Cloud Custodian

Cloud Custodian is an open source utility that provides automated governance. This tool allows us to identify and remediate AWS services and configurations that do not meet our policies.

We are using Cloud Custodian Lambdas to enforce Direct Supply policies. When a service is deployed that is out of compliance, the lambda will immediately remediate the issue by restricting or deleting the service and notifying the resource owner using the `OwnerEmail` or `SNSTopicARN` tags.

Policy Examples and useful commands

Sample Policy - deletes unencrypted AMIs:

```
- name: delete-unencrypted-ec2
    comments: Identifies EC2 instances with unencrypted EBS volumes and terminates them.
    resource: ec2
    filters:
      - type: ebs
        key: Encrypted
        value: false
    actions:
      - terminate
      - type: notify
        template: redefault.html
        subject: "[cloud-custodian {{ account }}] Unencrypted EC2 Instances DELETED in {{ region }}"
        violation_desc: "The following EC2(s) are not encrypted"
        action_desc: have been terminated per <a href="https://docs.example.com/aws-policies">standar
        questions_email: you@example.com
        questions_slack: YourSlackChannel
        to:
          - cloud-custodian@directsupply.com
          - resource-owner
        transport:
          type: sqs
          queue: https://sqs.us-east-1.amazonaws.com/12345678910/custodian-sqs-queue
```

Note the notify action - this will trigger an email to be sent to both the resource owner and the cloud-custodian distribution list.

To test a policy locally without actually running the policy (assuming you have installed custodian locally)

```
vault-auth
export AWS_PROFILE=directsupply-sandbox

source custodian/bin/activate
custodian run --dryrun --cache-period 0 --output-dir ./cloud-custodian policy/policy_file_name.yml
```

or all of the policies

```
vault-auth
export AWS_PROFILE=directsupply-sandbox

source custodian/bin/activate
export AWS_DEFAULT_REGION=us-east-1; export ACCOUNT_ID=689557391783; export ACCOUNT_NAME=sandbox;
./dry_run.sh ${AWS_DEFAULT_REGION}
```

Template and syntax helpers are available from the CLI: use `$ custodian schema` to get a list of resources that can be governed by Cloud Custodian. use `$ custodian schema [resource]` to see the list of filters and actions that can be applied to that resource type. for details on an object's specific filter or action, you can use `$ custodian schema [resource].[filter|action].[FilterName|ActionName]`.

For example, to find more on detaching an EBS volume:

```
$ custodian schema ebs.actions.detach

Help
----

Detach an EBS volume from an Instance.

If 'Force' Param is True, then we'll do a forceful detach
of the Volume. The default value for 'Force' is False.

 :example:

 .. code-block:: yaml

         policies:
           - name: instance-ebs-volumes
             resource: ebs
             filters:
               VolumeId :  volumeid
             actions:
               - detach

Schema
------

{
    "additionalProperties": false,
    "required": [
        "type"
    ],
    "type": "object",
    "properties": {
        "force": {
            "type": "boolean"
        },
        "type": {
            "enum": [
                "detach"
            ]
        }
    }
}
```

Testing the Mailer

There exists a lambda that that can be utilized to test the c7n mailer component. To see the policy, take a look at `policy\mailer-test.yml`. It will send an email to whomever is tagged as `OwnerEmail`. It is fired if you create an S3 bucket with the `MailerTest` tag existing.

Example Direct Supply Policies:

Service	Requirement	Policy	Policy Type
ALB	Require access logs are enabled	Enable access logs on ALB	HIPAA compliance
ALB	Require ALB listeners on HTTP redirect to HTTPS	Delete listener on HTTP that doesn't redirect to HTTPS	HIPAA compliance
EC2	Require AMI encryption at rest	Deregisters original unencrypted AMI	HIPAA compliance
EC2	Require encryption on all EC2 instance EBS volumes	Delete EC2 instance if EBS volumes not encrypted	HIPAA compliance

Service	Requirement	Policy	Policy Type
EBS	Require encryption on all EBS volumes	Delete EBS volumes if not encrypted	HIPAA compliance
EBS	Require encryption on backups	Delete unencrypted snapshots	HIPAA compliance
S3	Require encryption on S3 buckets	Enable S3 encryption	HIPAA compliance
Security Group	Do not allow open, unrestricted security group rules	Delete open security group rules (ingress 0.0.0.0/0)	HIPAA compliance

If your deployment was terminated by Custodian

If you deploy infrastructure that is subsequently altered or terminated by Custodian, it is the responsibility of the deploying person or team to ensure that Terraform State is properly updated.

If your deployment was modified by Custodian

You will need to import the updated resource or resources into state. You can use terraform import to pull the updated infrastructure configuration into state. Hashicorp documentat is available here The command to run is: `terraform import [deployment.address] [resource.id]` Example: `terraform import module.foo.security_group sg-0123456789`

If your deployment was terminated or removed by Custodian

You will need to remove your deployment from Terraform State. Hashicorp Documentation is available here The command to run is: `terraform state rm [deployment.address]` Example: `terraform state rm module.foo.security_group`

Whitelisting a Service

You may find that you want to use a service, but it has been blacklisted through policy. Please follow the following procedure in order to lift the blacklisting.

Review HIPAA eligibility requirements for AWS services

HIPAA Compliance HIPAA Eligible Services Reference HIPAA Security and Compliance on AWS Whitepaper

Cleanup blacklist:

While there exists a tool known as mugc within cloud-custodian, I was unsuccessful in getting it to work.

You can instead manually cleanup the blacklist item related to your service:

- Delete the policy in question and commit
- Delete the Lambda named like "custodian-" on all tiers where you're retiring it
- Delete the CloudWatch event rule that represents the schedule named like "custodian-"

Create whitelist policy/policies:

You likely need to create a policy that enforces some rules that allow you to use the service in a compliant audit passing manner. Examples for a HIPAA related cloud-custodian policy allowing usage of EC2 within Knox would

- Enforce encryption at rest on attached EBS/EFS
- Require additional policies ensuring network traffic is encrypted, such as:
- ALB uses SSL/TLS
- IPsec
- Backend VPC traffic is encrypted

See the resources below for a link to our gitlab repository.

Have policy reviewed and released

Work with Team CNAME to get the policy implemented, reviewed, and released.

Resources

- Our GitLab repository
- Cloud Custodian homepage
- c7n Mailer
- Cloud Custodian github
- Read the Docs