

# CodeCommit Configuration

Step 1: Create an IAM User with AWSCodeCommitPowerUser policy.

PermissionsGroupsTagsSecurity credentialsAccess Advisor

Permissions policies (1)


Permissions are defined by policies attached to the user directly or through groups.

Q Search

Filter by Type

All types

< 1 > ⚙

<input type="checkbox"/>	Policy name <a href="#">↗</a>	Type	Attached via <a href="#">↗</a>
<input type="checkbox"/>	 <a href="#">AWSCodeCommitPowerUser</a>	AWS managed	Directly

Step 2: Create Repositories

Developer Tools > CodeCommit > Repositories

Repositories Info

🔄

🔔 Notify ▼

📄 Clone URL ▼













👁 View repository

🗑 Delete repository

Create repository

Q

< 1 > ⚙

	Name ▼	Description	Last modified ▼	Clone URL
<input type="radio"/>	<a href="#">vaccination-appointment-service</a>	-	11 minutes ago	 <a href="#">HTTPS</a>  <a href="#">SSH</a>  <a href="#">HTTPS (GRC)</a>
<input type="radio"/>	<a href="#">vaccination-report-service</a>	-	1 hour ago	 <a href="#">HTTPS</a>  <a href="#">SSH</a>  <a href="#">HTTPS (GRC)</a>
<input type="radio"/>	<a href="#">vaccination-auth-service</a>	-	1 hour ago	 <a href="#">HTTPS</a>  <a href="#">SSH</a>  <a href="#">HTTPS (GRC)</a>
<input type="radio"/>	<a href="#">vaccination-registration-service</a>	-	1 hour ago	 <a href="#">HTTPS</a>  <a href="#">SSH</a>  <a href="#">HTTPS (GRC)</a>

Step 3: Add your SSH keys to the newly created user in Step 1 security credentials. Up to 5 SSH can be added per IAM user.

IAM > Users > Vaccine-SCM-user

## Vaccine-SCM-user [Info](#)

Delete

### Summary

ARN Vaccine-SCM-user	Console access Disabled	
Created August 28, 2023, 11:29 (UTC+06:00)	Last console sign-in -	

Permissions | Groups | Tags (1) | **Security credentials** | Access Advisor

### SSH public keys for AWS CodeCommit (5)

User SSH public keys to authenticate access to AWS CodeCommit repositories. You can have a maximum of five SSH public keys (active or inactive) at a time. [Learn more](#)

Actions ▾ **Upload SSH public key**

SSH Key ID	Uploaded	Status
	17 days ago	✓ Active
	17 days ago	✓ Active
	17 days ago	✓ Active
	16 days ago	✓ Active
	17 hours ago	✓ Active

Step 4: Again under Security Credentials for HTTPS access to your repositories you need to generate git credentials for your account.

### HTTPS Git credentials for AWS CodeCommit (1)

Generate a user name and password you can use to authenticate HTTPS connections to AWS CodeCommit repositories. You can have a maximum of 2 sets of credentials (active or inactive) at a time. [Learn more](#)

Actions ▾ **Generate credentials**

User name	Created	Status
<input type="radio"/> Vaccine-SCM-user-at-	17 days ago	✓ Active

Step 5: Copy the username and password that IAM generated for you, either by showing, copying, and then pasting this information into a secure file on your local computer, or by choosing Download credentials to download this information as a .CSV file. You need this information to connect to CodeCommit.

Step 6: Check your connection by cloning one of the repositories.

---

## ECR(Elastic Container Registry) Setup

Step 1: Go over to ECR and create a private repository with a name of your choosing.

[Amazon ECR](#) > [Repositories](#) > Create repository

## Create repository

### General settings

**Visibility settings** | [Info](#)  
Choose the visibility setting for the repository.

☒ **Private**  
Access is managed by IAM and repository policy permissions.

☐ **Public**  
Publicly visible and accessible for image pulls.

**Repository name**  
Provide a concise name. A developer should be able to identify the repository contents by the name.

amazonaws.com/ vms-images

10 out of 256 characters maximum (2 minimum). The name must start with a letter and can only contain lowercase letters, numbers, hyphens, underscores, periods and forward slashes.

**Tag immutability** | [Info](#)  
Enable tag immutability to prevent image tags from being overwritten by subsequent image pushes using the same tag. Disable tag immutability to allow image tags to be overwritten.

☐ **Disabled**

[i](#) Once a repository has been created, the visibility setting of the repository can't be changed.

Step 2: Next, go to Permissions>Edit JSON Policy and delete the default and set the following permissions for the repository

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": "*",  
      "Action": [  
        "ecr:BatchGetImage",  
        "ecr:DescribeImages",  
        "ecr:GetDownloadUrlForLayer",  
        "ecr:PullImage"  
      ]  
    }  
  ]  
}
```

---

## S3 Bucket Configuration

---

Step 1: Go over to S3 and create a private bucket for the project. Check if the settings matches the following screenshots and keep the defaults for rest of the configurations.

# Create bucket [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

## General configuration

Bucket name

vms-bucket

Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

AWS Region

Asia Pacific (Singapore) ap-southeast-1 ▼

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Choose bucket

## Object Ownership [Info](#)

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

- ☒ **ACLs disabled (recommended)**  
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

- ☐ **ACLs enabled**  
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership  
Bucket owner enforced

## Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

### ☒ Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

#### ☒ Block public access to buckets and objects granted through *new* access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

#### ☒ Block public access to buckets and objects granted through *any* access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

#### ☒ Block public access to buckets and objects granted through *new* public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

#### ☒ Block public and cross-account access to buckets and objects through *any* public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

## AWS EKS Setup

Configure the following in the machine you are going to access the cluster:

- AWS CLI
- AWS IAM Authenticator
- Kubectl

Run the following command to get kubeconfig file for the new cluster:

Linux:

```
aws eks --region (terraformoutput - rawregion) update --kubeconfig --name  
(terraform output -raw cluster_name)
```

Windows:

```
set region_code=region-code
```

```
set cluster_name=my-cluster
```

```
set account_id=111122223333
```

```
for /f "tokens=*" %%a in ('aws eks describe-cluster --region %region_code% --name %cluster_name% --query "cluster.endpoint" --output text') do set cluster_endpoint=%%a
```

```
for /f "tokens=*" %%a in ('aws eks describe-cluster --region %region_code% --name %cluster_name% --query "cluster.certificateAuthority.data" --output text') do set certificate_data=%%a
```

```
aws eks update-kubeconfig --region %region_code% --name %cluster_name%
```

```
aws eks --region ap-southeast-1 update-kubeconfig --name vaccination-system-eks
```

# AWS Load balancer Controller Configuration

## Create Identity provider

Step 1: Copy OpenIDConnect URL from EKS overview

The screenshot shows the AWS EKS console 'Overview' tab for a cluster named 'vaccination-system-eks'. The 'Details' section contains the following information:

- API server endpoint:** <https://82CC959839C655D1BD377CAB12C0BCD7.gr7.ap-southeast-1.eks.amazonaws.com>
- Certificate authority:** `LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSOtLS0tCk1JSURCVENDQWUyZ0F3SUJBZ0lUTHXRGRGNWMxMUF3RFFZSkVWklodmNOQVFF`
- OpenID Connect provider URL:** <https://oidc.eks.ap-southeast-1.amazonaws.com/id/82CC959839C655D1BD377CAB12C0BCD7> (highlighted with a red box)
- Cluster IAM role ARN:** [arn:aws:iam::678554781153:role/vaccination-system-eks-cluster-20230913021253612000000002](#)
- Created:** 17 minutes ago
- Cluster ARN:** `arn:aws:eks:ap-southeast-1:678554781153:cluster/vaccination-system-eks`
- Platform version:** [Info](#) eks.5

Step 2: Go to IAM console>Identity Provider and create a OpenID Connect provider using the connector provider URL copied in the earlier step. Use sts.amazonaws.com as the audience.

### Summary

Provider oidc.eks.ap-southeast-1.amazonaws.com/id/82CC959839C655D1BD377CAB12C0BCD7	Provider Type OpenID Connect	Creation Time September 13, 2023, 08:22 (UTC+06:00)	ARN arn:aws:iam::678554781153:oidc-provider/oidc.eks.ap-southeast-1.amazonaws.com/id/82CC959839C655D1BD377CAB12C0BCD7
---	---------------------------------	--	--

### Audiences (1)

Also known as client ID, audience is a value that identifies the application that is registered with an OpenID Connect provider.

Actions ▼

< 1 >

Audience
<input type="radio"/> sts.amazonaws.com

### Thumbprints (4)

Server certificate thumbprint is the hex-encoded SHA-1 hash value of the X.509 certificate used by the domain where the OpenID Connect provider makes its keys available.

Manage

You can add up to 5 thumbprints. This lets you maintain multiple thumbprints if the identity provider is rotating certificates.

- 9e99a48a9960b14926bb7f3b02e22da2b0ab7280
- 06b25927c42a721631c1efd9431e648fa62e1e39
- 2ad974a775f73cbdbbd8f5ac3a49255fa8fb1f8c
- 30e620af3b44d4a30c0a64d1ecb8713fab02ca9f

Step 3: Now create an IAM policy from AWS load balancer controller documentation for the version you are using. I am using v2.6.1 in this project.

[https://github.com/kubernetes-sigs/aws-load-balancer-controller/blob/v2.6.1/docs/install/iam\\_policy.json](https://github.com/kubernetes-sigs/aws-load-balancer-controller/blob/v2.6.1/docs/install/iam_policy.json)