
AWS Security Operations Lab Guide

Sponsored by



v1.0

Logistics

- **Format:** Overview presentation and lab setup, followed by paced exercises with a section recap.
- **Workshop Duration:** 2 hours
- **Target Audience:** Technical security users (security engineers, architects, DevOps) who have heard of Dome9 and know what Dome9 offers
- The organizing team is comprised of one speaker and 1-2 technical staff to help out and answer questions
- Participants bring their own laptops and have an AWS account setup (preferably beforehand) - **Please do this early on since it takes a few hours for a new AWS account to sync with a CFT template**
- Participants need to download the Cloudformation (**CFT**) template to run this lab
 - Please download CFT from the [Github here](#)

Table of Contents

1. **Lab Overview and AWS Setup** (10 -15 min)
 - a. Walkthrough of lab environment and exercises
 - b. Exercise 1.1: Setup AWS account
 - c. Exercise 1.2: Deploy sandbox environment in AWS
2. **AWS Security Operations Lab** (30 minutes)
 - a. Exercise 2.1: Identify zombie security group (no instance, but permissive rule)
 - b. Exercise 2.2: Identify an exposed internal asset in the AWS environment - Part 1
 - c. Exercise 2.3: Identify an exposed internal asset in the AWS environment - Part 2
 - d. Section Wrap Up
3. **Dome9 Overview and Onboarding** (10 -15 min)
 - i. Overview of Dome9
 - ii. Exercise 3.1: Connect Dome9 to your new AWS account
4. **AWS Security Operations Lab with Dome9** (30 min)
 - i. Exercise 4.1: Conduct inventory/asset review
 - ii. Exercise 4.2: Visualize security architecture
 - iii. Exercise 4.3: Identify zombie security group
 - iv. Exercise 4.4: Identify publicly accessible databases (2)
 - v. Section Wrap Up
5. **Security Posture Management Lab** (20 min)
 - i. Exercise 5.1: Enforce security policy
 - ii. Exercise 5.2: Active protection – Security group level
 - iii. Exercise 5.3: Active protection – Region level
 - iv. Section Wrap Up

6. S3 Security Lab (20 min)

- i. Exercise 6.1: Identify publicly exposed S3 buckets (2) using Dome9
- ii. Exercise 6.2: S3 Access Controls (ACL)
- iii. Exercise 6.3: S3 Access Controls (Bucket policies)
- iv. Section Wrap Up

7. Offboarding (5 min)

AWS Setup

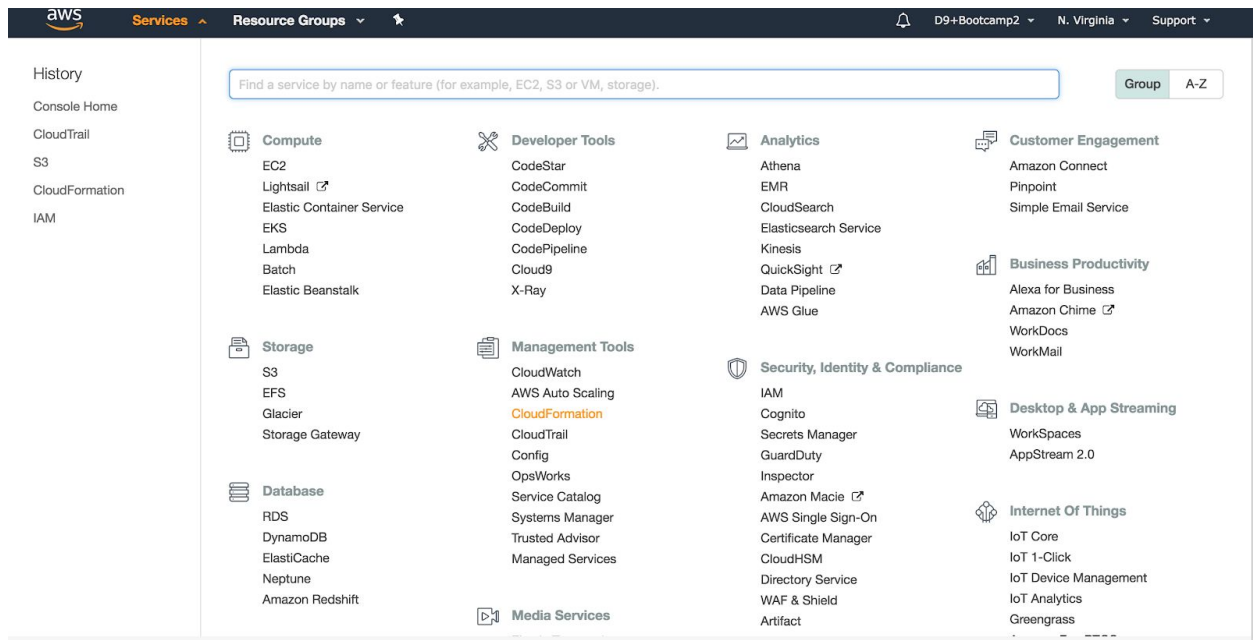
Exercise 1.1: Setup lab environment (Login to your lab AWS account)

The instructor should provide you with an AWS Account for this workshop. Ensure you have an AWS account setup before proceeding.

Exercise Complete!

Exercise 1.2: Deploy sandbox AWS environment

Navigate to AWS Cloudformation



Click on create stack and select “upload a template to S3” and choose the CFT file that you downloaded and click next

The screenshot shows the AWS CloudFormation console interface. At the top, there's a navigation bar with 'Services', 'Resource Groups', and 'CloudFormation' selected. The main heading is 'Create stack'. On the left, a sidebar lists 'Select Template', 'Specify Details', 'Options', and 'Review'. The 'Select Template' section is active, showing instructions: 'Select the template that describes the stack that you want to create. A stack is a group of related resources that you manage as a single unit.' There are three options: 'Design a template' (with a 'Design template' button), 'Choose a template' (with a dropdown menu), and 'Specify an Amazon S3 template URL' (with a 'No file chosen' button). Under 'Choose a template', the 'Upload a template to Amazon S3' option is selected, and a 'Choose File' button is visible next to it. At the bottom right, there are 'Cancel' and 'Next' buttons.

Create a stack name such as “<yourname>LoftLab” and select us-east1-a, 1-b, and 1-c for subnetAza, subnetAzb, subnetAzc and click next

The screenshot shows the 'Specify Details' step of the AWS CloudFormation console. The 'Stack name' field contains 'LoftLab'. Below this is the 'Parameters' section with four fields: 'AmiName' (ami-14c5486b), 'SubnetAZa' (us-east-1a), 'SubnetAZb' (us-east-1b), and 'SubnetAZc' (us-east-1c). Each field has a small description below it. At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons.

Click next **twice** and select “**I acknowledge that AWS CloudFormation might create resources**” and click create.

Rollback Triggers

No monitoring time provided

No rollback triggers provided

Advanced

Notification	
Termination Protection	Disabled
Timeout	none
Rollback on failure	Yes

Capabilities

i The following resource(s) require capabilities: [AWS::IAM::Role]

This template contains Identity and Access Management (IAM) resources that might provide entities access to make changes to your AWS account. Check that you want to create each of these resources and that they have the minimum required permissions. [Learn more.](#)

I acknowledge that AWS CloudFormation might create IAM resources.

[Quick Create Stack](#) (Create stacks similar to this one, with most details auto-populated)

Cancel Previous **Create**

You will need to wait 5 minutes for the CFT to automatically deploy the environment in your AWS account. At the end you should see the below screen:

The screenshot shows the AWS CloudFormation console interface. At the top, there are navigation tabs for 'Services', 'Resource Groups', and 'Stacks'. Below this, there are buttons for 'Create Stack', 'Actions', and 'Design template'. A filter is set to 'Active' and 'By Stack Name'. A table lists the stack 'demo' with a status of 'CREATE_COMPLETE' and a creation time of '2018-08-18 12:36:00 UTC-0700'. Below the stack list, there are tabs for 'Overview', 'Outputs', 'Resources', 'Events', 'Template', 'Parameters', 'Tags', 'Stack Policy', 'Change Sets', and 'Rollback Triggers'. The 'Events' tab is active, showing a list of events for the stack 'demo' on '2018-08-18'. The events table has columns for 'Status', 'Type', 'Logical ID', and 'Status Reason'. The events listed are:

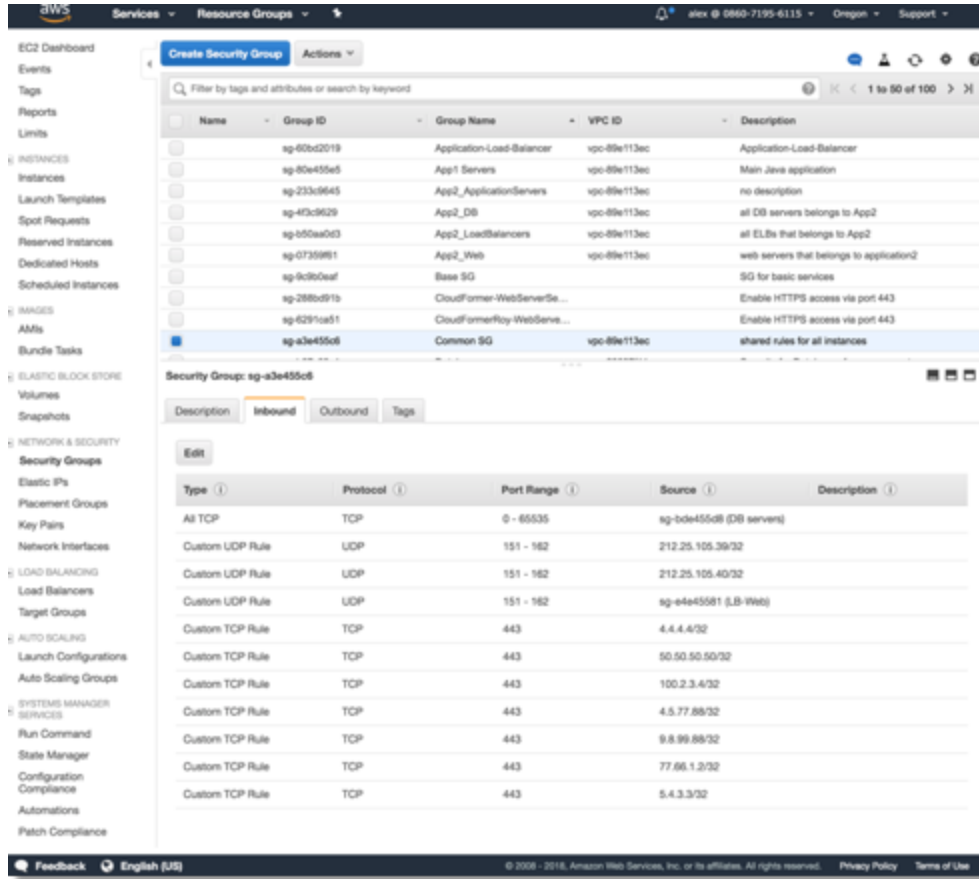
Status	Type	Logical ID	Status Reason
CREATE_COMPLETE	AWS::CloudFormation::Stack	demo	
CREATE_COMPLETE	AWS::ElasticLoadBalancingV2::LoadBalancer	Alb	
CREATE_COMPLETE	AWS::EC2::Instance	RabbitMQ1	
CREATE_COMPLETE	AWS::EC2::Instance	AgentService2	
CREATE_COMPLETE	AWS::EC2::Instance	monitoring2	
CREATE_COMPLETE	AWS::EC2::Instance	appserver2	
CREATE_COMPLETE	AWS::EC2::Instance	appserver1	
CREATE_COMPLETE	AWS::ElasticLoadBalancingV2::LoadBalancer	WebLB	

Exercise Complete! You have now deployed the sandbox AWS environment in your account.

AWS Security Policy Lab

Exercise 2.1: Identify zombie security group

In your AWS account, navigate to **N.Virginia region**. Explore EC2 instances, Security Groups, IAM and other services.



Hint: A zombie security group is a security group that has a permissive rule but has no instances tied to it!

Exercise Complete! You have now found your zombie policy!

Exercise 2.2: Identify an exposed internal asset in the AWS environment - Part 1

Dome9 Lab Guide

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public
appserver2	i-007936eda68ce9980	t2.nano	us-east-1a	running	2/2 checks ...	None	ec2-54-160-12-37.com...	54.160.12.3
agentservice1	i-023aa5ba3311bc0ae	t2.nano	us-east-1a	running	2/2 checks ...	None	ec2-54-90-180-225.co...	54.90.180.2
DB1	i-0332d2dfcbde92aea	t2.nano	us-east-1a	running	2/2 checks ...	None	ec2-52-55-112-2.comp...	52.55.112.2
webapp1	i-040ff87d30686aabd	t2.nano	us-east-1a	running	2/2 checks ...	None	ec2-34-235-163-45.co...	34.235.163.4
appserver1	i-04733d6d0ed7832...	t2.nano	us-east-1a	running	2/2 checks ...	None	ec2-107-23-207-78.co...	107.23.207.7
rabbitMQ	i-04a0360b33f65a436	t2.nano	us-east-1a	running	2/2 checks ...	None	ec2-52-90-183-107.co...	52.90.183.11
notificationr	i-04da8ba8f35d6e636	t2.nano	us-east-1b	running	2/2 checks ...	None	ec2-34-236-237-219.co...	34.236.237.2
monitoring2	i-04f821c018b2dacbc	t2.nano	us-east-1a	running	2/2 checks ...	None	ec2-52-54-79-241.com...	52.54.79.24
webapp2	i-050dfc294f75b1f6a	t2.nano	us-east-1b	running	2/2 checks ...	None	ec2-34-201-13-61.com...	34.201.13.6
monitoring	i-07af2582b19947fd6	t2.nano	us-east-1a	running	2/2 checks ...	None	ec2-54-173-177-56.co...	54.173.177.1
mongodb	i-0a796e390ba0900...	t2.nano	us-east-1a	running	1/2 checks ...	None	ec2-54-173-177-56.co...	54.173.177.1
bastion	i-0b0049503e8d238...	t2.nano	us-east-1c	running	2/2 checks ...	None	ec2-54-210-215-178.co...	54.210.215.1

There is an internal asset that is exposed to the public. Can you find it?

Exercise 2.3: Identify an exposed internal asset in the AWS environment - Part 2

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public
appserver2	i-007936eda68ce9980	t2.nano	us-east-1a	running	2/2 checks ...	None	ec2-54-160-12-37.com...	54.160.12.3
agentservice1	i-023aa5ba3311bc0ae	t2.nano	us-east-1a	running	2/2 checks ...	None	ec2-54-90-180-225.co...	54.90.180.2
DB1	i-0332d2dfcbde92aea	t2.nano	us-east-1a	running	2/2 checks ...	None	ec2-52-55-112-2.comp...	52.55.112.2
webapp1	i-040ff87d30686aabd	t2.nano	us-east-1a	running	2/2 checks ...	None	ec2-34-235-163-45.co...	34.235.163.4
appserver1	i-04733d6d0ed7832...	t2.nano	us-east-1a	running	2/2 checks ...	None	ec2-107-23-207-78.co...	107.23.207.7
rabbitMQ	i-04a0360b33f65a436	t2.nano	us-east-1a	running	2/2 checks ...	None	ec2-52-90-183-107.co...	52.90.183.11
notificationr	i-04da8ba8f35d6e636	t2.nano	us-east-1b	running	2/2 checks ...	None	ec2-34-236-237-219.co...	34.236.237.2
monitoring2	i-04f821c018b2dacbc	t2.nano	us-east-1a	running	2/2 checks ...	None	ec2-52-54-79-241.com...	52.54.79.24
webapp2	i-050dfc294f75b1f6a	t2.nano	us-east-1b	running	2/2 checks ...	None	ec2-34-201-13-61.com...	34.201.13.6
monitoring	i-07af2582b19947fd6	t2.nano	us-east-1a	running	2/2 checks ...	None	ec2-54-173-177-56.co...	54.173.177.1
mongodb	i-0a796e390ba0900...	t2.nano	us-east-1a	running	1/2 checks ...	None	ec2-54-173-177-56.co...	54.173.177.1
bastion	i-0b0049503e8d238...	t2.nano	us-east-1c	running	2/2 checks ...	None	ec2-54-210-215-178.co...	54.210.215.1

There is another internal asset that is exposed to the public (this one is harder to find)
Good luck!

Exercise Complete! You have now found your exposed assets!

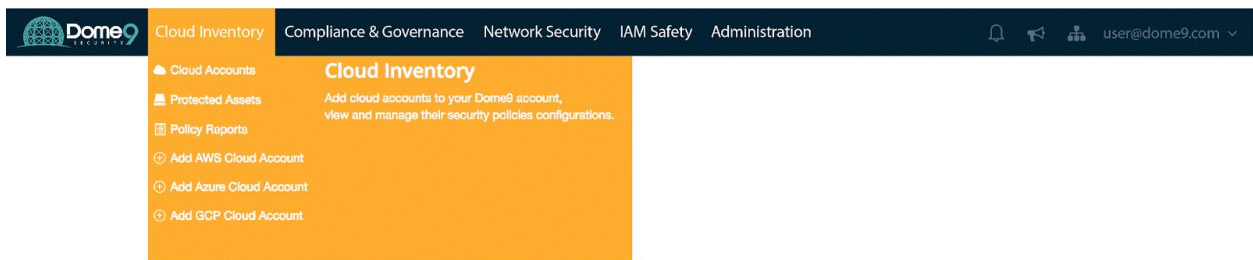
Dome9 Onboarding and Setup

Exercise 3.1: Connect Dome9 to your new AWS account

Onboard an AWS account

Onboarding an AWS account involves creating policies and attaching it to roles for Dome9 to use. For simplicity, the policies and roles have already been created. Follow these steps to onboard your AWS account to Dome9.

1. On the Dome9 console navigate to **Cloud Inventory** and select Add AWS Account.



2. Select the Dome9 operation mode, **Read/Write** to be used for the account.

Select operation mode

Monitor (Read-Only) Mode

In the Monitor mode, Dome9 Arc can be used for visualization, monitoring and auditing, and will not modify or actively manage your cloud environment.

Available in Monitor (Read-Only) Mode:

- Dome9 Clarity for visualization of network security
- Change notifications
- Audit trail
- Compliance reports
- Alerts
- Policy reports

When to Choose Monitor (Read-Only) Mode:

- You have another source of automation to manage your policies
- You want to manage your security group rules directly, rather than delegating to Dome9

[GET STARTED!](#)

Full-Protection (Read/Write) Mode

In the Full-Protection(Read/Write) Mode, Dome9 Arc can be used to actively manage your security posture and enforce best practices.

Available in Full-Protection (R/W) Mode:

- Dynamic Access Leases - time-limited, on-demand resource access
- Security group management console to edit policies in-place
- Tamper Protection and Region Lock for active enforcement
- Reusable policy objects such as IP Lists and DNS Objects
- Dome9 Clarity for visualization of network security
- Change notifications
- Audit trail
- Compliance reports
- Alerts
- Policy reports

When to Choose Full-Protection (R/W) Mode

- You want to use Dome9 Arc as your system of authority for security management
- You want to use Dome9's active management and enforcement capabilities to maintain a closed-by-default security posture

Note that even when you are using Dome9 **Full-Protection (Read/Write) Mode** you'll still be able to set individual security groups to **Monitor (Read-Only) Mode**

[GET STARTED!](#)



3. Click next again
4. Sign to the AWS console (aws.amazon.com) in a new browser tab or window (keep the Dome9 console open, as you will be switching between the two in the following steps).
5. Click **Services** and select the **IAM**
6. Select **Policies** and search for Dome9 and you should see two policies created. Click and review the policy for **dome9-write**

One entity was detached from the dome9-write-policy policy.

Create policy Policy actions

Filter policies Q dome9 Showing 2 results

	Policy name	Type	Used as	Description
<input type="radio"/>	Dome9-readonly-policy	Customer managed	None	read only
<input checked="" type="radio"/>	dome9-write-policy	Customer managed	None	Write policy for Dome9

7. In the AWS console, click **Roles** and “**Create new Role**”
8. Select Role Type: ‘**Another AWS Account**’, under options mark the ‘**Required External ID**’ option.
9. Enter the following:
 - AccountId: 634729597623
 - External ID: E+7NvdTUqNKZNoCSQ0L53@64
 - Require MFA: NOT checked

Create role 1 2 3

Select type of trusted entity

AWS service
EC2, Lambda and others

Another AWS account
Belonging to you or 3rd party

Web identity
Cognito or any OpenID provider

SAML 2.0 federation
Your corporate directory

Allows entities in other accounts to perform actions in this account. [Learn more](#)

Specify accounts that can use this role

Account ID* ⓘ

Options Require external ID (Best practice when a third party will assume this role)

You can increase the security of your role by requiring an optional external identifier, which prevents "confused deputy" attacks. This is recommended if you do not own or have administrative access to the account that can assume this role. The external ID can include any characters that you choose. To assume this role, users must be in the trusted account and provide this exact external ID. [Learn more](#)

External ID

Important: The console does not support using an external ID with the Switch Role feature. If you select this option, entities in the trusted account must use the API, CLI, or a custom federation proxy to make cross-account iam:AssumeRole calls. [Learn more](#)

Require MFA ⓘ

* Required
Cancel **Next: Permissions**

10. **READ-WRITE:** Make sure the following policies are selected:

- **SecurityAudit** (AWS managed policy).
- **AmazonInspectorReadOnlyAccess** (AWS managed policy).
- **dome9-write-policy**, that you created before. You can search for 'dome9' in the filter

11. Set Role Name with your choice ('Dome9-Connect' makes sense) and click on 'Create Role'

The screenshot shows the AWS IAM console 'Create role' page, step 3 of 3. The page is titled 'Create role' and 'Review'. Below the title, it says 'Provide the required information below and review this role before you create it.' The form has the following fields:

- Role name***: A text input field containing 'Dome9-Connect'. Below it, a note says 'Use alphanumeric and '+=, @-_' characters. Maximum 64 characters.'
- Role description**: A large text area that is currently empty. Below it, a note says 'Maximum 1000 characters. Use alphanumeric and '+=, @-_' characters.'
- Trusted entities**: A text field containing 'The account 634729597623'.
- Policies**: A list of three policies: 'SecurityAudit', 'AmazonInspectorReadOnlyAccess', and 'Dome9-readonly-policy'. Each policy name is preceded by a small orange icon and followed by an external link icon.
- Permissions boundary**: A text field containing 'Permissions boundary is not set'.

At the bottom of the page, there is a legend for '* Required', and three buttons: 'Cancel', 'Previous', and 'Create role'.

12. Copy the **Role ARN** value, and enter it in the **Role ARN** field in the Dome9 console.

13. Click 'Finish'

14. Review the onboarded cloud account summary.
15. At the end of the onboarding process all the Security Groups will be in **Read-Write** mode.

Exercise Complete! You have now connected your Dome9 to your AWS account

Security Architecture Review Lab

Exercise 4.1: Conduct inventory/asset review

Switch into your Dome9 console for the remaining part of this module

Dome9 presents a single console view of all your assets, on all platforms, from which you can search or filter for specific assets of interest, and see details about their security posture. In this section of the Dome9 console, you can see a summary of all the assets in your VPCs that are protected by Dome9. These assets can include, for example, instances (such as EC2s), RDSs, and load balancers. Dome9 fetches information about these assets from the cloud platforms (AWS, Azure, Google) and presents it in a console view.

1) View your Protected Assets

The main page shows a list of your assets that are protected by Dome9, organized by region. Filter the list using the filters on the left, or search for assets by name in the search bar.

The screenshot shows the Dome9 Cloud Inventory interface. The top navigation bar includes 'Cloud Inventory', 'Compliance & Governance', 'Network Security', 'IAM Safety', and 'Administration'. Below this, the 'Protected Assets' section is active, displaying a list of assets for the account 'AWS > N. Virginia > vpc-034dca60511e63fdb'. The assets are listed in a table with columns for asset name, type, and IP address. The assets include DB1, Demo-Alb-17VXXSZA5YMSG, Demo-LambdaFunction-ENNM13X(YAHS), bastion, and several EC2 instances (t2.nano) with various IDs and IP addresses. The interface also features a search box, filter options, and a 'Publicly Accessible' section.

For each asset in the list, the type, and its external IP address (if it has one) are shown.

Click on one of the assets in the list to see more details for it.

Please enter how many total EC2 instances are displayed in the google form.

2) View your Security Groups

The main page shows a list of all your managed security groups, in all your Dome9 managed accounts, on all cloud providers.

Filter the list using the search box or filter options on the left. You can filter by account, VPC, cloud region, protection method (full, read-only), and the number of instances or alerts.

The screenshot shows the Dome9 Security Groups interface. The top navigation bar includes 'Cloud Inventory', 'Compliance & Governance', 'Network Security', 'IAM Safety', and 'Administration'. The current view is 'Security Groups' for the 'productionVPC' in 'AWS > N. Virginia'. The left sidebar contains filters for Region (N. Virginia selected), VPC (productionVPC selected), Tags, Number of instances (0 to 2), Number of alerts (0 to 2), Operation Mode (Manage selected), and Tamper Protection. The main table lists security groups with their instance and alert counts.

Security Group Name	Instances	Alerts
Demo-AgentServiceSG-1VHM88J7Q2Z0B	2	2
Demo-DBServersSG-NSFU0OAWT5RY	1	0
Demo-DevopSG-13SQSHJNV9AJV	0	1
Demo-LambdaSG-1LAMFLCQ8CJ6	0	0
Demo-MQSG-1UHK4G0BMVPP7	1	1
Demo-MonitoringSG-1OE9BZ4APK6R4	1	2
Demo-NotificationServerSG-VPSD01C706WE	1	0
Demo-WebMonitorSG-55MGJUL28QDG	1	1
Demo-WebappSG-SYSUWBJB2BRR	2	1
Demo-appserverSG-169ORD4NDY8JE	2	1
Demo-default-1AN3SSLBMN72B	1	0
default	0	0
prod-alb-sg	0	0
prod-bastion-ie-sg	1	0
prod-mongo-sg	1	1
prod-webapp-elb-sg	0	0

Please enter how many total SGs are displayed in the google form.

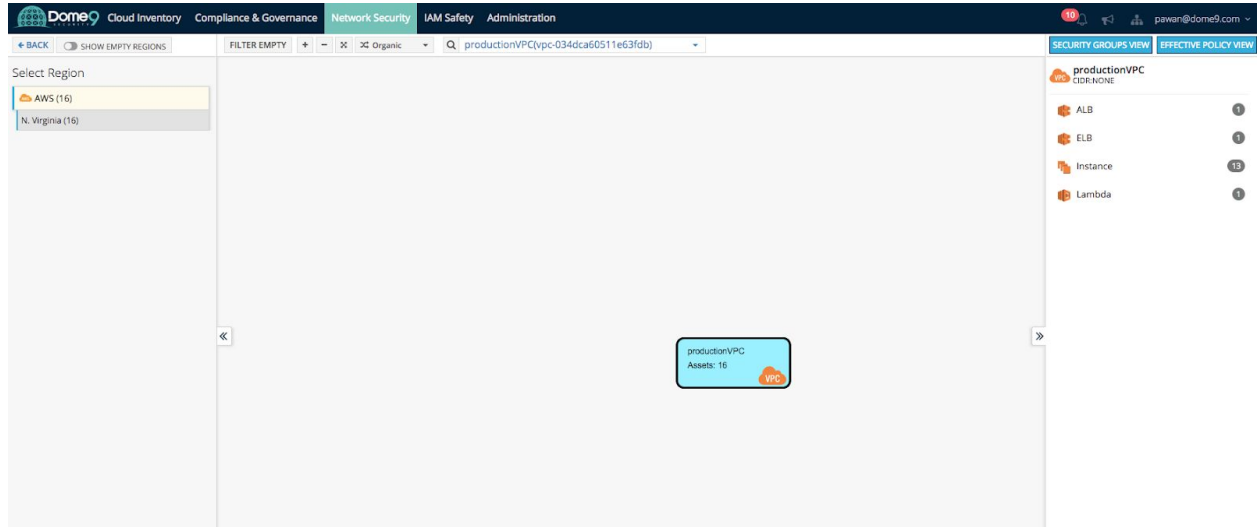
Exercise Complete! You have now explored your AWS instances and reviewed your inventory within the protected asset view of Dome9.

Exercise 4.2: Visualize security architecture

Dome9 Clarity gives a graphical visualization of the security groups in your cloud environment, and their effects on the cloud assets in the environment. It shows the security groups, traffic sources, and permitted traffic paths in the cloud network. The view is organized logically, according to the level of exposure of the Security Group to the external world. You can use Clarity to analyze your cloud network for security issues such as access to sensitive components from the internet, or to troubleshoot it for connectivity issues such as blocked paths to components.

1) Select a cloud environment

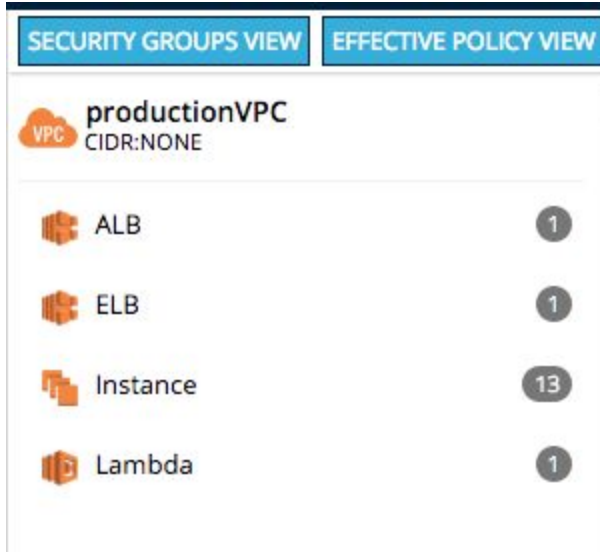
Select **Clarity** from the main menu. A list of your cloud accounts is shown on the left.



2) View an environment with the Security Group view

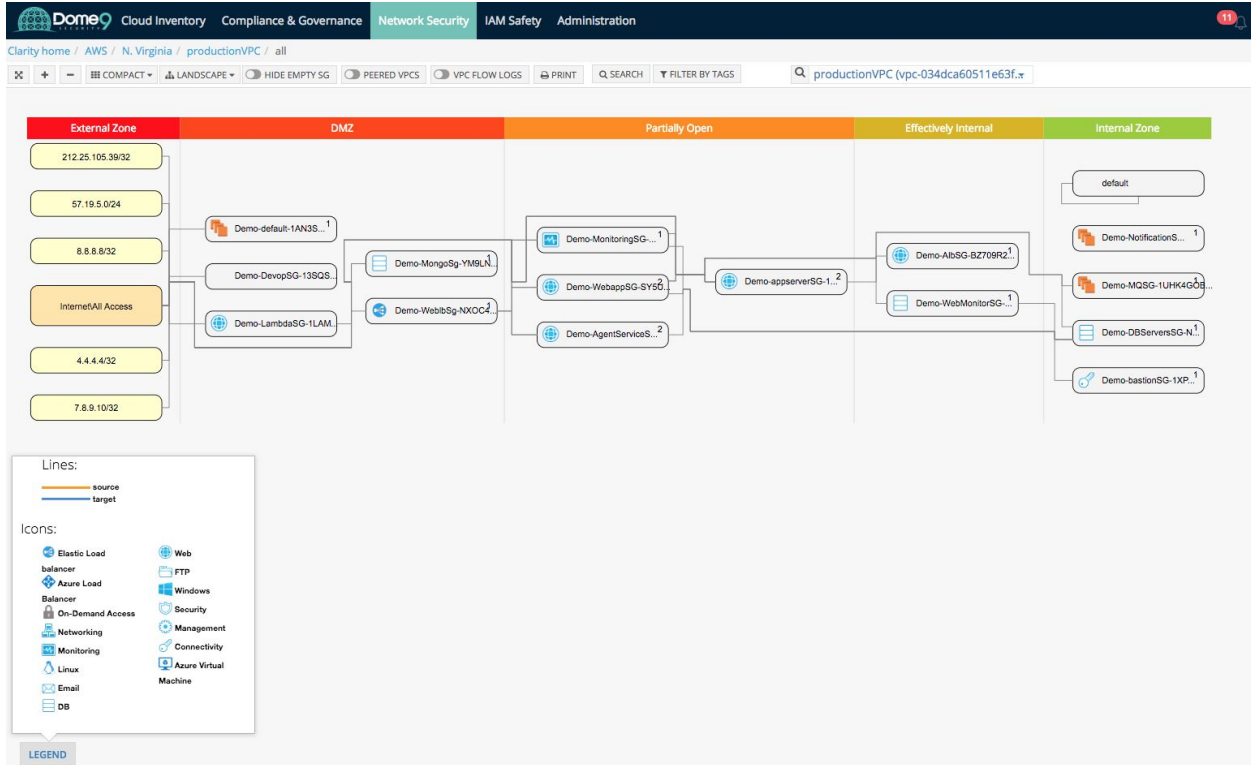
In this step, an environment will be visualized with the Security Group view. This view shows all the Security Groups.

- a) In Clarity, select a cloud environment in one of your accounts (in the previous section), and then select **Security Groups** from the list in the menu bar on the upper right.

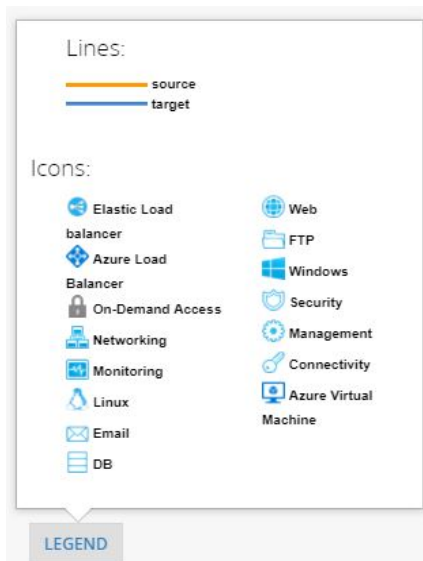


You can tell how the security groups (SG) interact with each other and can now understand whether any internal assets are communicating with the public world based on their inbound and outbound rules. The different swimlanes (which are auto classified by Dome9, so customers don't need to manually create them) represent security groups with various levels of exposure to the public.

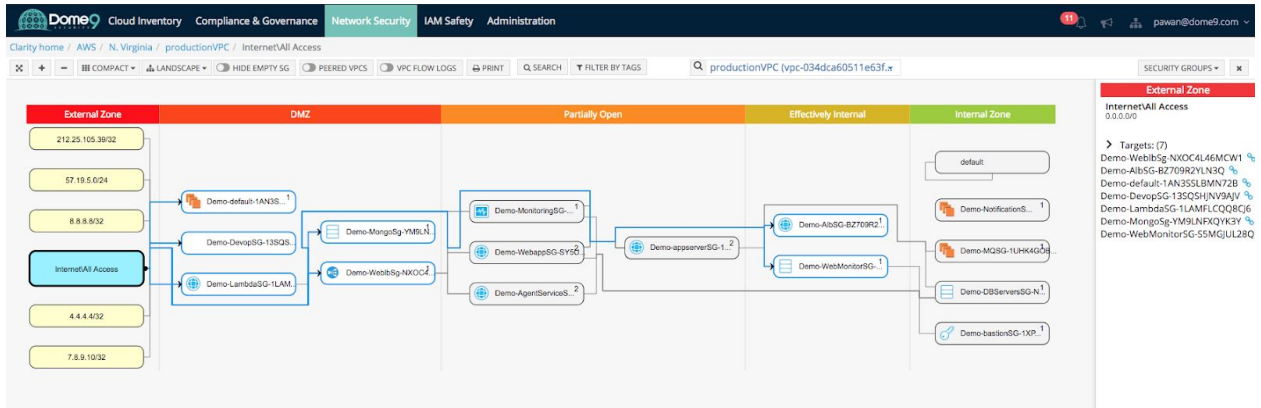
Dome9 Lab Guide



b) Click the **Legend** button to show what each icon represents.



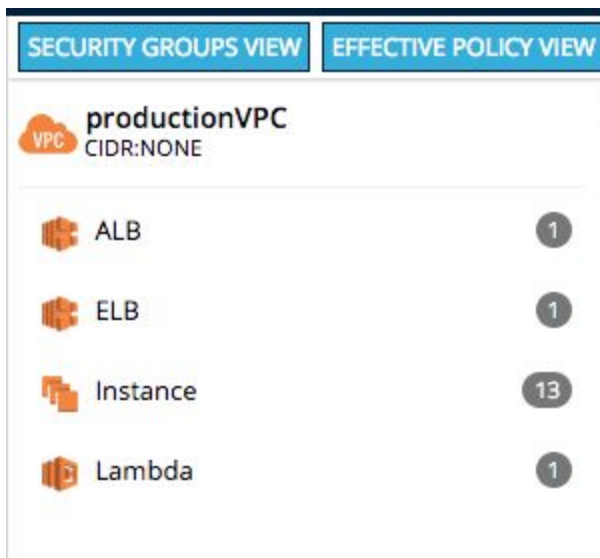
c) Click on **Internet/All Access** block. The source block is highlighted in the view, and the Security Groups that affect this source are highlighted.



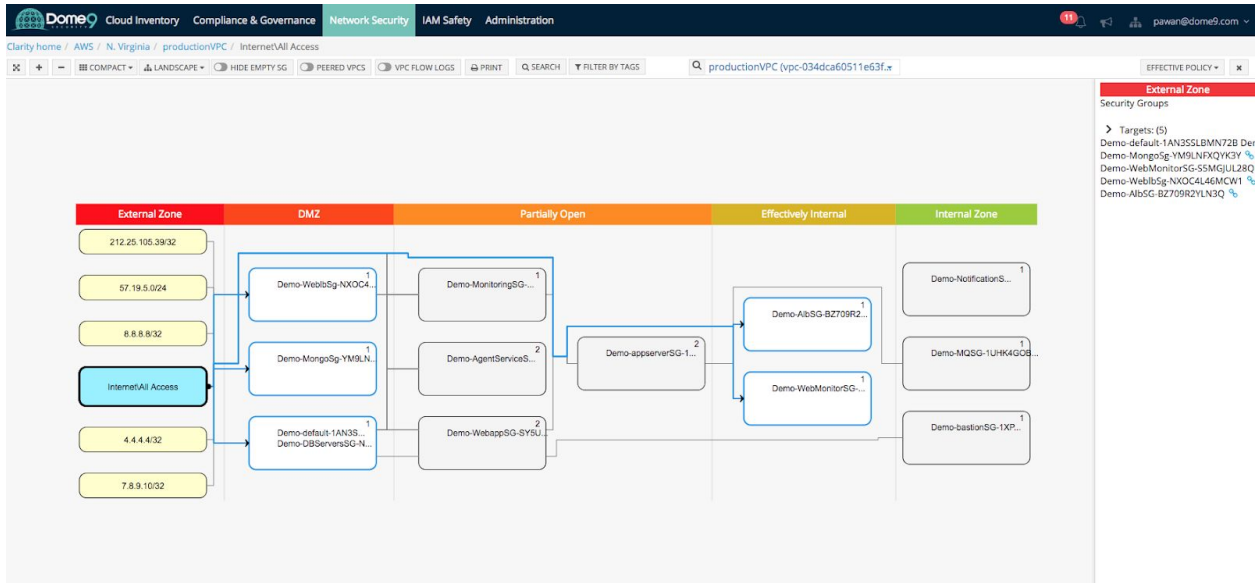
3) View an environment in the Effective Policy view

The Effective Policy view groups Security Groups that affect a common asset, and hides those that do not affect any assets.

- a) Select **Effective Policy** in the list in the menu bar at the upper right.



- b) This shows the VPC in the Effective Policy view



- c) This view also shows the Security Groups and Sources, organized by zone. In this view, however, the Security Groups that affect the same asset are grouped together. Security Groups that affect a number of assets may appear several times in the view. Security Groups that do not affect any assets are hidden.

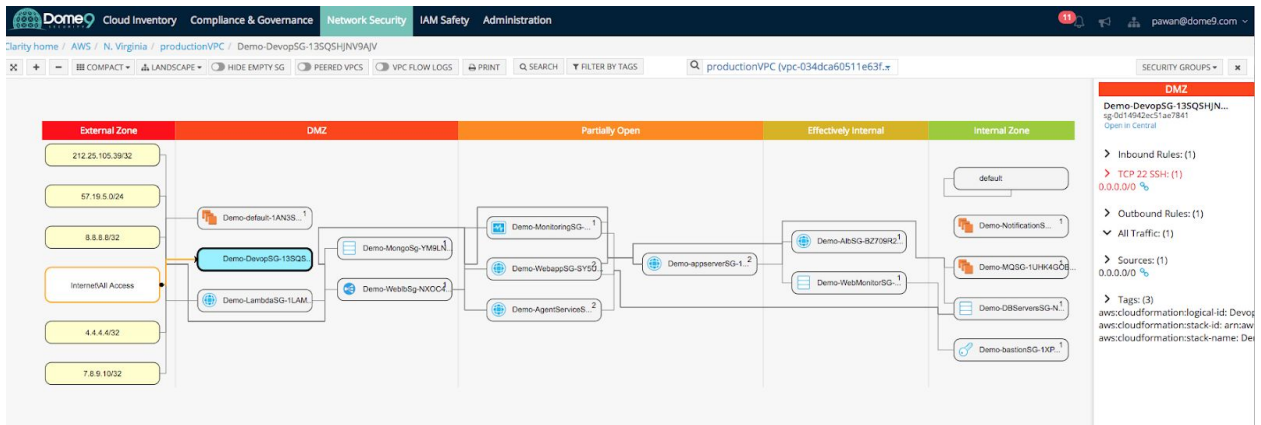
Exercise Complete! You have visualized your AWS security configurations in the VPC from a logical firewall view (SG view) and instance policy view (effective policy view)

Exercise 4.3: Identify zombie security group

Let's go back into Clarity's security group view. See if you can identify a zombie security group. This is a SG that has no instance attached to it but has an exposed policy (TCP 22 0.0.0.0/0) to the public.

Answer: Don't peek just yet..

Dome9 Lab Guide



Key	Value
aws:cloudformation:logical-id	DevopSG
aws:cloudformation:stack-id	arn:aws:cloudformation:us-east-1:290175429794:stack/Demo/052ae50095af11e6-a469-500c286e1a36
aws:cloudformation:stack-name	Demo

Name	Dome9 Description	Protocol/Port	State	Allowed Sources
SSH		TCP:22	Open	

This is a low severity issue yet is important to be aware of as it is just one click away from exposing your internal servers. With Clarity you can find such issues and take appropriate actions to fix such misconfigurations.

Exercise Complete! You have now identified the zombie security group exposure

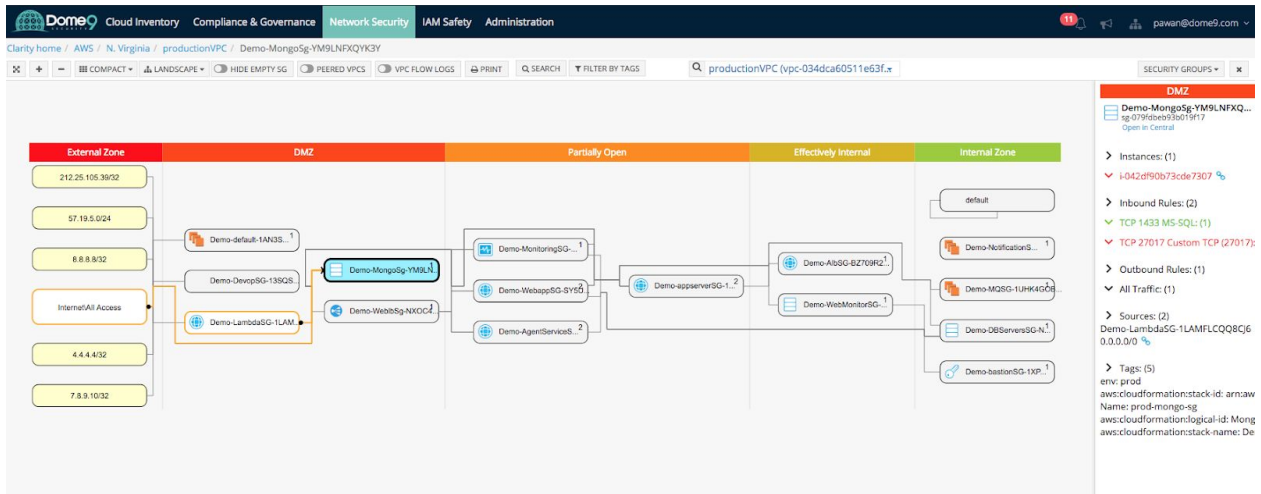
Exercise 4.4: Identify publicly accessible databases

In this section, we will investigate and find 2 different database exposures. Let's jump into Clarity.

Challenge 1: Detect DB Exposure – part 1 (easy)

In Clarity, try to find an exposed DB:

Answer: Here you see that this SG has 0.0.0.0/0 on port 27017 associated with the mongoDB instance. This is a high severity issue as one of your internal DB servers is now wide open. Click on the SG for more details.



AWS Security Group: Demo-MongoSg-YM9LNFXQYK3Y (sg-079fd90b73cde7307)

Account: AWS
 Region: N. Virginia (us_east_1)
 VPC: productionVPC (vpc-034dca60511e63f1db)
 Group Description: MongoDB security group

Tags

Key	Value
env	prod
aws:cloudformation:stack-id	arn:aws:cloudformation:us-east-1:290175429794:stack/Demo/052a6500-95af-11e8-a469-500c286e1a36
Name	prod-mongo-sg
aws:cloudformation:logical-id	MongoSg
aws:cloudformation:stack-name	Demo

Inbound Services

Name	Dome9 Description	Protocol/Port	State	Allowed Sources
Custom TCP (27017)		TCP:27017	Open	
MS-SQL		TCP:1433	Partial	Demo-LambdaSG-1LAMFLCQ8CJ6

Outbound Services: Default

Group Members

State	Name	Type	IP Address
Open	(i-042df90b73cde7307)	t2.nano	54.205.191.132 10.10.4.46

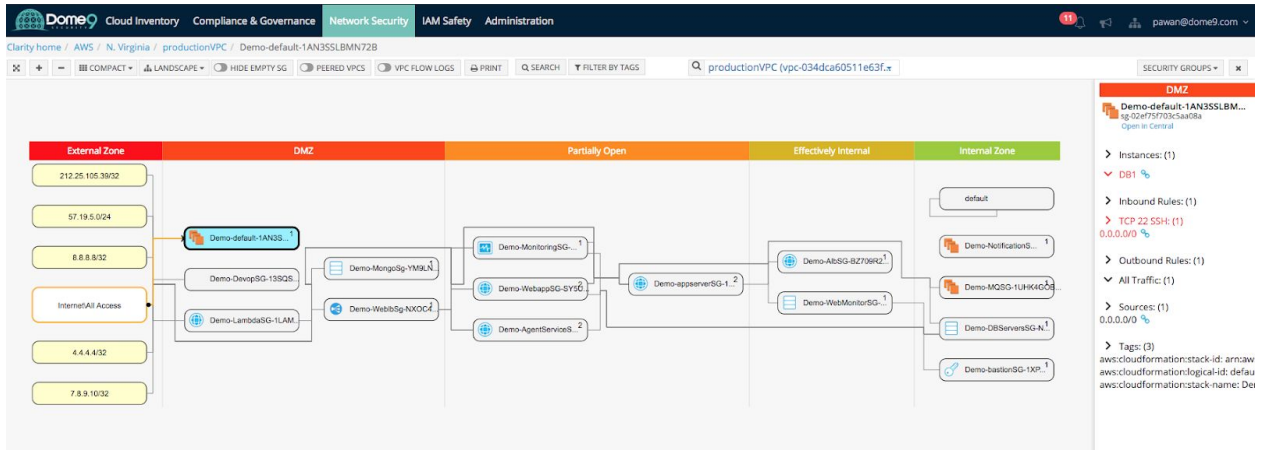
Referencing Security Groups: No referencing

You can see the open port in this view. We will fix these issues in the next lab.

Challenge 2: Detect DB Exposure – part 2 (hard)

Find another exposed DB, this one is a bit tricky.

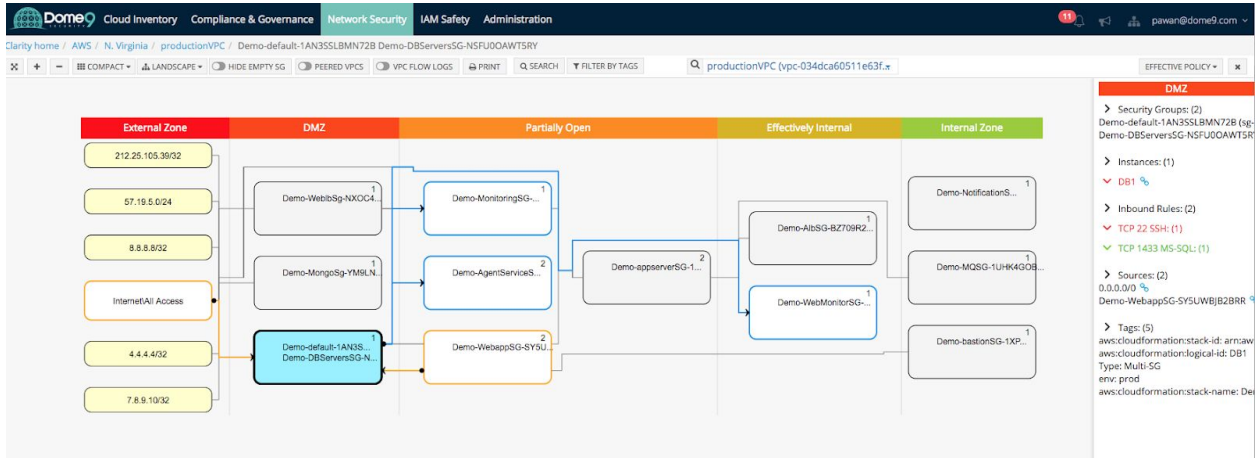
Answer: Click the default SG and hover over the right. Here you see default SG with SSH wide open and that DB1 is associated with this default SG making the database exposed to the public



In the effective policy view below – you can see default SG is in the DMZ zone and has one instance assignment (DB1, which is also part of the DB servers group).

Digging deeper, we realize that even if the rules associated with the security group are correct, an instance can still be assigned to the wrong security group. This is due to a misconfiguration that occurred due to assignment of multiple security groups, and specifically an incorrect default SG to DB1 instance.

The effective policy view brings this to light and tells you what security groups an instance belongs to, and therefore what the effective security policy is. Now it is clear – DB1 is exposed because it belongs to not only the internal DB Servers SG, but also to the Default SG which is in the DMZ.



Exercise Complete! You have now investigated both database exposures in Clarity

Security Posture Management Lab

Exercise 5.1: Enforce security policy

Fix misconfiguration in default security group by deleting the open SSH rule. For the purpose of this lab, lets keep it completely closed.

Now back in **Dome9 console** - go to Clarity, click the default SG we saw as the culprit, and turn on full protection mode at the top right in the entity explorer view:

Protection Mode Changed Successfully

AWS Security Group: Demo-default-1AN3SSLBMN72B (sg-02ef75f703c5aa08a)

Account: AWS
Region: N. Virginia (us_east_1)
VPC: productionVPC (vpc-d34dca60511e63fab)
Group Description: Default security group

Tags

Key	Value
aws:cloudformation:stack-id	arn:aws:cloudformation:us-east-1:290175429794:stack/Demo/052a6500-95af-11e8-a469-50c286e1a36
aws:cloudformation:logical-id	default
aws:cloudformation:stack-name	Demo

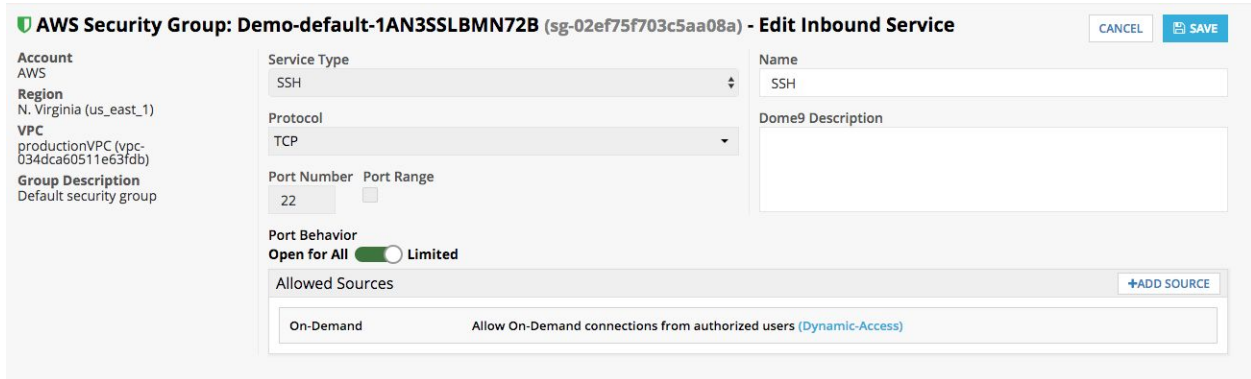
Inbound Services

Name	Dome9 Description	Protocol/Port	State	Allowed Sources
SSH		TCP:22	Open	

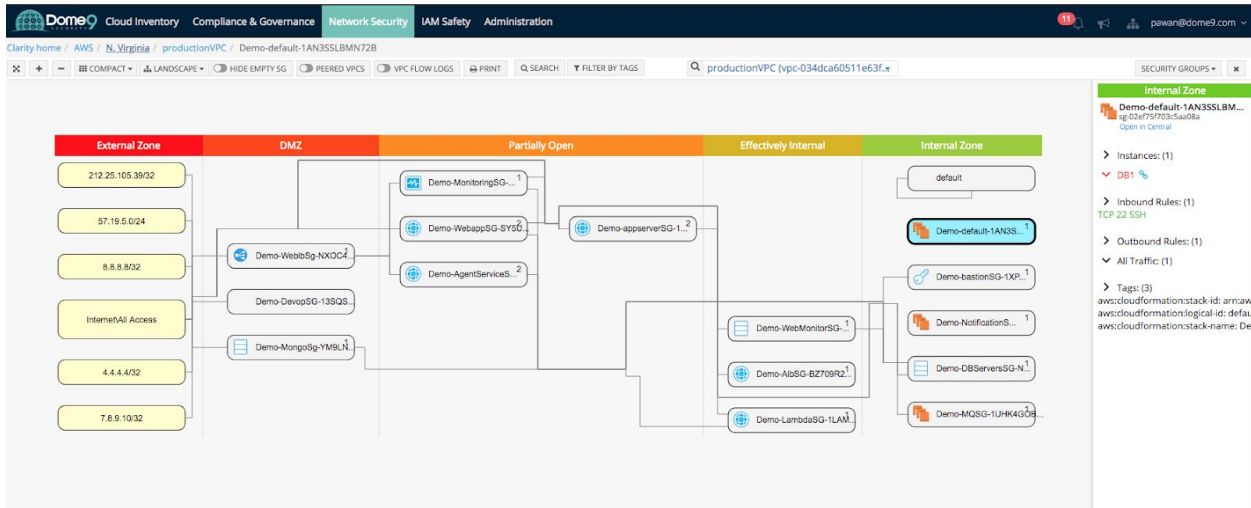
Group Members

State	Name	Type	IP Address
On	DB1 (i-0d11eaaaf78636450)	t2.nano	35.153.69.123 10.10.4.151

Fix the issue by closing the SG for now



You can also check Clarity to see the correct topology below:



Exercise Complete! You have now fixed the misconfigured security group.

Exercise 5.2: Active protection – SG level

Let's try to mess with this by making changes in the AWS console. Make SG change in AWS by going to the default SG.

The screenshot shows the AWS Management Console interface for creating a security group. The left sidebar contains navigation options like EC2 Dashboard, INSTANCES, IMAGES, ELASTIC BLOCK STORE, NETWORK & SECURITY, and LOAD BALANCING. The main content area is titled 'Create Security Group' and shows a list of existing security groups. The 'Default security group' (sg-02ef75f703c5aa08a) is selected. Below the list, the 'Inbound' tab is active, and a message states 'This security group has no rules'.

And now enable 0.0.0.0/0 SSH rule for default SG

The screenshot shows the 'Edit inbound rules' dialog box. A new rule is being added with the following details: Type: SSH, Protocol: TCP, Port Range: 22, Source: Custom 0.0.0.0/0, and Description: e.g. SSH for Admin Desktop. The 'Add Rule' button is visible, and a note at the bottom states: 'NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.' The 'Save' button is highlighted in blue.

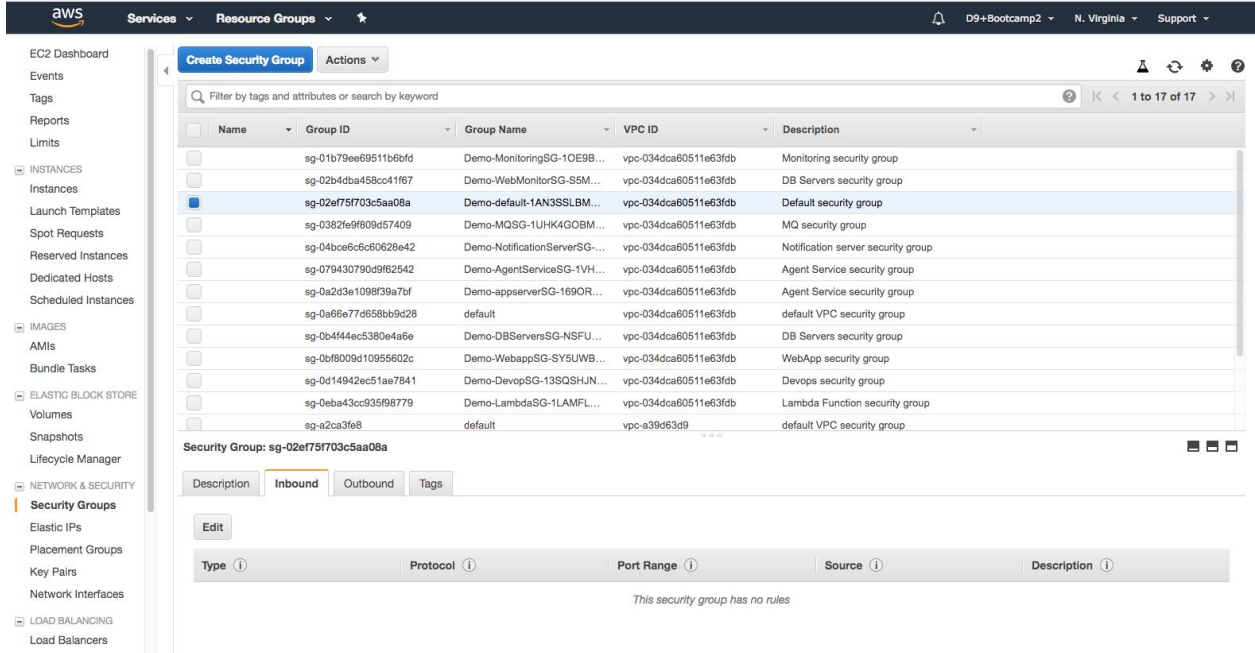
You should now be able to see the change in Dome9 console. Dome9 also rolls configuration back and adds the user event in the audit trail for further analysis. You can see the activity trail in the audit log which is extracted context from CloudTrail

Events			
Timestamp	Origin	User / IP Address	Description
2018-07-12 19:07:56	Dome9 Audit	system	Security group tamper detected and handled Security group tamper detected (Group modified). Security group: 'Demo-DBServersSG-1OXTU4VY4ZFT (sg-0bd91e7cf547a00ef)' of 'AWS>us_east_1>vpc-03de4c22bec2c90d3'. Handling method: Dome9 system has overridden the new settings with the approved policy. The following inbound rules were discovered: TCP-22-22: 0.0.0.0/0.

You should see the configuration reversed back to the gold standard configuration

The screenshot displays the AWS console interface for a security group. On the left, there's a sidebar with account and region information. The main area shows the security group's configuration, including tags and inbound services. The 'Inbound Services' section is expanded, showing a rule for SSH (TCP port 22) that is currently 'Closed'. The 'Group Members' section shows a single member, 'DB1', which is an EC2 instance with IP address 18.209.214.249.

You can also confirm it in the AWS console



Exercise Complete! You have now enabled enforcement of security policies and eliminate configuration drift at a security group level in AWS

Exercise 5.3: Active protection – Region level

Let's turn on Region Lock first in Dome9 (Lock down Sao Paulo region)

Go to the cloud inventory and select cloud accounts and navigate to Sao Paulo region.

Dome9 Lab Guide

The screenshot shows the AWS IAM console interface for Region Lock. On the left, account information is displayed: Account Number 290175429794, Added At Aug 1, 2018 10:24 AM, Total Instances 13, Total Full Protection Security Groups 1, and Total Read-Only Security Groups 31. The main area features a search bar and a table of regions. Each row in the table includes the region name, its detection mode (all are 'Read-Only (Monitor mode)'), the number of instances, and the number of full and read-only security groups. An 'EDIT' button is present for each region.

Region	Detection Mode	Instances	Security Groups	
Canada Central	Read-Only (Monitor mode) ⓘ	0	0 Full protection 1 Read-Only	EDIT
Frankfurt	Read-Only (Monitor mode) ⓘ	0	0 Full protection 1 Read-Only	EDIT
Ireland	Read-Only (Monitor mode) ⓘ	0	0 Full protection 1 Read-Only	EDIT
London	Read-Only (Monitor mode) ⓘ	0	0 Full protection 1 Read-Only	EDIT
Mumbai	Read-Only (Monitor mode) ⓘ	0	0 Full protection 1 Read-Only	EDIT
N. California	Read-Only (Monitor mode) ⓘ	0	0 Full protection 1 Read-Only	EDIT
N. Virginia	Read-Only (Monitor mode) ⓘ	13	1 Full protection 16 Read-Only	EDIT
Ohio	Read-Only (Monitor mode) ⓘ	0	0 Full protection 1 Read-Only	EDIT
Oregon	Read-Only (Monitor mode) ⓘ	0	0 Full protection 1 Read-Only	EDIT
Paris	Read-Only (Monitor mode) ⓘ	0	0 Full protection 1 Read-Only	EDIT
Seoul	Read-Only (Monitor mode) ⓘ	0	0 Full protection 1 Read-Only	EDIT
Singapore	Read-Only (Monitor mode) ⓘ	0	0 Full protection 1 Read-Only	EDIT
Sydney	Read-Only (Monitor mode) ⓘ	0	0 Full protection 1 Read-Only	EDIT
São Paulo	Read-Only (Monitor mode) ⓘ	0	0 Full protection 2 Read-Only	EDIT

Let's assume this would be an unused foreign region where you would not expect activity happening. You can turn on Region Lock to ensure no changes are made to this region unless its from within Dome9.

Dome9 Lab Guide

Edit Region - AWS - São Paulo

Newly Detected Groups Behavior



Read-Only (Monitor mode)

Newly detected Security Groups and their rules will be imported and will be set as 'Read-Only' Security Groups. These groups will be monitored by Dome9 for changes but can still be changed outside Dome9.



Full protection (Dome9 managed)

Newly detected Security Groups will be imported with their initial rules to the Dome9 service, and thereafter be treated as 'Full Protection (Dome9 Managed)' Security Groups. Any further security policy changes should be made using the Dome9 console or API. Dome9 Policy Tamper Protection will be added to the Security Group and will be activated after the import.



Region lock

Newly detected Security Groups will be imported and will immediately have their rules cleared (deleted) on both ingress and egress. This mode ensures that no network changes will be made to the region, unless they are performed on the Dome9 console.

Note: This will delete all ingress and egress rules from any newly discovered Security Groups. This mode is recommended for highly secured setups that are not allowed to be changed outside of the Dome9 system. This mode is also highly recommended for inactive regions.

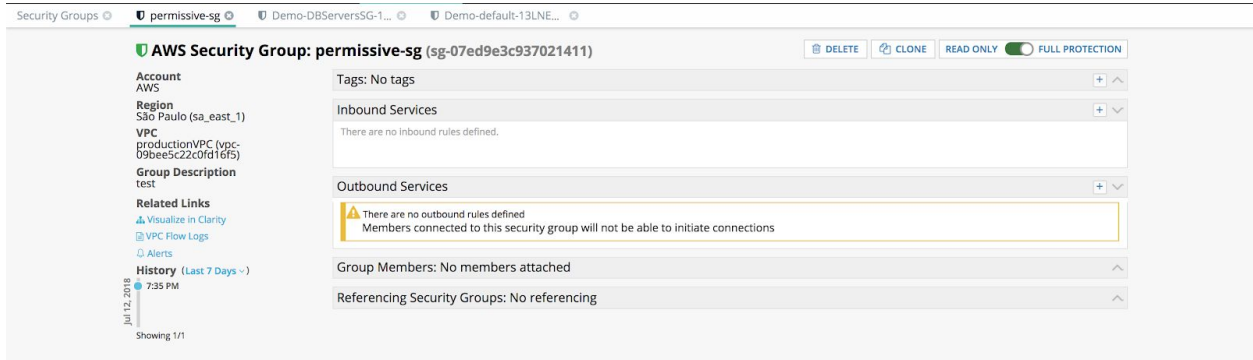
São Paulo select entire region

	Read-Only (Monitor mode) ? select all	Full protection (Dome9 managed) ? select all
productionVPC (vpc-09bee5c22c0fd16f5)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
permissive-sg (sg-07ed9e3c937021411)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
default (sg-032df64f273446774)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Demo-SandBox-bastionSG-2ZQ77PNH45BA (sg-0170baf9218b4da06)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
vpc-c896f0af	Read-Only (Monitor mode) ? select all	Full protection (Dome9 managed) ? select all
default (sg-ff45a086)	<input checked="" type="checkbox"/>	<input type="checkbox"/>
SG1 (sg-0357eeab77f385426)	<input checked="" type="checkbox"/>	<input type="checkbox"/>

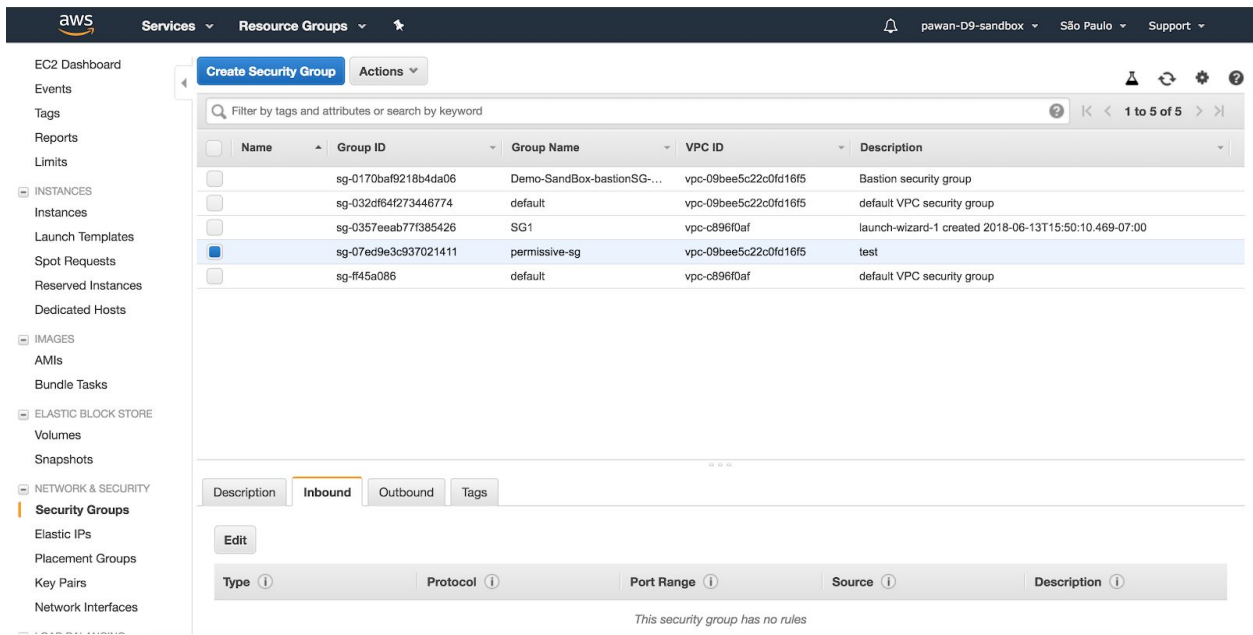
CLOSE SAVE

Go to the AWS console and create a new SG in Sao Paolo

Go back to Dome9 console, you should see this new SG's inbound/outbound rules deleted.



You can also confirm it in the AWS console in Sao Paolo



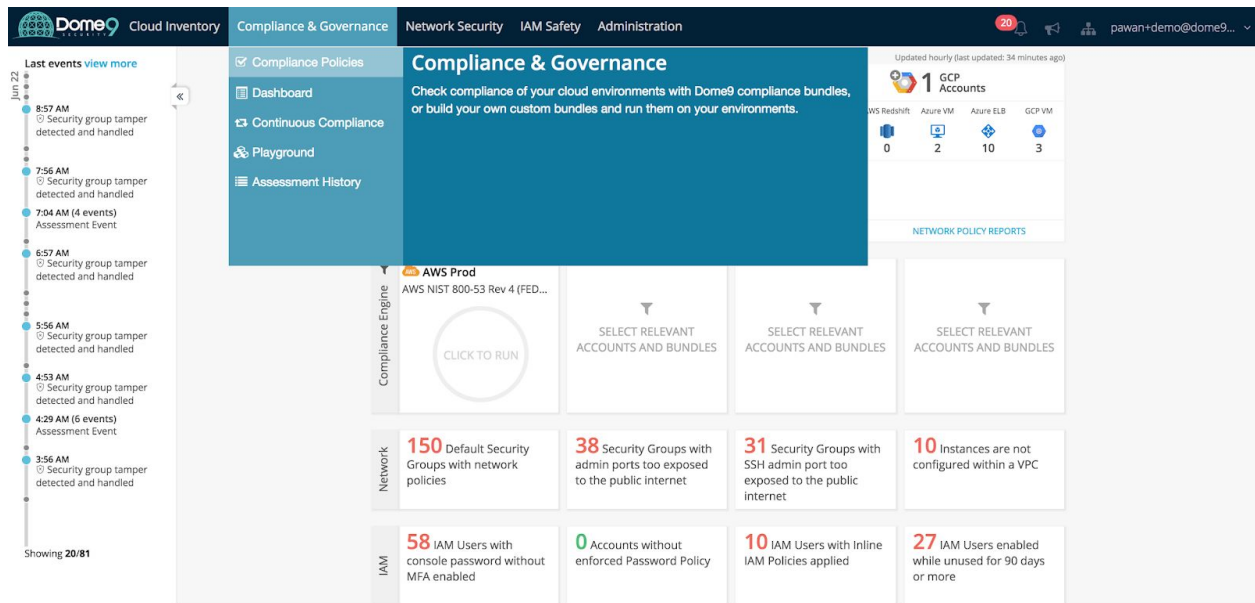
Exercise Complete! You have now enabled enforcement of security policies and eliminate configuration drift at a region level in AWS.

S3 Security Lab

The goal of this exercise is to help you understand how to control S3 bucket access. We will focus on how to ensure specific role can interact with S3 and thereby not allowing any anonymous/outside users have access to sensitive data in the bucket. We will also look at least privilege concept, where you as an admin decide who has access (whitelist) to the most sensitive S3 operations and deny everyone else.

Exercise 6.1: Detecting Exposed Buckets

1. Navigate to Compliance Engine Tab



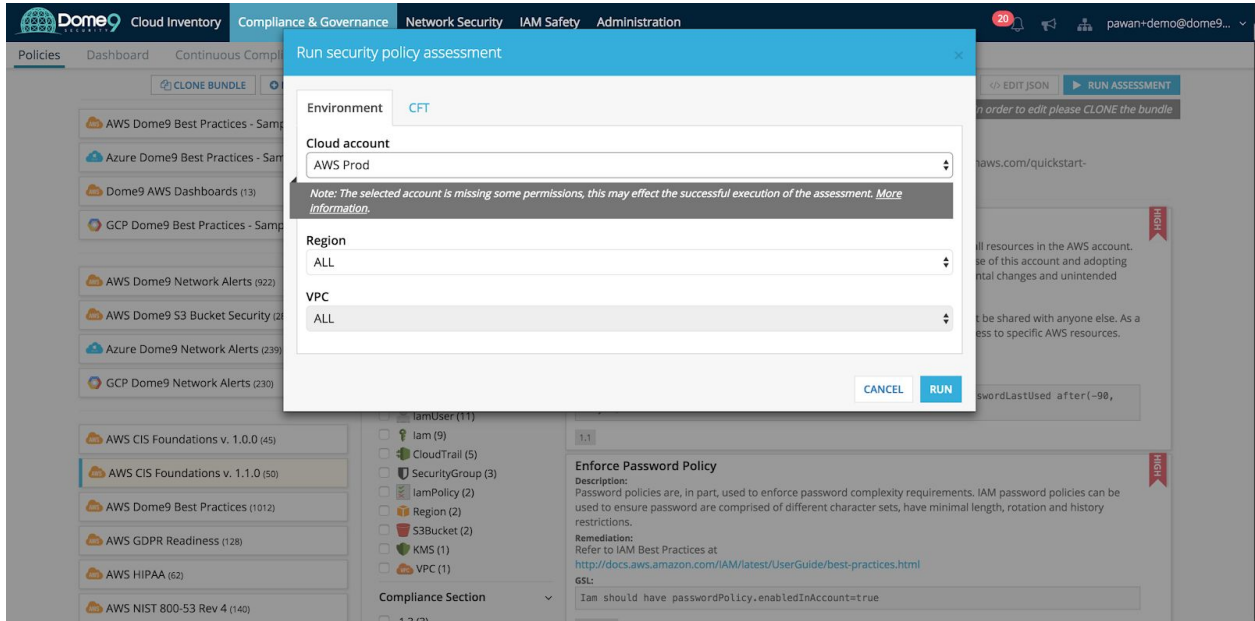
2. Explore Compliance Engine page

Dome9 Lab Guide

The screenshot displays the Dome9 Compliance & Governance interface. The top navigation bar includes 'Policies', 'Dashboard', 'Continuous Compliance', 'Playground', and 'Assessment History'. The main content area shows a list of bundles on the left, with 'AWS CIS Foundations v. 1.1.0' selected. The right pane provides details for this bundle, including a description, severity levels (Medium, High, Low), validation types (List<CloudTrail>, iamUser, iam, CloudTrail, SecurityGroup, iamPolicy, Region, S3Bucket, KMS, VPC), and compliance sections. Two specific findings are highlighted with red 'HIGH' severity tags: 'Avoid the use of the 'root' account' and 'Enforce Password Policy'. Each finding includes a description, remediation steps, and a corresponding GSL (Global Security Language) rule.

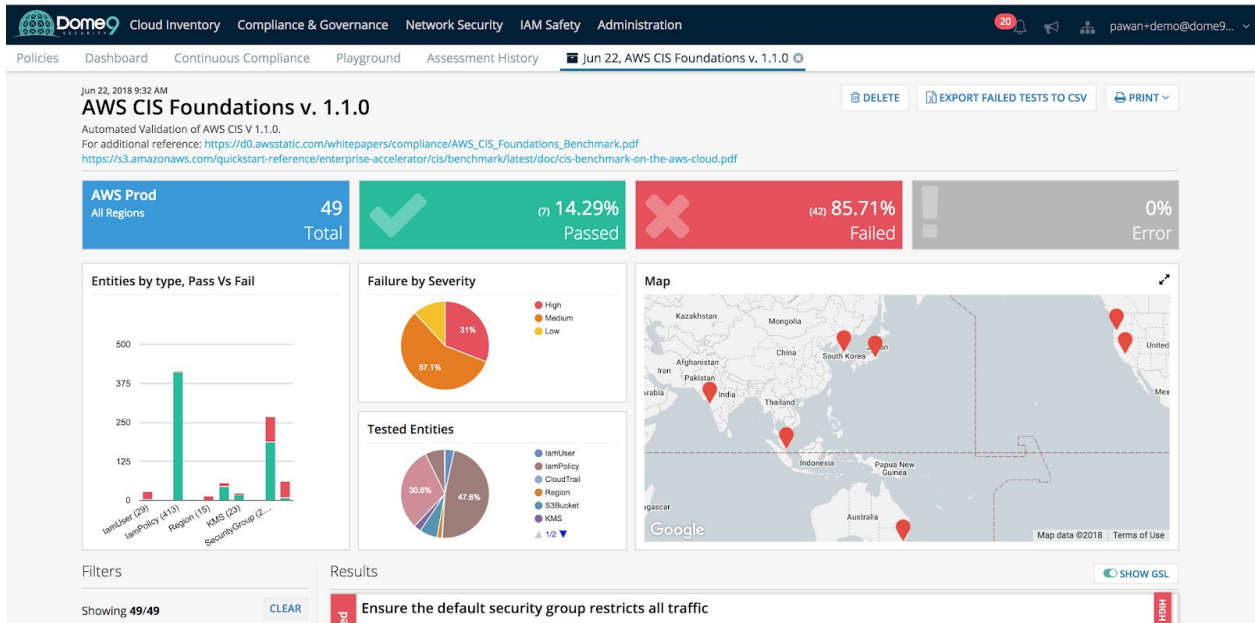
Dome9 Compliance Engine monitors and scans your AWS/Azure/GCP infrastructure to ensure alignment with compliance standards such as PCI-DSS, NIST 800-53, GDPR, CIS etc.

3. Select CIS Benchmarks and run bundle assessment



Select your AWS environment in the cloud account section. Dome9 also provides the option to BYOC (Bring your own CFT) and scan your template before pushing to production.

5. Analyze the report findings



Within a few seconds, the Compliance Engine runs an assessment and provides detailed findings in an easy to use dashboard. You can see the number of test cases passed vs failed, as well as test cases segmented geographically, across regions and by test severities.

6. Filter Analyze key test results for **S3 buckets**

1. Ensure buckets are not publicly accessible

How many buckets are publicly accessible?

Exercise Complete! You have now detect publicly exposed buckets!

Exercise 6.2: S3 Access Controls (exposed ACLs)

Navigate to the S3 console in AWS console. We will start with bucket 1

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with 'aws' logo, 'Services', 'Resource Groups', and a search icon. Below that, a banner for 'Stream Video to AWS for Analytics' is visible. The main content area shows the 'Amazon S3' console with a search bar and buttons for 'Create bucket', 'Delete bucket', and 'Empty bucket'. A summary bar indicates '6 Buckets', '2 Public', and '1 Regions'. Below this is a table listing buckets with columns for 'Bucket name', 'Access', 'Region', and 'Date created'. The first two buckets are marked as 'Public', while the others are 'Not public *'. A footer note states: '* Objects might still be publicly accessible due to object ACLs. Learn more'.

Bucket name	Access	Region	Date created
awsloft-s3bucket1-rohw1yk24pf	Public	US East (N. Virginia)	Aug 21, 2018 5:38:51 PM GMT-0400
awsloft-s3bucket2-si1m6gzusrf	Public	US East (N. Virginia)	Aug 21, 2018 5:38:50 PM GMT-0400
awsloft-s3bucket3-e3vxx6glt6uf	Not public *	US East (N. Virginia)	Aug 21, 2018 5:38:50 PM GMT-0400
awsloft-s3bucket4-1jpyh3i02a98z	Not public *	US East (N. Virginia)	Aug 21, 2018 5:38:51 PM GMT-0400
awsloft-s3bucket5-4mp0szrrf0kn	Not public *	US East (N. Virginia)	Aug 21, 2018 5:38:51 PM GMT-0400
cf-templates-ob34rv71lzd-us-east-1	Not public *	US East (N. Virginia)	Aug 19, 2018 2:03:30 PM GMT-0400

Fix <yourCFT>S3Bucket1

Exercise 6.3: S3 Access Controls (Bucket policies)

Now let's move to <yourCFT>S3Bucket2. Write JSON policy (start from the existing configuration seen below)

Amazon S3 > demo-s3bucket2-1fras4mi5msbu

Overview Properties **Permissions** Management

Access Control List **Bucket Policy** CORS configuration

Bucket policy editor ARN: arn:aws:s3:::demo-s3bucket2-1fras4mi5msbu
Type to add a new policy or edit an existing policy in the text area below.

Delete Cancel Save

```

1 {
2   "Version": "2012-10-17",
3   "Id": "MyPolicy",
4   "Statement": [
5     {
6       "Sid": "Put",
7       "Effect": "Allow",
8       "Principal": "*",
9       "Action": "s3:GetObject",
10      "Resource": "arn:aws:s3:::demo-s3bucket2-1fras4mi5msbu/*"
11    }
12  ]
13 }

```

Documentation Policy generator

Bucket Policy: Access Management - Only Allow **S3-Role** to put and get objects in this bucket.

Sample snippet for Principal parameter:

```

"Principal": {
  "AWS": "arn:aws:iam::<ACCOUNT NUMBER>:role/<INSERT S3 ROLE NAME HERE>"
},

```

Hint: copy/paste the Principal parameter to include S3 role name. It can be found by navigating to **Services -> IAM -> Roles -> <your S3 Role Name>**

Bucket Policy: Least Privilege enforcement - Ensure deletion of S3 bucket can only be done based on your specific AWS account id (this is important for sensitive buckets that may have other accounts accessing it, and you want to make sure the most critical operations are whitelisted)

Sample JSON to add in your bucket policy (don't forget to add , after previous statement)

```
,
{
  "Sid": "Stmnt1526361042800",
  "Effect": "Deny",
  "Principal": "*",
  "Action": "s3:DeleteBucket",
  "Resource": "arn:aws:s3:::<copy your S3 bucket resource name in existing policy>",
  "Condition": {
    "StringNotLike": {
      "aws:userid": "YOURAWSACCOUNTID"
    }
  }
}
```

Exercise Complete! You have now ensured no buckets are publicly exposed!

Offboarding

Please navigate to the Cloudformation and click the stack and delete stack. You are now done!