C3 AI Installation Guide - AWS

Version 8.3

29 September 2023

Legal Notice

C3.ai products and services are sold subject to the C3.ai terms and conditions agreed at the time of purchase. Except as expressly permitted in that agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means the C3.ai products, services, or documentation.

The information contained herein is subject to change without notice, and is not warranted to be error-free. The information is provided by C3.ai "as-is" for informational purposes only, without representation or warranty of any kind, and C3.ai or its affiliated companies will not be liable for errors or omissions with respect to the information. The only warranties for C3.ai products and services are those that are set forth in the express warranty statements, if any, accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. If you find any errors, please report them to us in writing.

If this software or documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then use the following notice: "U.S. GOVERNMENT END USERS: C3.ai programs, including any integrated software, any programs installed on any hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government."

C3.ai materials are not intended for use in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control systems, life support machines or other equipment in which the failure the C3.ai materials could lead to death, personal injury, or severe physical or environmental damage. C3.ai disclaims any and all liability arising out of, or related to, any such use of the C3.ai materials.

Information contained in this document regarding third party product or services does not constitute a license from C3.ai to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party. C3.ai is not responsible for and expressly disclaims all warranties of any kind with respect to third-party content, products, and services. C3.ai is not responsible for any loss, costs, or damages incurred due to the access to or use of third-party content, products, or services, except as set forth in a written agreement between you and C3.ai.

Any software coding samples included in this documentation are examples only and are not intended to be used in a production environment. The code is provided "as-is" and use of any code is at your own risk. C3.ai does not warrant the correctness or completeness of the code given herein, and C3.ai is not liable for errors or damages caused by usage of the code.

The business names used in this documentation are fictitious and are not intended to identify any real companies currently or previously in existence.

C3 AI, C3.ai, and the C3.ai logos are trademarks or registered trademarks of C3.ai, Inc. in the United States and/or other countries. All other product names, trademarks, and registered trademarks are the property of their respective owners.

Table of Contents

Overview

C3 AI Applications support flexible deployment options including C3 AI-managed cloud deployments or customer-managed deployments. The choice of deployment option will have implications on project timelines, service-level agreement (SLA), and RACI requirements.

C3 AI-managed cloud deployment

For C3 AI-managed cloud deployment, C3 AI provides a dedicated subaccount with dedicated compute, storage, and networking resources. All cloud resources are dedicated to your account and will not be shared with any other customer. C3 AI employs industry leading cyber security and access control practices to protect your applications and data.

C3 AI-managed cloud deployments are typically completed within two (2) days of the scheduled deployment start. All the services are hosted in C3 AI's Amazon Web Services (AWS) cloud account.

C3 AI customer-managed deployment

You can optionally deploy the C3 AI cluster in your own AWS Virtual Private Cloud (VPC), a feature known as customer-managed deployment. You can use a customer-managed deployment to exercise additional control over your network configurations to comply with specific cloud security and governance standards your organization may require.

An AWS VPC allows you to provision a logically isolated section of the AWS cloud where you can launch AWS resources in a virtual private secure network. The VPN is the network location for your C3 AI clusters.

A deployment start schedule for a customer-managed deployment is dependent on the customer.

C3 AI customer-managed deployment overview

In C3 AI, a cluster is a C3 AI deployment in the cloud that functions as the environment for developing and deploying C3 AI Applications. Your organization can choose to have multiple clusters or just one, depending on your needs.

A customer-managed deployment is a good solution if you have:

- Security policies that prevent Platform-as-a-Service (PaaS) or Software-as-a-Service (SaaS) providers from creating VPCs in your own AWS account.
- An approval process to create a new VPC, in which the VPC is configured and secured and well-documented by internal information security or cloud engineering teams.

- A team with Terraform expertise and a change management system that are available for on-going management of infrastructure for the C3 AI Cluster.

Benefits include:

- Lower privilege level: You maintain more control of your own AWS account. And you do not need to grant C3 AI as many permissions as you do for a C3 AI-managed deployment. For example, there is no need for permission to create VPCs.
- Maintain more control of your own AWS account and limit outgoing connections.

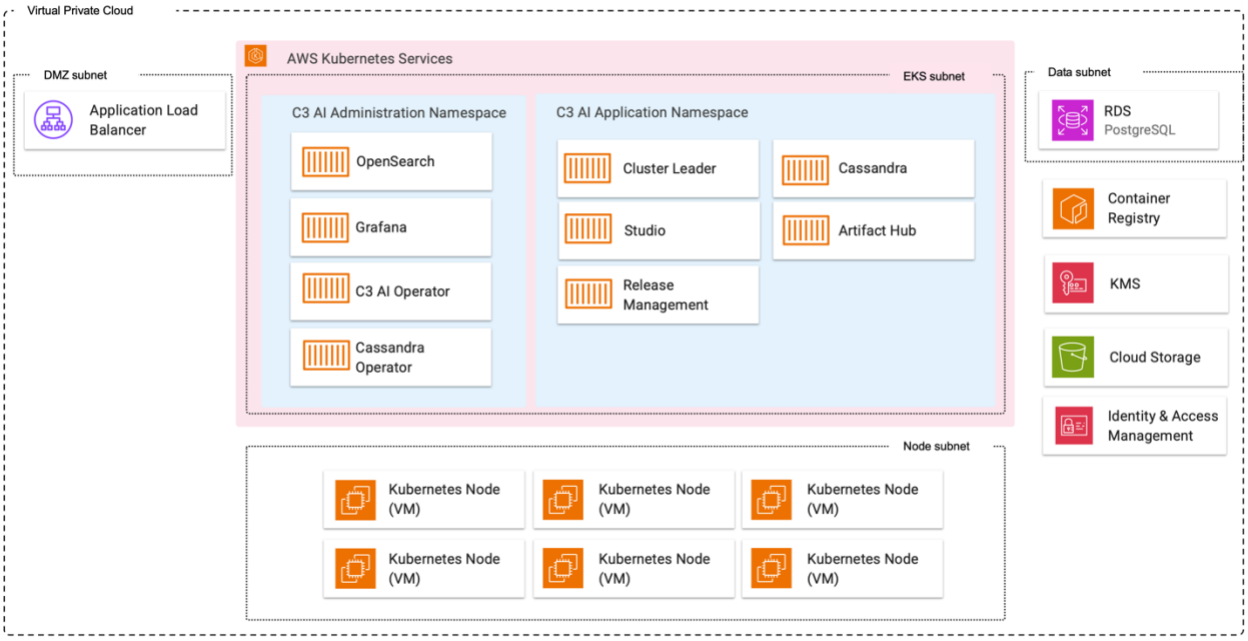

Figure 1. AWS Architecture with C3 AI Platform Deployment

Requirements

The C3 AI Platform requires specific AWS cloud services and infrastructure for successful deployment, as well as specific access requirements for C3 AI Operations to install, administer, and upgrade the C3 AI Platform and C3 AI Applications.

The following sections describe the specific services and access needs, including network configurations and subnet requirements, security group egress and ingress rules, and subnet-level access control lists (ACLs).

Required Amazon cloud services

The table below describes the Amazon cloud infrastructure services required by the C3 AI Platform. You are required to provide the services below configured to C3 AI specifications as documented in the HashiCorp Terraform scripts.

Amazon Cloud Service	Version	Description
Elastic Kubernetes Engine (EKS)	1.25	Operating environment responsible for the deployment, scaling, and management of the C3 AI Platform and C3 AI Applications.
Key Vault	Current version	Scalable, centralized, fast cloud key management.
RDS (PostgreSQL)	11.17	Relational database service (RDS) required for internal operations of the C3 AI Platform.
S3	Current version	Reliable and secure object storage used for the management of application and platform configuration and other ancillary tasks.
Identity and Access Management (IAM)	Current version	Fine-grained access control and visibility for centrally managing cloud service account resources.
Virtual Private Cloud (VPC)	Current version	Dedicated, isolated network for inter-C3Cluster communication.
EC2	Current version	Compute instances required by Amazon EKS.

Amazon cloud access requirements

The table below describes the access requirements for C3 AI Operations to install, administer, and upgrade the C3 AI Applications and C3 AI Platform.

Access Requirements	Description
A dedicated Amazon sub account	When creating the project, C3 AI requires: (1) Account identifier, (2) Cloud region.
IAM user account for each C3 AI Operations member	C3 AI Operations personnel must be granted IAMReadOnlyAccess and IAMUserChangePassword managed policies.
Secure internet access to the Amazon sub account	Secure, remote access via internet (VPN access is acceptable) to a bastion host from which C3 AI Operations personnel can administer cloud infrastructure and C3 AI services.
A bastion host accessible by C3 AI Operations to manage the cluster	The bastion host will be used by C3 AI Operations to administer the C3 AI Applications and C3 AI Platform. Software utilities required to be installed on the bastion host must include: RedHat 8, AWS Command Line Interface (CLI) v2 v2.49.0, kubectl v1.2.5, Helm v3.8.
Access to C3 AI third-party library and image repositories	Access to C3 AI and third-party repositories for the container images, Python libraries, NodeJS libraries, and runtime billing data collection. If connecting to remote C3 AI, Python, and NodeJS artifact repositories violates security standards, the C3 AI Platform can be configured to connect to local artifact repositories (such as, AWS Container registry, JFrog, and Anaconda Enterprise).
X.509 certificate for terminating network encryption	A fully qualified domain name for C3 Cluster ingress configuration (for example, c3project.customer.com). You are responsible for providing the private and public key (and the certificate chain if necessary) from the x509 certificate to C3 AI. These are placed in a Kubernetes secret and used by C3 AI cluster ingress controller.

Network configuration

To deploy the C3 AI Platform in your own VPC, you must create the VPC following the requirements enumerated in VPC requirements section below.

VPC requirements

Your VPC must meet the requirements described in this section to host a C3 AI cluster.

VPC region

The Amazon region where the deployment will occur. Refer to AWS documentation for a list of available regions.

VPC sizing

The C3 AI Platform requires two (2) CIDR blocks.

IP Address Range	Description
10.0.0.0/22	Routable space, used by RDS (Postgres), EKS Cluster, and node pools.
172.0.0.0/16	Non-routable space, used by EKS pods.

The VPC must have DNS hostnames and DNS resolution enabled.

Subnets

C3 AI must have access to at least two subnets for each cluster with each subnet in a different availability zone. The subnets are:

- A subnet for the EKS cluster and node pools
- A subnet for Kubernetes pods

Each subnet must have a netmask between /16 and /22.

Subnet route table

The route table for workspace subnets must have quad-zero (0.0.0.0/0) traffic that targets the appropriate network device.

Additional subnet requirements

- Subnets must be private.
- Subnets must have outbound access to the public network using a NAT gateway and internet gateway, or other similar customer-managed appliance infrastructure.
- The NAT gateway must be set up in its own subnet that routes quad-zero (0.0.0.0/0) traffic to an internet gateway or other customer-managed appliance infrastructure.

Security groups

C3 AI must have access to at least one AWS security group and no more than five security groups. You can reuse existing security groups rather than create new ones.

Security groups must have the following rules.

Egress (outbound)

- Allow all TCP and UDP access to the workspace security group (for internal traffic)
- Allow TCP access to 0.0.0.0/0 for these ports:
    - 443: for C3 AI infrastructure, cloud data sources, and library repositories

Ingress (inbound)

- 443: for C3 AI application access
- 22: for SSH access to a bastion host

Subnet-level network ACLs

Subnet-level network ACLs must not deny ingress or egress to any traffic.

- ALLOW ALL from Source 0.0.0.0/0. This rule must be prioritized.
- Egress:
    - Allow all traffic to the C3 AI cluster VPC CIDR, for internal traffic.
    - Allow TCP access to 0.0.0.0/0 for these ports:
        - 443: for C3 AI infrastructure, cloud data sources, and library repositories.

HashiCorp Terraform Configuration

HashiCorp Terraform is a popular open-source tool for creating safe and predictable cloud infrastructure across several cloud providers. Terraform scripts are used to create the cloud infrastructure required by the C3 AI Platform and automate the deployment of the C3 AI Platform in your AWS account.

NOTE: For C3 AI customer-managed deployments, any customization performed on the Terraform scripts must be reapplied with each version of the Terraform scripts from C3 AI.

Getting started

In this section, you install and configure requirements to use Terraform. You then configure Terraform authentication. Following completion of this section, you go to the "Installation Steps" section below to deploy and configure the cloud infrastructure required by the C3 AI Platform.

Requirements

To use Terraform to create cloud infrastructure resources required by the C3 AI Platform in your AWS account, you must have the following:

- An AWS account.
- On your local development machine, you must have:
    - The HashiCorp Terraform CLI. See Install Terraform on the Terraform website to download the binary of the required Terraform version specified in the `main.tf` file example in the "Installation Steps" section below. Select AMD64 or ARM64 depending on the which matches the client hardware from which you will run the Terraform scripts.
    - The AWS CLI
    - The eksctl command-line tool. See Install eksctl on the Amazon EKS website.
- The following environment variables:
    - AWS_ACCESS_KEY_ID, set to the value of your AWS user's access key ID. See Programmatic access in the AWS General Reference.
    - AWS_SECRET_ACCESS_KEY, set to the value of your AWS user's secret access key. See Programmatic access in the AWS General Reference.
    - AWS_REGION, set to the value of the AWS Region code for your AWS account. See Regional endpoints in the AWS General Reference.

Installation Steps

Installation of the C3 AI Platform on AWS is a multi-step process due to limitations of Terraform and AWS-specific configuration requirements. The installation process is the following:

1. Create the VPC and required AWS services.
2. Grant C3 AI Operations access to EKS as a Kubernetes administrator.
3. C3 AI Operations installs and configures required EKS options.
4. Update your EKS node pool configuration.
5. Validate the VPC and configuration of required AWS services.
6. C3 AI Operations completes the installation of the C3 AI Platform.

To create a VPC, C3 AI requires the use of HashiCorp Terraform and will provide a set of Terraform scripts to assist you in the creation of the VPC and required AWS Services. If you are unfamiliar with Terraform, review their Get Started – AWS documentation.

NOTE: See the "HashiCorp Terraform Requirements" section above to ensure all requirements are met prior to completing the installation steps below.

A description of the Terraform modules is below.

Terraform Module	Description
bootstrap	Configures the necessary IAM roles and policies to allow a Terraform orchestrator to deploy all services required by the C3 AI Platform on AWS.
c3cluster	Coordinates the execution of all other Terraform modules.
eks-cluster	Configures AWS Elastic Kubernetes Service (EKS), including VPC configuration, endpoint access, authorized IP addresses, and the version of Kubernetes used by the cluster.
eks-nodepool	Configures the AWS EKS node groups, including default instance size, required subnet, and permissions assigned to each node.
firewall	Configures ingress and egress security rules.
iam	Configures the required IAM roles and policies.
kms	Configures the AWS Key Management service.

Terraform Module	Description
network	Configures the VPC, including public and private subnets, internet gateway, CIDR blocks, DHCP, and NAT.
postgres	Creates an AWS RDS database and assigns the database to the database subnet.
vault	Seeds the application role and secret in the AWS Key vault.
S3	This module configures the AWS S3 buckets to be used with the C3 AI cluster.

In addition to the required tools listed in the “HashiCorp Terraform Requirements” section, install TFSwitch, which is a tool used to switch easily between Terraform versions.

See Install TFSwitch and TFSwitch Quick Start on the TFSwitch website for more information.

1. Create the VPC and required AWS services

This guide shows you how to create the cloud infrastructure services required by the C3 AI Platform using HashiCorp Terraform on AWS.

1.1 Run the bootstrap module

This module creates the necessary IAM roles and policies to configure the VPC and required AWS services. Configure a new `main.tf` file below, replacing the CAPITALIZED variable names with your values.

NOTE: The cluster name must adhere to the following restrictions: name must include not include a hyphen and must start with a letter; only lowercase letters are allowed with no other special characters or diacritics (accented letters); and, should be less than 18 characters in length.

```
module "bootstrap" {

  ##Link will be provided by C3 AI
  source = "https://<c3_url>/aws-bootstrap-x.y.z.zip"
  cluster_name = "CLUSTER_NAME"
  region = "AWS_REGION"
  account_id = "AWS_ACCOUNT_ID"

  ##A list of all ARNs that will need to deploy c3cluster Terraform scripts
  trusted_identifier_arns = ["ARN_TO_ASSUME_ROLE"]
  partition = "AWS_PARTITION"

}
```

```
terraform {
  required_version = "= 1.4.6"
}

provider "aws" {
  region = "AWS_REGION"
}
```

1.2 After configuring the main.tf file, run the following Terraform commands from the same directory

```
tfswitch
terraform init
terraform plan --out out.plan
terraform apply "out.plan"
```

NOTE: If you receive a “Command not found: Terraform” after running the commands above, the `terraform` binary might not be in your path. See the Get Started in AWS – Install Terraform page on the HashiCorp Terraform website for more information.

1.3 Run the c3cluster module

This module coordinates execution of all other Terraform modules. Configure a new `main.tf` in a separate directory from the bootstrap module, replacing the CAPITALIZED variable names with your values. Note that you must assume the `IAM_ROLE_NAME` role created by the bootstrap module.

Contact your account manager for the list of IP addresses required by C3 AI. These values will be used to update the `ip_allowlist` section below.

NOTE: C3 AI requires a set of CIDR blocks to be whitelisted for C3 AI Operations to deploy the C3 AI Platform.

```
provider "aws" {

  region = "AWS_REGION"

 assume_role {
  ##iam_role_name from bootstrap module
  role_arn = "ARN_FOR_IAM_ROLE_NAME"
 }
}

module "c3cluster" {
  ##Link will be provided by C3 AI
```

```
  source       = "https://<c3_url>/aws-c3cluster-x.y.z.zip"
  account_id   = "AWS_ACCOUNT_ID"
  c3_region    = "AWS_REGION"
  cluster_name = "CLUSTER_NAME"
  ip_allowlist = [
    {
      cidr_blocks = [
          "DESIRED_CIDR_BLOCKS"
      ],
      display_name = "Whitelisted IPs"
    },
  ]

  eks_node_pools = {
    "c3" : {
      machine_type = "r6i.4xlarge"
      node_count   = 0
      min_count    = 0
      max_count    = 9
    }
  }
}

terraform {
  required_version = "= 1.4.6"
}
```

1.4 After creating the main.tf file, run the example below from the same directory as the main.tf file

```
tfswitch
terraform init
terraform plan --out out.plan
terraform apply "out.plan"
```

2. Grant C3 AI Operations access to EKS as a Kubernetes administrator

To grant C3 AI Operations access to the EKS with administration permissions, do the following.

2.1 Create role C3_CLUSTER_ID-kubeconfig-01 with trust relationship for the individual user or group Amazon Resource Names (ARNs)

See Create an IAM role using a custom trust policy on the AWS website for more information.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::ACCOUNTID:RESOURCE_TYPE"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

NOTE: Replace:

- ACCOUNTID with ID of the AWS account that owns the resource, without the hyphens. For example, `123456789012`.
- RESOURCE_TYPE with user, group, or the resource type to be assigned to the user. For example, `user/firstname.lastname`. See also Amazon Resource Names on the AWS website for more information.

2.2 Create and attach policy to C3 AI Operations IAM users giving permissions to assume the kubeconfig IAM role and generate kubeconfig

Create the policy by doing the following:

1. In the AWS Identity and Access Management (IAM) dashboard, go to Access Management and Policies.
2. In the Permissions tab, enter the following JSON.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Resource": [
                "arn:aws:iam::ACCOUNTID:role/CLUSTER-kubeconfig-01"
            ]
        },
        {
            "Sid": "Statement2",
            "Effect": "Allow",
            "Action": [
                "eks:DescribeCluster"
            ],
            "Resource": [
                "arn:aws:eks:REGION:ACCOUNTID:cluster/CLUSTER-kube-01"
            ]
        },
        {
            "Sid": "ManageOwnAccessKeys",
            "Effect": "Allow",
            "Action": [
                "iam:CreateAccessKey",
                "iam:DeleteAccessKey",
                "iam:GetAccessKeyLastUsed",
                "iam:GetUser",
                "iam:ListAccessKeys",
                "iam:UpdateAccessKey",
                "iam:TagUser"
            ],
            "Resource": "arn:aws:iam::ACCOUNTID:RESOURCE_TYPE"
        }
    ]
}
```

NOTE: Replace:

- CLUSTER with the name of the EKS cluster. The cluster name must adhere to the following restrictions: name must include not include a hyphen and must start with a letter; only lowercase letters are allowed with no other special characters or diacritics (accented letters); and, should be less than 18 characters in length.
- REGION with AWS region code associated with the EKS cluster.
- ACCOUNTID with ID of the AWS account that owns the resource, without the hyphens. For example, `123456789012`.

- RESOURCE_TYPE with user, group, or the resource type to be assigned to the user. Use the value set in Step 3.1.

Attach the JSON policy to the C3 AI Operations IAM user by doing the following:

NOTE: Contact the COE for the specific C3 AI Operations IAM users that need to be added.

1. In the Entities Attached tab, select IAM Users from the Entity Type drop-down menu, and enter the applicable Entity Names.
2. Then, click Attach Policy.

2.3 Grant the kubeconfig role administration permissions on the Kubernetes cluster

EKS clusters use IAM users and roles to control access to the cluster. The rules are implemented in a config map called aws-auth. eksctl provides commands to read and edit this config map.

To create an identity mapping for C3 AI Operations personnel, do the following.

2.3.1. Assume the IAM_ROLE_NAME role created by the bootstrap module.

This will be the role ending with `c3icrole-01` unless modified by you.

```
aws sts assume-role --role-arn arn:aws:iam::ACCOUNTID:role/IAM_ROLE_NAME --
role-session-name c3aiops
```

Note: Replace:

- IAM_ROLE_NAME with the IAM_ROLE_NAME created by the bootstrap module.

The output of the above command will be as follows.

```
Output: ASSUMEDROLEUSER <ARN> <Role Session Name>

CREDENTIALS <AWS_ACCESS_KEY_ID> <Timestamp> <AWS_SECRET_ACCESS_KEY>
<AWS_SESSION_TOKEN>
```

2.3.2 Ensure the following environment variables are set using the output of the `aws sts assume-role` above.

- AWS_ACCESS_KEY_ID, set to the value returned by the `aws sts assume-role` command in Step 3.3.1 above. See Programmatic access in the AWS General Reference.
- AWS_SECRET_ACCESS_KEY, set to the value returned by the `aws sts assume-role` command in Step 3.3.1 above. See Programmatic access in the AWS General Reference.
- AWS_REGION, set to the value of the AWS Region code for your AWS account. See Regional endpoints in the AWS General Reference.
- AWS_SESSION_TOKEN, set to the AWS session token value returned by the `aws sts assume-role` command in Step 3.3.1 above. You receive this value as part of the temporary credentials returned by successful requests to assume a role.

2.3.3. Create the identity mapping for C3 AI Operations personnel.

This command must be run on one of the whitelisted IPs in the `c3cluster` module.

```
eksctl create iamidentitymapping \
  --cluster CLUSTER-kube-01 --region=REGION_CODE \
  --arn arn:aws:iam::ACCOUNTID:role/CLUSTER-kubeconfig-01  --username
opsrole --group system:masters \
  --no-duplicate-arns
```

NOTE: Replace:

- REGION_CODE with your AWS Region's code
- ACCOUNTID with your AWS account identifier
- CLUSTER with the name of the cluster. The cluster name must adhere to the following restrictions: name must include not include a hyphen and must start with a letter; only lowercase letters are allowed with no other special characters or diacritics (accented letters); and, should be less than 18 characters in length.

Once creation of the EKS identity mapping is complete, notify C3 AI Operations and they will complete the configuration of the AWS CNI driver for Kubernetes, EBS CNI driver, Kubernetes Autoscaler, and the Calico network policy engine for EKS.

3. C3 AI Operations installs and configures required EKS options

C3 AI Operations will configure the AWS CNI Driver for Kubernetes, EBS CNI Driver, Kubernetes Autoscaler, and the Calico network policy engine for EKS. A description of each item is below.

- **AWS CNI plugin for Kubernetes** - Deployed on each Amazon EC2 node in your Amazon EKS cluster. The add-on creates elastic network interfaces and attaches them to your Amazon EC2 nodes. The add-on also assigns a private IPv4 or IPv6 address from your VPC to each pod and service.
- **EBS CSI driver** - Allows Amazon Elastic Kubernetes Service (Amazon EKS) clusters to manage the lifecycle of Amazon EBS volumes for persistent volumes.
- **Kubernetes Autoscaler** - Uses Amazon EC2 Auto Scaling Groups to manage node groups. Cluster Autoscaler typically runs as a Deployment in your cluster.
- **Calico network policy engine** - Implements network segmentation and tenant isolation, useful when you want to create separate environments for development, staging, and production.

4. Update your EKS node pool configuration

C3 AI Operations will request that you set the minimum and maximum EKS node pool size based on expected use. As this is an infrastructure change, C3 AI recommends the change be made using HashiCorp Terraform. Below is the HashiCorp Terraform script used by C3 AI Operations to update the EKS node pool with a default machine type of `r6i.4xlarge` and a minimum of 1 and maximum of 9 nodes per availability zone.

```
provider "aws" {
  region = "AWS_REGION"

 assume_role {
  ##iam_role_name from bootstrap module
  role_arn = "ARN_FOR_IAM_ROLE_NAME"
 }
}

module "c3cluster" {
  ##Link will be provided by C3 AI
  source       = "LINK_TO_C3CLUSTER_MODULE"
  account_id   = "AWS_ACCOUNT_ID"
  c3_region    = "AWS_REGION"
  cluster_name = "CLUSTER_NAME"
  ip_allowlist = [
```

```
    {
      cidr_blocks = [
         "DESIRED_CIDR_BLOCKS",
      ],
      display_name = "Whitelisted IPs"
    },
  ]

  eks_node_pools = {
    "c3" : {
      machine_type = "r6i.4xlarge"
      node_count   = 1
      min_count    = 1
      max_count    = 9
    }
  }
}

terraform {
required_version = "= 1.4.6"
}
```

Execute the above Terraform script using the following commands.

```
terraform init
terraform plan --out out.plan
terraform apply out.plan
```

After updating your EKS node pool configuration run the C3 AI infrastructure validation utility to ensure that no additional infrastructure changes have been applied. All checks must pass for C3 AI Operations to be able to deploy the C3 AI Platform on your Kubernetes cluster.

5. Validate the VPC and configuration of required AWS services

After the VPC and required cloud services are configured, you are required to execute the C3 AI Cluster Validation Utility and provide the results to C3 AI. If all checks performed by the C3 AI Cluster Validation Utility pass, the VPC is suitable for C3 AI Operations to deploy the C3 AI Platform on the Kubernetes cluster.

NOTE: If the cluster validation utility fails, you must remediate all exceptions. All checks must pass for C3 AI Operations to be able to deploy the C3 AI Platform on your Kubernetes cluster. See the next section for details.

Once the checks are successfully completed, provide C3 AI Operations access to the cluster. Refer to the subsequent section for more information.

5.1 Run the C3 AI Cluster Validation Utility and provide results to C3 AI Operations

To run the C3 AI Cluster Validation Utility, do the following:

1. Download and unzip the c3prereqs module from `https://<c3_url>/c3prereqs-x.y.z.zip`
2. Edit the `config.env` to only include the Kubernetes cluster, removing the cloud deployment types that are not applicable.

```
#KUBECONFIG=""

# Minimum helm version required
MINIMUM_HELM_VERSION="3.12"

# Minimum kubernetes version required. Used for client and server
MINIMUM_KUBERNETES_VERSION="1.25"

# Timeout for the c3prereqs job
JOB_TIMEOUT="300s"

## Select Cloud to gcp or aws or azure ; uncomment appropriate cloud-
specific sections below
C3_CLOUD=""

C3_CLUSTER_NAME=""

## AWS SPECIFIC
# C3_AWS_ACCOUNT=""
# C3_AWS_SERVICE_ACCOUNT="c3"

## Set to true if cloud is AWS
S3_BUCKET_NAME=""
```

```
## GCP SPECIFIC
# C3_GCP_PROJECT_ID=
# C3_GCP_SERVICE_ACCOUNT="c3"


## AZURE SPECIFIC
# C3_AZURE_CLIENT_ID=


# Jfrog auth info
C3_REGISTRY_URL="registry.c3.ai"
C3_REGISTRY_USER=""
C3_REGISTRY_PASS=""

# Postgres validation
POSTGRES_USERNAME=""
POSTGRES_PASSWORD=""
POSTGRES_HOST=""
POSTGRES_PORT=
POSTGRES_DATABASE=""
```

3. Run the c3prereqs script. The script will record the results in the reports folder.
4. Review the report.

If the report indicates the VPC is ready for C3 AI Operations to deploy the C3 AI Platform on the Kubernetes cluster, provide the report to C3 AI Operations.

If the report fails, remediate all exceptions and rerun the C3 AI Cluster Validation Utility.

5.2 Provide C3 AI Operations access to the cluster

In addition to the output of the validation utility, you must provide C3 AI Operations with the following.

Title	Description
C3 AI Operations credentials	Credentials for C3 AI Operations team members.
EKS cluster name	By default, CLUSTER-kube-01. Confirm this in the AWS console by going to “Elastic Kubernetes Service” and identifying the newly created cluster. **NOTE**: The cluster name must adhere to the following restrictions: name must include not include a hyphen and must start with a letter; only lowercase letters are allowed with no other special characters or diacritics (accented letters); and, should be less than 18 characters in length.
Region	The AWS region associated with the EKS cluster

Title	Description
Role-Arn to get Kubeconfig	The ARN of the role created in Step 3; likely, arn:aws:iam::ACCOUNTID:role/C3_CLUSTER_ID-kubeconfig-01. Copy it from IAM → Roles → find the role you created in Step 3.
AWS Postgres endpoint	From AWS console, go to RDS Services, locate <C3_CLUSTER_ID>-pg-shared and provide the Endpoint value for the instance.
AWS Postgres Admin password (after changing it)	From RDS services, locate <C3_CLUSTER_ID>-pg-shared and hit the Modify button at the top right. Configure a new password by pressing "Auto generate a password" or by filling in new password and take note of it. Or, if you do not have permissions to do so, run aws rds modify-db-instance --db-instance-identifier --master-user-password <pwd> as the role C3_CLUSTER_ID-c3icrole-01
EKS Pod Subnets with Availability Zones	Go to VPC → Subnets. Search for subnets with name having a prefix of C3_CLUSTER_ID-sn-ekspod, taking note of the Subnet ID and Availability Zone of each one.
EKS security group ID	Go to VPC → Security Groups. Search for security groups with a prefix of C3_CLUSTER_ID-sg-eks - this should show one security group. Take note of its Security group ID.
S3 Bucket name	By default, ACCOUNTID--CLUSTERNAME. Confirm this in the AWS console by going to S3 and identifying this bucket.
Domain name	A fully qualified domain name for C3 AI Cluster ingress configuration (for example, c3project.customer.com)
Public and private key	The private and public key (and the certificate chain if necessary) from the x509 certificate. This will be required for ingress configuration.

6. C3 AI Operations completes the installation of the C3 AI Platform

With the infrastructure properly configured and EKS node pool configuration updated, C3 AI Operations will continue with the installation of the C3 AI Platform.

At the conclusion of the VPC creation and the deployment of the C3 AI Platform, the AWS Cloud environment will resemble the following.

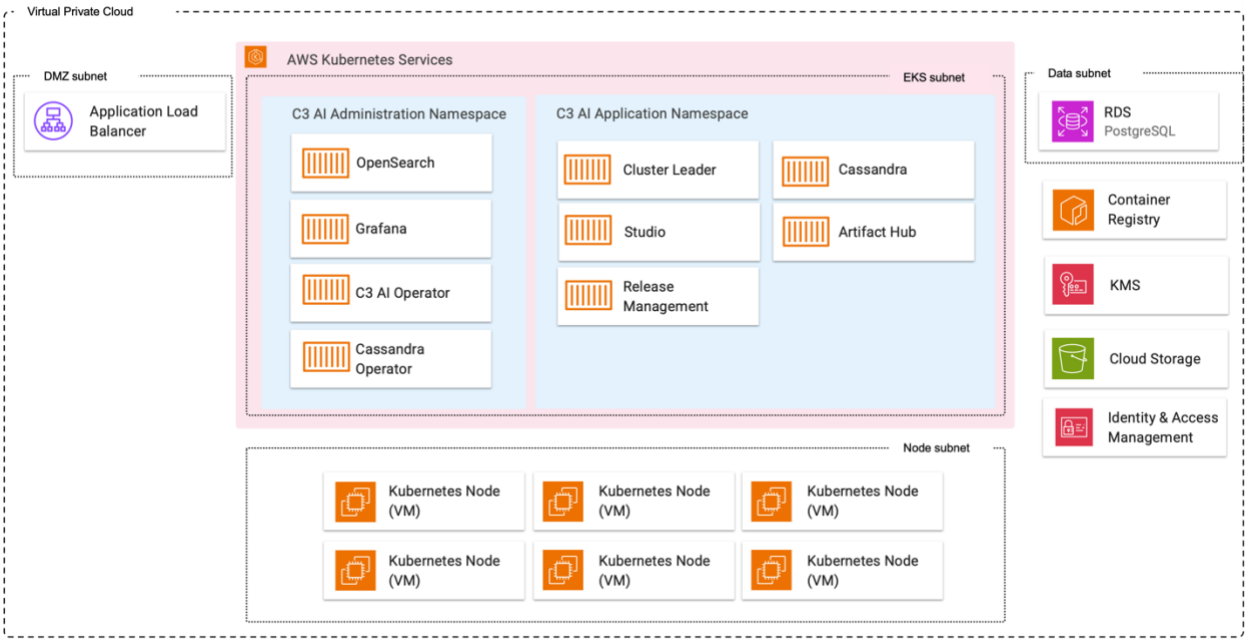

Figure 2. AWS Architecture with C3 AI Platform Deployment