C3 AI Installation Guide – Microsoft Azure

Version 8.5

9 December 2024

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 deployments, C3 AI provides a dedicated subscription with dedicated compute, storage, and networking resources. All cloud resources are dedicated to your subscription 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 Microsoft Azure cloud account.

C3 AI customer-managed deployment

You can optionally deploy the C3 AI cluster in your own Azure Virtual Network (VNet), 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 Azure Virtual Network (VNet) allows you to provision a logically isolated section of the Azure Cloud where you can launch Azure resources in a virtual private secure network. The VNet 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 VNets in your own Azure subscription.
- An approval process to create a new VNet, in which the VNet is configured, 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 Azure subscription. And you do not need to grant C3 AI as many permissions as you do for a C3 AI-managed cloud deployment. For example, there is no need for permission to create VNets.
- Maintain more control of your own Azure account and limit outgoing connections.

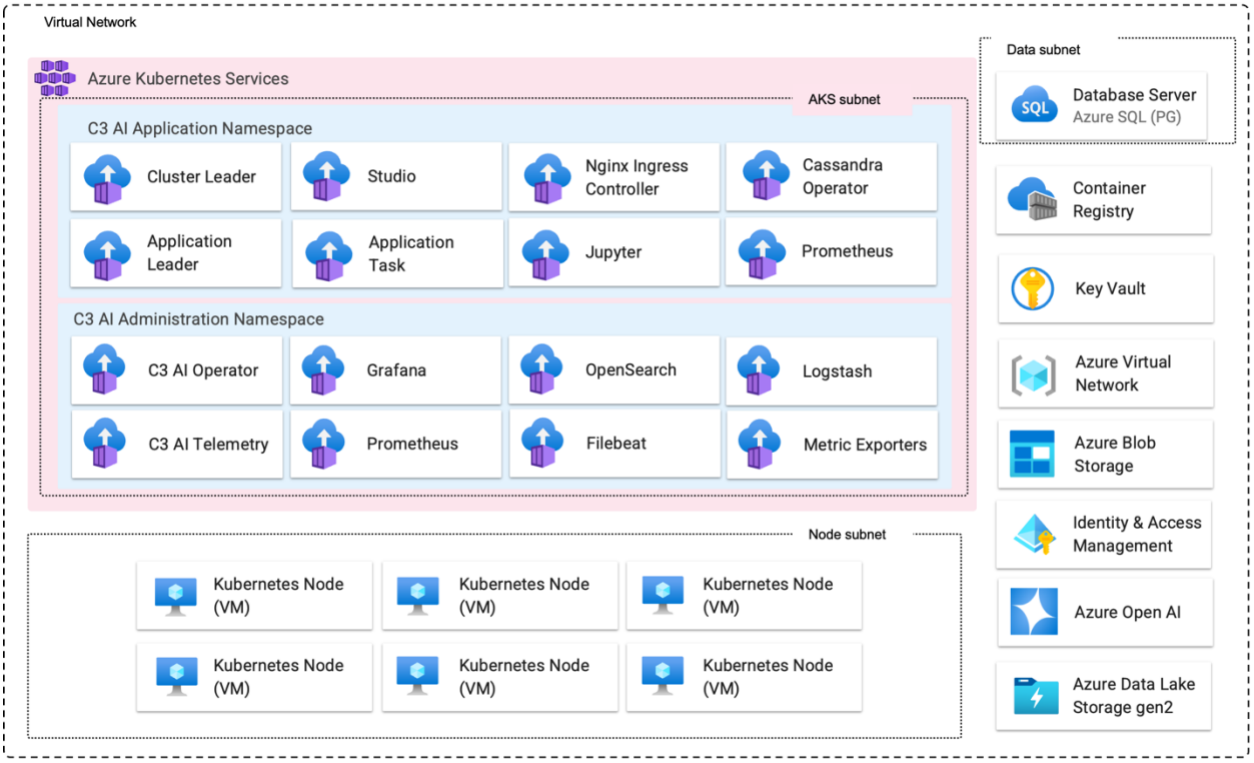

Figure 1. Microsoft Azure Architecture with C3 AI Platform Deployment

Requirements

The C3 AI Platform requires specific Microsoft Azure 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.

NOTE: Contact the C3 AI Center of Excellence (CoE) for requirements detailed in the Build of Materials (BOM) for the target major.minor release. High-level requirements are located on the Developer Portal for patch versions https://developer.c3.ai/85x-c3-ai-platform-install-and-requirements.

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 Microsoft Azure cloud services

The table below describes the Microsoft Azure 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.

Microsoft Azure Service	Version	Description
Azure Kubernetes Engine (AKS)	1.30	Operating environment responsible for the deployment, scaling, and management of the C3 AI Platform and C3 AI Applications.
Azure Key Vault	Current version	Securely store and manage sensitive information such as secrets, keys, and certificates.
Azure PostgreSQL (Flexible servers)	15	Relational data required for internal operations of the C3 AI Platform.
Azure Blob Store	Current version	Reliable and secure object storage used for the management of application and platform configuration and other ancillary tasks.
Azure Identity and Access Management (IAM)	Current version	Fine-grained access control and visibility for centrally managing cloud service account resources.
Azure Resource Manager	3.89.0	To get and set resources (IAM policy) by project.
Azure Virtual Network (VNet)	Current version	Dedicated, isolated network for inter-C3Cluster communication.

Microsoft Azure Service	Version	Description
Azure Virtual Machines	Current version	Compute instances required by Azure Kubernetes Engine.
Azure Open AI Service	Current version	Advanced AI models developed by OpenAI, such as GPT-4, DALL-E, and Codex.
Azure Data Lake Service gen2	Current Version	Data storage solution designed for big data analytics.

Microsoft Azure 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 Azure subscription	When creating the project, C3 AI requires: (1) Subscription identifier, (2) Cloud region.
Secure internet access to the Azure cloud subscription	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, Azure Command-Line Interface (CLI) v2.49.0, kubectl v1.30, Helm v3.12+, HashiCorp Terraform (1.7.0, < 1.8.0), Python 3, and Docker.
Access to C3 AI and 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 Azure Container registry, JFrog, and Anaconda Enterprise.
X.509 certificate for terminating network encryption	A fully qualified domain name for C3 AI 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 Azure Virtual Network (VNet), you must create the VNet following the steps enumerated in the VNet requirements section below.

VNet requirements

Your VNet must meet the following requirements to host a C3 AI cluster.

- Azure subscription
- VNet region
- VNet sizing
- VNet IP address ranges
- DNS
- Subnets
- Security groups
- Subnet-level network ACLs

Azure subscription

The Azure subscription containing the C3 AI Platform must have end-to-end encryption enabled using encryption at host. When you enable encryption at host, data stored on the host is encrypted at rest and flows encrypted to the Azure Storage service. Encryption at host must be enabled and can be accomplished using the Azure Portal or CLI. To enable encryption at host using the Azure CLI, run the following.

```
az feature register --name EncryptionAtHost --namespace Microsoft.Compute
```

VNet region

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

VNet sizing

You can share one VNet with multiple clusters in a single Azure subscription. However, you cannot reuse subnets or security groups between clusters. Be sure to size your VNet and subnets to C3 AI specifications.

VNet IP address ranges

C3 AI does not limit netmasks for the VNet, but each workspace subnet must have a netmask between /16 and /22.

DNS

The VNet must have DNS hostnames and DNS resolution enabled.

Subnets

C3 AI must have access to at least two subnets for each cluster. There should be one (1) of each of the following subnets:

- DMZ public subnets for load balancing
- Data private subnets for Postgres
- AKS private subnets for the AKS cluster
- AKS Pod private non-routable subnets for pods running in the AKS cluster

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

NOTE: Additional subnets might be required depending on whether it is a C3 AI-managed or customer-managed deployment: including, Azure Bastion, Tool, Key Vault, and Service Account subnets. Contact the C3 AI CoE for more information.

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 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 Azure 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)

- Allow all TCP and UDP access to the workspace security group (for internal traffic)
- 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 VNet 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 Azure subscription.

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 “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 Azure account, you must have the following:

- An Azure subscription and Azure resource group.
- An account-level admin user in your Azure 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 to matches the client hardware from which you will run the Terraform scripts.
    - The Azure CLI, signed in through the `az login` command with a user that has `Owner` rights to your subscription to access Microsoft Azure Cloud. See How to install the Azure CLI and Sign in with Azure CLI for more information.
    - The Kubernetes CLT - kubectl. See the Kubernetes website for more information about kubectl and related commands for infrastructure creation and deployment.
    - The Helm CLI. See Installing Helm on the Helm website for more information.
- Privileges to deploy, operate, and delete the infrastructure services. See the "README.md" file in the downloaded Registry folder for the most up-to-date information.

NOTE: As a security best practice, when authenticating with automated tools, systems, scripts, and apps, C3 AI recommends you sign in through the `az login` command with an Azure Active Directory (Azure AD) service principal. See Sign in with a service principal and Authenticating with Azure Service Principal for more information.

Installation Steps

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

1. Create the Microsoft Azure Virtual Network (VNet) and required Azure services.
2. Validate the configuration of the VNet and required Azure services and provide C3 AI Operations access to the cluster.

To create a VNet, C3 AI requires the use of HashiCorp Terraform and will provide a set of Terraform scripts to assist you in the creation of the VNet and required Azure services.

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. See the README.md file in the downloaded Registry folder for the most up-to-date information.

Terraform Module	Description
aks-cluster	Configures Azure Kubernetes Service (AKS), including VNet configuration, endpoint access, authorized IP addresses, and the version of Kubernetes used by the cluster.
aks-nodepool	Configures the AKS node groups, including default instance size, required subnet, and permissions assigned to each node.
aks-sa	Configures the managed identity to be used by the C3 AI cluster.
bootstrap	Configures the necessary Identity and Access Management (IAM) roles and policies to allow a Terraform orchestrator to deploy all services required for the C3 AI Platform on Azure.
c3cluster	Coordinates the execution of all other Terraform modules.
delegated-iam	Configures the IAM roles and role assignment to the C3 AI cluster.
firewall	Configures ingress and egress security rules.
iam	Configures the required IAM roles and policies.
kms	Configures the Azure Key Management service.
network	Configures the VNet, including public and private subnets, internet gateway, CIDR blocks, DHCP, and NAT.
postgres	Creates an Azure PostgreSQL database and assigns the database to the database subnet.
resource-group	Seeds the application role and secret in the Azure Secrets Manager.
storage-account	This module configures the Azure storage account 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 VNet and required Azure services

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

1.1 Run the bootstrap module

This module creates the necessary IAM roles and policies to configure the VNet and required Azure 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:

- For Dev and QA clusters: <stg><cloud><customerabbreviation>; in which <cloud> is az. For example, `stgazcust`
- For Production clusters: <prd><cloud>customerabbreviation>; in which <cloud> is az. For example, `prdazcust`
- The cluster name should not include a hyphen and must start with a letter; only lowercase letters and numbers are allowed with no other special characters or diacritics (accented letters); and should be less than 15 characters total.

NOTE: For more configuration options, download the Terraform module from C3 AI Registry folder provided by C3, and view the “Inputs” section in the main README.md file.

```
module "bootstrap" {
  source                  = "<c3_url>/tf-registry__c3/azure/c3//modules/bootstrap"
  version                 = ">VERSION_NUMBER"
  cluster_name            = "CLUSTER_NAME" # Replace with Name of c3 deployment
  region                  = "REGION"
  setup_role_principal_id = "IDENTITY_OBJECT_ID" # Object_id of the identity used to
assume the infrastructure creation role

}


provider "azurerm" {
  features {}
  partner_id      = "AZURE_PARTNER_ID" # Microsoft partner ID, please reach out to
C3 AI CoE if you don't know it. (Optional)
  tenant_id       = "AZURE_TENANT_ID"
  subscription_id = "AZURE_SUBSCRIPTION_ID"
}

terraform {
  required_version = ">= 1.7.0, < 1.8.0"
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "3.89.0"
    }
```

```
  }
}
```

NOTE: Replace:

- CLUSTER_NAME with the name of the C3 AI cluster. The cluster name must adhere to the following restrictions:
  - For Dev and QA clusters: <stg><cloud><customer_abbreviation>; in which <cloud> is az. For example, `stgazcust`
  - For Production clusters: <prd><cloud>customer_abbreviation>; in which <cloud> is az. For example, `prdazcust`
  - The cluster name should not include a hyphen and must start with a letter; only lowercase letters and numbers are allowed with no other special characters or diacritics (accented letters); and should be less than 15 characters total.
- VERSION_NUMBER with the version of the `bootstrap` module listed on the C3 AI BOM for the release version.
- REGION with the region where the infrastructure will be deployed. See the Azure regions mapping list for more information.
- IDENTITY_OBJECT_ID is the object identifier used to assume the infrastructure creation role. To get the Object ID, navigate to "Azure Active Directory", search for User, then select the user and get their Object ID.
- AZURE_PARTNER_ID is the Microsoft partner ID. Contact the C3 AI CoE for more information.
- AZURE_TENANT_ID with the Azure tenant where the C3 AI Platform will be deployed.
- AZURE_SUBSCRIPTION_ID with the Azure subscription ID.

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 Azure – Install CLI 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.

Be sure to login as the service principal that you specified as `setup_role_principal_id` in the bootstrap module.

Contact your account manager for C3 Control IPs (the list of IP addresses required by C3 AI). These IP addresses enable access for C3 AI Services to manage and maintain the C3 AI cluster. Replace the `CIDR_TO_WHITELIST` parameter below with the list of IP addresses.

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

NOTE: C3 AI infrastructure Terraform modules create a new VNet and subnets. If your organization must separately create these network artifacts, the Terraform module can be modified to utilize them rather than create new ones. For details, refer to the examples/existing_network/README.md file contained in the Terraform module documentation.

main.tf

```
module "c3cluster" {
  source       = "<c3_url>/tf-registry__c3/azure/c3"
  version      = ">VERSION_NUMBER"
  c3_region    = "REGION"
  cluster_name = "CLUSTER_NAME" # Replace with Name of C3 deployment

  # Please reach out to C3 CoE to obtain C3 control Ips
  ip_allowlist = [
    "CIDR_TO_WHITELIST",
  ]

  # Please reach out to C3 AI CoE to obtain the list of Domains to whitelist for
CORS policy
  storage_cors_domains = ["http://*.DOMAIN_NAME"]
  pg_create_mode       = "Default" # Only use Default at creation time

}

provider "azurerm" {
  features {}
  partner_id      = "AZURE_PARTNER_ID" # Microsoft partner ID, please reach out to
C3 AI CoE if you don't know it
  tenant_id       = "AZURE_TENANT_ID"
  subscription_id = "AZURE_SUBSCRIPTION_ID"
}

terraform {
  required_version = ">= 1.7.0, < 1.8.0"
```

```
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "3.89.0"
    }
  }
}
```

NOTE: Replace:

- AZURE_TENANT_ID with the Azure tenant where the C3 AI Platform will be deployed.
- AZURE_SUBSCRIPTION_ID with the Azure subscription ID.
- CIDR_TO_WHITELIST with the list required C3 AI IP addresses.
- REGION with the Azure region where the C3 AI Platform will be deployed. See the Azure regions mapping list for more information.
- CLUSTER_NAME with the name of the C3 AI cluster. The cluster name must adhere to the following restrictions:
    - For Dev and QA clusters: <stg><cloud><customerabbreviation>; in which <cloud> is az. For example, `stgazcust`
    - For Production clusters: <prd><cloud>customerabbreviation>; in which <cloud> is az. For example, `prdazcust`
    - The cluster name should not include a hyphen and must start with a letter; only lowercase letters and numbers are allowed with no other special characters or diacritics (accented letters); and should be less than 15 characters total.
    - VERSION_NUMBER with the version of the `c3cluster` module listed on the C3 AI BOM for the release version.

1.3.1 Implement CORS policy for C3 AI Ex Machina

If the installation of the C3 AI Platform includes C3 AI Ex Machina, setting the C3 AI CORS domain is all that is necessary. The CORS policy facilitates file uploads for C3 AI Ex Machina.

See `storage_cors_domains` in the main.tf example above.

Also, see the cors_rules.tf template example in the Terraform modules for more configuration details.

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

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

1.5 Add the Microsoft Azure Workload Identity

In version 8.5 this step is being handled at the terraform level using the "workload_identity_enabled" here Terraform Registry which ultimately runs the helm command here Mutating Admission Webhook - Azure AD Workload Identity.

For information purposes only - To integrate the C3 AI Cluster with the Microsoft Azure cloud service provider, you need to set up and deploy the Microsoft Azure Workload Identity. Once the Microsoft Azure Workload Identity is setup and deployed, the C3 AI Cluster is authorized to access other services within the Microsoft Azure cloud service.

See the Microsoft Azure Workload Identity website for more information.

See the Deploy an Azure Kubernetes Service (AKS) website for steps to connect to the Kubernetes Azure Cluster.

Limitations on Workload Identity for Secure Access to Azure

If your application or service relies on Workload Identity for secure access to Azure resources, you need to be aware of certain limitations when ingesting data from Parquet or AVRO files. Specifically:

- **Unsupported Features**: Several platform features are not supported when using Workload Identity on Azure, including the use of Parquet and AVRO for data ingestion, C3 AI Datasets, Spark in Jupyter (when interacting with the filesystem), and Ex Machina. While these features are essential for many applications, they currently do not function correctly with Workload Identity.
- **Alternative Authentication Methods**: Ingestion of data in Parquet and AVRO formats is still possible, but it requires using a different authentication method than Workload Identity. You must configure the cluster (the computing resource or environment where data processing occurs) to use service principal credentials instead. This may involve additional setup to ensure the service principal has the appropriate permissions to access the data.
- **Service Principal Authentication**: A service principal is an identity created for use with applications, hosted services, and automated tools to access Azure resources. Using service principal credentials involves providing a client ID and a client secret or certificate for authentication, rather than relying on Workload Identity.

While these limitations currently exist, they highlight the need for careful planning when designing applications that will rely on Workload Identity for Azure resource access, particularly regarding data ingestion scenarios.

2. Validate the configuration of the VNet and required Azure services and provide C3 AI Operations access to the cluster

After the VNet 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 VNet 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.

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

Contact the C3 AI CoE for more information and to obtain the C3 AI Cluster Validation Utility.

Run the C3 AI Cluster Validation Utility to determine whether the infrastructure requirements are fulfilled to allow the C3 AI Operations to deploy the C3 AI Platform.

If the C3 AI Cluster Validation Utility indicates the VNet is ready for C3 AI Operations to deploy the C3 AI Platform on the Kubernetes cluster, provide the output to C3 AI Operations.

If the output indicates the VNet is not ready, remediate all exceptions and rerun the C3 AI Cluster Validation Utility.

2.2 Provide C3 AI Operations access to the cluster

In addition to the output of the C3 AI Cluster Validation Utility, you must provide C3 AI Operations with the following.

Title	Description
C3 AI Operations credentials	Credentials for C3 AI Operations team members
Subscription ID	Subscription ID used in the execution of the Terraform scripts
Tenant ID	Azure Tenant ID used in the execution of the Terraform scripts

Title	Description
Resource Group Name	From Azure Portal, go to “Resource groups” and search “CLUSTERNAME”. By default, it should be CLUSTERNAME-rsgp-c3-01.
AKS cluster name	The name of the AKS cluster where the C3 AI Platform will be installed. By default, this will be CLUSTERNAME-kube-01; confirm this by going to “Kubernetes services” in the Azure Portal. **NOTE**: The cluster name must adhere to the following restrictions: • For Dev and QA clusters: <stg><cloud><customerabbreviation>; in which <cloud> is az. For example, `stgazcust` • For Production clusters: <prd><cloud>customerabbreviation>; in which <cloud> is az. For example, `prdazcust` The cluster name should not include a hyphen and must start with a letter; only lowercase letters and numbers are allowed with no other special characters or diacritics (accented letters); and should be less than 15 characters total.
Region	The Azure region associated with the AKS cluster, from “Kubernetes services” → the AKS cluster → Overview → Location. See the Azure regions mapping list for more information.
Azure SQL Postgres endpoint	From Azure Portal, go to “Azure Database for PostgreSQL servers” → CLUSTERNAME-pg-shared → Overview → Server name
Azure SQL Postgres credentials	Credentials required for the C3 AI Platform to connect to PostgreSQL. These can be set by pressing “Reset password” in the Azure Portal at “Azure Database for PostgreSQL servers” → CLUSTERNAME-pg-shared.
C3 Managed Identity Client ID	From Azure Portal, go to “Managed identities”, filter to the resource group CLUSTERNAME-rsgp-c3-01, and click CLUSTERNAME-c3-mi-01. Share the “Client ID”.
Storage Account Keys	From Azure Portal, go to “Storage accounts” → filter to resource group CLUSTERNAME-rsgp-c3-01, and click the one named c3CLUSTERNAME. Select “Access Keys” and share key1.
Domain name	A fully qualified domain name for C3 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.

It is strongly recommended that the sharing of Postgres credential and certificates occur using Azure Vault.

To grant C3 AI Operations AKS cluster administration permissions in the subscription, create a new role assignment for a C3 AI Operations user, granting them the Azure Kubernetes Service Cluster Admin role.

```
az login

az role assignment create \
  --assignee <OBJECT_ID_OF_THE_USER> \
  --role "Azure Kubernetes Service Cluster Admin Role" \
  --scope
/subscriptions/<SUBSCRIPTION_ID>/resourceGroups/<C3_CLUSTER_RESOURCE_GROUP>
```

NOTE: Replace:

- <OBJECT_ID_OF_THE_USER> with the user, group, or service principal. Supported format: object id, user sign-in name, or service principal name.
- <SUBSCRIPTION_ID> with the subscription identifier.
- <C3_CLUSTER_RESOURCE_GROUP> with the resource group associated with the cluster.