Administrator Guide

Amazon Nimble Studio

Amazon Nimble Studio: Administrator Guide

Table of Contents

End of support notice: On October 22, 2024, AWS will discontinue support for Amazon Nimble Studio. After October 22, 2024, you will no longer be able to access the Nimble Studio console or Nimble Studio resources.

What is Amazon Nimble Studio?

Nimble Studio provides infrastructure and centralized management for a suite of applications and services that artists can use to produce visual effects, animation, and games content in the cloud.

With Nimble Studio, you get essential tools for user and group management. You can also add and manage applications, including AWS Thinkbox and Nimble Studio File Transfer.

Nimble Studio features a unified interface that puts all of your studio resources in one place. You can onboard users, assign applications, and attach permissions specific to their job function. Nimble Studio requires no AWS experience, and you can set it up in about five minutes.

Contents

Features and benefits

Here are some of the features and benefits you get with Nimble Studio:

- Use Nimble Studio at no charge; pay only for the studio resources that your applications use.
- Centrally manage your studio, check its status, and gain high-level insights into its operation.
- Add and manage Nimble Studio applications, users and groups, and attach permissions.
- Securely manage access to studio resources with AWS Identity and Access Management (IAM) policies and roles.
- Manage sign-in in security for studio users and external identity providers with AWS IAM Identity Center (IAM Identity Center).
- Organize and easily find studio resources with tags to your studio resources.

Related applications

Nimble Studio provides applications for digital content creators to operate a cloud-based studio for producing visual effects (VFX), animation, and interactive content.

You can install these applications to your local computer or in the cloud with an Amazon Elastic Compute Cloud (Amazon EC2) instance. You can also use Amazon Simple Storage Service (Amazon S3) to safely transfer and store digital media assets. This means you can use Nimble Studio to reduce the costs of physical infrastructure, equipment, and technical staff.

Nimble Studio currently provides the following applications:

- **AWS Thinkbox**: Thinkbox software includes the render farm manager Thinkbox Deadline, and the 3D plugin, Thinkbox Krakatoa. You can use Thinkbox software to help you increase your studio's creative output on premises, in the cloud with Amazon EC2, or a combination of both. For more information, see AWS Thinkbox Products.
- **Nimble Studio File Transfer**: File Transfer accelerates media asset transfers of digital media assets to and from Amazon S3. File Transfer provides a graphical user interface, which you can use to quickly move thousands of large media files. For more information, see the What is Nimble Studio File Transfer page.

Pricing for Nimble Studio

There is no charge to set up Nimble Studio and use it to manage your studio infrastructure, users, security, and services.

However, if you set up services and applications in your studio, you might be charged for storage and other studio resources. For more information about Nimble Studio application pricing, see the individual application's pricing page.

For information about managing your AWS costs, see the AWS Cost Explorer Service and AWS Budgets.

Get started with Nimble Studio

Nimble Studio setup and deployment takes about five minutes.

After you familiarize yourself with Nimble Studio Concepts and terminology, see Getting started with Amazon Nimble Studio. In it, you'll find step-by-step instructions for deploying your studio.

Concepts and terminology for Amazon Nimble Studio

To help you get started with Amazon Nimble Studio, and understand how it works, you can refer to the key concepts and terminology in this guide.

Key features

Amazon Nimble Studio

Amazon Nimble Studio is an AWS service that enables creative studios to produce visual effects, animation, and interactive content entirely in the cloud, from storyboard sketch to final deliverable.

Amazon Nimble Studio console

The **Nimble Studio console** is a portion of the **AWS Management Console** that is devoted to our admin IT customers. This console is where admins create their cloud studio and manage many settings. For instance, the Studio manager page allows you to add or remove resources, add applications, and grant permissions to users and groups.

Amazon Nimble Studio portal

The **Nimble Studio portal** provides a user interface for day-to-day interactions with Nimble Studio applications and services. Users sign in directly to the portal with their user name and password without having to interact with the **AWS Management Console**.

Nimble Studio File Transfer

File Transfer accelerates media asset transfers of digital media assets to and from Amazon Simple Storage Service (Amazon S3). File Transfer provides a graphical user interface, which you can use to quickly move thousands of large media files. For more information, see the What is Nimble Studio File Transfer page.

AWS Thinkbox

Thinkbox software includes the render farm manager Thinkbox Deadline, and the 3D plugin, Thinkbox Krakatoa. You can use Thinkbox software to help you increase your studio's creative output on premises, in the cloud with Amazon EC2, or a combination of both. For more information, see AWS Thinkbox Products.

Key concepts and terminology

AWS managed policies

An AWS managed policy is a standalone policy that is created and administered by AWS. Standalone policy means that the policy has its own Amazon Resource Name (ARN) that includes the policy name. For example, arn:aws:iam::aws:policy/IAMReadOnlyAccess is an AWS managed policy. For more information about ARNs, see IAM ARNs.

AWS managed policies are used for granting permissions to common job functions. Job function policies are maintained and updated by AWS when new services and API operations are introduced. For example, the **AdministratorAccess** job function provides full access and permissions delegation to every service and resource in AWS. Whereas, partial-access AWS managed policies such as AmazonMobileAnalyticsWriteOnlyAccess and AmazonEC2ReadOnlyAccess can provide specific levels of access to AWS services without allowing full access. For learn more about access policies, see Understanding access level summaries within policy summaries.

AWS Management Console

The AWS Management Console is a web application that provides access to a broad collection of service consoles for managing AWS services.

Each service also includes its own console. These consoles offer a wide range of tools for cloud computing. There's even a service that helps with billing and cost management.

AWS IAM Identity Center (IAM Identity Center)

IAM Identity Center is an AWS service that makes it easy to centrally manage access to multiple AWS accounts and business applications. With IAM Identity Center, you can provide users with single sign-on access to all their assigned accounts and applications from one place. You can also centrally manage multi-account access and user permissions to all of your accounts in AWS Organizations. For more information, visit AWS IAM Identity Center FAQs.

AWS PrivateLink

AWS PrivateLink provides private connectivity between VPCs, AWS services, and your on-premises networks, without exposing your traffic to the public internet. AWS PrivateLink makes it easy to connect services across different accounts and VPCs. AWS PrivateLink is available for a monthly fee that is billed to your AWS account.

Digital Content Creation (DCC)

Digital Content Creation (DCC) refers to the category of applications that are used to produce creative content, including Blender, Nuke, Maya, and Houdini.

Regions

Nimble Studio offers eleven AWS Regions from which to choose deploy your studio. Regions are where essential studio infrastructure exists, such as your data and applications.

The Region should be located closest to your studio users. This reduces lag and improves data transfer speeds.

Studio

A studio is the top-level container for other Nimble Studio-related resources. Your cloud studio manages the Nimble Studio web portal and the connections to essential resources in your AWS account such as your VPC, user directory, and storage encryption keys.

Studio applications

Studio components are configurations within a customer's Nimble Studio that tell the service how to access resources like file systems, license servers, and render farms in your AWS account.

Nimble Studio contains a number of subtypes of studio components including a shared file system, compute farm, Active Directory, and license component. These subtypes describe resources that you would like your studio to use.

Studio resources

Studio resources is a term that encapsulates the things a studio needs in their daily operations. When describing how resources fit into the infrastructure of a cloud studio, they might be also referred to as studio components.

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value that you define.

Tags enable you to categorize your AWS resources in different ways. For example, you could define a set of tags for your account's Amazon Elastic Compute Cloud (Amazon EC2) instances that help you track each instance's owner and stack level. Tags also enable you to integrate your

organization's shared file systems and render farms with Nimble Studio, to keep your workflows uninterrupted while you move your workforce to the cloud.

With tags, you can categorize your AWS resources by purpose, owner, or environment. This is useful when you have many resources of the same type—you can quickly identify a specific resource based on the tags that you've assigned to it.

Setting up for Nimble Studio

This tutorial is for administrator users who want to set up an Amazon Nimble Studio.

The following sections will guide you through the steps that you need to complete before deploying a studio in Nimble Studio.

Contents

Set up IAM

Review the following AWS Identity and Access Management (IAM) documentation before you start.

- Security best practices in IAM
- Sign in to your AWS account as an admin user to complete the remaining setup.

Sign up for an AWS account

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

   When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign administrative access to a user, and use only the root user to perform tasks that require root user access.

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to https://aws.amazon.com/ and choosing **My Account**.

Create a user with administrative access

After you sign up for an AWS account, secure your AWS account root user, enable AWS IAM Identity Center, and create an administrative user so that you don't use the root user for everyday tasks.

Secure your AWS account root user

1. Sign in to the AWS Management Console as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

   For help signing in by using root user, see Signing in as the root user in the *AWS Sign-In User Guide*.

2. Turn on multi-factor authentication (MFA) for your root user.

   For instructions, see Enable a virtual MFA device for your AWS account root user (console) in the *IAM User Guide*.

Create a user with administrative access

1. Enable IAM Identity Center.

   For instructions, see Enabling AWS IAM Identity Center in the *AWS IAM Identity Center User Guide*.

2. In IAM Identity Center, grant administrative access to a user.

   For a tutorial about using the IAM Identity Center directory as your identity source, see Configure user access with the default IAM Identity Center directory in the *AWS IAM Identity Center User Guide*.

Sign in as the user with administrative access

- To sign in with your IAM Identity Center user, use the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

For help signing in using an IAM Identity Center user, see Signing in to the AWS access portal in the *AWS Sign-In User Guide*.

Assign access to additional users

1. In IAM Identity Center, create a permission set that follows the best practice of applying least-privilege permissions.

   For instructions, see Create a permission set in the *AWS IAM Identity Center User Guide*.
2. Assign users to a group, and then assign single sign-on access to the group.

   For instructions, see Add groups in the *AWS IAM Identity Center User Guide*.

Related resources

- Security Best Practices in IAM
- AWS service quotas - AWS General Reference

Getting started with Amazon Nimble Studio

This chapter shows how to use the Nimble Studio console to create your studio's infrastructure, confirm the AWS Region, review settings, and create your studio. You can also customize your setup with additional settings.

For first-time AWS customers, see the Setting up for Nimble Studio tutorials.

Topics

Setting up Nimble Studio

This guide shows you how to configure your infrastructure, review your settings, and create your studio. You can also customize your studio with Additional studio settings.

Step 1: Configure studio infrastructure

Your studio's infrastructure consists of the following components:

- **Studio display name**: The **Studio display name** is how you can identify your studio — for example *AnyCompany Studio*. Your studio's name also determines your **Studio portal URL**. You can change the **Studio display name** after you complete setup, at any time.
- **Studio portal URL**: You can access your studio by using the **Studio portal URL**. The URL is based on the **Studio display name** — for example *https://anycompanystudio.awsapps.com*. You can change the **Studio portal URL** after you complete setup, at any time.
- **AWS Region**: The **AWS Region** is the physical location for a collection of AWS data centers. When you set up your studio, the Region defaults to the closest location to you. You should change the Region so it is located closest to your users. This reduces lag and improves data transfer speeds.

> **Important**
>
> You can't change your Region after you finish setting up Nimble Studio.

Complete the tasks in this section to configure your studio's infrastructure.

To configure your studio's infrastructure

1. Sign in to the **AWS Management Console** and open the Nimble Studio console.
2. Choose **Setup Nimble Studio**, and then choose **Next**.
3. Enter the **Studio display name** — for example **`AnyCompany Studio`**.
4. **(Optional)** To change the **Studio portal name**, choose **Edit URL**.
5. **(Optional)** To change the **AWS Region** so it's closest to your studio users, choose **Change Region**.
   a. Select the Region closest to your users.
   b. Choose **Apply Region**.
6. **(Optional)** To further customize your studio setup, select **Additional studio settings**.
7. To review your settings before you create your studio, choose **Next**.

Step 2: Review and create your studio

After you configure your studio's infrastructure, you can review, make changes, and create your studio.

To review and create your studio

1. On the **Review and create** page, review your **Studio infrastructure**.
2. Confirm that the **AWS Region** is closest to your studio users.
3. **(Optional)** Choose **Edit** to make changes to your studio setup.
4. When you're ready, choose **Create studio**.

Additional studio settings

Nimble Studio setup includes additional studio settings. With these settings, you can view all the changes Nimble Studio setup makes to your AWS account, configure your studio user role, and change your encryption key type. You can also add optional tags to your studio resources.

Configure studio user role

An AWS service can assume a service role to perform actions on your behalf. Nimble Studio requires a studio user role for it to give users access to resources in your studio.

You can attach AWS Identity and Access Management (IAM) managed policies to the studio user role. The policies allow users to perform certain actions, such as creating jobs in a specific Nimble Studio application. Because applications depend on specific conditions in the managed policy, if you don't use the managed policies, the application might not perform as expected.

You can change the studio user role after you complete setup, at any time. For more information about user roles, see IAM Roles.

The following tabs contain instructions for two different use cases. To create and use a new service role, choose the **New service role** tab. To use an existing service role, choose the **Existing service role** tab.

New service role

To create and use a new service role

1. Select **Create and use a new service role**.
2. **(Optional)** Enter a **Service user role** name.
3. Choose **View permission details** for more information about the role.

Existing service role

To use an existing service role

1. Select **Use an existing service role**.
2. Open the dropdown list to choose an existing service role.
3. **(Optional)** Choose **View in IAM console** for more information about the role.

AWS IAM Identity Center

AWS IAM Identity Center is a cloud-based single sign-on service for managing users and groups. IAM Identity Center can also be integrated with your enterprise single sign-on (SSO) provider so that users can sign in with their company account.

Nimble Studio enables IAM Identity Center by default, and it is required to set up and use Nimble Studio. For more information, see What is AWS IAM Identity Center.

Configure AWS KMS encryption key

AWS Key Management Service (AWS KMS) keys are the primary type of KMS key that you can use to encrypt, decrypt, and re-encrypt your data.

Nimble Studio includes the following AWS KMS encryption key types:

- **AWS owned key** – AWS owned keys are KMS keys that the AWS service owns and manages for use in multiple AWS accounts. AWS owned keys do not reside in your AWS account, but Nimble Studio can use an AWS owned key to protect the resources in your account.

  To use AWS KMS, you don't need to create or maintain the key or its key policy. There is no charge to use AWS owned keys and they do not count against AWS KMS quotas for your AWS account.
- **Customer managed AWS KMS key** – A customer managed key is a KMS key in your AWS account that you create, own, and manage.

  You have full control over these KMS keys. Customer managed keys incur a monthly fee. They also incur a fee for each API request to AWS KMS beyond the free tier. For more information about AWS KMS pricing, see AWS Key Management Service pricing.

The encryption key type **cannot** be changed after you complete setup. For more information about AWS KMS and encryption key types, see the AWS KMS documentation.

To choose a different encryption key type

1. Select **Choose a different AWS KMS key (advanced)**.
2. Select an AWS KMS key or enter an Amazon resource number (ARN).
3. Choose **Create AWS KMS key**.

Configure tags

Tags act as labels for organizing your Nimble Studio resources. You can add up to 50 tags to identify, organize, filter, and search for resources.

Each tag consists two parts, which you define: a tag **Key** and an optional tag **Value** — for example, key: `domain` and value: `anycompanystudio.com`.

You can add or remove tags after you complete setup, at any time. For more information about tags, see Tagging your AWS resources.

To add tags to your studio resources

1. Choose **Add new tag**.
2. Enter the tag **Key**.
3. **(Optional)** Enter the tag **Value**.

Delete a studio

If you no longer need a studio, you can delete it. When you delete your studio, only the studio infrastructure is deleted. Your other AWS resources, such as user roles, policies, and application data remain intact.

> **Important**
>
> You **can't** restore a studio after you delete it.

To delete your studio

1. Sign in to the AWS Management Console and open the Nimble Studio console.
2. Select **Studio overview**.
3. Choose **Actions**, then select **Delete studio**.
4. Enter **`delete`**, then choose **Delete**.

Security in Amazon Nimble Studio

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from data centers and network architectures that are built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to Amazon Nimble Studio, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

> **Important**
>
> It's highly recommended that you read and familiarize yourself with the Security Pillar - AWS Well-Architected Framework. This article contains key principles to securing your AWS infrastructure.

This documentation helps you understand how to apply the shared responsibility model when using Nimble Studio. The following topics show you how to configure Nimble Studio to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Nimble Studio resources.

More Information

- Security Pillar - AWS Well-Architected Framework
- Security for the AWS Cloud Development Kit (AWS CDK) (AWS CDK)
- Security in Amazon Virtual Private Cloud
- AWS security credentials

- Security in Amazon EC2
  - Linux
  - Windows

Set up AWS account security

This guide shows how to set up your AWS account to receive notifications when your resources are compromised, and to allow specific AWS account users to access it. To secure your AWS account and track your resources, complete the following steps.

Contents

Delete your account's access keys

You can allow programmatic access to your AWS resources from the AWS Command Line Interface (AWS CLI) or with AWS APIs. However, AWS recommends that you don't create or use the access keys associated with your root account for programmatic access.

If you still have access keys, we recommend that you delete those and create a user. Then, grant that user only the permissions needed for the APIs that you're planning to call. You can use that user to issue access keys.

For more information, see Managing Access Keys for Your AWS account in the *AWS General Reference guide*.

Enable multi-factor authentication

Multi-factor authentication (MFA) is a security capability that provides a layer of authentication in addition to your user name and password.

MFA works like this: After you sign in with your user name and password, you must also provide an additional piece of information that only you have physical access to. This information can come from a dedicated MFA hardware device, or from an app on a phone.

You must select the type of MFA device that you want to use from the list of supported MFA devices. For a hardware device, keep the MFA device in a secure location.

If you use a virtual MFA device (like a phone app), think about what might happen if your phone is lost or damaged. One approach is to keep the virtual MFA device that you use in a safe place. Another option is to activate more than one device at the same time, or use a virtual MFA option for device key recovery.

To learn more about MFA, see Enabling a Virtual Multi-Factor Authentication (MFA) Device.

Related resources

- Getting Started with Multi-Factor Authentication
- Securing Access to AWS Using MFA

Enable CloudTrail in all AWS Regions

You can track all activity in your AWS resources by using AWS CloudTrail. We recommend that you turn on CloudTrail now. This can help AWS Support and your AWS solutions architect troubleshoot a security or configuration issue, later.

To enable CloudTrail logging in all AWS Regions, see AWS CloudTrail Update – Turn On in All Regions and Use Multiple Trails.

To learn more about CloudTrail, see Turn On CloudTrail: Log API Activity in Your AWS account. To learn how CloudTrail monitors Nimble Studio, see Logging Nimble Studio calls using AWS CloudTrail.

Set up Amazon GuardDuty and notifications

Amazon GuardDuty is a continuous security monitoring service that analyzes and processes the following:

- Data sources
- Amazon VPC Flow Logs
- AWS CloudTrail management event logs
- CloudTrail S3 data event logs
- DNS logs

Amazon GuardDuty identifies unexpected and potentially unauthorized and malicious activity within your AWS environment. Malicious activity can include issues like escalations of privileges, uses of exposed credentials, or communication with malicious IP addresses or domains. To identify these activities, GuardDuty uses threat intelligence feeds, such as lists of malicious IP addresses and domains, and machine learning. For example, GuardDuty can detect compromised Amazon EC2 instances serving malware or mining bitcoin.

GuardDuty also monitors AWS account access behavior for signs of compromise. This includes unauthorized infrastructure deployments, like instances deployed in an AWS Region that has never been used. It also includes unusual API calls, like a password policy change to reduce password strength.

GuardDuty informs you of the status of your AWS environment by producing security findings. You can view these findings in the GuardDuty console or through Amazon CloudWatch events.

Set up an Amazon SNS topic and endpoint

Follow the instructions in the Setup an Amazon SNS topic and endpoint tutorial.

Set up an EventBridge event for GuardDuty findings

Create a rule for EventBridge to send events for all findings that GuardDuty generates.

To create an EventBridge event for GuardDuty findings

1. Sign in to the Amazon EventBridge console: https://console.aws.amazon.com/events/
2. In the navigation pane, choose **Rules**. Then choose **Create rule**.
3. Enter a **Name** and **Description** for the new rule. Then choose **Next**.
4. Leave **AWS events or EventBridge partner events** selected for **Event source**.
5. In **Event pattern**, choose **AWS services** for the **Event source**. Then **GuardDuty** for the **AWS services**, and **GuardDuty Finding** for the **Event type**. This is the topic that you created in Set up an Amazon SNS topic and endpoint.
6. Choose **Next**.
7. For **Target 1**, select **AWS service**. Choose **SNS topic** in the **Select a target** dropdown. Then choose your **GuardDuty_to_Email** topic.
8. In the **Additional settings** section: Use the **Configure target input** dropdown to choose **Input transformer**. Select **Configure input transformer**.
9. Enter the following code into the **Input path** field in the **Target input transformer** section.

```
{
    "severity": "$.detail.severity",
    "Account_ID": "$.detail.accountId",
    "Finding_ID": "$.detail.id",
    "Finding_Type": "$.detail.type",
    "region": "$.region",
    "Finding_description": "$.detail.description"
}
```

10. To format the email, enter the following code into the **Template** field.

```
"AWS <Account_ID> has a severity <severity> GuardDuty finding type <Finding_Type>
 in the <region> region."
"Finding Description:"
"<Finding_description>. "
"For more details open the GuardDuty console at https://console.aws.amazon.com/
guardduty/home?region=<region>#/findings?search=id=<Finding_ID>"
```

11. Choose **Create**. Then choose **Next**.
12. (Optional) Add tags if you're using tags to track your AWS resources.
13. Choose **Next**.
14. Review your rule. Then choose **Create rule**.

Now that you've set up your AWS account security, you can grant access to specific users and receive notifications when your resources are compromised.

Data protection in Amazon Nimble Studio

The AWS shared responsibility model applies to data protection in Amazon Nimble Studio. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM).

That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with Nimble Studio or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

The AWS shared responsibility model applies to data protection in Amazon Nimble Studio. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You're responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use.

For more information about data privacy, see the Data Privacy FAQ. For information about data protection in the European Union, visit the GDPR Center.

Encryption at rest

Nimble Studio protects sensitive studio data by encrypting it at rest using encryption keys stored in AWS Key Management Service (AWS KMS). Encryption at rest is available in all AWS Regions where Nimble Studio is available. The studio data that we encrypt includes the name and descriptions

of all resource types, as well as studio component scripts, script parameters, mount points, share names, and other data.

Encrypting data means that sensitive data that is saved on disks isn't readable by any user or application without a valid key. Encrypted data can be securely stored at rest and can be decrypted only by a party with authorized access to the managed key.

For information about how Nimble Studio uses AWS KMS for encrypting data at rest, see Key management for Amazon Nimble Studio.

Using grants with AWS KMS keys

A grant is a policy instrument that allows AWS principals to use AWS KMS keys in cryptographic operations. It can also let them view a KMS key with the command `DescribeKey`, and create and manage grants.

Grants are commonly used by AWS services that integrate with AWS KMS to encrypt your data at rest. The service creates a grant on behalf of a user in the account, uses its permissions, and retires the grant as soon as its task is complete.

When Nimble Studio creates your studio, we provide two roles for Nimble Studio portal users: user and admin roles. Nimble Studio creates grants on customer managed keys for these roles to provides them access to studio encrypted data.

> **Important**
>
> If you delete a grant, the Nimble Studio portal will be unusable for users, until the admin creates a new grant.

For details about how AWS services use grants, see How AWS services use AWS KMS or the Encryption at rest topic in the service's user guide or developer guide.

Encryption in transit

The following table provides information about how data is encrypted in transit. Where applicable, other data protection methods for Nimble Studio are also listed.

Data	Network path	Protection

Web assets such as images and JavaScript files	The network path is between Nimble Studio users and Nimble Studio.	Data encryption uses TLS 1.2 or later.
Pixel and related streaming traffic	The network path is between Nimble Studio users and Nimble Studio.	Encrypted using 256-bit Advanced Encryption Standard (AES-256), and transported using TLS 1.2 or later.
API traffic	The path is between Nimble Studio users and Nimble Studio.	Encrypted using TLS 1.2 or later. Requests to create a connection are signed using SigV4.

Key management for Amazon Nimble Studio

When creating a new studio, you can choose one of the following keys to encrypt your studio data:

- AWS owned KMS key – Default encryption type. The key is owned by Nimble Studio (no additional charge).
- Customer managed KMS key – The key is stored in your account and is created, owned, and managed by you. You have full control over the key. AWS KMS charges apply.

Deleting a customer managed KMS key in AWS Key Management Service (AWS KMS) is destructive and potentially dangerous. It irreversibly deletes the key material and all metadata associated with the key. After a customer managed KMS key is deleted, you can no longer decrypt the data that was encrypted by that key. This means that the data becomes unrecoverable.

This is why AWS KMS gives customers a waiting period of up to 30 days before deleting the key. The default waiting period is 30 days.

About the waiting period

Because it's destructive and potentially dangerous to delete a customer managed KMS key, we require you to set a waiting period of 7 – 30 days. The default waiting period is 30 days.

However, the actual waiting period might be up to 24 hours longer than the one you scheduled. To get the actual date and time when the key will be deleted, use the DescribeKey operation. You can also see the scheduled deletion date of a key in the AWS KMS console on the key's detail page, in the **General configuration** section. Notice the time zone.

During the waiting period, the customer managed key's status and key state is **Pending deletion**.

- A customer managed KMS key that is pending deletion can't be used in any cryptographic operations.
- AWS KMS doesn't rotate the backing keys of customer managed AWS KMS keys that are pending deletion.

For more information about deleting a customer managed AWS KMS key see Deleting customer master keys.

Data security measures

For data protection purposes, we recommend that you protect AWS account credentials and set up individual accounts with AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put sensitive identifying information, such as customer account numbers, into free-form fields such as a **Name** field. This includes when you work with Amazon Nimble Studio or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into Amazon Nimble Studio or other services might get picked up for inclusion in diagnostic logs. When you provide a URL to an external server, don't include credentials information in the URL to validate your request to that server.

Diagnostic data and metrics

During the deployment and deletion of StudioBuilder, Amazon Nimble Studio collects certain metrics that we use to diagnose issues and improve Nimble Studio's features and user experience.

Types of metrics collected

- Usage information – The generic commands and subcommands that are run.
- Errors and diagnostic information – The status and duration of commands that are run, including exit codes, internal exception names, and failures.
- System and environment information – The Python version, operating system (Windows, Linux, or macOS), and environment in which StudioBuilder is run.

Identity and Access Management for Amazon Nimble Studio

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. Administrators control who can be **authenticated** (signed in) and **authorized** (have permissions) to use Amazon Nimble Studio resources. IAM is an AWS service that you can use with no additional charge.

Topics

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Nimble Studio.

Service user – If you use the Nimble Studio service to do your job, then you are a service user. In this case, your administrator will provide you with the credentials and permissions that you need to access your assigned resources. As you use more Nimble Studio features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you can't access a feature in Nimble Studio, see Troubleshooting Amazon Nimble Studio identity and access.

Service administrator – If you're in charge of Nimble Studio resources at your company, you probably have full access to Nimble Studio. It's your job to determine which Nimble Studio features and resources your employees should access. Then, submit requests to your administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Nimble Studio, see How Amazon Nimble Studio works with IAM.

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. For more information about signing in using the AWS Management Console, see Signing in to the AWS Management Console as an IAM user or root user in the **IAM User Guide**.

You need to be **authenticated** (signed in to AWS) as the AWS account root user, a user, or by assuming an IAM role. You can also use your company's single sign-on authentication or even sign in using Google or Facebook. In these cases, your administrator previously set up identity federation using IAM roles. When you access AWS using credentials from another company, you're assuming a role indirectly.

To sign in directly to the AWS Management Console, use your password with your root user email address or your username. You can access AWS programmatically using your root user or user access keys.

AWS provides SDK and command line tools to cryptographically sign your request using your credentials. If you don't use AWS tools, sign the request yourself. Do this using **Signature Version 4**, a protocol for authenticating inbound API requests. For more information about authenticating requests, see Signature Version 4 signing process in the AWS General Reference .

Regardless of the authentication method that you use, you might also be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Using multi-factor authentication (MFA) in AWS in the IAM User Guide.

AWS account root user

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account **root user** and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

Users and groups

A **user** is an identity within your AWS account that has specific permissions for a single person or application. A user can have long-term credentials or a set of access keys. To learn how to generate access keys, see Managing access keys for IAM users in the IAM User Guide. When you generate access keys for a user, view and securely save the key pair. You can't recover the secret access key in the future. Instead, generate a new access key pair.

An **IAM group** is an identity that specifies a collection of users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named **IAMAdmins** and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see When to create a user (instead of a role) in the IAM User Guide.

IAM roles

An **IAM role** is an identity within your AWS account that has specific permissions. It is similar to a user, but isn't associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles in the IAM User Guide.

IAM roles with temporary credentials are useful in the following situations:

- **Temporary user permissions** – A user can assume an IAM role to temporarily take on different permissions for a specific task.

- **Federated user access** – Instead of creating a user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as **federated users**. AWS assigns a role to a federated user when access is requested through an identity provider. For more information about federated users, see Federated users and roles in the IAM User Guide.
  - **Membership** – Nimble Studio uses a concept called 'membership' to provide a user access to a particular launch profile. Membership allows studio administrators to delegate resource access to users, without having to write, or understand, IAM policies. When a Nimble Studio administrator creates a membership for a user in a launch profile, the user is authorized to perform IAM actions that are required to use a launch profile, such as viewing its properties and starting a streaming session using that launch profile.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. Service roles provide access only within your account and can't be used to grant access to services in other accounts. An administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the IAM User Guide.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Nimble Studio doesn't support service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the IAM User Guide.

To learn whether to use IAM roles or users, see When to create an IAM role (instead of a user) in the IAM User Guide.

Managing access using policies

You control access in AWS by creating policies and attaching them to IAM identities or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. You can sign in as the root user or a user, or you can assume an IAM role. When

you then make a request, AWS evaluates the related identity-based or resource-based policies. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the IAM User Guide.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and for what **conditions**.

Every IAM entity (user or role) starts with no permissions. In other words, by default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as a user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and for what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the IAM User Guide.

Identity-based policies can be further categorized as **inline policies** or **managed policies**. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choosing between managed policies and inline policies in the IAM User Guide.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM **role trust policies** and Amazon S3 **bucket policies**. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified

principal can perform on that resource and for what conditions. Specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs) in Nimble Studio

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they don't use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the **Amazon Simple Storage Service Developer Guide**.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of the entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field aren't limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the IAM User Guide.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in Organizations. Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs work in the AWS Organizations User Guide.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session

policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the IAM User Guide.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the IAM User Guide.

How Amazon Nimble Studio works with IAM

Before you use IAM to manage access to Nimble Studio, learn what IAM features are available to use with Nimble Studio.

IAM features you can use with Amazon Nimble Studio

IAM feature	Nimble Studio support
Policy actions for Nimble Studio	Yes
Policy resources for Nimble Studio	Yes
Policy condition keys for Nimble Studio	Yes
Access control lists (ACLs) in Nimble Studio	No
Attribute-based access control (ABAC) with Nimble Studio	Yes
Using temporary credentials with Nimble Studio	Yes
Cross-service principal permissions for Nimble Studio	Yes
Service roles for Nimble Studio	Yes
Service-linked roles for Nimble Studio	No

To get a high-level view of how Nimble Studio and other AWS services work with most IAM features, see AWS services that work with IAM in the IAM User Guide.

Identity-based policies for Nimble Studio

Supports identity-based policies	Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as a user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and for what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the IAM User Guide.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions for which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it's attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the IAM User Guide.

Identity-based policy examples for Amazon Nimble Studio

To view examples of Nimble Studio identity-based policies, see Identity-based policy examples for Amazon Nimble Studio.

Resource-based policies within Nimble Studio

Supports resource-based policies	No

Nimble Studio doesn't support resource-based policies or cross-account access. Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM **role trust policies** and Amazon S3 **bucket policies**. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and for what conditions. Specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Policy actions for Nimble Studio

Supports policy actions	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and for what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as **permission-only actions** that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called **dependent actions**.

Include actions in a policy to grant permissions to perform the associated operation.

To see a list of Nimble Studio actions, see Actions defined by Amazon Nimble Studio in the **Service Authorization Reference**.

Policy actions in Nimble Studio use the following prefix before the action:

```
nimble
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
      "nimble:action1",
      "nimble:action2"
         ]
```

To view examples of Nimble Studio identity-based policies, see Identity-based policy examples for Amazon Nimble Studio.

Policy resources for Nimble Studio

Supports policy resources	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and for what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as **resource-level permissions**.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To view examples of Nimble Studio identity-based policies, see Identity-based policy examples for Amazon Nimble Studio.

Policy condition keys for Nimble Studio

Supports policy condition keys	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and for what **conditions**.

The `Condition` element (or `Condition` **block**) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an user permission to access a resource only if it's tagged with their username. For more information, see IAM policy elements: variables and tags in the **IAM User Guide**.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the **IAM User Guide**.

To view examples of Nimble Studio identity-based policies, see Identity-based policy examples for Amazon Nimble Studio.

Access control lists (ACLs) in Nimble Studio

Supports ACLs	No

Nimble Studio doesn't support access control lists (ACLs). ACLs control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they don't use the JSON policy document format.

Attribute-based access control (ABAC) with Nimble Studio

Supports ABAC (tags in policies)	Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called **tags**. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they're trying to access.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

For more information about ABAC, see What is ABAC? in the IAM User Guide. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the IAM User Guide.

Using temporary credentials with Nimble Studio

Supports temporary credentials	Yes

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the **IAM User Guide**.

You're using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your

company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switching to a role (console) in the IAM User Guide.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

Cross-service principal permissions for Nimble Studio

Supports principal permissions	Yes

Service roles for Nimble Studio

Supports service roles	Yes

A service role is an IAM role that a service assumes to perform actions on your behalf. Service roles provide access only within your account and can't be used to grant access to services in other accounts. An administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the IAM User Guide.

> **Warning**
>
> Changing the permissions for a service role might break Nimble Studio functionality. Edit service roles only when Nimble Studio provides guidance to do so.

Service-linked roles for Nimble Studio

Supports service-linked roles	No

Nimble Studio doesn't support service-linked roles. A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on

your behalf. Service-linked roles appear in your IAM account and are owned by the service. An administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing service-linked roles, see AWS services that work with IAM. Find a service in the table that includes a `Yes` in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

Identity-based policy examples for Amazon Nimble Studio

By default, users and roles don't have permission to create or modify Nimble Studio resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An administrator must create IAM policies that grant users and roles permission to perform actions on the resources that they need. The administrator must then attach those policies to the users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see Creating policies on the JSON tab in the IAM User Guide.

Topics

- Policy best practices

Policy best practices

Identity-based policies are very powerful. They determine whether someone can create, access, or delete Nimble Studio resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started using AWS managed policies** – To start using Nimble Studio quickly, use AWS managed policies to give your employees the permissions they need. These policies are already available in your account and are maintained and updated by AWS. For more information, see Get started using permissions with AWS managed policies in the IAM User Guide.
- **Grant least privilege** – When you create custom policies, grant only the permissions required to perform a task. Start with a minimum set of permissions and grant additional permissions as necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later. For more information, see Grant least privilege in the **IAM User Guide**.

- **Enable MFA for sensitive operations** – For extra security, require users to use multi-factor authentication (MFA) to access sensitive resources or API operations. For more information, see Using multi-factor authentication (MFA) in AWS in the IAM User Guide.
- **Use policy conditions for extra security** – To the extent that it's practical, define the conditions that your identity-based policies allow access to a resource. For example, you can write conditions to specify a range of allowable IP addresses that a request must come from. You can also write conditions to allow requests only within a specified date or time range, or to require the use of SSL or MFA. For more information, see IAM JSON policy elements: Condition in the IAM User Guide.

AWS managed policies for Amazon Nimble Studio

To add permissions to users, groups, and roles, it's easier to use AWS managed policies than to write policies yourself. It takes time and expertise to create IAM customer managed policies that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see AWS managed policies in the *IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services don't remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the **ReadOnlyAccess** AWS managed policy provides read-only access to all AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see AWS managed policies for job functions in the *IAM User Guide*.

Your end users will access Amazon Nimble Studio primarily using the Nimble Studio portal. When creating your studio using StudioBuilder or the Nimble Studio console, one IAM role is created for each studio persona: the studio administrator and the studio user. Each has the respective IAM managed policy attached. The Nimble Studio portal provides an experience where users can only list and use the resources that they have permission to access.

The Nimble Studio portal provides an experience where users can only list and use the resources to which they have access and the portal depends on the content of these policies to operate correctly. Nimble Studio end users will use the portal to access their cloud studio. So, when admins create their studio using StudioBuilder, one IAM role is created for each person who needs to access the studio. This includes the studio administrator and the studio user, each with their respective IAM managed policy attached.

For a list and descriptions of job function policies, see AWS managed policies for job functions in the IAM User Guide.

AWS managed policy: `AmazonNimbleStudio-LaunchProfileWorker`

You can attach the `AmazonNimbleStudio-LaunchProfileWorker` policy to your IAM identities.

Attach this policy to EC2 instances created by Nimble Studio Builder to grant access to resources needed by Nimble Studio launch profile workers.

Permissions details

This policy includes the following permissions.

- ds - Allows LaunchProfile workers to discover connection information about the AWS Managed Microsoft AD associated with a LaunchProfile.
- ec2 - Allows LaunchProfile workers to discover security group and subnet information for connecting to a LaunchProfile.
- fsx - Allows LaunchProfile workers to discover connection information to Amazon FSx volumes associated with a LaunchProfile.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeSecurityGroups",
        "fsx:DescribeFileSystems",
        "ds:DescribeDirectories"
      ],
      "Resource": [
        "*"
```

```
      ],
      "Condition": {
        "StringEquals": {
          "aws:CalledViaLast": "nimble.amazonaws.com"
        }
      },
      "Sid": "GetLaunchProfileInitializationDependencies"
    }
  ],
  "Version": "2012-10-17"
}
```

AWS managed policy: AmazonNimbleStudio-StudioAdmin

You can attach the AmazonNimbleStudio-StudioAdmin policy to your IAM identities.

Attach this policy to the Admin role associated with your studio to grant access to Amazon Nimble Studio resources associated with the studio admin and related studio resources in other services.

Permissions details

This policy includes the following permissions.

- nimble - Allows Studio Users access to Nimble resources that have been delegated to them by StudioAdmins.
- sso - Allows Studio Users the ability to view the names of other users in the studio.
- identitystore - Allows Studio Users the ability to view the names of other users in the studio.
- ds - Allows Nimble Studio to add virtual workstations to the AWS Managed Microsoft AD associated with the studio.
- ec2 - Allows Nimble Studio to attach virtual workstations to your configured VPC.
- fsx - Allows Nimble Studio to connect virtual workstations to your configured Amazon FSx volumes.
- cloudwatch - Allows Nimble Studio to retrieve CloudWatch metrics.

```
{
  "Statement": [
    {
      "Sid": "StudioAdminFullAccess",
      "Effect": "Allow",
```

```
      "Action": [
        "nimble:CreateStreamingSession",
        "nimble:GetStreamingSession",
        "nimble:StartStreamingSession",
        "nimble:StopStreamingSession",
        "nimble:CreateStreamingSessionStream",
        "nimble:GetStreamingSessionStream",
        "nimble:DeleteStreamingSession",
        "nimble:ListStreamingSessionBackups",
        "nimble:GetStreamingSessionBackup",
        "nimble:ListEulas",
        "nimble:ListEulaAcceptances",
        "nimble:GetEula",
        "nimble:AcceptEulas",
        "nimble:ListStudioMembers",
        "nimble:GetStudioMember",
        "nimble:ListStreamingSessions",
        "nimble:GetStreamingImage",
        "nimble:ListStreamingImages",
        "nimble:GetLaunchProfileInitialization",
        "nimble:GetLaunchProfileDetails",
        "nimble:GetFeatureMap",
        "nimble:PutStudioLogEvents",
        "nimble:ListLaunchProfiles",
        "nimble:GetLaunchProfile",
        "nimble:GetLaunchProfileMember",
        "nimble:ListLaunchProfileMembers",
        "nimble:PutLaunchProfileMembers",
        "nimble:UpdateLaunchProfileMember",
        "nimble:DeleteLaunchProfileMember"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "sso-directory:DescribeUsers",
        "sso-directory:SearchUsers",
        "identitystore:DescribeUser",
        "identitystore:ListUsers"
      ],
      "Resource": [
        "*"
      ]
```

```
    },
    {
      "Effect": "Allow",
      "Action": [
        "ds:CreateComputer",
        "ds:DescribeDirectories",
        "ec2:DescribeSubnets",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface",
        "ec2:CreateNetworkInterfacePermission",
        "ec2:DeleteNetworkInterfacePermission",
        "ec2:DescribeSecurityGroups",
        "fsx:DescribeFileSystems"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:CalledViaLast": "nimble.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "cloudwatch:GetMetricData",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloudwatch:namespace": "AWS/NimbleStudio"
        }
      }
    }
  ],
  "Version": "2012-10-17"
}
```

AWS managed policy: `AmazonNimbleStudio-StudioUser`

You can attach the `AmazonNimbleStudio-StudioUser` policy to your IAM identities.

Attach this policy to the User role associated with your studio to grant access to Amazon Nimble Studio resources associated with the studio user and related studio resources in other services.

Permissions details

This policy includes the following permissions.

- nimble - Allows Studio Users access to Nimble resources that have been delegated to them by StudioAdmins.
- sso - Allows Studio Users the ability to view the names of other users in the studio.
- identitystore - Allows Studio Users the ability to view the names of other users in the studio.
- ds - Allows Nimble Studio to add virtual workstations to the AWS Managed Microsoft AD associated with the studio.
- ec2 - Allows Nimble Studio to attach virtual workstations to your configured VPC.
- fsx - Allows Nimble Studio to connect virtual workstations to your configured Amazon FSx volumes.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ds:CreateComputer",
        "ec2:DescribeSubnets",
        "ec2:CreateNetworkInterfacePermission",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterfacePermission",
        "ec2:DeleteNetworkInterface",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeSecurityGroups",
        "fsx:DescribeFileSystems",
        "ds:DescribeDirectories"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:CalledViaLast": "nimble.amazonaws.com"
        }
```

```
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "sso-directory:DescribeUsers",
        "sso-directory:SearchUsers",
        "identitystore:DescribeUser",
        "identitystore:ListUsers"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "nimble:ListLaunchProfiles"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "nimble:requesterPrincipalId": "${nimble:principalId}"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "nimble:ListStudioMembers",
        "nimble:GetStudioMember",
        "nimble:ListEulas",
        "nimble:ListEulaAcceptances",
        "nimble:GetFeatureMap",
        "nimble:PutStudioLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "nimble:StartStreamingSession",
        "nimble:StopStreamingSession",
        "nimble:DeleteStreamingSession",
```

```
        "nimble:GetStreamingSession",
        "nimble:CreateStreamingSessionStream",
        "nimble:GetStreamingSessionStream",
        "nimble:ListStreamingSessions"
        "nimble:ListStreamingSessionBackups",
        "nimble:GetStreamingSessionBackup"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "nimble:ownedBy": "${nimble:requesterPrincipalId}"
        }
      }
    }
  ],
  "Version": "2012-10-17"
}
```

Nimble Studio updates to AWS managed policies

View details about updates to AWS managed policies for Amazon Nimble Studio since this service began tracking these changes.

Change	Description	Date
AWS managed policy: AmazonNimbleStudio-StudioUser - Updated policy	Amazon Nimble Studio updated a policy to use the latest version of the Identity Store service.	September 22, 2023
AWS managed policy: AmazonNimbleStudio-StudioAdmin - Updated policy	Amazon Nimble Studio updated a policy to use the latest version of the Identity Store service.	September 22, 2023
AWS managed policy: AmazonNimbleStudio-StudioUser - Updated policy	Amazon Nimble Studio updated a policy to allow studio users to view their workstation backups.	December 20, 2022

Change	Description	Date
AWS managed policy: AmazonNimbleStudio-StudioAdmin - Updated policy	Amazon Nimble Studio updated the policy to allow studio admins to view their workstation backups.	December 20, 2022
AWS managed policy: AmazonNimbleStudio-StudioUser - Updated policy	Amazon Nimble Studio updated a policy to allow studio admins to retrieve CloudWatch metrics.	November 11, 2021
AWS managed policy: AmazonNimbleStudio-StudioUser - Updated policy	Amazon Nimble Studio updated the policy to allow studio users to start and stop their workstations.	November 1, 2021
AWS managed policy: AmazonNimbleStudio-StudioAdmin - Updated policy	Amazon Nimble Studio updated the policy to allow studio admins to start and stop their workstations.	November 1, 2021
AWS managed policy: AmazonNimbleStudio-StudioUser - Updated policy	Amazon Nimble Studio updated the policy to conditionally allow access to streaming-session resources based on `nimble:ownedBy` instead of `nimble:createdBy` .	August 16, 2021
AWS managed policy: AmazonNimbleStudio-StudioUser – New policy	Amazon Nimble Studio added a new policy that allows access to resources associated with the studio user and related studio resources in other services.	April 28, 2021

Change	Description	Date
AWS managed policy: AmazonNimbleStudio-StudioAdmin - New policy	Amazon Nimble Studio added a new policy that allows access to resources associated with the studio admin and related studio resources in other services.	April 28, 2021
AWS managed policy: AmazonNimbleStudio-LaunchProfileWorker – New policy	Amazon Nimble Studio added a new policy that allows access to resources needed by Nimble Studio launch profile workers.	April 28, 2021
Amazon Nimble Studio started tracking changes	Amazon Nimble Studio started tracking changes for its AWS managed policies.	April 28, 2021

Cross-service confused deputy prevention

The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action. In AWS, cross-service impersonation can result in the confused deputy problem. Cross-service impersonation can occur when one service (the calling service) calls another service (the called service). The calling service can be manipulated to use its permissions to act on another customer's resources in a way it shouldn't otherwise have permission to access. To prevent this, AWS provides tools that help you protect your data for all services with service principals that have been given access to resources in your account.

We recommend using the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in resource policies to limit the permissions that Identity and Access Management (IAM) gives Amazon Nimble Studio to access your resources. If you use both global condition context keys, the `aws:SourceAccount` value and the account in the `aws:SourceArn` value must use the same account id when used in the same policy statement.

The value of `aws:SourceArn` must be the studio's ARN and `aws:SourceAccount` must be your account id. You won't know what the studio id is until the studio is created because it's generated by Nimble Studio. Once your studio is created, you can update the trust policy with the final studio id set as the `aws:SourceArn`.

The most effective way to protect against the confused deputy problem is to use the `aws:SourceArn` global condition context key with the full ARN of the resource. If you don't know the full ARN of the resource or if you're specifying multiple resources, use the `aws:SourceArn` global context condition key with wildcards (*) for the unknown portions of the ARN. For example, `arn:aws:nimble::123456789012:*`.

Your end users assume your studio role when they sign in to the Nimble Studio portal. When you create your studio, AWS configures the role and evaluates the policy. AWS evaluates the policy every subsequent time one of your users logs in to the Nimble Studio portal. When you create a studio, you can't modify the `aws:SourceArn`. After you finish creating your studio, you can use your studioArn for the `aws:SourceArn`.

The following example is an assume role policy that shows how you can use the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in Nimble Studio to prevent the confused deputy problem.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "identity.nimble.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "StringLike": {
          "aws:SourceArn": "arn:aws:nimble:us-west-2:123456789012:studio/*"
        }
      }
```

```
    }
  ]
}
```

Troubleshooting Amazon Nimble Studio identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Nimble Studio and IAM.

Topics

I'm not authorized to perform an action in Nimble Studio.

If you receive an error that you're not authorized to perform an action, your policies must be updated to allow you to perform the action.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but doesn't have the fictional `nimble:`*`GetWidget`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 nimble:GetWidget on resource: my-example-widget
```

In this case, the policy for the `mateojackson` user must be updated to allow access to the *`my-example-widget`* resource by using the `nimble:`*`GetWidget`* action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I'm not authorized to perform iam:PassRole.

If you receive an error that you aren't authorized to perform the `iam:PassRole` action, then contact your administrator for assistance. Ask them to update your policies to allow you to pass a role to Nimble Studio.

Some AWS services allow you to pass an existing role to that service, instead of creating a new service role or service-linked role. To do this, you need permissions to pass the role to the service.

The following example error occurs when an user named `johndoe` tries to use the console to perform an action in Nimble Studio. However, the action requires the service to have permissions granted by a service role. John doesn't have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/johndoe is not authorized to perform: iam:PassRole
```

In this case, John asks his administrator to update his policies to grant permission to perform the `iam:PassRole` action.

I want to view my access keys.

Amazon Nimble Studio doesn't provide access keys. To learn about secret access keys, see Managing access keys in the IAM User Guide.

> **Important**
>
> Don't provide your access keys to a third party, even to help find your canonical user ID. By doing this, you might give someone permanent access to your account.

When you create an access key pair, you're prompted to save the access key ID and secret access key in a secure location. The secret access key is available only at the time you create it. If you lose your secret access key, add new access keys to your user. You can have a maximum of two access keys. If you already have two, delete one key pair before creating a new one. To view instructions, see Managing access keys in the IAM User Guide.

I'm an administrator and want to allow others to access Nimble Studio.

To allow others to access Nimble Studio, create an IAM entity (user or role) for the person or application that needs access. They will use the credentials for that entity to access AWS. Then, attach a policy to the entity that grants them the correct permissions.

Nimble Studio provides you with the `AmazonNimbleStudio-StudioUser` in the AWS Management Console. The IT admin who manages the Console uses this policy to grant studio access to others.

For a tutorial about using the admin policy, view the Setting up for Nimble Studio guide. To learn how to attach existing policies to users, like user and launch profile policies, see Creating IAM users (console).

For information about importing policies, see Creating your first IAM delegated user and group in the IAM User Guide.

I want to allow people outside of my AWS account to access my Nimble Studio resources.

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Nimble Studio supports these features, see How Amazon Nimble Studio works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the IAM User Guide.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the IAM User Guide.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the IAM User Guide.
- To learn the difference between using roles and resource-based policies for cross-account access, see How IAM roles differ from resource-based policies in the IAM User Guide.

Security event logging and monitoring with Nimble Studio

Monitoring is an important part of maintaining the reliability, availability, and performance of Amazon Nimble Studio and your AWS solutions. Collect monitoring data from all of the parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs.

AWS and Nimble Studio provide tools for monitoring your resources and responding to potential incidents, including Logging Nimble Studio calls using AWS CloudTrail and AWS CloudFormation User Guide.

For more information about how Amazon Nimble Studio works with AWS CloudFormation, including examples of JSON and YAML templates, see the Amazon Nimble Studio resource and property reference in the AWS CloudFormation User Guide. To understand how to use CloudFormation templates, see AWS CloudFormation concepts.

Topics

Logging Nimble Studio calls using AWS CloudTrail

Amazon Nimble Studio is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Nimble Studio. CloudTrail captures all API calls for Nimble Studio as events. The calls captured include calls from the Nimble Studio console and code calls to the Amazon Nimble Studio operations.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Nimble Studio. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Nimble Studio, the IP address from which the request was made, who made the request, when it was made, and additional details.

Nimble Studio information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Nimble Studio, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for Nimble Studio, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs.

For more information, see the following:

Overview for creating a trail

CloudTrail supported services and integrations

Configuring Amazon SNS notifications for CloudTrail

Receiving CloudTrail log files from multiple Regions

Receiving CloudTrail log files from multiple accounts

Nimble Studio actions are logged by CloudTrail and are documented in the Amazon Nimble Studio API Reference. For example, calls to the CreateStudio, GetStudio and DeleteStudio actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another service.

For more information, see the CloudTrail user Identity element.

Understanding Nimble Studio log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

This JSON example shows three actions:

- ACTION_1: CreateStudio
- ACTION_2: GetStudio
- ACTION_3: DeleteStudio

```
{
    "eventVersion": "0",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLE-PrincipalID:EXAMPLE-Session",
        "arn": "arn:aws:sts::111122223333:assumed-role/EXAMPLE-UserName/EXAMPLE-
Session",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE-accessKeyId",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EXAMPLE-PrincipalID",
                "arn": "arn:aws:iam::111122223333:role/EXAMPLE-UserName",
                "accountId": "111122223333",
                "userName": "EXAMPLE-UserName"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-03-08T23:25:49Z"
            }
        }
    },
    "eventTime": "2021-03-08T23:25:49Z",
    "eventSource": "nimble.amazonaws.com",
    "eventName": "CreateStudio",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "EXAMPLE-userAgent",
    "requestParameters": {
        "displayName": "Studio Name",
        "studioName": "EXAMPLE-studioName",
        "userRoleArn": "arn:aws:iam::111122223333:role/EXAMPLE-ServiceRole-User",
        "adminRoleArn": "arn:aws:iam::111122223333:role/EXAMPLE-ServiceRole-Admin"
    },
    "responseElements": {},
    "requestID": "EXAMPLE-requestID",
    "eventID": "EXAMPLE-eventID",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
```

```
    "recipientAccountId": "111122223333"
},
{
    "eventVersion": "0",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLE-PrincipalID:EXAMPLE-Session",
        "arn": "arn:aws:sts::111122223333:assumed-role/EXAMPLE-UserName/EXAMPLE-
Session",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE-accessKeyId",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EXAMPLE-PrincipalID",
                "arn": "arn:aws:iam::111122223333:role/EXAMPLE-UserName",
                "accountId": "111122223333",
                "userName": "EXAMPLE-UserName"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-03-08T23:44:25Z"
            }
        }
    },
    "eventTime": "2021-03-08T23:44:25Z",
    "eventSource": "nimble.amazonaws.com",
    "eventName": "GetStudio",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "EXAMPLE-userAgent",
    "requestParameters": {
        "studioId": "us-west-2-EXAMPLE-studioId"
    },
    "responseElements": null,
    "requestID": "EXAMPLE-requestID",
    "eventID": "EXAMPLE-eventID",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
},
```

```
{
    "eventVersion": "0",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EXAMPLE-PrincipalID:EXAMPLE-Session",
        "arn": "arn:aws:sts::111122223333:assumed-role/EXAMPLE-UserName/EXAMPLE-
Session",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE-accessKeyId",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EXAMPLE-PrincipalID",
                "arn": "arn:aws:iam::111122223333:role/EXAMPLE-UserName",
                "accountId": "111122223333",
                "userName": "EXAMPLE-UserName"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2021-03-08T23:45:14Z"
            }
        }
    },
    "eventTime": "2021-03-08T23:44:14Z",
    "eventSource": "nimble.amazonaws.com",
    "eventName": "DeleteStudio",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "EXAMPLE-userAgent",
    "requestParameters": {
        "studioId": "us-west-2-EXAMPLE-studioId"
    },
    "responseElements": {
        "studio": {
            "adminRoleArn": "arn:aws:iam::111122223333:role/EXAMPLE-ServiceRole-Admin",
            "displayName": "My New Studio Name",
            "homeRegion": "us-west-2",
            "ssoClientId": "EXAMPLE-ssoClientId",
            "state": "DELETING",
            "statusCode": "DELETING_STUDIO",
            "statusMessage": "Deleting studio",
            "studioEncryptionConfiguration": {
                "keyType": "AWS_OWNED_CMK"
```

```
            },
            "studioId": "us-west-2-EXAMPLE-studioId",
            "studioName": "EXAMPLE-studioName",
            "studioUrl": "https://sso111122223333.us-
west-2.portal.nimble.amazonaws.com",
            "tags": {},
            "userRoleArn": "arn:aws:iam::111122223333:role/EXAMPLE-ServiceRole-User"
        }
    },
    "requestID": "EXAMPLE-requestID",
    "eventID": "EXAMPLE-eventID",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```


In the example, you'll notice that the events show the Region, IP address, and other "requestParameters" such as the "userRoleArn" and "adminRoleArn" that will help you identify the event. You can see the time and date in the "creationDate", and the source where the request originated, which is marked as "eventSource": "nimble.amazonaws.com".

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in IAM or AWS STS, that activity is recorded in a CloudTrail event along with other AWS service events in Event history. You can view, search, and download recent events in your AWS account.

AWS CloudTrail captures all API calls for IAM and AWS Security Token Service (AWS STS) as events, including calls from the console and API calls. To learn more about using CloudTrail with IAM and AWS STS, see Logging IAM and AWS STS API calls with AWS CloudTrail.

For more information about CloudTrail, see AWS CloudTrail User Guide .

For information about other monitoring services that Amazon offers, see the Amazon CloudWatch User Guide.

Compliance validation for Amazon Nimble Studio

Amazon Nimble Studio follows the shared responsibility model, and compliance is shared between AWS and our customers.

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying baseline environments on AWS that are security and compliance focused.
- Architecting for HIPAA Security and Compliance on Amazon Web Services – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

  > **Note**
  >
  > Not all AWS services are HIPAA eligible. For more information, see the HIPAA Eligible Services Reference.

- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.

- Amazon GuardDuty – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Infrastructure security in Amazon Nimble Studio

As a managed service, Amazon Nimble Studio is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access Nimble Studio through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Security best practices for Nimble Studio

Amazon Nimble Studio provides a number of security features to consider as you develop and implement your own security policies. The following best practices are general guidelines and don't represent a complete security solution. Because these best practices might not be appropriate or sufficient for your environment, treat them as helpful considerations rather than prescriptions.

Monitoring

Monitoring is an important part of maintaining the reliability, availability, and performance of Nimble Studio and your AWS solutions. For more information about monitoring and responding to events, see Security event logging and monitoring with Nimble Studio.

Data protection

For data protection purposes, we recommend that you protect AWS account credentials and set up individual accounts with AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put sensitive identifying information, such as your customers' account numbers, into free-form fields such as a **Name** field. This includes when you work with Amazon Nimble Studio or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into Amazon Nimble Studio or other services might get picked up for inclusion in diagnostic logs. When you provide a URL to an external server, don't include credentials information in the URL to validate your request to that server.

Permissions

Manage access to AWS resources using users, IAM roles, and by granting the least priviledge to users. Establish credential management policies and procedures for creating, distributing, rotating, and revoking AWS access credentials. For more information, see IAM Best Practices in the IAM User Guide.

Support for Nimble Studio

This section provides support options for Nimble Studio, such how to get help while deploying or using the service and its related applications.

Contents

Nimble Studio forum

If you have questions about Nimble Studio, you can visit the Nimble Studio forum. There you can get answers from the community and AWS forum moderators about Nimble Studio features, technical issues, and troubleshooting help.

Applications support

Nimble Studio provides additional documentation for the following applications.

AWSThinkboxDeadline

For help with your render farm or to learn how Deadline works, see AWSThinkboxDeadline documentation.

Nimble Studio File Transfer

To learn how File Transfer works, see the Nimble Studio File Transfer User Guide.

AWS Support Center

The AWS Support Center is a hub for creating and managing your support cases. It provides access to a variety of resources, including billing and technical solutions, a knowledge center, knowledge center videos, AWS documentation, plus training and certification.

AWS Support plans

AWS Support plans help you optimize performance, stay secure, avoid downtime, and control costs. For more information about AWS Support plans, see Compare AWS Support Plans.

For more information about how AWS can support you, visit the Contact us page.

Document history

- API version: latest
- Latest documentation update: October 2, 2024

The following table describes important changes in each release of the *Nimble Studio Administrator Guide*.

Change	Description	
End of support notice	End of support notice: On October 22, 2024, AWS will discontinue support for Amazon Nimble Studio. After October 22, 2024, you will no longer be able to access the Nimble Studio console or Nimble Studio resources.	October 2, 2024
AWS managed policy updates	Updated the `AmazonNimbleStudio-StudioUser` and `AmazonNimbleStudio-StudioAdmin` policies to use the latest version of the AWS IAM Identity Center service.	September 22, 2023
New service and guide	This is the initial release of Amazon Nimble Studio and the **Amazon Nimble Studio Administrator Guide**.	June 19, 2023

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.