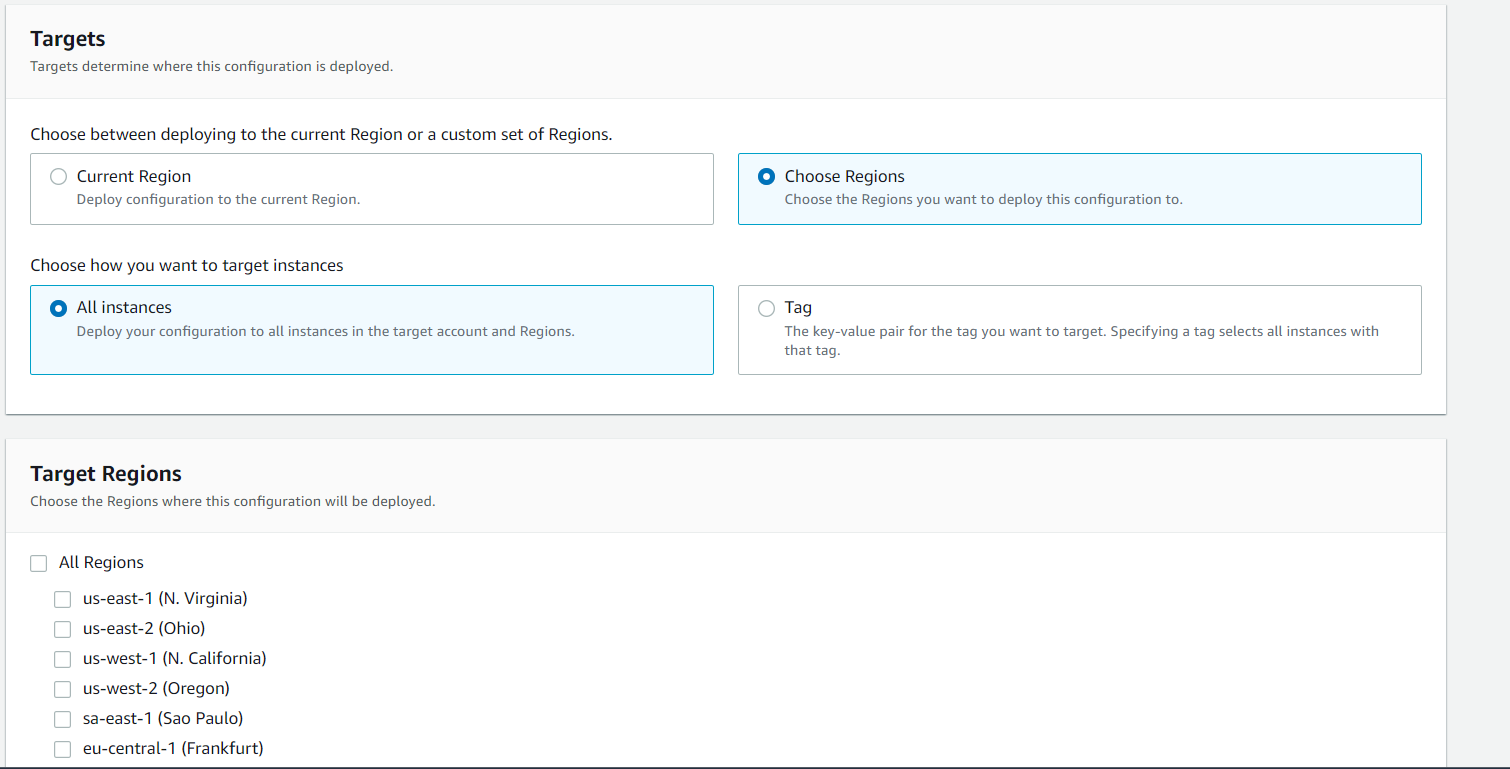


**AWS System Manager Configuration**







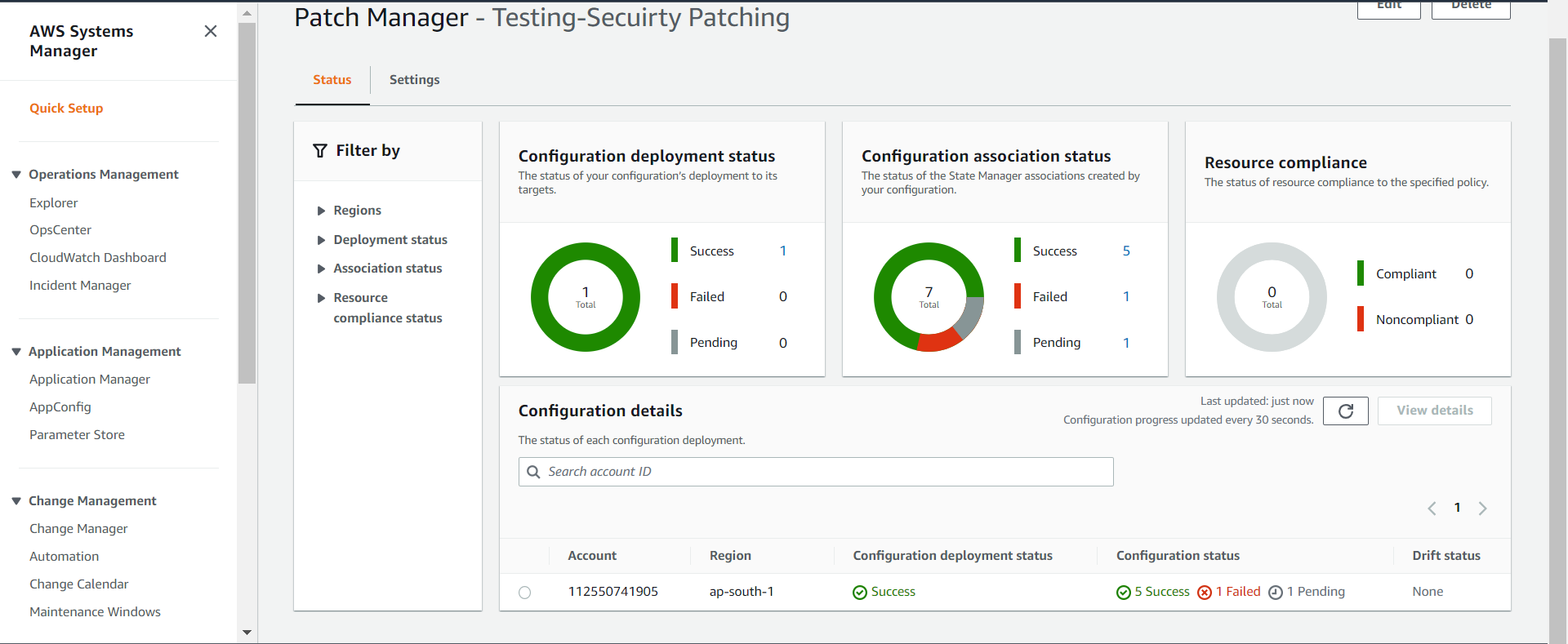


**Patch Management Policy**

Configuration Details

1. Configuration Name:
2. Scanning and Installation:
3. Patch baseline:
4. Patching Log store: S3
5. Target region:
6. Specific Node or All Node:

Once we setup the Patch management policy setup we could see the following Dashboard for what are the resources comes under the compliance or Non-Compliances.

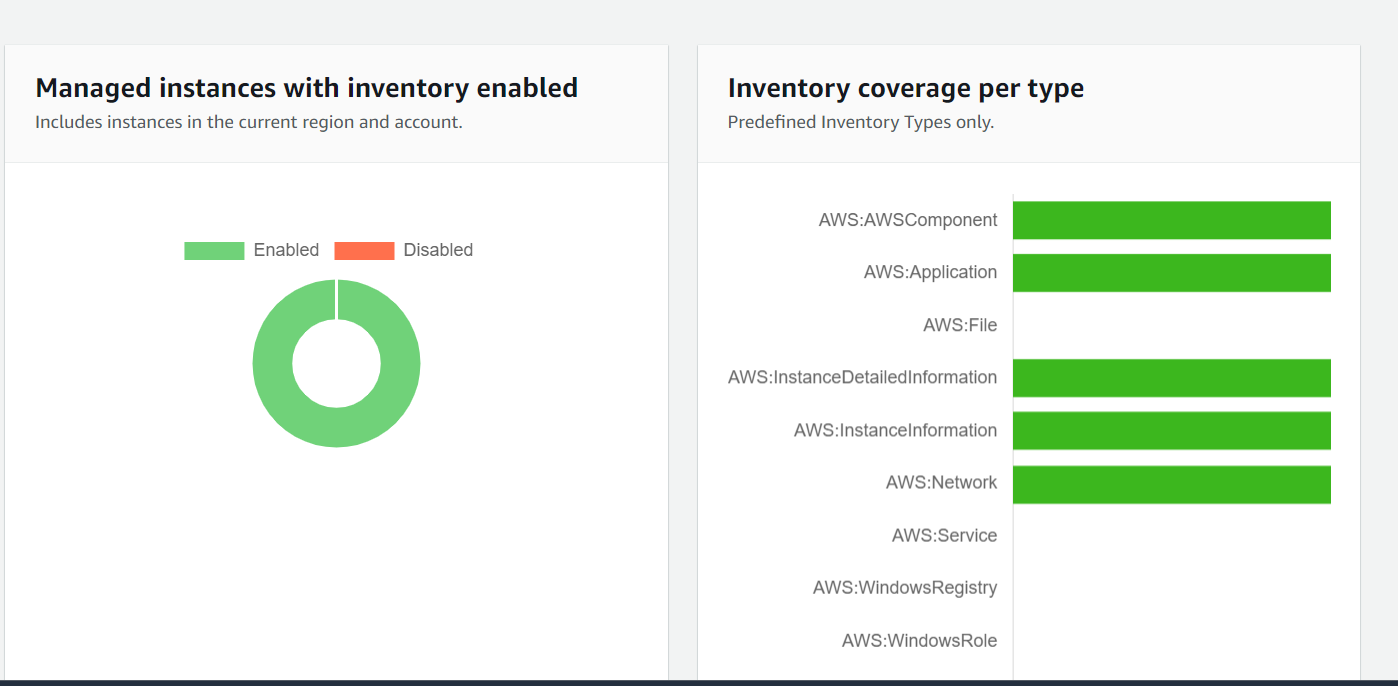


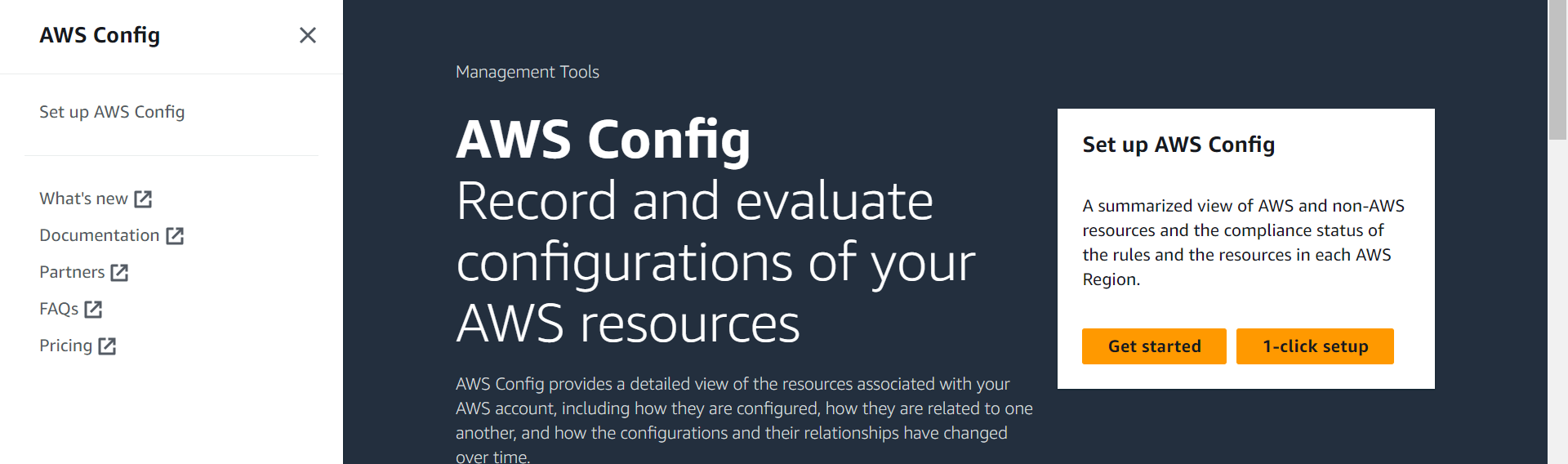
**AWS Config and SSM**

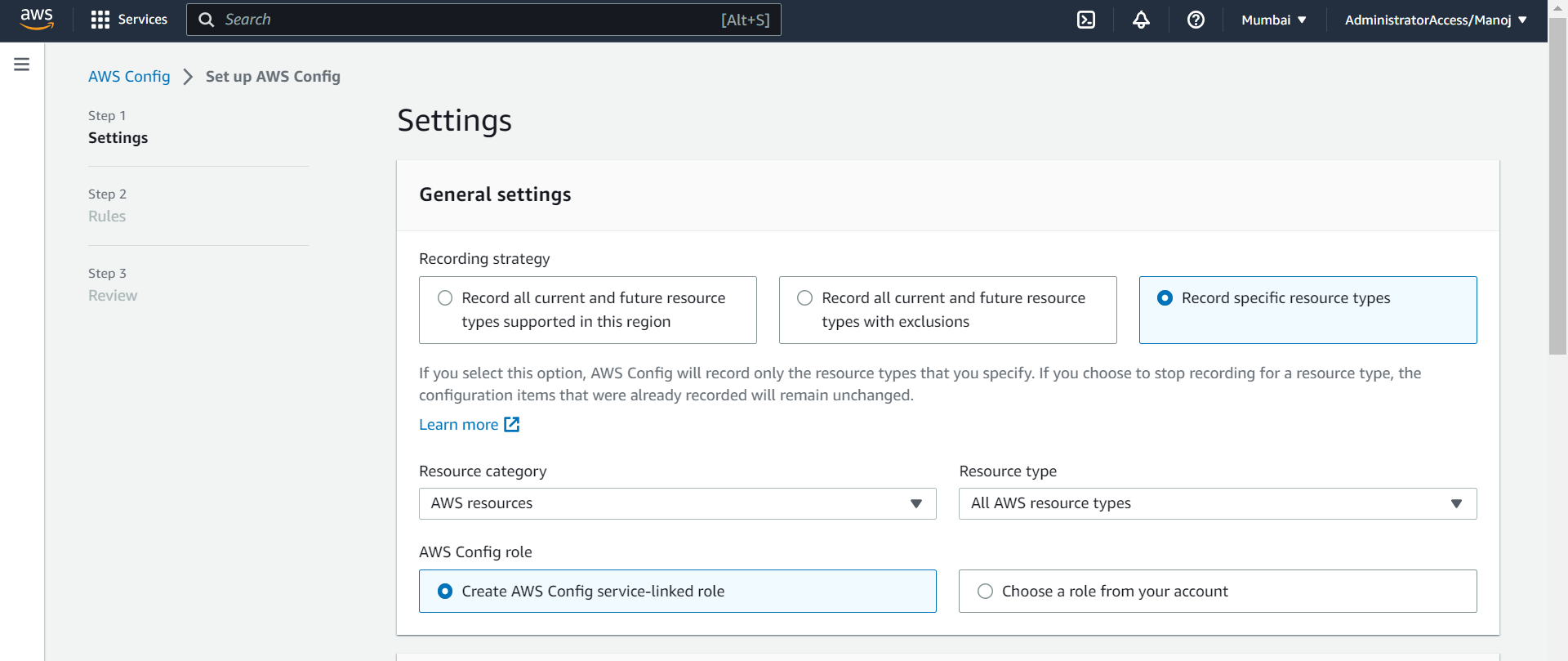
**Features**

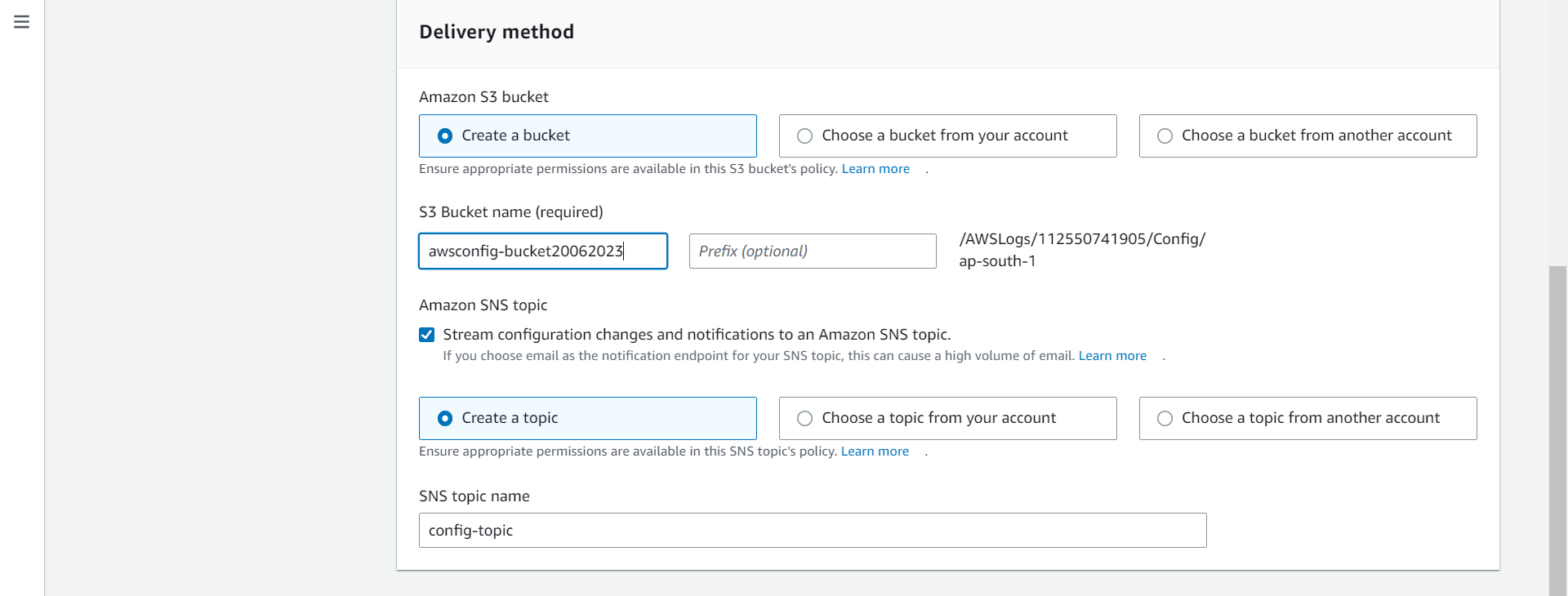
* Tracking and Maintaining the resources inventory based on the region wise.
* Tracking the all the changes based on our compliance rules.
* Ex: If someone changes the port number or IP in security group we could get the alert from their AWS SSM.
* Compliance rules will be creating based on our convinces.

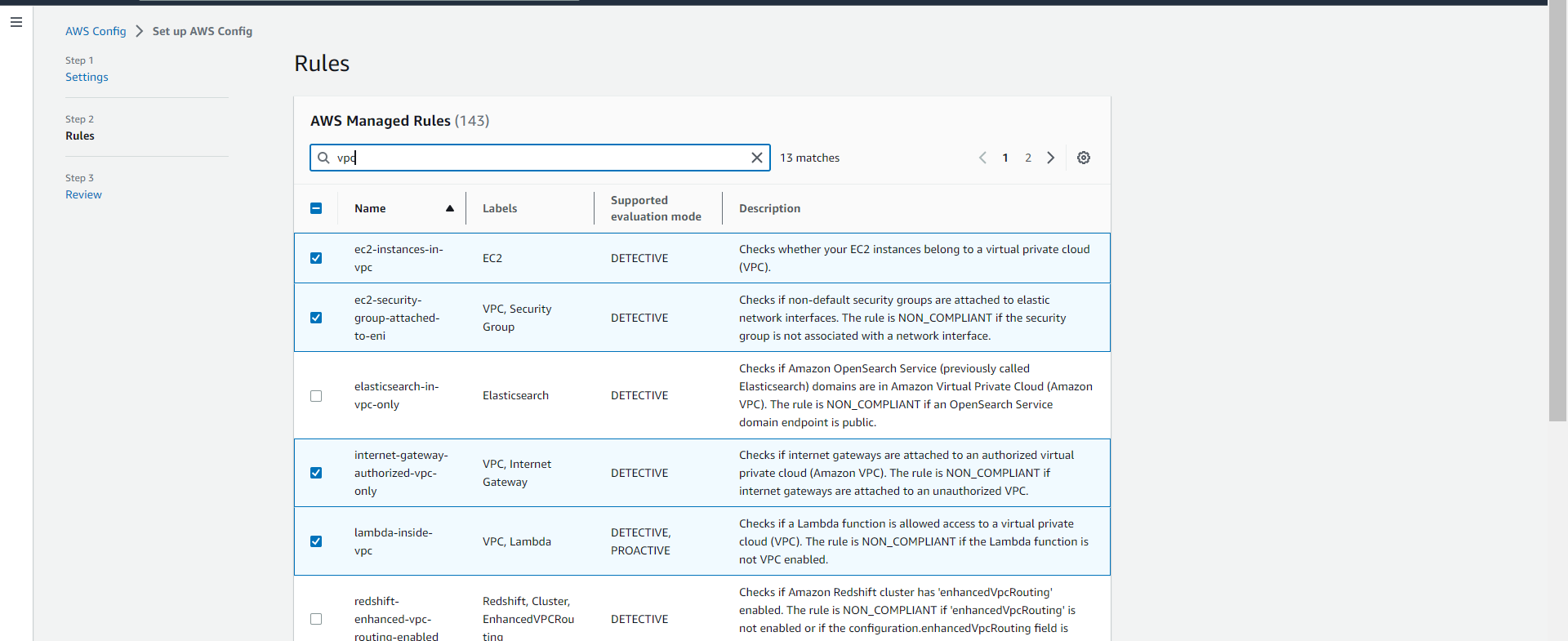
Inventory Dash Board

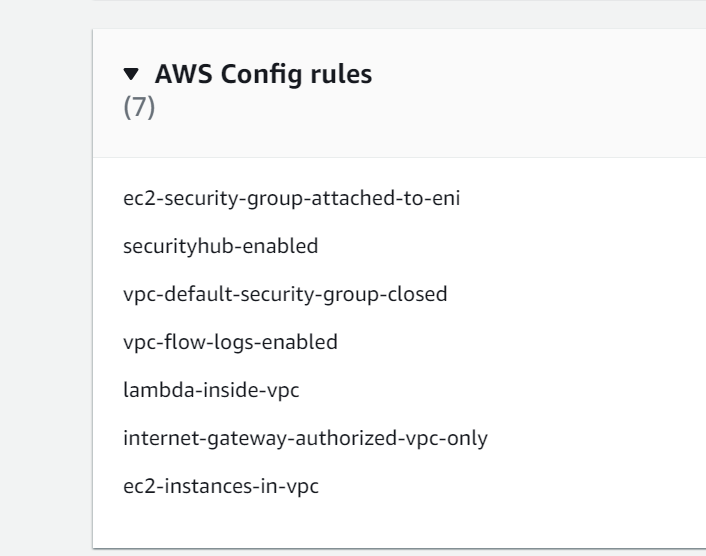






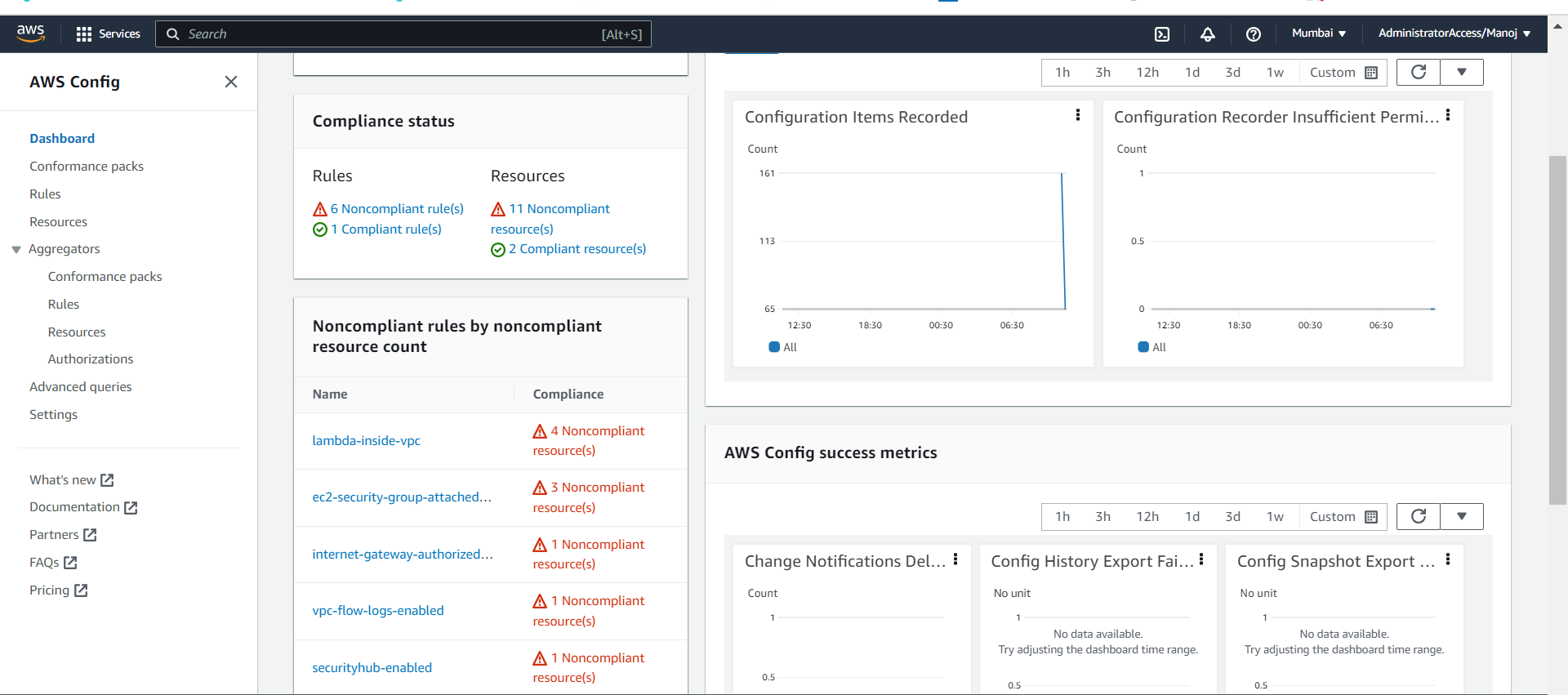






Final dashboard of AWS Config shows the what are all the rules making the compliance and non-compliance.

Based on our rules it shows the compliance or Non-Compliance.



**Rules**

We can create a rule using the following three method.

1. AWS Managed Rules.
2. Custom rule using the Lambda.
3. Custom rule using the Guard Duty.

**Cloud Trail**

**Trial Name:**

**Storage Location:**

**KMS for Log:**

**SNS Notifications:**

**CloudWatch Logs:**

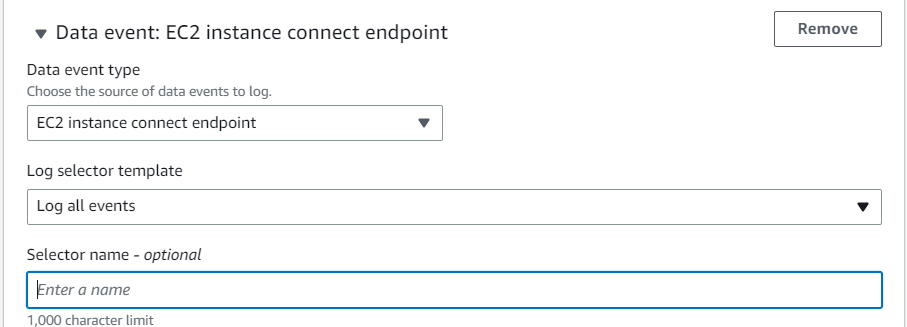
**Role for CloudWatch Logs:**

**Event Type:** Management Type, Data events and Insight events. (Choose any one of them based on our requirements)

Using the above configuration, we could create a cloud trial event with destination as a CloudWatch logs.

**EC2-INSTANCE CLOUD TRIAL CLOUD WATCH LOG**

For Ec2-instance Activity capture we can choose the Event type as Data event. And choose as an Ec2-Instance Connection Endpoint.



**AWS Trust Advisor**

The following feature is a main pillar of AWS Trust advisor.

1. Cost optimization.
2. Performance.
3. Security.
4. Fault tolerance.
5. Service limits.

Basis Support is only used for the **security** and **service limits** pillar other than we need to get the Business support.

Examples of recommendations.

**Cost optimization**

1. AWS lambda functions with Excessive timeout
2. Amazon Ec2 reserved Instance Lease Expiration.
3. Underutilized Amazon EBS Volume.
4. Idle Load balancer.

**Performance.**

1. Large Number of Rules in an Ec2-Security Group
2. High utilization Amazon Ec2-Instance.
3. CloudFront Alternate Domain Name.
4. AWS Well-architected high-risk issues for performance efficiency.

**Fault Tolerance**

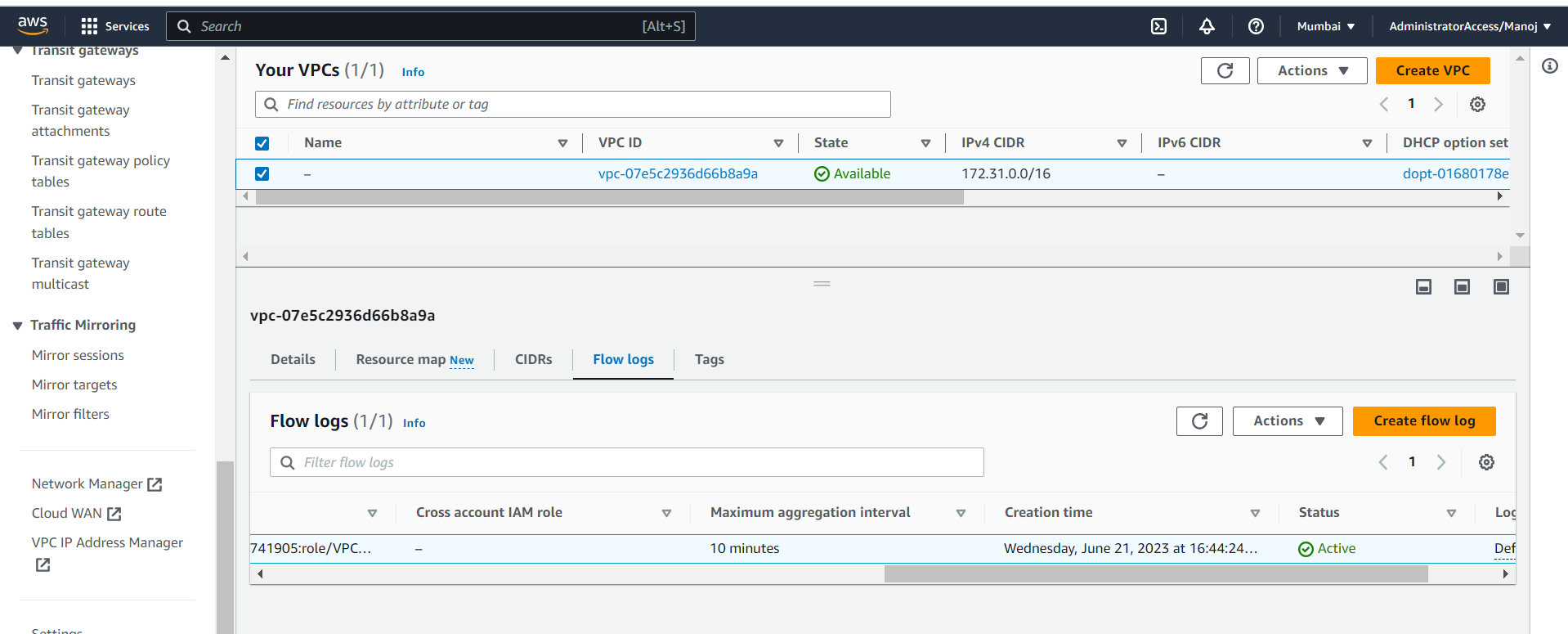
1. Amazon S3 Bucket versioning.
2. Amazon RDS Multi-AZ.
3. Amazon Ec2-Availability Zones.
4. ELB Cross Zone Load balancing.
5. Amazon EBS Snapshot.

**VPC FLOW LOGS**

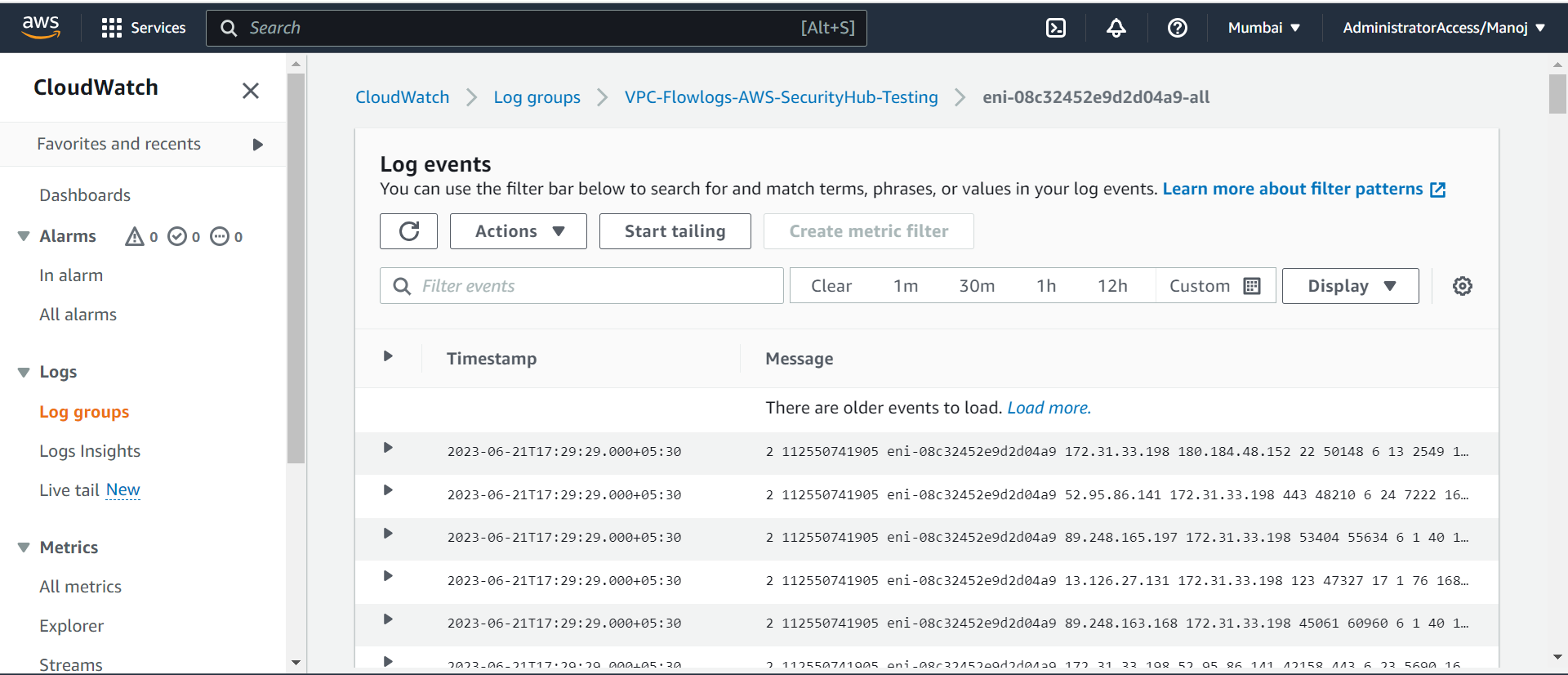
Amazon VPC Flow Logs enable you to capture information about the network traffic moving to and from network interfaces within your VPC. You can use VPC Flow Logs as a centralized, single source of information to monitor different network aspects of your VPC.

We collect, store and analyse network flow logs. We use this information to trouble shoot connectivity and security issues, and to make sure that network access rules are working as expected.

1. Create a **CloudWatch log Group** for storing the VPC flow logs
2. Created a **role** for Full access to CloudWatch log group from VPC
3. Enabling the **VPC Flow log** and set up the destination as a cloud Watch Log Group.
4. We could analysis the log using the AWS Analytics tools.



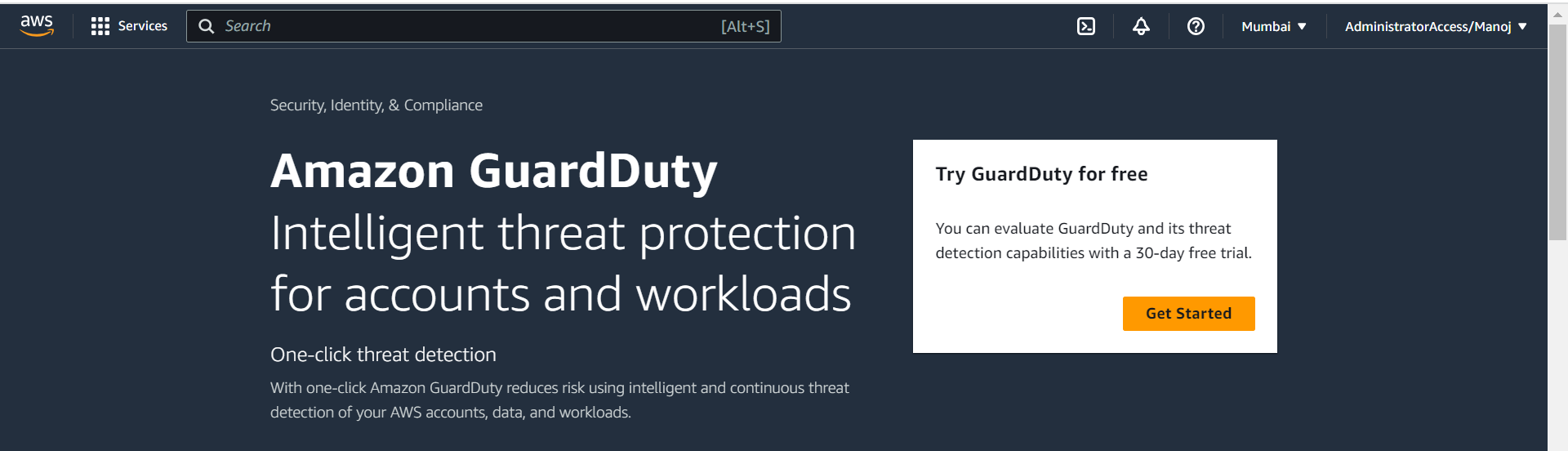
Once we enabled the VPC flow log we can see the network flow log details in CloudWatch log group. The following image is the example of cloud Watch log group.



**AWS Guard Duty**

Amazon Guard Duty is an intelligent threat detection service that provides customers with an accurate and easy way to continuously monitor and protect their Amazon Web Services accounts, workloads, and data stored in Amazon S3.

Guard Duty analyses billions of events across your Amazon Web Services accounts from Amazon CloudTrail Management Events (Amazon Web Services user and API activity in your accounts), Amazon CloudTrail S3 Data Events (Amazon S3 activity), Amazon VPC Flow Logs (network traffic data), and DNS Logs (name query patterns).



The following is AWS Managed policy for Guard Duty. We can’t modify the policy.

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"ec2:DescribeInstances",

"ec2:DescribeImages",

"ec2:DescribeVpcEndpoints",

"ec2:DescribeSubnets",

"ec2:DescribeVpcPeeringConnections",

"ec2:DescribeTransitGatewayAttachments",

"organizations:ListAccounts",

"organizations:DescribeAccount",

"s3:GetBucketPublicAccessBlock",

"s3:GetEncryptionConfiguration",

"s3:GetBucketTagging",

"s3:GetAccountPublicAccessBlock",

"s3:ListAllMyBuckets",

"s3:GetBucketAcl",

"s3:GetBucketPolicy",

"s3:GetBucketPolicyStatus"

],

"Resource": "\*"

}

]

}

**Features**

We could **Automate the Threat Response and Remediation**. Amazon Guard Duty offers HTTPS APIs, CLI tools, and Amazon CloudWatch Events to support automated security responses to security findings. For Example, you can automate the response workflow by using CloudWatch Events as an event sources to trigger an Amazon Lambda Function.

