TX Prevent: Installing TX Prevent on AWS EC2

CloudFormation

Version 1.0, 2024-10-02

Introduction

This document will guide you through an installation of TX Prevent into your AWS environment by using CloudFormation.

Architecture

The TX Prevent deployment architecture leverages several Amazon Web Services (AWS) components to provide a highly available and secure product.

Runtime sensors will deploy onto EC2 instances alongside the applications or services you want to watch. These sensors communicate with the {ThreatX} Prevent control plane services.

□ Threa□tX Control Plane Services

Runtime Analyzer

A data aggregator, analysis engine, and event router. Connects to ThreatX and emits vulnerability metadata.

Scan Template Service

Ingests passively detected vulnerability data and generates highly targeted scan templates. Executes individual scans are executed and returns the results after determining efficacy.

OTEL Collector

Coming soon! This service will collect logs and metrics from the sensors and other control plane services and send them back to ThreatX for enhanced product support.

Context Diagram

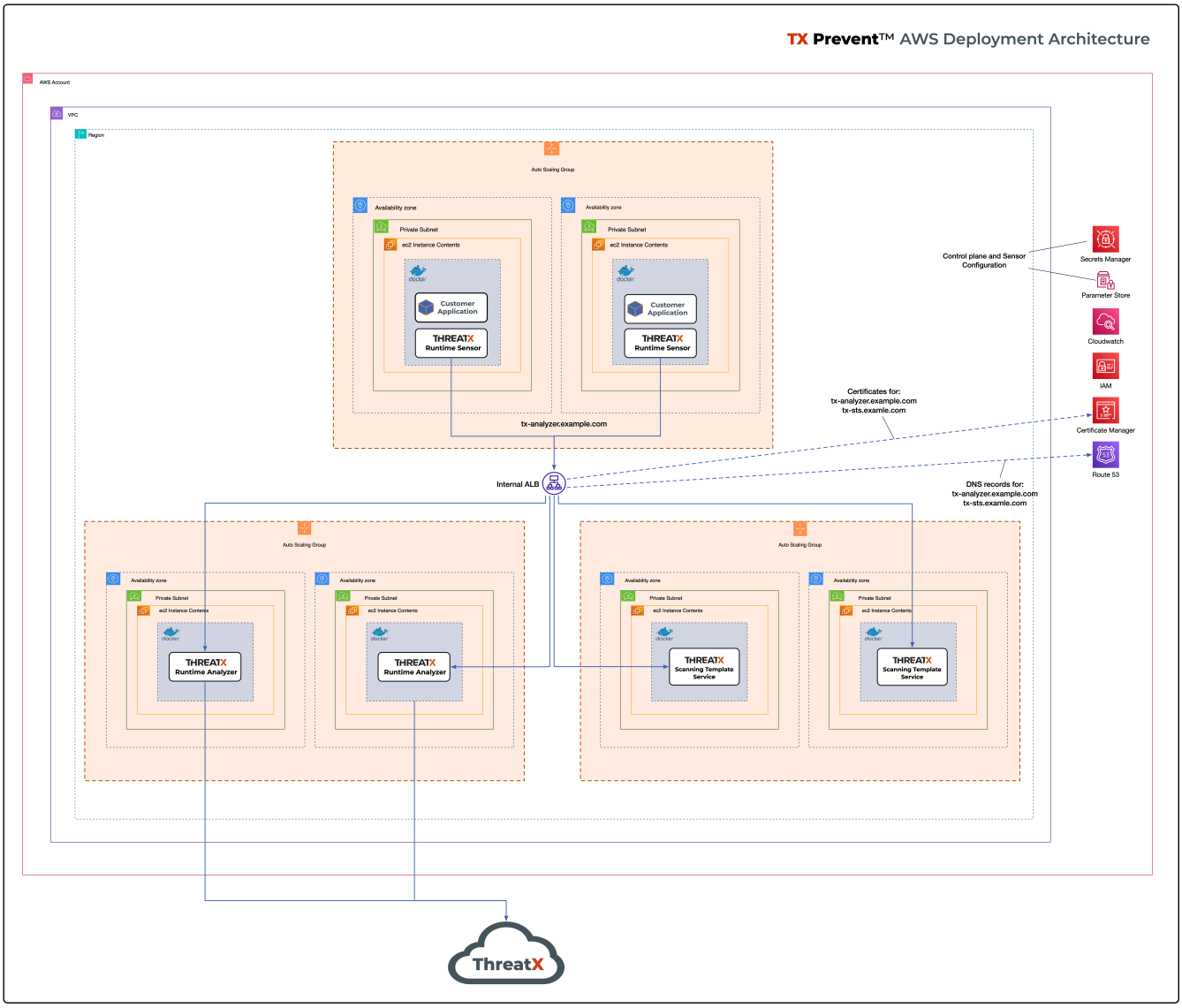

Figure 1. TX Prevent Deployment Architecture

High Availability

- For each control plane service, instances are created in multiple availability zones
- The instances are deployed in Auto Scaling Groups (ASG) where they are continuously monitored to ensure the desired number of healthy instances

Security

- All control plane services are deployed into private subnets and are never publicly exposed
- All traffic to Control plane services is encrypted using TLS with Amazon provisioned certificates

AWS Components and Services

Application Load Balancer (ALB)

Fronts the ThreatX Prevent control plane services. Each control plane service has multiple instances in at least *two* availability zones for high availability with the ALB distributing traffic between them.

Auto Scaling Group (ASG)

Maintains the desired number of healthy service EC2 instances. If an instance becomes unhealthy or is unexpectedly terminated the ASG will create another instance.

Parameter Store

Configuration properties for sensors and control plane services.

Secrets Manager

Sensitive configuration properties.

Route53

DNS records for the control plane services.

Amazon Certificate Manager (ACM)

Provisioning certificates for the control plane services.

CloudWatch

Log aggregation of all logging event from the ThreatX Prevent sensors and control plane services. Provides for the querying of the log information.

Planning

This deployment requires familiarity with the following AWS services:

- Amazon VPC
- AWS CloudFormation
- AWS Route53

Only for deployments requiring VPC connectivity between the ThreatX Prevent VPC and other VPC containing monitored application/service:

- AWS VPC Peering
- AWS Transit Gateway

Prerequisites

Preflight Checklist

The following items must be completed before the deployment can begin.

- ☐ **Valid ThreatX Tenant ID** (customer name)
- ☐ **Valid ThreatX API Key** (See: ThreatX Sensor API Key)
- ☐ **AWS user or role** with either the **AdministratorAccess** policy or our custom deployment IAM policy
- ☐ **EC2 key pair** for SSH access to the EC2 instances. (See EC2 Key Pair)
- ☐ **Docker** installed on the EC2 instances where the sensors will be deployed
- ☐ **AWS Route53 Hosted Zone** for DNS records and certificates of control plane services
- ☐ *VPC* with at least:
  - **2** private subnets
  - **1** public subnet
  - **1** internet gateway
  - **1** NAT gateway

Basic VPC Topology

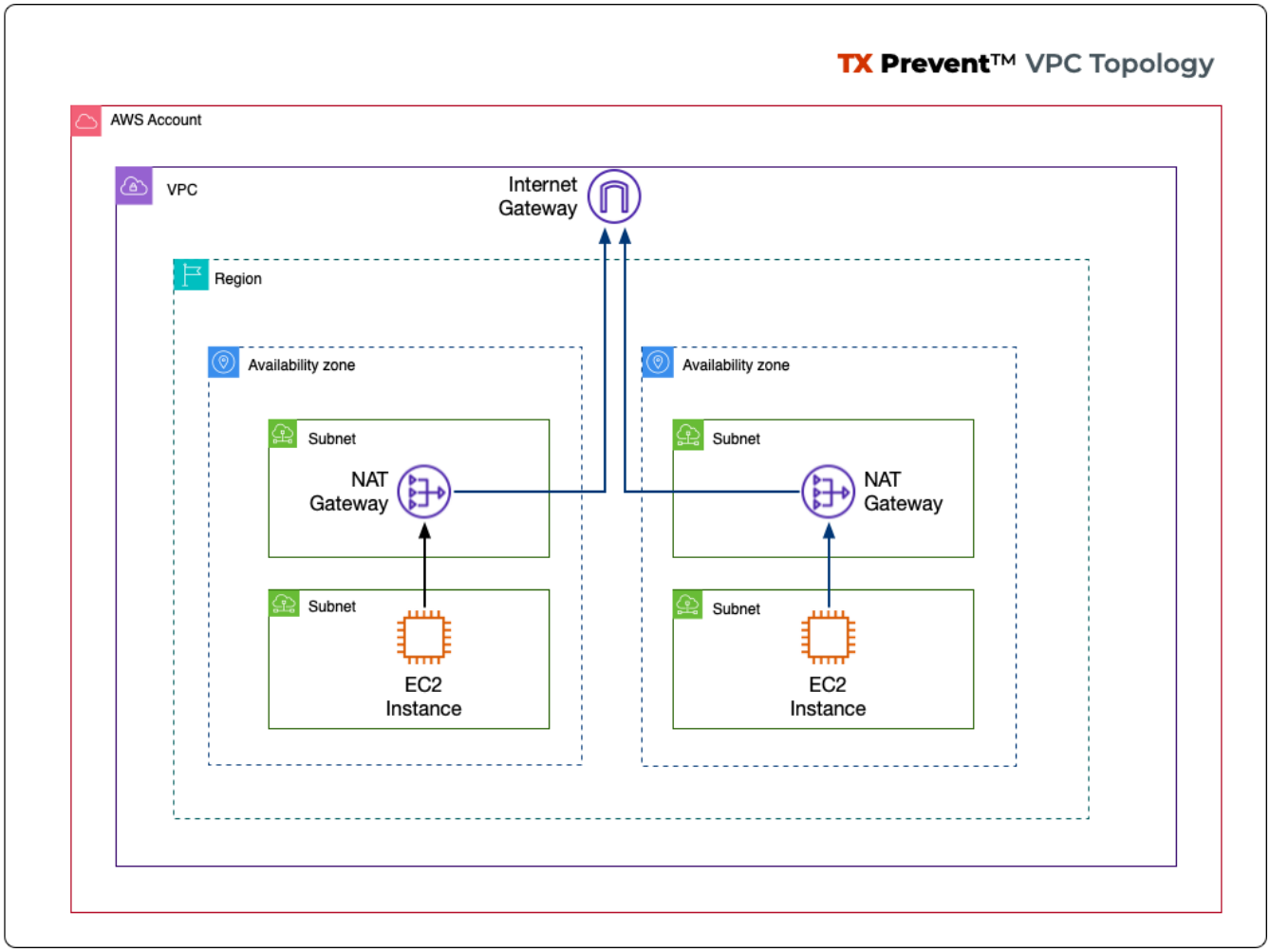

Figure 2. TX Prevent Standard VPC Topology

Creating an EC2 Key Pair

The EC2 Key pair will be used to SSH into the ThreatX Control Plane EC2 instances. To create one for the install follow the steps below:*

1. Open the **AWS EC2 Console**.
2. Select **Main Menu (left) › Network & Security › Key Pairs**
3. On the **Key pairs** page, click **[Create key pair]**
4. On the **Create Key Pair** page:
   a. Enter a name (e.g., *<threatx-prevent>*)
   b. Select **RSA**
   c. Select **.pem** format
   d. Add any **Tags** that you want
   e. Click on **[Create key pair]**.

The private key will then be downloaded to your system.

Put this key in a safe place. It can be used to SSH into any of ThreatX Prevent EC2's.

Runtime Sensor System Requirements

Resources

It is recommended to have **at least 2 cores** and **300MB of memory** available on the EC2 instance that they will be running on.

Network Connectivity

If Sensors are deployed into a *different VPC than that of the control plane*, VPC peering or Transit Gateway connectivity will need to be setup between the VPCs.

Scanning Requirements

You may need to adjust security groups to allow ingress traffic from the Scan Template Service to the target endpoints.

Control Plane Deployment

Get The CloudFormation Template

Download the ThreatX Prevent CloudFormation template - threatx-prevent.yaml.

Template Parameters

▼ *(show/hide)* **ThreatX Prevent CloudFormation Template Parameters**

Table 1. AWS Properties

Key	Type	Default	Description
VPC	String		A virtual private cloud (VPC) to install into. See VPC Setup
Subnets	List<String>		At least two private subnets in different Availability Zones in the selected VPC
HostedZoneId	String		The ID of the Hosted Zone in Route53 to add DNS record to. Must align with the specified Hosted Zone Name.
HostedZoneName	String		The Hosed Zone Name in Route53 for the control plane service DNS records. Must align with the specified Hosted Zone Id.
KeyName	String		Name of an existing EC2 key pair to allow SSH access to the control plane's EC2 instances

Table 2. Product Configuration

Key	Type	Default	Description
TenantId	String		The Tenant ID for ThreatX Prevent
ApiKey	String		The API key for ThreatX Prevent
GatewayHostname	String	threatx-gateway-production-v1.xplat-production.threatx.io	The Gateway hostname for ThreatX Prevent
AnalyzerTags	String		The tag values for the Runtime Analyzer
AnalyzerDesiredInstances	Number	2	Number of desired Runtime Analyzer instances

Key	Type	Default	Description
AnalyzerImageTag	String	`1.0.1`	The tag for the Runtime Analyzer docker image
AnalyzerInstanceType	String	`t3.small`	The EC2 instance type for the Runtime Analyzer instances
StsDesiredInstances	Number	`2`	Number of desired Scan Template Service instances
StsImageTag	String	`1.0.1`	The tag for the Scan Template Service docker image
StsInstanceType	String	`t3.small`	The EC2 instance type for the Scan Template Service instances
LogLevel	String	`info`	The logging level to use for all services

Deployment Steps

1. Sign in to your AWS account via the AWS Console. Select the desired region for the deployment.
2. Open the **CloudFormation** console
3. Select **[Create stack]** and **[With new resources (standard)]**
4. Select **[Choose an existing template]**. Then add the URL for the ThreatX Prevent template to the **Amazon S3 URL** field:

```
https://threatx-prevent-cf-template.s3.amazonaws.com/threatx-prevent.yaml
```

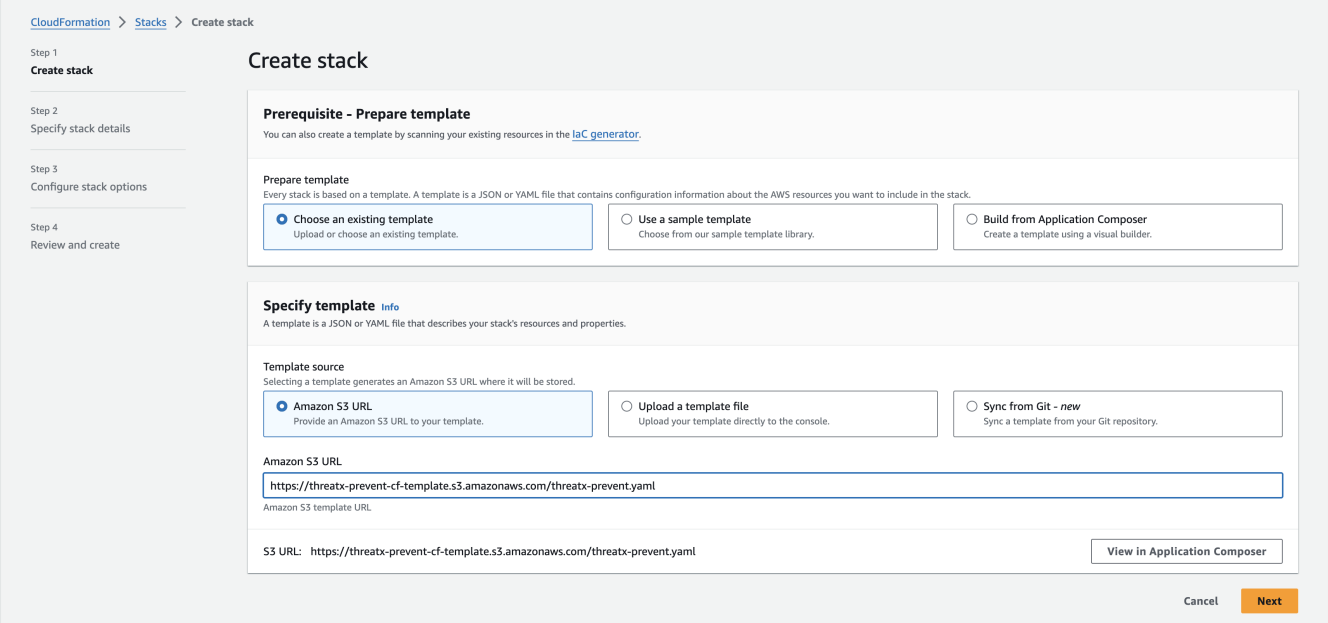

Figure 3. ThreatX Prevent Standard VPC Setup

1. On the **Specify stack details** page:
   a. For the **Stack Name**, enter: *ThreatXPrevent*
   b. Review all the parameters (Template Parameters) for the template. Provide values for the parameters that require input. For all other parameters, review the default settings and customize them as necessary. When you are finished, select **[Next]**.
2. On the **Configure Stack Options** page:
   a. (optional) Specify tags for the resources in your stack and set any advanced options you want.
   b. When you finish, choose **[Next]**.
3. On the **Review** page:
   a. Review and confirm all of the template settings.
   b. Under **Capabilities**, review and select the check boxes to acknowledge.
   c. Choose **[Create Stack]**.

The ThreatX Prevent deployment is ready when the stack status is **CREATE_COMPLETE**. Stack creation should take 5 to 10 minutes.

You can watch creation events under the **Event** tab. To view all the created resources, choose the **Outputs** tab.

Runtime Sensor Deployment

Sensor IAM Policy

The Runtime Sensor will try push its log information into a CloudWatch group that was created during the Control Plane Deployment: *<cloudformation-stackname>-ThreatXPrevent-sensor*

To do so, the EC2 instance that the sensor is running on will need to have the following IAM policy attached to its role:

- **<cloudformation-stackname>-sensor-log-policy**

Launch the ThreatX Prevent Sensor

```
docker run -i -p 80:80 -p 50051:50051 \
  --network host \
  --log-driver=awslogs \
  --log-opt awslogs-region=us-east-1 \
  --log-opt awslogs-group=<cloudformation-stackname>-ThreatXPrevent-sensor \ ①
  --mount type=bind,source=./AmazonRootCA1.pem,target=/AmazonRootCA1.pem \ ②
  --cap-add=NET_ADMIN \
  --cap-add=SYS_ADMIN \
  -e SENSOR_TAGS=raap-example.raap-example-deployment \  ③
  -e INTERFACE=<see table below> \ ④
  -e RUST_LOG=info \
  -e RUST_BACKTRACE=1 \
  -e ANALYZER_URL=https://tx-analyzer.xplat-sandbox.threatx.io:50051 \
  -e ANALYZER_TLS_ENABLED=true \
  -e TARGET_ENVIRONMENT=docker \
  -e ANALYZER_TLS_CA_PEM=./AmazonRootCA1.pem \
  -v /sys/kernel/tracing:/sys/kernel/tracing:ro \
  public.ecr.aws/threatx/raap/threatx-runtime-sensor:1.0.0
```

① The CloudWatch log group name must match the name of the log group created by the CloudFormation stack for the ThreatX Prevent sensor logs.

② The Amazon CA certificate must be mounted into the container for the sensor to trust the control plane certificates. Download: www.amazontrust.com/repository/AmazonRootCA1.pem

③ For the most accurate tracking of events at the application level the ThreatX Prevent sensor needs to derive the name of the application that it is monitoring on the EC2 instance. This should be set the name of the application that this sensor is working alongside.

④ The network interface name must match the name of the network interface for the EC2 instance that the sensor is running on. See the table below for the correct name for your distribution.

Network Interface Names for Common Linux Distributions

Table 3. Network Interfaces

Distribution	Interface
Amazon Linux 2023	enX0

Distribution	Interface
Amazon Linux 2	eth0
Ubuntu	enX0
SUSE	eth0
Debian	enX0
RHEL	eth0

If your distribution is not listed, you can find the correct interface name by running the `ip a` command on the EC2 instance.

CloudWatch Logs

The following CloudWatch log groups will be created. They will collect all log output from the ThreatX Prevent sensors and all instances of the control plane services:

- `<cloudformation-stackname>-ThreatXPrevent-sensor`
- `<cloudformation-stackname>-ThreatXPrevent-analyzer`
- `<cloudformation-stackname>-ThreatXPrevent-sts`

CloudFormation IAM Permissions

There are two options for obtaining the permissions needed to create the ThreatX Prevent stack:

1. Using an existing user or role with the **AdministratorAccess** policy
2. Creating a new custom IAM policy with the minimum required permissions according to least privilege which will be assigned to the existing user or role you want to use for installation (continue reading next section)

Configure AWS with the Minimum Permissions Required for Stack Creation

Now we will create a custom policy with the minimum permissions required to create the ThreatX Prevent stack.

Create a Custom Policy

- On the **AWS Services** page, Select **[IAM]**.
- From **IAM Dashboard**, select **Main Menu (left)** › **Policies**
- On the **Policies** page, Select **[Create policy]**
- On the **Specify Permissions** page, under the **JSON** tab:
  1. Copy the JSON below into the Policy editor.
  2. Replace all placeholder instances with your actual values:
     - `<account-id>` with your *AWS Account ID*
     - `<hosted-zone-id>` with your *AWS Route53 Hosted Zone ID*

tx-prevent-cf-iam-policy.json

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Logs2",
            "Effect": "Allow",
            "Action": "logs:DescribeLogGroups",
            "Resource": "*"
        },
        {
```

```
            "Sid": "Logs",
            "Effect": "Allow",
            "Action": "logs:*",
            "Resource": "arn:aws:logs:*:<account-id>:log-group:ThreatXPrevent*:*"
        },
        {
            "Sid": "LaunchTemplates",
            "Action": "ec2:CreateLaunchTemplate",
            "Effect": "Allow",
            "Resource": "arn:aws:ec2:*:<account-id>:launch-template/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeParameters"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:*"
            ],
            "Resource": "arn:aws:ssm:*:<account-id>:parameter/ThreatXPrevent*"
        },
        {
            "Sid": "EC2",
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:AuthorizeSecurityGroupEgress"
            ],
            "Resource": "arn:aws:ec2:*:<account-id>:security-group/*",
            "Condition": {
                "StringLike": {
                    "aws:ResourceTag/aws:cloudformation:stack-name": "ThreatXPrevent*"
                }
            }
        },
        {
            "Sid": "EC2v3",
            "Effect": "Allow",
            "Action": [
                "ec2:TerminateInstances",
                "ec2:DeleteSecurityGroup",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:RunInstances",
                "ec2:DescribeInstances",
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeKeyPairs",
```

```
                "ec2:CreateSecurityGroup",
                "ec2:CreateTags",
                "ec2:DescribeSecurityGroups",
                "ec2:CreateLaunchTemplate",
                "ec2:DescribeLaunchTemplates",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:DeleteLaunchTemplate",
                "ec2:CreateLaunchTemplateVersion",
                "ec2:DeleteLaunchTemplateVersions",
                "ec2:ModifyLaunchTemplate",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeLoadBalancerAttributes",
                "elasticloadbalancing:DescribeListeners",
                "elasticloadbalancing:DescribeListenerCertificates",
                "elasticloadbalancing:DescribeSSLPolicies",
                "elasticloadbalancing:DescribeRules",
                "elasticloadbalancing:DescribeTargetGroups",
                "elasticloadbalancing:DescribeTargetGroupAttributes",
                "elasticloadbalancing:DescribeTargetHealth",
                "elasticloadbalancing:DescribeTags",
                "elasticloadbalancing:DescribeTrustStores"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ElasticLoadbalancing",
            "Effect": "Allow",
            "Action": [
                "elasticloadbalancing:CreateLoadBalancer",
                "elasticloadbalancing:DeleteLoadBalancer",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:RemoveTags",
                "elasticloadbalancing:DescribeTags",
                "elasticloadbalancing:AddTags",
                "elasticloadbalancing:DescribeRules",
                "elasticloadbalancing:DescribeListeners"
            ],
            "Resource": "*"
        },
        {
            "Sid": "TargetGroup",
            "Effect": "Allow",
            "Action": [
                "elasticloadbalancing:CreateTargetGroup",
                "elasticloadbalancing:DeleteTargetGroup",
                "elasticloadbalancing:ModifyTargetGroup",
                "elasticloadbalancing:ModifyTargetGroupAttributes"
            ],
            "Resource": "arn:aws:elasticloadbalancing:*:<account-
id>:targetgroup/ThreatXPrevent*"
        },
```

```
        {
            "Sid": "ElasticLoadbalancingV2",
            "Effect": "Allow",
            "Action": [
                "elasticloadbalancing:SetLoadBalancerPoliciesOfListener",
                "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
                "elasticloadbalancing:ModifyLoadBalancerAttributes",
                "elasticloadbalancing:ConfigureHealthCheck",
                "elasticloadbalancing:SetWebAcl",
                "elasticloadbalancing:ModifyListener",
                "elasticloadbalancing:AddListenerCertificates",
                "elasticloadbalancing:RemoveListenerCertificates",
                "elasticloadbalancing:ModifyRule",
                "elasticloadbalancing:CreateListener"
            ],
            "Resource": "arn:aws:elasticloadbalancing:*:<account-
id>:loadbalancer/app/ThreatXPrevent*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticloadbalancing:CreateRule",
                "elasticloadbalancing:DeleteRule",
                "elasticloadbalancing:DeleteListener"
            ],
            "Resource": [
                "arn:aws:elasticloadbalancing:*:<account-
id>:listener/app/ThreatXPrevent*",
                "arn:aws:elasticloadbalancing:*:<account-id>:listener-
rule/app/ThreatXPrevent*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "elasticloadbalancing:RegisterTargets",
                "elasticloadbalancing:DeregisterTargets"
            ],
            "Resource": "arn:aws:elasticloadbalancing:*:*:targetgroup/*/*"
        },
        {
            "Sid": "IAM",
            "Effect": "Allow",
            "Action": [
                "iam:CreateInstanceProfile",
                "iam:DeleteInstanceProfile",
                "iam:GetInstanceProfile",
                "iam:GetRole",
                "iam:RemoveRoleFromInstanceProfile",
                "iam:CreateRole",
                "iam:DeleteRole",
```

```
                "iam:AddRoleToInstanceProfile",
                "iam:PassRole",
                "iam:DeleteRolePolicy",
                "iam:GetRolePolicy",
                "iam:GetPolicy",
                "iam:CreatePolicy",
                "iam:DeletePolicy",
                "iam:ListPolicyVersions",
                "iam:TagRole",
                "iam:DetachRolePolicy",
                "iam:AttachRolePolicy"
            ],
            "Resource": [
                "arn:aws:iam::<account-id>:role/ThreatXPrevent*",
                "arn:aws:iam::<account-id>:policy/ThreatXPrevent*",
                "arn:aws:iam::<account-id>:instance-profile/ThreatXPrevent*"
            ]
        },
        {
            "Sid": "IAMv2",
            "Effect": "Allow",
            "Action": "iam:PutRolePolicy",
            "Resource": [
                "arn:aws:iam::<account-id>:role/ThreatXPrevent*",
                "arn:aws:iam::<account-id>:policy/ThreatXPrevent*"
            ]
        },
        {
            "Sid": "ACM",
            "Effect": "Allow",
            "Action": "acm:*",
            "Resource": "arn:aws:acm:*:<account-id>:certificate/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetResourcePolicy",
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret",
                "secretsmanager:ListSecretVersionIds",
                "secretsmanager:CreateSecret",
                "secretsmanager:PutSecretValue",
                "secretsmanager:TagResource",
                "secretsmanager:DeleteSecret"
            ],
            "Resource": "arn:aws:secretsmanager:*:<account-
id>:secret:/ThreatXPrevent*"
        },
        {
            "Effect": "Allow",
            "Action": "secretsmanager:ListSecrets",
```

```
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "route53:ChangeResourceRecordSets",
                "route53:GetHostedZone"
            ],
            "Resource": "arn:aws:route53:::hostedzone/<hosted-zone-id>"
        },
        {
            "Effect": "Allow",
            "Action": "route53:GetChange",
            "Resource": "arn:aws:route53:::change/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:PutScalingPolicy",
                "autoscaling:DescribePolicies",
                "autoscaling:DeletePolicy",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:DescribeScalingActivities"
            ],
            "Resource": "*"
        }
    ]
}
```

1. When you are complete, click **[Next]**
2. Give the policy a name (e.g., *threatx-prevent-install*)
3. Add a **Tag**:
   - **Key**: *product*
   - **Value**: *threatx-prevent*
4. Click **[Create Policy]**.

Creating A New Role For The Installation

From the **IAM Console**…

1. In the **main menu** to the left, select **Access Management › Roles**
2. Click the **[Create Role]** button.

From the **Create Role** page…

1. Verify that the AWS service button is selected.
2. From the list, select *CloudFormation* and click **[Next]**.

3. In the **Filter Policies** field, locate and select the checkbox of the policy you created. Click **[Next]**.
4. For **Role Name**, enter *threatx-prevent-install*.
5. Add a **Tag**:
    - **Key**: *product*
    - **Value**: *threatx-prevent*
6. Click **[Create Role]**

Use The New Role To Create The Stack

From the **Configure Stack Options Page** …

1. Locate the **Permissions** section
2. In the **IAM Role Name** field, select the newly created role: *threatx-prevent-install*