Implementation Guide

Simple File Manager for Amazon EFS

Simple File Manager for Amazon EFS: Implementation Guide

Table of Contents

Deploy a web user interface to manage your Amazon EFS file systems

Publication date: *July 2021 (last update: June 2024)*

Simple File Manager for Amazon EFS helps you to directly interact with data in your Amazon Elastic File System (Amazon EFS) file systems without deploying an Amazon Elastic Compute Cloud (Amazon EC2) instance. It features a web user interface (web UI) where you can browse, upload, and download files in existing EFS file systems, without specialized Linux knowledge. For example, you can use this solution to upload a machine learning (ML) model file to an Amazon EFS file system that is used by an AWS Lambda function for ML inference.

This guide provides infrastructure and configuration information for planning and deploying the Simple File Manager for Amazon EFS in the Amazon Web Services (AWS) Cloud. It includes links to an AWS CloudFormation template that launches and configures the AWS services required to deploy this solution using AWS best practices for security and availability.

This implementation guide provides an overview of the Simple File Manager for Amazon EFS solution, its reference architecture and components, considerations for planning the deployment, and configuration steps for deploying the solution to the Amazon Web Services (AWS) Cloud.

The intended audience for implementing the Simple File Manager for Amazon EFS in their environment includes solution architects, business decision makers, DevOps engineers, data scientists, and cloud professionals.

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

If you want to . . .	Read . . .
Know the cost for running this solution. The estimated cost for running this solution in the US East (N. Virginia) Region is USD $0.78 per month.	Cost
Understand the security considerations for this solution.	Security

If you want to . . .	Read . . .
Know how to plan for quotas for this solution.	Quotas
Know which AWS Regions support this solution.	Supported AWS Regions
View or download the AWS CloudForm ation template included in this solution to automatically deploy the infrastructure resources (the “stack”) for this solution.	AWS CloudFormation template
Access the source code and optionally use the AWS Cloud Development Kit (AWS CDK) to deploy the solution.	GitHub repository

Features and benefits

The Simple File Manager for Amazon EFS solution provides the following features.

Access the solution using a web UI

This solution provides a web interface for you to manage the contents of your Elastic File System (EFS) without requiring a mounted EC2 instance.

Limit EC2 instance access

The solution helps you limit EC2 instance access because it allows users to oversee EFS operations without being granted EC2 permissions.

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

This solution includes a Service Catalog AppRegistry resource to register the solution’s CloudFormation template and its underlying resources as an application in both AppRegistry and Application Manager. With this integration, you can centrally manage the solution’s resources and enable application search, reporting, and management actions.

Use cases

Simplify the management of Elastic File Systems

Some users would like a simpler method to manage their data in Amazon EFS. This solution makes it easy to make changes to the contents of an Elastic File System by using a simple graphical user interface.

Manage Elastic File Systems at a lower cost through serverless technology

Some users don't require dedicated EC2 or networking infrastructure. By eliminating the need to maintain an active EC2 instance, the solution reduces the costs of managing the contents of an Elastic File System.

Concepts and definitions

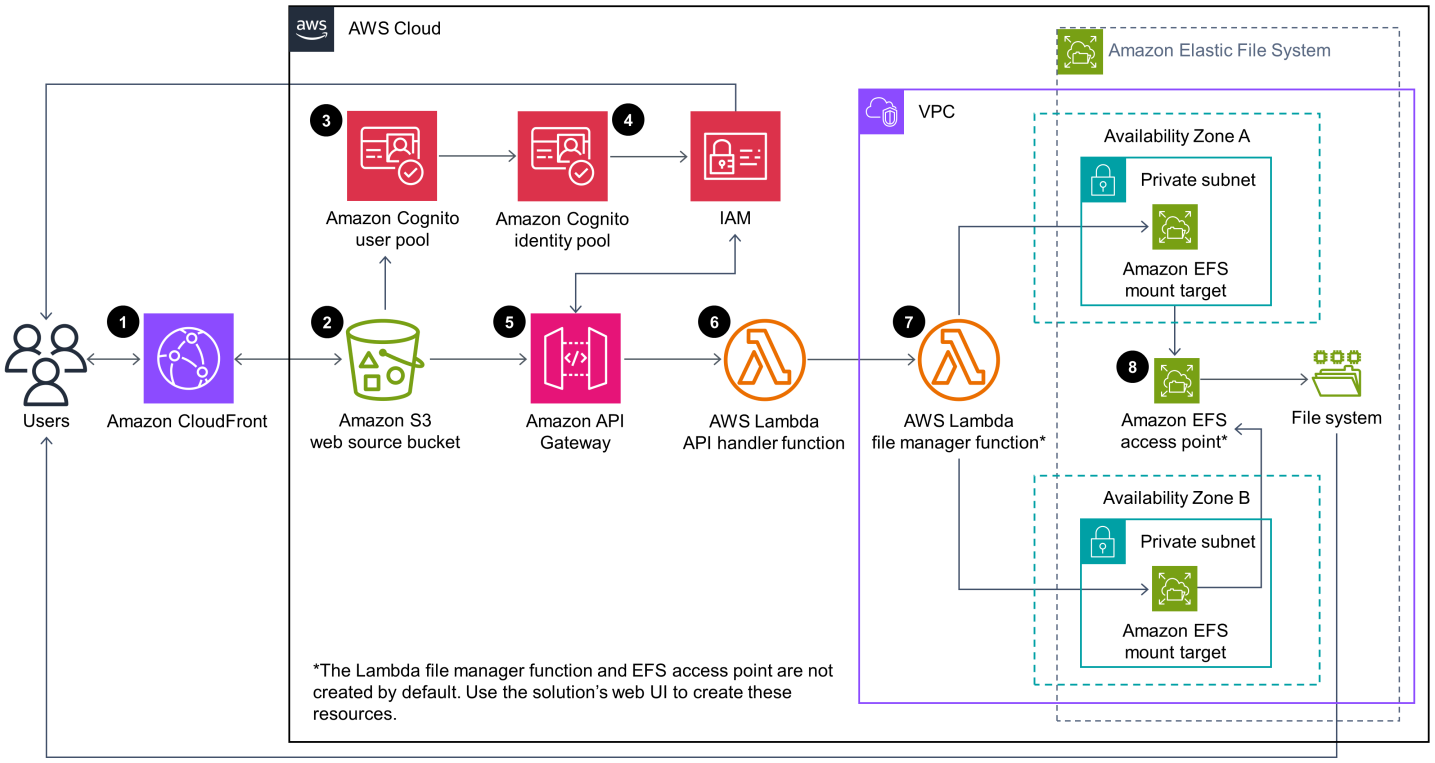
For a general reference of AWS terms, see the AWS Glossary.

Architecture overview

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

Architecture diagram

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



Simple File Manager for Amazon EFS architecture on AWS

The AWS CloudFormation template deploys the following infrastructure:

1. An Amazon CloudFront distribution to serve the Simple File Manager web UI.
2. An Amazon Simple Storage Service (Amazon S3) web source bucket for hosting the CloudFront distribution.
3. An Amazon Cognito user pool to provide a user directory.
4. An Amazon Cognito identity pool to provide federation with AWS Identity and Access Management (IAM) for authentication and authorization to the web UI.

5. An Amazon API Gateway file manager REST API to proxy file system operations from the web UI to your Amazon EFS file system. AWS IAM roles are created for the API to operate.
6. An AWS Lambda API handler function to support the file manager API.
7. An AWS Lambda file manager function to connect to the Amazon EFS file system.
8. An Amazon EFS access point to allow Amazon EFS file system access from AWS Lambda.

> **Note**
>
> The AWS Lambda file manager function (7) and Amazon EFS access point (8) are not automatically deployed by this solution's CloudFormation template. Create these resources after deployment with the solution's web UI.

AWS Well-Architected design considerations

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

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

Operational excellence

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

- Resources defined as IaC using CloudFormation

Security

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

- IAM used for permissions
- Roles follow least-privilege access. Containing the minimum permissions required for the solution to function properly.

Reliability

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

- The solution uses serverless AWS services to perform compute operations (Lambda) to ensure high availability and recovery from service failure
- The solution is automatically tested and deployed each day, looking for failures that may arise as libraries are updated.

Performance efficiency

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

- The solution uses serverless architecture, responding quickly to different needs.

Cost optimization

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

- The solution uses serverless architecture, leaving the customer to pay only for the actions they make.
- The solution prevents the need for an EC2 instance when managing Elastic Filesystems. Avoiding duration based costs.

Sustainability

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

- The solution uses serverless services to minimize the environment impact compared to always-on architecture.

Architecture details

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

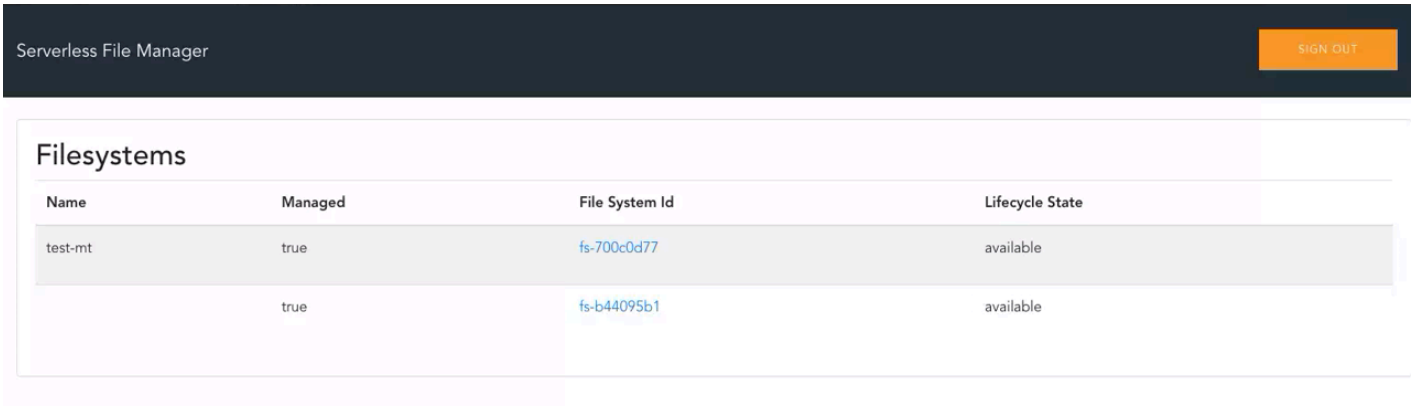
AWS services in this solution

AWS service	Description
Amazon API Gateway	**Core.** Deploys entry point to proxy operations to the Lambda functions.
Amazon CloudFront	**Core.** Creates a Cloudfront Distribution to host the web UI.
Amazon Elastic File System	**Core.** Creates a mountable access point to allow the Lambda functions to access your Amazon EFS.
AWS Lambda	**Core.** Deploys Lambda functions to interact and perform operations on the EFS through the mounted access point.
Amazon S3	**Core.** Deploys a bucket for storing the web source delivered by the CloudFront distribut ion.
Amazon Cognito	**Supporting.** Provides authorization of users to web UI.
AWS Systems Manager	**Supporting.** Provides application-level resource monitoring and visualization of resource operations and cost data.

Web UI

The solution provides a web UI that you can use to interact with your EFS file systems. It is designed to allow you to create AWS Lambda file manager functions that connect to your EFS file system. The file manager API sends file system operations to this Lambda function. Currently, the web UI supports the following file operations:

- Create a directory
- List files
- Delete files
- Upload files
- Download files



Web UI home page

Plan your deployment

This section describes the cost, network security, quotas, and other deployment considerations prior to deploying the solution.

Supported AWS Regions

This solution uses Amazon Cognito, which is not currently available in all AWS Regions. You must launch this solution in a Region where Amazon Cognito is available. For the most current availability by Region, see the AWS Regional Services List.

Simple File Manager for Amazon EFS is available in the following AWS Regions:

Region name	
US East (Ohio)	Asia Pacific (Singapore)
US East (N. Virginia)	Asia Pacific (Sydney)
US West (Northern California)	Asia Pacific (Tokyo)
US West (Oregon)	Europe (Franfurt)
Asia Pacific (Mumbai)	Europe (Ireland)
Asia Pacific (Seoul)	Europe (London)

Cost

You are responsible for the cost of the AWS services used while running the Simple File Manager for Amazon EFS solution, which can vary based on the following factors:

- Number of Amazon API Gateway requests per month.
- Number of AWS Lambda invocations per month.
- Volume of web traffic delivered from Amazon CloudFront per month. (The solution uses CloudFront to deliver the web UI with ultra-low latency performance and high availability to your end users.)

- Number of active users per month authenticated with Amazon Cognito.

As of this revision, the cost for running this solution with the default settings in the US East (N. Virginia) Region is approximately **$0.78/month**. This solution is based entirely on serverless AWS services. Therefore, when the solution is not in use, you only pay for data storage.

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

Sample cost table

The following table provides a sample cost breakdown for deploying this solution with the default parameters in the US East (N. Virginia) Region for one month.

AWS service	Dimensions	Cost [USD]
Amazon EFS	Standard storage/GB-month	$0.30
Amazon API Gateway	50,000 requests/month	$0.17
AWS Lambda	100,000 invocations/month (avg 300 ms duration and 128 MB memory)	$0.08
Amazon CloudFront	Regional data transfer out to internet: first 10 TB Regional data transfer out to origin: all data transfer HTTPS Requests: 50,000 requests/ month X ($0.01/10,000 requests)	$0.085 $0.02 $0.055
Amazon S3	Storage (0.04 GB) and 50,000 get requests/month	$0.02
Amazon Cognito	10 users x ($0.0055/monthly active users (MAUs))	$0.055

AWS service	Dimensions	Cost [USD]
Monthly total:		**$0.78**

This cost estimate does not account for resources related to Amazon VPC that the solution interacts with and for which you may incur additional charges. Prices are subject to change. For full details, refer to the pricing webpage for each AWS service used in this solution.

Security

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

API security

The solution's Amazon API Gateway file manager REST API is secured with an IAM authorizer. Valid AWS Identity and Access Management (IAM) credentials are granted to a user via AWS Security Token Service (AWS STS) after they successfully authenticate with the solution-deployed Amazon Cognito user pool and if they belong to an Amazon Cognito group that has an adequately scoped IAM role associated with it. Refer to the sequence diagram on GitHub for further details.

IAM roles

AWS Identity and Access Management (IAM) roles allow customers to assign granular access policies and permissions to services and users on the AWS Cloud. This solution creates IAM roles that grant the solution's AWS Lambda functions access to create Regional resources. The solution also creates the EFSFileManagerIamRole IAM role that has Amazon API Gateway **execute-api** permissions on all file manager API endpoints.

Amazon CloudFront

This solution deploys a web UI hosted in an Amazon Simple Storage Service (Amazon S3) bucket. To help reduce latency and improve security, this solution includes an Amazon CloudFront distribution with an origin access identity, which is a CloudFront user that provides public access to

the solution's website bucket contents. For more information, refer to Restricting Access to Amazon S3 Content by Using an Origin Access Identity in the *Amazon CloudFront Developer Guide*.

AWS Lambda

When you create AWS Lambda file manager functions, you must ensure that the default settings fit your use case and security requirements. By allowing the default settings, the users created in the Simple File Manager for Amazon EFS Amazon Cognito user pool will have full access to files present in the directory specified.

Limitations

As of October 2022, uploads and downloads with this solution have been tested with files up to 1 GB in size. This solution supports larger file uploads, but anything larger than 1 GB has not been tested.

This solution assumes that you have an existing Amazon EFS Filesystem deployed with mount targets configured in the region you are deploying the solution in. The solution also requires that the security group associated with the filesystem you are attempting to manage has permissible ingress and egress rules for allowing NFS traffic to the solutions File Manager Lambda function. The solution will prevent you from creating a manager function if it does not detect correct rules in place. For additional details on this subject, refer to the Controlling network access to Amazon EFS file systems page in the Amazon EFS user guide.

Quotas

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

Quotas for AWS services in this solution

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

Use the following links to go to the page for that service. To view the service quotas for all AWS services in the documentation without switching pages, view the information in the Service endpoints and quotas page in the PDF instead.

AWS CloudFormation quotas

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

Deploy the solution

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

Deployment process overview

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

Before you launch the solution, review the cost, architecture, network security, and other considerations discussed earlier in this guide.

Time to deploy: Approximately 15 minutes

> **Note**
>
> If you have previously deployed this solution, see Update the solution for update instructions.

Step 1. Launch the stack

- Launch the AWS CloudFormation template into your AWS account.
- Review the template's parameters and enter or adjust the default values as needed.

Step 2. Reset auto-generated password

- Sign in and create a new password.

Step 3. Create a file manager Lambda function

- Use AWS Lambda to create a file manager function for your Amazon EFS file system.

Step 4. Create Amazon Cognito users

- Create Amazon Cognito users for all your users and add them to the Amazon Cognito group.

AWS CloudFormation template

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

View template simple-file-manager-for-amazon-efs.template

file-manager-for-amazon-efs.template - Creates the solution's API Gateway REST API and instantiates the nested templates for the application. Use this template to launch the solution and all associated components. The default configuration deploys Amazon API Gateway, AWS Lambda, Amazon S3, Amazon CloudFront, and Amazon Cognito, but you can customize the template to meet your specific needs.

The following nested templates are automatically deployed:

efs-file-manager-web.yaml - This nested template creates the web UI resources:Amazon CloudFront distribution, Amazon S3 hosting bucket, AWS IAM roles, and AWS CloudFormation custom resources to support automated deployment.

efs-file-manager-auth.yaml - This nested template creates the authentication and authorization resources: Amazon Cognito User Pool, Amazon Cognito identity pool, Amazon Cognito Application Client, AWS IAM role for application access, and an AWS CloudFormation custom resource to perform Amazon Cognito Role Mapping.

The following template is integrated with the solution's web UI to create a file manager Lambda function, which provides the solution access to a file system:

efs-file-manager-ap-lambda.template - Creates a file manager AWS Lambda function and Amazon EFS access point for a specified Amazon EFS file system. The template does not require customizations; however, you can download this template from the Simple File Manager for Amazon EFS GitHub repository to customize or repurpose the template for your own needs.

Step 1: Launch the stack

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

Time to deploy: Approximately 15 minutes

1. Sign in to the AWS Management Console and select the button to launch the `simple-file-manager-for-amazon-efs.template` AWS CloudFormation template.

   Launch solution

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

   > **Note**
   >
   > This solution uses Amazon Cognito, which is not currently available in all AWS Regions. You must launch this solution in an AWS Region where Amazon Cognito is available. For the most current availability by Region, refer to the AWS Regional Services List.

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

Parameter	Default	Description
AdminEmail	*<Requires input>*	The email address of the user that will use the solution. This user will have access to the files and file systems within the AWS account.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review and create** page, review and confirm the settings. Check the boxes acknowledging that the template will create AWS Identity and Access Management (IAM) resources and requires certain capabilities.

9. Choose **Submit** to deploy the stack.

   You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a CREATE_COMPLETE status in approximately 15 minutes.

Step 2. Reset auto-generated password

During stack creation, the solution sends you an email containing your initial login credentials.

1. Sign in to the AWS CloudFormation console.
2. Select the solution's root stack.
3. Select the **Outputs** tab and choose **EFSFileSimpleWebsiteUrl** in the **Value** column.
4. To sign in, use the initial credentials sent to the admin email.
5. Follow the prompts to create a new password.

   Upon successful authentication, the application routes you to the web UI home page, where all the EFS file systems are displayed in your account for the Region where the stack was deployed.
6. For convenience, save the address.

Step 3. Create a file manager Lambda function

To grant Simple File Manager for Amazon EFS access to a file system, create a file manager Lambda function.

1. Log in to the solution's web UI. For the site URL, refer to Step 2. Reset auto-generated password.
2. Under **Filesysystems**, select the **false** link.
3. On the **Create file manager lambda** page, enter the following information:
   - **User ID** - The numeric POSIX user ID that Lambda will use to make file system requests.
   - **Group ID** - The numeric POSIX group ID that Lambda will use to make file system requests.
   - **Path** - Top of Form
   - The file system directory that the solution will use as the root directory. Any files present in this directory are accessible to the application.

     If you are unsure what the options are, the default values work for most Simple File Manager for Amazon EFS use cases.

4. Choose **Submit** and wait for the application to complete the request.
5. After completion, you are routed back to the web UI's home page.

> **Note**
>
> Lambda can take several minutes to provision a new function. Allow 1-2 minutes if the managed state returns **Creating** and refresh the page.

6. The link previously labeled **false** now returns **true** and the file system ID is now an active link.
7. Select the file system ID link to access the file system.

Step 4. Create Amazon Cognito users

This solution uses Amazon Cognito to manage all users and authentication. It creates a user for you during deployment and sends an email at the address provided with temporary credentials.

Use the following procedure to create additional users:

1. Sign in to the AWS Cognito console.
2. Choose **Manage User Pools.**
3. Choose **SimpleFileManagerUserPool.**
4. In the navigation pane, under **General Settings**, choose **Users and groups**.
5. From the **Users** tab, choose **Create user**.
6. In the **Create user** box, enter values for all required fields.

Form field	Required?	Description
Username	Yes	The user name that you will use to log in to Simple File Manager for Amazon EFS. For convenience, it is best to match this value to the Email form field.
Send an invitation	Yes (email only)	When selected, sends a notification as a reminder

Form field	Required?	Description
		of the temporary password. Select **Email only**. If you select SMS (default), an error message displays, but the user is still created.
Temporary Password	Yes	Enter a temporary password. The user is forced to change this when they log in to Simple File Manager for Amazon EFS for the first time.
Phone Number	No	Enter a phone number in international format, for example, +44. Ensure that the **Mark phone number as verified?** box is selected.
Email	Yes	Enter a valid email address. Ensure that the **Mark email as verified?** box is selected.

7. Choose **Create user**.
8. Choose **Groups**.
9. Choose the group with the description **User group for Simple File Manager Admins**.
10. Choose **Add users.**
11. Choose the **plus icon** next to the user name of the user you just created.
12. Repeat this process to create as many users as you need.

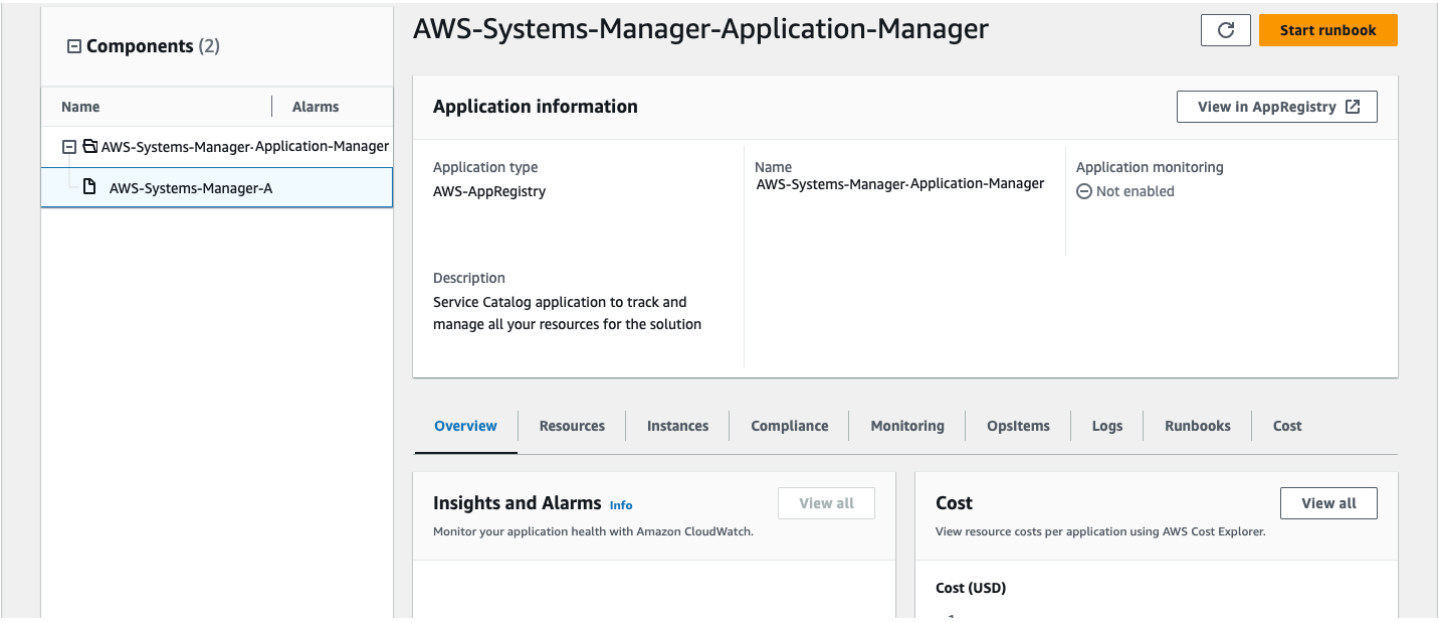
Monitor the solution with Service Catalog AppRegistry

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

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

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

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



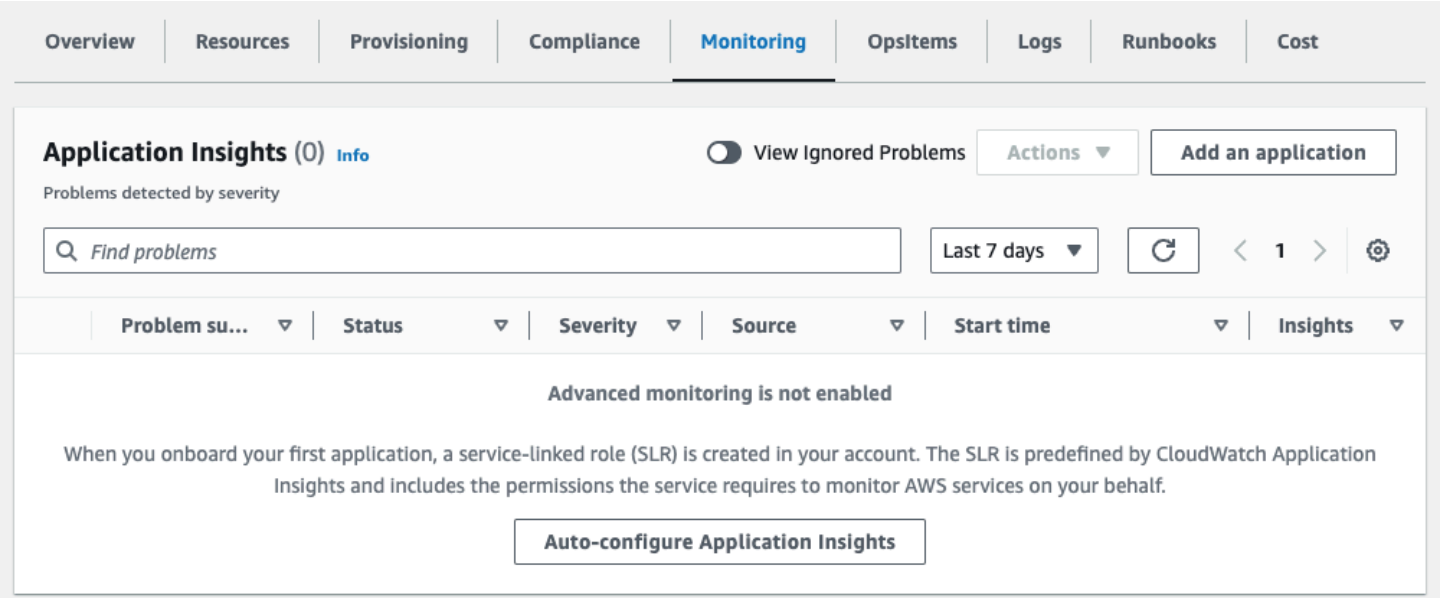
Solution stack in Application Manager

Activate CloudWatch Application Insights

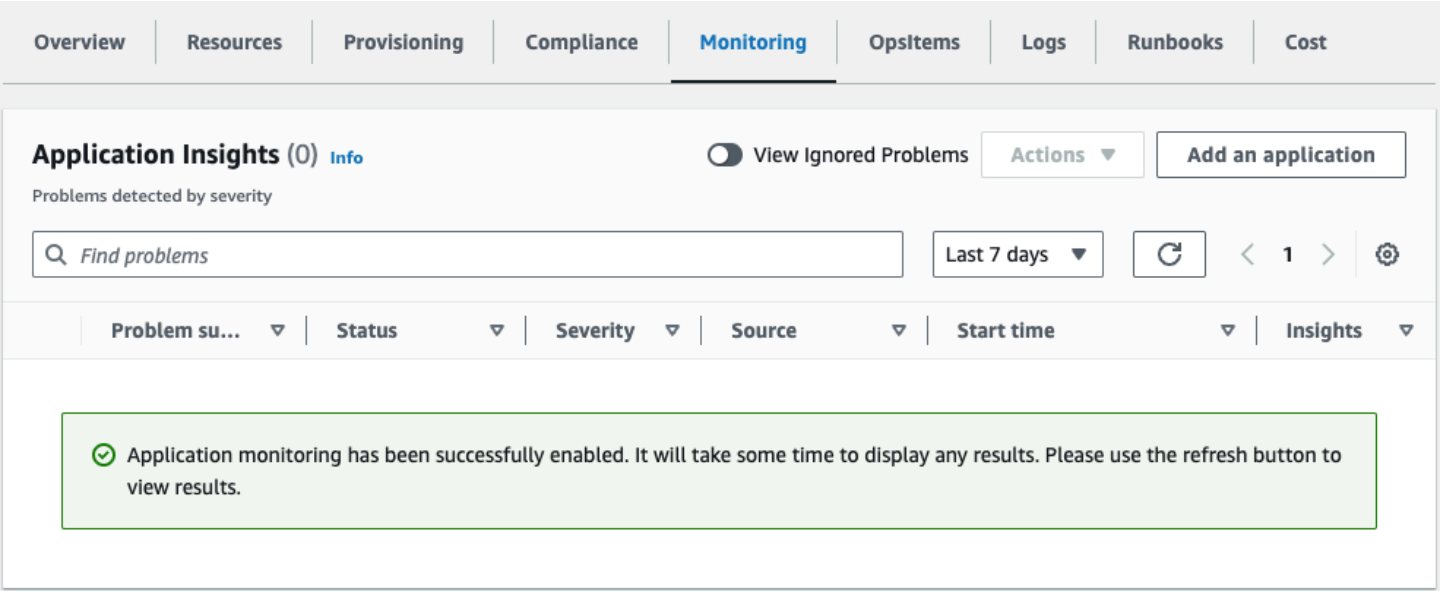
1. Sign in to the Systems Manager console.

2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, search for the application name for this solution and select it.

   The application name will have App Registry in the **Application Source** column, and will have a combination of the solution name, Region, account ID, or stack name.
4. In the **Components** tree, choose the application stack you want to activate.
5. In the **Monitoring** tab, in **Application Insights**, select **Auto-configure Application Insights**.



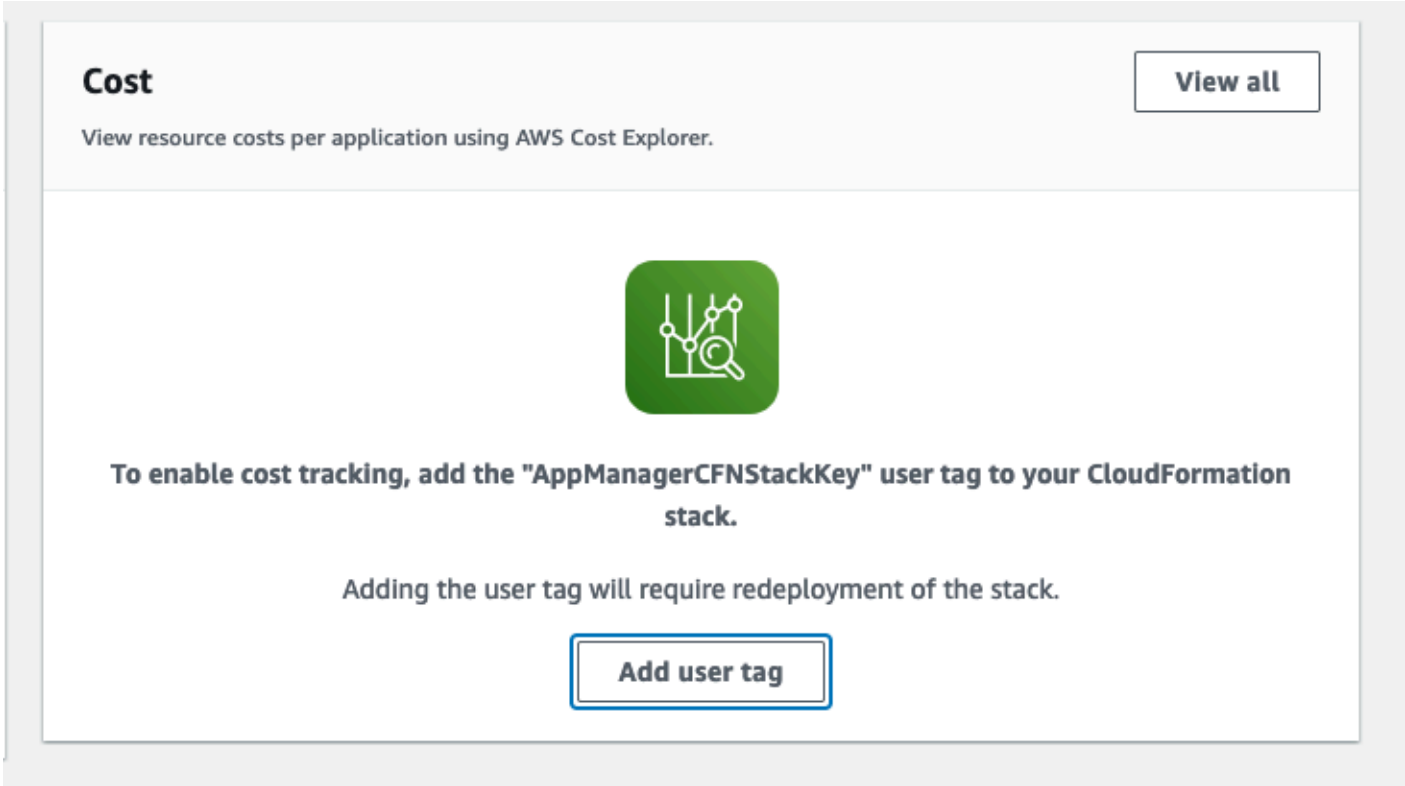
Monitoring for your applications is now activated and the following status box appears:



Confirm cost tags associated with the solution

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

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



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

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

Activate cost allocation tags associated with the solution

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

To activate cost allocation tags:

1. Sign in to the AWS Billing and Cost Management and Cost Management console.
2. In the navigation pane, select **Cost Allocation Tags**.
3. On the **Cost allocation tags** page, filter for the `AppManagerCFNStackKey` tag, then select the tag from the results shown.
4. Choose **Activate**.

AWS Cost Explorer

You can see the overview of the costs associated with the application and application components within the Application Manager console through integration with AWS Cost Explorer. Cost Explorer helps you manage costs by providing a view of your AWS resource costs and usage over time.

1. Sign in to the AWS Cost Management console.
2. In the navigation menu, select **Cost Explorer** to view the solution's costs and usage over time.

Update the solution

If you have previously deployed the solution, follow this procedure to update the solution's CloudFormation stack to get the latest version of the solution's framework.

1. Sign in to the AWS CloudFormationconsole, select your existing **SimpleFileManager** CloudFormation stack, and select **Update**.
2. Select **Replace current template**.
3. Under Specify template:
   a. Select **Amazon S3 URL**.
   b. Copy the link of the `simple-file-manager-for-amazon-efs.template` the section called "AWS CloudFormation template".
   c. Paste the link in the **Amazon S3 URL** box.
   d. Verify that the correct template URL shows in the **Amazon S3 URL** text box, and choose **Next**. Choose **Next** again.
4. Under **Parameters**, review the parameters for the template and modify them as necessary.
5. Choose **Next**.
6. On the **Configure stack options** page, choose **Next**.
7. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources.
8. Choose **View change set** and verify the changes.
9. Choose **Update stack** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a UPDATE_COMPLETE status in approximately 15 minutes.

Update post deployment resources

For updates that include changes to post deployment resources, which are outlined in the release notes, you must follow these additional steps to update the solution. Complete the standard update process before proceeding with these steps.

1. Log in to the solution's web UI.

2. Choose the **true** link for a file system.
3. Choose **Delete**.
4. Wait for the **deleting** status to change to **false**.
5. Recreate the file manager Lambda function. For details, refer to Step 3. Create a file manager Lambda function.
6. Repeat this process for each managed file system in the **Filesystems** table.

Troubleshooting

If you need help with this solution, contact AWS Support to open a support case for this solution.

Contact AWS Support

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

Create case

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

How can we help?

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

Additional information

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

Help us resolve your case faster

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

Solve now or contact us

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

Uninstall the solution

You can uninstall the Simple File Manager for Amazon EFS solution from the AWS Management Console or by using the AWS Command Line Interface. You must manually delete the file manager Lambda and EFS access point created by this solution before you delete the CloudFormation stack.

Delete file manager Lambda and EFS access point

1. Log in to the solution's web UI.
2. Choose the **true** link for a filesystem.
3. Choose **Delete**.
4. Wait for the **deleting** status to change to **false**.
5. Repeat this process for each managed file system in the **Filesystems** table.

Using the AWS Management Console

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

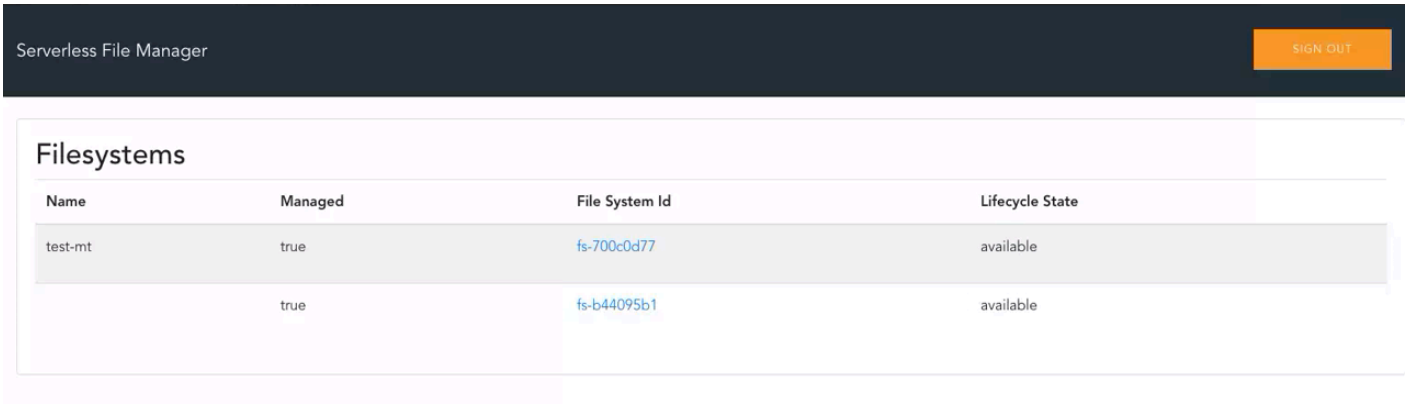
Using AWS Command Line Interface

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

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

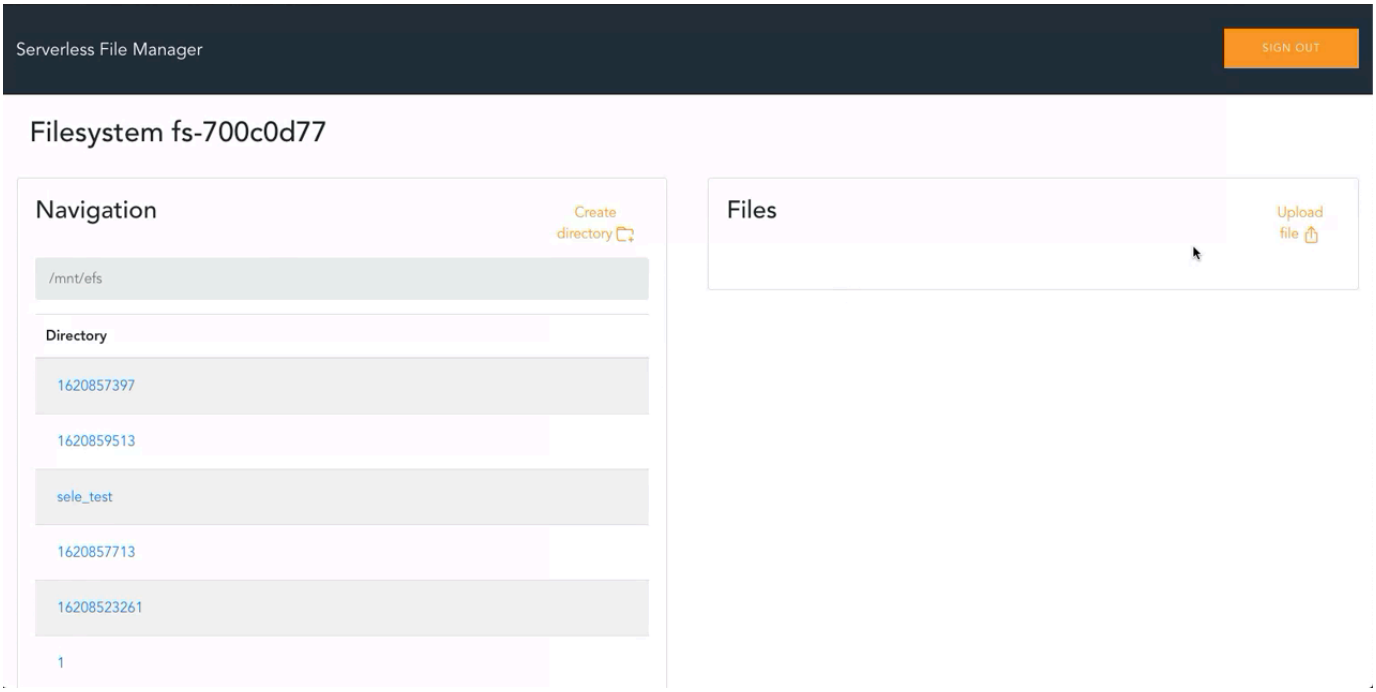
Use the solution

1. Log in to the solution's web UI. For the site URL, refer to Step 2. Reset auto-generated password.
2. Under **Filesysystems**, select the file system ID link to access the file system.



Web UI home page

3. Choose the relevant button or directory to perform the relavant operation:
   - Create a directory
   - List files
   - Delete files
   - Upload files
   - Download files



Web UI within selected file system

4. To return to the home page, choose **Serverless File Manager**.

Developer guide

This section provides the source code for the solution.

Source code

Visit the Simple File Manager for Amazon EFS GitHub repository to download the source files for this solution and to share your customizations with others. Refer to the README.md file for additional information.

Reference

This section includes pointers to related resources and a list of builders who contributed to this AWS Solution.

Related AWS documentation

- Using Amazon EFS with Lambda
- Amazon Cognito user pools
- Amazon Cognito identity pools

Contributors

- Brandon Dold
- Eddie Goynes
- Andrea Amorosi
- Simon Krol
- Garvit Singh

Revisions

Date	Change
July 2021	Initial release (v1.4.0)
August 2022	Release version 1.4.1: Updated the Python version in the Lambda functions from version 3.6 to version 3.8.
October 2022	Release version 1.5.0: AppRegistry: AppRegistry Application Stack Association and Application Insights. For more information about new features, refer to the CHANGELOG.md file in the GitHub repository.
April 2023	Release version 1.5.1: Minor updates and bug fixes. Also, mitigated impact caused by new default settings for S3 Object Ownership (ACLs disabled) for all new S3 buckets. For more information about new features, refer to the CHANGELOG.md file in the GitHub repository.
June 2023	Release version 1.5.2: Fix for changes to Amazon EFS TagResource permission. For more information about new features, refer to the CHANGELOG.md file in the GitHub repository.
September 2023	Release version 1.5.3: Merged Amazon S3 web bucket policies. Added downline dependencies to `NOTICE.txt` . Upgraded Node.js and Python versions and fixed NPM vulnerabilities. For more information about new features, refer to the CHANGELOG.md file in the GitHub repository.

Date	Change
October 2023	Release version 1.5.4: Security patch. For more information, refer to the CHANGELOG.md file in the GitHub repository.
November 2023	Release version 1.5.5: Updated crypto.js and react-dev-tools dependencies to fix security vulnerabilities. Updated urllib3 dependency to v1.26.18. For more information, refer to the CHANGELOG.md file in the GitHub repository.
November 2023	Documentation update: Added Confirm cost tags associated with the solution to the Monitoring the solution with AWS Service Catalog AppRegistry section.
April 2024	Release version 1.5.6: Updated Axios and various sub-dependency versions to resolve security vulnerabilities. For more information, refer to the CHANGELOG.md file in the GitHub repository.
May 2024	Release version 1.5.7: Updated API Handler Python runtime to 3.11 due to Python 3.8 Lambda runtime deprecation. Updated spoke template descriptions to include suffix. For more information, refer to the CHANGELOG .md file in the GitHub repository.
June 2024	Release version 1.5.8: Updated package versions to resolve security vulnerabilities. For more information, refer to the CHANGELOG .md file in the GitHub repository.

Notices

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