Implementation Guide

Modular Cloud Studio on AWS

Modular Cloud Studio on AWS: Implementation Guide

Table of Contents

Overview

Modular Cloud Studio (MCS) on AWS is a solution that helps studios and production teams to build secure, scalable, and highly customizable content production studios in the AWS Cloud. You can build a tailored, cloud-based production environment within hours without extensive cloud expertise or costly upfront investments. MCS simplifies the setup by providing you with integration choices and automating deployment. Its modular framework presents options to launch remote workstations, storage, and other modules with Amazon Web Services (AWS) services and AWS Partner production systems. You can securely expand studio access to global talent, scale resources to meet project demands, and add capabilities with additional modules from the AWS Marketplace, where you can find software that runs on AWS. This way, your teams can focus on creative innovation, not technical logistics.

Media companies and entertainment studios face the growing need to become more flexible, responsive businesses that can pursue creative opportunities as they arise. Using the cloud can help these companies to:

- Accommodate new projects without complex planning and capital expenditure
- Access remote talent and vendors globally using distributed workflows

Some organizations might have concerns about migrating to the cloud. MCS helps mitigate many of these concerns:

- MCS automates and simplifies the process of setting up, integrating, and configuring regional environments for geographically diverse teams.
- MCS provides the capability to use Third-Party Modules and custom modules, so that you can keep using the products you're already familiar with.
- MCS deploys within 5-10 minutes, and you can then deploy the modules within hours.

This guide will help you build and configure a cloud studio on AWS with MCS. Read this guide for the reference architecture, components, planning considerations, and steps involved in deploying and configuring your cloud studio.

The intended audience for using this solution's features and capabilities in their environment includes system administrators, solution architects, and cloud professionals who are responsible for content production workloads and studio technology.

Use this navigation table to quickly find answers to these questions:

If you want to ...	Read ...
Know the cost for running this solution. The estimated cost for running this solution varies based on your deployment configuration and use. For example, the estimated cost in the US East (N. Virginia) Region is USD $591.55 per month for AWS resources when deploying internal MCS modules in the hub Region. This cost doesn't include modules containing AWS Independent Software Vendor (ISV) software.	Cost
Understand the security considerations for this solution. The solution deploys AWS resources within a virtual private cloud (VPC) with limited access. The solution automatically creates a default administrator user in an Amazon Cognito user pool, which is a user directory for web and mobile app authentication and authorization.	Security
Know how to plan for quotas for this solution. Make sure you have sufficient quotas for each of the services implemented in this solution, including AWS CloudFormation quotas that you should be aware when launching the stack. CloudFormation launches this solution from a template and takes care of provisioning and configuring the necessary AWS resources for you.	Quotas
Know which AWS Regions support this solution. Individual MCS modules might be available in different AWS Regions. An AWS Region is a physical location in the world where AWS has clustered data centers. Each group of logical data centers is called an Availability Zone. Each AWS Region consists of a minimum of three, isolated, and physically separate Availability Zones within a geographic area.	Supported AWS Regions
View or download the CloudFormation template included in this solution to automatically deploy the infrastructure resources (the "stack") for this	AWS CloudFormation template

If you want to ...	Read ...
solution. Templates are declarative configuration files that specify the resources you want to provision in your CloudFormation stacks.	

Features and benefits

The solution provides the following features:

Launch a production-ready studio in hours

With automated deployment, you can deploy a fully configurable virtual studio tailored to your project needs in a few hours, eliminating lengthy setup delays and becoming accessible to remote talent.

Assemble and customize your ideal toolset

Avoid vendor lock-in by incorporating your choice of leading AWS Partner and Third-Party integrations into collaborative pipelines personalized to your requirements, ensuring flexibility for each new production—or whenever your needs evolve.

Improve integration for smoother workflows

Each incremental module recognizes other active modules, and orchestration of each infrastructure component is automatic. This way, you can reduce manual integration and focus your engineering resources on creative innovation.

Deploy and scale studios to propel business growth

Create content production studios aligned to project demands. Dynamically scale your infrastructure rapidly, or gradually adopt additional cloud resources at a pace that meets your specific needs. Realize returns faster while avoiding major upfront expenditures. By using automation, decrease the time to learn configuration details and reduce the risk of misconfiguration.

Integration with AWS Service Catalog AppRegistry, myApplications, and Application Manager, a capability of AWS Systems Manager

This solution includes a Service Catalog AppRegistry resource to register the solution's CloudFormation template and its underlying resources as an application in both Service

Catalog AppRegistry and Application Manager. It also includes an application registered with myApplications. With this integration, centrally manage the solution's resources and enable application search, reporting, and management actions.

Use cases

Expand physical facility capabilities

Accommodate multiple productions by building a different virtual studio for each project you need, and outfit each one with the right tools for the job.

Grow beyond geographic boundaries

Help production teams to scale and access talent anywhere in the world to align with project needs, without accumulating excess infrastructure.

Onboard vendors securely

Create an environment that allows external vendors to do creative work in a single, centralized hub on your systems, instead of sending data back and forth and risking confidential IP.

Accelerate cloud migration

Automate the deployment of scalable, secure, global content production environments, while aligning to MovieLabs 2030 Vision and its guidance for infrastructure interoperability.

Concepts and definitions

This section describes key concepts and defines terminology specific to MCS:

module

A CloudFormation deployment launched by Service Catalog through the MCS web console or API. Service Catalog provisions the module stack, and CloudFormation takes care of provisioning and configuring the necessary AWS resources for you.

enable module

Enabling means activating a module such that its resources are included in the MCS content production studio. In other words, deploying the CloudFormation stack that represents the module.

register module

Registering makes an external module known to MCS and available for an MCS administrator to enable. Registering does not enable the module.

AWS developed MCS modules

The set of modules developed by AWS that are included with MCS and available when MCS is deployed. When you deploy MCS, all of these modules are available without an explicit registration step. Additionally, these modules cannot be deregistered.

Third-Party Modules

Similar to AWS Partner storage modules, the MCS admin user must explicitly register third-party modules with MCS to make them available to users.

> **Note**
>
> Modular Cloud Studio on AWS allows you to deploy and manage a scalable, secure, and global content production infrastructure in the cloud. This includes custom modules, developed by AWS Partners or other third parties, that you can choose to use ("Third-Party Modules"). AWS does not own or otherwise have any control over Third-Party Modules. Your use of the Third-Party Modules is governed by any terms provided to you by the Third-Party Module providers when you acquired your license to use them (for example, their terms of service, license agreement, acceptable use policy, and privacy policy). You are responsible for ensuring that your use of the Third-Party Modules comply with any terms governing them, and any laws, rules, regulations, policies, or standards that apply to you. You are also responsible for making your own independent assessment of the Third-Party Modules that you use. AWS does not make any representations, warranties, or guarantees regarding the Third-Party Modules, which are "Third-Party Content" under your agreement with AWS. Modular Cloud Studio on AWS is offered to you as "AWS Content" under your agreement with AWS.

AWS Partner storage modules

A curated list of storage modules from AWS ISVs that are treated similar to Third-Party Modules or custom modules. MCS displays the AWS Partner storage modules, letting users know that these modules exist and can be registered with MCS.

Note

AWS Partner storage modules are easily discoverable. However, to make them available to users, the MCS admin user must review and register these modules with MCS.

hub Region

The Region from which you launch the solution.

spoke Region

Region from which you launch a module, different from the hub Region. You can optionally use spoke Regions to increase availability and reliability for geographically diverse teams.

Note

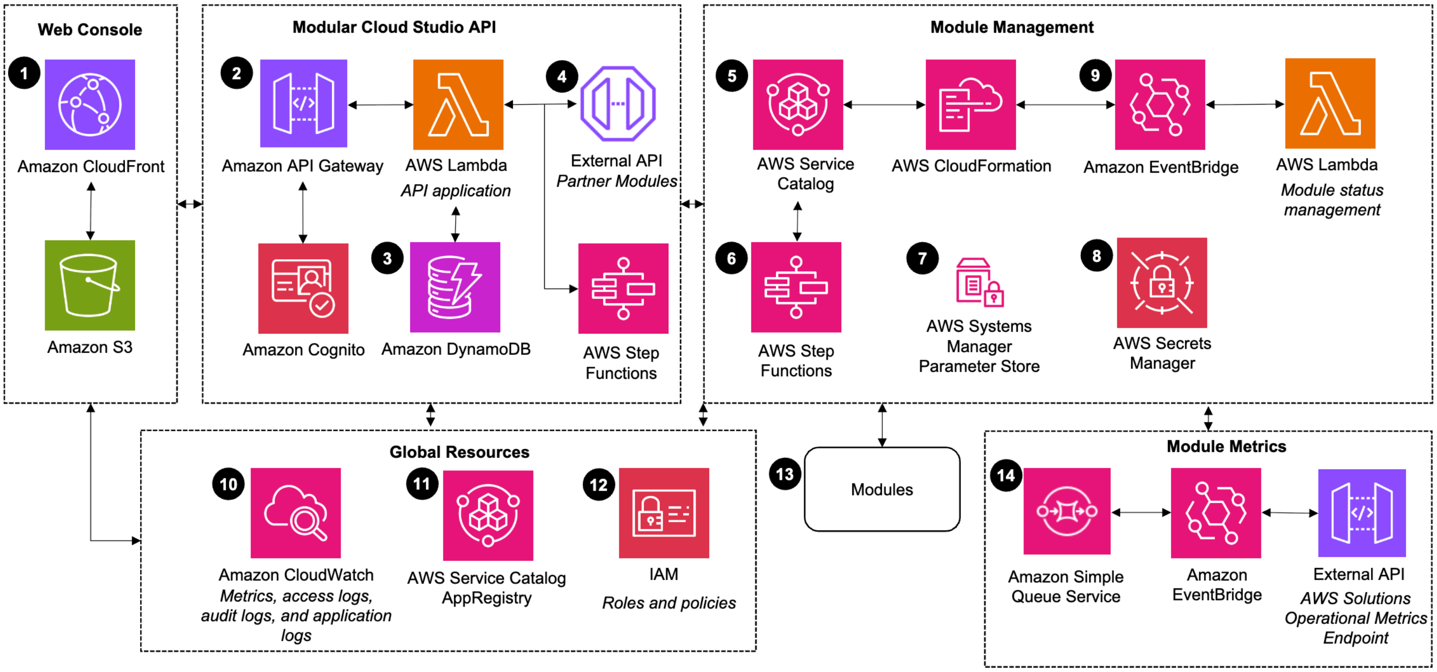
For a general reference of AWS terms, see the AWS Glossary.

Architecture overview

This section provides a reference implementation architecture diagram for the components deployed with this solution.

Architecture diagram

Deploying this solution with the default parameters deploys the following components in the your AWS account.



> **Note**
>
> AWS CloudFormation resources are created from AWS Cloud Development Kit (AWS CDK) constructs.

The high-level process flow for the solution components deployed with the AWS CloudFormation template is as follows:

1. Amazon CloudFront caches and delivers a single-page application built in React hosted as a static website in an Amazon Simple Storage Service (Amazon S3) bucket.

2. A REST API Gateway integrates with Amazon Cognito and then passes along authenticated requests to an AWS Lambda function. The Lambda function handles all API requests coming from the frontend.
3. Amazon Dynamo DB contains several tables that manage information about available modules and the state of enabled modules.
4. External location which hosts Custom modules developed by AWS Partners.
5. AWS Service Catalog hosts the AWS CloudFormation templates for all previously included modules and Third-Party Modules that are registered post-deployment.
6. AWS Step Functions is used to manage registering and de-registering Third-Party Modules.
7. AWS Systems Manager Parameter Store contains module parameters that contain sensitive information. Some parameters are deployed by the MCS stack while others are deployed by modules. See the Developer guide for more information.
8. AWS Secrets Manager contains module parameters that contain sensitive information.
9. Amazon EventBridge is configured to listen to CloudFormation events about modules that are passed along to a Lambda function. The Lambda function processes the events and updates the module's information in the solution's Amazon DynamoDB tables.
10. Amazon CloudWatch log groups collect and store logs across the solution.
11. The solution registers resources deployed by the stack against AWS Service Catalog AppRegistry and an application on myApplications.
12. AWS Identity and Access Management (IAM) roles and policies are used across the solution to manage access and permissions.
13. You can launch this solution's modules via the web console or API.
14. Amazon Simple Queue Service (SQS) delivers operational metrics to Amazon EventBridge where they are transformed and sent to an API destination for monitoring.

AWS Well-Architected design considerations

This solution uses the best practices from the AWS Well-Architected Framework, which helps customers design and operate reliable, secure, efficient, and cost-effective workloads in the cloud.

This section describes how the design principles and best practices of the Well-Architected Framework benefit this solution.

Operational excellence

This section describes how we architected this solution using the principles and best practices of the operational excellence pillar.

- This solution automates the deployment and configuration of your cloud environment by using AWS services and AWS Partner integrations.
- MCS tracks the assets that are deployed with CloudWatch and AWS CloudTrail. It also tracks logs from Amazon Elastic Compute Cloud (Amazon EC2), Amazon FSx for Windows File Server, and AWS Directory Service to provide observability into the infrastructure and solution components.

Security

This section describes how we architected this solution using the principles and best practices of the security pillar.

- AWS resources that are deployed by the solution, such as Amazon EC2 instances and networking components installed in modules, are deployed within a VPC with limited access.
- Upon deployment, the solution automatically creates a default administrator in the Amazon Cognito user pools. This user is part of the administrator group, which assumes the MCS Administrator IAM role. This role grants administrator permissions such as installing modules and viewing stored secrets within the account.
- The solution securely stores sensitive data classified as confidential, such as administrator username and password used in the Managed Active Directory module, in Secrets Manager.
- The MCS web interface is publicly available via CloudFront, and the traffic travels through HTTPS protocol.
- Users must authenticate via Amazon Cognito to use the MCS web console. The solution only allows authorized requests, whether the MCS API is accessed through the provided web interface or through a custom client. The MCS API is provided through Amazon API Gateway by using an Amazon Cognito authorizer.

Reliability

This section describes how we architected this solution using the principles and best practices of the reliability pillar.

- MCS simplifies the deployment of the workloads required to build a studio in the cloud, automates the configuration and integration of modules, which helps to avoid misconfigurations.
- Optionally, you can configure the solution to use FSx for Windows File Server, which sets up and provisions file servers and storage volumes, replicates data, manages failover and failback, and eliminates much of the administrative overhead.

Performance efficiency

This section describes how we architected this solution using the principles and best practices of the performance efficiency pillar.

- The solution helps users to launch a global studio in the cloud within hours.
- The solution supports the deployment of MCS modules across multiple AWS Regions. This provides lower latency and a better experience for editors, content creators, and other production users.
- The MCS management layer is entirely serverless and event-driven, removing the need to run and maintain physical servers. Data is stored in Amazon S3 and DynamoDB, and static web assets are served through CloudFront. The API is provided through API Gateway and Lambda.

Cost optimization

This section describes how we architected this solution using the principles and best practices of the cost optimization pillar.

- The cost for running MCS varies, based on how it is configured to deploy and how it is subsequently used over time. Some examples that influence cost include the following:
  - Number of Amazon EC2 workstations
  - How long your Amazon EC2 workstations run daily
  - How much data you transfer into MCS storage resources

See Cost for more detail.

- You can measure the efficiency of the workloads, and the costs associated with delivery, by using AWS Service Catalog AppRegistry or AWS myApplications. See Monitoring the solution with AWS Service Catalog AppRegistry for more detail.

Sustainability

This section describes how we architected this solution using the principles and best practices of the sustainability pillar.

- The solution uses managed and serverless services where possible to minimize the environmental impact of the backend services.
- Travel and transportation is a significant source of carbon emissions in media and entertainment workflows. MCS helps video editors and other post-production team members to work on remote cloud-based virtual desktops to lessen the need to travel to a facility to perform their work.
- Customers can deploy MCS in one of the supported Regions (hub), and optionally enable additional Regions (spoke), based on both business requirements and sustainability goals to optimize performance, cost, and carbon footprint.
- You can deploy your MCS studio close to end users, resulting in reduced latency, reduced distance that network traffic must travel, and fewer total network resources required to support your workload.
- MCS can help you optimize team member resources for the activities performed by using virtual desktops to limit upgrade and device requirements.
- You can use shared file systems or storage such as Amazon FSx for Windows File Server to access common data, avoid data duplication, and allow for more efficient infrastructure for your workloads.
- The modular design of MCS helps you to size cloud resources to match the needs of a specific project, lower a workload’s environmental impact, reduce costs, and maintain performance benchmarks.
- Using managed services supported in MCS shifts the responsibility to AWS, which has insights across millions of customers that can help drive new innovations and efficiencies. Managed services also distribute the environmental impact of the service across many users because of the multi-tenet control planes.

Architecture details

This section describes the components and AWS services that make up this solution and the architecture details on how these components work together.

AWS services in this solution

AWS service	Description
AWS CloudFormation	**Core**. Used to deploy the solution and develop MCS internal and Third-Party Modules.
Amazon CloudFront	**Core**. Used to cache and deliver the MCS web console hosted in Amazon S3.
Amazon Cognito	**Core**. Provides authentication to the MCS web console and API.
Amazon DynamoDB	**Core**. Used to store information about MCS modules and the state of the modules.
Amazon EC2	**Core**. Used to run the workstations managed by the MCS Workstation Management module. MCS uses Amazon EC2 Image Builder to build Windows and Linux Amazon Machine Images (AMIs) used in the solution.
AWS Global Accelerator	**Core**. Used to manage connections between MCS Workstation Management module and Amazon EC2 workstations.
IAM	**Core**. Used to authorize access to MCS using roles to manage resources effectively. MCS resources are limited by roles and policies defined in IAM and in Cognito user pools.
AWS Lambda	**Core**. Handles the processing logic for adding, updating, editing, or deleting MCS modules and storing sensitive information in Secrets Manager.
Amazon RDS for PostgreSQL	**Core**. Used as a database for the Leostream Broker EC2 instances.

AWS service	Description
Amazon Route 53	**Core**. Used to manage domain resolution to load balancer addresses.
AWS Secrets Manager	**Core**. Used to store module parameters that contain sensitive information.
AWS Service Catalog	**Core**. Used to manage the portfolio of MCS modules and to provision the CloudFormation stack when modules are enabled.
Amazon VPC	**Core**. Used to deploy an isolated virtual networking environment to build the MCS studio. Users can create a new VPC or import an existing one.
Amazon CloudWatch	**Supporting.** Used for monitoring the solution and logs.
Amazon EventBridge Event Bus	**Supporting.** Listens to CloudFront changes and invokes Lambda to update the state of MCS modules in DynamoDB.
Amazon EventBridge Pipe	**Supporting.** Used to process and transform operational metrics from Amazon SQS and deliver them anonymously to an API destination for monitoring.
Amazon SQS	**Supporting.** Used to deliver operational metrics to EventBridge Pipe
Amazon Simple Storage Service	**Supporting.** Provides object storage for content used in the MCS web console.
AWS Systems Manager Parameter Store	**Supporting.** Provides application-level resource monitoring, visualization of resource operations, and secrets management.
Amazon DCV	**Supporting**. Used to connect users securely to the workstations.
AWS Directory Service	**Optional**. Used to deploy an instance of AWS Managed Microsoft AD.

AWS service	Description
Amazon FSx for Windows File Server	**Optional**. Used to deploy a fully managed shared file system built on Windows Server.
Amazon FSx for Lustre	**Optional**. Used to deploy a fully managed shared file system built on Lustre.
AWS Step Functions	**Optional**. Used to register and deregister MCS Third-Party Modules.

Module categories

MCS supports five module categories: Network, Identity, Storage, Workstation Management, and Custom.

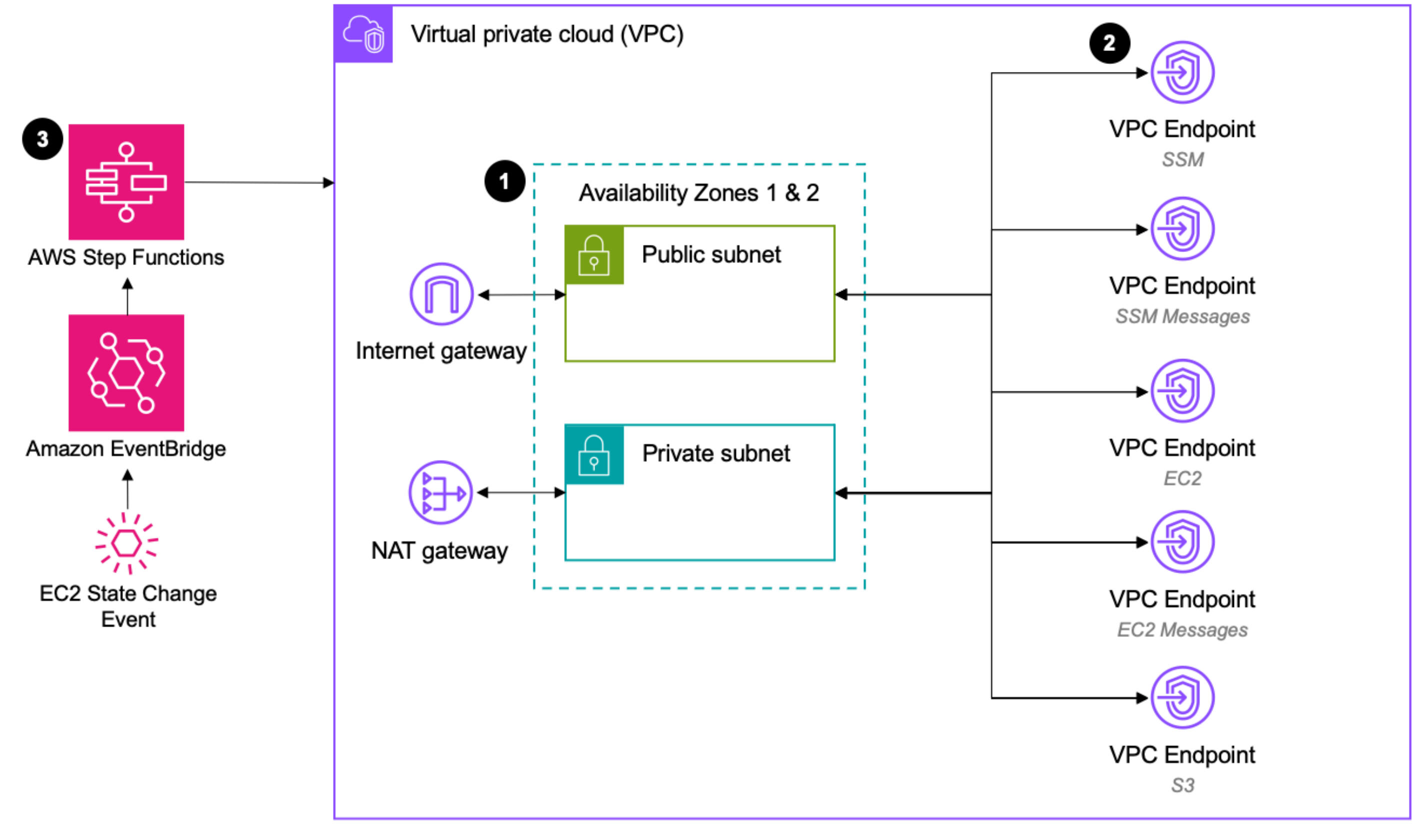
Network modules

Network modules create the necessary resources for other modules and components to communicate with each other.

The following Network modules are available in MCS after deployment:

- Managed VPC module - Deploys a new VPC
- Unmanaged VPC module - Receives existing VPC information from an input form

Managed VPC module

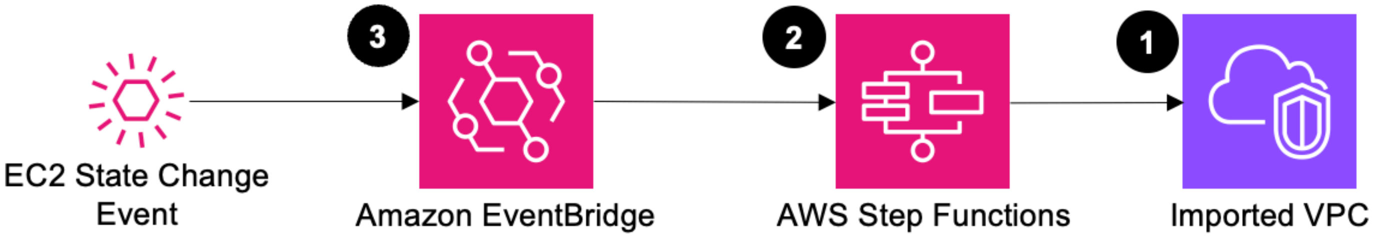


1. The Solution deploys a VPC with two Availability Zones. Each zone contains:
   - One public subnet - routes traffic to an internet gateway
   - One private subnet - routes traffic to a NAT gateway

> **Note**
>
> Pixel streaming traffic doesn't travel through the NAT gateway.

2. The solution creates VPC Endpoints to ensure that internal traffic to these services connects privately and doesn't traverse the public internet.
3. Default EventBridge buses in each enabled region send EC2 instance state change events to a state machine for applying tags to any EC2 instance launched within an MCS VPC.

Unmanaged VPC module

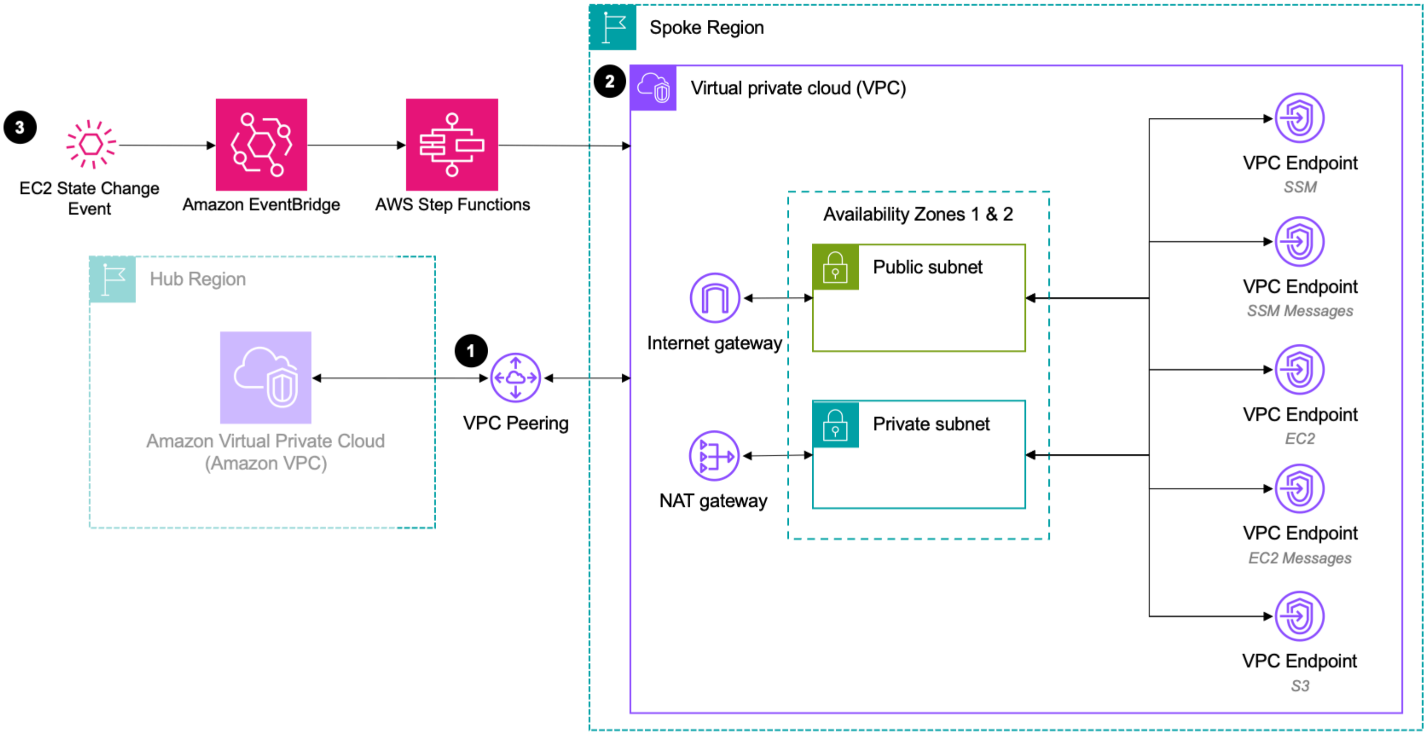


1. The solution can utilize an existing VPC for module deployment. However, any additional configuration required for module functionality must be managed by the MCS administrator.
2. You can optionally enable EventBridge infrastructure for automatic EC2 tagging through the Unmanaged VPC's deployment parameters (default: Yes). When enabled, default EventBridge buses in each enabled region send EC2 instance state change events to a state machine for applying tags to any EC2 instance launched within an MCS VPC.
3. To toggle EC2 tagging post-deployment, locate the EventBridge rule created by your Unmanaged VPC deployment (rule name contains `EC2InstanceTagging`) and choose **Disable** or **Enable** as needed.

Note

If EventBridge EC2 Tagging Parameter is disabled at deployment, tagging infrastructure will not be deployed and the feature cannot be enabled later. If enabled, tagging can be toggled on or off post-deployment by enabling or disabling the EventBridge Rule.

Spoke Managed VPC module



1. The solution establishes a VPC peering connection between the existing VPC in the hub Region and the VPC being created in this module, enabling inter-VPC communication.
2. The solution creates a VPC spanning two Availability Zones. Each zone contains:
    - One public subnet - routes traffic through an Internet Gateway
    - One private subnet - routes outbound traffic through a NAT gateway
3. Default EventBridge buses in each enabled region send EC2 instance state change events to a state machine for applying tags to any EC2 instance launched within an MCS VPC.

Identity modules

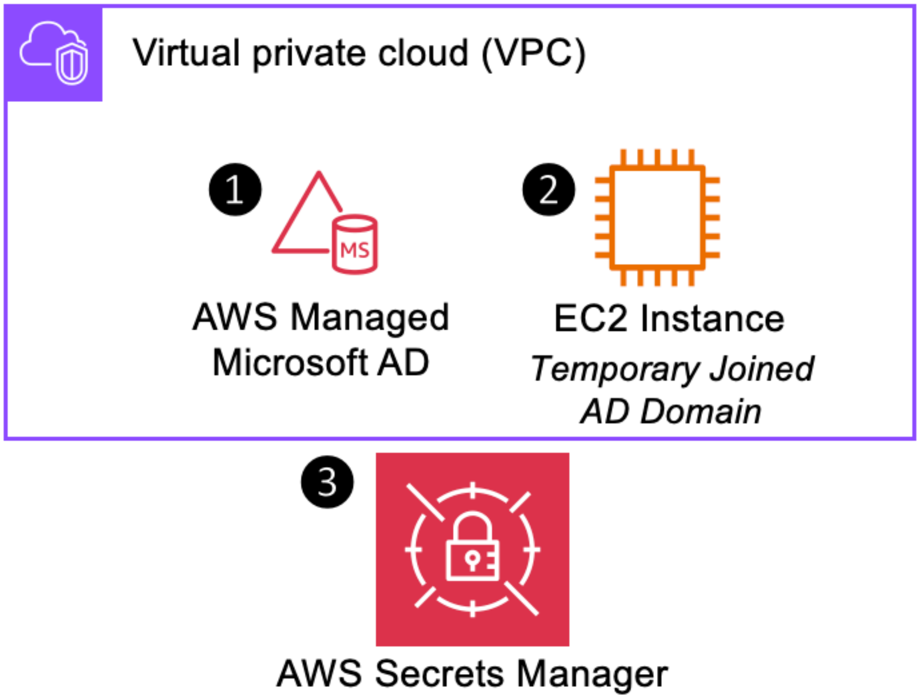
Identity modules create the necessary resources to allow users to interact with MCS and the post production environment.

The following Identity modules are available in MCS after deployment:

- Managed Active Directory module - Deploys a new Microsoft Active Directory instance under standard edition

- Unmanaged Active Directory module - Receives existing Microsoft Active Directory information from an input form

Managed Active Directory module



1. Directory Service deploys an instance of AWS Managed Microsoft AD under standard edition.
2. The Active Directory module deploys a temporary EC2 instance that:
    - Joins to the AWS Managed Microsoft AD domain
    - Sets password policy for domain users (90-day expiration)
    - Self-terminates after approximately 5 minutes
3. User credentials generated during deployment are automatically stored in AWS Secrets Manager.

Spoke Managed Identity module

1. Directory Service deploys an AD Connector instance that establishes a connection to the Microsoft AD instance in the Hub environment.

Storage modules

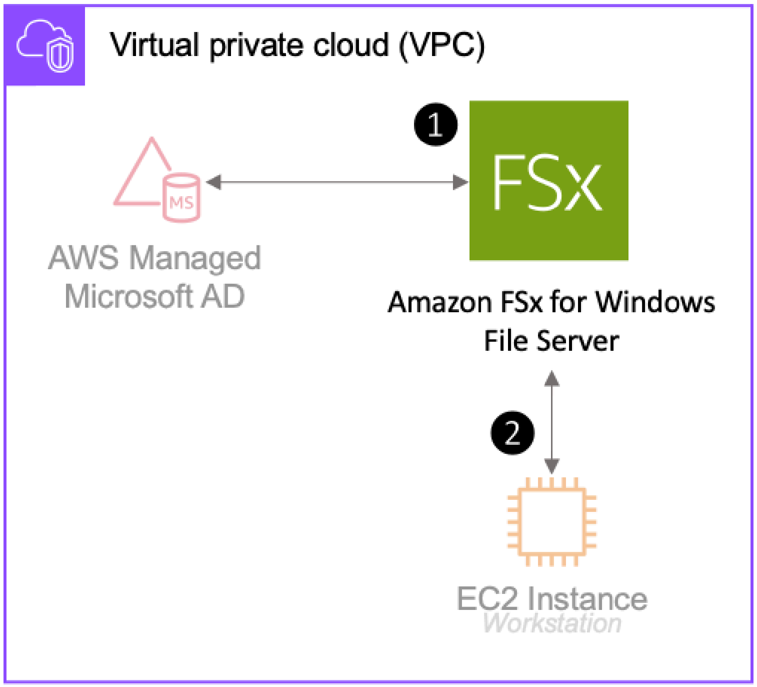
Storage modules create the necessary resources to allow workstations to save and retrieve data from file systems in the post production environment. The following Storage modules are available in MCS after deployment:

- FSx for Windows File Server module - Deploys a new Amazon FSx Windows Server file system and registers to the Microsoft Active Directory
- FSx for Lustre File Server module - Deploys a new Amazon FSx Lustre file system

Note

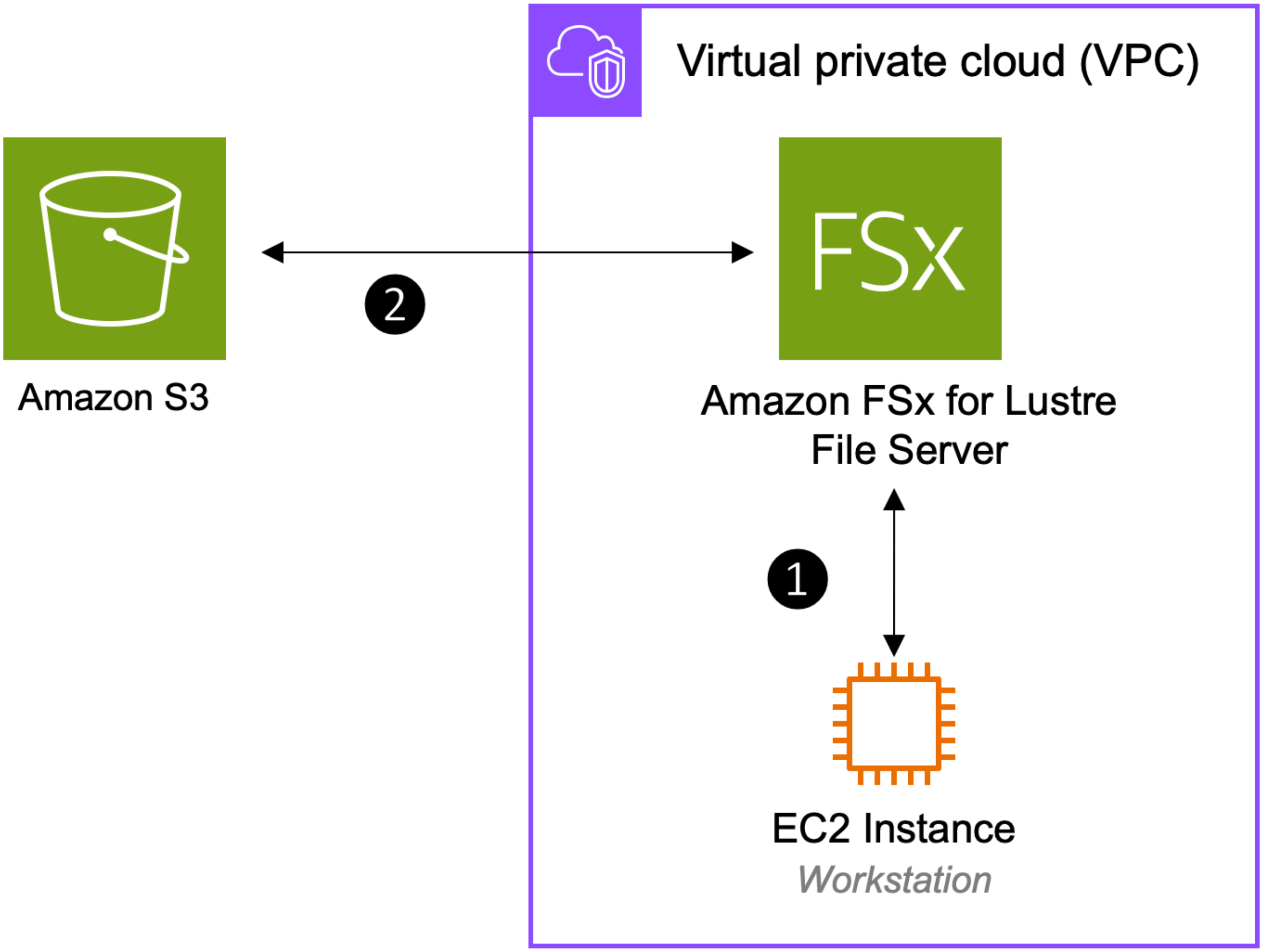
Modular Cloud Studio on AWS allows you to deploy and manage a scalable, secure, and global content production infrastructure in the cloud. This includes custom modules, developed by AWS Partners or other third parties, that you can choose to use ("Third-Party Modules"). AWS does not own or otherwise have any control over Third-Party Modules. Your use of the Third-Party Modules is governed by any terms provided to you by the Third-Party Module providers when you acquired your license to use them (for example, their terms of service, license agreement, acceptable use policy, and privacy policy). You are responsible for ensuring that your use of the Third-Party Modules comply with any terms governing them, and any laws, rules, regulations, policies, or standards that apply to you. You are also responsible for making your own independent assessment of the Third-Party Modules that you use. AWS does not make any representations, warranties, or guarantees regarding the Third-Party Modules, which are "Third-Party Content" under your agreement with AWS. Modular Cloud Studio on AWS is offered to you as "AWS Content" under your agreement with AWS.

Amazon FSx for Windows File Server module


Virtual private cloud (VPC)
1
FSx
AWS Managed Microsoft AD
MS
Amazon FSx for Windows File Server
2
EC2 Instance
Workstation

1. The solution deploys the Amazon FSx for Windows File Server file system and integrates it with the Microsoft Active Directory instance deployed by the Identity module.
2. You can mount this file system manually onto workstations started by the Leostream Broker module module.

Amazon FSx for Lustre File Server module


Virtual private cloud (VPC)
Amazon S3
2
FSx
Amazon FSx for Lustre File Server
1
EC2 Instance
Workstation

1. You can mount this file system manually onto workstations started by the Leostream Broker module module.
2. You can optionally specify an S3 Path to enable a Data Repository Association for the file system.

Workstation Management modules

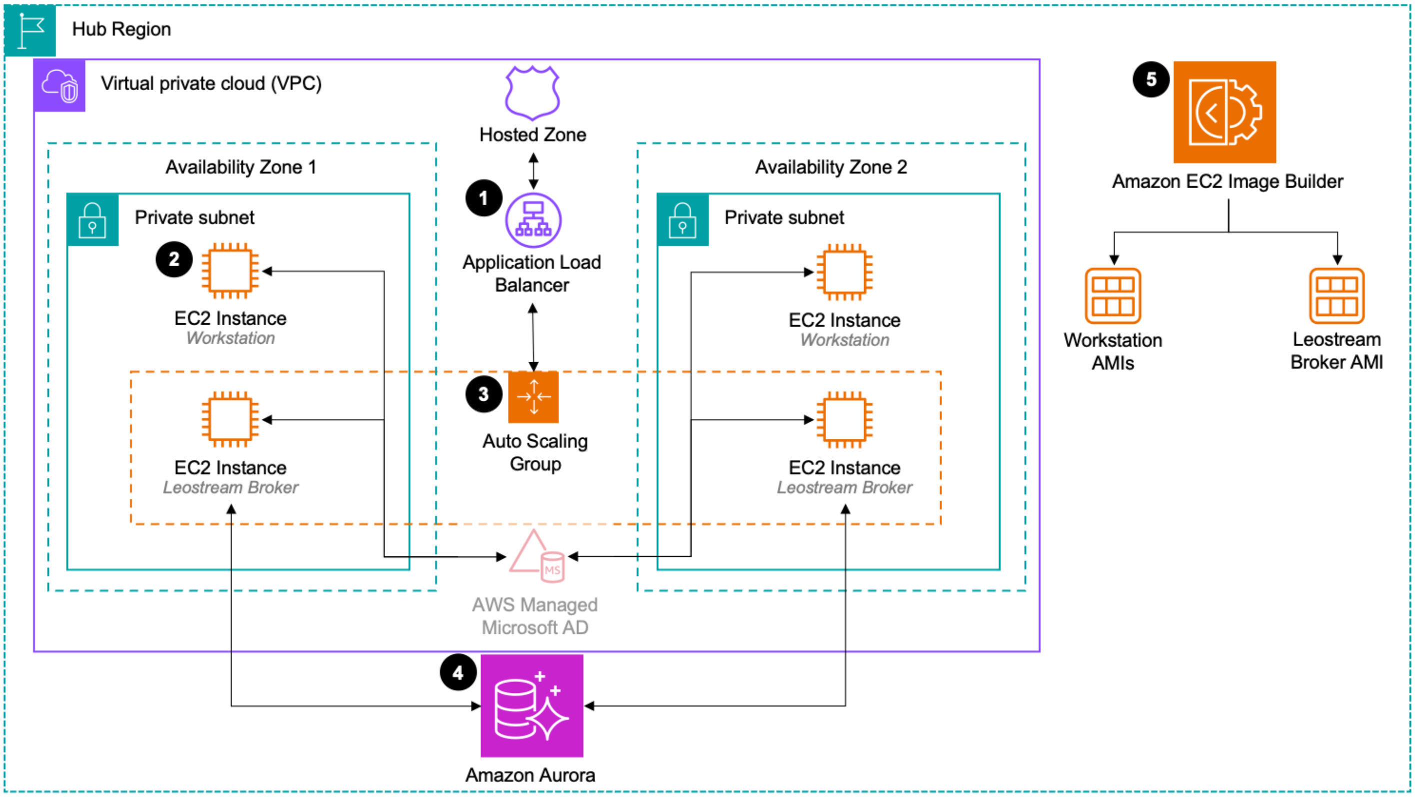
Workstation Management modules create the necessary resources to provide virtual workstations to users in the post-production environment. This way, users can gain cloud performance by connecting remotely only from their local laptop, and handle auto-scaling based on policies.

The following Workstation Management modules are available in MCS after deployment:

- Leostream Broker module - Manages assignment and auto scaling of workstations
- Leostream Gateway module - Manages connections to workstations

> **Note**
>
> Modular Cloud Studio on AWS allows you to deploy and manage a scalable, secure, and global content production infrastructure in the cloud. This includes custom modules, developed by AWS Partners or other third parties, that you can choose to use ("Third-Party Modules"). AWS does not own or otherwise have any control over Third-Party Modules. Your use of the Third-Party Modules is governed by any terms provided to you by the Third-Party Module providers when you acquired your license to use them (for example, their terms of service, license agreement, acceptable use policy, and privacy policy). You are responsible for ensuring that your use of the Third-Party Modules comply with any terms governing them, and any laws, rules, regulations, policies, or standards that apply to you. You are also responsible for making your own independent assessment of the Third-Party Modules that you use. AWS does not make any representations, warranties, or guarantees regarding the Third-Party Modules, which are "Third-Party Content" under your agreement with AWS. Modular Cloud Studio on AWS is offered to you as "AWS Content" under your agreement with AWS.

Leostream Broker module

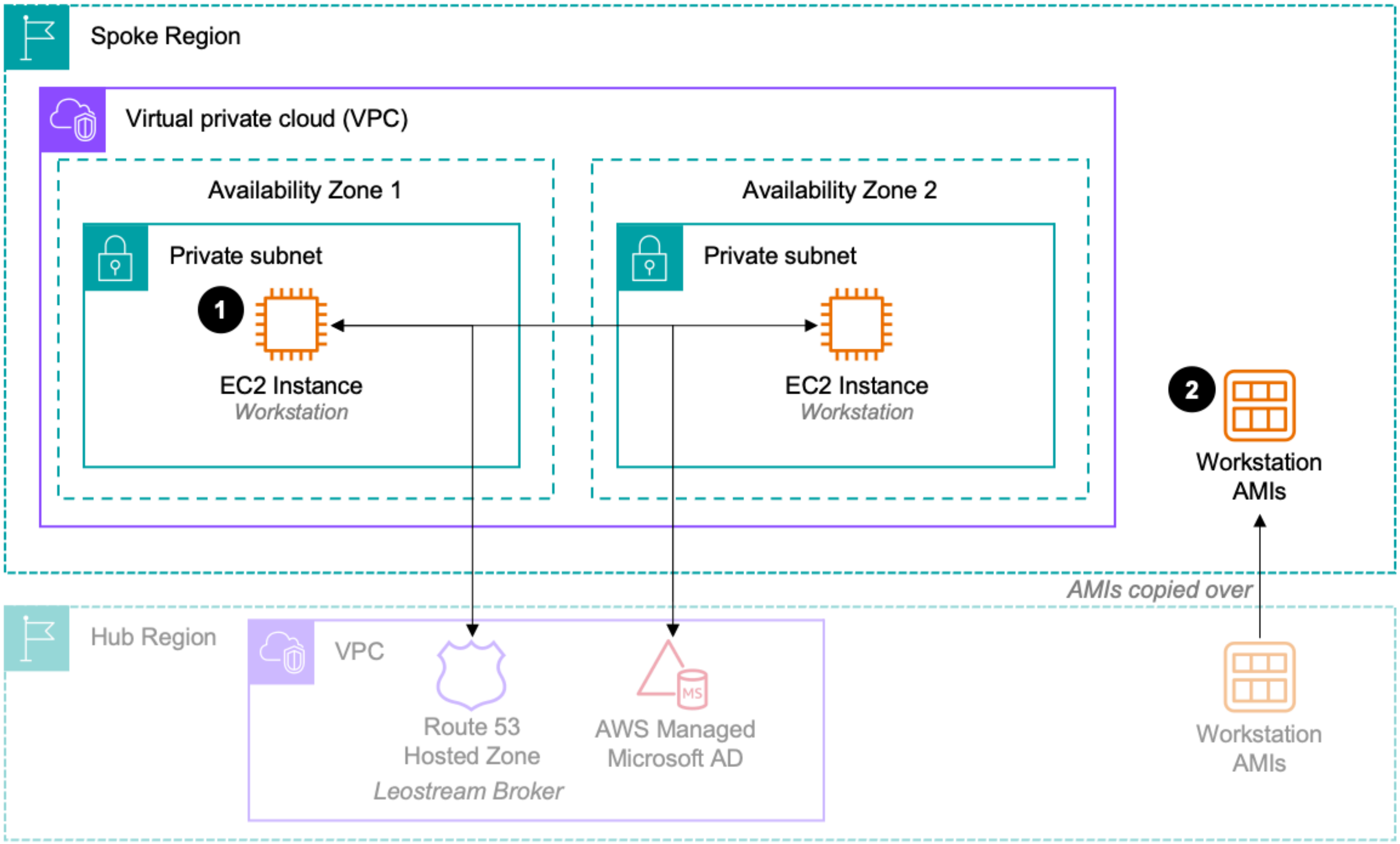


1. A privately hosted zone in Amazon Route 53 routes requests to an Application Load Balancer that is accessible through a private subnet. This Application Load Balancer manages connections to an Amazon EC2 Auto Scaling Group.
2. This module manages Leostream workstations on Amazon EC2.
3. An Amazon EC2 Auto Scaling Group maintains the necessary number of Leostream Broker instances on Amazon EC2.
4. The Leostream Broker EC2 instances use an Amazon Relational Databases Service (Amazon RDS) for PostgreSQL database.

   Database configuration:

   - A dedicated "leostream" user is provisioned with specific permissions
   - User can create, read, update, and delete databases and tables
   - Operates with restricted privileges compared to the default administrator account
   - Default admin user credentials are not exposed to or utilized by Leostream modules
5. Amazon EC2 Image Builder is used upon deployment to build the AMI for the Leostream Broker EC2 instances along with both Windows and Linux AMIs that the Leostream Broker module uses.

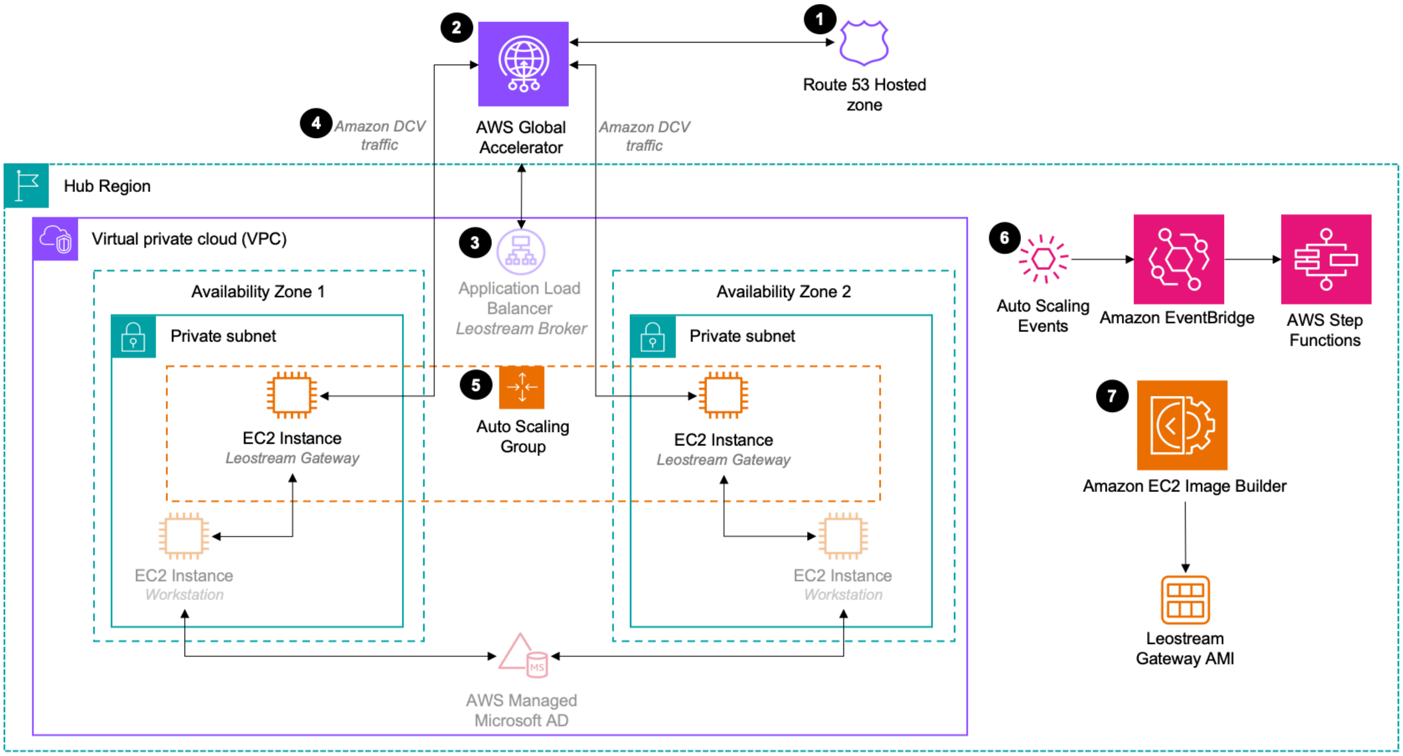
Spoke Leostream Broker module



1. The Leostream Broker cluster in the hub variant of this module manages workstations on Amazon EC2.

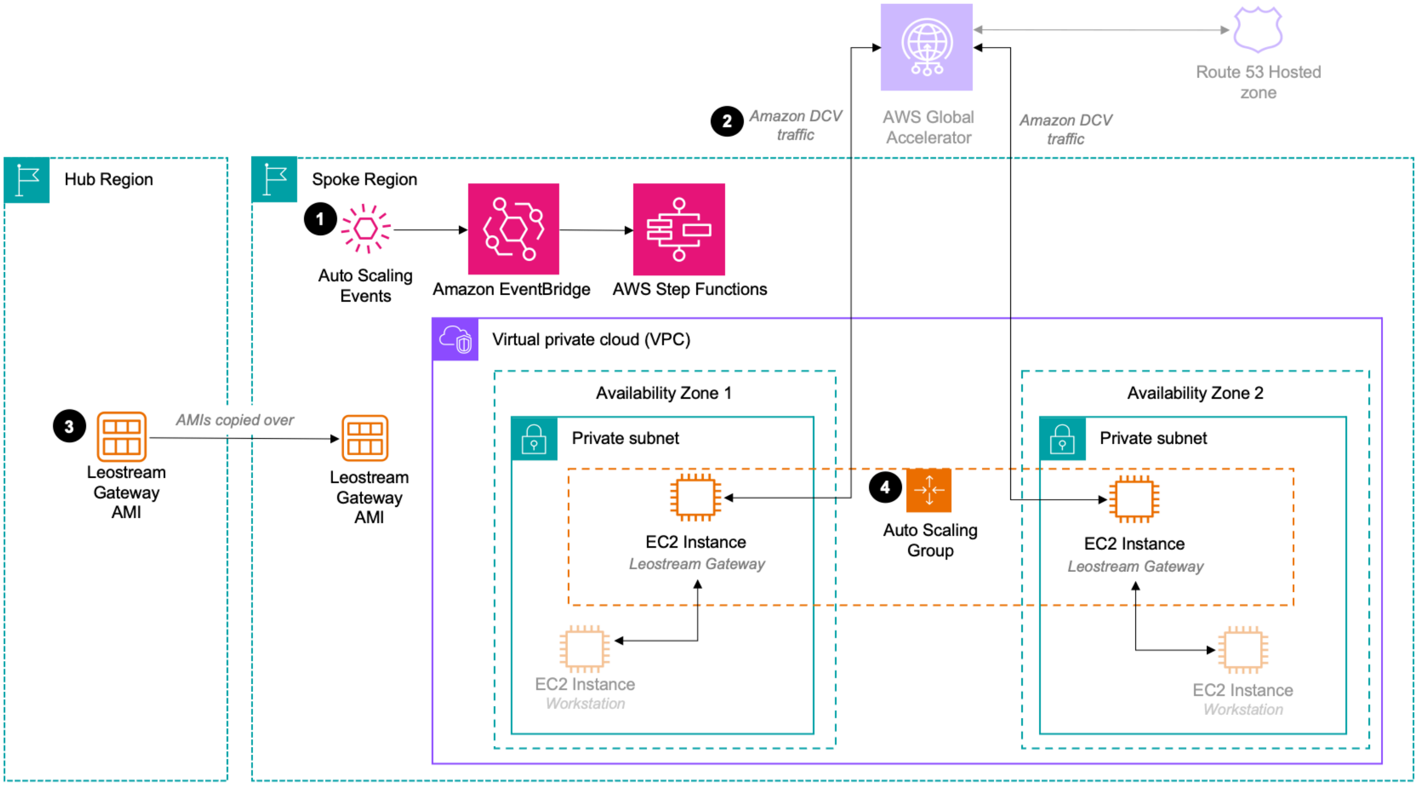
2. Workstations use the same AMIs built by the Leostream Broker module in the hub Region. The Spoke Leostream Broker module copies AMIs from the hub Region into the spoke Region.

Leostream Gateway module



1. Optionally, when a certificate and hosted zone are configured, a hosted zone on Amazon Route 53 routes requests to AWS Global Accelerator.
2. The module deploys an AWS Global Accelerator and uses it to manage connections with Leostream Gateway to either a workstation or the Leostream Broker cluster.
3. The module uses the Application Load Balancer deployed by the Leostream Broker module which manages traffic to the Auto Scaling Group for Leostream Broker instances.
4. Amazon DCV traffic is routed securely between the AWS Global Accelerator, Leostream Gateway, and workstations.
5. An Auto Scaling Group maintains the necessary number of Leostream Gateway instances on Amazon EC2.
6. Auto Scaling events invoke workflows in AWS Step Functions via Amazon EventBridge to manage Leostream Gateway registrations.
7. The AMI used by Leostream Gateway EC2 instances is built during deployment using Amazon EC2 Image Builder.

Spoke Leostream Gateway module



1. Auto Scaling events invoke workflows in AWS Step Functions via Amazon EventBridge to manage Leostream Gateway registrations.
2. Amazon DCV traffic is routed securely between the AWS Global Accelerator, Leostream Gateway, and workstations.
3. Workstations use the same AMIs built by the Leostream Broker module in the hub Region.
4. An Auto Scaling Group maintains the necessary number of Leostream Gateway instances on Amazon EC2.

Custom modules

You can bring your own custom modules, or modules developed by AWS Partners or third parties to MCS. You can register your own custom modules under the Custom category if they don't belong to the other four categories. Modules registered and enabled under the Custom category are displayed under the **Custom** menu in the MCS user interface.

Custom modules developed by AWS Partners are hosted in GitHub, and they are imported into MCS as `AVAILABLE` status when the solution is first deployed. You can fetch the latest listings from the repository when new modules are added or existing modules are modified or removed from the Module Library page.

Note

Modular Cloud Studio on AWS allows you to deploy and manage a scalable, secure, and global content production infrastructure in the cloud. This includes custom modules, developed by AWS Partners or other third parties, that you can choose to use ("Third-Party Modules"). AWS does not own or otherwise have any control over Third-Party Modules. Your use of the Third-Party Modules is governed by any terms provided to you by the Third-Party Module providers when you acquired your license to use them (for example, their terms of service, license agreement, acceptable use policy, and privacy policy). You are responsible for ensuring that your use of the Third-Party Modules comply with any terms governing them, and any laws, rules, regulations, policies, or standards that apply to you. You are also responsible for making your own independent assessment of the Third-Party Modules that you use. AWS does not make any representations, warranties, or guarantees regarding the Third-Party Modules, which are "Third-Party Content" under your agreement with AWS. Modular Cloud Studio on AWS is offered to you as "AWS Content" under your agreement with AWS.

DynamoDB Tables

MCS uses five DynamoDB tables: Registered Modules, External Modules, Modules Mapping, Enabled Modules, and Regions.

Registered Modules table

The Registered Modules table contains modules that are registered and can be enabled or disabled.

Attributes:

- `registered_module_pk` - Primary key consisting of `module_name` and `region_type`.
- `module_version` - The version of the module.
- `category` - Can be one of the following: `Network`, `Identity`, `WorkstationManagement`, `Storage`, `Custom`, or `SpokeRegionInfrastructure`.
- `input_parameters_hub` Systems Manager parameter inputs for the modules that exist in the hub Region.

- `input_parameters_local` - Systems Manager parameter inputs for the modules that exist in its local Region (hub or spoke).
- `is_external` - Boolean value indicating if the module is a Third-Party Module.
- `module_name` - Name of the module.
- `region_type` - Indicates whether the module can be enabled in the hub Region, a spoke Region, or both. Can be one of the following: `HUB`, `SPOKE`, or `BOTH`.
- `servicecatalog_portfolio_id` - MCS Service Catalog portfolio.
- `servicecatalog_product_id` - Service Catalog product of the module.
- `status` - Can be one of the following: `REGISTERED`, `REGISTERING IN PROGRESS`, `REGISTER FAILED`, `DE-REGISTERING IN PROGRESS`, `DE-REGISTER FAILED`, or `ENABLING IN PROGRESS`.

External Modules table

The External Modules table contains Third-Party Modules that are registered or are available to be registered.

Attributes:

- `module_name` - Name of the module.
- `category` - Can be one of the following: `Network`, `Identity`, `WorkstationManagement`, `Storage`, `Custom`, or `SpokeRegionInfrastructure`.
- `created_at` - When the module was created.
- `display_name` - Module name to display in the UI.
- `is_custom` - Indicates whether the module was registered post-deployment of MCS.
- `manifest_url` - The URL for the manifest.
- `status` - Can be one of the following: `AVAILABLE`, `REGISTERED`, `REGISTER IN PROGRESS`, `REGISTER FAILED`, `DE-REGISTER IN PROGRESS`, or `DE-REGISTER FAILED`.
- `updated_at` - When the module was updated.
- `registered_version` - Semantic version of the registered module. Field is empty if a module is available but not registered.

Modules Mapping table

The Modules Mapping table contains Systems Manager parameter paths and the registered modules that output them.

Attributes:

- `param_name` - Systems Manager parameter path following the MCS deployment ID.
- `infrastructure` - Boolean value indicating if the parameter is output by MCS infrastructure.
- `module_pks` - List of registered modules that output param_name.

Enabled Modules table

The Enabled Modules table contains modules that have been enabled.

Attributes:

- `enabled_module_pk` - Primary key consisting of module_name, region_type, and module_region.
- `active_dependents` - List of enabled modules that are dependent on this module.
- `category` - Can be one of the following: `Network`, `Identity`, `WorkstationManagement`, `Storage`, `Custom`, or `SpokeRegionInfrastructure`.
- `creation_time` - Time that the module was created.
- `deployment_uuid` - Unique ID assigned when the Service Catalog product is provisioned.
- `input_parameters` - CloudFormation parameters.
- `last_update_time` - Time that the module was most recently updated.
- `module_name` - Name of the module.
- `module_region` - Region in which the module is enabled.
- `module_region_category` - Type of Region in which the module is enabled. Can be one of the following: `Hub` or `Spoke`.
- `module_version` - Version of the module enabled.
- `region_type` - Indicates whether the module can be enabled in the hub Region, a spoke Region, or both. Can be one of the following: `HUB`, `SPOKE`, or `BOTH`.
- `servicecatalog_provisioned_product_id` - Service Catalog provisioned product ID for the module enabled.

- status - Can be one of the following: ENABLED, ENABLING IN PROGRESS, ENABLE FAILED, DISABLED, DISABLING IN PROGRESS, or DISABLE FAILED.

Regions table

The Regions table consists of AWS Regions that can be enabled or disabled in MCS. The hub Region cannot be disabled.

Attributes:

- name - AWS Region name.
- date_enabled - Date that the Region was enabled.
- enablement_status - Can be one of the following: ENABLED, ENABLING IN PROGRESS, ENABLE FAILED, DISABLED, DISABLING IN PROGRESS, or DISABLE FAILED.
- is_hub - Boolean value indicating if the Region is the hub Region.
- provisioned_product_id - Service Catalog provisioned product ID.

Lock table

The Lock table is a distributed locking mechanism that prevents multiple processes from executing the same critical section simultaneously.

Attributes:

- lock_name - Name of the process currently being executed
- time_to_live - TTL timestamp for automatic cleanup

Note

DynamoDB automatically deletes items where time_to_live date had been expired, but the deletion does not occur instantly after expiration and it can take few days until deletion.
See Using time to live (TTL) in DynamoDB for more details

Plan your deployment

This section describes the cost, security, Regions and other considerations before deploying the solution.

Supported AWS Regions

MCS is available in the following AWS Regions:

Region name	
US East (Ohio)	Asia Pacific (Tokyo)
US East (N. Virginia)	Canada (Central)
US West (Northern California)	Europe (Frankfurt)
US West (Oregon)	Europe (Ireland)
Asia Pacific (Mumbai)	Europe (London)
Asia Pacific (Seoul)	Europe (Paris)
Asia Pacific (Singapore)	Europe (Stockholm)
Asia Pacific (Sydney)	South America (São Paulo)

Third-Party Modules may be available in different Regions. Refer to the module's manifest data to view its supported Regions.

Cost

You are responsible for the cost of the AWS services used while running this solution. As of this revision, the cost for running this solution with the default settings in the US East (N. Virginia) Region is approximately **$591.55 per month** when deploying the main stack, Managed VPC module, Managed Active Directory module, and FSx for Windows File Server module in the hub Region. These costs are for the resources shown in the Sample cost table.

Note

Third-Party modules' costs are not included in the monthly cost estimate, including Leostream workstation management modules and storage partner modules.

We recommend creating a budget through AWS Cost Explorer to help manage costs. Prices are subject to change. For full details, refer to the pricing webpage for each AWS service used in this solution.

Sample cost table

Total cost varies depending on how many modules and Regions you deploy. The following tables give a sample cost breakdown for deploying this solution and internal hub modules with the default parameters in the US East (N. Virginia) Region for one month.

MCS stack deployment

AWS service	Dimensions	Cost [USD]
Amazon API Gateway	First 333 million REST API calls per month	$ 3.50
Amazon Cognito	1,000 active users per month without the advanced security feature	$ 0.00
Amazon CloudFront	1,000,000 HTTPS requests	$ 1.00
Amazon S3	<1 GB storage for web assets and logging	$ 0.023

AWS Lambda	Modules = 5
	Requests = <1,000,000 = $ 0.20
	Enable module = 3,000 ms duration x + $ 0.0000000021 per ms
	$ 0.20 + (3,000 x $ 0.0000000021) x 5 = $0.2000315
		$ 0.20

AWS service	Dimensions	Cost [USD]
Systems Manager Parameter Store	Standard parameters and throughput	$0.00
Amazon DynamoDB	<1 GB storage, <1M write request units (WRUs) and read request units (RRUs)	$ 1.75
AWS Service Catalog	<1,000 API calls	$ 0.70
Amazon EventBridge Event Bus	AWS default service events	$ 0.00
Amazon EventBridge Pipe	<1M requests after filtering per month	$ 0.40
Amazon EventBridge API Destination	<1M requests per month	$ 0.20
Amazon Simple Queue Service	Standard queue with <1M requests per month	$ 0.00
AWS Step Functions	<4,000 state transitions AWS Free Tier	$ 0.00
Amazon CloudWatch	AWS Free Tier	$ 0.00
	Total:	**$ 7.77 [USD] / month**

Managed VPC module

AWS service	Dimensions	Cost [USD]
Amazon VPC	Public IPv4 address NAT Gateway cost is highly variable depending on modules deployed	$ 3.65
Systems Manager Parameter Store	Standard parameters and throughput	$ 0.00

AWS service	Dimensions	Cost [USD]
Amazon CloudWatch	AWS Free Tier	$ 0.00
	Total:	**$ 3.65 [USD] / month**

Managed Active Directory module

AWS service	Dimensions	Cost [USD]
AWS Directory Service	$0.12 per hour	$ 87.60
Systems Manager Parameter Store	Standard parameters and throughput	$ 0.00
AWS Secrets Manager	1 secret	$ 0.45
Amazon CloudWatch	AWS Free Tier	$ 0.00
EC2	t3.micro (5 minute deployment)	<$ 0.01
	Total:	**$ 88.05 [USD] / month**

FSx for Windows File Server module

AWS service	Dimensions	Cost [USD]
Amazon FSx for Windows File Server	256 GiB SSD storage capacity, 64 MBps throughput	$ 288.28
Systems Manager Parameter Store	Standard parameters and throughput	$ 0.00
Amazon CloudWatch	AWS Free Tier	$ 0.00

AWS service	Dimensions	Cost [USD]
	Total:	**$ 288.28 [USD] / month**

FSx for Lustre File Server module

AWS service	Dimensions	Cost [USD]
Amazon FSx for Lustre File Server	1,200 GiB SSD storage capacity, LZ4 Compression Disabled	$ 168.19
Systems Manager Parameter Store	Standard parameters and throughput	$ 0.00
Amazon CloudWatch	AWS Free Tier	$ 0.00
	Total:	**$ 168.19 [USD] / month**

Third-Party modules cost

This solution includes Third-Party Leostream workstation management modules available for deployment, and storage partner modules available for registration and deployment.

> **Note**
>
> Refer to the Leostream documentation or contact Leostream for more detailed and up-to-date Leostream module costs.
> Refer to individual Third-Party module support page or contact partners for their respective costs.

Here is a **simplified cost table** for core AWS services in the hub region for Leostream modules using default settings. Actual costs may vary depending on your configuration and chosen modules:

Leostream Broker module

AWS service	Dimensions	Cost [USD]
Amazon RDS	db.r6g.large (Aurora Postgresql)	$ 229.95
Application Load Balancer		$ 16.51
EC2	t3.large (min 2 by default): ($0.112 / hour) * 24 hour * 31 days * 2 = $166.66	$ 166.66
EC2	g4dn.xlarge with Windows OS ($0.71 / hour) * 24 hour * 31 days = $528.24	$ 528.24
EC2	g4dn.xlarge with Linux OS ($0.584 / hour) * 24 hour * 31 days = $434.50	$ 434.50
Route 53	Hosted Zone (per-request cost assumed to be negligible)	$ 0.50
	Total:	**$ 1376.36 [USD] / month**

Leostream Gateway module

AWS service	Dimensions	Cost [USD]
Application Load Balancer		$ 16.51
EC2	m5.xlarge (min 2 by default) with RHEL OS ($0.269 / hour) * 24 hour * 31 days *2 = $400.27	$ 400.27

AWS service	Dimensions	Cost [USD]
AWS Global Accelerator	Standard	$ 18
AWS Global Accelerator Data Transfer	Varies depending on regions used	~ $ 0.015 per GB
EC2 Egress	First 100 GB per month is free	~ $ 0.09 per GB
EC2 Elastic IP Address	2 used by Global Accelerator	$ 7.32
	Total (excluding data transfer costs):	**$ 442.10 [USD] / month**

Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared responsibility model reduces your operational burden because AWS operates, manages, and controls the components including the host operating system, the virtualization layer, and the physical security of the facilities in which the services operate. For more information about AWS security, visit AWS Cloud Security.

IAM roles

This solution creates IAM roles that grant the solution's Lambda functions access to create Regional resources. These Lambda functions are invoked when:

- The solution creates custom resources during stack deployments
- The MCS API is called
- AWS Step Functions run when registering and de-registering modules

A stack set execution IAM role is required to provision and terminate Service Catalog products when enabling and disabling modules. This role has PowerUserAccess, allowing it to create and update IAM roles as needed for modules.

Amazon CloudFront

This solution deploys a web console hosted in an S3 bucket. To help reduce latency and improve security, this solution includes a CloudFront distribution with an origin access identity, which is a CloudFront user that provides public access to the solution's website bucket contents. For more information, see Restricting Access to Amazon S3 Content by Using an Origin Access Identity in the *Amazon CloudFront Developer Guide*.

CloudFront and API Gateway minimum TLS version

The solution uses a default CloudFront domain, which sets the minimum allowed TLS version to v1.0 by default. For enhanced security, we recommend to configuring the minimum TLS version to v1.2. To achieve this, you must set up a custom CloudFront domain. Follow the instructions provided in Set up a custom CloudFront domain in the *Amazon CloudFront Developer Guide*.

The solution also uses a default API Gateway domain, which sets the minimum allowed TLS version to v1.0 by default. For more information, see Choose a security policy for your REST API custom domain in API Gateway in the *Amazon API Gateway Developer Guide*.

Security groups

The solution creates security groups designed to control and isolate network traffic between the module resources and the VPC created or imported in the Network modules.

We recommend that you review the security groups and further restrict access as needed after deployment. See Control traffic to your AWS resources using security groups for more information.

The following modules create security groups to allow traffic to/from the VPC:

- **Managed Active Directory module -** Allow the default virtual private network (VPN) Domain Name System (DNS) to resolve names from Microsoft Active Directory
- **Leostream Broker module -** Environment configuration and AMI pipelines
- **Leostream Gateway module -** Automation and Application Load Balancers
- **FSx for Windows File Server module -** FSx file system

Secrets Manager

AWS Secrets Manager securely stores and manages sensitive credentials generated by MCS modules. This service provides automatic encryption, access control, and audit logging for all stored secrets.

Secrets created by modules

The following modules automatically create and manage secrets in AWS Secrets Manager:

Identity Module (AWS Managed Microsoft AD)

- StudioAdmin user credentials - Default admin user for end-user access
- SA_AdConnectorUser credentials - Service account for cross-region AD communication
- SA_McsModulesUser credentials - General service account for module integrations

Leostream Broker Module

- API service user credentials - Authentication for Leostream API operations
- Amazon RDS database credentials - Database connection credentials for the Leostream broker

Password Management

- This solution does not provide automatic secrets rotation. Depending on your security requirements, you may consider manually rotating the credentials for your Leostream Connection Broker database.
- AWS Managed Microsoft AD passwords expire every 90 days and require manual rotation.
- Follow the steps in Password Rotation to update passwords across all dependent services.

Security.txt

The solution does not include a `security.txt` file in the website files. This file is intended to provide information about the owner or operator of a publicly accessible website, such as security contacts and responsible disclosure policies.

Since the Modular Cloud Studio on AWS website is a private, login-protected application that you control, a `security.txt` file isn’t necessary or applicable. The frontend application is only accessible to authorized users of your organization, so there is no need to publicly disclose security information.

If you have specific security or responsible disclosure needs for your Modular Cloud Studio on AWS deployment, we recommend managing that information separately from the frontend application. This solution is designed to provide you the flexibility to configure and extend it as needed for your specific requirements.

Denial-of-service protections

The API exposed by the solution has throttling settings configured to limit requests. The maximum number of requests per second is set to 50, with a burst rate of 10 requests. This helps protect the API from abuse or unintended high traffic. For more details on the API throttling configuration, see Throttle requests to your REST APIs for better throughput in API Gateway in the *Amazon API Gateway Developer Guide*.

AWS Global Accelerator is protected by AWS Shield Standard by default. This means AWS Shield automatically enforces rate limiting on resources the Global Accelerator sends traffic to. For more details, see AWS Shield mitigation logic for AWS Global Accelerator standard accelerators in the *AWS WAF Developer Guide*.

Configuring Amazon EBS snapshot encryption

Before deploying the solution, you must configure your AWS account to encrypt Amazon Elastic Block Store (Amazon EBS) snapshots automatically. This helps ensure that all Amazon EBS snapshots created during the process of building the Leostream AMIs are encrypted for enhanced security and compliance.

For detailed instructions on how to enable default encryption for Amazon EBS snapshots in your account, see Encrypt EBS snapshots by default in the *Amazon EBS User Guide*.

Leostream database user

When you deploy the solution, the Leostream Broker module creates and then connects to a dedicated Amazon RDS database cluster. The Leostream Broker process uses the default `postgres` database user to access this Amazon RDS cluster.

⚠ **Important**

The default `postgres` user has superuser privileges, which grants it full administrative access to the database.

We recommend reviewing your security and compliance requirements to determine if using the default `postgres` superuser account is appropriate for your environment. This database is only used by the Leostream Broker, and many actions a superuser can normally take against a PostgreSQL database aren't possible in a managed database on Amazon RDS.

API Gateway Security

The API Gateway used in this solution defaults to allowing TLS 1.0 and above. For enhanced security, we recommend configuring a custom domain with a higher minimum TLS version. See the Enhanced TLS Security section in the "Use the solution" chapter for guidance on setting up a custom domain with TLS 1.2+.

Content Security Policy

The solution deploys a CloudFront Distribution with preset Content Security Policies. One of these policies has the value `https://*.amazonaws.com`, which is used to connect with the solution's resources. If this policy grants broader permissions than required for your use case, consider restricting access to specific domains by configuring CloudFront distribution settings through the AWS Management Console.

Quotas

Service quotas, also referred to as limits, are the maximum number of service resources or operations for your AWS account.

Quotas for AWS services in this solution

Make sure you have sufficient quota for each of the services implemented in this solution. For more information, see AWS service quotas.

To view the service quotas for all AWS services in the documentation without switching pages, view the information in the Service endpoints and quotas page in the PDF instead.

AWS CloudFormation quotas

Your AWS account has CloudFormation quotas that you should be aware of when launching the stack in this solution. By understanding these quotas, you can avoid limitation errors that would prevent you from deploying this solution successfully. For more information, see AWS CloudFormation quotas in the in the *AWS CloudFormation User's Guide*.

Deploy the solution

This solution uses AWS CloudFormation templates and stacks to automate its deployment. The CloudFormation template specifies the AWS resources included in this solution and their properties. The CloudFormation stack provisions the resources described in the template.

Deployment process overview

Follow the step-by-step instructions to configure and deploy the solution into your account.

Before you launch the solution, review the the section called “Cost”, architecture overview, security, and other considerations discussed in this guide.

Time to deploy: Approximately 7-10 minutes to deploy the MCS solution stack.

> ⚠ **Important**
>
> This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. AWS owns the data gathered though this survey. Data collection is subject to the AWS Privacy Notice.
> To opt out of this feature, download the template, modify the AWS CloudFormation mapping section, and then use the AWS CloudFormation console to upload your updated template and deploy the solution. For more information, see the Anonymized data collection section of this guide.

AWS CloudFormation Template

You can download the CloudFormation template for this solution before deploying it.

View template

ModularCloudStudioOnAwsStack.template - Use this template to launch the solution and all associated components. The default configuration deploys the core and supporting solutions found in the AWS services in this solution section, but you can customize the template to meet your specific needs.

> **Note**
>
> CloudFormation resources are created from AWS CDK constructs.

This AWS CloudFormation template deploys Modular Cloud Studio in the AWS Cloud.

Launch the stack

Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

If you previously deployed MCS in the same account, confirm that your previous stack uninstalled successfully. Follow the steps described in the Uninstall the solution and Troubleshooting and sections to disable the MCS modules and de-register the Third-Party modules.

Time to deploy: Approximately 7-10 minutes

1. Sign into AWS Management Console and select the button to launch `ModularCloudStudioOnAwsStack.template` CloudFormation template.

   Launch solution

2. The template launches in the US East (N. Virginia) Region by default. To launch the solution in a different Region, use the Region selector in the console navigation bar.

   > **Note**
   >
   > This solution is not currently available in all AWS Regions. You must launch this solution in an AWS Region where the solution is available. See Supported AWS Regions for more information.

3. On the **Create stack** page, verify that the correct template URL is in the **Amazon S3 URL** text box and choose **Next.**
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, see IAM and AWS STS quotas in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
AdminEmail	`<Requires input>`	The admin email address to use for authorization to access the MCS web console or receive the email that contains an URL to the Leostream Broker admin secret.

6. Select **Next**
7. On the **Configure stack options** page, ensure that 10 or fewer tags are configured. You can auto-apply these main solution stack tags to all the modules. You can also review and remove them when you deploy each module.
8. Choose **Next**.
9. On the **Review** page, review and confirm the settings. Select the box acknowledging that the template creates IAM resources.
10. Choose **Next**.
11. Choose Submit to deploy the stack. You can view the status of the stack in the AWS CloudFormation console in the Status column. You should receive a CREATE_COMPLETE status in approximately 7 minutes.
12. After the stack in CloudFormation is successfully created, select the stack that you created. Then navigate to the **Outputs** tab and find the MCS web console URL defined in **CloudFrontURL**.
13. You will receive an email with a temporary password to access the MCS web console. You must reset the password on your first login following the prompt. If you haven't received an email after the stack deployment is completed, check your spam folder.
14. You can add MCS web console users by navigating to the MCS user pool in the Amazon Cognito console, and clicking **Create user** under Users from the left navigation.

Use the solution

This section provides a user guide for using the AWS solution, including Region enablement, module enablement, module library, and password rotation.

Region enablement

This section is optional and only required if you want to set up a multi-Region deployment. The solution initially deploys in a single hub Region. All additional AWS Regions enabled through this workflow become spoke Regions that can connect back to the hub Region for centralized management.

Enable a Region

1. Navigate to the MCS web console (see Launch the stack for details).
2. Select **AWS Regions** from the left navigation pane.
3. Choose **Add Region** in the top right corner.
4. Choose the spoke Region that you want to enable from the dropdown menu.
5. Choose **Enable**.
6. The Region appears in **AWS Regions Enabled** with an `ENABLING IN PROGRESS` status.

Enabling the spoke Region takes approximately 5 minutes. After the process completes, the Region status changes to `ENABLED`.

> **Note**
>
> Spoke Regions inherit the same tags as the main Solution stack.

After enabling a spoke Region, you can deploy modules to it by following the detailed steps in Module enablement and choosing the spoke Region during module deployment.

The Enable Region API does the following:

1. Checks if the Region exists in the Regions DynamoDB table.
2. Updates Region status to `ENABLING IN PROGRESS` in the Regions DynamoDB table.

3. Provisions the spoke Region infrastructure Service Catalog product.
4. If the previous step is successful, updates the Region `provisioned_product_id` attribute and status to `ENABLED` in the Regions DynamoDB table. Otherwise, sets the status to `ENABLE FAILED`.

Disable a Region

Prerequisites

Before disabling a spoke Region, you must first disable all modules deployed in that Region. The system will prevent Region disabling if any modules are still enabled.

To disable modules, see Disable a module for detailed instructions on how to disable each module in the Region.

Steps to disable a Region

1. Navigate to the MCS web console (see Launch the stack for details).
2. Select **AWS Regions** from the left navigation pane.
3. Choose an enabled spoke Region to disable, and choose **Disable Region**.
4. Choose **Confirm**.
5. The Region appears in **AWS Regions Enabled** with a `DISABLING IN PROGRESS` status.

Disabling the spoke Region takes approximately 5 minutes. After the process completes, the Region becomes available for re-enablement if needed.

The Disable Region API does the following:

1. Checks if the Region exists in the Regions DynamoDB table.
2. Checks if there are modules enabled in the Region in the Enabled Modules DynamoDB table. If so, an error is thrown.
3. Updates Region status to `DISABLING IN PROGRESS` in the Regions DynamoDB table.
4. Terminates the provisioned Service Catalog product.
5. If the previous step is successful, updates the Region `provisioned_product_id` attribute to be empty and status to `DISABLED` in the Regions DynamoDB table. Otherwise, sets the status to `DISABLE FAILED`.

Module enablement

Module enablement is the core functionality of Modular Cloud Studio on AWS, allowing you to deploy and manage infrastructure components across your AWS environment. Modules are pre-configured infrastructure templates that provide specific capabilities such as networking, identity management, storage, and workstation management.

Modules can be deployed to supported Regions that are enabled and may have dependencies on other modules. For example, most modules require a Network module to be deployed first to provide the underlying VPC infrastructure. The solution automatically validates dependencies and guides you through the deployment process.

Module deployment typically involves selecting the target Region, configuring module-specific parameters, applying tags, and reviewing the configuration before deployment. The deployment process uses AWS Service Catalog to provision resources consistently and securely.

To set up a complete Modular Cloud Studio environment, follow these deployment steps in order. Note that some steps are optional depending on your specific requirements:

Step 1. Enable Network modules

Step 2. Enable Identity modules

Step 3. Enable Leostream Broker module

Step 4. Enable Leostream Gateway module

Step 5. Enable Storage modules (Optional)

Step 6. Manual configurations

Enable a module

1. Navigate to the MCS web console (see Launch the stack for details).
2. In the left navigation pane, choose a module category: **Network**, **Identity**, **Workstation Management**, **Storage** or **Custom**
3. Select **Deploy New Module**.
4. Select a Region from the **Select Region** dropdown menu.

5. Select a module from the **Select *<Module Category>* module** dropdown menu. Some module categories have a **Create** or **Import** option. **Create** means that new resources are created when the module is enabled. **Import** means that the resources already exist and the user will provide necessary inputs for the module.
6. Select **Next**.
7. Configure the module settings if the parameter fields are populated.
8. Select **Next**.
9. Optionally **Tag** the resources that will be deployed by the module. This allows you to assign metadata to your resources, including custom resources created by the solution. Each tag is a label consisting of a user-defined key and value pair, and you can add up to 10 tags per module. For any imported module, the tags will only be applied to resources created by MCS, and they will not be applied to imported resources originally created outside of MCS. For example, the tags won't apply to VPC resources from the Import Amazon VPC module.
10. Select **Next**.
11. Review module settings.
12. Select **Deploy** module.
13. The module appears in **Module deployments** with an `ENABLING IN PROGRESS` status.

The Enable Module API does the following:

1. Checks if the Region is enabled.
2. Checks if the module is registered.
3. Checks if module dependencies are enabled.
4. Updates the module status to `ENABLING IN PROGRESS` in the Registered Modules DynamoDB table and Enabled Modules DynamoDB table.
5. Provisions the Service Catalog Product.
6. Updates the module `servicecatalog_provisioned_product_id` attribute in the Enabled Modules DynamoDB table.
7. Updates the `active_dependents` attribute on module dependencies in the Enabled Modules DynamoDB table.
8. If step 7 is successful, updates module status to `ENABLED` in the Enabled Modules DynamoDB table. If unsuccessful, updates status to `ENABLE FAILED`.

9. Updates module status to REGISTERED in the Registered Modules DynamoDB table.

Disable a module

Prerequisites

Before disabling a module, you must ensure that dependent modules are disabled first. The system will prevent you from disabling a module if other modules still depend on it, so you must work backwards through the dependency chain. For example, you need to disable Workstation Management modules before disabling the Identity modules they depend on.

> ⚠ **Important**
>
> Verify that all important data has been properly backed up before proceeding. For storage modules, ensure data has been backed up or migrated. For workstation management modules, confirm that user data and configurations have been saved. Disabling a module will terminate all associated AWS resources, and any data stored on those resources may be permanently lost.

Steps to disable a module

1. Navigate to the MCS web console (see Launch the stack for details).
2. In left the navigation pane, choose a module category: **Network**, **Identity**, **Workstation Management**, **Storage** or **Custom**.
3. Choose a module to disable, and select **Disable**.
4. Select **OK**.
5. The module appears in **Module deployments** with a `DISABLING IN PROGRESS` status.

The Disable Module API does the following:

1. Checks if any dependent modules are enabled. If so, an error is thrown.
2. Updates the module status to `DISABLING IN PROGRESS` in the Registered Modules DynamoDB table and Enabled Modules DynamoDB table.
3. Terminates the provisioned Service Catalog product.

4. If the previous step is successful, updates the `active_dependents` attribute on module dependencies and updates module status to `DISABLED` in the Enabled Modules DynamoDB table. If unsuccessful, updates status to `DISABLE FAILED` and doesn't update `active_dependents` on module dependencies.
5. Updates module status to `REGISTERED` in the Registered Modules DynamoDB table.

Step 1: Enable Network modules

Follow these steps to enable the Network modules.

1. Navigate to the MCS web console (see Launch the stack for details).
2. Select **Network** from the left navigation pane.
3. Choose **Deploy New Module**.
4. Based on your use cases, follow the steps in Create Amazon VPC for generating a new VPC, or follow the steps in Import Amazon VPC for importing the existing VPC by providing the required attributes.

Option 1.a: Create Amazon VPC

1. For **Select Region**, select the Region where you want the VPC to be created. There should be only one hub Region option if you haven't deployed any spoke Regions.
2. For **Select Network** module, select Create Amazon VPC and choose **Next**.
3. For **Configure VPC settings**, review the parameters for this module and modify them as necessary. This module uses the following default values.

Parameter	Default	Description

Availability Zones	`<Region>a,`
	`<Region>b`
		(Select 2) List of Availability Zones to use for the subnets in the VPC. The logical order is preserved.

VPC CIDR	`10.0.0.0/16`	CIDR block for the VPC.

Private Subnet CIDR List	`10.0.0.0/19,`
	`10.0.32.0/19`
		Comma delimited list of CIDR blocks for private subnets 1 and 2, located in Availability Zones 1 and 2, respectively.

Parameter	Default	Description
		Note: CIDR ranges in each Region must not overlap. The default values provided don't overlap with each other, and are within the default VPC CIDR range provided.
Public Subnet CIDR List	`10.0.128.0/20, 10.0.144.0/20`	Comma delimited list of CIDR blocks for public subnets 1 and 2, located in Availability Zones 1 and 2, respectively. **Note:** CIDR ranges in each Region must not overlap. The default values provided don't overlap with each other, and are within the default VPC CIDR range provided.
Enable VPC Flow Logs	`true`	Set to `true` to create VPC flow logs for the VPC and publish them to CloudWatch. If you set it to `false`, the VPC flow logs won't be created.
VPC Flow Logs Traffic Type	`REJECT`	The type of traffic to log. You can log traffic that the resource accepts (`ACCEPT`) or rejects (`REJECT`), or `ALL` Traffic.

4. For **Configure Tag Settings**, review the tags for this module and modify them as necessary. By default, this module uses tags defined in the main solution stack.

5. Choose **Next**.

6. On the **Review** page, verify all the parameters that you provided and choose **Deploy Module** if you confirm that they are correct.

7. The status of the network module shows as **Enabling in progress**. The deployment of this module takes approximately five minutes. After the deployment is complete, the status of the network module shows as **Enabled**.

Option 1.b: Import Amazon VPC

Pre-deployment requirements

1. Availability Zones
    a. Ensure that selected Availability Zones host the necessary default instance types used by MCS:
        i. t3.large, m5.xlarge, g4dn.xlarge, r6g.large
        ii. If other instance types are desired, ensure that they are available in the VPC's Availability Zones
    b. Use AWS CLI command to verify, e.g. for us-east-1:

```
$ aws ec2 describe-instance-type-offerings \
  --location-type availability-zone \
  --filters Name=instance-type,Values=t3.large,m5.xlarge,g4dn.xlarge,r6g.large \
  --region us-east-1 \
  --query 'InstanceTypeOfferings[].Location' \
  --output text | tr '\t' '\n' | sort | uniq
```

2. Subnet Configuration
    a. At least 2 public subnets across different Availability Zones which will be used by MCS
    b. At least 2 private subnets across different Availability Zones which will be used by MCS
3. Internet Connectivity
    a. Public subnets must have route tables with routes to an Internet Gateway (IGW)
    b. Private subnets must have route tables with routes to NAT Gateways (NGW)
4. Required VPC Endpoints
    a. Interface Endpoints
        i. com.amazonaws.[region].ssm
        ii. com.amazonaws.[region].ssmmessages
        iii. com.amazonaws.[region].ec2
        iv. com.amazonaws.[region].ec2messages
    b. Gateway Endpoint
        i. com.amazonaws.[region].s3

Note

All endpoints must be associated with the private subnets where MCS workloads will run. If the Endpoint already exists, this requires that you navigate to that endpoint's configuration page, select "Manage Subnets", and ensure that the endpoint is associated with the private subnets that you will provide to MCS. If the Endpoint does not already exist, ensure that during creation of the endpoint, that the subnets that you will provide to MCS are selected. In addition, the security group associated with these endpoints must be configured to allow all traffic from the VPC CIDR source.

VPC Peering must be configured between hub and spoke VPCs. For more information, see Work with VPC peering connections. Ensure that the route tables are configured correctly for the VPC peering connection. For more information, see Update your route tables for a VPC peering connection.

Validation Testing

Before proceeding with the MCS Unmanaged VPC Module deployment, validate your VPC configuration:

1. Launch an Amazon Linux 2023 instance in one of the private subnets
2. Ensure the instance has an IAM Role with `AmazonSSMManagedInstanceCore` permissions
3. Attempt to connect to the instance using AWS Systems Manager Session Manager
4. If you see "Instance is not connected to Session Manager" or the "Connect" button is disabled, troubleshoot your VPC endpoint configuration, network routes, and security groups.
5. A successful connection confirms proper network configuration.

Deploying the MCS Unmanaged VPC Module

1. For **Select Region**, select the Region where you want the VPC to be imported from. There should be only one hub Region option if you have not deployed any spoke Regions.

> **Note**
>
> The VPC must exist in the same account and Region where the Network module is being enabled.

2. For **Select Network** module, select **Import Amazon VPC** and choose **Next**.
3. For **Configure VPC settings**, review the parameters for this module and modify them as necessary. This module uses the following default values.

Parameter	Default	Description	Notes
VPC ID	*<Requires input>*	Identifier of the existing VPC.	
VPC CIDR	*<Requires input>*	VPC CIDR block.	
Private Subnet IDs	*<Requires input>*	Exactly two comma separated Subnet IDs for the private subnets.	
Public Subnet IDs	*<Requires input>*	Exactly two comma separated Subnet IDs for the public subnets.	
Private Subnet Route Table IDs	*<Requires input>*	Exactly two comma separated Route table IDs for private subnets.	See Notes below
Public Subnet Route Table IDs	*<Requires input>*	Exactly two comma separated Route table IDs for public subnets.	See Notes below

Parameter	Default	Description	Notes
Availability Zones	*<Requires input>*	(Select 2) List of Availability Zones to use for the subnets in the VPC. The logical order is preserved.	

> **Note**
>
> If there is only one Route Table available, you can duplicate the entry. MCS expects exactly two comma delimited values to be provided. For example:
>
> ```
> rtb-prv123456,rtb-prv123456
> ```

4. For **Configure Tag Settings**, review the tags for this module and modify them as necessary. By default, this module uses tags defined in the main solution stack.
5. Choose **Next**.
6. On the **Review** page, verify all the parameters that you provided. If they are correct, choose **Deploy Module**.
7. The status of the network module shows as **Enabling in progress**. The deployment of this module takes approximately five minutes. After the deployment is complete, the status of the network module shows as **Enabled**.

Step 2: Enable Identity modules

Follow these steps to enable the Identity module.

1. Navigate to the MCS web console (see Launch the stack for details).
2. Select **Identity** from the left navigation pane.
3. Choose **Deploy New Module**.
4. Based on your use cases, follow the steps in Create AWS Managed Microsoft Active Directory for creating a new AWS Directory Service instance, or follow the steps in Import Custom Microsoft Active Directory to import an existing Active Directory by providing the required attributes.

Option 2.a: Create AWS Managed Microsoft Active Directory

1. For **Select Region**, select the Region where you want the Directory Service to be created. There should be only one hub Region option if you have not deployed any spoke Regions.
2. For **Select Identity** module, select **Create AWS Managed Microsoft Active Directory** and choose **Next**.
3. For **Configure AD settings**, review the parameters for this module and modify them as necessary. This module uses the following default values.

Parameter	Default	Description
Domain name	studio.mcs.internal	Domain name for the AWS Managed Microsoft AD.

4. For **Configure Tag Settings**, review the tags for this module and modify them as necessary. By default, this module uses tags defined in the main solution stack.
5. For **Review and deploy module**, choose **Deploy Module**.
6. The status of the Identity module shows as **Enabling in progress**. The deployment of this module takes approximately 30 minutes. After the deployment is complete, the status of the Identity module shows as **Enabled**.
7. An AWS Managed Microsoft AD will be created under Standard Edition using `mad.mcs.int` as the DNS name. To retrieve the StudioAdmin credentials, navigate to the AWS Secrets Manager console and locate the secret at `/[MCSDeploymentId]/Identity/StudioAdminActiveDirectoryLoginCredentials`. Select the **Overview** tab and click the **Retrieve secret value** button to display both the StudioAdmin username and password. Alternatively, you can access the credentials directly by clicking the **View** button on the MCS Web UI and following the direct link to the secret.

> **Note**
>
> When modifying the StudioAdmin password through AWS Directory Service console, ensure you manually update the corresponding secret in AWS Secrets Manager to maintain synchronization. Follow the steps to reset the user password.

8. Sign in to the AWS Directory Service console, and follow the steps for Creating an AWS Managed Microsoft AD user if additional users are needed.

⚠ **Important**

In addition to the StudioAdmin user, three additional users are created by the managed AD module:

1. **Admin**
   - Required user created by the directory service
   - Password location in Secret Manager: `/[MCSDeploymentId]/Identity/DefaultAdminActiveDirectoryLoginCredentials`
2. **SA_AdConnectorUser**
   - Created by the MCS Managed AD module
   - Service account used by AD Connectors in the spoke regions
   - Password location in Secret Manager: `/[MCSDeploymentId]/Identity/AdConnectorServiceAccountActiveDirectoryLoginCredentials`
   - Follow the steps in Password Rotation to update the password
3. **SA_McsModulesUser**
   - Created by the MCS Managed AD module
   - Service account used by modules for AD configuration setup
   - Password location in Secret Manager: `/[MCSDeploymentId]/Identity/McsModulesServiceAccountActiveDirectoryLoginCredentials`
   - Follow the steps in Password Rotation to update the password

Option 2.b: Import Custom Microsoft Active Directory

Pre-deployment requirements

1. DNS Resolver Security Group
   a. Ensure a security group exists for the Route 53 resolver endpoint in your target VPC
   b. Verify the security group has two outbound rules configured to allow DNS traffic:
      i. Type: DNS (TCP), Destination: 0.0.0.0/0, Port: 53
      ii. Type: DNS (UDP), Destination: 0.0.0.0/0, Port: 53
2. Route 53 Outbound Endpoint

  a. Ensure a Route 53 outbound endpoint exists in the VPC where your Active Directory domain controllers are located
  b. Verify the endpoint is configured with appropriate IP addresses in private subnets across different Availability Zones
  c. Confirm the endpoint is associated with the DNS Resolver Security Group

3. Route 53 Resolver Rule
   a. Ensure a resolver rule exists for your Active Directory domain name
   b. Verify the rule is associated with the target VPC and the outbound endpoint
   c. Confirm the rule forwards DNS queries to your Active Directory domain controllers

Deploying the MCS Unmanaged Active Directory Module

1. For **Select Region**, select the Region where you want the Directory Service to be created. There should be only one hub Region option if you have not deployed any spoke Regions.
2. For **Select Identity module**, select **Import Custom Microsoft Active Directory** and choose **Next.**
3. For **Configure AD settings**, review the parameters for this module and modify them as necessary. This module uses the following default values.

Parameter	Default	Description
Domain Name	*<_Requires input_>*	The domain name of MCS unmanaged Active Directory module.
IP Address1	*<_Requires input_>*	The first IP address of MCS unmanaged Active Directory module.
IP Address2	*<_Requires input_>*	The second IP address of MCS unmanaged Active Directory module.
Region	*<_Requires input_>*	The Region where the existing directory resides.

4. For **Configure Tag Settings**, review the tags for this module and modify them as necessary. By default, this module uses tags defined in the main solution stack.
5. Choose **Next**.

6. On the **Review** page, verify all the parameters that you provided and choose **Deploy Module** if you confirm that they are correct.
7. The status of the Identity module shows as **Enabling in progress**. The deployment of this module takes approximately five minutes. After the deployment is complete, the status of the network module shows as **Enabled.**
8. Required manual configuration: navigate to `/[MCSDeploymentId]/Identity/McsModulesServiceAccountActiveDirectoryLoginCredentials` in the secret manager, update the credentials with your Active Directory service by replacing the username and password fields.

> ⚠ **Important**
>
> The service account is essential for MCS modules configuration, such as Amazon FSx for Windows and Leostream broker module. Failed to update the credentials before deployment will cause module deployment failure and prevent proper service configuration.

Step 3: Enable Leostream Broker module

Follow these steps to enable the Leostream Broker module.

> ⓘ **Note**
>
> Modular Cloud Studio on AWS allows you to deploy and manage a scalable, secure, and global content production infrastructure in the cloud. This includes custom modules, developed by AWS Partners or other third parties, that you can choose to use ("Third-Party Modules"). AWS does not own or otherwise have any control over Third-Party Modules. Your use of the Third-Party Modules is governed by any terms provided to you by the Third-Party Module providers when you acquired your license to use them (for example, their terms of service, license agreement, acceptable use policy, and privacy policy). You are responsible for ensuring that your use of the Third-Party Modules comply with any terms governing them, and any laws, rules, regulations, policies, or standards that apply to you. You are also responsible for making your own independent assessment of the Third-Party Modules that you use. AWS does not make any representations, warranties, or guarantees regarding the Third-Party Modules, which are "Third-Party Content" under your agreement

with AWS. Modular Cloud Studio on AWS is offered to you as "AWS Content" under your agreement with AWS.

When you use MCS to deploy the Leostream Broker module, a 30-day trial license is automatically provided. During this trial period, you might see an `Invalid License` message upon logging in to the Leostream Connection Broker. However, you can inspect the remaining days of the trial within the Connection Broker interface. To continue using the Leostream Broker module beyond the 30-day trial period, you must contact Leostream directly to obtain a full license, then update the license key.

Note

Make sure your account has access to use the *g4dn.xlarge* EC2 instance type if you want to use Windows or Linux workstation AMI. Otherwise, the deployment will fail. See service quotas for more details.

1. Navigate to the MCS web console (see Launch the stack for details).
2. Select **Workstation Management** from the left navigation pane.
3. Choose **Deploy New Module**.
4. For **Select Region**, select the Region where you want the Leostream Broker module. There should be only one hub Region option if you have not deployed any spoke Regions.
5. For **Select Workstation Management module**, select **Leostream Broker** and choose **Next**.
6. For **Configure workstation management settings**, review the parameters for this module and modify them as necessary. This module uses the following default values.

Parameter	Default	Description
Leostream Broker Fully Qualified Domain Name (optional)	*Optional input*	Specify the FQDN that will be routed to the broker load balancer. If no accompanying Certificate ID is supplied, a self-signed certificate will be generated for this domain. (This parameter is required if you specified a Certificate ID).

Parameter	Default	Description
Leostream Broker Certificate ID (optional)	*Optional input*	Specify the Certificate ID or ARN imported from AWS Certificate Manager to validate your FQDN. If you leave this field blank, the module will create a self-signed certificate from the domain you previously provided. **Note** See Get certificates ready in AWS Certificate Manager for more information on how to set up a certificate.
Leostream License Contact Email	*<_Requires input_>*	Contact email to use for the Leostream license. The free Leostream Broker license included in this module will be registered under this email.
Leostream License Contact Name	*<_Requires input_>*	Contact name to use for the Leostream license. The free Leostream Broker license included in this module will be registered under this name.
Leostream Broker Package Location	https://s3.amazonaws.com/downloads.leostream.com/leostream-broker-2024.1.7-1.x86_64.rpm	Amazon S3 download URL for the Leostream Broker RPM package.
Leostream Broker Max Instances Count	5	The maximum amount of Leostream Broker instances that the Auto Scaling Group can scale up to.

Parameter	Default	Description
Workstation Provision Threshold	1	Start provisioning new workstations if the number of available workstations is less than this threshold. The Leostream Broker will not provision if the number of workstations reaches the set maximum count. **Note** The initial value must be an integer of 1 or greater. If you need a different value later, you can open the Leostream console and change the value to any non-negative integer, including 0.
Workstation Max Count	2	Maximum number of workstations to be provisioned by the Leostream Broker. This number applies for each Windows and Linux pool.
Workstation Windows2022 AMI	`Yes`	Select if you want to deploy the Windows AMI.
Amazon DCV Windows Server URL	`https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/Servers/nice-dcv-server-x64-Release-2024.0-18131.msi`	URL to download the Amazon DCV Windows server file.

Parameter	Default	Description
Leostream Windows Agent URL	https://downloads.leostream.com/LeostreamAgentSetup2024-1-4-0.exe	URL to download the Leostream Windows agent file.
Workstation Rocky Linux8 AMI	No	Select if you want to deploy the Linux AMI. If you select Yes, ensure that you have subscribed Rocky Linux 8 on the AWS Marketplace.
Amazon DCV Linux Server URL	https://d1uj6qtbmh3dt5.cloudfront.net/2024.0/Servers/nice-dcv-2024.0-18131-el8-x86_64.tgz	URL to download the Amazon DCV Linux server file.
Leostream Linux Agent URL	https://downloads.leostream.com/LeostreamAgentJava-5.3.18.0.jar	URL to download the Leostream Linux agent file.

7. For **Configure Tag Settings**, review the tags for this module and modify them as necessary. By default, this module uses tags defined in the main solution stack.

8. Choose **Next**.

9. On the **Review** page, verify all the parameters you provided and choose **Deploy Module** if you confirm they are correct.

10. The status of the Leostream Broker will be shown as **Enabling in progress**. The deployment of this module takes approximately 1 hour. If you selected Yes on either **Workstation Windows 2022 AMI** or **Workstation Rocky Linux 8 AMI**, the deployment might take up to 3 hours. After the deployment is complete, the status of the storage module will be shown as **Enabled**.

11. Leostream broker's local **Admin** user is created for managing the application. To retrieve the Leostream local **Admin** credentials, you can sign in to the AWS Secrets Manager console, and select the secret: `/[MCSDeploymentID]/WorkstationManagement/Leostream/Console/AdminUserCredentials`. Choose the **Overview** tab, then choose the **Retrieve secret value** button to display the user login and password. Alternatively, you can access the credentials directly by clicking the **View** button on the MCS Web UI and following the direct link to the secret.

12. Modular Cloud Studio on AWS automatically configures Leostream Broker internal resources during the deployment process of this module. The updated resources include:
    - Remote Authentication Servers
    - AWS Center
    - EC2 Workstation Pools
    - Policies
    - Power Control Plans and Release Plans

Step 4: Enable Leostream Gateway module

Follow these steps to enable the Leostream Broker module.

> **Note**
>
> Modular Cloud Studio on AWS allows you to deploy and manage a scalable, secure, and global content production infrastructure in the cloud. This includes custom modules, developed by AWS Partners or other third parties, that you can choose to use ("Third-Party Modules"). AWS does not own or otherwise have any control over Third-Party Modules.

Your use of the Third-Party Modules is governed by any terms provided to you by the Third-Party Module providers when you acquired your license to use them (for example, their terms of service, license agreement, acceptable use policy, and privacy policy). You are responsible for ensuring that your use of the Third-Party Modules comply with any terms governing them, and any laws, rules, regulations, policies, or standards that apply to you. You are also responsible for making your own independent assessment of the Third-Party Modules that you use. AWS does not make any representations, warranties, or guarantees regarding the Third-Party Modules, which are "Third-Party Content" under your agreement with AWS. Modular Cloud Studio on AWS is offered to you as "AWS Content" under your agreement with AWS.

1. Navigate to the MCS web console (see Launch the stack for details).
2. Select **Workstation Management** from the left navigation pane.
3. Choose **Deploy New Module**.
4. For **Select Region**, select the Region where you want the Leostream Broker module. There should be only one hub Region option if you have not deployed any spoke Regions.
5. For **Select Workstation Management module**, select **Gateway with Amazon DCV**, and choose **Next**.
6. For **Configure workstation management settings**, review the parameters for this module and modify them as necessary. This module uses the following default values.

Parameter	Default	Description
Fully Qualified Domain Name (optional)	*Optional input*	Specify the FQDN that will be routed to the gateways to access the connection broker and workstations. (This parameter is required if you specified a **Certificate ID** or **Route 53 Hosted Zone ID**).
Certificate ID (optional)	*Optional input*	Specify the Certificate ID or ARN imported from AWS Certificate Manager to validate your FQDN. If you leave this field blank, the module creates a self-signed certificate from the domain that you previously provided.

Parameter	Default	Description
		Note **Note**: See Get certificates ready in AWS Certificate Manager for more information on how to set up a certificate.
Route53 Hosted Zone ID (optional)	*Optional input*	Specify an Amazon Route 53 public hosted zone ID if you want the module to add a record routing your FQDN to the gateways. Leave this blank if you aren't using Amazon Route 53 to route your domain (including if you didn't specify a FQDN), or you don't want the record created for you (you will need to create a record pointing to the AWS Global Accelerator).
Cluster Instance Type	`m5.xlarge`	Amazon EC2 instance type to use for Leostream gateway cluster instances.
Min Cluster Instances	2	The minimum number of gateway instances allowed in the gateway cluster.
Max Cluster Instances	4	The maximum number of gateway instances allowed in the gateway cluster.
Port Range Bottom	`20001`	Bottom (starting) port of random port range used by the gateway to communicate over Amazon DCV. Provide an integer value between 1024 and 65535 for this field.
Port Range Top	`23000`	Top (ending) port of random port range used by the gateway to communicate over Amazon DCV. Provide an integer value between 1024 and 65535 for this field, ensuring that it is higher than the value specified for the **Port Range Bottom**.

7. For **Configure Tag Settings**, review the tags for this module and modify them as necessary. By default, this module uses tags defined in the main solution stack.

8. Choose **Next**.
9. On the **Review** page, verify all the parameters that you provided and choose **Deploy Module** if you confirm that they are correct.
10. The status of the Leostream Gateway shows as **Enabling in progress**. The deployment of this module takes approximately 1 hour. After the deployment is complete, the status of the Leostream Gateway module shows as **Enabled**.
11. Choose **External Link**. This opens a new window to the Leostream log in page.

> **Note**
>
> If you provided a FQDN in the previous steps, you'll be directed to the domain with the certificate that you provided. If you didn't provide the information, you'll be directed to the AWS Global Accelerator using a self-signed certificate. In this case, depending on your browser setting, you might see a privacy error with warnings about your connection not being private.

12. Sign in as a Leostream local admin user (Step 5: Enable Leostream Broker module step 11) to access the Leostream Connection Broker and manage configurations.
13. To access workstations through Leostream, sign in using your Active Directory credentials (Step 3: Enable Identity modules step 7 if you created a new AD using MCS). When signing in, use the username format `your-username@mad.mcs.int`.

    Download the Amazon DCV Client from https://www.amazondcv.com. After the connection is established, send the Ctrl+Alt+Delete command from the Connection menu in the Amazon DCV Client to unlock the workstation and proceed to the login screen.

Step 5: Enable Storage modules (Optional)

The solution supports two Amazon FSx storage options. Choose the storage module that best fits your workload requirements.

Amazon FSx for Windows File Server module

Follow these steps to enable the Amazon FSx for Windows File Server module.

1. Navigate to the MCS web console (see Launch the stack for details).
2. Select **Storage** from the left navigation pane.

3. Choose **Deploy New Module**.
4. For **Select Region**, select the Region where you want the FSx for Windows File Server module. There should be only one hub Region option if you have not deployed any spoke Regions.
5. For **Select Storage module**, select **Amazon FSx for Windows File Server** and choose **Next**.
6. For **Configure storage settings**, review the parameters for this module and modify them as necessary. This module uses the following default values.

Parameter	Default	Description
Automatic Backup Retention Period	30	Choose the number of days that Amazon FSx should retain automatic backups for this file system.
Throughput Capacity	64	The sustained speed for your file system. The system can burst to higher speeds. Values range from 32 MB/s to 12288 MB/s.
SSD Storage Capacity	256	Specify the size (in GiB) of the Amazon FSx storage that you would like to create. The allowed value is minimum 32 GiB and maximum 65536 GiB.

7. For **Configure Tag Settings**, review the tags for this module and modify them as necessary. By default, this module uses tags defined in the main solution stack.
8. Choose **Next**.
9. On the **Review** page, verify all the parameters that you provided and choose **Deploy Module** if you confirm that they are correct. The status of the storage shows as **Enabling in progress**. The deployment of this module takes approximately 30 minutes. After the deployment is complete, the status of the Storage module shows as **Enabled**.

10Follow the manual configuration steps in the guide or you can follow the instructions by clicking the **View** button on the MCS Web UI to complete the manual configuration.

Amazon FSx for Lustre module

Follow these steps to enable the Amazon FSx for Lustre module.

1. Navigate to the MCS web console (see Launch the stack for details).

2. Select **Storage** from the left navigation pane.
3. Choose **Deploy New Module**
4. For **Select Region**, select the Region where you want the FSx for Lustre module. There should be only one hub Region option if you have not deployed any spoke Regions.
5. For **Select Storage module**, select **Amazon FSx for Lustre** and choose **Next**.
6. For **Configure storage settings**, review the parameters for this module and modify them as necessary. This module uses the following default values.

Parameter	Default	Description
Storage Capacity	1200	Storage capacity in GiB. Minimum is 1,200 GiB; maximum is 24,000 GiB
Compression Type	NONE	Data compression reduces the physical disk space needed to store file data. Select LZ4 to enable data compression
S3 Path for Data Repository Association (optional)	*Optional input*	Specify an S3 Path to create a Data Repository Association within the Lustre Filesystem. If you leave this field blank, no Data Repository Association will be created.
Lustre Mount Path for Data Repository Association (optional)	/	Specify a path within the Lustre Filesystem to mount an S3 Data Repository Association. This parameter is required if you specified an S3 Path for Data Repository Association.

7. For **Configure Tag Settings**, review the tags for this module and modify them as necessary. By default, this module uses tags defined in the main solution stack.
8. Choose **Next**.
9. On the **Review** page, verify all the parameters that you provided and choose **Deploy Module** if you confirm that they are correct. The status of the storage shows as **Enabling in progress**. The deployment of this module takes approximately 10 minutes with no Data Repository Association, and approximately 20 minutes if you specify an S3 Path for Data Repository Association. After the deployment is complete, the status of the Storage module shows as **Enabled**.

10Follow the manual configuration steps in the guide or you can follow the instructions by clicking the **View** button on the MCS Web UI to complete the manual configuration.

> **Note**
>
> MCS deploys a SCRATCH filesystem and does not automatically create a backup when the module is disabled. However, if a Data Repository Association is deployed, data is bidirectionally synchronized between S3 and Lustre, and the data remains in S3 after module disablement.

Step 6: Manual configurations

Complete the following manual configurations based on your storage module deployment and workstation operating system. Choose the appropriate configuration that matches your setup:

Configure FSx for Windows File Server on Windows workstations

Use this configuration when you have deployed the FSx for Windows File Server storage module and are using Windows-based workstations.

1. Follow the instructions in Mapping a file share on an Amazon EC2 Windows instance to mount the Amazon FSx for Windows File Server file system on Windows workstations.

Configure FSx for Windows File Server on Linux workstations

Use this configuration when you have deployed the FSx for Windows File Server storage module and are using Linux-based workstations.

1. Follow the instructions in Manually join an Amazon EC2 Linux instance to your AWS Managed Microsoft AD.
2. Follow the instructions in Mounting a file share on an Amazon EC2 Linux instance to mount the Amazon FSx for Windows File Server file system on Linux workstations.

Configure FSx for Lustre on Linux workstations

Use this configuration when you have deployed the FSx for Lustre storage module and are using Linux-based workstations.

1. Follow the instructions in Manually join an Amazon EC2 Linux instance to your AWS Managed Microsoft AD.
2. Follow the instructions in Installing the Lustre client to install the Lustre client on Linux workstations.
3. Follow the instructions in Mounting from an Amazon Electic Compute Cloud instance to mount the Amazon FSx for Lustre file system on Linux workstations.
4. If using a data repository, follow Using data repositories with Amazon FSx for Lustre and POSIX metadata support to customize access and permissions to the Lustre file system.

Module Library

The Module Library serves as the central hub for managing Third-Party Modules in your MCS deployment that extend the capabilities of your cloud studio environment.

The Module Library page provides two main functions:

- **AWS Partners modules discovery**: Import and synchronize available AWS Partners modules that can be registered with your MCS deployment. These modules are sourced from a curated list maintained in the AWS Solutions GitHub repository.
- **Third-Party module management**: Register Third-Party Modules. De-register modules that are no longer needed in your environment.

> **Note**
>
> Modular Cloud Studio on AWS allows you to deploy and manage a scalable, secure, and global content production infrastructure in the cloud. This includes custom modules, developed by AWS Partners or other third parties, that you can choose to use ("Third-Party Modules"). AWS does not own or otherwise have any control over Third-Party Modules. Your use of the Third-Party Modules is governed by any terms provided to you by the Third-Party Module providers when you acquired your license to use them (for example, their terms of service, license agreement, acceptable use policy, and privacy policy). You are responsible for ensuring that your use of the Third-Party Modules comply with any terms governing them, and any laws, rules, regulations, policies, or standards that apply to you. You are also responsible for making your own independent assessment of the Third-Party Modules that you use. AWS does not make any representations, warranties, or guarantees regarding the Third-Party Modules, which are "Third-Party Content" under your agreement

with AWS. Modular Cloud Studio on AWS is offered to you as "AWS Content" under your agreement with AWS.

Synchronize AWS Partners modules

The Module Library includes a **Sync partner modules** button that allows you to import the latest AWS Partners modules from the listing hosted in the AWS Solutions GitHub repository. This ensures you have access to the most current AWS Partners modules available for registration.

> **Note**
>
> If a module is registered, it will not be updated with the latest data available in the AWS Solutions GitHub repository after synchronization.
> If an AWS Partners module listing is removed from the AWS Solutions GitHub repository, it will delete that module from the MCS deployment if the module is not registered.

To synchronize AWS Partners modules:

1. Navigate to the MCS web console (see Launch the stack for details).
2. Select **Module Library** from the left navigation pane.
3. Click **Sync partner modules** button.
4. The system will fetch the latest AWS Partners module listings from the AWS Solutions GitHub repository and update the available modules in the Module Library page. This synchronization process imports new AWS Partners modules that have been added to the AWS Solutions GitHub repository and updates information for existing and non-registered modules.
5. During the synchronization process, modules cannot be registered.
6. Once the process has finished, the page will reflect the last date and time the AWS Partners modules were synchronized.

The synchronization process does the following:

1. Checks if the process can be triggered by looking up the Lock DynamoDB table and acquire the lock by inserting the lock data to the same table.
2. Fetch the latest AWS Partners modules data from the AWS Solutions GitHub repository.

3. Execute the AWS Partners module synchronization state machine

The AWS Partners module synchronization state machine does the following:

1. Insert the module into the External modules DynamoDB table if a new module is found in the AWS Solutions GitHub repository.
2. Update the module data from the External modules DynamoDB table if the module already exists and the status is AVAILABLE.
3. Delete modules from the External modules DynamoDB table with AVAILABLE status if they are no longer hosted in AWS Solutions GitHub repository.
4. Release the process lock from the Lock DynamoDB table.

Register a module

Module registration makes Third-Party Modules available for deployment in your MCS environment. You can register modules from the available AWS Partner module listings or by providing a custom module manifest URL.

1. Navigate to the MCS web console (see Launch the stack for details).
2. Select **Module Library** from the left navigation pane.
3. To register an available AWS Partners module, choose the module row and select **Register *<Module Name>*** .
4. To register a third party module, select **Register new module.**
5. Type the URL of the module's manifest file, then select **Next**.
6. Select a module revision from the **Select Revision** dropdown menu and verify that the information is correct. Then select **Register module**.
7. Only one module can be registered at a time, and partner modules cannot be synchronized during the registration process
8. After registration is complete, the module appears in the **Module Library** with a REGISTERED status.

The module registration process does the following:

1. Checks if the process can be triggered by looking up the Lock DynamoDB table and acquire the lock by inserting the lock data to the same table.

2. Verify that a module can be registered by checking if status is `AVAILABLE`.
3. Execute the module registration state machine.

The module registration state machine does the following:

1. Validate the module can be registered by checking the status is not already `REGISTERED`
2. Inserts or updates module outputs in the Modules Mapping DynamoDB table.
3. Creates a Service Catalog product and adds it to the MCS Service Catalog Portfolio. A stack set constraint is added to configure the Regions that the module can be deployed in.
4. Inserts or updates the module in the External Modules DynamoDB table.
5. Inserts or updates the module in the Registered Modules DynamoDB table.
6. Release the process lock from the Lock DynamoDB table.
7. In any case of failures, it updates module status to REGISTER FAILED in External Modules DynamoDB table and Registered Modules DynamoDB table.

De-register a module

Module de-registration removes third party modules from your MCS environment, making them unavailable for deployment. This process returns the module to an available state if the module is an AWS Partners module.

1. Navigate to the MCS web console (see Launch the stack for details).
2. Select **Module Library** from the left navigation pane.
3. In the navigation pane, choose **Module Library**.
4. Choose the module to de-register, and select **De-register module**.
5. Select **Yes**.

The module de-registration does the following:

1. Ensure the module to be de-registered is not an internal module provided by MCS.
2. Check that the module is not enabled. Enabled modules cannot be de-registered.
3. Update module status to `DEREGISTER IN PROGRESS`.
4. Execute de-registration state machine.

The de-registration state machine does the following:

1. Removes module and outputs from the Modules Mapping DynamoDB table.
2. Deletes the module's Service Catalog product.
3. Deletes the module from the External Modules DynamoDB table. If it is an AWS Partners Module, it does not get deleted and its status is updated to **Available**.
4. Deletes the module from the Registered Modules DynamoDB table.
5. In any case of failures, it updates module status to `DEREGISTER FAILED` in External Modules DynamoDB table and Registered Modules DynamoDB table.

Password rotation

Overview and prerequisites

Password rotation is a critical security practice that helps maintain the integrity of your MCS deployment. AWS Managed Microsoft AD passwords expire every 90 days and must be rotated manually to prevent service disruptions.

Before you begin:

- Plan this activity during a maintenance window as users may experience temporary authentication issues
- Ensure you have administrative access to AWS Directory Service, Secrets Manager, and relevant service consoles

Active Directory password rotation

When you create an AWS Managed Microsoft AD through the identity module, three account users are created for authentication throughout the solution:

- **StudioAdmin** - Admin user for end-user access
- **SA_AdConnectorUser** - Service account for cross-region AD communication
- **SA_McsModulesUser** - General service account for MCS modules (e.g., syncing Microsoft AD users with Leostream module)

Step 1: Reset passwords in AWS Managed Microsoft AD

1. Navigate to the AWS Directory Service console
2. Locate the AWS Managed Microsoft AD instance associated with MCS (default domain: `studio.mcs.internal`)

> **Tip**
>
> If you're unsure of the Directory ID, log in to the MCS console via the CloudFront URL, go to the Identity tab, and click External Link.

3. Click on the Directory ID to open the directory details
4. Click **Actions** → **Reset User Password**
5. For each user:
   a. Enter the username
   b. Generate a secure password meeting complexity requirements
   c. Enter and confirm the new password
   d. Record the password securely for use in subsequent steps
   e. Click **Reset Password**
   f. Wait for confirmation message before proceeding to the next user

Step 2: Synchronize password changes

After resetting passwords in Active Directory, you must update the corresponding secrets and configurations in dependent services.

Update AWS Secrets Manager

1. Navigate to the AWS Secrets Manager console
2. Update the following secrets with their corresponding new passwords:

User	Secret Name Pattern
StudioAdmin	`/[MCSDeploymentId]/Identity/StudioAdminActiveDirectoryLoginCredentials`
SA_AdConnectorUser	`/[MCSDeploymentId]/Identity/AdConnectorServiceAccountActiveDirectoryLoginCredentials`
SA_McsModulesUser	`/[MCSDeploymentId]/Identity/McsModulesServiceAccountActiveDirectoryLoginCredentials`

3. For each secret:
    a. Click on the secret name
    b. Click **Retrieve Secret Value**
    c. Update the password field with the corresponding new password
    d. **Save** the changes

Update AD Connector (spoke Regions)

For each spoke region with an AD Connector:

1. Use the Region Selector to navigate to the spoke region
2. Go to the Directory Service console
3. Click on the AD Connector with the MCS domain name
4. Navigate to **Network and Security**
5. Scroll to **Service Account Credentials** and click **Update**
6. Set the password to match the new SA_AdConnectorUser password
7. Click **Update**
8. Wait for the status to show "Active" before proceeding to the next region

> ⚠ **Important**
>
> Wait approximately 1 hour before attempting to log in to workstations in spoke Regions after updating the AD Connector password.

Update Leostream Active Directory authentication

1. Log in to the Leostream management dashboard using the admin user
2. Navigate to **Setup** → **Authentication Servers** → **Edit**
3. Locate the authentication server configuration section
4. Update the password field with the new SA_McsModulesUser password
5. Click **Save**
6. Test the connection by attempting to authenticate a test user

After completing this update, you can log in to Leostream Gateway with Amazon DCV using the new StudioAdmin password or any other user credentials from the AWS Managed Microsoft AD.

Update storage modules

Amazon FSx for Lustre: Password changes are automatically synchronized. No manual action required.

Amazon FSx for Windows: Manual password synchronization is required.

1. Navigate to the Amazon FSx console
2. Click on your Windows file system
3. Go to **Network and Security**
4. Locate the **Service Account** section with SA_McsModulesUser
5. Click **Update** next to the service account credentials
6. Set the password to match the new SA_McsModulesUser password
7. Monitor the **Updates** section for completion of the Service Account Credential update
8. Verify the file system status remains "Available" after the update

If the Amazon FSx Windows module shows as misconfigured after password expiration:

1. Click **Attempt Recovery** to reconfigure the module
2. Wait for the update to complete
3. Verify the module status returns to available

Manually rotating the Leostream database secret

This solution doesn't provide automatic secrets rotation. Depending on your security requirements, you may consider manually rotating the credentials for your Leostream Connection Broker database. Follow these steps to manually rotate PostgreSQL database credentials:

1. **Log into to the admin dashhoard with admin**

   Log into the Leostream Broker through the Leostream Gateway with "admin" credentials. That is located at: `/[MCSDeploymentId]/WorkstationManagement/Leostream/Console/AdminUserCredentials`.

2. **Switch Leostream Credentials**

   This step is necessary. Without this temporary switch, the gateway cannot connect to the broker when the new password is changed. To update the corresponding credentials in the Leostream Connection Broker, see the Leostream Administrator's Guide. This updates the Leostream settings to use the new database password. Under the Systems > Maintenance, choose DATABASE OPTIONS > Switch to PostgreSQL database. You will use the postgres default admin credentials to make this switch. This is located at `LeostreamBrokerStorageSitCD-*`.

3. **Leostream Connection Broker Restart Time**

   The Broker will take a couple of minutes to restart for you to be able to log in.

4. **Update the PostgreSQL user password**

   To change the password of the PostgreSQL "leostream" user, follow the instructions provided in the PostgreSQL documentation SQL ALTER USER Command. Ensure you modify only the "leostream" user credentials, not the default administrator account. This helps you ensure that the database credentials are updated correctly at the database level.

5. **Update secret in Secrets Manager**

   Locate the secret at: `/[MCSDeploymentId]/WorkstationManagement/Leostream/Database/Credentials`, then update the secret with the new credentials.

```
*Update Leostream credentials*
```

To update the corresponding credentials in the Leostream Connection Broker, see the Leostream Administrator's Guide. This updates the Leostream settings to use the new database password. Under the Systems > Maintenance, choose DATABASE OPTIONS > Switch to PostgreSQL database. You will switch back to the "leostream" user. This is located at `/[MCSDeploymentId]/WorkstationManagement/Leostream/Database/Credentials`.

6. **Leostream Connection Broker Restart Time**

   The Broker will take a couple of minutes to restart for you to be able to log in.

The following secrets can be rotated using a similar process:

- `/[MCSDeploymentId]/WorkstationManagement/Leostream/API/ServiceUserCredentials`
- `/[MCSDeploymentId]/WorkstationManagement/Leostream/Console/AdminUserCredential`

Enhanced TLS Security

This section provides guidance on configuring custom domains to enhance TLS security for the API Gateway endpoint.

Overview and prerequisites

By default, the API Gateway URL uses AWS-managed TLS configuration that allows TLS 1.0 and above. For enhanced security, you can configure a custom domain with stronger TLS requirements.

Before you begin:

- Ensure you own or control a domain name
- Obtain an SSL/TLS certificate for your domain (from AWS Certificate Manager or imported)
- Verify you have permissions to update DNS records for your domain
- Plan for a maintenance window, as this change may briefly impact API accessibility

Configuration Steps

Follow the AWS documentation to Choose a security policy for your REST API custom domain in API Gateway.

After setting up the custom domain, complete the MCS-specific configuration:

1. Navigate to the S3 bucket containing your MCS frontend configuration
2. Locate the runtime configuration file
3. Update the API endpoint URL to use your custom domain
4. Invalidate the CloudFront cache to ensure the new configuration is used

Verification

After completing the configuration:

1. Test the custom domain endpoint to ensure it’s accessible
2. Verify TLS version using a tool like SSL Labs or `openssl`:

   ```
   openssl s_client -connect your-custom-domain:443
   ```

Security Considerations

- While the original API Gateway URL remains accessible, ensure your application only uses the custom domain endpoint
- Regular certificate rotation and renewal should be part of your maintenance procedures
- Monitor certificate expiration dates in AWS Certificate Manager

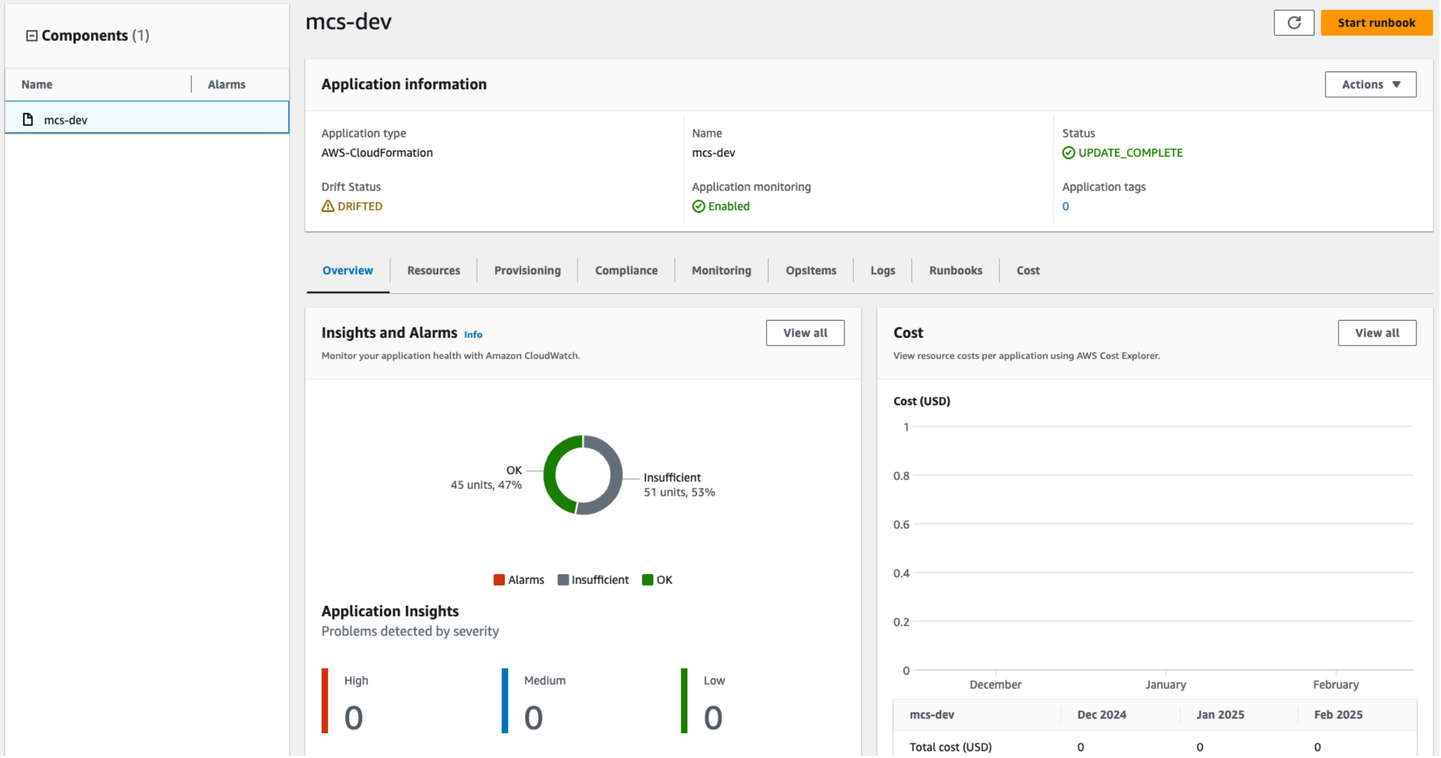
Monitoring the solution with AWS Service catalog appregistry

The solution includes a Service Catalog AppRegistry resource to register the CloudFormation template and underlying resources as an application in both Service Catalog AppRegistry and Application Manager.

Application Manager gives you an application-level view into this solution and its resources so that you can:

- Monitor its resources, costs for the deployed resources across stacks and AWS accounts, and logs associated with this solution from a central location.
- View operations data for the solution's AWS resources (such as deployment status, Amazon CloudWatch alarms, resource configurations, and operational issues) in the context of an application.

The following figure depicts an example of the application view for this solution stack in Application Manager.



Application type
AWS-CloudFormation

Name
mcs-dev

Status
UPDATE_COMPLETE

Drift Status
DRIFTED

Application monitoring
Enabled

Application tags
0

Monitor your application health with Amazon CloudWatch.

View resource costs per application using AWS Cost Explorer.

mcs-dev	Dec 2024	Jan 2025	Feb 2025
Total cost (USD)	0	0	0

> **Note**
>
> You must activate CloudWatch Application Insights, AWS Cost Explorer, and cost allocation tags associated with this solution. They are not activated by default.

The following logs are captured and stored in this solution:

- Application logs
- Access Logs
- Audit Logs
- Default metrics

Most logs are stored with a 10-year retention period under these prefix patterns:

- /aws/vendedlogs/lambda/modular-cloud-studio-on-aws/deployment-id/…
- /aws/vendedlogs/states/modular-cloud-studio-on-aws/deployment-id/…
- /modular-cloud-studio-on-aws/deployment-id/…

Exception: The following logs could not be altered and have a "Never Expire" retention setting that cannot be modified:

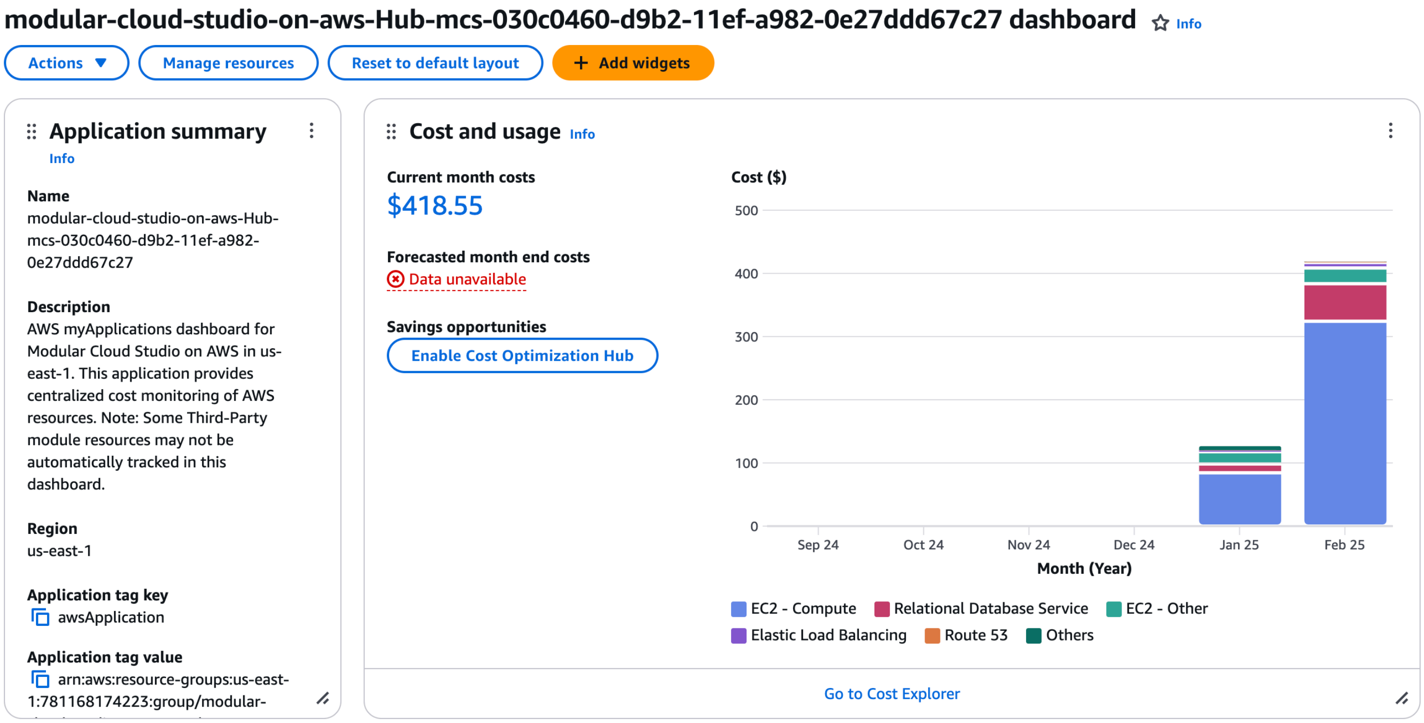
- Image Builder logs: /aws/imagebuilder…
- Cross-Region AWS SDK logs: /aws/lambda/StackSet-SC-AccountId-CrossRegionAwsSdk…
- API Gateway execution logs: API-Gateway-Execution-Logs_…/prod

myApplications Dashboard

The solution also registers resources under an application on the AWS myApplications dashboard. From this centralized dashboard, you can view further cost insights and configure and view further metrics for your solution by using services such as Security Hub, CloudWatch, and Cost Explorer.

The following figure depicts an example of the application view for this solution stack in myApplications.

Application View



modular-cloud-studio-on-aws-Hub-mcs-030c0460-d9b2-11ef-a982-0e27ddd67c27 dashboard

Application summary

Name
modular-cloud-studio-on-aws-Hub-mcs-030c0460-d9b2-11ef-a982-0e27ddd67c27

Description
AWS myApplications dashboard for Modular Cloud Studio on AWS in us-east-1. This application provides centralized cost monitoring of AWS resources. Note: Some Third-Party module resources may not be automatically tracked in this dashboard.

Region
us-east-1

Application tag key
awsApplication

Application tag value
arn:aws:resource-groups:us-east-1:781168174223:group/modular-

Cost and usage

Current month costs
$418.55

Forecasted month end costs
Data unavailable

Savings opportunities
Enable Cost Optimization Hub

🟦 EC2 - Compute 🟥 Relational Database Service 🟩 EC2 - Other
🟪 Elastic Load Balancing 🟧 Route 53 🟩 Others

The application name follows the naming schema `modular-cloud-studio-on-aws-[Hub|Spoke]-[MCSDeploymentId]`. Each application is deployed and managed on a region-by-region basis. Hub region have the application created during initial deployment, spoke regions receive their dedicated applications automatically upon successful region enablement. Any EC2 instance launched within an MCS VPC will have its associated costs included and tracked under the respective application.

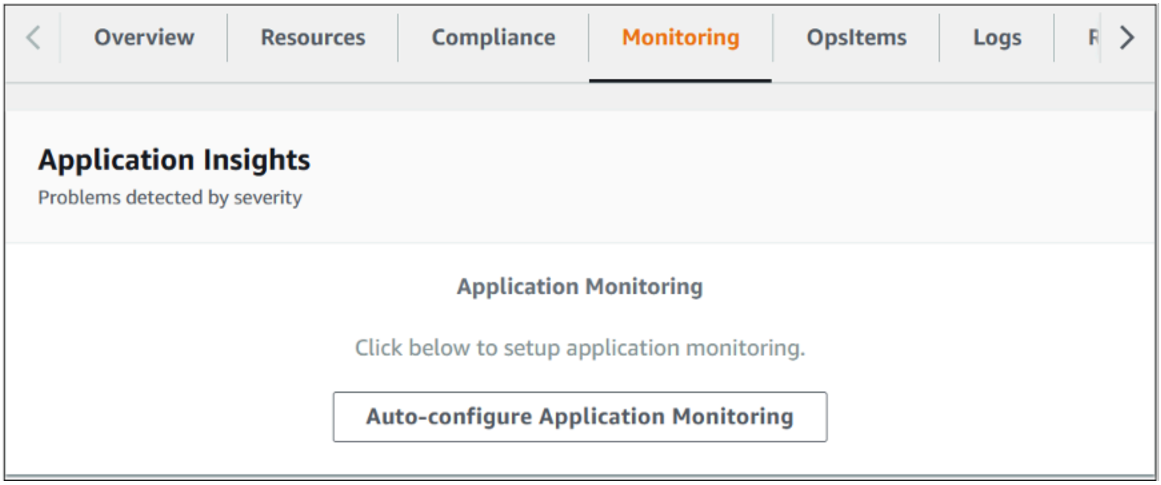
> **Note**
>
> Some Third-Party module resources may not be automatically tracked in this dashboard.

Activate CloudWatch Application Insights

1. Sign in to the Systems Manager console.
2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, choose **AppRegistry applications**.
4. In **AppRegistry applications**, search for the application name for this solution and select it.

The next time you open Application Manager, you can find the new application for your solution in the **AppRegistry application** category.

5. In the **Components** tree, choose the application stack you want to activate.
6. In the **Monitoring** tab, in **Application Insights**, select **Auto-configure Application Monitoring**.



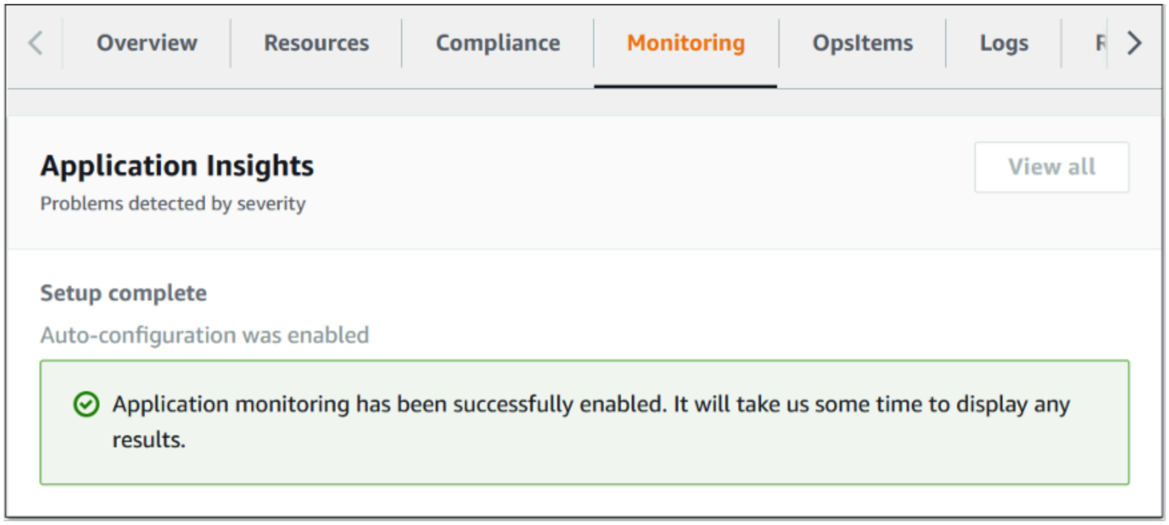
Application Insights

Problems detected by severity

Application Monitoring

Click below to setup application monitoring.

Monitoring for your applications is now activated and the following status box appears:



Application Insights

Problems detected by severity

Setup complete

Auto-configuration was enabled

Application monitoring has been successfully enabled. It will take us some time to display any results.

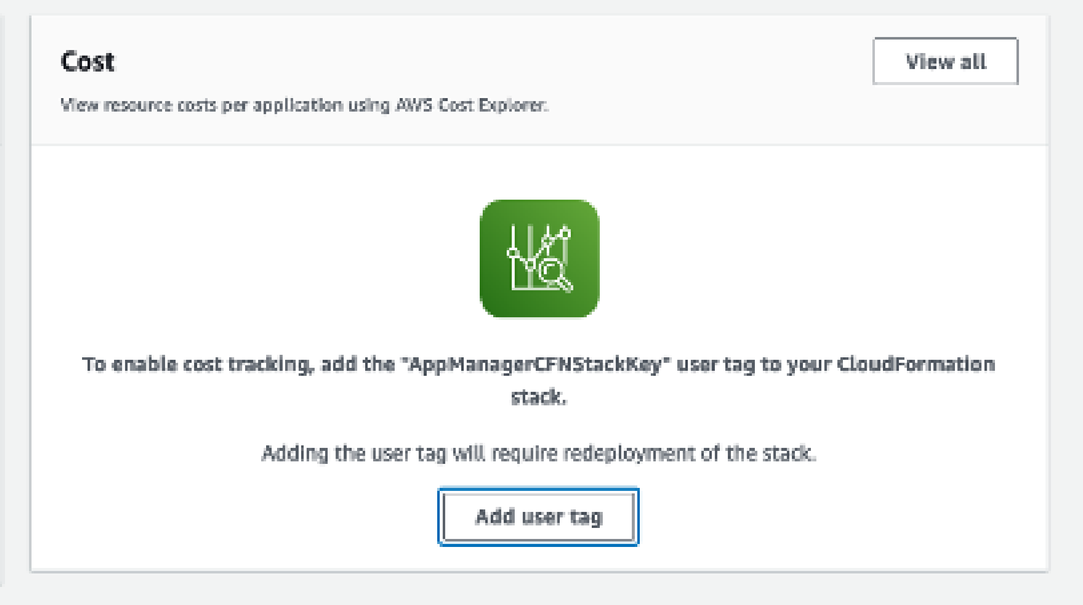
Confirm cost tags associated with the solution

After you activate cost allocation tags associated with the solution, you must confirm the cost allocation tags to see the costs for this solution. To confirm cost allocation tags:

1. Sign in to the Systems Manager console.

2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, choose the application name for this solution and select it.
4. In the **Overview** tab, in **Cost**, select **Add user tag**.

Cost tab



Cost

View resource costs per application using AWS Cost Explorer.

To enable cost tracking, add the "AppManagerCFNStackKey" user tag to your CloudFormation stack.

Adding the user tag will require redeployment of the stack.

5. On the **Add user tag** page, enter `confirm`, then select **Add user tag**.

The activation process can take up to 24 hours to complete and the tag data to appear.

Activate cost allocation tags associated with the solution

After you activate Cost Explorer, you must activate the cost allocation tags associated with this solution to see the costs for this solution. The cost allocation tags can only be activated from the management account for the organization. To activate cost allocation tags:

1. Sign in to the [AWS Billing and Cost Management console](https://console.aws.amazon.com/billing/home).
2. In the navigation pane, select **Cost Allocation Tags**.
3. On the **Cost allocation tags** page, filter for the `AppManagerCFNStackKey` tag, then select the tag from the results shown.
4. Choose **Activate**. The activation process can take up to 24 hours to complete and the tag data to appear.

Activate AWS Cost Explorer

You can see the overview of the costs associated with the application and application components within the Application Manager console through integration with AWS Cost Explorer which must be first activated. Cost Explorer helps you manage costs by providing a view of your AWS resource costs and usage over time. To activate Cost Explorer for the solution:

1. Sign in to the AWS Cost Management console.
2. In the navigation pane, select **Cost Explorer**.
3. On the **Welcome to Cost Explorer** page, choose **Launch Cost Explorer**.

The activation process can take up to 24 hours to complete. Once activated, you can open the Cost Explorer user interface to further analyze cost data for the solution.

Troubleshooting

This section provides troubleshooting instructions for deploying and using the solution.

Known limitations addresses unsupported features of the solution. Known issues provides instructions to mitigate known errors. If these instructions do not address your issue, Contact AWS Support provides instructions for opening an AWS Support case for this solution.

Known limitations

Limitation: Spoke Region Leostream Gateway routing

When a user connects to a workstation, the Global Accelerator directs them to the nearest Leostream Gateway. The user can only establish a successful connection if the workstation is located in the same spoke Region or in the hub Region. Routing between different spoke Regions with the Leostream Gateway is not supported.

For instance, consider a setup where `us-east-1` is the hub Region, and `us-west-2` and `eu-central-1` are spoke Regions. If a user near `eu-central-1` attempts to connect to a workstation located in `us-west-2`, the connection will fail.

Limitation: vCPU capacity requirement

When building Windows or Linux AMIs in the workstation module, the solution uses an EC2 Image Builder pipeline that requires a g4dn.xlarge instance. By default, AWS accounts have a vCPU limit of 0 for G-type instances. You may encounter the following error message:

```
An error occurred (VcpuLimitExceeded) when calling the RunInstances
operation: You have requested more vCPU capacity than your current vCPU
limit of 0 allows for the instance bucket that the specified instance type
belongs to.
```

You will need to request a quota increase in the AWS Service Quotas console.

Known issues

Problem: Register module failed

If during module registration, you received an error message for a Third-Party Module, check that the manifest file is correct, and that template is a valid CloudFormation template. If there are problems with these files, the MCS web console shows an error with more information.

Resolution

Complete the following task to clean up a module from the `Register Failed` state:

1. In the hub Region, sign in to the Service Catalog console.
2. Ensure that no products were created in Service Catalog for the module that was registered. If there were, disassociate the from any MCS portfolio and remove the product. See Deleting provisioned products for more information.
3. Sign in to the DynamoDB console.
4. In the **Registered Modules** table, check for a row that represents the module that failed registration. If it exists, remove that row.
5. In the **Modules Mapping** table, check for a row that contains the name of the module that failed registration. It will be in the field called **module_pks**. If it is the only entry in that row, remove that row. Otherwise, modify that list and only remove the module partition key from it, leaving the others in place.
6. In the **External Module*** table, if the imported the module was custom made and not one of the Third-Party Modules, remove the row. Otherwise, change the status of it to `AVAILABLE`.
7. Refresh the UI and try to register the module again.

Problem: Enable module failed

If you receive a **CREATE_FAILED** status when enabling a module, sign in to the Service Catalog console and ensure that the provisioned product received the correct inputs at deployment.

Resolution

Follow the instructions in Disable a module to clean up a module from the Enable Failed state.

Problem: Disable module failed

If you received a **DELETE_FAILED** message when disabling a module, sign in to the Service Catalog console and ensure that the provisioned product received the correct inputs at deployment.

Resolution

Complete the following task to clean up a module from the `Disable Failed` state:

1. In the hub Region, sign in to the Service Catalog console.
2. Navigate to **Provisioned Products** using the left hand navigation panel.
3. Find the provisioned product for the module that failed to disable. For Third-Party Modules, it will have the same name as what is listed in the manifest file. Terminate the provisioned product for this module.
4. Sign in to the DynamoDB console.
5. In the **Enabled Modules** table, check for a row that represents the module that failed to disable. The row will have a field in the **module_name** column that corresponds to the name of the module that failed disable. If this is a Third-Party Module, the name is listed in the module's manifest. Remove that row.
6. In the **Enabled Modules** table, check for a row that has the module name listed in **active_dependents**. Remove any mention of that module in that column, without removing other entries in the list.
7. In the **Registered Modules** table, ensure that the module's status is `REGISTERED`.
8. Ensure that there are no remnants of the module that failed to delete. For example, deleting the Service Catalog product for Leostream Broker doesn't remove the AMIs or EC2 instances.
9. Refresh the UI and try disabling the module again.

Problem: Deregister module failed

If you receive a **FAILED** message when de-registering a module during spoke stack deployment, follow these steps. If the module has been enabled, disable the module first.

Resolution

Complete the following task to clean up a module from the `De-Register Failed` state:

1. In the hub Region, sign in to the Service Catalog console.
2. Ensure that no product exists in Service Catalog for the module you are attempting to de-register. If one exists, disassociate it from the MCS portfolio and remove the product.
3. Sign in to the DynamoDB console.
4. In the **Registered Modules** table, check for a row that represents the module that failed registration. If it exists, remove that row.
5. In the **Modules Mapping** table, check for a row that contains the name of the module that failed registration. It will be in the field called **module_pks**. If it is the only entry in that row, remove that row. Otherwise, modify that list and only remove the module partition key from it, leaving the others in place.
6. In the **External Module** table, if the imported the module was custom made and not one of the Third-Party Modules, remove the row. Otherwise, change the status of it to `AVAILABLE`.

Problem: Reset MCS admin credentials or add new user

These steps can only be completed if the user has privileges to create or edit credentials in Amazon Cognito. As such, this section is applicable to the admin that deployed MCS originally and has advanced permissions. If you want to reset the admin credentials or add a new authorized user, follow these steps.

Resolution

Complete the following tasks to update login information to the MCS portal:

1. In the hub Region, navigate to the Amazon Cognito console.
2. Select the user pool for your MCS deployment.
3. If you want to reset the admin password, select that user, and in the following screen navigate to **Actions**, then **Reset Password**. Otherwise, select **Create user** and follow the steps there. Associate an email with the new user.

Problem: Enable Module failed with error message "Invalid Logging Configuration: The CloudWatch Logs Resource Policy size was exceeded."

Resolution

This error occurs when the CloudWatch Logs resource policy exceeds the maximum size limit. These resource policies exist at the account level. To resolve this issue, follow the instructions in Disable a module to clean up a module from the Enable Failed state. Then, modify your CloudWatch Logs resource policy to reduce its size or add a wildcard to vendedlog resource policies. For more information, consult the CloudWatch Log documentation regarding CloudWatch Log Resource Policies. Once done, re-enable the module.

Problem: Connection to Leostream Workstation failed with "Unable to connect" error message when using Unmanaged Active Directory

If you received an "Unable to connect" error within your Amazon DCV client and are using an Unmanaged Active Directory to deploy your Leostream broker and gateway, follow these steps.

Resolution

This error occurs when the required DNS resolver resources are not properly configured for your Unmanaged Active Directory, preventing domain name resolution and blocking connections to Leostream workstations. Follow the instructions in Import Custom Microsoft Active Directory to ensure you have the correct configurations for Unmanaged Active Directory.

Problem: Leostream Gateway module enablement failure due to Leostream API unauthorized error code (401)

During deployment, the Leostream Gateway module may occasionally receive a 401 (Unauthorized) response from the Leostream Broker cluster API. This occurs despite the system properly generating new authentication tokens and implementing standard retry mechanisms. This causes the deployment of the Leostream Gateway module to fail.

Our engineering team has confirmed this is an issue within the Leostream system and is actively working with Leostream to implement a permanent resolution.

Problem: Enable Module failed with error message "Invalid Logging Configuration: The CloudWatch Logs Resource Policy size was exceeded." 93

Resolution

If you encounter this API authentication issue during deployment, please follow these steps:

1. Disable the affected Leostream Gateway module instance via Modular Cloud Studio UI
2. Wait until the Leostream Gateway module disables — If you encounter a disablement failure, follow steps below
3. Redeploy the module

Note: In rare instances where module deletion is incomplete via the standard interface, use the CloudFormation console to complete the process:

1. Navigate to the CloudFormation console
2. Locate and select the relevant stack
3. Choose the "Delete" option from the Actions menu
4. If prompted about deletion failures for custom resources, you may safely ignore deleting those to proceed removing the stack

Problem: Building the AMI of Leostream Broker or Leostream Gateway in non-USA regions sometimes results in connection issues that rollback the stack

AMI builds for Leostream Broker and Gateway EC2 instances occasionally fail in non-US regions due to connectivity issues with Leostream’s hosted servers. These failures typically manifest as connection timeouts (HTTP 522 errors) or end-of-file errors when downloading package installers, often occurring during off-peak hours, and result in CloudFormation stack rollbacks.

Resolution

Disable the failed module from MCS UI, retry enabling the module again.

Third-Party module issues

This solution provides access to AWS Partner modules through the Module Library. For issues related to third-party modules, including: licensing questions, technical support, or implementation assistance, you can contact the partner company directly by:

1. Navigating to the Module Library (see Module Library)
2. Locating the specific module
3. Clicking **Support Info** to access the partner’s contact information

Contact AWS Support

If you have AWS Developer Support, AWS Business Support, or AWS Enterprise Support, you can use the Support Center to get expert assistance with this solution. The following sections provide instructions.

Create case

1. Sign in to Support Center.
2. Choose **Create case**.

How can we help?

1. Choose **Technical**.
2. For **Service**, select **Solutions**.
3. For **Category**, select **Other Solutions**.
4. For **Severity**, select the option that best matches your use case.
5. When you enter the **Service**, **Category**, and **Severity**, the interface populates links to common troubleshooting questions. If you can’t resolve your question with these links, choose **Next step: Additional information**.

Additional information

1. For **Subject**, enter text summarizing your question or issue.
2. For **Description**, describe the issue in detail.
3. Choose **Attach files**.
4. Attach the information that AWS Support needs to process the request.

Help us resolve your case faster

1. Enter the requested information.
2. Choose **Next step: Solve now or contact us**.

Solve now or contact us

1. Review the **Solve now** solutions.
2. If you can’t resolve your issue with these solutions, choose **Contact us**, enter the requested information, and choose **Submit**.

Uninstall the solution

You can uninstall the MCS solution from the AWS Management Console or by using the AWS Command Line Interface (AWS CLI). You must manually delete the modules and some of the core resources created by this solution. AWS Solutions do not automatically delete dependents of modules and storage backup resources in case you have stored data to retain. As such, see the following information on how to delete S3 buckets, CloudWatch logs, EC2 AMIs (from Leostream Broker module) and SSM parameters (from Leostream Broker module).

> ⚠ **Important**
>
> Before uninstalling the solution, ensure that all modules and all spoke Regions have been disabled, and any Third-Party modules registered to the solution have been de-registered.

Using the AWS Management Console

1. Sign in to the CloudFormation console.
2. On the **Stacks** page, select this solution's installation stack.
3. Choose **Delete**.

Using AWS Command Line Interface

Determine whether the AWS Command Line Interface (AWS CLI) is available in your environment. For installation instructions, see What Is the AWS Command Line Interface in the *AWS CLI User Guide*. After confirming that the AWS CLI is available, run the following command.

```
$ aws cloudformation delete-stack --stack-name <installation-stack-name>
```

Manual delete retained resources

After deleting the core stack, see the following sections for how to delete remaining resources.

Deleting the Amazon S3 buckets

This solution is configured to retain the solution-created Amazon S3 bucket (for deploying in an opt-in Region) if you decide to delete the AWS CloudFormation stack to prevent accidental data loss. After uninstalling the solution, you can manually delete this S3 bucket if you do not need to retain the data. Follow these steps to delete the Amazon S3 bucket.

1. Sign in to the Amazon S3 console.
2. Choose **Buckets** from the left navigation pane.
3. Locate the S3 buckets that begin with *<stack-name>* .
4. Select the S3 bucket and choose **Delete**.

To delete the S3 bucket using AWS CLI, run the following command:

```
$ aws s3 rb s3://<bucket-name> --force
```

Deleting the CloudWatch Logs

This solution retains the CloudWatch Logs if you decide to delete the AWS CloudFormation stack to prevent against accidental data loss. After uninstalling the solution, you can manually delete the logs if you do not need to retain the data. Follow these steps to delete the CloudWatch Logs.

1. Sign in to the Amazon CloudWatch console.
2. Choose **Log Groups** from the left navigation pane.
3. Locate the log groups created by the solution.
4. Select one of the log groups.
5. Choose **Actions** and then choose **Delete**.

Repeat the steps until you have deleted all the solution log groups.

Deleting the Amazon EC2 AMIs

This solution is configured to retain the Amazon EC2 AMIs if you enable and then disable the Leostream Broker stack. After uninstalling the solution, you can manually delete the Amazon EC2 AMIs if you do not need to retain the data. Follow these steps:

1. Sign in to the Amazon EC2 console.
2. Choose **AMIs** from the left navigation pane under **Images**.
3. Select the **AMIs** you wish to remove, and choose **Actions** → **Deregister AMI**.

To delete the DynamoDB tables using AWS CLI, run the following command:

```
$ aws ec2 deregister-image --image-id <image-id>
```

Deleting the SSM Parameters

This solution is configured to retain Systems Manager Parameters if you enable and then disable the Leostream Broker stack. After uninstalling the solution, you can manually delete the Systems Manager Parameters if you do not need to retain the data. Follow these steps:

1. Sign in to the Systems Manager console.
2. Choose **Parameter Store** from the left navigation pane.
3. Select the **Parameters** you wish to remove, and choose **Delete**.

To delete the Parameters tables using AWS CLI, run the following command:

```
$ aws ssm delete-parameter --name <parameter-name>
```

Deleting the FSx for Windows File Server Backups

This solution retains the FSx for Windows File Server backups if you decide to disable the AWS FSx for Windows File Server module to prevent against accidental data loss. After uninstalling the solution, you can manually delete the backups if you do not need to retain the data. Follow these steps to delete the FSx Backups.

1. Sign in to the Amazon FSx console.
2. Choose **Backups** from the left navigation pane.
3. Select the backup created by the Amazon FSx for Windows File Server file system when it was disabled.
4. Choose **Actions**, and then choose **Delete Backup**.

Developer guide

This section provides instructions for creating Third-Party Modules, schemas for module metadata and module manifest, module parameters, and an API reference.

Create Third-Party Modules for MCS

You can create your own third-party MCS modules by following these steps:

Step 1: Design your Third-Party Module

Step 2: Create the CloudFormation template

Step 3: Create the assets referenced by the template

Step 4: Create the module metadata

Step 5: Create the module manifest

Step 6: Create module intercommunication

Step 7: Create module instructions (optional)

Step 1: Design your Third-Party Module

Beneath the surface, an MCS module is a CloudFormation stack defined by a CloudFormation template. When the module is registered with MCS, it is added to a product portfolio in Service Catalog.

MCS needs additional details about the module, such as the module type (for example, Network, Identity, Workstation Management, Storage, or Custom), revision, and dependencies on resources from other modules. This metadata is necessary for module discovery and registration.

To define a module, you need:

- A CloudFormation template
- Assets referenced by the template
- Module metadata (as part of the CloudFormation template)
- Module revision manifest file

Conceptually, registered module data is referenced as follows:

```
 Modular Cloud Studio on AWS
 \
 \ (Module)
 \-------------> Module Revision Manifest
|
|(1.0.0)
 +----------> AWS CloudFormation Template + Module Metadata
|\
|\---> CFN Resource Assets
|
|(2.0.0)
 +----------> AWS CloudFormation Template + Module Metadata
|\
|\---> CFN Resource Assets
|
|(2.1.0)
 +----------> AWS CloudFormation Template + Module Metadata
|\
|\---> CFN Resource Assets
|
|(3.0.0)
 +----------> AWS CloudFormation Template + Module Metadata
 Metadata
 \
 \---> CFN Resource Assets
```

Step 2: Create the CloudFormation template

The CloudFormation template defines the infrastructure that makes up the module. When you register a new module, MCS uses ValidateTemplate to validate the template and extract parameters. The template must be accessible so that MCS can fetch the template.

MCS fetches the CloudFormation template and generates a checksum to store along with the registration metadata. When the module is enabled, MCS verifies that the template still matches the checksum to ensure that it didn't get corrupted or modified since it was registered. If the checksum doesn't match, MCS reports the mismatch as an error, and the module isn't enabled.

For instructions on how to create CloudFormation templates, see Working with CloudFormation templates in the *AWS CloudFormation User Guide*.

Step 3: Create the assets referenced by the template

The assets must be accessible so that the CloudFormation template can access the assets when deploying the stack. For example, the asset can be a public Amazon S3 object.

Step 4: Create the module metadata

MCS needs additional metadata about a module that isn't part of the native CloudFormation template. The module metadata is stored in the CloudFormation template in the Metadata section. The metadata is stored with the template and no linkage is necessary with an external file.

The module metadata schema requires the following additional information specific to MCS:

- Module type
- Module name
- Dependencies on other modules
- Module revision number

Step 5: Create the module manifest

To track MCS module updates, you must list module revisions in an external module manifest file. The module manifest must be accessible so that MCS can read it. There is only a single manifest file per module. When you publish a new revision of a module, update the manifest to reflect that a new revision is available.

The module manifest schema must meet the following requirements:

- Be in JSON format
- Include the following:
  - Name of the module author/owner (company name)
  - Description text
  - Optional URL to the web page with more information about the module
  - Module name
  - Module category (Network, Identity, WorkstationManagement, Storage, PixelStreaming, or Custom)
- Contain an array of revisions, each of which:

- Specifies the URL(s) to the CloudFormation template:
  - Use `TemplateUrl` when the hub and spoke modules share the same template.
  - Use `TemplateUrls` when the hub and spoke modules have separate templates, or for hub-only modules.

> **Note**
>
> **Note:** These two fields are mutually exclusive.

- Includes the revision number.
- Includes compatibility information specifying which revisions of MCS it is compatible with.
- Contains details about what's new in the revision.

When registering a new external module, the MCS admin user only requires the URL of the module manifest file. Optionally, the user can also specify a revision number to access a specific revision of the module. If the revision is not supplied, MCS assumes that the user needs the latest compatible revision of the module.

Step 6: Create module intercommunication

To facilitate a pattern known as dynamic dependency loading, the configuration data is stored on the Systems Manager Parameter Store so that it can be lazy-loaded exactly when it is needed by an MCS module.

The following is the structure of all parameters output by MCS:

```
/{deployment_id}/{module_type}/{component}
```

- **deployment_id** - This value is generated when MCS is first deployed and is configured on the Lambda function serving API requests. When deploying any module (including Third-Party Modules), the `deployment_id` is provided as a CloudFormation parameter. The `deployment_id` is always prefixed with `mcs-`.
- **module_type** - This value is the type of the module providing the output. The same type can be used by multiple mutually-exclusive modules that provide the same output such as AWS Managed Microsoft AD, compared to unmanaged Microsoft Active Directory.

- **component** - The name of the component providing the output. There could be one or multiple paths as part of this value.

As an example, see the following Managed Active Directory module with its input and output parameters (created after completing the steps to Create Third-Party Modules for MCS):

- MCS Managed Active Directory module - inputs:

```
/{deployment_id}/Network/VpcId
/{deployment_id}/Network/PrivateSubnet1/AZ
/{deployment_id}/Network/PrivateSubnet1/SubnetID
/{deployment_id}/Network/PrivateSubnet2/AZ
/{deployment_id}/Network/PrivateSubnet2/SubnetID
```

- MCS Managed Active Directory module - outputs:

```
/{deployment_id}/Identity/ActiveDirectoryId
/{deployment_id}/Identity/ActiveDirectoryServerIP1
/{deployment_id}/Identity/ActiveDirectoryServerIP2
/{deployment_id}/Identity/ActiveDirectoryDomainName
/{deployment_id}/Identity/ActiveDirectorySecretArn
/{deployment_id}/Identity/DefaultActiveDirectoryLoginCredentials
/{deployment_id}/Identity/StudioAdminDirectoryLoginCredentials
```

For more information on parameters, see the Module parameters section

Step 7: Create module instructions (optional)

This step allows you to provide custom, user-friendly instructions within the MCS interface, enhancing the user experience by offering clear guidance on module usage after enablement.

To implement module instructions, you'll need to create an AWS Lambda function that returns instruction content. The Lambda function must be named with the prefix `MCSInstructionGenerationLambda-` and be referenced in your CloudFormation outputs with the key `InstructionGenerationLambdaArn`. Append your stack id (without hyphens) to the Lambda function name is recommended to ensure unique naming across multiple deployment in the same region.

The function can return either a plain string or a JSON object formatted as `{"content": "your instruction string"}`. A basic example implementation for reference:

```
TEMPLATE_FILE_NAME = "template.html"
def handler(event, context):
 with open(
 TEMPLATE_FILE_NAME,
 "r",
 encoding="utf-8") as f:
 instructions = f.read()
 return {"content": instructions}
```

When formatting your instructions, note that inline CSS is supported, but external CSS is not. HTML tags will inherit Cloudscape default styling. The maximum size for the instruction string is **250 KB**.

You can verify successful implementation when a "View" link appears in the module’s row on the UI after deployment. This feature enhances usability by providing context-specific documentation directly within the MCS interface, helping users better understand and use the module’s feature.

Module metadata schema

```
$schema: http://json-schema.org/draft-04/schema#
title: Modular Cloud Studio on AWS Module Metadata
description: >-
 Metadata for a Modular Cloud Studio on AWS module.
 The module metadata is included in the metadata section of an AWS CloudFormation
 template.
https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/metadata-section-
structure.html
 This schema describes how the template metadata must be formatted to describe a
 module.
 The Module property is required and serves as the root of the Module Metadata.
 Any additional properties are allowed as part of AWS CloudFormation Metadata.

type: object
additionalProperties: true

required:
 - Module
```

```
properties:
 Module:
 $ref: '#/definitions/Module'

definitions:
 Module:
 title: Module
 description: Root of Modular Cloud Studio on AWS Module Metadata

 type: object
 additionalProperties: false

 required:
 - MetadataType
 - MetadataVersion
 - Revision

 properties:
 MetadataType:
 description: Indicates that this is Modular Cloud Studio on AWS Metadata
 type: string
 enum:
 - Modular Cloud Studio on AWS
 MetadataVersion:
 title: Metadata version
 description: >-
 Version of this metadata. 2024-01-23 is the only supported version.
 type: string
 enum:
 2024-01-23
 Revision:
 title: Revision identifier
 description: Semantic version that is unique from all other revisions.
 type: string
 Inputs:
 title: Inputs - optional
 description: Parameters this module depends on from other modules.
 type: array
 minItems: 1
 items:
 $ref: '#/definitions/Input'
 Outputs:
 title: Outputs - optional
 description: Parameters this module creates to share with other modules.
```

```
type: array
minItems: 1
items:
$ref: '#/definitions/Output'

Input:
title: Input
description: Required input parameter from SSM parameter store.
type: object
additionalProperties: false
required:
- Name
- Type
properties:
Name:
$ref: '#/definitions/ParameterName'
Type:
$ref: '#/definitions/ParameterType'
Remote: $ref: '#/definitions/RemoteParameter
Description:
type: string
minLength: 1
maxLength: 1024

Output:
title: Output
description: Output parameter this modules creates in SSM parameter store.
type: object
additionalProperties: false
required:
- Name
- Type
properties:
Name:
$ref: '#/definitions/ParameterName'
Type:
$ref: '#/definitions/ParameterType'
Description:
type: string
minLength: 1
maxLength: 1024

ParameterName:
title: Name
```

```
 description: >-
 The name of the parameter this module creates or depends on. Parameters are AWS
 Systems Manager (SSM) parameters so the names follow the constraints for a SSM
 parameter name.

 This name is almost a fully qualified name. Each Modular Cloud Studio on AWS
 deployment generates a unique deployment ID that must be used as the root of the
 parameter name. This name includes only the part of the fully qualified name that
 follows the deployment ID. For example, if the fully qualified parameter name is as
 follows...

 '/mcs-123abc46def/Network/VPC/vpc-id'


 ...this name value should be as follows...


 '/Network/VPC/vpc-id'


 It must begin with a slash character ('/') followed the category, then a slash
 character ('/'), then the rest of the name.
 type: string
 minLength: 2
 maxLength: 512
 pattern:
'^/(Network|Identity|WorkstationManagement|Storage|PixelStreaming|Core)(/[a-zA-
Z0-9_.-]+)\{1,14}$'

 ParameterType:
 title: Type
 description: Data type of the parameter.
 type: string
 maxLength: 128
 enum:
 - 'ssm:string'

 RemoteParameter:
 title: Remote
 description: Set this property to True if a Spoke module has a dependency on a
 parameter in the Hub region.
 type: boolean
 default: false
```

Module manifest schema

```
$schema: http://json-schema.org/draft-04/schema#
title: Modular Cloud Studio on AWS Module Manifest
description: >-
 A Module Manifest describes a module that can be registered with Modular Cloud
 Studio on AWS.
 It lists all available revisions of the module describing where to find them.

type: object
additionalProperties: false

required:
 - $manifest
 - $manifestVersion
 - Name
 - Description
 - Owner
 - Category
 - Revisions

properties:
 $manifest:
 title: Manifest
 description: Indicates that this is a Modular Cloud Studio on AWS Manifest
 type: string
 enum:
 - Modular Cloud Studio on AWS
 $manifestVersion:
 title: Manifest version
 description: Version of this manifest. 2024-01-23 is the only supported version.
 type: string
 enum:
 2024-01-23
 Name:
 title: Module name
 description: An easily identifiable name for the module.
 type: string
 maxLength: 8191
pattern: '^[a-zA-Z0-9]+(?:\s[a-zA-Z0-9]+)*$'
 Description:
 title: Module description
 description: >-
```

```
A description of the module that helps consumers understand what it does.
type: string
maxLength: 8191
Owner:
title: Owner
description: The person or organization that publishes this module.
type: string
maxLength: 8191
Category:
title: Module category
description: |-
One of the supported module categories:

* Network
* Identity
* WorkstationManagement
* Storage
* PixelStreaming
* Custom
type: string
enum:
- Network
- Identity
- WorkstationManagement
- Storage
- PixelStreaming
- Custom
SupportDescription:
title: Support description - optional
description: >-
The description of how users should use the email contact and support link.
type: string
maxLength: 8191
SupportEmail:
title: Support email - optional
description: The email address to report issues with the module.
type: string
maxLength: 254
SupportUrl:
title: Support URL - optional
description: >-
The URL to a site where users can find support information or file tickets.
type: string
pattern: '^https?://'
```

```
 maxLength: 2083
 Revisions:
 title: Module revisions
 description: List of all available module revisions.
 type: array
 minItems: 1
 uniqueItems: true
 items:
 $ref: '#/definitions/ModuleRevision'

definitions:
 ModuleRevision:
 description: >-
 Details about where to find a specific revision of the module.

 type: object
 additionalProperties: false

 required:
 - Revision
 - SupportedRegions
 - Compatibility

minProperties: 4
maxProperties: 4
additionalProperties: false

 properties:
 Revision:
 title: Revision identifier
 description: Semantic version that is unique from all other revisions. Check https://
regex101.com/r/vkijKf/1/ for more information
 type: string
 pattern: '^(0|[1-9]\\d*)(\\.(0|[1-9]\\d*)){1,2}$'
 TemplateUrl:
 title: Template URL
 description: |-
 The URL of the AWS CloudFormation template in Amazon S3.
 pattern: '^https://'
 maxLength: 2083
 TemplateUrls:
 title: Template URLs
 description: >-
```

```
The URLs of the AWS CloudFormation templates for hub and spoke modules in Amazon S3.
The Spoke property is optional and not used for hub-only modules.
type: object
required:
- Hub
properties:
Hub:
title: Hub Template URL
description: >-
The URL of the AWS CloudFormation template for the hub module in Amazon S3.
type: string
pattern: ^https://
maxLength: 2083
Spoke:
title: Spoke Template URL
description: >-
The URL of the AWS CloudFormation template for the spoke module in Amazon S3.
type: string
pattern: ^https://
maxLength: 2083
SupportedRegions:
title: Supported regions
description: >-
A module may depend on AWS services that are not available in all AWS
regions. A module must explicitly specify which AWS regions are supported.
type: array
minItems: 1
uniqueItems: true
items:
type: string
Compatibility:
$ref: '#/definitions/Compatibility'

Compatibility:
title: Compatibility
description: >-
If a revision is compatible with specific versions of Modular Cloud Studio
on AWS, it can optionally include a compatibility specification.
type: object
additionalProperties: false
required:
- MinimumMcsVersion
- MaximumMcsVersion
properties:
```

```
MinimumMcsVersion:
title: Minimum Modular Cloud Studio on AWS version
description: >-
Semantic Versioning identifier of the earliest version of Modular Cloud
Studio on AWS that this revision of the module is compatible with.
type: string
minLength: 1
pattern: '^(0|[1-9]\d*)(\.(0|[1-9]\d*)){1,2}$'
MaximumMcsVersion:
title: Maximum Modular Cloud Studio on AWS version
description: >-
Semantic Versioning identifier of the latest version of Modular Cloud
Studio on AWS that this revision of the module is compatible with.
type: string
minLength: 1
pattern: '^(0|[1-9]\d*)(\.(0|[1-9]\d*)){1,2}$'
```

Module parameters

Managed VPC - Hub

```
Outputs:

- SSM parameter store
  - /Network/VpcId
  - /Network/VpcCidr
  - /Network/PrivateSubnet1/AZ
  - /Network/PrivateSubnet1/SubnetID
  - /Network/PrivateSubnet1/RouteTableID
  - /Network/PublicSubnet1/AZ
  - /Network/PublicSubnet1/SubnetID
  - /Network/PublicSubnet1/RouteTableID
  - /Network/PrivateSubnet2/AZ
  - /Network/PrivateSubnet2/SubnetID
  - /Network/PrivateSubnet2/RouteTableID
  - /Network/PublicSubnet2/AZ
  - /Network/PublicSubnet2/SubnetID
  - /Network/PublicSubnet2/RouteTableID
```

Managed VPC - Spoke

```
Inputs:
```

```
- SSM parameter store
  - /Core/MCSStack/Name
  - /Core/HubRegion
  - /Network/VpcId (Remote)
  - /Network/VpcCidr (Remote)

Outputs:

- SSM parameter store
  - /Network/VpcId
  - /Network/VpcCidr
  - /Network/PrivateSubnet1/AZ
  - /Network/PrivateSubnet1/SubnetID
  - /Network/PrivateSubnet1/RouteTableID
  - /Network/PublicSubnet1/AZ
  - /Network/PublicSubnet1/SubnetID
  - /Network/PublicSubnet1/RouteTableID
  - /Network/PrivateSubnet2/AZ
  - /Network/PrivateSubnet2/SubnetID
  - /Network/PrivateSubnet2/RouteTableID
  - /Network/PublicSubnet2/AZ
  - /Network/PublicSubnet2/SubnetID
  - /Network/PublicSubnet2/RouteTableID
```

Unmanaged VPC

```
Inputs:

- SSM Parameter Store
  - /Core/HubRegion
  - /Core/MCSStack/Name

Outputs:

- SSM Parameter Store
  - /Network/VpcId
  - /Network/VpcCidr
  - /Network/PrivateSubnet1/AZ
  - /Network/PublicSubnet1/AZ
  - /Network/PrivateSubnet1/SubnetID
  - /Network/PublicSubnet1/SubnetID
  - /Network/PrivateSubnet1/RouteTableID
  - /Network/PublicSubnet1/RouteTableID
  - /Network/PrivateSubnet2/AZ
  - /Network/PublicSubnet2/AZ
  - /Network/PrivateSubnet2/SubnetID
  - /Network/PublicSubnet2/SubnetID
  - /Network/PrivateSubnet2/RouteTableID
  - /Network/PublicSubnet2/RouteTableID
```

Managed Active Directory - Hub

```
Inputs:

- SSM Parameter Store
  - /Network/VpcId
  - /Network/PrivateSubnet1/AZ
  - /Network/PrivateSubnet1/SubnetID
  - /Network/PrivateSubnet2/AZ
  - /Network/PrivateSubnet2/SubnetID

Outputs:

- SSM Parameter Store
  - /Identity/ActiveDirectoryId
  - /Identity/ActiveDirectoryServerIP1
  - /Identity/ActiveDirectoryServerIP2
  - /Identity/ActiveDirectoryDomainName
  - /Identity/McsModulesActiveDirectorySecretArn
- Secrets Manager
  - /Identity/DefaultAdminActiveDirectoryLoginCredentials
  - /Identity/StudioAdminActiveDirectoryLoginCredentials
  - /Identity/AdConnectorServiceAccountActiveDirectoryLoginCredentials
  - /Identity/McsModulesServiceAccountActiveDirectoryLoginCredentials
```

Managed Active Directory - Spoke

```
Inputs:

- SSM Parameter Store
  - /Core/HubRegion
  - /Network/VpcId
  - /Network/PrivateSubnet1/AZ
  - /Network/PrivateSubnet2/AZ
  - /Network/PrivateSubnet1/SubnetID
  - /Network/PrivateSubnet2/SubnetID
  - /Network/PrivateSubnet1/RouteTableID
  - /Network/PrivateSubnet2/RouteTableID
  - /Network/VpcCidr (Remote)
  - /Identity/ActiveDirectoryId (Remote)
  - /Identity/ActiveDirectoryServerIP1 (Remote)
  - /Identity/ActiveDirectoryServerIP2 (Remote)
  - /Identity/ActiveDirectoryDomainName (Remote)
- Secrets Manager
  - /Identity/McsModulesServiceAccountActiveDirectoryLoginCredentials (Remote)

Outputs:

- SSM Parameter Store
  - /Identity/ActiveDirectoryId
  - /Identity/ActiveDirectoryServerIP1
  - /Identity/ActiveDirectoryServerIP2
  - /Identity/ActiveDirectoryDomainName
  - /Identity/McsModulesActiveDirectorySecretArn
- Secrets Manager
  - /Identity/McsModulesServiceAccountActiveDirectoryLoginCredentials
```

Unmanaged Active Directory

```
Outputs:

- SSM Parameter Store
  - /Identity/ActiveDirectoryId
  - /Identity/ActiveDirectoryServerIP1
  - /Identity/ActiveDirectoryServerIP2
  - /Identity/ActiveDirectoryDomainName
  - /Identity/McsModulesActiveDirectorySecretArn
  - /Identity/Region
- Secrets Manager
  - /Identity/McsModulesServiceAccountActiveDirectoryLoginCredentials
```

Leostream Broker - Hub

```
Inputs:

- SSM Parameter Store
  - /Core/Tag/Key
  - /Core/Tag/Value/Linux
  - /Core/Tag/Value/Windows
  - /Network/VpcId
  - /Network/VpcCidr
  - /Network/PrivateSubnet1/AZ
  - /Network/PrivateSubnet1/SubnetID
  - /Network/PrivateSubnet1/RouteTableID
  - /Network/PrivateSubnet2/AZ
  - /Network/PrivateSubnet2/SubnetID
  - /Network/PrivateSubnet2/RouteTableID
  - /Identity/ActiveDirectoryServerIP1
  - /Identity/ActiveDirectoryServerIP2
  - /Identity/ActiveDirectoryDomainName
  - /Identity/McsModulesActiveDirectorySecretArn

Outputs:

- SSM Parameter Store
  - /WorkstationManagement/Leostream/DNSName
  - /WorkstationManagement/CustomResource/AmiAutomationLambda/ARN
  - /WorkstationManagement/ImageBuilder/InstanceProfile/Name
  - /WorkstationManagement/Leostream/Database/Credentials
  - /WorkstationManagement/Workstation/Windows/AMI-Id
  - /WorkstationManagement/Workstation/Linux/AMI-Id
  - /WorkstationManagement/Leostream/API/ServiceUserCredentials/SecretArn
  - /WorkstationManagement/Leostream/BrokerInstanceRoleArn
  - /WorkstationManagement/Leostream/RDS/ServiceUserCredentials/SecretArn
  - /WorkstationManagement/Leostream/BrokerHostedZoneId
  - /WorkstationManagement/Leostream/BrokerHostedZoneId
  - /WorkstationManagement/Leostream/BrokerSecurityGroupId
  - /WorkstationManagement/WorkstationSecurityGroupId
  - /WorkstationManagement/Leostream/DatabaseSecurityGroupId
  - /WorkstationManagement/Workstation/Windows/AMI-Deployed
  - /WorkstationManagement/Workstation/Linux/AMI-Deployed
  - /WorkstationManagement/Leostream/BrokerLoadBalancerArn
  - /WorkstationManagement/Leostream/BrokerLoadBalancerSecurityGroupId
  - /WorkstationManagement/Leostream/BrokerHttpsListenerArn
- Secrets Manager
  - /WorkstationManagement/Leostream/Console/AdminUserCredentials
  - /WorkstationManagement/Leostream/API/ServiceUserCredentials
```

Leostream Broker - Spoke

```
Inputs:
```

```
- SSM Parameter Store
  - /Core/HubRegion
  - /Network/VpcId
  - /Network/VpcCidr
  - /Network/PrivateSubnet1/AZ
  - /Network/PrivateSubnet1/SubnetID
  - /Network/PrivateSubnet1/RouteTableID
  - /Network/PrivateSubnet2/AZ
  - /Network/PrivateSubnet2/SubnetID
  - /Network/PrivateSubnet2/RouteTableID
  - /Identity/ActiveDirectoryServerIP1
  - /Identity/ActiveDirectoryServerIP2
  - /Identity/ActiveDirectoryDomainName
  - /Identity/ActiveDirectorySecretArn
  - /WorkstationManagement/Leostream/DNSName (Remote)
  - /WorkstationManagement/ImageBuilder/InstanceProfile/Name (Remote)
  - /WorkstationManagement/Workstation/Windows/AMI-Id (Remote)
  - /WorkstationManagement/Workstation/Linux/AMI-Id (Remote)
  - /WorkstationManagement/Leostream/BrokerInstanceRoleArn (Remote)
  - /WorkstationManagement/Leostream/BrokerSecurityGroupId (Remote)
  - /WorkstationManagement/WorkstationSecurityGroupId (Remote)
  - /WorkstationManagement/Leostream/DatabaseSecurityGroupId (Remote)
  - /WorkstationManagement/Workstation/Windows/AMI-Deployed (Remote)
  - /WorkstationManagement/Workstation/Linux/AMI-Deployed (Remote)
  - /WorkstationManagement/Leostream/BrokerHostedZoneId (Remote)
- Secrets Manager
  - /WorkstationManagement/Leostream/API/ServiceUserCredentials (Remote)
  - /WorkstationManagement/Leostream/Database/Credentials (Remote)
  - /WorkstationManagement/Leostream/Console/AdminUserCredentials (Remote)

Outputs

- SSM Parameter Store
  - /WorkstationManagement/Leostream/DNSName
  - /WorkstationManagement/ImageBuilder/InstanceProfile/Name
  - /WorkstationManagement/Leostream/API/ServiceUserCredentials/SecretArn
  - /WorkstationManagement/Leostream/RDS/ServiceUserCredentials/SecretArn
  - /WorkstationManagement/Leostream/Console/AdminUserCredentials/SecretArn
- Secrets Manager
  - /WorkstationManagement/Leostream/API/ServiceUserCredentials
  - /WorkstationManagement/Leostream/Database/Credentials
  - /WorkstationManagement/Leostream/Console/AdminUserCredentials
```

Leostream Gateway with Amazon DCV - Hub

```
Inputs:

- SSM Parameter Store
  - /WorkstationManagement/Leostream/DNSName
  - /WorkstationManagement/Leostream/API/ServiceUserCredentials/SecretArn
  - /WorkstationManagement/Leostream/RDS/ServiceUserCredentials/SecretArn
  - /WorkstationManagement/ImageBuilder/InstanceProfile/Name
  - /Network/VpcId
  - /Network/VpcCidr
  - /Network/PrivateSubnet1/AZ
  - /Network/PrivateSubnet1/SubnetID
  - /Network/PrivateSubnet1/RouteTableID
  - /Network/PrivateSubnet2/AZ
  - /Network/PrivateSubnet2/SubnetID
  - /Network/PrivateSubnet2/RouteTableID
  - /WorkstationManagement/Leostream/BrokerLoadBalancerArn
  - /WorkstationManagement/Leostream/BrokerLoadBalancerSecurityGroupId
  - /WorkstationManagement/Leostream/BrokerHttpsListenerArn
  - /Identity/ActiveDirectoryDomainName
- Secrets Manager
  - /Identity/StudioAdminActiveDirectoryLoginCredentials
  - /WorkstationManagement/Leostream/Console/AdminUserCredentials

Outputs:

- SSM Parameter Store
  - /PixelStreaming/AmazonDcv/PublicDomain
  - /PixelStreaming/AmazonDcv/LeostreamGateway/AMI-Id
```

Leostream Gateway with Amazon DCV - Spoke

```
Inputs:

- SSM Parameter Store
  - /Core/HubRegion
  - /Network/VpcId
  - /Network/VpcCidr
  - /Network/PrivateSubnet1/AZ
  - /Network/PrivateSubnet1/SubnetID
  - /Network/PrivateSubnet1/RouteTableID
  - /Network/PrivateSubnet2/AZ
  - /Network/PrivateSubnet2/SubnetID
  - /Network/PrivateSubnet2/RouteTableID
  - /WorkstationManagement/Leostream/DNSName
  - /WorkstationManagement/Leostream/API/ServiceUserCredentials/SecretArn
  - /WorkstationManagement/Leostream/RDS/ServiceUserCredentials/SecretArn
  - /PixelStreaming/AmazonDcv/PublicDomain (Remote)
  - /PixelStreaming/AmazonDcv/LeostreamGateway/AMI-Id (Remote)

Outputs:

- SSM Parameter Store
  - /PixelStreaming/AmazonDcv/PublicDomain
```

FSx for Windows File Server

```
Inputs:

- SSM Parameter Store
  - /Network/VpcId
  - /Network/VpcCidr
  - /Network/PrivateSubnet1/AZ
  - /Network/PrivateSubnet1/SubnetID
  - /Network/PrivateSubnet1/RouteTableID
  - /Network/PrivateSubnet2/AZ
  - /Network/PrivateSubnet2/SubnetID
  - /Network/PrivateSubnet2/RouteTableID
  - /Identity/ActiveDirectoryServerIP1
  - /Identity/ActiveDirectoryServerIP2
  - /Identity/ActiveDirectoryDomainName
- Secrets Manager
  - /Identity/McsModulesServiceAccountActiveDirectoryLoginCredentials

Outputs:

- SSM Parameter Store
  - /Storage/FSxWindowsFile/FSxWindowsname
  - /Storage/FSxWindowsFile/FSxResourceARN
```

Spoke Region Infrastructure

```
Inputs:

- SSM Parameter Store
```

```
  - /Core/MCSStack/Name (Remote)
  - /Core/AdminEmail (Remote)
  - /Core/Tag/Key (Remote)
  - /Core/Tag/Value/Linux (Remote)
  - /Core/Tag/Value/Windows (Remote)

Outputs:

- SSM Parameter Store
  - /Core/HubOrSpoke
  - /Core/HubRegion
  - /Core/Tag/Key
  - /Core/Tag/Value/Linux
  - /Core/Tag/Value/Windows
  - /Core/MCSStack/Name
  - /Core/MyApplication/Tag
  - /Core/AdminEmail
```

API reference

This section provides an API reference for the MCS solution.

POST /modules/deregistered

Body parameter schema

```
{
 "module_name": "string"
}
```

Parameters

Name	In	Type	Required	Description
body	body	object	true	none
⇒ **module_name**	body	string	false	none

POST /modules/disabled

Body parameter schema

Parameters

```
{
  "disable": {
    "name": "string",
    "servicecatalogProvisionedProductId": "string",
    "moduleRegion": "string",
    "regionType": "string"
  }
}
```

Name	In	Type	Required	Description
body	body	object	true	none
⇒ disable	body	object	false	none
⇒⇒ name	body	string	false	none
⇒⇒ servicecatalogProvisionedProductId	body	string	false	none
⇒⇒ moduleRegion	body	string	false	none
⇒⇒ regionType	body	string	false	none

GET /modules/enabled

Response schema

Status code 200

Name	Type	Required	Restrictions	Description
⇒ **name**	string	false	none	none
⇒ **servicecatalogProvisionedProductId**	string	false	none	none
⇒ **version**	string	false	none	none
⇒ **region**	string	false	none	none
⇒ **category**	string	false	none	none
⇒ **lastUpdateTime**	string	false	none	none
⇒*creationTime*	string	false	none	none
⇒ **status**	string	false	none	none
⇒ **cloudformationInputParameters**	[object]	false	none	none
⇒⇒ **Value**	string	false	none	none
⇒⇒ **Key**	string	false	none	none
⇒ **consoleUrl**	string	false	none	none
⇒ **stackId**	string	false	none	none
⇒ **activeDependents**	[string]	false	none	none
⇒ **regionType**	string	false	none	none

Name	Type	Required	Restrictions	Description
⇒ **moduleRegionCategory**	string	false	none	none

POST /modules/enabled

Body parameter schema

```
{
  "module": {
    "name": "string",
    "version": "string",
    "region": "string",
    "regionType": "string",
    "tags": {
      "useMCSTags": true,
      "customTags": [
        {
          "Key": "string",
          "Value": "string"
        }
      ]
    },
    "inputParameters": [
      {
        "Key": "string",
        "Value": "string"
      }
    ]
  }
}
```

Parameters

Name	In	Type	Required	Description
body	body	object	true	none
⇒ module	body	object	false	none

Name	In	Type	Required	Description
⇒⇒ name	body	string	false	none
⇒⇒ version	body	string	false	none
⇒⇒ region	body	string	false	none
⇒⇒ regionType	body	string	false	none
⇒⇒ tags	body	object	false	none
⇒⇒⇒ useMCSTags	body	boolean	false	If set to `true`, non-internal tags from MCS Core stack are used.
⇒⇒⇒ customTags	body	[object]	false	This field is ignored if **useMCSTags** is set to `true`. Otherwise, the tags from this field will are used.
⇒⇒⇒⇒ Key	body	string	false	none
⇒⇒⇒⇒ Value	body	string	false	none
⇒⇒ inputParameters	body	[object]	false	none
⇒⇒⇒ Key	body	string	false	none
⇒⇒⇒Value	body	string	false	none

GET /modules/partner

Example responses

200 response

Name	Meaning	Description	Schema
200	OK	200 response	Inline

Response schema

Status code 200

Name	Type	Required	Restrictions	Description
⇒ modules	[object]	false	none	none
⇒⇒ name	string	false	none	none
⇒⇒ displayName	string	false	none	none
⇒⇒ category	string	false	none	none
⇒⇒ manifestUrl	string	false	none	none
⇒⇒ status	string	false	none	none
⇒⇒ createAt	string	false	none	none
⇒⇒ updatedAt	string	false	none	none
⇒⇒ isCustom	boolean	false	none	none
⇒ last_updated_date	string	false	none	none

PUT /modules/partner/sync

Body parameter schema

Request body must be empty

Example responses

202 response

Name	Meaning	Description	Schema
202	Accepted	Partner module synchronization started successfully.	Inline
409	Conflict	Partner modules are being synchronized and/or registered by another process.	Inline
502	Bad Gateway	An error occurred with the remote server hosting the partner module manifests.	Inline

GET /modules/registered

Example responses

200 response

Name	Meaning	Description	Schema
200	OK	200 response	Inline

Response schema

Status code 200

Name	Type	Required	Restrictions	Description
⇒ name	string	false	none	none
⇒ version	string	false	none	none
⇒ status	string	false	none	none
⇒ category	string	false	none	none
⇒ serviceCatalogPortfolioId	string	false	none	none
⇒ serviceCatalogProductId	string	false	none	none
⇒ inputParametersLocal	[string]	false	none	none
⇒ inputParametersHub	[string]	false	none	none
⇒ regionType	string	false	none	none
⇒ isExternal	boolean	false	none	none

POST /modules/registered

Body parameter schema

```
{
  "params": {
    "manifestUrl": "string",
    "revision": "string"
  }
}
```

Parameters

Name	In	Type	Required	Description
body	body	object	true	none
⇒ params	body	object	false	none
⇒⇒ manifestUrl	body	string	false	none
⇒⇒ revision	body	string	false	none

Example responses

200 response

Status	Meaning	Description	Schema
200	OK	200 response	Inline

GET /modules/registered/inputs

Parameters

Name	In	Type	Required	Description
name	query	string	true	none
version	query	string	true	none
region	query	string	true	none
regionType	query	string	true	Either HUB or SPOKE

Response schema

Status code 200

Name	Type	Required	Restrictions	Description
⇒ cloudformationInputKeys	[object]	false	none	none
⇒⇒ name	string	false	none	none
⇒⇒ category	string	false	none	none
⇒⇒ constraints	object	false	none	none
⇒⇒⇒ allowedPattern	string	false	none	none
⇒⇒⇒ allowedValues	[string]	false	none	none
⇒⇒ default	string	false	none	none
⇒⇒ description	string	false	none	none
⇒ mcsTags	[object]	false	none	none
⇒⇒ key	string	false	none	none
⇒⇒ value	string	false	none	none

GET /modules/validate

Parameters

Name	In	Type	Required	Description
manifest_url	query	string	true	none

Response schema

Status code 200

Name	Type	Required	Restrictions	Description
⇒ data	object	false	none	Represents the manifest JSON file defined in the developer guide

GET /regions

Response schema

Status code 200

Name	Type	Required	Restrictions	Description
⇒ regions	[object]	false	none	none
⇒⇒ name	string	false	none	none
⇒⇒ isHub	boolean	false	none	none
⇒⇒ enablementStatus	string	false	none	none
⇒⇒ provisionedProductId	string	false	none	none
⇒⇒ dateEnabled	string	false	none	none

PUT /regions

Body parameter schema

```
{
  "region": {
    "name": "string",
    "enablementStatus": "string"
  }
}
```

Parameters

Name	In	Type	Required	Description
body	body	object	true	none
⇒ region	body	object	false	none
⇒⇒ name	body	string	false	none
⇒⇒ enablementStatus	body	string	false	none

Response schema

Status code 200

Name	Type	Required	Restrictions	Description
⇒ regions	object	false	none	none
⇒⇒ name	string	false	none	none
⇒⇒ isHub	string	false	none	none
⇒⇒ enablementStatus	string	false	none	none

Name	Type	Required	Restrictions	Description
⇒⇒ provisionedProductId	string	false	none	none
⇒⇒ dateEnabled	string	false	none	none

Reference

This solution includes information about an optional feature for collecting unique metrics for this solution and a list of builders who contributed to this solution.

Anonymized data collection

This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When invoked, the following information is collected and sent to AWS:

- **Solution ID** - The AWS solution identifier
- **Unique ID (UUID)** - Randomly generated, unique identifier for each deployment
- **Timestamp** - Data-collection timestamp
- **Example: Instance Data** - Count of the state and type of instances managed by the EC2 Scheduler in each AWS Region

Example data:

Running:{t2.micro: 2}, {m3.large: 2} Stopped:{t2.large: 1}, {m3.xlarge:3}

AWS owns the data gathered through this survey. Data collection is subject to the Privacy Notice. To opt out of this feature, complete the following steps before launching the AWS CloudFormation template.

1. Download the CloudFormation template to your local hard drive.
2. Open the CloudFormation template with a text editor.
3. Modify the CloudFormation template mapping section from:

```
AnonymizedData:
 SendAnonymizedData:
 Data: Yes
```

```
AnonymizedData:
 SendAnonymizedData:
```

```
Data: No
```

4. Sign in to the AWS CloudFormation console.
5. Select **Create stack**.
6. On the **Create stack page, Specify template** section, select **Upload a template file**.
7. Under **Upload a template file**, choose **Choose file** and select the edited template from your local hard drive.
8. Choose **Next** and follow the steps in Launch the stack

Contributors

- Akash Garg
- Brad Hong
- Colin McCoy
- David Chung
- Di Gao
- Eddie Goynes
- Eric Thoman
- James Wang
- Jiali Zhang
- Michael Nguyen
- Raul Marquez
- Ryan Love
- San Dim Ciin
- Spencer Sutton

Revisions

Date	Change
August 2025	v1.1.0 • Moved third-party modules listing to GitHub for better visibility • Added Amazon FSx for Lustre storage module • Added error messages to Status for enabled failed modules • Added custom domain name to Identity Managed AD module • Added option to enable/disable EC2 and EBS tagging in Network Import VPC module • Updated security configurations and patched vulnerabilities • Improved Leostream modules' deployment reliability • Removed limitation on single MCS deployment per region
March 2025	v1.0.0 Initial general availability (GA) release

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers, or licensors. AWS products or services are provided "as is" without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.