C3 AI Infrastructure Scripts / HashiCorp Terraform Deployment – Amazon Web Services GovCloud

Version 8.11

23 July 2026

AWS GOVCLOUD (US) EDITION

Legal Notices

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. Reverse engineering, disassembly, or decompilation of this C3 AI software

The information contained herein is subject to change without notice. THE INFORMATION AND DOCUMENTATION ARE PROVIDED "AS IS" AND "AS AVAILABLE," WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, ACCURACY, COMPLETENESS, AND NON-INFRINGEMENT. The information is provided by C3.ai 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 or any commitment by C3.ai to deliver any product, code, functionality, or service. 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 the following notice is applicable: "U.S. GOVERNMENT END USERS: C3.ai programs (including any integrated software, any programs embedded, installed, or activated on hardware, and modifications of such programs) and C3.ai computer documentation or other C3.ai data delivered to or accessed by U.S. Government end users are "commercial computer software," or "commercial computer software documentation," pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations, including FAR 12.212, FAR 27.405-3, and, for Department of Defense acquisitions, DFARS 227.7202-1 through 227.7202-4. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of (i) C3.ai programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), (ii) C3.ai computer documentation, and/or (iii) other C3.ai data, is subject to the rights and limitations specified in the license or subscription contained in the applicable contract. The terms governing the U.S. Government's use of C3.ai cloud services are defined by the applicable contract for such services. 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, other equipment or any other inherently dangerous applications in which the failure or malfunction of the C3.ai materials could lead to death, personal injury, or severe physical or environmental damage. If you use the C3.ai materials in any such application, you are responsible for taking all appropriate fail-safe, backup, redundancy, and other measures to ensure their safe use. 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 products 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. The use of any third-party trademark in this document is for identification purposes only and does not imply endorsement by, or affiliation with, the trademark owner.
All rights not expressly granted in the applicable license or subscription agreement, or in this notice, are reserved by C3.ai, Inc. and its affiliates.

Table of Contents

C3 AI Infrastructure Scripts / HashiCorp Terraform Deployment

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 Agentic AI Platform and automate the deployment of the C3 Agentic AI Platform in your AWS account.

Initial Configuration

Ahead of the C3 AI infrastructure deployment, you will need the tools mentioned in Tools and Libraries installed on the host that will perform the install. In addition, you will also need the following Terraform providers:

- registry.terraform.io/hashicorp/aws: 6.28.0
- registry.terraform.io/hashicorp/local: 2.5.2
- registry.terraform.io/hashicorp/null: ~> 3.2
- registry.terraform.io/hashicorp/tls: 4.0.5
- registry.terraform.io/hashicorp/cloudinit: 2.3.3

Note that the exact Terraform module versions will vary from version to version of the C3 AI Platform – be sure to use the exact versions referenced in the C3 AI Terraform modules.

For air-gapped systems and installs, you will need to locally download these modules for the appropriate bastion host architecture and make them available to the bastion host, either via local mirrors or hosted mirrors like JFrog Artifactory.

For portability, we recommend locally downloading and referencing the Terraform modules. You can do so with the following commands:

```
mkdir -p ./modules/3.0.17/ ./modules/bootstrap/ ./modules/c3cluster/
curl -u "myuser:mypassword" -o "./modules/3.0.17.zip"
      https://jfrog.c3.ai/artifactory/tf-registry/c3/aws/c3/3.0.17.zip
unzip ./modules/3.0.17.zip -d ./modules/3.0.17
```

Note that:

- 3.0.17 is the C3 AI Terraform module version for C3 AI Platform version v8.11. Reach out to your C3 AI Center of Excellence for the right version for your install
- myuser:mypassword is the C3 AI registry credentials, same as those for registry.c3.ai. Reach out to your C3 AI Center of Excellence for these.

Before executing any of the Terraform commands, you will need to have configured AWS CLI on the host that you intend on running the infrastructure deployment scripts from.

Bootstrap Module

The bootstrap module sets up the necessary permissions for the c3cluster module to be ran. See below for an example configuration of a Terraform overlay for C3 AI's bootstrap module.

bootstrap/main.tf

Create a new file at "./bootstrap/main.tf". Edit it and paste in the following contents:

```
module "bootstrap" {
  source       = "../modules/3.0.17/modules/bootstrap"
  cluster_name = "stgawscustomer"
  region       = "us-gov-west-1"
  account_id   = "123456789012"
  partition    = "aws-us-gov"

  trusted_identifier_arns = [
    "arn:aws-us-gov:iam::123456789012:role/c3-user-role-name"
  ]
  tags = { "mycustomtag1" : "example", "mycustomtag2" : "example" }
}

terraform {
  backend "s3" {
    bucket  = "my-terraform-state-bucket"
    key     = "stgawscustomer-bootstrap.tfstate"
    region  = "us-gov-west-1"
    encrypt = true
  }
}

provider "aws" {
  region = "us-gov-west-1"
}
```

Be sure to configure or set the following:

- "source" / the version in the module reference may need to change for your install depending on the infrastructure version.
- Replace "stgawscustomer" with your cluster name (see Preferred Cluster Naming Convention).
- Change "region" to your desired region. Be sure to communicate this to the C3 AI Center of Excellence.
- Account ID should be changed to your actual AWS GovCloud Account ID.
- Partition should be adjusted as needed.
- "trusted_identifier_arns" should change to the AWS IAM Role(s) that you will assign to C3 AI infrastructure administrator personnel.
- "tags" are optional and will add tags to all resources.
- "backend" should change to the remote location that you intend on storing the Terraform state

C3 AI provisions an AWS WAF by default on the public-facing Application Load Balancer (ALB), enabled out of the box and governed by the following variables: managed_waf_rules, waf_rate_based_rules, waf_enforce_managed_rules, and waf_cloudwatch_logging_enabled.

Customers with an existing firewall or edge protection solution must disable the C3-provisioned WAF to avoid duplicate or conflicting controls. To disable the WAF, set the following in your c3cluster configuration: waf_enforce_managed_rules = false.

The bootstrap module contains additional variables – if you need additional configuration parameters, look at the file at "modules/3.0.17/modules/bootstrap/variables.tf".

Deployment

Once you have configured the bootstrap main.tf file, you will need to run the following to initialize the Terraform install:

```
cd bootstrap
terraform init
```

This step will attempt to pull down all dependent Terraform providers and initialize your state backend.

Once that succeeds, you will need to then execute a Terraform plan:

```
terraform plan -out tfplan
```

Finally, once you have reviewed the plan and ensured its accuracy for your installation, then you can deploy the bootstrap infrastructure via the following:

```
terraform apply tfplan
```

Once the bootstrap infrastructure is successfully deployed, the C3 AI infrastructure administrator personnel can proceed with the c3cluster installation.

C3Cluster Module

For reference, the c3cluster module configuration is outlined below. The c3cluster module deploys all the compute and storage workloads as well as some additional IAM roles and policies. The overall deployment process is like bootstrap. Note that c3cluster can only be deployed after the bootstrap module has been successfully deployed.

c3cluster/main.tf

Create a new file at "./c3cluster/main.tf". Edit it and paste in the following contents:

```
module "c3cluster" {
  source       = "../modules/3.0.17"
  cluster_name = "stgawscustomer"
  c3_region    = "us-gov-west-1"
  account_id   = "123456789012"
  ip_allowlist = [{
    cidr_blocks  = ["1.2.3.4/32"],
    display_name = "C3 Allowed IPs"
  }]
  eks_version = "1.34"
  vpc_endpoint_use_fips = true
  eks_authorized_ips = ["10.1.2.3/32"]

  pg_instance_class = "db.m6i.xlarge"

  default_nodegroup_http_tokens = "required"
```

```
  enable_delete_protection = true

  eks_default_node_pools = {
    "c3ondemand" = {
      machine_type = "r6i.4xlarge"
      http_tokens  = "required"
    }
    "c3spot" = {
      machine_type = "r6i.4xlarge"
      http_tokens  = "required"
    }
    "c3fallback" = {
      machine_type = "r6i.4xlarge"
      http_tokens  = "required"
    }
    "c3gpu" = {
      machine_type = "g4dn.metal"
      http_tokens  = "required"
    }
    "c3obs" = {
      machine_type = "c6i.2xlarge"
      http_tokens  = "required"
    }
    "c3logs" = {
      machine_type = "r6i.2xlarge"
      http_tokens  = "required"
    }
    "c3cass" = {
      machine_type = "r6i.xlarge"
      http_tokens  = "required"
    }
  }

  tags = { "mycustomtag1" : "example", "mycustomtag2" : "example" }
}

terraform {
  backend "s3" {
    bucket  = "my-terraform-state-bucket"
    key     = "stgawscustomer-c3cluster.tfstate"
    region  = "us-gov-west-1"
    encrypt = true
  }
}

provider "aws" {
  region = "us-gov-west-1"
}
```

Be sure to configure or set the following:

- "pg_password" must be changed to a strong, unique password via terraform.tfvars or via a secrets manager. Never deploy with the module default.
- Ensure interface endpoints are routed to GovCloud FIPS-validated endpoints
- "source" / the version in the module reference may need to change for your install depending on the infrastructure version.
- Replace "stgawscustomer" with your cluster name (see Preferred Cluster Naming Convention).

- Change “region” to your desired region.
- Account ID should be changed to your actual AWS GovCloud Account ID.
- “tags” are optional and will add tags to all resources.
- “backend” should change to the remote location that you intend on storing the Terraform state. The exact state bucket and key must differ from bootstrap.
- “ip_allowlist” should be changed to contain the CIDR ranges of end user and C3 AI administrative IPs that should be allowed to access the C3 AI Platform deployment web interface.
- “eks_authorized_ips” should be changed to the CIDR ranges of IPs that should be allowed to access the EKS Kubernetes API endpoint. This likely will be a range or IP of your EC2 bastion host, a range of C3 AI VPN IPs if remote management is permitted.
- In general, “instance_class” and “machine_type” may need to be adjusted to reflect the machine types available in your region.

The c3cluster module contains many additional variables – if you need additional configuration parameters, look at the file at “modules/3.0.17/variables.tf”. There are also examples for various common configurations at “modules/3.0.17/examples”.

NOTE: The Terraform module enables deletion protection by default for EKS clusters, RDS instances, S3 buckets, and KMS keys (enable_delete_protection = true). This prevents accidental destruction of production infrastructure. Set enable_delete_protection = false only for test/development environments.

Deployment

Once you have configured the c3cluster main.tf file, you will need to run the following to initialize the Terraform install:

```
cd c3cluster
terraform init
```

This step will attempt to pull down all dependent Terraform providers and initialize your state backend.

Once that succeeds, you will need to then execute a Terraform plan:

```
terraform plan -out tfplan
```

Finally, once you have reviewed the plan and ensured its accuracy for your installation, then you can deploy the bootstrap infrastructure via the following:

```
terraform apply tfplan
```

Infrastructure deployment for c3cluster usually takes several minutes due to the startup times of EKS and RDS. Once the c3cluster infrastructure is successfully deployed, the C3 AI infrastructure administrator personnel can then proceed with the Helm installation of the C3 AI Agentic Platform if the engineer has sufficient access to the EKS cluster (i.e. private EKS deployment necessitates bastion host).

Network Resources

Below is a table of external network resources required for specific C3 AI Platform functionality, descriptions as to purpose and requirement, and suitable hosted alternatives in air-gapped or egress-limited environments.

For private or air-gapped deployments where nodes cannot reach public endpoints, configure VPC interface endpoints using the vpc_endpoint_services variable. Supply the required service short-names (e.g., eks, ecr.api, ecr.dkr, sts, kms, logs, monitoring, secretsmanager, elasticloadbalancing, ec2, ssm). For GovCloud/FedRAMP environments, also set vpc_endpoint_use_fips = true to use FIPS-compliant endpoints.

Egress (Outbound)

Note that all these resources are listed for outbound / egress access (from the cluster to the Internet), not ingress. If a resource is listed as required and has alternatives listed, then any one of the alternatives suffices in place of the original resource. List of hosted alternatives may not be exhaustive.

Resources / URLs	Purpose	Required?	Hosted Alternatives
Core Deployment + Platform			
AWS resource endpoints	EKS cluster operation and C3 AI Platform management of AWS resources	**Yes**	-
registry.c3.ai (C3 AI image registry)	Image registry for C3 AI images. C3 AI only requires images sourced from C3 AI's private container registry.	**Yes** Any OCI-compliant registry also suffices as long as all relevant C3 AI images are made available or mirrored into the registry.	Amazon Elastic Container Registry JFrog Artifactory Sonatype Nexus Red Hat Quay
Language Package Repositories			
files.pythonhosted.org **pypi.org** **pypi.python.org**	Python dependencies from PyPI. Required for the platform to dynamically create Python runtime environments.	**Yes**	C3 AI ArtifactHub (built into C3 AI Studio) JFrog Artifactory Sonatype Nexus
conda.anaconda.org **repo.anaconda.com** **repo.continuum.io**	Python dependencies as Conda packages. Required for the platform to dynamically create Python runtime environments.	**Yes**	C3 AI ArtifactHub (built into C3 AI Studio) JFrog Artifactory Sonatype Nexus
Source Code + ML Model Repositories			

github.com	Source code hosting and version control platform. Access needed for CI/CD builds of C3 AI applications for faster deployments.	No	GitLab Enterprise Azure DevOps Repos
huggingface.co	Repository for pre-trained machine learning models and datasets.	No	C3 AI Model Registry Service (built into C3 AI Studio) JFrog Artifactory Models stored in S3
C3 AI Central Managed Services			
c3ai.grafana.net	C3 AI's central cloud Grafana deployment for centralized monitoring	No Only required if central monitoring is permitted.	A locally hosted Grafana is included in a C3 AI Platform installation.
jfrog.c3.ai	C3 AI's JFrog Artifactory instance, typically used for pulling Terraform modules for IaC deployments.	No	One-time download of relevant Terraform modules per install/upgrade is sufficient.
prdgkemis.c3.ai	C3 AI's central Model Inference Service (MIS) endpoint. Not applicable for new or future deployments.	No	C3 AI endpoint for MIS (Management Information System) access
telemetry.c3.ai **44.230.42.147/32** **54.187.151.165/32**	C3 AI central telemetry endpoint for usage data.	No Only required if central monitoring is desired and permitted.	Telemetry data can be locally stored into deployed S3.
vault.c3iot.io	C3 AI's HashiCorp vault instance for external secrets and credentials management.	No Only necessary for centrally managed public cloud deployments. See the commercial C3 AI AWS install guide.	-
Application / User Case Specific			
Other use-case specific endpoints	If your application requires other access to use-case specific resources, such as external databases, datastores, etc., then those must also be whitelisted.	**Yes**	-

Ingress (Inbound)

Inbound access is generally only required from end user IP ranges over HTTP(S) ports and from specific administrative IPs for SSH access if solutions like AWS SSM are not being used. This table specifically lists external to internal traffic, not traffic privately routed between deployed resources.

Resources / URLs	From	To	Purpose	Required?	Alternatives

End user web access (HTTP/HTTPS)	Allowed inbound end user IP CIDR blocks	DMZ subnets or equivalent	For end users to access deployed C3 AI applications. HTTP is required to automatically upgrade to HTTPS.	Yes	-
Administrative access to bastion (SSH)	Allowed inbound IP admin CIDR blocks	DMZ subnets or equivalent	SSH access to bastion host if SSH is being used	Yes (if SSH is being used)	AWS SSM
Deployed S3 bucket	Allowed inbound IP admin CIDR blocks	Data subnets or equivalent	For seeding C3 AI Platform configuration	Yes (if S3 deployed with private networking)	Bastion host + AWS CLI
Deployed EKS cluster (:443)	Allowed inbound IP admin CIDR blocks	EKS subnets or equivalent	For EKS cluster management	Yes (if EKS deployed with private networking)	Bastion host + AWS CLI

Preferred Cluster Naming Convention

The cluster name is used throughout the deployment to name AWS resources, IAM roles, security groups, subnets, and Kubernetes objects. Choosing a name that follows the required convention is critical. Cluster names can be configurable to support customer requirements. Changing a cluster name after deployment requires rebuilding infrastructure.

Environment	Format	Example
Dev / QA	stgaws{customerabbreviation}	stgawscust
Production	prdaws{customerabbreviation}	prdawscust

Cluster name should be alphanumerical, lowercase, and at most 15 characters long.