Duality Platform

User Guide and Platform Overview

Table of contents

Prerequisites

The scope of this topic is to detail the prerequisites for the Duality platform installation. The prerequisites vary depending on the usage and computations that run in the collaboration.

For a data collaboration project, prerequisites must be met by each of the participating parties.

Node Software Requirements

The following describes the software requirements for each node:

- **Operating System:** Linux Server X86_64 - Ubuntu 24.04 / RHEL 9.0+ (*Tested with RHEL 9.0 through 9.4*)
- **Software:** unzip (sudo apt install unzip)

> **NOTE:**
> Running any 3rd party application on the same node as the Duality Platform node may impact performance and stability and therefore dedicated nodes are required.
> For information regarding hardware requirements or performance optimization for a specific use case or application, contact Duality support at support@dualitytech.com.

Infrastructure Requirements

The following describes the infrastructure requirements for each node:

Infrastructure	Collaboration Manager	Node*
CPU	32 vCPU	32 vCPU
Memory	64GB RAM	64GB RAM
Storage	500GB SSD	500GB SSD (based on use case requirements)
AWS instance	c6a.8xlarge	c6a.8xlarge
GCP instance	c2d-highcpu-32	Node - c2d-highcpu-32 Enclave - n2d-standard-16
Azure instance	F32s v2	F32s v2

Node* refers to all participant roles.

TEE Server

The TEE server is used to run the secure computation. Spinning up the server to be used as the TEE is described in the Install and Configure Enclave (GCP article, AWS article) article.

NOTE: Platform-Specific Guidelines for Enclave AI Projects

AWS

- A single machine serves as both the compute node and the enclave host.
- Use the same instance type listed for Node*, but **double the vCPU and memory**
- Storage remains the same as a regular participant node.

GCP

- The compute node uses the standard Node* instance type.
- A separate VM must be created for the enclave, sized according to the table (n2d-standard-16).
- For test encironments VMs can be sized down to c2d-highcpu-16.

Network and Connectivity

For a visual representation of this setup, see the Communication Map page.

The following describes the **connectivity** requirements between the participating nodes:

From	To	Port	Direction	Purpose
All Nodes	Collaboration Manager	TCP/443	Inbound	Control/Management API over HTTPS/HTTP
All Nodes	Collaboration Manager	TCP(TLS)/5671	Inbound	Internal System Comm. and Data Transfer over RabbitMQ
All Nodes	Collaboration Manager	TCP(TLS)/9200	Inbound	Log Transfer to Elasticsearch
Client	Any Node	TCP/443	Inbound	Internal Users Web (+Logs Kibana) /API over HTTPS/HTTP
All Nodes	Collaboration Manager	TCP(TLS)/9000	Inbound	Internal System Data Transfer to MinIO
Asset Owner Nodes	Compute Node	TCP/8003, TCP/8002	Inbound	*FL Data Transfer (intermediate results) - flare_admin_port, flare_port
Asset Owner Nodes	Compute Node	TCP/8083, TCP/8082	Inbound	*FL Data Transfer in TEE (intermediate results) - flare_admin_port, flare_port
Compute Node	Enclave Node	TCP/6250	Inbound	*Required for GCP installations running AI projects.. Enables secure data transfer between the Compute node and the Enclave node. Used for encrypted payload exchange during FL execution within the TEE.
Asset Owners & Analyzing Party	Internet			*Required for Enclave installations in AI projects. Enables completing the attestation service with the CSP attestation server.

> **NOTE:** Alternative ports can be configured based on requirements.

The following describes the **network** requirements:

1. **SSH:** If an SSH connection is used, access to the machine terminal is enabled via port TCP/22.
2. **SSM**: If an AWS Systems Manager (SSM) connection is used, the VM must have the **SSM Agent** installed and configured with the appropriate **IAM permissions**. Additionally, the VM must be able to reach `ssm.<region>.amazonaws.com` either through a **public IP and internet gateway** or via **VPC endpoints for SSM** if internet access is restricted.
3. **Static IP:**
    a. The Collaboration Manager needs to be accessible on the same IP address from all participants.
    b. All participants must also have a static IP that they use to connect from.
4. **FQDN**: For AI projects, the FQDN must be set to the Compute node's public IP address.

Firewall SSL Inspection

The Collaboration Manager and nodes communicate using end-to-end TLS encryption. If SSL inspection is enabled on the firewall, the firewall intercepts and re-signs TLS traffic, breaking the certificate trust chain and preventing successful communication between the CM and nodes.

To solve this issue, disable SSL inspection on the firewall for all traffic between the Collaboration Manager and the nodes.

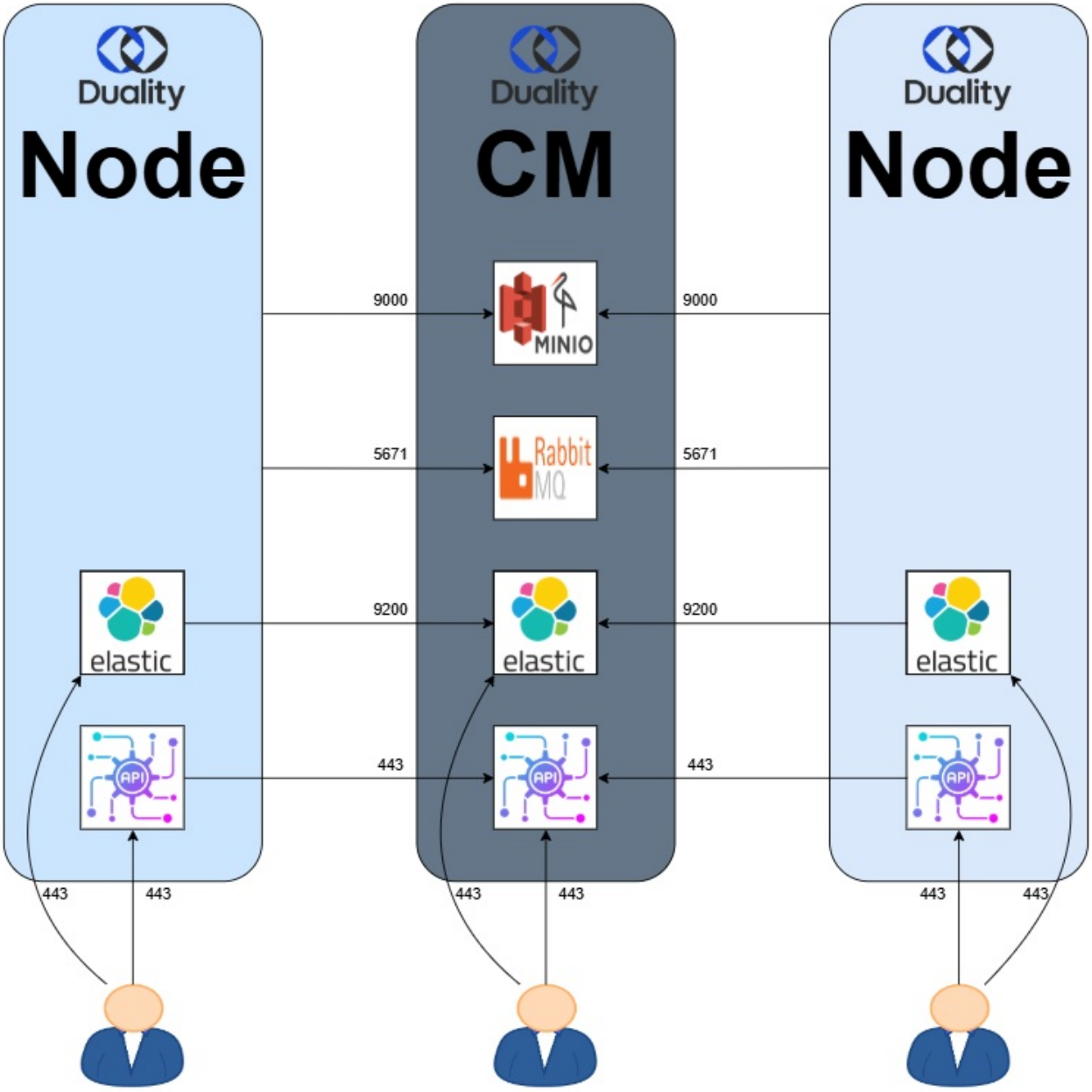
> **NOTE:** Health checks or telnet tests verify basic TCP connectivity but do not validate SSL trust. TLS handshake failures may still occur if SSL inspection is enabled.

Offline Installation

NOTE: For offline installation, contact Duality Support at **support@dualitytech.com** for the required prerequisites (as specified in the setup.sh script).

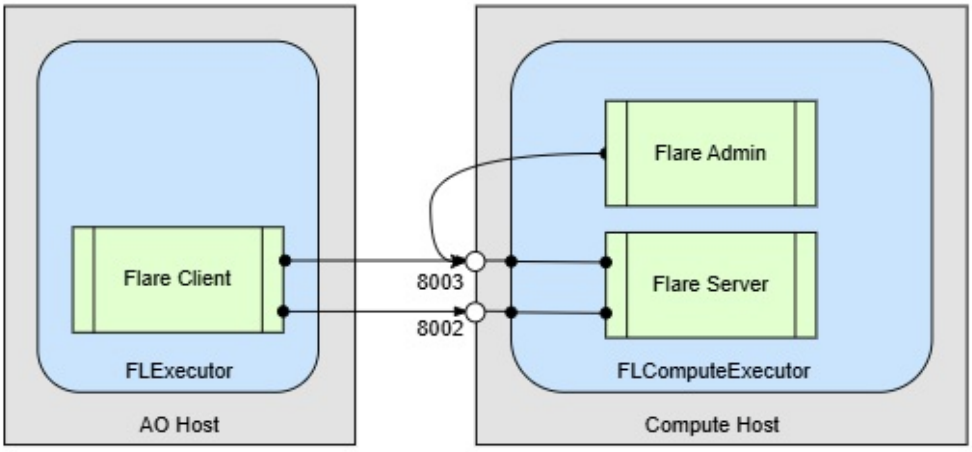
Communication Map

Connectivity Requirements for All Project Configurations



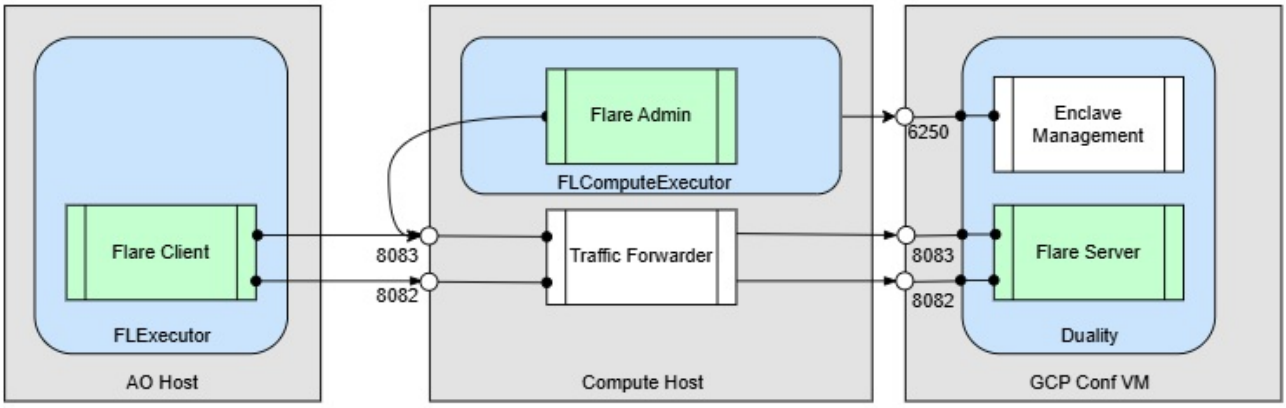
Connectivity Requirements for Different AI Project Configurations

Communication between the Asset Owner and the Compute when no TEE is involved in the collaboration:



GCP TEE:

Communication between the Asset Owner, Compute, and Enclave when a TEE is used in the collaboration:



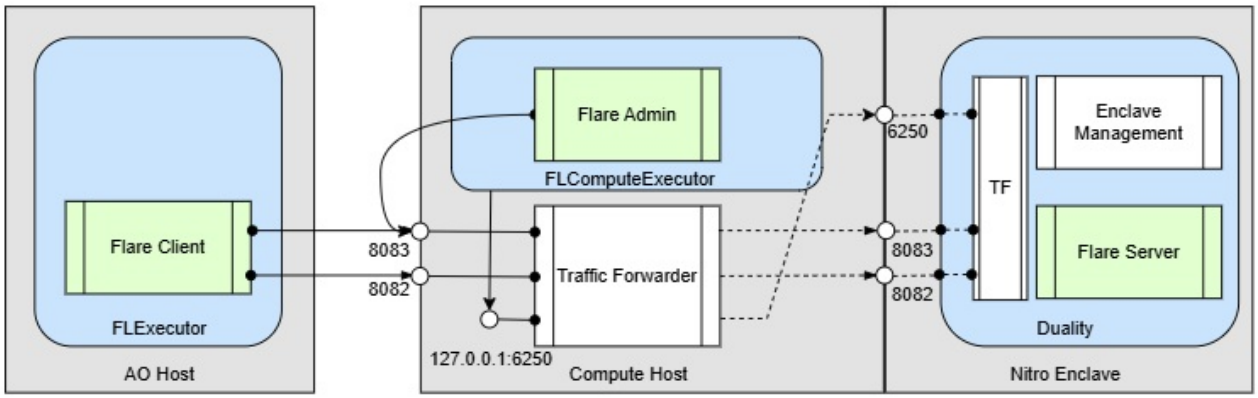
NOTE:

The port forwarder (also referred to as the proxy service) is responsible for securely relaying network communication between the Compute Node and the Enclave, enabling FL workflows within a TEE. The proxy must be installed, configured, and started on the host (Compute Node) in order for the enclave to properly receive input data and return results.

For detailed installation and configuration steps on GCP, see: **Install - GCP Enclave**

AWS TEE:

Communication between the Asset Owner, Compute, and Enclave when a TEE is used in the collaboration:



NOTE:

The port forwarder (also referred to as the proxy service) is responsible for securely relaying network communication between the Compute Node and the Enclave, enabling FL workflows within a TEE. The proxy must be installed, configured, and started on the host (Compute Node) in order for the enclave to properly receive input data and return results.

For detailed installation and configuration steps on AWS, see: **Install - AWS Enclave**

Data and 3rd-Party Integrations

Supported Data Sources

Data sources are **connected via read-only ODBC**. Access on the standard port number* is required.

The latest stable version of the database server is recommended. The versions below are indicative only:

Database	Supported Version	Port Number*
MySQL	8.0	3306
PostgreSQL	16.8	5432
Oracle	19	1521
MSSQL	SQL Server 2019	1433
Snowflake	-	

Note: Additional data sources can be supported on request.

Supported Integrations

Integration	Latest Supported Version
Browser - Chrome	Version >= 138

Supported Object Stores

Object Store	Provider
Amazon S3	AWS
Google Cloud Storage	GCP
Azure Blob Storage	Azure
Databricks FileStore	Databricks

Index Query - Memory Allocation

Index Query efficiently searches and retrieves encrypted data, significantly enhancing query performance and scalability. For additional information, refer to Index Query

When creating an Index Query, the platform requires specific disk space and memory allocations. It is essential to consider these requirements when determining the resource needs for the Data Owner node. The memory allocation for the Index Query must be added to the infrastructure requirements.

Calculate the additional memory as follows:

Plaintext | Copy

```
Memory required for the number of concurrent Index Queries + Memory for the largest Index Query
```

Example

You are planning to have five Index Queries (EXISTS operator) in the platform:

- Two Index Queries running on a table of 10M records
- Three Index Queries running on a table of 1M records

You decide to allocate fixed memory for only two concurrent Index Queries. One for the 1M table and the other one for the 10M table.

In this case, you need 6 GB + 16 GB + 16 GB (this includes the additional memory allocation for the largest query). The total memory allocation required is 38GB.

Operation/# of Records	1M	10M	100M
EXISTS	6 GB	16 GB	128 GB
COUNT	7 GB	40 GB	N/A
RETRIEVE	6 GB-18 GB	42 GB-48 GB	N/A

Note: The exact required memory allocation is determined according to query characteristics.

Duality Recommendation

- For EXISTS up to 100M Records – 32vCPU 128 GB RAM
- For COUNT / RETRIEVE up to 10M Records – 32vCPU 128 GB RAM
- For COUNT / RETRIEVE up to 50M Records – 48vCPU 192 GB RAM
- For COUNT / RETRIEVE up to 100M Records – 64vCPU 320 GB RAM

For additional information, refer to Index Query or see detailed benchmarking results.

Appendix: Index Query Memory Requirement

These tables help you determine the required RAM and expected performance metrics for your Index Query based on database size and query batch size.

Memory Requirements - EXISTS (inc. offline duration, memory peak)

Database Size	Query Batch Size	Offline Time	Online RAM (GB)
1M	1	4.18s	1.7
	1024	7.92s	2.5
	11041	6.87s	2.5
16M	1	69s	10
	1024	134s	22
	4096	117s	23
	11041	112s	~24
50M	1	337s	~30
	1024	479s	~56
100M	1	664s	~58

Memory Requirements - RETRIEVE (inc. offline duration, memory peak)

Rows	Label Bytes	Batch Size	Offline Duration	Offline VmPeak (GB)	Load to Memory Time	Load to Memory VmPeak (GB)	Online Session Time
10M	8	1000	19m30s	42	52s	34	8s
12.5M	16	1000	5m56s	34	18s	32	8.4s
25M	16	1000	48m	63	35s	61	7.5s
50M	16	1000	2h	116	1m11s	120	6.7s
100M	16	1000	2h47m	221	2m54s	236	9.5s
200M	16	1000	7h35m	431	32m	469	21s
100M	16	1000	5h10m	221	2m52s	237	10s

Duality AI - Storage Inside the Enclave

Available Storage Inside the Enclave

AWS Nitro Enclaves do not have physical disk storage. Instead, a portion of the EC2 instance's memory is used to emulate storage inside the Enclave.

When estimating how much working space your Enclave will have (e.g., for writing intermediate results), use the following approximate formula:

Plaintext | Copy

```
Available Storage ≈ (EnclaveMemory[GB] / 2) - ImageSize[GB] - 2
```

Example (64GB EC2 Instance):

- EnclaveMemory ≈ 28 GB
- ImageSize ≈ 5 GB
- Meaning the available storage is ≈ 28/2 - 5 - 2 = ~7 GB

Install - Collaboration Manager

Duality Platform Installation - Collaboration Manager

The scope of this topic is to assist the user in the installation, setup, and configuration of the Duality Platform in a collaboration framework between two or more participating organizations (instances).

Each of the participants in the collaboration, the Collaboration Manager, or any of the nodes, is installed on a single machine.

The Duality Platform installation begins with verifying all prerequisites, followed by the automated setup using default configurations. The entire process typically completes within a few minutes. Detailed step-by-step instructions are provided below.

> **NOTE:**
> Throughout the document, the term 'Orchestrator' is used interchangeably with the 'Collaboration Manager' regarding the installation process. It is important to clarify that the Collaboration Manager is the actual product name, while the term 'Orchestrator' serves as a technical reference to the same entity.

Prepare the Installation Components

1. Verify that all requirements specified in the Prerequisites have been met.
2. Ensure SSH access (the "ubuntu" user is used in the following example).
3. Create a folder to place the installation files:

```
cd /home/ubuntu
mkdir <version_number>
```

5. Place the following installation file in the **/home/ubuntu/<version_number>** folder:
   node_deployment.zip
6. Unpack the deployment files with the **unzip** command:

```
unzip orchestrator_deployment.zip
```

- If the unzip utility is not installed, install it using: **sudo apt install unzip**

- If the unzip utility is not installed, install it using: **sudo apt install unzip**

> **NOTE:** To retrieve the files, contact Duality Support at support@dualitytech.com

Installation Process

> **NOTE:** Only use sudo if it is explicitly stated in the command. Do not use sudo to try and resolve issues unless instructed, doing so may cause permission or environment-related problems.

Setup and Configure the Duality Platform Software

1. Browse to the install folder, and set up the server:

Plaintext	Copy

```
cd install/
./setup.sh
```

2. The following message indicates that the setup completed: *"Setup stage completed"*
3. Restart your system now before continuing to the configuration stage, then reconnect to the server via SSH:

Plaintext	Copy

```
sudo reboot
```

4. Browse to the install folder, and configure the deployment:

Plaintext	Copy

```
cd install/
./configure.sh
```

5. Fill in the required information according to the prompt questions.

Installation Questionnaire - Collaboration Manager

> **NOTE:** Default values are indicated in [squared brackets].

By proceeding, you acknowledge that you have read and agree to the End User License Agreement (EULA)

Read the EULA: https://dualitytech.com/eula/

Read the EULA: https://duality.cloud/eula

Do you agree to the Duality Platform EULA? [y/N]:

------ Common configuration ------

1. Default admin username [admin]:

2. Default admin password (Minimum 7 characters) []:

3. Name for the Duality Platform deployment (Organization, Role, Etc) []: #*This is the name of your organization*

4. Public host address of the Duality Platform deployment []: *#The IP address or FQDN that other participants use to communicate with this node*

5. Select Duality Platform Communication Security Level (TLS_NO_VERIFY, TLS, MTLS) [TLS_NO_VERIFY]:

> **NOTE:** See Setting Up mTLS article for additional information.
> All participants in the collaboration must use the same TLS policy.
> This TLS policy cannot be changed after installation. Modifying it later is complex and may require reinstallation.
> Ensure all participants use the same value and select it carefully.

------ Docker configuration ------

Note: These configuration parameters must be provided by Duality support

6. Username for the docker registry []:

7. Password/Token for the docker registry []:

---- Central log configuration ---

> **NOTE:** See Logs and Audit article for additional information.

8. Which log types should be sent to the central log? (ALL, NON_SENSITIVE, NONE) [NON_SENSITIVE]:

9. Host of the central log server []: *#Define and share the Central Log host address with all participants.*

10. Participant password to central log []: #Define and share the Central Log participant password with all relevant participants.

The following message indicates that the configuration has been completed: *"Configuration stage completed"*

Pull and Start Services

NOTE: If the required Docker images are already available locally (for example, in offline or air-gapped environments), running the image pull command is not required.

1. Browse to the install folder, and start the services with the latest available images:

Plaintext	Copy

```
cd install/
./orchestrator.sh start --pull
```

The following message indicates that the installation completed: *"Deployment of Duality Platform completed"*

NOTE: The **--pull** parameter ensures authorization with the docker repository and downloads the latest available images. This process may take several minutes.
If the required docker images are already available locally, run the start command without the **--pull** flag.

Post-Installation Validation

Verify that the UI is Running and Accessible

1. Browse to: https://<Machine IP>/
2. Verify that the Web UI is available.
3. Proceed to Setup the Duality Platform Application and complete the relevant steps for your role.

Configuration Values to Share with Node Participants

After completing the Collaboration Manager installation, share the following values with each Node participant in your collaboration.
These details are required for them to complete their installation:

1. **Docker Registry Credentials** (for steps 6 and 7 of the configuration).
2. **Central Log Host Address** – The IP address or hostname configured for the central log server (step 9 of the configuration).
3. **Central Log Participant Password** – The password you configured for node participants to authenticate to the central log (step 10 of the configuration).
4. **TLS Security Level** – The TLS policy selected during configuration (step 5 of the configuration).

NOTE: All participants must use the same TLS policy.

Install - Node

Install Duality Platform - Node

The scope of this topic is to assist the user in the installation, setup, and configuration of the Duality Platform in a collaboration framework between two or more participating organizations (instances).

Each of the participants in the collaboration, the Collaboration Manager, or any of the nodes, is installed on a single machine.

The Duality Platform installation begins with verifying all prerequisites, followed by the automated setup using default configurations. The entire process typically completes within a few minutes. Detailed step-by-step instructions are provided below.

Prepare the Installation Components

1. Verify that all requirements specified in the Prerequisites have been met.
2. Verify that all required configuration values have been obtained from the Collaboration Manager
3. Ensure SSH access (the "ubuntu" user is used in the following example)
4. Place the **node_deployment.zip** installation file in the `/home/ubuntu/<version_number>` folder
5. Unpack the deployment files with the **unzip** command:

Plaintext	Copy

```
unzip node_deployment.zip
```

- If the unzip utility is not installed, install it using: **sudo apt install unzip**

> **NOTE:** To retrieve the files, contact Duality Support at support@dualitytech.com

Configuration Values to Obtain from Collaboration Manager

Before you begin configuring your Node, ensure you have received the following values from your Collaboration Manager:

1. **Docker Registry Credentials** – Provided by Duality Support or the Collaboration Manager (for steps 6 and 7 of the configuration).
2. **Central Log Host Address** – Provided by the Collaboration Manager (for step 9 of the configuration).
3. **Central Log Participant Password** – Provided by the Collaboration Manager (for step 10 of the

configuration).

4. **TLS Security Level** – The TLS policy selected for the collaboration (for step 5 of the configuration).

Installation Process

> **NOTE:** Only use sudo if it is explicitly stated in the command. Do not use sudo to try and resolve issues unless instructed, doing so may cause permission or environment-related problems.

Setup and Configure the Duality Platform Software

1. Browse to the install folder, and set up the server:

Plaintext	Copy

```
cd install/
./setup.sh
```

2. The following message indicates that the setup completed: *"Setup stage completed"*
3. Restart your system now before continuing to the configuration stage, then reconnect to the server via SSH:

Plaintext	Copy

```
sudo reboot
```

> **NOTE: For AWS deployments with Enclave (TEE)**
>
> If you are installing a **Compute Node** that will also **host an Enclave**, stop here.
> Do not proceed to step 3 (./configure.sh) and do not start the Node services until the Enclave installation is complete.
> Refer to the **TEE Server setup** instructions, and return to this step once the Enclave has been installed.

4. Browse to the install folder, and configure the deployment:

Plaintext	Copy

```
cd install/
./configure.sh
```

5. Fill in the required information according to the prompt questions.

Installation Questionnaire - Node

> **NOTE:** Default values are indicated in [squared brackets].

By proceeding, you acknowledge that you have read and agree to the End User License Agreement (EULA)

Read the EULA: https://dualitytech.com/eula/

Do you agree to the Duality Platform EULA? [y/N]:

------ **Common configuration** ------

1. Default admin username [admin]:

2. Default admin password (Minimum 7 characters) []:

3. Name for the Duality Platform deployment (Organization, Role, Etc) []: #*This is the name of your organization*

4. Public host address of the Duality Platform deployment []: *#The IP address or FQDN that other participants use to communicate with this node*

5. Select Duality Platform Communication Security Level (TLS_NO_VERIFY, TLS, MTLS) [TLS_NO_VERIFY]:

> **NOTE:** See Setting Up mTLS article for additional information.
> All participants in the collaboration must use the same TLS policy.
> This TLS policy cannot be changed after installation. Modifying it later is complex and may require reinstallation.
> Ensure all participants use the same value and select it carefully.

------ **Docker configuration** ------

Note: These configuration parameters must be provided by Duality support

6. Username for the docker registry []:

7. Password/Token for the docker registry []:

---- **Central log configuration** ---

> **NOTE:** See Logs and Audit article for additional information.

8. Which log types should be sent to the central log? (ALL, NON_SENSITIVE, NONE) [NON_SENSITIVE]:

9. Host of the central log server []: *#Obtain the Central log host address from the Collaboration Manager*

10. Participant password to central log []: *#Obtain the Central log participant password from the Collaboration Manager*

The following message indicates that the configuration has been completed: *"Configuration stage completed"*

Pull and Start Services

> **NOTE:** If the required Docker images are already available locally (for example, in offline or air-gapped environments), running the image pull command is not required.

1. Browse to the install folder, and start the services with the latest available images:

Plaintext	Copy

```
cd install/
./node.sh start --pull
```

The following message indicates that the installation completed: *"Deployment of Duality Platform completed"*

> **NOTE:** The **--pull** parameter ensures authorization with the docker repository and downloads the latest available images. This process may take several minutes.
> If the required docker images are already available locally, run the start command without the **--pull** flag.

Post-Installation Validation

Verify that the UI is Running and Accessible

1. Browse to: https://<Machine IP>/
2. Verify that the Web UI is available.
3. Proceed to Setup the Duality Platform Application and complete the relevant steps for your role.

TEE Server

If the node is designated as an Enclave, refer to the TEE Server setup instructions. The steps to spin up the TEE server are provided in:

- Install - AWS Enclave
- Install - GCP Enclave

Install - AWS Enclave

This chapter explains how to deploy a Compute Node with an integrated Nitro Enclave (TEE) on AWS. TEEs enhance security by isolating sensitive computations, ensuring data remains protected even during processing.

In AWS deployments, the Enclave runs within the same EC2 instance as the Compute Node, using isolated memory and CPU resources.

This step-by-step guide is designed for users with basic familiarity with AWS.

Installation Order - AWS Compute Node with Enclave:

When installing a Compute Node that will also host an Enclave:

1. You must first complete the Node installation up to and including the 'setup.sh' step.
2. Then, follow the steps in this guide to install and run the Enclave.
3. Once the Enclave is running, return to the Node installation and complete the configuration 'configure.sh' and node start.

Prerequisites

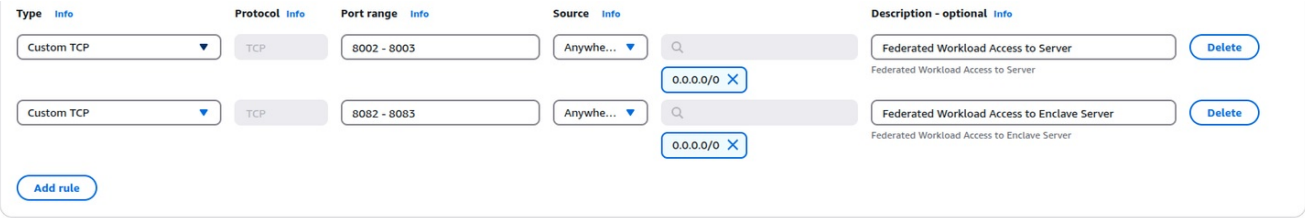
1. Admin Permissions:
    a. Ensure you have admin-level access to your AWS VPC to configure services and deploy resources.
    b. Access to the Internet is required for the Nitro Enclave CLI installation.

Installation & Configuration Flow

Step 1 - Ensure Network Connectivity

If Federated Workloads are expected, the Security Group for Compute-TEE should allow inbound access from Asset Owners who contribute data to ports **TCP 8002, 8003, 8082, and 8083**.

Inbound rules Info



Step 2 - Launch Compute-TEE Instance

1. Launch EC2 Instance following the Node Infrastructure Requirements (e.g c6a.8xlarge).
2. Make sure to associate the instance with the Security Group created in the previous step.
3. In **Advanced Details**:

- Make sure **Nitro Enclave** is set to **Enable**:

- Make sure **Metadata version** is set to **V2 only (token required)**:

Step 3 - IAM Role

An EC2 Service IAM Role has to be assigned to the instance. For Duality Platform-managed KMS, no specific permissions are required, but a role has to be assigned (“dummy role”).

Step 4 - IP Address

Make sure that the instance has a permanent public IP address (elastic IP) or FQDN.

Step 5 - Begin Duality Software Installation

Connect to the instance over SSH and install the Duality Platform Node as specified in the Install - Node guide, **up to and including Step 2** ‘setup.sh’.

Do not proceed to the configuration step ‘configure.sh’ or start the Node services until the Enclave installation is complete.

> **NOTE: If you are not using SSH to access the instance, contact Duality support at**

NOTE: If you are not using SSH to access the instance, contact Duality support at support@dualitytech.com

Step 6 - Install Nitro Enclave CLI on the Instance (Parent Side)

This step installs the Nitro Enclave CLI on the EC2 parent instance.

The CLI is required to interact with and manage Enclaves, for example, to build the Enclave image or run the Enclave.

1. Place the **setup_parent.sh** installation file in the `/home/ubuntu` folder

NOTE: To retrieve the file, contact Duality support at support@dualitytech.com

2. Make the setup script executable:

```
chmod +x setup_parent.sh
```

3. Run:

```
sudo ./setup_parent.sh
```

The script downloads, compiles, and installs Nitro Enclave CLI from the AWS code repository (Internet access is required).

4. When the installation is completed (this might take several minutes), reboot the instance.

```
sudo reboot
```

5. After rebooting, verify the installation. Check the **nitro-enclaves-allocator.service** status:

```
sudo systemctl status nitro-enclaves-allocator.service
```

Expected output:

```
```

```
● nitro-enclaves-allocator.service - Nitro Enclaves Resource Allocator
     Loaded: loaded (/lib/systemd/system/nitro-enclaves-
allocator.service; enabled; vendor preset: enabled)
     Active: active (exited) since Wed 2025-03-19 08:08:32 UTC; 2min 9s
ago
    Process: 694 ExecStart=/usr/bin/nitro-enclaves-allocator
(code=exited, status=0/SUCCESS)
   Main PID: 694 (code=exited, status=0/SUCCESS)
        CPU: 2.479s

Mar 19 08:08:30 ip-172-31-40-205 nitro-enclaves-allocator[694]: Will try
to reserve 28339 MB of memory on node 1.
Mar 19 08:08:30 ip-172-31-40-205 nitro-enclaves-allocator[694]:
Configuring the huge page memory...
Mar 19 08:08:32 ip-172-31-40-205 nitro-enclaves-allocator[694]: -
Reserved 27 pages of type: 1048576kB.
Mar 19 08:08:32 ip-172-31-40-205 nitro-enclaves-allocator[694]: -
Reserved 346 pages of type: 2048kB.
Mar 19 08:08:32 ip-172-31-40-205 nitro-enclaves-allocator[694]: Done.
Mar 19 08:08:32 ip-172-31-40-205 nitro-enclaves-allocator[694]: Auto-
generated the enclave CPU pool:
8,24,9,25,10,26,11,27,12,28,13,29,14,30,15,31.
Mar 19 08:08:32 ip-172-31-40-205 nitro-enclaves-allocator[694]:
Configuring the enclave CPU pool...
Mar 19 08:08:32 ip-172-31-40-205 nitro-enclaves-allocator[694]: Done.
Mar 19 08:08:32 ip-172-31-40-205 nitro-enclaves-allocator[694]:
Successfully allocated Nitro Enclaves resources: 28339 MiB, 16 CPUs
Mar 19 08:08:32 ip-172-31-40-205 systemd[1]: Finished Nitro Enclaves
Resource Allocator.
```

6. Check the **nitro-cli** version:

Plaintext | Copy

```
nitro-cli --version
```

Expected output:

Plaintext | Copy

```
Nitro CLI 1.4.0
```

Step 7 - Create EIF image from Duality TEE Enclave Image

This step builds the Enclave Image File (EIF), which contains the Enclave software and is used later to launch the Enclave.

1. Pull the Duality TEE Enclave image from Docker Hub, replace **<enclave-aws-fl:release-4.4.0>** with the current version:

Plaintext | Copy

```
docker login -u <username> -p <token>
docker pull dualitytech/enclave-aws-fl:release-4.4.0
```

2. Create a local file **duality.eif** from the Duality TEE enclave image, replace **<enclave-aws-fl:release-4.4.0>** with the current version (this might take several minutes):

Plaintext | Copy

```
nitro-cli build-enclave --docker-uri dualitytech/enclave-aws-fl:release-4.4.0 --output-file duality.eif
```

The command prints the Enclave platform configuration register (PCR) values.

> **NOTE: Save these values for the following steps.**

Expected output:

Plaintext | Copy

```
Enclave Image successfully created.
{
  "Measurements": {
    "HashAlgorithm": "Sha384 { ... }",
    "PCR0":
"43c5bb4185289da97117b0af61e8a4276a94c1d70dd5a2ce1e8b97224222cb3ba0f594a
39cc125f490ceb4c90bba2d10",
    "PCR1":
"0343b056cd8485ca7890ddd833476d78460aed2aa161548e4e26bedf321726696257d62
3e8805f3f605946b3d8b0c6aa",
    "PCR2":
"dc50791fa16b7b7e116552d0a3d641bbc3196c2759f8c04e9f8ebf84330249218b9882b
cd7cefb8b693d75150dddf123"
  }
}
```

Step 8 - Run Enclave

If the Duality Platform Node was fully installed and started before this Enclave setup, you must stop the Node service before running Step 8. You will be instructed to start it again after the Enclave is up and running.

Plaintext | Copy

```
cd install
./node.sh stop
```

The following step launches the Enclave process using the previously built EIF image.

1. Ensure you're in the **directory containing the** `duality.eif` **file** before running the command.

Plaintext | Copy

```
cd ..
```

2. Check the **cpu_count** and **memory_mib** values in the Enclave allocation service config file:

Plaintext | Copy

```
cat /etc/nitro_enclaves/allocator.yaml
```

Expected output (example):

Plaintext Copy

```
memory_mib: 28339
cpu_count: 16
```

> **NOTE:**
>
> AWS Nitro Enclaves use a portion of the instance's memory to emulate internal storage, as there is access to disk. Refer to Duality AI - Storage inside the Enclave for more information.

3. Run the Enclave, using the **cpu_count** and **memory_mib** values from the Enclave allocation service config file when executing the **run-enclave** command. For example: **--cpu-count 16 --memory 28339**

Production Deployment (v4.4)

Run the Enclave in Production Mode:

Plaintext Copy

```
sudo nitro-cli run-enclave --enclave-name duality --eif-path duality.eif
--enclave-cid 16 --cpu-count <cpu_count> --memory <memory_mib>
```

Debug Deployment (v4.4)

Run the Enclave in Debug Mode:

Plaintext Copy

```
sudo nitro-cli run-enclave --enclave-name duality --eif-path duality.eif
--enclave-cid 16 --cpu-count <cpu_count> --memory <memory_mib> --debug-
mode
```

The command prints the Enclave info:

Plaintext Copy

```
Started enclave with enclave-cid: 16, memory: 28339 MiB, cpu-ids: [8,
24, 9, 25, 10, 26, 11, 27, 12, 28, 13, 29, 14, 30, 15, 31]
{
  "EnclaveName": "duality",
  "EnclaveID": "i-0b6003e3657e22c3f-enc195a8bb3ecd8644",
  "ProcessID": 2183,
  "EnclaveCID": 16,
  "NumberOfCPUs": 16,
  "CPUIDs": [8, 24, 9, 25, 10, 26, 11, 27, 12, 28, 13, 29, 14, 30, 15,
31],
  "MemoryMiB": 36840
}
```

If the Node was already installed before the Enclave setup, start the Node service. Otherwise, proceed to Step 9.

Plaintext | Copy

```
cd install
./node.sh start
```

Step 9 - Install and Activate Proxy Service (AWS)

> **NOTE:** Upon instance restart, run Step 8 - Run Enclave. Then Restart the Proxy Service and verify the output.

The following step must be performed on the Compute Node. In this step, you will configure and start the proxy, which is required for Enclave-Compute networking. Instructions correspond to the README file located in *tee_utils*.

Navigate to the *tee_utils* folder within the *install* directory. This folder contains all utilities required for setting up the FL port forwarder.

Plaintext | Copy

```
cd install/tee_utils
```

1. **Install socat** on the host machine:

Plaintext | Copy

```
sudo apt install -y socat
```

2. **Copy the systemd** service file:

Plaintext Copy

```
sudo cp duality-enclave-proxy.service /etc/systemd/system/
```

3. **Edit the relevant cloud-vendor-specific script** and update the required address variables:

Find the **DOCKER_BRIDGE_IP** variable by running:

Plaintext Copy

```
ip -4 a s docker0 | grep inet | awk '{print $2}' | cut -d'/' -f1
```

Update the **DOCKER_BRIDGE_IP** variable (for AWS):

Plaintext Copy

```
sudo nano aws_duality_enclave_proxy.sh
```

4. **Copy and rename the updated script** to the system directory as duality_enclave_proxy.sh:

Plaintext Copy

```
sudo cp aws_duality_enclave_proxy.sh
/etc/systemd/system/duality_enclave_proxy.sh
```

Update execution permissions:

Plaintext Copy

```
sudo chmod +x /etc/systemd/system/duality_enclave_proxy.sh
```

5. **Enable and start** the proxy service:

Plaintext Copy

```
sudo systemctl enable --now duality-enclave-proxy.service
```

Verify that the proxy service is running:

Plaintext Copy

```
sudo systemctl status duality-enclave-proxy.service
```

Expected output:

Plaintext Copy

```
● duality-enclave-proxy.service - Initialize the FL enclave proxy on
boot
     Loaded: loaded (/etc/systemd/system/duality-enclave-proxy.service;
enabled; vendor preset: enabled)
     Active: active (running) since Mon 2025-04-21 11:38:27 UTC; 12s ago
   Main PID: 22242 (bash)
      Tasks: 4 (limit: 75552)
     Memory: 4.0M
        CPU: 10ms
     CGroup: /system.slice/duality-enclave-proxy.service
             ├─22242 /bin/bash
/etc/systemd/system/duality_enclave_proxy.sh
             ├─22244 socat TCP-
LISTEN:6250,bind=172.17.0.1,reuseaddr,fork VSOCK-CONNECT:16:6250
             ├─22245 socat TCP-LISTEN:8082,bind=0.0.0.0,reuseaddr,fork
VSOCK-CONNECT:16:8082
             └─22246 socat TCP-LISTEN:8083,bind=0.0.0.0,reuseaddr,fork
VSOCK-CONNECT:16:8083
Apr 21 11:38:27 ip-172-31-39-104 systemd[1]: Started Initialize the FL
enclave proxy on boot.
Apr 21 11:38:27 ip-172-31-39-104 bash[22242]: Proxy started
```

Platform Configuration - JSON Templates

NOTE:

The JSON file including the specific values and needs to be shared with the relevant participants in order to register

them as resources within the Duality Platform UI.
Proceed to the Resources (KMS & Enclaves) page for instructions on how to register the Enclave and KMS JSON within the platform.

Replace any placeholders within the JSON templates below:

- For **access_point**: Use the IP address of the Docker bridge (docker0 interface) on the host machine.
- For **measurements**: Use the **PCR** values that were shared.

Enclave - AWS

Detect the **DOCKER_BRIDGE_IP** address by running the following command:

Plaintext | Copy

```
ip -4 a s docker0 | grep inet | awk '{print $2}'
```

The command prints the IP address of the 'docker0' interface on the parent. For example, 172.17.0.1/16.

Production Mode:

JSON | Copy

```
{
    "access_point" : {
        "host" : "172.17.0.1",
        "port" : 6250
    },
    "measurements" : {
   "PCR0":
"a80e895883a3e152cd30c817523a0f18a98350bb3c98ba12c127227a2af89aca0b094aa
78fb70df4c688f43ad3fa6ded",
   "PCR1":
"0343b056cd8485ca7890ddd833476d78460aed2aa161548e4e26bedf321726696257d62
3e8805f3f605946b3d8b0c6aa",
   "PCR2":
"1ec9b6763fb9d54829b6f259605d4994176d2136fc3acde362a0c8d69244389de16bc8f
432a536d961fd36732ab16e7d"
    }
}
```


Debug Mode:

JSON | Copy


```
{
    "access_point" : {
        "host" : "172.17.0.1",
        "port" : 6250
    },
    "measurements" : {
   "PCR0":
"0000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000"
    }
}
```


KMS - AWS

Production Mode:

JSON | Copy

```
{
  "attestation": {
    "aud": "duality_kms",
    "iss": "AWS-NITRO-DETECTED",
    "pcrs": {
      "PCR0":
"a80e895883a3e152cd30c817523a0f18a98350bb3c98ba12c127227a2af89aca0b094aa
78fb70df4c688f43ad3fa6ded",
      "PCR1":
"0343b056cd8485ca7890ddd833476d78460aed2aa161548e4e26bedf321726696257d62
3e8805f3f605946b3d8b0c6aa",
      "PCR2":
"1ec9b6763fb9d54829b6f259605d4994176d2136fc3acde362a0c8d69244389de16bc8f
432a536d961fd36732ab16e7d"
    }
  }
}
```


Debug Mode:

JSON | Copy


```
{
  "attestation": {
    "aud": "duality_kms",
    "iss": "AWS-NITRO-DETECTED",
    "pcrs": {
      "PCR0":
"000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000"
    }
  }
}
```


Attestation for Offline Environments

If your environment does **not have internet access** and cannot reach the public attestation verification service, you must include the `signature_validation_key` field in the configuration. This enables verification of the attestation

must include the `signature_validation_key` field in the configuration. This enables verification of the attestation report locally, without relying on an external service.

JSON Copy

```
{
  "attestation": {
    "aud": "duality_kms",
    "iss": "AWS-NITRO-DETECTED",
    "pcrs": {
      "PCR0":
"a80e895883a3e152cd30c817523a0f18a98350bb3c98ba12c127227a2af89aca0b094aa
78fb70df4c688f43ad3fa6ded",
      "PCR1":
"0343b056cd8485ca7890ddd833476d78460aed2aa161548e4e26bedf321726696257d62
3e8805f3f605946b3d8b0c6aa",
      "PCR2":
"1ec9b6763fb9d54829b6f259605d4994176d2136fc3acde362a0c8d69244389de16bc8f
432a536d961fd36732ab16e7d"
    }
  },
    "signature_validation_key": "-----BEGIN CERTIFICATE-----
MIICETCCAZagAwIBAgIRAPkxdWgbkK/hHUbMtOTn+FYwCgYIKoZIzj0EAwMwSTELMAkGA1UE
BhMCVVMxDzANBgNVBAoMBkFtYXpvbjEMMAoGA1UECwwDQVdTMRswGQYDVQQDDBJhd3Mubml0
cm8tZW5jbGF2ZXMwHhcNMTkxMDI4MTMyODA1WhcNNDkxMDI4MTQyODA1WjBJMQswCQYDVQQG
EwJVUzEPMA0GA1UECgwGQW1hem9uMQwwCgYDVQQLDANBV1MxGzAZBgNVBAMMEmF3cy5uaXRy
by1lbmNsYXZlczB2MBAGByqGSM49AgEGBSuBBAAiA2IABPwCVOumCMHzaHDimtqQvkY4MpJz
bolL//Zy2YlES1BR5TSksfbb48C8WBoyt7F2Bw7eEtaaP+ohG2bnUs990d0JX28TcPQXCEPZ
3BABIeTPYwEoCWZEh8l5YoQwTcU/9KNCMEAwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQU
kCW1DdkFR+eWw5b6cp3PmanfS5YwDgYDVR0PAQH/BAQDAgGGMAoGCCqGSM49BAMDA2kAMGYC
MQCjfy+Rocm9Xue4YnwWmNJVA44fA0P5W2OpYow9OYCVRaEevL8uO1XYru5xtMPWrfMCMQCi
85sWBbJwKKXdS6BptQFuZbT73o/gBh1qUxl/nNr12UO8Yfwr6wPLb+6NIwLz3/Y=-----END
CERTIFICATE-----"
}
```

NOTE: When adding the signature_validation_key from a PEM file (e.g., -----BEGIN CERTIFICATE-----) into a JSON configuration, do not paste it as a raw multi-line string. This will result in an invalid JSON error during parsing.

Troubleshooting

Restarting the Enclave (and the Proxy Service)

After restarting the virtual machine, ensure the steps in the following order are performed to bring up all components correctly:

1. Start the Enclave: Step 8 - Run Enclave.
2. Restart the Proxy Service: Restart the Proxy Service and verify the output.

Enclave Status (and view PCRs)

Plaintext	Copy

```
sudo nitro-cli describe-enclaves
```

Expected output:

Plaintext	Copy

```
 {
    "EnclaveName": "duality",
    "EnclaveID": "i-0044f5da4239653ea-enc196575169b0961a",
    "ProcessID": 15317,
    "EnclaveCID": 16,
    "NumberOfCPUs": 16,
    "CPUIDs": [8, 24, 9, 25, 10, 26, 11, 27, 12, 28, 13, 29, 14, 30, 15,
31],
    "MemoryMiB": 28340,
    "State": "RUNNING",
    "Flags": "DEBUG_MODE",
    "Measurements": {
      "HashAlgorithm": "Sha384 { ... }",
      "PCR0":
"8ca96fd775dda00e75cf5ca577bde975db780cd4e75a3b161f2216c37adaf130a8764eb
620316d72da5a80e7024e81e2",
      "PCR1":
"0343b056cd8485ca7890ddd833476d78460aed2aa161548e4e26bedf321726696257d62
3e8805f3f605946b3d8b0c6aa",
      "PCR2":
"4262ac37cbd54e23397ae9a67fbd9294772bbb518825fe4ab5871cd718eed1eebfc94c5
28ed0890740e361ab91fcbefd"
    }
  }
```


Enclave Stop

Plaintext | Copy

```
sudo nitro-cli terminate-enclave --all
```

Expected output:

Plaintext | Copy

```
Successfully terminated enclave i-06d30a7d489f2b4e7-enc19661b9618b1095.
{
  "EnclaveName": "duality",
  "EnclaveID": "i-06d30a7d489f2b4e7-enc19661b9618b1095",
  "Terminated": true
}
```

Update Allocation Service Config File (allocator.yaml)

If the values require change, edit the file using:

Plaintext	Copy

```
sudo nano /etc/nitro_enclaves/allocator.yaml
```

Then restart the enclave service:

Plaintext	Copy

```
sudo systemctl restart nitro-enclaves-allocator.service
```

Remove/Replace Enclave Image File (EIF)

Stop the Enclave

Consider keeping a backup of the previous EIF file:

Plaintext	Copy

```
mv duality.eif duality-backup.eif
```

Remove the previous EIF file:

Plaintext	Copy

```
rm duality.eif
```

Expected output:

Plaintext	Copy

```
**TBC
```

Proxy Service Restart

If not all ports are assigned successfully, restart the proxy service:

Plaintext	Copy

```
sudo systemctl restart --now duality-enclave-proxy.service
```

Then, verify the proxy service status.

Proxy Service Status

Plaintext	Copy

```
systemctl status duality-enclave-proxy.service
```

Expected output:

Plaintext	Copy

```
● duality-enclave-proxy.service - Initialize the FL enclave proxy on
boot
     Loaded: loaded (/etc/systemd/system/duality-enclave-proxy.service;
enabled; vendor preset: enabled)
     Active: active (running) since Mon 2025-04-21 07:55:20 UTC; 21s ago
   Main PID: 21573 (bash)
      Tasks: 4 (limit: 75552)
     Memory: 2.9M
        CPU: 8ms
     CGroup: /system.slice/duality-enclave-proxy.service
             ├─21573 /bin/bash
/etc/systemd/system/duality_enclave_proxy.sh
             ├─21574 socat TCP-
LISTEN:6250,bind=172.17.0.1,reuseaddr,fork VSOCK-CONNECT:16:6250
             ├─21575 socat TCP-LISTEN:8082,bind=0.0.0.0,reuseaddr,fork
VSOCK-CONNECT:16:8082
             └─21576 socat TCP-LISTEN:8083,bind=0.0.0.0,reuseaddr,fork
VSOCK-CONNECT:16:8083
Apr 21 07:55:20 ip-172-31-43-248 systemd[1]: Started Initialize the FL
enclave proxy on boot.
Apr 21 07:55:20 ip-172-31-43-248 bash[21573]: Proxy started
```

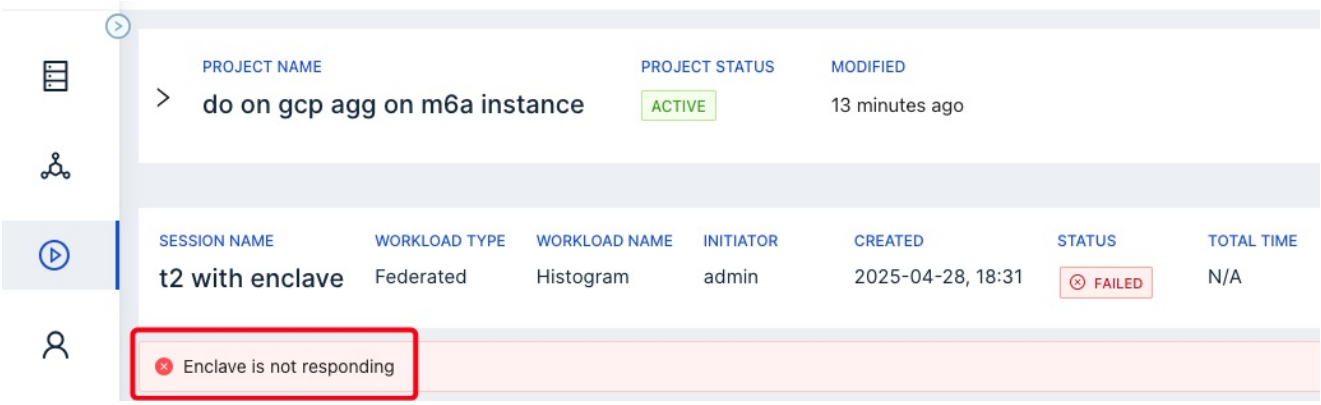
Verify the following message does not appear:

Plaintext | Copy

```
socat[722] E bind(5, {AF=2 172.17.0.1:6250}, 16): Cannot assign
requested address
```

Common Errors

Enclave is not responding



Solution: Perform the Enclave Restart Process.

Install - GCP Enclave

The following topic explains how to deploy a Trusted Execution Environment (TEE) on the Google Cloud Platform (GCP). TEEs enhance security by isolating sensitive computations, ensuring that data remains confidential even during processing.

This step-by-step guide assumes that you are familiar with GCP.

Prerequisites

- **Network Access:** Port TCP/6250 must be open for inbound traffic from the Compute Node to the Enclave VM to enable secure data transfer during FL execution in a TEE environment.
- **GCP Project**: The Compute and Enclave should reside within the same GCP project, share the same VPC, and ideally operate within the same private network for optimal performance and security.

Installation & Configuration Flow

The following sections outline the steps required to install and configure a TEE based on GCP Confidential Space. Steps 1–5 guide you through setting up and configuring the necessary resources, while Step 6 covers launching and installing the Enclave.

After completing the installation, you can retrieve the **google_service_account** and **image_digest**, which are required to configure the Enclave within the Duality Platform UI.

1. **Install gcloud CLI:**
    - The gcloud command-line tool is needed to interact with GCP services. Follow the GCP Installation Guide (https://cloud.google.com/sdk/docs/install) to set it up.
2. **Admin Permissions:**
    - Ensure you have admin-level access to your GCP project to configure services and deploy resources.
3. **Project and Network Setup:**
    - Deploy the TEE in the same GCP project as your Compute node.
    - Use a private network for secure communication between the TEE and the Compute node.
4. **Compute Node Installed:**
    - This topic assumes the Compute node was already deployed and installed.
5. **Run Commands from the Compute Instance (recommended):**
    - It is recommended (but not mandatory) to execute the deployment commands from a compute instance in the same project.

6. **Install Docker**:

    Ensure Docker is installed and configured. Follow the Docker Installation Guide to install Docker, and add your

user to the Docker group to run commands without Sudo Installation Steps

All commands in this section should be run in the gcloud CLI, which can be accessed through your terminal or command prompt after installation.

Step 1 - Enable Required Services

The services below must be enabled on the GCP project level to enable the TEE functionality.

Run these commands in the gcloud CLI:

Plaintext	Copy

```
gcloud services enable confidentialcomputing.googleapis.com
gcloud services enable artifactregistry.googleapis.com
```

Step 2 - Create a Service Account for TEE

A service account is a special type of Google Cloud account used by applications or virtual machines to interact securely with GCP services. It acts as an identity for your TEE operations and enables authentication and authorization without relying on user credentials.

Run these commands in the gcloud CLI, replacing the placeholders with your project-specific details.

- **Purpose:** This account will authenticate and execute actions securely.
- **Instructions:**

1. **Set environment variables** for `<PROJECT_ID>` and `<SERVICE_ACCOUNT_NAME>` : These will be used in subsequent commands
2. **Replace the placeholders** `<DESCRIPTION>` and `<DISPLAY_NAME>` directly in the command below.

Parameter requirements:

- `<SERVICE_ACCOUNT_NAME>` must be a valid GCP service account name.
- `<PROJECT_ID>` must be unique across all GCP projects.
- Both values must be 6–30 characters, start with a lowercase letter, and contain only lowercase letters, numbers, and dashes (no spaces or uppercase letters).

Set Environment Variables:

Plaintext	Copy

```
export PROJECT_ID=<PROJECT_ID>
export SERVICE_ACCOUNT_NAME=<SERVICE_ACCOUNT_NAME>
```

Create the Service Account:

Plaintext Copy

```
gcloud iam service-accounts create ${SERVICE_ACCOUNT_NAME} \
  --description="<DESCRIPTION>" --display-name="<DISPLAY_NAME>"
```

Expected Output:

Plaintext Copy

```
Created service account [<SERVICE_ACCOUNT_NAME>].
```

Set the Service Account Identifier:

Plaintext Copy

```
export
service_account="${SERVICE_ACCOUNT_NAME}@${PROJECT_ID}.iam.gserviceaccou
nt.com"
```

Step 3 - Add Permissions to the Service Account

Add roles to the service account to ensure it has the necessary permissions.

- **Purpose:** These roles allow the account to interact with confidential workloads, log activities, and manage repositories.
- **Instructions:** Run the following.

Plaintext Copy

```
gcloud projects add-iam-policy-binding ${PROJECT_ID} \
  --member="serviceAccount:${service_account}" --
role="roles/iam.securityAdmin"
gcloud projects add-iam-policy-binding ${PROJECT_ID} \
  --member="serviceAccount:${service_account}" --
role="roles/confidentialcomputing.workloadUser"
gcloud projects add-iam-policy-binding ${PROJECT_ID} \
  --member="serviceAccount:${service_account}" --
role="roles/logging.logWriter"
gcloud projects add-iam-policy-binding ${PROJECT_ID} \
  --member="serviceAccount:${service_account}" --
role="roles/artifactregistry.repoAdmin"
gcloud projects add-iam-policy-binding ${PROJECT_ID} \
  --member="serviceAccount:${service_account}" --
role="roles/artifactregistry.reader"
gcloud projects add-iam-policy-binding ${PROJECT_ID} \
  --member="serviceAccount:${service_account}" --
role="roles/artifactregistry.writer"
```

Verify successful policy binding by running the command:

Plaintext | Copy

```
gcloud projects get-iam-policy ${PROJECT_ID} \
    --flatten="bindings[].members" \
    --filter="bindings.members:serviceAccount:${service_account}" \
    --format='table(bindings.role, bindings.members)'
```

Expected output:

Plaintext | Copy

```
ROLE: roles/artifactregistry.reader
MEMBERS: serviceAccount:
<SERVICE_ACCOUNT_NAME>@<PROJECT_ID>.iam.gserviceaccount.com

ROLE: roles/artifactregistry.repoAdmin
MEMBERS: serviceAccount:
<SERVICE_ACCOUNT_NAME>@<PROJECT_ID>.iam.gserviceaccount.com

ROLE: roles/artifactregistry.writer
MEMBERS: serviceAccount:
<SERVICE_ACCOUNT_NAME>@<PROJECT_ID>.iam.gserviceaccount.com

ROLE: roles/confidentialcomputing.workloadUser
MEMBERS: serviceAccount:
<SERVICE_ACCOUNT_NAME>@<PROJECT_ID>.iam.gserviceaccount.com

ROLE: roles/iam.securityAdmin
MEMBERS: serviceAccount:
<SERVICE_ACCOUNT_NAME>@<PROJECT_ID>.iam.gserviceaccount.com

ROLE: roles/logging.logWriter
MEMBERS: serviceAccount:
<SERVICE_ACCOUNT_NAME>@<PROJECT_ID>.iam.gserviceaccount.com
```

Step 4 - Create Artifacts Repository for TEE Container Image

An artifact repository is a storage location in GCP for container images, binaries, and other artifacts. It is essential to securely store and manage the TEE container image that will be deployed in the GCP environment.

- **Purpose:** This repository will store the TEE container image; this must be a valid GCP Artifacts Repository name.
- **Instructions**: Run the commands below in the gcloud CLI to create and configure your artifact repository.

> **NOTE**: A valid repository name must use only lowercase letters, numbers, and hyphens.

Set Repository Variable:

Plaintext	Copy

```
export REPOSITORY_NAME=<REPOSITORY_NAME>
```

Create Artifact Repository:

Plaintext Copy

```
gcloud artifacts repositories create ${REPOSITORY_NAME} --repository-format=docker --location=us
```

Expected output:

Plaintext Copy

```
Created repository [<REPOSITORY_NAME>].
```

Configure Docker:

Plaintext Copy

```
gcloud auth configure-docker us-docker.pkg.dev
```

For other locations, see the following documentation:
https://cloud.google.com/sdk/gcloud/reference/artifacts/repositories/create

Step 5 - Upload Duality TEE Image to the Artifacts Repository

The TEE image is a unique image provided by Duality that contains all libraries and other necessary resources required for running a computation inside the TEE.

- **Purpose:** Upload the TEE container image to the artifact repository.
- **Instructions**: Use the gcloud CLI to **pull, tag, and push** the TEE container image to the artifact repository.

> **NOTE:** You will need to use the token provided by Duality to login to docker before you run the PULL command.

> **NOTE:** Be sure to choose the the relevant image for the TEE from Docker Hub. e.g., for version 4.4 use **<dualitytech/enclave-gcp-fl:release-4.4.0>**. Contact Duality support for additional help if needed.

Set TEE Image and Repository Variables:

Plaintext Copy

```
export duality_image=dualitytech/enclave-gcp-fl:release-4.4.0
export tee_image=us-
docker.pkg.dev/${PROJECT_ID}/${REPOSITORY_NAME}/enclave-gcp-fl:release-
4.4.0
```

Pull, tag, and push the TEE Container Image:

Plaintext | Copy

```
docker pull ${duality_image}
docker tag ${duality_image} ${tee_image}
docker push ${tee_image}
```

Expected output:

Plaintext | Copy

```
release-4.4.0: digest: sha256:<image_digest> size: 4096
```

NOTE: If another location was chosen for the Artifacts Repository in the previous step, make sure to use the corresponding link for the **${tee_image}** variable.

Step 6 - Deploy Confidential VM

After completing all pre-configurations, the final step is to deploy a confidential VM with the following parameters:

- **NAME:** VM instance name.
- **ZONE:** GCP zone for deployment.
- **MACHINE_TYPE:** Recommended: n2d-standard-16 (16vCPUs, 64 GB RAM).
- **DISK_SIZE:** Recommended: 50 GB.

NOTE: If you are only deploying the VM and Steps 1–5 were completed previously (e.g., as part of a separate setup process), ensure that the **tee_image** variable is manually updated with the **full path** to the TEE container image in your artifact repository.

Production Deployment (v4.4)

Plaintext | Copy

```
gcloud compute instances create <NAME> \
    --confidential-compute --shielded-secure-boot \
    --maintenance-policy=TERMINATE --scopes=cloud-platform \
    --zone=<ZONE> --machine-type=<MACHINE_TYPE> \
    --boot-disk-size=<DISK_SIZE>GB \
    --image-project=confidential-space-images \
    --image-family=confidential-space \
    --service-account=${service_account} \
    --metadata="^~^tee-image-reference=${tee_image}"
```

Debug Deployment (v4.4)

Plaintext | Copy

```
gcloud compute instances create <NAME> \
    --confidential-compute --shielded-secure-boot \
    --maintenance-policy=TERMINATE --scopes=cloud-platform \
    --zone=<ZONE> --machine-type=<MACHINE_TYPE> \
    --boot-disk-size=<DISK_SIZE>GB \
    --image-project=confidential-space-images \
    --image-family=confidential-space-debug \
    --service-account=${service_account} \
    --metadata="^~^tee-image-reference=${tee_image}~tee-container-log-redirect=true"
```

Note: The new CLI works from GCP management console gcloud (519.0.0), but not from the ubuntu latest available gcloud (475.0.0).

Expected output:

Plaintext | Copy

```
NAME: <NAME>
ZONE: <ZONE>
MACHINE_TYPE: <MACHINE_TYPE>
PREEMPTIBLE:
INTERNAL_IP: <INTERNAL_IP>
EXTERNAL_IP: <EXTERNAL_IP>
STATUS: RUNNING
```

Step 7 - Install and Activate Proxy Service (GCP)

The following step must be performed on the Compute Node. In this step, you will configure and start the proxy, which is required for Enclave-Compute networking. Instructions correspond to the README file located in *tee_utils*.

Navigate to the *tee_utils* folder within the *install* directory. This folder contains all utilities required for setting up the FL port forwarder.

Plaintext Copy

```
cd install/tee_utils
```

1. **Install socat** on the host machine:

Plaintext Copy

```
sudo apt install -y socat
```

2. **Copy the systemd** service file:

Plaintext Copy

```
sudo cp duality-enclave-proxy.service /etc/systemd/system/
```

3. **Edit the relevant cloud-vendor-specific script** and update the required address variables:

Update the **ENCLAVE_ADDR** variable (for GCP) - to <INTERNAL_ENCLAVE_IP> from which it will communicate with the Compute Node.

Plaintext Copy

```
sudo nano gcp_duality_enclave_proxy.sh
```

4. **Copy and rename the updated script** to the system directory as duality_enclave_proxy.sh:

Plaintext Copy

```
sudo cp gcp_duality_enclave_proxy.sh
/etc/systemd/system/duality_enclave_proxy.sh
```

Update execution permissions:

Plaintext Copy

```
sudo chmod +x /etc/systemd/system/duality_enclave_proxy.sh
```

5. **Enable and start** the proxy service:

Plaintext Copy

```
sudo systemctl enable --now duality-enclave-proxy.service
```

Verify the proxy service status:

Plaintext Copy

```
sudo systemctl status duality-enclave-proxy.service
```

Expected output:

Plaintext Copy

```
● duality-enclave-proxy.service - Initialize the FL enclave proxy on
boot
     Loaded: loaded (/etc/systemd/system/duality-enclave-proxy.service;
enabled; vendor preset: enabled)
     Active: active (running) since Fri 2025-06-20 10:01:25 UTC; 25s ago
   Main PID: 38643 (bash)
      Tasks: 3 (limit: 38503)
     Memory: 2.0M
        CPU: 13ms
     CGroup: /system.slice/duality-enclave-proxy.service
             ├─38643 /bin/bash
/etc/systemd/system/duality_enclave_proxy.sh
             ├─38644 socat TCP-LISTEN:8082,bind=0.0.0.0,reuseaddr,fork
TCP:10.128.0.2:8082
             └─38645 socat TCP-LISTEN:8083,bind=0.0.0.0,reuseaddr,fork
TCP:10.128.0.2:8083
Jun 20 10:01:25 nci-agg-43-mirror systemd[1]: Started Initialize the FL
enclave proxy on boot.
Jun 20 10:01:25 nci-agg-43-mirror bash[38643]: Proxy started
```

> **NOTE:** If you change the enclave instance IP, modify the **ENCLAVE_ADDR** variable (for GCP) in **/etc/systemd/system/duality_enclave_proxy.sh** and Restart the Proxy Service.

Retrieve the Google Service Account and Image Digest

Verify that the TEE instance is running as expected and retrieve the **google_service_account** and **image_digest**, which are required to configure the enclave within the Duality Platform.

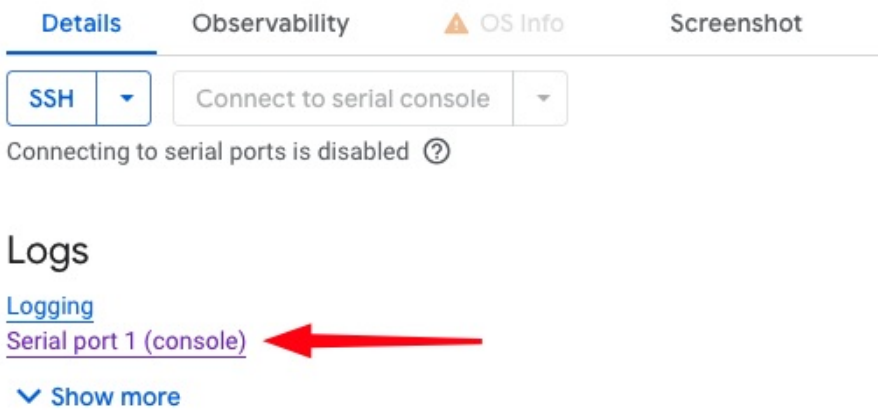
> **NOTE**: These values will be shared with the participants for the Enclave and KMS JSON configuration within the Duality Platform UI (TEE attestation).

There are two ways to run the validation: using the Google Cloud Console or using the gcloud CLI

Using the Google Cloud Console

1. Navigate to the **VM instances** page in the console. To access the Google Cloud Console, log in to your GCP account at console.cloud.google.com and navigate to the **Compute Engine** section.

2. Select the TEE instance.
3. Scroll down to the **Logs** section.
4. Click **Serial port 1 (console)** to view the instance logs output.



Using the gcloud CLI

Run the following command to fetch the serial port logs:

Plaintext	Copy

```
gcloud compute instances get-serial-port-output <NAME> --zone <ZONE>
```

Logs Validation

Verify that the following output, which indicates that the server started with port forwarding 6250, appears:

Plaintext	Copy

```
INFO: ServerTransport: HTTPS Server Started on 0.0.0.0:6250 (AF:2)
```

Verify that the following output, which indicates that the TEE instance launched the Duality container, appears:

Plaintext	Copy

```
2024/09/15 13:54:41 Launch Spec: {ImageRef:us-docker.pkg.dev/tee-demo-
401413/tee-demo-401413-repo/enclave-gcp-fl:release-4.3.0
SignedImageRepos:[] RestartPolicy:Never Cmd:[] Envs:
[{Name:COMPUTE_SERVER_ADDR Value:10.128.0.4} {Name:COMPUTE_SERVER_PORT
Value:5000}] AttestationServiceAddr: ImpersonateServiceAccounts:[]
ProjectID:tee-demo-401413 Region:us-west1 Hardened:false
LogRedirect:true Experiments:{EnableTestFeatureForImage:false
EnableSignedContainerImage:false}}
2024/09/15 13:55:45 Operator Input Image Ref    : us-docker.pkg.dev/tee-
demo-401413/tee-demo-401413-repo/enclave-gcp-fl:release-4.3.0
2024/09/15 13:55:45 Image Digest                :
sha256:a1728550c03e88407972038bb457f6a91dcd15675c400c813aad89c60dcb4405
2024/09/15 13:55:45 Operator Override Env Vars :
[COMPUTE_SERVER_ADDR=10.128.0.4 COMPUTE_SERVER_PORT=6250]
2024/09/15 13:55:45 Operator Override Cmd       : []
2024/09/15 13:55:45 Exposed Ports:              : map[]
2024/09/15 13:55:45 Image Labels                :
map[org.opencontainers.image.ref.name:ubuntu
org.opencontainers.image.version:20.04
tee.launch_policy.allow_env_override:COMPUTE_SERVER_ADDR,COMPUTE_SERVER_
PORT,LOG_LEVEL]
2024/09/15 13:55:45 Image ID                    :
sha256:2749838b9dd4f5c7629860bfab285fcdd6468577de754767aec8bf5e20d93f73
2024/09/15 13:55:45 Image Annotations           : map[]
2024/09/15 13:55:45 refreshing attestation verifier OIDC token
```

Verify that the following output, which contains the attestation report, appears:

Plaintext | Copy

```
2024/09/15 13:55:45 refreshing attestation verifier OIDC token
2024/09/15 13:55:46 {
  "aud": "https://sts.googleapis.com",
  "dbgstat": "enabled",
  "eat_profile": "https://cloud.google.com/confidential-
computing/confidential-space/docs/reference/token-claims",
  "exp": 1726412146,
  "google_service_accounts": [ "tee-demo-srv-acc@tee-demo-
401413.iam.gserviceaccount.com" ],

  "hwmodel": "GCP_AMD_SEV",
  "iat": 1726408546,
```

```
  "iss": "https://confidentialcomputing.googleapis.com",
  "nbf": 1726408546,
  "oemid": 11129,
  "secboot": true,
  "sub": "https://www.googleapis.com/compute/v1/projects/tee-demo-
401413/zones/us-west1-b/instances/confidential1",
  "submods": {
    "confidential_space": {},
    "container": {
      "args": [ "/duality/tee_duality_gcp/enclave_gcp/run.sh" ],
      "env": { < environment variables > },
      "env_override": { "COMPUTE_SERVER_ADDR": "10.128.0.4",
"COMPUTE_SERVER_PORT": "6250" }
      "image_digest":
"sha256:a1728550c03e88407972038bb457f6a91dcd15675c400c813aad89c60dcb4405
",
      "image_id":
"sha256:2749838b9dd4f5c7629860bfab285fcdd6468577de754767aec8bf5e20d93f73
",
      "image_reference": "us-docker.pkg.dev/tee-demo-401413/tee-demo-
401413-repo/enclave-gcp-fl:release-4.3.0",
      "restart_policy": "Never"
    },
    "gce": {
      "instance_id": "6961199820166888986",
      "instance_name": "confidential1",
      "project_id": "tee-demo-401413",
      "project_number": "815434821327",
      "zone": "us-west1-b"
    }
  },
  "swname": "CONFIDENTIAL_SPACE",
  "swversion": [ "230901" ]
}
```


Platform Configuration - JSON Templates

NOTE:

The JSON file including the specific values and needs to be shared with the relevant participants in order to register them as resources within the Duality Platform UI.

> Proceed to the Resources (KMS & Enclaves) page for instructions on how to register the Enclave and KMS JSON within the platform.

Replace any placeholders within the JSON templates below:

- For **access_point**: Use the IP address of the Enclave node.
- For **measurements**: Use the **google_service_account** and **image_digest** values that were shared.

Enclave - GCP

JSON | Copy

```
{
  "access_point": {
    "host": "<INTERNAL_ENCLAVE_IP>",
    "port": 6250
  },
  "measurements": {
    "google_service_account": "
<SERVICE_ACCOUNT_NAME@PROJECT_ID.iam.gserviceaccount.com>",
    "image_digest": "sha256:<image_digest>"
  }
}
```

KMS - GCP

Production Mode:

JSON | Copy

```
{
```

```
{
   "attestation":{
      "aud":"duality_kms",
      "iss":"https://confidentialcomputing.googleapis.com",
      "secboot":true,
      "hwmodel":"GCP_AMD_SEV",
      "swname":"CONFIDENTIAL_SPACE",
      "google_service_accounts":[
         "<SERVICE_ACCOUNT_NAME@PROJECT_ID.iam.gserviceaccount.com>"
      ],
      "submods":{
         "confidential_space":{
            "support_attributes":{
               "$has":"STABLE"
            }
         },
         "container":{
            "image_digest":"sha256:<image_digest>",
            "restart_policy":"Never"
         }
      }
   }
}
```

Debug Mode:

JSON | Copy

```
{
```

```
{
   "attestation":{
      "aud":"duality_kms",
      "iss":"https://confidentialcomputing.googleapis.com",
      "secboot":true,
      "hwmodel":"GCP_AMD_SEV",
      "swname":"CONFIDENTIAL_SPACE",
      "google_service_accounts":[
         "<SERVICE_ACCOUNT_NAME@<ROJECT_ID.iam.gserviceaccount.com>"
      ],
      "submods":{
         "container":{
            "image_digest":"sha256:<image_digest>",
            "restart_policy":"Never"
         }
      }
   }
}
```

Attestation for Offline Environments

If your environment does **not have internet access** and cannot reach the public attestation verification service, you must include the `signature_validation_key` field in the configuration. This enables verification of the attestation report locally, without relying on an external service.

JSON | Copy

```
{
```

```
{
   "attestation":{
      "aud":"duality_kms",
      "iss":"https://confidentialcomputing.googleapis.com",
      "secboot":true,
      "hwmodel":"GCP_AMD_SEV",
      "swname":"CONFIDENTIAL_SPACE",
      "google_service_accounts":[
         "<SERVICE_ACCOUNT_NAME@PROJECT_ID.iam.gserviceaccount.com>"
      ],
      "submods":{
         "confidential_space":{
            "support_attributes":{
               "$has":"STABLE"
            }
         },
         "container":{
            "image_digest":"sha256:<image_digest>",
            "restart_policy":"Never"
         }
      }
   },
    "signature_validation_key": "-----BEGIN CERTIFICATE-----
MIICETCCAZagAwIBAgIRAPkxdWgbkK/hHUbMtOTn+FYwCgYIKoZIzj0EAwMwSTELMAkGA1UE
BhMCVVMxDzANBgNVBAoMBkFtYXpvbjEMMAoGA1UECwwDQVdTMRswGQYDVQQDDBJhd3Mubml0
cm8tZW5jbGF2ZXMwHhcNMTkxMDI4MTMyODA1WhcNNDkxMDI4MTQyODA1WjBJMQswCQYDVQQG
EwJVUzEPMA0GA1UECgwGQW1hem9uMQwwCgYDVQQLDANBV1MxGzAZBgNVBAMMEmF3cy5uaXRy
by1lbmNsYXZlczB2MBAGByqGSM49AgEGBSuBBAAiA2IABPwCVOumCMHzaHDimtqQvkY4MpJz
bolL//Zy2YlES1BR5TSksfbb48C8WBoyt7F2Bw7eEtaaP+ohG2bnUs990d0JX28TcPQXCEPZ
3BABIeTPYwEoCWZEh8l5YoQwTcU/9KNCMEAwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQU
kCW1DdkFR+eWw5b6cp3PmanfS5YwDgYDVR0PAQH/BAQDAgGGMAoGCCqGSM49BAMDA2kAMGYC
MQCjfy+Rocm9Xue4YnwWmNJVA44fA0P5W2OpYow9OYCVRaEevL8uO1XYru5xtMPWrfMCMQCi
85sWBbJwKKXdS6BptQFuZbT73o/gBh1qUxl/nNr12UO8Yfwr6wPLb+6NIwLz3/Y=-----END
CERTIFICATE-----"
}
```


NOTE: When adding the signature_validation_key from a PEM file (e.g., -----BEGIN CERTIFICATE-----) into a JSON configuration, do not paste it as a raw multi-line string. This will result in an invalid JSON error during parsing.

Troubleshooting

Enclave Status

Plaintext | Copy

```
gcloud compute instances describe <INSTANCE_NAME> --zone=<ZONE>
```

Enclave Instance Change

If you change the enclave instance IP, modify the **ENCLAVE_ADDR** variable (for GCP) in /etc/systemd/system/duality_enclave_proxy.sh to match the new value and restart the proxy service.

Proxy Service Restart

If not all ports are assigned successfully, restart the proxy service:

Plaintext | Copy

```
sudo systemctl restart --now duality-enclave-proxy.service
```

Then, verify the proxy service status.

Proxy Service Status

Plaintext | Copy

```
systemctl status duality-enclave-proxy.service
```

Expected output:

Plaintext | Copy

```
● duality-enclave-proxy.service - Initialize the FL enclave proxy on
```

```
boot
     Loaded: loaded (/etc/systemd/system/duality-enclave-proxy.service;
enabled; vendor preset: enabled)
     Active: active (running) since Fri 2025-06-20 10:01:25 UTC; 25s ago
   Main PID: 38643 (bash)
      Tasks: 3 (limit: 38503)
     Memory: 2.0M
        CPU: 13ms
     CGroup: /system.slice/duality-enclave-proxy.service
             ├─38643 /bin/bash
/etc/systemd/system/duality_enclave_proxy.sh
             ├─38644 socat TCP-LISTEN:8082,bind=0.0.0.0,reuseaddr,fork
TCP:10.128.0.2:8082
             └─38645 socat TCP-LISTEN:8083,bind=0.0.0.0,reuseaddr,fork
TCP:10.128.0.2:8083
Jun 20 10:01:25 nci-agg-43-mirror systemd[1]: Started Initialize the FL
enclave proxy on boot.
Jun 20 10:01:25 nci-agg-43-mirror bash[38643]: Proxy started
```

Common Errors

Failed to pull image

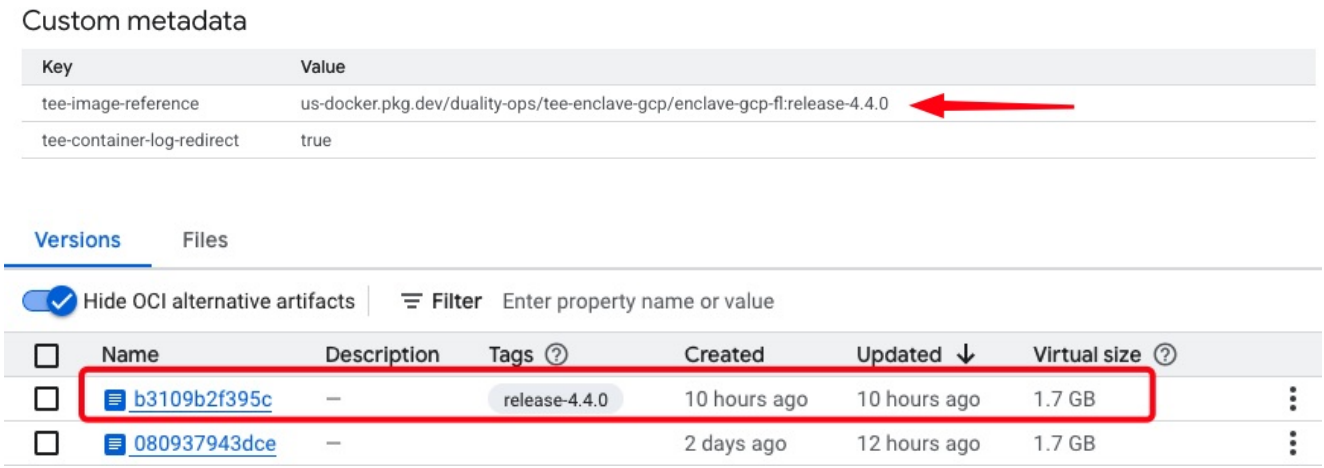
Plaintext	Copy

```
ERROR msg="cannot pull the image: failed to pull image with retries, the
last error is: failed to resolve reference"us-docker.pkg.dev/duality-
ops/tee-enclave-gcp/enclave-gcp-fl:release-4.4.0": us-
docker.pkg.dev/duality-ops/tee-enclave-gcp/enclave-gcp-fl:release-4.4.0:
not found"
```

```
time=2025-08-14T18:07:04.358Z level=INFO msg="Serial Console logger initialized"
time=2025-08-14T18:07:04.577Z level=INFO msg="Boot completed" duration_sec=49.81
time=2025-08-14T18:07:04.635Z level=INFO msg="TEE container launcher initiating" build_commit=9ed621c
time=2025-08-14T18:07:06.347Z level=INFO msg="Launch Spec: {Experiments:{EnableTestFeatureForImage:true EnableTempFSMount:
time=2025-08-14T18:07:06.457Z level=INFO msg="Updated TPM DA params: &{LockoutCounter:0 MaxTries:32 RecoveryTime:7200 Lock
```

```
time=2025-08-14T18:07:06.572Z level=INFO msg="Launch started" duration_sec=2.235448011
time=2025-08-14T18:07:08.648Z level=ERROR msg="cannot pull the image: failed to pull image with retries, the last error is
time=2025-08-14T18:07:08.737Z level=INFO msg="Workload completed" workload=us-docker.pkg.dev/duality-ops/tee-enclave-gcp/e
time=2025-08-14T18:07:08.813Z level=INFO msg="TEE container launcher exiting" exit_code=4 exit_msg="VM remains running"
```

Solution: Check the Artifacts Repository, package, and tags in the instance custom metadata and verify that the correct image was pushed (verify image_digest).



Upgrade/ Uninstall Duality Platform

Duality Platform Upgrade

> **NOTE:**
> Before commencing the upgrade process create a local backup of the Duality Platform.
> For any assistance, contact support@dualitytech.com

To pull a new patch for the Duality Platform (vX.X.X.xxxx):

1. Browse to the install folder:

```
cd install/
```

- For the Collaboration Manager:

```
./orchestrator.sh restart --pull
```

- For the Node:

```
./node.sh restart --pull
```

The following message indicates that the upgrade has been completed: *"Upgrade of Duality Platform completed"*

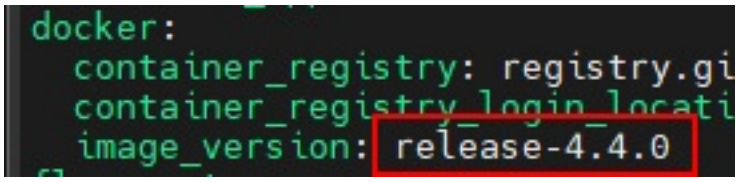
To upgrade the Duality Platform with a new version (vX.X.X):

1. Browse to the config folder:

```
cd install/config/
```

2. Set the `image_version` parameter in the **node_advanced_config.yml** file to the required value. See Advanced Configurations for additional information.

Plaintext	Copy
`nano node_advanced_config.yml`	



3. Save and exit
4. Follow the instructions above to pull the newest patch for this version

Enclave Upgrade

AWS Enclave

Remove/Replace the Enclave Image File (EIF)

Follow the steps in Install - AWS Enclave:

- Step 7 - **Create (new) EIF Image from Duality TEE Enclave Image**
- Step 8 - **Run (new) Enclave**
- **Proxy Service Restart**

> **NOTE:** Update resource JSON values (Enclave and KMS) in the UI.

GCP Enclave

Follow the steps in Install - GCP Enclave:

- Step 5 - **Upload (new) Duality TEE Image to the Artifacts Repository**
- Stop the instance
- Under Custom metadata, update the `tee-image-reference` value to match the new version

tee-container-log-redirect	true

- Start the instance
- Proxy Service Restart
- Retrieve the (new) Google Service Account and Image Digest

> **NOTE:** Update resource JSON values (Enclave and KMS) in the UI

Duality Platform Uninstall

> **NOTE:**
> The steps below remove ALL content from the instances, proceed with caution.
> If Duality Platform shares the instances with any additional software, contact support@dualitytech.com

To uninstall Duality Platform software:

1. Browse to the install folder:

Plaintext	Copy

```
cd install/
```

2. Uninstall the Duality Platform by running:

Plaintext	Copy

```
./uninstall.sh
```

   - The following message indicates that the uninstall has been completed: “Uninstall completed”

3. Browse to the home folder:

Plaintext	Copy

```
cd ~
```

4. Delete the install folder:

Plaintext	Copy

```
rm -rf install
```

NOTE:

If you plan to reinstall the platform after uninstallation, you must restart the installation process from the setup.sh script.

Resources (KMS & Enclave)

Resources (KMS & Enclave)

Before participants can associate a **KMS or Enclave** resource to a project, these resources must first be added (registered) in JSON format to the UI:

- **KMS:** Used to encrypt and decrypt assets during sessions.
- **Enclave:** Carries out secure computation inside a TEE.

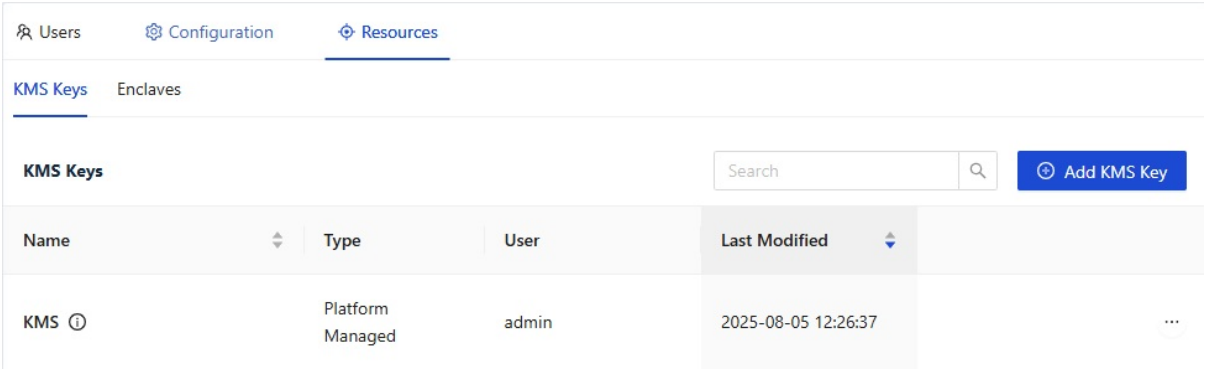
NOTE: The JSON includes information retrieved during the enclave installation process and varies according to the specific CSP.
View the required template for each specific CSP:

- AWS - JSON Templates
- GCP - JSON Templates

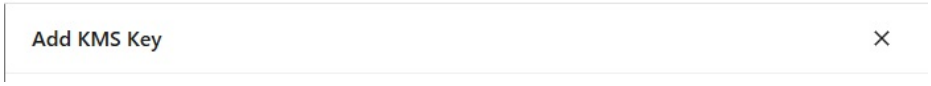
Add KMS Resource

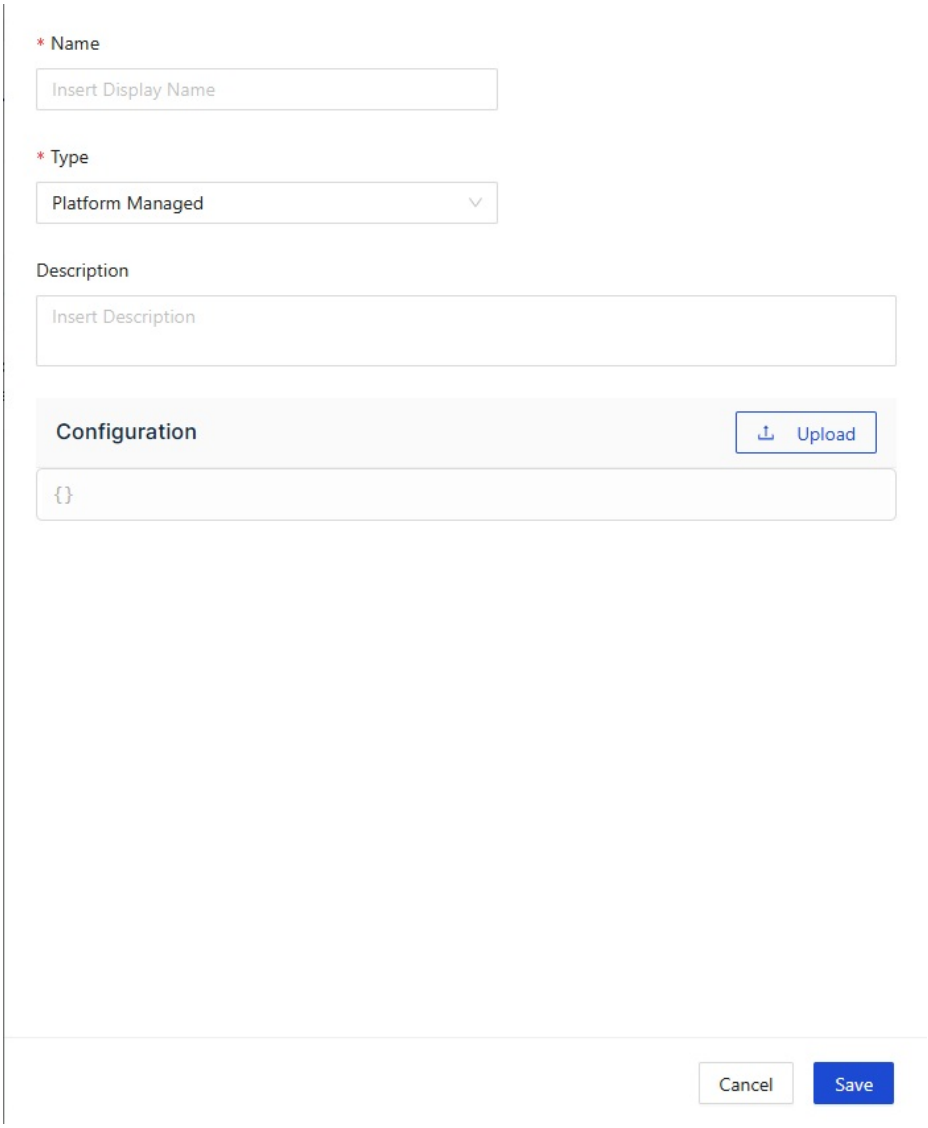
To add a KMS:

1. In **Admin**, select the **Resources** > **KMS Keys** tab.



2. Click **Add KMS Key**.



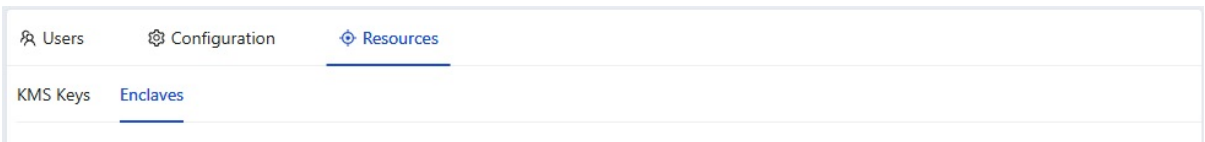


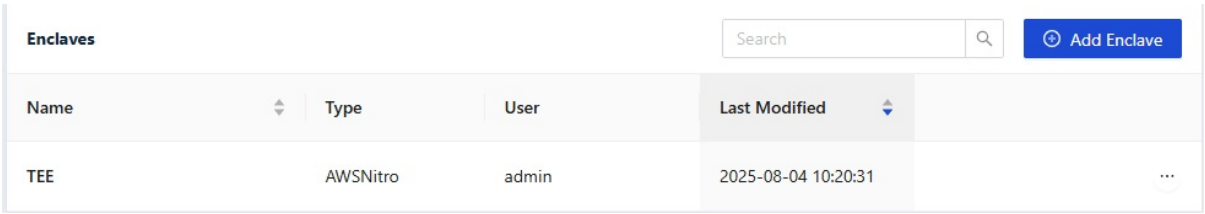
3. Fill in the following fields:
    - **Name** – A unique identifier.
    - **Type** – `Platform Managed`
    - **Description** – (Optional) Describe the key.
4. Click **Upload** to provide the KMS configuration in JSON format.
5. Click **Save** to complete registration.

Add Enclave Resource

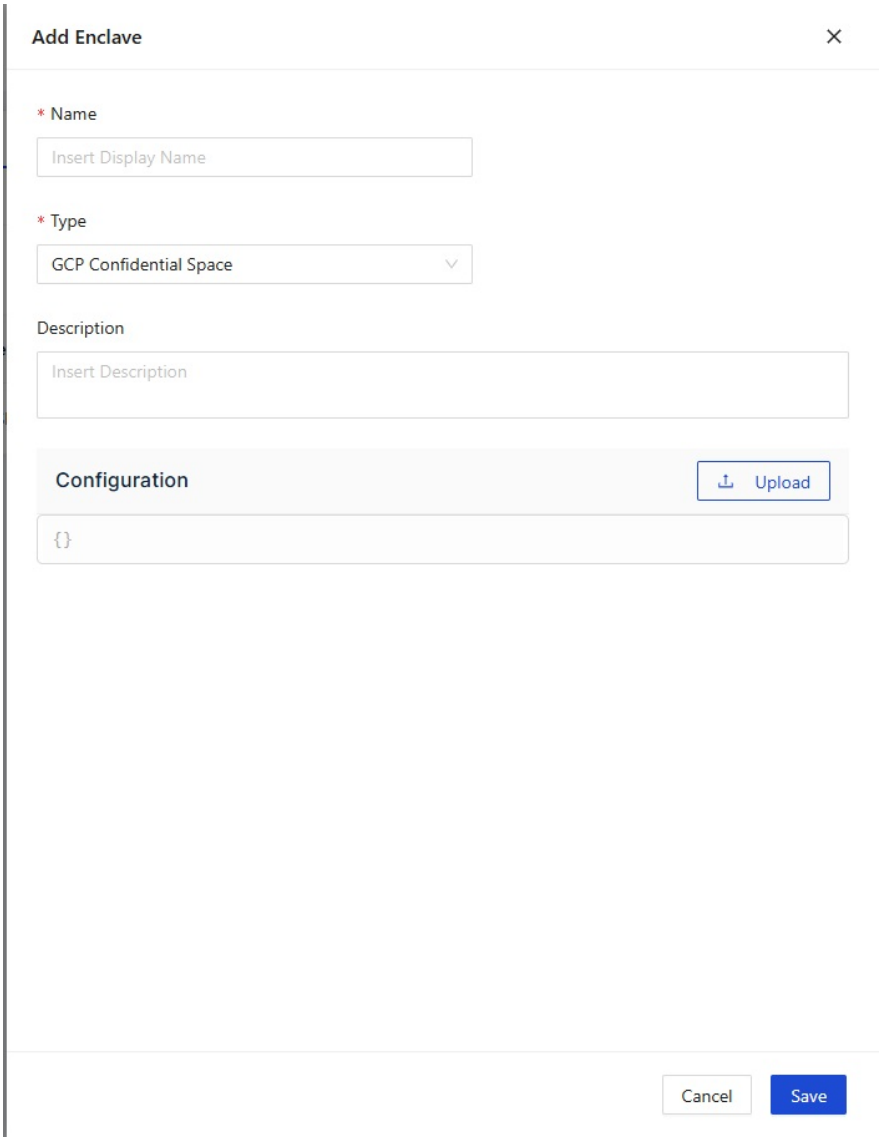
To add an Enclave:

1. In **Admin**, select the **Resources** > **Enclave** tab.





2. Click **Add Enclave**.



3. Fill in the following fields:
    - **Name** – A unique identifier.
    - **Type** – Currently supports:
        - `GCP Confidential Space`
        - `AWS Nitro Enclave`
    - **Description** – (Optional) Describe the enclave usage.
4. Click **Upload** to provide the Enclave configuration in JSON format.
5. Click **Save** to complete registration.

Networking (FODN)

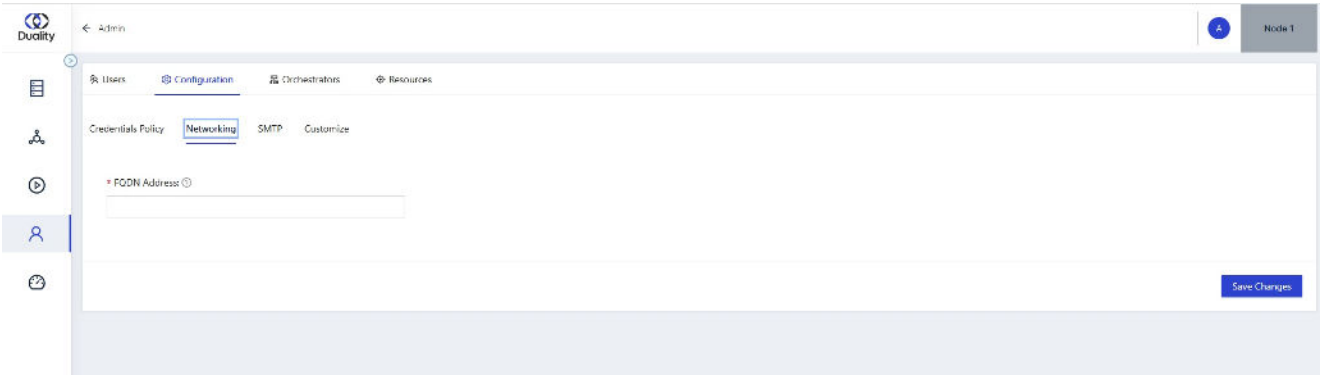
Networking (FQDN)

Use the Networking tab to configure the FQDN of the Compute node. This setting is only required if the collaboration is configured as an AI project.

The FQDN ensures that the Compute node is directly accessible by all Asset Owners participating in the collaboration. It should be set to the Compute node's FQDN or public IP address to allow external access.

To edit the Networking settings:

1. In **Admin**, select the **Configuration** > **Networking** tab.
2. Enter the value of the FQDN of the Compute node.
3. Click **Save Changes** to apply the configuration.



Handshake Process

This topic is relevant for the Collaboration Manager and the participating Nodes.

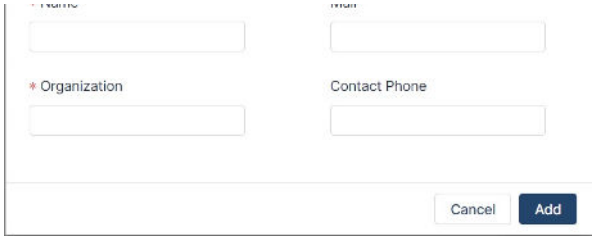
Connecting the nodes requires a “handshake” between the Collaboration Manager node and participating Nodes.

Once the Collaboration Manager has initiated the “handshake”, the participating Nodes can complete the “handshake” on their end.

From the Collaboration Manager to the Nodes

1. Login to the Collaboration Manager URL: https://<Machine IP>/
2. Click **Participants**.

3. Click **+ ADD PARTICIPANT** and enter the required details.



- **Name** [mandatory]
- **Organization** [mandatory]
- **Mail** [optional]
- **Contact Phone** [optional]

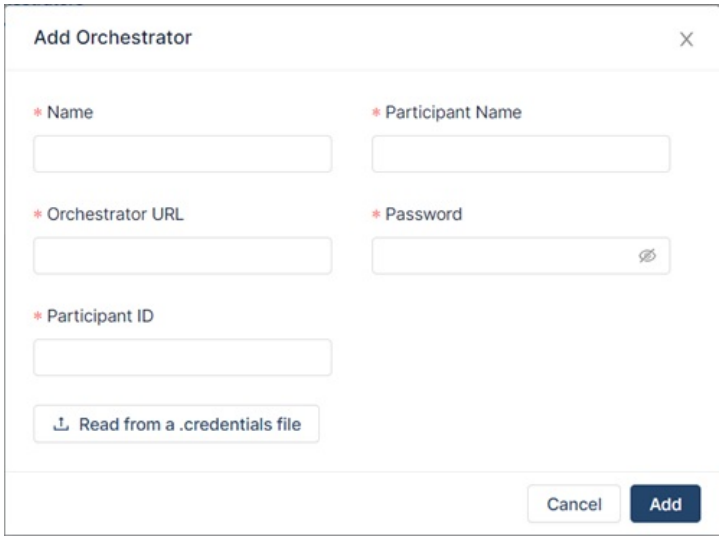
4. Click **Add**.
5. A configuration file is downloaded to the local machine. Send this file to the participating organization.

> **NOTE:** Repeat this process for each of the participating nodes.

6. Verify that all the nodes are visible in the Collaboration Participants list.

From the Nodes to the Collaboration Manager

1. Login to the Node URL: https://<Machine IP>/
2. Click **Admin**.
3. Select the **Orchestrator** tab.
4. Click **+ ADD ORCHESTRATOR**.



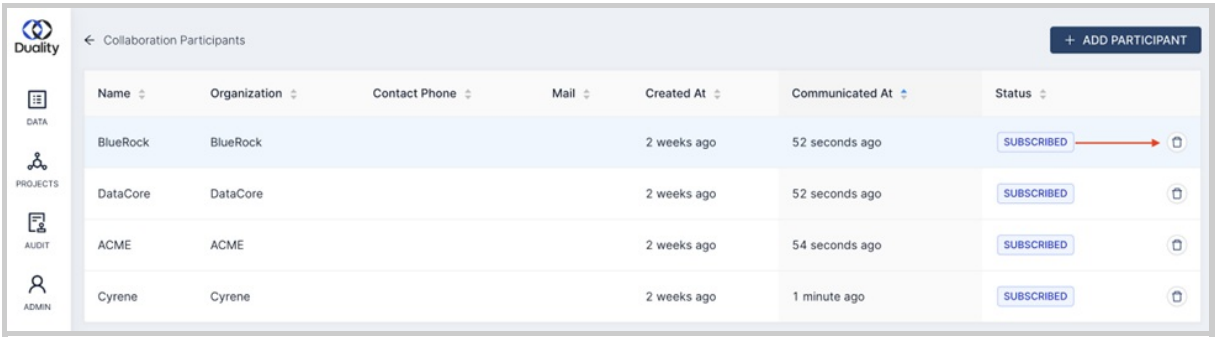
5. Fill in the **Name** [mandatory].
6. Click **Read from a .credentials file**.
7. Upload the handshake file provided by the Collaboration Manager.
8. Click **Add**.

Disconnecting/Reconnecting a Node

NOTE: This is equivalent to renaming a participant.

1. Login to the Node URL: https://<Machine IP>/
2. Click **Admin**.
3. Select the **Orchestrator** tab.
4. Click **Delete Orchestrator**.

5. Login to the Collaboration Manager URL: https://<Machine IP>/
6. Click **Participants**.
7. Click **Delete Participant**.



8. SSH into the Node, and browse to the install folder to restart the service:

   cd /install

   For Collaboration Manager: ./orchestrator.sh restart

   For Node: ./node.sh restart

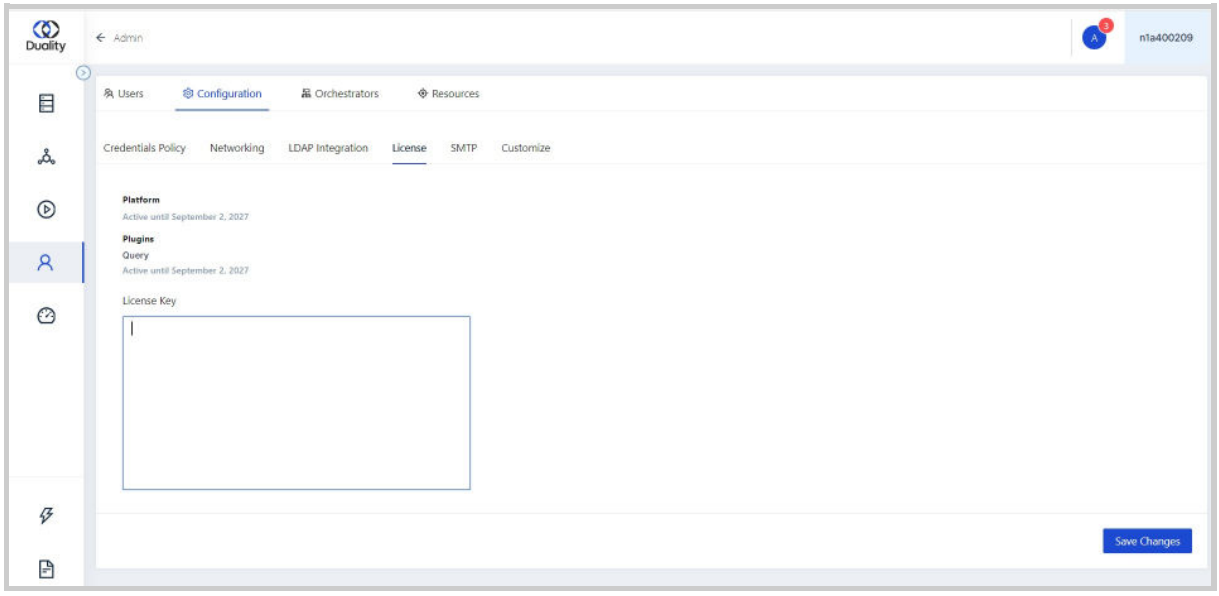
9. The node is removed. Follow the node handshake procedure to re-connect the node to the Collaboration Manager.

License

To enable the use of the platform and the application plug-ins (different plugin is installed per project type), you will receive a license key from Duality generated in accordance with the licensed products and the respective licensing period of each product.

To invoke the license:

1. In **Admin**, select the **Configuration > License** tab.



2. Copy the license key that you received, and paste it into the **License Key** field.
3. Click **Save Changes** to invoke the key.

The page refreshes, displaying the platform and supported application plug-ins, as well as the expiration date of their respective licenses.